403A Project 1

Ross Lewis, Qian (Alice) Li, Adam Jacobson, Chunxiao Jing

October 24, 2018

1 a

We will use the Affairs data from AER.

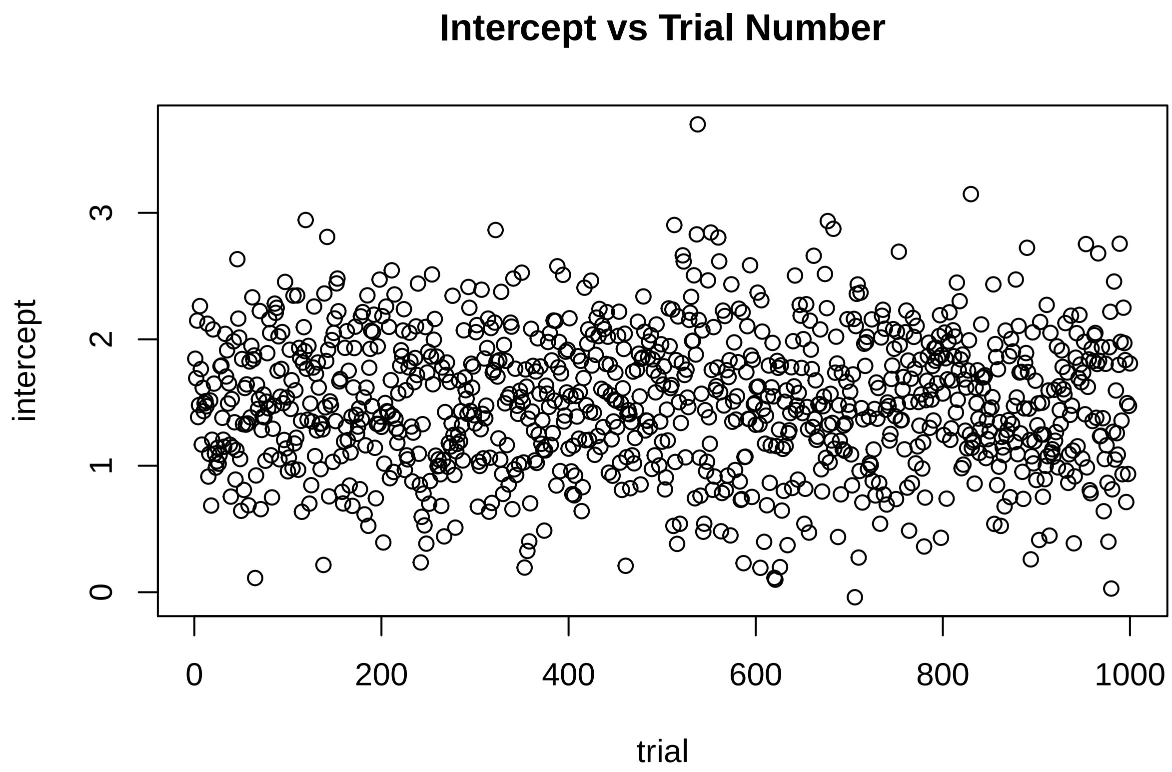
```
library(AER)
library(ggplot2)
library(reshape)
library(boot)
library("plotrix")
```

```
data("Affairs")
m2=lm(Affairs$affairs~Affairs$age+Affairs$yearsmarried)
summary(m2)
```

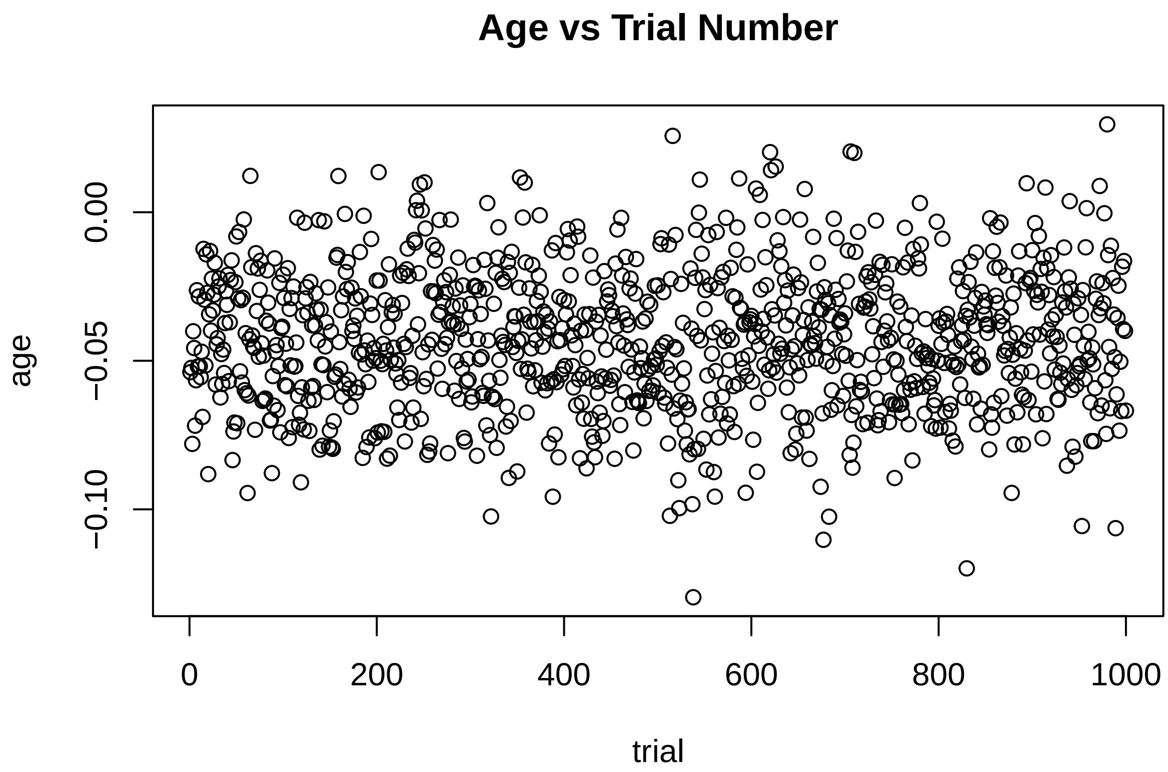
```
##
## Call:
## lm(formula = Affairs$affairs ~ Affairs$age + Affairs$yearsmarried)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6300 -1.7312 -0.9970 -0.3918 11.6199
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.53484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

```
##From the output, we are sure that the LS estimates of the two parameters is significant at 95%

#use boot to estimate the Bootstrap estimates of the model parameters
set.seed(1234)
rsq <- function(formula, data, indices) {
  d <- data[indices,]
  bootstrap <- lm(formula, data=d)
  return(coef(bootstrap))
}

results <- boot(data=Affairs, statistic=rsq,
                R=1000, formula=affairs~age+yearsmarried)
```

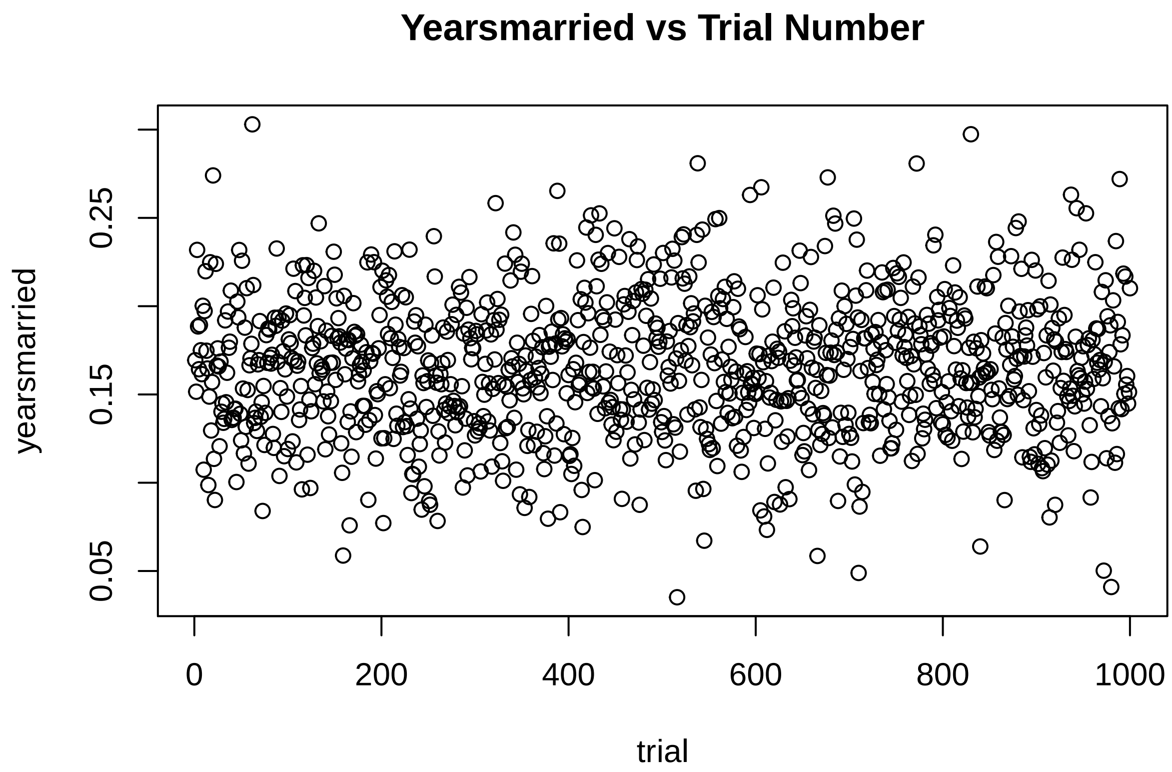
```
#estimate vs trial
plot(results$t[,1],xlab="trial",ylab="intercept",main="Intercept vs Trial Number")#intercept
```



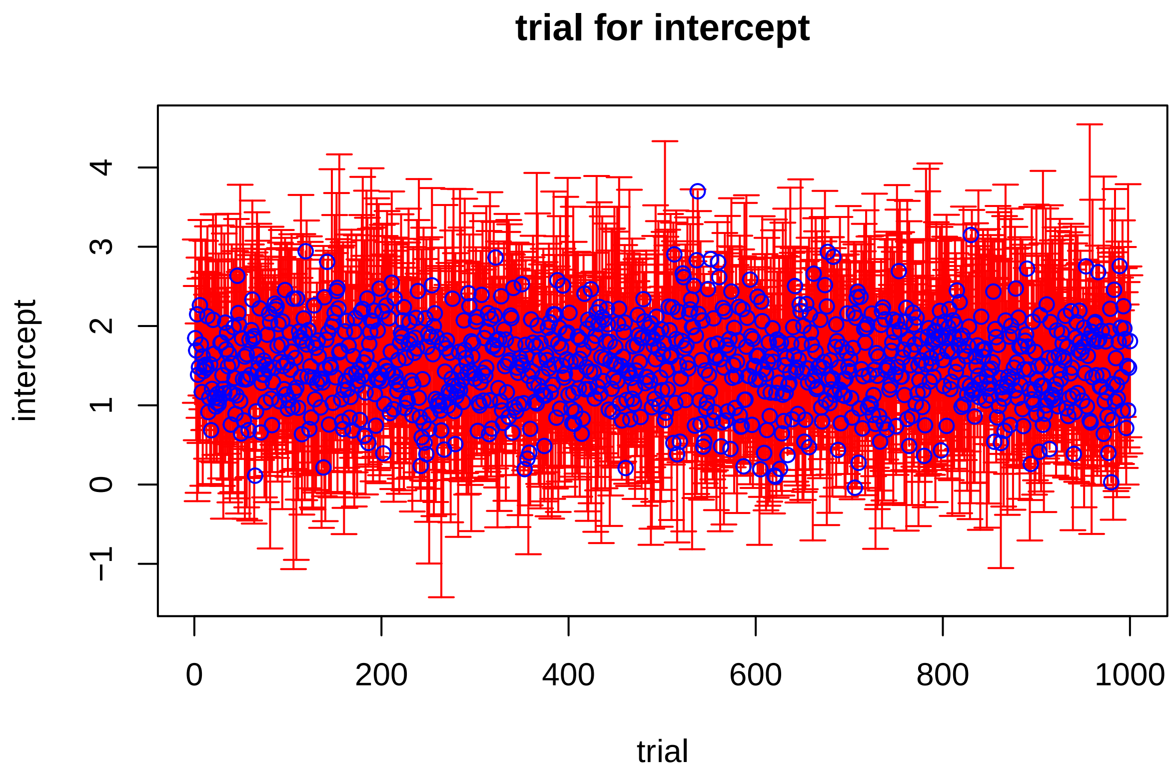

```
plot(results$t[,2],xlab="trial",ylab="age",main="Age vs Trial Number")#age
```



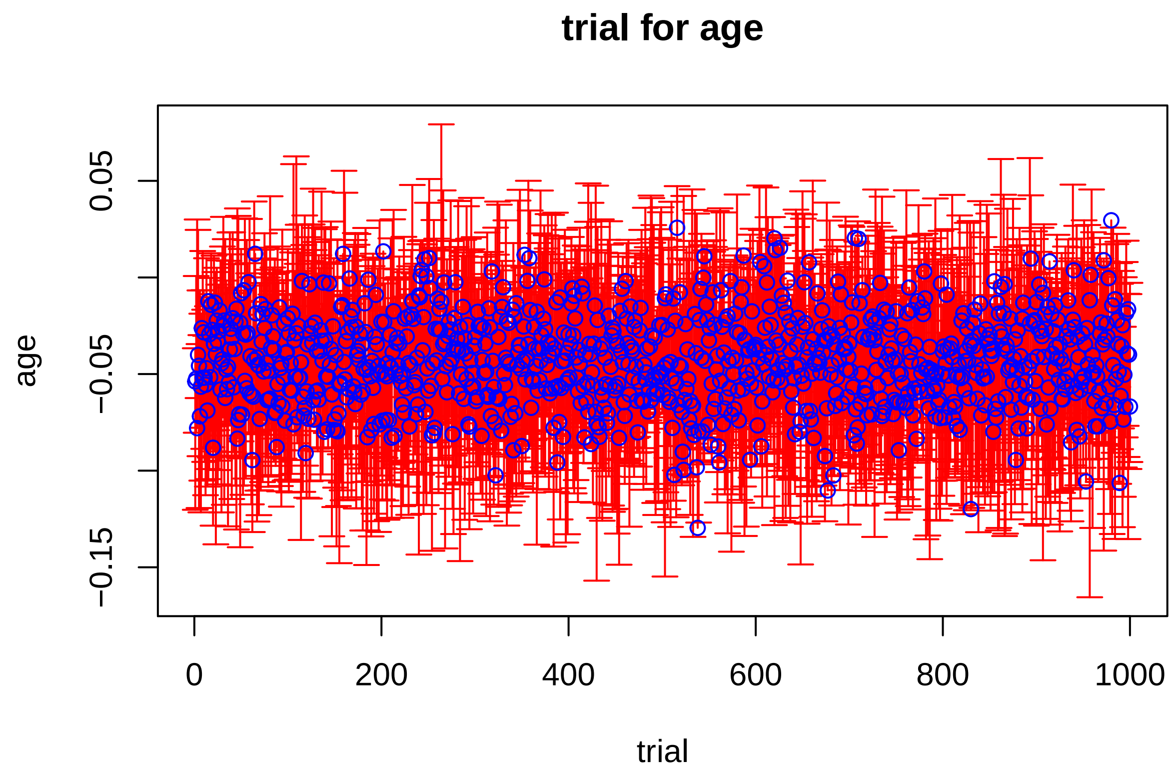

```
plot(results$t[,3],xlab="trial",ylab="yearsmarried",main="Yearsmarried vs Trial Number")#yearsmarried
```



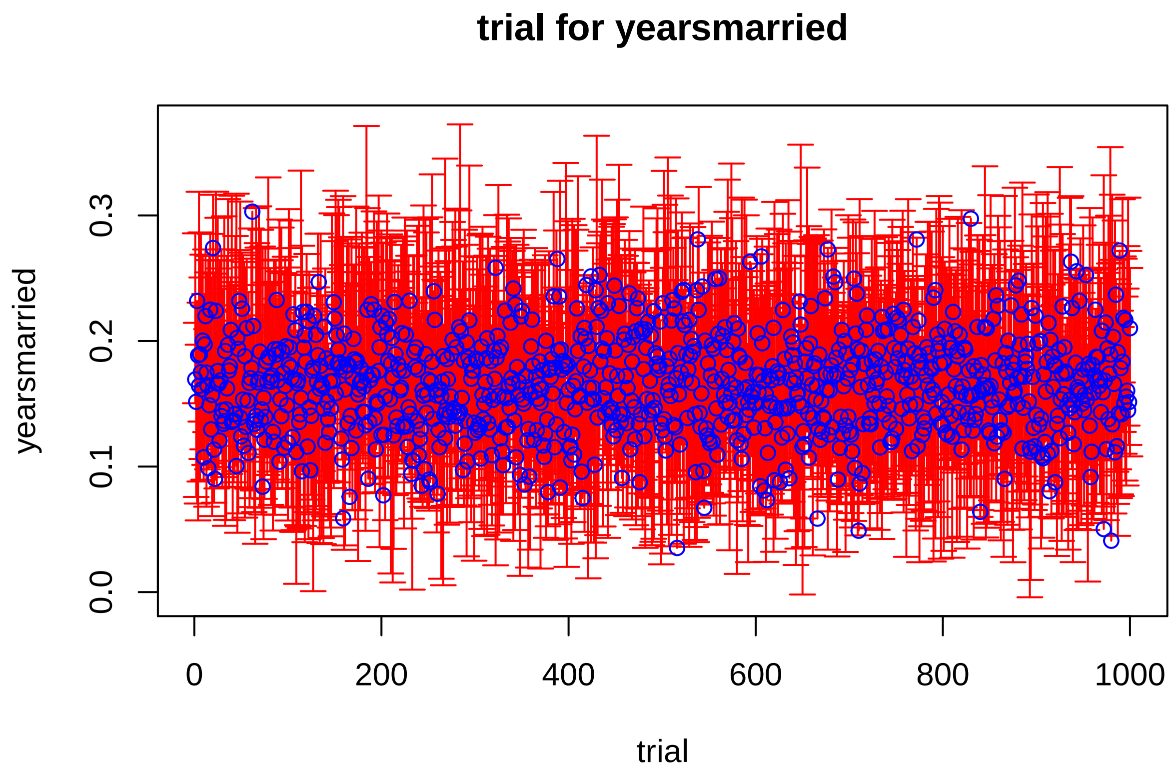

```
#ci vs trial
ab<- function(formula, data, indices) {
  d <- data[indices,]
  bootstrap <- lm(formula, data=d)
  return(confint(bootstrap))
}

ci <- boot(data=Affairs, statistic=ab,
           R=1000, formula=affairs~age+yearsmarried)

##ci for intercept
plotCI(x=1:1000,results$t[,1],ui=ci$t[,4],li=ci$t[,1],col="blue",scol="red",xlab="trial",ylab="intercep
```




```
##ci for age
plotCI(x=1:1000,results$t[,2],ui=ci$t[,5],li=ci$t[,2],col="blue",scol="red",xlab="trial",ylab="age",mai
```




```
##ci for yearsmarried
plotCI(x=1:1000,results$t[,3],ui=ci$t[,6],li=ci$t[,3],col="blue",scol="red",xlab="trial",ylab="yearsmar
```



The results of parameters and CI change all the time through the 1000 trials. So the stability of bootstrap estimates is not very good.

```
#compare estimats
mean(results$t[,1])# intercept
```

```
## [1] 1.505649
```

```
m2$coefficients[1]
```

```
## (Intercept)
##    1.534838
```

```
mean(results$t[,2])# age
```

```
## [1] -0.0432134
```

```
m2$coefficients[2]
```

```
## Affairs$age
##  -0.0449423
```

```
mean(results$t[,3])# yearsmarried
```

```
## [1] 0.1651957
```

```
m2$coefficients[3]
```

```
## Affairs$yearsmarried
##            0.1688902
```

```r
#compare CI
boot.ci(results, type="bca", index=1)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 1)
##
## Intervals :
## Level       BCa
## 95%   ( 0.477,  2.622 )
## Calculations and Intervals on Original Scale
```

```r
boot.ci(results, type="bca", index=2)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 2)
##
## Intervals :
## Level       BCa
## 95%   (-0.0875,  0.0010 )
## Calculations and Intervals on Original Scale
```

```r
boot.ci(results, type="bca", index=3)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca", index = 3)
##
## Intervals :
## Level       BCa
## 95%   ( 0.0948,  0.2520 )
## Calculations and Intervals on Original Scale
```

```r
confint(m2)
```

```
##                              2.5 %        97.5 %
## (Intercept)           0.45922653  2.6104497887
## Affairs$age          -0.08935362 -0.0005309759
## Affairs$yearsmarried  0.09484536  0.2429350150
```

(1)The result of LS estimates is 1.534838 for intercept,-0.0449423 for age and 0.1688902 for yearsmarried. The the mean of overall bootstrap results is 1.505649 for intercept, -0.0432134 for age and 0.1651957 for yearsmarried. These values are quite close to each other.

(2)The confidence intervals of Bootstrap estimates is also close to that of LS estimates.In addition, it is noticed that the LS estimates is always in the CI of bootstrap, while the estimates of bootstrap is also in the CI of LS.

1 b

```r
library(ggplot2)
library(bayesplot)
```

```
## This is bayesplot version 1.6.0
## - Online documentation and vignettes at mc-stan.org/bayesplot
## - bayesplot theme set to bayesplot::theme_default()
##    * Does _not_ affect other ggplot2 plots
##    * See ?bayesplot_theme_set for details on theme setting
##
## Attaching package: 'bayesplot'
## The following object is masked from 'package:plotrix':
##
##     plot_bg
```

```r
library(MCMCpack)
```

```
## Loading required package: coda
## Loading required package: MASS
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
## ## Copyright (C) 2003-2018 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
## ##
## ## Support provided by the U.S. National Science Foundation
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```r
#use mcmc regression
set.seed(1234)
mcmc_regression<-MCMCregress(affairs~age+yearsmarried,data = Affairs)
MCMC<-summary(mcmc_regression)
MCMC
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean      SD  Naive SE Time-series SE
## (Intercept)   1.53552 0.54460 0.0054460      0.0054479
## age          -0.04508 0.02252 0.0002252      0.0002267
## yearsmarried  0.16922 0.03784 0.0003784      0.0003784
## sigma2       10.49724 0.61140 0.0061140      0.0061140
##
## 2. Quantiles for each variable:
```

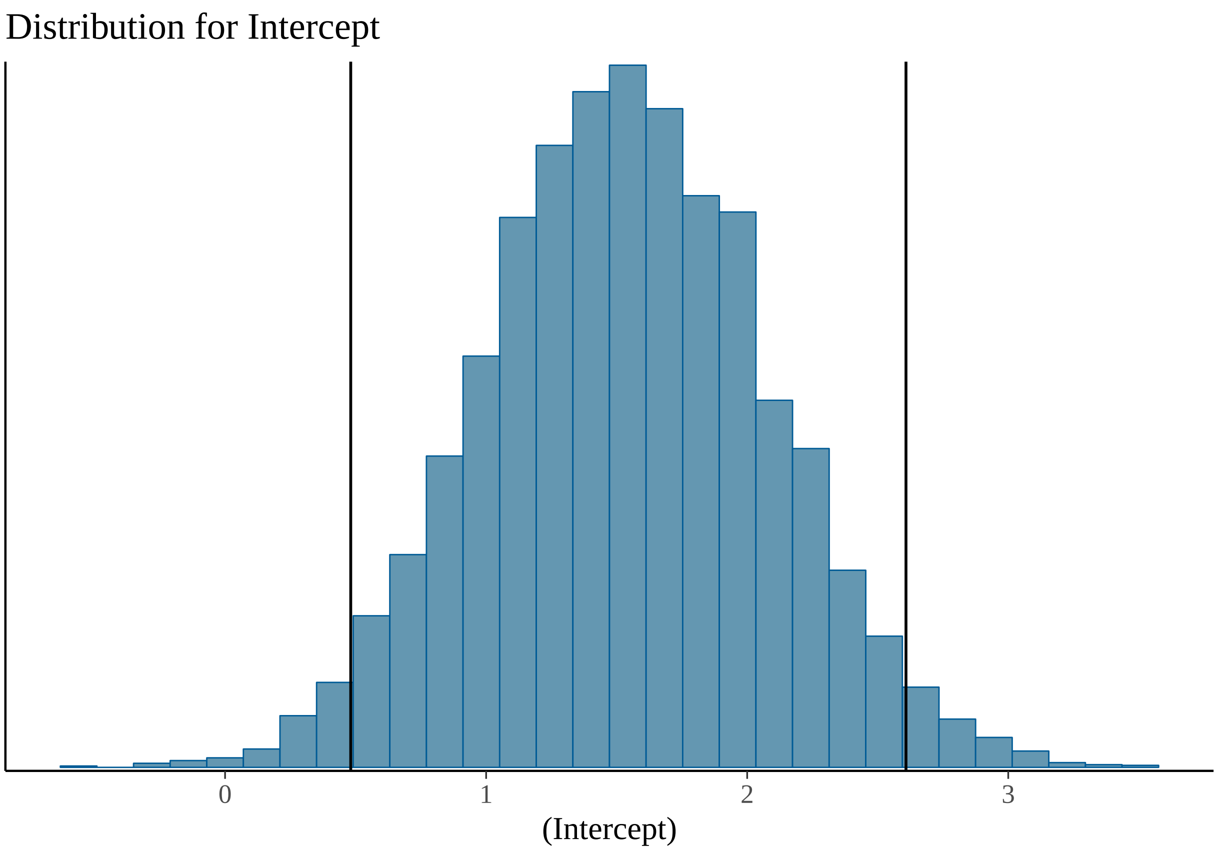
```
##
##                  2.5%      25%     50%      75%     97.5%
## (Intercept)   0.48136  1.16409  1.5321  1.90486  2.608247
## age          -0.08904 -0.06033 -0.0448 -0.02963 -0.001961
## yearsmarried  0.09577  0.14328  0.1691  0.19510  0.243632
## sigma2        9.36190 10.07436 10.4734 10.89080 11.772888
```

```
#extract posterior from fitted model
posterior<-as.array(mcmc_regression)
dimnames(posterior)
```

```
## [[1]]
## NULL
##
## [[2]]
## [1] "(Intercept)"  "age"          "yearsmarried" "sigma2"
```

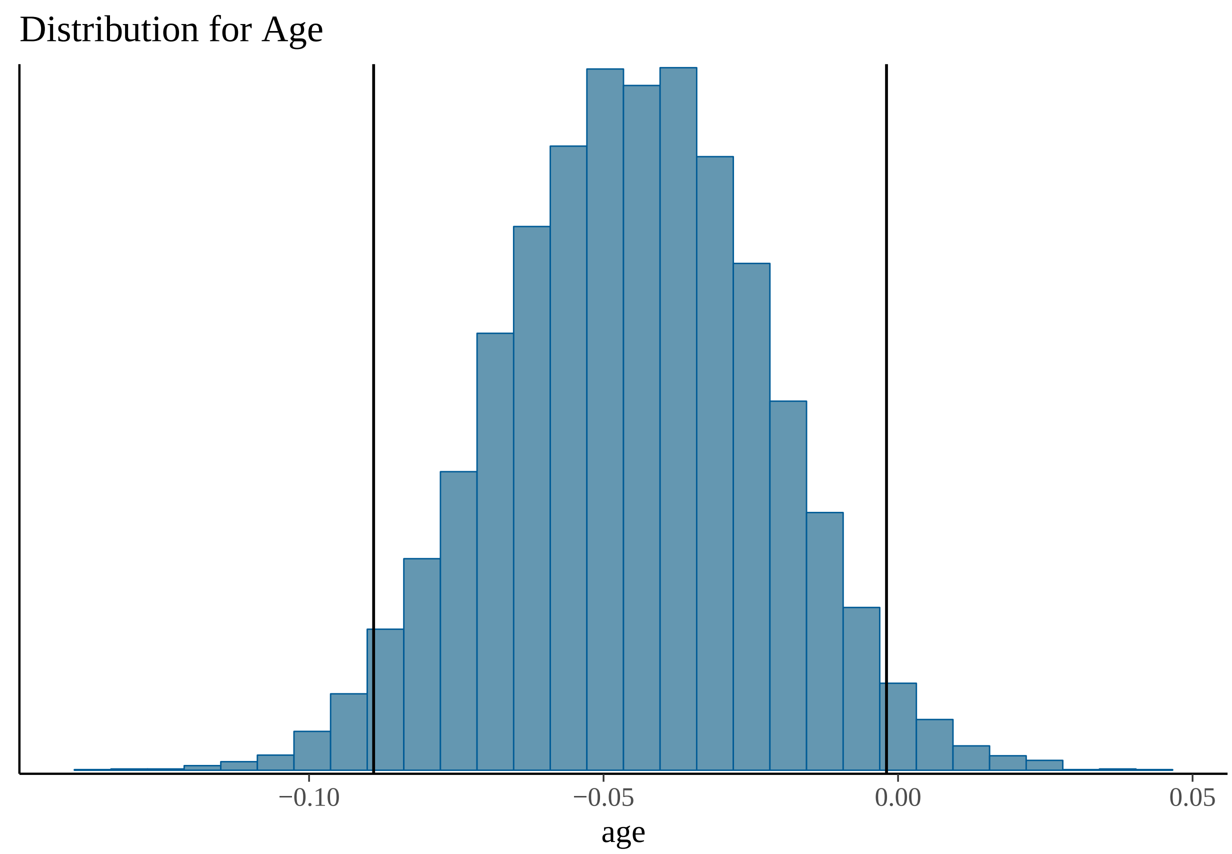
```
#plot posterior distribution
color_scheme_set("blue")
mcmc_hist(posterior,pars ="(Intercept)")+vline_at(v=MCMC$quantiles[1,1])+vline_at(v=MCMC$quantiles[1,5])
```

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



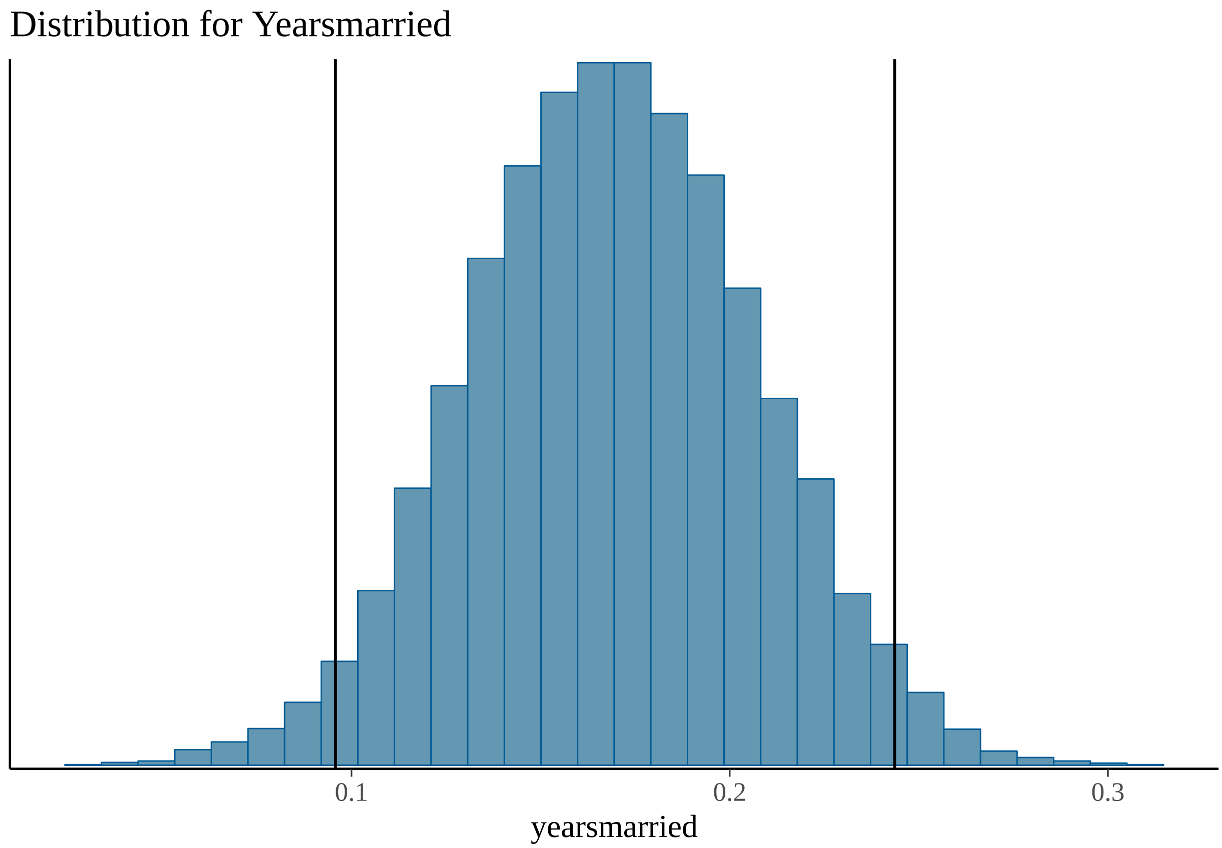

```
mcmc_hist(posterior,pars ="age")+vline_at(v=MCMC$quantiles[2,1])+vline_at(v=MCMC$quantiles[2,5])+labs(ti
```

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




```
mcmc_hist(posterior,pars ="yearsmarried")+vline_at(v=MCMC$quantiles[3,1])+vline_at(v=MCMC$quantiles[3,5
```

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



(1) The result of LS estimates is 1.534838 for intercept,-0.0449423 for age and 0.1688902 for yearsmarried. The result of MCMCregression is 1.53552 for intercept, -0.04508 for age and 0.16922 for yearsmarried. These values are also quite close to each other.

(2) LS confidence intervals for parameters are (0.45922653, 2.6104497887) for intercept, (-0.08935362, -0.0005309759) for age and (0.09484536, 0.2429350150) for years married. And MCMCregression parameters' CI are (0.48136, 2.608247) for intercept,(-0.08904, -0.001961) for age and (0.09577, 0.243632) for yearsmarried. The vectors of two estimate methods are close.

(3) Also, it is noticed that the LS estimates is always in the CI of bootstrap, while the estimates of bootstrap is also in the CI of LS.

1 c

We think Bayesian regression is better.

Though the parameters estimated from 3 different methods are quite close to each other, they still have some differences.

(1)Bayesian vs LS: LS regression may be more useful is the sample size is large enough. However the number of observations in "Affairs" is 601, it is not large enough to get an unbiased result through LS regression. As for bayesian regression, it is suitable for a small sample size. So bayesian is better compared with LS regression

(2)Bayesian vs Bootstrap: Bootstrap assumes that "sampling with replacement from the original sample of size n mimics taking a sample of size n from a larger population", but bootstrap may fail in a small sample size. In addition, bootstrap is only based on the original data, we cannot get more information from it.

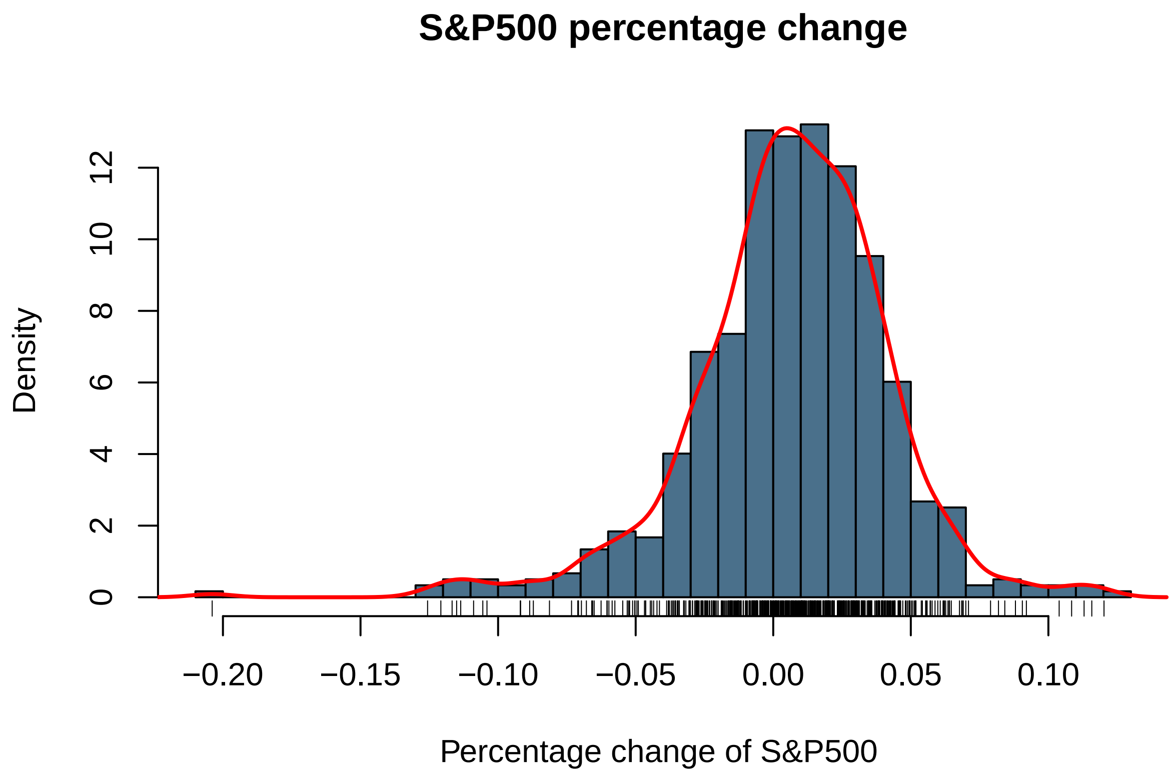
2 a

```
setwd("C:/Users/rossw/Documents/MAE Program/Econometrics 403A/Homework")
#setwd("C:/Users/alice/OneDrive/Documents/R")
library(Quandl)
mykey="srRKGuDiUuuNeR1C_88u"
```

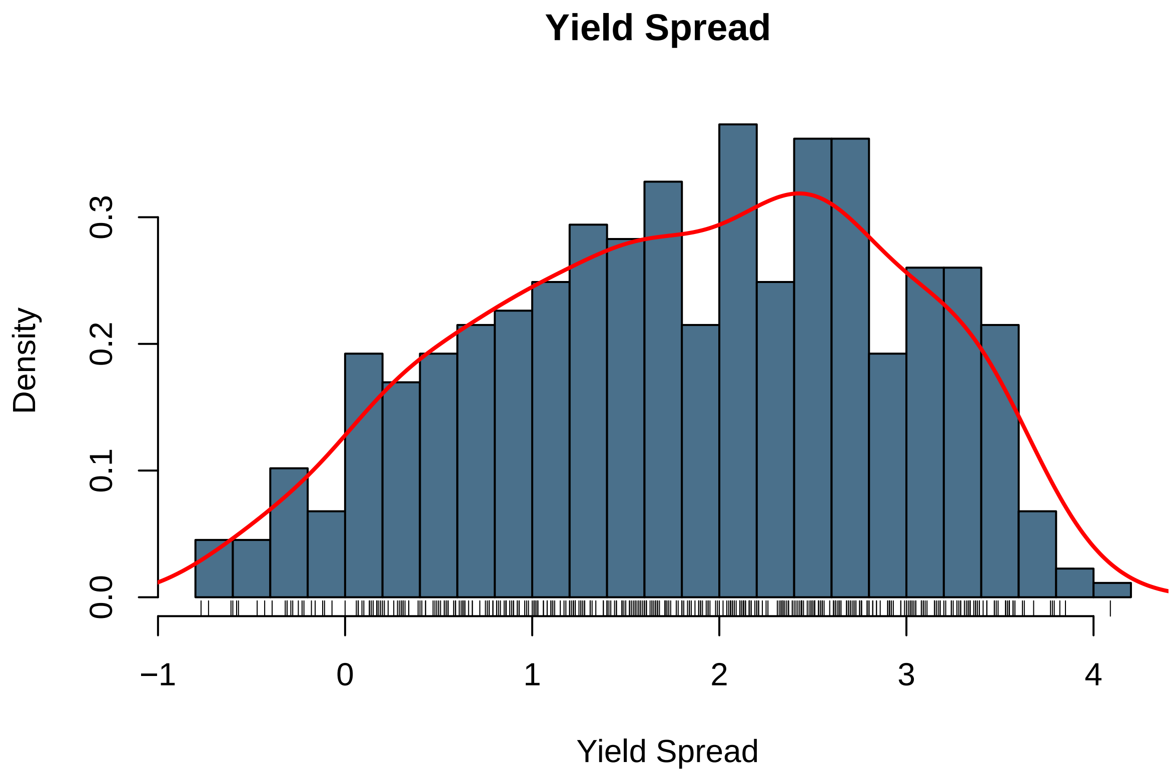
```
## S&P 500 index
data_sp500<-Quandl("MULTPL/SP500_REAL_PRICE_MONTH", api_key="srRKGuDiUuuNeR1C_88u", collapse="monthly",s
sp500<-data_sp500$Value
##S&P500 percentage change
data_sp500$diff = c(-diff(data_sp500$Value, lag = 1),NA)
data_sp500 <- data_sp500[complete.cases(data_sp500), ]
##data_sp500$diff = na.omit(data_sp500$diff)
data_sp500$perc = data_sp500$diff/(data_sp500$Value-data_sp500$diff)

##Yield Spread
data_yield<-Quandl("FRED/T10Y3M", api_key="srRKGuDiUuuNeR1C_88u", collapse="monthly", start_date="1968-0
yield<-data_yield$Value

## Three month treasury rate
data_rate3<-Quandl("FRED/TB3MS", api_key="srRKGuDiUuuNeR1C_88u", collapse="monthly", start_date="1968-09
rate3<-data_rate3$Value

## Japanese Central Bank's interest rates
## This data downlode from: https://fred.stlouisfed.org/series/IRSTCB01JPM156N
data_Japan<-read.csv("interest-rate-in-Japan.csv",stringsAsFactors=FALSE)
Japanrate<-data_Japan$IRSTCB01JPM156N
```

Plot the histogram of the data and overlay the respective density curve.

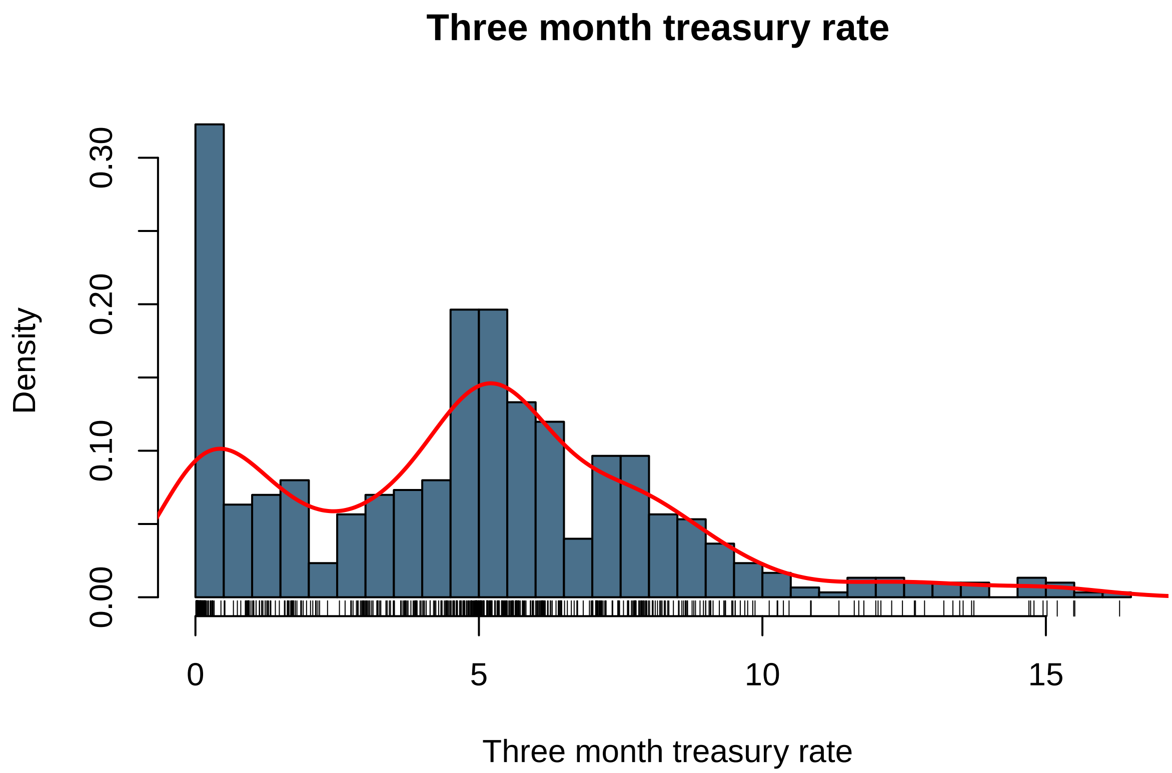
```
#data_sp500$perc
hist(na.omit(data_sp500$perc),breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main ="S&P500 percentage c
lines(density(na.omit(data_sp500$perc)),col="red",lwd=2)
rug(data_sp500$perc)
```




```
## Yield Spread
hist(yield,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Yield Spread ",xlab = " Yield Spread
lines(density(yield),col="red",lwd=2)
rug(yield)
```

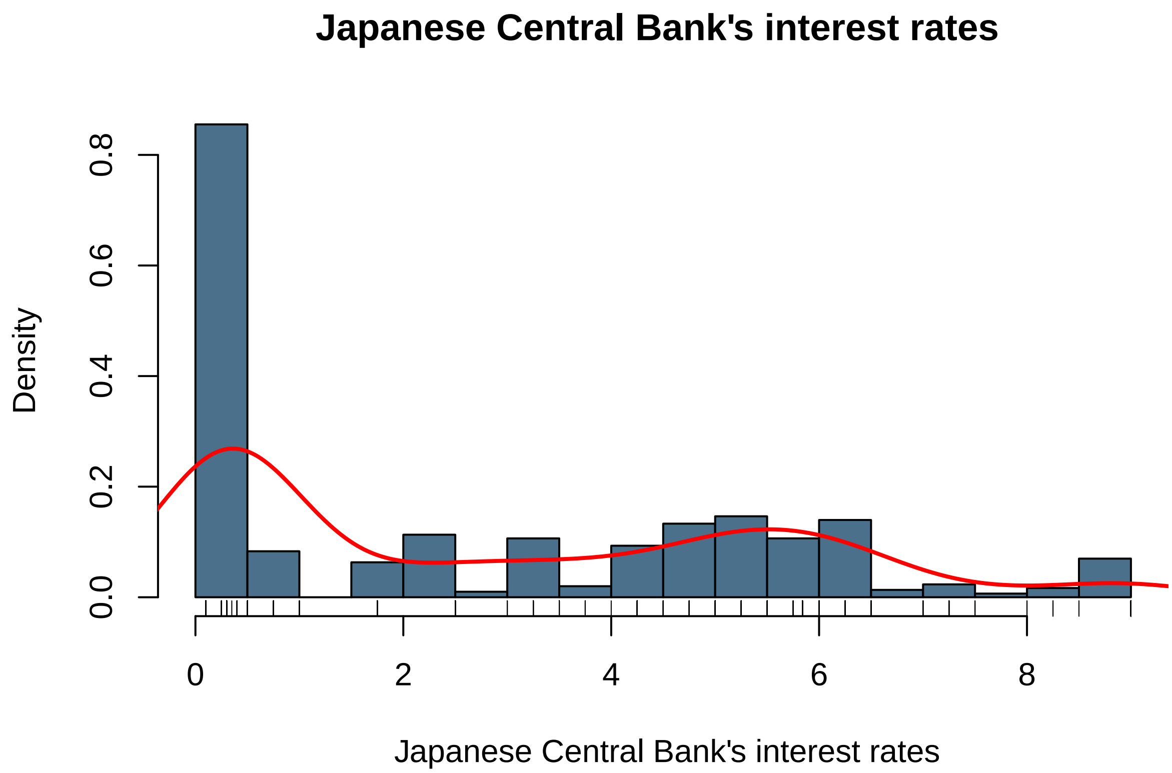



```
## Three month treasury rate
hist(rate3,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Three month treasury rate ",xlab = "
lines(density(rate3),col="red",lwd=2)
rug(rate3)
```




```
#rate3[which(rate3 < 0)]
```

```
##Japanese Central Bank's interest rates
hist(Japanrate,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Japanese Central Bank's interest
lines(density(Japanrate),col="red",lwd=2)
rug(Japanrate)
```



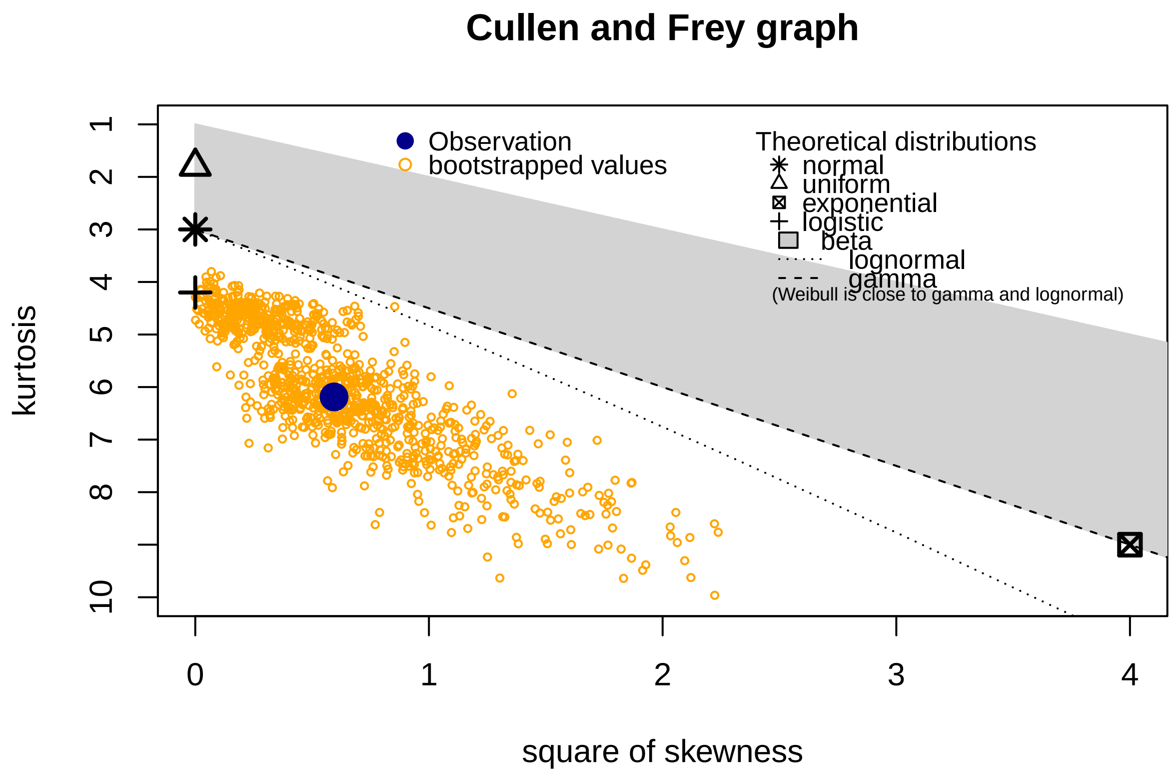
2 b

Fit a distribution to each histogram

```
library(fitdistrplus)
```

Percentage change of S&P 500

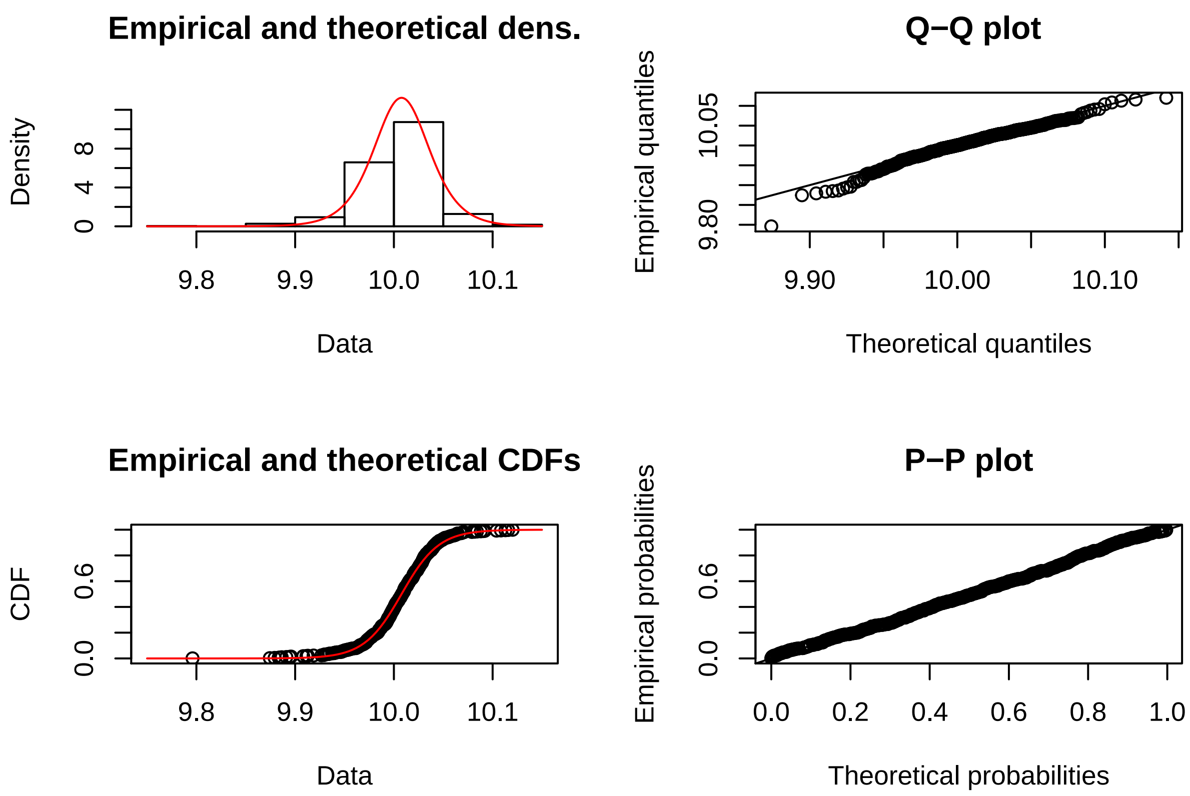
```
## Percentage change of S&P 500
descdist(data_sp500$perc,boot = 1001)
```



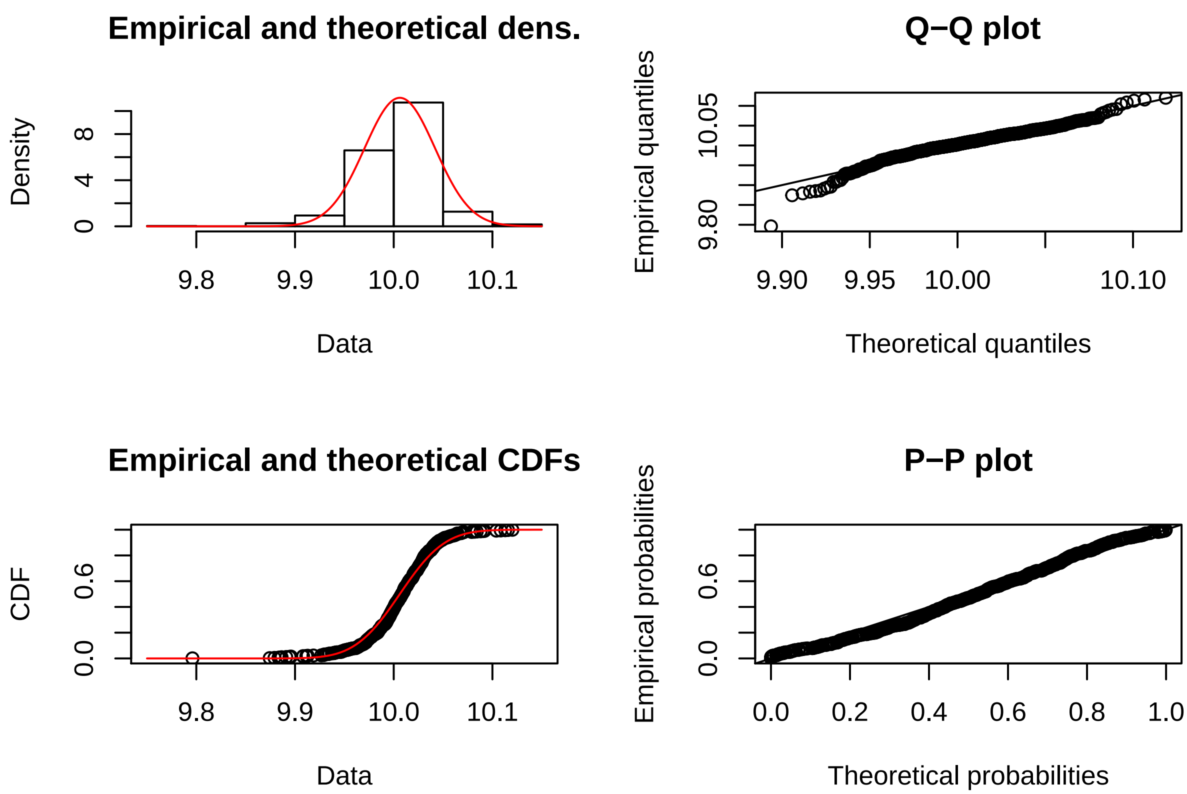

```
## summary statistics
## ------
## min:  -0.2039114   max:  0.1202171
## median:  0.008303749
## mean:  0.006178183
## estimated sd:  0.03581928
## estimated skewness:  -0.770703
## estimated kurtosis:  6.187739
```

From this graph, we can see Percentage change of S&P 500 may fit logistic ,normal and lognormal distribution. So we need further test:

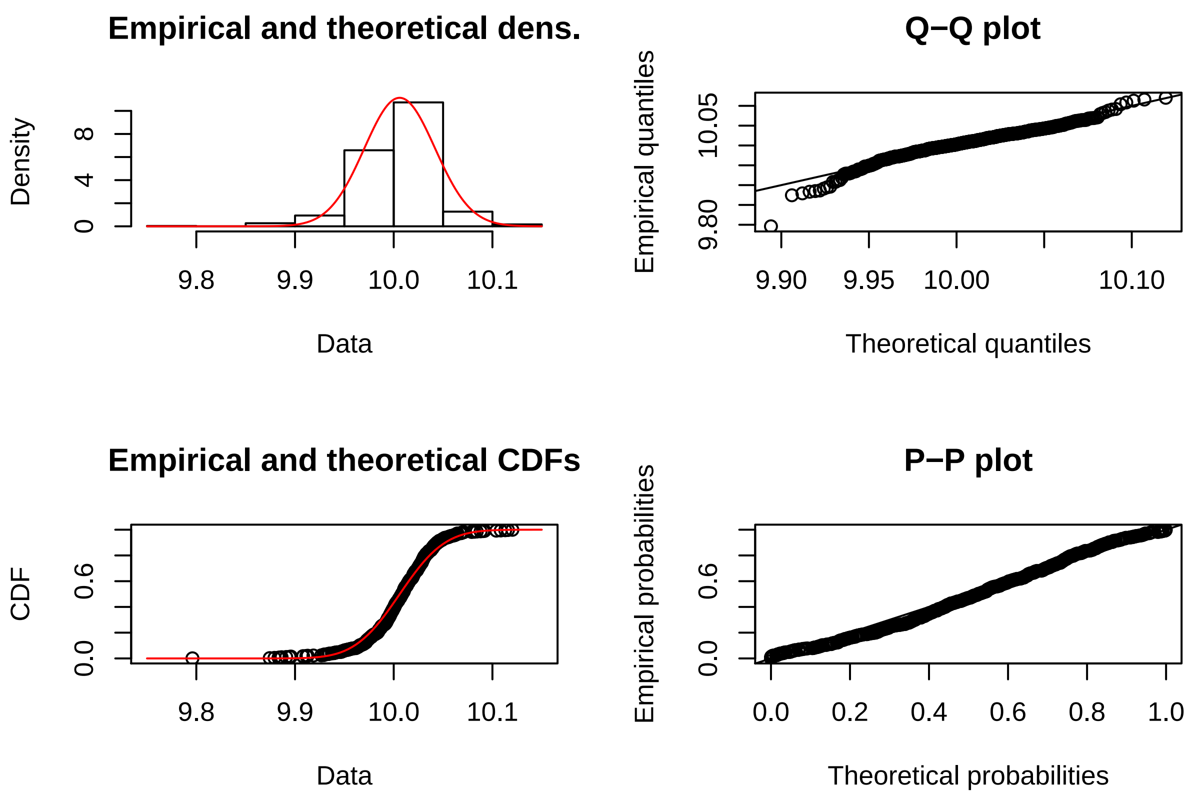
```
##fit logistic
data_sp500$perc_10<-data_sp500$perc+10
fitsp500_logis<-fitdist(data_sp500$perc_10,"logis")
plot(fitsp500_logis)
```



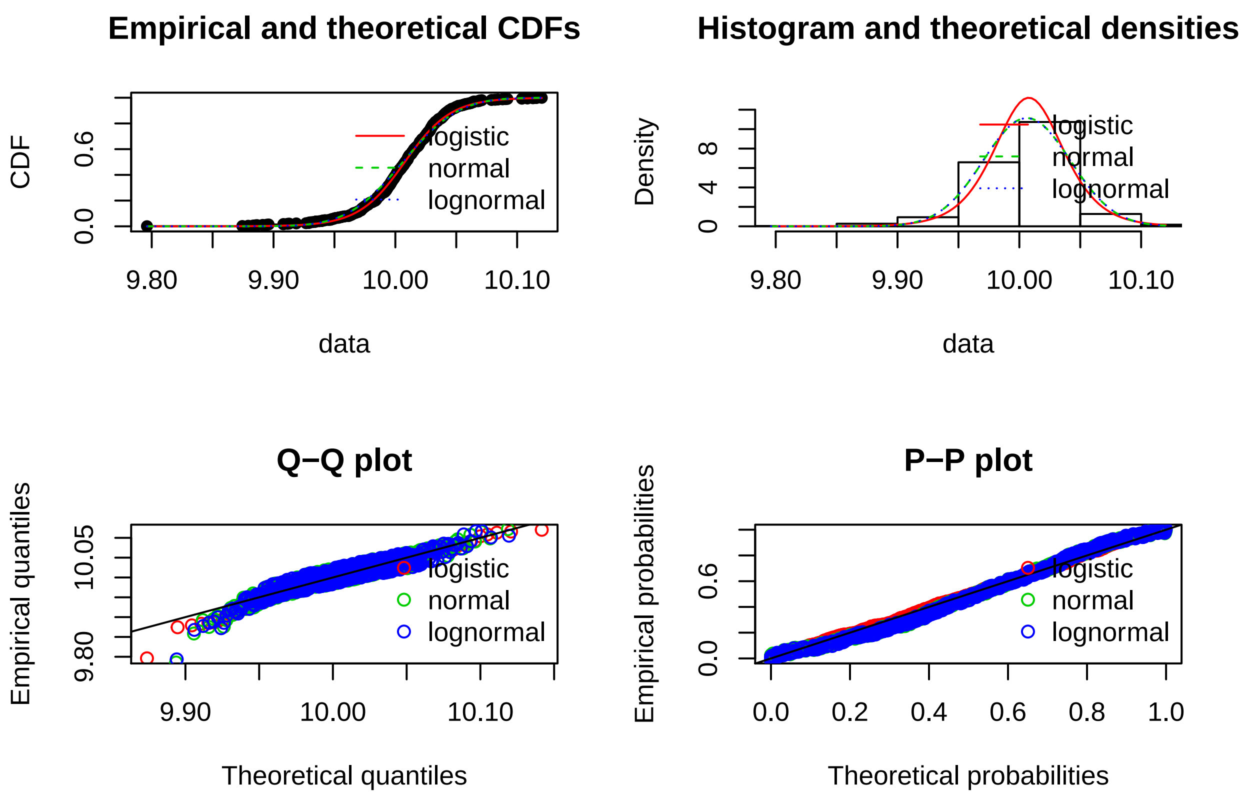

```
##fit normal
fitsp500_norm<-fitdist(data_sp500$perc_10,"norm")
plot(fitsp500_norm)
```




```
##fit lognormal distribution
fitsp500_lnorm<-fitdist(data_sp500$perc_10,"lnorm")
plot(fitsp500_lnorm)
```




```
par(mfrow=c(2,2))
plot.legend<-c("logistic","normal","lognormal")
cdfcomp(list(fitsp500_logis,fitsp500_norm,fitsp500_lnorm),legendtext=c("logistic","normal","lognormal")
denscomp(list(fitsp500_logis,fitsp500_norm,fitsp500_lnorm),legendtext=c("logistic","normal","lognormal"
qqcomp(list(fitsp500_logis,fitsp500_norm,fitsp500_lnorm),legendtext=c("logistic","normal","lognormal"))
ppcomp(list(fitsp500_logis,fitsp500_norm,fitsp500_lnorm),legendtext=c("logistic","normal","lognormal"))
```



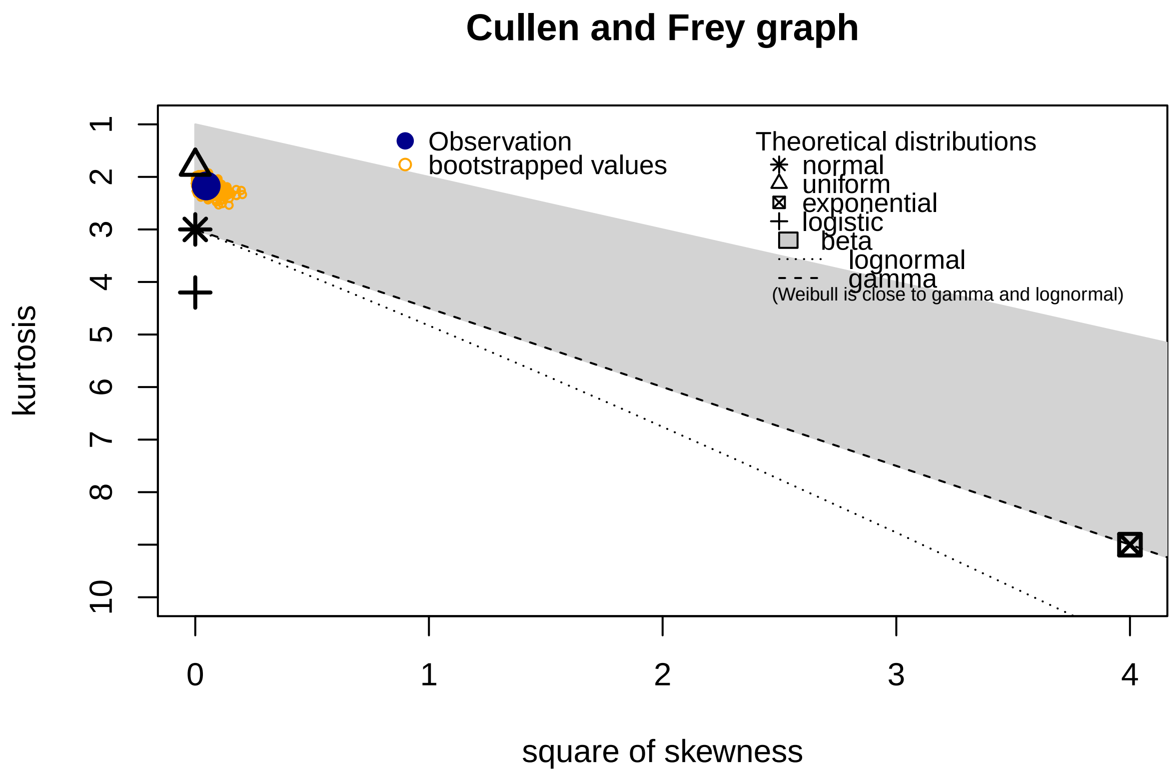

```
gofstat(list(fitsp500_logis,fitsp500_norm,fitsp500_lnorm),fitnames=c("logistic","normal","lognormal"))
```

```
## Goodness-of-fit statistics
##                                logistic     normal   lognormal
## Kolmogorov-Smirnov statistic 0.02981486 0.0720991 0.07285427
## Cramer-von Mises statistic   0.09682896 0.7329897 0.75178514
## Anderson-Darling statistic   1.11302419 4.7813910 4.89357358
##
## Goodness-of-fit criteria
##                                  logistic    normal lognormal
## Akaike's Information Criterion -2335.529 -2281.756 -2280.080
## Bayesian Information Criterion -2326.742 -2272.969 -2271.293
```

Because there are negtive values in the percentage changes of S&P 500, we add a constant to the returns first and then fit the distribution. It turns out that Percentage change of S&P 500 fits logistic distribution best.

Yield Spread

```
descdist(yield,boot = 1001)
```



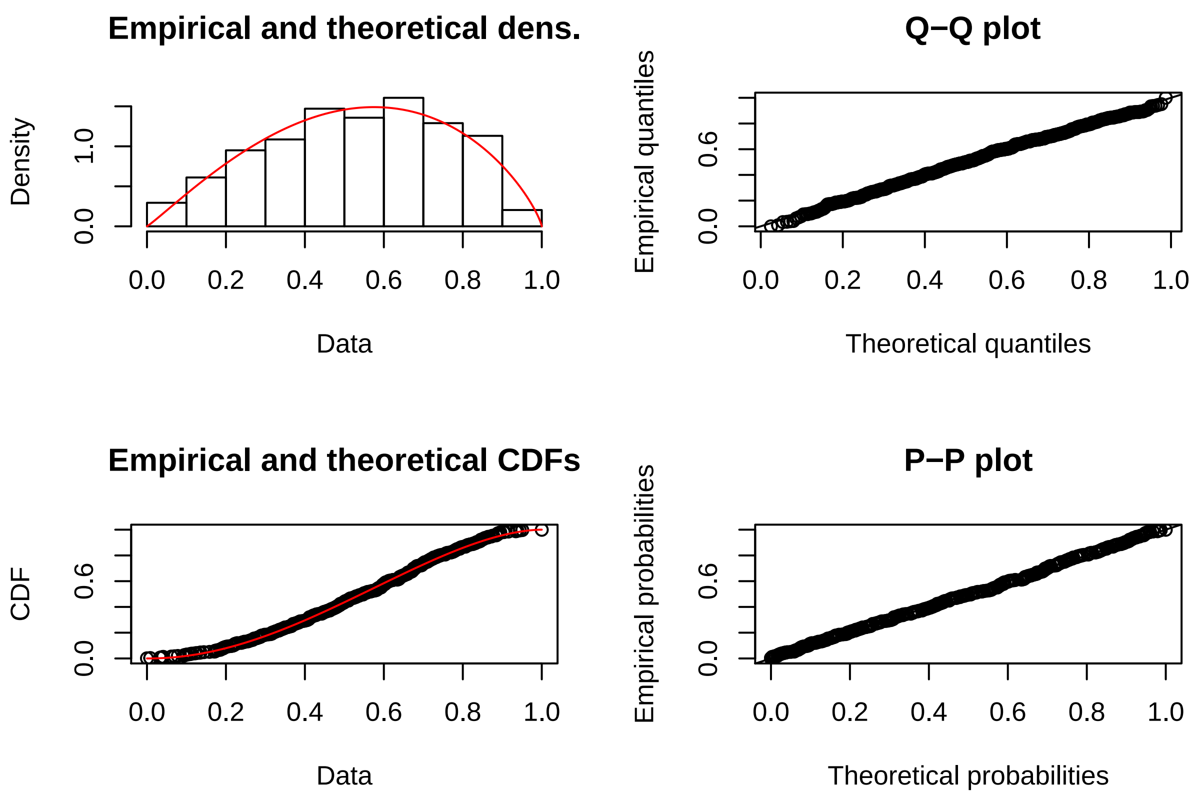

```
## summary statistics
## ------
## min:  -0.77   max:  4.09
## median:  1.89
## mean:  1.819525
## estimated sd:  1.097206
## estimated skewness:  -0.2164585
## estimated kurtosis:  2.171586
```

This variable may fit beta, uniform and normal distribution. So we choose these three distributions to have a further test.

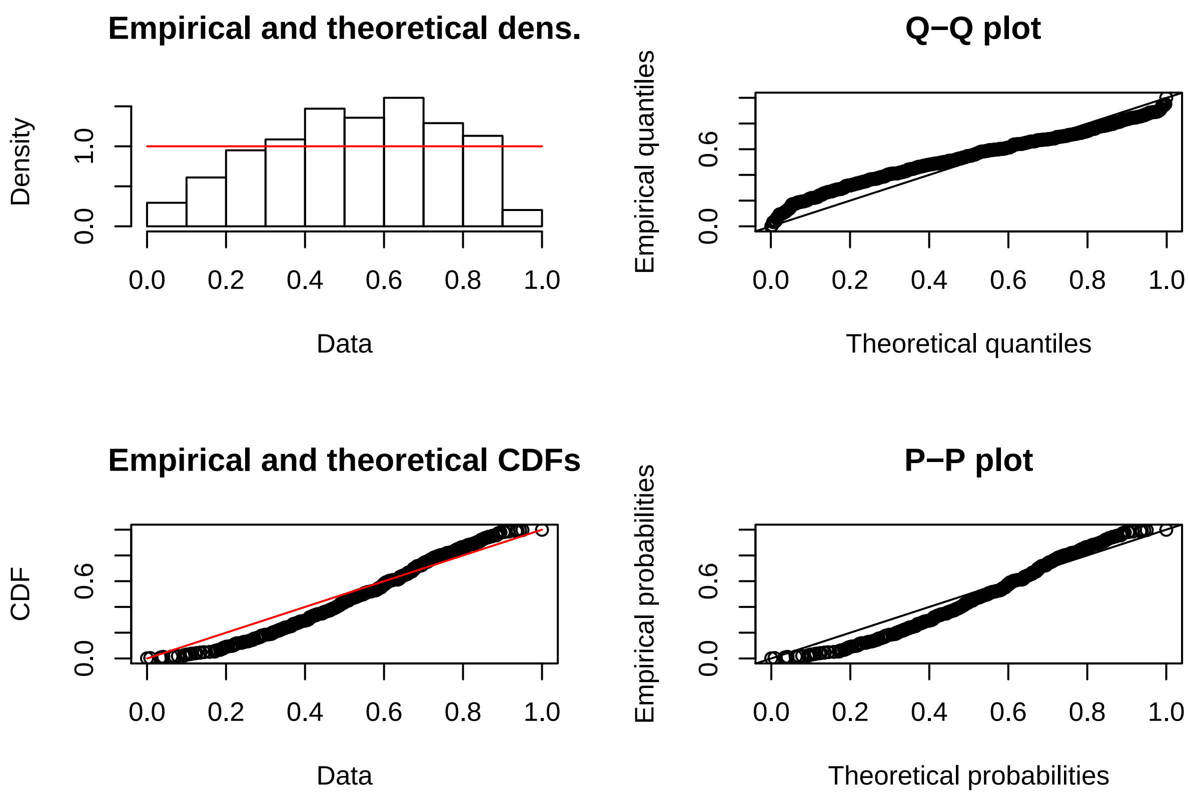
```
#fit beta distribution
yield_b<-(yield-min(yield))/max((yield-min(yield)))
fityield_beta<-fitdist(yield_b,"beta", method="mge")
```

```
## Warning in fitdist(yield_b, "beta", method = "mge"): maximum GOF estimation
## has a default 'gof' argument set to 'CvM'
```

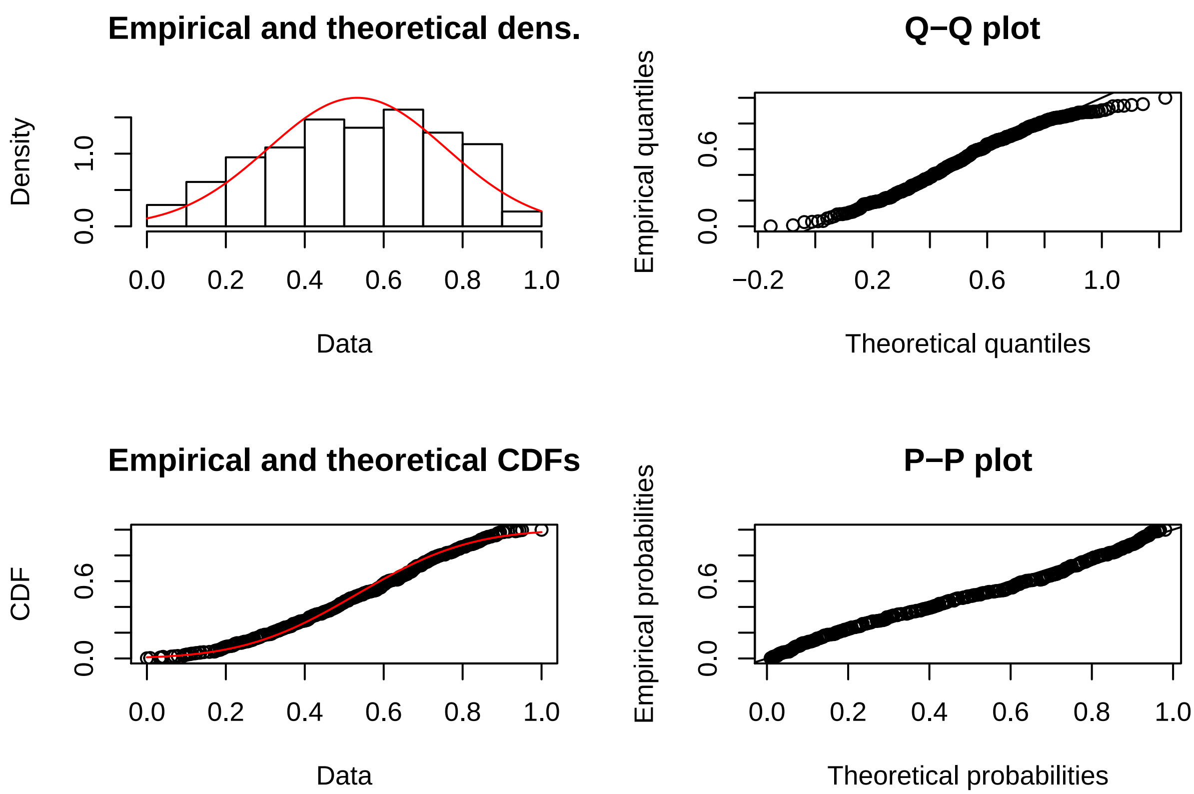
```
plot(fityield_beta)
```



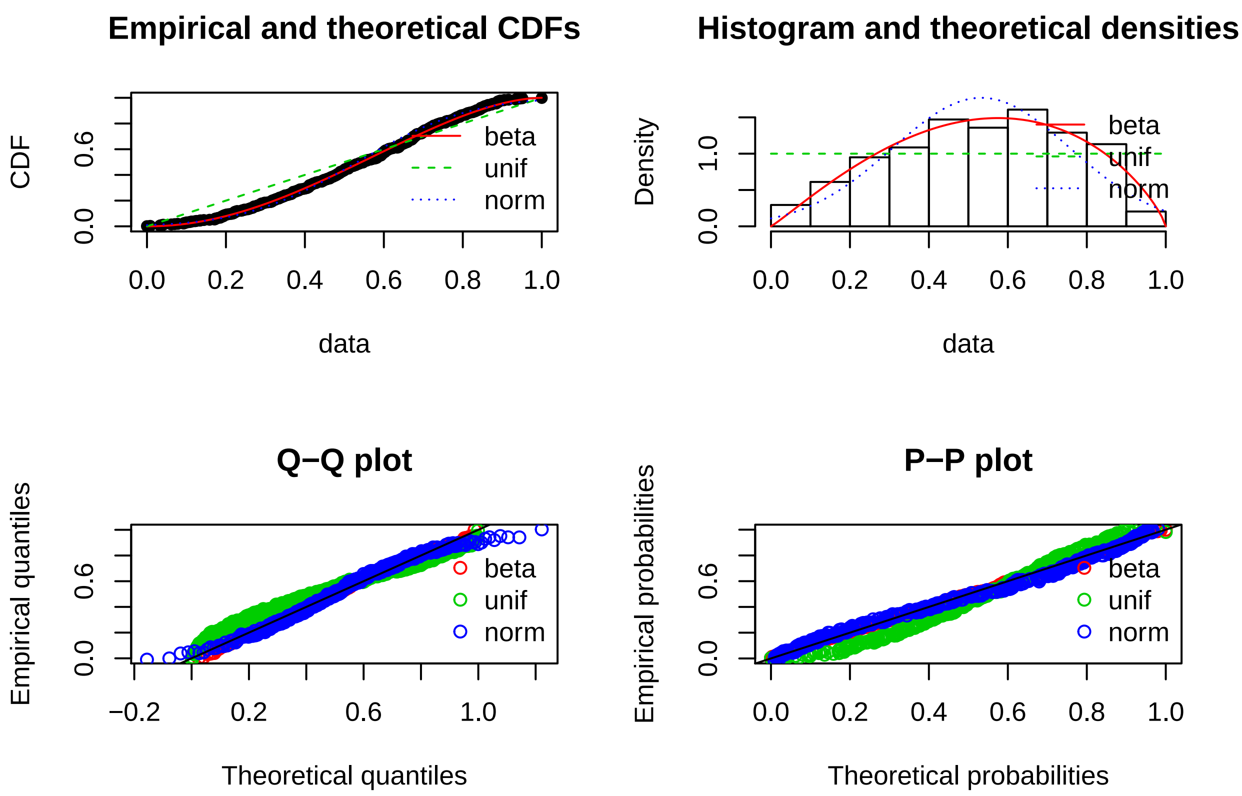

```
#fit uniform distribution
fityield_unif<-fitdist(yield_b,"unif")
plot(fityield_unif)
```




```
##fit normal distribution
fityield_norm<-fitdist(yield_b,"norm")
plot(fityield_norm)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","uniform","normal")
cdfcomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
denscomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
qqcomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
ppcomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
```



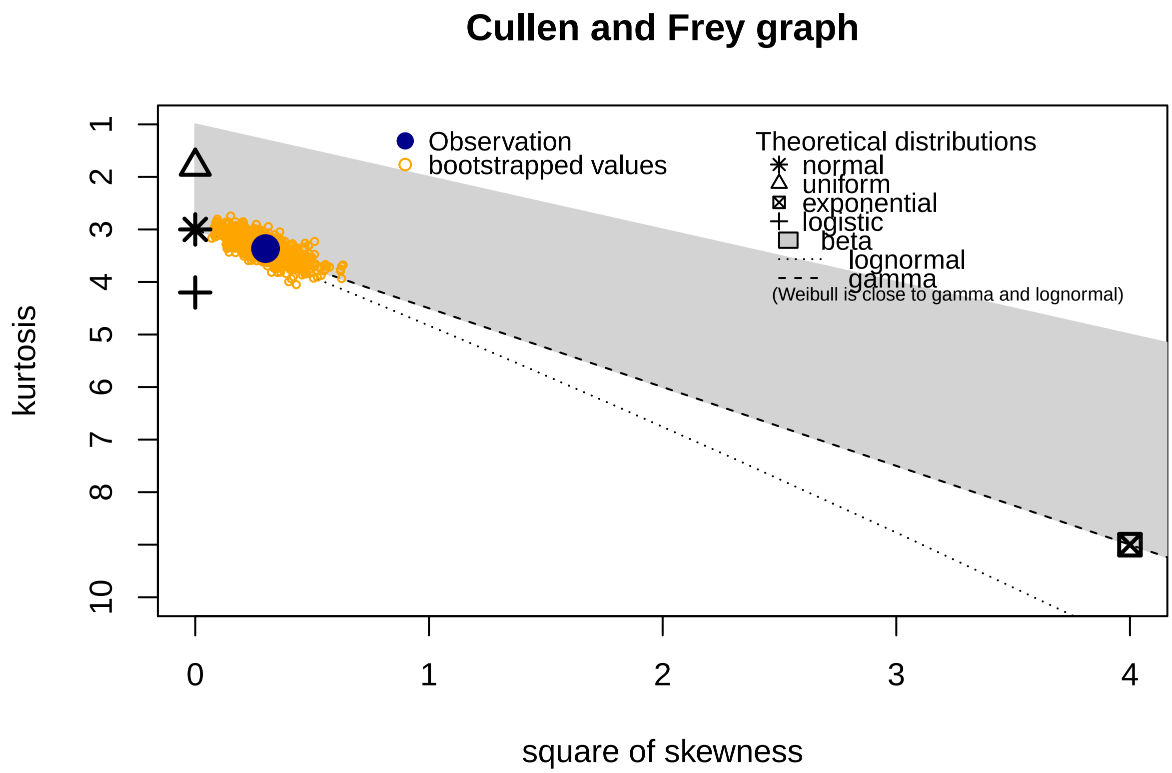

```r
gofstat(list(fityield_beta,fityield_unif,fityield_norm),fitnames=c("beta","unif","norm"))
```

```
## Goodness-of-fit statistics
##                                    beta      unif       norm
## Kolmogorov-Smirnov statistic 0.02711853 0.1249744 0.06392794
## Cramer-von Mises statistic   0.04209458 2.6044176 0.36127436
## Anderson-Darling statistic          Inf       Inf 2.39844648
##
## Goodness-of-fit criteria
##                                beta unif      norm
## Akaike's Information Criterion  Inf   NA -58.29172
## Bayesian Information Criterion  Inf   NA -50.10910
```

From the ftiing grapghs of distribution and "R Console", we can conclude that Yield Spread fits well with the beta distribution.

Three month treasury rate

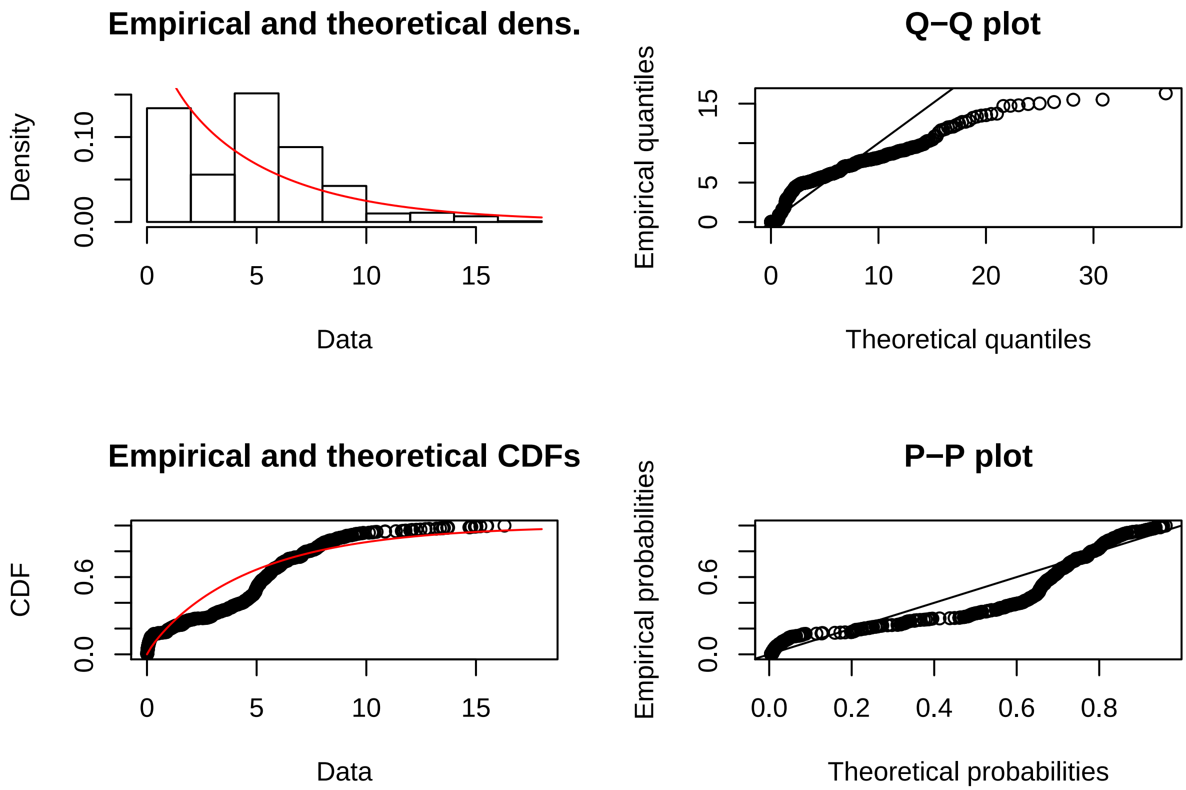
```r
descdist(rate3,boot = 1001)
```



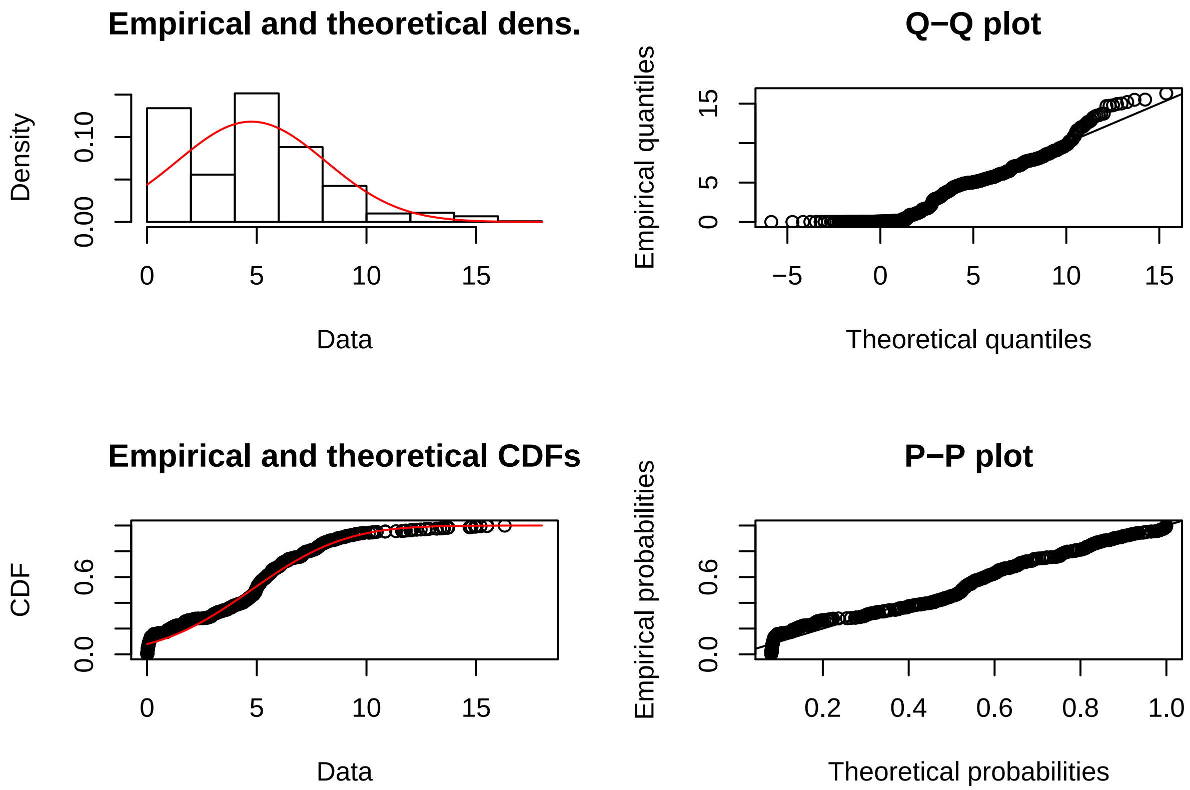

```
## summary statistics
## ------
## min:  0.01   max:  16.3
## median:  4.96
## mean:  4.754526
## estimated sd:  3.380753
## estimated skewness:  0.5484265
## estimated kurtosis:  3.364228
```

This variable may fit gamma, normal and lognormal distribution. So we choose these three distributions to have a further test.

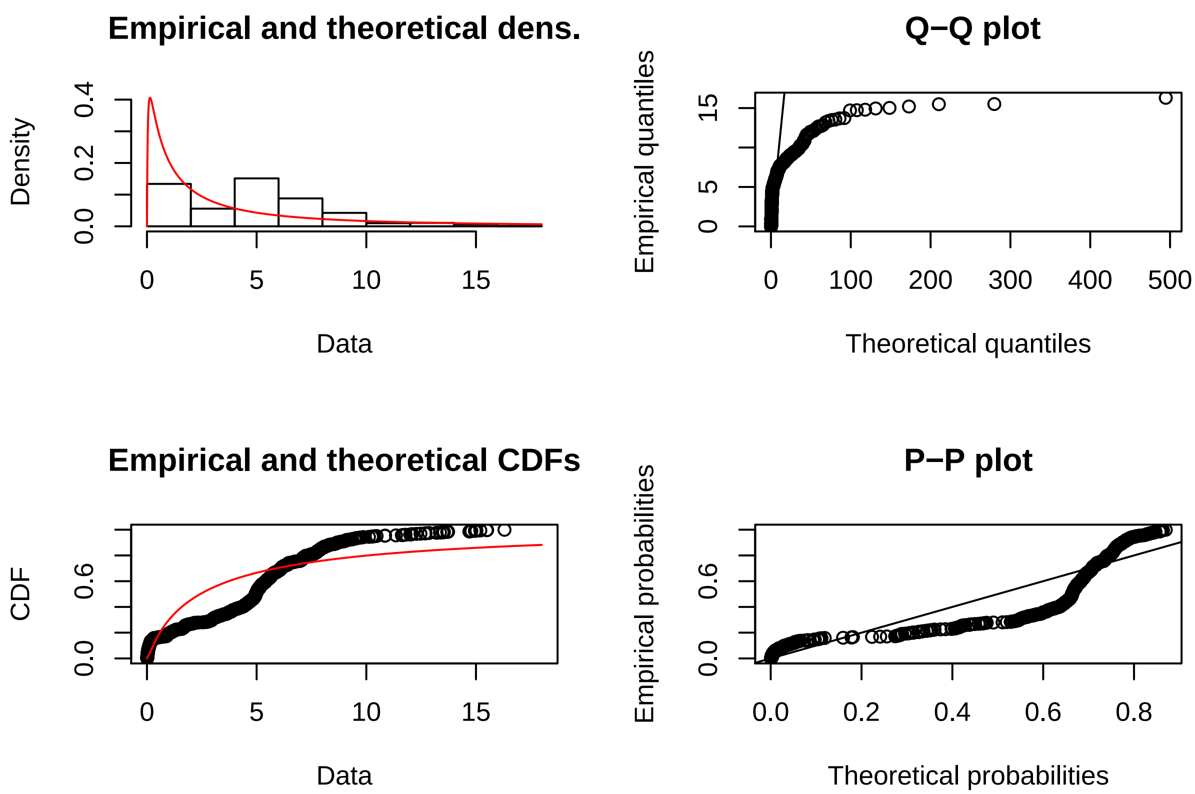
```
##fit gamma distribution
fitrate3_gamma<-fitdist(rate3,"gamma")
plot(fitrate3_gamma)
```



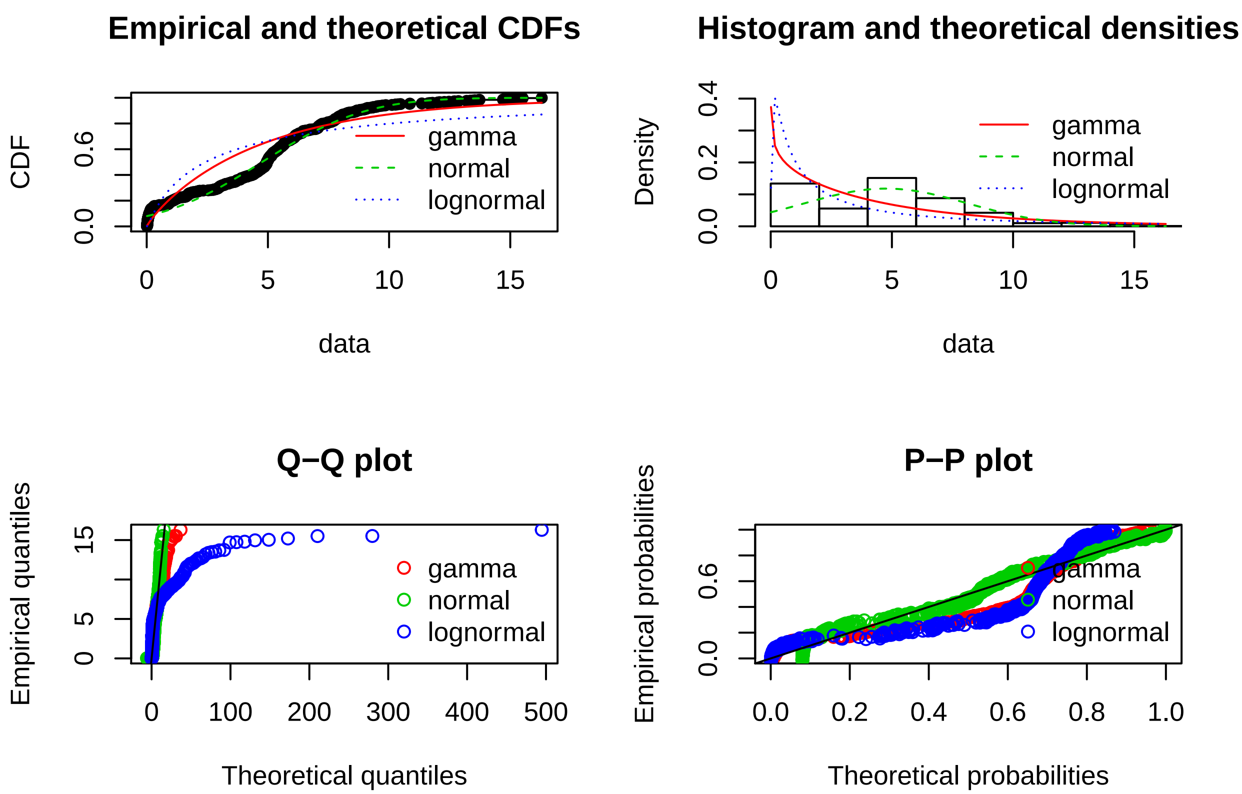

```
##fit normal
fitrate3_norm<-fitdist(rate3,"norm")
plot(fitrate3_norm)
```




```r
##fit lnorm
fitrate3_lnorm<-fitdist(rate3,"lnorm")
plot(fitrate3_lnorm)
```




```
par(mfrow=c(2,2))
plot.legend<-c("gamma","normal","lognormal")
cdfcomp(list(fitrate3_gamma,fitrate3_norm,fitrate3_lnorm),legendtext=c("gamma","normal","lognormal"))
denscomp(list(fitrate3_gamma,fitrate3_norm,fitrate3_lnorm),legendtext=c("gamma","normal","lognormal"))
qqcomp(list(fitrate3_gamma,fitrate3_norm,fitrate3_lnorm),legendtext=c("gamma","normal","lognormal"))
ppcomp(list(fitrate3_gamma,fitrate3_norm,fitrate3_lnorm),legendtext=c("gamma","normal","lognormal"))
```



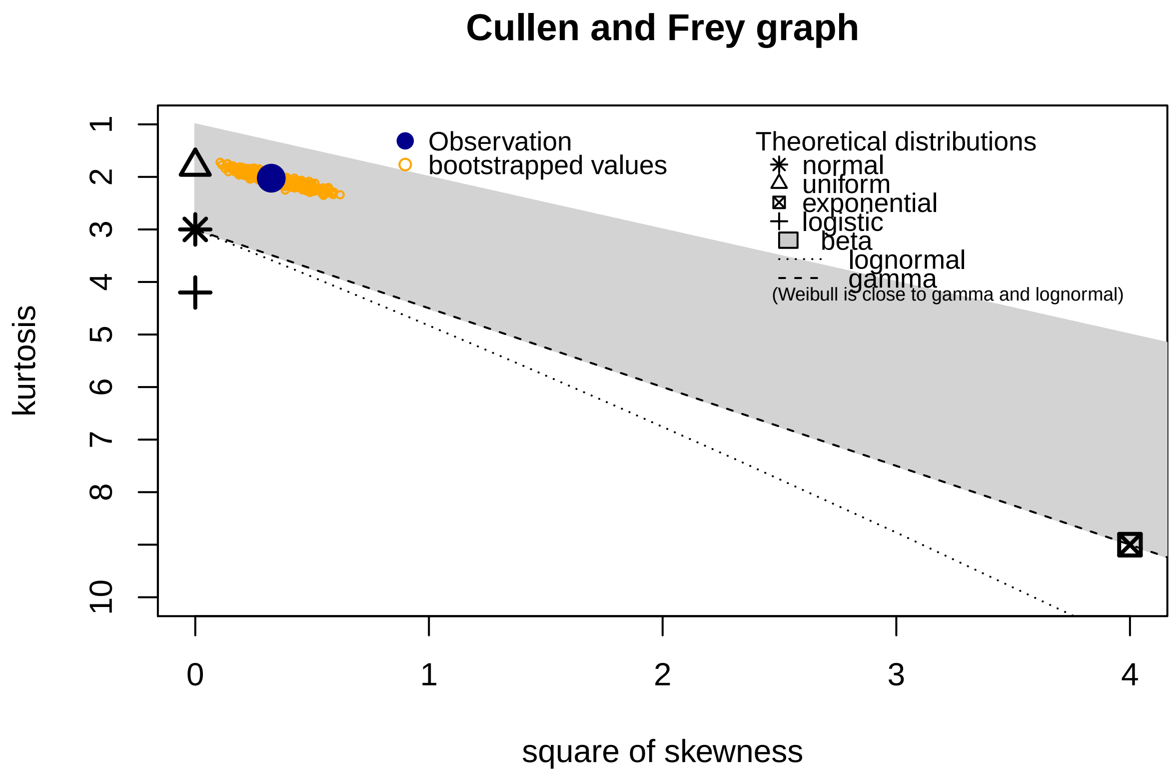

```r
gofstat(list(fitrate3_gamma,fitrate3_norm,fitrate3_lnorm),fitnames=c("gamma","normal","lognormal"))
```

```
## Goodness-of-fit statistics
##                                   gamma      normal   lognormal
## Kolmogorov-Smirnov statistic  0.2116631 0.08007573  0.2497716
## Cramer-von Mises statistic    6.8795784 0.80540628 11.2034572
## Anderson-Darling statistic   35.6438824 6.90134003 59.7348432
##
## Goodness-of-fit criteria
##                                   gamma   normal lognormal
## Akaike's Information Criterion 3072.473 3172.718  3407.262
## Bayesian Information Criterion 3081.270 3181.515  3416.059
```

From the graphs, we know that three month treasury rate fits normal distribution well.

Japanese Central Bank's interest rates

```r
descdist(Japanrate,boot = 1001)
```



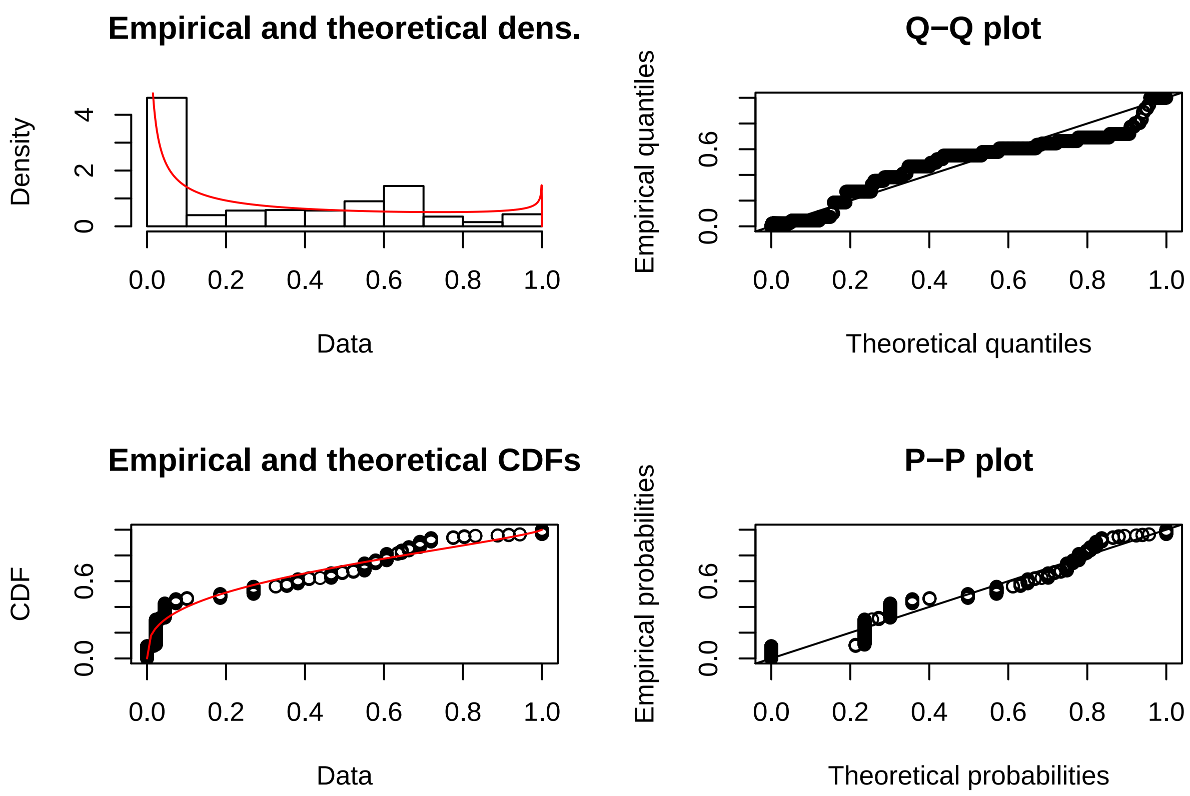

```
## summary statistics
## ------
## min:  0.1   max:  9
## median:  1.75
## mean:  2.824759
## estimated sd:  2.697216
## estimated skewness:  0.5702897
## estimated kurtosis:  2.025226
```

This variable may fit beta and uniform distribution. So we choose these distributions to have a further test.

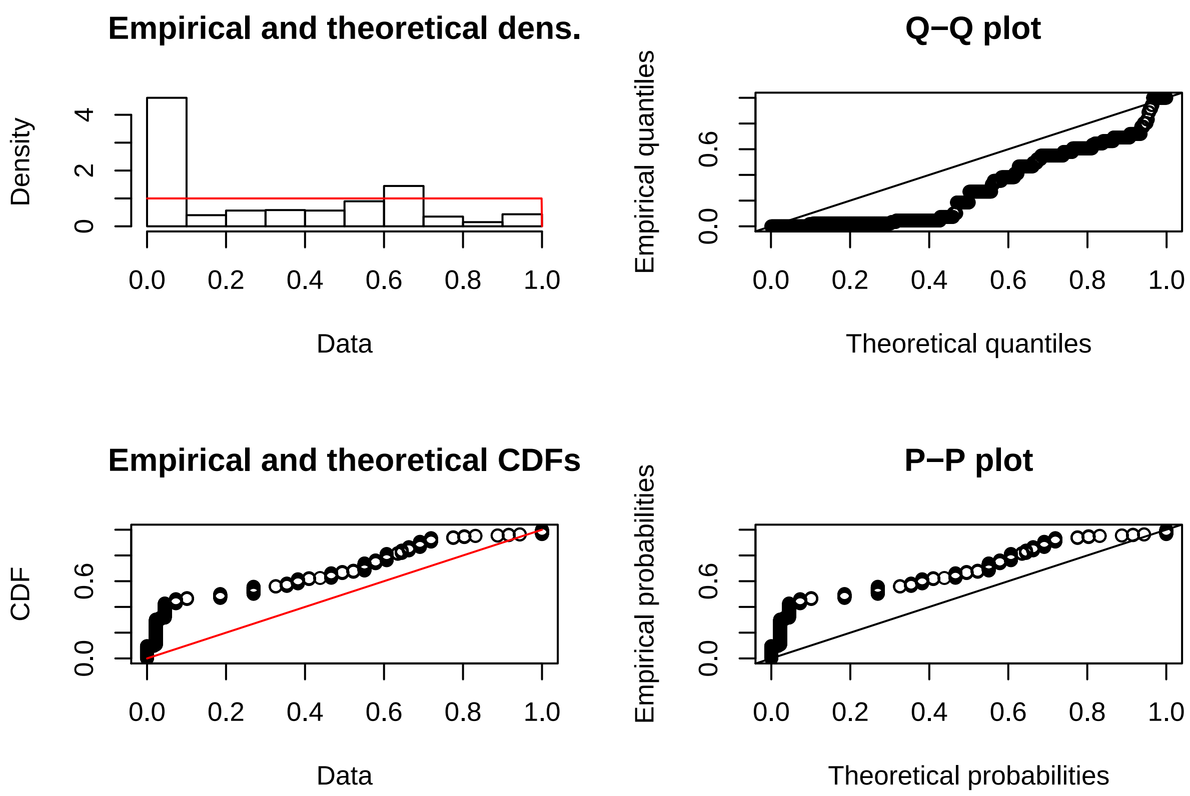
```
##fit beta
Japanrate_b<-(Japanrate-min(Japanrate))/max(Japanrate-min(Japanrate))
fitJapanrate_beta<-fitdist(Japanrate_b,"beta", method = "mge")
```

```
## Warning in fitdist(Japanrate_b, "beta", method = "mge"): maximum GOF
## estimation has a default 'gof' argument set to 'CvM'
```

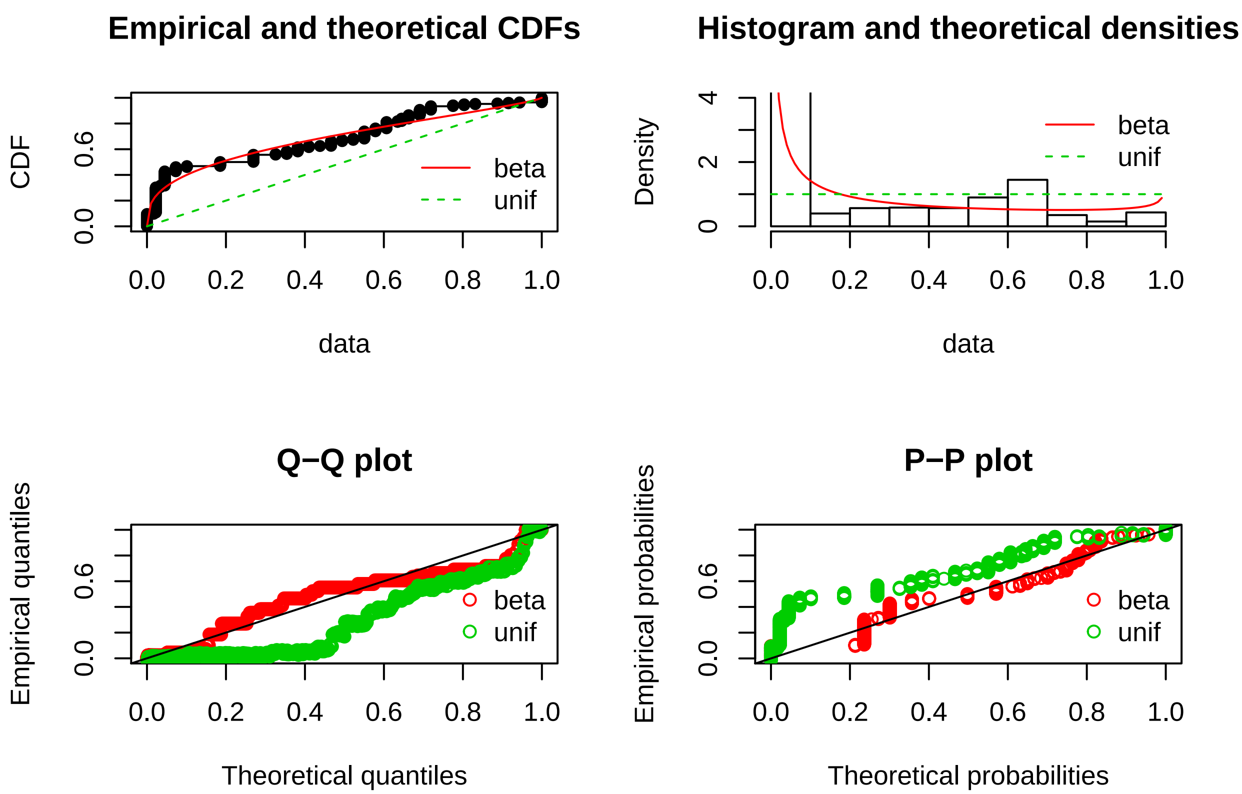
```
plot(fitJapanrate_beta)
```




```
##fit uniform
fitJapanrate_unif<-fitdist(Japanrate_b,"unif")
plot(fitJapanrate_unif)
```




```r
par(mfrow=c(2,2))
plot.legend<-c("beta","unif")
cdfcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
denscomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"),ylim = (0:4))
qqcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
ppcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
```




```
gofstat(list(fitJapanrate_beta,fitJapanrate_unif),fitnames=c("beta","unif"))
```

```
## Goodness-of-fit statistics
##                                   beta       unif
## Kolmogorov-Smirnov statistic 0.1295996  0.3878648
## Cramer-von Mises statistic   2.0443679 28.0328440
## Anderson-Darling statistic         Inf        Inf
##
## Goodness-of-fit criteria
##                                beta unif
## Akaike's Information Criterion -Inf   NA
## Bayesian Information Criterion -Inf   NA
```

Japanese Central Bank's interest rates fits the beta distribution well.

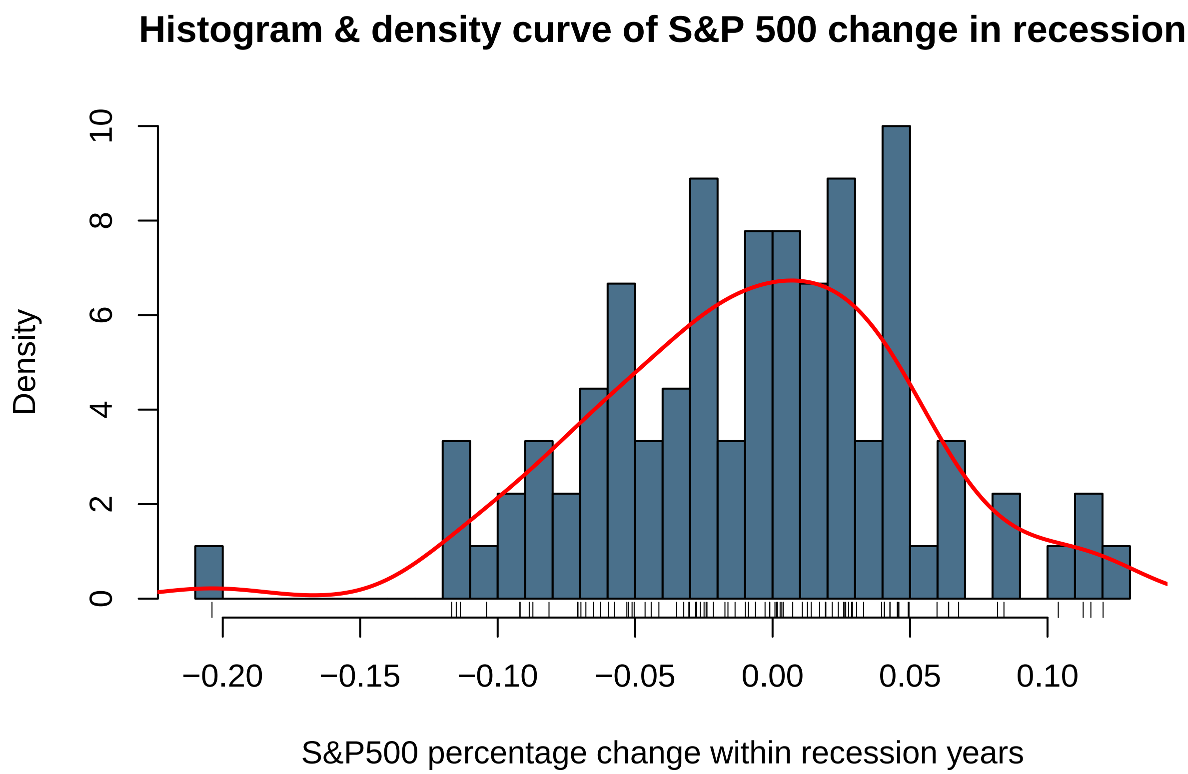
2 c

```
##install.packages("dplyr")
library(dplyr)
```

Percentage changes of S&P500 within recession years

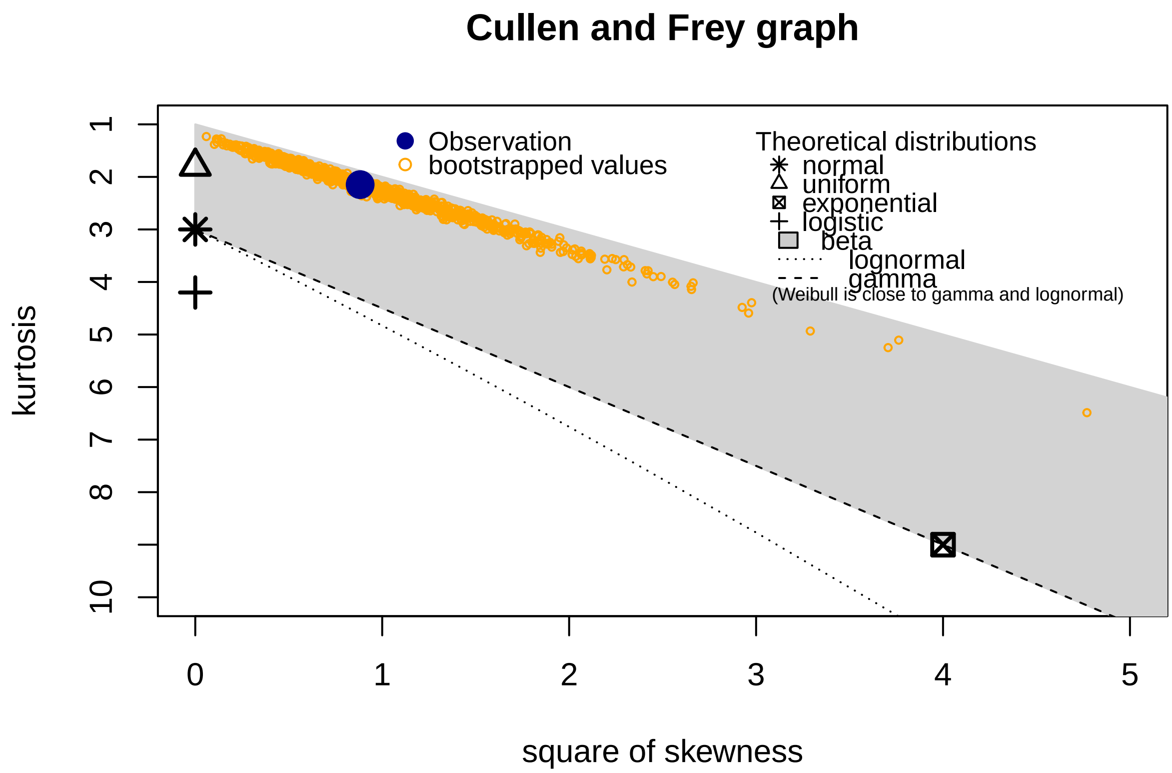
```
recession500 = subset(data_sp500, data_sp500$Date > '1969-11-30' & data_sp500$Date < '1970-12-02')
recession500 = rbind(recession500,subset(data_sp500, data_sp500$Date > '1973-11-01' & data_sp500$Date <
recession500 = rbind(recession500,subset(data_sp500, data_sp500$Date > '1980-01-01' & data_sp500$Date <
recession500 = rbind(recession500,subset(data_sp500, data_sp500$Date > '1981-07-01' & data_sp500$Date <
```

```
recession500 = rbind(recession500,subset(data_sp500, data_sp500$Date > '1990-07-01' & data_sp500$Date <
recession500 = rbind(recession500,subset(data_sp500, data_sp500$Date > '2001-03-01' & data_sp500$Date <
recession500 = rbind(recession500,subset(data_sp500, data_sp500$Date > '2007-12-01' & data_sp500$Date <
##histgram and density curve
hist(recession500$perc,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogram & density curve
lines(density(recession500$perc),col="red",lwd=2)
rug(recession500$perc)
```



Fit distribution

```
descdist(recession500$Value,boot = 1001)
```



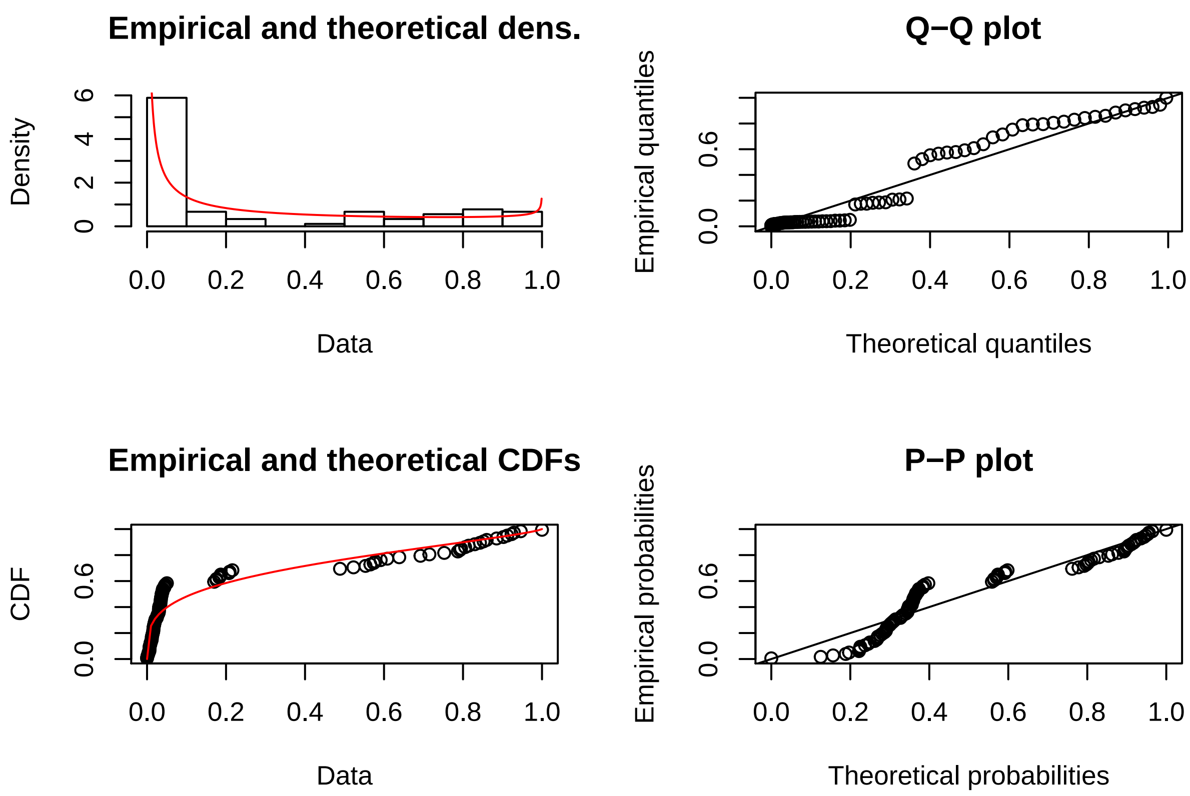

```
## summary statistics
## ------
## min:  67.07   max:  1479.22
## median:  119.8
## mean:  444.5651
## estimated sd:  485.3159
## estimated skewness:  0.939225
## estimated kurtosis:  2.147509
```

From this graph, we can see Percentage change of S&P 500 may fit beta and uniform distribution. So we need further test:

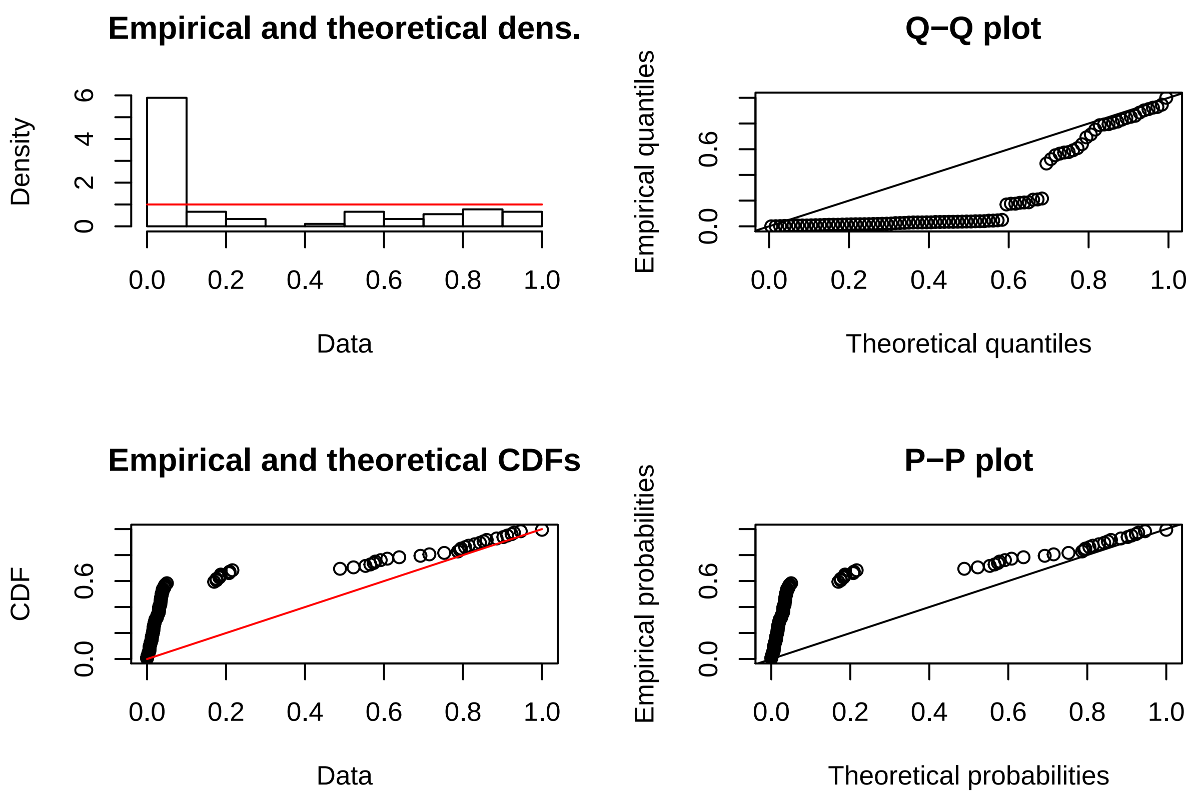
```
#fit beta distribution
recession500$Value_b<-(recession500$Value-min(recession500$Value))/max((recession500$Value-min(recessio
fitsp500_beta<-fitdist(recession500$Value_b,"beta", method="mge")
```

```
## Warning in fitdist(recession500$Value_b, "beta", method = "mge"): maximum
## GOF estimation has a default 'gof' argument set to 'CvM'
```

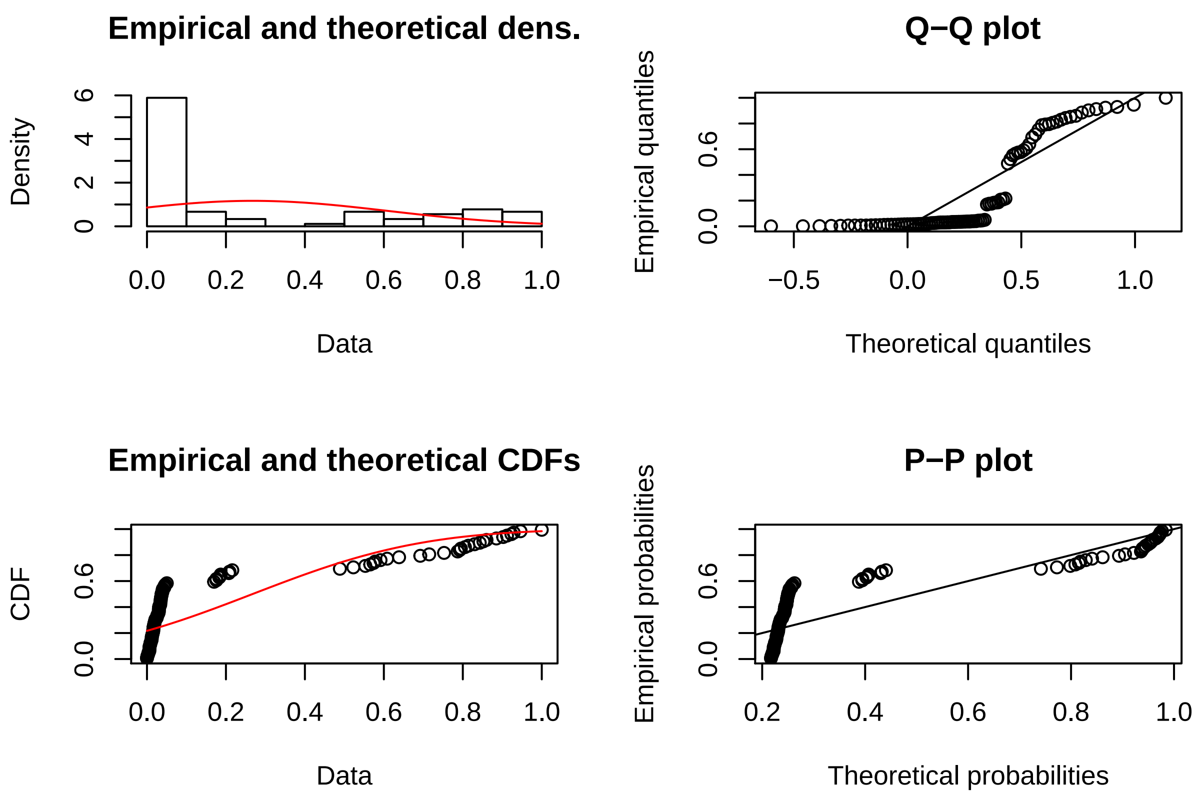
```
plot(fitsp500_beta)
```



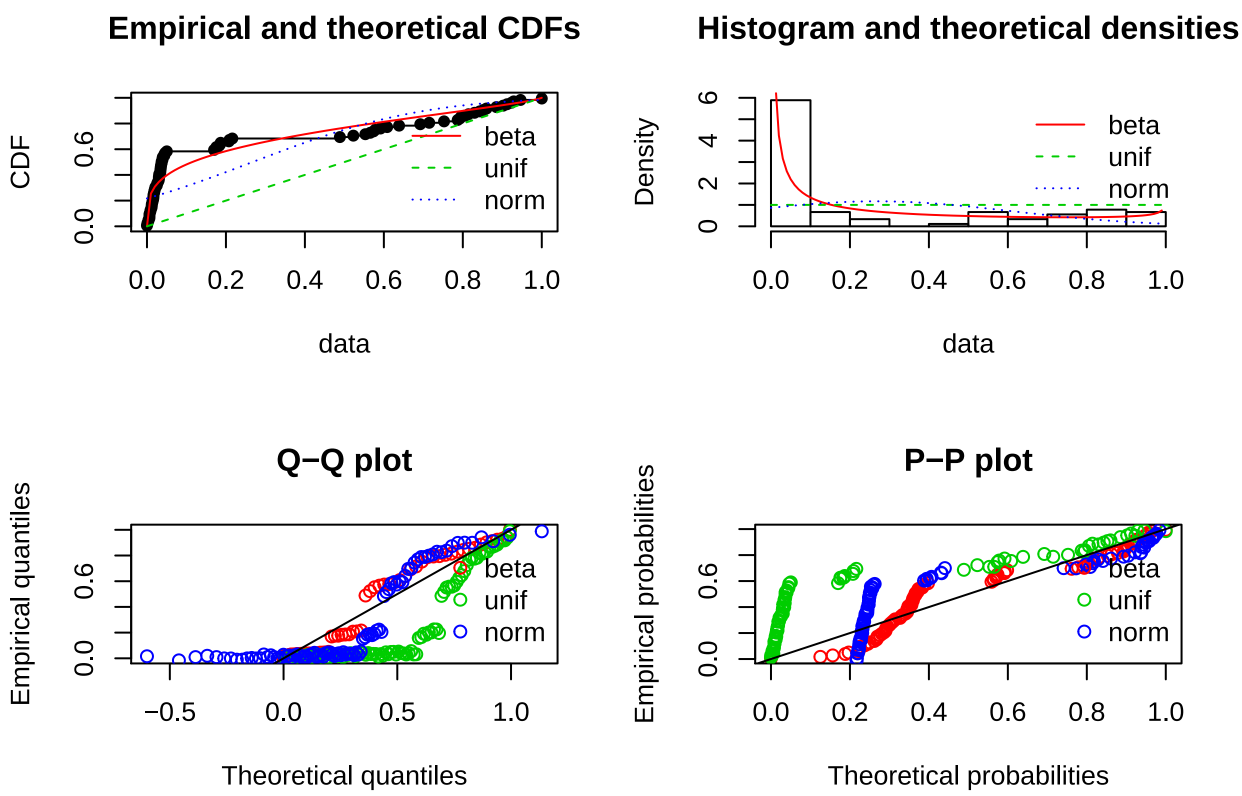

```
#fit uniform
fitsp500_unif<-fitdist(recession500$Value_b,"unif")
plot(fitsp500_unif)
```




```r
#fit normal
fitsp500_norm<-fitdist(recession500$Value_b,"norm")
plot(fitsp500_norm)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","uniform","normal")
cdfcomp(list(fitsp500_beta,fitsp500_unif,fitsp500_norm),legendtext=c("beta","unif","norm"))
denscomp(list(fitsp500_beta,fitsp500_unif,fitsp500_norm),legendtext=c("beta","unif","norm"),ylim = (0:6
qqcomp(list(fitsp500_beta,fitsp500_unif,fitsp500_norm),legendtext=c("beta","unif","norm"))
ppcomp(list(fitsp500_beta,fitsp500_unif,fitsp500_norm),legendtext=c("beta","unif","norm"))
```




```r
gofstat(list(fitsp500_beta,fitsp500_unif,fitsp500_norm),fitnames=c("beta","unif","norm"))
```

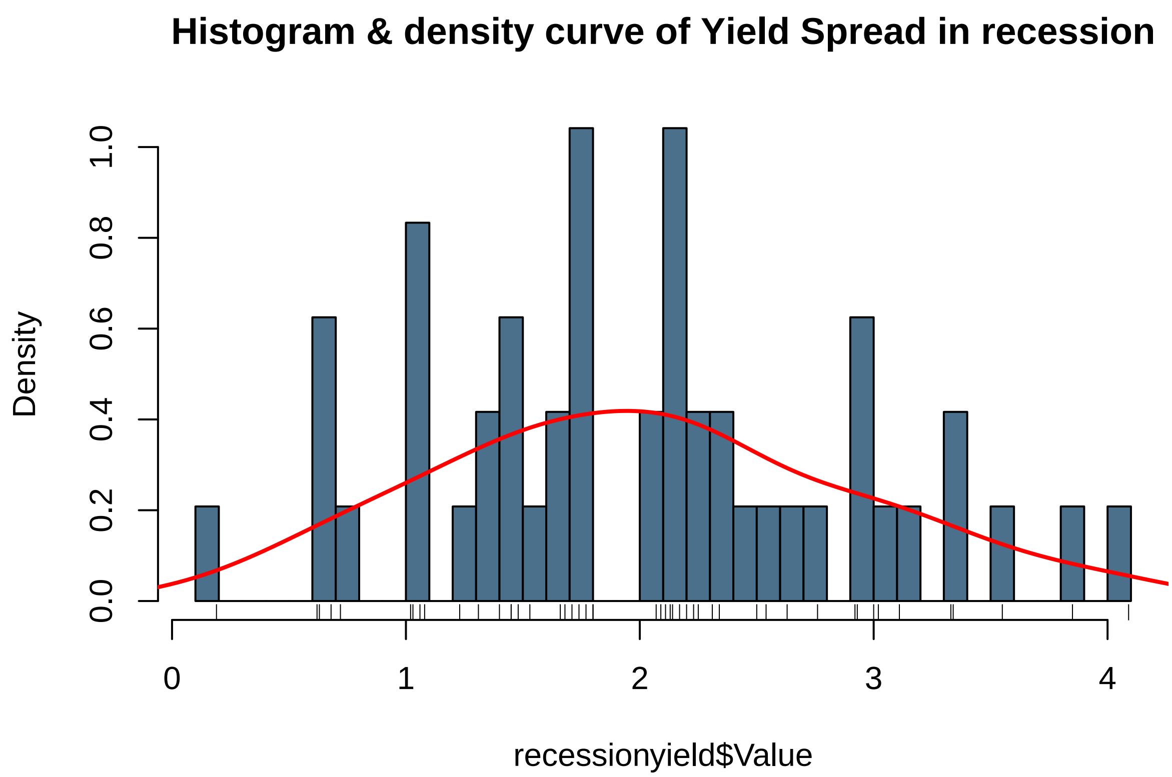
```
## Goodness-of-fit statistics
##                                   beta      unif      norm
## Kolmogorov-Smirnov statistic 0.1912236 0.5385897  0.326179
## Cramer-von Mises statistic   0.6991297 7.3537319  2.099206
## Anderson-Darling statistic         Inf       Inf 11.297950
##
## Goodness-of-fit criteria
##                                beta unif     norm
## Akaike's Information Criterion -Inf   NA 66.15099
## Bayesian Information Criterion -Inf   NA 71.15061
```

From the graphs, we can see that the percentage changes of S&P500 within recession years fits beta distribution well.

Yield Spread within recession years

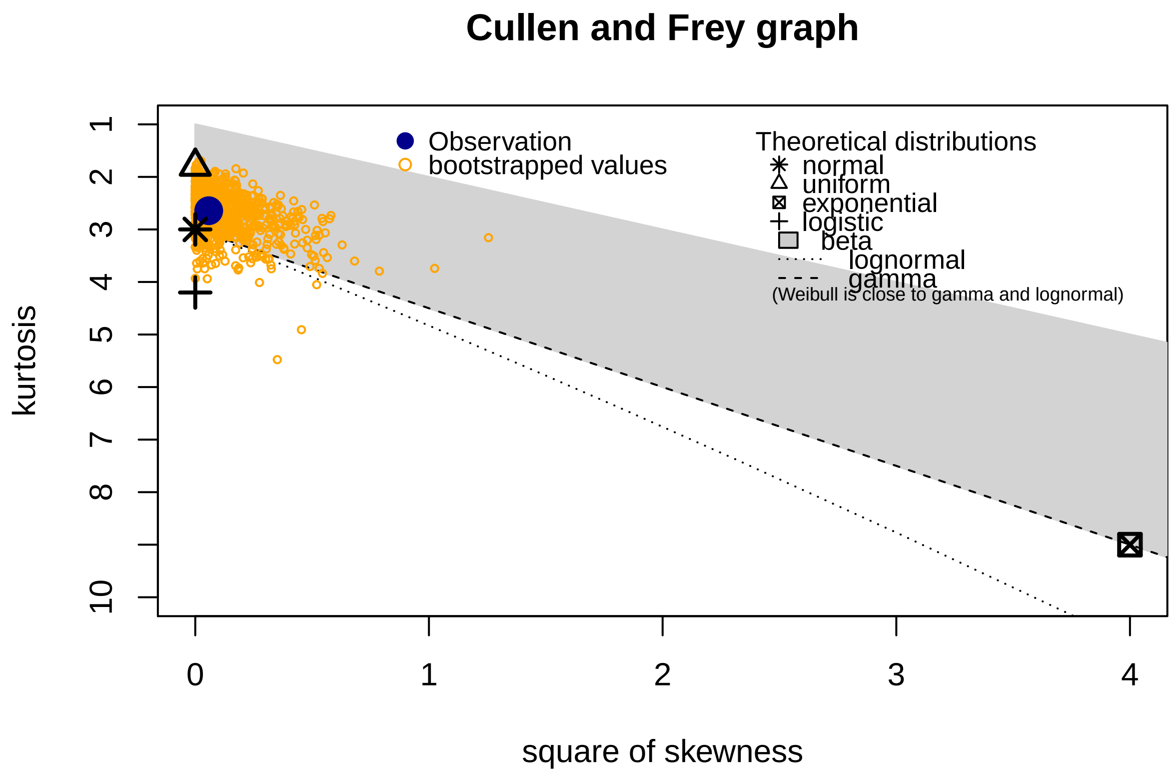
```r
recessionyield = subset (data_yield,data_yield$Date > '1969-11-30' & data_yield$Date < '1970-12-02')
recessionyield = rbind(recessionyield,subset(data_yield, data_yield$Date > '1973-11-01' & data_yield$Da
recessionyield = rbind(recessionyield,subset(data_yield, data_yield$Date > '1980-01-01' & data_yield$Da
recessionyield = rbind(recessionyield,subset(data_yield, data_yield$Date > '1981-07-01' & data_yield$Da
recessionyield = rbind(recessionyield,subset(data_yield, data_yield$Date > '1990-07-01' & data_yield$Da
recessionyield = rbind(recessionyield,subset(data_yield, data_yield$Date > '2001-03-01' & data_yield$Da
recessionyield = rbind(recessionyield,subset(data_yield, data_yield$Date > '2007-12-01' & data_yield$Da
##histgram
```

```
hist(recessionyield$Value,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogram & density cu
lines(density(recessionyield$Value),col="red",lwd=2)
rug(recessionyield$Value)
```



Fit distribution

```
descdist(recessionyield$Value,boot = 1001)
```



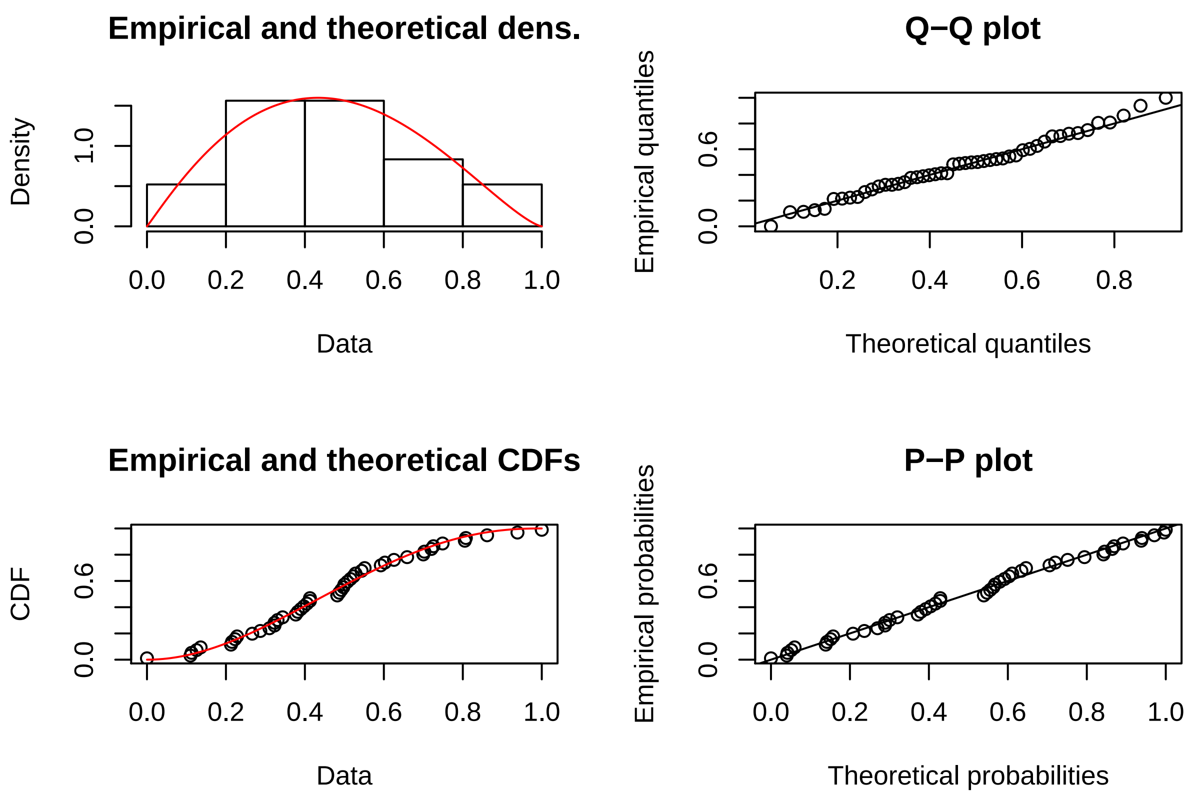

```
## summary statistics
## ------
## min:  0.19   max:  4.09
## median:  2.08
## mean:  2.013542
## estimated sd:  0.8968467
## estimated skewness:  0.2400121
## estimated kurtosis:  2.64421
```

From this graph, we assume that Yield Spread within recession years may fit beta,uniform and normal distribution. Here we have some further test:

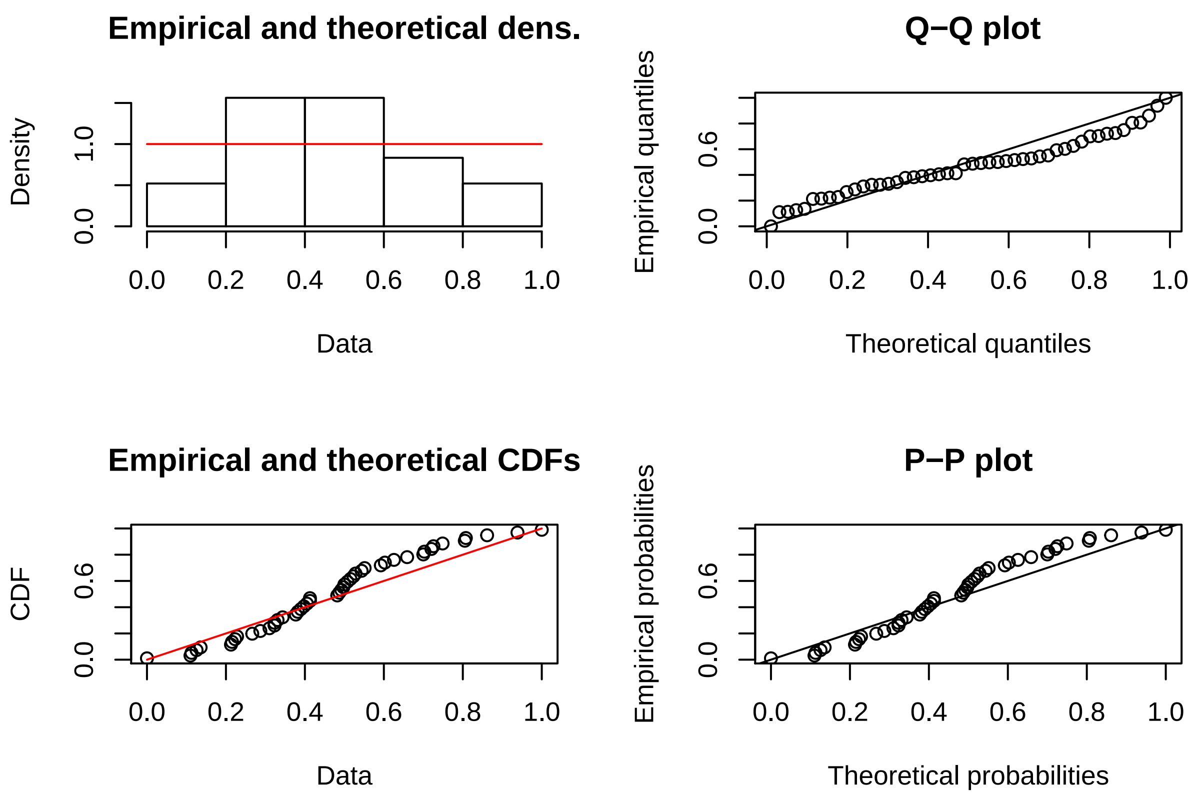
```
#fit beta distribution
recessionyield$Value_b<-(recessionyield$Value-min(recessionyield$Value))/max((recessionyield$Value-min(r
fityield_beta<-fitdist(recessionyield$Value_b,"beta", method="mge")
```

```
## Warning in fitdist(recessionyield$Value_b, "beta", method = "mge"): maximum
## GOF estimation has a default 'gof' argument set to 'CvM'
```

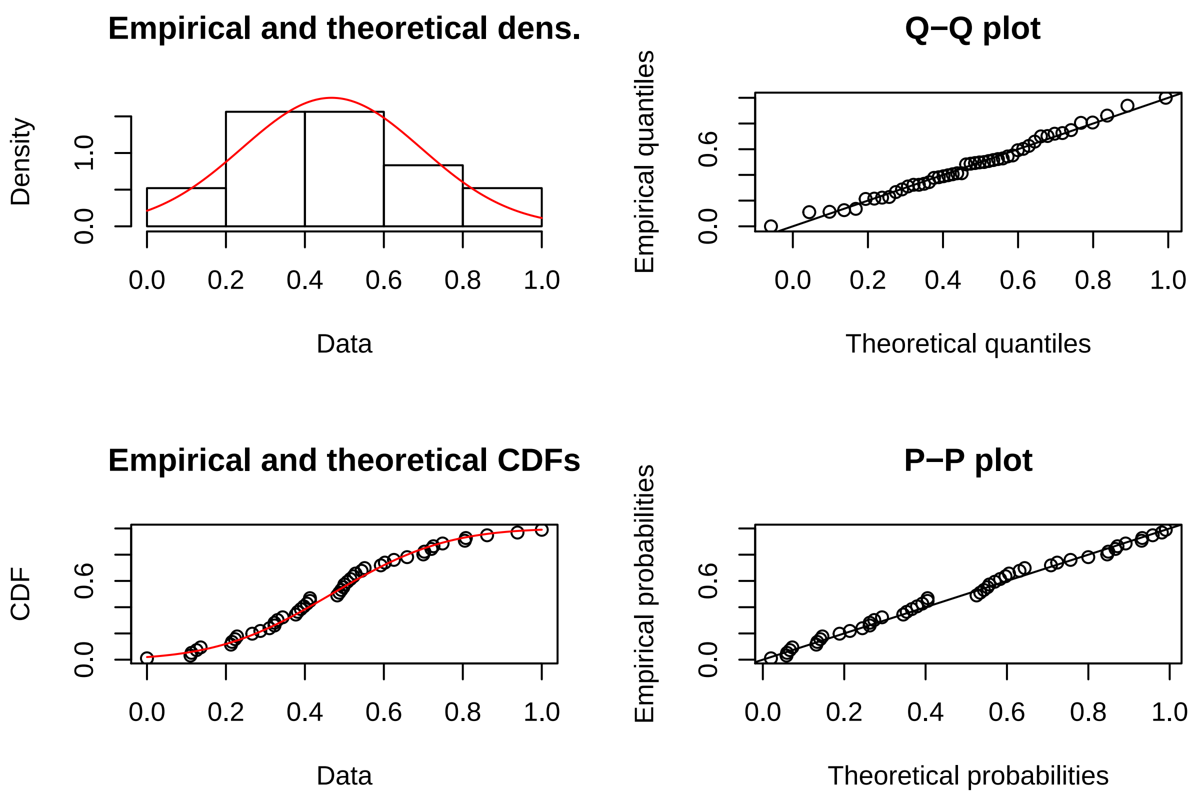
```
plot(fityield_beta)
```



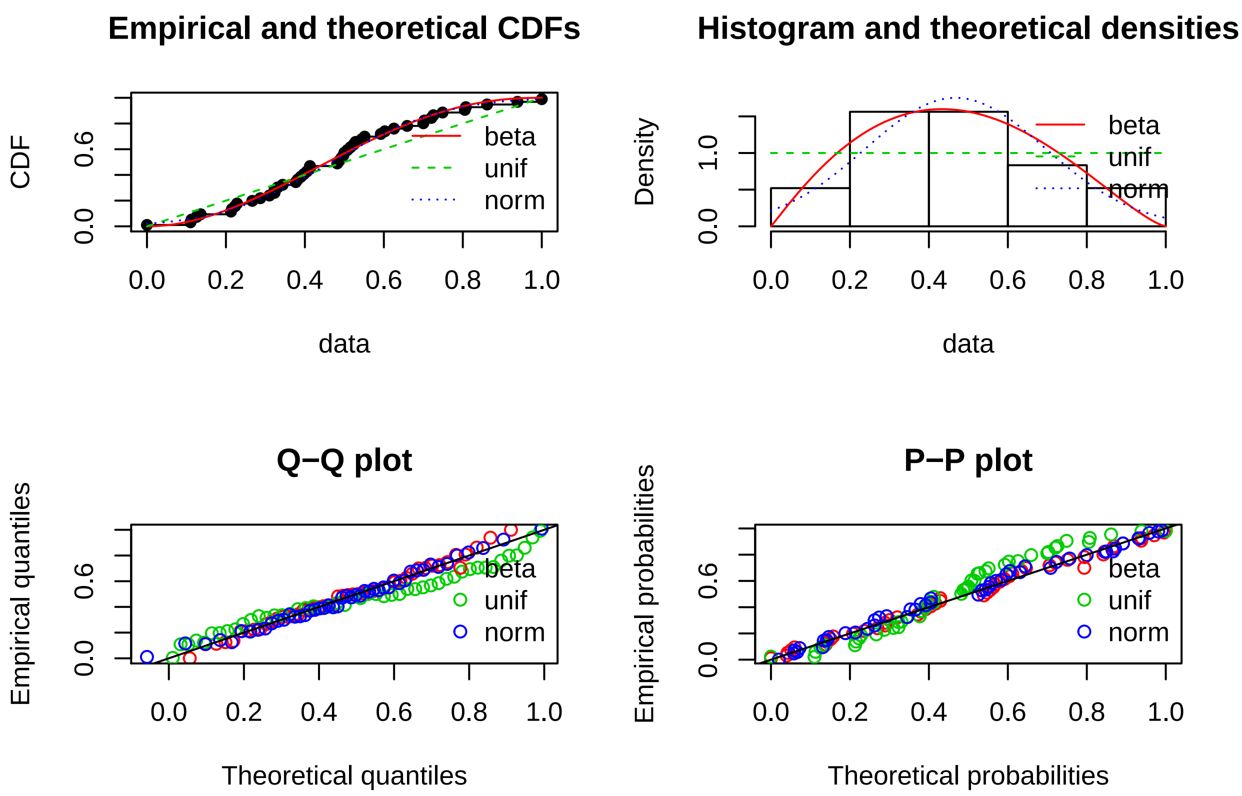

```r
#fit uniform distribution
fityield_unif<-fitdist(recessionyield$Value_b,"unif")
plot(fityield_unif)
```




```r
##fit normal distribution
fityield_norm<-fitdist(recessionyield$Value_b,"norm")
plot(fityield_norm)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","uniform","normal")
cdfcomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
denscomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
qqcomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
ppcomp(list(fityield_beta,fityield_unif,fityield_norm),legendtext=c("beta","unif","norm"))
```




```
gofstat(list(fityield_beta,fityield_unif,fityield_norm),fitnames=c("beta","unif","norm"))
```

```
## Goodness-of-fit statistics
##                                    beta      unif       norm
## Kolmogorov-Smirnov statistic 0.06237089 0.1570513 0.07424278
## Cramer-von Mises statistic   0.02880751 0.3420603 0.03309901
## Anderson-Darling statistic          Inf       Inf 0.20998069
##
## Goodness-of-fit criteria
##                                beta unif      norm
## Akaike's Information Criterion  Inf   NA -1.897766
## Bayesian Information Criterion  Inf   NA  1.844636
```

From the fitting grapghs of distribution and "R Console", we can conclude that Yield Spread in recession years fits well with the beta distribution.

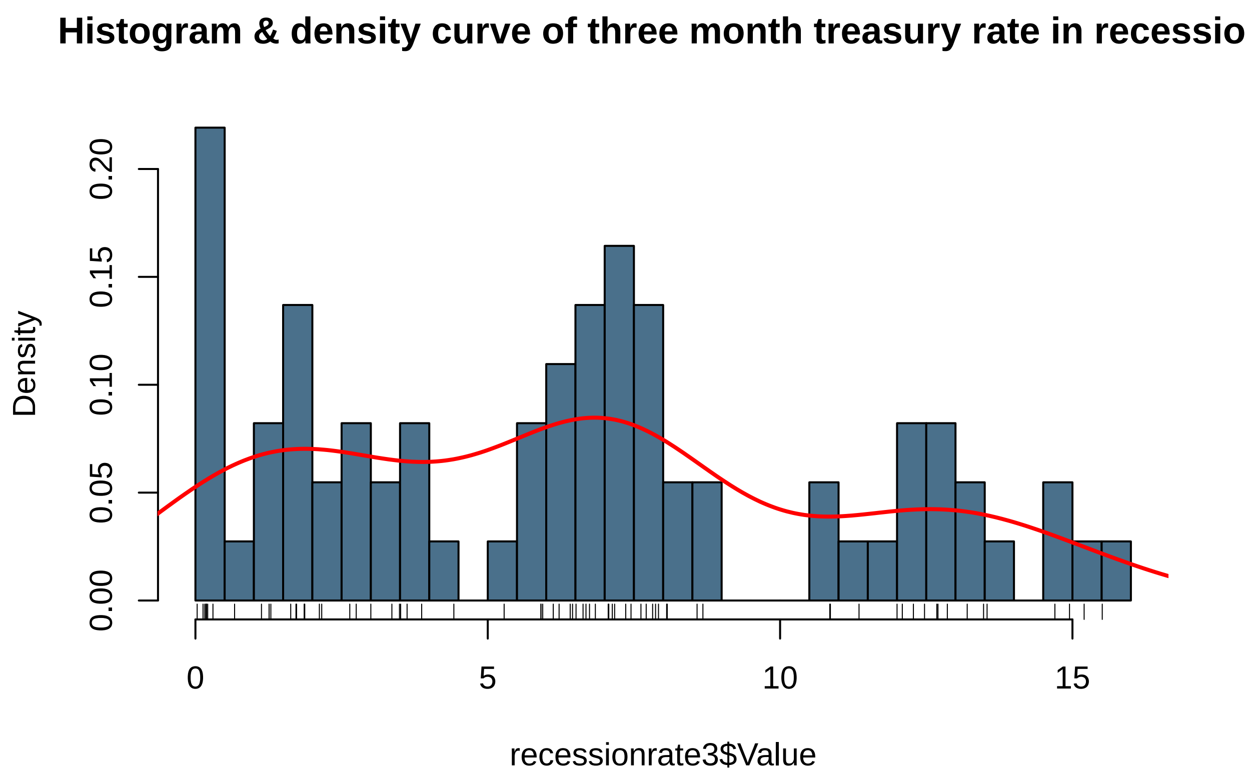
Three month treasury rate in recession years

```
recessionrate3 = subset(data_rate3, data_rate3$Date > '1969-11-30' & data_rate3$Date < '1970-12-02')
recessionrate3 = rbind(recessionrate3,subset(data_rate3, data_rate3$Date > '1973-11-01' & data_yield$Da
```

```
## Warning in data_rate3$Date > "1973-11-01" & data_yield$Date < "1975-04-02":
## longer object length is not a multiple of shorter object length
```

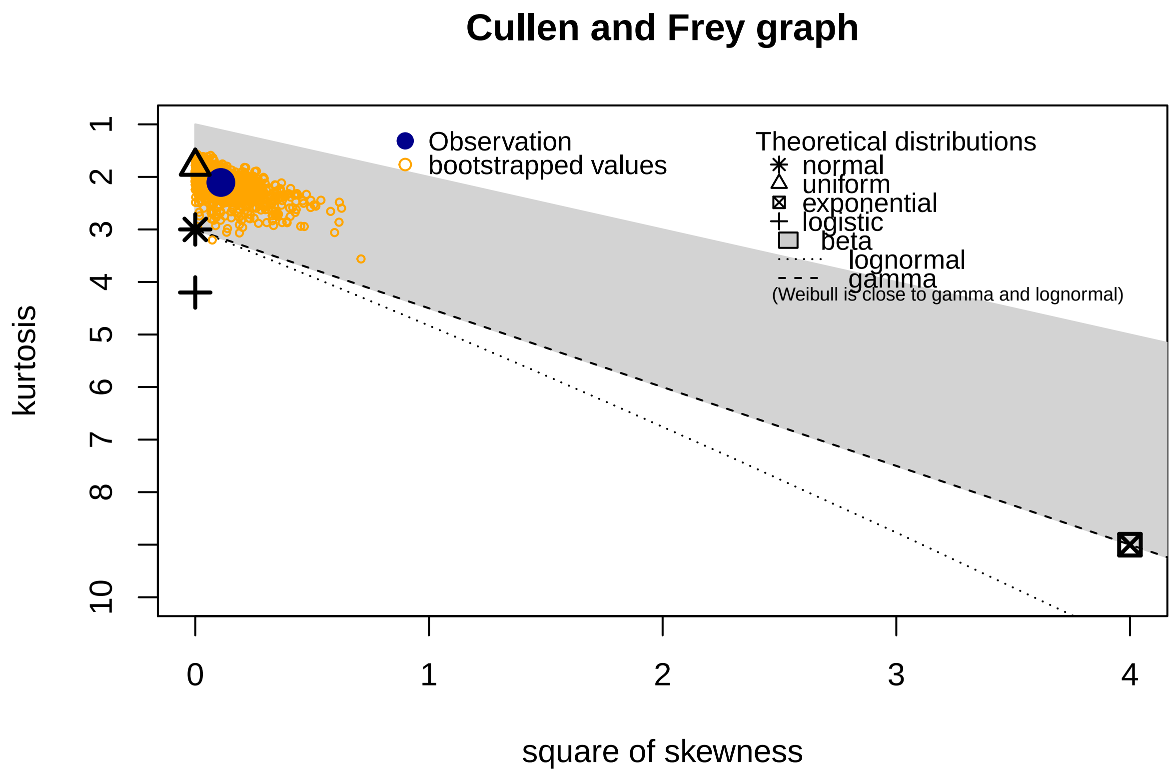
```
recessionrate3 = rbind(recessionrate3,subset(data_rate3, data_rate3$Date > '1980-01-01' & data_rate3$Da
recessionrate3 = rbind(recessionrate3,subset(data_rate3, data_rate3$Date > '1981-07-01' & data_rate3$Da
recessionrate3 = rbind(recessionrate3,subset(data_rate3, data_rate3$Date > '1990-07-01' & data_rate3$Da
```

```
recessionrate3 = rbind(recessionrate3,subset(data_rate3, data_rate3$Date > '2001-03-01' & data_rate3$Da
recessionrate3 = rbind(recessionrate3,subset(data_rate3, data_rate3$Date > '2007-12-01' & data_rate3$Da
#hist
hist(recessionrate3$Value,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogram & density cu
lines(density(recessionrate3$Value),col="red",lwd=2)
rug(recessionrate3$Value)
```



fit distribution

```
descdist(recessionrate3$Value,boot = 1001)
```



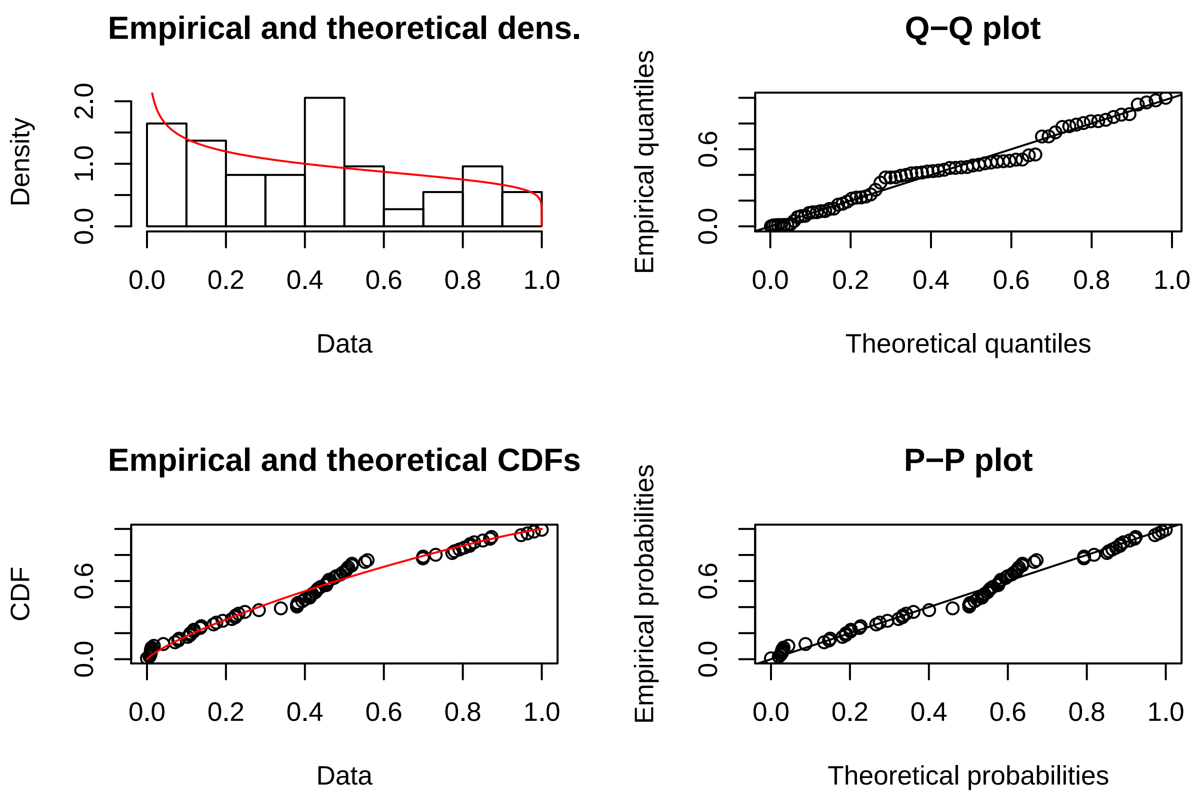

```
## summary statistics
## ------
## min:  0.03   max:  15.51
## median:  6.51
## mean:  6.402877
## estimated sd:  4.490998
## estimated skewness:  0.3311099
## estimated kurtosis:  2.106218
```

This variable may fit beta nornal and uniform distribution. So we choose these three distributions to have a further test.

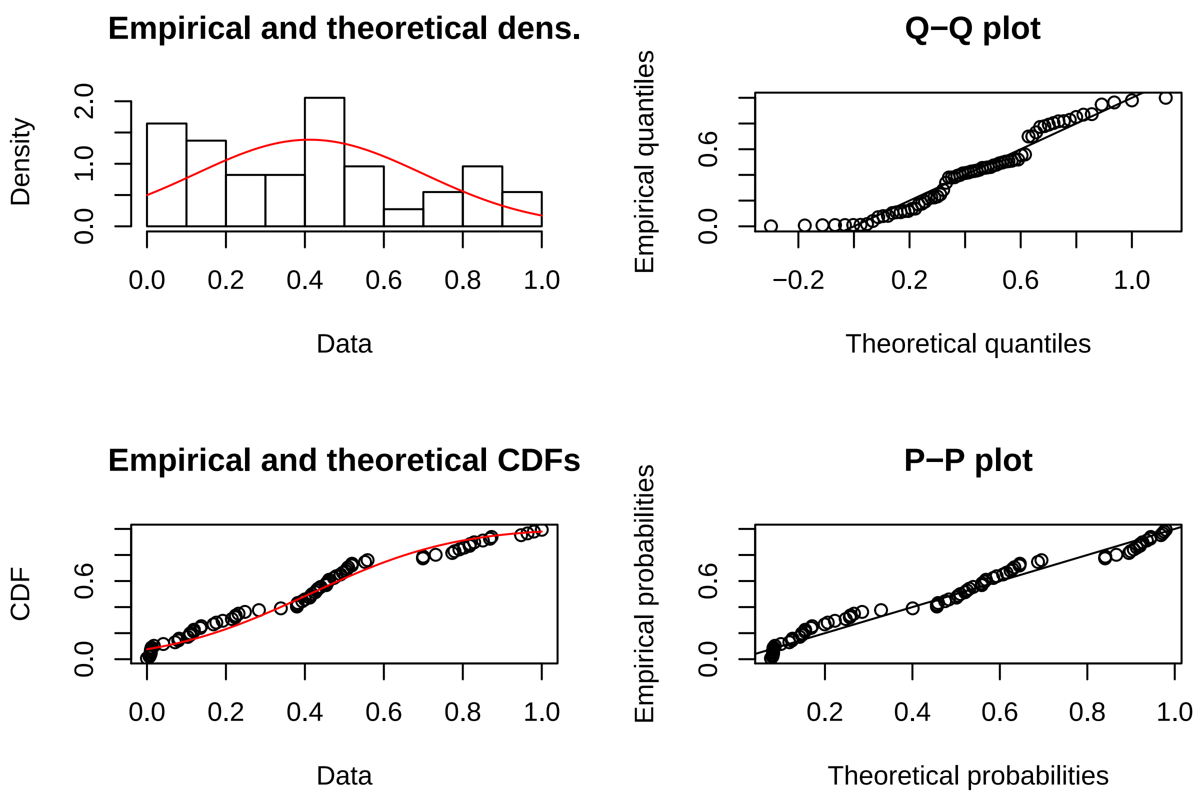
```
##fit beta distribution
recessionrate3$Value_b<-(recessionrate3$Value-min(recessionrate3$Value))/max(recessionrate3$Value-min(re
fitrate3_beta<-fitdist(recessionrate3$Value_b,"beta", method="mge")
```

```
## Warning in fitdist(recessionrate3$Value_b, "beta", method = "mge"): maximum
## GOF estimation has a default 'gof' argument set to 'CvM'
```

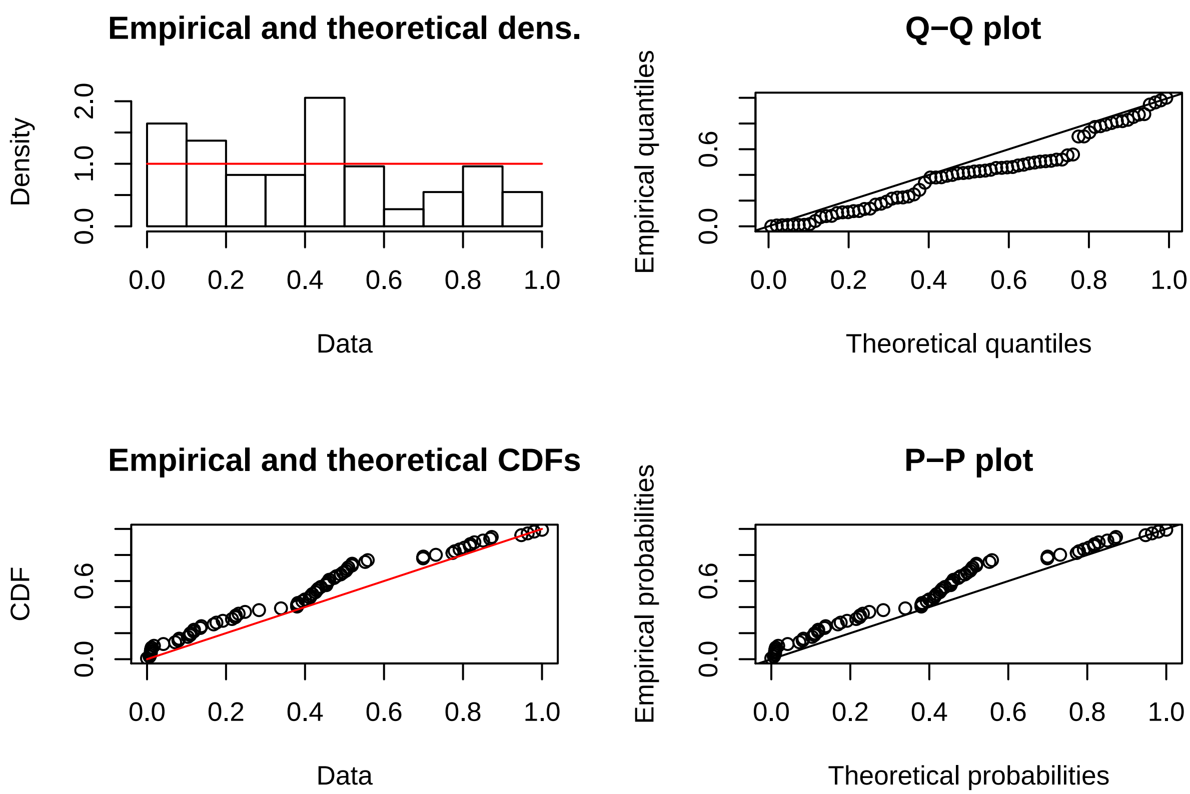
```
plot(fitrate3_beta)
```



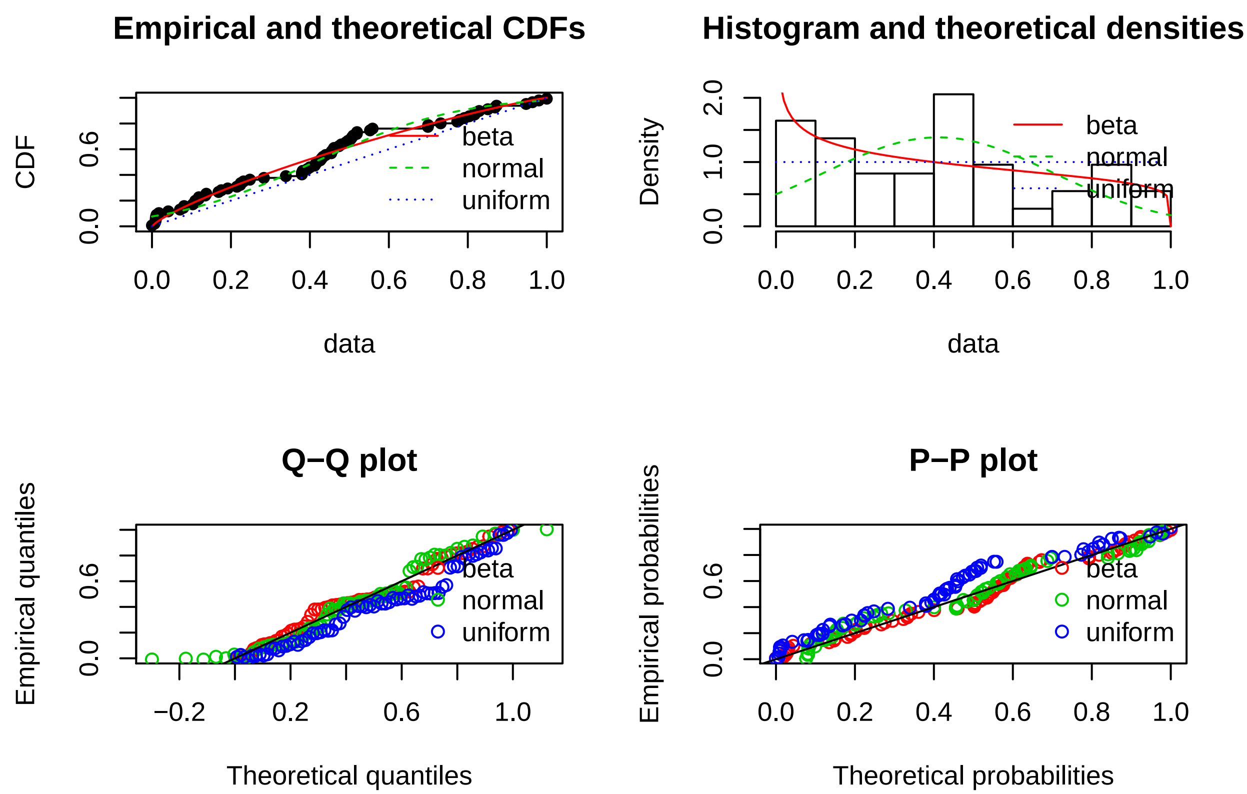

```
##fit normal
fitrate3_norm<-fitdist(recessionrate3$Value_b,"norm")
plot(fitrate3_norm)
```




```r
##fit uniform
fitrate3_unif<-fitdist(recessionrate3$Value_b,"unif")
plot(fitrate3_unif)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","normal","uniform")
cdfcomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif),legendtext=c("beta","normal","uniform"))
denscomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif),legendtext=c("beta","normal","uniform"),ylim =
qqcomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif),legendtext=c("beta","normal","uniform"))
ppcomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif),legendtext=c("beta","normal","uniform"))
```




```
gofstat(list(fitrate3_beta,fitrate3_norm,fitrate3_unif),fitnames=c("beta","normal","uniform"))
```

```
## Goodness-of-fit statistics
##                                   beta     normal   uniform
## Kolmogorov-Smirnov statistic 0.1049348 0.09400874 0.2203462
## Cramer-von Mises statistic   0.1174020 0.17366960 0.7635152
## Anderson-Darling statistic         Inf 1.25911585       Inf
##
## Goodness-of-fit criteria
##                                beta   normal uniform
## Akaike's Information Criterion  NaN 29.48691      NA
## Bayesian Information Criterion  NaN 34.06783      NA
```

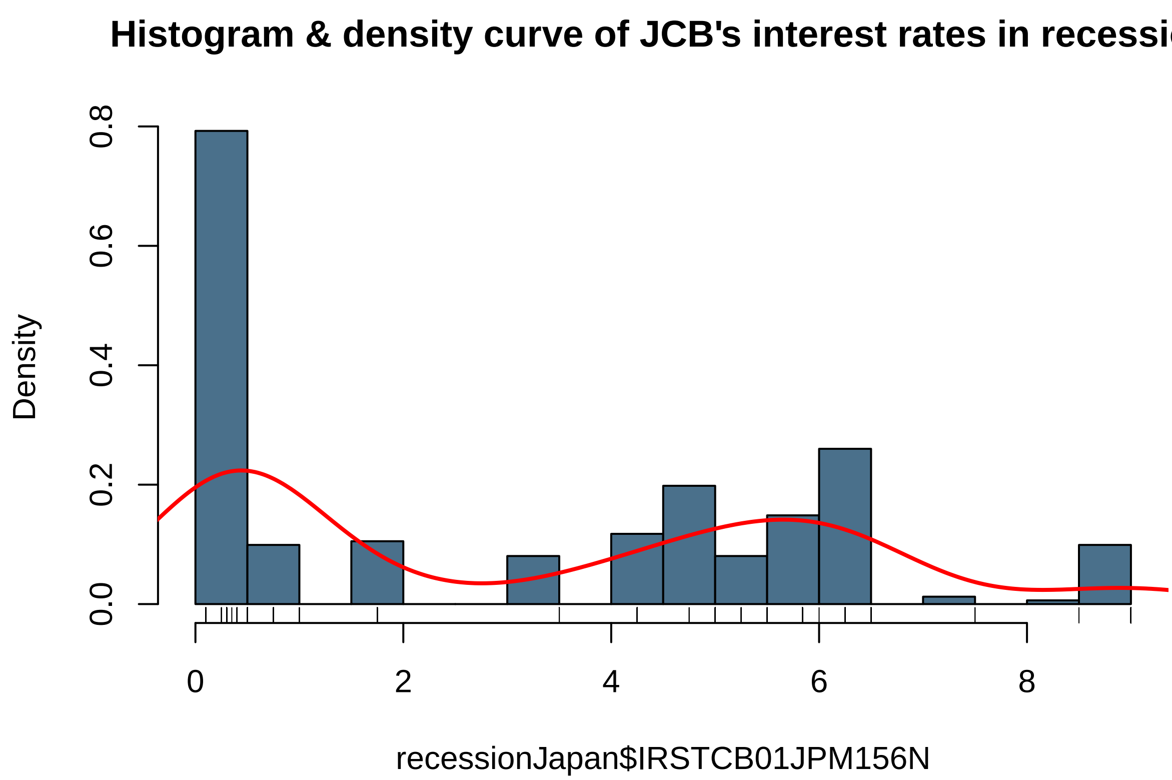
From the graphs, Three month treasury rate in recession years fits the normal distribution well.

Japanese Central Bank's interest rates in recession years

Find the recession years defined by OECD. Retrived from: https://fred.stlouisfed.org/series/DEUREC

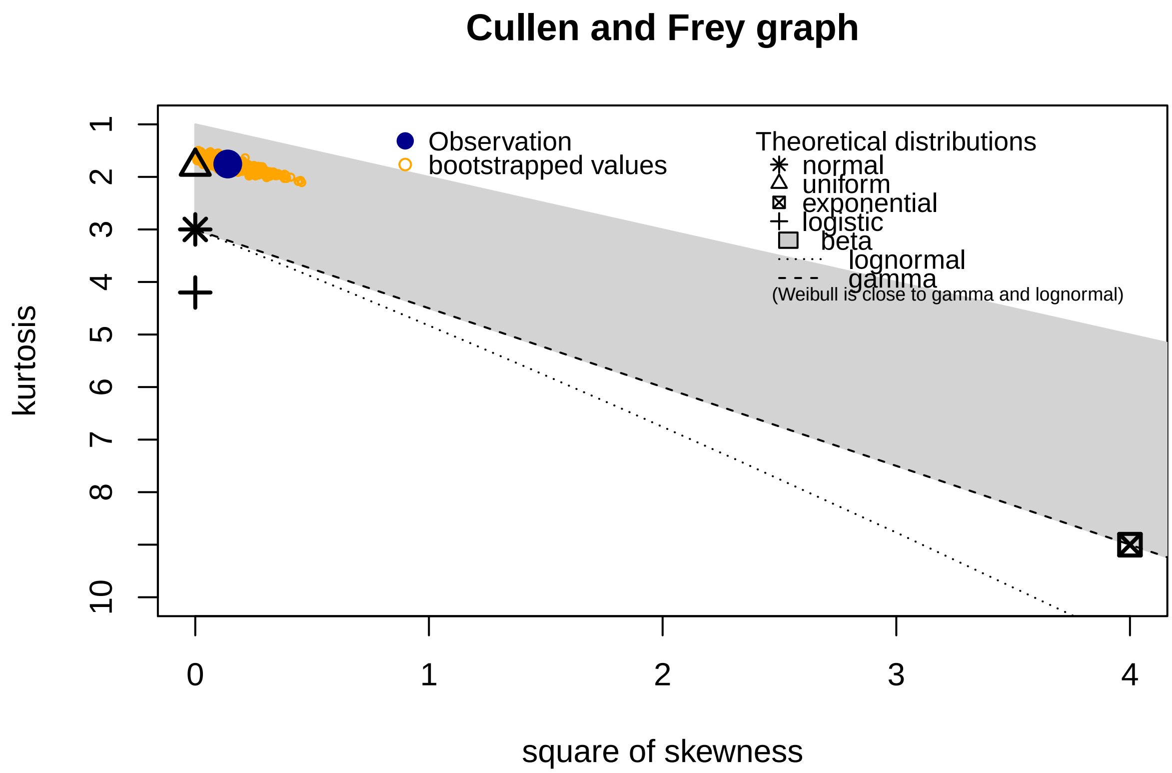
```
recessionJapan=subset(data_Japan, data_Japan$DATE < '1970-09-01' )
recessionJapan = rbind(recessionJapan, subset(data_Japan, data_Japan$DATE > '1968-09-01' & data_Japan$D
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '1973-11-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '1972-03-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '1975-07-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '1982-12-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '1993-10-01' & data_Japan$DA
```

```
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '1996-04-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '1999-02-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '2005-03-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '2009-07-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '2013-05-01' & data_Japan$DA
recessionJapan = rbind(recessionJapan,subset(data_Japan, data_Japan$DATE > '2015-10-01' & data_Japan$DA

hist(recessionJapan$IRSTCB01JPM156N,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogram &
lines(density(recessionJapan$IRSTCB01JPM156N),col="red",lwd=2)
rug(recessionJapan$IRSTCB01JPM156N)
```



fit distribution

```
descdist(recessionJapan$IRSTCB01JPM156N,boot = 1001)
```



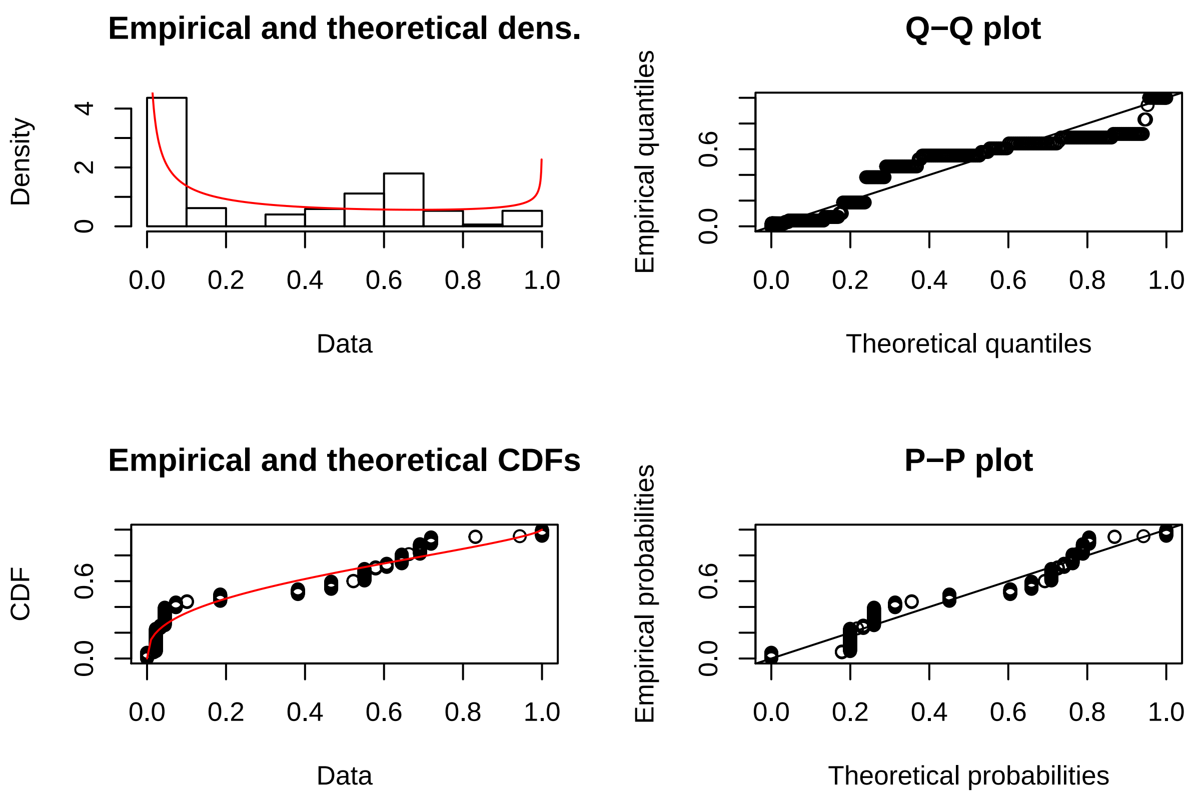

```
## summary statistics
## ------
## min:  0.1   max:  9
## median:  3.5
## mean:  3.152693
## estimated sd:  2.811875
## estimated skewness:  0.3729817
## estimated kurtosis:  1.75455
```

This variable may fit beta and uniform distribution. So we choose these distributions to have a further test.

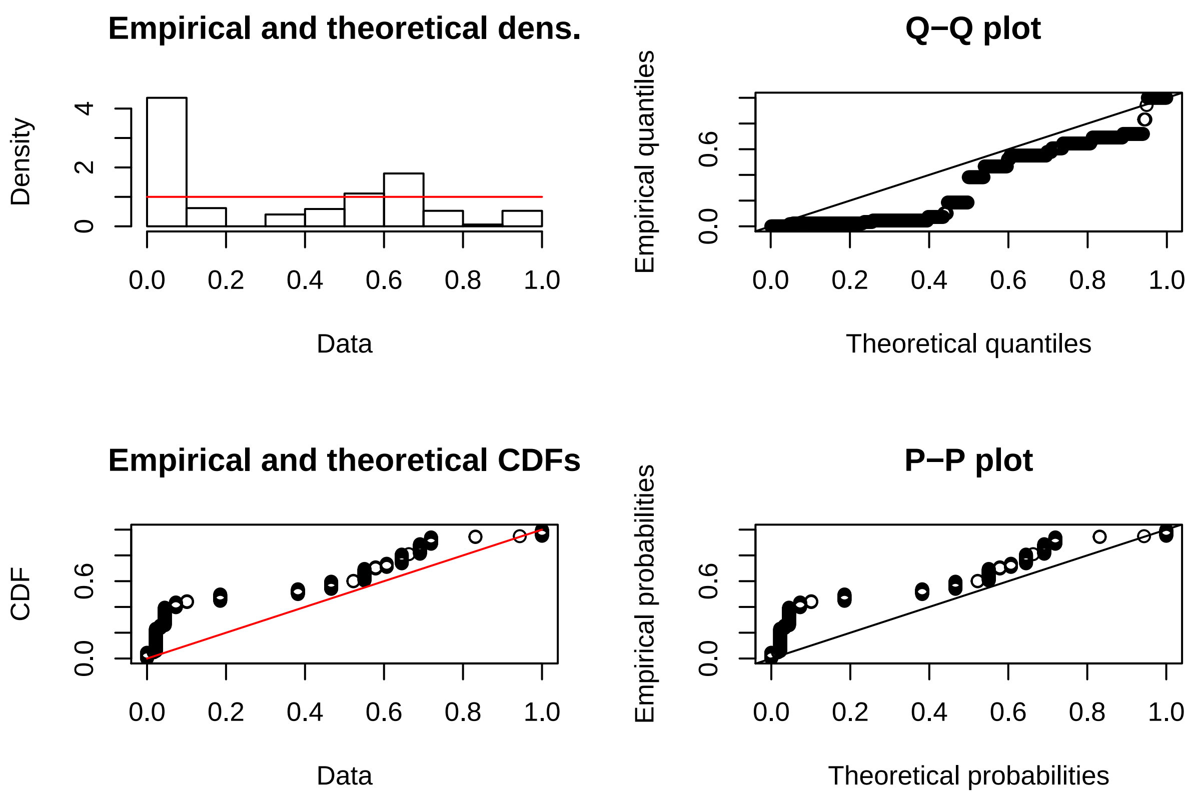
```
##fit beta
recessionJapan$IRSTCB01JPM156N_b<-(recessionJapan$IRSTCB01JPM156N-min(recessionJapan$IRSTCB01JPM156N))/
fitJapanrate_beta<-fitdist(recessionJapan$IRSTCB01JPM156N_b,"beta", method = "mge")
```

```
## Warning in fitdist(recessionJapan$IRSTCB01JPM156N_b, "beta", method =
## "mge"): maximum GOF estimation has a default 'gof' argument set to 'CvM'
```

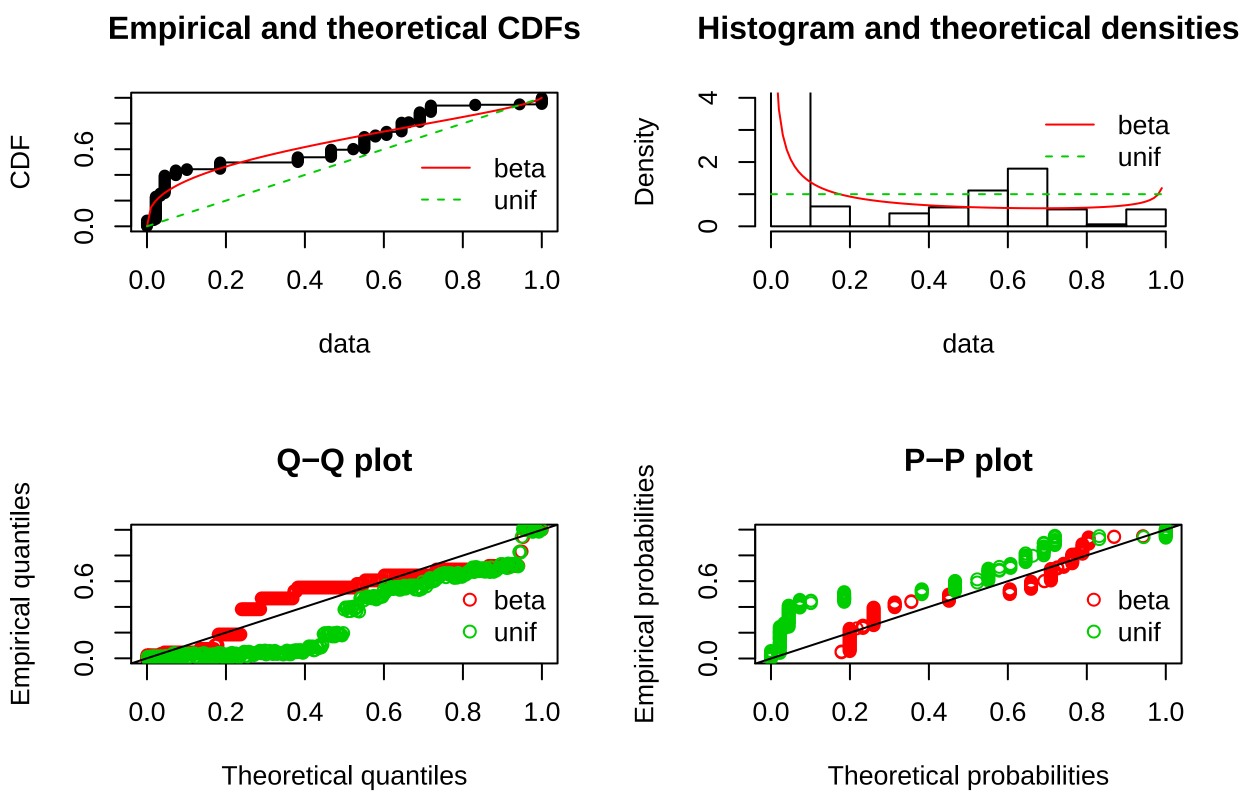
```
plot(fitJapanrate_beta)
```




```
##fit uniform
fitJapanrate_unif<-fitdist(recessionJapan$IRSTCB01JPM156N_b,"unif")
plot(fitJapanrate_unif)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","unif")
cdfcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
denscomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"),ylim = (0:4))
qqcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
ppcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
```




```
gofstat(list(fitJapanrate_beta,fitJapanrate_unif),fitnames=c("beta","unif"))
```

```
## Goodness-of-fit statistics
##                                  beta       unif
## Kolmogorov-Smirnov statistic 0.143730  0.3634988
## Cramer-von Mises statistic   1.592664 10.9924331
## Anderson-Darling statistic        Inf        Inf
##
## Goodness-of-fit criteria
##                                beta unif
## Akaike's Information Criterion -Inf   NA
## Bayesian Information Criterion -Inf   NA
```

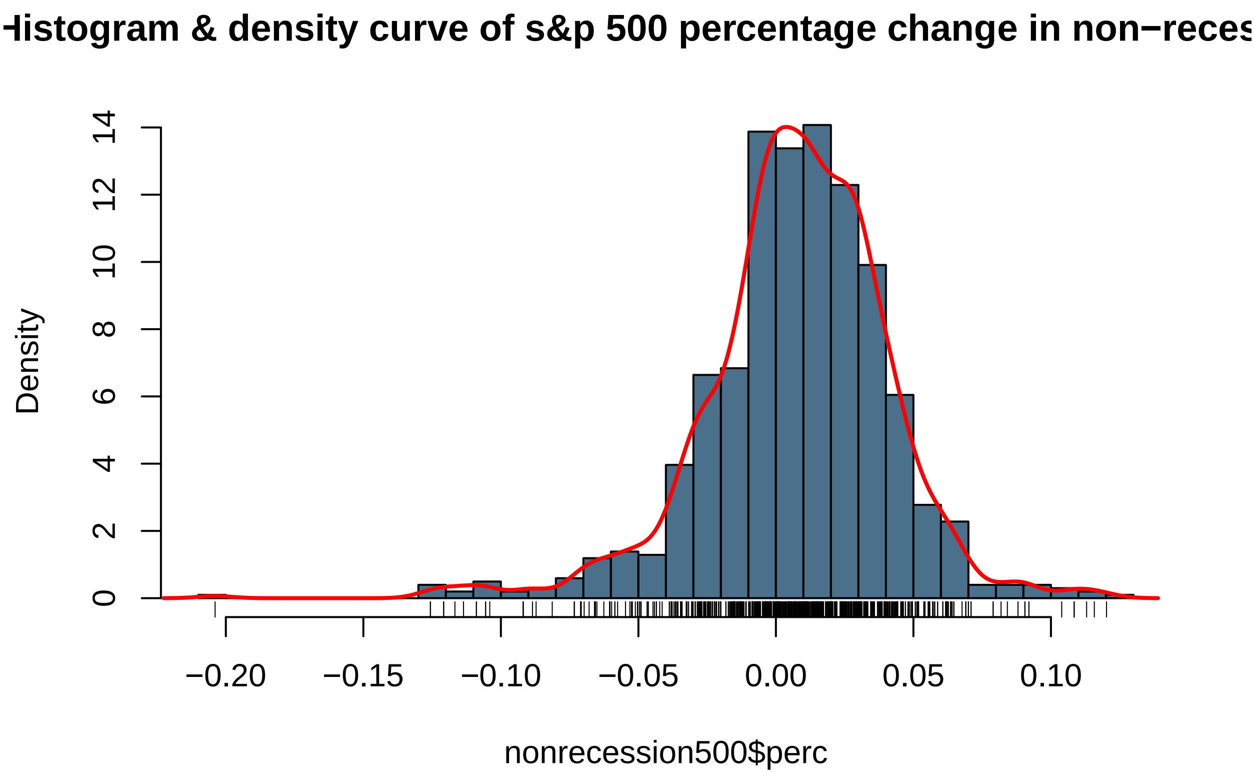
From the graphs, we can see that Japanese Central Bank's interest rates in recession fits the beta distribution well.

2 d

S&P500 percentage change with non-recession years

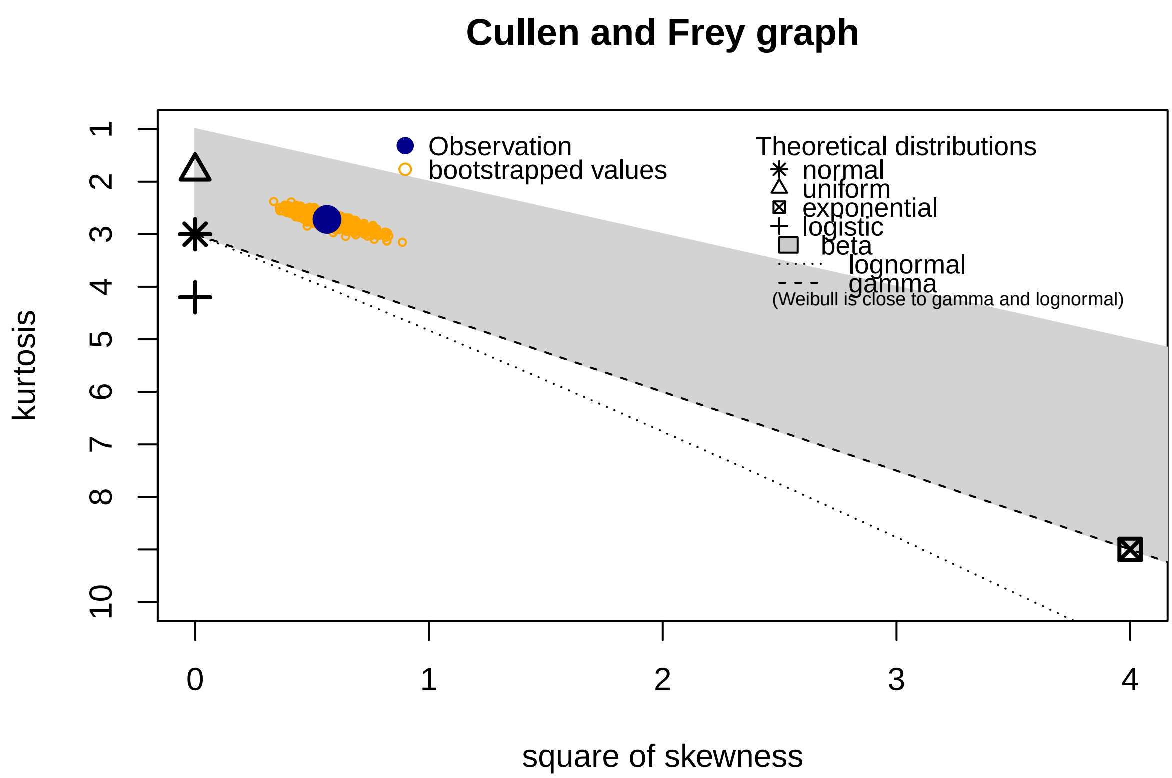
```
nonrecession500 = subset(data_sp500, data_sp500$Date < '1969-11-30' )
nonrecession500 = rbind(nonrecession500,subset(data_sp500, data_sp500$Date > '1970-12-02' & data_sp500$D
nonrecession500 = rbind(nonrecession500,subset(data_sp500, data_sp500$Date > '1975-04-02' & data_sp500$D
nonrecession500 = rbind(nonrecession500,subset(data_sp500, data_sp500$Date > '1980-08-01' & data_sp500$D
nonrecession500 = rbind(nonrecession500,subset(data_sp500, data_sp500$Date > '1982-12-01' & data_sp500$D
nonrecession500 = rbind(nonrecession500,subset(data_sp500, data_sp500$Date > '1991-04-01' & data_sp500$D
nonrecession500 = rbind(nonrecession500,subset(data_sp500, data_sp500$Date > '2001-12-01' & data_sp500$D
```

```
nonrecession500 = rbind(nonrecession500,subset(data_sp500, data_sp500$Date > '2009-07-01'))
##hist
hist(nonrecession500$perc,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogram & density cu
lines(density(nonrecession500$perc),col="red",lwd=2)
rug(nonrecession500$perc)
```



Fit distribution

```
descdist(nonrecession500$Value,boot = 1001)
```



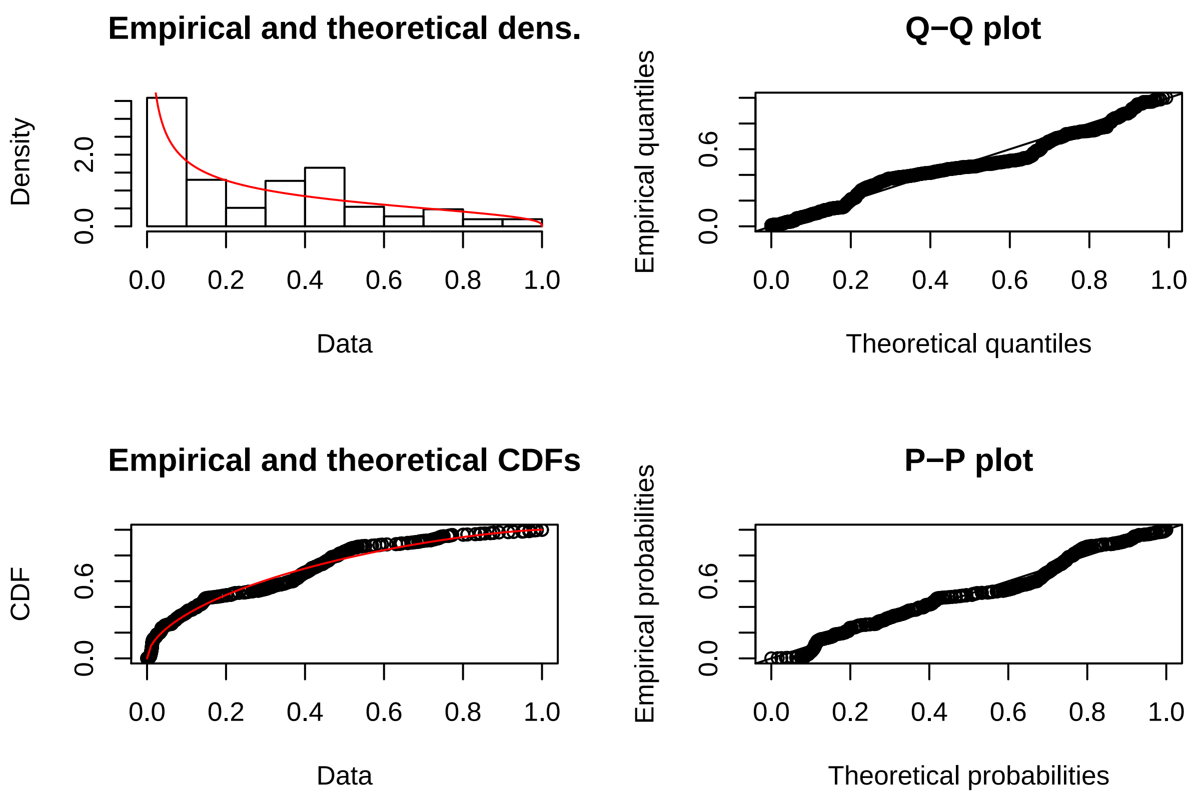

```
## summary statistics
## ------
## min:  67.07   max:  2789.8
## median:  649.54
## mean:  831.3937
## estimated sd:  687.1594
## estimated skewness:  0.7510358
## estimated kurtosis:  2.718304
```

From this graph, we assume that Yield Spread S&P500 percentage change with non-recession years may fit beta distribution. Here we have some further test:

```
#fit beta distribution
nonrecession500$Value_b<-(nonrecession500$Value-min(nonrecession500$Value))/max((nonrecession500$Value-m
fitsp500_beta<-fitdist(nonrecession500$Value_b,"beta", method="mge")
```

```
## Warning in fitdist(nonrecession500$Value_b, "beta", method = "mge"):
## maximum GOF estimation has a default 'gof' argument set to 'CvM'
```

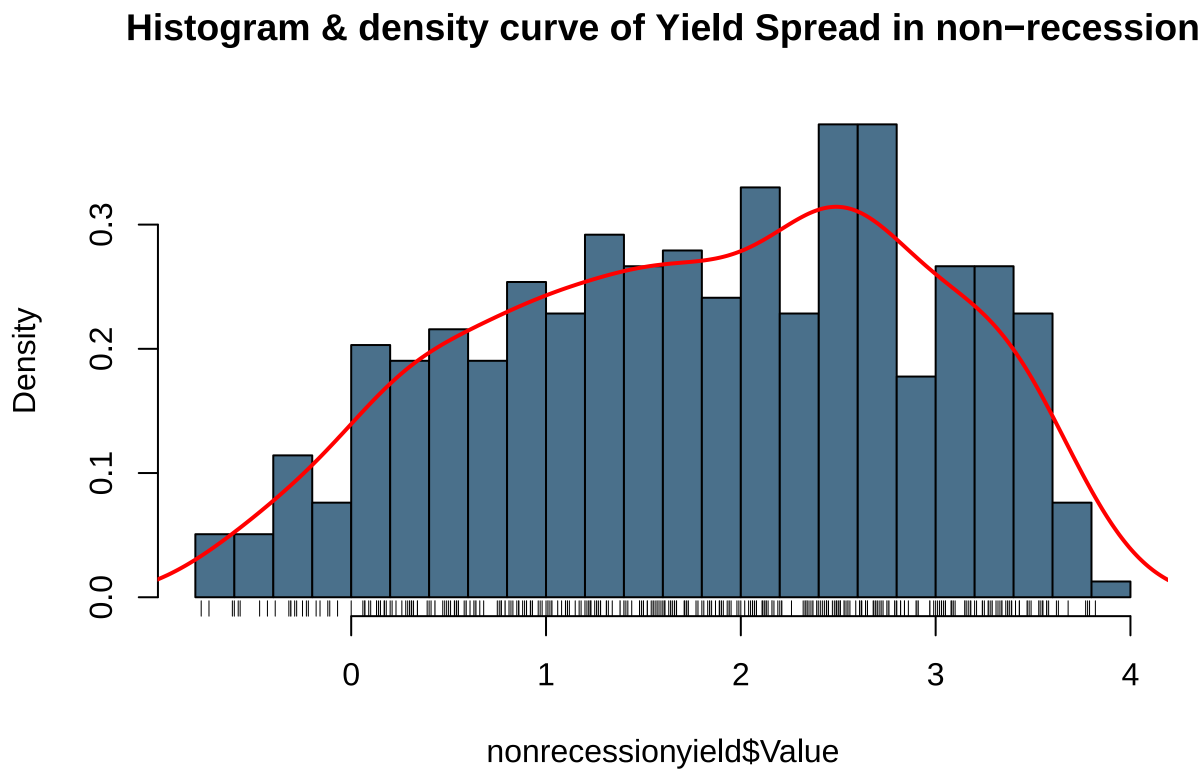
```
plot(fitsp500_beta)
```



From grapghs, we can conclude that S&P500 percentage change with non-recession years fits well with the beta distribution.

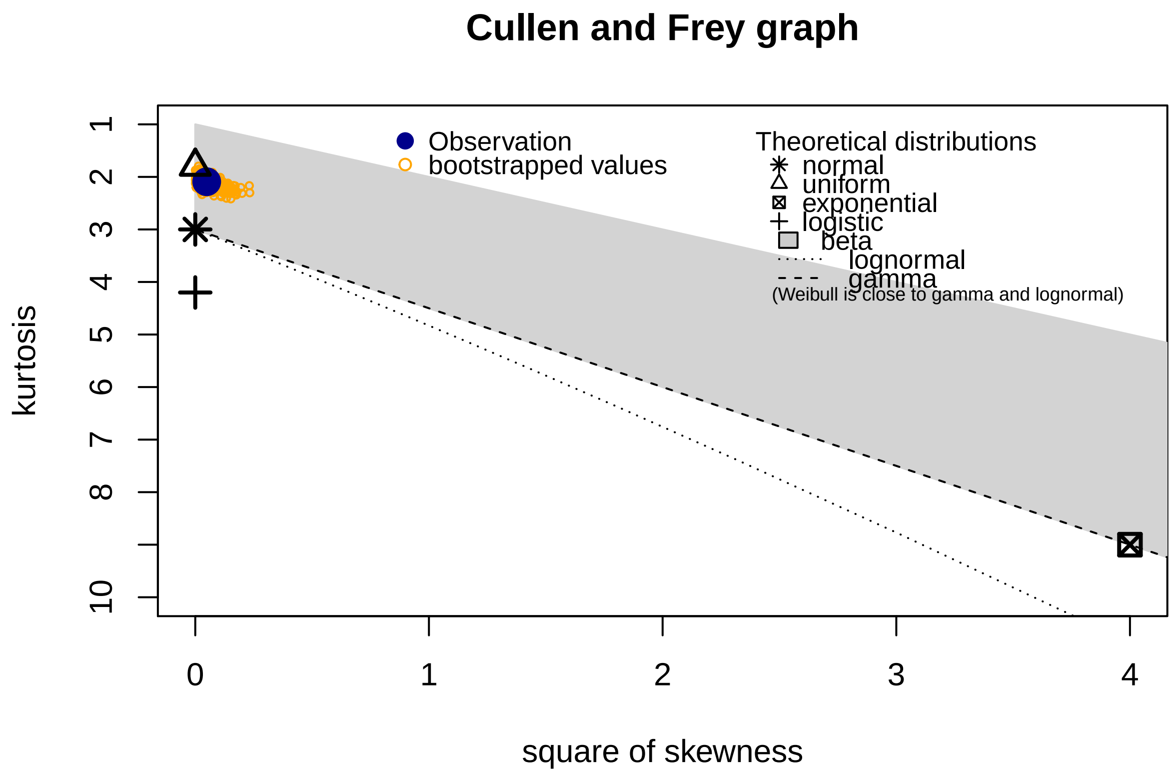
Yield Spread with non-recession years

```
nonrecessionyield = subset(data_yield,data_yield$Date< '1969-11-30')
nonrecessionyield = rbind(nonrecessionyield,subset(data_yield, data_yield$Date > '1970-12-02' & data_yie
nonrecessionyield = rbind(nonrecessionyield,subset(data_yield, data_yield$Date > '1975-04-02' & data_yie
nonrecessionyield = rbind(nonrecessionyield,subset(data_yield, data_yield$Date > '1980-08-01' & data_yie
nonrecessionyield = rbind(nonrecessionyield,subset(data_yield, data_yield$Date >  '1982-12-01'& data_yie
nonrecessionyield = rbind(nonrecessionyield,subset(data_yield, data_yield$Date > '1991-04-01' & data_yie
nonrecessionyield = rbind(nonrecessionyield,subset(data_yield, data_yield$Date > '2001-12-01' & data_yie
nonrecessionyield = rbind(nonrecessionyield,subset(data_yield, data_yield$Date > '2009-07-01'))
##histgram
hist(nonrecessionyield$Value,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogram & density
lines(density(nonrecessionyield$Value),col="red",lwd=2)
rug(nonrecessionyield$Value)
```



Fit distribution

```
descdist(nonrecessionyield$Value,boot = 1001)
```



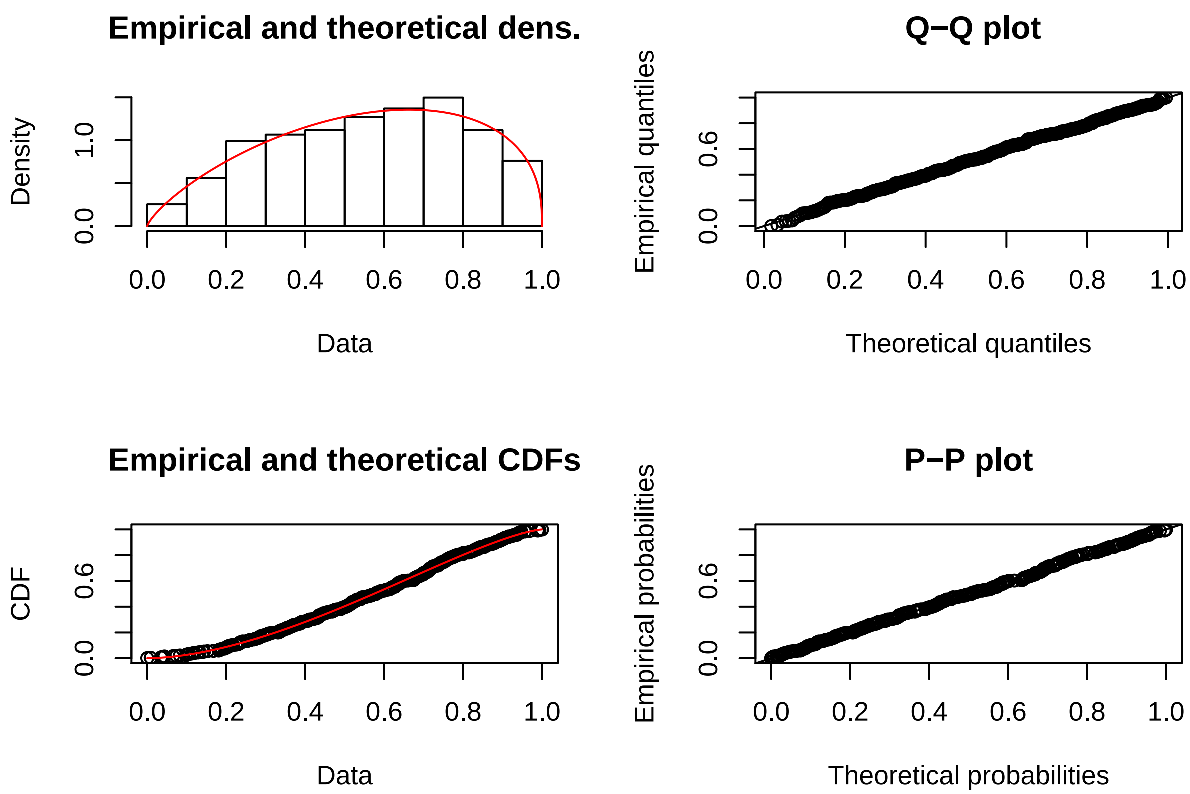

```
## summary statistics
## ------
## min:  -0.77   max:  3.82
## median:  1.89
## mean:  1.795888
## estimated sd:  1.117831
## estimated skewness:  -0.2212659
## estimated kurtosis:  2.092114
```

From this graph, we assume that Yield Spread in non-recession years may fit beta and uniform distribution. Here we have some further test:

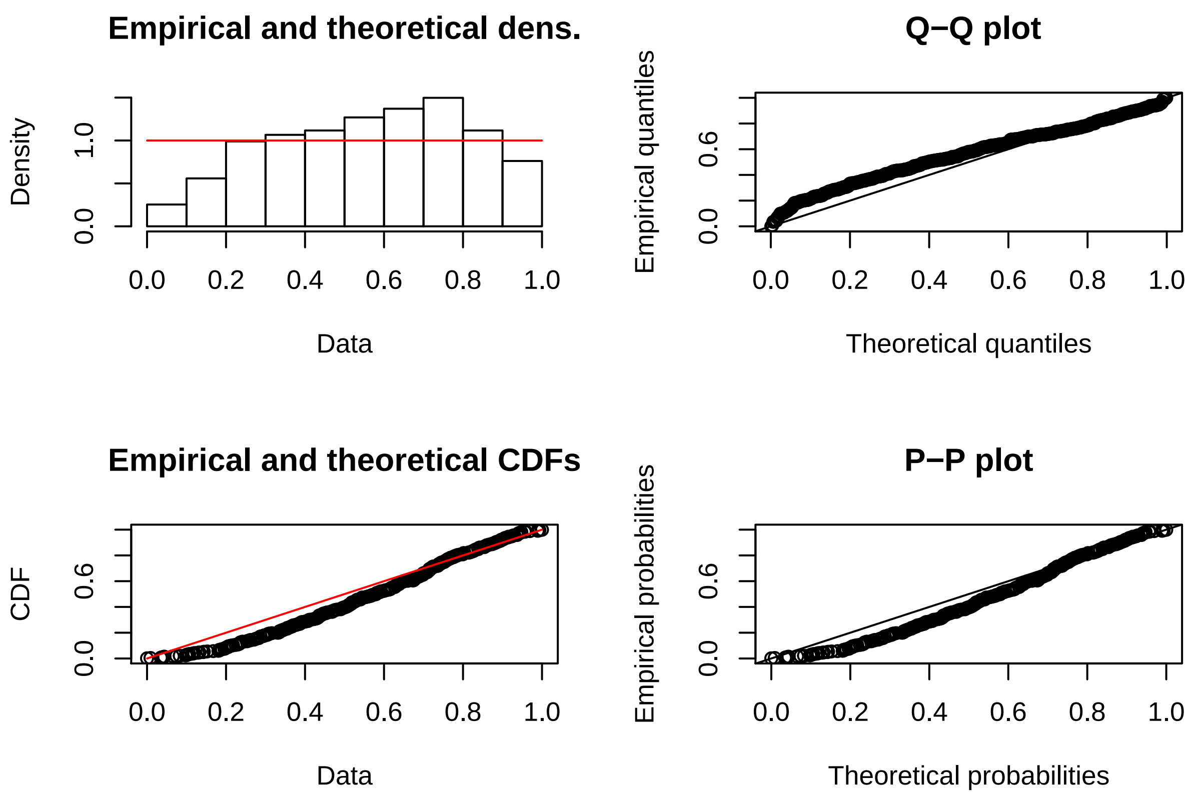
```
#fit beta distribution
nonrecessionyield$Value_b<-(nonrecessionyield$Value-min(nonrecessionyield$Value))/max((nonrecessionyiel
fityield_beta<-fitdist(nonrecessionyield$Value_b,"beta", method="mge")
```

```
## Warning in fitdist(nonrecessionyield$Value_b, "beta", method = "mge"):
## maximum GOF estimation has a default 'gof' argument set to 'CvM'
```

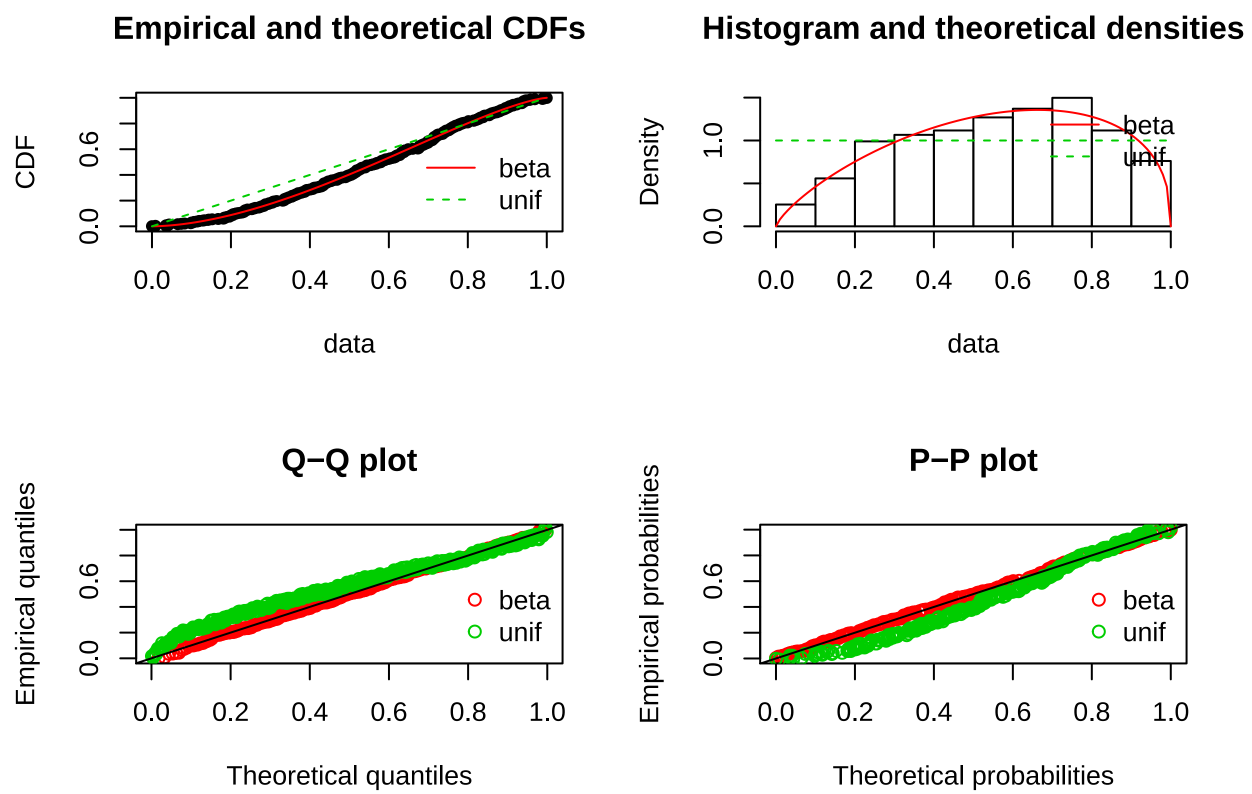
```
plot(fityield_beta)
```




```
#fit uniform distribution
fityield_unif<-fitdist(nonrecessionyield$Value_b,"unif")
plot(fityield_unif)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","uniform")
cdfcomp(list(fityield_beta,fityield_unif),legendtext=c("beta","unif"))
denscomp(list(fityield_beta,fityield_unif),legendtext=c("beta","unif"))
qqcomp(list(fityield_beta,fityield_unif),legendtext=c("beta","unif"))
ppcomp(list(fityield_beta,fityield_unif),legendtext=c("beta","unif"))
```




```
gofstat(list(fityield_beta,fityield_unif),fitnames=c("beta","unif"))
```

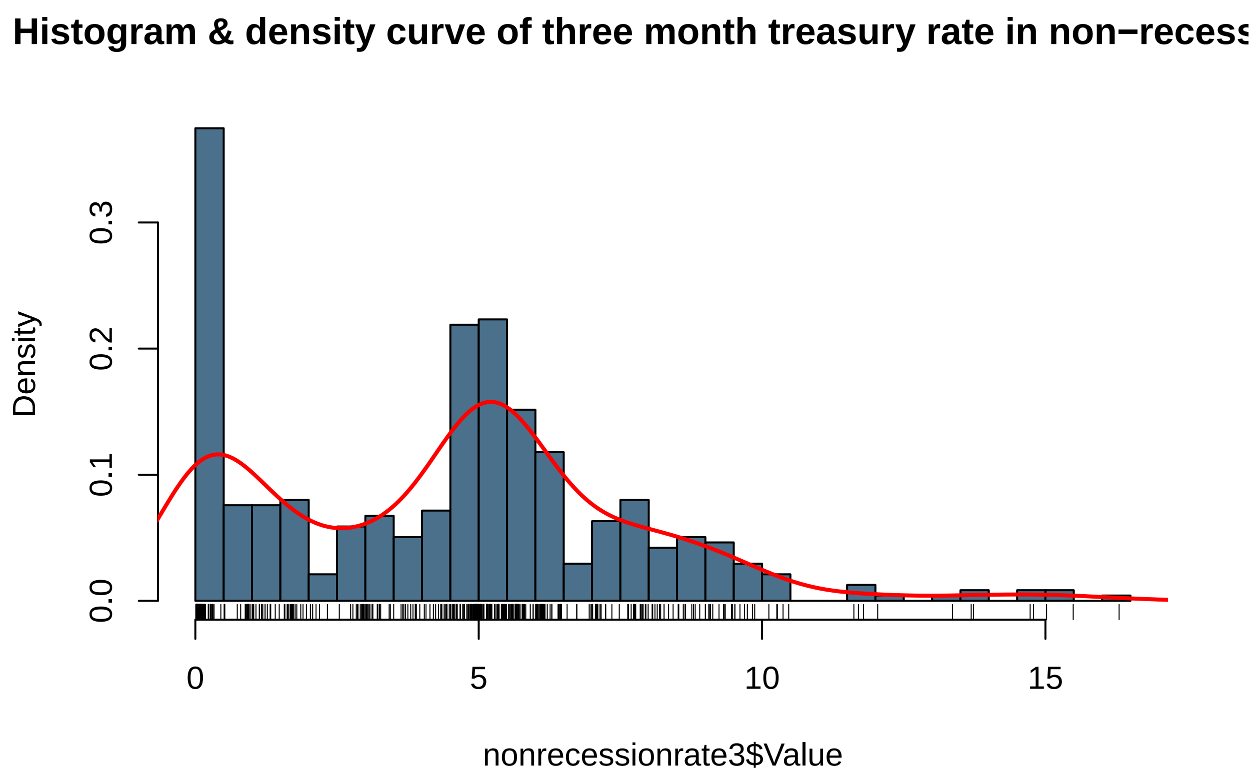
```
## Goodness-of-fit statistics
##                                    beta      unif
## Kolmogorov-Smirnov statistic 0.03002566 0.1331851
## Cramer-von Mises statistic   0.02918169 2.4800284
## Anderson-Darling statistic          Inf       Inf
##
## Goodness-of-fit criteria
##                                beta unif
## Akaike's Information Criterion  Inf   NA
## Bayesian Information Criterion  Inf   NA
```

From the ftiing grapghs of distribution and "R Console", we can conclude that Yield Spread in non-recession years fits well with the beta distribution.

Three month treasury rate in non-recession years

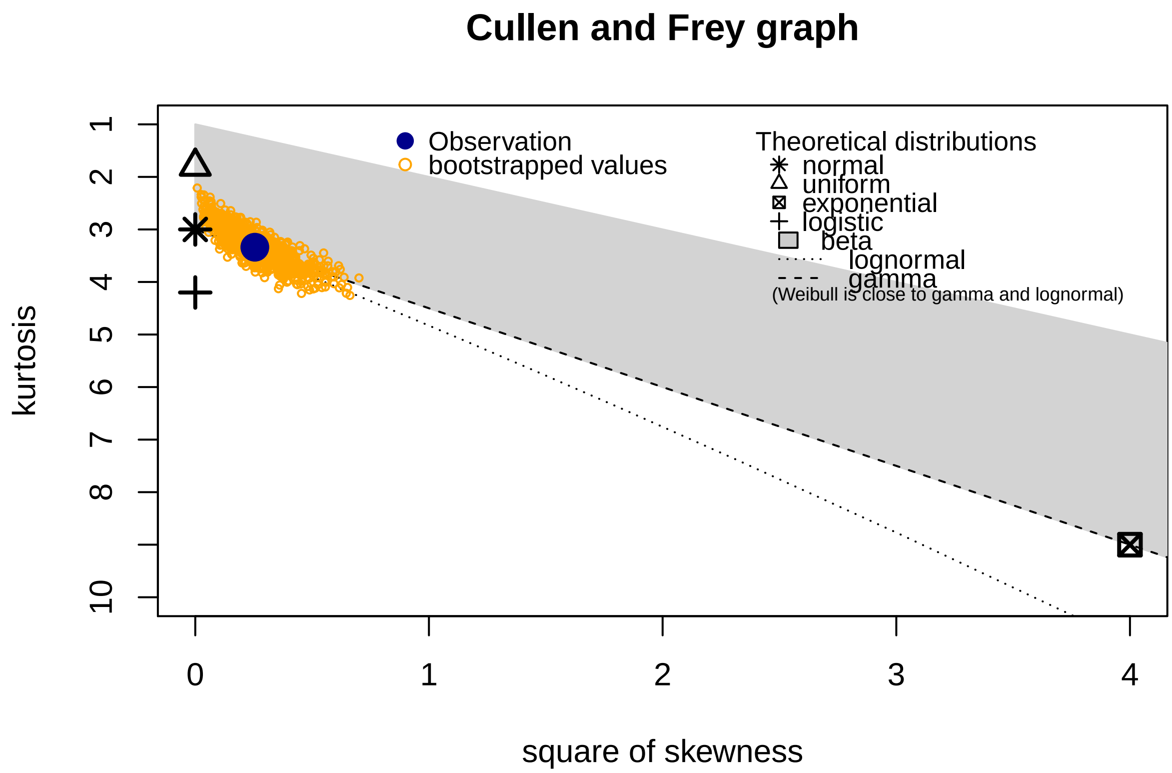
```
nonrecessionrate3=subset(data_rate3,data_rate3$Date< '1969-11-30')
nonrecessionrate3 = rbind(nonrecessionrate3,subset(recessionyield, data_yield$Date > '1970-12-02' & data
nonrecessionrate3 = rbind(nonrecessionrate3,subset(data_rate3, data_rate3$Date > '1975-04-02'& data_rate
nonrecessionrate3 = rbind(nonrecessionrate3,subset(data_rate3, data_rate3$Date > '1980-08-01' & data_ra
nonrecessionrate3 = rbind(nonrecessionrate3,subset(data_rate3, data_rate3$Date > '1982-12-01' & data_ra
nonrecessionrate3 = rbind(nonrecessionrate3,subset(data_rate3, data_rate3$Date > '1991-04-01' & data_ra
nonrecessionrate3 = rbind(nonrecessionrate3,subset(data_rate3, data_rate3$Date > '2001-12-01' & data_ra
nonrecessionrate3 = rbind(nonrecessionrate3,subset(data_rate3, data_rate3$Date > '2009-07-01'))
```

```r
#hist
hist(nonrecessionrate3$Value,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogram & densit
lines(density(nonrecessionrate3$Value),col="red",lwd=2)
rug(nonrecessionrate3$Value)
```



fit distribution

```r
descdist(nonrecessionrate3$Value,boot = 1001)
```



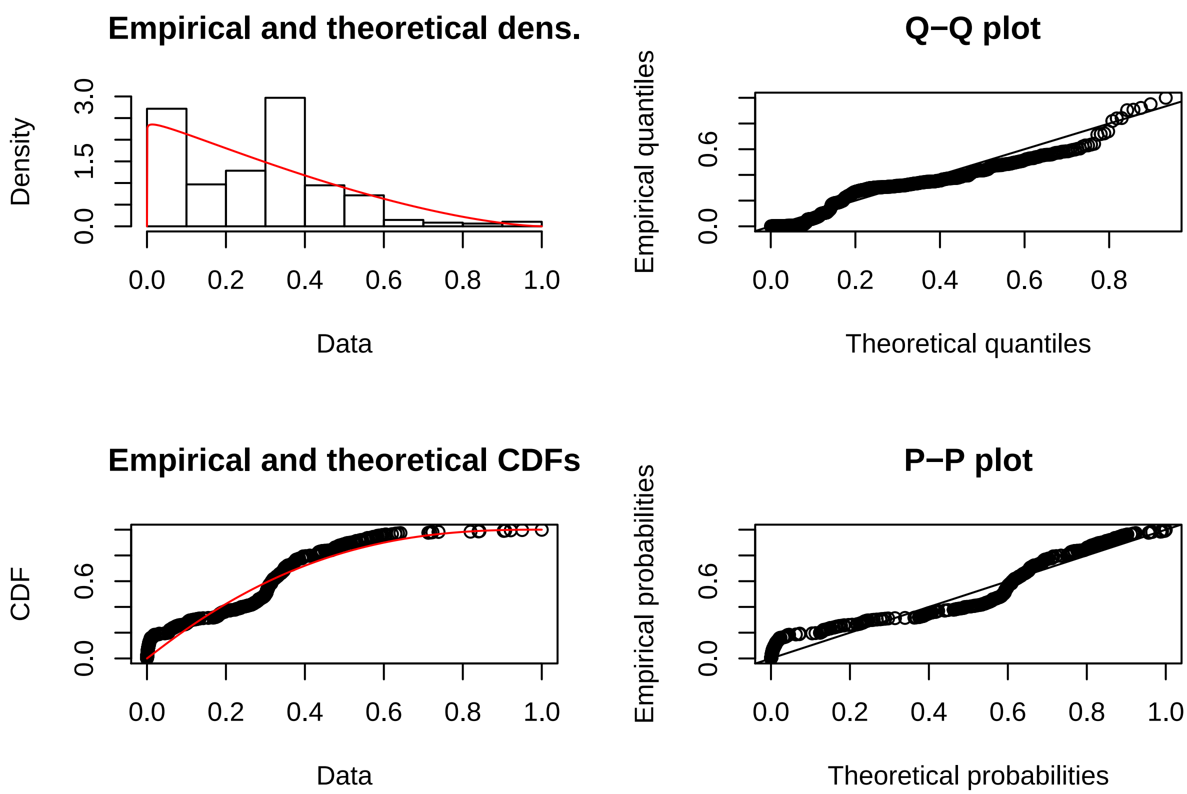

```
## summary statistics
## ------
## min:  0.01   max:  16.3
## median:  4.91
## mean:  4.382105
## estimated sd:  3.223406
## estimated skewness:  0.5049214
## estimated kurtosis:  3.338593
```

This variable may fit beta normal, uniform, and gamma distribution. So we choose these distributions to have a further test.

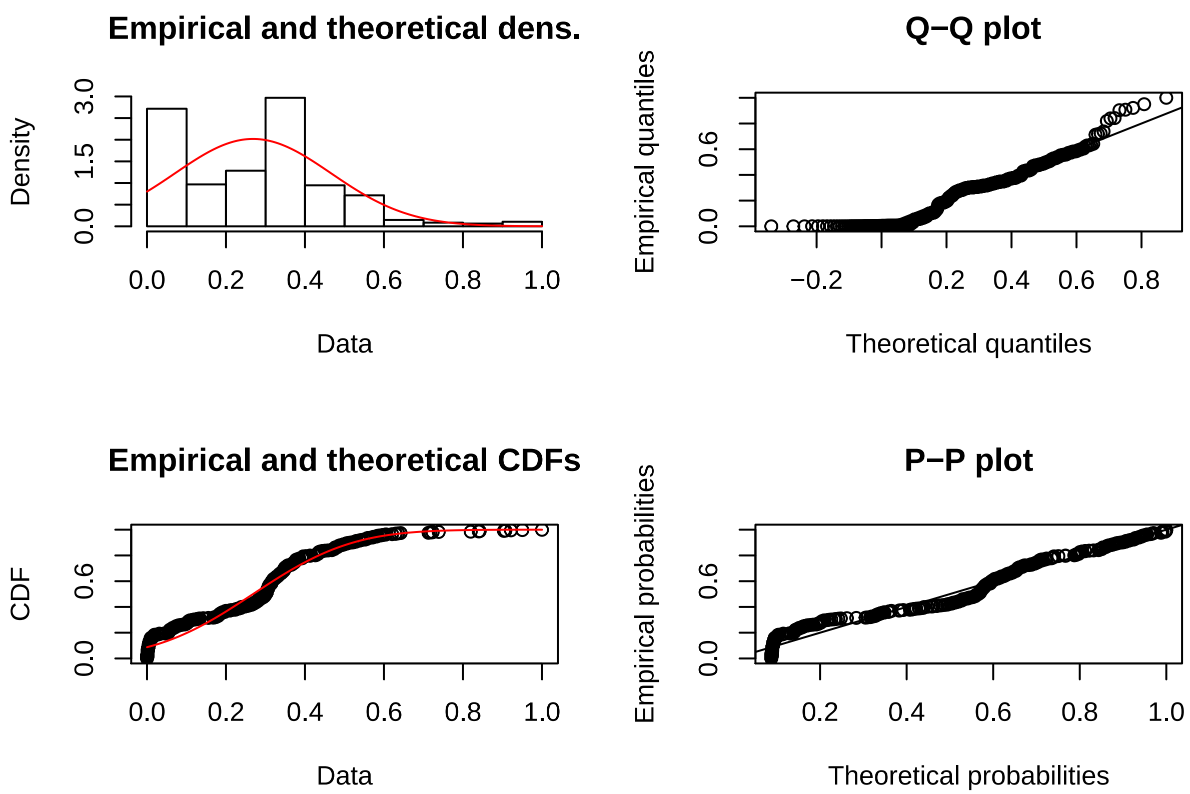
```
##fit beta distribution
nonrecessionrate3$Value_b<-(nonrecessionrate3$Value-min(nonrecessionrate3$Value))/max(nonrecessionrate3$
fitrate3_beta<-fitdist(nonrecessionrate3$Value_b,"beta", method="mge")
```

```
## Warning in fitdist(nonrecessionrate3$Value_b, "beta", method = "mge"):
## maximum GOF estimation has a default 'gof' argument set to 'CvM'
```

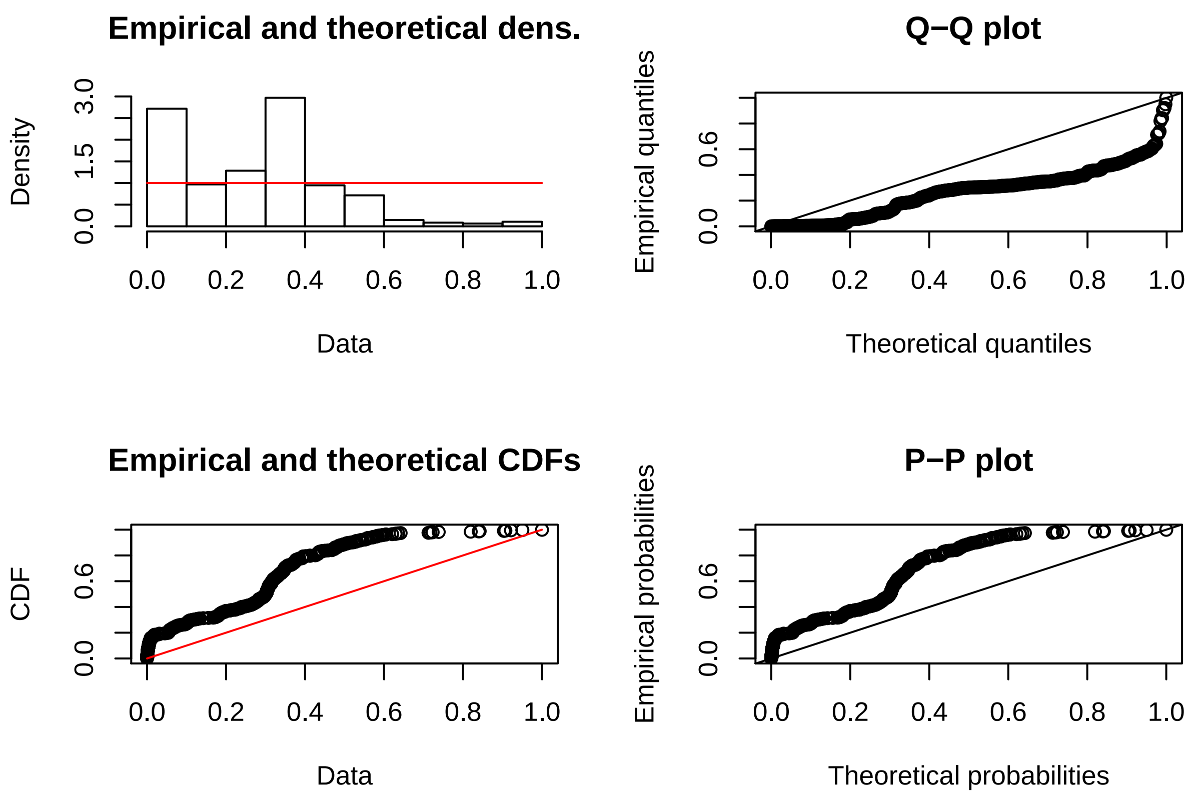
```
plot(fitrate3_beta)
```




```
##fit normal
fitrate3_norm<-fitdist(nonrecessionrate3$Value_b,"norm")
plot(fitrate3_norm)
```



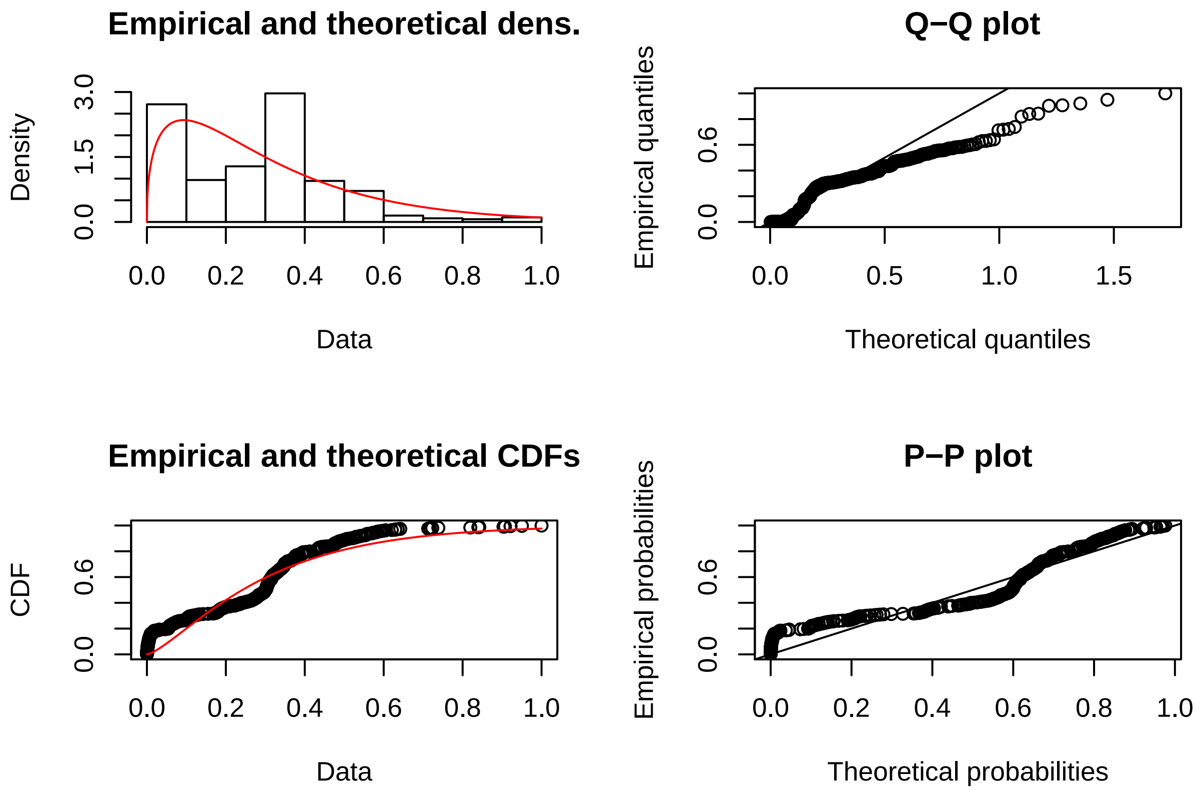

```
##fit uniform
fitrate3_unif<-fitdist(nonrecessionrate3$Value_b,"unif")
plot(fitrate3_unif)
```



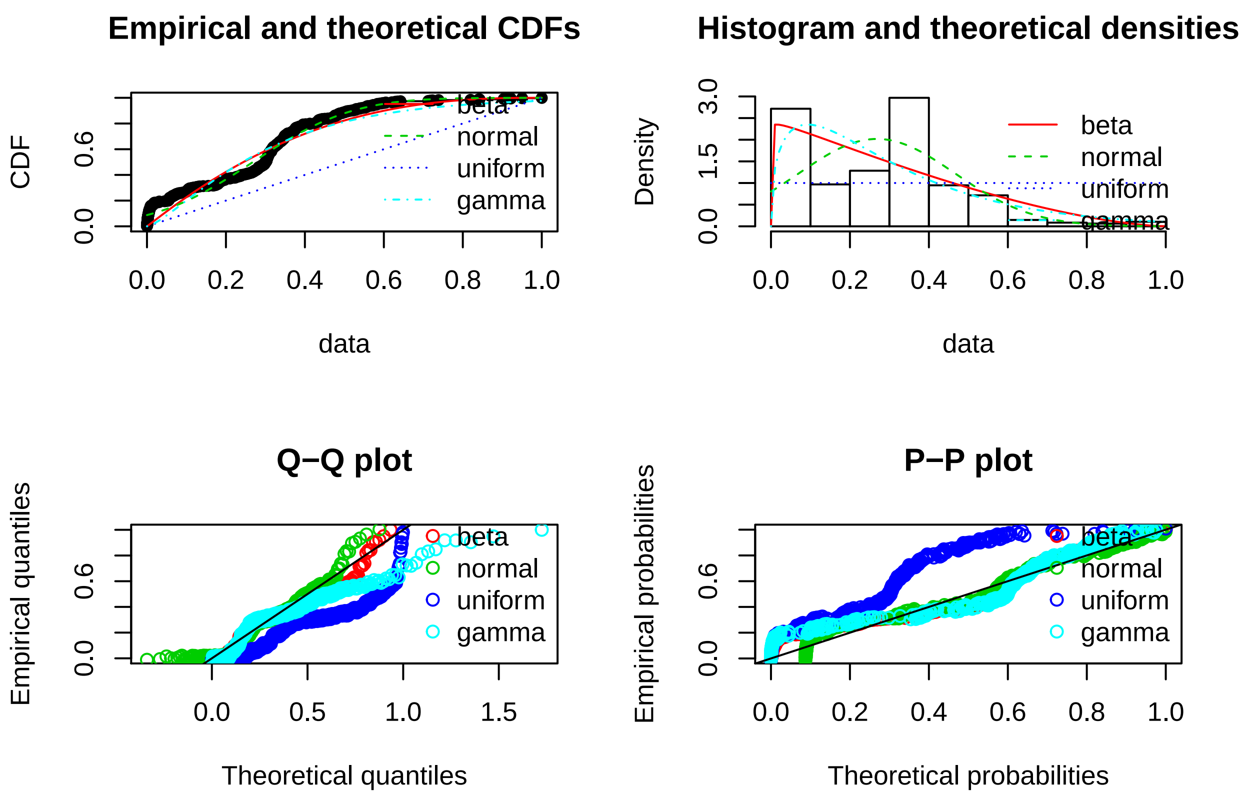

```
##fit gamma
fitrate3_gamma<-fitdist(nonrecessionrate3$Value_b,"gamma", method="mge")
```

```
## Warning in fitdist(nonrecessionrate3$Value_b, "gamma", method = "mge"):
## maximum GOF estimation has a default 'gof' argument set to 'CvM'
```

```
plot(fitrate3_gamma)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","normal","uniform","gamma")
cdfcomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif,fitrate3_gamma),legendtext=c("beta","normal","uni
denscomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif,fitrate3_gamma),legendtext=c("beta","normal","u
qqcomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif,fitrate3_gamma),legendtext=c("beta","normal","uni
ppcomp(list(fitrate3_beta,fitrate3_norm,fitrate3_unif,fitrate3_gamma),legendtext=c("beta","normal","uni
```




```
gofstat(list(fitrate3_beta,fitrate3_norm,fitrate3_unif,fitrate3_gamma),fitnames=c("beta","normal","unif
```

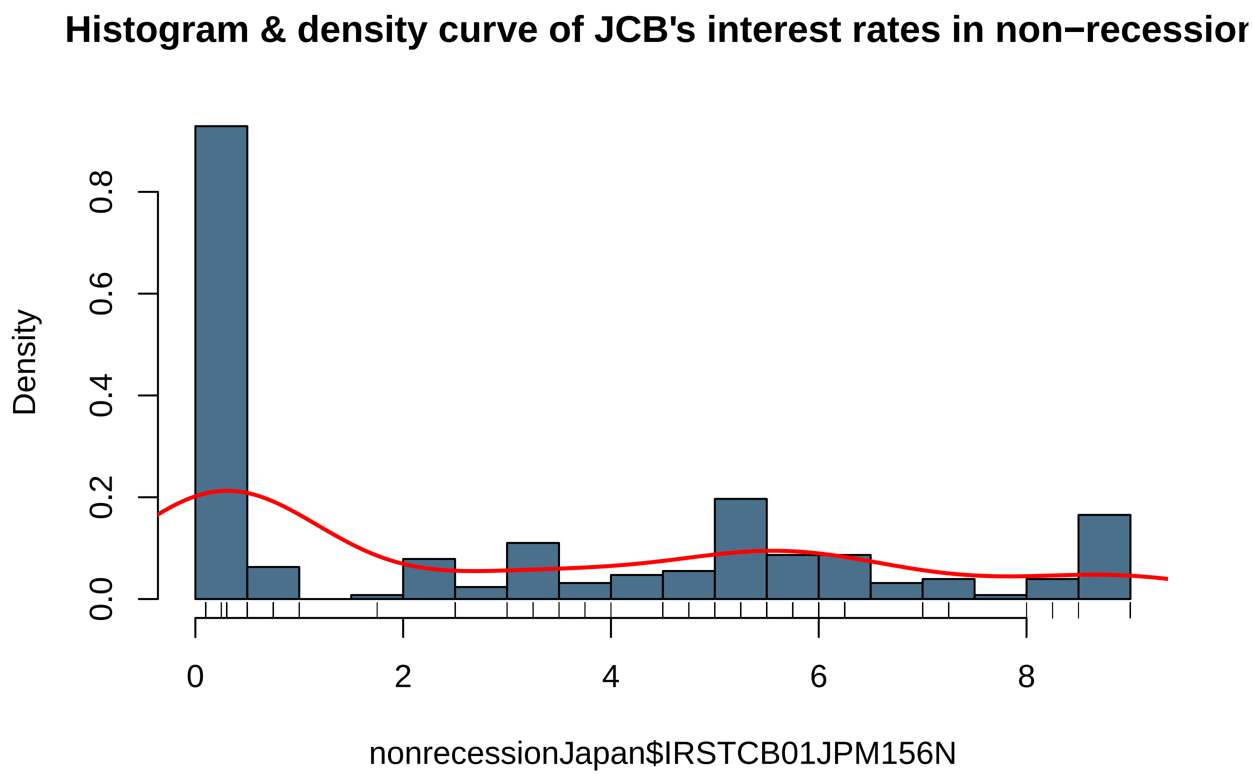
```
## Goodness-of-fit statistics
##                                   beta     normal    uniform     gamma
## Kolmogorov-Smirnov statistic 0.1401719 0.08881966  0.3977358 0.1609426
## Cramer-von Mises statistic   2.3901103 1.04151073 31.4957600 3.0530690
## Anderson-Darling statistic         Inf 7.56424555        Inf       Inf
##
## Goodness-of-fit criteria
##                                beta    normal uniform gamma
## Akaike's Information Criterion  Inf -188.1166      NA   Inf
## Bayesian Information Criterion  Inf -179.7900      NA   Inf
```

Three month treasury rate in non-recession years fits the normal distribution well.

Japanese Central Bank's interest rates in nonrecession years

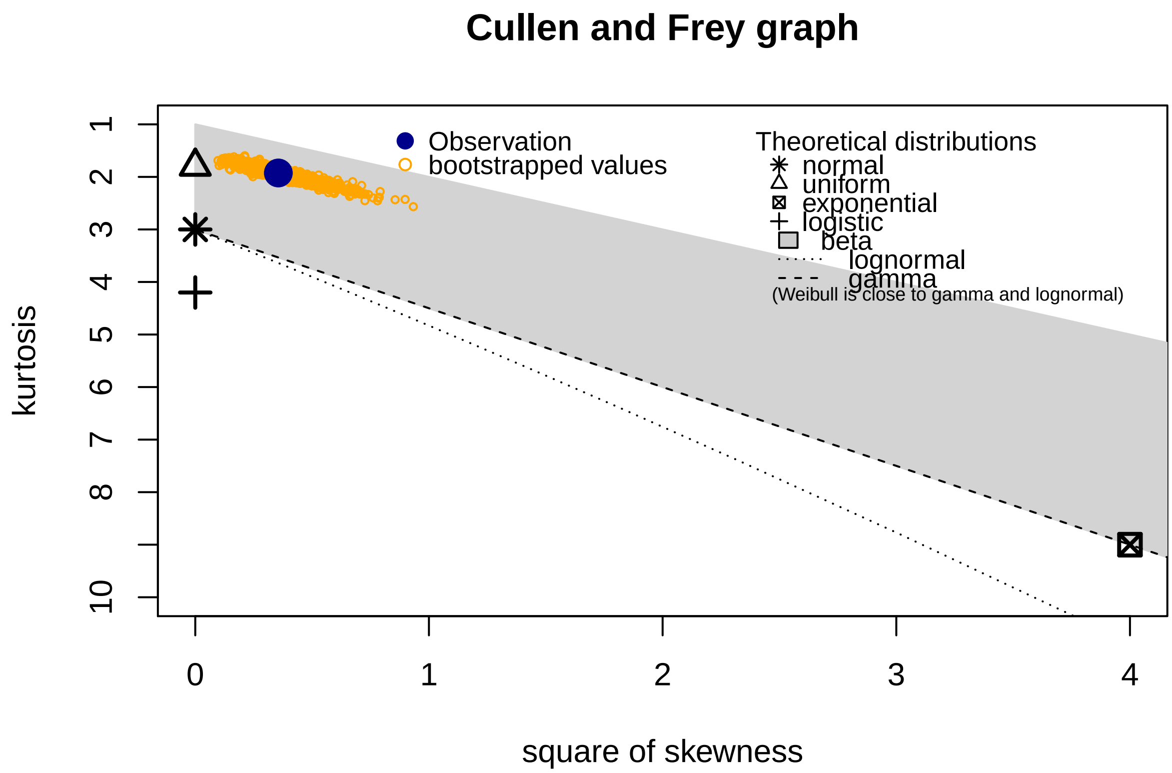
```
nonrecessionJapan = subset(data_Japan, data_Japan$DATE > '1970-09-01' & data_Japan$DATE < '1972-03-01')
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '1973-04-30' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '1979-11-30' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '1985-09-30' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE >  '1991-03-30'& data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '1995-06-30' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '1998-03-01' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '2001-06-01' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '2008-03-01' & data_Ja
```

```
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '2011-08-01' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '2014-04-01' & data_Ja
nonrecessionJapan = rbind(nonrecessionJapan,subset(data_Japan, data_Japan$DATE > '2017-10-01' ))

hist(nonrecessionJapan$IRSTCB01JPM156N,breaks=30,col="skyblue4",freq = FALSE,prob=TRUE,main = "Histogra
lines(density(nonrecessionJapan$IRSTCB01JPM156N),col="red",lwd=2)
rug(nonrecessionJapan$IRSTCB01JPM156N)
```



fit distribution

```
descdist(nonrecessionJapan$IRSTCB01JPM156N,boot = 1001)
```



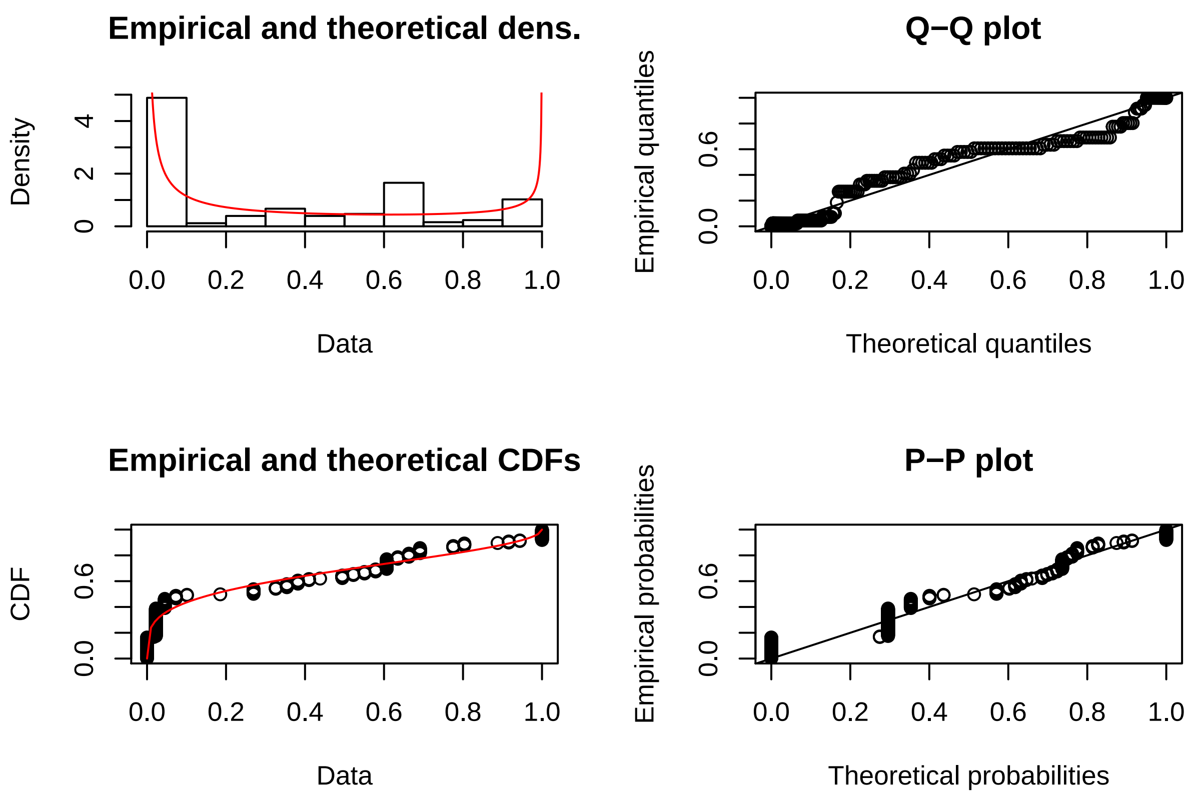

```
## summary statistics
## ------
## min:  0.1   max:  9
## median:  2.125
## mean:  3.030315
## estimated sd:  3.087692
## estimated skewness:  0.5963175
## estimated kurtosis:  1.925783
```

This variable may fit beta and uniform distribution. So we choose these distributions to have a further test.

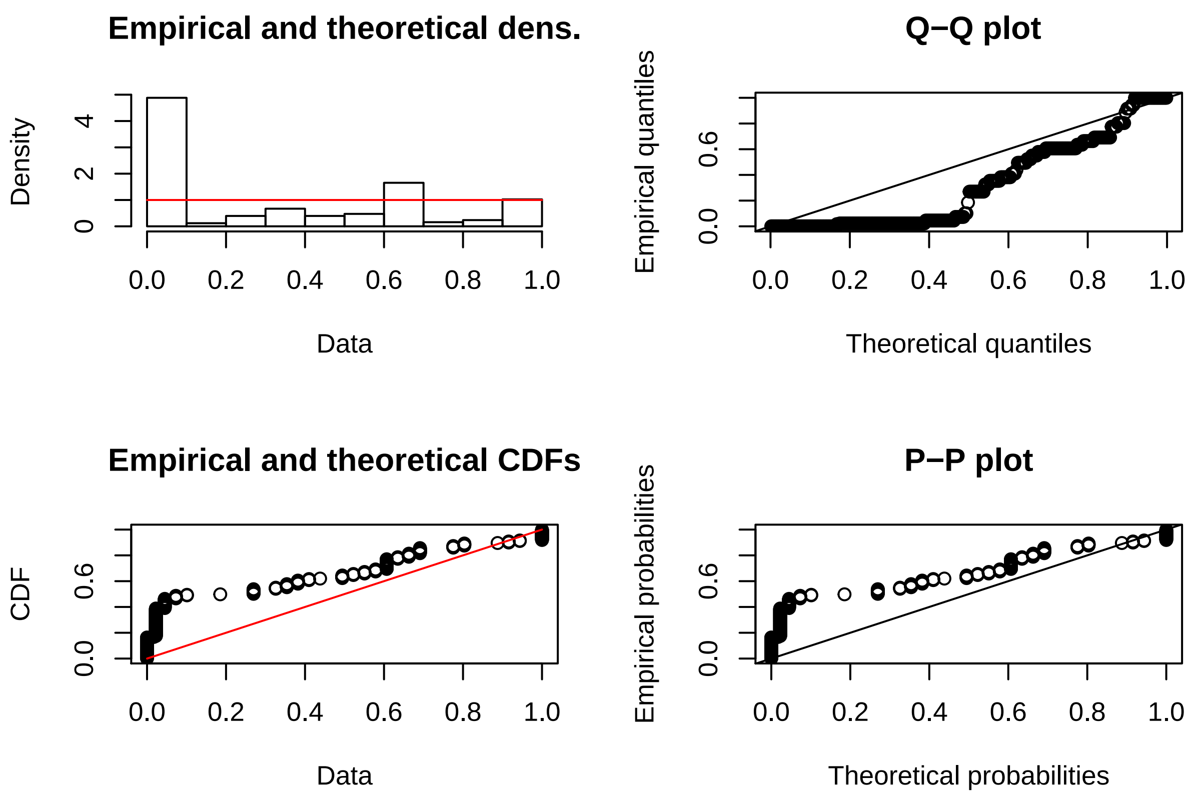
```
##fit beta
nonrecessionJapan$IRSTCB01JPM156N_b<-(nonrecessionJapan$IRSTCB01JPM156N-min(nonrecessionJapan$IRSTCB01J
fitJapanrate_beta<-fitdist(nonrecessionJapan$IRSTCB01JPM156N_b,"beta", method = "mge")
```

```
## Warning in fitdist(nonrecessionJapan$IRSTCB01JPM156N_b, "beta", method =
## "mge"): maximum GOF estimation has a default 'gof' argument set to 'CvM'
```

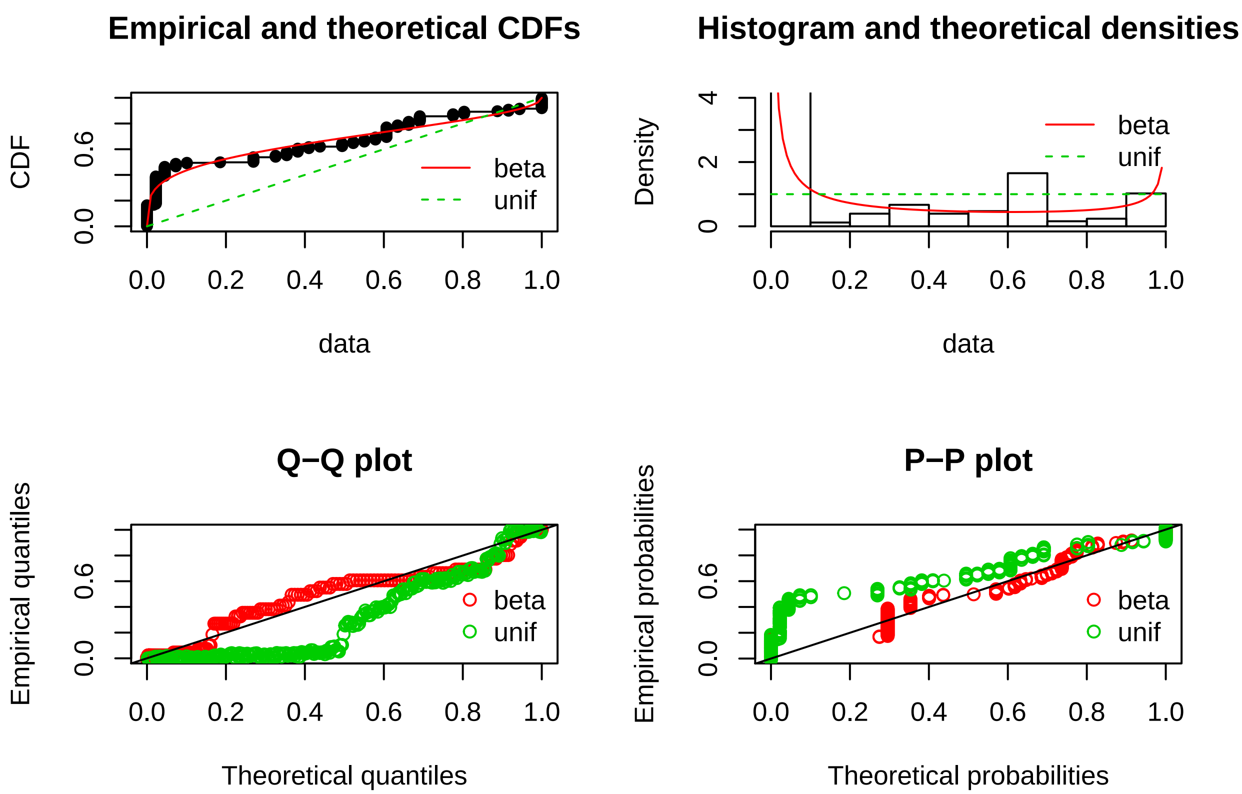
```
plot(fitJapanrate_beta)
```




```
##fit uniform
fitJapanrate_unif<-fitdist(nonrecessionJapan$IRSTCB01JPM156N_b,"unif")
plot(fitJapanrate_unif)
```




```
par(mfrow=c(2,2))
plot.legend<-c("beta","unif")
cdfcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
denscomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"),ylim = (0:4))
qqcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
ppcomp(list(fitJapanrate_beta,fitJapanrate_unif),legendtext=c("beta","unif"))
```




```r
gofstat(list(fitJapanrate_beta,fitJapanrate_unif),fitnames=c("beta","unif"))
```

```
## Goodness-of-fit statistics
##                                   beta       unif
## Kolmogorov-Smirnov statistic 0.1653543  0.4196231
## Cramer-von Mises statistic   1.0440847 11.2171554
## Anderson-Darling statistic         Inf        Inf
##
## Goodness-of-fit criteria
##                                beta unif
## Akaike's Information Criterion -Inf   NA
## Bayesian Information Criterion -Inf   NA
```

Japanese Central Bank's interest rates in nonrecession years fits the beta distribution well.

2 e

About the data value, our intuition tells us that during the recession years, the changes in the S&P 500 will be more negative, the Yield Spread will be smaller, the interest rates will go down, and the Japanese Central Bank's interest rates will go down. The histgrams show that our intuitioin is correct.

As for the distribution, there is noticeable differences between the estimated distributions based on how we subset the data.It's because the situations of economy in recession and in non-recession are totally diffrent. In recession years, economy is not good and it will display in the data values, which will influence distribution.

2 f

```r
# Do x and y come from the same distribution?
#ks.test(x, y)
##S&P500 percentage change
ks.test(recession500$perc,nonrecession500$perc)
```

```
## Warning in ks.test(recession500$perc, nonrecession500$perc): p-value will
## be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  recession500$perc and nonrecession500$perc
## D = 0.25629, p-value = 3.861e-05
## alternative hypothesis: two-sided
```

```r
## Yield Spread
ks.test(recessionyield$Value,nonrecessionyield$Value)
```

```
## Warning in ks.test(recessionyield$Value, nonrecessionyield$Value): p-value
## will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  recessionyield$Value and nonrecessionyield$Value
## D = 0.17248, p-value = 0.1567
## alternative hypothesis: two-sided
```

```r
##Three month treasury rate
ks.test(recessionrate3$Value,nonrecessionrate3$Value)
```

```
## Warning in ks.test(recessionrate3$Value, nonrecessionrate3$Value): p-value
## will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  recessionrate3$Value and nonrecessionrate3$Value
## D = 0.32695, p-value = 2.667e-06
## alternative hypothesis: two-sided
```

```r
##Japanese Central Bank's interest rates
ks.test(recessionJapan$IRSTCB01JPM156N,nonrecessionJapan$IRSTCB01JPM156N)
```

```
## Warning in ks.test(recessionJapan$IRSTCB01JPM156N, nonrecessionJapan
## $IRSTCB01JPM156N): p-value will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  recessionJapan$IRSTCB01JPM156N and nonrecessionJapan$IRSTCB01JPM156N
## D = 0.15757, p-value = 0.001717
## alternative hypothesis: two-sided
```

Except the Yield Spread, other variables have small D and p-value in K-T test, so we should reject the null, which means the pairs of values we test come from different distributions. As fot the Yield Spread, its D = 0.17248, p-value = 0.1567, which means Yield Spread in recession and nonrecession come from the same distribution. But we still can draw the conclution that there are noticeable differences between the estimated distributions based on how you subset the data. So that's why when we predict the economy and design our model, we should try to capture diffrent economic states.

3 a

Microsoft is the most aggresive firm and Exxon is the most defensive firm according the beta values.

```
## get data from excel
library(readxl)
capmdat <- read_excel("C:/Users/rossw/Documents/MAE Program/Econometrics 403A/Homework/capm4.xlsx")

##  perform regressions
regdis=lm(capmdat$dis-capmdat$riskfree ~ (capmdat$mkt-capmdat$riskfree))
regge=lm(capmdat$ge-capmdat$riskfree ~ (capmdat$mkt-capmdat$riskfree))
reggm=lm(capmdat$gm-capmdat$riskfree ~ (capmdat$mkt-capmdat$riskfree))
regibm=lm(capmdat$ibm-capmdat$riskfree ~ (capmdat$mkt-capmdat$riskfree))
regmsft=lm(capmdat$msft-capmdat$riskfree ~ (capmdat$mkt-capmdat$riskfree))
regxom=lm(capmdat$xom-capmdat$riskfree ~ (capmdat$mkt-capmdat$riskfree))
## get max and min betas to determine most aggresive and defensive firm
cof= cbind(coefficients(regdis), coefficients(regge),coefficients(reggm),coefficients(regibm),coefficien
colnames(cof)=c("dis","ge","gm","ibm","msft","xom")
rownames(cof)=c("Intercept","Betas")
cof
```

```
##                   dis          ge          gm         ibm        msft
## Intercept -0.03289025 -0.03292527 -0.04428567 -0.02667249 -0.02677718
## Betas      0.86241707  0.86929395  1.23480687  1.15520974  1.28641872
##                   xom
## Intercept -0.02257146
## Betas      0.38064435
```

```
rownames(cof)=c("Intercept", "Beta")
apply(cof, 1, FUN=max) #microsoft
```

```
##   Intercept        Beta
## -0.02257146  1.28641872
```

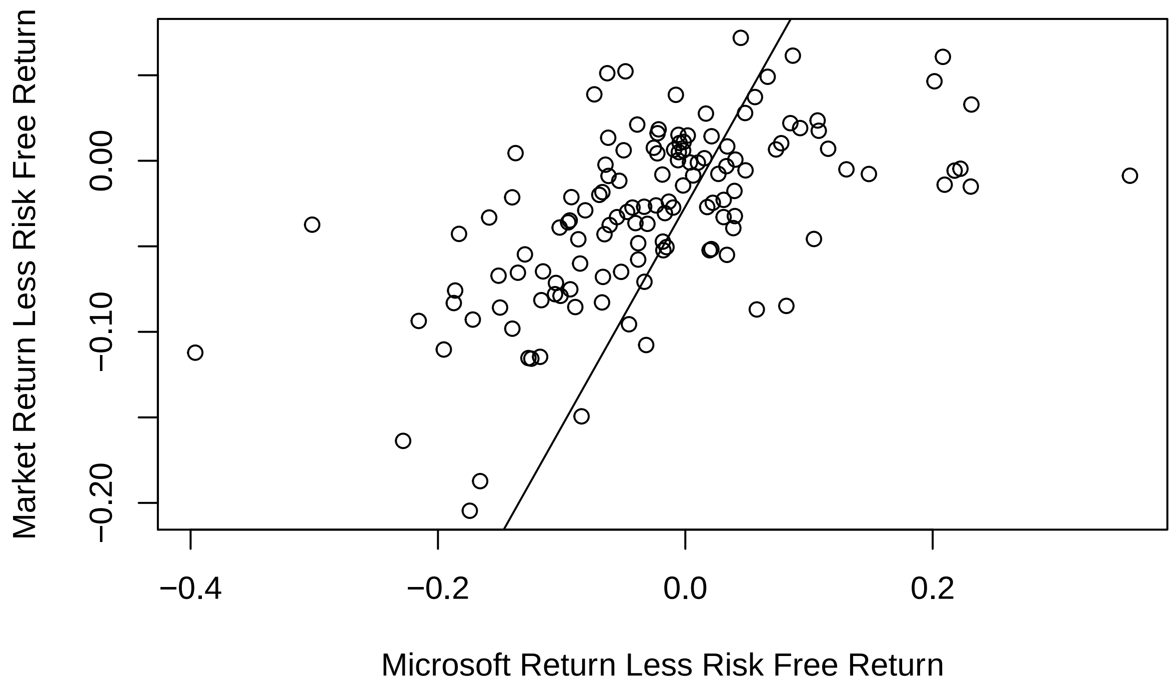
```
apply(cof, 1, FUN=min) #exxon
```

```
##   Intercept        Beta
## -0.04428567  0.38064435
```

3 b

The values of the intercepts of our regression do seem to be close to 0.

```
##plot microsoft scatter and regression
library(readxl)
capmdat <- read_excel("C:/Users/rossw/Documents/MAE Program/Econometrics 403A/Homework/capm4.xlsx")
regmsft=lm(capmdat$msft-capmdat$riskfree ~ (capmdat$mkt-capmdat$riskfree))
plot(capmdat$msft-capmdat$riskfree, capmdat$mkt-capmdat$riskfree, xlab ="Microsoft Return Less Risk Fre
abline(regmsft)
```



3 c

The estimates of the beta values change very little.

```
## regress with forced 0 intercept
regdis0=lm(capmdat$dis-capmdat$riskfree ~ (0 + capmdat$mkt-capmdat$riskfree))
regge0=lm(capmdat$ge-capmdat$riskfree ~ (0 + capmdat$mkt-capmdat$riskfree))
reggm0=lm(capmdat$gm-capmdat$riskfree ~ (0 + capmdat$mkt-capmdat$riskfree))
regibm0=lm(capmdat$ibm-capmdat$riskfree ~ (0 + capmdat$mkt-capmdat$riskfree))
regmsft0=lm(capmdat$msft-capmdat$riskfree ~ (0 + capmdat$mkt-capmdat$riskfree))
regxom0=lm(capmdat$xom-capmdat$riskfree ~ (0 + capmdat$mkt-capmdat$riskfree))

cof0=cbind(coefficients(regdis0), coefficients(regge0),coefficients(reggm0),coefficients(regibm0),coeff
rownames(cof0)=c("Beta")

#Slope with intercept = 0 compared to original slope
cof0
```

```
##           [,1]      [,2]     [,3]     [,4]     [,5]      [,6]
## Beta 0.8271039 0.8339432 1.187259 1.126572 1.257669 0.3564102
```

```
cof
```

```
##                   dis          ge          gm         ibm        msft
## Intercept -0.03289025 -0.03292527 -0.04428567 -0.02667249 -0.02677718
## Beta       0.86241707  0.86929395  1.23480687  1.15520974  1.28641872
##                   xom
## Intercept -0.02257146
```

```
## Beta       0.38064435
```

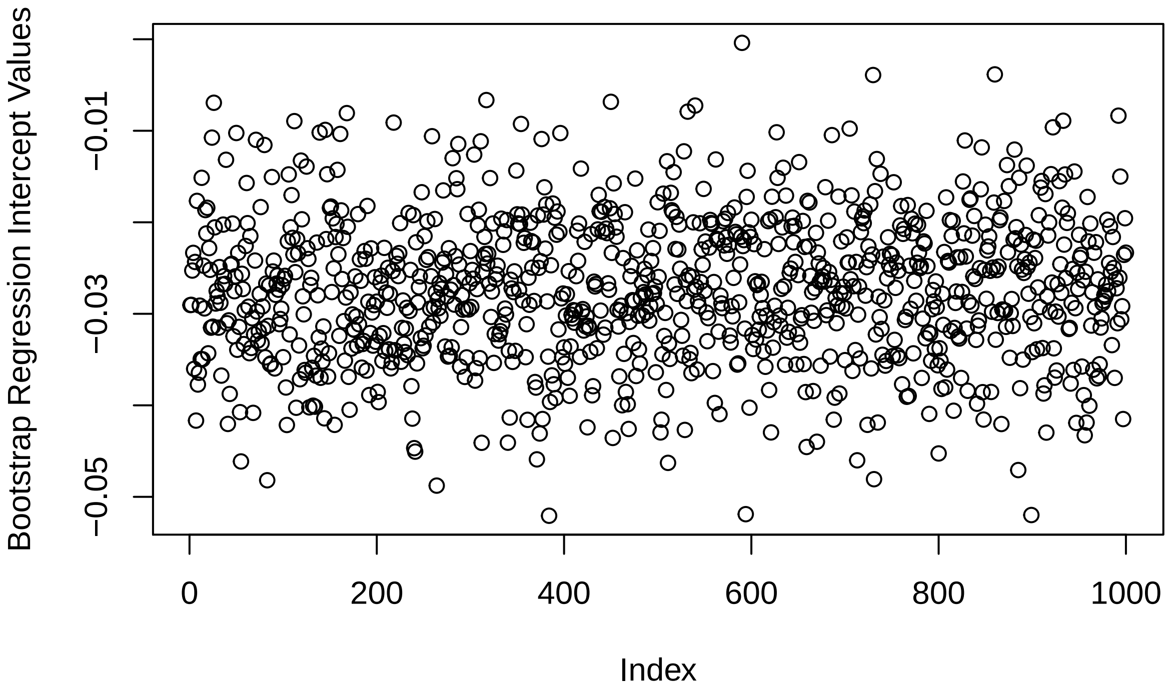
```
## confidence intervals
condis0=confint(regdis0, level=0.95)
conge0=confint(regge0, level=0.95)
congm0=confint(reggm0, level=0.95)
conibm0=confint(regibm0, level=0.95)
conmsft0=confint(regmsft0, level=0.95)
conxom0=confint(regxom0, level=0.95)
conint=cbind(condis0, conge0, congm0, conibm0, conmsft0, conxom0)
rownames(conint)=c("Confidence Intervals")
conint
```

```
##                          2.5 %   97.5 %     2.5 %   97.5 %     2.5 %
## Confidence Intervals 0.5498112 1.104397 0.6064239 1.061463 0.7631819
##                        97.5 %     2.5 %   97.5 %     2.5 %   97.5 %
## Confidence Intervals 1.611336 0.8569335 1.396211 0.9256323 1.589706
##                          2.5 %    97.5 %
## Confidence Intervals 0.1540818 0.5587385
```

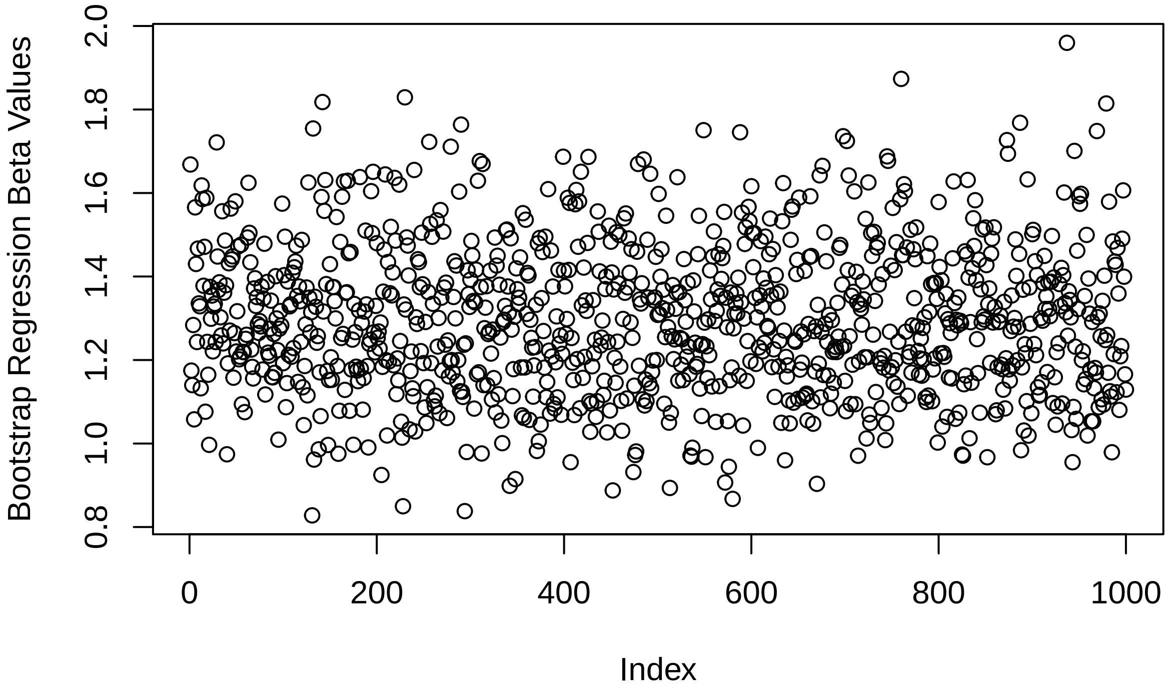
3 d

Our the betas, alphas, and confidence interval lengths all seem to be relatively stable accross all of the iterations.

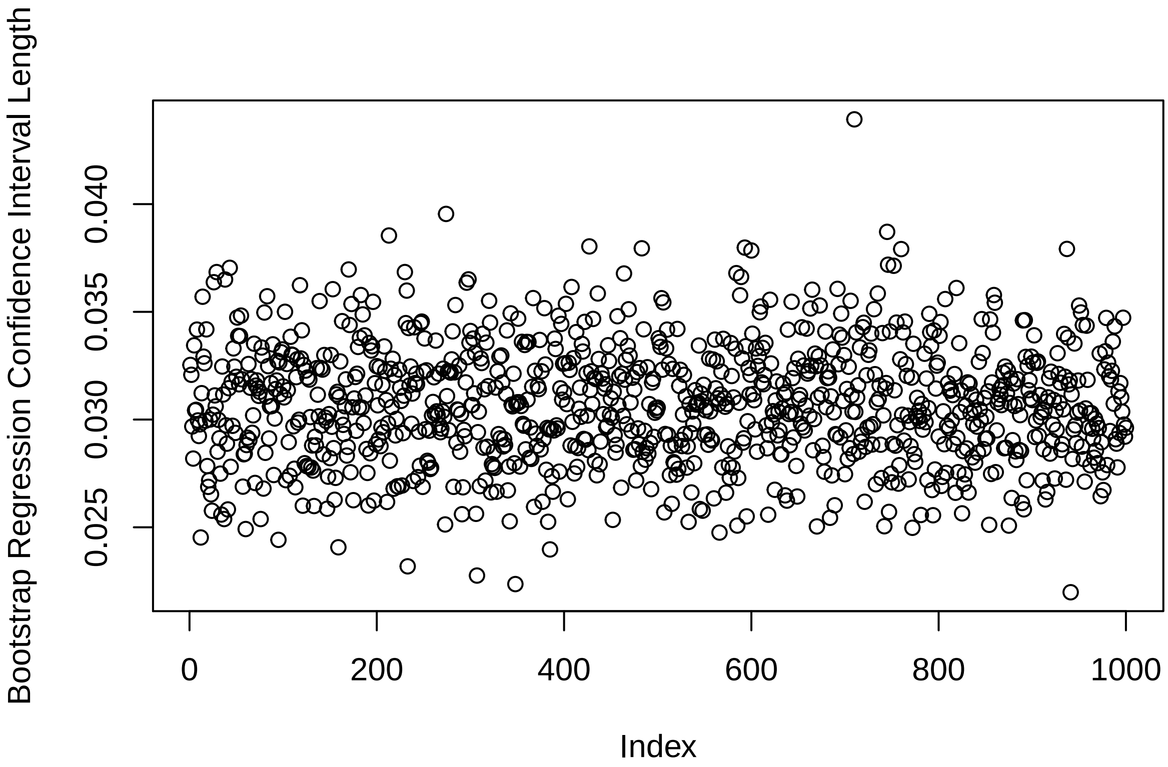
```
##bootstrapping microsoft
library(dplyr)
msfta=numeric(1000)
msftb=numeric(1000)
msftdaup=numeric(1000)
msftdbup=numeric(1000)
msftdadown=numeric(1000)
msftdbdown=numeric(1000)
for(i in 1:1000){
    samp=sample_n(capmdat, replace=T, 132)
    bootlm=lm(samp$msft-samp$riskfree ~ (samp$mkt - samp$riskfree))
    cofboot=cbind(coefficients(bootlm))
    msfta[i]=cofboot[1]
    msftb[i]=cofboot[2]
    x=confint(bootlm)
    msftdaup[i]=x[1,2]
    msftdbup[i]=x[2,2]
    msftdadown[i]=x[1,1]
    msftdbdown[i]=x[1,2]
}

plot(msfta, ylab="Bootstrap Regression Intercept Values")
```




```r
plot(msftb, ylab="Bootstrap Regression Beta Values")
```




```
plot(msftdaup-msftdadown, ylab="Bootstrap Regression Confidence Interval Length")
```



3 e

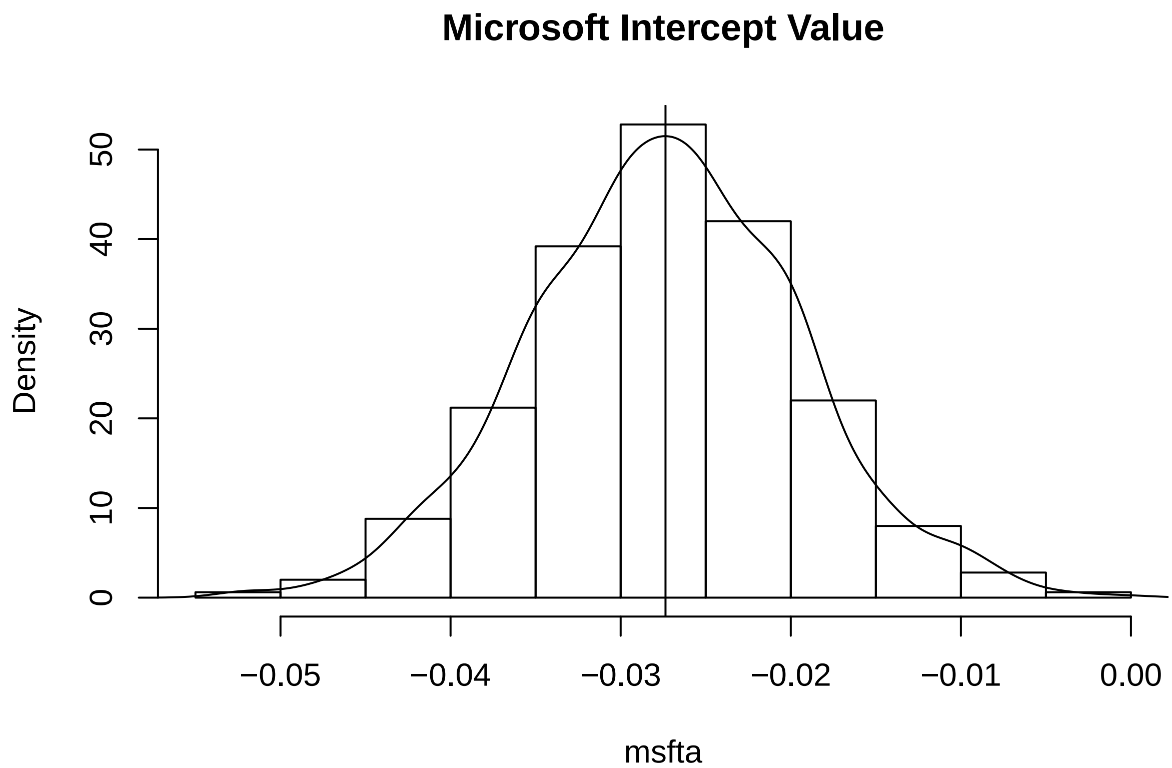
```r
mean(msfta)
```

```
## [1] -0.02736325
```

```r
"alpha mean"
```

```
## [1] "alpha mean"
```

```r
hist(msfta,probability = T, main="Microsoft Intercept Value" )
lines(density(msfta))
abline(v = mean(msfta))
```




```r
sd(msfta)
```

```
## [1] 0.007861395
```

```r
"alpha standard deviation"
```

```
## [1] "alpha standard deviation"
```

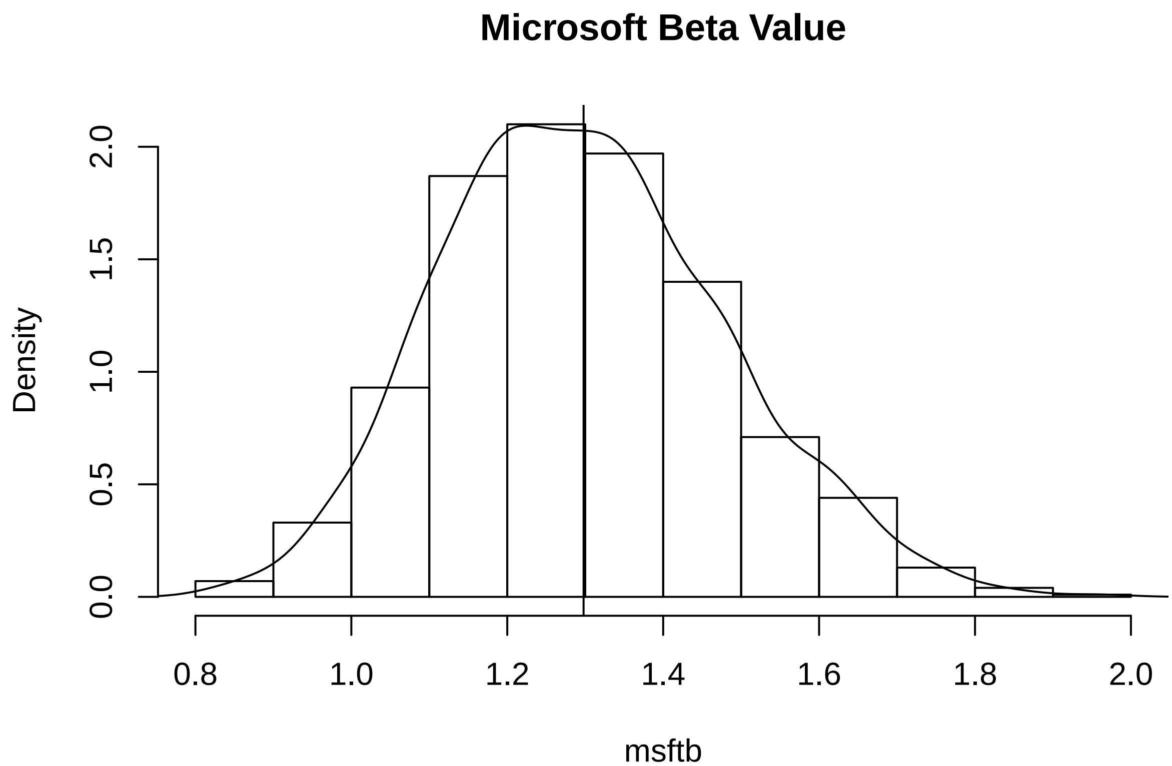
```r
mean(msftb)
```

```
## [1] 1.297885
```

```r
"beta mean"
```

```
## [1] "beta mean"
```

```r
hist(msftb,probability = T, main="Microsoft Beta Value")
lines(density(msftb))
abline(v = mean(msftb))
```




```
sd(msftb)
```

```
## [1] 0.1804736
```

```
"beta standard deviation"
```

```
## [1] "beta standard deviation"
```

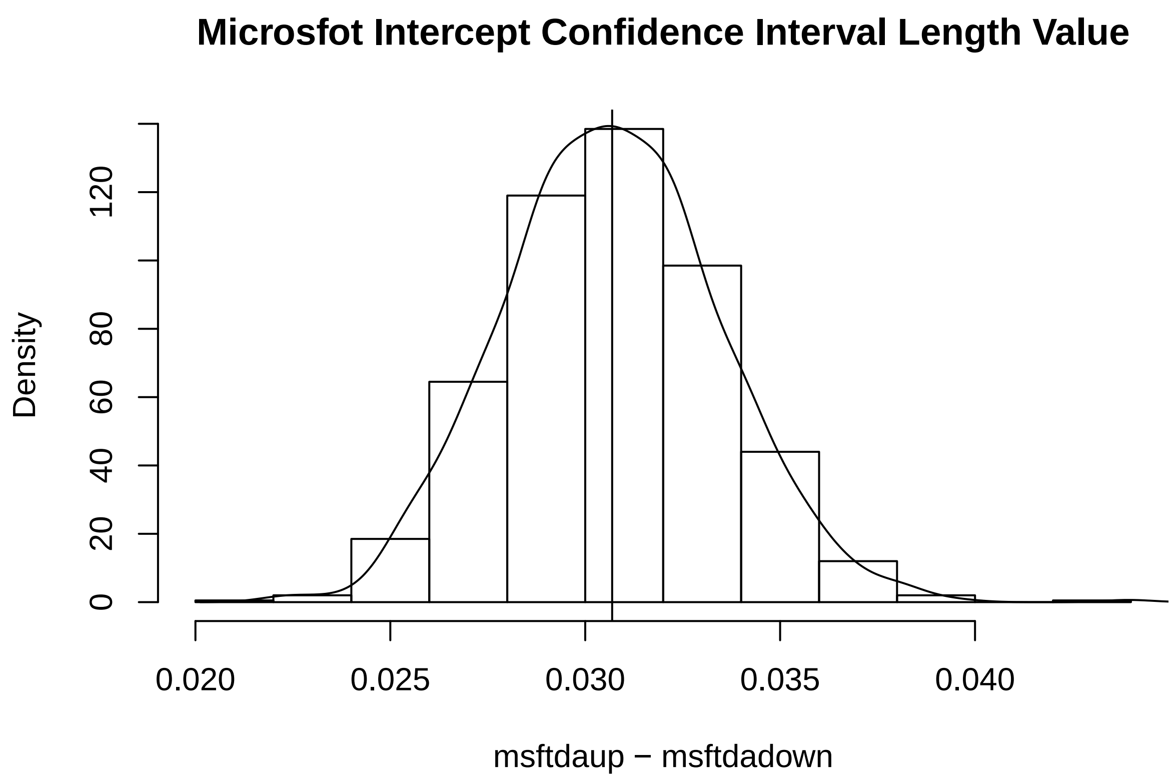
```
mean(msftdaup-msftdadown)
```

```
## [1] 0.03069078
```

```
"alpha confidence interval mean"
```

```
## [1] "alpha confidence interval mean"
```

```
hist(msftdaup-msftdadown,probability = T, main="Microsfot Intercept Confidence Interval Length Value")
lines(density(msftdaup-msftdadown))
abline(v = mean(msftdaup-msftdadown))
```




```
sd(msftdaup-msftdadown)
```

```
## [1] 0.00276287
```

```
"alpha confidence interval standard deviation"
```

```
## [1] "alpha confidence interval standard deviation"
```

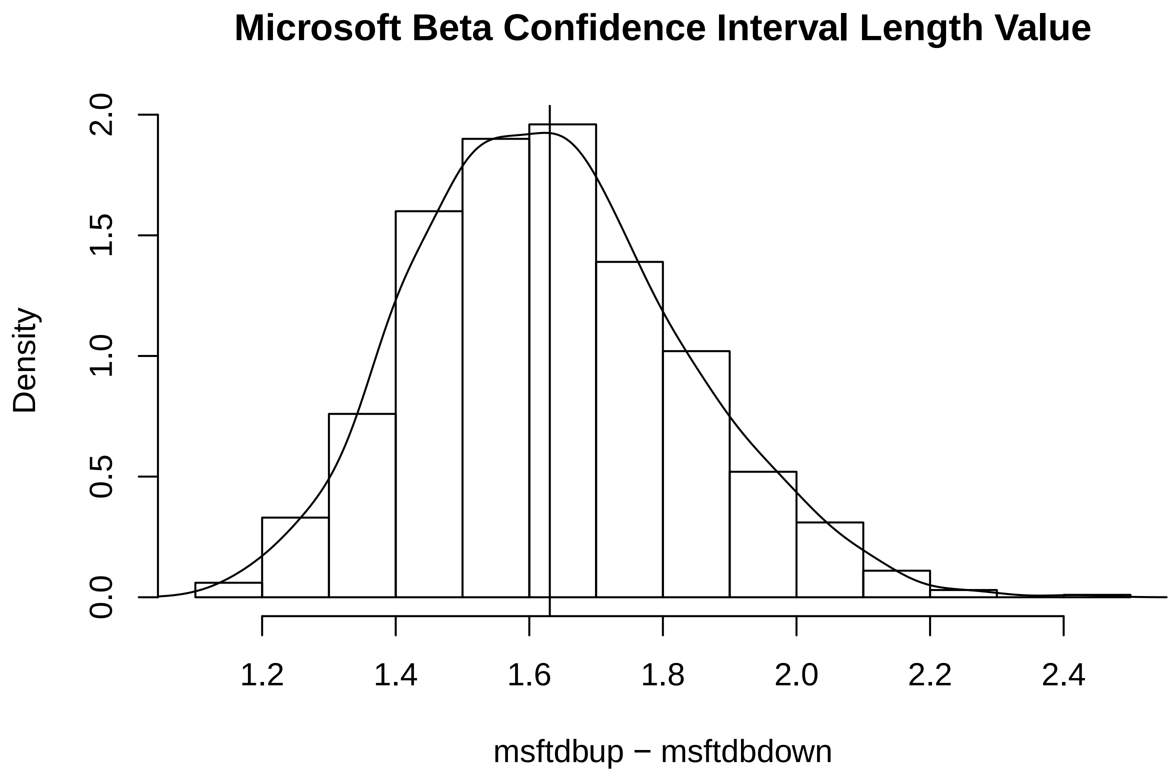
```
mean(msftdbup-msftdbdown)
```

```
## [1] 1.630666
```

```
"beta confidence interval mean"
```

```
## [1] "beta confidence interval mean"
```

```
hist(msftdbup-msftdbdown,probability = T, main="Microsoft Beta Confidence Interval Length Value")
lines(density(msftdbup-msftdbdown))
abline(v = mean(msftdbup-msftdbdown))
```




```r
sd(msftdbup-msftdbdown)
```

```
## [1] 0.2000027
```

```r
"beta confidence interval standard deviation"
```

```
## [1] "beta confidence interval standard deviation"
```

3 f

The results of the bootstrap and original regression results are very close to each other. Forced to choose between the two we would favor the boostrapped results since the boostrapping process tells us more about the structure of the data and produces a distribution of the estimate instead of a single number.

4 a

$$P(C+ \mid T+) = \frac{P(T+ \mid C+)P(C+)}{P(T+ \mid C+)P(C+) + P(T+ \mid C-)P(C-)}$$

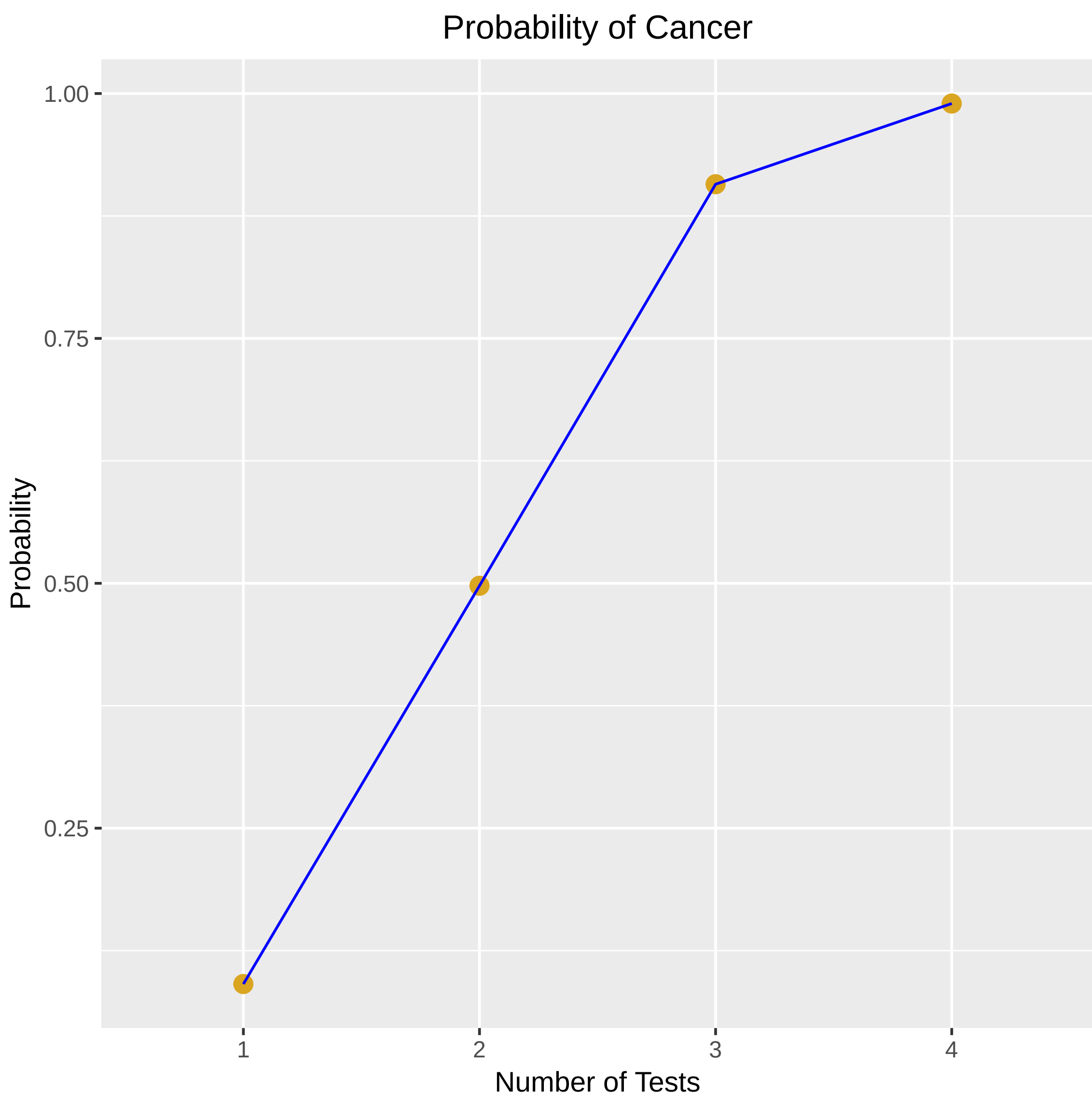
$$P(C+ \mid T+) = \frac{.99 * .01}{.99 * .01 + .1 * .99}$$

$$P(C+ \mid T+) = \frac{.99 * .01}{.99 * .01 + .1 * .99}$$

$$P(C+ \mid T+) = \frac{1}{11} = 9.091\%$$

4 b

```r
library(ggplot2)
probs = c()
curprob = .01
#probs = append(probs,curprob)

while(curprob < .95){
  curprob = (.99*curprob)/(.99*curprob+.1*(1-curprob))
  #print(curprob)
  probs = append(probs,curprob)
}
probs = data.frame(probability=probs)
probs$id = row.names(probs)

ggplot(probs, aes(id, probability,group = 1)) +
  geom_point(col='goldenrod',size=3) +
  geom_line(col='blue') +
  xlab('Number of Tests') + ylab("Probability") +
  ggtitle('Probability of Cancer') +
  theme(plot.title = element_text(hjust = 0.5))
```



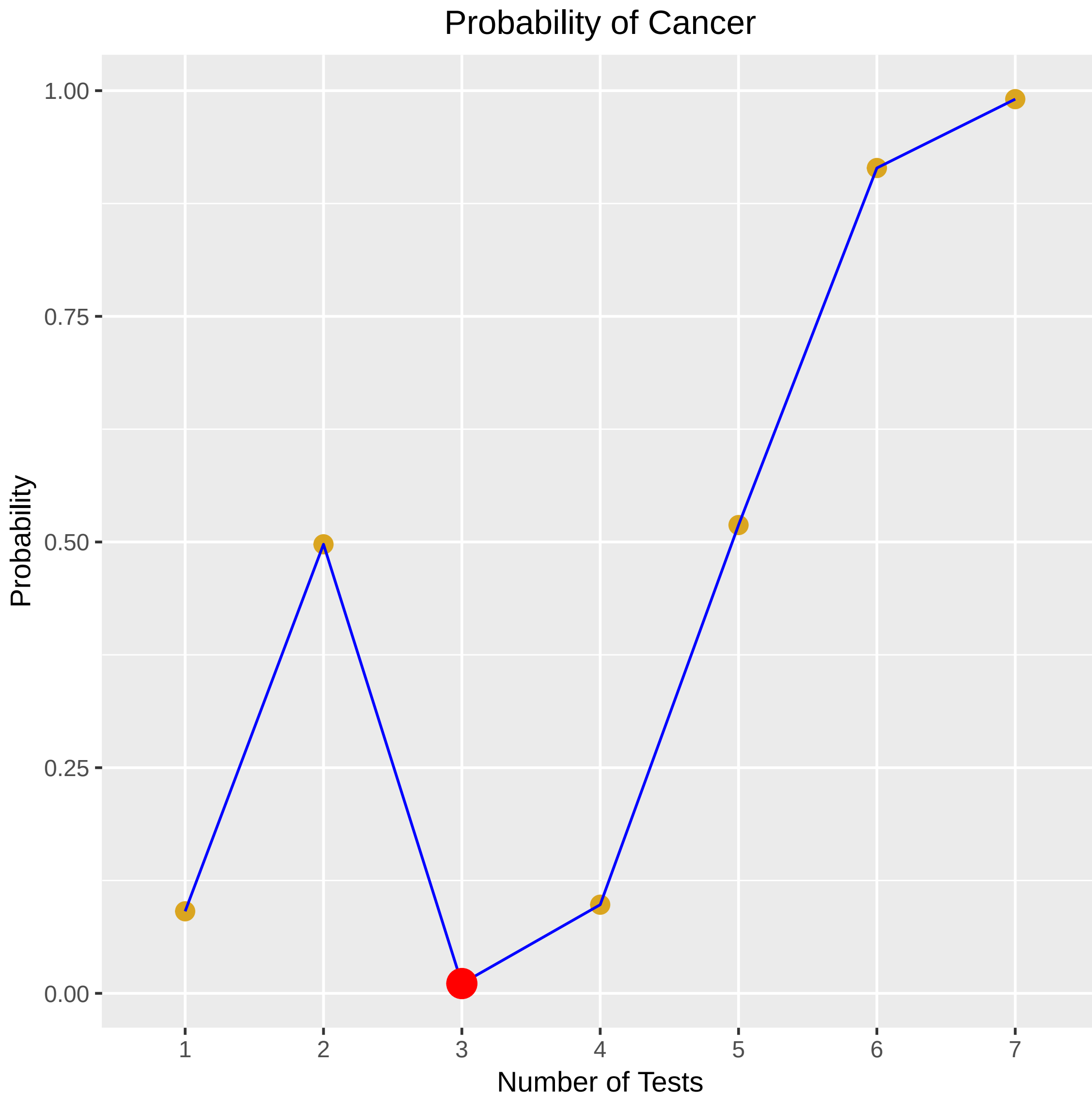
It will only take 4 readings to know that there is a 95% chance that cancer is present. We can see from that graph that the probability increases quickly after the first few trials.

4 c

$$P(C+\,|\,T-) = \frac{P(T-\,|\,C+)P(C+)}{P(T-\,|\,C+)P(C+) + P(T-\,|\,C-)P(C-)}$$

```
probs = c()
curprob = .01
#probs = append(probs,curprob)
trialnum = 1

while(curprob < .95){
  if(trialnum == 3){
    curprob = (.01*curprob)/(.01*curprob+.9*(1-curprob))
  }else{
    curprob = (.99*curprob)/(.99*curprob+.1*(1-curprob))
```

```
  }

  #print(curprob)
  probs = append(probs,curprob)
  trialnum = trialnum+1
}

probs = data.frame(probability=probs)
probs$id = row.names(probs)

ggplot(probs, aes(id, probability,group = 1)) +
  xlab('Number of Tests') + ylab("Probability") +
  ggtitle('Probability of Cancer') +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_point(col='goldenrod',size=3) +
  geom_line(col='blue') +
  geom_point(data=probs[3, ], colour="red", size=5)
```



We still have to have four tests after the false negative to be 95% sure cancer is present. The third reading brings us near our starting probability for cancer, so we from that point we only have to do as many tests as it took initially.