Project 1

David Contento

October 27, 2018

Question 1

Part A) Bootstraping

```
library(AER)
```

Using the "Affairs" data set, we created a linear regression model that tested the dependency of number of affairs a subject had with the subject's age and years married.

```
data("Affairs")
head(Affairs)
```

```
##    affairs gender age yearsmarried children religiousness education
## 4        0   male  37        10.00       no             3        18
## 5        0 female  27         4.00       no             4        14
## 11       0 female  32        15.00      yes             1        12
## 16       0   male  57        15.00      yes             5        18
## 23       0   male  22         0.75       no             2        17
## 29       0 female  32         1.50       no             2        17
##    occupation rating
## 4           7      4
## 5           6      4
## 11          1      4
## 16          6      5
## 23          6      3
## 29          5      5
```

```
relation<-lm(affairs~age+yearsmarried, data=Affairs)
summary(relation)
```

```
##
## Call:
## lm(formula = affairs ~ age + yearsmarried, data = Affairs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6300 -1.7312 -0.9970 -0.3918 11.6199
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.53484    0.54768   2.802  0.00524 **
## age          -0.04494    0.02261  -1.987  0.04733 *
## yearsmarried  0.16889    0.03770   4.480 8.96e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.235 on 598 degrees of freedom
## Multiple R-squared:  0.04124,    Adjusted R-squared:  0.03804
## F-statistic: 12.86 on 2 and 598 DF,  p-value: 3.396e-06
```

As we can see above, the LS estimates of our parameters are all statistically significant at the 5% level, meaning there is at least a 95% chance our parameters are statistically significant to our dependent variable of total number of affairs. The below code represents the confidence interval for each parameter.

```
CI1<-confint(relation,'(Intercept)', level= 0.95)
CI1
```

```
##                 2.5 %  97.5 %
## (Intercept) 0.4592265 2.61045
```

```
CI2<-confint(relation,'age', level= 0.95)
CI2
```

```
##           2.5 %        97.5 %
## age -0.08935362 -0.0005309759
```

```
CI3<-confint(relation,'yearsmarried', level= 0.95)
CI3
```

```
##                   2.5 %   97.5 %
## yearsmarried 0.09484536 0.242935
```

We then bootstrap our LS estimates along with their respective confidence intervals below:

```
betaestimates<-NULL
coint1<-NULL
coint2<-NULL
coint3<-NULL

n=length(Affairs$affairs)
y=1000

for(draw in 1:y){
  bootstrap= Affairs[sample(1:n, size=n, replace=TRUE),]
  linear= lm(affairs~age+yearsmarried, data=bootstrap)
  betaestimates= rbind(betaestimates,coef(linear))

  coint1=rbind(coint1,confint(linear,'(Intercept)',level=0.95))
  coint2=rbind(coint2,confint(linear,'age',level=0.95))
  coint3=rbind(coint3,confint(linear,'yearsmarried',level=0.95))
}
```

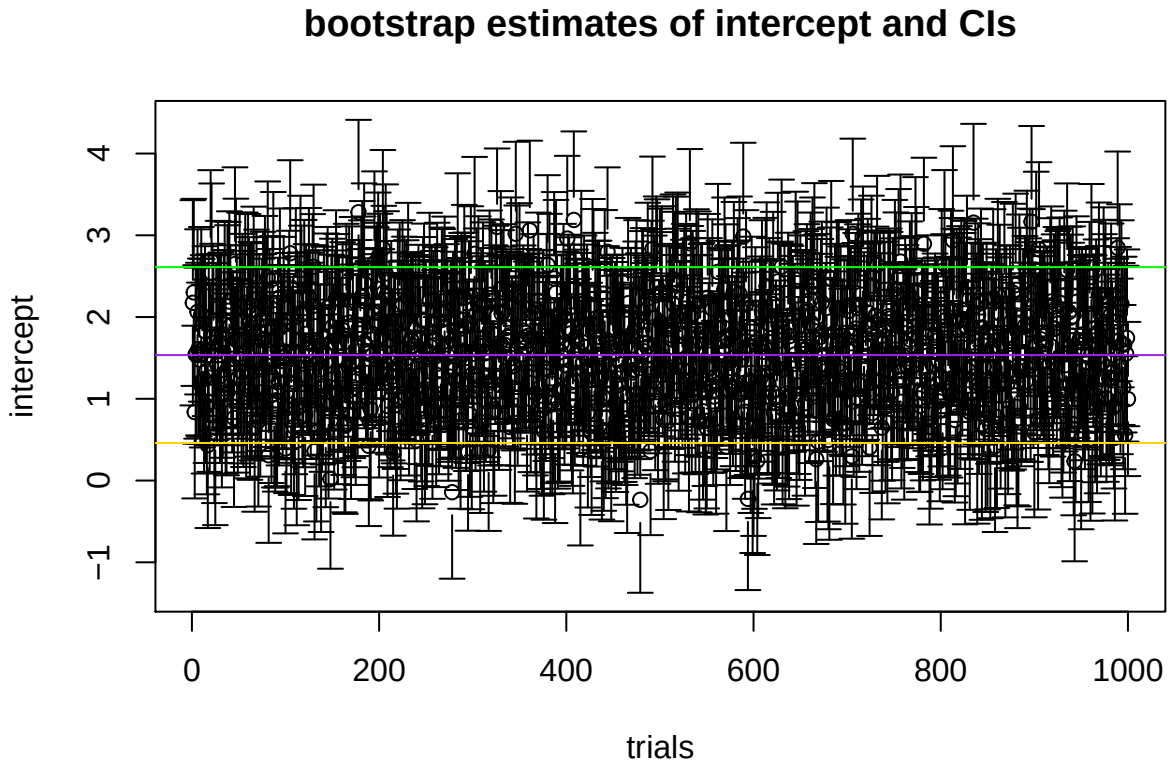
The following plots the bootstrapped estimates and their respective confidence intervals vs. the number of trials:

```
library(gplots)
```

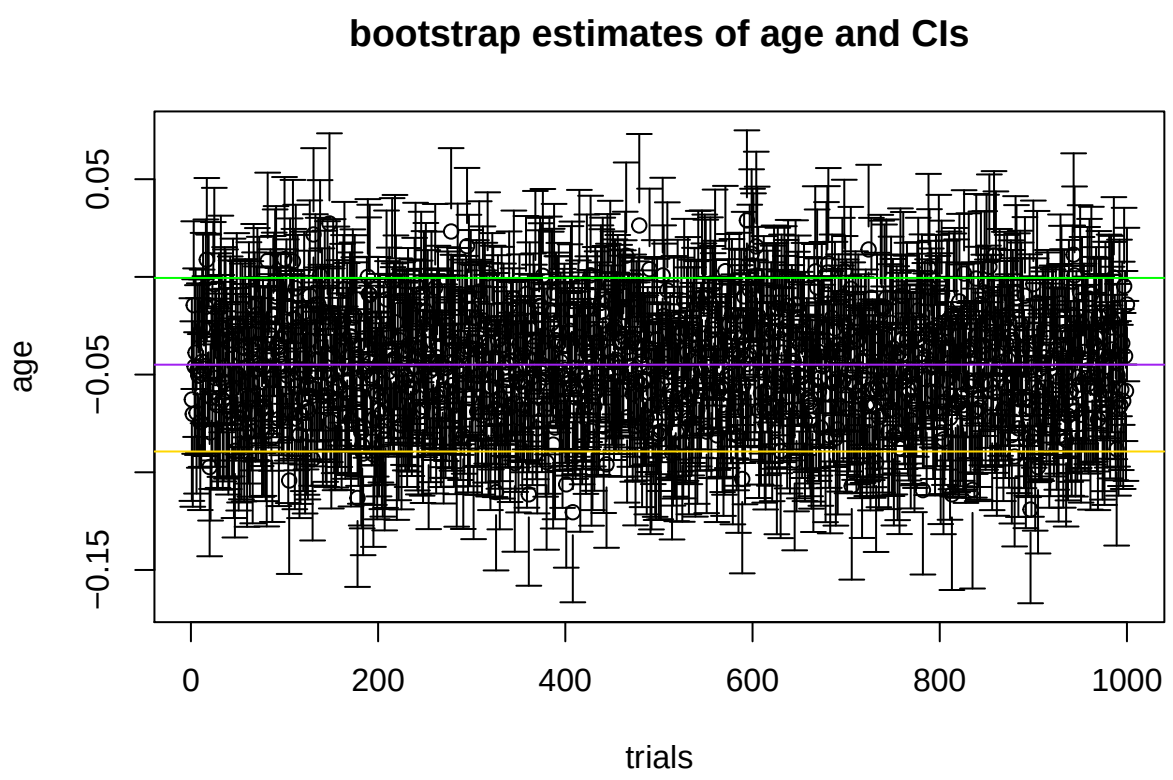
```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

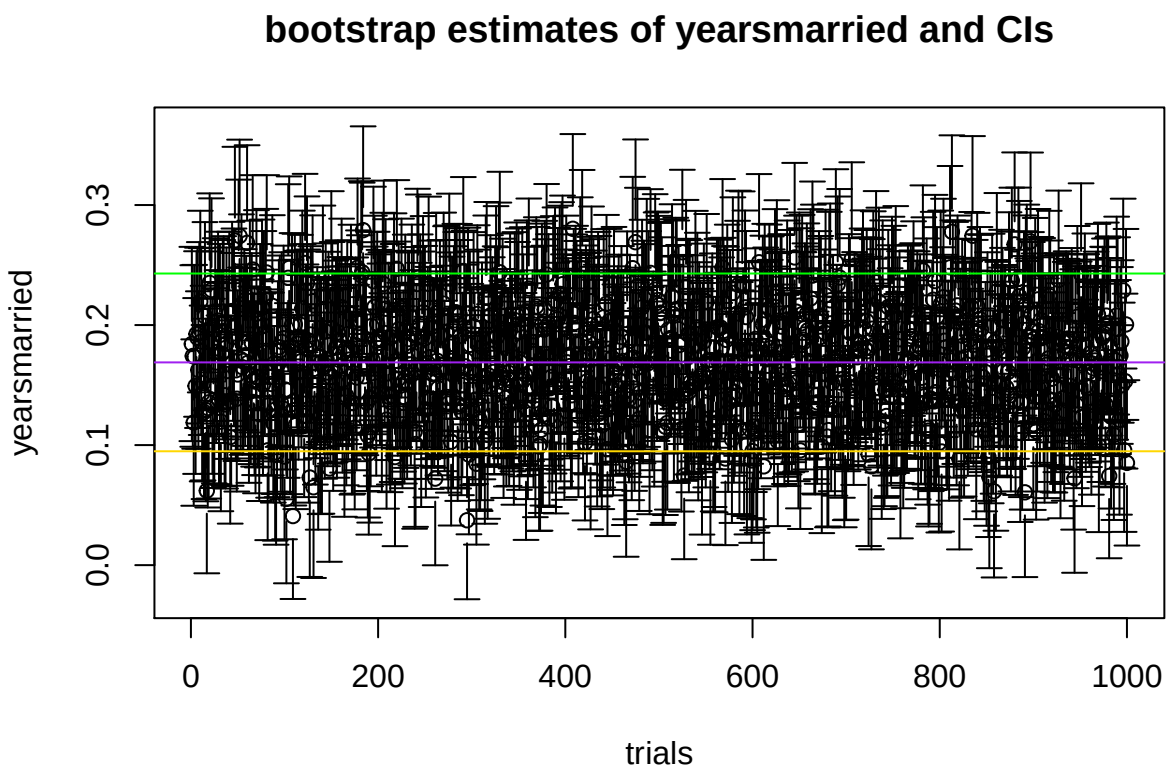
```
plotCI(1:1000,betaestimates[,1],ui=coint1[,2], li=coint1[,1], ylab="intercept", xlab="trials", main= "b

abline(h=1.53484,col="purple")
abline(h=0.4592265,col="gold")
abline(h=2.61045,col="green")
```




```
plotCI(1:1000,betaestimates[,2],ui=coint2[,2], li=coint2[,1], ylab="age", xlab="trials", main= "bootstra

abline(h=-0.04494, col="purple")
abline(h=-0.08935362, col="gold")
abline(h=-0.0005309759, col="green")
```




```
plotCI(1:1000,betaestimates[,3],ui=coint3[,2], li=coint3[,1], ylab="yearsmarried", xlab="trials", main=

abline(h=0.16889, col="purple")
abline(h=0.09484536, col="gold")
abline(h=0.242935, col="green")
```



The purple line represents the LS estimates of each respective parameter; the gold line represents the lower limit of the confidence intervals, and the green line represents the upper limit of the confidence intervals.

Based on these plots, we can determine that the bootstrap estimates are stable, as most of the trial estimates fall within the same range as our LS estimates.

Part B) Markov Chain Monte Carlo

First, we must run a Bayesian regression using MCMC and analyze the estimates:

```
library(MCMCpack)
linear.MCMC<-MCMCregress(affairs~age+yearsmarried, data=Affairs,burnin=100,mcmc=1000,thin=1,b0=c(0,0,0)
summary(linear.MCMC)
```

```
##
## Iterations = 101:1100
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean      SD  Naive SE Time-series SE
## (Intercept)   1.54692 0.53726 0.0169896      0.0209490
```

```
## age          -0.04527 0.02224 0.0007032      0.0007773
## yearsmarried  0.16912 0.03826 0.0012098      0.0012098
## sigma2       10.49649 0.61356 0.0194025      0.0216409
##
## 2. Quantiles for each variable:
##
##                  2.5%      25%      50%      75%     97.5%
## (Intercept)   0.49998  1.19287  1.55656  1.92171  2.604042
## age          -0.08870 -0.05947 -0.04557 -0.02985 -0.001808
## yearsmarried  0.09639  0.14352  0.16829  0.19484  0.242648
## sigma2        9.32727 10.08466 10.47313 10.87775 11.795634
```

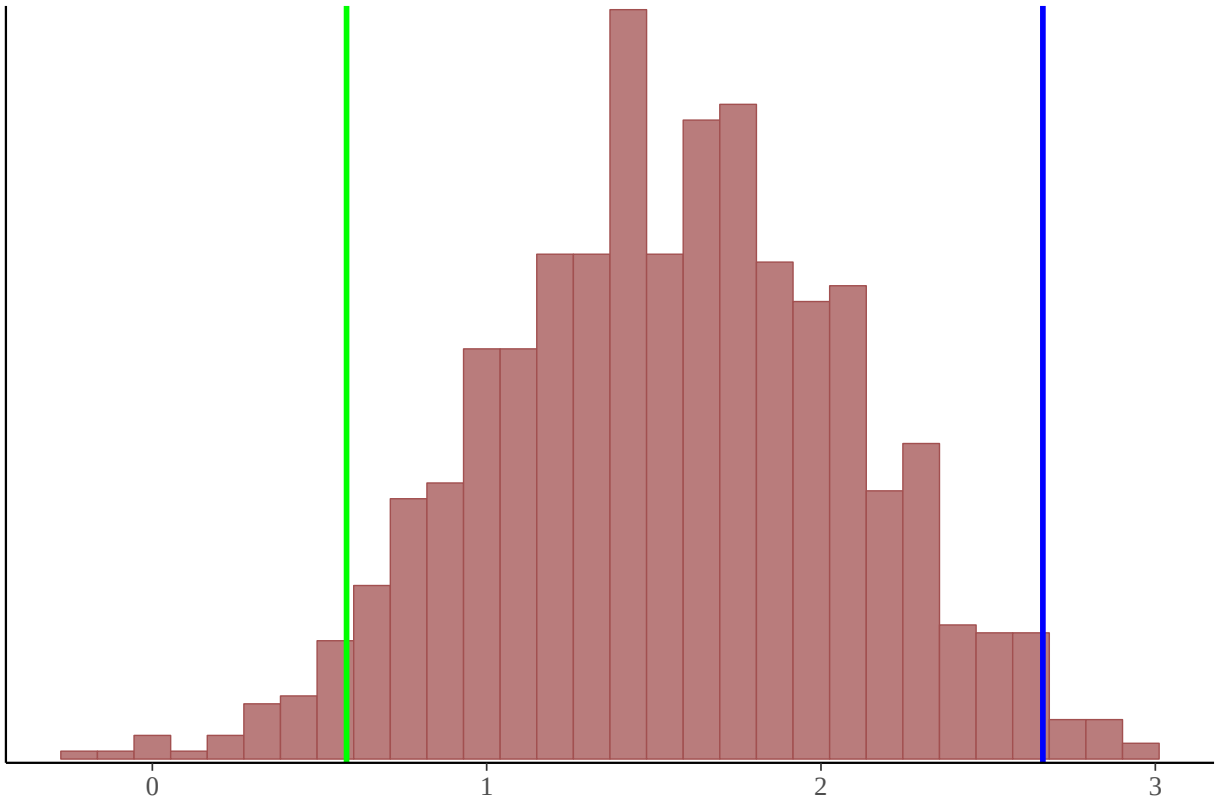
In order to view the Bayesian regression's respective credible intervals:

```
HPDinterval(linear.MCMC)
```

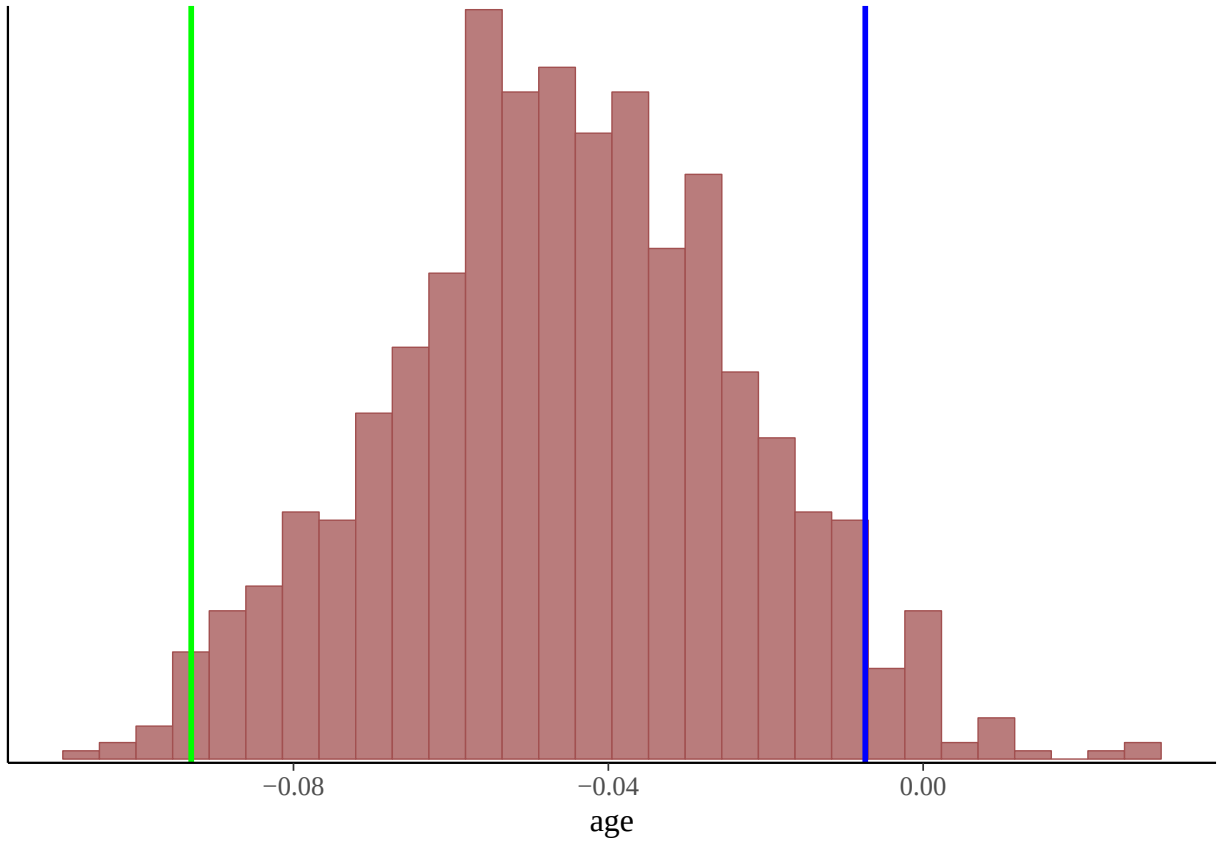
```
##                    lower        upper
## (Intercept)   0.58086335  2.663640909
## age          -0.09299628 -0.007357002
## yearsmarried  0.10054704  0.245405672
## sigma2        9.23911602 11.634740267
## attr(,"Probability")
## [1] 0.95
```

Now we will plot the respective parameter posterior distributions and credible intervals:

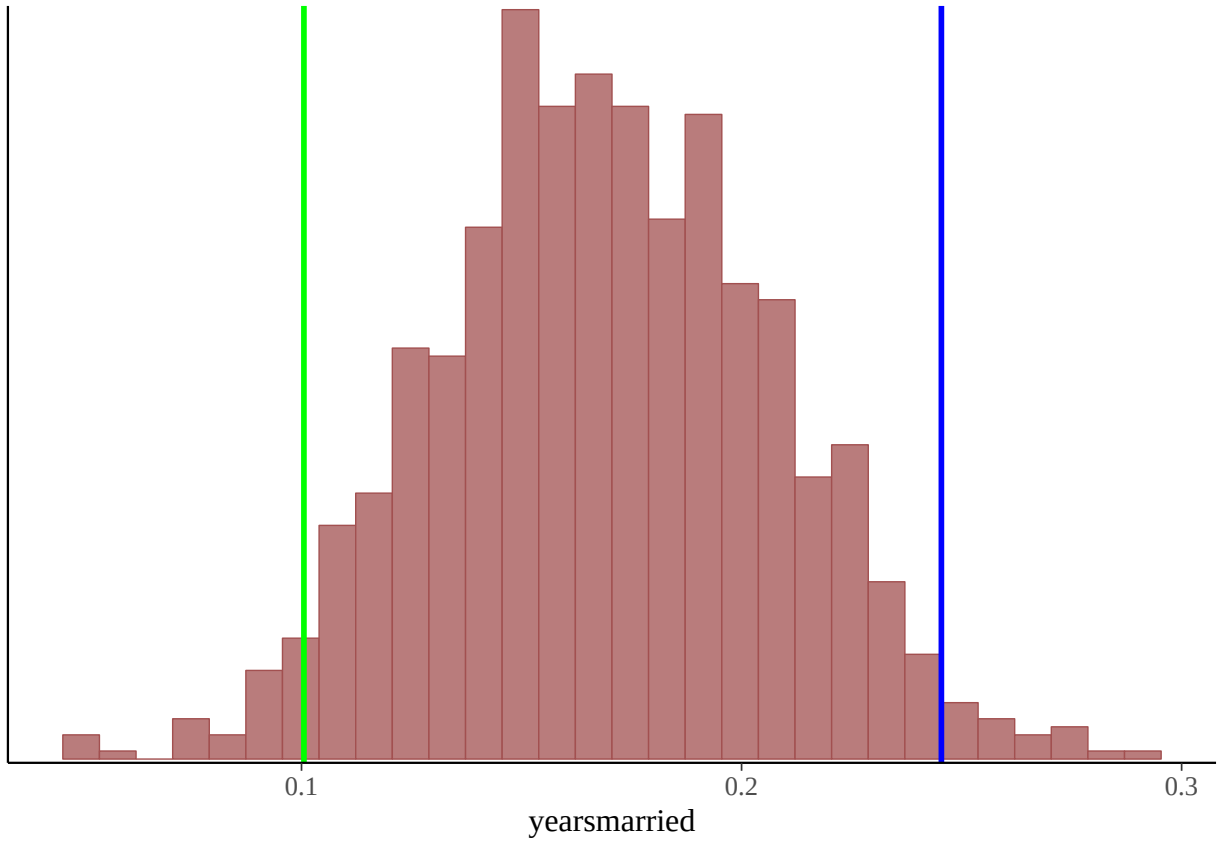
```
library(bayesplot)
post<-as.array(linear.MCMC)

color_scheme_set("red")
pmcmc <-mcmc_hist(post,pars=c("(Intercept)"))
pmcmc + vline_at(0.58086335, linetype = 1, size = 1, color = "green") + vline_at(2.663640909, linetype =
```




```
color_scheme_set("red")
pmcmc <-mcmc_hist(post,pars=c("age"))
pmcmc + vline_at(-0.09299628, linetype = 1, size = 1, color = "green") + vline_at(-0.007357002, linetyp
```




```
color_scheme_set("red")
pmcmc <-mcmc_hist(post,pars=c("yearsmarried"))
pmcmc + vline_at(0.10054704, linetype = 1, size = 1, color = "green") + vline_at(0.245405672, linetype =
```



The estimates from our Bayesian regression compare favorably to the estimates from our LS and Bootstrapped estimates.

Part C) Analysis

All three methods produced relatively comparable results. In our opinion, we would favor the Bayesian method. This is because the Bayesian method accounts for any potentially new information that could influence our results. The LS and Bootstrapped methods assume that there is enough data in our data set of 601 observations to make substantial inferences about how dependent our "affairs" variable is to the "age" and "years married" variable. The Bayesian method accounts for more information, and therefore a more accurate interpretation.

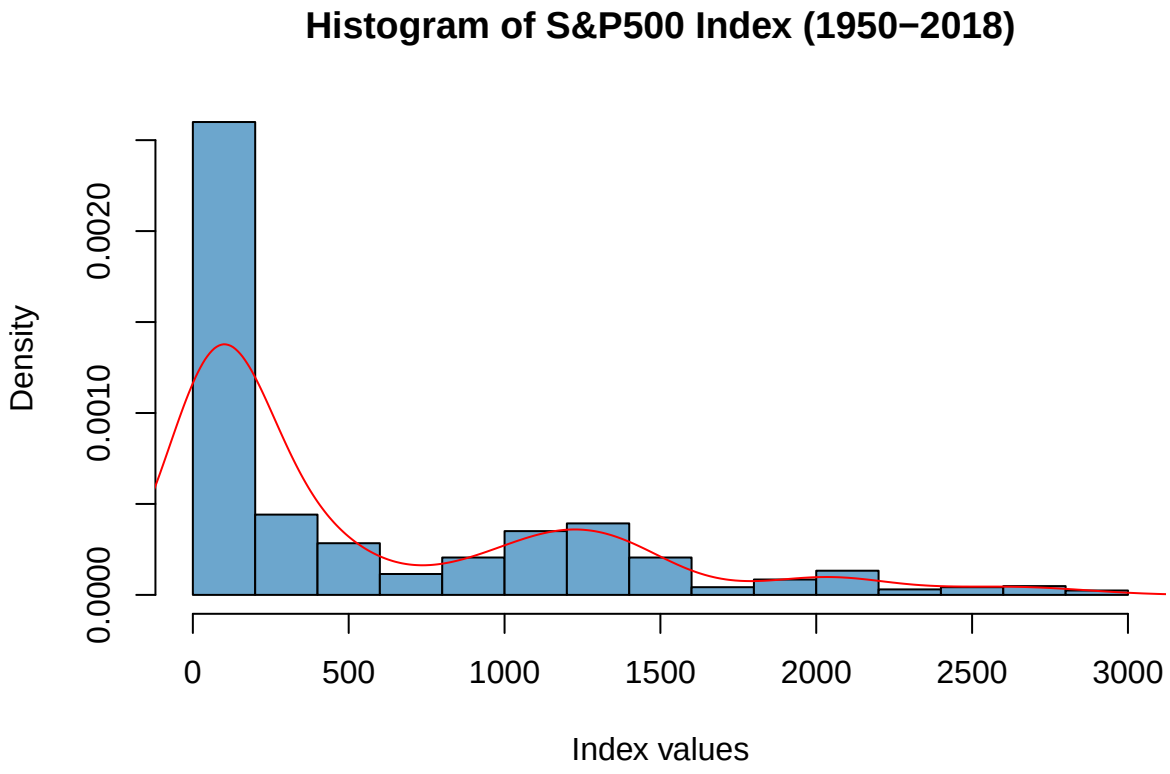
Question 2

Part A) Creating Histograms and Density Curves for Data Sets

-Histogram and Density Curve of S&P500 Index Data

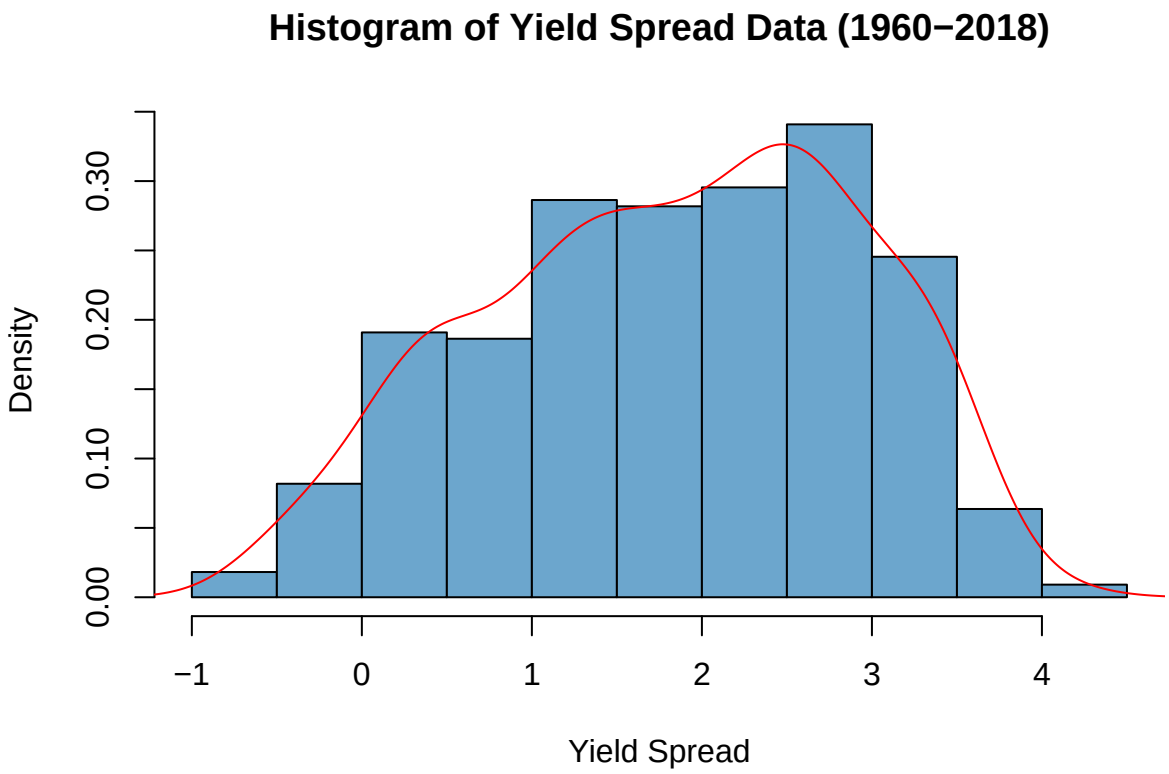
```
setwd("C:/Users/David/Desktop/Grad school work/Fall 2018/403A/Project 1")
library(readr)
sp500=read_csv("sp500.csv")
sp500_close=as.numeric(sp500$Close)
```

```
hist(sp500_close, freq=F, col="skyblue3", main="Histogram of S&P500 Index (1950-2018)", xlab="Index valu
lines(density(sp500_close), col="red")
```



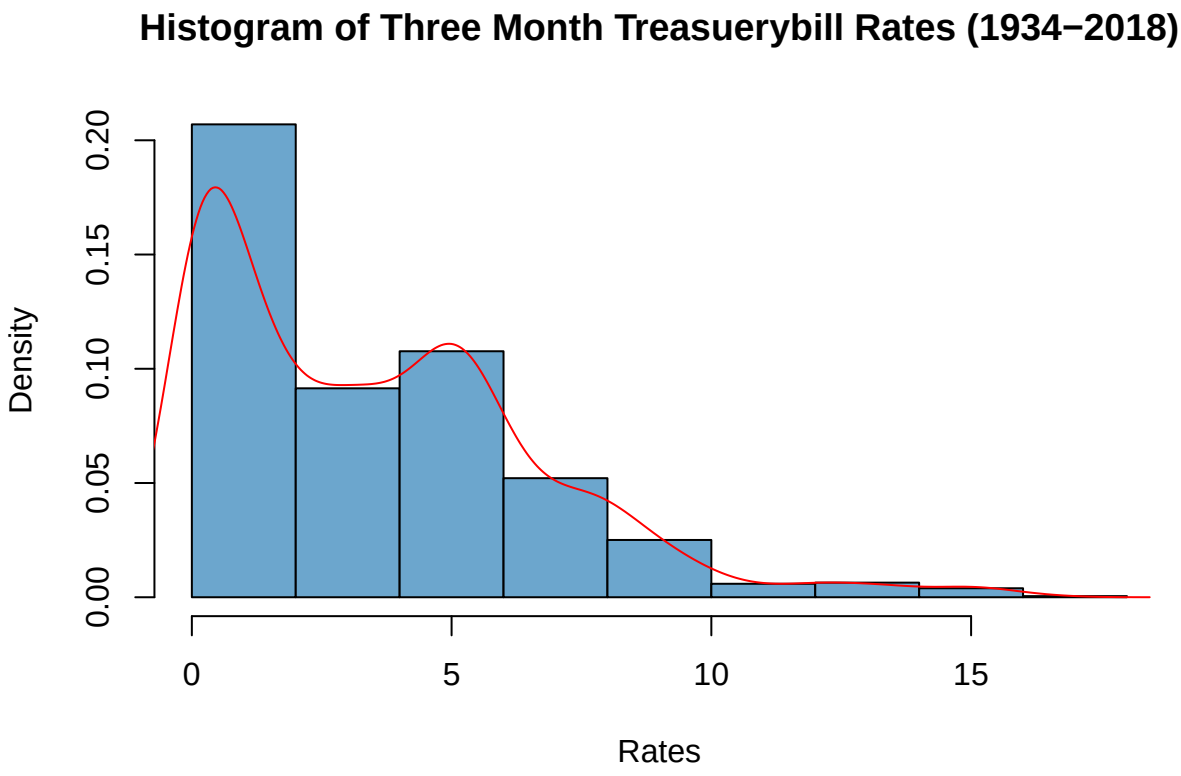
-Histogram and Density Curve of Yield Spread Data

```
setwd("C:/Users/David/Desktop/Grad school work/Fall 2018/403A/Project 1")
library(readr)
Yield_Spread=read_csv("Yield Spread.csv")
hist(Yield_Spread$T10Y3M, freq=F, col="skyblue3", main="Histogram of Yield Spread Data (1960-2018)", xla
lines(density(Yield_Spread$T10Y3M), col="red")
```



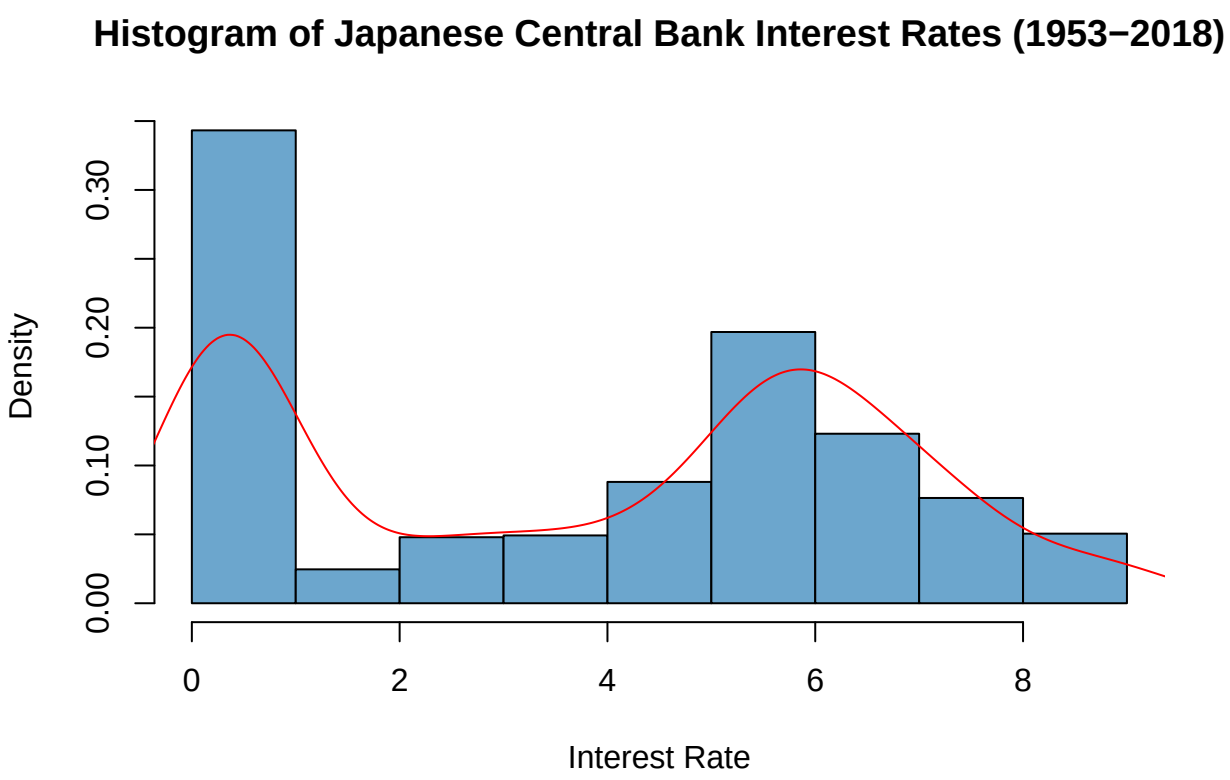
-Histogram and Density Curve of Three Month Treasurybill Data

```
setwd("C:/Users/David/Desktop/Grad school work/Fall 2018/403A/Project 1")
library(readr)
Three_treasury=read_csv("tmo.csv")
hist(Three_treasury$TB3MS, freq=F, col="skyblue3", main="Histogram of Three Month Treasuerybill Rates (
lines(density(Three_treasury$TB3MS), col="red")
```



-Histogram and Density Curve of Japanese Central Bank Interest Rate

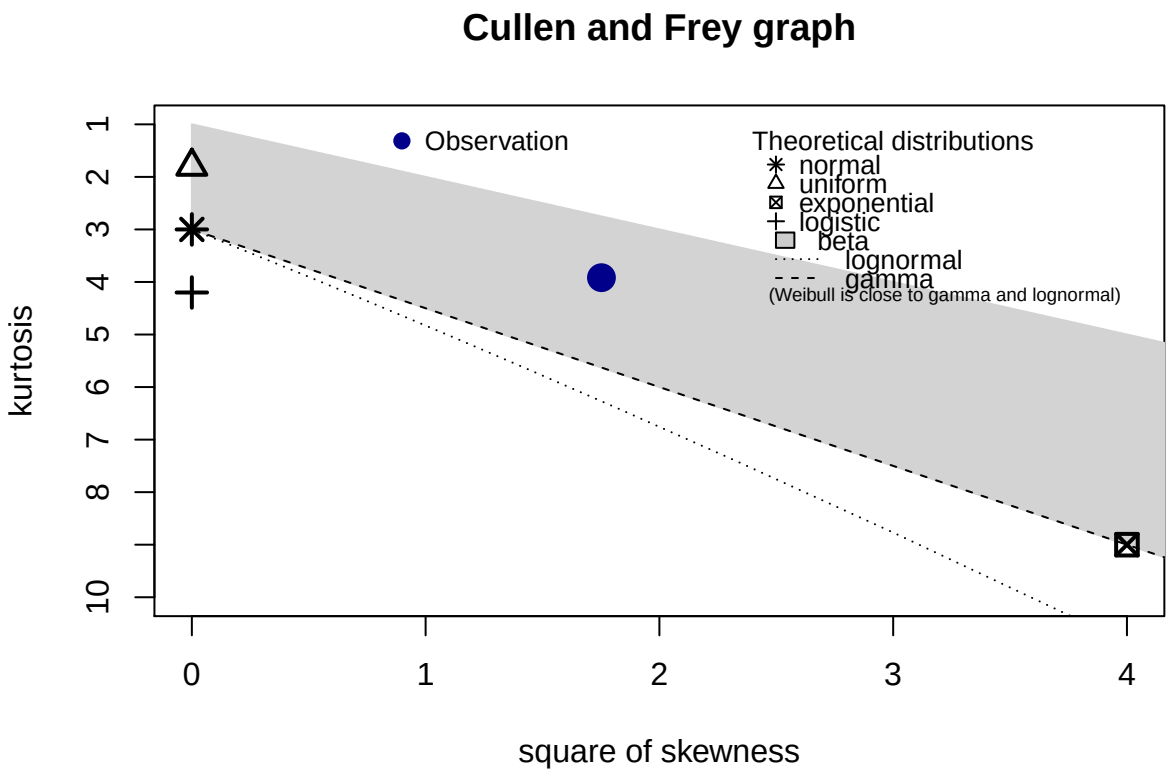
```
setwd("C:/Users/David/Desktop/Grad school work/Fall 2018/403A/Project 1")
library(readr)
Japanese_central_bank=read_csv("Japanese central bank.csv")
hist(Japanese_central_bank$INTDSRJPM193N, freq=F, col="skyblue3", main="Histogram of Japanese Central Ba
lines(density(Japanese_central_bank$INTDSRJPM193N), col="red")
```



Part B) Fitting Distributions To The Datasets

I) Fitting Distributions To S&P500 Index Data

First we used a Cullen-Frey test in order to determine which distributions fit our data the best.




```
## summary statistics
## ------
## min:  17.05   max:  2913.98
## median:  166.07
## mean:  566.9244
## estimated sd:  674.2909
## estimated skewness:  1.323693
## estimated kurtosis:  3.917863
```

We used the following code to fit the closest distribution's our Cullen-Frey graph suggested for our data. The command fitdist, which can be seen below, uses MLE to find the parameters that would best fit the distributions to the data set.

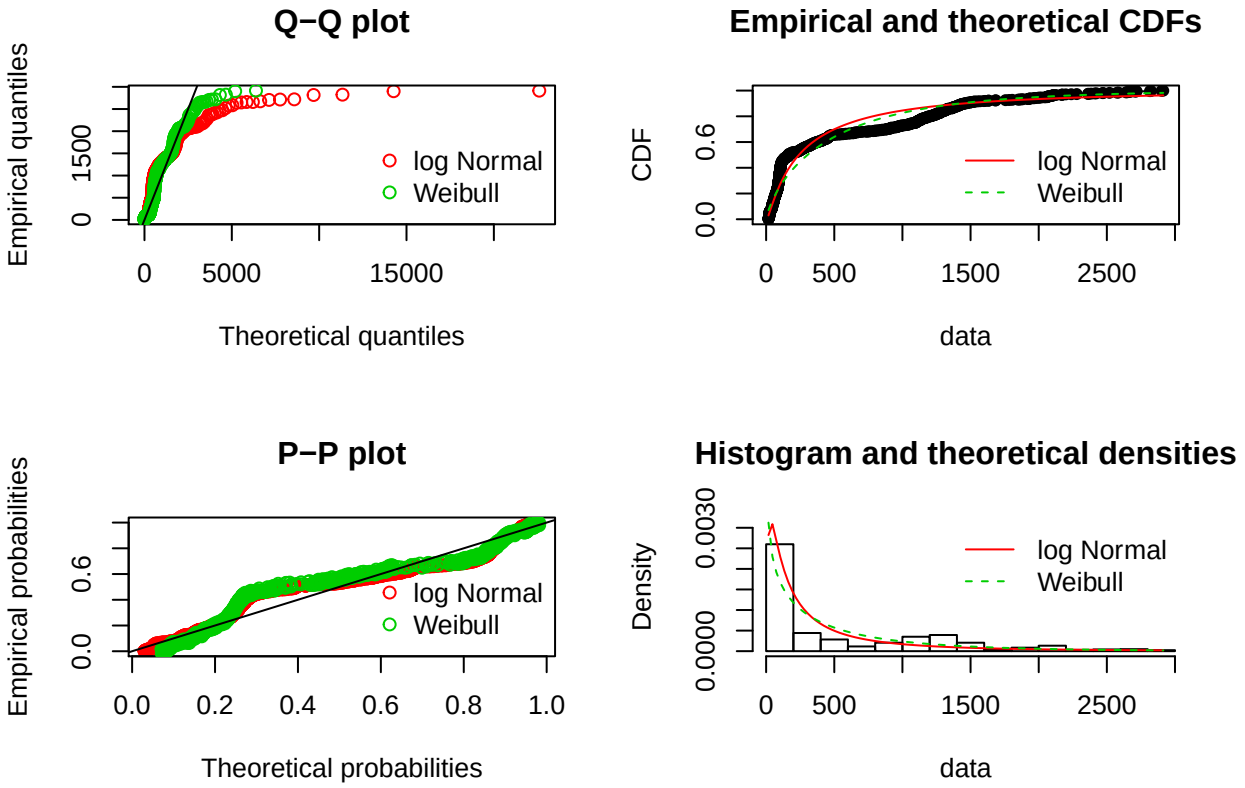
```
sp500.lognorm=fitdist(as.numeric(sp500_close),"lnorm")
sp500.weibull=fitdist(as.numeric(sp500_close), "weibull")
sp500.lognorm
```

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##         estimate Std. Error
## meanlog 5.481790 0.04881737
## sdlog   1.403871 0.03451901
```

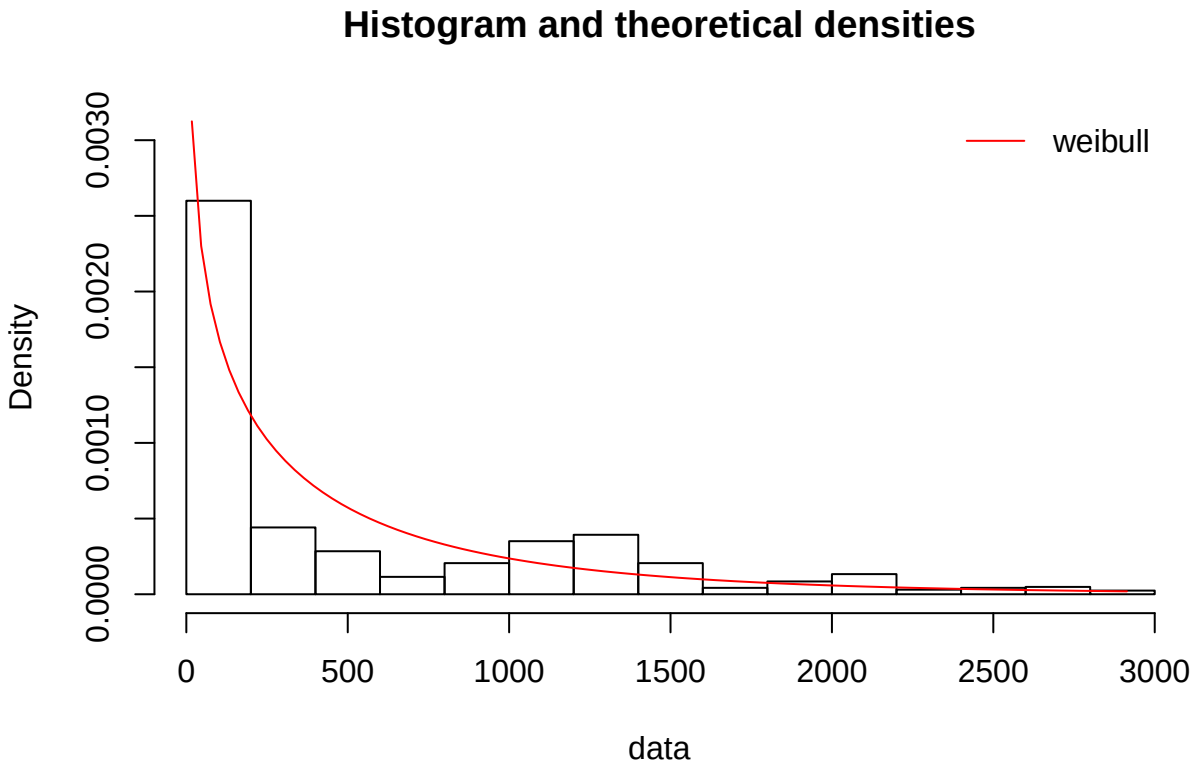
```
sp500.weibull
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##          estimate  Std. Error
## shape   0.7781503  0.02103345
## scale 486.6108434 23.03861789
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The histograms and theoretical densities graph above shows the histogram of the S&P500 data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.

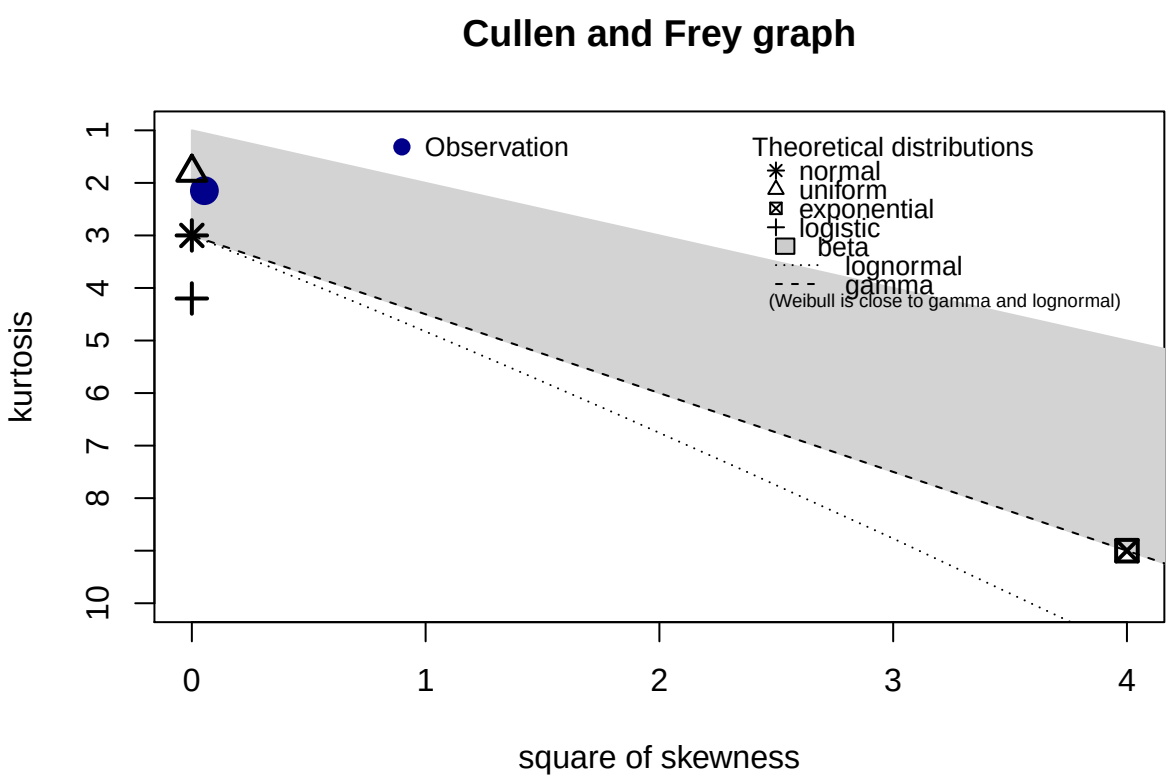


Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Weibull distribution.

We reused much of the code we created for the S&P500 index data for the three other data sets. Therefore, in order to prevent unnecessary redundancy much of the code is not shown again since only variable names are changed.

II) Fitting Distributions To The Yield Spread Data Set

For the Yield spread data, we again ran a Cullen-Frey test in order to determine which distributions our data fit the closest.



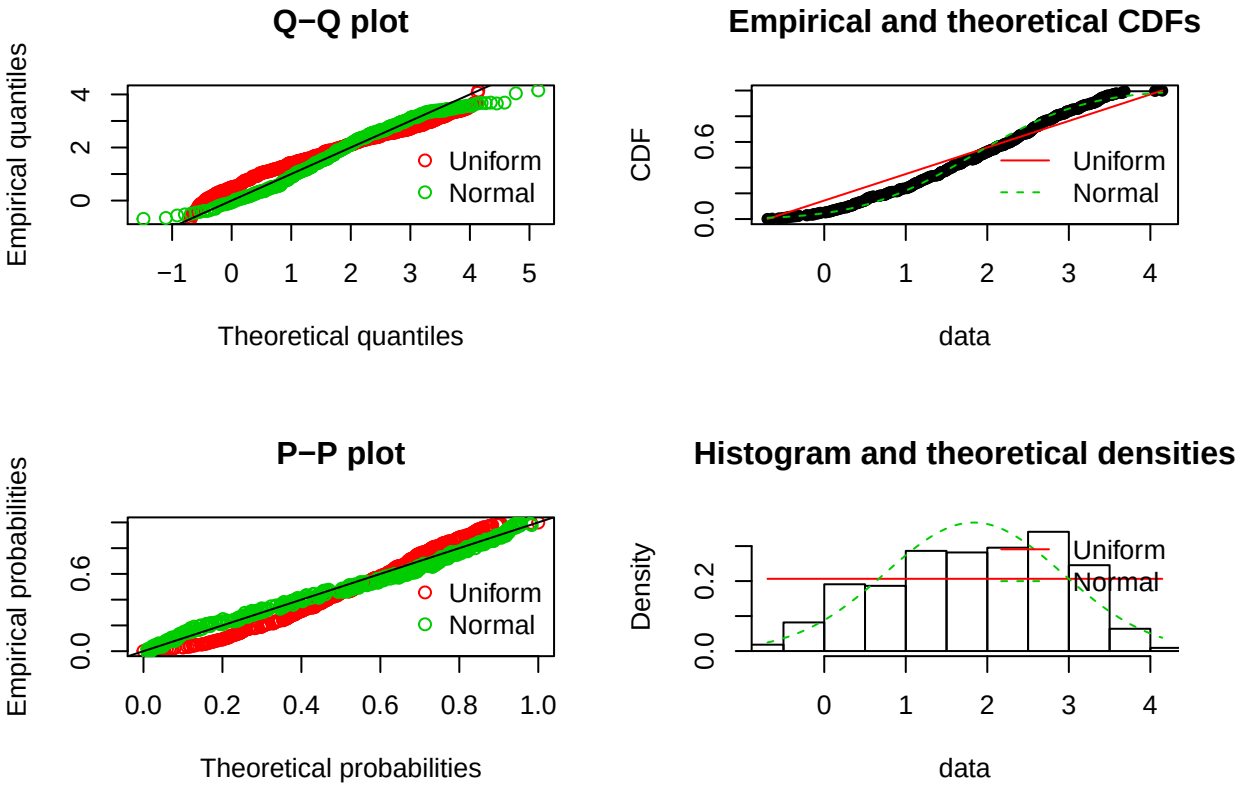

```
## summary statistics
## ------
## min:  -0.6965   max:  4.14619
## median:  1.9205
## mean:  1.83341
## estimated sd:  1.087209
## estimated skewness:  -0.2315276
## estimated kurtosis:  2.147746
```

Using the code we created before, we fit the distributions the Cullen-Frey test suggested. According to the Cullen-Frey graph a uniform or normal distribution would best fit our data. The fitdist command uses MLE to find the parameters that would best fit the distribution to the data.

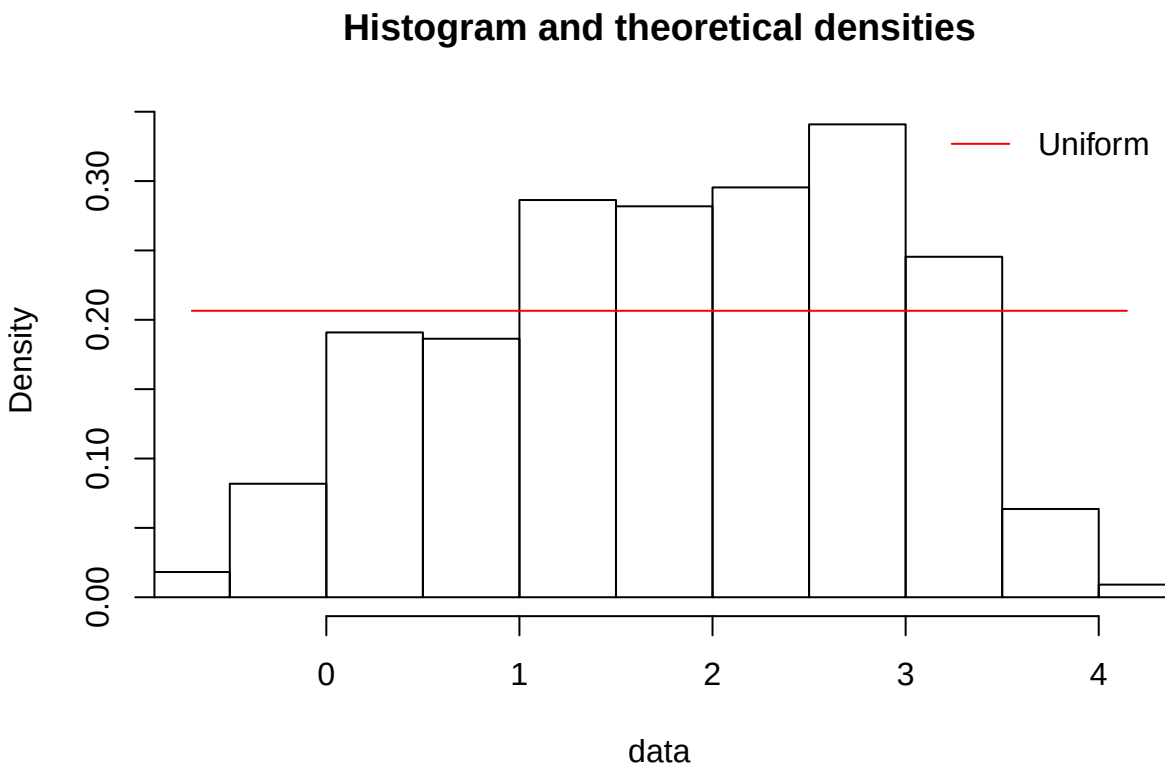
```
## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters:
##     estimate Std. Error
## min -0.69650         NA
## max  4.14619         NA
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## mean 1.833410 0.05177173
## sd   1.085973 0.03660800
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



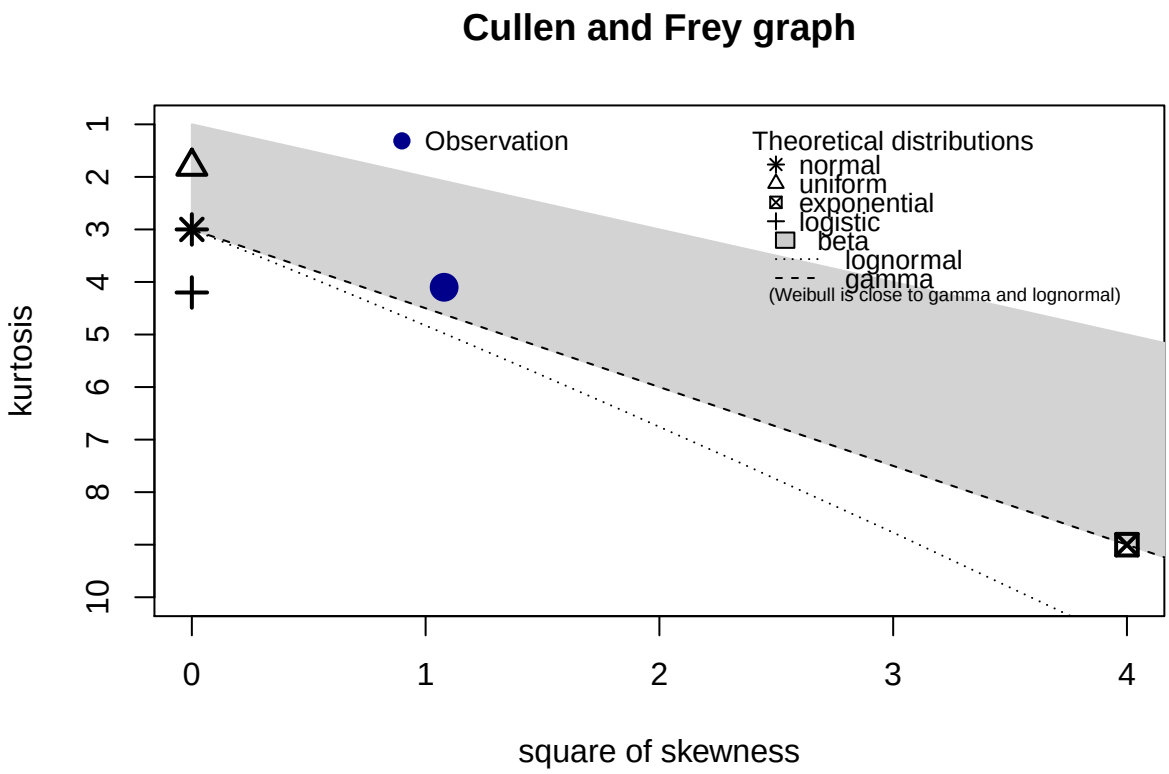
The graph below shows the histogram of the yield spread data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.



Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Uniform distribution.

III) Fitting Distributions To The Three Month Treasury Rate Data

The first thing we did is construct a Cullen-Frey graph using the data set.




```
## summary statistics
## ------
## min:  0.01   max:  16.3
## median:  2.97
## mean:  3.483972
## estimated sd:  3.16359
## estimated skewness:  1.038876
## estimated kurtosis:  4.100597
```

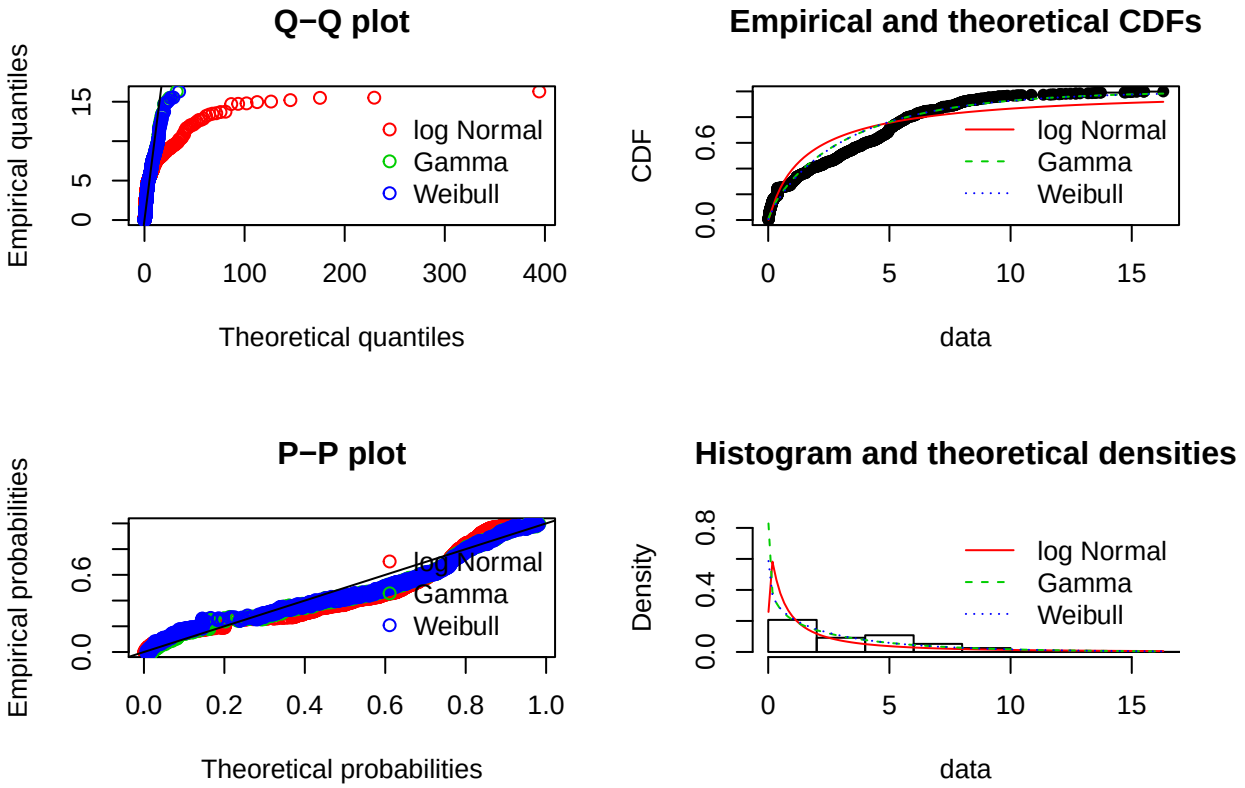
Using the same code we used for the other data sets, we fit the suggested distributions from the Cullen-Frey graph to the three month treasury rate data using MLE. The fitdist command uses MLE in order to find the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##          estimate Std. Error
## meanlog 0.4444436 0.05264957
## sdlog   1.6790178 0.03722881
```

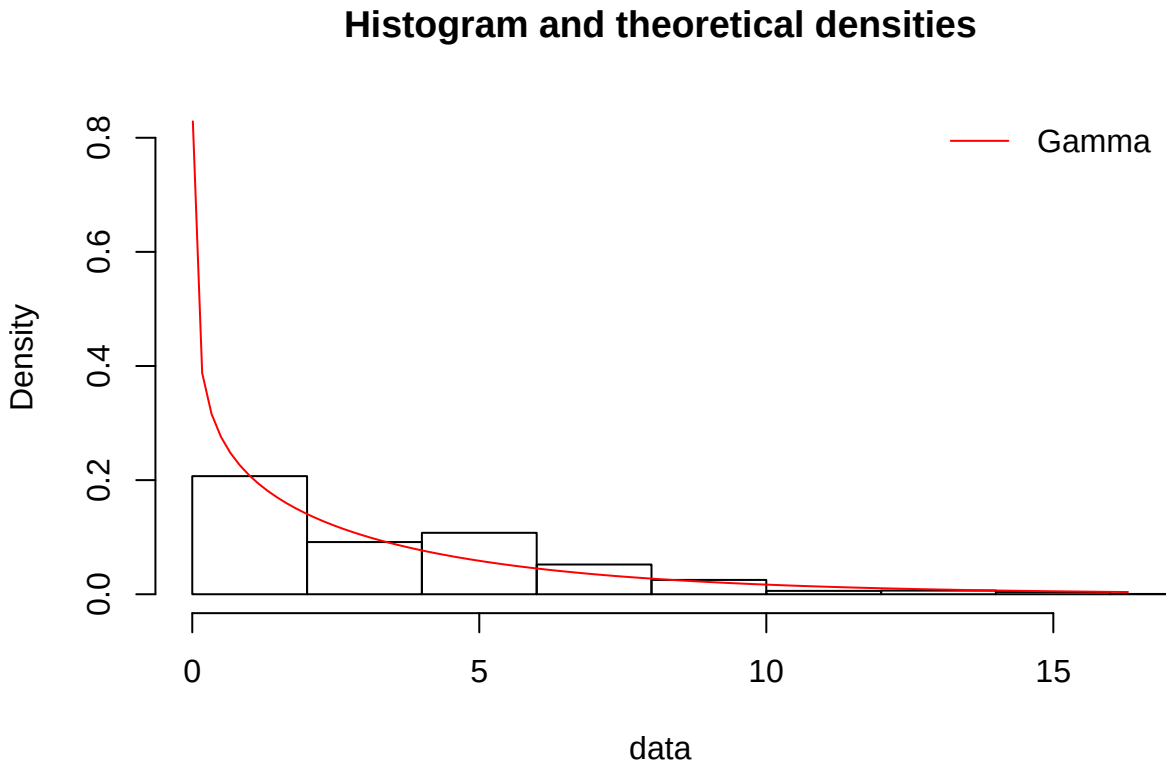
```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape 0.7454787 0.02833528
## rate  0.2139417 0.01124682
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape 0.8604127 0.02275481
## scale 3.2685126 0.12460675
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



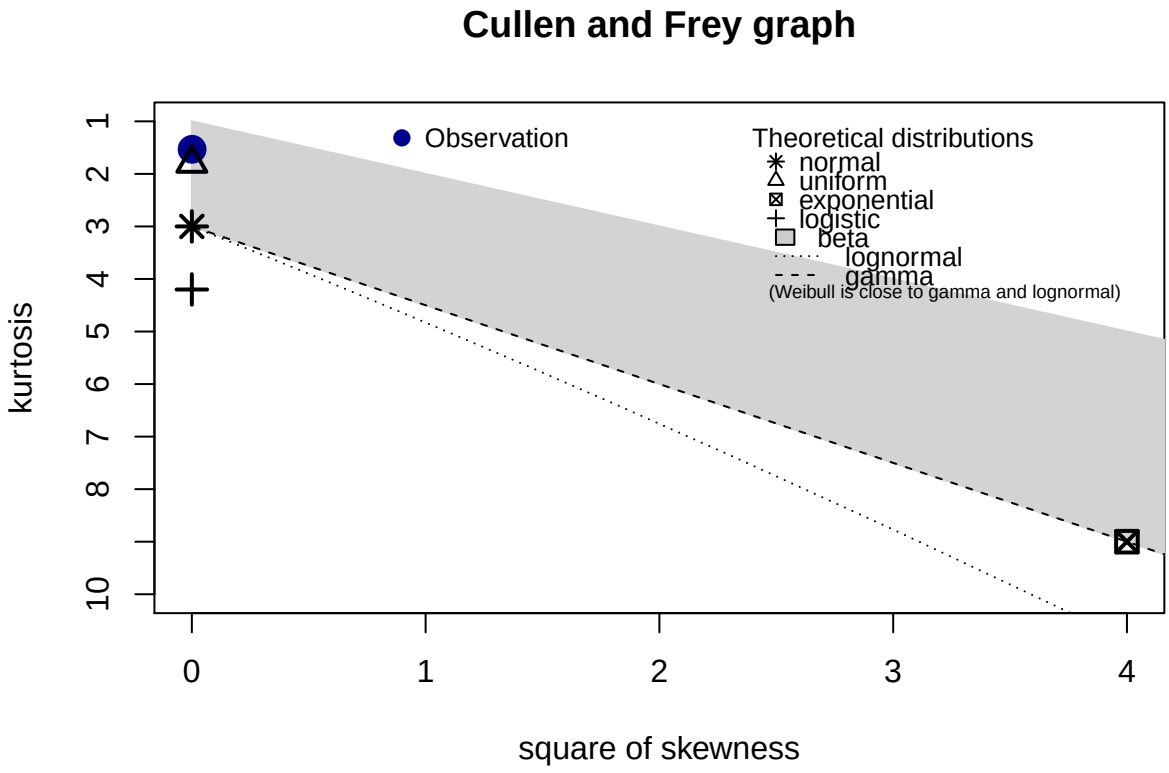
The graph below shows the histogram of the three month treasury bill data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.



Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Gamma distribution.

IV) Fitting Distributions to the Japanese Central Bank's Interest Rates

Similar to what we did with the other three data sets we used a Cullen-Frey test in order to determine which distributions our data closely represented.




```
## summary statistics
## ------
## min:  0.1   max:  9
## median:  4.5
## mean:  3.791943
## estimated sd:  2.859542
## estimated skewness:  -0.03282352
## estimated kurtosis:  1.532277
```

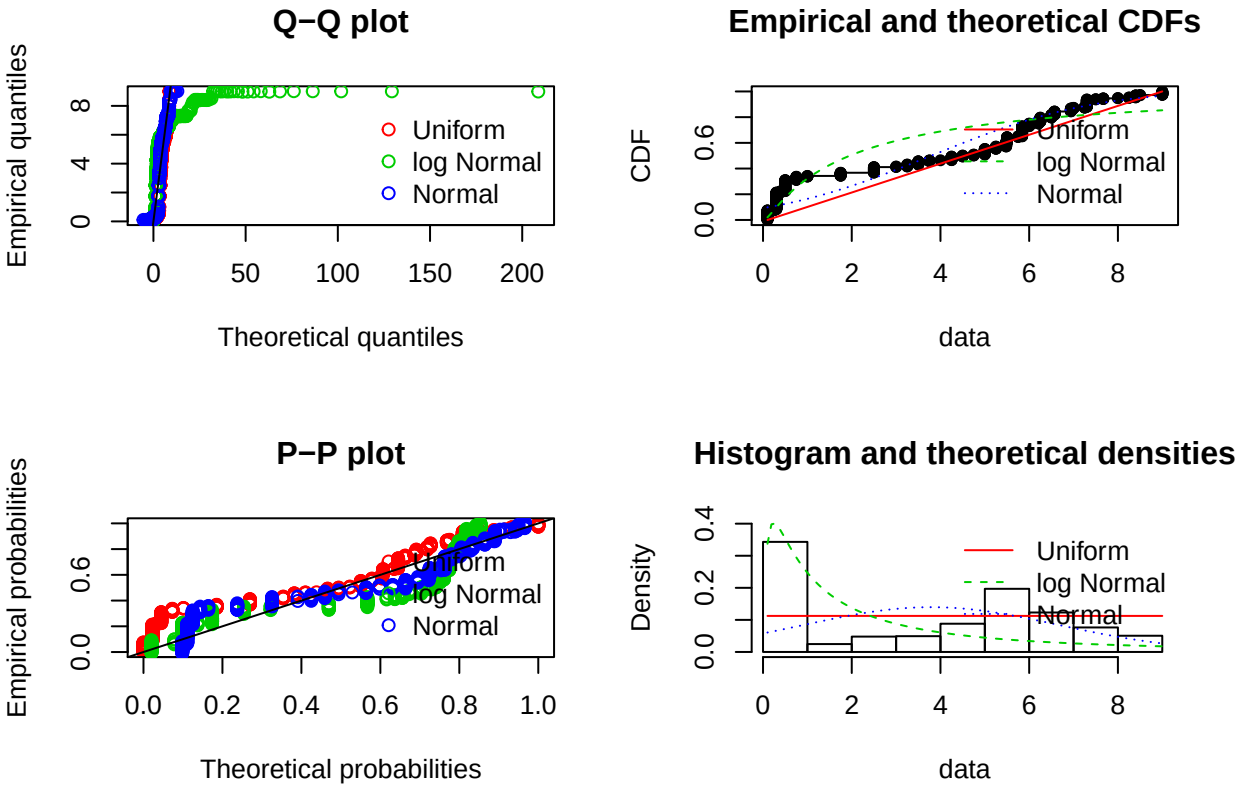
Using the same Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters:
##     estimate Std. Error
## min      0.1         NA
## max      9.0         NA
```

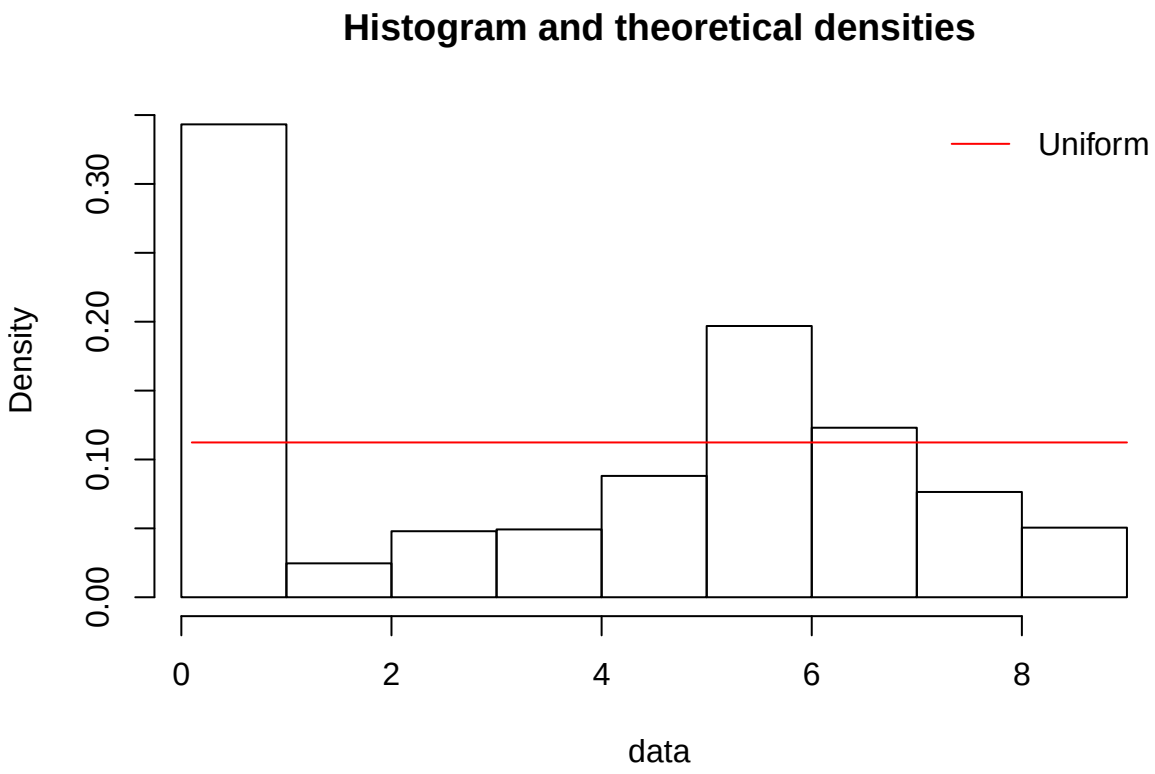
```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##         estimate Std. Error
## meanlog 0.670802 0.05224746
## sdlog   1.451690 0.03694446
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## mean 3.791943 0.10285049
## sd   2.857689 0.07272624
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the Japanese central bank interest rate data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.



Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Uniform distribution.

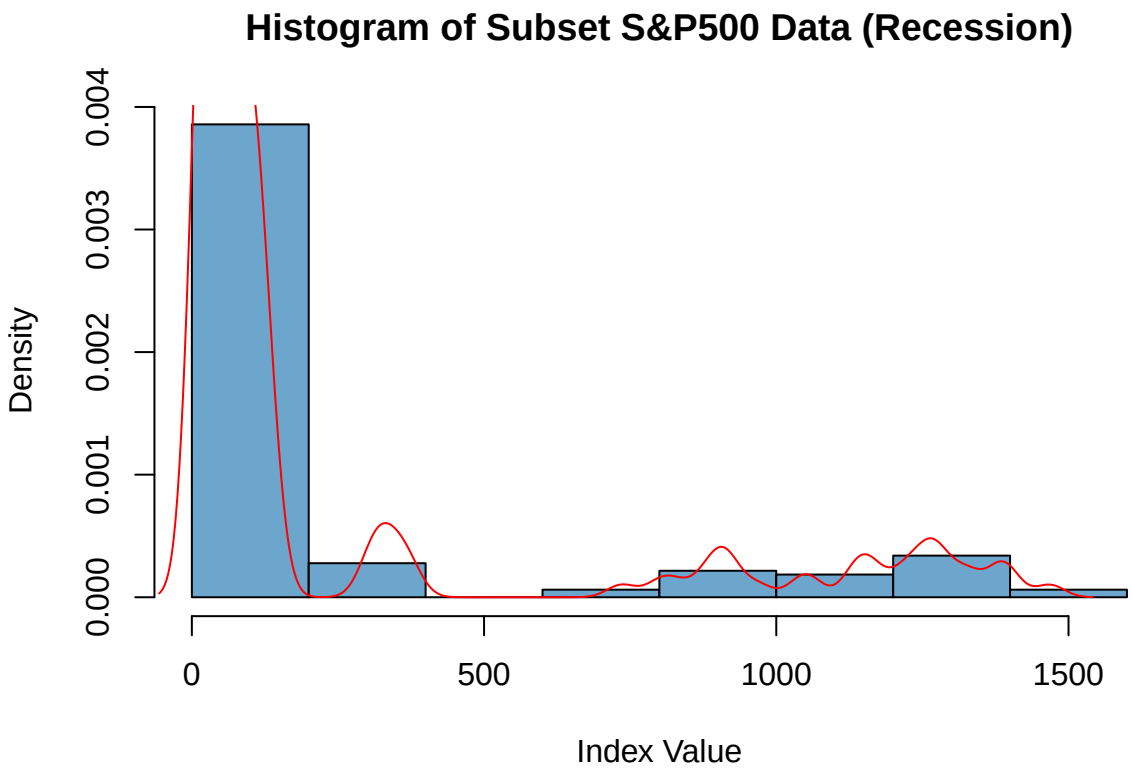
Part C) Subsetting The Data Into Period's of Recessions and Fitting Appropriate Distributions.

Subsetting S&P500 Index Data (recession)

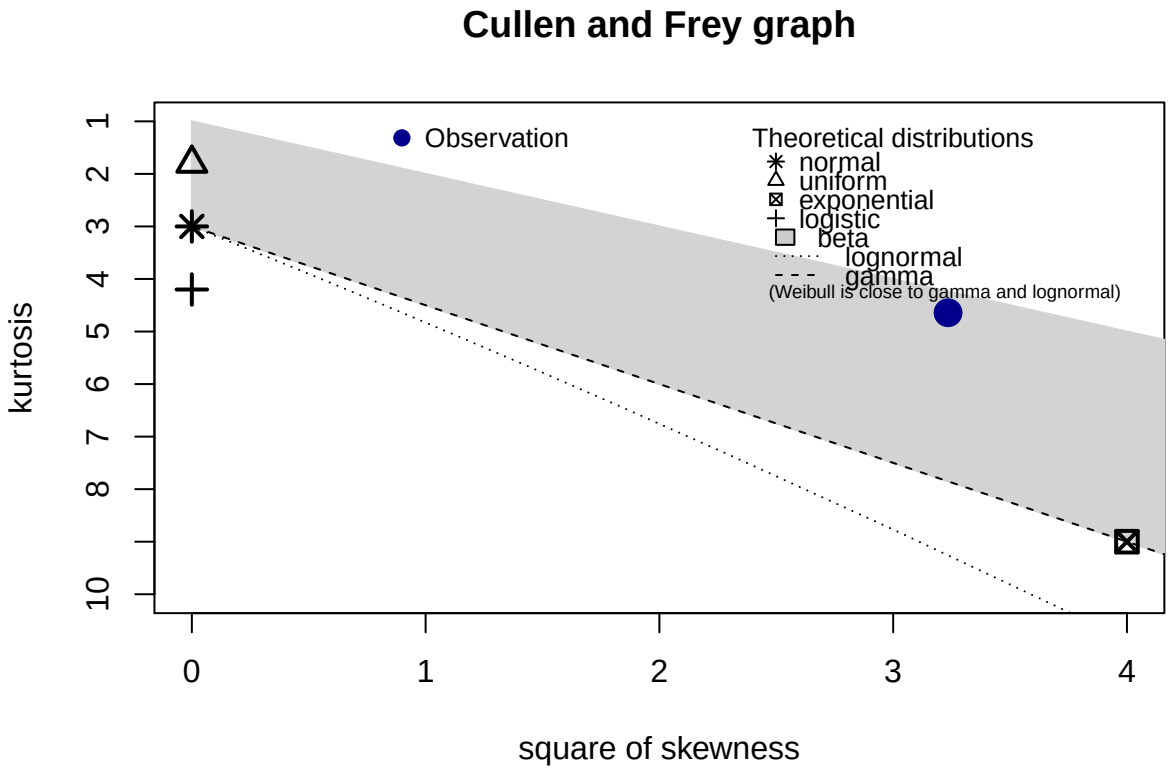
Using the following code we were able to subset the S&P500 data to include points generated only in times of recession. Using the FRED database we were also able to find the dates when the U.S. was experiencing a recession. We included these dates in our code, which can be seen below.

```
myfunc=function(x,y){sp500[sp500$Date >= x & sp500$Date <= y,]}
rec1=myfunc("1950-02-01","1954-05-01")
rec2=myfunc("1957-08-01","1958-04-01")
rec3=myfunc("1960-04-01","1961-02-01")
rec4=myfunc("1969-12-01","1970-11-01")
rec5=myfunc("1973-11-01","1975-03-01")
rec6=myfunc("1980-01-01","1980-07-01")
rec7=myfunc("1981-07-01","1982-11-01")
rec8=myfunc("1990-07-01","1991-03-01")
rec9=myfunc("2001-03-01","2001-11-01")
rec10=myfunc("2007-12-01","2009-06-01")
```

Below is a histogram of the subset S&P500 (recession) data along with its respective density curve.



Below is the Cullen-Frey graph we created using the subset (recession) S&P500 data. This graph shows us which distributions our data closely resembles.




```
## summary statistics
## ------
## min:  17.22   max:  1468.36
## median:  78.68
## mean:  259.6901
## estimated sd:  413.5424
## estimated skewness:  1.79856
## estimated kurtosis:  4.64286
```

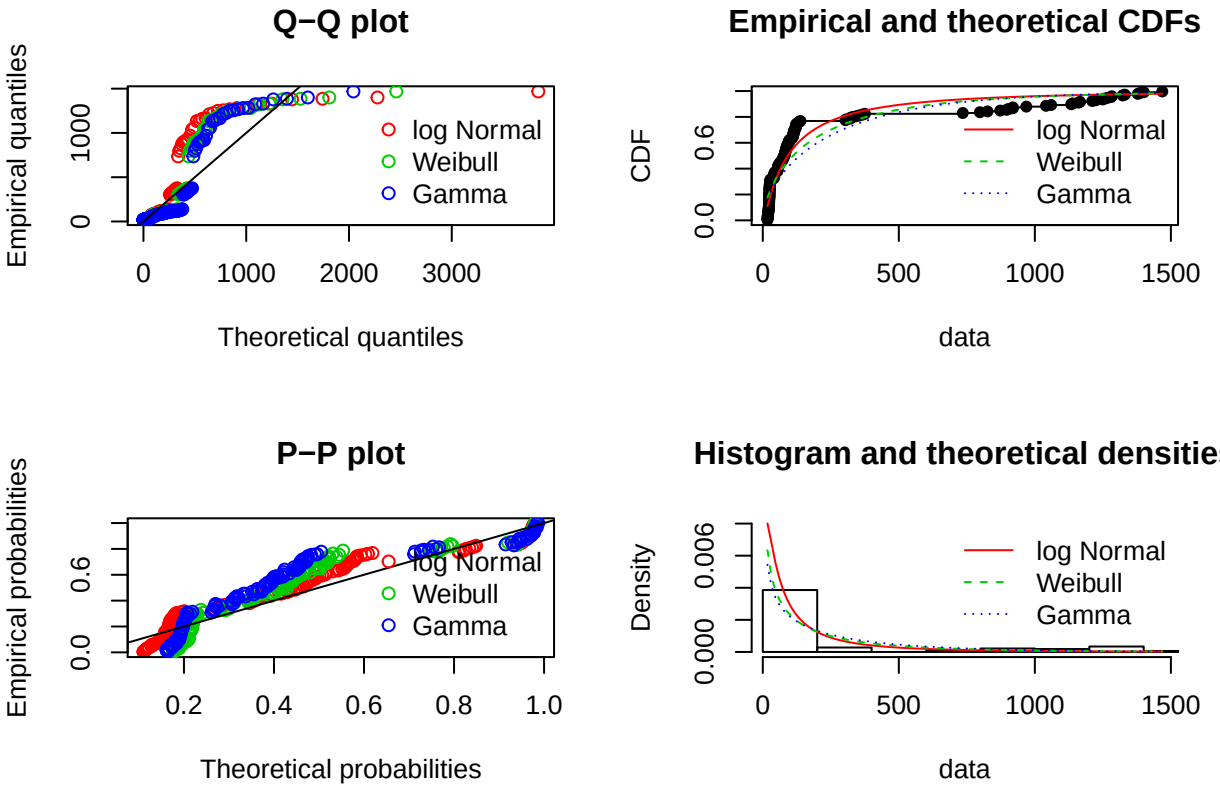
Using the following Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##         estimate Std. Error
## meanlog 4.518692 0.10715855
## sdlog   1.363906 0.07577235
```

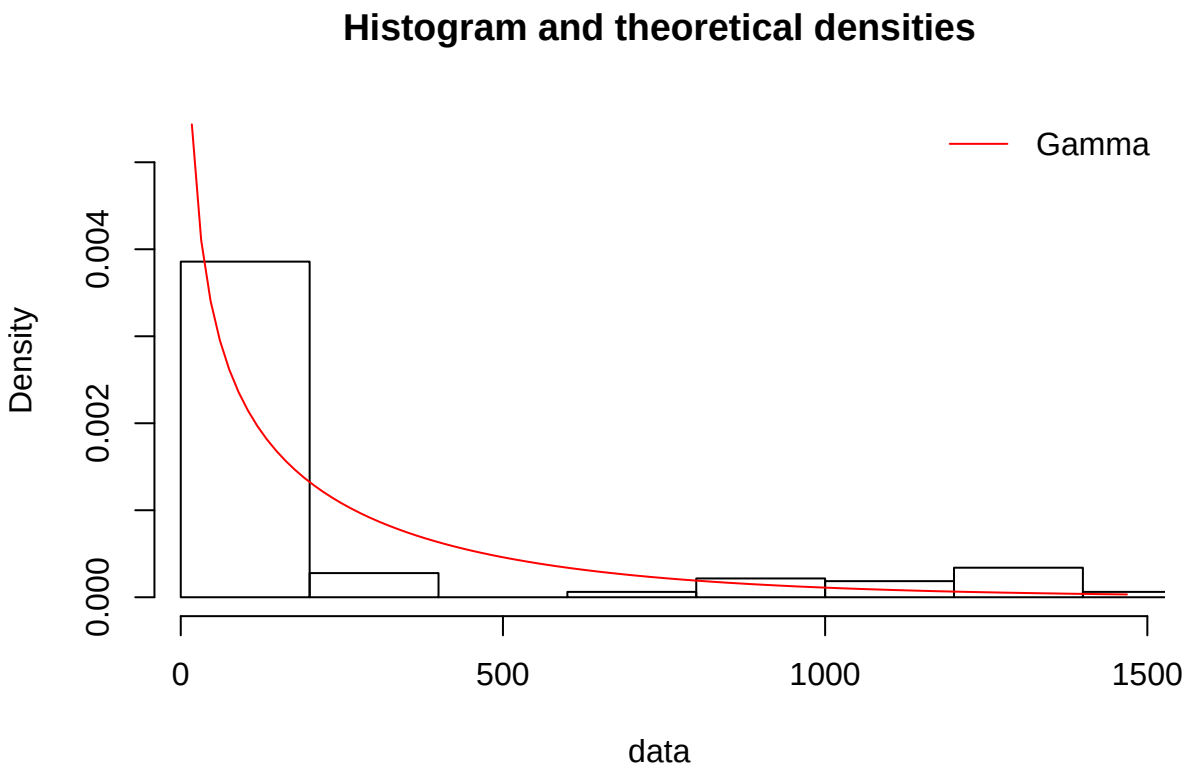
```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##          estimate  Std. Error
## shape   0.6848892  0.03913523
## scale 189.8568123 23.19475657
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##          estimate  Std. Error
## shape 0.594694653 0.048123162
## rate  0.002289748 0.000199823
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the S&P500 data (recession) along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.

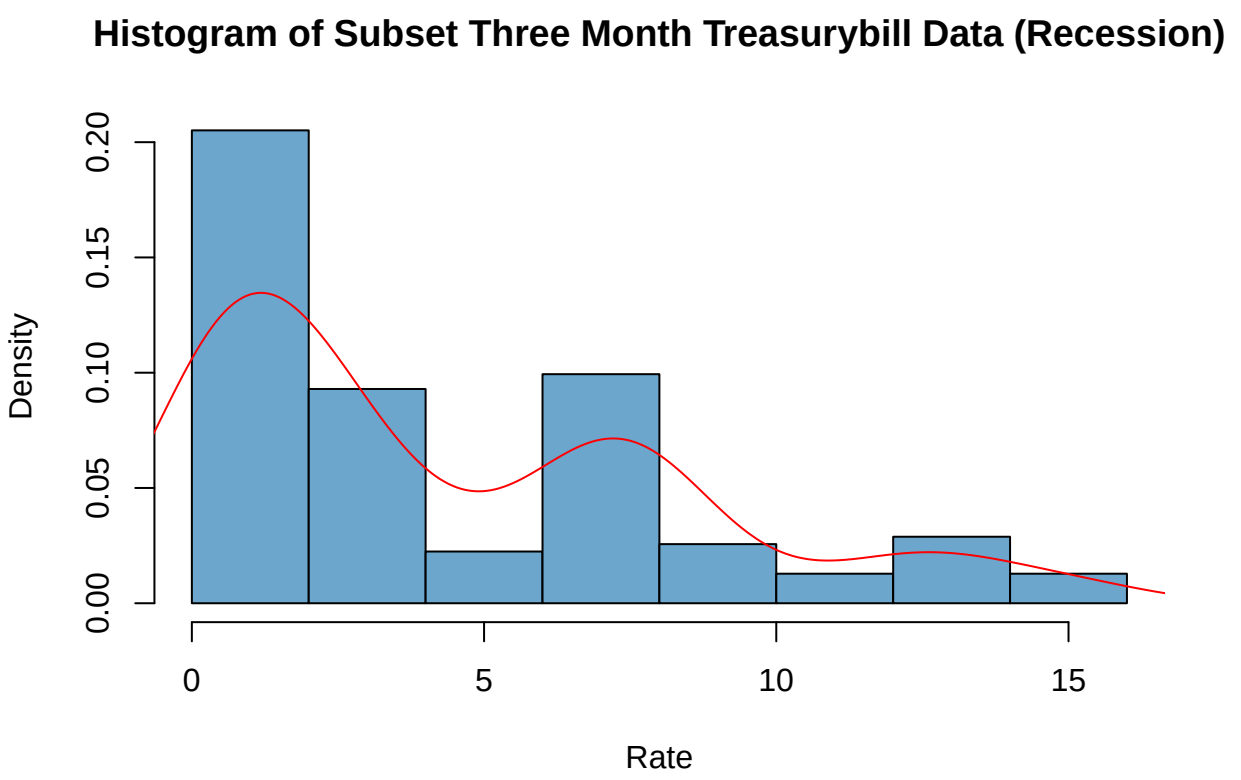


Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Gamma distribution.

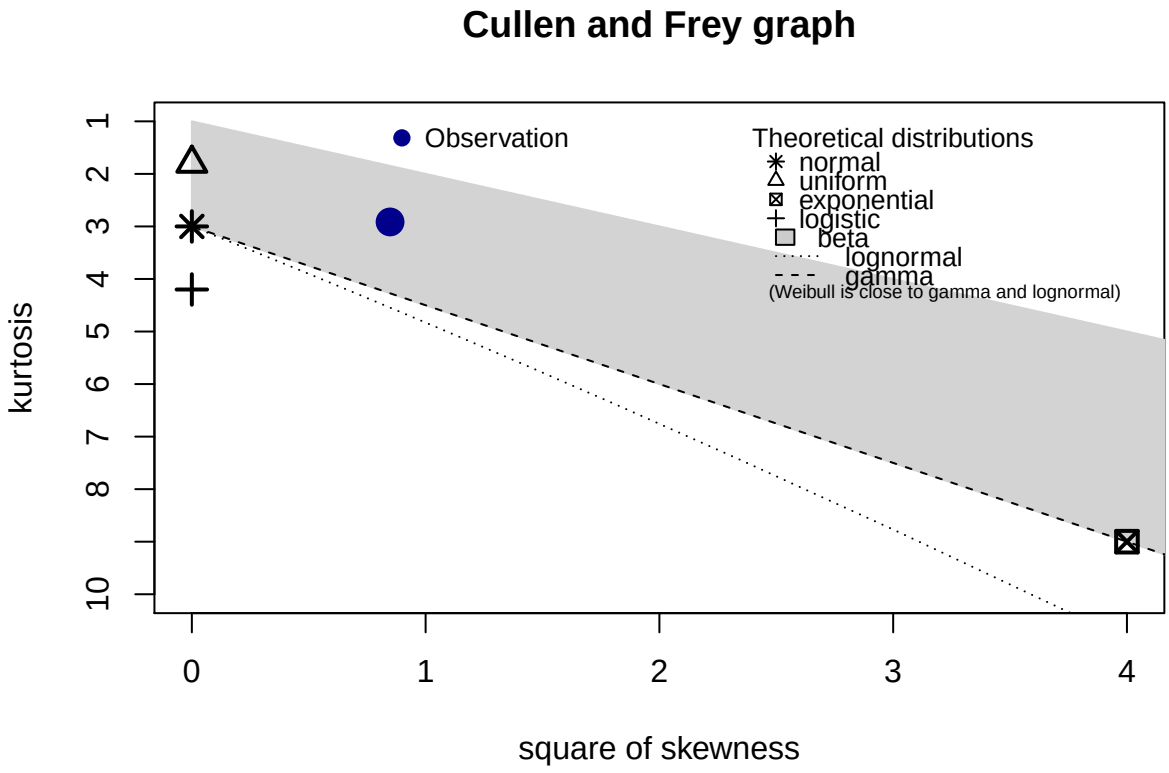
Subsetting Three Month Treasury Bill Data (recession)

Using the following code we were able to subset the three month treasury bill data to include points generated only in times of recession. Using the FRED database we were also able to find the dates when the U.S. was experiencing a recession. We included these dates in our code, which can be seen below.

```
myfunc=function(x,y){Three_treasury[Three_treasury$DATE >= x & Three_treasury$DATE <= y,]}
rec1=myfunc("1937-05-01","1938-06-01")
rec2=myfunc("1945-02-01","1945-10-01")
rec3=myfunc("1948-11-01","1949-10-01")
rec4=myfunc("1953-07-01","1954-05-01")
rec5=myfunc("1957-08-01","1958-04-01")
rec6=myfunc("1960-04-01","1961-02-01")
rec7=myfunc("1969-12-01","1970-11-01")
rec8=myfunc("1973-11-01","1975-03-01")
rec9=myfunc("1980-01-01","1980-07-01")
rec10=myfunc("1981-07-01","1982-11-01")
rec11=myfunc("1990-07-01","1991-03-01")
rec12=myfunc("2001-03-01","2001-11-01")
rec13=myfunc("2007-12-01","2009-06-01")
```



Below is the Cullen-Frey graph we created using the subset (recession) three month treasury bill data. This graph shows us which distributions our data closely resembles.



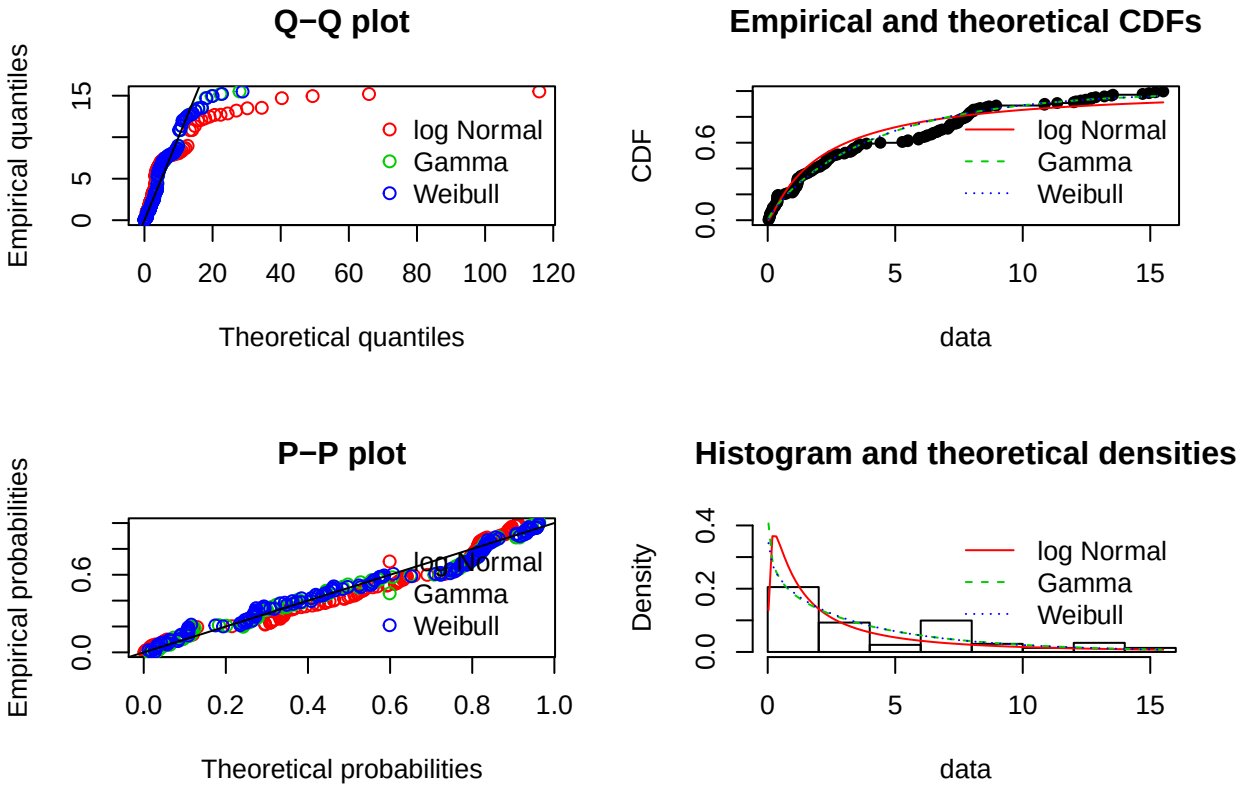

```
## summary statistics
## ------
## min:  0.03   max:  15.51
## median:  2.56
## mean:  4.355064
## estimated sd:  4.111714
## estimated skewness:  0.9208294
## estimated kurtosis:  2.914588
```

With the following Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

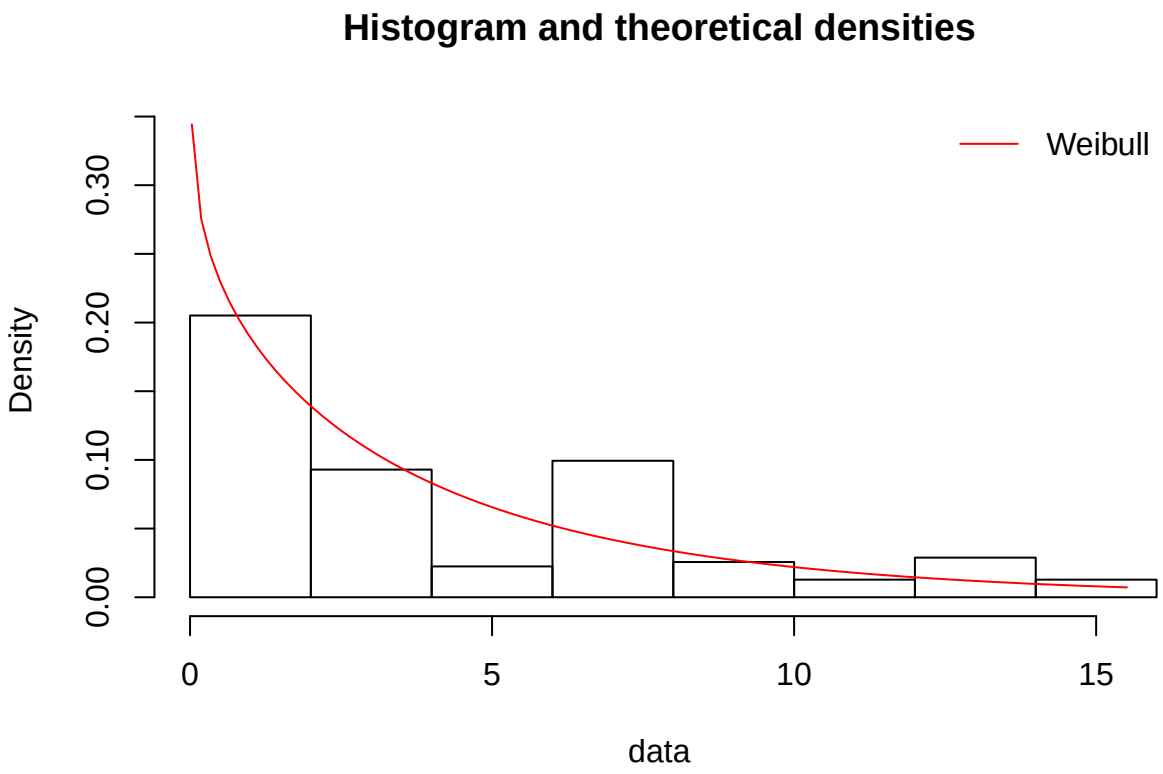
```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##          estimate Std. Error
## meanlog 0.7570337 0.11735426
## sdlog   1.4657543 0.08298182

## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape 0.8272170 0.08106208
## rate  0.1898985 0.02501377
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape 0.9041584 0.05929365
## scale 4.1651402 0.38755657
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the three month treasury bill data (recession) along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.

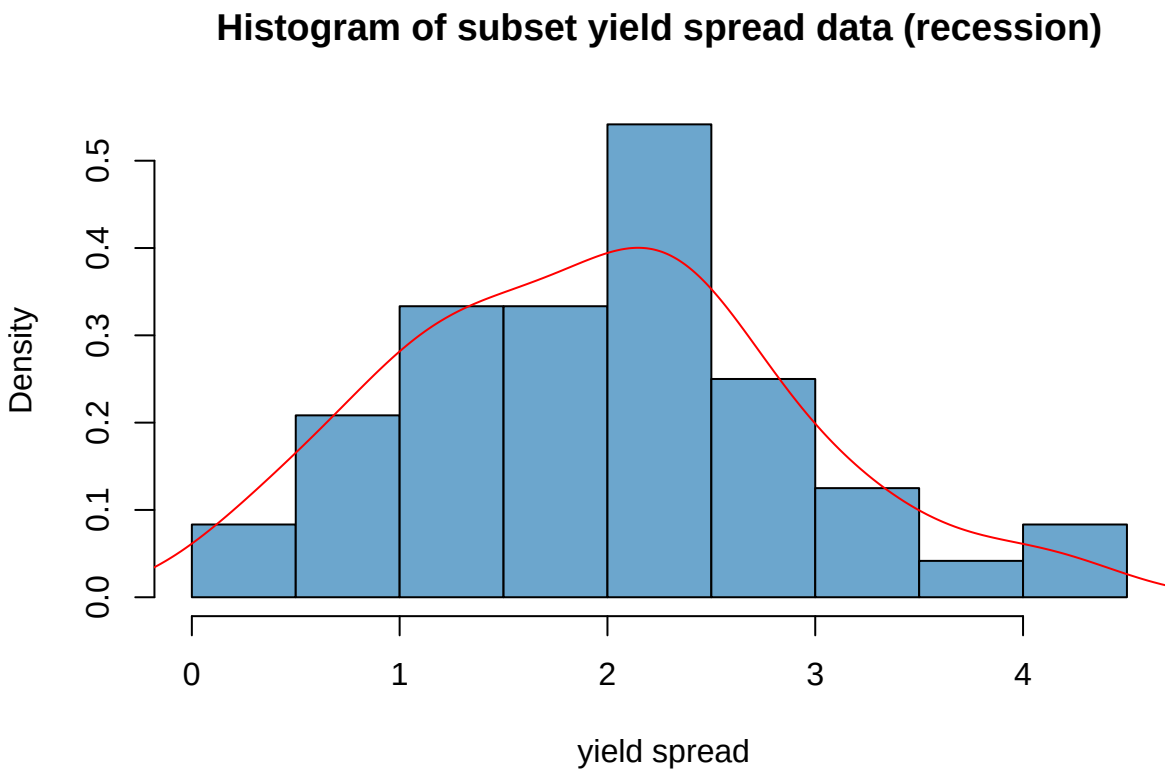


Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Weibull distribution.

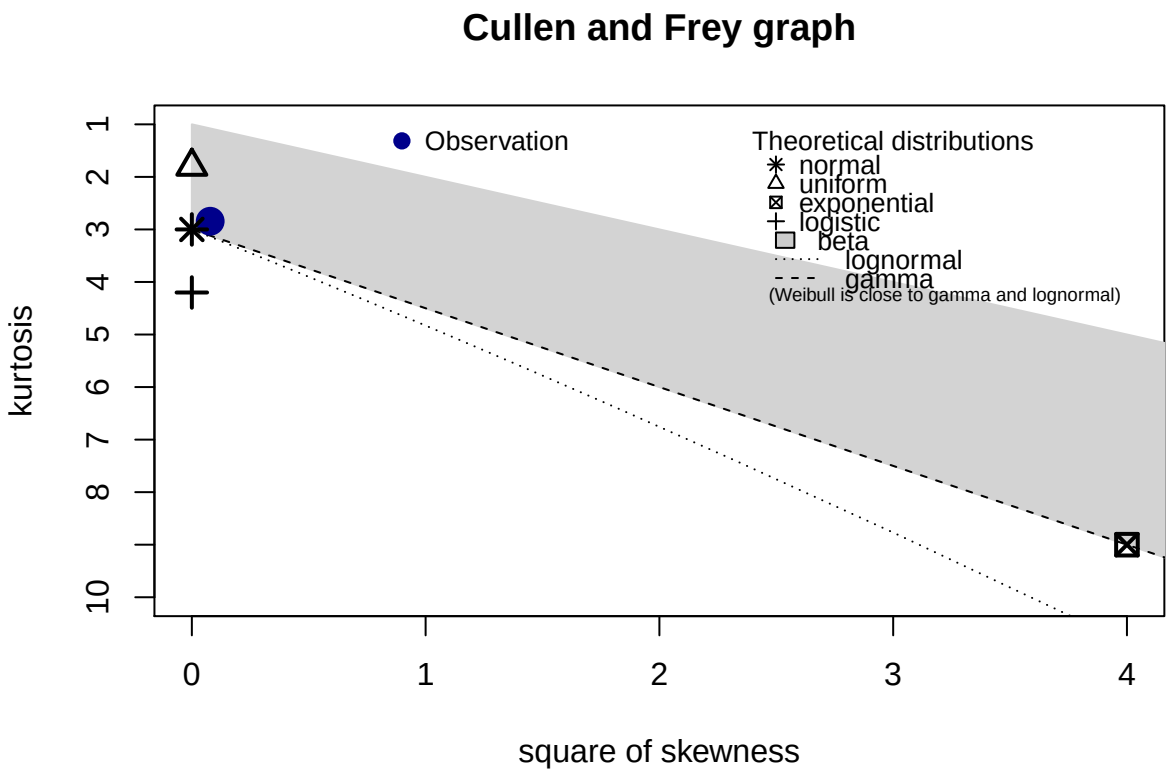
Subsetting Yield spread Data (recession)

Using the following code we were able to subset the Yield spread data to include points generated only in times of recession. Using the FRED database we were also able to find the dates when the U.S. was experiencing a recession. We included these dates in our code, which can be seen below.

```
myfunc=function(x,y){Yield_Spread[Yield_Spread$DATE >= x & Yield_Spread$DATE <= y,]}
rec1=myfunc("1960-04-01","1961-02-01")
rec2=myfunc("1969-12-01","1970-11-01")
rec3=myfunc("1973-11-01","1975-03-01")
rec4=myfunc("1980-01-01","1980-07-01")
rec5=myfunc("1982-01-01","1982-11-01")
rec6=myfunc("1990-07-01","1991-03-01")
rec7=myfunc("2001-03-01","2001-11-01")
rec8=myfunc("2007-12-01","2009-06-01")
```



Below is the Cullen-Frey graph we created using the subset (recession) yield spread data. This graph shows us which distributions our data closely resembles.




```
## summary statistics
## ------
## min:  0.1461111   max:  4.14619
## median:  2.027956
## mean:  1.936405
## estimated sd:  0.929337
## estimated skewness:  0.2802915
## estimated kurtosis:  2.844028
```

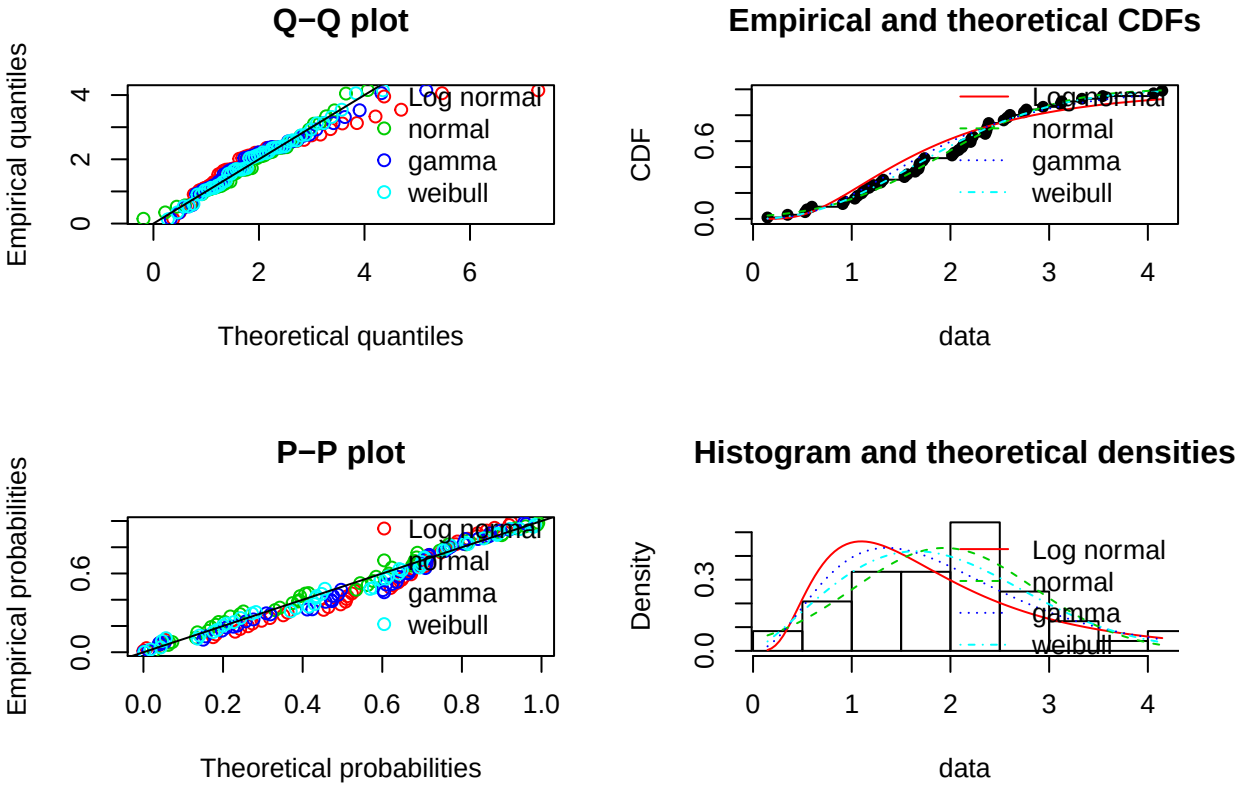
Using the following Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##          estimate Std. Error
## meanlog 0.5037561 0.09263803
## sdlog   0.6418151 0.06550426
```

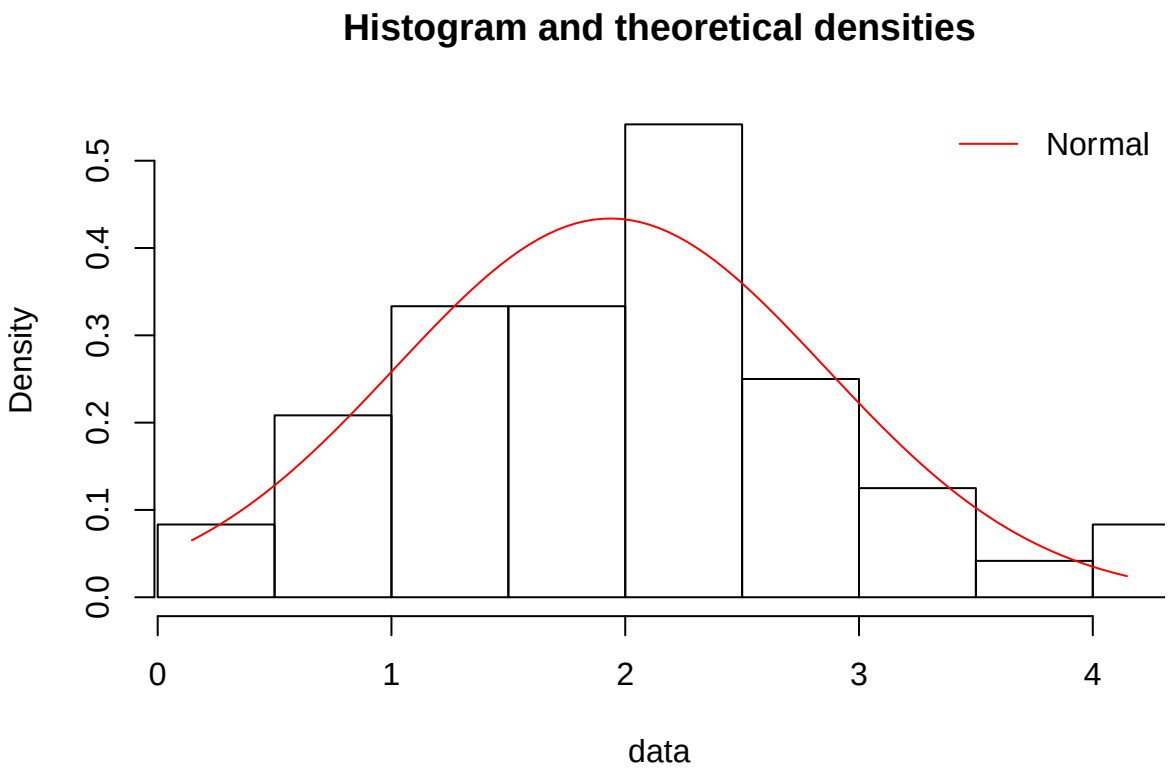
```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##       estimate Std. Error
## mean 1.9364052 0.13273361
## sd   0.9196054 0.09385633
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##       estimate Std. Error
## shape 3.340984  0.6508036
## rate  1.725347  0.3626528

## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##       estimate Std. Error
## shape 2.197008  0.2535973
## scale 2.179404  0.1501533
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the yield spread data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.



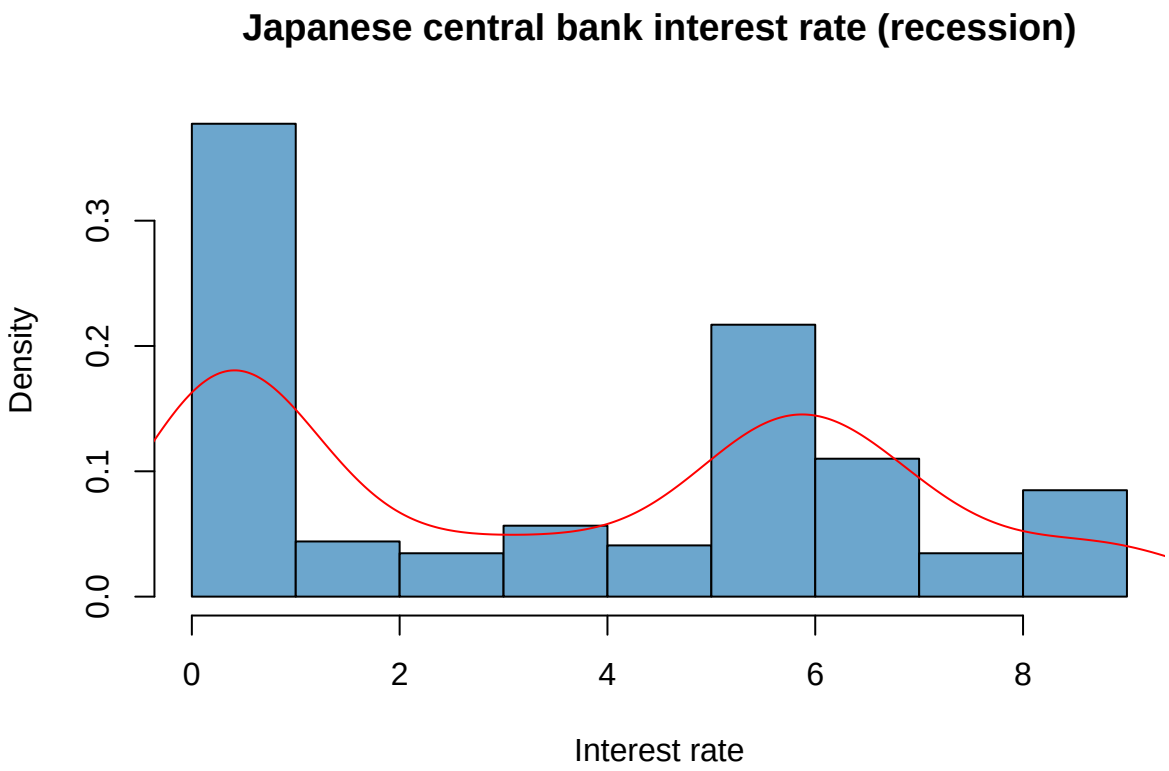
Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Normal distribution.

Subsetting Japanese Central Bank Interest Rate data (based on OECD recession indicators)

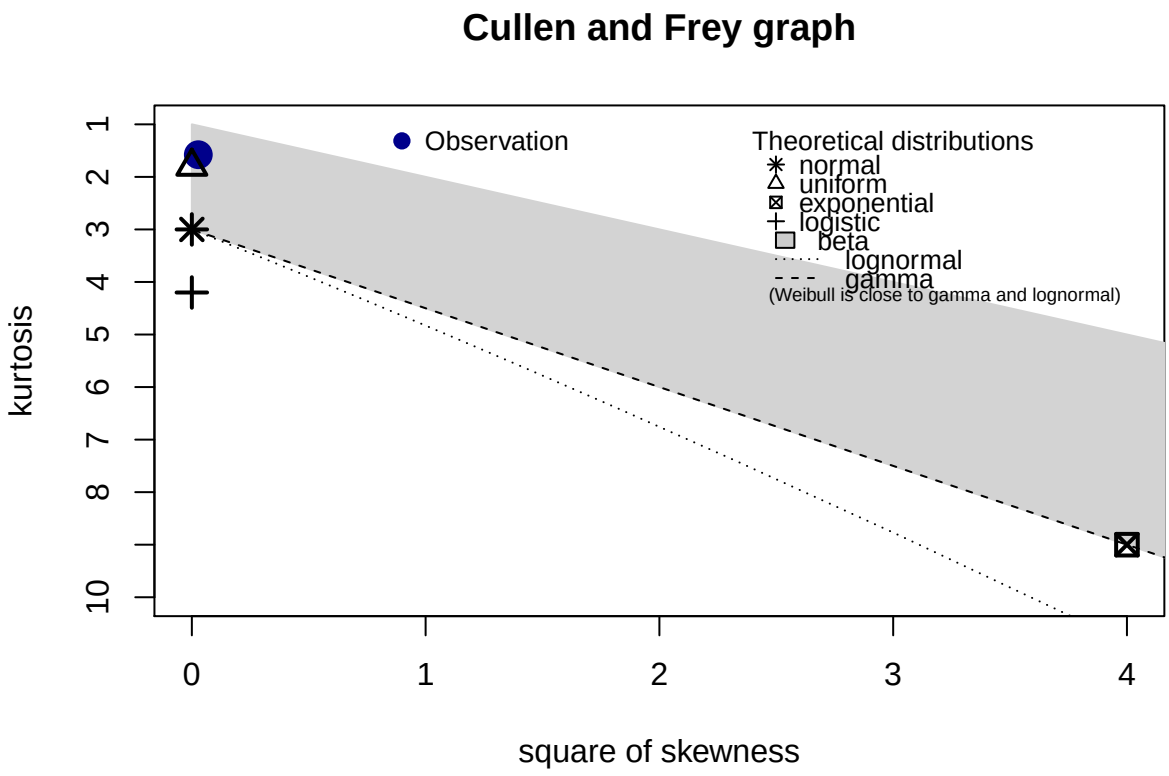
Using the following code we were able to subset the Japanese Central bank interest rate data to include points generated only in times of recession. Using the FRED database we were also able to find the dates when when Japan experienced a recession (according to OECD recession based indicators). We included these dates in our code, which can be seen below.

```
myfunc=function(x,y){Japanese_central_bank[Japanese_central_bank$DATE >= x & Japanese_central_bank$DATE
rec1=myfunc("1953-07-01","1954-05-01")
rec2=myfunc("1957-08-01","1958-04-01")
rec3=myfunc("1960-11-01","1961-02-01")
rec4=myfunc("1961-12-01","1963-02-01")
rec5=myfunc("1964-04-01","1965-11-01")
rec6=myfunc("1970-03-01","1971-10-01")
rec7=myfunc("1973-04-01","1975-02-01")
rec8=myfunc("1979-06-01","1980-05-01")
rec9=myfunc("1982-03-01","1983-05-01")
rec10=myfunc("1985-09-01","1987-02-01")
rec11=myfunc("1990-08-01","1994-10-01")
rec12=myfunc("1997-01-01","1999-05-01")
rec13=myfunc("2001-01-01","2002-01-01")
rec14=myfunc("2004-03-01","2004-12-01")
```

```
rec15=myfunc("2008-02-01","2009-04-01")
rec16=myfunc("2010-08-01","2012-09-01")
rec17=myfunc("2013-10-01","2015-12-01")
```



Below is the Cullen-Frey graph we created using the subset (recession) Japanese central bank interest rate data. This graph shows us which distributions our data closely resembles. Using it we can determine which graph will best fit out data.




```
## summary statistics
## ------
## min:  0.1   max:  9
## median:  3.625
## mean:  3.637264
## estimated sd:  2.984438
## estimated skewness:  0.1662902
## estimated kurtosis:  1.577699
```

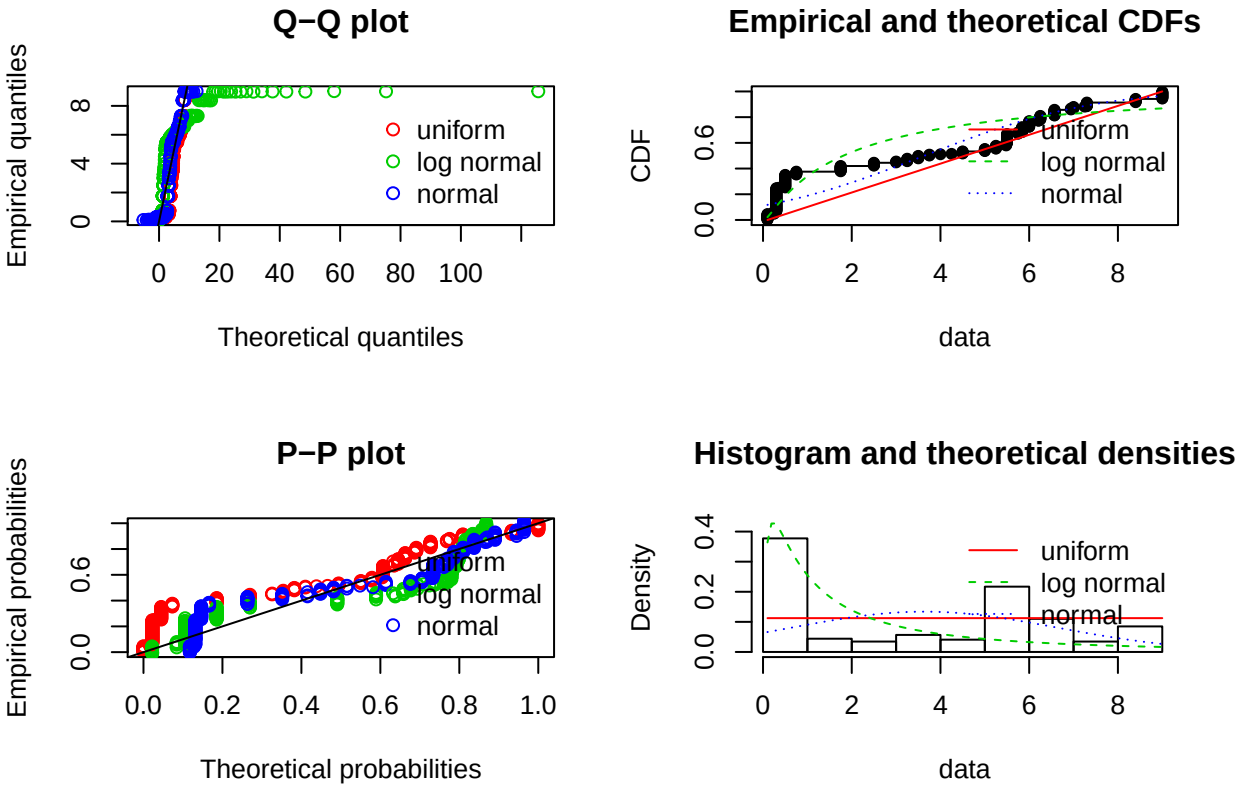
Using the following Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters:
##     estimate Std. Error
## min      0.1         NA
## max      9.0         NA
```

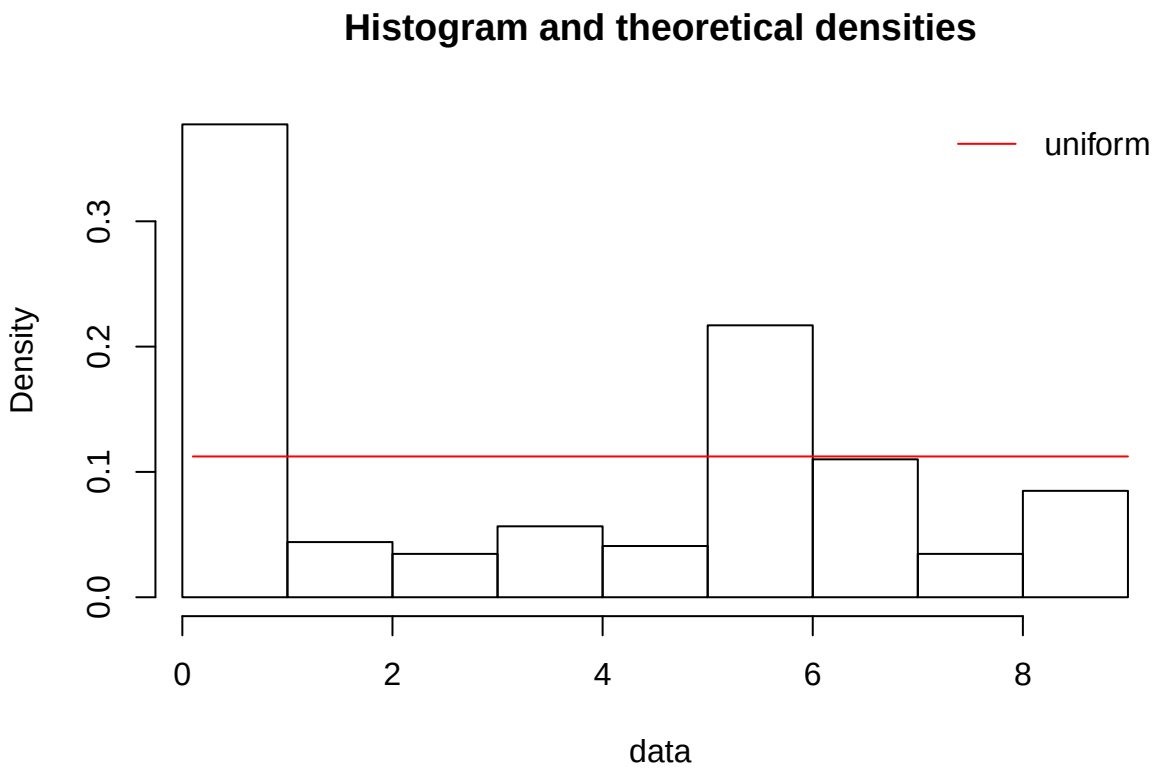
```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##          estimate Std. Error
## meanlog 0.5925758 0.08052718
## sdlog   1.4360053 0.05694119
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## mean 3.637264  0.1670956
## sd   2.979742  0.1181544
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the Japanese central bank interest rate (recession) data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.



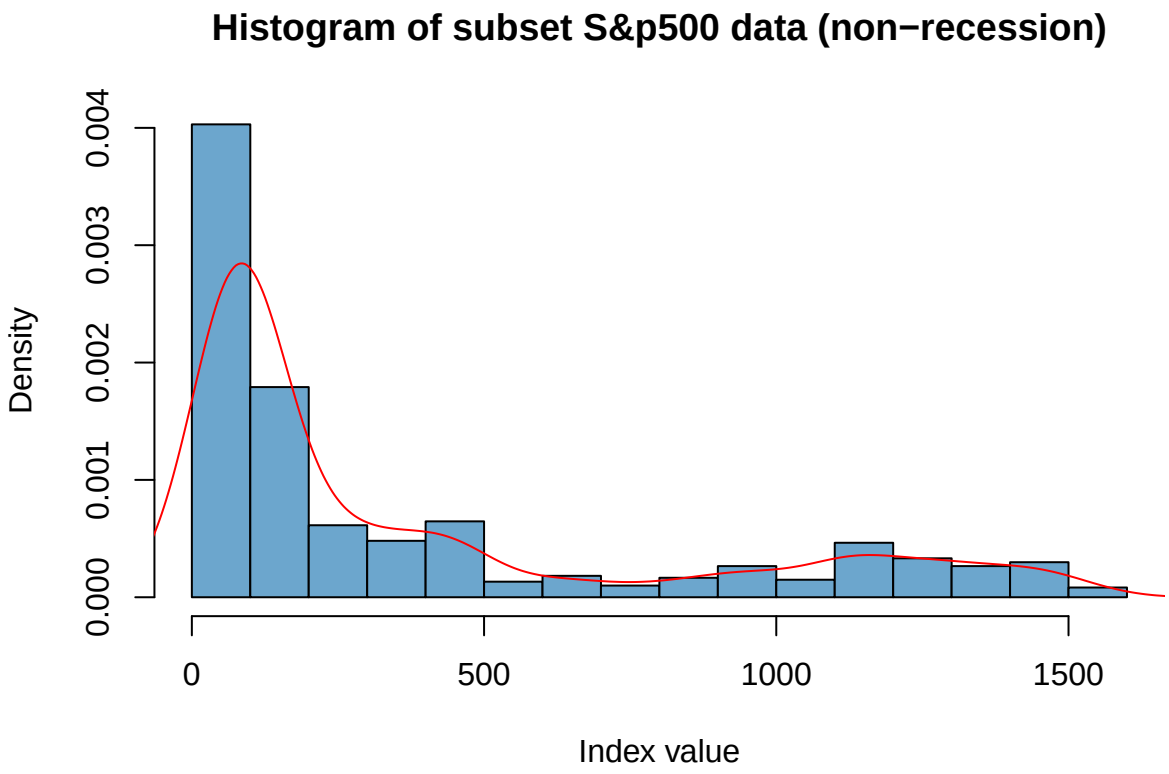
Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Uniform distribution.

Part D) Subsetting The Data Into Period's of non-Recession and Fitting Appropriate Distributions.

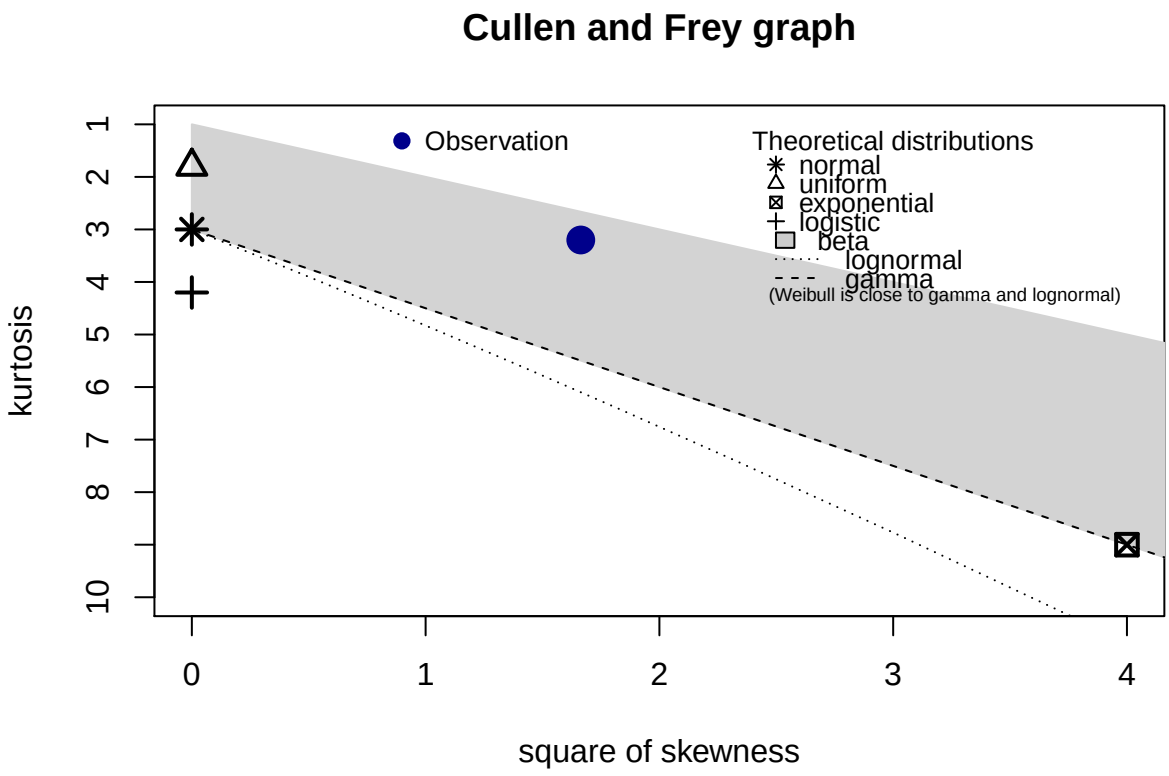
Subsetting S&P500 Index Data (non-recession)

We again subset our S&P500 data, but this time into periods where the economy is not in a recession. Using the FRED database we were able to calculate the dates when the economy is not in a recession.The code used to subset the data as well as the dates can be seen below.

```
myfunc=function(x,y){sp500[sp500$Date >= x & sp500$Date <= y,]}
rec1=myfunc("1950-10-01","1953-07-01")
rec2=myfunc("1954-05-01","1957-08-01")
rec3=myfunc("1958-04-01","1960-04-01")
rec4=myfunc("1961-02-01","1969-12-01")
rec5=myfunc("1970-11-01","1973-11-01")
rec6=myfunc("1975-03-01","1980-01-01")
rec7=myfunc("1980-07-01","1981-07-01")
rec8=myfunc("1982-11-01","1990-07-01")
rec9=myfunc("1991-03-01","2001-03-01")
rec10=myfunc("2001-11-01","2007-12-01")
rec11=myfunc("2009-06-01","2018-08-01")
```



Below is the Cullen-Frey graph we created using the subset (non-recession) S&P500 data. This graph shows us which distributions our data closely resembles and therefore which distribution best represents our data.




```
## summary statistics
## ------
## min:  19.51   max:  1549.38
## median:  118.05
## mean:  378.9348
## estimated sd:  443.9672
## estimated skewness:  1.289864
## estimated kurtosis:  3.200738
```

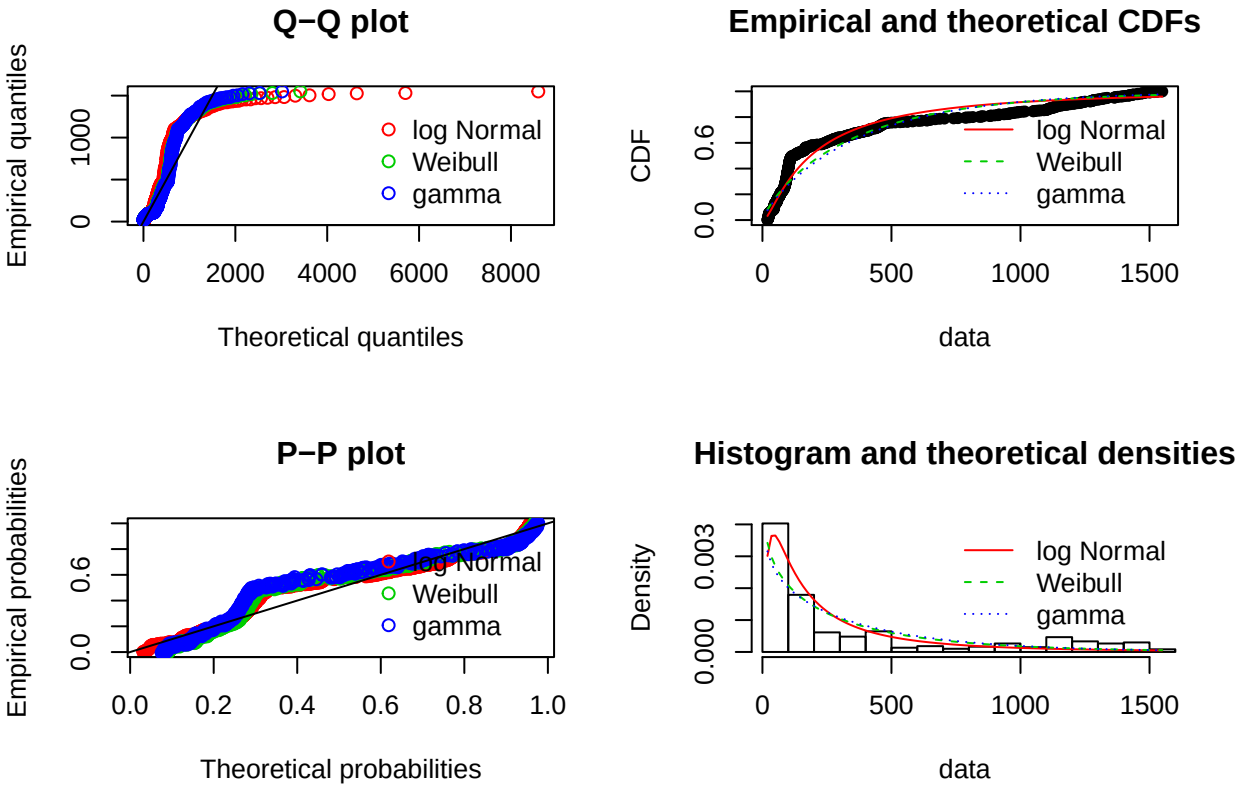
Using the following Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##         estimate Std. Error
## meanlog 5.228544 0.04959039
## sdlog   1.217745 0.03506560
```

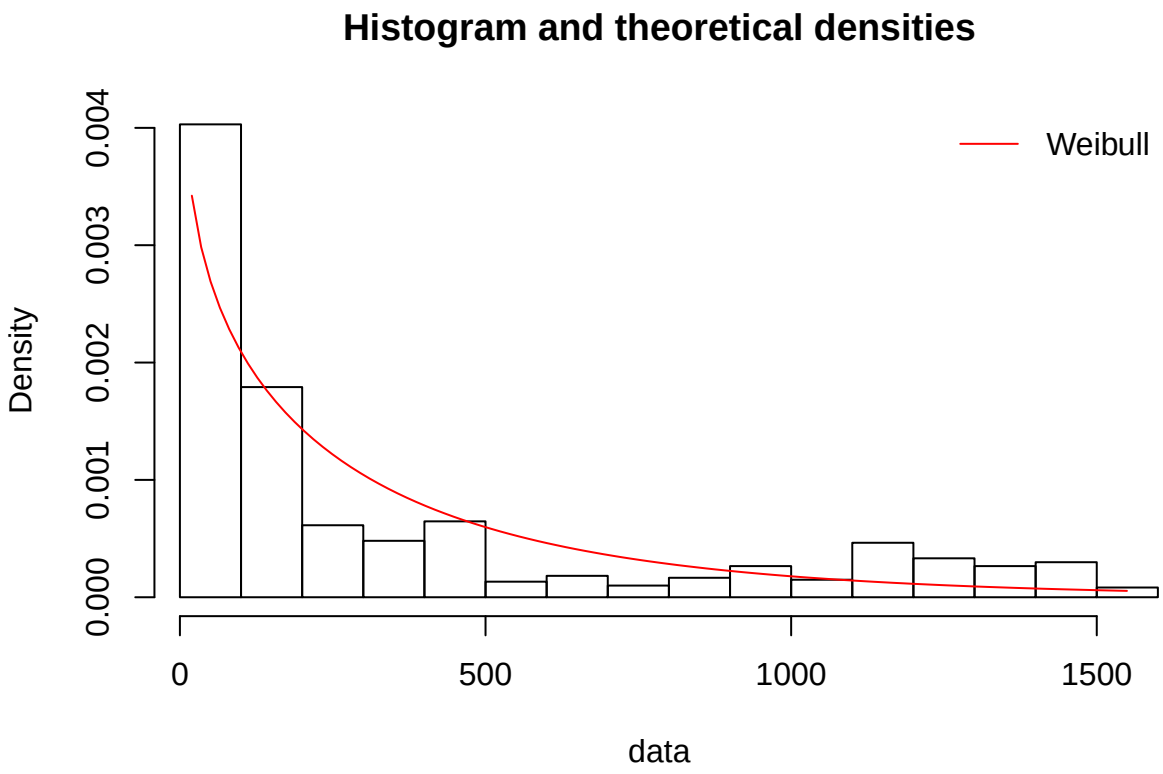
```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##          estimate  Std. Error
## shape   0.8571029  0.02655744
## scale 347.2272695 17.50773049
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##          estimate   Std. Error
## shape 0.833004714 3.321811e-02
## rate  0.002198759 8.073943e-05
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the S&P500 data (non-recession) along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.

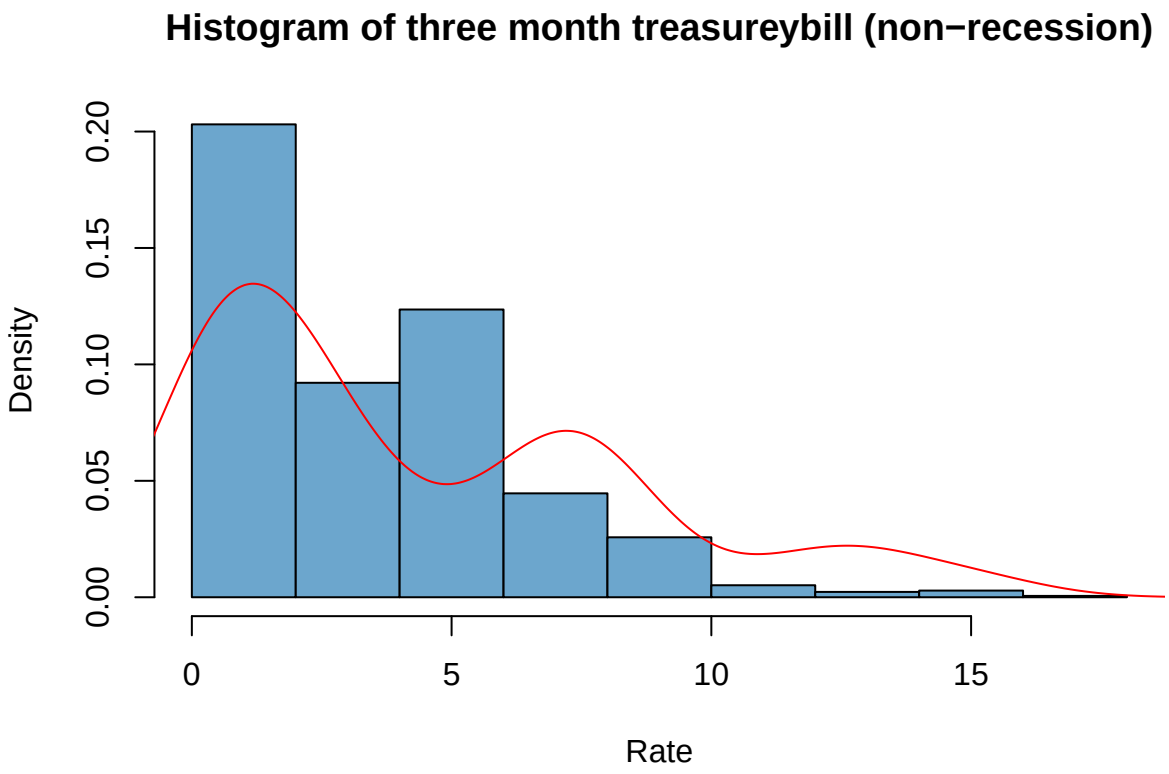


Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Weibull distribution.

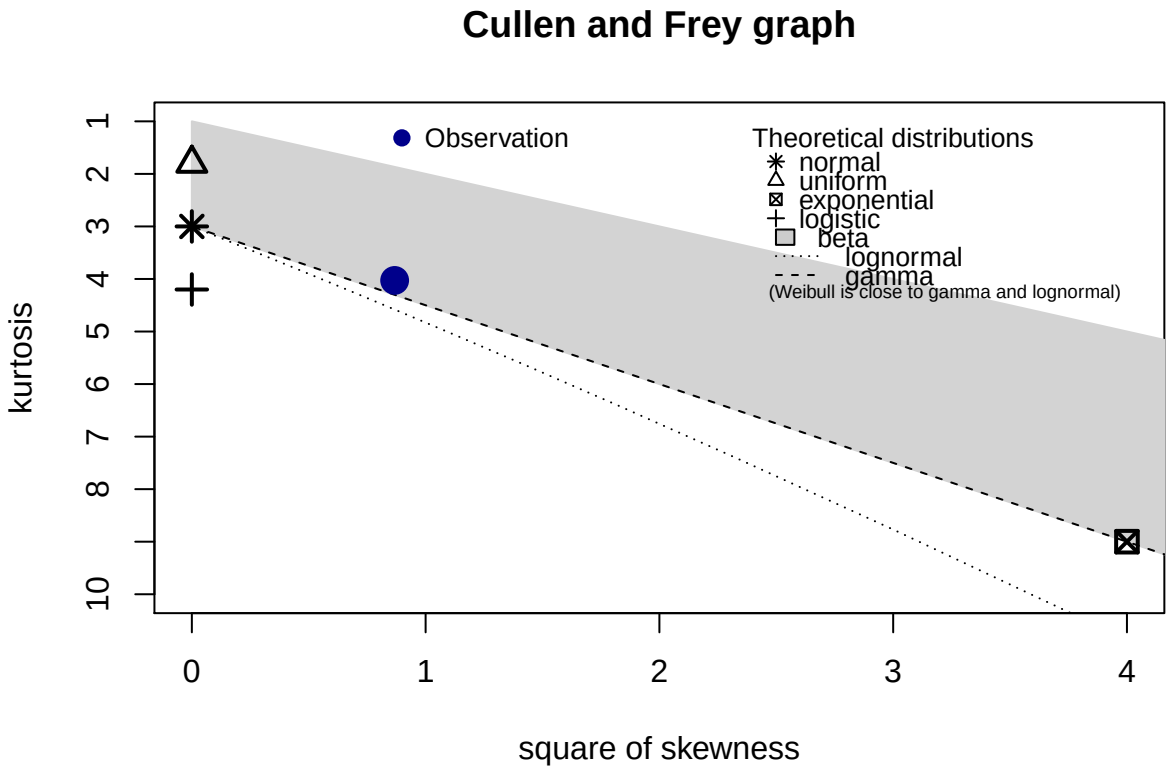
Subsetting Three Month Treasury Bill Data (non-recession)

Using the following code I was able to subset the three month treasury bill data to include points generated only in times when a recession is NOT occurring. The code and the dates used can be seen below.

```
myfunc=function(x,y){Three_treasury[Three_treasury$DATE >= x & Three_treasury$DATE <= y,]}
rec1=myfunc("1934-01-01","1937-05-01")
rec2=myfunc("1938-06-01","1945-02-01")
rec3=myfunc("1945-10-01","1948-11-01")
rec4=myfunc("1950-10-01","1953-07-01")
rec5=myfunc("1954-05-01","1957-08-01")
rec6=myfunc("1958-04-01","1960-04-01")
rec7=myfunc("1961-02-01","1969-12-01")
rec8=myfunc("1970-11-01","1973-11-01")
rec9=myfunc("1975-03-01","1980-01-01")
rec10=myfunc("1980-07-01","1981-07-01")
rec11=myfunc("1982-11-01","1990-07-01")
rec12=myfunc("1991-03-01","2001-03-01")
rec13=myfunc("2001-11-01","2007-12-01")
rec14=myfunc("2009-06-01","2018-08-01")
```



Below is the Cullen-Frey graph we created using the subset (non-recession) three month treasury bill data. This graph shows us which distributions our data closely resembles and therefore which distribution best represents our data.




```
## summary statistics
## ------
## min:  0.01   max:  16.3
## median:  3.06
## mean:  3.383158
## estimated sd:  2.979411
## estimated skewness:  0.9314343
## estimated kurtosis:  4.027583
```

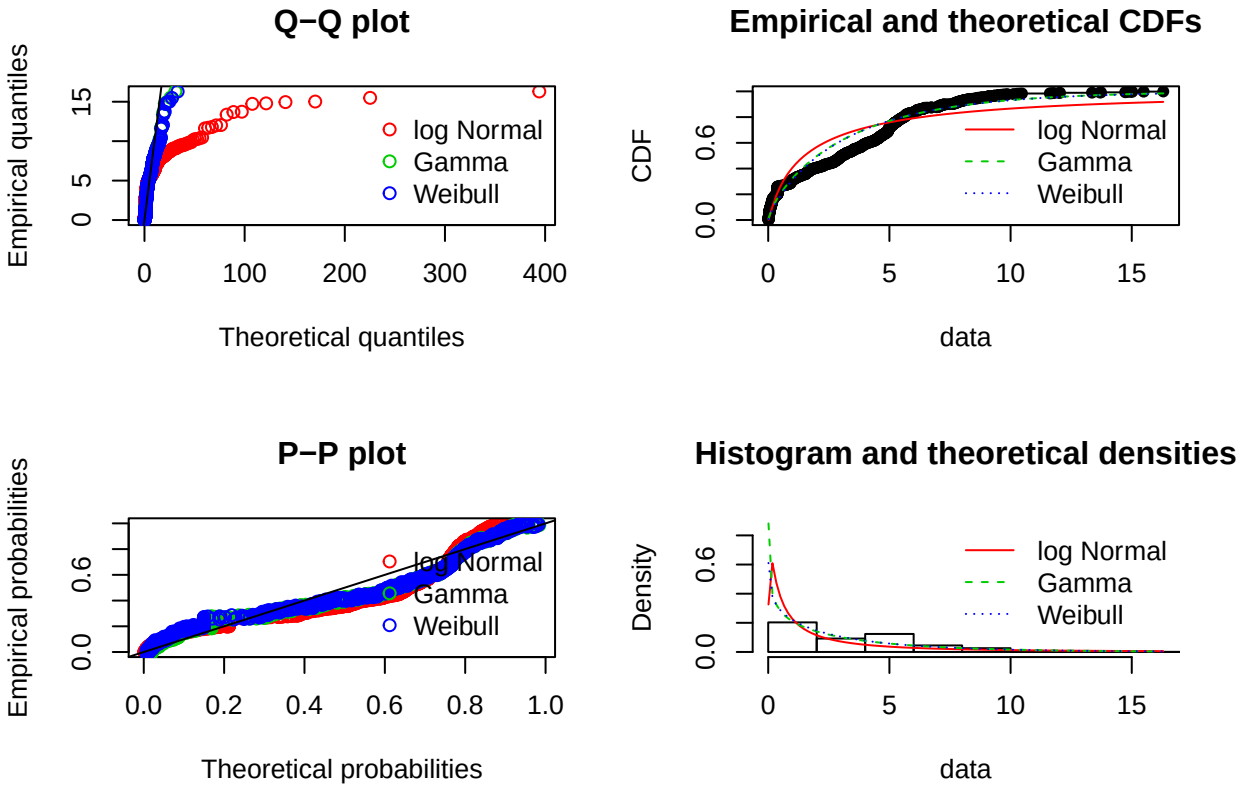
With the Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##          estimate Std. Error
## meanlog 0.4026866 0.05797071
## sdlog   1.7138166 0.04099142
```

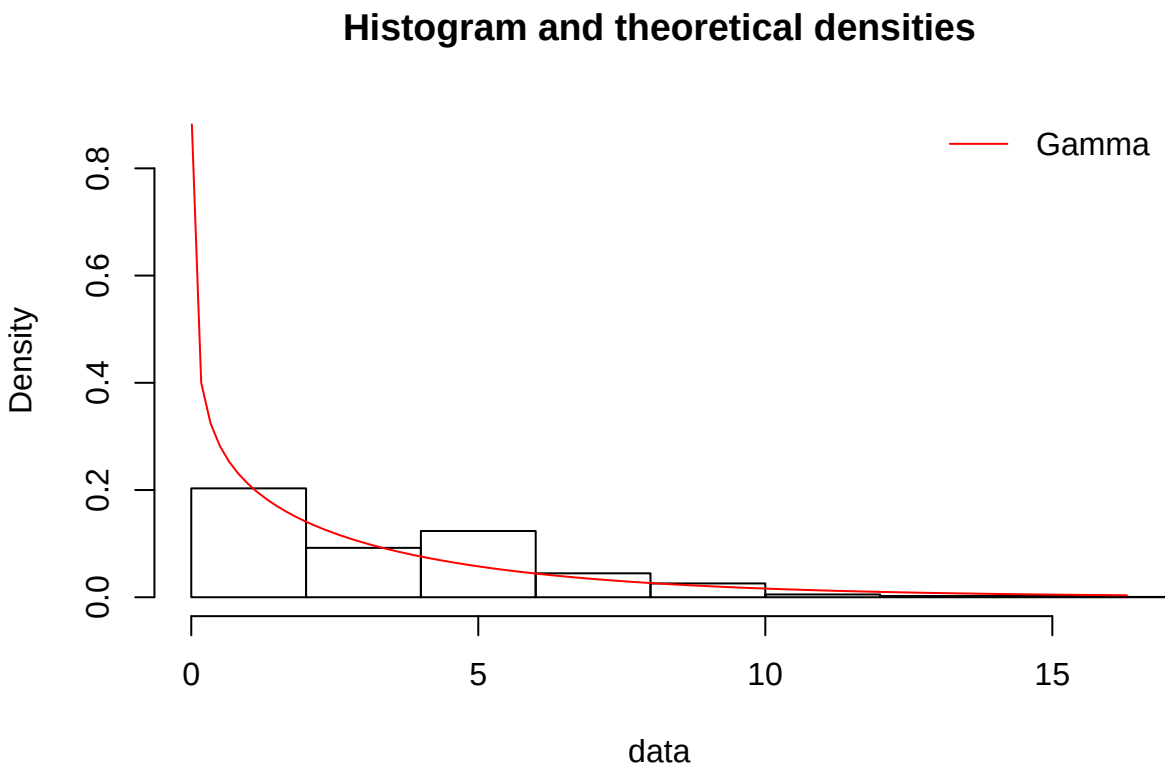
```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape 0.7354315 0.03011525
## rate  0.2173898 0.01235956
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape 0.8571277 0.02471212
## scale 3.1725916 0.13081138
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the three month treasury bill data (non-recession) along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.

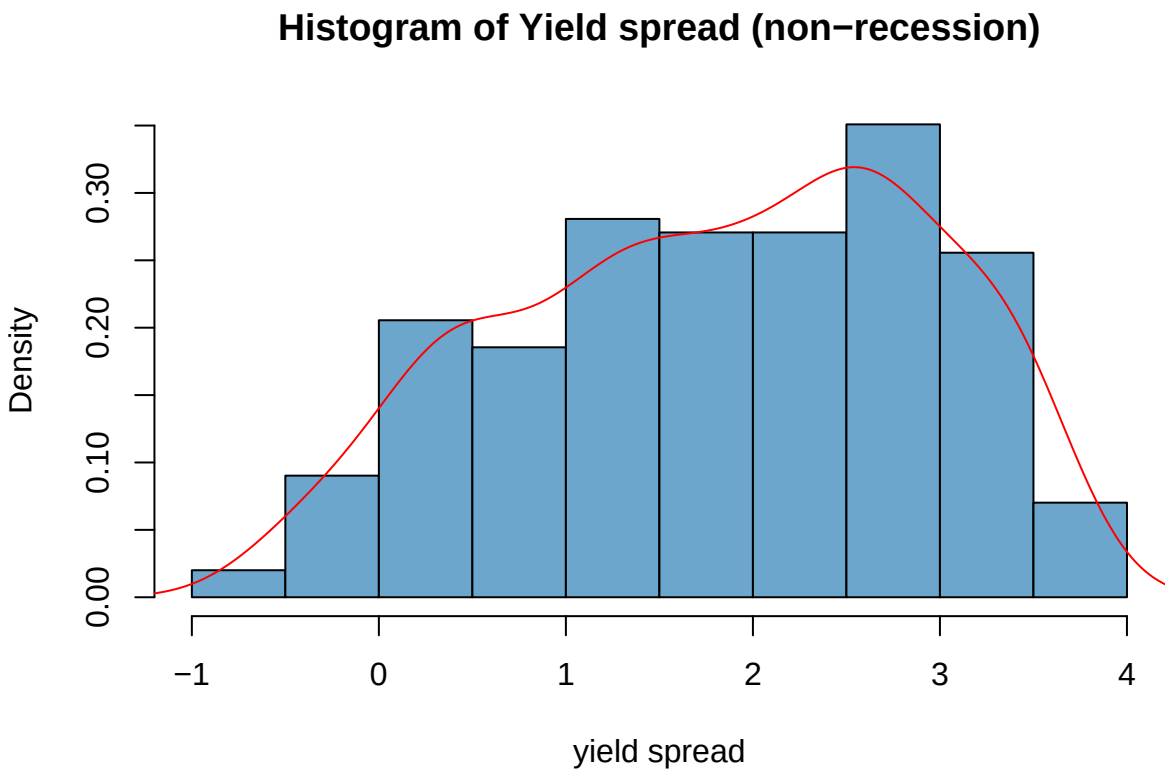


Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Weibull distribution.

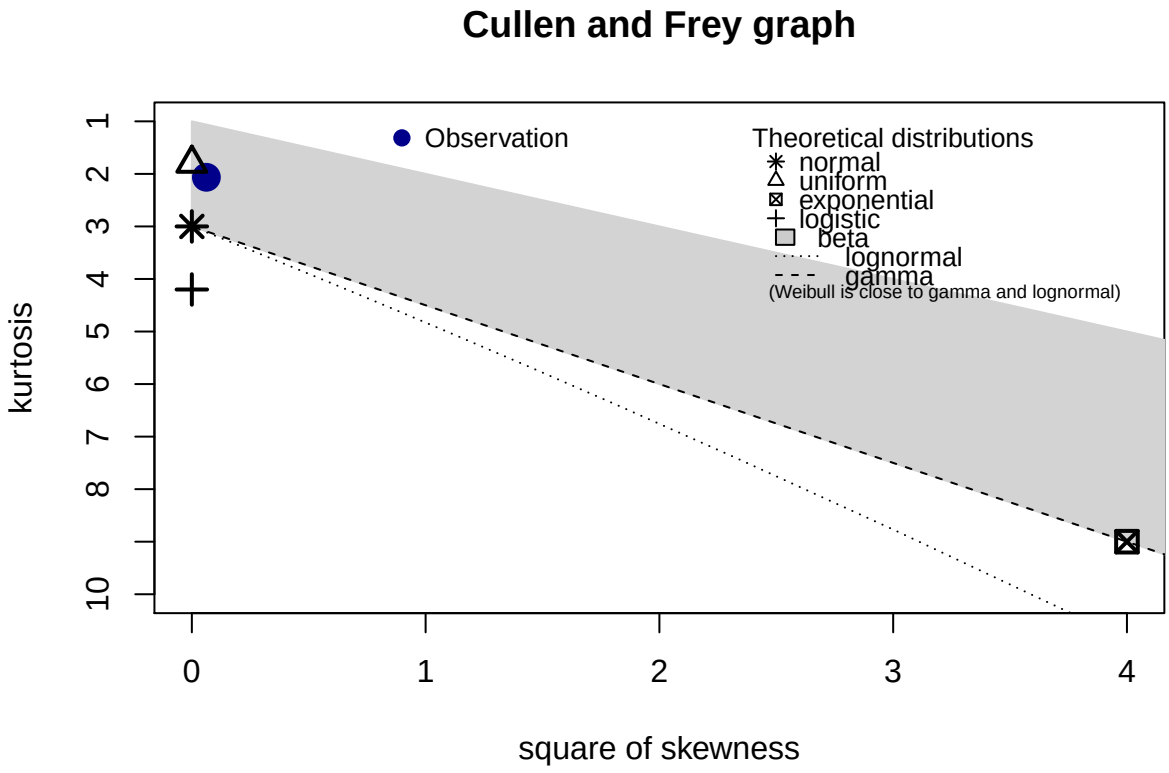
Subsetting Yield spread Data (non-recession)

Using the following code I was able to subset (non-recession) the yield spread data to include points generated only in times when a recession is NOT occurring. The code and the dates used can be seen below.

```
myfunc=function(x,y){Yield_Spread[Yield_Spread$DATE >= x & Yield_Spread$DATE <= y,]}
rec1=myfunc("1961-02-01","1969-12-01")
rec2=myfunc("1970-11-01","1973-11-01")
rec3=myfunc("1975-03-01","1980-01-01")
rec4=myfunc("1982-11-01","1990-07-01")
rec5=myfunc("1991-03-01","2001-03-01")
rec6=myfunc("2001-11-01","2007-12-01")
rec7=myfunc("2009-06-01","2018-10-01")
```



Below is the Cullen-Frey graph we created using the subset (non-recession) yield spread data. This graph shows us which distributions our data closely resembles and therefore which distribution best represents our data.

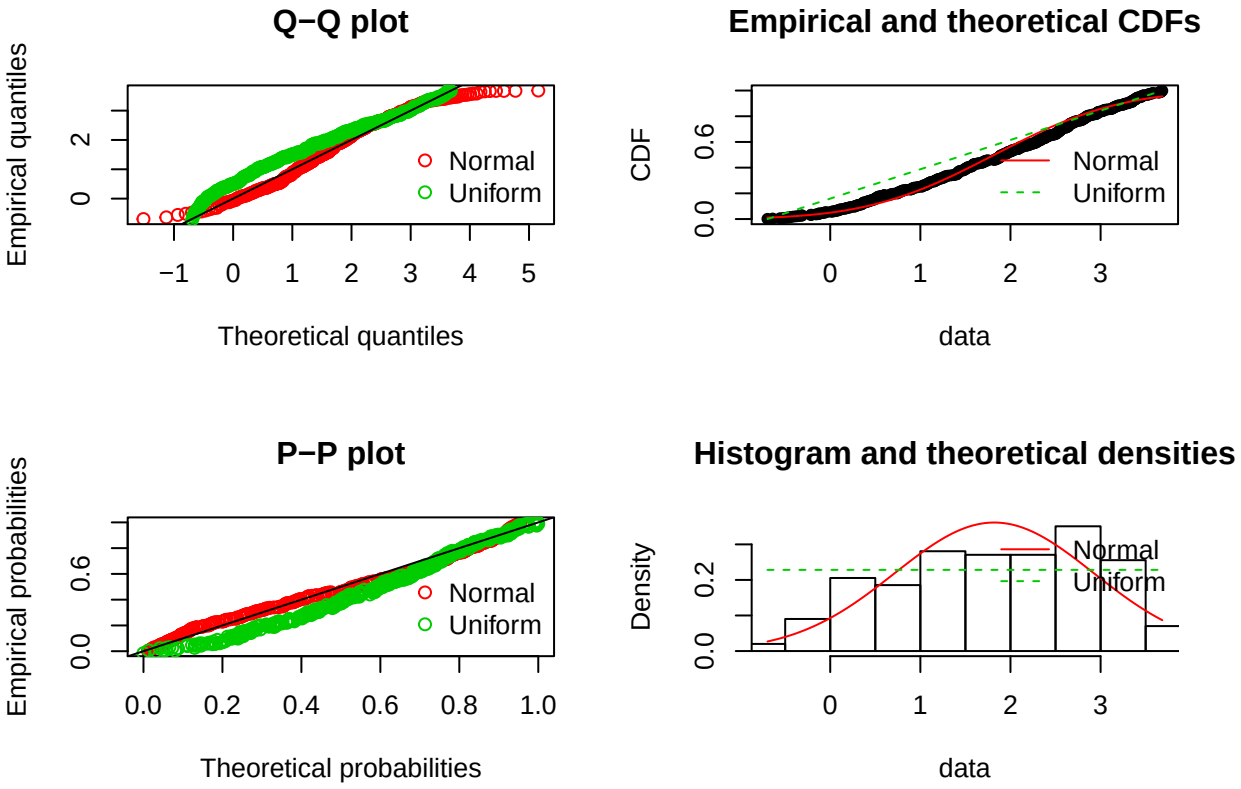



```
## summary statistics
## ------
## min:  -0.6965   max:  3.684545
## median:  1.9135
## mean:  1.82015
## estimated sd:  1.105077
## estimated skewness:  -0.2493874
## estimated kurtosis:  2.06473
```

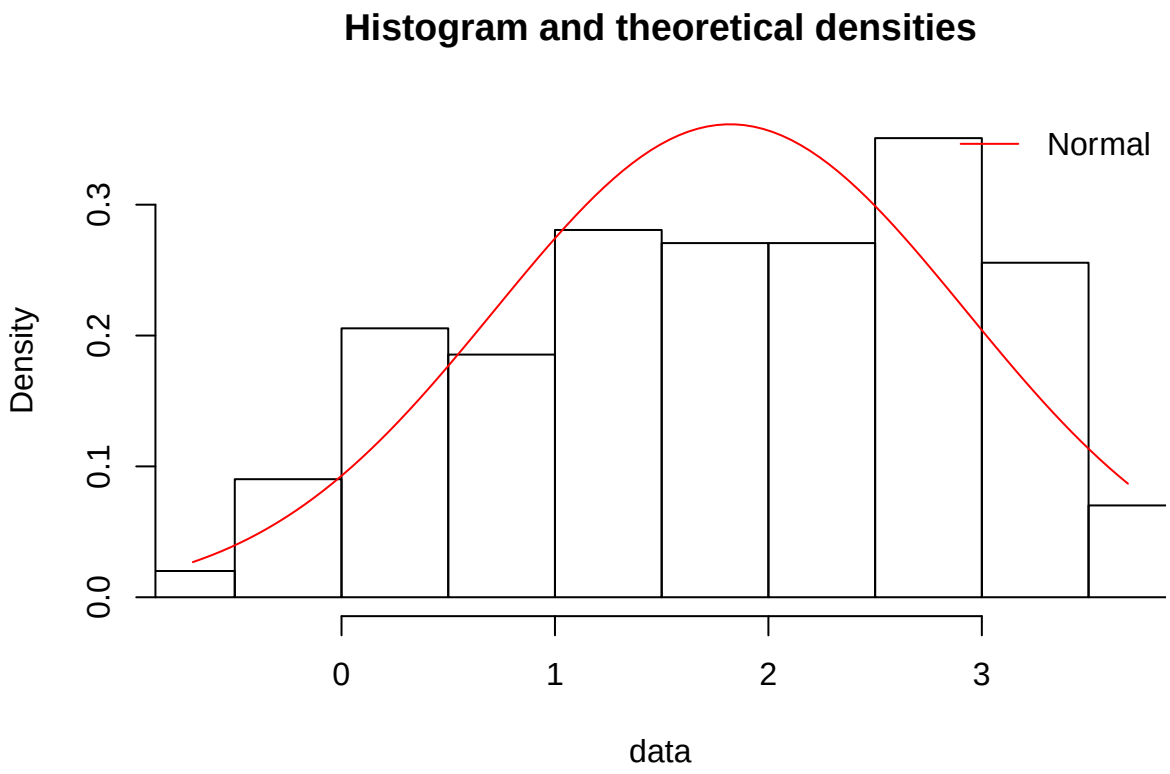
Using the following Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## mean 1.820150 0.05525369
## sd   1.103692 0.03907012

## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## min -0.696500         NA
## max  3.684545         NA
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the yield spread data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.



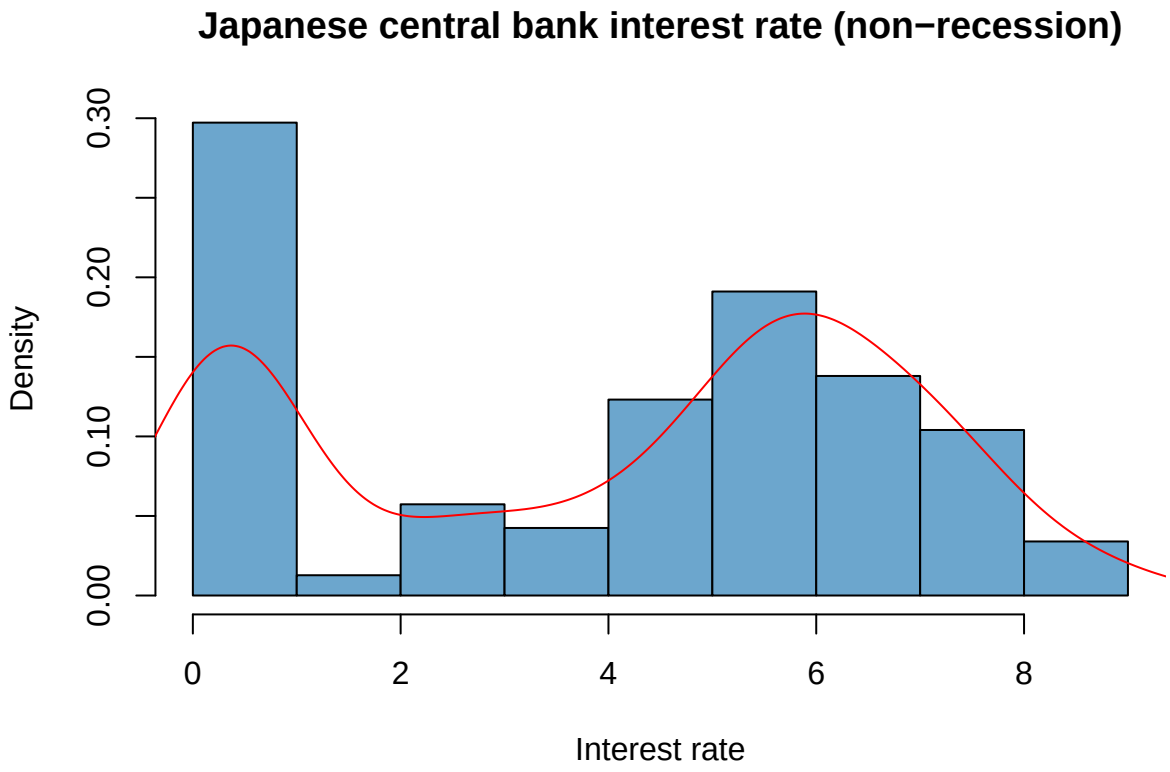
Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Normal distribution.

Subsetting Japanese Central Bank Interest Rate data (non-recession)(based on OECD recession indicators)

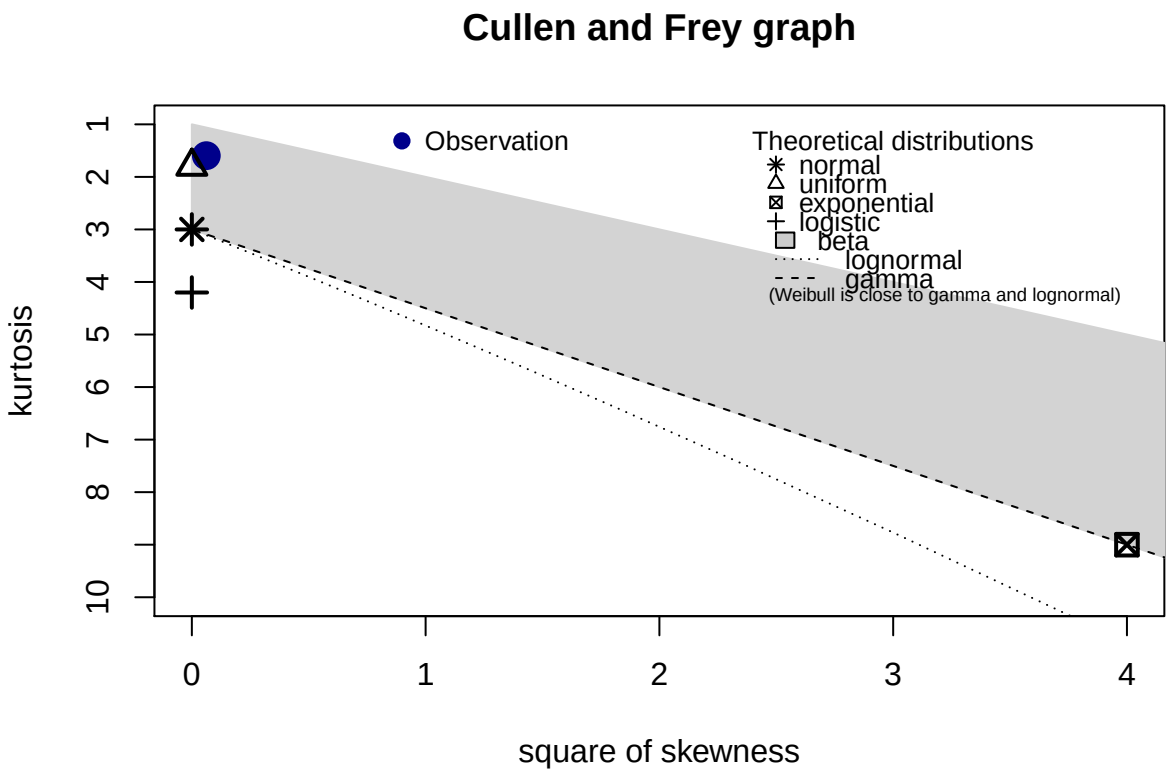
Using the following code we were able to subset the Japanese Central bank interest rate data to include points generated only in times of when is NOT occurring. Using the FRED database we were also able to find the dates when when Japan did not experience a recession (according to OECD recession based indicators). We included these dates in our code, which can be seen below.

```
myfunc=function(x,y){Japanese_central_bank[Japanese_central_bank$DATE >= x & Japanese_central_bank$DATE
rec1=myfunc("1953-01-01","1953-07-01")
rec2=myfunc("1954-05-01","1957-08-01")
rec3=myfunc("1958-04-01","1960-11-01")
rec4=myfunc("1961-02-01","1961-12-01")
rec5=myfunc("1963-02-01","1964-04-01")
rec6=myfunc("1965-11-01","1970-03-01")
rec7=myfunc("1971-10-01","1973-04-01")
rec8=myfunc("1975-02-01","1979-06-01")
rec9=myfunc("1980-05-01","1982-03-01")
rec10=myfunc("1983-05-01","1985-09-01")
rec11=myfunc("1987-02-01","1990-08-01")
rec12=myfunc("1994-10-01","1997-01-01")
rec13=myfunc("1999-05-01","2001-01-01")
```

```
rec14=myfunc("2002-01-01","2004-03-01")
rec15=myfunc("2004-12-01","2008-02-01")
rec16=myfunc("2009-04-01","2010-08-01")
rec17=myfunc("2012-09-01","2013-10-01")
```



Below is the Cullen-Frey graph we created using the subset (non-recession) Japanese central bank interest rate data. This graph shows us which distributions our data closely resembles and therefore which distribution best represents our data.




```
## summary statistics
## ------
## min:  0.1   max:  9
## median:  5
## mean:  4.039469
## estimated sd:  2.752952
## estimated skewness:  -0.248671
## estimated kurtosis:  1.595961
```

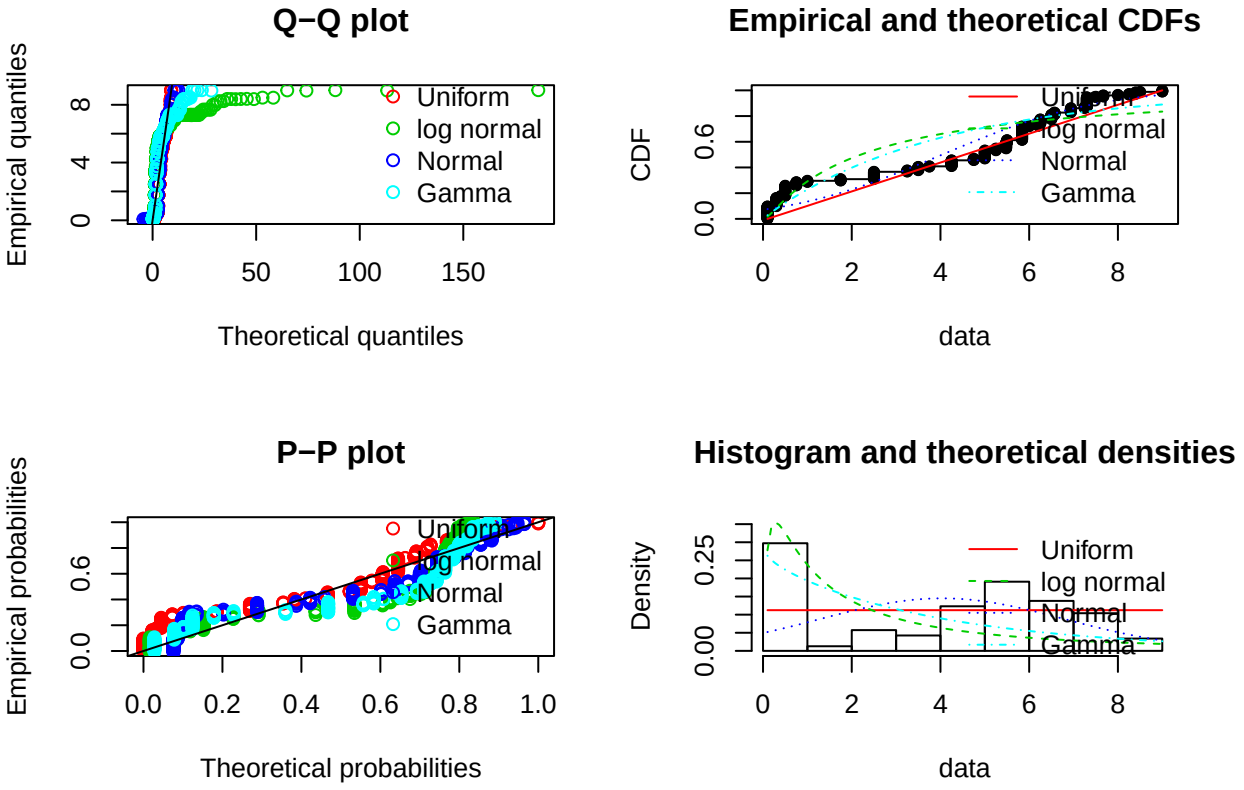
Again, Using the same Fitdist command we are able to find the parameters that best fit each suggested distribution to the data. The Fitdist command uses MLE in order to find the best parameters to fit the data set. The output below states the parameters that would best fit the data for each distribution.

```
## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters:
##     estimate Std. Error
## min      0.1         NA
## max      9.0         NA
```

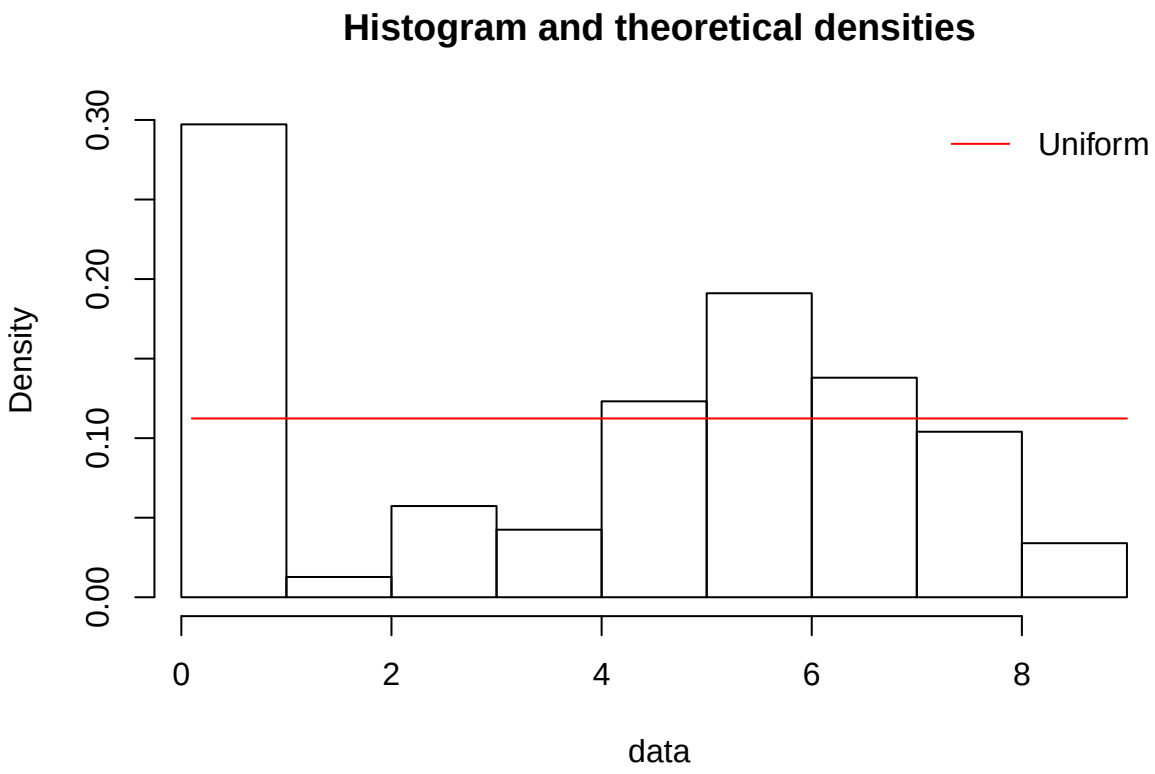
```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##         estimate Std. Error
## meanlog 0.791679 0.06651383
## sdlog   1.443519 0.04703228
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## mean 4.039469 0.12671459
## sd   2.750028 0.08960069

## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape 0.9601072 0.05488686
## rate  0.2376538 0.01759152
```

Below we can see the cdf graph which provides a plot of the empirical distribution and each fitted distribution in cdf. The theoretical distribution graph provides a density plot of each fitted distribution with the histogram of the data. the pp plot graph provides a plot of the probabilities of each fitted distribution (x-axis) against the empirical probabilities. The qq plot graph provides a plot of the quantiles of each theoretical distribution (x-axis) against the empirical quantiles of the data. Lastly the denscomp plot provides a density plot of each fitted distribution with the histogram of the data.



The graph below shows the histogram of the Japanese central bank interest rate data along with the distributions we fitted from the Cullen-Frey graph. We can use this histogram along with the Cullen-Frey graph, and the other plots we created to determine which distribution best fits the data.



Based on the Cullen-Frey graph, the qq plot, the cdf plot, and the pp plot we determined that the best distribution to fit the data would be the Uniform distribution.

Part E) Comparing Subset data

Based on the data that was subset we can conclude that there is some significant differences between their distributions. Based on the Cullen-Frey graphs and other plots we can conclude that some of the recession and non-recession data sets are significantly different in terms of their distributions. For example the S&P500 data and the Three month treasury bill data differed in their distribution when subset. We can also assume that the recession and non-recession data will have different distributions based on the fact that they come from different periods with differing economic conditions. When it comes to the difference between the total data and the subset data we believe that the data is very closely distributed. This assumption is based on the fact that some of the subset data sets share similar distributions to the total data.

Part F) Kolmogorov-Smirnov Tests

Recession vs non-recession

When comparing the recession subset data vs the non-recession subset data we get very small p-values from our Kolmogorov-Smirnov Test. This suggests that our data comes from different different distributions.

```
ks.test(sp500rec, sp500nrec)
```

```
##
```

```
##  Two-sample Kolmogorov-Smirnov test
##
## data:  sp500rec and sp500nrec
## D = 0.29519, p-value = 4.328e-10
## alternative hypothesis: two-sided
```

The very small p-value above suggests that the S&P500 recession data and the S&P500 non-recession data come from different distributions. This confirms the assumption we made base on the data in part E.

```
#(total three month treasury data vs non-recession three month treasury)
ks.test(Three_treasuryrec, Three_treasurynrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Three_treasuryrec and Three_treasurynrec
## D = 0.21069, p-value = 1.575e-05
## alternative hypothesis: two-sided
```

The very small p-value above suggests that the three month treasury bill recession data and the three month treasury bill non-recession data come from different distributions. This confirms our assumption in part E.

```
#(total yield spread data vs non-recession yield spread)
ks.test(yield_spreadrec, yield_spreadnrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  yield_spreadrec and yield_spreadnrec
## D = 0.12892, p-value = 0.4747
## alternative hypothesis: two-sided
```

The large p-value from this Kolmogorov-Smirnov test suggests that the two data sets come from the same distribution. This goes against the assumption we made in Part E. This suggests that recession yield spread data and non-recession yield spread data is significantly different.

```
#(total japanese central bank data vs non-recession japanese central bank)
ks.test(japcenrec, japcennrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  japcenrec and japcennrec
## D = 0.1114, p-value = 0.01797
## alternative hypothesis: two-sided
```

The very small p-value above suggests that the Japanese central bank interest rate recession data and the Japanese central bank interest non-recession data come from different distributions. This confirms our assumption in part E.

Total vs Recession

```r
#(total S&p500 data vs recession S&P500)
ks.test(sp500_close, sp500rec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  sp500_close and sp500rec
## D = 0.30551, p-value = 2.085e-11
## alternative hypothesis: two-sided
```

The very small p-value above suggests that the S&P500 recession data and the S&P500 total data come from different distributions. This actually goes against the assumption we made in part E. three month treasury bill

```r
#(total three month treasury data vs recession three month treasury)
ks.test(Three_treasury$TB3MS, Three_treasuryrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Three_treasury$TB3MS and Three_treasuryrec
## D = 0.18344, p-value = 0.0002228
## alternative hypothesis: two-sided
```

The very small p-value above suggests that the total three month treasury bill data and the three month treasury bill recession data come from different distributions. This actually goes against the assumption we made in part E.This suggests that recession three month treasury bill data is significantly different than the total.

```r
#(total three month treasury data vs recession three month treasury)
ks.test(Yield_Spread$T10Y3M, yield_spreadrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Yield_Spread$T10Y3M and yield_spreadrec
## D = 0.11402, p-value = 0.6271
## alternative hypothesis: two-sided
```

The large p-value from this Kolmogorov-Smirnov test suggests that the two data sets come from the same distribution. This confirms the assumption we made based on the data in part E. The assumption being that the total data and the subset data will have similar distributions since the subset data is included in the total data.

```r
#(total japanese central bank data vs recession japanese central bank)
ks.test(Japanese_central_bank$INTDSRJPM193N, japcenrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Japanese_central_bank$INTDSRJPM193N and japcenrec
## D = 0.053508, p-value = 0.5392
## alternative hypothesis: two-sided
```

The large p-value from this Kolmogorov-Smirnov test suggests that the two data sets (total Japanese interest data vs subset (recession) Japanese interest data) come from the same distribution. This confirms the assumption we made based on the data in part E. The assumption being that the total data and the subset data will have similar distributions since the subset data is included in the total data.

Total vs non-recession

```
#(total S&P500 data vs non-recession S&P500)
ks.test(sp500_close, sp500nrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  sp500_close and sp500nrec
## D = 0.11255, p-value = 0.000291
## alternative hypothesis: two-sided
```

The very small p-value above suggests that the S&P500 non-recession data and the total S&P500 data come from different distributions. This actually goes against the assumption we made in part E. This suggests that non-recession s&P500 data is significantly different than the total.

```
#(total three month treasury data vs non-recession three month treasury)
ks.test(Three_treasury$TB3MS, Three_treasurynrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Three_treasury$TB3MS and Three_treasurynrec
## D = 0.027248, p-value = 0.8763
## alternative hypothesis: two-sided
```

The large p-value from this Kolmogorov-Smirnov test above suggests that the two data sets come from the same distribution. This confirms the assumption we made based on the data in part E. The assumption being that the total data and the subset data will have similar distributions since the subset data is included in the total data.

```
#(total yield spread data vs non-recession yield spread)
ks.test(Yield_Spread$T10Y3M, yield_spreadnrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Yield_Spread$T10Y3M and yield_spreadnrec
## D = 0.014901, p-value = 1
## alternative hypothesis: two-sided
```

The large p-value from this Kolmogorov-Smirnov test above suggests that the two data sets (total yield spread data vs subset (non-recession) yield spread data) come from the same distribution. This confirms the assumption we made based on the data in part E. The assumption being that the total data and the subset data will have similar distributions since the subset data is included in the total data.

```r
#(total japanese central bank data vs non-recession japanese central bank)
ks.test(Japanese_central_bank$INTDSRJPM193N, japcennrec)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Japanese_central_bank$INTDSRJPM193N and japcennrec
## D = 0.057897, p-value = 0.2806
## alternative hypothesis: two-sided
```

The large p-value from this Kolmogorov-Smirnov test suggests that the two data sets (total Japanese interest data vs subset (non-recession) Japanese interest data) come from the same distribution. This confirms the assumption we made based on the data in part E. The assumption being that the total data and the subset data will have similar distributions since the subset data is included in the total data.

Even though some of the Kolmogorov-Smirnov tests went against the assumptions we made in part E. The majority of them confirmed the assumption we made based on the data. Those assumptions were that the recession and non-recession data had different distributions since they came from different period's that had different economic conditions. The other assumption was that the total data and the subset data would be similar since the subset data comes from and is included in the total data. These assumptions were also made using the data we received from the Cullen-Frey graphs.

Question 3

Part A) Estimating CAPM model

```r
library("readxl")
data=read_excel("capm4.xlsx")
markport=data$mkt-data$riskfree #calculating market premium
riskprem=data-data$riskfree #calculating risk premium on each stock

regmobile=lm(riskprem$xom~markport)
regmicro=lm(riskprem$msft~markport)
regge=lm(riskprem$ge~markport)
reggm=lm(riskprem$gm~markport)
regibm=lm(riskprem$ibm~markport)
regdis=lm(riskprem$dis~markport)
```

First we calculated the risk premium on the market portfolio and the risk premium for each stock. Then we ran a regression with these values in order to calculate each stocks CAPM model. The output below shows the intercept and β_1 values of our regression with each stock. Using the given β values for each regression we can determine which stocks are defensive and which are aggressive.

```r
summary(regmobile) #Exxon-Mobile
```

```
##
## Call:
## lm(formula = riskprem$xom ~ markport)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.127422 -0.032706 -0.002982  0.027316  0.216216
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.007966   0.005118  -1.556    0.122
## markport     0.461258   0.088607   5.206 7.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0505 on 130 degrees of freedom
## Multiple R-squared:  0.1725, Adjusted R-squared:  0.1661
## F-statistic:  27.1 on 1 and 130 DF,  p-value: 7.349e-07
```

Exxon-Mobiles β value is the smallest value generated from all the regressions. This means that this stock is the most defensive stock in our portfolio. This means that the Exxon-Mobile stock is not sensitive to variation in the entire stock market.

```
summary(regmicro) #Microsoft
```

```
##
## Call:
## lm(formula = riskprem$msft ~ markport)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26864 -0.05569 -0.00845  0.04261  0.35678
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.013737   0.009061   1.516    0.132
## markport    1.259919   0.156861   8.032 5.03e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0894 on 130 degrees of freedom
## Multiple R-squared:  0.3317, Adjusted R-squared:  0.3265
## F-statistic: 64.51 on 1 and 130 DF,  p-value: 5.034e-13
```

Microsofts β value is 1.26 which is the highest beta value we have. This means that the Microsoft stock is the most aggressive stock in our portfolio. This indicates that Microsoft stock in very sensitive to variation in the stock market.

```
summary(regge) #GE
```

```
##
## Call:
## lm(formula = riskprem$ge ~ markport)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
```

```
## -0.156837 -0.036767 -0.004774  0.034106  0.181055
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.005324   0.005518  -0.965    0.336
## markport     0.858974   0.095525   8.992 2.48e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05444 on 130 degrees of freedom
## Multiple R-squared:  0.3835, Adjusted R-squared:  0.3787
## F-statistic: 80.86 on 1 and 130 DF,  p-value: 2.477e-15
```

GE's β value is .859 which indicates that the stock is relatively defensive since the value is less than 1. This means that GE's stock is the second most defensive stock in our portfolio. Therefore the stock does not fluctuate very much with variation in the market.

```
summary(reggm) #GM
```

```
##
## Call:
## lm(formula = riskprem$gm ~ markport)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40666 -0.06120 -0.00273  0.06278  0.29125
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.007248   0.011393  -0.636    0.526
## markport     1.146838   0.197242   5.814 4.46e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1124 on 130 degrees of freedom
## Multiple R-squared:  0.2064, Adjusted R-squared:  0.2003
## F-statistic: 33.81 on 1 and 130 DF,  p-value: 4.464e-08
```

GM's β value is 1.14 which indicates that the stock is relatively aggressive since its value is greater than 1. As a result GM's stock is relatively sensitive to variation in the overall stock market.

```
summary(regibm) #IBM
```

```
##
## Call:
## lm(formula = riskprem$ibm ~ markport)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.262998 -0.039921 -0.002788  0.038935  0.269202
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.010207   0.007114   1.435    0.154
## markport    1.148245   0.123152   9.324 3.83e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07019 on 130 degrees of freedom
## Multiple R-squared:  0.4007, Adjusted R-squared:  0.3961
## F-statistic: 86.93 on 1 and 130 DF,  p-value: 3.829e-16
```

IBM's β value from our regression comes out to be 1.14 which suggests that the stock is aggressive. This also means that the stock is sensitive to fluctuations in the variance of the overall market. This makes IBM the second most aggressive stock in our portfolio

```
summary(regdis) #Disney
```

```
##
## Call:
## lm(formula = riskprem$dis ~ markport)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.182443 -0.028738 -0.007054  0.027853  0.276871
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.00366    0.00694  -0.527    0.599
## markport     0.91460    0.12015   7.612 4.87e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06848 on 130 degrees of freedom
## Multiple R-squared:  0.3083, Adjusted R-squared:  0.303
## F-statistic: 57.94 on 1 and 130 DF,  p-value: 4.866e-12
```

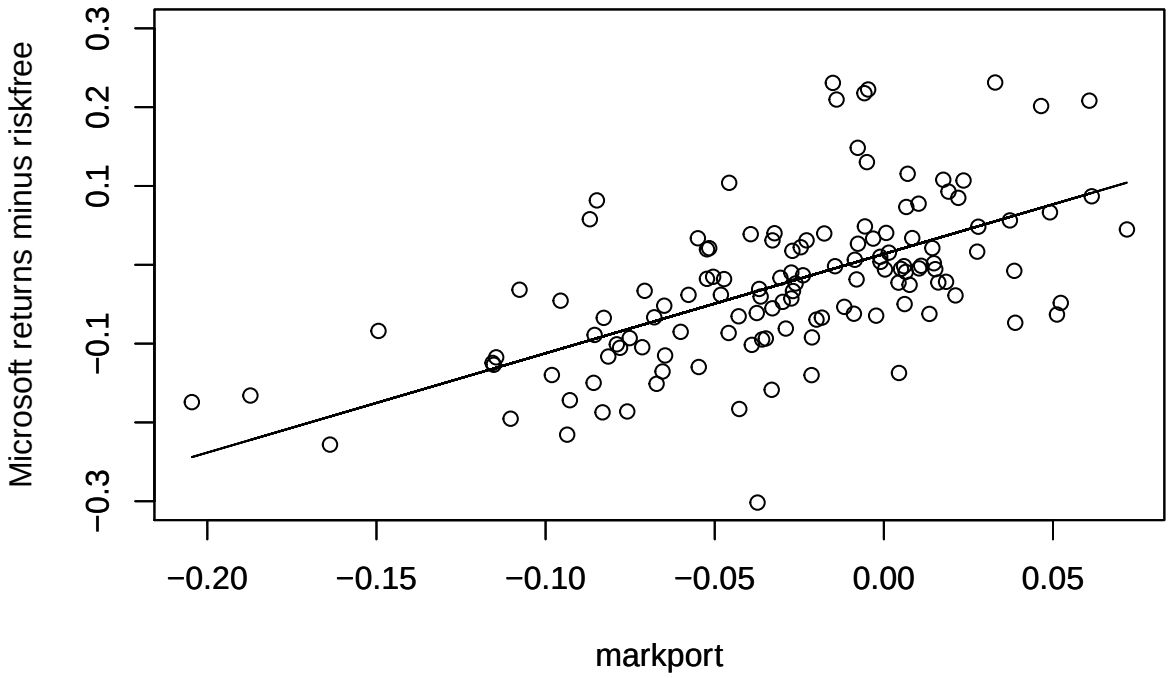
Lastly, Disneys β value comes out to .914. This indicates that Disney is a defensive stock and so therefore is not relatively sensitive to variation in the overall stock market.

According to our regression results Microsoft is the most aggressive and Exxon-Mobile is the most defensive.

Part B) Plotting Microsoft stock regression line with scatter data

All of our estimated α values are very small and clustered around zero. The α value farthest from zero comes out to be .013, which is still relatively close to zero. This would seem to affirm the finance theory assumption that the alpha values should be zero given our estimates. Below you will find the code used to generate the scatter plot of the Microsoft stock along with its fitted regression line.

```
#plot of Mircosoft stock and fitted line
plot(markport,riskprem$msft, ylim=c(-.3,.3), ylab="Microsoft returns minus riskfree")
par(new=T)
plot(markport, regmicro$fitted.values, type="l", ylim=c(-.3,.3), ylab="")
```



Part C) Estimating β_1 given that α=0

Below you will find the code used to run a regression with the assumption that α is zero. The beta estimates and confidence intervals for each stock can been in the summary of each regression.

```
#regression with the assumption alpha is zero
regmobilezero=lm(riskprem$xom~markport-1)
regmicrozero=lm(riskprem$msft~markport-1)
reggezero=lm(riskprem$ge~markport-1)
reggmzero=lm(riskprem$gm~markport-1)
regibmzero=lm(riskprem$ibm~markport-1)
regdiszero=lm(riskprem$dis~markport-1)

#Beta estimates
summary(regmobilezero) #Exxon-Mobile
```

```
##
## Call:
## lm(formula = riskprem$xom ~ markport - 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.133878 -0.040743 -0.006133  0.019296  0.208411
##
## Coefficients:
```

```
##          Estimate Std. Error t value Pr(>|t|)
## markport  0.53191    0.07651   6.952 1.53e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05077 on 131 degrees of freedom
## Multiple R-squared:  0.2695, Adjusted R-squared:  0.2639
## F-statistic: 48.33 on 1 and 131 DF,  p-value: 1.531e-10
```

```
summary(regmicrozero) #Microsoft
```

```
##
## Call:
## lm(formula = riskprem$msft ~ markport - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26857 -0.04153  0.00489  0.05142  0.36945
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## markport   1.1381     0.1354   8.407 6.18e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08984 on 131 degrees of freedom
## Multiple R-squared:  0.3504, Adjusted R-squared:  0.3455
## F-statistic: 70.67 on 1 and 131 DF,  p-value: 6.184e-14
```

```
summary(reggezero) #GE
```

```
##
## Call:
## lm(formula = riskprem$ge ~ markport - 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.163981 -0.042103 -0.008105  0.029787  0.180365
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## markport  0.90619    0.08202   11.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05443 on 131 degrees of freedom
## Multiple R-squared:  0.4824, Adjusted R-squared:  0.4784
## F-statistic: 122.1 on 1 and 131 DF,  p-value: < 2.2e-16
```

```
summary(reggmzero) #GM
```

```
##
```

```
## Call:
## lm(formula = riskprem$gm ~ markport - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41527 -0.06368 -0.00793  0.05847  0.28786
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## markport    1.211      0.169   7.166 5.03e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1122 on 131 degrees of freedom
## Multiple R-squared:  0.2816, Adjusted R-squared:  0.2761
## F-statistic: 51.35 on 1 and 131 DF,  p-value: 5.026e-11
```

```
summary(regibmzero) #IBM
```

```
##
## Call:
## lm(formula = riskprem$ibm ~ markport - 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.251126 -0.030083  0.003168  0.046038  0.278618
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## markport   1.0577     0.1062   9.961   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07047 on 131 degrees of freedom
## Multiple R-squared:  0.431,  Adjusted R-squared:  0.4266
## F-statistic: 99.21 on 1 and 131 DF,  p-value: < 2.2e-16
```

```
summary(regdiszero) #Disney
```

```
##
## Call:
## lm(formula = riskprem$dis ~ markport - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18236 -0.03291 -0.01024  0.02550  0.27625
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## markport   0.9471     0.1029   9.204 7.15e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.06829 on 131 degrees of freedom
## Multiple R-squared:  0.3927, Adjusted R-squared:  0.3881
## F-statistic:  84.7 on 1 and 131 DF,  p-value: 7.145e-16
```

```
#Constructing confidence intervals
CIregmobilezero=confint(regmobilezero, level=.95) #Exxon-Mobile
CIregmicrozero=confint(regmicrozero, leve=.95) #Microsoft
CIreggezero=confint(reggezero, level=.95) #GE
CIreggmzero=confint(reggmzero, level=.95) #GM
CIregibmzero=confint(regibmzero, level=.95) #IBM
CIregdiszero=confint(regdiszero, level=.95) #Disney
```

```
#Confidence intervals
CIregmobilezero #Exxon-Mobile
```

```
##              2.5 %   97.5 %
## markport 0.3805591 0.683262
```

```
CIregmicrozero #Microsoft
```

```
##              2.5 %   97.5 %
## markport 0.8702739 1.405898
```

```
CIreggezero #GE
```

```
##              2.5 %   97.5 %
## markport 0.7439442 1.068436
```

```
CIreggmzero  #GM
```

```
##              2.5 %   97.5 %
## markport 0.8767821 1.545456
```

```
CIregibmzero #IBM
```

```
##              2.5 %   97.5 %
## markport 0.8476453 1.267785
```

```
CIregdiszero #Disney
```

```
##              2.5 %  97.5 %
## markport 0.7434917 1.15062
```

After running the regressions with the assumption that α is zero we found that β values of each regression did not change very much. At most the β values differs by .13 between the two regressions. This would suggest that the econometric model and the financial model do not differ by much in terms of the beta value given.

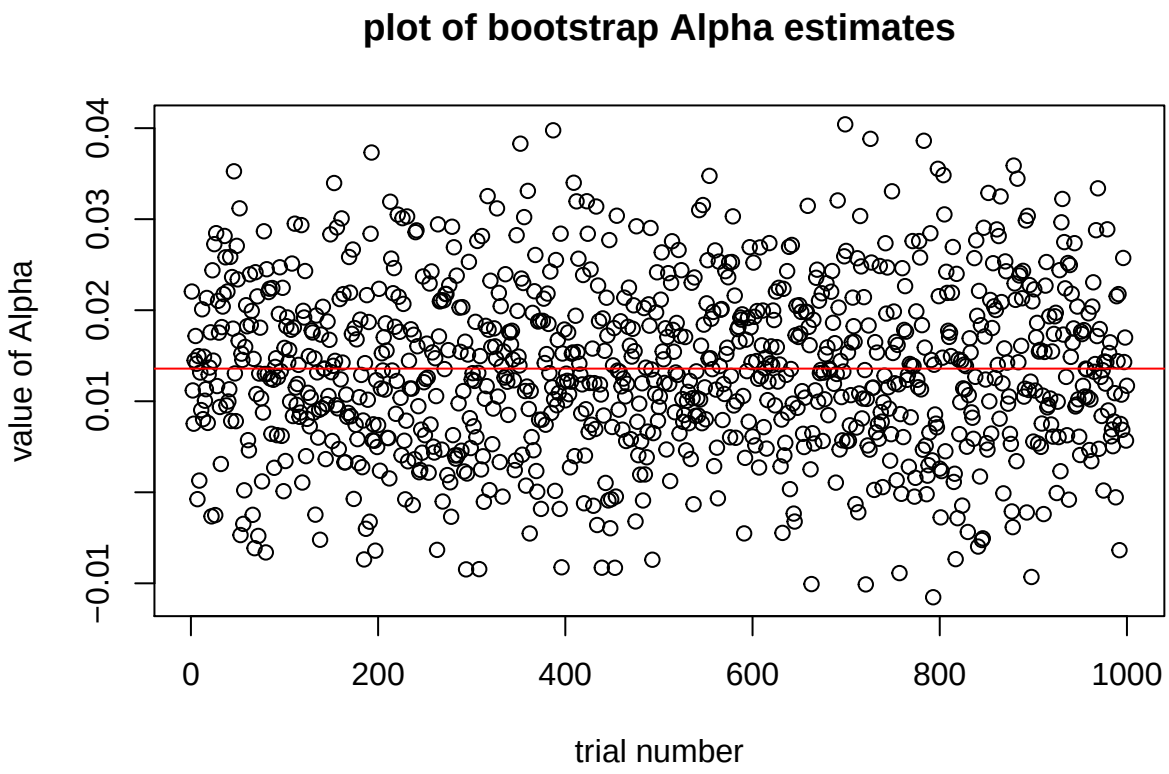
Part D) Bootstrapping Microsoft

Below you will find the code used to generate the boot strap samples from the Microsoft stock. As well as the code used to create a plot of the estimated model parameters across each realization.

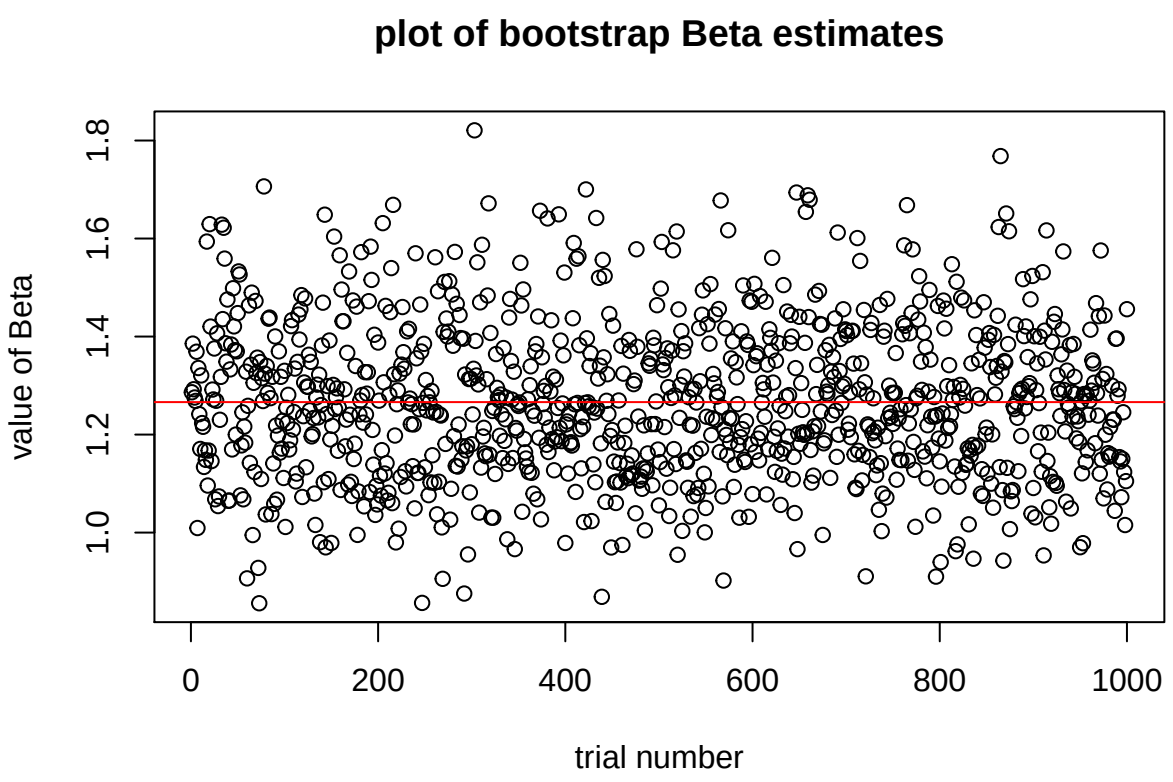
```
alpha=NULL
beta=NULL
dalpha=NULL
dbeta=NULL
x=length(data$msft)
#bootstrap process
for(i in 1:1000){bootstrap=riskprem[c(sample(c(1:length(data$date)), size=length(data$date),replace=T))
regfinal=lm(bootstrap$msft~bootstrap$mkt);
#getting our alpha and beta estimates
alpha[i]=regfinal$coefficients[1];
beta[i]=regfinal$coefficients[2];
#constructing interval lengths
confidint=confint(regfinal, level=.95);
dalpha[i]=confidint[1,2]-confidint[1,1];
dbeta[i]=confidint[2,2]-confidint[2,1]
}
```

Below you will find the plots of the alpha values and beta values generated from each realization of the bootstrap process. The red line represents the overall mean for the bootstrap process. when it came to calculating the confidence interval length for each trial we simply took the upper bound of each realization and subtracted the lower bound. We then plotted these length values.

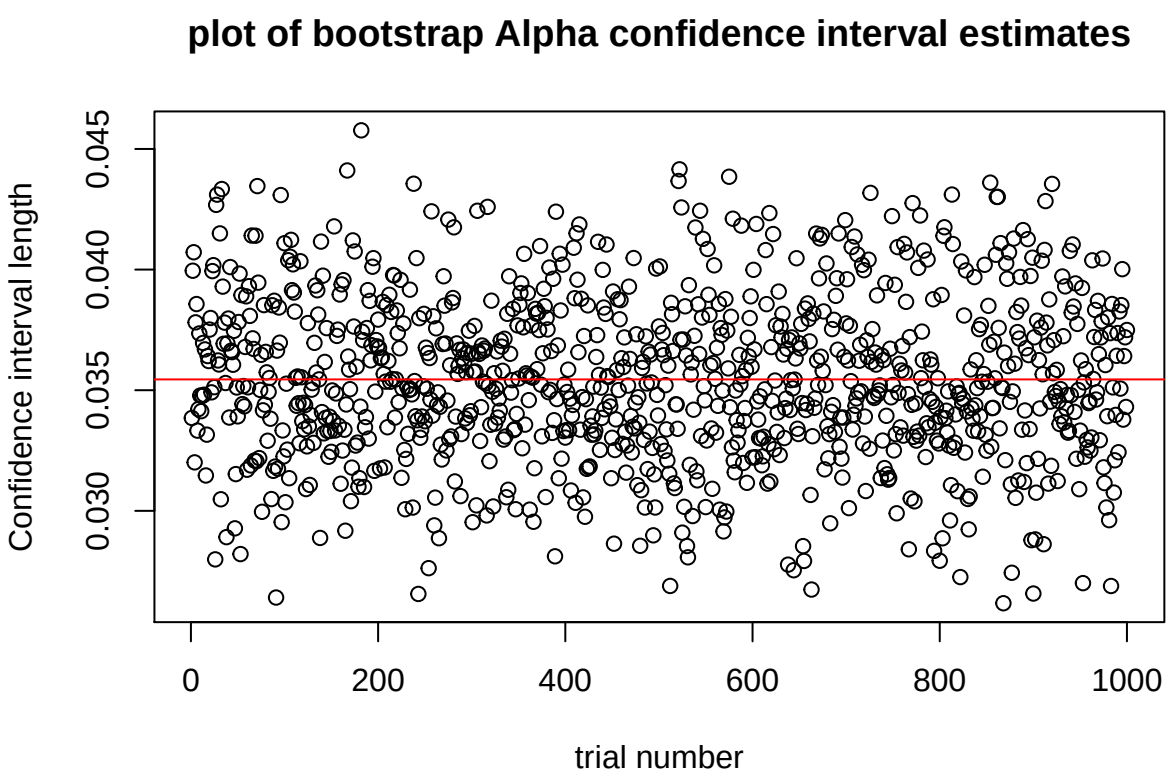
```
plot(alpha, main="plot of bootstrap Alpha estimates", ylab="value of Alpha", xlab="trial number")
abline(h=mean(alpha), col="red")
```



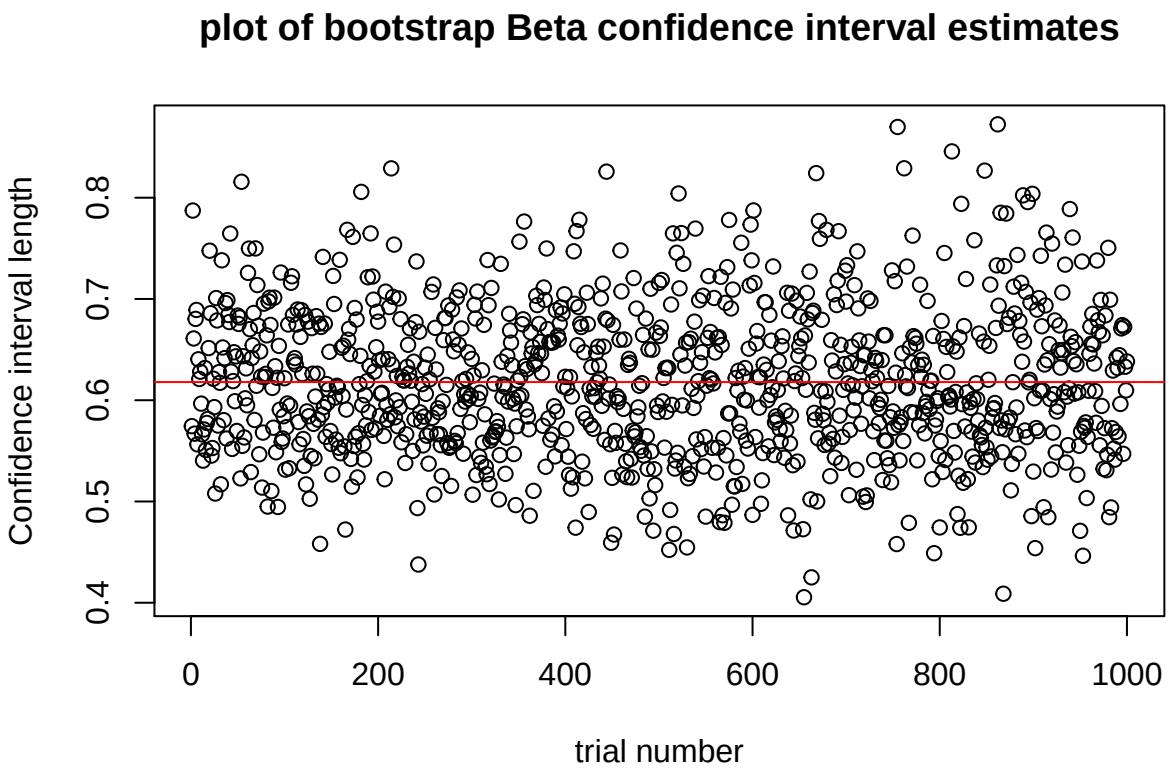

```
plot(beta, main="plot of bootstrap Beta estimates", ylab="value of Beta", xlab="trial number")
abline(h=mean(beta), col="red")
```




```
plot(dalpha, main="plot of bootstrap Alpha confidence interval estimates", ylab="Confidence interval le
abline(h=mean(dalpha), col="red")
```




```
plot(dbeta, main="plot of bootstrap Beta confidence interval estimates", ylab="Confidence interval leng
abline(h=mean(dbeta), col="red")
```



Part E) Mean and volatility of estimates and their respective histograms

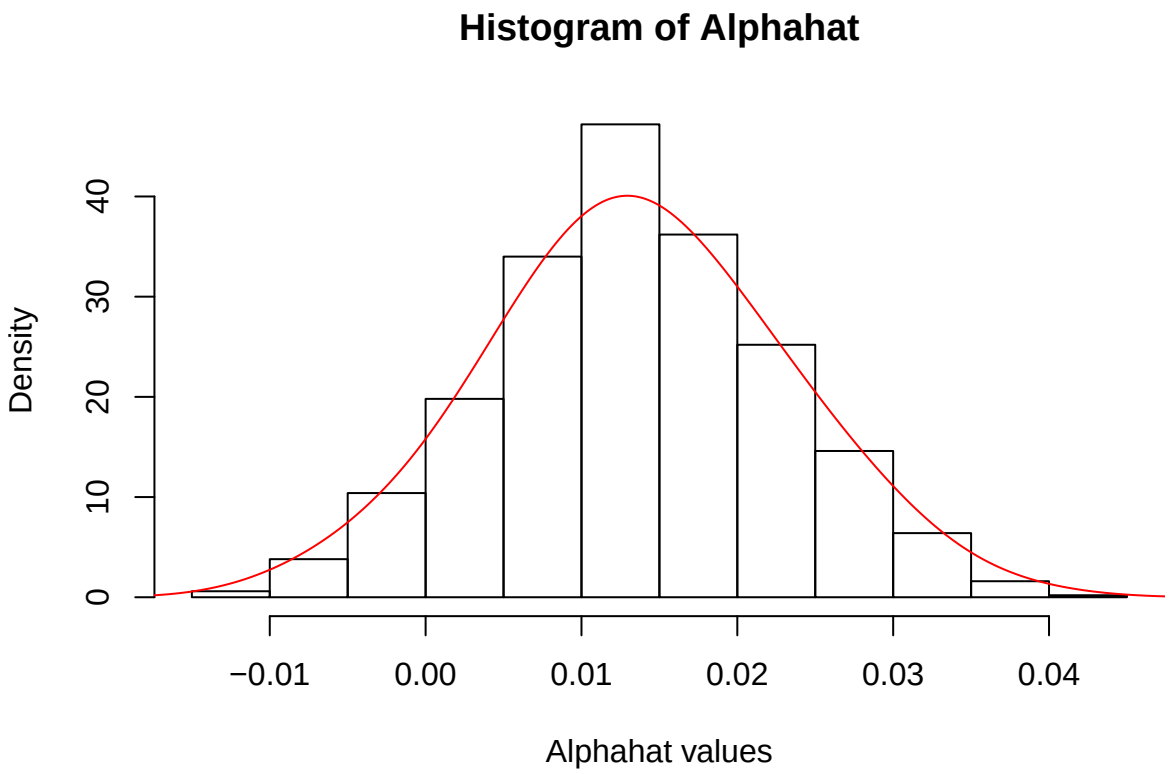
Below you'll find the mean and the volatility (standard deviation) of the $\hat{\alpha}$ estimates respectively. you will also find the the histogram of the estimate along with its respective density curve.

```
mean(alpha)
```

```
## [1] 0.01358472
```

```
sd(alpha)
```

```
## [1] 0.009191939
```



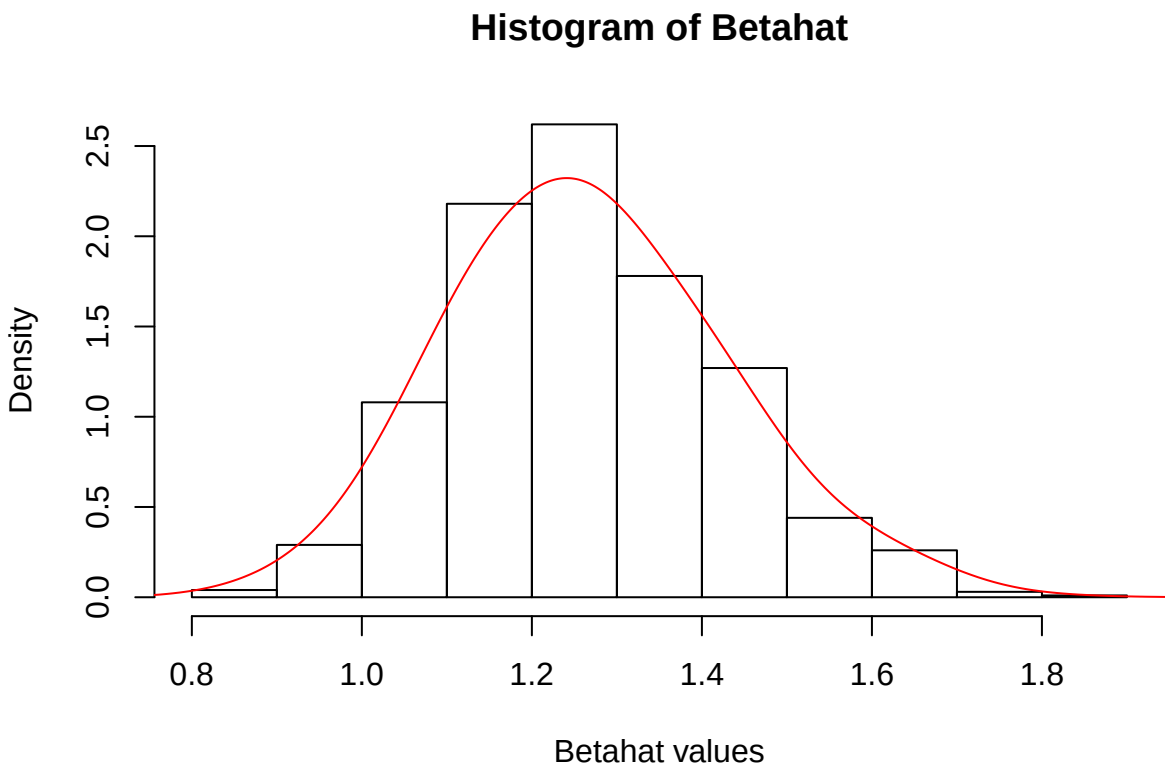
Below you'll find the mean and the volatility (standard deviation) of the $\hat{\beta}$ estimates respectively. you will also find the the histogram of the estimate along with its respective density curve. .22348

```
mean(beta)
```

```
## [1] 1.266229
```

```
sd(beta)
```

```
## [1] 0.1585163
```



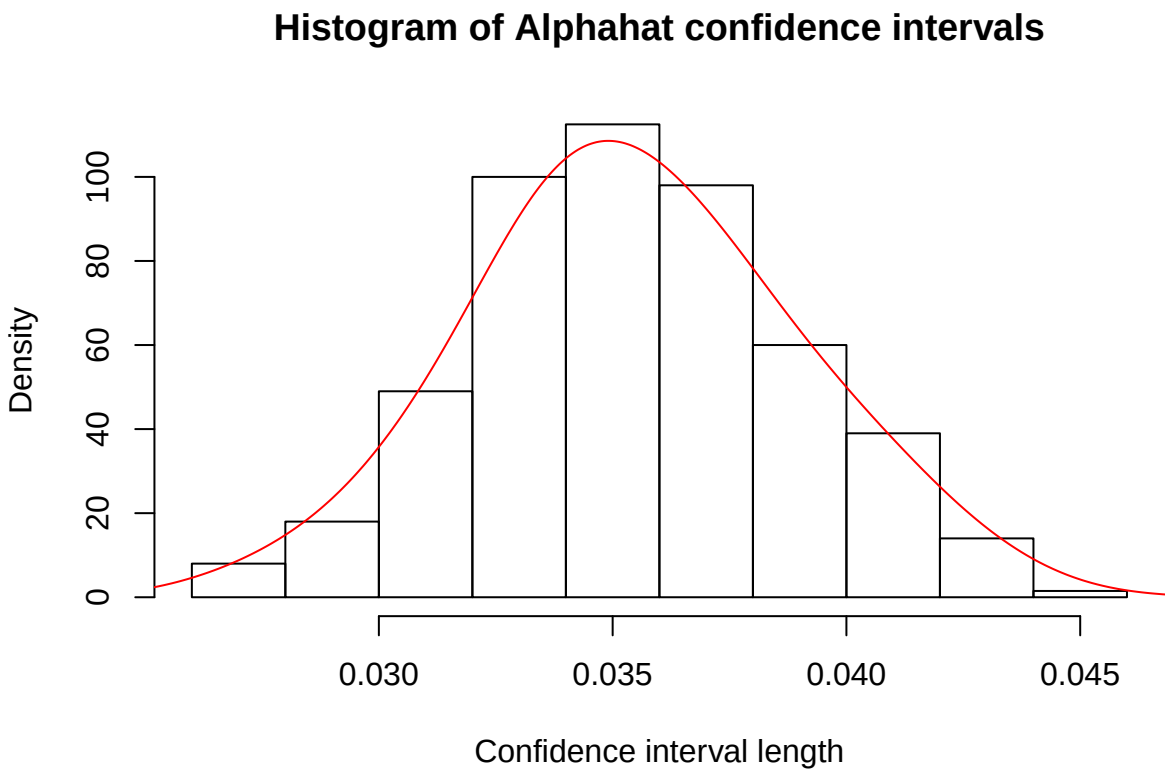
Below you'll find the mean and the volatility (standard deviation) of the $d\hat{\alpha}$ estimates respectively. you will also find the the histogram of the estimate along with its respective density curve.

```
mean(dalpha)
```

```
## [1] 0.03544791
```

```
sd(dalpha)
```

```
## [1] 0.00343088
```



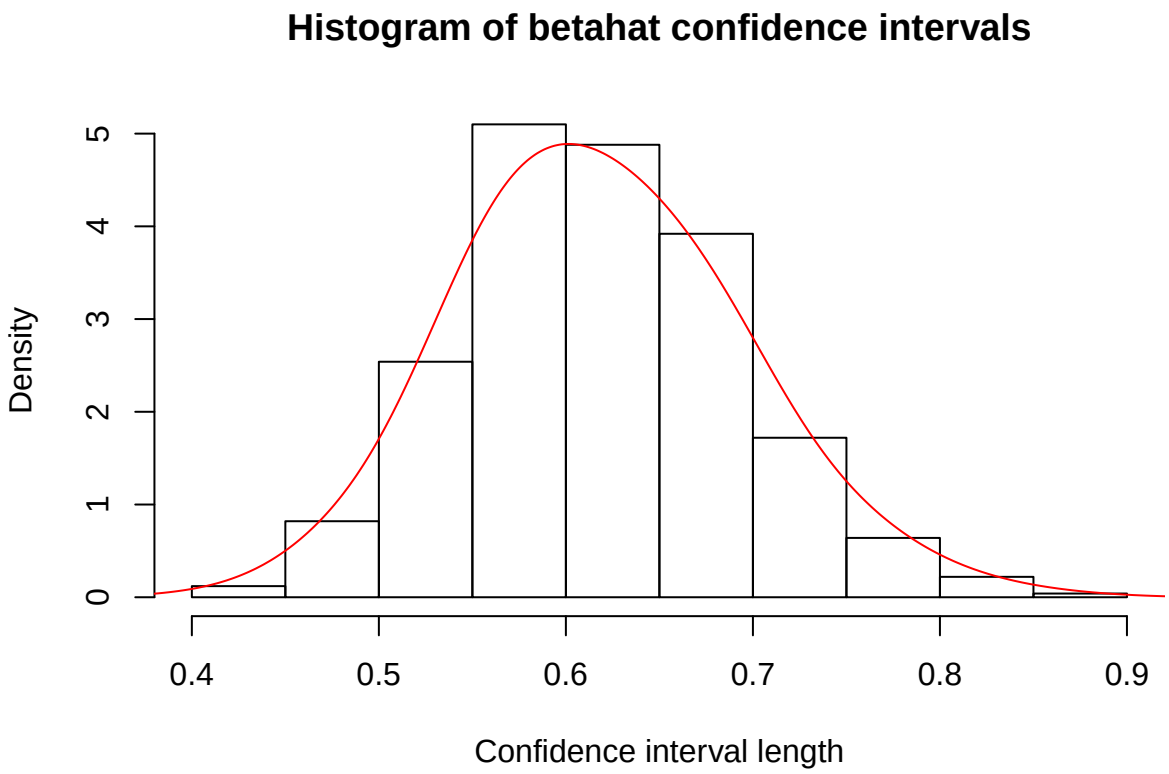
Below you'll find the mean and the volatility (standard deviation) of the $d\hat{\beta}$ estimates respectively. you will also find the the histogram of the estimate along with its respective density curve.

```
mean(dbeta)
```

```
## [1] 0.6179117
```

```
sd(dbeta)
```

```
## [1] 0.07402158
```



Part F) Comparing Bootstrap and linear regression

our results from part E and Part A are very similar. The estimated Alpha and Beta values from both the linear regression and the bootstrap regression do not differ by much. At MOST the beta and alpha estimates differ by .02. Although they are very similar we can confidently choose the bootstrap estimates to be the more reliable and trustworthy results. This is because bootstrap is much more thorough and provides more realizations through multiple scenarios where as linear regression simply goes through one realization (the data we provide). Bootstrapping provides multiple realization by sampling with replacement from the data provided. As a result we prefer the bootstrap estimates.

Question 4

Part A) Probability of cancer after one positive test

The code used to determine the probability of having cancer after one positive test is shown below. We used the equation for Bayesian theorem for sequential learning when constructing our code (shown below).

$P(A|X) = \frac{P(X|A)P(A)}{P(X|A)P(A)+P(X|\ A)P(\ A)}$

```
#ct=probability of having cancer
#cf=probability of not having cancer
#ptct=probability of having cancer given a postive test
#ptcf=probability of not having cancer given a positive test
```

```
#ctn=new probability of having cancer
ct=1
cf=99
ptct=99
ptcf=10
ctn=0
ctn=((ptct*ct)/((ptct*ct)+(ptcf*cf)))
ct=ctn
cf=(1-ctn)
probabilityofcancer=(ctn*100)
print(probabilityofcancer)
```

```
## [1] 9.090909
```

After one positive test the probability of having cancer is 9.09% as shown above.

Part B) How many positive test does it take to have a probabiltiy of cancer over 95%?

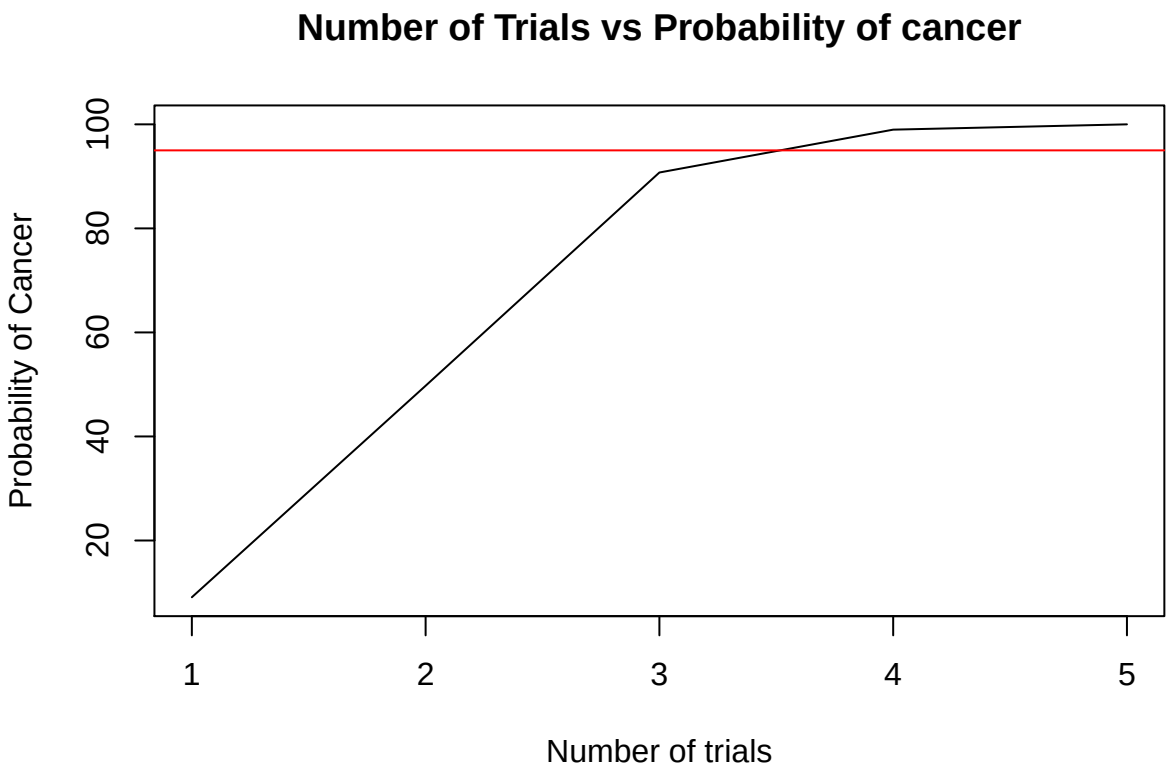
The code used in order to determine the probability of having cancer given a positive test is shown below.The code is designed to loop until the chance of having cancer is less than 95%.

```
#ct=proability of having cancer
#cf=probability of not having cancer
#ptct=probability of having cancer given a postive test
#ptcf=probability of not having cancer given a positive test
#ctn=new probability of having cancer
ct=1
cf=99
ptct=99
ptcf=10
ctn=0
while(ctn < .95){if(ctn < .95){ctn=((ptct*ct)/((ptct*ct)+(ptcf*cf)));
ct=ctn; cf=(1-ctn);
probabilityofcancer=(ctn*100);
print(probabilityofcancer)}else{print(probabilityofcancer)}}
```

```
## [1] 9.090909
## [1] 49.74874
## [1] 90.7416
## [1] 98.9799
```

The outputs above show the probability of having cancer after 1st, 2nd, 3rd,& 4th positive tests respectively.A plot showing the probability of having cancer after a certain number of positive tests is shown below. As we can see from the graph is takes 4 positive tests in a row to reach a probability of cancer greater than 95%.

```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
## graphical parameter "type" is obsolete
```



Part C) What is the probability of having cancer after a negative test on the third trial.

```
#ct=proability of having cancer
#cf=probability of not having cancer
#ptct=probability of having cancer given a postive test
#ptcf=probaiblity of not having cancer given a positive test
#ctn=new probability of having cancer
#pfct=probability of having a false test given cancer
#pfcf=probability of having a false test given not having cancer
ct=49.78
cf=50.22
ptct=99
ptcf=10
ctn=.4978
pfct=1
pfcf=90

#The third test being negative drops to the proability of having cancer to 1.08%
ctn=((pfct*ct)/((pfct*ct)+(pfcf*cf)))
ct=ctn
cf=(1-ctn)
probabilityofcancer=(ctn*100)
print(probabilityofcancer)
```

```
## [1] 1.089378
```

After 2 positive tests and a third negative test the probability of having cancer drops from 49.78% to 1.08%.

```
#It takes another 4 positive tests (7 tests in total) for the probability of having cancer to reach > 9
while(ctn < .95){ctn=((ptct*ct)/((ptct*ct)+(ptcf*cf)));
ct=ctn;
cf=(1-ctn);
probabilityofcancer=(ctn*100); print(probabilityofcancer)}
```

```
## [1] 9.831621
## [1] 51.91056
## [1] 91.44322
## [1] 99.06365
```

The output above shows the probability of having cancer after the 3rd, 4th, 5th, 6th, and 7th positive test respectively. A plot showing the probability of having cancer after a certain number of positive tests is shown below. As we can see from the graph, the probability of having cancer drops significantly when we get a third negative test. It takes another 4 positive tests afterwords to reach a probability of having cancer to be greater than 95%.

```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
## graphical parameter "type" is obsolete
```