Assignment 1

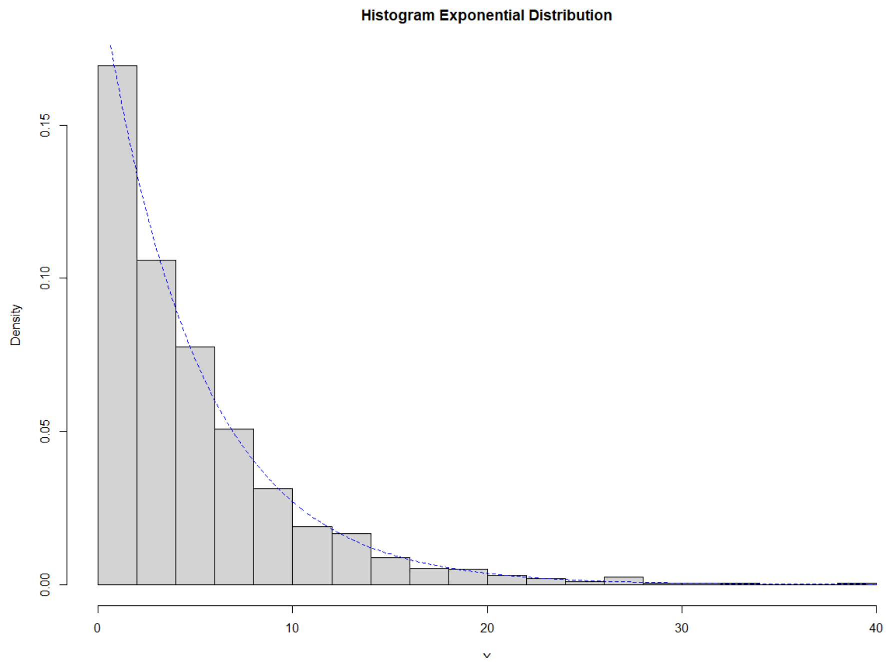
1) Problem 1.2 on page 63 of the book by Carmona

1.1)

```
# 1. generate sample of size N = 1024 with rate 0.2 and stored in vector X

X <- rexp(1024,0.2)
```

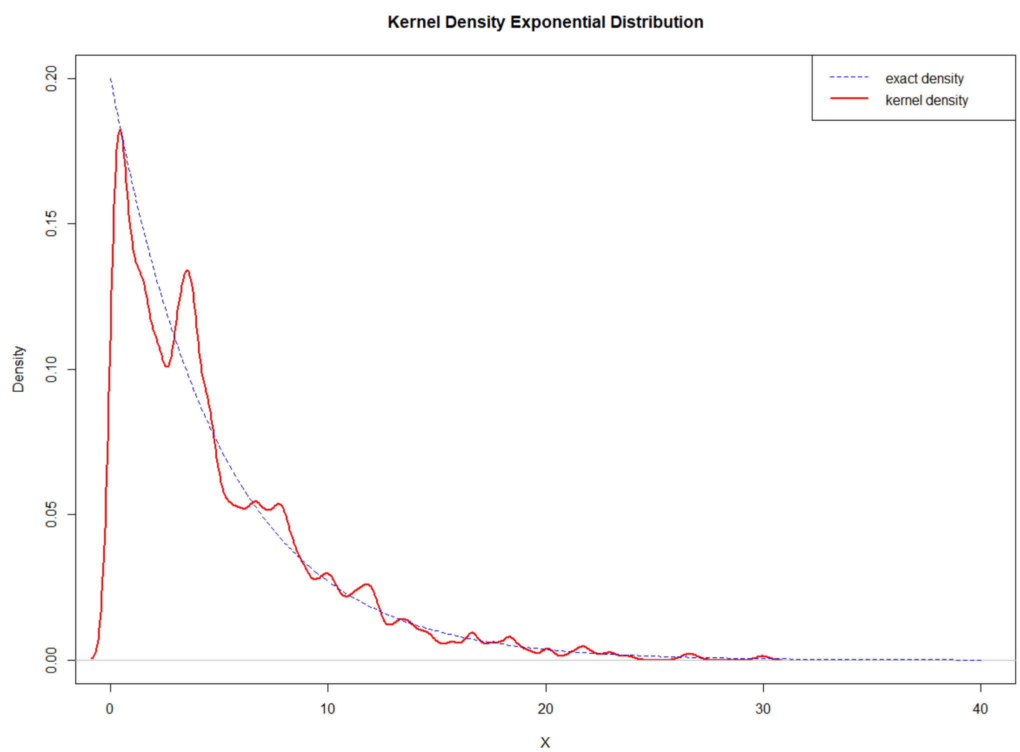
1.2)

```
# 2. plot histogram of X with 25 bins

hist(X, main="Histogram Exponential Distribution", prob=TRUE, breaks=25)

# 2. plot exact (theoretical) density

x <- seq(0, 40, 0.1)
lines(x, dexp(x, rate = 0.2), col = "blue", lwd=1, lty=2)

# 2. legend

legend("topright", legend="exact density", col=c("blue"), lty=2, lwd=1)
```



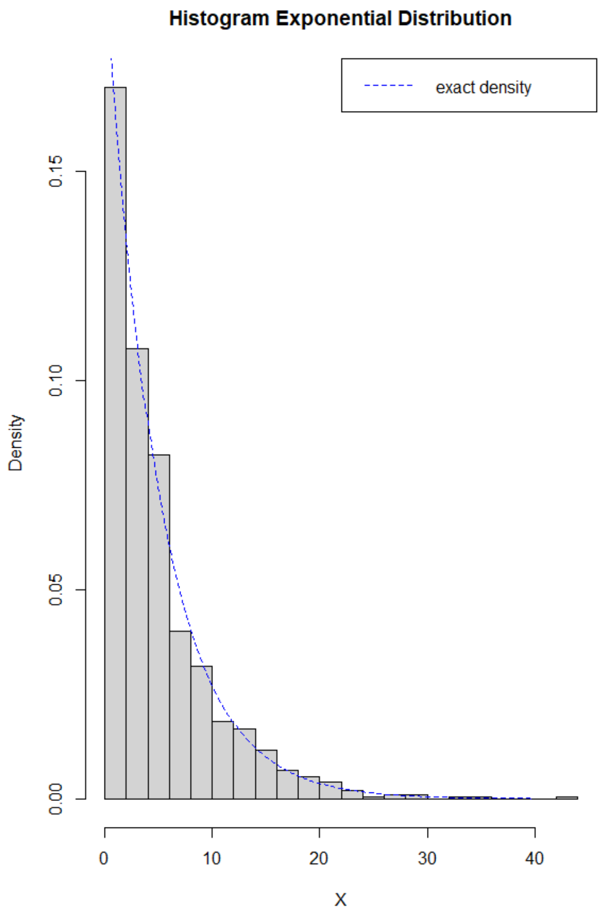
1.3)

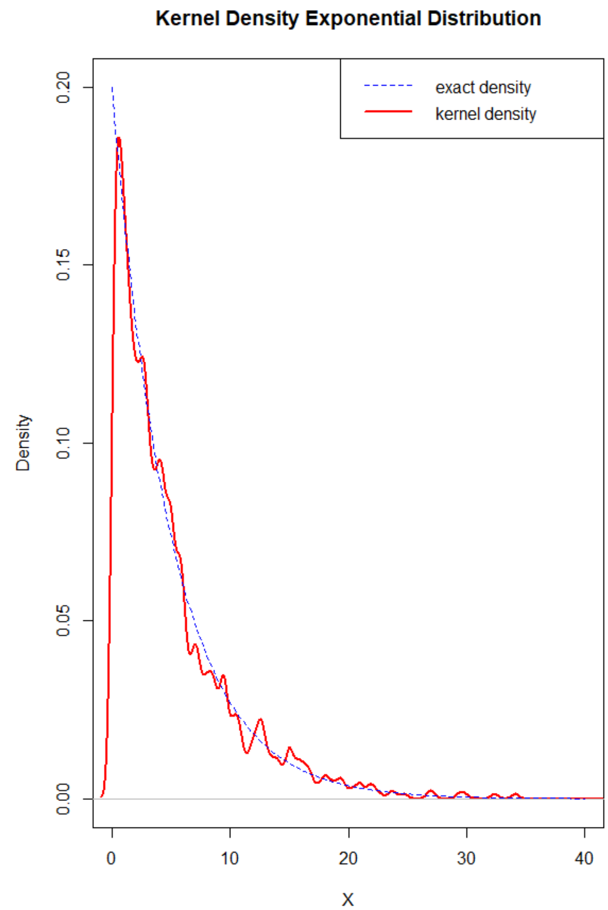
```
# 3. plot kernel density of X with 2.5 bandwidth
plot(density(X, kernel = "gaussian", bw=0.3), main="Kernel Density Exponential Distribution", xlab="X", col="red", lwd=2, xlim=c(0, 40), ylim=c(0, 0.2))
# 3. plot exact (theoretical) density
x <- seq(0, 40, 0.1)
lines(x, dexp(x, rate = 0.2), col = "blue", lwd=1, lty=2)
```



1.4)

The Histogram Exponential Density Plot is constructed using bins = 25 and the Kernel Density Exponential Distribution is constructed using the bandwidth = 0.3. The right chart looks more volatile and has more jagged edges and sharp drops, while the left chart - though also not smooth - seems to be a better fit. After trying a couple bandwidths for the kernel density estimation chart, it seems like a lower bandwidth is more suitable because it causes the chart to be more concentrated and taller at 0. A higher bandwidth will smooth out the chart but will also flatten out the curve, which means it will not capture the exponential distribution correctly. The histogram is a better fit because it looks closer than the kernel density estimation to the exact theoretical density.





2) Give your interpretation of each of the four Q-Q plots by filling in the blanks below.

Plot 1:

The right tail of Y is (**heavier than**/lighter than/similarly heavy-tailed to) normal distribution.

The left tail of Y is (heavier than/**lighter than**/similarly heavy-tailed to) normal distribution.

<u>Plot 2:</u>
The right tail of Z is (heavier than/lighter than/**similarly heavy-tailed to**) normal distribution.
The left tail of Z is (heavier than/**lighter than**/similarly heavy-tailed to) normal distribution.

<u>Plot 3:</u>
The right tail of T is (**heavier**/lighter) than the right tail of X.

<u>Plot 4:</u>
The right tail of T is (**heavier**/lighter) than the right tail of E.

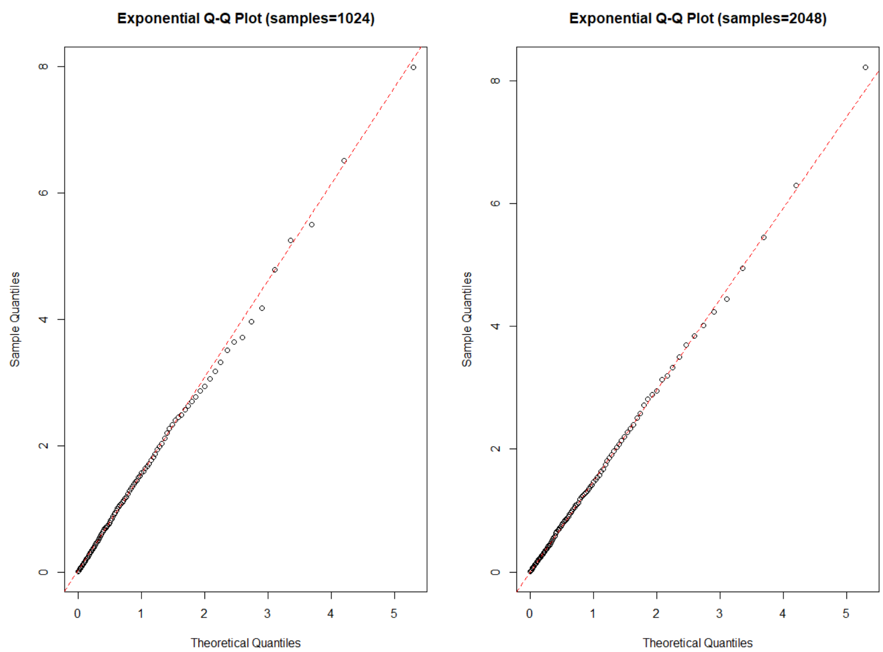
3) Problem 1.9, part 1 on page 66 of the book by Carmona.

3.1)

```
# 1. exponential distribution generator

myrexp <- function(N, LAMBDA) {
  u <- runif(N)
  x <- (-1/LAMBDA)*log(1-u)
  return(x)
}
```

3.2)

```
# 2. test exponential generator

N = 1024
mean = 1.5 # (1/mean = LAMBDA)

par(mfrow = c(1, 2))

# plot histogram of exponential distribution with N and mean

hist(myrexp(N, 1/mean), main="Histogram Home-Grown Exponential Distribution (N)", prob=TRUE, breaks=25, xlab="X", ylim=c(0, 0.8))

# plot histogram of exponential distribution with 2N and mean

hist(myrexp(2*N, 1/mean), main="Histogram Home-Grown Exponential Distribution (2N)", prob=TRUE, breaks=25, xlab="X", ylim=c(0, 0.8))

# plot generated exponential distribution on exponential Q-Q plot

for (samples in c(N, 2*N)) {
  Z <- myrexp(samples, 1/mean)     # random sample from exponential distribution
  p <- ppoints(100)    # 100 equally spaced points on (0,1), excluding endpoints
  q <- quantile(Z,p=p) # percentiles of the sample distribution

  plot(qexp(p), q, main=paste("Exponential Q-Q Plot (samples=",samples,")", sep=""), xlab="Theoretical Quantiles",ylab="Sample Quantiles")
  qqline(q, distribution=qexp,col="red", lty=2)
}
```

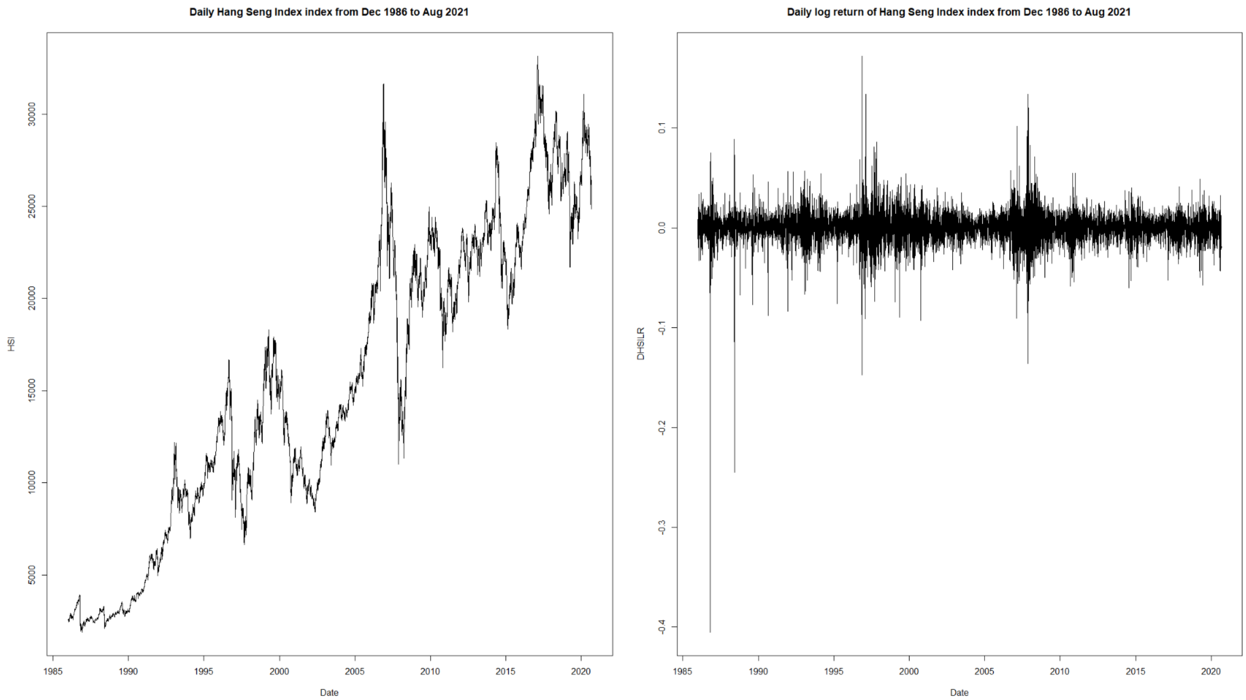


The performance for the simulation function myrexp is satisfactory because based on the exponential Q-Q plot above, it shows a straight line for both samples N and 2N, with 2N more concentrated. The above exponential Q-Q plot has the Sample Quantiles on the y-axis, which is generated from the myrexp function, and it also has the Theoretical Quantiles on the x-axis. A straight line from the Q-Q plot shows that the Sample Quantiles and the Theoretical Quantiles have generally the same distribution, which is exponential. This means that the simulation function performed as expected.

4) Conduct following analysis on daily close price of DHSI.

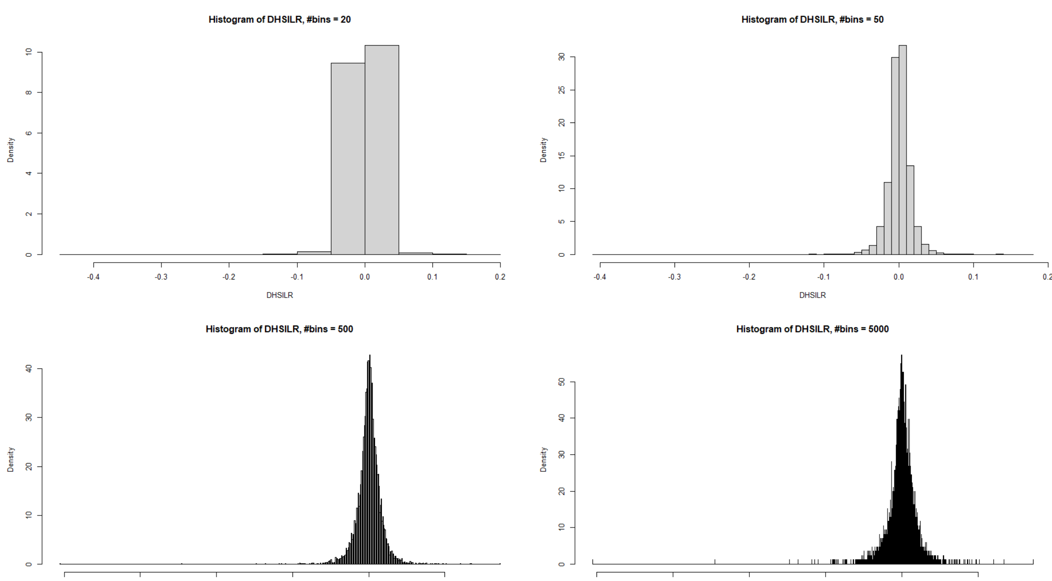
4.1)

```
# 0. Load DHSI data set
setwd("C:/Users/User/OneDrive/School/ISOM4530/Assignment/1")  # set directory
DHSI <- read.table("DHSI.csv",header = T, sep=",")
# 0. Extract variables
HSI <- (DHSI$Close)
HSI_time <- seq(from=1986,to=2020.67,length.out=length(HSI))
par(mfrow = c(1, 2))
# 1. HSI close price time series
plot(HSI_time,HSI,type="l",xlab="Date",main="Daily Hang Seng Index index from Dec 1986 to Aug 2021")
DHSILR <- diff(log(HSI))
# 1. HSI log return
plot(HSI_time[2:length(HSI)], DHSILR,type="l",xlab="Date",main="Daily log return of Hang Seng Index index from Dec 1986 to Aug 2021")
```



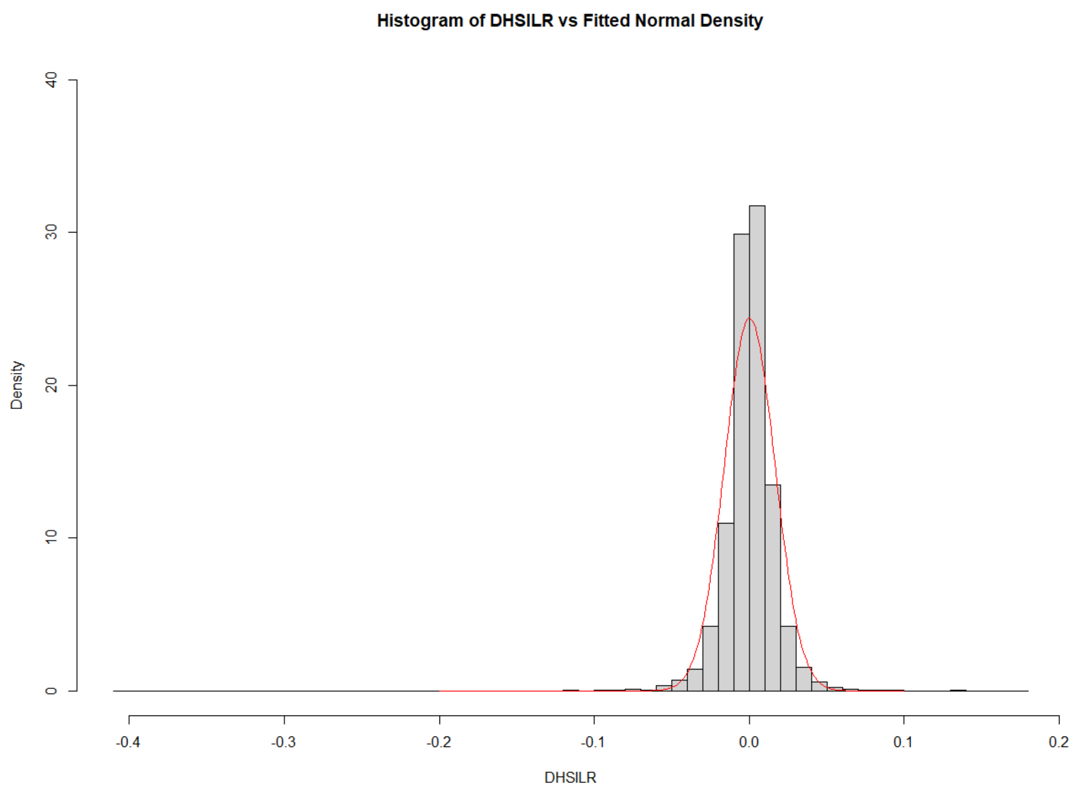
4.2)

```r
# 2. HSI log return histogram with varying bins

par(mfrow = c(2, 2))

for (bins in c(20,50,500,5000)) {
  hist(DHSILR, main=paste("Histogram of DHSILR, #bins =",bins), prob=TRUE, breaks=bins)
}
```



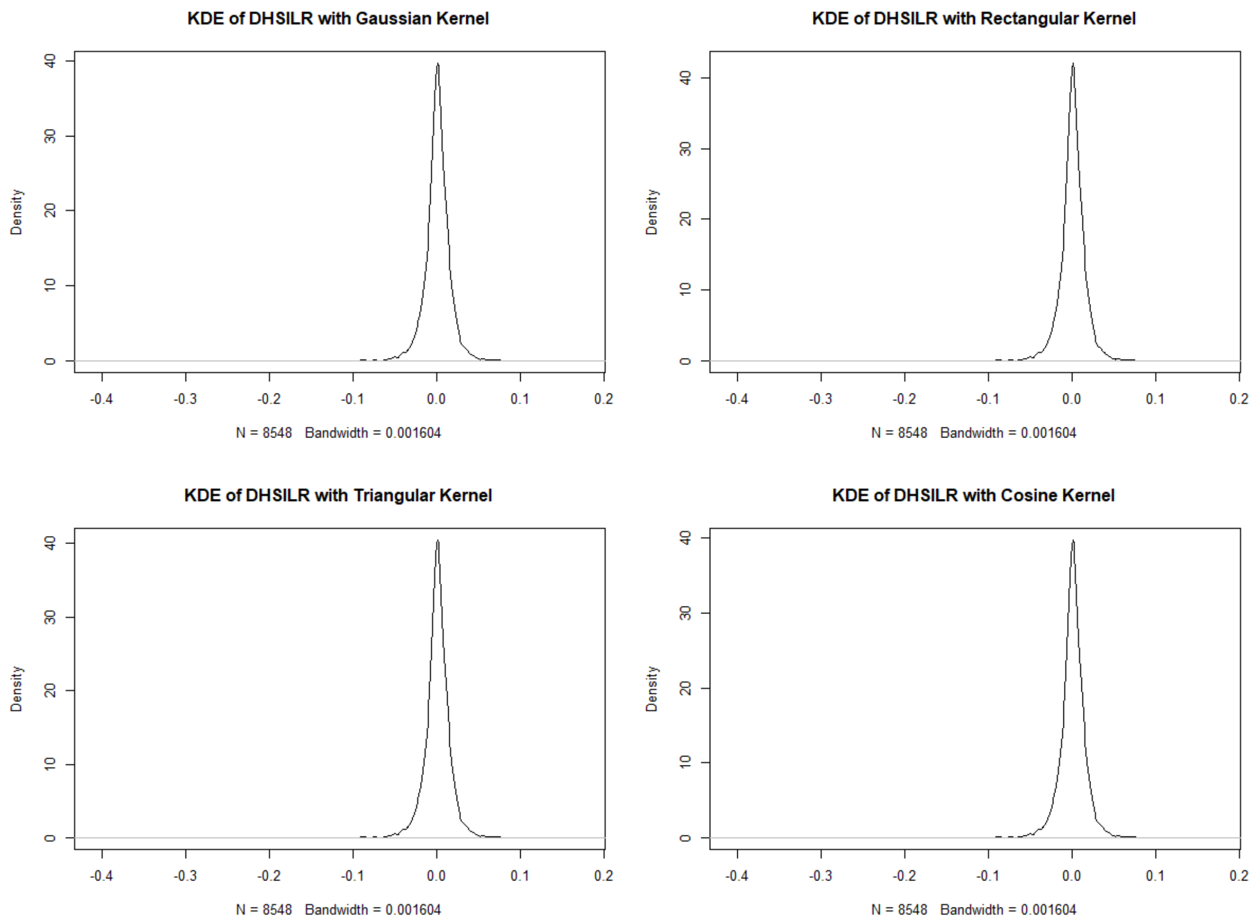
4.3)

```
# 3. HSI log return histogram with bin = 50 and superimposed normal density curve

par(mfrow = c(1, 1))

hist(DHSILR, main="Histogram of DHSILR vs Fitted Normal Density", prob=TRUE, breaks=50, ylim=c(0, 40))

mu_DHSILR <- mean(DHSILR)
sd_DHSILR <- sd(DHSILR)

x<-seq(-0.2,0.1,by=0.001)
y<-dnorm(x,mean=mu_DHSILR,sd = sd_DHSILR)
points(x,y,type="l",col="red")
```



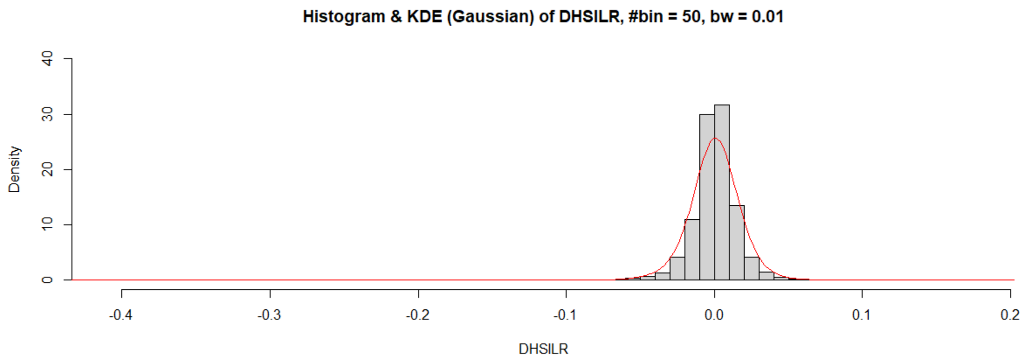
4.4)

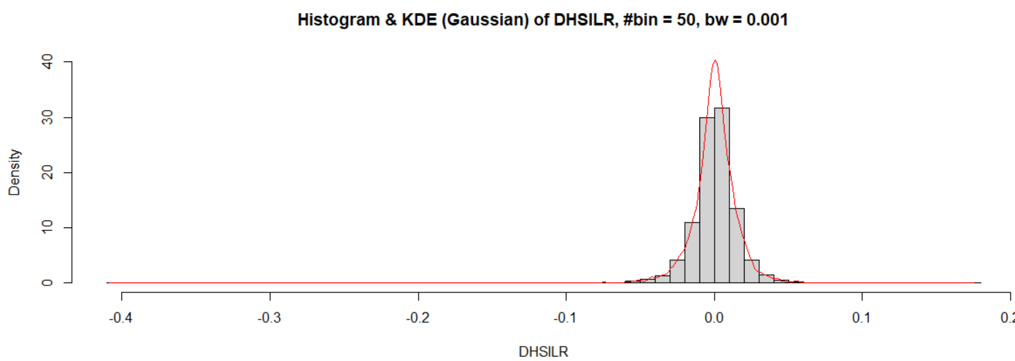
```
# 4. kernel density estimation of HSI with varying kernels

library(stringr)

par(mfrow = c(2, 2))

for (kernel_type in c("gaussian", "rectangular", "triangular", "cosine")) {
  plot(density(DHSILR, kernel=kernel_type), lwd=1, main=paste("KDE of DHSILR with",str_to_title(kernel_type),"Kernel"))
}
```



4.5)

```
# 5. kernel density estimation of HSI with gaussian kernel

par(mfrow=c(2,1))

for (bandwidth in c(0.01,0.001)) {
  hist(DHSILR,breaks=50,  freq = F, main=paste("Histogram & KDE (Gaussian) of DHSILR, #bin = 50, bw =",bandwidth), ylim=c(0,40))
  points(density(DHSILR, kernel = "gaussian", bw = bandwidth) ,type="l",col="red")
}
```





4.6)

```
# 6. empirical VaR computation / VaR under normal assumption

q <- 0.01

# 6. empirical

VaR_emp <- -quantile(DHSILR,q)
VaR_emp

# 6. normal
mu_DHSILR <- mean(DHSILR)
sd_DHSILR <- sd(DHSILR)

VaR_normal <- - qnorm(q,mu_DHSILR, sd_DHSILR)
VaR_normal
```

<u>Empirical VaR:</u> 0.04327484 or 4.33%
<u>VaR under Normal Assumption:</u> 0.03777452 or 3.78%

4.7)

```r
# 7. empirical expected shortfall and expect shortfall under normal assumption

q<-0.01

# 7 . empirical

VaR_emp <- - quantile(DHSILR,q)
ES_emp <- mean(- DHSILR[- DHSILR > VaR_emp])

# 7. normal

mu_DHSILR <- mean(DHSILR)
sd_DHSILR <- sd(DHSILR)
VaR_normal <-  - qnorm(q,mu_DHSILR, sd_DHSILR)

N<-100000
X<-rnorm(N,mu_DHSILR,sd_DHSILR)
ES_normal <- mean( - X[- X > VaR_normal])

# 7. results

c(ES_emp, ES_normal)
```

<u>Empirical Expected Shortfall:</u> 0.06837496 or 6.84%
<u>Expected Shortfall under Normal Assumption:</u> 0.04321267 or 4.32%