IS605 Final Exam

Marco Siqueira Campos

The solutions are in Github with follow address:

Pdf file: https://github.com/MarcoSCampos/testdata/blob/master/MAScampos_final.pdf

MSword file: https://github.com/MarcoSCampos/testdata/blob/master/MAScampos_final.docx

1. Instructions:

Your final is due by the end of day on 05/24/2016. You should post your solutions to your GitHub account. You are also expected to make a short presentation during our last meeting (3-5 minutes) or post a recording to the board. This project will show off your ability to understand the elements of the class.

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.

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.

Competition information:

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.

2. Probability:

Calculate as a minimum the below probabilities a through c. Assume the small letter "x" is estimated as the 4th quartile of the X variable, and the small letter "y" is estimated as the 2d quartile of the Y variable. Interpret the meaning of all probabilities.

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

Follow steps was made:

a-Choose one quantitative variable: **LotArea.**

b-Find the value of 2nd Quartile (median) for **SalePrice.**

c-Find the value of 4rd Quartile for **LotArea.**

d-Compute the frequency (probability) for combination **SalePrice** and **LotArea.**

e-Compute the probability *'a'* to *'c'*.

My X quantitative variable is **LotArea** and the 4thQuartile is the maximum value=215200

P(X>x)=P(LotArea>215245) = 0

P(X≤x)=P(LotArea≤215245) = 1

```
> library(psych)
> describe(train$LotArea)
   vars    n     mean      sd median trimmed     mad  min    max   range  skew kur
tosis
X1    1 1460 10516.83 9981.26 9478.5 9563.28 2962.23 1300 215245 213945 12.18   2
02.26
       se
X1 261.22
```

My Y variable is **SalePrice** and the 2nd Quartile (median) is = 163000

P(Y>y)=P(SalePrice>163000) = 0.49863

P(Y≤y)=P(SalePrice≤163000) = 0.50137

```
> describe(train$SalePrice)
   vars    n     mean      sd median  trimmed     mad   min    max  range skew
X1    1 1460 180921.2 79442.5 163000 170783.3 56338.8 34900 755000 720100 1.88
   kurtosis      se
X1      6.5 2079.1

> nrow(subset(train, SalePrice > 163000 ))
[1] 728
> nrow(subset(train, SalePrice > 163000 ))/length(train$SalePrice)
[1] 0.4986301
> length(train$SalePrice)-nrow(subset(train, SalePrice > 163000 ))
[1] 732
> (length(train$SalePrice)-nrow(subset(train, SalePrice > 163000 )))/length(train
$SalePrice)
[1] 0.5013699
```

Contingency table

X\Y	P(Y $\leq y$) P(SalePrice $\leq$16300)	P(Y>y) P(SalePrice>16300)	Total
P(X>x) P(LotArea>215245)	0	0	0
P(X$\leq x$) P(LotArea$\leq$215245)	732/1460	728/1460	1460/1460
total	732/1460	728/1460	1460/1460

Tab. 1 train$

```
> nrow(train[ which( train$LotArea < 215246 & train$SalePrice > 163000),])
[1] 728
> nrow(train[ which( train$LotArea < 215246 & train$SalePrice < 163001),])
[1] 732
> nrow(train[ which( train$LotArea > 215245 & train$SalePrice > 163000),])
[1] 0
> nrow(train[ which( train$LotArea > 215245 & train$SalePrice < 163001),])
[1] 0
```

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

P(A|B) = P(A∩ B) / P(B) = 0/(728/1460) = 0

b. P(X>x, Y>y)

P(A∩B) = 0

c. P(X<x | Y>y) => P(X≤x | Y>y)

P(A|B) = P(A∩ B) / P(B) = (728/1460) /(728/1460) = 1

Interpret the meaning of all probabilities.

a. Is the probability of P(X>x) with reduced sample space to P(Y>y)

In this case is probability of **LotArea**>215246 with only sample space of **SalePrice**>163000.

As P(**LotArea**>215246) = 0, the final probability will be 0.

b. Is the probability that both P(X>x) and P(Y>y) , is the probability that both P(X>x) and P(Y>y), or **LotArea**>215246 and **SalePrice**>163000 happen, is equivalent to P(A∩B).

As **LotArea**>215246 is 0 the probability will be zero.

c. Is the probability of P(X≤x) with reduced sample space to P(Y>y)

In this case is probability of **LotArea**≤215246 with only sample space of **SalePrice**>163000.

The final probability will be 1.

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

No, this splitting the train data cannot change the independence of data, for P(X and Y) is only P(X) * P(Y) when is the variable are independent.

To avoid cell with 0 at chi-squared test I will use for test X = P(X>x), x is the 3rd quartile and Y = P(Y>y), y is the 2nd quartile

Variable		Frequency	Probability
X = P(X>3Q) = P(LotArea>11601.5)	=	365	365/1460 = 0.25
Y = P(Y>2Q)= P(SalePrice>163000)	=	728	728/1460 = 0.4986

```
> quantile(train$LotArea, probs=0.75)
    75%
11601.5
> quantile(train$SalePrice, probs=0.5)
   50%
163000
> nrow(subset(train, SalePrice > 163000 ))
[1] 728
> nrow(subset(train, LotArea > 11601.5 ))
[1] 365
```

P(X|Y) = P(X)P(Y)

For P(X|Y) I counted the frequency in data base 276/728 = 0.379

P(X|Y)=0.379 ≠ P(X).P(Y) = 0.25 * 0.4986 = 0.12465

Contingency table for Chi-squared test

X\Y	**P(Y ≤ y)** P(SalePrice≤163000	**P(Y>y)** P(SalePrice>163000)	**Total**
P(X>x) P(LotArea>11601.5)	89	276	365
P(X≤ x) P(LotArea≤11601.5)	643	452	1095
total	732	728	1460

Tab. 2

```
> nrow(train[ which( train$LotArea > 11601.5 & train$SalePrice > 163000),])
[1] 276
> nrow(train[ which( train$LotArea > 11601.5 & train$SalePrice < 163001),])
[1] 89

> library(MASS)
> X_GT = c(89, 276)
> X_LT = c(643, 452)
> XY = as.data.frame(rbind(X_GT, X_LT))
> names(XY) = c('Y_LT', 'Y_GT')
> XY
     Y_LT Y_GT
X_GT   89  276
X_LT  643  452
> chisq.test(XY)
        Pearson's Chi-squared test with Yates' continuity correction
data:  XY
X-squared = 127.74, df = 1, p-value < 2.2e-16
```

The row and the column variables are statistically significantly associated (*p-value* < 0.05), there is no independence between X and Y.

3. Descriptive and inferencial 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. Using the transformed variables, run a correlation analysis and interpret. Test the hypothesis that the correlation between these variables is 0 and provide a 99% confidence interval. Discuss the meaning of your analysis.

Descriptive analysis for X, **LotArea**

```
> describe(train$LotArea)
   vars    n     mean      sd median trimmed     mad  min    max  range   skew kur
tosis
X1    1 1460 10516.83 9981.26 9478.5 9563.28 2962.23 1300 215245 213945 12.18   2
02.26
       se
X1 261.22


> par(mfrow=c(2,2))
> hist(train$LotArea, col = "blue")
> boxplot(train$LotArea, main="Boxplot LotArea")
> qqnorm(train$LotArea)
```

```
> qqline(train$LotArea)
> par(mfrow=c(1,1))
```

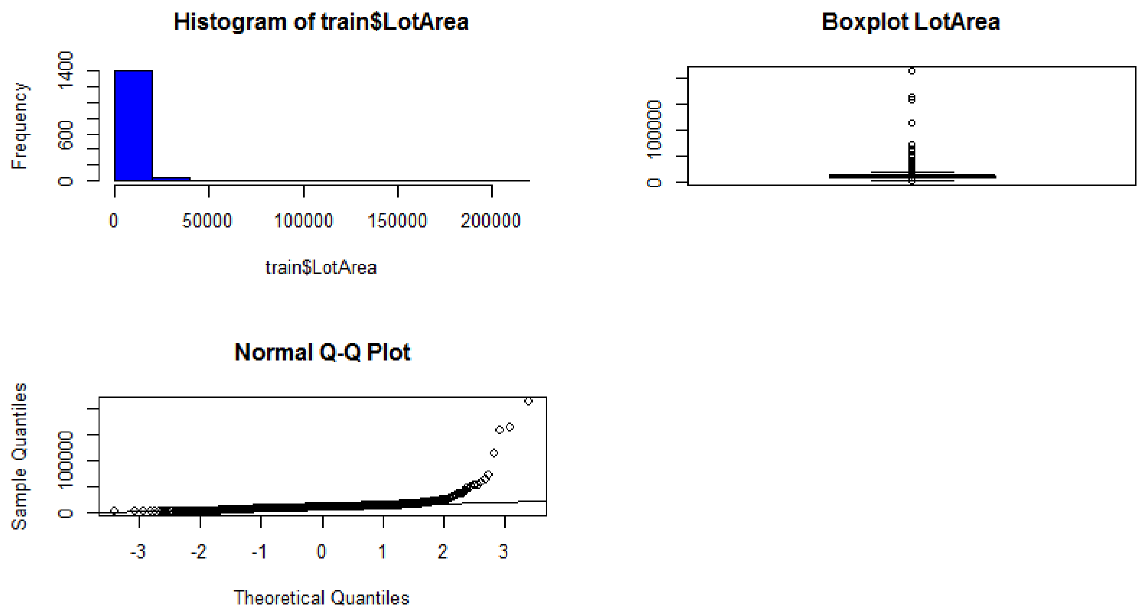


Fig. 1

Descriptive analysis for Y, **SalePrice**

```
> describe(train$SalePrice)
   vars    n     mean      sd median  trimmed     mad   min    max  range skew
X1    1 1460 180921.2 79442.5 163000 170783.3 56338.8 34900 755000 720100 1.88
   kurtosis     se
X1      6.5 2079.1
```

```
> par(mfrow=c(2,2))
> hist(train$SalePrice, col = "red")
> boxplot(train$SalePrice, main="Boxplot SalePrice")
> qqnorm(train$SalePrice)
> qqline(train$SalePrice)
> par(mfrow=c(1,1))
```

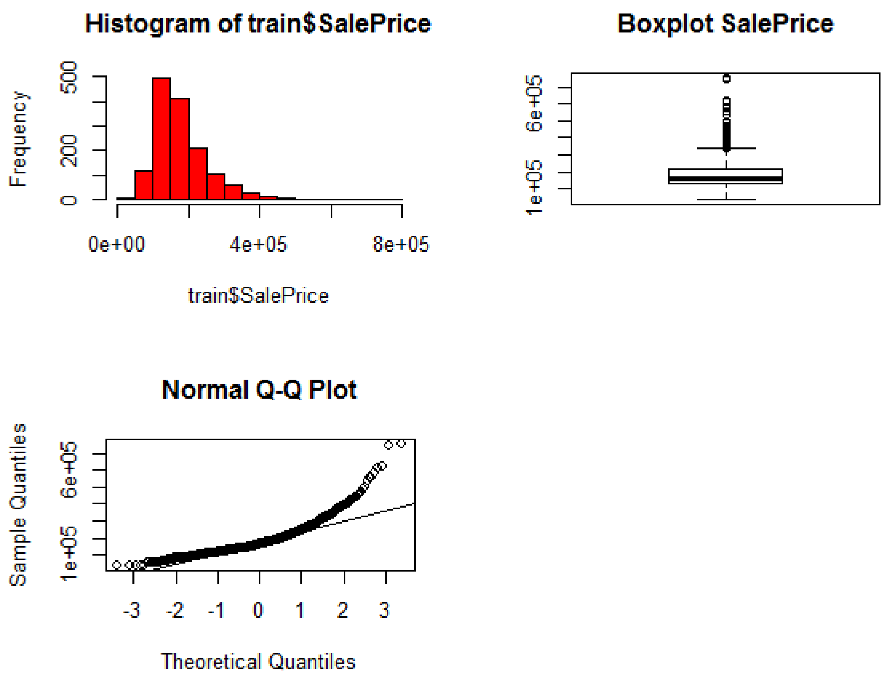


Fig.2

```
> plot(train$LotArea,train$SalePrice, main = "Scatterplot SalePrice by LotArea ",
col="red")
```

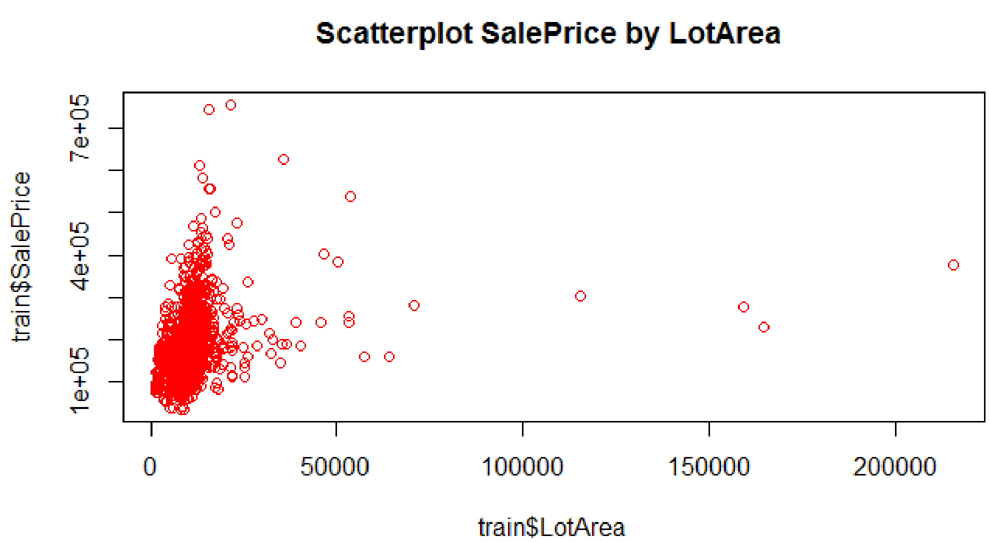


Fig. 3 Scatterplot Sales price and Lot Area

```
> cor.test(train$SalePrice,train$LotArea, method = "pearson", alternative
= "two.sided", estimate="rho", conf.level = 0.99)

        Pearson's product-moment correlation

data:  train$SalePrice and train$LotArea
t = 10.445, df = 1458, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
99 percent confidence interval:
 0.2000196 0.3254375
sample estimates:
      cor
0.2638434
```

Discuss:

The two distribution, are asymmetric, right skew, are not normal. The outliers at boxplot is not a real outlier is more characteristic at this kind of distribution.

The **LotArea** have a huge variation, we have a high concentration in beginning and we have some cases with very large area, the 3rd quartile is far from the maximum value.

The scatterplot show a light positive relationship between the two variables.

The correlation test without transformation show a positive and weak but significant correlation, with r=0.26.

For Box-Cox transformation our greatest objective is to reduce the non-normality of the residual for X (**LotArea**) and Y (**SalePrice**), because our main interest is in correlation between X and Y, and do not individually do the transformation of X and Y;

We found the lambda parameter of Y that give the better results at the residual.

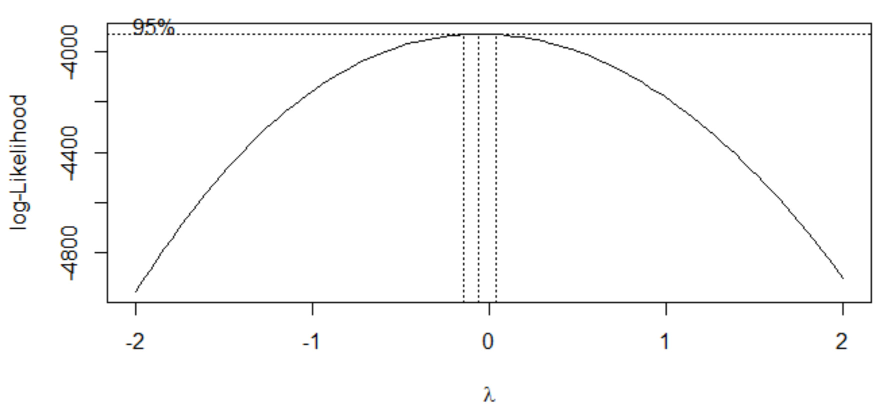


Fig. 4 Box-Cox Lambda Plot

The Box-Cox transformation Z = Y^{λ} = $Y^{-0.06}$

Lambda = -0.06060606

```
> library (MASS)
> lm( train$SalePrice~ train$LotArea)
> bc <- boxcox(train$SalePrice ~ train$LotArea)
> lambda <- bc$x[which.max(bc$y)]
> mnew <- lm(train$SalePrice^lambda ~ train$LotArea)
> op <- par(pty = "s", mfrow = c(1, 2))
> qqnorm(m$residuals, main="Normal QQ Plot - after"); qqline(m$residuals)
> qqnorm(mnew$residuals, main="Normal QQ Plot - before trans"); qqline(mnew$resid
uals)
> par(op)
```

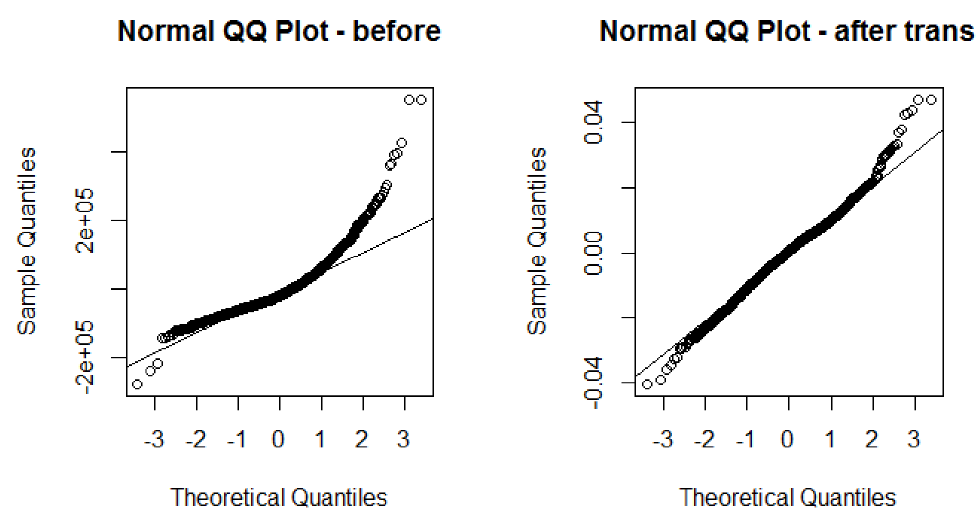


Fig. 4 Residual QQ Plot before and after lambda transformation

Discuss:
Correlation between X and Z (Y transformed).
The correlation between X and Z is: r = -0.2558218, see below.
Hypothesis test for:

H_0: θ = 0

H_1: θ ≠ 0

As $p < 0.05$, we reject the null hypothesis, H_0: θ = 0, and the correlation coefficient is statistically different from zero for significant level of α=0.05.

The confidence interval for 99% is P(-0.3177247 > θ > -0.1917473)=0.99

```
> cor.test(train$SalePrice^lambda,train$LotArea, method = "pearson", alternative
= "two.sided", estimate="rho", conf.level = 0.99)
```

```
        Pearson's product-moment correlation

data:  train$SalePrice^lambda and train$LotArea
t = -10.104, df = 1458, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
99 percent confidence interval:
 -0.3177247 -0.1917473
sample estimates:
       cor
-0.2558218
```

Discuss:
The meaning of the correlation analysis is: There is weak linear negative correlation between X and Z (Y transformed variable by Box-Cox). The correlation coefficient is -0.26, and the Confidence Interval with 0.99 is (-0.32, -0.19).
A warning must be made here, in the original correlation, X and Y, the correlation was positive, the transformation altered meaning. The analysis here must be done considering that the correlation is really positive rather than negative, in this case we have to consider only the value and not the signal.

4. Linear Algebra and Correlation:

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.

To do the correlation matrix the follow variable was choose: **SalePrice, LotArea, GarageArea** and **MasVnrArea**.

Correlation matrix

```
m<-train[,c("LotArea","SalePrice","GarageArea","MasVnrArea")]
cor(na.omit(m))
             LotArea SalePrice GarageArea MasVnrArea
LotArea    1.0000000 0.2646740  0.1807779  0.1041598
SalePrice  0.2646740 1.0000000  0.6224917  0.4774930
GarageArea 0.1807779 0.6224917  1.0000000  0.3730665
MasVnrArea 0.1041598 0.4774930  0.3730665  1.0000000
```

Invert the correlation matrix, the precision matrix

```
> mcor<-cor(na.omit(m))
> solve(mcor)
               LotArea  SalePrice  GarageArea  MasVnrArea
LotArea     1.07670387 -0.2811282 -0.03239281  0.03417217
SalePrice  -0.28112823  1.9162968 -0.94284209 -0.53399337
GarageArea -0.03239281 -0.9428421  1.65371565 -0.16337131
```

```
MasVnrArea  0.03417217 -0.5339934 -0.16337131  1.31236711
```

Multiplying the correlation matrix by the precision matrix

```
> mp<-solve(mcor)
> round(mcor%*%mp,4)
           LotArea SalePrice GarageArea MasVnrArea
LotArea          1         0          0          0
SalePrice        0         1          0          0
GarageArea       0         0          1          0
MasVnrArea       0         0          0          1
```

Multiplying the precision matrix by correlation matrix

```
> round(mp%*%mcor,4)
           LotArea SalePrice GarageArea MasVnrArea
LotArea          1         0          0          0
SalePrice        0         1          0          0
GarageArea       0         0          1          0
MasVnrArea       0         0          0          1
```

For both, multiplying the correlation matrix by the precision matrix and multiplying the precision matrix by correlation matrix, the results is identity matrix.

5. 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 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.

I choose **LotArea** for analyses the distribution

Check the minimum value:

```
> min(train$LotArea)
[1] 1300
```

Check if the distribution fit with Weibull and exponential distribution.

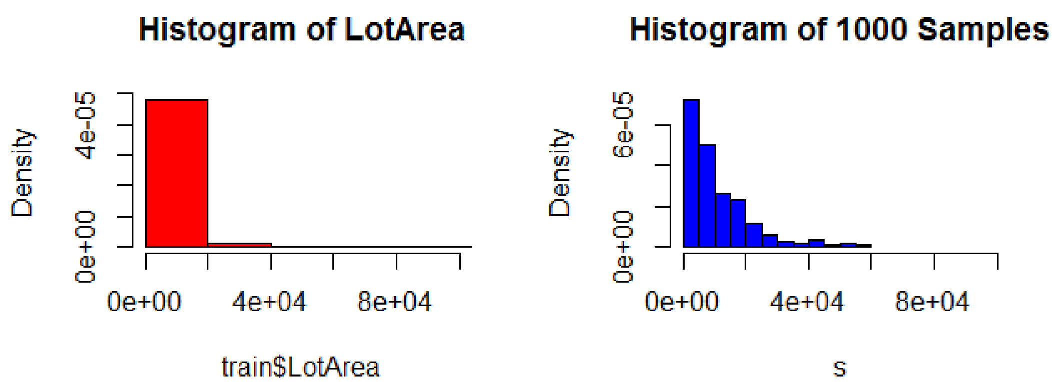
```
> fitdistr(train$LotArea, "weibull")
      shape           scale
```

```
  1.448518e+00    1.158547e+04
 (2.131213e-02) (2.211674e+02)
```

```
> fitdistr(train$LotArea, "exponential")
       rate
  9.508570e-05
 (2.488507e-06)
```

Exponential distribution was chosen, it has lower error.

```
> s<- rexp(1000,fitdistr(train$LotArea,"exponential")$estimate)
> par(mfrow=c(1,2))
> hist(train$LotArea, freq = FALSE, main="Histogram of LotArea", xlim = c(0,10000
0), col = "red")
> hist(s, freq = FALSE, main = "Histogram of 1000 Samples", xlim=c(0,100000), col
="blue")
> par(mfrow=c(1,1))
```



Comparing percentiles 1%, 5%, 50%, 95% and 99% from original and modeled data.

```
> quantile(train$LotArea, probs = c(0.01,0.05,0.5,0.95,0.99))
      1%        5%       50%       95%       99% 
 1680.00   3311.70   9478.50 17401.15 37567.64 
> qexp(c(0.01,0.05,0.5,0.95,0.99), rate = fitdistr(train$LotArea,"exponential")$e
stimate, lower.tail = TRUE, log.p = FALSE)
[1]    105.6977    539.4428   7289.7097 31505.6013 48431.7831
```

	1%	5%	50%	95%	99%
Original	1,680.00	3,311.70	9,478.50	17,401.15	37,567.64
Modeled	105.70	539.44	7,289.71	31,505.60	48,431.78

Tab. 3

Discuss:
There are significant differences between the original data and the modeled data, the original data are more concentrated, the decay of the modeled data is smoother on the right. The biggest different is at right tail the 95% percentile occurs at value 17,401.15, earlier, for original data and 31,05.60 for modeled data.

6. 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.**

For do this I adopted the standard lm multiple regression with factors, due to time constraint only a very simple approach was used, the focus was run the model not optimization/competition.

a. The first step was selecting the variables, removing the variables with high auto-correlation, high VIF and removing variables with high *p*-value. To save space the steps here was omitted.
b. One main issue is how to deal with missing values, the following strategy was done, the idea is working with simple approach:
    a. For quantitative variable, the NA was changed for median, for train and for test data.
    b. For factor variables NA was converted to a factor, was created a "dummy" factor for NA. (of course only in the case it was significant)
    c. In the case at we have NA in factor variable only in the train data (in my model, 3 cases: "*MSZoning*", "*Exterior1st*" and "*KitchenQual*") this variables was dropped from the model.

```
# load file
train<-read.csv("train.csv",stringsAsFactors=FALSE)
test<-read.csv("test.csv",stringsAsFactors=FALSE)

# bind the files to simplify
full<-bind_rows(train,test)

#create dummy factor
full$MasVnrType[is.na(full$MasVnrType)]<-"wo"
…
# change to factor
full$street<-as.factor(full$Street)
full$LandContour<-as.factor(full$LandContour)
full$LotConfig<-as.factor(full$LotConfig)
full$LandSlope<-as.factor(full$LandSlope)
full$Neighborhood<-as.factor(full$Neighborhood)
full$Condition1<-as.factor(full$Condition1)
full$MasVnrType<-as.factor(full$MasVnrType)
```

```
....
# split the files
ftrain<-full[1:1460,]
ftest<-full[1461:2919,]
# Change NA for median
ftrain$MasVnrArea[is.na(ftrain$MasVnrArea)]<-median(na.omit(ftrain$MasVnrArea))
ftest$LotFrontage[is.na(ftest$LotFrontage)]<-median(na.omit(ftest$LotFrontage))
ftest$MasVnrArea[is.na(ftest$MasVnrArea)]<-median(na.omit(ftest$MasVnrArea))
ftest$BsmtFinSF1[is.na(ftest$BsmtFinSF1)]<-median(na.omit(ftest$BsmtFinSF1))
ftest$BsmtFinSF2[is.na(ftest$BsmtFinSF2)]<-median(na.omit(ftest$BsmtFinSF2))
....
# fit the model
fit1 <- lm(SalePrice ~ LotArea+OverallQual+OverallCond+YearBuilt+
        MasVnrArea+BsmtFinSF2+BsmtUnfSF+TotalBsmtSF+
        X1stFlrSF+X2ndFlrSF+BedroomAbvGr+
        KitchenAbvGr+Fireplaces+GarageCars+
        Street+LandContour+LotConfig+
        LandSlope+Neighborhood+Condition1+Condition2+
        BldgType+RoofMatl+MasVnrType+
        ExterQual+BsmtQual+BsmtExposure+
        GarageQual+GarageCond+PoolQC+MoSold, data=ftrain)
```

Model performance for train data:

```
summary(fit1)

Call:
lm(formula = SalePrice ~ LotArea + OverallQual + OverallCond +
    YearBuilt + MasVnrArea + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF +
    X1stFlrSF + X2ndFlrSF + BedroomAbvGr + KitchenAbvGr + Fireplaces +
    GarageCars + Street + LandContour + LotConfig + LandSlope +
    Neighborhood + Condition1 + Condition2 + BldgType + RoofMatl +
    MasVnrType + ExterQual + BsmtQual + BsmtExposure + GarageQual +
    GarageCond + PoolQC + MoSold, data = ftrain)

Residuals:
    Min      1Q  Median      3Q     Max
-180755  -10182     432   10412  180755

Coefficients: (1 not defined because of singularities)
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -1.493e+06  1.273e+05 -11.728  < 2e-16 ***
LotArea              6.816e-01  9.565e-02   7.126 1.68e-12 ***
OverallQual          7.760e+03  9.432e+02   8.228 4.46e-16 ***
OverallCond          6.957e+03  6.851e+02  10.154  < 2e-16 ***
YearBuilt            4.778e+02  5.724e+01   8.348  < 2e-16 ***
MasVnrArea           2.183e+01  5.764e+00   3.786 0.000160 ***
BsmtFinSF2          -1.374e+01  4.413e+00  -3.114 0.001884 **
```

```
BsmtUnfSF           -1.814e+01  1.921e+00  -9.441  < 2e-16 ***
TotalBsmtSF          4.241e+01  4.225e+00  10.036  < 2e-16 ***
X1stFlrSF            5.271e+01  4.244e+00  12.420  < 2e-16 ***
X2ndFlrSF            5.939e+01  2.645e+00  22.459  < 2e-16 ***
BedroomAbvGr        -4.596e+03  1.165e+03  -3.947 8.34e-05 ***
KitchenAbvGr        -1.554e+04  5.173e+03  -3.004 0.002718 **
Fireplaces           2.794e+03  1.290e+03   2.165 0.030536 *
GarageCars           8.635e+03  1.519e+03   5.684 1.61e-08 ***
StreetPave           3.707e+04  1.149e+04   3.226 0.001284 **
LandContourHLS       1.090e+04  5.031e+03   2.166 0.030463 *
LandContourLow      -7.269e+03  6.093e+03  -1.193 0.233102
LandContourLvl       5.398e+03  3.548e+03   1.521 0.128391
LotConfigCulDSac     4.867e+03  3.104e+03   1.568 0.117113
LotConfigFR2        -7.417e+03  3.946e+03  -1.880 0.060345 .
LotConfigFR3        -1.274e+04  1.288e+04  -0.989 0.322987
LotConfigInside     -6.238e+02  1.733e+03  -0.360 0.718948
LandSlopeMod         4.763e+03  3.835e+03   1.242 0.214469
LandSlopeSev        -2.747e+04  9.828e+03  -2.796 0.005254 **
NeighborhoodBlueste -1.397e+04  1.837e+04  -0.760 0.447115
NeighborhoodBrDale  -1.160e+04  9.782e+03  -1.186 0.235742
NeighborhoodBrkSide -1.391e+04  8.193e+03  -1.698 0.089774 .
NeighborhoodClearCr -1.942e+04  8.733e+03  -2.224 0.026322 *
NeighborhoodCollgCr -1.468e+04  6.962e+03  -2.108 0.035231 *
NeighborhoodCrawfor  3.260e+03  8.141e+03   0.400 0.688872
NeighborhoodEdwards -2.556e+04  7.542e+03  -3.389 0.000722 ***
NeighborhoodGilbert -2.007e+04  7.406e+03  -2.709 0.006825 **
NeighborhoodIDOTRR  -2.032e+04  8.709e+03  -2.333 0.019789 *
NeighborhoodMeadowV -1.241e+04  9.138e+03  -1.358 0.174622
NeighborhoodMitchel -3.370e+04  7.737e+03  -4.356 1.43e-05 ***
NeighborhoodNAmes   -2.428e+04  7.355e+03  -3.301 0.000990 ***
NeighborhoodNoRidge  1.732e+04  8.064e+03   2.148 0.031859 *
NeighborhoodNPkVill -2.207e+03  1.052e+04  -0.210 0.833805
NeighborhoodNridgHt  1.695e+04  7.307e+03   2.320 0.020485 *
NeighborhoodNWAmes  -2.893e+04  7.560e+03  -3.827 0.000136 ***
NeighborhoodOldTown -2.264e+04  8.007e+03  -2.827 0.004766 **
NeighborhoodSawyer  -2.077e+04  7.752e+03  -2.680 0.007463 **
NeighborhoodSawyerW -1.511e+04  7.447e+03  -2.029 0.042699 *
NeighborhoodSomerst  2.872e+03  7.115e+03   0.404 0.686502
NeighborhoodStoneBr  3.311e+04  8.120e+03   4.078 4.81e-05 ***
NeighborhoodSWISU   -1.321e+04  9.145e+03  -1.445 0.148788
NeighborhoodTimber  -2.223e+04  7.928e+03  -2.804 0.005118 **
NeighborhoodVeenker -2.311e+03  1.000e+04  -0.231 0.817337
Condition1Feedr      5.995e+03  4.876e+03   1.230 0.219043
Condition1Norm       1.425e+04  4.003e+03   3.561 0.000383 ***
Condition1PosA       1.318e+04  9.737e+03   1.354 0.176010
```

```
Condition1PosN       1.705e+04  7.232e+03   2.358 0.018536 *
Condition1RRAe      -1.260e+04  8.594e+03  -1.467 0.142734
Condition1RRAn       1.401e+04  6.665e+03   2.102 0.035729 *
Condition1RRNe       3.142e+03  1.785e+04   0.176 0.860313
Condition1RRNn       3.888e+03  1.237e+04   0.314 0.753243
Condition2Feedr     -6.777e+03  2.212e+04  -0.306 0.759415
Condition2Norm      -6.789e+03  1.909e+04  -0.356 0.722212
Condition2PosA       2.574e+04  3.127e+04   0.823 0.410557
Condition2PosN      -2.317e+05  2.685e+04  -8.632  < 2e-16 ***
Condition2RRAe      -2.021e+04  3.105e+04  -0.651 0.515182
Condition2RRAn      -8.686e+03  3.103e+04  -0.280 0.779542
Condition2RRNn      -1.165e+03  2.594e+04  -0.045 0.964183
BldgType2fmCon      -6.167e+03  5.591e+03  -1.103 0.270245
BldgTypeDuplex      -7.115e+03  5.750e+03  -1.238 0.216109
BldgTypeTwnhs       -3.385e+04  5.057e+03  -6.693 3.20e-11 ***
BldgTypeTwnhsE      -2.525e+04  3.276e+03  -7.709 2.45e-14 ***
RoofMatlCompShg      6.781e+05  3.435e+04  19.743  < 2e-16 ***
RoofMatlMembran      7.284e+05  4.377e+04  16.644  < 2e-16 ***
RoofMatlMetal        7.125e+05  4.380e+04  16.267  < 2e-16 ***
RoofMatlRoll         6.739e+05  4.222e+04  15.960  < 2e-16 ***
RoofMatlTar&Grv      6.671e+05  3.460e+04  19.278  < 2e-16 ***
RoofMatlWdShake      6.819e+05  3.629e+04  18.790  < 2e-16 ***
RoofMatlWdShngl      7.220e+05  3.531e+04  20.450  < 2e-16 ***
MasVnrTypeBrkFace    1.310e+04  6.546e+03   2.002 0.045518 *
MasVnrTypeNone       1.928e+04  6.608e+03   2.917 0.003590 **
MasVnrTypeStone      1.946e+04  6.969e+03   2.793 0.005295 **
MasVnrTypewo         1.062e+04  1.092e+04   0.972 0.331014
ExterQualFa         -2.418e+04  9.205e+03  -2.627 0.008712 **
ExterQualGd         -3.480e+04  4.460e+03  -7.804 1.19e-14 ***
ExterQualTA         -3.761e+04  4.954e+03  -7.591 5.90e-14 ***
BsmtQualFa          -1.640e+04  6.049e+03  -2.711 0.006797 **
BsmtQualGd          -2.794e+04  3.225e+03  -8.663  < 2e-16 ***
BsmtQualTA          -2.390e+04  3.968e+03  -6.024 2.19e-09 ***
BsmtQualwo           1.739e+03  2.523e+04   0.069 0.945057
BsmtExposureGd       1.402e+04  2.988e+03   4.693 2.97e-06 ***
BsmtExposureMn      -2.996e+03  2.937e+03  -1.020 0.307981
BsmtExposureNo      -5.515e+03  2.016e+03  -2.735 0.006316 **
BsmtExposurewo      -1.064e+04  2.424e+04  -0.439 0.660660
GarageQualFa        -1.532e+05  2.745e+04  -5.580 2.91e-08 ***
GarageQualGd        -1.471e+05  2.810e+04  -5.237 1.90e-07 ***
GarageQualPo        -1.713e+05  3.365e+04  -5.090 4.08e-07 ***
GarageQualTA        -1.528e+05  2.714e+04  -5.629 2.20e-08 ***
GarageQualwo        -7.513e+03  1.751e+04  -0.429 0.667890
GarageCondFa         1.307e+05  3.256e+04   4.014 6.29e-05 ***
GarageCondGd         1.247e+05  3.336e+04   3.737 0.000194 ***
```

```
GarageCondPo         1.421e+05  3.488e+04   4.074 4.90e-05 ***
GarageCondTA         1.341e+05  3.217e+04   4.167 3.28e-05 ***
PoolQCFa            -1.040e+05  2.484e+04  -4.185 3.04e-05 ***
PoolQCGd            -5.573e+04  2.550e+04  -2.186 0.028997 *
PoolQCwo            -1.095e+05  1.781e+04  -6.148 1.03e-09 ***
MoSold2             -9.617e+03  4.757e+03  -2.022 0.043412 *
MoSold3             -4.175e+03  4.079e+03  -1.024 0.306143
MoSold4             -4.194e+03  3.893e+03  -1.077 0.281521
MoSold5              8.116e+02  3.728e+03   0.218 0.827673
MoSold6             -2.854e+03  3.655e+03  -0.781 0.435018
MoSold7             -1.461e+03  3.685e+03  -0.397 0.691735
MoSold8             -5.890e+03  3.962e+03  -1.486 0.137383
MoSold9             -3.758e+03  4.516e+03  -0.832 0.405431
MoSold10            -9.474e+03  4.230e+03  -2.240 0.025271 *
MoSold11            -4.480e+03  4.279e+03  -1.047 0.295284
MoSold12            -3.971e+03  4.591e+03  -0.865 0.387253
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23810 on 1347 degrees of freedom
Multiple R-squared:  0.9171,	Adjusted R-squared:  0.9102 
F-statistic:   133 on 112 and 1347 DF,  p-value: < 2.2e-16
```

```
# predict with new data, test data
pred1<-as.data.frame(predict(fit1, newdata = ftest))

# Organize the file to send to kaggle
pred1 <- rownames_to_column(pred1, "Id")
names(pred1)[names(pred1)=="predict(fit1, newdata = ftest)"] <- "SalePrice"
pred1$Id<-as.numeric(pred1$Id)

# file to send
write.csv(pred1, "Kaggle.csv", row.names = FALSE)
```

However, using the standard way for multiple regression gave negative value for test data, a new approach needs to be done, what I did:

a) Remove the intercept.
b) Remove coefficients with negative value.

New regression model:

```
fit2 <- lm(SalePrice ~ LotArea+MasVnrArea+OverallQual+OverallCond+
          MasVnrArea+TotalBsmtSF+
          X1stFlrSF+X2ndFlrSF+
          Fireplaces+GarageCars+
          Street+LandContour+LotConfig+
```

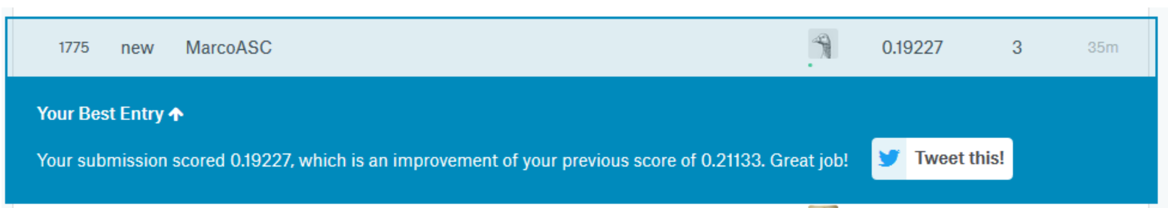
```
             LandSlope+Neighborhood+Condition1+Condition2+
             BldgType+RoofMatl+MasVnrType+
             ExterQual+BsmtQual+BsmtExposure+
             GarageQual+GarageCond+PoolQC+MoSold-1,data=ftrain)

# predict with new data test data
pred2<-as.data.frame(predict(fit2, newdata = ftest))

# Organize the file to send to kaggle
pred2 <- rownames_to_column(pred2, "Id")
names(pred2)[names(pred2)=="predict(fit2, newdata = ftest)"] <- "SalePrice"
pred2$Id<-as.numeric(pred1$Id)

# file to send
write.csv(pred2, "Kaggle1.csv", row.names = FALSE)
```

For this was possible run the multiple regression and gave the follow result from Kaggle:



This model didn’t take advantage of all variable I tried other approach, I did a random forest regression with the main significant variable, follow my model.

```
set.seed(0808)
ranf = randomForest(formula=SalePrice ~ LotArea+OverallQual+OverallCond+YearBuilt+
                    MasVnrArea+BsmtFinSF2+BsmtUnfSF+TotalBsmtSF+
                    X1stFlrSF+X2ndFlrSF+BedroomAbvGr+
                    KitchenAbvGr+Fireplaces+GarageCars+
                    LandSlope+Neighborhood+
                    Condition1+Condition2+
                    BldgType+RoofMatl+MasVnrType+
                    ExterQual+BsmtQual+BsmtExposure+
                    Functional+GarageQual+GarageCond+
                    PoolQC+MoSold, data=ftrain)


# predict with new data test data
previsao = predict(ranf,ftrain)

# Organize the file to send to kaggle
pred3 <- rownames_to_column(previsao, "Id")
names(pred3)[names(pred3)=="predict(ranf, ftest)"] <- "SalePrice"
pred3$Id<-as.numeric(pred3$Id)
```

# file to send
write.csv(pred3, "Kaggle3.csv", row.names = FALSE)

This model improved the results, with lower error and I jumped 225 positions.

Discuss:
The fist model give nice fit R^2=0.92 for train data using main numerical and factor variable, totaling 31 variables, and it was possible to predict all cases for test data, without any NA, however when I try to do with test data the model predict negative value with test data.
I removed the intercept and all variables with negative coefficients, the model ran at Kaggle and was possible received a rank position.
However, the last approach was not smart because drop significant variables, I tried random forest regression that gave a better result.

References:

1- https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/boxcox.html
2- http://rcompanion.org/handbook/I_12.html
3- http://stackoverflow.com/questions/33999512/how-to-use-the-box-cox-power-transformation-in-r
4- https://stat.ethz.ch/R-manual/R-patched/library/stats/html/cor.test.html
5- https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html