QF301. Lecture 5 In-Class Assignment.

2021-09-27

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

By filling out the following fields, you are signing this pledge. No assignment will get credit without being pledged.

Name: Siddharth Iyer

CWID: 10447455

Date: 9/24/2021

Question 1 (100pt)

Question 1.1

```
CWID = 10447455 #Place here your Campus wide ID number, this will personalize
#your results, but still maintain the reproduceable nature of using seeds.
#If you ever need to reset the seed in this assignment, use this as your seed
#Papers that use -1 as this CWID variable will earn 0's so make sure you change
#this value before you submit your work.
personal = CWID %% 10000
set.seed(personal)
```

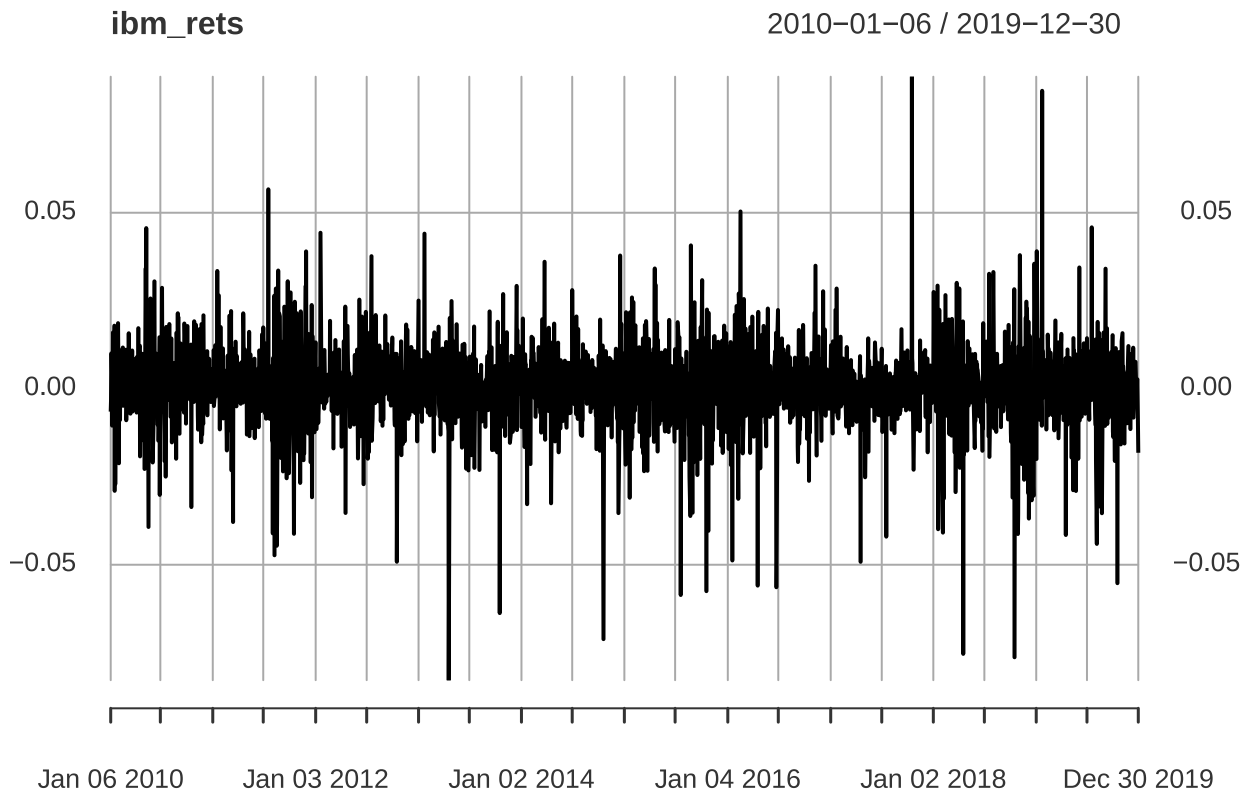
Obtain the daily log returns for IBM from January 1, 2010 until December 31, 2019. Plot these returns.

Solution:

```
library(quantmod)
getSymbols(c("IBM"), from="2010-01-01", to = "2019-12-31")
```

```
## [1] "IBM"
```

```
IBM = IBM$IBM.Adjusted
ibm_rets = dailyReturn(IBM)[2:nrow(IBM)]
ibm_rets = ibm_rets[-1]
plot(ibm_rets)
```



Question 1.2

Create a data frame consisting of 1 day lagged returns (and current returns).

Solution:

```
r1 = as.numeric(lag(ibm_rets,k=1))[-1]
r0 = as.numeric(ibm_rets)[-1]
df = data.frame(r0, r1)
```

Question 1.3

Split your data into a training set and a testing set (75% in training set). Create an AR(1) model for your stock returns. Provide the test mean squared error.

Solution:

```
train=sample(length(r0),length(r0)*3/4,replace=FALSE)    # rows for training set

ar1 = glm(r0~.,data=df,subset=train)
```

```r
pred=predict(ar1,df[-train,])   # test set prediction
MSE = mean((pred-df$r0[-train])^2)   #Test MSE
cat("Test MSE (AR): ", MSE, "\n")
```

```
## Test MSE (AR):  0.0001485524
```

```r
#Compare with naive/constant predictor:
MSE0 = mean((mean(df$r0[train])-df$r0[-train])^2) #Test MSE
cat("Test MSE (Base): ", MSE0, "\n")
```

```
## Test MSE (Base):  0.0001477311
```

Question 1.4

Evaluate if your AR(1) model is weakly stationary or unit-root nonstationary.

Solution:

```r
#Check coefficients/check unit-root nonstationary
summary(ar1)$coefficients
```

```
##                  Estimate   Std. Error    t value  Pr(>|t|)
## (Intercept)  0.0001430957 0.0002855181  0.5011791 0.6163037
## r1          -0.0297146591 0.0230757903 -1.2876984 0.1980093
```

```r
b0 = summary(ar1)$coefficients[1,1]
b1 = summary(ar1)$coefficients[2,1]
b1.se = summary(ar1)$coefficients[2,2]
abs(b1) < 1 #If true then suspect weakly stationary
```

```
## [1] TRUE
```

```r
#Test for unit-root nonstationary
t = (b1 - 1)/b1.se
t
```

```
## [1] -44.62316
```

```r
pnorm(t) #Large enough data, treat as normal
```

```
## [1] 0
```

```r
#If small enough then reject the null hypothesis
```

This time series is very likely weakly stationary because it has a very low p-value.

Question 1.5

If your model is weakly stationary, provide the (long-run) average returns and (first 5) autocovariances for your model. How do these compare with the empirical values? If your model is (unit-root) nonstationary, please interpret your model. Give as much detail as possible.

Solution:

```
cat("Annualized Average Log-returns: ", mean(ibm_rets)*252, "\n")
```

```
## Annualized Average Log-returns:  0.05024016
```

```
#Forecasting:
N = length(df$r0)
sig = sqrt(MSE)
hat_r = matrix(nrow=length(df$r0)-5,ncol=5)
e = matrix(nrow=N-5,ncol=5)
var_e = rep(NA, 5)

hat_r[,1] = b0 + b1*df$r0[1:(N-5)]
e[,1] = df$r0[2:(N-4)] - hat_r[,1]
var_e[1] = sig^2
for (l in seq(2,5)) {
  hat_r[,l] = b0 + b1*hat_r[,l-1]
  e[,l] = df$r0[(l+1):(N-5+l)]
  var_e[l] = sig^2 + b1^2*var_e[l-1] #Why is this the correct equation?
}

#Compare empirical and theoretical forecast standard deviation
cat("Theoretical Std: ", sqrt(var_e), "\n")
```

```
## Theoretical Std:  0.01218821 0.01219359 0.01219359 0.01219359 0.01219359
```

```
emp_e = c(var(e[,1]),var(e[,2]),var(e[,3]),var(e[,4]),var(e[,5]))
cat("Empirical Std: ", sqrt(emp_e), "\n")
```

```
## Empirical Std:  0.01234556 0.0123401 0.01233827 0.0123374 0.01234277
```

```
#How well does the empirical forecast follow the theoretical formulation:
cat("Empirical vs Theoretical: ", sqrt(emp_e[2]) - sqrt((1+b1^2)*emp_e[1]), "\n")
```

```
## Empirical vs Theoretical:  -1.090619e-05
```

The theoretical values are very close to the actual values. They appear to be constant over time.