DATA VISUALIZATION LAB MANUAL
MR23-1CS0150
Index

S.No	Task
Week1	Import a sales dataset and perform different data manipulation techniques.
Week 2	Perform different data pre-processing techniques on the sales dataset.
Week 3	Conduct a complete data analysis on a given student results dataset and derive insights using the ggplot2 package in R.
Week 4	Perform a data analysis on the weather dataset and extract insights using the ggplot2 package in R. Utilize Histograms, Box plots, Bar charts, Scatter plots, and Line charts to visualize the data.
Week 5	Merge two DataFrames and apply various data manipulation techniques using the Pandas library in Python.
Week 6	Use the Python 'Matplotlib' to perform a thorough data analysis and extract insights from a given Housing dataset.

WEEK-1

Import a sales dataset and perform below data manipulation techniques.

1.Add new rows
2.Create new column “total_revenue” by multiplying quantity sold by the price.
3.Delete rows.
4.Delete column.
5.Rename “Quantity” column to “Quantity_sold”.
6.Create new columns for “day”,“month” and “year” from “Order Date”.
7.Add +2 to “Quantity” variable of South Region.

```
# load sales dataset
data = read.csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/sales_data.csv",fileEncodi
ng = "UTF-8-BOM")
#examin the data
head(data)

##   Row.ID       Order.ID Order.Date  Ship.Date       Country Region
## 1      1 CA-2016-152156 08-11-2016 11-11-2016 United States  South
## 2      2 CA-2016-152156 08-11-2016 11-11-2016 United States  South
## 3      3 CA-2016-138688 12-06-2016 16-06-2016 United States   West
## 4      4 US-2015-108966 11-10-2015 18-10-2015 United States  South
## 5      5 US-2015-108966 11-10-2015 18-10-2015 United States  South
## 6      6 CA-2014-115812 09-06-2014 14-06-2014 United States   West
##          Category    Sales Quantity
## 1       Furniture 261.9600        2
## 2       Furniture 731.9400        3
## 3 Office Supplies  14.6200        2
## 4       Furniture 957.5775        5
## 5 Office Supplies  22.3680        2
## 6       Furniture  48.8600        7

# Check the dimentions
dim(data)

## [1] 690   9

#check the structure of the data
str(data)

## 'data.frame':    690 obs. of  9 variables:
##  $ Row.ID    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Order.ID  : Factor w/ 321 levels "CA-2014-101476",..: 157 157 146 273 273 1
2 12 12 12 12 ...
##  $ Order.Date: Factor w/ 262 levels "01-02-2014","01-03-2014",..: 60 60 91 86
86 66 66 66 66 66 ...
##  $ Ship.Date : Factor w/ 279 levels "01-05-2016","01-06-2016",..: 99 99 143 16
7 167 126 126 126 126 126 ...
##  $ Country   : Factor w/ 1 level "United States": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Region    : Factor w/ 4 levels "Central","East",..: 3 3 4 3 3 4 4 4 4 4 ...
##  $ Category  : Factor w/ 3 levels "Furniture","Office Supplies",..: 1 1 2 1 2
1 2 3 2 2 ...
##  $ Sales     : num  262 731.9 14.6 957.6 22.4 ...
##  $ Quantity  : int  2 3 2 5 2 7 4 6 3 5 ...

tail(data)

##     Row.ID       Order.ID Order.Date  Ship.Date       Country Region
## 685    685 US-2017-168116 04-11-2017 04-11-2017 United States  South
```

```
## 686    686 CA-2014-157784 05-07-2014 08-07-2014 United States  South
## 687    687 CA-2014-157784 05-07-2014 08-07-2014 United States  South
## 688    688 CA-2014-157784 05-07-2014 08-07-2014 United States  South
## 689    689 CA-2017-161480 25-12-2017 29-12-2017 United States   East
## 690    690 US-2014-117135 21-06-2014 23-06-2014 United States  South
##            Category   Sales Quantity
## 685 Office Supplies 167.440        2
## 686      Technology 479.970        3
## 687 Office Supplies  14.620        2
## 688 Office Supplies  19.440        3
## 689       Furniture 191.984        2
## 690       Furniture 104.010        1
```

1.ADD rows

```
df <- data.frame(
  Row.ID = c(693L, 694L),
  Order.ID = c("CA-2016-789123", "US-2018-987654"),
  Order.Date = c("05-11-2015", "14-12-2016"),
  Ship.Date = c("12-11-2015", "20-12-2016"),
  Country = c("United States", "United States"),
  Region = c("West", "Central"),
  Category = c("Technology", "Furniture"),
  Sales = c(543.8, 789.6),
  Quantity = c(2L, 7L)
)

data = rbind(data,df)
head(data)
```

```
##   Row.ID       Order.ID Order.Date  Ship.Date       Country Region
## 1      1 CA-2016-152156 08-11-2016 11-11-2016 United States  South
## 2      2 CA-2016-152156 08-11-2016 11-11-2016 United States  South
## 3      3 CA-2016-138688 12-06-2016 16-06-2016 United States   West
## 4      4 US-2015-108966 11-10-2015 18-10-2015 United States  South
## 5      5 US-2015-108966 11-10-2015 18-10-2015 United States  South
## 6      6 CA-2014-115812 09-06-2014 14-06-2014 United States   West
##          Category    Sales Quantity
## 1       Furniture 261.9600        2
## 2       Furniture 731.9400        3
## 3 Office Supplies  14.6200        2
## 4       Furniture 957.5775        5
## 5 Office Supplies  22.3680        2
## 6       Furniture  48.8600        7
```

```
dim(data)
```

```
## [1] 692   9
```

```
print(data[data$Row.ID==693, ])
```

```
##     Row.ID       Order.ID Order.Date  Ship.Date       Country Region
## 691    693 CA-2016-789123 05-11-2015 12-11-2015 United States   West
##       Category Sales Quantity
## 691 Technology 543.8        2
```

2.Create new column “Total_revenue” by multiplying quantity sold by the price.

```
library(dplyr)
```

```
data = mutate(data, Total_revenue=Sales*Quantity)
head(data)
```

```
##   Row.ID       Order.ID Order.Date  Ship.Date       Country Region
## 1      1 CA-2016-152156 08-11-2016 11-11-2016 United States  South
## 2      2 CA-2016-152156 08-11-2016 11-11-2016 United States  South
## 3      3 CA-2016-138688 12-06-2016 16-06-2016 United States   West
## 4      4 US-2015-108966 11-10-2015 18-10-2015 United States  South
## 5      5 US-2015-108966 11-10-2015 18-10-2015 United States  South
## 6      6 CA-2014-115812 09-06-2014 14-06-2014 United States   West
##          Category    Sales Quantity Total_revenue
## 1       Furniture 261.9600        2       523.920
## 2       Furniture 731.9400        3      2195.820
## 3 Office Supplies  14.6200        2        29.240
## 4       Furniture 957.5775        5      4787.887
## 5 Office Supplies  22.3680        2        44.736
## 6       Furniture  48.8600        7       342.020
```

3.Delete first 5 rows.

```
data = data[-1:-5, ]
dim(data)
```

```
## [1] 687  10
```

4.Delete “Row.ID” column.

```
data$Row.ID = NULL
head(data)
```

```
##          Order.ID Order.Date  Ship.Date       Country Region
## 6  CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 7  CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 8  CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 9  CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 10 CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 11 CA-2014-115812 09-06-2014 14-06-2014 United States   West
##           Category    Sales Quantity Total_revenue
## 6        Furniture   48.860        7       342.020
## 7  Office Supplies    7.280        4        29.120
## 8       Technology  907.152        6      5442.912
## 9  Office Supplies   18.504        3        55.512
## 10 Office Supplies  114.900        5       574.500
## 11       Furniture 1706.184        9     15355.656
```

5.Reaname “Quantity” column to “Quantity_sold”.

```
data = rename(data, Quantity_sold=Quantity)
head(data)
```

```
##          Order.ID Order.Date  Ship.Date       Country Region
## 6  CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 7  CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 8  CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 9  CA-2014-115812 09-06-2014 14-06-2014 United States   West
```

```
## 10 CA-2014-115812 09-06-2014 14-06-2014 United States   West
## 11 CA-2014-115812 09-06-2014 14-06-2014 United States   West
##           Category    Sales Quantity_sold Total_revenue
## 6        Furniture   48.860             7       342.020
## 7  Office Supplies    7.280             4        29.120
## 8       Technology  907.152             6      5442.912
## 9  Office Supplies   18.504             3        55.512
## 10 Office Supplies  114.900             5       574.500
## 11       Furniture 1706.184             9     15355.656
```

6.Create new columns for “Order_day”,“Order_month” and “Order_year” from “Order.Date”.

```
library(tidyr)
data = data %>% separate(Order.Date, into=c("Order_day","Order_month","Order_year
"), sep='-')
head(data)
```

```
##          Order.ID Order_day Order_month Order_year  Ship.Date
## 6  CA-2014-115812        09          06       2014 14-06-2014
## 7  CA-2014-115812        09          06       2014 14-06-2014
## 8  CA-2014-115812        09          06       2014 14-06-2014
## 9  CA-2014-115812        09          06       2014 14-06-2014
## 10 CA-2014-115812        09          06       2014 14-06-2014
## 11 CA-2014-115812        09          06       2014 14-06-2014
##          Country Region        Category    Sales Quantity_sold
## 6  United States   West       Furniture   48.860             7
## 7  United States   West Office Supplies    7.280             4
## 8  United States   West      Technology  907.152             6
## 9  United States   West Office Supplies   18.504             3
## 10 United States   West Office Supplies  114.900             5
## 11 United States   West       Furniture 1706.184             9
##    Total_revenue
## 6        342.020
## 7         29.120
## 8       5442.912
## 9         55.512
## 10       574.500
## 11     15355.656
```

7.Add +2 to “Quantity” variable of South Region.

```
head(data[data$Region=="South", ])
```

```
##          Order.ID Order_day Order_month Order_year  Ship.Date
## 13 CA-2017-114412        15          04       2017 20-04-2017
## 44 CA-2017-139619        19          09       2017 23-09-2017
## 70 CA-2016-119823        04          06       2016 06-06-2016
## 73 US-2015-134026        26          04       2015 02-05-2015
## 74 US-2015-134026        26          04       2015 02-05-2015
## 75 US-2015-134026        26          04       2015 02-05-2015
##          Country Region        Category   Sales Quantity_sold
## 13 United States  South Office Supplies  15.552             3
## 44 United States  South Office Supplies  95.616             2
```

```
## 70 United States  South Office Supplies  75.880             2
## 73 United States  South       Furniture 831.936             8
## 74 United States  South       Furniture  97.040             2
## 75 United States  South Office Supplies  72.784             1
##    Total_revenue
## 13        46.656
## 44       191.232
## 70       151.760
## 73      6655.488
## 74       194.080
## 75        72.784
```

```
data$Quantity_sold[data$Region == "South"] <- data$Quantity_sold[data$Region == "
South"] + 2
head(data[data$Region=="South", ])
```

```
##          Order.ID Order_day Order_month Order_year  Ship.Date
## 13 CA-2017-114412        15          04       2017 20-04-2017
## 44 CA-2017-139619        19          09       2017 23-09-2017
## 70 CA-2016-119823        04          06       2016 06-06-2016
## 73 US-2015-134026        26          04       2015 02-05-2015
## 74 US-2015-134026        26          04       2015 02-05-2015
## 75 US-2015-134026        26          04       2015 02-05-2015
##          Country Region        Category   Sales Quantity_sold
## 13 United States  South Office Supplies  15.552             5
## 44 United States  South Office Supplies  95.616             4
## 70 United States  South Office Supplies  75.880             4
## 73 United States  South       Furniture 831.936            10
## 74 United States  South       Furniture  97.040             4
## 75 United States  South Office Supplies  72.784             3
##    Total_revenue
## 13        46.656
## 44       191.232
## 70       151.760
## 73      6655.488
## 74       194.080
## 75        72.784
```

WEEK-2

Perform below data pre-processing techniques on the sales dataset.

1. Delete Unnecessary columns
2. Handle missing values
3. Remove duplicate data
4. Create Country, Order_year and Order_Id from Order_Id variable
5. Remove outliers from sales column

```
# load sales dataset
data = read.csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/sales_data_preprocess.csv",fileEnc
oding = "UTF-8-BOM")
#examin the data
head(data)
```

```
##   Row.ID       Order.ID Order.Date  Ship.Date Region        Category
## 1      1 CA-2016-152156 08-11-2016 11-11-2016  South       Furniture
## 2      2 CA-2016-152156 08-11-2016 11-11-2016
## 3      3 CA-2016-138688 12-06-2016 16-06-2016   West Office Supplies
## 4      4 US-2015-108966 11-10-2015 18-10-2015  South       Furniture
## 5      5 US-2015-108966 11-10-2015 18-10-2015  South Office Supplies
## 6      6 CA-2014-115812 09-06-2014 14-06-2014   West
##      Sales Quantity
## 1 261.9600        2
## 2 731.9400        3
## 3  14.6200        2
## 4 957.5775        5
## 5  22.3680        2
## 6  48.8600        7
```

1. Delete Unnecessary columns

```
# Row.ID not required for analysis.Delete Row.ID
data$Row.ID = NULL
head(data)
```

```
##         Order.ID Order.Date  Ship.Date Region        Category    Sales
## 1 CA-2016-152156 08-11-2016 11-11-2016  South       Furniture 261.9600
## 2 CA-2016-152156 08-11-2016 11-11-2016                        731.9400
## 3 CA-2016-138688 12-06-2016 16-06-2016   West Office Supplies  14.6200
## 4 US-2015-108966 11-10-2015 18-10-2015  South       Furniture 957.5775
## 5 US-2015-108966 11-10-2015 18-10-2015  South Office Supplies  22.3680
## 6 CA-2014-115812 09-06-2014 14-06-2014   West                  48.8600
##   Quantity
## 1        2
## 2        3
## 3        2
## 4        5
## 5        2
## 6        7
```

2. Handle missing values

```
#replace blank values with NA
data[data == ""] = NA
head(data)
```

```
##         Order.ID Order.Date  Ship.Date Region        Category    Sales
## 1 CA-2016-152156 08-11-2016 11-11-2016  South       Furniture 261.9600
## 2 CA-2016-152156 08-11-2016 11-11-2016   <NA>            <NA> 731.9400
## 3 CA-2016-138688 12-06-2016 16-06-2016   West Office Supplies  14.6200
## 4 US-2015-108966 11-10-2015 18-10-2015  South       Furniture 957.5775
## 5 US-2015-108966 11-10-2015 18-10-2015  South Office Supplies  22.3680
## 6 CA-2014-115812 09-06-2014 14-06-2014   West            <NA>  48.8600
##   Quantity
## 1        2
## 2        3
## 3        2
## 4        5
## 5        2
## 6        7
```

```
# find the percentage of missing values column wise
missing_percentage = colSums(is.na(data))/nrow(data)*100
print(missing_percentage)
```

```
##   Order.ID Order.Date  Ship.Date     Region   Category      Sales
##   0.000000   0.000000   0.000000   3.890490   5.187320   1.873199
##   Quantity
##   0.000000
```

```
# replace Sales missing values by mean()

#calculate mean of sales
mean_sales = mean(data$Sales, na.rm = TRUE)

#replace by mean
data$Sales = replace(data$Sales, is.na(data$Sales), mean_sales)
```

```
Mode = function(x){
  a = table(x)
  mode_value = names(a[which.max(a)])
  return(mode_value)
}

# replace Region and Category missing values by mode

# find the mode of Region and replace
region_mode = Mode(data$Region)
print(region_mode)
```

```
## [1] "West"
```

```
data$Region = replace(data$Region, is.na(data$Region), region_mode)
```

```r
#find the mode of Category and replace
category_mode = Mode(data$Category)
print(category_mode)

## [1] "Office Supplies"

data$Category = replace(data$Category, is.na(data$Category), category_mode)
```

3. Remove duplicate data

```r
# Using unique() in Base R
dim(data)

## [1] 694   7

data = unique(data)
dim(data)

## [1] 690   7
```

4. Create Country, Order_year and Id from Order_Id variable

```r
library(tidyr)

data = data %>% separate(Order.ID, into = c("Country","Order_year","Id"), sep = "-")
data$Order.ID = NULL
head(data)

##   Country Order_year     Id Order.Date  Ship.Date Region        Category
## 1      CA       2016 152156 08-11-2016 11-11-2016  South       Furniture
## 2      CA       2016 152156 08-11-2016 11-11-2016   West Office Supplies
## 3      CA       2016 138688 12-06-2016 16-06-2016   West Office Supplies
## 4      US       2015 108966 11-10-2015 18-10-2015  South       Furniture
## 5      US       2015 108966 11-10-2015 18-10-2015  South Office Supplies
## 6      CA       2014 115812 09-06-2014 14-06-2014   West Office Supplies
##      Sales Quantity
## 1 261.9600        2
## 2 731.9400        3
## 3  14.6200        2
## 4 957.5775        5
## 5  22.3680        2
## 6  48.8600        7
```

5. Remove outliers from sales column

```r
dim(data)

## [1] 690   9
```

```r
Q1 = quantile(data$Sales, 0.25)
Q3 = quantile(data$Sales, 0.75)

IQR = Q3-Q1

lower_bound = Q1 - 1.5*IQR
upper_bound = Q3 + 1.5*IQR

outliers = data$Sales < lower_bound | data$Sales > upper_bound
print(dim(data[outliers, ]))

#remove outlier rows
data = data[!outliers, ]
dim(data)
```

```
## [1] 690   9

## [1] 80   9

## [1] 610   9
```

WEEK-3

Conduct a complete data analysis on a given student results dataset and derive insights using the ggplot2 package in R.

```
# load sales dataset
data = read.csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/students_marks.csv",fileEncoding =
 "UTF-8-BOM")
#examin the data
head(data)
```

```
##   id    Name Gender Age Section Science English History Maths
## 1  1 Bronnie Female  13       C      21      81      62    49
## 2  2  Lemmie   Male  15       B      29      41      17    40
## 3  3   Danya Female  14       C      12      87      16    96
## 4  4   Denna Female  14       B      15      53      82    33
## 5  5 Jocelin   Male  14       A      43       6       3    21
## 6  6 Malissa Female  14       C      98      51      85    76
```

```
str(data)
```

```
## 'data.frame':    250 obs. of  9 variables:
##  $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name   : Factor w/ 247 levels "Abel","Adah",..: 47 148 68 73 132 157 117 36 62 228
 ...
##  $ Gender : Factor w/ 2 levels "Female","Male": 1 2 1 1 2 1 1 2 2 2 ...
##  $ Age    : int  13 15 14 14 14 14 14 14 15 15 ...
##  $ Section: Factor w/ 3 levels "A","B","C": 3 2 3 2 1 3 2 2 1 3 ...
##  $ Science: int  21 29 12 15 43 98 38 25 39 35 ...
##  $ English: int  81 41 87 53 6 51 74 51 16 25 ...
##  $ History: int  62 17 16 82 3 85 54 41 22 37 ...
##  $ Maths  : int  49 40 96 33 21 76 60 80 49 27 ...
```

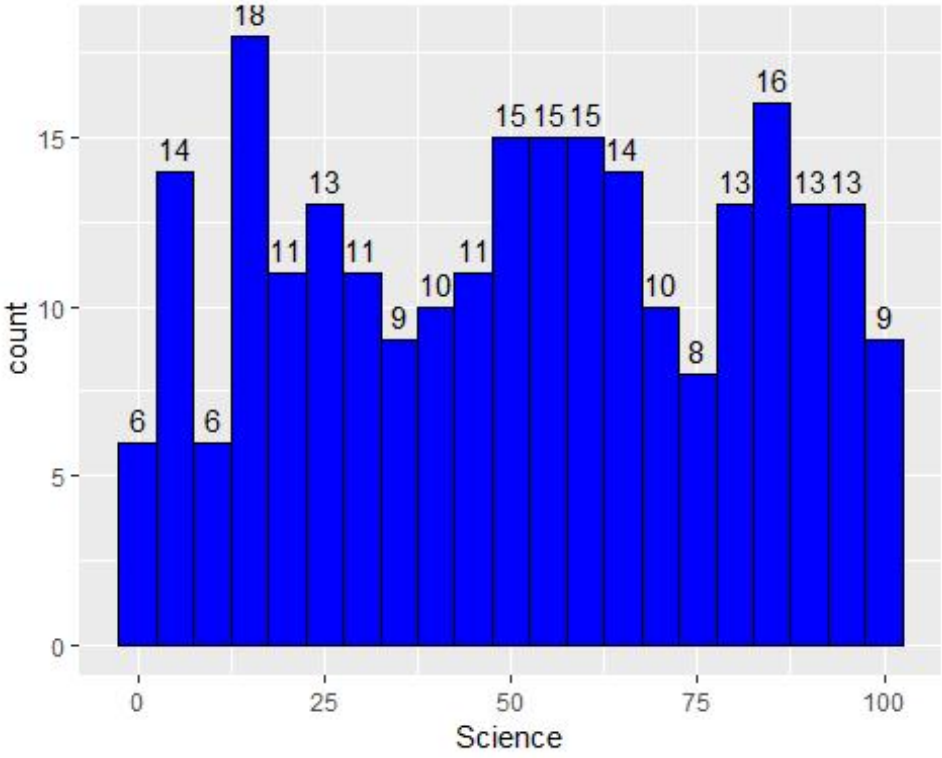
```
# find the percentage of missing values column wise
missing_percentage = colSums(is.na(data))/nrow(data)*100
print(missing_percentage)
```

```
##      id    Name  Gender     Age Section Science English History   Maths
##       0       0       0       0       0       0       0       0       0
```

1. Distribution of Science and English Marks

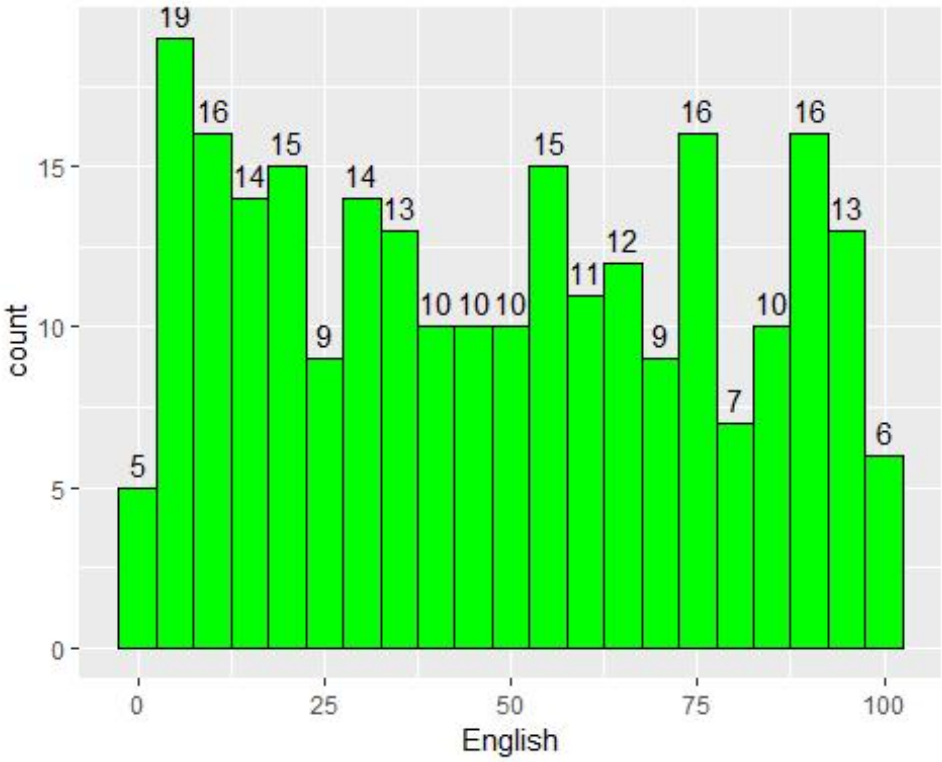
```
library(ggplot2)

# Assuming 'data' is your dataframe
ggplot(data, aes(x = Science)) +
  geom_histogram(binwidth = 5, fill = 'blue', color = 'black') +
  stat_bin(binwidth = 5, geom = "text", aes(label = ..count..), vjust = -0.5) +
  ggtitle("Distribution of Science Marks")
```

"Distribution of Science Marks"




```
ggplot(data, aes(x = English)) +
  geom_histogram(binwidth = 5, fill = 'green', color = 'black') +
  stat_bin(binwidth = 5, geom = "text", aes(label = ..count..), vjust = -0.5) +
  ggtitle("Distribution of English Marks")
```

Distribution of English Marks




```
##
```

Answer the below questions from the above histogram plots:

1. How many students are there with science marks > 75 (approximately)?

2. How many students are there with English marks > 75 (approximately)?

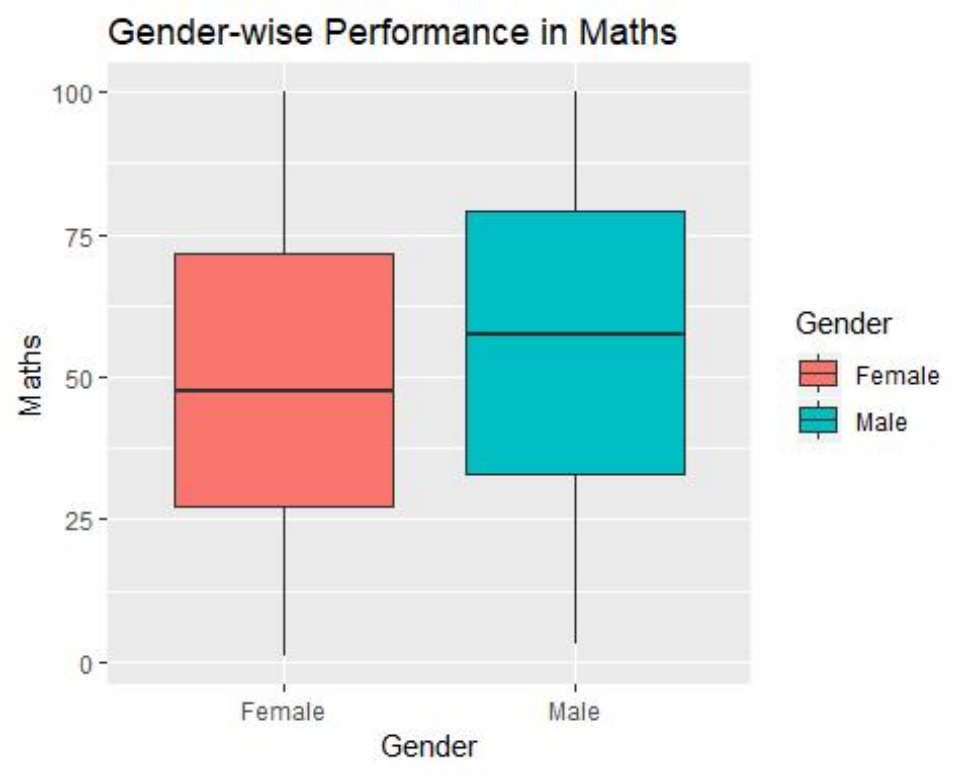
3. How many students are there with science marks < 35 (approximately)?

2. Gender-wise Performance of Maths and History marks

```
ggplot(data, aes(x = Gender, y = History, fill = Gender)) +
  geom_boxplot() +
  ggtitle("Gender-wise Performance in History")
```


```
ggplot(data, aes(x = Gender, y = Maths, fill = Gender)) +
  geom_boxplot() +
```

```
  ggtitle("Gender-wise Performance in Maths"
```

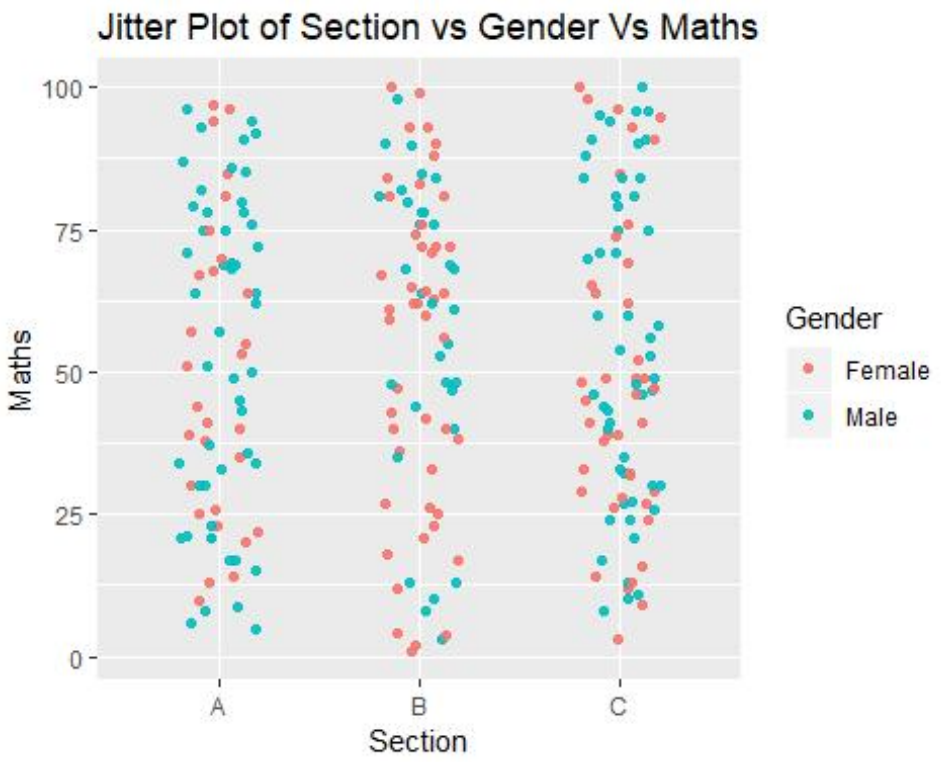


Answer the below questions from above box plots:

1. Which gender has the highest average math score?
2. Are there any outliers in the math marks?
3. Which gender performed well in the math exam?

3. Section and gender wise Performance of maths subject

```
ggplot(data, aes(x = Section, y = Maths, color = Gender)) +
  geom_jitter(width = 0.2, height = 0.1, alpha = 0.9) +
  labs(title = "Jitter Plot of Section vs Gender Vs Maths",
       x = "Section",
       y = "Maths")
```

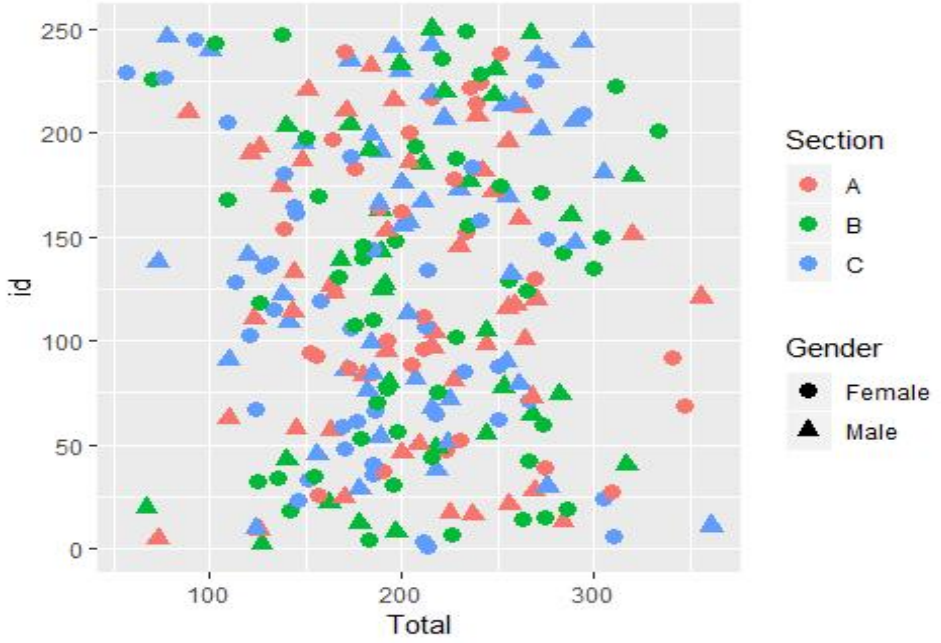


Answer the below questions from above jitter plot:

1. Draw jitter plot for remaining subjects also.
2. Which gender from what section performed well in the math,science,english and History exams?

4. Calculate total marks and analyze them with id,section and gender

```
library(dplyr)

# create total column

data = mutate(data, Total = Maths + Science + English + History)
head(data)

##   id    Name Gender Age Section Science English History Maths Total
## 1  1 Bronnie Female  13       C      21      81      62    49   213
## 2  2  Lemmie   Male  15       B      29      41      17    40   127
## 3  3   Danya Female  14       C      12      87      16    96   211
## 4  4   Denna Female  14       B      15      53      82    33   183
## 5  5 Jocelin   Male  14       A      43       6       3    21    73
## 6  6 Malissa Female  14       C      98      51      85    76   310

ggplot(data, aes(x = Total,y = id, shape = Gender, color = Section)) +
  geom_point(size = 3)
```

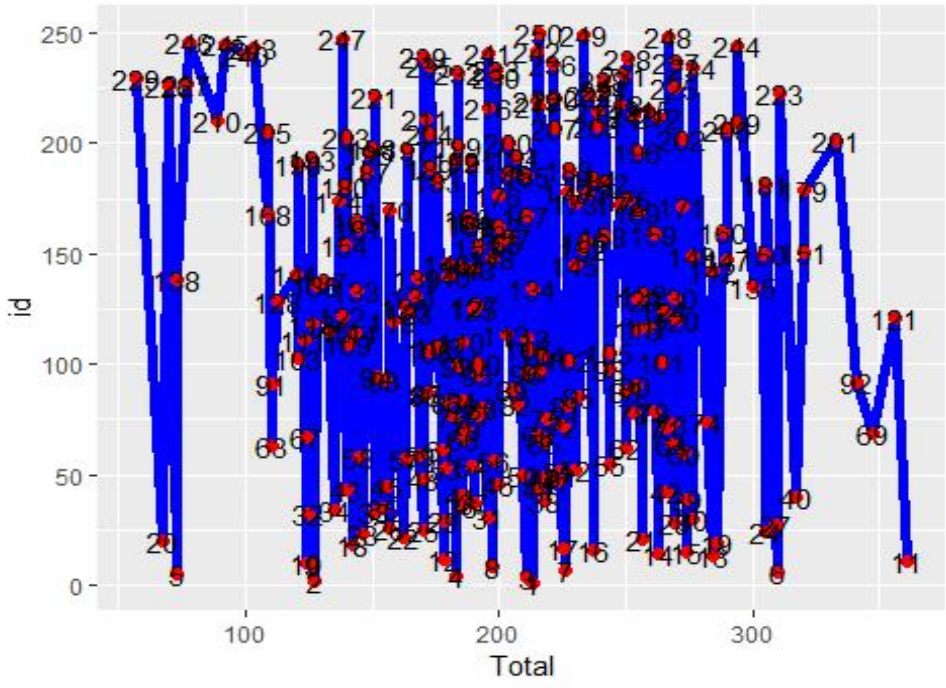


Answer the below questions from above scatter plot:

1. student from which section and gender got the highest total marks.
2. student from which section and gender got the least total marks.

5. Line plot between id and total marks

```
ggplot(data, aes(x = Total, y = id)) +
  geom_line(size = 2, color = "blue") +
  geom_point(color = "red", size = 2) +
  geom_text(aes(label=id))
```



Answer the below questions from above line plot:

What is the ID of the student who got the highest marks?
What is the ID of the student who got the least marks?

WEEK-5

Merge two Data Frames and apply various data manipulation techniques.

Merge two Data Frames

```
import pandas as pd
# read the files
data1 = pd.read_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/salesdata.csv")
data1.head()
```

	Order ID	Order Date	Ship Date	Customer ID	Country	City	State	Postal Code	Region	Category	Sales	Quantity	Discount	Profit
0	CA-2016-152156	08-11-2016	11-11-2016	CG-12520	United States	Henderson	Kentucky	42420	South	Furniture	261.9600	2.0	0.00	41.9136
1	CA-2016-152156	08-11-2016	11-11-2016	CG-12520	United States	Henderson	Kentucky	42420	South	Furniture	731.9400	3.0	0.00	219.5820

```
data2 = pd.read_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/returnsdata.csv")
data2.head()
```

	Returned	Order ID
0	Yes	CA-2017-153822
1	Yes	CA-2017-129707
2	Yes	CA-2014-152345
3	Yes	CA-2015-156440
4	Yes	US-2017-155999

```
# merging two dataframes using inner join
data = pd.merge(data1, data2, on='Order ID', how='inner')
data.head()
```

	Order ID	Order Date	Ship Date	Customer ID	Country	City	State	Postal Code	Region	Category	Sales	Quantity	Discount	Profit	Returne
0	CA-2014-143336	27-08-2014	01-09-2014	ZD-21925	United States	San Francisco	California	94109	West	Office Supplies	8.56	2.0	0.0	2.4824	Ye
1	CA-2014-143336	27-08-2014	01-09-2014	ZD-21925	United States	San Francisco	California	94109	West	Technology	213.48	3.0	0.2	16.0110	Ye

```
data.shape
```

```
(104, 15)
```

Different data manipulation techniques

1. Delete rows

```
# Delete 2nd and 41th rows
data = data.drop([1,40])
data.shape
```

```
(102, 15)
```

2.Delete columns 'Customer ID', 'Postal Code'.

```
data = data.drop(['Customer ID', 'Postal Code'], axis=1)
data.head()
```

	Order ID	Order Date	Ship Date	Country	City	State	Region	Category	Sales	Quantity	Discount	Profit	Returned
0	CA-2014-143336	27-08-2014	01-09-2014	United States	San Francisco	California	West	Office Supplies	8.56	2.0	0.0	2.4824	Yes
2	CA-2014-143336	27-08-2014	01-09-2014	United States	San Francisco	California	West	NaN	22.72	4.0	0.2	7.3840	Yes
3	CA-2016-	17-06-	18-06-	United	Troy	New York	East	Office	208.56	6.0	0.0	52.1400	Yes

3. Modify the values

```
# Round the 'Profit' column to 2 decimal places
data['Profit'] = data['Profit'].round(2)
data.head()
```

	Order ID	Order Date	Ship Date	Country	City	State	Region	Category	Sales	Quantity	Discount	Profit	Returned
0	CA-2014-143336	27-08-2014	01-09-2014	United States	San Francisco	California	West	Office Supplies	8.56	2.0	0.0	2.48	Yes
2	CA-2014-143336	27-08-2014	01-09-2014	United States	San Francisco	California	West	NaN	22.72	4.0	0.2	7.38	Yes
3	CA-2016-	17-06-	18-06-	United	Troy	New York	East	Office	208.56	6.0	0.0	52.14	Yes

4.Create new column from existing columns

```
# Create 'Price_per_Unit' column
data['Price_per_Unit'] = data['Sales'] / data['Quantity']

# Extract the year from 'Order ID'
data['OrYear'] = df['Order ID'].str.split('-').str[1]
data.head()
```

	Order ID	Order Date	Ship Date	Country	City	State	Region	Category	Sales	Quantity	Discount	Profit	Returned	Price_per_Unit
0	CA-2014-143336	27-08-2014	01-09-2014	United States	San Francisco	California	West	Office Supplies	8.56	2.0	0.0	2.4824	Yes	4.280
2	CA-2014-143336	27-08-2014	01-09-2014	United States	San Francisco	California	West	NaN	22.72	4.0	0.2	7.3840	Yes	5.680

5. Handle missing data

```
import numpy as np
# replace blank strings with 'NaN'
data = data.replace('',np.nan)

# calculate % of missing values columnwise
missing_percentage = data.isna().sum()/len(data)*100
missing_percentage
```

```
Order ID        0.000000
Order Date      0.000000
Ship Date       0.000000
Country         0.000000
City            0.000000
State           0.000000
Region          0.000000
Category        6.862745

Sales           0.000000
Quantity        5.882353
Discount        0.000000
Profit          0.000000
Returned        0.000000
Price_per_Unit  5.882353
dtype: float64
```

```
# fill the missing values of Category,Quantity and Price_per_Unit columns
data['Category'] = data['Category'].fillna(data['Category'].mode()[0])
data['Quantity'] = data['Quantity'].fillna(data['Quantity'].mean())
data['Price_per_Unit'] = data['Price_per_Unit'].fillna(data['Price_per_Unit'].mean())

# calculate % of missing values columnwise
missing_percentage = data.isna().sum()/len(data)*100 missing_percentage
```

```
Order ID        0.0
Order Date      0.0
Ship Date       0.0
Country         0.0
City            0.0
State           0.0
Region          0.0
Category        0.0
Sales           0.0
Quantity        0.0
Discount        0.0
Profit          0.0
Returned        0.0
Price_per_Unit  0.0
dtype: float64
```

```
data.shape
```

```
(102, 14)
```

6. Remove duplicate entries

```
data = data.drop_duplicates()
data.shape
```

```
(102, 14)
```

No duplicates rows

WEEK-6

Use the Python ‘Matplotlib’ to perform a thorough data analysis and extract insights from a given Housing dataset.

```
import pandas as pd
# read the Housing dataset
data =
pd.read_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/Housing.cs
v")

data.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	furnishing
0	6195000	5500	3	2	4	yes	yes	no	no	yes	1	semi-fu
1	6195000	6350	3	2	3	yes	yes	no	no	yes	0	fu
2	6195000	5500	3	2	1	yes	yes	yes	no	no	2	fu
3	6160000	4500	3	1	4	yes	no	no	no	yes	0	unfu
4	6160000	5450	4	2	1	yes	no	yes	no	yes	0	semi-fu

```
# check the
shape of
dataset
data.shape
```

```
(299, 12)
```

```
data.info()
```

```
<class
'pandas.c
ore.frame.
DataFrame
'>
RangeInde
x: 299
entries,
0 to 298
Data
columns
(total 12
columns):
 #   Column            Non-Null Count  Dtype
--  ------------------------ -----
 0   price             299 non-null    int64
 1   area              299 non-null    int64
 2   bedrooms          299 non-null    int64
 3   bathrooms         299 non-null    int64
 4   stories           299 non-null    int64
 5   mainroad          299 non-null    object
 6   guestroom         299 non-null    object
 7   basement          299 non-null    object
 8   hotwaterheating   299 non-null    object
 9   airconditioning   299 non-null    object
 10  parking           299 non-null    int64
 11  furnishingstatus  299 non-null    object
d
t
```

```
y
p
e
s
:

i
n
t
6
4
(
6
)
,

o
b
j
e
c
t
(
6
)

m
e
m
o
r
y

u
s
a
g
e
:

2
8
.
2
+

K
B
```

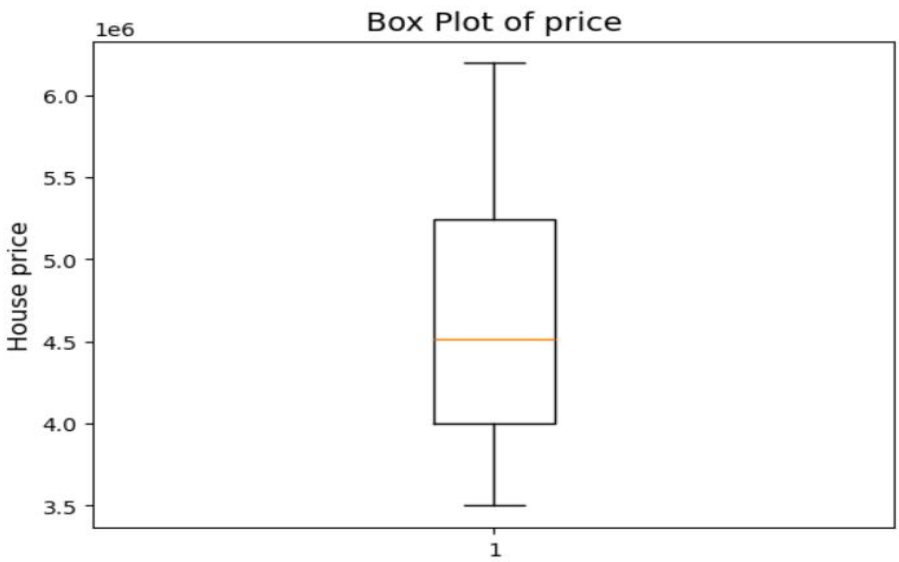
check the missing values

data.isnull().sum()

```
price               0
area                0
bedrooms            0
bathrooms           0
stories             0
mainroad            0
guestroom           0
basement            0
hotwaterheating     0
airconditioning     0
parking             0
furnishingstatus    0
dtype: int64
```

1. Box plot for price

```
import matplotlib.pyplot as
plt

# Create box plot for the 'price'
column
plt.boxplot(data['price'])

# Add title and labels
plt.title('Box Plot of price',
fontsize=14)
plt.ylabel('House price',
fontsize=12)

# Display the plot
plt.show()
```



Average house price = 4500000

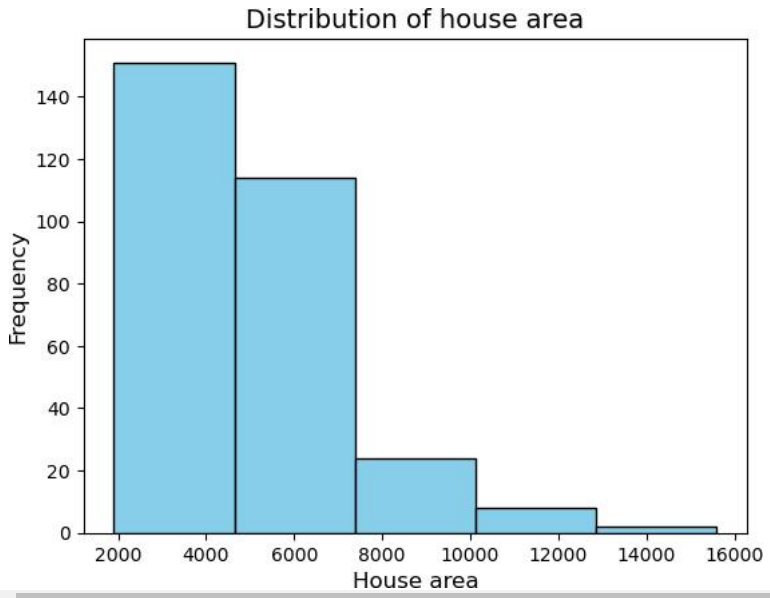
There are no outliers

range of house price = around 4000000 to 5400000

2. Histogram for area

```
#Create histogram for area
```

```
plt.hist(data['area'], bins=5, edgecolor='black',
color='skyblue')

# Add labels and title
plt.title('Distribution of house area',
fontsize=14) plt.xlabel('House area',
fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Display the plot
plt.show()
```

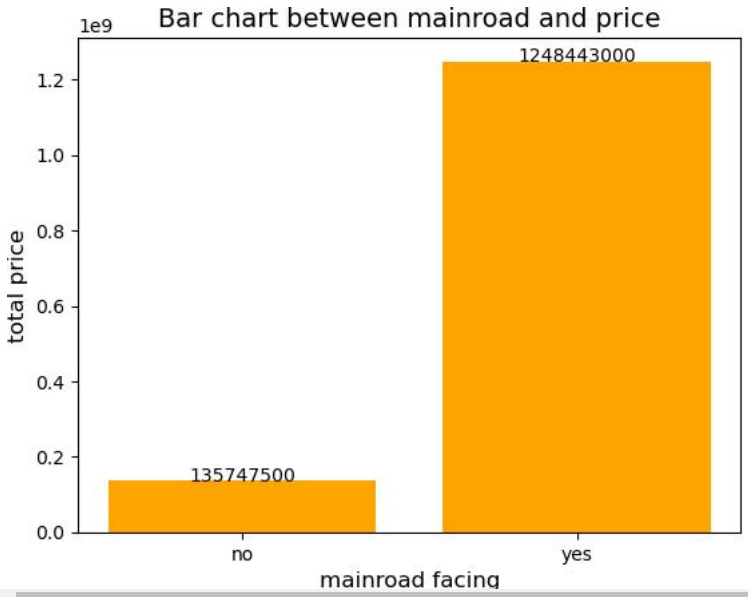


- *most of the house area is in the range from 2000 sqft to 7200 sqft*

3. Bar chart between mainroad and price

```
# Group data by 'mainroad' and sum the price
grouped_data = data.groupby('mainroad')['price'].sum()
plt.bar(grouped_data.index, grouped_data.values,
color='orange')
# Add labels and title
plt.xlabel('mainroad facing',
fontsize=12) plt.ylabel('total
price', fontsize=12)
plt.title('Bar chart between mainroad and price',
fontsize=14)
# Add data labels on top of the bars
```

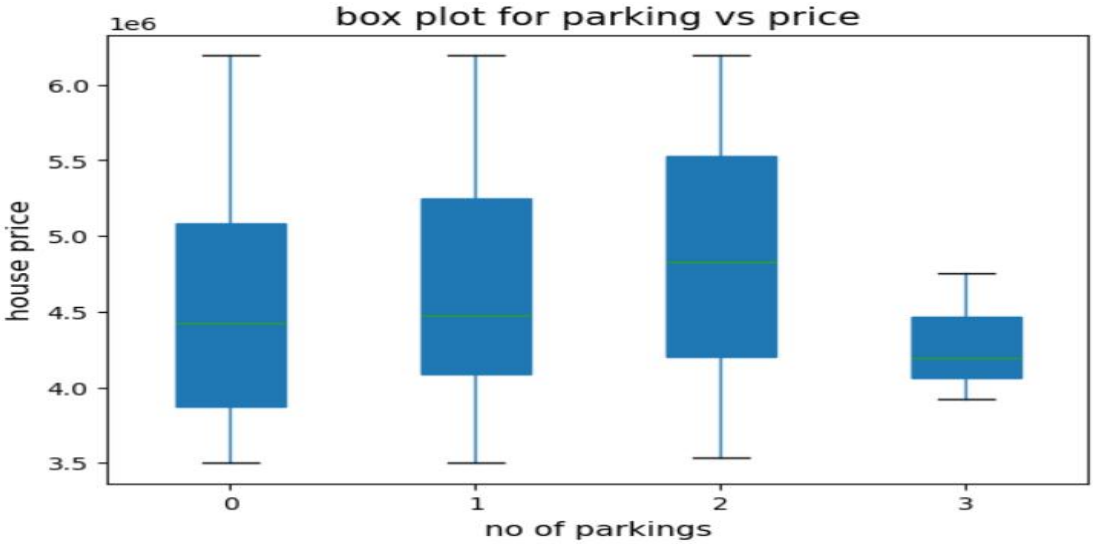
```
for i, value in enumerate(grouped_data.values):
  plt.text(i, value, str(value), ha='center', fontsize=10)
# Display
the plot
plt.sho
w()
```



- *The houses facing the main road are the most expensive.*

4. box plot for parking vs price

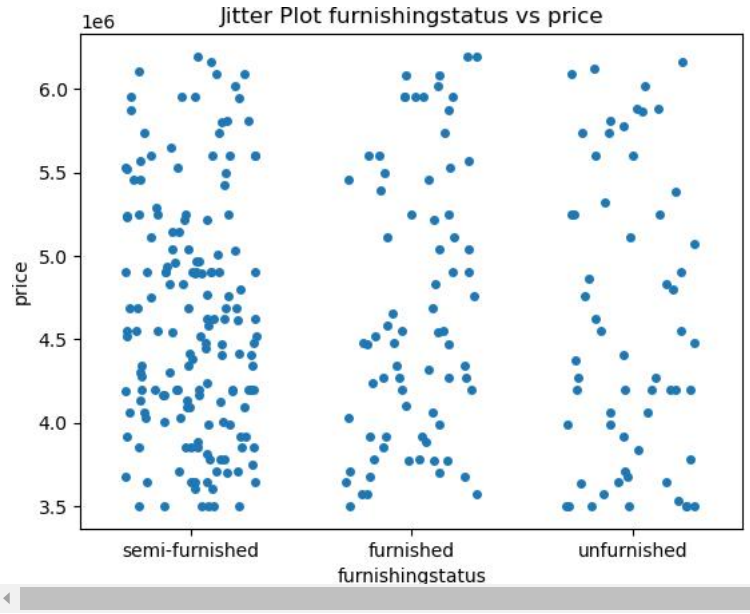
```
# Create box plot for Sales grouped by Region
data.boxplot(column='price', by='parking', grid=False,
patch_artist=True)

# Add title and labels
plt.title('box plot for parking vs price', fontsize=14)
# Remove default 'Boxplot grouped by Region'
plt.suptitle('')
plt.xlabel('no of parkings',
fontsize=12)
 plt.ylabel('house price',
fontsize=12)
# Display the plot
plt.show()
```



- *The houses with 2 parking spaces are the most expensive.*

5. jitter plot for furnishingstatus vs price

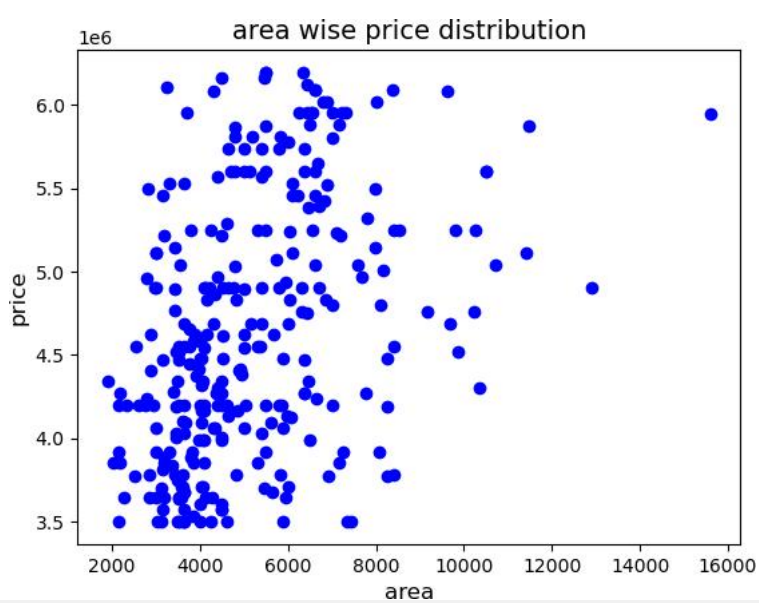
```
import seaborn as sns
import matplotlib.pyplot as plt

# Create a jitter plot Region vs Sales
sns.stripplot(x=data['furnishingstatus'], y=data['price'],
jitter=0.3)

# Add labels and title
plt.xlabel('furnishingstatus')
plt.ylabel('price')
plt.title('Jitter Plot furnishingstatus vs
price')

# Show the plot
plt.show()
```



No insights

6. scatter plot between area and price

```
# Create scatter plot
plt.scatter(data['area'], data['price'], color='blue')

# Add labels and title
plt.title('area wise price distribution', fontsize=14)
plt.xlabel('area', fontsize=12)
plt.ylabel('price', fontsize=12)

# Display the plot
plt.show()
```



- *There exists a bit positive relation between area and price*

7. subplots among guestroom vs basement vs price

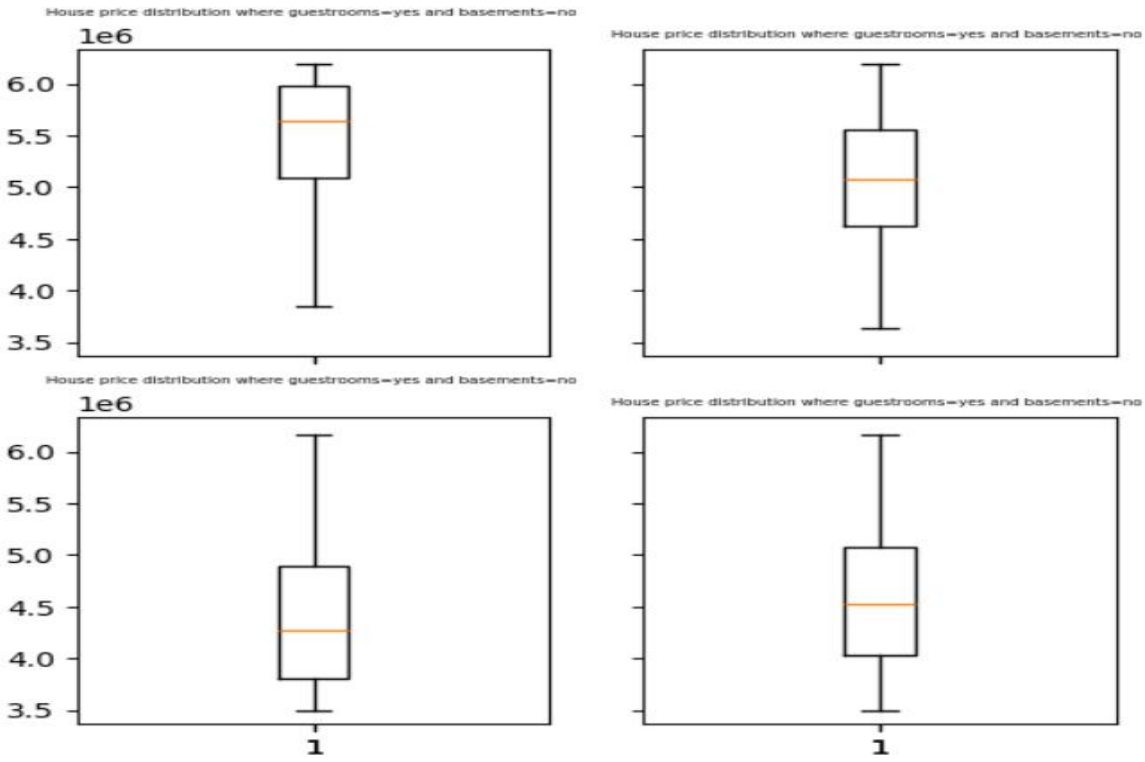
```
data.guestroom.unique()
```

```
array(['yes', 'no'], dtype=object)
```

```
data.basement.unique()
```

```
array(['no', 'yes'], dtype=object)
```

```
import matplotlib.pyplot as plt

# Create a figure with four subplots sharing both x and y axes
fig, axes = plt.subplots(2, 2, sharex=True, sharey=True,
figsize=(10, 10))

# Get the unique regions from the data
guestrooms = data['guestroom'].unique()
basements = data['basement'].unique()

# Plot sales by country for each region
for i, x in enumerate(guestrooms):
    for j, y in enumerate(basements):
        g_data = data[(data['guestroom'] == x) &
(data['basement'] == y)]
        axes[i,j].boxplot(g_data['price'])
        axes[i][j].set_title(f'House price distribution where
                guestrooms={x} and basements={y} ',size = 8)

# Display the plots
plt.show()
```


House price distribution where guestrooms=yes and basements=no
1e6
6.0
5.5
5.0
4.5
4.0
3.5
House price distribution where guestrooms=yes and basements=no
House price distribution where guestrooms=yes and basements=no
1e6
6.0
5.5
5.0
4.5
4.0
3.5
1
House price distribution where guestrooms=yes and basements=no
1