R_Assignment

1: Identify distributions from data.

Task:To identify which distribution generated which column as well as estimating the parameters of each distributions.

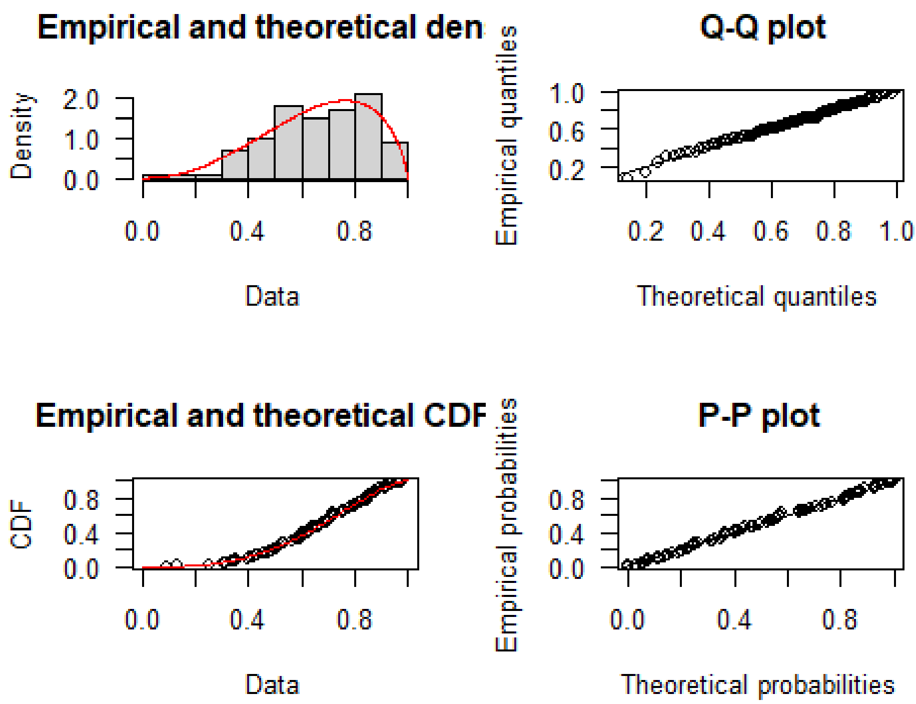
Comment: Initially, I will import the data set from csv file. And, I will introduce each column separately.

```
df1 <- read.csv("C:\\Users\\Pramod\\Downloads\\R_assessment1_data (1).csv")
x1 <- df1[,1]
x2 <- df1[,2]
x3 <- df1[,3]
x4 <- df1[,4]
x5 <- df1[,5]
x6 <- df1[,6]
```

Beta Distribution

```
summary(x1)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    2.00    2.52    3.00    5.00

summary(x2)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.0150   0.6955   1.9600  13.3188   5.1930 573.7003

summary(x3)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.08763 0.51985 0.67764 0.65902 0.82009 0.97935

summary(x4)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01116 2.69697 4.52534 4.68666 6.65404 9.93440

summary(x5)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01035 0.07053 0.27317 0.37214 0.61987 1.54673

summary(x6)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.00    7.00   10.00    9.17   11.00   17.00
```

```
library(fitdistrplus)

## Loading required package: MASS

## Loading required package: survival

fit_beta <- fitdist(x3,"beta")
plot(fit_beta, las = 1)
```



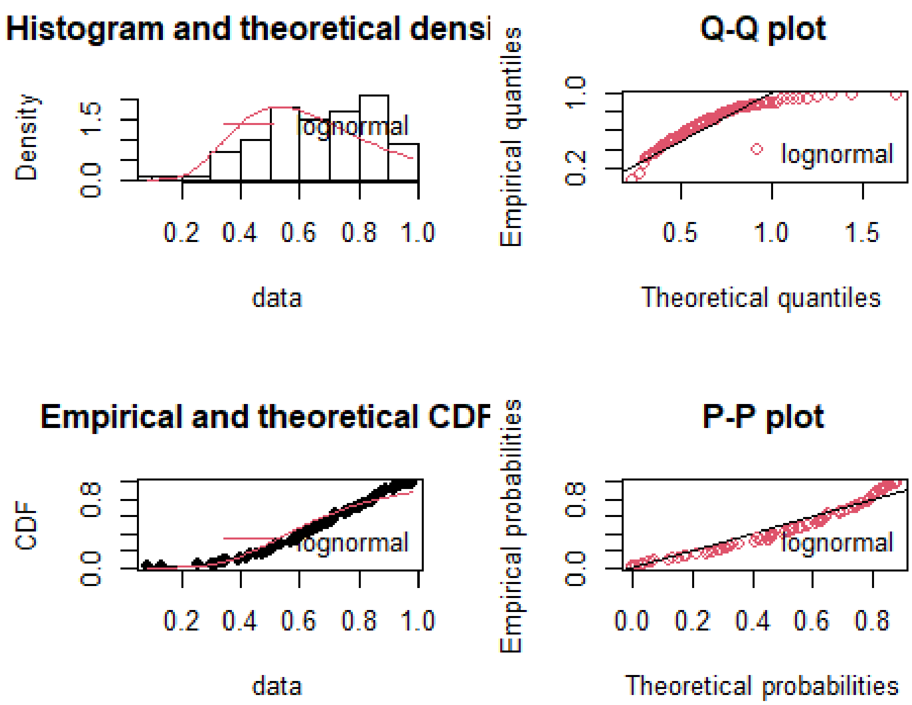

```
summary(fit_beta)

## Fitting of the distribution ' beta ' by maximum likelihood
## Parameters :
##         estimate Std. Error
## shape1 3.245912  0.4556391
## shape2 1.694080  0.2220968
## Loglikelihood:  28.70409   AIC:  -53.40819   BIC:  -48.19785
## Correlation matrix:
##           shape1    shape2
## shape1 1.0000000 0.8036855
## shape2 0.8036855 1.0000000
```

Comment: :Since the data in x1,x2,x4,x5,x6 lies outside of beta distribution condition,hence we use beta distribution for x3 only. Moreover, Beta distribution perfectly fit the x3 data.

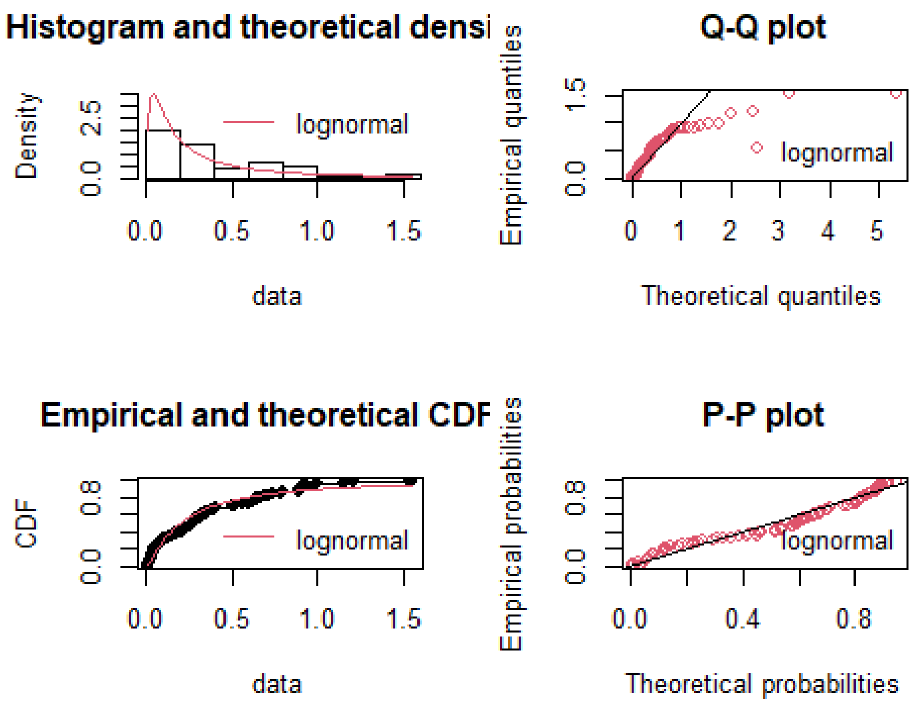
Log normal distribution.

```
fln <- fitdist(x3, "lnorm")
par(mfrow = c(2, 2))
```

```
plot.legend <- c("lognormal")
denscomp(list(fln), legendtext = plot.legend)
qqcomp(list(fln), legendtext = plot.legend)
cdfcomp(list(fln), legendtext = plot.legend)
ppcomp(list(fln), legendtext = plot.legend)
```




```
summary(fln)

## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters :
##             estimate Std. Error
## meanlog -0.4772386 0.03896419
## sdlog    0.3896419 0.02755102
## Loglikelihood:  0.08272371   AIC:  3.834553   BIC:  9.044893
## Correlation matrix:
##                meanlog        sdlog
## meanlog  1.00000e+00 -3.81385e-12
## sdlog   -3.81385e-12  1.00000e+00

fln <- fitdist(x5, "lnorm")
par(mfrow = c(2, 2))
plot.legend <- c("lognormal")
denscomp(list(fln), legendtext = plot.legend)
qqcomp(list(fln), legendtext = plot.legend)
cdfcomp(list(fln), legendtext = plot.legend)
ppcomp(list(fln), legendtext = plot.legend)
```

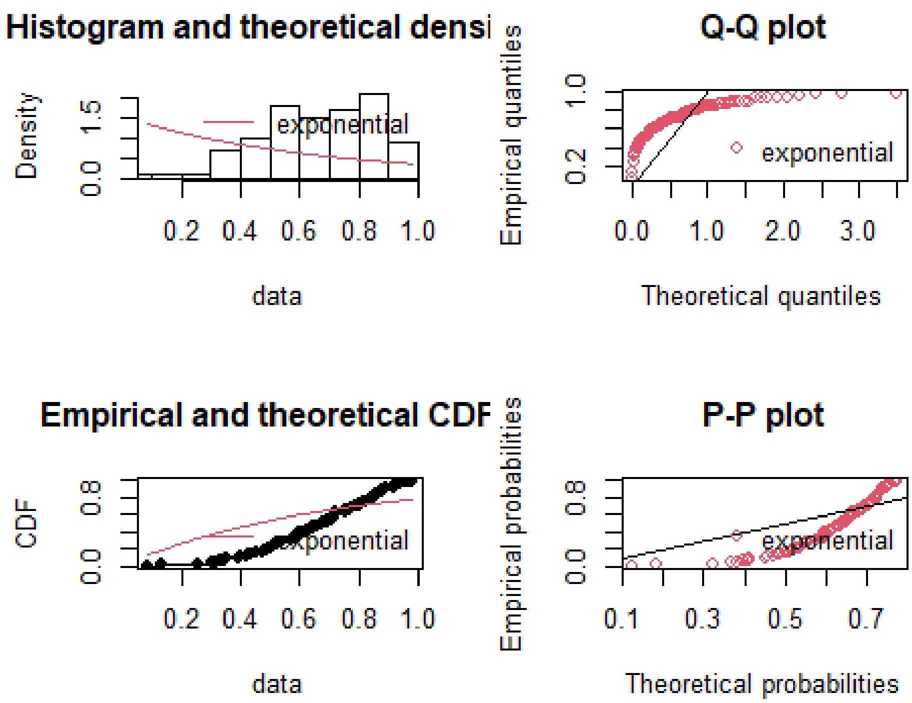



```
summary(fln)

## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters :
##           estimate Std. Error
## meanlog -1.603541 0.12730323
## sdlog    1.273032 0.09001672
## Loglikelihood:  -5.679936   AIC:  15.35987   BIC:  20.57021
## Correlation matrix:
##                meanlog         sdlog
## meanlog  1.000000e+00 -3.816753e-12
## sdlog   -3.816753e-12  1.000000e+00
```

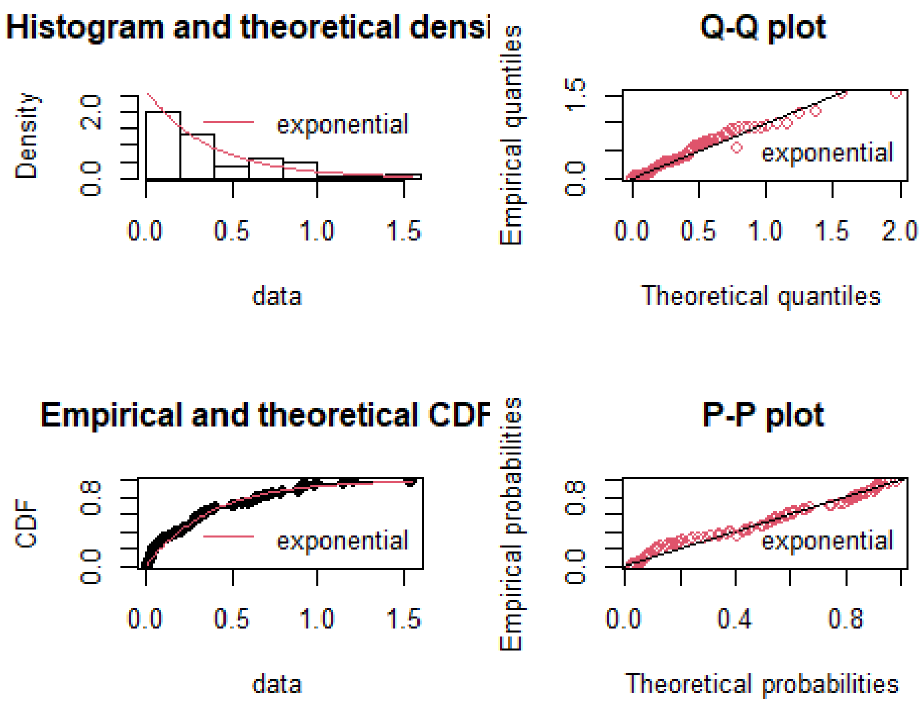
Comment: After a proper analysis with the data sets, it has observed that log normal distribution suited well enough with x3 and x5.

Exponential Distribution

```
fexp <- fitdist(x3,"exp")
par(mfrow = c(2, 2))
plot.legend <- c("exponential")
denscomp(list(fexp), legendtext = plot.legend)
qqcomp(list(fexp), legendtext = plot.legend)
cdfcomp(list(fexp), legendtext = plot.legend)
ppcomp(list(fexp), legendtext = plot.legend)
```




```
summary(fexp)

## Fitting of the distribution ' exp ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## rate 1.517397  0.1517396
## Loglikelihood:  -58.30039   AIC:  118.6008   BIC:  121.2059

fexp <- fitdist(x5,"exp")
par(mfrow = c(2, 2))
plot.legend <- c("exponential")
denscomp(list(fexp), legendtext = plot.legend)
qqcomp(list(fexp), legendtext = plot.legend)
cdfcomp(list(fexp), legendtext = plot.legend)
ppcomp(list(fexp), legendtext = plot.legend)
```

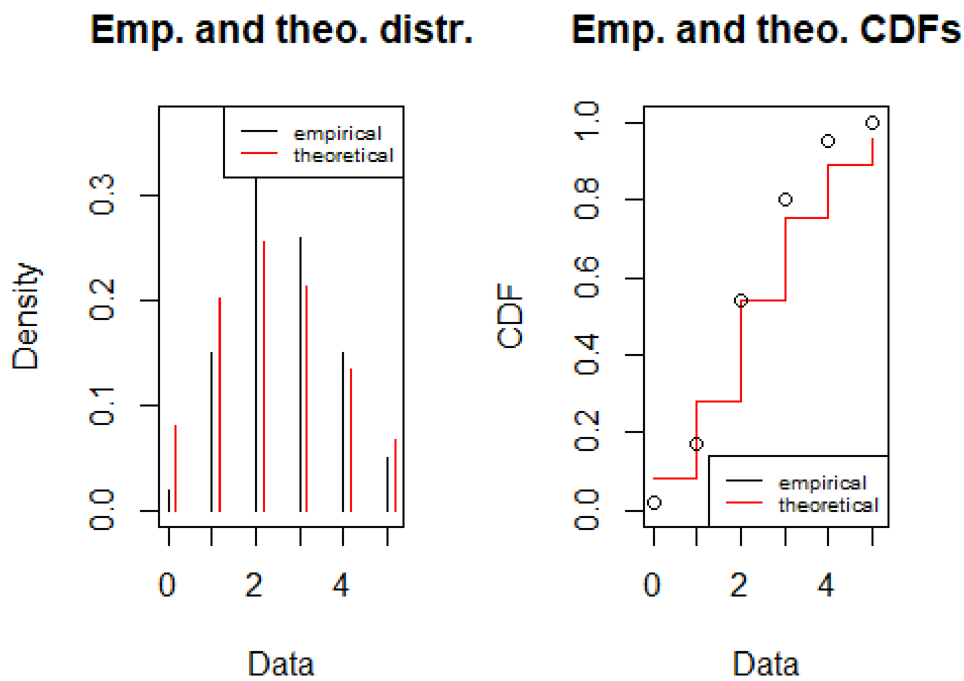



```
summary(fexp)

## Fitting of the distribution ' exp ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## rate 2.687142  0.2687142
## Loglikelihood:  -1.152167   AIC:  4.304335   BIC:  6.909505
```

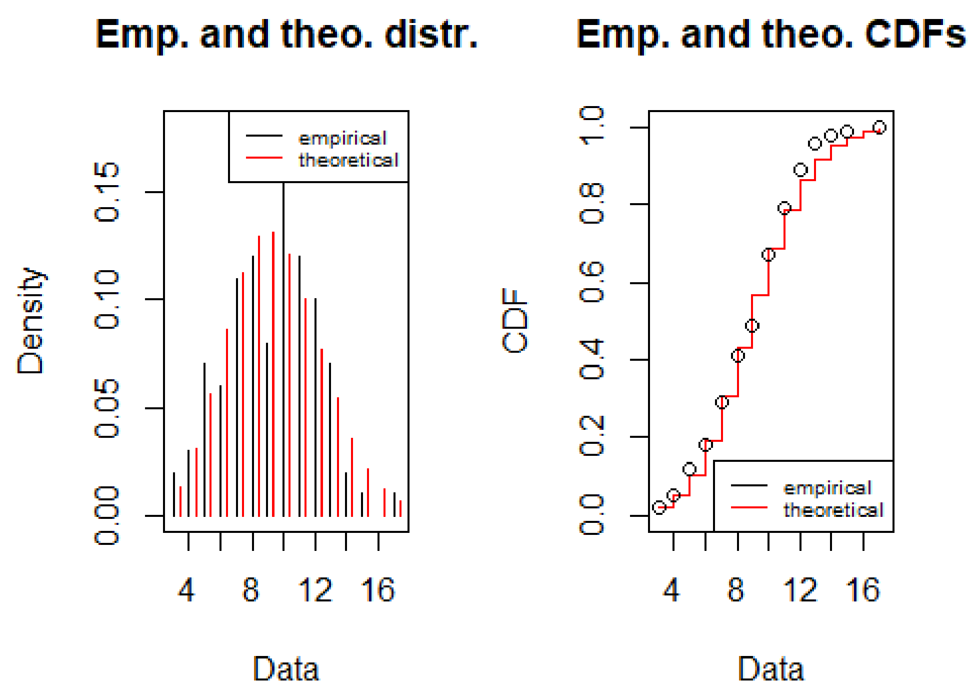
Note: For exponential distribution data set x3 and x5 produce better results as compare to other data sets, hence exponential distribution generated x3 and x5.

Poisson Distribution

```
fpois<- fitdist(x1, "pois")
plot(fpois)
```




```
summary(fpois)

## Fitting of the distribution ' pois ' by maximum likelihood
## Parameters :
##         estimate Std. Error
## lambda      2.52  0.1587451
## Loglikelihood:  -162.9272   AIC:  327.8544   BIC:  330.4596

fpois<- fitdist(x6, "pois")
plot(fpois)
```

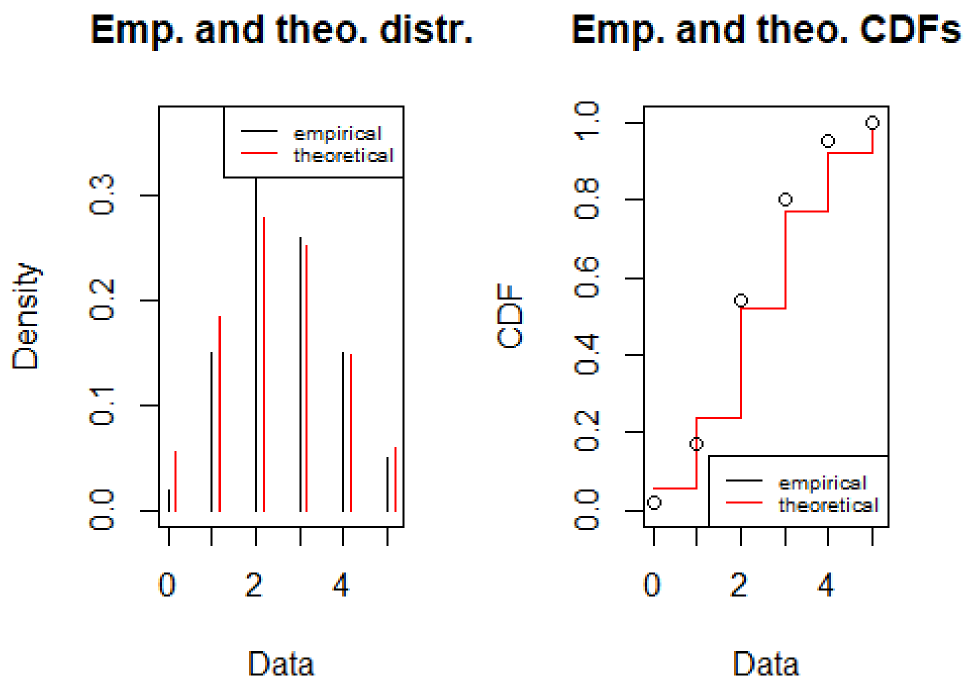



```
summary(fpois)

## Fitting of the distribution ' pois ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## lambda     9.17  0.3028201
## Loglikelihood:  -245.969   AIC:  493.9379   BIC:  496.5431
```

Comment: Poisson Distribution only fit to the data set x1 and x6 and it well fitted with x1.

Binomial Distribution

```
fitBinom=fitdist(x1, dist="binom", fix.arg=list(size=10),
start=list(prob=0.5))
plot(fitBinom)
```

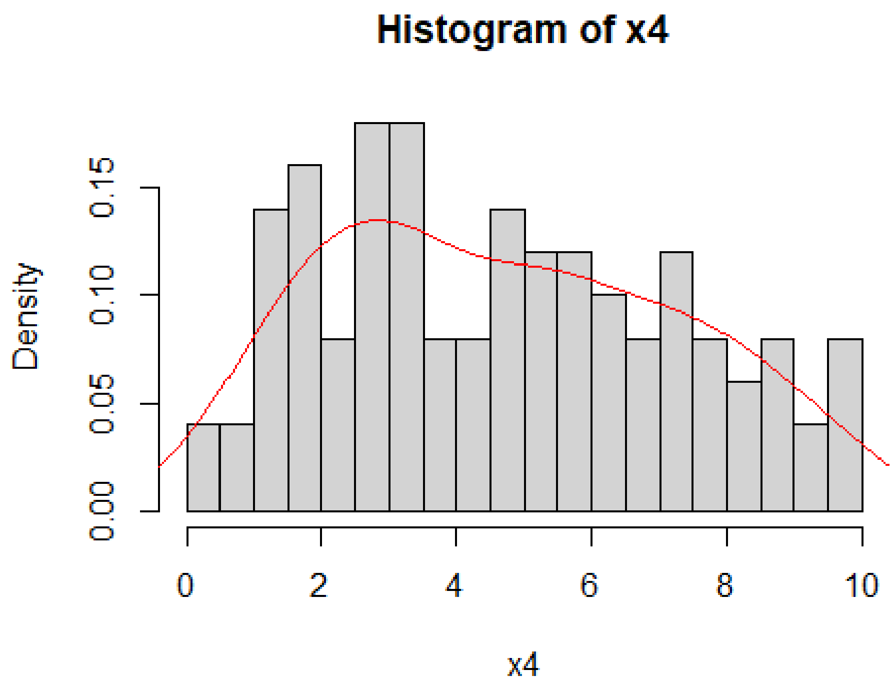



```
summary(fitBinom)

## Fitting of the distribution ' binom ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## prob 0.2520003 0.01372923
## Fixed parameters:
##       value
## size    10
## Loglikelihood:  -156.8079   AIC:  315.6158   BIC:  318.2209
```

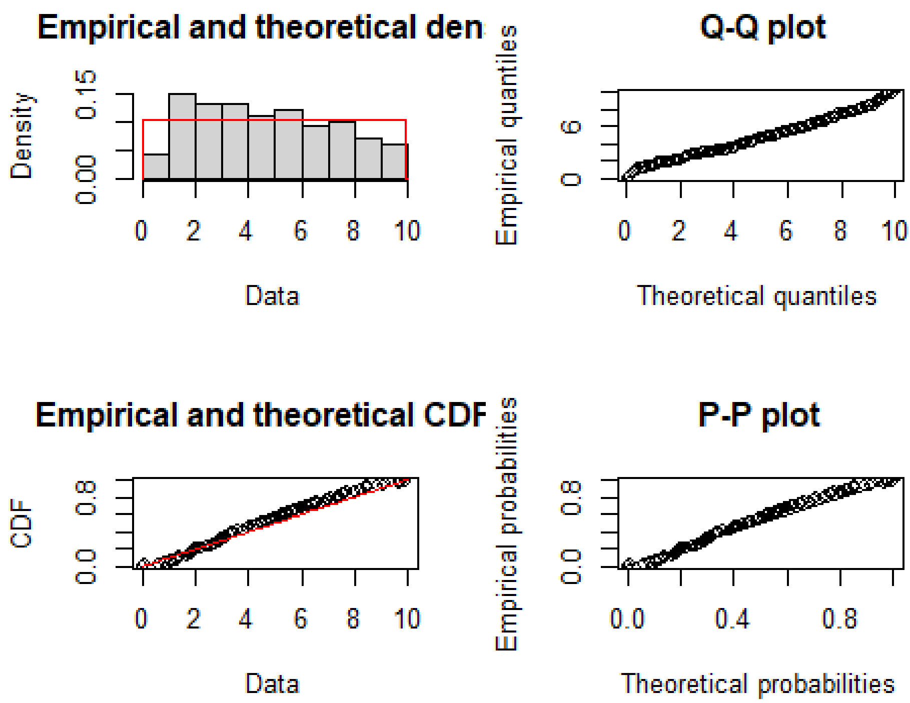
Comment: After checking the binomial distribution to each of the data set, it was found that, it generate data set x1 and x6. Additionally, it fitted well with small data set(like n=10)

Uniform distribution

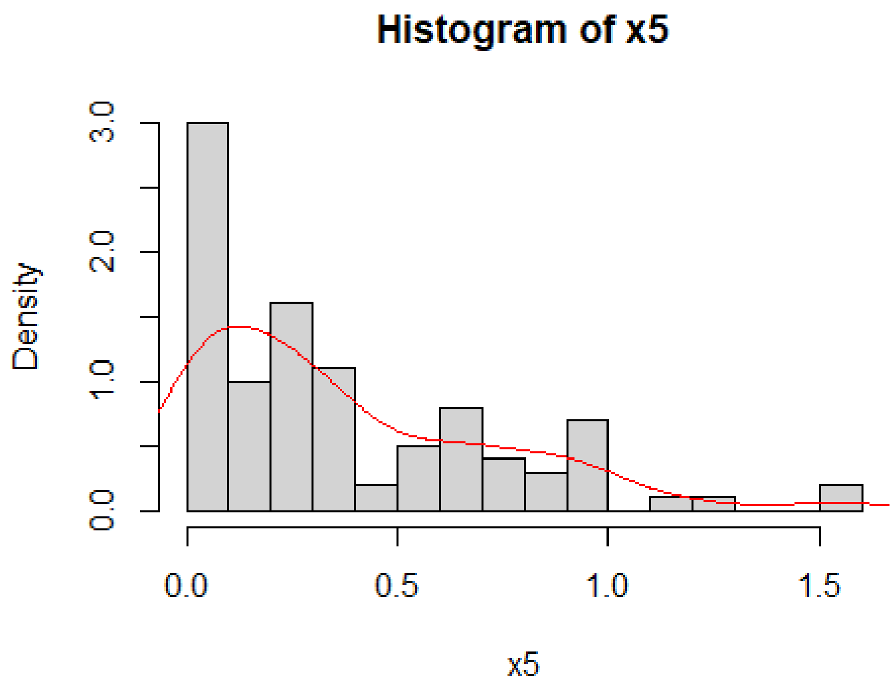
```
hist(x4,freq=FALSE,breaks=20)
lines(density(x4),col='red')
```



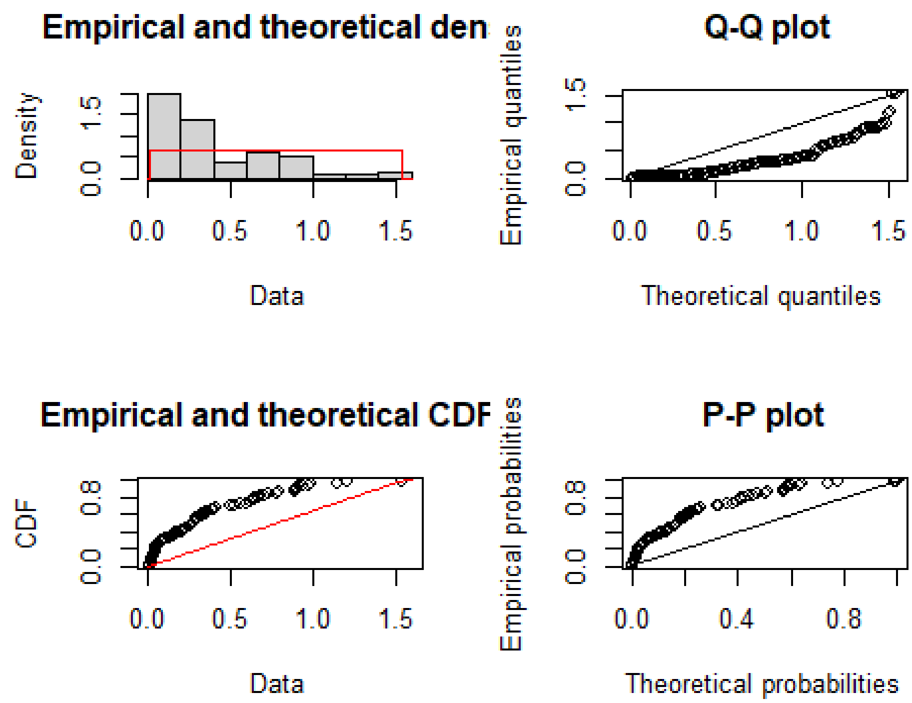

```
fit_x4 <- fitdist(x4,"unif")
mlex4 <-mledist(x4,"unif")
mle_x4 <- mlex4$estimate
plot(fit_x4)
```




```
summary(fit_x4)

## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## min 0.01115655         NA
## max 9.93439525         NA
## Loglikelihood:  -229.4879   AIC:  462.9759   BIC:  468.1862
## Correlation matrix:
## [1] NA

hist(x5,freq=FALSE,breaks=20)
lines(density(x5),col='red')
```




```r
fit_x4 <- fitdist(x5,"unif")
mlex4 <-mledist(x5,"unif")
mle_x4 <- mlex4$estimate
plot(fit_x4)
```




```
summary(fit_x4)

## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## min 0.01034691         NA
## max 1.54672983         NA
## Loglikelihood:  -42.94309   AIC:  89.88618   BIC:  95.09652
## Correlation matrix:
## [1] NA
```

Comment: Uniform distribution generate the data set x4 and x5.

2: Likelihood ratio test.

5: Use classical tests on given data set and observe their behaviour with given parameters.

```
library("readxl")
df2 <- read_excel("C:\\Users\\Pramod\\Downloads\\R_assessment1_norm
(1).xlsx")

## New names:
## * `` -> ...1

x <- df2[,2]
y <- df2[,3]
```

```
mu0 <- 10
mu1 <- 15
alpha <- 0.05
sample_gauss0 <- rnorm(100, mean = mu0, sd = 100)
sample_gauss1 <- rnorm(100, mean = mu1, sd = 100)
c_1<-qnorm(p=1-alpha, mean=mu0, sd=1/sqrt(100))
Reject0_v0 <- (mean(sample_gauss0)>=c_1)
Reject1_v1 <- (mean(sample_gauss1)>=c_1)
log_LRT_stat0 <- sum((sample_gauss0-mu1)**2-(sample_gauss0-mu0)**2)/2
log_LRT_stat1 <- sum((sample_gauss1-mu1)**2-(sample_gauss1-mu0)**2)/2
mean_H0 = (mu0-mu1)**2*100/2
sd_H0 = sqrt(100)*abs(mu0-mu1)
log_k_alpha <- qnorm(p=alpha, mean=mean_H0, sd=sd_H0)
Reject0 <- (log_LRT_stat0<=log_k_alpha)
Reject1 <- (log_LRT_stat1<=log_k_alpha)
```

Comment:For the data from x we fail to reject the null hypothesis.

```
mu0 <- 10
mu1 <- 15
alpha <- 0.05
sample_gauss0_y <- rnorm(100, mean = mu0, sd = 100)
sample_gauss1_y <- rnorm(100, mean = mu1, sd = 100)
c_1<-qnorm(p=1-alpha, mean=mu0, sd=1/sqrt(100))
Reject0_v0 <- (mean(sample_gauss0_y)>=c_1)
Reject1_v1 <- (mean(sample_gauss1_y)>=c_1)
log_LRT_stat0_y <- sum((sample_gauss0_y-mu1)**2-(sample_gauss0_y-mu0)**2)/2
log_LRT_stat1_y <- sum((sample_gauss1_y-mu1)**2-(sample_gauss1_y-mu0)**2)/2
mean_H0_y = (mu0-mu1)**2*100/2
sd_H0_y = sqrt(100)*abs(mu0-mu1)
log_k_alpha_y <- qnorm(p=alpha, mean=mean_H0_y, sd=sd_H0_y)
Reject0 <- (log_LRT_stat0_y<=log_k_alpha_y)
Reject1 <- (log_LRT_stat1_y<=log_k_alpha_y)
```

Comment: Here again we fail to reject the null hypothesis.

6:Compute the p-values

```
p_value0 <- pnorm(log_LRT_stat0, mean=mean_H0, sd=sd_H0)
p_value1 <- pnorm(log_LRT_stat1, mean=mean_H0, sd=sd_H0)
```

Comment: here the p values are 1 and 0 respectively.

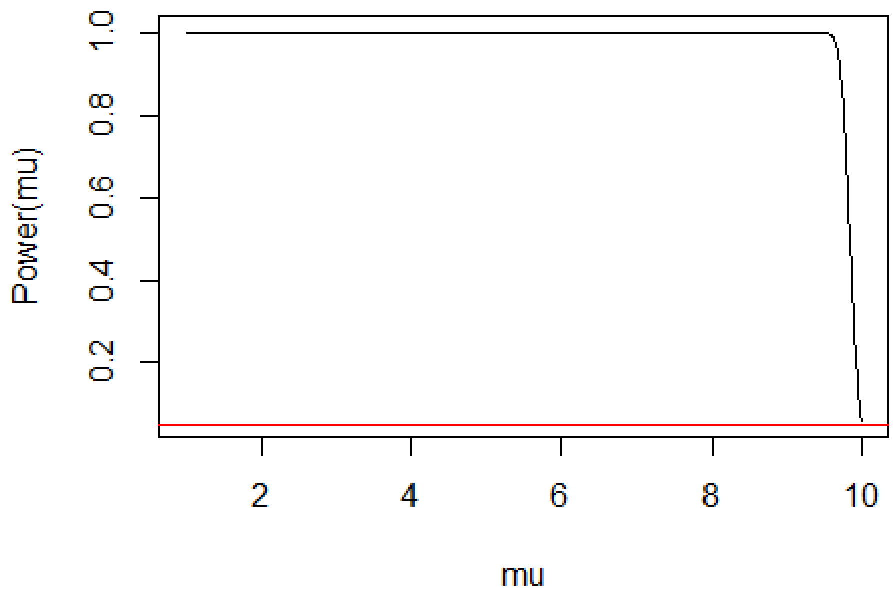
```
p_value0 <- pnorm(log_LRT_stat0_y, mean=mean_H0_y, sd=sd_H0_y)
p_value1 <- pnorm(log_LRT_stat1_y, mean=mean_H0_y, sd=sd_H0_y)
```

Comment: here the p values are 0.99(1) and 1 resp.

7:Power function test

```
Power <- function(mu1)
{
  mean_H1 = -(mu0-mu1)**2*100/2
```

```r
  sd_H1 = sqrt(100)*abs(mu0-mu1)
  mean_H0 = (mu0-mu1)**2*100/2
  sd_H0 = sqrt(100)*abs(mu0-mu1)
  log_k_alpha <- qnorm(p=alpha, mean=mean_H0, sd=sd_H0)
  res <- pnorm(log_k_alpha, mean=mean_H1, sd=sd_H1)
  return(res)
}
mu <- seq(from=1.0001, to=10, by=0.01)
plot(mu, Power(mu), type = "l")
abline(a=alpha, b=0, col="red")
```



Comment: Power function graph for given data set.

8: More powerful test for same alpha is given by:

estimate sample size via power analysis

from statsmodels.stats.power import TTestIndPower

parameters for power analysis

effect = 0.7

alpha = 0.05

power = 0.7

```
# perform power analysis
analysis = TTestIndPower()
result = analysis.solve_power(effect, power=power, nobs1=None, ratio=1.0, alpha=alpha)
print('Sample Size: %.3f' % result)
```

2. Likelihood ratio test:→

1. Given $\Lambda(x) = \frac{L(\mu_0; x)}{L(\mu_1; x)}$

Hence LRT is given by:-

$$\Lambda(x) = \begin{cases} 1 & : \text{if } \bar{x} \le \mu_0 \\ \frac{L(\mu)}{L(\bar{x})} & : \text{if } \bar{x} > \mu_0 \end{cases}$$

Now the required expression is:-

$$\Lambda(x) = \frac{L(\mu_0;x)}{L(\bar{x};x)} = \frac{(2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{1}{2}\sigma^2 \sum (x_i-\mu_0)^2}}{(2\pi\sigma^2)^{-n/2} e^{-1/2\sigma^2 \sum (x_i-\bar{x})^2}}$$

$$\Lambda(x) = \frac{e^{-1/2\sigma^2 \sum(x_i-\mu_0)^2}}{e^{-1/2\sigma^2 \sum(x_i-\bar{x})^2}} = e^{-n/2\sigma^2[\mu_0^2+\bar{x}^2 - 2\bar{x}\mu_0]}$$

$$\Lambda(x) = \exp\left[-\frac{n}{2\sigma^2}\cdot(\mu_0-\bar{x})^2\right] \le K$$

$$-\frac{n}{2\sigma^2}(\bar{x}-\mu_0)^2 \le \log K$$

P.T.O

$$\Rightarrow \frac{(\bar{x}-\mu_0)}{\sigma/\sqrt{n}} \le \sqrt{-2\log k} \le k'$$

Hence

LRT rejects if $\boxed{\frac{\bar{x}-\mu_0}{\sigma/\sqrt{n}} \geq k''}$

2.

Proof :- Let say $\bar{X} \sim N(\mu_0, \frac{\sigma^2}{n})$ and $\bar{x} = \frac{\sum x_i}{n}$ — ①

LRT is given by :-

$$\Lambda(x) = \frac{L(\mu_0; x)}{L(\mu_1; x)} = \frac{(2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}\sum(x_i-\mu_0)^2}}{(2\pi\sigma^2)^{-n/2} e^{-1/2\sigma^2 \sum(x_i-\mu_1)^2}}$$

$$\Lambda(x) = e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(2x_i(\mu_1-\mu_0)+(\mu_0^2-\mu_1^2))}$$

taking log both sides

$$\log\Lambda(x) = -\frac{n}{2\sigma^2}(\mu_0^2-\mu_1^2) - \frac{2n\bar{x}}{2\sigma^2}(\mu_1-\mu_0)$$

$$\log\Lambda(x) + \frac{n}{2\sigma^2}(\mu_0^2-\mu_1^2) = \frac{n\bar{x}}{\sigma^2}(\mu_0-\mu_1)$$

Hence

$$\bar{x} = \frac{\sigma^2}{n(\mu_0-\mu_1)}\left[\log\Lambda(x) + \frac{n}{2\sigma^2}(\mu_0^2-\mu_1^2)\right]$$ — ②

equating ① and ②. We proved the required Statement.

3. • We have,

$$\frac{L(\mu_0; x)}{L(\mu_1 : x)} = e^{\frac{-1}{2\sigma^2} \Sigma (2x_i (\mu_1 - \mu_0) + (\mu_0^2 - \mu_1^2))}$$

$$\frac{L(\mu_0; x)}{L(\mu_1; x)} \leq A \text{ or } \log\left[\frac{L(\mu_0; x)}{L(\mu_1; x)}\right] \leq \log A$$

$$\Rightarrow \frac{-1}{2\sigma^2} \Sigma (2x_i (\mu_1 - \mu_0) + (\mu_0^2 - \mu_1^2)) \leq \log A$$

$$\bar{x} \geq \frac{\sigma^2 \log(A)}{n(\mu_1 - \mu_0)} + \frac{1}{2}(\mu_1 - \mu_0) = K$$

Hence

$$\boxed{\bar{x} \geq K}$$

• $P(\bar{X} > K | H_0) = 0.05$

$n = 100, \ \mu_0 = 10, \ \sigma^2 = 100$

$\bar{x} \sim N(10, 1)$

$$P\left(\frac{\bar{x} - 10}{\sqrt{1/100}} > \frac{K_\alpha - 10}{\sqrt{1/100}}\right) = 0.05$$

$\Rightarrow$

$$P\left(z \geq \frac{K - 10}{\sqrt{1/100}}\right) = 0.05$$

$$\Rightarrow \frac{K - 10}{\sqrt{1/100}} = 1.65 \Rightarrow \boxed{K = 10.52}$$

Date

Page

4. $$\left| \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \right| \geq 1.96$$

$$\Rightarrow -1.96 \leq \frac{\bar{x} - 10}{1} \leq 1.96$$

$$\Rightarrow 8.04 \leq \bar{x} \leq 11.96$$

Hence the value of $K_A = 8.04$.