Data 605 Final

Alejandro Osborne

December 20, 2017

```r
library(ggplot2)
library(MASS)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.3
```

```
## Loading required package: lattice
```

```r
library(DT)
```

```
## Warning: package 'DT' was built under R version 3.4.3
```

```r
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 3.4.3
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.3
```

```
## corrplot 0.84 loaded
```

```r
library(Rmisc)
```

```
## Warning: package 'Rmisc' was built under R version 3.4.3
```

```
## Loading required package: plyr
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:reshape':
##
##     rename, round_any
```

```r
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:reshape':
##
##     rename
```

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

```
## The following objects are masked from 'package:stats':
```

```
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.4.3
##
## Attaching package: 'car'
## The following object is masked from 'package:psych':
##
##     logit
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
htd <- read.csv("C:\\Users\\Alex O\\Downloads\\train.csv")
X <- htd$GrLivArea
Y <- htd$SalePrice
```

I chose GrLivArea: Above grade (ground) living area square feet, as the independent variable and now we create a subset with just X and Y:

```
newhtd <- subset(htd, select = c(GrLivArea, SalePrice))
names(newhtd) <- c("X","Y")
```

Probability

a.$\mathbf{P(X > x \mid Y > y)}$

```
x <- quantile(X, probs = 0.25)
y <- quantile(Y, probs = 0.5)
x
```

```
##    25%
## 1129.5
```

```
y
```

```
##    50%
## 163000
```

```
a <- length(newhtd$X[newhtd$X > x & newhtd$Y > y])/length(newhtd$Y[newhtd$Y > y])
a
```

```
## [1] 0.989011
```

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

```r
b <- length(newhtd$X[newhtd$X > x & newhtd$Y>y]) /
            nrow(newhtd)
b
```

```
## [1] 0.4931507
```

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

```r
c <- length(newhtd$X[newhtd$X < x & newhtd$Y > y]) /
      nrow(newhtd)
c
```

```
## [1] 0.005479452
```

II. Does splitting the training data in this fashion make them independent? In other words, does P(XY)=P(X)P(Y)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.

```r
p_xy <- nrow(subset(htd, htd$GrLivArea > x & htd$SalePrice > y)) / nrow(htd)
p_xy
```

```
## [1] 0.4931507
```

```r
p_x <- nrow(subset(htd, htd$GrLivArea > x)) / nrow(htd)
p_x
```

```
## [1] 0.75
```

```r
p_y <- nrow(subset(htd, htd$SalePrice > y)) / nrow(htd)
p_y
```

```
## [1] 0.4986301
```

```r
p_x*p_y
```

```
## [1] 0.3739726
```

This shows that splitting it in this fashion does not them independent

Evaluate with Chi-Square Test:

```r
tab1 <- table(htd$GrLivArea, htd$SalePrice)
chisq.test(tab1)
```

```
## Warning in chisq.test(tab1): Chi-squared approximation may be incorrect

##
##  Pearson's Chi-squared test
##
## data:  tab1
## X-squared = 589730, df = 569320, p-value < 2.2e-16
```

We reject the null due to this p value - X and Y are not independent.

Descriptive and Inferential Statistics

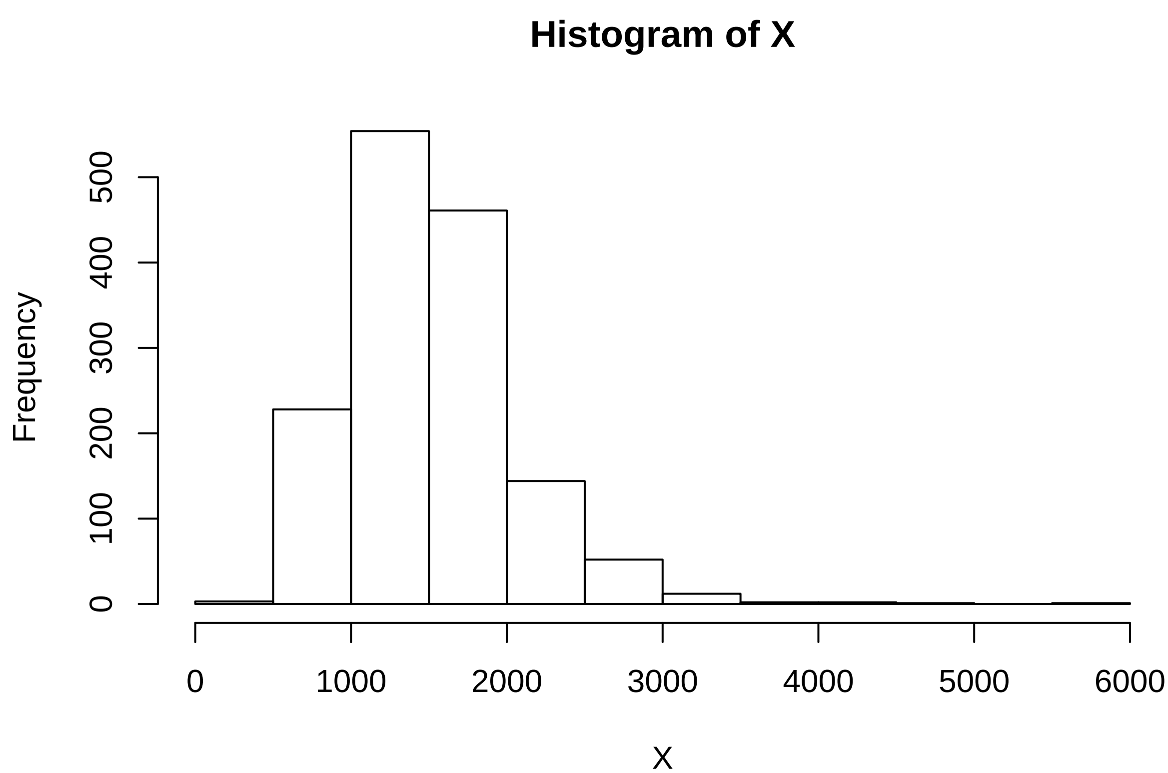
```
summary(X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1130    1464    1515    1777    5642
```

```
describe(X)
```

```
##    vars    n    mean     sd median trimmed    mad min  max range skew
## X1    1 1460 1515.46 525.48   1464 1467.67 483.33 334 5642  5308 1.36
##    kurtosis    se
## X1     4.86 13.75
```

```
hist(X)
```



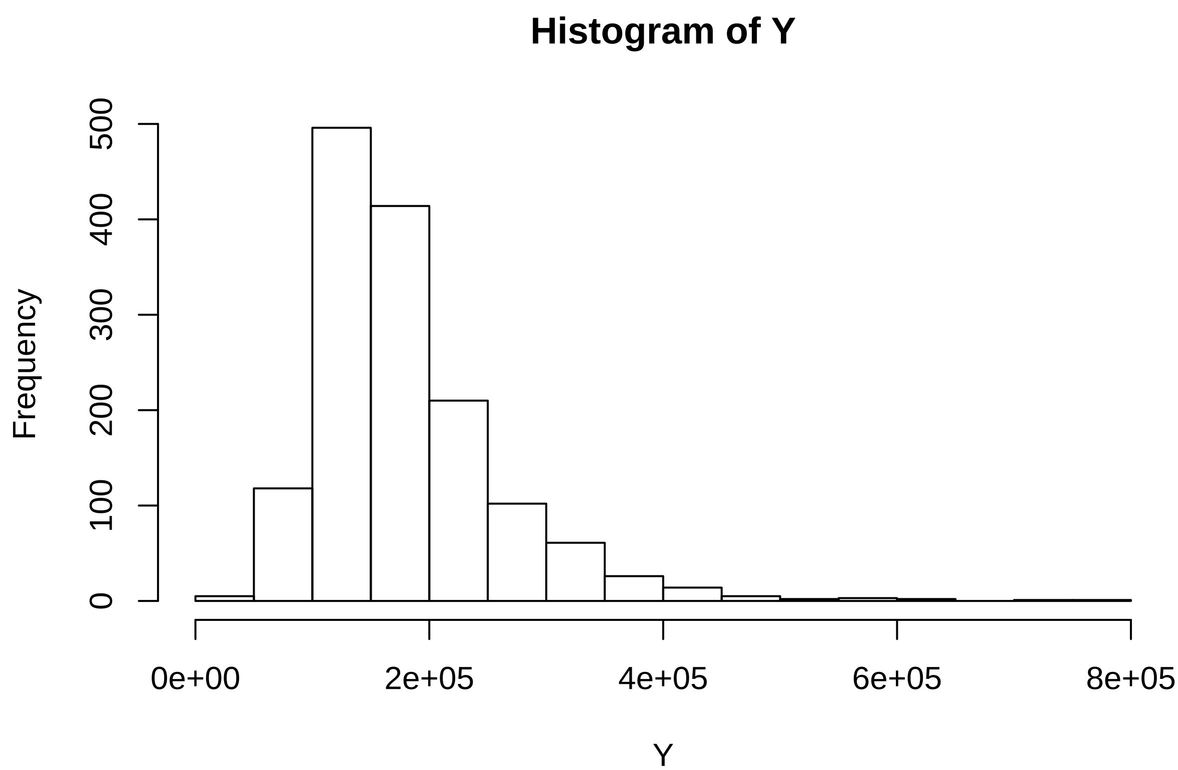

```
summary(Y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  129975  163000  180921  214000  755000
```

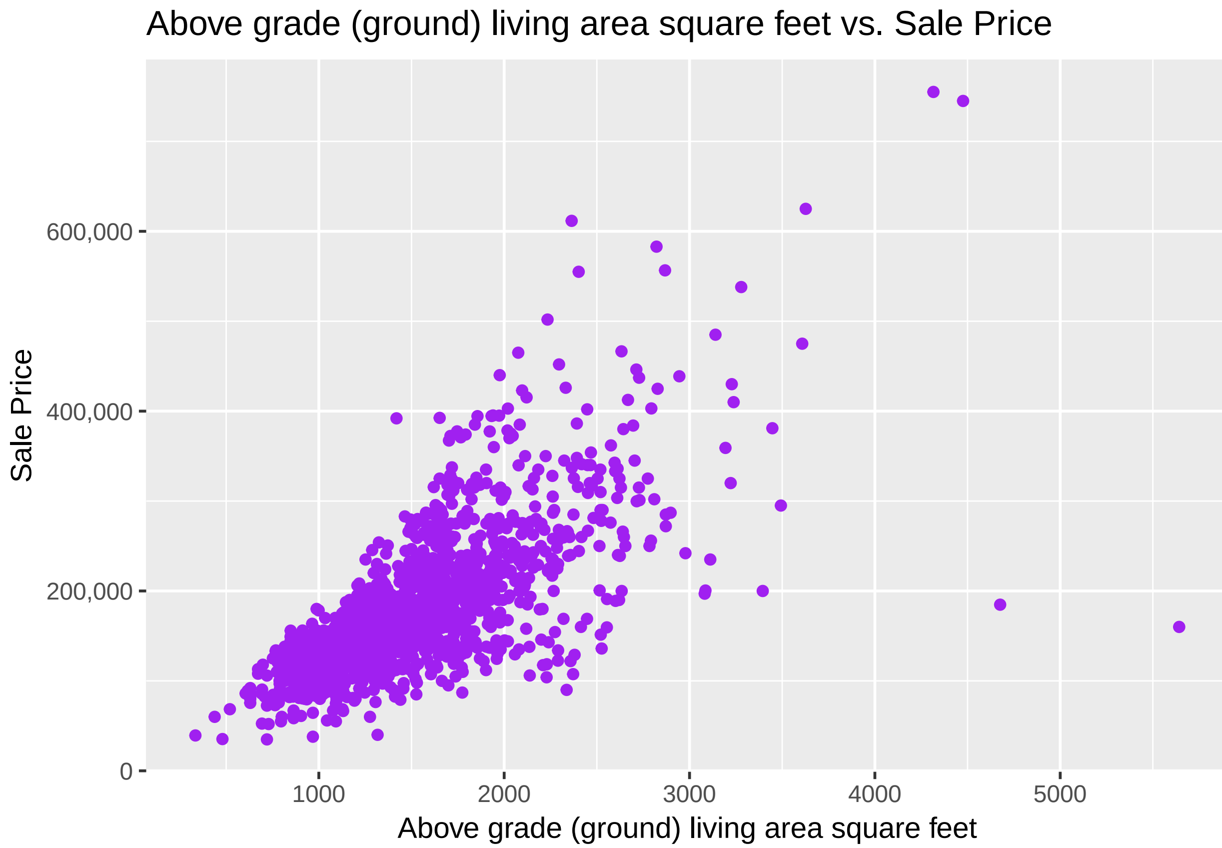
```
describe(Y)
```

```
##    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
```

```r
hist(Y)
```

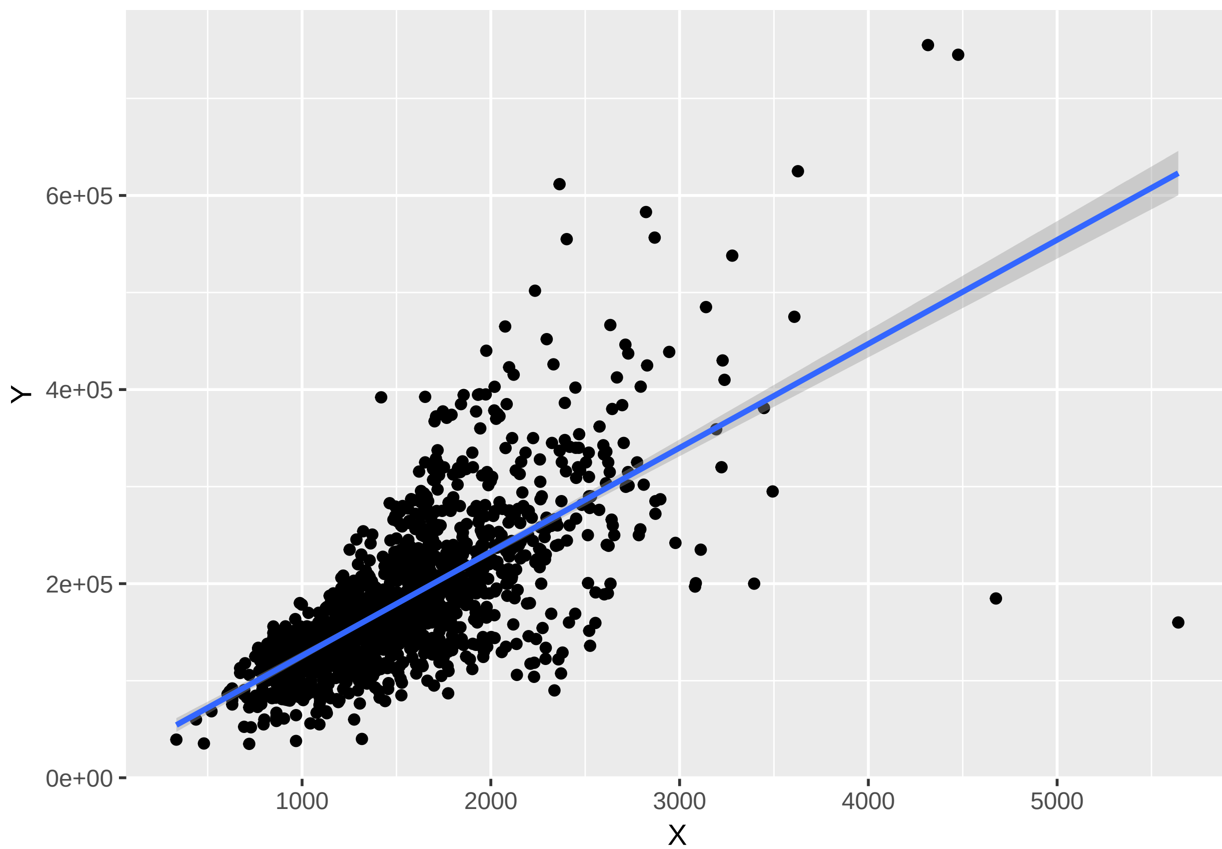



```r
ggplot(htd, aes(x = X, y = Y)) + geom_point(color='purple') + labs(title = "Above grade (ground) living
```



There seems to be a strong correlation

```
qplot(X,Y, data=newhtd) + stat_smooth(method=lm)
```



Box-Cox:

```
summary(powerTransform(cbind(Y,X)~1, data=newhtd))
```

```
## bcPower Transformations to Multinormality
##   Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
## Y   -0.0306           0      -0.1078       0.0467
## X   -0.0172           0      -0.1194       0.0851
##
## Likelihood ratio tests about transformation parameters
##                                 LRT df       pval
## LR test, lambda = (0 0)   0.6334332  2 0.7285372
## LR test, lambda = (1 1) 885.9108958  2 0.0000000
```

The estimated λ values are both close to 0:

```
lnY <- log(Y)
lnX <- log(X)
cor(lnY, lnX)
```

```
## [1] 0.7302549
```

The correlation coefficient shows a strong relationship between the two transformed variables.

Linear Algebra and Correlation

TotalBsmtSF: Total square feet of basement area X1stFlrSF: First Floor square feet LotArea: Total rooms above grade

```
corrmatrix <- cor(htd[c("TotalBsmtSF","X1stFlrSF","LotArea","SalePrice")])
corrmatrix
```

```
##             TotalBsmtSF X1stFlrSF   LotArea SalePrice
## TotalBsmtSF   1.0000000 0.8195300 0.2608331 0.6135806
## X1stFlrSF     0.8195300 1.0000000 0.2994746 0.6058522
## LotArea       0.2608331 0.2994746 1.0000000 0.2638434
## SalePrice     0.6135806 0.6058522 0.2638434 1.0000000
```

```
invcormat <- solve(corrmatrix)
invcormat
```

```
##               TotalBsmtSF  X1stFlrSF       LotArea  SalePrice
## TotalBsmtSF  3.2603189969 -2.3064606 -0.0005917129 -0.6029380
## X1stFlrSF   -2.3064606461  3.2656838 -0.2448004553 -0.4987333
## LotArea     -0.0005917129 -0.2448005  1.1116224106 -0.1446182
## SalePrice   -0.6029379875 -0.4987333 -0.1446182309  1.7102662
```

```
round(corrmatrix %*% invcormat, 15)
```

```
##             TotalBsmtSF X1stFlrSF LotArea SalePrice
## TotalBsmtSF           1         0       0         0
## X1stFlrSF             0         1       0         0
## LotArea               0         0       1         0
## SalePrice             0         0       0         1
```

```
round(invcormat %*% corrmatrix, 15)
```

```
##             TotalBsmtSF X1stFlrSF LotArea SalePrice
## TotalBsmtSF           1         0       0         0
## X1stFlrSF             0         1       0         0
## LotArea               0         0       1         0
## SalePrice             0         0       0         1
```

Calculus-Based Probability & Statistics

```
mylndist <- fitdistr(X, "lognormal")
mylndist$estimate
```

```
##   meanlog     sdlog
## 7.2677744 0.3334362
```

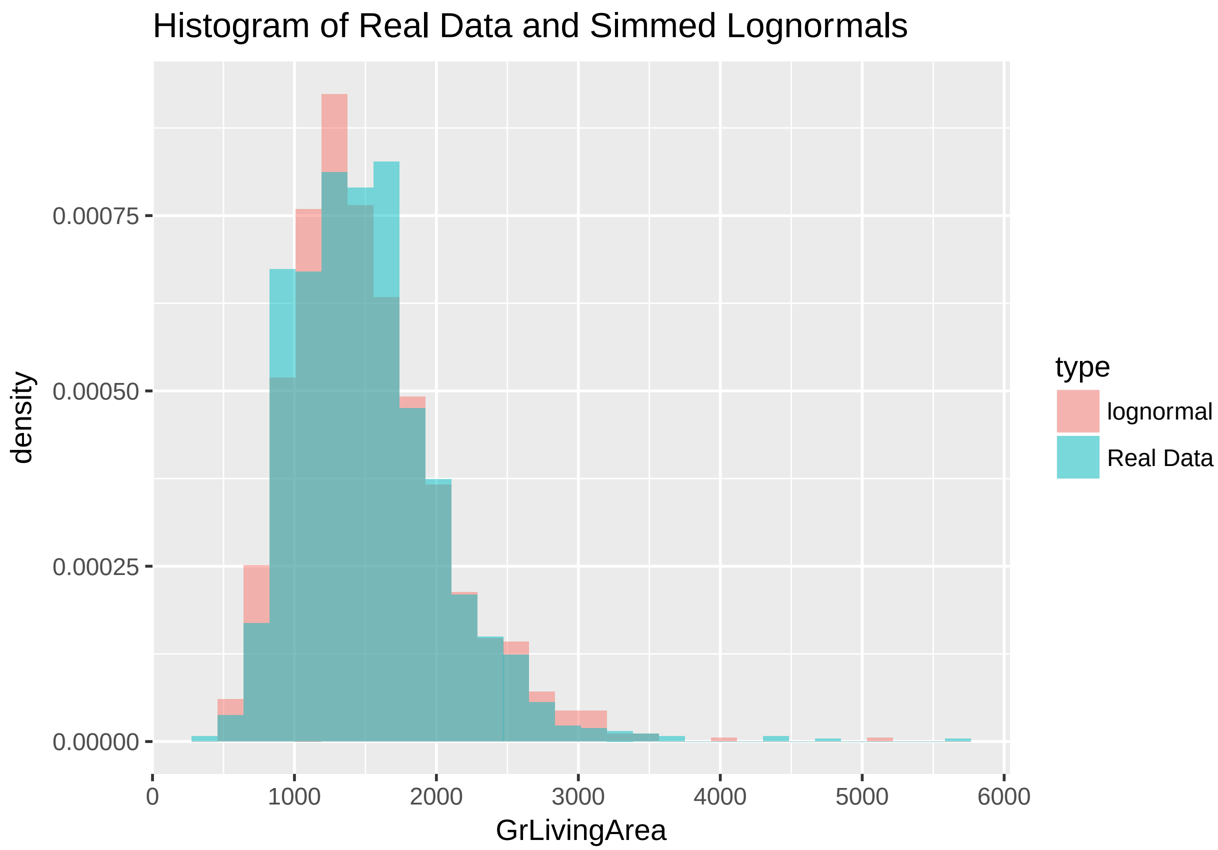
```
mylndist$loglik
```

```
## [1] -11079.08
```

```
set.seed(1)
sample_sel <- rlnorm(n=1000, meanlog = mylndist$estimate["meanlog"], sdlog = mylndist$estimate["sdlog"])
reald <- data.frame(GrLivingArea = X)
selected_fit <- data.frame(GrLivingArea = sample_sel)
```

```
reald$type <- "Real Data"; selected_fit$type <- "lognormal"; mytransd <- rbind(reald, selected_fit)
ggplot(mytransd, aes(x=GrLivingArea,fill = type)) + geom_histogram(alpha = 0.5, aes(y = ..density..), p
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



It is not a perfect fit by any means but we can see that the lognormal provides a more than adequate fit to the raw data.

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.

```
lregmod <- lm(SalePrice ~ LotArea + OverallQual + OverallCond + TotalBsmtSF + X1stFlrSF + GrLivArea + B
# summary of model
summary(lregmod)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + OverallQual + OverallCond +
##     TotalBsmtSF + X1stFlrSF + GrLivArea + BsmtFullBath + FullBath +
##     GarageCars + GarageArea, data = htd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -471722  -18751   -1351   15077  298213
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.300e+05  7.348e+03 -17.698  < 2e-16 ***
## LotArea       4.887e-01  1.058e-01   4.619 4.19e-06 ***
## OverallQual   2.403e+04  1.086e+03  22.136  < 2e-16 ***
## OverallCond   3.627e+03  9.225e+02   3.932 8.83e-05 ***
## TotalBsmtSF   1.948e+01  4.272e+00   4.559 5.57e-06 ***
## X1stFlrSF     8.077e+00  4.888e+00   1.652   0.0987 .
## GrLivArea     4.048e+01  2.895e+00  13.986  < 2e-16 ***
## BsmtFullBath  1.546e+04  2.050e+03   7.538 8.36e-14 ***
## FullBath      6.115e+03  2.508e+03   2.438   0.0149 *
## GarageCars    1.545e+04  2.983e+03   5.180 2.54e-07 ***
## GarageArea    9.001e+00  1.022e+01   0.880   0.3788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37600 on 1449 degrees of freedom
## Multiple R-squared:  0.7776, Adjusted R-squared:  0.776
## F-statistic: 506.5 on 10 and 1449 DF,  p-value: < 2.2e-16
```

```r
testd <- read.csv("C:\\Users\\Alex O\\Downloads\\test.csv")
```

```r
predictor <- predict(lregmod, testd, type="response")
datapredict <- data.frame(Id=names(predictor),SalePrice=predictor)
write.csv(datapredict,"predictor.csv", row.names=FALSE)
```

-Kaggle Score = 1.09 User Name = adcosborne