Problem Set 3
MFE 402: Econometrics
Professor Rossi
Student: Xiahao Wang
ID: 305229864

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.

ANS: Run the following code:

```
library(DataAnalytics)
data(multi)
e_2.1 = lm(p2~p1,data = multi)$residuals
lm(Sales ~ e_2.1, data = multi)
```

```
##
## Call:
## lm(formula = Sales ~ e_2.1, data = multi)
##
## Coefficients:
## (Intercept)        e_2.1
##       517.1        108.8
```

Verify using Multi-linear Regression:

```
out=with(multi,
     lm(Sales~p1+p2)
)
lmSumm(out)
```

```
## Multiple Regression Analysis:
##     3 regressors(including intercept) and 100 observations
##
## lm(formula = Sales ~ p1 + p2)
##
## Coefficients:
##             Estimate Std Error t value p value
## (Intercept)   115.70     8.548   13.54       0
## p1            -97.66     2.669  -36.60       0
## p2            108.80     1.409   77.20       0
## ---
## Standard Error of the Regression:  28.42
## Multiple R-squared:  0.987  Adjusted R-squared:  0.987
## Overall F stat: 3717.29 on 2 and 97 DF, pvalue= 0
```

The coefficient is 108.8

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)
y <- as.matrix(y,ncol=1)
x <- as.matrix(x, ncol=6)
b <- chol2inv(chol(crossprod(x))) %*% crossprod(x,y)
e <- y - x%*%b
ssq <- sum(e*e)/(length(y)-ncol(x))
var_b <- ssq * chol2inv(chol(crossprod(x)))
std_err <- sqrt(diag(var_b))
names(std_err) <- c("intercept", "canada","uk","australia", "france","germany","japan")
```

Coefficients

```
b <- as.vector(b)
names(b) <- c("intercept", "canada","uk","australia", "france","germany","japan")
b
```

```
##    intercept       canada           uk    australia       france
##  0.006135614  0.444362109  0.225690196 -0.056688434  0.166742081
##      germany        japan
## -0.064792831 -0.051027942
```

Standard Errors

```
std_err
```

```
##  intercept     canada         uk  australia     france    germany
## 0.00230897 0.06958673 0.06491489 0.05036627 0.06133779 0.05723881
##      japan
## 0.03461495
```

Verify Using `lm`

```
out = lm(usa~canada + uk + australia + france + germany + japan, data=countryret)
lmSumm(out)
```

```
## Multiple Regression Analysis:
##     7 regressors(including intercept) and 107 observations
##
## lm(formula = usa ~ canada + uk + australia + france + germany +
##     japan, data = countryret)
##
## Coefficients:
##              Estimate Std Error t value p value
## (Intercept)  0.006136  0.002309    2.66   0.009
## canada       0.444400  0.069590    6.39   0.000
## uk           0.225700  0.064910    3.48   0.001
## australia   -0.056690  0.050370   -1.13   0.263
## france       0.166700  0.061340    2.72   0.008
## germany     -0.064790  0.057240   -1.13   0.260
```

```
## japan        -0.051030  0.034610   -1.47   0.144
## ---
## Standard Error of the Regression:  0.02257
## Multiple R-squared:  0.566  Adjusted R-squared:  0.54
## Overall F stat: 21.74 on 6 and 100 DF, pvalue= 0
```

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

(a)

```
library(reshape2)
data(Vanguard)
data(marketRf)
vwretd_date <-marketRf[c("date","vwretd")]
VWNFX <- subset(Vanguard, ticker == "VWNFX")
vanguard_reshaped <- dcast(VWNFX, date~ticker,value.var="mret")
merged_data <- merge(vanguard_reshaped,vwretd_date,by="date")
out <- lm(VWNFX~vwretd, data = merged_data)
lmSumm(out)
```

```
## Multiple Regression Analysis:
##     2 regressors(including intercept) and 336 observations
##
## lm(formula = VWNFX ~ vwretd, data = merged_data)
##
## Coefficients:
##             Estimate Std Error t value p value
## (Intercept) 0.001074 0.0009468    1.13   0.257
## vwretd      0.889100 0.0207300   42.90   0.000
## ---
## Standard Error of the Regression:  0.01698
## Multiple R-squared:  0.846  Adjusted R-squared:  0.846
## Overall F stat: 1840.19 on 1 and 334 DF, pvalue= 0
```

Standard error of the regression is 0.01698 Using the formual for s_{pred}:

$$s_{pred} = s\sqrt{1 + \frac{1}{N} + \frac{(X_f - \bar{X})^2}{\sum_{i=0}^{N}(X_i - \bar{X})^2}}$$

```
b_1 <- out$coefficients[2]
b_0 <- out$coefficients[1]
x_i <- merged_data$vwretd
x_bar <- mean(merged_data$vwretd)

x_f <- 0.05
N <- 336
df <- N-2

y.hat <- b_0 + b_1 * merged_data$vwretd
x <- merged_data$vwretd
```

```
s_square <- sum((merged_data$VWNFX - y.hat)^2)/(df)
s <- sqrt(s_square)
denominator <- sum((x_i - x_bar)^2)

spred <- s * sqrt(1 + (1/N) + (x_f - (x_bar))^2 / denominator )

t_value <- qt(0.95, df = 334)
fittedV <- b_0 + b_1 * x_f
upper <- b_0 + b_1 * x_f + t_value * spred
lower <- b_0 + b_1 * x_f - t_value * spred
predict_res <- data.frame(as.numeric(lower), as.numeric(upper), as.numeric(fittedV))
colnames(predict_res) <- c("Lower", "Upper","Fitted")
predict_res
```

```
##        Lower      Upper     Fitted
## 1 0.01744646 0.07361465 0.04553055
```

(b) verify using `predict`

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

(a) Initialize matrix μ and Σ

```
mu <- matrix(c(0.01,0.015,0.025))
covariance_mat <- matrix(c(0.0016, 0.0010, 0.0015,
                           0,0.0020,0.0019,0,0,0.0042), nrow=3, ncol=3, byrow=TRUE)
variance <- matrix(diag(covariance_mat))
sigma <- sqrt(variance)
D <- sigma %*% t(sigma)
covariance_mat * (1/D)
```

```
##      [,1]     [,2]      [,3]
## [1,]    1 0.559017 0.5786376
## [2,]    0 1.000000 0.6555623
## [3,]    0 0.000000 1.0000000
```

verify using `cov2cor`

```
cov2cor(covariance_mat)
```

```
##      [,1]     [,2]      [,3]
## [1,]    1 0.559017 0.5786376
## [2,]    0 1.000000 0.6555623
## [3,]    0 0.000000 1.0000000
```

(b)

```r
covariance_mat <- matrix(c(0.0016, 0.0010, 0.0015, 0.001,
                           0.0020,0.0019,0.0015,0.0019,0.0042), nrow=3, ncol=3, byrow=TRUE)
weights <- matrix(c(0.3,0.4,0.3), nrow = 3, ncol=1)
expected_ret <- t(weights) %*% mu
variance_port <- t(weights) %*% covariance_mat %*% weights
sd_port <- sqrt(variance_port)
qn4_res <- data.frame(expected_ret, sd_port)
colnames(qn4_res) <- c("Expected Return", "Standard Deviation")
qn4_res
```

```
##   Expected Return Standard Deviation
## 1          0.0165         0.04252058
```

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
data(Vanguard)
van <- Vanguard[,c(1,2,5)]
v_reshaped <- dcast(van,date~ticker, value.var = "mret")
reshaped_cleaned <- v_reshaped[-which(is.na(v_reshaped$VWNFX)),]
data(marketRf)
Van_mkt <- merge(reshaped_cleaned,marketRf, by="date")
mkt_up <- ifelse(Van_mkt$vwretd > 0,1,0)
Van_mkt$upvw <- mkt_up * Van_mkt$vwretd
Van_mkt$dwnvw <- (1-mkt_up) * Van_mkt$vwretd
mkt_timing <- lm(VWNFX~upvw+dwnvw, data = Van_mkt)
lmSumm(mkt_timing)
```

```
## Multiple Regression Analysis:
##     3 regressors(including intercept) and 336 observations
##
## lm(formula = VWNFX ~ upvw + dwnvw, data = Van_mkt)
##
## Coefficients:
##             Estimate Std Error t value p value
## (Intercept) 0.001526  0.001486    1.03   0.305
## upvw        0.876200  0.038740   22.62   0.000
## dwnvw       0.901600  0.037830   23.83   0.000
## ---
## Standard Error of the Regression:  0.017
## Multiple R-squared:  0.846  Adjusted R-squared:  0.846
## Overall F stat: 917.85 on 2 and 333 DF, pvalue= 0
```

```r
R <- matrix(c(0,1,-1), byrow=TRUE, nrow=1)
r <- c(0)
X <- cbind(c(rep(1, nrow(Van_mkt))), Van_mkt$upvw, Van_mkt$dwnvw)
b <- as.vector(mkt_timing$coef)
QFmat <- chol2inv(chol(crossprod(X)))
QFmat <- R%*%QFmat%*%t(R)
Violation <- R%*%b - matrix(r, ncol=1)
fnum <- t(Violation)%*%chol2inv(chol(QFmat))%*%Violation
n_minus_k <- nrow(Van_mkt) - length(b)
fdenom <- nrow(R)*sum(mkt_timing$resid ** 2)/n_minus_k
f <- fnum/ fdenom
pvalue <- 1 - pf(f, df1 = nrow(R), df2 =n_minus_k)
qn5_res <- data.frame(f , pvalue)
colnames(qn5_res) <- c("F", "P value")
qn5_res
```

```
##           F   P value
## 1 0.1558804 0.6932308
```

Given as fvalue is small and pvalue is bigger than 5% CI, we fail to reject the hypothesis.

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)
```

```
## 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
getSymbols(Symbols = 'AAPL')
```

```
## '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.

##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
```

```
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

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

```r
dif_log_price <- diff(log(AAPL$AAPL.Close), difference = 1)
acf(dif_log_price[-1,])
```