Data 605 Final Exam

Code

Armenoush Aslanian-Persico

December 2016

Hide

```
library(MASS)
library(knitr)
library(dplyr)
library(ggplot2)
library(DT)
library(reshape)
library(corrplot)
library(Rmisc)
```

Hide

```
df <- read.csv("train.csv")
```

Introduction

Below is the dataset of house prices available from Kaggle.com. The dataset has 1459 observations of houses in Ames, Iowa, and 79 variables potentially contributing to the house sale price.

The full dataset and dictionary are available at: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data)

Hide

```
#kable(head(df))
datatable(df, options = list( pageLength = 5, lengthMenu = c(5, 10, 40),   initComplete = JS(
    "function(settings, json) {",
    "$(this.api().table().header()).css({'background-color': '#01975b', 'color': '#fff'});",
    "}")), rownames=TRUE)
```

Show 5 entries

Search:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig
1	1	60	RL	65	8450	Pave		Reg	Lvl	AllPub	Inside
2	2	20	RL	80	9600	Pave		Reg	Lvl	AllPub	FR2
3	3	60	RL	68	11250	Pave		IR1	Lvl	AllPub	Inside
4	4	70	RL	60	9550	Pave		IR1	Lvl	AllPub	Corner
5	5	60	RL	84	14260	Pave		IR1	Lvl	AllPub	FR2

Showing 1 to 5 of 1,460 entries

Previous 1 2 3 4 5 … 292 Next

Part 1: Variables

Pick one of the quanititative independent variables from the training data set (train.csv) , and define that variable as X. Make sure this variable is skewed to the right! Pick the dependent variable and define it as Y.

Hide

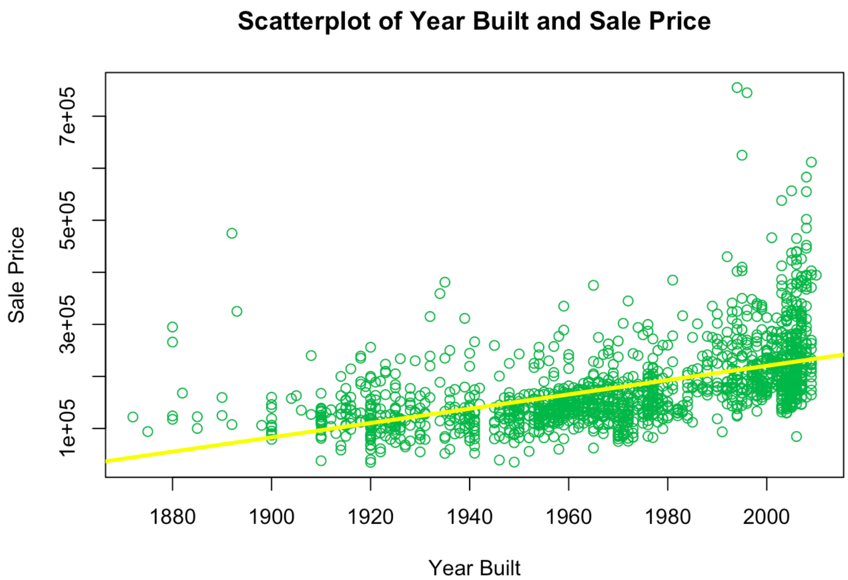
```
#test variable
X1<-df$OverallQual
Y1<-df$SalePrice

plot(X1,Y1)
hist(Y1, col="blue", main="Histogram of Overall Quality")
```

Hide

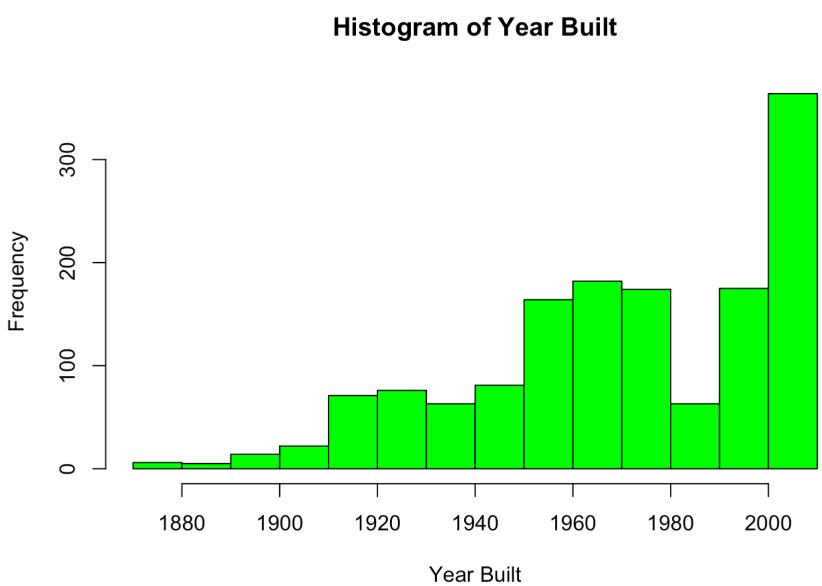
```
#chosen variable
X<-df$YearBuilt
Y<-df$SalePrice

plot(X,Y, col="#4caf50", main="Scatterplot of Year Built and Sale Price", xlab = "Year Built", ylab="Sale Price")
abline(lm(Y~X), col="yellow", lwd=3) # regression line (y~x)
```



Hide

```
hist(X, col="green", main="Histogram of Year Built", xlab = "Year Built")
```



Hide

```
hist(Y, col="#80cbc4", main="Histogram of Sale Price", xlab = "Sale Price")
```

Hide

```
print("Summary of X variable: Year Built")
```

```
## [1] "Summary of X variable: Year Built"
```

Hide

```
summary(X)
```

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

Hide

```
print("Summary of Y variable: Sale Price")
```

```
## [1] "Summary of Y variable: Sale Price"
```

Hide

```
summary(Y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  130000  163000  180900  214000  755000
```

Part 2: Probability

Probability. Calculate as a minimum the below probabilities a through c. Assume the small letter “x” is estimated as the 3d 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. In addition, make a table of counts as shown below.

a. $$p_1 = p(X > x|Y > y)$$

Given an above median sale price, the probability that a house has a year built greater than the third quartile.

Hide

```
XQ3<-quantile(X, probs=0.75) #2000 #3rd quartile of X variable
YQ2<-quantile (Y, probs=0.50) #163000 #2nd quartile, or median, of Y variable

n<-(nrow(df))
yearbuilt<-as.numeric(df$YearBuilt)
saleprice<-as.numeric(df$SalePrice)

nYQ2<-nrow(subset(df,saleprice>YQ2))


p1<-nrow(subset(df, yearbuilt > XQ3 & saleprice>YQ2))/nYQ2
p1
```

```
## [1] 0.4436813
```

b. $$p_2 = p(X > x, Y > y)$$

Given the complete data set, the probability that a house has a year built greater than the third quartile and a sale price above median value.

Hide

```
p2<-nrow(subset(df, yearbuilt > XQ3 & saleprice>YQ2))/n
p2
```

```
## [1] 0.2212329
```

c. $$p_3 = p(X < x | Y > y)$$

Given an above median selling price, the probability that a house has a year built less than [less than or equal to] the third quartile.

Hide

```
p3<-nrow(subset(df, yearbuilt <=XQ3 & saleprice>YQ2))/nYQ2
p3
```

```
## [1] 0.5563187
```

Hide

```
c1<-nrow(subset(df, yearbuilt <=XQ3 & saleprice<=YQ2))/n
c2<-nrow(subset(df, yearbuilt <=XQ3 & saleprice>YQ2))/n
c3<-c1+c2
c4<-nrow(subset(df, yearbuilt >XQ3 & saleprice<=YQ2))/n
c5<-nrow(subset(df, yearbuilt >XQ3 & saleprice>YQ2))/n
c6<-c4+c5
c7<-c1+c4
c8<-c2+c5
c9<-c3+c6
```

Hide

```
dfcounts<-matrix(round(c(c1,c2,c3,c4,c5,c6,c7,c8,c9),3), ncol=3, nrow=3, byrow=TRUE)
colnames(dfcounts)<-c(
"<=2d quartile",
">2d quartile",
"Total")
rownames(dfcounts)<-c("<=3rd quartile",">3rd quartile","Total")

print("Quartile Matrix by Percentage")
```

```
## [1] "Quartile Matrix by Percentage"
```

Hide

```
dfcounts<-as.table(dfcounts)
dfcounts
```

```
##                <=2d quartile >2d quartile Total
## <=3rd quartile         0.473        0.277 0.751
## >3rd quartile          0.028        0.221 0.249
## Total                  0.501        0.499 1.000
```

Hide

```
print("Quartile Matrix by Count")
```

```
## [1] "Quartile Matrix by Count"
```

Hide

```
dfvals<-round(dfcounts*1460,0)
dfvals
```

```
##                <=2d quartile >2d quartile Total
## <=3rd quartile           691          404  1096
## >3rd quartile             41          323   364
## Total                    731          729  1460
```

Part 3: Independence

Does splitting the training data in this fashion make them independent? Let A be the new variable counting those observations above the 3d quartile for X, and let B be the new variable counting those observations for the 2d quartile for Y. Does P(A|B)=P(A)P(B)? Check mathematically, and then evaluate by running a Chi Square test for association.

Hide

```
papb<-c4*c5
print (paste0("p(A)*p(B)=", round(papb,5)))
```

```
## [1] "p(A)*p(B)=0.00621"
```

$$p(A|B) = p(X > x|Y > y) = 0.444$$

$$p(A) * p(B) = 0.006$$

$$p(A|B)! = p(A) * p(B)$$

Hide

```
mat <- matrix(c(691, 404, 41, 323), 2, 2, byrow=T)

chisq.test(mat, correct=TRUE)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  mat
## X-squared = 291.61, df = 1, p-value < 2.2e-16
```

Hide

```
#test of alternate chi sq approach
A<-subset(df, df$YearBuilt>XQ3)
B<-subset(df, df$SalePrice>YQ2)
chisq.test(A, B) #issue with variable class
```

Part 4: Statistics

Provide univariate descriptive statistics and appropriate plots for the training data set. Provide a scatterplot of X and Y.

> Also see Part 1.

Hide

```
isnum <- sapply(df, is.numeric)
dfnum<-df[ , isnum]
summary(dfnum)
```

```
##        Id           MSSubClass     LotFrontage       LotArea
##  Min.   :   1.0   Min.   : 20.0   Min.   : 21.00   Min.   :  1300
##  1st Qu.: 365.8   1st Qu.: 20.0   1st Qu.: 59.00   1st Qu.:  7554
##  Median : 730.5   Median : 50.0   Median : 69.00   Median :  9478
##  Mean   : 730.5   Mean   : 56.9   Mean   : 70.05   Mean   : 10517
##  3rd Qu.:1095.2   3rd Qu.: 70.0   3rd Qu.: 80.00   3rd Qu.: 11602
##  Max.   :1460.0   Max.   :190.0   Max.   :313.00   Max.   :215245
##                                   NA's   :259
##   OverallQual      OverallCond      YearBuilt     YearRemodAdd
##  Min.   : 1.000   Min.   :1.000   Min.   :1872   Min.   :1950
##  1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954   1st Qu.:1967
##  Median : 6.000   Median :5.000   Median :1973   Median :1994
##  Mean   : 6.099   Mean   :5.575   Mean   :1971   Mean   :1985
##  3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000   3rd Qu.:2004
##  Max.   :10.000   Max.   :9.000   Max.   :2010   Max.   :2010
##
##    MasVnrArea       BsmtFinSF1       BsmtFinSF2        BsmtUnfSF
##  Min.   :   0.0   Min.   :   0.0   Min.   :   0.00   Min.   :   0.0
##  1st Qu.:   0.0   1st Qu.:   0.0   1st Qu.:   0.00   1st Qu.: 223.0
##  Median :   0.0   Median : 383.5   Median :   0.00   Median : 477.5
##  Mean   : 103.7   Mean   : 443.6   Mean   :  46.55   Mean   : 567.2
##  3rd Qu.: 166.0   3rd Qu.: 712.2   3rd Qu.:   0.00   3rd Qu.: 808.0
##  Max.   :1600.0   Max.   :5644.0   Max.   :1474.00   Max.   :2336.0
##  NA's   :8
##   TotalBsmtSF       X1stFlrSF      X2ndFlrSF     LowQualFinSF
##  Min.   :   0.0   Min.   : 334   Min.   :   0   Min.   :  0.000
##  1st Qu.: 795.8   1st Qu.: 882   1st Qu.:   0   1st Qu.:  0.000
##  Median : 991.5   Median :1087   Median :   0   Median :  0.000
##  Mean   :1057.4   Mean   :1163   Mean   : 347   Mean   :  5.845
##  3rd Qu.:1298.2   3rd Qu.:1391   3rd Qu.: 728   3rd Qu.:  0.000
##  Max.   :6110.0   Max.   :4692   Max.   :2065   Max.   :572.000
##
##    GrLivArea     BsmtFullBath     BsmtHalfBath        FullBath
##  Min.   : 334   Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:1130   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.000
##  Median :1464   Median :0.0000   Median :0.00000   Median :2.000
##  Mean   :1515   Mean   :0.4253   Mean   :0.05753   Mean   :1.565
##  3rd Qu.:1777   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:2.000
##  Max.   :5642   Max.   :3.0000   Max.   :2.00000   Max.   :3.000
##
##     HalfBath       BedroomAbvGr    KitchenAbvGr    TotRmsAbvGrd
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   : 2.000
##  1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.: 5.000
##  Median :0.0000   Median :3.000   Median :1.000   Median : 6.000
##  Mean   :0.3829   Mean   :2.866   Mean   :1.047   Mean   : 6.518
##  3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.: 7.000
##  Max.   :2.0000   Max.   :8.000   Max.   :3.000   Max.   :14.000
##
##    Fireplaces     GarageYrBlt     GarageCars      GarageArea
##  Min.   :0.000   Min.   :1900   Min.   :0.000   Min.   :   0.0
##  1st Qu.:0.000   1st Qu.:1961   1st Qu.:1.000   1st Qu.: 334.5
##  Median :1.000   Median :1980   Median :2.000   Median : 480.0
##  Mean   :0.613   Mean   :1979   Mean   :1.767   Mean   : 473.0
##  3rd Qu.:1.000   3rd Qu.:2002   3rd Qu.:2.000   3rd Qu.: 576.0
##  Max.   :3.000   Max.   :2010   Max.   :4.000   Max.   :1418.0
##                  NA's   :81
##    WoodDeckSF      OpenPorchSF     EnclosedPorch      X3SsnPorch
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00
##  Median :  0.00   Median : 25.00   Median :  0.00   Median :  0.00
##  Mean   : 94.24   Mean   : 46.66   Mean   : 21.95   Mean   :  3.41
##  3rd Qu.:168.00   3rd Qu.: 68.00   3rd Qu.:  0.00   3rd Qu.:  0.00
##  Max.   :857.00   Max.   :547.00   Max.   :552.00   Max.   :508.00
##
##   ScreenPorch        PoolArea          MiscVal             MoSold
##  Min.   :  0.00   Min.   :  0.000   Min.   :    0.00   Min.   : 1.000
##  1st Qu.:  0.00   1st Qu.:  0.000   1st Qu.:    0.00   1st Qu.: 5.000
##  Median :  0.00   Median :  0.000   Median :    0.00   Median : 6.000
##  Mean   : 15.06   Mean   :  2.759   Mean   :   43.49   Mean   : 6.322
##  3rd Qu.:  0.00   3rd Qu.:  0.000   3rd Qu.:    0.00   3rd Qu.: 8.000
##  Max.   :480.00   Max.   :738.000   Max.   :15500.00   Max.   :12.000
##
##      YrSold       SalePrice
##  Min.   :2006   Min.   : 34900
##  1st Qu.:2007   1st Qu.:129975
##  Median :2008   Median :163000
##  Mean   :2008   Mean   :180921
##  3rd Qu.:2009   3rd Qu.:214000
##  Max.   :2010   Max.   :755000
##
```

Confidence interval

Provide a 95% CI for the difference in the mean of the variables.

Hide

```
#t.test(x,y)
t.test(df$YearBuilt, df$SalePrice)
```

```
##
##  Welch Two Sample t-test
##
## data:  df$YearBuilt and df$SalePrice
## t = -86.071, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -183028.3 -174871.6
## sample estimates:
##  mean of x  mean of y
##   1971.268 180921.196
```

Selective correlation matrix for chosen variables

Derive a correlation matrix for two of the quantitative variables you selected.

Test the hypothesis that the correlation between these variables is 0 and provide a 99% confidence interval. Discuss the meaning of your analysis.

Hide

```
myvars<-data.frame(df$YearBuilt, df$SalePrice)
#head(myvars) #view header
cor(myvars)
```

```
##              df.YearBuilt df.SalePrice
## df.YearBuilt    1.0000000    0.5228973
## df.SalePrice    0.5228973    1.0000000
```

Hide

```
cor.test(df$YearBuilt, df$SalePrice, conf.level = 0.99)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$YearBuilt and df$SalePrice
## t = 23.424, df = 1458, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 99 percent confidence interval:
##  0.4721529 0.5701878
## sample estimates:
##       cor
## 0.5228973
```

Hide

```
t.test(df$YearBuilt, df$SalePrice, conf.level = 0.99)
```

```
##
##  Welch Two Sample t-test
##
## data:  df$YearBuilt and df$SalePrice
## t = -86.071, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 99 percent confidence interval:
##  -184312.4 -173587.5
## sample estimates:
##  mean of x  mean of y
##   1971.268 180921.196
```

Hide

```
mymx<-as.matrix(cor(myvars))
```

With a 99 percent confidence level, the correlation between Year Built and Sale Price is estimated to be between 0.47 and 0.57.

Part 5: Correlation

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.

Correlation Matrix, Precision Matrix, Identity Matrix

Hide

```
#my correlation matrix
mymx
```

```
##              df.YearBuilt df.SalePrice
## df.YearBuilt    1.0000000    0.5228973
## df.SalePrice    0.5228973    1.0000000
```

Hide

```
#inverse of my correlation matrix, precision matrix
ginvmymx<-ginv(mymx)
ginvmymx
```

```
##            [,1]       [,2]
## [1,]  1.3763140 -0.7196709
## [2,] -0.7196709  1.3763140
```

Hide

```
#corr mat x precision mat
mymxginv<-mymx%*%ginvmymx
round(mymxginv,2)
```

```
##              [,1] [,2]
## df.YearBuilt    1    0
## df.SalePrice    0    1
```

Hide

```
#precision mat x corr mat
ginvmymx<-ginvmymx%*%mymx
round(ginvmymx,2)
```

```
##      df.YearBuilt df.SalePrice
## [1,]            1            0
## [2,]            0            1
```

Principal Components Analysis

Conduct principal components analysis (research this!) and interpret. Discuss.

Header of all quantitative variables

Hide

```
#Correlation matrix of all quantitative variables in dataframe

kable(head(dfnum))
```

Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmt
1	60	65	8450	7	5	2003	2003	196	706	0	150	8
2	20	80	9600	6	8	1976	1976	0	978	0	284	12
3	60	68	11250	7	5	2001	2002	162	486	0	434	9
4	70	60	9550	7	5	1915	1970	0	216	0	540	7
5	60	84	14260	8	5	2000	2000	350	655	0	490	11
6	50	85	14115	5	5	1993	1995	0	732	0	64	7

Header of correlation matrix for all quantitative variables

Hide

```
cormatrix<-cor(dfnum)
cordf<-as.data.frame(cormatrix)
kable(head(cordf))
```

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinS
Id	1.0000000	0.0111565	NA	-0.0332255	-0.0283648	0.0126089	-0.0127127	-0.0219976	NA	-0.0050240	-0.0059
MSSubClass	0.0111565	1.0000000	NA	-0.1397811	0.0326277	-0.0593158	0.0278501	0.0405810	NA	-0.0698357	-0.0656
LotFrontage	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	
LotArea	-0.0332255	-0.1397811	NA	1.0000000	0.1058057	-0.0056363	0.0142277	0.0137884	NA	0.2141031	0.1111
OverallQual	-0.0283648	0.0326277	NA	0.1058057	1.0000000	-0.0919323	0.5723228	0.5506839	NA	0.2396660	-0.0591
OverallCond	0.0126089	-0.0593158	NA	-0.0056363	-0.0919323	1.0000000	-0.3759832	0.0737415	NA	-0.0462309	0.0402

Header of variables with correlation greater than 0.5

Hide

```
#Source from http://stackoverflow.com/questions/7074246/show-correlations-as-an-ordered-list-not-as-a-large-matrix

cordf[cordf == 1] <- NA #drop correlation of 1, diagonals
cordf[abs(cordf) < 0.5] <- NA # drop correlations of less than 0.5
cordf<-as.matrix(cordf)
cordf2<- na.omit(melt(cordf))
kable(head(cordf2[order(-abs(cordf2$value)),])) # sort by highest correlations
```

	X1	X2	value
1016	GarageArea	GarageCars	0.8824754
1053	GarageCars	GarageArea	0.8824754
632	TotRmsAbvGrd	GrLivArea	0.8254894
891	GrLivArea	TotRmsAbvGrd	0.8254894
470	X1stFlrSF	TotalBsmtSF	0.8195300
507	TotalBsmtSF	X1stFlrSF	0.8195300

Hide

```
#corrplot(cordf, type = "upper", tl.col = "black", tl.srt = 45)
```

Hide

```
#test of alternate corr approach
myvars<-data.frame(df$YearBuilt, df$SalePrice)
head(myvars)
```

All variables with correlation to Sale Price greater than 0.5

Hide

```
cordf2<-as.data.frame(cordf2)
# head(cordf2) #view head
# str(cordf2) #view structure
topcors <- cordf2[ which(cordf2$X2=='SalePrice'),]

topcorsdf<-topcors[order(-abs(topcors$value)),]# sort by highest correlations
```

Hide

```
cors1<-data.frame(topcorsdf$X1,topcorsdf$X2,topcorsdf$value)
kable(cors1)
```

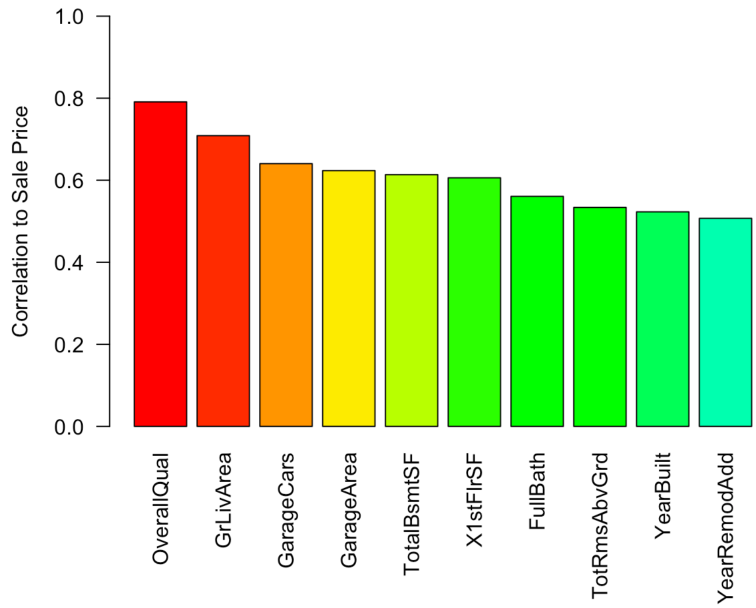
topcorsdf.X1	topcorsdf.X2	topcorsdf.value
OverallQual	SalePrice	0.7909816
GrLivArea	SalePrice	0.7086245
GarageCars	SalePrice	0.6404092
GarageArea	SalePrice	0.6234314
TotalBsmtSF	SalePrice	0.6135806
X1stFlrSF	SalePrice	0.6058522
FullBath	SalePrice	0.5606638
TotRmsAbvGrd	SalePrice	0.5337232

topcorsdf.X1	topcorsdf.X2	topcorsdf.value
YearBuilt	SalePrice	0.5228973
YearRemodAdd	SalePrice	0.5071010

Plot of correlation to Sale Price

Hide

```
par(mar=c(8,8,1,1))
barplot(topcorsdf$value, ylab="Correlation to Sale Price", ylim=c(0,1), col=rainbow(20), las=2, names.arg=topcorsdf$X1)
```



Variables with strongest correlation to Sale Price in descending order:

- OverallQual
- GrLivArea
- GarageCars
- GarageArea
- TotalBsmtSF
- X1stFlrSF
- FullBath
- TotRmsAbvGrd
- YearBuilt
- YearRemodAdd

Hide

```
cormatdata <- select(df, OverallQual, GrLivArea, GarageCars, GarageArea, TotalBsmtSF, X1stFlrSF, FullBath, TotRmsAbvGrd)
```

```
## Warning in combine_vars(vars, ind_list): '.Random.seed' is not an integer
## vector but of type 'NULL', so ignored
```

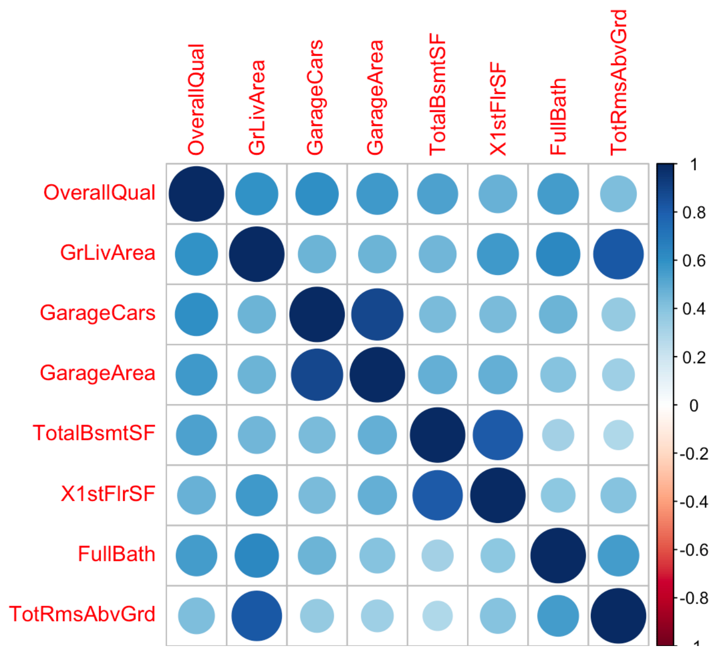
Hide

```
cormat1 <- cor(cormatdata)
cormat1
```

```
##              OverallQual GrLivArea GarageCars GarageArea TotalBsmtSF
## OverallQual    1.0000000 0.5930074  0.6006707  0.5620218   0.5378085
## GrLivArea      0.5930074 1.0000000  0.4672474  0.4689975   0.4548682
## GarageCars     0.6006707 0.4672474  1.0000000  0.8824754   0.4345848
## GarageArea     0.5620218 0.4689975  0.8824754  1.0000000   0.4866655
## TotalBsmtSF    0.5378085 0.4548682  0.4345848  0.4866655   1.0000000
## X1stFlrSF      0.4762238 0.5660240  0.4393168  0.4897817   0.8195300
## FullBath       0.5505997 0.6300116  0.4696720  0.4056562   0.3237224
## TotRmsAbvGrd   0.4274523 0.8254894  0.3622886  0.3378221   0.2855726
##              X1stFlrSF  FullBath TotRmsAbvGrd
## OverallQual  0.4762238 0.5505997    0.4274523
## GrLivArea    0.5660240 0.6300116    0.8254894
## GarageCars   0.4393168 0.4696720    0.3622886
## GarageArea   0.4897817 0.4056562    0.3378221
## TotalBsmtSF  0.8195300 0.3237224    0.2855726
## X1stFlrSF    1.0000000 0.3806375    0.4095160
## FullBath     0.3806375 1.0000000    0.5547843
## TotRmsAbvGrd 0.4095160 0.5547843    1.0000000
```

Hide

```
corrplot(cormat1, method="circle")
```



Part 6: Sampling

Calculus-Based Probability & Statistics. Many times, it makes sense to fit a closed form distribution to data.

For your variable that is skewed to the right, shift it so that the minimum value is above zero. 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 (https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html)).

Minimum value is above zero

Hide

```
#check that min val is not 0
min(df$YearBuilt)
```

```
## [1] 1872
```

Run fitdistr to fit an exponential probability density function.

Hide

```
fit <- fitdistr(df$YearBuilt, "exponential")
```

Find the optimal value of λ for this distribution, and then take 1000 samples from this exponential distribution using this value (e.g., rexp(1000, λ)).

Hide

```
#optimal value of λ for this distribution

lambda <- fit$estimate
sampledf <- rexp(1000, lambda)
lambda
```

```
##         rate
## 0.0005072877
```

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

Hide

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

sampledf<-data.frame(as.numeric(sampledf))
colnames(sampledf)[1] <- "sample"
str(sampledf)
```

```
## 'data.frame':    1000 obs. of  1 variable:
##  $ sample: num  1253 2534 6874 6833 43724 ...
```

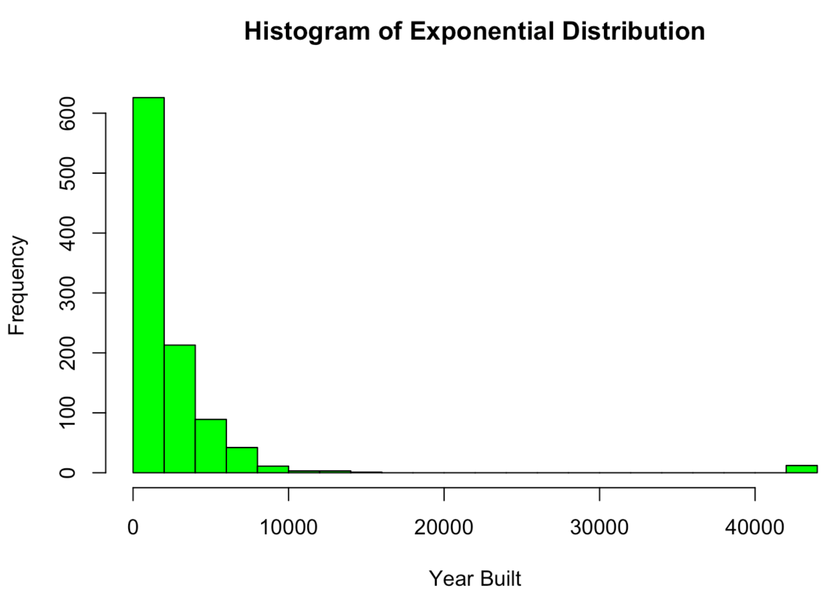
Hide

```
head(sampledf)
```

```
##       sample
## 1  1252.9546
## 2  2534.0248
## 3  6874.1797
## 4  6833.2280
## 5 43724.1191
## 6   275.5495
```

Hide

```
hist(sampledf$sample, col="green", main="Histogram of Exponential Distribution", xlab = "Year Built", breaks=30)
```



Using the exponential pdf, find the 5th and 95th percentiles using the cumulative distribution function (CDF).

Hide

```
#find the 5th and 95th percentiles
print("5th percentile")
```

```
## [1] "5th percentile"
```

Hide

```
qexp(.05,rate = lambda)
```

```
## [1] 101.1128
```

Hide

```
print("95th percentile")
```

```
## [1] "95th percentile"
```

Hide

```
qexp(.95, rate = lambda)
```

```
## [1] 5905.391
```

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

Hide

```
#95% confidence interval from the empirical data
CI(df$YearBuilt, 0.95)
```

```
##    upper     mean    lower
## 1972.818 1971.268 1969.717
```

Finally, provide the empirical 5th percentile and 95th percentile of the data. Discuss.

Hide

```
quantile(df$YearBuilt, .05)
```

```
##   5%
## 1916
```

Hide

```
quantile(df$YearBuilt, .95)
```

```
##  95%
## 2007
```

Part 7: Modeling

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.

Test Model 1: AIC in a Stepwise Algorithm

Hide

```
#test of alternate model
modvars <- df[, which(sapply(df, function(x) sum(is.na(x))) == 0)]
model1 <- step(lm(df$SalePrice ~ ., modvars), direction = 'backward', trace = FALSE)
model1

#dfglm <- glm(df$SalePrice ~ ., family=binomial, data = df)
#dfstep <- stepAIC(dfglm, trace = FALSE)
#dfstep$anova
```

Test Model 2: Multiple Linear Regression

Hide

```
fit <- lm(df$SalePrice ~ df$OverallQual + df$GrLivArea + df$GarageCars + df$GarageArea, data=df)
summary(fit) # show results
```

```
##
## Call:
## lm(formula = df$SalePrice ~ df$OverallQual + df$GrLivArea + df$GarageCars +
##     df$GarageArea, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -372594  -21236   -1594   18625  301129
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -98436.050   4820.467 -20.420  < 2e-16 ***
## df$OverallQual  26988.854   1067.393  25.285  < 2e-16 ***
## df$GrLivArea       49.573      2.555  19.402  < 2e-16 ***
## df$GarageCars   11317.522   3126.297   3.620 0.000305 ***
## df$GarageArea      41.478     10.627   3.903 9.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40420 on 1455 degrees of freedom
## Multiple R-squared:  0.7418, Adjusted R-squared:  0.7411
## F-statistic:  1045 on 4 and 1455 DF,  p-value: < 2.2e-16
```

Using intercepts from regression summary, create multiple linear regression model.

$$SalePrice = 26988.854 * OverallQual + 49.573 * GrLivArea + 11317.522 * GarageCars + 41.478 * GarageArea - 98436.050$$

Hide

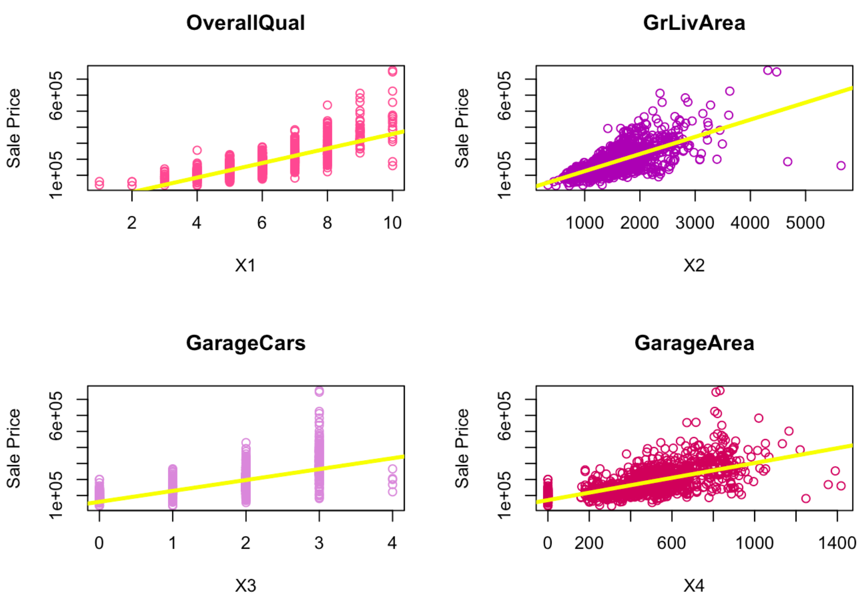
```
par(mfrow=c(2,2))
X1<-df$OverallQual
X2<-df$GrLivArea
X3<-df$GarageCars
X4<-df$GarageArea
Y<-df$SalePrice

plot(X1,Y, col="#f06292", main="OverallQual", ylab="Sale Price")
abline(lm(Y~X1), col="yellow", lwd=3) # regression line (y~x)

plot(X2,Y, col="#9c27b0", main="GrLivArea", ylab="Sale Price")
abline(lm(Y~X2), col="yellow", lwd=3) # regression line (y~x)

plot(X3,Y, col="#ce93d8", main="GarageCars", ylab="Sale Price")
abline(lm(Y~X3), col="yellow", lwd=3) # regression line (y~x)

plot(X4,Y, col="#c2185b", main="GarageArea", ylab="Sale Price")
abline(lm(Y~X4), col="yellow", lwd=3) # regression line (y~x)
```



Load test data set and create calculated column using equation for multiple linear regression. Select required columns and export to csv for contest entry.

Hide

```
dftest <- read.csv("test.csv")
#str(dftest)
#nrow(dftest)

SalePrice<-((26988.854*df$OverallQual) + (49.573*df$GrLivArea) +  (11317.522*df$GarageCars) + (41.478*df$GarageArea)
-98436.050)
#head(SalePrice)

dftest<-dftest[,c("Id","OverallQual","GrLivArea","GarageCars","GarageArea")]

kable(head(dftest))
```

Id	OverallQual	GrLivArea	GarageCars	GarageArea
1461	5	896	1	730
1462	6	1329	1	312
1463	5	1629	2	482
1464	6	1604	2	470
1465	8	1280	2	506
1466	6	1655	2	440

Hide

```
#tail(dftest)

submission <- cbind(dftest$Id,SalePrice)
```

```
## Warning in cbind(dftest$Id, SalePrice): number of rows of result is not a
## multiple of vector length (arg 1)
```

Hide

```
colnames(submission)[1] <- "Id"
submission[submission<0] <- median(SalePrice) #clear negatives due to missing values
submission<-as.data.frame(submission[1:1459,])
kable(head(submission))
```

Id	SalePrice
1461	220620.7
1462	167773.1
1463	226877.0
1464	236184.2
1465	295064.4
1466	146571.1

Hide

```
#str(submission)
#dim(submission)
```

Export CSV and submit to Kaggle.

Eval set to FALSE for reader convenience.

Hide

```
write.csv(submission, file = "submissionAAP.csv", quote=FALSE, row.names=FALSE)
```

Kaggle score: 0.60114