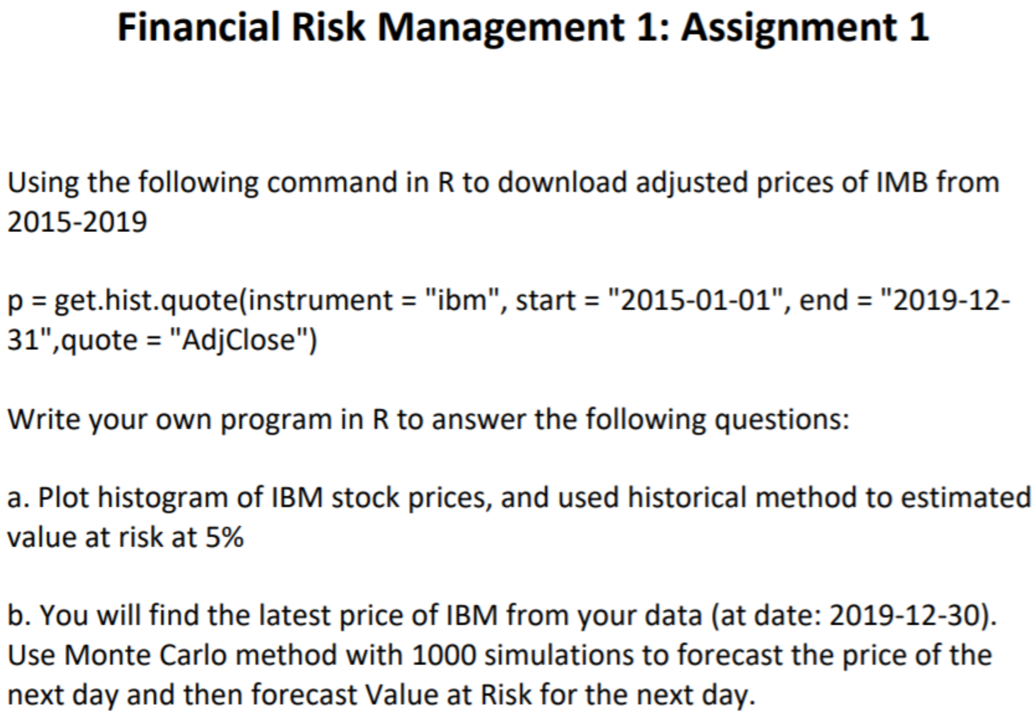
FRM 1 - ASSIGNMENT 1

NGUYEN NGOC PHUNG

5/7/2021



Financial Risk Management 1: Assignment 1

Using the following command in R to download adjusted prices of IMB from 2015-2019

p = get.hist.quote(instrument = "ibm", start = "2015-01-01", end = "2019-12-31",quote = "AdjClose")

Write your own program in R to answer the following questions:

a. Plot histogram of IBM stock prices, and used historical method to estimated value at risk at 5%

b. You will find the latest price of IBM from your data (at date: 2019-12-30). Use Monte Carlo method with 1000 simulations to forecast the price of the next day and then forecast Value at Risk for the next day.

## GET THE IBM STOCK PRICE FROM 2015-2019

```
# IMPORT THE LIBRARY
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library('quantmod')
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library('tidyquant')
```

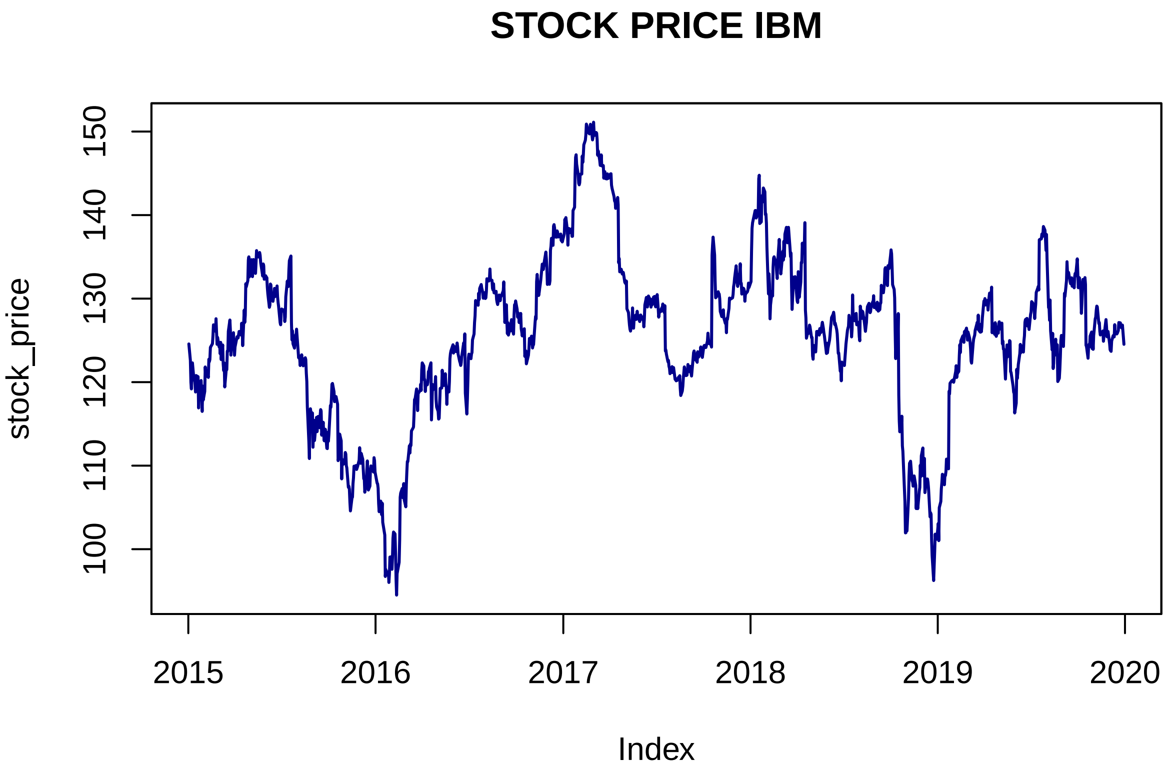
```
## Warning: package 'tidyquant' was built under R version 4.0.3

## Loading required package: lubridate

## Warning: package 'lubridate' was built under R version 4.0.3

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Loading required package: PerformanceAnalytics

## Warning: package 'PerformanceAnalytics' was built under R version 4.0.3

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend

## == Need to Learn tidyquant? ==================================================
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimiza
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

```r
# GET THE STOCK PRICE DATA
stock_price= get.hist.quote(instrument = "ibm", start = "2015-01-01", end = "2019-12- 31",quote = "AdjC
```

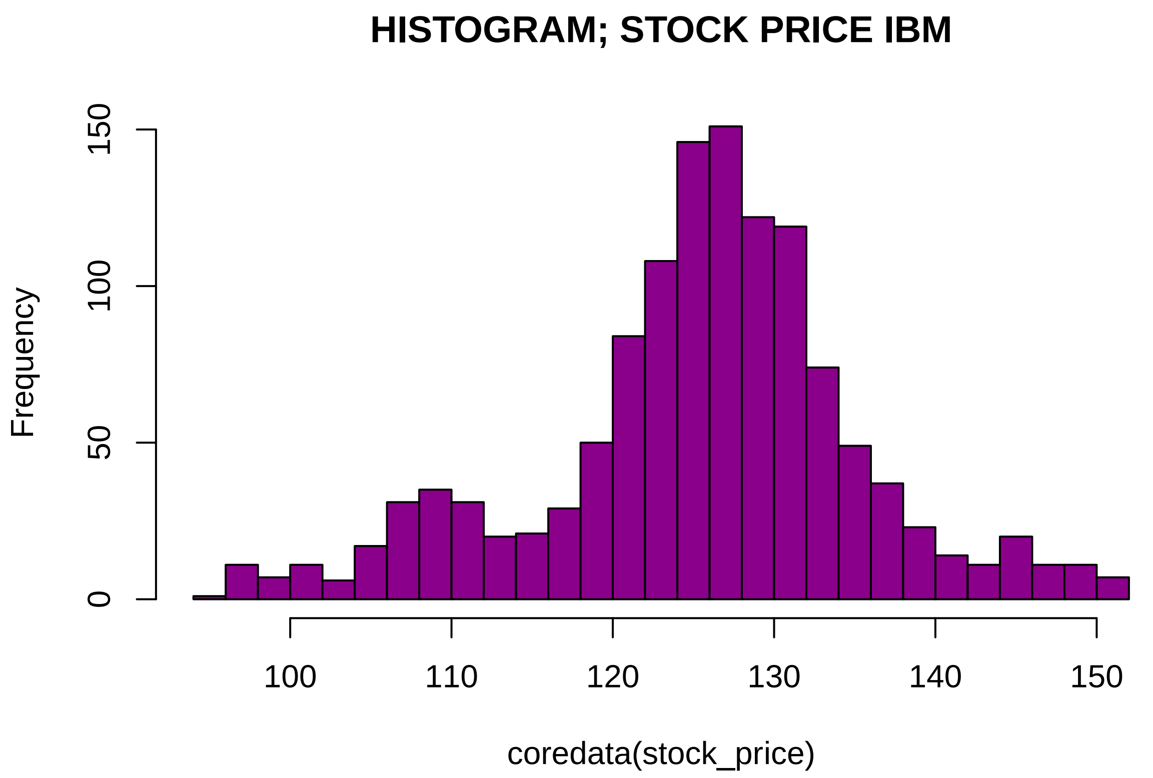
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## time series starts 2015-01-02
## time series ends   2019-12-30
```

PLOTTING AND HISTOGRAM OF STOCK PRICE St

```
# Plotting of stock price
plot(stock_price,col="darkblue",lwd=1.5, main= 'STOCK PRICE IBM')
```



```
# Histogram of stock price
hist(coredata(stock_price), col='darkmagenta', breaks=40, main='HISTOGRAM; STOCK PRICE IBM')
```

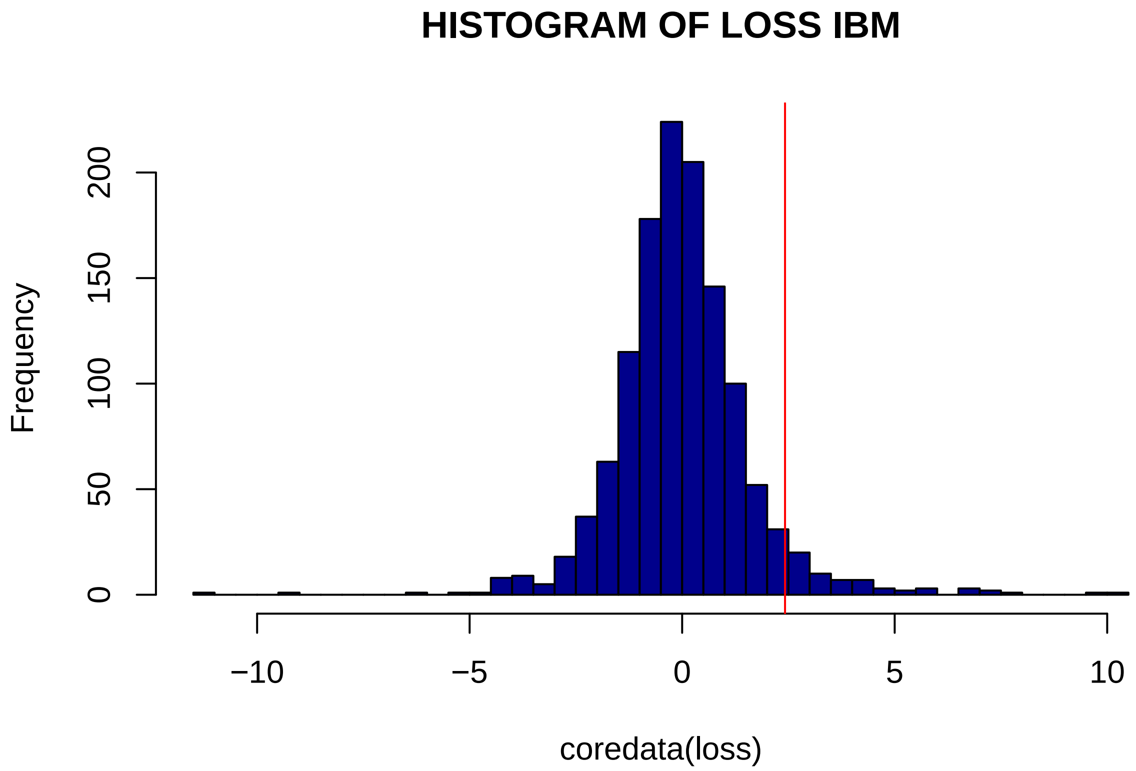


HISTORICAL METHOD : VALUE AT RISK AT 5% (95% CONFIDENCE LEVEL)

```r
# First, we calculate the loss of each day by taking S(t-1)-S(t)
loss <- -diff(stock_price,differences = 1)
n<- length(loss)

# Second, we sort the loss in ascending order @-@
sorted_loss<- sort(coredata(loss))


# Third, we calculate the VaR 95% by taking the 1193rd value of sorted_loss (n*0.95= 1193.2)
# So VaR 95% of IBM is:
VaR_95= sorted_loss[1193]
VaR_95
```

```
## [1] 2.420417
```

```r
# Plotting
hist(coredata(loss), col='darkblue', breaks=40, main='HISTOGRAM OF LOSS IBM')
abline(v=VaR_95,col='red')
```



MONTE-CARLO METHOD:

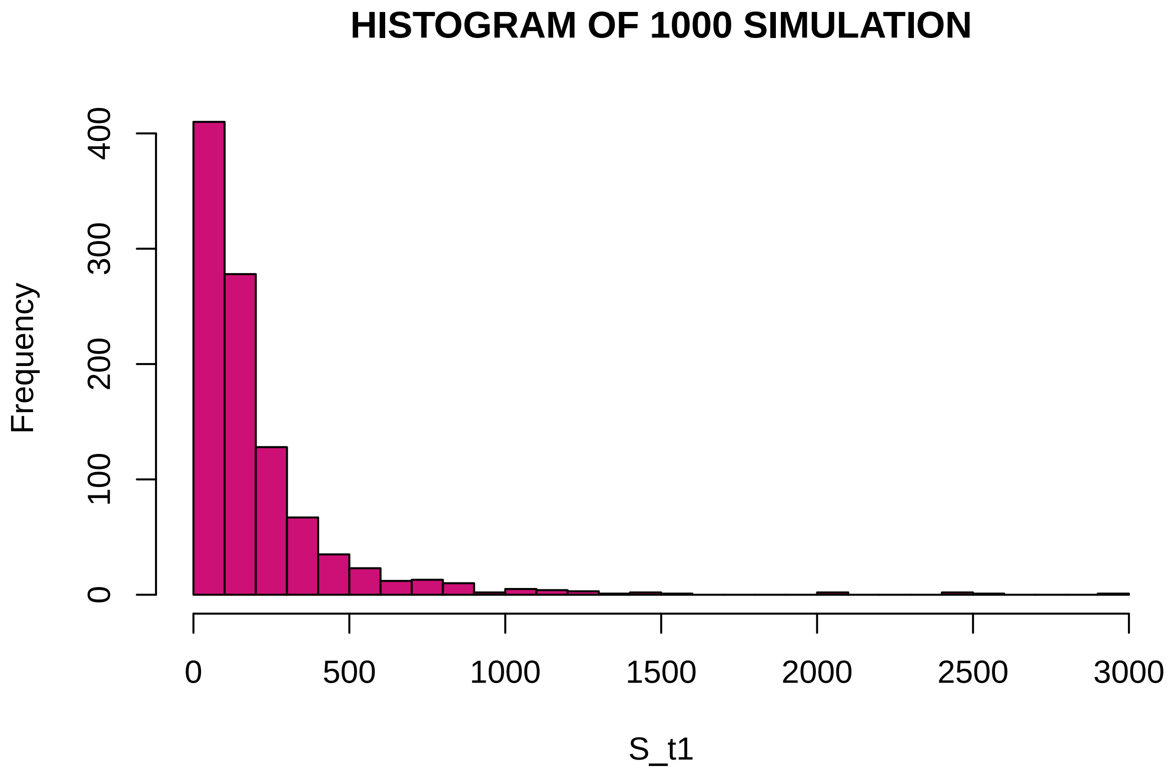
```
# Loss in the last day (30-12-2019) is:
S_t<- coredata(stock_price)[1256]

# We simulate 1000 Z_t from N(0,1) @-@
Z_t<-rnorm(1000)
Z_t[1:10]
```

```
##  [1]  0.13210981  0.58559528 -0.21838621 -0.65035644  1.68336425 -2.11417898
##  [7] -1.28902619 -0.87962979 -2.06145227  0.01333112
```

```
# Calculate the mean and standard deviation:
u<-mean(diff(log(stock_price),1)[-1,])
sigma<-sd(diff(log(stock_price),1)[-1,])
c(u, sigma)
```

```
## [1] 1.233769e-05 1.298964e-02
```

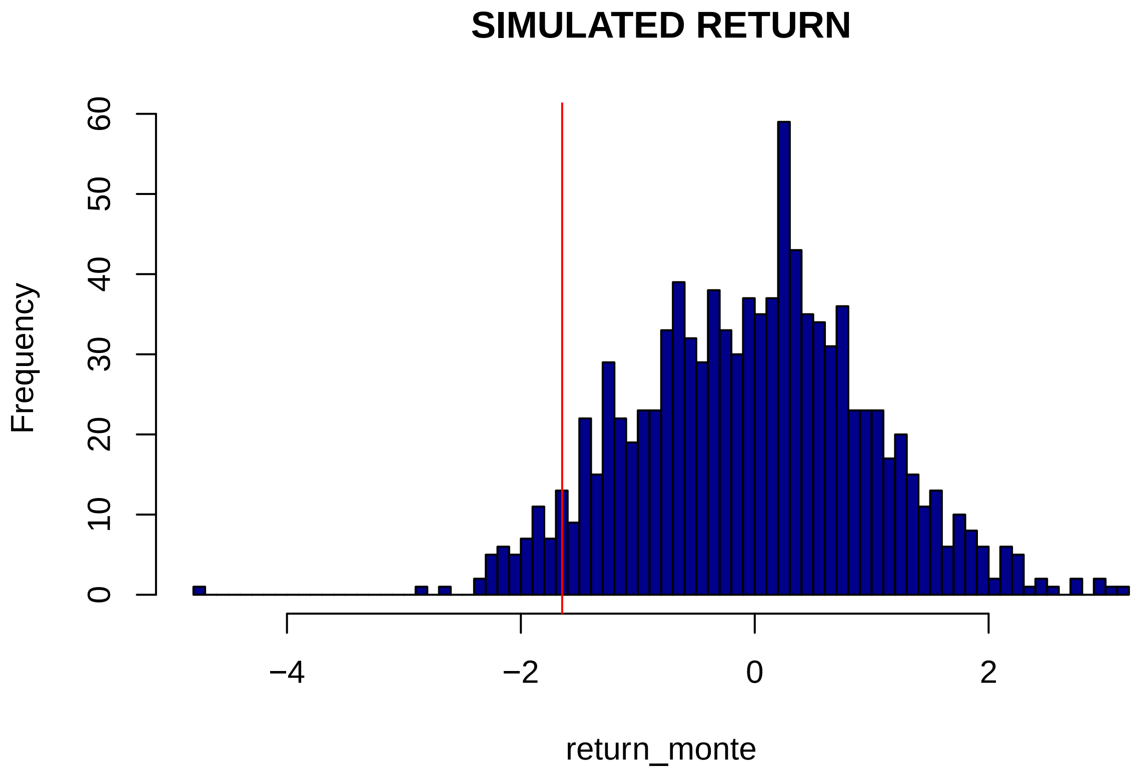
```
# So 1000 scenarios of S(t+1) is:
S_t1<-S_t*exp((u-(sigma^2)/2) + Z_t)

# Plotting
hist(S_t1, col='deeppink3', breaks=40, main='HISTOGRAM OF 1000 SIMULATION')
```



```
# Now, we calculate VaR by taking 5% quantile from the return (estimated) @-@
return_monte <- log(S_t1/S_t)
VaR_95_monte <- quantile(return_monte,prob=0.05)
VaR_95_monte
```

```
##        5%
## -1.646906
```

```
# Histogram of the simulated return @-@
hist(return_monte, breaks = 70, col='darkblue', main='SIMULATED RETURN')
abline(v=VaR_95_monte,col='red')
```


SIMULATED RETURN
Frequency
0
10
20
30
40
50
60
−4
−2
0
2
return_monte