Computational Mathematics

# Computational Mathematics

Solutions should be provided in a format that can be shared on R Pubs and Git hub and You are also expected to make a short presentation via YouTube and post that recording to the board.

## Problem 1.

Using R, generate a random variable X that has 10,000 random uniform numbers from 1 to N, where N can be any number of your choosing greater than or equal to 6. Then generate a random variable Y that has 10,000 random normal numbers with a mean of (N+1)/2.

Answer:

Generate a random variable X

# set seed value  
set.seed(1)  
N <- 6  
X <- runif(10000, min = 1, max = N)

Generate a random variable Y

# mean   
mu <- (N+1)/2  
Y <- rnorm(10000 , mean = mu)

Probability. Calculate as a minimum the below probabilities a through c. Assume the small letter “x” is estimated as the median of the X variable, and the small letter “y” is estimated as the 1st quartile of the Y variable. Interpret the meaning of all probabilities.

### 5 points

#### a. P(X>x | X>y) = 0.7875256

Answer:

# first calculate x and y  
x <- median(X)  
y <- summary(Y)[2][[1]]  
  
  
#p(A|B) = P(AB)/P(B)  
sum(X>x & X > y)/sum(X>y)

## [1] 0.7875256

The probability of X greater than median value of X given that X is greater than first quartile of y is 0.78.

#### b. P(X>x, Y>y) = 0.3754

Answer:

#P(AB)  
pab <- sum(X>x & Y>y)/length(X)

The probability of X greater than median value of X and Y is greater than first quartile of y is 0.3754.

#### c. P(Xy) = 0.2124744

Answer:

#p(A|B) = P(AB)/P(B)  
sum(X<x & X > y)/sum(X>y)

## [1] 0.2124744

The probability of X less than median value of X given that X is greater than first quartile of y is 0.2124744.

### 5 points. 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.

Answer:

tab <- c(sum(X<x & Y < y),  
 sum(X < x & Y == y),  
 sum(X < x & Y > y))  
tab <- rbind(tab,  
 c(sum(X==x & Y < y),  
 sum(X == x & Y == y),  
 sum(X == x & Y > y))  
   
 )  
tab <- rbind(tab,  
 c(sum(X>x & Y < y),  
 sum(X > x & Y == y),  
 sum(X > x & Y > y))  
 )  
tab <- cbind(tab, tab[,1] + tab[,2] + tab[,3])  
tab <- rbind(tab, tab[1,] + tab[2,] + tab[3,])  
colnames(tab) <- c("Y<y", "Y=y", "Y>y", "Total")  
rownames(tab) <- c("X<x", "X=x", "X>x", "Total")  
knitr::kable(tab)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Y<y | Y=y | Y>y | Total |
| X<x | 1254 | 0 | 3746 | 5000 |
| X=x | 0 | 0 | 0 | 0 |
| X>x | 1246 | 0 | 3754 | 5000 |
| Total | 2500 | 0 | 7500 | 10000 |

We’ve made joint and marginal probability table. Now we’ll test the condition

# P(X>x and Y>y)  
3754/10000

## [1] 0.3754

#P(X>x)P(Y>y)  
((5000)/10000)\*(7500/10000)

## [1] 0.375

we can see that the condition holds since P(X>x and Y>y) = 0.3754 and P(X>x)P(Y>y) = 0.375 are approximately equal.

### 5 points. Check to see if independence holds by using Fisher’s Exact Test and the Chi Square Test. What is the difference between the two? Which is most appropriate?

Answer:

Fisher’s Exact Test

fisher.test(table(X>x,Y>y))

##   
## Fisher's Exact Test for Count Data  
##   
## data: table(X > x, Y > y)  
## p-value = 0.8716  
## alternative hypothesis: true odds ratio is not equal to 1  
## 95 percent confidence interval:  
## 0.9202847 1.1052820  
## sample estimates:  
## odds ratio   
## 1.00857

The p-value is greater than zero we don’t reject the null hypothesis. Two events are independent.

The Chi Square Test

chisq.test(table(X>x,Y>y))

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: table(X > x, Y > y)  
## X-squared = 0.026133, df = 1, p-value = 0.8716

The p-value is greeter than zero we don’t reject the null hypothesis. Two events are independent.

Fisher’s exact test the null of independence of rows and columns in a contingency table with fixed marginals.

Chi-squared test tests contingency table tests and goodness-of-fit tests.

Fisher’s exact test is appropriate here. Since the contingency table are fixed here in the table.

## 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.

Load the libraries

library(readr)  
library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.3.1   
## ✔ tibble 2.0.1 ✔ dplyr 0.8.0.1  
## ✔ tidyr 0.8.3 ✔ stringr 1.4.0   
## ✔ ggplot2 3.1.0 ✔ forcats 0.4.0

## ── Conflicts ─────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

Read the data

train <- read\_csv("train.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## Id = col\_double(),  
## MSSubClass = col\_double(),  
## LotFrontage = col\_double(),  
## LotArea = col\_double(),  
## OverallQual = col\_double(),  
## OverallCond = col\_double(),  
## YearBuilt = col\_double(),  
## YearRemodAdd = col\_double(),  
## MasVnrArea = col\_double(),  
## BsmtFinSF1 = col\_double(),  
## BsmtFinSF2 = col\_double(),  
## BsmtUnfSF = col\_double(),  
## TotalBsmtSF = col\_double(),  
## `1stFlrSF` = col\_double(),  
## `2ndFlrSF` = col\_double(),  
## LowQualFinSF = col\_double(),  
## GrLivArea = col\_double(),  
## BsmtFullBath = col\_double(),  
## BsmtHalfBath = col\_double(),  
## FullBath = col\_double()  
## # ... with 18 more columns  
## )

## See spec(...) for full column specifications.

test <- read\_csv("test.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## Id = col\_double(),  
## MSSubClass = col\_double(),  
## LotFrontage = col\_double(),  
## LotArea = col\_double(),  
## OverallQual = col\_double(),  
## OverallCond = col\_double(),  
## YearBuilt = col\_double(),  
## YearRemodAdd = col\_double(),  
## MasVnrArea = col\_double(),  
## BsmtFinSF1 = col\_double(),  
## BsmtFinSF2 = col\_double(),  
## BsmtUnfSF = col\_double(),  
## TotalBsmtSF = col\_double(),  
## `1stFlrSF` = col\_double(),  
## `2ndFlrSF` = col\_double(),  
## LowQualFinSF = col\_double(),  
## GrLivArea = col\_double(),  
## BsmtFullBath = col\_double(),  
## BsmtHalfBath = col\_double(),  
## FullBath = col\_double()  
## # ... with 17 more columns  
## )  
## See spec(...) for full column specifications.

### 5 points. Descriptive and Inferential Statistics.

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

Provide a scatter-plot matrix for at least two of the independent variables and the dependent variable.

Derive a correlation matrix for any three quantitative variables in the data-set.

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

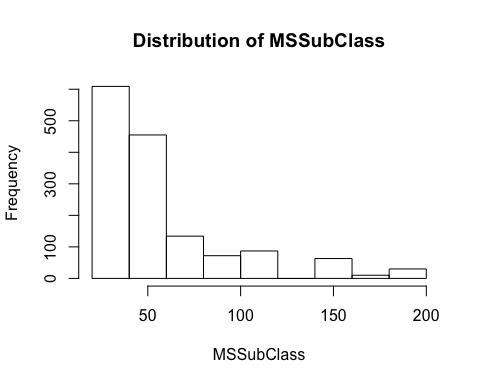
#### uni-variate descriptive statistics

summary(train)

## Id MSSubClass MSZoning LotFrontage   
## Min. : 1.0 Min. : 20.0 Length:1460 Min. : 21.00   
## 1st Qu.: 365.8 1st Qu.: 20.0 Class :character 1st Qu.: 59.00   
## Median : 730.5 Median : 50.0 Mode :character Median : 69.00   
## Mean : 730.5 Mean : 56.9 Mean : 70.05   
## 3rd Qu.:1095.2 3rd Qu.: 70.0 3rd Qu.: 80.00   
## Max. :1460.0 Max. :190.0 Max. :313.00   
## NA's :259   
## LotArea Street Alley LotShape   
## Min. : 1300 Length:1460 Length:1460 Length:1460   
## 1st Qu.: 7554 Class :character Class :character Class :character   
## Median : 9478 Mode :character Mode :character Mode :character   
## Mean : 10517   
## 3rd Qu.: 11602   
## Max. :215245   
##   
## LandContour Utilities LotConfig   
## Length:1460 Length:1460 Length:1460   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## LandSlope Neighborhood Condition1   
## Length:1460 Length:1460 Length:1460   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Condition2 BldgType HouseStyle OverallQual   
## Length:1460 Length:1460 Length:1460 Min. : 1.000   
## Class :character Class :character Class :character 1st Qu.: 5.000   
## Mode :character Mode :character Mode :character Median : 6.000   
## Mean : 6.099   
## 3rd Qu.: 7.000   
## Max. :10.000   
##   
## OverallCond YearBuilt YearRemodAdd RoofStyle   
## Min. :1.000 Min. :1872 Min. :1950 Length:1460   
## 1st Qu.:5.000 1st Qu.:1954 1st Qu.:1967 Class :character   
## Median :5.000 Median :1973 Median :1994 Mode :character   
## Mean :5.575 Mean :1971 Mean :1985   
## 3rd Qu.:6.000 3rd Qu.:2000 3rd Qu.:2004   
## Max. :9.000 Max. :2010 Max. :2010   
##   
## RoofMatl Exterior1st Exterior2nd   
## Length:1460 Length:1460 Length:1460   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## MasVnrType MasVnrArea ExterQual ExterCond   
## Length:1460 Min. : 0.0 Length:1460 Length:1460   
## Class :character 1st Qu.: 0.0 Class :character Class :character   
## Mode :character Median : 0.0 Mode :character Mode :character   
## Mean : 103.7   
## 3rd Qu.: 166.0   
## Max. :1600.0   
## NA's :8   
## Foundation BsmtQual BsmtCond   
## Length:1460 Length:1460 Length:1460   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2   
## Length:1460 Length:1460 Min. : 0.0 Length:1460   
## Class :character Class :character 1st Qu.: 0.0 Class :character   
## Mode :character Mode :character Median : 383.5 Mode :character   
## Mean : 443.6   
## 3rd Qu.: 712.2   
## Max. :5644.0   
##   
## BsmtFinSF2 BsmtUnfSF TotalBsmtSF Heating   
## Min. : 0.00 Min. : 0.0 Min. : 0.0 Length:1460   
## 1st Qu.: 0.00 1st Qu.: 223.0 1st Qu.: 795.8 Class :character   
## Median : 0.00 Median : 477.5 Median : 991.5 Mode :character   
## Mean : 46.55 Mean : 567.2 Mean :1057.4   
## 3rd Qu.: 0.00 3rd Qu.: 808.0 3rd Qu.:1298.2   
## Max. :1474.00 Max. :2336.0 Max. :6110.0   
##   
## HeatingQC CentralAir Electrical 1stFlrSF   
## Length:1460 Length:1460 Length:1460 Min. : 334   
## Class :character Class :character Class :character 1st Qu.: 882   
## Mode :character Mode :character Mode :character Median :1087   
## Mean :1163   
## 3rd Qu.:1391   
## Max. :4692   
##   
## 2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath   
## Min. : 0 Min. : 0.000 Min. : 334 Min. :0.0000   
## 1st Qu.: 0 1st Qu.: 0.000 1st Qu.:1130 1st Qu.:0.0000   
## Median : 0 Median : 0.000 Median :1464 Median :0.0000   
## Mean : 347 Mean : 5.845 Mean :1515 Mean :0.4253   
## 3rd Qu.: 728 3rd Qu.: 0.000 3rd Qu.:1777 3rd Qu.:1.0000   
## Max. :2065 Max. :572.000 Max. :5642 Max. :3.0000   
##   
## BsmtHalfBath FullBath HalfBath BedroomAbvGr   
## Min. :0.00000 Min. :0.000 Min. :0.0000 Min. :0.000   
## 1st Qu.:0.00000 1st Qu.:1.000 1st Qu.:0.0000 1st Qu.:2.000   
## Median :0.00000 Median :2.000 Median :0.0000 Median :3.000   
## Mean :0.05753 Mean :1.565 Mean :0.3829 Mean :2.866   
## 3rd Qu.:0.00000 3rd Qu.:2.000 3rd Qu.:1.0000 3rd Qu.:3.000   
## Max. :2.00000 Max. :3.000 Max. :2.0000 Max. :8.000   
##   
## KitchenAbvGr KitchenQual TotRmsAbvGrd Functional   
## Min. :0.000 Length:1460 Min. : 2.000 Length:1460   
## 1st Qu.:1.000 Class :character 1st Qu.: 5.000 Class :character   
## Median :1.000 Mode :character Median : 6.000 Mode :character   
## Mean :1.047 Mean : 6.518   
## 3rd Qu.:1.000 3rd Qu.: 7.000   
## Max. :3.000 Max. :14.000   
##   
## Fireplaces FireplaceQu GarageType GarageYrBlt   
## Min. :0.000 Length:1460 Length:1460 Min. :1900   
## 1st Qu.:0.000 Class :character Class :character 1st Qu.:1961   
## Median :1.000 Mode :character Mode :character Median :1980   
## Mean :0.613 Mean :1979   
## 3rd Qu.:1.000 3rd Qu.:2002   
## Max. :3.000 Max. :2010   
## NA's :81   
## GarageFinish GarageCars GarageArea GarageQual   
## Length:1460 Min. :0.000 Min. : 0.0 Length:1460   
## Class :character 1st Qu.:1.000 1st Qu.: 334.5 Class :character   
## Mode :character Median :2.000 Median : 480.0 Mode :character   
## Mean :1.767 Mean : 473.0   
## 3rd Qu.:2.000 3rd Qu.: 576.0   
## Max. :4.000 Max. :1418.0   
##   
## GarageCond PavedDrive WoodDeckSF OpenPorchSF   
## Length:1460 Length:1460 Min. : 0.00 Min. : 0.00   
## Class :character Class :character 1st Qu.: 0.00 1st Qu.: 0.00   
## Mode :character Mode :character Median : 0.00 Median : 25.00   
## Mean : 94.24 Mean : 46.66   
## 3rd Qu.:168.00 3rd Qu.: 68.00   
## Max. :857.00 Max. :547.00   
##   
## EnclosedPorch 3SsnPorch ScreenPorch PoolArea   
## Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.000   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.000   
## Median : 0.00 Median : 0.00 Median : 0.00 Median : 0.000   
## Mean : 21.95 Mean : 3.41 Mean : 15.06 Mean : 2.759   
## 3rd Qu.: 0.00 3rd Qu.: 0.00 3rd Qu.: 0.00 3rd Qu.: 0.000   
## Max. :552.00 Max. :508.00 Max. :480.00 Max. :738.000   
##   
## PoolQC Fence MiscFeature   
## Length:1460 Length:1460 Length:1460   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## MiscVal MoSold YrSold SaleType   
## Min. : 0.00 Min. : 1.000 Min. :2006 Length:1460   
## 1st Qu.: 0.00 1st Qu.: 5.000 1st Qu.:2007 Class :character   
## Median : 0.00 Median : 6.000 Median :2008 Mode :character   
## Mean : 43.49 Mean : 6.322 Mean :2008   
## 3rd Qu.: 0.00 3rd Qu.: 8.000 3rd Qu.:2009   
## Max. :15500.00 Max. :12.000 Max. :2010   
##   
## SaleCondition SalePrice   
## Length:1460 Min. : 34900   
## Class :character 1st Qu.:129975   
## Mode :character Median :163000   
## Mean :180921   
## 3rd Qu.:214000   
## Max. :755000   
##

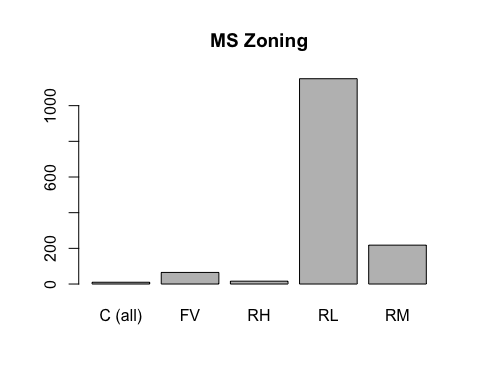
#### Plots

hist(train$MSSubClass, main="Distribution of MSSubClass",xlab="MSSubClass")

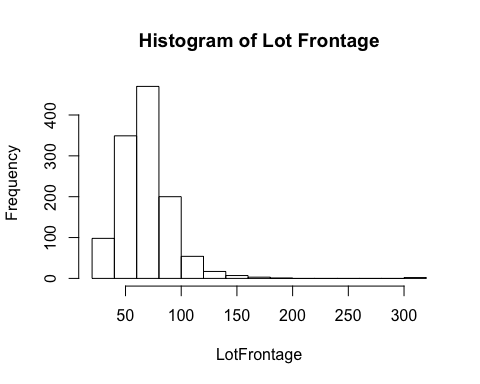


MSSubClass is left skewed.

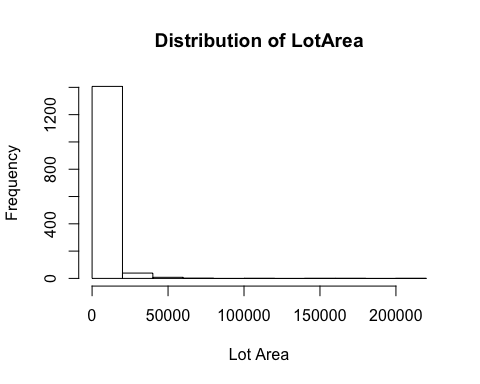
barplot(table(train$MSZoning), main="MS Zoning")

 RL has the highest frequency , C lowest frequency.

hist(train$LotFrontage,main="Histogram of Lot Frontage",xlab="LotFrontage")

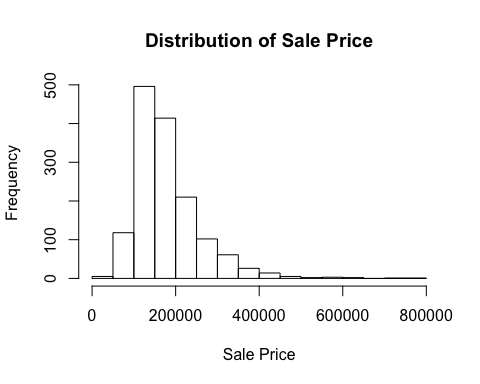
 LotFrontage is left skewed.

hist(train$LotArea,main="Distribution of LotArea",xlab="Lot Area")



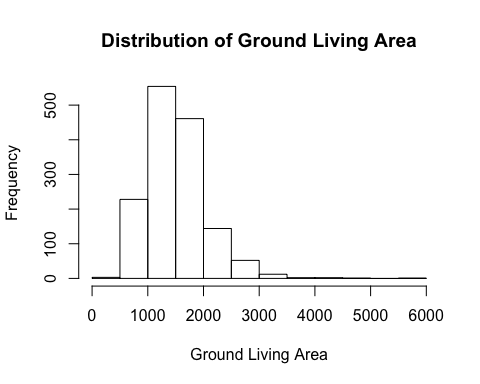
Lot Area is left skewed with very high small values.

hist(train$SalePrice,main="Distribution of Sale Price",xlab="Sale Price")



Sales price is slightly approximately normally distributed. .

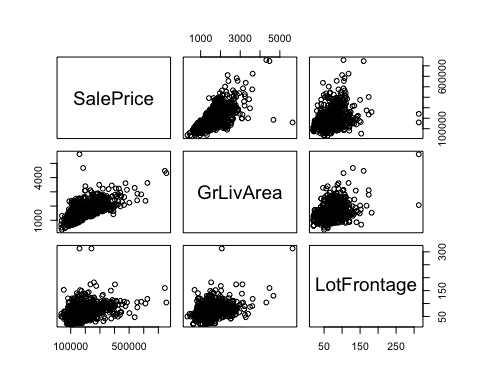
hist(train$GrLivArea,main="Distribution of Ground Living Area",xlab="Ground Living Area")



Ground Living Area is approximately normally distributed.

#### Scatterplot matrix for “SalePrice”,“GrLivArea”,“LotFrontage”

pairs(train[,c("SalePrice","GrLivArea","LotFrontage")])



From the scatter plot we can see that GrLiveArea and LotFrontage are positively correlated with Sale Price.

#### Correlation matrix for any three quantitative variables

SalePrice , GrLivArea and TotalBsmtSF

cormat <- cor(train[,c("SalePrice","GrLivArea","TotalBsmtSF")])  
cormat

## SalePrice GrLivArea TotalBsmtSF  
## SalePrice 1.0000000 0.7086245 0.6135806  
## GrLivArea 0.7086245 1.0000000 0.4548682  
## TotalBsmtSF 0.6135806 0.4548682 1.0000000

SalePrice shows strong positive correlation with GrLivArea and moderate correlation with TotalBsmTSF.

GrLivArea shows Strong positive correlation with SalePrice and weak positive correlation with TotalBsmSF.

TotalBsmSF shows moderate positive correlation with SalePrice and weak positive correlation with GrLivArea.

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

Null Hypothesis: The correlation between GrLivArea and SalePrice is 0 Alternative Hypothesis: The correlation between GrLivArea and SalePrice is other than 0

cor.test(train$SalePrice, train$GrLivArea, conf.level = 0.8)

##   
## Pearson's product-moment correlation  
##   
## data: train$SalePrice and train$GrLivArea  
## t = 38.348, df = 1458, p-value < 0.00000000000000022  
## alternative hypothesis: true correlation is not equal to 0  
## 80 percent confidence interval:  
## 0.6915087 0.7249450  
## sample estimates:  
## cor   
## 0.7086245

Since the the p value of the test is less than 0.05 at 5% level of significance we reject the null hypothesis and conclude that the correlation between GrLivArea and SalePrice is other than 0. 80 percent confidence interval of the test is 0.6939620 0.7285864

Null Hypothesis: The correlation between TotalBsmtSF and SalePrice is 0 Alternative Hypothesis: The correlation between TotalBsmtSF and SalePrice is other than 0

cor.test(train$SalePrice, train$TotalBsmtSF, conf.level = 0.8)

##   
## Pearson's product-moment correlation  
##   
## data: train$SalePrice and train$TotalBsmtSF  
## t = 29.671, df = 1458, p-value < 0.00000000000000022  
## alternative hypothesis: true correlation is not equal to 0  
## 80 percent confidence interval:  
## 0.5922142 0.6340846  
## sample estimates:  
## cor   
## 0.6135806

Since the the p value of the test is less than 0.05 at 5% level of significance we reject the null hypothesis and conclude that the correlation between TotalBsmtSF and SalePrice is other than 0.

80 percent confidence interval of the test is 0.5792077 0.6239328

Null Hypothesis: The correlation between TotalBsmtSF and GrLivArea is 0 Alternative Hypothesis: The correlation between TotalBsmtSF and GrLivArea is other than 0

cor.test(train$GrLivArea, train$TotalBsmtSF, conf.level = 0.8)

##   
## Pearson's product-moment correlation  
##   
## data: train$GrLivArea and train$TotalBsmtSF  
## t = 19.503, df = 1458, p-value < 0.00000000000000022  
## alternative hypothesis: true correlation is not equal to 0  
## 80 percent confidence interval:  
## 0.4278380 0.4810855  
## sample estimates:  
## cor   
## 0.4548682

Since the the p value of the test is less than 0.05 at 5% level of significance we reject the null hypothesis and conclude that the correlation between GrLivArea and TotalBsmtSF is other than 0.

80 percent confidence interval of the test is 0.4327076 0.4879552

#### family wise error

FWE <- 1 - (1 - .05)^2   
FWE

## [1] 0.0975

There is a 9.75% chance of type 1 error. Since the chance is low I will not be worried for family wise error .

### 5 points. Linear Algebra and Correlation. Invert your 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.

Answer:

# find inverse  
precision\_mat <- solve(cormat)  
  
# Multiply the correlation matrix by the precision matrix  
cor\_prec <- cormat %\*% precision\_mat  
cor\_prec

## SalePrice GrLivArea  
## SalePrice 1.00000000000000022204460 -0.00000000000000002081668  
## GrLivArea 0.00000000000000005551115 1.00000000000000000000000  
## TotalBsmtSF 0.00000000000000000000000 0.00000000000000005551115  
## TotalBsmtSF  
## SalePrice 0.0000000000000000000000  
## GrLivArea 0.0000000000000001110223  
## TotalBsmtSF 1.0000000000000000000000

# multiply the precision matrix by the correlation matrix  
prec\_cor <- precision\_mat %\*% cormat  
prec\_cor

## SalePrice GrLivArea  
## SalePrice 0.9999999999999997779554 -0.0000000000000001665335  
## GrLivArea 0.0000000000000002012279 1.0000000000000004440892  
## TotalBsmtSF 0.0000000000000000000000 0.0000000000000001110223  
## TotalBsmtSF  
## SalePrice -0.0000000000000001110223  
## GrLivArea 0.0000000000000001665335  
## TotalBsmtSF 1.0000000000000000000000

# LU Decomposistion  
library(pracma)

##   
## Attaching package: 'pracma'

## The following object is masked from 'package:purrr':  
##   
## cross

lu(cormat)

## $L  
## SalePrice GrLivArea TotalBsmtSF  
## SalePrice 1.0000000 0.00000000 0  
## GrLivArea 0.7086245 1.00000000 0  
## TotalBsmtSF 0.6135806 0.04031325 1  
##   
## $U  
## SalePrice GrLivArea TotalBsmtSF  
## SalePrice 1 0.7086245 0.6135806  
## GrLivArea 0 0.4978513 0.0200700  
## TotalBsmtSF 0 0.0000000 0.6227098

#### 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  for this distribution, and then take 1000 samples from this exponential distribution using this value (e.g., rexp(1000, )). 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.

Answer: We select LotArea as it’s skewed to the right.

#### optimal value of exponential for this distribution

library(MASS)

##   
## Attaching package: 'MASS'

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

# Fitting of univariate distribution  
(fd <- fitdistr(train$LotArea, "exponential"))

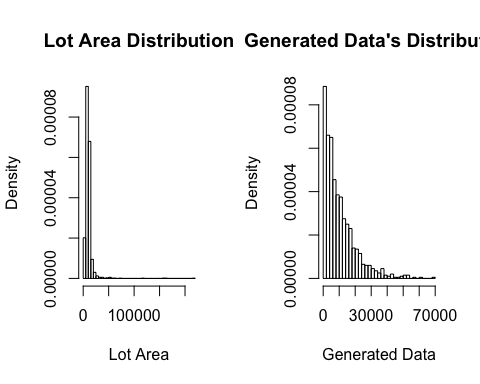
## rate   
## 0.000095085704   
## (0.000002488507)

# optimam value of lambda  
fd$estimate

## rate   
## 0.0000950857

#### 1000 samples from this exponential distribution using this value

values <- rexp(1000, rate = fd$estimate)  
par(mfrow=c(1,2))  
# Actual vs simulated distribution  
hist(train$LotArea, breaks=40, prob=TRUE, xlab="Lot Area",  
 main="Lot Area Distribution")  
hist(values, breaks=40, prob=TRUE, xlab="Generated Data",  
 main="Generated Data's Distribution")



From the two plots we can see that our Lot Area approximately fits a exponential distribution. The fit isn’t very well here.

#### 5th and 95th percentiles using the cumulative distribution function (CDF)

Fn <- ecdf(values)  
values[Fn(values)==0.05]

## [1] 402.4144

values[Fn(values)==0.95]

## [1] 30176.06

5% is 651.0724 and 95% is 31118.42

#### 95% confidence interval from the empirical data

t.test(values)$conf.int

## [1] 9667.527 10915.180  
## attr(,"conf.level")  
## [1] 0.95

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

t.test(train$LotArea)$conf.int

## [1] 10004.42 11029.24  
## attr(,"conf.level")  
## [1] 0.95

### 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.

Answer:

#### Model Summary:

For building model I’ve removed the variables with very large number of missing values. Then recoded the categorical variables to numerical variable. After that I’ve fitted a multiple regression model. After fitting the multiple regression model I’ve used step wise regression to select best set of predictor variables.

#### Final Model:

Based on our final model model’s R squared value is 0.8373. It’s a good fitted model. The assumptions of multiple linear regression are satisfied here.

#### We’ll see number of missing values in the variables

sapply(train, function(x){sum(is.na(x))})

## 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 1stFlrSF 2ndFlrSF   
## 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 3SsnPorch   
## 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   
## 0

We’ll remove variables having a large number of missing values. we’ll also remove irremovable,YearBuilt

train <-train[, !colnames(train) %in% c("Id","Alley","PoolQC","Fence","MiscFeature","FireplaceQu","LotFrontage","YearBuilt","YearRemodAdd")]  
  
test <- test[, !colnames(test) %in% c("Alley","PoolQC","Fence","MiscFeature","FireplaceQu","LotFrontage","YearBuilt","YearRemodAdd")]  
  
# convert categorical to numeric  
  
train <- train%>%  
 mutate\_if(is.character, as.factor)%>%  
 mutate\_if(is.factor, as.integer)  
  
test <- test %>%  
 mutate\_if(is.character, as.factor)%>%  
 mutate\_if(is.factor, as.integer)

Now we’ll take only complete cases

train <- na.omit(train)

#### We’ll not fit a multiple regression model

model\_fit <- lm(SalePrice~., data = train)  
summary(model\_fit)

##   
## Call:  
## lm(formula = SalePrice ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -434183 -13770 -848 12990 292689   
##   
## Coefficients: (2 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2169226.04952 1405502.84800 1.543 0.122988   
## MSSubClass -151.76908 50.65553 -2.996 0.002788 \*\*   
## MSZoning -2016.34133 1642.98609 -1.227 0.219959   
## LotArea 0.37915 0.11051 3.431 0.000621 \*\*\*  
## Street 37981.07458 16087.18883 2.361 0.018379 \*   
## LotShape -1327.90285 698.94808 -1.900 0.057678 .   
## LandContour 4069.04766 1487.17385 2.736 0.006304 \*\*   
## Utilities -50137.11987 34166.29186 -1.467 0.142503   
## LotConfig 169.82021 577.93533 0.294 0.768929   
## LandSlope 5837.59431 4148.29648 1.407 0.159605   
## Neighborhood 212.15266 169.02832 1.255 0.209663   
## Condition1 -518.49429 1065.63381 -0.487 0.626655   
## Condition2 -7975.19606 3451.46842 -2.311 0.021011 \*   
## BldgType -251.46567 1595.80939 -0.158 0.874814   
## HouseStyle -913.21759 691.12993 -1.321 0.186626   
## OverallQual 12660.37979 1291.66460 9.802 < 0.0000000000000002 \*\*\*  
## OverallCond 4053.20893 1037.82540 3.905 0.000098993057 \*\*\*  
## RoofStyle 2449.61702 1196.33409 2.048 0.040805 \*   
## RoofMatl 4248.28868 1598.10003 2.658 0.007952 \*\*   
## Exterior1st -945.09263 570.40793 -1.657 0.097793 .   
## Exterior2nd 310.63359 511.62505 0.607 0.543860   
## MasVnrType 4267.07351 1640.62044 2.601 0.009406 \*\*   
## MasVnrArea 30.06517 6.34084 4.742 0.000002361198 \*\*\*  
## ExterQual -8571.34766 2120.40336 -4.042 0.000056111669 \*\*\*  
## ExterCond 597.64250 1417.80253 0.422 0.673442   
## Foundation 2878.41233 1843.28975 1.562 0.118641   
## BsmtQual -8897.77673 1528.18705 -5.822 0.000000007341 \*\*\*  
## BsmtCond 2973.00730 1486.52840 2.000 0.045717 \*   
## BsmtExposure -3659.85244 936.09504 -3.910 0.000097300199 \*\*\*  
## BsmtFinType1 -1130.83469 677.53469 -1.669 0.095356 .   
## BsmtFinSF1 9.14935 6.23033 1.469 0.142212   
## BsmtFinType2 564.82759 1430.73675 0.395 0.693071   
## BsmtFinSF2 11.64129 10.08717 1.154 0.248689   
## BsmtUnfSF 2.58118 6.12091 0.422 0.673316   
## TotalBsmtSF NA NA NA NA   
## Heating -5278.08873 6159.10500 -0.857 0.391631   
## HeatingQC -899.49739 655.14726 -1.373 0.170004   
## CentralAir 3877.65021 5407.64455 0.717 0.473464   
## Electrical -77.04228 1026.39533 -0.075 0.940178   
## `1stFlrSF` 44.49957 7.01697 6.342 0.000000000315 \*\*\*  
## `2ndFlrSF` 46.22219 5.03502 9.180 < 0.0000000000000002 \*\*\*  
## LowQualFinSF 19.24862 23.19353 0.830 0.406744   
## GrLivArea NA NA NA NA   
## BsmtFullBath 7241.53771 2609.58652 2.775 0.005602 \*\*   
## BsmtHalfBath 1486.63102 4071.52125 0.365 0.715076   
## FullBath 4054.68437 2879.04753 1.408 0.159275   
## HalfBath -104.56228 2686.93015 -0.039 0.968964   
## BedroomAbvGr -4225.53750 1839.34309 -2.297 0.021763 \*   
## KitchenAbvGr -21272.85025 6426.10647 -3.310 0.000958 \*\*\*  
## KitchenQual -8834.97591 1558.62651 -5.668 0.000000017823 \*\*\*  
## TotRmsAbvGrd 3197.02629 1257.66722 2.542 0.011139 \*   
## Functional 4095.04411 1050.43599 3.898 0.000102 \*\*\*  
## Fireplaces 3738.10903 1775.10858 2.106 0.035414 \*   
## GarageType 277.51547 667.54160 0.416 0.677680   
## GarageYrBlt -43.65696 70.67950 -0.618 0.536901   
## GarageFinish -621.85098 1534.73606 -0.405 0.685410   
## GarageCars 14558.02658 2937.92370 4.955 0.000000820201 \*\*\*  
## GarageArea -1.21827 10.02204 -0.122 0.903268   
## GarageQual -157.82834 1843.19045 -0.086 0.931776   
## GarageCond 1793.55971 2139.21278 0.838 0.401953   
## PavedDrive 4022.71154 2478.52216 1.623 0.104832   
## WoodDeckSF 18.87330 7.87463 2.397 0.016687 \*   
## OpenPorchSF -15.90478 15.54537 -1.023 0.306446   
## EnclosedPorch -6.42617 16.35079 -0.393 0.694372   
## `3SsnPorch` 22.97673 30.16114 0.762 0.446322   
## ScreenPorch 43.64636 16.46591 2.651 0.008132 \*\*   
## PoolArea -26.16388 22.58834 -1.158 0.246963   
## MiscVal 0.04267 1.81403 0.024 0.981235   
## MoSold -155.06018 342.01553 -0.453 0.650359   
## YrSold -1044.20540 696.10119 -1.500 0.133843   
## SaleType -526.77360 605.10830 -0.871 0.384168   
## SaleCondition 2684.26064 921.03004 2.914 0.003626 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 32460 on 1268 degrees of freedom  
## Multiple R-squared: 0.8396, Adjusted R-squared: 0.8308   
## F-statistic: 96.17 on 69 and 1268 DF, p-value: < 0.00000000000000022

#### We’ll now do a stepwise regression based on ACI criterion

step\_model <- step(model\_fit, trace = 0)  
summary(step\_model)

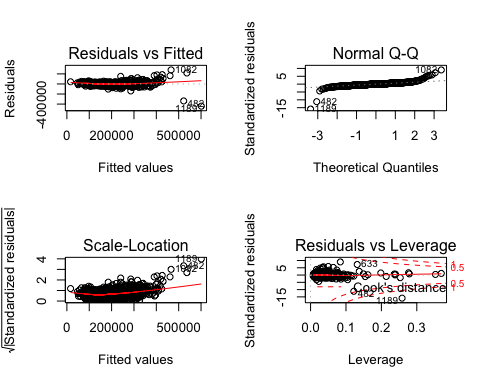
##   
## Call:  
## lm(formula = SalePrice ~ MSSubClass + MSZoning + LotArea + Street +   
## LotShape + LandContour + LandSlope + Condition2 + HouseStyle +   
## OverallQual + OverallCond + RoofStyle + RoofMatl + Exterior1st +   
## MasVnrType + MasVnrArea + ExterQual + Foundation + BsmtQual +   
## BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + `1stFlrSF` +   
## `2ndFlrSF` + BsmtFullBath + FullBath + BedroomAbvGr + KitchenAbvGr +   
## KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + GarageCars +   
## PavedDrive + WoodDeckSF + ScreenPorch + SaleCondition, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -441172 -13550 -957 13442 278991   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -68655.8503 38693.5272 -1.774 0.076239 .   
## MSSubClass -159.8789 27.4961 -5.815 0.000000007641103 \*\*\*  
## MSZoning -2503.0986 1534.7315 -1.631 0.103139   
## LotArea 0.3814 0.1060 3.597 0.000333 \*\*\*  
## Street 40806.0476 15627.6939 2.611 0.009128 \*\*   
## LotShape -1310.5266 669.9525 -1.956 0.050662 .   
## LandContour 3914.9209 1436.8188 2.725 0.006522 \*\*   
## LandSlope 6115.5394 4036.2642 1.515 0.129978   
## Condition2 -7336.6650 3325.0186 -2.207 0.027523 \*   
## HouseStyle -1224.6140 616.7163 -1.986 0.047277 \*   
## OverallQual 12959.5731 1248.8115 10.378 < 0.0000000000000002 \*\*\*  
## OverallCond 4247.8548 928.1763 4.577 0.000005179174355 \*\*\*  
## RoofStyle 2632.4405 1158.2961 2.273 0.023208 \*   
## RoofMatl 4131.9524 1555.7715 2.656 0.008007 \*\*   
## Exterior1st -614.3119 301.9824 -2.034 0.042128 \*   
## MasVnrType 4335.1776 1582.5650 2.739 0.006241 \*\*   
## MasVnrArea 30.2912 6.1226 4.947 0.000000850306146 \*\*\*  
## ExterQual -8542.1790 2043.1130 -4.181 0.000030972233573 \*\*\*  
## Foundation 3189.2880 1670.1013 1.910 0.056400 .   
## BsmtQual -8946.3960 1491.0981 -6.000 0.000000002557006 \*\*\*  
## BsmtCond 3202.5638 1404.1962 2.281 0.022727 \*   
## BsmtExposure -3678.7800 901.4594 -4.081 0.000047592852072 \*\*\*  
## BsmtFinType1 -1168.7326 649.9609 -1.798 0.072384 .   
## BsmtFinSF1 5.8341 3.1549 1.849 0.064652 .   
## `1stFlrSF` 45.5519 4.8433 9.405 < 0.0000000000000002 \*\*\*  
## `2ndFlrSF` 45.2873 4.1751 10.847 < 0.0000000000000002 \*\*\*  
## BsmtFullBath 7523.9163 2355.1435 3.195 0.001434 \*\*   
## FullBath 4197.5166 2518.4225 1.667 0.095810 .   
## BedroomAbvGr -4439.2566 1771.3141 -2.506 0.012325 \*   
## KitchenAbvGr -21663.3923 6032.6496 -3.591 0.000342 \*\*\*  
## KitchenQual -8746.4950 1523.4941 -5.741 0.000000011699055 \*\*\*  
## TotRmsAbvGrd 3456.5692 1200.5186 2.879 0.004052 \*\*   
## Functional 3843.4812 1016.0138 3.783 0.000162 \*\*\*  
## Fireplaces 4035.7108 1688.6046 2.390 0.016992 \*   
## GarageCars 14425.7154 1972.8288 7.312 0.000000000000458 \*\*\*  
## PavedDrive 4949.3592 2348.8421 2.107 0.035296 \*   
## WoodDeckSF 18.4401 7.5978 2.427 0.015359 \*   
## ScreenPorch 41.4813 15.9000 2.609 0.009188 \*\*   
## SaleCondition 2596.0989 873.9699 2.970 0.003028 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 32290 on 1299 degrees of freedom  
## Multiple R-squared: 0.8373, Adjusted R-squared: 0.8326   
## F-statistic: 176 on 38 and 1299 DF, p-value: < 0.00000000000000022

#### Our Final model

R squared values 0.8373 indicates that our model is a very good model. Our fitted multiple regression model is 83.73% accurate in predicting Sales price based on the dependent variables. Since the F tests p value less than 0.05 at 5% level of significance our model is a valid model.

#### Residual Analysis

par(mfrow=c(2,2))  
plot(step\_model)



From the residuals plot we can see that the assuptions of multiple regression model are satisfied. The residuals are approximately normally distributed. There is not heteroscedacity and pattern in the residuals. Do prediction

predicted <- predict(step\_model, test)  
sub <- data.frame(Id = test$Id, predicted)  
write.csv(sub,"submission.csv")

#### Kaggle Submission

Kaggle username is **kaggle\_username**. Final score is **kaggle user name**.