SCohenDevries_Final

Sam Cohen-Devries
5/27/2019

Generate a random variable X that has 10,000 random uniform numbers from 1 to N

```
N <- 8
mu <- (N+1)/2
sig <- mu
X <- runif(10000,0,N)
```

Then generate a random variable Y that has 10,000 random normal numbers with a mean of mu=sigma=(N+1)/2.

```
Y <- rnorm(10000,mu,sig)
```

"x" is estimated as the median of the X variable

```
x <- median(X)
```

"y" is estimated as the 1st quartile of the Y variable

```
y <- quantile(Y,0.25)
```

```
pX_gt_x <- length(X[X>x])/length(X)
pX_gt_y <- length(X[X>y])/length(X)
```

```
#P(X>x|X>y) = P(X>x intersect X>y)/P(X>y) = P(X>x)/P(X>y)
pX_gt_x/pX_gt_y
## [1] 0.6060606
#P(X>x, Y>y)
pX_gt_x*pX_gt_y
## [1] 0.4125
#P(X<x | X>y)
length(X[X<x&X>y])/length(X[X>y])
## [1] 0.3939394
```

Investigate whether P(X>x and Y>y)=P(X>x)P(Y>y) by building a table and evaluating the marginal and joint probabilities.

```
df <- data.frame(cbind(X,Y))
df$y <- y
df$x <- x
df2 <-subset(df,X>x & Y>y)
nrow(df2)/nrow(df)
## [1] 0.3743
(nrow(subset(df,X>x))/nrow(df)) * (nrow(subset(df,Y>y)))/nrow(df)
## [1] 0.375
```

Check to see if independence holds by using Fisher's Exact Test and the Chi Square Test.

```
#fisher.test(X,Y)
chisq.test(table(X,Y))
## Warning in chisq.test(table(X, Y)): Chi-squared approximation may be
## incorrect
##
##  Pearson's Chi-squared test
##
## data:  table(X, Y)
## X-squared = 99990000, df = 99980000, p-value = 0.2397
```

What is the difference between the two? Which is most appropriate? Fisher's exact test is most accurate when working with small sample sizes, which is not the case here. Fisher's tests also provides an exact p-value, but it makes some assumptions about the data structure that may or may not be accurate. For larger sample sizes, we can expect the Chi-Square to be most useful. In this case, the Chi Square Test appears to be most appropriate.

---

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

##

```
##
## Attaching package: 'matrixcalc'
## The following object is masked from 'package:corpcor':
##
##     is.positive.definite
```

Provide univariate descriptive statistics and appropriate plots for the training data set

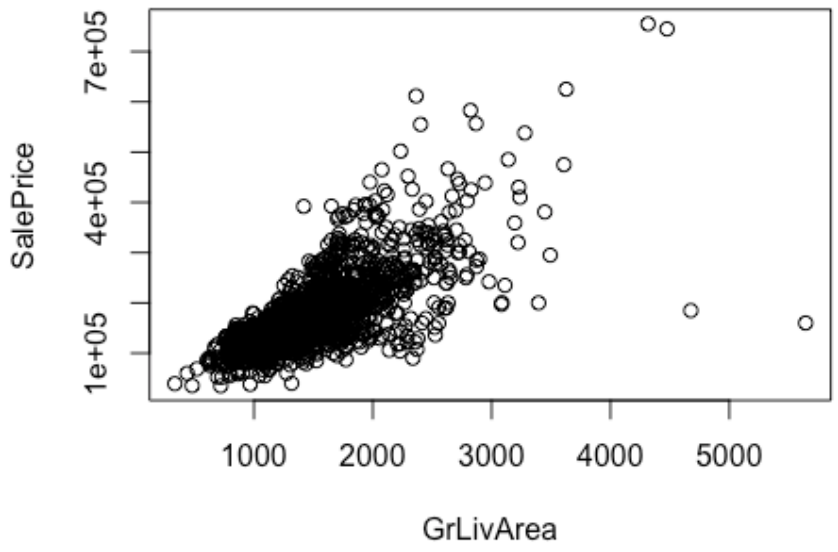
```
summary(train)
##        Id          MSSubClass      MSZoning     LotFrontage
##  Min.   :   1.0   Min.   : 20.0   C (all):  10   Min.   : 21.00
##  1st Qu.: 365.8   1st Qu.: 20.0   FV     :  65   1st Qu.: 59.00
##  Median : 730.5   Median : 50.0   RH     :  16   Median : 69.00
##  Mean   : 730.5   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
```

```
##  Max.   :2010   ConLw  :  5   Partial:125   Max.   :755000
##                 (Other):  9
```

```
hist(train$SalePrice)
```


```
plot(SalePrice ~ GrLivArea,data=train)
```



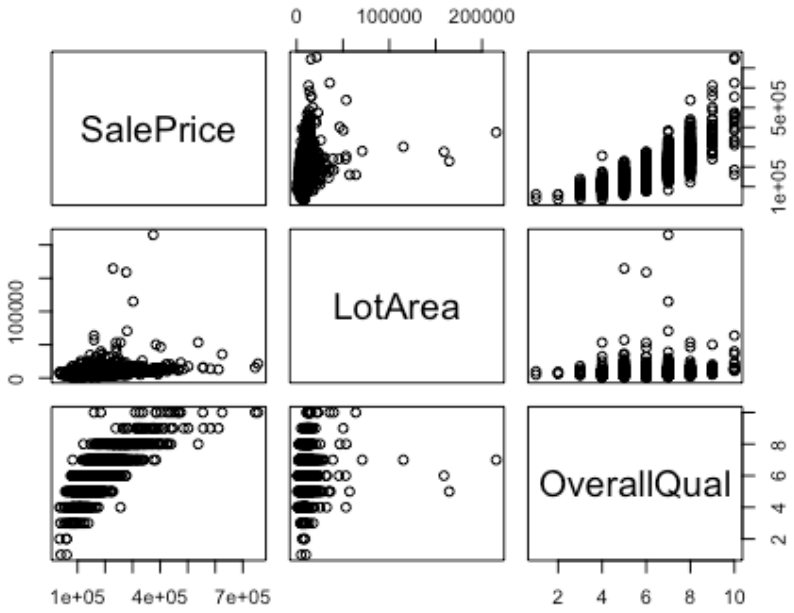
Provide a scatterplot matrix for at least two of the independent variables and the dependent variable

```
sapply(train,class) #checking for numeric variables
##           Id   MSSubClass     MSZoning  LotFrontage      LotArea
##    "integer"    "integer"     "factor"    "integer"    "integer"
##       Street        Alley     LotShape  LandContour    Utilities
##     "factor"     "factor"     "factor"     "factor"     "factor"
##    LotConfig    LandSlope Neighborhood   Condition1   Condition2
##     "factor"     "factor"     "factor"     "factor"     "factor"
##     BldgType   HouseStyle  OverallQual  OverallCond    YearBuilt
##     "factor"     "factor"    "integer"    "integer"    "integer"
## YearRemodAdd    RoofStyle     RoofMatl  Exterior1st  Exterior2nd
##    "integer"     "factor"     "factor"     "factor"     "factor"
##   MasVnrType   MasVnrArea    ExterQual    ExterCond   Foundation
##     "factor"    "integer"     "factor"     "factor"     "factor"
##     BsmtQual     BsmtCond BsmtExposure BsmtFinType1   BsmtFinSF1
##     "factor"     "factor"     "factor"     "factor"    "integer"
```

```
##  BsmtFinType2    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##      "factor"     "integer"     "integer"     "integer"      "factor"
##     HeatingQC    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF
##      "factor"      "factor"      "factor"     "integer"     "integer"
##  LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##     "integer"     "integer"     "integer"     "integer"     "integer"
##      HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##     "integer"     "integer"     "integer"      "factor"     "integer"
##    Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##      "factor"     "integer"      "factor"      "factor"     "integer"
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##      "factor"     "integer"     "integer"      "factor"      "factor"
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##      "factor"     "integer"     "integer"     "integer"     "integer"
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##     "integer"     "integer"      "factor"      "factor"      "factor"
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##     "integer"     "integer"     "integer"      "factor"      "factor"
##     SalePrice
##     "integer"
res <- cor(train[,c("SalePrice","OpenPorchSF","YearRemodAdd")])
round(res, 2)
##              SalePrice OpenPorchSF YearRemodAdd
## SalePrice         1.00        0.32         0.51
## OpenPorchSF       0.32        1.00         0.23
## YearRemodAdd      0.51        0.23         1.00
```

Derive a correlation matrix for any three quantitative variables in the dataset

```
cm <- pairs(~SalePrice+LotArea+OverallQual,data=train)
```



Test the hypotheses that the correlations between each pairwise set of variables is 0 and provide an 80% confidence interval.

```
cor.test(train$SalePrice,train$OverallQual,conf.level=.8)
##
##  Pearson's product-moment correlation
##
## data:  train$SalePrice and train$OverallQual
## t = 49.364, df = 1458, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.7780752 0.8032204
## sample estimates:
##       cor
## 0.7909816
cor.test(train$SalePrice,train$LotArea,conf.level=.8)
##
##  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
## 80 percent confidence interval:
##  0.2323391 0.2947946
## sample estimates:
##       cor
## 0.2638434
cor.test(train$OverallQual,train$SalePrice,conf.level=.8)
##
##  Pearson's product-moment correlation
##
## data:  train$OverallQual and train$SalePrice
## t = 49.364, df = 1458, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.7780752 0.8032204
## sample estimates:
##       cor
## 0.7909816
cor.test(train$OverallQual,train$LotArea,conf.level=.8)
##
##  Pearson's product-moment correlation
##
## data:  train$OverallQual and train$LotArea
## t = 4.0629, df = 1458, p-value = 5.106e-05
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.07250156 0.13887424
## sample estimates:
##       cor
## 0.1058057
cor.test(train$LotArea,train$SalePrice,conf.level=.8)
##
##  Pearson's product-moment correlation
##
## data:  train$LotArea and train$SalePrice
## 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(train$LotArea,train$OverallQual,conf.level=.8)
##
##  Pearson's product-moment correlation
##
## data:  train$LotArea and train$OverallQual
## t = 4.0629, df = 1458, p-value = 5.106e-05
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.07250156 0.13887424
## sample estimates:
##       cor
## 0.1058057
```

Discuss the meaning of your analysis. Would you be worried about familywise error? Why or why not?

Using the cor.test function in R, we were able to reject the null hypothesis that the correlation is 0, with a high signifigance (p-value < 0.05). Due to the high level of signifigance, I would not be worried about familywise error.

Invert your correlation matrix from above.

```
pm <- solve(res)
```

Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix.

```
pm %*% res
##                SalePrice   OpenPorchSF  YearRemodAdd
## SalePrice     1.000000e+00 -5.551115e-17 -1.110223e-16
## OpenPorchSF  -1.387779e-17  1.000000e+00  0.000000e+00
## YearRemodAdd  1.110223e-16  0.000000e+00  1.000000e+00
```

```
res %*% pm
##                SalePrice   OpenPorchSF  YearRemodAdd
## SalePrice     1.000000e+00 -2.081668e-17  0.000000e+00
## OpenPorchSF  -2.775558e-17  1.000000e+00 -5.551115e-17
## YearRemodAdd  0.000000e+00 -1.387779e-17  1.000000e+00
```

Conduct LU decomposition on the matrix.

```
lu.decomposition(pm)
## $L
##            [,1]       [,2] [,3]
## [1,]  1.0000000  0.0000000    0
## [2,] -0.2119548  1.0000000    0
## [3,] -0.4591361 -0.2262976    1
##
## $U
##          [,1]       [,2]       [,3]
## [1,] 1.428114 -0.3026956 -0.6556985
## [2,] 0.000000  1.0539747 -0.2385120
## [3,] 0.000000  0.0000000  1.0000000
```

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

```
skews <- data.frame(colName = character(), skewness = numeric())
for(i in colnames(train)){
  if(class(train[,i])=="integer" & !is.na(skewness(train[,i]))){
  skews <- rbind(skews,data.frame(i,skewness(train[,i])))
  }
}
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
```

```
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
## Warning in Ops.factor(x, mean(x)): '-' not meaningful for factors
```

```
skews
```

```
##                i skewness.train...i..
## 1             Id           0.00000000
## 2     MSSubClass           1.40476562
## 3        LotArea          12.18261502
## 4    OverallQual           0.21649836
## 5    OverallCond           0.69164401
## 6      YearBuilt          -0.61220121
## 7   YearRemodAdd          -0.50252776
## 8     BsmtFinSF1           1.68204129
## 9     BsmtFinSF2           4.24652141
## 10     BsmtUnfSF           0.91837835
## 11   TotalBsmtSF           1.52112395
## 12     X1stFlrSF           1.37392896
## 13     X2ndFlrSF           0.81135997
## 14  LowQualFinSF           8.99283329
## 15     GrLivArea           1.36375364
## 16  BsmtFullBath           0.59484237
## 17  BsmtHalfBath           4.09497490
## 18      FullBath           0.03648647
## 19      HalfBath           0.67450925
## 20  BedroomAbvGr           0.21135511
## 21  KitchenAbvGr           4.47917826
## 22  TotRmsAbvGrd           0.67495173
## 23    Fireplaces           0.64823107
## 24    GarageCars          -0.34184538
## 25    GarageArea           0.17961125
## 26    WoodDeckSF           1.53820999
## 27   OpenPorchSF           2.35948572
## 28 EnclosedPorch           3.08352575
## 29    X3SsnPorch          10.28317840
## 30   ScreenPorch           4.11374731
## 31      PoolArea          14.79791829
## 32       MiscVal          24.42652237
## 33        MoSold           0.21161746
## 34        YrSold           0.09607079
## 35     SalePrice           1.87900860
```

```
train$LotArea.trans <- train$LotArea ^ 2
```

fit an exponential probability density function

```
lot.fit <- fitdistr(train$LotArea.trans, densfun="exponential")
```

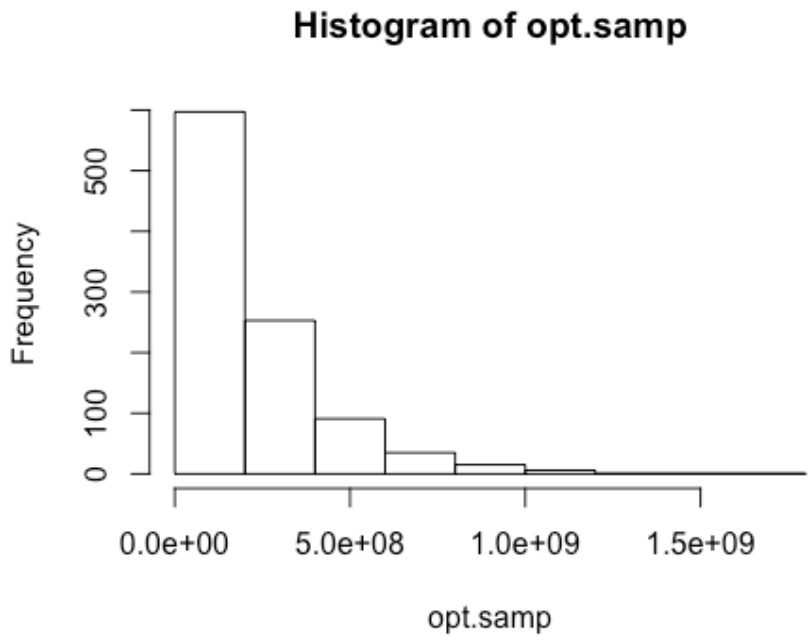
```
lot.fit <- fitdistr(train$LotArea.trans, densfun="exponential")
lot.fit.func <- function(x){lot.fit$estimate * exp(-lot.fit$estimate * x)}
```

Find the optimal value of lamba for this distribution, and then take 1000 samples from this exponential distribution using this value

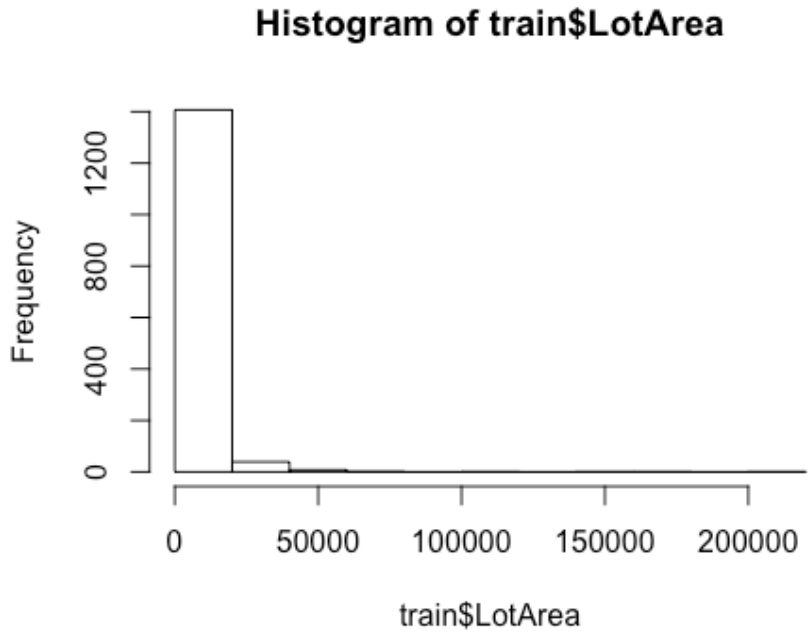
```
opt.lam <- optim(lot.fit$estimate,lot.fit.func)
## Warning in optim(lot.fit$estimate, lot.fit.func): one-dimensional optimization by Nelder-Mead is
unreliable:
## use "Brent" or optimize() directly
opt.samp <- rexp(1000,opt.lam$value)
```

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

```
hist(opt.samp)
```




```
hist(train$LotArea)
```



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

```
qexp(.05,rate = opt.lam$value)
## [1] 10779854
qexp(.05,rate = opt.lam$value, lower.tail = FALSE )
## [1] 629586347
```

```
## [1] 629586547
```

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

```
ci <- qt(1-(.95/2),df=length(train$LotArea)-1)*sd(train$LotArea)/sqrt(length(train$LotArea))
print(cbind(mean(train$LotArea)-ci,mean(train$LotArea)+ci))
##          [,1]     [,2]
## [1,] 10500.44 10533.21
```

provide the empirical 5th percentile and 95th percentile of the data

```
quantile(train$LotArea,.05)
##     5%
## 3311.7
quantile(train$LotArea,.95)
##      95%
## 17401.15
```

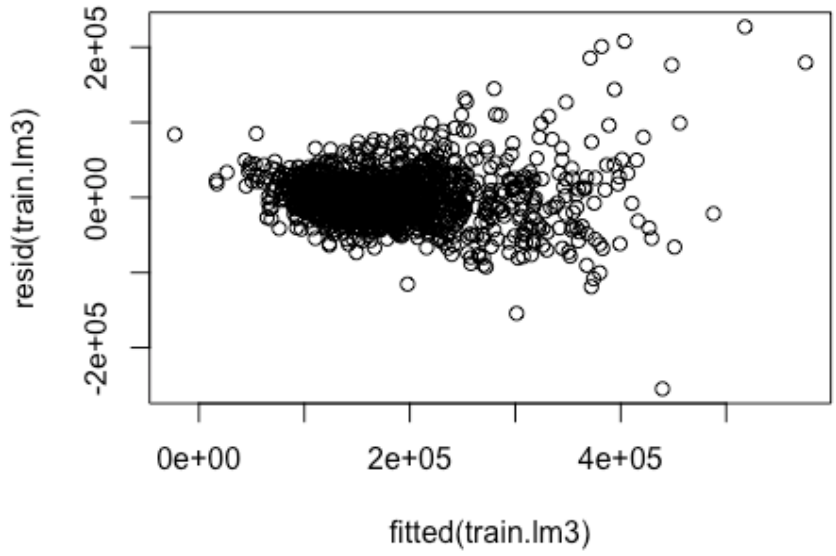
Build some type of multiple regression model

```
colSums(is.na(train))
##           Id    MSSubClass      MSZoning   LotFrontage       LotArea
##            0             0             0           259             0
##       Street         Alley     LotShape   LandContour     Utilities
##            0          1369             0             0             0
##    LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##            0             0             0             0             0
##     BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##            0             0             0             0             0
##  YearRemodAdd     RoofStyle     RoofMatl   Exterior1st   Exterior2nd
##            0             0             0             0             0
##   MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##            8             8             0             0             0
##     BsmtQual     BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##           37            37            38            37             0
##  BsmtFinType2    BsmtFinSF2    BsmtUnfSF   TotalBsmtSF       Heating
##           38             0             0             0             0
##    HeatingQC    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF
##            0             0             1             0             0
##  LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##            0             0             0             0             0
##     HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##            0             0             0             0             0
##   Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##            0             0           690            81            81
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##           81             0             0            81            81
##   PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##            0             0             0             0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##            0             0          1453          1179          1406
##      MiscVal       MoSold        YrSold      SaleType SaleCondition
##            0             0             0             0             0
##    SalePrice LotArea.trans
##            0             0
drops <- c("MiscFeature","Fence","PoolQC","FireplaceQu","Alley")
train<- train[ ,!(names(train) %in% drops)]

for(i in colnames(train)){
  train[,i][is.na(train[,i])] <- sample(train[,i][!is.na(train[,i])],length(train[,i][is.na(train[,i])]))
}

train.lm <- lm(SalePrice ~ ., data = na.omit(train))
#train.lm.stepped <- step(train.lm, direction = "backward", trace=FALSE )
#summary(train.lm.stepped)

#keeping only columns w high signifigance; eliminating redundant variables
(BsmtSF,GarageCond...)
keeps <- c("LotArea"
,"LandSlope"
,"Neighborhood"
,"Condition1"
,"Condition2"
,"OverallQual"
,"OverallCond"
,"YearBuilt"
,"RoofMatl"
```

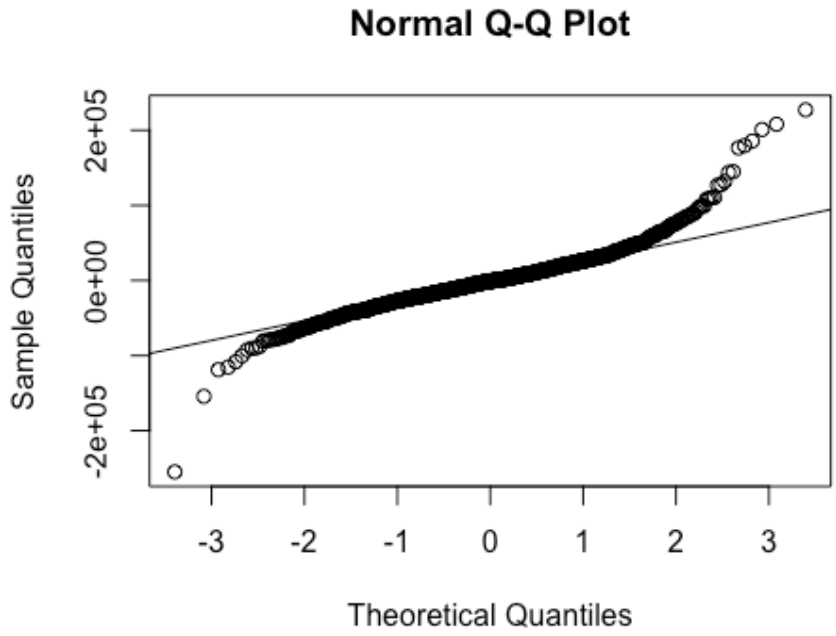
```
,"ExterQual"
,"BsmtQual"
,"TotalBsmtSF"
,"GarageQual"
,"PoolArea"
,"SaleCondition"
,"SalePrice"
)
train2 <- train[,keeps]
train.lm2 <- lm(SalePrice ~ ., data = na.omit(train2))
summary(train.lm2)
##
## Call:
## lm(formula = SalePrice ~ ., data = na.omit(train2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -163422  -18753   -1290   15890  232720
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -8.098e+05  1.621e+05  -4.995 6.64e-07 ***
## LotArea              1.208e+00  1.240e-01   9.738  < 2e-16 ***
## LandSlopeMod         1.729e+04  4.632e+03   3.732 0.000197 ***
## LandSlopeSev        -3.970e+04  1.347e+04  -2.948 0.003255 **
## NeighborhoodBlueste  1.092e+04  2.557e+04   0.427 0.669564
## NeighborhoodBrDale  -5.453e+03  1.250e+04  -0.436 0.662830
## NeighborhoodBrkSide  1.537e+04  1.076e+04   1.428 0.153495
## NeighborhoodClearCr  3.243e+04  1.145e+04   2.834 0.004670 **
## NeighborhoodCollgCr  1.980e+04  8.773e+03   2.257 0.024151 *
## NeighborhoodCrawfor  4.632e+04  1.055e+04   4.391 1.21e-05 ***
## NeighborhoodEdwards  9.292e+03  9.771e+03   0.951 0.341774
## NeighborhoodGilbert  2.748e+04  9.330e+03   2.945 0.003280 **
## NeighborhoodIDOTRR  -4.731e+02  1.144e+04  -0.041 0.967026
## NeighborhoodMeadowV  6.345e+03  1.223e+04   0.519 0.603890
## NeighborhoodMitchel  7.956e+02  1.009e+04   0.079 0.937184
## NeighborhoodNAmes    1.135e+04  9.407e+03   1.207 0.227705
## NeighborhoodNoRidge  1.045e+05  9.973e+03  10.482  < 2e-16 ***
## NeighborhoodNPkVill  7.020e+03  1.434e+04   0.490 0.624456
## NeighborhoodNridgHt  4.348e+04  9.433e+03   4.609 4.42e-06 ***
## NeighborhoodNWAmes   1.901e+04  9.753e+03   1.949 0.051460 .
## NeighborhoodOldTown  1.060e+04  1.045e+04   1.014 0.310723
## NeighborhoodSawyer   1.307e+04  1.001e+04   1.306 0.191734
## NeighborhoodSawyerW  2.727e+04  9.631e+03   2.831 0.004703 **
## NeighborhoodSomerst  2.486e+04  9.074e+03   2.739 0.006239 **
## NeighborhoodStoneBr  6.146e+04  1.080e+04   5.688 1.56e-08 ***
## NeighborhoodSWISU    2.390e+04  1.197e+04   1.997 0.046017 *
## NeighborhoodTimber   2.045e+04  1.025e+04   1.995 0.046270 *
## NeighborhoodVeenker  3.808e+04  1.343e+04   2.835 0.004644 **
## Condition1Feedr      5.430e+03  6.726e+03   0.807 0.419648
## Condition1Norm       7.424e+03  5.503e+03   1.349 0.177554
## Condition1PosA       2.578e+04  1.363e+04   1.892 0.058740 .
## Condition1PosN       2.652e+04  1.007e+04   2.633 0.008564 **
## Condition1RRAe      -1.162e+04  1.202e+04  -0.966 0.334155
## Condition1RRAn       4.712e+03  9.198e+03   0.512 0.608556
## Condition1RRNe       4.960e+03  2.507e+04   0.198 0.843220
## Condition1RRNn      -1.010e+04  1.672e+04  -0.604 0.546085
## Condition2Feedr     -6.507e+03  2.960e+04  -0.220 0.826050
## Condition2Norm      -2.981e+03  2.545e+04  -0.117 0.906767
## Condition2PosA       2.454e+04  4.303e+04   0.570 0.568546
## Condition2PosN      -1.854e+05  3.662e+04  -5.063 4.68e-07 ***
## Condition2RRAe       2.660e+04  4.276e+04   0.622 0.534020
## Condition2RRAn      -3.902e+04  4.278e+04  -0.912 0.361852
## Condition2RRNn      -1.120e+04  3.515e+04  -0.318 0.750155
## OverallQual          2.023e+04  1.175e+03  17.227  < 2e-16 ***
## OverallCond          5.272e+03  9.385e+02   5.618 2.33e-08 ***
## YearBuilt            1.767e+02  7.567e+01   2.335 0.019697 *
## RoofMatlCompShg      5.162e+05  3.909e+04  13.205  < 2e-16 ***
## RoofMatlMembran      5.434e+05  5.363e+04  10.132  < 2e-16 ***
## RoofMatlMetal        5.505e+05  5.426e+04  10.146  < 2e-16 ***
## RoofMatlRoll         5.274e+05  5.168e+04  10.205  < 2e-16 ***
## RoofMatlTar&Grv      5.197e+05  4.020e+04  12.930  < 2e-16 ***
## RoofMatlWdShake      5.385e+05  4.249e+04  12.675  < 2e-16 ***
## RoofMatlWdShngl      5.865e+05  4.105e+04  14.286  < 2e-16 ***
## ExterQualFa         -4.973e+04  1.231e+04  -4.039 5.65e-05 ***
## ExterQualGd         -4.308e+04  6.172e+03  -6.980 4.54e-12 ***
```

```
## ExterQualTA          -5.002e+04  6.890e+03  -7.260 6.41e-13 ***
## BsmtQualFa           -3.474e+04  7.856e+03  -4.422 1.05e-05 ***
## BsmtQualGd           -4.081e+04  4.261e+03  -9.579  < 2e-16 ***
## BsmtQualTA           -4.273e+04  5.005e+03  -8.537  < 2e-16 ***
## TotalBsmtSF           3.867e+01  2.897e+00  13.348  < 2e-16 ***
## GarageQualFa         -3.946e+04  1.847e+04  -2.137 0.032769 *
## GarageQualGd         -3.026e+04  1.989e+04  -1.522 0.128297
## GarageQualPo         -6.772e+04  2.687e+04  -2.521 0.011828 *
## GarageQualTA         -3.101e+04  1.771e+04  -1.751 0.080218 .
## PoolArea              1.406e+02  2.435e+01   5.776 9.42e-09 ***
## SaleConditionAdjLand  4.511e+03  1.788e+04   0.252 0.800788
## SaleConditionAlloca   1.501e+04  1.076e+04   1.396 0.163053
## SaleConditionFamily  -2.566e+02  8.426e+03  -0.030 0.975708
## SaleConditionNormal   8.009e+03  3.614e+03   2.216 0.026866 *
## SaleConditionPartial  2.463e+04  5.126e+03   4.805 1.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33890 on 1390 degrees of freedom
## Multiple R-squared:  0.8266, Adjusted R-squared:  0.818
## F-statistic: 96.05 on 69 and 1390 DF,  p-value: < 2.2e-16
drops2 <- c("Condition1","Condition2","YearBuilt","GarageQual","SaleCondition")
train3 <- train2[ ,!(names(train2) %in% drops2)]

train.lm3 <- lm(SalePrice ~ ., data = na.omit(train3))
summary(train.lm3)
##
## Call:
## lm(formula = SalePrice ~ ., data = na.omit(train3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -254848  -18954    -934   16337  227132
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -4.378e+05  4.277e+04 -10.237  < 2e-16 ***
## LotArea              1.140e+00  1.248e-01   9.133  < 2e-16 ***
## LandSlopeMod         1.705e+04  4.714e+03   3.617 0.000309 ***
## LandSlopeSev        -3.776e+04  1.367e+04  -2.762 0.005811 **
## NeighborhoodBlueste  3.739e+03  2.622e+04   0.143 0.886612
## NeighborhoodBrDale  -1.340e+04  1.275e+04  -1.051 0.293370
## NeighborhoodBrkSide  7.848e+02  1.032e+04   0.076 0.939413
## NeighborhoodClearCr  2.386e+04  1.164e+04   2.050 0.040545 *
## NeighborhoodCollgCr  1.707e+04  8.984e+03   1.900 0.057678 .
## NeighborhoodCrawfor  3.324e+04  1.025e+04   3.244 0.001208 **
## NeighborhoodEdwards -2.526e+03  9.762e+03  -0.259 0.795849
## NeighborhoodGilbert  2.493e+04  9.521e+03   2.618 0.008929 **
## NeighborhoodIDOTRR  -1.735e+04  1.100e+04  -1.577 0.114941
## NeighborhoodMeadowV -1.714e+03  1.246e+04  -0.138 0.890595
## NeighborhoodMitchel -4.711e+03  1.030e+04  -0.457 0.647474
## NeighborhoodNAmes    2.799e+03  9.472e+03   0.296 0.767620
## NeighborhoodNoRidge  9.790e+04  1.014e+04   9.651  < 2e-16 ***
## NeighborhoodNPkVill -7.983e+02  1.464e+04  -0.055 0.956530
## NeighborhoodNridgHt  4.366e+04  9.671e+03   4.514 6.88e-06 ***
## NeighborhoodNWAmes   1.356e+04  9.826e+03   1.380 0.167660
## NeighborhoodOldTown -7.200e+03  9.829e+03  -0.733 0.463969
## NeighborhoodSawyer   3.884e+03  1.007e+04   0.386 0.699875
## NeighborhoodSawyerW  2.024e+04  9.727e+03   2.081 0.037631 *
## NeighborhoodSomerst  2.577e+04  9.285e+03   2.775 0.005586 **
## NeighborhoodStoneBr  6.076e+04  1.106e+04   5.496 4.61e-08 ***
## NeighborhoodSWISU    7.545e+03  1.150e+04   0.656 0.511808
## NeighborhoodTimber   1.773e+04  1.049e+04   1.690 0.091185 .
## NeighborhoodVeenker  2.983e+04  1.368e+04   2.181 0.029349 *
## OverallQual          2.023e+04  1.178e+03  17.177  < 2e-16 ***
## OverallCond          5.375e+03  9.352e+02   5.748 1.11e-08 ***
## RoofMatlCompShg      4.849e+05  3.954e+04  12.264  < 2e-16 ***
## RoofMatlMembran      5.097e+05  5.449e+04   9.354  < 2e-16 ***
## RoofMatlMetal        5.210e+05  5.518e+04   9.443  < 2e-16 ***
## RoofMatlRoll         4.935e+05  5.279e+04   9.348  < 2e-16 ***
## RoofMatlTar&Grv      4.923e+05  4.055e+04  12.140  < 2e-16 ***
## RoofMatlWdShake      5.098e+05  4.286e+04  11.894  < 2e-16 ***
## RoofMatlWdShngl      5.626e+05  4.154e+04  13.545  < 2e-16 ***
## ExterQualFa         -4.676e+04  1.231e+04  -3.797 0.000152 ***
## ExterQualGd         -4.039e+04  6.178e+03  -6.538 8.71e-11 ***
```

```
## ExterQualTA        -4.966e+04  6.882e+03  -7.216 8.68e-13 ***
## BsmtQualFa         -4.609e+04  7.664e+03  -6.014 2.30e-09 ***
## BsmtQualGd         -4.363e+04  4.290e+03 -10.169  < 2e-16 ***
## BsmtQualTA         -4.826e+04  4.936e+03  -9.775  < 2e-16 ***
## TotalBsmtSF         3.828e+01  2.915e+00  13.132  < 2e-16 ***
## PoolArea            1.343e+02  2.442e+01   5.498 4.55e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34780 on 1415 degrees of freedom
## Multiple R-squared:  0.8141, Adjusted R-squared:  0.8083
## F-statistic: 140.8 on 44 and 1415 DF,  p-value: < 2.2e-16
plot(fitted(train.lm3),resid(train.lm3))
```




```
qqnorm(resid(train.lm3))
qqline(resid(train.lm3))
```




```
drops3 <- c("Neighborhood")
train4 <- train3[ ,!(names(train3) %in% drops3)]

train.lm4 <- lm(SalePrice ~ ., data = na.omit(train4))
summary(train.lm4)
##
```

```
## Call:
## lm(formula = SalePrice ~ ., data = na.omit(train4))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -312203  -20992   -1480   17348  279100
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.183e+05  4.431e+04 -11.698  < 2e-16 ***
## LotArea          1.322e+00  1.297e-01  10.190  < 2e-16 ***
## LandSlopeMod     1.986e+04  4.953e+03   4.009 6.41e-05 ***
## LandSlopeSev    -4.085e+04  1.463e+04  -2.793  0.00529 **
## OverallQual      2.468e+04  1.187e+03  20.792  < 2e-16 ***
## OverallCond      4.699e+03  9.656e+02   4.867 1.26e-06 ***
## RoofMatlCompShg  5.585e+05  4.234e+04  13.189  < 2e-16 ***
## RoofMatlMembran  5.859e+05  5.864e+04   9.992  < 2e-16 ***
## RoofMatlMetal    6.140e+05  5.920e+04  10.371  < 2e-16 ***
## RoofMatlRoll     5.686e+05  5.692e+04   9.989  < 2e-16 ***
## RoofMatlTar&Grv  5.594e+05  4.350e+04  12.859  < 2e-16 ***
## RoofMatlWdShake  5.698e+05  4.584e+04  12.431  < 2e-16 ***
## RoofMatlWdShngl  6.283e+05  4.465e+04  14.073  < 2e-16 ***
## ExterQualFa     -5.942e+04  1.307e+04  -4.547 5.91e-06 ***
## ExterQualGd     -4.072e+04  6.540e+03  -6.226 6.25e-10 ***
## ExterQualTA     -5.945e+04  7.207e+03  -8.248 3.59e-16 ***
## BsmtQualFa      -5.188e+04  7.994e+03  -6.490 1.18e-10 ***
## BsmtQualGd      -4.208e+04  4.451e+03  -9.455  < 2e-16 ***
## BsmtQualTA      -5.524e+04  5.047e+03 -10.945  < 2e-16 ***
## TotalBsmtSF      4.253e+01  3.014e+00  14.113  < 2e-16 ***
## PoolArea         1.326e+02  2.648e+01   5.007 6.21e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38010 on 1439 degrees of freedom
## Multiple R-squared:  0.7742, Adjusted R-squared:  0.771
## F-statistic: 246.7 on 20 and 1439 DF,  p-value: < 2.2e-16
test <- read.csv('/Users/samandleo/Downloads/test.csv')
keeps <- keeps[keeps != "SalePrice"]
test <- test[,c(keeps,"Id")]
test <- test[ , !(names(test) %in% drops2)]
for(i in colnames(test)){
  test[,i][is.na(test[,i])] <- sample(test[,i][!is.na(test[,i])],length(test[,i][is.na(test[,i])]))
}

test$SalePrice <- predict(train.lm3,test)
colnames(test)
##  [1] "LotArea"      "LandSlope"    "Neighborhood" "OverallQual"
##  [5] "OverallCond"  "RoofMatl"     "ExterQual"    "BsmtQual"
##  [9] "TotalBsmtSF"  "PoolArea"     "Id"           "SalePrice"
submission.scd <- subset(test,select=c("Id","SalePrice"))

write.csv(submission.scd,file="submission_scd.csv",row.names=FALSE)
```

Username: SamCD

Score: 0.20441