DATA 605 - Final Exam

Omer Ozeren

Table of Contents

Problem 1 : 1
  Probability 2
  A: $P(X > x|X > y)$ 2
  B: $P(X > x, Y > y)$ 2
  C: $P(X < x|X > y)$ 2
5 points 3
5 points 4
Problem 2 4
  Load Data from Kaggle 5
5 points 6
  ScatterPlot 15
  Correlation matrix 16
  Hypothesis and 80% confidence interval 17
  Familywise Error 19
5 points 19
Calculus-Based Probability & Statistics 20
  Univariate distribution of LotArea 21
CDF 23
Modeling 23
  I'll now do a stepwise regression based on ACI criterion 25
  Residual Analysis 26
  Export submission 27
  Kaggle Submission 27

Problem 1 :

Using R, generate a random variable X that has 10,000 random uniform numbers from 1 to N, where N can be any number of your choosing greater than or equal to 6. Then generate a random variable Y that has 10,000 random normal numbers with a mean of mean=std=(N+1)/2

```
#10,000 random uniform numbers from 1 to N
N=9
# 10,000 random uniform numbers from 1 to N
X = runif(10000, 1,N)
# 10,000 random normal numbers with a mean of mean=std=(N+1)/2
mu <- (N+1)/2
std <- (N+1)/2
Y = rnorm(10000, mean = mu,sd = std)
```

Probability

Calculate as a minimum the below probabilities a through c. Assume the small letter "x" is estimated as the median of the X variable, and the small letter "y" is estimated as the 1st quartile of the Y variable. Interpret the meaning of all probabilities

```
XY<- cbind(X,Y)
var <- nrow(XY)
x <- median(X)
y <- quantile(Y, 0.25,names=FALSE)
```

A: $P(X > x|X > y)$

$$P(X > x|X > y) = \frac{P(X > x \text{ } and \text{ } X > y)}{P(X > y)}$$

```
XGy <- length(which(X>y))
XGy_XGx <- length(which(X>y & X>x))
XGy_XGx/XGy

## [1] 0.5425347
```

B: $P(X > x, Y > y)$

We know the statistics of half of the values in X are above the median, and 75% of the values in Y are above the first quartile

$$P(X > x, Y > Y) = P(X > x \text{ } and \text{ } Y > y)$$

$$P(X > x) = 0.5$$

$$P(Y > y) = 0.75$$

$$P(X > x \text{ } and \text{ } Y > y) = (0.5)(0.75) = 0.375$$

C: $P(X < x|X > y)$

```
XGy <- length(which(X>y))
XGy_xGX <- length(which(X>y & X<x))
```

```r
XGy_xGX/XGy

## [1] 0.4574653
```

5 points.

Investigate whether P(X>x and Y>y)=P(X>x)P(Y>y) by building a table and evaluating the marginal and joint probabilities**

```r
tab <- c(sum(X<x & Y < y),
       sum(X < x & Y == y),
       sum(X < x & Y > y))
tab <- rbind(tab,
              c(sum(X==x & Y < y),
       sum(X == x & Y == y),
       sum(X == x & Y > y))

             )
tab <- rbind(tab,
              c(sum(X>x & Y < y),
       sum(X > x & Y == y),
       sum(X > x & Y > y))
             )
tab <- cbind(tab, tab[,1] + tab[,2] + tab[,3])
tab <- rbind(tab, tab[1,] + tab[2,] + tab[3,])
colnames(tab) <- c("Y<y", "Y=y", "Y>y", "Total")
rownames(tab) <- c("X<x", "X=x", "X>x", "Total")
knitr::kable(tab)
```

	Y<y	Y=y	Y>y	Total
X<x	1263	0	3737	5000
X=x	0	0	0	0
X>x	1237	0	3763	5000
Total	2500	0	7500	10000

```r
# P(X>x and Y>y)
3747/10000

## [1] 0.3747

#P(X>x)P(Y>y)
((5000)/10000)*(7500/10000)

## [1] 0.375
```

we can see that the condition holds since P(X>x and Y>y) = 0.3754 and P(X>x)P(Y>y) = 0.375 are approximately equal.

5 points.

Check to see if independence holds by using Fisher's Exact Test and the Chi Square Test. What is the difference between the two? Which is most appropriate?

Fisher's Exact Test

```
fisher.test(table(X>x,Y>y))

##
##  Fisher's Exact Test for Count Data
##
## data:  table(X > x, Y > y)
## p-value = 0.5637
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.9381439 1.1267272
## sample estimates:
## odds ratio
##   1.028128
```

The p-value is greater than zero we don't reject the null hypothesis. Two events are independent.

The Chi Square Test

```
chisq.test(table(X>x,Y>y))

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(X > x, Y > y)
## X-squared = 0.33333, df = 1, p-value = 0.5637
```

The p-value is greeter than zero we don't reject the null hypothesis. Two events are independent.

Fisher's exact test the null of independence of rows and columns in a contingency table with fixed marginals.

Chi-squared test tests contingency table tests and goodness-of-fit tests.

Fisher's exact test is appropriate here. Since the contingency table are fixed here in the table.

Problem 2

You are to register for Kaggle.com (free) and compete in the House Prices: Advanced Regression Techniques competition. https://www.kaggle.com/c/house-prices-advanced-regression-techniques . I want you to do the following.

Load the libraries

```
library(readr)

## Warning: package 'readr' was built under R version 3.5.3

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages ----------------------------------------------------
------------------------------------------------------------------- tidyverse
1.2.1 --

## v ggplot2 3.1.0       v purrr   0.3.0
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.3.1
## v ggplot2 3.1.0       v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.3

## Warning: package 'tidyr' was built under R version 3.5.3

## Warning: package 'dplyr' was built under R version 3.5.3

## Warning: package 'forcats' was built under R version 3.5.3

## -- Conflicts -------------------------------------------------------------
-------------------------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 3.5.3
```

Load Data from Kaggle

```
# Load training data from GitHub
path <-
('https://raw.githubusercontent.com/omerozeren/DATA605/master/Final_Exam/trai
n.csv')
con <- file(path, open="r")
train <- read.csv(con, header=T, stringsAsFactors = F)
close(con)

# Load test data from GitHub
path <-
('https://raw.githubusercontent.com/omerozeren/DATA605/master/Final_Exam/test
.csv')
con <- file(path, open="r")
test <- read.csv(con, header=T, stringsAsFactors = F)
close(con)
```

5 points.

Descriptive and Inferential Statistics.

Provide univariate descriptive statistics and appropriate plots for the training data set. Provide a scatterplot matrix for at least **two** of the independent variables and the dependent variable. Derive a correlation matrix for any **three** quantitative variables in the dataset. Test the hypotheses that the correlations between each pairwise set of variables is 0 and provide an 80% confidence interval. Discuss the meaning of your analysis. Would you be worried about familywise error? Why or why not?

Summary of Train Data

```
summary(train)
```

```
##        Id           MSSubClass      MSZoning          LotFrontage    
##  Min.   :   1.0   Min.   : 20.0   Length:1460        Min.   : 21.00  
##  1st Qu.: 365.8   1st Qu.: 20.0   Class :character   1st Qu.: 59.00  
##  Median : 730.5   Median : 50.0   Mode  :character   Median : 69.00  
##  Mean   : 730.5   Mean   : 56.9                      Mean   : 70.05  
##  3rd Qu.:1095.2   3rd Qu.: 70.0                      3rd Qu.: 80.00  
##  Max.   :1460.0   Max.   :190.0                      Max.   :313.00  
##                                                      NA's   :259     
##     LotArea          Street              Alley             LotShape        
##  Min.   :  1300   Length:1460        Length:1460        Length:1460       
##  1st Qu.:  7554   Class :character   Class :character   Class :character  
##  Median :  9478   Mode  :character   Mode  :character   Mode  :character  
##  Mean   : 10517                                                           
##  3rd Qu.: 11602                                                           
##  Max.   :215245                                                           
##                                                                           
##  LandContour         Utilities          LotConfig        
##  Length:1460        Length:1460        Length:1460       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##                                                          
##   LandSlope         Neighborhood        Condition1       
##  Length:1460        Length:1460        Length:1460       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##                                                          
##   Condition2          BldgType          HouseStyle         OverallQual    
##  Length:1460        Length:1460        Length:1460        Min.   : 1.000  
##  Class :character   Class :character   Class :character   1st Qu.: 5.000  
```

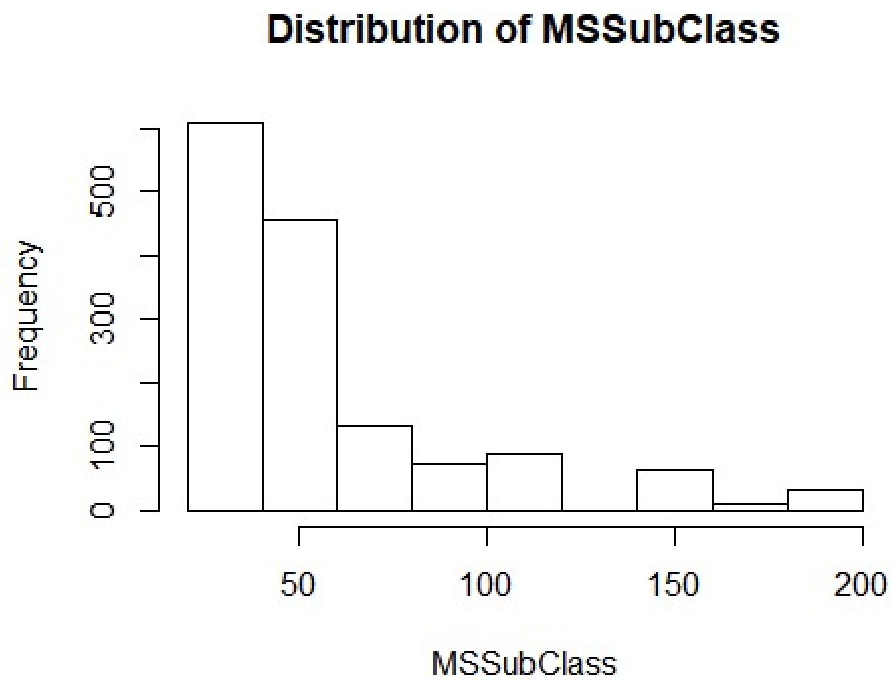
```
##  Mode  :character   Mode  :character   Mode  :character   Median : 6.000  
##                                                           Mean   : 6.099  
##                                                           3rd Qu.: 7.000  
##                                                           Max.   :10.000  
##                                                                           
##   OverallCond      YearBuilt     YearRemodAdd   RoofStyle        
##  Min.   :1.000   Min.   :1872   Min.   :1950   Length:1460       
##  1st Qu.:5.000   1st Qu.:1954   1st Qu.:1967   Class :character  
##  Median :5.000   Median :1973   Median :1994   Mode  :character  
##  Mean   :5.575   Mean   :1971   Mean   :1985                     
##  3rd Qu.:6.000   3rd Qu.:2000   3rd Qu.:2004                     
##  Max.   :9.000   Max.   :2010   Max.   :2010                     
##                                                                  
##    RoofMatl         Exterior1st        Exterior2nd       
##  Length:1460        Length:1460        Length:1460       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##                                                          
##   MasVnrType          MasVnrArea      ExterQual          ExterCond        
##  Length:1460        Min.   :   0.0   Length:1460        Length:1460       
##  Class :character   1st Qu.:   0.0   Class :character   Class :character  
##  Mode  :character   Median :   0.0   Mode  :character   Mode  :character  
##                     Mean   : 103.7                                        
##                     3rd Qu.: 166.0                                        
##                     Max.   :1600.0                                        
##                     NA's   :8                                             
##   Foundation          BsmtQual           BsmtCond        
##  Length:1460        Length:1460        Length:1460       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##                                                          
##  BsmtExposure       BsmtFinType1         BsmtFinSF1     BsmtFinType2      
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460       
##  Class :character   Class :character   1st Qu.:   0.0   Class :character  
##  Mode  :character   Mode  :character   Median : 383.5   Mode  :character  
##                                        Mean   : 443.6                     
##                                        3rd Qu.: 712.2                     
##                                        Max.   :5644.0                     
##                                                                           
##    BsmtFinSF2        BsmtUnfSF       TotalBsmtSF       Heating         
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0.0   Length:1460       
##  1st Qu.:   0.00   1st Qu.: 223.0   1st Qu.: 795.8   Class :character  
##  Median :   0.00   Median : 477.5   Median : 991.5   Mode  :character  
##  Mean   :  46.55   Mean   : 567.2   Mean   :1057.4                     
```

```
##  3rd Qu.:   0.00   3rd Qu.: 808.0   3rd Qu.:1298.2
##  Max.   :1474.00   Max.   :2336.0   Max.   :6110.0
##
##   HeatingQC          CentralAir         Electrical          X1stFlrSF
##  Length:1460        Length:1460        Length:1460        Min.   : 334
##  Class :character   Class :character   Class :character   1st Qu.: 882
##  Mode  :character   Mode  :character   Mode  :character   Median :1087
##                                                           Mean   :1163
##                                                           3rd Qu.:1391
##                                                           Max.   :4692
##
##    X2ndFlrSF     LowQualFinSF       GrLivArea     BsmtFullBath
##  Min.   :   0   Min.   :  0.000   Min.   : 334   Min.   :0.0000
##  1st Qu.:   0   1st Qu.:  0.000   1st Qu.:1130   1st Qu.:0.0000
##  Median :   0   Median :  0.000   Median :1464   Median :0.0000
##  Mean   : 347   Mean   :  5.845   Mean   :1515   Mean   :0.4253
##  3rd Qu.: 728   3rd Qu.:  0.000   3rd Qu.:1777   3rd Qu.:1.0000
##  Max.   :2065   Max.   :572.000   Max.   :5642   Max.   :3.0000
##
##   BsmtHalfBath        FullBath        HalfBath       BedroomAbvGr
##  Min.   :0.00000   Min.   :0.000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000
##  Median :0.00000   Median :2.000   Median :0.0000   Median :3.000
##  Mean   :0.05753   Mean   :1.565   Mean   :0.3829   Mean   :2.866
##  3rd Qu.:0.00000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :2.00000   Max.   :3.000   Max.   :2.0000   Max.   :8.000
##
##   KitchenAbvGr   KitchenQual         TotRmsAbvGrd     Functional
##  Min.   :0.000   Length:1460        Min.   : 2.000   Length:1460
##  1st Qu.:1.000   Class :character   1st Qu.: 5.000   Class :character
##  Median :1.000   Mode  :character   Median : 6.000   Mode  :character
##  Mean   :1.047                      Mean   : 6.518
##  3rd Qu.:1.000                      3rd Qu.: 7.000
##  Max.   :3.000                      Max.   :14.000
##
##    Fireplaces    FireplaceQu         GarageType         GarageYrBlt
##  Min.   :0.000   Length:1460        Length:1460        Min.   :1900
##  1st Qu.:0.000   Class :character   Class :character   1st Qu.:1961
##  Median :1.000   Mode  :character   Mode  :character   Median :1980
##  Mean   :0.613                                         Mean   :1979
##  3rd Qu.:1.000                                         3rd Qu.:2002
##  Max.   :3.000                                         Max.   :2010
##                                                        NA's   :81
##  GarageFinish         GarageCars      GarageArea      GarageQual
##  Length:1460        Min.   :0.000   Min.   :   0.0   Length:1460
##  Class :character   1st Qu.:1.000   1st Qu.: 334.5   Class :character
##  Mode  :character   Median :2.000   Median : 480.0   Mode  :character
##                     Mean   :1.767   Mean   : 473.0
##                     3rd Qu.:2.000   3rd Qu.: 576.0
##                     Max.   :4.000   Max.   :1418.0
```

```
##
##   GarageCond         PavedDrive          WoodDeckSF       OpenPorchSF
##  Length:1460        Length:1460        Min.   :  0.00   Min.   :  0.00
##  Class :character   Class :character   1st Qu.:  0.00   1st Qu.:  0.00
##  Mode  :character   Mode  :character   Median :  0.00   Median : 25.00
##                                        Mean   : 94.24   Mean   : 46.66
##                                        3rd Qu.:168.00   3rd Qu.: 68.00
##                                        Max.   :857.00   Max.   :547.00
##
##  EnclosedPorch      X3SsnPorch       ScreenPorch        PoolArea
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :  0.00   Median :  0.00   Median :  0.00   Median :  0.000
##  Mean   : 21.95   Mean   :  3.41   Mean   : 15.06   Mean   :  2.759
##  3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :552.00   Max.   :508.00   Max.   :480.00   Max.   :738.000
##
##     PoolQC             Fence            MiscFeature
##  Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     MiscVal             MoSold           YrSold       SaleType
##  Min.   :    0.00   Min.   : 1.000   Min.   :2006   Length:1460
##  1st Qu.:    0.00   1st Qu.: 5.000   1st Qu.:2007   Class :character
##  Median :    0.00   Median : 6.000   Median :2008   Mode  :character
##  Mean   :   43.49   Mean   : 6.322   Mean   :2008
##  3rd Qu.:    0.00   3rd Qu.: 8.000   3rd Qu.:2009
##  Max.   :15500.00   Max.   :12.000   Max.   :2010
##
##  SaleCondition        SalePrice
##  Length:1460        Min.   : 34900
##  Class :character   1st Qu.:129975
##  Mode  :character   Median :163000
##                     Mean   :180921
##                     3rd Qu.:214000
##                     Max.   :755000
##
```

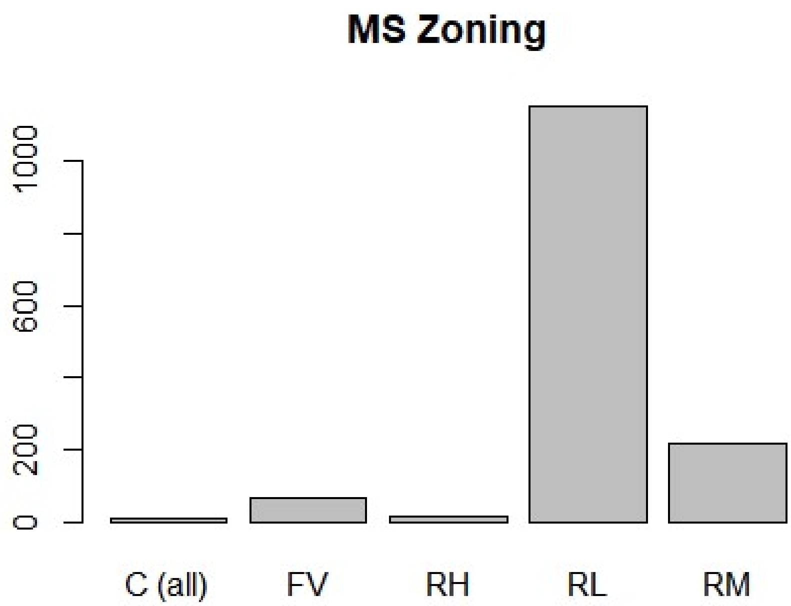
Plots of Train Data

```
hist(train$MSSubClass, main="Distribution of MSSubClass",xlab="MSSubClass")
```



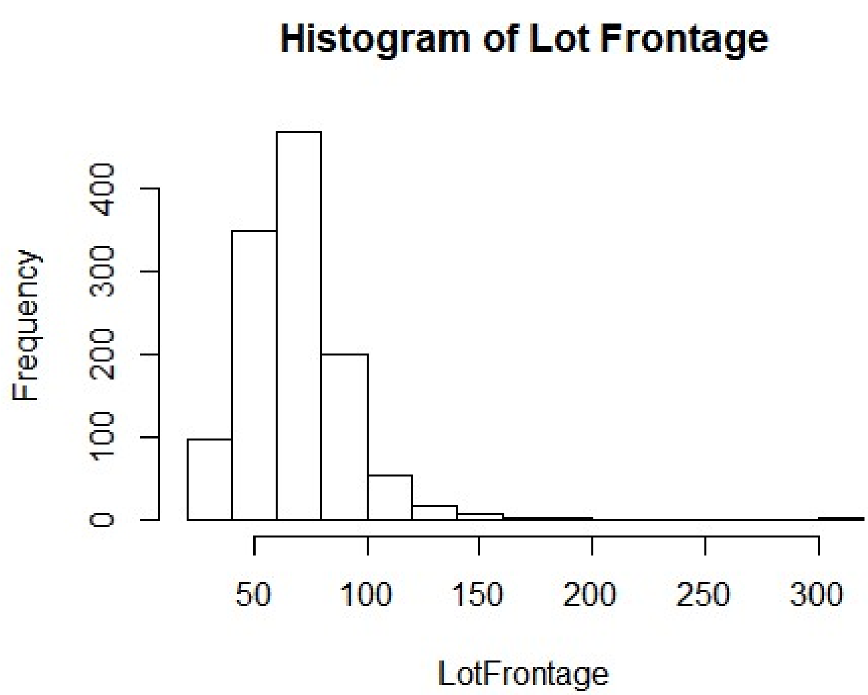
MSSubClass is left skewed.

```r
barplot(table(train$MSZoning), main="MS Zoning")
```



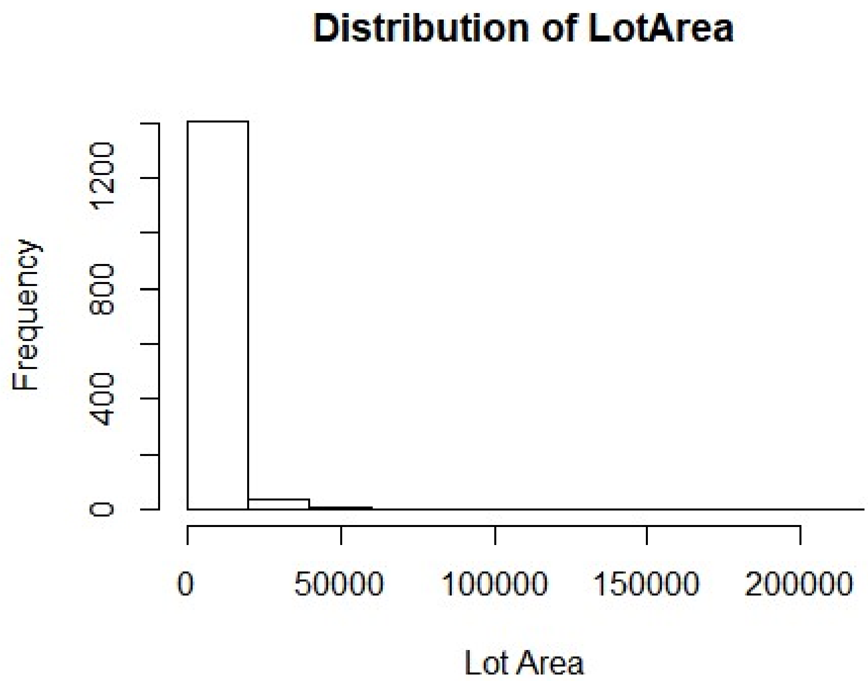
RL has the highest frequency , C lowest frequency.

```
hist(train$LotFrontage,main="Histogram of Lot Frontage",xlab="LotFrontage")
```



LotFrontage is left skewed.

```r
hist(train$LotArea,main="Distribution of LotArea",xlab="Lot Area")
```

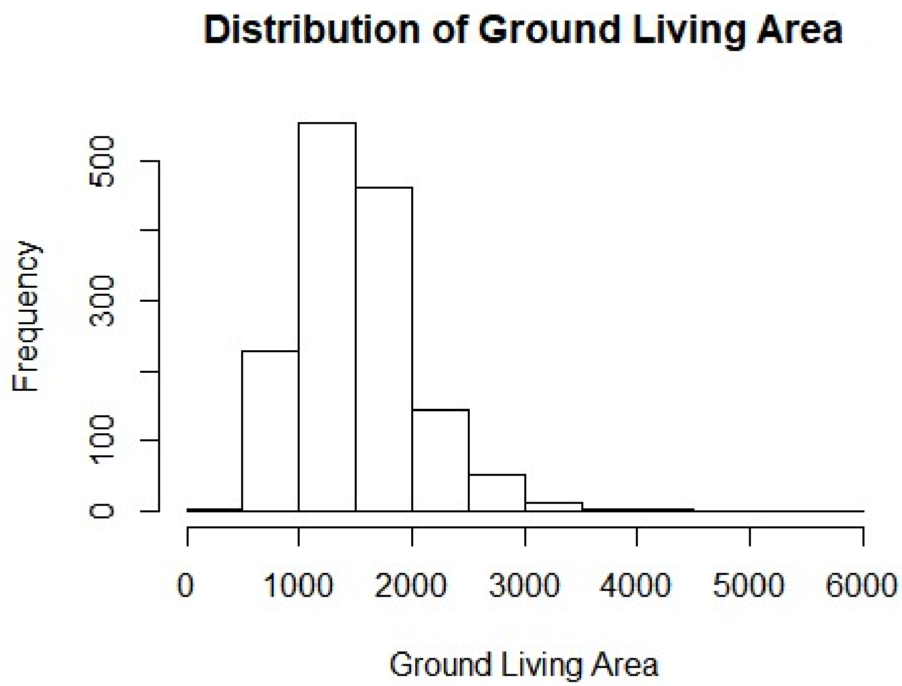


Lot Area is left skewed with very high small values.

```r
hist(train$SalePrice,main="Distribution of Sale Price",xlab="Sale Price")
```

Sales price is slightly approximately normally distributed. .

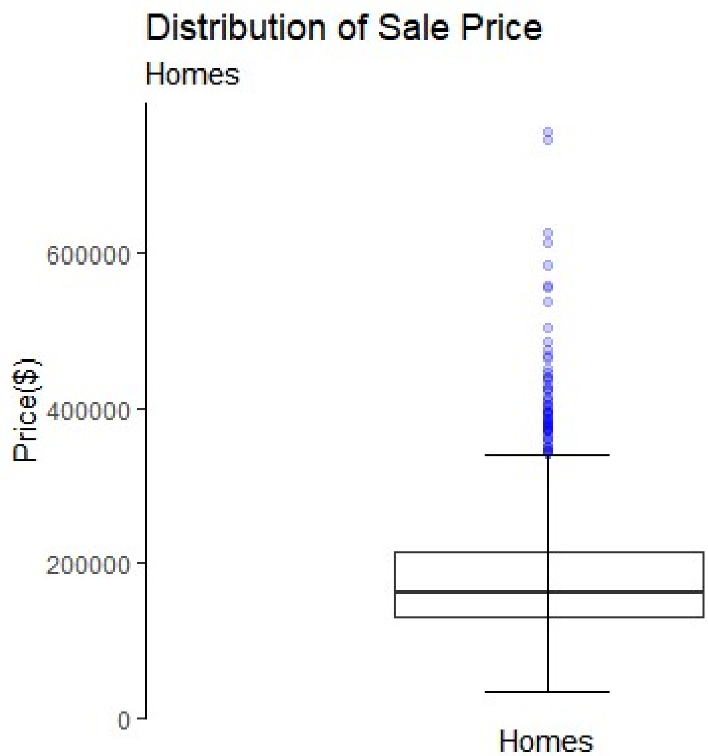
```
hist(train$GrLivArea,main="Distribution of Ground Living Area",xlab="Ground
Living Area")
```



Ground Living Area is approximately normally distributed.

Since the SalePrice column will be the target variable, we'll start there and look at how it is distributed.

```
# Plot SalePrice
train %>% ggplot(aes(y=SalePrice)) +
  geom_boxplot(outlier.color="blue", outlier.alpha = 0.2) +
  scale_x_discrete() +
  stat_boxplot(geom ='errorbar',width=.3) +
  labs(title="Distribution of Sale Price",
       subtitle="Homes", y="Price($)",
       x="Homes") + theme_classic()
```

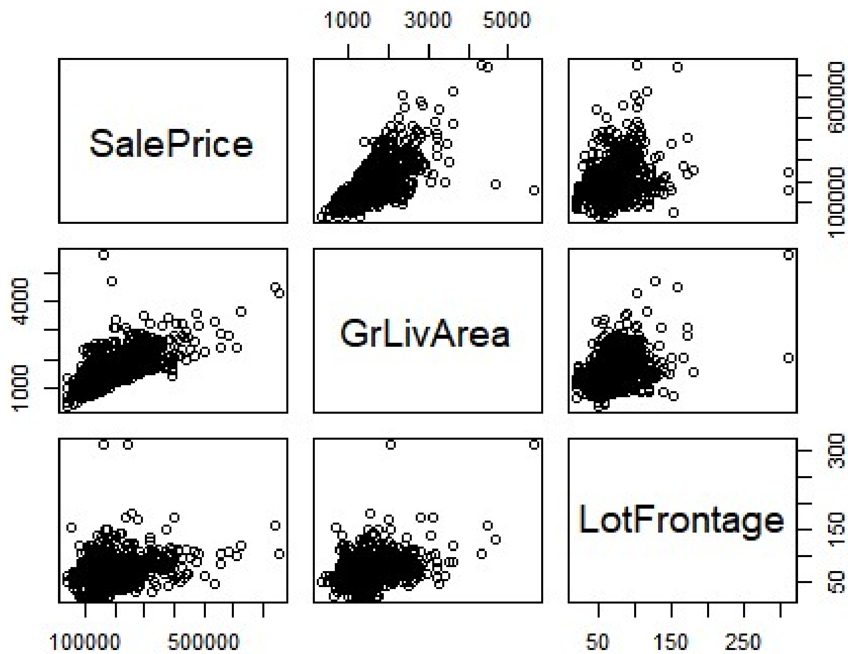


The Plot above displays that the mean price of houses below $200K and they are mostly evenly distributed with some significant outliers above $600K range.

ScatterPlot

Scatterplot matrix for "SalePrice","GrLivArea","LotFrontage"

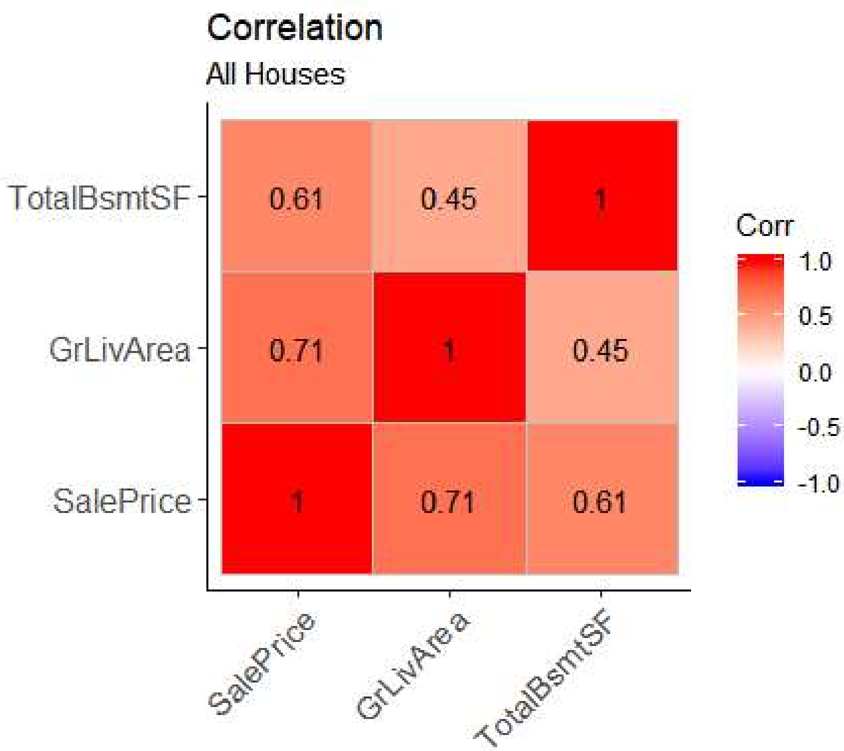
```
pairs(train[,c("SalePrice","GrLivArea","LotFrontage")])
```



From the scatter plot we can see that GrLiveArea and LotFrontage are positively correlated with Sale Price. Since Most of the sale prices are concentrated between 100k and 300k, while the lot sizes have much less spread. The larger lot sizes do not necessarily belong to the most expensive properties, which is why we do not see a stronger correlation.

Correlation matrix

```
cormat <- cor(train[,c("SalePrice","GrLivArea","TotalBsmtSF")])
cormat

##             SalePrice GrLivArea TotalBsmtSF
## SalePrice   1.0000000 0.7086245   0.6135806
## GrLivArea   0.7086245 1.0000000   0.4548682
## TotalBsmtSF 0.6135806 0.4548682   1.0000000

# Subset of variables
train_cor <- train  %>% dplyr::select(SalePrice, GrLivArea, TotalBsmtSF)

# Compute correlations
corr <- cor(train_cor)
ggcorrplot(corr,lab=TRUE, ggtheme = ggplot2::theme_classic) +
  labs(title="Correlation",subtitle="All Houses")
```



The graph above displays that Sale Price shows strong positive correlation with "GrLivArea"" and moderate correlation with TotalBsmTSF. In Addition,"GrLivArea"" shows Strong positive correlation with SalePrice and weak positive correlation with "TotalBsmSF" and also "TotalBsmSF"" shows moderate positive correlation with SalePrice and weak positive correlation with "GrLivArea".

Hypothesis and 80% confidence interval

Test the hypotheses that the correlations between each pairwise set of variables is 0 and provide an 80% confidence interval.Discuss the meaning of your analysis. Would you be worried about familywise error? Why or why not?

Null (Ho) Hypothesis: The correlation between GrLivArea and SalePrice is 0
Alternative(H1) Hypothesis: The correlation between GrLivArea and SalePrice is other than 0

```
cor.test(train$GrLivArea, train$TotalBsmtSF, conf.level = 0.8)

##
##  Pearson's product-moment correlation
##
## data:  train$GrLivArea and train$TotalBsmtSF
## t = 19.503, df = 1458, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.4278380 0.4810855
## sample estimates:
```

```
##       cor
## 0.4548682
```

Since the the p value of the test is less than 0.05 at 5% level of significance we reject the null hypothesis and conclude that the correlation between **GrLivArea** and **TotalBsmtSF** is other than 0.

80 percent confidence interval of the test is 0.4327076 0.4879552

```
cor.test(train$SalePrice, train$TotalBsmtSF, conf.level = 0.8)

##
##  Pearson's product-moment correlation
##
## data:  train$SalePrice and train$TotalBsmtSF
## t = 29.671, df = 1458, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.5922142 0.6340846
## sample estimates:
##       cor
## 0.6135806
```

Since the the p value of the test is less than 0.05 at 5% level of significance we reject the null hypothesis and conclude that the correlation between **SalePrice** and **TotalBsmtSF** is other than 0.

80 percent confidence interval of the test is 0.5922142 0.6340846

```
cor.test(train$SalePrice, train$GrLivArea, conf.level = 0.8)

##
##  Pearson's product-moment correlation
##
## data:  train$SalePrice and train$GrLivArea
## t = 38.348, df = 1458, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.6915087 0.7249450
## sample estimates:
##       cor
## 0.7086245
```

Since the the p value of the test is less than 0.05 at 5% level of significance we reject the null hypothesis and conclude that the correlation between **SalePrice** and **GrLivArea** is other than 0.

80 percent confidence interval of the test is 0.6915087 0.7249450

Familywise Error

type I error is the rejection of a true null hypothesis (also known as a “false positive” finding or conclusion)

```
FWE <- 1 - (1 - .05)^2
FWE

## [1] 0.0975
```

There is a 9.75% chance of type 1 error. Since the chance is low I will not be worried for family wise error .

5 points.

Linear Algebra and Correlation.

Invert your correlation matrix from above. (This is known as the precision matrix and contains variance inflation factors on the diagonal.) Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix. Conduct LU decomposition on the matrix.

Invert your correlation matrix.This is known as the precision matrix and contains variance inflation factors on the diagonal.

```
# find inverse
precision_mat <- solve(cormat)
```

Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix.

```
# Multiply the correlation matrix by the precision matrix
cor_prec <- cormat %*% precision_mat
cor_prec

##                             SalePrice                  GrLivArea
## SalePrice   1.00000000000000022204460 -0.00000000000000002081668
## GrLivArea   0.00000000000000005551115  1.00000000000000000000000
## TotalBsmtSF 0.00000000000000000000000  0.00000000000000005551115
##                          TotalBsmtSF
## SalePrice   0.0000000000000000000000
## GrLivArea   0.0000000000000001110223
## TotalBsmtSF 1.0000000000000000000000

#  multiply the precision matrix by the correlation matrix
prec_cor <-   precision_mat %*% cormat
prec_cor

##                            SalePrice                 GrLivArea
## SalePrice   0.9999999999999997779554 -0.0000000000000001665335
## GrLivArea   0.0000000000000002012279  1.0000000000000004440892
```

```
## TotalBsmtSF 0.0000000000000000000000  0.0000000000000001110223
##                           TotalBsmtSF
## SalePrice   -0.0000000000000001110223
## GrLivArea    0.0000000000000001665335
## TotalBsmtSF  1.0000000000000000000000
```

```r
# LU Decomposistion
library(pracma)
```

```
## Warning: package 'pracma' was built under R version 3.5.3
```

```
##
## Attaching package: 'pracma'
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```r
lu(cormat)
```

```
## $L
##             SalePrice  GrLivArea TotalBsmtSF
## SalePrice   1.0000000 0.00000000           0
## GrLivArea   0.7086245 1.00000000           0
## TotalBsmtSF 0.6135806 0.04031325           1
##
## $U
##             SalePrice GrLivArea TotalBsmtSF
## SalePrice           1 0.7086245   0.6135806
## GrLivArea           0 0.4978513   0.0200700
## TotalBsmtSF         0 0.0000000   0.6227098
```

Calculus-Based Probability & Statistics.

Many times, it makes sense to fit a closed form distribution to data. Select a variable in the Kaggle.com training dataset that is skewed to the right, shift it so that the minimum value is absolutely above zero if necessary. Then load the MASS package and run fitdistr to fit an exponential probability density function. (See https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html). Find the optimal value of ??? for this distribution, and then take 1000 samples from this exponential distribution using this value (e.g., rexp(1000, ???)). Plot a histogram and compare it with a histogram of your original variable. Using the exponential pdf, find the 5th and 95th percentiles using the cumulative distribution function (CDF). Also generate a 95% confidence interval from the empirical data, assuming normality. Finally, provide the empirical 5th percentile and 95th percentile of the data. Discuss.

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.5.3
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

Univariate distribution of LotArea

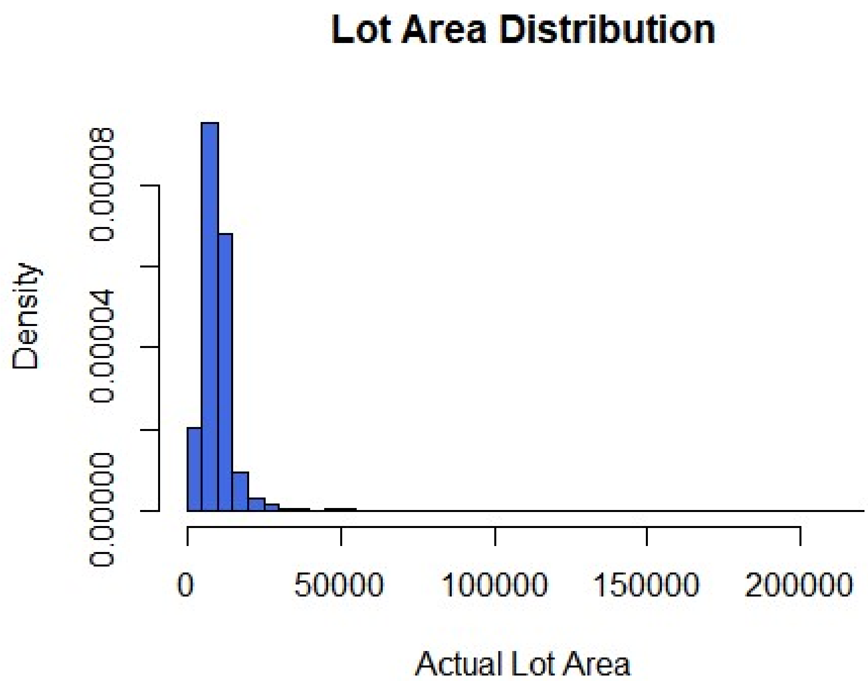
```
(expdf <- fitdistr(train$LotArea, "exponential"))

##          rate
##   0.000095085704
##  (0.000002488507)

# get value of lambda from exponential distribution
lambda <- expdf$estimate

# expected value of lambda
rate <- 1 / lambda
rate

##     rate
## 10516.83
```

Then, take 1000 samples from this exponential distribution using this value. (e.g., rexp(1000, some_val))((()))

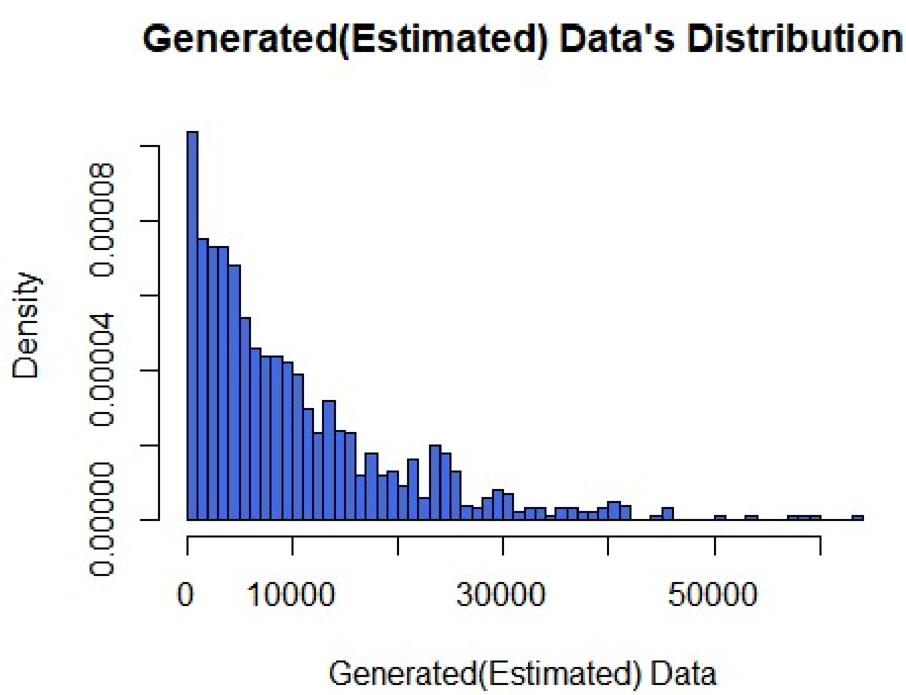
```
# 1000 samples from exponential distribution using lambda
expdf_samp <- rexp(1000, lambda)
```

Plot a histogram and compare it with a histogram of your original variable.

```
# Actual vs simulated distribution
hist(train$LotArea, breaks=50, prob=TRUE,col="royalblue", xlab="Actual Lot
Area",
     main="Lot Area Distribution")
```




```
hist(expdf_samp, breaks=50, prob=TRUE,col="royalblue",
xlab="Generated(Estimated) Data",
     main="Generated(Estimated) Data's Distribution")
```



As we can see plots here that our Lot Area approximately fits a exponential distribution. The fit does not do good job here.Let's look at the summary table to understand the details

```
# Actuals Data summary Table
summary(expdf_samp)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     2.28  2977.46  7154.88 10147.36 14053.89 63968.07

# Generated Data summary Table
summary(train$LotArea)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1300    7554    9478   10517   11602  215245
```

CDF

5th and 95th percentiles using the cumulative distribution function (CDF)

```
# 5 and 95 percentile of exponential pdf
qexp(c(.05, .95), rate = lambda)

## [1]   539.4428 31505.6013
```

Also generate a 95% confidence interval from the empirical data, assuming normality

```
# 95% confidence interval for sample mean (assuming normality)
func <- qnorm(0.95)
a <- func * sd(train$LotArea)/sqrt(length(train$LotArea))
paste("CI for Population Mean: ",round(mean(train$LotArea - a),2)," - ",
      round(mean(train$LotArea + a),2),sep='')

## [1] "CI for Population Mean: 10087.16 - 10946.5"
```

Modeling

In Model Data engineering part,I initiall start to find the variables with very large number of missing values.Below table show missing values in traindata

```
#Missing values table
sapply(train, function(x){sum(is.na(x))})

##            Id    MSSubClass      MSZoning   LotFrontage       LotArea
##             0             0             0           259             0
##        Street         Alley      LotShape   LandContour     Utilities
##             0          1369             0             0             0
##     LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##             0             0             0             0             0
##      BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##             0             0             0             0             0
##  YearRemodAdd     RoofStyle      RoofMatl   Exterior1st   Exterior2nd
##             0             0             0             0             0
```

```
##     MasVnrType     MasVnrArea      ExterQual      ExterCond     Foundation 
##              8              8              0              0              0 
##       BsmtQual       BsmtCond   BsmtExposure   BsmtFinType1     BsmtFinSF1 
##             37             37             38             37              0 
##   BsmtFinType2     BsmtFinSF2      BsmtUnfSF    TotalBsmtSF        Heating 
##             38              0              0              0              0 
##      HeatingQC     CentralAir     Electrical      X1stFlrSF      X2ndFlrSF 
##              0              0              1              0              0 
##   LowQualFinSF      GrLivArea   BsmtFullBath   BsmtHalfBath       FullBath 
##              0              0              0              0              0 
##       HalfBath   BedroomAbvGr   KitchenAbvGr    KitchenQual   TotRmsAbvGrd 
##              0              0              0              0              0 
##     Functional     Fireplaces    FireplaceQu     GarageType    GarageYrBlt 
##              0              0            690             81             81 
##   GarageFinish     GarageCars     GarageArea     GarageQual     GarageCond 
##             81              0              0             81             81 
##     PavedDrive     WoodDeckSF    OpenPorchSF  EnclosedPorch     X3SsnPorch 
##              0              0              0              0              0 
##    ScreenPorch       PoolArea         PoolQC          Fence    MiscFeature 
##              0              0           1453           1179           1406 
##        MiscVal         MoSold         YrSold       SaleType  SaleCondition 
##              0              0              0              0              0 
##      SalePrice 
##              0
```

By looking at the table, I will remove the columns that have large missings from train and test data sets

```
train <-train[, !colnames(train) %in%
c("Id","Alley","PoolQC","Fence","MiscFeature","FireplaceQu","LotFrontage","Ye
arBuilt","YearRemodAdd")]

test <- test[, !colnames(test) %in%
c("Alley","PoolQC","Fence","MiscFeature","FireplaceQu","LotFrontage","YearBui
lt","YearRemodAdd")]
```

The next step is Encoding “converting categoricals to numerics”

```
# Encoding

train <- train%>%
  mutate_if(is.character, as.factor)%>%
  mutate_if(is.factor, as.integer)

test <- test %>%
   mutate_if(is.character, as.factor)%>%
  mutate_if(is.factor, as.integer)

# omit the missing values in train data and test
train <- na.omit(train)
```

```r
# Replace numeric NAs with 0
test <- test %>% mutate_if(is.numeric, ~replace(., is.na(.), 0))
```

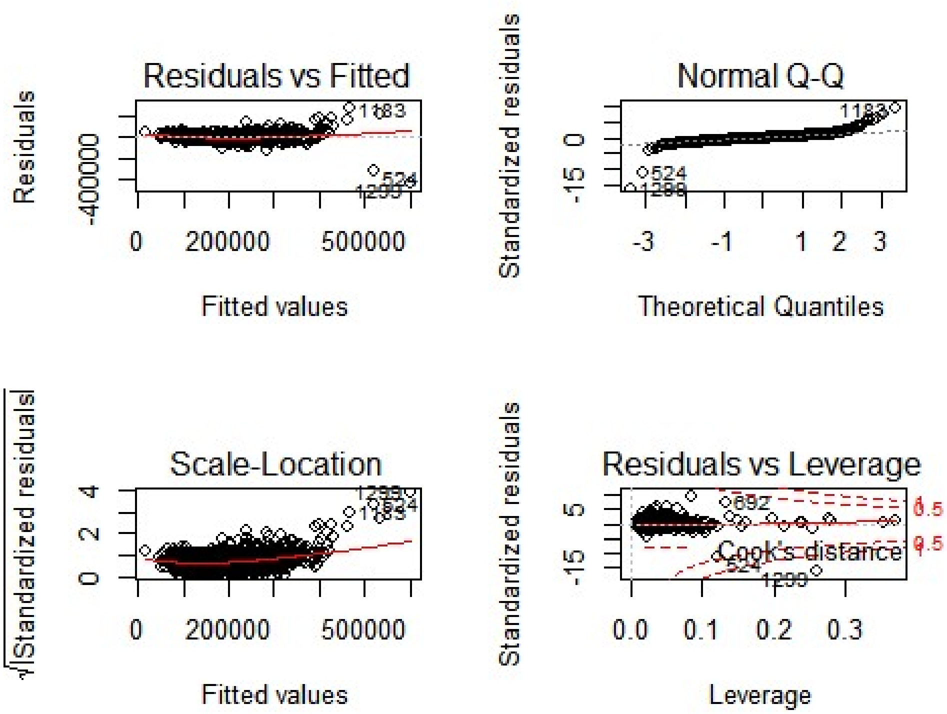
I'll now do a stepwise regression based on ACI criterion

```r
model_fit <- lm(SalePrice~., data = train)
step_model <- step(model_fit, trace = 0)
summary(step_model)

##
## Call:
## lm(formula = SalePrice ~ MSSubClass + MSZoning + LotArea + Street +
##     LotShape + LandContour + LandSlope + Condition2 + HouseStyle +
##     OverallQual + OverallCond + RoofStyle + RoofMatl + Exterior1st +
##     MasVnrType + MasVnrArea + ExterQual + Foundation + BsmtQual +
##     BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + X1stFlrSF +
##     X2ndFlrSF + BsmtFullBath + FullBath + BedroomAbvGr + KitchenAbvGr +
##     KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + GarageCars +
##     PavedDrive + WoodDeckSF + ScreenPorch + SaleCondition, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -441172  -13550    -957   13442  278991
##
## Coefficients:
##                  Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)   -68655.8503  38693.5272  -1.774             0.076239 .
## MSSubClass      -159.8789     27.4961  -5.815    0.000000007641103 ***
## MSZoning       -2503.0986   1534.7315  -1.631             0.103139
## LotArea            0.3814      0.1060   3.597             0.000333 ***
## Street         40806.0476  15627.6939   2.611             0.009128 **
## LotShape       -1310.5266    669.9525  -1.956             0.050662 .
## LandContour     3914.9209   1436.8188   2.725             0.006522 **
## LandSlope       6115.5394   4036.2642   1.515             0.129978
## Condition2     -7336.6650   3325.0186  -2.207             0.027523 *
## HouseStyle     -1224.6140    616.7163  -1.986             0.047277 *
## OverallQual    12959.5731   1248.8115  10.378 < 0.0000000000000002 ***
## OverallCond     4247.8548    928.1763   4.577    0.000005179174355 ***
## RoofStyle       2632.4405   1158.2961   2.273             0.023208 *
## RoofMatl        4131.9524   1555.7715   2.656             0.008007 **
## Exterior1st     -614.3119    301.9824  -2.034             0.042128 *
## MasVnrType      4335.1776   1582.5650   2.739             0.006241 **
## MasVnrArea        30.2912      6.1226   4.947    0.000000850306146 ***
## ExterQual      -8542.1790   2043.1130  -4.181    0.000030972233573 ***
## Foundation      3189.2880   1670.1013   1.910             0.056400 .
## BsmtQual       -8946.3960   1491.0981  -6.000    0.000000002557006 ***
## BsmtCond        3202.5638   1404.1962   2.281             0.022727 *
## BsmtExposure   -3678.7800    901.4594  -4.081    0.000047592852072 ***
## BsmtFinType1   -1168.7326    649.9609  -1.798             0.072384 .
## BsmtFinSF1         5.8341      3.1549   1.849             0.064652 .
## X1stFlrSF         45.5519      4.8433   9.405 < 0.0000000000000002 ***
```

```
## X2ndFlrSF          45.2873       4.1751  10.847 < 0.0000000000000002 ***
## BsmtFullBath     7523.9163    2355.1435   3.195             0.001434 **
## FullBath         4197.5166    2518.4225   1.667             0.095810 .
## BedroomAbvGr    -4439.2566    1771.3141  -2.506             0.012325 *
## KitchenAbvGr   -21663.3923    6032.6496  -3.591             0.000342 ***
## KitchenQual     -8746.4950    1523.4941  -5.741    0.000000011699055 ***
## TotRmsAbvGrd     3456.5692    1200.5186   2.879             0.004052 **
## Functional       3843.4812    1016.0138   3.783             0.000162 ***
## Fireplaces       4035.7108    1688.6046   2.390             0.016992 *
## GarageCars      14425.7154    1972.8288   7.312    0.000000000000458 ***
## PavedDrive       4949.3592    2348.8421   2.107             0.035296 *
## WoodDeckSF         18.4401       7.5978   2.427             0.015359 *
## ScreenPorch        41.4813      15.9000   2.609             0.009188 **
## SaleCondition    2596.0989     873.9699   2.970             0.003028 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32290 on 1299 degrees of freedom
## Multiple R-squared:  0.8373, Adjusted R-squared:  0.8326
## F-statistic:   176 on 38 and 1299 DF,  p-value: < 0.00000000000000022
```

Residual Analysis

```r
par(mfrow=c(2,2))
plot(step_model)
```



The residuals are approximately normally distributed. There is not heteroscedacity and pattern in the residuals.

```r
forecast <- predict(step_model, test)
results <- data.frame(Id = test$Id, SalePrice=forecast)
```

Export submission

```r
#Write to .csv for submission to Kaggle
write.csv(results, file = "submission_omerozeren.csv", row.names = FALSE)
```

Kaggle Submission

My Kaggle user name is **omerozeren**, and the resulting score on Kaggle.com from this model is **0.21620**.