Final

David Moste

12/14/2020

Problem 1

I started by creating all four sets: X, Y, x, and y. I then used R to calculate all the listed probabilities.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

```r
set.seed(12345)
N <- 10
mu <- (N+1)/2
sigma <- (N+1)/2
X <- runif(10000,1,N)
Y <- rnorm(10000,mean = mu,sd = sigma)
df <- data.frame(X,Y)

x <- median(X)
y <- summary(Y)[2]

df_Xy <- df %>% filter(X > y)
df_Yy <- df %>% filter(Y > y)

# Part a
df_a <- df_Xy %>% filter(X > x)
prob_a <- nrow(df_a)/nrow(df_Xy)
prob_a
```

```
## [1] 0.5512679
```

```r
# Part b
df_b <- df_Yy %>% filter(X > x)
prob_b <- nrow(df_b)/nrow(df)
prob_b
```

```
## [1] 0.3808
```

```r
# Part c
df_c <- df_Xy %>% filter(X < x)
prob_c <- nrow(df_c)/nrow(df_Xy)
prob_c
```

```
## [1] 0.4487321
```

$P_A = 0.55$
$P_B = 0.38$
$P_C = 0.45$

The next step was to create the marginal and joint probability table.

```r
# Probability table
XY <- df %>% filter(X > x, Y > y) %>% nrow()
Xy <- df %>% filter(X > x, Y < y) %>% nrow()
Xx_total <- XY/nrow(df) + Xy/nrow(df)

xY <- df %>% filter(X < x, Y > y) %>% nrow()
xy <- df %>% filter(X < x, Y < y) %>% nrow()
xX_total <- xY/nrow(df) + xy/nrow(df)

Yy_total <- XY/nrow(df) + xY/nrow(df)
yY_total <- Xy/nrow(df) + xy/nrow(df)
total <- Xx_total + xX_total

table <- data.frame(rbind(XY/nrow(df),Xy/nrow(df),XY/nrow(df) + Xy/nrow(df)),
                    rbind(xY/nrow(df),xy/nrow(df),xY/nrow(df) + xy/nrow(df)),
                    rbind(Yy_total,yY_total,total))
names(table) <- c("X > x","X < x","Total")
row.names(table) <- c("Y > y","Y < y","Total")
table
```

```
##        X > x  X < x Total
## Y > y 0.3808 0.3692  0.75
## Y < y 0.1192 0.1308  0.25
## Total 0.5000 0.5000  1.00
```

As can be seen, it is clear that the two probabilities are equal.
$P(X > x, Y > y) = P(X > x)\,P(Y > y)$

The nextr step is to check for independence. Fisher's Test works for small datasets while Chi Squared is more appropriate for larger datasets. Therefore I would choose Chi Squared for this problem.

```r
# Check independence
count_table <- matrix(c(XY,Xy,xY,xy),nrow = 2, ncol = 2, byrow = FALSE)
fisher.test(count_table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  count_table
## p-value = 0.007904
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.032680 1.240419
## sample estimates:
## odds ratio
##   1.131777
```

```r
chisq.test(count_table)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  count_table
## X-squared = 7.0533, df = 1, p-value = 0.007912
```

Problem 2

Descriptive and Inferential Statistics

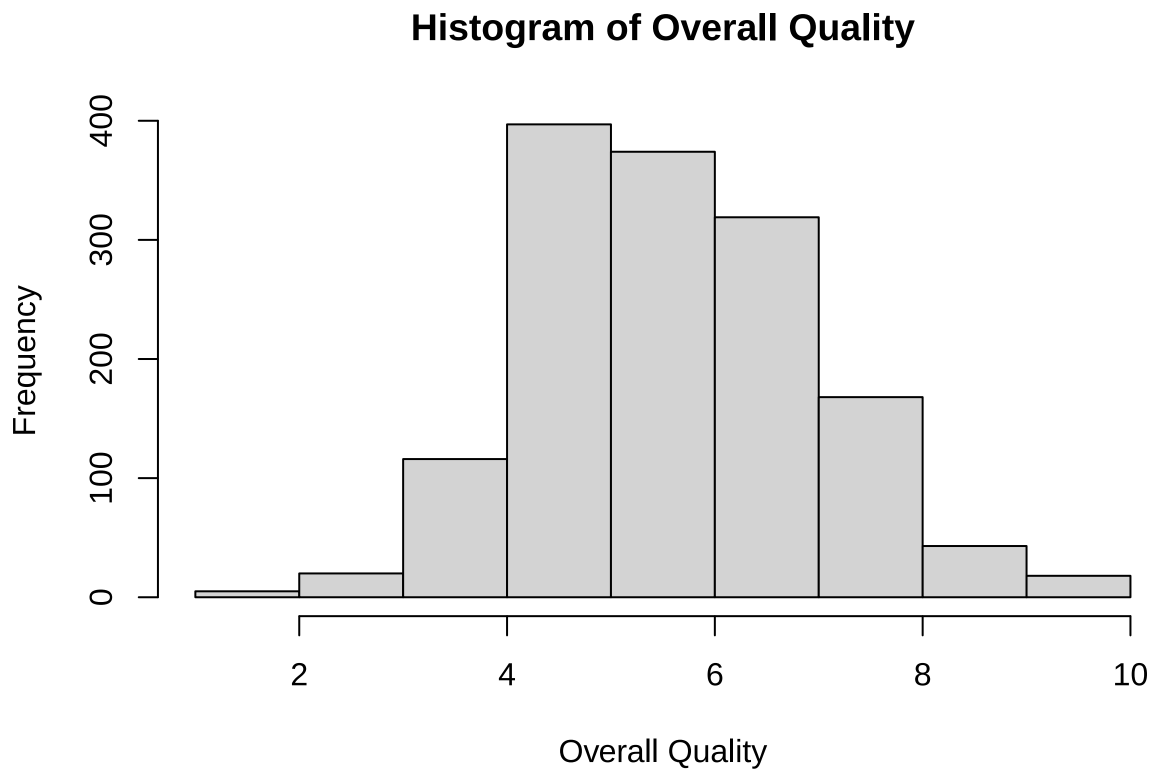
```r
train <- read.csv('https://raw.githubusercontent.com/dmoste/DATA605/master/train.csv')
test <- read.csv('https://raw.githubusercontent.com/dmoste/DATA605/master/test.csv')

train[is.na(train)] = 0

# Summary and histogram of Overall Quality
summary(train$OverallQual)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   6.000   6.099   7.000  10.000
```

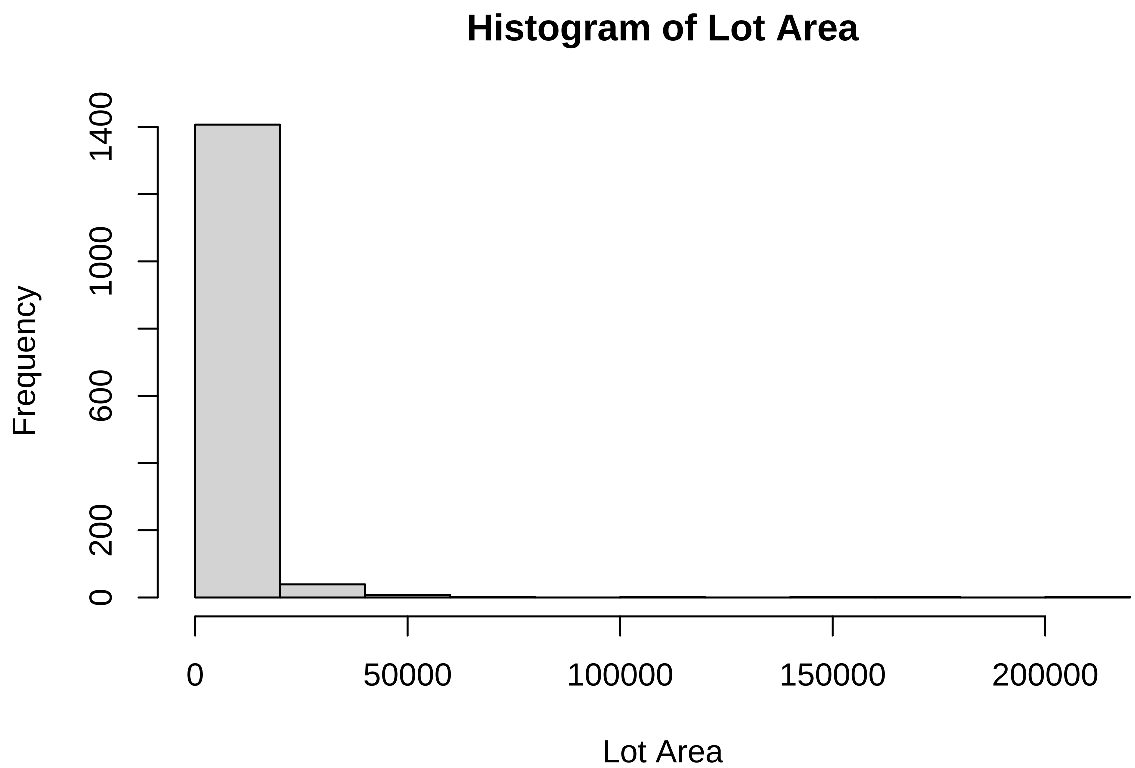
```r
hist(train$OverallQual,
     xlab = "Overall Quality",
     main = "Histogram of Overall Quality")
```




```r
# Summary and histogram of Lot Area
summary(train$LotArea)
```

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

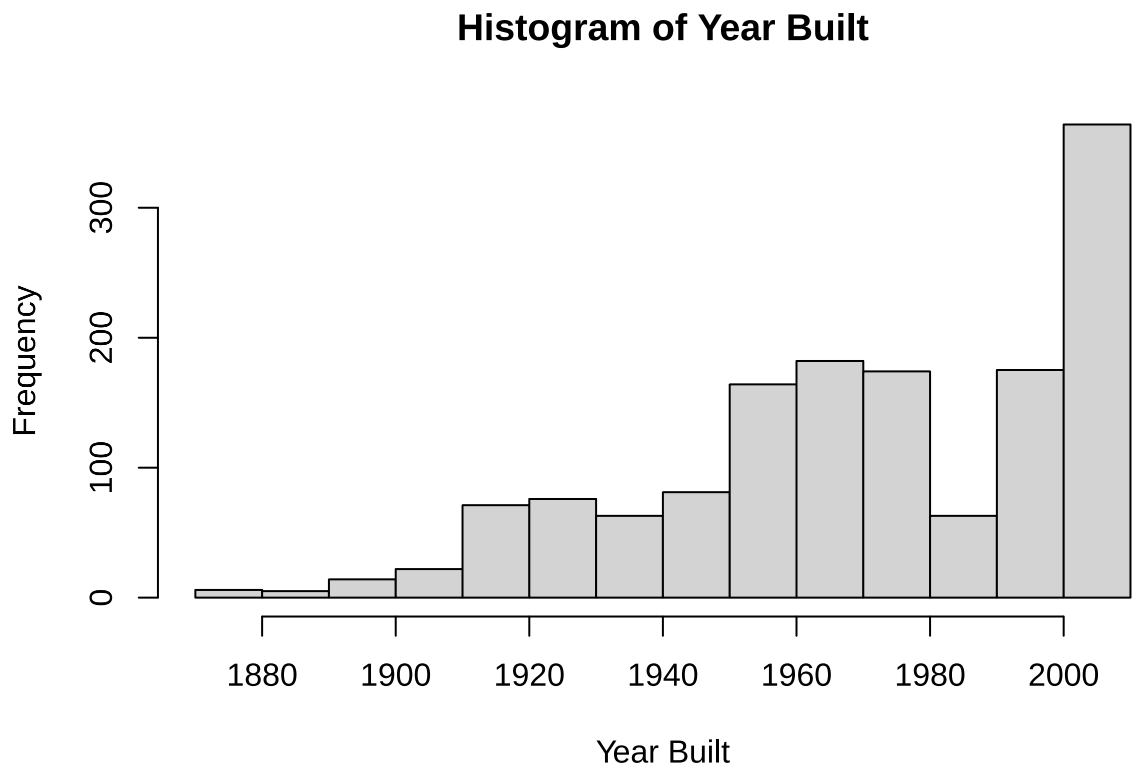
```r
hist(train$LotArea,
     xlab = "Lot Area",
     main = "Histogram of Lot Area")
```



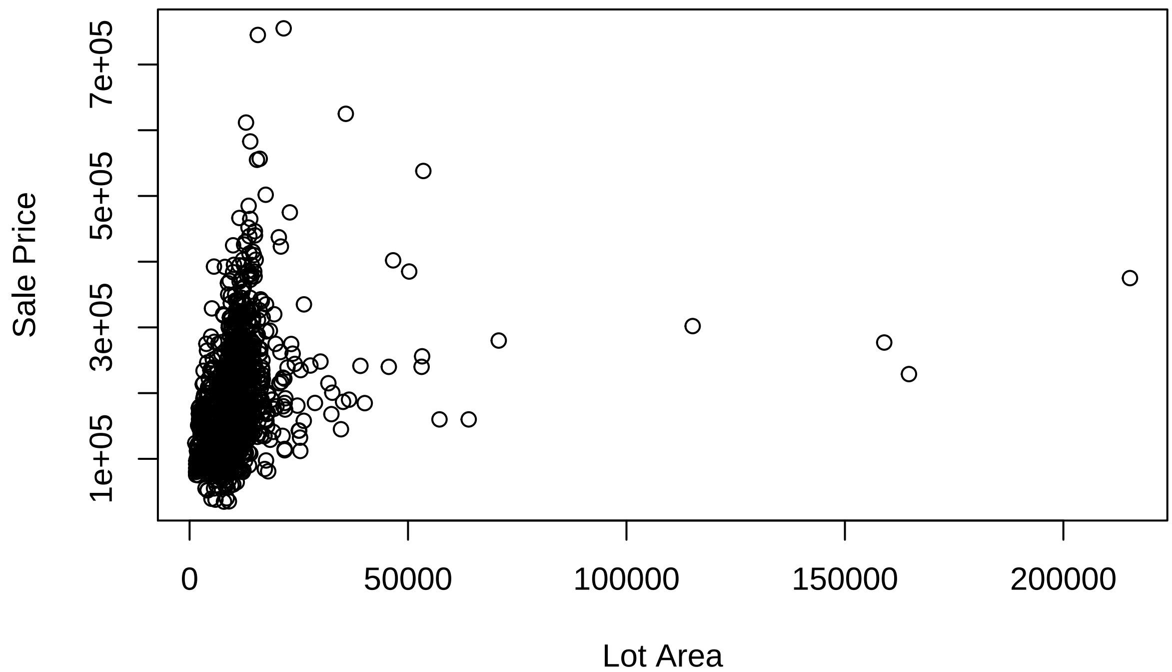

```r
# Summary and histogram of Year Built
summary(train$YearBuilt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1872    1954    1973    1971    2000    2010
```

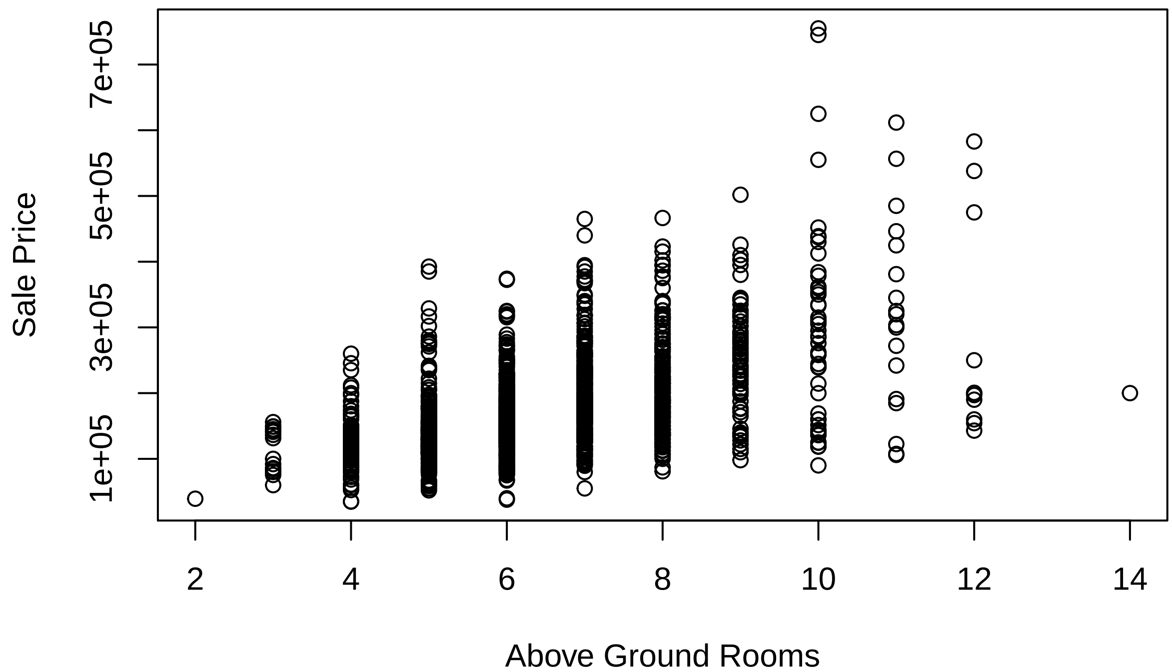
```r
hist(train$YearBuilt,
     xlab = "Year Built",
     main = "Histogram of Year Built")
```




```r
# Plot Lot Area vs Sale Price
plot(train$LotArea,
     train$SalePrice,
     xlab = "Lot Area",
     ylab = "Sale Price")
```




```r
# Plot Total above ground rooms vs Sale Price
plot(train$TotRmsAbvGrd,
     train$SalePrice,
     xlab = "Above Ground Rooms",
     ylab = "Sale Price")
```




```
# Remove outliers
train <- train %>% filter(LotArea < 100000)

# Create a correlation matrix for Lot Area, Year Built, and Sale Price
correlation <- cor(train %>% dplyr::select(LotArea, YearBuilt, SalePrice) %>% as.matrix())
correlation
```

```
##              LotArea  YearBuilt SalePrice
## LotArea   1.00000000 0.04230918 0.3544944
## YearBuilt 0.04230918 1.00000000 0.5255868
## SalePrice 0.35449443 0.52558678 1.0000000
```

```
# Run a correlation test with 80% confidence interval
cor.test(train$LotArea, train$SalePrice, conf.level = 0.8)
```

```
##
##  Pearson's product-moment correlation
##
## data:  train$LotArea and train$SalePrice
## t = 14.456, df = 1454, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.3247557 0.3835329
## sample estimates:
##       cor
## 0.3544944
```

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

```
##
##  Pearson's product-moment correlation
##
## data:  train$YearBuilt and train$SalePrice
## t = 23.558, df = 1454, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.5008255 0.5494885
## sample estimates:
##       cor
## 0.5255868
```

The correlation between Lot Area and Sale Price is 0.35, which indicates they are weakly correlated at best. The correlation between Year Built and Sale Price is 0.52, which is another weak correlation. The p-value is less than 0.05 though for both, which suggests the correlation is real.

Linear Algebra and Correlation

```r
# Invert the correlation matrix
precision = solve(correlation)

# Multiply the correlation and precision matrices
round(correlation %*% precision)
```

```
##           LotArea YearBuilt SalePrice
## LotArea         1         0         0
## YearBuilt       0         1         0
## SalePrice       0         0         1
```

```r
round(precision %*% correlation)
```

```
##           LotArea YearBuilt SalePrice
## LotArea         1         0         0
## YearBuilt       0         1         0
## SalePrice       0         0         1
```

```r
# Conduct LU decomposition
library(matrixcalc)
```

```
## Warning: package 'matrixcalc' was built under R version 4.0.3
```

```r
decomp <- matrixcalc::lu.decomposition(correlation)
decomp
```

```
## $L
##             [,1]      [,2] [,3]
```

```
## [1,] 1.00000000 0.000000    0
## [2,] 0.04230918 1.000000    0
## [3,] 0.35449443 0.511504    1
##
## $U
##      [,1]       [,2]      [,3]
## [1,]    1 0.04230918 0.3544944
## [2,]    0 0.99820993 0.5105884
## [3,]    0 0.00000000 0.6131657
```

Calculus-Based Probability & Statistics

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.3
```

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

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

```r
# Fit Exponential Probability Density
ep = MASS::fitdistr(train$LotArea, "exponential")
ep
```

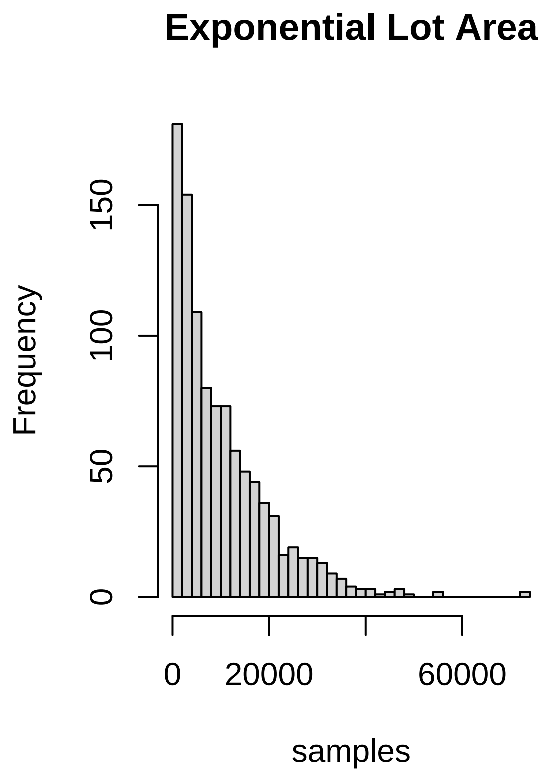
```
##        rate
##   9.904415e-05
##  (2.595662e-06)
```

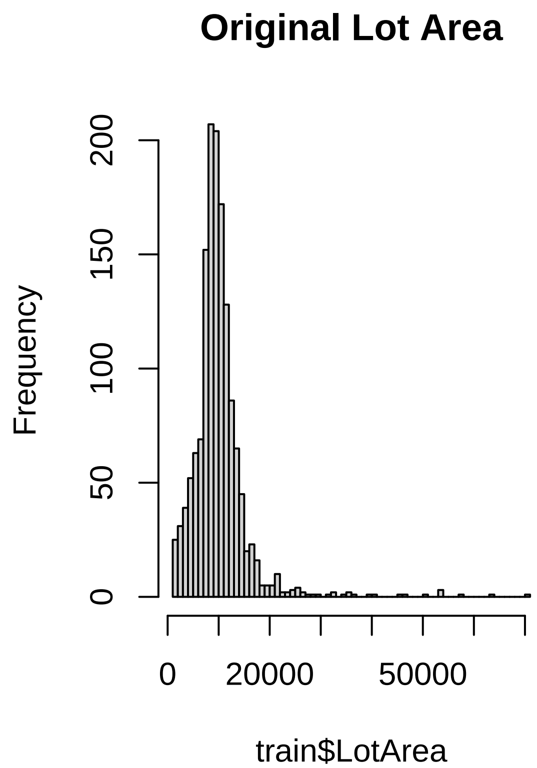
```r
# Find the optimal lambda
lambda = ep$estimate
lambda
```

```
##         rate
## 9.904415e-05
```

```r
# Get 1000 values
samples = rexp(1000, lambda)

# Plot histogram of original and exponential
par(mfrow = c(1, 2))
hist(samples, breaks = 50, main = "Exponential Lot Area")
hist(train$LotArea, breaks = 50, main = "Original Lot Area")
```






```r
# Percentiles based on exponential
qexp(0.05, rate = lambda)
```

```
## [1] 517.8831
```

```r
qexp(0.95, rate = lambda)
```

```
## [1] 30246.43
```

```r
# Empirical 95% confidence interval assuming normailty
me <- qnorm(0.975)*nrow(train)/sqrt(sd(train$LotArea))
upper <- mean(train$LotArea) + me
lower <- mean(train$LotArea) - me

# Empirical percentiles
quantile(train$LotArea, 0.05)
```

```
##     5%
## 3294.5
```

```r
quantile(train$LotArea, 0.95)
```

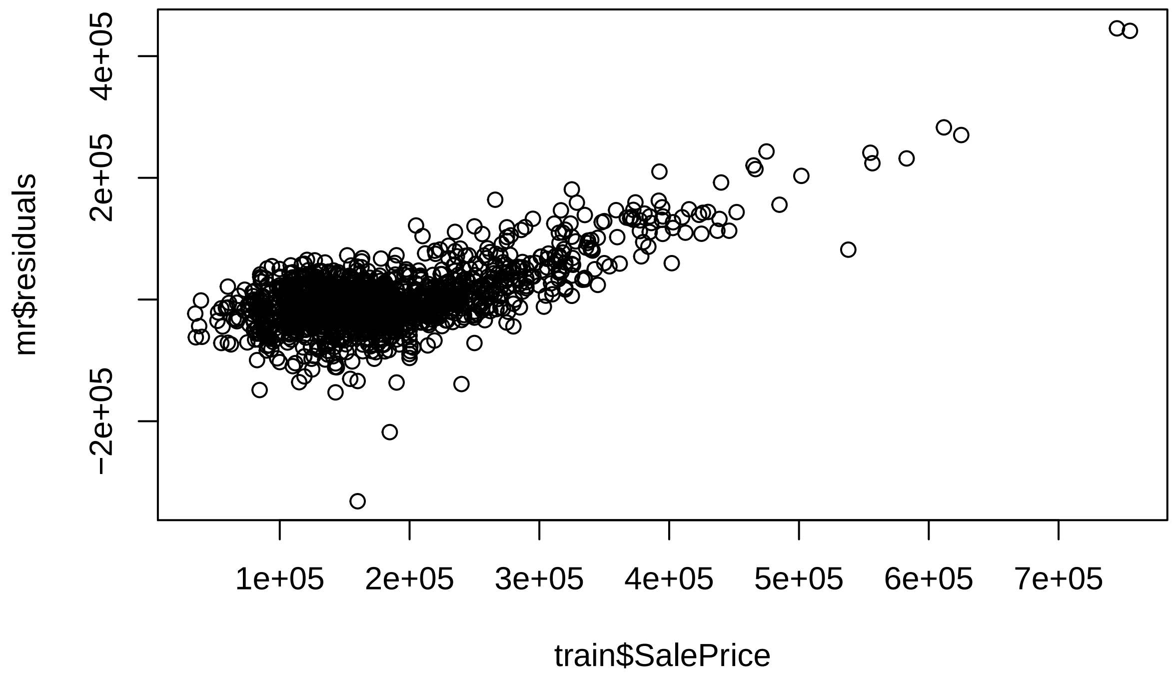
```
##   95%
## 17108
```

Modeling

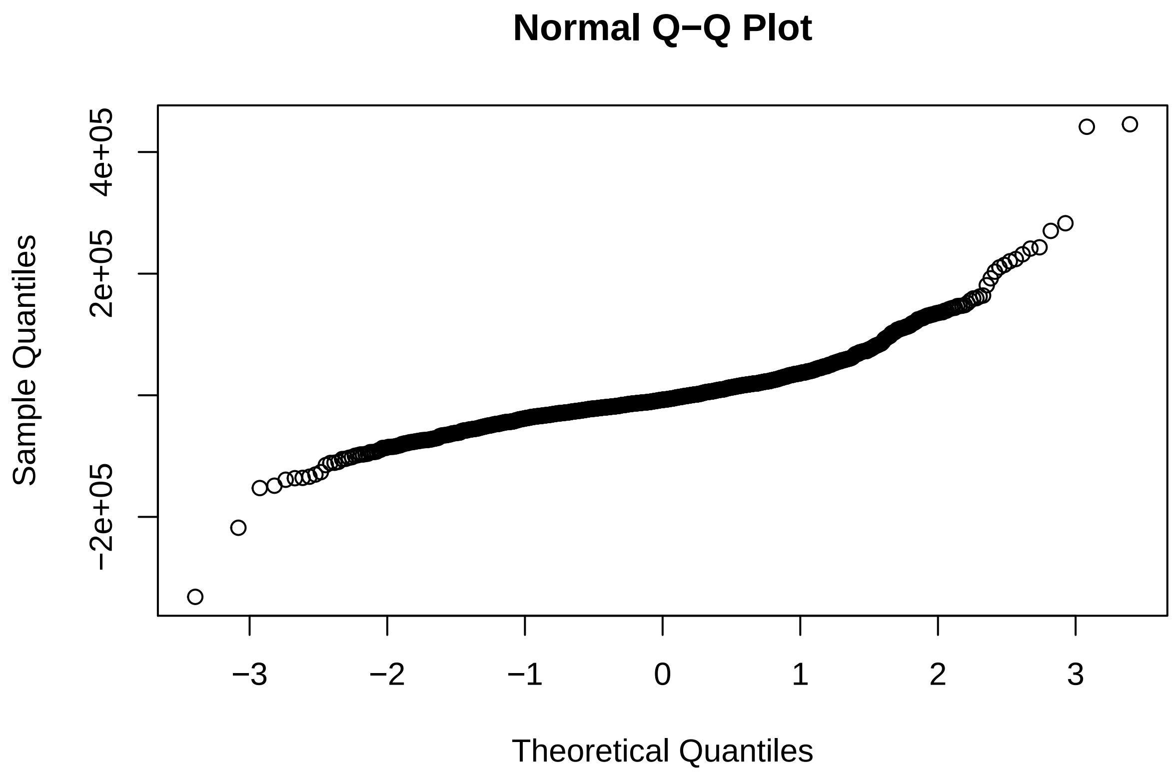
```r
# Create multiple regression
mr <- lm(SalePrice ~ LotArea + YearBuilt + TotRmsAbvGrd, data = train)
summary(mr)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + YearBuilt + TotRmsAbvGrd,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -331486  -27636   -7406   19341  445648
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.444e+06  9.110e+04  -26.83   <2e-16 ***
## LotArea       2.809e+00  2.605e-01   10.79   <2e-16 ***
## YearBuilt     1.248e+03  4.640e+01   26.91   <2e-16 ***
## TotRmsAbvGrd  2.078e+04  9.080e+02   22.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53270 on 1452 degrees of freedom
## Multiple R-squared:  0.5494, Adjusted R-squared:  0.5484
## F-statistic:   590 on 3 and 1452 DF,  p-value: < 2.2e-16
```

```r
plot(mr$residuals ~ train$SalePrice)
```




```
qqnorm(mr$residuals)
```




```
submission <- (2.809*test$LotArea)+(1248*test$YearBuilt)+(20780*test$TotRmsAbvGrd)-2444000

#submission <- cbind(test,submission) %>% select(Id, submission)
#names(submission) <- c("Id","SalePrice")
#write.csv(submission, file = "DMosteSubmission.csv")
```

My username is davidmoste and I hade a score of 0.285.