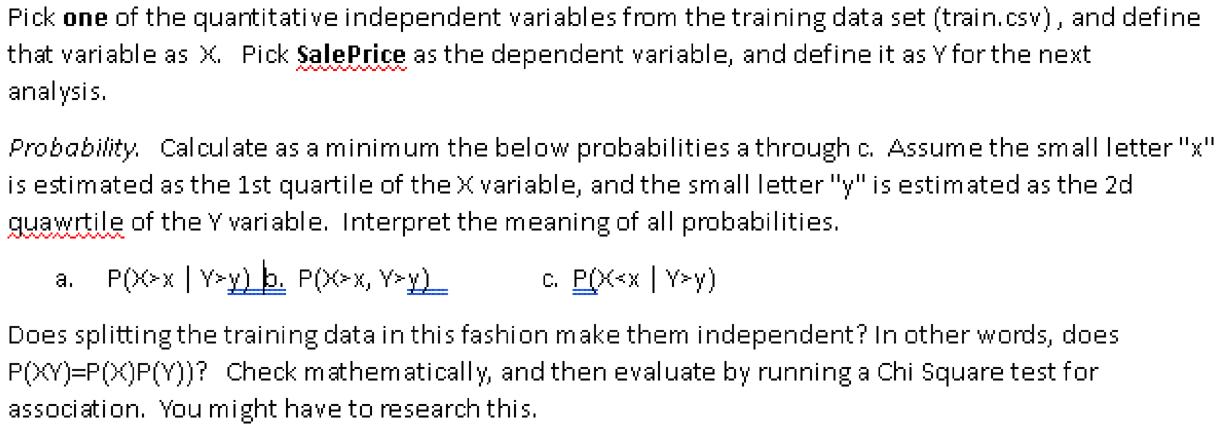
DATA 605 - Final

Michael Muller

December 18, 2017

```
df = read.csv('train.csv')
```

————————> Section 1



Pick **one** of the quantitative independent variables from the training data set (train.csv), and define that variable as X. Pick **SalePrice** as the dependent variable, and define it as Y for the next analysis.

Probability. Calculate as a minimum the below probabilities a through c. Assume the small letter "x" is estimated as the 1st quartile of the X variable, and the small letter "y" is estimated as the 2d quawrtile of the Y variable. Interpret the meaning of all probabilities.

a. P(X>x | Y>y) b. P(X>x, Y>y) c. P(X<x | Y>y)

Does splitting the training data in this fashion make them independent? In other words, does P(XY)=P(X)P(Y))? Check mathematically, and then evaluate by running a Chi Square test for association. You might have to research this.

Figure 1:

I have chosen 'TotRmsAbvGrd' or Total Rooms above grade (does not include basement bedrooms)

```
rbind(c('Min','Median','Max'),c(min(df$TotRmsAbvGrd),median(df$TotRmsAbvGrd),max(df$TotRmsAbvGrd)))
```

```
##      [,1]  [,2]     [,3]
## [1,] "Min" "Median" "Max"
## [2,] "2"   "6"      "14"
```

```
X = df$TotRmsAbvGrd
x = quantile(X,.25)
Y = df$SalePrice
y = quantile(Y,.5)
rbind(c('y','x'),c(y,x))
```

```
##      50%      25%
## [1,] "y"      "x"
## [2,] "163000" "5"
```

a. P(X > x | Y > y)

```
denom = dim(df)[1]
pab =  length(X[X>x&Y>y])/denom
pb = length(Y[Y>y])/denom
answera = pab/pb
```

The probability that ‘Total Rooms above grade’ will be above 6, given that the ‘sale price’ is above 163000 is 0.9010989

Could more rooms be linked to a higher sale price!?

b. $P(X > x \ \& \ Y > y)$

```
pa = length(X[X>x])/denom
pb = length(Y[Y>y])/denom
answerb = pa*pb
```

The probability that ‘Total Rooms above grade’ will be above 6, and ‘sale price’ will be above 163000 is 0.3654344

c. $P(X < x \mid Y > y)$

```
pab =  length(X[X<x&Y>y])/denom
pb = length(Y[Y>y])/denom
answerc = pab/pb
```

The probability that ‘Total Rooms above grade’ will be below 6, given that the ‘sale price’ is above 163000 is 0.0178571

d. Does P(XY)=P(X)P(Y)? Check mathematically and using CS Test

```
#Rule of multiplication P(A and B) = P(A) P(B|A)...P(A) P(B|A) vs P(A)P(B)
pab == pb*pa
```

```
## [1] FALSE
```

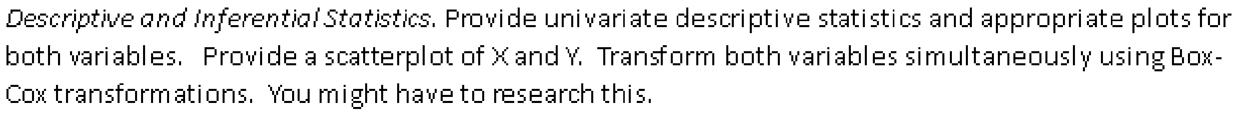
```
ct = table(df$TotRmsAbvGrd,df$SalePrice)
chisq.test(ct)
```

```
##
##  Pearson's Chi-squared test
##
## data:  ct
## X-squared = 8753, df = 7282, p-value < 2.2e-16
```

High degrees of freedom and an impossibly low p-value under 5% means that we would reject a null hypothesis, asserting that these variables are dependent.

Splitting the data in this manner does not make the variables independent.

——————> Section 2



Descriptive and Inferential Statistics. Provide univariate descriptive statistics and appropriate plots for both variables. Provide a scatterplot of X and Y. Transform both variables simultaneously using Box-Cox transformations. You might have to research this.

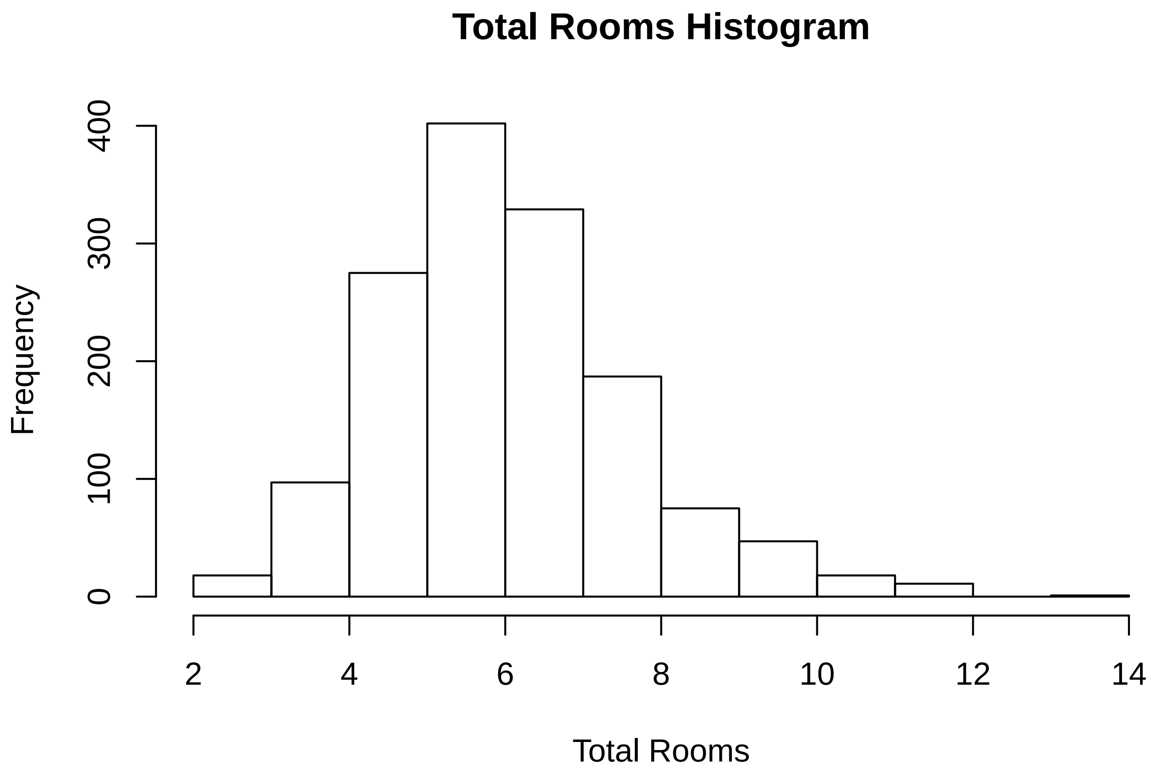
Figure 2:

Univariate statistics + plots

```
describe(X) # Total Rooms Above Grade
```

```
##    vars    n mean   sd median trimmed  mad min max range skew kurtosis
## X1    1 1460 6.52 1.63      6    6.41 1.48   2  14    12 0.67     0.87
##      se
## X1 0.04
```

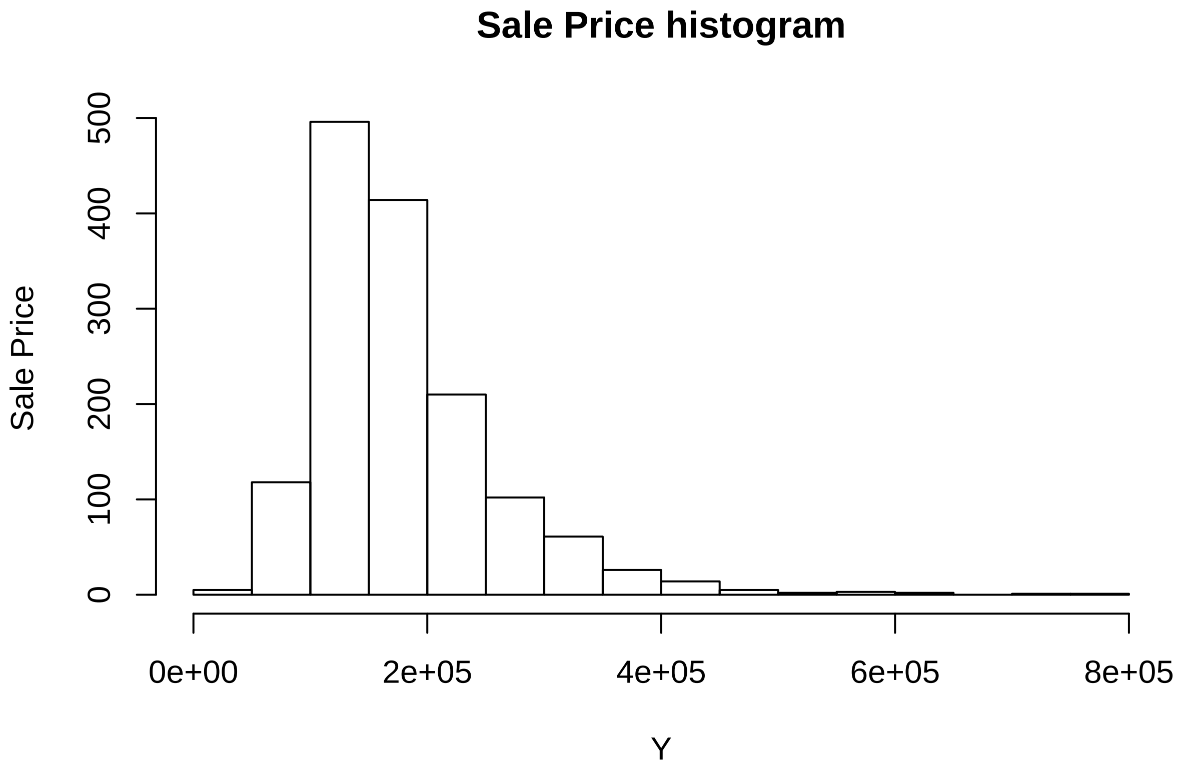
```
hist(X,main='Total Rooms Histogram',xlab='Total Rooms')
```



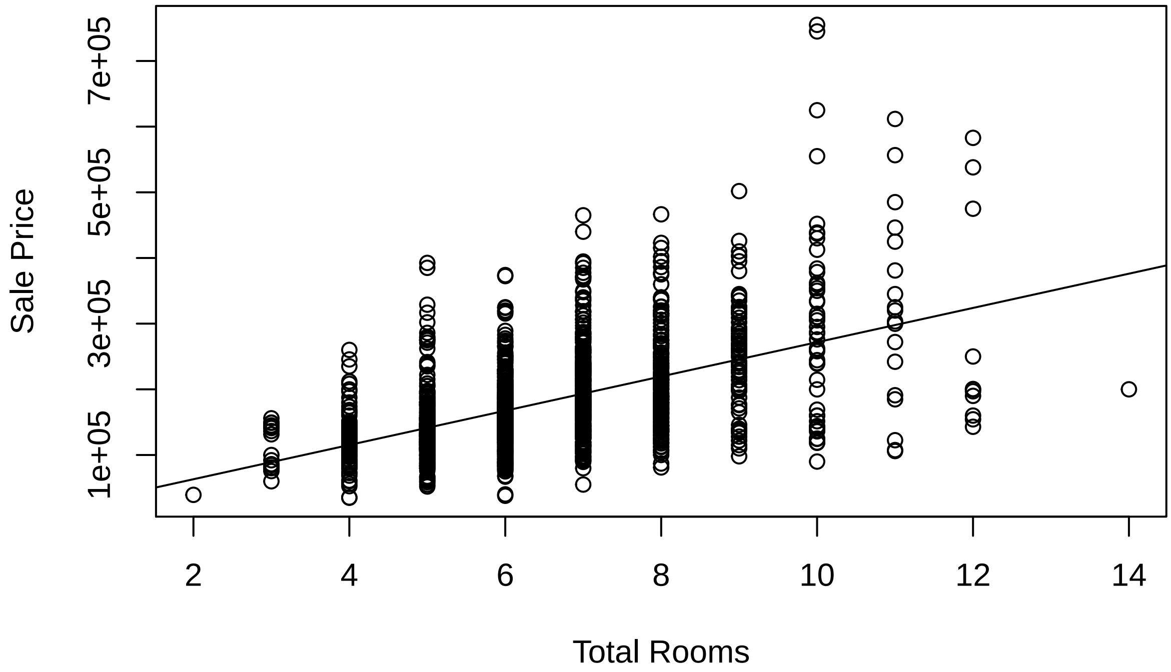
```
describe(Y) # Sale Price
```

```
##    vars    n     mean      sd median  trimmed     mad   min    max  range
## X1    1 1460 180921.2 79442.5 163000 170783.3 56338.8 34900 755000 720100
##    skew kurtosis      se
## X1 1.88      6.5 2079.11
```

```
hist(Y,main='Sale Price histogram',ylab='Sale Price')
```

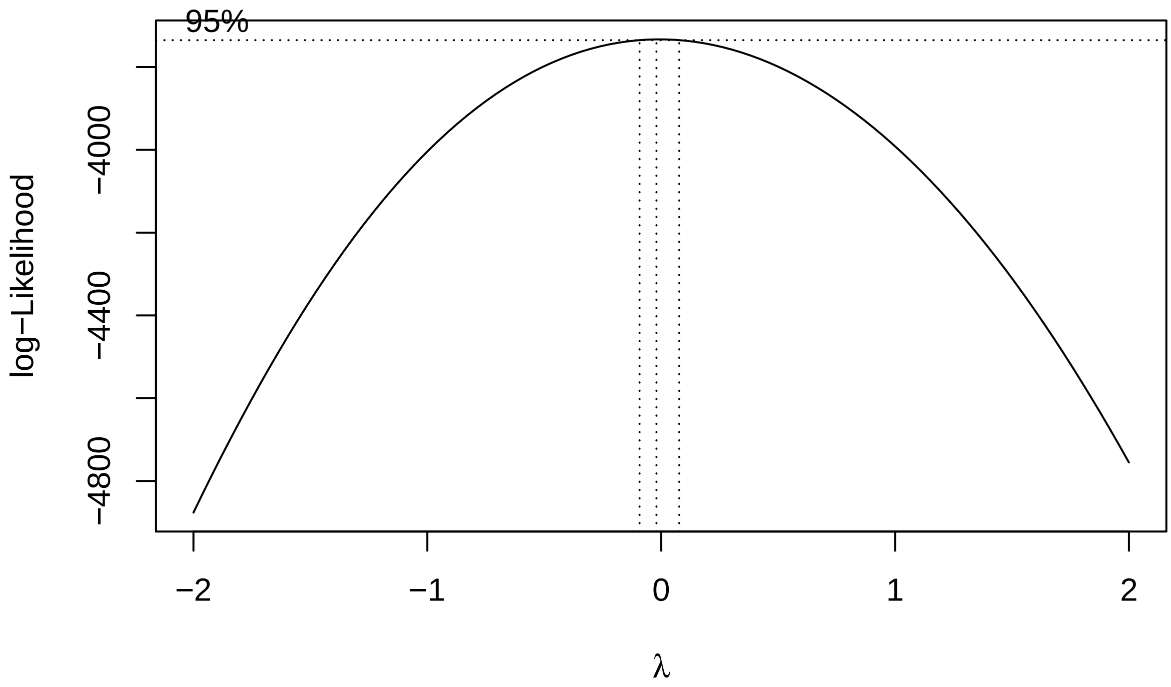


```r
plot(x=X,xlab = 'Total Rooms',y=Y,ylab='Sale Price')
abline(lm(Y~X))
```



Boxcox transformation and comparative plots

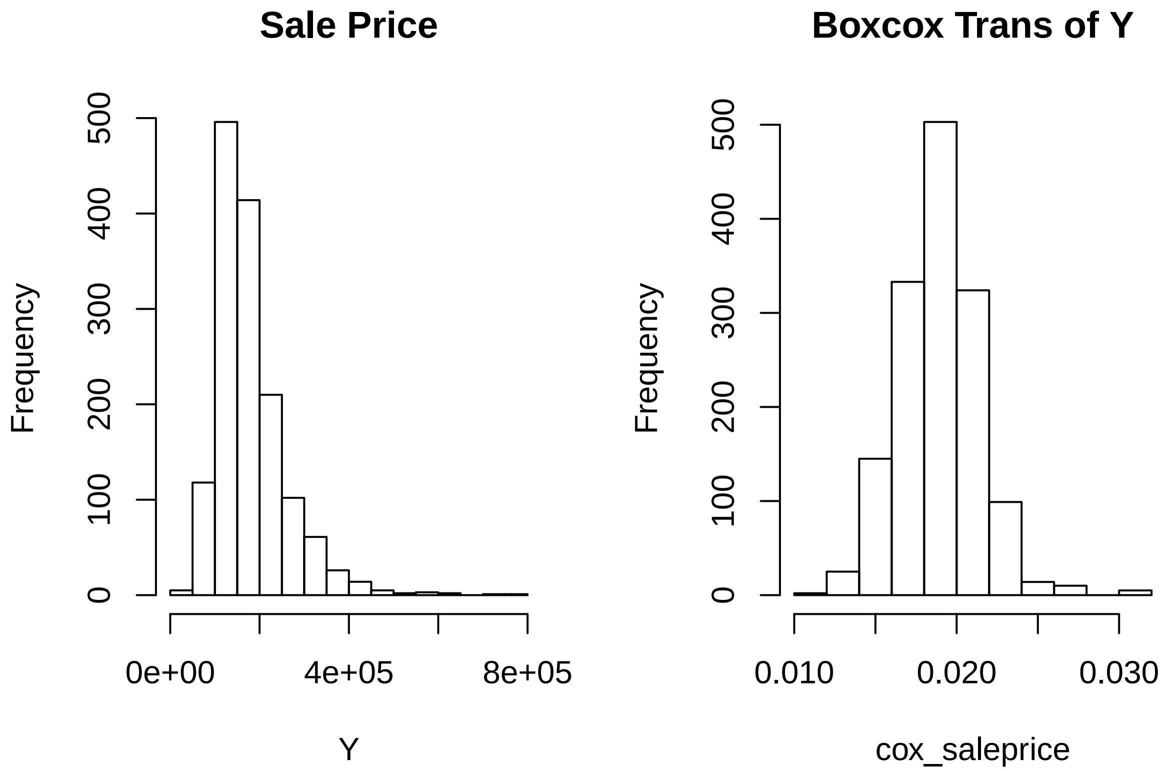
```
slm = lm(Y~X) #Create simple regression model
bc = boxcox(slm) # Find lambdas
```



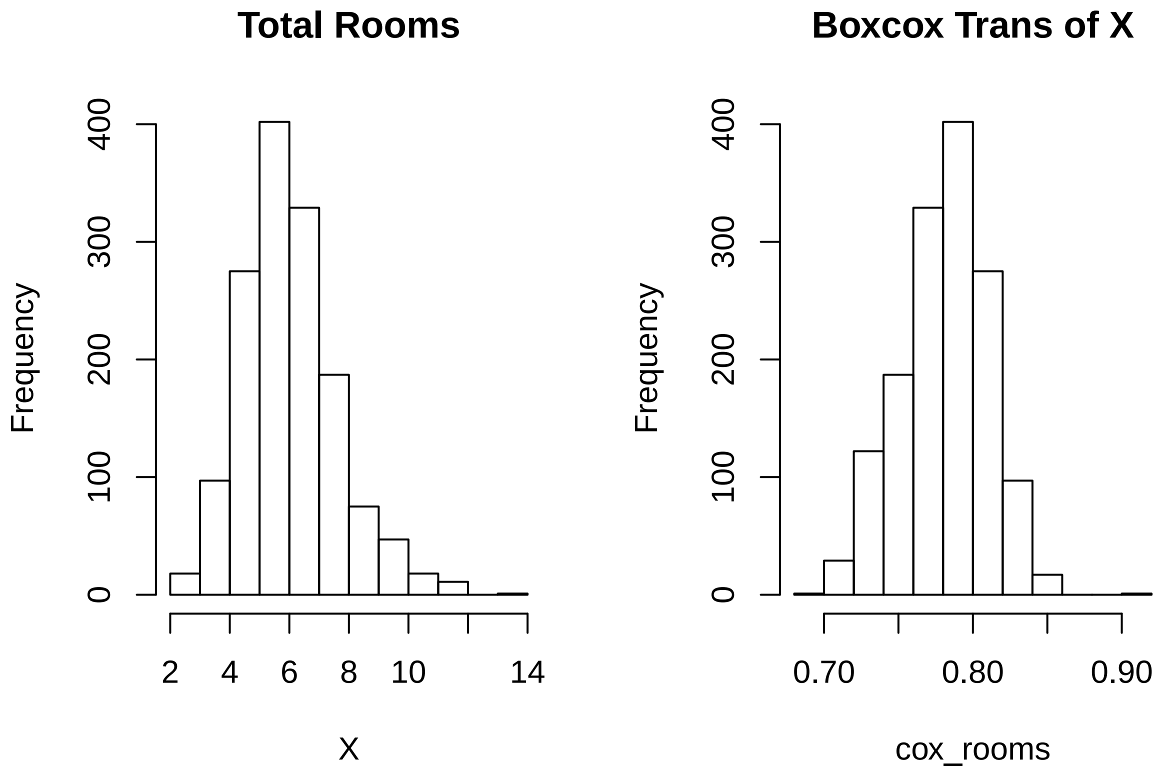
```
transformed_model = as.data.frame(bc)
lambdaM = bc$x[which.max(bc$y)] # Get lambda for model optimization; though we're not using it...Object
lambdaX = BoxCox.lambda(X) # lambda for X
lambdaY = BoxCox.lambda(Y) # lambda for Y
```

Lambdas are between -2 and 2 (-0.020202), boxcox transformation is appropriate.

```
cox_saleprice =  Y^lambdaY
cox_rooms = X^lambdaX
par(mfrow=c(1,2))
hist(Y,main="Sale Price")
hist(cox_saleprice,main="Boxcox Trans of Y")
```



```r
par(mfrow=c(1,2))
hist(X,main='Total Rooms')
hist(cox_rooms,main='Boxcox Trans of X')
```



———————> **Section 3**

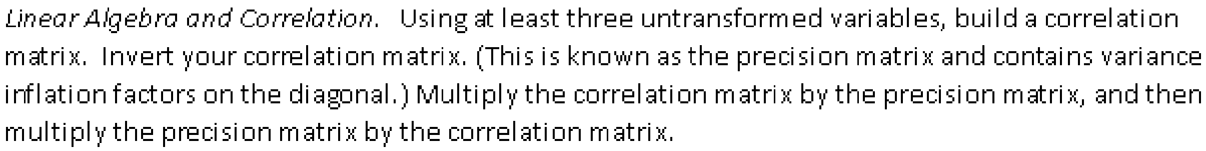

Linear Algebra and Correlation. Using at least three untransformed variables, build a correlation matrix. Invert your correlation matrix. (This is known as the precision matrix and contains variance inflation factors on the diagonal.) Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix.

Figure 3:

3 variables = Garage Area, Total Rooms Abv Grade, Sale Price

```
adf = data.frame(GarageArea=df$GarageArea,Rooms=df$TotRmsAbvGrd,SalePrice=df$SalePrice)
correlationMatrix = cor(adf)
correlationMatrix
```

```
##            GarageArea     Rooms SalePrice
## GarageArea  1.0000000 0.3378221 0.6234314
## Rooms       0.3378221 1.0000000 0.5337232
## SalePrice   0.6234314 0.5337232 1.0000000
```

```
invertedCorrelationMatrix = solve(correlationMatrix)
invertedCorrelationMatrix
```

```
##              GarageArea       Rooms  SalePrice
## GarageArea  1.63586571 -0.01162571 -1.0136452
## Rooms      -0.01162571  1.39841102 -0.7391165
## SalePrice  -1.01364520 -0.73911651  2.0264219
```

```
round(correlationMatrix%*%invertedCorrelationMatrix,0)
```

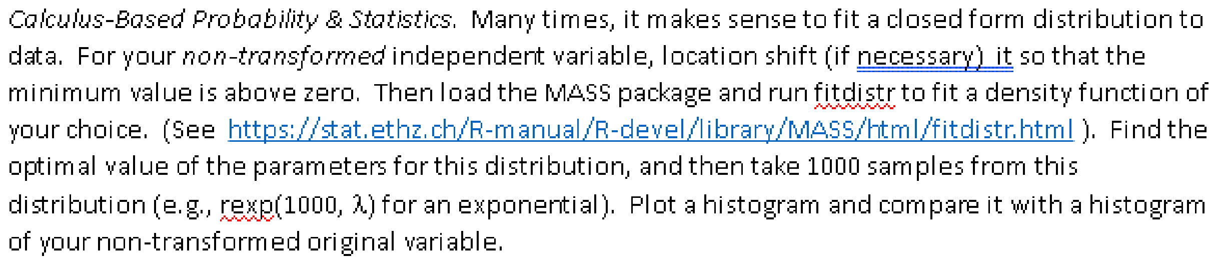
```
##            GarageArea Rooms SalePrice
## GarageArea          1     0         0
## Rooms               0     1         0
## SalePrice           0     0         1
```

```
round(invertedCorrelationMatrix%*%correlationMatrix,0)
```

```
##            GarageArea Rooms SalePrice
## GarageArea          1     0         0
## Rooms               0     1         0
## SalePrice           0     0         1
```

Ladies and gentlemen, the identity matrix!

——————————–> Section 4



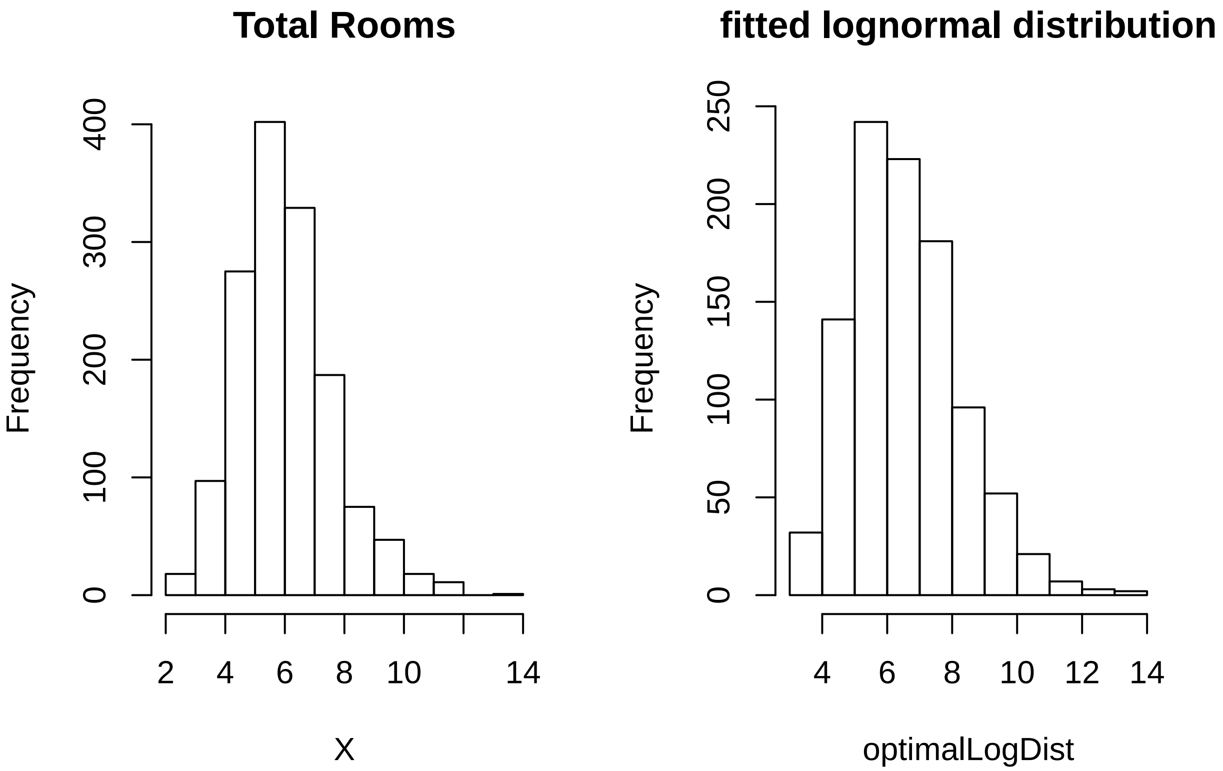
Calculus-Based Probability & Statistics. Many times, it makes sense to fit a closed form distribution to data. For your *non-transformed* independent variable, location shift (if necessary) it so that the minimum value is above zero. Then load the MASS package and run fitdistr to fit a density function of your choice. (See https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html). Find the optimal value of the parameters for this distribution, and then take 1000 samples from this distribution (e.g., rexp(1000, λ) for an exponential). Plot a histogram and compare it with a histogram of your non-transformed original variable.

Figure 4:

As seen before; total rooms variable is between 2 & 14. No location shift is necessary

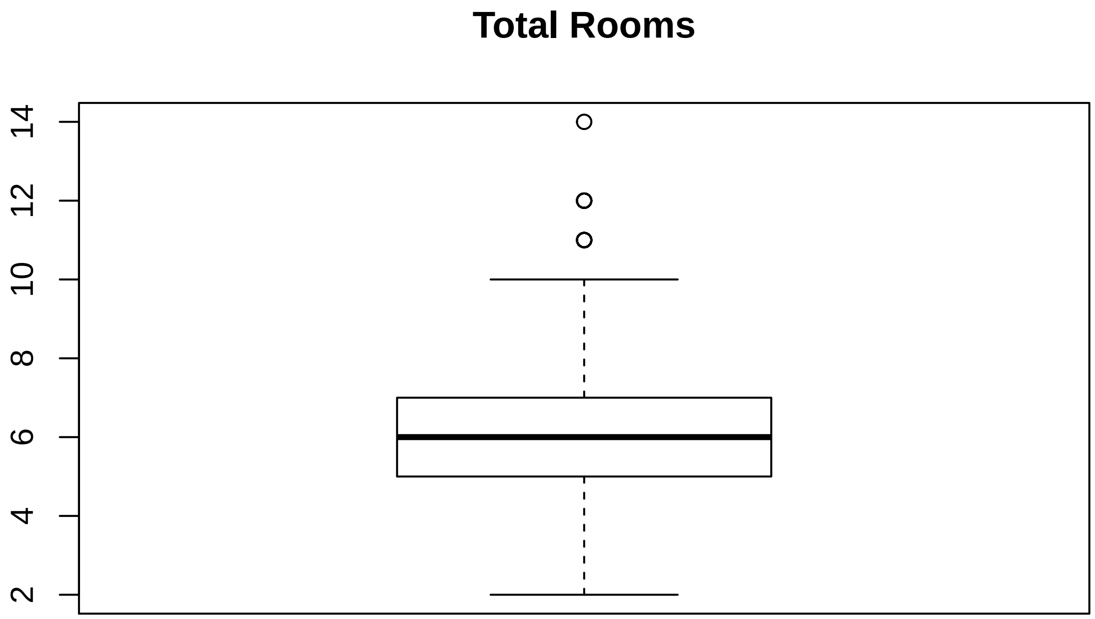
I believe the total rooms variable looks similar to a lognormal distribution. Let us see if I'm right.

```
aFit = fitdistr(X,'log-normal')
optimalLogDist = rlnorm(1000,meanlog = aFit$estimate[[1]],sdlog = aFit$estimate[[2]])
par(mfrow=c(1,2))
hist(X,main='Total Rooms')
hist(optimalLogDist,main='fitted lognormal distribution')
```

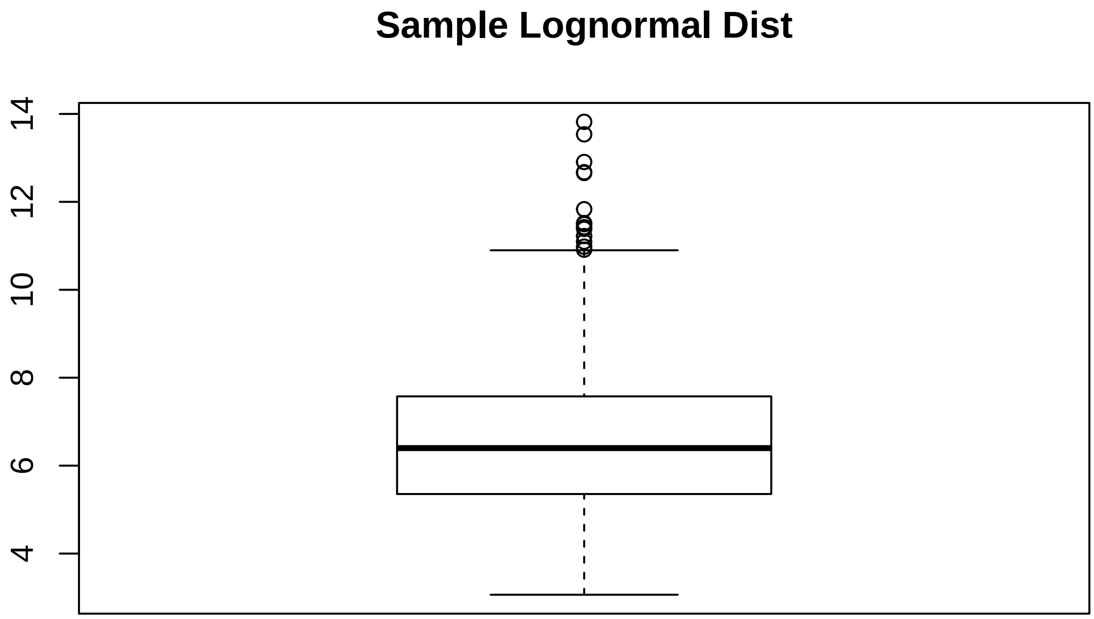


These histograms are are very similar, perhaps a boxplot might help interpret the differences.

```
boxplot(X,main='Total Rooms')
```

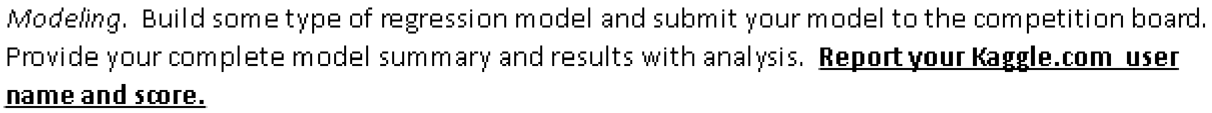


```
boxplot(optimalLogDist,main='Sample Lognormal Dist')
```



Looks like the lognormal distribution, with optimal parameters produces many high valued outliers (as expected of lognormal to have a greater right skew) and raises the 1st & 3rd quartiles + median. While keeping the distance between max and min relatively the same.

——————————> Section 5



Modeling. Build some type of regression model and submit your model to the competition board. Provide your complete model summary and results with analysis. **Report your Kaggle.com user name and score.**

Figure 5:

Build some type of regression model : Multiple linear regression

```
trainDF = data.frame(condition = df$OverallCond, quality = df$OverallQual, rooms = df$TotRmsAbvGrd, are
mlr = lm(Y~. , data=trainDF)
```

Model Summary

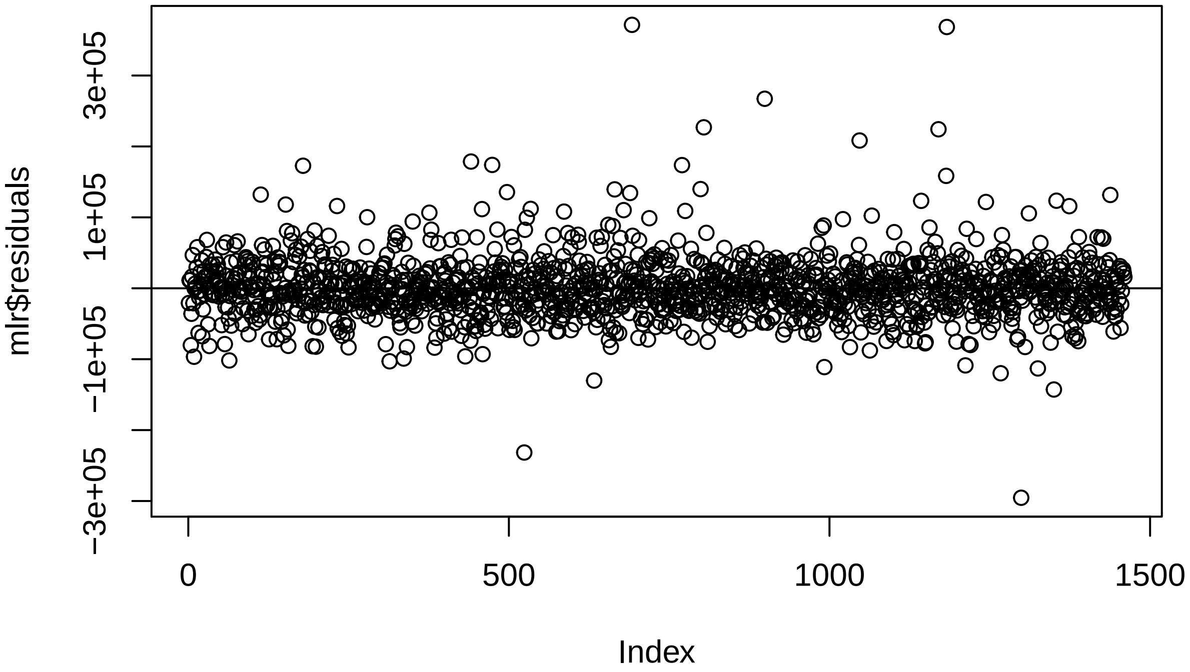
```
summary(mlr)
```

```
##
## Call:
## lm(formula = Y ~ ., data = trainDF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -295378  -24334   -2095   20872  371460
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.383e+05  8.624e+03  -16.04   <2e-16 ***
## condition   -1.347e+02  1.039e+03   -0.13    0.897
## quality      3.929e+04  9.237e+02   42.53   <2e-16 ***
## rooms        1.039e+04  7.940e+02   13.08   <2e-16 ***
## area         1.202e+00  1.175e-01   10.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43970 on 1455 degrees of freedom
## Multiple R-squared:  0.6945, Adjusted R-squared:  0.6936
## F-statistic: 826.7 on 4 and 1455 DF,  p-value: < 2.2e-16
```

High F-Statistic with an impossibly low p-value. We see a trend in my model to undervalue most sales. I imagine this is the direct result of having many low value observations with high leverage. This model explains almost 70% of the response variable variation.

Model Analysis

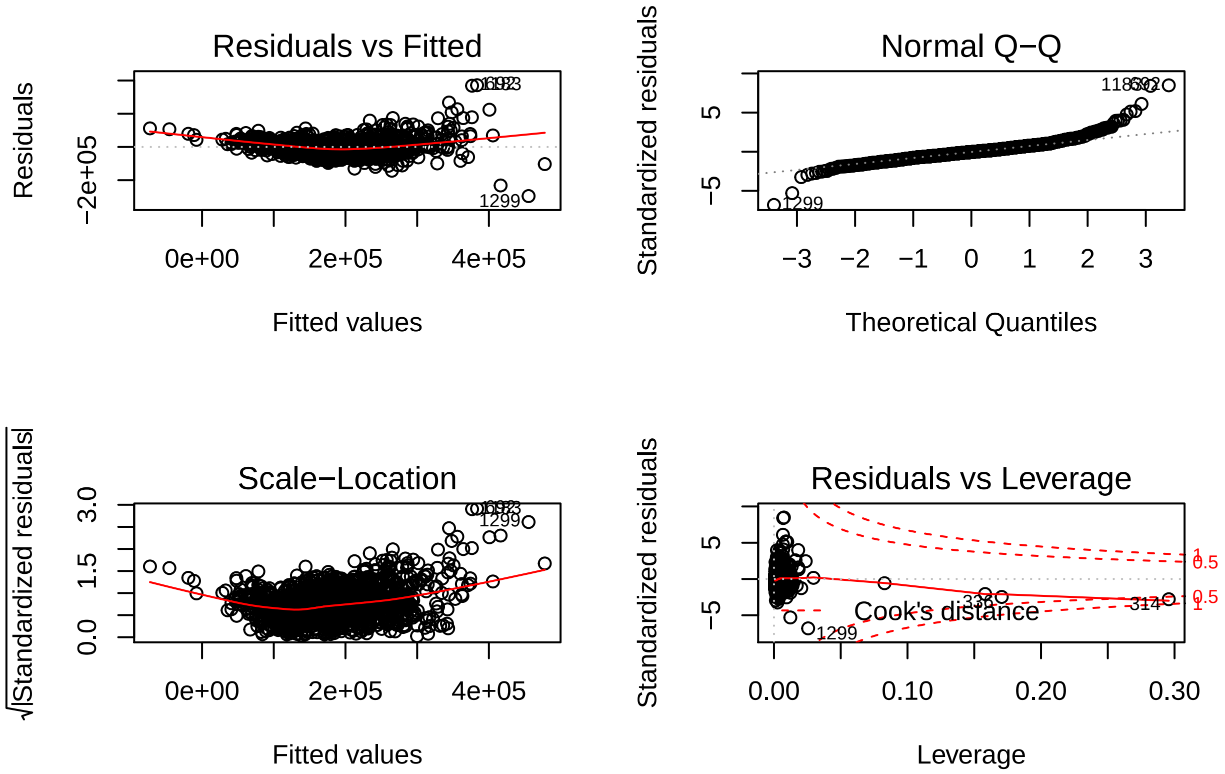
```
plot(mlr$residuals)
abline(0,0)
```



This model fits very well according to standard residual plot

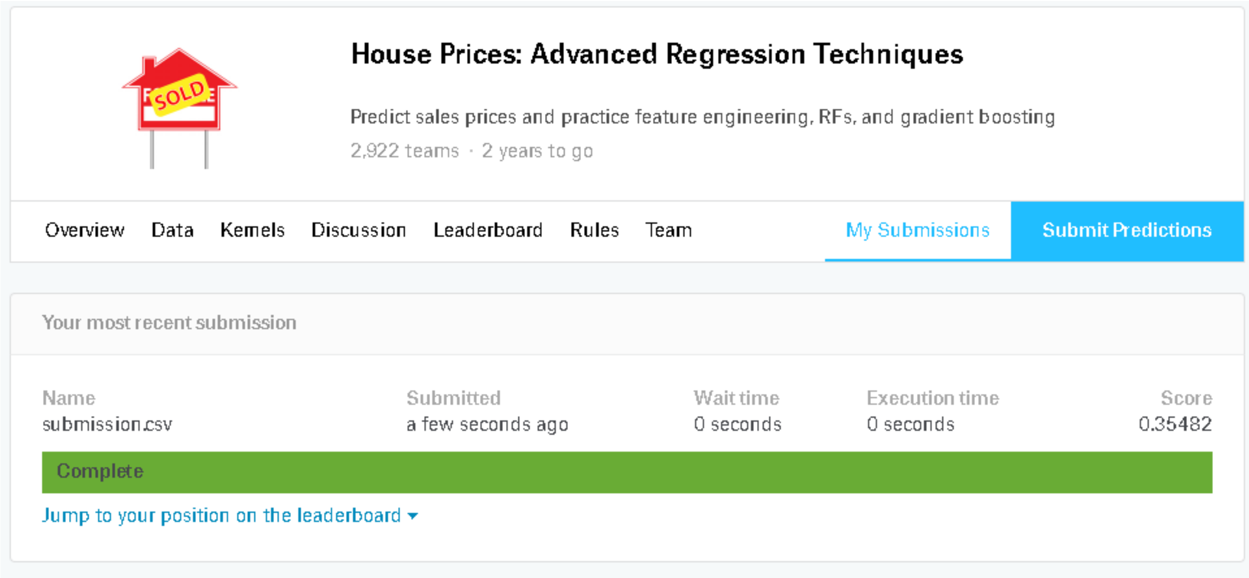
Analysis

```
par(mfrow = c(2,2))
plot(mlr)
```



My model performs well. These plots tell me my initial impression was correct; it was low value, high leverage observations that created an undervaluing trend in Sale Price. in order to come up with a much stronger model I would need to look at many more variables in order to determine which properties have unique aspects, treat them as outliers and remove them from the model fitting process. Judging from the normal Q-Q plot; I would want to slice off both extreme high and extreme low sale price observations for a better estimate on most Lots.

```
kaggleDF = read.csv('test.csv')
kaggleDF = data.frame(Id=kaggleDF$Id,condition=kaggleDF$OverallCond,quality=kaggleDF$OverallQual,rooms =
predictions = cbind(kaggleDF$Id,predict(mlr,kaggleDF))
colnames(predictions) = c('Id','SalePrice')
#write.csv(predictions,'C:/Users/Exped/Desktop/Data 604 Final/submission.csv',row.names=FALSE)
```



House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting

2,922 teams · 2 years to go

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission.csv	a few seconds ago	0 seconds	0 seconds	0.35482

Complete

Figure 6: