



उच्च संकाय प्रशिक्षण केंद्र के अंतर्गत
CENTRE OF ADVANCED FACULTY TRAINING



प्रशिक्षण पुस्तिका-I
Training Manual-I

on

कृषि आँकड़ों के मॉडलिंग एवं पूर्वानुमान के लिए सांख्यिकी एवं मशीन लर्निंग तकनीके
**Statistical and Machine Learning Techniques for Modeling
and Forecasting Agricultural Data**

दिसम्बर 20, 2019 - जनवरी 09, 2020

December 20, 2019 - January 09, 2020

पाठ्यक्रम समन्वयक	:	डॉ मृनमय राय
Course Coordinator	:	Dr. Mrinmoy Ray
पाठ्यक्रम सहसमन्वयक	:	श्री शिवस्वामी जी.पी.
Co-Course Coordinator	:	Mr. Shivaswamy G P
पाठ्यक्रम सहसमन्वयक	:	डॉ हरीश कुमार एच.वी.
Co-Course Coordinator	:	Dr. Harish Kumar H V

पूर्वानुमान एवं कृषि प्रणाली मॉडलिंग प्रभाग
भा.कृ.अ.प. - भारतीय कृषि सांख्यिकी अनुसंधान संस्थान
लाइब्रेरी एवेन्यू , पूसा नई दिल्ली -110012

**Division of Forecasting and Agricultural Systems Modeling
ICAR-Indian Agricultural Statistics Research Institute
Library Avenue, Pusa, New Delhi-110012**

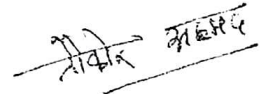
2019-2020

प्राक्कथन

भा.कृ.अनु.प.-भारतीय कृषि सांख्यिकी अनुसंधान संस्थान देश में कृषि सांख्यिकी, संगणक अनुप्रयोग और जैवसूचना विज्ञान के विषयों में एक प्रमुख संस्थान है। संस्थान सांख्यिकीय आनुवांशिकी, परीक्षण अभिकल्पना, प्रतिदर्शप पद्धतियाँ, बायोमेट्रिक्स, सांख्यिकीय मॉडलिंग, पूर्वानुमान तकनीक, अर्थमिति, संगणक अनुप्रयोग और जैव-सूचना विज्ञान जैसे विभिन्न क्षेत्रों में अनुसंधान और प्रशिक्षण कार्यक्रम आयोजित करने में व्यस्त हैं। सांख्यिकीय मॉडलिंग कृषि में विविध अनुप्रयोगों के कारण अनुसंधान का एक महत्त्वपूर्ण क्षेत्र है तथा नीति निर्माताओं और कृषि वैज्ञानिकों के लिए उपयोगी है। कृषि आँकड़ों के मॉडलिंग एवं पूर्वानुमान के लिए सांख्यिकी एवं मशीन लर्निंग तकनीक नामक प्रशिक्षण कार्यक्रम सिद्धांत और अनुप्रयोगों का एक मिश्रण है। पाठ्यक्रम के अन्तर्गत विभिन्न विषय शामिल किए गए हैं, किन्तु सीमित नहीं है; फज़ी-रेखीय समाश्रयण, लॉज़िस्टिक समाश्रयण, क्वान्टाइल प्रतिगमन, अरेखिय सांख्यिकी मॉडल, फसल पूर्वानुमान तकनीकें, एरिमा और विरिमा काल श्रृंखला मॉडलिंग, फज़ीकाल श्रृंखला मॉडलिंग, इकोनोमेट्रिक मॉडलिंग, काउंट डाटा मॉडलिंग, संरचनात्मक काल श्रृंखला मॉडलिंग, अरेखिय काल श्रृंखला मॉडलिंग, कृत्रिम तंत्रिका नेटवर्क, अनुवांशिक एलगोरिथम, सपोर्ट वेक्टर मशीन, हाईब्रिड काल श्रृंखला मॉडलिंग, कार्ट, स्टोकास्टिक वोलेटिलिटी मॉडल, मार्कोव चेन विश्लेषण, बेसियन काल श्रृंखला मॉडल, रिसेम्पलिंग आधारित प्रतिगमन, रिमोट सेंसिंग, जी.आई.एस तथा कृषि में पूर्वानुमान तकनीक मॉडल का अनुप्रयोग इत्यादि।

इस पाठ्यक्रम के संकाय प्रख्यात सांख्यिकीयविद हैं। जो सांख्यिकीय मॉडलिंग के क्षेत्र में निपुण हैं। इसके अलावा, अतिथि संकाय अपने कार्य क्षेत्र में विषय ज्ञाता होने के कारण प्रसिद्ध शोधकर्ता हैं और वे MNCF, New Delhi; ICAR-IARI, New Delhi; ICAR-IISS Bhopal; B.C.K.V Mohanpur; ICAR-IIRR, Hyderabad; दिल्ली विश्वविद्यालय और केंद्रीय रेशम बोर्ड जैसे प्रतिष्ठित संगठनों से हैं। सदर्थ पुस्तिका प्रतिभागियों के लिए भविष्य में उपयोगी ज्ञान धरोहर के रूप में सहायक होगी। मैं आशा करता हूँ इस प्रशिक्षण से प्राप्त अनुभव उन्हें अधिक कुशलता से अनुसंधान करने के लिए सक्षम बनाएंगे, इस बहुमूल्य संदर्भ पुस्तिका को समय पर तैयार करने के लिए पाठ्यक्रम समन्वयक, पाठ्यक्रम सह-समन्वयक और आयोजन समिति बधाई के पात्र हैं।

नई दिल्ली
दिसम्बर 20, 2019



(तौकीर अहमद)

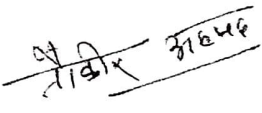
निदेशक (का), भा.कृ.अनु.प.-भा.कृ.सां.अनु.सं.

FOREWORD

ICAR-Indian Agricultural Statistics Research Institute is a premier Institute in the disciplines of Agricultural Statistics, Computer Application and Bioinformatics in the country. The Institute has been engaged in conducting research and organizing training programmes in various areas, like Statistical Modelling, Forecasting Techniques, Design of Experiments, Sampling Techniques, Statistical Genetics and Genomics, Computer Applications and Bioinformatics. The present training programme on “Statistical and Machine Learning Techniques for Modeling and Forecasting Agricultural Data” has been planned in such a way that it is a blend of theory and applications. The aim of this training programme is to familiarize the Faculty members/ Scientists/ Researchers at various State Agricultural Universities/ ICAR Institutes with statistical techniques along with machine learning based models for modeling and forecasting of agricultural data in order to draw statistically valid inferences and to help them in upgrading the research, teaching and training skills. Moreover The various topics covered under the course include, but not limited to: Logistic Regression, Quantile Regression, Nonlinear Statistical Models, Crop Forecasting Techniques, ARIMA and Hierarchical time-series modelling, STARMA, VARIMA Time-Series Modelling, Fuzzy Time-Series Modelling, Econometric Modeling, Count Data Modeling, Nonlinear Time-Series Modelling, Artificial Neural Network, Recurrent Neural Network, Genetic Algorithms, Support Vector Machine, Hybrid Time Series Modeling, CART, Stochastic Volatility Models, Bayesian Time Series Modeling, Resampling based Regression, Remote Sensing and GIS etc. along with conventional topics.

The faculty for this course comprises eminent statisticians from ICAR-IASRI, well established in the field of Modeling and Forecasting. Besides, the guest faculties are renowned researchers having sound knowledge in their fields of specialization and also are from reputed organizations like MNCFC, New Delhi; ICAR-IARI, New Delhi; ICAR-IISS Bhopal; B.C.K.V Mohanpur; ICAR-IIRR, Hyderabad; University of Delhi, New Delhi and Central Silk Board. The ‘Reference Manual’ brought out should serve as a useful wealth of knowledge to the participants for their future use. I am sure that the experience gained from this training will enable them to conduct research more efficiently. I wish to complement Course Coordinators and the Organizing Committee for bringing out this valuable document on time.

New Delhi
December 20, 2019


(Tauqueer Ahmad)
Director (A), ICAR-IASRI

आमुख


भारतीय कृषि सांख्यिकी अनुसंधान संस्थान, कृषि सांख्यिकी, संगणक अनुप्रयोगों और जैव सूचना विज्ञान के क्षेत्रों में उपक्रम अनुसंधान, शिक्षा और प्रशिक्षण के लिए एक प्रमुख राष्ट्रीय संस्थान माना जाता है। संस्थान राष्ट्रीय कृषि अनुसंधान प्रणाली एवं राष्ट्रीय कृषि सांख्यिकी प्रणाली में योगदान करने एवं इन्हें सुदृढ़ बनाने के लिए सलाहकार एवं परामर्श सेवाएँ प्रदान करने हेतु विभिन्न महत्वपूर्ण योगदान प्रदान कर रहा है जिसका सीधा प्रभाव राष्ट्रीय नीतियों पर पड़ता है, इस कारण संस्थान को गर्वित स्थान प्राप्त है। सूचना प्रौद्योगिकी के क्षेत्र में प्रगति के साथ, संस्थान वर्तमान जरूरतों और कार्य-पद्धति की चुनौतियों तथा कृषि अनुसंधान की गुणवत्ता के लिए अनुकूल परिस्थितियों उपलब्ध करा रहा है। कृषि के क्षेत्र में सांख्यिकीय मॉडलिंग और पूर्वानुमान संस्थान में अनुसंधान के महत्वपूर्ण विषयों में से एक है। समस्या के महत्व को ध्यान में रखते हुए संस्थान के वैज्ञानिक कृषि के विभिन्न उप कार्यक्षेत्रों में पूर्वानुमान के लिए विभिन्न मॉडलिंग विधियों का अध्ययन करने में जुटे हुए हैं।

भा.कृ.अ.प. और राज्य कृषि विश्वविद्यालयों की क्षमता को मजबूत बनाने के लिए और हमारे अनुसंधान को आवश्यकता के अनुसार एवं विष्वक्स्तर पर प्रतिस्पर्धी बनाने के लिए यह महत्वपूर्ण है कि विभिन्न अनुसंधान गतिविधियों में कार्यरत वैज्ञानिकों को कृषि में पूर्वानुमान के संदर्भ में महत्वपूर्ण क्षेत्र सांख्यिकीय मॉडलिंग के आधुनिक विकास से अवगत करवाया जाए। इसी संदर्भ में, कृषि सांख्यिकी एवं संगणक अनुप्रयोग में पूर्वानुमान एवं कृषि प्रणाली मॉडलिंग प्रभाग में उच्च संकाय प्रशिक्षण कार्यक्रम के अन्तर्गत भारतीय कृषि सांख्यिकी अनुसंधान संस्थान, नई दिल्ली, शिक्षा प्रभाग, भारतीय कृषि अनुसंधान परिषद्, नई दिल्ली के संरक्षण में 20 दिसम्बर 2019 से 09 जनवरी, 2020 तक "कृषि आँकड़ों के मॉडलिंग एवं पूर्वानुमान के लिए सांख्यिकी एवं मशीन लर्निंग तकनीक" प्रशिक्षण का आयोजन कर रहा है। प्रशिक्षण कार्यक्रम का उद्देश्य कृषि क्षेत्र में पूर्वानुमान के लिए वर्तमान सांख्यिकीय मॉडलिंग विधिया तथा कृषि डेटा के मॉडलिंग और पूर्वानुमान के लिए मशीन लर्निंग आधारित मॉडल को विभिन्न सॉफ्टवेयर (SAS, R, PYTHON, तथा STATA) के माध्यम से विभिन्न राज्य कृषि विश्वविद्यालयों/भारतीय कृषि अनुसंधान परिषद् के संस्थानों में संकाय सदस्यों/वैज्ञानिकों को अवगत करना है। इससे उन्हें अनुसंधान, शिक्षण और प्रशिक्षण में अपनी क्षमताओं को उन्नत करने में सहायता मिलेगी।

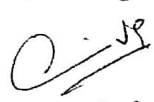
पाठ्यक्रम में मॉडलिंग और कृषि डेटा के पूर्वानुमान के लिए सांख्यिकीय और मशीन सीखने की तकनीक दोनों को शामिल करना शामिल है। पाठ्यक्रम में विभिन्न विषय हैं परन्तु सीमित नहीं हैं: फ़जी रेखीय समाश्रयण, लॉजिस्टिक समाश्रयण, क्वान्टाइल प्रतिगमन, अरेखिय सांख्यिकी मॉडल, फसल पूर्वानुमान तकनीक, एरिमा और पदानुकमित समय-श्रृंखला मॉडलिंग, स्टारमा विरिमा काल श्रृंखला मॉडलिंग, फजी काल श्रृंखला मॉडलिंग, इकोनोमेट्रिक मॉडलिंग, काउंट डाटा मॉडलिंग, संरचनात्मक काल श्रृंखला मॉडलिंग, अरेखिय काल श्रृंखला मॉडलिंग, कृत्रिम तंत्रिका नेटवर्क, आवर्तक तंत्रिका नेटवर्क, आनुवांषिक एलगोरिथम, स्पोर्ट वेक्टर मशीन, हाइब्रिड काल श्रृंखला मॉडलिंग, कार्ट, रेन्डम फोरेस्ट तकनीक, स्टोकेस्टिक वोलेटिलिटी मॉडल, मार्कोव चैन विश्लेषण, बेसियन काल श्रृंखला मॉडल ए रिसेम्पलिंग आधारित प्रतिगमन, रिमोट सेंसिंग, जी.आई.एस तथा कृषि में पूर्वानुमान तकनीक मॉडल का अनुप्रयोग इत्यादि साथ पारंपरिक विषयों के बारे में जानकारी देना है।

हम संस्थान के संकाय और अतिथि संकाय का धन्यवाद करते हैं जिन्होंने अपना बहुमूल्य समय समर्पित कर इस पाठ्यक्रम को सार्थक और सफल बनाने में सहायता की तथा जिनके अथक प्रयासों से यह संदर्भ पुस्तिका समय पर तैयार हो सकी। इस प्रशिक्षण का आयोजन करने के लिए शिक्षा प्रभाग भारतीय कृषि अनुसंधान परिषद्, नई दिल्ली द्वारा प्रदान की गई आवश्यक धन राशि के लिए हम धन्यवाद करते हैं। हम विभिन्न भा.कृ.अ.प. संस्थानों और राज्य कृषि विश्वविद्यालयों के आभारी हैं जिन्होंने अपने वैज्ञानिको/प्रोफेसरों को इस प्रशिक्षण में प्रतिभागिता हेतु नियुक्त किया। हम डॉ. तौकीर अहमद, निदेशक, भा.कृ.सां.अ.सं. के कृतज्ञ हैं, जिन्होंने हमें इस पाठ्यक्रम को आयोजित करने का उत्तरदायित्व सौंपा। हम डॉ. के.एन. सिंह, प्रधान पूर्वानुमान एवं कृषि प्रणाली मॉडलिंग प्रभाग के आभारी हैं जिन्होंने बहुमूल्य मार्गदर्शन किया और पाठ्यक्रम के सुचारु रूप से संचालन के लिए आवश्यक सुविधाएँ उपलब्ध करावाईं। इस प्रशिक्षण कार्यक्रम से जुड़े विभिन्न पहलुओं और गतिविधियों को सरल बनाने में सहायता करने के लिए प्रशासनिक, वित्तीय, तकनीकी और सहायक स्टाफ के कर्मचारियों के लिए हम कृतज्ञता प्रकट करते हैं। अंत में, हम उन सभी का धन्यवाद करते हैं जिन्होंने प्रत्यक्ष या परोक्ष रूप में इस संदर्भ पुस्तिका को तैयार करने में सहायता प्रदान की।

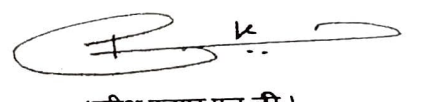
नई दिल्ली
दिसम्बर 20, 2019



(मूनमय राय)
पाठ्यक्रम समन्वयक



(शिवस्वामी जी.पी.)
पाठ्यक्रम सहसमन्वयक



(हरीश कुमार एच.वी.)
पाठ्यक्रम सहसमन्वयक

PREFACE

The institute occupies a place of pride in providing advisory and consultancy services to support and strengthen National Agricultural Research System as well as National Agricultural Statistics System by making several significant contributions which have a direct impact on the national policies. With the advances in information technology, the institute has all along been adapting itself to the current needs and methodological challenges and quality enrichment of agricultural research. Statistical modeling and Forecasting in the domain of agriculture is one of the important subjects of research at the Institute. Considering the importance of the problem, scientists in the institute are engaged in studying various modeling approaches for their forecasting applications in different sub domains of agriculture.

As a capacity strengthening initiative in ICAR institutes and SAUs and in order to make our research need based and globally competitive, it is important that the scientists engaged in various research activities are exposed to the latest developments taking place in the important area of statistical modeling in the context of forecasting in agriculture. Forecasting and Agricultural Systems Modeling division of ICAR-IASRI is organizing Centre of Advanced Faculty Training (CAFT) programme "Statistical and Machine Learning Techniques for Modeling and Forecasting Agricultural Data" from 20th December, 2019 to 09th January, 2020 at ICAR-IASRI, New Delhi under the aegis of Education Division, ICAR, New Delhi. The aim of the training programme is to provide exposure to Faculty members/ Scientists at various State Agricultural Universities/ ICAR institutes on current Statistical modeling methodologies along with machine learning based models for modeling and forecasting of agricultural data through use of various software packages (SAS, R, PYTHON and STATA) with particular emphasis on applications in agriculture. This would help them in upgrading their capabilities in research, teaching and training.

The course is structured to include both statistical and machine learning techniques for modelling and forecasting of agricultural data. The various topics covered under the course include, but not limited to: Logistic Regression, Quantile Regression, Nonlinear Statistical Models, Crop Forecasting Techniques, ARIMA and Hierarchical time-series modelling, STARMA, VARIMA Time-Series Modelling, Fuzzy Time-Series Modelling, Econometric Modeling, Count data Modeling, Nonlinear Time-Series Modelling, Artificial Neural Network, Recurrent Neural Network, Genetic algorithms, Support Vector Machine, Hybrid Time Series Modeling, CART, Stochastic volatility models, Bayesian Time Series Modeling, Remote Sensing and GIS, Applications of Technology Forecasting models in agriculture etc. along with conventional topics.

We take this opportunity to thank the faculty of the institute and the guest faculty who devoted their valuable time in making this course meaningful and successful and whose efforts helped in bringing out this manual on time. Necessary funds provided by Education Division, ICAR, New Delhi for conducting this training are duly acknowledged. We are also thankful to the various ICAR Institutes and State Agricultural Universities for deputing their scientists/ professors to this course. We are indebted to Dr. Tauqueer Ahmad, Director (Acting), ICAR-IASRI for entrusting us with the responsibility of organizing this course. We are also thankful to Dr. K N Singh, Head, Division of Forecasting and Agricultural Systems Modeling for his valuable guidance and making all necessary facilities available for smooth conduct of the course. We also place on record our thankfulness to the F&ASM division staff, administrative, financial, technical and auxiliary staff for their wholehearted support in facilitating various items and activities for this training. Finally, we are thankful to one and all, especially who helped us in preparing this manual.

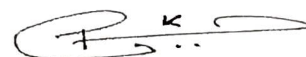
New Delhi
Dec 20, 2019



(Mrinmoy Ray)
Course Coordinator



(Shivaswamy G P)
Course Co-Coordinator



(HarishKumar H V)
Course Co-Coordinator

CONTENTS

Chapter	Title	Page No.
1	SAS for Statistical Procedures Rajender Parsad	1-52
2	An Introduction to R Software B. N. Mandal	53-70
3	An Introduction to STATA software Anuja A R, Shivaswamy G P, K N Singh, Rajesh T and HarishKumar H V	71-86
4	Basic Statistical Methods Pradip Basak	87-100
5	Classification and Regression Trees (CART) Ramasubramanian V.	101-108
6	Self-Organizing Maps (SOM) Ramasubramanian V.	109-114
7	Cluster Analysis Md. Wasi Alam	115-142
8	Logit, Probit and Tobit Models Shivaswamy G P, K N Singh, Anuja A R, Rajesh T and HarishKumar H V	143-152
9	ARIMA Model for Time-Series Forecasting Kanchan Sinha, Mrinmoy Ray, Achal Lama and K.N. Singh	153-158
10	ARIMA-Intervention Model Mrinmoy Ray, Ramasubramanian V., K N Singh and Kanchan Sinha	159-174
11	An Overview of Cointegration Analysis Kanchan Sinha, K.N. Singh, Mrinmoy Ray and Achal Lama	175-180
12	Hierarchical Time Series Modeling Soumen Pal	181-194
13	Introduction to Multivariate Time Series Model Achal Lama, K N Singh, R S Shekhawat and Bishal Gurung	195-202
14	Application of Bayesian methodology for Time Series Analysis Achal Lama, K N Singh, R S Shekhawat and Bishal Gurung	203-212
15	Wavelet Frequency Domain Approach for Time-Series Modeling Ranjit Kumar Paul	213-232
16	Least Absolute Shrinkage and Selection Operator (LASSO) Dwijesh Chandra Mishra and Sayanti Guha Majumdar	233-240
17	Linear and Integer Programming H.V. HarishKumar, Rajesh T, Shivaswamy G P, Anuja A R	241-256
18	Autoregressive and Distributed-Lag Models Rajesh T, H.V. HarishKumar, Anuja A.R, and Shivaswamy G.P.	257-265

SAS for Statistical Procedures

Rajender Parsad
ICAR-IASRI, New Delhi-110 012
Rajender.Parsad@icar.gov.in

1. Introduction

SAS (Statistical Analysis System) software is comprehensive software which deals with many problems related to Statistical analysis, Spreadsheet, Data Creation, Graphics, etc. It is a layered, multivendor architecture. Regardless of the difference in hardware, operating systems, etc., the SAS applications look the same and produce the same results. The three components of the SAS System are Host, Portable Applications and Data. Host provides all the required interfaces between the SAS system and the operating environment. Functionalities and applications reside in Portable component and the user supplies the Data. We, in this course will be dealing with the software related to perform statistical analysis of data.

Windows of SAS

1. Program Editor : All the instructions are given here.
2. Log : Displays SAS statements submitted for execution and messages
3. Output : Gives the output generated

Rules for SAS Statements

1. SAS program communicates with computer by the SAS statements.
2. Each statement of SAS program must end with semicolon (;).
3. Each program must end with run statement.
4. Statements can be started from any column.
5. One can use upper case letters, lower case letters or the combination of the two.

Basic Sections of SAS Program

1. DATA section
2. CARDS section
3. PROCEDURE section

Data Section

We shall discuss some facts regarding data before we give the syntax for this section.

Data value: A single unit of information, such as name of the specie to which the tree belongs, height of one tree, etc.

Variable: A set of values that describe a specific data characteristic e.g. diameters of all trees in a group. The variable can have a name upto a maximum of 8 characters and must begin with a letter or underscore. Variables are of two types:

Character Variable: It is a combination of letters of alphabet, numbers and special characters or symbols.

Numeric Variable: It consists of numbers with or without decimal points and with + or -ve signs.

Observation: A set of data values for the same item i.e. all measurement on a tree.

Data section starts with Data statements as

DATA NAME (it has to be supplied by the user);

Input Statements

Input statements are part of data section. This statement provides the **SAS** system the name of the variables with the format, if it is formatted.

List Directed Input

- Data are read in the order of variables given in input statement.
- Data values are separated by one or more spaces.
- Missing values are represented by period (.).
- Character values are followed by \$ (dollar sign).

Example

```
Data A;  
INPUT ID SEX $ AGE HEIGHT WEIGHT;  
CARDS;  
1 M 23 68 155  
2 F . 61 102  
3. M 55 70 202  
;
```

Column Input

Starting column for the variable can be indicated in the input statements for example:

```
INPUT ID 1-3 SEX $ 4 HEIGHT 5-6 WEIGHT 7-11;  
CARDS;  
001M68155.5  
2F61 99  
3M53 33.5  
;
```

Alternatively, starting column of the variable can be indicated along with its length as

```
INPUT @ 1 ID 3.  
@ 4 SEX $ 1.  
@ 9 AGE 2.  
@ 11 HEIGHT 2.  
@ 16 V_DATE MMDDYY 6.  
;
```

Reading More than One Line Per Observation for One Record of Input Variables

```

INPUT # 1 ID 1-3 AGE 5-6 HEIGHT 10-11
# 2 SBP 5-7 DBP 8-10;
CARDS;
001 56 72
    140 80
;

```

Reading the Variable More than Once

Suppose id variable is read from six columns in which state code is given in last two columns of id variable for example:

```

INPUT @ 1 ID 6. @ 5 STATE 2.;
      OR
INPUT ID 1-6 STATE 5-6;

```

Formatted Lists

```

DATA B;
INPUT ID @1(X1-X2)(1.)
      @4(Y1-Y2)(3.);
CARDS;
11 563789
22 567987
;
PROC PRINT;
RUN;

```

Output

Obs.	ID	x1	x2	y1	y2
1	11	1	1	563	789
2	22	2	2	567	987

```

DATA C;
INPUT X Y Z @;
CARDS;
1 1 1 2 2 2 5 5 5 6 6 6
1 2 3 4 5 6 3 3 3 4 4 4
;
PROC PRINT;
RUN;

```

Output

Obs.	X	Y	Z
1	1	1	1
2	1	2	3

```

DATA D;
INPUT X Y Z @@;

```

```

CARDS;
1 1 1 2 2 2 5 5 5 6 6 6
1 2 3 4 5 6 3 3 3 4 4 4
;
PROC PRINT;
RUN;

```

Output:

Obs.	X	Y	Z
1	1	1	1
2	2	2	2
3	5	5	5
4	6	6	6
5	1	2	3
6	4	5	6
7	3	3	3
8	4	4	4

DATA FILES

SAS System Can Read and Write

- A. Simple ASCII files are read with input and infile statements
- B. Output Data files

Creation of SAS Data Set

```

DATA EX1;
INPUT GROUP $ X Y Z;
CARDS;
T1 12 17 19
T2 23 56 45
T3 19 28 12
T4 22 23 36
T5 34 23 56
;

```

Creation of SAS File From An External (ASCII) File

```

DATA EX2;
INFILE 'B:MYDATA';
INPUT GROUP $ X Y Z;
OR
DATA EX2A;
FILENAME ABC 'B:MYDATA';
INFILE ABC;
INPUT GROUP $ X Y Z;
;

```

Creation of A SAS Data Set and An Output ASCII File Using an External File

```

DATA EX3;
FILENAME IN 'C:MYDATA';

```



```

FILENAME OUT 'A:NEWDATA';
INFILE IN;
FILE OUT;
INPUT GROUP $ X Y Z;
TOTAL =SUM (X+Y+Z);
PUT GROUP $ 1-10 @12 (X Y Z TOTAL)(5.);
RUN;

```

This above program reads raw data file from 'C: MYDATA', and creates a new variable TOTAL and writes output in the file 'A: NEWDATA'.

Creation of SAS File from an External (*.csv) File

```

data EX4;
infile'C:\Users\Admn\Desktop\sscvars.csv' dlm=' ';
/*give the exact path of the file, file should not have column headings*/
input sn loc $ year season $ crop $ rep trt gyield syield return kcal;
/*give the variables in ordered list in the file*/
/*if we have the first row as names of the columns then we can write in the above statement
firstobs=2 so that data is read from row 2 onwards*/
biomass=gyield+syield; /*generates a new variable*/
proc print data=EX4;
run;

```

Note: To create a SAS File from a *.txt file, only change csv to txt and define delimiter as per file created.

Creation of SAS File from an External (*.xls) File

Note: it is always better to copy the name of the variables as comment line before Proc Import.

```

/* name of the variables in Excel File provided the first row contains variable name*/
proc import datafile = 'C:\Users\Desktop\DATA_EXERCISE\descriptive_stats.xls'
/*give the exact path of the file*/
out = descriptive_stats replace; /*give output file name*/
proc print;
run;

```

If we want to make some transformations, then we may use the following statements:

```

data a1;
set descriptive_stats;
x = fs45+fw;
run;

```

Here **proc import** allows the SAS user to import data from an EXCEL spreadsheet into SAS. The **datafile** statement provides the reference location of the file. The **out** statement is used to name the SAS data set that has been created by the import procedure. **Print procedure** has been utilized to view the contents of the SAS data set **descriptive_stats**. When we run above codes we obtain the output which will same as shown above because we are using the same data.

Creating a Permanent SAS Data Set

```
LIBNAME XYZ 'C:\SASDATA';  
DATA XYZ.EXAMPLE;  
INPUT GROUP $ X Y Z;  
CARDS;  
.....  
.....  
.....  
RUN;
```

This program reads data following the cards statement and creates a permanent SAS data set in a subdirectory named \SASDATA on the C: drive.

Using Permanent SAS File

```
LIBNAME XYZ 'C:\SASDATA';  
PROC MEANS DATA=XYZ.EXAMPLE;  
RUN;
```

TITLES

One can enter upto 10 titles at the top of output using TITLE statement in your procedure.

```
PROC PRINT;  
TITLE 'HEIGHT-DIA STUDY';  
TITLE3 '1999 STATISTICS';  
RUN;
```

Comment cards can be added to the SAS program using
/* COMMENTS */;

FOOTNOTES

One can enter upto 10 footnotes at the bottom of your output.

```
PROC PRINT DATA=DIAHT;  
FOOTNOTE '1999';  
FOOTNOTE5 'STUDY RESULTS';  
RUN;
```

For obtaining output as RTF file, use the following statements

```
Ods rtf file='xyz.rtf' style =journal;  
Ods rtf close;
```

For obtaining output as PDF/HTML file, replace rtf with pdf or html in the above statements.

If we want to get the output in continuous format, then we may use

```
Ods rtf file='xyz.rtf' style =journal bodytitle startpage=no;
```

LABELLING THE VARIABLES

```
Data dose;  
title 'yield with factors N P K';  
input N P K Yield;
```

```

Label N = "Nitrogen";
Label P = " Phosphorus";
Label K = " Potassium";
cards;
...
...
...
;
Proc print;
run;

```

We can define the linesize in the output using statement OPTIONS. For example, if we wish that the output should have the linesize (number of columns in a line) is 72 use Options linesize =72; in the beginning.

2. Statistical Procedure

SAS/STAT has many capabilities using different procedures with many options. There are a total of 73 PROCs in SAS/STAT. SAS/STAT is capable of performing a wide range of statistical analysis that includes:

1. Elementary / Basic Statistics
 2. Graphs/Plots
 3. Regression and Correlation Analysis
 4. Analysis of Variance
 5. Experimental Data Analysis
 6. Multivariate Analysis
 7. Principal Component Analysis
 8. Discriminant Analysis
 9. Cluster Analysis
 10. Survey Data Analysis
 11. Mixed model analysis
 12. Variance Components Estimation
 13. Probit Analysis
- and many more...

A brief on SAS/STAT Procedures is available at

<http://support.sas.com/rnd/app/da/stat/procedures/Procedures.html>

Example 2.1: To Calculate the Means and Standard Deviation:

```

DATA TESTMEAN;
INPUT GROUP $ X Y Z;
CARDS;
CONTROL 12 17 19
TREAT1 23 25 29
TREAT2 19 18 16
TREAT3 22 24 29
CONTROL 13 16 17
TREAT1 20 24 28
TREAT2 16 19 15

```



```
TREAT3 24 26 30
CONTROL 14 19 21
TREAT1 23 25 29
TREAT2 18 19 17
TREAT3 23 25 30
;
PROC MEANS;
VAR X Y Z;
RUN;
```

The default output displays mean, standard deviation, minimum value, maximum value of the desired variable. We can choose the required statistics from the options of PROC MEANS. For example, if we require mean, standard deviation, median, coefficient of variation, coefficient of skewness, coefficient of kurtosis, etc., then we can write

```
PROC MEANS mean std median cv skewness kurtosis;
VAR X Y Z;
RUN;
```

The default output is 6 decimal places, desired number of decimal places can be defined by using option maxdec=.... For example, for an output with three decimal places, we may write

```
PROC MEANS mean std median cv skewness kurtosis maxdec=3;
VAR X Y Z;
RUN;
```

For obtaining means group wise use, first sort the data by groups using

```
Proc sort;
By group;
Run;
And then make use of the following
```

```
PROC MEANS;
VAR X Y Z;
by group;
RUN;
```

Or alternatively, we may use

```
PROC MEANS;
CLASS GROUP;
VAR X Y Z;
RUN;
```

For obtaining descriptive statistics for a given data one can use PROC SUMMARY. In the above example, if one wants to obtain mean standard deviation, coefficient of variation, coefficient of skewness and kurtosis, then one may utilize the following:

```
PROC SUMMARY PRINT MEAN STD CV SKEWNESS KURTOSIS;
CLASS GROUP;
VAR X Y Z;
RUN;
```

Most of the Statistical Procedures require that the data should be normally distributed. For testing the normality of data, PROC UNIVARIATE may be utilized.

```
PROC UNIVARIATE NORMAL;  
VAR X Y Z;  
RUN;
```

If different plots are required then, one may use:

```
PROC UNIVARIATE DATA=TEST NORMAL PLOT;  
/*plot option displays stem-leaf, boxplot & Normal prob plot*/  
VAR X Y Z;  
/*creates side by side BOX-PLOT group-wise. To use this option first sort the file on by  
variable*/  
BY GROUP;  
HISTOGRAM/KERNEL NORMAL; /*displays kernel density along with normal curve*/  
PROBPLOT; /*plots probability plot*/  
QQPLOT X/NORMAL SQUARE; /*plot quantile-quantile QQ-plot*/  
CDFPLOT X/NORMAL; /*plots CDF plot*/  
/*plots pp plot which compares the empirical cumulative distribution function (ecdf) of a  
variable with a specified theoretical cumulative distribution function. The beta, exponential,  
gamma, lognormal, normal, and Weibull distributions are available in both statements.*/  
PPPLOT X/NORMAL;  
RUN;
```

Example 2.2: To Create Frequency Tables

```
DATA TESTFREQ;  
INPUT AGE $ ECG CHD $ CAT $ WT; CARDS;  
<55 0 YES YES 1  
<55 0 YES YES 17  
<55 0 NO YES 7  
<55 1 YES NO 257  
<55 1 YES YES 3  
<55 1 YES NO 7  
<55 1 NO YES 1  
55+ 0 YES YES 9  
55+ 0 YES NO 15  
55+ 0 NO YES 30  
55+ 1 NO NO 107  
55+ 1 YES YES 14  
55+ 1 YES NO 5  
55+ 1 NO YES 44  
55+ 1 NO NO 27  
;  
PROC FREQ DATA=TESTFREQ;  
TABLES AGE*ECG/MISSING CHISQ;  
TABLES AGE*CAT/LIST;  
RUN;
```

SCATTER PLOT

```
PROC PLOT DATA = DIAHT;  
PLOT HT*DIA = '*';  
/*HT=VERTICAL AXIS DIA = HORIZONTAL AXIS.*/  
RUN;
```

CHART

```
PROC CHART DATA = DIAHT;  
VBAR HT;  
RUN;
```

```
PROC CHART DATA = DIAHT;  
HBAR DIA;  
RUN;
```

```
PROC CHART DATA = DIAHT;  
PIE HT;  
RUN;
```

Example 2.3: To Create A Permanent SAS DATASET and use that for Regression

```
LIBNAME FILEX 'C:\SAS\RPLIB';
```

```
DATA FILEX.RP;
```

```
INPUT X1-X5;
```

```
CARDS;
```

```
1 0 0 0 5.2  
.75 .25 0 0 7.2  
.75 0 .25 0 5.8  
.5 .25 .25 0 6.3  
.75 0 0 .25 5.5  
.5 0 .25 .25 5.7  
.5 .25 0 .25 5.8  
.25 .25 .25 .25 5.7  
;  
RUN;
```

```
LIBNAME FILEX 'C:\SAS\RPLIB';
```

```
PROC REG DATA=FILEX.RP;
```

```
MODEL X5 = X1 X2/P;
```

```
MODEL X5 = X1 X2 X3 X4 / SELECTION = STEPWISE;
```

```
TEST: TEST X1-X2=0;
```

```
RUN;
```

Various other commonly used PROC Statements are PROC ANOVA, PROC GLM; PROC CORR; PROC NESTED; PROC MIXED; PROC RSREG; PROC IML; PROC PRINCOMP; PROC VARCOMP; PROC FACTOR; PROC CANCELL; PROC DISCRIM, etc. Some of these are described in the sequel.

PROC TTEST is the procedure that is used for comparing the mean of a given sample. This PROC is also used for compares the means of two independent samples. The paired observations t test compares the mean of the differences in the observations to a given number. The underlying assumption of the t test in all three cases is that the observations are random samples drawn from normally distributed populations. This assumption can be checked using the UNIVARIATE procedure; if the normality assumptions for the t test are not satisfied, one should analyze the data using the NPARIWAY procedure. PROC TTEST computes the group comparison t statistic based on the assumption that the variances of the two groups are equal. It also computes an approximate t based on the assumption that the variances are unequal (the Behrens-Fisher problem). The following statements are available in PROC TTEST.

```
PROC TTEST <options>;
CLASS variable;
PAIRED variables;
BY variables;
VAR variables;
FREQ Variables;
WEIGHT variable;
```

No statement can be used more than once. There is no restriction on the order of the statements after the PROC statement. The following options can appear in the PROC TTEST statement.

ALPHA= p : option specifies that confidence intervals are to be $100(1-p)\%$ confidence intervals, where $0 < p < 1$. By default, PROC TTEST uses ALPHA=0.05. If p is 0 or less, or 1 or more, an error message is printed.

COCHRAN: option requests the Cochran and Cox approximation of the probability level of the approximate t statistic for the unequal variances situation.

H0= m : option requests tests against m instead of 0 in all three situations (one-sample, two-sample, and paired observation t tests). By default, PROC TTEST uses H0=0.

A CLASS statement giving the name of the classification (or grouping) variable must accompany the PROC TTEST statement in the two independent sample cases. It should be omitted for the one sample or paired comparison situations. The class variable must have two, and only two, levels. PROC TTEST divides the observations into the two groups for the t test using the levels of this variable. One can use either a numeric or a character variable in the CLASS statement.

In the statement PAIRED *PairLists*, the *PairLists* in the PAIRED statement identifies the variables to be compared in paired comparisons. You can use one or more *PairLists*. Variables or lists of variables are separated by an asterisk (*) or a colon (:). Examples of the use of the asterisk and the colon are shown in the following table.

The PAIRED Statements	Comparisons made
PAIRED A*B;	A-B
PAIRED A*B C*D;	A-B and C-D
PAIRED (A B)*(C B);	A-C, A-B and B-C
PAIRED (A1-A2)*(B1-B2);	A1-B1, A1-B2, A2-B1 and A2-B2

PAIRED (A1-A2):(B1-B2);	A1-B1 and A2-B2
-------------------------	-----------------

PROC ANOVA performs analysis of variance for balanced data only from a wide variety of experimental designs whereas PROC GLM can analyze both balanced and unbalanced data. As ANOVA takes into account the special features of a balanced design, it is faster and uses less storage than PROC GLM for balanced data. The basic syntax of the ANOVA procedure is as given:

```
PROC ANOVA < Options>;
  CLASS variables;
  MODEL dependents = independent variables (or effects)/options;
  MEANS effects/options;
  ABSORB variables;
  FREQ variables;
  TEST H = effects E = effect;
  MANOVA H = effects E = effect;
        M = equations/options;
  REPEATED factor - name levels / options;
  BY variables;
```

The PROC ANOVA, CLASS and MODEL statements are must. The other statements are optional. The CLASS statement defines the variables for classification (numeric or character variables - maximum characters =16).

The MODEL statement names the dependent variables and independent variables or effects. If no effects are specified in the MODEL statement, ANOVA fits only the intercept. Included in the ANOVA output are F-tests of all effects in the MODEL statement. All of these F-tests use residual mean squares as the error term. The MEANS statement produces tables of the means corresponding to the list of effects. Among the options available in the MEANS statement are several multiple comparison procedures viz. Least Significant Difference (LSD), Duncan's New multiple - range test (DUNCAN), Waller - Duncan (WALLER) test, Tukey's Honest Significant Difference (TUKEY). The LSD, DUNCAN and TUKEY options takes level of significance ALPHA = 5% unless ALPHA = options is specified. Only ALPHA = 1%, 5% and 10% are allowed with the Duncan's test. 95% Confidence intervals about means can be obtained using CLM option under MEANS statement.

The TEST statement tests for the effects where the residual mean square is not the appropriate term such as main - plot effects in split - plot experiment. There can be multiple MEANS and TEST statements (as well as in PROC GLM), but only one MODEL statement preceded by RUN statement. The ABSORB statement implements the technique of absorption, which saves time and reduces storage requirements for certain type of models. FREQ statement is used when each observation in a data set represents 'n' observations, where n is the value of FREQ variable. The MANOVA statement is used for implementing multivariate analysis of variance. The REPEATED statement is useful for analyzing repeated measurement designs and the BY statement specifies that separate analysis are performed on observations in groups defined by the BY variables.

PROC GLM for analysis of variance is similar to using PROC ANOVA. The statements listed for PROC ANOVA are also used for PROC GLM. In addition; the following more statements can be used with PROC GLM:

```
CONTRAST 'label' effect name< ... effect coefficients > </options>;
ESTIMATE 'label' effect name< ... effect coefficients > </options>;
ID variables;
LSMEANS effects < / options >;
OUTPUT < OUT = SAS-data-set>keyword=names< ... keyword = names>;
RANDOM effects < / options >;
WEIGHT variables
```

Multiple comparisons as used in the options under MEANS statement are useful when there are no particular comparisons of special interest. But there do occur situations where preplanned comparisons are required to be made. Using the CONTRAST, LSMEANS statement, we can test specific hypothesis regarding pre - planned comparisons. The basic form of the CONTRAST statement is as described above, where label is a character string used for labeling output, effect name is class variable (which is independent) and effect - coefficients is a list of numbers that specifies the linear combination parameters in the null hypothesis. The contrast is a linear function such that the elements of the coefficient vector sum to 0 for each effect. While using the CONTRAST statements, following points should be kept in mind.

How many levels (classes) are there for that effect. If there are more levels of that effect in the data than the number of coefficients specified in the CONTRAST statement, the PROC GLM adds trailing zeros. Suppose there are 5 treatments in a completely randomized design denoted as T_1, T_2, T_3, T_4, T_5 and null hypothesis to be tested is

$$H_0: T_2+T_3 = 2T_1 \text{ or } -2T_1+T_2+T_3 = 0$$

Suppose in the data treatments are classified using TRT as class variable, then effect name is TRT CONTRAST 'TIVS 2&3' TRT -2 1 1 0 0; Suppose last 2 zeros are not given, the trailing zeros can be added automatically. The use of this statement gives a sum of squares with 1 degree of freedom (d.f.) and F-value against error as residual mean squares until specified. The name or label of the contrast must be 20 characters or less.

The available CONTRAST statement options are

E: prints the entire vector of coefficients in the linear function, i.e., contrast.

E = effect: specifies an effect in the model that can be used as an error term

ETYPE = n : specifies the types (1, 2, 3 or 4) of the E effect.

Multiple degrees of freedom contrasts can be specified by repeating the effect name and coefficients as needed separated by commas. Thus the statement for the above example

```
CONTRAST 'All' TRT -2 1 1 0 0, TRT 0 1 -1 0 0;
```

This statement produces two d.f. sum of squares due to both the contrasts. This feature can be used to obtain partial sums of squares for effects through the reduction principle, using

sums of squares from multiple degrees of freedom contrasts that include and exclude the desired contrasts. Although only $t-1$ linearly independent contrasts exists for t classes, any number of contrasts can be specified.

The ESTIMATE statement can be used to estimate linear functions of parameters that may or may not be obtained by using CONTRAST or LSMEANS statement. For the specification of the statement only word CONTRAST is to be replaced by ESTIMATE in CONTRAST statement.

Fractions in effects coefficients can be avoided by using DIVISOR = Common denominator as an option. This statement provides the value of an estimate, a standard error and a t-statistic for testing whether the estimate is significantly different from zero.

The LSMEANS statement produces the least square estimates of CLASS variable means i.e. adjusted means. For one-way structure, there are simply the ordinary means. The least squares means for the five treatments for all dependent variables in the model statement can be obtained using the statement.

LSMEANS TRT / options;

Various options available with this statement are:

STDERR: gives the standard errors of each of the estimated least square mean and the t-statistic for a test of hypothesis that the mean is zero.

PDIF: Prints the p - values for the tests of equality of all pairs of CLASS means.

SINGULAR: tunes the estimability checking. The options E, E=, E-TYPE = are similar as discussed under CONTRAST statement.

Adjust=T: gives the probabilities of significance of pairwise comparisons based on T-test.

Adjust=Tukey: gives the probabilities of significance of pairwise comparisons based on Tukey's test

Lines: gives the letters on treatments showing significant and non-significant groups

When the **predicted** values are requested as a MODEL statement option, values of variable specified in the ID statement are printed for identification besides each observed, predicted and residual value. The OUTPUT statement produces an output data set that contains the original data set values alongwith the predicted and residual values.

Besides other options in PROC GLM under MODEL statement we can give the option: 1. solution 2. $xpx (=X'X)$ 3 . I (g-inverse)

PROC GLM recognizes different theoretical approaches to ANOVA by providing four types of sums of squares and associated statistics. The four types of sums of squares in PROC GLM are called Type I, Type II, Type III and Type IV.

The Type I sums of squares are the classical sequential sums of squares obtained by adding the terms to the model in some logical sequence. The sum of squares for each class of effects

is adjusted for only those effects that precede it in the model. Thus the sums of squares and their expectations are dependent on the order in which the model is specified.

The Type II, III and IV are ‘partial sums of squares’ in the sense that each is adjusted for all other classes of the effects in the model, but each is adjusted according to different rules. One general rule applies to all three types: the estimable functions that generate the sums of squares for one class of squares will not involve any other classes of effects except those that “contain” the class of effects in question.

For example, the estimable functions that generate SS (AB) in a three- factor factorial will have zero coefficients on main effects and the (A × C) and (B × C) interaction effects. They will contain non-zero coefficient on the (A × B × C) interaction effects, because A × B × C interaction “contains” A × B interaction.

Type II, III and IV sums of squares differ from each other in how the coefficients are determined for the classes of effects that do not have zero coefficients - those that contain the class of effects in question. The estimable functions for the Type II sum of squares impose no restriction on the values of the non-zero coefficients on the remaining effects; they are allowed to take whatever values result from the computations adjusting for effects that are required to have zero coefficients. Thus, the coefficients on the higher-order interaction effects and higher level nesting effects are functions of the number of observations in the data. In general, the Type II sums of squares do not possess of equitable distribution property and orthogonality characteristic of balanced data.

The Type III and IV sums of squares differ from the Type II sums of squares in the sense that the coefficients on the higher order interaction or nested effects that contain the effects in question are also adjusted so as to satisfy either the orthogonality condition (Type III) or the equitable distribution property (Type IV).

The coefficients on these effects are no longer functions of the n_{ij} and consequently, are the same for all designs with the same general form of estimable functions. If there are no empty cells (no $n_{ij} = 0$) both conditions can be satisfied at the same time and Type III and Type IV sums of squares are equal. The hypothesis being tested is the same as when the data is balanced.

When there are empty cells, the hypotheses being tested by the Type III and Type IV sums of squares may differ. The Type III criterion of orthogonality reproduces the same hypotheses one obtains if effects are assumed to add to zero. When there are empty cells this is modified to “the effects that are present are assumed to be zero”. The Type IV hypotheses utilize balanced subsets of non-empty cells and may not be unique. For a 2x3 factorial for illustration purpose adding the terms to the model in the order A, B, AB various types sums of squares can be explained as follows:

Effect	Type I	Type II	Type III	Type IV
General Mean	$R(\mu)$	$R(\mu)$		
A	$R(A/\mu)$	$R(A/\mu, B)$	$R(A/\mu, B, AB)$	
B	$R(B/\mu, A)$	$R(B/\mu, A)$	$R(B/\mu, A, AB)$	

A*B	R(A*B/ μ, A, B)	R(A*B/ μ, A, B)	R(AB/ μ, A, B)	
-----	----------------------	----------------------	---------------------	--

R (A/ μ) is sum of squares adjusted for μ , and so on.

Thus in brief the four sets of sums of squares Type I, II, III & IV can be thought of respectively as sequential, each - after-all others, Σ -restrictions and hypotheses.

There is a relationship between the four types of sums of squares and four types of data structures (balanced and orthogonal, unbalanced and orthogonal, unbalanced and non-orthogonal (all cells filled), unbalanced and non-orthogonal (empty cells)). For illustration, let n_{IJ} denote the number of observations in level I of factor A and level j of factor B. Following table explains the relationship in data structures and Types of sums of squares in a two-way classified data.

	Data Structure Type			
	1	2	3	4
Effect	Equal n_{IJ}	Proportionate n_{IJ}	Disproportionate non-zero n_{IJ}	Empty Cell
A	I=II=III=IV	I=II,III=IV	III=IV	
B	I=II=III=IV	I=II,III=IV	I=II,III=IV	I=II
A*B	I=II=III=IV	I=II=III=IV	I=II=III=IV	I=II=III=IV

In general,
 I=II=III=IV (balanced data); II=III=IV (no interaction models)
 I=II, III=IV (orthogonal data); III=IV (all cells filled data).

Proper Error terms: In general F-tests of hypotheses in ANOVA use the residual mean squares in other terms are to be used as error terms. For such situations PROC GLM provides the TEST statement which is identical to the test statement available in PROC ANOVA. PROC GLM also allows specification of appropriate error terms in MEANS LSMEANS and CONTRAST statements. To illustrate it let us use split plot experiment involving the yield of different irrigation (IRRIG) treatments applied to main plots and cultivars (CULT) applied to subplots. The data so obtained can be analysed using the following statements.

```
data splitplot;
input REP IRRIG CULT YIELD;
cards;
...
...
...
;
PROC print; run;
PROC GLM;
class rep, irrig cult;
model yield = rep irrig rep*irrig cult irrig* cult;
test h = irrig e = rep * irrig;
```

```
contrast 'IRRIG1 Vs IRRIG2' irrig 1 -1 / e = rep* irrig;
run;
```

As we know here that the irrigation effects are tested using error (A) which is sum of squares due to rep* irrig, as taken in test statement and contrast statement respectively.

In Test statement	H	=	numerator for - source of variation and
	E	=	denominator source of variation

It may be noted here that the PROC GLM can be used to perform analysis of covariance as well. For analysis of covariance, the covariate should be defined in the model without specifying under CLASS statement.

PROC RSREG fits the parameters of a complete quadratic response surface and analyses the fitted surface to determine the factor levels of optimum response and performs a ridge analysis to search for the region of optimum response.

```
PROC RSREG < options >;
MODEL responses = independents / <options >;
RIDGE < options >;
WEIGHT variable;
ID variable;
By variable;
run;
```

The PROC RSREG and model statements are required. The BY, ID, MODEL, RIDGE, and WEIGHT statements are described after the PROC RSREG statement below and can appear in any order.

The PROC RSREG statement invokes the procedure and following options are allowed with the PROC RSREG:

DATA = SAS - data-set	: specifies the data to be analysed.
NOPRINT	: suppresses all printed results when only the output data set is required.
OUT	: SAS-data-set: creates an output data set.

The model statement without any options transforms the independent variables to the coded data. By default, PROC RSREG computes the linear transformation to perform the coding of variables by subtracting average of highest and lowest values of the independent variable from the original value and dividing by half of their differences. Canonical and ridge analyses are performed to the model fit to the coded data. The important options available with the model statement are:

NOCODE	: Analyses the original data.
ACTUAL	: specifies the actual values from the input data set.
COVAR = n	: declares that the first n variables on the independent side of the model are simple linear regression (covariates) rather than factors in the quadratic response surface.
LACKFIT	: Performs lack of fit test. For this the repeated observations must appear together.
NOANOVA	: suppresses the printing of the analysis of variance and parameter estimates from the model fit.

NOOPTIMAL (NOOPT): suppresses the printing of canonical analysis for quadratic response surface.

NOPRINT : suppresses both ANOVA and the canonical analysis.

PREDICT : specifies the values predicted by the model.

RESIDUAL : specifies the residuals.

A RIDGE statement computes the ridge of the optimum response. Following important options available with RIDGE statement are

MAX: computes the ridge of maximum response.

MIN: computes the ridge of the minimum response.

At least one of the two options must be specified.

NOPRINT: suppresses printing the ridge analysis only when an output data set is required.

OUTR = SAS-data-set: creates an output data set containing the computed optimum ridge.

RADIUS = coded-radii: gives the distances from the ridge starting point at which to compute the optimum.

PROC REG is the primary SAS procedure for performing the computations for a statistical analysis of data based on a linear regression model. The basic statements for performing such an analysis are

PROC REG;

MODEL list of dependent variable = list of independent variables/ model options;

RUN;

The PROC REG procedure and model statement without any option gives ANOVA, root mean square error, R-squares, Adjusted R-square, coefficient of variation etc.

The options under model statement are

P: It gives predicted values corresponding to each observation in the data set. The estimated standard errors are also given by using this option.

CLM: It yields upper and lower 95% confidence limits for the mean of subpopulation corresponding to specific values of the independent variables.

CLI: It yields a prediction interval for a single unit to be drawn at random from a subpopulation.

STB: Standardized regression coefficients.

XPX, I: Prints matrices used in regression computations.

NOINT: This option forces the regression response to pass through the origin. With this option total sum of squares is uncorrected and hence R-square statistic are much larger than those for the models with intercept.

However, if no intercept model is to be fitted with corrected total sum of squares and hence usual definition of various statistic viz R^2 , MSE etc. are to be retained then the option RESTRICT intercept = 0; may be exercised after the model statement.

For obtaining residuals and studentized residuals, the option 'R' may be exercised under model statement and Cook's D statistic.

The 'INFLUENCE' option under model statement is used for detection of outliers in the data and provides residuals, studentized residuals, diagonal elements of HAT MATRIX, COVRATIO, DFFITS, DFBETAS, etc.

For detecting multicollinearity in the data, the options 'VIF' (variance inflation factors) and 'COLLINOINT' or 'COLLIN' may be used.

Besides the options for weighted regression, output data sets, specification error, heterogeneous variances etc. are available under PROC REG.

PROC PRINCOMP can be utilized to perform the principal component analysis.

Multiple model statements are permitted in PROC REG unlike PROC ANOVA and PROC GLM. A model statement can contain several dependent variables.

The statement model $y_1, y_2, y_3, y_4 = x_1 x_2 x_3 x_4 x_5$; performs four separate regression analyses of variables y_1, y_2, y_3 and y_4 on the set of variables x_1, x_2, x_3, x_4, x_5 .

Polynomial models can be fitted by using independent variables in the model as $x_1 = x$, $x_2 = x^{**}2$, $x_3 = x^{**}3$, and so on depending upon the order of the polynomial to be fitted. From a variable, several other variables can be generated before the model statement and transformed variables can be used in model statement. LY and LX gives Logarithms of Y & X respectively to the base e and LogY, LogX gives logarithms of Y and X respectively to the base 10.

TEST statement after the model statement can be utilized to test hypotheses on individual or any linear function(s) of the parameters.

For e.g. if one wants to test the equality of coefficients of x_1 and x_2 in $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ regression model, statement

TEST 1: TEST $x_1 - x_2 = 0$;

Label: Test < equation ..., equation >;

The fitted model can be changed by using a separate model statement or by using DELETE variables; or ADD variables; statements.

The PROC REG provides two types of sums of squares obtained by SS1 or SS2 options under model statement. Type I SS are sequential sum of squares and Types II sum of squares are partial SS are same for that variable which is fitted at last in the model.

For most applications, the desired test for a single parameter is based on the Type II sum of squares, which are equivalent to the t-tests for the parameter estimates. The Type I sum of squares, however, are useful if there is a need for a specific sequencing of tests on individual coefficients as in polynomial models.

PROC ANOVA and PROC GLM are general purpose procedures that can be used for a broad range of data classification. In contrast, PROC NESTED is a specialized procedure that is useful only for nested classifications. It provides estimates of the components of variance using the analysis of variance method of estimation. The CLASS statement in

PROC NESTED has a broader purpose than it does in PROC ANOVA and PROC GLM; it encompasses the purpose of MODEL statement as well. But the data must be sorted appropriately. For example in a laboratory microbial counts are made in a study, whose objective is to assess the source of variation in number of microbes. For this study n_1 packages of the test material are purchased and n_2 samples are drawn from each package i.e. samples are nested within packages. Logarithm transformation is to be used for microbial counts. PROPER SAS statements are:

```
PROC SORT; By package sample;
PROC NESTED;
CLASS package sample;
Var logcount;
run;
```

Corresponding PROC GLM statements are

```
PROC GLM;
Class package sample;
Model Logcount= package sample (package);
```

The F-statistic in basic PROC GLM output is not necessarily correct. For this RANDOM statement with a list of all random effects in the model is used and Test option is utilized to get correct error term. However, for fixed effect models same arguments for proper error terms hold as in PROC GLM and PROC ANOVA. For the analysis of the data using linear mixed effects model, PROC MIXED of SAS should be used. The best linear unbiased predictors and solutions for random and fixed effects can be obtained by using option 's' in the Random statement.

PROCEDURES FOR SURVEY DATA ANALYSIS

PROC SURVEYMEANS procedure produces estimates of population means and totals from sample survey data. You can use PROC SURVEYMEANS to compute the following statistics:

- estimates of population means, with corresponding standard errors and t tests
- estimates of population totals, with corresponding standard deviations and t tests
- estimates of proportions for categorical variables, with standard errors and t tests
- ratio estimates of population means and proportions, and their standard errors
- confidence limits for population means, totals, and proportions
- data summary information

PROC SURVEYFREQ procedure produces one-way to n -way frequency and crosstabulation tables from sample survey data. These tables include estimates of population totals, population proportions (overall proportions, and also row and column proportions), and corresponding standard errors. Confidence limits, coefficients of variation, and design effects are also available. The procedure also provides a variety of options to customize your table display.

PROC SURVEYREG procedure fits linear models for survey data and computes regression coefficients and their variance-covariance matrix. The procedure allows you to specify classification effects using the same syntax as in the GLM procedure. The procedure also provides hypothesis tests for the model effects, for any specified estimable linear functions of the model parameters, and for custom hypothesis tests for linear combinations of the

regression parameters. The procedure also computes the confidence limits of the parameter estimates and their linear estimable functions.

PROC SURVEYLOGISTIC procedure investigates the relationship between discrete responses and a set of explanatory variables for survey data. The procedure fits linear logistic regression models for discrete response survey data by the method of maximum likelihood, incorporating the sample design into the analysis. The SURVEYLOGISTIC procedure enables you to use categorical classification variables (also known as CLASS variables) as explanatory variables in an explanatory model, using the familiar syntax for main effects and interactions employed in the GLM and LOGISTIC procedures.

The SURVEYSELECT procedure provides a variety of methods for selecting probability-based random samples. The procedure can select a simple random sample or a sample according to a complex multistage sample design that includes stratification, clustering, and unequal probabilities of selection. With probability sampling, each unit in the survey population has a known, positive probability of selection. This property of probability sampling avoids selection bias and enables you to use statistical theory to make valid inferences from the sample to the survey population.

PROC SURVEYSELECT provides methods for both equal probability sampling and sampling with probability proportional to size (PPS). In PPS sampling, a unit's selection probability is proportional to its size measure. PPS sampling is often used in cluster sampling, where you select clusters (groups of sampling units) of varying size in the first stage of selection. Available PPS methods include without replacement, with replacement, systematic, and sequential with minimum replacement. The procedure can apply these methods for stratified and replicated sample designs.

3. Exercises

Example 3.1: An experiment was conducted to study the hybrid seed production of bottle gourd (*Lagenaria siceraria (Mol) Standl*) Cv. Pusa hybrid-3 under open field conditions during Kharif-2005 at Indian Agricultural Research Institute, New Delhi. The main aim of the investigation was to compare natural pollination and hand pollination. The data were collected on 10 randomly selected plants from each of natural pollination and hand pollination on number of fruit set for the period of 45 days, fruit weight (kg), seed yield per plant (g) and seedling length (cm). The data obtained is as given below:

Group	No. of fruit	Fruit weight (kg)	Seed yield/plant (g)	Seedling length (cm)
1	7.0	1.85	147.70	16.86
1	7.0	1.86	136.86	16.77
1	6.0	1.83	149.97	16.35
1	7.0	1.89	172.33	18.26
1	7.0	1.80	144.46	17.90
1	6.0	1.88	138.30	16.95
1	7.0	1.89	150.58	18.15
1	7.0	1.79	140.99	18.86

1	6.0	1.85	140.57	18.39
1	7.0	1.84	138.33	18.58
2	6.3	2.58	224.26	18.18
2	6.7	2.74	197.50	18.07
2	7.3	2.58	230.34	19.07
2	8.0	2.62	217.05	19.00
2	8.0	2.68	233.84	18.00
2	8.0	2.56	216.52	18.49
2	7.7	2.34	211.93	17.45
2	7.7	2.67	210.37	18.97
2	7.0	2.45	199.87	19.31
2	7.3	2.44	214.30	19.36

{ Here 1 denotes natural pollination and 2 denotes the hand pollination }

1. Test whether the mean of the population of Seed yield/plant (g) is 200 or not.
2. Test whether the natural pollination and hand pollination under open field conditions are equally effective or are significantly different.
3. Test whether hand pollination is better alternative in comparison to natural pollination.

Procedure:

For performing analysis, input the data in the following format. {Here Number of fruit (45 days) is termed as nfs45, Fruit weight (kg) is termed as fw, seed yield/plant (g) is termed as syp and Seedling length (cm) is termed as sl. It may, however, be noted that one can retain the same name or can code in any other fashion}.

```
data ttest1; /*one can enter any other name for data*/
input group  nfs45  fw      syp  sl;
cards;
.....
.....
.....
;
```

*To answer the question number 1 use the following SAS statements

```
proc ttest H0=200;
```

```
var syp;
```

```
run;
```

*To answer the question number 2 use the following SAS statements;

```
proc ttest;
```

```
class group;
```

```
var nfs45 fw syp sl;
```

```
run;
```

To answer the question number 3 one has to perform the one tail t-test. The easiest way to convert a two-tailed test into a one-tailed test is take half of the p-value provided in the output of 2-tailed test output for drawing inferences. The other way is using the options sides in proc statement. Here we are interested in testing whether hand pollination is better alternative in comparison to natural pollination, therefore, we may use Sides=L as

```
proc ttest sides=L;
class group;
var nfs45 fw syp sl;
run;
```

Similarly this option can also be used in one sample test and for right tail test Sides=U is used.

Exercise 3.2: A study was undertaken to find out whether the average grain yield of paddy of farmers using laser levelling is more than the farmers using traditional land levelling methods. For this study data on grain yield in tonne/hectare was collected from 59 farmers (33 using traditional land levelling methods and 26 using new land leveller) and is given as:

Traditional	Laser		Traditional	Laser
3.67	3.6		3.79	3.95
4.04	3.7		3.17	5.3
3.49	5.3		3.58	5.8
2.75	4.4		4.08	2.8
2.63	5.4		4.25	3.0
2.46	3.4		5.21	4.78
2.50	3.5		5.63	4.07
2.88	8.2		3.42	4.88
2.45	7.5		3.88	4.37
2.46	7.6		3.29	
2.67	7.0		3.92	
2.38	7.4		2.25	
2.42	3.4		2.58	
2.54	3.6		3.25	
3.88	5.6		3.46	
3.88	5.6		3.79	
3.42	5.4			

Test whether the traditional land levelling and laser levelling give equivalent yields or are significantly different.

Procedure:

For performing analysis, input the data in the following format. {Here traditional land levelling is termed as LL, laser levelling as LL, method of levelling as MLevel and grain yield in t/ha as gyld. It may, however, be noted that one can retain the same name or can code in any other fashion}.

```

data ttestL; /*one can enter any other name for data*/
input MLevel gyld;
cards;
.....
.....
.....
;

```

*To answer the question number 1 use the following SAS statements

```

proc ttest data =ttestL;
var gyld;
run;

```

Exercise 3.3: The observations obtained from 15 experimental units before and after application of the treatment are the following:

Unit No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Before	80	73	70	60	88	84	65	37	91	98	52	78	40	79	59
After	82	71	95	69	100	71	75	60	95	99	65	83	60	86	62

1. Test whether the mean score before application of treatment is 65.
2. Test whether the application of treatments has resulted into some change in the score of the experimental units.
3. Test whether application of treatment has improved the scores.

Procedure:

```

data ttest;
input sn preapp postapp;
cards;
1 80 82
2 73 71
3 70 95
4 60 69
5 88 100
6 84 71
7 65 75
8 37 60
9 91 95
10 98 99
11 52 65
12 78 83
13 40 60
14 79 86
15 59 62
;

```

*For objective 1, use the following;
PROC TTEST H0=65;

```
VAR PREAPP;
RUN;
*For objective 2, use the following;
PROC TTEST;
PAIRED PREAPP*POSTAPP;
RUN;
```

```
*For objective 3, use the following;
PROC TTEST sides=L;
PAIRED PREAPP*POSTAPP;
RUN;
```

Exercise 3.4: In F_2 population of a breeding trial on pea, out of a total of 556 seeds, the frequency of seeds of different shape and colour are: 315 rounds and yellow, 101 wrinkled and yellow, 108 round and green, 32 wrinkled and green. Test at 5% level of significance whether the different shape and colour of seeds are in proportion of 9:3:3:1 respectively.

Procedure:

```
/*rndyel=round and yellow, rndgrn=round and green, wrnkyel=wrinkled and yellow,
wrnkgrn=wrinkled and green*/;
```

```
data peas;
input shape_color $ count;
cards;
rndyel 315
rndgrn 108
wrnkyel 101
wrnkgrn 32
;
```

```
proc freq data=peas order=data;
weight count ;
tables shape_color / chisq testp=(0.5625 0.1875 0.1875 0.0625);
exact chisq;
run;
```

Exercise 3.5: The educational standard of adoptability of new innovations among 600 farmers are given as below:

Educational standard			
Adoptability	Matric	Graduate	Illiterate
Adopted	100	60	80
Not adopted	50	20	290

Draw the inferences whether educational standard has any impact on their adoptability of innovation.

Procedure:

```
data innovation;
input edu $ adopt $ count;
```



```

cards;
Matric adopt 100
Matric Noadopt 50
grad adopt 60
grad Noadopt 20
illit adopt 80
illit Noadopt 290
;
proc freq order=data;
  weight count ;
  tables edu*adopt / chisq ;
run;

```

Exercise 3.6: An Experiment was conducted using a Randomized complete block design in 5 treatments a, b, c, d & e with three replications. The data (yield) obtained is given below:

Treatment(TRT)					
Replication(REP)	a	b	c	d	e
1	16.9	18.2	17.0	15.1	18.3
2	16.5	19.2	18.1	16.0	18.3
3	17.5	17.1	17.3	17.8	19.8

1. Perform the analysis of variance of the data.
2. Test the equality of treatment means.
3. Test $H_0: 2T_1=T_2+T_3$, where as T_1, T_2, T_3, T_4 and T_5 are treatment effects.

Procedure:

Prepare a SAS data file using

DATA Name;

INPUT REP TRT \$ yield;

Cards;

...

...

...

;

Print data using PROC PRINT. Perform analysis using PROC ANOVA, obtain means of treatments and obtain pairwise comparisons using least square differences, Duncan's New Multiple range tests and Tukey's Honest Significant difference tests. Make use of the following statements:

PROC Print;

PROC ANOVA;

Class REP TRT;

Model Yield = REP TRT;

Means TRT/lst;

Means TRT/duncan;

Means TRT/tukey;

Run;

Perform contrast analysis using PROC GLM.

Proc glm;
 Class rep trt;
 Model yld = rep trt;
 Means TRT/lsd;
 Means TRT/duncan;
 Means TRT/tukey
 Contrast '1 Vs 2&3' trt 2 -1 -1; Run;

Exercise 3.7: In order to select suitable tree species for Fuel, Fodder and Timber an experiment was conducted in a randomized complete block design with ten different trees and four replications. The plant height was recorded in cm. The details of the experiment are given below:

Plant Height (Cms): Place – Kanpur

Name of Tree	Spacing	Replications			
		1	2	3	4
A. Indica	4x4	144.44	145.11	104.00	105.44
D. Sisso	4x2	113.50	118.61	118.61	123.00
A. Procer	4x2	60.88	90.94	80.33	92.00
A. Nilotic	4x2	163.44	158.55	158.88	153.11
T. Arjuna	4x2	110.11	116.00	119.66	103.22
L. Loucoc	4x1	260.05	102.27	256.22	217.80
M. Alba	4x2	114.00	115.16	114.88	106.33
C. Siamia	4x2	91.94	58.16	76.83	79.50
E. Hybrid	4x1	156.11	177.97	148.22	183.17
A. Catech	4x2	80.2	108.05	45.18	79.55

Analyze the data and draw your conclusions.

Exercise 3.8: An experiment was conducted with 49 crop varieties (TRT) using a simple lattice design. The layout and data obtained (Yld) is as given below:

REPLICATION (REP)-I

BLOCKS(BLK)						
1	2	3	4	5	6	7
22(7)	10(12)	45(22)	37(25)	18(33)	30(33)	5(28)
24(20)	14(26)	44(21)	41(23)	19(17)	34(31)	6(74)
28(25)	8(42)	43(16)	40(11)	21(13)	35(10)	7(14)
27(68)	9(13)	47(37)	42(24)	17(10)	32(12)	2(14)
25(4)	13(10)	49(13)	36(30)	15(36)	29(22)	1(16)
26(11)	12(21)	48(21)	39(34)	20(30)	33(33)	3(11)
23(45)	11(11)	46(12)	38(15)	16(14)	31(18)	4(7)

REPLICATION (REP)-II

BLOCKS(BLK)						
1	2	3	4	5	6	7
22(29)	18(64)	20(25)	23(45)	5(19)	3(13)	14(60)

8(127)	25(31)	27(71)	16(22)	19(47)	24(23)	49(72)
43(119)	46(85)	13(51)	2(13)	47(86)	17(51)	21(10)
1(24)	11(51)	48(121)	37(85)	40(33)	10(30)	42(23)
36(58)	4(39)	41(22)	9(10)	12(48)	31(50)	35(54)
29(97)	39(67)	6(75)	30(65)	33(73)	38(30)	28(54)
15(47)	32(93)	34(44)	44(5)	26(56)	45(103)	7(85)

1. Perform the analysis of variance of the data. Also obtain Type II SS.
2. Obtain adjusted treatment means with their standard errors.
3. Test the equality of all adjusted treatment means.
4. Test whether the sum of 1 to 3 treatment means is equal to the sum of 4 to 6 treatments.
5. Estimate difference between average treatment 1 average of 2 to 4 treatment means.
6. Divide the between block sum of squares into between replication sum of squares and between blocks within replications sum of squares.
7. Assuming that the varieties are a random selection from a population, obtain the genotypic variance.
8. Analyze the data using block effects as random.

PROCEDURE

Prepare the DATA file.

DATA Name;

INPUT REP BLK TRT yield;

Cards;

....

....

....

;

Print data using PROC PRINT. Perform analysis of 1 to 5 objectives using PROC GLM. The statements are as follows:

Proc print;

Proc glm;

Class rep blk trt;

Model yld= blk trt/ss2;

Contrast 'A' trt 1 1 1 -1 -1 -1;

Estimate 'A' trt 3 -1 -1 -1/divisor=3;

Run;

The objective 6 can be achieved by another model statement.

Proc glm;

Class rep blk trt;

Model yield= rep blk (rep) trt/ss2;

run;

The objective 7 can be achieved by using the another PROC statement

Proc Varcomp Method=type1;

Class blk trt;

Model yield = blk trt/fixed = 1;

Run;

The above obtains the variance components using Hemderson's method. The methods of maximum likelihood, restricted maximum likelihood, minimum quadratic unbiased estimation can also be used by specifying method =ML, REML, MIVQE respectively. Objective 8 can be achieved by using PROC MIXED.

```
Proc Mixed ratio covtest;
Class blk trt;
Model yield = trt;
Random blk/s;
Lsmeans trt/pdiff;
Store lattice;
Run;
PROC PLM SOURCE = lattice;
LSMEANS trt /pdiff lines;
RUN;
```

Exercise 3.9: Analyze the data obtained through a Split-plot experiment involving the yield of 3 Irrigation (IRRIG) treatments applied to main plots and two Cultivars (CULT) applied to subplots in three Replications (REP). The layout and data (YLD) is given below:

Replication-I			Replication -II			Replication-III		
I1	I2	I3	I1	I2	I3	I1	I2	I3
C1	C1	C1	C1	C1	C1	C1	C1	C1
(1.6)	(2.6)	(4.7)	(3.4)	(4.6)	(5.5)	(3.2)	(5.1)	(5.7)
C2	C2	C2	C2	C2	C2	C2	C2	C2
(3.3)	(5.1)	(6.8)	(4.7)	(1.1)	(6.6)	(5.6)	(6.2)	(4.5)

Perform the analysis of the data. (HINT: Steps are given in text).

Remark 3.9.1: Another way proposed for analysis of split plot designs is using replication as random effect and analyse the data using PROC MIXED of SAS. For the above case, the steps for using PROC MIXED are:

```
PROC MIXED COVTEST;
CLASS rep irrig cult;
MODEL yield = irrig cult irrig*cult / DDFM=KR;
RANDOM rep rep*irrig;
LSMEANS irrig cult irrig*cult / PDIFF;
STORE spd;
run;
/* An item store is a special SAS-defined binary file format used to store and restore information with a hierarchical structure*/
```

```
/* The PLM procedure performs post fitting statistical analyses for the contents of a SAS item store that was previously created with the STORE statement in some other SAS/STAT procedure*/
```

```
PROC PLM SOURCE = SPD;
LSMEANS irrig cult irrig*cult /pdiff lines;
RUN;
```

Remark 3.9.2: In Many experimental situations, the split plot designs are conducted across environments and a pooled is required. One way of analysing data of split plot designs with two factors A and B conducted across environment is

```
PROC MIXED COVTEST;
CLASS year rep a b;
MODEL yield = a b a*b / DDFM=KR;
/* DDFM specifies the method for computing the denominator degrees of freedom for the tests of fixed effects
resulting from the MODEL*/
RANDOM year rep(year) year*a year*rep*a year*a*b;
LSMEANS a b a*b / PDIFF;
STORE spd1;
run;
PROC PLM SOURCE = SPD1;
LSMEANS a b a*b/pdiff lines;
RUN;
```

Exercise 3.10: An agricultural field experiment was conducted in 9 treatments using 36 plots arranged in 4 complete blocks and a sample of harvested output from all the 36 plots are to be analysed blockwise by three technicians using three different operations. The data collected is given below:

Block-1				Block-2			
Technician				Technician			
Operation	1	2	3	Operation	1	2	3
1	1(1.1)	2(2.1)	3(3.1)	1	1(2.1)	4(5.2)	7(8.3)
2	4(4.2)	5(5.3)	6(6.3)	2	2(3.2)	5(6.7)	8(9.9)
3	7(7.4)	8(8.7)	9(9.6)	3	3(4.5)	6(7.6)	9(10.3)
Block-3				Block-4			
Technician				Technician			
Operation	1	2	3	Operation	1	2	3
1	1(1.2)	6(6.3)	8(8.7)	1	1(3.1)	9(11.3)	5(7.8)
2	9(9.4)	2(2.7)	4(4.8)	2	6(8.1)	2(4.5)	7(9.3)
3	5(5.9)	7(7.8)	3(3.3)	3	8(10.7)	4(6.9)	3(5.8)

1. Perform the analysis of the data considering that technicians and operations are crossed with each other and nested in the blocking factor.
2. Perform the analysis by considering the effects of technicians as negligible.
3. Perform the analysis by ignoring the effects of the operations and technicians.

Procedure:

Prepare the data file.

```
DATA Name;
INPUT BLK TECH OPER TRT OBS;
Cards;
.....
```

....
....
;

Perform analysis of objective 1 using PROC GLM. The statements are as follows:

```
Proc glm;  
Class blk tech oper trt;  
Model obs= blk tech (blk) oper(blk) trt/ss2;  
Lsmeans trt oper(blk)/pdiff;  
Run;
```

Perform analysis of objective 2 using PROC GLM with the additional statements as follows:

```
Proc glm;  
Class blk tech oper trt;  
Model obs= blk oper(blk) trt/ss2;  
run;
```

Perform analysis of objective 3 using PROC GLM with the additional statements as follows:

```
Proc glm;  
Class blk tech oper trt;  
Model obs = blk trt/ss2;  
run;
```

Exercise 3.11: A greenhouse experiment on tobacco mosaic virus was conducted. The experimental unit was a single leaf. Individual plants were found to be contributing significantly to error and hence were taken as one source causing heterogeneity in the experimental material. The position of the leaf within plants was also found to be contributing significantly to the error. Therefore, the three positions of the leaves *viz.* top, middle and bottom were identified as levels of second factor causing heterogeneity. 7 solutions were applied to leaves of 7 plants and number of lesions produced per leaf was counted. Analyze the data of this experiment.

Plants							
Leaf Position	1	2	3	4	5	6	7
Top	1(2)	2(3)	3(1)	4(5)	5(3)	6(2)	7(1)
Middle	2(4)	3(3)	4(2)	5(6)	6(4)	7(2)	1(1)
Bottom	4(3)	5(4)	6(7)	7(6)	1(3)	2(4)	3(7)

The figures at the intersections of the plants and leaf position are the solution numbers and the figures in the parenthesis are number of lesions produced per leaf.

Procedure:

```
Prepare the data file.  
DATA Name;  
INPUT plant posi $ trt count;  
Cards;  
....  
....
```


....
;

Perform analysis using PROC GLM. The statements are as follows:

```
Proc glm;  
Class plant posi trt count;  
Model count= plant posi trt/ss2;  
Lsmeans trt/pdiff; Run;
```

Exercise 3.12: The following data was collected through a pilot sample survey on Hybrid Jowar crop on yield and biometrical characters. The biometrical characters were average Plant Population (PP), average Plant Height (PH), average Number of Green Leaves (NGL) and Yield (kg/plot).

1. Obtain correlation coefficient between each pair of the variables PP, PH, NGL and yield.
2. Fit a multiple linear regression equation by taking yield as dependent variable and biometrical characters as explanatory variables. Print the matrices used in the regression computations.
3. Test the significance of the regression coefficients and also equality of regression coefficients of a) PP and PH b) PH and NGL
4. Obtain the predicted values corresponding to each observation in the data set.
5. Identify the outliers in the data set.
6. Check for the linear relationship among the biometrical characters.
7. Fit the model without intercept.
8. Perform principal component analysis.

No.	PP	PH	NGL	Yield
1	142.00	0.5250	8.20	2.470
2	143.00	0.6400	9.50	4.760
3	107.00	0.6600	9.30	3.310
4	78.00	0.6600	7.50	1.970
5	100.00	0.4600	5.90	1.340
6	86.50	0.3450	6.40	1.140
7	103.50	0.8600	6.40	1.500
8	155.99	0.3300	7.50	2.030
9	80.88	0.2850	8.40	2.540
10	109.77	0.5900	10.60	4.900
11	61.77	0.2650	8.30	2.910
12	79.11	0.6600	11.60	2.760
13	155.99	0.4200	8.10	0.590
14	61.81	0.3400	9.40	0.840
15	74.50	0.6300	8.40	3.870
16	97.00	0.7050	7.20	4.470
17	93.14	0.6800	6.40	3.310
18	37.43	0.6650	8.40	1.570
19	36.44	0.2750	7.40	0.530
20	51.00	0.2800	7.40	1.150
21	104.00	0.2800	9.80	1.080
22	49.00	0.4900	4.80	1.830
23	54.66	0.3850	5.50	0.760

24	55.55	0.2650	5.00	0.430
25	88.44	0.9800	5.00	4.080
26	99.55	0.6450	9.60	2.830
27	63.99	0.6350	5.60	2.570
28	101.77	0.2900	8.20	7.420
29	138.66	0.7200	9.90	2.620
30	90.22	0.6300	8.40	2.000
31	76.92	1.2500	7.30	1.990
32	126.22	0.5800	6.90	1.360
33	80.36	0.6050	6.80	0.680
34	150.23	1.1900	8.80	5.360
35	56.50	0.3550	9.70	2.120
36	136.00	0.5900	10.20	4.160
37	144.50	0.6100	9.80	3.120
38	157.33	0.6050	8.80	2.070
39	91.99	0.3800	7.70	1.170
40	121.50	0.5500	7.70	3.620
41	64.50	0.3200	5.70	0.670
42	116.00	0.4550	6.80	3.050
43	77.50	0.7200	11.80	1.700
44	70.43	0.6250	10.00	1.550
45	133.77	0.5350	9.30	3.280
46	89.99	0.4900	9.80	2.690

Procedure:

Prepare a data file

Data mlr;

Input PP PH NGL Yield;

Cards;

....

....

;

For obtaining correlation coefficient, Use PROC CORR;

Proc Corr;

Var PP PH NGL Yield;

run;

For fitting of multiple linear regression equation, use PROC REG

Proc Reg;

Model Yield = PP PH NGL/ p r influence vif collin xpx i;

Test 1: Test PP =0; Test 2: Test PH=0;

Test 3: Test NGL=0;

Test 4: Test PP-PH=0;

Test 4a: Test PP=PH=0;

Test 5: Test PH-NGL=0;

Test 5a: Test PH=NGL=0;

Model Yield = PP PH NGL/noint;

run;

Proc reg;

Model Yield = PP PH NGL;

Restrict intercept =0;

Run;

For diagnostic plots

Proc Reg plots(unpack)=diagnostics;

Model Yield = PP PH NGL;

run;

For variable selection, one can use the following option in model statement:

Selection=stepwise sls=0.10;

For performing principal component analysis, use the following:

PROC PRINCOMP;

VAR PP PH NGL YIELD;

run;

Example 3.13: An experiment was conducted at Division of Agricultural Engineering, IARI, New Delhi for studying the capacity of a grader in number of hours when used with three different speeds and two processor settings. The experiment was conducted using a factorial completely randomised design in 3 replications. The treatment combinations and data obtained on capacity of grader in hours given as below:

Replication	speed	Processor setting	trt	cgrader
1	1	1	1	1852
1	1	2	2	1848
1	1	3	3	1855
1	2	1	4	2270
1	2	2	5	2279
1	2	3	6	2272
1	3	1	7	3035
1	3	2	8	3042
1	3	3	9	3028
2	1	1	1	1845
2	1	2	2	1855
2	1	3	3	1860
2	2	1	4	2276
2	2	2	5	2275
2	2	3	6	2248
2	3	1	7	3036
2	3	2	8	3033
2	3	3	9	3038
3	1	1	1	1851
3	1	2	2	1840
3	1	3	3	1840
3	2	1	4	2265

3	2	2	5	2280
3	2	3	6	2278
3	3	1	7	3040
3	3	2	8	3028
3	3	3	9	3040

Experimenter was interested in identifying the best combination of speed and processor setting that gives maximum capacity of the grader in hours.

Solution: This data can be analysed as per procedure of factorial CRD and one can use the following SAS steps for performing the analysis:

Data ex1a;

Input rep speed proset cgrader;

/*here rep: replication; proset: processor setting and cgrader: capacity of the grader in hours*/

Cards;

```

1 1 1 1852
1 1 2 1848
1 1 3 1855
. . . .
. . . .
. . . .
3 3 1 3040
3 3 2 3028
3 3 3 3040

```

;

Proc glm data=ex1;

Class speed proset;

Model cgrader=speed proset speed*proset;

Lsmeans speed proset speed*proset/pdiff adjust=tukey lines;

Run;

The above analysis would identify test the significance of main effects of speed and processor setting and their interaction. Through this analysis one can also identify the speed level (averaged over processor setting) {Processor Setting (averaged over speed levels)} at which the capacity of the grader is maximum. The multiple comparisons between means of combinations of speed and processor setting would help in identifying the combination at which capacity of the grader is maximum.

Exercise 3.14: An experiment was conducted with five levels of each of the four fertilizer treatments nitrogen, Phosphorus, Potassium and Zinc. The levels of each of the four factors and yield obtained are as given below. Fit a second order response surface design using the original data. Test the lack of fit of the model. Compute the ridge of maximum and minimum responses. Obtain predicted residual Sum of squares.

N	P ₂ O ₅	K ₂ O	Zn	Yield
---	-------------------------------	------------------	----	-------

40	30	25	20	11.28
40	30	25	60	8.44
40	30	75	20	13.29
40	90	25	20	7.71
120	30	25	20	8.94
40	30	75	60	10.9
40	90	25	60	11.85
120	30	25	60	11.03
120	30	75	20	8.26
120	90	25	20	7.87
40	90	75	20	12.08
40	90	75	60	11.06
120	30	75	60	7.98
120	90	75	60	10.43
120	90	75	20	9.78
120	90	75	60	12.59
160	60	50	40	8.57
0	60	50	40	9.38
80	120	50	40	9.47
80	0	50	40	7.71
80	60	100	40	8.89
80	60	0	40	9.18
80	60	50	80	10.79
80	60	50	0	8.11
80	60	50	40	10.14
80	60	50	40	10.22
80	60	50	40	10.53
80	60	50	40	9.5
80	60	50	40	11.53
80	60	50	40	11.02

Procedure:

Prepare a data file.

```
/* yield at different levels of several factors */
```

```
title 'yield with factors N P K Zn';
```

```
data dose;
```

```
input n p k Zn y ; label y = "yield" ;
```

```
cards;
```

```
.....
```

```
.....
```

```
.....
```

```
;
```

```
*Use PROC RSREG.
```

```
ods graphics on;
```

```
proc rsreg data=dose plots(unpack)=surface(3d);
```

```
model y= n p k Zn/ nocode lackfit press;
```

```
run;
```

```
ods graphics off; *If we do not want surface plots, then we may
```

```
proc rsreg;
model y= n p k Zn/ nocode lackfit press;
Ridge min max;
run;
```

Exercise 3.15: Fit a second order response surface design to the following data. Take replications as covariate.

Fertilizer1	Fertilizer2	X ₁	X ₂	Yields(lb/plot)	
				Replication I	Replication II
50	15	-1	-1	7.52	8.12
120	15	+1	-1	12.37	11.84
50	25	-1	+1	13.55	12.35
120	25	+1	+1	16.48	15.32
35	20	$-\sqrt{2}$	0	8.63	9.44
134	20	$+\sqrt{2}$	0	14.22	12.57
85	13	0	$-\sqrt{2}$	7.90	7.33
85	27	0	$+\sqrt{2}$	16.49	17.40
85	20	0	0	15.73	17.00

Procedure:

```
Prepare a data file.
/* yield at different levels of several factors */
title 'yield with factors x1 x2';
data respcov;
input fert1 fert2 x1 x2 yield ;
cards;
.....
.....
.....
;
/*Use PROC RSREG.*/
ODS Graphics on;
proc rsreg plots(unpack)=surface(3d);
model yield = rep fert1 fert2/ covar=1 nocode lackfit ;
Ridge min max;
run;
```

Exercise 3.16: Following data is related to the length(in cm) of the ear-head of a wheat variety 9.3, 18.8, 10.7, 11.5, 8.2, 9.7, 10.3, 8.6, 11.3, 10.7, 11.2, 9.0, 9.8, 9.3, 10.3, 10, 10.1 9.6, 10.4. Test the data that the median length of ear-head is 9.9 cm.

Procedure:

This may be tested using any of the three tests for location available in Proc Univariate viz. Student's *t* test, the sign test, and the Wilcoxon signed rank test. All three tests produce a test statistic for the null hypothesis that the mean or median is equal to a given value μ_0 against the two-sided alternative that the mean or median is not equal to μ_0 . By default, PROC UNIVARIATE sets the value of μ_0 to zero. You can use the MU0= option in the PROC UNIVARIATE statement to specify the value of μ_0 . If the data is from a normal population,

then we can infer using t-test otherwise non-parametric tests sign test, and the Wilcoxon signed rank test may be used for drawing inferences.

Procedure:

```

data npsign;
input length;
cards;
9.3
18.8
10.7
11.5
 8.2
 9.7
10.3
 8.6
11.3
10.7
11.2
 9.0
 9.8
 9.3
10.3
10.0
10.1
 9.6
10.4
;
PROC UNIVARIATE DATA=npsign MU0=9.9;
VAR length;
HISTOGRAM / NO PLOT ;
      RUN;
QUIT;

```

Exercise 3.17: An experiment was conducted with 21 animals to determine if the four different feeds have the same distribution of Weight gains on experimental animals. The feeds 1, 3 and 4 were given to 5 randomly selected animals and feed 2 was given to 6 randomly selected animals. The data obtained is presented in the following table.

Feeds	Weight gains (kg)					
1	3.35	3.8	3.55	3.36	3.81	
2	3.79	4.1	4.11	3.95	4.25	4.4
3	4	4.5	4.51	4.75	5	
4	3.57	3.82	4.09	3.96	3.82	

Procedure:

```

data np;
input feed wt;
datalines;

```

```

1    3.35
1    3.80
1    3.55
1    3.36
1    3.81
2    3.79
2    4.10
2    4.11
2    3.95
2    4.25
2    4.40
3    4.00
3    4.50
3    4.51
3    4.75
3    5.00
4    3.57
4    3.82
4    4.09
4    3.96
4    3.82

```

```

;
PROC NPAR1WAY DATA=np WILCOXON; /*for performing Kruskal-Walis test*/;
  VAR wt;
  CLASS feed;
RUN;

```

Example 3.18: Finney (1971) gave a data representing the effect of a series of doses of carotene (an insecticide) when sprayed on *Macrosiphoniella sanborni* (some obscure insects). The Table below contains the concentration, the number of insects tested at each dose, the proportion dying and the probit transformation (probit+5) of each of the observed proportions.

Concentration (mg/l)	No. of insects (n)	No. of affected (r)	%kill (P)	Log concentration (x)	Empirical probit
10.2	50	44	88	1.01	6.18
7.7	49	42	86	0.89	6.08
5.1	46	24	52	0.71	5.05
3.8	48	16	33	0.58	4.56
2.6	50	6	12	0.41	3.82
0	49	0	0	-	-

Perform the probit analysis on the above data.

Procedure

```

data probit;
input con n r;

```



```

datalines;
10.2 50 44
7.7 49 42
5.1 46 24
3.8 48 16
2.6 50 6
0 49 0
;
ods html;
Proc Probit log10 ;
Model r/n=con/lackfit inversecl;
title ('output of probit analysis');
run;
ods html close;

```

Model Information	
Data Set	WORK.PROBIT
Events Variable	r
Trials Variable	n
Number of Observations	5
Number of Events	132
Number of Trials	243
Name of Distribution	Normal
Log Likelihood	-120.0516414
Number of Observations Read	6
Number of Observations Used	5
Number of Events	132
Number of Trials	243

Algorithm converged.

Goodness-of-Fit Tests			
Statistic	Value	DF	Pr > ChiSq
Pearson Chi-Square	1.7289	3	0.6305
L.R. Chi-Square	1.7390	3	0.6283

Response-Covariate Profile	
Response Levels	2
Number of Covariate Values	5

Since the chi-square is small ($p > 0.1000$), fiducial limits will be calculated using a t value of 1.96

Type III Analysis of Effects			
Wald			
Effect	DF	Chi-Square	Pr > ChiSq
Log10(con)	1	77.5920	<.0001

Analysis of Parameter Estimates							
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-2.8875	0.3501	-3.5737	-2.2012	68.01	<.0001
Log10(con)	1	4.2132	0.4783	3.2757	5.1507	77.59	<.0001

Probit Model in Terms of Tolerance Distribution	
MU	SIGMA
0.68533786	0.23734947

Estimated Covariance Matrix for Tolerance Parameters		
	MU	SIGMA
MU	0.000488	-0.000063
SIGMA	-0.000063	0.000726

Probit Analysis on Log10(con)			
Probability	Log10(con)	95% Fiducial Limits	
0.01	0.13318	-0.03783	0.24452
0.02	0.19788	0.04453	0.29830
0.03	0.23893	0.09668	0.33253
0.04	0.26981	0.13584	0.35834
0.05	0.29493	0.16764	0.37940
0.06	0.31631	0.19466	0.39737
0.07	0.33506	0.21832	0.41316
0.08	0.35184	0.23946	0.42733
0.09	0.36711	0.25866	0.44026
0.10	0.38116	0.27631	0.45218
0.15	0.43934	0.34898	0.50192
0.20	0.48558	0.40618	0.54202
0.25	0.52525	0.45467	0.57700
0.30	0.56087	0.49759	0.60904
0.35	0.59388	0.53666	0.63942
0.40	0.62521	0.57295	0.66905
0.45	0.65551	0.60716	0.69861
0.50	0.68534	0.63983	0.72870
0.55	0.71516	0.67142	0.75986
0.60	0.74547	0.70240	0.79265
0.65	0.77679	0.73330	0.82766
0.70	0.80980	0.76480	0.86563
0.75	0.84543	0.79777	0.90761
0.80	0.88510	0.83352	0.95533
0.85	0.93133	0.87427	1.01188
0.90	0.98951	0.92456	1.08401
0.91	1.00357	0.93658	1.10155
0.92	1.01883	0.94960	1.12065
0.93	1.03562	0.96387	1.14170
0.94	1.05436	0.97976	1.16526
0.95	1.07574	0.99783	1.19218
0.96	1.10086	1.01898	1.22388
0.97	1.13174	1.04490	1.26294
0.98	1.17279	1.07924	1.31498
0.99	1.23750	1.13315	1.39721

Probit Analysis on con			
Probability	con	95% Fiducial Limits	
0.01	1.35888	0.91657	1.75599
0.02	1.57718	1.10799	1.98745
0.03	1.73353	1.24935	2.15043
0.04	1.86129	1.36724	2.28215

Probit Analysis on con			
Probability	con	95% Fiducial Limits	
0.05	1.97212	1.47110	2.39553
0.06	2.07163	1.56554	2.49671
0.07	2.16302	1.65317	2.58917
0.08	2.24825	1.73565	2.67506
0.09	2.32868	1.81410	2.75586
0.10	2.40526	1.88932	2.83257
0.15	2.75005	2.23349	3.17629
0.20	3.05900	2.54788	3.48353
0.25	3.35157	2.84884	3.77571
0.30	3.63808	3.14478	4.06477
0.35	3.92538	3.44084	4.35935
0.40	4.21897	3.74068	4.66710
0.45	4.52389	4.04724	4.99582
0.50	4.84549	4.36343	5.35423
0.55	5.18995	4.69265	5.75260
0.60	5.56506	5.03963	6.20374
0.65	5.98127	5.41132	6.72450
0.70	6.45363	5.81830	7.33883
0.75	7.00531	6.27722	8.08377
0.80	7.67532	6.81590	9.02252
0.85	8.53758	7.48633	10.27723
0.90	9.76143	8.40534	12.13411
0.91	10.08243	8.64132	12.63428
0.92	10.44313	8.90434	13.20233
0.93	10.85466	9.20181	13.85792
0.94	11.33346	9.54469	14.63036
0.95	11.90537	9.95006	15.56609
0.96	12.61427	10.44674	16.74479
0.97	13.54388	11.08927	18.32046
0.98	14.88655	12.00168	20.65263
0.99	17.27807	13.58779	24.95808

Interpretation: The goodness-of-fit tests (p-values = 0.6305, 0.6283) suggest that the distribution and the model fits the data adequately. In this case, the fitting is done on normal equivalent deviate only without adding 5. Therefore, log LD50 or lof ED50 corresponds to the value of Probit=0. Log LD50 is obtained as 0.685338. Therefore, the stress level at which the 50% of the insects will be killed is $(10^{0.685338}=4.845$ mg/l). Similarly the stress level at which 65% of the insects will be killed is $(10^{0.776793} = 5.981$ mg/l). Although both values are given in the table above.

4. Discussion

We have initiated a link “Analysis of Data” at Design Resources Server (www.iasri.res.in/design) to provide steps of analysis of data generated from designed experiments by using statistical packages like SAS, SPSS, MINITAB, and SYSTAT, MS-EXCEL etc. For details and live examples one may refer to the link Analysis of data at <http://www.iasri.res.in/design/Analysis%20of%20data/Analysis%20of%20Data.html>.

How to see SAS/STAT Examples?

One can learn from the examples available at <http://support.sas.com/rnd/app/examples/STATexamples.html>

How to use HELP?

Help → SAS help and Documentation → Contents → Learning to use SAS → Sample SAS Programs → SAS/STAT ...

5. Strengthening Statistical Computing for NARS

NAIP Consortium on Strengthening Statistical Computing for NARS (www.iasri.res.in/sscnars) targets at providing

- research guidance in statistical computing and computational statistics and creating sound and healthy statistical computing environment
- Providing advanced, versatile, and innovative and state-of the art high end statistical packages to enable them to draw meaningful and valid inferences from their research.

The efforts also involve designing of intelligent algorithms for implementing statistical techniques particularly for analysing massive data sets, simulation, bootstrap, etc.

The objectives of the consortium are:

- To strengthen the high end statistical computing environment for the scientists in NARS;
- To organize customized training programmes and also to develop training modules and manuals for the trainers at various hubs; and
- To sensitize the scientists in NARS with the statistical computing capabilities available for enhancing their computing and research analytics skills.

This consortium has provided the platform for closer interactions among all NARS organizations.

Capacity Building

For capacity building of researchers in the usage of high end statistical computing facility and statistical techniques,

- **209** trainers have been trained through 30 working days training programmes;
- **2166** researchers have been trained through 104 training programmes of one week duration each in the usage.

The capacity building efforts have paved the way for publishing research papers in the high impact factor journals.

Indian NARS Statistical Computing Portal

For providing service oriented computing, developed and established Indian NARS Statistical Computing portal, which is available to NARS users through IP authentication at <http://stat.iasri.res.in/sscnarsportal>. Any researcher from Indian NARS may obtain User name and password from Nodal Officers of their respective NARS organizations, list available at www.iasri.res.in/sscnars. It is a paradigm of computing techniques that operate on software-as-a-service). There is no need of installation of statistical package at client side. Following 24 different modules of analysis of data are available on this portal, which have been classified into four broad categories as

Basic Statistics

- Descriptive Statistics
- Univariate Distribution Fitting
- Test of Significance based on t-test
- Test of Significance based on Chi-square test
- Correlation Analysis
- Regression Analysis

Designs of Experiments

- Completely randomized designs
- Block Designs (includes both complete and incomplete block designs)
- Combined Block Designs
- Augmented Block Designs
- Resolvable Block Designs
- Nested Block Designs
- Row-Column Designs
- Cross Over Designs
- Split Plot Designs
- Split-Split-Plot Designs
- Split Factorial (main A, sub $B \times C$) designs
- Split Factorial (main $A \times B$, sub $C \times D$) designs
- Strip Plot Designs
- Response Surface Designs

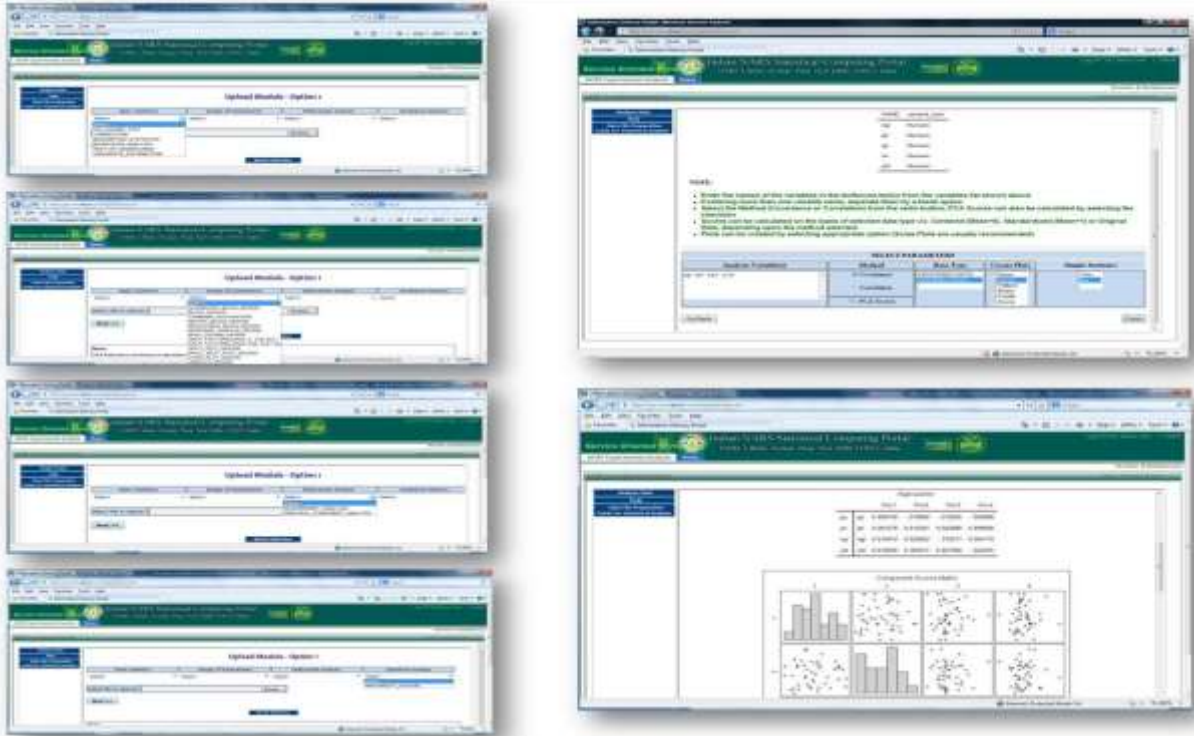
Multivariate Analysis

- Principal Component Analysis
- Linear Discriminant Analysis

Statistical Genetics

- Estimation of Heritability from half- sib data
- Estimation of variance-Covariance matrix from Block Designs

The above modules can be used by uploading *.xlsx, *.csv and *.txt files and results can be saved as *.RTF or *.pdf files. This has helped them in analyzing their data in an efficient manner without losing any time.



Requirements of Excel Files during analysis over Indian NARS Statistical Computing Portal

1. Excel file must have the .xls, .xlsx, .csv or .txt extensions
2. This system will only consider the first sheet of the excel file which has name appearing first in lexicographic order. It will not analyze the data which lies in subsequent sheets in excel file.
3. Do not put period (.) or Zero (0) to display missing values in the treatment. It will not consider as missing. Please leave the missing observations as blank cells.
4. If you are getting some wrong analysis then kindly check your excel file. Go to First Column, first cell and then press Ctrl+Shift+End. It will select all the filled rows and columns. If it selects some missing rows and columns then kindly delete those rows and columns otherwise it will give wrong analysis result.
5. Do not use special characters in the variable/column names. Also variable names should not start with spaces.
6. Do not use any formatting to the Excel sheet including formats or expressions to the cell values. It should be data value.
7. If the First row cells has been merged then it will not detect as Column/Variable names.
8. If any rows or columns are hidden then it will be displayed during the analysis.

Basic Statistics

9. **Descriptive Statistics:** The data file should contain at least one quantitative analysis variable.
10. **Univariate Distribution Fitting:** The data file should contain at least one quantitative numeric variable.
11. **Test of Significance based on t-distribution:** The data file should contain at least one quantitative variable name and one classificatory variable.

12. **Chi-Square Test:** The data file should contain at least one categorical variable and weights or frequency counts variable if frequencies are entered in a separate column. Data may also have classificatory in it.
13. **Correlation:** The data file should contain at least two quantitative variables.
14. **Regression Analysis:** The data file should contain at least one Dependent and one Independent variable.

Design of Experiments

15. **Unblock Design:** Prepare a data file containing one variable to describe the Treatment details and at least one response/ dependent variable in the experimental data to be analyzed. Also, the treatment details may be coded or may have actual names (i.e. data values, for variable describing treatment column may be in numeric or character). The maximum length of treatment value is 20 characters. The variables can be entered in any order.
16. **Block Design:** Prepare a data file containing two variables to describe the block and treatment details. There should be at least one response/ dependent variable in the experimental data to be analyzed. Also, the block/treatment details may be coded or may have actual names (i.e. data values, for variables describing block and treatment column may be in numeric or character). The maximum length of treatment value is 20 character. The variables can be entered in any order. (These conditions are applicable to other similar experimental designs also)
17. **Combined Block Design:** The data file should contain three variables to describe Environment, Block, Treatment variables and at least one Dependent variable.
18. **Augmented Block Design:** The data file should contain two variables to describe Block & Treatment variables and at least one Dependent variable. At present, Portal supports only numeric treatment and block variables for augmented designs. An augmented block design involves two sets of treatments known as check or control and test treatments. The treatments should be numbered in such a fashion that the check or control treatments are numbered first followed by test treatments. For example, if there are 4 control treatments and 8 test treatments, then the control treatments are renumbered as 1, 2, 3, 4 and tests are renumbered as 5, 6, 7, 8, 9, 10, 11, 12.
19. **Resolvable Block Design:** The data file should contain three variables to describe the Replication, Block, Treatment variables and at least one Dependent/ response variable.
20. **Nested Block Design:** The data file should contain three variables to describe Block, SubBlock, Treatment variables and at least one Dependent variable.
21. **Row Column Design:** The data file should contain three variables to describe Row, Column, Treatment variables and at least one Dependent variable.
22. **Crossover Design:** Create a data file with at least 5 variables, one for units, one for periods, one treatments, one for residual, and one for the dependent or analysis variable. For performing analysis using the portal, please rearrange the data in the following order: animal numbers as units; periods can be coded as 1, 2, 3, and so on, treatments as alphabets or numbers (coding could be done as follows: for every first period the number one has assigned (fixed) and for other periods code 1 to 3 are given according to the treatment received by the unit in the previous period) and residual effect as residual. It may, however, be noted that one can retain the same name or can

code in any other fashion. A carry-over or residual term has the special property as a factor, or class variate, of having no level in the first period because the treatment in the first period is not affected by any residual or carry over effect of any treatment. When we consider the residual or carryover effect in practice the fact that carry-over or residual effects will be adjusted for period effects (by default all effects are adjusted for all others in these analysis). As a consequence, any level can be assigned to the residual variate in the first period, provided the same level is always used. An adjustment for periods then removes this part of the residual term. (For details a reference may made to Jones, B. and Kenward,M.G. 2003. Design and Analysis of Cross Over Trials. Chapman and Hall/CRC. New York . Pp: 212)

23. **Split Plot Design:** The data file should contain three variables to describe Replication, Main Plot, Sub Plot variables and at least one Dependent variable.
24. **Split Split Plot Design:** The data file should contain four variables to describe Replication, Main Plot, Sub Plot, and Sub-Sub Plot Treatment variables and at least one Dependent variable.
25. **Split Factorial (Main A, Sub B×C) Plot Design** The data file should contain four variables to describe Replication, Main Plot, Sub Plot(1){levels of factor 1 in sub plot} , and Sub Plot(2)){levels of factor 21 in sub plot} Treatment variables and at least one Dependent variable.
26. **Split Factorial (Main A×B, Sub C×D) Plot Design:** Create a data file with at least 6 variables, one for block or replication, one for main plot- treatment factor 1, one main plot- treatment factor 2, one for subplot- treatment factor 1, one for subplot- treatment factor 2 and at least one for the dependent or analysis variable. If the data on more than one dependent variable is collected in the same experiment, the data on all variables may be entered in additional columns. One may give actual levels used for different factors applied in main plot-treatment factor 1, main plot- treatment factor 2, subplot- treatment factor 1 and subplot- treatment factor 2. Please remember that there should not be any space between a single data value. Main plot- treatment factor 1, main plot- treatment factor 2, subplot- treatment factor 1, subplot- treatment factor 2 treatments and block numbers may be coded as 1, 2, 3 and so on. One can have character values also.
27. **Strip Plot Design:** The data file should contain at least 4 variables to describe Replication, Horizontal Strip, Vertical Strip variables and at least one Dependent variable.
28. **Response Surface Design:** The data file should contain at least one treatment factor variable and at least one dependent variable

Multivariate Analysis

29. **Principal Component Analysis:** The data file should contain at least one quantitative analysis variable.
30. **Discriminant Analysis:** The data file should contain at least one quantitative analysis variable and a classificatory variable.

Statistical Genetics

31. **Genetic Variance Covariance:** Create a data file with at least 4 variables, one for blocking variable, one for treatments and at least two analysis variable.
32. **Heritability Estimation from Half-Sib Data:** The data file should contain at least one quantitative analysis variable and a classificatory variable.

Other IP Authenticated Services

Following can also be accessed through IP authenticated networks:

- Web Report Studio: <http://stat.iasri.res.in/sscnarswebreportstudio>
- BI DashBoard: <http://stat.iasri.res.in/sscnarsbidashboard>
- Web OLAP Viewer: <http://sas.iasri.res.in:8080/sscnarswebolapviewer>
- E-Miner 6.1: <http://sas.iasri.res.in:6401/AnalyticsPlatform>
- E-Miner 7.1: <http://stat.iasri.res.in/SASEnterpriseMinerJWS/Status>

Accessing SAS E-Miner through URL (IP Authenticated Services)

For Accessing E-miner 6.1 and 7.1 through URLs, following ports should be open

Server	Ports
1) Metadata server	8561
2) Object spawner	8581
3) Table Server	2171
4) Remote Server	5091
5) SAS App. Olap Server	5451
6) SAS Deployment Tester Server	10021
7) Analytics Platform Server	6411
8) Framework Server	22031

However, if you are accessing only E-miner 6.1, then following port need not be opened.

Framework Server	22031
------------------	-------

Steps for accessing SAS Enterprise Miner 6.1 and SAS Enterprise Miner 7.1 separately SAS Enterprise Miner 6.1

Pre-requisite:

- JRE 1.5 Update 15
- If Firewall and proxy has been implemented then kindly open following ports:

Server	Ports
1) Metadata server	8561
2) Object spawner	8581
3) Table Server	2171
4) Remote Server	5091
5) SAS App. OLAP Server	5451
6) SAS Deployment Tester Server	10021
7) Analytics Platform Server	6411

Steps to be followed:

- If you have installed multiple Java Runtime Environment then
Go to Control Panel → Java → Java tab → View → Keep check on JRE 1.5.0_15 and
Uncheck all others
- Check the entry of the **sas.iasri.res.in** in the host file, if not then open host file
C:\Windows\System32\drivers\etc and edit the host file by entering the IP as shown
below or specify the internal/external IP given by IASRI. Internal IP is to be specified

only at IASRI, New Delhi. All other NARS organizations should specify external IP only which is: 203.197.217.209 sas.iasri.res.in sas as shown below



- Now Go to URL: <http://sas.iasri.res.in:6401/AnalyticsPlatform>
- Click on Launch and then Run

SAS Enterprise Miner 7.1

Pre-requisite:

- JRE 1.6 Update 16 or higher
- If Firewall and/or proxy has been implemented then kindly open the following ports:

Server	Ports
1) Metadata server	8561
2) Object spawner	8581
3) Framework Server	22031
4) Remote Server	5091
5) SAS App. Olap Server	5451
6) SAS Deployment Tester Server	10021

Steps to be followed:

- If you have installed multiple Java Runtime Environment then
Go to Control Panel → Java → Java tab → View → Keep check on JRE 1.6.0_16 or higher available version and Uncheck all other
- Check the entry of the **stat.iasri.res.in** in the host file, if not then open host file **C:\Windows\System32\drivers\etc** and edit the host file by entering the IP as shown below or specify the internal/external IP given by IASRI, New Delhi. Internal IP is to be specified only at IASRI, New Delhi. All other NARS organizations should specify external IP only which is: 14.139.56.156 stat.iasri.res.in stat (earlier 203.197.217.221 stat.iasri.res.in stat) as shown below stat.iasri.res.in stat as shown below

```

hosts - Notepad
File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
# This is a sample hosts file used by Microsoft TCP/IP for Windows.
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10       x.acme.com         # x client host
#
# localhost name resolution is handled within DNS itself.
#
#       127.0.0.1       localhost       L0174INA2.apac.sas.com  L0174INA2.in.sas.com  L0174INA2
#       ::1            localhost
#       127.0.0.1       www.presentation-3d.com
#
#to access Eminer 7.1 internally
10.10.10.21       stat.iasri.res.in       stat
#to access Eminer 7.1 Externally
203.197.217.221  stat.iasri.res.in       stat

```

- Now Go to URL: <http://stat.iasri.res.in/SASEnterpriseMinerJWS/Status>
- Click on Launch and then Run

Please note: You cannot run both E-Miner 6.1 and E-Miner 7.1 together. If you want to run JMP 6.1 then JAVA 1.5.0_15 should be available and for running JMP 7.1, JAVA version 1.6 onwards should be available on your system.

Indian NARS Statistical Computing Portal and other IP authenticated services are best viewed in **Internet Explorer 6 to 8 and Firefox 2.0.0.11 and 3.0.6**

Macros Developed

Macros have been developed for some commonly used statistical analysis and made available at Project Website www.iasri.res.in/sscnars. Following macros have been developed:

- **Analysis of Experimental Data**
 - Analysis of data from Augmented Block designs
<http://www.iasri.res.in/sscnars/augblkdsgn.aspx>
 - Analysis of data from Split Factorial (main A, Sub B × C) designs
<http://www.iasri.res.in/sscnars/spltfctdsgn.aspx>
 - Analysis of data from Split Factorial (Main A×B, Sub C) designs
<http://www.iasri.res.in/sscnars/spltfctdsgnm2s1.aspx>
 - Analysis of data from Split Factorial (main A×B, Sub C × D) designs
<http://www.iasri.res.in/sscnars/spltfactm2s2.aspx>
 - Analysis of data from Split Split Plot designs
<http://www.iasri.res.in/sscnars/spltpltdsgn.aspx>
 - Analysis of data from Strip Plot designs
<http://www.iasri.res.in/sscnars/StripPlot.aspx>
 - Analysis of data from Strip-Split Plot designs
<http://www.iasri.res.in/sscnars/stripsplit.aspx>
- **Econometric Analysis**
 - Econometric Analysis ((diversity indices, instability index, compound growth rate, Garret scoring technique and Demand analysis using LA-AIDS model) and available at <http://www.iasri.res.in/sscnars/ecoanalysis.aspx>
- **Generation of Designs**

- Generation of Polycross designs
<http://www.iasri.res.in/sscnars/polycrossdesign.aspx>
- Generation of TFNBCB designs
<http://www.iasri.res.in/sscnars/TFNBCBdesigns.aspx>
- Generating Treatment Combination SFTSMCRS
<http://www.iasri.res.in/sscnars/sftsmcrs.aspx>
- **Statistical Genetics**
Estimation of heritability along with its standard error from half sib data
<http://www.iasri.res.in/sscnars/heritability.aspx>

How to see updated version of reference manual?

Reference manual is updated regularly and updated version may be downloaded from
<http://www.iasri.res.in/sscnars/contentmain.htm>

References

Littel, R.C., Freund, R.J. and Spector, P.C. (1991). *SAS System for Linear Models, Third Edition*. SAS Institute Inc.

Searle, S.R. (1971). *Linear Models*. John Wiley & Sons, New York.

Searle, S.R., Casella, G and McCulloch, C.E. (1992). *Analysis of Variance Components*. John Wiley & Sons, New York.

www.sas.com;

www.support.sas.com

www.iasri.res.in/design

www.iasri.res.in/sscnars

<http://stat.iasri.res.in/sscnarsportal>

An Introduction to R Software

B. N. Mandal
ICAR-IASRI, New Delhi
bn.mandal@icar.gov.in

1 Introduction

R is a powerful programming language for statistical analysis. This software is an implementation of S programming language which was designed by John Chambers at Bell Labs. R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand. It is currently developed by R Development Core Team. The name ‘R’ is from the first names of first two authors and partly due to its inheritance from ‘S’. Recently R has become one of world’s popular statistical analysis software, because of the following reasons:

- (i) R is free, open source and capable of almost any statistical analysis including the most recently developed statistical methodologies.
- (ii) R provides very good graphical facilities.
- (iii) R is easily extensible through new contributions from statisticians and researchers around the globe, which also makes R quite different from other popular statistical analysis software. In fact, R community is highly active in terms of new contributions in the form of packages to R.

2 Getting and installing R

To be able to use R, it needs to be installed in computer. R is available for free download from any one of the mirror sites of Comprehensive R Archive Network (CRAN) in <http://cran.r-project.org/>. For downloading, it is better to select a mirror located nearer to you. R is available for installation in Windows/Macintosh/Unix platforms. To install R in a given machine, first double-click the downloaded file R.exe, then select language as ‘English’. R setup wizard window will appear. Select on ‘Next’ and accept most of the default settings during the installation. Latest version as on 26.11.2019 available is 3.6.1.

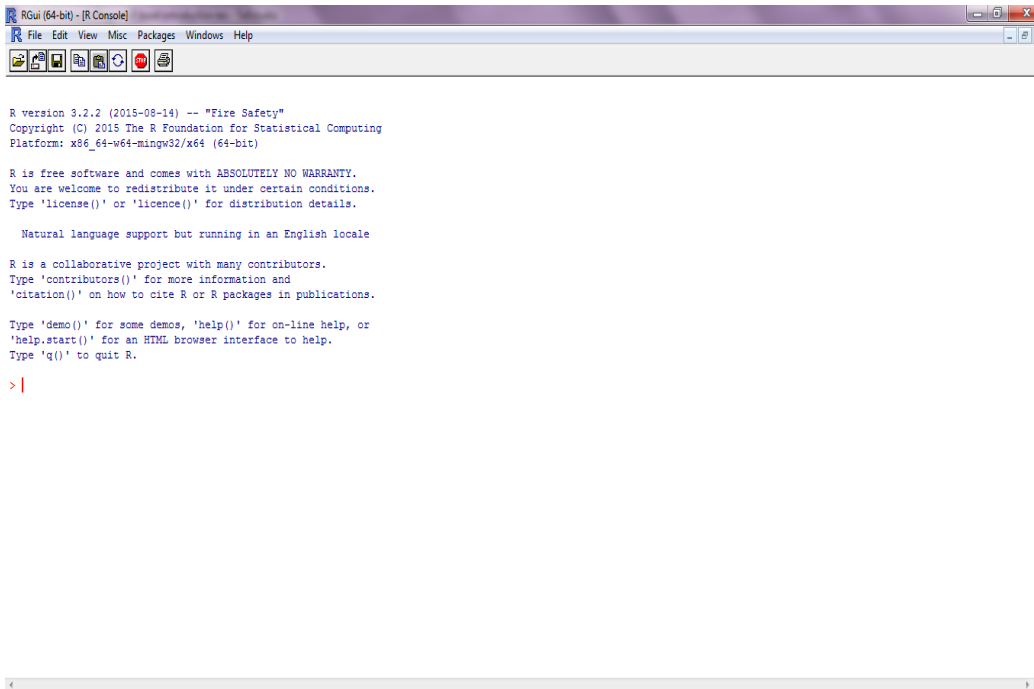


Figure 1: R console window

3 Using R

3.1 Starting R

To start R, click on start menu → all programs → R → R icon and a screen as shown in Figure 1 appears. The white blank screen is called R Console and this is the place where all R codes are written and outputs appear, unless outputs are directed to some external files.

There will be a toolbar at the top of the Console and a few menus in the R window. To know what the buttons on the toolbar does, hold your mouse on the button for some time, a description of the button will appear. There is an R editor which can be used to write and edit R codes. R editor window is just like a text editor with facilities for select, cut, copy, paste, typing text, deleting text etc. This window opens by clicking on File → New Script in the menu bar. The codes written in R editor window needs to be passed to the R console for execution by clicking on ‘Run line or selection button’ on the toolbar in the R editor window

3.2 R commands

There is a > symbol in the Console. The commands are typed after this symbol and then the Enter button needs to be pressed. When a command is written in the Console and the Enter button is pressed, R reads the commands and returns some results or some error message on the Console. For example, if you type

```
> 2+8
```

```
[1] 10
```

In this case R added 2 and 8 and returned the result 10. Now, type

```
> 2+*5
Error: unexpected '*' in "2+*"
```

This time R has returned an error message, because '+' is not a defined operator in R. Therefore, one should know what to type in the Console, otherwise, it will always give an error message. So R works interactively by returning results to the commands one by one, like in the following example:

```
> x=2
> x*5
[1] 10
```

R is mainly an expression language and comes with a syntax. Some common rules of R commands are

- (i) R commands are case sensitive, so AB, Ab, aB and ab are different objects.
- (ii) All alphanumeric characters, '.' and '_' are allowed as symbols. In some countries accented letters are also allowed.
- (iii) An R name must start with '.' or a letter, and if it starts with '.' the second character must not be a digit. Names are unlimited in length.
- (iv) Elementary commands consist of either expressions or assignments. An expression is evaluated, printed (unless specifically made invisible), and the value is lost. An assignment also evaluates an expression and passes the value to a variable but the result is not automatically printed.
- (v) Commands are separated either by a semi-colon (;), or by a newline.
- (vi) Curly braces '{' and '}' are used to create a block of codes.
- (vii) Any line starting with a # till the end of the line is a comment and are not evaluated. Comments can be placed anywhere.
- (viii) If a command is not complete at the end of a line, R will give a different prompt, by default '+' on second and subsequent lines and continue to read input until the command is syntactically complete. This prompt may be changed by the user.
- (ix) Length of a command at the Console is limited to about 4095 bytes (not characters).

3.3 Working directory

The working directory refers to the directory or folder where R is currently working. By default the working directory is “My documents” or ‘Documents’. You can get the working directory by using code

```
> getwd()
[1] "C:/Users/User/Documents"
```

R can read and open files from working directory directly without specifying any path. Similarly, it can save files and write to files in the working directory directly. One can reset the working directory to a different folder using the code below.

```
> setwd("C:/Users/User/Documents/DOE handbook/")
```

In the beginning of an R session, it is better to set the working directory to a folder where most of the data files and codes are located.

3.4 Data types in R

R is an object oriented language and therefore, all data types in R are some kind of object. Objects may be variables, vectors, matrices, arrays, character strings, functions, or more general structures built from such components. During an R session, objects are created and stored by name. One can use the command

```
> objects()
```

to display the names of the objects which are currently stored within R. The collection of objects currently stored is called the workspace. One can remove objects using the function `rm()`. For example, the following code removes objects `x` and `y` from the workspace.

```
> rm(x, y)
```

An object created during an R session can be saved in a file for use in future R sessions. The entire workspace of an R session and the history of all the commands used during the session can also be saved. Some commonly encountered objects are discussed below.

a) Vectors: Simplest object in R is a vector. A vector is a collection of elements. For example,

```
> x = c(10, 15, 20, 25, 26)
```

creates a vector of 5 numbers. Here the object `x` contains those numbers and the function `c()` is used to assign those numbers to the object `x`. Vectors can be of three types i) numeric ii) character and iii) logical. A numeric vector contains numbers, a character vector contains characters and a logical vector can contain values TRUE, FALSE or NA.

- b) Matrices: A matrix object also is a collection of elements but it has two dimensions. They can also be numeric, character or logical in nature. Following is an example of creating a matrix.

```
> x=matrix(c("a","b","c","d"),nrow=2)
> x
[,1] [,2]
[1,] "a"  "c"
[2,] "b"  "d"
```

- c) Arrays: Arrays are multi-dimensional generalization of vectors and matrices. A two-dimensional array is a matrix. Arrays can have more than two dimensions.
- d) Factors: Factor objects are used to specify categorical or classificatory or grouping variables. For example, males and females are two levels of a variable gender. Then gender can be thought of a factor object.

```
> gender=c("M", "F", "M")
> gender=as.factor(gender)
> levels(gender)
[1] "F" "M"
```

Factor variables are particularly useful in analysis of variance and in linear model with grouping variables.

- e) Lists: A list is a collection of objects where each object can be of different type. For example, a list can have first object as a vector, second object as a matrix and third object as a data frame.

```
> mylist=list(x=c(10,20,30),y=matrix(1:6,nrow=3))
> mylist
$x
[1] 10 20 30
$y
[,1] [,2]
[1,]  1   4
[2,]  2   5
[3,]  3   6

> mylist[[1]]
[1] 10 20 30
```

- f) Data frames: A data frame is a two dimensional object. But unlike matrices, different columns of data frame can be different types, for example some columns can be numeric, some columns can be character, some columns can be factors. Here a column generally refers to a variable.

```

> age=c(20,25,28,30,26)
> weight=c(50,53,54,55,51)
> mydata=data.frame(age,weight)
> mydata
  age weight
1  20     50
2  25     53
3  28     54
4  30     55
5  26     51

```

The `data.frame()` function is used to create a data frame.

- g) Functions: Functions in R are a kind of objects which takes one or more inputs and produces some result(s) as output. R has a number of in-built functions. R also provides facility to create new functions by users. R has huge number of in-built functions. As a simple example, to obtain the mean and variance of a set of numbers 10, 13, 21, 34, 51, 32, 45, 32, 17, 29, 41, 52, the following code can be used.

```

> x=c(10,13,21,34,51,32,45,32,17,29,41,52)
> mean(x)
[1] 31.41667
> var(x)
[1] 200.9924

```

Here, `c()`, `mean()` and `var()` are in-built functions of R. The function `c()` assigns those numbers to the object `x`. The commands `mean(x)` and `var(x)` computes the mean and variance of an object `x`. Here, `x` is the input, also called argument, to the function `mean()` and `var()`.

A complete list of in-built functions is available in the document R reference manual. The R reference manual opens by clicking on [Help Manuals \(in PDF\) R reference](#). It opens the full reference manual. It contains a complete list of all the functions and objects in base R. Apart from in-built functions, a large number external functions are available in contributed packages. Contributed packages are nothing but a collection of functions written by the authors of the packages to perform specific analysis. The manual of a package contains the details of the functions provided in that package. To know what are the argument(s) of a function and how to use it, you can use the code `help(functionname)` where `functionname` is the name of the function. This opens an html page in browser containing the details of the function. For example, `help(aov)` gives the details of the usage of the function `aov()`.

4 R packages

Though most of the standard statistical analysis are available in base R, but sometimes some contributed R packages are needed to do some specific analysis. An R package is a bundle

of functions and codes for performing some statistical or mathematical analysis which are generally not covered in base R. This facility of R packages extends the usefulness of R greatly. A large number of packages, as of 09 February 2016 around 7,800 packages are available on CRAN for a variety of analysis and the number is increasing day by day.

4.1 Downloading and installing a package

To use an R package, download the package from CRAN and then install and load it in an R session. A package can be downloaded from within R or from outside R. On a Windows machine which is connected to internet, a package can be installed by clicking on Packages → Install package(s) from the menu bar. This will open a list of mirrors. Select a mirror and then, from the available list of packages in the website, select the desired packages. They will be installed into R.

To keep a copy of the downloaded package, you can visit any CRAN mirror web page and download the package. You can then install it by clicking on ‘Packages’ and then clicking on ‘Install package(s) from local zip files...’ and then select the zip file containing the package. After you have installed a package, you need to load it to R. For this click on ‘Packages’ and then click on ‘Load package...’. Alternatively, you can type `library(packagename)` in the console to load a package where `packagename` is the name of the package. For example, to load a package `agricolae`, you need to type

```
> library(agricolae)
```

A package is to be installed just once, but to use it for analysis, it needs to be loaded every time R is started. To use a package, you should download the manual of the package. The package manual contains documentation on functions which are available in that package. Sometimes more than one package may be needed for analysis. It is always better to load only the required packages. There is no need to load packages which are not required in a session, because loading a number of packages slows down R.

5 Reading data in R

(i) Loading data in R directly:

Data with few variables and few observations can be read in R by typing in the Console R as shown in the following example.

```
> month<-c('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct',
'Nov', 'Dec')
> rainfall<-c(5,4,8,7,9,20,30,35,24,15,10,8)
> mydata=data.frame(month,rainfall)
> mydata
  month rainfall
1   Jan         5
2   Feb         4
```

3	Mar	8
4	Apr	7
5	May	9
6	Jun	20
7	Jul	30
8	Aug	35
9	Sep	24
10	Oct	15
11	Nov	10
12	Dec	8

Note that `data.frame()` function combines the vectors `month` and `rainfall` into a data frame called `mydata`. Note that a dataset in R is always in the form of a two-dimensional array with columns representing variables and rows representing individual observations. Sometimes one may be interested to know the names of variables in a data set loaded in R. For example, to know the names of the variables in data set `mydata`, one can use following command:

```
> names(mydata)
[1] "month" "rainfall"
```

The `scan()` function can also be used to read data directly typed in R console. For example,

```
> y<-scan()
1: 393 55 32 40
5: 2 1 3 5
9:
Read 8 items
> y
[1] 393 55 32 40 2 1 3 5
```

When entering data from keyboard using `scan()` function, one has to hit enter button when one does not want to type any more data. Then R stops scanning and loads the data into the object. The function `scan()` is also able to read data from external file.

(ii) Loading data in R from an external file:

Most often the data may not be just a few observations. There may be quite many variables and observations. In that case, the data may be in a spreadsheet or some other external file, or from some other statistical software or from some web page. R provides facilities for loading data from each of them.

Reading data from text file:

Data in text file should be kept such that the individual observations are separated with a delimiter. Some commonly used delimiters are ‘,’,’:’,‘t’,‘ ’ i.e., blank space, ‘\t’,

'@', '&', '*' etc. But be sure that none of the observations or variables in the data set have any of those characters, otherwise data will be loaded improperly and there may be error in loading of data. Consider a text file with following observations with comma(',') as a delimiter.

```
Jan,5
Feb,4
Mar,8
Apr,7
May,9
Jun,20
Jul,30
Aug,35
Sep,24
Oct,15
Nov,10
Dec,8
```

Let the file name is "rainfall.txt" and is kept in the working directory. This data can be loaded in R by using the function `read.table()` as follows:

```
> mydata2=read.table("rainfall.txt",header=TRUE,sep=",")
> mydata2
  month rainfall
1   Jan         5
2   Feb         4
3   Mar         8
4   Apr         7
5   May         9
6   Jun        20
7   Jul        30
8   Aug        35
9   Sep        24
10  Oct        15
11  Nov        10
12  Dec         8
```

The first argument of `read.table()` refers to the external file. The second argument `header=TRUE` tells R that there is header in the rainfall.txt file, and those are used as variable names for the data. If there is no header in a text file, then `header=FALSE` should be used. Third argument `sep=","` tells R that observations are separated by a ','. There are other arguments to `read.table()` function, but these three are essential. The details of the usage of the function `read.table()` is available with `help(read.table)` in the Console.

There are some other functions to read files with specific delimiters. The function `read.csv()` function loads comma separated value (csv) files, i.e., files with comma delimited observations, `read.csv2()` function loads data from semicolon (;) delimited files, `read.delim()` and `read.delim2()` functions load data from tab delimited files.

(iii) Reading data from a webpage:

Suppose some data is available on a webpage. To read a dataset from a web page the function `read.table()` can be used with the complete address of the page. For example,

```
> webdata=read.table("http://data.princeton.edu/wws509/datasets/
effort.dat")
```

```
> webdata
```

	setting	effort	change
Bolivia	46	0	1
Brazil	74	0	10
Chile	89	16	29
Colombia	77	16	25
CostaRica	84	21	29
Cuba	89	15	40
DominicanRep	68	14	21
Ecuador	70	6	0
ElSalvador	60	13	13
Guatemala	55	9	4
Haiti	35	3	0
Honduras	51	7	7
Jamaica	87	23	21
Mexico	83	4	9
Nicaragua	68	0	7
Panama	84	19	22
Paraguay	74	3	6
Peru	73	0	2
TrinidadTobago	84	15	29
Venezuela	91	7	11

(iv) Loading data from a spreadsheet:

To load data from an excel file to R, the relevant worksheet may be saved into a tab delimited text file or into a csv file and then the text file or .csv may be loaded using `read.table()` or `read.csv()` function. However, if to read the data from excel directly into R, a package called `xlsx` is needed. An example of loading data from excel is shown below.

```
> library(xlsx)
```

```
> read.xlsx("myfile.xlsx", sheetName = "Sheet1")
```

6 Some statistical analysis examples

In this Section, some examples of statistical analysis using R is given. For this purpose, we use the PlantGrowth data set available in R.

```
> data(PlantGrowth)
> PlantGrowth
  weight group
1  4.17  ctrl
2  5.58  ctrl
3  5.18  ctrl
4  6.11  ctrl
5  4.50  ctrl
6  4.61  ctrl
7  5.17  ctrl
8  4.53  ctrl
9  5.33  ctrl
10 5.14  ctrl
11 4.81 trt1
12 4.17 trt1
13 4.41 trt1
14 3.59 trt1
15 5.87 trt1
16 3.83 trt1
17 6.03 trt1
18 4.89 trt1
19 4.32 trt1
20 4.69 trt1
21 6.31 trt2
22 5.12 trt2
23 5.54 trt2
24 5.50 trt2
25 5.37 trt2
26 5.29 trt2
27 4.92 trt2
28 6.15 trt2
29 5.80 trt2
30 5.26 trt2
```

Basic summary of the dataset can be found using `summary()` function.

```
> summary(PlantGrowth)
  weight      group
Min.   :3.590  ctrl:10
1st Qu.:4.550  trt1:10
```



```
Median :5.155   trt2:10
Mean   :5.073
3rd Qu.:5.530
Max.   :6.310
```

Note the `summary()` function has returned mean, median, 1st and 3rd quartiles and minimum and maximum of the weight variable. Since the group variable is factor object, so it has returned number of observations for each level of the group variable. To get the variance of weight variable, the `var()` function can be used.

```
> var(PlantGrowth$weight)
[1] 0.49167
```

To have an idea of distribution of weight for each group in the data, boxplots can be obtained using `boxplot()` function as shown below.

```
> boxplot(weight~group,data=PlantGrowth)
```

To get the mean and standard deviation of weight for each group, the `aggregate()` function can be used as shown below.

```
> aggregate(weight~group,data=PlantGrowth,mean)
  group weight
1  ctrl  5.032
2  trt1  4.661
3  trt2  5.526
```

```
> aggregate(weight~group,data=PlantGrowth,sd)
  group  weight
1  ctrl 0.5830914
2  trt1 0.7936757
3  trt2 0.4425733
```

R has function for performing t-tests. The function `t.test()` is available for this purpose. For example, to test the hypothesis that the population mean of weight variable is 5, the following command can be used.

```
> t.test(PlantGrowth$weight,mu=5)
```

One Sample t-test

```
data: PlantGrowth$weight
t = 0.5702, df = 29, p-value = 0.5729
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
4.811171 5.334829
sample estimates:
mean of x
5.073
```

Two-independent sample t-test can also be performed using `t.test()` function. For example, the following example tests the hypothesis whether the population mean of weights of `trt1` group is equal to the population mean of weights of `trt2` group.

```
> t.test(weight~group,data=PlantGrowth,subset=group!="ctrl",
var.equal=TRUE)
```

Two Sample t-test

```
data: weight by group
t = -3.0101, df = 18, p-value = 0.007518
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-1.4687336 -0.2612664
sample estimates:
mean in group trt1 mean in group trt2
4.661                5.526
```

A paired t-test can be performed with an additional argument `paired=TRUE` in the above function. However that should be done only when the sample observations for the two groups are matched or paired.

Since the above data is from an experiment and the objective was to see whether the average weight of three groups differ significantly or not, an analysis of variance (ANOVA) can be performed. R provides `aov()` function for performing ANOVA for balanced data. If the data is not balanced with respect to the grouping variables, then it is better to use the `lm()` function which fits linear model. The following commands show the uses of both the functions.

```
> aov1=aov(weight~group,data=PlantGrowth)
> summary(aov1)
          Df Sum Sq Mean Sq F value Pr(>F)
group      2  3.766   1.8832   4.846 0.0159 *
Residuals 27 10.492   0.3886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> lm1=lm(weight~group,data=PlantGrowth)
> anova(lm1)
Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value Pr(>F)
group      2  3.7663   1.8832   4.8461 0.01591 *
Residuals 27 10.4921   0.3886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the function `anova()` has been used on the fitted `lm1` object to get ANOVA table. The ANOVA tables are similar from both `aov1` and `lm1` objects. Both these objects contain a number of other terms such as fitted values, residuals, coefficients, degrees of freedoms etc.

Since the `anova` suggests that the groups differ significantly at 5% level with respect to weight, a post hoc test (Tukey's Honest significant difference test) can be performed using `TukeyHSD()` function to compare the groups pair wise.

```
> TukeyHSD(aov1)
Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = weight ~ group, data = PlantGrowth)

$group
      diff      lwr      upr    p adj
trt1-ctrl -0.371 -1.0622161 0.3202161 0.3908711
trt2-ctrl  0.494 -0.1972161 1.1852161 0.1979960
trt2-trt1  0.865  0.1737839 1.5562161 0.0120064
```

The result suggest treatment 1 and treatment 2 groups are significantly different from each other. Note that the `TukeyHSD()` function takes an object of class "aov" as argument. It will not work on an object of class "lm". To get least significant difference (LSD) or to perform other post hoc tests such as Duncan's multiple range test, additional packages need to be used. Some important packages with respect to this book are `agricolae`, `car` and `lsmeans`. LSD can be computed after installing and loading the package in R.

```
> library(agricolae)
> LSD.test(aov1,"group",console=TRUE)
> # Or alternatively
> LSD.test(lm1,"group",console=TRUE)
```

```
Study: aov1 ~ "group"
```

```
LSD t Test for weight
```

```
Mean Square Error: 0.3885959
```

```
group, means and individual ( 95 %) CI
```

	weight	std	r	LCL	UCL	Min	Max
ctrl	5.032	0.5830914	10	4.627526	5.436474	4.17	6.11
trt1	4.661	0.7936757	10	4.256526	5.065474	3.59	6.03
trt2	5.526	0.4425733	10	5.121526	5.930474	4.92	6.31

alpha: 0.05 ; Df Error: 27
Critical Value of t: 2.051831

Least Significant Difference 0.5720126
Means with the same letter are not significantly different.

Groups, Treatments and means		
a	trt2	5.526
ab	ctrl	5.032
b	trt1	4.661

The package `lsmeans` is useful for computing least square means as is done in SAS.
For example

```
> library(lsmeans)
> lsmeans(aov1,"group")
> # or alternatively
> lsmeans(lm1,"group")
group lsmean      SE df lower.CL upper.CL
ctrl   5.032 0.1971284 27 4.627526 5.436474
trt1   4.661 0.1971284 27 4.256526 5.065474
trt2   5.526 0.1971284 27 5.121526 5.930474
```

Confidence level used: 0.95

Pairwise comparisons of treatments is possible using `pairs()` function in `lsmeans` package.

```
> lsm=lsmeans(aov1,"group")
> pairs(lsm)
contrast      estimate      SE      df t.ratio  p.value
ctrl - trt1    0.371 0.2787816 27    1.331   0.3909
ctrl - trt2   -0.494 0.2787816 27   -1.772   0.1980
trt1 - trt2   -0.865 0.2787816 27   -3.103   0.0120
```

P value adjustment: tukey method for a family of 3 means

Often we are interested in compact letter display of the treatments. Under compact letter display, treatments with same letters are not significantly different. This display is particularly useful if number of treatments is more. For the above data, one can have compact letter display using following code.

```
> cld(lsm,Letters="ABCDE")
```

group	lsmean	SE	df	lower.CL	upper.CL	.group
trt1	4.661	0.1971284	27	4.256526	5.065474	A
ctrl	5.032	0.1971284	27	4.627526	5.436474	AB
trt2	5.526	0.1971284	27	5.121526	5.930474	B

Confidence level used: 0.95

P value adjustment: tukey method for a family of 3 means
 significance level used: alpha = 0.05

Often type III sum of squares are desired. The package car is useful for such situations.

```
> library(car)
> Anova(aov1,type="III")
> # or alternatively
> Anova(lm1,type="III")
Anova Table (Type III tests)
```

```
Response: weight
              Sum Sq Df  F value  Pr(>F)
(Intercept) 253.210  1 651.6029 < 2e-16 ***
group         3.766  2   4.8461 0.01591 *
Residuals    10.492 27
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing the significance of contrasts is often required. Both the packages lsmeans and car are useful for this. For example, to test the significance of the contrast $2 \times \text{ctrl} - \text{trt1} - \text{trt2}$, one can use the following commands.

```
> lsm=lsmeans(lm1,"group")
> con1=contrast(lsm, list(con1 = c(2,-1,-1)))
> con1
contrast estimate      SE df t.ratio p.value
con1          -0.123 0.4828639 27  -0.255  0.8009
> lht(lm1,con1@linfct)
Linear hypothesis test
```

```
Hypothesis:
- grouptrt1 - grouptrt2 = 0
```

```
Model 1: restricted model
Model 2: weight ~ group
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	28	10.517				
2	27	10.492	1	0.025215	0.0649	0.8009

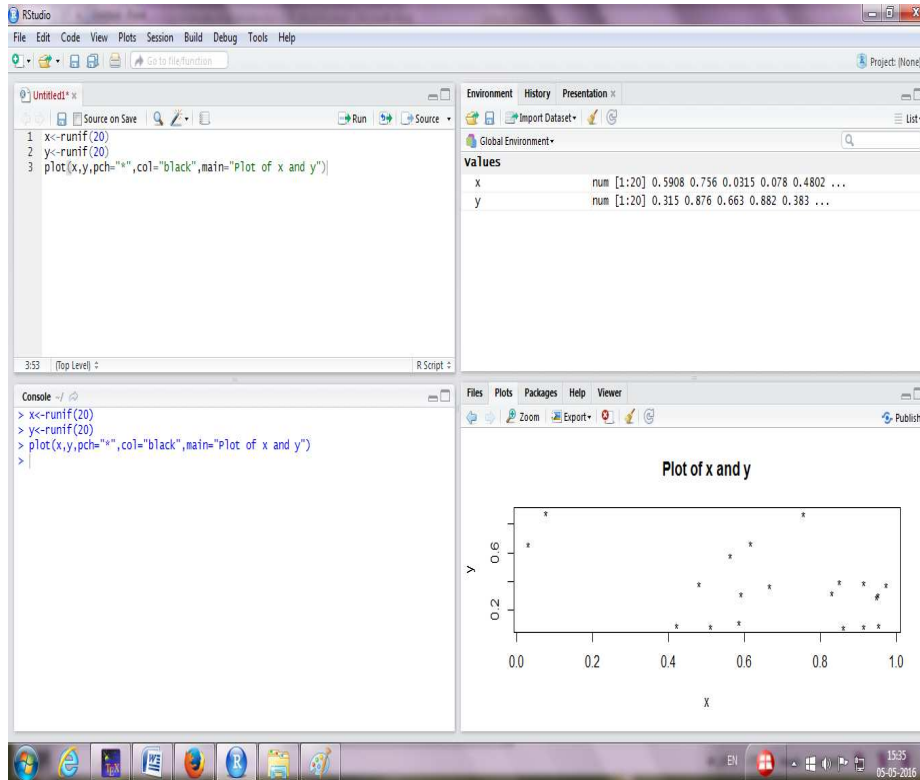


Figure 2: Rstudio Interface

The function `lht()` is for linear hypothesis test and is available in `car` package. All these packages have many other functions. To get details about the available functions in these packages, please refer to the manuals of these packages.

7 RStudio

RStudio is an integrated development environment (IDE) for doing statistical analysis and other tasks using computing power of R software. RStudio is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application; and RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server. RStudio Desktop is available for Microsoft Windows, Mac OS X, and Linux.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux). The free edition of RStudio desktop can be downloaded from <https://www.rstudio.com/>.

RStudio provides more user friendly interface to use R. For using RStudio in machine, base R must be installed in that machine. The interface of RStudio is given in Figure 2.

The top left window in Figure 2 has the editor for writing R codes. Bottom left window is similar to R console where R codes get executed. Top right window of RStudio has two

tabs: Environments and History, respectively. In the Environment tab, we can see which objects are currently loaded in R. The History tab gives the history of commands already executed. Bottom right window of RStudio has several tabs namely Files, Plots, Packages, Help and Viewer. The Files tab can be used to explore different files, Plots tab is used to view plots produced from R codes, Packages tab shows the packages already installed and also allows easy installation and loading of packages in a session. Help tab can be used to see the help on R functions and Viewer tab can be used to see local web content.

An Introduction to STATA software

Anuja A R, Shivaswamy G P, K N Singh, Rajesh T and HarishKumar H V
ICAR- IASRI, New Delhi
anuja.ar@icar.gov.in

STATA is a general-purpose statistical software package. It can be considered a “stat package,” like SAS, SPSS, RATS, or e Views. The software is widely used by researchers in the field of social sciences and bio medicine. The name STATA represents the words Statistics and data. The software can be used in statistical analysis, preparing graphs, simulations and programming.

STATA is a paid licensed software. Licenses can be either annual or perpetual. There are different builds for this software which is presented in Table 1.

Table 1: **STATA – major builds**

Build	Purpose
STATA/IC	Standard version
STATA/MP	Multiprocessor computers (including dual-core and multicore processors)
STATA/SE	Extended version for large databases

The major difference between the versions is the number of variables allowed in memory, which is limited to 2,047 in standard STATA /IC, but can be much larger in STATA/SE or STATA /MP. The number of observations in any version is limited only by memory. Each copy of STATA includes a complete set of manuals (over 6,000 pages) in PDF format, hyperlinked to the on-line help. New versions of Stata is released in roughly every two years. The recent version–‘STATA 16’ was released on June 2019. The newer versions ensure backward compatibility. STATA is capable of holding data very efficiently. The **compress** command, which will check to see whether each variable may be held in fewer bytes than its current allocation. To make most effective use of Stata with large datasets, use a computer with a 64-bit operating system.

USER INTERFACE

Once the Stata window is opened, the screen will look like Figure 1 below.

- **Command box:** The command box is for typing STATA commands

- **Variable list:** Variables of the data set will be available in the box
- **Review box:** The executed commands can be reviewed from the review box
- **Result Box:** Results will be reported in the result box

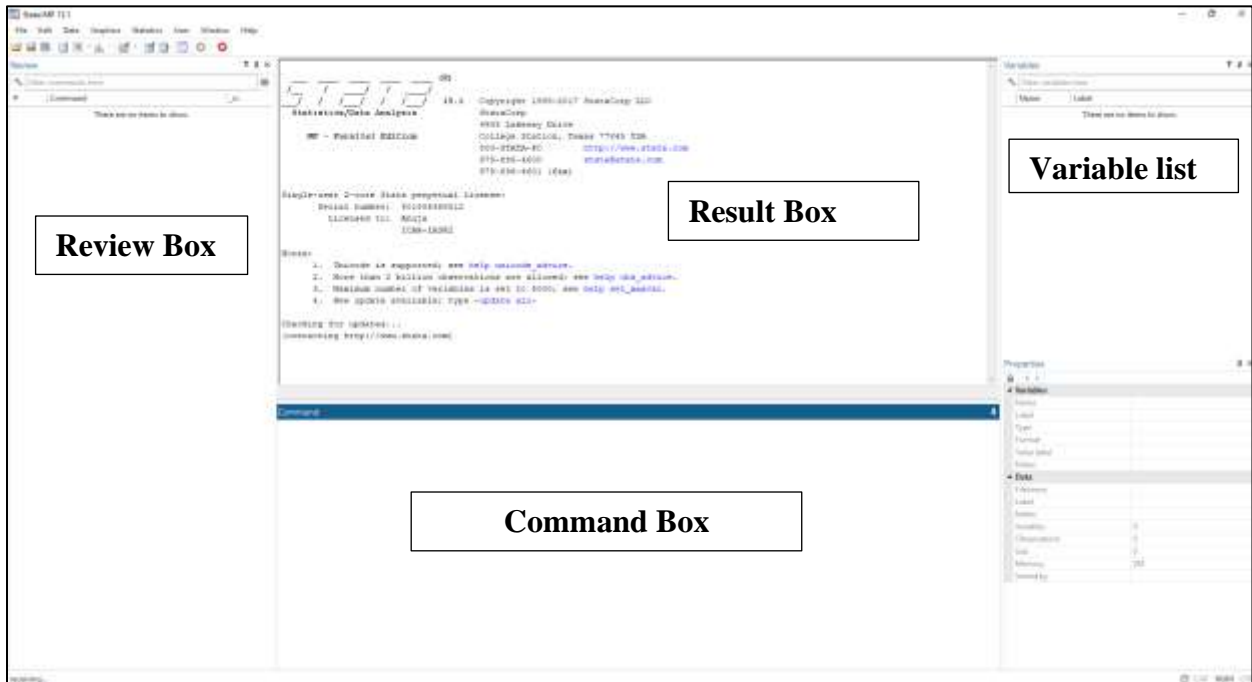


Figure 1: Description of STATA window (STATA Window -Version 15)

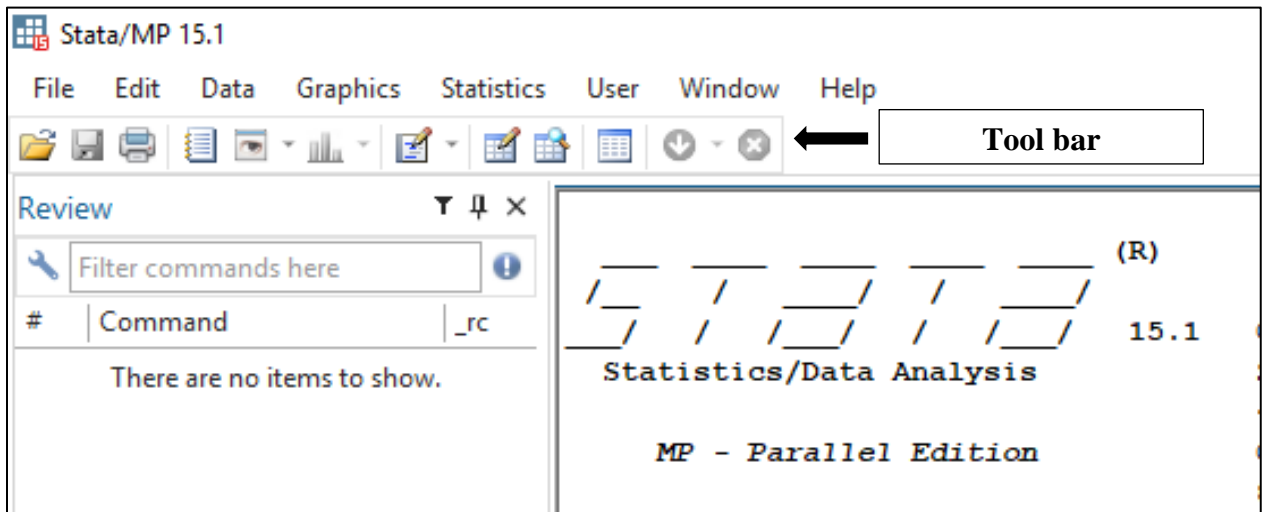


Figure 2: Toolbar of STATA window (STATA Window -Version 15)

A command, to Stata, is a term instructing the program to perform some action. Stata has command-face interface which aids in replicable analyses. The software also provides built in commands using menus and dialogue boxes. Stata permits user-written commands to be directly downloaded from the internet.

Commands can be entered using drop down menu or by typing. Drop down menu can be useful for beginners. The typed command can be executed by pressing enter. There are two methods to enter commands in the command box. First, you can write executable statements, line by line from the command line, and execute the codes. Alternatively, you can write an entire program that contains a group of executable statements, then submit the program from the command line. Once you are familiar with the commands you can use the abbreviations also.

Examples : **h** for **help** , **sum** for **Summarize**

The Toolbar contains icons (Figure 2) - Open and save files, Print results, control Logs, and manipulate windows. Some very important tools allow to open the Do-File Editor, the Data Editor and the Data Browser. The Data Editor and Data Browser present a spreadsheet-like view of the data. Do-File editor helps to construct a file of Stata commands, or “do-file”, and execute it in whole or in part from the editor. A common strategy is to set up a directory for each task in a convenient location in the file system. This can be automated in a do-file with the **cd** command.

The Help Browser, which opens in a Viewer window, provides hyperlinks, in blue, to additional help pages. At the foot of each help screen, there are hyperlinks to the full manuals, which are accessible in PDF format. The links will take you directly to the appropriate page of the manual. You may also search for help at the command line with help command. But what if you don't know the exact command name? Then you may use search or its expanded version, **findit**, each of which may be followed by one or several words. Archived commands in Stata can be accessed using **SSC install** command.

DATA - Format compatibility, Structure and Storage

STATA can import data from formats such as CSV and spreadsheet formats. Stata at a time operates on a single data set and holds it in its virtual memory. The virtual memory has

limitations in holding large datasets. Efficient internal storage helps to overcome this issue. The data set is in rectangular format with all variables holding the same number of observations. In Stata names of data set and variables are **case sensitive** in nature

CREATING SUBDIRECTORY

In STATA, external files are stored in a folder/ sub directory. After constructing a sub directory, the same can be accessed through typing the pathway of command line. Suppose the pathway of a folder is d:\practical\data1, to access the same, type

```
cd d:\practical\data1
```

and press enter. Stata will read the data sets from this folder and save results to this folder.

DATA FILE: Open, Save and Close

Data files of Stata have. **dta** extension. All data files in Stata are read in RAM. The command **set memory**

helps to set RAM. Before opening a data file, memory can be boosted using the command

```
Set memory 80000
```

Using drop down menu, a file can be opened, saved and closed (Figure 4).

Opening a data file

```
File | Open
```

Saving a data file

```
File| Save As.
```

It is advisable to keep a backup copy of the original data set.

Closing a data file

```
File| Exit
```

VIEWING THE DATA

Stata provides two ways to observe the variables and observations in the spreadsheet format - “browse” and “edit”. The **browse** command allows to see the data but not make changes, whereas the **edit** command allows both to browse and to make changes.

To browse, enter browse into the Command window or select the Browse icon (third from the right, a spreadsheet with a magnifying glass on it) (Figure 3). To edit, enter edit into the

Command window or select the Edit icon (fourth from the right, a spreadsheet with no magnifying glass).

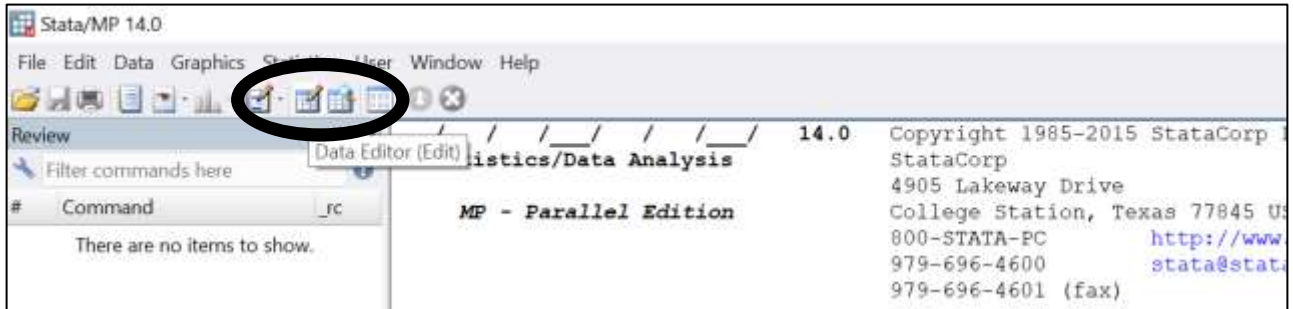


Figure 3: Tools for viewing data in a STATA window (STATA Window -Version 14)

CREATING AND SUBMITTING A DO FILE

Although Stata can be run interactively by just typing one command at a time, Stata commands can also be submitted in batches by using a “do file.” A do file is simply a text file which contains a series of Stata commands. The Stata commands can be written in the same order as you would enter them interactively, and Stata then runs these commands automatically instead of your having to type them in line by line. Anything on a line following a double slash (//) is ignored.

To start creating a do file, click on the Do file editor button (fifth from the right, looks like an envelope with a pencil on it), choose the Do file editor option under the Window menu, or type **doedit** in the Command window. Note that since a do file is a written list of commands as entered in the Command window, you cannot use the Stata menus within a do file. Instead you need to use the typed (Command window) commands.

KEEPING LOG RECORD OF A STATA SESSION

To keep a log of your activity, a log file can be created. Log file contains all the commands and results from the ongoing Stata session that are available in the result section. The log file can be saved either in .log (text file) and .smcl (formatted log file). Command for creating a log file

log using “path of the file”

or using drop down menu

File |log |begin |type file name

The log file can be closed using the command

log close

or using drop down menu

File |log |close

A data set named `practstata.dta` can be loaded using the command **use practstata**. The variables in the given data set can be created or dropped. The changes can be saved using command **save practstata,replace** and press enter. Command **clear** helps to erase the data from current memory.

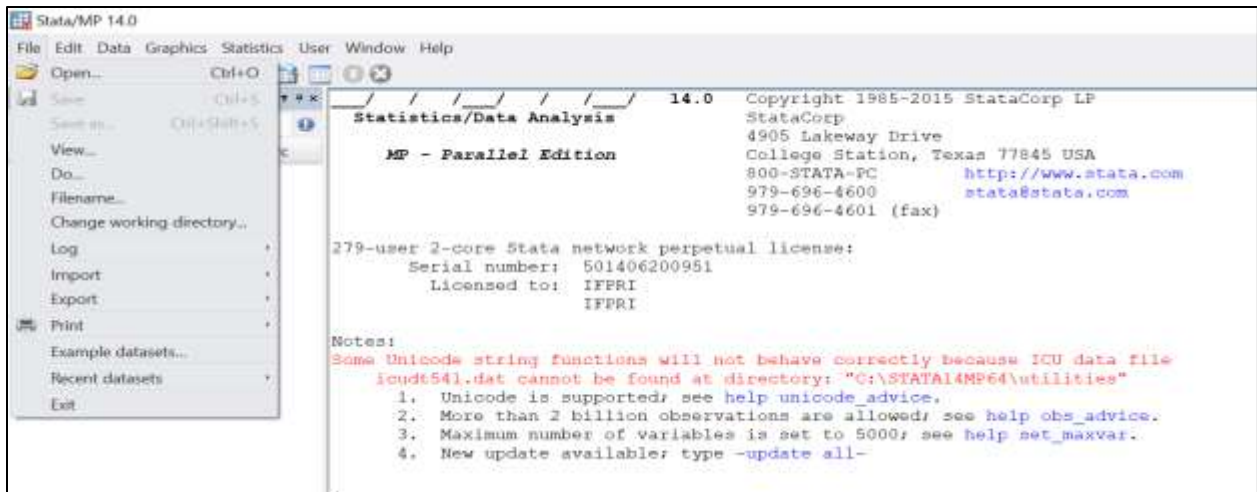


Figure 4: Options under File menu of STATA window (STATA Window -Version 14)

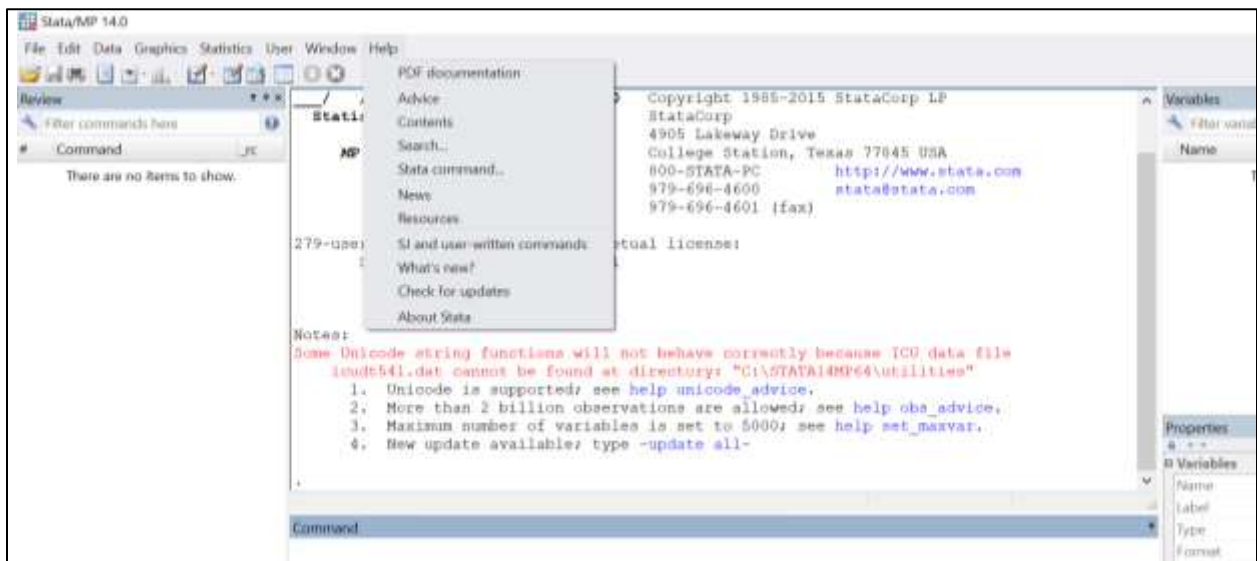


Figure 5: Options under Help menu of STATA window (STATA Window -Version 14)

USING HELP COMMAND

To access Stata's help, you will either select Help from the menus (Figure 5), or use the help and search commands. Regardless of the method you use, results will be shown in the Viewer or Results windows. Blue text indicates a hypertext link, so you can click to go to related entries. In order to ask help, type command

help

For getting information of the contents of data sets, type

help describe.

Command help generates a pop up box that describes the syntax for the 'describe' command. The executed command can be reviewed in review box. By clicking the command in review box, the command can be reused.

WORKING WITH DATA

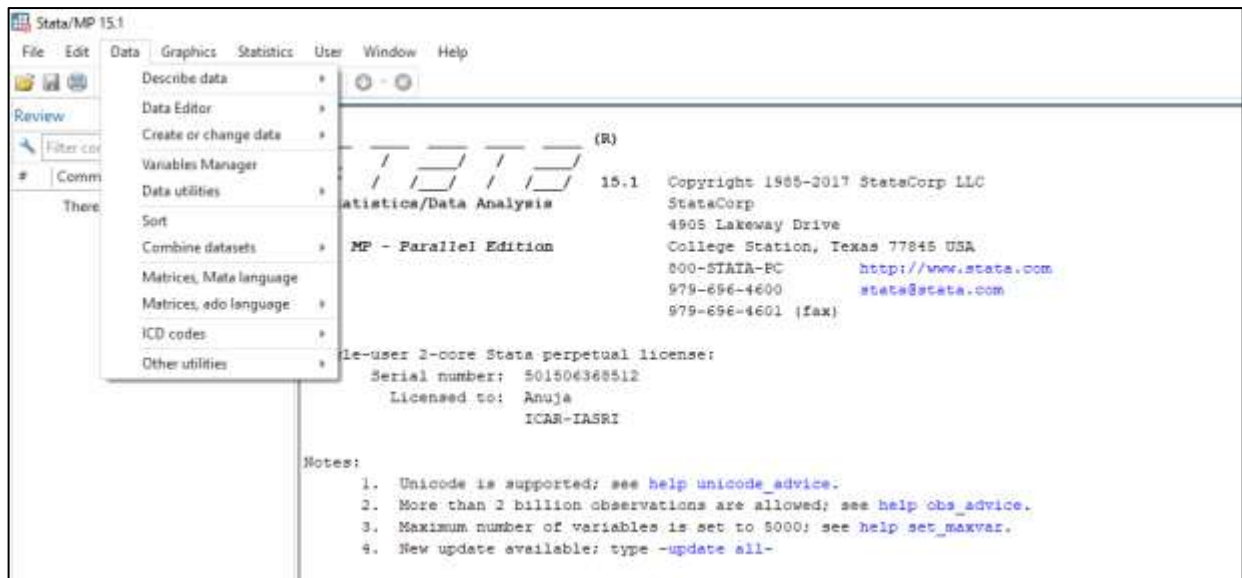


Figure 6: Options under Data menu of STATA window (STATA Window -Version 15)

Figure 6 displays the options available under 'Data' in the drop down menu bar of Stata. The variables of interest can be kept or dropped using command **keep or drop**. Command **codebook** helps to describe data contents. The summary of selected variables in the data set

can be estimated using the command **summarize**. Using this command provides information on the number of observations, mean, standard deviation, minimum and maximum values.

The percentage distribution of categorical variables can be obtained using command **tab** and to get a table combining two categorical variables command **tab var1 var2, col** can be used.

USING OPERATORS IN STATA

Arithmetic operators

+ add

- subtract

* multiply

/ divide

^ raise to the power

Relations operators

= = equal

~= not equal

> Greater than

> = greater than or equal to

< = less than or equal to

Single equal sign (=) is used for assigning a value to a variable.

Logical operators

& and

| or (pipe sign; what you get when you hit Shift and the “\” key)

~ not

CREATING AND CHANGING VALUES OF VARIABLES

A new variable can be generated using the command **generate (abbreviation- gen)**

gen yield= production/area

gen tot_area=(area1+area2)

A new binary variable can be generated using

gen migration=1 if mig==1

replace migration=1 if mig==0

No two variables can have the same name.

LABELING VARIABLES AND VALUES

Labeling variables and values helps to keep track of the coding of the variables and what they represent. Example: Code “1” – Access to extension services. To attach a label to a variable and its values use the command **label variable**.

USING FUNCTIONS

Functions are special calculations used with other commands, such as generate or replace. Stata has the capability to calculate many functions. Command **ln(x)** – calculates the natural log of x, where x may be a constant or a variable such as “income”.

gen logincome = ln(income)

List - Prints all variables and observations to the screen.

codebook - Provides even more information (mean, standard deviation, range, percentiles, labels, number of missing values, etc.) about a variable

ANALYSIS OF CONTINUOUS (SCALE) VARIABLES

tabstat variable

Correlations: Command **corr** price1 price2

Estimating linear models: **regress** yield seed credit marketaccess if state==1

Estimating non-linear models (logit and probit)

logit credit distance kcc bank

probit credit distance kcc bank

To find a predicted probability for each observation based on the most recent model run and save these as a variable called “phat”:

predict phat

MAKING GRAPHS USING STATA

Graphs can be created using drop down menu (Figure 7) or using commands. The options are self-explanatory.

Histogram: `histogram` variable

Scatterplot: `scatter` variable1 variable2

Bar graphs: `graph bar (mean)` variable1

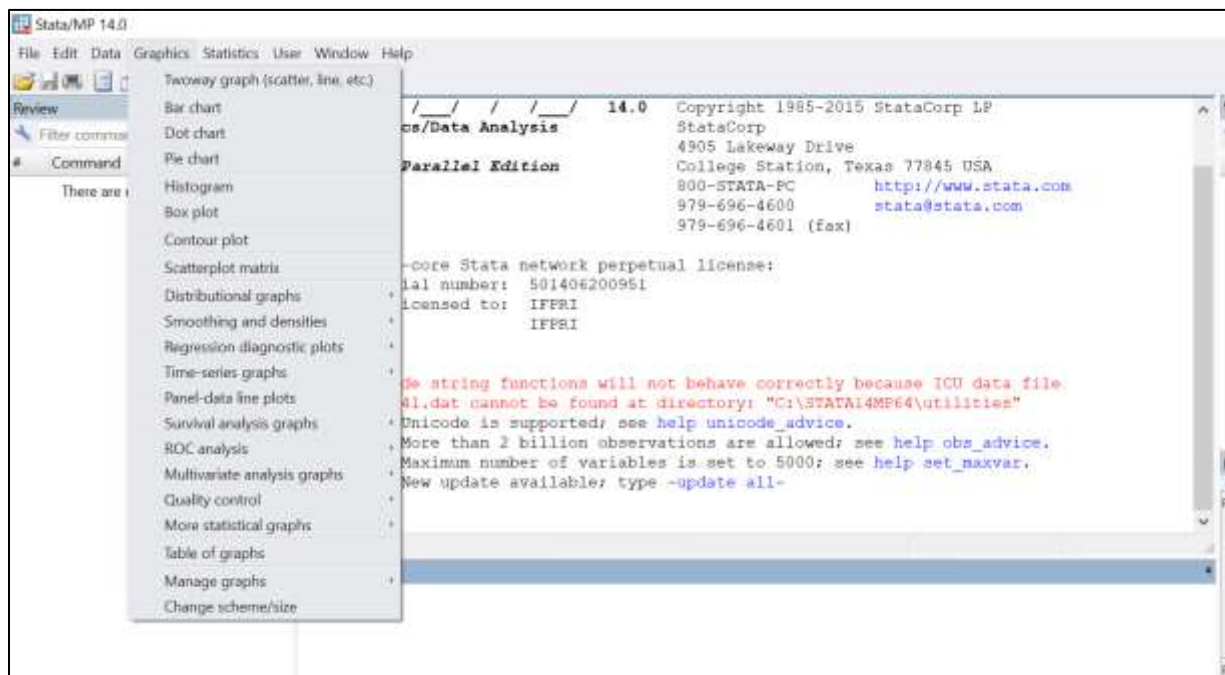


Figure 7: Options under Graphics menu of STATA window (STATA Window -Version 14)

The graphs can be saved using drop down menu. The easiest way to incorporate the graph into a Word document is to copy the graph to the clipboard using Edit | Copy Graph and then paste it into the document.

STATA ADVANTAGES

Stata is an excellent tool for data manipulation. Stata provides all of the standard univariate, bivariate and multivariate statistical tools, from descriptive statistics and t-tests through one-, two- and N-way ANOVA, regression, principal components, and the like. Stata's regression capabilities are full-featured. It has a very powerful set of techniques for the analysis of limited dependent variables: logit, probit, ordered logit and probit, multinomial logit, and the like.

Stata graphics are excellent tools for exploratory data analysis, and can produce high-quality 2-D publication-quality graphics in several dozen different forms.

SOME USEFUL STATA COMMANDS

help	: online help on a specific command
findit	: online references on a keyword or topic
ssc	: access routines from the SSC Archive
log	: log output to an external file
tsset	: define the time indicator for time series or panel data
compress	: economize on space used by variables
pwd	: print the working directory
cd	:change the working directory
clear	:clear memory
exit	:exit the program (,clear if dataset is not saved)

DATA MANIPULATION COMMANDS

generate	: create a new variable
replace	: modify an existing variable
rename	: rename variable
sort	: change the sort order of the dataset
drop	: drop certain variables and/or observations
keep	: keep only certain variables and/or observations
append	: combine datasets by stacking
merge	: merge datasets (one-to-one or match merge)
encode	: generate numeric variable from categorical variable
recode	: recode categorical variable
destring	: convert string variables to numeric
local	: define or modify a local macro (scalar variable)
describe	: describe a data set or current contents of memory
use	: load a Stata data set
save	: write the contents of memory to a Stata data set
collapse	: make a dataset of summary statistics
tab	: abbreviation for tabulate: 1- and 2-way tables

table : tables of summary statistics

STATISTICAL COMMANDS

Figure 8 portrays the statistical commands available under Stata version 15.

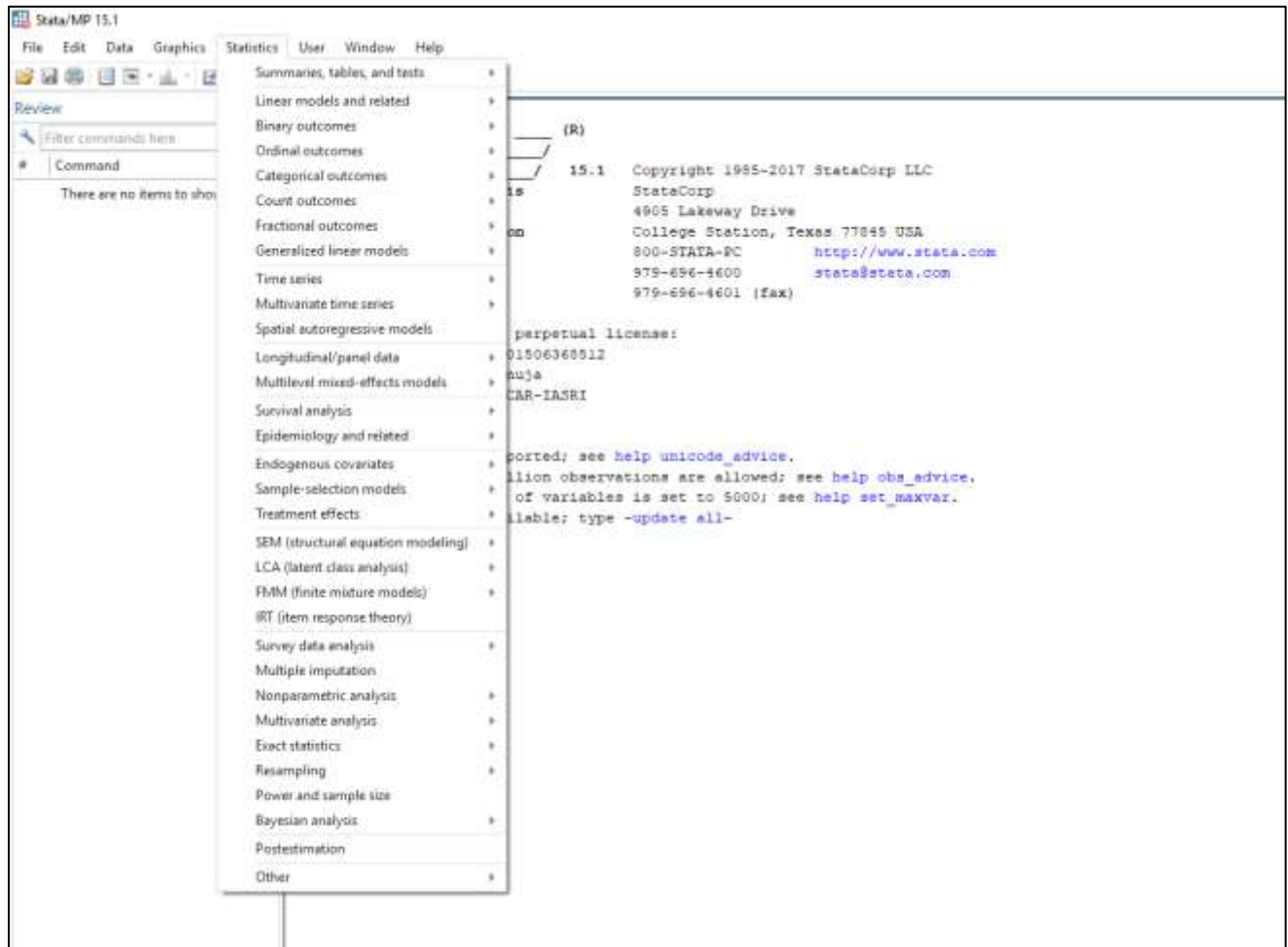


Figure 8: Options under Statistics menu of STATA window (STATA Window -Version 15)

summarize : descriptive statistics

correlate : correlation matrices

ttest : perform 1-, 2-sample and paired t-tests

anova : 1-, 2-, n-way analysis of variance

regress : least squares regression

predict : generate fitted values, residuals, etc.

test : test linear hypotheses on parameters

lincom : linear combinations of parameters
cnsreg : regression with linear constraints
testnl : test nonlinear hypothesis on parameters
margins : marginal effects (elasticities, etc.)
ivregress : instrumental variables regression
prais : regression with AR(1) errors
sureg : seemingly unrelated regressions
reg3 : three-stage least squares
qreg : quantile regression

LIMITED DEPENDENT VARIABLE ESTIMATION COMMANDS

logit, logistic : logit model, logistic regression
probit : binomial probit model
tobit : one- and two-limit Tobit model
cnsreg : Censored normal regression (generalized Tobit)
ologit, oprobit : ordered logit and probit models
mlogit : multinomial logit model
poisson : Poisson regression
heckman : selection model

TIME SERIES ESTIMATION COMMANDS

arima : Box–Jenkins models, regressions with ARMA errors
arfima : Box–Jenkins models with long memory errors
arch : models of autoregressive conditional heteroskedasticity
dfgls : unit root tests
corrgram : correlogram estimation
var : vector autoregressions (basic and structural)
irf : impulse response functions, variance decompositions
vec : vector error–correction models (cointegration)

sspace : state-space models
dfactor : dynamic factor models
ucm : unobserved-components models
rolling : prefix permitting rolling or recursive estimation over subsets

PANEL DATA ESTIMATION COMMANDS

xtreg, fe : fixed effects estimator
xtreg, re : random effects estimator
xtgls : panel-data models using generalized least squares
xtivreg : instrumental variables panel data estimator
xtlogit : panel-data logit models
xtprobit : panel-data probit models
xtpois : panel-data Poisson regression
xtgee : panel-data models using generalized estimating equations
xtmixed : linear mixed (multi-level) models
xtabond : Arellano-Bond dynamic panel data estimator

GRAPHICS COMMANDS:

twoway produces a variety of graphs, depending on options listed

histogram

twoway scatter - scatterplot

twoway line - line plot

tsline - time-series plot

twoway area - area plot

References

Fall, B. E. (2008). Introduction to STATA. Retrieved from <https://www3.nd.edu/~wevans1/ecoe60303/Introduction%20to%20STATA.pdf>

Introductory Guide to using Stata (2009) Retrieved from <https://sites.hks.harvard.edu/fs/pnorris/DPI403%20Fall09/STM103%20Introductory%20Guide%20to%20using%20Stata.pdf>

Baum, C. F. (2011). Introduction to Stata. Boston College. Retrieved from <http://fmwww.bc.edu/GStat/docs/StataIntro.pdf>
<https://www.stata.com/>

Basic Statistical Methods

Pradip Basak
ICAR-IASRI, New Delhi

1.1 Introduction

Statistics is a very broad subject, with applications in a vast number of different fields. Generally, one can say that statistics is the methodology for collecting, analyzing, interpreting and drawing conclusions from the data.

Statistics has been defined differently by the authors from time to time. Some authors define it as Statistical Data i.e. numerical statements of the facts while others define it as Statistical Methods i.e. principles and techniques used in collecting and analyzing the data.

But the Statistics defined as Statistical Data is inadequate because Statistics is not merely confined to the collection of data only but other aspects like presentation, analysis and interpretation etc. are also the parts of Statistics.

The best definition of Statistics was given by Croxton and Cowden according to whom the Statistics is the science which deals with the collection, analysis and interpretation of numerical data/facts.

1.2 Classification

The process of arranging data into homogenous group or classes according to some common characteristics of the data is called Classification.

(1) Qualitative Base:

When the data are classified according to some quality or attributes such as sex, religion, literacy, intelligence etc.

(2) Quantitative Base:

When the data are classified by quantitative characteristics like heights, weights, ages, income etc.

(3) Geographical Base:

When the data are classified by geographical regions or location, like states, provinces, cities, countries etc.

(4) Chronological or Temporal Base:

When the data are classified or arranged by their time of occurrence, such as years, months, weeks, days etc.

1.3 Tabulation of Data

The process of placing classified data into tabular form is known as Tabulation. A table is a symmetric arrangement of statistical data in rows and columns.

1.4 Diagrams and Graphs of Statistical Data

One of the most effective way of representation of statistical data may is through diagrams and graphs. The commonly used diagrams and graphs are as below:

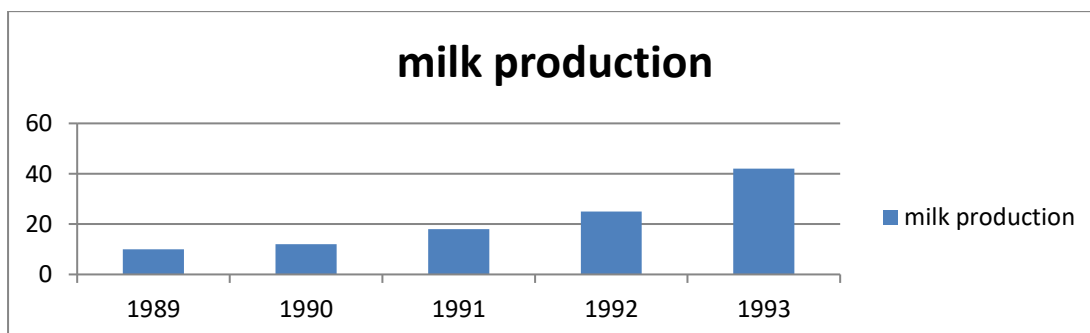
1.5 Types of Diagrams/Charts:

1. Simple Bar Chart
2. Multiple Bar Chart
3. Component Bar Chart or Sub-Divided Bar Chart
4. Simple Component Bar Chart
5. Percentage Component Bar Chart
6. Sub-Divided Rectangular Bar Chart
7. Pie Chart

(1) Simple Bar Chart

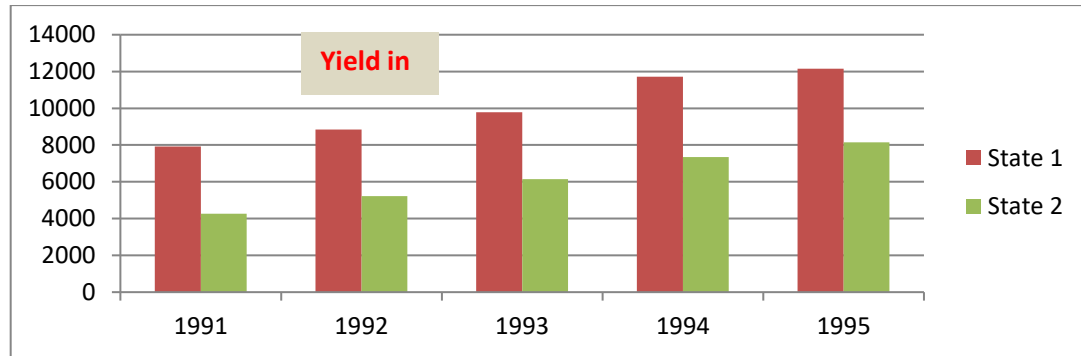
In simple bar chart, we make bars of equal width but variable length, i.e. the magnitude of a quantity is represented by the height or length of the bars. Following steps are undertaken in drawing a simple bar diagram:

Simple Bar Chart showing the Milk Production



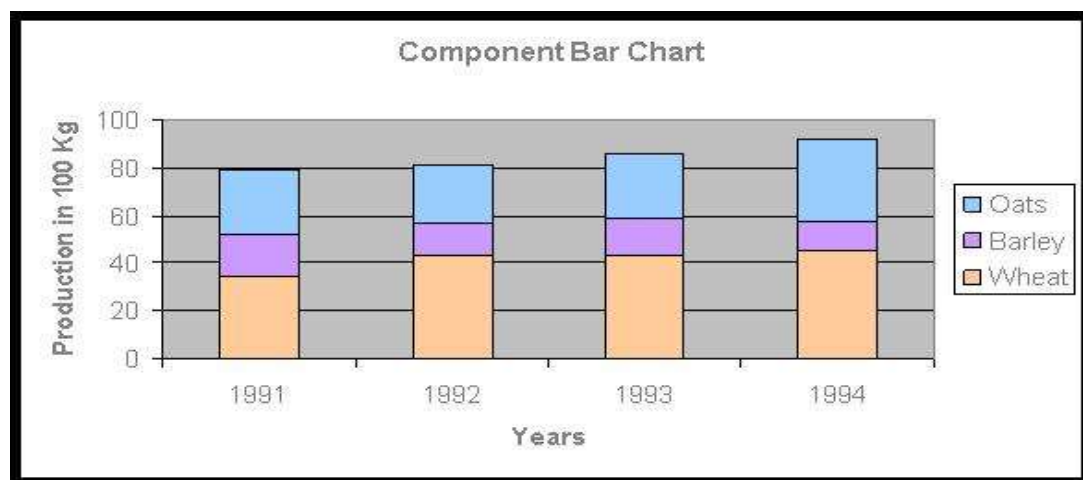
(2) Multiple Bar Charts

By multiple bars diagram, two or more sets of inter-related data are represented.



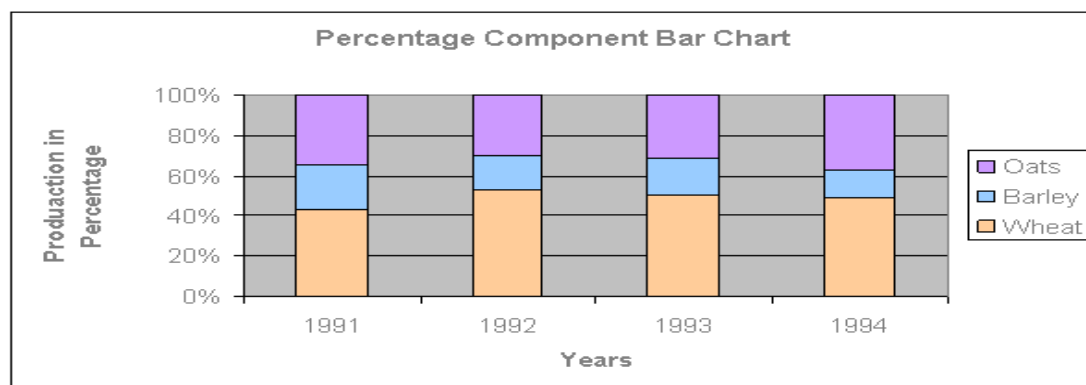
(3) Component Bar Chart

Component bar chart is used to represent data in which the total magnitude is divided into different components.



(4) Percentage Component Bar Chart

Component Bar Charts/Sub-divided Bar Charts may be drawn on percentage basis. To draw sub-divided bar chart on percentage basis, we express each component as the percentage of its respective total.



(5) Pie Chart

Pie chart is used to compare the relation between the whole and its components. To construct a pie chart (sector diagram), we draw a circle with radius (square root of the total). The total angle of the circle is 360° . The angles of each component are calculated by the formula, $Angle\ of\ Sector = \frac{Component\ Part}{Total} \times 360^{\circ}$.

These angles are made in the circle by mean of a protractor to show different components. The arrangement of the sectors is usually anti-clock wise.

Example:

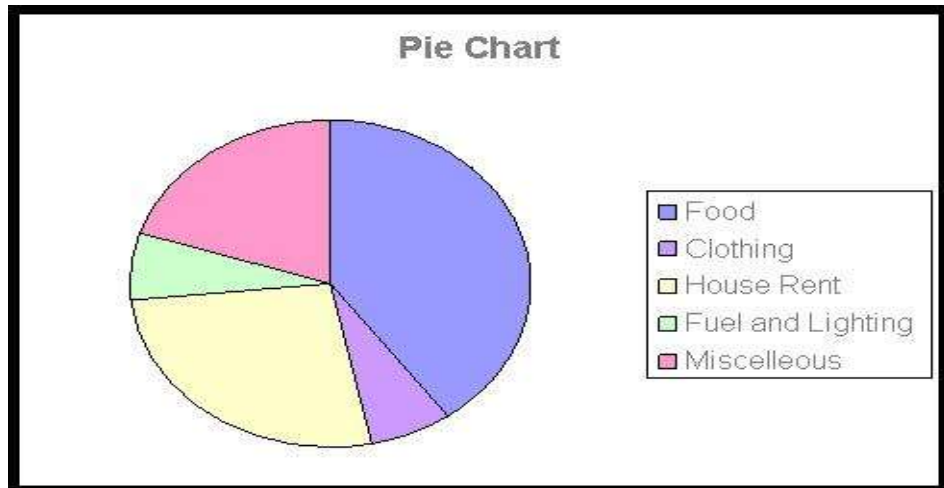
The following table gives the details of monthly budget of a family.

Item of Expenditure	Family Budget
Food	Rs. 600
Clothing	Rs.100
House Rent	Rs.400
Fuel and Lighting	Rs.100
Miscellaneous	Rs.300
Total	Rs.1500

Solution: The necessary computations are given below:

$$Angle\ of\ Sector = \frac{Component\ Part}{Total} \times 360^{\circ}$$

Items	Family Budget		
	Expenditure (Rs.)	Angle of Sectors	Cumulative Angle
Food	600	144°	144°
Clothing	100	24°	168°
House Rent	400	96°	264°
Fuel and Lighting	100	24°	288°
Miscellaneous	300	72°	360°
Total	1500	360°	

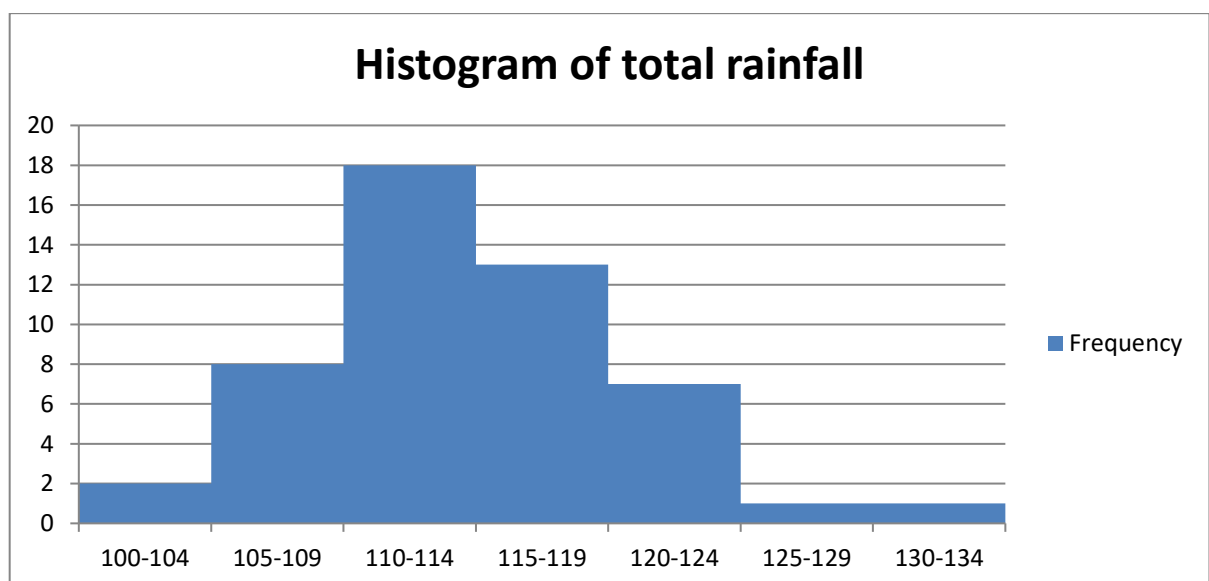


1.6 Most Common Graphs:

1. Histogram,
2. Frequency polygon,
3. Cumulative frequency graph or Ogive.

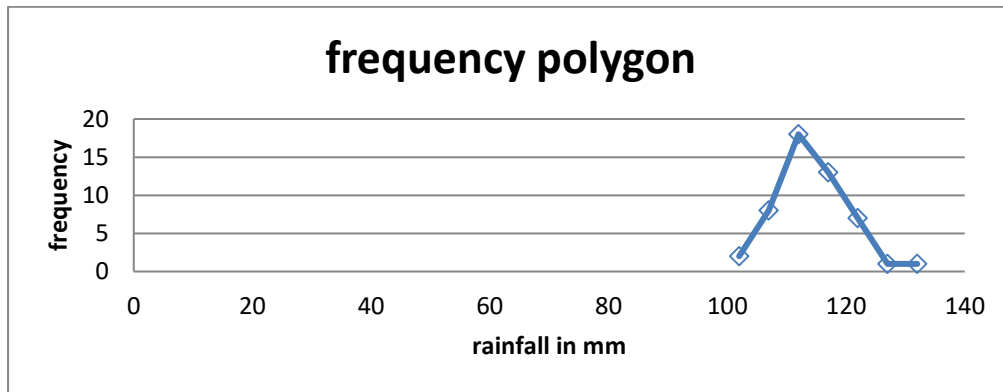
(1) Histogram

The histogram is a graph that uses contiguous vertical bars to display the frequency of the data (unless the frequency equals 0) contained in each class. The heights of the bars equal the frequency (after certain scale has been chosen) and the bases of the bars lie on the corresponding class.



(2) Frequency Polygon

A frequency polygon is a graph that displays the data by using lines that connect points plotted for the frequencies at the midpoints of the classes. In the Cartesian system OXY the midpoints are the first coordinates of the vertices of the polygon and the frequencies are the second coordinates.



(3) Ogive

An OGIVE is a graph that represents the cumulative frequencies for the classes in a frequency distribution. It shows how many of values of the data are below certain boundary.

1.7 Measures of Central Tendency

Measures of central tendency are the statistical constants which enable us to comprehend the whole of the distribution/data in to a single value or it is the value of the variable under study which is representative of the entire distribution.

1.8 Ideal Measures of Central Tendency:

An average possesses all or most of the following qualities (characteristics) is considered a good average:

1. It should be rigidly defined.
2. It should be easy to calculate and easy to understand.
3. It should be based on all the observations.
4. It should be suitable for further mathematical/algebraic treatment.
5. It should not be affected by extreme values.
6. It should be affected at least as possible by the fluctuations of the sample values.

1.9 Types of Measures of Central Tendency:

1. Arithmetic Mean
2. Median
3. Mode
4. Geometric Mean
5. Harmonic Mean

(1) Arithmetic Mean

Arithmetic mean of a variable or set of given observations is quotient of sum of the given observations and the number of the observations.

The arithmetic mean can be computed for both ungroup data (raw data: a data without any statistical treatment) and grouped data (a data arranged in tabular form containing different groups). If x is a variable having n observations, arithmetic mean abbreviated as A M and denoted by \bar{X} can be computed by using any of the following formula;

For ungrouped Data:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$$

For grouped Data:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^k f_i x_i$$

Where

x_i – different observations of the variable under study

f_i – frequencies of different class intervals/groups

n – number of observations and

k - number of classes under group frequency distribution.

(2) Median

Median of a given distribution is the value of the variable which divides the distribution in to two equal parts. It is the value such that number of observations preceding as well as succeeding from the median is equal or which exceeds and exceeded by the same number of observations. Median is thus a **Positional Average** only.

First of all, the given observations of the distribution are arranged in ascending/descending order in case of ungrouped data. Median is calculated as follows;

(i) If number of observations is odd

$$\text{Median} = \text{Value of } \left(\frac{n+1}{2}\right)^{\text{th}} \text{ item}$$

(ii) If the number of observations is even

$$\text{Median} = \text{Average of } \left(\frac{n}{2}\right)^{\text{th}} \text{ and } \left(\frac{n+1}{2}\right)^{\text{th}} \text{ items}$$

Median for Grouped data:

In case of grouped data (discrete frequency distribution), a separate column of cumulative frequencies is made. Find the number $n/2$. See the cumulative frequency in which this number $n/2$ falls. The corresponding x_i value will be the median of the grouped distribution.

In case of the grouped data (continuous frequency distribution), a separate column of cumulative frequencies is also made. Find the number $n/2$. See the cumulative frequency in which this number $n/2$ falls. The corresponding class interval is called the Median Class. After locating the Median Class, following formula is used for calculation of median.

$$\text{Median} = l + \frac{h}{f} \left(\frac{n}{2} - c\right)$$

Where,

l = Lower class limit of the Median Class

f = Frequency of the Median Class

$n = \Sigma f$ = Sum of the frequencies of various class intervals

c = Cumulative frequency of the class preceding the Median Class

h = Class interval size of the Median Class

(3) Mode

Mode is the value which occurs most frequently in the given set of observations i.e. it is the value of the variable which is predominant in the given set of observations. If the data having only one mode the distribution is said to be uni-modal and is said to be bi-modal, if data have two modes.

For ungrouped data, mode is calculated by inspecting the given data. The value which occurs maximum number of times in the distribution is called the Mode of the given distribution.

For grouped data, locate the Modal Class/Group. The class/group which has the maximum frequency is called the Modal Class/Group. After locating the Modal

Class/Group, the following formula is applied for calculation of Mode of the given frequency distribution.

$$Mode = l + \frac{f_m - f_1}{(f_m - f_1) + (f_m - f_2)} \times h$$

Where,

l is the lower class limit of the modal group,

f_m is the frequency of the modal group

f_1 is the frequency of the class interval preceding the modal group

f_2 is the frequency of the class interval preceding the modal group

h is the class interval of the modal group

(4) Geometric Mean

Geometric mean of a set of n observations is the n^{th} root of the multiplication of all the n observations. Hence the geometric mean denoted by G; of n observations $x_i, i = 1, 2, \dots, n$ is given by the formula

$$G = (x_1 \cdot x_2 \cdot \dots \cdot x_n)^{1/n}$$

$$G = \text{Antilog} \left[\frac{1}{n} \sum_{i=1}^n \log x_i \right]$$

In case of grouped frequency distribution, geometric mean is given by the formula

$$G = \text{Antilog} \left[\frac{1}{n} \sum_{i=1}^n f_i \log x_i \right] \text{ where } n = \sum_{i=1}^n f_i$$

The Geometric Mean of the values 10, 5, 15, 8, 12 is given by

$$\begin{aligned} G &= \sqrt[5]{10 \times 5 \times 15 \times 8 \times 12} \\ &= \sqrt[5]{72000} = (72000)^{\frac{1}{5}} = 9.36 \end{aligned}$$

By log method

x	$\log xi$
10	1.0000
5	0.6990
15	1.1761
8	0.9031
12	1.0792
Total	$\Sigma \log xi = 4.8573$

$$G = \text{Antilog} \left(\frac{\Sigma \log xi}{n} \right)$$

$$G = \text{Antilog} \left(\frac{4.8573}{5} \right)$$

$$G = \text{Antilog}(0.9715) = 9.36$$

(5) Harmonic mean

Harmonic mean is defined as the quotient of “number of the given values” and “sum of the reciprocals of the given values”.

Harmonic mean in mathematical terms is defined as follows:

For ungrouped data:
$$HM = \frac{n}{\Sigma\left(\frac{1}{x}\right)}$$

For grouped data:
$$HM = \frac{\Sigma f}{\Sigma\left(\frac{f}{x}\right)}$$

The Harmonic Mean of the numbers: 13.5, 14.5, 14.8, 15.2 and 16.1 is given by

x	$\frac{1}{x}$
13.2	0.0758
14.2	0.0704
14.8	0.0676
15.2	0.0658
16.1	0.0621
Total	$\Sigma\left(\frac{1}{x}\right) = 0.3417$

$$HM = \frac{5}{0.3417} = 14.63$$

1.10 Measures of Dispersion

Measures of central tendency give us single figure which represent the entire distribution or set of observations or around which the observations of the set of data concentrated. But they are inadequate to give us the complete idea of the distribution because they do not tell us the extent to which the observations of the distribution vary from the central value. There may be more than one distributions having the same central value but there may be the wide variation in the different observations of the distribution. The observation may be close to the central value or they may be spread away from the central value. If the observations are close to the central value, we say that dispersion or variation is small. If the observations are spread away from the central value, we say dispersion is more.

Suppose we have three groups of students who have obtained the following marks in a test. The arithmetic means of the three groups are also given below;

Group A: 46, 48, 50, 52, 54 $\bar{X}_A = 50$

Group B: 30, 40, 50, 60, 70 $\bar{X}_B = 50$

Group C: 10, 30, 50, 70, 90 $\bar{X}_C = 50$

All the three sets of observations have the same arithmetic mean i.e. 50. But we see that the variation/dispersion of the other values to the central value is less in Group A in comparison of group B and Group C or we may also say that the variation/dispersion in the observations are more in Group C in comparison of the other two groups.

Thus in order to give a proper idea about the overall nature of the given values of a distribution or set of data, it is necessary to state how are the values of the distribution scattered/dispersed from the measures of central tendency? Therefore, the measures of dispersion may be defined as a statistics signifying the extent of the variations of items of the given set of observations around the measure of central tendency.

For the study of dispersion, there are some measures which show whether the dispersion is small or large. There are two types of measure of dispersion;

- (a) Absolute Measure of Dispersion
- (b) Relative Measure of Dispersion

(a) Absolute Measures of Dispersion:

These measures give us an idea about the amount of dispersion in a set of observations.

- 1. Range
- 2. Quartile Deviation or Semi Inter Quartile Range
- 3. Mean Deviation
- 4. Variance and Standard deviation

(b) Relative Measure of Dispersion:

These measures are calculated for the comparison of dispersion in two or more than two sets of observations. These measures are free of the units in which the original data is measured. The relative measures of dispersion are:

- 1. Coefficient of Range
- 2. Coefficient of Quartile Deviation
- 3. Coefficient of Mean Deviation
- 4. Coefficient of Variation

1.11 Range

Range is defined as the difference between the maximum and the minimum values of the given observations. If x_m denotes the maximum value and x_0 denotes the minimum value, range is defined as:

$$\text{Range} = x_m - x_0$$

$$\text{Coefficient of Range} = \frac{x_m - x_0}{x_m + x_0}$$

1.12 Quartile Deviation/Semi Inter Quartile Range

It is based on the lower quartile Q_1 and the upper quartile Q_3 .

$$\text{Quartile Deviation} = \frac{Q_3 - Q_1}{2}$$

$$\text{Coefficient of Quartile Deviation} = \frac{Q_3 - Q_1}{Q_3 + Q_1}$$

1.13 Mean Deviation

The mean deviation is defined as the arithmetic mean of the absolute deviations of all the values taken from some suitable average which may be the arithmetic mean, the median or the mode.

The mean deviation of a set of sample data in which the suitable average (AM) is \bar{X} , is given by the relation:

$$\text{Mean Deviation} = \frac{\sum |X - \bar{X}|}{n}$$

For frequency distribution

$$\text{Mean Deviation} = \frac{\sum f |X - \bar{X}|}{\sum f}$$

Mean deviation is a better measure of dispersion than Range and Quartile Deviation.

1.14 Coefficient Of Mean Deviation

Coefficient of Mean Deviation is given by

$$\text{Coefficient of Mean Deviation} = \frac{\text{Mean deviation}}{\text{AM}}$$

1.15 Variance and Standard Deviation

The standard deviation is defined as the positive square root of the mean of the squares of all the deviations taken from arithmetic mean of the data. The standard deviation is denoted by σ and is given by

Population Standard Deviation is given as

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

Sample Standard Deviation is given as

$$s = \sqrt{\frac{1}{N - 1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

The unit of standard deviation is same as the units of the original observations.

The Variance is the square of the standard deviation. The standard deviation plays a dominating role for the study of variation in the data. It is widely used for the analysis of measure of dispersion.

As far as the important statistical tools are concerned, the first important tool is the arithmetic mean \bar{X} and the second important tool is the standard deviation. Both are based on all the observations and are subject to mathematical treatment.

1.16 Coefficient of Standard Deviation and Coefficient of Variation

The standard deviation is the absolute measure of dispersion. Its relative measure is called standard coefficient of dispersion or coefficient of standard deviation. It is given

$$\text{Coefficient of Standard Deviation} = \frac{\sigma}{\bar{X}}$$

The coefficient of variation (CV) is given by the formula

$$\text{Coefficient of Variation} = \frac{\sigma}{\bar{X}} \times 100$$

Coefficient of variation is a pure number and the unit of observations cannot be mentioned with its value. It is written in percentage. When its value is 20%, it means that when the mean of the observations is assumed equal to 100, their standard deviation will be 20.

Coefficient of variation is used to compare the degree of dispersion/variation in different sets of data particularly the data which differ in their means or differ in the units of measurement. The wages of workers may be in dollars and the consumption of meat in their families may be in kilograms. The standard deviation of wages in dollars cannot be compared with the standard deviation of the quantity of meat in kilograms. Both the standard deviations need to be converted into coefficient of variation for comparison. Suppose the value of coefficient of variation of wages is 10% and the value of coefficient of variation of meat is 25%. This means that the wages of workers are consistent in comparison of their consumption of meat. We say that there is greater variation in their consumption of meat. The observations about the quantity of meat are more dispersed than their wages.

Classification and Regression Trees (CART)

**Ramasubramanian V.
ICAR-IASRI, New Delhi
r.subramanian@icar.gov.in**

1. Introduction

In certain research studies, development of a reliable decision rule, which can be used to classify new observations into some predefined categories, plays an important role. The existing traditional statistical methods are inappropriate to use in certain specific situations, or of limited utility, in addressing these types of classification problems. There are a number of reasons for these difficulties. First, there are generally many possible “predictor” variables which makes the task of variable selection difficult. Traditional statistical methods are poorly suited for this sort of multiple comparisons. Second, the predictor variables are rarely nicely distributed. Many variables (in agriculture and other real life situations) are not normally distributed and different groups of subjects may have markedly different degrees of variation or variance. Third, complex interactions or patterns may exist in the data. For example, the value of one variable (e.g., age) may substantially affect the importance of another variable (e.g., weight). These types of interactions are generally difficult to model and virtually impossible to model when the number of interactions and variables becomes substantial. Fourth, the results of traditional methods may be difficult to use. For example, a multivariate logistic regression model yields a probability for different classes of the dependent variable, which can be calculated using the regression coefficients and the values of the explanatory variable. But practitioners generally do not think in terms of probability but, rather in terms of categories, such as “presence” versus “absence”. Regardless of the statistical methodology being used, the creation of a decision rule requires a relatively large dataset.

In recent times, there has been increasing interest in the use of Classification And Regression Tree (CART) analysis. CART analysis is a tree-building technique which is different from traditional data analysis methods. In a number of studies, CART has been found to be quite effective for creating decision rules which perform aswell or better than rules developed using more traditional methods aiding development of DSS (Decision

Support Systems). In addition, CART is often able to uncover complex interactions between predictors which may be difficult or impossible using traditional multivariate techniques. It is now possible to perform a CART analysis with a simple understanding of each of the multiple steps involved in its procedure.

Classification tree methods such as CART are convenient way to produce a prediction rule from a set of observations described in terms of a vector of features and a response value. The aim is to define a general prediction rule which can be used to assign a response value to the cases solely on the bases of their predictor (explanatory) variables. Tree-structured classifications are not based on assumptions of normality and user-specified model statements, as are some conventional methods such as discriminant analysis and ordinary least square regression.

Tree based classification and regression procedure have greatly increased in popularity during the recent years. Tree based decision methods are statistical systems that mine data to predict or classify future observations based on a set of decision rules and are sometimes called rule induction methods because the reasoning process behind them is clearly evident when browsing the trees. The CART methodology have found favour among researchers for application in several areas such as agriculture, medicine, forestry, natural resources management etc. as alternatives to the conventional approaches such as discriminant function method, multiple linear regression, logistic regression etc. In CART, the observations are successively separated into two subsets based on associated variables significantly related to the response variable; this approach has an advantage of providing easily comprehensible decision strategies. CART can be applied either as a classification tree or as a regressive tree depending on whether the response variable is categorical or continuous. Tree based methods are not based on any stringent assumptions. These methods can handle large number of variables, are resistant to outliers, non-parametric, more versatile, can handle categorical variables, though computationally more intensive. They can be applied to data sets having both a large number of cases and a large number of variables, and are extremely robust to outliers. These are not based on assumptions such as normality and user-specified model statements, as are some conventional methods such as discriminant analysis or ordinary least square (OLS) regression. Yet, unlike the case for other

nonparametric methods for classification and regression, such as kernel-based methods and nearest neighbor methods, the resulting tree-structured predictors can be relatively simple functions of the predictor variables which are easy to use.

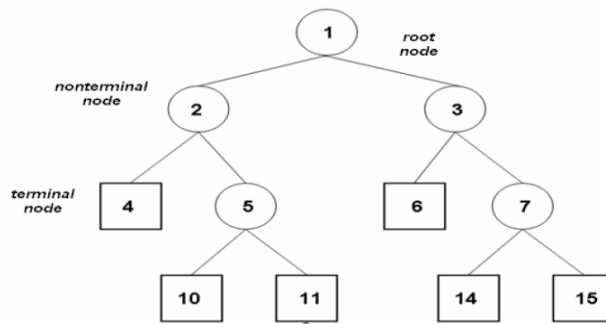
CART can be a good choice for the analysts as they give fairly accurate results quickly, than traditional methods. If more conventional methods are called for, trees can still be helpful if there are a lot of variables, as they can be used to identify important variables and interactions. These are also invariant to the monotonic transformations of the explanatory variables and do not require the selection of the variable in advance as in regression analysis.

Agriculture being a highly uncertain occupation, classification and prediction in the field of agriculture aid planners to take proactive measures. Keeping in view the requirements to develop a sound classificatory system and that the potentials of the tree based methods for this purpose has not fully been explored, it will be of interest to employ these methodologies upon a suitable data set in the field of agriculture. More importantly, since the real world data often does not satisfy the usual assumptions like that of normality, homoscedasticity etc it can be taken up as a motivation to find such a classificatory rule where assumptions of such rules fail. Apart from all these, tree-based methods are one among the promising data mining tools that provide easily comprehensible decision strategy.

Tree based applications originated in the 1960s with the development of AID (Automatic Interaction Detector) by Morgan and Sonquist in the 1960s as regression trees. Further modifications in this technique was carried out to result in THAID (THeta AID) by Morgan and Messenger (1973) to produce classification trees and CHAID (CHi AID) by Kass in the late 1970s. Breiman *et al.* (1984) developed CART (Classification and Regression Trees) which is a sophisticated program for fitting trees to data. Breiman, again in 1994, developed the bagging predictors which is a method of generating multiple versions of a predictor and using them to get an aggregated predictor. A good account of the CART methodology can be found in many recent books, say, Izenman (2008). An application of classification trees in the field of agriculture can be found in Sadhu *et al.* (2014).

2. CART methodology

The conventional CART methodology is outlined briefly. Following is a schematic representation of a conventional CART tree structure:



The unique starting point of, say, a classification tree, is called a root node and consists of the entire learning set \mathcal{L} at the top of the tree. A node is a subset of the set of variables, and it can be terminal or nonterminal node. A nonterminal (or parent) node is a node that splits into two left and right child nodes (binary split). Such a binary split is determined by a condition on the value of a single variable, where the condition is either satisfied or not satisfied by the observed value of that variable. All observations in \mathcal{L} that have reached a particular (parent) node and satisfy the condition for that variable drop down to one of the two *child* nodes; the remaining observations at that (parent) node that do not satisfy the condition drop down to the other *child* node. A node that does not split is called a terminal node and is assigned a class label. Each observation in \mathcal{L} falls into one of the terminal nodes. When an observation of unknown class is “dropped down” the tree and ends up at a terminal node, it is assigned the class corresponding to the class label attached to that node. There may be more than one terminal node with the same class label. To produce a tree-structured model using recursive binary partitioning, CART determines the best split of the learning set \mathcal{L} to start with and thereafter the best splits of its subsets on the basis of various issues such as identifying which variable should be used to create the split, and determining the precise rule for the split, determining when a node of the tree is a terminal one, and assigning a predicted class to each terminal node. The assignment of predicted classes to the terminal nodes is relatively simple, as is determining how to make the splits, whereas determining the right-sized tree is not so straightforward. After growing a fully expanded tree, a tree of optimum size is obtained. In a particular type of tree building called ‘exhaustive search’, at each stage of recursive partitioning, all of the allowable ways of

splitting a subset of \mathcal{L} are considered, and the one which leads to the greatest increase in node purity is chosen. This can be accomplished using what is called an “impurity function”, which is nothing but a function of the proportion of the learning sample belonging to the possible classes of the response variable. To choose the best split over all variables, first the best split for a given variable has to be determined. To assess the goodness of a potential split, the value of the ‘impurity function’ such as Gini diversity index and the Entropy function can be calculated using the cases in the learning sample corresponding to the parent node, and subtract from this the weighted average of the impurity for the two *child* nodes, with the weights proportional to the number of cases of the learning sample corresponding to each of the *child* nodes, to get the decrease in the overall impurity that would result from the split. To select the way to split a subset of \mathcal{L} in the tree growing procedure, all allowable ways of splitting can be considered, and the one which will result in the greatest decrease in node impurity (or, in other words, greatest increase in the node purity) can be chosen.

In order to grow a tree, the starting point is the root node, which consists of the learning set \mathcal{L} . Using the “goodness of split” criterion for a single variable, the tree algorithm finds the best split at the root node for each of the variables. The best split s at the root node is then defined as the one that has the largest value of this goodness of split criterion over all single-variable best splits at that node. Next is to split each of the *child* nodes of the root node in the same way. The above computations are repeated for each of the *child* nodes except that this time only the observations in that specific *child* node are considered for the calculations rather than all the observations. When these splits are completed, the splitting is continued with the subsequent nodes. This sequential splitting procedure of building a tree layer-by-layer is hence called recursive partitioning. If every parent node splits in two *child* nodes, the result is called a binary tree. If the binary tree is grown until none of the nodes can be split any further, then the tree is said to be saturated. Usually, first a very large tree is grown, splitting subsets in the current partition of \mathcal{L} even if a split does not lead to an appreciable decrease in impurity. Then a sequence of smaller trees can be created by “pruning” the initial large tree, where in the pruning process, splits that were made are removed and a tree having a fewer number of nodes is produced. The crucial part of creating a good tree-structured classification model is determining how complex the tree should be. If

nodes continue to be created until no two distinct values of the independent variables for the cases in the learning sample belong to the same node, the tree may be overfitting the learning sample and not be a good classifier of future cases. On the other hand, if a tree has only a few terminal nodes, then it may be that it is not making enough use of information in the learning sample, and classification accuracy for future cases will suffer. Initially, in the tree-growing procedure, the predictive accuracy typically increases as more nodes are created and the partition gets finer. But it is usually seen that at some point the misclassification rate for future cases will start to get worse as the tree becomes more complex. In order to compare the prediction accuracy of various tree-structured models, there needs to be a way to estimate a given tree's misclassification rate for the future observations, a measure named 'resubstitution estimate' of the misclassification rate is obtained by using the tree to classify the members of the learning sample (that were used to create the tree), and observing the proportion that are misclassified. More often, a better estimate of a tree's misclassification rate can be obtained using an independent "test set", which is a collection of cases coming from the same population or distribution as the learning set. Like the learning set, for the test set the true class for each case is known in addition to the values for the predictor variables. The test set estimate of the misclassification rate is just the proportion of the test set cases that are misclassified when predicted classes are obtained using the tree created from the learning set. The learning set and the test set are both composed of cases for which the true class is known in addition to the values for the predictor variables. Generally, about one third of the available cases should be set aside to serve as a test set, and the rest of the cases should be used as learning set. But sometimes a smaller fraction, such as one tenth, is also used and then resorting to 10-fold cross validation. A specific way to create a useful sequence of different-sized trees is to use "minimum cost-complexity pruning". In this process, a nested sequence of subtrees of the initial large tree is created by "weakest-link cutting". With weakest-link cutting (pruning), all of the nodes that arise from a specific nonterminal node are pruned off (leaving that specific node itself as terminal node), and the specific node selected is the one for which the corresponding pruned nodes provide the smallest per node decrease in the resubstitution misclassification rate. If two or more choices for a cut in the pruning process would produce the same per node decrease in the resubstitution misclassification rate, then

pruning off the largest number of nodes is preferred. The sequence of subtrees produced by the pruning procedure serves as the set of candidate subtrees for the model, and to obtain the classification tree, all that remains to be done is to select the one which will hopefully have the smallest misclassification rate for future observations. The selection of final tree is based on estimated misclassification rates, obtained using a test set or by cross validation.

References:

Breiman, L., Freidman, J.H., Olshen, R.A. and Stone, C.J. (1984). *Classification and regression trees*. Wadsworth, Belmont CA.

Izenman, A.J. (2008). *Modern multivariate statistical techniques: Regression, classification and manifold learning*. Springer, New York.

Morgan, J.N. and Messenger, R.C. (1973). THAID: a sequential search program for the analysis of nominal scale dependent variables. Institute for Social Research, University of Michigan, Ann Arbor, MI.

Sadhu, S.K., Ramasubramanian, V., Rai, A. and Kumar, A. (2014). Decision tree based models for classification in agricultural ergonomics, *Statistics and Applications*, 12(1&2), 21-33.

Self-Organizing Maps (SOM)

**Ramasubramanian V.
ICAR-IASRI, New Delhi
r.subramanian@icar.gov.in**

Various approaches are available for classification and prediction. One of such techniques is the Self-Organizing Feature Maps (SOFMs) also known as Kohonen neural networks which comes under the category of unsupervised learning. Rather than just coming under the purview of data analysis, these techniques are being increasingly recognized for data mining purposes when viewed in the context of what are called Very Large Data-Bases (VLDBs). A practical treatment on SOFM based Kohonen networks can be found in Haykin (1996).

Classification methods include the conventional clustering methods (e.g. K-means), discriminant function method and SOFM while predictive models include decision trees (e.g., CART - Classification And Regression Trees), neural networks (the most popular type of architectures being MLP – MultiLayer Perceptron) and statistical models (e.g. MLR - Multiple Linear Regression, Logistic regression etc.). Decision trees are nothing but classification systems that predict or classify future observations based on a set of decision rules and are sometimes called rule induction methods because the reasoning process behind them is clearly evident when browsing the trees. Neural network models are used when the underlying relationship between the different variables in the system are unknown (which are complex and typically non-linear). For completeness, the statistical models include both linear and non-linear regression approaches assuming explicit functional forms.

The very first thing to be aware of while employing any classification method or prediction model is of ascertaining whether the nature of the problem requires a ‘supervised’ or an ‘unsupervised’ approach. The supervised problem occurs when there is a known membership class or output associated with each input in the ‘training’ data set i.e. the set upon which the method or model will be fitted or employed. The unsupervised problem means that one deals with a set of data which have no specific associated classes or outputs attached.

The Kohonen architecture of neural networks is a special type of architecture and is totally different from other types and solely meant for classification rather than prediction. Kohonen network offers a considerably different approach to ANNs and are designed primarily for unsupervised learning rather than for supervised problems. Here, the training data set contains only input variables and no outputs. It is a 'self-organizing' system, which automatically adapts itself in such a way that similar input objects are associated with the topological close neurons in the ANN. The phrase 'topological close neurons' means that neurons that are physically located close to each other will react similar to similar inputs, while the neurons that are far apart in the lay-out of the ANN will react quite different to similar inputs.

The principal goal is to transform an incoming input pattern of arbitrary dimension into a two-dimensional discrete map. Neurons in the network are arranged in a two-dimensional grid and there happens a competition among these neurons to represent the input pattern. The 'winning' neurons and the similar pattern neurons i.e. the neighboring neurons are placed in contiguous locations in output space. The neurons learn to pin-point the location of the neuron in the ANN that is most 'similar' to the input vector. Here, the phrase 'location of the most similar neuron' has to be taken in a very broad sense. It can mean the location of the closest neuron with the smallest or with the largest Euclidean distance to the input vector, or it can mean the neuron with the largest output in the entire network for this particular input vector etc. In other words, in the Kohonen network, a 'rule' deciding which of all neurons will be selected after the input vector enters the ANN is mandatory. During the training in the Kohonen's ANN, the multidimensional neurons self-organise themselves in the two-dimensional plane in such a way that the objects from the multidimensional measurement space are mapped into the plane of neurons with respect to some internal property correlated to the m-dimensional measurement space of objects.

The correction of weights is carried out after the input of each input object in the following four steps:

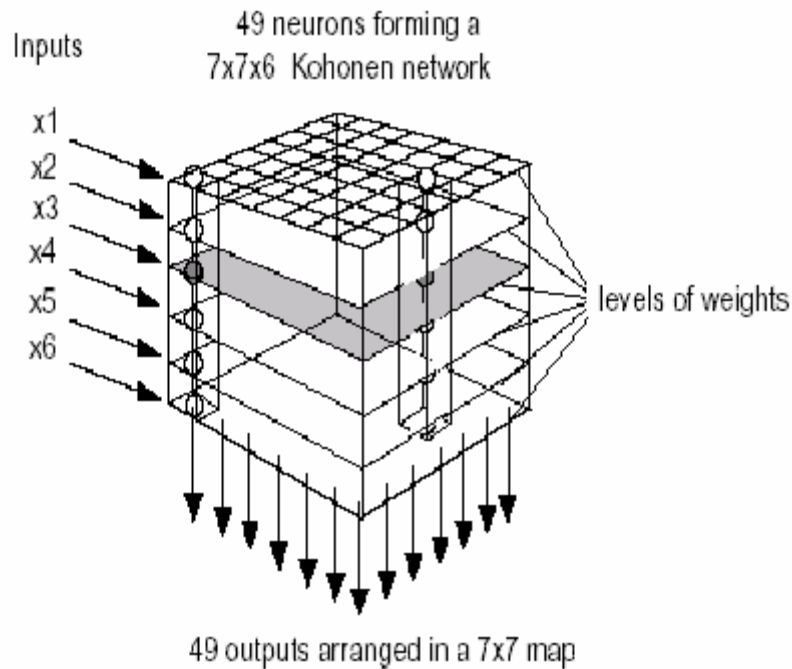
- (i) the neuron with the most 'distinguished' response of all (in a sense explained above) is selected and named the 'central' or the 'most excited' neuron
- (ii) the maximal neighbourhood around this central neuron is determined.

- (iii) the 'correction factor' is calculated for each neighbourhood ring separately (the correction changes according to the distance and time of training)
- (iv) the 'weights' in neurons of each neighbourhood are corrected according to a pre-specified equation

The most important difference is that the neurons in the error back propagation learning (in that of the most famous multi-layer perceptron type of architected neural network) tries to yield quantitatively an answer as close as possible to the target, while in the Kohonen approach the neurons learn to pin-point the location of the neuron in the ANN that is most 'similar' to the input vector.

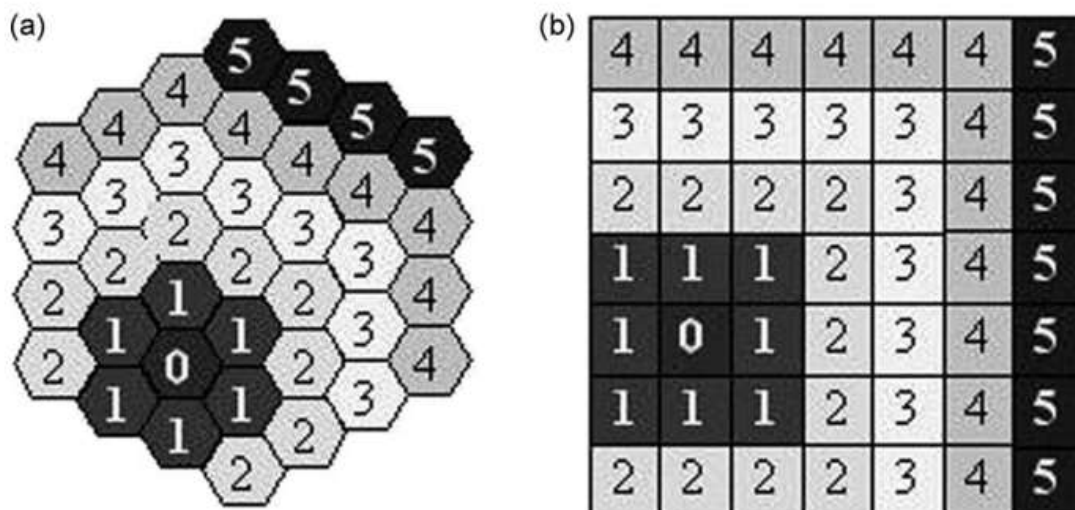
In order to make things clear, let us consider the following figure wherein there are six input variables along with a two-dimensional map of order 7×7 . The neurons are in the columns associating the input variables with the (i, j) -th neuron in the output map, with weights at various levels corresponding to the inputs. That is, because the Kohonen ANN has only one layer of neurons, the specific input variable, let us say the i -th variable x_i is always received in all neurons of the ANN by the weight placed in the i -th position. If the neurons are presented as columns of weights then all i -th weights in all neurons can be regarded as the weights of the i -th level (Zupan, 1994).

Because the Kohonen ANN has only one layer of neurons the specific input variable, let us say the i -th variable, x_i , is always received in all neurons of the ANN, by the weight placed at the i -th position. If the neurons are presented as columns of weights then all i -th weights in all neurons can be regarded as the weights of the i -th level. This is especially important because the neurons are usually ordered in a two-dimensional formation.



Thus, the main goal of Kohonen is to perform a non-linear mapping from a high-dimensional variable space to a low-dimensional (usually 2D) target space so that the distance and proximity relations between the samples or, in a single word, the topology, are preserved. The target space used in Kohonen mapping is a two-dimensional array of neurons fully connected to the input layer, onto which the samples are mapped. Introducing the preservation of topology, results in specifying for each node in the Kohonen layer, a defined number of neurons as nearest neighbors, second-nearest neighbors and so on.

The layout of neurons in the Kohonen ANN is another important feature to be discussed (Marini *et al.*, 2007). The neighborhood of a neuron is usually considered to be hexagonal [see (a) in figure below] or square [see (b) in figure below] which means that each neuron has eight or six nearest neighbors, respectively.



The main issue in Kohonen learning is that similar input vectors excite neurons which are very close in the 2D layer. From an algorithmic point of view, Kohonen mapping implements competitive learning, i.e. only one neuron in the 2D layer is selected after each input is presented to the network (winner takes-all). The winning neuron c is selected as the one having the weight vector most similar to the input pattern. After the winning neuron in the Kohonen layer is selected, the weights of each other neuron in the Kohonen layer are updated on the basis of the difference between their old value and the values of the input vector; this correction is scaled according to the topological distance from the winner.

References

Haykin, S. (1996). *Neural networks: A comprehensive foundation*, Pearson Education, Asia.

Marini, F., Magri, A. L., Bucci, R. and Magri, A.D. (2007). Use of different artificial neural networks to resolve binary blends of monocultivar Italian olive oils, *Analytica Chimica Acta*, **599**, 232–240.

Zupan, J. (1994). Introduction to Artificial Neural Network (ANN) methods: What they are and how to use them, *Acta Chimica Slovenica*, **41** (3), 327-352.

Cluster Analysis

Md. Wasi Alam
ICAR-IASRI, New Delhi
wasi@iasri.res.in

With the advances in genomics, microarray technologies and large amounts of data produced, the interest in unsupervised classification has increased due to the emergence of several new areas of applications. In multivariate setup, when there is no prior idea about groups, unsupervised classification is done through clustering techniques. These include document clustering and the analysis of web use data; gene expression data from microarray technologies, where one goal is to find of those genes that act together, hierarchical clustering algorithm has been applied to identify groups of co-regulated yeast genes; in image analysis, where clustering is used for image segmentation and quantization; In data mining, which started from the search for groupings of customers and products in massive retail data sets.

Cluster Analysis

Cluster analysis classifies a set of observations into two or more mutually exclusive unknown groups based on combinations of clustering variables. Thus, it is a tool of unsupervised learning. The purpose of cluster analysis is to discover a system of organizing observations, usually people, into groups, where members of the groups share properties in common. It is cognitively easier for people to predict behavior or properties of people or objects based on group membership, all of whom share similar properties. It is generally cognitively difficult to deal with individuals and predict behavior or properties based on observations of other behaviors or properties. For example, a person might wish to predict how an animal would respond to an invitation to go for a walk. He or she could be given information about the size and weight of the animal, top speed, average number of hours spent sleeping per day, and so forth and then combine that information into a prediction of behavior. Alternatively, the person could be told that an animal is either a cat or a dog. The latter information allows a much broader range of behaviors to be predicted. The trick in cluster analysis is to collect information and combine it in ways that allow classification into useful groups, such as dog or cat.

Cluster Analysis VS Discriminant Analysis

- Cluster analysis (unsupervised classification) classifies unknown groups while discriminant function analysis (Supervised classification) classifies known groups.
- In Discriminant Analysis (DA), the clusters (groups) are determined beforehand and the object is to determine the linear combination of independent variables which best discriminate among the clusters. In Cluster Analysis (CA), the clusters are not pre-determined and in fact the object is to determine the best way in which the cases may be clustered into groups.
- The procedure for doing a discriminant function analysis is well established. There are few options, that need to be specified when doing a discriminant function analysis. Cluster analysis, on the other hand, allows many choices. Each choice may result in a different grouping structure.

The purpose of this lecture is to give an understanding of how cluster analysis works, which has been explained at the end with an illustrative example. The interested reader is referred to a more comprehensive work on cluster analysis

Application of Cluster Analysis

Cluster analysis is the statistical method of partitioning a sample into homogeneous classes to produce an operational classification. Such a classification may help:

- To formulate hypotheses concerning the origin of the sample, e.g. in evolution studies
- To cluster the genes on the basis of expression levels
- To describe a sample in terms of a typology, e.g. for market analysis or administrative purposes
- To predict the future behavior of population types, e.g. in modeling economic prospects for different industry sectors
- To optimize functional processes, e.g. business site locations or product design
- To assist in identification, e.g. in diagnosing diseases
- To measure the different effects of treatments on classes within the population, e.g. with analysis of variance

Stages in Cluster Analysis for Decision process

Stage-1 Research Problem (Select the Objectives): Taxonomy description, Data Simplification, Reveal relationships, Select Clustering variables

- Stage-2
1. Research Design issues: Can outliers be detected? Should the data be standardized?
 2. Select a similarity measure: Are the Clustering variables metric or Nonmetric?

Metric Data: Is the focus on pattern or Proximity?

Pattern: Correlation measure of similarity-Correlation Coefficients

Proximity: Distance measure of similarity- Euclidean Distance, City Block distance, Mahalanobis distance

Non-metric Data: Association of similarity- Matching coefficients

3. Standardization options: Standardizing variables, Standardizing by observations

Stage-3 Assumptions: Is the sample representative of the population?
Is multicollinearity enough to affect the results?

Stage-4 1. Selecting a clustering algorithm: Hierarchical, Nonhierarchical and Combination

2. How many clusters are formed?
Examine increases in agglomeration coefficient, examine dendrogram and vertical icicle plots, conceptual considerations
3. Cluster analysis respecification: were observation deleted as outlier, members of small cluster

Stage-5 Interpreting the clusters: Examine cluster centroids, Name clusters based on clustering variables

Stage-6 Validating and profiling the clusters: validation with selected outcome variables, profiling with additional descriptive variables.

Methods of Cluster Analysis

I. Hierarchical Methods (no need of pre-specified number of clusters): The following two general methods of hierarchical clustering methods [1, 4,5,6] are available

- **The Divisive Hierarchical Techniques** start by assuming a single group, partitioning that group into subgroups, partitioning these subgroups further into subgroups and so on until each object forms its own subgroup.
- **The Agglomerative Hierarchical Techniques (commonly used)** start with each object describing a subgroup, and then combine like subgroups into more inclusive subgroups until only one group remains. The agglomerative techniques have been described along with illustrative example at the end of this lecture.

Merit of Hierarchical Methods:

- No need of determining the number of clusters in advance.
- It is fast and taking less computer time.
- This method is valid for any kind of data like quantitative, binary, or count data.

Demerit of Hierarchical Methods:

- Outliers greatly affect the results particularly with the Complete Linkage (CLINK)
- Hierarchical Methods are not amenable to analyze very large samples. As sample size increases, the data storage requirements increase dramatically.

Steps involved in Agglomerative Hierarchical Techniques for grouping N objects (items or variables)

1. Start with N clusters, each containing a single entity and an N x N Symmetric matrix of distances (or similarities) $D = \{ d_{ik} \}$.

2. Search the distance matrix for the nearest (most similar) pair of clusters. Let the distance between most similar clusters U and V be d_{UV} .
3. Merge clusters U and V. Label the newly formed cluster (UV). Update the entries in the distance matrix by
 - a. Deleting the rows and columns corresponding to clusters U and V
 - b. Adding a row and column giving the distances between cluster (UV) and the remaining clusters.
4. Repeat steps 2 and 3 a total of N-1 items (all objects will be in a single cluster at termination of the algorithm). Record the identity of clusters that are merged and the levels (distances or similarities) at which the mergers take place.

To clarify the concepts, an illustrative example has been presented at the end of this lecture both in SAS (1.3) and SPSS (1.1, 1.2).

Kinds of Agglomerative Hierarchical Methods

1. Linkage Methods

- **Single Linkage (SLINK):** Single linkage (nearest neighbor) computes the distance between two subgroups as the minimum distance between any two members of opposite groups
 - **Complete Linkage (CLINK):** Complete linkage (furthest neighbor) computes the distance between subgroups in each step as the maximum distance between any two members of the different groups.
 - **Average Linkage:** Average linkage computes the distance between subgroups at each step as the average of the distances between the two subgroups.
2. **Ward's Method:** Agglomerative Hierarchical clustering procedure in which the similarity used to join clusters is calculated as the sum of squares between the two clusters summed over all variables. This method has the tendency to result in

clusters of approximately equal size due to its minimization of within-group variation.

3. **Centroid Method:** Agglomerative Hierarchical clustering algorithm in which similarity between clusters is measured as the distance between cluster centroids. When two clusters are combined, a new centroid is computed. Thus, cluster centroids migrate, or move, as the clusters are combined.

Hierarchical cluster analysis data considerations

Data: The variables can be quantitative, binary, or count data. Scaling of variables is an important issue--differences in scaling may affect cluster solution(s). If variables have large differences in scaling (for example, one variable is measured in centimeter and the other is measured in grams.), one should consider standardizing them (this can be done automatically by the Hierarchical Cluster Analysis procedure in Standard Statistical Software like SPSS).

Assumptions: The distance or similarity measures used should be appropriate for the data analyzed. Also, one should include all relevant variables in your analysis. Omission of influential variables can result in a misleading solution. Because hierarchical cluster analysis is an exploratory method, results should be treated as tentative until they are confirmed with an independent sample.

Hierarchical cluster analysis Plots

Dendrogram: Displays a dendrogram. Dendrograms can be used to assess the cohesiveness of the clusters formed and can provide information about the appropriate number of clusters to keep.

Icicle: Displays an icicle plot, including all clusters or a specified range of clusters. Icicle plots display information about how cases are combined into clusters at each iteration of the analysis. Orientation allows you to select a vertical or horizontal plot.

Hierarchical cluster analysis statistics

Agglomeration schedule: Displays the cases or clusters combined at each stage, the distances between the cases or clusters being combined, and the last cluster level at which a case (or variable) joined the cluster.

- **Proximity matrix:** Gives the distances or similarities between items.
- **Cluster Membership:** Displays the cluster to which each case is assigned at one or more stages in the combination of clusters.

Hierarchical cluster analysis transform values

The following alternatives are available for transforming values in standard Statistical software:

- **Z scores:** Values are standardized to z scores, with a mean of 0 and a standard deviation of 1.
- **Range -1 to 1:** Each value for the item being standardized is divided by the range of the values.
- **Range 0 to 1:** The procedure subtracts the minimum value from each item being standardized and then divides by the range.
- **Maximum magnitude of 1:** The procedure divides each value for the item being standardized by the maximum of the values.
- **Mean of 1:** The procedure divides each value for the item being standardized by the mean of the values.

Choice of Distance / Similarity / Dissimilarity/ Resemblance coefficients

Hierarchical cluster analysis measures for interval data

One of the following dissimilarity measures can be used for interval data:

- **Euclidean distance:** This is probably the most commonly chosen type of distance. It simply is the geometric distance in the multidimensional space. It is computed as distance $(x,y) = \{ \sum_i (x_i - y_i)^2 \}^{1/2}$. The Euclidean (and squared Euclidean) distances are usually computed from raw data, and not from standardized data. This method has certain advantages (e.g., the distance between any two objects is not affected by the addition of new objects to the analysis, which may be outliers). However, the

distances can be greatly affected by differences in scale among the dimensions from which the distances are computed.

- **Squared Euclidean distance:** It is the square of the standard Euclidean distance and is used in order to place progressively greater weight on objects that are further apart.

This distance is computed as $\text{distance}(x,y) = \sum_i (x_i - y_i)^2$

- **Pearson correlation:** The product-moment correlation between two vectors of values.
- **Cosine:** The cosine of the angle between two vectors of values.
- **Chebychev:** The maximum absolute difference between the values for the items. This distance measure may be appropriate in cases when one wants to define two objects as different if they are different on any one of the dimensions. The Chebychev distance is computed as

$\text{distance}(x,y) = \text{Maximum}|x_i - y_i|$

- **City-block (Manhattan) distance:** This distance is simply the average difference across dimensions. In most cases, this distance measure yields results similar to the simple Euclidean distance. However, note that in this measure, the effect of single large differences (outliers) is dampened (since they are not squared). The city-block distance is computed as

$\text{distance}(x,y) = \sum_i |x_i - y_i|$

- **Minkowski:** The pth root of the sum of the absolute differences to the pth power between the values for the items.
- **Customized:** The rth root of the sum of the absolute differences to the pth power between the values for the items.

Hierarchical cluster analysis measures for binary data

One of the following dissimilarity measures can be used for binary data:

- **Euclidean distance:** Computed from a fourfold table as $\text{SQRT}(b+c)$, where b and c represent the diagonal cells corresponding to cases present on one item but absent on the other.

- **Squared Euclidean distance:** Computed as the number of discordant cases. Its minimum value is 0, and it has no upper limit.
- **Size difference:** An index of asymmetry. It ranges from 0 to 1.
- **Pattern difference:** Dissimilarity measure for binary data that ranges from 0 to 1. Computed from a fourfold table as $bc/(n^2)$, where b and c represent the diagonal cells corresponding to cases present on one item but absent on the other and n is the total number of observations.
- **Variance:** Computed from a fourfold table as $(b+c)/4n$, where b and c represent the diagonal cells corresponding to cases present on one item but absent on the other and n is the total number of observations. It ranges from 0 to 1.
- **Dispersion:** This similarity index has a range of -1 to 1.
- **Shape:** This distance measure has a range of 0 to 1, and it penalizes asymmetry of mismatches.
- **Simple matching:** This is the ratio of matches (1-1 and 0-0) to the total number of values. Equal weight is given to matches and nonmatches.
- **Phi 4-point correlation:** This index is a binary analog of the Pearson correlation coefficient. It has a range of -1 to 1.
- **Lambda:** This index is Goodman and Kruskal's lambda. Corresponds to the proportional reduction of error (PRE) using one item to predict the other (predicting in both directions). Values range from 0 to 1.
- **Anderberg's D:** Similar to lambda, this index corresponds to the actual reduction of error using one item to predict the other (predicting in both directions). Values range from 0 to 1.
- **Dice:** This is an index in which joint absences are excluded from consideration, and matches are weighted double. Also known as the Czekanowski or Sorensen measure.
- **Hamann:** This index is the number of matches minus the number of nonmatches, divided by the total number of items. It ranges from -1 to 1.
- **Jaccard:** This is an index in which joint absences are excluded from consideration. Equal weight is given to matches and non-matches. Also known as the similarity ratio.

- **Kulczynski 1:** This is the ratio of joint presences to all nonmatches. This index has a lower bound of 0 and is unbounded above. It is theoretically undefined when there are no nonmatches; however, the software assigns an arbitrary value of 9999.999 when the value is undefined or is greater than this value.
- **Kulczynski 2:** This index is based on the conditional probability that the characteristic is present in item, given that it is present in the other. The separate values for each item acting as predictor of the other are averaged to compute this value.
- **Lance and Williams:** Computed from a fourfold table as $(b+c)/(2a+b+c)$, where a represents the cell corresponding to cases present on both items, and b and c represent the diagonal cells corresponding to cases present on one item but absent on the other. This measure has a range of 0 to 1. (Also known as the Bray-Curtis nonmetric coefficient.)
- **Ochiai:** This index is the binary form of the cosine similarity measure. It has a range of 0 to 1.
- **Rogers and Tanimoto:** This is an index in which double weight is given to nonmatches.
- **Russel and Rao:** This is a binary version of the inner (dot) product. Equal weight is given to matches and nonmatches. This is the default for binary similarity data.
- **Sokal and Sneath 1:** This is an index in which double weight is given to matches.
- **Sokal and Sneath 2:** This is an index in which double weight is given to nonmatches, and joint absences are excluded from consideration.
- **Sokal and Sneath 3:** This is the ratio of matches to nonmatches. This index has a lower bound of 0 and is unbounded above. It is theoretically undefined when there are no nonmatches; however, the software assigns an arbitrary value of 9999.999 when the value is undefined or is greater than this value.
- **Sokal and Sneath 4:** This index is based on the conditional probability that the characteristic in one item matches the value in the other. The separate values for each item acting as predictor of the other are averaged to compute this value.

- **Sokal and Sneath 5:** This index is the squared geometric mean of conditional probabilities of positive and negative matches. It is independent of item coding. It has a range of 0 to 1.
- **Yule's Y:** This index is a function of the cross-ratio for a 2 x 2 table and is independent of the marginal totals. It has a range of -1 to 1. Also known as the coefficient of colligation.
- **Yule's Q:** This index is a special case of Goodman and Kruskal's gamma. It is a function of the cross-ratio and is independent of the marginal totals. It has a range of -1 to 1.

Hierarchical cluster analysis measures for count data

One of the following measures can be used for count data:

1. **Chi-square measure:** This measure is based on the chi-square test of equality for two sets of frequencies. This is the default for count data.
2. **Phi-square measure:** This measure is equal to the chi-square measure normalized by the square root of the combined frequency.

II. Non-Hierarchical Methods:

This Procedures produce only a single cluster solution for a set of cluster seeds. Instead of using the treelike construction process found in the hierarchical procedures, cluster seeds are used to group objects within a prespecified distance of the seeds. For example, if four cluster seeds are specified, only four clusters are formed. Nonhierarchical procedures do not produce results for all possible numbers of clusters as is done with a hierarchical procedure.

In SPSS, you can select one of two methods for classifying cases, either updating cluster centers iteratively or classifying only. You can save cluster membership, distance information, and final cluster centers. Optionally, you can specify a variable whose values are used to label casewise output. You can also request analysis of variance, F statistics.

Example: What are some identifiable groups of television shows that attract similar audiences within each group? With k-means cluster analysis, you could cluster

television shows (cases) into k homogeneous groups based on viewer characteristics. This can be used to identify segments for marketing, or you can cluster cities (cases) into homogeneous groups so that comparable cities can be selected to test various marketing strategies.

Merit of Nonhierarchical method:

- The results are less susceptible to the outliers in the data, the distance measure used and the inclusion of irrelevant or inappropriate variables.
- Nonhierarchical method can be applied to much larger data sets than hierarchical techniques.
- Non random seed points generally provide better result

Demerit of Nonhierarchical method:

- Valid only for quantitative data.
- This method perform better only with the use of non-random (specified) seed points, thus, the use of nonhierarchical techniques with random seed points is markedly inferior to the hierarchical techniques.
- In many instances the researcher gets a different final solution for each set of specified seed points.
- Even a non random (specified) seed points does not guarantee an optimal clustering of observations.
- Random seed point provide poor performance.
- Choice of seed point/number of clusters is a major challenge in this method.
- Only analysis and validation can ensure the best representation of the structures which can lead to select the correct answer to the researcher.

Steps in Non-Hierarchical/ K-means Methods

Macqueen (8) suggested the term K-means for describing his algorithm [7] that assigns each item to the cluster having the nearest centroid (mean).

- Step-1 Partition the items into K initial clusters.
- Step-2 Proceed through the list of items, assigning an item to the cluster whose centroid (mean) is nearest (distance is usually computed using Euclidean distance with either standardized or unstandardized observations). Recalculate the centroid for the cluster receiving the new item and for the cluster losing the items.
- Step-3 Repeat step-2 until no more reassignments take place.

To clarify the concepts, an illustrative example has been presented at the end of this lecture in SPSS (2.1, 2.2).

Kinds of Non-Hierarchical Methods

1. **Sequential Threshold Method:** Non-Hierarchical clustering procedure that begins by selecting one cluster seed. All clusters within a prespecified distance are then included in that cluster. Subsequent cluster seeds are selected until all objects are grouped into a cluster.

Selection of Seed Points in SAS: FASTCLUS program in SAS deals with the sequential threshold procedure designed for large data sets.

2. **Parallel Threshold Method:** This method is the opposite of the Sequential Threshold Method. It is the Non-Hierarchical clustering procedure that selects several cluster seeds simultaneously in the beginning and assign objects within the threshold distance to the nearest seed. Threshold distance can be adjusted to include fewer or more objects in the clusters.

Selection of Seed Points in SPSS: The syntax QUICK CLUSTER in SPSS deals with the parallel threshold methods, which establishes the seed points as user supplied points or select them randomly from all observations.

- 3. Optimizing procedure:** It is the Non-Hierarchical clustering procedure that allows for the reassignment of objects from the originally assigned cluster to another cluster on the basis of overall optimizing criterion. This method is similar to the above two methods except that it allows for reassignment of objects. If in the course of assigning objects, an object becomes closer to another cluster that is not the cluster to which it is currently assigned, then an optimizing procedure switches the object to the more similar (closer) cluster.

Nonhierarchical (K-means) data considerations:

Data. Variables should be quantitative at the interval or ratio level. If your variables are binary or counts, use the Hierarchical Cluster Analysis procedure.

Assumptions. Distances are computed using simple Euclidean distance. If you want to use another distance or similarity measure, use the Hierarchical Cluster Analysis procedure. Scaling of variables is an important consideration--if your variables are measured on different scales (for example, one variable is expressed in dollars and another is expressed in years), your results may be misleading. In such cases, you should consider standardizing your variables before you perform the k-means cluster analysis (this can be done in the Descriptive procedure). The procedure assumes that you have selected the appropriate number of clusters and that you have included all relevant variables. If you have chosen an inappropriate number of clusters or omitted important variables, your results may be misleading.

How many clusters to be formed (Stopping Rule)?

- It is the most perplexing issue for researcher.
- All the existing criteria and guidelines for determining the number of clusters are adhoc and complex.
- There are following two classes of stopping rules
 - First class of stopping rule is relatively simple, examines some measure of similarity or distance between clusters at each successive step, with the cluster solution defined when the similarity measure exceeds a specified value or

when the successive values between steps makes a sudden jump. When a large increase occurs, the researcher selects the prior cluster solution on the logic that its combination caused a substantial decrease in similarity.

- Second class of stopping rule attempts to apply some form of statistical rule or adapt a statistical test, such as the point-biserial / Tau correlations or likelihood ratio. Although some of these such as cubic clustering Criterion (CCC) contained in SAS have been shown to have notable success, many seen overcomplex for the improvement they provide over simpler measures. There are a no. of other specific procedures that have been proposed, but none have been found to be substantially better in all situations.

III. Combination of Hierarchical and Non-Hierarchical

By combination of both the methods we gain benefits of one over other. Use a hierarchical method to specify cluster seed points for a nonhierarchical method, First, a hierarchical technique can establish the number of clusters, profile the cluster centers, profile the cluster centers, and identify any obvious outliers. After outliers are eliminated, the remaining observations can then be clustered by nonhierarchical method with the cluster centers from the hierarchical results as the initial seed points. In this way, the advantages of the hierarchical methods are complemented by the ability of the Non-Hierarchical methods to fine-tune the results by allowing the switching of cluster membership.

How Does Agglomerative Hierarchical Cluster Analysis (e.g. Average Linkage Method) Work ?

The primary objective of the CA is to define the structure of the data by placing the most similar observations into groups, which is done by addressing three iterative basic questions

1. How do we measure similarity?

Depending upon the kinds of trait considered, among all the available distance measures explained earlier, select one of them. In the following illustrative example, trait considered is quantitative and the distance measure considered is Simple Euclidean distance. First, we calculate similarity among ten genotypes based on simple Euclidean distance between each pair of genotypes. Proximity matrix obtained through Statistical Software packages provide such information. Smaller distance indicates greater similarity such as BAHAGIA and CP_231 are most similar (3.367) and BAHAGIA and KHIRADHA are the most dissimilar (46.229).

2. How do we perform clusters?

Once we have similarity measure, we perform clustering by following either of the foregone clustering methods. In the following illustrative example, we use average linkage method for clustering the genotypes, which is done by following the simple rule. Identify the most similar (closest) genotypes not already in the same cluster and combine their clusters. We apply this rule repeatedly, starting with each genotype in its own cluster and combining two clusters at a time until all genotypes are in a single cluster.

3. How many clusters do we form?

A hierarchical method results in a number of cluster solutions [9]-in this case they range from one cluster solution to nine cluster solution. But which one should we choose? We know that as we move from single member cluster, homogeneity decreases. So why not stay at ten clusters, the most homogeneous possible? The problem is that we have not defined any structure with ten clusters. So, the researcher must view each cluster solution for its description of structure balanced against the homogeneity of the clusters. In the following example we use a very simple measure of homogeneity: the average distances of all genotypes within clusters. In the initial solution with ten clusters, overall similarity measure is 0- no observation is paired with another.

Illustrative Example

The nature of cluster analysis can be illustrated by considering a simple bivariate example. Suppose in a breeding research program a breeder wishes to determine the clusters (groups) of ten Rice genotypes (ADT_26, Bahagia, Bircogan, CP_231, Cauvery, Gharbhar, Junga, Jalgaon, Karahani, Khiradha) in which genotypes are similar or homogeneous in performance within the cluster with respect to the clustering variables plant height and grain yield.

Genotypes	P height	Grain yld
ADT_26	165	26.86
BAHAGIA	142.67	16.53
BIRCOGAN	157.67	18.7
CP_231	146	17.03
CAUVERY	146.33	20.66
GHARBHAR	161.33	16.1
JUNGA	179	25.8
JALGAON_	167	20.06
KARAHANI	171.66	30.4
KHIRADHA	188.67	21.13

1.1 To Obtain Agglomerative Hierarchical Cluster Analysis (Average Linkage method) in SPSS

From the menus choose: Analyze → Classify → Hierarchical Cluster...

If you are clustering cases, select at least one numeric variable. If you are clustering variables, select at least three numeric variables. Optionally, you can select an identification variable to label cases.

1.2 Syntax for Hierarchical Cluster Analysis in SPSS

```
PROXIMITIES adt_26 bahagia bircogan cp_231 cauvery gharbhar junga jalgaon_
karahani khiradha
/MATRIXOUT ('C:\DOCUME~1\IASRI\LOCALS~1\Temp\spss1376\spssclus.tmp')
/VIEW= VARIABLE
/MEASURE= EUCLID
/PRINT NONE
/STANDARDIZE= NONE .
CLUSTER
/MATRIX IN ('C:\DOCUME~1\IASRI\LOCALS~1\Temp\spss1376\spssclus.tmp')
/METHOD WAVERAGE
/PRINT SCHEDULE CLUSTER(2,4)
```

```

/PRINT DISTANCE
/PLOT DENDROGRAM .
ERASE FILE= 'C:\DOCUME~1\IASRI\LOCALS~1\Temp\spss1376\spssclus.tmp'.

```

1.3 Code for Agglomerative Hierarchical Cluster Analysis in SAS

```

data genotype;
input Gentp$ PLANT_HT GRAIN_YL;
cards;
ADT_26          165.00      26.86
BAHAGIA         142.67      16.53
BIRCOGAN        157.67      18.70
CP_231          146.00      17.03
CAUVERY         146.33      20.66
GHARBHAR        161.33      16.10
JUNGA           179.00      25.80
JALGAON         167.00      20.06
KARAHANI        171.66      30.40
KHRADHA         188.67      21.13
;

/*----- Average linkage -----*/
proc cluster data=genotype
run;
proc tree horizontal spaces=2;
  id Gentp;
run;
/*----- Centroid method -----*/
proc cluster data=genotype method=centroid pseudo;
  id Gentp;
run;

proc tree horizontal spaces=2;
  id Gentp;
run;
/*----- Density linkage with 3rd-nearest-neighbor --*/
proc cluster data= genotype method=density k=3;
  id Gentp;
run;
proc tree horizontal spaces=2;
  id Gentp;
run;
/*----- Single linkage -----*/
proc cluster data= genotype method=single;
  id Gentp;
run;
proc tree horizontal spaces=2;
  id Gentp;

```

```

run;
/* Ward's minimum variance */
proc cluster data= genotype method=ward pseudo;
  id Gentp;
run;
proc tree horizontal spaces=2;
  id Gentp;
run;

```

R-codes for above problem

Illustrative Example

The nature of cluster analysis can be illustrated by considering a simple bivariate example. Suppose in a breeding research program a breeder wishes to determine the clusters (groups) of ten Rice genotypes (ADT_26,Bahagia, Bircogan,CP_231, Cauvery, Gharbhar, Junga, Jalgaon, Karahani, Khiradha) in which genotypes are similar or homogeneous in performance within the cluster with respect to the clustering variables plant height and grain yield.

Genotypes	P height	Grain yld
ADT_26	165	26.86
BAHAGIA	142.67	16.53
BIRCOGAN	157.67	18.7
CP_231	146	17.03
CAUVERY	146.33	20.66
GHARBHAR	161.33	16.1
JUNGA	179	25.8
JALGAON_	167	20.06
KARAHANI	171.66	30.4
KHIRADHA	188.67	21.13

R-codes for above problem

```

ph=c(165, 142.67, 157.67, 146, 146.33, 161.33, 179, 167, 171.66, 188.67)
gy=c(26.86,16.53,18.7,17.03, 20.66, 16.1, 25.8, 20.06, 30.4, 21.13)
gntp=c("ADT26 ", "BAHAGIA ", "BIRCOGAN ", "CP_231 ", "CAUVERY ",
"GHARBHAR ", "JUNGA ", "JALGAON ", "KARAHANI ", "KHIRADHA ")
d=data.frame(ph,gy,gntp)
d # data can be viewed in R console as given below
w=data.frame(ph,gy)
row.names(w)= c("ADT26 ", "BAHAGIA ", "BIRCOGAN ", "CP_231 ", "CAUVERY ",
"GHARBHAR ", "JUNGA ", "JALGAON ", "KARAHANI ", "KHIRADHA ")
w
d1=round(dist(d),2) # for Euclidean distance

```

```

d2=dist(w)
tree=hclust(d2, "average") # average linkage method
tree
plot(tree ) # plots dendrogram
plot(tree, hang=-0.01, main="single linkage", sub=" ")
m= tree$merge # compute mergers, row I is merger formed at step I(negative no. refer to individual genotype while positive numbers are clusters)
h=tree$height
data.frame(m,h)

```

2.1 To Obtain a Nonhierarchical (K-Means) Cluster Analysis in SPSS

From the menus choose: Analyze → Classify → K-Means Cluster...

Select the variables to be used in the cluster analysis. Specify the number of clusters. The number of clusters must be at least two and must not be greater than the number of cases in the data file. Select either the Iterate and classify method or the classify only method. Optionally, you can select an identification variable to label cases.

Maximum iterations. Limits the number of iterations in the k-means algorithm. Iteration stops after this many iterations even if the convergence criterion is not satisfied. This number must be between 1 and 999.

To reproduce the algorithm used by the Quick Cluster command prior to version 5.0, set Maximum iterations to 1.

Convergence Criterion. Determines when iteration ceases. It represents a proportion of the minimum distance between initial cluster centers, so it must be greater than 0 but not greater than 1. If the criterion equals 0.02, for example, iteration ceases when a complete iteration does not move any of the cluster centers by a distance of more than two percent of the smallest distance between any of the initial cluster centers.

Use running means. Allows you to request that cluster centers be updated after each case is assigned. If you do not select this option, new cluster centers are calculated after all cases have been assigned.

2.2 Syntax for Nonhierarchical (k-means) cluster analysis

2.1 By specified Cluster seed points

```

QUICK CLUSTER
  plant_ht grain_y1
/ INITIAL= ( )
/MISSING=LISTWISE
/CRITERIA= CLUSTER (5) MXITER(10) CONVERGE(0)
/METHOD=KMEANS (UPDATE )

```

/SAVE CLUSTER DISTANCE
 /PRINT ID(case lbl) INITIAL ANOVA CLUSTER DISTAN.

2.2 By Random Selection of Cluster Seed Points

SET SEED 346578
 QUICK CLUSTER
 / CRITERIA=CLUSTER(5) NOINITIAL
 /PRINT ID(Case_lbl) INITIAL ANOVA CLUSTER DISTAN

Problem 2: Suppose five individuals possess the following characteristics (Data taken from Johnson and Wichern (1988):

Genotypes/Individual	Height (in inch)	Weight (in lb)	Eye Color	Hair Color	Handedness	Sex
G1	68	140	Green	Blond	Right	Female
G2	73	185	Brown	Brown	Right	Male
G3	67	165	Blue	Blond	Right	Male
G4	64	120	Brown	Brown	Right	Female
G5	76	210	Brown	Brown	Left	Male

Define six binary variables $X_1, X_2, X_3, X_4, X_5, X_6$ as

$$X_1 = \begin{cases} 1 & \text{height} \geq 72 \text{ in} \\ 0 & \text{height} < 72 \text{ in} \end{cases} \quad X_2 = \begin{cases} 1 & \text{weight} \geq 150 \text{ lb} \\ 0 & \text{weight} < 150 \text{ lb} \end{cases}$$

$$X_3 = \begin{cases} 1 & \text{1 brown eyes} \\ 0 & \text{0 otherwise} \end{cases} \quad X_4 = \begin{cases} 1 & \text{1 blond hair} \\ 0 & \text{0 not blond hair} \end{cases}$$

$$X_5 = \begin{cases} 1 & \text{1 right handed} \\ 0 & \text{0 left handed} \end{cases} \quad X_6 = \begin{cases} 1 & \text{1 female} \\ 0 & \text{0 male} \end{cases}$$

Find the groups of genotypes which are homogeneous within group and heterogeneous between groups.

Hints: The scores for individuals 1 and 2 on the $p=6$ binary variables are

Genotype	X_1	X_2	X_3	X_4	X_5	X_6
G1	0	0	0	1	1	1
G2	1	1	1	0	1	0

and the number of matches and mismatches are indicated in the two-way array

	G2		
	1	0	
G1	1	2	3

0	3	0	3
	4	2	6

*Euclidean distance is used in both the cases (standardize the quantitative data).

Binary Variables

- A contingency table for binary data

		Object j		
		1	0	sum
Object i	1	a	b	a+b
	0	c	d	c+d
sum		a+c	b+d	p

Simple matching coefficient (invariant, if the binary variable is symmetric): $d(i, j) = \frac{a+d}{a+b+c+d}$

- Jaccard coefficient of dissimilarity:

$$d(i, j) = \frac{b+c}{a+b+c}$$

- Jaccard's dissimilarity coefficient

- $d(i, j) = \frac{b+c}{a+b+c}$

- Jaccard's similarity coefficient

$$d^*(i, j) = \frac{a}{a+b+c} = 1 - d(i, j)$$

a = # of attributes positive for both objects

q = # of attributes 1 for i and 0 for j

c = # of attributes 0 for i and 1 for j

When used for binary attributes, the Jaccard index is very similar to the **Simple Matching coefficient**. The main difference is that the SMC has the term 0-0 matches in its numerator and denominator, whereas the Jaccard index does not.

Example 1:

Consider five Genotypes (A, B, C, D, E) with the following distance matrix (the data could be molecular or morphological distances): A & B are closest (20 units): join them into one cluster (AB) joining at 20, and recalculate the average distance from C, D, and E to (AB). [For example, the distance from C to (AB) = $(60 + 50)/2 = 55$, and the distance from D to (AB) = $(100 + 90)/2 = 95$]. This gives:

	A	B	C	D	E
A	0	-	-	-	-
B	20	0	-	-	-
C	60	50	0	-	-
D	100	90	40	0	-
E	90	80	50	30	0

1

A & B are closest (20 units): join them into one cluster (AB) joining at 20, and recalculate the average distance from C, D, and E to (AB). [For example, the distance from C to (AB) = $(60 + 50)/2 = 55$, and the distance from D to (AB) = $(100 + 90)/2 = 95$]. This gives:

	(AB)	C	D	E
(AB)	0	-	-	-
C	55	0	-	-
D	95	40	0	-
E	85	50	30	0

D & E are closest (30 units): join them into one cluster (DE) joining at 30, and recalculate the average distances between (AB), C, and (DE). [For example, the distance from (AB) to (DE) = $(95 + 85)/2 = 90$]. This gives:

83

is:

	(AB)	C	(DE)
(AB)	0	-	-
C	55	0	-
(DE)	90	45	0

C & (DE) are closest (45 units): join them into one cluster **(CDE)** joining at 45, and recalculate the average distance between **(CDE)** and **(AB)**. This gives:

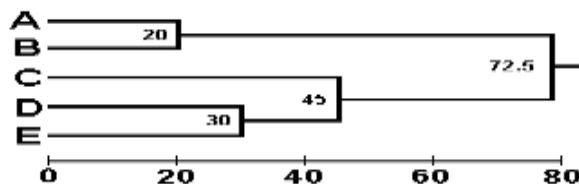
	(AB)	(CDE)
(AB)	0	-
(CDE)	72.5	0

The two clusters join at 72.5. This completes the analysis.

84

UPGMA

These results may be presented as a **phenogram** with nodes at 20, 30, 45, and 72.5 units. The phenogram can be interpreted as indicating that **A & B** are similar to each other, as are **D & E**, and that **C** is more similar to **D & E** :



The unweighted pair-group method with arithmetic mean (UPGMA) is a popular distance analysis method.

85

R-codes for cluster analysis and bi-plots

Data set from R: USArrests

x=USArrests

d=round(dist(x),2)

tree=hclust(d,"average")

tree=hclust(d,"single")

tree=hclust(d, "complete")

```

tree=hclust(d, "centroid" )
p=plot(tree)
m=tree$merge
h=tree$height
plot(tree, hang=-0.01, main= "single")
pc=princomp(USArrests, cor = TRUE)
loadings(pc)
biplot(pc)
princomp(iris[,1:4])
summary(princomp(iris[,1:4]))
hc=hclust(dist(USArrests), "single")
memb=cutree(hc, k=10)
plot(hc, hang=-1, main= "single")
plot(hc)
*****
hc.a=hclust(dist(USArrests), "average" )

p=plot(hc.a, hang=-1, main= "average linkage")
cutree(hc.a, k=4)
hc=hclust(dist(USArrests), "average" )
hc
memb=cutree(hc, k=10)
cent=NULL
for(k in 1:10) {
cent=rbind(cent,colMeans(USArrests[memb == k, , drop=FALSE]))
}
hcl=hclust(dist(cent), method= "average",
members=table(memb))
par(mfrow=c(1,2))
plot(hc,labels=FALSE, hang=-1,
main= "original Tree ")
plot(hcl,labels=FALSE, hang=-1,
main= "Re-start from 10 clusters ")
par(mfrow =c(1,3))
hc.a=hclust(dist(USArrests), "average" )
p=plot(hc.a, hang=-1, main= "average linkage")
hc.s=hclust(dist(USArrests), "single")
ps=plot(hc.s, hang=-2, main= "single linkage")
ps

hc.c=hclust(dist(USArrests), "complete")
pc=plot(hc.c, hang=-1, main= "complete linkage")
pc
hc.a=hclust(dist(USArrests), "average" )
*****
p=plot(hc.a, hang=-1, main= "average linkage")

```

```

cutree(hc.a, k=4)
tmp=cutree(hc.a, k=4)
for ( i in 1:4) {
cat(" Cluster", i, "consists of\n")
print(names(tmp[tmp==i]))
cat("\nAverage crime levels in this cluster are\n")
print(round(apply(USArrests[tmp==i,], 2, mean,),2))
cat("\n\n")
}
x=USArrests
x
d=dist(x,2)
d
hc=hclust(d, "average")
hc
summary(hc)
plot(hc)

cd=hc$merge
cd

h=hc$height
h

```

REFERENCES

- Aldenderfer, Mark, S., and Blashfield, R.k. (1984). *Cluster Analysis*. Thousand Oaks, Calif: Sage Publications.
- Anderberg, M. (1973). *Cluster Analysis for Applications*. New York: Academic Press.
- Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998). Cluster Analysis and Display of Genome Wide Expression Patterns. *Proc. Natl Acad. Sci., USA*, **95**, 14863-14866.
- Green, P.E. (1978). *Analyzing Multivariate Data*. Hinsdale,III.: Holt, Rinehart & Winston.
- Green, P.E., Carrol, J.D. (1978). *Mathematical Tools for Applied Multivariate Analysis*. New York: Academic Press.
- Hair, J.J.F., Anderson, R.E., Tatham,R.L., Black, W.C.(1998). *Multivariate Data Analysis*. Pearson Education, Singapore, Pte. Ltd.
- Johnson, R.A., Wichern, D.W. (1996). *Applied Multivariate Statistical Analysis*. Prentice Hall of India Private Limited, New Delhi 110001.
- MacQueen, J.B (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and probability*. **1**,Berkeley, Calif: University of California Press, 281-297.
- Milligan, G.W., Cooper, M.C. (1985). An Examination of Procedures for Determining the Number of Clusters in a Data Set. *Psychometrika* **50(2)**:159-179.
- Romesburg, H.C. (1984). *Cluster analysis for Researchers*. Life time learning publications Belmont, California.
- Sneath, P.H.A. and Sokal, R.R. (1973). *Numerical Taxonomy*. San Francisco: Freeman Press.

Logit, Probit and Tobit Models

Shivaswamy G. P., K N Singh, Anuja A. R., Rajesh T and HarishKumar H V
ICAR-IASRI, New Delhi
Shivaswamy.gp@icar.gov.in

Introduction

Regression analysis is a statistical method of studying functional relationship between a dependent or response variable and one or more independent or explanatory variables. In classical linear regression models (CLRM), response variable is implicitly assumed as quantitative and explanatory variables can be quantitative or qualitative. Parameter estimation in the CLRM is based on important assumptions such as linearity of model in parameters, though response and explanatory variables may or may not linearly related; explanatory variables are independent of the error terms; independent and identically distributed error terms with zero mean and constant variance; and equal reliability of observations. However, when the response variable is qualitative, these basic assumptions of CLRM may not hold. For instance, one wants to study the adoption of high yielding varieties of a crop which is a binary response variable. It takes only two values: 1 if the variety is adopted and 0 if it is not. There are several instances, where the response variable is binary. Suppose one wants to study the determinants of access to institutional credit such as age, gender, education, social status, land holding etc. Whether a farmer has access to institutional credit is a binary variable taking values 0 or 1, 0 meaning no access and 1 meaning access to institutional credit. Similarly, other examples can be crop diversification status as a function of various quantitative or qualitative independent variables. In such cases, error terms are not normally distributed and the variance is not constant thereby violating the homoscedasticity assumption. The statistical models preferred for the analysis of such a binary responses are logit and probit models as these models do not make assumptions on the distribution of explanatory variables.

1. The Logistic regression

Logistic regression analysis is used when the dependent variable is qualitative and normality assumption is not satisfied. This model was developed by Cox (1958). Logistic regression is appropriate when the dependent and independent variables are non-linearly related. Logit is

transformation of logistic regression to make it linear. In case of logit transformation, binary variable (Koutsoyiannis, 2001).

In case of adoption of a rice variety, the decision to adopt or not to adopt by i th individual depends on the latent variable Y_i^* which in turn depends on explanatory variables such as age, education, farm size, access to institutional credit, irrigation facility and training.

$$Y_i^* = BX + u_i \quad (1)$$

Where B is the vector of parameters, X is the vector of explanatory variables and u_i is the error term of i^{th} individual

It is assumed that

$$Y_i = \begin{cases} 0 & \text{if } Y_i^* \leq 0 \\ 1 & \text{if } Y_i^* > 0 \end{cases}$$

That is if individual's utility index exceeds Y_i^* farmer will adopt a variety, but if it is less than Y_i^* then variety is not adopted.

In case of probability of adoption of variety ($Y=1$), the logistic regression function can be expressed as

$$P_i = Pr(Y_i = 1|X = x) = \frac{1}{(1 + e^{-Z_i})} \quad (2)$$

Where, $Z_i = BX + u_i$

The probability that variety is not adopted ($Y=0$) is given by

$$(1 - P_i) = \frac{1}{(1 + e^{Z_i})}$$

Where, as Z_i ranges from $-\infty$ to $+\infty$, P_i ranges between 0 and 1. And, the model is nonlinear both in response variables X and parameters Bs.

Further, to make the logistic regression function linear in the parameters, we take the ratio of probability that farmer adopts a variety to probability that he is not;

$$\frac{P_i}{1 - P_i} = \frac{1}{\frac{1}{(1 + e^{-Z_i})}}$$

$$\frac{P_i}{1 - P_i} = e^{Z_i} \quad (3)$$

Where, $P_i/(1 - P_i)$ is known as the odds ratio in favor of adoption of a variety i.e. the ratio of probability that a farmer adopts a variety to probability that he does not adopt.

Equation can be transformed by taking natural logarithm as follows

$$\ln\left(\frac{P_i}{1 - P_i}\right) = Z_i \quad (4)$$

Log of the odds ratio is known as the logit which is nothing but a linear transformation of the logistic regression model.

Estimation of logit model

Usual OLS method cannot be used to estimate the logit model despite its linearity properties due to problem of undefined expressions. Rather, Maximum likelihood estimation method is used for estimation.

Variables used for the logit analysis of determinants of variety adoption example are as follows

Adoption=1 for adopters and 0 for non-adopters

Age in years

Education=1 if educated; 0 otherwise Credit=1 if there is access to institutional credit; 0 otherwise

Irrigation=1 if there is access to irrigation; 0 for non-access

Training=1 if undergone training; 0 otherwise

Table 1 Sample data set for logit and probit analysis

Observations	Adoption	Age	Education	Farm size	Credit	Irrigation	Training
1	0	70	1	9.01	0	0	0
2	0	30	0	2.136	1	1	0
3	1	40	1	1.12	1	1	0
4	1	60	0	1.003	0	1	0
5	0	30	1	2.61	1	0	1
6	0	60	1	1.23	0	1	1
7	1	45	1	2.434	1	0	0
.							
.							
.							
1763	1	52	1	1.705	1	0	0

Table 2 provides the results of logit model for the adoption of variety example, which are obtained by *STATA* using the command *logit*.

Table 2 Logit estimates of adoption of a rice variety

	Coefficient	Standard error	Z statistic	Prob>Z
Age	0.002	0.003	0.06	0.951
Education	-0.070	0.102	-0.07	0.945
Farm size	-0.014	0.015	-0.92	0.36
Institutional credit	0.489	0.098	4.98	0
Irrigation access	0.299	0.097	3.07	0.002
Training	0.096	0.285	0.34	0.735
Constant	-0.351	0.213	-1.64	0.1
Number of observations	1763			
McFadden R ²	0.014			

The results of logit model show that access to institutional credit and irrigation are statistically significant at 1 percent level of significance. It is interpreted as access to institutional credit increases the average logit value by 0.489. Access to irrigation is also interpreted similarly. Other variables such as age, education and training are statistically insignificant meaning they do not have visible impact on adoption of a variety.

In case of CLRM, R^2 indicates the goodness of fit showing the proportion of variation in the dependent variable explained by the independent variables in the model. But, in case of binary regression models, R^2 is not meaningful for which McFadden R^2 or pseudo R^2 is discussed in the literature. The value of McFadden R^2 ranges between 0 and 1. In our example its value is 0.014. It should be noted that in qualitative regression models, the expected sign of the regression coefficients and their statistical significance are more important than the goodness of fit measures.

We can express the logit coefficients in terms of odds ratio (Table 3) by using following *STATA* command *logit adoption age education farmsize credit irrigation_access training, or*

Table 3 Odds ratio for adoption versus non-adoption

	Odds ratio	Standard error	Z statistic	Prob>Z
Age	1.000	0.004	0.060	0.951
Education	0.993	0.102	-0.070	0.945
Farmsize	0.986	0.015	-0.920	0.360
Institutional credit	1.632	0.161	4.980	0.000
Irrigation	1.349	0.131	3.070	0.002
Training	1.101	0.315	0.340	0.735
Constant	0.704	0.151	-1.640	0.100
Number of observations	1763			
McFadden R^2	0.014			

The odds ratios are obtained by taking the exponential of logit coefficients given in Table 2. The interpretation of the odds ratio depends on whether its value is greater than 1 or less than 1. Odds ratios of greater than 1 indicate the increased chance of adoption as compared to non-adoption. On the other hand, odds ratio of less than 1 indicates the decreased chance of adoption. Odds ratio of 1 suggests that chances of adopting and not adopting are even. In our example, two variables institutional credit and irrigation have the odds ratios of greater than 1 meaning increased chance of adopting a variety as against non-adoption.

Estimation of marginal effects

Marginal effects depicts the marginal impact of one unit change in the explanatory variable on the probability of adoption of a variety. It is a way depicting the model estimates in terms of

probabilities which helps in interpreting in terms of magnitude. Note that instead of computing the marginal effect for each independent variable on the probability of adoption, it is computed for the average values of variables. It is to be noted that for quantitative response variables marginal effect is the derivative ($\frac{dy}{dx}$) of dependent variable (y) with respect of independent variable (x) that is rate of change of y with respect to x. However for qualitative independent variable which takes the discrete values 0 and 1 as in our example, marginal effect is estimated for the discrete change in the qualitative variable from 0 to 1.

Table 4 Marginal effects of logit model

	Marginal effects	Standard error	Z statistic	Prob>Z	Mean value
Age	0	0.001	0.060	0.951	50.558
Education	-0.001	0.026	-0.070	0.945	0.647
Farmsize	-0.003	0.004	-0.920	0.360	2.771
Institutional credit	0.121	0.024	5.030	0.000	0.450
Irrigation	0.074	0.024	3.080	0.002	0.573
Training	0.024	0.071	0.340	0.735	0.029
Number of observations	1763				
McFadden R ²	0.014				

The interpretation of marginal effects in our example is that unit change in age and farm size does not have statistically significant impact on the rate of change in the probability of adoption. For qualitative variable, an access to institutional credit has significant positive impact in the probability of adoption of variety by about 0.121. Similarly access to irrigation increases the probability of adoption by about 0.074.

2. Probit model

Like logit, probit model is used when the response variable is qualitative. Error terms in the probit model follow normal distribution.

For arriving at the probit model, equation 1 can be translated into,

$$\begin{aligned}
 Pr(Y_i = 1|X = x) &= Y_i^* > 0 \\
 &= Pr(u_i > -B'X)
 \end{aligned}$$

$$= Pr\left(\frac{u_i}{\sigma} > \frac{-B'X}{\sigma}\right)$$

$$Pr(Y_i = 1|X = x) = \Phi\left(\frac{-B'X}{\sigma}\right) \quad (5)$$

Estimation of probit model

Probit model is estimated based on the maximum likelihood function which finds coefficients that maximize the probability of $Y_i = 1$ (Spermann, 2008).

Using *STATA* command *probit*, ML estimates of the probit model for adoption of variety are given in Table 5

Table 5 Probit estimates of adoption of variety

	Coefficient	Standard error	Z statistic	Prob>Z
Age	0.000	0.002	0.070	0.943
Education	-0.004	0.064	-0.060	0.951
Farm size	-0.009	0.010	-0.930	0.354
Institutional credit	0.306	0.061	4.990	0.000
Irrigation	0.187	0.061	3.070	0.002
Training	0.061	0.178	0.340	0.730
Constant	-0.221	0.134	-1.650	0.098
Number of observations	1763			
McFadden R ²	0.014			

Although coefficients of logit and probit models are different, the interpretation of coefficients is similar. Institutional credit and irrigation are statistically significant at 1 percent level of significance. It should be noted that only sign of the logit and probit models are interpreted but not the magnitude. The coefficients of logit and probit models are different and can be comparable after multiplying probit coefficients by about 1.81. However, marginal effects of probit and logit models are similar (Table 6). Logit and probit functions are almost similar with both s shaped curves. The main difference between the logit and probit models is that the logistic distribution has slightly flatter tails. Therefore, there is no compelling reason for choosing one model over another (Halloran, 2018).

Table 6 Marginal effects of probit model

	Marginal effects	Standard error	Z statistic	Prob>Z	Mean value
Age	0	0.001	0.060	0.951	50.558
Education	-0.001	0.026	-0.070	0.945	0.647
Farm size	-0.003	0.004	-0.920	0.360	2.771
Institutional credit	0.121	0.024	5.030	0.000	0.450
Irrigation	0.074	0.024	3.080	0.002	0.573
Training	0.024	0.071	0.340	0.735	0.029
Number of observations	1763				
McFadden R ²	0.014				

3. Tobit model

Tobit model, also called as censored regression model or limited dependent variable regression model, was proposed by Tobin in 1958. A censored sample is a sample in which information on dependent variable is available for only some observations in a sample. If we use OLS on censored data set, estimates obtained will be inconsistent meaning coefficients will not necessarily approach the true population parameters as sample size increases (Gujarati, 2003). In such cases, Tobit model is used for analyzing censored sample.

The model can be expressed as

$$Y = X\beta + u \text{ If } \beta'X + u > 0;$$

$$= 0 \text{ otherwise}$$

Such that the residual, $u \sim N(0, \sigma^2)$

Where Y is the $(n \times 1)$ vector of dependent variable, β is the $(k \times 1)$ vector of unknown parameters, and X is the $(n \times k)$ vector of exogenous variables.

The model can be estimated using maximum likelihood method or Heckman two step procedure.

The application of the model can be explained with the help of labour economics example. Using the data set which contains information on both working and non-working married women, suppose we want to estimate the extent of participation of working women in

labour force. Here the dependent variable (extent of participation in hours per year) is continuous and lower limit of dependent variable is zero which means non-participation in labour market. Tobit model can be applied here as it employs all information collected for both working women (where dependent variable is more than zero) and non-working women (dependent variable -zero).

In another example, suppose we want to estimate the amount of money spent by an individual on meat in relation to socio economic variables. Now there are two groups of consumers. One set m_1 , about whom we have information on the independent variables (age, education, income etc.) as well as the dependent variable (the amount of money spent on meat items) and another set m_2 about whom we have information only on the independent variables but not on dependent variable. Here we cannot neglect observations on second group as the OLS estimates of parameters using only first group of observations will be biased and inconsistent. In this case we can use tobit model for a better estimation of parameters.

References

Cox, D. R. 1958. The regression analysis of binary sequences (with discussion). Journal of the Royal Statistical Society B, 20, 215-242

Gujarati, D, N. 2003. Basic econometrics (4th ed.), New York: Mc Graw Hill Publications

Halloran, S. 2018. Logit and probit models, accessed in September 2019
http://www.columbia.edu/~so33/SusDev/Lecture_9.pdf

Koutsoyiannis, A, 2001. Theory of Econometrics, (2nd ed.), New York: Palgrave Macmillan Limited

Spermann, A. 2009. The probit model, accessed in September 2019
https://www.empiwifo.uni-freiburg.de/lehre-teaching-1/summer-term-09/materials-microeconometrics/probit_7-5_09.pdf

ARIMA Model for Time-Series Forecasting

Kanchan Sinha, Mrinmoy Ray, Achal Lama and K.N. Singh
ICAR-IASRI, New Delhi

1. Introduction

Time series modelling deals with time based data (minutes, hours, days, weeks, months, years) to derive hidden insights to make informed decision making. In time series modelling, past observations of the same variable are collected and analyzed to develop a model describing the underlying relationship. The model is then used to extrapolate the time series into the future. This modeling approach is particularly useful when little knowledge is available on the underlying data generating process or when there is no satisfactory explanatory model that relates the prediction variable to other explanatory variables. Time series models are very useful when data are serially correlated.

For time series data, the most important graphical form is a *time plot* in which the data are plotted over time. A time plot immediately reveals any trends over time, any regular seasonal behavior and other systematic features of the data. Further, one of the implicit assumptions of time series forecasting is that future will behave like past. This is achieved through the stationarity condition of a series. Therefore, stationarity is a necessary condition in building an appropriate model for forecasting. A series is said to be weakly stationary if its mean, variance and auto-covariance are constant over time i.e., they are time invariant. In other words, stationarity means that there is no growth or decline in the data. The visual plot of a time series is often enough to convince a forecaster that the data are stationary or non-stationary. Such an intuitive feel is the starting point of more formal test of stationarity. One simple test of stationarity is based on the autocorrelation function (ACF). The autocorrelation refers to the way the observations in a time series are related to each other and is measured by the simple correlation between current observation (y_t) and observation from p periods before the current one (y_{t-p}). It ranges from -1 to +1. The plot of autocorrelation against lag is known as correlogram or ACF plot. The ACF plot of a stationary data drop to zero relatively quickly, while for a non-stationary series they are significantly different from zero for several time lags. The autocorrelation function (ACF) plot of a non-stationary data displays a typical pattern with a slow decrease in the size of autocorrelations. Partial autocorrelations are used to measure the degree of association between y_t and y_{t-p} when the y -effects at other time lags $1, 2, \dots, p-1$ are removed.

There are several statistical tests to determine the stationarity of a series. These are also known as *unit root tests*. The most widely used statistical test for stationarity is Augmented Dickey-Fuller (ADF) test.

The ADF test consists of estimating the following regression equation:

$$\Delta y_t = \beta_1 + \beta_2 t + \delta y_{t-1} + \sum_{i=1}^h \alpha_i \Delta y_{t-i} + \varepsilon_t \quad (i)$$

where Δy_t denotes the differenced series i.e., $y_t - y_{t-1}$. β_1 and β_2 are the parameters of regression model and h is the lag length. The number of lagged difference terms to include is often determined empirically, so that the error term in (i) is serially uncorrelated. In ADF we test whether $\delta = 0$ i.e., we have a unit root, meaning the time series under consideration is nonstationary. When the observed time series presents trend and heteroscedasticity, differencing and transformation are often applied to the data to remove the trend and stabilize variance before a model can be fitted.

2. Model Specification

Auto Regressive Integrated Moving Average (ARIMA) model is used to forecast future values of a series based on past values. Here “AR” means lags of the differenced series appearing in the forecasting equation; “MA” is the lag of the forecast errors and a time series which needs to be differenced for making it stationary is termed as “integrated.” Generally, a non-seasonal ARIMA model is denoted as ARIMA (p, d, q) where p and q are the order of autoregressive and moving average order and d is the order of differencing.

2.1 Autoregressive (AR) Model

Auto Regressive (AR) model utilizes the dependent relationship between an observation and some number of lagged observations. Consider, the values of a process at equally spaced time periods $t, t - 1, t - 2, \dots, t - p$ by $y_t, y_{t-1}, y_{t-2}, \dots, y_{t-p}$ then y_t can be described by the following expression and denoted as AR(p) model.

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t \quad (ii)$$

An autoregressive operator of order p can also be written using backshift operator B as

$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p \quad (iii)$$

such that $B y_t = y_{t-1}$, and the autoregressive model can be defined as $\varphi(B) y_t = \varepsilon_t$.

2.2 Moving Average (MA) Model

Moving Average (MA) model uses the dependency between an observation and a residual error applied to lagged observations. MA (q) model is defined as

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (\text{iv})$$

where q is the parameter of how many lagged observations to be added. A moving average operator of order q is also defined as

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q \quad (\text{v})$$

where B is the backshift operator such that $By_t = y_{t-1}$ and the moving average model can be expressed as $y_t = \theta(B)\varepsilon_t$.

2.3 ARIMA model

In an ARIMA model, the future value of a variable is assumed to be a linear function of several past observations and random errors. That is, the underlying process that generate the time series has the following form

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (\text{vi})$$

where, y_t and ε_t are the actual observation and random error at time period t , respectively; φ_i ($i= 1, 2, \dots, p$) and θ_j ($j= 1, 2, \dots, q$) are model parameters. p and q are integers and often referred to as orders of the model. Random errors ε_t are assumed to be independently and identically distributed with a mean zero and a constant variance σ^2 .

Equation (vi) entails several important special cases of the ARIMA family of models. If $q=0$, then it becomes an AR model of order p . When $p=0$, the model reduces to an MA model of order q . One central task of the ARIMA (p, d, q) model building is to determine the appropriate model order (p, d, q) where d is the order of differencing. The different steps to analyze and forecast of a time series are described in the following manner:

Step 1: Stationarity of the time series

The series being analyzed must be stationary. A stationary time series has the property that its statistical properties such as the mean and variance are constant over time. As indicated earlier the presence of stationarity in the data can be identified by simply plotting the raw data or by plotting the autocorrelation and partial autocorrelation function. Statistical tests like Dickey-Fuller test, augmented Dickey-Fuller test are also available to test the

stationarity. Stationarity in variance could be achieved by some modes of transformation, say, logarithms or square root of transformation. If the series exhibits a trend over time or seasonality or if some other nonstationary pattern exists, the series is differenced repeatedly until the time series becomes stationary.

Step 2: Identification of the model

Candidate ARIMA models are identified once the time series becomes stationary. After obtaining the autocorrelation function (ACF) and partial autocorrelation function (PACF), multiple ARIMA models that closely fit the data can be identified. In the identification step, the order of tentative models could be obtained by looking for significant autocorrelation and partial autocorrelation function.

Step-3: Estimation of model parameters

Once few tentative models are specified, estimation of the model parameters is straightforward. The parameters are estimated through maximum likelihood function such that an overall measure of errors is minimized or the likelihood function is maximized.

Step-4: Diagnostic checking

Different models can be obtained with various combinations of AR and MA individually and collectively. The best model is obtained with following diagnostics.

The most suitable ARIMA model is selected using the smallest Akaike Information Criterion (AIC) or Schwarz-Bayesian Criterion (SBC). AIC is given by the following equation

$$AIC = (-2 \log L + 2m) \quad (\text{vii})$$

where, $m = p + q$ and L is the likelihood function. SBC is also used as an alternative to AIC which is written as

$$SBC = \log \sigma^2 + (m \log n) / n \quad (\text{viii})$$

If the model is not adequate, a new tentative model should be identified and the above steps should be repeated. Diagnostic information may help suggest alternative model(s). These steps of model building process are typically repeated several times until a satisfactory model is finally selected. The final model can then be used for prediction

This step is basically to check if the model assumptions about the errors are satisfied. This is achieved by performing portmanteau test. The test is utilized to see whether the model residuals are white noise. The null hypothesis tested is that the current set of residual is white noise. The Ljung-Box test statistic is utilized to check the residuals that can be expressed as

$$Q = n(n+2) \sum_{k=1}^h (n-k)^{-1} r_k^2 \quad (\text{ix})$$

where, h is the maximum lag, n is the no. of observations, m is the number of parameters in the model. If the data are white noise, the Ljung-Box Q statistic has a chi-square distribution with $(h-m)$ degrees of freedom.

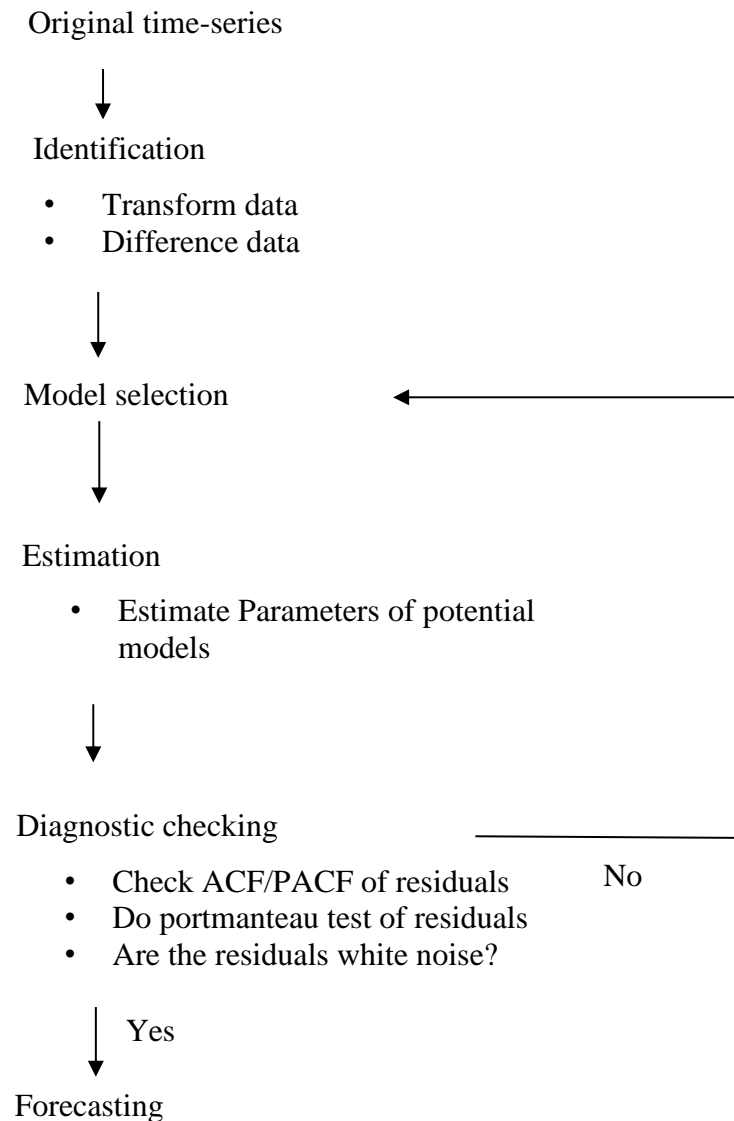


Figure: Schematic representation of the Box-Jenkins methodology for time series modeling

Step-5: Forecast the time series and check accuracy of the model

Finally, different criteria based on error terms is used to inspect the forecasting ability of the method utilized. The most commonly used accuracy measures whose scale depends on the scale of the data is root mean square error (RMSE). This is suitable when comparing different methods applied to the same set of data, but should not be used for example, when

comparing across data sets that have different scales. Root mean square error (RMSE), which measures the overall performance of a model can be expressed as

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (x)$$

where y_t is the actual value for time t , \hat{y}_t is the predicted value for time t and n is the number of predictions.

Mean Absolute Percentage Error (MAPE) is another measure to check how much a dependent series varies from its model-predicted level. Percentage errors have the advantage of being scale independent, and so are frequently used to compare forecast performance across different data sets and can be written as

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|y_t - \hat{y}_t|}{y_t} \times 100 \quad (xi)$$

3. Conclusion

Time series modeling is one of the active areas of research in which historical data of the same variable is used to model the underlying relationship and the developed model is used for projecting the future values. Forecasting of agricultural time series data is one of the challenging areas of time series modeling. Box-Jenkins or ARIMA model is one of the popular linear time series models and can be applied in different domain of agriculture to forecast time series dataset showing linear pattern.

4. Suggested Readings

- Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). Time series analysis: Forecasting and control (3rd ed.). *Prentice Hall*, New Delhi.
- Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, **26(3)**, 1–22.
- Makridakis, S., Wheelwright, S. C., Hyndman, R. J. (1998). Forecasting methods and applications, (third edition,). *Wiley*, New York.

ARIMA-Intervention Model

Mrinmoy Ray, Ramasubramanian V., K N Singh and Kanchan Sinha
ICAR-IASRI, New Delhi

1. **Introduction:**

Various statistical approaches are in the literature for time series analysis, modelling and forecasting in different domains. They can be used more easily for forecasting purposes because historical sequences of observations on study variables are readily available at equally spaced intervals over discrete points of time. These successive observations are statistically dependent and time series modeling is concerned with techniques for the analysis of such dependencies. Thus in time series modeling, the prediction of values for the future periods is based on the pattern of past values of the variable under study, but not generally on explanatory variables which may affect the system. There are two main reasons for resorting to time series models. First, the system may not be understood, and even if it is understood it may be extremely difficult to measure the cause and effect relationship, second, many a time, collection of information on causal factors affecting the study variable(s) may be cumbersome /impossible and hence availability of long series data on explanatory variables is a problem. In such situations, the time series models are a boon to forecasters.

Every time series modelling approach has its own advantages and limitations. Time series intervention models have advantages in certain situations. Sometimes, it may be known that certain exceptional external events called 'interventions' could affect the time series under study. Under such circumstances, 'transfer function' models may be used to account for the effects of the intervention event on the series but wherein the input series (apart from the main variable) will be in the form of a simple indicator variable to indicate the presence or absence of the event. Here transfer function modeling refers to accounting for the dynamic relationship between two time series Y_t and X_t (the latter is the input series) wherein past values of both series may be used in forecasting Y_t , leading to a considerable reduction in the errors of the forecast.

Intervention analysis or event study is used to assess the impact of a special event on the time series of interest. Alternatively, intervention analysis may be undertaken to adjust for any unusual values in the series Y_t that might have resulted as a consequence of the intervention event. This will ensure that the results of the time series analysis of the

series, such as the structure of the fitted model, estimates of model parameters, and forecasts of future values, are not seriously distorted by the influence of these unusual values. In agriculture, intervention may occur due to introduction of new variety, new environmental regulations, economic policy changes, strikes, special promotion campaigns, natural disaster etc. There are broadly three kinds of intervention viz., step, pulse and ramp.

In its simplest form, intervention analysis itself may be regarded as a generalization of the two-sample problem (corresponding to pre and post intervention periods) to the case where the error or noise term is autocorrelated rather than independent. It is well known that the usual two-sample procedures are not robust against alternatives involving autocorrelation. Moreover, in many intervention analysis applications, time series data may be expensive or otherwise difficult to collect. In such cases, ‘power functions’ are helpful, because they can be used to determine the probability that a proposed intervention analysis application will detect a meaningful change. Power is the statistical term used for the probability that a test will reject the null hypothesis of no change at a given level of significance for a prescribed change. McLeod and Vingilis (2005) have suggested power computation methods for use with time series analyses for the certain cases of intervention analysis. They have also showed that power function also helps to compute the sample size required for intervention analysis.

2. Time Series Intervention Model:

Intervention model is a model in time series which is used to explore the impact on the series from external factors. Suppose that Y_t is a time series the ARIMA (p, d, q) model is written as:

$$\nabla^d Y_t = \phi_1 \nabla^d Y_{t-1} + \dots + \phi_p \nabla^d Y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.1)$$

where

ϕ = autoregressive parameter

θ = moving average parameter

d = degree of differencing

B = backshift operator

ε_t =white noise

Let,

$$\phi(B) = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)$$

$$\theta(B) = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$$

Now the ARIMA model can be written as-

$$\nabla^d Y_t = \frac{\theta(B)}{\phi(B)} \varepsilon_t$$

(2.2)

Sometimes the time series is depends on season i.e. Y_t depends Y_{t-s} , Y_{t-2s} etc. where s is the length of periodicity. Linearly this relation may be represented as-

$$\nabla^D Y_t = \Phi_s Y_{t-s} + \dots + \Phi_{P_s} Y_{t-P_s} - \Theta_s \varepsilon_{t-1} - \dots - \Theta_{Q_s} \varepsilon_{t-Q} + \varepsilon_t$$

(2.3)

where

Φ =seasonal autoregressive parameter

Θ = seasonal moving average parameter

D= degree of seasonal differencing i.e.

Let,

$$\Phi(B^s) = (1 - \Phi_1 B^s - \Phi_2 B^{2s} - \dots - \Phi_p B^{ps})$$

$$\Theta(B) = (1 - \Theta_1 B - \Theta_2 B^2 - \dots - \Theta_q B^q)$$

D=degree of seasonal differencing

Now the model can be written as-

$$\nabla_s^D Y_t = \frac{\Theta(B)}{\Phi(B)} \varepsilon_t$$

(2.4)

The ‘seasonal non-seasonal multiplicative’ model with the above notation can be written as-

$$\nabla^d \nabla_s^D Y_t = \frac{\theta(B)\Theta(B^s)}{\phi(B)\Phi(B^s)} \varepsilon_t$$

(2.5)

The time series input-output model is of the form

$$Y_t = \delta_1 Y_{t-1} + \dots + \delta_r Y_{t-r} + \omega_0 X_{t-b} - \omega_1 X_{t-b-1} - \dots - \omega_s X_{t-b-s} + N_t$$

(2.6)

where,

X_t =exogenous variable

b =delay parameter

N_t =

Let,

$$\delta(B) = (1 - \delta_1 B - \dots - \delta_r B^r)$$

$$\omega(B) = (\omega_0 - \omega_1 B - \dots - \omega_s B^s)$$

Then the input-output model can be written as

$$Y_t = \frac{\omega(B)}{\delta(B)} B^b X_t + N_t$$

(2.8)

When N_t is generated as an ARIMA process then the input-output model is known as transfer function model. The ARIMA model can be extended to Seasonal ARIMA model in a straightforward manner as shown above.

Intervention models are special cases of transfer function modeling in which the exogenous variable is a deterministic categorical variable. Accordingly, an intervention model with Seasonal ARIMA process can be written as

$$Y_t = \frac{\omega(B)}{\delta(B)} B^b I_t + \frac{\theta(B)\Theta(B^s)}{\phi(B)\Phi(B^s)} \varepsilon_t$$

(2.9)

where

I_t =dummy variable

The term $\frac{\omega(B)}{\delta(B)}B^b$ is called the intervention component, and the model may be extended to include several intervention components and thereby to account for several types of interventions that influence the process.

3. Input variables:

In general, there are three types of functions in the intervention variable. The intervention type of step function starts from a given time till the last time period. Mathematically, the intervention type of step function is written as:

$$I_t = \begin{cases} 0 & t \neq T \\ 1 & t \geq T \end{cases} \quad (3.1)$$

with T is time of intervention when it first occurred.

The pulse function is the intervention type occurs only particular period of time. Mathematically, the intervention type of pulse function is usually written as:

$$I_t = \begin{cases} 0 & t \neq T \\ 1 & t = T \end{cases} \quad (3.2)$$

Apart from pulse and step functions, there is another kind of function known as ramp function. Mathematically, the intervention type of ramp function is usually written as:

$$I_t = \begin{cases} 0 & t \neq T \\ t - T & t > T \end{cases} \quad (3.3)$$

To fix ideas, the illustration is given with the help of pulse function.

Indicator coding for pulse, step and ramp functions are given in Table 3.1.

Table3.1 Values of Intervention variable under different functions
(when it occurred in the 6th time point)

Time t	Pulse function I_t	Step function I_t	Ramp function I_t
--------	----------------------	---------------------	---------------------

1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	1	1	1
7	0	1	2
8	0	1	3
9	0	1	4
10	0	1	5

4. Graphical representation of input and various types of outputs

The input variable step and pulse function can be graphically represented in below in Fig-4.1 and Fig-4.2 respectively.



Fig-4.1



Fig-4.2

Output

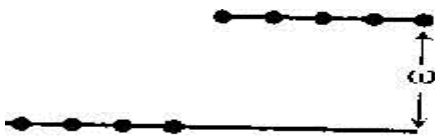


Fig-4.3

$$\omega B S_i^{(T)}$$

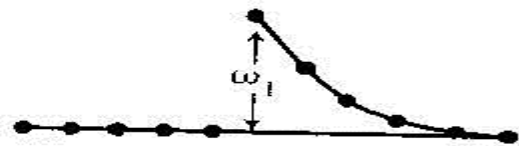


Fig-4.4

$$\frac{\omega_1 B}{1 - \delta B} P_i^{(T)}$$

In Fig-4.3 step intervention has occurred output pattern can be described by single parameter ω and the response of the input abrupt-permanent and in Fig-4.4 pulse intervention has occurred and the output pattern can be described by two parameters ω_1 and δ and the response due to intervention is abrupt-temporary.

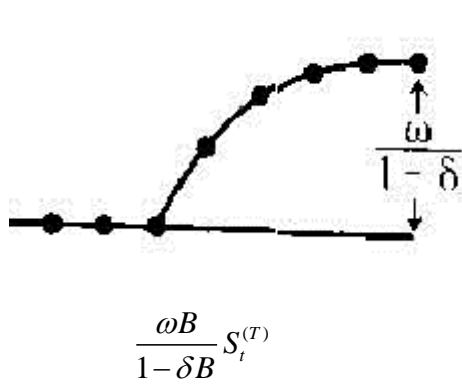


Fig-4.5

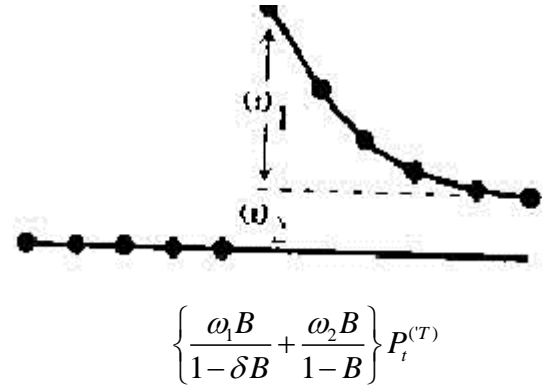


Fig-4.6

In Fig-4.5 step intervention has occurred and the response type is Gradual permanent and in Fig-4.6 pulse intervention has occurred and the response of the input can be described by two intervention component one component for explaining the abrupt temporary effect and other explaining the permanent effect.

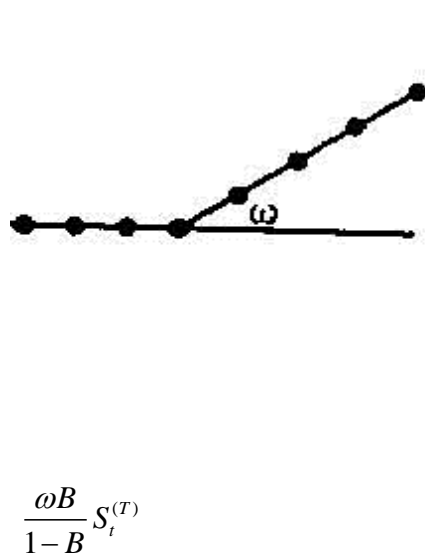


Fig-4.7

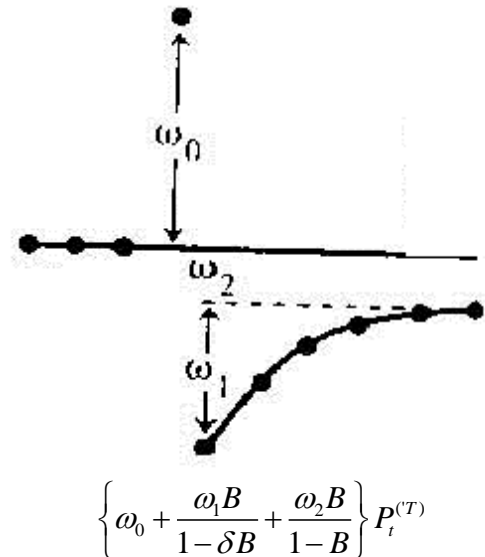


Fig-4.8

In Fig-4.7 step intervention has occurred and the response type is gradual increasing and in Fig-4.8 pulse intervention has occurred and the response type can be explained by three different intervention component.

5. Intervention model of null order pulse function:

The Intervention Type Null Order Pulse Function can be written as:

$$Y_t = \omega I_t + N_t$$

(5.1)

With:

Y_t = Response variable at t

ω = Intervention effect on Y_t

I_t = Intervention variable

N_t = ARIMA in pre- intervention data period

In Eq. (5.1), the effect of I on Y is assumed to have an intervention element. The estimated value of ω can be used to estimate the difference between the two periods before and after the occurrence of intervention. In general, the effect of I on Y are categorized as temporary, gradual, permanent or after delay in certain time.

6. Intervention model of first order pulse function:

Null order pulse function occurs when the δ parameter is zero. An alternative approach which accommodates another sort of such an effect as gradual is denoted as first order pulse function. If the intervention component is denoted as

$$Y_t^* = Y_t - N_t$$

(6.1)

then an additional parameter is needed to define $f(I_t)$ as follows

$$Y_t^* = \frac{\omega}{1 - \delta B} I_t$$

(6.2)

such that $-1 < \delta < 1$

After simplifying, the Eq. (vii) can be written as:

$$Y_t^* = \delta Y_{t-1}^* + \omega I_t$$

(6.3) since $Y_{t-1}^* = \delta Y_{t-2}^* + \omega I_{t-1}$ and $|\delta| < 1$, and so on recursively, the equation

becomes:

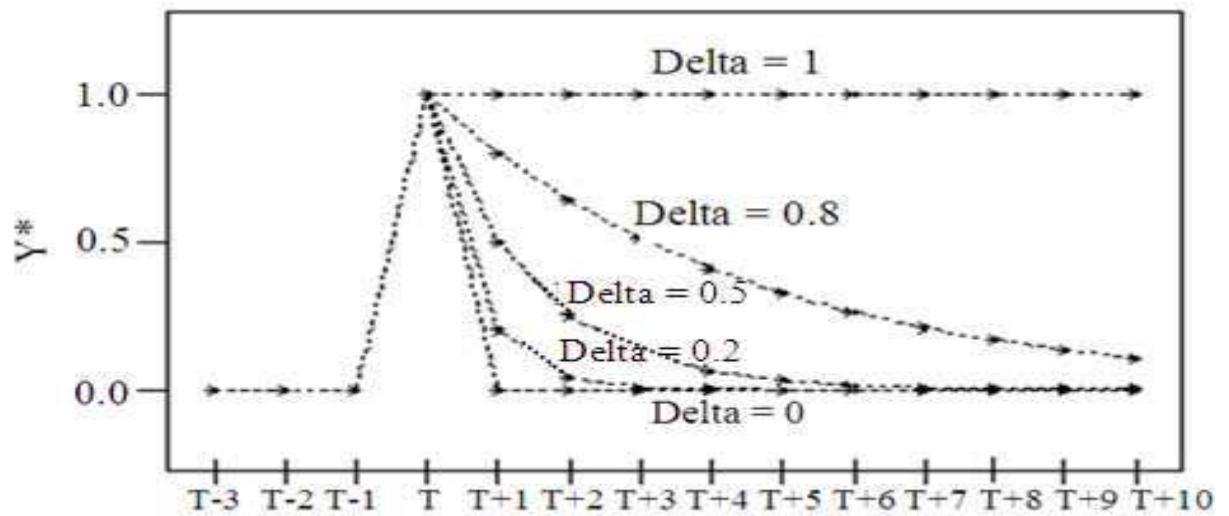
$$Y_t^* = \omega \sum_{j=0}^{\infty} \delta^j I_{t-j}$$

(6.4)

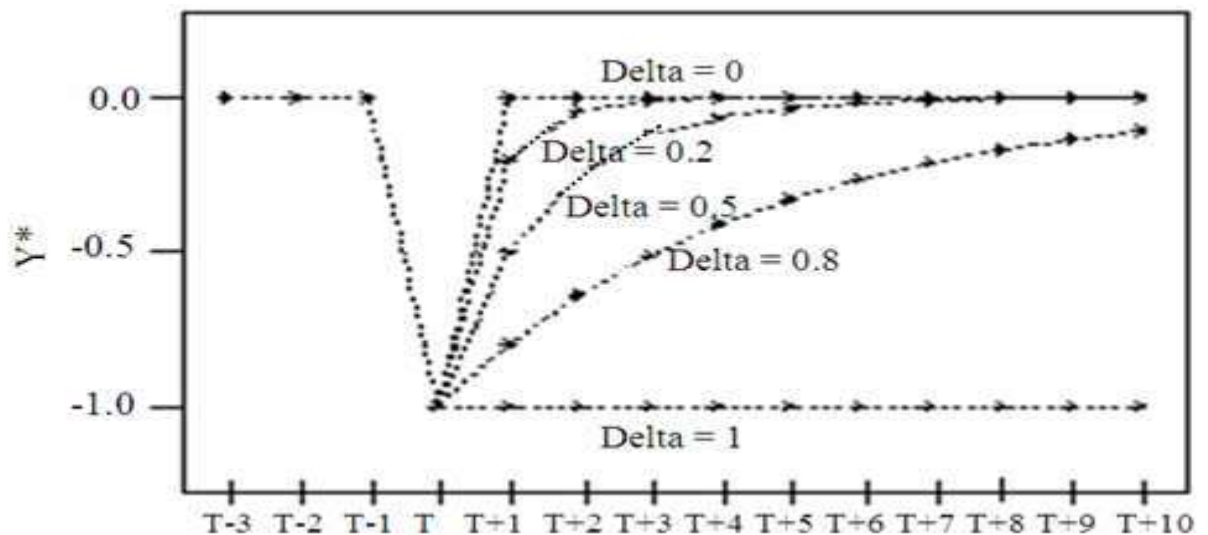
If Eq. (6.4) is applied for k ($k = 0, 1, 2, \dots$) periods after intervention, the equation below can be obtained:

$$\begin{aligned} Y_{T+k}^* &= \omega(I_{T+k} + \delta I_{T+k-1} + \delta^2 I_{T+k-2} + \dots + \delta^k I_{T+k-k} + \delta^{k+1} I_{T+k-(k+1)} + \dots) \\ &= \omega(0+0+0+\dots+0+\delta^k 1+0+0+\dots) \\ &= \omega \delta^k \end{aligned}$$

(6.5)



(a)



(b)

Fig-6.1 Intervention response with single pulse occurred in $t = T$

Equation (ix) means that the effect from pulse will vanish gradually corresponding to geometric sequence which is determined by δ value. Figure 1(a) shows Y_t^* value for model with value $\omega = 1$, Fig. 1(b) for value $\omega = -1$ and single pulse occurred in $t = T$ for some δ (delta) value with $|\delta| < 1$ and $\delta \neq 0$. From Fig. 1, it can also be seen that Y_t^* value approaches to zero asymptotically. In this case, as δ approaches to 0, then the effect of I to Y will be for a minimum period. On the other hand, as δ approaches to 1, the effect of I will last longer. For the extreme case of $\delta = 1$, we can get the permanent effect of I to Y which is shown in Fig. 1.

7. Procedures for intervention model:

The intervention response which is written as Y_t^* in essence are values of the differences (errors) between the original data after the intervention period and the corresponding ARIMA model forecasts of the same fitted on the pre-intervention data period.

The time series data Y_t are divided into Data I data time series period before the intervention occurred, Y_{1t} and Data II-data time series period after the intervention occurred, Y_{2t} :

Step I: The forming of ARIMA model for data time series period before intervention, Y_{1t} . The model building with respect to identification of autoregressive and moving average parameters will be done in the usual way based on the Autocorrelation Function (ACF) and the partial autocorrelation function (PACF).

Step II: In order to identification of the order of b, r and s cross-correlation function is used in case of transfer function model but in case of intervention model cross-correlation function cannot be used because the intervention variable are not continuous, cross correlations between the intervention variable and the dependent variable are meaningless the only way to identify the intervention model is impulse response function which is given below-

Table-6.1

$\frac{\omega(B)}{\delta(B)}$	Impulse response																						
ω_0	<table border="1"> <caption>Impulse response for ω_0</caption> <thead> <tr> <th>Time</th> <th>Value</th> </tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>0</td></tr> <tr><td>3</td><td>0</td></tr> <tr><td>4</td><td>0</td></tr> <tr><td>5</td><td>0</td></tr> <tr><td>6</td><td>0</td></tr> <tr><td>7</td><td>0</td></tr> <tr><td>8</td><td>0</td></tr> <tr><td>9</td><td>0</td></tr> <tr><td>10</td><td>0</td></tr> </tbody> </table>	Time	Value	1	10	2	0	3	0	4	0	5	0	6	0	7	0	8	0	9	0	10	0
Time	Value																						
1	10																						
2	0																						
3	0																						
4	0																						
5	0																						
6	0																						
7	0																						
8	0																						
9	0																						
10	0																						
$\omega_0 + \omega_1 B$	<table border="1"> <caption>Impulse response for $\omega_0 + \omega_1 B$</caption> <thead> <tr> <th>Time</th> <th>Value</th> </tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>9</td></tr> <tr><td>3</td><td>0</td></tr> <tr><td>4</td><td>0</td></tr> <tr><td>5</td><td>0</td></tr> <tr><td>6</td><td>0</td></tr> <tr><td>7</td><td>0</td></tr> <tr><td>8</td><td>0</td></tr> <tr><td>9</td><td>0</td></tr> <tr><td>10</td><td>0</td></tr> </tbody> </table>	Time	Value	1	10	2	9	3	0	4	0	5	0	6	0	7	0	8	0	9	0	10	0
Time	Value																						
1	10																						
2	9																						
3	0																						
4	0																						
5	0																						
6	0																						
7	0																						
8	0																						
9	0																						
10	0																						
$\frac{\omega_0}{1 + \delta B}$	<table border="1"> <caption>Impulse response for $\frac{\omega_0}{1 + \delta B}$</caption> <thead> <tr> <th>Time</th> <th>Value</th> </tr> </thead> <tbody> <tr><td>1</td><td>10</td></tr> <tr><td>2</td><td>9</td></tr> <tr><td>3</td><td>8</td></tr> <tr><td>4</td><td>7</td></tr> <tr><td>5</td><td>6</td></tr> <tr><td>6</td><td>5</td></tr> <tr><td>7</td><td>4</td></tr> <tr><td>8</td><td>3</td></tr> <tr><td>9</td><td>2</td></tr> <tr><td>10</td><td>1</td></tr> </tbody> </table>	Time	Value	1	10	2	9	3	8	4	7	5	6	6	5	7	4	8	3	9	2	10	1
Time	Value																						
1	10																						
2	9																						
3	8																						
4	7																						
5	6																						
6	5																						
7	4																						
8	3																						
9	2																						
10	1																						
$\frac{\omega_0 + \omega_1 B}{1 + \delta B}$	<table border="1"> <caption>Impulse response for $\frac{\omega_0 + \omega_1 B}{1 + \delta B}$</caption> <thead> <tr> <th>Time</th> <th>Value</th> </tr> </thead> <tbody> <tr><td>1</td><td>9</td></tr> <tr><td>2</td><td>10</td></tr> <tr><td>3</td><td>8</td></tr> <tr><td>4</td><td>7</td></tr> <tr><td>5</td><td>6</td></tr> <tr><td>6</td><td>5</td></tr> <tr><td>7</td><td>4</td></tr> <tr><td>8</td><td>3</td></tr> <tr><td>9</td><td>2</td></tr> <tr><td>10</td><td>1</td></tr> </tbody> </table>	Time	Value	1	9	2	10	3	8	4	7	5	6	6	5	7	4	8	3	9	2	10	1
Time	Value																						
1	9																						
2	10																						
3	8																						
4	7																						
5	6																						
6	5																						
7	4																						
8	3																						
9	2																						
10	1																						

Step III: The parameter estimates of intervention model will be tested by using statistical tests (such as t or z-tests, the latter will be elaborated in a subsequent section). Model suitability will be diagnosed through residual assumption test for adherence to white noise.

8. Illustration I:

Ismail *et al.* (2009) considered a real data of five star hotels' occupancy in Bali city of South East Asian nations from January 1997 until September 2005. Figure 8.1 shows the pattern of hotel occupancy which is relatively stable since 1997 period until an extreme decrease in occupancy immediately after the first Bali bomb in October 2002.

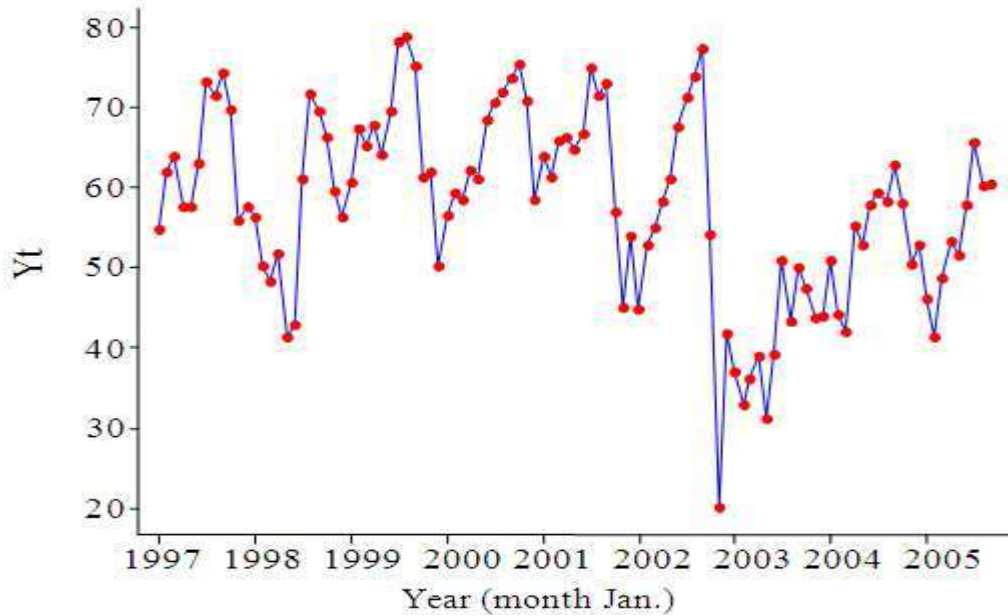


Fig-8.1

Firstly, they divided the data into two different parts which correspond to preintervention and postintervention data. After that they have fitted the ARIMA model in the Preintervention data and forecast until 2005.

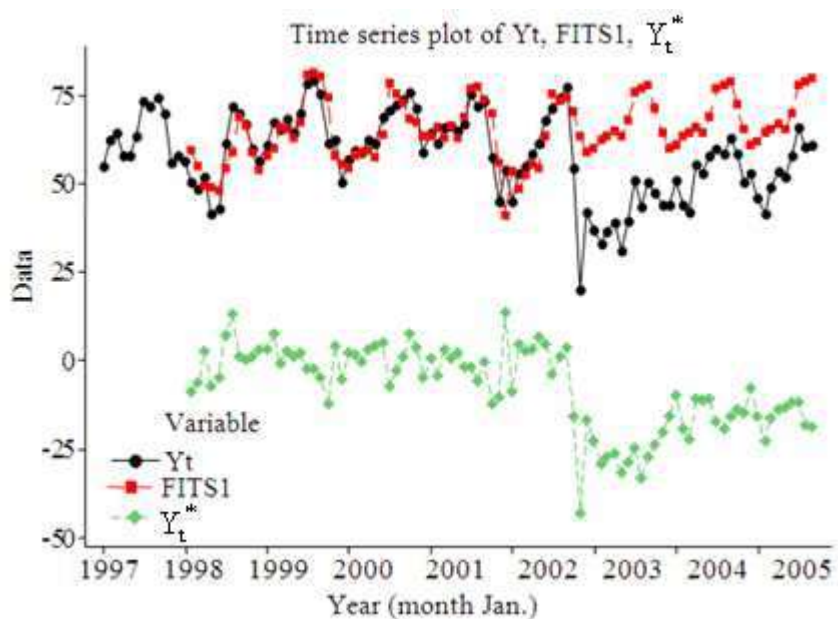


Fig-8.2

Figure -8.2 shows that the model is of good fit upto preintervention period but there is a huge gap between the forecast value and the actual value in the postintervention period which reveals that the first Bali's bomb has directly effected a reduction in the hotel occupancy levels. From the impulse response function they have found two intervention components with first component associated with $b=0$, $s=2$ and $r=1$ intervention and the second component associated with $b = 7$, $s = 0$ and $r = 1$.

Combining these two components, the intervention model which is appropriate is written as:

$$Y_t^* = \left(\frac{\omega_0 - \omega_1 B - \omega_2 B^2}{1 - \delta_1 B} + \frac{\omega_3 B^7}{1 - \delta_2 B} \right) I_t$$

(7.1)

SAS output of the estimated parameters has been reported in their paper as under:

Table-8.1

Conditional least squares							
Par.	Estimate	Error	t-value	Pr> t	Lag	variable	shift
NUM1	-15.6665	5.4939	-2.85	0.0056	0	It	0
NUM1,1	26.93222	5.6068	4.8	<.0001	1	It	0
NUM1,2	-19.7596	5.7184	-3.46	0.0009	2	It	0
DEN1,1	0.99622	0.05397	18.46	<.0001	1	It	0
NUM2	-8.73648	5.10589	-1.71	0.0911	0	It	7
DEN1,1	0.75165	0.40471	1.86	0.0671	1	It	7

Thus after estimation of parameters the model has been obtained as

$$Y_t^* = \left(\frac{-15.67 - 26.93B + 19.76B^2}{1 - 0.996B} + \frac{-8.74B^7}{1 - 0.752B} \right) I_t$$

(8.2)

One of the interesting results is that estimated value for δ is 0.996 which is near to 1. From this, it can be interpreted that the effect of pulse function is permanent.

Figure 8.3 shows the predicted values for the hotel occupancy levels using the estimated intervention model. The graph shows that the intervention model is of good fit.

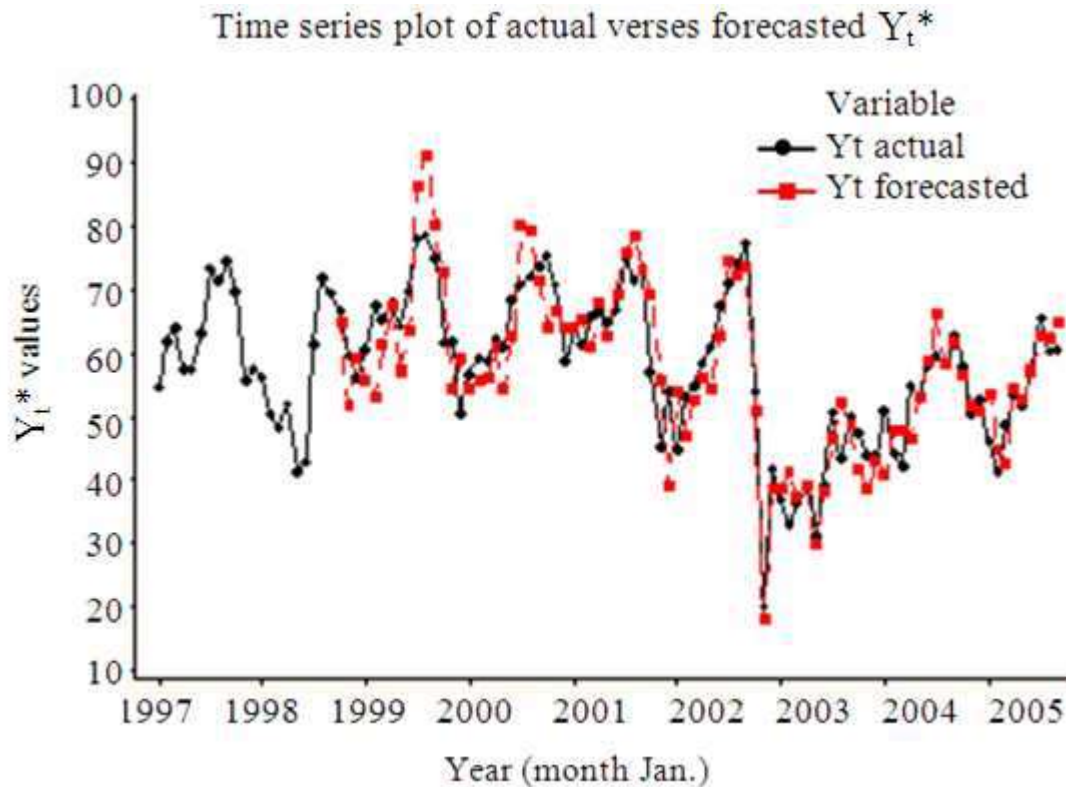


Fig-8.3

Intervention model in the Eq. (8.2) shows that pulse function intervention of the first Bali bomb that occurred in October 12, 2002 directly effected hotel occupancy levels. This effect continues to fluctuate which finally on gradually became a permanent effect until last observation i.e September 2005.

9. Concluding remarks

Intervention analysis or event study is used to assess the impact of a special event on the time series of interest. The time series intervention analysis and modelling is outlined considering three chief intervention input functions viz., pulse, step and ramp. Thereafter, the pulse function is taken particularly for explaining the functional forms of the orders that they can take. The illustration taken from Ismail *et al.* (2009) relating to monthly data of five star hotels' occupancy in Bali city and the impact of the occurrence

of bombing in October, 2002 shows that the Intervention model is more appropriate for forecasting as compared to the conventional ARIMA model.

References:

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994), *Time Series Analysis: Forecasting and Control (3rd ed.)*, San Francisco: Holden-Day.

Box, G.E.P., and Tiao,G.C. (1975).Intervention Analysis with Application to Economic and Environment Problems. *Journal of the American Statistical Associations*, **70**.70-79

Ismail, Z., Suhartono., Yahaya, A., and Efendi, R.(2009). Intervention Model for Analyzing the Impact of Terrorism to Tourism Industry. *J. Math. & Stat.*,**4**. 322-329.

Mcleod, A. I., and Vingilis, E. R.(2005). Power computations for intervention analysis. *Technometrics*,**47**.174-181.

Mcleod, A.I., and Vingilis, E. R.(2008). Power computations in time series analyses for traffic safety intervention. *Accident Analysis and Prevention*, **40**. 1244-1248.

Ray, M., Rai, A., Singh, K. N., V., Ramasubramanian and Kumar, A. (2017). Technology forecasting using time series intervention based trend impact analysis for wheat yield scenario in India. *Technological Forecasting and Social Change*, **118**, 128-133.

An Overview of Cointegration Analysis

Kanchan Sinha, K.N. Singh, Mrinmoy Ray and Achal Lama
ICAR-IASRI, New Delhi

1. Introduction

In econometrics, cointegration analysis is used to estimate and test stationary linear relations, or cointegration relations, between non-stationary time series variables such as consumption and income, interest rates at different maturities, and stock prices. The significance of cointegration analysis is its intuitive appeal for dealing with difficulties that arise when using non-stationary series, particularly those that are assumed to have a long-run equilibrium relationship. Cointegration is a statistical property possessed by time series variables that is defined by the concepts of stationarity and the order of integration of the series. A time series is said to be stationary if its mean and variance are constant over time and the value of covariance between two time periods depends only on the distance or gap or lag between the two time periods and not on the actual time at which the covariance is computed. A non-stationary time series data will have time varying mean or variance or both. In econometric analysis, most often the interest variables are non-stationary in nature, therefore econometric models with nonstationary stochastic variables should be formulated in such a way that the results will provide valid and meaningful economic and statistical interpretation. When the observed time series presents trend, differencing is often applied to the data to remove the trend before a model can be applied. The order of integration of a series is given by the number of times the series must be differenced in order to produce a stationary series. A series generated by the first difference is integrated of order 1 denoted as $I(1)$ while a time series as $I(0)$, is stationary in nature.

In many economic and financial market, cointegration plays an important role in terms of the strength and speed of price transmission between markets across various regions of a country. The degree to which consumers and producers can benefit, depends on how domestic markets are integrated with world markets and how integrated the different regional markets are with each other. With high integration among markets, low barriers to trade (price are similar) and increases fluidity between markets while with low integration, high barriers to trade causing price fluctuation between them. Earlier, the price correlation coefficients and regression analysis were used to explore whether or not markets were connected by price changes. However, price correlation coefficients can be misleading due to the existence of trends or unit roots in the data. The regression analysis

in measuring market integration was customized using the time series variables in their first difference order, but this caused the loss of long-run information. Cointegration analysis, on the other hand, allows eliminating the presence of unit roots and permits to stay away from specious results, thus enhancing the accuracy of research findings. Cointegration implies for Granger causality between the variables, meaning that if two markets are integrated, the price in one market, would commonly be found to Granger-cause the price in the other market, and/or vice versa. Therefore, Granger causality provides additional evidence as to whether and in which direction price transmission is occurring between two markets.

2. Model Specification

2.1 Vector Autoregressive(VAR) process

A VAR is a simple extension of the $AR(P)$ framework and is given by:

$$Y_t = \delta + A_1 Y_{t-1} + A_2 Y_{t-2} + \dots + A_k Y_{t-k} + u_t \quad (i)$$

where, $u_t \sim IN(0, \Sigma)$

where, $Y_t = (Y_{1t}, Y_{2t}, \dots, Y_{nt})'$ is $(n \times 1)$ random vector of endogenous variables, each of the A_i is an $(n \times n)$ matrix of parameters, δ is a fixed $(n \times 1)$ vector of intercept terms.

Finally,

$u_t = (u_{1t}, u_{2t}, \dots, u_{nt})'$ is a n -dimensional white noise or innovation process, i.e., $E(u_t) = 0$, $E(u_t, u_t') = \Sigma$ and $E(u_t, u_s') = 0$ for $s \neq t$. The covariance matrix Σ is assumed to be non-singular.

2.2 Cointegration process

Cointegration analysis is used to examine whether long-run equilibrium relationships exist between two or more series. The long-run relationship is given as:

$$P_t^1 = \alpha_0 + \alpha_1 P_t^2 + \varepsilon_t \quad (ii)$$

where P_t^1 and P_t^2 is the price of a commodity in two different markets (say). If ε_t is stationary, then market prices are said to be cointegrated. The cointegration analysis reflects the long-run movement of price series, although in the short run they may drift apart. Johansen's multivariate cointegration approach is used to examine cointegration between two price series. Before conducting cointegration test, it is mandatory to perform stationarity test. Augmented Dickey-Fuller (ADF) test is performed to check stationarity for the series. The ADF test is based on the regression of original price series including the

intercept, trend, regression of first difference series and lags of the differenced series. The variables that are integrated of the same order may be cointegrated, and the unit root test finds out which variables are integrated of same order, for example; if integrated by order one then it is denoted as $I(1)$ and if integrated of order p then it is denoted as $I(p)$. The ADF unit root test equation can be expressed as follows:

$$\Delta y_t = \beta_1 + \beta_2 t + \delta y_{t-1} + \sum_{i=1}^m \alpha_i \Delta y_{t-i} + \varepsilon_t \quad (\text{iii})$$

where Δy_t is a vector to be tested for cointegration, t is time or trend variable. Δy_t is the first difference i.e., ($\Delta y_t = y_t - y_{t-1}$), ε_t is a white noise term. The null hypothesis that, $H_0: \delta = 0$; signifying unit root, states that the time series is non-stationary while the alternative hypothesis, $H_1: \delta < 0$, signifies that the time series is stationary, thereby rejected the null hypothesis. Since ADF tests tell us whether a time series is integrated or not, therefore the test is known as a ‘‘Test for integration’’.

2.3 Johansen’s Cointegration Tests

A cointegrated system can be written as:

$$\Delta y_t = \sum_{i=1}^k \Gamma_i \Delta y_{t-i} + \alpha \beta' y_{t-k} + \varepsilon_t \quad (\text{iv})$$

where y_t is the price series, Δy_t is the first difference i.e., ($\Delta y_t = y_t - y_{t-1}$), and the matrix $\alpha \beta'$ is $n \times n$ with rank ($0 \leq r \leq n$), which is the rank of linear independent cointegration relations in the vector space of matrix. The Johansen’s method of cointegrated system is a restricted maximum likelihood method with rank restriction on matrix $\Pi = \alpha \beta'$. The rank of Π can be obtained by using λ_{trace} or λ_{max} test statistics. The test statistics can be written as:

$$\lambda_{trace} = -T \sum_{i=r+1}^n \ln (1 - \hat{\lambda}_i) \quad \forall r = 0, 1, \dots, n - 1 \quad (\text{v})$$

where $\hat{\lambda}_i$ ’s are the Eigen values representing the strength of the correlation between the first difference part and the error-correction part. Now the following hypotheses are tested, under null hypothesis, $H_0: \text{rank of } \Pi = r$ and under alternative hypothesis, $H_1: \text{rank of } \Pi > r$. where r is the number of cointegration equations. The above test is carried under the condition of cointegrating equation has only intercept (no trend) and the original price series follows a trend since the mean and variance are non-constant over a period of time (non-stationary).

2.4. Granger Causality test

The time series being cointegrated, Granger causality test is carried out to examine the direction of causality. If two markets are integrated, then price in one market would commonly found to Granger cause the price in other market and/or vice versa. Granger causality provides additional evidence as to whether and in which direction price transmission is occurred between two series.

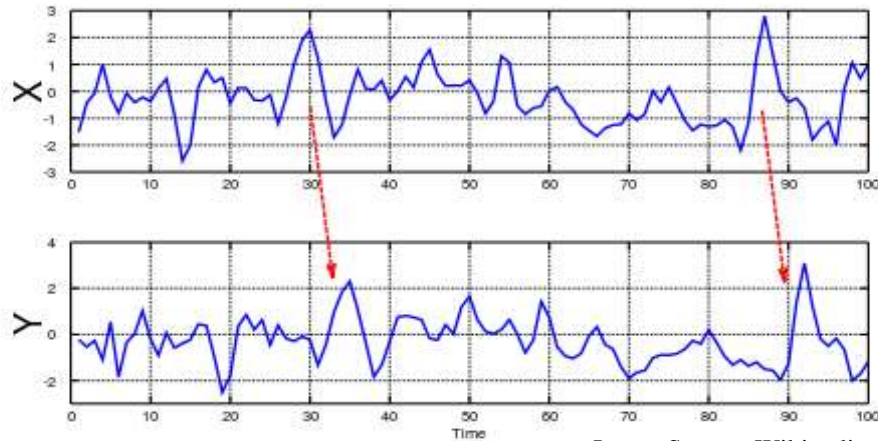


Image Source: Wikipedia

Figure: Time series X Granger-causes time series Y; the patterns in X are approximately repeated in Y after some time lag (two examples are indicated with arrows). Therefore, past values of X can be used for the prediction of future values of Y.

A VAR (2) model is applied in order to assess the causality of the price series.

$$\begin{pmatrix} y_t \\ x_t \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{pmatrix} y_{t-1} \\ x_{t-1} \end{pmatrix} + \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{pmatrix} y_{t-2} \\ x_{t-2} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix} \quad (\text{vi})$$

The matrix relation can be written in individual form as:

$$y_t = a + c_{11}y_{t-1} + c_{12}x_{t-1} + d_{11}y_{t-2} + d_{12}x_{t-2} + \varepsilon_{1t} \quad (\text{vii})$$

$$x_t = b + c_{21}y_{t-1} + c_{22}x_{t-1} + d_{21}y_{t-2} + d_{22}x_{t-2} + \varepsilon_{2t} \quad (\text{viii})$$

The restrictions imposed to test the causality can be described as:

lags of y do not explain the value of x so, $c_{21} = 0$ and $d_{21} = 0$

lags of x do not explain the value of y so, $c_{12} = 0$ and $d_{12} = 0$

Hence, the null hypothesis for Granger causality test is defined as:

$$H_0: c_{12} = d_{12} = 0 \text{ (} x_t \text{ does not Granger cause } y_t \text{)}$$

$$H_0: c_{21} = d_{21} = 0 \text{ (} y_t \text{ does not Granger cause } x_t \text{)}$$

2.5 Vector Error Correction Model (VECM)

If series are cointegrated a vector error correction model (VECM) is estimated that can be seen as a VAR model including a variable representing the deviations from the long-run equilibrium. VECM model has two distinct characteristics: first, an ECM is dynamic in the sense that it involves lags of the dependent and explanatory variables; it thus captures the short-run adjustments from past disequilibrium and contemporaneous changes in the explanatory variables to equilibrium. Second, the ECM is transparent in displaying the cointegrating relationship between or among the variables. Equation (ix) shows a VECM for three variables including a constant, the error correction term, lagged term and random error.

$$\begin{bmatrix} \Delta P_t^B \\ \Delta P_t^C \\ \Delta P_t^H \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} ECT_{-1} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} \Delta P_{t-1}^B \\ \Delta P_{t-1}^C \\ \Delta P_{t-1}^H \end{bmatrix} + \begin{bmatrix} \varepsilon_t^{PB} \\ \varepsilon_t^{PC} \\ \varepsilon_t^{PH} \end{bmatrix} \quad (ix)$$

In equation (ix), P_t^B , P_t^C and P_t^H represents time series datasets from three different markets. This VECM representation is particularly interesting as it allows estimating how the variables adjust deviations towards the long-run equilibrium. The error correction coefficient (a_i) reflects the speed of adjustment while the coefficients of lagged explanatory variables give an indication of short-run adjustments. The coefficient of ECT must be negative and significantly different from zero. The negative ECT implies that if there is a deviation from the existing and long-run equilibrium, there would be an adjustment back to long-run equilibrium in subsequent periods.

3. Conclusion

In a developing economy like India, market integration plays an important role concerning the formulation of proper agricultural marketing policies, determining marketing efficiency, production decisions of farmers and diversification to high value crops and many more. Here, the concept of cointegration and related techniques are briefly described. The type of study can help researchers and practitioners to identify the extent to which prices/information in different markets move together in their respective field of study.

4. Suggested Readings

Engle, R.F. and Granger, C.W. J. (1987) Cointegration and error correction: Representation, estimation and testing, *Econometrica*, **50**, 987-1007

Sinha, K, Paul, R.K. and Bhar, L. M. (2016) Price Transmission and Causality in major onion markets of India. *Journal of the Society for Application of Statistics in Agriculture and Allied Sciences (SASAA)*, **1(2)**, 35-40

Paul, R. K. and Sinha, K. (2015) Spatial market integration among major coffee markets in India, *Journal of the Indian Society of Agricultural Statistics*, **69 (3)**, 281-287

Sahu, P.K., Dey, S., Sinha, K. Singh, H. and Narsimaiaha, L. (2019) Cointegration and Price Discovery Mechanism of Major Spices in India, *American Journal of Applied Mathematics and Statistics*, **7(1)**, 18-24

Hierarchical Time Series Modeling

Soumen Pal
ICAR-IASRI, New Delhi

Time-series data collected in many situations are hierarchical in structure. These dataset generally contain information in clusters which can be combined into another series of interest. Here, the time series are aggregated along the hierarchy based on dimensional attributes such as location. Thus, a hierarchical time-series is a collection of several time-series data that are correlated in a hierarchical manner. By contrast, a collection of time-series that are aggregated in a number of non-hierarchical ways, are called a grouped time-series.

Hierarchical time-series comprises of several dataset maintaining certain hierarchical relationship among them. A cross-sectional hierarchical structure is an arrangement of items in which the items are ordered above, below or on the same level as others. For example, the national economic account is divided into production, income and outlay and capital transaction. Production is further disaggregated into production in India and production in the rest of world; income, outlay and capital transactions each further can be classified into persons, companies, public corporations, general government and rest of world. It is an example of hierarchical time-series since here the order of disaggregation is unique. In demographic forecasting, the infant mortality count in India can be grouped by gender; again, within each gender, mortality counts can be further classified according to geographic location, e.g. state. This second example is called a grouped time-series where the order of disaggregation is not unique. The infant mortality counts in India can also be first disaggregated by states and then by genders. Therefore, the order is not important.

There are certain specialized strategies viz. top-down, bottom-up, middle-out and optimal approaches which take care of predicting future values for such multilevel data. For forecasting of individual series at different levels of hierarchy, a method of aggregation or disaggregation is followed. Existing approaches to hierarchical time-series forecasting usually involve either a top-down method or a bottom-up method or a combination of both methods viz. middle-out approach. The top-down approach at first provides forecasting for

the aggregated series at the topmost level of the hierarchy, then disaggregating the forecasts in the lower levels based on historical or forecast properties. The bottom-up method follows the reverse approach i.e. forecasting each of the most disaggregated series at the lowest level of the hierarchy and then using aggregation to obtain the forecasts at higher levels of the hierarchy. Middle-out approach starts forecasting at an intermediate level of the hierarchy selected by the user and then following bottom-up approach for the upper level forecasting and top-down approach for the lower level forecasting. Hyndman *et al.* (2011) proposed a statistical method for optimal hierarchical forecasting as well.

Figure 1 illustrates a hierarchical structure with $H = 2$ level hierarchy. Level 0 is the *Total* series which is completely aggregated. Level 1 stands for the first level of disaggregation, and so on down to the bottom level H , which consists of the most disaggregated series. Let $Y_{J,t}$ be the t^{th} observation ($t = 1, 2, \dots, n$) of series Y_J which corresponds to node J on the hierarchical arrangement. As in Figure 1, a sequence of letters is used to represent the individual nodes. For example, $Y_{B,t}$ symbolizes the t^{th} observation of the series corresponding to node B at level 1, $Y_{CA,t}$ stands for the t^{th} observation of the series corresponding to node CA at level 2, and so on. Thus an individual node is represented by the actual letter sequence and the length of the letter sequence refers to the level. Y_t denotes the t^{th} observation for the total aggregate level. Let l_i represent the total number of series for level i and $l = l_0 + l_1 + \dots + l_H$ denotes the total number of series in the hierarchy. In this case, $l = 1 + 3 + 9 = 13$.

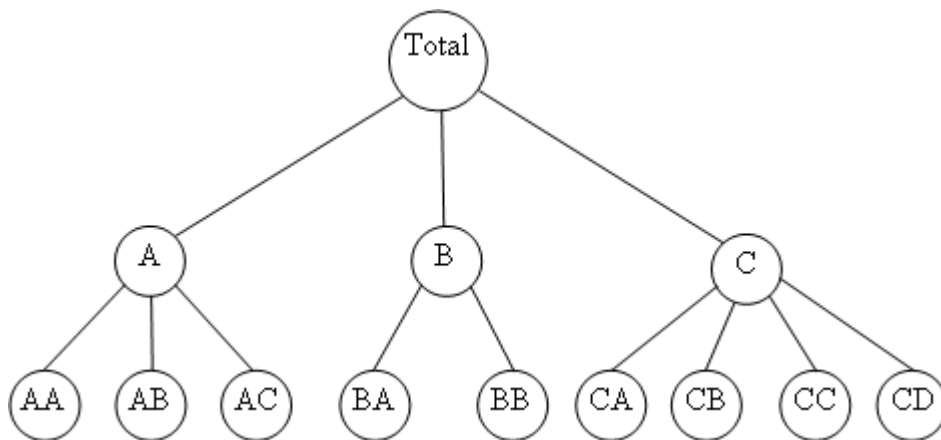


Fig 1 A two level hierarchical tree diagram

$$\tilde{\mathbf{Y}}_n(h) = \mathbf{SP}\hat{\mathbf{Y}}_n(h) \quad (2)$$

where \mathbf{S} is the $l \times l_H$ order summing matrix and \mathbf{P} is a matrix of order $l_H \times l$. Depending on the hierarchical approach, role of \mathbf{P} differs. Thus, linearly combining the independent base forecasts $\hat{\mathbf{Y}}_n(h)$, the final revised forecasts $\tilde{\mathbf{Y}}_n(h)$ can be produced by any hierarchical forecasting approach.

Under assumption that the base (independent) forecasts are unbiased, it can be written that

$$\mathbb{E}[\hat{\mathbf{Y}}_n(h)] = \mathbb{E}[\mathbf{Y}_n(h)]. \quad (3)$$

The necessary condition for the revised hierarchical forecasts to be unbiased is:

$$\mathbb{E}[\tilde{\mathbf{Y}}_n(h)] = \mathbb{E}[\mathbf{Y}_n(h)] = \mathbf{SE}[\mathbf{Y}_{H,n}(h)]. \quad (4)$$

If $\boldsymbol{\beta}_n(h) = \mathbb{E}[\mathbf{Y}_{H,n+h} | \mathbf{Y}_1, \dots, \mathbf{Y}_n]$ represents the mean of the forecast values of the bottom level H , then

$$\mathbb{E}[\tilde{\mathbf{Y}}_n(h)] = \mathbf{SPE}[\hat{\mathbf{Y}}_n(h)] = \mathbf{SPS}\boldsymbol{\beta}_n(h). \quad (5)$$

Therefore, the unbiasedness of the revised forecast will hold if and only if the following condition is satisfied:

$$\mathbf{SPS} = \mathbf{S}. \quad (6)$$

Again, if $\boldsymbol{\Sigma}_h$ is the variance of the base forecasts $\hat{\mathbf{Y}}_n(h)$ given by

$$\boldsymbol{\Sigma}_h = \text{Var}[\hat{\mathbf{Y}}_n(h) | \mathbf{Y}_1, \dots, \mathbf{Y}_n], \quad (7)$$

then Hyndman *et al.* (2011) showed that, for all of the methods that can be represented by (2), the variance of the revised forecasts is given by

$$\text{Var}[\tilde{\mathbf{Y}}_n(h) | \mathbf{Y}_1, \dots, \mathbf{Y}_n] = \mathbf{SP}\boldsymbol{\Sigma}_h\mathbf{P}'\mathbf{S}'. \quad (8)$$

Thus, prediction intervals on the revised forecasts can be obtained provided $\boldsymbol{\Sigma}_h$ can be reliably estimated. In the present paper, estimation of this covariance matrix has not been covered.

Bottom-up approach

The bottom-up method is one of the commonly used methods of hierarchical forecasting (Dangerfield and Morris 1992; Zellner and Tobias 2000; Espasa *et al.* 2002). The bottom-up method provides first independent base forecasts for most disaggregated series at the lowest level of the hierarchy and then aggregate these base forecasts upwards to obtain revised forecasts for rest of the series in the hierarchy. For the hierarchy of Figure 1, at first h -step-ahead base forecasts are generated for the bottom level series: $\hat{Y}_{AA,n}(h)$, $\hat{Y}_{AB,n}(h)$, $\hat{Y}_{AC,n}(h)$, $\hat{Y}_{BA,n}(h)$, $\hat{Y}_{BB,n}(h)$, $\hat{Y}_{CA,n}(h)$, $\hat{Y}_{CB,n}(h)$, $\hat{Y}_{CC,n}(h)$ and $\hat{Y}_{CD,n}(h)$. Aggregating these up the hierarchy we get h -step-ahead forecasts for the rest of the series: $\tilde{Y}_{A,n}(h) = \hat{Y}_{AA,n}(h) + \hat{Y}_{AB,n}(h) + \hat{Y}_{AC,n}(h)$, $\tilde{Y}_{B,n}(h) = \hat{Y}_{BA,n}(h) + \hat{Y}_{BB,n}(h)$, $\tilde{Y}_{C,n}(h) = \hat{Y}_{CA,n}(h) + \hat{Y}_{CB,n}(h) + \hat{Y}_{CC,n}(h) + \hat{Y}_{CD,n}(h)$ and $\tilde{Y}_n(h) = \tilde{Y}_{A,n}(h) + \tilde{Y}_{B,n}(h) + \tilde{Y}_{C,n}(h)$. Thus, in case of bottom-up approach, the revised forecasts for the bottom level series are equal to the base forecasts.

This can be represented by the general form of eq. (2), due to Athanasopoulos *et al.* (2009):

$$\mathbf{P} = [\mathbf{0}_{l_H \times (l-l_H)} \mid \mathbf{I}_{l_H}] \quad (9)$$

where $\mathbf{0}_{i \times j}$ is the $i \times j$ null matrix. Here, \mathbf{P} matrix extracts only bottom-level forecasts from $\hat{\mathbf{Y}}_n(h)$. These bottom-level forecasts are then aggregated by the summation matrix \mathbf{S} to produce the revised forecasts for the whole hierarchy. By using this approach, the property of unbiasedness in (6) is satisfied. In case of bottom-up approach, the revised forecasts for the bottom level series are equal to the base forecasts.

Top-down approach

Another commonly applied method in hierarchical forecasting is the top-down approach (Zotteri *et al.* 2005; Widiarta *et al.* 2007). This approach involves first generating base forecasts for the *Total* series on the top of the hierarchy and then disaggregating these downwards based on the proportions of the data. Once the bottom level forecasts have been generated, the summing matrix \mathbf{S} can be used to generate forecasts for the rest of the series in the hierarchy. It is evident that for top-down approaches the top level revised forecasts are

equal to the top level base forecasts. In terms of the general form of eq. (2), due to Athanasopoulos *et al.* 2009, it can be written that

$$\mathbf{P} = [\mathbf{p} \mid \mathbf{0}_{l_H \times (l-1)}] \quad (10)$$

where $\mathbf{p} = [p_1, p_2, \dots, p_{l_H}]'$ are a set of proportions for the bottom level series. So the role of \mathbf{P} here is to distribute the top level forecasts for forecasting the bottom level series. Let $\hat{Y}_{j,n}^{(i)}(h)$ be the h -step-ahead forecast of the series that corresponds to the node which is i levels above j . Also, let $\Sigma(\hat{Y}_{i,n}(h))$ be the sum of the h -step-ahead forecasts below node i which are directly connected to node i . For example in Figure 1, $\Sigma(\hat{Y}_{l,n}^{(2)}(h)) = \Sigma(\hat{Y}_{\text{Total},n}(h)) = \hat{Y}_{A,n}(h) + \hat{Y}_{B,n}(h) + \hat{Y}_{C,n}(h)$. Then, it can be written that,

$$p_j = \prod_{i=0}^{L-1} \frac{\hat{Y}_{j,n}^{(i)}(h)}{\Sigma(\hat{Y}_{j,n}^{(i+1)}(h))} \quad (11)$$

for $j = 1, 2, \dots, l_H$.

If $\hat{Y}_{\text{Total},n}(h)$ is generated for the top level series of the hierarchy in Figure 1, the revised final forecasts moving down the farthest middlemost branch of the hierarchy will be,

$$\tilde{Y}_{B,n}(h) = \left(\frac{\hat{Y}_{B,n}(h)}{\hat{Y}_{A,n}(h) + \hat{Y}_{B,n}(h) + \hat{Y}_{C,n}(h)} \right) \times \hat{Y}_{\text{Total},n}(h)$$

and

$$\tilde{Y}_{BB,n}(h) = \left(\frac{\hat{Y}_{BB,n}(h)}{\hat{Y}_{BA,n}(h) + \hat{Y}_{BB,n}(h)} \right) \times \tilde{Y}_{B,n}(h).$$

Hence,

$$\tilde{Y}_{BB,n}(h) = \left(\frac{\hat{Y}_{BB,n}(h)}{\hat{Y}_{BA,n}(h) + \hat{Y}_{BB,n}(h)} \right) \left(\frac{\hat{Y}_{B,n}(h)}{\hat{Y}_{A,n}(h) + \hat{Y}_{B,n}(h) + \hat{Y}_{C,n}(h)} \right) \times \hat{Y}_{\text{Total},n}(h).$$

Consequently,

$$p_5 = \left(\frac{\hat{Y}_{BB,n}(h)}{\hat{Y}_{BA,n}(h) + \hat{Y}_{BB,n}(h)} \right) \left(\frac{\hat{Y}_{B,n}(h)}{\hat{Y}_{A,n}(h) + \hat{Y}_{B,n}(h) + \hat{Y}_{C,n}(h)} \right).$$

In the similar way, other proportions are obtained. In top-down approaches the top level revised forecasts is equal to the top level base forecasts, i.e., $\tilde{y}_t(h) = \hat{y}_t(h)$.

Middle-out approach

In this approach, forecasts are produced at an intermediate level of hierarchy, and then disaggregated to obtain forecasts at lower levels and aggregated for higher level forecasts. Thus, the middle-out approach is a combination of bottom-up and top-down approaches. In practice, production houses apply this method to study demand forecasting (Lo *et al.* 2008). At first, base forecasts are produced for all the series of the selected middle level. Then, for the series above the middle level, revised forecasts are constructed using the bottom-up approach by aggregating the middle-level base forecasts upwards. For the series below the middle level, revised forecasts are generated using a top-down approach by disaggregating the middle level base forecasts downwards.

Optimal Combination approach

The optimal combination approach due to Hyndman *et al.* (2011), involves first generating independent base forecast for each series in the hierarchy and, provided the base forecasts are unbiased, produces unbiased revised forecasts which are consistent across the levels of the hierarchy. The h -step-ahead base forecasts can be written by the linear regression model as

$$\hat{Y}_n(h) = \mathbf{S}\boldsymbol{\beta}_h + \boldsymbol{\varepsilon}_h \quad (12)$$

where $\boldsymbol{\beta}_h = E(\hat{Y}_{H,n}(h) | \mathbf{Y}_1, \dots, \mathbf{Y}_n)$ is the unknown mean of the base forecasts of the bottom level H , and $\boldsymbol{\varepsilon}_h$ has zero mean and covariance matrix $\text{Var}[\boldsymbol{\varepsilon}_h] = \boldsymbol{\Sigma}_h$. Provided the $\boldsymbol{\Sigma}_h$ is known, the generalised least squares estimation procedure can be used to obtain the minimum variance unbiased estimate of $\boldsymbol{\beta}_h$. This can be written as

$$\hat{\boldsymbol{\beta}}_n(h) = (\mathbf{S}'\boldsymbol{\Sigma}_h^{-1}\mathbf{S})^{-1}\mathbf{S}'\boldsymbol{\Sigma}_h^{-1}\hat{Y}_n(h) \quad (13)$$

where Σ_h^\dagger is the Moore-Penrose generalized inverse of Σ_h . This leads to the following revised forecasts

$$\tilde{Y}_n(h) = S\hat{\beta}_n(h) = SP\hat{Y}_n(h) \quad (14)$$

where $P = (S'\Sigma_h^\dagger S)^{-1}S'\Sigma_h^\dagger$. This fulfils the unbiasedness property of (6).

Illustration:

Hierarchical time-series data of Sorghum production (in '000 tonne) in India is taken for illustration. The structure of the hierarchy is presented in Table 1.

Table 1 Hierarchical structure of Sorghum production data

Level	Total series per level
India	1
Season	2
States and Union Territories	19

For more details on this structure refer to Annexure-I.

For this hierarchical time-series data, modeling and forecasting have been done using *hts* package (Hyndman *et al.* 2013) in R software. Figure 2 and 3 illustrate the time-series data of sorghum production across all hierarchical levels. The bottom-most level i.e. level 2 exhibits time-series of seasonal sorghum production for each of the states, however, for clarity, (seasonal) production of some of the major growing states are only illustrated in figure 3. Level 1 shows production of sorghum in kharif and rabi seasons separately in India. Time-series of total sorghum production in India is displayed in level 0. For each of the graphical representation, *y*-axis represents production in '000 tonne and *x*-axis indicates the time period (year).

R Code:

```
dd = read.table(file.choose(),header=F) #read the data file.
aa = as.matrix(dd) #store the data as matrix.
colnames(aa) =
c("AA","AB","AC","AD","AE","AF","AG","AH","AI","AJ","AK","AL","AM","
BA","BB","BC","BD","BE","BF") #names of the bottom time series.
# 2 child nodes associated with level 1,
# which are followed by 13 and 6 sub-child nodes respectively at
level 2.
nodes = list(2,c(13,6))
abc = ts(aa,start=1963)
y = hts(abc,nodes)
plot(y,lwd=2)
```

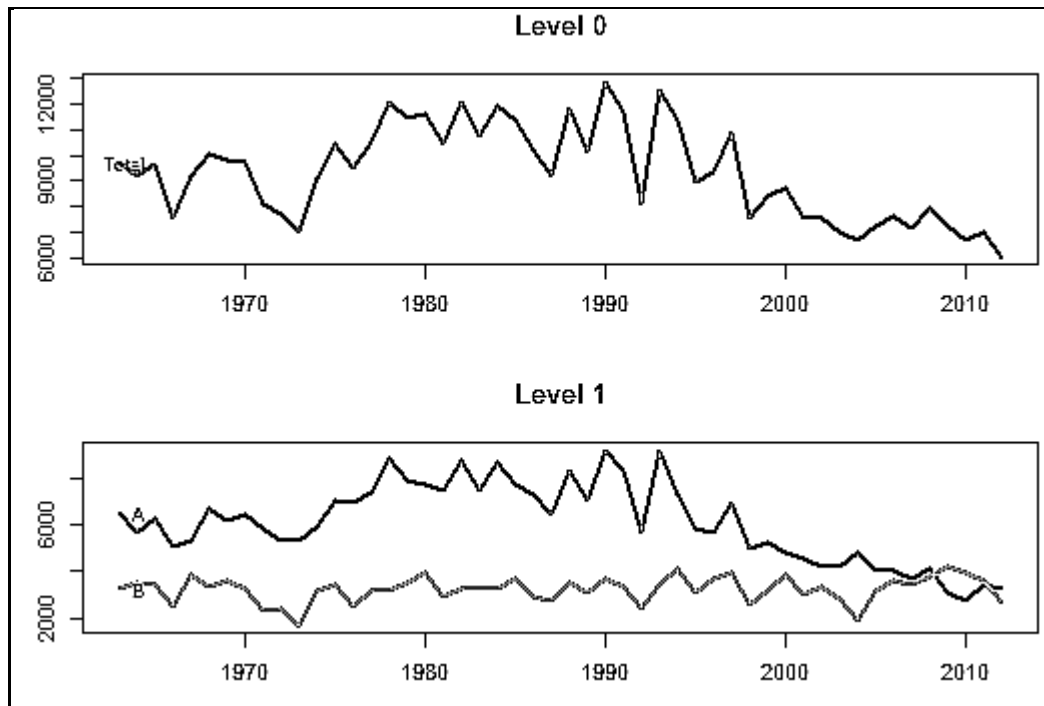


Fig 2 Hierarchical time-series of sorghum production at level 0 and 1

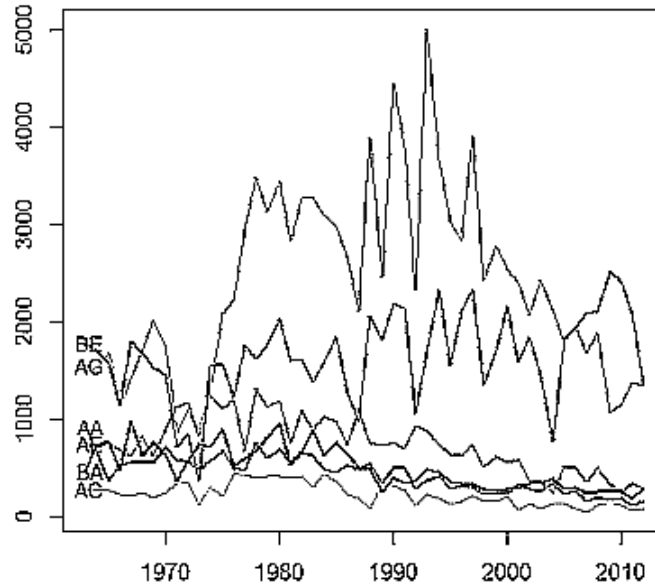


Fig 3 Hierarchical time-series of sorghum production at level 2 for selective states

```

data = window(y, start=1963, end=2006) # in-sample
test = window(y, start=2007, end=2012) # out-of-sample

# bottom-up
f1 = forecast(data, h=6, method="bu", fmethod="arima")
# out-of-sample accuracy measures for forecast
accuracy.gts(f1, test)
for (i in 1:6){
  print(mean(accuracy.gts(f1, test)[i,]))
}

# top-down forecast proportions ("tdfp")
f2 = forecast(data, h=6, method="tdfp", fmethod="arima")
# out-of-sample accuracy measures for forecast
accuracy.gts(f2, test)
for (i in 1:6){
  print(mean(accuracy.gts(f2, test)[i,]))
}

# middle-out
f3 = forecast(data, h=6, method="mo", fmethod="arima", level=1)
# out-of-sample accuracy measures for forecast
accuracy.gts(f3, test)
for (i in 1:6){

  print(mean(accuracy.gts(f3, test)[i,]))
}
# optimal combination
f4 = forecast(data, h=6, method="comb", fmethod="arima")

```

```
accuracy.gts(f4, test)
for (i in 1:6){
  print(mean(accuracy.gts(f4, test)[i,]))
}
```

It has been observed that the middle-out method produces best out-of-sample forecasts followed by top-down approach. As the middle-out method outperforms other methods for this particular dataset, the final forecasting of sorghum production for the year 2015 till 2017 has been produced by using this approach.

```
f.mo = forecast(y, h=5, method="mo", fmethod="arima", level=1)
all.ts = allts(f.mo)
for (i in 1:22)
{
  print(all.ts[,i])
}
plot(f.mo, include=10)
```

Table 4 depicts the forecasted values for all the series at each level.

Table 4 Forecasting of sorghum production at all levels for 2015-2017

Level	2015	2016	2017
Top level			
1 Total	6188.44	6257.35	6124.05
Level 1			
2 A	3051.93	2991.52	2931.12
3 B	3136.51	3265.82	3192.93
Level 2			
4 AA	119.50	108.39	97.28
5 AB	2.94	2.88	2.87
6 AC	80.28	76.38	72.49
7 AD	30.33	29.98	29.63
8 AE	293.89	284.73	275.60
9 AF	532.25	512.66	493.13
10 AG	1338.82	1334.01	1329.40
11 AH	5.53	5.52	5.52
12 AI	326.72	326.90	327.13
13 AJ	132.90	127.80	122.40
14 AK	182.14	175.42	168.72
15 AL	3.33	3.31	3.30
16 AM	3.30	3.53	3.65
17 BA	215.04	210.49	194.35
18 BB	51.41	50.34	46.50

19	BC	1100.41	1131.02	1098.46
20	BD	2.72	2.45	2.05
21	BE	1681.28	1780.05	1766.38
22	BF	85.65	91.46	85.18

Figure 4 and 5 illustrate forecasting of sorghum production across all hierarchical levels for each series by using middle-out approach.

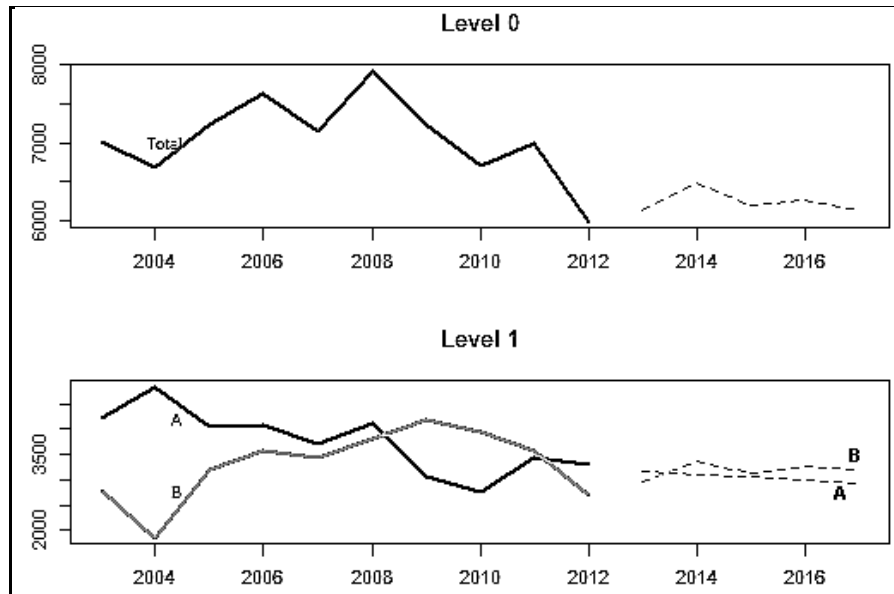


Fig 4 Hierarchical forecasting of sorghum production at level 0 and 1

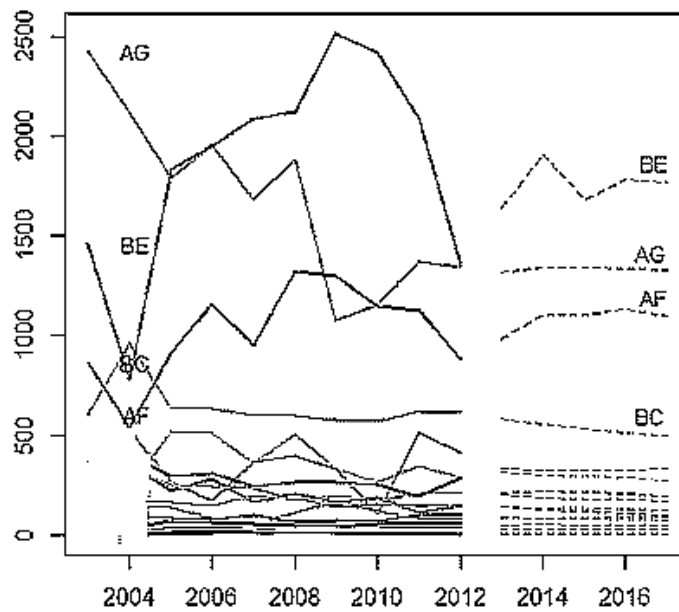


Fig 5 Hierarchical forecasting of sorghum production at level 2

The bottom-most level i.e. level 2 exhibits, through dotted lines, forecast value of seasonal sorghum production for each of the states, however, for clarity of graphical representation, only few of those are labeled. Level 1 shows forecasted production of sorghum in kharif and rabi seasons separately for India. Predicted future values of total sorghum production in India are displayed in level 0. For each of the graphical representation, y-axis represents production in '000 tonne.

REFERENCES

- Athanasopoulos G, Ahmed R A and Hyndman R J. 2009. Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting* **25**(1):146-66.
- Dangerfield B J and Morris J S. 1992. Top-down or bottom-up: Aggregate versus disaggregate extrapolations. *International Journal of Forecasting* **8**(2):233-41.
- Espasa A, Senra E and Albacete R. 2002. Forecasting inflation in the European Monetary Union: A disaggregated approach by countries and by sectors. *The European Journal of Finance* **8**(4):402-21.
- Hyndman R J, Ahmed R A, Athanasopoulos G and Shang H L. 2011. Optimal combination forecasts for hierarchical time-series. *Computational Statistics and Data Analysis* **55**(9): 2579-89.
- Hyndman R J, Ahmed R A and Shang H L. 2013. hts: Hierarchical and grouped time-series. R package version 4.3, URL <http://CRAN.R-project.org/package=hts>.
- Lo S, Wang F and Lin J T. 2008. Forecasting for the LCD monitor market. *Journal of Forecasting* **27**(4):341-56.
- Widiarta H, Viswanathan S and Piplani R. 2007. On the effectiveness of top-down approach for forecasting autoregressive demands. *Naval Research Logistics* **54**(2):176-88.
- Zellner A and Tobias J. 2000. A note on aggregation, disaggregation and forecasting performance. *Journal of Forecasting* **19**(5):457-65.
- Zotteri G, Kalchschmidt M and Caniato F. 2005. The impact of aggregation level on forecasting performance. *International Journal of Production Economics* **93-94**:479-91.

ANNEXTURE-I

Top level		
1	Total	India
Level 1: Season		
2	A	Kharif
3	B	Rabi
Level 2: State/Union territory		
4	AA	Kharif-Andhra Pradesh
5	AB	Kharif-Bihar*
6	AC	Kharif-Gujrat
7	AD	Kharif-Haryana
8	AE	Kharif-Karnataka
9	AF	Kharif-Madhya Pradesh**
10	AG	Kharif-Maharashtra
11	AH	Kharif-Orrisa
12	AI	Kharif-Rajasthan
13	AJ	Kharif-Tamil Nadu
14	AK	Kharif-Uttar Pradesh
15	AL	Kharif-Delhi
16	AM	Kharif-Others***
17	BA	Rabi-Andhra Pradesh
18	BB	Rabi-Gujrat
19	BC	Rabi-Karnataka
20	BD	Rabi-Madhya Pradesh
21	BE	Rabi-Maharashtra
22	BF	Rabi-Tamil Nadu

* (including Jharkhand)

** (including Chhattisgarh)

*** (Comprising of Dadra and Nagar Haveli, Jammu and Kashmir, Kerala, Nagaland, Puducherry, Punjab and West Bengal whose contributions are either nil in some of the years or insignificant compared to total sorghum production of India)

Introduction to Multivariate Time Series Model

Achal Lama, K N Singh, R S Shekhawat and Bishal Gurung
ICAR-IASRI, New Delhi
achal.lama@icar.gov.in

1. Introduction

Modelling and forecasting of major economic phenomenon involve a large number of variables, thus it must be addressed using the multivariate time-series methods. A large number of time-series models have been proposed in literature as alternatives to structural econometric models in economic forecasting applications. But, after the pioneering work of Sims (1980) the VAR models have received much attention. The class of VAR models is a special case of more general Vector Autoregressive Moving Average (VARMA) models. The VAR models were initially used as macroeconomic models, but it has been found as a promising alternative to structural econometric models where simultaneous forecasts are required for a collection of related microeconomic variables.

Let $y_t = (y_{1t}, y_{2t}, \dots, y_{nt})'$ denote an $(n \times 1)$ vector of time series variables. The basic p -lag vector autoregressive VAR (p) model has the form:

$$y_t = A + B_1 y_{t-1} + B_2 y_{t-2} + B_3 y_{t-3} + \dots + B_p y_{t-p} + \varepsilon_t$$

where, A is $k \times 1$ vector of intercepts, B_i ($i=1, 2, \dots, p$) is $k \times k$ matrices of parameters and $\varepsilon_t \sim iidN(0, \Sigma)$

To have an understanding of the VAR model let us consider a simple two variable case and the lag be one, i.e., $p = 1$ and $k = 2$. The VAR model can be represented as:

$$\begin{aligned} y_t &= \begin{bmatrix} y_{1t} \\ y_{2t} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{bmatrix} \\ &= A + B y_{t-1} + \varepsilon_t \end{aligned}$$

Now further defining

$$x_t = (y_{t-1}, \dots, y_{t-p})'$$

and,

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_T \end{bmatrix}$$

The likelihood function can be derived in two parts as follows:

$$\alpha|\Sigma, y \sim N(\hat{\alpha}, \Sigma \otimes (X'X)^{-1})$$

and

$$\Sigma^{-1}|y \sim W(S^{-1}, T - K - M - 1)$$

where, $\hat{B} = (X'X)^{-1}X'Y$ is the OLS estimates of B and $\hat{\alpha} = \text{vec}(\hat{B})$ and $S = (Y - X\hat{B})'(Y - X\hat{B})$.

2. Fitting of VAR model

In this section attempt has been made to summarize broadly the steps followed for modelling a multivariate time series data using VAR model. The steps are as follows:

1. Determination of Stationarity of the time series

The Stationarity of the data sets used is tested. As, it is the basic assumption which needs to be satisfied before proceeding for analysis of any time-series data. Statistical tests like Dickey-Fuller test, Augmented Dickey-Fuller (ADF) test, KPSS (Kwiatkowski, Phillips, Schmidt, and Shin) test, Philips-Perron test are available to test the stationarity. If required, the series needs to be differenced to make it stationary in mean.

2. Identification the mean model

After the time-series is stationary we go for identifying the mean model for the series. This is done by fitting the simple ARIMA (Autoregressive integrated moving average) model. The ARIMA (p,d,q) is determined by the ACF (Autocorrelation function) and PACF (Partial autocorrelation function) values of the stationary series. The parameter p is determined by the ACF value and q by the PACF value and d refers to order of differencing done to the original series to make it stationary. This procedure is useful for univariate case, but in case of multivariate setup we have to use either VAR or Vector Error Correction (VEC) model. The selection of the model depends upon the nature of the series to be modelled. If the series have long run dependency among themselves one need to use the VEC or else VAR model is used. Johansen test for cointegration is used to have an insight into the dependency relationship with the alternate hypothesis being the presence of cointegration. If we do not find cointegration between the series then we use the VAR model to identify the mean process. The order of the VAR model is identified

based on minimum Akaike Information Criterion (AIC) or Schwarz-Bayesian Criterion (SBC). AIC is given by

$$AIC = (-2\log L + 2k)$$

where, k is the number of parameters of the model and L is the likelihood function.

SBC is also used as an alternative to AIC which is given by

$$SBC = \log \sigma^2 + (k \log n) / n$$

3. Residual diagnostics

The parameters of the VAR model is estimated through maximum likelihood function such that an overall measure of errors is minimized or the likelihood function is maximized. This step is basically to check if the model assumptions about the errors are satisfied. To achieve this we used the autocorrelation function (ACF) plots for testing the presence of serial correlation in the residual series at various lags, we have also used the Q-Q normal probability plots to check the normality assumption of the residuals. If the residuals are either serially correlated or non-normal, we need to repeat Step 2.

4. Model comparison

The efficiency of the VAR model for forecasting is compared on the basis of following criterion:

Root Mean Squared Error (RMSE) expressed as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (Y_{p,t} - Y_{0,t})^2}$$

Mean absolute error (MAE) expressed as:

$$MAE = \frac{1}{N} \sum_{i=1}^N |(Y_{p,i} - Y_{0,i})|$$

where

Y_0 and Y_p are the observed and forecasted time series

N is the number of data points

The VAR model with minimum RMSE and MAE is selected for forecasting the multivariate time series under consideration.

The above described steps are followed and implemented using R software by calling the package “vars”. VAR model can be implemented in various other software like SAS, EViews, etc.

3. Impulse Response (IR)

Apart from modelling and forecasting VAR model is used for studying the impulse of one variable on the other variables of the system. As, VAR models represent the correlations among a set of variables, they are often used to analyse certain aspects of the relationships between the variables of interest. Granger (1969) has defined a concept of causality which, under suitable conditions, is fairly easy to deal within the context of VAR models. Therefore, it has become quite popular in recent years. The idea is that a cause cannot come after the effect. If a variable x affects a variable z , the former should help improving the predictions of the latter variable. Ω_t is the information set containing all the relevant information in the universe available upto and including period t . $z_t(h|\Omega_t)$ be the optimal (minimum MSE) h -step predictor of the process z_t at origin t , based on the information in Ω_t . The corresponding forecast MSE: $\Sigma_t(h|\Omega_t)$. The process x_t is said to cause z_t in Granger's sense if

$$\Sigma_t(h|\Omega_t) < \Sigma_t(h|\Omega_t \{x_s : s \leq t\}) \text{ for at least one } h = 1, 2, \dots$$

$\Omega_t \{x_s : s \leq t\}$ is the set containing all the relevant information in except for the information in the past and present of the x_t process. If z_t can be predicted more efficiently if the information in the x_t process is taken into account in addition to all other information, then x_t is Granger-causal for z_t . Granger-causality may not tell us the complete story about the interactions between the variables of a system. In applied work, it is often of interest to know the response of one variable to an impulse in another variable in a system that involves a number of further variables as well. One would like to investigate the impulse response relationship between two variables in a higher dimensional system. If there is a reaction of one variable to an impulse in another variable we may call the latter causal for the former. We will study this type of causality by tracing out the effect of an exogenous shock or innovation in one of the variables on some or all of the other variables. This kind of impulse response analysis is called multiplier analysis.

4. Data and implementation

For illustration we have taken three series namely the Food Wholesale price index (FWPI), monthly rainfall (mm) data and the Fiscal deficit data to study the BVAR model. The FWPI data was collected from Office of the Economic Adviser, Ministry of Commerce and Industry, Government of India. The rainfall data was collected from the official website of India Meteorological Department (IMD), Ministry of Earth Sciences, Government of India. The Fiscal deficit data was collected from the official website of Reserve Bank of India (RBI). All three series contain 120 data points from January, 2005 to December, 2014 out of which 114 points were used for model building purpose and the remaining 6 points were kept for validation. The time plot of the data set is depicted by Figure 1 and the descriptive statistics is given in Table 1. Looking at the time plot of the series we can identify the presence of seasonality in the rainfall series, hence the series were seasonally adjusted following standard procedure. From the descriptive statistics we can have an idea that the series under consideration is slightly skewed, has a small amount of kurtosis and also the series is non-normal. VAR model was fitted as per the steps described in Section 2 and hence forecasts were also obtained. Estimates of VAR model is presented in Table 2. The impulse response graph is depicted in Figure 2. From Table 2 we could find the interdependencies among the three series used for analysis. This helps the researcher to describe the movement of the series together. Further, with help of impulse response function we can find exactly how one series is affected by other series independently. This provides a good insight into the transfer of shocks between the variables in the system.

Table 1. Descriptive statistics

	FISCAL	FWPI	RAINFALL
Mean	302.30	167.28	94.12
Median	294.69	164.10	44.90
Maximum	1273.83	265.30	334.10
Minimum	-911.50	97.60	1.70
Std. Dev.	321.03	49.98	95.24
Skewness	-0.14	0.34	0.97
Kurtosis	4.58	1.84	2.48
Jarque-Bera	13.04	9.01	20.45
Probability	< 0.01	<0.01	< 0.01

Table 2. VAR model estimates

	FISCAL_D	FWPI_D	RAINFALL_SA
FISCAL_D(-1)	-0.71	0.0002	0.008
	(0.09)	(0.001)	(0.01)
FISCAL_D(-2)	-0.60	0.002	-0.009
	(0.10)	(0.001)	(0.01)
FISCAL_D(-3)	-0.33	0.004	0.005
	(0.12)	(0.001)	(0.01)
FISCAL_D(-4)	-0.28	0.002	0.001
	(0.11)	(0.001)	(0.01)
FISCAL_D(-5)	-0.35	0.001	0.004
	(0.09)	(0.001)	(0.01)
FWPI_D(-1)	-2.47	0.14	-1.88
	(8.78)	(0.09)	(1.06)
FWPI_D(-2)	-9.49	-0.003	0.61
	(8.64)	(0.09)	(1.05)
FWPI_D(-3)	13.30	0.17	-1.91
	(8.57)	(0.09)	(1.04)
FWPI_D(-4)	-23.09	-0.25	0.25
	(8.97)	(0.09)	(1.09)
FWPI_D(-5)	-14.10	-0.30	-0.41
	(8.85)	(0.09)	(1.07)
RAINFALL_SA(-1)	0.62	0.01	-0.06
	(0.85)	(0.009)	(0.10)
RAINFALL_SA(-2)	0.33	-0.01	-0.09
	(0.87)	(0.009)	(0.10)
RAINFALL_SA(-3)	1.25	-0.01	0.10
	(0.86)	(0.009)	(0.10)
RAINFALL_SA(-4)	1.34	-0.01	0.03
	(0.89)	(0.009)	(0.10)
RAINFALL_SA(-5)	-0.94	0.01	0.07
	(0.80)	(0.008)	(0.09)

C	-197.59	2.34	97.63
	(197.70)	(2.17)	(24.03)

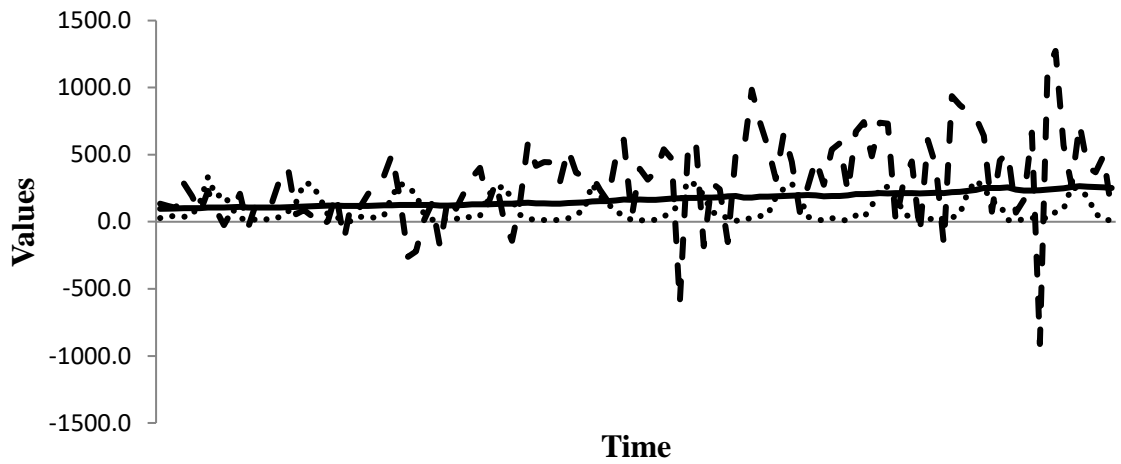


Figure 1. Time plot of Food WPI (solid line), Rainfall (dotted line) and Fiscal (dashed line)

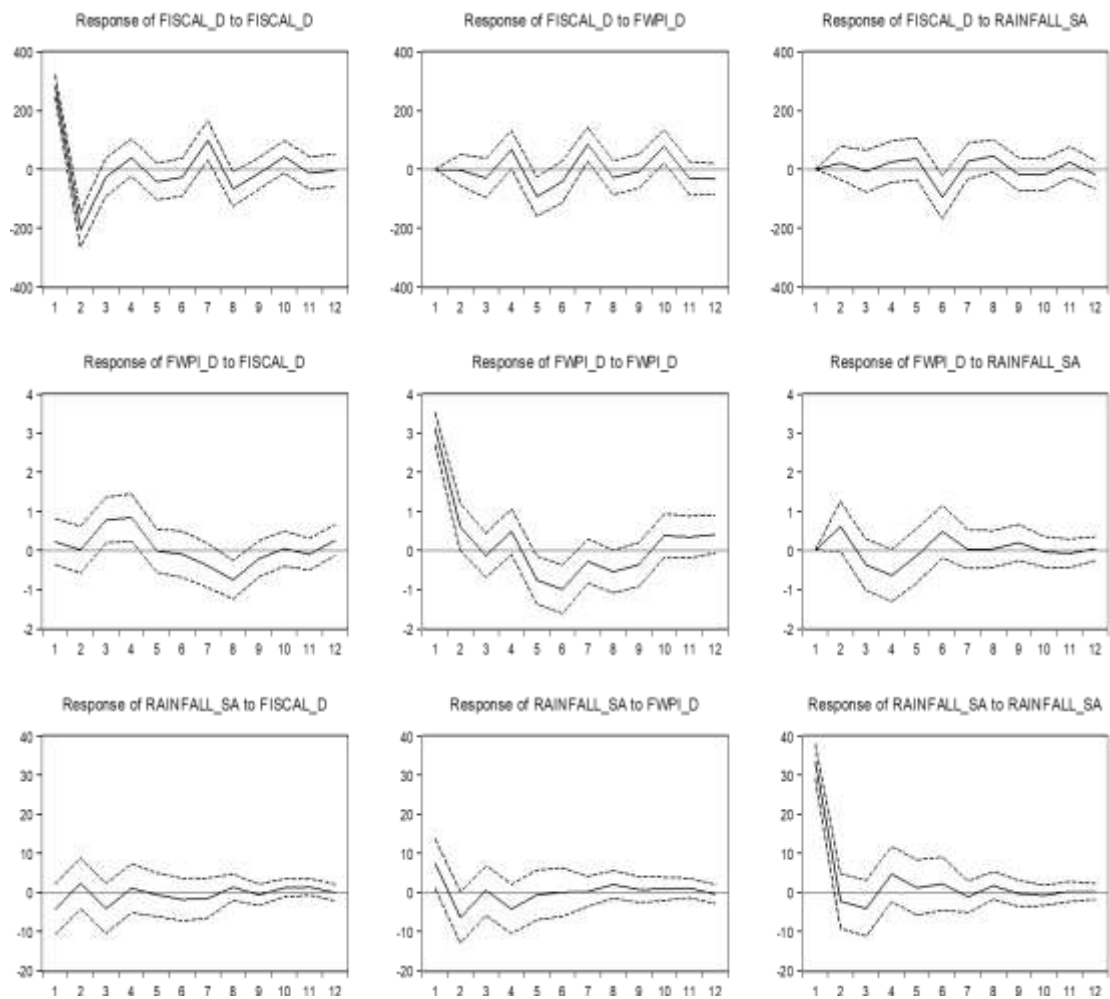


Figure 2. The impulse response graph of VAR model

Bibliography:

- Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods, *Econometrica: Journal of the Econometric Society*, 424-438.
- Kynclová, P., Filzmoser, P. and Hron, K. (2015). Modeling Compositional Time Series with Vector Autoregressive Models. *Journal of Forecasting*, **34**, 303–314.
- Jouini, T. (2015). Efficient Multistep Forecast Procedures for Multivariate Time Series. *Journal of Forecasting*, **34**, 604–618.
- Lama, A., Jha, G.K., Gurung, B., Paul, R.K. and Sinha, K. (2016). VAR-MGARCH Models for Volatility Modelling of Pulses Prices: An Application. *Journal of the Indian Society of Agricultural Statistics*, **70**, 145-151.
- Lama, A. (2017). Investigations on Bayesian multivariate time-series models. *Unpublished PhD thesis*, PG School, ICAR-IARI.
- Litterman, R. (1980). *Techniques for Forecasting with Vector Autoregressions*. University of Minnesota, Ph. D. Dissertation, Minneapolis.
- Ramos, F., F., R. (2003) Forecasts of market shares from VAR and BVAR models: a comparison of their accuracy. *International Journal of Forecasting*, **19**, 95-110.
- Sims, C. (1980). Macroeconomics and reality. *Econometrica*, **48**, 1-48.

Application of Bayesian methodology for Time Series Analysis

Achal Lama, K N Singh, R S Shekhawat and Bishal Gurung
ICAR-IASRI, New Delhi
achal.lama@icar.gov.in

1. Introduction

Bayesian estimation and inference has a number of advantages in statistical modelling. The core issue of Bayesian analysis is of incorporating prior information by specifying the prior distributions. The very basic assumption of a Bayesian framework is that the data is not exhaustive to explain all the underlying behavior of the series. Thus priors are to be assigned to the parameters of the model and then posterior is estimated under that prior information. Specifying the prior brings extra information or data based on the combined sources of information (prior and likelihood). Bayesian analysis also provides the density of the parameters of the model unlike the point or interval estimates provided by the classical approaches. In case of financial markets information flows freely, thus incorporating that information by means of prior seems justified. Thus, in last decade the use of Bayesian framework for analyzing financial time series data has accelerated. The basis of Bayesian estimation is the Bayes' Theorem. Let us consider the parametric space θ which is the vector of the parameters of the model with a prior density function $\pi(\theta)$ and \mathbf{Y} is the data vector. According to Bayes' rule, the posterior density

$$\pi(\theta|y) \propto L(Y|\theta)\pi(\theta)$$

where, $L(Y|\theta)$ is the likelihood function. The straightforward way to estimate θ is to compute the posterior mean of θ as follows:

$$\hat{\theta} = \int \theta \pi(\theta|y) d\theta$$

There are several ways of specifying a prior distribution in the Bayesian analysis. One can make use of non-informative priors or if the priors need to be analytically tractable then conjugate priors are to be used. A prior is said to be non-informative if it provides no information about the distribution of the parameter to be estimated. And in case of conjugate priors the posterior distribution of the parameters to be estimated is of the same class as that of the prior distribution. This helps in better statistical inference as the distribution of the posterior is of some known form. Another advantage of using conjugate prior is that for incorporating new information one needs to just update the values of hyperparameters instead of changing the prior distribution altogether. If the likelihood belong to exponential family of distribution such as normal distribution,

deriving conjugate prior is relatively easy. Conjugate properties of the exponential family of distribution are well discussed by Lee (2004). In practical situations, the selection of prior distribution depends largely upon the need of the researcher and the complexity of the problem addressed. Under the Bayesian framework the estimation of the parameters is done mainly using the Markov Chain Monte Carlo (MCMC) method. The reason behind using MCMC methods is the fact that this method comes with two very desirable properties that are essential for Bayesian analysis. The first one being the capability of handling high dimensional problems efficiently, and secondly the ability of drawing random samples directly from the posterior distribution. To understand the process let us assume that we want to have information regarding a distribution π^* , of which we have information unto the point C , where $C = \sum_{\theta \in E} \pi(\theta)$ with an assumption that the state space E is either finite or countable. Then the distribution of π^* will be $\pi(\theta)|C$, as its probability mass function. The main purpose of using MCMC method is to obtain the posterior distribution as

$$\pi^*(\theta|y) = \frac{f(y|\theta)p(\theta)}{\sum_{\theta \in E} f(y|\theta)p(\theta)}$$

For obtaining the posterior distribution, following steps are followed:

1. An ergodic Markov Chain $\theta_0, \theta_1, \theta_2, \dots$ is set up which results in a stationary posterior distribution.
2. Using Markov Chain simulate $\theta_0, \theta_1, \theta_2, \dots, \theta_{l+k}$ for large l and k .
3. Discard the first $l-1$ samples with $l+k$ sufficiently large to obtain.
4. Obtain the expectation and other statistics using the $l+k$ samples, this is done to obtain stationary values.

The expectation of the posterior distribution is important as it is used to estimate the parameters of the model that we have used in our study. And it is calculated as

$$\begin{aligned} E_{\pi^*(\theta)} &= \sum_{\theta \in E} \theta \pi^*(\theta|y) \\ &= \frac{\sum_{\theta \in E} \theta f(y|\theta)p(\theta)}{\sum_{\theta \in E} f(y|\theta)p(\theta)} \end{aligned}$$

But, if the posterior distribution is high dimensional or else complicated, it is difficult to obtain closed form solutions for C . The answer to this is the MCMC method. The two very widely used MCMC algorithms are Metropolis-Hastings (MH) algorithm and Gibbs sampling. Gibbs sampling is considered to be a special sampler of MH algorithm.

2. Sampling Algorithms

2.1. Metropolis-Hastings algorithm

The MH algorithm is a popular algorithm which is used to obtain a sequence of random samples from a proposed distribution $q(\theta, \xi)$ where direct sampling is difficult. The algorithm is first proposed by Metropolis *et al.*, (1953) and extended by Hastings (1970). The idea of MH is simple, in this method a proposal point ξ is generated from the proposal distribution $q(\theta, \cdot)$ with an acceptance probability as

$$\alpha(\theta, \xi) = \min\{1, r(\theta, \xi)\}, \text{ where}$$

$$r(\theta, \xi) = \frac{\pi(\xi)q(\theta, \xi)}{\pi(\theta)q(\theta, \xi)}$$

This process can be thought of as generating a random number X from a uniform distribution $U [0,1]$ and accepting the state ξ if $X < \alpha(\theta, \xi)$, otherwise the point θ is rejected and the algorithm remains in the same state. The quantity $r(\theta, \xi)$ is known as the MH ratio and hence the algorithm as MH algorithm. This algorithm can be summarized in following steps

1. A proposal distribution is selected with transition matrix $Q=(q(I,j))_{I,j \in E}$. Select an integer s between 1 and n .
2. Assign $n=0$ and $\theta_0=s$.
3. A random variable θ is generated such that $P(\theta = j) = q(\theta_n, j)$ and X is generated independently.
4. If $X < \alpha(s, j)$, then $\theta=j$, otherwise $\theta = s$.
5. Next n is set as $n=n+1$ and $\theta_n = \theta$
6. Go to step 3.

Random walk algorithm is a special case of the MH algorithm, in which the proposed distribution has the symmetric property $q(\theta / s) = q(s / \theta)$. This method performs poorly when we need to deal with high dimensional models, as dimension increases the acceptance rate.

2.2. Gibbs sampling

Gibbs sampling is another widely used MCMC algorithm named after Josiah Willard Gibbs but proposed by Geman and Geman (1984). This algorithm is simple, easily implemented and can handle the problem of high dimensionality. As, already discussed conjugate priors are very handy in Bayesian analysis, but there are many situations where it is difficult to construct a joint conjugate prior for several parameters. But in such situations conditional conjugate priors can be obtained relatively easily. Gibbs sampling uses the concept of conditional priors and converts the multidimensional problem into a low dimensional problem. The advantage of using conditional conjugate prior is that it takes the same distributional form as that of posterior distribution. Let us assume a data set $y = (y_1, y_2, \dots, y_n)$ and distribution of each y_i have v parameters, $\theta = (\theta_1, \dots, \theta_v)$. For each $j=1, 2, \dots, v$, a one-dimensional conjugate prior $p(\theta_j)$ is defined and the conditional posterior is computed by using Bayes theorem. The Gibbs sampling procedure is iterated through the following steps:

1. The initial parameter vector $(\theta_1^0, \dots, \theta_v^0)$ is defined.
2. Parameter vector is updated by sampling as follows:

$$\theta_1^1 \propto p(\theta_1 | \theta_2^0, \dots, \theta_v^0, y)$$

$$\theta_2^1 \propto p(\theta_2 | \theta_1^1, \theta_3^0, \dots, \theta_v^0, y)$$

.

.

.

$$\theta_v^1 \propto p(\theta_v | \theta_1^1, \theta_2^1, \dots, \theta_{v-1}^1, y)$$

3. Using this updated values as starting parameter values the sampling is repeated M times. M is a constant which is selected to be sufficiently large and referred to as burn-in period.

4. After simulating $\{\theta^{(M+1)}, \theta^{(M+2)}, \dots, \theta^{(M+n)}\}$ from the Gibbs sampling Bayesian inferences are drawn.

The main drawback of this method is that it is infeasible to apply when complete conditional distribution is not known.

3. Bayesian time series models

The Bayesian framework has been extended to a large number time series models. In this write up I have concentrated on MGARCH and VAR models. Both these models are multivariate in nature and at present are being largely used in macroeconomics. Here we will briefly describe the MGARCH and VAR model. Let $Y_t = (y_{1t}, y_{2t}, \dots, y_{kt})'$ denote an $(k \times 1)$ vector of time series variables. The basic p -lag vector autoregressive VAR (p) model has the form:

$$Y_t = A + B_1 Y_{t-1} + B_2 Y_{t-2} + B_3 Y_{t-3} + \dots + B_p Y_{t-p} + \varepsilon_t$$

where, A is $k \times 1$ vector of intercepts, B_i ($i = 1, 2, \dots, p$) is $k \times k$ matrices of parameters and $\varepsilon_t \sim iidN(0, \Sigma)$

For a multivariate time series $y_t = (y_{1t}, \dots, y_{kn})'$ the MGARCH model is given by:

$$y_t = H_t^{1/2} \varepsilon_t$$

where, $H_t^{1/2}$ is $k \times k$ positive-definite matrix and of the conditional variance of y_t . k is the number of series and $t=1, 2, \dots, n$ (number of observations). It is with the specification of conditional variance that the MGARCH model changes. Engle and Kroner (1995) introduced the BEKK model which is the direct generalization of the univariate GARCH model. The resulting variance is dependent on the amount of currently available information. A general GARCH (p, q) model (Bollerslev, 1986) can be defined as:

$$h_t = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_p \varepsilon_{t-p}^2 + \beta_1 h_{t-1} + \dots + \beta_q h_{t-q}, \quad \alpha_i > 0, \beta_i > 0, \quad \alpha_i + \beta_i < 1$$

where, h_t is the conditional variances which depends on the previous error terms as well as previous conditional variances of the process.

Equation (2) can be transferred into multivariate GARCH model with a generalization of the resulting variance matrix H_t

$$H_t = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$$

Each element of H_t depends on the p delayed values of the squared ε_t , the cross product of ε_t and on the q delayed values of elements from H_t . In general, multivariate GARCH (I, I) model can be written as:

$$H_t = C_0' C_0 + \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} \begin{pmatrix} \varepsilon_1^2 & \varepsilon_1 \varepsilon_2 & \varepsilon_1 \varepsilon_3 \\ \varepsilon_2 \varepsilon_1 & \varepsilon_2^2 & \varepsilon_2 \varepsilon_3 \\ \varepsilon_3 \varepsilon_1 & \varepsilon_3 \varepsilon_2 & \varepsilon_3^2 \end{pmatrix} \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} + \dots$$

$$\begin{pmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{pmatrix} \begin{pmatrix} h_{11} & h_1 h_2 & h_1 h_3 \\ h_2 h_1 & h_{22} & h_2 h_3 \\ h_3 h_1 & h_3 h_2 & h_{33} \end{pmatrix} \begin{pmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{pmatrix}$$

In compact form, the above equation can also be written as:

$$H_t = C_0' C_0 + A' \varepsilon_{t-1} \varepsilon_{t-1}' A + B' H_{t-1} B$$

For 2 variable case the model can be represented as:

$$H_t = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \varepsilon_{1,t-1}^2 & \varepsilon_{1,t-1} \varepsilon_{2,t-1} \\ \varepsilon_{2,t-1} \varepsilon_{1,t-1} & \varepsilon_{2,t-1}^2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} H_{t-1} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$

$$h_{11,t} = c_{11} + a_{11}^2 \varepsilon_{1,t-1}^2 + 2a_{11}a_{21} \varepsilon_{1,t-1} \varepsilon_{2,t-1} + a_{21}^2 \varepsilon_{2,t-1}^2 + g_{11}^2 h_{11,t-1} + 2g_{11}g_{21} h_{12,t-1} + g_{21}^2 h_{22,t-1}$$

$$h_{12,t} = c_{12} + a_{11}a_{21} \varepsilon_{1,t-1}^2 + (a_{21}a_{12} + a_{11}a_{22}) \varepsilon_{1,t-1} \varepsilon_{2,t-1} + a_{21}a_{22} \varepsilon_{2,t-1}^2 + g_{11}g_{12} h_{11,t-1} + (g_{21}g_{12} + g_{11}g_{22}) h_{12,t-1} + g_{21}g_{22} h_{22,t-1}$$

$$h_{22,t} = c_{22} + a_{12}^2 \varepsilon_{1,t-1}^2 + 2a_{12}a_{22} \varepsilon_{1,t-1} \varepsilon_{2,t-1} + a_{22}^2 \varepsilon_{2,t-1}^2 + g_{12}^2 h_{11,t-1} + 2g_{12}g_{22} h_{12,t-1} + g_{22}^2 h_{22,t-1}$$

As already discussed for Bayesian analysis one has to assign priors to the parameters of the model. Thus priors for MGARCH and VAR models are defined accordingly. In case of MGARCH model the Normal priors with different parametric ranges for different parameters to be estimated. The constant terms of each model is assigned $N(0, 10)$ priors, whereas the other parameters are assigned $N(0, 100)$ priors. The reason behind using

Normal priors is due to its conjugate property. The priors are assigned following Fioruci, *et al.*, (2014) as it yields satisfactory results in case of MGARCH models. For VAR model three different priors are used namely Minnesota, Normal-Wishart and Independent- Normal Wishart. Each of the priors differs from each other, in the manner in which they specify the mean and the variance of the coefficient's distribution. The specification of the Minnesota prior is as follows:

$$\alpha \propto N(\underline{\alpha}_{Min}, \underline{V}_{Min})$$

If \underline{V}_i denotes the block of \underline{V}_{Min} associated with the K coefficients in equation i and $\underline{V}_{i,jj}$ as its diagonal elements, then a common implementation of the Minnesota prior would set:

$$\underline{V}_{i,jj} = \frac{a_1}{p^2} \text{ for coefficients on own lags}$$

$$\frac{a_2 \sigma_{ii}}{p^2 \sigma_{jj}} \text{ for coefficients on lags of variable } j \neq i$$

$$\underline{a}_3 \sigma_{ii} \text{ for coefficients on exogenous variables}$$

This prior simplifies the complicated choice of fully specifying all the elements of \underline{V}_{Min} in choosing three scalars $\underline{a}_1, \underline{a}_2, \underline{a}_3$.

The next prior used is a natural conjugate prior Normal-Wishart. The form of the prior is as follows:

$$\alpha | \Sigma \propto N(\underline{\alpha}, \Sigma \otimes \underline{V})$$

$$\Sigma^{-1} \propto W(\underline{S}^{-1}, \underline{v})$$

where, $\underline{\alpha}, \underline{V}, \underline{v}$ and \underline{S} are to be selected by the experimenter depending upon the data set in use. Then the posterior of this prior is as follows:

$$\alpha | \Sigma, y \propto N(\bar{\alpha}, \Sigma \otimes \bar{V})$$

$$\Sigma^{-1} | y \propto W(\bar{S}^{-1}, \bar{v})$$

where,

$$\bar{\alpha} = \text{vec}(\bar{B})$$

$$\bar{B} = \bar{V} \left[\underline{V}^{-1} \underline{B} + X' X B \right]$$

$$\bar{S} = S + \underline{S} + B' X' X B + \underline{B}' \underline{V}^{-1} \underline{B} - \bar{B}' (\underline{V}^{-1} + X' X) \bar{B}$$

$$\bar{v} = T + \underline{v}$$

The third prior taken up is the independent Normal-Wishart, which has the following form:

$$p(\beta, \Sigma^{-1}) = p(\beta)p(\Sigma^{-1})$$

where

$$\beta \sim N(\underline{\beta}, \underline{V}_\beta)$$

and

$$\Sigma^{-1} \sim W(\underline{S}^{-1}, \underline{v})$$

This prior allows for the prior covariance matrix \underline{V}_β , to take any values chosen by the researcher, rather than the restrictive $\Sigma \otimes \underline{V}$ form of the natural conjugate prior. In this prior, the joint posterior $p(\beta, \Sigma^{-1}|y)$ does result in an easily computable form that would allow easy Bayesian analysis this is due to the fact that posterior means and variances do not have analytical forms.

4. Data description and illustration

For illustration purpose I have implemented Bayesian framework in MGARCH model with BEKK specification. A data set which contains two monthly series namely International and Domestic price indices of edible oils is taken. The International oil price index was collected from the World Bank Commodity Prices Indices (Pink Sheet) available from its official website. And, Domestic edible oils price index was collected from Office of the Economic Adviser, Ministry of Commerce and Industry, Government of India. The monthly data set contains 313 data points (January, 1990 to January, 2016). As discussed earlier priors are assigned and then using MCMC the posterior distribution is obtained. Time plot of the series used are depicted by Figure 1. The parameter estimates obtained using Bayesian framework are reported in Table 1. Also the conditional volatility obtained are presented in Figure 2.

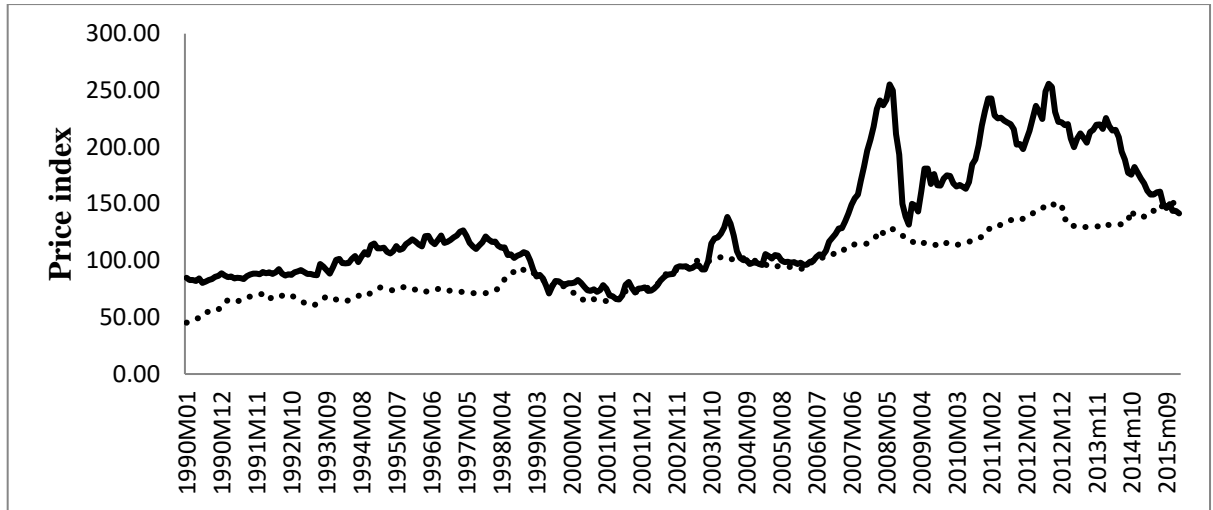


Figure 1. Time plot of International (bold) and Domestic (dotted) edible oils price indices

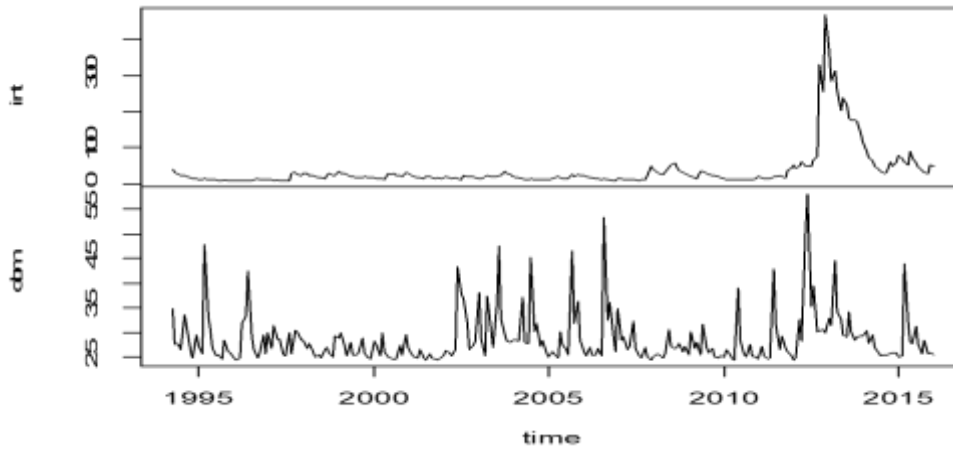


Figure 2. Conditional variance of Domestic (dom) and International (int) edible oils price indices after fitting MGARCH-BEKK model

Table 1. Estimates of Bayesian MGARCH-BEKK model for International and Domestic edible oils price indices

Coefficients	Estimate	Std. Error	t value	P value
C11	-0.158	0.026	-6.076	<0.001
C21	0.193	0.025	7.720	<0.001
C22	0.079	0.026	3.038	<0.001
A11	0.231	0.024	9.625	<0.001

A21	0.488	0.025	19.520	<0.001
A12	-0.166	0.026	-6.384	<0.001
A22	-0.004	0.025	-0.160	0.873
B11	-0.025	0.027	-0.925	0.355
B21	0.507	0.026	19.500	<0.001
B12	0.374	0.025	14.960	<0.001
B22	0.264	0.026	10.153	<0.001

Bibliography

- Asai, M., McAleer, M. and Yu, J. (2006). Multivariate stochastic volatility: a review. *Econometric Reviews*, **25**(2–3), 145–175.
- Asai, M. (2015). Bayesian analysis of general asymmetric multivariate garch models and news impact curves. *Journal of Japan Statistical Society*, **45**, 129-144.
- Bauwens, L., Laurent, S. and Rombouts, J. V. K. (2006). Multivariate GARCH models: a survey. *Journal of Applied Econometrics*, **21**, 79-109.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics*, **31**, 307-327.
- Carriero, A., Kapetanios, G. and Marcellino, M. (2009). Forecasting exchange rates with a large Bayesian VAR. *International Journal of Forecasting*, **25**, 400–417.
- Engle, R.F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of U.K. inflation. *Econometrica*, **50**, 987-1008.
- Fioruci, J. A., Ehlers, R. S. and Filho, M. G. A. (2014). Bayesian multivariate GARCH models with dynamic correlations and asymmetric error distributions. *Journal of Applied Statistics*, **41**, 320-331.
- Lama, A. (2017). Investigations on Bayesian multivariate time-series models. *Unpublished PhD thesis*, PG School, ICAR-IARI.
- Sims, C. (1980). Macroeconomics and reality. *Econometrica*, **48**, 1-48.

Wavelet Frequency Domain Approach for Time-Series Modeling

Ranjit Kumar Paul
ICAR-IASRI, New Delhi
ranjit.paul@icar.gov.in ranjitstat@gmail.com

1. Introduction

Autoregressive integrated moving average (ARIMA) methodology (Box *et al.*, 2007), which is a parametric approach, has virtually dominated analysis of time-series data during last several decades. Here role of various explanatory variables enter into the model “implicitly” through response variable observations at past epochs. However, quite often it is not possible to postulate appropriate parametric form for the underlying phenomenon and, in such cases; “Nonparametric” approach is called for. Accordingly, in recent years, an extremely powerful methodology of “Wavelet analysis” is rapidly emerging (Vidakovic, 1999; Percival and Walden, 2000). Although, a number of research papers have been published dealing with various theoretical aspects of wavelets, their application to data is still a difficult task.

Wavelet analysis can be studied in two ways: one is in “time domain” another is in “frequency domain”. In respect of the former, Sunilkumar and Prajneshu (2004) applied wavelet thresholding approach for modelling and forecasting of monthly meteorological subdivisions rainfall in Eastern U. P., India. For the latter approach, Almasri *et al.* (2008) have recently proposed a test statistic by using wavelet decompositions to test the significance of trend in a time-series data. The most difficult problem of testing for linear trend is the presence of dependence among the residuals because of which, tests for trend based on the classical ordinary least squares (OLS) regression are inappropriate. In many situations, the error autocovariance function exhibits a slow decay reflecting the possible presence of long memory process. The wavelet analysis, however, has been extensively used for such purposes, since it suitably matches the structure of these processes. The autocovariance function of the wavelet transformed series exhibits different behaviour, in the sense that autocovariance functions of the transformed series decay hyperbolically fast at a rate much faster than the original process. In general, the series that are correlated in the time domain become almost uncorrelated in the wavelet domain.

Agricultural performance of a country, generally, depends to a large extent on the quantum and distribution of rainfall. So its accurate modelling is vital in planning and policy making. Accordingly, several attempts have been made in the past to develop models for describing rainfall. In Indian context, Rajeevan *et al.* (2004) have provided an excellent review of multiple and power regression models employed since 1988 along with various modifications made in these models from time to time, particularly in the identification of relevant explanatory variables. The purpose of this lecture is to discuss and apply wavelet methodology in frequency domain for estimation and testing of significance of trend in India's monsoon rainfall data during the period 1979 to 2006.

2. Basics of Wavelets

The term *wavelet* is used to refer to a set of basic functions with a very special structure which is the key to the main fundamental properties of wavelets and their usefulness in statistics. Wavelets are fundamental building block functions, analogous to the trigonometric sine and cosine functions. As with a sine or cosine wave, a wavelet function oscillates about zero. This oscillating property makes the function a *wave*. However, the oscillations for a wavelet damp down to zero, hence the name *wavelet*. If $\psi(\cdot)$ be a real valued function defined over the real axis $(-\infty, \infty)$ and satisfying two basic properties:

(i) The integral of $\psi(\cdot)$ is zero:

$$\int_{-\infty}^{\infty} \psi(u) du = 0$$

(ii) The square of $\psi(\cdot)$ integrates to unity:

$$\int_{-\infty}^{\infty} \psi^2(u) du = 1$$

Then the function $\psi(\cdot)$ is called a wave.

2.1 Discrete Fourier transform

Transformation of a function to its wavelet components has much in common with transforming a function to its Fourier components. An introduction to wavelets begins with a discussion of the usual Fourier transformation. The French mathematician Jean-Baptiste

Fourier discovered that any class of square-integrable functions, defined on the interval $[-\pi, \pi]$, can be decomposed into component functions constructed by standard trigonometric functions. A function f belongs to the square-integrable space $L^2[a, b]$ if

$$\int_a^b f^2(x)dx < \infty$$

Fourier's results states that any function $f \in L^2[-\pi, \pi]$ can be expressed as an infinite sum of dilated cosine and sine functions given by

$$f(x) = \frac{1}{2} a_0 + \sum_{j=1}^{\infty} (a_j \cos(jx) + b_j \sin(jx)) \quad (1)$$

where

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos jx dx \quad j = 0, 1, 2, \dots$$

$$b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin jx dx \quad j = 1, 2, \dots$$

The series expansion is regarded as a transform, taking a function f into a set of coefficients a_j and b_j . The *Fourier series expansion* is extremely useful in that any L^2 function can be written in terms of very simple building block functions: sines and cosines, because the set of functions $\{\sin(j.), \cos(j.), j=1,2,\dots\}$, together with the constant function, form a *basis* for the function space $L^2[-\pi, \pi]$ which is orthonormal. A sequence of functions $\{f_j\}$ are orthonormal if the f_j 's are pairwise orthogonal and if $\|f_j\|=1$, for all j .

2.2 Wavelet analysis versus Fourier analysis

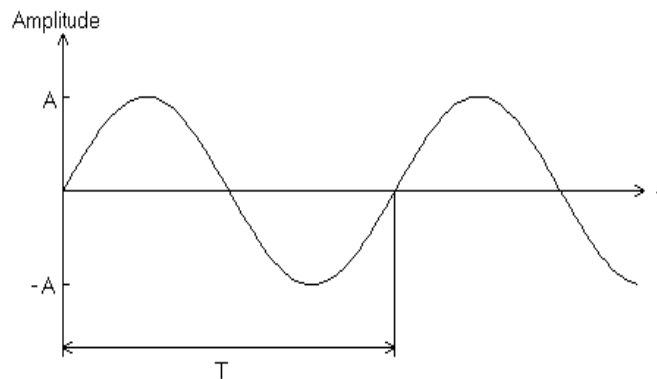
There is an obvious analogy between wavelet analysis and Fourier analysis in the sense that both the techniques aim to represent a function as a linear superposition of basis functions. In the case of wavelet analysis, the basis functions are the wavelets $\{\psi_{j,k}\}$ whereas in Fourier analysis they are the exponentials, $\{e^{iwx} = \cos wx + i \sin wx\}$. The most obvious difference is that the wavelet basis are indexed by two parameters (j and k) and have an infinite set of possible basis functions whereas Fourier basis functions are indexed by the single parameter w and have only a single set of basis functions. Comparing Fourier and wavelet analyses, the essential point is that the sines and cosines of the standard Fourier analysis have specificity only in frequency, whereas the special structure of a wavelet basis

provides specificity in location (via translation) and also specificity in ‘frequency’ (via dilation).

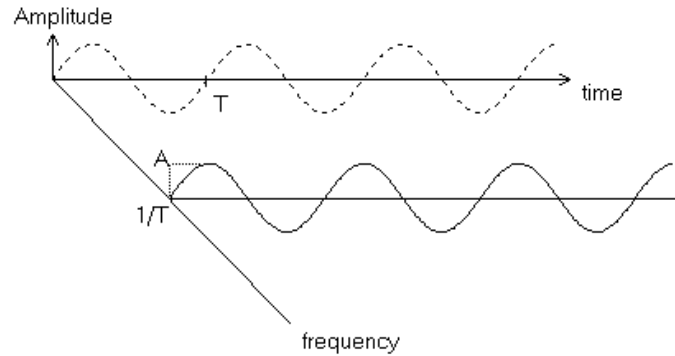
This means that wavelet analysis provides information not only on what frequency components are present but also when or where they are occurring. Another notable feature is that the wavelet transforms of a function are *localized*, *i.e.*, are time varying and depend only on the properties of the function in the neighborhood of each time point. This implies that if the function has singularities (such as discontinuities or ‘spikes’), these will affect only the wavelet transform near the singularities. By contrast the Fourier transforms depends on the global properties of the function and any singularity in the function will affect all such transforms. Hence wavelets have significant advantages over basic Fourier analysis when the function under study function has singularities.

2.3 Time domain versus Frequency domain

The most common representation of signals and waveforms is in the time domain. However, most signal analysis techniques work only in the frequency domain. The concept of the frequency domain representation of a signal is quite difficult. The frequency domain is simply another way of representing a signal. For example, consider a simple sinusoid.



The time - amplitude axes on which the sinusoid is shown define the *time plane*. If an extra axis is added to represent frequency, then the sinusoid would be as illustrated below.



The frequency - amplitude axes define the *frequency plane* in a manner similar to the way the time plane is defined by the time - amplitude axes. This frequency plane is what is represented when the spectrum of a signal is shown. The frequency plane is orthogonal to the time plane, and intersects with it on a line which is the amplitude axis.

Note that the time signal can be considered to be the projection of the sinusoid onto the time plane (time - amplitude axes). The actual sinusoid can be considered to be as existing some distance along the frequency axis away from the time plane. This distance along the frequency axis is the frequency of the sinusoid, equal to the inverse of the period of the sinusoid. The waveform also has a projection onto the frequency plane. These two projections mean that the sinusoid appears as a sinusoid in the time plane (time - amplitude axes), and as a line in the frequency plane (frequency - amplitude axes) going up from the frequency of the sinusoid to a height equal to the amplitude of the sinusoid.

3. Discrete Wavelet Transform (DWT)

There are two main waves of wavelets. The first wave resulted what is known as the continuous wavelet transform (CWT), which is designed to work with time-series defined over the entire real axis; the second is the discrete wavelet transform (DWT) which deals with series defined essentially over a range of integers. DWT of a time-series observation is used to capture high and low frequency components. This, in turn, would enable modelling of time-series data through computation of inverse DWT.

The basic reason why the DWT is such an effective analysis tools are the following:

- (i) The DWT re expresses a time-series in terms of coefficients that are associated with a particular time and a particular dyadic scale 2^{j-1} . These

coefficients are fully equivalent to the original series in that we can perfectly reconstruct a time-series from its DWT coefficients.

- (ii) The DWT allows us to partition the energy in a time-series into pieces that are associated with different scales and times. Energy decomposition is very close to the statistical technique known as the analysis of variance (ANOVA).
- (iii) The DWT effectively decorrelates a wide variety of time-series that occurs quite commonly in the physical applications. This property is the key to the use of DWT in the statistical methodology.
- (iv) The DWT can be computed using an algorithm that is faster than the celebrated fast Fourier transform algorithm.

Computation of DWT is carried out by “Pyramid algorithm” discussed below:

The first stage for computing the DWT simply consists of transforming the time-series \mathbf{X} of length $N = 2^J$ into the $N/2$ first level wavelet coefficients \mathbf{W}_1 and the $N/2$ first level scaling coefficients \mathbf{V}_1 . Precisely, to obtain unit scale wavelet coefficients, time-series $\{X_t : t = 0, \dots, N-1\}$ is circularly filtered with filter h_l , $l = 1, 2, \dots, L-1$, where L is the width of the filter and must be an even integer. For h_l to have width L , it must satisfy the conditions: $h_0 \neq 0$ and $h_{L-1} \neq 0$. Now define $h_l = 0$ for $l < 0$ and $l \geq L$ so that h_l is actually an infinite sequence with at most L nonzero values. A wavelet filter must satisfy the following three basic properties:

$$\sum_{l=0}^{L-1} h_l = 0, \quad \sum_{l=0}^{L-1} h_l^2 = 1 \quad \text{and} \quad \sum_{l=0}^{L-1} h_l h_{l+2n} = \sum_{l=-\infty}^{\infty} h_l h_{l+2n} = 0,$$

for all nonzero integers n . Compute

$$2^{1/2} \tilde{W}_{1,t} = \sum_{l=0}^{L-1} h_l X_{(t-l) \bmod N}, \quad t = 0, 1, \dots, N-1. \quad (2)$$

Now define $N/2$ wavelet transforms for unit scale corresponding to $t=0, \dots, N/2-1$ as

$$W_{1,t} = 2^{1/2} \tilde{W}_{1,2t+1} = \sum_{l=0}^{L-1} h_l X_{(2t+1-l) \bmod N}, \quad (3)$$

This procedure is called “Downsampling” procedure. To obtain first stage scaling coefficients, define scaling filter $g_l = (-1)^{l+1} h_{L-1-l}$.

Then the first level scaling coefficients are

$$V_{1,t} = 2^{1/2} \tilde{V}_{1,2t+1} = \sum_{l=0}^{L-1} g_l X_{(2t+1-l) \bmod N} \quad (4)$$

The second stage of Pyramid algorithm consists of treating $\{V_{1,t}\}$ in the same way as $\{X_t\}$ was treated in the first stage. Then we circularly filter $\{V_{1,t}\}$ separately with $\{h_l\}$ and $\{g_l\}$ and subsample to produce two new series, namely

$$W_{2,t} = \sum_{l=0}^{L-1} V_{1,(2t+1-l) \bmod N/2} \quad (5)$$

$$V_{2,t} = \sum_{l=0}^{L-1} V_{1,(2t+1-l) \bmod N/2}, \quad t=0,1,\dots,N/4-1. \quad (6)$$

Above procedure is repeated J times to obtain 2^J DWT's. There are $J-2$ subsequent stages to the Pyramid algorithm. For $j = 3, \dots, J$, the j^{th} stage transforms V_{j-1} of length $N/2^{j-1}$ into W_j and V_j each of length $N/2^j$. At the j^{th} stage, the elements of V_{j-1} are filtered separately with wavelet filter $\{h_l\}$, and scaling filter $\{g_l\}$. The filter outputs are subsampled to form respectively W_j and V_j . The elements of V_j are called the scaling coefficients for level j , while those of W_j contain the desired wavelet coefficients for level j . At the end of J^{th} stage, the DWT coefficient W is formed by concatenating the $J+1$ vectors.

Let P be an $N \times N$ real valued matrix defining the DWT and satisfying the orthonormality property $P'P = I_N$, where I_N is the $N \times N$ identity matrix. Then the DWT (W) of the time-series vector X may be computed by $W = P X$. Now the elements of the vector W are decomposed into $J+1$ subvectors. The first J subvectors contains all of the DWT coefficients for scale τ_j . Then W can be written as

$$W = [W'_1 \ W'_2 \ \dots \ W'_J \ V'_J]'$$

4. Multiresolution Analysis (MRA)

Consider the wavelet synthesis of X

$$X = P'W = \sum_{j=1}^J P'_j W_j + Q'_J V_J, \quad (7)$$

where P_j and Q_J matrices are defined by partitioning the rows of P commensurate with the partitioning of W into W_1, \dots, W_J and V_J . Thus the $N/2 \times N$ matrix P_1 is formed from the $n = 0$ up to $n = N/2-1$ rows of P ; the $N/4 \times N$ matrix P_2 is formed from the $n = N/2$ up to $n =$

$3N/4-1$ rows; and so forth, until we come to the $1 \times N$ matrices \mathbf{P}_J and \mathbf{Q}_J , which are the last two rows of \mathbf{P} .

Thus

$$\mathbf{P} = [P_1 P_2 \dots P_J Q_J]'$$

Now define $\mathbf{D}_j = \mathbf{P}'_j \mathbf{W}_j$ for $j = 1, \dots, J$, which is an N dimensional column vector whose elements are associated with changes in \mathbf{X} at scale τ_j ; i.e., $\mathbf{W}_j = \mathbf{P}_j \mathbf{X}$ represents the portion of the analysis $\mathbf{W} = \mathbf{P} \mathbf{X}$ attributable to scale τ_j , while $\mathbf{P}'_j \mathbf{W}_j$ is the portion of the synthesis $\mathbf{X} = \mathbf{P}' \mathbf{W}$ attributable to scale τ_j . Let $\mathbf{S}_J = \mathbf{Q}'_J \mathbf{V}_J$ which has all its elements equal to the sample mean \bar{X} . Then it can be seen that

$$\mathbf{X} = \sum_{j=1}^J \mathbf{D}_j + \mathbf{S}_J, \quad (8)$$

which defines a multiresolution analysis (MRA) of \mathbf{X} ; i.e., the time-series \mathbf{X} is expressed as the sum of a constant vector \mathbf{S}_J and J other vectors $\mathbf{D}_j, j = 1, \dots, J$ each of which contains a time-series related to variations in \mathbf{X} at a certain scale. \mathbf{D}_j is called the j^{th} level wavelet detail.

5. Estimation of Trend by Wavelets

Some times it is important to decompose a time-series into different components of variations like, low frequencies (trend), and high-frequency (noise) components. And the multiresolution analysis is used for decomposing and describing the low frequencies and high-frequency components in the data in a scale by scale basis. Consider the following model for a time-series data $\{X_t\}$:

$$X_t = \mu + T_t + Z_t, \quad t = 0, \dots, N-1, \quad (9)$$

where μ is a constant term, T_t is an unknown deterministic polynomial trend function of order r , Z_t is a residual term which is a long-memory process defined by $(1-B)^\delta Z_t = \varepsilon_t$, where, δ is the long memory parameter, $\{\varepsilon_t\}$ is a Gaussian white noise process with mean zero and $\sigma_\varepsilon^2 > 0$. Here, B , is the back shift operator such that $BZ_t = Z_{t-1}$.

Now, since $\mathbf{W} = [\mathbf{W}'_1 \mathbf{W}'_2 \dots \mathbf{W}'_J \mathbf{V}'_J]'$, the vector \mathbf{W} can be written as sum of two vectors: $\mathbf{W} = \mathbf{W}_w + \mathbf{W}_s$, where \mathbf{W}_w is an $N \times I$ vector containing the wavelet coefficients and

zeros at all other locations, and \mathbf{W}_s is an $N \times I$ vector containing the scaling coefficients and zeros at all other locations. Since $\mathbf{X} = \mathbf{P} \mathbf{W}$, therefore,

$$\mathbf{X} = \mathbf{P} \mathbf{W} = \mathbf{P} \mathbf{W}_s + \mathbf{P} \mathbf{W}_w = \hat{T} + \hat{Z}, \quad (10)$$

where \hat{T} is an estimator of the polynomial trend T at level J , while \hat{Z} is the estimate of residual Z . The issue of choosing the level of the estimate depends on the goal of application. J should be chosen small for detecting the local trends and cycles. In other applications, J is set to be large, if the aim is to detect the global trend.

The orthonormality of the matrix \mathbf{P} implies that the DWT is an energy preserving transform so that

$$\|\mathbf{X}\|^2 = \|\mathbf{W}\|^2 = \sum_{t=1}^N X_t^2 \quad (11)$$

Given the structure of the wavelet coefficients, the energy in \mathbf{X} is decomposed, on a scale by scale basis, via

$$\|\mathbf{X}\|^2 = \|\mathbf{W}\|^2 = \sum_{j=1}^J \|\mathbf{W}_j\|^2 + \|\mathbf{V}_J\|^2 \quad (12)$$

so that $\|\mathbf{W}_j\|^2$ represents the contribution to the energy of $\{X_t\}$ due to changes at scale τ_j .

whereas $\|\mathbf{V}_J\|^2$ represents the contribution due to variations at scale τ_J . So the estimated variance of the time-series in terms of wavelet and scaling coefficients can be expressed as:

$$\begin{aligned} \sigma_X^2 &= \frac{1}{N} \sum_{t=1}^N (X_t - \bar{X})^2 = \frac{1}{N} \|\mathbf{W}\|^2 - \bar{X}^2 = \frac{1}{N} \sum_{j=1}^J \|\mathbf{W}_j\|^2 + \frac{1}{N} \|\mathbf{V}_J\|^2 - \bar{X}^2 \\ &= \sum_{j=1}^J \hat{v}_X^2(\tau_j) + \hat{\sigma}_{S_j}^2 \end{aligned} \quad (13)$$

where $\hat{v}_X^2(\tau_j)$ is the estimated variance of the wavelet coefficients at scale τ_j , and $\hat{\sigma}_{S_j}^2$ is the estimated variance of the trend.

For testing the null hypothesis $H_0: Trend = 0$, Almasri *et al.* (2008) proposed a test statistic that can discriminate between this null hypothesis and the alternative hypothesis $H_1: Trend \neq 0$ is defined as follows:

$$G = \frac{\hat{\sigma}_{S_j}^2}{\sum_{j=1}^J \hat{v}_x^2(\tau_j)} \quad (14)$$

The test statistics $(N - N/2^J)/(N/2^J - 1)G$ will follow an F distributed with $(N/2^J - 1)$ and $(N - N/2^J)$ degrees of freedom shown (under the normality assumption of the scaling coefficients). The distribution of the test statistic is unknown, however, in situations when the errors are not normally distributed and when they exhibit some form of dependency. It is, therefore, important to generate empirical critical values in such situations in order to investigate the properties of the test statistic. This is done by means of simulation experiments.

The wavelet estimate has the advantage over the Fourier transform in terms of the localization in time and frequency, which means that the detail of the estimate is seen to vary with t . This property provides additional information of variability on different scales (different J) in the case when there is a long memory process, because such processes appear to be local trends and cycles, which are, however, disappear after some time.

An important issue is how to choose the wavelet filter. A central factor to use a particular wavelet is to match the characteristics of the series analyzed. The Haar wavelet, which is a piecewise constant function, preserves the discontinuities, and therefore it is most suitable to identify a structural break in the data. By contrast, other wavelets with $L > 2$ are smoother and tend to blur the discontinuities. In general, the wavelets with a wider support (L is big) are smoother but spatially less localized, while the wavelets with a narrow support (L is small) are more spatially localized but less smooth.

6. Basis Functions

Every two dimensional vector (x,y) is a combination of the vectors $(1,0)$ and $(0,1)$. These two vectors are said to be the *basis* vectors for (x,y) because multiplying x by $(1,0)$ yields the vector $(x,0)$ and multiplying y by $(0,1)$ yields the vector $(0, y)$. The sum of these two will yield (x,y) . Extending this theory to functions, the sines and cosines are the basis functions of the Fourier transform. Orthogonality requirement for sines and cosines chosen can be set by choosing appropriate combination of sine and cosine function terms whose

inner product add up to zero. The particular sets of functions that are orthogonal and that construct $f(x)$ are the orthogonal basis functions for the problem (Vidakovic, 1999).

A variety of different wavelet families now exist which enable orthonormal wavelet bases to be generated for a wide class of function spaces. Two wavelet bases *viz*, Haar and Daubechies systems are discussed here.

6.1 The Haar System

The simplest wavelet basis for $L^2(\mathbb{R})$ is the Haar basis. The Haar function is a bonafide wavelet, though not used much in practice, uses a mother wavelet given by

$$\psi(x) = \begin{cases} 1, & 0 \leq x < 1/2, \\ -1, & 1/2 \leq x \leq 1, \\ 0, & \text{otherwise} \end{cases}$$

The Haar wavelet is piecewise constant over intervals of length one-half and can be expressed by a picture as follows (Fig.1).

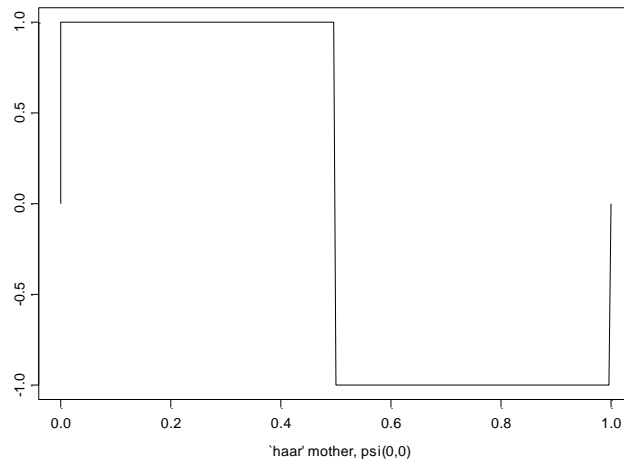


Fig. 1. The Haar function

Haar wavelets possess the property of *compact support*, which means that it will vanish outside of a finite interval. But if instead of a general function in $L^2(\mathbb{R})$, one wants to analyze a function with much less or more regularity, then the expansion given by the Haar system is inappropriate may be due to bad decay of coefficients at infinity. Again the Haar

wavelets are not continuously differentiable. Due to these drawbacks, Haar wavelets are less useful in data analysis. Replacing the scaling function in the Haar system by a more regular function produces a system with a much better behavior with respect to smooth functions. Daubechies (1992) proposed such a family of smooth wavelet basis.

6.2 Daubechies Wavelet Bases

By imposing an appealing set of regularity conditions, Daubechies (1992) came up with a useful class of wavelet filters, all of which yield a DWT in accordance with the notion of differences of adjacent averages. The definition for this class of filters can be expressed in terms of the squared gain function for the associated Daubechies scaling filters g_l , $l = 0, \dots, L-1$:

$$G^D(f) \equiv 2 \cos^L(\pi f) \sum_{l=0}^{L/2-1} \binom{L/2-1+l}{l} \sin^{2l}(\pi f),$$

where L is a positive even integer.

Using the relationship $H^D(f) = G^D(f + 1/2)$, the corresponding Daubechies wavelet filters have squared gain functions satisfying

$$H^D(f) \equiv 2 \sin^L(\pi f) \sum_{l=0}^{L/2-1} \binom{L/2-1+l}{l} \cos^{2l}(\pi f)$$

$H^D(\cdot)$ can be considered as the squared gain function of the equivalent filter for a filter cascade.

Apart from the above, there are other families of smooth wavelet bases that provide compactly supported orthonormal wavelets and are continuously differentiable, like those proposed by Stromberg, Meyer and Battle (Ogden, 1997).

8. An Illustration (Ghosh et al (2010), and Paul et al (2011))

For estimation of trend by wavelet methodology, the Indian monsoon rainfall during the years 1879 to 2006 is considered. The monsoon rainfall is calculated as the sum of daily rainfalls from 1st June to 31st September of a year. The data set is obtained from the website (www.tropmet.res.in) of the Indian Institute of Tropical Meteorology, Pune, India. The rainfall data depicts a cyclical variation with possibly a declining trend. The trend in the monsoon rainfall has been estimated through ARIMA methodology as well as by using

wavelets approach. Different wavelets have been used for analyzing the rainfall data in a scale by scale basis to reveal the localized nature of the data set.

8.1 Modelling of rainfall data in the framework of autoregressive process

Assuming presence of deterministic linear trend in the rainfall series, following model is fitted:

$$Y_t = \mu + \delta t + \varepsilon_t, t = 1, 2, \dots, T \quad (18)$$

where ε_t 's are uncorrelated with zero mean and constant variance σ_ε^2 . Let

$$\hat{\varepsilon}_t = Y_t - \hat{\mu} - \hat{\delta}t$$

The fitted trend equation is obtained as:

$$Y_t = 863.718 - 0.234 t$$

(14.226) (0.191)

where the values within brackets () denote corresponding standard errors of estimates. The trend is not significant at 5% level of significance. The graph of trend is displayed in Fig. 3.

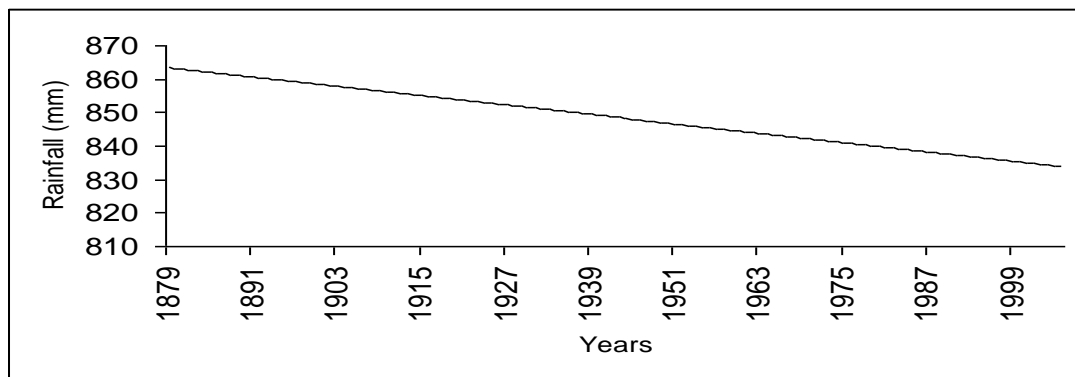


Fig. 3. Trend in Indian monsoon rainfall data

8.2 Trend analysis through wavelet approach

The discrete wavelet transforms and the multiresolution analysis is done on the basis of “Haar” wavelet, and Daubechies 4 (D4) wavelet. The DWT coefficients are shown in Figure 5 and Figure 6. The wavelet coefficients are related to differences (of various order) of (weighted) average values of portions of X_t concentrated in time. Wavelet coefficients are

plotted as bars, up or down. The sizes of the bars are relative to magnitudes of coefficients. The number of wavelet coefficients at the lowest resolution level (level = 1) is exactly half the number of original data points and the number of coefficients decreases by half at each level (Nason and Sachs, 1999).

The coefficients at the top (below) are “high-frequency” (“low frequency”) information. The wavelet coefficients do not remain constant over time and reflects the changes of the data at various time-epochs. The locations of abrupt jumps can be spotted by looking for vertical (between levels) clustering of relatively large coefficients. From the wavelet coefficients plotted above, the original function can be reconstructed by using Inverse discrete wavelet transform (IDWT). The above mentioned pattern can also be verified from the multiresolution analysis (MRA) of the time-series exhibited in Figs. 7 and 8.

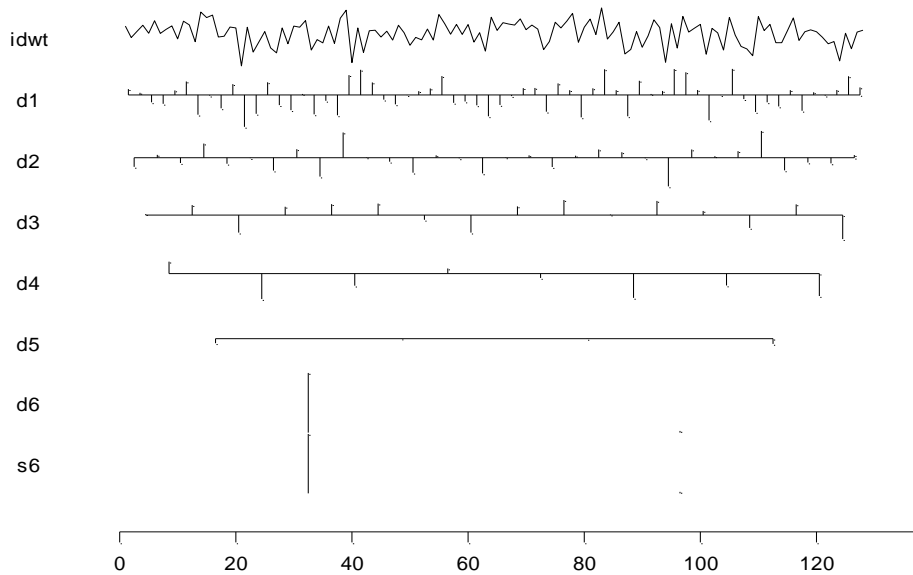


Fig. 5. DWT by D4 wavelet at level 6

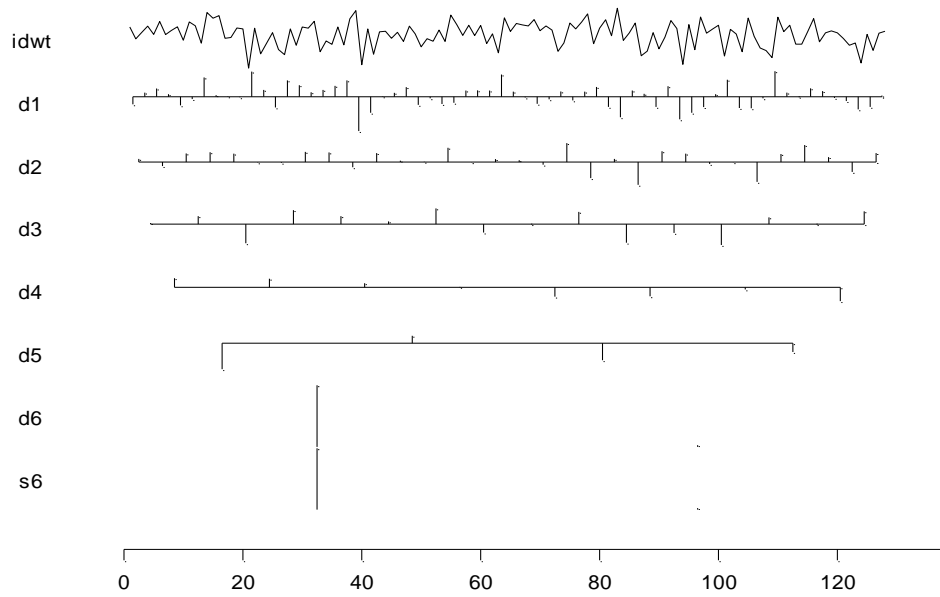


Fig. 6. DWT by Haar wavelet at level 6

The estimate of trend of the rainfall data computed by Haar and D4 wavelets for the levels 6 are given below (Figure 9-10). As the level increases the declining global trend present in the data set is depicted clearly.

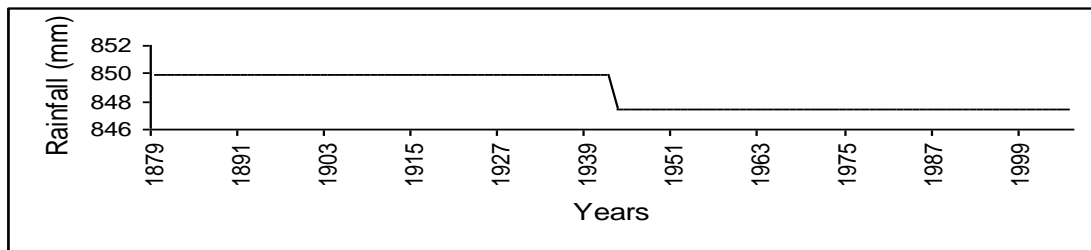


Fig. 9. Estimate of trend by Haar wavelet at level 6

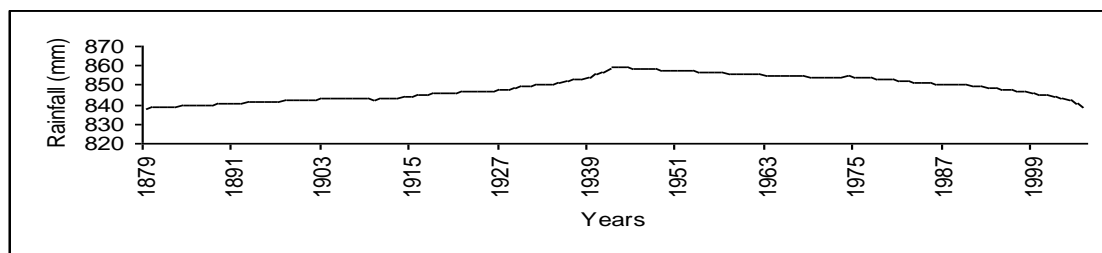


Fig. 10. Estimate of trend by Daubechies (D4) wavelet at level 6

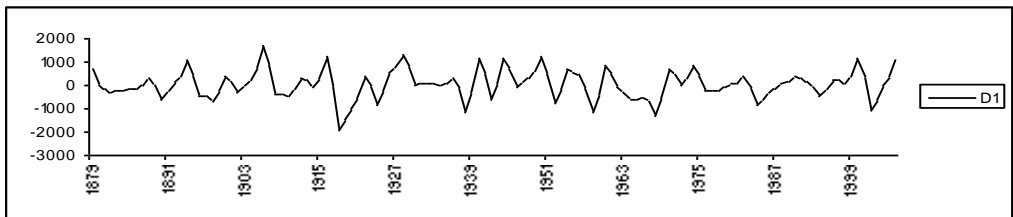
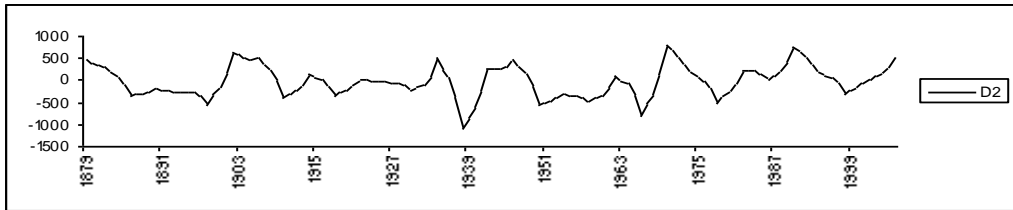
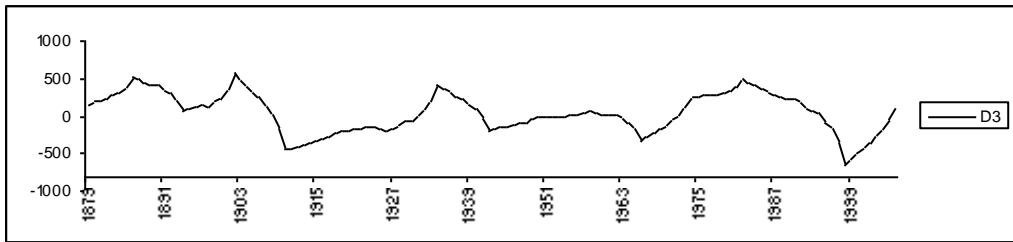
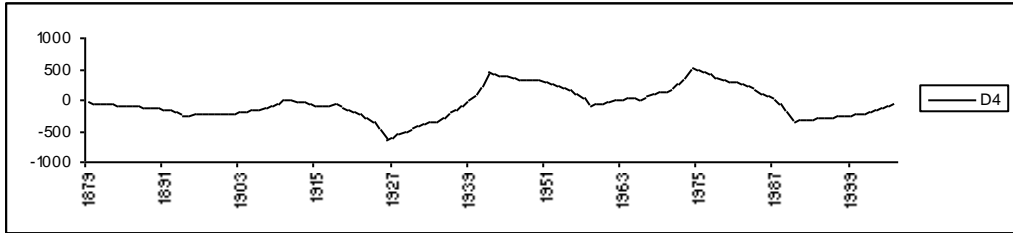
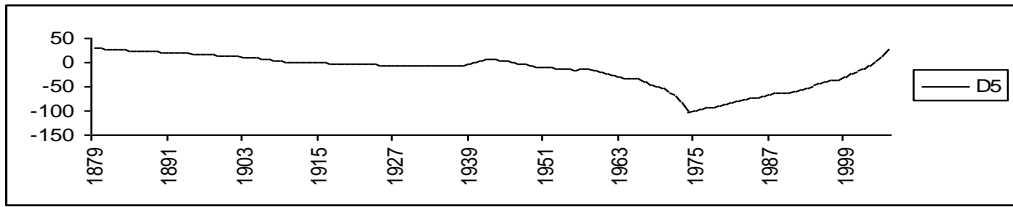
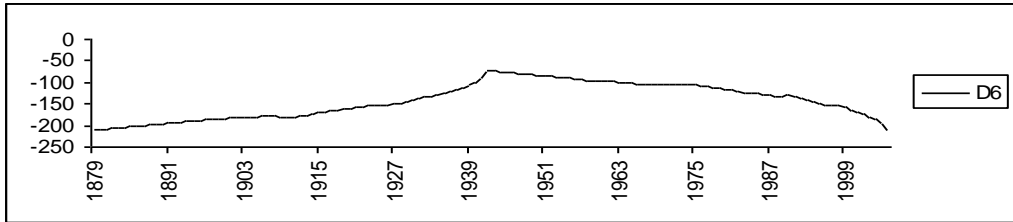
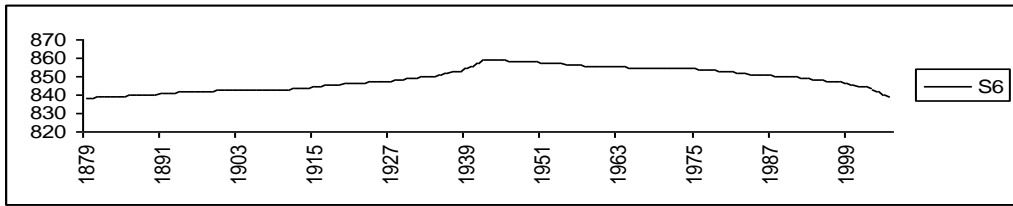


Fig. 7. MRA by D4 wavelet at level 6

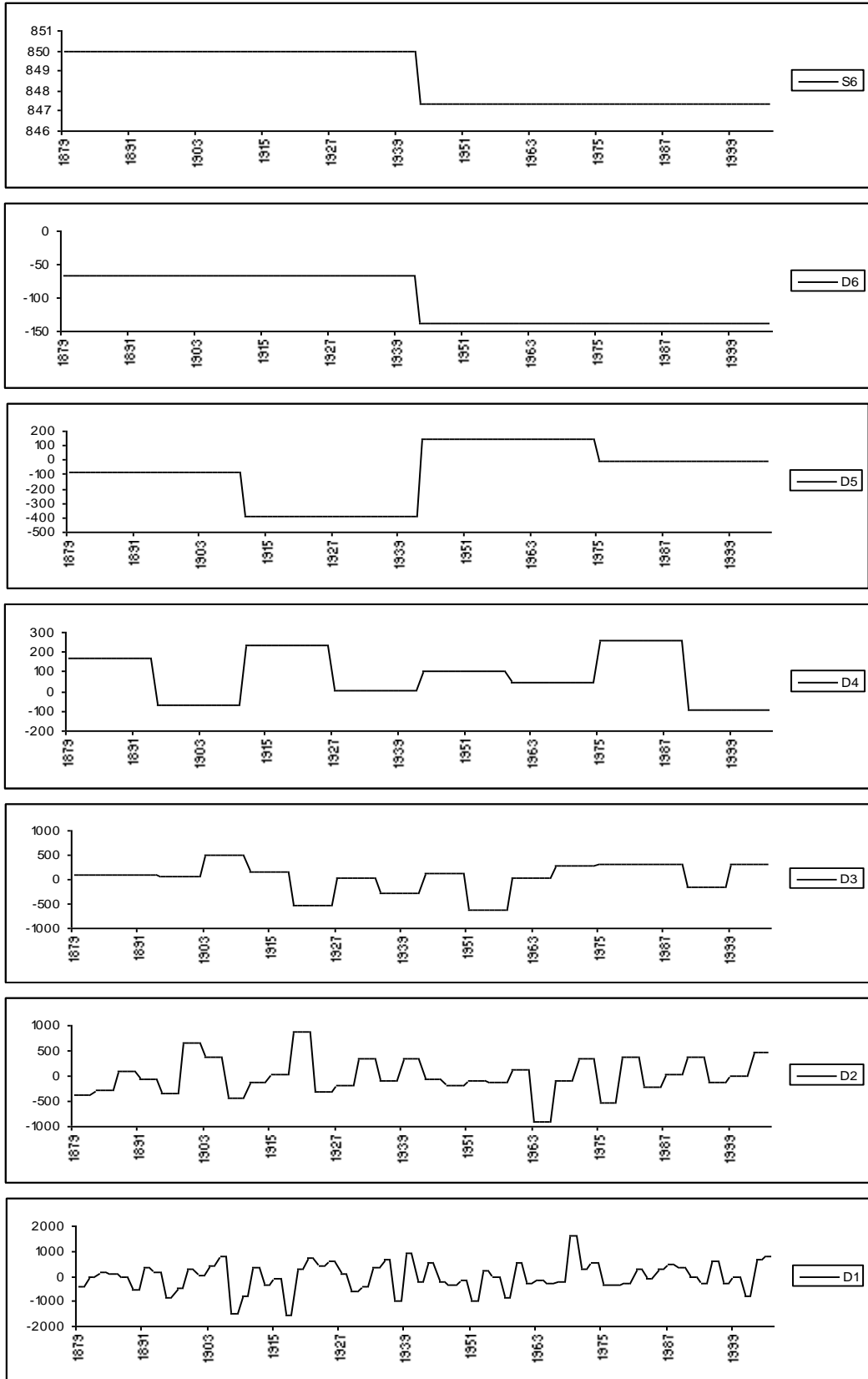


Fig. 8. MRA by Haar wavelet at level 6

The discrete wavelet transform (DWT) and multiresolution analysis (MRA) of India's monsoon rainfall time-series data reveal differential behaviours at different time epochs at different scales. Two wavelets namely; Daubechies (D4) and Haar wavelets are used for estimation of trend in the rainfall data. It is found that the monsoon rainfall in India is showing a declining trend over the years, which can have very serious repercussions from "Global Warming" point of view. This important feature, however, could not be captured by ARIMA methodology.

References:

- Almasri, A., Locking, H. and Shukur, G. (2008). Testing for climate warming in Sweden during 1850–1999, using wavelets analysis. *J. Appl. Stat.*, **35**, 431-43.
- Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (2007). *Time-Series Analysis: Forecasting and Control*. 3rd edition. Pearson education, India.
- Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM, Philadelphia.
- Ghosh, H., Paul, R. K. and Prajneshu, (2010). Wavelet Frequency Domain Approach for Statistical Modeling of Rainfall Time-Series Data. *Journal of Statistical Theory and Practice*, **4** (4)
- Kulkarni, J. R. (2000). Wavelet analysis of the association between the southern oscillation and Indian summer monsoon. *Int. J. Climatol.*, **20**, 89-104.
- Nason, G. P. and von Sachs, R. (1999). Wavelet analysis in time series analysis. *Philosophical Transactions of Royal Society of London, A* **357**, 2511-2526
- Ogden, T. (1997). *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhauser, Boston
- Paul, R. K., Prajneshu, and Ghosh, H. (2011). Wavelet methodology for estimation of trend in Indian monsoon rainfall time-series data. *Indian Journal of Agricultural Science*, **81** (3), 96-98.
- Percival, D. B. and Walden, A. T. (2000). *Wavelet methods for time series analysis*. Cambridge Univ. Press, U.K.

- Rajeevan, M., Pai, D. S., Dikshit, S. K. and Kelkar, R. R. (2004): IMD's new operational models for long – range forecast of southwest monsoon rainfall over India and their verification for 2003. *Curr. Sci.*, **86**, 422 - 31.
- Sunilkumar, G. and Prajneshu (2004). Modelling and forecasting meteorological subdivisions rainfall data using wavelet thresholding approach. *Cal. Stat. Assn. Bull.*, **54**, 255-68.
- Vidakovic, B. (1999). *Statistical Modeling by Wavelets*. John Wiley, New York

Least Absolute Shrinkage and Selection Operator (LASSO)

Dwijesh Chandra Mishra and Sayanti Guha Majumdar
ICAR-IASRI, New Delhi

In a cause and effect relationship, we try to predict the dependent variable (response variable) on the basis of independent variable (explanatory variable). The independent variable is the cause, and the dependent variable is the effect. The models related to it are given below.

1. Linear Least-Squares Regression

Least squares linear regression is a method for predicting the value of a dependent variable Y , based on the value of an independent variable X . The Linear least-square regression is the simplest model, which can be written as

$$y_i = \mu + \sum_{j=1}^p X_{ij}\beta_j + e_i$$

where, $i = 1 \dots n$ individual, $j = 1 \dots p$ predictor variables, y_i is the phenotypic value for individual i , μ is the overall mean, X_{ij} is an element of the incidence matrix corresponding to predictor j , individual i , β_j is regression coefficient associated with predictor j , and e_i is a random residual which follows $N(0, \sigma_e^2)$.

Problems with Linear Least-Squares Regression

This model does not work, if the available number of markers (explanatory variables) is greater than the number of individuals (sample size) available. This problem is commonly known as $p > n$ problem, where p is the number of explanatory variables and n is the sample size. In order to overcome this problem a stepwise procedure of least squares regression can be performed. First, least squares regression analysis was performed on each explanatory variable separately using above model. Then the likelihood of every explanatory variable was plotted against the position of the explanatory variable which helped in identifying the segments having significant effects. Finally, segments having significant effects were used simultaneously by the model to estimate their individual effects. But this approach also has some drawbacks, like it does not fully take advantage of all available variable information as only explanatory variable with a significant effect are included in the final model.

2. Ridge Regression

Ridge regression (Hoerl and Kennard, 1970) is a penalized regression model which has been introduced to overcome the problem of multi-collinearity in the explanatory variable data.

Ridge regression minimizes the penalized sum of squares:

$$|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|^2 + \lambda^2 \boldsymbol{\beta}' \boldsymbol{\beta}$$

where, λ is the penalty parameter, and the estimate of the regression coefficient is given by:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

where, \mathbf{I} is a $p \times p$ identity matrix. The penalty parameter λ can be calculated by several different methods, for example, by plotting $\hat{\boldsymbol{\beta}}$ as a function of λ and choosing the smallest λ that results in a stable estimate of $\hat{\boldsymbol{\beta}}$. Hoerl *et al.* (1975) have proposed another way to choose λ using an automated procedure. The estimate of λ is given by:

$$\lambda = \frac{rs^2}{(\hat{\boldsymbol{\beta}})'(\hat{\boldsymbol{\beta}})}$$

where, r is the number of parameters in the model except the intercept, s^2 is the residual mean square obtained by linear least squares estimation, and $\hat{\boldsymbol{\beta}}$ is the vector of least squares estimates of regression coefficients.

Ridge regression estimator of $\boldsymbol{\beta}$ is biased and this increase in bias is compensated by the decrease in variance. As a result, we get an estimator $\hat{\boldsymbol{\beta}}_R$ with smallest MSE. Another advantage of ridge regression is that it can be used when available markers are more than the sample size to overcome the " $p > n$ " problem. But ridge regression does not set any coefficients to zero and thus does not give any easily interpretable model.

3. Least Absolute Shrinkage and Selection Operator (LASSO)

In order to overcome the problems of least squared regression and ridge regression, LASSO was first introduced by Tibshirani (1996). LASSO stands for Least Absolute Shrinkage and Selection Operator. In this model, the good features of both subset selection and ridge regression were retained. LASSO shrinks some coefficients and set other coefficients to zero. We can write the model for optimization problem as

$$(\hat{\alpha}, \hat{\beta}) = \arg \min \left\{ \sum_{i=1}^N \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\}$$

Subject to

$$\sum_j |\beta_j| \leq t$$

... eq (1)

where, $i = 1 \dots n$ individual, $j = 1 \dots p$ predictor/ explanatory variable, y_i is the phenotypic value for individual i , x_{ij} is an element of the incidence matrix corresponding to predictor j , individual i , β_j is regression coefficient associated with marker j , $t \geq 0$ is a tuning parameter and for all t , the solution for α is $\hat{\alpha} = \bar{y}$. Without loss of generality, it can be assumed that $\bar{y} = 0$ and thus we can omit α from the equation. LASSO is specifically suitable, when the number of predictor variables is larger than the sample size.

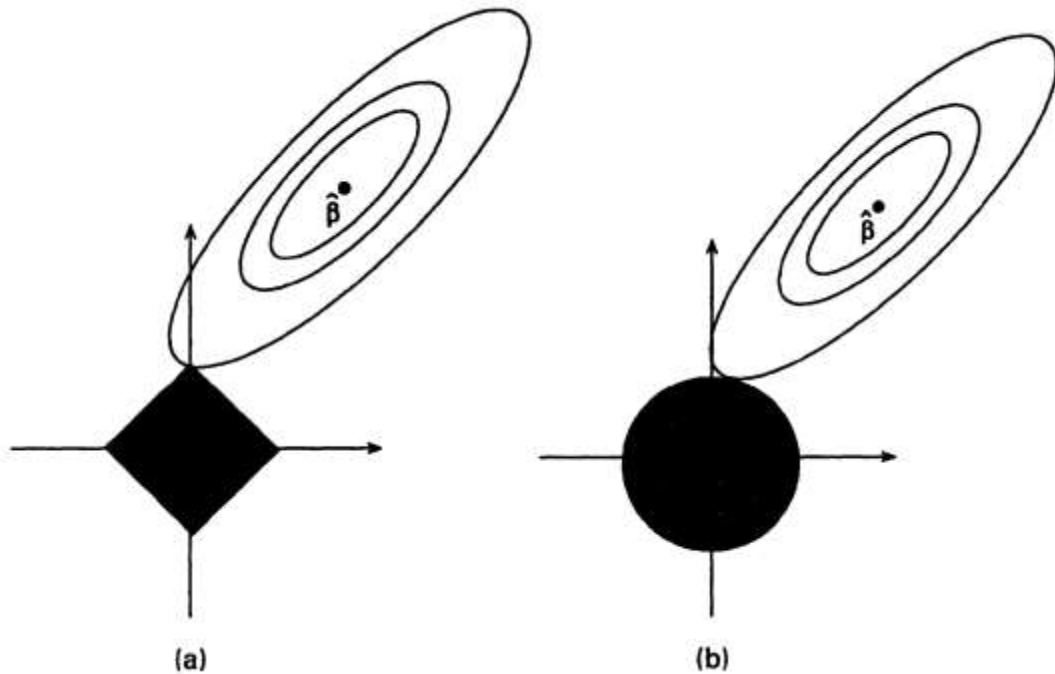


Figure 1: Estimation in case of (a) LASSO and (b) Ridge regression (Tibshirani, 1996)

Figure 1 shows the geometrical solution for lasso and ridge regression. In Figure 1(a), the LASSO solution is the first place that the contours touch the square. If this occur at a corner

then the corresponding predictor has coefficient zero. But in Figure 1(b), there are no corners for the contours to hit and hence we will not get coefficient to be zero.

3.1 Algorithm for finding LASSO solution

If we fix $\lambda \geq 0$, then the eq (1) can be expressed as a least squares problem with 2^p inequality constraints corresponding to the 2^p different possible signs for the β_j s. However, $m = 2^p$ may be very large and the direct application of this procedure is not practical. The problem can be solved by satisfying Kuhn-Tucker conditions (Lawson and Hansen, 1974).

Let $g(\beta) = \sum_{i=1}^N (y_i - \sum_j \beta_j x_{ij})^2$, and let $\delta_i, i = 1, 2, \dots, 2^p$ be the p -tuples of the form $(\pm 1, \pm 1, \dots, \pm 1)$. G_E is the matrix whose rows are δ_i and $E = \{i, \delta_i^T \beta = t\}$, $S = \{i, \delta_i^T \beta < t\}$ where E is the equality set and S is the slack set.

The outline of the algorithm is as follows

- (a) Start with $E = \{i_0\}$ where $\delta_{i_0} = \text{sign}(\hat{\beta}^0)$, $\hat{\beta}^0$ is the overall least squares estimate.
- (b) Find $\hat{\beta}$ to minimize $g(\beta)$ subject to $G_E \beta \leq t1$.
- (c) While $\{\sum |\hat{\beta}_j| > t\}$,
- (d) add i to the set E where $\delta_i = \text{sign}(\hat{\beta})$. Find $\hat{\beta}$ to minimize $g(\beta)$ subject to $G_E \beta \leq t1$.

This procedure must always converge in a finite number of steps since one element is added to set E at each step and there is a total of 2^p elements. The final iterate is the solution to the eq (1).

3.2 Variants of LASSO

There are several variants of LASSO, such as Bayesian LASSO, Fused LASSO, HSIC LASSO, Group LASSO etc. The model for these variants are described below.

(i) Bayesian LASSO

Park and Casella (2008) introduced the Bayesian LASSO method for estimating the regression coefficients by combining LASSO and Bayesian analysis. Tibshirani (1996) noticed that the LASSO estimates of the regression coefficients can be viewed as posterior mode estimates

assuming that the regression coefficients have double exponential prior distributions. The likelihood function can be defined as

$$f(\mathbf{y}|\mu, \mathbf{X}, \boldsymbol{\beta}, \sigma^2) \sim N(\mu + \mathbf{X}\boldsymbol{\beta}, \sigma^2 I)$$

where \mathbf{y} is the $n \times 1$ data vector, μ is the overall mean vector, $\boldsymbol{\beta}$ is a vector of the coefficients associated with each predictor, and \mathbf{X} is the design matrix that connects $\boldsymbol{\beta}$ to \mathbf{y} . $N(\mu + \mathbf{X}\boldsymbol{\beta}, \sigma^2 I)$ denotes the normal density with mean $\mu + \mathbf{X}\boldsymbol{\beta}$ and variance $\sigma^2 I$ where I is an $n \times n$ identity matrix. The prior distribution on the coefficients β_j s $j = 1 \dots p$ can be written as $P(\beta_j|\tau_j^2) \sim N(0, \tau_j^2)$, and the prior distribution on τ_j is $P(\tau_j|\lambda) \sim \text{Exp}(\lambda)$ where $\text{Exp}(\lambda)$ denotes the exponential distribution with rate parameter λ .

(ii) HSIC LASSO or Kernelized LASSO

A feature-wise non-linear LASSO of the following form was proposed by Yamada *et al.* (2014), which is called as HSIC (Hilbert-Schmidt Independence Criterion, Gretton *et al.*, 2005) LASSO.

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{2} \|\bar{\mathbf{L}} - \sum_{k=1}^p \beta_k \bar{\mathbf{K}}^{(k)}\|_{\text{Frob}}^2 + \lambda \|\boldsymbol{\beta}\|_1$$

$$\text{s.t. } \beta_1, \dots, \beta_p \geq 0,$$

where, $\|\cdot\|_{\text{Frob}}$ is the Frobenius norm, $\bar{\mathbf{K}}^{(k)} = \boldsymbol{\Gamma} \mathbf{K}^{(k)} \boldsymbol{\Gamma}$ and $\bar{\mathbf{L}} = \boldsymbol{\Gamma} \mathbf{L} \boldsymbol{\Gamma}$ are centered Gram matrices, $K_{i,j}^{(k)} = K(x_{k,i}, x_{k,j})$ and $L_{i,j} = L(y_i, y_j)$ are Gram matrices, $K(x, x')$ and $L(y, y')$ are kernel functions, $\boldsymbol{\Gamma} = \mathbf{I}_n - \frac{1}{2} \mathbf{1}_n \mathbf{1}_n^T$ is the centering matrix, \mathbf{I}_n is the n -dimensional identity matrix, and $\mathbf{1}_n$ is the n -dimensional vector with all ones. Here, a non-negativity constraint is employed so that meaningful features are selected. As output, Gram matrix \mathbf{L} is used to select features in HSIC LASSO, it is possible to naturally incorporate structured outputs via kernels. Moreover, we can perform feature selection even if the training dataset consists of input x and its affinity information \mathbf{L} link structures between inputs. Differences from the original formulation of LASSO are that in this case kernel functions K and L are different and non-negativity constraint is imposed. The first term in this equation means that we are regressing

the output kernel matrix $\bar{\mathbf{L}}$ by a linear combination of feature-wise input kernel matrices $\{\bar{\mathbf{K}}^{(k)}\}_{k=1}^p$.

(iii) Group LASSO

Yuan & Lin (2007) proposed the group lasso which solves the convex optimization problem

$$\min_{\beta \in \mathbb{R}^p} \left(\left\| \mathbf{y} - \sum_{l=1}^L X_l \beta_l \right\|_2^2 + \lambda \sum_{l=1}^L \sqrt{p_l} \|\beta_l\|_2 \right)$$

where the $\sqrt{p_l}$ terms accounts for the varying group sizes, and $\|\cdot\|_2^2$ is the Euclidean norm (not squared). This procedure acts like the lasso at the group level: depending on λ , an entire group of predictors may drop out of the model. If the group sizes are all one, it reduces to the lasso.

(iv) Fused Lasso

Fused lasso was first proposed by Tibshirani *et al.* in 2005 and is formulated as

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{y} - X^T \beta\|_2^2 + \lambda_1 \sum_{i=1}^p |\beta_i| + \lambda_2 \sum_{i=2}^p |\beta_i - \beta_{i-1}|$$

where $\beta \in \mathbb{R}^p$ and $\lambda_1, \lambda_2 \geq 0$. Furthermore, the variables (i.e., β) are assumed to have a meaningful ordering, like forming a chain structure. Due to the L_1 penalties on both single variables and consecutive pairs, solutions tend to be sparse and smooth, i.e., consecutive variables tend to be similar. The third term is usually called the “fusion penalty”. The classical fused lasso method was proposed to pursue sparse segments on a chain of variables. Thus, a natural generalization of 1D fused lasso aims to promote smoothness over neighboring variables on a general graph.

Application of LASSO:

The applications of LASSO includes

- (i) **Feature selection/ Variable selection**

Variable selection can be performed by using LASSO. Certain predictor variables associated with non-zero coefficients can be selected as most relevant variable in relation to the response variable. HSIC LASSO can be used for non-linear feature (variable) selection (Guha Majumdar *et al.*, 2019).

(ii) Prediction (Genomic prediction)

LASSO can also be used for genomic prediction and selection of breeding material. For detail procedure of genomic prediction reader can refer to Guha Majumdar *et al.*, 2019.

(iii) Forecasting

LASSO Regression can be applied to temperature forecasting (Spencer *et al.*, 2018), Economic forecasting etc.

(iv) LASSO regression model can be used for microRNA-target regulatory network construction (Lu *et al.*, 2011).

(v) Group LASSO can be used for microarray data analysis (Ma *et al.*, 2007).

References:

- Gretton, A., Bousquet, O., Smola, A. and Scholkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. *Algorithmic Learning Theory (ALT)*. pp 63-77. Springer.
- Guha Majumdar, S., Rai, A. and Mishra, D. C. (2019). Identification of genetic markers for increasing agricultural productivity: An empirical study. *Indian Journal of Agricultural Sciences*, **89** (10): 1708–13.
- Guha Majumdar, S., Rai, A. and Mishra, D.C. (2019). Integrated Framework for Selection of Additive and Nonadditive Genetic Markers for Genomic Selection. *Journal of Computational Biology*. <http://doi.org/10.1089/cmb.2019.0223>
- Hoerl, A.E. and Kennard, R.W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**:55-67.
- Hoerl, A. E., Kennard, R.W. and Baldwin, K. F. (1975). Ridge regression: some simulations. *Communications in Statistics*, **4**: 105–123.
- Lawson, C. and Hansen, R. (1974). Solving least squares problems. Prentice-Hall.

- Lu, Y., Zhou, Y., Qu, W., Deng, M. and Zhang, C. (2011). A Lasso regression model for the construction of microRNA-target regulatory networks. *Bioinformatics*, **27**(17):2406–2413. <https://doi.org/10.1093/bioinformatics/btr410>
- Ma, S., Song, X. and Huang, J. (2007). Supervised group Lasso with applications to microarray data analysis. *BMC Bioinformatics*, **8**(60). doi:10.1186/1471-2105-8-60
- Park, T., and Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association*, **103**: 681–686.
- Spencera, B., Alfandi, O. and Al-Obeidat, F. (2018). A Refinement of Lasso Regression Applied to Temperature Forecasting. *Procedia Computer Science*, **130**: 728–735.
- Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of Royal Statistical Society*, **58**:267-288.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005), Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**:91-108. doi:[10.1111/j.1467-9868.2005.00490.x](https://doi.org/10.1111/j.1467-9868.2005.00490.x)
- Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P. and Sugiyama, M. (2014). High-Dimensional Feature Selection by Feature-Wise Kernelized Lasso. *Neural Computation*, **26**:185-207.
- Yuan, M. and Lin, Y. (2007). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, **68**(1):49-67.

Linear and Integer Programming

H.V. HarishKumar, Rajesh T, Shivaswamy G P, Anuja A R
ICAR- IASRI, New Delhi
harishkumar.hv@icar.gov.in

I. Introduction

Linear programming (LP) is a mathematical modeling technique designed to optimize (maximize or minimize) the usage of limited resources. To define “LP is a mathematical technique of studying wherein we consider maximization (or minimization) of a linear expression (called the objective function) subjected to a number of linear equalities and inequalities (called linear restrictions)”.

II. History and application of LP

In 1939, during World War II, a Soviet economist Leonid Kantorovich used LP to plan expenditures and returns in order to reduce costs of the army and to increase losses incurred to the enemy.

LP has wide applications in various fields like military, industry, agriculture, transportation, health system, economics and behavioral sciences etc., and is also utilized for some engineering problems. Transportation, energy, telecommunications, and manufacturing are the major industries that use linear programming models. LP has proven useful in modeling diverse types of problems in planning, routing, scheduling, assignment, and design.

III. Assumptions of linear programming

1. The LP models are “*deterministic*” in nature: Assumes everything is certain and equation is mathematical in nature.
2. The LP models are “*proportional*” in nature: This condition follows directly from linearity assumptions for objective function and constraints. This means that the objective function and constraints expand and contract proportionately to the level of each activity. This condition represents constant returns to scale rather than economies or diseconomies of scale.

3. The LP models are “*additive*” in nature: That is the Left Hand Side (LHS) should be equal to Right Hand Side (RHS). The assumption of proportionality guarantees linearity if and only if the joint effects or interactions are non-existent. That means the total contribution of all activities is identical to sum of the constraints per each activity individually.
4. The decision variables are “*divisible*”: That is the fractional levels for decision variables are permissible, the objective function and constraints are continuous function.
5. Non-negativity: The value of variables must be zero or positive but not negative.

So LP is a special case of mathematical programming to achieve the best outcome (such as maximum profit or minimum cost) in a mathematical model whose requirements are represented by linear relationships.

Here is a simple example.

Reddy Mikks (R-M) company produce both interior and exterior paints from two raw materials M_1 and M_2 . The following table provides the basic data of the problem

Table 1: Basic data of problem

Particulars	Tonnes of raw material required per tonne of		Maximum availability with R-M (tonnes)
	Exterior paint	Interior paint	
Raw material M_1	6	4	24
Raw material M_2	1	2	6
Profit per tonne (\$ 000's)	5	4	-

- The market survey restricts maximum daily demand of interior paints to 2 tonnes.
- Additionally the daily demand for interior paint cannot exceed that of exterior paint more than 1 tonne.
- The R-M company wants to determine the optimum product mix of interior and exterior paints that maximizes total daily profit.

Let us formulate the problem

LP model includes three basic elements

1) **Decision variables** that we seek to determine

X_1 =Production of exterior paints (in tonnes)

X_2 =Production of interior paints (in tonnes)

2) **Objective function** that we aim to optimize

Main objective is to maximize total daily profit

Let Z represents total daily profit (in \$ 000's)

$$\text{Max } Z = 5X_1 + 4X_2$$

3) **The constraints** that we need to satisfy

a) Restriction on raw material usage: The usage of raw material for production of both paints should not exceed raw material availability.

➤ **Usage of raw material M₁: $6X_1 + 4X_2 \leq 24$**

➤ **Usage of raw material M₂: $X_1 + 2X_2 \leq 6$**

b) Demand restrictions

➤ **Maximum daily demand of interior paint is limited to 2 tonnes: $X_2 \leq 2$**

➤ **Excess of daily production (daily demand for interior paint cannot exceed that of exterior paint more than 1 tonne): $X_2 - X_1 \leq 1$**

So the LP model for above optimization problem looks like below

$$\text{Max } Z = 5X_1 + 4X_2$$

Subject to;

$$6X_1 + 4X_2 \leq 24$$

$$X_1 + 2X_2 \leq 6$$

$$X_2 \leq 2$$

$$X_2 - X_1 \leq 1$$

$$X_1 \text{ \& } X_2 \geq 0$$

IV. Standard form of LP model

To solve LP problem manually it must be put in a common form which we call as standard form.

Properties of standard form are

- **All the constraints should be expressed as equations by adding slack or surplus and or artificial variables.**

A constraint of the type \leq (\geq) can be converted to an equation by adding *slack* variable to (subtracting *surplus* variable from) the left side of the constraint.

Ex 1: $3X_1+2X_2 \leq 6$

$3X_1+2X_2 +S_1=6$, where S_1 is a slack variable represents the unused amount of resources

Ex 2: $2X_1+X_2 \geq 6$

$2X_1+X_2 -S_2=6$, where S_2 is a surplus variable represents the excess amount of resources

Note: The introduction of slack and surplus variables alters neither the nature of the constraint nor the objective function. Accordingly such variables are incorporated into objective function with zero co-efficient.

- **The right hand side of each constraint should be made non-negative (if not).**

The RHS of the equation can always be made non-negative by multiplying both the sides by -1.

Ex: $2X_1+3X_2-7X_3=-5$ can be written as $-2X_1-3X_2+7X_3=5$

$2X_1-X_2 \leq -5$ can be written as $-2X_1+X_2 \geq 5$ (the direction of inequality is reversed when both sides are multiplied by -1)

- **The objective function must be maximization type**

Solving of maximization problem is easier than solving of minimization problem. So we can convert minimization form to maximization form for easy calculation and later we can interpret it as minimization solution. The maximization of a function is equivalent to minimization of a negative of the same function and vice-versa.

For a given set of constraints,

Max $Z=5X_1+2X_2+3X_3$ is mathematically **equivalent** to Min $(-Z) = -5X_1-2X_2-3X_3$.

Equivalence means that for the same set of constraints the optimal values of X_1 , X_2 and X_3 are the same in both cases. The only difference is that the values of the objective function, although equal numerically, will appear with opposite signs.

Table 2: General and standard form of LP model involving only less than or equal constraints (\leq)

General form	Standard form
Max $Z = 5X_1+4X_2$ subject to; $6X_1+4X_2 \leq 24$ $X_1+2X_2 \leq 6$ $X_2 \leq 2$ $X_2 - X_1 \leq 1$ $X_1 \& X_2 \geq 0$	Max $Z = 5X_1+4X_2+0S_1+0S_2+0S_3+0S_4$ subject to; $6X_1+4X_2 +S_1 =24$ $X_1+2X_2 + S_2 =6$ $X_2 +S_3=2$ $X_2 - X_1 + S_4=1$ $X_1 , X_2, S_1, S_2, S_3 \& S_4 \geq 0$

V. Artificial variable (AV):

In case of problems with infeasible solution artificially we introduce a variable into objective function to obtain feasible solution. We use AV only to start solution and subsequently force them to be zero in the solution otherwise the resulting solution will be infeasible. To guarantee such assignments in the optimal solution, AVs are incorporated into objective function with very large positive co-efficient in minimization problem or very large negative co-efficient in maximization problem.

AVs do change the nature of constraint since they are added only to one side of inequality. That is if the original constraint is an equation ($=$) or of the type greater than or equal to (\geq), then we have no longer basic starting feasible solution.

Table 3: General and standard form of LP model involving all kind of constraints ($\leq, =, \geq$)

General form	Standard form
Max $Z = 5X_1+2X_2$ subject to;	Max $Z = 5X_1+2X_2+0S_1+0S_2-MA_1-MA_2$ subject to;

$6X_1+X_2 \leq 6$	$6X_1+X_2 +S_1 =6$
$4X_1+3X_2 \geq 12$	$4X_1+3X_2 - S_2+A_1 =12$
$X_1+X_2 =1$	$X_1+X_2 +A_2=2$
$X_1 \& X_2 \geq 0$	$X_1, X_2, S_1, S_2, A_1 \& A_2 \geq 0$

VI. Solution to LP problem:

There are two approaches for solving LP problems.

1) Graphical approach and

2) Simplex technique

1) Graphical approach: LP problems which involve only two decision variables can be solved graphically. Since it is not possible to display the set of feasible solution for more than two variables in a graph for locating best optimal solution. There are two graphical solution methods namely, extreme point solution method and iso-profit (Cost) function line method. Of these, extreme point solution method is most commonly used method for solving LP problem involving two decision variables.

Extreme point solution method: Extreme point refers to corner of the feasible region i.e. the point lies at the intersection of two constraint equations. In this method, the co-ordinates of all corner or extreme points of the feasible region are determined and then value of the objective function at each of these points is computed and compared. The co-ordinates of an extreme point where the optimal (maximum or minimum) value of the objective function is found represent the solution of the given LP problem.

Example:

$$\text{Max } Z = 5X_1 + 7X_2$$

Subjected to:

$$X_1 \leq 6$$

$$2X_1 + 3X_2 \leq 19$$

$$X_1 + X_2 \leq 8$$

$$X_1, X_2 \geq 0$$

Solution:

Here we are not going to add any slack or surplus variable but we are just putting it into equation.

$$X_1=6$$

$$2X_1 + 3X_2 = 19$$

$$X_1 + X_2 = 8$$

$X_1 + X_2 = 8$ Extreme points: a) $X_1=0, X_2=8$, Co-ordinates:(0,8) b) $X_2=0, X_1=8$, Co-ordinates:(8,0)	$2X_1 + 3X_2 = 19$ Extreme points: a) $X_1=0, X_2=6.33$, Co-ordinates: (0,6.33) b) $X_2=0, X_1=9.5$, Co-ordinates: (9.5,0)	$X_1=6$ Extreme points: a) $X_1=6, X_2=0$, (6,0)
--	---	---

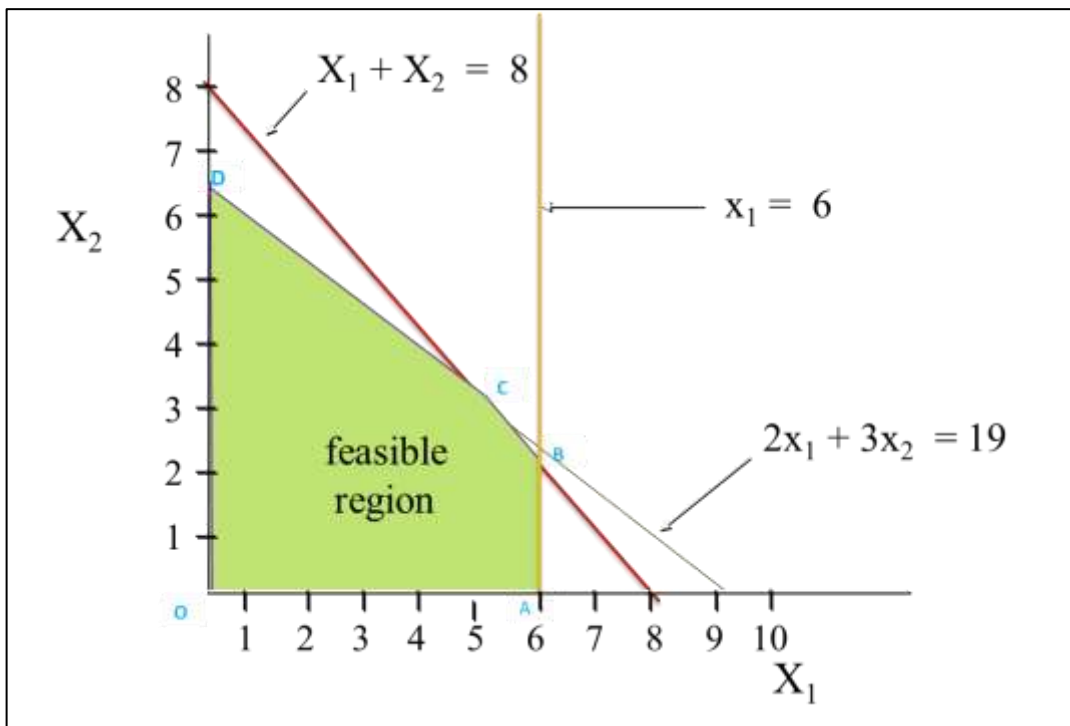


Figure 1: Combined-Constraint Graph Showing Feasible Region

The shaded zone is called feasible area where all the constraints holds good or this region satisfies all constraints so it is called **feasible region**.

- The corners or vertices of the feasible region are referred to as the extreme points.

- An optimal solution to an LP Maximization problem can be found at an extreme point of the feasible region.
- When looking for the optimal solution, you do not have to evaluate all feasible solution points.
- Consider only the extreme points of the feasible region.

Table 4: Value of objective function at extreme points of feasible region

Extreme Point	Co-Ordinates	Z value ($Z=5X_1+7X_2$)
O	(0,0)	0
A	(6,0)	30
B	(6,2)	44
C	(5,3)	46
D	(0,6.33)	44.31

At point C all constraints are satisfied and the Z value is highest hence it is optimal point.

Solution: At $X_1=5$ and $X_2=3$, Max $Z=46$

2) Simplex method:

It is an algorithm adopted to solve LP problem which employs an iterative procedure that starts at a feasible corner point, normally the origin and systematically moves from one feasible point to another point until it reaches optimum point.

Linear programming solvers are now part of many spreadsheet packages, such as Microsoft Excel. The leading commercial package is “LINDO”. We can solve LP problems in packages like “R” and “SAS” also.

VII. Special cases in simplex method of application

1. Degeneracy:

In case of model consisting of at least one redundant (No longer needed or not useful) constraint then the optimum value won't improve upon iterations instead same solution is generated over the iterations.

Example:

$$\text{Max } Z = 3X_1 + 9X_2$$

Subject to;

$$X_1 + 4X_2 \leq 8$$

$$X_1 + 2X_2 \leq 4$$

$$X_1 \text{ \& } X_2 \geq 0$$

In above case the first constraint is a redundant constraint.

2. Alternative optima:

Alternative optima exists when objective function running parallel to one of the constraints. Then the objective function will assume same optimal value at more than one solution point.

Example:

$$\text{Max } Z = 2X_1 + 4X_2$$

Subject to;

$$X_1 + 2X_2 \leq 5$$

$$X_1 + X_2 \leq 4$$

$$X_1 \text{ \& } X_2 \geq 0$$

In above case the objective function runs parallel to first constraint.

3. Unbounded solutions

The solution to a maximization LP problem is unbounded if the value of the solution may be made indefinitely large without violating any of the constraints. Sometimes feasible solution for the given LP problem exists and this has infinite values for the objective function. For real problems, this is the result of improper formulation.

4. Infeasible or non-existent solutions

No unique solution to the LP problem satisfies all the constraints, including the non-negativity conditions. Graphically, this means a feasible region does not exist. Causes includes formulation error, too high expectations by management or too many restrictions have been placed on the problem (i.e. the problem is over-constrained).

VIII. Integer programming:

In case of linear programming, the decision variables considered are supposed to take any real value. However in practical situations it makes no sense in assigning a real value to a variable where it has meaning only when it takes only integer values. To be clear let us consider a practical problem like optimum size of herd in a dairy project, it makes no sense if our optimal value from LP solution is 5.8.

In such situations, we naturally tend to round-off the optimal value to the nearest integer value say “6” in above example. However, the round-off may have following fundamental problems,

- a) The round-off solution may not be feasible.
- b) The objective function value given by the rounded-off solutions (even if some are feasible) may not be the optimal one.
- c) Even if some of the rounded-off solutions are optimal, checking all the rounded-off solutions is computationally expensive.

So integer programming deals with the solution of mathematical programming problems in which some or all the variables can assume non-negative integer values only.

Types of integer programming problems

- 1) Pure integer programming problem: An integer programming problem in which all variables are required to be integers.
- 2) Mixed integer programming problem: If some variables are restricted to be integer and some are not restricted i.e. can be continuous or fractional.
- 3) Binary integer programming problem/ 0-1 programming problems: If some or all variables are restricted to be either “0” or “1”. It can be pure or mixed.

The general form of integer programme is as below

$$\text{Max } Z = 7X_1 + 9X_2$$

subject to;

$$-X_1 + 3X_2 \leq 6$$

$$7X_1 + X_2 \leq 35$$

X_1 & X_2 are non-negative integers.

IX. Applications of Linear Programming in agriculture

Case-1: Naidu Dairy farm uses at least 800 Kg's of *Special feed* daily. The *Special feed* is a mixture of corn silage and soybean meal with the following composition,

Table 5: Constituents of special feed

Feed stuff	In terms of Kg per every Kg of feed stuff		
	Protein	Fiber	Cost (Rs./Kg)
Corn silage	0.09	0.02	20
Soybean meal	0.60	0.06	62

The dietary requirements of *Special feed* must have at least 30 per cent protein and at most 5 per cent fiber. Now the Naidu Dairy farm wishes to determine the daily minimum cost of feed mix?

Solution:

Decision variables:

X_1 = Quantity of corn silage to be used in feed mix (Kg's)

X_2 = Quantity of soybean meal to be used in feed mix (Kg's)

Objective function

$$\text{Min } Z = 20X_1 + 62X_2$$

Constraints

Demand constraint (Daily requirement): $X_1 + X_2 \geq 800$

Protein constraint: $0.09X_1 + 0.60X_2 \geq 0.30(X_1 + X_2)$ on simplification $-0.21 X_1 + 0.30 X_2 \geq 0$

Fiber constraint: $0.02X_1 + 0.06X_2 \leq 0.05 (X_1 + X_2)$ on simplification $-0.03 X_1 + 0.01 X_2 \leq 0$

Overall the LP model looks like

$$\text{Min } Z = 20X_1 + 62X_2$$

Subjected to,

$$X_1 + X_2 \geq 800$$

$$-0.21 X_1 + 0.30 X_2 \geq 0$$

$$-0.03 X_1 + 0.01 X_2 \leq 0$$

$$X_1 \& X_2 \geq 0$$

R code for the above LP problem

```
library(lpSolve)
obj=c(20,62)
mat=matrix(c(1,1,-0.21,0.3,-0.03,0.01), nrow=3, byrow=TRUE)
rhs=c(800,0,0)
dir=c(">=", ">=", "<=")
prod.sol= lp("min", obj, mat, dir, rhs, compute.sens = TRUE)
prod.sol$status
prod.sol$objval
prod.sol$solution
prod.sol$duals
prod.sol$duals.from
prod.sol$duals.to
prod.sol$sens.coef.from
prod.sol$sens.coef.to
```

A) Optimal solution

Z	29835.29
X1	470.58
X2	329.41

The daily minimum cost of feed mix by using 470.58 Kg of corn silage and 329.41 Kg of soybean meal is Rs. 29835.29.

B) Sensitivity analysis

a) Maximum change in resource availability (RHS of binding constraints)

Binding Constraint	Shadow price	RHS	Sensitivity (Range)
Special feed	37.29	800	0 to 1×10^{30}
Protein	82.35	0	-168 to 138

b) Maximum change in marginal cost (Co-efficients of DV's in objective function)

Variable	Value of DV's	Unit price	Sensitivity (Range)
Corn silage (X1)	470.58	20	-43.40 to 62.00
Soybean meal (X2)	329.41	62	20 to 1×10^{30}

X. Applications of Integer Programming in agriculture

Case 2: Venkatesh, a Crop+Dairy farming system based farmer wishes to maximize the total revenue with the available resources. The below table provides the information on the resource availability and the information on resource requirement for the enterprises from his past experience. Ragi being the regular diet of Venkatesh's family he needs minimum 1 acre of his land to be under the same which also serves the fodder security of his dairy. Since the dairy is earning him the regular income for family maintenance he insists at least one cross breed (CB) cow in his farming system.

Resources	Availability	Per unit requirement			
		Tomato	Cabbage	Ragi	CB Cow
Land (Acres)	4	-	-	-	-
Labour (Man days)	350	180	65	32	38
Capital (Rs.)	250000	125000	65000	12500	33000
Water (acre inches)	100	24.5	17.8	9.4	0.5
Returns (Rs.)	-	280000	135000	19000	65000

Solution:

Decision variables:

X_1 = Area under Tomato crop to be taken (Acres)

X_2 = Area under Cabbage crop to be taken (Acres)

X_3 = Area under Ragi crop to be taken (Acres)

X_4 = Number of cross breed cows to be considered in his farming system

Objective function

Max $Z=280000X_1+135000X_2+19000 X_3+65000 X_4$

Constraints

Land constraint (Overall): $X_1+X_2+ X_3\leq 4$

Labour constraint: $180X_1+65X_2+ 32X_3+ 38X_4\leq 350$

Capital constraint: $125000X_1+65000X_2+ 12500X_3+ 33000X_4\leq 250000$

Water constraint: $24.5X_1+17.8X_2+ 9.4X_3+ 0.5X_4\leq 100$

Constraint for Ragi mandate: $X_3\geq 1$

Constraint for Dairy mandate: $X_4\geq 1$

Overall the LP model looks like

Max $Z=280000X_1+135000X_2+19000 X_3+65000 X_4$

Subjected to,

$$X_1+X_2+ X_3\leq 4$$

$$180X_1+65X_2+ 32X_3+ 38X_4\leq 350$$

$$125000X_1+65000X_2+ 12500X_3+ 33000X_4\leq 250000$$

$$24.5X_1+17.8X_2+ 9.4X_3+ 0.5X_4\leq 100$$

$$X_3\geq 1$$

$$X_4\geq 1$$

$$X_1+X_2+ X_3\geq 0 \text{ \& } X_4 \text{ is a non-negative integer}$$

R code for the above LP problem

```
library(lpSolve)
```

```
obj=c(280000,135000,19000,65000)
```

```
mat=matrix(c(1,1,1,0,180,65,32,38,125000,65000,12500,33000,24.5,17.8,9.4,0.5,0,0,1,0,0,0,0,1), nrow=6, byrow=TRUE)
```

```
rhs=c(4,350,250000,100,1,1)
```

```

dir=c("<=", "<=", "<=", "<=", ">=", ">=")
prod.sol= lp("max", obj, mat, dir, rhs, int.vec=4, compute.sens = TRUE)
prod.sol$status
prod.sol$objval
prod.sol$solution
prod.sol$duals
prod.sol$duals.from
prod.sol$duals.to
prod.sol$sens.coef.from
prod.sol$sens.coef.to

```

A) Optimal solution

Z	536713.28
X1	1.37
X2	0.50
X3	1
X4	1

The maximum total revenue that the farmer can achieve is Rs. 5,36,713.3/- by cultivating Tomato, Cabbage and Ragi in 1.37, 0.50 and 1 acre respectively, along with 1 CB cow.

B) Sensitivity analysis

a) Maximum change in resource availability (RHS of binding constraints)

Binding Constraint	Shadow price	RHS	Sensitivity (Range)
Labour	370.62	350	283.20 to 364.48
Capital	1.70	250000	239944.4 to 284847.8
Ragi	-14188.81	1	0 to 1.98
CB cow	-5391.60	1	0 to 2.52

b) Maximum change in marginal profit (Co-efficients of DV's in objective function)

Variable	Value of DV's	Unit price	Sensitivity (Range)
Tomato (X1)	1.37	280000	259615.4 to 373846.15

Cabbage (X2)	0.50	135000	118802.5 to 145600
Ragi (X3)	1	19000	-1*10 ³⁰ to 33188.81
CB cow (X4)	1	65000	-1*10 ³⁰ to 70391.61

X. References

Dorfman, R. 1996. Linear Programming & Economic Annalysis. McGraw-Hill. New York.

Hadley, G. 1997. Linear programming. Narosa publishing house. New Delhi.

Rao, S.S. 2007. Engineering Optimization: Theory and Practice. New Age International Publishers. New Delhi.

Taha, H.A. 2007. Operation Research: In Introduction. Seventh edition. Prentice Hall India. New Delhi.

<https://nptel.ac.in/courses/105108127/>

Autoregressive and Distributed-Lag Models

Rajesh T, H.V. HarishKumar, Anuja A R, and Shivaswamy G P.
ICAR- IASRI, New Delhi
Rajesh.T@icar.gov.in

I. Introduction

The regression analysis involving time series data, which includes not only the current but also the lagged (past) values of the explanatory variables (the X's) is called a **distributed-lag model**. Similarly, regression analysis involving time series data, which includes not only the current but also the lagged (past) values of the dependent variables (the Y's) is called a **autoregressive model**.

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + u_t$$

represents a distributed-lag model, whereas

$$Y_t = \alpha + \beta X_t + \gamma Y_{t-1} + u_t$$

represents an autoregressive model.

Autoregressive models are also known as dynamic models since they portray the time path of the dependent variable in relation to its past value(s). Autoregressive and distributed-lag models are used extensively in econometric analysis, and in the present chapter we will look at such models with a view to study about the role of lags in economics, reasons for the lags and theoretical justification for the commonly used lagged models in empirical econometrics.

II. THE ROLE OF “TIME,” OR “LAG,” IN ECONOMICS

The dependence of a variable Y (the dependent variable) on another variable(s) X (the explanatory variable) is rarely instantaneous in economics. Very often, Y responds to X with a lapse of time and such a lapse of time is called a lag. We consider an example to illustrate the nature of the lag.

EXAMPLE

THE CONSUMPTION FUNCTION

Assume that a person receives a salary increase of ₹ 1000 in annual pay, and suppose that this is a permanent increase. Then what will be the effect of this increase in income on the

annual consumption expenditure of that person? In general, people do not rush to spend all the increase immediately after such a gain in income. Thus, that person may decide to increase consumption expenditure by ₹ 400 in the first year following the income increase, by another ₹ 300 in the next year, and by another ₹ 200 in the following year, saving the remainder. By the end of the third year, the person's annual consumption expenditure will be increased by ₹ 900. We can thus write the consumption function as

$$Y_t = \text{constant} + 0.4X_t + 0.3X_{t-1} + 0.2X_{t-2} + u_t \dots\dots\dots(1.1)$$

Where, Y is consumption expenditure and X is income.

The above equation shows that the effect of an increase in income of ₹ 1000 is distributed over a period of 3 years. Such models are therefore called as distributed-lag models because the effect of a given cause (income) is spread over a number of time periods. In general we may write

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \dots + \beta_k X_{t-k} + u_t \dots\dots\dots(1.2)$$

which is a distributed-lag model with a finite lag of k time periods. The coefficient β_0 is known as the short-run, or impact, multiplier because it gives the change in the mean value of Y following a unit change in X in the same time period. If the change in X is maintained at the same level thereafter, then, $(\beta_0 + \beta_1)$ gives the change in (the mean value of) Y in the next period, $(\beta_0 + \beta_1 + \beta_2)$ in the following period, and so on. These partial sums are called interim, or intermediate, multipliers. Finally, after k periods we obtain

$$\sum_{i=0}^k \beta_i = \beta_0 + \beta_1 + \beta_2 + \dots + \beta_k = \beta \dots\dots\dots(1.3)$$

which is known as the long-run, or total, distributed-lag multiplier, provided the sum β exists.

If we define

$$\beta_i^* = \frac{\beta_i}{\sum \beta_i} = \frac{\beta_i}{\beta} \dots\dots\dots(1.4)$$

we obtain “standardized” β_i . Partial sums of the standardized β_i then give the proportion of the long-run, or total, impact felt by a certain time period.

Going back to the consumption regression (1.1), we can see that the short-run multiplier, which is nothing but the short-run marginal propensity to consume (MPC), is 0.4, whereas the long-run multiplier, which is the long-run MPC, is $0.4 + 0.3 + 0.2 = 0.9$. That is, following a ₹ 1 increase in income, the consumer will increase his or her consumption level by about 40 paise in the year of increase, by another 30 paise in the next year, and by yet another 20 paise in the following year. The long-run impact of an increase of ₹ 1 in income is thus 90 paise. If we divide each β_i by 0.9, we obtain, respectively, 0.44, 0.33, and 0.23, which indicate that 44 percent of the total impact of a unit change in X on Y is felt immediately, 77 percent after one year, and 100 percent by the end of the second year.

III. ESTIMATION OF DISTRIBUTED-LAG MODELS

Consider the following distributed-lag model in one explanatory variable:

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \dots + u_t \dots \dots \dots (1.5)$$

where we have not defined the lag length, that is, how far back into the past we want to go. These type of model is called as infinite (lag) model, whereas a model of the type as shown in equation 1.2 is called as finite (lag) distributed-lag model, where the lag length k is specified.

We can adopt two approaches to estimate the α and β 's of equation 1.5.

- (1) ad hoc estimation and
- (2) a priori restrictions on the β 's (Assumption: β 's follow some systematic pattern).

We will discuss ad hoc estimation in this section.

Ad Hoc Estimation of Distributed-Lag Models

As the explanatory variable X_t is assumed to be non-stochastic, X_{t-1} , X_{t-2} , and so on, are non-stochastic, too. Therefore, the ordinary least squares (OLS) can be applied to (1.5). This is the approach taken by Alt and Tinbergen. They suggest that one may proceed sequentially to estimate (1.5); that is, first regress Y_t on X_t , then regress Y_t on X_t and X_{t-1} , then regress Y_t on X_t , X_{t-1} , and X_{t-2} , and so on. We need to stop this sequential procedure when the

regression coefficients of the lagged variables start becoming statistically insignificant and/or the coefficient of at least one of the variables changes signs from positive to negative or vice versa. Based on this principle, Alt regressed fuel oil consumption Y on new orders X on the quarterly data for the period 1930–1939, and the results were as follows:

$$\hat{Y}_t = 8.37 + 0.171X_t$$

$$\hat{Y}_t = 8.27 + 0.111X_t + 0.064X_{t-1}$$

$$\hat{Y}_t = 8.27 + 0.109X_t + 0.071X_{t-1} - 0.055X_{t-2}$$

$$\hat{Y}_t = 8.32 + 0.108X_t + 0.063X_{t-1} + 0.022X_{t-2} - 0.020X_{t-3}$$

Alt chose the second regression as the “best” one because in the last two equations the sign of X_{t-2} was not stable and in the last equation the sign of X_{t-3} was negative, which may be difficult to interpret economically.

THE KOYCK APPROACH TO DISTRIBUTED-LAG MODELS

Koyck has proposed a new method of estimating distributed-lag models. If we assume that the β 's are all of the same sign in the infinite lag distributed-lag model (1.5), Koyck assumes that they decline geometrically as follows.

$$\beta_k = \beta_0 \lambda^k \quad k = 0, 1, \dots \dots \dots (1.6)$$

where λ , such that $0 < \lambda < 1$, is known as the rate of decline, or decay, of the distributed lag and $1 - \lambda$ is known as the speed of adjustment.

What (1.6) postulates is that each successive β coefficient is numerically less than each preceding β (since $\lambda < 1$), implying that as one goes back into the distant past, the effect of that lag on Y_t becomes progressively smaller. After all, current and recent past incomes are expected to affect current consumption expenditure more heavily than income in the distant past. Geometrically, the Koyck scheme is depicted in Figure 1.

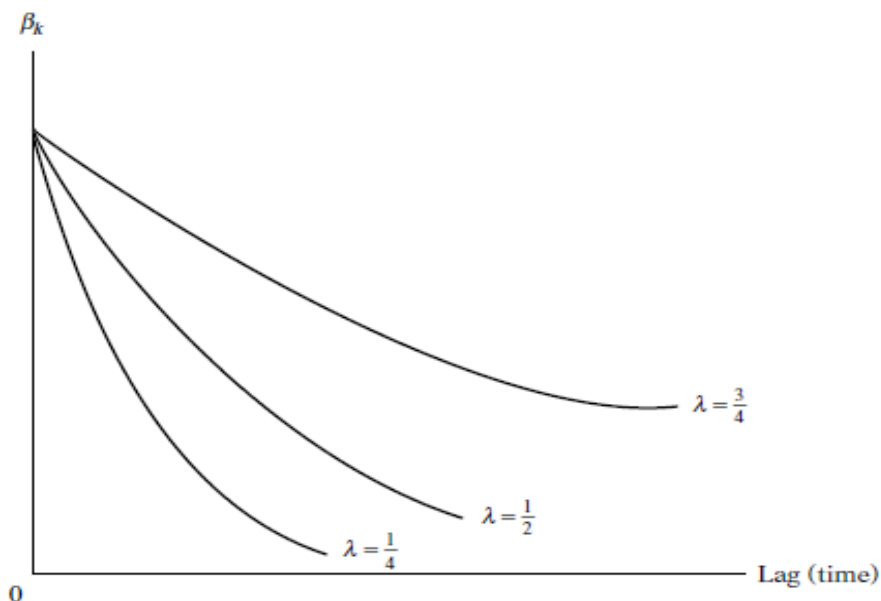


FIGURE 1 Koyck scheme (declining geometric distribution).

As we can see from the figure that value of the lag coefficient β_k depends on both the common β_0 and the value of λ . The closer the value of λ to 1, the slower the rate of decline in β_k , whereas the closer it is to zero, the more rapid the decline in β_k . Distant past values of X in the former case will exert sizable impact on Y_t , whereas their influence on Y_t in the latter case will diminish quickly. This pattern can be seen from the following illustration:

λ	β_0	β_1	β_2	β_3	β_4	β_5	...	β_{10}
0.75	β_0	$0.75\beta_0$	$0.56\beta_0$	$0.42\beta_0$	$0.32\beta_0$	$0.24\beta_0$...	$0.06\beta_0$
0.25	β_0	$0.25\beta_0$	$0.06\beta_0$	$0.02\beta_0$	$0.004\beta_0$	$0.001\beta_0$...	0.0

Note these features of the Koyck scheme: (1) By assuming non-negative values for λ , Koyck rules out the β 's from changing sign; (2) by assuming $\lambda < 1$, he gives lesser weight to the distant β 's than the current ones; and (3) he ensures that the sum of the β 's, which gives the long-run multiplier, is finite, namely,

$$\sum_{k=0}^{\infty} \beta_k = \beta_0 + \left(\frac{1}{1-\lambda}\right)$$

As a result of (1.6), the infinite lag model (1.5) may be written as

$$Y_t = \alpha + \beta_0 X_t + \beta_0 \lambda X_{t-1} + \beta_0 \lambda^2 X_{t-2} + \dots + u_t \quad \dots \dots \dots (1.7)$$

The model is still not amenable to easy estimation since a large (literally infinite) number of parameters remain to be estimated and the parameter λ enters in a highly nonlinear form: Strictly speaking, the method of linear regression analysis cannot be applied to such a model. But now Koyck suggests an ingenious way out. According to his model, we need to lag (1.7) by one period to obtain

$$Y_{t-1} = \alpha + \beta_0 X_{t-1} + \beta_0 \lambda X_{t-2} + \beta_0 \lambda^2 X_{t-3} + \dots + u_{t-1} \quad \dots \dots \dots (1.8)$$

Then multiply (1.8) by λ to obtain

$$\lambda Y_{t-1} = \lambda \alpha + \lambda \beta_0 X_{t-1} + \beta_0 \lambda^2 X_{t-2} + \beta_0 \lambda^3 X_{t-3} + \dots + \lambda u_{t-1} \quad \dots \dots \dots (1.9)$$

By subtracting (1.9) from (1.7), we will get

$$Y_t - \lambda Y_{t-1} = \alpha(1 - \lambda) + \beta_0 X_t + (u_t - \lambda u_{t-1}) \quad \dots \dots \dots (1.10)$$

or, rearranging,

$$Y_t = \alpha(1 - \lambda) + \beta_0 X_t + \lambda Y_{t-1} + v_t \quad \dots \dots \dots (1.11)$$

Where $v_t = (u_t - \lambda u_{t-1})$, a moving average of u_t and u_{t-1} .

The procedure described above is known as the Koyck transformation. By comparing (1.11) with (1.5), we can see the tremendous simplification accomplished by Koyck. Whereas before we had to estimate α and an infinite number of β 's, but now we have to estimate only three unknowns: α , β_0 , and λ . Now there is no reason to expect multicollinearity. In a sense multicollinearity is resolved by replacing X_{t-1} , X_{t-2} , \dots , by a single variable, namely, Y_{t-1} .

The partial sums of the standardized β_i tell us the proportion of the long-run, or total, impact felt by a certain time period. In general, the mean or median lag is often used to characterize the nature of the lag structure of a distributed lag model.

➤ **The Median Lag**

The median lag is the time required for the first half, or 50 percent, of the total change in Y following a unit sustained change in X. For the Koyck model, the median lag is as follows

$$\text{Koyck model: Median lag} = -\frac{\log 2}{\log \lambda}$$

Thus, the median lag is 0.4306 if $\lambda = 0.2$, but the median lag is 3.1067 if $\lambda = 0.8$. In the former case 50 percent of the total change in Y is accomplished in less than half a period, whereas in the latter case it takes more than 3 periods to accomplish the 50 percent change.

➤ **The Mean Lag**

Provided all β_k are positive, the mean, or average, lag is defined as

$$\text{Koyck model: Mean lag} = -\frac{\lambda}{1 - \lambda}$$

Thus, if $\lambda = 0.5$, the mean lag is 1. The median and mean lags serve as a summary measure of the speed with which Y responds to X.

EXAMPLE

PER CAPITA PERSONAL CONSUMPTION

This example studies per capita personal consumption expenditure (PPCE) in relation to per capita disposable income (PPDI) in India for the period 1972–2018. As an illustration of the Koyck model, consider the data given in the table (Gujarati et al., 2012).

Year	PPCE	PPDI	PPCE(-1)	Year	PPCE	PPDI	PPCE(-1)
1972	7639	8501	7542	1996	15109	17777	14422
1973	7639	8667	7639	1997	15806	18242	15109
1974	7936	8993	7639	1998	16336	18672	15806
1975	8178	9221	7936	1999	16760	18838	16336
1976	8605	9827	8178	2000	17320	19506	16760
1977	9097	10360	8605	2001	17664	19886	17320
1978	9559	10831	9097	2002	17833	20047	17664
1979	9760	11223	9559	2003	17614	19875	17833
1980	10276	11658	9760	2004	17974	20314	17614
1981	10586	11929	10276	2005	18359	20259	17974
1982	10721	12329	10586	2006	18848	20578	18359
1983	11022	12767	10721	2007	19148	20919	18848
1984	11634	13278	11022	2008	19601	21312	19148
1985	12137	14111	11634	2009	20131	21831	19601
1986	11914	13860	12137	2010	20949	22897	20131
1987	12086	14057	11914	2011	21816	23330	20949

1988	12685	14504	12086	2012	22626	24235	21816
1989	13130	14894	12685	2013	22971	24453	22626
1990	13603	15470	13130	2014	23378	24983	22971
1991	13796	15697	13603	2015	23809	25301	23378
1992	13582	15706	13796	2016	24477	25998	23809
1993	13645	15983	13582	2017	25043	26202	24477
1994	13710	16184	13645	2018	25594	26771	25043
1995	14422	16594	13710				

The result of regression of Per Capita Consumption Expenditure (PPCE) on Per Capita Personal Disposal Income (PPDI) and lagged PPCE is as follows:

Dependent Variable: PPCE

Method: Least Squares

Sample (adjusted): 1972-2018

<i>Regression Statistics</i>	
Multiple R	0.999105
R Square	0.998211
Adjusted R Square	0.998129
Standard Error	225.1053
Observations	47

	<i>Coefficients</i>	<i>Standard Error</i>	<i>t Stat</i>	<i>P-value</i>
Intercept	-237.251	153.9967	-1.54063	0.130569
PPDI	0.2132	0.070481	3.024651	0.004145
PPCE(-1)	0.7978	0.073183	10.9018	4.36E-14

Adjusted R Square is 99.82 percent, which indicates that 99.82 percent of the variation in PPCE is explained by PPDI and PPCE Lag. From the regression analysis, we found that $\beta_0 = 0.2132$ and $\lambda = 0.7978$. β_0 gives the short run (immediate) effect i.e. if PPDI increases by 1 percent then PPCE will increase by 0.2132 percent in the same year. Long run multiplier is given by the following equation;

$$\text{Long run multiplier} = \beta_0 \left(\frac{1}{1-\lambda} \right) \approx 1.0537$$

In words, a sustained increase of 1 rupee in PPDI will eventually lead to about 1.05 rupees increase in PPCE. The long-run consumption function can be written as:

$$PPCE_t = -1247.1351 + 1.0537(PPDI_t)$$

It can be obtained by dividing the short-run consumption function by 0.2029 and dropping the lagged PPDI term. The median lag is given by;

$$\text{Median lag} = -\frac{\log 2}{\log \lambda} = -\frac{\log(2)}{\log(0.7971)} = 3.0589$$

i.e. 50 percent of this total effect of increase in PPDI on PPCE is felt after 3 years.

IV. References

Alt. F. F., 1942, Distributed Lags, *Econometrica*, **10**: 113-128

Gujarati. D. N., Porter. D. C. and Gunashekar. S., 2012, Basic Econometrics (Fifth edition).
Mc Graw Hill Education (India) Private Limited, New Delhi, pp: 656-713.

Tinbergen. J., 1949, Long-Term Foreign Trade Elasticities, *Metroeconomica*, **1**: 174-185