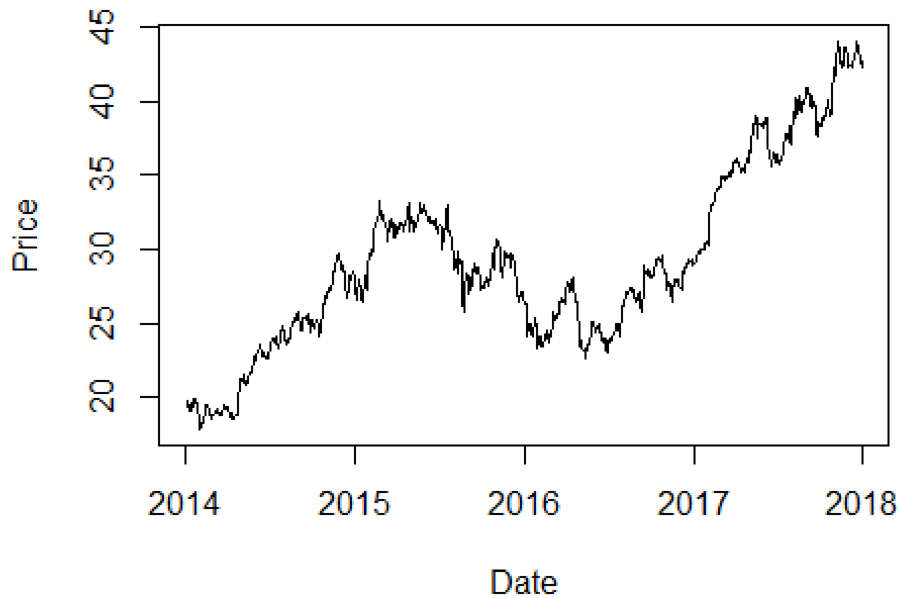
metropolis_approachv4.R

cmlem

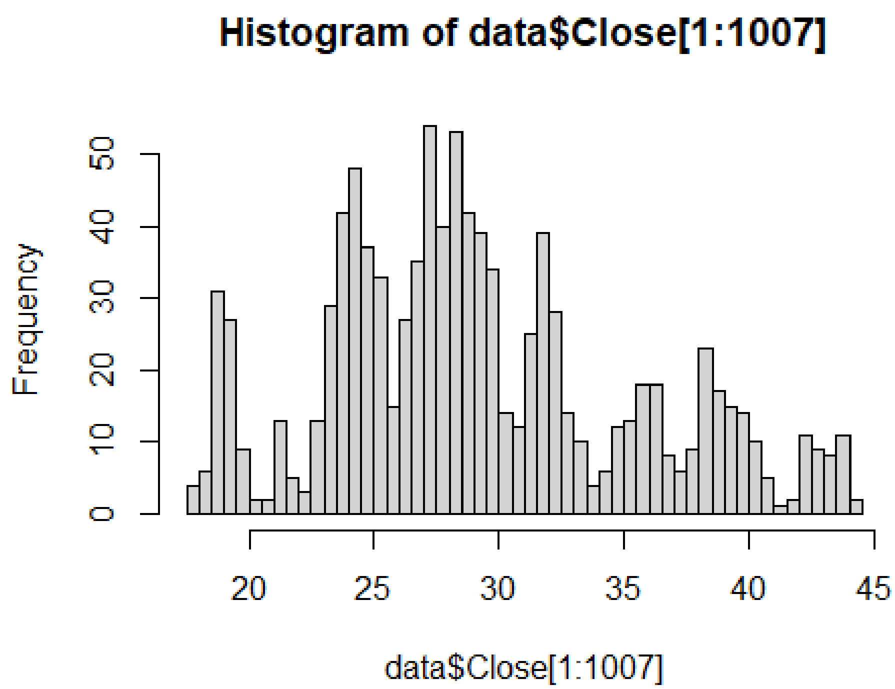
2024-04-30

```
# helper functions
Normal_Kernel_Estimate <- function(x = -10000:10000, xi = data, h, n =
length(data)) {

  f_hat <- c() # Set empty vector to store values for f_hat
  xvalues <- c() # Set empty vector for graph
  for (i in x/100) {
    distance <- (i - xi) / h
    f_hat_value <- 1/(length(distance)*h) * sum(dnorm(distance)) # Use normal
density as kernel function

    f_hat <- append(f_hat, f_hat_value)
    xvalues <- append(xvalues, i)
  }
  return(list(xvalues=xvalues, f_hat=f_hat))
}


target_distribution <- function(x, target_dist_x, target_dist_p){
  round_x = round(100*x)/100
  x_index = which(target_dist_x==round_x)
  p = target_dist_p[x_index]
  return(p)
}

#Main function
Metropolis_Hastings <- function(xt, n, data, kernel_bandwidth, proposal_sd){
  # generates a Markov Chain of length n, starting at value xt.
  # The target distribution is generated from a kernel density estimate using
data
  xt_vector <- c() # Set a vector to track all values of xt:

  KDE = Normal_Kernel_Estimate(xi = data, h=kernel_bandwidth)

  for (i in 1:n){
    xt_vector <- append(xt_vector, xt) # Update vector
    x_new <- rnorm(1, mean=xt, sd=proposal_sd) # Set x* from the proposal
distribution

    # Calculate the inputs to the Metropolis-Hastings Ratio:
    f_xt <- target_distribution(xt, KDE$xvalues, KDE$f_hat)
```

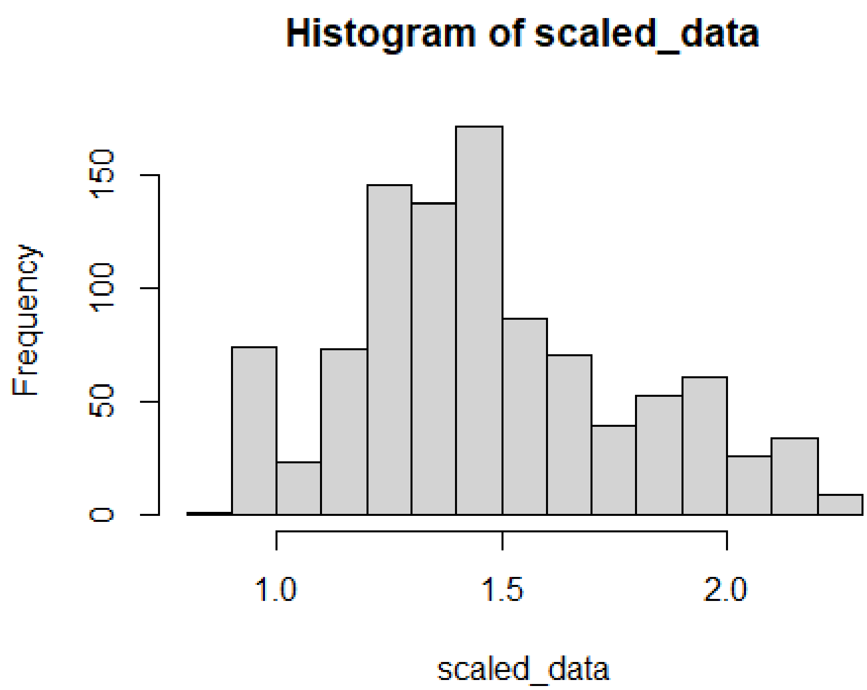
```r
    f_xnew <- target_distribution(x_new, KDE$xvalues, KDE$f_hat)
    g_xt <- dnorm(xt, x_new, proposal_sd)
    g_xnew <- dnorm(x_new, xt, proposal_sd)

    MH_Ratio <- (f_xnew * g_xt) / (f_xt * g_xnew)
    # Sample from Uniform(0,1) to determine if x* is accepted
    u <- runif(1)
    if(MH_Ratio > u){
      xt <- x_new
    }
  }
  return(xt_vector)
}

validate = function(data, kernel_bandwidth, proposal_sd){
  log_scaled_data = log(data$Close[1:1321]/data$Open[1])
  test_data = log(data$Close[2330:length(data$Close)]/data$Open[2330])

  # Build MC based on historic data
  MC = Metropolis_Hastings(0, 6000, log_scaled_data, kernel_bandwidth,
proposal_sd)
  MC_stationary = MC[5000+(1:length(test_data))]

  # Compare histogram of Markov Chain to data: Compute Delta_MC
  test_hist = hist(test_data, breaks = (-100:100)/10)
  MC_hist = hist(MC_stationary, breaks = (-100:100)/10)
  delta = sum(abs(test_hist$density - MC_hist$density))
  return(delta)
}

# extract and transform data
data<-read.csv(file='G:/My Drive/JHU/EN625664ComputationalStats/Final
Project/data/AAPL.csv',sep=",",header=TRUE)

plot(as.Date(data$Date[1:1007]), data$Close[1:1007], type='l', xlab = 'Date',
ylab = 'Price')
```



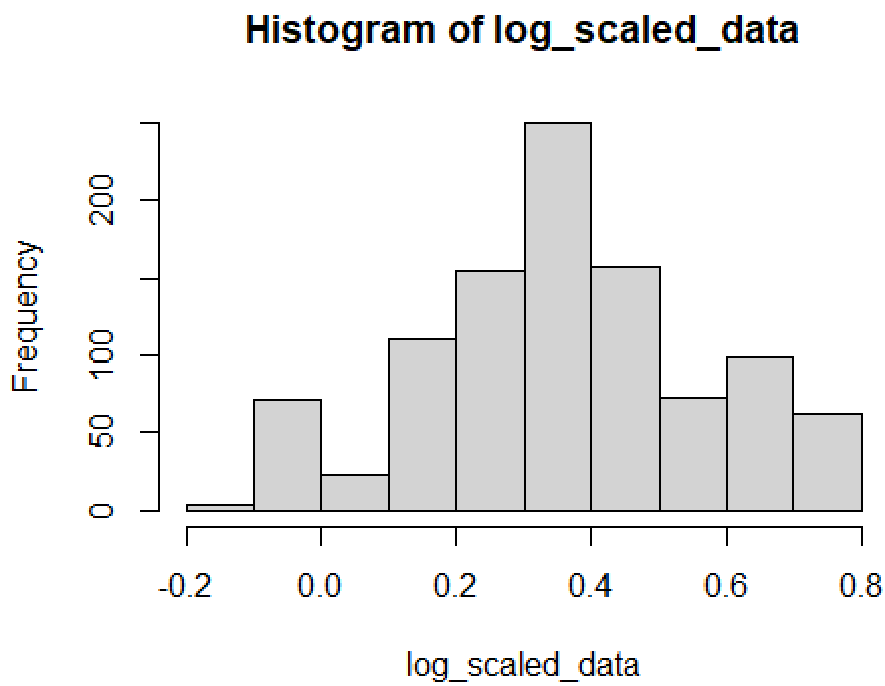

```r
hist(data$Close[1:1007], 40)
```



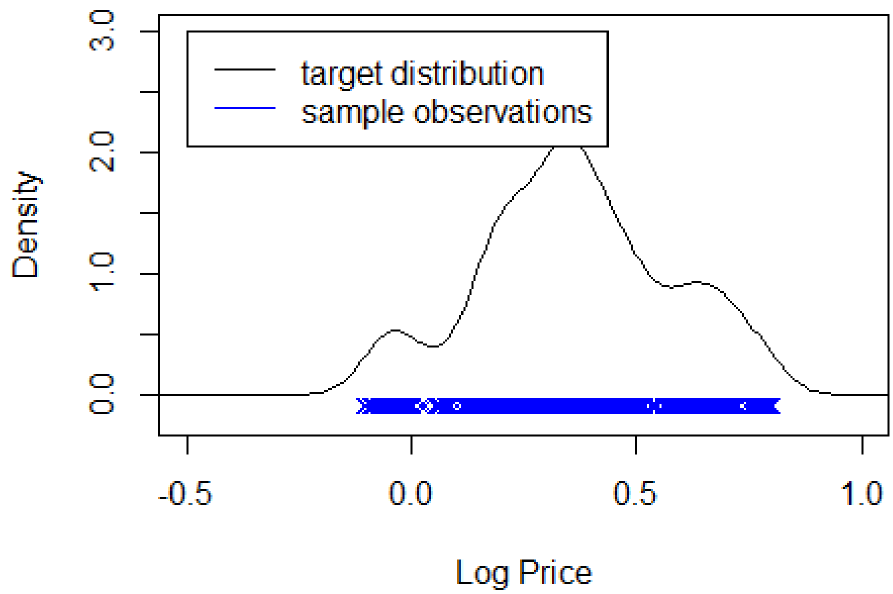
```r
scaled_data = data$Close[1:1007]/data$Open[1]
hist(scaled_data)
```



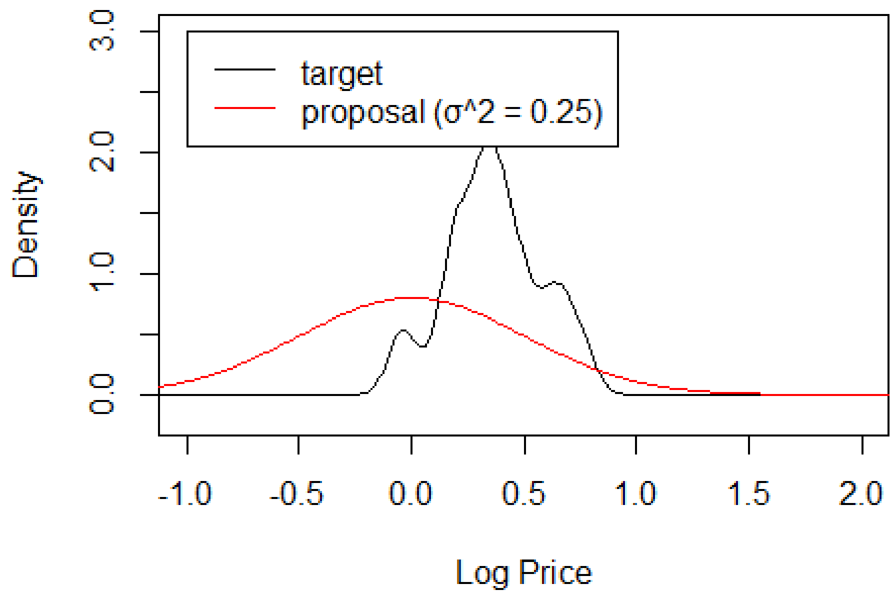

```r
log_scaled_data = log(scaled_data)
hist(log_scaled_data)
```



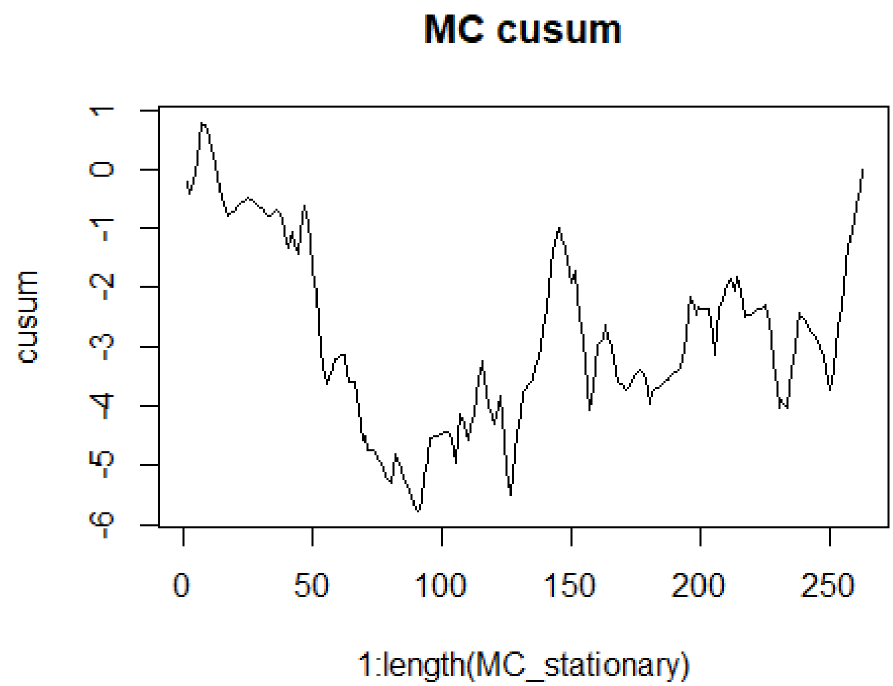

```r
h_Silverman <- (4 / (3*length(log_scaled_data))) ^ (1/5) *
sd(log_scaled_data) # Silverman's Rule of Thumb
h_Terrell <- 3 * ((1/(2*sqrt(pi))) / (35 * length(log_scaled_data))) ^ (1/5)
* sd(log_scaled_data) # Terrell's Maximal Smoothing Principle
h_SJ <- bw.SJ(log_scaled_data) # Sheather-Jones Approach

test_data = data$Close[2330:length(data$Close)]/data$Open[2330]

# Build MC based on historic data
MC = Metropolis_Hastings(0, 20000, log_scaled_data, h_Silverman, .5)

# Show that proposal distribution more diffuse than target distribution
KDE = Normal_Kernel_Estimate(xi = log_scaled_data, h=h_Silverman) # target
proposal <- dnorm(KDE$xvalues, 0, .5) # proposal
plot(log_scaled_data, rep(-0.1, length(log_scaled_data)), xlim=c(-.5,1),
ylim=c(-.2, 3), pch=4, xlab = 'Log Price', ylab = 'Density', col='blue')
lines(KDE$xvalues, KDE$f_hat)
legend(-.5,3,legend=c('target distribution', 'sample observations'),
col=c('black', 'blue'), lty=1)
```



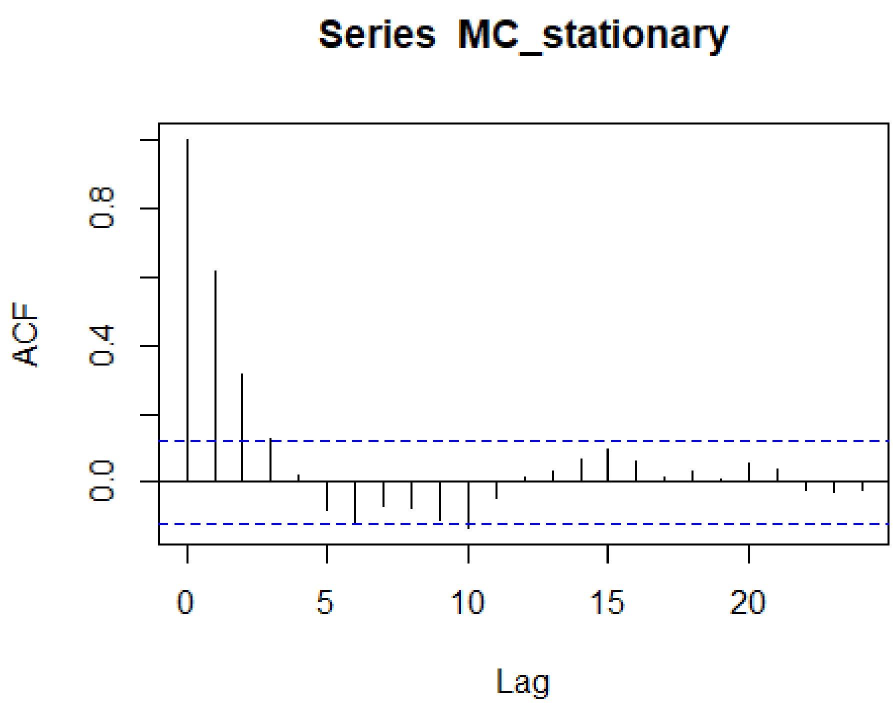

```
plot(KDE$xvalues, KDE$f_hat, type='l', xlim=c(-1,2), ylim=c(-.2, 3), xlab =
'Log Price', ylab = 'Density')
lines(KDE$xvalues, proposal, col='red')
legend(-1,3,legend=c('target', 'proposal (σ^2 = 0.25)'), col=c('black',
'red'), lty=1)
```



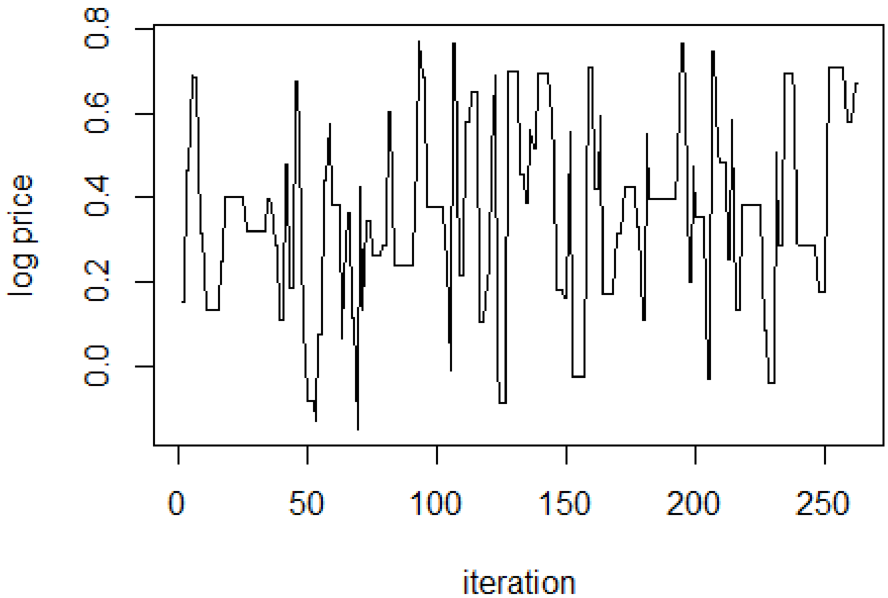

```
KDE$f_hat[0:5]

## [1] 0 0 0 0 0

proposal[0:5]

## [1] 0 0 0 0 0

#   Show the MC is in the stationary distribution
MC_stationary = MC[10000+(1:length(test_data))]
#MC_stationary = MC[0+(1:20000)]
theta_hat = mean(MC_stationary)
cusum = cumsum(MC_stationary - theta_hat)
plot(1:length(MC_stationary), cusum, type='l', main='MC cusum')
```



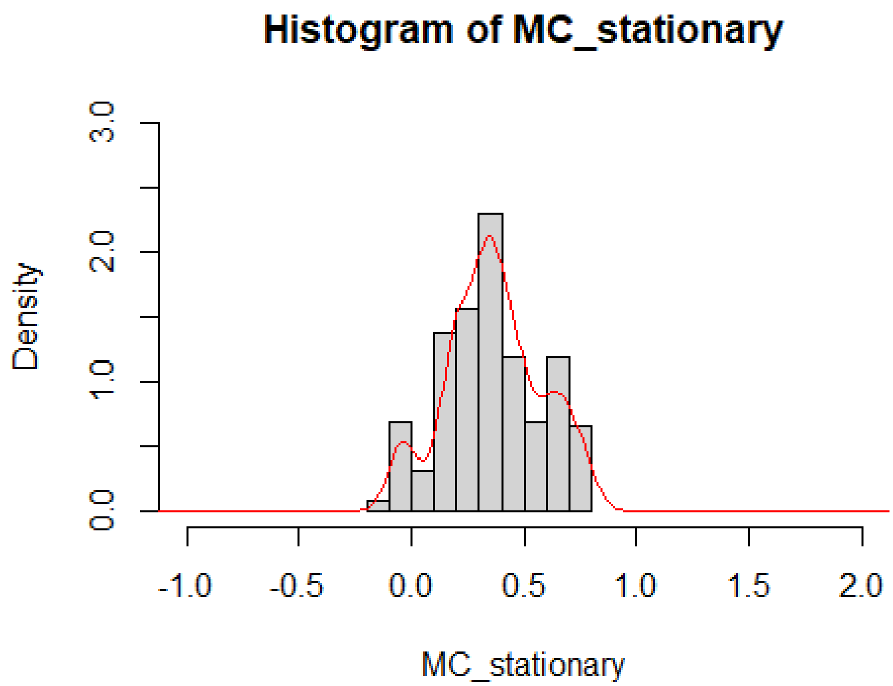

```r
acf(MC_stationary)
```



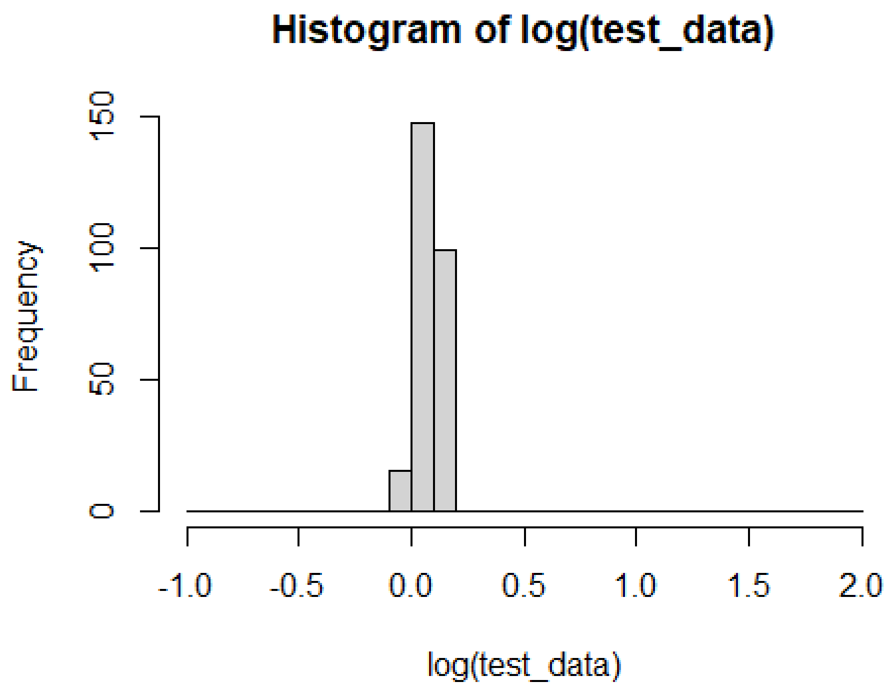

```r
plot(MC_stationary, type='l', xlab = 'iteration', ylab = 'log price')
```



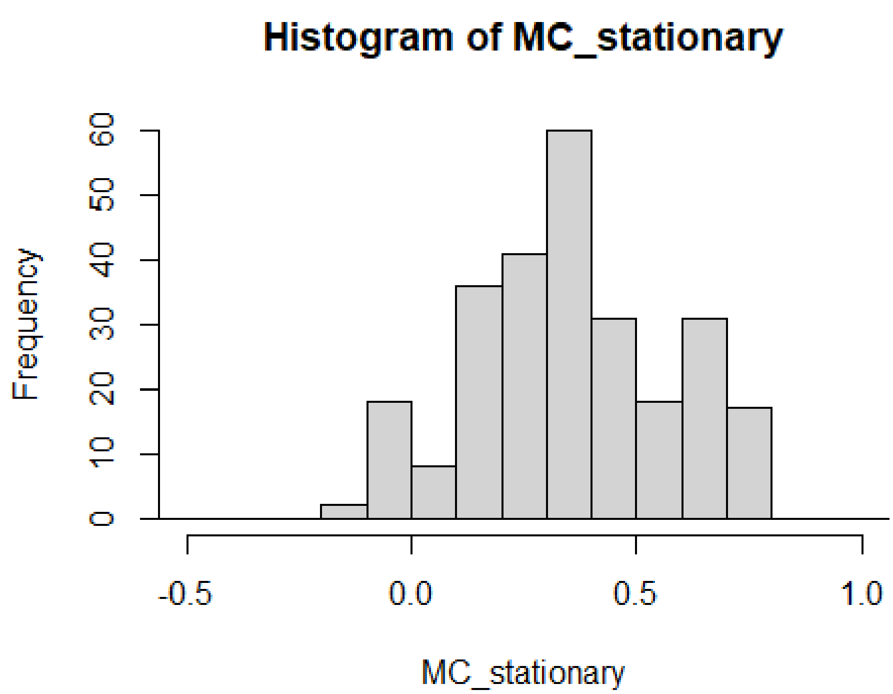

```
# compare histogram of Markov chain resembles target distribution
nbins = round(1 + log2(length(MC_stationary)))
hist(MC_stationary, breaks=nbins, xlim=c(-1,2), ylim=c(0,3), freq=FALSE)
lines(KDE$xvalues, KDE$f_hat, col='red')
```



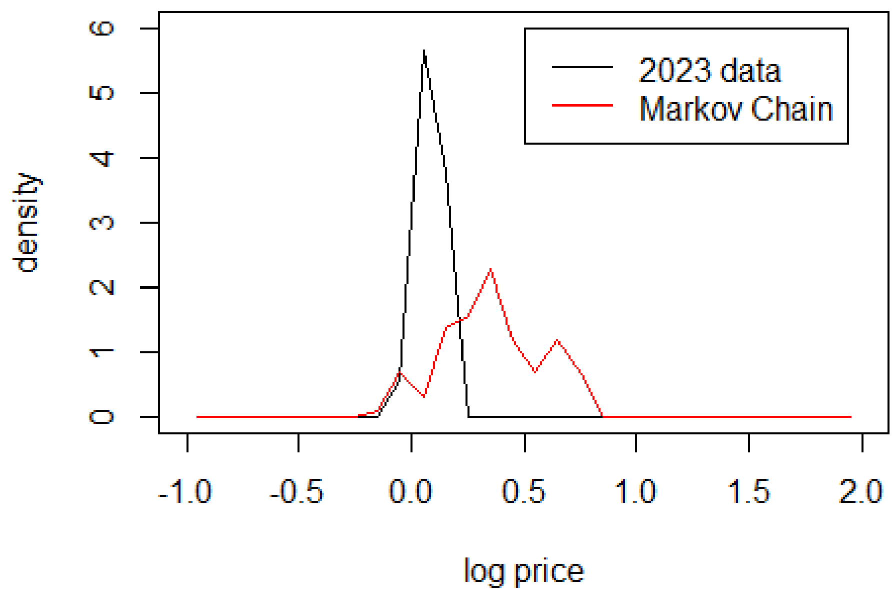

```r
# Compare histogram of Markov Chain to data: Compute Delta_MC


nbins = round(1 + log2(length(test_data)))
test_hist = hist(log(test_data), breaks = (-10:20)/10)
```



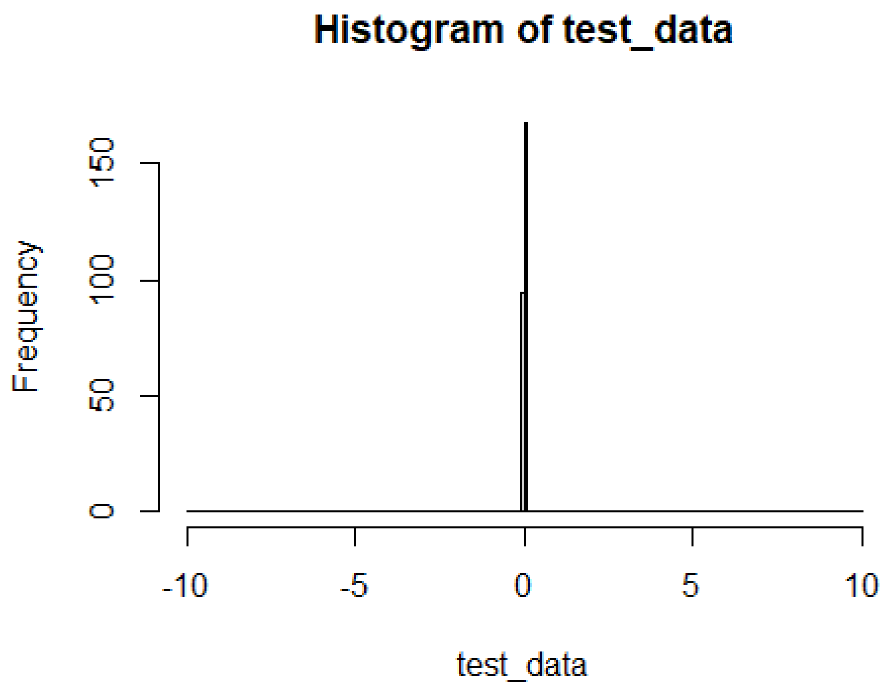

```
MC_hist = hist(MC_stationary, breaks = (-10:20)/10, xlim=c(-.5, 1))
```

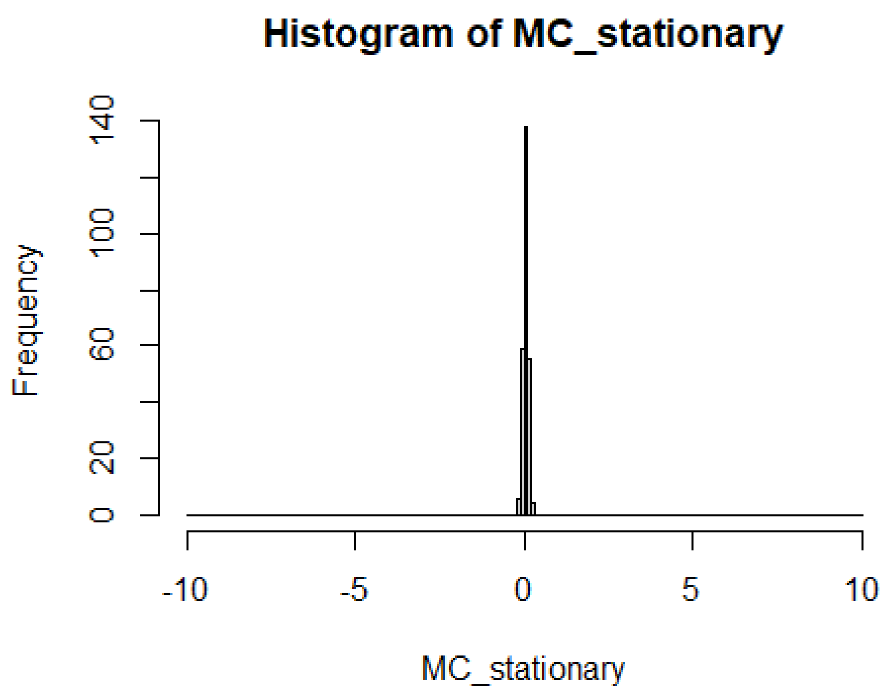


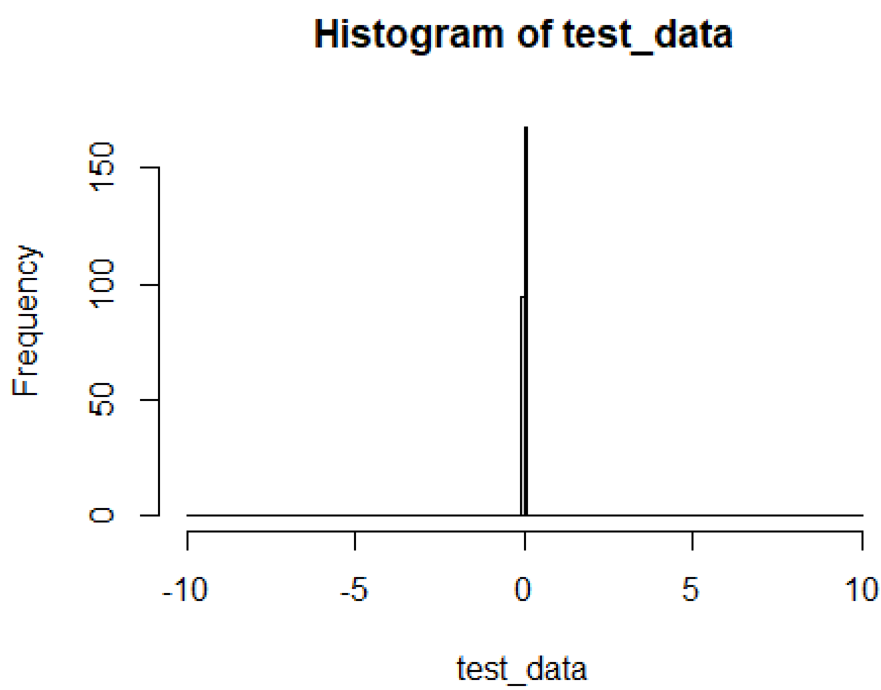
```r
plot(test_hist$mids, test_hist$density, type='l', col='black', xlim=c(-1,2),
ylim=c(0,6), ylab = 'density', xlab = 'log price')
lines(MC_hist$mids, MC_hist$density, col='red')
legend(.5,6,legend=c('2023 data', 'Markov Chain'), col=c('black', 'red'),
lty=1)
```

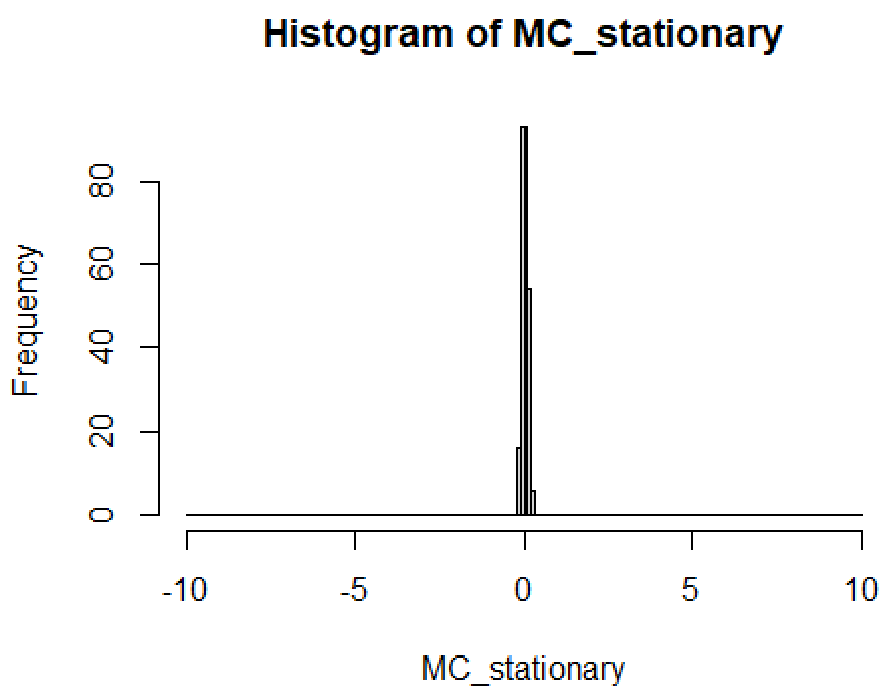


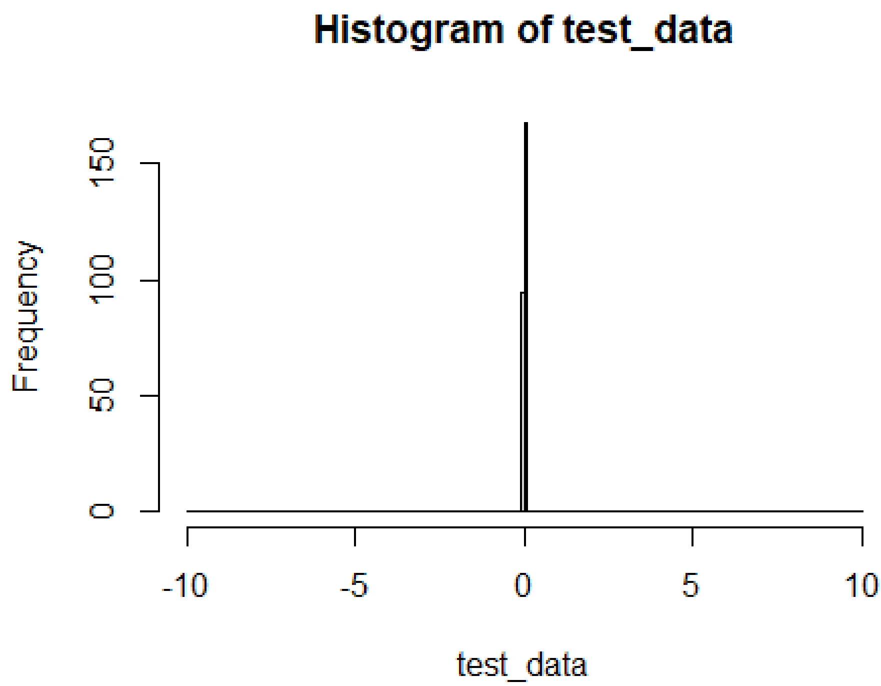

```r
delta = sum(abs(test_hist$density - MC_hist$density))

data<-read.csv(file='G:/My Drive/JHU/EN625664ComputationalStats/Final
Project/data/PG.csv',sep=",",header=TRUE)
# effect of kernel bandwidth on delta
bandwidths = c(0.001, 0.002, 0.005, 0.01, .02, .05, 0.1, 0.2, 0.5, 1, 2)
deltas = c()
for (bandwidth in bandwidths) {
  deltas = c(deltas, validate(data, bandwidth, .5))
}
```

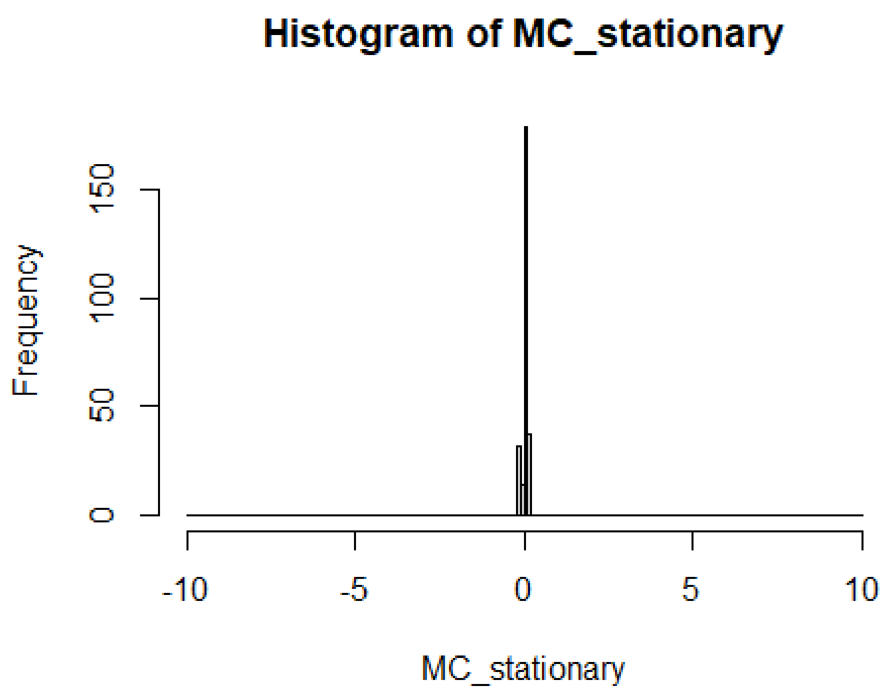

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

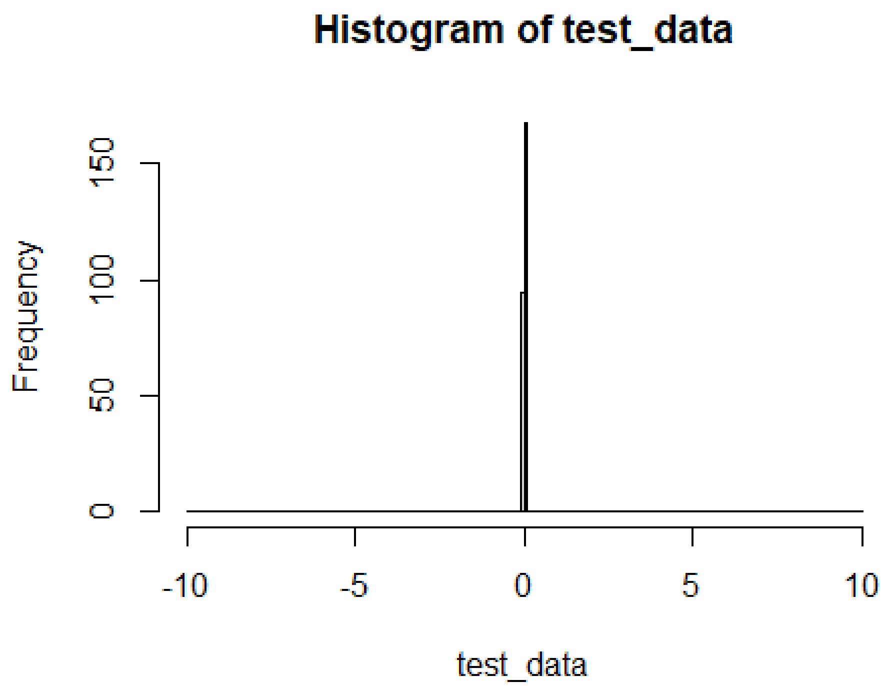

Histogram of MC_stationary
Frequency
0
20
60
100
140
-10
-5
0
5
10
MC_stationary

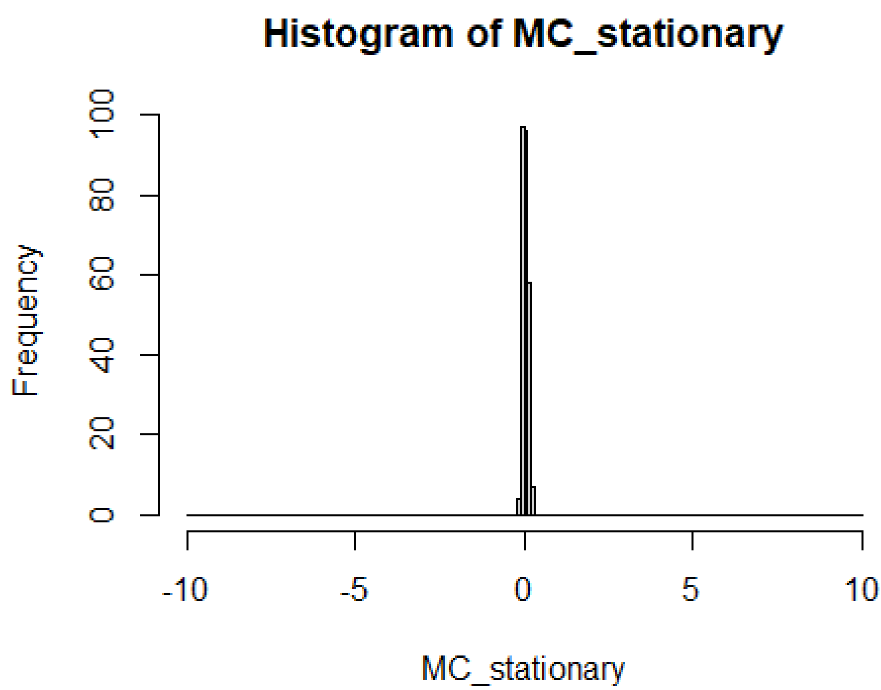

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

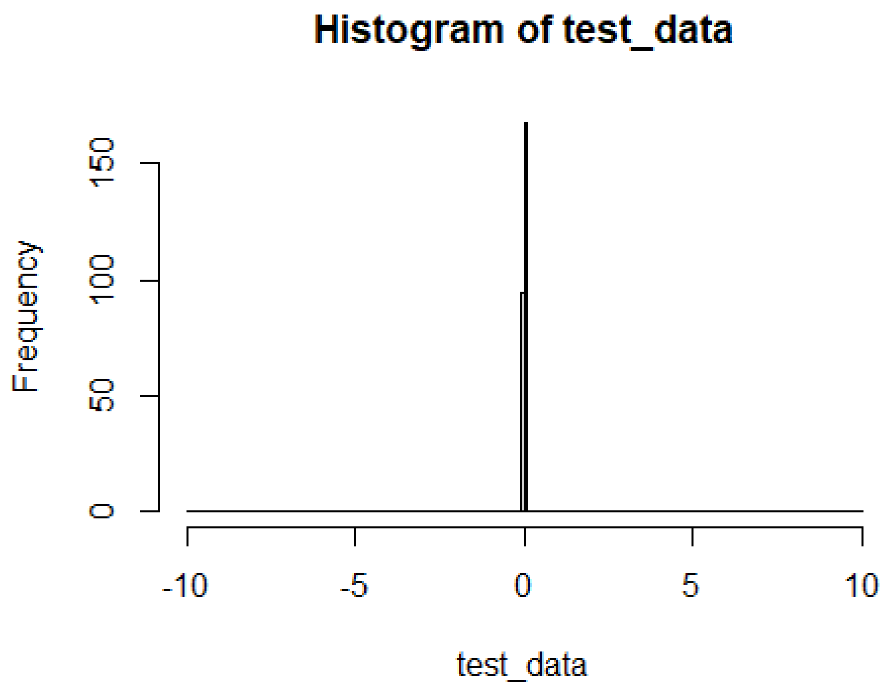

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

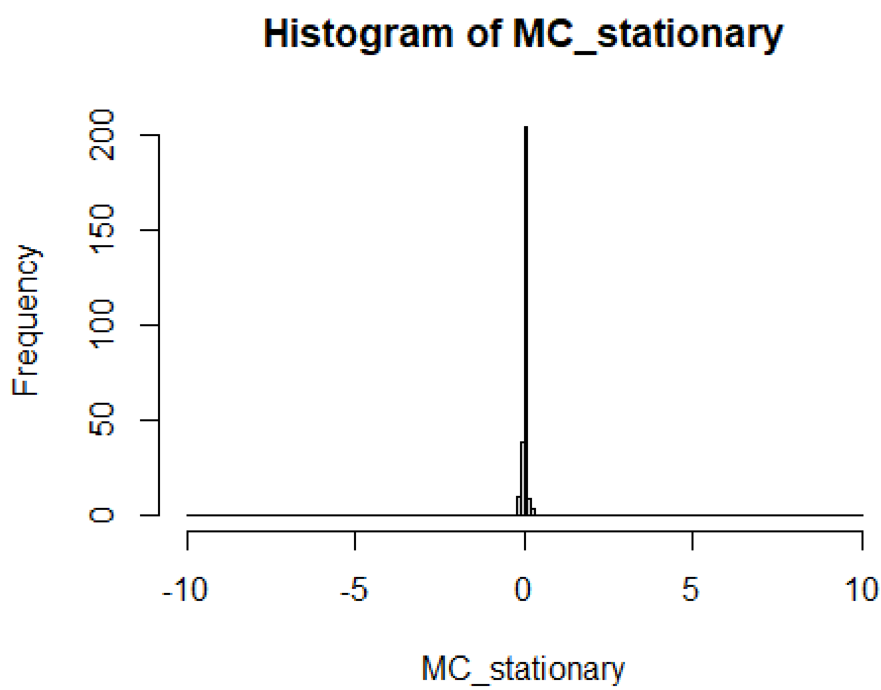

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

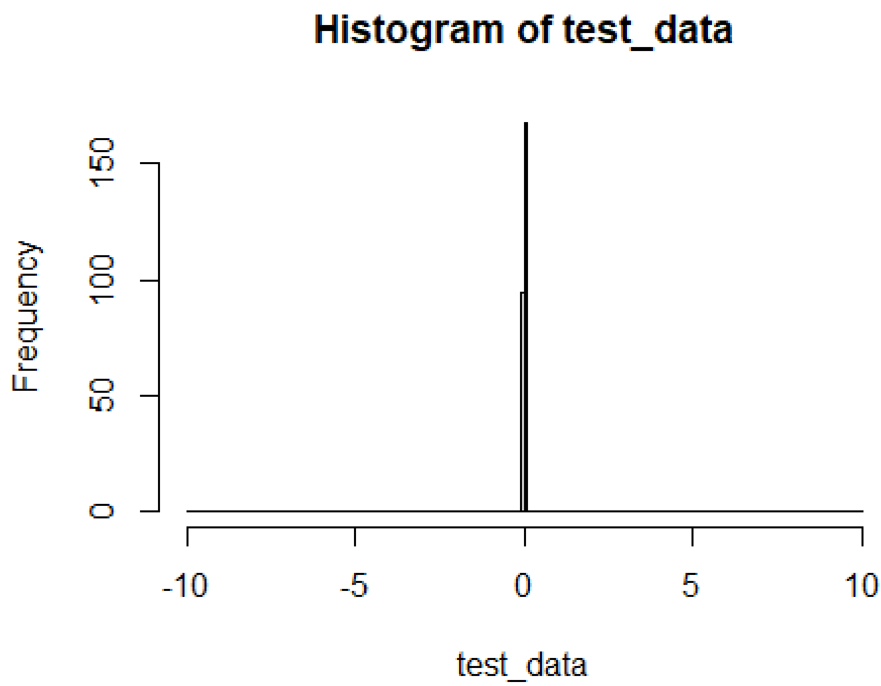

Histogram of MC_stationary
Frequency
0
50
100
150
-10
-5
0
5
10
MC_stationary

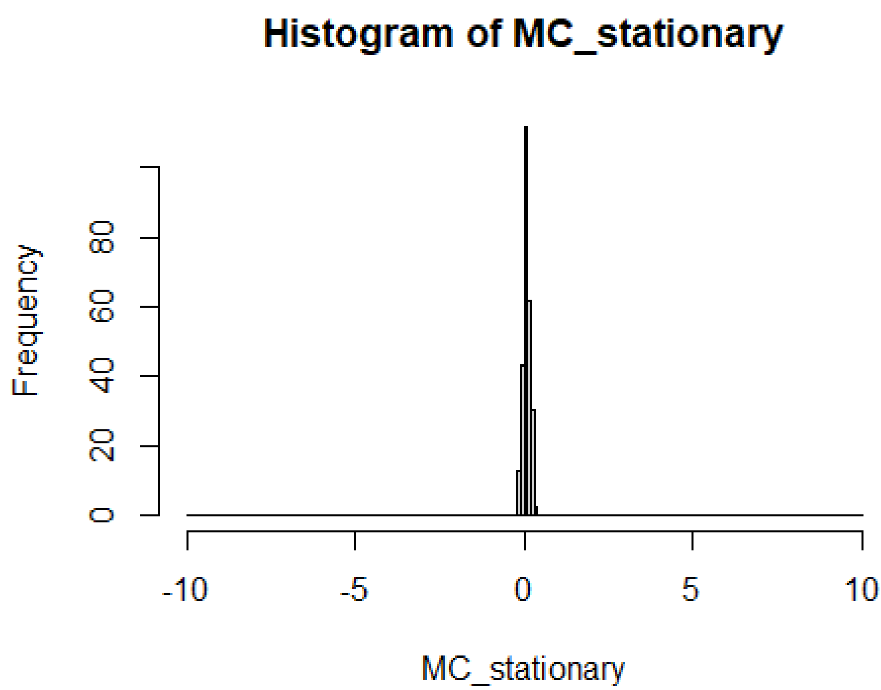

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

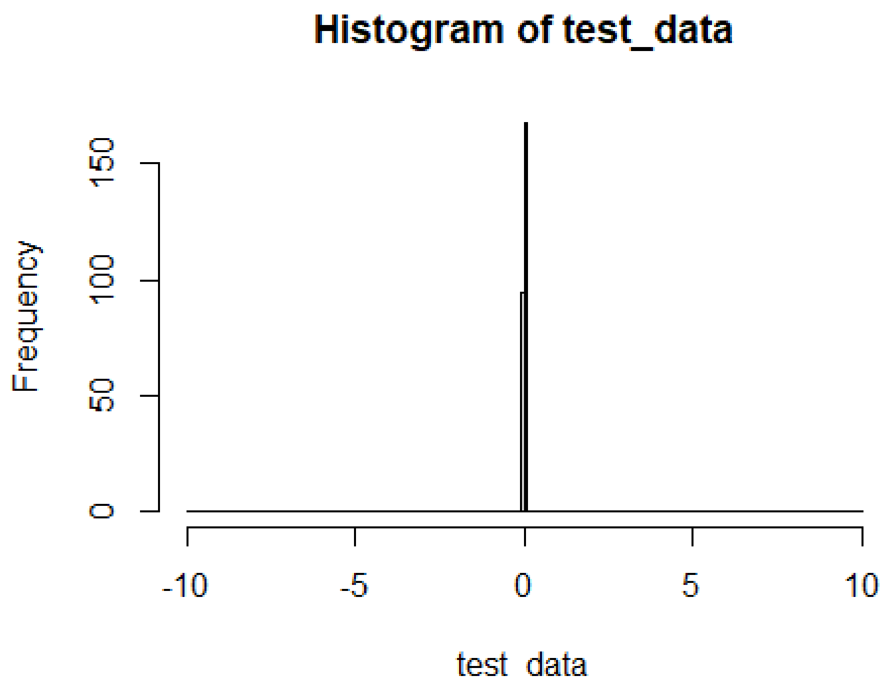

Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary

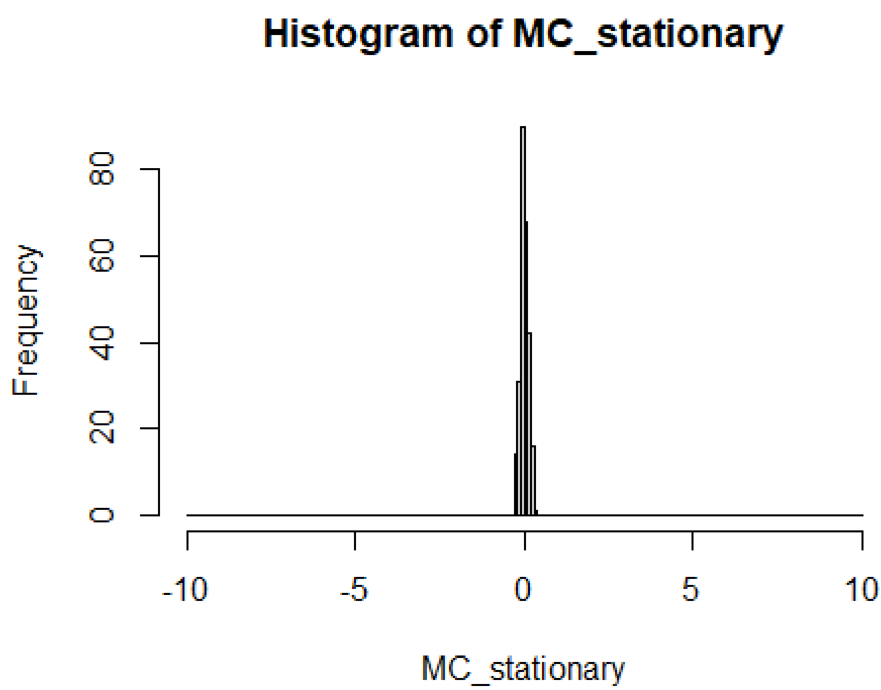

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

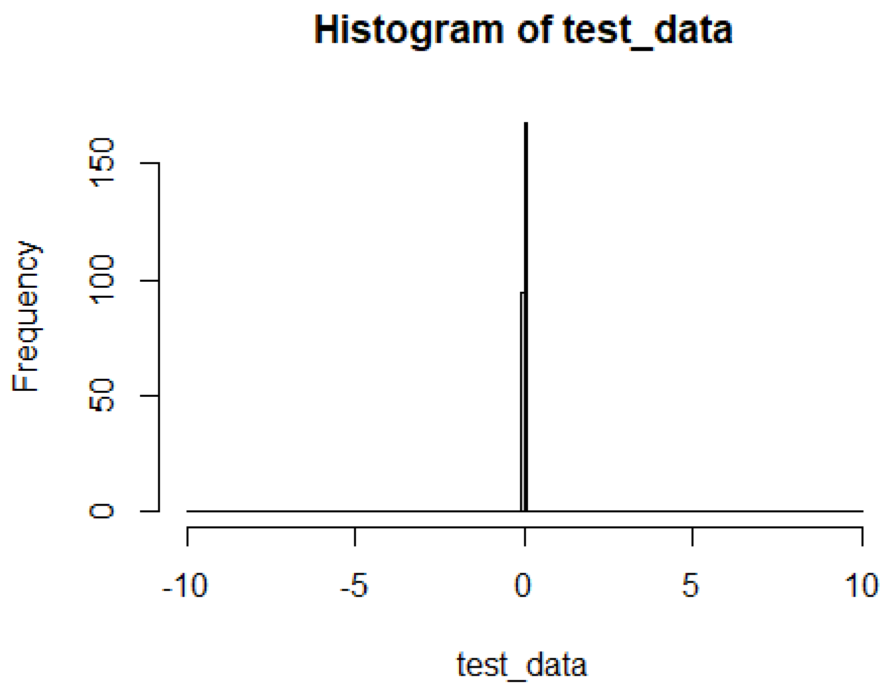

Histogram of MC_stationary
Frequency
0
50
100
150
200
-10
-5
0
5
10
MC_stationary

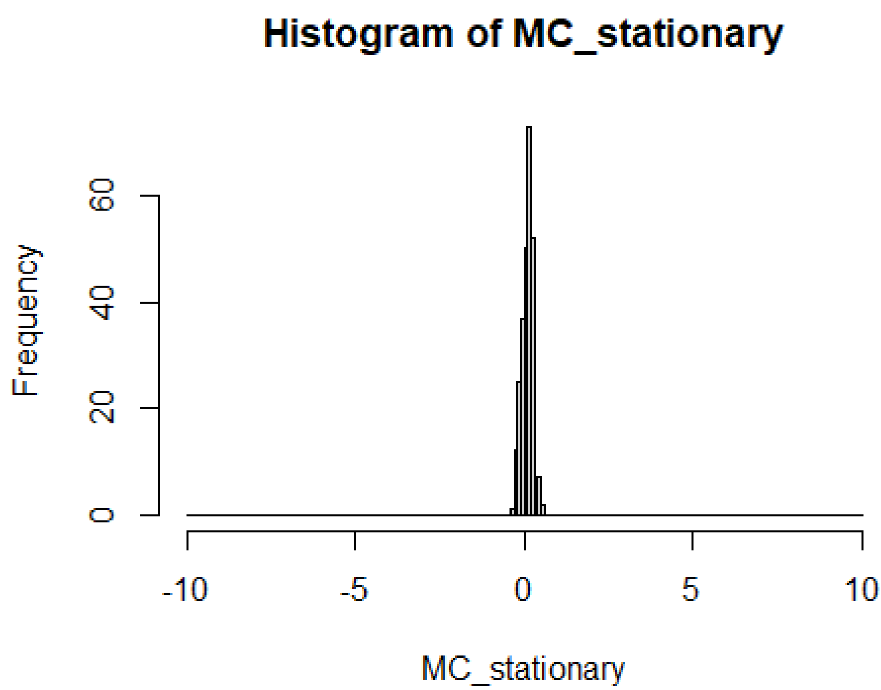

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

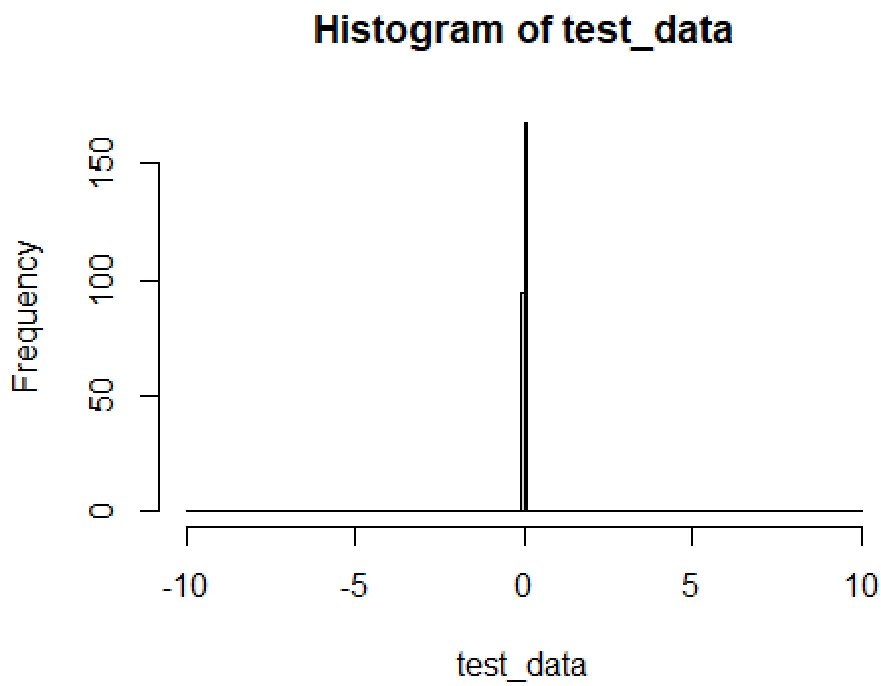

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

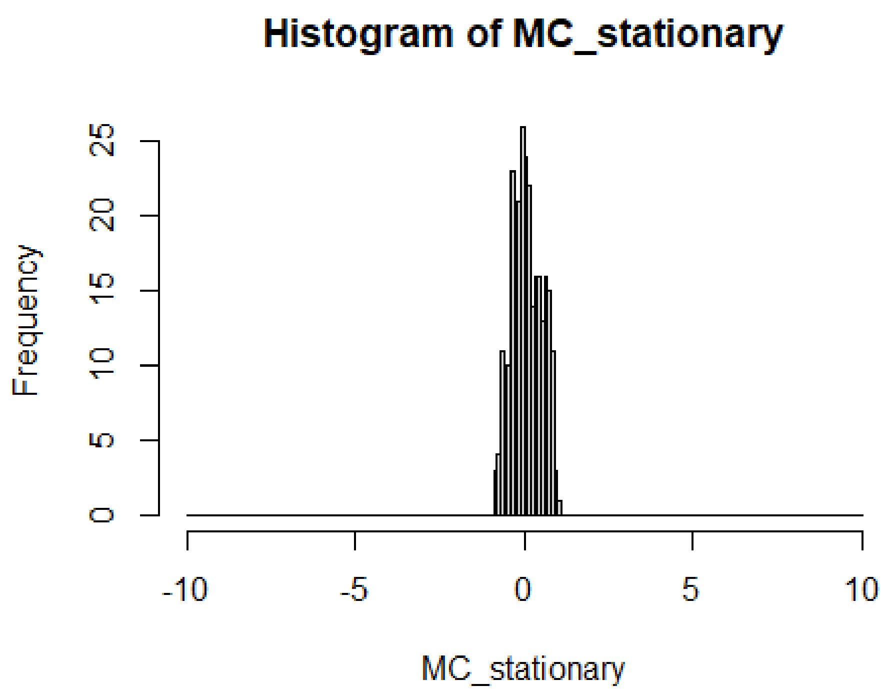

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

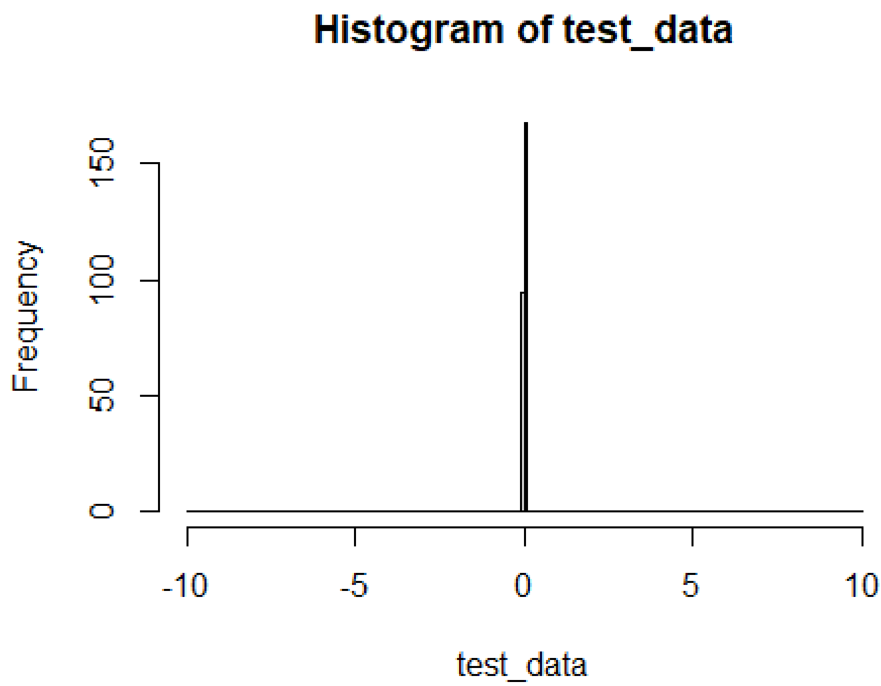

Histogram of MC_stationary
Frequency
0
20
40
60
-10
-5
0
5
10
MC_stationary

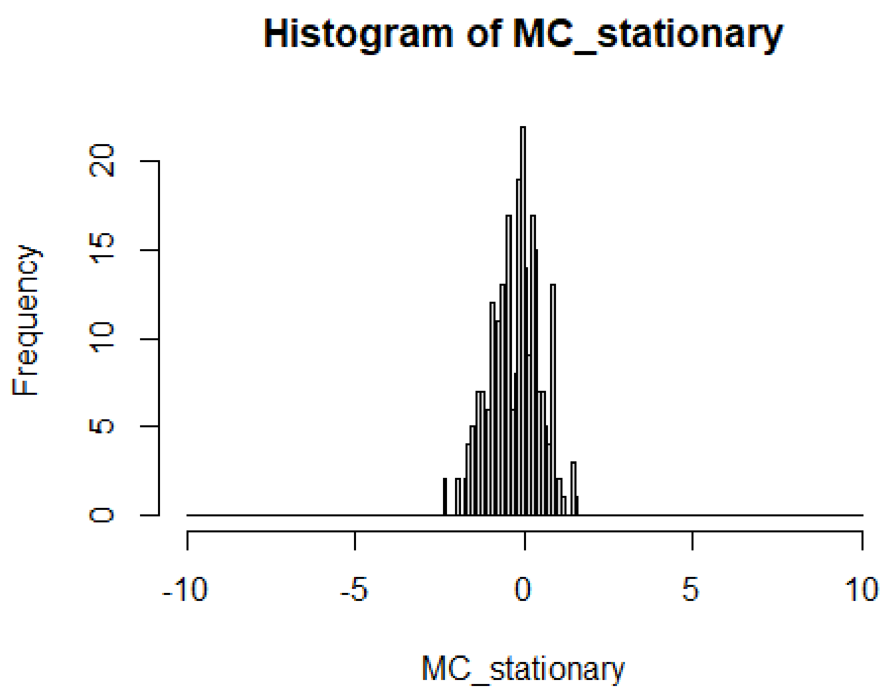


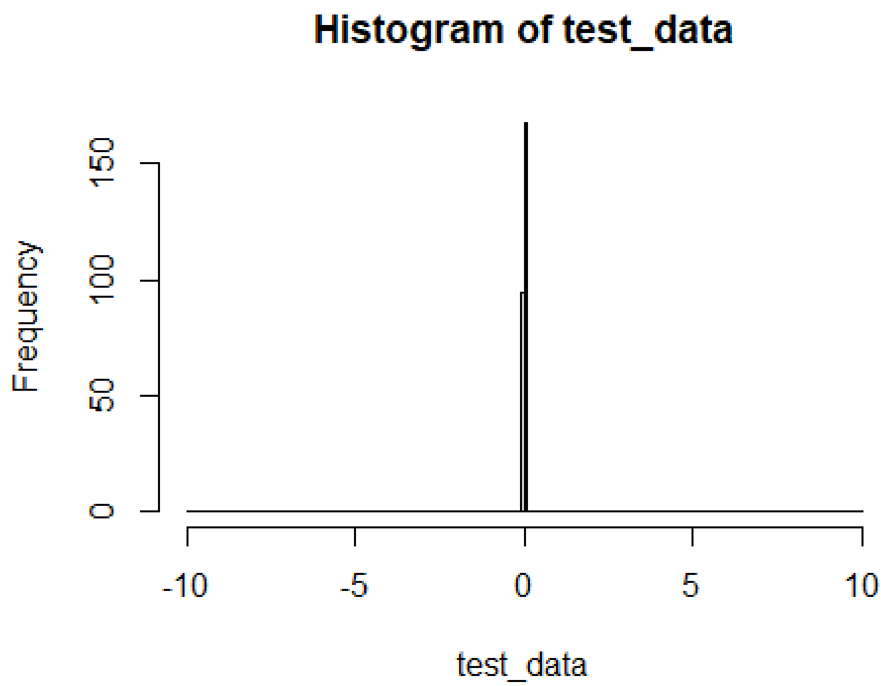
Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

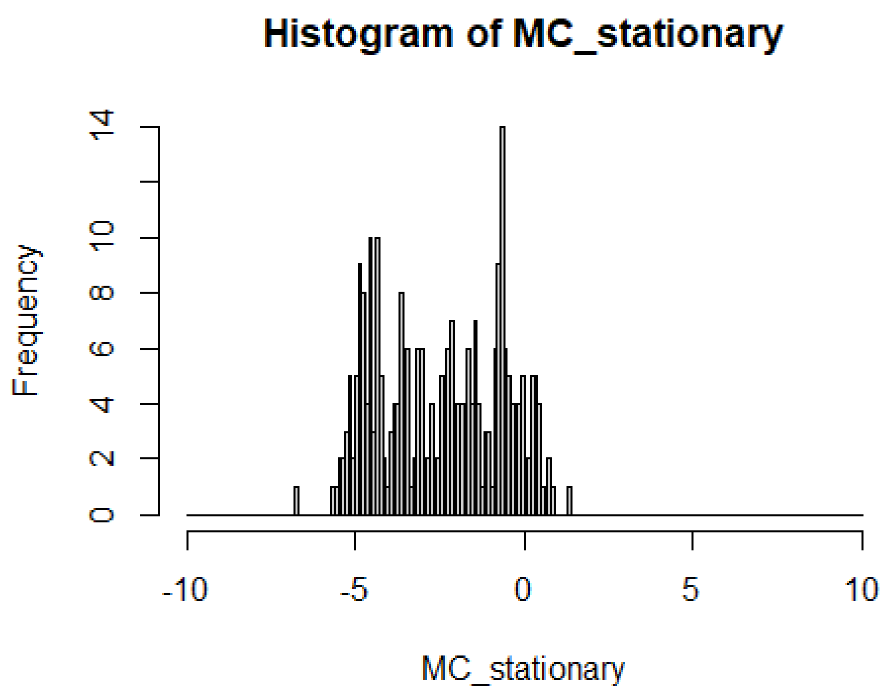


Histogram of MC_stationary
Frequency
0
5
10
15
20
25
-10
-5
0
5
10
MC_stationary


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

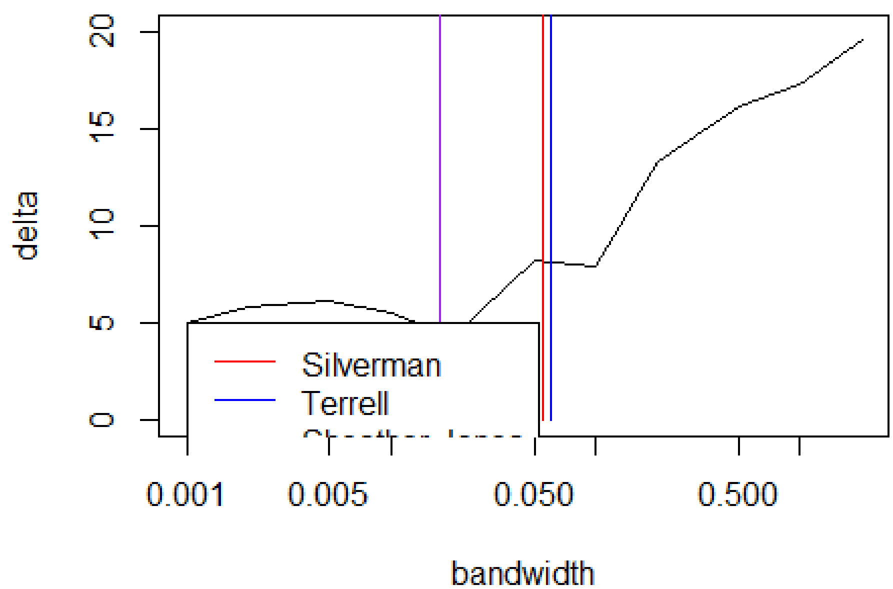

Histogram of MC_stationary
Frequency
0
5
10
15
20
-10
-5
0
5
10
MC_stationary



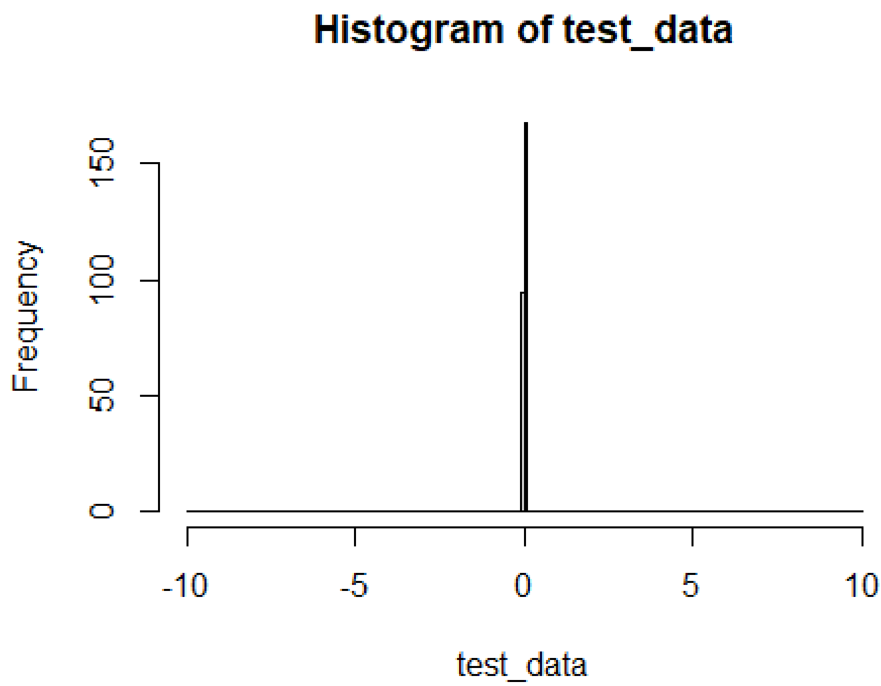


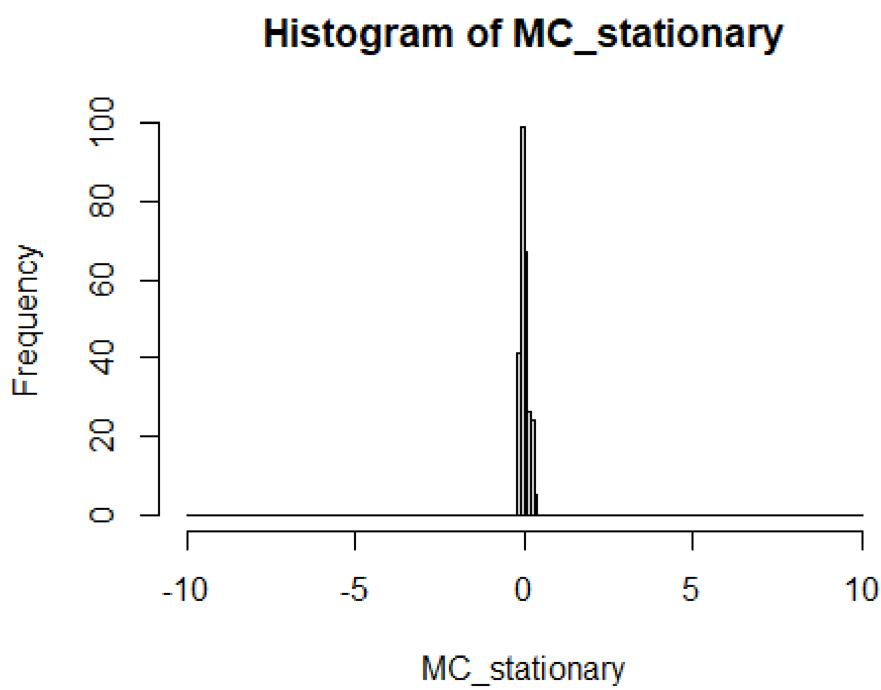

```r
plot(bandwidths, deltas, type='l', xlab = 'bandwidth', ylab = 'delta', log =
'x', ylim=c(0,20))
lines(c(h_Silverman, h_Silverman), c(0, 50), col='red')
lines(c(h_Terrell, h_Terrell), c(0, 50), col='blue')
```

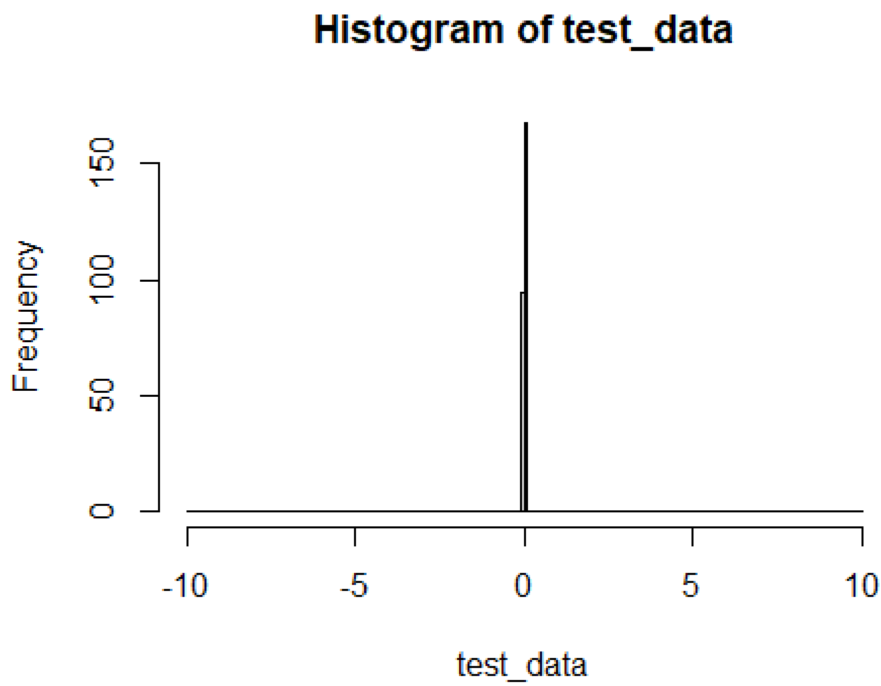
```
lines(c(h_SJ, h_SJ), c(0, 50), col='purple')
legend(.001,5,legend=c('Silverman', 'Terrell', 'Sheather-Jones'),
col=c('red', 'blue', 'purple'), lty=1)
```



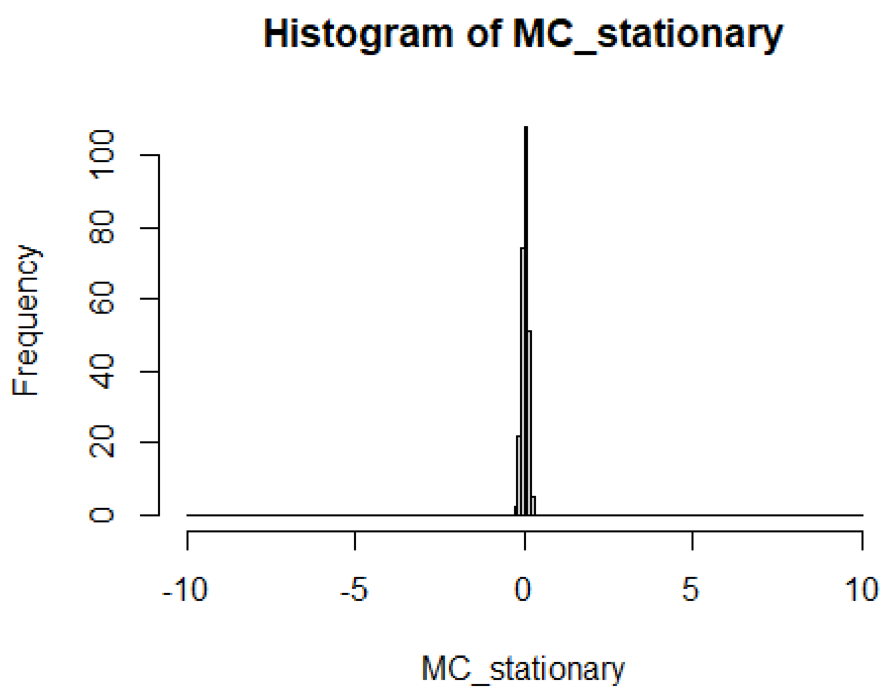

```
# effect of proposal var on delta
variances = c(0.001, 0.002, 0.005, 0.01, .02, .05, 0.1, 0.2, 0.5, 1, 2)
deltas = c()
for (variance in variances) {
  deltas = c(deltas, validate(data, h_Silverman, sqrt(variance)))
}
```


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

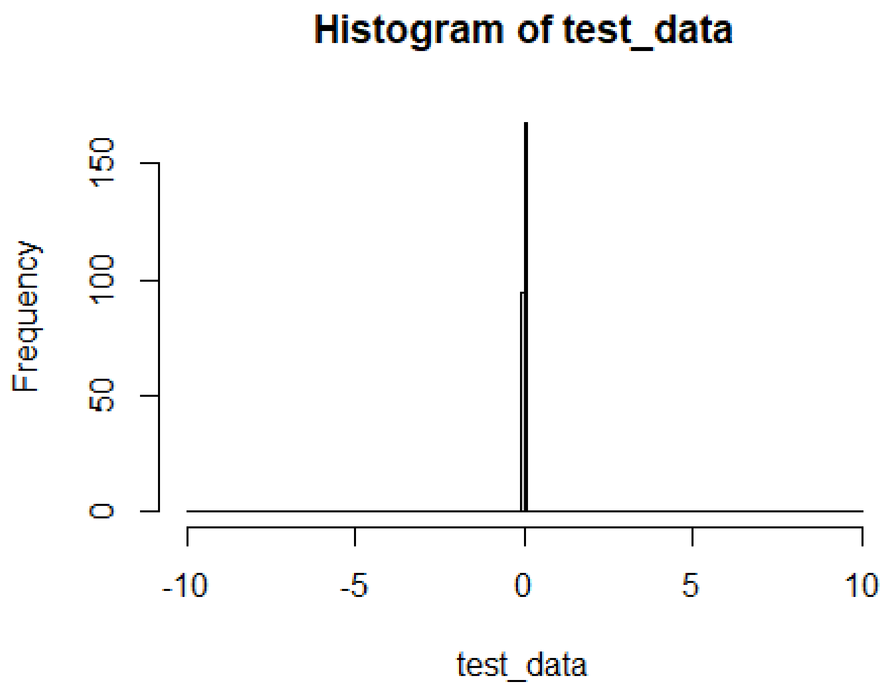

Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary

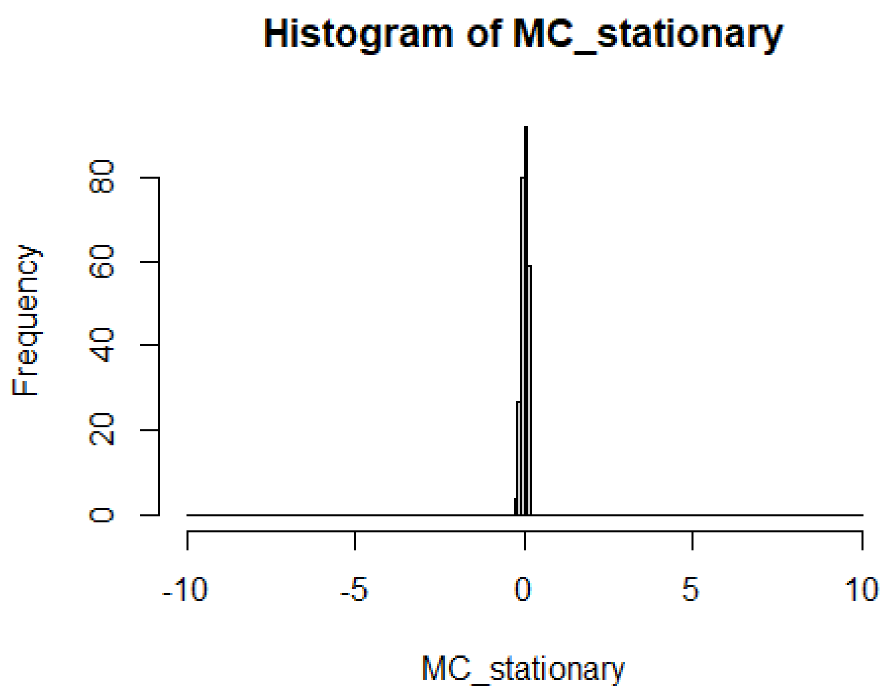


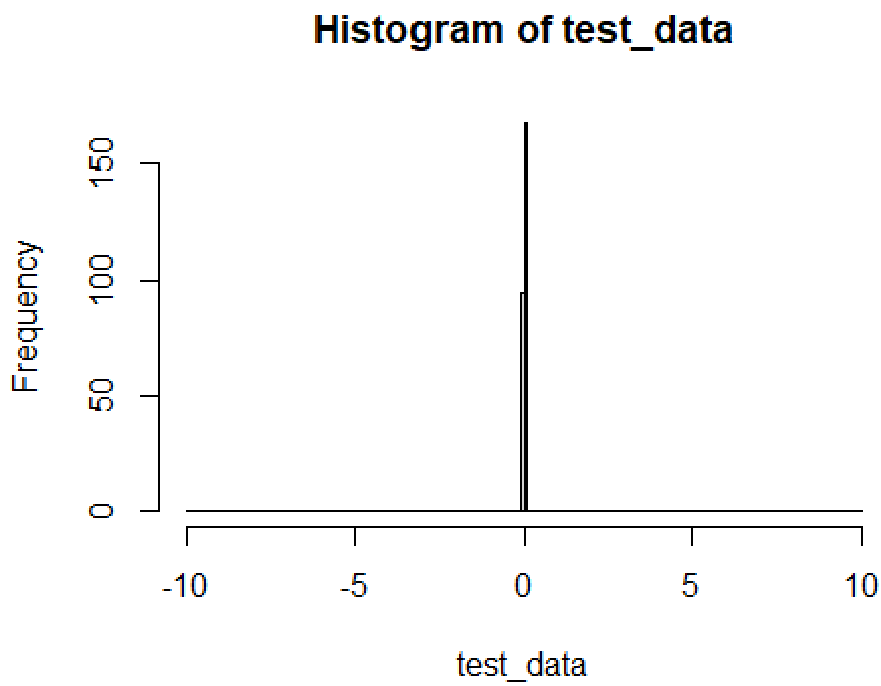
Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

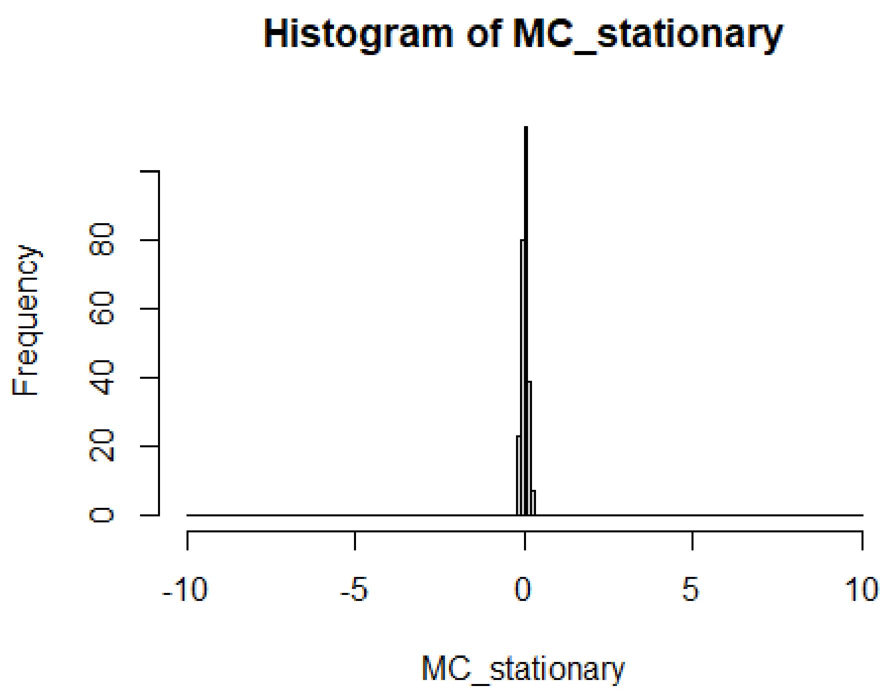


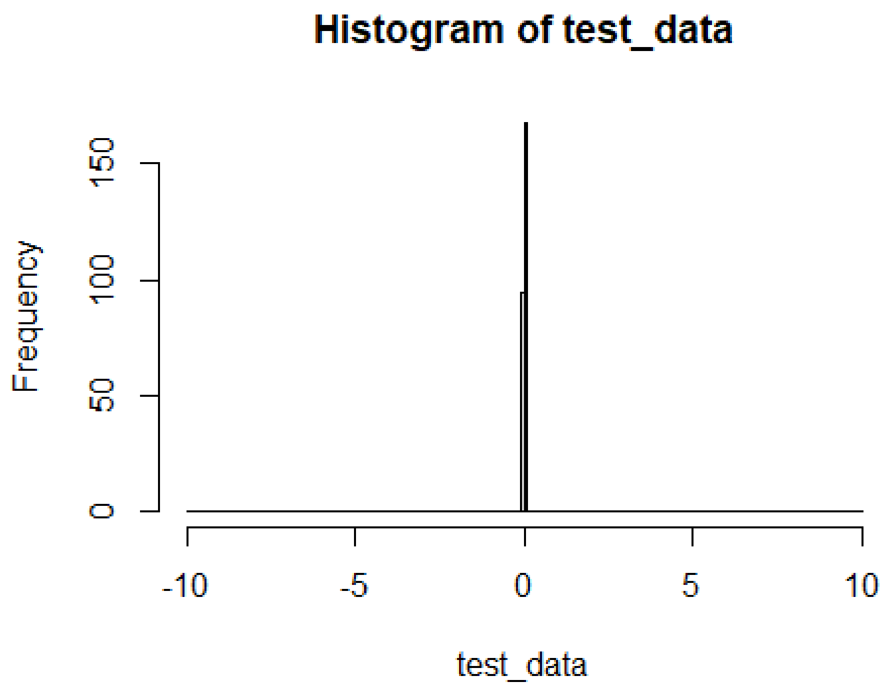
Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

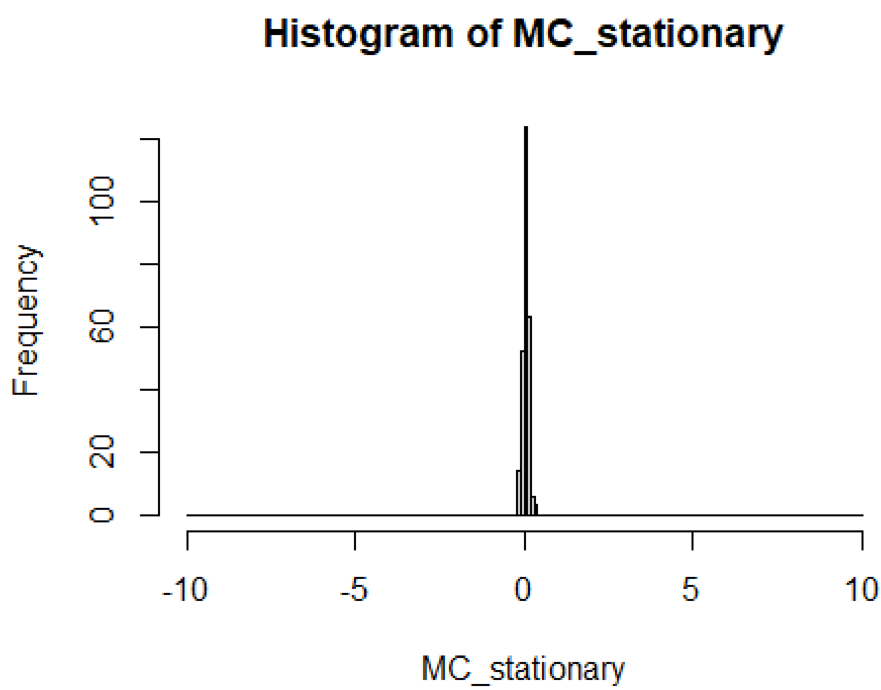

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

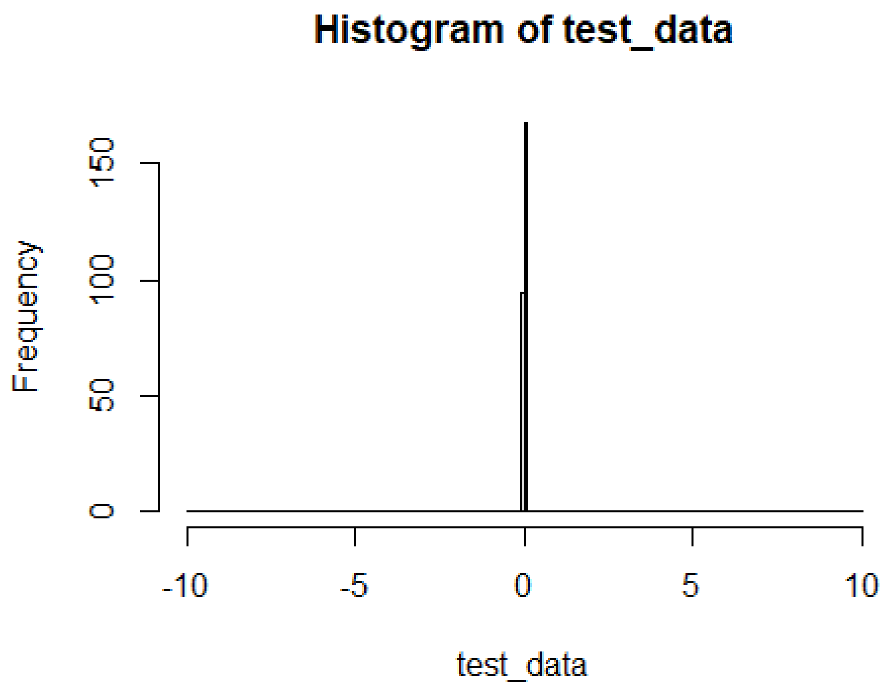

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary



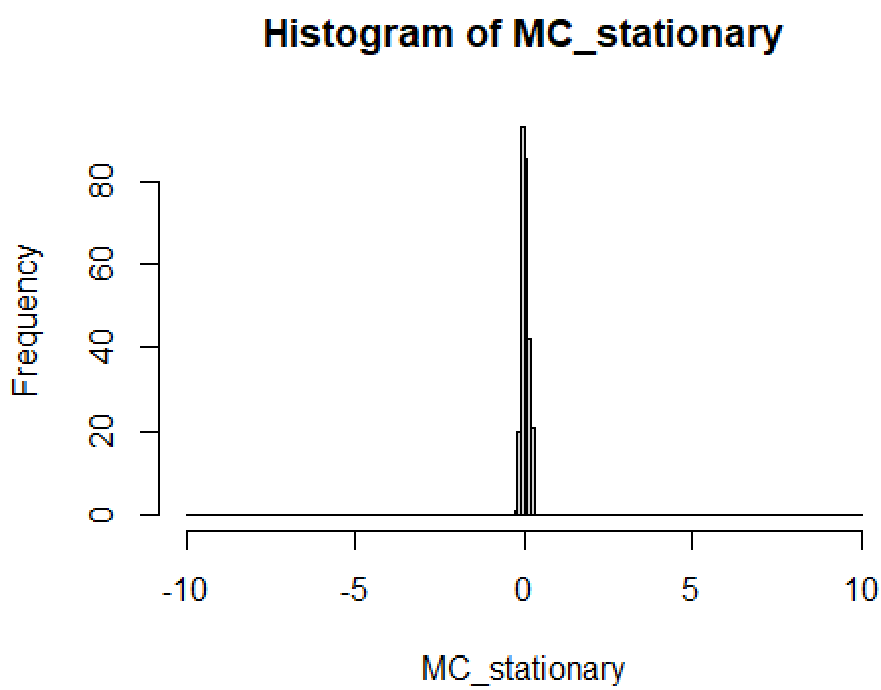
Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data



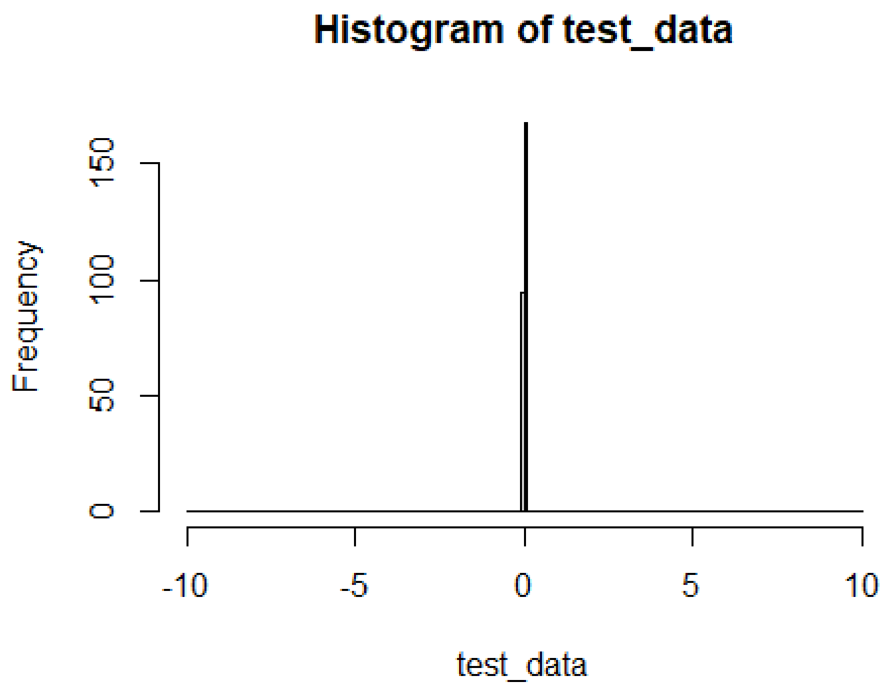
Histogram of MC_stationary
Frequency
0
20
60
100
-10
-5
0
5
10
MC_stationary



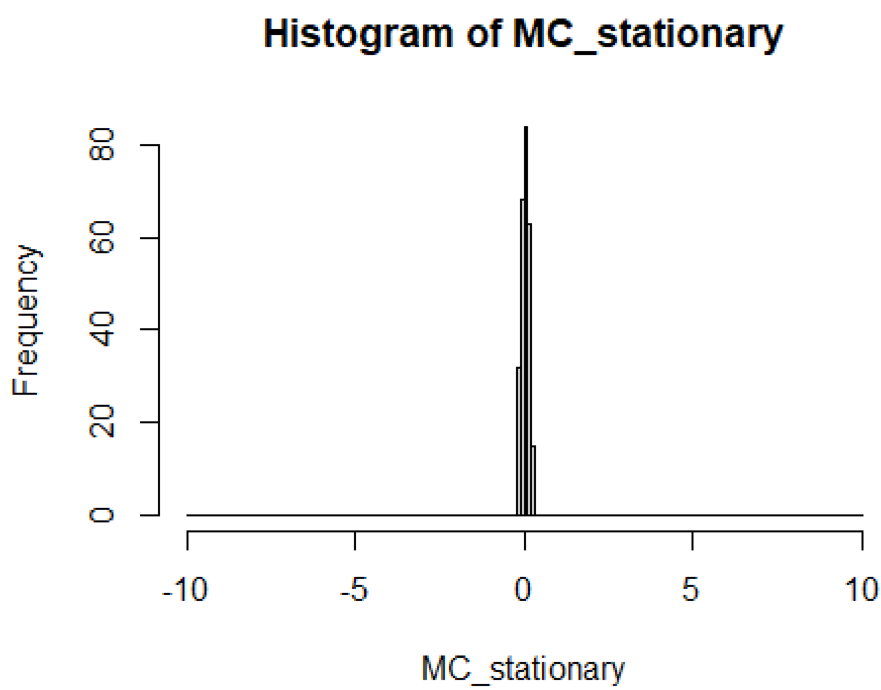
Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data



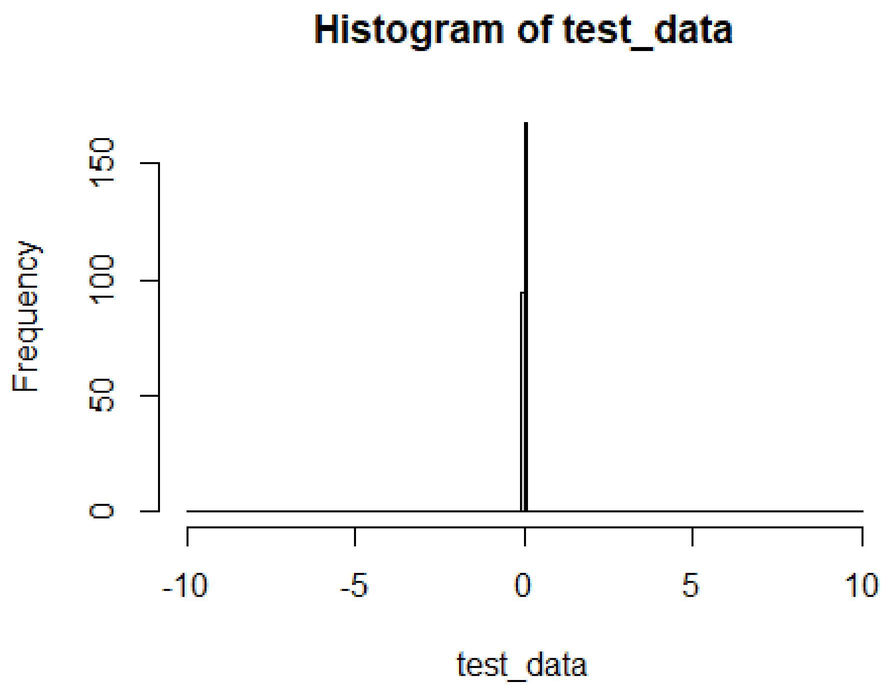
Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

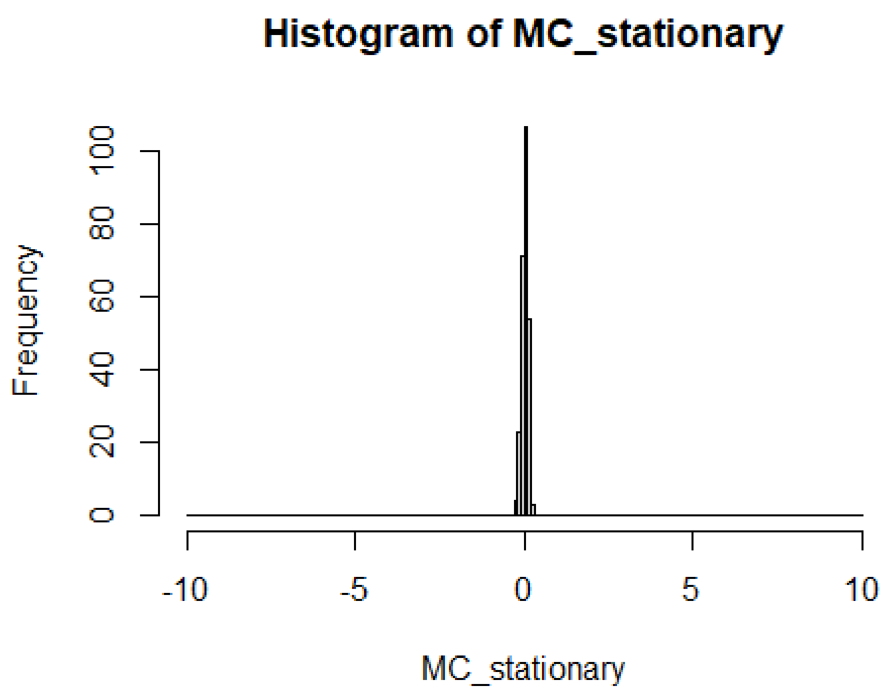


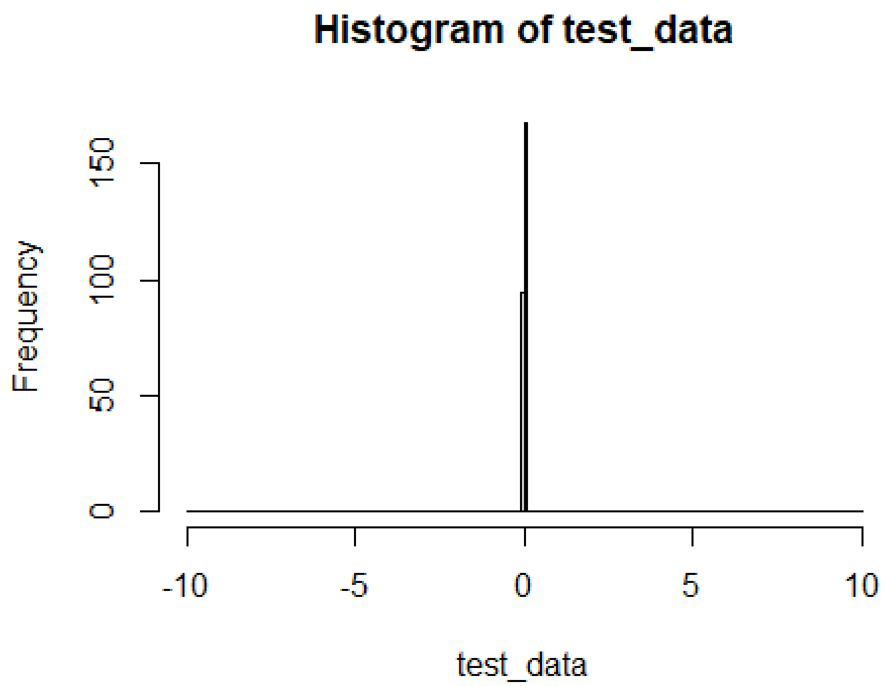
Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

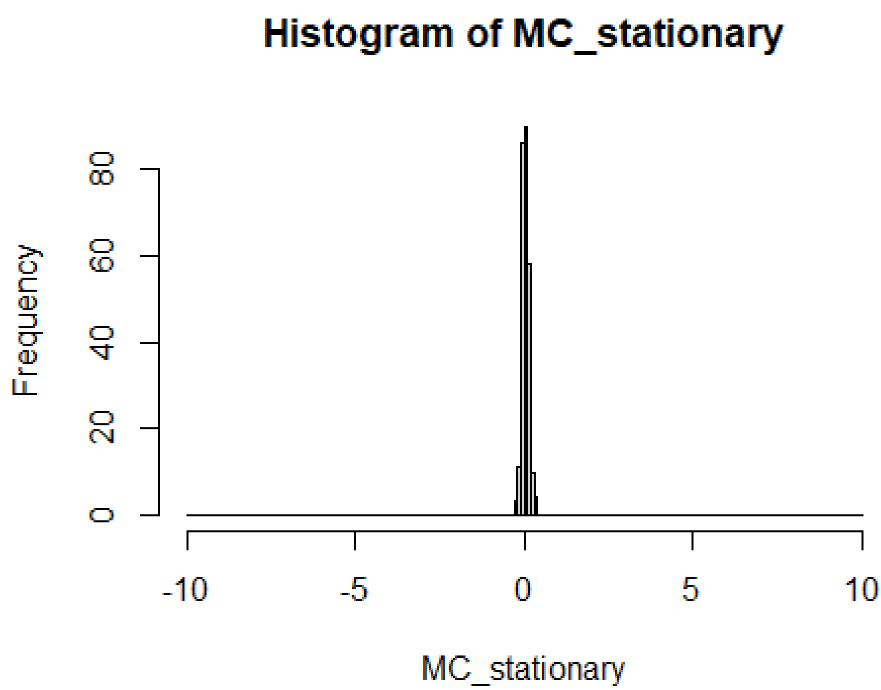


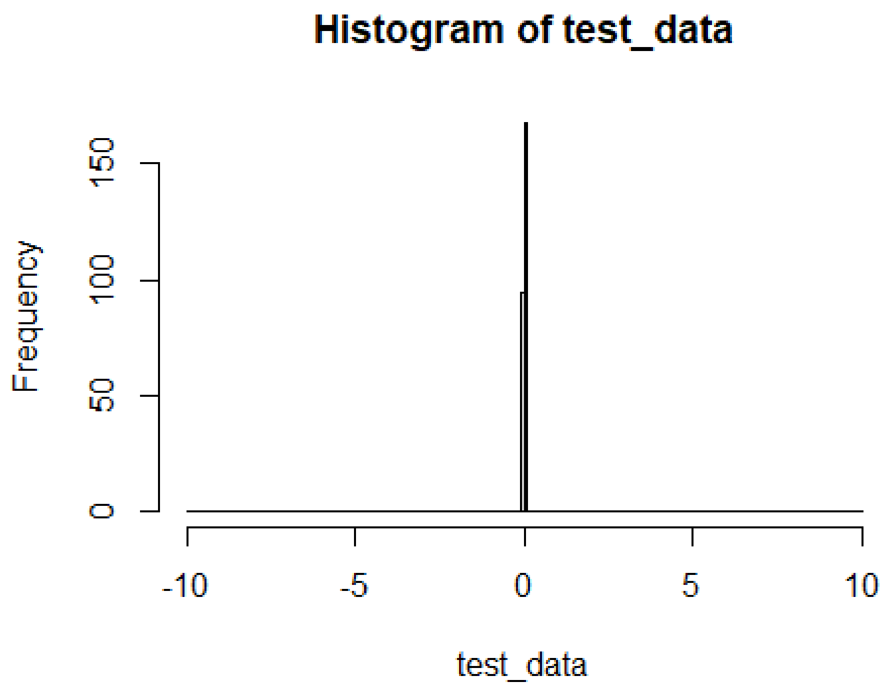
Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

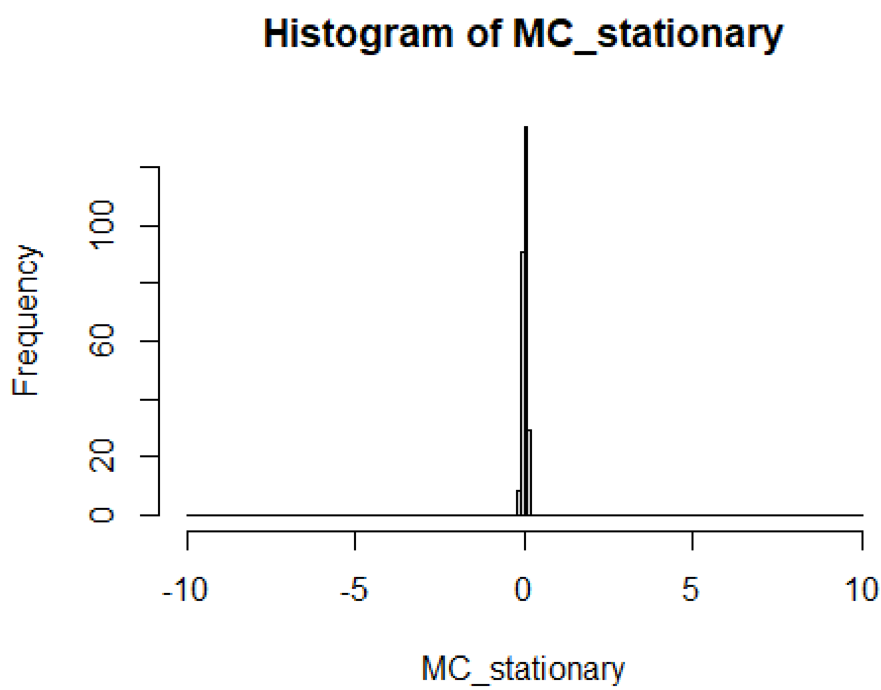

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

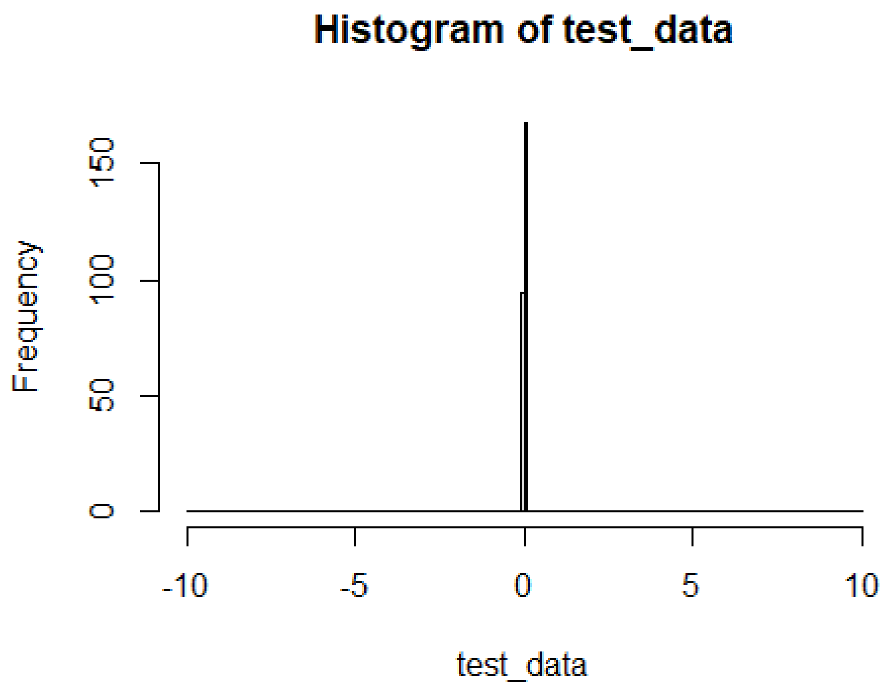

Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary

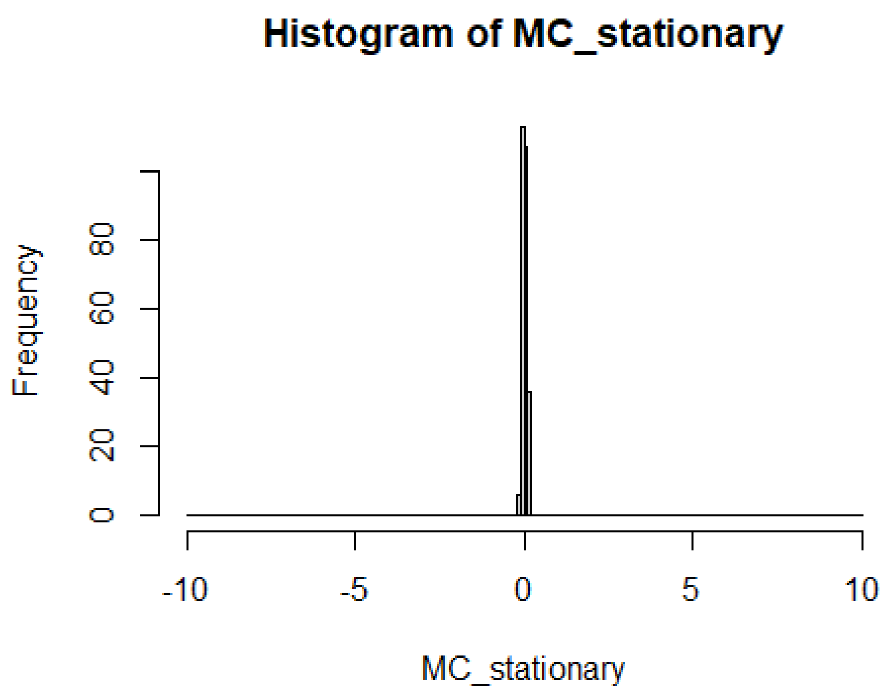

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

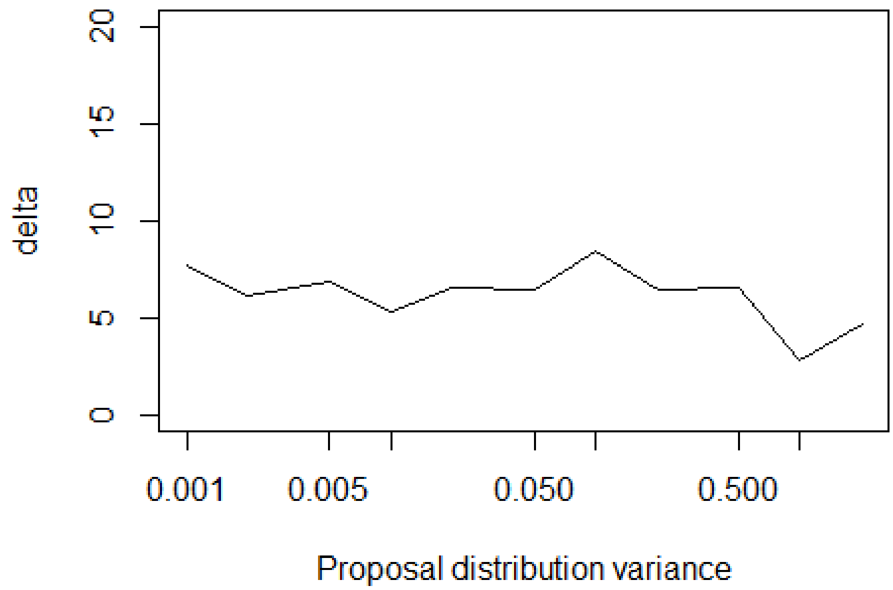

Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
20
60
100
-10
-5
0
5
10
MC_stationary

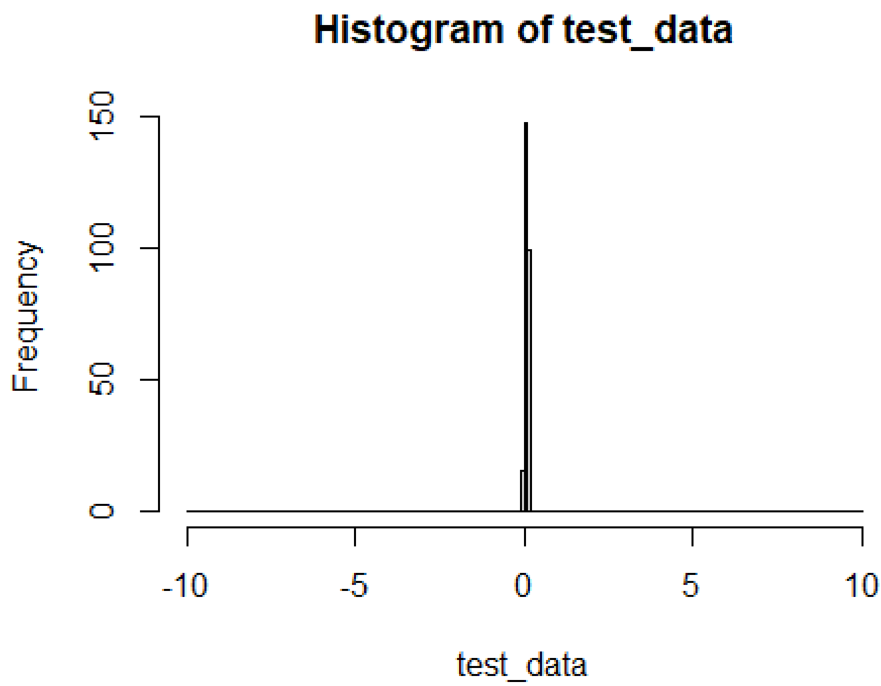


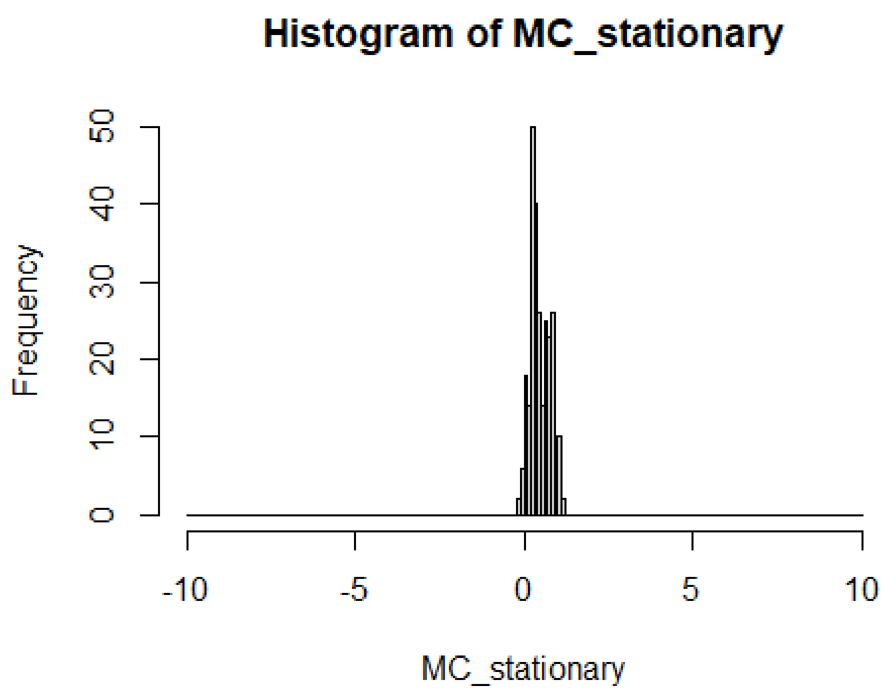



```
plot(variances, deltas, type='l', xlab = 'Proposal distribution variance',
ylab = 'delta', log = 'x', ylim=c(0,20))
```

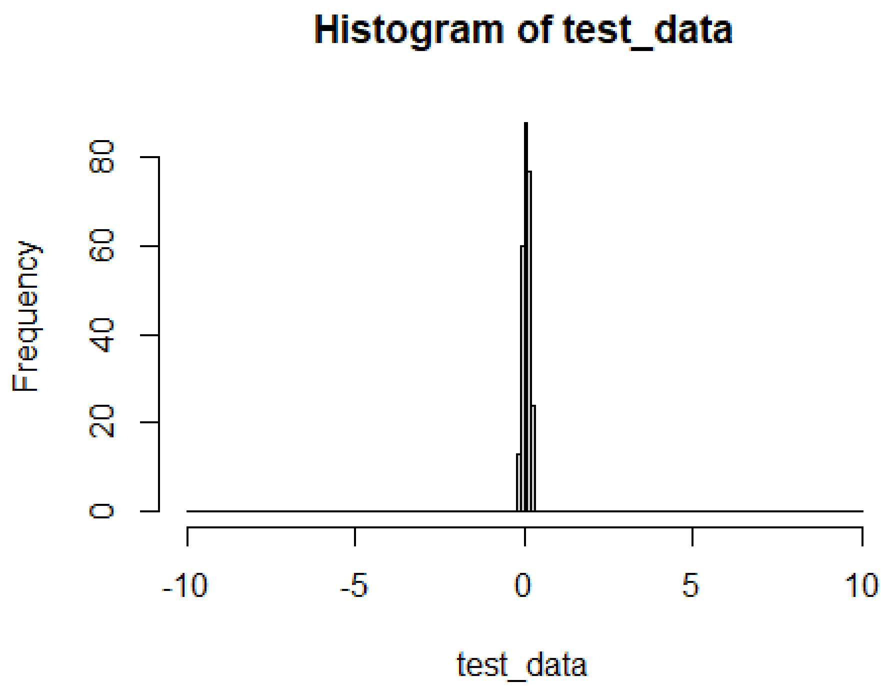


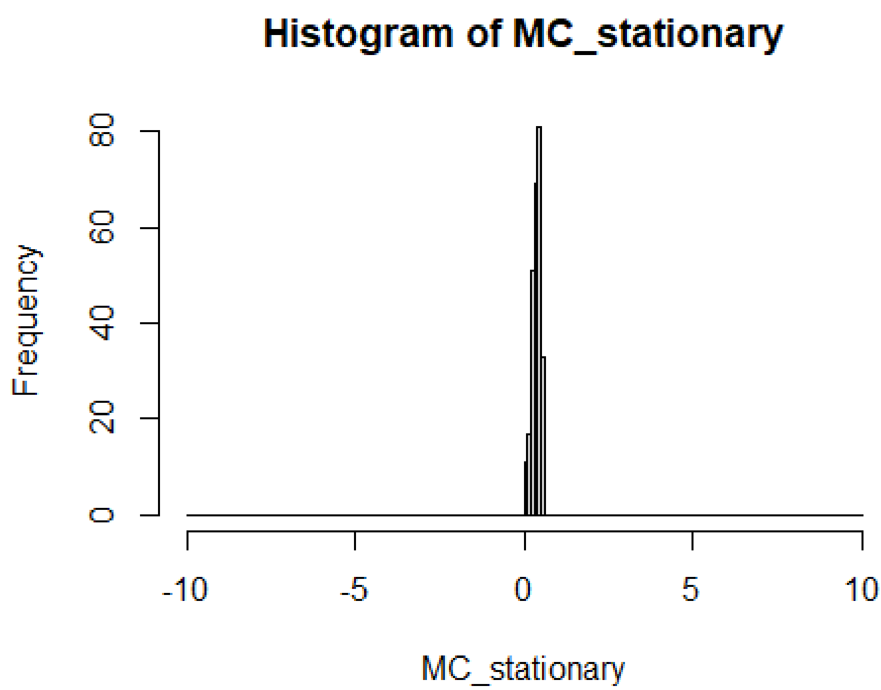

```
setwd('G:/My Drive/JHU/EN625664ComputationalStats/Final Project/data/')
files = dir()
deltas_by_stock = c()
for (f in files) {
  print(f)
  data<-read.csv(file=f,sep=",",header=TRUE)
  deltas_by_stock = c(deltas_by_stock, validate(data, h_Silverman, .5))
}

## [1] "AAPL.csv"
```


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

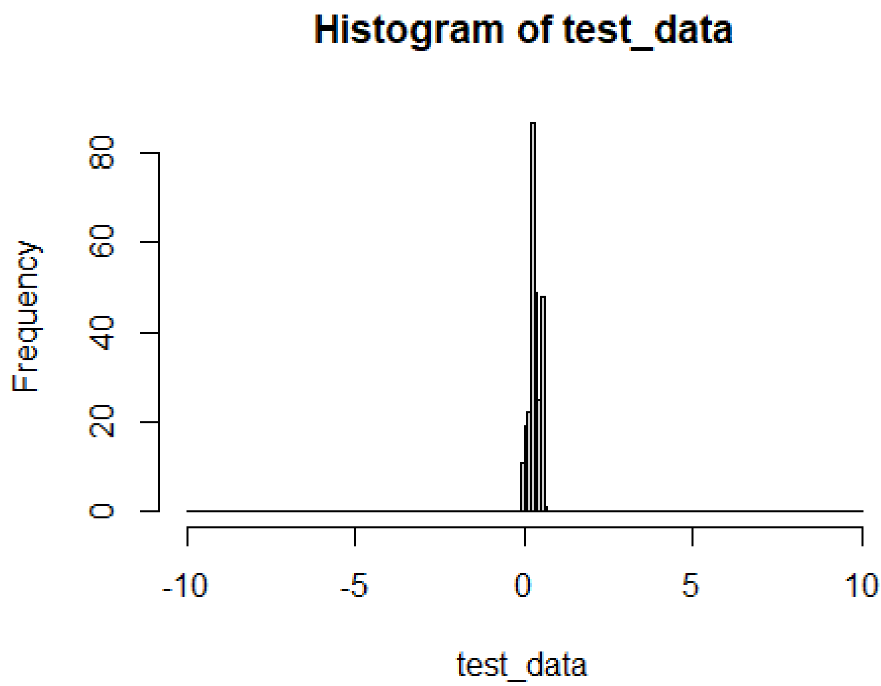

Histogram of MC_stationary
Frequency
0
10
20
30
40
50
-10
-5
0
5
10
MC_stationary

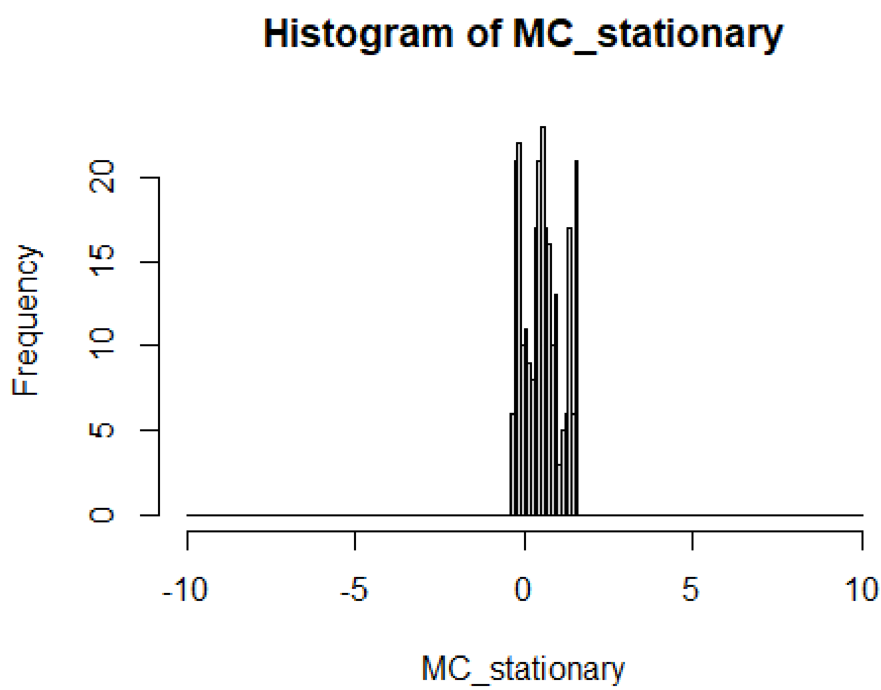
```
## [1] "AMGN.csv"
```


Histogram of test_data
Frequency
0
20
40
60
80
-10
-5
0
5
10
test_data

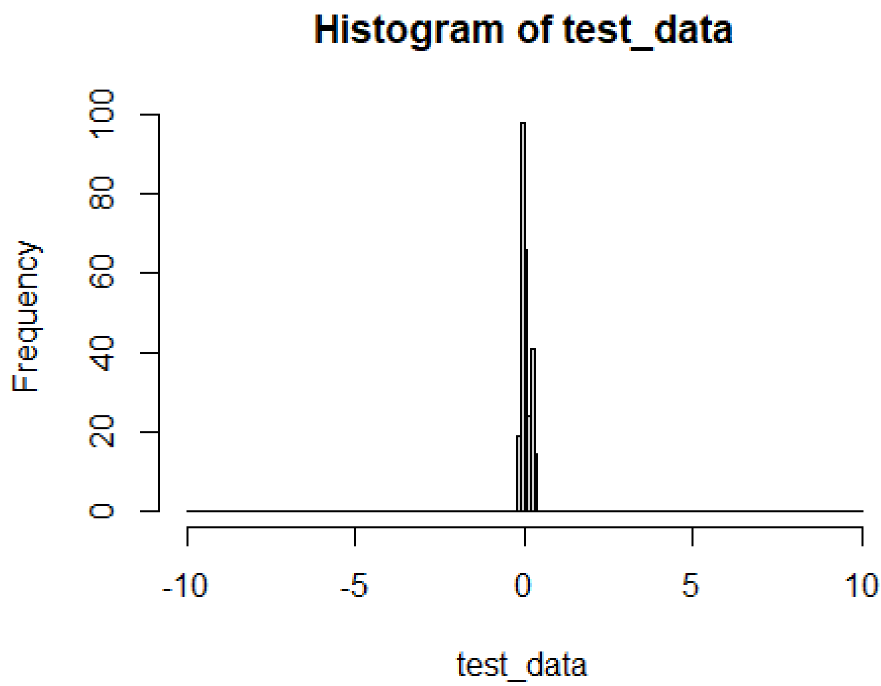

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

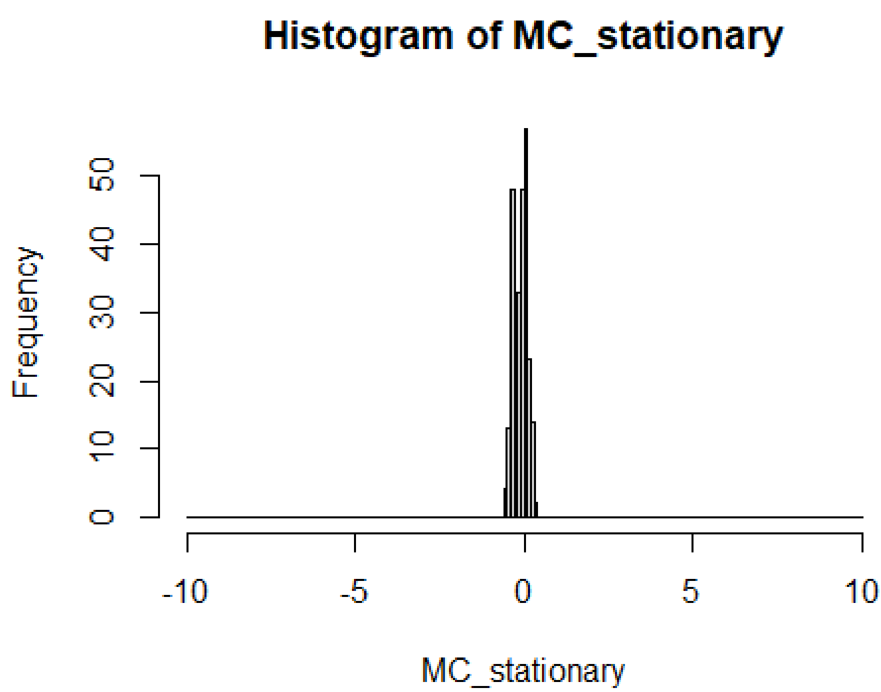
```
## [1] "AMZN.csv"
```


Histogram of test_data
Frequency
0
20
40
60
80
-10
-5
0
5
10
test_data

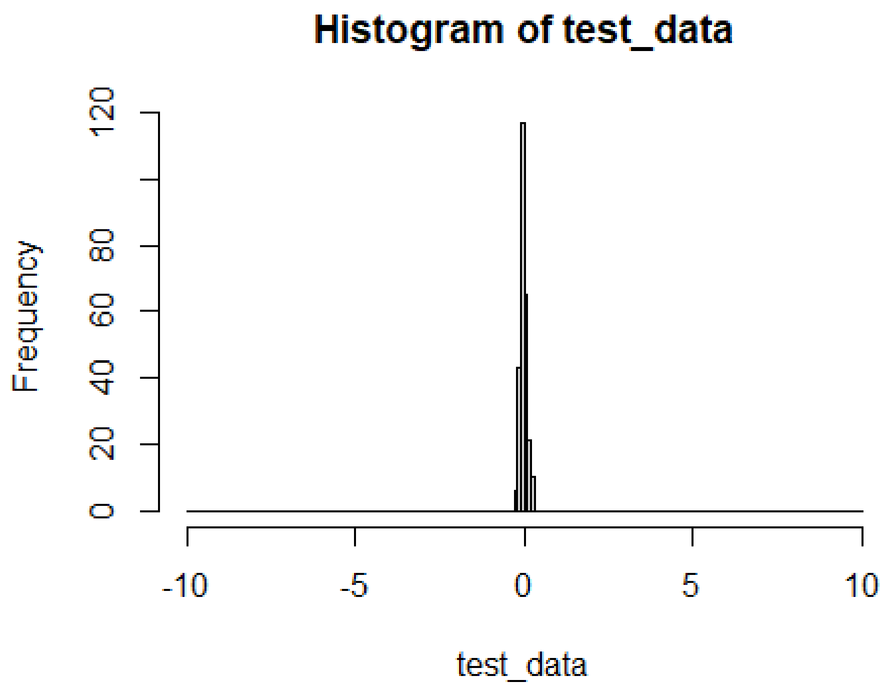

Histogram of MC_stationary
Frequency
0
5
10
15
20
-10
-5
0
5
10
MC_stationary

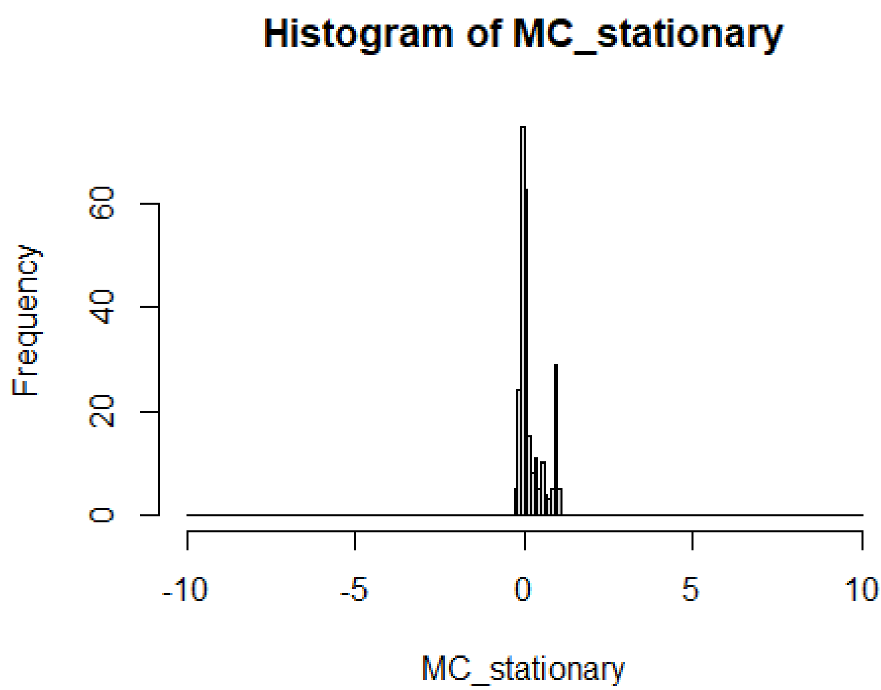
```
## [1] "AXP.csv"
```


Histogram of test_data
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
test_data

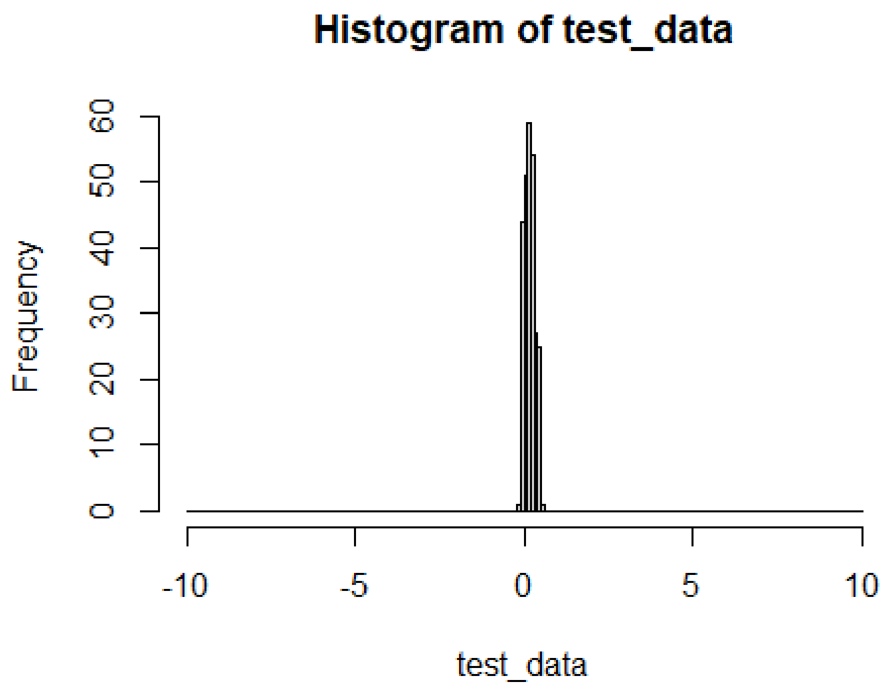

Histogram of MC_stationary
Frequency
0
10
20
30
40
50
-10
-5
0
5
10
MC_stationary

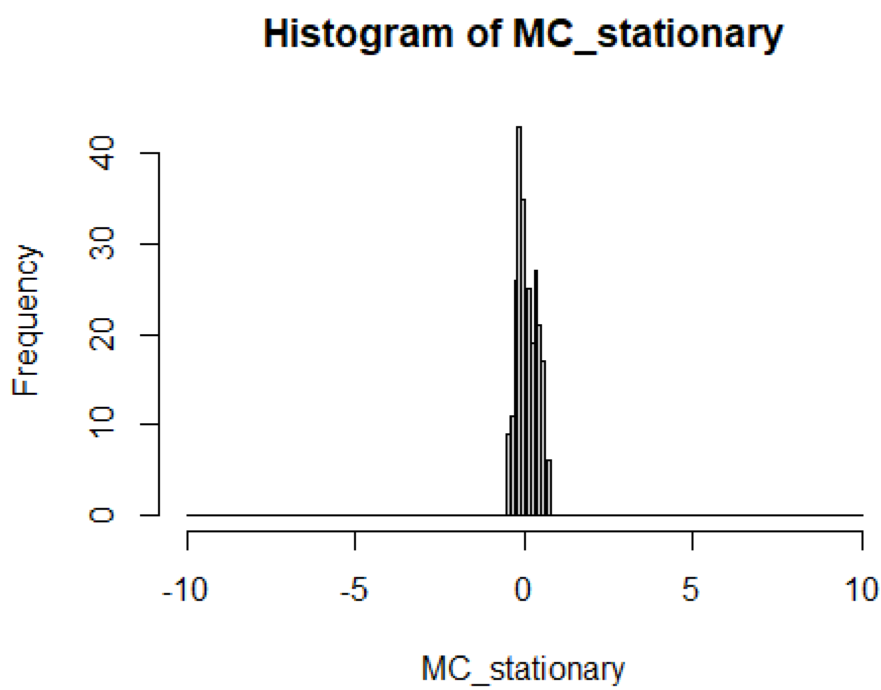
```
## [1] "BA.csv"
```


Histogram of test_data
Frequency
0
20
40
60
80
120
-10
-5
0
5
10
test_data

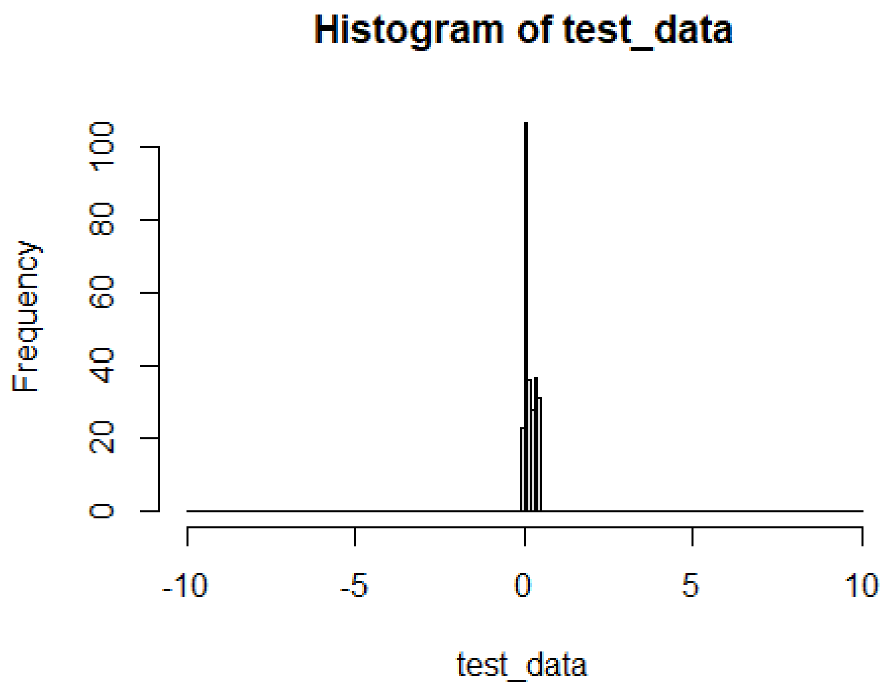

Histogram of MC_stationary
Frequency
0
20
40
60
-10
-5
0
5
10
MC_stationary

```
## [1] "CAT.csv"
```

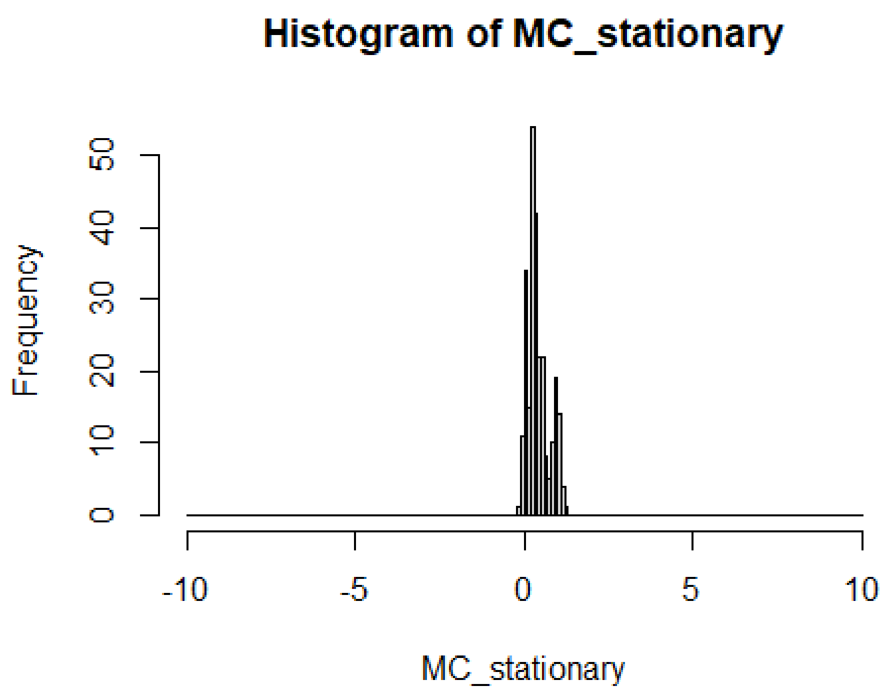

Histogram of test_data
Frequency
0
10
20
30
40
50
60
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
10
20
30
40
-10
-5
0
5
10
MC_stationary

```
## [1] "CRM.csv"
```

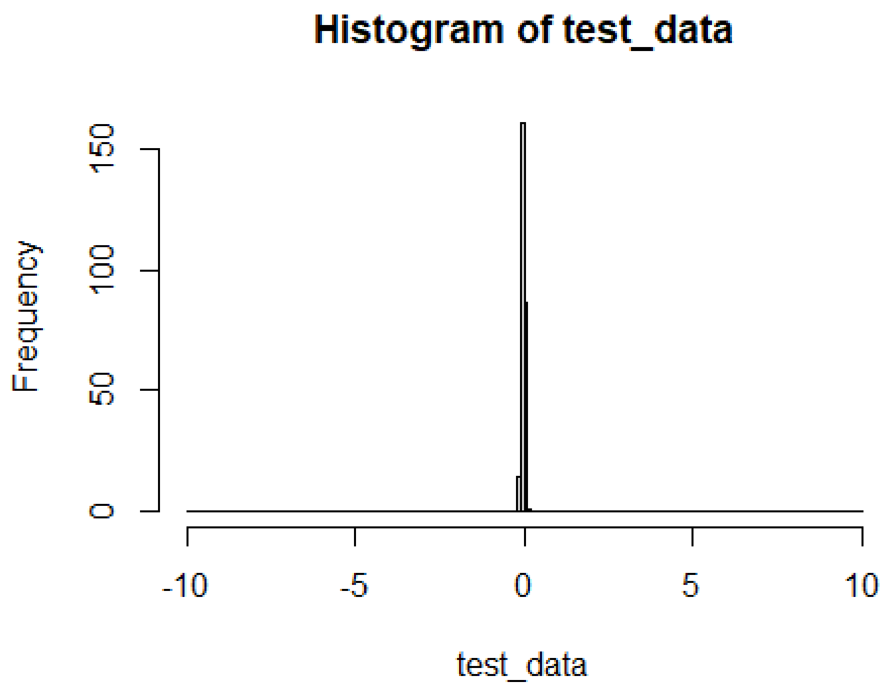


Histogram of test_data
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
test_data

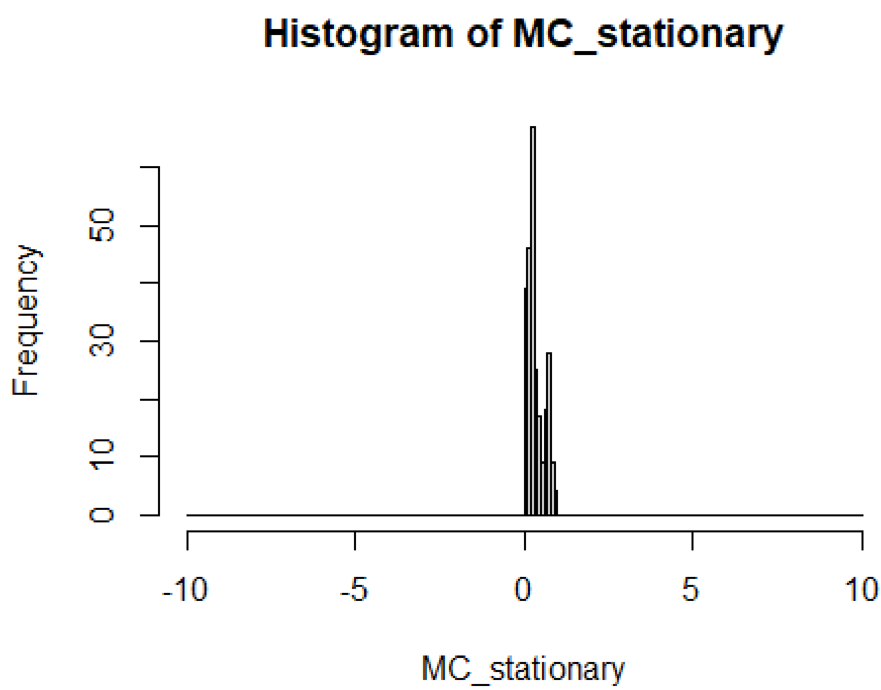


Histogram of MC_stationary
Frequency
0
10
20
30
40
50
-10
-5
0
5
10
MC_stationary


```
## [1] "CSCO.csv"
```

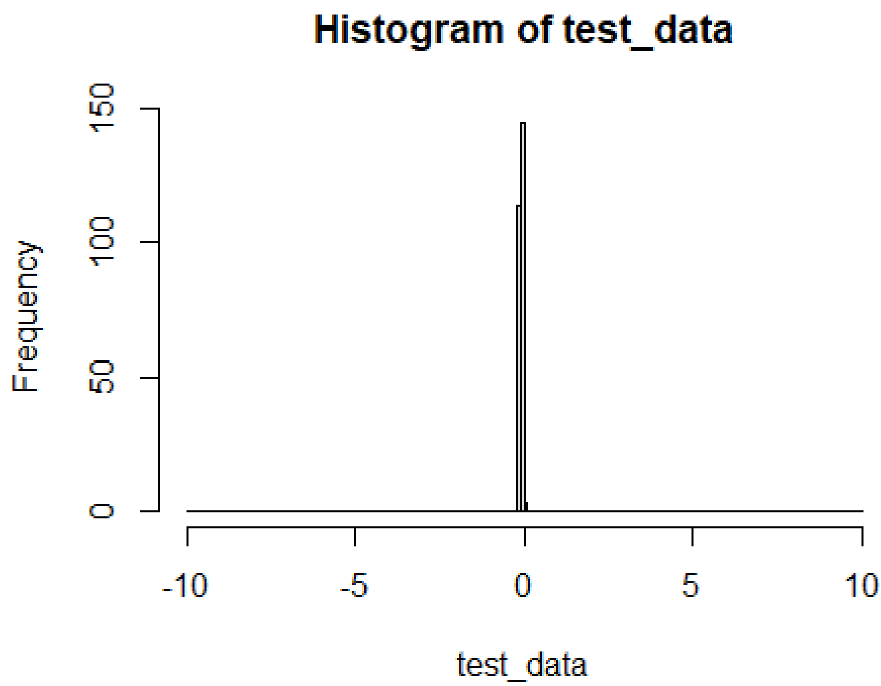


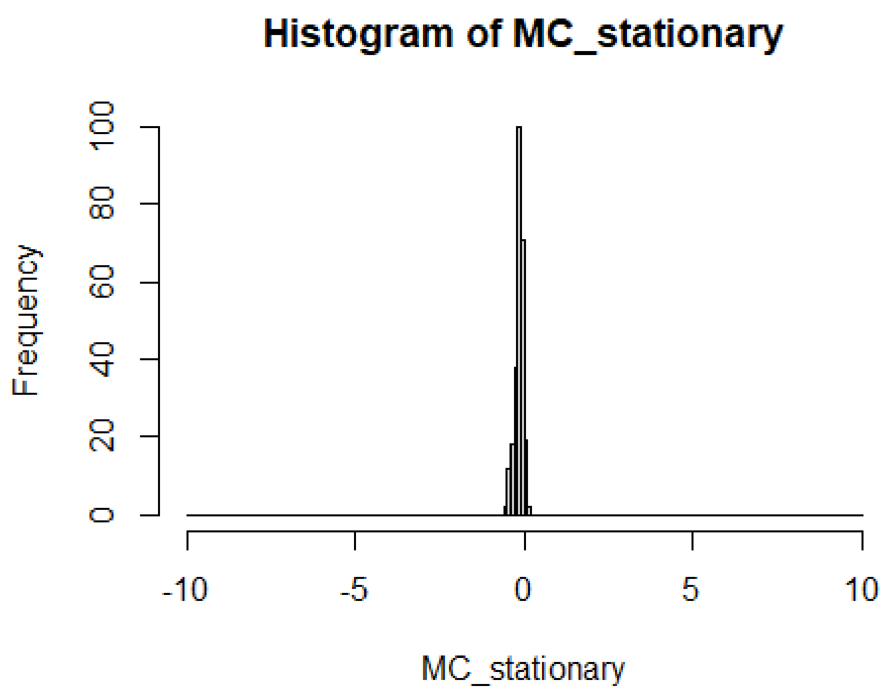
Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data



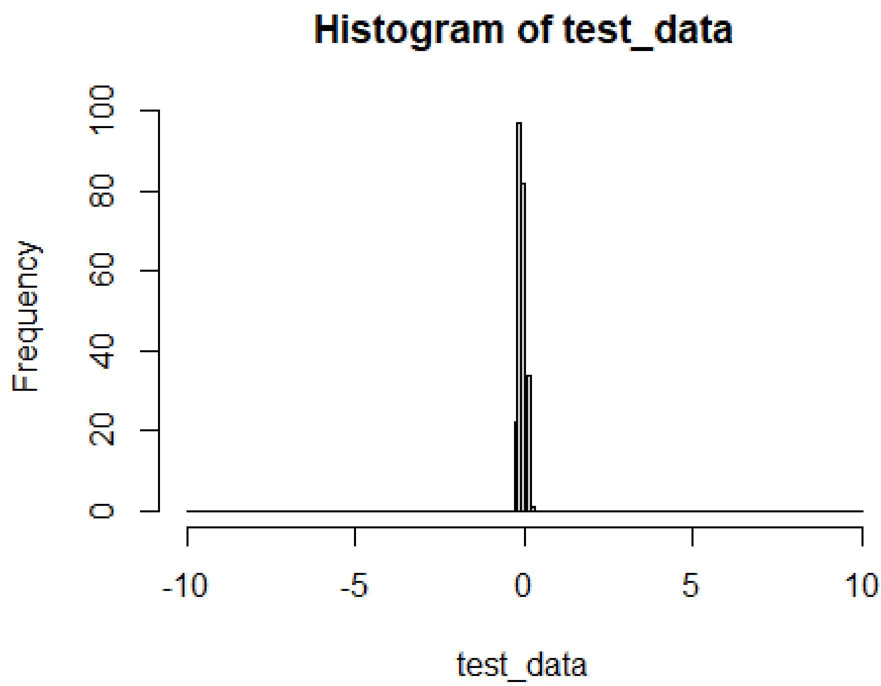
Histogram of MC_stationary
Frequency
0
10
30
50
-10
-5
0
5
10
MC_stationary

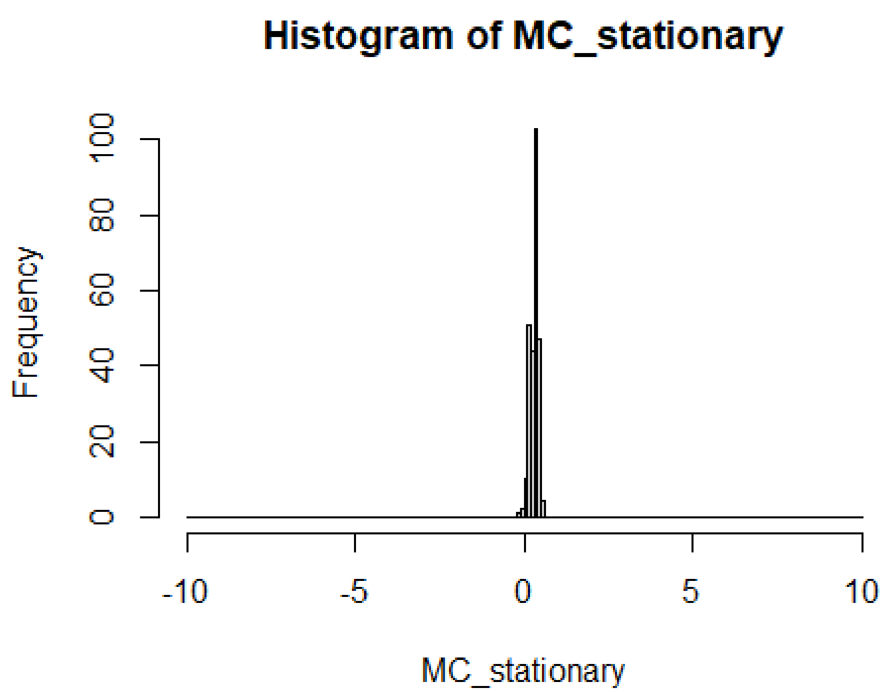

```
## [1] "CVX.csv"
```


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

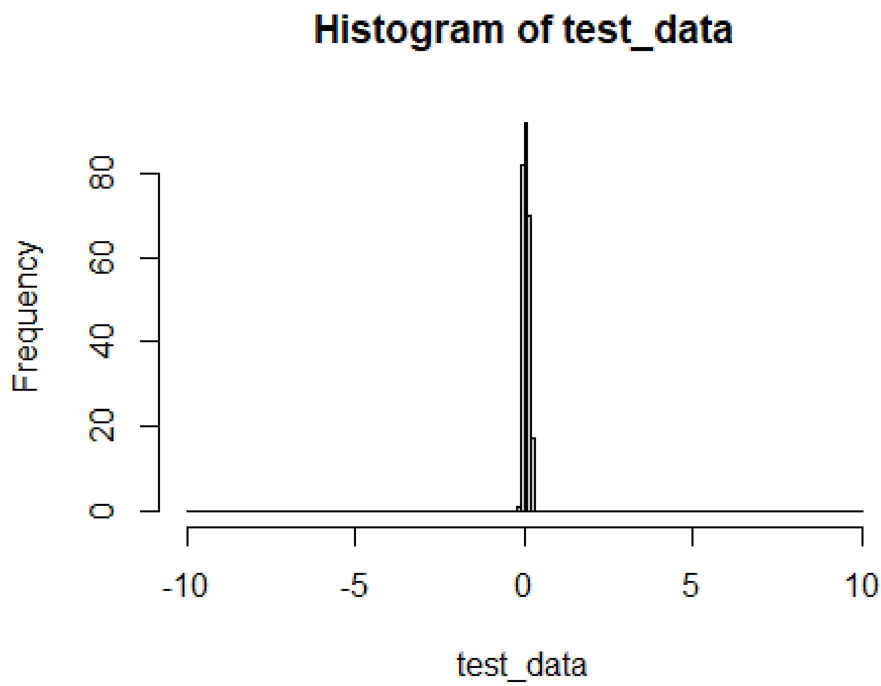

Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary

```
## [1] "DIS.csv"
```

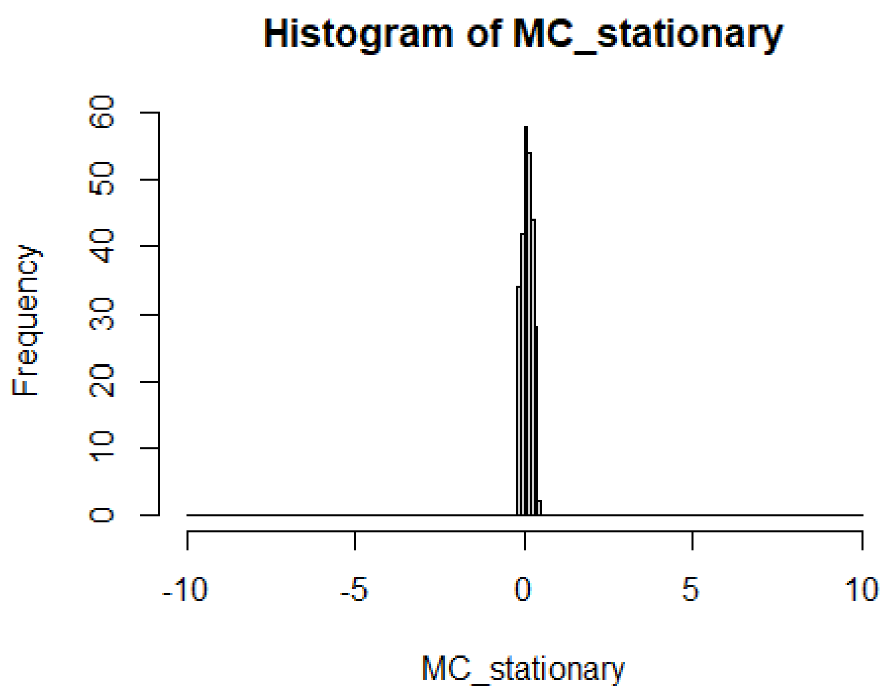

Histogram of test_data
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary

```
## [1] "GS.csv"
```

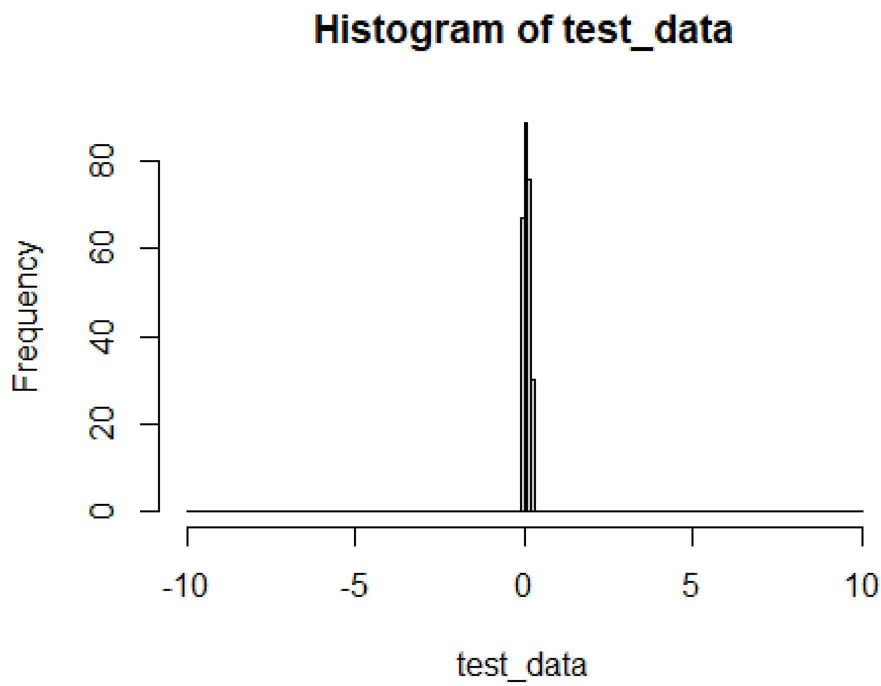


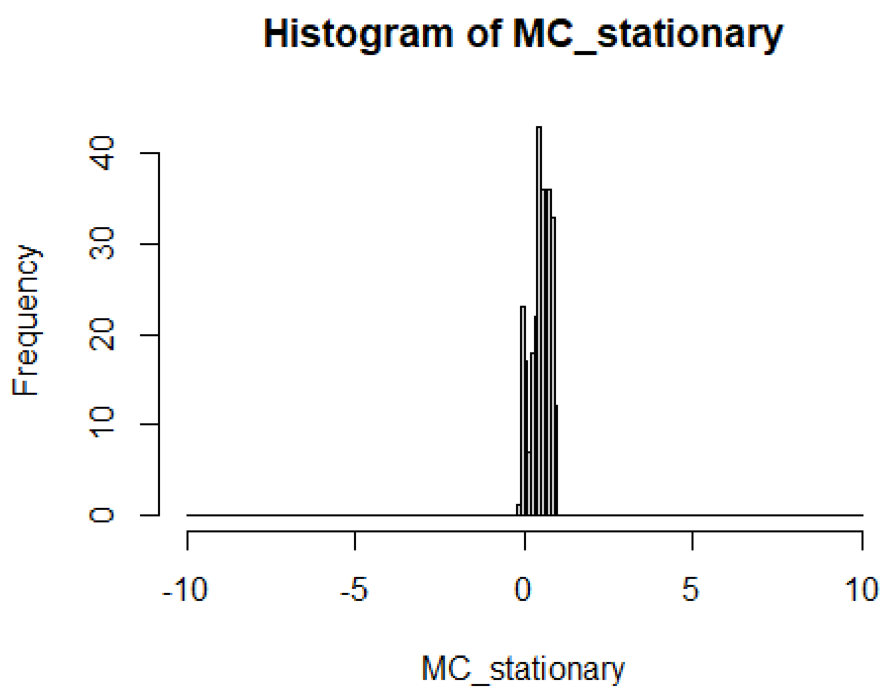
Histogram of test_data
Frequency
0
20
40
60
80
-10
-5
0
5
10
test_data



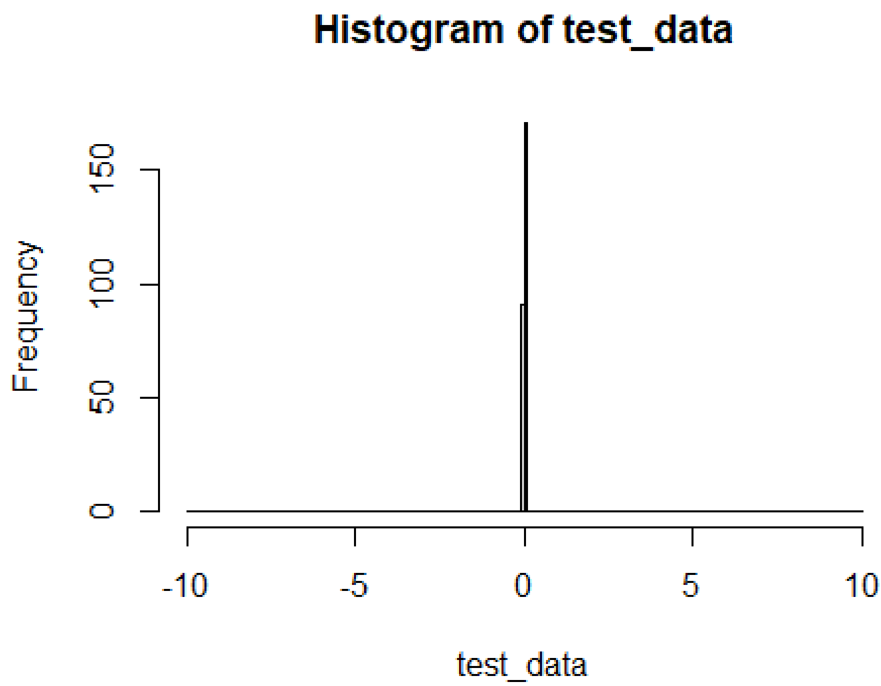
Histogram of MC_stationary
Frequency
0
10
20
30
40
50
60
-10
-5
0
5
10
MC_stationary


```
## [1] "HD.csv"
```

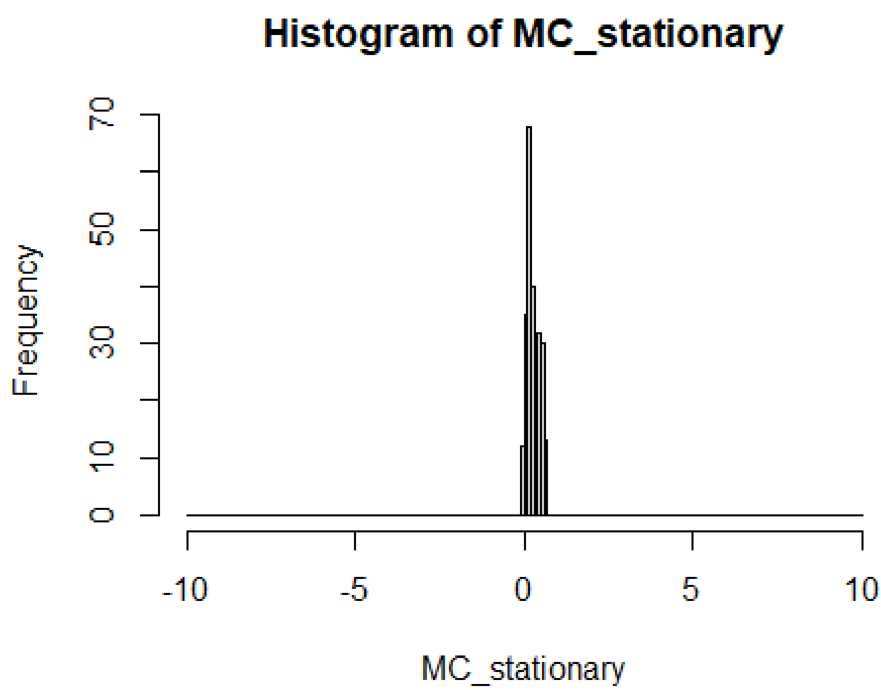

Histogram of test_data
Frequency
0
20
40
60
80
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
10
20
30
40
-10
-5
0
5
10
MC_stationary

```
## [1] "HON.csv"
```

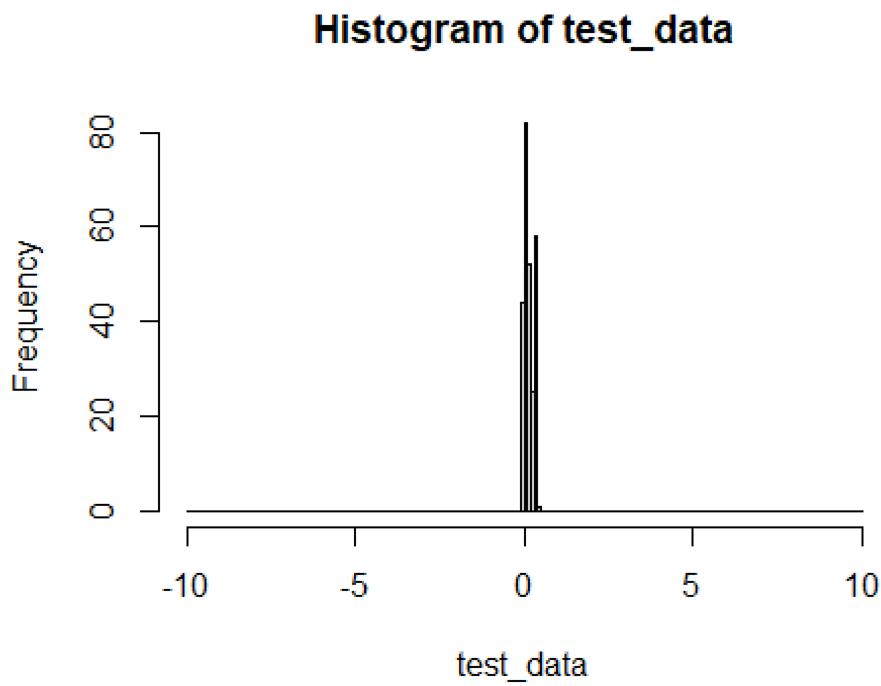


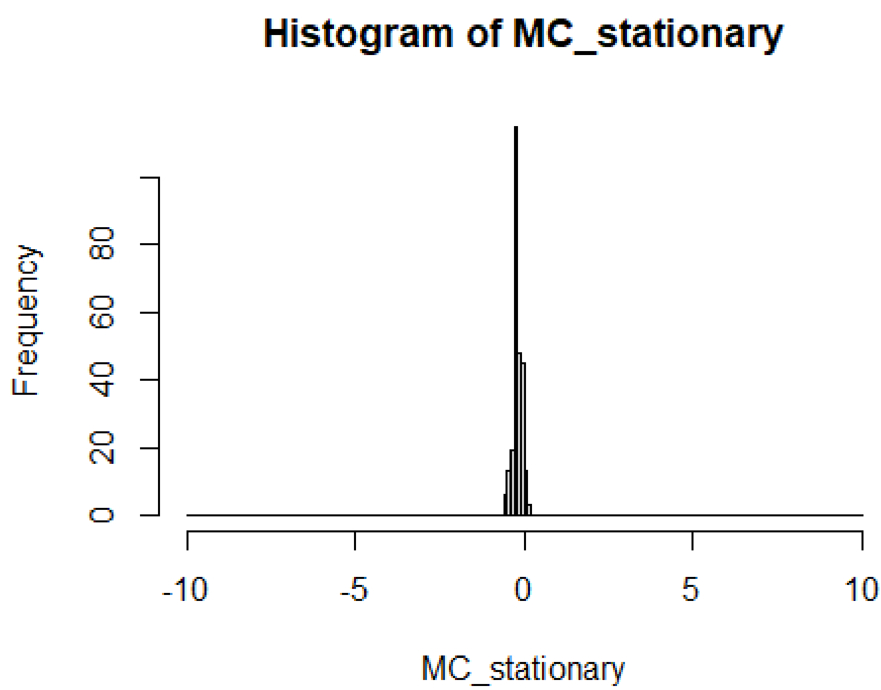
Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data



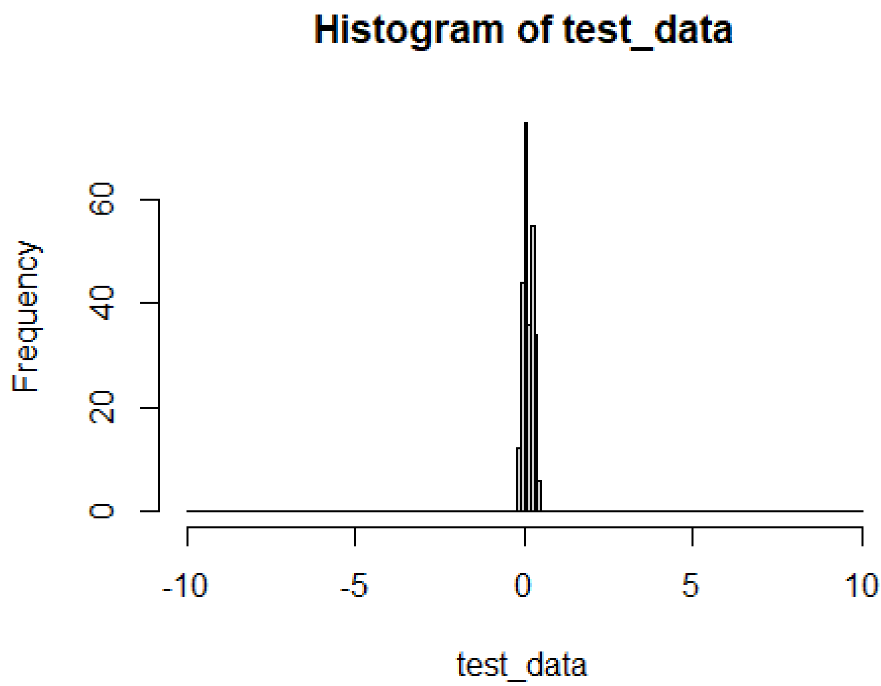
Histogram of MC_stationary
Frequency
0
10
30
50
70
-10
-5
0
5
10
MC_stationary


```
## [1] "IBM.csv"
```

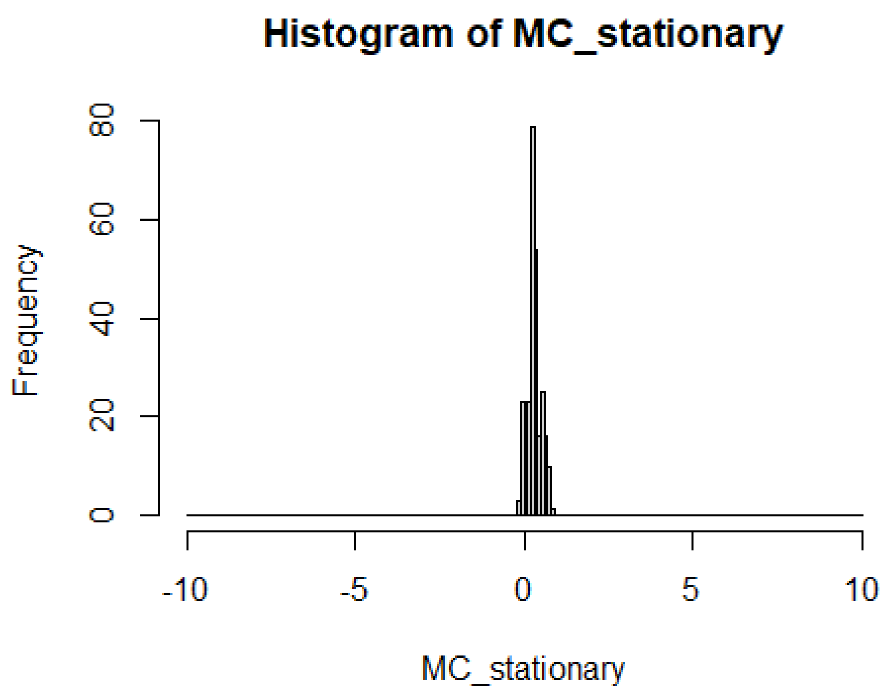

Histogram of test_data
Frequency
0
20
40
60
80
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

```
## [1] "INTC.csv"
```

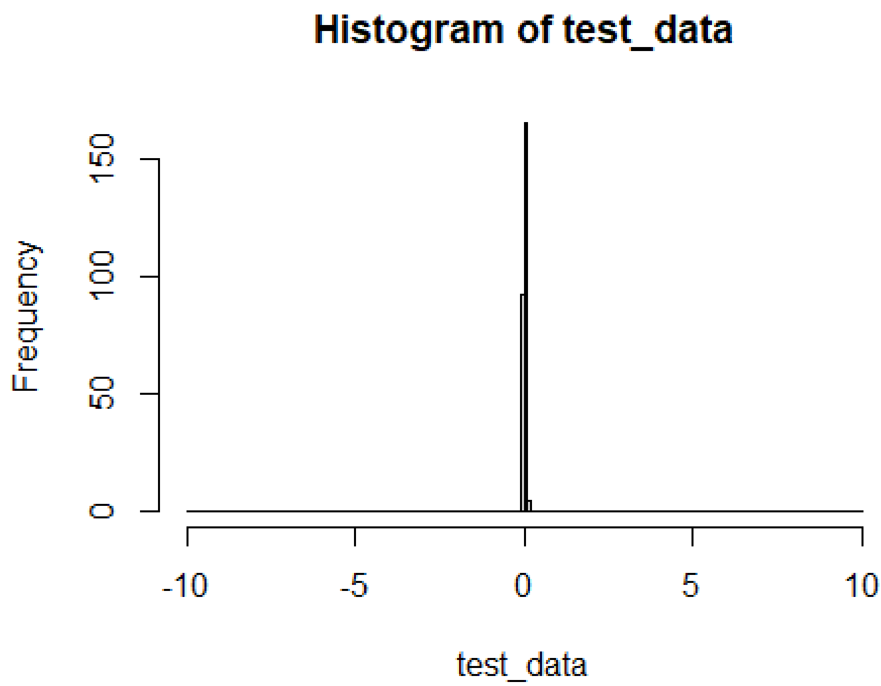


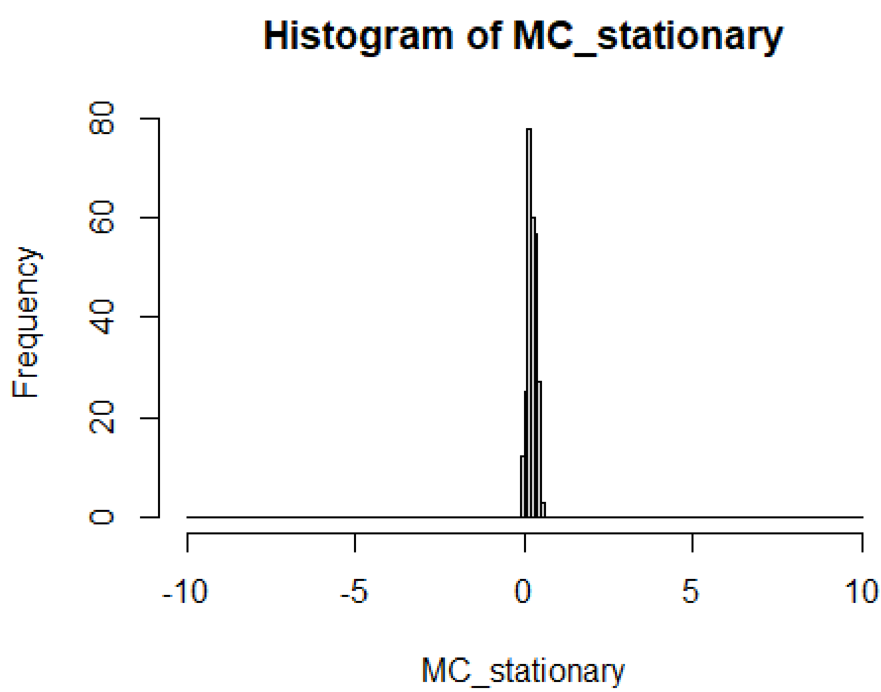
Histogram of test_data
Frequency
0
20
40
60
-10
-5
0
5
10
test_data



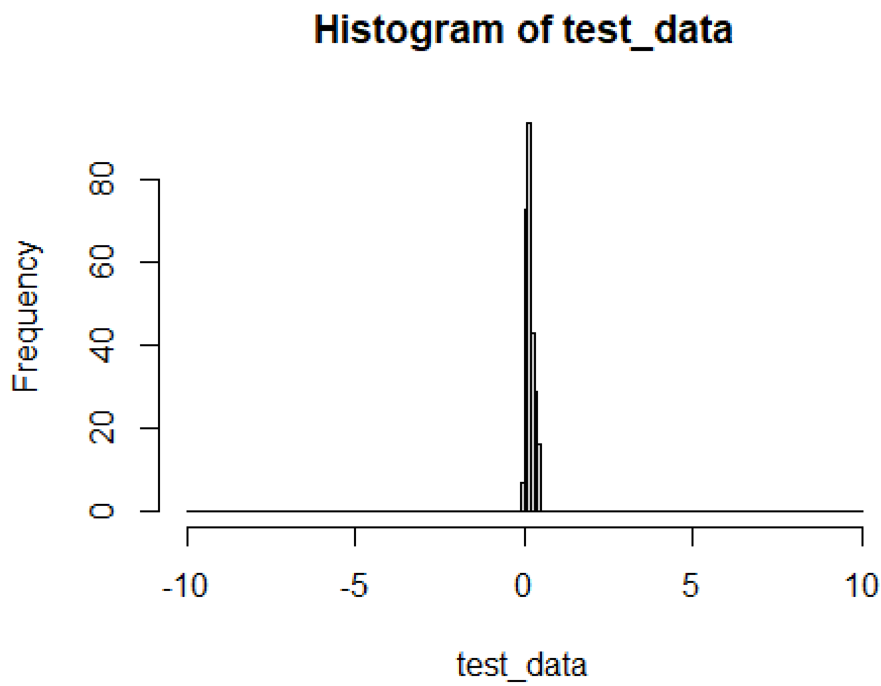
Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

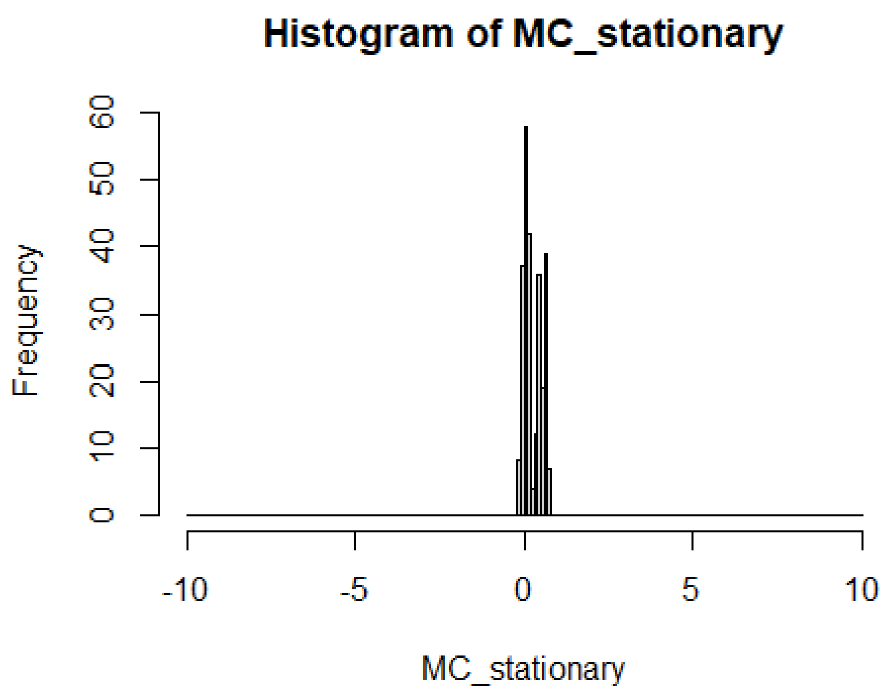

```
## [1] "JNJ.csv"
```


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

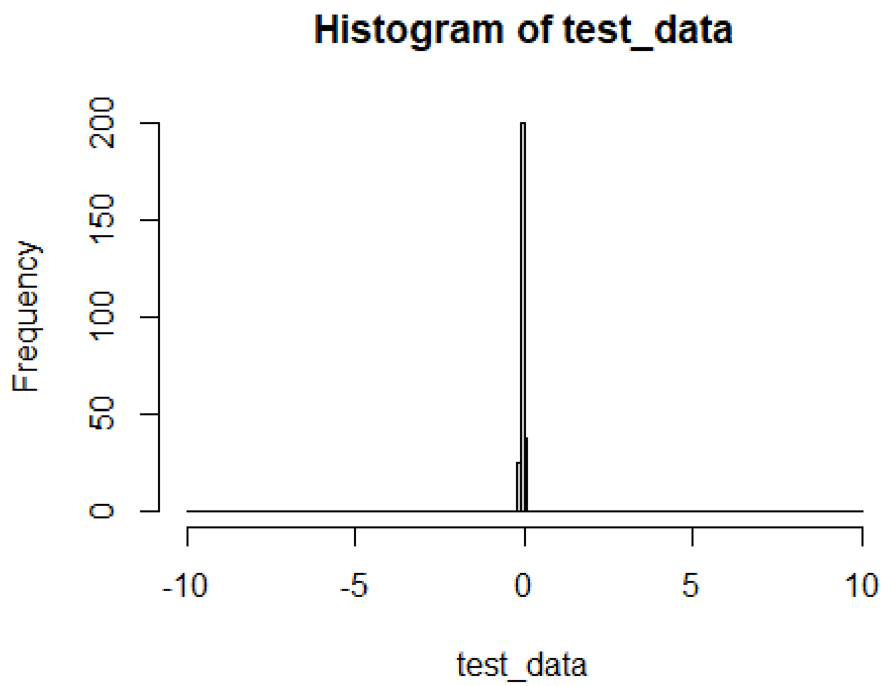

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

```
## [1] "JPM.csv"
```

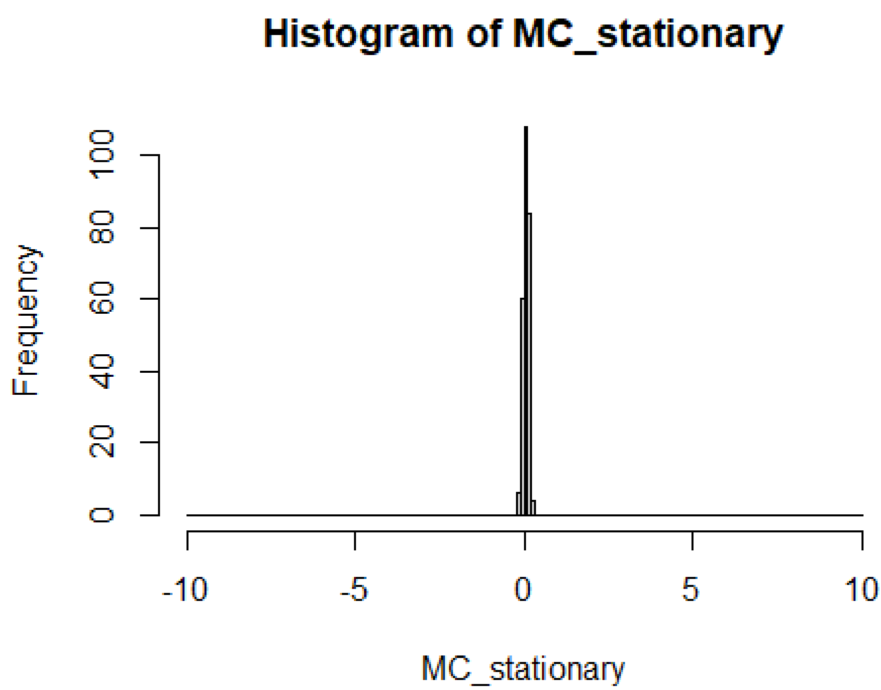

Histogram of test_data
Frequency
0
20
40
60
80
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
10
20
30
40
50
60
-10
-5
0
5
10
MC_stationary

```
## [1] "KO.csv"
```

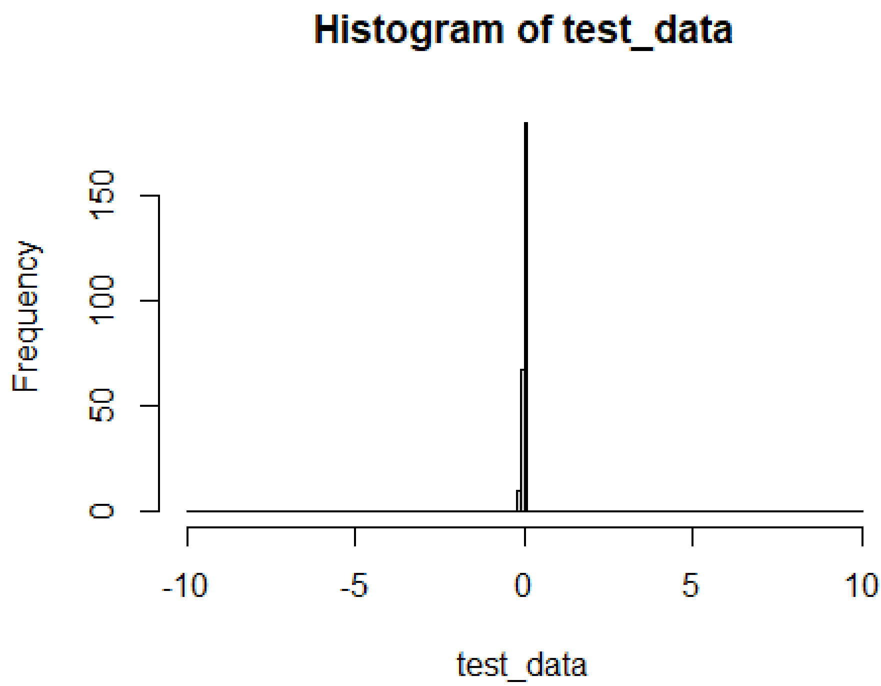


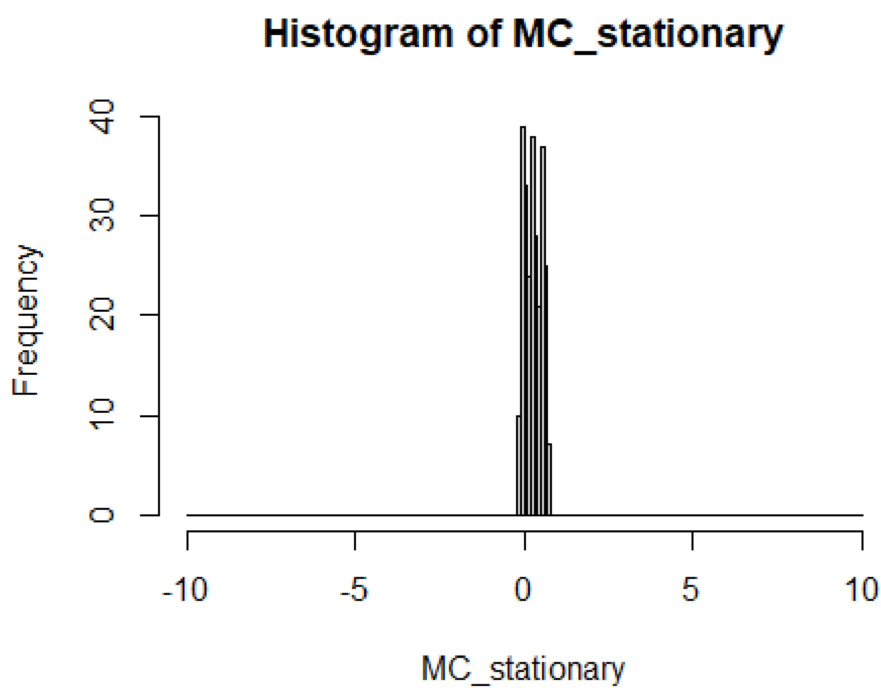
Histogram of test_data
Frequency
0
50
100
150
200
-10
-5
0
5
10
test_data



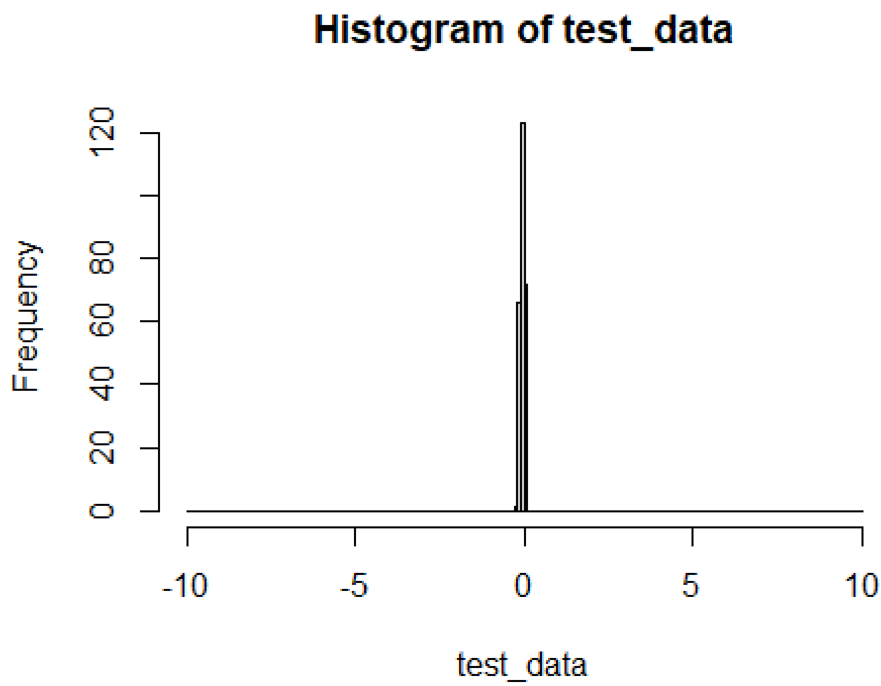
Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary

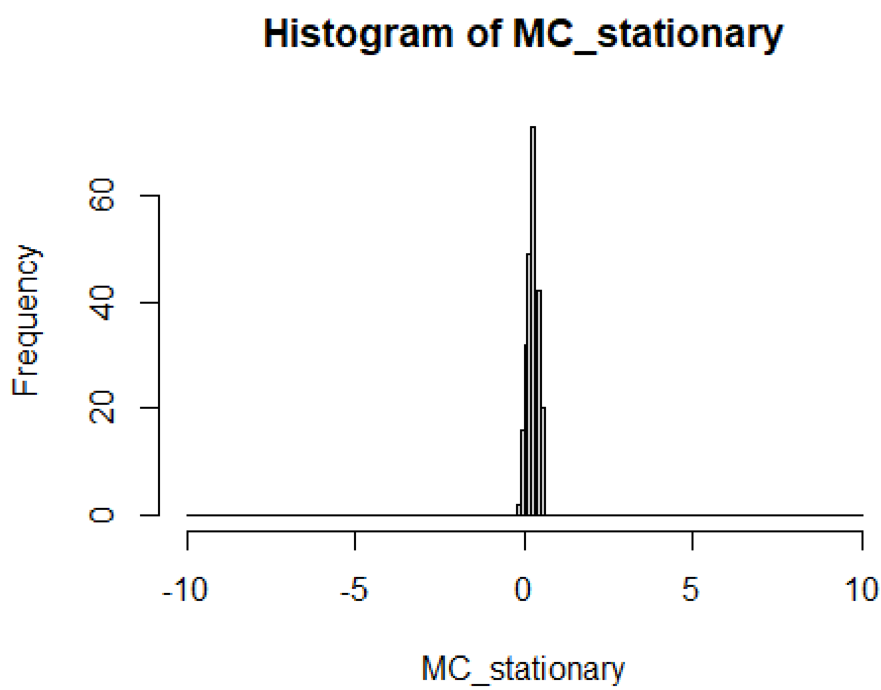

```
## [1] "MCD.csv"
```


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

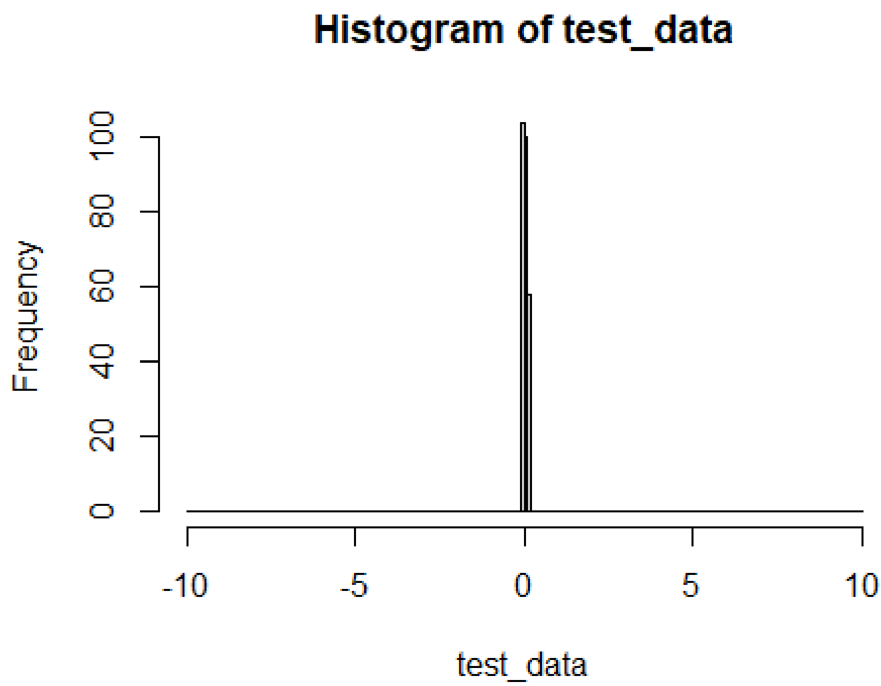

Histogram of MC_stationary
Frequency
0
10
20
30
40
-10
-5
0
5
10
MC_stationary

```
## [1] "MMM.csv"
```

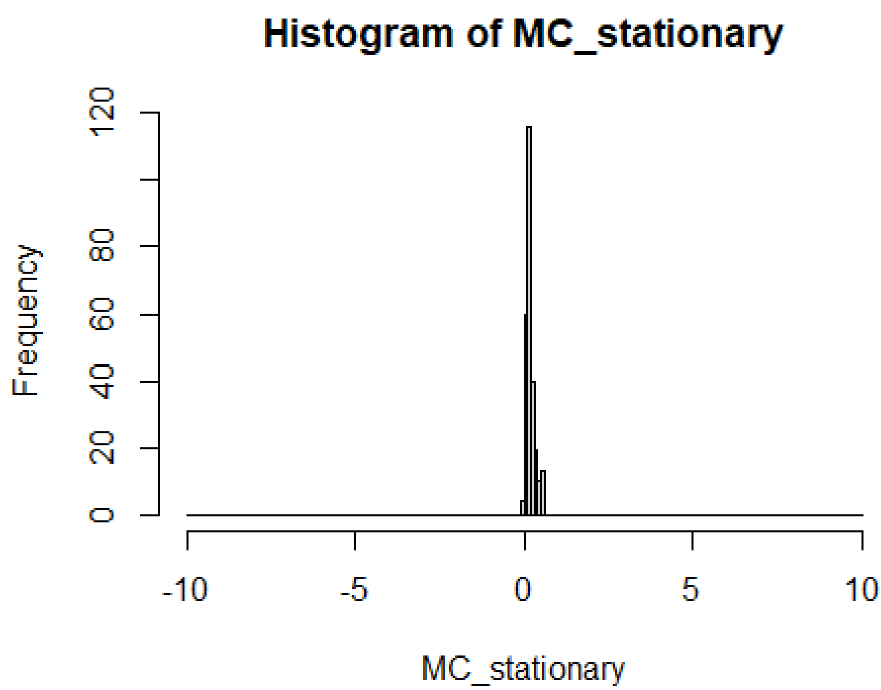

Histogram of test_data
Frequency
0
20
40
60
80
120
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
20
40
60
-10
-5
0
5
10
MC_stationary

```
## [1] "MRK.csv"
```

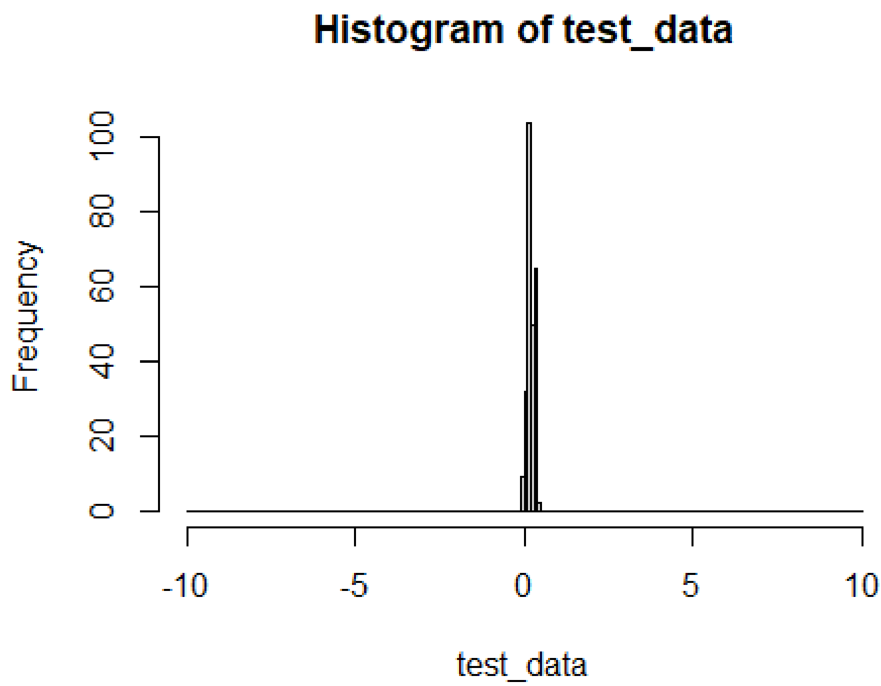


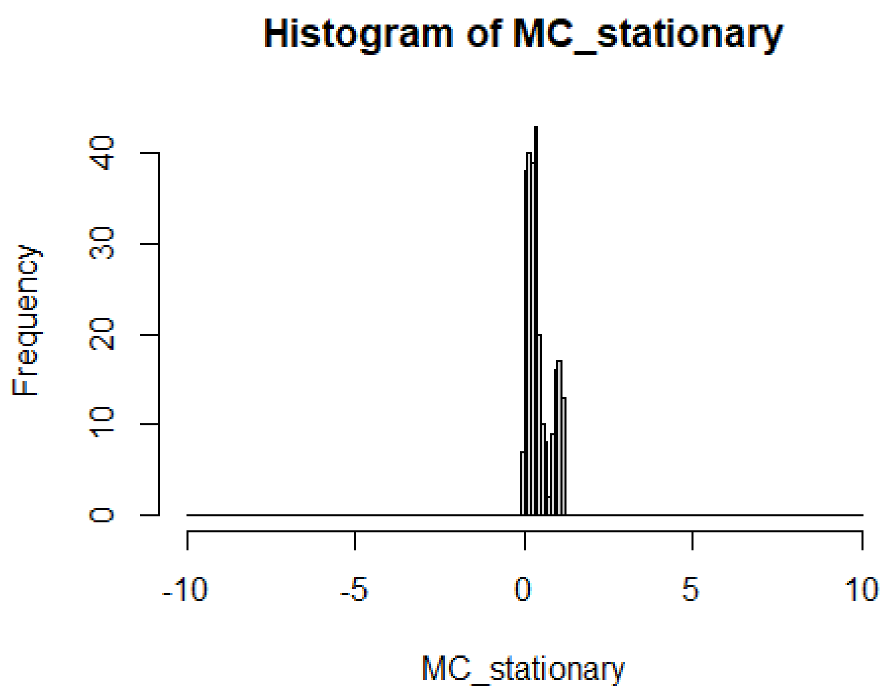
Histogram of test_data
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
test_data



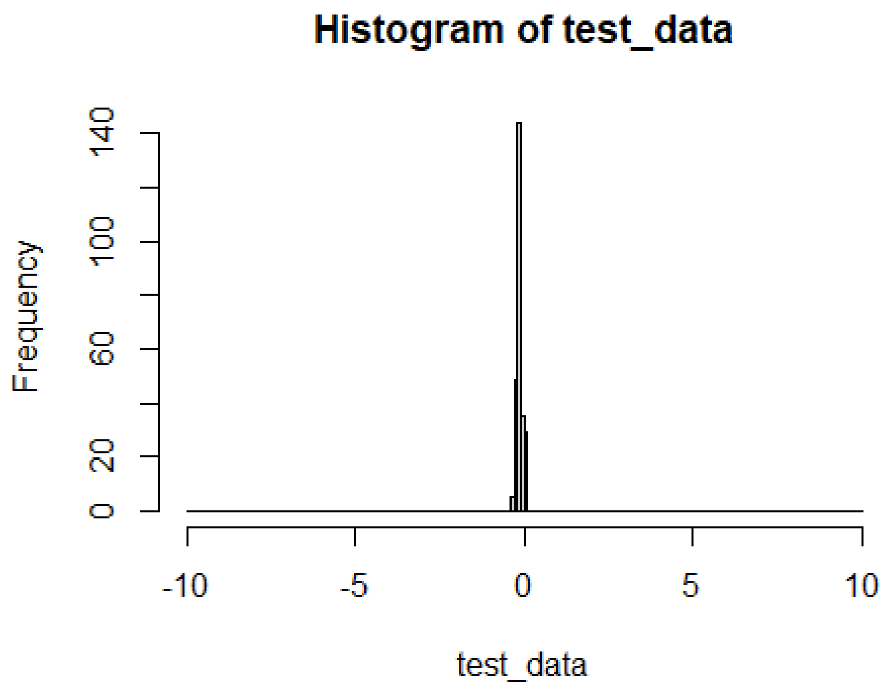
Histogram of MC_stationary
Frequency
0
20
40
60
80
120
-10
-5
0
5
10
MC_stationary

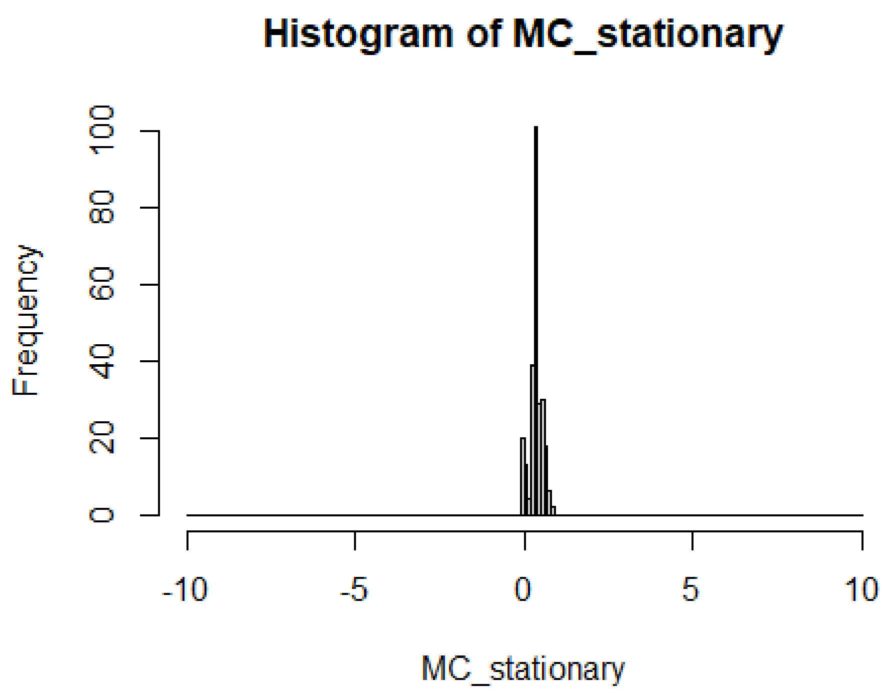

```
## [1] "MSFT.csv"
```


Histogram of test_data
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
test_data

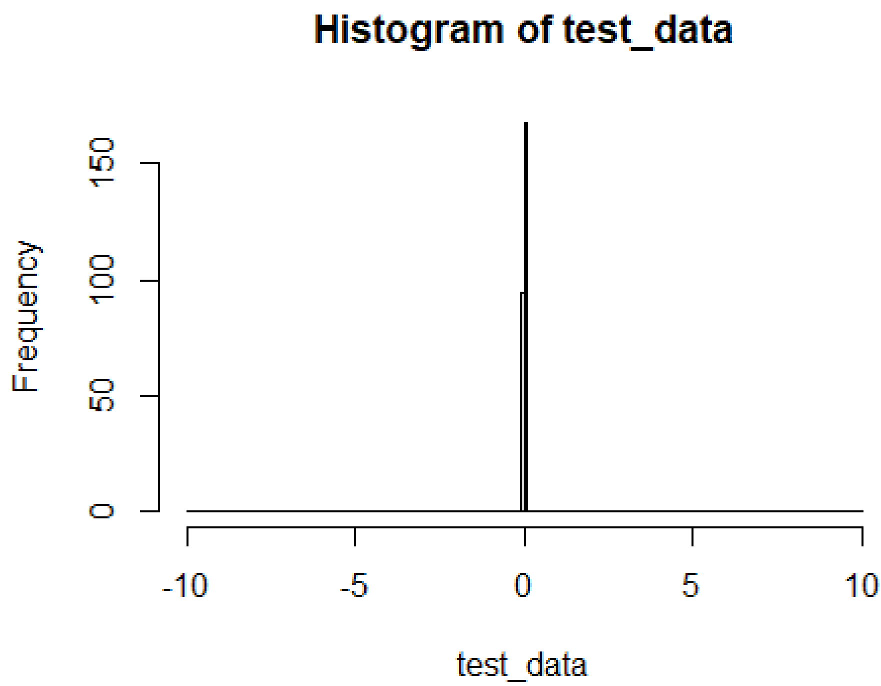

Histogram of MC_stationary
Frequency
0
10
20
30
40
-10
-5
0
5
10
MC_stationary

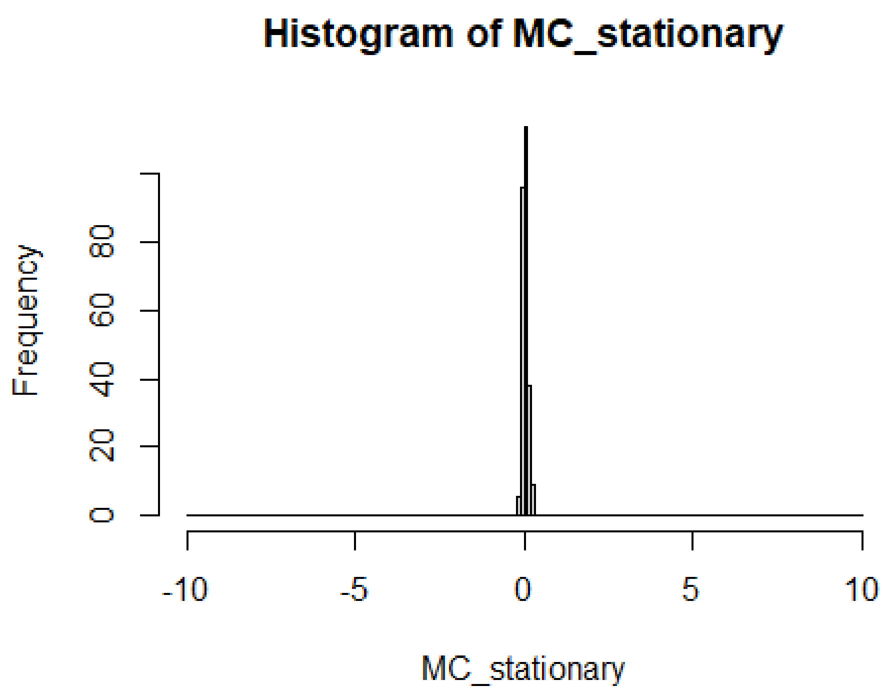
```
## [1] "NKE.csv"
```


Histogram of test_data
Frequency
0
20
60
100
140
-10
-5
0
5
10
test_data

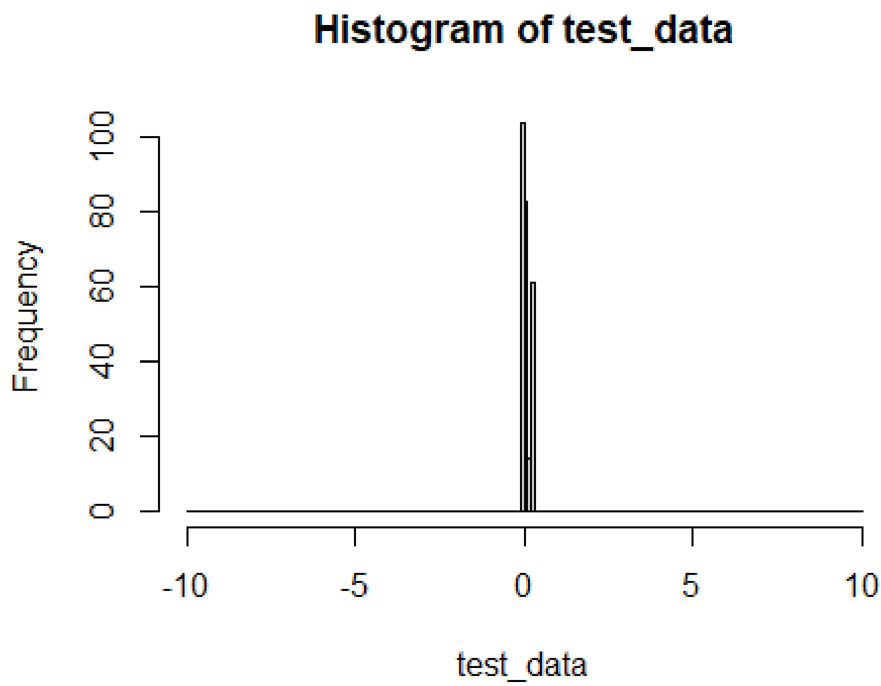

Histogram of MC_stationary
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
MC_stationary

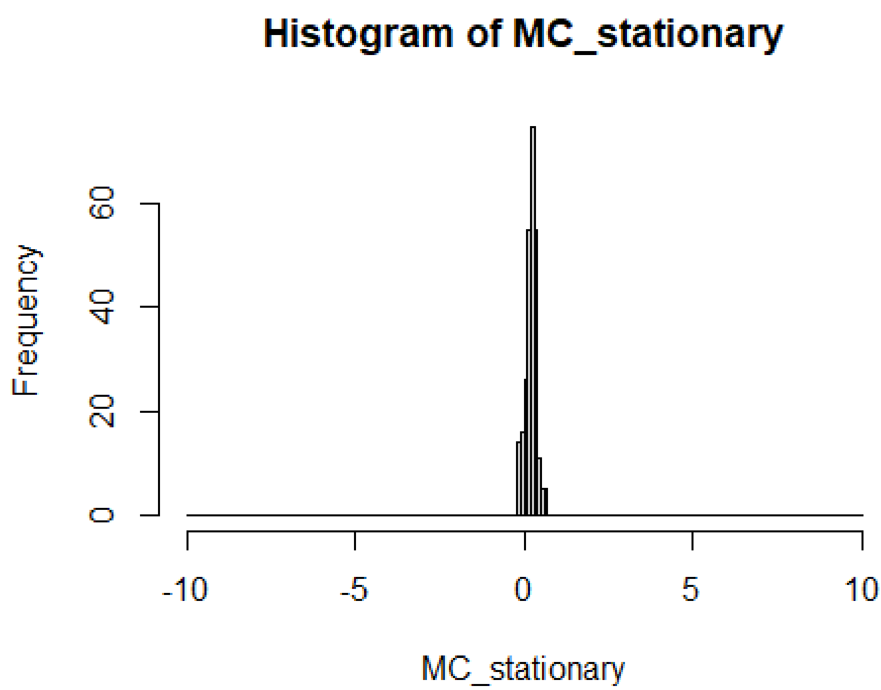
```
## [1] "PG.csv"
```


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

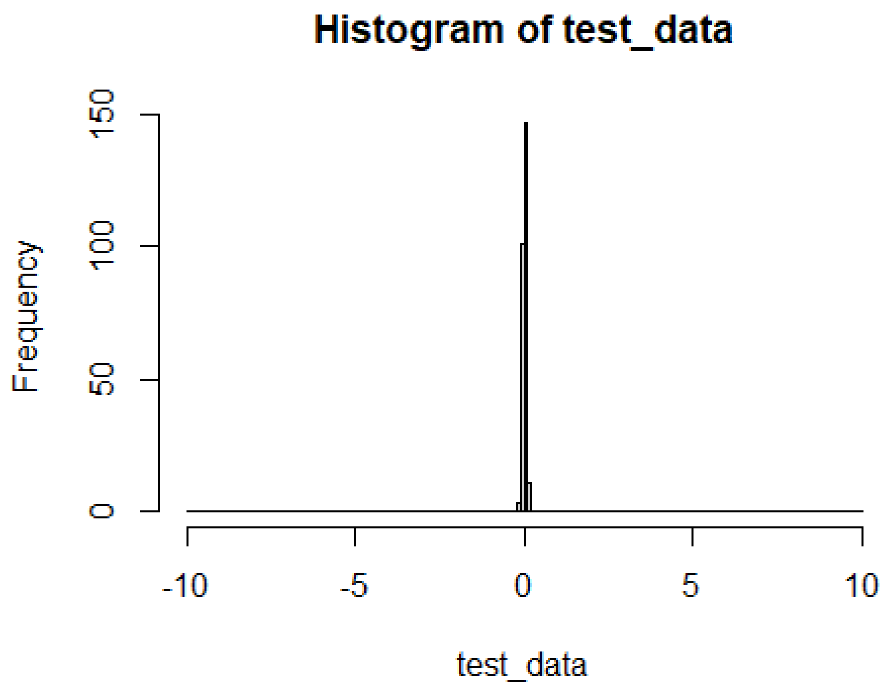

Histogram of MC_stationary
Frequency
0
20
40
60
80
-10
-5
0
5
10
MC_stationary

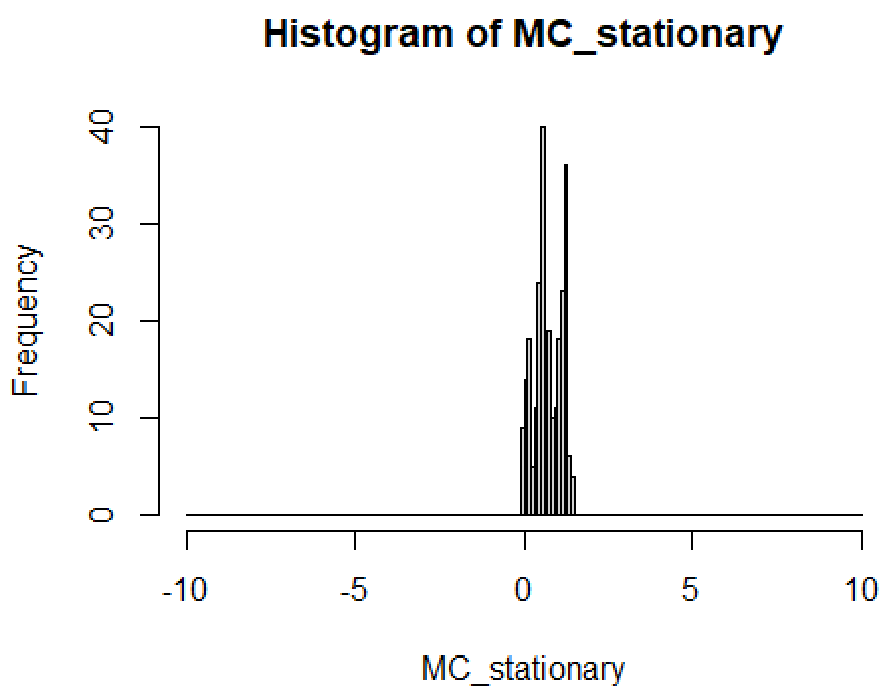
```
## [1] "TRV.csv"
```


Histogram of test_data
Frequency
0
20
40
60
80
100
-10
-5
0
5
10
test_data

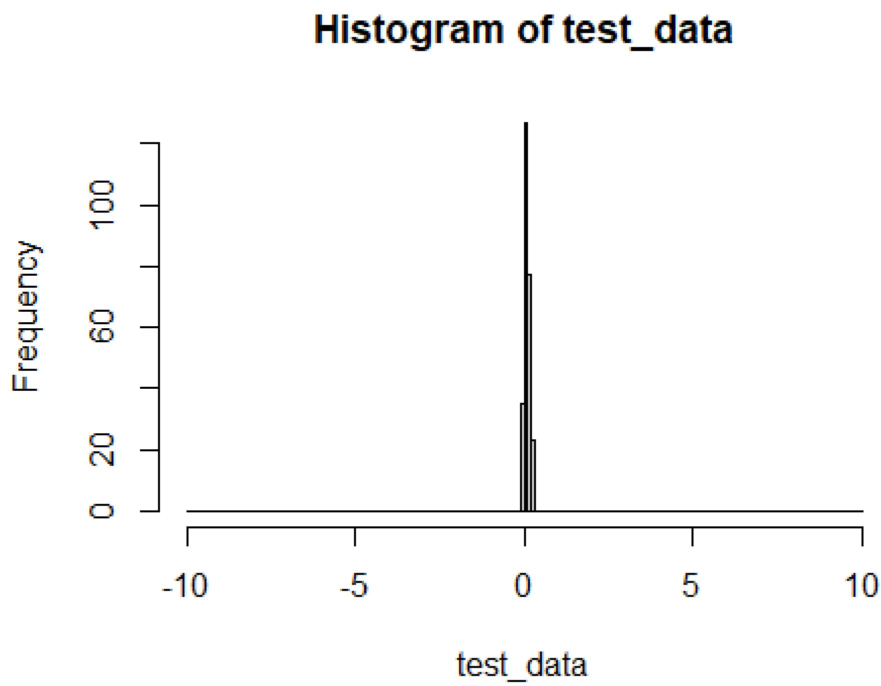

Histogram of MC_stationary
Frequency
0
20
40
60
-10
-5
0
5
10
MC_stationary

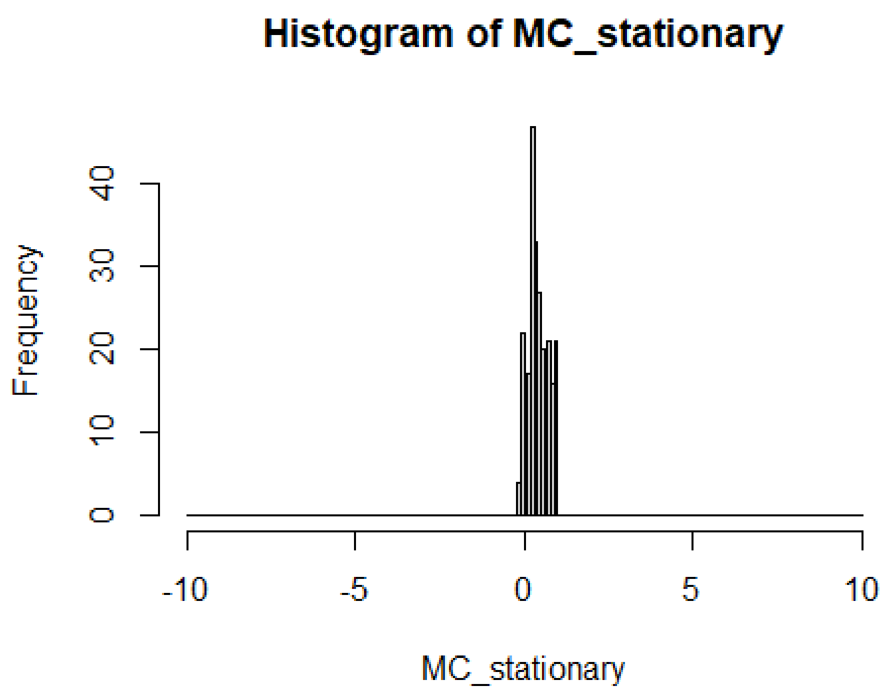
```
## [1] "UNH.csv"
```


Histogram of test_data
Frequency
0
50
100
150
-10
-5
0
5
10
test_data

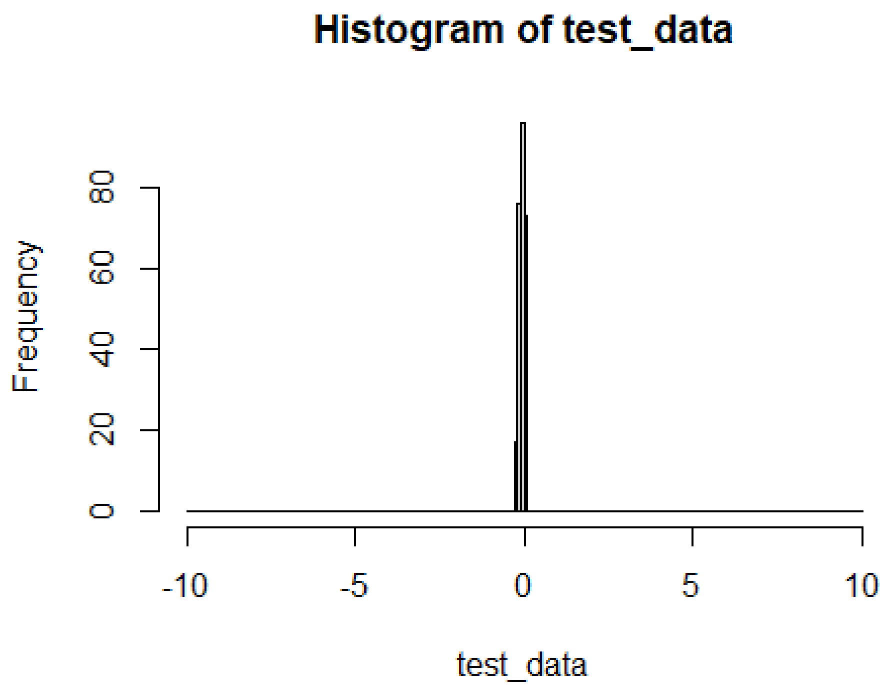

Histogram of MC_stationary
Frequency
0
10
20
30
40
-10
-5
0
5
10
MC_stationary

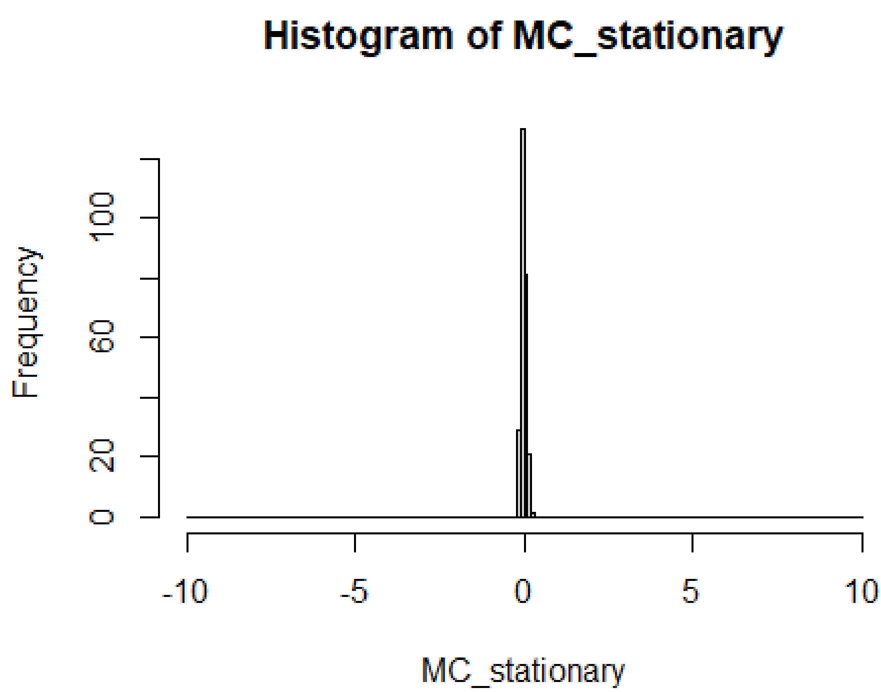
```
## [1] "V.csv"
```


Histogram of test_data
Frequency
0
20
60
100
-10
-5
0
5
10
test_data

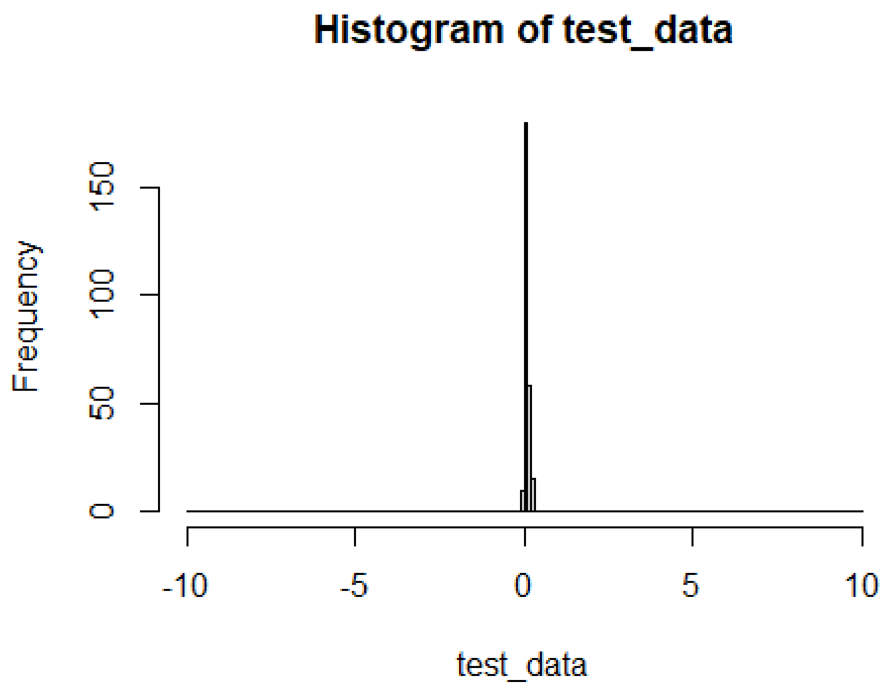

Histogram of MC_stationary
Frequency
0
10
20
30
40
-10
-5
0
5
10
MC_stationary

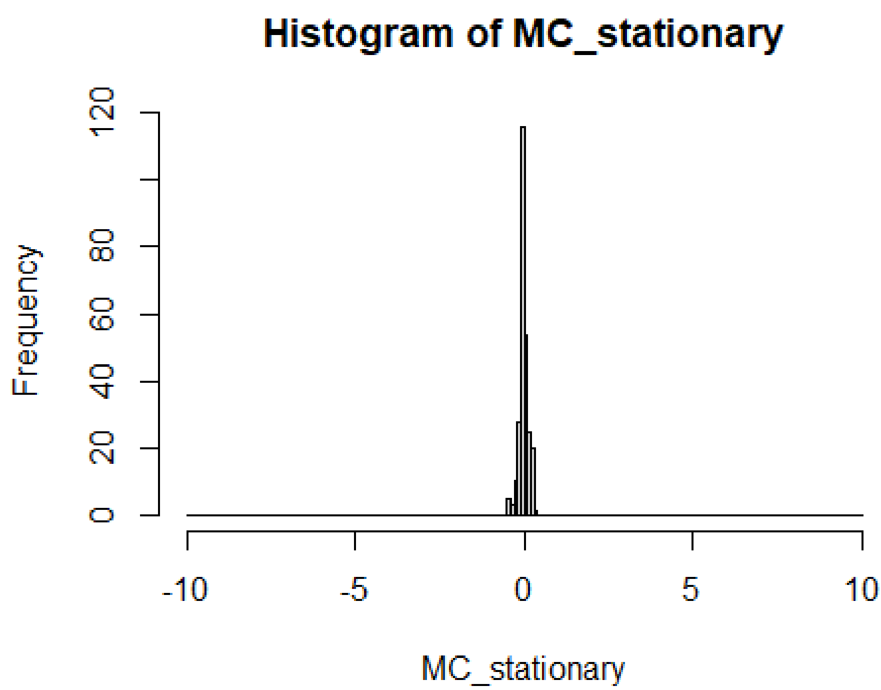
```
## [1] "VZ.csv"
```


Histogram of test_data
Frequency
0
20
40
60
80
-10
-5
0
5
10
test_data


Histogram of MC_stationary
Frequency
0
20
60
100
-10
-5
0
5
10
MC_stationary

```
## [1] "WMT.csv"
```






```
symbols = c()
for (f in files){
  symbols = c(symbols, substring(f, 1, nchar(f)-4))
}
```

```
deltas_sorted = sort(deltas_by_stock, index.return=TRUE, decreasing = TRUE)
barplot(deltas_sorted$x, names.arg=symbols[deltas_sorted$ix], las = 1,
horiz=TRUE, xlab = 'delta', xlim = c(0, 20))
```