BRIDE BENSON
S2 CSBS
ROLL NO:20

R PROGRAMMING CODES AND OUTPUT

#1.5

```
Fibonacci <-c(1:10)
Fibonacci[1]<-1
Fibonacci[2]<-1
for(i in 3:10) Fibonacci[i]<-Fibonacci[i-2]+Fibonacci[i-1]
print("First 10 Fibonacci Numbers:")
Fibonacci
```

Output

```
[1] "First 10 Fibonacci Numbers:"
 [1]  1  1  2  3  5  8 13 21 34 55

[Execution complete with exit code 0]
```

#1.6

```
prime_numbers <- function(n) {
if (n >= 2)
{
 x = seq(2, n)
 prime_nums = c()
 for (i in seq(2, n))
 {
 if (any(x == i))
 {
  prime_nums = c(prime_nums, i)
  x = c(x[(x %% i) != 0], i)
 }
}
return(prime_nums)
}
else
{
 stop("Number should be at least 2.")
}
}
```

prime_numbers(15)

Output

```
[1]  2  3  5  7 11 13

[Execution complete with exit code 0]
```

#1.7

```
factors<-function(x)
{
 print(paste("The factors of ",x,"are :"))
 for(i in 1:x)
 {
  if(x%%i==0)
  {
   print(i)
  }
 }
}
factors(9)
```

Output

```
[1] "The factors of  9 are :"
[1] 1
[1] 3
[1] 9

[Execution complete with exit code 0]
```

#1.9

```
date<-as.Date("2003/01/01")
n=365
olddate<-date-n
newdate<-date+n
print(paste("Original date is: ",date))
print(paste("Subtracted date is: ",olddate))
print(paste("Added date is: ",newdate))
```

Output

```
[1] "Original date is:  2003-01-01"
[1] "Subtracted date is:  2002-01-01"
[1] "Added date is:  2004-01-01"

[Execution complete with exit code 0]
```

#1.10

```
date1<-as.Date("2004/12/21")
date2<-as.Date("2002/10/23")
print(paste("Date 1 is: ",date1))
print(paste("Date 2 is: ",date2))
difftime(date1,date2,units="days")
```

Output

```
[1] "Date 1 is:  2004-12-21"
[1] "Date 2 is:  2002-10-23"
Time difference of 790 days

[Execution complete with exit code 0]
```

#2.1

```
vect=c(155,261,132000,423.4,321,137000,105,240,118000,157.64,260,139000)
row.names<-c("Stock Prices","Employees","Revenue")
column.names<-c("2018","2019")
matrix.names<-c("Apple","Microsoft")
result<-array(c(vect),dim=c(3,2,2),dimnames=list(row.names,column.names,matrix.name
s))
print(result)
```

Output

```
, , Apple

               2018     2019
Stock Prices    155    423.4
Employees       261    321.0
Revenue      132000 137000.0

, , Microsoft

               2018      2019
Stock Prices    105    157.64
Employees       240    260.00
Revenue      118000 139000.00


[Execution complete with exit code 0]
```

#2.2(cont of 2.1)

```
diff_price<-result[1,2,1]-result[1,1,1]
avg<-(result[1,2,1]+result[1,1,1])/2
perc_diff<-100*(diff_price/avg)
print('The percentage difference is :')
perc_diff
print('The 2018 Apple Stock Price is :' )
result["Stock Prices","2018","Apple"]
```

```
[1] "The percentage difference is :"
[1] 92.80775
[1] "The 2018 Apple Stock Price is :"
[1] 155

[Execution complete with exit code 0]
```

#2.3

```
row.names=c("R1 ","R2", "R3","R4","R5")
col.names=c("C1","C2","C3")
array<-array(seq(from=50,length.out=15,by=2),dim=c(5,3,1),dimnames=list(row.names,c
ol.names))
```

array

Output

```
, , 1

    C1 C2 C3
R1  50 60 70
R2  52 62 72
R3  54 64 74
R4  56 66 76
R5  58 68 78


[Execution complete with exit code 0]
```

#2.4

```
matrix1<-matrix(1:20,nrow=5,ncol=4)
print("Matrix 1: 5x4 ")
matrix1

mr1<-c("Row1","Row2","Row3")
mc1<-c("Col1","Col2","Col3")
matrix2<-matrix(1:9,nrow=3,ncol=3,byrow=TRUE,dimnames=list(mr1,mc1))
print("Matrix 2: 3x3 with labels and filled by rows ")
matrix2

mr2<-c("Row1","Row2")
mc2<-c("Col1","Col2")
matrix3<-matrix(1:4,nrow=2,ncol=2,byrow=FALSE,dimnames=list(mr2,mc2))
print("Matrix 3: 2x2 with labels and filled by columns ")
matrix3
```

Output

```
[1] "Matrix 1: 5x4 "
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20
[1] "Matrix 2: 3x3 with labels and filled by
rows "
     Col1 Col2 Col3
Row1    1    2    3
Row2    4    5    6
Row3    7    8    9
[1] "Matrix 3: 2x2 with labels and filled by
columns "
     Col1 Col2
Row1    1    3
Row2    2    4

[Execution complete with exit code 0]
```

#2.5

```
Matrix = matrix(1:9, nrow = 3)
print(Matrix)
M2 = Matrix
for (i in 1:nrow(M2))
{
  for (j in 1:ncol(M2))
  {
    M2[i, j] <- Matrix[j, i]
  }
}
print(M2)
```

Output

```
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

[Execution complete with exit code 0]
```

#2.6

```
m <- matrix(1:9, nrow=3)
m
n <- matrix(9:17, nrow=3)
n
matrix_mul<-m %*% n
print(matrix_mul)
```

Output

```
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

[Execution complete with exit code 0]
```

#3.1

```
x <- list("1", "2", "3","4","5","6")
print("The original list is ")
print(x)
result = rev(x)
print("The reversed list is")
print(result)
```

Output

```
[1] "The original list is "
[[1]]
[1] "1"

[[2]]
[1] "2"

[[3]]
[1] "3"

[[4]]
[1] "4"

[[5]]
[1] "5"

[[6]]
[1] "6"

[1] "The reversed list is"
[[1]]
[1] "6"

[[2]]
[1] "5"

[[3]]
[1] "4"

[[4]]
[1] "3"

[[5]]
[1] "2"
```

```
[[6]]
[1] "1"


[Execution complete with exit code 0]
```

#3.2

```
sqdata<-list(1,2,3,4,5,6,7,8,9,10)
sqdata
squared<-unlist(sqdata)
for(i in 1:10) squared[i]<-squared[i]^2
squared
```

Output

```
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] 3

[[4]]
[1] 4
```

```
[[5]]
[1] 5

[[6]]
[1] 6

[[7]]
[1] 7

[[8]]
[1] 8
```

```
[[9]]
[1] 9

[[10]]
[1] 10

 [1]   1   4   9  16  25  36  49  64  81 100

[Execution complete with exit code 0]
```

#3.3

```
nestedlist<-list(list(0,3), list(2,6), list(4,9))
print("Original nested list:")
nestedlist
extract<-lapply(nestedlist,'[[',2)
print("Second element of the nested list:")
extract
```

#3.4

```
list1 = list(2,3,1,0)
list2 = list("A", "S", "D","T","S")
print("Original lists: ")
list1
list2
merged<-c(list1, list2)
print("Merged lists: ")
merged
```

#3.5

```
vlist<-list(3,6,9)
vlist
ulist<-unlist(vlist)
ulist
```

#3.6

```
det_emp=data.frame(Name=c("Christian","Anastasia","Paige", "Alicia","Zayn"),
                   Age=c(30,28,25,26,24),
                   Gender=c("M","F","F","F","M"),
                   Designation=c("C.E.O.","Manager","H.R.","Receptionist","Salesman"),

Contact=c("0000000000","1111111111","3333333333","6666666666","9999999999")
)
print("Details of the employees:")
```

```
det_emp
summary(det_emp)
```

Output

```
[1] "Details of the employees:"
       Name Age Gender  Designation    Contact
1 Christian  30      M       C.E.O. 0000000000
2 Anastasia  28      F      Manager 1111111111
3     Paige  25      F         H.R. 3333333333
4    Alicia  26      F Receptionist 6666666666
5      Zayn  24      M     Salesman 9999999999
     Name                Age           Gender
Designation
 Length:5           Min.   :24.0   Length:5
Length:5
 Class :character   1st Qu.:25.0   Class
:character   Class :character
 Mode  :character   Median :26.0   Mode
:character   Mode  :character
                    Mean   :26.6
                    3rd Qu.:28.0
                    Max.   :30.0
   Contact
 Length:5
 Class :character
 Mode  :character
```

#3.7

```
sem1_data = data.frame(
  Name = c("personA", "personB", "personC", "personD", "personE"),
  sCGPA = c(10, 9, 4, 8, 6),
  Attempts = c(3, 2, 1, 3, 1 ),
  Pass = c("yes", "yes", "no", "yes", "yes")
)
print("Original dataframe:")
sem1_data
print("Statistical summary and nature of the data of the said dataframe:")
summary(sem1_data)
str(sem1_data)
```

Output

```
[1] "Original dataframe:"
     Name sCGPA Attempts Pass
1 personA    10        3  yes
2 personB     9        2  yes
3 personC     4        1   no
4 personD     8        3  yes
5 personE     6        1  yes
[1] "Statistical summary and nature of the data
of the said dataframe:"
     Name               sCGPA         Attempts
Pass
Length:5
 Class :character   1st Qu.: 6.0   1st Qu.:1
Class :character
 Mode  :character   Median : 8.0   Median :2
Mode  :character
                    Mean   : 7.4   Mean   :2
                    3rd Qu.: 9.0   3rd Qu.:3
                    Max.   :10.0   Max.   :3
'data.frame':   5 obs. of  4 variables:
 $ Name    : chr  "personA" "personB" "personC"
"personD" ...
 $ sCGPA   : num  10 9 4 8 6
$ Attempts: num  3 2 1 3 1
$ Pass    : chr  "yes" "yes" "no" "yes" ...

[Execution complete with exit code 0]
```

#3.8

```
sem1_data = data.frame(
  Name = c("Nobody", "Somebody", "Anybody", "Everybody", "Somebody"),
  SCGPA = c(10, 9, 4, 8, 6),
  Attempts = c(3, 2, 3, 3, 1 ),
  Pass = c("yes", "yes", "no", "yes", "yes")
)
print("Original dataframe:")
sem1_data
```

```
extractdf<-sem1_data[c(3,5),c(1,3)]
print("Extracted data:")
extractdf
```

Output

```
[1] "Original dataframe:"
     Name sCGPA Attempts Pass
1 personA    10        3  yes
2 personB     9        2  yes
3 personC     4        1   no
4 personD     8        3  yes
5 personE     6        1  yes
[1] "Extracted data:"
     Name Attempts
3 personC        1
5 personE        1
```

```
#3.9
df<-data.frame("c1"=c(1,2,3,4),"c2"=c(5,6,7,8))
df
df[nrow(df)+1,]<-c(9,10)
df
```

Output

```
  c1 c2
1  1  5
2  2  6
3  3  7
4  4  8
  c1 c2
1  1  5
2  2  6
3  3  7
4  4  8
5  9 10
```

```
#3.10
data<-data.frame(Name=c("martha","beth","mark"),
                 SGPA=c(10,9.4,9.8),Pass=c("Y","Y","Y"))
print("Original dataframe")
data
```

```
data<-data[with(data,order(SGPA)),]
print("Sorted dataframe is :")
data
```

Output

```
[1] "Original dataframe"
    Name SGPA Pass
1 martha 10.0    Y
2   beth  9.4    Y
3   mark  9.8    Y
[1] "Sorted dataframe is :"
    Name SGPA Pass
2   beth  9.4    Y
3   mark  9.8    Y
1 martha 10.0    Y

[Execution complete with exit code 0]
```

#3.11

```
data<-read.csv("empdata.csv")
data
print("The max salary is :")
sal <- max(data$salary)
print(sal)

print("The Person with max salary is")
val <- subset(data, salary == max(salary))
print(val)

print("People in IT department")
val <- subset( data, dept == "IT")
print(val)

print("People joined on or after 2014")
val <- subset(data,as.Date(start_date) > as.Date("2014-01-01"))
print(val)

write.csv(val,"output.csv")
print("Output.csv :")
newdata <- read.csv("output.csv")
print(newdata)
```

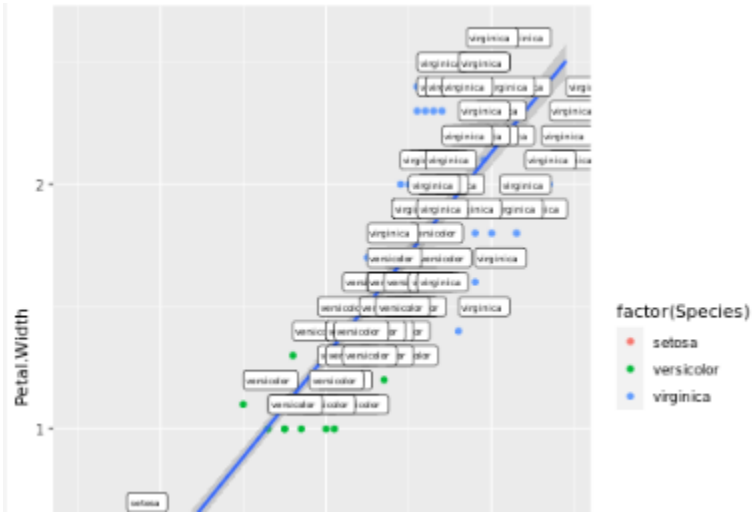
```
7  7 Anisha  932.8 2013-07-31 Operations
>
> print("People in IT department")
[1] "People in IT department"
> val <- subset( data, dept == "IT")
> print(val)
  id     name salary start_date dept
1  1  Shubham  613.3 2012-01-01   IT
3  3 Vaishali   63.0 2014-11-15   IT
6  6    Sumit  588.0 2013-05-21   IT
>
> print("People joined on or after 2014")
[1] "People joined on or after 2014"
> val <- subset(data,as.Date(start_date) > as.Date("2014-01-01"))
> print(val)
  id     name salary start_date    dept
3  3 Vaishali  63.00 2014-11-15      IT
4  4   Nishka 749.00 2014-05-11      HR
5  5   Gunjan 863.25 2015-03-27 Finance
8  8    Akash 712.50 2014-06-17  Financ
>
> write.csv(val,"output.csv")
> print("Output.csv :")
```

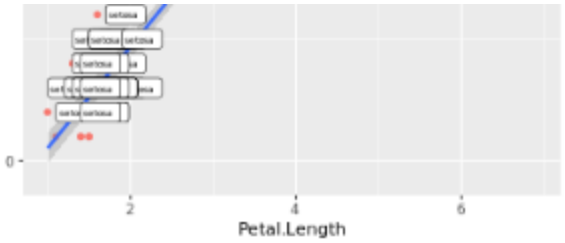
#4.1

```
data("iris")
library(ggplot2)
ggplot(iris,aes(x=Petal.Length,y=Petal.Width))+geom_point(aes(color=factor(Species)))
+geom_smooth(method="lm")+geom_label(aes(label=Species,hjust=0),nudge_y=0.1,size
=2)
```

Output

```
`geom_smooth()` using formula 'y ~ x'

[Execution complete with exit code 0]
```



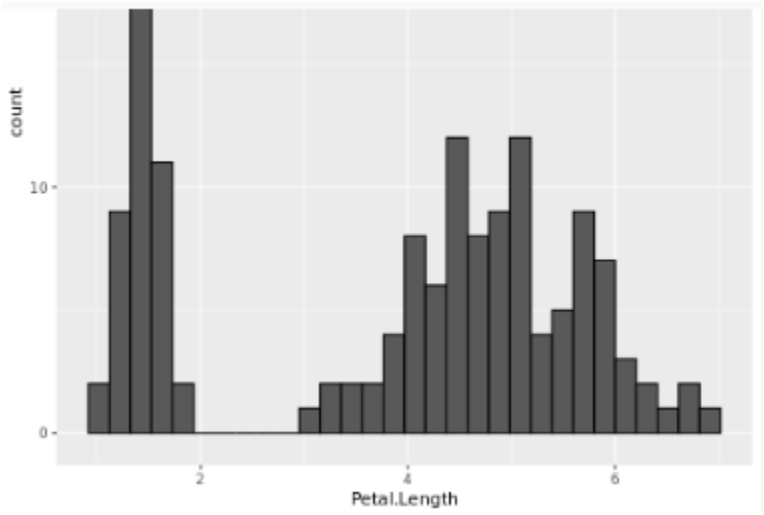


#4.2

```
library(ggplot2)
ggplot(iris,aes(x=Petal.Length))+geom_histogram(color="Black")
```

Output

```
`stat_bin()` using `bins = 30`. Pick better
value with `binwidth`.

[Execution complete with exit code 0]
```

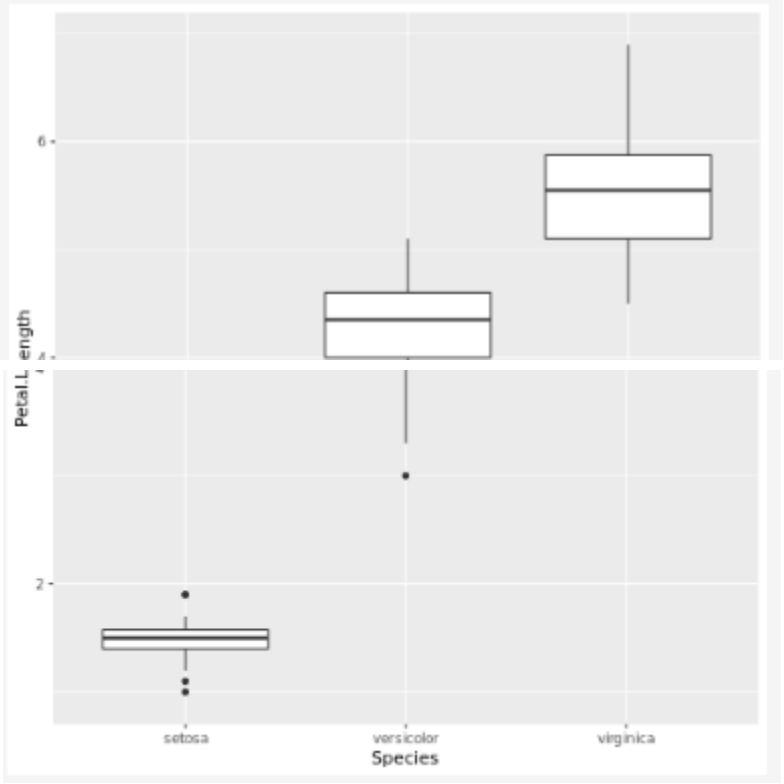


#4.3

```
library(ggplot2)
ggplot(iris,aes(Species,Petal.Length))+geom_boxplot()
```

Output

```
[Execution complete with exit code 0]
```



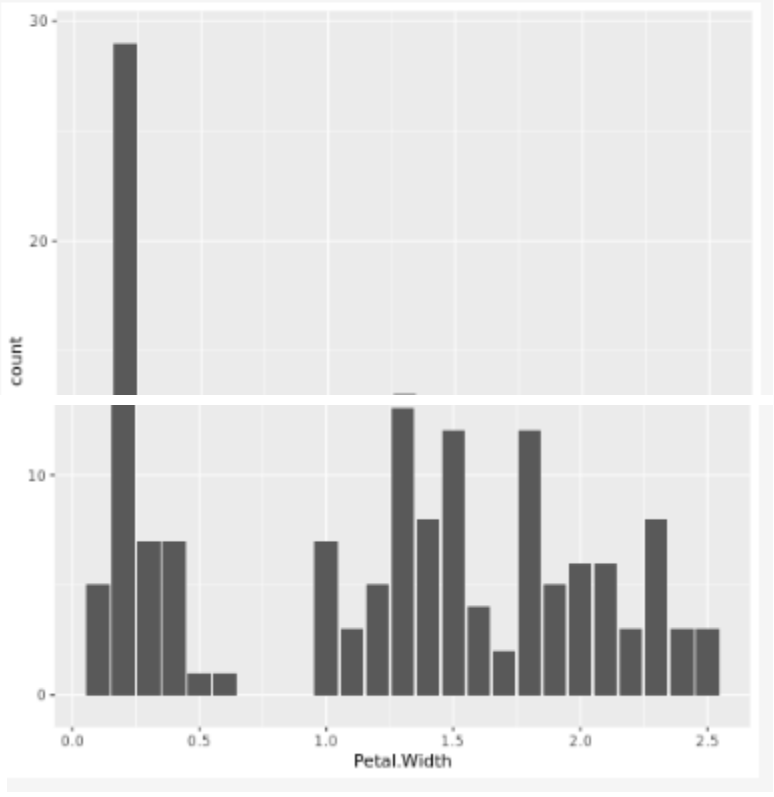
#4.4

```
library(ggplot2)
```

ggplot(iris, aes(x = Petal.Width)) + geom_bar()

Output

[Execution complete with exit code 0]



[install.packages("ggplot2")
install.packages("dplyr")
install.packages("broom")
install.packages("ggpubr")

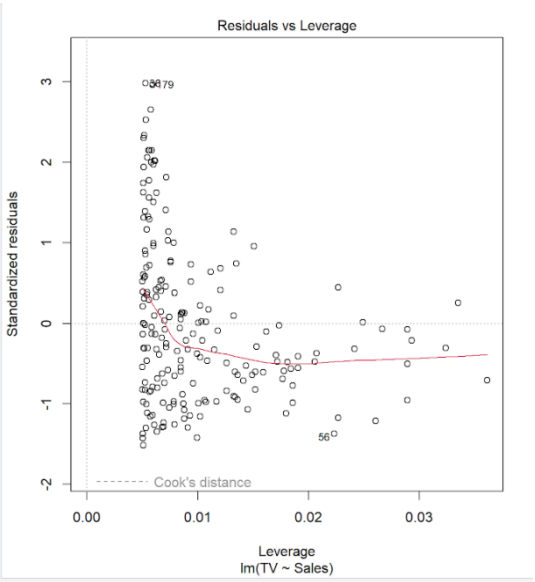
library(ggplot2)
library(dplyr)
library(broom)
library(ggpubr)]

#5.1
sales <- read.csv("tvmarketing.csv")
print(sales)
summary(sales)
dim(sales)

ggplot(sales, aes(x = TV, y = Sales)) +

```
  geom_point() +
  stat_smooth(method = lm)

model <- lm(Sales ~ TV, data = sales)
model

summary(model)
…..
library(ggplot2)
tv<-read.csv("tv.csv")
data(tv)
relation <- lm(Sales~TV,data=tv)
relation
plot(relation)
```

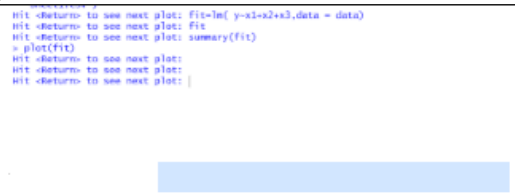
```
Hit <Return> to see next plot: fit=lm(TV~Sales,data = data)
Hit <Return> to see next plot: fit
> summary(fit)

Call:
lm(formula = TV ~ Sales, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-81.06 -40.12 -11.14  27.75 159.39

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -33.4502    10.8969   -3.07  0.00244 **
Sales        12.8717     0.7285   17.67  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.62 on 198 degrees of freedom
Multiple R-squared:  0.6119,    Adjusted R-squared:  0.6099
F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16

> plot(fit)
Hit <Return> to see next plot:
```

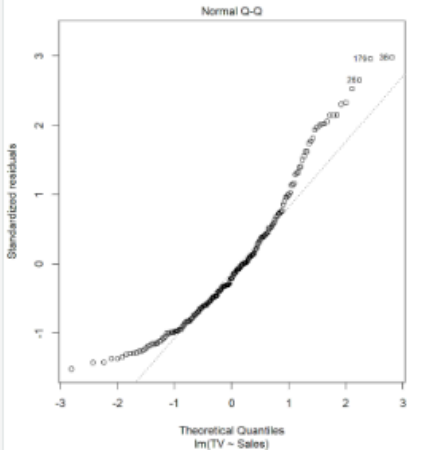


#5.2

```
job <- read.csv("jobprof.csv")
print(job)
summary(job)
dim(job)

ggplot(job, aes(x = y, y = x1)) +
  geom_point() +
  stat_smooth(method = lm)

ggplot(job, aes(x = y, y = x2)) +
  geom_point() +
  stat_smooth(method = lm)
```

```
ggplot(job, aes(x = y, y = x3)) +
  geom_point() +
  stat_smooth(method = lm)

model <- lm(y ~ x1 + x2 + x3, data = job)
model
summary(model)
…..
library(ggplot2)
jobprof<-read.csv("jobprof.csv")
data(jobprof)
model <- lm(y~x1+x2+x3,data=jobprof)
model
plot(model)
```





BRIDE BENSON
S2 CSBS
ROLL NO:20