

Final Project

Jonathan Hernandez

December 5, 2018

Problem 1

let $X = X_2$ and $Y = Y_2$ that is

```
X <- c(7.4, 6.4, 8.5, 9.5, 11.8, 8.8, 8.4, 5.1, 11.4, 15.1, 12.6, 8.0, 10.3, 10.4,
      9.5, 9.5, 15.1, 6.6, 15.4, 8.2)

Y <- c(20.8, 14.6, 18.0, 7.3, 19.4, 13.5, 14.7, 15.3, 12.6, 13.0, 13.1, 10.3, 14.9,
      14.8, 16.2, 15.7, 16.3, 11.5, 12.2, 11.8)
```

a) $P(X > x|Y > y)$ where x and y are the 3rd quartile and 1st quartile of x and y respectively.

- First find the 1st quartile of Y and $P(Y > y)$

```
x_3q <- quantile(X,.75)
y_1q <- quantile(Y,.25)
c(x_3q, y_1q)
```

```
## 75% 25%
## 11.5 12.5
```

```
p_ge_y <- length(Y[Y > y_1q]) / length(Y) #  $P(Y > y)$ 
p_ge_y
```

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

- $P(X > x|Y > y) = P(X > x \cap Y > y)/P(Y > y)$ the numerator is the probability that both X and Y are above their respective quartiles.
- We see that $P(Y > y) = 0.75$ from above and using the `intersect()` function we can see how many values both operators in the intersection have in common:

```
x <- X[X > x_3q]
y <- Y[Y > y_1q]
p_x_and_y <- intersect(x, y)
p_x_and_y
```

```
## [1] 12.6
```

- only one value of out 20 (value of 12.6) so $P(X > x \cap Y > y) = 1/20$
- Finally computing the conditional probability gives $P(X > x|Y > y) =$

```
(length(p_x_and_y)/20) / p_ge_y
```

```
## [1] 0.06666667
```

b) $P(X > x, Y > y)$ this is the joint probability or the intersection

- This was calculated in a) and was denoted as $1/20$

c) $P(X < x|Y > y)$ that is what is $P(X < x)$ given $P(Y > y)$

- $P(X < x|Y > y) = P(X < x \cap Y > y)/P(Y > y)$ we found $P(Y > y) = 0.75$ earlier, now let's find the numerator.

```
x <- X[X < x_3q]
p_x_and_y <- intersect(x, y)
p_x_and_y
```

```
## numeric(0)
```

- Thus $P(X > x, Y > y) = 0$
- In addition, make a table of counts as shown below:

```
##
## | x/y      | <=3rd quartile | > 3rd quartile | Total |
## |-----|-----|-----|-----|
## | <=1st   |                 |                 |       |
## | quartile|                 |                 |       |
## |-----|-----|-----|-----|
## | > 1st   |                 |                 |       |
## | quartile|                 |                 |       |
## |-----|-----|-----|-----|
## | Total   |                 |                 |       |
## |-----|-----|-----|-----|
```

- For this we compute the joint probabilities for each of the 4 boxes and add them up

```
x_1q <- quantile(X,.25)
y_3q <- quantile(Y,.75)

x1 <- X[X <= x_1q]
x2 <- X[X > x_1q]

y1 <- Y[Y <= y_3q]
y2 <- Y[Y > y_3q]

p_leq_x_leq_y <- intersect(x1, y1) # P(X <= 1st quartile, Y <= 3rd quartile)
p_leq_x_ge_y <- intersect(x1, y2) # P(X <= 1st quartile, Y > 3rd quartile)
p_ge_x_leq_y <- intersect(x2,y1) # P(X > 1st quartile, Y <= 3rd quartile)
p_ge_x_ge_y <- intersect(x2,y2) # P(X > 1st quartile, Y > 3rd quartile)

p_leq_x_leq_y
```

```
## numeric(0)
```

```
p_leq_x_ge_y
```

```
## numeric(0)
```

```
p_ge_x_leq_y
```

```
## [1] 11.8 12.6 10.3
```

```
p_ge_x_ge_y
```

```
## numeric(0)
```

- Populating the table gives

x/y	<=3rd quartile	> 3rd quartile	Total
<=1st quartile	0	0	0
> 1st quartile	3	0	3
Total	3	0	3

- Does splitting the training data in this fashion make them independent? Let A be the new variable counting those observations above the 1st quartile for X, and let B be the new variable counting those observations above the 1st quartile for Y. Does $P(AB) = P(A)P(B)$? Check mathematically, and then evaluate by running a Chi Square test for association

```
c(x_1q,y_1q) # 1st quartiles of X and Y
```

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

```
## 8.15 12.50
```

```
p_A <- length(X[X > x_1q]) / length(X)
```

```
p_B <- length(Y[Y > y_1q]) / length(Y)
```

```
p_AB <- length(intersect(X[X > x_1q], Y[Y > y_1q])) / 20
```

```
p_AB
```

```
## [1] 0.05
```

```
p_A * p_B
```

```
## [1] 0.5625
```

```
p_AB == (p_A * p_B) #  $P(A)P(B)$ 
```

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

- We see that the variables are not independent by looking the values and equality above.
- Now using a Chi Squared test to test

```
dat <- data.frame(X,Y)
chisq <- chisq.test(dat)
chisq
```

```
##
## Pearson's Chi-squared test
##
## data:  dat
## X-squared = 15.213, df = 19, p-value = 0.709
```

- Using the chi squared test, we see that the p-value is about 0.7. This means that the variables X and Y are not stastically significantly associated.

Problem 2

- 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.
- 5 points. Descriptive and Inferential Statistics. Provide univariate descriptive statistics and appropriate plots for the training data set. Provide a scatterplot matrix for at least two of the independent variables and the dependent variable. Derive a correlation matrix for any THREE quantitative variables in the dataset. Test the hypotheses that the correlations between each pairwise set of variables is 0 and provide a 80% confidence interval. Discuss the meaning of your analysis. Would you be worried about familywise error? Why or why not?
- 5 points. Linear Algebra and Correlation. Invert your 3 x 3 correlation matrix from above. (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. Conduct LU decomposition on the matrix.
- 5 points. Calculus-Based Probability & Statistics. Many times, it makes sense to fit a closed form distribution to data. Select a variable in the Kaggle.com training dataset that is skewed to the right, shift it so that the minimum value is absolutely above zero if necessary. 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>). 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)`). Plot a histogram and compare it with a histogram of your original variable. Using the exponential pdf, find the 5th and 95th percentiles using the cumulative distribution function (CDF). Also generate a 95% confidence interval from the empirical data, assuming normality. Finally, provide the empirical 5th percentile and 95th percentile of the data. Discuss.
- 10 points. Modeling. Build some type of multiple 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.

Load the data and examine it

```
household <- read.csv("all/train.csv")
dim(household)
```

```
## [1] 1460 81
```

```
summary(household)
```

```
##           Id           MSSubClass         MSZoning      LotFrontage
##  Min.      : 1.0      Min.      : 20.0      C (all): 10      Min.      : 21.00
## 1st Qu.: 365.8      1st Qu.: 20.0      FV      : 65      1st Qu.: 59.00
## Median : 730.5      Median : 50.0      RH      : 16      Median : 69.00
## Mean   : 730.5      Mean   : 56.9      RL      :1151      Mean   : 70.05
## 3rd Qu.:1095.2      3rd Qu.: 70.0      RM      : 218      3rd Qu.: 80.00
## Max.    :1460.0      Max.    :190.0                      Max.    :313.00
##                                     NA's     :259
##           LotArea       Street       Alley       LotShape  LandContour
##  Min.      : 1300      Grvl: 6      Grvl: 50      IR1:484      Bnk: 63
## 1st Qu.: 7554      Pave:1454      Pave: 41      IR2: 41      HLS: 50
## Median : 9478                      NA's:1369      IR3: 10      Low: 36
## Mean      :10517                      Reg:925      Lvl:1311
## 3rd Qu.: 11602
## Max.      :215245
##
## Utilities      LotConfig      LandSlope      Neighborhood      Condition1
## AllPub:1459      Corner : 263      Gtl:1382      Names :225      Norm :1260
## NoSeWa: 1      CulDSac: 94      Mod: 65      CollgCr:150      Feedr : 81
##                                     FR2 : 47      Sev: 13      OldTown:113      Artery : 48
##                                     FR3 : 4      Edwards:100      RRAn : 26
##                                     Inside :1052      Somerst: 86      PosN : 19
##                                     Gilbert: 79      RRAe : 11
##                                     (Other):707      (Other): 15
## Condition2      BldgType      HouseStyle      OverallQual
## Norm :1445      1Fam :1220      1Story :726      Min. : 1.000
## Feedr : 6      2fmCon: 31      2Story :445      1st Qu.: 5.000
## Artery : 2      Duplex: 52      1.5Fin :154      Median : 6.000
## PosN : 2      Twnhs : 43      SLvl : 65      Mean : 6.099
## RRNn : 2      TwnhsE: 114      SFoyer : 37      3rd Qu.: 7.000
## PosA : 1      1.5Unf : 14      Max. :10.000
## (Other): 2      (Other): 19
## OverallCond      YearBuilt      YearRemodAdd      RoofStyle
## Min. :1.000      Min. :1872      Min. :1950      Flat : 13
## 1st Qu.:5.000      1st Qu.:1954      1st Qu.:1967      Gable :1141
## Median :5.000      Median :1973      Median :1994      Gambrel: 11
## Mean :5.575      Mean :1971      Mean :1985      Hip : 286
## 3rd Qu.:6.000      3rd Qu.:2000      3rd Qu.:2004      Mansard: 7
## Max. :9.000      Max. :2010      Max. :2010      Shed : 2
##
## RoofMatl      Exterior1st      Exterior2nd      MasVnrType      MasVnrArea
## CompShg:1434      VinylSd:515      VinylSd:504      BrkCmn : 15      Min. : 0.0
## Tar&Grv: 11      HdBoard:222      MetalSd:214      BrkFace:445      1st Qu.: 0.0
```

```

## WdShngl: 6 MetalSd:220 HdBoard:207 None :864 Median : 0.0
## WdShake: 5 Wd Sdng:206 Wd Sdng:197 Stone :128 Mean : 103.7
## ClyTile: 1 Plywood:108 Plywood:142 NA's : 8 3rd Qu.: 166.0
## Membran: 1 CemntBd: 61 CmentBd: 60 Max. :1600.0
## (Other): 2 (Other):128 (Other):136 NA's :8
## ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure
## Ex: 52 Ex: 3 BrkTil:146 Ex :121 Fa : 45 Av :221
## Fa: 14 Fa: 28 CBlock:634 Fa : 35 Gd : 65 Gd :134
## Gd:488 Gd: 146 PConc :647 Gd :618 Po : 2 Mn :114
## TA:906 Po: 1 Slab : 24 TA :649 TA :1311 No :953
## TA:1282 Stone : 6 NA's: 37 NA's: 37 NA's: 38
## Wood : 3
##
## BsmtFinType1 BsmtFinSF1 BsmtFinType2 BsmtFinSF2
## ALQ :220 Min. : 0.0 ALQ : 19 Min. : 0.00
## BLQ :148 1st Qu.: 0.0 BLQ : 33 1st Qu.: 0.00
## GLQ :418 Median : 383.5 GLQ : 14 Median : 0.00
## LwQ : 74 Mean : 443.6 LwQ : 46 Mean : 46.55
## Rec :133 3rd Qu.: 712.2 Rec : 54 3rd Qu.: 0.00
## Unf :430 Max. :5644.0 Unf :1256 Max. :1474.00
## NA's: 37 NA's: 38
## BsmtUnfSF TotalBsmtSF Heating HeatingQC CentralAir
## Min. : 0.0 Min. : 0.0 Floor: 1 Ex:741 N: 95
## 1st Qu.: 223.0 1st Qu.: 795.8 GasA :1428 Fa: 49 Y:1365
## Median : 477.5 Median : 991.5 GasW : 18 Gd:241
## Mean : 567.2 Mean :1057.4 Grav : 7 Po: 1
## 3rd Qu.: 808.0 3rd Qu.:1298.2 OthW : 2 TA:428
## Max. :2336.0 Max. :6110.0 Wall : 4
##
## Electrical X1stFlrSF X2ndFlrSF LowQualFinSF
## FuseA: 94 Min. : 334 Min. : 0 Min. : 0.000
## FuseF: 27 1st Qu.: 882 1st Qu.: 0 1st Qu.: 0.000
## FuseP: 3 Median :1087 Median : 0 Median : 0.000
## Mix : 1 Mean :1163 Mean : 347 Mean : 5.845
## SBrkr:1334 3rd Qu.:1391 3rd Qu.: 728 3rd Qu.: 0.000
## NA's : 1 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 KitchenQual
## Min. :0.0000 Min. :0.000 Min. :0.000 Ex:100
## 1st Qu.:0.0000 1st Qu.:2.000 1st Qu.:1.000 Fa: 39
## Median :0.0000 Median :3.000 Median :1.000 Gd:586
## Mean :0.3829 Mean :2.866 Mean :1.047 TA:735
## 3rd Qu.:1.0000 3rd Qu.:3.000 3rd Qu.:1.000
## Max. :2.0000 Max. :8.000 Max. :3.000
##
## TotRmsAbvGrd Functional Fireplaces FireplaceQu GarageType

```

```

## Min. : 2.000 Maj1: 14 Min. :0.000 Ex : 24 2Types : 6
## 1st Qu.: 5.000 Maj2: 5 1st Qu.:0.000 Fa : 33 Attchd :870
## Median : 6.000 Min1: 31 Median :1.000 Gd :380 Basment: 19
## Mean : 6.518 Min2: 34 Mean :0.613 Po : 20 BuiltIn: 88
## 3rd Qu.: 7.000 Mod : 15 3rd Qu.:1.000 TA :313 CarPort: 9
## Max. :14.000 Sev : 1 Max. :3.000 NA's:690 Detchd :387
## Typ :1360 NA's : 81
## GarageYrBlt GarageFinish GarageCars GarageArea GarageQual
## Min. :1900 Fin :352 Min. :0.000 Min. : 0.0 Ex : 3
## 1st Qu.:1961 RFn :422 1st Qu.:1.000 1st Qu.: 334.5 Fa : 48
## Median :1980 Unf :605 Median :2.000 Median : 480.0 Gd : 14
## Mean :1979 NA's: 81 Mean :1.767 Mean : 473.0 Po : 3
## 3rd Qu.:2002 3rd Qu.:2.000 3rd Qu.: 576.0 TA :1311
## Max. :2010 Max. :4.000 Max. :1418.0 NA's: 81
## NA's :81
## GarageCond PavedDrive WoodDeckSF OpenPorchSF EnclosedPorch
## Ex : 2 N: 90 Min. : 0.00 Min. : 0.00 Min. : 0.00
## Fa : 35 P: 30 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.00
## Gd : 9 Y:1340 Median : 0.00 Median : 25.00 Median : 0.00
## Po : 7 Mean : 94.24 Mean : 46.66 Mean : 21.95
## TA :1326 3rd Qu.:168.00 3rd Qu.: 68.00 3rd Qu.: 0.00
## NA's: 81 Max. :857.00 Max. :547.00 Max. :552.00
##
## X3SsnPorch ScreenPorch PoolArea PoolQC
## Min. : 0.00 Min. : 0.00 Min. : 0.000 Ex : 2
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.000 Fa : 2
## Median : 0.00 Median : 0.00 Median : 0.000 Gd : 3
## Mean : 3.41 Mean : 15.06 Mean : 2.759 NA's:1453
## 3rd Qu.: 0.00 3rd Qu.: 0.00 3rd Qu.: 0.000
## Max. :508.00 Max. :480.00 Max. :738.000
##
## Fence MiscFeature MiscVal MoSold
## GdPrv: 59 Gar2: 2 Min. : 0.00 Min. : 1.000
## GdWo : 54 Othr: 2 1st Qu.: 0.00 1st Qu.: 5.000
## MnPrv: 157 Shed: 49 Median : 0.00 Median : 6.000
## MnWw : 11 TenC: 1 Mean : 43.49 Mean : 6.322
## NA's :1179 NA's:1406 3rd Qu.: 0.00 3rd Qu.: 8.000
## Max. :15500.00 Max. :12.000
##
## YrSold SaleType SaleCondition SalePrice
## Min. :2006 WD :1267 Abnorml: 101 Min. : 34900
## 1st Qu.:2007 New : 122 AdjLand: 4 1st Qu.:129975
## Median :2008 COD : 43 Alloca : 12 Median :163000
## Mean :2008 ConLD : 9 Family : 20 Mean :180921
## 3rd Qu.:2009 ConLI : 5 Normal :1198 3rd Qu.:214000
## Max. :2010 ConLw : 5 Partial: 125 Max. :755000
## (Other): 9

```

```
str(household)
```

```

## 'data.frame': 1460 obs. of 81 variables:
## $ Id : int 1 2 3 4 5 6 7 8 9 10 ...
## $ MSSubClass : int 60 20 60 70 60 50 20 60 50 190 ...
## $ MSZoning : Factor w/ 5 levels "C (all)","FV",...: 4 4 4 4 4 4 4 5 4 ...

```

```

## $ LotFrontage : int 65 80 68 60 84 85 75 NA 51 50 ...
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
## $ Street : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 ...
## $ Alley : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA ...
## $ LotShape : Factor w/ 4 levels "IR1","IR2","IR3",...: 4 4 1 1 1 1 4 1 4 4 ...
## $ LandContour : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 4 4 4 4 4 ...
## $ Utilities : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 ...
## $ LotConfig : Factor w/ 5 levels "Corner","CulDSac",...: 5 3 5 1 3 5 5 1 5 1 ...
## $ LandSlope : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",...: 6 25 6 7 14 12 21 17 18 4 ...
## $ Condition1 : Factor w/ 9 levels "Artery","Feedr",...: 3 2 3 3 3 3 3 5 1 1 ...
## $ Condition2 : Factor w/ 8 levels "Artery","Feedr",...: 3 3 3 3 3 3 3 3 1 ...
## $ BldgType : Factor w/ 5 levels "1Fam","2fmCon",...: 1 1 1 1 1 1 1 1 1 2 ...
## $ HouseStyle : Factor w/ 8 levels "1.5Fin","1.5Unf",...: 6 3 6 6 6 1 3 6 1 2 ...
## $ OverallQual : int 7 6 7 7 8 5 8 7 7 5 ...
## $ OverallCond : int 5 8 5 5 5 5 5 6 5 6 ...
## $ YearBuilt : int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
## $ YearRemodAdd : int 2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
## $ RoofStyle : Factor w/ 6 levels "Flat","Gable",...: 2 2 2 2 2 2 2 2 2 ...
## $ RoofMatl : Factor w/ 8 levels "ClyTile","CompShg",...: 2 2 2 2 2 2 2 2 2 ...
## $ Exterior1st : Factor w/ 15 levels "AsbShng","AsphShn",...: 13 9 13 14 13 13 13 7 4 9 ...
## $ Exterior2nd : Factor w/ 16 levels "AsbShng","AsphShn",...: 14 9 14 16 14 14 14 7 16 9 ...
## $ MasVnrType : Factor w/ 4 levels "BrkCmn","BrkFace",...: 2 3 2 3 2 3 4 4 3 3 ...
## $ MasVnrArea : int 196 0 162 0 350 0 186 240 0 0 ...
## $ ExterQual : Factor w/ 4 levels "Ex","Fa","Gd",...: 3 4 3 4 3 4 3 4 4 ...
## $ ExterCond : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 5 5 5 ...
## $ Foundation : Factor w/ 6 levels "BrkTil","CBlock",...: 3 2 3 1 3 6 3 2 1 1 ...
## $ BsmtQual : Factor w/ 4 levels "Ex","Fa","Gd",...: 3 3 3 4 3 3 1 3 4 4 ...
## $ BsmtCond : Factor w/ 4 levels "Fa","Gd","Po",...: 4 4 4 2 4 4 4 4 4 ...
## $ BsmtExposure : Factor w/ 4 levels "Av","Gd","Mn",...: 4 2 3 4 1 4 1 3 4 4 ...
## $ BsmtFinType1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",...: 3 1 3 1 3 3 3 1 6 3 ...
## $ BsmtFinSF1 : int 706 978 486 216 655 732 1369 859 0 851 ...
## $ BsmtFinType2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",...: 6 6 6 6 6 6 6 6 2 6 ...
## $ BsmtFinSF2 : int 0 0 0 0 0 0 0 32 0 0 ...
## $ BsmtUnfSF : int 150 284 434 540 490 64 317 216 952 140 ...
## $ TotalBsmtSF : int 856 1262 920 756 1145 796 1686 1107 952 991 ...
## $ Heating : Factor w/ 6 levels "Floor","GasA",...: 2 2 2 2 2 2 2 2 2 ...
## $ HeatingQC : Factor w/ 5 levels "Ex","Fa","Gd",...: 1 1 1 3 1 1 1 1 3 1 ...
## $ CentralAir : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 ...
## $ Electrical : Factor w/ 5 levels "FuseA","FuseF",...: 5 5 5 5 5 5 5 5 5 2 5 ...
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath : int 1 0 1 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath : int 0 1 0 0 0 0 0 0 0 0 ...
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...
## $ HalfBath : int 1 0 1 0 1 1 0 1 0 0 ...
## $ BedroomAbvGr : int 3 3 3 3 4 1 3 3 2 2 ...
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 2 2 ...
## $ KitchenQual : Factor w/ 4 levels "Ex","Fa","Gd",...: 3 4 3 3 3 4 3 4 4 ...
## $ TotRmsAbvGrd : int 8 6 6 7 9 5 7 7 8 5 ...
## $ Functional : Factor w/ 7 levels "Maj1","Maj2",...: 7 7 7 7 7 7 7 7 3 7 ...
## $ Fireplaces : int 0 1 1 1 1 0 1 2 2 2 ...

```



```
## $ FireplaceQu : Factor w/ 5 levels "Ex","Fa","Gd",...: NA 5 5 3 5 NA 3 5 5 5 ...
## $ GarageType : Factor w/ 6 levels "2Types","Attchd",...: 2 2 2 6 2 2 2 2 6 2 ...
## $ GarageYrBlt : int 2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
## $ GarageFinish : Factor w/ 3 levels "Fin","RFn","Unf": 2 2 2 3 2 3 2 2 3 2 ...
## $ GarageCars : int 2 2 2 3 3 2 2 2 2 1 ...
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...
## $ GarageQual : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 5 5 2 3 ...
## $ GarageCond : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ PavedDrive : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...
## $ EnclosedPorch: int 0 0 0 272 0 0 0 228 205 0 ...
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolQC : Factor w/ 3 levels "Ex","Fa","Gd": NA NA NA NA NA NA NA NA NA NA ...
## $ Fence : Factor w/ 4 levels "GdPrv","GdWo",...: NA NA NA NA NA 3 NA NA NA NA ...
## $ MiscFeature : Factor w/ 4 levels "Gar2","Othr",...: NA NA NA NA NA 3 NA 3 NA NA ...
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...
## $ MoSold : int 2 5 9 2 12 10 8 11 4 1 ...
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
## $ SaleType : Factor w/ 9 levels "COD","Con","ConLD",...: 9 9 9 9 9 9 9 9 9 9 ...
## $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",...: 5 5 5 1 5 5 5 5 1 5 ...
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

```
head(household)
```

```
## Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape
## 1 1 60 RL 65 8450 Pave <NA> Reg
## 2 2 20 RL 80 9600 Pave <NA> Reg
## 3 3 60 RL 68 11250 Pave <NA> IR1
## 4 4 70 RL 60 9550 Pave <NA> IR1
## 5 5 60 RL 84 14260 Pave <NA> IR1
## 6 6 50 RL 85 14115 Pave <NA> IR1
## LandContour Utilities LotConfig LandSlope Neighborhood Condition1
## 1 Lvl AllPub Inside Gtl CollgCr Norm
## 2 Lvl AllPub FR2 Gtl Veenker Feedr
## 3 Lvl AllPub Inside Gtl CollgCr Norm
## 4 Lvl AllPub Corner Gtl Crawfor Norm
## 5 Lvl AllPub FR2 Gtl NoRidge Norm
## 6 Lvl AllPub Inside Gtl Mitchel Norm
## Condition2 BldgType HouseStyle OverallQual OverallCond YearBuilt
## 1 Norm 1Fam 2Story 7 5 2003
## 2 Norm 1Fam 1Story 6 8 1976
## 3 Norm 1Fam 2Story 7 5 2001
## 4 Norm 1Fam 2Story 7 5 1915
## 5 Norm 1Fam 2Story 8 5 2000
## 6 Norm 1Fam 1.5Fin 5 5 1993
## YearRemodAdd RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType
## 1 2003 Gable CompShg VinylSd VinylSd BrkFace
## 2 1976 Gable CompShg MetalSd MetalSd None
## 3 2002 Gable CompShg VinylSd VinylSd BrkFace
## 4 1970 Gable CompShg Wd Sdng Wd Shng None
## 5 2000 Gable CompShg VinylSd VinylSd BrkFace
```

## 6	1995	Gable	CompShg	VinylSd	VinylSd	None	
##	MasVnrArea	ExterQual	ExterCond	Foundation	BsmtQual	BsmtCond	BsmtExposure
## 1	196	Gd	TA	PConc	Gd	TA	No
## 2	0	TA	TA	CBlock	Gd	TA	Gd
## 3	162	Gd	TA	PConc	Gd	TA	Mn
## 4	0	TA	TA	BrkTil	TA	Gd	No
## 5	350	Gd	TA	PConc	Gd	TA	Av
## 6	0	TA	TA	Wood	Gd	TA	No
##	BsmtFinType1	BsmtFinSF1	BsmtFinType2	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	
## 1	GLQ	706	Unf	0	150	856	
## 2	ALQ	978	Unf	0	284	1262	
## 3	GLQ	486	Unf	0	434	920	
## 4	ALQ	216	Unf	0	540	756	
## 5	GLQ	655	Unf	0	490	1145	
## 6	GLQ	732	Unf	0	64	796	
##	Heating	HeatingQC	CentralAir	Electrical	X1stFlrSF	X2ndFlrSF	LowQualFinSF
## 1	GasA	Ex	Y	SBrkr	856	854	0
## 2	GasA	Ex	Y	SBrkr	1262	0	0
## 3	GasA	Ex	Y	SBrkr	920	866	0
## 4	GasA	Gd	Y	SBrkr	961	756	0
## 5	GasA	Ex	Y	SBrkr	1145	1053	0
## 6	GasA	Ex	Y	SBrkr	796	566	0
##	GrLivArea	BsmtFullBath	BsmtHalfBath	FullBath	HalfBath	BedroomAbvGr	
## 1	1710	1	0	2	1	3	
## 2	1262	0	1	2	0	3	
## 3	1786	1	0	2	1	3	
## 4	1717	1	0	1	0	3	
## 5	2198	1	0	2	1	4	
## 6	1362	1	0	1	1	1	
##	KitchenAbvGr	KitchenQual	TotRmsAbvGrd	Functional	Fireplaces	FireplaceQu	
## 1	1	Gd	8	Typ	0	<NA>	
## 2	1	TA	6	Typ	1	TA	
## 3	1	Gd	6	Typ	1	TA	
## 4	1	Gd	7	Typ	1	Gd	
## 5	1	Gd	9	Typ	1	TA	
## 6	1	TA	5	Typ	0	<NA>	
##	GarageType	GarageYrBlt	GarageFinish	GarageCars	GarageArea	GarageQual	
## 1	Attchd	2003	RFn	2	548	TA	
## 2	Attchd	1976	RFn	2	460	TA	
## 3	Attchd	2001	RFn	2	608	TA	
## 4	Detchd	1998	Unf	3	642	TA	
## 5	Attchd	2000	RFn	3	836	TA	
## 6	Attchd	1993	Unf	2	480	TA	
##	GarageCond	PavedDrive	WoodDeckSF	OpenPorchSF	EnclosedPorch	X3SsnPorch	
## 1	TA	Y	0	61	0	0	
## 2	TA	Y	298	0	0	0	
## 3	TA	Y	0	42	0	0	
## 4	TA	Y	0	35	272	0	
## 5	TA	Y	192	84	0	0	
## 6	TA	Y	40	30	0	320	
##	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold YrSold
## 1	0	0	<NA>	<NA>	<NA>	0	2 2008
## 2	0	0	<NA>	<NA>	<NA>	0	5 2007
## 3	0	0	<NA>	<NA>	<NA>	0	9 2008

```
## 4      0      0 <NA> <NA>      <NA>      0      2    2006
## 5      0      0 <NA> <NA>      <NA>      0     12    2008
## 6      0      0 <NA> MnPrv      Shed      700     10    2009
##   SaleType SaleCondition SalePrice
## 1      WD      Normal    208500
## 2      WD      Normal    181500
## 3      WD      Normal    223500
## 4      WD      Abnorml    140000
## 5      WD      Normal    250000
## 6      WD      Normal    143000
```

Descriptive Statistics

- Let's look at some plots and see the trend of the data more closely. We'll start with plotting and visualizing the quantitative variables such as LotArea, LotFrontage, MasVnrArea, and SalePrice to see how the data behave.
- Note the dependant variable is the SalePrice a continous numerical variable.

```
library(ggplot2)
library(gridExtra)
library(dplyr)
```

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

## The following object is masked from 'package:gridExtra':
##
##   combine

## The following objects are masked from 'package:stats':
##
##   filter, lag

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

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16
```

```
library(FeatureHashing)
```

```
max(household$SalePrice)
```

```
## [1] 755000
```

```
# quantitative variable plots
```

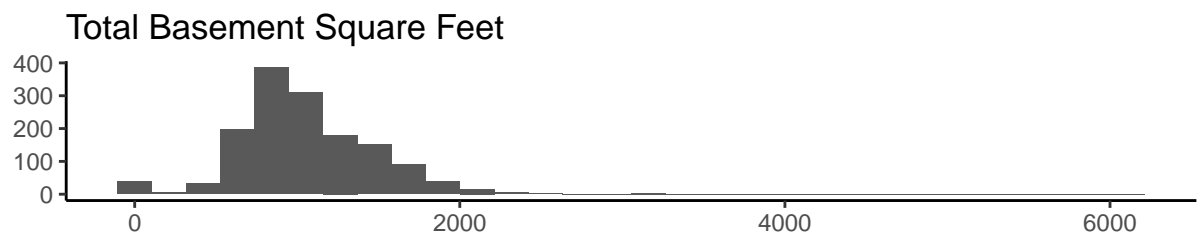
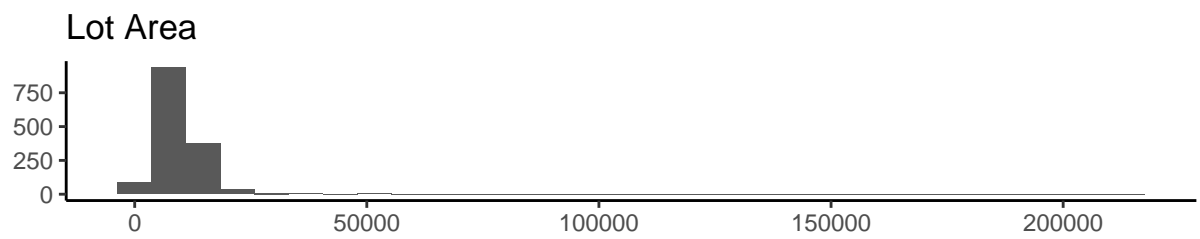
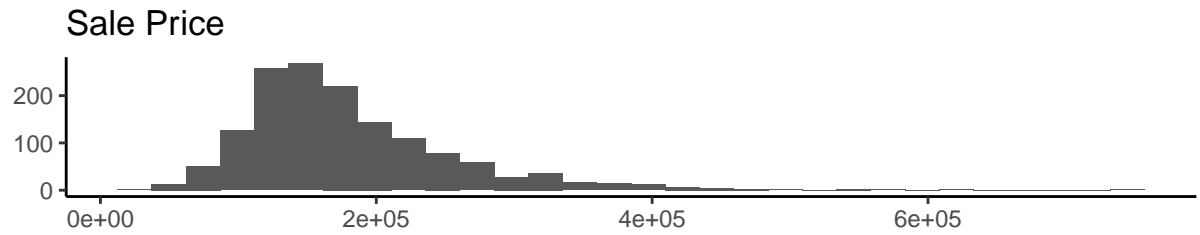
```
saleprice_plot <- ggplot(household, aes(SalePrice)) +  
  geom_histogram() +  
  ggtitle("Sale Price") +  
  xlab("") + ylab("") +  
  theme_bw() +  
  theme_classic()
```

```
lotarea_plot <- ggplot(household, aes(LotArea)) +  
  geom_histogram() +  
  ggtitle("Lot Area") +  
  xlab("") + ylab("") +  
  theme_bw() +  
  theme_classic()
```

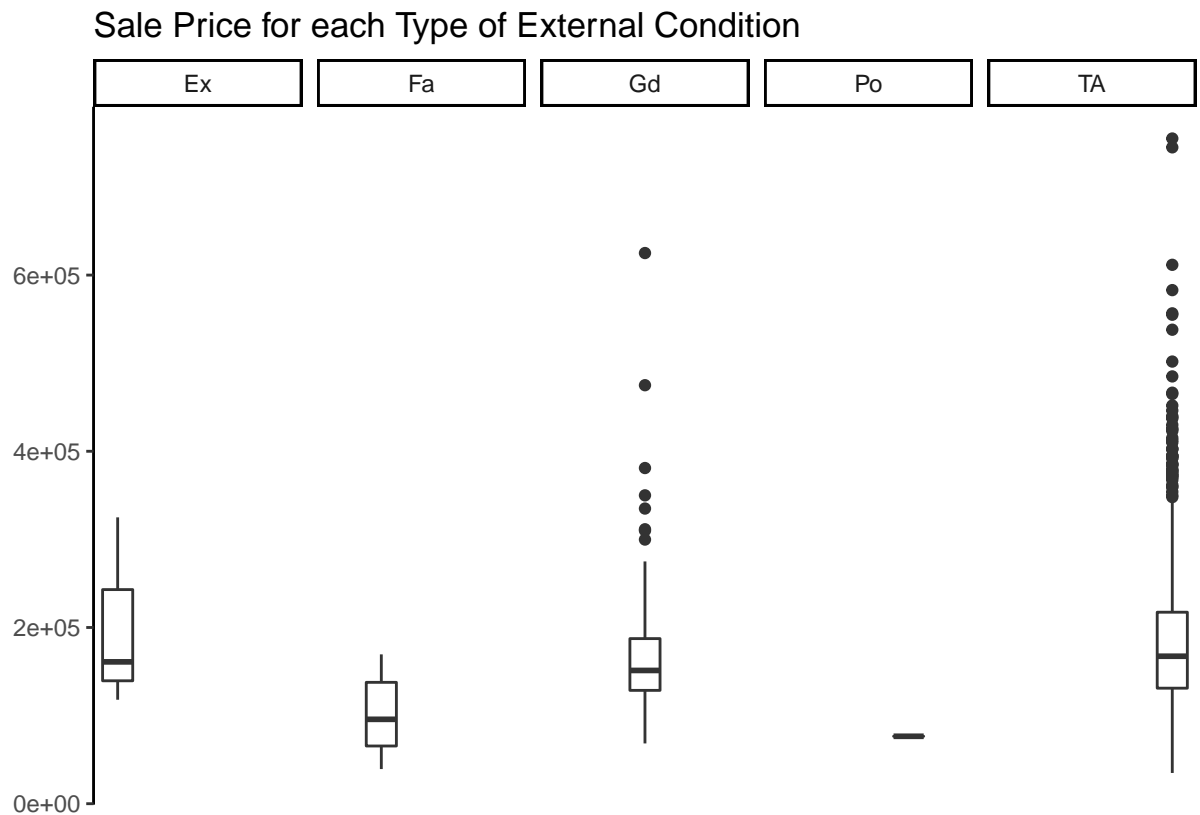
```
totalbsmtsf_plot <- ggplot(household, aes(TotalBsmtSF)) +  
  geom_histogram() +  
  ggtitle("Total Basement Square Feet") +  
  xlab("") + ylab("") +  
  theme_bw() +  
  theme_classic()
```

```
grid.arrange(saleprice_plot, lotarea_plot, totalbsmtsf_plot)
```

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

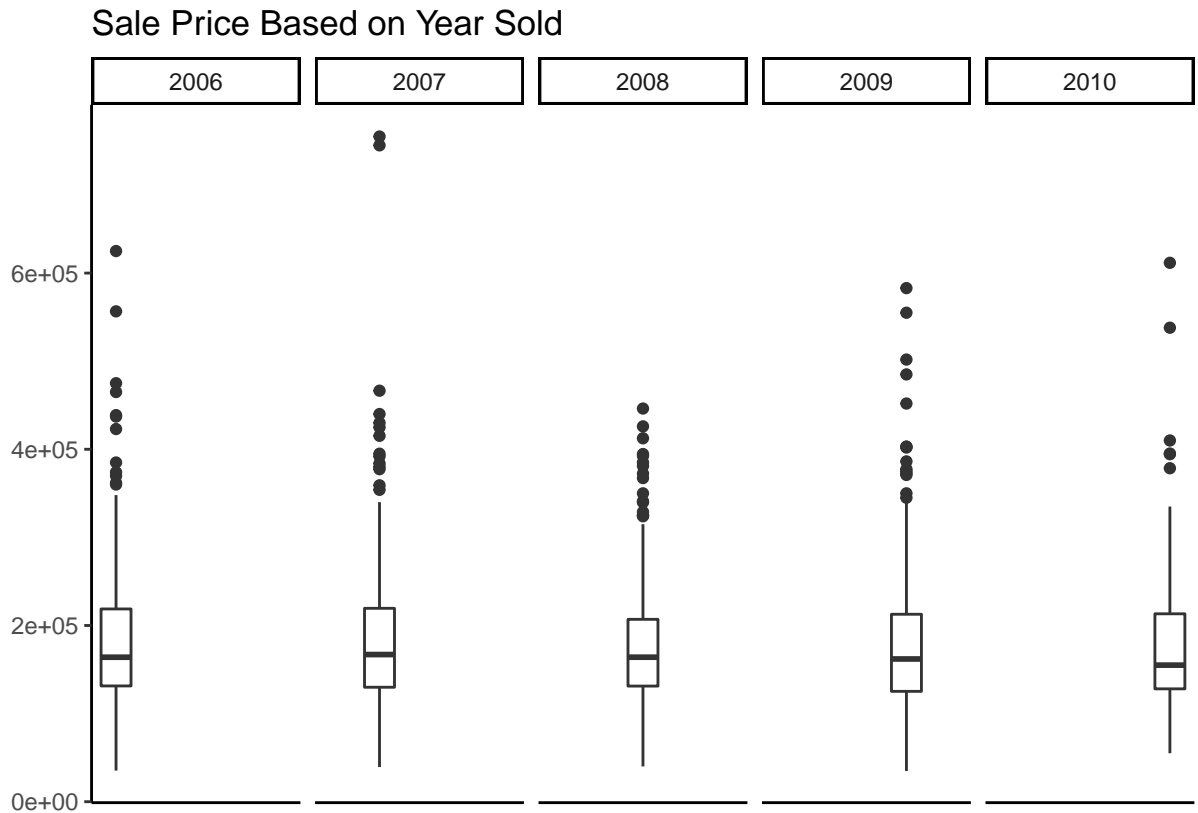


```
ggplot(household, aes(ExterCond, SalePrice)) +
  geom_boxplot() +
  facet_grid(~ ExterCond) +
  ggtitle("Sale Price for each Type of External Condition ") +
  xlab("") + ylab("") +
  theme_bw() +
  theme_classic() +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.line.x = element_blank())
```



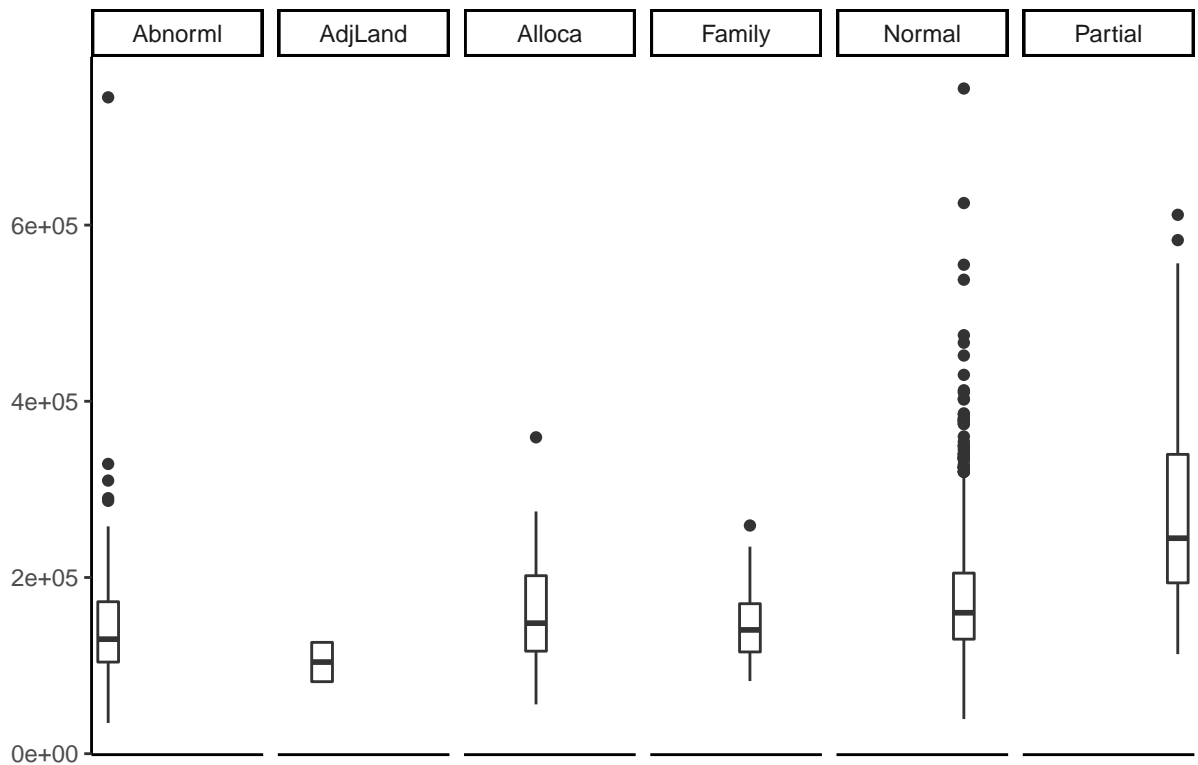
```
ggplot(household, aes(YrSold, SalePrice)) +
  geom_boxplot() +
  facet_grid(. ~ YrSold) +
  ggtitle("Sale Price Based on Year Sold") +
  xlab("") + ylab("") +
  theme_bw() +
  theme_classic() +
  theme(axis.ticks.x=element_blank(),
        axis.text.x = element_blank())
```

Warning: Continuous x aesthetic -- did you forget aes(group=...)?

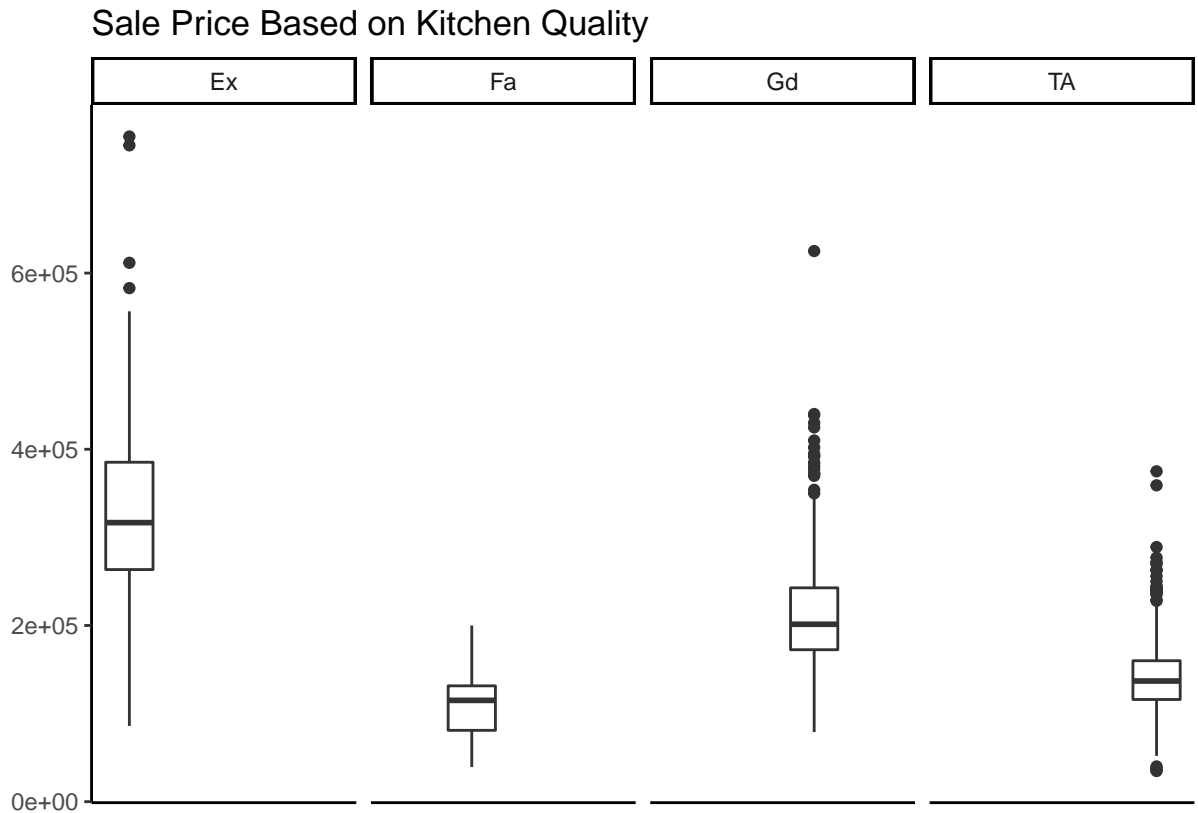


```
ggplot(household, aes(SaleCondition, SalePrice)) +
  geom_boxplot() +
  facet_grid(~ SaleCondition) +
  ggtitle("Sale Price Based on Sale Condition") +
  xlab("") + ylab("") +
  theme_bw() +
  theme_classic() +
  theme(axis.ticks.x=element_blank(),
        axis.text.x = element_blank())
```

Sale Price Based on Sale Condition



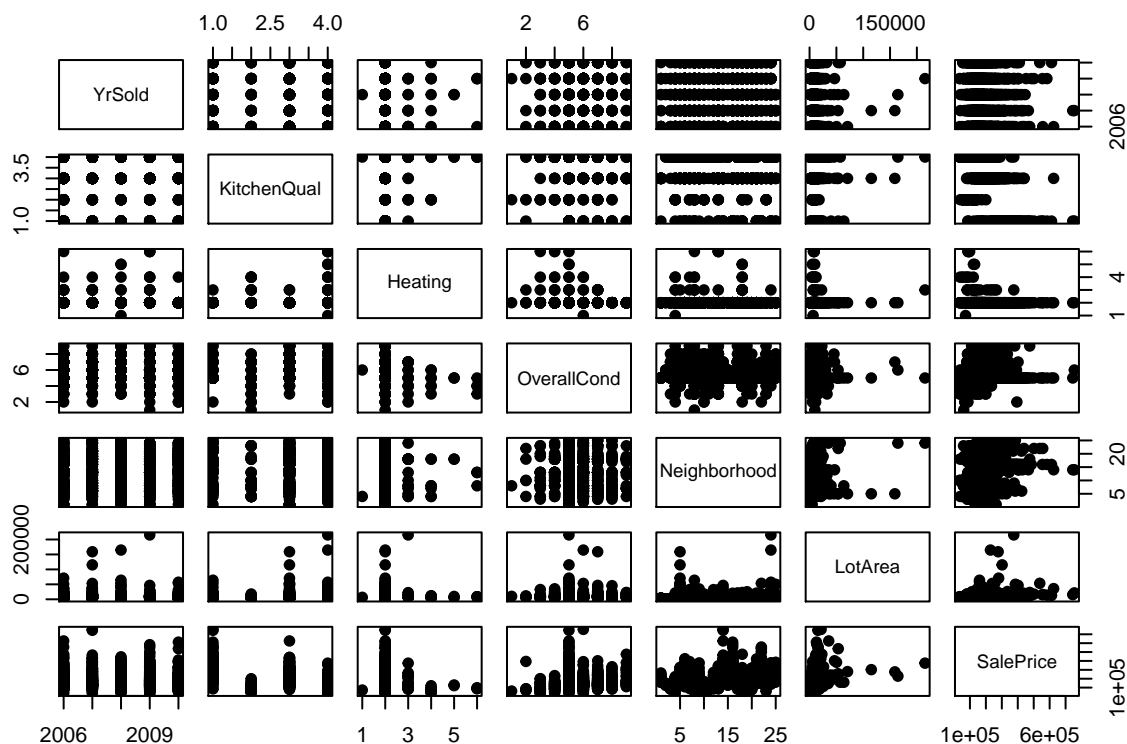
```
ggplot(household, aes(KitchenQual, SalePrice)) +
  geom_boxplot() +
  facet_grid(~ KitchenQual) +
  ggtitle("Sale Price Based on Kitchen Quality") +
  xlab("") + ylab("") +
  theme_bw() +
  theme_classic() +
  theme(axis.ticks.x=element_blank(),
        axis.text.x = element_blank())
```

- Plots show that the Sale price is quite left skewed and the lot area is heavily left-skewed; good for analysis later.
- We also see that the median price of sold homes is about the same for each year. For normal sale conditions, there are heavy outliers and that could make a influence in our model and analysis. The same goes for fairly decent homes in okay external condition.
- Let's create a scatterplot matrix using the pairs() function and see the visualization. I will look at the most common things I feel are most important in looking for a place to call home such as Year sold, kitchen quality, heating, overall condition, neighborhood, lot area, and sale price.

```
columns_scatterplotmatrix <- c("YrSold", "KitchenQual", "Heating", "OverallCond", "Neighborhood",
                              "LotArea", "SalePrice")

pairs(household[, columns_scatterplotmatrix], pch = 19)
```



- Let's also examine the correlation matrix as well for 3 quantitative variables:

- Sale Price: Continuous
- Garage Area: Continuous
- Lot Area: Continuous

```
cor_matrix <- cor(as.matrix(household[, c("SalePrice", "LotArea", "GarageArea"))))
cor_matrix
```

```
##           SalePrice   LotArea GarageArea
## SalePrice  1.0000000  0.2638434  0.6234314
## LotArea    0.2638434  1.0000000  0.1804028
## GarageArea 0.6234314  0.1804028  1.0000000
```

- With our 3x3 matrix, let's do a 80% confidence interval using the hypothesis below:

- Null hypothesis: $cor(x, y) = 0$ that is there is not correlation between the two variables in question
- Alternative hypothesis: $cor(x, y) \neq 0$ that is there is some correlation big or small between the two variables.

```
cor.test(household$SalePrice, household$LotArea, method = "pearson", conf.level = 0.8)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: household$SalePrice and household$LotArea  
## t = 10.445, df = 1458, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 80 percent confidence interval:  
## 0.2323391 0.2947946  
## sample estimates:  
## cor  
## 0.2638434
```

```
cor.test(household$LotArea, household$GarageArea, method = "pearson", conf.level = 0.8)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: household$LotArea and household$GarageArea  
## t = 7.0034, df = 1458, p-value = 3.803e-12  
## alternative hypothesis: true correlation is not equal to 0  
## 80 percent confidence interval:  
## 0.1477356 0.2126767  
## sample estimates:  
## cor  
## 0.1804028
```

```
cor.test(household$SalePrice, household$GarageArea, method = "pearson", conf.level = 0.8)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: household$SalePrice and household$GarageArea  
## t = 30.446, df = 1458, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 80 percent confidence interval:  
## 0.6024756 0.6435283  
## sample estimates:  
## cor  
## 0.6234314
```

- Based on these correlation tests, we can see that we can reject the null hypothesis and favor the alternative that is $cor(x, y) \neq 0$ for the variables chosen.
- There is a quite strong positive correlation between SalePrice (dependent variable) and GarageArea (independent variable). This makes sense as if one is buying a home, the sale price changes based on the area of the garage.
- Lot area doesn't have strong correlation with regards to sale price which it could be lot area may not have much impact on sale price. Same also goes for lot area and garage area.
- We are 80% confident the true correlation is within the intervals above for the specified variables.

- Familywise error is defined as $FWE \leq 1 - (1 - \alpha_{IT})^c$ and is the probability of coming to at least one false conclusion in a series of hypothesis tests. α_{IT} is the alpha level for an individual test (in this case 0.2) and c is the number of comparisons. c = 3 test and computing the familywise error gives
- $FWE \leq 1 - (1 - \alpha_{IT})^c = 1 - (1 - 0.2)^3 = 0.488$ which is quite high considering only 3 tests were made and something that would have to be concern of getting a type 1 error.

Linear Algebra and Correlation

- Per the description of this section, let's invert our 3x3 matrix from above that is

```
inv_cor_matrix <- solve(cor_matrix) # precision matrix
inv_cor_matrix
```

```
##           SalePrice      LotArea  GarageArea
## SalePrice  1.7016986 -0.26625940 -1.01285847
## LotArea    -0.2662594  1.07530074 -0.02799273
## GarageArea -1.0128585 -0.02799273  1.63649778
```

- Now multiply the precision matrix by the correlation matrix and do the other way around, then do LU Decomposition

```
# precision matrix x correlation matrix
inv_cor_matrix %%% cor_matrix
```

```
##           SalePrice      LotArea  GarageArea
## SalePrice  1.000000e+00 -5.551115e-17  0.000000e+00
## LotArea     7.979728e-17  1.000000e+00  6.591949e-17
## GarageArea  0.000000e+00  0.000000e+00  1.000000e+00
```

```
cor_matrix %%% inv_cor_matrix
```

```
##           SalePrice      LotArea  GarageArea
## SalePrice      1 2.428613e-17          0
## LotArea         0 1.000000e+00          0
## GarageArea      0 3.469447e-17          1
```

```
library(matrixcalc) # LU Decomposition
lu.decomposition(cor_matrix %%% inv_cor_matrix)
```

```
## $L
##      [,1]      [,2] [,3]
## [1,]    1 0.000000e+00    0
## [2,]    0 1.000000e+00    0
## [3,]    0 3.469447e-17    1
##
## $U
##      [,1]      [,2] [,3]
## [1,]    1 2.428613e-17    0
## [2,]    0 1.000000e+00    0
## [3,]    0 0.000000e+00    1
```

```
lu.decomposition(inv_cor_matrix %*% cor_matrix) # they're not commutative
```

```
## $L
##           [,1] [,2] [,3]
## [1,] 1.000000e+00    0    0
## [2,] 7.979728e-17    1    0
## [3,] 0.000000e+00    0    1
##
## $U
##           [,1]           [,2]           [,3]
## [1,]    1 -5.551115e-17 0.000000e+00
## [2,]    0 1.000000e+00 6.591949e-17
## [3,]    0 0.000000e+00 1.000000e+00
```

```
min(household$TotalBsmtSF)
```

```
## [1] 0
```

Calculus Based Probability & Statistics

- For this, we will choose the total basement square feet or TotalBsmtSF variable.

```
library(MASS)
```

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

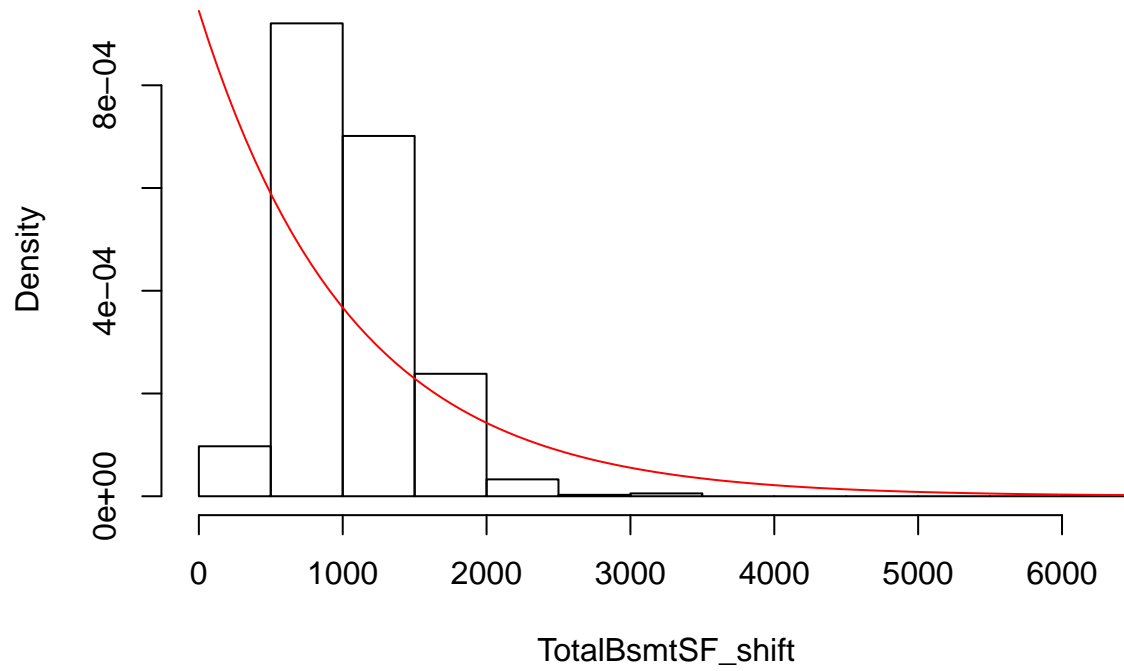
## The following object is masked from 'package:dplyr':
##
##      select
```

```
# shift TotalBsmtSF variable so min > 0
TotalBsmtSF_shift <- household$TotalBsmtSF + 1.0
TotalBsmtSF_fit <- fitdistr(TotalBsmtSF_shift, "exponential")
TotalBsmtSF_fit$estimate # optimal value of the rate parameter lambda
```

```
##           rate
## 0.0009447961
```

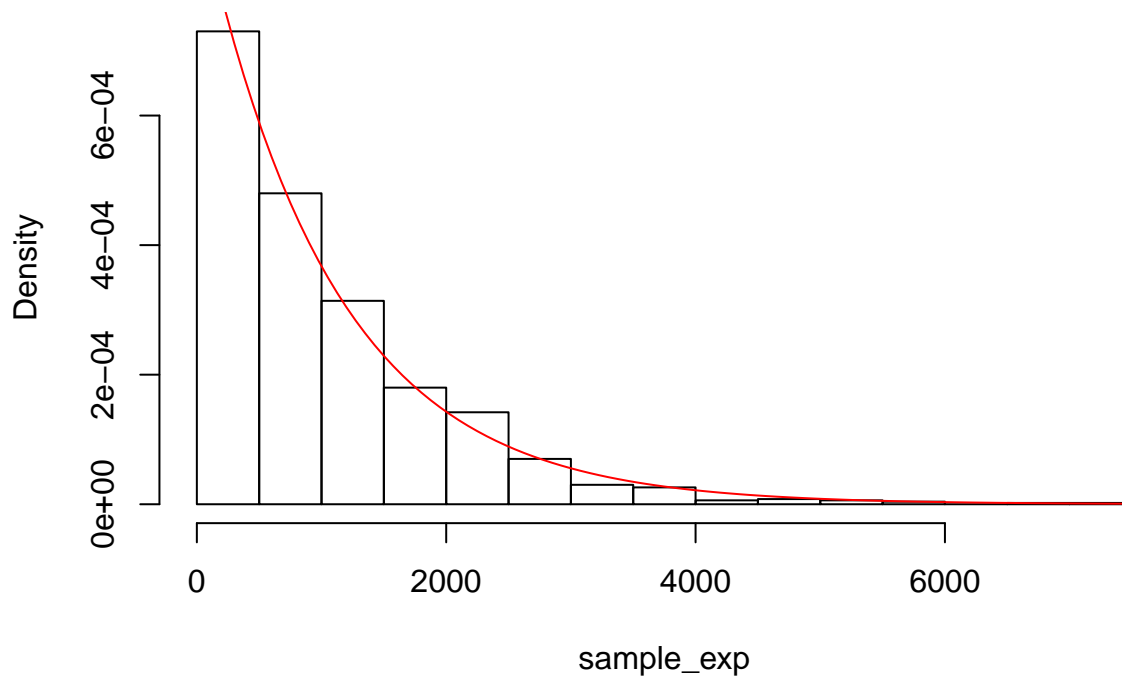
```
par(mfrow=c(1,1))
hist(TotalBsmtSF_shift, pch = 20, prob=TRUE)
curve(dexp(x, TotalBsmtSF_fit$estimate), col="red", add=T)
```

Histogram of TotalBsmtSF_shift



```
# take 1000 random samples from a exponential distribution
sample_exp <- rexp(1000, rate=TotalBsmtSF_fit$estimate)
hist(sample_exp, pch = 20, prob=TRUE)
curve(dexp(x, TotalBsmtSF_fit$estimate), col="red", add=T)
```

Histogram of sample_exp



```
# use the cdf that is 1 - exp(-lambda*x)
CDF_sample_exp <- 1 - exp(-TotalBsmtSF_fit$estimate*household$TotalBsmtSF)
# 5% and 95% percentiles
quantile(CDF_sample_exp, .05)
```

```
##          5%
## 0.3877585
```

```
quantile(CDF_sample_exp, .95)
```

```
##          95%
## 0.8091424
```

```
# 95% confidence interval assuming normality (mean and sd are the same for a exponential that is 1/lambda)
```

```
mean_exp <- 1/TotalBsmtSF_fit$estimate
sd_exp <- mean_exp
```

```
# standard error
```

```
sd_error <- qnorm(0.95)* (sd_exp / sqrt(length(household$TotalBsmtSF)))
```

```
left_ci <- mean_exp - sd_error
```

```
right_ci <- mean_exp + sd_error
```

```
# confidence interval
```

```
c(left_ci, right_ci)
```

```
##      rate      rate
## 1012.866 1103.992
```

```
# quantile of empricial data
emp_data <- ecdf(household$TotalBsmtSF)
emp_data(5)
```

```
## [1] 0.02534247
```

```
emp_data(95)
```

```
## [1] 0.02534247
```

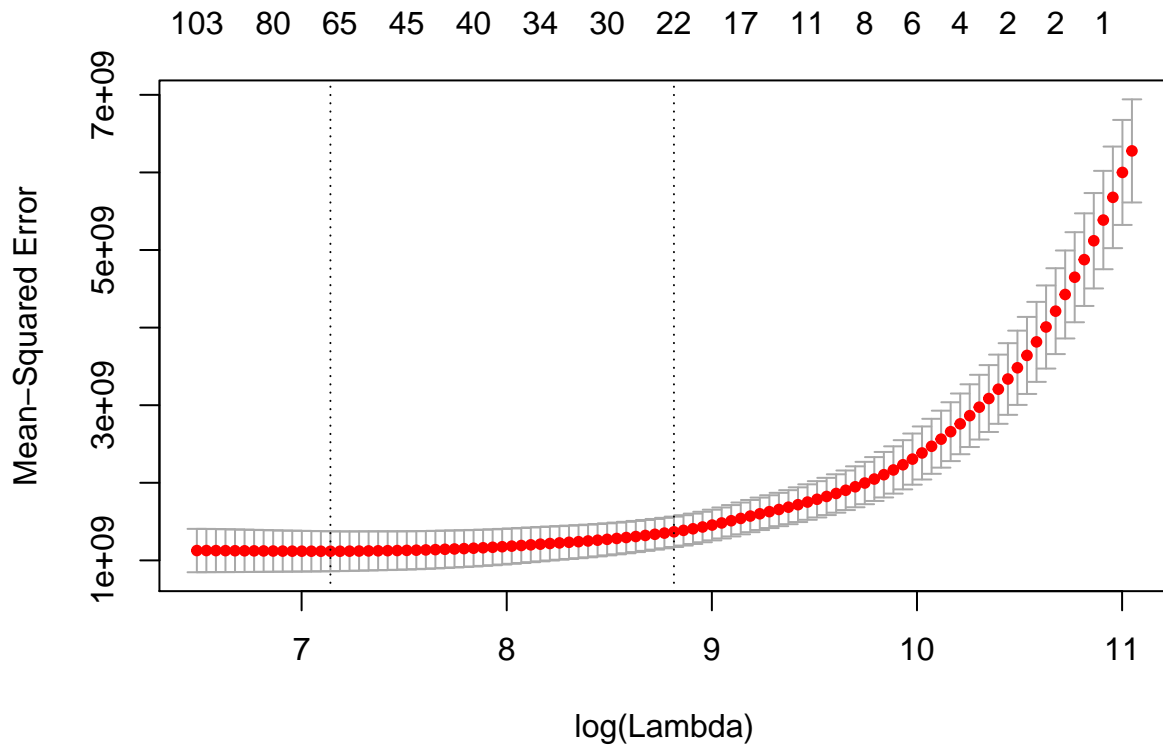
- Assuming normality for a exponential distribution is a good idea as our best fit rate $1/\lambda$ is within our 95% confidence interval so we are 95% confident the true value of $1/\lambda$ falls within (1012.866, 1103.992).

Modeling

- Let's use feaature selection such as LASSO to select parameters for our multiple regression model and submit our scores to kaggle. I will also use the technique of feature hashing when dealing with categorical variables and then use the glmnet library and it's function `cv.glmnet()` to come up with a model and then predict the Sale Prices of the test dataset.
- Feature hashing example: <http://amunategui.github.io/feature-hashing/>
- LASSO (Least absolute shrinkage and selection operator) is used to to avoid overfitting by penalizing large coefficients and it can shrink some coefficients of the feautres so in turn it also does feature selection.
- LASSO introduction: <http://ricardoscr.github.io/how-to-use-ridge-and-lasso-in-r.html>

```
features <- setdiff(names(household), "SalePrice")
objtrain_hashed <- hashed.model.matrix(~., data=household[, features], hash.size = 2^12,
                                     transpose = FALSE)
objtrain_hashed <- as(objtrain_hashed, "dgCMatrix")

cv.fit <- cv.glmnet(x=objtrain_hashed, y=household$SalePrice, type.measure = "mse")
plot(cv.fit)
```

- We see that the algorithm based on the plot has the lambda minimum value of about 65 variables and based on the graph, taking the lambda minimum (first dotted line on the left) has a smaller mean squared error. This lambda value will be used in our predictions of the test dataset.
- Finally let's use this model to predict the Saleprice in the test dataset and submit it to kaggle.

```
household_test <- read.csv("all/test.csv")
objtest_hashed <- hashed.model.matrix(~., data=household_test[, features], hash.size =2^12,
                                     transpose = FALSE)
objtest_hashed <- as(objtest_hashed, "dgCMatrix")
household_predict <- predict(cv.fit, objtest_hashed, s="lambda.min")

# append the ID to the predictions
household_predict <- data.frame(Id=household_test$Id, SalePrice=household_predict)
colnames(household_predict) <- c("Id", "SalePrice")
write.csv(household_predict, file = "all/household_predictions.csv",
          row.names = FALSE)
```