PBSR Assignment 2

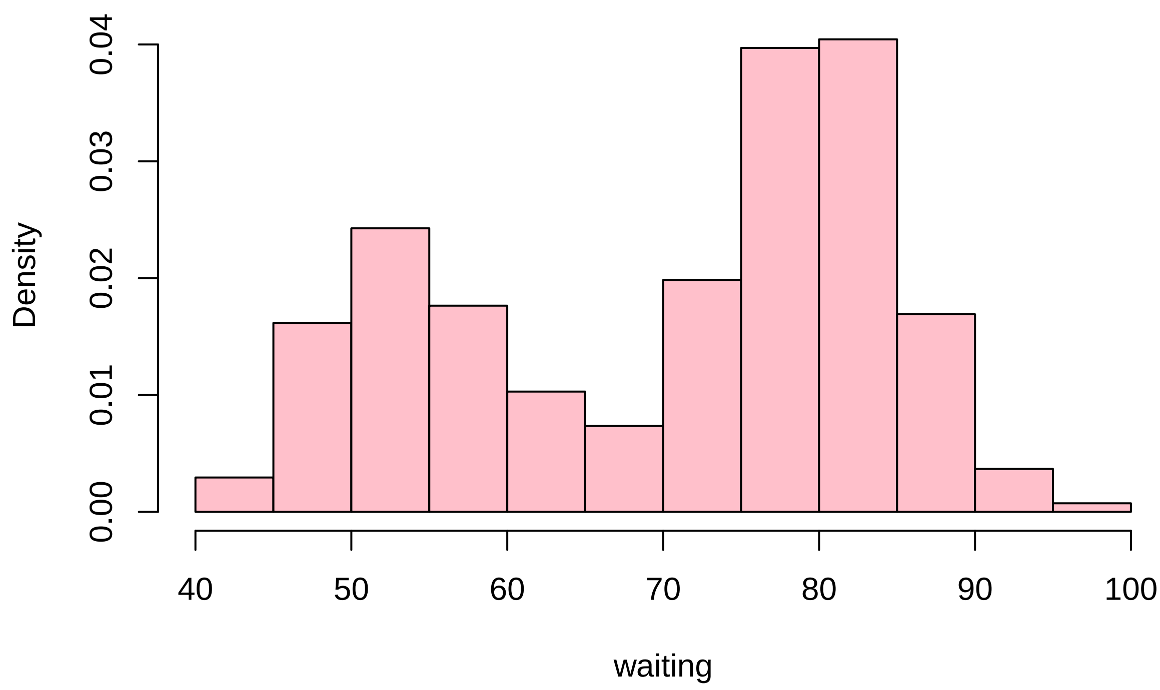
Sayantika Sengupta, Aniket Saha, Anjan

2022-11-15

Problem 3:

```
library(scales)
# library(AICcmodavg)
```

```
attach(faithful)
hist(faithful$waiting,xlab = 'waiting',probability = T,col='pink',main='')
```




```
data_q3 <- faithful
x <- sort(data_q3$waiting)
```

comparing 3 models

```
# model 1
p <- length(x[x<65])/length(x)
as <- mean(x[x<65])
```

```r
ass <- var(x[x<65])
s <- ass/as
a <- as/s
mu <- mean(x[x>=65])
sigma <- sd(x[x>=65])
theta_inital <- c(p, a, s, mu, sigma)
neg_log_likelihood <- function(theta, data){
    n = length(data)

    p = theta[1]
    a = theta[2]
    s = theta[3]
    mu = theta[4]
    sigma = theta[5]

    l = 0
    for (i in 1:n) {
        l = l + log(p*dgamma(data[i], shape = a, scale = s) + (1-p)*dnorm(data[i], mean = mu, sd = sigm
    }
    return(-l)
}
fit = optim(theta_inital,
      neg_log_likelihood,
      data = x,
      control = list(maxit = 1500),
      lower = c(0, 0, 0, -Inf, 0),
      upper = c(1, Inf, Inf, Inf, Inf),
      method="L-BFGS-B")
theta_1 = fit$par
theta_1
```

```
## [1]   0.3574036 101.4861312   0.5371140  80.0176038   5.9487453
```

```r
p = theta_1[1]
a = theta_1[2]
s = theta_1[3]
mu = theta_1[4]
sigma = theta_1[5]
model_1 = p*dgamma(x, shape = a, scale = s) + (1-p)*dnorm(x, mean = mu, sd = sigma)
aic_1 <- 2*5 + neg_log_likelihood(theta_1, x)
```

```r
# model 2
p <- length(x[x<65])/length(x)
as_1 <- mean(x[x<65])
ass_1 <- var(x[x<65])
s_1 <- ass_1/as_1
a_1 <- as_1/s_1
as_2 <- mean(x[x>=65])
ass_2 <- var(x[x>=65])
s_2 <- ass_2/as_2
a_2 <- as_2/s_2
theta_inital <- c(p, a_1, s_1, a_2, s_2)
neg_log_likelihood <- function(theta, data){
    n <- length(data)
```

```
    p <- theta[1]
    a_1 <- theta[2]
    s_1 <- theta[3]
    a_2 <- theta[4]
    s_2 <- theta[5]

    l <- 0
    for (i in 1:n) {
        l = l + log(p*dgamma(data[i], shape = a_1, scale = s_1) + (1-p)*dgamma(data[i], shape = a_2, sca
    }
    return(-l)
}
fit = optim(theta_inital,
      neg_log_likelihood,
      data = x,
      control = list(maxit = 1500),
      lower = c(0, 0, 0, 0, 0),
      upper = c(1, Inf, Inf, Inf, Inf),
      method="L-BFGS-B")
theta_2 <- fit$par
theta_2
```

```
## [1]   0.3582592 101.5126436   0.5371146 169.2757878   0.4728250
```

```
p <- theta_2[1]
a_1 <- theta_2[2]
s_1 <- theta_2[3]
a_2 <- theta_2[4]
s_2 <- theta_2[5]
model_2 <- p*dgamma(x, shape = a_1, scale = s_1) + (1-p)*dgamma(x, shape = a_2, scale = s_2)
aic_2 <- 2*5 + neg_log_likelihood(theta_2, x)
```

```
# model 3
p <- length(x[x<65])/length(x)
m_1 <- mean(x[x<65])
v_1 <- var(x[x<65])
sigma2_1 <- log((v_1/m_1^2) + 1)
mu_1 <- log(m_1) - sigma2_1/2
m_2 <- mean(x[x>=65])
v_2 <- var(x[x>=65])
sigma2_2 <- log((v_2/m_2^2) + 1)
mu_2 <- log(m_2) - sigma2_2/2
theta_inital <- c(p, mu_1, sqrt(sigma2_1), mu_2, sqrt(sigma2_2))
neg_log_likelihood <- function(theta, data) {
    n <- length(data)

    p <- theta[1]
    mu_1 <- theta[2]
    sigma_1 <- theta[3]
    mu_2 <- theta[4]
    sigma_2 <- theta[5]

    l <- 0
    for (i in 1:n) {
```

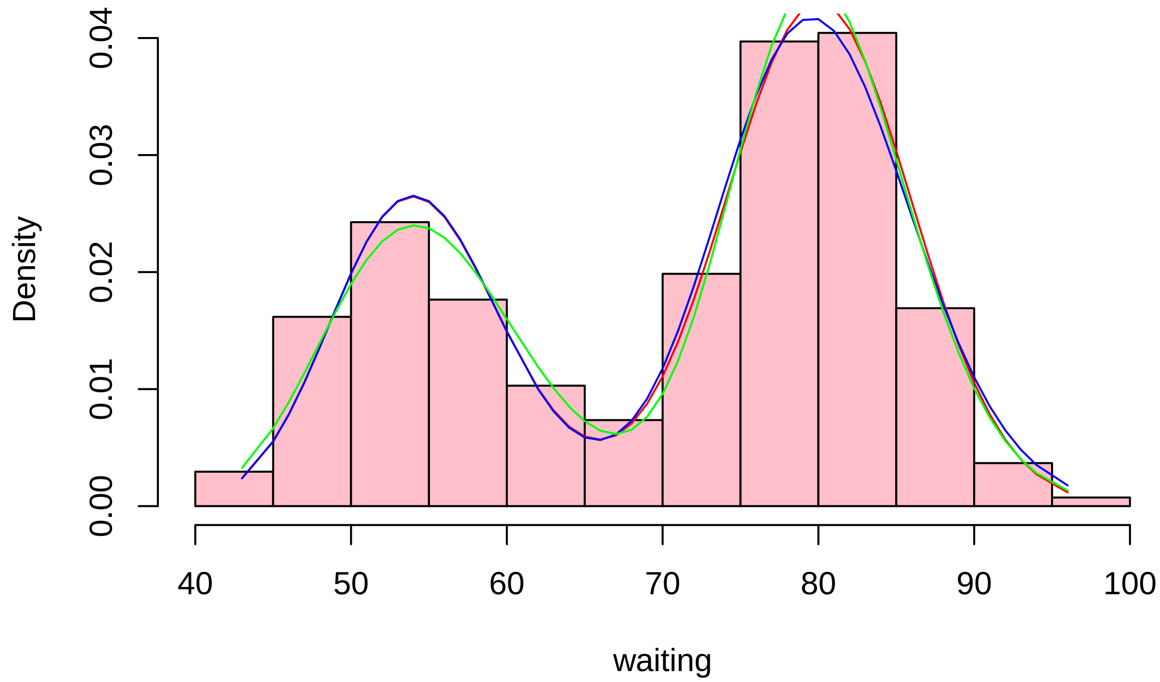
```
        l = l + log(p*dlnorm(data[i], meanlog = mu_1, sdlog = sigma_1) + (1-p)*dlnorm(data[i], meanlog =
    }

    return(-l)
}
fit = optim(theta_inital,
      neg_log_likelihood,
      data = x,
      control = list(maxit = 1500),
      lower = c(0, -Inf, 0, -Inf, 0),
      upper = c(1, Inf, Inf, Inf, Inf),
      method="L-BFGS-B")
theta_3 <- fit$par
theta_3
```

```
## [1] 0.37613816 4.00383608 0.11485512 4.38430182 0.06973823
```

```
p <- theta_3[1]
mu_1 <- theta_3[2]
sigma_1 <- theta_3[3]
mu_2 <- theta_3[4]
sigma_2 <- theta_3[5]
model_3 <- p*dlnorm(x, meanlog = mu_1, sdlog = sigma_1) + (1-p)*dlnorm(x, meanlog = mu_2, sdlog = sigma_
aic_3 <- 2*5 + neg_log_likelihood(theta_3, x)
```

```
hist(x, xlab = 'waiting', probability = T, col='pink', main='')
lines(x, model_1, col = "red")
lines(x, model_2, col = "blue")
lines(x, model_3, col = "green")
```




```r
results <- data.frame(
    models = c("Gamma + Normal", "Gamma + Gamma", "Lognormal + Lognormal"),
    AIC = c(aic_1, aic_2, aic_3)
)
results
```

```
##                  models      AIC
## 1        Gamma + Normal 1043.748
## 2         Gamma + Gamma 1044.362
## 3 Lognormal + Lognormal 1042.710
```

Based on the AIC value of all the models, we choose the third model as it has the lowest AIC value.

The required probability $\mathbb{P}[60 < \texttt{waiting} < 70]$ is:

```r
p <- theta_3[1]
mu_1 <- theta_3[2]
sigma_1 <- theta_3[3]
mu_2 <- theta_3[4]
sigma_2 <- theta_3[5]
reqd_prob <- (p*plnorm(70, meanlog = mu_1, sdlog = sigma_1) + (1-p)*plnorm(70, meanlog = mu_2, sdlog = s
reqd_prob
```

```
## [1] 0.09081323
```

Problem 5:

```
library(quantmod)
getSymbols('TCS.NS')#getting the tcs data
```

```
## [1] "TCS.NS"
```

```
tail(TCS.NS)
```

```
##            TCS.NS.Open TCS.NS.High TCS.NS.Low TCS.NS.Close TCS.NS.Volume
## 2022-11-04      3217.0     3220.05    3166.15      3217.40       1464013
## 2022-11-07      3229.0     3242.80    3195.10      3233.70       1474498
## 2022-11-09      3249.8     3249.80    3201.65      3216.05       1162267
## 2022-11-10      3170.0     3225.00    3170.00      3205.65       1573092
## 2022-11-11      3269.6     3341.60    3255.05      3315.95       3265394
## 2022-11-14      3324.0     3349.00    3309.00      3335.50       1342074
##            TCS.NS.Adjusted
## 2022-11-04         3217.40
## 2022-11-07         3233.70
## 2022-11-09         3216.05
## 2022-11-10         3205.65
## 2022-11-11         3315.95
## 2022-11-14         3335.50
```

```
plot(TCS.NS$TCS.NS.Adjusted)#visualizing the tcs data
```

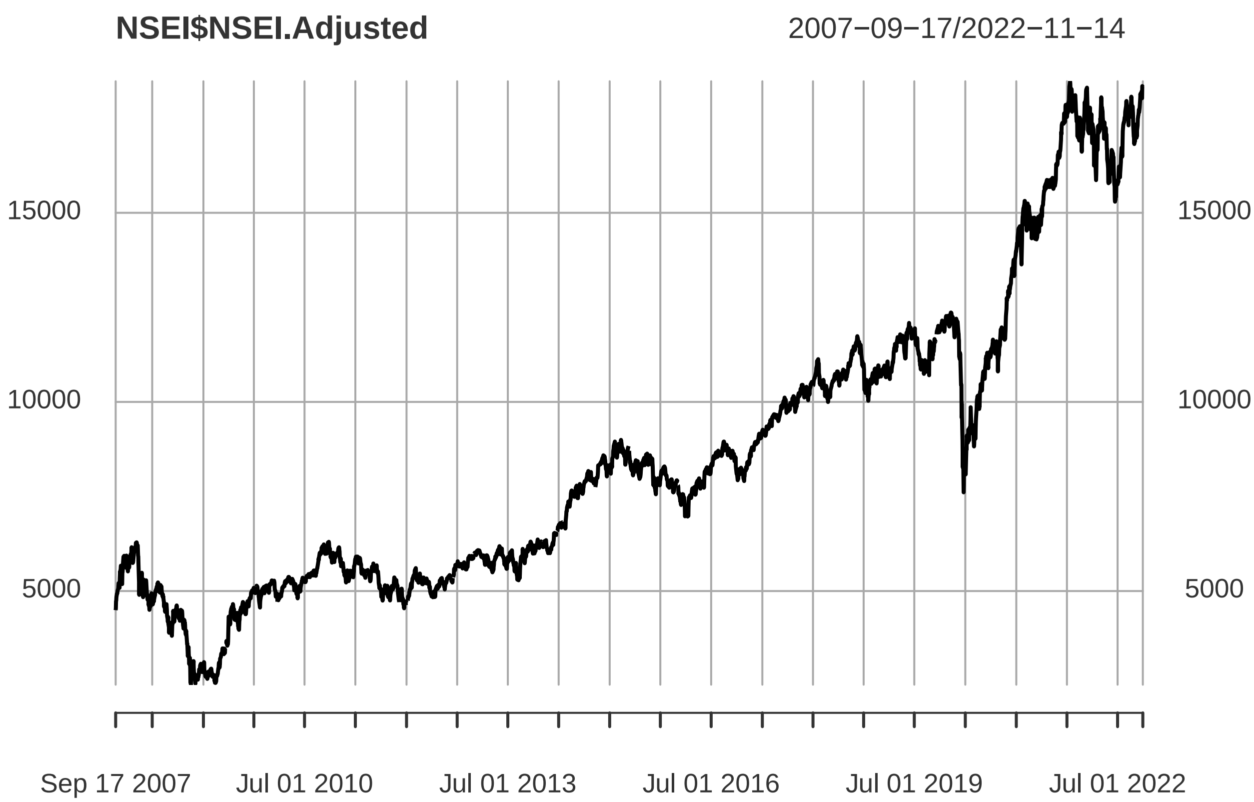

```
getSymbols('^NSEI')#getting nsei dataset
```

```
## [1] "^NSEI"
```

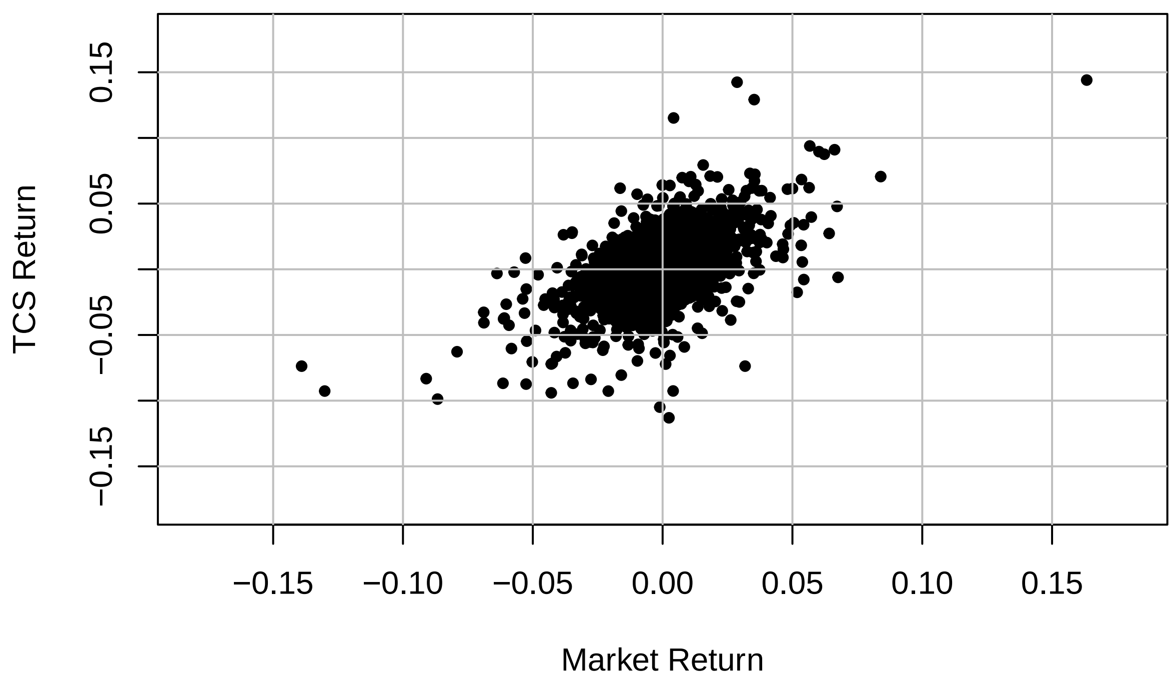
```
tail(NSEI)
```

```
##            NSEI.Open NSEI.High NSEI.Low NSEI.Close NSEI.Volume NSEI.Adjusted
## 2022-11-04  18053.40  18135.10 18017.15   18117.15      267900      18117.15
## 2022-11-07  18211.75  18255.50 18064.75   18202.80      314800      18202.80
## 2022-11-09  18288.25  18296.40 18117.50   18157.00      307200      18157.00
## 2022-11-10  18044.35  18103.10 17969.40   18028.20      256500      18028.20
## 2022-11-11  18272.35  18362.30 18259.35   18349.70      378500      18349.70
## 2022-11-14  18376.40  18399.45 18311.40   18329.15      301400      18329.15
```

```
plot(NSEI$NSEI.Adjusted)#plotting the nsei data
```




```
TCS_rt = diff(log(TCS.NS$TCS.NS.Adjusted))
Nifty_rt = diff(log(NSEI$NSEI.Adjusted))
retrn = cbind.xts(TCS_rt,Nifty_rt)
retrn = na.omit(data.frame(retrn))
plot(retrn$NSEI.Adjusted,retrn$TCS.NS.Adjusted
      ,pch=20
      ,xlab='Market Return'
      ,ylab='TCS Return'
      ,xlim=c(-0.18,0.18)
      ,ylim=c(-0.18,0.18))
grid(col='grey',lty=1)
```



It is given that

$$r_{tcs} = \alpha + \beta r_{ni} + \epsilon$$

MME method:

```
rtcs = mean(retrn$TCS.NS.Adjusted)
rni = mean(retrn$NSEI.Adjusted)
row = cor(retrn$TCS.NS.Adjusted,retrn$NSEI.Adjusted)
sdtcs = sqrt(var(retrn$TCS.NS.Adjusted))
sdnifty = sqrt(var(retrn$NSEI.Adjusted))
n = nrow(retrn)
b1 = row*(sdtcs/sdnifty)
a1 = sdtcs - b1*sdnifty
rtcs_hat = a1 + b1 * retrn$NSEI.Adjusted
error = retrn$TCS.NS.Adjusted - rtcs_hat
sigma1 = sqrt(var(error))
Method_of_Moments <- c(a1,b1,sigma1)
Method_of_Moments
```

```
## [1] 0.008859627 0.743683875 0.016184664
```

Ordinary least square method:

```
lin_mod = summary(lm(TCS.NS.Adjusted~NSEI.Adjusted, data = retrn))
a2 = lin_mod$coefficients [1,1]
b2 = lin_mod$coefficients [2,1]
Method_of_Moments <- c(a1,b1,sigma1)
k = (n-2)/n
```

```
errornew = retrn$TCS.NS.Adjusted - (a2+ b2*retrn$NSEI.Adjusted)
sigma2 = (sqrt(var(errornew)))*k
OLS <- c(a2,b2, sigma2)
```

Representing the estimates as a dataframe:

```
Parameters <- c("alpha", "beta", "sigma")
table = data.frame(Parameters, Method_of_Moments,OLS)
table
```

```
##   Parameters Method_of_Moments          OLS
## 1      alpha       0.008859627 0.0004628228
## 2       beta       0.743683875 0.7436838751
## 3      sigma       0.016184664 0.0161758725
```

Estimated rise in price of tcs

```
est1 = 3200 - ( 200*b1)
est2 = 3200 - ( 200*b2)
est = c(est1,est2)
est
```

```
## [1] 3051.263 3051.263
```