605 Final Exam

Jen Abinette

2023-05-13

Problem 1

Probability Density 1: X ~ Gamma

Using R, generate a random variable X that has 10,000 random Gamma pdf values. A Gamma pdf is completely described by n (a size parameter) and lambda (a shape parameter). Choose any n greater 3 and an expected value (lambda) between 2 and 10 (you choose).

```
set.seed(1)
n <- 4
l <- 4
X <- rgamma(n=10000, shape=n, rate=l)
```

Probability Density 2: Y ~ Sum of Exponentials

Then generate 10,000 observations from the sum of n exponential pdfs with rate/shape parameter (lambda). The n and lambda must be the same as in the previous case.

```
set.seed(1)
Y <- ( rexp(10000,l) + rexp(10000,l) + rexp(10000,l) + rexp(10000,l) )
```

Probability Density 3: Z ~ Exponential

Then generate 10,000 observations from a single exponential pdf with rate/shape parameter (lambda).

```
set.seed(1)
Z <- rexp(10000,l)
```

1a. Calculate the empirical expected value (means) and variances of all three pdfs.

```
paste("For X: Mean is ",round(mean(X),4)," and Variance is ",round(var(X),4))
```

```
## [1] "For X: Mean is  1.001  and Variance is  0.2494"
```

```r
paste("For Y: Mean is ",round(mean(Y),4)," and Variance is ",round(var(Y),4))
```

```
## [1] "For Y: Mean is  0.998  and Variance is  0.2497"
```

```r
paste("For Z: Mean is ",round(mean(Z),4)," and Variance is ",round(var(Z),4))
```

```
## [1] "For Z: Mean is  0.2496  and Variance is  0.0645"
```

1b.

Using calculus, calculate the expected value and variance of the Gamma pdf (X)

$$E(X) = \frac{\alpha}{\beta} = \frac{n}{l} = \frac{4}{4} = 1 \qquad OR \qquad E(X) = \int x \times f(x)dx$$

```r
integrand <- function(x) x * dgamma(x, shape=n, rate=l)
exp_value <- integrate(integrand, lower = 0, upper = Inf)$value
print(exp_value)
```

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

$$V(X) = \frac{\alpha}{\beta^2} = \frac{n}{l^2} = \frac{4}{4^2} = \frac{1}{4} \qquad OR \qquad V(X) = E(X^2) - E(X)^2$$

```r
integrand <- function(x) x^2 * dgamma(x, shape=n, rate=l)
print( integrate(integrand, lower = 0, upper = Inf)$value - exp_value^2 )
```

```
## [1] 0.25
```

Using the moment generating function for exponentials, calculate the expected value of the single exponential (Z) and the sum of exponentials (Y)

$$M(t) = \frac{\lambda}{\lambda - t}$$

Expected Value of Z ~ Single Exponential:

$$M'(t) = \frac{\lambda}{(\lambda - t)^2} \qquad E(Z) = M'(0) = \frac{\lambda}{(\lambda - 0)^2} = \frac{1}{\lambda}$$

$$E(Z) = \frac{1}{\lambda} = \frac{1}{4}$$

Expected Value of Y ~ Sum of n Exponentials: Given Y is the sum of n exponential pdfs and lambda is equal for Y and Z then

$$E(Y) = 4 \times E(Z) = 4 \times \frac{1}{4} = 1$$

1c-e. Probability. For pdf Z (the exponential), calculate empirically probabilities c through c. Then evaluate through calculus whether the memoryless property holds.

The memoryless property applies to the exponential and geometric probability distributions as the probability of a future outcome is unaffected by past outcomes such as flipping a coin or rolling a dice. This means that the probability of event A is not affected by event B so we need only find the probability A. P(A|B) = P(A)

$$P(Z > \lambda | Z > \lambda/2)$$

```
1-pexp(l)
```

```
## [1] 0.01831564
```

$$P(Z > 2\lambda | Z > \lambda)$$

```
1-pexp(2*l)
```

```
## [1] 0.0003354626
```

$$P(Z > 3\lambda | Z > \lambda)$$

```
1-pexp(3*l)
```

```
## [1] 6.144212e-06
```

Loosely investigate whether P(YZ) = P(Y) P(Z) by building a table with quartiles and evaluating the marginal and joint probabilities.

```
qY = quantile(Y, probs = c(.25,.5,.75,1))
qY
```

```
##       25%       50%       75%      100%
## 0.6302107 0.9127258 1.2775591 3.8958482
```

```
qZ = quantile(Z, probs = c(.25,.5,.75,1))
qZ
```

```
##        25%        50%        75%       100%
## 0.07025417 0.17366343 0.34415375 2.29611286
```

```r
# Build Empty Table to be filled
table <- matrix(0, nrow = 5, ncol = 5)
colnames(table) <- c("1st Y", "2nd Y", "3rd Y", "4th Y", "Sum")
rownames(table) <- c("1st Z", "2nd Z", "3rd Z", "4th Z", "Sum")

# Add Joint probabilities
for (i in 1:3) {
  for (j in 1:3) {
    prob_joint <- mean(Y >= qY[i] & Z >= qZ[j])
    table[i, j] <- prob_joint
  }
}

# Marginal probabilities
marginal_Y <- colSums(table[1:3, 1:3])
marginal_Z <- rowSums(table[1:3, 1:3])

# Compute the sum
table[5, 1:4] <- c(marginal_Y, sum(marginal_Y))
table[1:4, 5] <- c(marginal_Z, sum(marginal_Z))
table[5, 5] <- sum(table[5, 1:4])

print(table)
```

```
##        1st Y  2nd Y  3rd Y  4th Y    Sum
## 1st Z 0.6074 0.4459 0.2434 0.0000 1.2967
## 2nd Z 0.4205 0.3265 0.2003 0.0000 0.9473
## 3rd Z 0.2184 0.1798 0.1201 0.0000 0.5183
## 4th Z 0.0000 0.0000 0.0000 0.0000 2.7623
## Sum   1.2463 0.9522 0.5638 2.7623 5.5246
```

Fisher's Exact and Chi Square Tests

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?

```r
fisher.test(table[1:3, 1:3])
```

```
## Warning in fisher.test(table[1:3, 1:3]): 'x' has been rounded to integer: Mean
## relative difference: 0.9222387
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table[1:3, 1:3]
## p-value = 1
## alternative hypothesis: two.sided
```

```r
chisq.test(table[1:3, 1:3])
```

```
## Warning in chisq.test(table[1:3, 1:3]): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table[1:3, 1:3]
## X-squared = 0.0058904, df = 4, p-value = 1
```

We cannot reject the null hypothesis for either test given the p-value = 1. Fisher's exact test performs a test of the independence of rows and columns in a contingency table with fixed marginals and is used for small sample sizes so for our dataset the chi-square approximation would be more appropriate to use.

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

Descriptive and Inferential Statistics

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

```r
url = 'https://raw.githubusercontent.com/JAbinette/CUNY-605-Final/main/train.csv'
train <- read.csv( url, header = TRUE, sep = ",", stringsAsFactors = FALSE)
dim(train)
```

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

```r
summary(train)
```

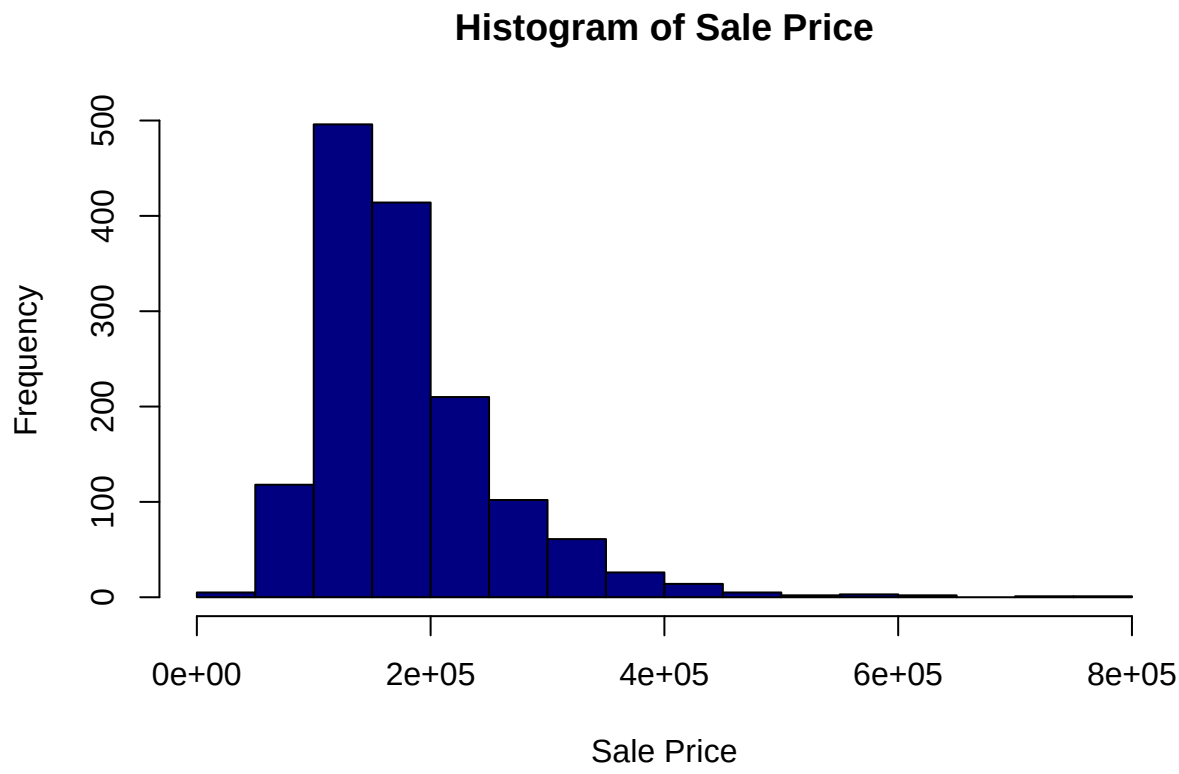
```
##        Id           MSSubClass      MSZoning          LotFrontage
##  Min.   :   1.0   Min.   : 20.0   Length:1460        Min.   : 21.00
##  1st Qu.: 365.8   1st Qu.: 20.0   Class :character   1st Qu.: 59.00
##  Median : 730.5   Median : 50.0   Mode  :character   Median : 69.00
##  Mean   : 730.5   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          LandSlope
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
##
##  Neighborhood        Condition1         Condition2          BldgType
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   HouseStyle         OverallQual      OverallCond      YearBuilt
##  Length:1460        Min.   : 1.000   Min.   :1.000   Min.   :1872
##  Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954
##  Mode  :character   Median : 6.000   Median :5.000   Median :1973
##                     Mean   : 6.099   Mean   :5.575   Mean   :1971
##                     3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000
##                     Max.   :10.000   Max.   :9.000   Max.   :2010
##
##   YearRemodAdd   RoofStyle           RoofMatl         Exterior1st
##  Min.   :1950   Length:1460        Length:1460        Length:1460
##  1st Qu.:1967   Class :character   Class :character   Class :character
##  Median :1994   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1985
##  3rd Qu.:2004
##  Max.   :2010
##
##  Exterior2nd         MasVnrType          MasVnrArea      ExterQual
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median :   0.0   Mode  :character
##                                        Mean   : 103.7
##                                        3rd Qu.: 166.0
##                                        Max.   :1600.0
##                                        NA's   :8
##   ExterCond          Foundation          BsmtQual           BsmtCond
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :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          X1stFlrSF
##  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
##
##    X2ndFlrSF     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
```

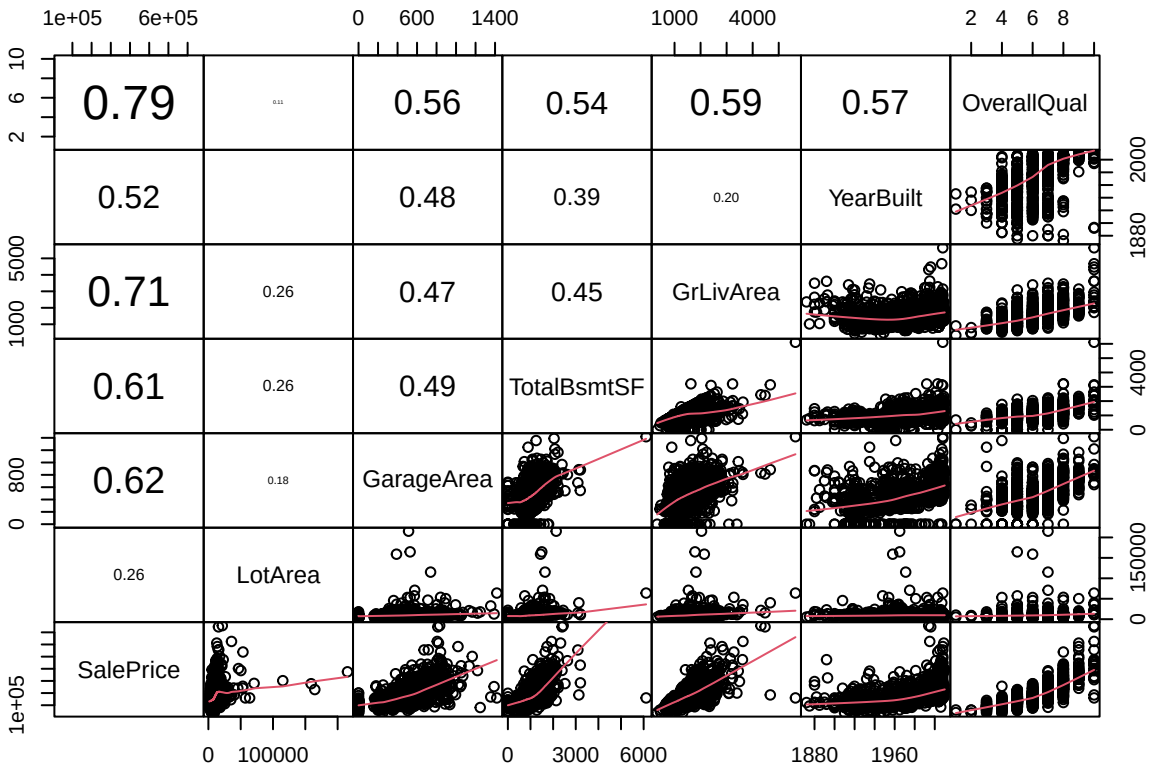
```
##  Mode  :character   Mode  :character   Median :  0.00   Median : 25.00
##                                        Mean   : 94.24   Mean   : 46.66
##                                        3rd Qu.:168.00   3rd Qu.: 68.00
##                                        Max.   :857.00   Max.   :547.00
##
##  EnclosedPorch      X3SsnPorch      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           MiscVal
##  Length:1460        Length:1460        Length:1460        Min.   :    0.00
##  Class :character   Class :character   Class :character   1st Qu.:    0.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    0.00
##                                                           Mean   :   43.49
##                                                           3rd Qu.:    0.00
##                                                           Max.   :15500.00
##
##      MoSold           YrSold       SaleType         SaleCondition
##  Min.   : 1.000   Min.   :2006   Length:1460        Length:1460
##  1st Qu.: 5.000   1st Qu.:2007   Class :character   Class :character
##  Median : 6.000   Median :2008   Mode  :character   Mode  :character
##  Mean   : 6.322   Mean   :2008
##  3rd Qu.: 8.000   3rd Qu.:2009
##  Max.   :12.000   Max.   :2010
##
##    SalePrice
##  Min.   : 34900
##  1st Qu.:129975
##  Median :163000
##  Mean   :180921
##  3rd Qu.:214000
##  Max.   :755000
##
```

```r
hist(train$SalePrice, xlab="Sale Price", main="Histogram of Sale Price", col="navy")
```



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

```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
pairs(~ SalePrice+LotArea+GarageArea+TotalBsmtSF+GrLivArea+YearBuilt+OverallQual, data=train, lower.pan
```



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 an 80% confidence interval.

```
library(stats)
cor_matrix <- cor(train[c("SalePrice","GrLivArea", "GarageArea", "OverallQual")])
cor_matrix
```

```
##             SalePrice GrLivArea GarageArea OverallQual
## SalePrice   1.0000000 0.7086245  0.6234314   0.7909816
## GrLivArea   0.7086245 1.0000000  0.4689975   0.5930074
## GarageArea  0.6234314 0.4689975  1.0000000   0.5620218
## OverallQual 0.7909816 0.5930074  0.5620218   1.0000000
```

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

```
##
##  Pearson's product-moment correlation
##
## data:  train$GrLivArea and train$SalePrice
## t = 38.348, df = 1458, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.6915087 0.7249450
## sample estimates:
##       cor
## 0.7086245
```

```r
cor.test(train$GarageArea, train$SalePrice, conf.level = 0.8)
```

```
##
##  Pearson's product-moment correlation
##
## data:  train$GarageArea and train$SalePrice
## 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
```

```r
cor.test(train$OverallQual, train$SalePrice, conf.level = 0.8)
```

```
##
##  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
```

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

As shown above, all three variables (GrLivArea, GarageArea, & OverallQual) have a statistically significant and strong positive correlation (p-values less than .001 & correlation > .5) with SalePrice. Family-wise error describes the probability of making type I errors when performing multiple hypotheses tests. I'm not concerned in this case given these correlations are strong and the p-values are all less than .001.

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.

```r
# Create precision matrix
precision_matrix <- solve(cor_matrix)
# Multiply precision by correlation matrices
cor_by_precision <- cor_matrix %*% precision_matrix
# Multiply correlation by precision matrices
precision_by_cor <- precision_matrix %*% cor_matrix

cor_matrix
```

```
##             SalePrice GrLivArea GarageArea OverallQual
```

```
## SalePrice   1.0000000 0.7086245  0.6234314   0.7909816
## GrLivArea   0.7086245 1.0000000  0.4689975   0.5930074
## GarageArea  0.6234314 0.4689975  1.0000000   0.5620218
## OverallQual 0.7909816 0.5930074  0.5620218   1.0000000
```

```
precision_matrix
```

```
##              SalePrice   GrLivArea  GarageArea OverallQual
## SalePrice    3.8379944 -1.26043717 -0.75349666  -1.8648529
## GrLivArea   -1.2604372  2.02315854 -0.07177202  -0.1624280
## GarageArea  -0.7534967 -0.07177202  1.67296469  -0.3016792
## OverallQual -1.8648529 -0.16242800 -0.30167921   2.7409356
```

```
cor_by_precision
```

```
##                SalePrice    GrLivArea    GarageArea   OverallQual
## SalePrice   1.000000e+00 2.775558e-17 -2.775558e-17  0.000000e+00
## GrLivArea   2.220446e-16 1.000000e+00 -5.551115e-17 -2.220446e-16
## GarageArea  4.440892e-16 5.551115e-17  1.000000e+00  0.000000e+00
## OverallQual 2.220446e-16 2.775558e-17 -1.110223e-16  1.000000e+00
```

```
precision_by_cor
```

```
##                 SalePrice     GrLivArea   GarageArea   OverallQual
## SalePrice    1.000000e+00  0.000000e+00 2.220446e-16  0.000000e+00
## GrLivArea   -2.775558e-17  1.000000e+00 2.775558e-17  0.000000e+00
## GarageArea  -2.775558e-17 -5.551115e-17 1.000000e+00 -1.110223e-16
## OverallQual  0.000000e+00  2.220446e-16 2.220446e-16  1.000000e+00
```

Conduct LU decomposition on the matrix.

```
# Conduct LU decomposition
library("pracma")
```

```
## Warning: package 'pracma' was built under R version 4.2.2
```

```
##
## Attaching package: 'pracma'
```

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

```
LU <- lu(cor_matrix)
LU
```

```
## $L
##             SalePrice  GrLivArea GarageArea OverallQual
## SalePrice   1.0000000 0.00000000  0.0000000           0
## GrLivArea   0.7086245 1.00000000  0.0000000           0
```

```
## GarageArea  0.6234314 0.05467234  1.0000000           0
## OverallQual 0.7909816 0.06527753  0.1100643           1
##
## $U
##             SalePrice GrLivArea GarageArea OverallQual
## SalePrice           1 0.7086245  0.6234314  0.79098160
## GrLivArea           0 0.4978513  0.0272187  0.03249851
## GarageArea          0 0.0000000  0.6098451  0.06712219
## OverallQual         0 0.0000000  0.0000000  0.36483893
```

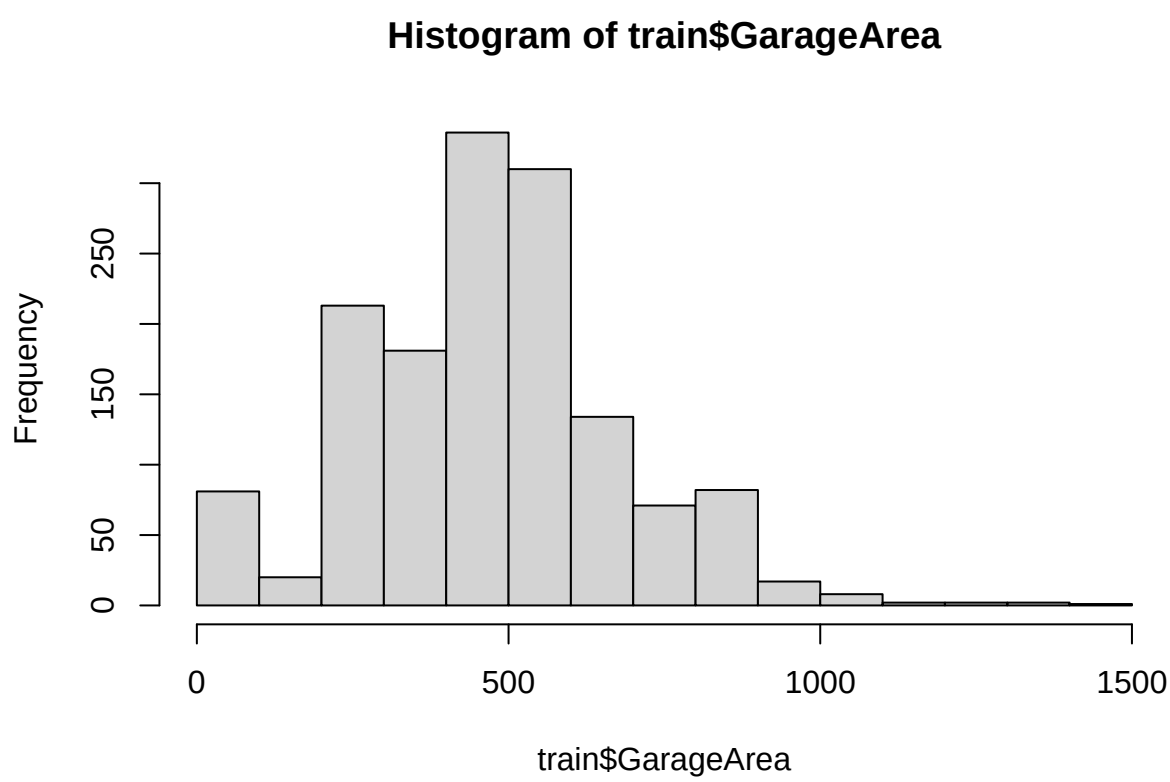
Calculus-Based Probability & Statistics

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.

```
min(train$GarageArea)
```

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

```
hist(train$GarageArea)
```



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 lambda

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

```
library(MASS)
```

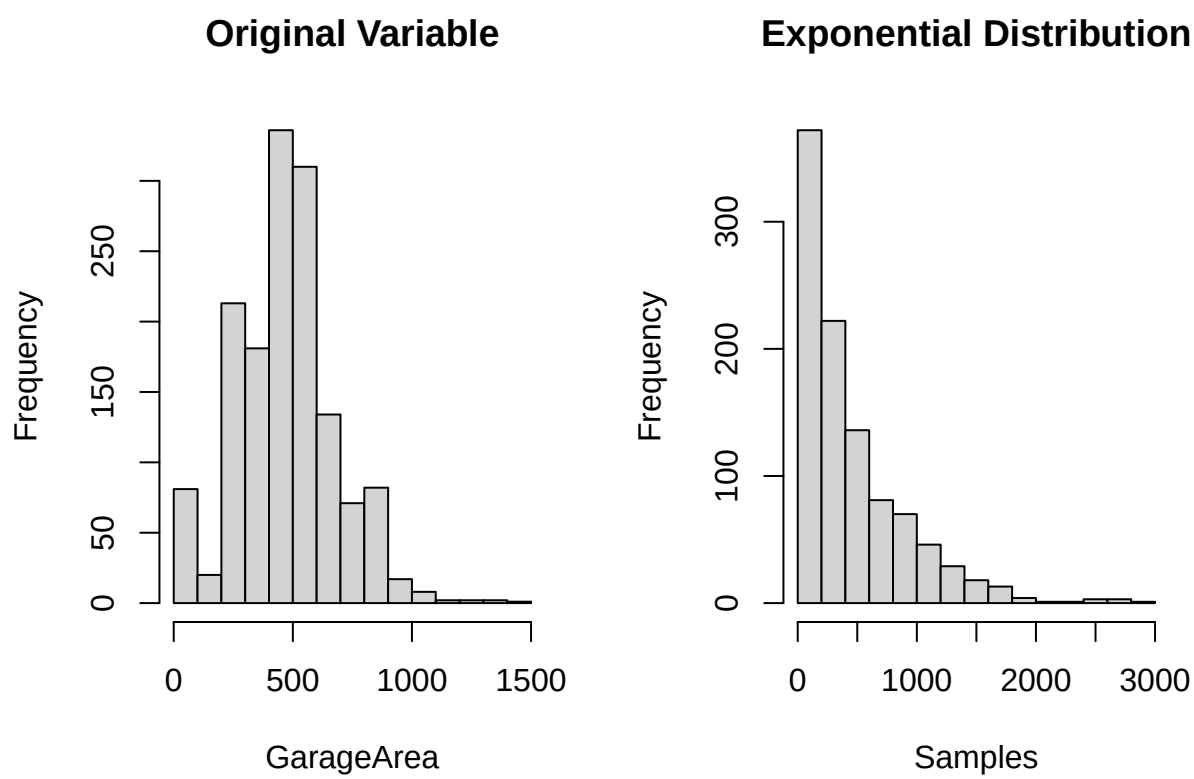
```
##
## Attaching package: 'MASS'

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

```
fit_exp <- fitdistr(train$GarageArea, "exponential")
# Lambda = rate
lambda <- fit_exp$estimate["rate"]
# Take 1000 samples from this exponential distribution
samp <- rexp(1000, lambda)
```

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

```
par(mfrow = c(1, 2))
hist(train$GarageArea, main = "Original Variable", xlab = "GarageArea")
hist(samp, main = "Exponential Distribution", xlab = "Samples")
```



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.

```
qexp(c(.05, .95), rate = lambda)
```

```
## [1]   24.26071 1416.92186
```

```
quantile(samp, c(.05, .95))
```

```
##         5%        95%
##   25.08284 1346.04068
```

```
quantile(train$GarageArea, c(.05, .95))
```

```
##    5%   95%
##   0.0 850.1
```

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.

Research Question - How can we predict sale price?

```
train.lm <- lm(SalePrice ~ LotArea+GarageArea+TotalBsmtSF+GrLivArea+YearBuilt+OverallQual+BldgType+Hous
summary(train.lm)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + GarageArea + TotalBsmtSF +
##     GrLivArea + YearBuilt + OverallQual + BldgType + HouseStyle,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -486463  -18092   -1857   13904  275890
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -8.348e+05  9.161e+04  -9.112  < 2e-16 ***
## LotArea           5.300e-01  1.055e-01   5.024 5.68e-07 ***
## GarageArea        3.542e+01  6.080e+00   5.826 7.00e-09 ***
## TotalBsmtSF       9.076e+00  3.784e+00   2.398 0.016601 *
## GrLivArea         6.454e+01  3.704e+00  17.423  < 2e-16 ***
## YearBuilt         3.783e+02  4.847e+01   7.806 1.13e-14 ***
## OverallQual       2.205e+04  1.175e+03  18.762  < 2e-16 ***
## BldgType2fmCon   -5.824e+03  6.953e+03  -0.838 0.402405
## BldgTypeDuplex   -2.897e+04  5.573e+03  -5.198 2.30e-07 ***
## BldgTypeTwnhs    -2.017e+04  6.083e+03  -3.316 0.000937 ***
## BldgTypeTwnhsE   -1.234e+04  3.893e+03  -3.171 0.001552 **
```

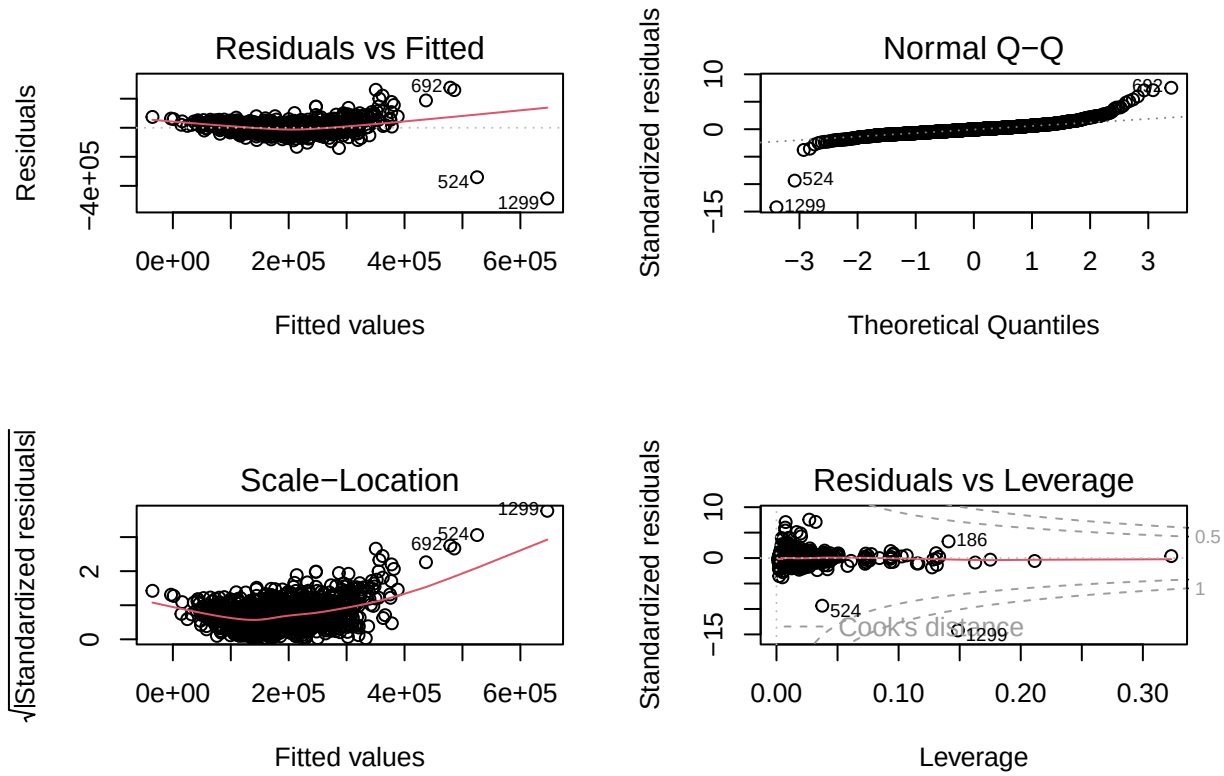
```
## HouseStyle1.5Unf  1.399e+04  1.062e+04   1.317 0.188147
## HouseStyle1Story  1.711e+04  4.055e+03   4.219 2.61e-05 ***
## HouseStyle2.5Fin -2.385e+04  1.413e+04  -1.688 0.091705 .
## HouseStyle2.5Unf -2.379e+04  1.179e+04  -2.018 0.043761 *
## HouseStyle2Story -2.605e+03  3.961e+03  -0.658 0.510876
## HouseStyleSFoyer  2.521e+04  7.449e+03   3.385 0.000732 ***
## HouseStyleSLvl    5.719e+03  5.754e+03   0.994 0.320465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37070 on 1442 degrees of freedom
## Multiple R-squared:  0.7848, Adjusted R-squared:  0.7823
## F-statistic: 309.4 on 17 and 1442 DF,  p-value: < 2.2e-16
```

```r
# Backwise selection
# Remove TotalBsmtSF variable
train2.lm <- lm(SalePrice ~ LotArea+GarageArea+GrLivArea+YearBuilt+OverallQual+BldgType, data=train)
summary(train2.lm)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + GarageArea + GrLivArea + YearBuilt +
##     OverallQual + BldgType, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -427875  -20212   -2086   16254  300410
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -9.622e+05  8.590e+04 -11.202  < 2e-16 ***
## LotArea         7.136e-01  1.070e-01   6.669 3.65e-11 ***
## GarageArea      4.744e+01  6.151e+00   7.712 2.28e-14 ***
## GrLivArea       5.196e+01  2.641e+00  19.672  < 2e-16 ***
## YearBuilt       4.548e+02  4.515e+01  10.072  < 2e-16 ***
## OverallQual     2.309e+04  1.167e+03  19.782  < 2e-16 ***
## BldgType2fmCon -9.513e+03  7.156e+03  -1.329  0.18391
## BldgTypeDuplex -2.204e+04  5.586e+03  -3.945 8.37e-05 ***
## BldgTypeTwnhs  -3.023e+04  6.119e+03  -4.941 8.66e-07 ***
## BldgTypeTwnhsE -1.263e+04  4.009e+03  -3.149  0.00167 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38310 on 1450 degrees of freedom
## Multiple R-squared:  0.7689, Adjusted R-squared:  0.7675
## F-statistic: 536.1 on 9 and 1450 DF,  p-value: < 2.2e-16
```

Test Model Assumptions of Linearity, Nearly Normal Residuals, and Constant Variability

```r
par(mfrow=c(2,2))
plot(train.lm)
```



Use Multiple Regression Model to predict Sales Price for test.csv dataset

```
url = 'https://raw.githubusercontent.com/JAbinette/CUNY-605-Final/main/test.csv'
test <- read.csv( url, header = TRUE, sep = ",", stringsAsFactors = FALSE)
# Predicts the future values
test$SalePrice <- predict(train.lm, newdata = test)
```

Two cases are missing data - Build additional Models to Predict the Sales Price

```
MissingValues <- subset.data.frame(test, is.na(SalePrice), select=c(Id,LotArea,GarageArea,TotalBsmtSF,G

train_no_TotalBsmt.lm <- lm(SalePrice ~ LotArea + GarageArea + GrLivArea + YearBuilt + OverallQual +
                            BldgType + HouseStyle, data=train)
train_no_GarageArea.lm <- lm(SalePrice ~ LotArea+TotalBsmtSF+GrLivArea+YearBuilt+OverallQual+
                             BldgType+HouseStyle, data=train)

MissingValues$SalePrice[MissingValues$Id == 2121] <- predict(train_no_TotalBsmt.lm, newdata = test)
```

```
## Warning in MissingValues$SalePrice[MissingValues$Id == 2121] <-
## predict(train_no_TotalBsmt.lm, : number of items to replace is not a multiple of
## replacement length
```

```
MissingValues$SalePrice[MissingValues$Id == 2577] <- predict(train_no_GarageArea.lm, newdata = test)
```

```
## Warning in MissingValues$SalePrice[MissingValues$Id == 2577] <-
```

```
## predict(train_no_GarageArea.lm, : number of items to replace is not a multiple
## of replacement length
```

```r
test$SalePrice[test$Id == 2121] <- 119506.70
test$SalePrice[test$Id == 2577] <- 132903.30

# Export to CSV without quotes
predictions <- subset.data.frame(test, select = c("Id", "SalePrice") )
library(utils)
# write.csv(predictions, file='SalePricePredictions.csv',quote=FALSE, row.names=FALSE)
knitr::include_graphics("Kaggle House Price Prediction Score.jpg")
```