Great Learning

CERTIFICATE OF COMPLETION

Presented to

Shannee Ahirwar

For successfully completing a free online course

Python for Data Analysis in Hindi

Provided by

Great Learning Academy

(On April 2023)

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

B. Tech. VI Semester

Jan - June 2023

Lab Assignment

On

Data Analytics Lab [CS 605]

Submitted To:

Archana Choubey

Senior Assistant Professor

Submitted By:

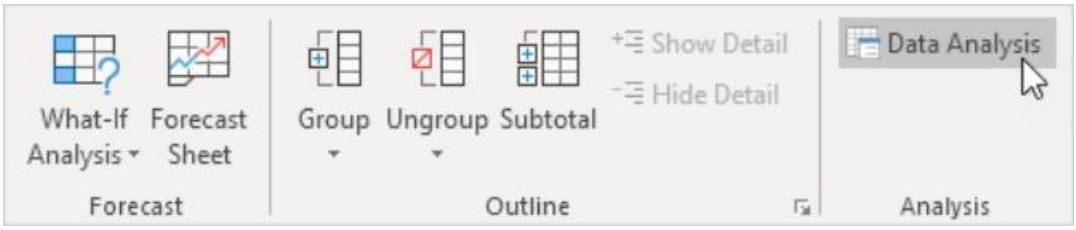
Shannee Ahirwar

Enrollment No. 0827CS201225

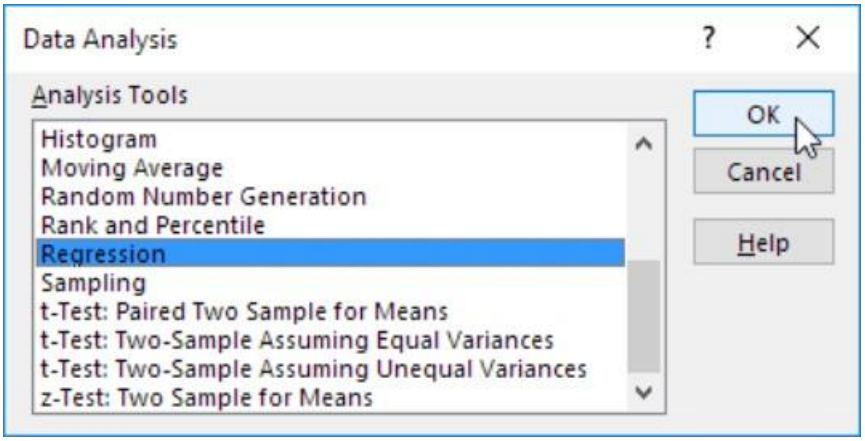
EXPERIMENT– 1

AIM-Apply Linear Regression on a dataset using MS-Excel as a tool.

1. On the Data tab, in the Analysis group, click Data Analysis.



2. Select Regression and click OK.



3. Select the Y Range This is the predictor variable (also called dependent variable).

4. Select the X Range These are the explanatory variables (also called independent variables). These columns must be adjacent to each other.

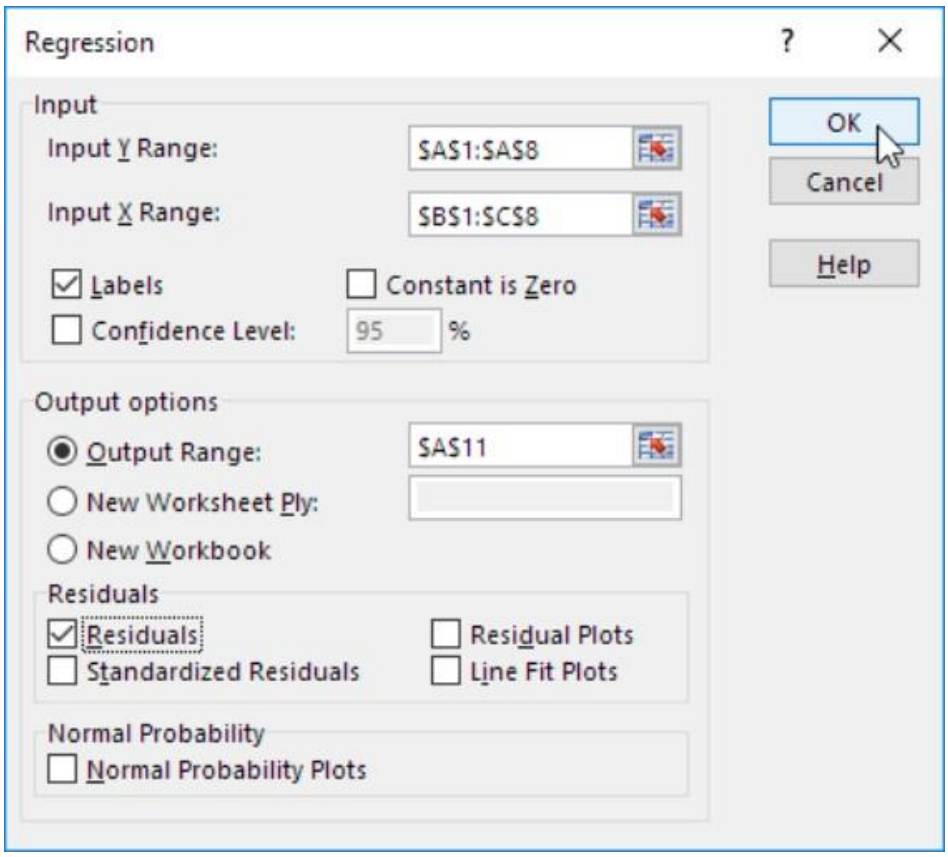
5. Check Labels.

6. Click in the Output Range box and select cell A11.

7. Check Residuals.

8. Click OK.

Excel will display the regression results in the specified output range. The results will include coefficients, standard errors, t-values, p-values, and R-squared value, which can be used to interpret the results of the Linear Regression analysis



For Dataset:

Quantity Sold	Price	Advertisement
2000	2	200
3000	2	250
5000	3	300
10000	2	500

Summary Output:

Regression Statistics	
Multiple R	0.99915074
R Square	0.99830221
Adjusted R Square	0.99490662
Standard Error	254.000254
Observations	4

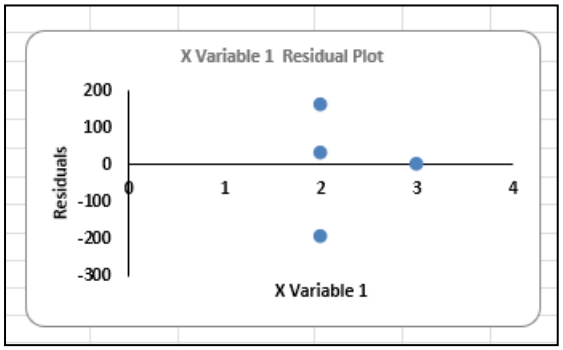
ANOVA

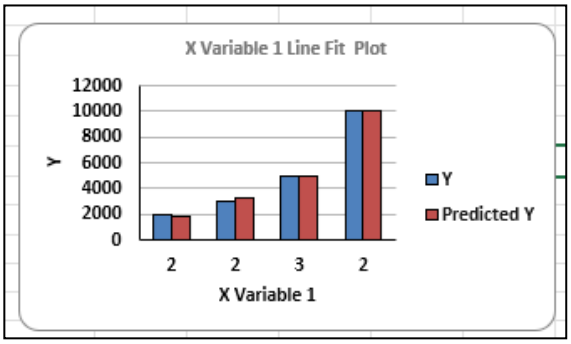
	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	37935483.87	1.9E+07	294	0.041204282
Residual	1	64516.12903	64516.1		
Total	3	38000000			

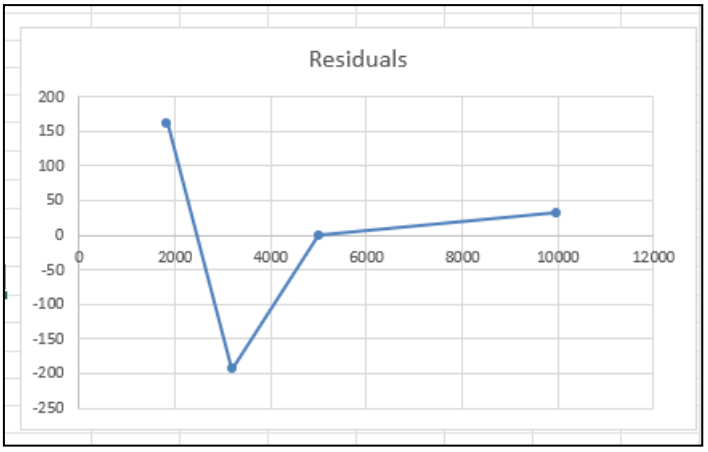
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*Upper 95.0%*
Intercept	-4483.87097	777.5465028	-5.7667	0.10931	-14363.536	5395.79	-14364	5395.79409
X Variable 1	451.612903	293.8849542	1.5367	0.36727	-3282.54949	4185.78	-3282.5	4185.7753
X Variable 2	27.0967742	1.117452134	24.2487	0.02624	12.8981986	41.2953	12.8982	41.2953498

RESIDUAL OUTPUT

Observation	*Predicted Y*	*Residuals*
1	1838.70968	161.2903226
2	3193.54839	-193.5483871
3	5000	0
4	9967.74194	32.25806452







EXPERIMENT– 2

AIM-Consider the dataset shared in the classroom and perform the following tasks on excel.

1. **Find Total cost of advertisement Row wise.**
2. **Find which advertisement mode contributed significantly in Sales**
3. **Find how many times sales crosses the threshold of 2000 crores**
4. **Draw scatter chart between TV and Sales, Radio and Sales and Newspaper and sales**
5. **Draw Line chart between TV and Sales, Radio and Sales and Newspaper and Sales**

1)

	TotalCost
TV	=SUM(A2:A201)
radio	=SUM(B2:B201)
newspaper	=SUM(C2:C201)
sales	=SUM(D2:D201)

Q1:

	TotalCost
TV	29408.5
radio	4652.8
newspaper	6110.8
sales	280450

2) It can be done by finding the correlation

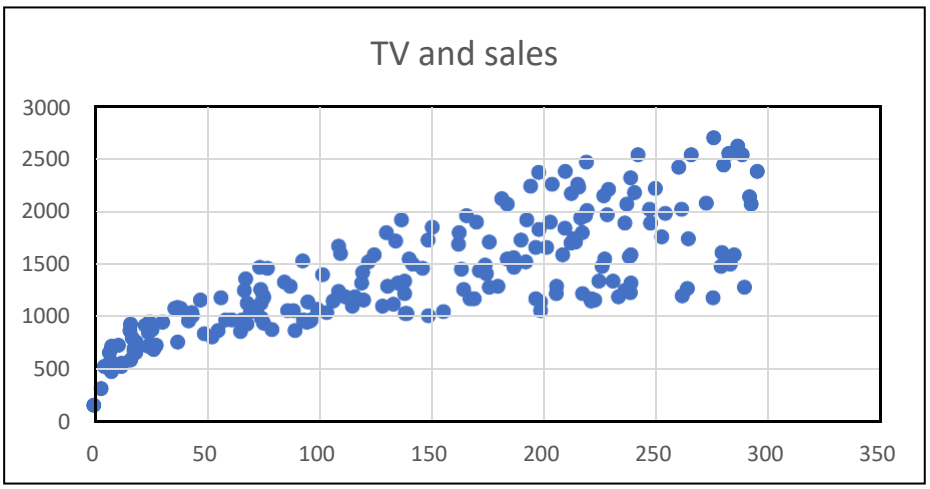
	TV	*radio*	*newspaper*
TV	1		
radio	0.054809	1	
newspaper	0.056648	0.354104	1

3)

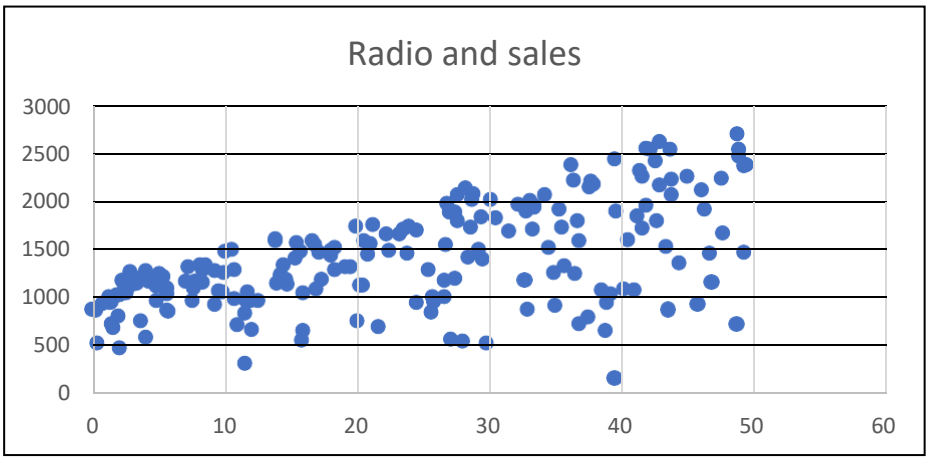
Q3:	31
	"=COUNTIF(D2:D201,">2000")"

4) Scatter Charts:

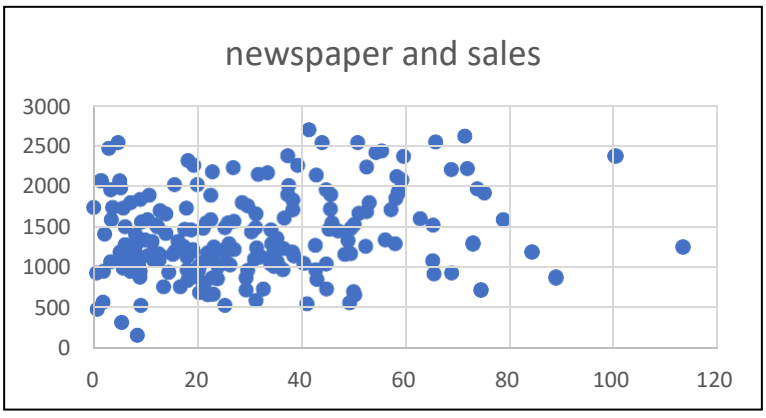
a) scatter chart between tv and sales:



b) Scatter Chart between Radio and sales:

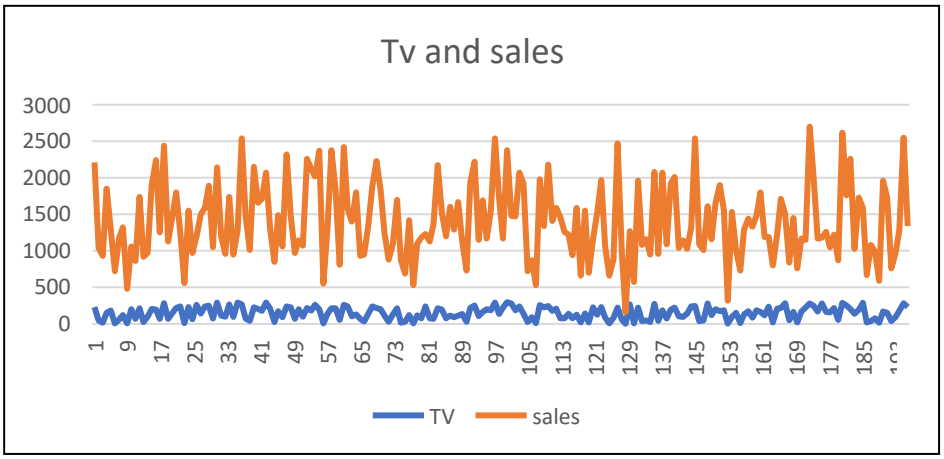


c) Scatter chart between newspaper and sales.

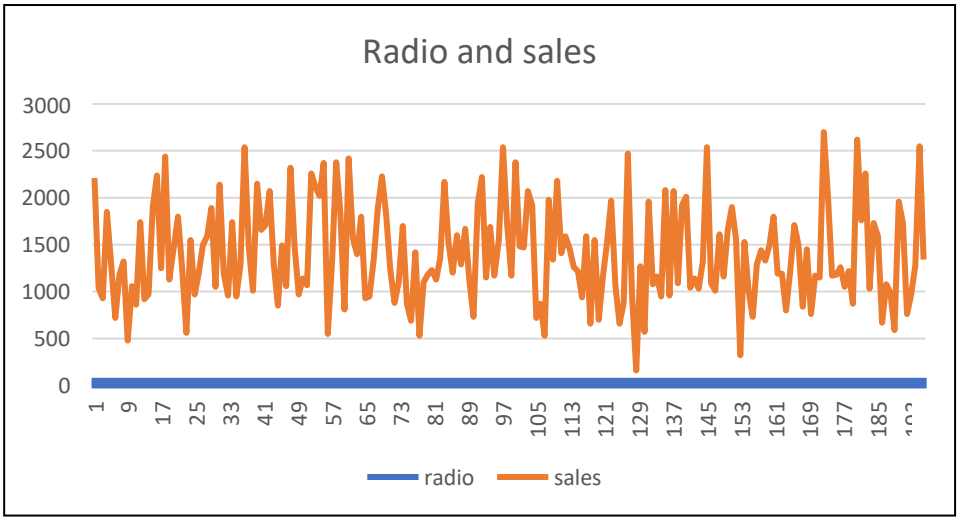


5) Line charts:

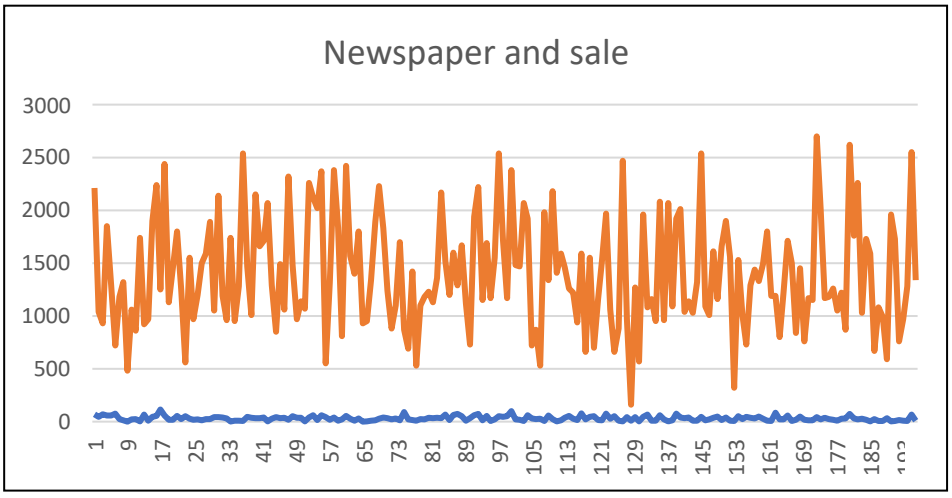
a) Line chart between tv and sales.



b) Line chart between Radio and sales.



C) Line chart between Newspaper and sales.



EXPERIMENT– 3

AIM-Consider a dataset shared in the classroom of semester results of students.

Consider the grades as-

Grades	Value
A+	10
A	9
B+	8
B	7
C+	6
C	5
D	4
F	Less than 4
ABS	Absent

Clean the data

1. **Removal of unwanted observations**
2. **Fixing Structural errors**
3. **Managing Unwanted outliers**
4. **Handling missing data**

Find out the following

1. **Topper of the class.**
2. **No of students failed.**
3. **Subject wise Result.**
4. **No of students passed.**
5. **Other observations if any.**

Removal of unwanted observations:

- Identify the unwanted observations in the dataset.
- Select the rows containing the unwanted observations.
- Right-click on the selected rows and click on "Delete" to remove the unwanted observations.

Fixing Structural errors:

- Identify the structural errors in the dataset.
- Correct the errors by editing the data in the respective cells.
- Verify that the corrections are reflected in the dataset.

Managing Unwanted outliers:

- Identify the outliers in the dataset.
- Determine the criteria for identifying outliers, e.g. values that are more than 3 standard deviations away from the mean.
- Replace the outlier values with either the mean, median or another appropriate value.
- Verify that the changes have been made to the dataset.

Handling missing data:

- Identify the missing data in the dataset.
- Determine the appropriate method for handling missing data, e.g. filling in the missing values with the mean or median of the column.

1. To find the topper of the class, follow these steps:
    - Select the range of grades for each subject.
    - Go to the "Formulas" tab and click on "Max" function.
    - Select the range of grades for each subject and click "Ok".
    - Repeat this for all the subjects to find the maximum grade obtained by any student in each subject.
    - Identify the student with the highest CGPA.

	Topper	8.33	"=MAX(L3:L63)"

2. To find the number of students failed, follow these steps:
    - Count the number of students who have obtained grades below the passing threshold for each subject.
    - Add up these counts for all subjects to find the total number of failed students.

(from fail col)			
fail	3		"=COUNTIF(N3:N63,">=1")"

3. To find the subject-wise result, follow these steps:
    - Identify the passing threshold for each subject.
    - Count the number of students who have obtained grades above the passing threshold for each subject.
    - Add up these counts for all subjects to find the total number of students passed in each subject.

Subjects	pass	fail
IT-601	60	1
IT-602	60	1
IT-603	60	1
IT-604	60	1
IT-601P	61	0
IT-602P	61	0
IT-605P	61	0
IT-606P	61	0
IT-608P	61	0

4. To find the number of students passed, follow these steps:
    - Count the number of students who have obtained grades above the passing threshold for all subjects. We can use countif to find the required answer.

5. Other observations:
    - We can also calculate the average and standard deviation of grades for each subject to get an idea of the distribution of grades.
    - We can also create charts or graphs to visualize the distribution of grades and identify any outliers or trends.
    - To calculate the average and standard deviation of the total marks, use the "AVERAGE" and "STDEV" functions. For example, if the "Total Marks" column is in column D, the formulas would be "=AVERAGE (D2:D100)" and "=STDEV (D2:D100)"

EXPERIMENT– 4

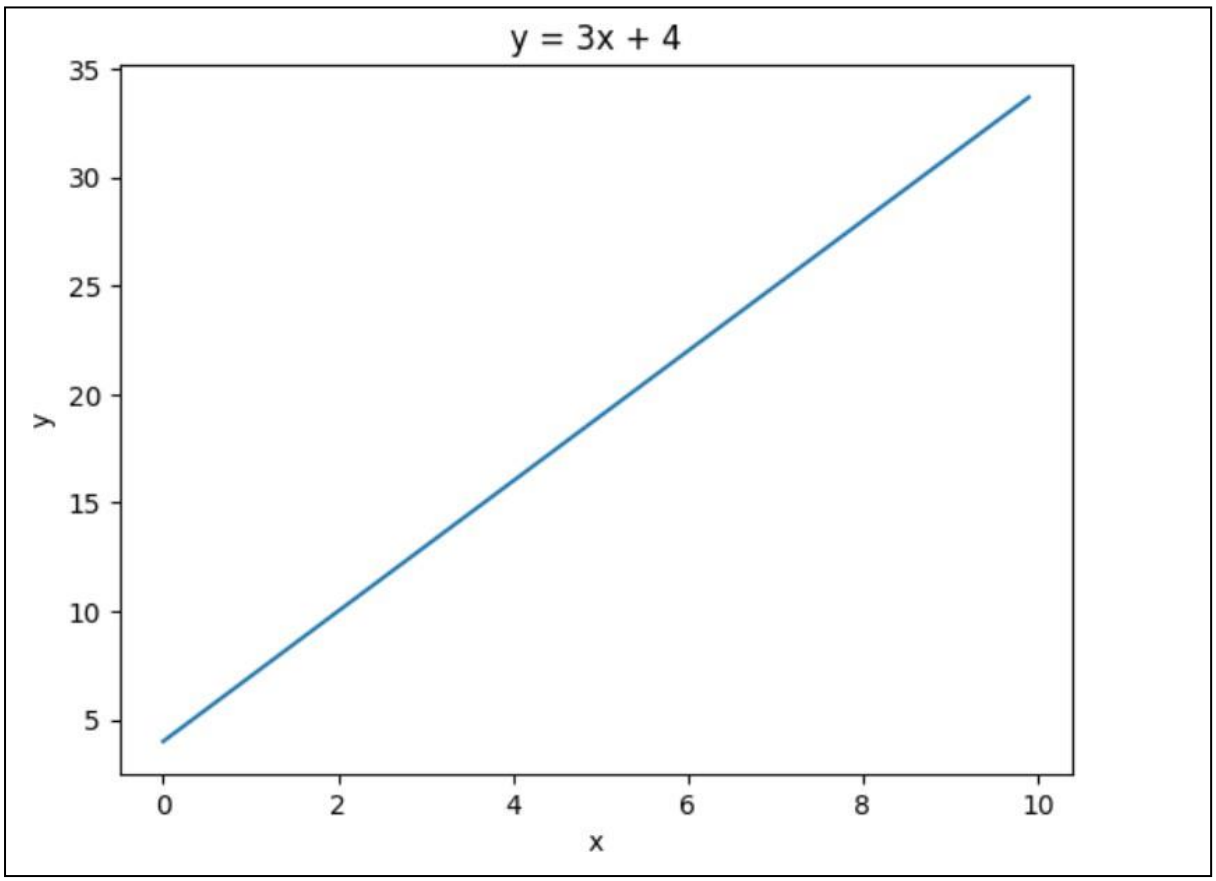
AIM-Using python take an array of 10 items and find out median, mean, average, standard deviation and variance. Also Plot the linear equation y=3*x+4 for different values of x.

```
import numpy as np
import matplotlib.pyplot as plt


arr = np.array([2, 4, 5, 6, 7, 8, 10, 12, 14, 16])


median = np.median(arr)
mean = np.mean(arr)
average = np.average(arr)
std_dev = np.std(arr)
variance = np.var(arr)


print("Array:", arr)
print("Median:", median)
print("Mean:", mean)
print("Average:", average)
print("Standard deviation:", std_dev)
print("Variance:", variance)


x = np.arange(0, 10, 0.1)
y = 3 * x + 4
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('y = 3x + 4')
plt.show()

```

OUTPUT:

```
Array: [ 2  4  5  6  7  8 10 12 14 16]
Median: 7.5
Mean: 8.4
Average: 8.4
Standard deviation: 4.184205702134504
Variance: 17.44
```

For Y=3x+4



EXPERIMENT– 5

AIM-Code these programming problems in python

1.Create a series with 100 random numbers.

2. Print Transpose of a Data Frame.

3. Sort a given Dataframe.

4. Import temp.csv file from local disk and Print its contents.

1) Code:

```
import pandas as pd
import numpy as np

s = pd.Series(np.random.randn(100))
print(s)
```

Output:

```
0     1.988261
1     1.038622
2     1.210303
3     0.547669
4    -0.446077
        ...
95   -0.185388
96    0.416527
97   -0.068722
98    0.288698
99    0.284054
Length: 100, dtype: float64
```

2) Code:

```
import pandas as pd

# create a data frame
df = pd.DataFrame({'A': [1, 4, 2, 3], 'B': [4, 2, 1, 5], 'C': [7, 9, 8, 6]})

# sort the data frame by column A in ascending order
df = df.sort_values('A', ascending=True)
print(df)
```

Output:

```
   0  1  2
A  1  2  3
B  4  5  6
C  7  8  9
```

3) Code:

```
import pandas as pd

# create a data frame
df = pd.DataFrame({'A': [1, 4, 2, 3], 'B': [4, 2, 1, 5], 'C': [7, 9, 8, 6]})

# sort the data frame by column A in ascending order
df = df.sort_values('A', ascending=True)
print(df)
```

Output:

```
   A  B  C
0  1  4  7
2  2  1  8
3  3  5  6
1  4  2  9
```

4) Code:

```
import pandas as pd

# read the csv file into a data frame
df = pd.read_csv('temp.csv')

# print the contents of the data frame
print(df)
```

EXPERIMENT– 6

AIM-Study of setting up the environment in the R studio.

Setting up the environment in R Studio involves the following steps:

1. Install R: First, you need to download and install R from the R project website (https://www.r-project.org/). Choose the appropriate version for your operating system and follow the installation instructions.

2. Install R Studio: Next, you need to download and install R Studio, which is an integrated development environment (IDE) for R programming. You can download the latest version of R Studio from the R Studio website (https://www.rstudio.com/products/rstudio/download/). Choose the appropriate version for your operating system and follow the installation instructions.

3. Open R Studio: Once you have installed R and R Studio, open R Studio. You will see four panels: the source panel on the top left, the console panel on the bottom left, the environment panel on the top right, and the plots panel on the bottom right.

4. Install packages: R has a vast collection of packages that can be installed to extend its functionality. You can install packages by running the following command in the console panel: install.packages("package_name"). Replace "package_name" with the name of the package you want to install. For example, to install the "ggplot2" package, run the following command: install.packages("ggplot2").

5. Load packages: Once you have installed a package, you need to load it into the environment to use its functions. You can load a package by running the following command in the console panel: library(package_name). Replace "package_name" with the name of the package you want to load. For example, to load the "ggplot2" package, run the following command: library(ggplot2).

6. Import data: You can import data into R Studio by clicking on the "Import Dataset" button in the environment panel or by running the following command in the console panel: read.csv("file_path"). Replace "file_path" with the path to your CSV file.

7. Start coding: Now you are ready to start coding in R Studio. You can write your code in the source panel and run it in the console panel by highlighting the code and pressing "Ctrl + Enter" on Windows or "Cmd + Enter" on Mac.

EXPERIMENT– 7

AIM-Perform statistical analysis and visualization of the dataset with R.

1. Load the necessary packages: Depending on the analysis and visualization you want to perform, you might need to load specific packages. For example, if you want to create plots, you can load the ggplot2 package using the following command: library(ggplot2)

2. Import the dataset: You can import the dataset into R using functions like read.csv(), read_excel(), or read.table(). For example, to import a CSV file named "mydata.csv", you can use the following command: mydata <- read.csv("mydata.csv")

3. Check the structure of the dataset: You can use the str() function to check the structure of the dataset, including the data type of each column, the number of observations, and the presence of missing values. For example, to check the structure of the "mydata" dataset, you can use the following command: str(mydata)

4. Calculate descriptive statistics: You can use the summary() function to calculate basic descriptive statistics, such as the mean, median, standard deviation, minimum, and maximum values of each column. For example, to calculate the descriptive statistics of the "mydata" dataset, you can use the following command: summary(mydata)

5. Perform statistical tests: Depending on your research question and hypothesis, you can perform different statistical tests, such as t-tests, ANOVA, regression analysis, or correlation analysis. You can use functions like t.test(), lm(), or cor.test() to perform these tests. For example, to perform a t-test to compare the mean values of two groups in the "mydata" dataset, you can use the following command: t.test(mydata$group1, mydata$group2)

6. Visualize the data: R provides a wide range of visualization options to create plots, charts, and graphs. You can use functions like plot(), ggplot(), or hist() to create visualizations. For example, to create a scatter plot of the "mydata" dataset with "column1" on the x-axis and "column2" on the y-axis, you can use the following command: plot(mydata$column1, mydata$column2) or ggplot(mydata, aes(x = column1, y = column2)) + geom_point()

7. Interpret the results: After performing the analysis and visualization, you need to interpret the results and draw conclusions based on your research question and hypothesis. You can use the summary statistics, statistical tests, and visualizations to support your interpretation.

EXPERIMENT– 8

AIM-Write a program for prediction using linear regression with Matlab.

```
% Load the dataset
data = readtable('dataset.csv');

% Separate the predictors (X) and response variable (y)
X = data(:, 1:end-1);
y = data(:, end);

% Split the data into training and testing sets
[trainX, trainY, testX, testY] = splitData(X, y, 0.8);

% Train the linear regression model using the training data
model = fitlm(trainX, trainY);

% Use the model to make predictions on the test data
predictions = predict(model, testX);

% Evaluate the performance of the model using mean squared error
mse = mean((testY - predictions).^2);

% Visualize the results using a scatter plot
scatter(testX, testY);
hold on;
plot(testX, predictions);
legend('Actual', 'Predicted');
xlabel('Predictor Variable');
ylabel('Response Variable');
title('Linear Regression Results');
```

In this program, we first load the dataset and separate the predictor variables (X) and response variable (y). We then split the data into training and testing sets using the splitData function, and train a linear regression model using the training data. We use the trained model to make predictions on the test data, and evaluate the performance of the model using mean squared error. Finally, we visualize the results using a scatter plot, with the actual values as points and the predicted values as a line.

EXPERIMENT– 9

AIM-Case study on application areas of Hypothesis testing, Probability Distribution Curve and Skewness in Data Analytics.

Case study on the application areas of Hypothesis testing, Probability Distribution Curve and Skewness in Data Analytics is as follows:

Case Study: Evaluating the Impact of a New Marketing Campaign

A company has launched a new marketing campaign to increase the sales of a particular product. The company wants to know whether the new campaign has been successful in achieving its goal. They have collected data on the sales of the product before and after the campaign.

Hypothesis Testing:

To test whether the new campaign has been successful, we can use hypothesis testing. We can set up the following null and alternative hypotheses:

Null hypothesis (H0): The new marketing campaign has no effect on the sales of the product.
Alternative hypothesis (Ha): The new marketing campaign has a positive effect on the sales of the product.
We can use a t-test to compare the means of the sales before and after the campaign. If the p-value is less than the significance level (usually set to 0.05), we reject the null hypothesis and conclude that the new campaign has had a significant positive effect on the sales of the product.

Probability Distribution Curve:

We can use probability distribution curves to visualize the distribution of the sales data before and after the campaign. We can use a histogram to plot the frequency of the sales data in different intervals. We can then fit a probability distribution curve to the histogram to see how well the data fit a particular distribution. For example, we can fit a normal distribution curve to the sales data and check if the data is normally distributed.

Skewness:

We can use skewness to measure the symmetry of the distribution of the sales data before and after the campaign. Skewness is a measure of the degree of asymmetry of the data around the mean. A positive skewness value indicates that the data is skewed to the right (has a longer

tail on the right side of the distribution), while a negative skewness value indicates that the data is skewed to the left (has a longer tail on the left side of the distribution).

If the sales data is significantly skewed, we may need to use non-parametric tests instead of t-tests. Non-parametric tests do not assume that the data is normally distributed and can be used to test hypotheses about non-normal distributions.

Overall, hypothesis testing, probability distribution curves, and skewness are all useful tools for analysing and interpreting data in a wide range of applications, including marketing campaigns, medical research, finance, and more.

TABLE OF CONTENTS