Er. PERUMAL MANIMEKALAI COLLEGE OF ENGINEERING

Accredited by NAAC ('A' Grade) & NBA (B.E. - CSE | ECE | EEE | MECH & B.TECH. - IT)

AN AUTONOMOUS INSTITUTION

NAME: ………………………….. BRANCH:………………………

YEAR: ………………………….. SEM :………………………

PROGRAMME: ……..…………………………………………………………..

REGISTER NO:

Certified that this bonafide record of the work done by the above Student of the

…………………………………………………………………………………..

Laboratory during the year 20 - 20

STAFF IN CHARGE **HEAD OF THE DEPARTMENT**

Submitted for the Evamination held on..at

Er. Perumal Manimekalai College of Engineering (Autonomous), Hosur.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Vision and Mission of the Institute

Vision:

PMC TECH strives to achieve excellence in technical Education through innovative teaching, learning and applied Multidisciplinary research with professional and ethical practices.

Mission:

PMC TECH will Endeavour

- To become the state of art teaching and learning center for Engineering and Technology, Research and Management Studies
- To have world class infrastructure for providing quality education and research towards creativity, self - discipline and ethical values
- To associate with industry, R&D and business organizations and to have connectivity with the society
- To create knowledge, based professionals to enrich their quality of life by empowering self and family.

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Vision and Mission of the Department

Vision

To develop innovative computer science professional with business management knowledge & skills.

Mission

- Develop Computer Science & business management graduates to meet the requirement of employment, research & entrepreneurship.
- Produce highly competent professional to provide solutions to the modern industrial & societal need.
- Train them for lifelong learning with ethical value to contribute for the sustainable development of our nation.

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Program Educational Objectives (PEOs)

1. Graduates will be able to perform in technical/managerial roles by thorough understanding of contemporary technologies as well as apply technology abstraction of computer science & business systems.

2. To enrich and enable graduates with the core competencies necessary for applying knowledge of computer science and Data analytics tools to store, retrieve, implement and analyze data in the context of business enterprise.

3. To enable graduates to gain employment in organizations and establish themselves as professionals by applying their technical skills and leadership qualities to solve real world problems and meet the diversified needs of industry, academia and research.

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Program Specific Outcomes (PSOs)

1. Analyze, design, and develop solutions in the areas of business process management to build quality products for industry and societal needs.

2. Innovate ideas and solutions for real time problems in the field of software developing applications by adapting emerging technologies and tools.

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Program Outcomes (POs)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

EX NO: 1	Explore the features of MS-EXCEL
Date:	

AIM:

To explore the features of MS-EXCEL

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office
Step 2: Enter text or a number in a cell
Step 3: Change the width of a column
Step 4: Enter And Edit Formula in Excel
Step 5: Wrap text in a cell
Step 6: Enter And Edit Formula in Excel
Step 7: Perform Auto fill and custom fill In Excel
Step 8: Save the file in desired location
Step 9: Close the Ms Excel application

Features of MS-EXCEL

What is Excel? Excel Definition

Microsoft's Excel spreadsheet programme is a part of the Office family of business software programmes. Users of Microsoft Excel can format, arrange, and compute data in a spreadsheet. By organising data using tools like Excel, Data Analysts or other users can make information easier to examine when data is added or altered. The Microsoft Office and Office 365 suites include Excel, which works with the other Office programmes. The spreadsheet application can be used on Windows, macOS, Android, and iOS devices.

Parts of MS-Excel window:

Quick Access Toolbar – Collection of buttons that provide one click access to commonly used commands such as Save, Undo or Redo. You can also customize this according to your preference.

Title Bar – A bar the display the name of active workbook

Ribbon – The main set of commands and controls organized task in Tabs and groups, you can also customize the ribbon according to your preference.

Column Headings – The letters that appear along the top of the worksheet to identify the different columns in the worksheet.

Worksheet Window – A window that displays an Excel worksheet, basically this is where you work all the tasks.

Vertical Scroll Bar – Scroll bar to use when you want to scroll vertically through the Worksheet window.

Horizontal Scroll Bar – Scroll bar to use when you want to scroll horizontally through the worksheet window.

Zoom Controls – Used for magnifying and shrinking of the active worksheet.

View Shortcuts – Buttons used to change how the worksheet content is displayed. Normal, Page Layout or Page Break Preview.

Sheet Tabs – Tabs the display the name of the worksheet in the workbook, by default its name sheet 1, sheet 2, etc. You can rename this to any name the best represents to your sheet.

Sheet Tab Scrolling Buttons – Buttons to scroll the sheet tabs in the workbook

Row Headings – The number that appears on the left of the worksheet window to identify the Different rows.

Select All Button – A button that selects all the cells in the active worksheet Active Cell – The cell selected in the active worksheet

Name Box – A box that displays the cell reference of the active cell.

Formula Bar – A bar that displays the value or formula entered in the active cell.

Office Button/File Tab – It provides access to workbook level features and program settings. You will notice that in Excel 2007 there is a circle

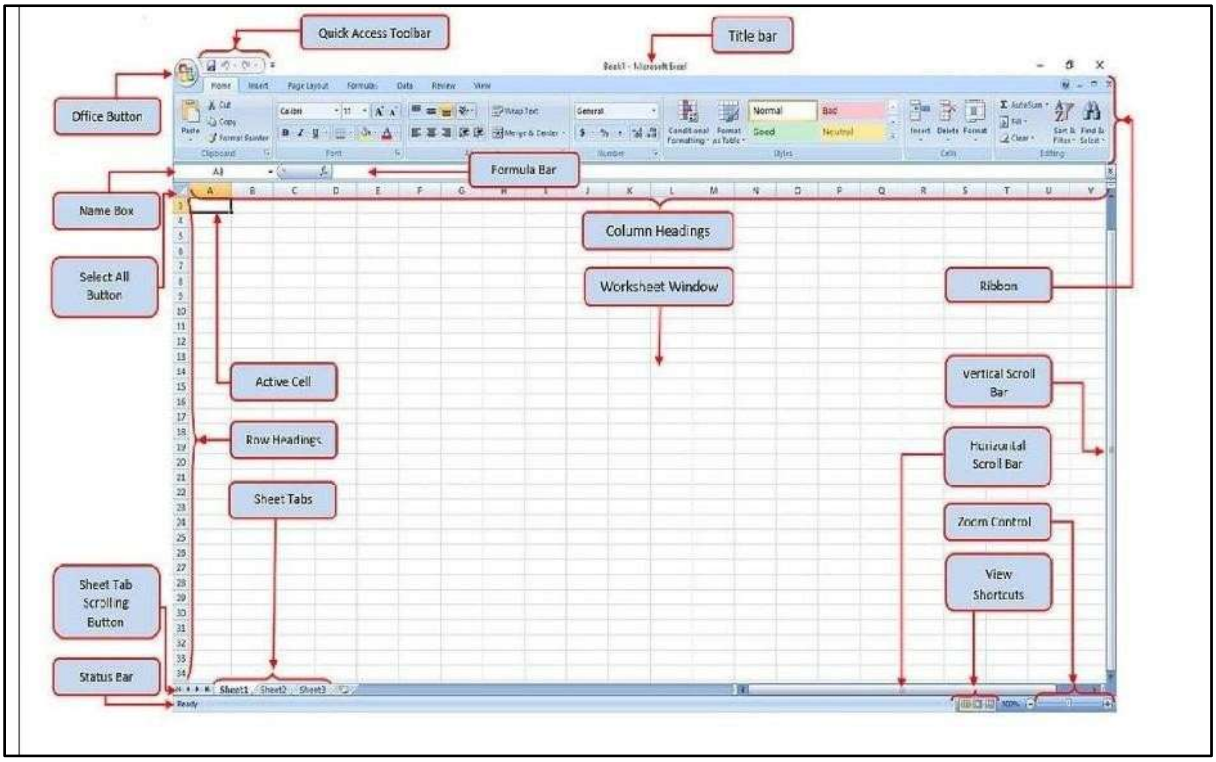
Features of Ms-Excel

Microsoft excel is an integrated electronic spread sheet program developed by Microsoft corporation.It includes the following features.

Autocalc: This feature is very useful to sum a group of numbers is selected them. Their sum will automatically appear in the status area.

Auto complete: Excel now intelligence to anticipate what you are going to type! Based upon entries you've already made, AutoComplete will try to figure out what you intended to type, once you've entered few letters.

Autocorrect: Excel can support automatically correct mistakes.



Better Drag-and-Drop: Do you want to move a group of cells? Excel's drag and drop feature lets you reposition selected portion of your spreadsheet by simply dragging them with your mouse.

Cell tips and Scroll Tips: To help you get around better with mouse, Excel now includes scroll tips. When you click and drag a scroll bar, a small window tells you what row or column you are heading for.

Number Formatting: It's easy to format numbers with excel's new number formatting feature. Select your numbers and choose cells command from format menu.

Templates and Template wizard: Excel's template facility has been greatly enhanced. You can choose from a variety of elegantly designed templates for your home or business. You can even have a template wizard link your worksheets to a database.

Shared Lists: you can now have worksheets that are shared simultaneously over a network.

Conditional Formatting: Conditional formatting helps users to quickly focus on important aspects of a spreadsheet or to highlight errors and to identify important patterns in data.

Sorting and Filtering: Excel spreadsheets help us make sense of large amounts of data. To make it easier to find what you need, you can reorder the data or pick out just the data you need, based on parameters you set within Excel. Sorting and filtering your data will save you time and make your spreadsheet more effective.

Excel Charts: Excel charts help you communicate insights & information with ease. By choosing your charts wisely and formatting them cleanly, you can convey a lot.

ENTERING AND EDITING DATA IN WORKSHEET

You have several options when you want to enter data manually in Excel. You can enter data in one cell, in several cells at the same time, or on more than one worksheet at the same time. The data that you enter can be numbers, text, dates, or times. You can format the data in a variety of ways. And, there are several settings that you can adjust to make data entry easier for you.

Enter text or a number in a cell

1. On the worksheet, click a cell.
2. Type the numbers or text that you want to enter, and then press Enter or Tab.

To enter data on a new line within a cell, enter a line break by pressing

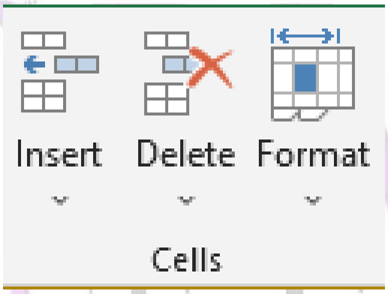
Alt + Enter Editing text or a number in a cell

1. Double click the cell containing the data you want to edit.
2. Make any changes to the cell contents.
3. Press enter key. The change will accept. To cancel your changes, press Ese key.

Change the width of a column

a. Click the cell for which you want to change the column width.

b. On the Home tab, in the Cells group, click Format



c. Under Cell Size, do one of the following:

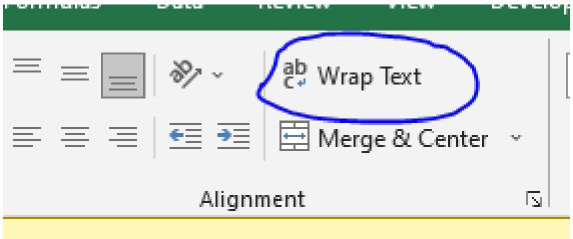
- To fit all text in the cell, click AutoFit Column Width.
- To specify a larger column width, click Column Width, and then type the width that you want in the Column width box.

If there are multiple lines of text in a cell, some of the text might not be displayed the way that you want. You can display multiple lines of text inside a cell by wrapping the text.

Wrap text in a cell

a. Click the cell in which you want to wrap the text.

b. On the Home tab, in the Alignment group, click Wrap Text.



Enter And Edit Formula in Excel

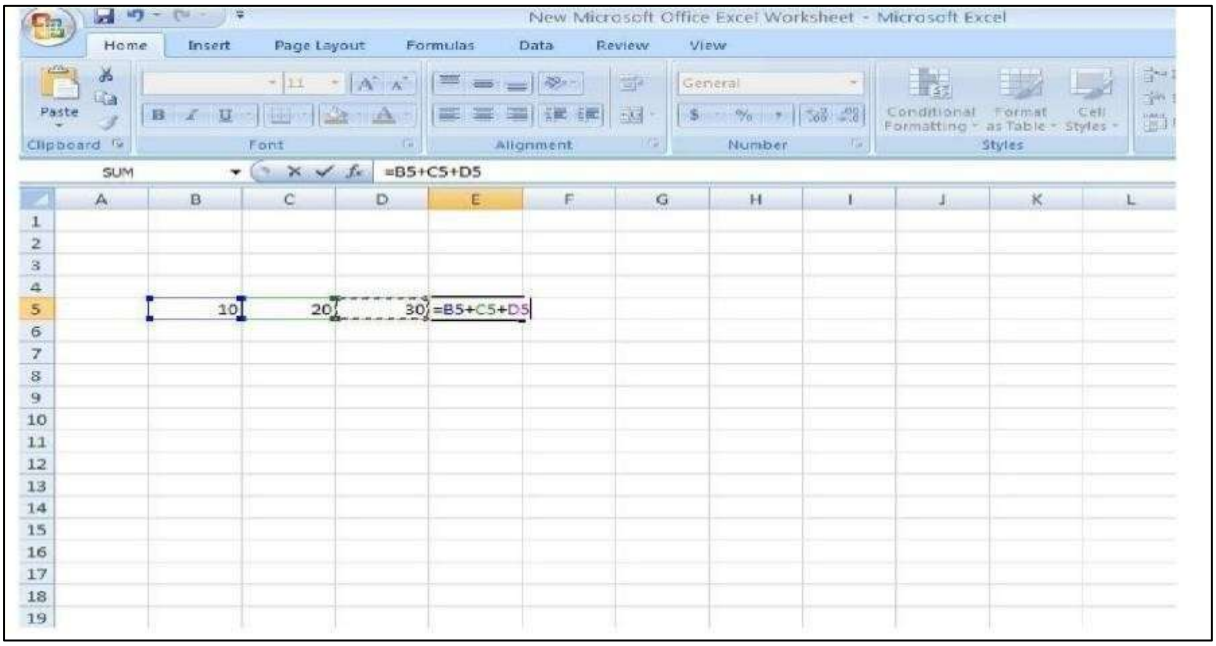
A formula performs calculations or other actions on the data in your worksheet. A formula always starts with an equal sign (=), which can be followed by numbers, math operators (like a + or - sign for addition or subtraction), and built-in Excel functions, which can really expand the power of a Formula.

For Example, in the above worksheet, the formula = B5+C5+D+ adds the contents 10+20+30 and produce the results. One can enter and edit formula in two ways.

1. Directly into cell by double clicking where the formula wants.
2. At formula bar after selection of required cell.

To edit an existing formula

- Click on the cell which contains the formula or results.
- Click in formula bar make necessary changes.
- Press enter key or click on check mark.



SUM =B5+C5+D5

	A	B	C	D	E
5		10	20	30	=B5+C5+D5

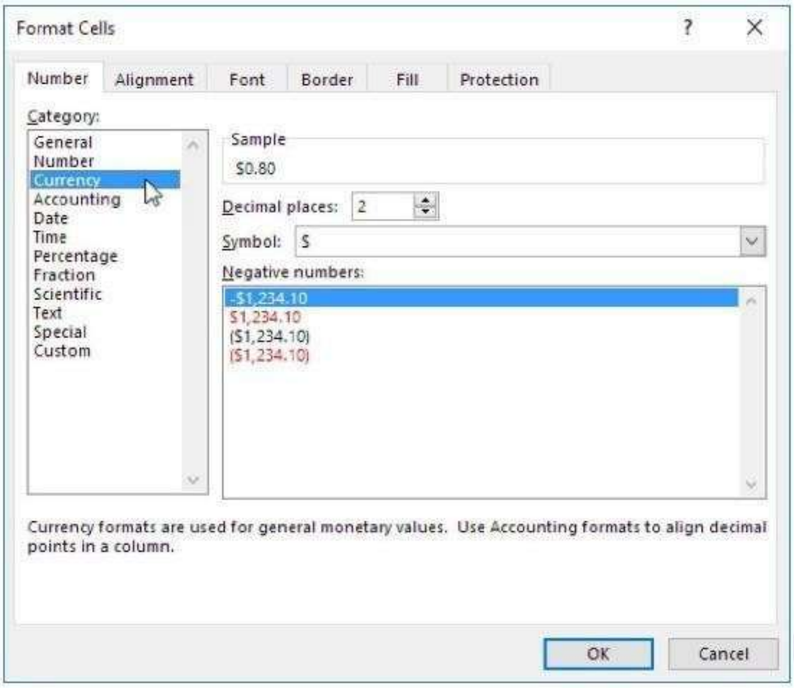
Number Formatting in excel

It is very common to enter various types of numbers for various applications. In Excel, you can use number formats to change the appearance of numbers, including dates and times, without changing the number behind the appearance. The number format does not affect the actual cell value, it changes the appearance only.

1. Select the cell or cells which contain numbers.
2. On the home tab, under Number group click on down arrow mark.
3. Right click your mouse; from the short hand menu select format cell option
4. It launches Formula cells window. Click on Number tab.

Auto fill and custom fill

Autofill is one of the features presents in the ms excel. When you’re typing a day, month, year and number the automatic series will be appeared by dragging it. This feature is called Autofill. For Example, if your typed “Jan” and then dragged then it displays months form” Jan to dec” like.



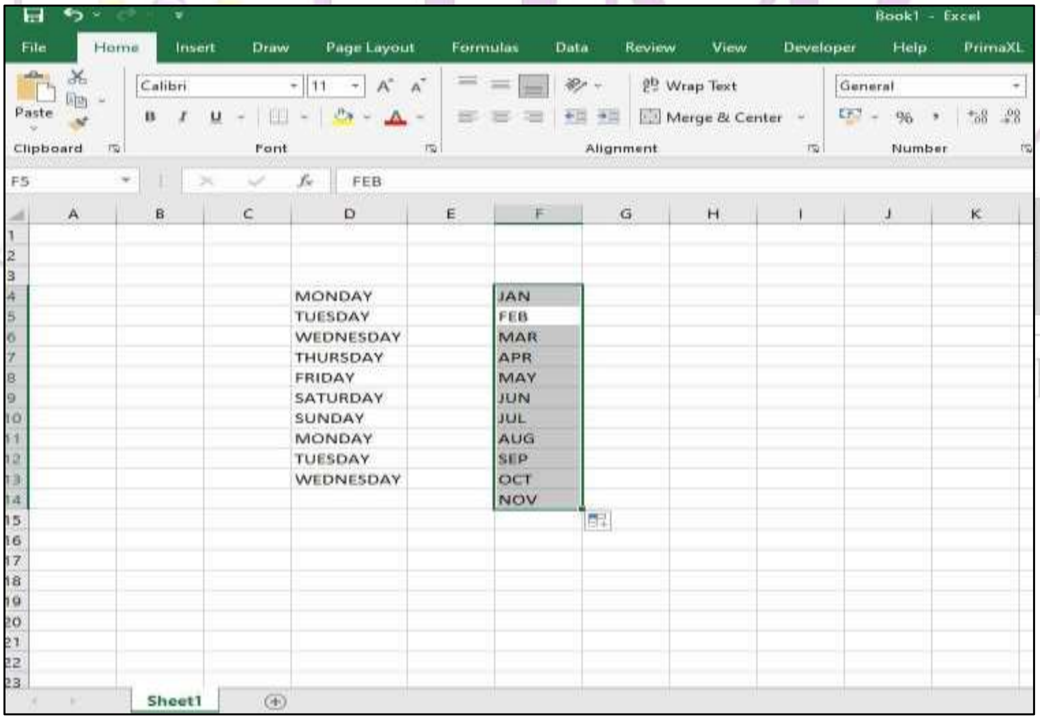
Sample
$0.80

Decimal places: 2

Symbol: $

Negative numbers:
-$1,234.10
$1,234.10
($1,234.10)
($1,234.10)

Currency formats are used for general monetary values. Use Accounting formats to align decimal points in a column.



F5 | FEB

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3											
4				MONDAY		JAN					
5				TUESDAY		FEB					
6				WEDNESDAY		MAR					
7				THURSDAY		APR					
8				FRIDAY		MAY					
9				SATURDAY		JUN					
10				SUNDAY		JUL					
11				MONDAY		AUG					
12				TUESDAY		SEP					
13				WEDNESDAY		OCT					
14						NOV					

EX NO: 2a	Numerical Operations (MAX, MIN, AVG, SUM, SQRT, ROUND)
Date:	

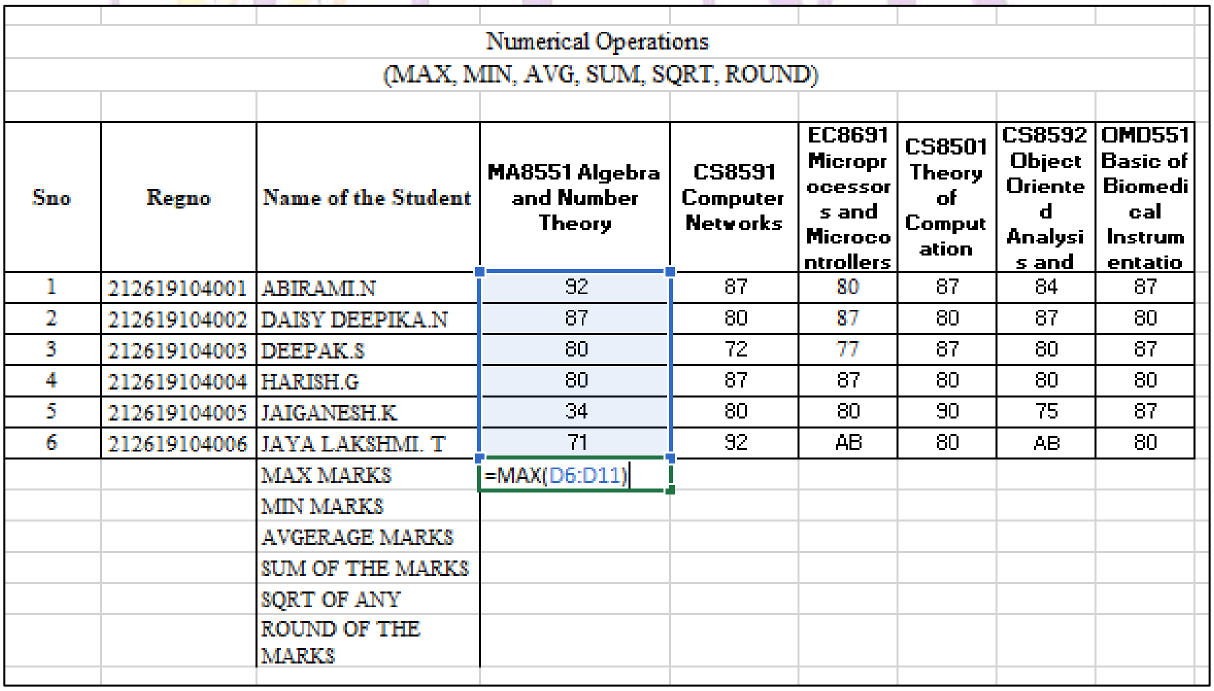
AIM:

To implement numerical operations using MS-EXCEL.

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office.
Step 2: Create datasheet for student marks in Ms Excel application.
Step 3: Calculate the Maximum of the given marks using max function.
Step 4: Calculate the Minimum of the given marks using MIN function.
Step 5: Calculate the average of the given marks using average function.
Step 6: Calculate the sum of the given marks using sum function.
Step 7: Calculate the square root of the given mark using SQRT function.
Step 8: Calculate the Round of the given mark using Roundup function.
Step 9: Display the desired output of all numerical operation in neat format.
Step 10: Save the excel file and Close the Ms Excel application

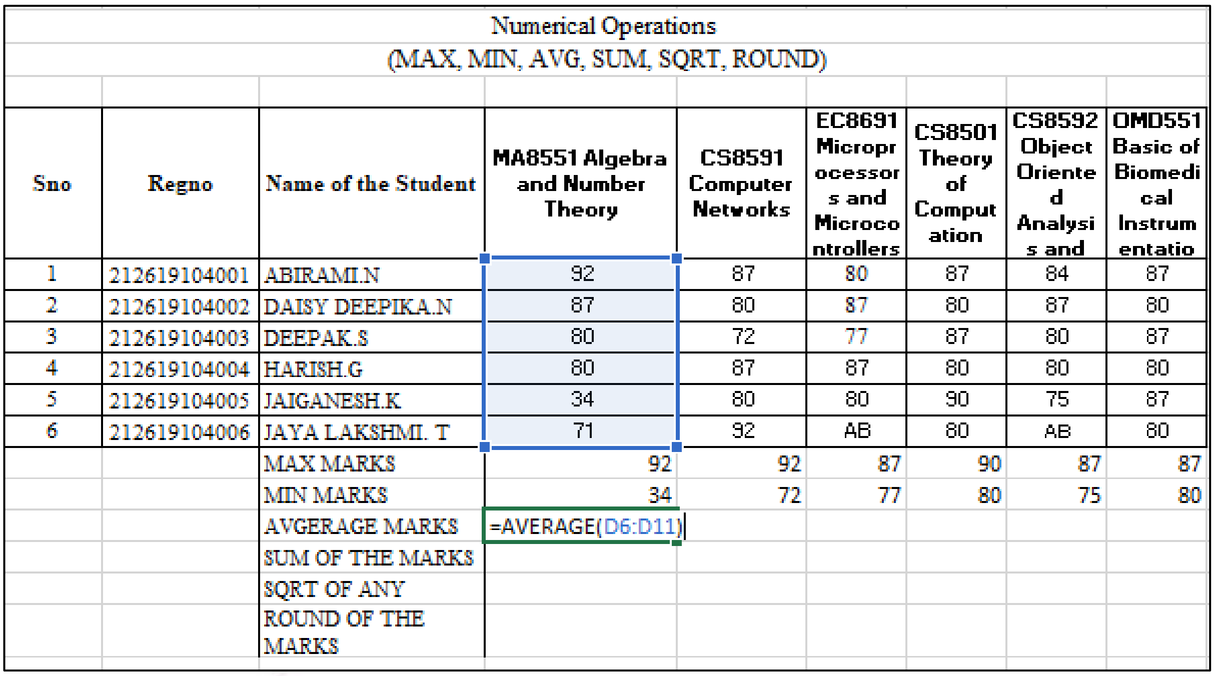
Finding Maximum of the given marks



Numerical Operations
(MAX, MIN, AVG, SUM, SQRT, ROUND)

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Micropr ocessor s and Microco ntrollers	CS8501 Theory of Comput ation	CS8592 Object Oriente d Analysi s and	OMD551 Basic of Biomedi cal Instrum entatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	AB	80	AB	80
		MAX MARKS	=MAX(D6:D11)					
		MIN MARKS						
		AVGERAGE MARKS						
		SUM OF THE MARKS						
		SQRT OF ANY						
		ROUND OF THE MARKS						

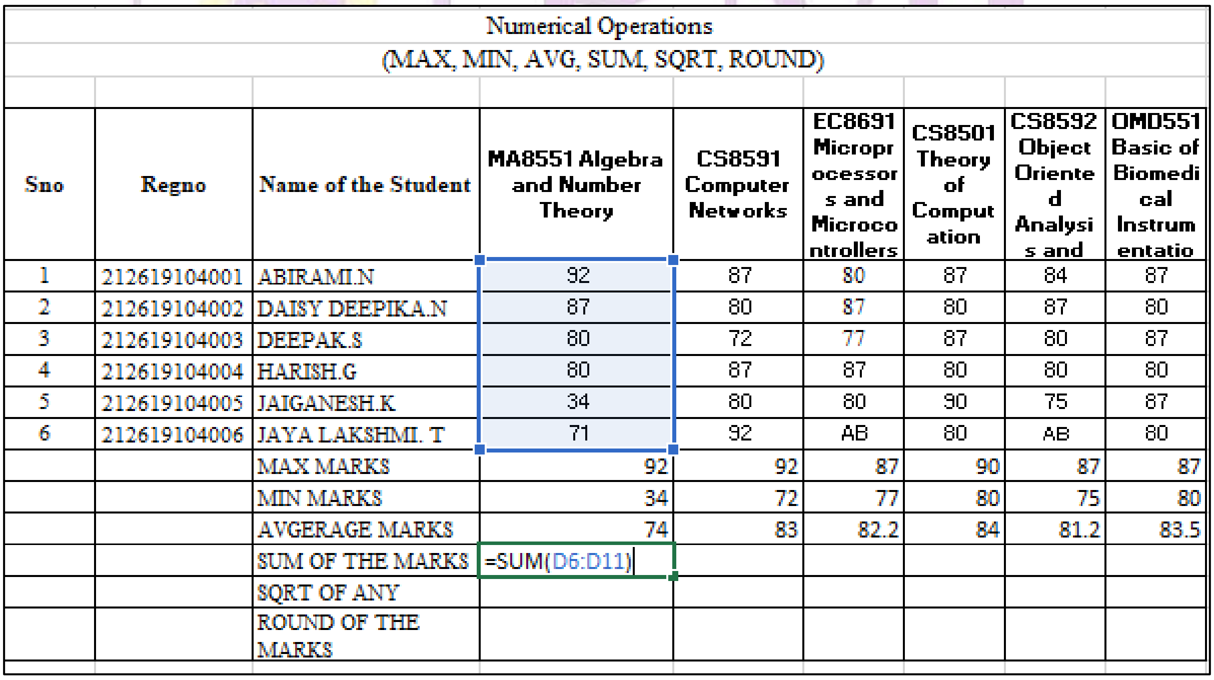
Finding average of the given marks



Numerical Operations
(MAX, MIN, AVG, SUM, SQRT, ROUND)

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	AB	80	AB	80
		MAX MARKS	92	92	87	90	87	87
		MIN MARKS	34	72	77	80	75	80
		AVGERAGE MARKS	=AVERAGE(D6:D11)					
		SUM OF THE MARKS						
		SQRT OF ANY						
		ROUND OF THE MARKS						

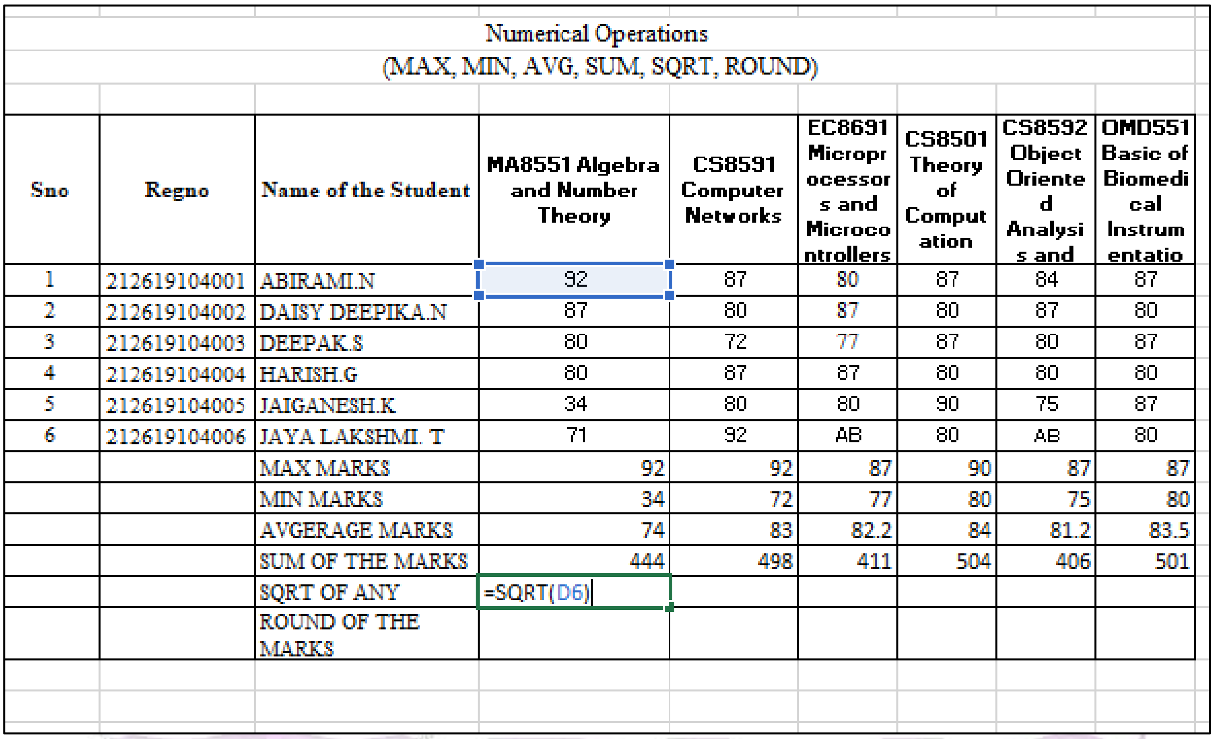
Finding sum of the given marks



Numerical Operations
(MAX, MIN, AVG, SUM, SQRT, ROUND)

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	AB	80	AB	80
		MAX MARKS	92	92	87	90	87	87
		MIN MARKS	34	72	77	80	75	80
		AVGERAGE MARKS	74	83	82.2	84	81.2	83.5
		SUM OF THE MARKS	=SUM(D6:D11)					
		SQRT OF ANY						
		ROUND OF THE MARKS						

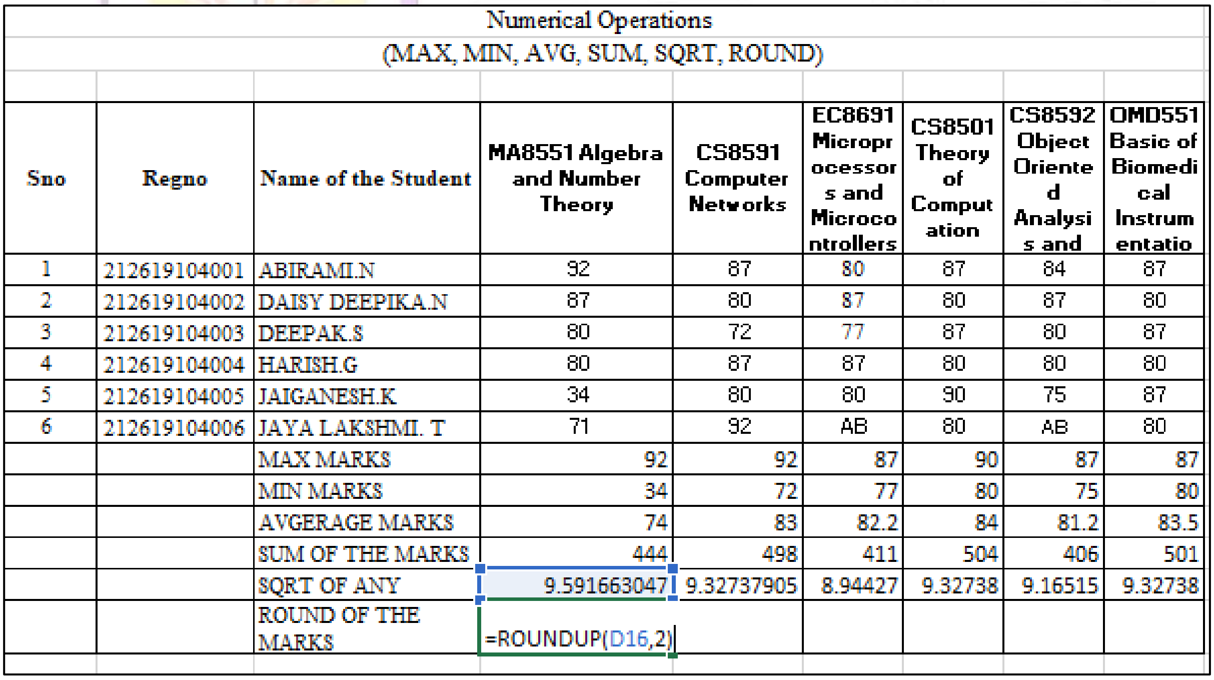
Finding SQRT of the given marks



Numerical Operations
(MAX, MIN, AVG, SUM, SQRT, ROUND)

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	AB	80	AB	80
		MAX MARKS	92	92	87	90	87	87
		MIN MARKS	34	72	77	80	75	80
		AVGERAGE MARKS	74	83	82.2	84	81.2	83.5
		SUM OF THE MARKS	444	498	411	504	406	501
		SQRT OF ANY	=SQRT(D6)					
		ROUND OF THE MARKS						

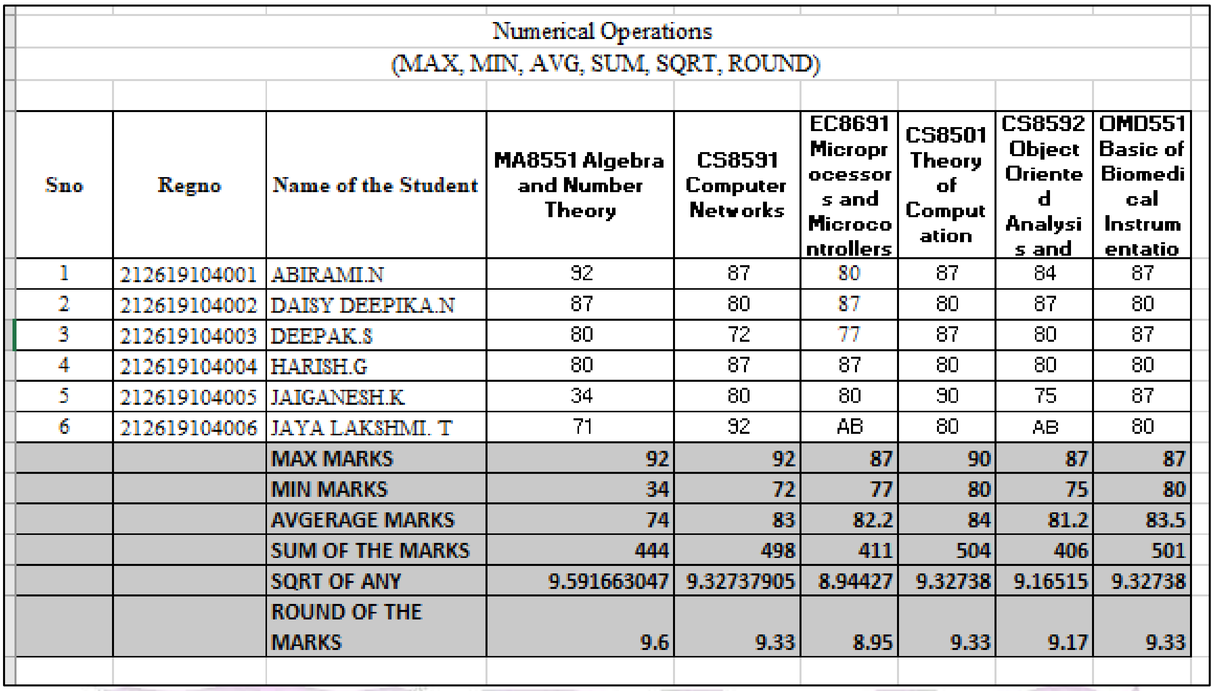
Finding Round of the given marks



Numerical Operations
(MAX, MIN, AVG, SUM, SQRT, ROUND)

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	AB	80	AB	80
		MAX MARKS	92	92	87	90	87	87
		MIN MARKS	34	72	77	80	75	80
		AVGERAGE MARKS	74	83	82.2	84	81.2	83.5
		SUM OF THE MARKS	444	498	411	504	406	501
		SQRT OF ANY	9.591663047	9.32737905	8.94427	9.32738	9.16515	9.32738
		ROUND OF THE MARKS	=ROUNDUP(D16,2)					

OUTPUT:



Numerical Operations
(MAX, MIN, AVG, SUM, SQRT, ROUND)

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	AB	80	AB	80
		MAX MARKS	92	92	87	90	87	87
		MIN MARKS	34	72	77	80	75	80
		AVGERAGE MARKS	74	83	82.2	84	81.2	83.5
		SUM OF THE MARKS	444	498	411	504	406	501
		SQRT OF ANY	9.591663047	9.32737905	8.94427	9.32738	9.16515	9.32738
		ROUND OF THE MARKS	9.6	9.33	8.95	9.33	9.17	9.33

EX NO: 2b	**Perform data import/export operations for different file formats**
Date:	

AIM:

To perform data import/export operations for different file formats using MS-EXCEL.

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office.
Step 2: Create datasheet for student marks in Ms Excel application.

Step 3: Save the excel file.

Step 4: Export the file into CSV file using file menu and export option.

Step 5: Next, import CSV file using data menu and get data option.

Step 6: Display the desired output in neat format.

Step 7: Save the excel file and Close the Ms Excel application.

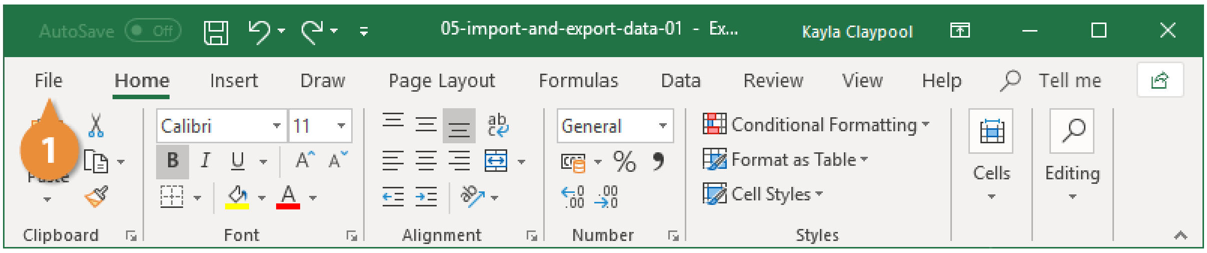
Data Import/Export Operations for Different File Formats

Excel can import and export many different file types aside from the standard .xslx format. If your data is shared between other programs, like a database, you may need to save data as a different file type or bring in files of a different file type.

EXPORT DATA

When you have data that needs to be transferred to another system, export it from Excel in a format that can be interpreted by other programs, such as a text or CSV file.

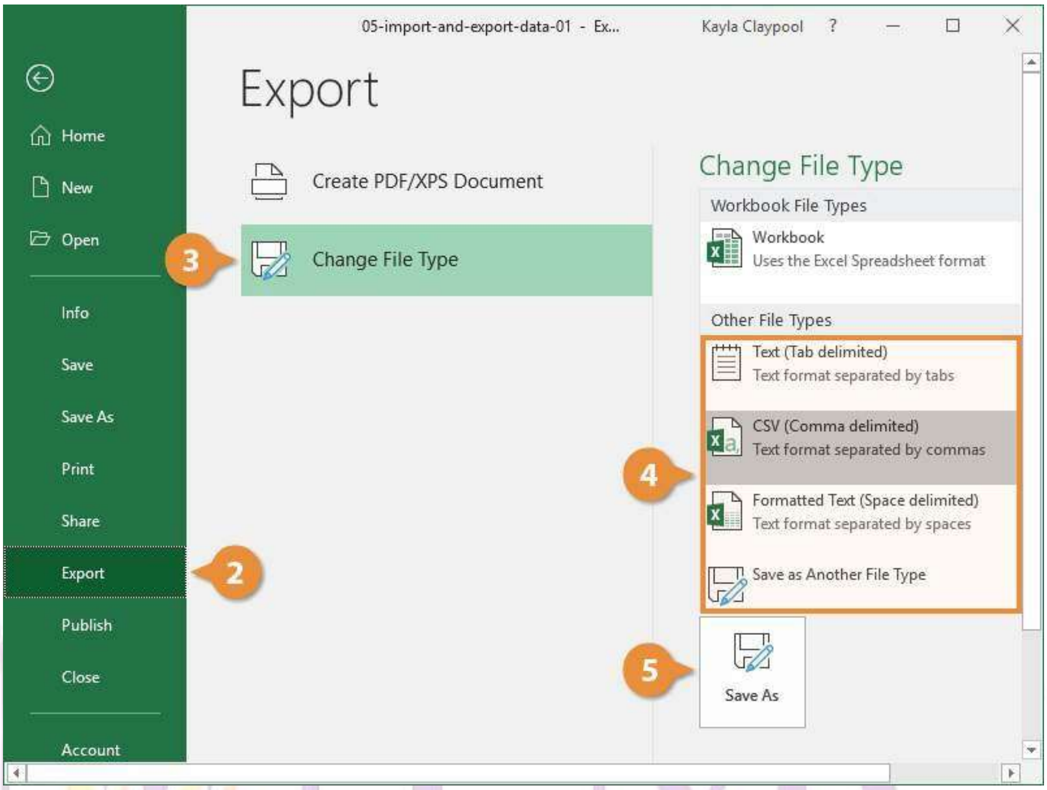
Click the **File** tab.



1. At the left, click **Export**.
2. Click the **Change File Type**.
3. Under Other File Types, select a file type.
   - **Text (Tab delimited):** The cell data will be separated by a tab.
   - **CSV (Comma delimited):** The cell data will be separated by a comma.
   - **Formatted Text (space delimited):** The cell data will be separated by a space.
   - **Save as Another File Type:** Select a different file type when the Save As dialog box appears.

The file type you select will depend on what type of file is required by the program that will consume the exported data.

4. Click **Save As**.

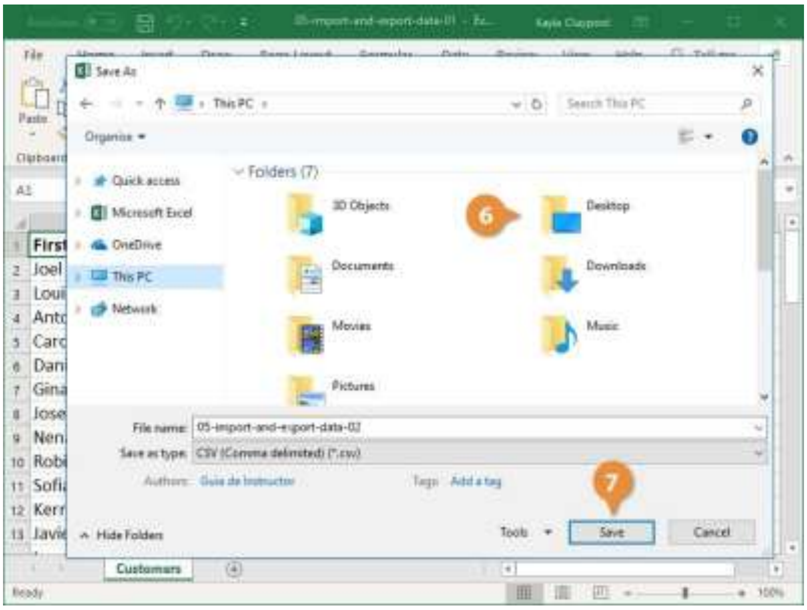


5. Specify where you want to save the file.
6. Click **Save**.

A dialog box appears stating that some of the workbook features may be lost.

7. Click **Yes**.

OUTPUT FOR EXPORTING THE FILE


File name: 05-import-and-export-data-02

Save as type: CSV (Comma delimited) (*.csv)

Authors: Guia de Instructor

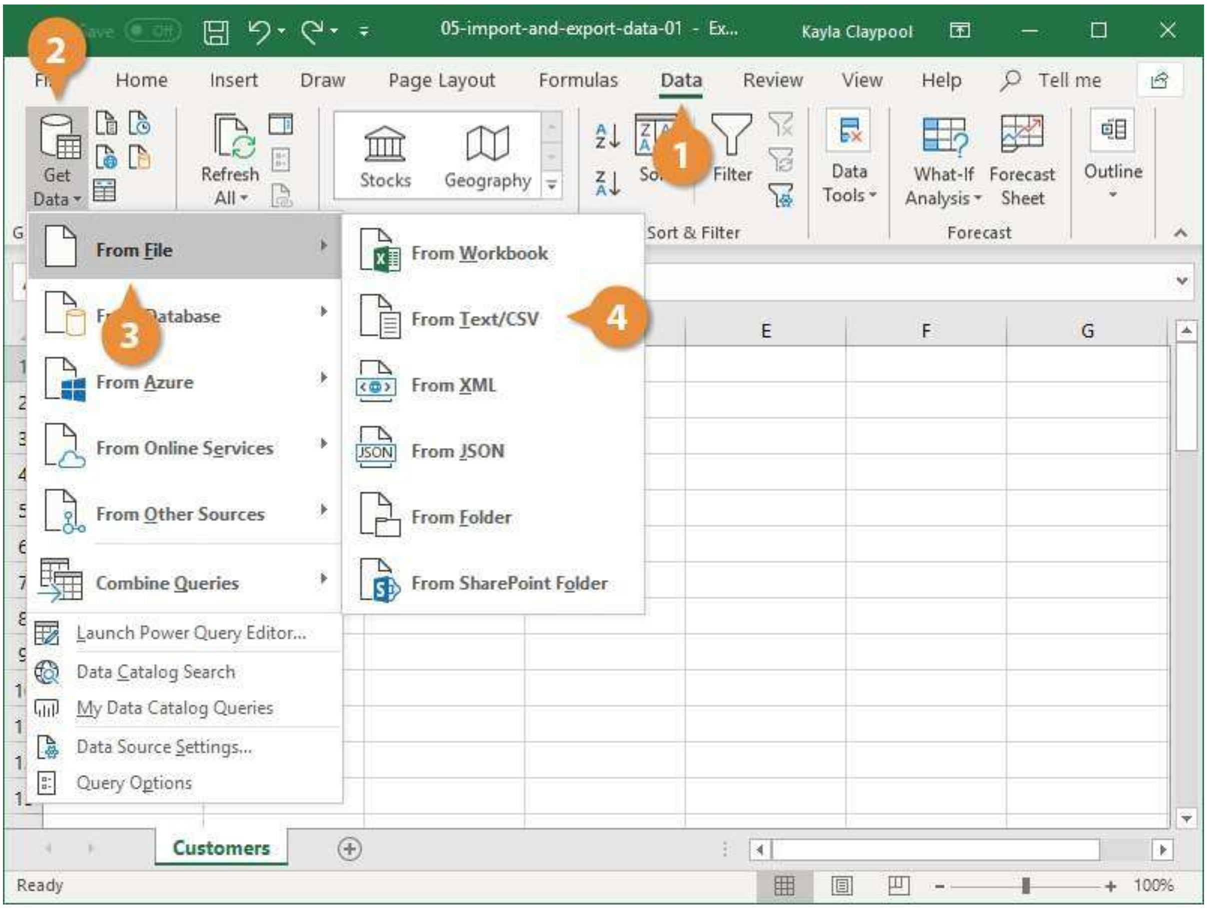
Import Data

Excel can import data from external data sources including other files, databases, or web pages.

1. Click the **Data** tab on the Ribbon.
2. Click the **Get Data** button.

   Some data sources may require special security access, and the connection process can often be very complex. Enlist the help of your organization's technical support staff for assistance.

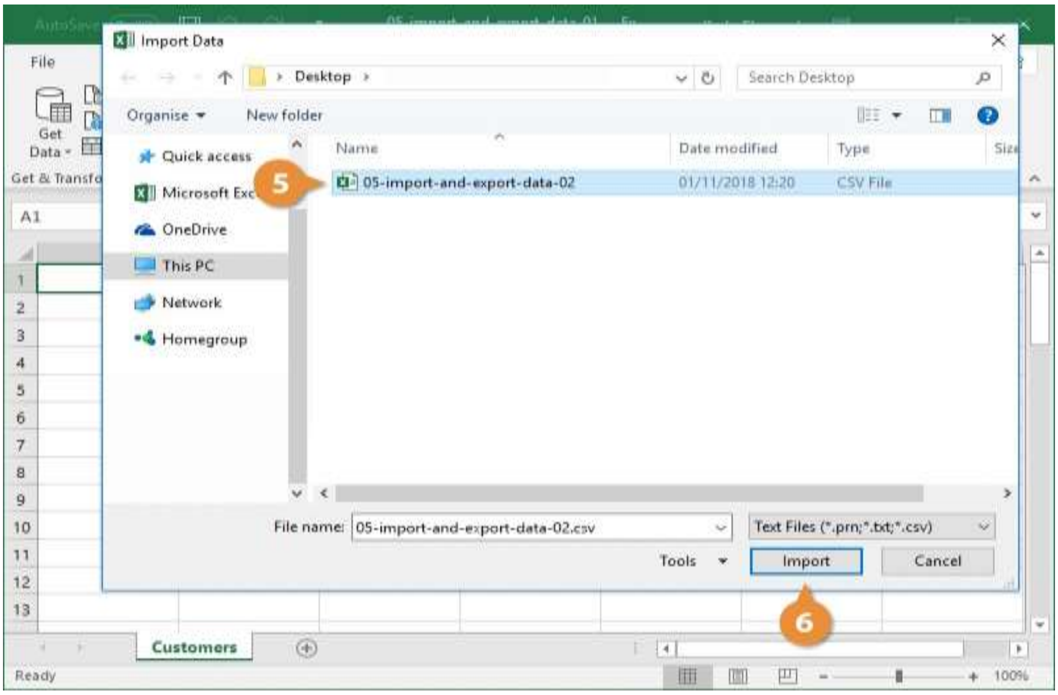
3. Select **From File**.
4. Select **From Text/CSV**.



If you have data to import from Access, the web, or another source, select one of those options in the Get External Data group instead.

5. Select the file you want to import.
6. Click **Import**.

   If, while importing external data, a security notice appears saying that it is connecting to an external source that may not be safe, click **OK**.



Name	Date modified	Type
05-import-and-export-data-02	01/11/2018 12:20	CSV File

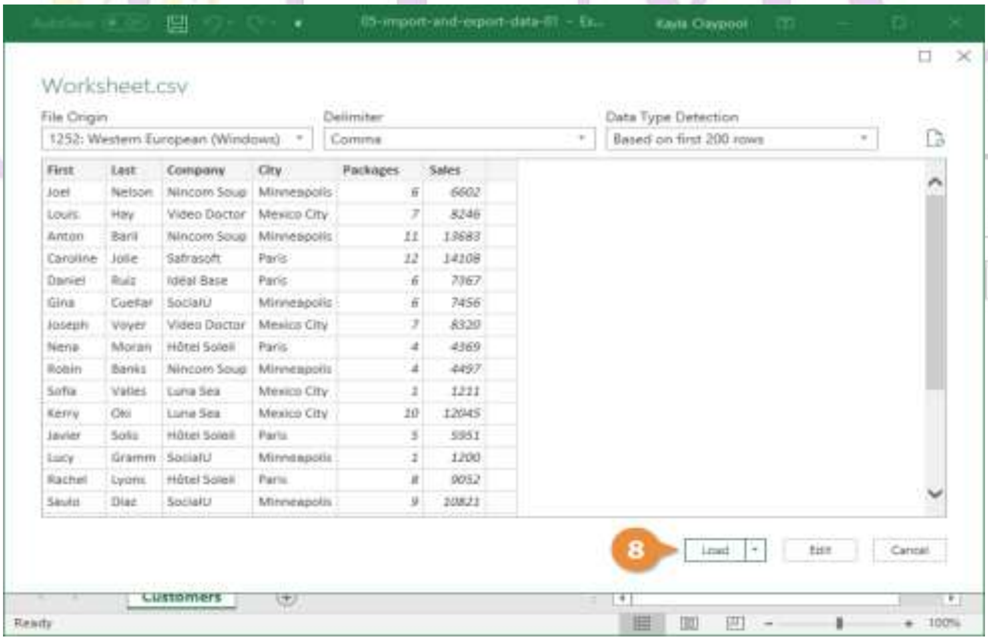
File name: 05-import-and-export-data-02.csv

Text Files (*.prn;*.txt;*.csv)

Tools ▾ Import Cancel

7. Verify the preview looks correct.
   Because we've specified the data is separated by commas, the delimiter is already set. If you need to change it, it can be done from this menu.
8. Click **Load**.

OUTPUT FOR IMPORT THE FILE:



Worksheet.csv

File Origin: 1252: Western European (Windows)

Delimiter: Comma

Data Type Detection: Based on first 200 rows

First	Last	Company	City	Packages	Sales
Joel	Nelson	Nincom Soup	Minneapolis	6	6602
Louis	Hay	Video Doctor	Mexico City	7	8246
Anton	Baril	Nincom Soup	Minneapolis	11	13683
Caroline	Jolie	Safrasoft	Paris	12	14108
Daniel	Ruiz	Idéal Base	Paris	6	7367
Gina	Cuellar	SocialU	Minneapolis	6	7456
Joseph	Voyer	Video Doctor	Mexico City	7	8320
Nena	Moran	Hôtel Soleil	Paris	4	4369
Robin	Banks	Nincom Soup	Minneapolis	4	4497
Sofia	Valles	Luna Sea	Mexico City	1	1211
Kerry	Oki	Luna Sea	Mexico City	10	12045
Javier	Solis	Hôtel Soleil	Paris	5	5951
Lucy	Gramm	SocialU	Minneapolis	1	1200
Rachel	Lyons	Hôtel Soleil	Paris	8	9052
Saulo	Diaz	SocialU	Minneapolis	9	10821

EX NO: 3	PERFORM STATISTICAL OPERATIONS [Mean, Median, Mode and Standard Deviation, Variance, Skewness, Kurtosis]
Date:	

AIM:

To Perform statistical operations using MS-EXCEL.

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office.

Step 2: Create datasheet for student marks in Ms Excel application.

Step 3: If you haven't already installed the Analysis ToolPak, Click the Microsoft Office button, then click on the Excel Options, and then select Add-Ins, Click Go, check the Analysis ToolPak box, and click Ok

Step 4: Select Data tab, then click on the Data Analysis option, then selects Descriptive Statistics from the list and Click Ok. [Data tab >> Data Analysis >> Descriptive Statistics]

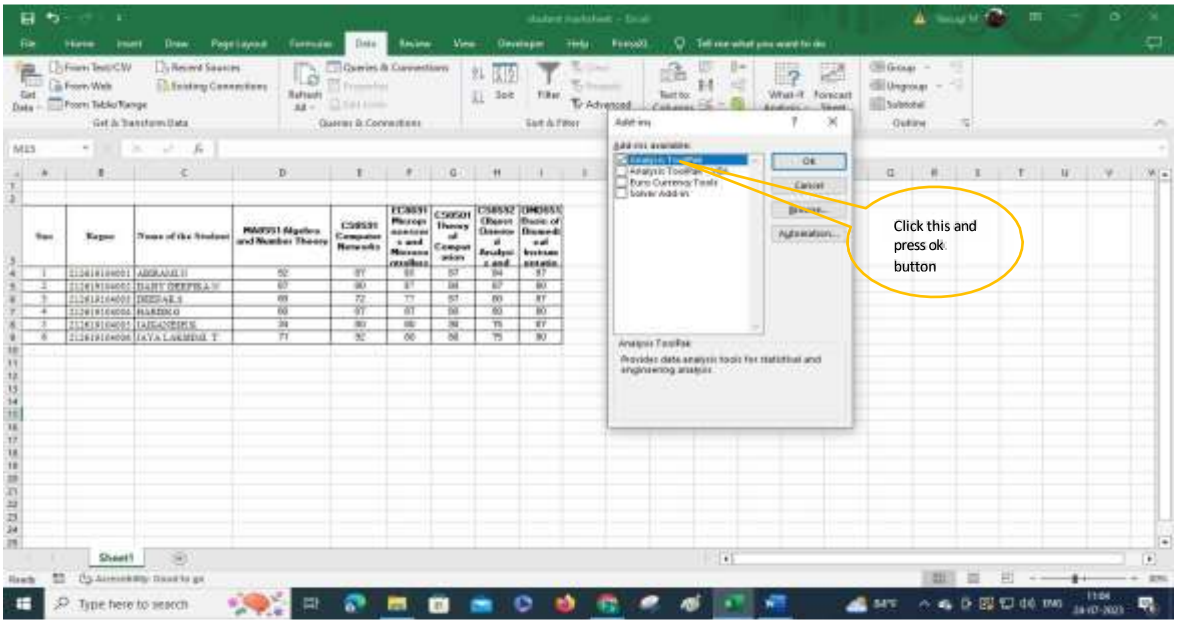
Step 5: In the Input Range we select the data, and then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 6: Check Summary Statistics and Confidence Level for Mean options. By default, the confidence level is 95%. You can change the level as per the hypothesis standard of study.

Step 7: When you click Ok, you will see the result in the selected output range.

Step 8: Save the excel file and Close the Ms Excel application.

PERFORM STATISTICAL OPERATIONS



Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basics of Biomedical Instrumentation
1	212619104001	ABIRAMI.H	52	87	88	87	84	87
2	212619104002	DAISY DEEPIKA.V	67	80	87	88	67	80
3	212619104003	DEEPA.K.S	69	72	77	87	80	87
4	212619104004	HARINI.G	60	87	87	88	80	80
5	212619104005	JAISANTHINI.K	39	80	88	39	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	60	88	75	90

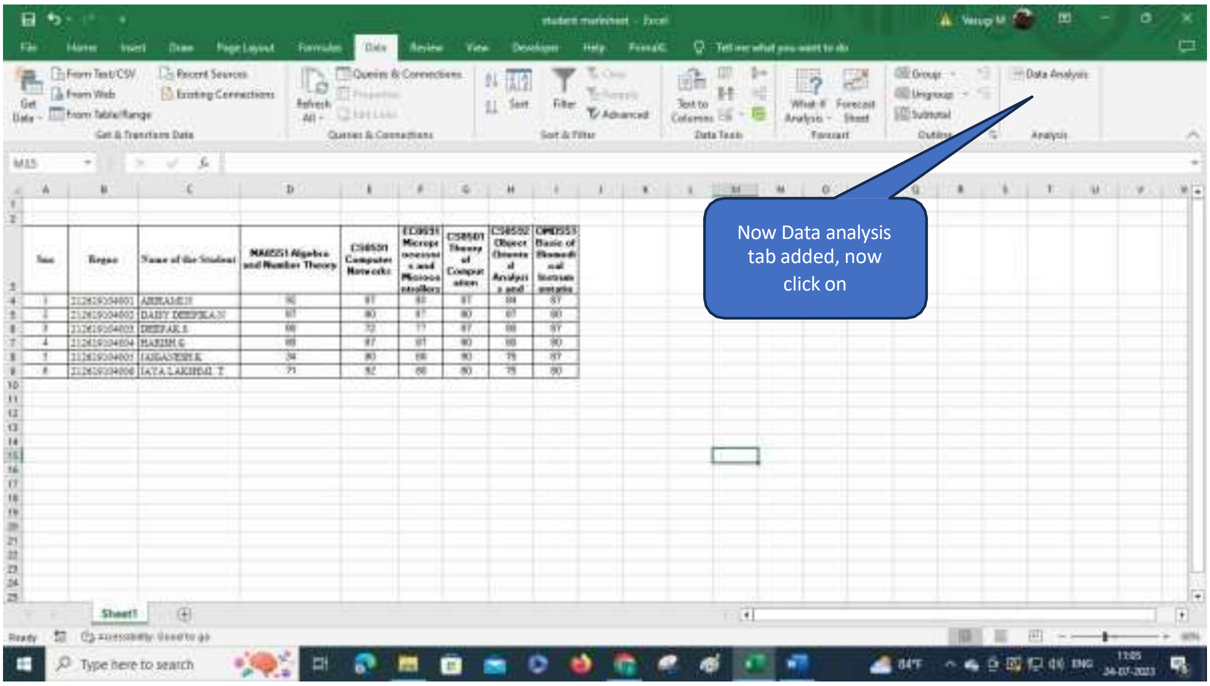
Add-ins

Add-ins available:

☑ Analysis ToolPak
□ Analysis ToolPak - VBA
□ Euro Currency Tools
□ Solver Add-in

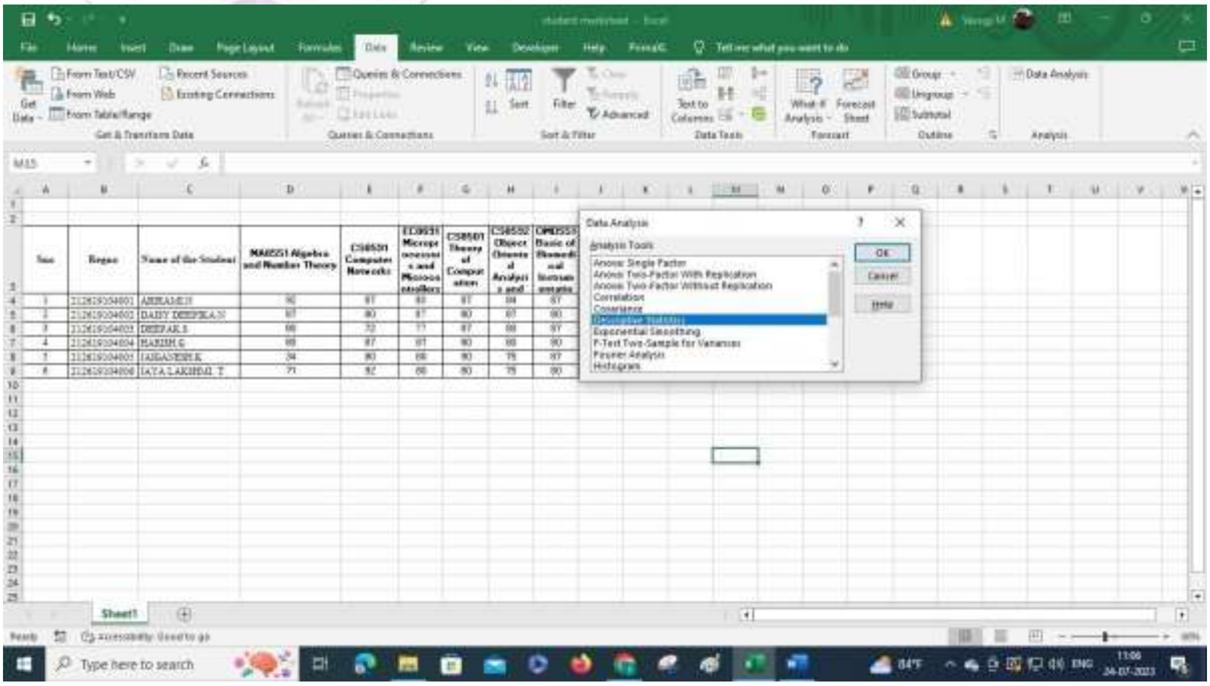
Analysis ToolPak

Provides data analysis tools for statistical and engineering analysis



Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD553 Basic of Biomedical Instrumentation
1	212619104001	ABIRAMI.S	92	87	83	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	88	72	77	87	88	87
4	212619104004	HARISH.G	88	87	87	80	88	80
5	212619104005	JAIGANESH.K	34	80	88	80	78	87
6	212619104006	JAYALAKSHMI.T	71	82	88	80	78	80

Now Data analysis tab added, now click on

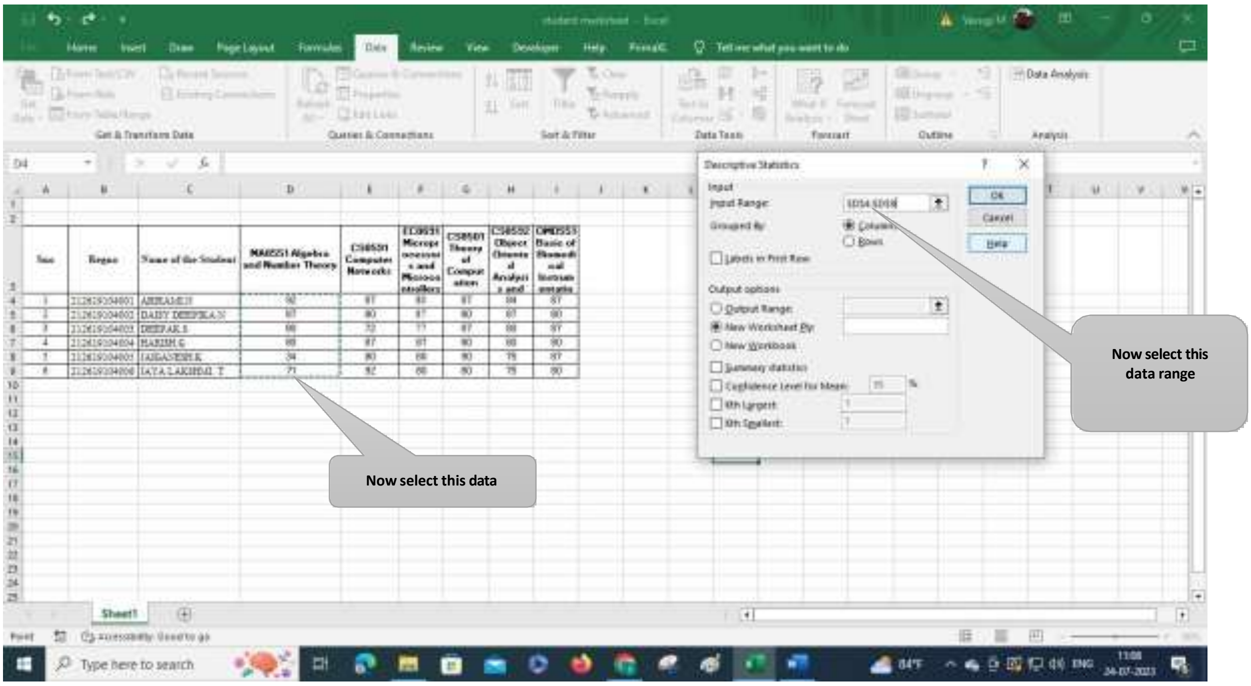


Data Analysis

Analysis Tools

- Anova: Single Factor
- Anova: Two-Factor With Replication
- Anova: Two-Factor Without Replication
- Correlation
- Covariance
- Descriptive Statistics
- Exponential Smoothing
- F-Test Two-Sample for Variances
- Fourier Analysis
- Histogram

Now select this option and press ok



D4

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD553 Basic of Biomedical Instrumentation
1	212619104001	ABIRAMI.S	92	87	83	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	88	72	77	87	88	87
4	212619104004	HARISH.G	88	87	87	90	88	90
5	212619104005	JAIGANESH.K	34	80	88	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	88	80	75	80

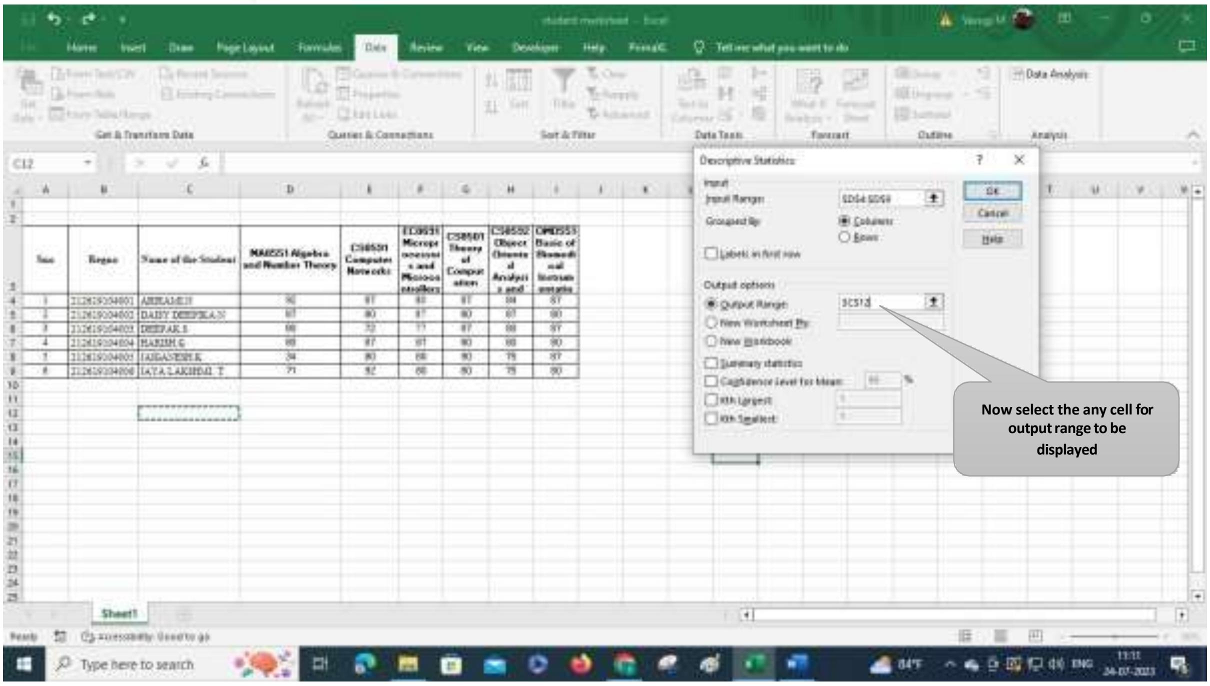
Descriptive Statistics

Input
Input Range: $D$4:$D$9
Grouped By: ● Columns ○ Rows
□ Labels in First Row

Output options
○ Output Range:
● New Worksheet Ply:
○ New Workbook
□ Summary statistics
□ Confidence Level for Mean: 95 %
□ Kth Largest: 1
□ Kth Smallest: 1

Now select this data

Now select this data range



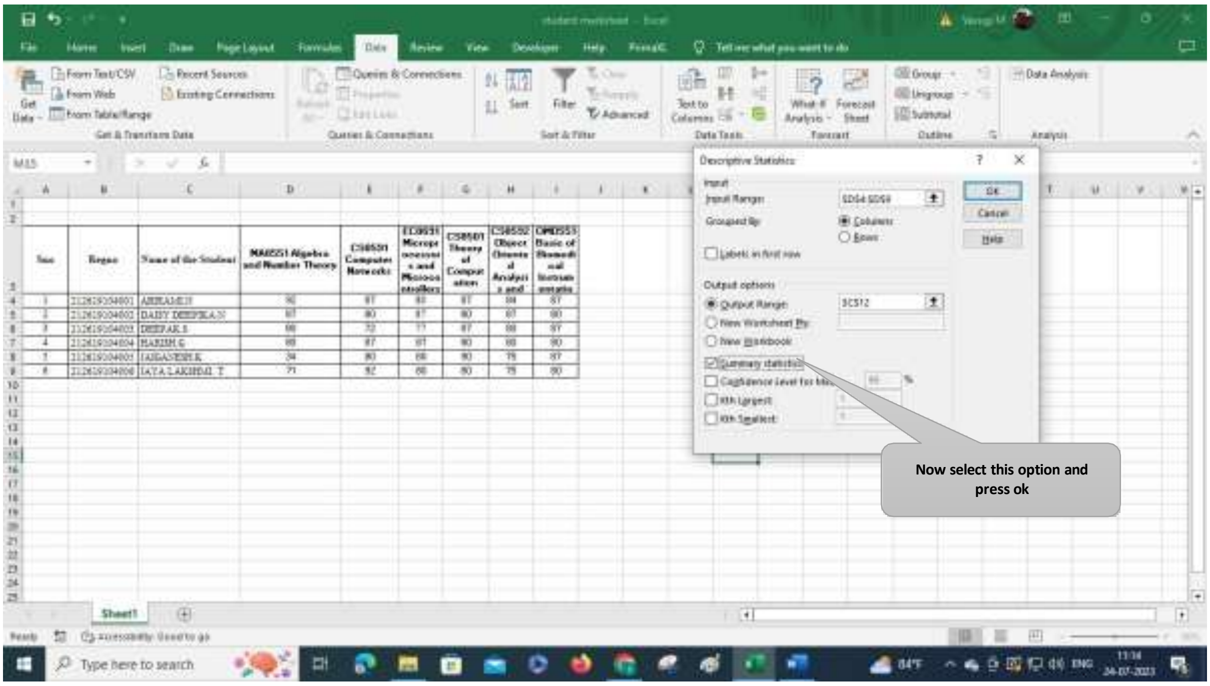
C12

Descriptive Statistics

Input
Input Range: $D$4:$D$9
Grouped By: ● Columns ○ Rows
□ Labels in first row

Output options
● Output Range: $C$12
○ New Worksheet Ply:
○ New Workbook
□ Summary statistics
□ Confidence Level for Mean: 95 %
□ Kth Largest: 1
□ Kth Smallest: 1

Now select the any cell for output range to be displayed



Descriptive Statistics

Input

Input Range: $D$4:$D$9

Grouped By: ● Columns ○ Rows

□ Labels in first row

Output options

● Output Range: $C$12

○ New Worksheet Ply:

○ New Workbook

☑ Summary statistics

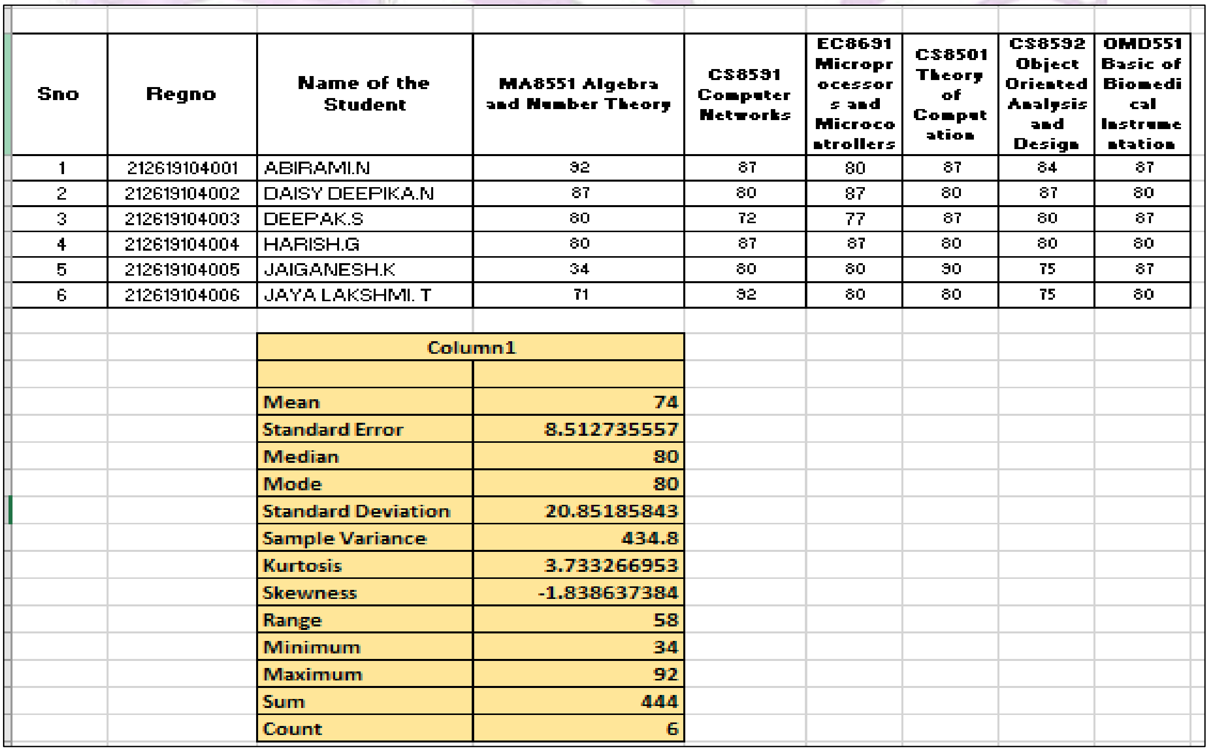
□ Confidence Level for Mean: 95 %

□ Kth Largest: 1

□ Kth Smallest: 1

Now select this option and press ok

OUTPUT:



Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and Design	OMD551 Basic of Biomedical Instrumentation
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	80	80	75	80

Column1	
Mean	74
Standard Error	8.512735557
Median	80
Mode	80
Standard Deviation	20.85185843
Sample Variance	434.8
Kurtosis	3.733266953
Skewness	-1.838637384
Range	58
Minimum	34
Maximum	92
Sum	444
Count	6

EX NO: 4a	Perform Z-test
Date:	

AIM:

To Perform Z-test operations using MS-EXCEL.

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office.

Step 2: Create datasheet for student marks in Ms Excel application.

Step 3: If you haven't already installed the Analysis ToolPak , Click the Microsoft Office button, then click on the Excel Options , and then select Add-Ins , Click Go, check the Analysis ToolPak box, and click Ok

Step 4: Select Data tab, then click on the Data Analysis option, then selects Descriptive Statistics from the list and Click Ok. [Data tab >> Data Analysis >>z-test two sample means]

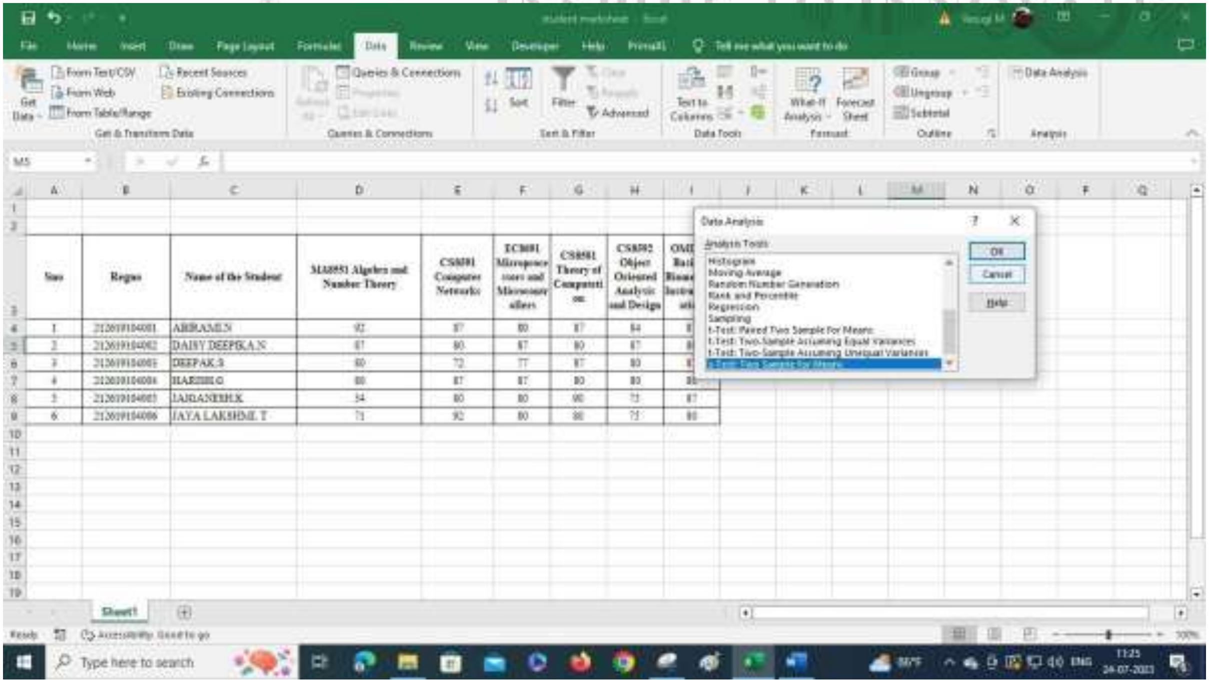
Step 5: In the Input Range we select range of the data for variable 1 and variable 2 and Give variable 1 and variable 2 value as 0.5. then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 6: Then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 7: When you click Ok, you will see the result in the selected output range.

Step 8: Save the excel file and Close the Ms Excel application.

Perform Z-test



S.no	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and Design	OMI Basi Biome Instru ati
1	212619104001	ABIRAMI.N	92	87	80	87	84	8
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	8
3	212619104003	DEEPAK.S	80	72	77	87	80	8
4	212619104004	HARSHINI.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	54	80	80	90	73	87
6	212619104006	JAYA LAKSHMI.T	73	92	80	80	73	80

Data Analysis

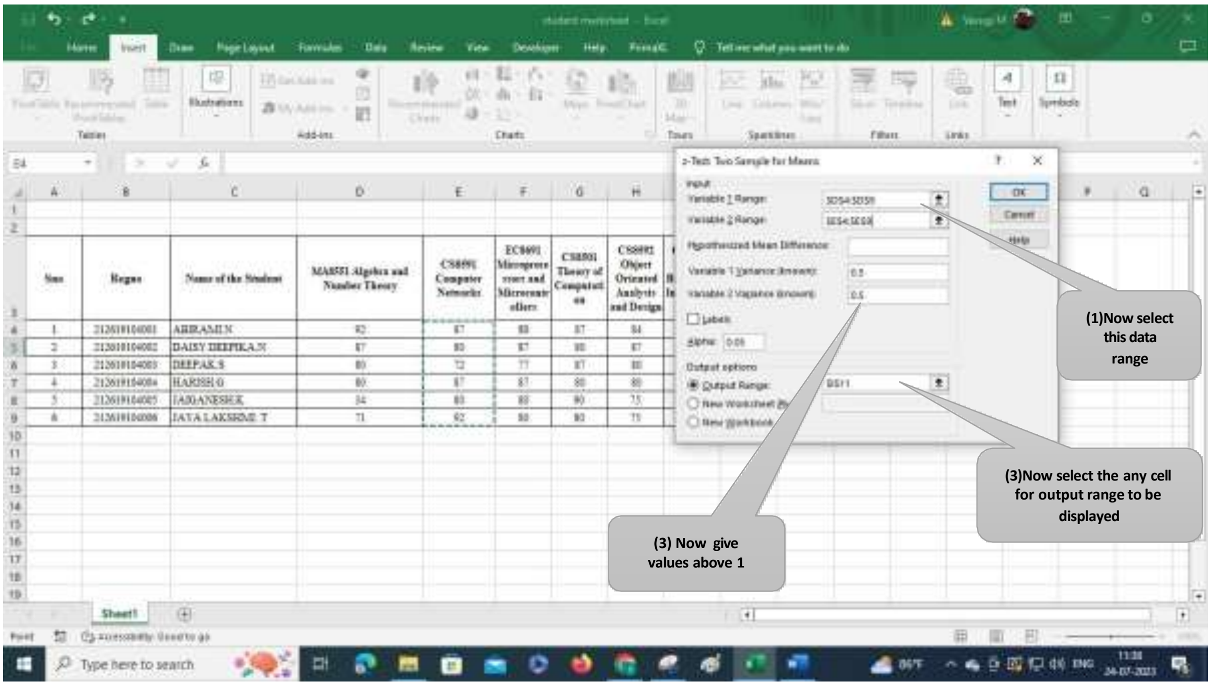
Analysis Tools

- Histogram
- Moving Average
- Random Number Generation
- Rank and Percentile
- Regression
- Sampling
- t-Test: Paired Two Sample for Means
- t-Test: Two-Sample Assuming Equal Variances
- t-Test: Two-Sample Assuming Unequal Variances
- z-Test: Two Sample for Means

OK

Cancel

Help



z-Test: Two Sample for Means

Input

Variable 1 Range: $D$4:$D$9

Variable 2 Range: $E$4:$E$9

Hypothesized Mean Difference:

Variable 1 Variance (known): 0.5

Variable 2 Variance (known): 0.5

□ Labels

Alpha: 0.05

Output options

● Output Range: $B$11

○ New Worksheet Ply:

○ New Workbook

(1)Now select this data range

(3) Now give values above 1

(3)Now select the any cell for output range to be displayed

OUTPUT:

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	80	80	75	80

z-Test: Two Sample for Means		
	Variable 1	*Variable 2*
Mean	74	83
Known Variance	0.5	0.5
Observations	6	6
Hypothesized Mean Di	0	
z	-22.045408	
P(Z<=z) one-tail	0	
z Critical one-tail	1.64485363	
P(Z<=z) two-tail	0	
z Critical two-tail	1.95996398	

EX NO: 4b	Perform T-test
Date:	

AIM:

To Perform T-test operations using MS-EXCEL.

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office.

Step 2: Create datasheet for student marks in Ms Excel application.

Step 3: If you haven't already installed the Analysis ToolPak , Click the Microsoft Office button, then click on the Excel Options , and then select Add-Ins , Click Go, check the Analysis ToolPak box, and click Ok

Step 4: Select Data tab, then click on the Data Analysis option, then selects Descriptive Statistics from the list and Click Ok. [Data tab >> Data Analysis >> T-test Paired two sample for means]

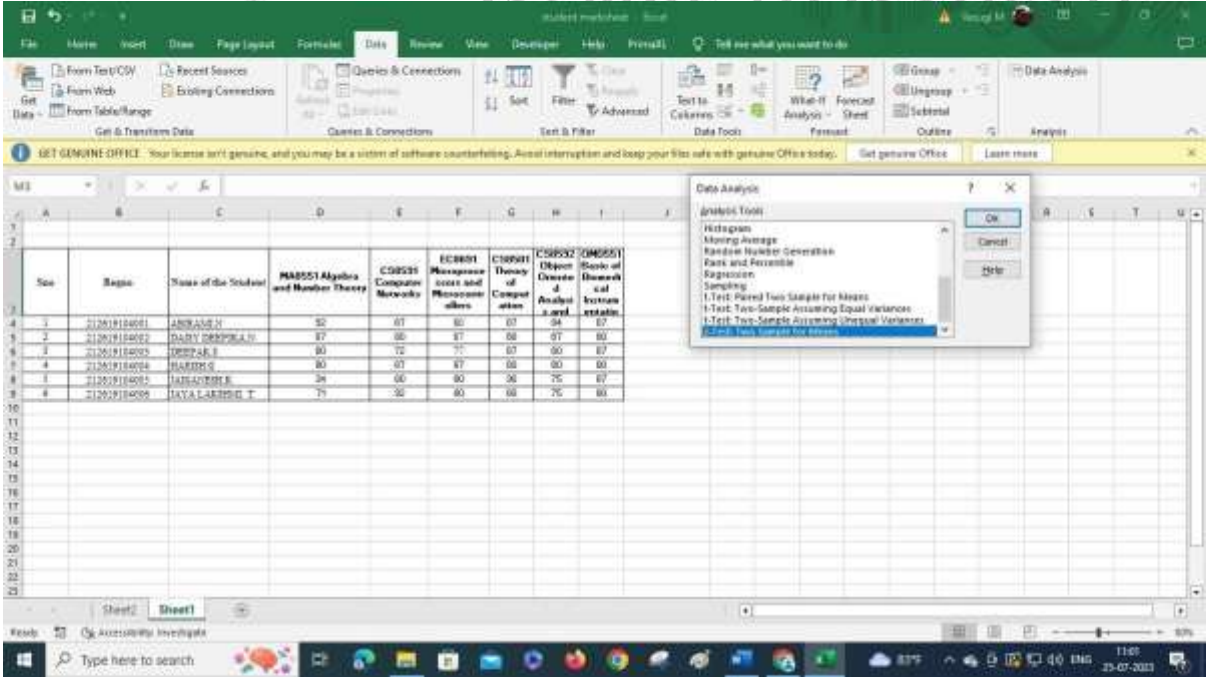
Step 5: In the Input Range we select range of the data for variable 1 and variable 2 and Give alpha value as 0.05. then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 6: Then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 7: When you click Ok, you will see the result in the selected output range.

Step 8: Save the excel file and Close the Ms Excel application.

Perform T-test

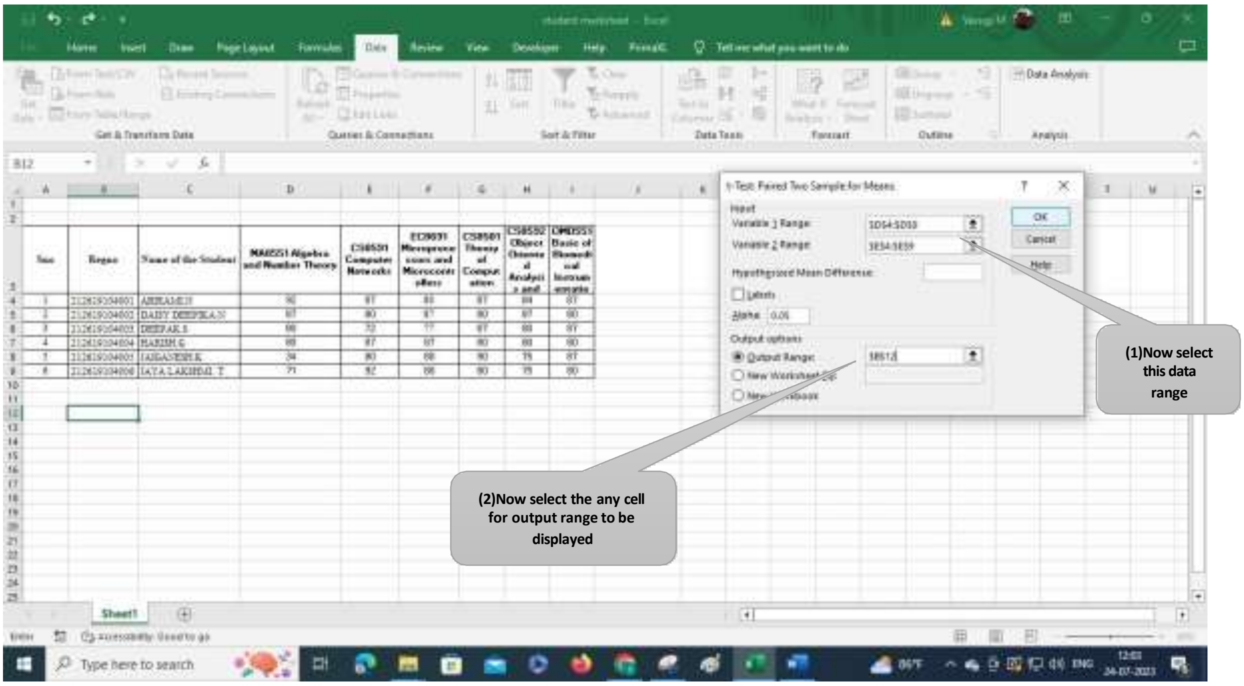


Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI N	92	67	60	67	94	67
2	212619104002	DAISY DEEPIKA N	87	60	87	68	67	60
3	212619104003	DEEPAK I	80	72	77	67	60	67
4	212619104004	HARISH G	80	67	87	68	80	80
5	212619104005	JAISANTHI K	34	60	80	36	75	67
6	212619104006	JAYA LAKSHMI T	71	32	80	68	75	80

Data Analysis

Analysis Tools

- Histogram
- Moving Average
- Random Number Generation
- Rank and Percentile
- Regression
- Sampling
- t-Test: Paired Two Sample for Means
- t-Test: Two-Sample Assuming Equal Variances
- t-Test: Two-Sample Assuming Unequal Variances
- z-Test: Two Sample for Means



t-Test: Paired Two Sample for Means

Input

Variable 1 Range: $D$4:$D$9

Variable 2 Range: $E$4:$E$9

Hypothesized Mean Difference:

□ Labels

Alpha: 0.05

Output options

● Output Range: $B$12

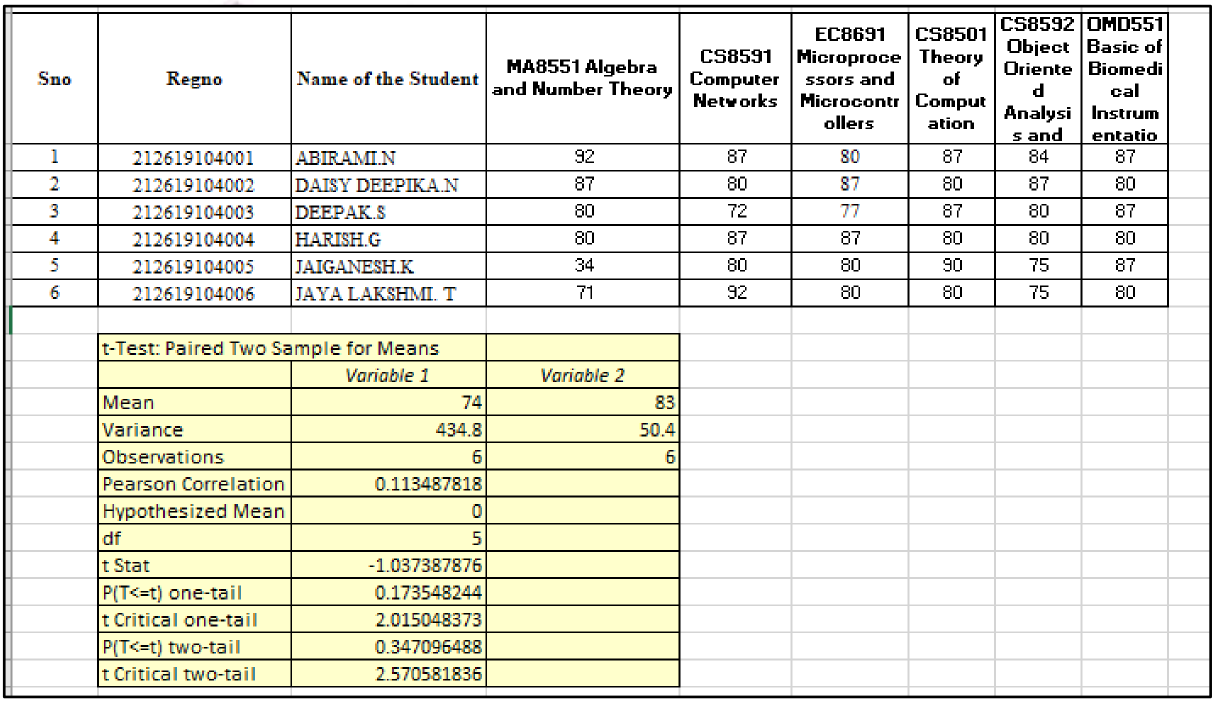
○ New Worksheet Ply:

○ New Workbook

(1)Now select this data range

(2)Now select the any cell for output range to be displayed

OUTPUT:



Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK.S	80	72	77	87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80	80	90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	80	80	75	80

t-Test: Paired Two Sample for Means		
	Variable 1	*Variable 2*
Mean	74	83
Variance	434.8	50.4
Observations	6	6
Pearson Correlation	0.113487818	
Hypothesized Mean	0	
df	5	
t Stat	-1.037387876	
P(T<=t) one-tail	0.173548244	
t Critical one-tail	2.015048373	
P(T<=t) two-tail	0.347096488	
t Critical two-tail	2.570581836	

EX NO: 4c	Perform ANOVA operations
Date:	

AIM:

To Perform ANOVA operations using MS-EXCEL.

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office.

Step 2: Create datasheet for student marks in Ms Excel application.

Step 3: If you haven't already installed the Analysis ToolPak , Click the Microsoft Office button, then click on the Excel Options , and then select Add-Ins , Click Go, check the Analysis ToolPak box, and click Ok

Step 4: Select Data tab, then click on the Data Analysis option, then selects Descriptive Statistics from the list and Click Ok. [Data tab >> Data Analysis >> Anova: Single factor]

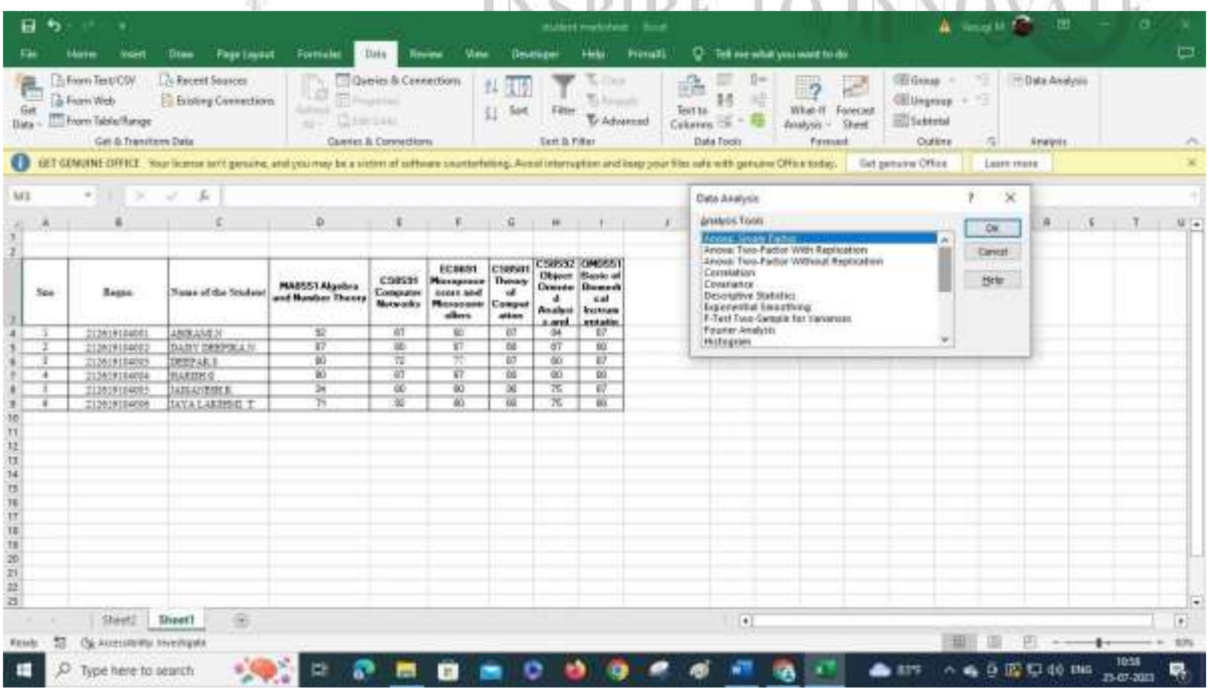
Step 5: In the Input Range we select range of the data and give alpha value as 0.05. then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 6: Then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 7: When you click Ok, you will see the result in the selected output range.

Step 8: Save the excel file and Close the Ms Excel application.

Perform ANOVA



GET GENUINE OFFICE Your license isn't genuine, and you may be a victim of software counterfeiting. Avoid interruption and keep your files safe with genuine Office today. Get genuine Office Learn more

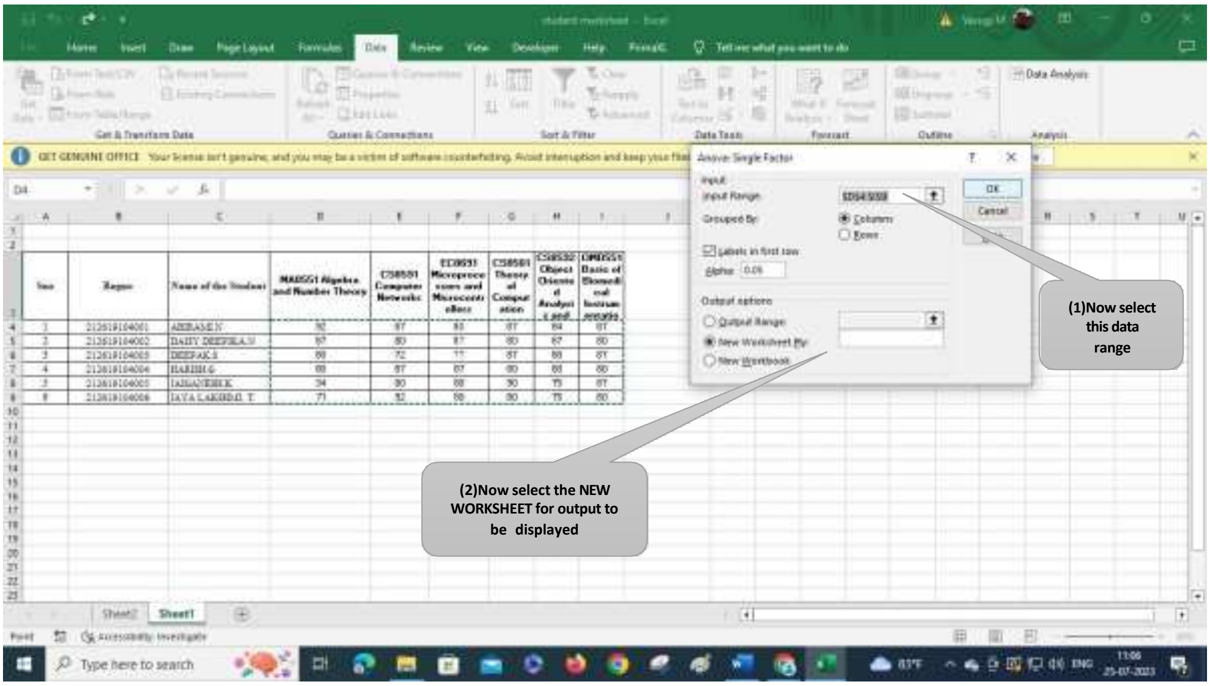
Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentation
1	212619104001	ABIRAMI N	92	67	80	87	94	87
2	212619104002	DAISY DEEPIKA N	87	80	87	88	87	80
3	212619104003	DEEPAK S	80	72	77	87	80	87
4	212619104004	HARISH G	80	67	87	88	80	80
5	212619104005	JAISAKTHI K	34	60	80	36	75	67
6	212619104006	JAYALAKSHMI T	75	32	80	88	75	80

Data Analysis

Analysis Tools

- Anova: Single Factor
- Anova: Two-Factor With Replication
- Anova: Two-Factor Without Replication
- Correlation
- Covariance
- Descriptive Statistics
- Exponential Smoothing
- F-Test Two-Sample for Variances
- Fourier Analysis
- Histogram

OK Cancel Help

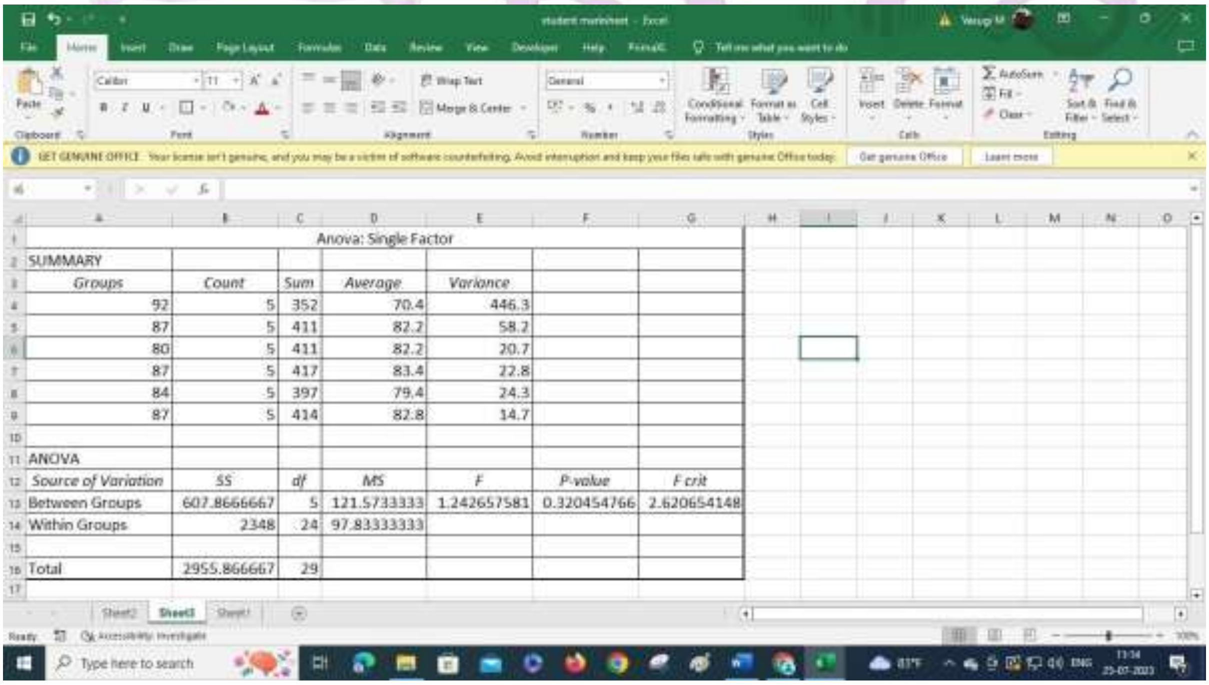


Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and	OMD551 Basic of Biomedical Instrumentatio
1	212619104001	ABIRAMI N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA S	87	80	87	80	87	80
3	212619104003	DEEPAK S	80	72	77	87	80	87
4	212619104004	HARISH G	80	87	87	80	80	80
5	212619104005	JAHANEESH K	34	90	80	90	75	87
6	212619104006	JAYA LAKSHMI T	71	82	80	80	75	80

(1)Now select this data range

(2)Now select the NEW WORKSHEET for output to be displayed

OUTPUT:



Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
92	5	352	70.4	446.3		
87	5	411	82.2	58.2		
80	5	411	82.2	20.7		
87	5	417	83.4	22.8		
84	5	397	79.4	24.3		
87	5	414	82.8	14.7		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	607.8666667	5	121.5733333	1.242657581	0.320454766	2.620654148
Within Groups	2348	24	97.83333333			
Total	2955.866667	29				

EX NO: 5	Perform data pre-processing operations - Handling Missing data
Date:	

AIM:

To handle the missing data in data pre-processing operations on the dataset using MS-EXCEL.

PROCEDURE:

Step 1: Start Ms Excel application in Ms- office.

Step 2: Create datasheet for student marks in Ms Excel application.

Step 3: If you haven't already installed the PrimaXL Addin, install it. Click the PrimaXL tab, choose missing

Step 4: In the Input Range we select marks of all subjects with missing values and select the Choice as "*filling of the missing data by taking average*" or"*filling of the missing data by random pick*".

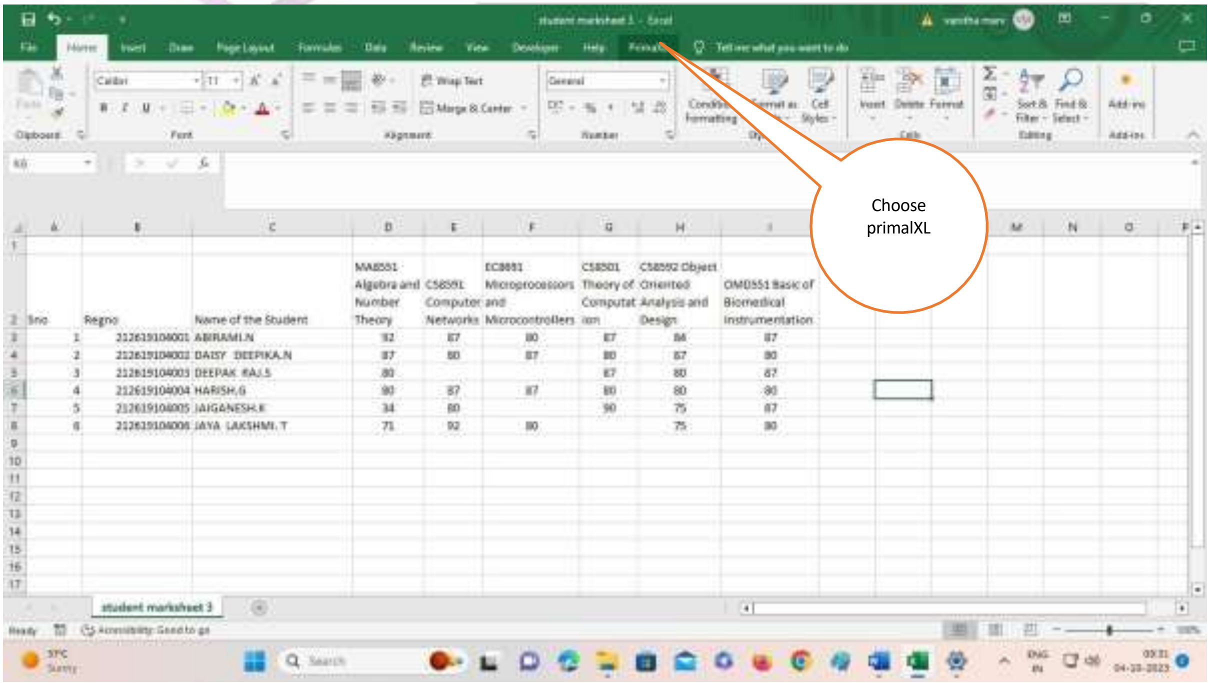
Step 5: Then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 6: Then select Output Range where you want the output to be stored. If you don't specify the output range it will throw output in the new worksheet.

Step 7: When you click Ok, you will see the result in the selected output range.

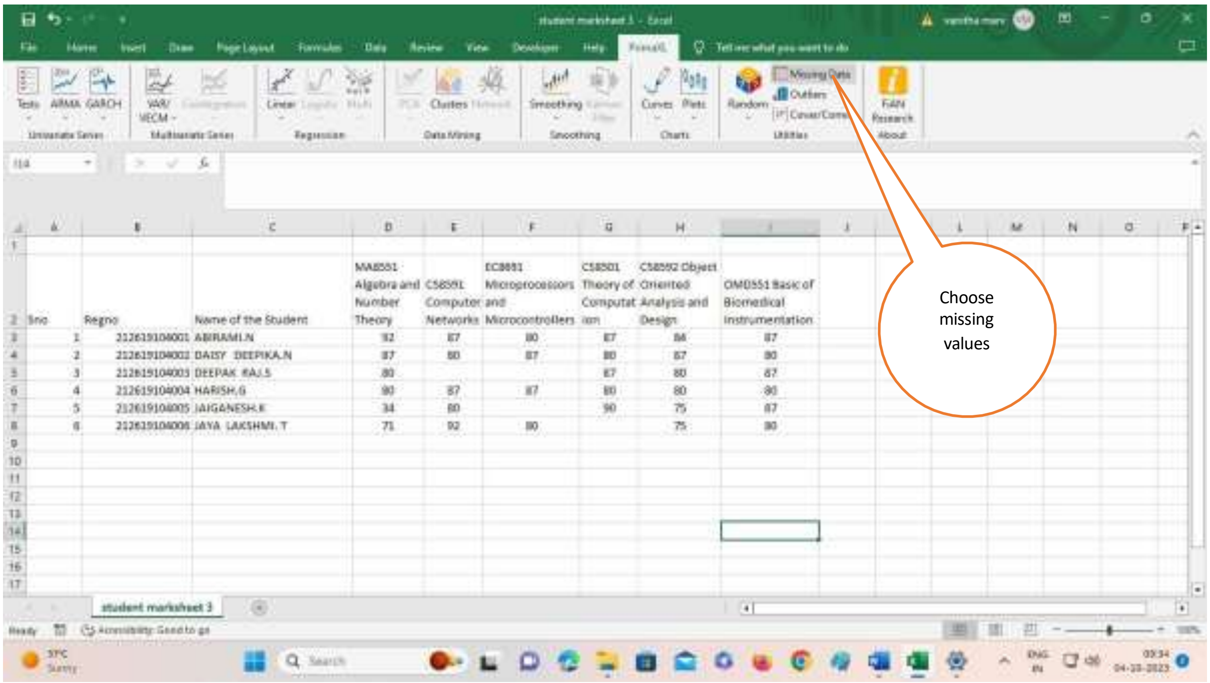
Step 8: Save the excel file and Close the Ms Excel application.

Perform data pre-processing operations - Handling Missing data



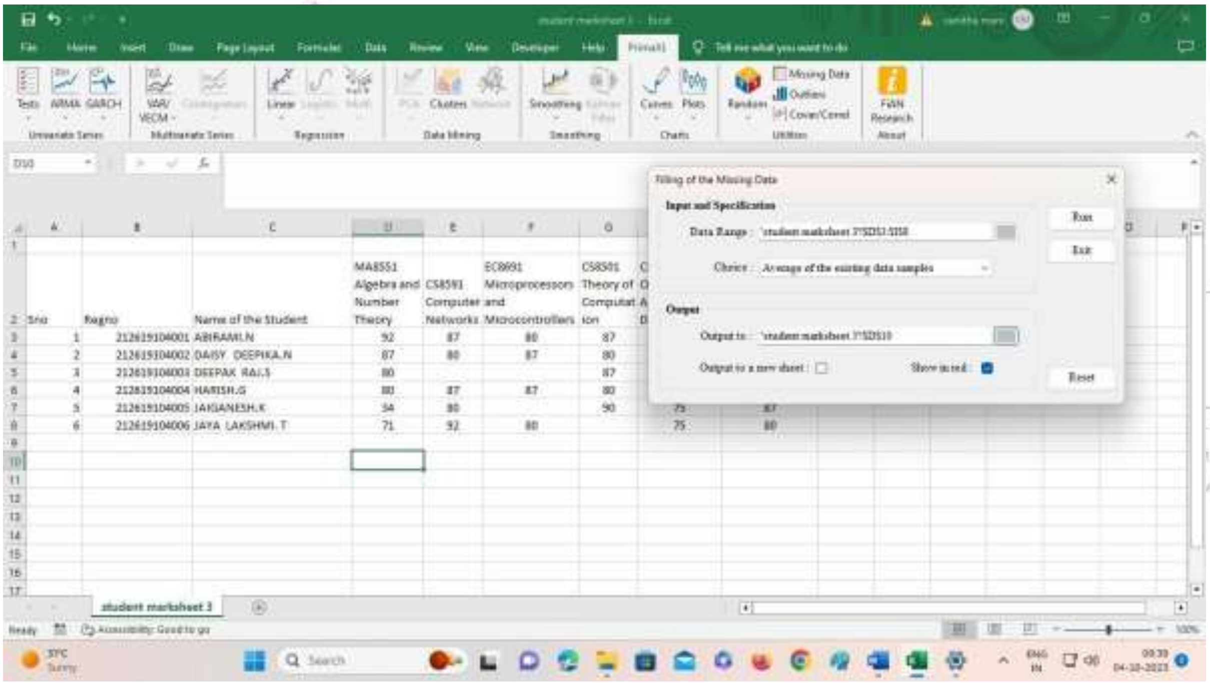
Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and Design	OMD551 Basic of Biomedical Instrumentation
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	87	80
3	212619104003	DEEPAK RAJ.S	80			87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80		90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	80		75	80

Choose primalXL


student marksheet 3 - Excel
File Home Insert Draw Page Layout Formulas Data Review View Developer Help PrimaXL Tell me what you want to do
Tests ARMA GARCH VAR/VECM Cointegration Linear Logistic Multi PCA Clusters Network Smoothing Kalman Filter Curves Plots Random Missing Data Outliers Covar/Correl FaAN Research
Univariate Series Multivariate Series Regression Data Mining Smoothing Charts Utilities About
student marksheet 3

Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microprocessors and Microcontrollers	CS8501 Theory of Computation	CS8592 Object Oriented Analysis and Design	OMD551 Basic of Biomedical Instrumentation
1	212619104001	ABIRAMI.N	92	87	80	87	84	87
2	212619104002	DAISY DEEPIKA.N	87	80	87	80	67	80
3	212619104003	DEEPAK RAJ.S	80			87	80	87
4	212619104004	HARISH.G	80	87	87	80	80	80
5	212619104005	JAIGANESH.K	34	80		90	75	87
6	212619104006	JAYA LAKSHMI. T	71	92	80		75	80

Choose missing values


student marksheet 3 - Excel
Missing Data Outliers Covar/Correl
student marksheet 3
Run
Exit
Reset

Filling of the Missing Data

Input and Specification

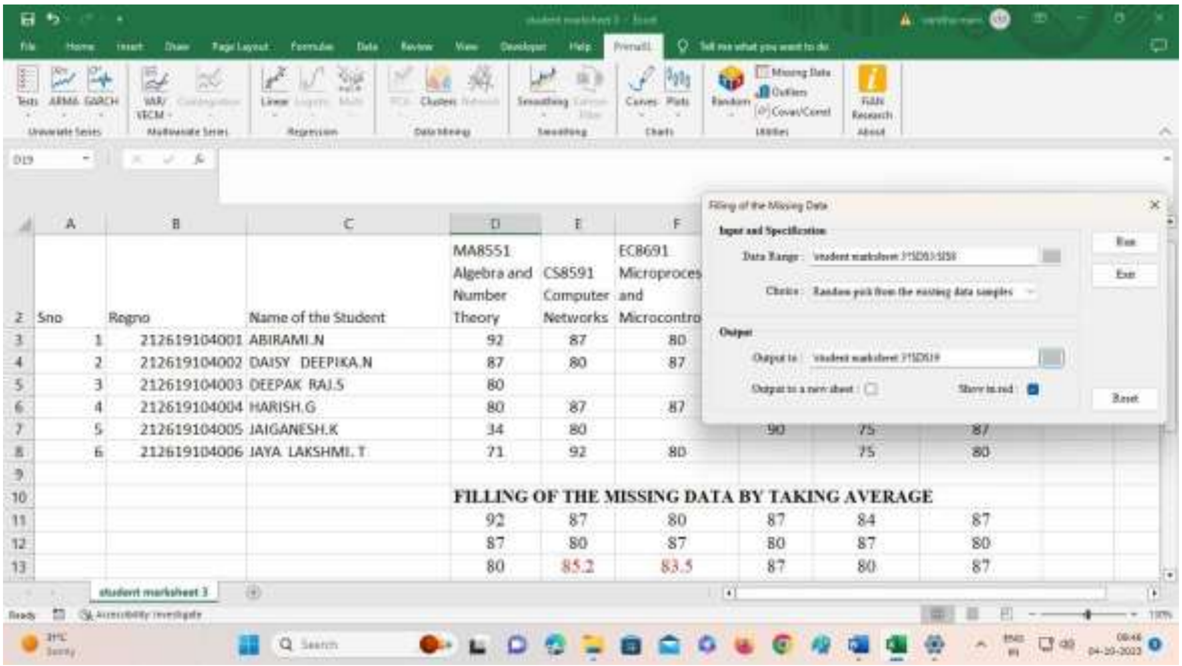
Data Range : 'student marksheet 3'!$D$3:$I$8

Choice : Average of the existing data samples

Output

Output to : 'student marksheet 3'!$D$10

Output to a new sheet : □ Show in red : ☑



Filling of the Missing Data

Input and Specification

Data Range : 'student marksheet 3'!$D$3:$I$8

Choice : Random pick from the missing data samples

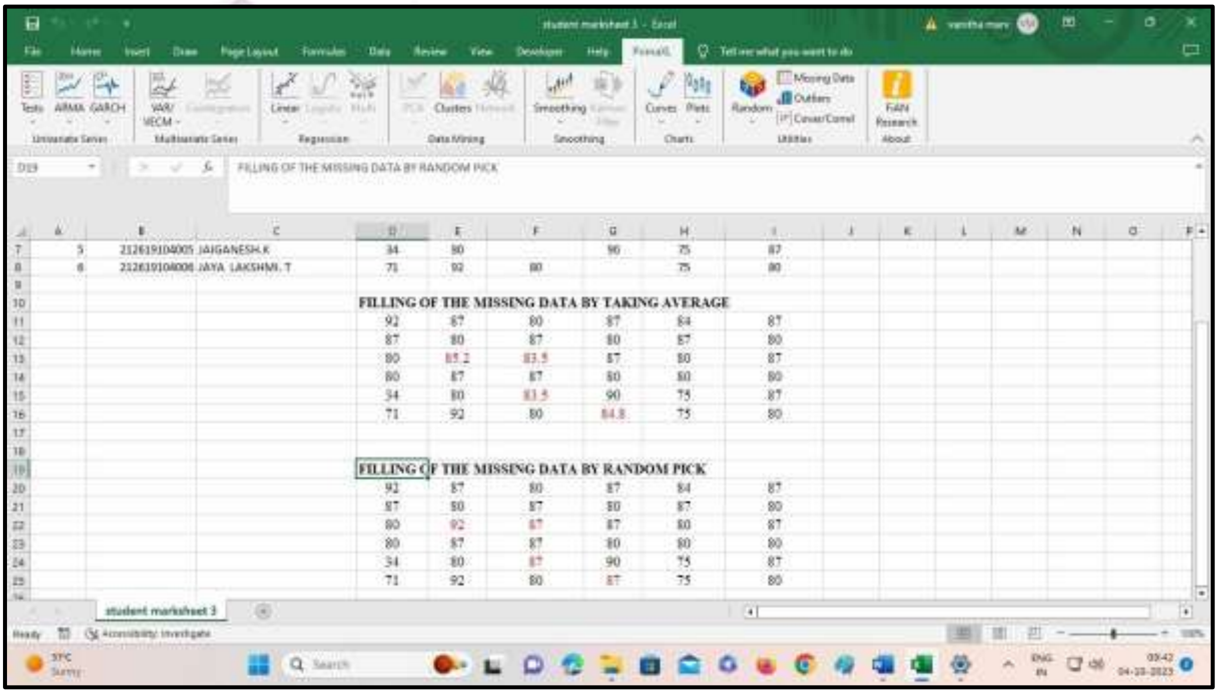
Output

Output to : 'student marksheet 3'!$D$19

Output to a new sheet : □ Show in red : ☑

	A	B	C	D	E	F	G	H	I
2	Sno	Regno	Name of the Student	MA8551 Algebra and Number Theory	CS8591 Computer Networks	EC8691 Microproces and Microcontro			
3	1	212619104001	ABIRAMI.N	92	87	80			
4	2	212619104002	DAISY DEEPIKA.N	87	80	87			
5	3	212619104003	DEEPAK RAJ.S	80					
6	4	212619104004	HARISH.G	80	87	87			
7	5	212619104005	JAIGANESH.K	34	80		90	75	87
8	6	212619104006	JAYA LAKSHMI. T	71	92	80		75	80
9									
10				FILLING OF THE MISSING DATA BY TAKING AVERAGE					
11				92	87	80	87	84	87
12				87	80	87	80	87	80
13				80	85.2	83.5	87	80	87

OUTPUT:



FILLING OF THE MISSING DATA BY RANDOM PICK

	A	B	C	D	E	F	G	H	I
7	5	212619104005	JAIGANESH.K	34	80		90	75	87
8	6	212619104006	JAYA LAKSHMI. T	71	92	80		75	80
9									
10				FILLING OF THE MISSING DATA BY TAKING AVERAGE					
11				92	87	80	87	84	87
12				87	80	87	80	87	80
13				80	85.2	83.5	87	80	87
14				80	87	87	80	80	80
15				34	80	83.5	90	75	87
16				71	92	80	84.8	75	80
17									
18									
19				FILLING OF THE MISSING DATA BY RANDOM PICK					
20				92	87	80	87	84	87
21				87	80	87	80	87	80
22				80	92	87	87	80	87
23				80	87	87	80	80	80
24				34	80	87	90	75	87
25				71	92	80	87	75	80

EX.NO: 6	**R STUDIO DATATYPES**
Date:	

AIM:

To execute the following R-objects

- Vectors
- Lists
- Matrices
- Array
- Factor
- Data Frames

PROCEDURE AND CODE:

Step 1: Open R studio version 4.4.2
Step 2: Write the following code in the console

Vector objects

1. **Logical**

```
> v<-FALSE
> print(class(v))
```

Output:

```
[1] "numeric"
```

2. **Numeric**

```
> v<-178.045
> print(class(v))
```

Output:

```
[1] "numeric"
```

3. **Integer**

```
> v<-20L
> print(class(v))
```

Output:

```
[1] "integer"
```

4. **Complex**

```
> v<-33i+5
> print(class(v))
```

Output:

```
[1] "complex"
```

5. **Character**

```
> v<-"String"
> print(class(v))
```

Output:

```
[1] "character"
```

6. **Raw**

```
> v<-charToRaw("new")
> print(class(v))
```

Output:

```
[1] "raw"
```

VECTORS

```
> apple <- c('red','green',"yellow")
> print(apple)
> print(class(apple))
```

Output:

```
[1] "red" "green" "yellow"
[1] "character"
```

LIST

```
> list1 <- list(c(2,5,3),21.3,sin)
> print(list1)
```

Output:

```
[[1]]
[1] 2 5 3
[[2]]
[1] 21.3
[[3]] function (x). Primitive("sin")
```

MATRICES

```
> M = matrix( c('a','a','b','c','b')
, nrow = 2, ncol = 3, byrow = TRUE)

    > print(M)
```

Output:

```
   [,1] [,2] [,3]
      [1,] "a" "a" "b"
      [2,] "c" "b" "a"
```

ARRAY

```
> a <- array(c('green','yellow'),
      dim = c(3,3,2))
> print(a)
```

Output:

```
, , 1
      [,1]   [,2]          [,3]
[1,] "green"  "yellow" "green"
[2,] "yellow" "green"  "yellow"
[3,] "green"  "yellow" "green"
, , 2
      [,1]     [,2]     [,3]
[1,] "yellow" "green"  "yellow"
[2,] "green"  "yellow" "green"
[3,] "yellow" "green"  "yellow"
```

FACTOR

```
> apple_colors <-c('green','green',
 'yellow','red','red','red','green')
> factor_apple <- factor(apple_colors)
> print(factor_apple)
> print(nlevels(factor_apple))
```

Output:

```
[1]   green     green   yellow    red
                 red       red        green
[1]   3
```

DATA FRAME

```
> BMI <- data.frame(
gender = c("Male","Male","Female"),
c(152, 171.5,165),
weight = c(81,93, 78),
Age = c(42,38,26))
> print(BMI)
```

Output:

```
   gender height weight Age height =
1      Male   152.0              81   42
2      Male   171.5              93   38
3 Female   165.0                 78   26
```

EX.NO: 7	R STUDIO -- VARIABLES, OPERATORS, FUNCTIONS
Date:	

AIM:

To execute R studio Variables, Operators, Functions in R studio console.

PROCEDURE AND CODE:

Step 1: Open **R studio** version **4.4.2**
Step 2: Write the following code in the console

VARIABLES:

1. Variable Assignment

```
 # Assignment using equal operator.
>  var.1 = c(0.5,187,266,33)
>  # Assignment using leftward operator.
>  var.2 <- c("CS","BS")
>  # Assignment using rightward operator.
>  c(FALSE,1) -> var.3
>  print(var.1)
>  cat ("var.1 is ", var.1 ,"\n")
>  cat ("var.3 is ", var.3 ,"\n")
```

Output:

```
[1]        0.5 187.0 266.0   33.0
var.1 is   0.5 187 266 33
var.3 is   0 1
```

2. Data Type of a Variable

```
>  var_x <- "Hello"
>  cat("The class of var_x is
",class(var_x),"\n")
>  var_x <- 34.5
>  cat(" Now the class of var_x is
",class(var_x),"\n")
>  var_x <- 287L
>  cat(" Next the class of var_x becomes
",class(var_x),"\n")
```

Output:

```
The class of var_x is character Now the class
of var_x is numeric
Next the class of var_x becomes integer
```

3. Finding Variables

```
>  print(ls())
```

Output:

```
[1]  "a""BMI"
"colours"  "factor_Lan"  "Lan_names"     "list1"
"M"  "v"   "var.1"   "var.2"
"var.3"   "var_x"
```

4. Deleting Variables

```
rm(var.3) print(var.3)
```

Output:

```
[1] "var.3"
Error  in  print(var.3):  object 'var.3' not found
```

OPERATORS:

ARITHMETIC OPERATORS

1. **Adding two vectors**

```
v <- c( 2,5.5,6)
t <- c(8, 3, 4)
print(v+t)
```

Output:

```
[1]  10.0 8.5 10.0
```

2. **Subtracting two vectors**

```
v <- c( 2,5.5,6)
t <- c(8, 3, 4)
print(v-t)
```

Output:

```
[1]  -6.0 2.5 2.0
```

3. **Multiplying two vectors**

```
v <- c( 2,5.5,6)
t <- c(8, 3, 4)
print(v*t)
```

Output:

```
[1]  16.0 16.5 24.0
```

4. **Dividing two vectors**

```
v <- c( 2,5.5,6)
t <- c(8, 3, 4)
print(v/t)
```

Output:

```
[1]  0.250000 1.833333           1.500000
```

5. **Remainder of first vector**

```
v <- c( 2,5.5,6)
t <- c(8, 3, 4)
print(v%%t)
```

Output:

```
[1]  2.0 2.5 2.0
```

RELATIONAL OPERATORS

1. **Greater than**

```
v <- c(2,5.5,6,9)
t <- c(8,2.5,14,9)
print(v>t)
```

Output:

```
[1] FALSE TRUE FALSE FALSE
```

2. **Less than**

```
v <- c(2,5.5,6,9)
t <- c(8,2.5,14,9)
print(v < t)
```

Output:

```
[1] TRUE FALSE TRUE FALSE
```

3. **Equal to**

```
v <- c(2,5.5,6,9)

t <- c(8,2.5,14,9)
print(v == t)
```

4. Less than or equal to

```
v <- c(2,5.5,6,9)

t <- c(8,2.5,14,9)
print(v<=t)
```

```
[1] TRUE FALSE TRUE TRUE
```

5. Greater than or equal to

```
v <- c(2,5.5,6,9)

t <- c(8,2.5,14,9)
```

Output:

```
[1] FALSE TRUE FALSE TRUE
```

```
print(v>=t)
```

LOGICAL & ASSIGNMENT OPERATORS

1. Logical AND

```
v <- c(3,0,TRUE,2+2i)
t <- c(1,3,TRUE,2+3i)
print(v&&t)
```

Output:

```
[1] TRUE
```

2. Logical OR

```
v <- c(0,0,TRUE,2+2i)
t <- c(0,3,TRUE,2+3i)
print(v||t)
```

Output:

```
[1] FALSE
```

3. Left assignment

```
v1 <- c(3,1,TRUE,2+3i) v2 <<-
c(3,1,TRUE,2+3i) v3 =
c(3,1,TRUE,2+3i)
print(v1)
print(v2)
print(v3)
```

Output:

```
[1] 3+0i 1+0i 1+0i 2+3i
[1] 3+0i 1+0i 1+0i 2+3i
[1] 3+0i 1+0i 1+0i 2+3i
```

4. Right assignment

```
c(3,1,TRUE,2+3i) -> v1
c(3,1,TRUE,2+3i) ->> v2
print(v1)
print(v2)
```

Output:

```
[1] 3+0i 1+0i 1+0i 2+3i
[1] 3+0i 1+0i 1+0i 2+3i
```

FUNCTIONS

Built-in-function

```
v <- c(3,0,TRUE,2+2i)
t <- c(1,3,TRUE,2+3i)
print(v&&t)
```

Output:

```
[1] TRUE
```

Logical OR

```
v <- c(0,0,TRUE,2+2i)
t <- c(0,3,TRUE,2+3i)
print(v||t)
```

Output:

```
[1] FALSE
```

Left assignment

```
v1 <- c(3,1,TRUE,2+3i) v2 <<-
c(3,1,TRUE,2+3i) v3 =
c(3,1,TRUE,2+3i)
print(v1)
print(v2)
print(v3)
```

Output:

```
[1] 3+0i 1+0i 1+0i 2+3i
[1] 3+0i 1+0i 1+0i 2+3i
[1] 3+0i 1+0i 1+0i 2+3i
```

Right assignment

```
c(3,1,TRUE,2+3i) -> v1 c(3,1,TRUE,2+3i) -
>> v2
print(v1)
print(v2)
```

Output:

```
[1] 3+0i 1+0i 1+0i 2+3i
[1] 3+0i 1+0i 1+0i 2+3i
```

Step 3: Execute the following code in R studio

EX.NO: 8	R STUDIO -- STRING, VECTORS, LIST, MATRICES, ARRAY
Date:	

AIM:

To execute R studio String, Vectors, List, Matrices, Array in R studio console.

PROCEDURE AND CODE:

Step 1: Open **R studio** version **4.4.2**
Step 2: Write the following code in the console
Step 3: Execute the following code in R studio

R – Strings

1. String manipulation

```
 a  <- "Hello"
 b  <- 'How'
 c  <- "are you? " print(paste(a,b,c))
print(paste(a,b,c, sep = "-"))
 print(paste(a,b,c, sep = "", collapse = ""))
```

Output:

```
[1] "Hello How are you? " [1]
"Hello-
How-are you? " [1]
"HelloHoware you? "
```

2. Formatting numbers & strings

```
result<-format(23.123456789,digits=9) print(result)
result<-format(c(6,13.14521), scientific =
TRUE) print(result)
```

Output:

```
[1] "23.1234568"
[1] "6.000000e+00"
"1.314521e+01"
[1] "23.47000"
```

3. Counting number of characters in string

```
result <- nchar("Count the number of characters")
print(result)
```

Output:

```
[1] 30
```

4. Changing the case

```
result <- toupper("Changing To Upper") print(result)
result <- tolower("Changing To Lower") print(result)
```

Output:

```
[1] "CHANGING TO UPPER"
 [1] "changing to lower"
```

5. Extracting parts of a string

```
result <- substring("Extract",5,7) print(result)
```

Output:

```
[1] "act"
```

R vectors

1. Vector Creation

```
print("abc"); print(12.5)
print(63L) print(TRUE)
print(2+3i)
print(charToRaw('hello'))
```

Output:

```
[1] "abc"
[1] 12.5
[1] 63
[1] TRUE
[1] 2+3i
[1] 68 65 6c 6c 6f
```

2. Accessing Vector Elements

```
t <-c("Sun","Mon","Tue","Wed"
,"Thurs","Fri","Sat") u <-
t[c(2,3,6)]
print(u)
v <- t[c(TRUE,FALSE,FALSE,FALSE
,FALSE,TRUE,FALSE)]
print(v)
x <- t[c(-2,-5)]
print(x)
y <- t[c(0,0,0,0,0,0,1)] print(y)
```

Output:

```
[1] "Mon" "Tue" "Fri"
  [1] "Sun" "Fri"
  [1] "Sun" "Tue"
"Wed" "Fri" "Sat"
  [1] "Sun
```

3. Vector manipulation

```
v1 <- c(3,8,4,5,0,11)
v2 <- c(4,11,0,8,1,2)
add.result <- v1+v2
print(add.result) sub.result <-
v1-v2 print(sub.result)
multi.result <- v1*v2
print(multi.result) divi.result <-
v1/v2 print(divi.result)
```

Output:

```
[1]  7 19  4 13  1 13
 [1] -1 -3  4 -3 -1  9
 [1] 12 88  0 40  0 22
 [1] 0.7500000 0.7272727
Inf 0.6250000 0.0000000
5.5000000
```

4. Vector element sorting

```
v <- c(3,8,4,5,0,11, -9, 304)
sort.result <- sort(v) print(sort.result)
revsort.result <- sort(v, decreasing = TRUE)
print(revsort.result)
v <- c("Red","Blue","yellow","violet") sort.result <-
sort(v) print(sort.result)
revsort.result <- sort(v,decreasing = TRUE)
print(revsort.result)
```

Output:

```
[1]  -9   0   3   4   5   8  11 304
 [1] 304  11   8   5   4   3   0  -9
 [1] "Blue"   "Red"     "violet"
"yellow"
 [1] "yellow" "violet" "Red"
"Blue"
```

R Lists

1. List creation

```
list_data <- list("Red", "Green", c(21,32,11), TRUE,
51.23, 119.1)
 print(list_data)
```

Output:

```
[[1]] [1] "Red"
[[2]] [1] "Green"
[[3]] [1] 21 32 11
[[4]] [1] TRUE
[[5]] [1] 51.23
[[6]] [1] 119.1
```

2. Naming List Elements

```
list_data <- list(c("Jan","Feb","Mar")
, matrix(c(3,9,5,1,-2,8), nrow = 2), list("green",12.3))
names(list_data) <- c("1st Quarter", "A_Matrix", "A
Inner list") print(list_data)
```

Output:

```
$`1st_Quarter`
  [1] "Jan" "Feb" "Mar"
  $A_Matrix
          [,1] [,2] [,3]
  [1,]         3       5      -2
  [2,]         9       1        8
  $A_Inner_list
  $A_Inner_list[[1]]
  [1] "green"
  $A_Inner_list[[2]]
  [1] 12.3
```

3. List manipulation

```
list_data <- list(c("Jan","Feb","Mar")
, matrix(c(3,9,5,1,-2,8), nrow = 2), list("green",12.3))
names(list_data) <- c("1st Quarter", "A_Matrix", "A
Inner list") list_data[4] <- "New element"
print(list_data[4])
list_data[4] <- NULL print(list_data[4])
list_data[3] <- "updated element" print(list_data[3])
```

Output:

```
[1]][1] "New element"
$<N
A>
NUL
L
$`A Inner list`
[1] "updated element"
```

4. Converting list to vector

```
list1 <- list(1:5) print(list1)
list2 <-list(10:13) print(list2)
v1 <- unlist(list1) v2 <-
unlist(list2) print(v1)
print(v2)
result <- v1+v2
print(result)
```

Output:

```
[[1]][1] 1 2 3 4 5
[[1]][1] 10 11 12 13
[1] 1 2 3 4 5
[1] 10 11 12 13 14
[1] 11 13 15 17 19
```

R matrices

1. Matrix creation

```
M <- matrix(c(3:14), nrow = 4, byrow =
TRUE)
print(M)
N <- matrix(c(3:14), nrow = 4, byrow =
FALSE)
print(N)
rownames = c("row1", "row2", "row3",
"row4")
colnames = c("col1", "col2", "col3")
P <- matrix(c(3:14), nrow = 4, byrow =
TRUE, dimnames = list(rownames, colnames))
print(P)
```

Output:

```
[,1] [,2] [,3]
 [1,]      3     4     5
 [2,]      6     7     8
 [3,]      9    10    11
 [4,]     12    13    14
      [,1] [,2] [,3]
 [1,]      3     7    11
 [2,]      4     8    12
 [3,]      5     9    13
 [4,]      6    10    14
      col1 col2 col3
row1       3     4     5
row2       6     7     8
 row3      9    10    11
 row4     12    13    14
```

2. Accessing elements of matrix

```
rownames = c("row1", "row2", "row3",
"row4")
colnames = c("col1", "col2", "col3")
P <- matrix(c(3:14), nrow = 4, byrow =
TRUE, dimnames = list(rownames,
colnames))
print(P[1,3])
print(P[4,2])
print(P[2,])
print(P[,3])
```

Output:

```
[1] 5
[1] 13
 col1 col2 col3
    6     7     8
     row1 row2 row3 row4
    5     8    11    14
```

3. Matrix Computations

```
matrix1 <- matrix(c(3, 9, -1, 4, 2, 6),
nrow = 2)
print(matrix1)
matrix2 <- matrix(c(5, 2, 0, 9, 3, 4),
nrow = 2)
print(matrix2)
result <- matrix1 + matrix2
cat("Result of addition","\n")
print(result)
result <- matrix1 - matrix2
cat("Result of subtraction","\n")
print(result)
```

Output:

```
[,1] [,2] [,3]
 [1,]      3    -1     2
 [2,]      9     4     6
      [,1] [,2] [,3]
 [1,]      5     0     3
 [2,]      2     9     4
 Result of addition [,1] [,2] [,3]
 [1,]      8    -1     5
 [2,]     11    13    10
 Result of subtraction
```

R Arrays

1. Naming column and rows

```
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
column.names <- c("COL1","COL2","COL3")
row.names <- c("ROW1","ROW2","ROW3")
matrix.names <- c("Matrix1","Matrix2")
result <- array(c(vector1,vector2),dim =
c(3,3,2),dimnames = list(row.names,colum
matrix.names))
print(result)
```

Output:

```
, , Matrix1
         COL1 COL2 COL3
  ROW1     5    10    13
  ROW2     9    11    14
  ROW3     3    12    15
  , , Matrix2
         COL1 COL2 COL3
  ROW1     5    10    13
  ROW2     9    11    14
  ROW3     3    12    15
```

2. Accessing elements of array

```
 vector1 <- c(5,9,3)
 vector2 <- c(10,11,12,13,14,15)
 column.names <- c("COL1","COL2","COL3")
 row.names <- c("ROW1","ROW2","ROW3")
 matrix.names <- c("Matrix1","Matrix2")
 result <- array(c(vector1,vector2),dim =
 c(3,3,2),dimnames = list(row.names,
 column.names, matrix.names))

print(result[3,,2])
 print(result[1,3,1])
 print(result[,,2]
```

Output:

```
COL1 COL2 COL3
    3     12     15
[1] 13
         COL1 COL2 COL3
 ROW1      5     10     13
 ROW2      9     11     14
 ROW3      3     12     1
```

EX.NO: 9	**R STUDIO -- FACTORS, DATA FRAMES, PACKAGES, DATA RESHAPING**
Date:	

AIM:

To execute R studio factors, data frames, packages and data reshaping in R studio console.

PROCEDURE AND CODE:

Step 1: Open **R studio** version **4.4.2**
Step 2: Write the following code in the console
Step 3: Execute the following code in R studio

<u>R – Factors</u>

1. **Factor creation**

```
data <- c("East","West","East","North",
"North","East","West","West") print(data)
print(is.factor(data)) factor_data <-
factor(data) print(factor_data)
print(is.factor(factor_data))
```

Output:

```
[1] "East"  "West" "East"  "North"
"North" "East"  "West"  "West"
[1] FALSE
[1] East  West  East North North
East  West Levels: East North
West
[1] TRUE
```

2. **Factors in data frame**

```
height <- c(132,151,162) weight <-
c(48,49,66)
gender <- c("male","male","female") input_data <-
data.frame height,weight,gender) print(input_data)
print(is.factor(input_data$gender)) print(input_data$gender)
```

Output:

```
   height weight gender
 1       132        48       male
 2       151        49       male
  3       162        66      female
[1] TRUE
[1] male male female Levels: female
male
```

3. **Changing the order of levels**

```
data <- c("East","West","East",”North”) factor_data <-
factor(data) print(factor_data)
new_order_data <- factor(factor_data, levels =
c("East","West","North"))
print(new_order_data)
```

Output:

```
[1] East West East North Levels:
East North West
[1] East West East North Levels:
East West North
```

4. Generating factor levels

```
v <- gl(3, 4, labels = c("Tampa", "Seattle","Boston"))
 print(v)
```

Output:

```
Tampa      Tampa      Tampa
Tampa      Seattle Seattle Seattle
Seattle Boston
[10] Boston   Boston Boston
Levels: Tampa Seattle Boston
```

R- Data Frames

1. Data Frames creation

```
emp.data <- data.frame(emp_name = c("Rick","Dan","Mic"),
salary = c(6230,515.2,611.0),
start_date = as.Date(c("2012-1-1", "2013-9- 2", "2014-3-5")),
stringsAsFactors = FALSE) print(emp.data)
```

Output:

```
emp_name salary start_date
1 Rick 6230               2012-1-1
2 Dan 515.20              2013-9-2
3 Mic 611.00              2014-3-5
```

2. Structure of data frame

```
height <- c(132,151,162) weight <-
c(48,49,66)
gender <- c("male","male","female") input_data <-
data.frame height,weight,gender) print(input_data)
print(is.factor(input_data$gender)) print(input_data$gender)
```

Output:

```
   height weight gender
 1      132        48      male
 2      151        49      male
 3      162        66      female
[1] TRUE
[1] male male female Levels: female
male
```

3. Changing the order of levels

```
data <- c("East","West","East",”North”) factor_data <-
factor(data) print(factor_data)
new_order_data <- factor(factor_data,levels
=c("East","West","North")) print(new_order_data)
```

Output:

```
[1] East West East North Levels: East
North West
[1] East West East North Levels: East
West North
```

4. Generating factor levels

```
v <- gl(3, 4, labels = c("Tampa", "Seattle","Boston"))
print(v)
```

Output:

```
Tampa         Tampa Tampa Tampa
        Seattle Seattle Seattle Seattle
Boston
[10] Boston Boston Boston
Levels: Tampa Seattle
Boston
```

R Packages

1. Check Available R Packages

```
.libPaths()
```

Output:

```
[2] "C:/Program Files/R/R-
3.2.2/library"
```

2. Get the list of all the packages installed

```
Base: The R Base Package
Boot: Bootstrap Functions
Class: Functions for Classification
cluster: "Finding Groups in Data":
Cluster Analysis Extended
Rousseeuw et al. --- etc---
```

R- data Reshaping

1. Joining Columns and Rows in a Data Frame

```
city <- c("Tampa","Seattle")
state <- c("FL","WA”) zipcode <-
c(33602,98104)
addresses <- cbind(city,state,zipcode) cat("# # # # The
First data frame\n") print(addresses)
new.address <- data.frame( city =
    c("Lowry"),
    state = c("CO”), zipcode =
    c("80230"),
    stringsAsFactors = FALSE
)
cat("# # # The Second data frame\n")
print(new.address)
 all.addresses <- rbind(addresses,new.address)
cat("# # # The combined data frame\n")
print(all.addresses)
```

Output:

```
 First data frame city  state  zipcode
"Tampa" "FL" "33602"
"Seattle" "WA"   "98104"

The Second data frame

city   state   zipcode
  1  Lowry CO   80230

The combined data frame

city   state   zipcode
  1    Tampa         FL        33602
  2  Seattle        WA       98104
  3  Lowry          CO       80230
```

2. Melt the Data

```
  molten.ships <- melt(ships, id = c("type","year"))
print(molten.ships)
```

```
    type year variable value 1 A 60 period
    60 2 A 60
    period 75 3 A 65 period 60
    4 A 65 period 75
    .............................................................. 9
    B 60 period 60 10 B 60
    period 75 11 B 65 period 60
    12 B 65 period 75 13 B 70
period 60 ...........
```

3. Cast the Molten Data

```
  recasted.ship <- cast(molten.ships,
  type+year~variable,sum)
   print(recasted.ship)
```

```
type year period service incidents
  1   A   60   135   190                  0
  2   A   65   135   2190                 7
  3   A   70   135   4865                 24
  4   A   75   135   2244                 11
  5   B   60   135   62058                68
  6   B   65   135   48979                111
  7   B   70   135   20163                56
```

EX.NO: 10	**R STUDIO – MEAN, MEDIAN, MODE, LINEAR REGRESSION, MULTIPLE REGRESSION**
Date:	

AIM:

To execute R studio factors, data frames, packages and data reshaping in R studio console.

PROCEDURE AND CODE:

Step 1: Open **R studio** version **4.4.2**
Step 2: Write the following code in the console
Step 3: Execute the following code in R studio

1. Mean finding

```
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)
result.mean <- mean(x) print(result.mean)
```

Output:

```
[1] 8.22
```

2. Applying Trim Option

```
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)
result.mean <- mean(x,trim = 0.3) print(result.mean)
```

Output:

```
[1] 5.55
```

3. Applying NA Option

```
x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)
result.mean <-  mean(x) print(result.mean)
result.mean <- mean(x,na.rm = TRUE) print(result.mean)
```

Output:

```
[1] NA
[1] 8.22
```

4. Median

```
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)
median.result <- median(x) print(median.result))
```

Output:

```
[1] 5.6
```

5.Mode

```
getmode <- function(v){uniqv <- unique(v)
      uniqv[which.max(tabulate(match(v,
 uniqv)))]}
 v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)
 result <- getmode(v)
print(result)
charv <-
```

Output:

```
[1] 2
[1] "it"
```

```
c("o","it","the","it","it")
result <- getmode(charv)
print(result)
```

R – LINEAR REGRESSION

1. Create Relationship Model

```
x <- c(151, 174, 138, 186, 128)
y <- c(63, 81, 56, 91, 47)
relation <- lm(y~x)
print(relation)
```

Output:

```
Call:
lm(formula = y ~ x) Coefficients: (Intercept)
                                   x
-38.4551                 0.6746
```

2. predict() Function

```
x <- c(151, 174, 138, 186)
y <- c(63, 81, 56, 91)
relation <- lm(y~x)
a <- data.frame(x = 170) result
<-  predict(relation,a)
print(result)
```

Output:

```
     1
76.22869
```

3. Visualize the Regression Graphically

```
x <- c(151, 174, 138)
y <- c(63, 81, 56, 91)
relation <- lm(y~x)
png(file = "linearregression.png")
```

1. Input data

```
input <-mtcars[,c("mpg","disp"
,"hp")] print(head(input))
```

Output:

```
            mpg       disp          hp
            Mazda RX4   21.0   160   110
  Mazda RX4 21.0 160                  110
   Datsun          710         22.8          108
  Hornet          21.4   258                  110
  Hornet          18.7   360                  175
  Valiant   18.1   225                        105
```

2. Create Relationship Model & get the Coefficients

```
input <-mtcars[,c("mpg","disp",
"hp","wt")]
model<-lm(mpg~disp+hp+wt,data=input)
print(model)
cat("# # # # The Coefficient Values
```

Output:

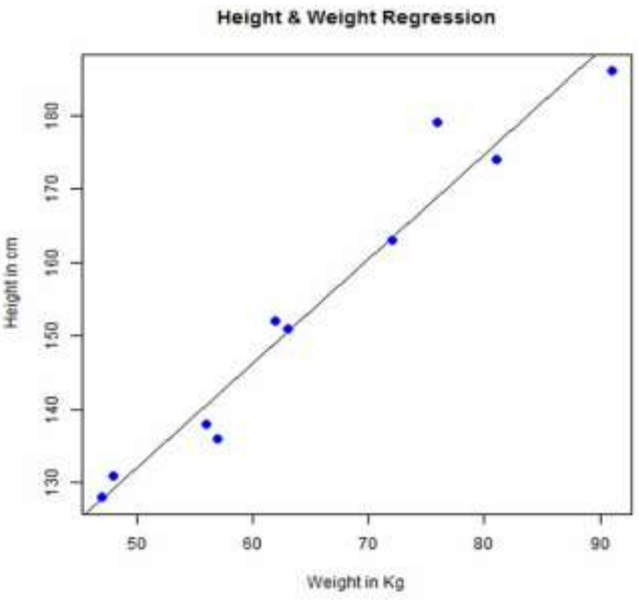
```
Call:
lm(formula = mpg ~ disp + hp
+ wt, data = input) Coefficients:
```

```
# # # ","\n")
a <- coef(model)[1]
print(a)
Xdisp <- coef(model)[2]
Xhp <- coef(model)[3]
Xwt <- coef(model)[4]
print(Xdisp)
print(Xhp)
print(Xwt)
```

```
(Intercept)   disp   hp                    wt
37.10   -0.0009 -0.031 -3.800
###The Coefficient Values### (Intercept)
37.10551
disp -0.0009370091
hp -0.03115655
wt -3.800891
```

```
plot(y,x,col = "blue",main = "Height & Weight Regression", abline(lm(x~y)),cex = 1.3,pch =
16,xlab = "Weight in Kg",ylab = "Height in cm")
dev.off()
```

Output:



EXP NO: 11	**Implement Machine Learning Techniques for Predictive Analytics**
Date:	

Aim:

To implement various machine learning algorithms such as linear regression, decision tree, and random forest for predictive analytics using a sample dataset.

Procedure:

1. Import necessary libraries.
2. Load a sample dataset (e.g., Boston Housing dataset).
3. Preprocess the dataset.
4. Split the dataset into training and test sets.
5. Train models: Linear Regression, Decision Tree, and Random Forest.
6. Evaluate model performance using metrics like RMSE or R².

Code (Python):

```
# Importing libraries
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
boston = load_boston()
df = pd.DataFrame(boston.data, columns=boston.feature_names)
df['PRICE'] = boston.target

# Splitting the dataset
X = df.drop('PRICE', axis=1)
y = df['PRICE']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

# Decision Tree
dt = DecisionTreeRegressor()
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)

# Random Forest
rf = RandomForestRegressor()
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

```
# Evaluation
print("Linear Regression R²:", r2_score(y_test, y_pred_lr))
print("Decision Tree R²:", r2_score(y_test, y_pred_dt))
print("Random Forest R²:", r2_score(y_test, y_pred_rf))
```

Expected Output (example):

```
Linear Regression R²: 0.68
Decision Tree R²: 0.75
Random Forest R²: 0.85
```

EXP NO: 12	**Predict the Customer Credit Risk for Credit Card Dataset using Linear Regression**
Date:	

Aim:

To build a predictive model using linear regression to determine customer credit risk based on features from a credit card dataset.

Procedure:

1. Import necessary libraries.
2. Load a sample credit card dataset.
3. Explore and preprocess the data (handle missing values, encode categorical variables).
4. Split the data into training and test sets.
5. Apply linear regression to predict customer credit risk score or default probability.
6. Evaluate model performance using appropriate metrics (MSE, R^2).

Code (Python):

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder

# Load dataset (replace with actual dataset if needed)
# Sample credit data
data = {
    'Age': [25, 35, 45, 20, 40, 33, 38],
    'Income': [50000, 70000, 120000, 30000, 90000, 85000, 95000],
    'Credit_Score': [600, 700, 750, 580, 720, 710, 730],
    'Default': [0, 0, 0, 1, 0, 0, 0]
}
df = pd.DataFrame(data)

# Define features and target
X = df[['Age', 'Income', 'Credit_Score']]
y = df['Default']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)

# Evaluation
print("Predicted Default Probabilities:", y_pred)
print("MSE:", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))
```

Expected Output (example):

```
Predicted Default Probabilities: [0.03 0.17]
MSE: 0.012
R² Score: 0.55
```

EXP NO: 13	Predict the Customer Credit Risk for Credit Card Dataset using Linear Regression
Date:	

Aim:

To use HR analytics and regression techniques to forecast the future demand for hourly employees based on historical workforce and business data.

Procedure:

1. Import required libraries and load the dataset.
2. Explore and preprocess the data (handle missing values, extract date features).
3. Select relevant features such as number of employees, turnover rate, business growth indicators, etc.
4. Build and train a regression model (Linear Regression or Time Series).
5. Predict the future demand for hourly employees.
6. Evaluate the model performance.

Code (Python) – Using a simplified dummy dataset

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np

# Sample dummy data: monthly demand in past years
data = {
   'Month': list(range(1, 13)),  # Representing January to December
   'Hourly_Employees_Demand': [120, 135, 150, 160, 155, 170, 180, 190, 185, 175, 160,
150]
}
df = pd.DataFrame(data)

# Features and target
X = df[['Month']]
y = df['Hourly_Employees_Demand']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Model
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Predict for next 6 months (months 13 to 18)
```

```
future_months = pd.DataFrame({'Month': list(range(13, 19))})
future_demand = model.predict(future_months)

# Print predictions
print("Predicted hourly employee demand for next 6 months:")
print(future_demand)

# Visualization
plt.scatter(X, y, color='blue', label='Actual')
plt.plot(df['Month'], model.predict(X), color='red', label='Trend')
plt.xlabel("Month")
plt.ylabel("Hourly Employees Demand")
plt.title("Hourly Employee Demand Forecast")
plt.legend()
plt.grid(True)
plt.show()
```

Expected Output:

```
Predicted hourly employee demand for next 6 months:
[195.5, 200.2, 204.9, 209.6, 214.3, 219.0]
```

The red line in the plot will show the forecast trend. You can replace this dummy data with actual HR datasets for more realistic modeling.

EXP NO: 14	Apply Analytics for Forecasting and Inventory Planning for a Large Retailer
Date:	

Aim:

To forecast future product demand and plan inventory efficiently using time series analysis techniques to minimize understocking and overstocking in a retail environment.

Procedure:

1. Import necessary libraries.
2. Load a retail dataset with historical sales data.
3. Preprocess and explore the dataset (convert dates, check for trends/seasonality).
4. Apply a time series forecasting model (e.g., ARIMA or simple moving average).
5. Use the forecast to recommend inventory planning.
6. Visualize the results with forecast overlay.

Code (Python) – Simple time series forecasting using moving average

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Sample monthly sales data for a product (dummy data)
data = {
    'Month': pd.date_range(start='2023-01-01', periods=12, freq='M'),
    'Sales': [150, 160, 170, 180, 175, 185, 190, 200, 210, 205, 215, 225]
}
df = pd.DataFrame(data)
df.set_index('Month', inplace=True)

# Plot original data
plt.figure(figsize=(10,5))
plt.plot(df['Sales'], label='Actual Sales')
plt.title('Monthly Sales Data')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.grid(True)
plt.legend()
plt.show()

# Apply ARIMA model for forecasting
model = ARIMA(df['Sales'], order=(1, 1, 1))  # p=1, d=1, q=1
model_fit = model.fit()

# Forecast next 6 months
forecast = model_fit.forecast(steps=6)
```

```
print("Sales forecast for next 6 months:")
print(forecast)

# Plot forecast
plt.figure(figsize=(10,5))
plt.plot(df['Sales'], label='Actual Sales')
plt.plot(pd.date_range(start='2024-01-01', periods=6, freq='M'), forecast, label='Forecast', linestyle='--')
plt.title('Sales Forecast vs Actual')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.grid(True)
plt.legend()
plt.show()
```

Expected Output:

```
Sales forecast for next 6 months:
2024-01-31    230.2
2024-02-29    235.6
2024-03-31    240.9
2024-04-30    246.1
2024-05-31    251.3
2024-06-30    256.4
Freq: M, dtype: float64
```

EXP NO: 15	**Perform Predictive Analytics for Customers' Behaviour in Marketing and Sales**
Date:	

Aim:

To use predictive analytics techniques to analyze and predict customer behavior for marketing and sales strategies using machine learning models such as classification or clustering.

Procedure:

1. Import necessary libraries.
2. Load a customer behavior dataset (e.g., online retail or marketing campaign data).
3. Preprocess data (handle missing values, encode categorical variables, scale features).
4. Choose target behavior (e.g., whether a customer will respond to a campaign).
5. Apply classification model (e.g., Logistic Regression, Decision Tree).
6. Evaluate the model using metrics such as accuracy, confusion matrix, and classification report.

Code (Python) – Classification to predict if a customer buys a product

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Sample dataset: customer behavior for marketing
data = {
    'Age': [22, 25, 47, 52, 46, 56, 55, 60],
    'Salary': [30000, 32000, 70000, 85000, 62000, 90000, 94000, 97000],
    'Purchased': [0, 0, 1, 1, 1, 1, 1, 1]  # 1 means customer purchased
}
df = pd.DataFrame(data)

# Features and target
X = df[['Age', 'Salary']]
y = df['Purchased']

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.25,
random_state=0)

# Logistic Regression model
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

Expected Output:

```
Accuracy: 1.0
Confusion Matrix:
[[1 0]
 [0 1]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```