Assignment -3

Q-1a)

Code

```
#importing the dataset and skipping for first four lines as they are not data

rs_man<-read.table(file.choose(),header = T,skip =4,sep=",")

View(rs_man)

rs_mandf<-data.frame(rs_man)


#changes the date format to month/day/year


rs_mandf[,21]<-as.Date(rs_mandf[,21],"%m/%d/%y")


#remove the 7th and 1st column from the data set as they are not significant for us

rs_mandf_modify<-rs_mandf[,-7]

rs_mandf_modify<-rs_mandf_modify[,-1]


#if value of the LAND.SQUARE.FEET==0 or GROSS.SQUARE.FEET==0 then we are setting it to null so that we can use rm.na=T in #methods


rs_mandf_modify$LAND.SQUARE.FEET[rs_mandf_modify$LAND.SQUARE.FEET==0] <- NA

rs_mandf_modify$GROSS.SQUARE.FEET[rs_mandf_modify$GROSS.SQUARE.FEET==0] <- NA

install.packages("stringr", dependencies=TRUE)

library(stringr)


#if apartment value is "" then we replace it to NA


rs_mandf_modify$APARTMENT.NUMBER[str_trim(rs_mandf_modify$APARTMENT.NUMBER)==""] <- NA
```

```
#rs_mandf_modify$APARTMENT.NUMBER[str_trim(rs_mandf_modify$APARTMENT.NUMBER)==""
|| #str_trim(rs_mandf_modify$APARTMENT.NUMBER)=="-"] <- NA


#we are removing , from the SALE.PRICE so that we can convert it into integer


rs_mandf_modify[,18]<-as.numeric(gsub(",","",rs_mandf_modify[,18]))


#plot(rs_mandf_modify[,19])--useless code


#we are removing , from the GROSS.SQUARE.FEET so that we can convert it into integer


rs_mandf_modify[,14]<-as.numeric(gsub(",","",rs_mandf_modify[,14]))


#plot(rs_mandf_modify[,14])---useless code


#we are removing , from the LAND.SQAURE.FEET so that we can convert it into integer

rs_mandf_modify[,13]<-as.numeric(gsub(",","",rs_mandf_modify[,13]))

#plot(rs_mandf_modify[,15])---useless code
```

Conclusion

We are clearing the data by doing following things

1)Changed the date format to month/day/year

2)Removing the 7^{th} and 1^{st} column from frame as they contains only null values

3) If value of the LAND.SQUARE.FEET==0 or GROSS.SQUARE.FEET==0 then we are setting it to null so that we can use rm.na=T in methods

4) We are removing , from the SALE.PRICE so that we can convert it into integer

5) We are removing , from the GROSS.SQUARE.FEET so that we can convert it into integer

6) We are removing , from the LAND.SQAURE.FEET so that we can convert it into integer

Q-1b)

Code

```
#importing the dataset and skipping for first four lines as they are not data

rs_man<-read.table(file.choose(),header = T,skip =4,sep=",")

View(rs_man)

rs_mandf<-data.frame(rs_man)


#changes the date format to month/day/year


rs_mandf[,21]<-as.Date(rs_mandf[,21],"%m/%d/%y")


#remove the 7th and 1st column from the data set as they are not significant for us

rs_mandf_modify<-rs_mandf[,-7]

rs_mandf_modify<-rs_mandf_modify[,-1]


#if value of the LAND.SQUARE.FEET==0 or GROSS.SQUARE.FEET==0 then we are setting it to null so that we can use rm.na=T in #methods


rs_mandf_modify$LAND.SQUARE.FEET[rs_mandf_modify$LAND.SQUARE.FEET==0] <- NA

rs_mandf_modify$GROSS.SQUARE.FEET[rs_mandf_modify$GROSS.SQUARE.FEET==0] <- NA

install.packages("stringr", dependencies=TRUE)

library(stringr)


#if apartment value is "" then we replace it to NA


rs_mandf_modify$APARTMENT.NUMBER[str_trim(rs_mandf_modify$APARTMENT.NUMBER)==""] <- NA
```

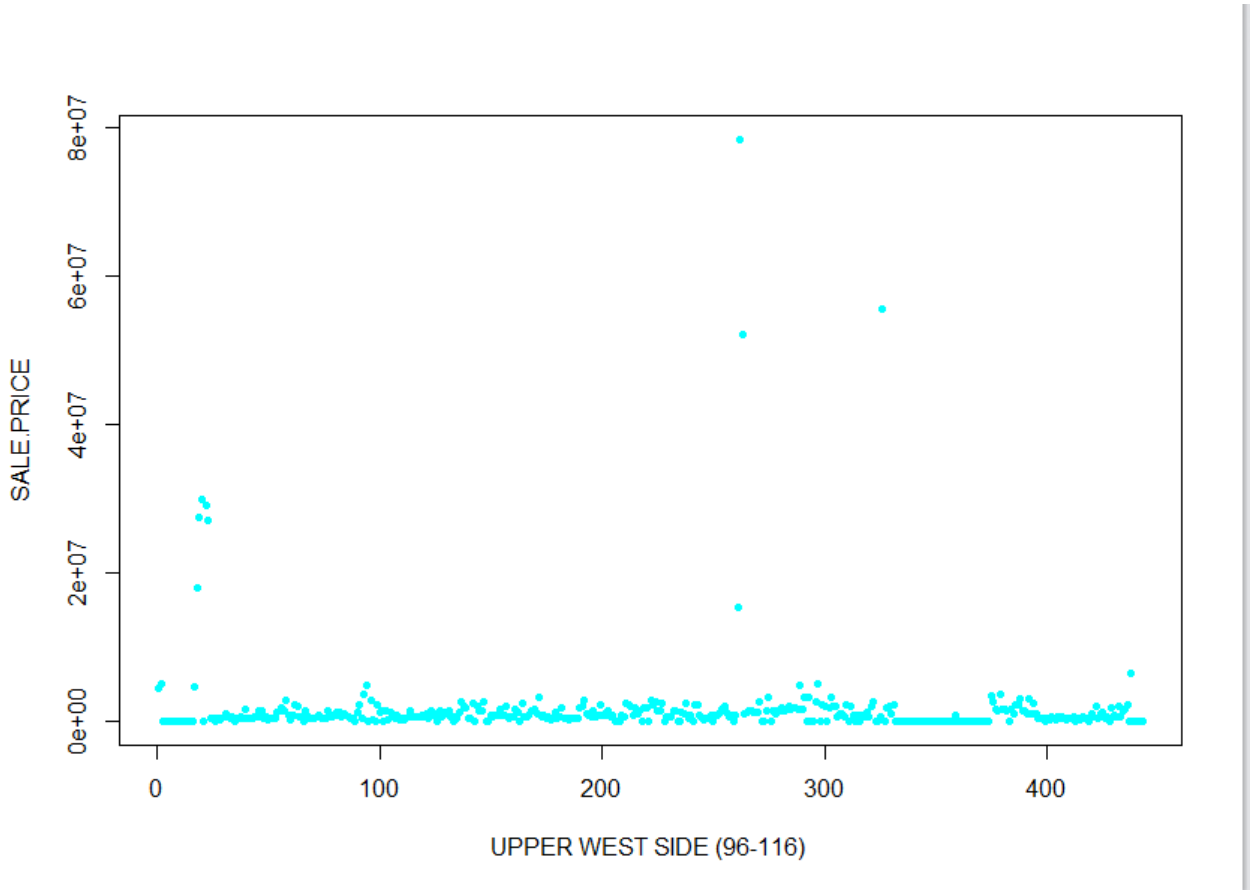
```
#rs_mandf_modify$APARTMENT.NUMBER[str_trim(rs_mandf_modify$APARTMENT.NUMBER)==""
|| #str_trim(rs_mandf_modify$APARTMENT.NUMBER)=="-"] <- NA


#we are removing , from the SALE.PRICE so that we can convert it into integer


rs_mandf_modify[,18]<-as.numeric(gsub(",","",rs_mandf_modify[,18]))


#plot(rs_mandf_modify[,19])--useless code


#we are removing , from the GROSS.SQUARE.FEET so that we can convert it into integer


rs_mandf_modify[,14]<-as.numeric(gsub(",","",rs_mandf_modify[,14]))


#plot(rs_mandf_modify[,14])---useless code


#we are removing , from the LAND.SQAURE.FEET so that we can convert it into integer

rs_mandf_modify[,13]<-as.numeric(gsub(",","",rs_mandf_modify[,13]))

#plot(rs_mandf_modify[,15])---useless code


#we are taking uniq counts of the neighborhood


plot(rs_mandf_modify$NEIGHBORHOOD,rs_mandf_modify$SALE.PRICE)

uniq<-unique(rs_mandf_modify$NEIGHBORHOOD)

length(uniq)


rs_mandf_modify.sale<-rs_mandf_modify[which(rs_mandf_modify$GROSS.SQUARE.FEET>0 &
rs_mandf_modify$LAND.SQUARE.FEET>0 & rs_mandf_modify$SALE.PRICE>0),]
```

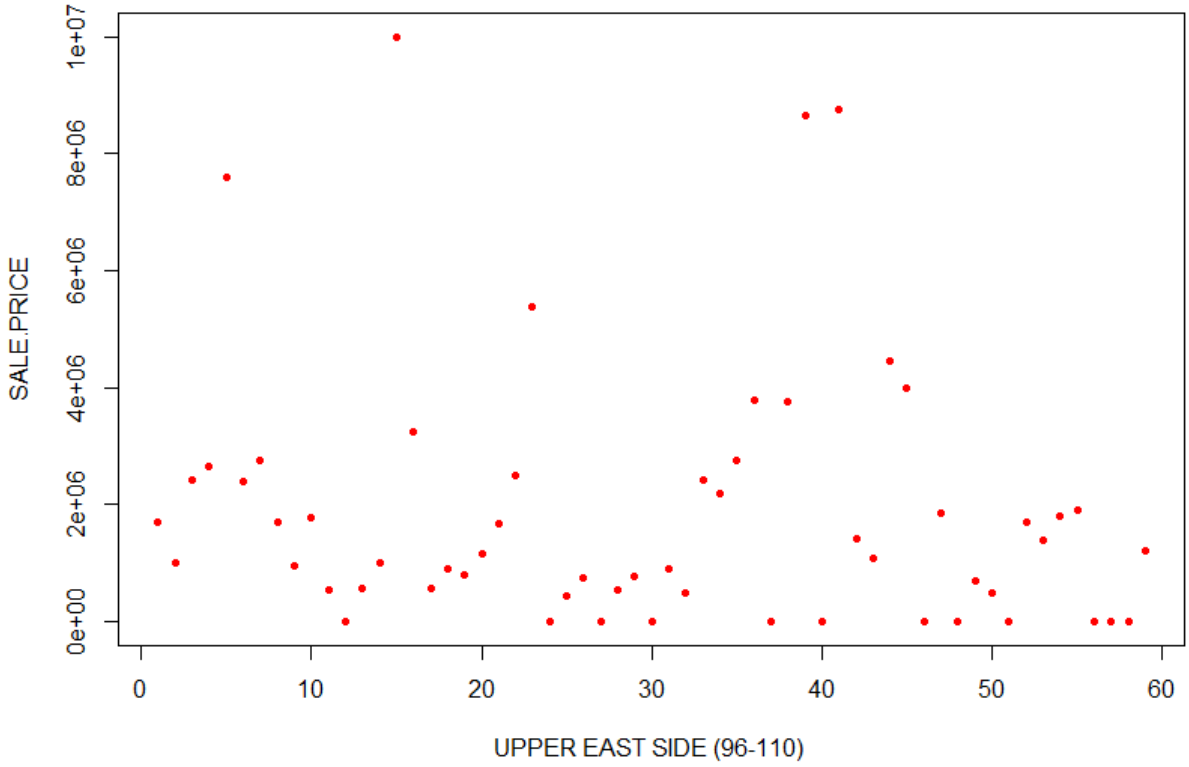
```
neighbourhoods_sd <- data.frame(neighbourhoods =
unique(rs_mandf_modify.sale$NEIGHBORHOOD))


for (i in 1:38)

{

  neighbourhoods_sd$sd_sale_price[i] <-
sd(rs_mandf_modify.sale$SALE.PRICE[rs_mandf_modify.sale$NEIGHBORHOOD ==
neighbourhoods_sd$neighbourhoods[i]],na.rm = TRUE)

  neighbourhoods_sd$sd_Commercial_units[i] <-
sd(rs_mandf_modify.sale$COMMERCIAL.UNITS[rs_mandf_modify.sale$NEIGHBORHOOD ==
neighbourhoods_sd$neighbourhoods[i]],na.rm = TRUE)

  neighbourhoods_sd$sd_Residential_units[i] <-
sd(rs_mandf_modify.sale$RESIDENTIAL.UNITS[rs_mandf_modify.sale$NEIGHBORHOOD ==
neighbourhoods_sd$neighbourhoods[i]],na.rm = TRUE)

  neighbourhoods_sd$sd_Land_Square_feet[i] <-
sd(rs_mandf_modify.sale$LAND.SQUARE.FEET[rs_mandf_modify.sale$NEIGHBORHOOD ==
neighbourhoods_sd$neighbourhoods[i]],na.rm = TRUE)

  neighbourhoods_sd$sd_built_square_feet[i] <-
sd(rs_mandf_modify.sale$GROSS.SQUARE.FEET[rs_mandf_modify.sale$NEIGHBORHOOD ==
neighbourhoods_sd$neighbourhoods[i]],na.rm = TRUE)


}

View(neighbourhoods_sd)


#rs_mandf_modify$SALE.PRICE[rs_mandf_modify$NEIGHBORHOOD==uniq[1]]


#we are plotting SALE.PRICE for all the neighbourhoods and their counts

for(i in 1:length(neighbourhoods_sd)){

plot(rs_mandf_modify$SALE.PRICE[rs_mandf_modify$NEIGHBORHOOD==uniq[i]],col=i,xlab=uniq[i],
ylab="SALE.PRICE",pch=20)

}


#we are polting graph of SALE.PRICE vS SALE.DATE
```

```
plot(rs_mandf_modify$SALE.DATE,rs_mandf_modify$SALE.PRICE,ylab="SALE.PRICE",xlab="SALE.DATE",pch=20)
```

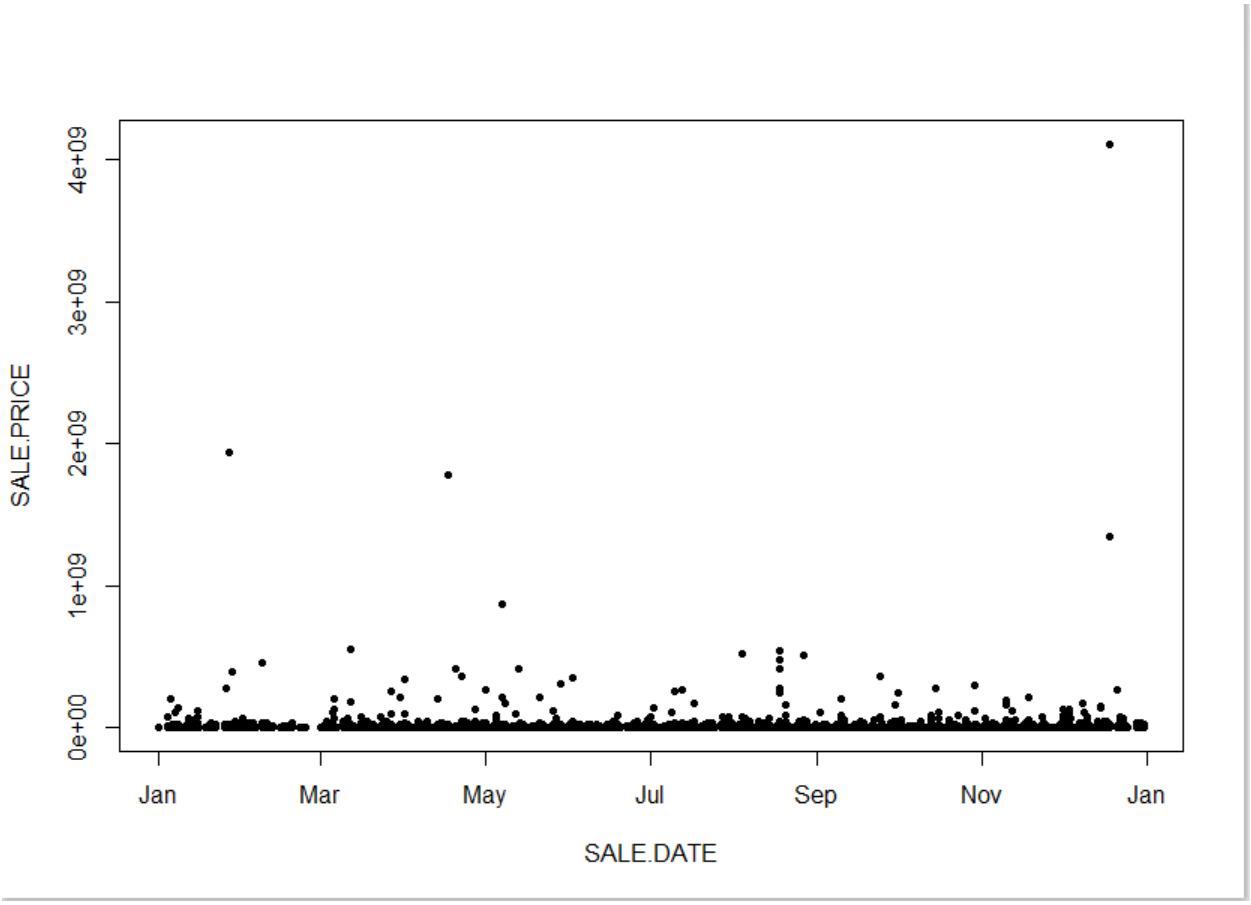
Conclusion

Below two graphs are for the NEIGHBORHOOD vs SALE.PRICE values. These graphs shows the combine SALE.PRICE for a NEIGHBORHOOD vs their counts.



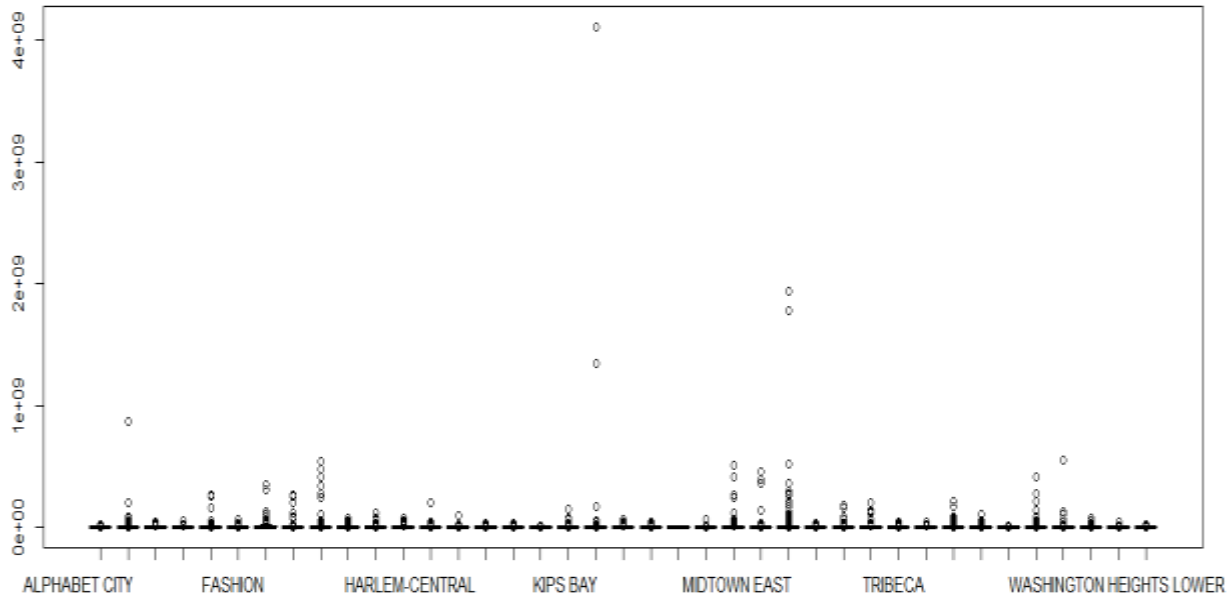


Below graph shows the relation between the SALE.PRICE and SALE.DATE



We can conclude from the above plot that there are higher SALE.PRICE from march to april as there are many and significant higher value points on y-axis than other months. We can also conclude that We have 3-4 outliers by seeing the value of SALE.PRICE values that are much higher.

	neighbourhoods	sd_sale_price
1	ALPHABET CITY	6027808.0
2	CHELSEA	31579653.2
3	CHINATOWN	7874913.8
4	CIVIC CENTER	18702161.2
5	CLINTON	49067714.1
6	EAST VILLAGE	9966050.1
7	FASHION	75984494.0
8	FINANCIAL	103570172.1
9	FLATIRON	69088545.8
10	GRAMERCY	17295636.9
11	GREENWICH VILLAGE-CENTRAL	24694601.4
12	GREENWICH VILLAGE-WEST	13551782.1
13	HARLEM-CENTRAL	16295847.8
14	HARLEM-EAST	13160644.9
15	HARLEM-UPPER	6255730.0
16	HARLEM-WEST	8559585.4
17	INWOOD	4690036.6
18	JAVITS CENTER	54209752.0
19	KIPS BAY	1256027566.8
20	LITTLE ITALY	16006453.9
21	LOWER EAST SIDE	9688493.6
22	MANHATTAN VALLEY	13102423.0
23	MIDTOWN CBD	125324720.1
24	MIDTOWN EAST	110546703.3
25	MIDTOWN WEST	104328098.9
26	MORNINGSIDE HEIGHTS	12575604.7
27	MURRAY HILL	44301138.1
28	SOHO	45129842.7
29	SOUTHBRIDGE	14714073.5
30	TRIBECA	6278728.6



We can conclude from the above graph that we have better mean for MIDTOWN BCD as there are more and higher values for that neighborhood and also the the values are distributed without any significant SD which means lower outliers.And we can say that for KIPS BAY we are getting one outlier as it's value is much higher than the ordinary values

Q-2a)

Code

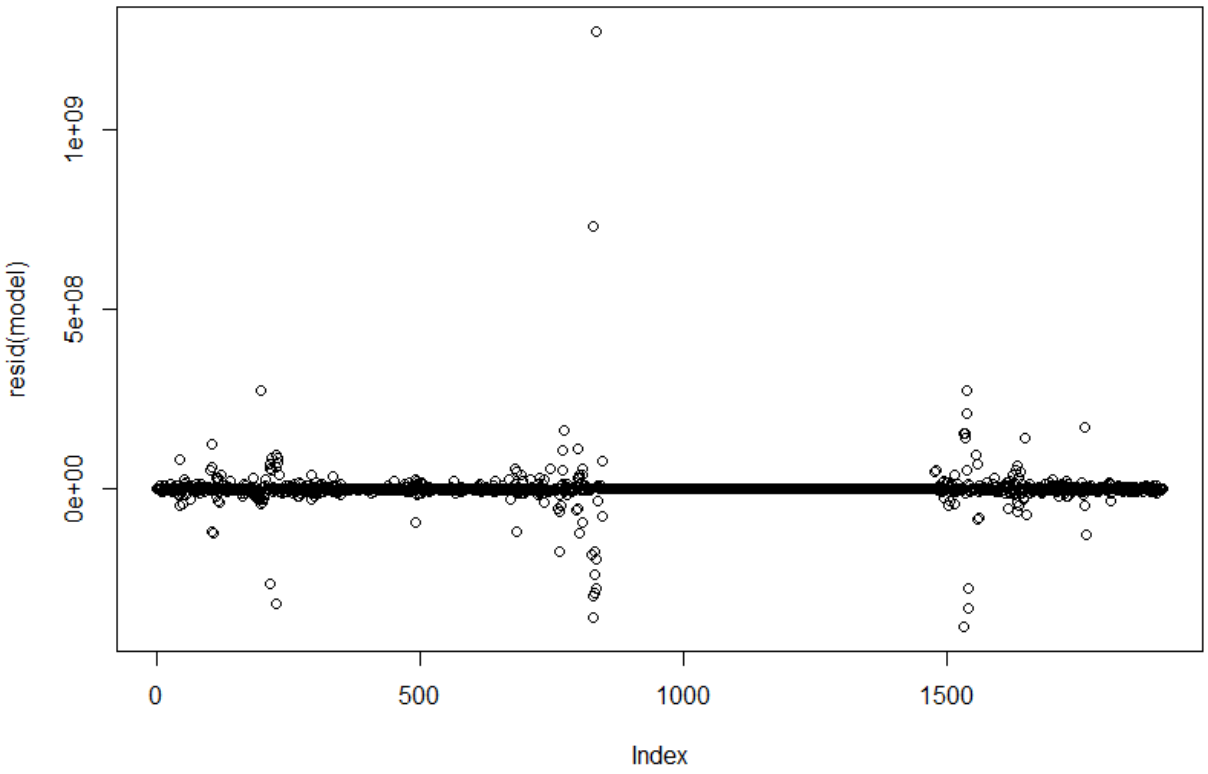
```
rs_mandf_modify.sale<-rs_mandf_modify[which(rs_mandf_modify$GROSS.SQUARE.FEET>0 &
rs_mandf_modify$LAND.SQUARE.FEET>0 & rs_mandf_modify$SALE.PRICE>0),]


model<-
lm((SALE.PRICE)~(GROSS.SQUARE.FEET)+(LAND.SQUARE.FEET)+factor(NEIGHBORHOOD)*factor(BUILDING.
CLASS.CATEGORY),data=rs_mandf_modify.sale)

summary(model)
```

Output

****Large number of rows for residuals and coefficient . So I just put the overall analysis**

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 49540000 on 1616 degrees of freedom
```

```
Multiple R-squared:  0.8574,   Adjusted R-squared:  0.8317
F-statistic: 33.29 on 292 and 1616 DF,  p-value: < 2.2e-16
```



Conclusion

We can see from the overall analysis that we got

`Multiple R-squared:  0.8574,    Adjusted R-squared:  0.8317`

Values. By seeing such promising R-squared value we can say that we can use this regression model for predicting sales price from rest of the predictors.

Q-2b)

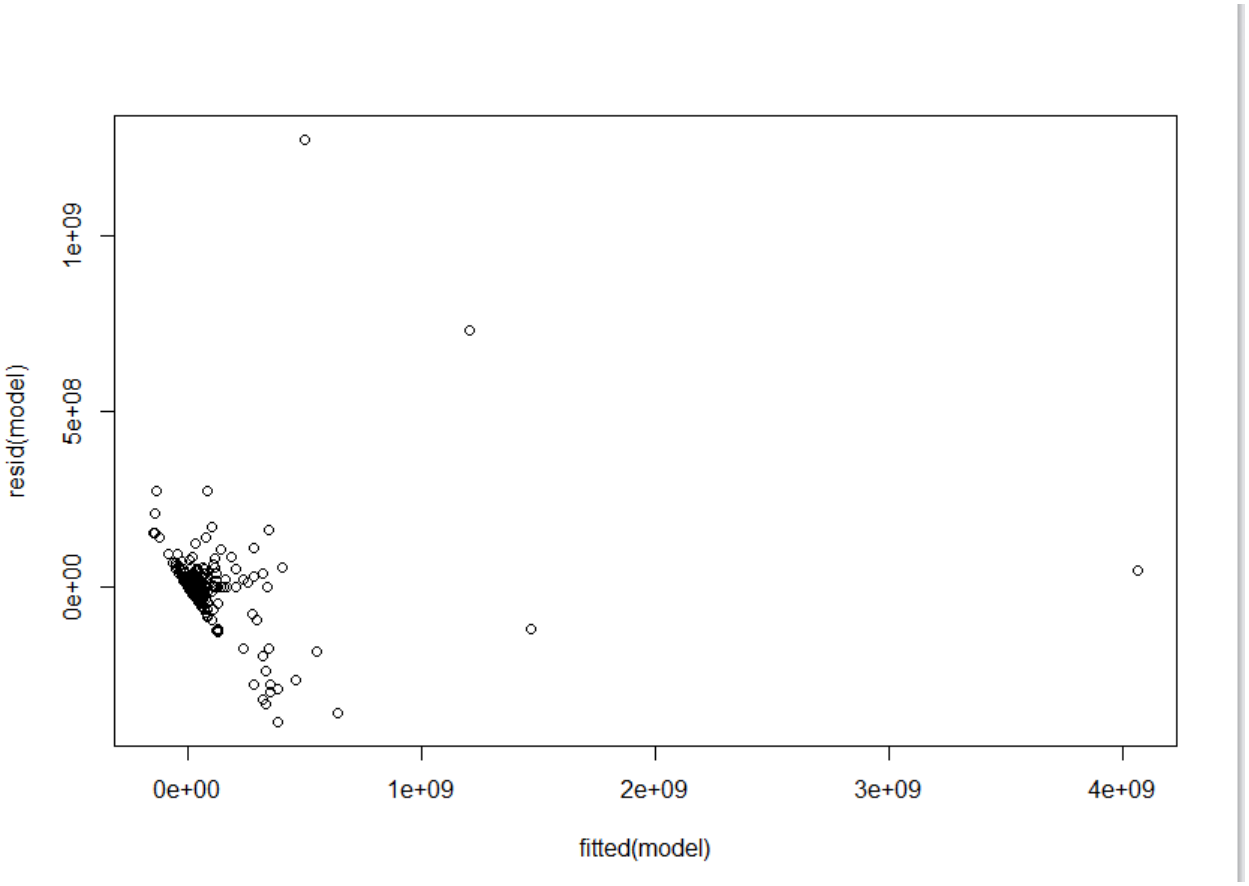
Code

```
rs_mandf_modify.sale<-rs_mandf_modify[which(rs_mandf_modify$GROSS.SQUARE.FEET>0 &
rs_mandf_modify$LAND.SQUARE.FEET>0 & rs_mandf_modify$SALE.PRICE>0),]
```

Fenil Tailor Netid-fst216

```
model<-
lm((SALE.PRICE)~(GROSS.SQUARE.FEET)+(LAND.SQUARE.FEET)+factor(NEIGHBORHOOD)*factor(BUILDING.
CLASS.CATEGORY),data=rs_mandf_modify.sale)

summary(model)

plot(resid(model)~fitted(model))
```

Graph



Conclusion

We know that the residual = 0 line corresponds to the estimated regression line for the model that we have build. By seeing the graph for residuals vs fitted model we can see that most of the SALE.PRICE value is in close proximity of regression line.We can also conclude that there are 5 outliers as the data points are far from regression line.

Q-2C)

Code:-

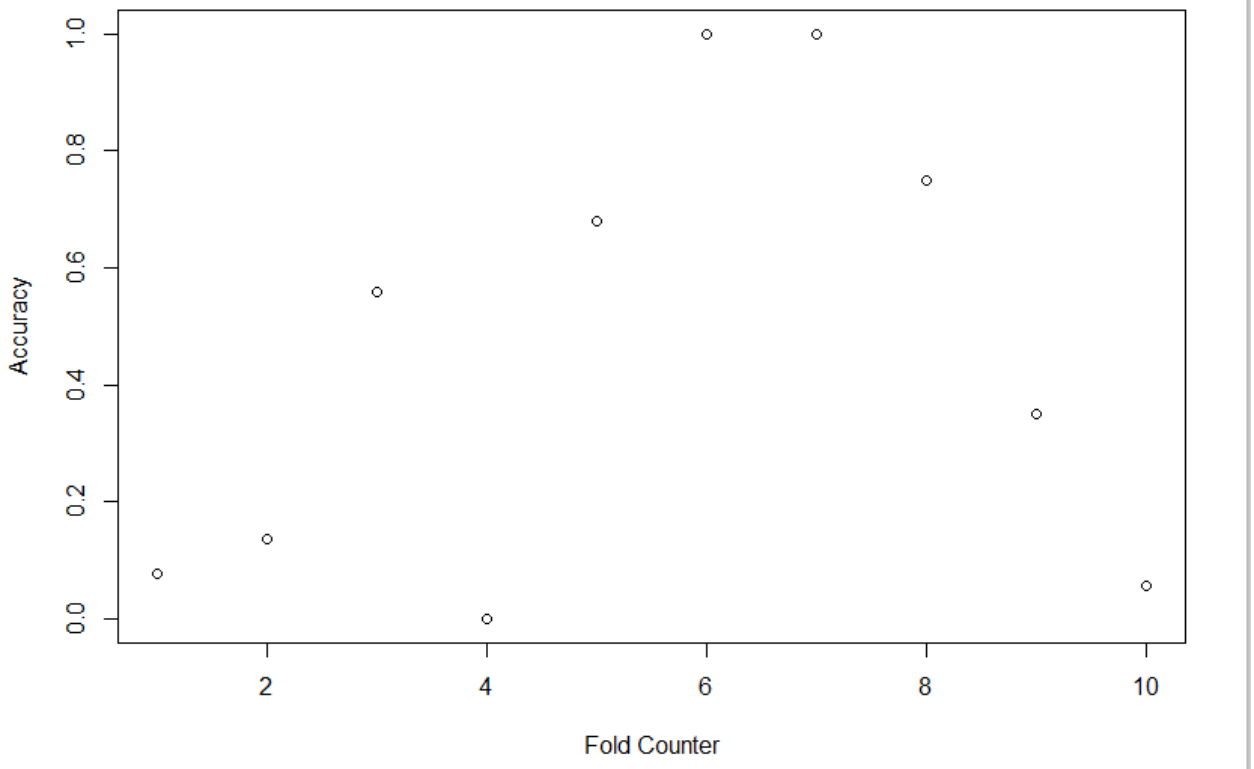
```
#Predicting neighborhood using following column


rs_mandf_frame1<-data.frame(rs_mandf_modify.sale[,9])
#View(rs_mandf_frame1)
#rs_mandf_frame1<-rs_mandf_frame1[sample(nrow(rs_mandf_frame1)),]


#created response variable dataframecoll using NEIGHBORHOOD


dataframecoll=data.frame(rs_mandf_modify.sale$NEIGHBORHOOD)
#View(rs_mandf_frame1)


#Created 10 equally size folds
folds <- cut(seq(1,nrow(rs_mandf_frame1)),breaks=10,labels=FALSE)


#Created vector Accaracyvector for storing accuracy and icoll to store iteration var i
Accuracyvector<-integer()
icoll<-integer()


#Performed 10 fold cross validation


for(i in 1:10){
  icoll<-c(icoll,i)
#Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- rs_mandf_frame1[testIndexes, ]
#print(length(testData$SALE.PRICE))
  trainData <- rs_mandf_frame1[-testIndexes, ]
#print(length(trainData$SALE.PRICE))
```

```
  trainres<-(dataframecoll[-testIndexes,])
  testres<-(dataframecoll[testIndexes,])
  library(e1071)
  library(class)
  library(caret)
  knn_req<-knn(train=data.frame(trainData),test=data.frame(testData),cl=trainres,k=3)
#print(knn_req)  Here I have created confusion matrix that give prediction for all the folds
  GetAccuracy<-confusionMatrix(testres,knn_req)
#print(confusionMatrix(testres,knn_req))
#print(GetAccuracy$overall[1])
  Accuracyvector<-c(Accuracyvector,GetAccuracy$overall[1])
}

#ploting acccuracy versus fold count

plot(icoll,Accuracyvector,xlab="Fold Counter",ylab="Accuracy")
sum=0
for(acc in Accuracyvector)
{
 sum=sum+acc
}

#printed the average accuracy of all the 10 folds

print(sum/length(Accuracyvector))
```

Output

```
[1] 0.4612565
```

The above value shows the average accuracy for fold values from 1 to 10



Conclusion

Below I have printed the over all statistics for confusion matrix for fold 3 for the observation and we can also see from the graph that the accuracy is matching with that plot.From the confusionMatrix we can easily said about the errors if we don't get all the prediction values in diagonal of the matrix.So basically we can say that which observation of the testdata goes into which prediction class by seeing the confusionMatrix and we can then calculate the error

```
Overall Statistics

               Accuracy : 0.576
                 95% CI : (0.0291, 0.1007)
    No Information Rate : 0.288
    P-Value [Acc > NIR] : 1

                  Kappa : 0.536
 Mcnemar's Test P-Value : NA
```

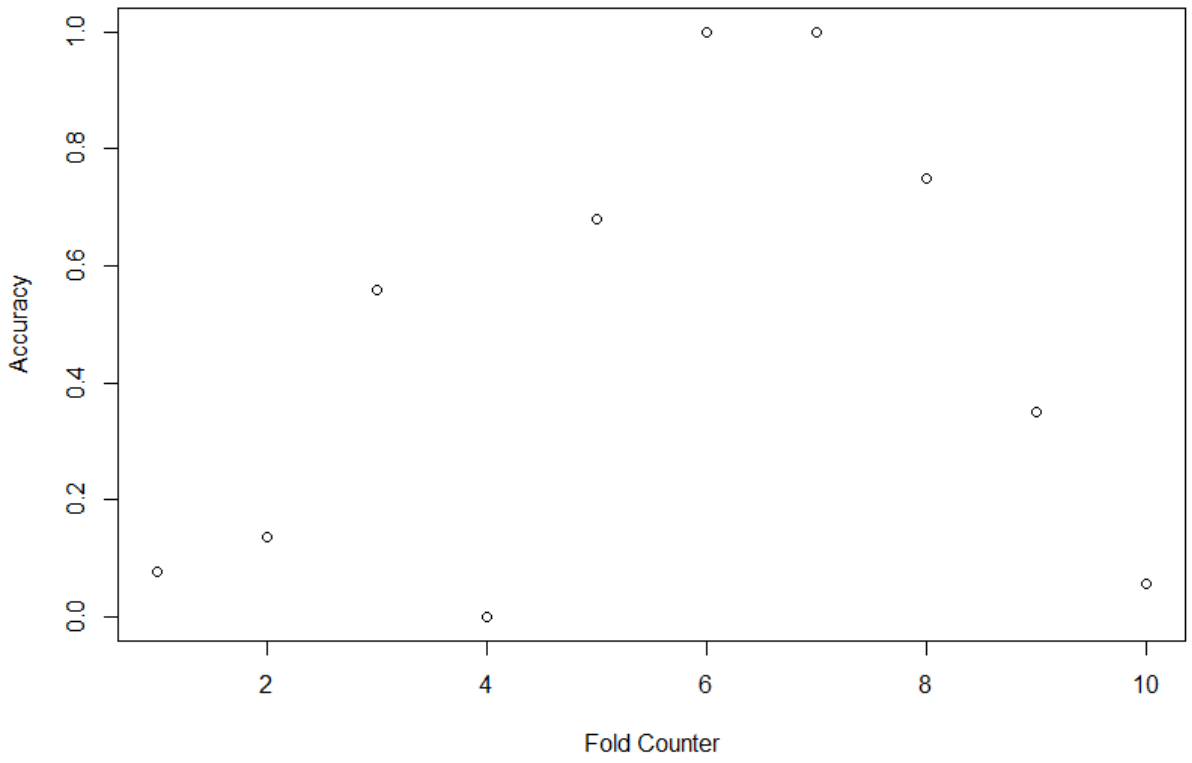
Q-2d)

Code

```
 #Predicting neighborhood using following column
rs_mandf_frame1<-data.frame(rs_mandf_modify.sale[,9])
#View(rs_mandf_frame1)
#rs_mandf_frame1<-rs_mandf_frame1[sample(nrow(rs_mandf_frame1)),]

#created response variable dataframecoll using NEIGHBORHOOD

dataframecoll=data.frame(rs_mandf_modify.sale$NEIGHBORHOOD)
#View(rs_mandf_frame1)

#Created 10 equally size folds
folds <- cut(seq(1,nrow(rs_mandf_frame1)),breaks=10,labels=FALSE)

#Created vector Accaracyvector for storing accuracy and icoll to store iteration var i
Accuracyvector<-integer()
icoll<-integer()

#Performed 10 fold cross validation

for(i in 1:10){
   icoll<-c(icoll,i)
#Segement your data by fold using the which() function
   testIndexes <- which(folds==i,arr.ind=TRUE)
   testData <- rs_mandf_frame1[testIndexes, ]
#print(length(testData$SALE.PRICE))
   trainData <- rs_mandf_frame1[-testIndexes, ]
#print(length(trainData$SALE.PRICE))
```

```
   trainres<-(dataframecoll[-testIndexes,])

   testres<-(dataframecoll[testIndexes,])

   knn_req<-knn(train=data.frame(trainData),test=data.frame(testData),cl=trainres,k=3)

#print(knn_req)  Here I have created confusion matrix that give prediction for all the folds

   GetAccuracy<-confusionMatrix(testres,knn_req)

#print(confusionMatrix(testres,knn_req))

#print(GetAccuracy$overall[1])

   Accuracyvector<-c(Accuracyvector,GetAccuracy$overall[1])

}


#ploting acccuracy versus fold count


plot(icoll,Accuracyvector,xlab="Fold Counter",ylab="Accuracy")

sum=0

for(acc in Accuracyvector)

{

  sum=sum+acc

}


#printed the average accuracy of all the 10 folds


print(sum/length(Accuracyvector))
```

Graph



Conclusion

From the graph we can see that for the first 5 folds the accuracy is continuously increasing for k=3 and then after the accuracy is continuously decreasing. From this observation we can say that there are more likely and similar values for the predictors for the 5^{th} and 6^{th} fold than any other fold.We can also say that predictors values are not that good and repetitive in sense for 1^{st} ,4^{th} ,2^{nd} and 10^{th} observation for the purpose of learning model.

Q-2e)

As we can see that our overall average accuracy for 10 fold cross validation is around 46% which is not significant. So our try should be to improve the accuracy of prediction by creating the folds so that there are more similar and significant values in the same fold .Thus, model can learn from training data set very efficiently and can predict the response variable for test data with more accuracy

Also from the graph we can see that for the first 5 folds the accuracy is continuously increasing for k=3 and then after the accuracy is continuously decreasing.From this observation we can say that there are more likely and similar values for the predictors for the 5^{th} and 6^{th} fold than any other fold.We can also say that predictors values are not that good and repetitive in sense for 1^{st} ,2^{nd} and 10^{th} observation for the purpose of learning model.

Q-3a)

Code

#we perform scaling on the variables to perform PCA

scaled_sales <- data.frame(scale(rs_mandf_modify.sale[,c(13,14,18)], scale = TRUE))

we perform PCA analyses on the data

PCA = prcomp(~GROSS.SQUARE.FEET+LAND.SQUARE.FEET+SALE.PRICE,data=scaled_sales,na.action =na.omit,scale=T)

str(PCA)

summary(PCA)

predict(PCA)

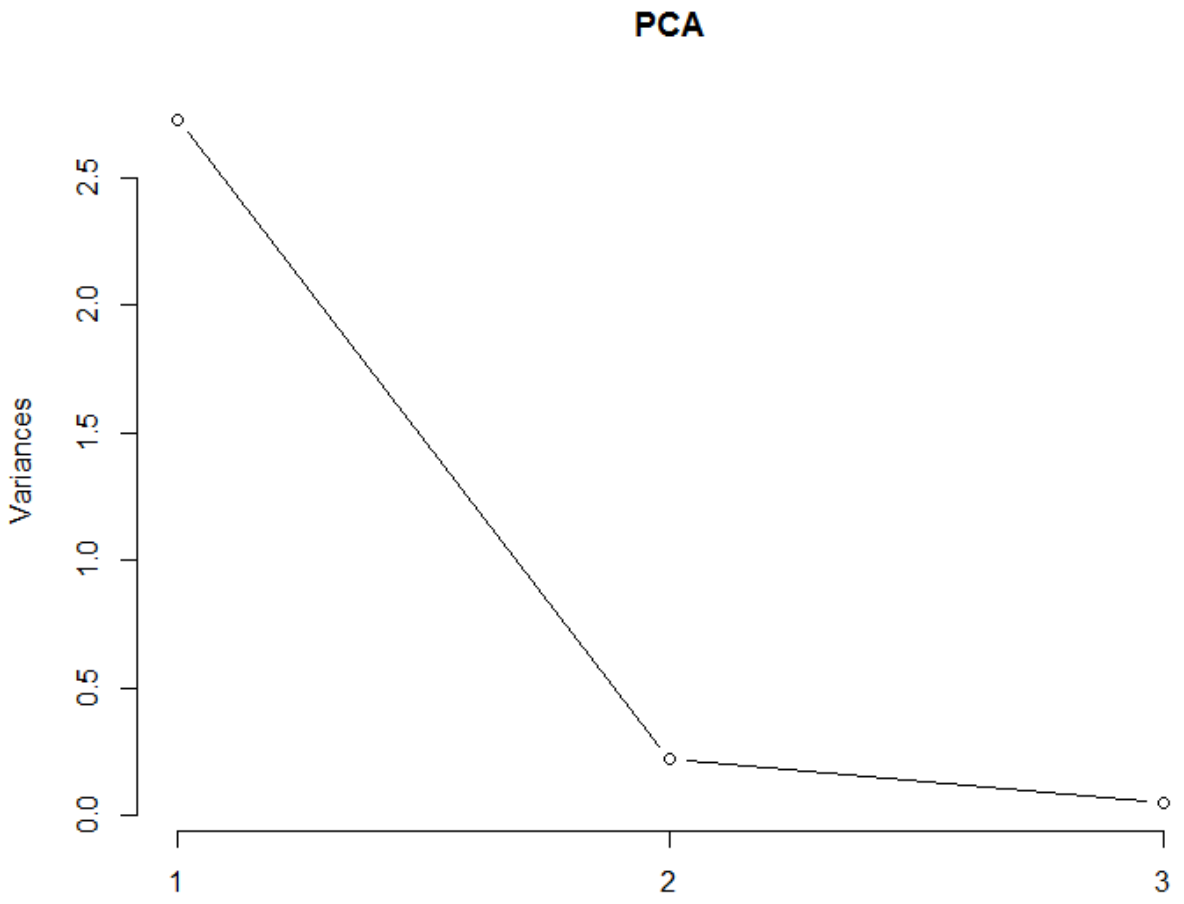
plot(PCA,type="l")

Output

```
> str(PCA)

List of 6
 $ sdev    : num [1:3] 1.651 0.471 0.226
 $ rotation: num [1:3, 1:3] 0.59 0.583 0.558 -0.308 -0.477 ...
  ..- attr(*, "dimnames")=List of 2
```

```
  .. ..$ : chr [1:3] "GROSS.SQUARE.FEET" "LAND.SQUARE.FEET" "SALE.PRICE"
  .. ..$ : chr [1:3] "PC1" "PC2" "PC3"
 $ center  : Named num [1:3] -2.89e-17 9.18e-18 -1.57e-17
  ..- attr(*, "names")= chr [1:3] "GROSS.SQUARE.FEET" "LAND.SQUARE.FEET" "SAL
E.PRICE"
 $ scale   : logi FALSE
 $ x       : num [1:1909, 1:3] -0.258 -0.273 -0.276 -0.185 -0.152 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:1909] "1" "2" "3" "5" ...
  .. ..$ : chr [1:3] "PC1" "PC2" "PC3"
 $ call    : language prcomp(formula = ~GROSS.SQUARE.FEET + LAND.SQUARE.FEET
+ SALE.PRICE, data = scaled_sales, na.action = na.omit)
 - attr(*, "class")= chr "prcomp"
```

```
> summary(PCA)
Importance of components:
                          PC1     PC2     PC3
Standard deviation     1.651 0.47089 0.22616
Proportion of Variance 0.909 0.07391 0.01705
Cumulative Proportion  0.909 0.98295 1.00000
```



Conclusion

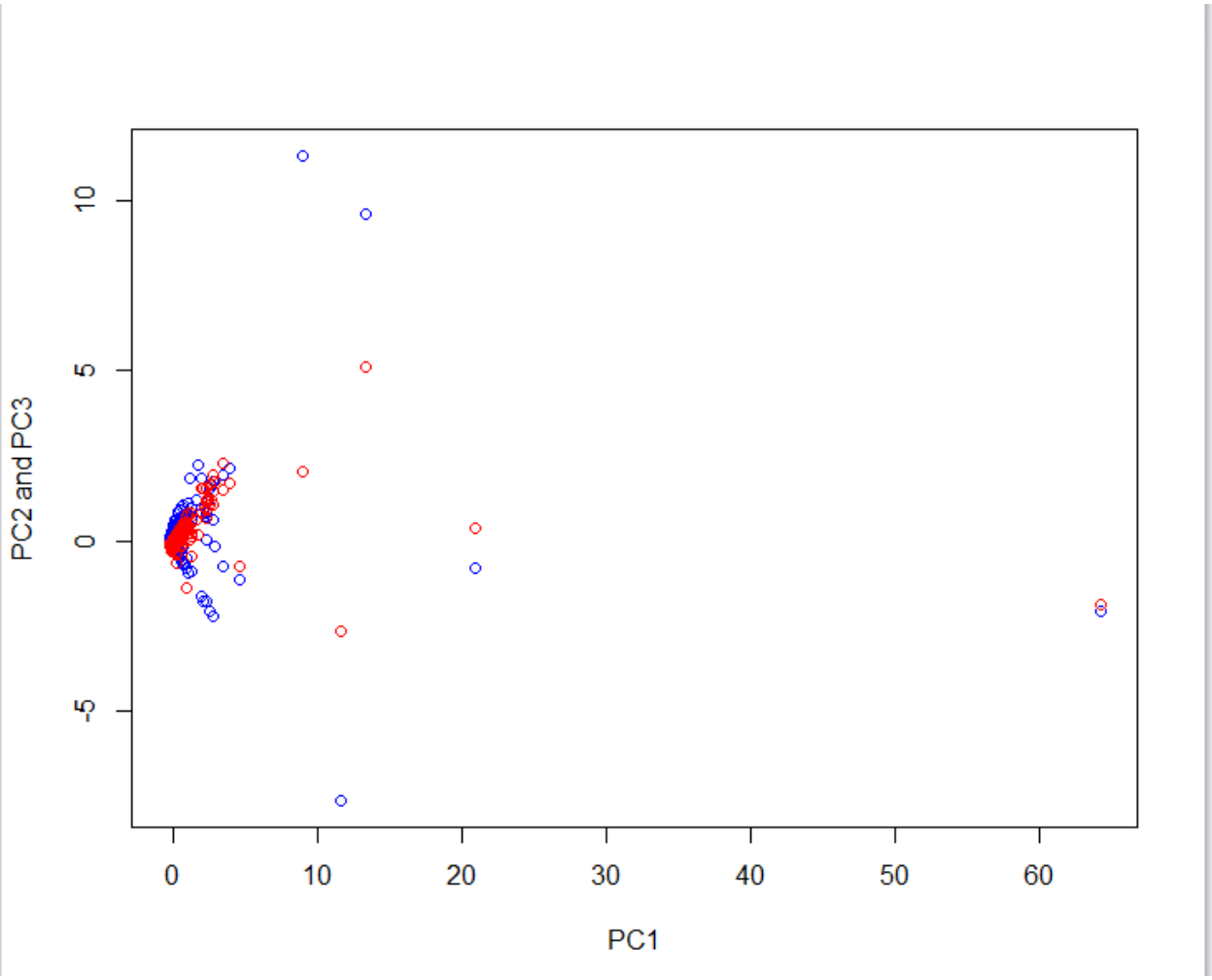
Have done the PCA using prcomp function and I have transformed standard deviation equal to one by using prcomp function which has SCALE=T in it's argument.

Q-3B)

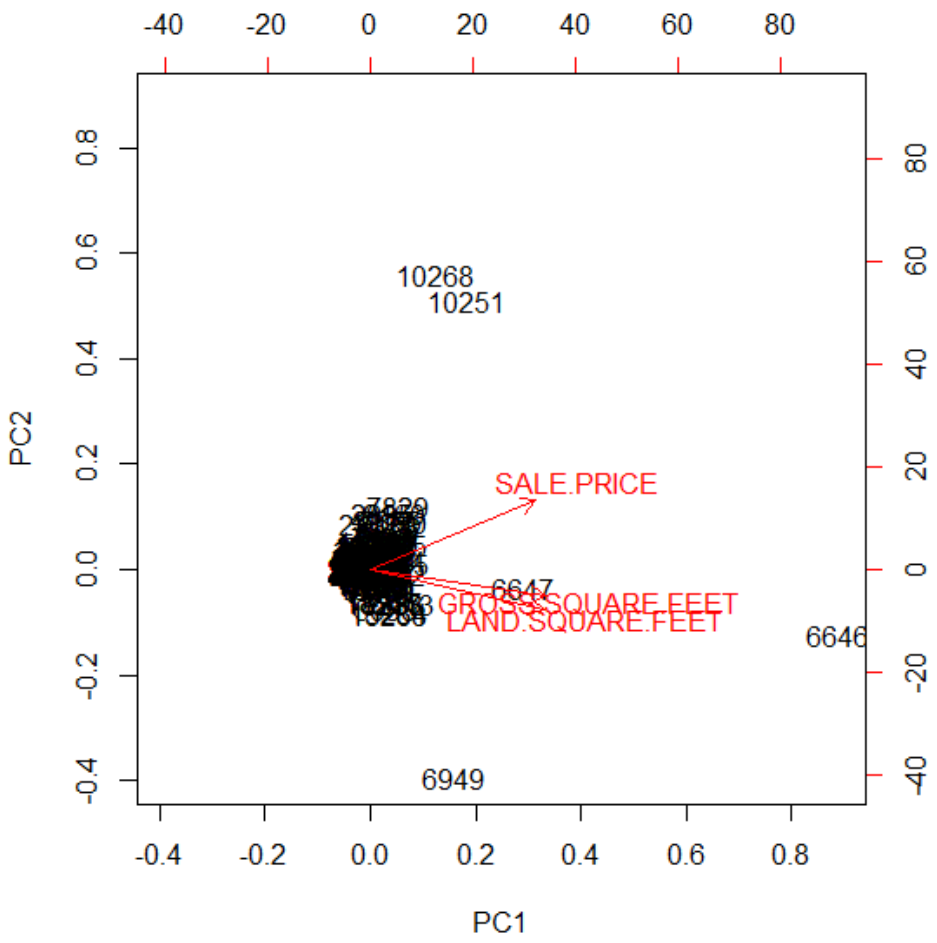
Code

```
# first we plot the first two vectors

scr=as.matrix(scaled_sales[,c(1:3)])%*%PCA$rot[,1:2]

plot(scr,col="blue",ylab="PC2 and PC3")

#now we plot the first and third vector

scr_1=as.matrix(scaled_sales[,c(1:3)])%*%PCA$rot[,c(1,3)]

points(scr_1,col="red")


#we can visualize the 3 vectors using an inbuilt function of r biplot


biplot(PCA)
```

Output



Bigplot Graph



Conclusion

For the first plot between rotational vector 1 and rotational vector 2,I have taken blue color

And for the second plot between rotational vector 1 and rotational vector 3,I have taken red color.

From the graphs we can say that all the three components

(SALE.PRICE,GROSS.SQUARE_FEET,LAND.SQUARE_FEET) on which we have apply prcomp are perpendicular to each other.And we can see from the summary of the prcomp function that variance of all the components are nearly equal to zero.