Problem Set 3
MFE 402: Econometrics
Professor Rossi

This problem set is designed to review material on the multiple regression model and time series. Include both your R code and output in your answers.

Question 1

Using a sequence of simple regressions computed in `R`, show how to obtain the multiple regression coefficient on $P2$ in the `multi` dataset from the `DataAnalytics` package.

```
library(DataAnalytics)
data("multi")
p2_p1 = lm(multi$p2 ~ multi$p1)

sales_e12 = lm(multi$Sales ~ p2_p1$residuals)
mr = lm(multi$Sales ~ multi$p1 + multi$p2)
cat("The coefficient on p2 is",sales_e12$coefficients[[2]],"\n")
```

```
## The coefficient on p2 is 108.7999
```

```
cat("The difference between using simple regressions and multiple
    regression is", sales_e12$coefficients[[2]] - mr$coefficients[[3]],
    "It's very close to 0.")
```

```
## The difference between using simple regressions and multiple
##     regression is 1.421085e-14 It's very close to 0.
```

Question 2

Use matrix formulas and `R` code – i.e., use `%*%` not `lm` – to reproduce the least squares coefficients and standard errors shown on slide 17 of Chapter II. The `countryret` dataset is in the `DataAnalytics` package.

```
data("countryret")
y = countryret$usa
x = cbind(rep(1,length(y)),countryret$canada, countryret$uk,
          countryret$australia, countryret$france, countryret$germany, countryret$japan)

coefficients = chol2inv(chol(crossprod(x))) %*% crossprod(x,y)

e = y - x %*% coefficients
ssq = sum(e*e)/(length(y) - ncol(x))
Var_b = ssq * chol2inv(chol(crossprod(x)))
std_err = sqrt(diag(Var_b))
row.names(coefficients) = c("Intercept","Canada","UK","Austrilia","France","Germany","Japan")
names(std_err) = c("Intercept","Canada","UK","Austrilia","France","Germany","Japan")
coefficients
```

```
##                    [,1]
## Intercept  0.006135614
## Canada     0.444362109
## UK         0.225690196
## Austrilia -0.056688434
## France     0.166742081
## Germany   -0.064792831
## Japan     -0.051027942
```

```r
std_err
```

```
##  Intercept     Canada         UK  Austrilia     France    Germany 
## 0.00230897 0.06958673 0.06491489 0.05036627 0.06133779 0.05723881 
##      Japan 
## 0.03461495
```

Question 3

Run the regression of `VWNFX` on `vwretd`.

a. Compute a 90% prediction interval for `VWNFX` when `vwretd` = 0.05 using the formulas in the class notes.

$$predict_interval = b_0 + b_1 X_f + -t^*_{N-2,\alpha/2} s_{pred}$$
$$s_{pred} = s(1 + \frac{1}{N} + \frac{(X_f - \bar{X})^2}{(N-1)s_x^2})^{1/2}$$

```r
library(reshape2)
data("Vanguard")
data("marketRf")
Van = Vanguard[c(1,2,5)]
V_reshaped = dcast(Van,date ~ ticker, value.var = "mret")
Van_mkt = merge(V_reshaped,marketRf,by="date")
vwretd = Van_mkt$vwretd
VWNFX = Van_mkt$VWNFX
vwretd = vwretd[-which(is.na(VWNFX))]
VWNFX = VWNFX[-which(is.na(VWNFX))]

out = lm(VWNFX ~ vwretd)
s = summary(out)[[6]]
n = length(vwretd)
s_pred = s * sqrt(1 + 1/n + (0.05 - mean(vwretd))^2/((n - 1) * var(vwretd)))
t = qt(0.95,df = n - 2)
interval_low = out$coefficients[1] + out$coefficients[2] * 0.05 - t * s_pred
interval_upper = out$coefficients[1] + out$coefficients[2] * 0.05 + t * s_pred

cat("90% confident interval is ", interval_low, "to",interval_upper)
```

```
## 90% confident interval is  0.01744646 to 0.07361465
```

b. Check your work in part (a) by computing a 90% prediction interval using R's `predict` command.

```
predict(out,data.frame(vwretd = 0.05),int = "prediction",level = 0.9)
```

```
##          fit        lwr        upr
## 1 0.04553055 0.01744646 0.07361465
```

Question 4

Define the mean return vector and the symmetric variance-covariance matrix for 3 assets as follows:

$$\mu = \begin{bmatrix} 0.010 \\ 0.015 \\ 0.025 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.0016 & 0.0010 & 0.0015 \\ & 0.0020 & 0.0019 \\ & & 0.0042 \end{bmatrix}$$

a. Compute the correlation matrix of these three assets from the variance-covariance matrix Σ by dividing the (i, j) element of Σ by σ_i and σ_j. You must use matrix operations (e.g., `diag()`, `X*Y`, or `X%*%Y`) in your answer. You may not use a loop and you may not use the `R` function `cov2cor`.

```
u = matrix(c(0.01,0.015,0.025),nrow = 3,ncol = 1,byrow = TRUE)
sigma = matrix(c(0.0016,0.0010,0.0015,0.0010,0.002,0.0019,0.0015,0.0019,0.0042),nrow = 3,
               ncol = 3, byrow = TRUE)
diagnal = sqrt(diag(sigma))^-1
diagnal_matrix = diag(diagnal,nrow = 3,ncol = 3)

sigma %*% diagnal_matrix * diagnal
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.5590170 0.5786376
## [2,] 0.5590170 1.0000000 0.6555623
## [3,] 0.5786376 0.6555623 1.0000000
```

b. Compute the mean and standard deviation of a portfolio made from these assets with weights $(0.3, 0.4, 0.3)$

```
weights = matrix(c(0.3,0.4,0.3),nrow = 1,ncol = 3,byrow = TRUE)
mean = weights %*% u
sd = sqrt(weights %*% sigma %*% t(weights))
cat("The mean and standard deviation of the portfolio are", mean, "and", sd, "correspondingly")
```

```
## The mean and standard deviation of the portfolio are 0.0165 and 0.04252058 correspondingly
```

Question 5

Using the same data as in Question 3 above and following the lecture slides (Chapter 3, section g), test the general linear hypothesis that $\beta_{up} = \beta_{down}$ in the following regression. Note that if you account for the `NA` values properly, you should get a slightly different result than what is presented in the lecture slides.

$$VWNFX_t = \alpha + \beta_{up} * vwretd_t^+ + \beta_{down} * vwretd_t^- + \varepsilon_t$$

```r
mkt_up = ifelse(vwretd>0,1,0)
mkt_down = 1 - mkt_up
mkt_up = mkt_up * vwretd
mkt_down = mkt_down * vwretd
mkt_timing = lm(VWNFX ~ mkt_up + mkt_down)
R = matrix(c(0,1,-1),byrow = TRUE,nrow = 1)
r = c(0)
x = cbind(c(rep(1,length(vwretd))),mkt_up,mkt_down)
b = as.vector(mkt_timing$coefficients)
QFmat = chol2inv(chol(crossprod(x)))
QFmat = R %*% QFmat %*% t(R)
Violation = R%*%b - matrix(r,ncol = 1)
fnum = t(Violation) %*% chol2inv(chol(QFmat)) %*% Violation
n_minus_k = length(mkt_up) - length(b)
fdenom = nrow(R) * sum(mkt_timing$residuals ** 2)/n_minus_k
f = fnum / fdenom
pvalue = 1 - pf(f,df1=nrow(R),df2 = n_minus_k)
cat("The F test value and p value are", f, "and", pvalue, "correspondingly")
```

```
## The F test value and p value are 0.1558804 and 0.6932308 correspondingly
```

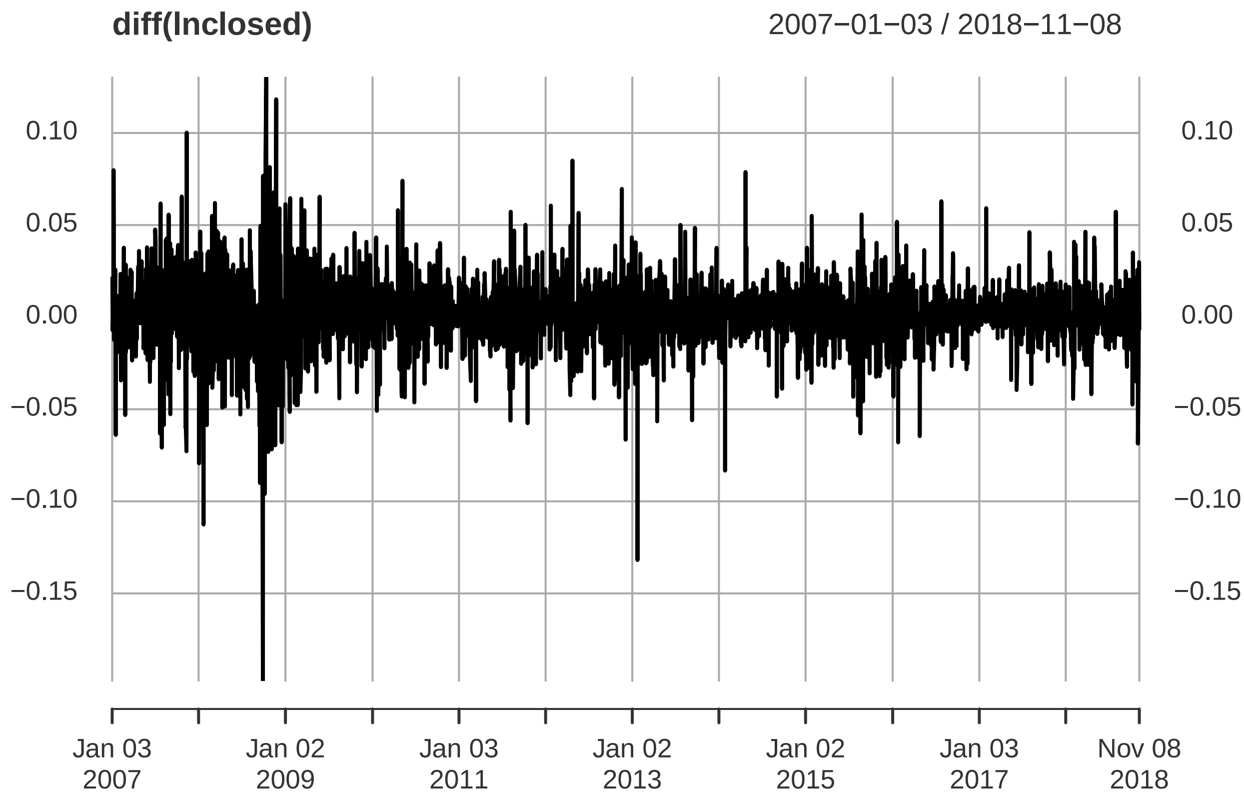
Question 6

Retrieve the Apple stock price series using the `quantmod` package (as done in the notes). Plot the autocorrelations of the difference in log prices.

```r
library("quantmod")
getSymbols("AAPL",getSymbols.yahoo.warning = FALSE)
```

```
## [1] "AAPL"
```

```r
lnclosed = log(AAPL[,4])
plot(diff(lnclosed))
```




```r
acf(diff(lnclosed),na.action = na.omit)
```