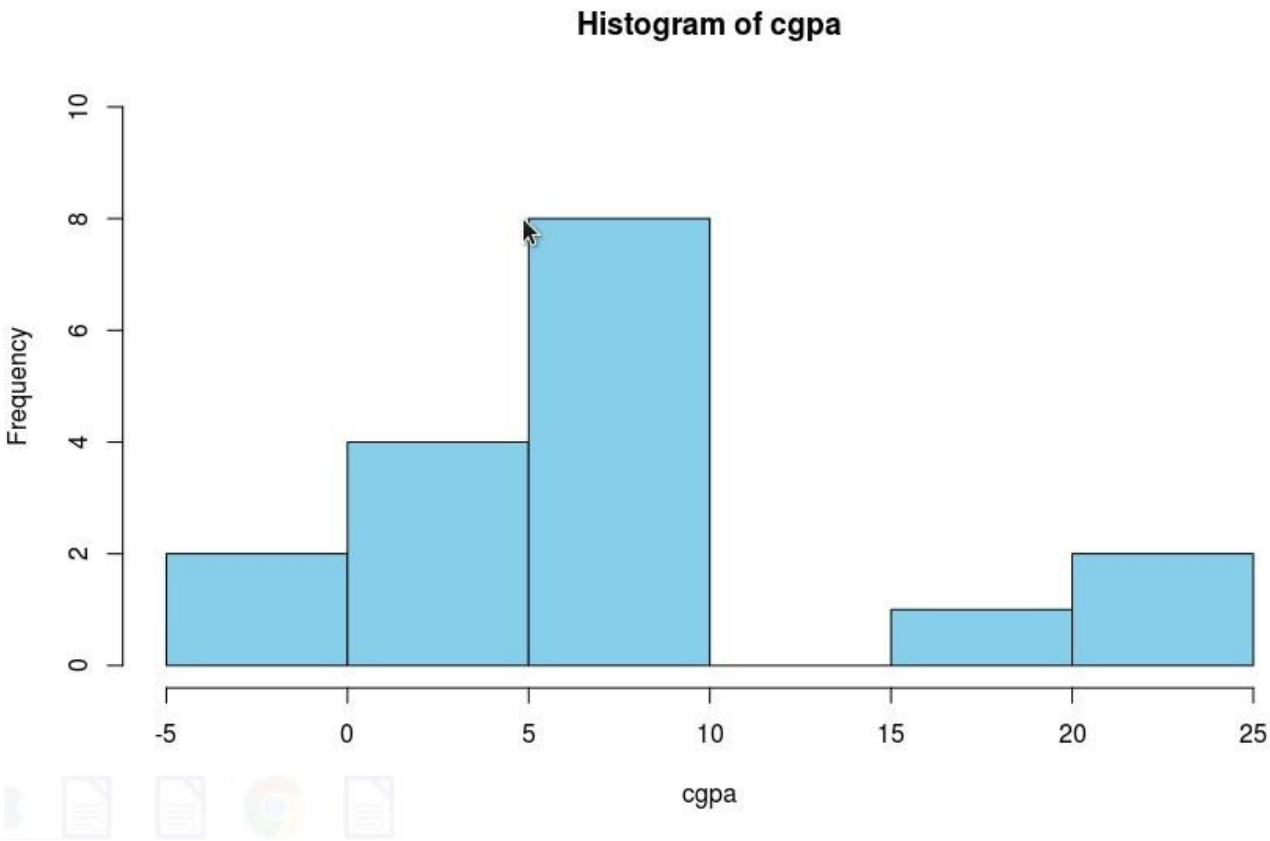
Reg. No: 16BCE0789

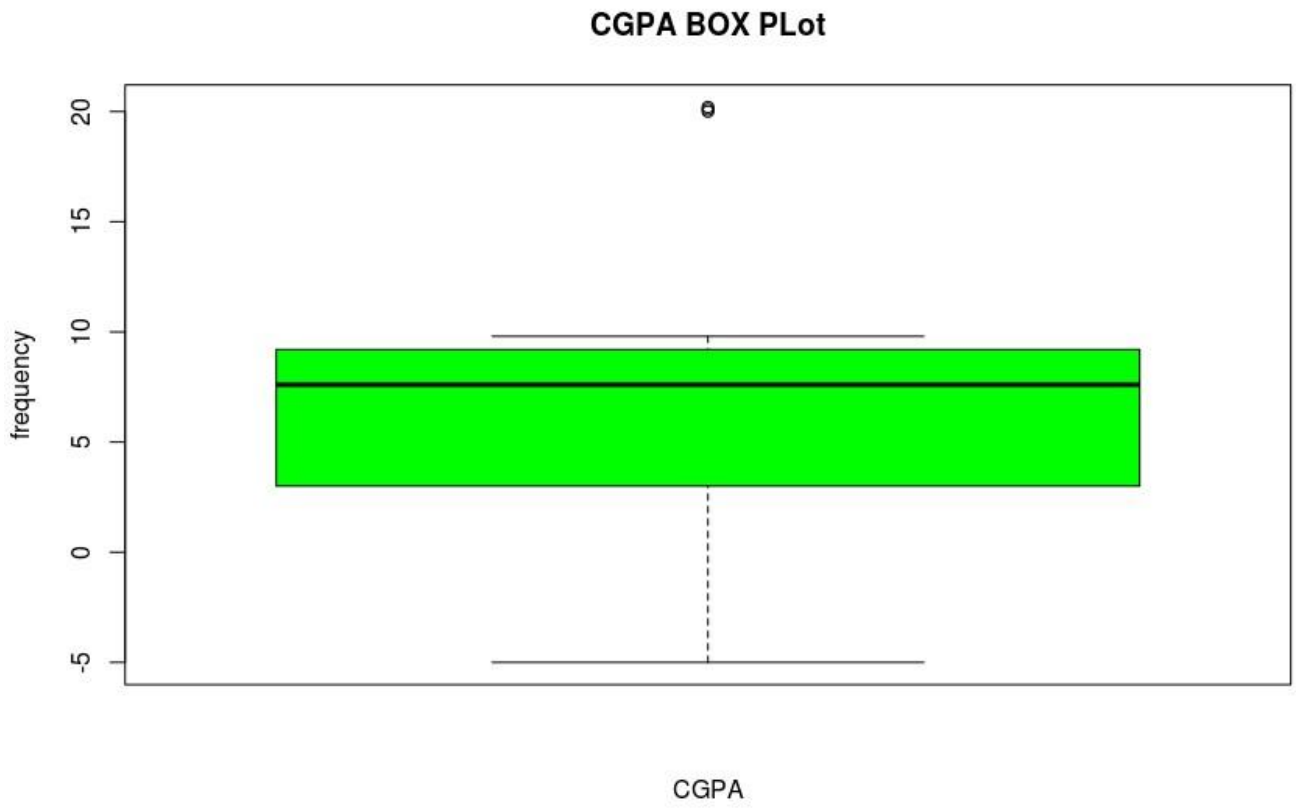
Program 1:

Data - CGPA

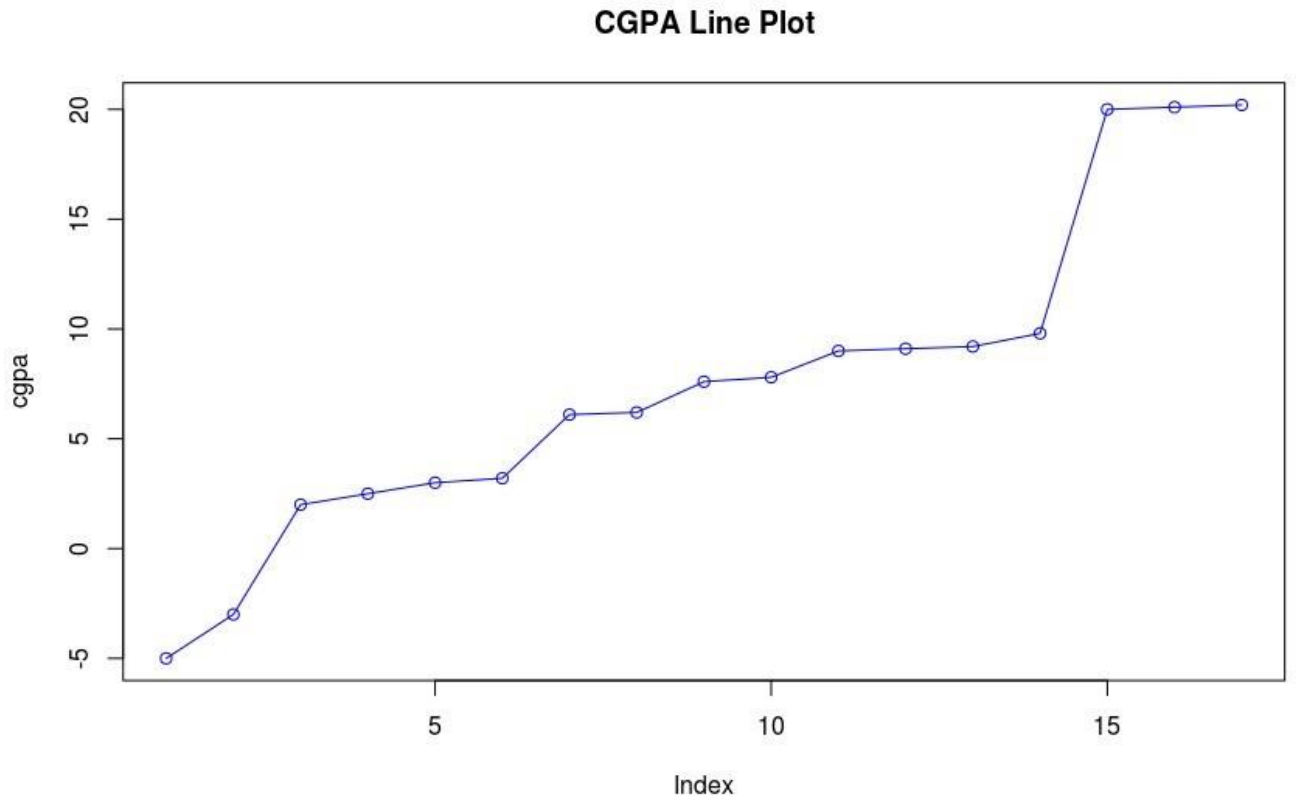
```
> cgpa<-c(-5.0,-
3.0,2.0,2.5,3.0,3.2,6.1,6.2,7.6,7.8,9,9.1,9.2,9.8,20,20.1,20.2)
> hist(cgpa,col="skyblue", xlim=c(-5,25), ylim=c(0,10), breaks=5)
```



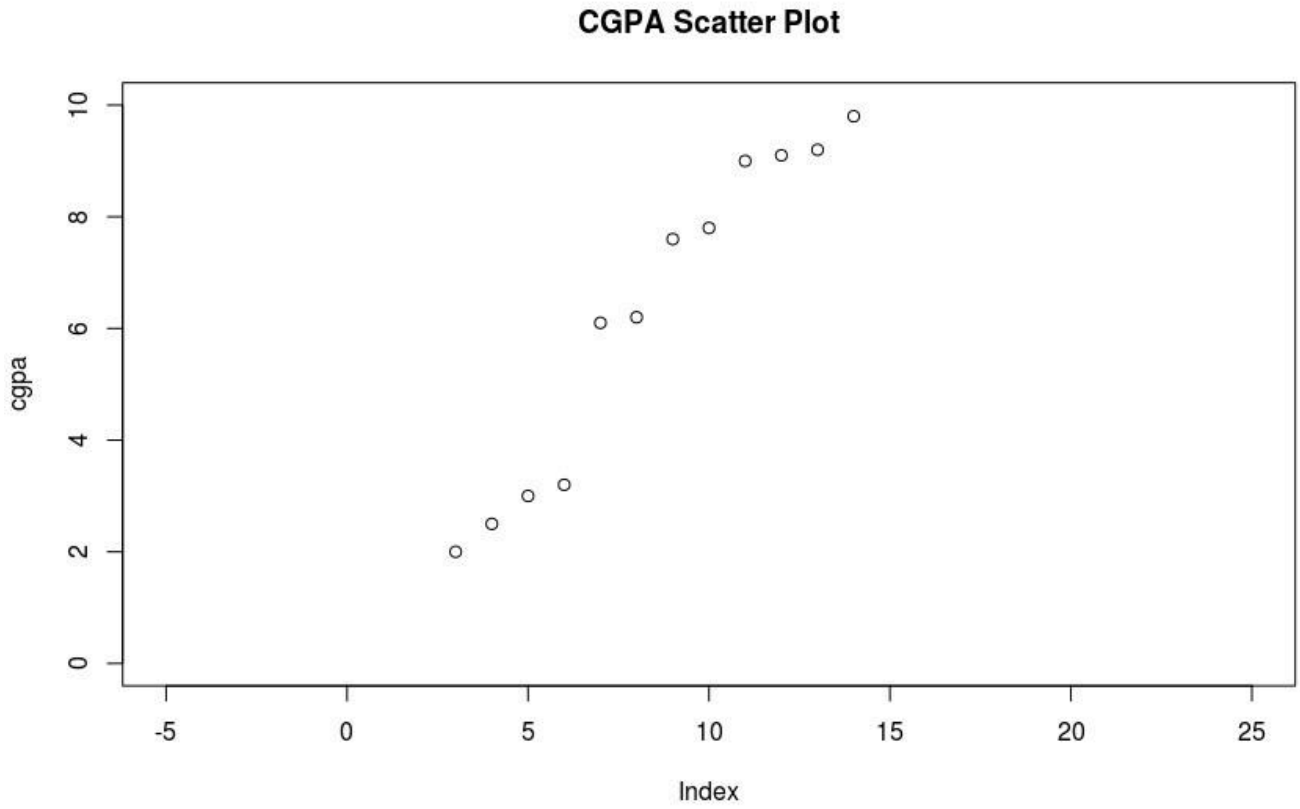

```
> boxplot(cgpa, xlab = "CGPA",ylab = "frequency", main = "CGPA BOX
PLot", varwidth=TRUE,
+         col="green")
```




```
> plot(cgpa, type="o", col="blue", main="CGPA Line Plot")
```

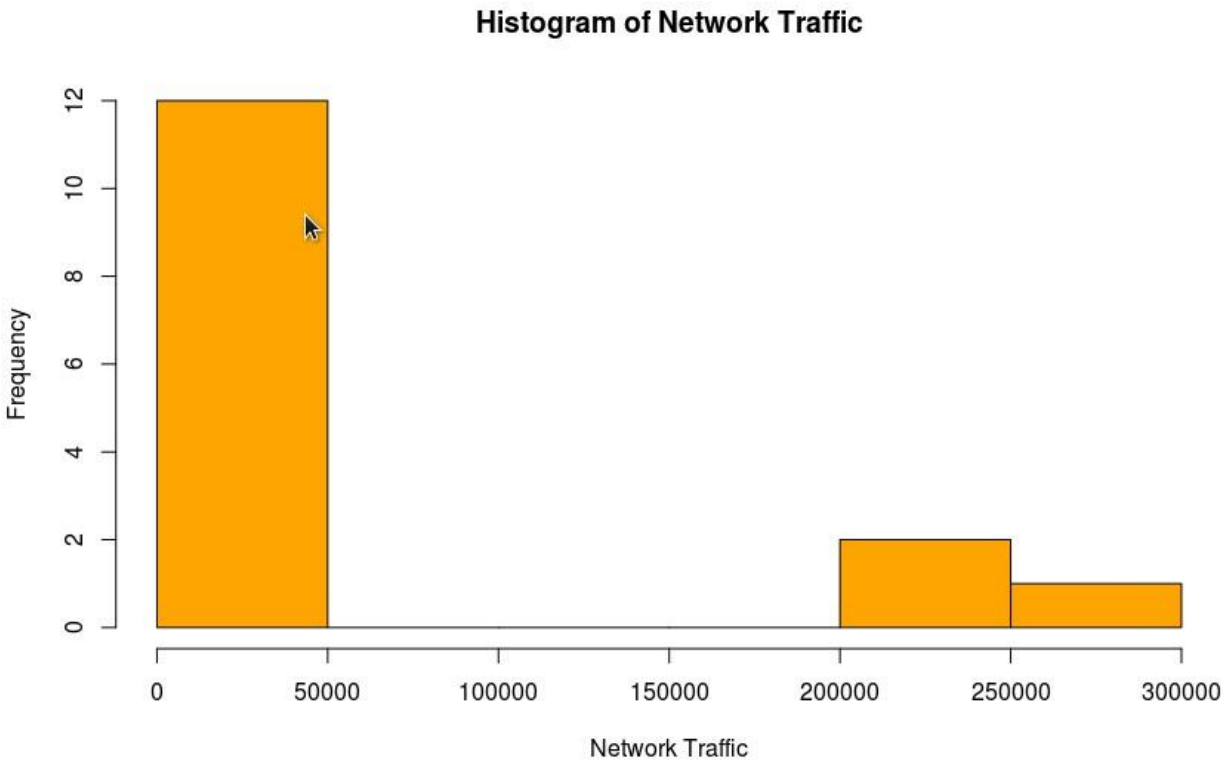



```
> plot(x = cgpa, xlim = c(-5,25), ylim = c(0,10), main = "CGPA Scatter
Plot")
```

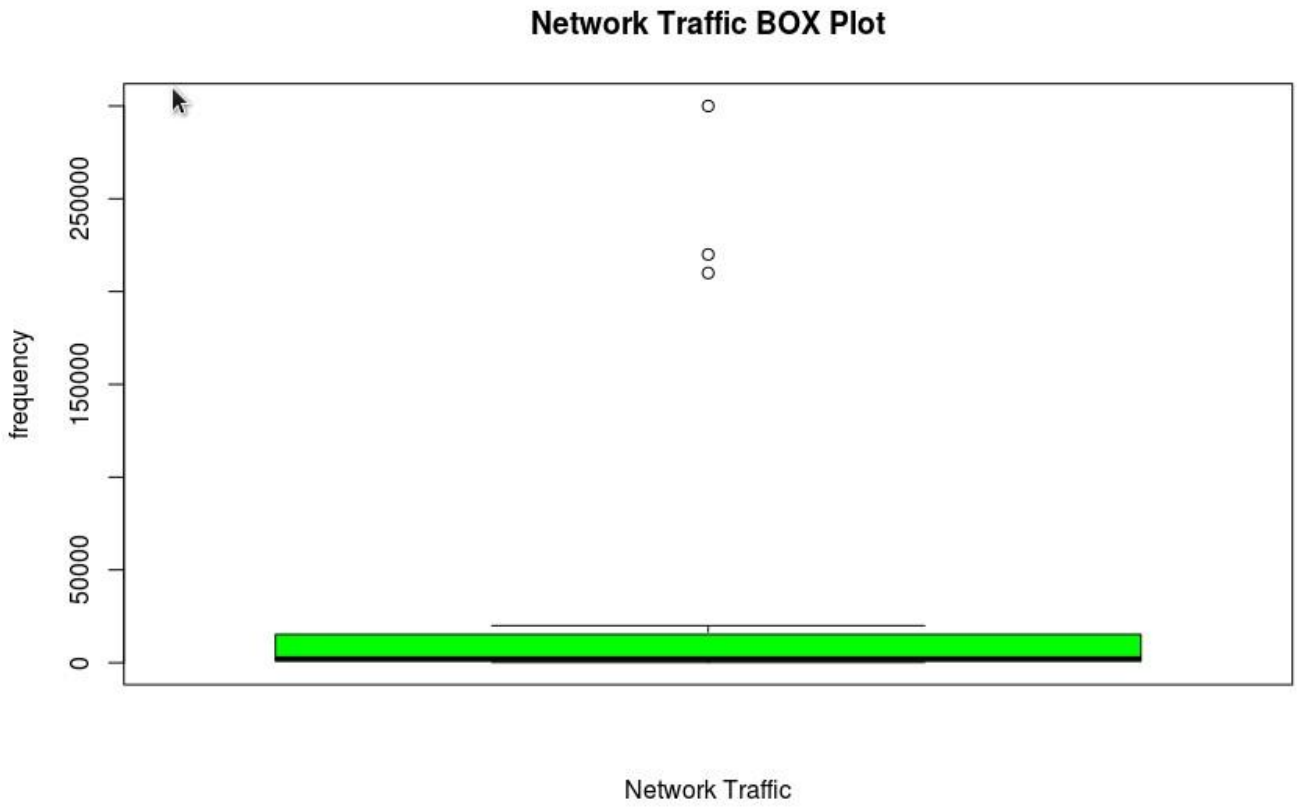


Data – Network Traffic

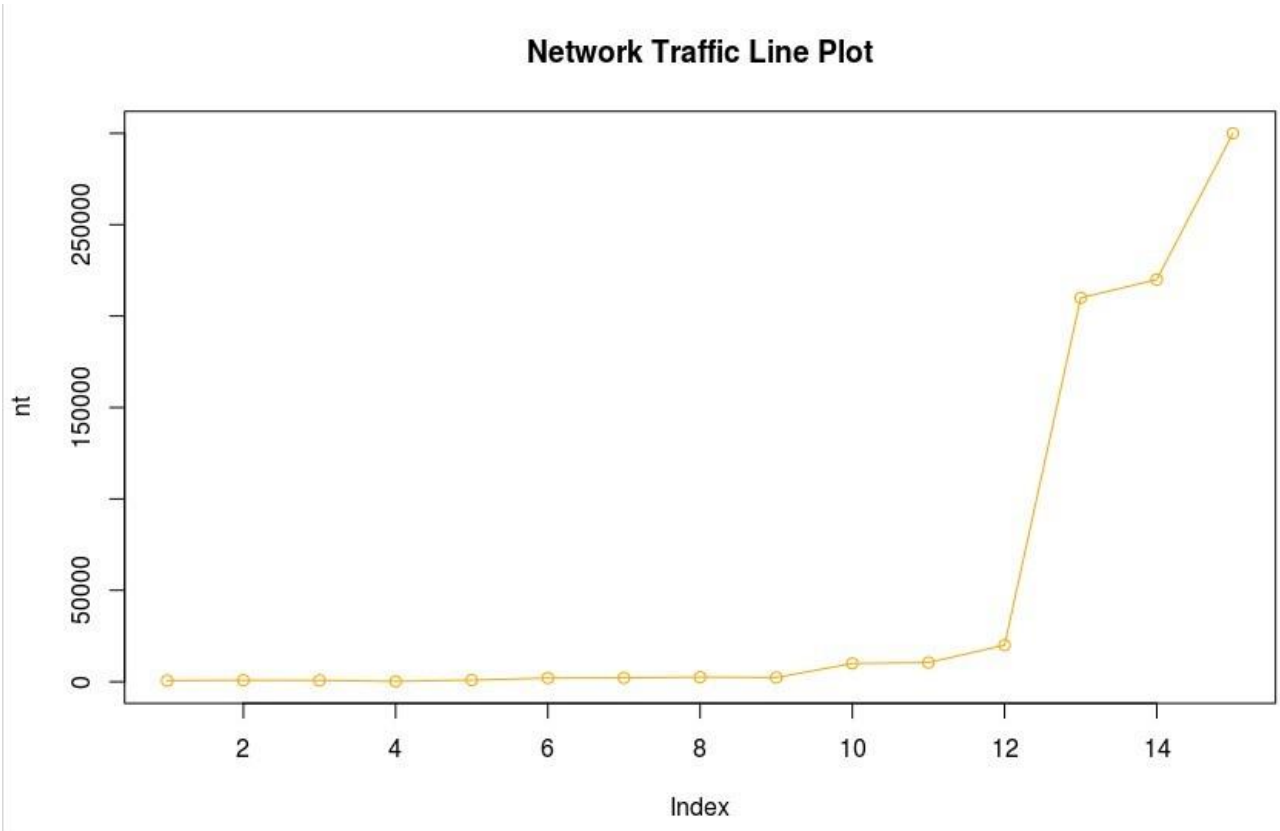
```
> nt <-
c(600,800,700,200,900,2000,2100,2500,2300,10000,10500,20000,210000,220
000,300000)
> hist(nt,col="orange", xlab="Network Traffic", main="Histogram of
Network Traffic")
```



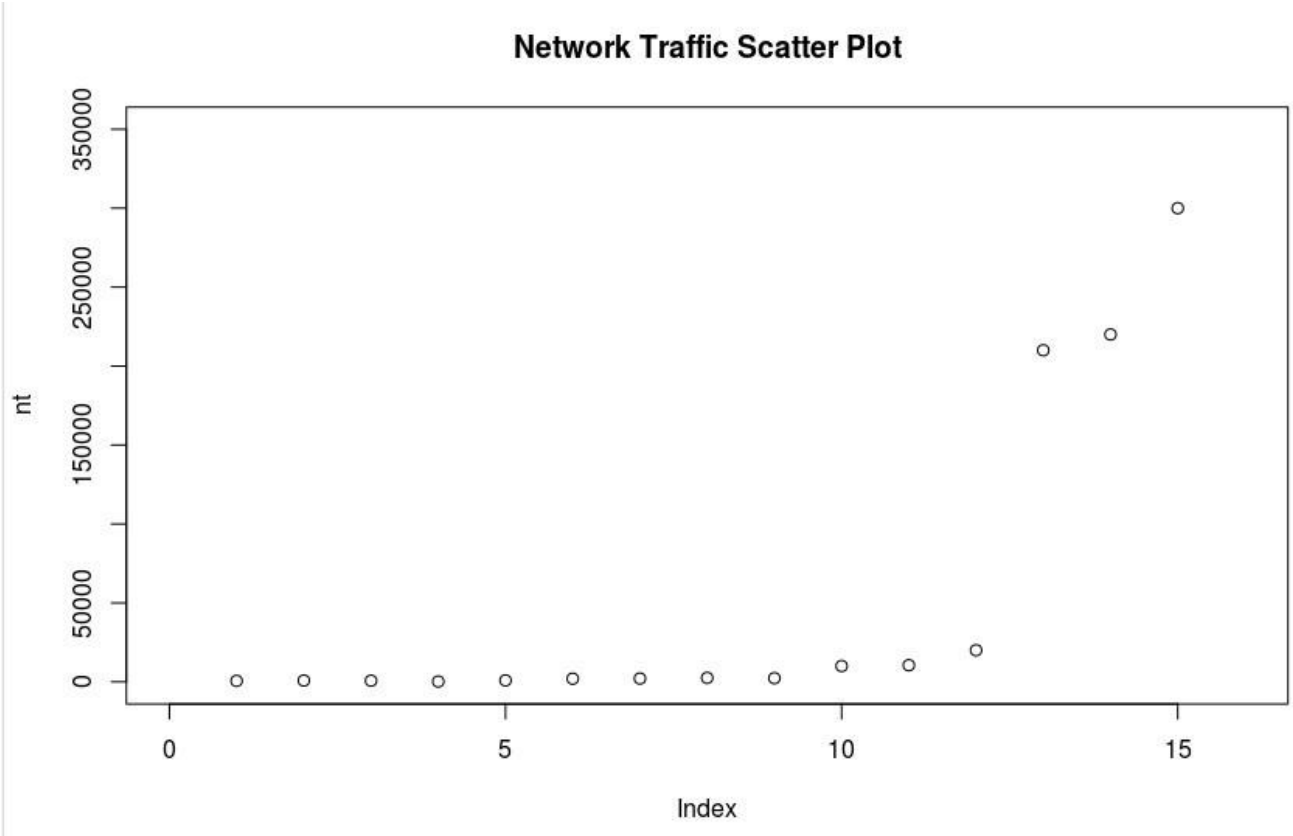

```
> boxplot(nt, xlab = "Network Traffic",ylab = "frequency", main =
"Network Traffic BOX Plot", varwidth=TRUE, col="green")
```




```
> plot(nt, type="o", col="orange", main="Network Traffic Line Plot")
```



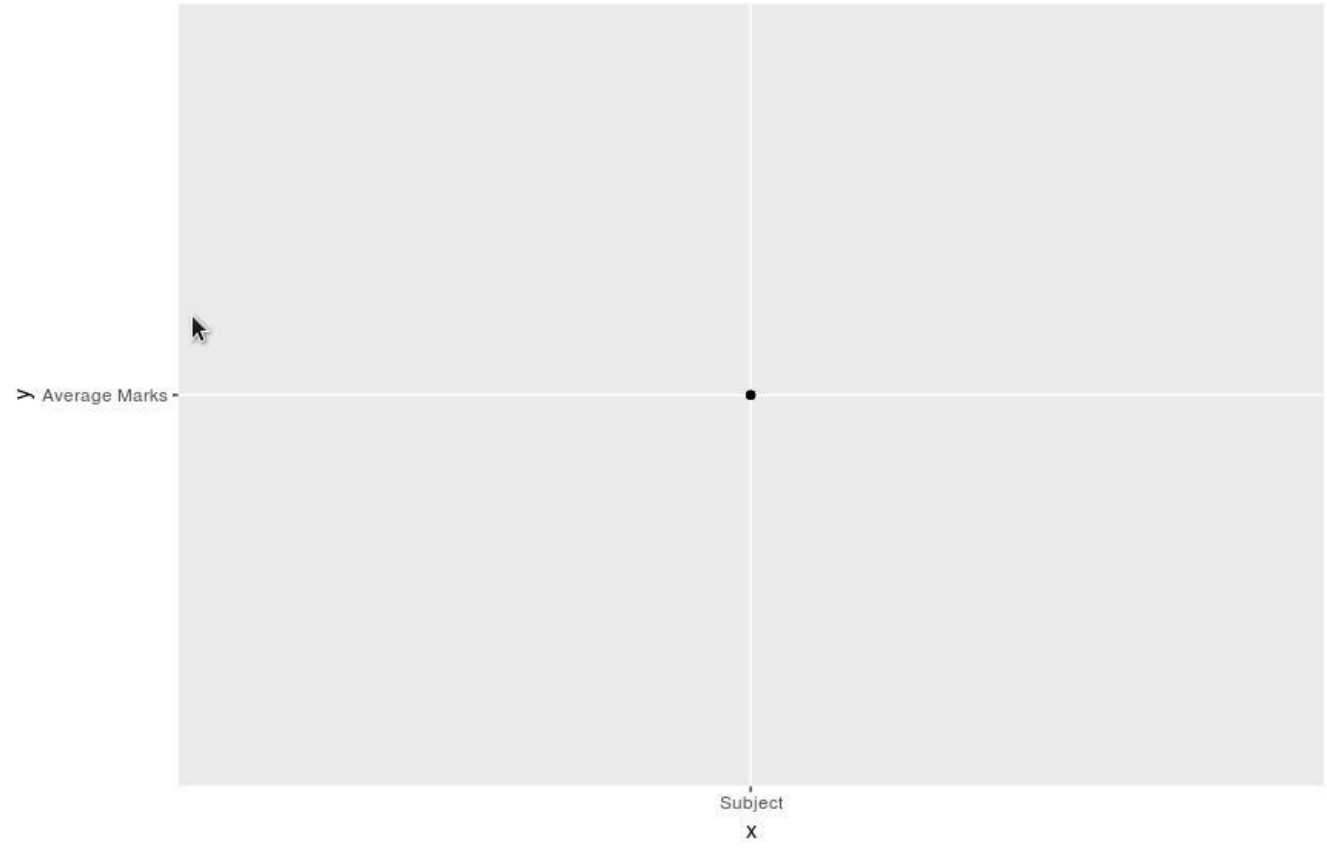

```
> plot(x = nt, xlim = c(0,16), ylim = c(0,350000), main = "Network
Traffic Scatter Plot")
```



Data – Average Marks

```
> lab1 = read.csv("lab1.csv")
> View(lab1)
> library(ggplot2)
> p=ggplot(lab1,aes(z="Subject",y="Average Marks",x="% people scored
average"))
> p+geom_point()
```

	Subject	Average Marks	% people scored average
1	Science	50	60
2	Maths	55	30
3	Social Studies	60	50
4	English	40	60
5	Second Language	55	40



Program 2:

Code:

X04cars_data <- read.csv("cars.csv")

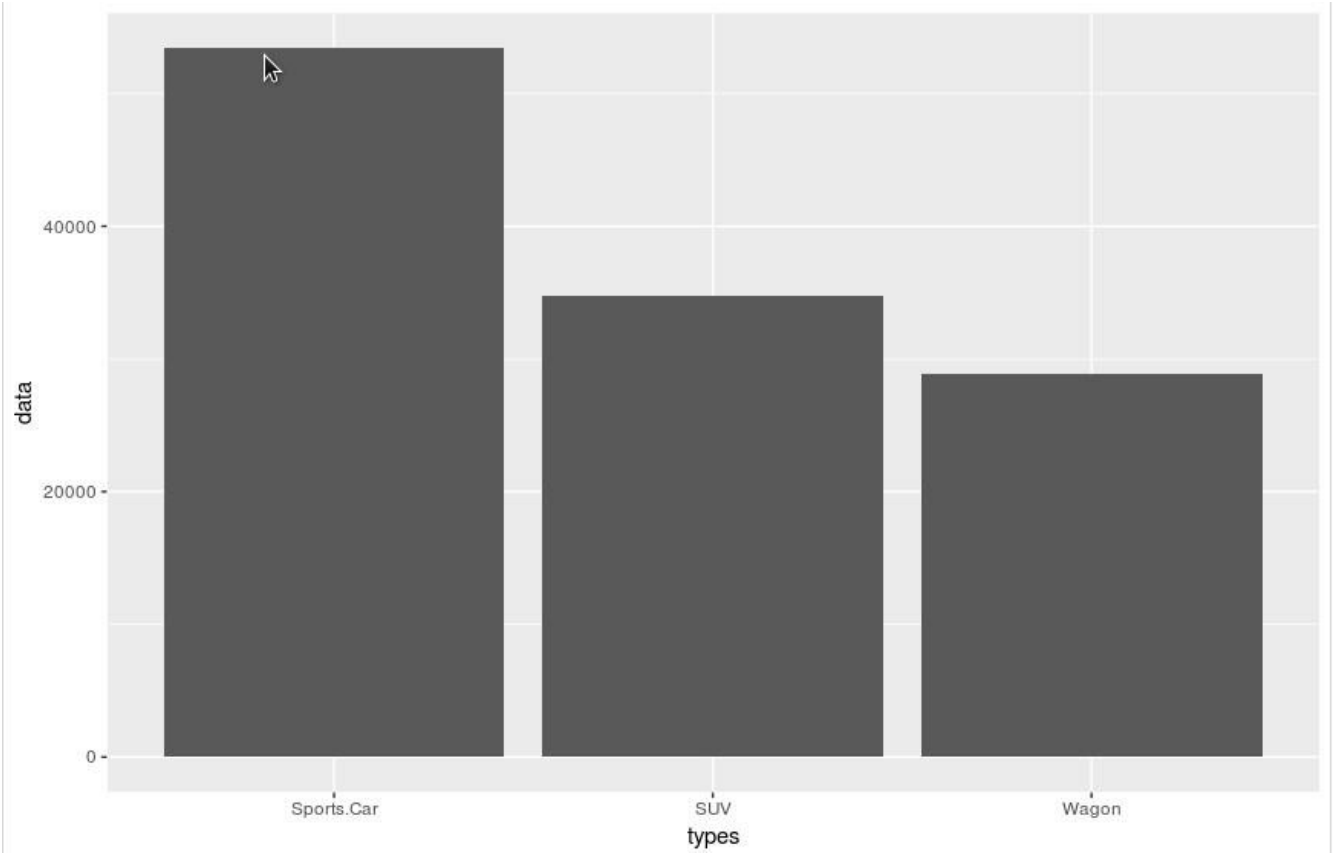
View(X04cars_data)

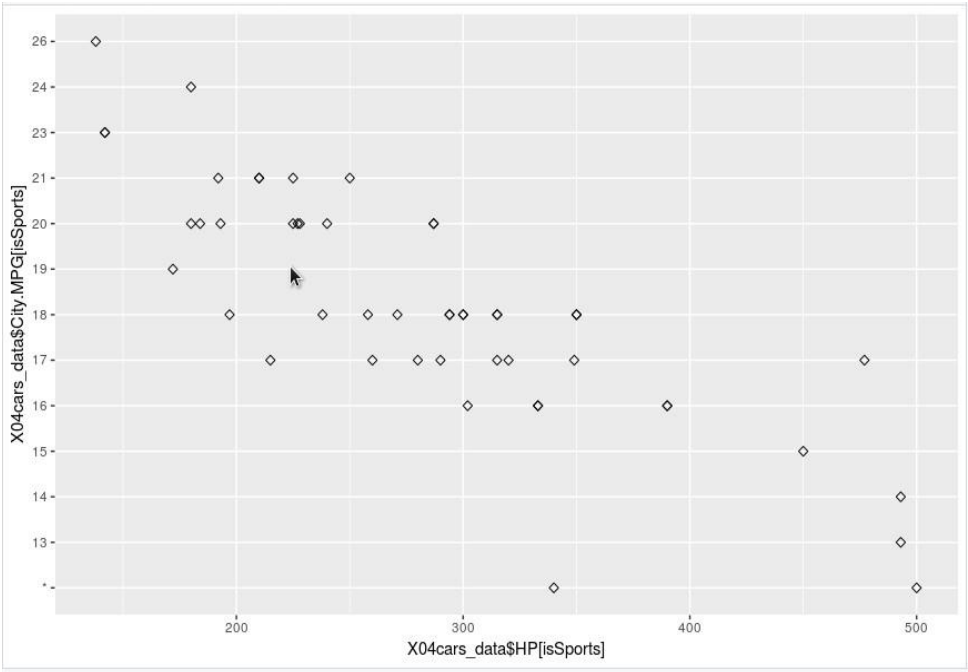
```
summary(X04cars_data)

##DATA

COMPOSITION##

slices = c(sum(X04cars_data$`Small/Sporty/ Compact/Large Sedan`),
sum(X04cars_data$`Sports Car`), sum(X04cars_data$SUV),
sum(X04cars_data$Wagon), sum(X04cars_data$Minivan),
sum(X04cars_data$Pickup), sum(X04cars_data$AWD), sum(X04cars_data$RWD))

lbls = names(X04cars_data[,2:9])

pie(slices, labels = lbls, main="Composition")

##DATA DISTRIBUTION##

library(ggplot2)

#isSports = (X04cars_data$`Sports Car` == 1)
#View(isSports)

isSports =
(X04cars_data$Sports.Car == 1)
isSUV = (X04cars_data$SUV == 1)
isWagon = (X04cars_data$Wagon
== 1)

a =
data.frame(group='Sports',value=c(X04cars_data[isSports,
10])) b =
```

```
data.frame(group='SUV',value=c(X04cars_data[isSUV, 10]))
c =
data.frame(group='Wagon',value=c(X04cars_data[isWagon
, 10])) #d =
data.frame(group='Sports',value=c(X04cars_data[isSports,1
0])) cars.data = rbind(a,b,c)
#ggplot(cars.data,aes(x=group,y=Retail.Price,fill=group)) +
geom_boxplot()
ggplot(a,aes(x=group,y=X04cars_data$Retail.Price,fill=grou
p)) + geom_boxplot()
ggplot(b,aes(x=group,y=X04cars_data$Retail.Price,fill=grou
p)) + geom_boxplot()
ggplot(c,aes(x=group,y=X04cars_data$Retail.Price,fill=grou
p)) + geom_boxplot()


#View(X04cars_data[isSports,3])


#DATA RELATION


ggplot(X04cars_data,aes(x=X04cars_data$Weight,y=X04cars_data$`Hwy MPG`)) +
geom_point(size=2,shape=23)
ggplot(X04cars_data,aes(x=X04cars_data$Weight,y=X04cars_data$`City MPG`)) +
geom_point(size=2,shape=23)
ggplot(X04cars_data[isSports,],aes(x=X04cars_data$HP[isSports],y=X04cars_data$`City
MPG`[isSports])) + geom_point(size=2,shape=23)


# DATA COMPARISON
```

```
car.types = names(X04cars_data)[3:5]
car.avgprice = c()
car.avgprice[1] = sum(X04cars_data[isSports,10])/sum(isSports)
car.avgprice[2] = sum(X04cars_data[isSUV,10])/sum(isSUV)
car.avgprice[3] = sum(X04cars_data[isWagon,10])/sum(isWagon)

car.compdata = data.frame(types=car.types,data=car.avgprice)
ggplot(data=car.compdata, aes(x=types, y=data)) + geom_bar(stat="identity")
```

Output:





Program 3:

```
> library(DT)
> data = read.csv("cars.csv", header = F)
> options(DT.options = list(pageLength = 429))
> catagorical = c("V2","V3","V4","V5","V6","V7","V8","V9")
> quantitative =
c("V10","V11","V12","V13","V14","V15","V16","V17","V18", "V19", "V20")
> ordinal = "V1"
> datatable(data) %>% formatStyle(catagorical,backgroundColor =
"lightblue") %>%
formatStyle(ordinal,backgroundColor = "yellow") %>%
formatStyle(quantitative,backgroundColor =
"red")
```

Show 429 entries Search:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20
1	Vehicle Name	Small/Sporty/ Compact/Large Sedan	Sports Car	SUV	Wagon	Minivan	Pickup	AWD	RWD	Retail Price	Dealer Cost	Engine Size (l)	Cyl	HP	City MPG	Hwy MPG	Weight	Wheel Base	Len	Width
2	Acura 3.5 RL 4dr	1	0	0	0	0	0	0	0	43755	39014	3.5	6	225	18	24	3880	115	197	72
3	Acura 3.5 RL w/Navigation 4dr	1	0	0	0	0	0	0	0	46100	41100	3.5	6	225	18	24	3893	115	197	72
4	Acura MDX	0	0	1	0	0	0	1	0	36945	33337	3.5	6	265	17	23	4451	106	189	77
5	Acura NSX coupe 2dr manual S	0	1	0	0	0	0	0	1	89765	79978	3.2	6	290	17	24	3153	100	174	71
6	Acura RSX Type S 2dr	1	0	0	0	0	0	0	0	23820	21761	2	4	200	24	31	2778	101	172	68
7	Acura TL 4dr	1	0	0	0	0	0	0	0	33195	30299	3.2	6	270	20	28	3575	108	186	72
8	Acura TSX 4dr	1	0	0	0	0	0	0	0	26990	24647	2.4	4	200	22	29	3230	105	183	69
9	Audi A4 1.8T 4dr	1	0	0	0	0	0	0	0	25940	23508	1.8	4	170	22	31	3252	104	179	70
10	Audi A4 3.0 4dr	1	0	0	0	0	0	0	0	31840	28846	3	6	220	20	28	3462	104	179	70
11	Audi A4 3.0 convertible 2dr	1	0	0	0	0	0	0	0	42490	38325	3	6	220	20	27	3814	105	180	70
12	Audi A4 3.0 Quattro 4dr auto	1	0	0	0	0	0	1	0	34480	31388	3	6	220	18	25	3627	104	179	70

Question 1

Problem 4

Design and implement an interactive visual idiom to help the users in choosing mobile phones with best price and features.

(What) Target - choosing mobile phone with best price and features

(Why) Tasks - Present

- Search

Set one goal for each task

i.e., present – communicate specific details precisely.

i.e., Search – browse for prices and features for a known mobile brand

Implement the goals using the following actions.

(How) Action – Encode, Navigate, Select, Arrange

CODE and OUTPUT

```
library(jsonlite) cp = fromJSON(txt = "Cell Phone Data.txt", simplifyDataFrame = TRUE) num.atts =

c(4,9,11,12,13,14,15,16,18,22)

cp[,num.atts] = sapply(cp[,num.atts], function (x) as.numeric(x)) cp$aspect.ratio =
cp$att_pixels_y / cp$att_pixels_x cp$isSmartPhone = ifelse(grepl("smart|iphone|blackberry",
cp$name, ignore.case=TRUE) == TRUE | cp$att_screen_size >= 4,
```

EXPERIMENT - 2

"Yes", "No") library(ggplot2) library(ggthemr)

library(scales) ggthemr("camoflauge")

ggplot(cp, aes(x=att_brand, y=price)) + geom_boxplot() + ggtitle("Mobile Phone Price by Brand") + theme(axis.text.x=element_text(angle=90, size=14, vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15), axis.title.y=element_text(size=15), plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_discrete("Brand")

ggplot(cp, aes(x=att_weight, y=price)) + geom_point(size=3) + ggtitle("Mobile Phone Price by Weight") + theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15), axis.title.y=element_text(size=15), plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_continuous("Weight (oz)") + stat_smooth(se=FALSE)

ggplot(cp, aes(x=att_screen_size, y=price)) + geom_point(size=3) + ggtitle("Mobile Phone Price by Screen Size") + theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15), axis.title.y=element_text(size=15), plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_continuous("Screen Size (in)") + stat_smooth(se=FALSE)

ggplot(cp, aes(x=att_ram, y=price)) + geom_point(size=3) + ggtitle("Mobile Phone Price by Amount of RAM") + theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15), axis.title.y=element_text(size=15), plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_continuous("RAM (gb)") + stat_smooth(se=FALSE) ggplot(cp, aes(x=att_sd_card, y=price)) + geom_point(size=3) + ggtitle("Mobile Phone Price by SD Card Capacity") + theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15), axis.title.y=element_text(size=15), plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_continuous("SD Card Capacity (gb)") + stat_smooth(se=FALSE)

ggplot(cp, aes(x=ifelse(cp$att_dual_sim == 1, "Yes", "No"), y=price)) + geom_boxplot() + ggtitle("Mobile Phone Price by Dual Sim") + theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15), axis.title.y=element_text(size=15), plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_discrete("Has Dual Sim Card?")

ggplot(cp, aes(x=att_storage, y=price)) + geom_point(size=3) + ggtitle("Mobile Phone Price by Storage Capacity") + theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15), axis.title.y=element_text(size=15), plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_continuous("Storage Capacity (gb)") + stat_smooth(se=FALSE)

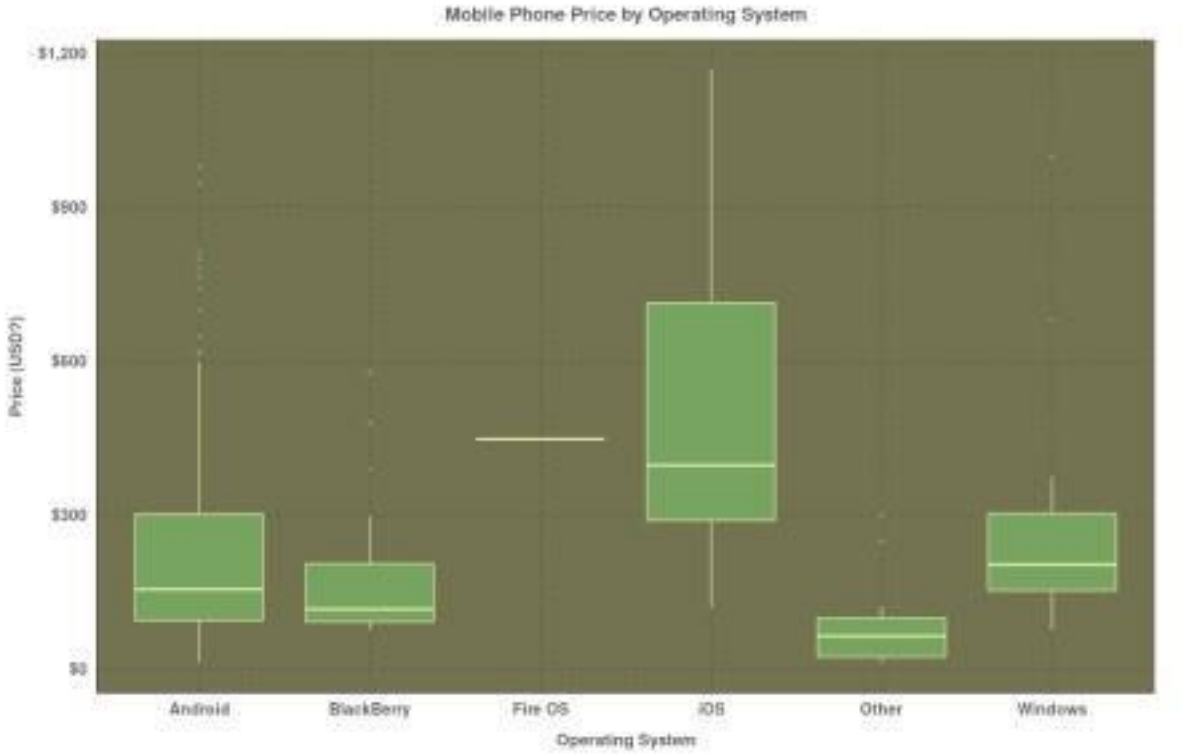
ggplot(cp, aes(x=att_battery_mah, y=price)) +

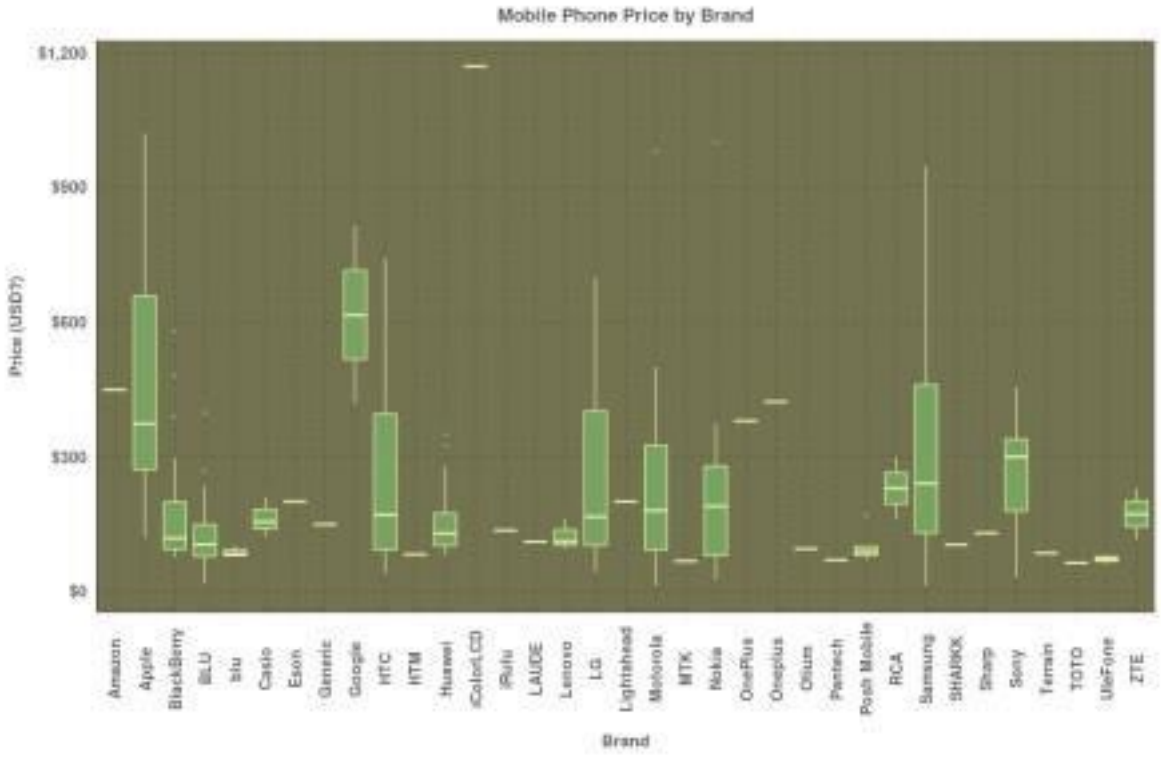
```
geom_point(size=3) + ggtitle("Mobile Phone Price by Battery Capacity") +
theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14),
axis.title.x=element_text(size=15), axis.title.y=element_text(size=15),
```

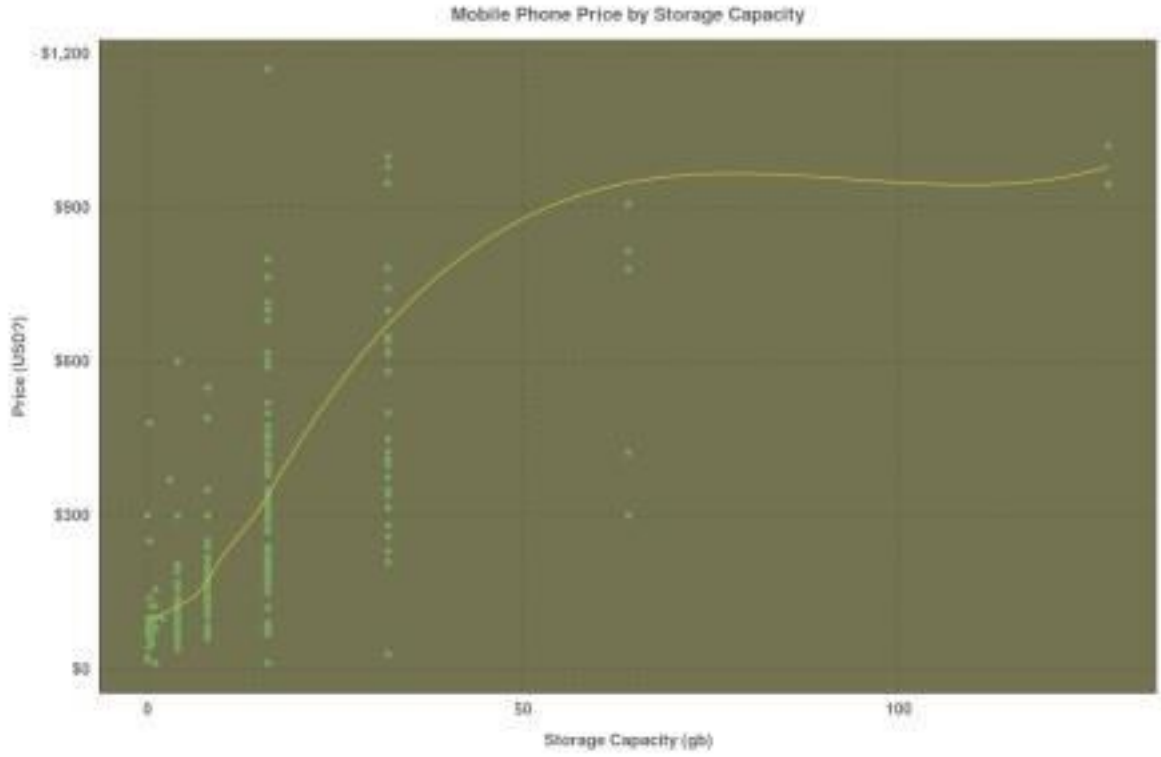
EXPERIMENT - 2

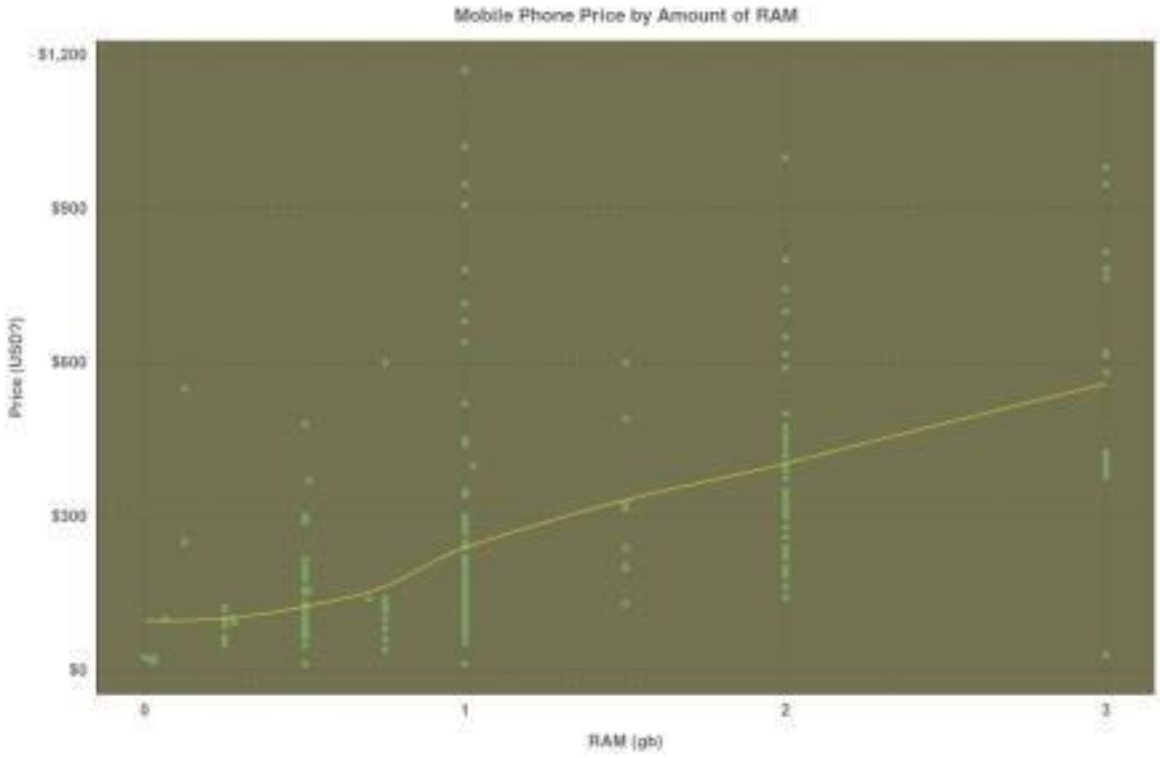
```
plot.title=element_text(size=17)) + scale_y_continuous(labels=dollar, name="Price (USD?)") +
scale_x_continuous("Battery Capacity (mAh)") + stat_smooth(se=FALSE)

ggplot(cp, aes(x=aspect.ratio, y=price)) + geom_point(size=3)
+ ggtitle("Mobile Phone Price by Aspect Ratio") + theme(axis.text.x=element_text(size=14,
vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15),
axis.title.y=element_text(size=15), plot.title=element_text(size=17)) +
scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_continuous("Aspect Ratio (Y
Pixels / X Pixels)") + stat_smooth(se=FALSE)

ggplot(cp, aes(x=isSmartPhone, y=price)) + geom_boxplot() +
ggtitle("Mobile Phone Price by Smart Phone Status") + theme(axis.text.x=element_text(size=14,
vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15),
axis.title.y=element_text(size=15), plot.title=element_text(size=17)) +
scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_discrete("Is it a Smart Phone?")

ggplot(cp, aes(x=att_os, y=price)) + geom_boxplot() +
ggtitle("Mobile Phone Price by Operating System") + theme(axis.text.x=element_text(size=14,
vjust=0.5), axis.text.y=element_text(size=14), axis.title.x=element_text(size=15),
axis.title.y=element_text(size=15), plot.title=element_text(size=17)) +
scale_y_continuous(labels=dollar, name="Price (USD?)") + scale_x_discrete("Operating System")
library(caret)
control = trainControl(method=                    "cv"    )

in_train = createDataPartition(cp$price, p=.8, list=FALSE)
model.gbm = train(price ~ att_brand + att_weight + att_screen_size +                att_ram +
att_sd_card + att_dual_sim +                att_storage +
att_battery_mah          +          att_os,  data=cp,          method="gbm",          trControl=control,
          verbose=FALSE, subset=in_train) cp$att_brand = factor(cp$)
```

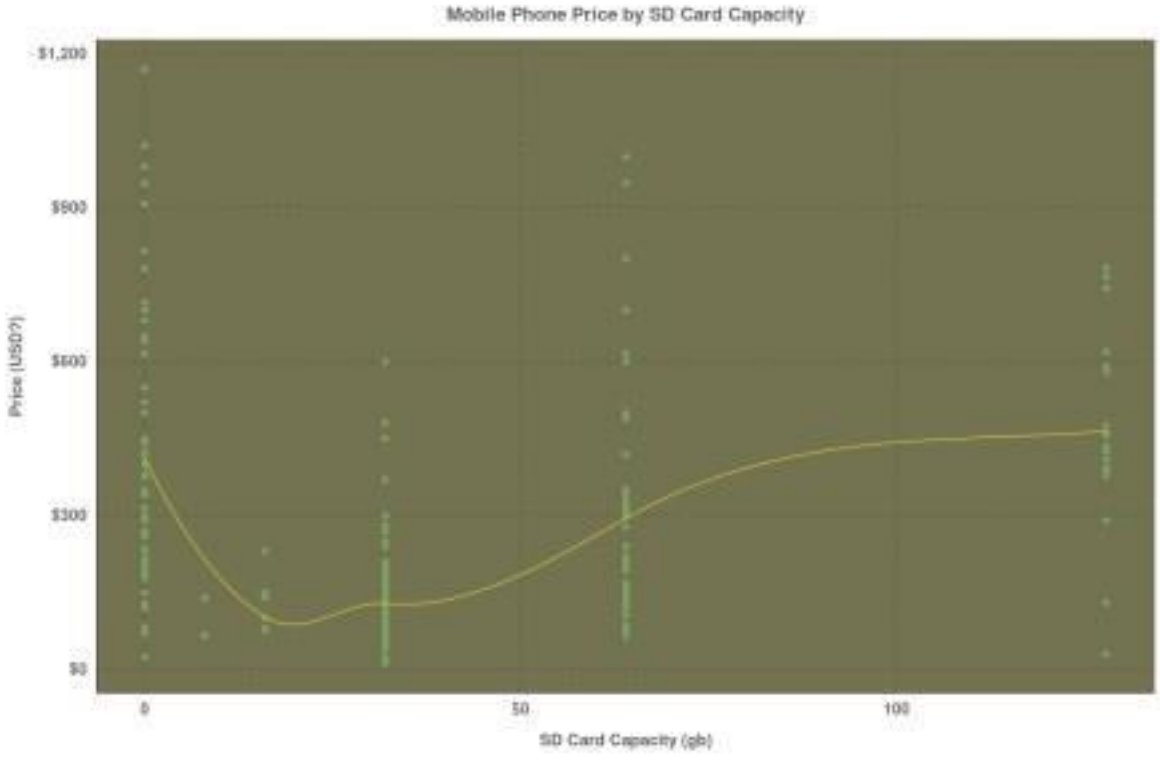
```
cp.test = cp[-in_train,] cp.test = subset(cp.test, att_brand != 'TOTO') cp.test
= na.omit(cp.test) cp.test$pred.price = predict(model.gbm, cp.test)

ggplot(cp.test, aes(x=pred.price, y=price)) +
geom_point(size=3) + ggtitle("Mobile Phone Price by Predicted Price") +
theme(axis.text.x=element_text(size=14, vjust=0.5), axis.text.y=element_text(size=14),
axis.title.x=element_text(size=15),
axis.title.y=element_text(size= 15) ,
plot.title=element_text(s ize= 17)) +
scale_y_continuous(labels=dollar, name= "Price (USD?)" ) +
scale_x_continuous( "Predicted Price" , labels=dollar) +
geom_abline(intercept= 0, slope= 1, colour= "yellow" ) +
stat_smooth(se=FALSE)
```



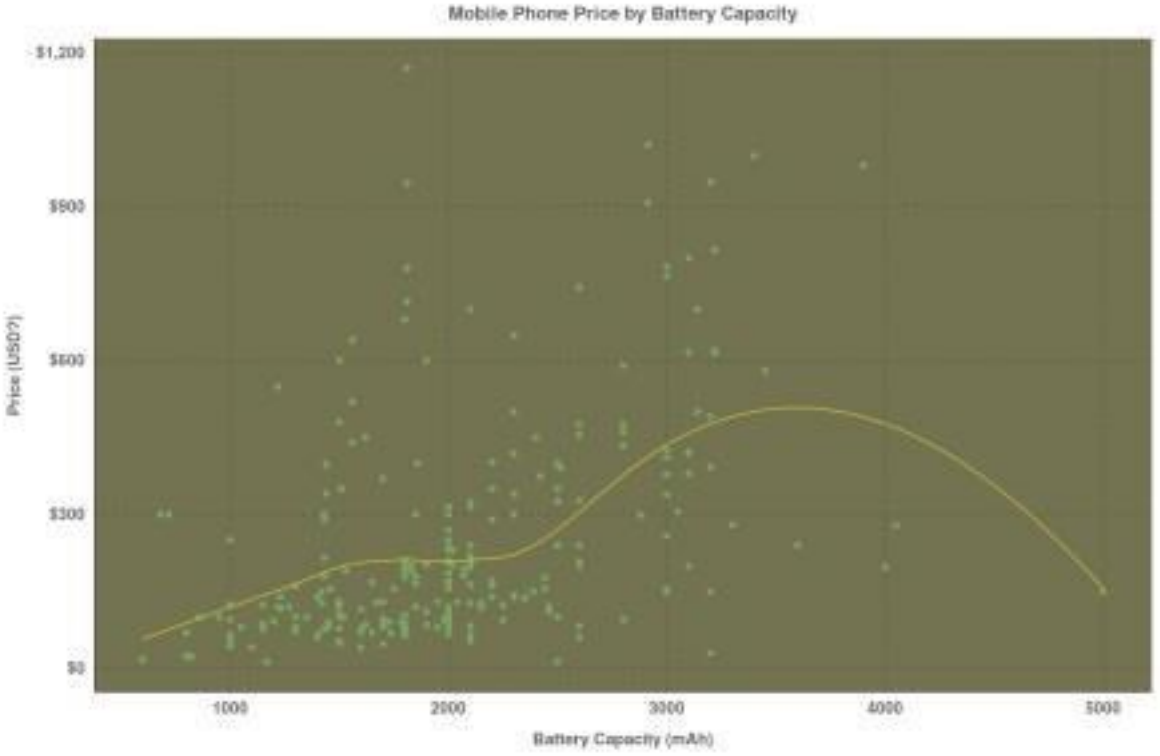

Mobile Phone Price by Brand
$1,200
$900
$600
$300
$0
Price (USD?)
Amazon
Apple
BlackBerry
BLU
Inlu
Casio
Eson
Generic
Google
HTC
HTM
Huawei
iColorCD
iRulu
LAUDE
Lenovo
LG
Lightshead
Motorola
MTK
Nokia
OnePlus
Oneplus
Oium
Pantech
Posh Mobile
RCA
Samsung
SHARKK
Sharp
Sony
Terrain
TOTO
UleFone
ZTE
Brand

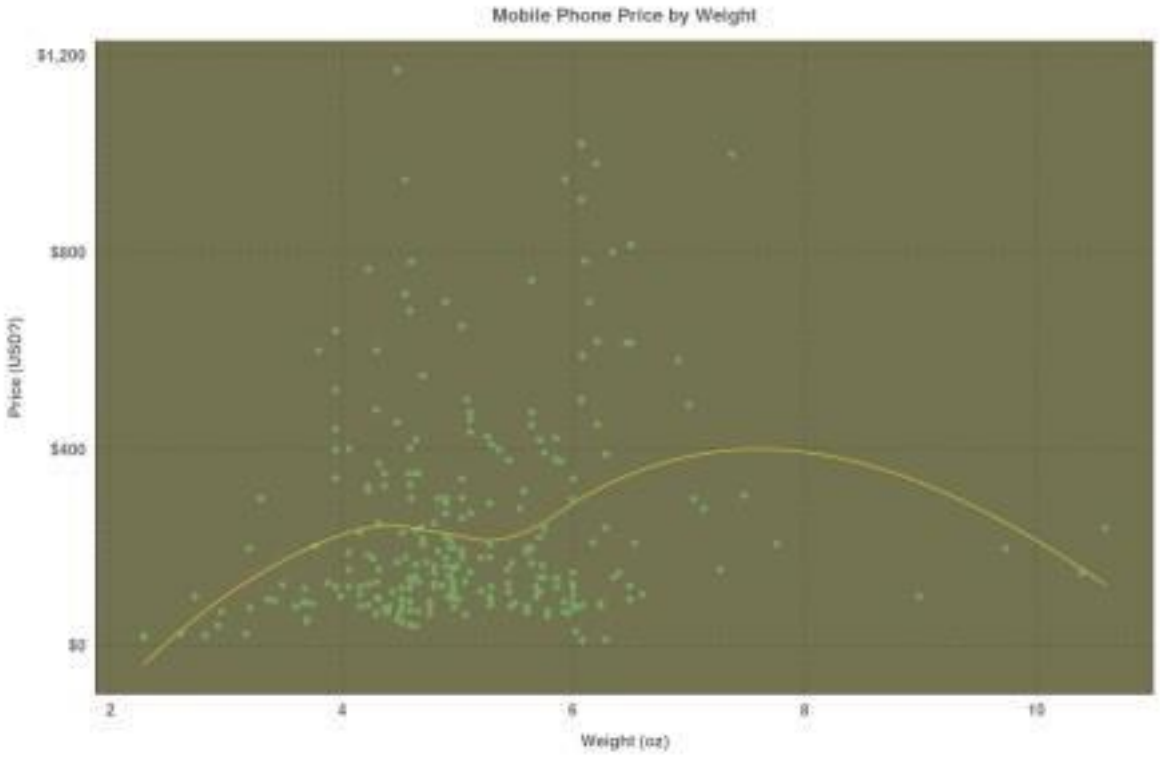

Mobile Phone Price by Storage Capacity
$1,200
$900
$600
$300
$0
Price (USD?)
0
50
100
Storage Capacity (gb)


Mobile Phone Price by Amount of RAM
$1,200
$900
$600
$300
$0
Price (USD)
0
1
2
3
RAM (gb)


Mobile Phone Price by SD Card Capacity
$1,200
$900
$600
$300
$0
Price (USD)
0
50
100
SD Card Capacity (gb)


Mobile Phone Price by Screen Size
$1,200
$900
$600
$300
$0
Price (USD)
2
3
4
5
6
7
Screen Size (in)


Mobile Phone Price by Battery Capacity
$1,200
$900
$600
$300
$0
Price (USD)
1000
2000
3000
4000
5000
Battery Capacity (mAh)



RESULT

Importance of each feature in the dataset

```
gbm variable importance

  only 20 most important variables shown (out of 41)

                  Overall
att_storage      100.0000
att_battery_mah   59.7597
att_weight        46.5410
att_ram           27.5871
att_osiOS         26.9977
att_screen_size   21.1106
att_sd_card       20.1130
att_brandSamsung   9.1220
```

Question 2

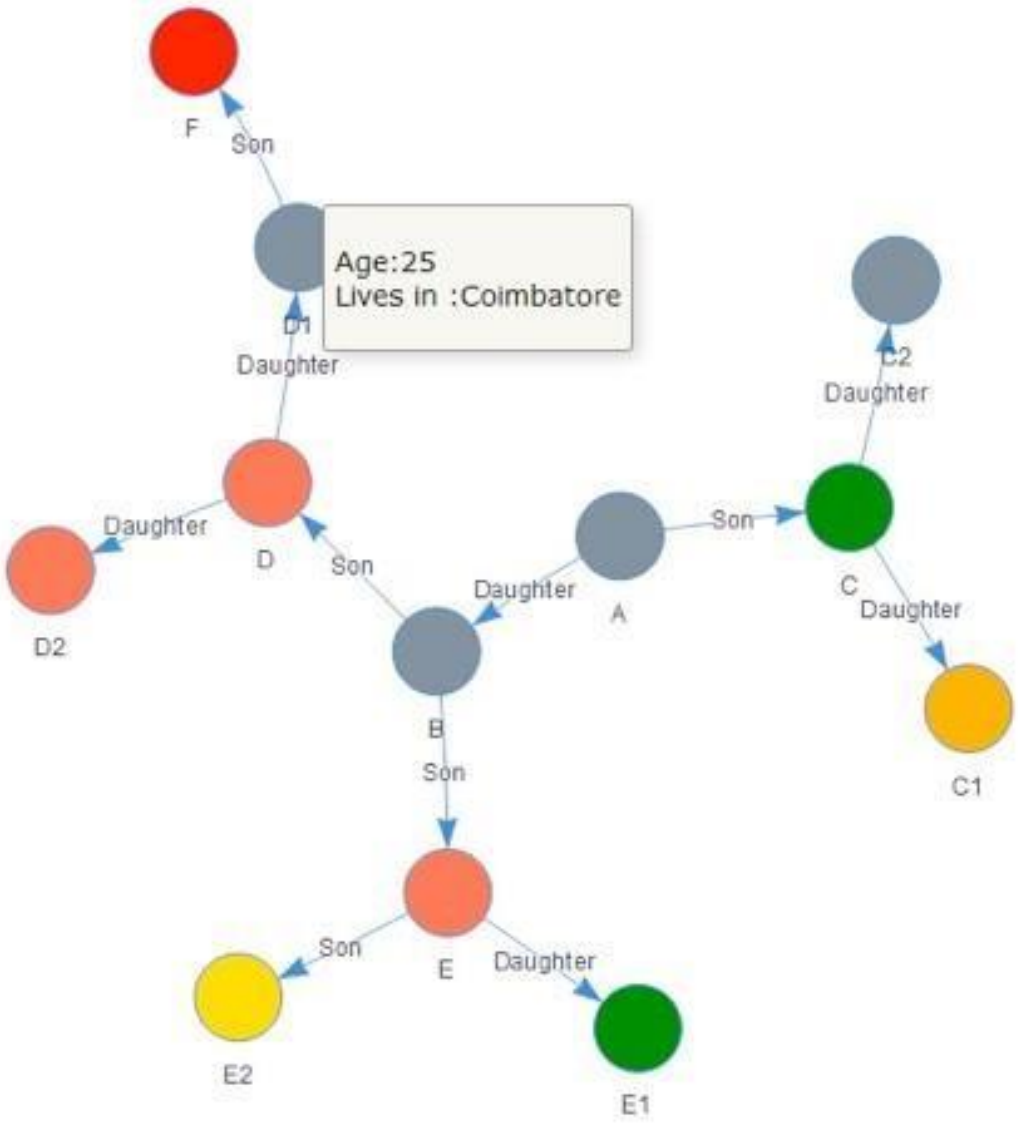
Problem 6

Generate a Node Edge Diagram (directed) for the data given in the table below. The attributes : Relationship, Age and Lives in should be displayed when there is a mouse click on each node. Use different shapes to differentiate gender. Also use similar node colors for the same places in which persons are living.

Source	Destination	Relationship	Age	Lives in
A	B	Daughter	75	Coimbatore
A	C	Son	78	Tiruppur
B	D	Son	55	Madurai
B	E	Son	52	Madurai
C	C1	Daughter	56	Shankarankovil
C	C2	Daughter	50	Coimbatore
D	D1	Daughter	25	Coimbatore
D	D2	Daughter	18	Madurai
E	E1	Daughter	20	Tiruppur
E	E2	Son	23	Nagarkovil
D1	F	Son	2	Thiruvarur

CODE and OUTPUT

```
data = read.csv("data.csv",header = TRUE)

Adata = data.frame(Node = 'A', Age = 100, Lives.in =
 Coimbatore )
 vertices = data.frame(Node = data$Destination, Age = data$Age, '

Lives.in = data$Lives.in) vertices =
rbind(Adata,vertices)

library(visNetwork) nodes = data.frame(id = vertices$Node, label= vertices$Node) nodes$title =
vertices$Age nodes$color.background = c("slategrey", "tomato",
"gold","orange","red","green")[as.factor(vertices$Lives.in)] nodes$title = paste0("<p>Age:",
vertices$Age,"<br>Lives in :", vertices$Lives.in ,"</p>")

links = data.frame(from = data$Source, to = data$Destination, label = data$Relationship)

visNetwork(nodes, links) %>% visEdges(arrows = 'to')
```


F
Son
Age:25
Lives in :Coimbatore
D1
Daughter
C2
Daughter
Daughter
D
Son
Son
A
C
Daughter
D2
Daughter
B
Son
C1
Son
E
Daughter
E2
E1

Q1)

For the data given in table fit a regression model for the attribute Loan Sanctioned. Use visualization techniques to identify suitable attributes (Age,Income,Expenditure) to construct the model.

After construction, simulate the model and obtain the predicted values. Show the residual plot for observed and simulated values for the attribute loan sanctioned.

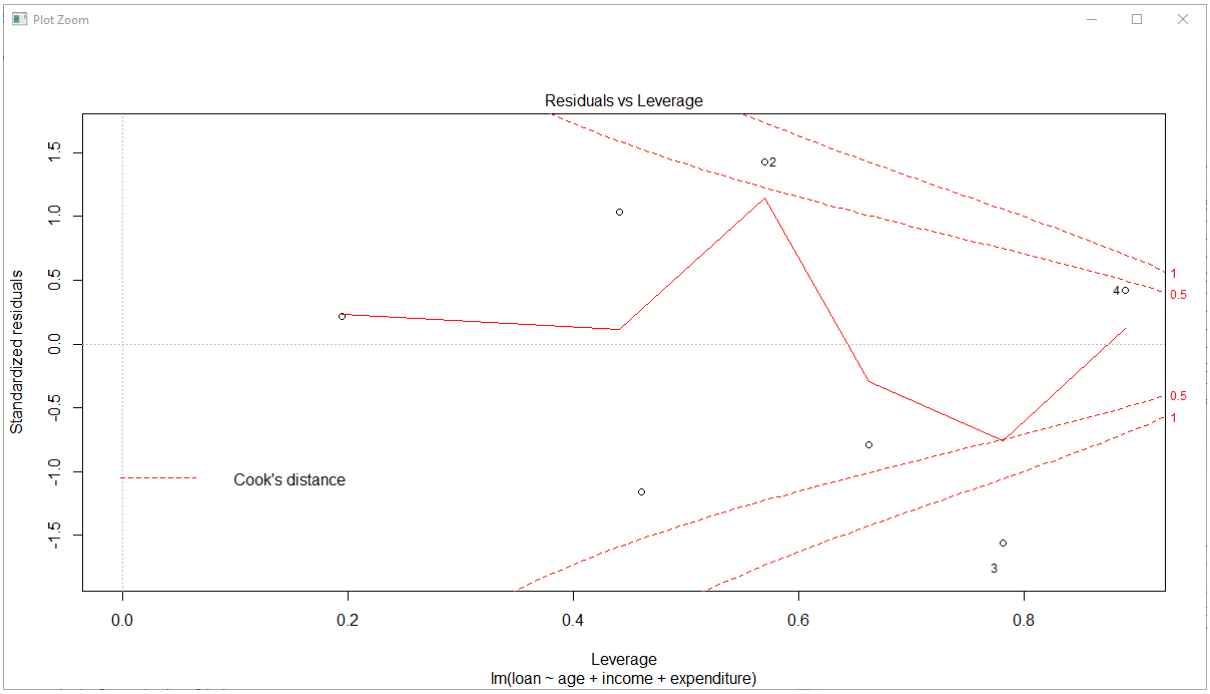
Loan Sanctioned	Age	Income	Expenditure
1000000	50	50000	40000
1200000	45	30000	25000
300000	20	10000	8000
3000000	35	60000	55000
4500000	30	50000	30000
3500000	40	60000	45000
5000000	45	80000	50000

CODE:

```
loan = c(1000000, 1200000, 300000, 3000000, 4500000, 3500000, 5000000)

age = c(50, 45, 20, 35, 30, 40, 45)

income = c(50000, 30000, 10000, 60000, 50000, 60000, 80000)

expenditure = c(40000, 25000, 8000, 55000, 30000, 45000, 50000)

#TODO: MAKE COMPARITIVE GRAPHS BETWEEN ABOVE

df = data.frame(loan, age,  income, expenditure)

df

model = lm(formula = loan ~ age + income + expenditure, data=df)
```

```
model
plot(model)


Y <- function(x, y, z) {
 t = 1555398.52 - 71082.69*x + 138.5*y - 81.38*z
 return(t)
}


predicted = c()
original = loan
for(i in 1:length(loan)) {
 predicted[i] = Y(age[i], income[i], expenditure[i])
}
df2 = data.frame(predicted, original)
df2
```

OUTPUT:



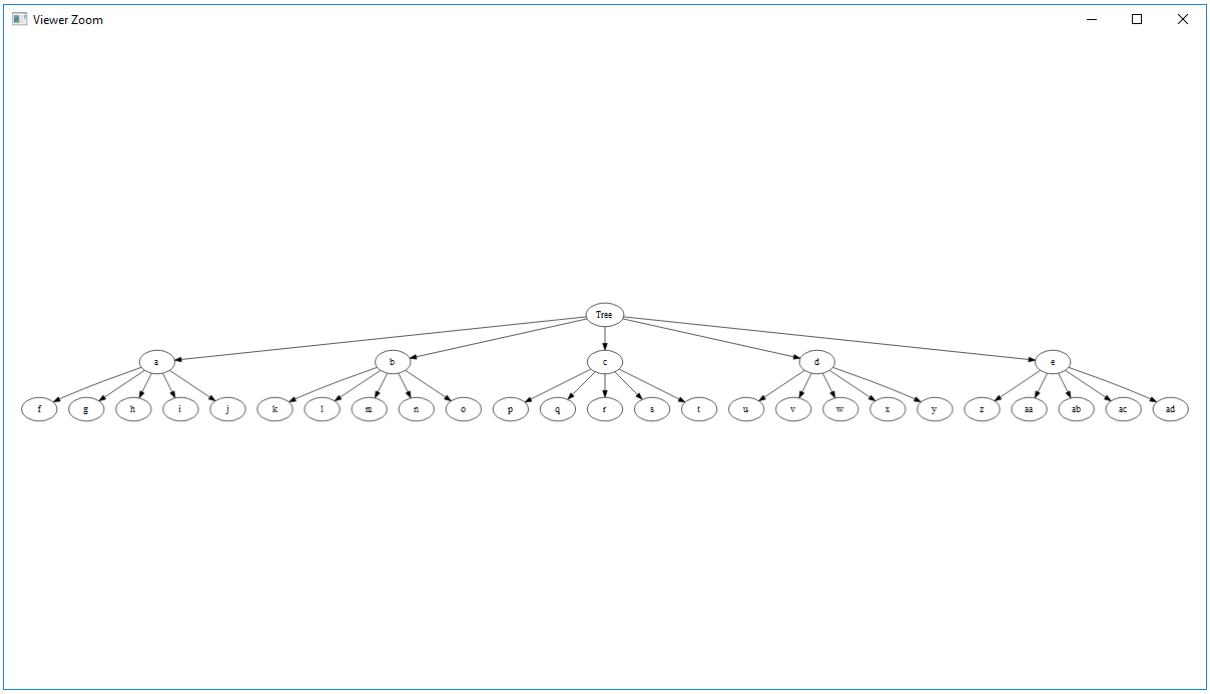
Q2) Generate tree data with 30 nodes with maximum of 5 children to each node and store it in a file. Visualize the stored data using radial spline, matrix and tree map.

CODE:

```
Main <- Node$new("Tree")

a <- Main$AddChild("a")

b <- Main$AddChild("b")

c <- Main$AddChild("c")

d <- Main$AddChild("d")

e <- Main$AddChild("e")


f <- a$AddChild("f")

g <- a$AddChild("g")

h <- a$AddChild("h")

i <- a$AddChild("i")

j <- a$AddChild("j")
```

```
k <- b$AddChild("k")
l <- b$AddChild("l")
m <- b$AddChild("m")
n <- b$AddChild("n")
o <- b$AddChild("o")

p <- c$AddChild("p")
q <- c$AddChild("q")
h <- c$AddChild("r")
s <- c$AddChild("s")
t <- c$AddChild("t")

u <- d$AddChild("u")
v <- d$AddChild("v")
w <- d$AddChild("w")
x <- d$AddChild("x")
y <- d$AddChild("y")

z <- e$AddChild("z")
aa <- e$AddChild("aa")
ab <- e$AddChild("ab")
ac <- e$AddChild("ac")
ad <- e$AddChild("ad")

plot(Main)
```

OUTPUT:



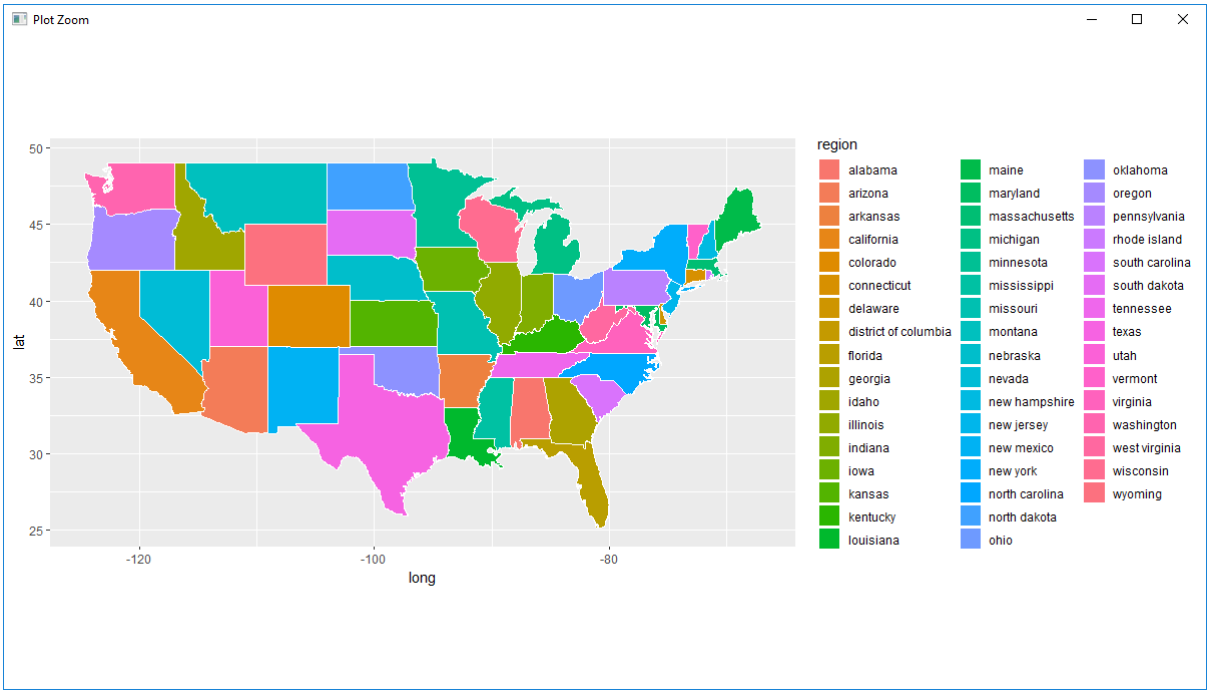
Q3)

Use any geo-visual package in R and show India with states. Use color coding to differentiate southern and northern states. Also provide interactive feature for selecting a specific state

CODE:

```
library(ggplot2)
library(ggmap)
library(maps)
library(mapdata)
usa <- map_data("usa")
ggplot() + geom_polygon(data = usa, aes(x=long, y = lat, group = group)) +
  coord_fixed(1.3)
states <- map_data("state")
ggplot(data = states) +
  geom_polygon(aes(x = long, y = lat, fill = region, group = group), color = "white") +
  coord_fixed(1.3) # do this to leave off the color legend
```

OUTPUT:



Q) Take a text document. Show the word cloud for the document.

```
> library("tm")
> library("SnowballC")
> library("wordcloud")
> library("RColorBrewer")
> text <- readLines(file.choose())
Warning message:
In readLines(file.choose()) :  incomplete final line found on 'C:\Users\Aditya Dhall\Desktop\text.txt'
> docs <- Corpus(VectorSource(text))
> inspect(docs)
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 3

[1] Lorem ipsum is a pseudo-Latin text used in web design, typography, layout, and printing in place of English to emphasise design elements over content. It's also called placeholder (or filler) text. It's a convenient tool for mock-ups. It helps to outline the visual elements of a document or presentation,
```

eg typography, font, or layout. Lorem ipsum is mostly a part of a Latin text by the classical author and philosopher Cicero. Its words and letters have been changed by addition or removal, so to deliberately render its content nonsensical; it's not genuine, correct, or comprehensible Latin anymore. While lorem ipsum's still resembles classical Latin, it actually has no meaning whatsoever. As Cicero's text doesn't contain the letters K, W, or Z, alien to latin, these, and others are often inserted randomly to mimic the typographic appearence of European languages, as are digraphs not to be found in the original.

[2]

[3] In a professional context it often happens that private or corporate clients corder a publication to be made and presented with the actual content still not being ready. Think of a news blog that's filled with content hourly on the day of going live. However, reviewers tend to be distracted by comprehensible content, say, a random text copied from a newspaper or the internet. The are likely to focus on the text, disregarding the layout and its elements. Besides, random text risks to be unintendedly humorous or offensive, an unacceptable risk in corporate environments. Lorem ipsum and its many variants have been employed since the early 1960ies, and quite likely since the sixteenth century.

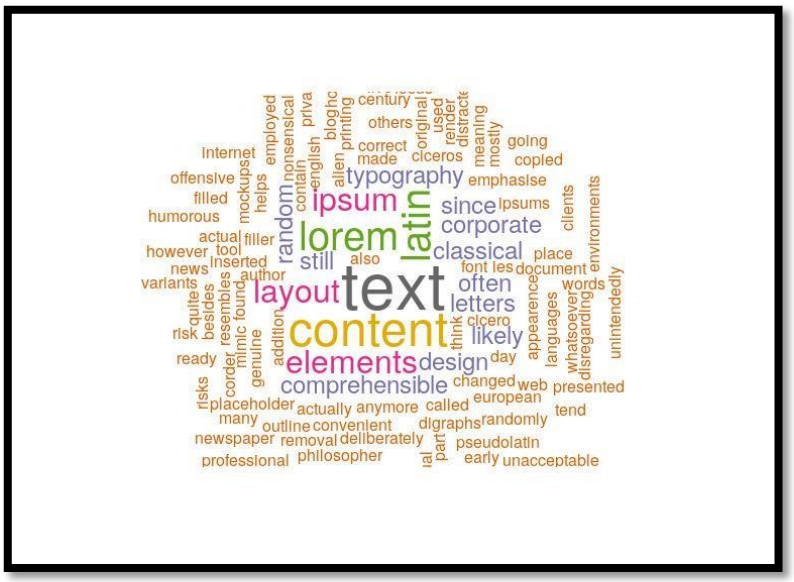
```
> toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
> docs <- tm_map(docs, toSpace, "/") Warning message:
In tm_map.SimpleCorpus(docs, toSpace, "/") : transformation drops documents

> docs <- tm_map(docs, toSpace, "@")
Warning message:

In tm_map.SimpleCorpus(docs, toSpace, "@") : transformation drops documents

> docs <- tm_map(docs, toSpace, "\\|")
Warning message:

In tm_map.SimpleCorpus(docs, toSpace, "\\|") :   transformation drops documents

> docs <- tm_map(docs, content_transformer(tolower))
Warning message:

In tm_map.SimpleCorpus(docs, content_transformer(tolower)) :   transformation drops documents

> docs <- tm_map(docs, removeNumbers)
Warning message:

In tm_map.SimpleCorpus(docs, removeNumbers) :   transformation drops
documents

> docs <- tm_map(docs, removeWords, stopwords("english"))
Warning message:

In tm_map.SimpleCorpus(docs, removeWords, stopwords("english")) :   transformation drops
documents

> docs <- tm_map(docs, removePunctuation)
Warning message:
```

In tm_map.SimpleCorpus(docs, removePunctuation) : transformation drops documents

```
> docs <- tm_map(docs, stripWhitespace)
Warning message:
```

In tm_map.SimpleCorpus(docs, stripWhitespace) : transformation drops documents

```
> dtm <- TermDocumentMatrix(docs)

> m <- as.matrix(dtm)
> v <- sort(rowSums(m),decreasing=TRUE)
> d <- data.frame(word = names(v),freq=v)
> head(d, 10)
                word freq text              text   7
content         content    5 latin             latin    4
lorem           lorem    4 elements          elements
3 ipsum           ipsum    3 layout            layout
3 classical       classical    2 comprehensible
comprehensible    2 design            design    2

>
>
> wordcloud(words = d$word, freq = d$freq, min.freq = 1,max.words=200, random.order=FALSE, rot.per=0.35,colors=brewer.pal(8, "Dark2"))
```



Q2) Take a time series data. Show the three components of the time series data.

```
> library(Quandl)
```

> Quandl.search("new home construction")
Uruguay - New Housing Construction Index

Code: SGE/URYHOUS

Desc: This index tracks the number of new houses built in Uruguay . Units: Index Points 1990=100, NSA Source: Instituto Nacional de Estadstica,

Uruguay

Freq: quarterly

Cols: Date | Value

Housing Market Indicators in New Brunswick: Construction

Code: CMHC/NB_HOUSINGCONSTRUCTION

Desc: These tables bring together information from a variety of sources. They provide an overview of housing conditions and trends in Canada and in each province and Census Metropolitan Area. Some tables assemble data for different markets, while others feature one particular market. Contents include information on housing construction, and sales; rental market trends; mortgage lending; the housing stock; household characteristics; and housing need. Tables also contain data on factors that influence housing markets, such as employment trends, demographic changes, and income growth.

Freq: annual

Cols: Year | Total Starts | Single Starts | Multiple Starts | Semi-

Detached | Row | Apartment | Starts Intended Mkt | Owned | Rental | Condo | Other Starts | Completions

New Privately-Owned Housing Units Under Construction: Total

Code: FRED/UNDCONTNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Under Construction: Total

Code: FRED/UNDCONTSA

Desc: Thousands of Units Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Under Construction in the West Census Region

Code: FRED/UNDCONWTSA

Desc: Thousands of Units Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Under Construction in the Midwest Census Region

Code: FRED/UNDCONMWTSA

Desc: Thousands of Units Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Under Construction in the Northeast Census Region

Code: FRED/UNDCONNETNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Under Construction: 1-Unit Structures

Code: FRED/UNDCON1UNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Under Construction in the Midwest Census Region

Code: FRED/UNDCONMWTNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Under Construction in the West Census

Region

Code: FRED/UNDCONWTNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

> Quandl.search("Housing Units Completed", source="FRED")
New Privately-Owned Housing Units Completed: Total

Code: FRED/COMPUTSA

Desc: Thousands of Units Seasonally Adjusted Annual Rate,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed: Total

Code: FRED/COMPUTNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed: 1-Unit Structures Code: FRED/COMPU1USA

Desc: Thousands of Units Seasonally Adjusted Annual Rate,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed: 1-Unit Structures

Code: FRED/COMPU1UNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed: 2-4 Unit Structures

Code: FRED/COMPU24UNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed: 2-4 Unit Structures

Code: FRED/COMPU24USA

Desc: Thousands of Units Seasonally Adjusted Annual Rate,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed: 5-Unit Structures or More

Code: FRED/COMPU5MUNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed: 5-Unit Structures or More

Code: FRED/COMPU5MUSA

Desc: Thousands of Units Seasonally Adjusted Annual Rate,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed in the Northeast Census Region

Code: FRED/COMPUNETNSA

Desc: Thousands of Units Not Seasonally Adjusted,

Freq: monthly

Cols: Date | Value

New Privately-Owned Housing Units Completed in the Midwest Census Region

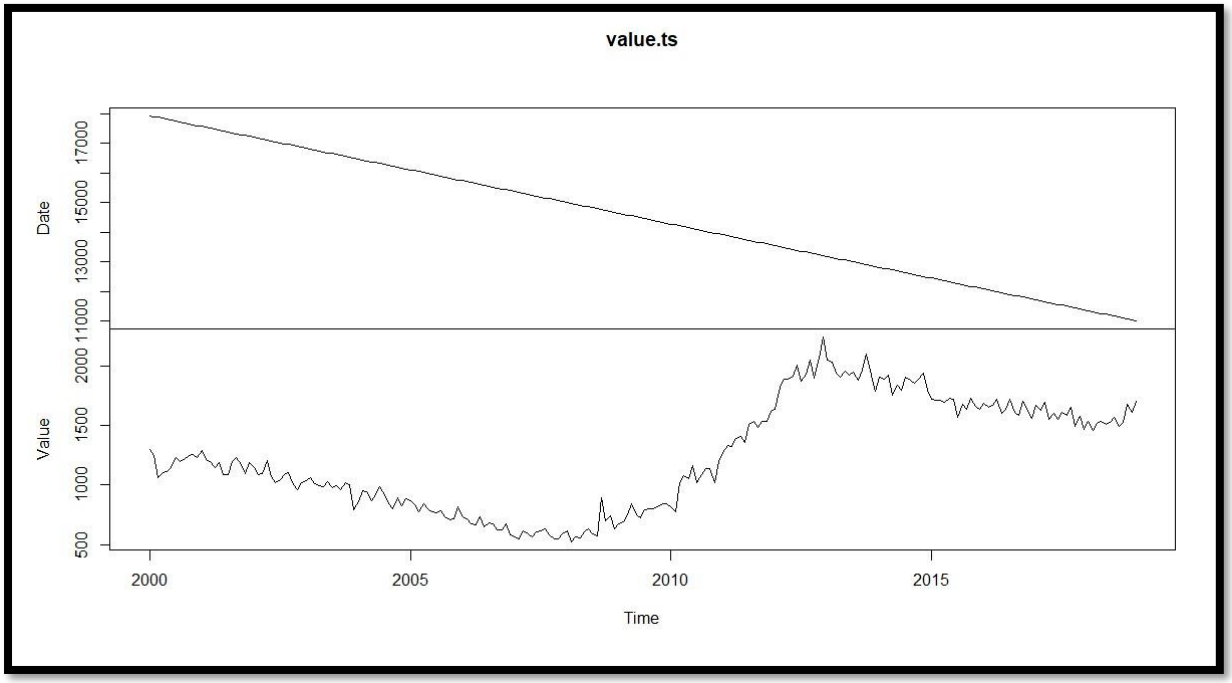
Code: FRED/COMPUMWTNSA

Desc: Thousands of Units Not Seasonally Adjusted,

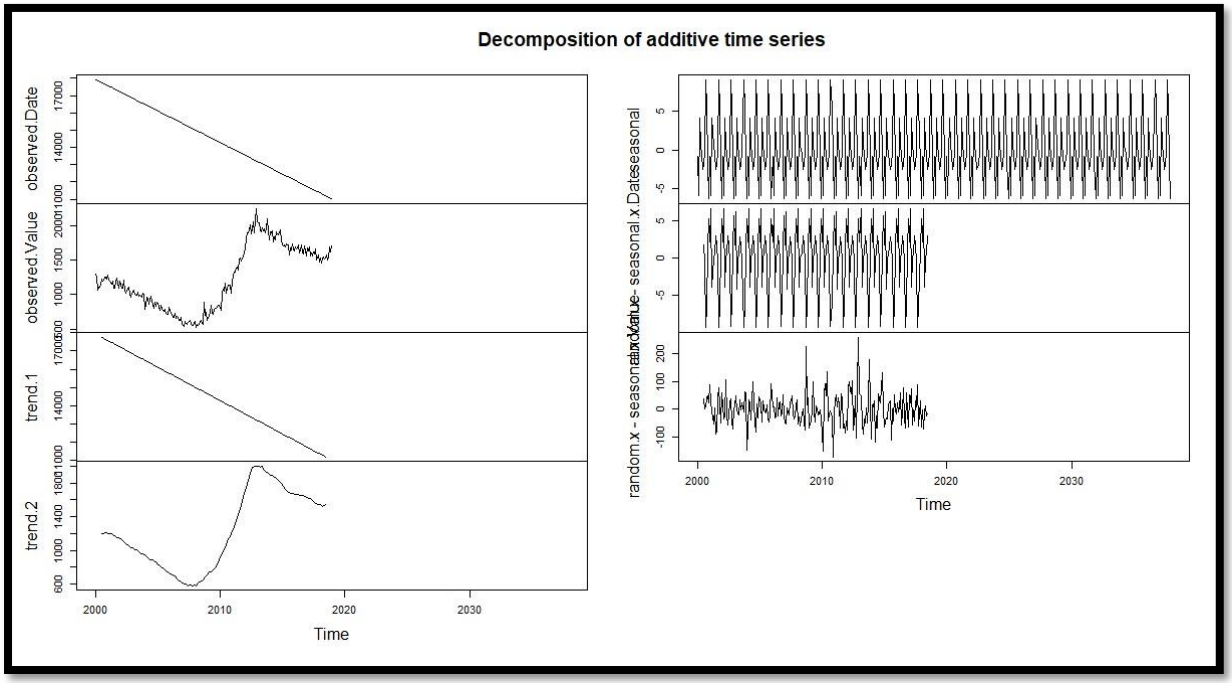
Freq: monthly

Cols: Date | Value

```
> Units = Quandl("FRED/COMPUTSA")
> head(Units)
      Date Value
1 2019-02-01  1303
2 2019-01-01  1247
3 2018-12-01  1063
4 2018-11-01  1101
5 2018-10-01  1111
6 2018-09-01  1148
> value.ts = ts(Value, data=Units, frequency=12, start=c(2000,1), end=c(2018,12)) >
plot(value.ts)
```




```
>  d = decompose(value.ts)
> plot(d)
```


Decomposition of additive time series
observed.Date
observed.Value
trend.1
trend.2
seasonal.x.Date
seasonal.x.Value
random.x - seasonal.x.Date
random.x - seasonal.x.Value
Time
Time