Basic Statistical Techniques

Hands-on-Session with R-Studio

Tanujit Chakraborty

PhD Scholar, Indian Statistical Institute, Kolkata.

Webpage : www.ctanujit.org

Mail : tanujitisi@gmail.com

Course Outline

Chapter	Topic	Chapter	Topic
1	Introduction to RStudio	6	Normality Tests
2	Descriptive Statistics	7	Analysis of Variance
3	Data Visualization	8	Regression Analysis
4	Data Pre-processing	9	Binary Logistic Regression
5	Test of Hypothesis	10	Ordinal Logistic Regression

1. INTRODUCTION TO RSTUDIO

INSTALLATION

1.Download R software from **http://cran.r-project.org/bin/windows/base/**

2.Run the R set up (exe) file and follow instructions

3.Double click on the R icon in the desktop and R window will open

4.Download RStudio from **http://www.rstudio.com/**

5.Run R studio set up file and follow instructions

6.Click on R studio icon, R Studio IDE Studio will load

7.Tools – Global Options – Appearances – Change Colour Size Theme

(if you wish to change the background, not a mandatory step)

4.Go to R-Script (Ctrl + Shift + N)

5. Write "Hello World !"

6.Save & Run (Ctrl + Enter)

Congrats ! You have written your very first R-Program

BASIC TASKS

Matrix multiplication – Code

Read the matrix A and B

```
A = matrix(c(21,57,89,31,7,98), nrow =2, ncol=3, byrow = TRUE)
B = matrix(c(24, 35, 15, 34, 56,25), nrow = 3, ncol = 2, byrow = TRUE)
```

Multiplication of matrices

```
C = A%*%B
C
```

Determinant – R Code

```
A = matrix(c(51, 10, 23, 64), nrow = 2, ncol =2, byrow =TRUE)
det(A)
```

Matrix Inverse – R code

```
A = matrix(c(51, 10, 23, 64), nrow = 2, ncol =2, byrow =TRUE)
solve(A)
```

BASIC TASKS

Eigen values and Eigen vectors – R Code

```
A = matrix(c(1, -2, 3, -4), nrow = 2, ncol = 2, byrow = TRUE)
eigen(A)
```

Generating 5 Random Numbers – R Code

```
x = rnorm(5, mean = 0, sd = 1)
x
```

Functional Help

```
?rnorm()
```

Package Installation

```
install.packages("ggplot")
```

Library Call (for use)

```
library(ggplot)
```

2. DESCRIPTIVE STATISTICS

DESCRIPTIVE STATISTICS

Exercise 1: The monthly credit card expenses of an individual in 1000 rupees is given in the file Credit_Card_Expenses.csv.

a.Read the dataset to R studio

b.Compute mean, median minimum, maximum, range, variance, standard deviation, skewness, kurtosis and quantiles of Credit Card Expenses

c.Compute default summary of Credit Card Expenses

d.Draw Histogram of Credit Card Expenses

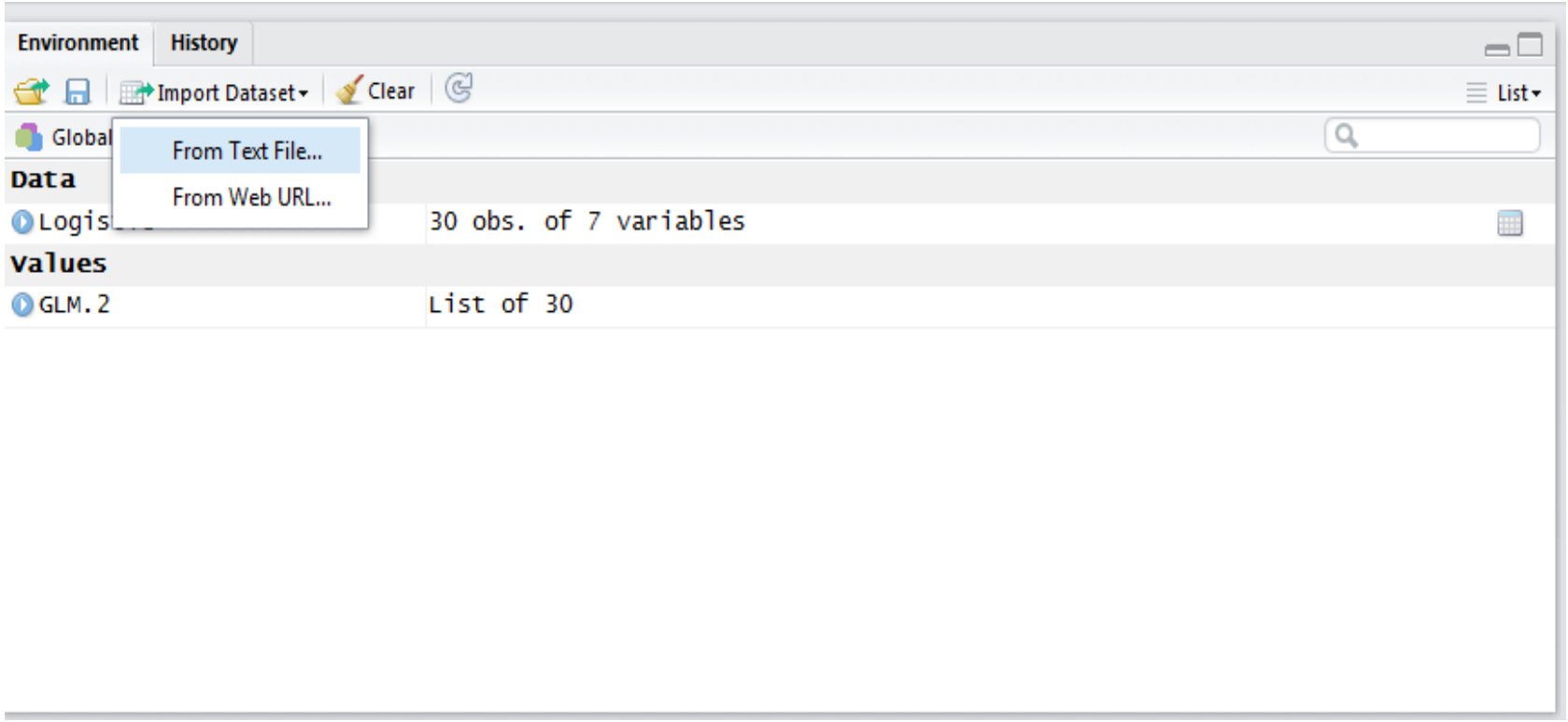
DESCRIPTIVE STATISTICS

The monthly credit card expenses of an individual in 1000 rupees is given below. Kindly summarize the data

Month	Credit Card Expenses	Month	Credit Card Expenses
1	55	11	63
2	65	12	55
3	59	13	61
4	59	14	61
5	57	15	57
6	61	16	59
7	53	17	61
8	63	18	57
9	59	19	59
10	57	20	63

DESCRIPTIVE STATISTICS

Reading a csv file to R Studio

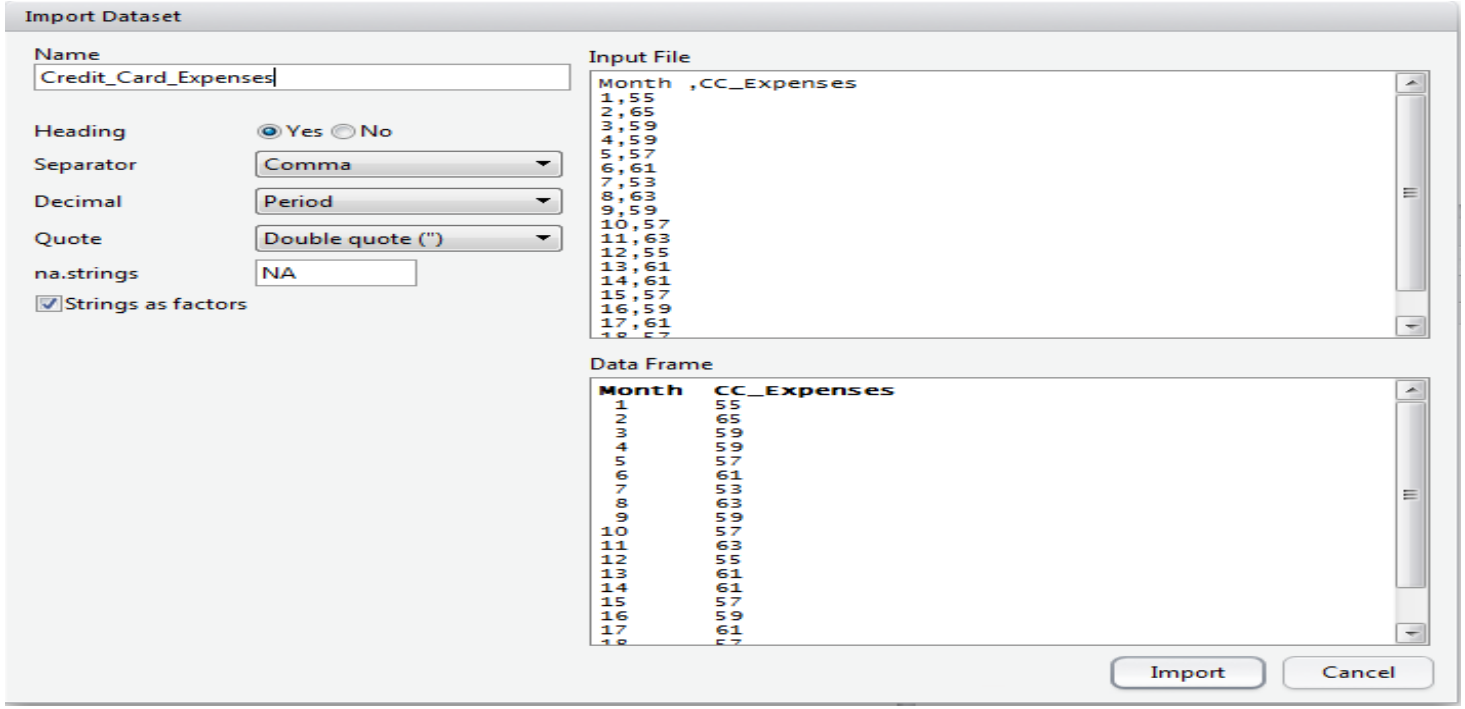


The file open dialog box will pop up

Browse to the file

DESCRIPTIVE STATISTICS

Reading a csv file to R Studio



Click Import button

R studio will read the data set to a data frame with specified name

DESCRIPTIVE STATISTICS

Reading a csv file to R Studio : Source code

➢Credit_Card_Expenses <- read.csv(“C:/Desktop/Data/Credit_Card_Expenses.csv")

To change the name of the data set to : mydata

> mydata = Credit_Card_Expenses

To display the contents of the data set

> print(mydata)

To read a particular column or variable of data set to a ne variable Example: Read CC_Expenses to CC

>CC = mydata$CC_Expenses

DESCRIPTIVE STATISTICS

Reading data from MS Excel formats to R Studio

Format	Code
Excel	library(xlsx) mydata <- read.xlsx("c:/myexcel.xlsx", “Sheet1”)

Reading data from databases to R Studio

Function	Description
odbcConnect(*dsn*, uid="", pwd="")	Open a connection to an ODBC database
sqlFetch(*channel*, *sqtable*)	Read a table from an ODBC database into a data frame
sqlQuery(*channel*, *query*)	Submit a query to an ODBC database and return the results
sqlSave(*channel*, *mydf*, tablename = *sqtable*, append = *FALSE*)	Write or update (append=True) a data frame to a table in the ODBC database
sqlDrop(*channel*, *sqtable*)	Remove a table from the ODBC database
close(*channel*)	Close the connection

DESCRIPTIVE STATISTICS

Operators - Arithmetic

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/% y	integer division 5%/%2

DESCRIPTIVE STATISTICS

Operators - Logical

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
! =	not equal to
!x	Not x
x | y	x OR y
x & y	x AND y
isTRUE(x)	test if X is TRUE

DESCRIPTIVE STATISTICS

Descriptive Statistics

Computation of descriptive statistics for variable CC

Function	Code	Value
Mean	> mean(CC)	59.2
Median	> median(CC)	59
Standard deviation	> sd(CC)	3.105174
Variance	> var(CC)	9.642105
Minimum	> min(CC)	53
Maximum	> max(CC)	65
Range	> range(CC)	53 65

DESCRIPTIVE STATISTICS

Descriptive Statistics

Function	Code
Quantile	> quantile(CC)

Output					
Quantile	0%	25%	50%	75%	100%
Value	53	57	59	61	65

Function	Code
Summary	>summary(CC)

Output					
Minimum	Q1	Median	Mean	Q3	Maximum
53	57	59	59.2	61	65

DESCRIPTIVE STATISTICS

Descriptive Statistics

Function	Code
describe	> library(psych) > describe(CC)

Output	
Statistics	Values
N	20
mean	59.2
sd	3.11
median	59
Trimmed	59.25
mad	2.97
min	53
Max	65
Range	12
Skew	-0.08
Kurtosis	-0.85
se	0.69

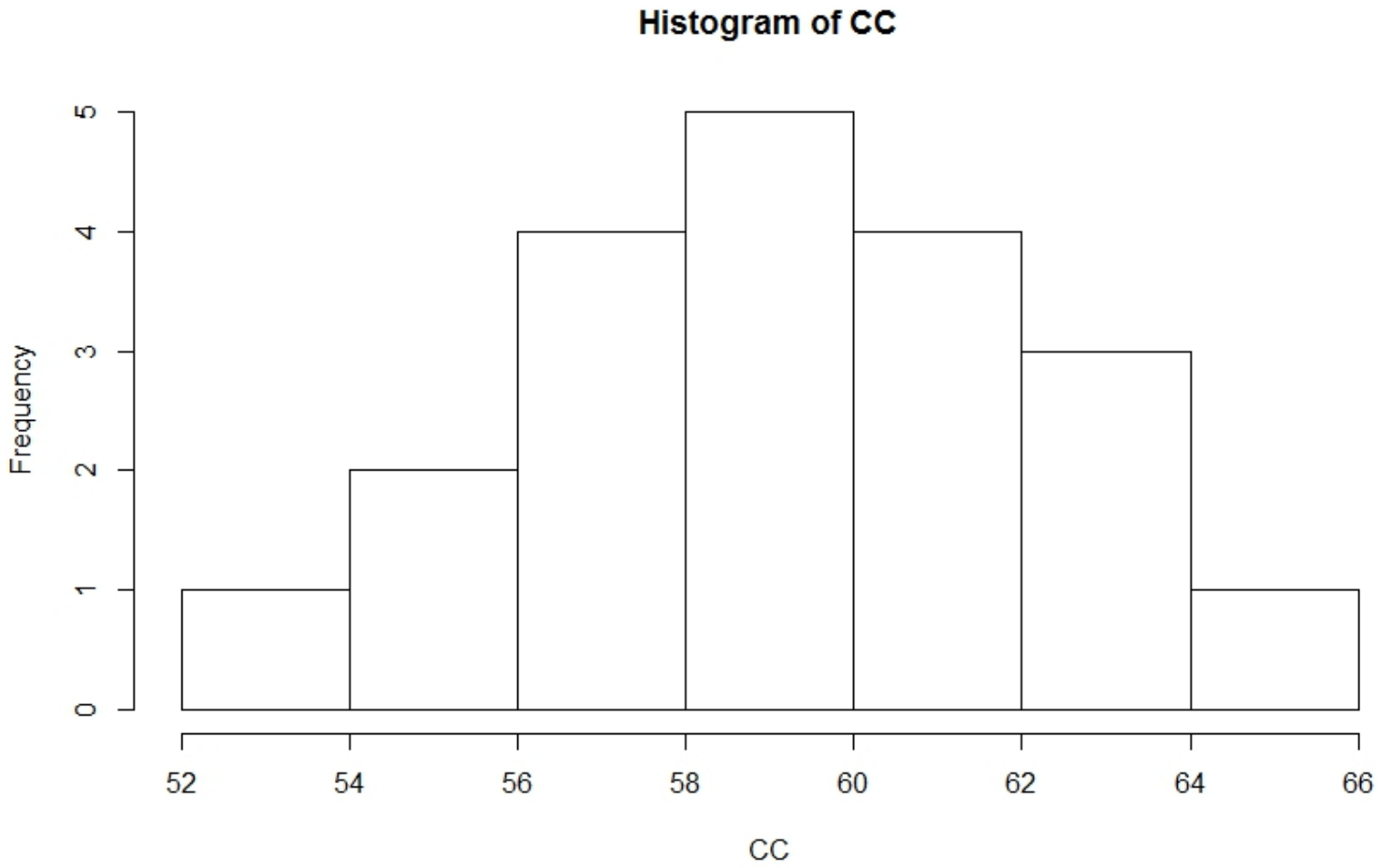
DESCRIPTIVE STATISTICS

Graphs

Graph	Code
Histogram	> hist(CC)
Histogram colour (“Blue”)	> hist(CC,col="blue")
Dot plot	> dotchart(CC)
Box plot	> boxplot(CC)
Box plot colour	> boxplot(CC, col="dark green")

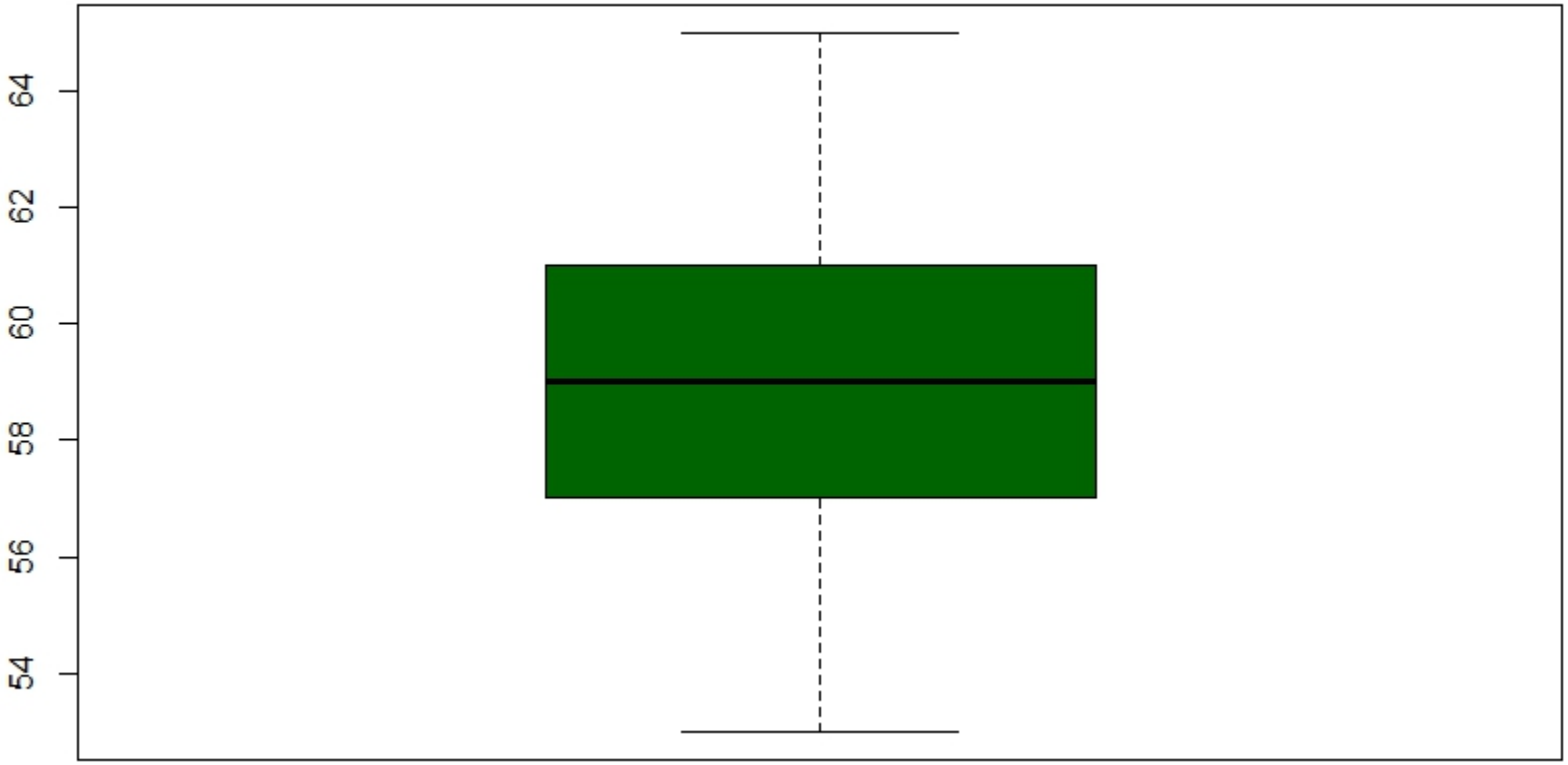
DESCRIPTIVE STATISTICS

Histogram : Variable - CC



DESCRIPTIVE STATISTICS

Box plot : Variable - CC



3. DATA VISUALIZATION

DATA VISUALIZATION

With ever increasing volume of data, it is impossible to tell stories without visualizations. Data visualization is an art of how to turn numbers into useful knowledge.

Popular Data Visualization Techniques:

1. Scatter Plot
2. Histogram
3. Bar & Stack Bar Chart
4. Box Plot
5. Area Chart
6. HeatMap
7. Correlogram

We'll use 'Big_Mart_Dataset.csv' example as shown below to understand how to create visualizations.

Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type
FDA15	9.300	Low Fat	0.016047301	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Typ
DRC01	5.920	Regular	0.019278216	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Typ
FDN15	17.500	Low Fat	0.016760075	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Typ
FDX07	19.200	Regular	0.000000000	Fruits and Vegetables	182.0950	OUT010	1998		Tier 3	Grocery Store
NCD19	8.930	Low Fat	0.000000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Typ
FDP36	10.395	Regular	0.000000000	Baking Goods	51.4008	OUT018	2009	Medium	Tier 3	Supermarket Typ
FDO10	13.650	Regular	0.012741089	Snack Foods	57.6588	OUT013	1987	High	Tier 3	Supermarket Typ
FDP10	NA	Low Fat	0.127469857	Snack Foods	107.7622	OUT027	1985	Medium	Tier 3	Supermarket Typ
FDH17	16.200	Regular	0.016687114	Frozen Foods	96.9726	OUT045	2002		Tier 2	Supermarket Typ
FDU28	19.200	Regular	0.094449590	Frozen Foods	187.8214	OUT017	2007		Tier 2	Supermarket Typ
FDY07	11.800	Low Fat	0.000000000	Fruits and Vegetables	45.5402	OUT049	1999	Medium	Tier 1	Supermarket Typ
FDA03	18.500	Regular	0.045463773	Dairy	144.1102	OUT046	1997	Small	Tier 1	Supermarket Typ
FDX32	15.100	Regular	0.100013500	Fruits and Vegetables	145.4786	OUT049	1999	Medium	Tier 1	Supermarket Typ
FDS46	17.600	Regular	0.047257328	Snack Foods	119.6782	OUT046	1997	Small	Tier 1	Supermarket Typ
FDF32	16.350	Low Fat	0.068024300	Fruits and Vegetables	196.4426	OUT013	1987	High	Tier 3	Supermarket Typ

1. Scatter Plot: It is used to see the relationship between two continuous variables. In our above mart dataset, if we want to visualize the items as per their cost data, then we can use scatter plot chart using two continuous variables, namely Item_Visibility & Item_MRP as shown.

Read data and simple scatter plot using function ggplot() with geom_point().

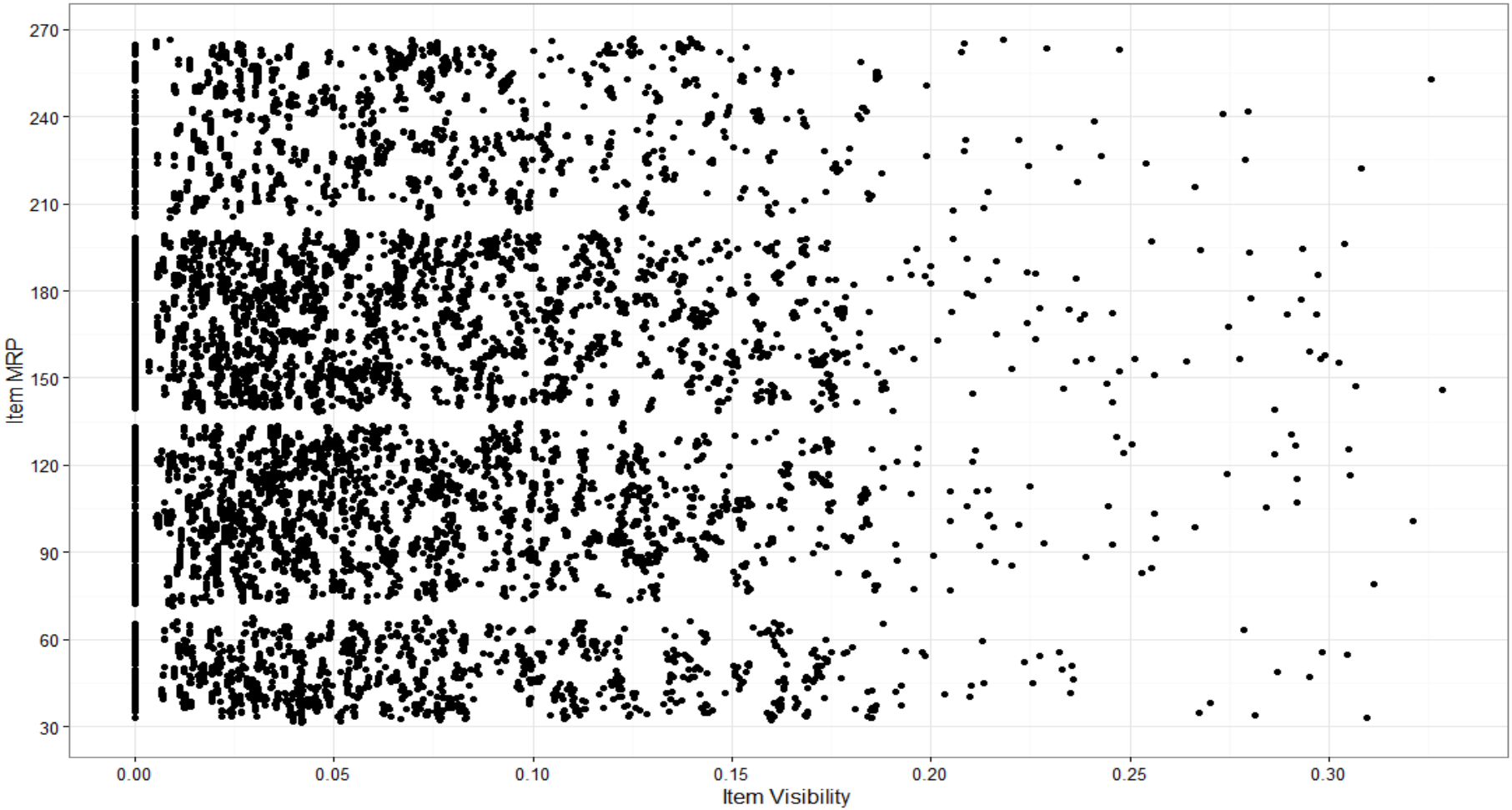
```
> train <-
read.csv("C:/Users/ISIUSER3/Desktop/CAIML_2019/Data/Big_Mart_Dataset.c
sv")

> view(train)

> library(ggplot2)

> ggplot(train, aes(Item_Visibility, Item_MRP)) + geom_point() +
scale_x_continuous("Item Visibility", breaks = seq(0,0.35,0.05))+
scale_y_continuous("Item MRP", breaks = seq(0,270,by = 30))+ theme_bw()
```

DATA VISUALIZATION

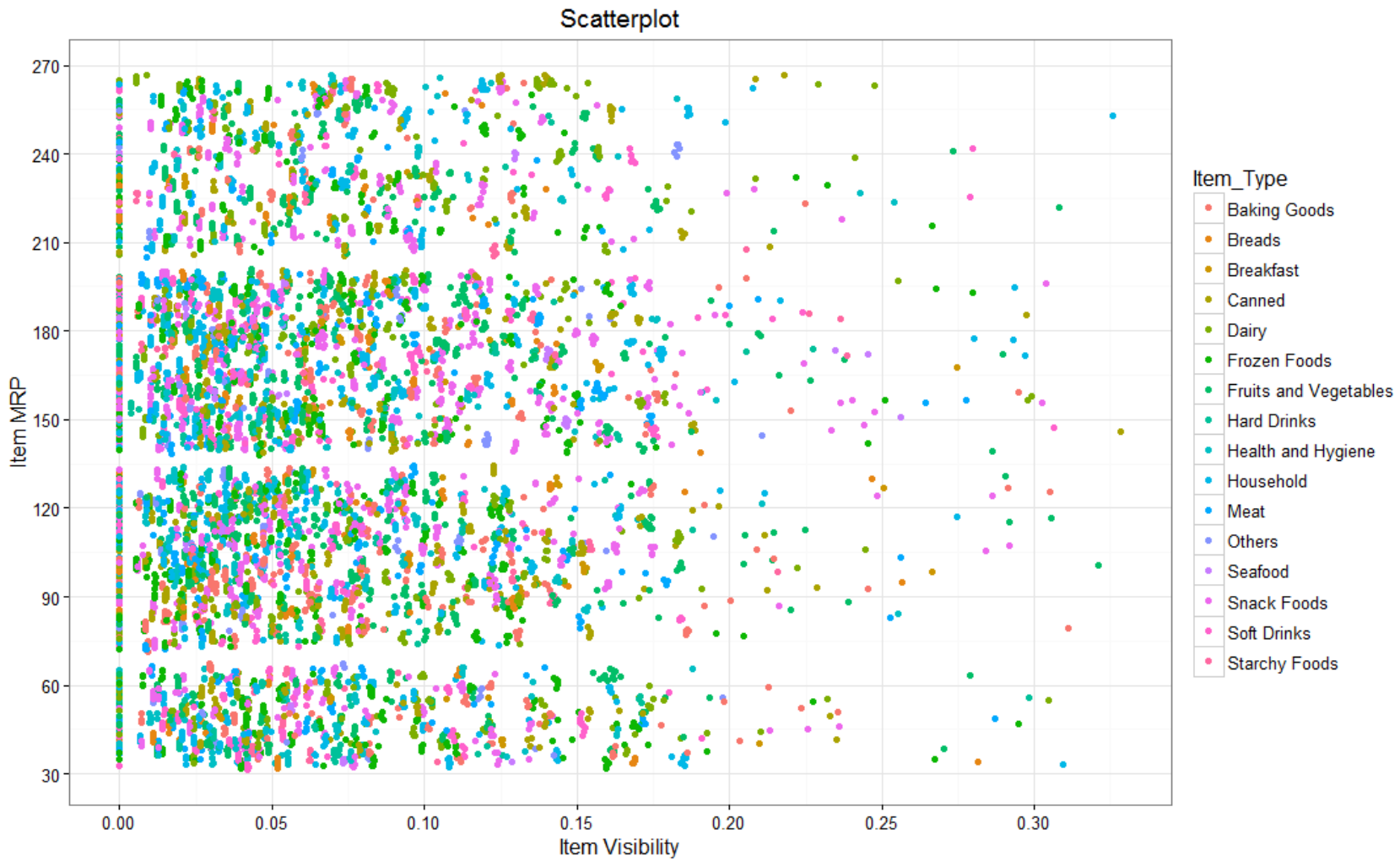


1. Scatter Plot: Now, we can view a third variable also in same chart, say a categorical variable (Item_Type) which will give the characteristic (item_type) of each data set. Different categories are depicted by way of different color for item_type in below chart.

Another scatter plot using function ggplot() with geom_point().

```
> library(ggplot2)

> ggplot(train, aes(Item_Visibility, Item_MRP)) + geom_point(aes(color = Item_Type)) + scale_x_continuous("Item Visibility", breaks = seq(0,0.35,0.05))+ scale_y_continuous("Item MRP", breaks = seq(0,270,by = 30))+ theme_bw() + labs(title="Scatterplot")
```


Scatterplot
Item MRP
270
240
210
180
150
120
90
60
30
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Item Visibility
Item_Type
Baking Goods
Breads
Breakfast
Canned
Dairy
Frozen Foods
Fruits and Vegetables
Hard Drinks
Health and Hygiene
Household
Meat
Others
Seafood
Snack Foods
Soft Drinks
Starchy Foods

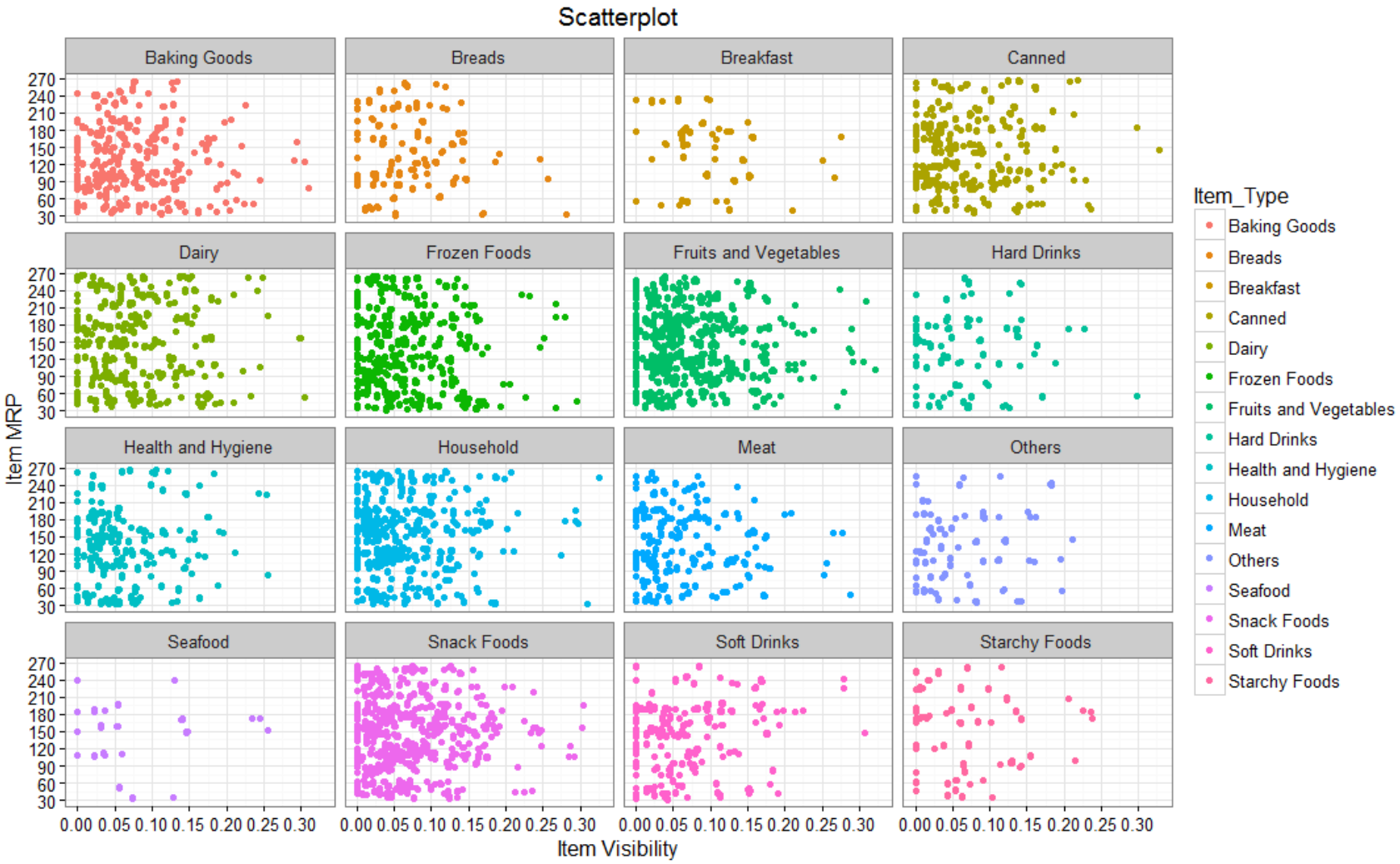
1. Scatter Plot: We can even make it more visually clear by creating separate scatter plots for each separate Item_Type as shown below.

Another scatter plot using function ggplot() with geom_point().

- library(ggplot2)

- ggplot(train, aes(Item_Visibility, Item_MRP)) + geom_point(aes(color = Item_Type)) + scale_x_continuous("Item Visibility", breaks = seq(0,0.35,0.05))+ scale_y_continuous("Item MRP", breaks = seq(0,270,by = 30))+ theme_bw() + labs(title="Scatterplot") + facet_wrap(~ Item_Type)

Here, facet_wrap works well & wraps Item_Type in rectangular layout.

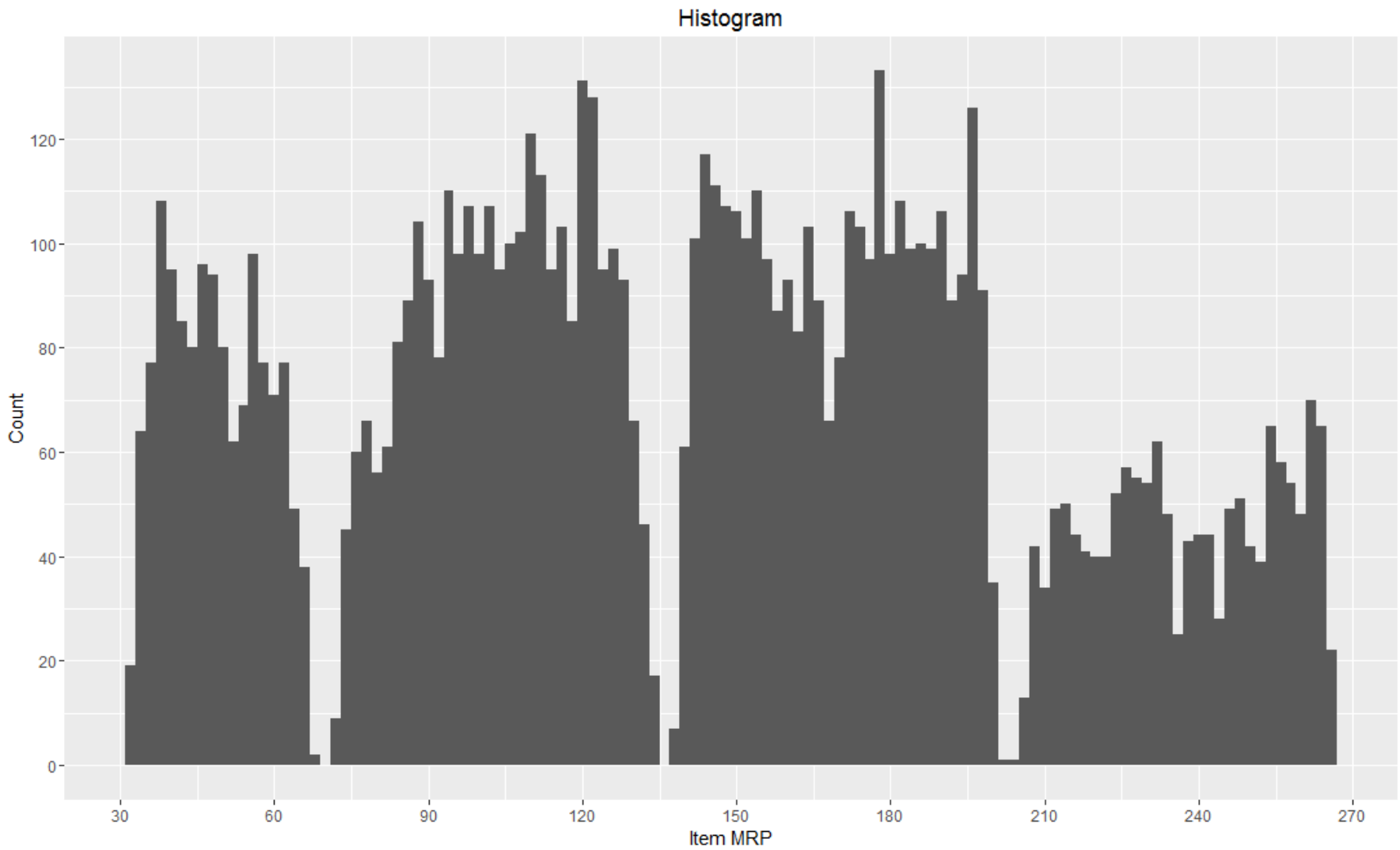

Scatterplot
Baking Goods
Breads
Breakfast
Canned
Dairy
Frozen Foods
Fruits and Vegetables
Hard Drinks
Health and Hygiene
Household
Meat
Others
Seafood
Snack Foods
Soft Drinks
Starchy Foods
Item MRP
270
240
210
180
150
120
90
60
30
0.00 0.05 0.10 0.15 0.20 0.25 0.30
Item Visibility
Item_Type
Baking Goods
Breads
Breakfast
Canned
Dairy
Frozen Foods
Fruits and Vegetables
Hard Drinks
Health and Hygiene
Household
Meat
Others
Seafood
Snack Foods
Soft Drinks
Starchy Foods

2. Histogram: It is used to plot continuous variable. It breaks the data into bins and shows frequency distribution of these bins. We can always change the bin size and see the effect it has on visualization.

For Big_Mart_Dataset, if we want to know the count of items on basis of their cost, then we can plot histogram using continuous variable Item_MRP as shown below.

Histogram plot using function ggplot() with geom_ histogram()

```
> ggplot(train, aes(Item_MRP)) + geom_histogram(binwidth = 2)+
scale_x_continuous("Item MRP", breaks = seq(0,270,by = 30))+
scale_y_continuous("Count", breaks = seq(0,200,by = 20))+  labs(title =
"Histogram")
```


Histogram
Count
0
20
40
60
80
100
120
30
60
90
120
150
180
210
240
270
Item MRP

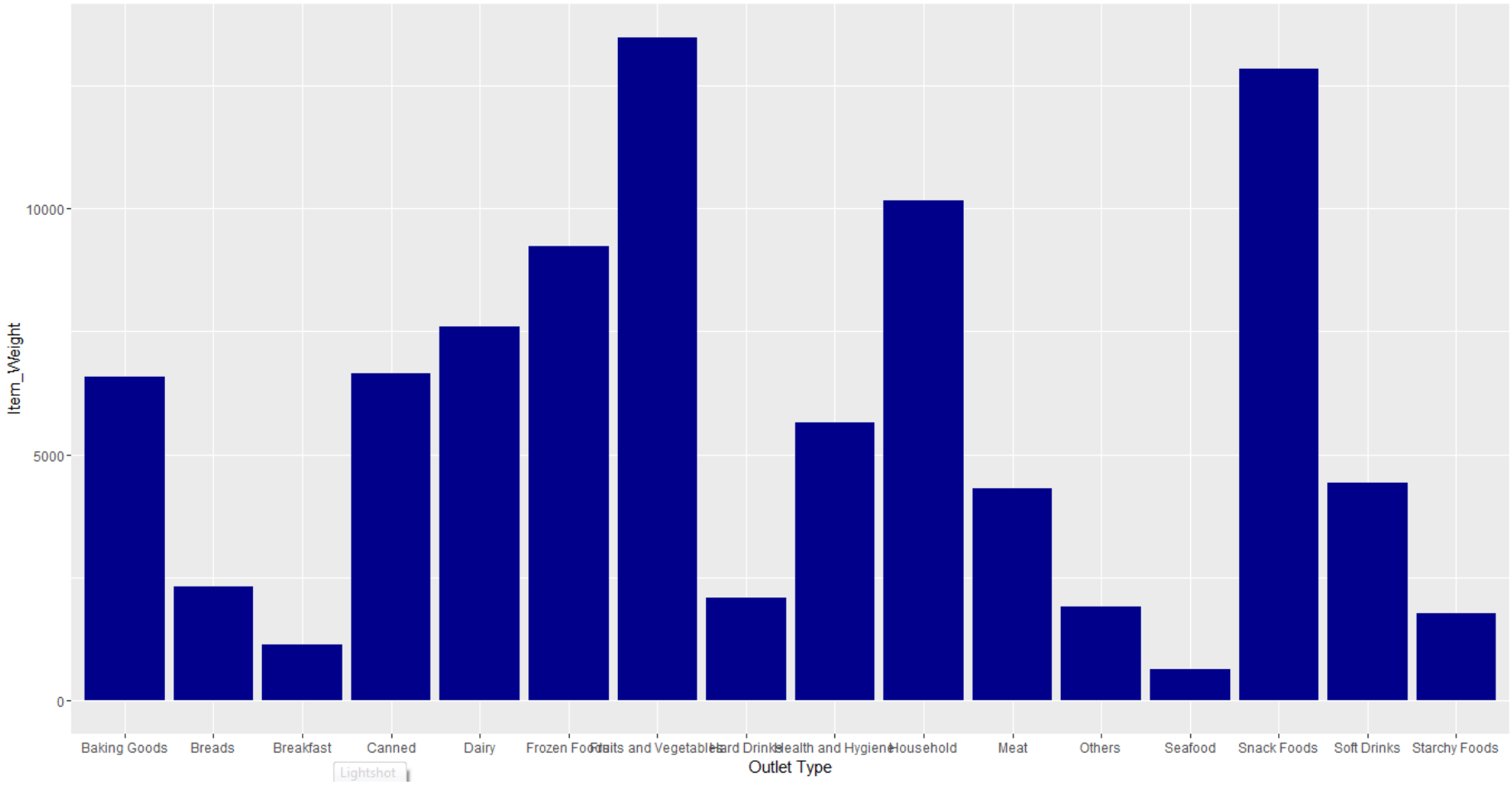
3. Bar Chart: It is used when you want to plot a categorical variable or a combination of continuous and categorical variable.

For Big_Mart_Dataset, if we want to know item weights (continuous variable) on basis of Outlet Type (categorical variable) on single bar chart as shown below.

Vertical Bar plot using function ggplot()

```
> ggplot(train, aes(Item_Type, Item_Weight)) + geom_bar(stat = "identity", fill = "darkblue") + scale_x_discrete("Outlet Type")+ scale_y_continuous("Item Weight", breaks = seq(0,15000, by = 500))+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) + labs(title = "Bar Chart")
```

DATA VISUALIZATION

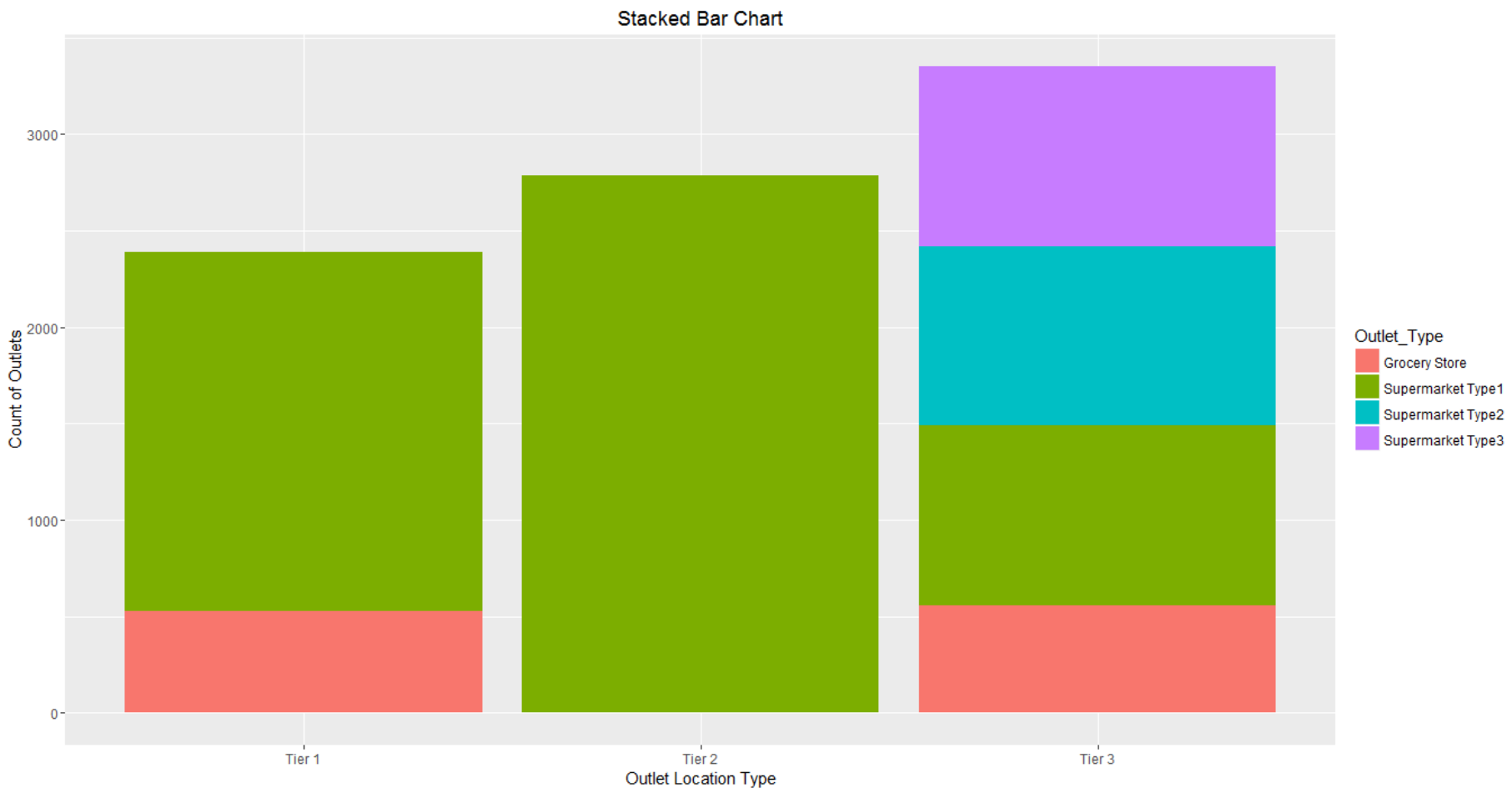


3. Stack Bar Chart: It is an advanced version of bar chart, used for visualizing a combination of categorical variables.

For Big_Mart_Dataset, if we want to know the count of outlets on basis of categorical variables like its type (Outlet Type) and location (Outlet Location Type) both, stack chart will visualize the scenario in most useful manner.

Stack Bar Chart using function ggplot()

```
> ggplot(train, aes(Outlet_Location_Type, fill = Outlet_Type)) +
geom_bar()+labs(title = "Stacked Bar Chart", x = "Outlet Location Type", y =
"Count of Outlets")
```

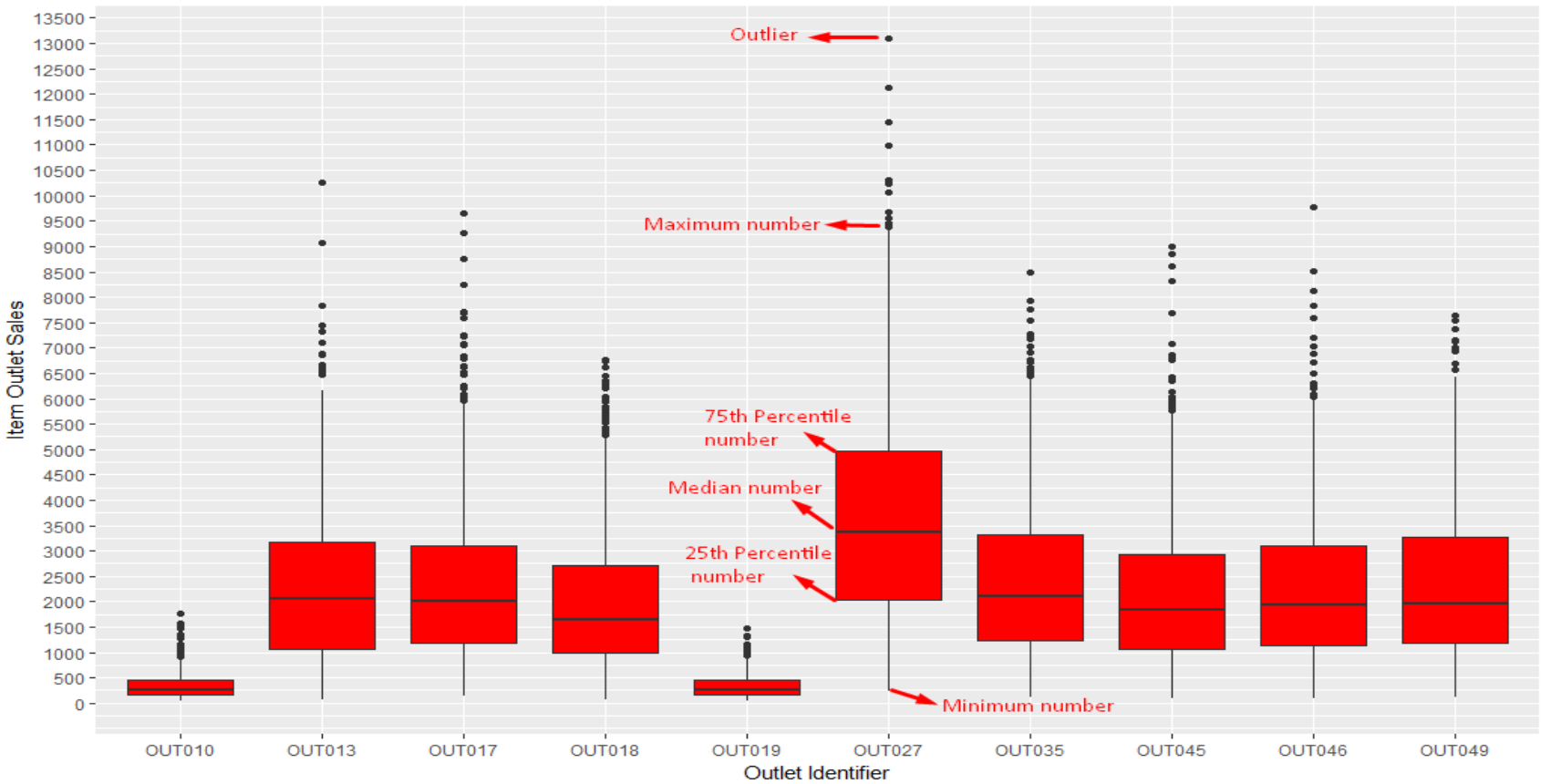

Stacked Bar Chart
Count of Outlets
3000
2000
1000
0
Tier 1
Tier 2
Tier 3
Outlet Location Type
Outlet_Type
Grocery Store
Supermarket Type1
Supermarket Type2
Supermarket Type3

4. Box Plot: It is used to plot a combination of categorical and continuous variables. This plot is useful for visualizing the spread of the data and detect outliers. It shows five statistically significant numbers- the minimum, the 25th percentile, the median, the 75th percentile and the maximum.

For Big_Mart_Dataset, if we want to identify each outlet's detailed item sales including minimum, maximum & median numbers, box plot can be helpful. In addition, it also gives values of outliers of item sales for each outlet as shown in below chart.

R Code:

```
> ggplot(train, aes(Outlet_Identifier, Item_Outlet_Sales)) + geom_boxplot(fill =
"red")+scale_y_continuous("Item Outlet Sales", breaks= seq(0,15000,
by=500))+labs(title = "Box Plot", x = "Outlet Identifier")
```



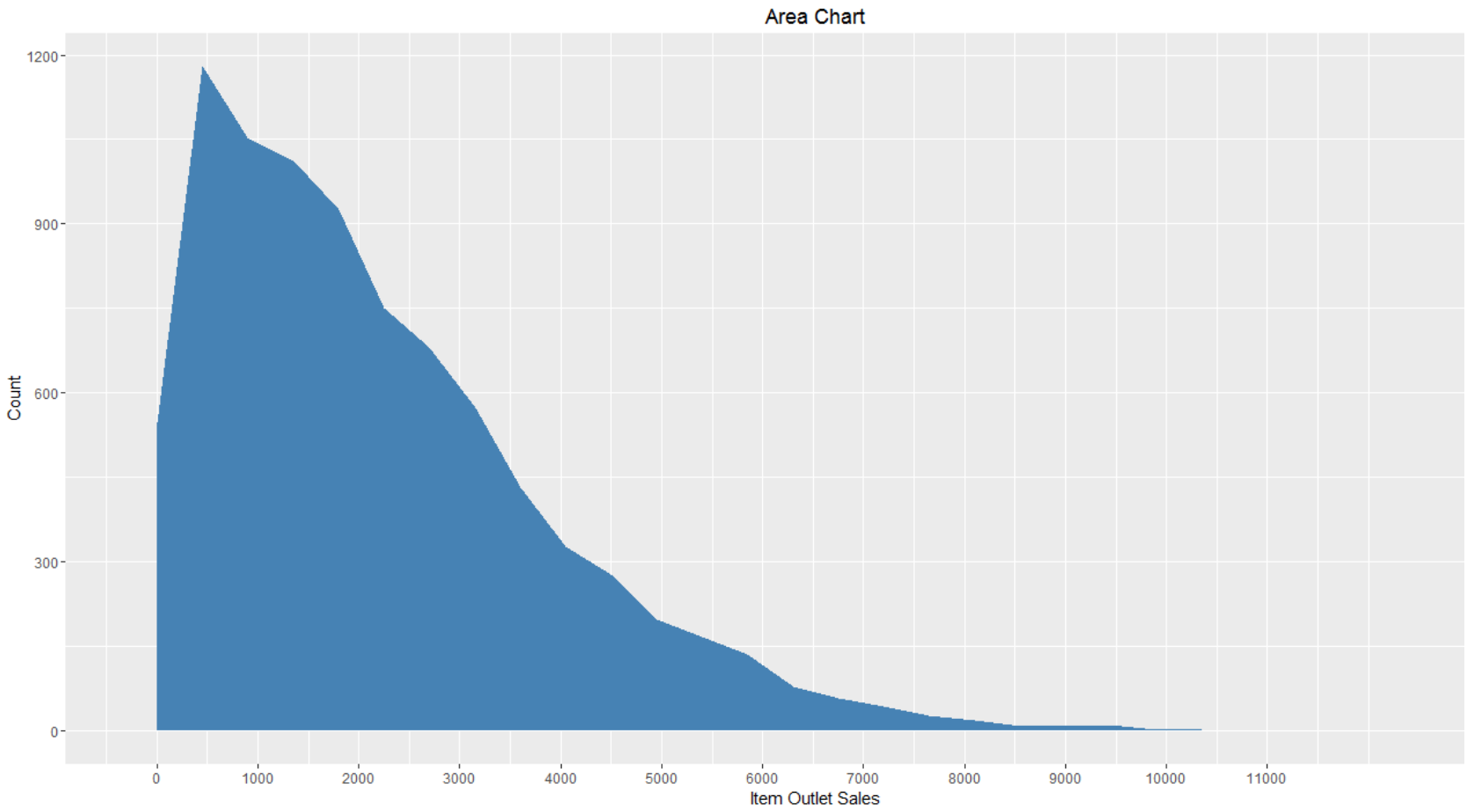
The black points are outliers. Outlier detection and removal is an essential step of successful data exploration.

5. Area Chart: It is used to show continuity across a variable or data set. It is very much same as line chart and is commonly used for time series plots. Alternatively, it is also used to plot continuous variables and analyse the underlying trends.

For Big_Mart_Dataset, when we want to analyse the trend of item outlet sales, area chart can be plotted as shown below. It shows count of outlets on basis of sales.

R Code:

```
> ggplot(train, aes(Item_Outlet_Sales)) + geom_area(stat = "bin", bins = 30, fill = "steelblue") + scale_x_continuous(breaks = seq(0,11000,1000))+ labs(title = "Area Chart", x = "Item Outlet Sales", y = "Count")
```



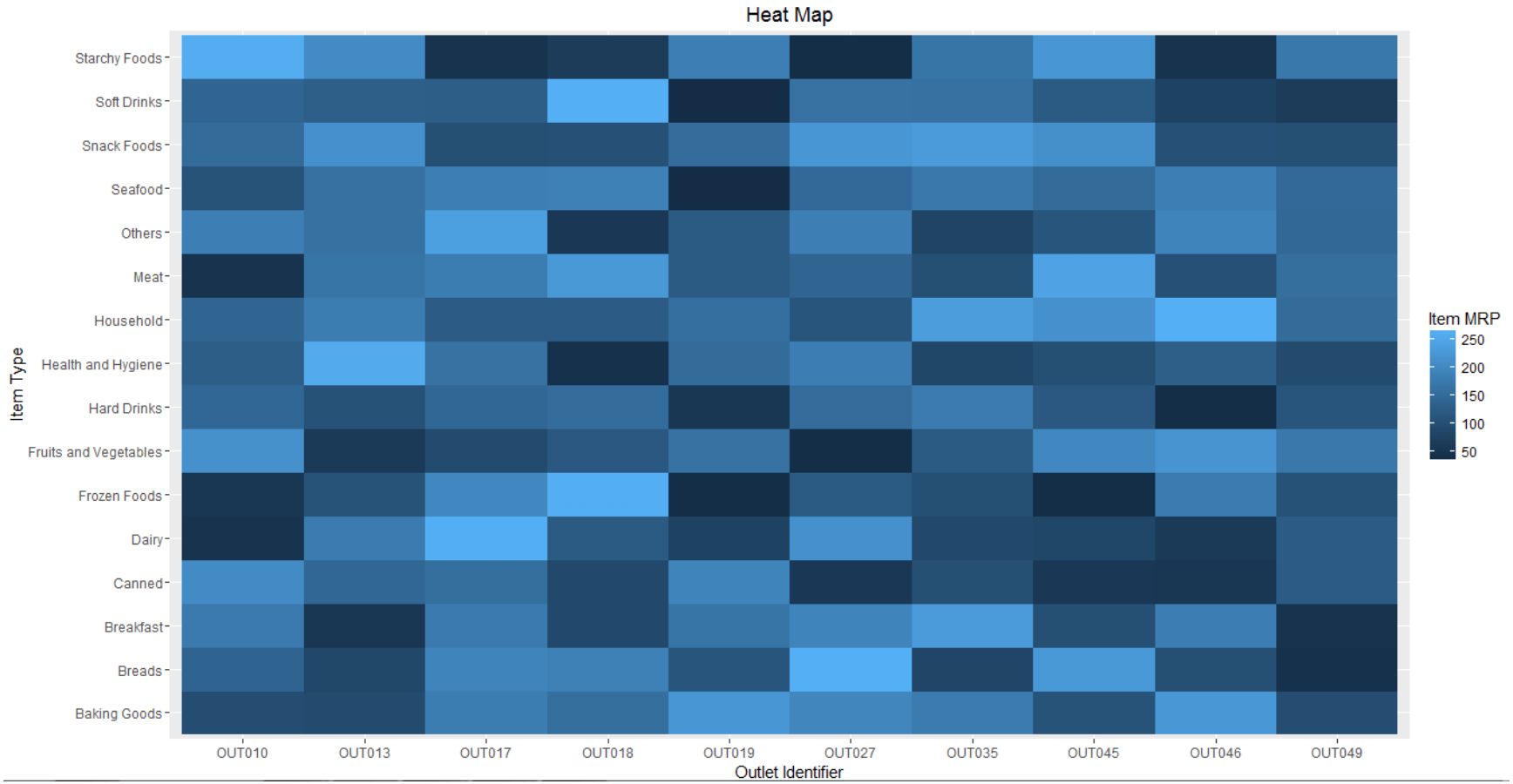
Area chart shows continuity of Item Outlet Sales using function ggplot() with geom_area.

6. Heat Map: It uses intensity (density) of colours to display relationship between two or three or many variables in a two dimensional image.

For Big_Mart_Dataset, if we want to know cost of each item on every outlet, we can plot heatmap as shown below using three variables Item MRP, Outlet Identifier & Item Type from our mart dataset.

R Code:

```
> ggplot(train, aes(Outlet_Identifier, Item_Type))+  geom_raster(aes(fill =
Item_MRP))+  labs(title ="Heat Map", x = "Outlet Identifier", y = "Item Type")+
scale_fill_continuous(name = "Item MRP")
```



The dark portion indicates Item MRP is close 50. The brighter portion indicates Item MRP is close to 250.

7. Correlogram: It is used to test the level of co-relation among the variable available in the data set. The cells of the matrix can be shaded or coloured to show the co-relation value.

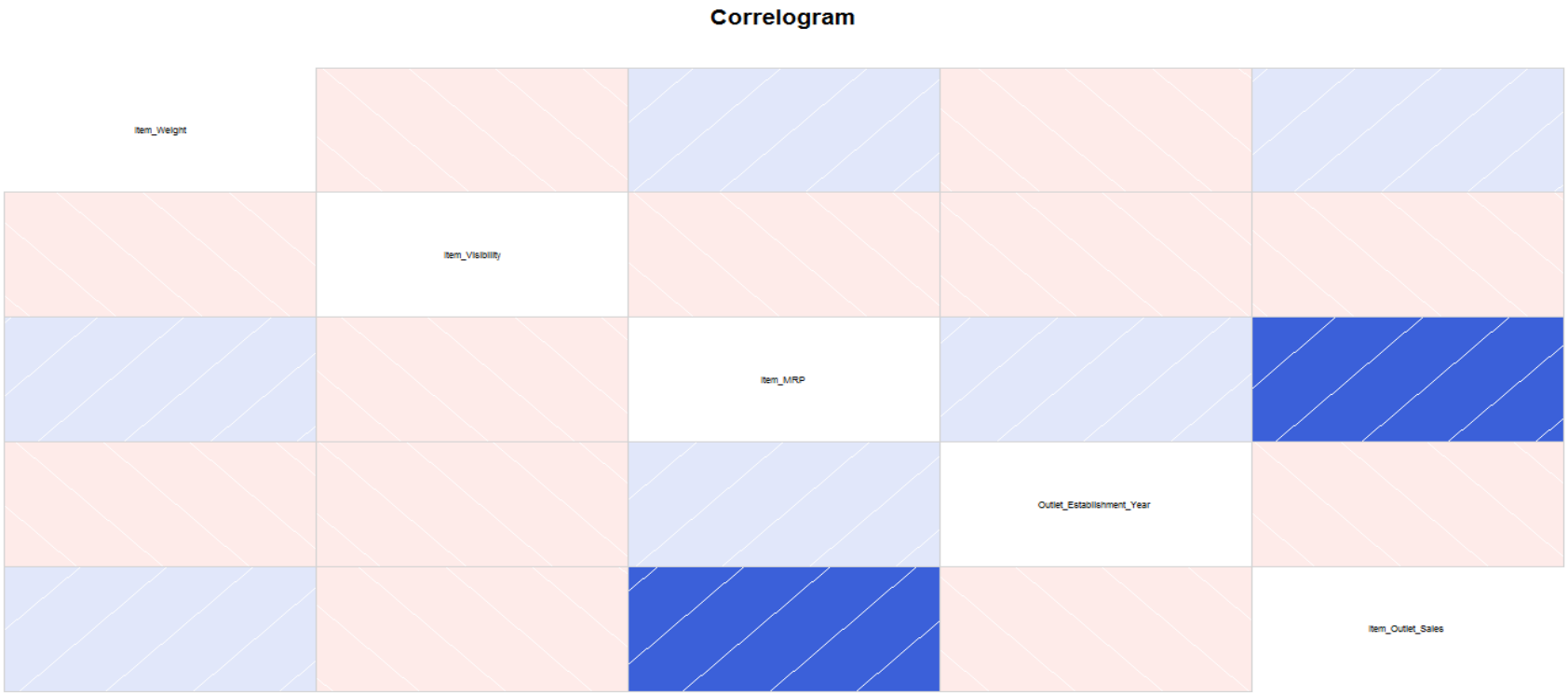
For Big_Mart_Dataset, check co-relation between Item cost, weight, visibility along with Outlet establishment year and Outlet sales from below plot.

R Code for simple correlogram using function corrgram():

```
> install.packages("corrgram")

> library(corrgram)

> corrgram(train, order=NULL, panel=panel.shade, text.panel=panel.txt,
main="Correlogram")
```



- Darker the colour, higher the co-relation between variables. Positive co-relations are displayed in blue and negative correlations in red colour. Colour intensity is proportional to the co-relation value.
- We can see that Item cost & Outlet sales are positively correlated while Item weight & its visibility are negatively correlated.

4. DATA PRE-PROCESSING

DATA PREPROCESSING

1. Missing value replenishment
2. Merging data files
3. Appending the data files
4. Transformation or normalization
5. Random Sampling

Missing Value Handling

Example: Suppose a telecom company wants to analyze the performance of its circles based on the following parameters

1. Current Month's Usage
2. Last 3 Month's Usage
3. Average Recharge
4. Projected Growth

The data set is given in next slide. (Missing_Values_Telecom Data)

Missing Value Handling

Example: Circle wise Data

SL No.	Current Month's Usage	Last 3 Month's Usage	Average Recharge	Projected Growth	Circle
1	5.1	3.5	99.4	99.2	A
2	4.9	3	98.6	99.2	A
3		3.2		99.2	A
4	4.6	3.1	98.5	9..2	A
5	5		98.4	99.2	A
6	5.4	3.9	98.3	99.4	A
7	7	3.2	95.3	98.4.	B
8	6.4	3.2	95.5	98.5	B
9	6.9	3.1	95.1	98.5	B
10		2.3	96	98.3	B
11	6.5	2.8	95.4	98.5	B
12	5.7		95.5	98.3	B
13	6.3	3.3		98.6	B
14	6.7	3.3	94.3	97.5	C
15	6.7	3	94.8	97.3	C
16	6.3	2.5	95	98.9	C
17		3	94.8	98	C
18	6.2	3.4	94.6	97.3	C
19	5.9	3	94.9	98.8	C

Missing Value Handling

Example: Read data and variables to R

```
> mydata = Missing_Values_Telecom
> cmusage = mydata[,2]
> l3musage = mydata[,3]
> avrecharge = mydata[,4]
```

Missing Value Handling

Option 1: Discard all records with missing values

```
>newdata = na.omit(mydata)

>write.csv(newdata,"E:/ISI/newdata.csv")
```

SL.No.	Current.Month.s.Usage	Last.3.Month.s.Usage	Average.Recharge	Projected.Growth	Circle
1	5.1	3.5	99.4	99.2	A
2	4.9	3	98.6	99.2	A
4	4.6	3.1	98.5	9..2	A
6	5.4	3.9	98.3	99.4	A
7	7	3.2	95.3	98.4.	B
8	6.4	3.2	95.5	98.5	B
9	6.9	3.1	95.1	98.5	B
11	6.5	2.8	95.4	98.5	B
14	6.7	3.3	94.3	97.5	C
15	6.7	3	94.8	97.3	C
16	6.3	2.5	95	98.9	C
18	6.2	3.4	94.6	97.3	C
19	5.9	3	94.9	98.8	C

Missing Value Handling

Option 2: Replace the missing values with variable mean, median, etc

Replacing the missing values with mean

Compute the means excluding ghe missing values

```
>cmusage_mean = mean(cmusage, na.rm = TRUE)
>l3musage_mean = mean(l3musage_mean, na.rm = TRUE)
> avrecharge_mean = mean(avrecharge, na.rm = TRUE)
```

Replace the missing values with mean

```
> cmusage[is.na(cmusage)]=cmusage_mean
> l3musage[is.na(l3musage)]= l3musage_mean
>avrecharge[is.na(avrecharge)]=avrecharge_mean
```

Missing Value Handling

Option 2: Replace the missing values with variable mean, median, etc

Replacing the missing values with mean

Replace the missing values with mean

```
> cmusage[is.na(cmusage)]=cmusage_mean
> l3musage[is.na(l3musage)]= l3musage_mean
>avrecharge[is.na(avrecharge)]=avrecharge_mean
```

Making the new file

```
> mynewdata = cbind(cmusage, l3musage, avrecharge, mydata[,5],mydata[,6])
> write.csv(mynewdata, "E:/ISI/mynewdata.csv")
```

Missing Value Handling

Option 2: Replace the missing values with variable mean, median, etc

Replacing the missing values with men

SL No	cmusage	l3musage	avrecharge	Proj Growth	Circle
1	5.1	3.5	99.4	11	1
2	4.9	3	98.6	11	1
3	5.975	3.2	96.14117647	11	1
4	4.6	3.1	98.5	1	1
5	5	3.105882353	98.4	11	1
6	5.4	3.9	98.3	12	1
7	7	3.2	95.3	6	2
8	6.4	3.2	95.5	7	2
9	6.9	3.1	95.1	7	2
10	5.975	2.3	96	5	2
11	6.5	2.8	95.4	7	2
12	5.7	3.105882353	95.5	5	2
13	6.3	3.3	96.14117647	8	2
14	6.7	3.3	94.3	3	3
15	6.7	3	94.8	2	3
16	6.3	2.5	95	10	3
17	5.975	3	94.8	4	3
18	6.2	3.4	94.6	2	3
19	5.9	3	94.9	9	3

TRANSFORMATION / NORMALIZATION

z transform:

Transformed data = (Data – Mean) / SD

Exercise : Normalize the variables in the Supply_Chain.csv ?

Read the files

```
>mydata  = Supply_Chain
> mydata = mydata[,2:7]
```

Normalize or standardize the variable

```
>mystddata = scale(mydata)
```

RANDOM SAMPLING

Example: Take a sample of size 60 (10%) randomly from the data given in the file bank-data.csv and save it as a new csv file?

Read the files

```
>mydata  = bank-data

> mysample = mydata[sample(1:nrow(mydata), 60, replace = FALSE),]

>write.csv(mysample,"E:/ISI/mysample.csv")
```

RANDOM SAMPLING

Example: Split randomly the data given in the file bank-data.csv into sets namely training (75%) and test (25%) ?

Read the files

```
>mydata  = bank-data

>sample = sample(2, nrow(mydata), replace = TRUE, prob = c(0.75, 0.25))
> sample1 = mydata[sample ==1, ]
> sample2 = mydata[sample ==2,]
```

5. TEST OF HYPOTHESIS

TEST OF HYPOTHESIS

Introduction:

In many situations, it is required to accept or reject a statement or claim about some parameter

Example:

1. The average cycle time is less than 24 hours
2. The % rejection is only 1%

The statement is called the hypothesis

The procedure for decision making about the hypothesis is called hypothesis testing

Advantages

1. Handles uncertainty in decision making
2. Minimizes subjectivity in decision making
3. Helps to validate assumptions or verify conclusions

TEST OF HYPOTHESIS

Commonly used hypothesis tests on mean of normal distribution:

- Checking mean equal to a specified value ($mu = mu_0$)
- Two means are equal or not ($mu_1 = mu_2$)

TEST OF HYPOTHESIS

Null Hypothesis:

A statement about the status quo

One of no difference or no effect

Denoted by H0

Alternative Hypothesis:

One in which some difference or effect is expected

Denoted by H1

TEST OF HYPOTHESIS

Methodology demo: To Test Mean = Specified Value (mu = mu_0)

Suppose we want to test whether mean of a process characteristic is 5 based on the following sample data from the process

4	4	5	5	6
5	4.5	6.5	6	5.5

Calculate the mean of the sample, xbar = 5.15

Compare xbar with specified value 5

or xbar - specified value = xbar - 5 with 0

If xbar - 5 is close to 0

then conclude mean = 5

else mean $\neq$ 5

TEST OF HYPOTHESIS

Methodology demo : To Test Mean = Specified Value (mu = mu_0)

Consider another set of sample data. Check whether mean of the process characteristic is 500

400	400	500	500	600
500	450	650	600	550

Mean of the sample, xbar = 515

xbar - 500 = 515 - 500 = 15

Can we conclude mean $\neq$ 500?

Conclusion:

Difficult to say mean = specified value by looking at xbar - specified value alone

TEST OF HYPOTHESIS

Methodology demo: To Test Mean = Specified Value (mu = mu_0)

Test statistic is calculated by dividing (xbar - specified value) by a function of standard deviation

To test Mean = Specified value

Test Statistic t_0 = (xbar - Specified value) / (SD / $\sqrt{n}$)

If test statistic is close to 0, conclude that Mean = Specified value

To check whether test statistic is close to 0, find out p value from the sampling distribution of test statistic

TEST OF HYPOTHESIS

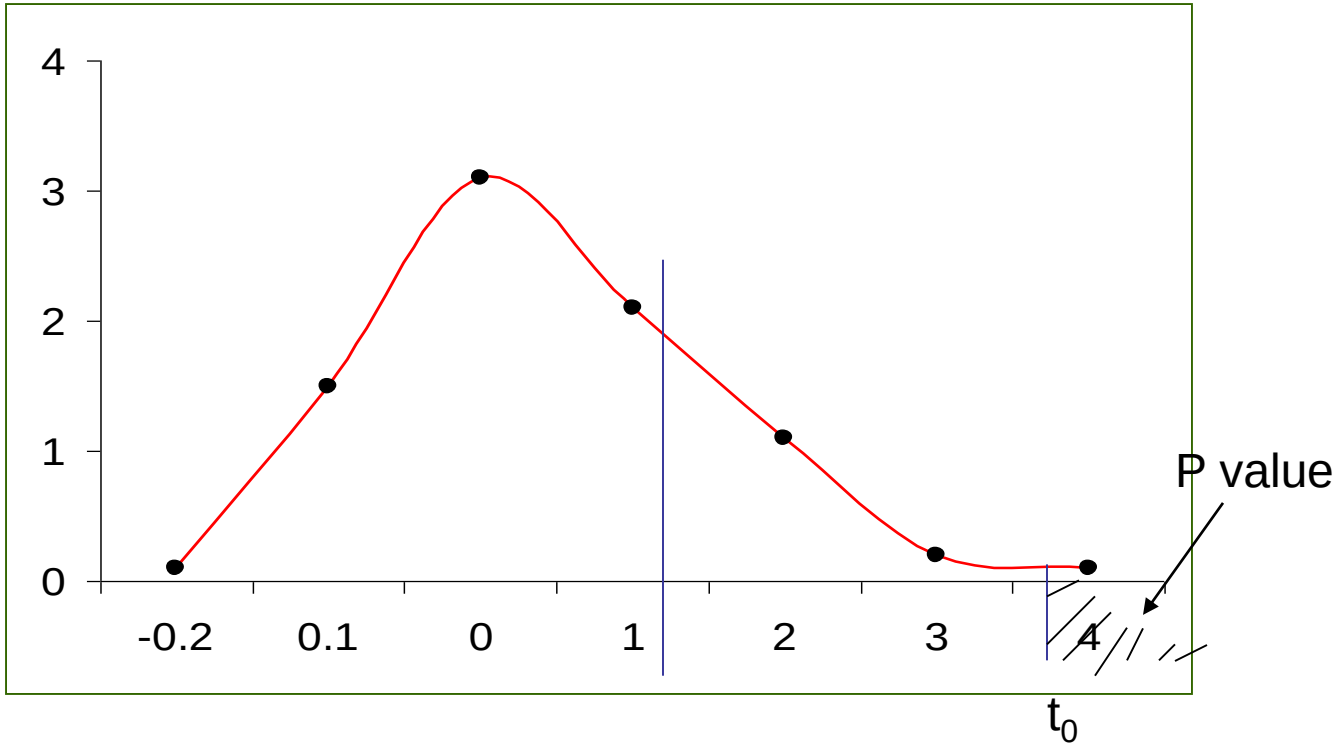
Methodology demo: To Test Mean = Specified Value

P value

The probability that such evidence or result will occur when H0 is true

Based on the reference distribution of test statistic

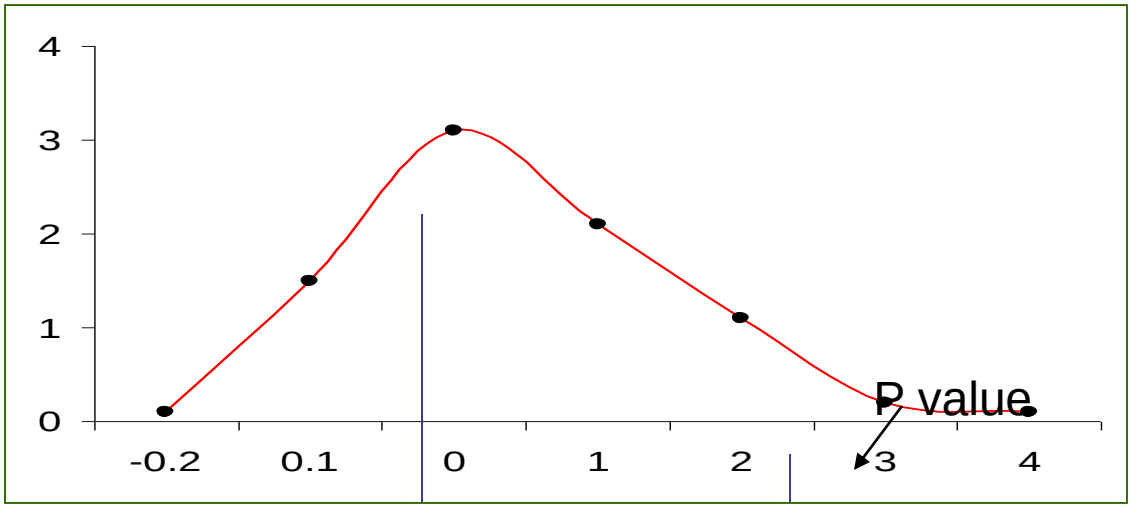
The tail area beyond the value of test statistic in reference distribution



TEST OF HYPOTHESIS

Methodology demo : To Test Mean = Specified Value

P value



t0

If test statistic t_0 is close to 0 then p will be high

If test statistic t_0 is not close to 0 then p will be small

If p is small , $p < 0.05$ (with alpha = 0.05), conclude that $t \neq 0$, then

Mean $\neq$ Specified Value, H0 rejected

TEST OF HYPOTHESIS

To Test Mean = Specified Value (mu = mu_0)

Example: Suppose we want to test whether mean of the process characteristic is 5 based on the following sample data

4	4	5	5	6
5	4.5	6.5	6	5.5

H0: Mean = 5

H1: Mean $\neq$ 5

Calculate xbar = 5.15

SD = 0.8515

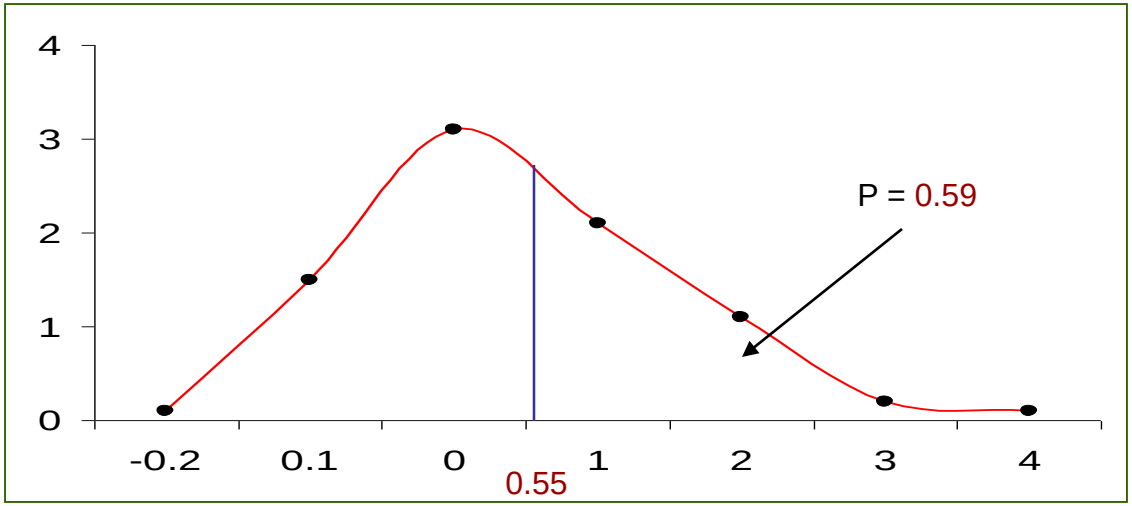
n = 10

Test statistic t_0 = (xbar - 5)/(SD / $\sqrt{n}$) = (5.15 - 5) / (0.8515 / $\sqrt{10}$) = 0.5571

TEST OF HYPOTHESIS

Example: To Test Mean = Specified Value (mu = mu_0)

$t_0 = 0.5571$



P ≥ 0.05, hence Mean = Specified value = 5.

H0: Mean = 5 is not rejected

TEST OF HYPOTHESIS

Hypothesis Testing: Steps

1. Formulate the null hypothesis H0 and the alternative hypothesis H1
2. Select an appropriate statistical test and the corresponding test statistic
3. Choose level of significance alpha (generally taken as 0.05)
4. Collect data and calculate the value of test statistic
5. Determine the probability associated with the test statistic under the null hypothesis using sampling distribution of the test statistic
6. Compare the probability associated with the test statistic with level of significance specified

TEST OF HYPOTHESIS

One sample t test

Exercise 1 : A company claims that on an average it takes only 40 hours or less to process any purchase order. Based on the data given below, can you validate the claim? The data is given in PO_Processing.csv

Reading data to mydata

```
> mydata = PO_Processing$Processing_Time
```

Performing one sample t test

```
> t.test(mydata, alternative = 'greater', mu = 40)
```

Statistics	Value
t	3.7031
df	99
P value	0.0001753

6. NORMALITY TEST

NORMALITY TEST

Normality test

A methodology to check whether the characteristic under study is normally distributed or not

Two Methods :

Normality test - Quantile – Quantile (Q- Q) plot

Plots the ranked samples from the given distribution against a similar number of ranked quantiles taken from a normal distribution

If the sample is normally distributed then the line will be straight in the plot

Normality test – Shapiro – Wilk test

H0: Deviation from bell shape (normality) = 0

H1 : Deviation from bell shape $\neq$ 0

If p value $\geq$ 0.05 (5%), then H0 is not rejected, distribution is normal

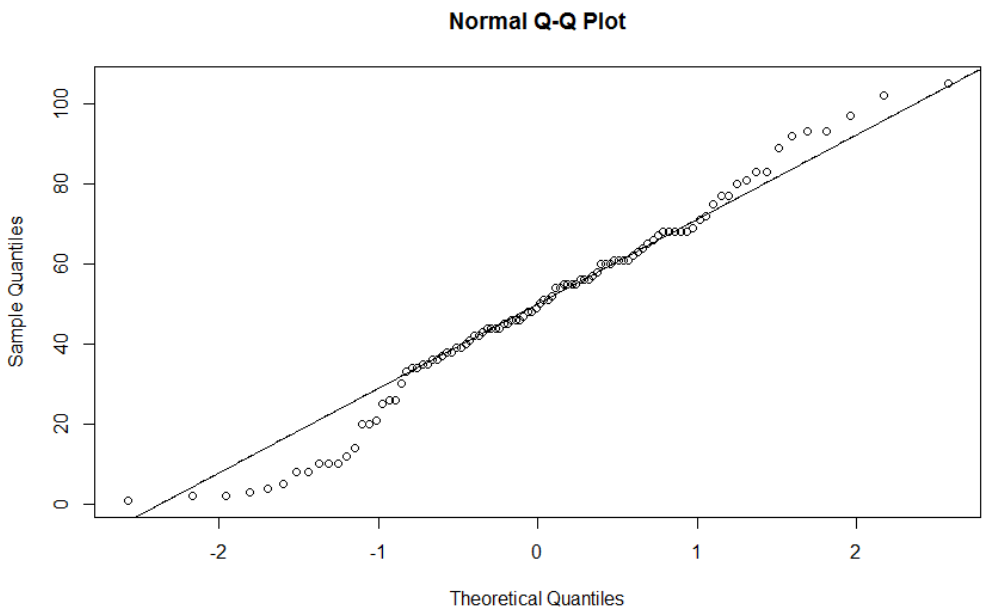
NORMALITY TEST

Normality test

Exercise 1 : The processing times of purchase orders is given in PO_Processing.csv. Is the distribution of processing time is normally distributed?

Normality Check using Normal Q – Q plot

```
> qqnorm(PT)
> qqline(PT)
```



NORMALITY TEST

Normality test

Exercise 1 : The processing times of purchase orders is given in PO_Processing.csv. Is the distribution of processing time is normally distributed?

Reading the data and variable

```
> mydata = PO_Processing
> PT = mydata$Processing_Time
```

Normality Check using Shapiro – Wilk test

```
> shapiro.test(PT)
```

Statistics	Value
W	0.9804
p value	0.1418

Conclusion: The data is Normal if p-value is above 0.05

7. ANALYSIS OF VARIANCE

ANALYSIS OF VARIANCE

ANOVA

Analysis of Variance is a test of means for two or more populations

Partitions the total variability in the variable under study to different components

H0 = $Mean_1 = Mean_2 = - - - = Mean_k$

Reject H0 if p – value < 0.05

Example:

To study location of shelf on sales revenue

ANALYSIS OF VARIANCE

One Way Anova : Example

An electronics and home appliance chain suspect the location of shelves where television sets are kept will influence the sales revenue. The data on sales revenue in lakhs from the television sets when they are kept at different locations inside the store are given in sales revenue data file. The location is denoted as 1:front, 2: middle & 3: rear. Verify the doubt? The data is given in Sales_Revenue_Anova.csv.

ANALYSIS OF VARIANCE

One Way Anova : Example

Factor: Location(A)

Levels : front, middle, rear

Response: Sales revenue

ANALYSIS OF VARIANCE

One Way Anova : Example

Step 1: Calculate the sum, average and number of response values for each level of the factor (location).

Level 1 Sum(A_1):

Sum of all response values when location is at level 1 (front)

= 1.55 + 2.36 + 1.84 + 1.72

= 7.47

nA_1: Number of response values with location is at level 1 (front)

= 4

ANALYSIS OF VARIANCE

One Way Anova : Example

Step 1: Calculate the sum, average and number of response values for each level of the factor (location).

Level 1 Average:

Sum of all response values when location is at level 1 / number of response values with location is at level 1

$= A_1 / nA_1 = 7.47 / 4 = 1.87$

ANALYSIS OF VARIANCE

One Way Anova : Example

Step 1: Calculate the sum, average and number of response values for each level of the factor (location).

	Level 1 (front)	Level 2 (middle)	Level 3 (rear)
Sum	A_1: 7.47	A_2: 30.31	A_3: 15.55
Number	nA_1: 4	nA_2: 8	nA_3: 6
Average	1.87	3.79	2.59

ANALYSIS OF VARIANCE

One Way Anova : Example

Step 2: Calculate the grand total (T)

T = Sum of all the response values

= 1.55 + 2.36 + - - - + 2.72 + 2.07 = 53.33

Step 3: Calculate the total number of response values (N)

N = 18

Step 4: Calculate the Correction Factor (CF)

CF = (Grand Total)^2 / Number of Response values

= T^2 / N = $(537.33)^2$ / 18 = 158.0049

ANALYSIS OF VARIANCE

One Way Anova : Example

Step 5: Calculate the Total Sum of Squares (TSS)

TSS = Sum of square of all the response values - CF

$= 1.55^2 + 2.36^2 + - - - + 2.72^2 + 2.07^2 – 158.0049$

$= 15.2182$

ANALYSIS OF VARIANCE

One Way Anova : Example

Step 6: Calculate the between (factor) sum of square

$$SS_A = A_1^2 / nA_1 + A_2^2 / nA_2 + A_3^2 / nA_3 - CF$$

$$= 7.47^2 / 4 + 30.31^2 / 8 + 15.55^2 / 4 – 158.0049$$

$$= 11.0827$$

Step 7: Calculate the within (error) sum of square

SS_e = Total sum of square – between sum of square

$$= TSS - SS_A = 15.2182 – 11.0827 = 4.1354$$

ANALYSIS OF VARIANCE

One Way Anova : Example

Step 8: Calculate degrees of freedom (df)

Total df = Total Number of response values - 1

= 18 - 1 = 17

Between df

= Number of levels of the factor - 1

= 3 - 1 = 2

Within df = Total df – Between df

= 17 - 2 = 15

ANALYSIS OF VARIANCE

One Way Anova : Example

Anova Table:

Source	df	SS	MS	F	F Crit	P value
Between	2	11.08272	5.541358	20.09949	3.68	0.0000
Within	15	4.135446	0.275696			
Total	17	15.21816				

MS = SS / df

$F = MS_{Between} / MS_{Within}$

F Crit =finv (probability, between df, within df) , probability = 0.05

P value = fdist (F, between df, within df)

ANALYSIS OF VARIANCE

One Way Anova : R Code

Reading data and variables to R

```
> mydata = Sales_Revenue_Anova
> location = mydata$Location
> revenue = mydata$Sales.Revenue
```

Converting location to factor

```
> location = factor(location)
```

Computing ANOVA table

```
> fit = aov(Revenue  ~  location)
> summary(fit)
```

ANALYSIS OF VARIANCE

One Way Anova : Decision Rule

If p value < 0.05, then

The factor has significant effect on the process output or response.

Meaning:

When the factor is changed from 1 level to another level,

there will be significant change in the response.

ANALYSIS OF VARIANCE

One Way Anova : Example Result

For factor Location, $p = 0.000 < 0.05$

Conclusion:

Location has significant effect on sales revenue

Meaning:

The sales revenue is not same for different locations like front, middle & rear

ANALYSIS OF VARIANCE

One Way Anova : Example Result

The expected sales revenue for different location under study is equal to level averages.

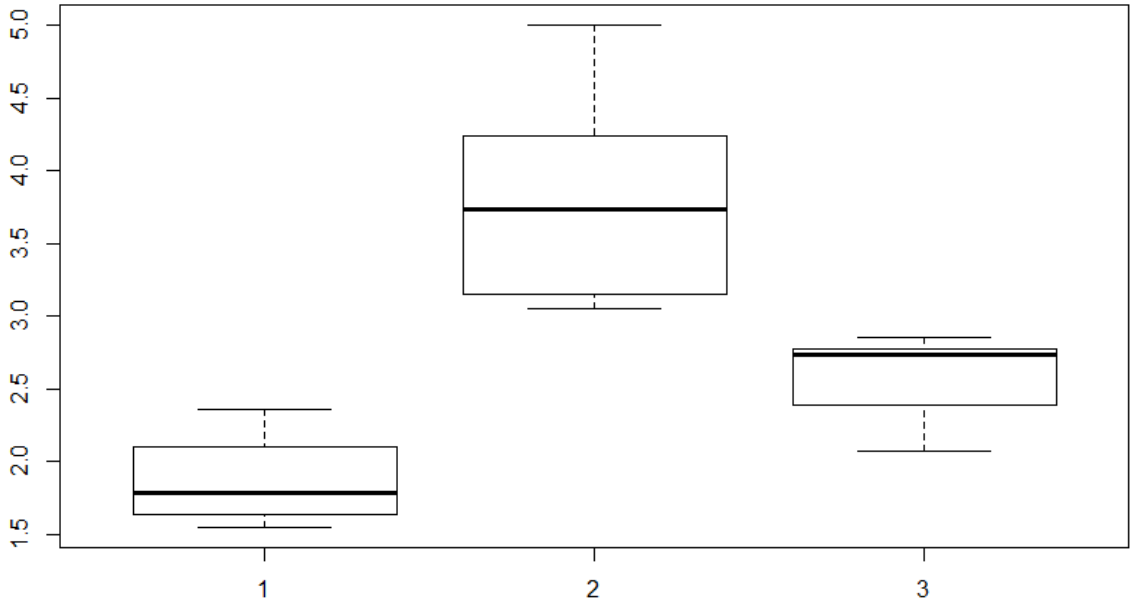
Location	Expected Sales Revenue
Front	1.8675
Middle	3.78875
rear	2.591667

```
> aggregate(Revenue ~ location, FUN = mean)
```

ANALYSIS OF VARIANCE

One Way Anova : Example Result

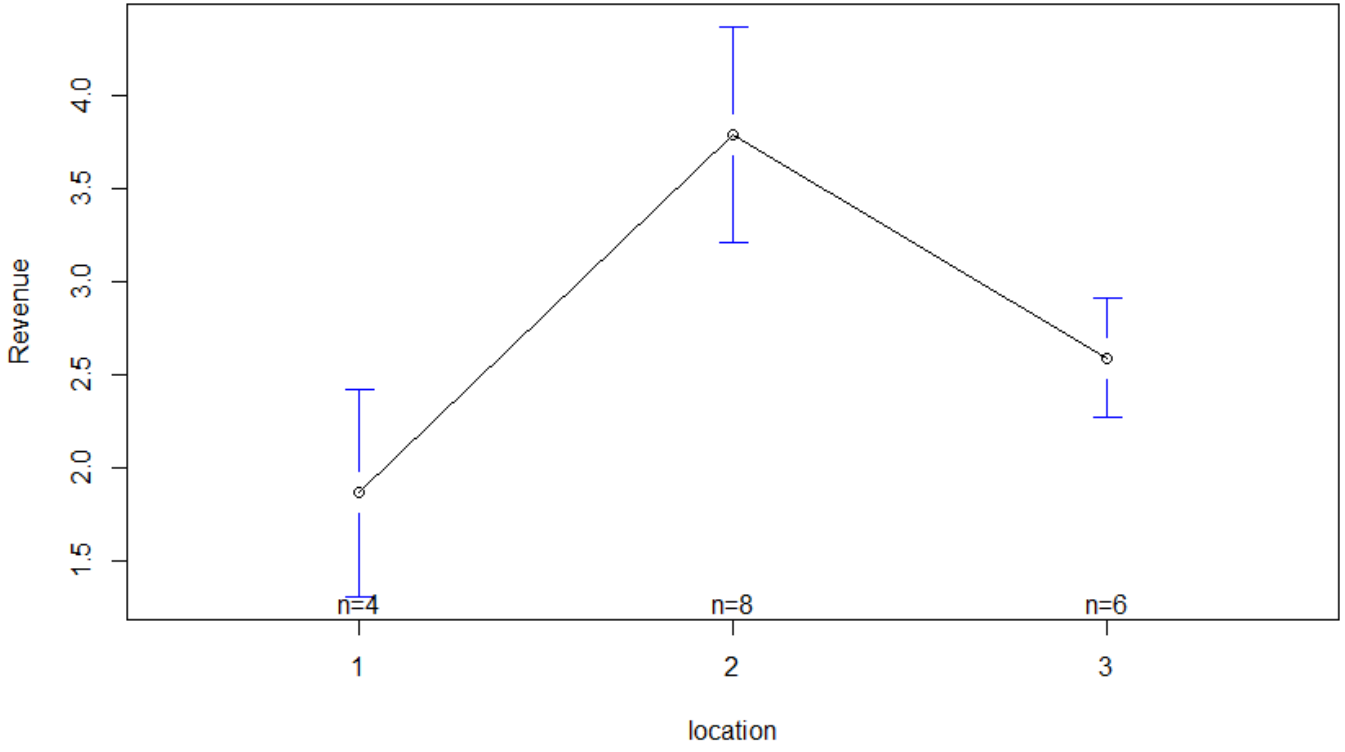
```
> boxplot(Revenue ~ location)
```



ANALYSIS OF VARIANCE

One Way Anova : Example Result

```
> library(gplots)
> plotmeans(Revenue ~ location)
```



ANALYSIS OF VARIANCE

One Way Anova : Tukey's Honestly Significant Difference (HSD) Test

Used to do pair wise comparison between the levels of factors

R code

```
>TukeyHSD(fit)
```

Comparison	Mean difference	Lower	Upper	p value
2 - 1	1.92125	1.086067	2.756433	0.0000
3 - 1	0.724167	-0.15619	1.604527	0.1158
3 - 2	-1.19708	-1.93365	-0.46052	0.0020

ANALYSIS OF VARIANCE

Anova logic:

Two Types of Variations:

1. Variation within the level of a factor
2. Variation between the levels of factor

ANALYSIS OF VARIANCE

Anova logic :

Variation between the level of a factor:

The effect of Factor.

Variation within the levels of a factor:

The inherent variation in the process or Process Error.

	Location		
	Front	Middle	rear
Sales Revenue	1.34 1.89 1.35 2.07 2.41 3.06	3.20 2.81 4.52 4.40 4.75 5.19 3.42 9.80	2.30 1.91 1.40 1.48

ANALYSIS OF VARIANCE

Anova logic :

If the variation between the levels of a factor is significantly higher than the inherent variation

then the factor has significant effect on response

To check whether a factor is significant:

Compare variation between levels with variation within levels

ANALYSIS OF VARIANCE

Anova logic :

Measure of variation between levels: MS of the factor ($MS_{between}$)

Measure of variation within levels: MS Error (MS_{within})

To check whether a factor is significant:

Compare MS of between with MS within

i.e. Calculate $F = MS_{between} / MS_{within}$

If F is very high, then the factor is significant.

ANALYSIS OF VARIANCE

Variation Within levels:

Ideally variation within all the levels should be same

To check whether variation within the levels are same or not

Do Bartlett's test

If p value ≥ 0.05, then variation within the levels are equal, otherwise not

R Code for Bartlett's test

```
> bartlett.test(Revenue, location, data = mydata)
```

ANALYSIS OF VARIANCE

Variation Within levels:

Bartlett's Test result for sales revenue (location of TV sets) example

Bartlett's K^2 Statistic	df	p value
3.8325	2	0.1472

Since p value = 0.1472 > 0.05, the variance within the levels are equal

8. REGRESSION ANALYSIS

REGRESSION ANALYSIS

Regression

Correlation helps

To check whether two variables are related

If related

Identify the type & degree of relationship

Regression helps

- To identify the exact form of the relationship
- To model output in terms of input or process variables

REGRESSION ANALYSIS

Exercise 1: The data from the pulp drying process is given in the file DC_Simple_Reg.csv. The file contains data on the dry content achieved at different dryer temperature. Develop a prediction model for dry content in terms of dryer temperature.

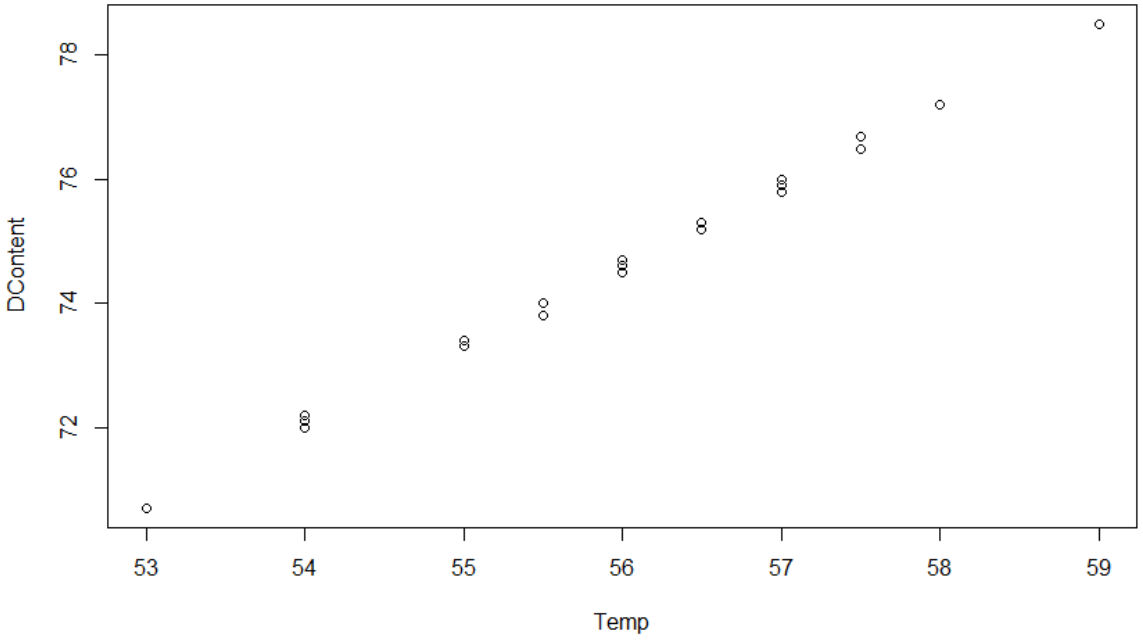
1. Reading the data and variables

```
> mydata = DC_Simple_Reg
> Temp = mydata$Dryer.Temperature
> DContent = mydata$Dry.Content
```

REGRESSION ANALYSIS

2. Constructing Scatter Plot

```
> plot(Temp, DContent)
```



REGRESSION ANALYSIS

3. Computing Correlation Matrix

```
> cor(Temp, DContent)
```

Attribute	Dry Content
Temperature	0.9992

Remark:
Correlation between y & x need to be high (preferably 0.8 to 1 to -0.8 to -1.0)

REGRESSION ANALYSIS

4: Performing Regression

```
> model = lm(DContent ~ Temp)

> summary(model)
```

Statistic	Value	Criteria
Residual standard error	0.07059	
Multiple R-squared	0.9984	> 0.6
Adjusted R-squared	0.9983	> 0.6

Model	df	F	p value
Regression	1	24497	0.000
Residual	40		
Total	41		

Criteria:
P value < 0.05

REGRESSION ANALYSIS

4: Performing Regression

Attribute	Coefficient	Std. Error	t Statistic	p value
Intercept	2.183813	0.463589	4.711	0.00
Temperature	1.293432	0.008264	156.518	0.00

Interpretation

The p value for independent variable need to be < significance level α (generally $\alpha = 0.05$)

Model: Dry Content = 2.183813 + 1.293432 x Temperature

REGRESSION ANALYSIS

5: Regression Anova

```
> anova(model)
```

ANOVA

Source	SS	df	MS	F	p value
Temp	122.057	1	122.057	24497	0.000
Residual	0.199	40	0.005		
Total	122.256	41			

Criteria: P value < 0.05

REGRESSION ANALYSIS

5: Residual Analysis

```
> pred = fitted(model)
> Res = residuals(model)
> write.csv(pred,"D:/ISI/DataSets/Pred.csv")
> write.csv(Res,"D:/ISI/DataSets/Res.csv")
```

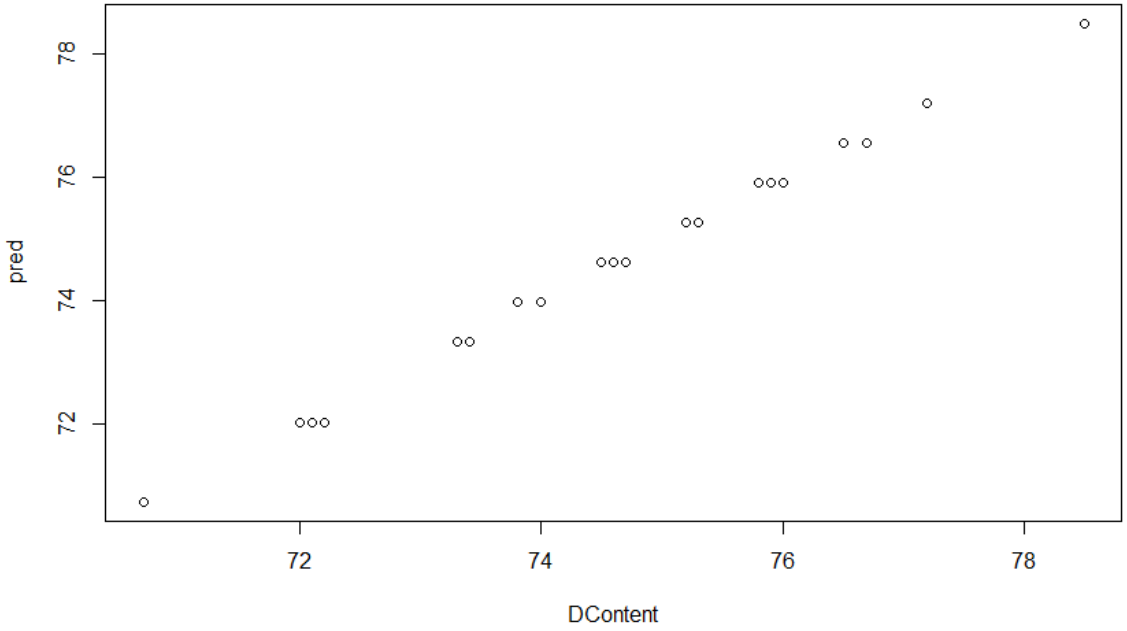
SL No.	Fitted	Residuals	SL No.	Fitted	Residuals
1	73.32259	-0.02259	22	74.61602	-0.01602
2	74.61602	-0.01602	23	75.26274	-0.06274
3	73.96931	0.030693	24	73.96931	0.030693
4	78.49632	0.00368	25	75.90946	-0.00946
5	74.61602	-0.01602	26	75.26274	0.03726
6	73.96931	0.030693	27	73.96931	0.030693
7	75.26274	-0.06274	28	78.49632	0.00368
8	77.20289	-0.00289	29	76.55617	-0.05617
9	75.90946	-0.00946	30	74.61602	-0.11602
10	74.61602	-0.01602	31	75.90946	0.090544
11	73.32259	-0.02259	32	76.55617	-0.05617
12	75.90946	-0.00946	33	76.55617	0.143828
13	75.90946	0.090544	34	75.90946	0.090544
14	74.61602	-0.01602	35	75.90946	-0.10946
15	74.61602	0.083977	36	73.96931	-0.16931
16	74.61602	-0.11602	37	73.32259	-0.02259
17	70.73573	-0.03573	38	74.61602	-0.01602
18	72.02916	-0.02916	39	73.32259	0.077409
19	72.02916	0.070841	40	75.90946	0.090544
20	72.02916	0.170841	41	73.96931	0.030693
21	70.73573	-0.03573	42	75.26274	-0.06274

REGRESSION ANALYSIS

5: Residual Analysis

Scatter Plot: Actual Vs Predicted (fit)

```
> plot(DContent, pred)
```

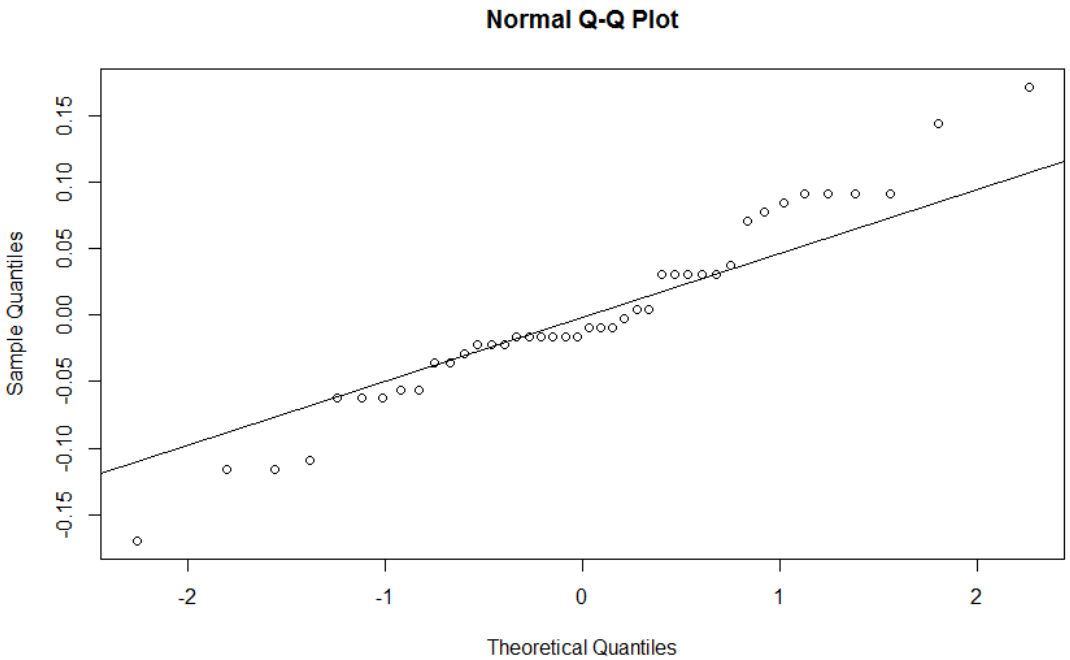


REGRESSION ANALYSIS

5: Residual Analysis

Normality Check on residuals

```
> qqnorm(Res)
> qqline(Res)
```



Residuals should be normally distributed or bell shaped

REGRESSION ANALYSIS

5: Residual Analysis

Normality Check on residuals

```
> shapiro.test(Res)
```

Shapiro-Wilk normality Test:	
W	p value
0.9693	0.3132

Residuals should be normally distributed or bell shaped

REGRESSION ANALYSIS

5: Residual Analysis

```
> plot(pred, Res)
> plot(Temp, Res)
```

Residuals should be independent and stable

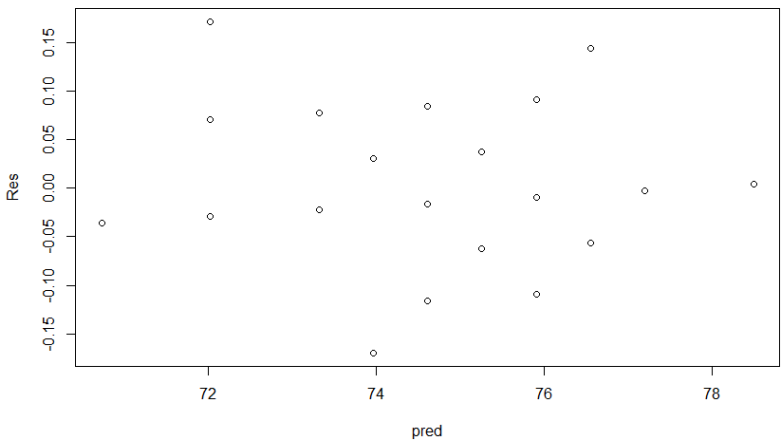
Plot the residuals against fitted value. The points in the graph should be scattered randomly and should not show any trend or pattern. The residuals should not depend in anyway on the fitted value.

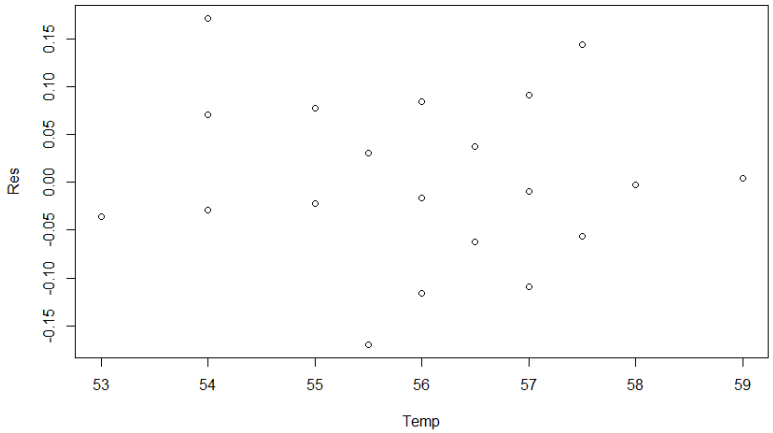
If there is a pattern then a transformation such as log y or $\sqrt{y}$ to be used

Similarly the residuals shall not depend on x. This can be checked by plotting residuals vs x. A pattern in this plot is an indication that the residuals are not independent of x. Instead of x, develop the model with a function of as predictor (Eg: x^2, 1/x, $\sqrt{x}$, log(x), etc.)

REGRESSION ANALYSIS

Residual Analysis





There is no trend or pattern on residuals vs fitted value ,residuals vs observation order or residuals vs x plot. Hence the assumptions of independence and stability of residuals are satisfied.

REGRESSION ANALYSIS

6: Outlier test

Observations with Bonferonni p – value < 0.05 are potential outliers

```
> library(car)
> outlierTest(model)
```

Observation	Studentized Residual	Bonferonni p value
20	2.723093	0.40417

REGRESSION ANALYSIS

7: Leave One Out Cross Validation (LOOCV)

- Split the data into two parts : training data and test data

  Test data consists of only one observation (x_1, y_1)

  Training data consists of the remaining n – 1 observations namely (x_2, y_2) , (x_3, y_3) , - - -, (x_n, y_n)
- Develop the model using n – 1 training data observations and predict the response y_1 of the test data observation

  Compute the residuals and mean square error $MSE_1 = (y_{1actual} - y_{1pred})^2$
- Repeat the process by taking (x_1, y_1) as test data and the remaining n – 1 observations as training data
- Compute MSE_2
- Repeating the procedure n times produces n squared errors MSE_1, MSE_2, - - -, MSE_n
- LOOCV estimate of the test MSE is the average of these n test error estimates

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

REGRESSION ANALYSIS

7: Leave One Out Cross Validation (LOOCV)

```
> library(boot)
> attach(mydata)
> mymodel = glm(Dry.Content ~ Dryer.Temperature)
> valid = cv.glm(mydata, mymodel)
> valid$delta[1]
```

Statistic	Value
Delta	0.005201004

REGRESSION ANALYSIS

Multiple Linear Regression

To model output variable y in terms of two or more variables.

General Form:

$$y = a + b_1x_1 + b_2x_2 + - - - + b_kx_k + \varepsilon$$

Two variable case:

$$y = a + b_1x_1 + b_2x_2 + \varepsilon$$

Where

a: intercept (the predicted value of y when all x's are zero)

b_j: slope (the amount change in y for unit change in x_j keeping all other x's constant, j = 1,2,---,k)

REGRESSION ANALYSIS

Exercise : The effect of temperature and reaction time affects the % yield. The data collected in given in the Mult-Reg_Yield file. Develop a model for % yield in terms of temperature and time?

Step 1: Correlation Analysis

Attribute	Time	Temperature	% Yield
Time	1.00	-0.01	0.90
Temperature	-0.01	1.00	-0.05
% Yield	0.90	-0.05	1.00

Correlation between xs & y should be high

Correlation between xs should be low

REGRESSION ANALYSIS

Step 2: Regression Output

Statistic	Value	Criteria
Adjusted R Square	0.7766	≥ 0.6

Regression ANOVA

Model	SS	df	MS	F	p value
Regression	6797.063	2	3398.531	27.07	0.0000
Residual	1632.08138	13	125.5447		
Total	8429.14438	15			

Criteria: P value < 0.05

REGRESSION ANALYSIS

Step 2: Regression Output

ANOVA

Source	SS	df	MS	F	p value
Time	6777.8	1	6777.8	53.9872	0.000
Temp	19.3	1	19.3	0.1534	0.702
Residual	1632.1	13	125.5		

Criteria: P value < 0.05

REGRESSION ANALYSIS

Step 2: Regression Output – Identify the model

Attribute	Coefficient	Std. Error	t Statistic	p value
Time	0.9061	0.12337	7.344	0.0000
Temperature	-0.0642	0.16391	-0.392	0.702
Intercept	-67.8844	40.58652	-1.67	0.118

Interpretation: Only time is related to % yield as p value < 0.05

REGRESSION ANALYSIS

Step 2: Regression Output – Identify the model

Attribute	Coefficient	Std. Error	t Statistic	p value
Time	0.9065	0.1196	7.580	0.0000
Intercept	-81.6205	19.7906	-4.124	0.00103

Model % Yield= 0.9065 x Time - 81.621

REGRESSION ANALYSIS

Step 3: Residual Analysis

SL No.	Temperature	% Yield	Predicted	Time
1	190	35.0	36.22	130
2	176	81.7	76.10	174
3	205	42.5	39.84	134
4	210	98.3	91.51	191
5	230	52.7	67.94	165
6	192	82.0	94.23	194
7	220	34.5	48.00	143
8	235	95.4	86.98	186
9	240	56.7	44.38	139
10	230	84.4	88.79	188
11	200	94.3	77.01	175
12	218	44.3	59.79	156
13	220	83.3	90.61	190
14	210	91.4	79.73	178
15	208	43.5	38.03	132
16	225	51.7	52.53	148

REGRESSION ANALYSIS

Step 3: Residual Analysis: Outlier detection

SL No.	Temperature	% Yield	Predicted	Time	Residuals	Std Residuals
1	190	35	36.22	130	-1.22	-0.126
2	176	81.7	76.1	174	5.60	0.5358
3	205	42.5	39.84	134	2.66	0.2686
4	210	98.3	91.51	191	6.79	0.6784
5	230	52.7	67.94	165	-15.24	-1.45
6	192	82	94.23	194	-12.23	-1.238
7	220	34.5	48	143	-13.50	-1.322
8	235	95.4	86.98	186	8.42	0.8272
9	240	56.7	44.38	139	12.32	1.2221
10	230	84.4	88.79	188	-4.39	-0.434
11	200	94.3	77.01	175	17.29	1.6575
12	218	44.3	59.79	156	-15.49	-1.479
13	220	83.3	90.61	190	-7.31	-0.727
14	210	91.4	79.73	178	11.67	1.1244
15	208	43.5	38.03	132	5.47	0.5582
16	225	51.7	52.53	148	-0.83	-0.081
				Mean	0.000	
				SD	10.4918	

REGRESSION ANALYSIS

Step 3: Residual Analysis:

Shapiro-Wilk normality Test: Yield data	
W	p value
0.9449	0.4132

REGRESSION ANALYSIS

6: Outlier test

Observations with Bonferonni p – value < 0.05 are potential outliers

```
> library(car)
> outlierTest(mymodel)
```

Observation	Studentized Residual	Bonferonni p value
11	1.781515	NA

REGRESSION ANALYSIS

7: Leave One Out Cross Validation (LOOCV)

```
> library(boot)
> attach(mydata)
> mymodel = glm(X.Yield ~ Time)
> myvalidation = cv.glm(mydata, mymodel)
> myvalidation$delta[1]
```

Statistic	Value
Delta	128.8541

REGRESSION ANALYSIS

Exercise : The effect of temperature, time and kappa number of pulp affects the % conversion of UB pulp to Cl_2 pulp. inspection. The data collected in given in the Mult_Reg_Conversion file. Develop a model for % conversion in terms of exploratory variables?

REGRESSION ANALYSIS

Step 1: Correlation Analysis

	Temperature	Time	Kappa #	% Conversion
Temperature	1.00	-0.96	0.22	0.95
Time	-0.96	1.00	-0.24	-0.91
Kappa #	0.22	-0.24	1.00	0.37
% Conversion	0.95	-0.91	0.37	1.00

Interpretation

High Correlation between % Conversion and Temperature & Time

High Correlation between Temperature & Time - Multicollinearity

REGRESSION ANALYSIS

Measure for Multicollinearity

Variance Inflation Factor (VIF)

Measures the correlation (linear association) between each x variable with other x's

$VIF_i = 1/(1- R_i^2)$

Where R_i is the coefficient for regressing x_i on other x's

Criteria: VIF > 5 indicates multicollinearity.

REGRESSION ANALYSIS

Regression Output

Statistic	Value	Criteria
Adjusted R Square	0.899	> 0.6

Regression ANOVA

Model	SS	df	MS	F	p value
Regression	1953.419	3	651.140	45.885	0.0000
Residual	170.290	12	14.191		
Total	2123.709	15			

REGRESSION ANALYSIS

Regression Output

	Coeff	Std. Error	t	p value
Constant	-121.27	55.43571	-2.19	0.0492
Temperature	0.12685	0.04218	3.007	0.0109
Time	-19.0217	107.92824	-0.18	0.863
Kappa #	0.34816	0.17702	1.967	0.0728

Variance-inflation factors (VIF)

```
> vif(mymodel)
```

x	VIF
Temperature	12.23
Time	12.33
Kappa #	1.062

REGRESSION ANALYSIS

Tackling Multicollinearity:

1. Remove one or more of highly correlated independent variable
2. Principal Component Regression
3. Partial Least Square Regression
4. Ridge Regression

REGRESSION ANALYSIS

Tackling Multicollinearity:

Method 1: Removing highly correlated variable – Stepwise Regression

Approach

- A null model is developed without any predictor variable x. In null model, the predicted value will be the overall mean of y
- Then predictor variables x's are added to the model sequentially
- After adding each new variable, the method also remove any variable that no longer provide an improvement in the model fit
- Finally the best model is identified as the one which minimizes Akaike information criterion (AIC)

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2$$

REGRESSION ANALYSIS

Tackling Multicollinearity:

Method 1: Removing highly correlated variable – Stepwise Regression

Akaike information criterion (AIC)

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

n: number of observations

$\hat{\sigma}^2$: estimate of error or residual variance

d: number of x variables included in the model

RSS: Residual sum of squares

REGRESSION ANALYSIS

Tackling Multicollinearity:

Method 1: Removing highly correlated variable – Stepwise Regression

R code

```
> library(MASS)
> mymodel = lm(X..Conversion ~ Temperature + Time + Kappa.number)
> step =stepAIC(mymodel, direction = "both")
```

Step	x's in the model	AIC
1	Temperature, Time & Kappa Number	45.8
2	Temperature & Kappa Number	43.9

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 1: Stepwise Regression

Attribute	Coefficient	Std. Error	t Statistic	p value
Temperature	0.13396	0.01191	11.250	0.0000
Kappa #	0.35106	0.16955	2.071	0.0589
Intercept	-130.68986	14.14571	-9.239	0.0000

% Conversion = 0.13396 * Temperature + 0.35106 * Kappa # - 130.68986

Variance-inflation factors (VIF)

x	VIF
Temperature	1.0526
Kappa #	1.0526

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 1: Stepwise Regression

```
> pred = predict(mymodel)
> res = residuals(mymodel)
 > cbind(X..Conversion, pred, res)
> mse = mean(res^2)
> rmse = sqrt(mse)
```

Statistic	Value
Mean Square Error (MSE)	10.7
Root Mean Square Error (RMSE)	3.27

REGRESSION ANALYSIS

k fold Cross Validation

Steps

1. Divide the data set into k equal subsets
2. Keep one subset (sample) for model validation
3. Develop the model using all the other k – 1 subsets data put together
4. Predict the responses for the test data and compute residuals
5. Return the test sample back to the original data set and take another subset for model validation
6. Go to step 3 and continue until all the subsets are tested with different models
7. Compute the overall Root Mean Square Residuals. RMSE of validation should not be high compared to the original model developed with all the data points together.

Note: when k = n, then k fold cross validation is same as leave one out cross validation

REGRESSION ANALYSIS

k fold Cross Validation

R code

```
> library(DAAG)
> cv.lm(mymodel, m = 16)
> cv.lm(mymodel, df = mydata, m = 16)
```

m: number of validations required. M = 16 = n, hence equal to leave one out cross validation

Model	MSE	RMSE
Original	10.7	3.27
Cross Validation	19.6	4.43

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

1. Perform principal component analysis on x variables
2. Use the principal components as x variables and develop the model

R Code : Principal Component Regression

```
> mydata = mydata[,2:5]
> attach(mydata)
> library(pls)
> mymodel = pcr(X..Conversion ~ ., data = mydata, scale = TRUE)
> summary(mymodel)
> mymodel$loadings
```

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

1. Perform principal component analysis on x variables
2. Use the principal components as x variables and develop the model

Cum % Variance	PC1	PC2	PC3
x	68.66	98.61	100
Conversion (y)	90.48	90.62	91.98

Component 1 or 1 & 2 may be sufficient to include in the model

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

1. Perform principal component analysis on x variables
2. Use the principal components as x variables and develop the model

Loadings	PC1	PC2	PC3
Temperature	-0.674	0.218	0.705
Time	0.677	-0.2	0.709
Kappa.number	-0.296	-0.955	0

Component 1 is taking care of information in temperature and Time and Component 2 is mostly representing kappa number

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

Principal Component Scores

SL No.	Comp 1	Comp 2	Comp 3
1	-1.079	1.2498	0.1202
2	-1.158	0.9967	0.1236
3	-1.273	0.6625	0.117
4	-1.371	0.2313	0.1563
5	-1.543	-0.362	0.1756
6	-1.889	-1.365	0.1558
7	0.4709	1.1733	-0.133
8	0.3133	0.8148	-0.173
9	0.0021	0.2622	-0.299
10	-0.257	-0.122	-0.428
11	-0.268	-0.763	-0.24
12	-0.432	-1.819	-0.07
13	2.2484	0.6246	-0.022
14	2.4329	0.165	0.2963
15	2.1218	-0.388	0.1699
16	1.6801	-1.362	0.0493

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 2: Principal Component Regression

Identifying the required number of components in the model

```
> pred = predict(mymodel, type = "response", ncomp = 1)
> res = X..Conversion - pred
> mse = mean(res^2)
> prednew = predict(mymodel, type = "response", ncomp = 2)
> resnew = X..Conversion - prednew
> msenew = mean(resnew^2)
```

Statistics	Regression with	
	PC1	PC1 & PC2
MSE	12.64226	12.45593

Since there is not much reduction in MSE by including the second principal component , only PC1 is required for modelling

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

Principal component regression involves the identification of a linear combinations of predictors that best represents the x variables

The response y is not used to help the determination of principal components

The response does not supervise the identification of principal components

Identifies the best linear combinations which best explains the predictor variables x but may not the ones best for predicting the response y

Partial least square regression is a supervised alternative to principal component regression

Partial least square method identifies the components or directions (linear combinations of x variables) using the response variable y.

Partial least square places highest weight on the variables that are most strongly related the response y

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

R code

```
> mydata = mydata[,2:5]
> attach(mydata)
> library(pls)
> mymodel = plsr(X..Conversion ~ ., data = mydata, scale = TRUE)
> summary(mymodel)
> mymodel$loading
```

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

Cum % Variance	PLS1	PLS2	PLS3
x	68.65	96.92	100
Conversion (y)	90.63	90.86	91.98

Loadings	PLS1	PLS2	PLS3
Temperature	0.677	0.344	0.299
Time	-0.679	-0.207	0.607
Kappa.number	0.285	-1.391	0.736

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial Least Square Regression

```
> ps = mymodel$scores
> score = ps[,1:2]
```

SL No	PLS1	PLS2
1	1.11324	0.89634
2	1.18502	0.73368
3	1.2913	0.51027
4	1.3792	0.25877
5	1.5361	-0.1142
6	1.85493	-0.7845
7	-0.4425	0.66627
8	-0.2949	0.40157
9	-0.0005	-0.0564
10	0.24599	-0.4059
11	0.24426	-0.6809
12	0.3833	-1.24
13	-2.2314	0.4067
14	-2.4222	0.35105
15	-2.1279	-0.1069
16	-1.7138	-0.8359

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 3: Partial least square regression

Identifying the required number of components in the model

```
> pred = predict(mymodel, data = mydata, scale = TRUE, ncomp = 1)
> res = X..Conversion - pred
> mse = mean(res^2)

> prednew = predict(mymodel, , data = mydata, scale = TRUE ,  ncomp = 2)
> resnew = X..Conversion - prednew
> msenew = mean(resnew^2)
```

Statistics	Regression with	
	PLS1	PLS11 & PLS2
MSE	12.44252	12.13185

Since there is not much reduction in MSE by including the second component , only PLS1 is required for modelling

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression

In least square regression, the coefficients β's of x variables are identified by minimizing the residual sum of squares

$$RSS = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2$$

In ridge regression, the coefficients β's of x variables are identified by minimizing a slightly different quantity

$$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

Where $\lambda \geq 0$ is a turning parameter and $\lambda \sum_{j=1}^{p} \beta_j^2$ is the shrinkage penalty,

which will be small when $\beta_1, \beta_2, - - -, \beta_p$ are close to zero.

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression

Ridge regression seeks coefficient estimates that fit the data well by minimizing the RSS and the tuning parameter λ has the effect of shrinking the estimates β_j towards zero
The value of λ s identified through 10 fold cross validation

10 fold Cross Validation

- Divide the data set into 10 equal parts
- Develop the model using 9 parts and test it with the remaining one part
- Repeat the process 10 times to get an unbiased estimate of MSE

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression

R Code

```
> library(glmnet)
> set.seed(1)
> y = mydata[,5]]
> x =mydata[,2:4]
> x = as.matrix(x)
```

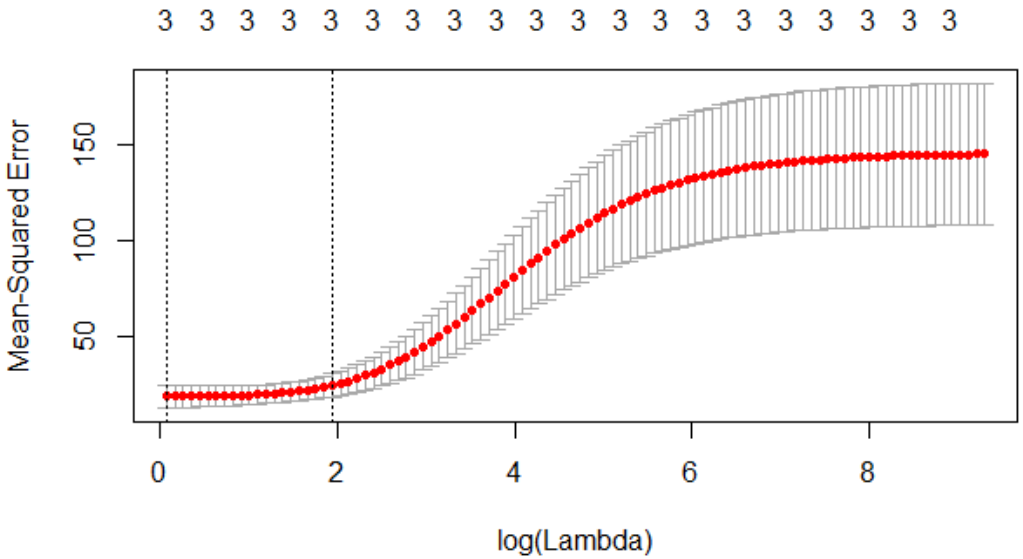
Cross Validation

```
> mymodel = cv.glmnet(x , y, alpha =0)
> plot(mymodel)
```

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression



Choose the λ which minimizes the mean square error

```
> bestlambda = mymodel$lambda.min
```

Best λ= 1.088771

REGRESSION ANALYSIS

Tackling Multi collinearity:

Method 4: Ridge regression

Develop the model with best λ and identify the coefficients

```
> mynewmodel = glmnet(x, y, alpha = 0)
> predict (mynewmodel, type = "coefficients", s = bestlambda)[1:4,]
```

Variable	Coefficients
(Intercept)	-63.0713
Temperature	0.0823
Time	-117.5048
Kappa.number	0.3268

CORRELATION & REGRESSION

Regression with dummy variables

When x's are not numeric but nominal

Each nominal or categorical variable is converted into dummy variables

Dummy variables takes values 0 or 1

Number of dummy variable for one x variable is equal to number of distinct values of that variable - 1

Example: A study was conducted to measure the effect of gender and income on attitude towards vocation. Data was collected from 30 respondents and is given in Travel_dummy_reg file. Attitude towards vocation is measured on a 9 point scale. Gender is coded as male = 1 and female = 2. Income is coded as low=1, medium = 2 and high = 3. Develop a model for attitude towards vocation in terms of gender and Income?

CORRELATION & REGRESSION

Regression with dummy variables

Variable		Dummy
Gender	Code	gender_Code
Male	1	0
Female	2	1

Variable		Dummy	
Income	Code	Income1	Income 2
Low	1	0	0
Medium	2	1	0
High	3	0	1

CORRELATION & REGRESSION

Regression with dummy variables

Read the fie and variables

```
> mydata =  Travel_dummy_Reg
> mydata = mydata[,2:4]
> gender = mydata$Gender
> Income = mydata$Income
> Attitude = mydata$Attitude
```

Converting categorical x's to factors

```
> gender = factor(gender)
> income = factor(income)
```

CORRELATION & REGRESSION

Regression with dummy variables – Output

```
> mymodel =  lm(attitude ~ genser + income)
> summary (mumodel)
```

Multiple R^2	0.8603
Adjusted R^2	0.8442
F Statistics	53.37
P value	0.00

	Estimate	Std. Error	t value	p value
(Intercept)	2.4	0.3359	7.145	0.00000
gender2	-1.6	0.3359	-4.763	0.00006
income2	2.8	0.4114	6.806	0.00000
income3	4.8	0.4114	11.668	0.00000

CORRELATION & REGRESSION

Regression with dummy variables – Output

> anova (mumodel)

	Df	Sum Sq	Mean Sq	F	p value
gender	1	19.2	19.2	22.691	0.0001
income	2	116.27	58.133	68.703	0.0000
Residuals	26	22	0.846		

9. BINARY LOGISTIC REGRESSION

BINARY LOGISTIC REGRESSION

Used to develop models when the output or response variable y is binary

The output variable will be binary, coded as either success or failure

Models probability of success p which lies between 0 and 1

Linear model is not appropriate

$$p = \frac{e^{a+b_1x_1+b_2x_2+---+b_kx_k}}{1+e^{a+b_1x_1+b_2x_2+---+b_kx_k}}$$

p: probability of success
x_i's : independent variables
a, b_1, b_2, ---: coefficients to be estimated

If estimate of $p \geq 0.5$, then classified as success, otherwise as failure

BINARY LOGISTIC REGRESSION

Usage: When the dependant variable (Y variable) is binary

Example: Develop a model to predict the number of visits of family to a vacation resort based on the salient characteristics of the families. The data collected from 30 households is given in Resort_Visit.csv

1. Reading the file and variables

```
> mydata = Resort_Visit
> visit = mydata$Resort_Visit
> income = mydata$Family_Income
> attitude = mydata$Attitude.Towards.Travel
> importance = mydata$Importance_Vacation
> size = mydata$House_Size
> age = mydata$Age._Head
```

2. Converting response variable to discrete

```
> visit = factor(visit)
```

BINARY LOGISTIC REGRESSION

3. Correlation Matrix

```
> cor(mydata)
```

	Resort_Visit	Family_Income	Attitude_Travel	Importance_Vacation	House_Size	Age_Head
Resort_Visit	1.00	-0.60	-0.27	-0.42	-0.59	-0.21
Family_Income	-0.60	1.00	0.30	0.23	0.47	0.21
Attitude_Travel	-0.27	0.30	1.00	0.19	0.15	-0.13
Importance_Vacation	-0.42	0.23	0.19	1.00	0.30	0.11
House_Size	-0.59	0.47	0.15	0.30	1.00	0.09
Age_Head	-0.21	0.21	-0.13	0.11	0.09	1.00

Interpretation: Correlation between X variables should be low

BINARY LOGISTIC REGRESSION

4. Converting response variable to discrete

```
> visit = factor(visit)
```

5. Checking relation between Xs and Y

```
> aggregate(income ~visit, FUN = mean)
> aggregate(attitude ~visit, FUN = mean)
> aggregate(importance ~visit, FUN = mean)
> aggregate(size ~visit, FUN = mean)
> aggregate(age ~visit, FUN = mean)
```

Resort_Visit	Mean				
	Family_Income	Attitude_Travel	Importance_Vacation	House_Size	Age_Head
0	58.5200	5.4000	5.8000	4.3333	53.7333
1	41.9133	4.3333	4.0667	2.8000	50.1333

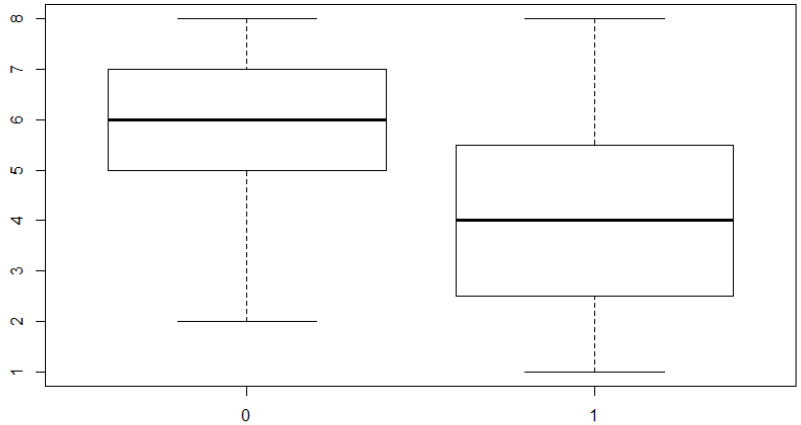
Higher the difference in means, stronger will be the relation to response variable

BINARY LOGISTIC REGRESSION

5. Checking relation between Xs and Y – box plot

```
> boxplot(income ~ visit)
> boxplot(attitude ~ visit)
> boxplot(importance ~ visit)
> boxplot(size ~ visit)
> boxplot(age ~ visit)
```

Income Vs visit



BINARY LOGISTIC REGRESSION

6. Perform Logistic regression

```
> model = glm(visit ~ income + attitude + importance + size + age, family = binomial(logit))
> summary(model)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	15.49503	6.68017	2.32	0.0204
Income	-0.11698	0.06605	-1.771	0.0766
attitude	-0.28129	0.33919	-0.829	0.4069
importance	-0.46157	0.32006	-1.442	0.1493
size	-0.80699	0.49314	-1.636	0.1018
age	-0.07019	0.07199	-0.975	0.3295

BINARY LOGISTIC REGRESSION

6. Perform Logistic regression - Anova

> anova(model,test = 'Chisq')> summary(model)

	Df	Deviance	Resid.Df	Resid.Dev	Pr(>Chi)
NULL	29	41.589			
income	1	12.9813	28	28.608	0.00031
attitude	1	0.4219	27	28.186	0.51598
importance	1	3.8344	26	24.351	0.05021
size	1	3.4398	25	20.911	0.06364
age	1	1.0242	24	19.887	0.31152

Since p value < 0.05 for Income, Importance_Vacation & Size, redo the modelling with important factors only

BINARY LOGISTIC REGRESSION

7. Perform Logistic regression - Modified

	Estimate	Std Error	z value	p value
(Intercept)	8.46599	3.02494	2.799	0.00513
Income	-0.10641	0.05156	-2.064	0.03904
Size	-0.93539	0.47632	-1.964	0.04955

Since p value < 0.05 for both factors, Income & Size, the response variable can be modelled in terms of those two factors

The model is

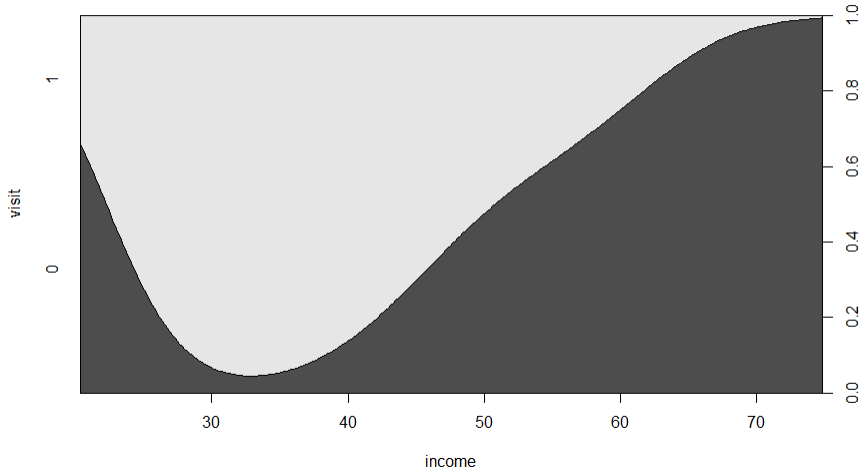
$$y = \frac{e^{8.46599 - 0.10641\,\text{Annual\_Income} - 0.93539\,Size}}{1 + e^{8.46599 - 0.10641\,\text{Annual\_Income} - 0.93539\,Size}}$$

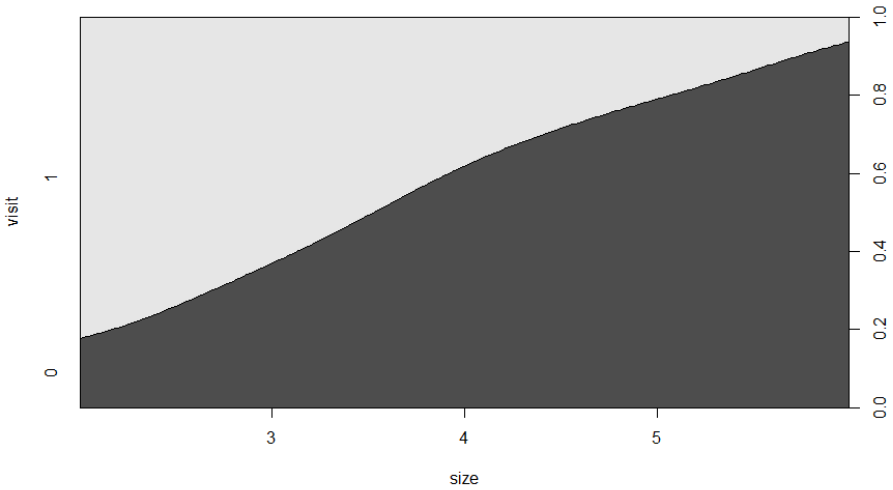
BINARY LOGISTIC REGRESSION

8. Conditional Density plots (Response Vs Factors)

Describing how the conditional distribution of a categorical variable y changes over a numerical variable x

```
> cdplot(visit ~ income)
> cdplot(visit ~ size)
```





BINARY LOGISTIC REGRESSION

9. Fitted Values and residuals

```
> predict(model,type = 'response')
> residuals(model,type = 'deviance')
> predclass = ifelse(predict(model, type ='response')>0.5,"1","0")
```

SL No.	Actual	Fitted	Residuals	Predicted Class	SL No.	Actual	Fitted	Residuals	Predicted Class
1	0	0.970979	-2.66073	1	16	1	0.904132	0.448954	1
2	0	0.059732	-0.35097	0	17	1	0.939523	0.353222	1
3	0	0.021049	-0.20627	0	18	1	0.880611	0.50426	1
4	0	0.202309	-0.67236	0	19	1	0.345537	1.457845	0
5	0	0.292461	-0.83182	0	20	1	0.724535	0.802777	1
6	0	0.014893	-0.17324	0	21	1	0.925508	0.393479	1
7	0	0.677783	-1.50501	1	22	1	0.677559	0.882337	1
8	0	0.038723	-0.28105	0	23	1	0.680103	0.878079	1
9	0	0.109432	-0.48145	0	24	1	0.516151	1.150092	1
10	0	0.030543	-0.24908	0	25	1	0.680326	0.877704	1
11	0	0.017609	-0.1885	0	26	1	0.77062	0.721887	1
12	0	0.050856	-0.32309	0	27	1	0.629425	0.962235	1
13	0	0.04202	-0.29301	0	28	1	0.954395	0.305541	1
14	0	0.601981	-1.35739	1	29	1	0.841493	0.587498	1
15	0	0.499424	-1.17643	0	30	1	0.900286	0.45835	1

BINARY LOGISTIC REGRESSION

10. Model Evaluation

```
> mytable = table(visit, predclass)
> mytable
> prop.table(mytable)
```

	Predicted Count		Total
Actual Count	0	1	
0	12	3	15
1	1	14	15
Total	13	17	30

	Predicted %		Total
Actual %	0	1	
0	40	10	50
1	3	47	50
Total	43	50	100

Statistics	Value
Accuracy %	87
Error %	13

Accuracy of $\geq$ 80 % is good

10. ORDINAL LOGISTIC REGRESSION

ORDINAL LOGISTIC REGRESSION

Used to develop models when the output or response variable y is prdinall

The output variable will be categorical, having more than two categories

ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Read the data file and variables

```
> dd = mydata$DD
> effort = mydata$Effort
> coverage = mydata$Test.Coverage
```

ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Make one of the classes (say “Low”) of output variable as the baseline level

```
> library(MASS)
> mymodel = polr(dd ~ effort + coverage)
> summary(mymodel)
```

ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Coefficients

effort	coverage
0.0234	0.0257

Intercepts

High | Low	Low | Medium
1.4947	3.925

ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Predicted values

```
> pred = predict(mymodel)
> fit = fitted(mymodel)
>fit
> output = cbind(dd, pred)
>write.csv(output, "E:/Infosys/Part 2/output.csv")
```

ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted

```
> mytable = table(dd, pred)
> mytable
> prop.table(mytable)
```

		Predicted		
Actual		High	Low	Medium
	High	8	42	0
	Low	0	105	0
	Medium	1	44	0

ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted (in %)

		Predicted		
Actual		High	Low	Medium
	High	4.0	21.0	0.00
	Low	0.00	52.50	0.00
	Medium	0.50	22.0	0.00

Accuracy = 4 + 52.5 + 0.00 = 0.565 = 56.5%

For other queries mail me at tanujitisi@gmail.com

THANK YOU