Computer Lab Session 2:
Running Simple Regression and Performing Hypothesis Testing in R
EF3450 C01/CA1(Dr. Yan's section)
Semester B 2017-18

This example is similar to the question in homework #2 except that it uses a different data and the hypothesis testing. This handout explains the R commands as well as output.

(R exercise) Simple Regression and Hypothesis Testing

Data on the monthly stock prices of IBM (p_IBM), S&P500 market index (p_SPX) and the risk free rate (r_f) are provided in the file **CAPM_dataset_3.csv** which is available on the class web page.

Calculate the monthly returns of the IBM and the S&P500 as

$$r_{security,t} = (P_{IBM,t} - P_{IBM,t-1})/P_{IBM,t-1} * 100\%$$

and

$$r_{mkt,t} = (P_{SPX,t} - P_{SPX,t-1})/P_{SPX,t-1} * 100\%$$

Estimate the CAPM model

$$(r_{security,t} - r_{f,t}) = \beta_1 + \beta_2 * (r_{mkt,t} - r_{f,t}) + e_t$$

(a) Report the estimated intercept and slope $\hat{\beta}_1$ and $\hat{\beta}_2$.

```
setwd('C:\\Users\\YOURACCOUNT\\Desktop')
data_ex2 <- read.csv('CAPM_dataset_3.csv',
                 stringsAsFactors = F)
data_ex2 <- transform(data_ex2,
                 Date = as.Date(Date, "%m/%d/%Y")) # Transform Column 'type'
str(data_ex2)
```

The 'read.csv()' function imports the data from the file "'CAPM_dataset_3.csv" and output the content to a data frame called "data_ex2". And we transform the Date column and replace the result to the same column

The data frame 'data_ex2' includes the following columns:
- Column 1 Date: First day of the month MM/DD/YYYY
- Column 2 p_IBM: Monthly price (average) of IBM (risky security),
- Column 3 p_SPX: Monthly price (average) of SP500 index (market), and
- Column 4 r_f: 3 month T-bill rate (risk free asset)

```
> str(data_ex2)
'data.frame':	199 obs. of  4 variables:
 $ Date : Date, format: "2000-11-01" "2000-12-01" "2001-01-01" "2001-02-01" ...
 $ p_IBM: num  93.5 85 112 99.9 96.2 ...
 $ p_SPX: num  1315 1320 1366 1240 1160 ...
 $ r_f  : num  0.511 0.479 0.405 0.4 0.351 ...
```

```
ret_name  <- c('r_IBM', 'r_SPX')
ret_df <- data.frame(matrix(ncol = length(ret_name) + 1 ,
                            nrow = nrow(data_ex2) - 1 ))
ret_df[,1] <- data_ex2[-1,1]
colnames(ret_df) <- c('Date', ret_name) #and set column names
ret_df[, -1] <- sapply(data_ex2[,-1],
                       function(x){
                         diff(x)/x[-length(x)]*100
                       }
)
ret_df$r_f <- data_ex2[-1, 'r_f']
ret_df$ex_r_IBM <- ret_df$r_IBM - ret_df$r_f
ret_df$ex_r_SPX  <- ret_df$r_SPX  - ret_df$r_f
str(ret_df)
```

Create a data frame for returns. Starting by assigning values of column 1 (data_ex2, excluding the first value) to the 1st column in ret_df. Then taking data_ex2 as a list and apply each element (column) excluding the first ('Date') to the function we wrote in 2nd argument, and return the value to each column in ret_df [2 and 3 excluding 1st]. We can generate two more columns for excess returns of risky asset and market but it is optional.

The data frame ret_df includes the following columns:
- Column 1 Date: First day of the month
- Column 2 r_IBM: Monthly return of IBM (risky security),
- Column 3 r_SPX: Monthly return of SP500 index (market),
- Column 4 r_f: 3 month T-bill rate (risk free asset).
- Column 5 ex_r_IBM: excess return of IBM (risky security), and
- Column 6 ex_r_SPX: excess return of SP500 index (market)

```
> str(ret_df)
'data.frame':   198 obs. of  6 variables:
 $ Date    : Date, format: "2000-12-01" "2001-01-01" "2001-02-01" "2001-03-01" ...
 $ r_IBM   : num  -9.09 31.76 -10.8 -3.72 19.71 ...
 $ r_SPX   : num  0.405 3.464 -9.229 -6.42 7.681 ...
 $ r_f     : num  0.479 0.405 0.4 0.351 0.318 ...
 $ ex_r_IBM: num  -9.57 31.36 -11.2 -4.07 19.4 ...
 $ ex_r_SPX: num  -0.0736 3.0585 -9.6291 -6.7717 7.3639 ...
```

```
reg_CAPM <- lm(I(r_IBM-r_f) ~ I(r_SPX-r_f), data = ret_df)
reg_CAPM2 <- lm(ex_r_IBM ~ ex_r_SPX, data = ret_df) #Alternatively
```

To run the regression we need to first fit the linear model (generate a lm object Regression). If excess returns are not generated as above, we need to use function I() to bracket those portions of a model formula where the operators are used in their arithmetic sense. [For example, in the formula y ~ a + I(b+c), the term b+c is to be interpreted as the sum of b and c.]

```
summary(reg_CAPM)
```

By providing a lm object to summary(), R will print out the regression result.

```
> summary(reg_CAPM) # summary(reg_CAPM2) will return same result

Call:
lm(formula = I(r_IBM - r_f) ~ I(r_SPX - r_f), data = ret_df)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.8293  -3.0462  -0.4632   2.3378  28.1807 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     0.08970    0.39952   0.225    0.823    
I(r_SPX - r_f)  1.01004    0.09498  10.634   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.609 on 196 degrees of freedom
Multiple R-squared:  0.3659,	Adjusted R-squared:  0.3626 
F-statistic: 113.1 on 1 and 196 DF,  p-value: < 2.2e-16
```

Inference using t-distribution requires df

Recap: Report the estimated intercept and slope $\hat{\beta}_1$ and $\hat{\beta}_2$, and answer whether IBM outperform or underperform the market and the volatility of IBM compared with the volatility of the market portfolio, which was covered in lecture/ lecture slide.

(b) Test the null hypothesis that $\beta_2 \le 1$ (the stock is a defensive asset), against the alternative hypothesis that $\beta_2 > 1$ (this stock is an aggressive asset) at the 5% level of significance by comparing the test statistic with the critical values.

(Hint: Since the hypothesized value under the null is 1 instead of 0, you cannot directly take the t-statistic from the computer output to carry out this hypothesis testing. You need to compute the test statistic using the formula $t = \dfrac{\hat{\beta}_2 - 1}{se(\hat{\beta}_2)}$ by yourself.)

This is a one-sided right-tail test. H0: beta_2 <= 1 against H1: beta_2 > 1
We first compute the test statistics

```
tmp_123 <- summary(reg_CAPM)
beta_1 <- tmp_123$coefficients[2,1]  #Beta_2 hat
sd_1 <- tmp_123$coefficients[2,2] #SE of Beta_2 hat
t_stat_1 <- ( beta_1 - 1 ) /  sd_1 # the test statistic
```

As we provide an lm object as the 1st argument for summary function and assign it to tmp_123, the regression summary will not be printed immediately. We could use it to retrieve the coefficient estimate and standard error to compute the test statistic ‘t_stat_1’

Then compare t_stat_1 with critical value (or calculate the associated p-value)

Approach 1: Compare the test statistic with critical value *test statistic* $t > t_{c,T-K}(\alpha)$

Critical values can be found using quantile function (are named in the form qxxx in R)
For Student's t distribution: qt
For normal distribution: qnorm.

```
qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

```
critical_val_t <- qt(1-0.05 , 196)
critical_val_t
# Calculated from the sample to a standard normal distribution
critical_val_norm <- qnorm(0.95)
# reject H0 if below return True, [Do not reject if below statement is False]
t_stat_1 > critical_val_t                #>critical_val_norm if using std norm
```

Approach 2: calculate the associated p-value using pt() or pnorm():

```
pt(q, df, ncp, lower.tail = TRUE, log.p = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

`lower.tail` logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.

```
## Reject H0 if p-value is smaller than 0.05,
## and cannot rejected H0 otherwise
1 - pt(t_stat_1,196)  < 0.05                   # 1 - pnorm(t_stat_1,196) < 0.05 if using std norm
```

```
> 1- pt(t_stat_1,196) > 1- pnorm(t_stat_1)
[1] 0.45798             [1] 0.457926

> tmp_123$coefficients
                 Estimate Std. Error    t value     Pr(>|t|)
(Intercept)     0.08969882 0.39952307  0.2245147 8.225907e-01
I(r_SPX - r_f)  1.01003596 0.09498343 10.6338125 3.839780e-21
```

(c) Test the null hypothesis that β_1 <= 0 (IBM is underperform or as same as market return), against the alternative hypothesis that IBM is outperform (β_1 >0), at the 5% level of significance using the reported p-value.

Again this is a one-sided right-tail test. We can compare the test statistics with critical value or calculate the associated p-value. Let's call corresponding test statistic 't_stat_2'

```
beta_2 <- tmp_123$coefficients[1,1] #Beta_1 hat
sd_2 <- tmp_123$coefficients[1,2]  #SE of Beta_1 hat
t_stat_2 <- ( beta_2 - 0 ) /  sd_2
t_stat_2 ## the test statistic
```

t-statistic from the summary output use 0 as hypothesized value under the null so we could use it diretly in this case (But of course that's no harm to carry out the computation

Approach 1: Compare the test statistic with critical value *test statistic* $t > t_c$

Then compare t_stat_1 with critical value (or calculate the associated p-value), which was computed in part b

```
# reject H0 if below return True, [Do not reject if below statement is False]
t_stat_2 > critical_val_t              #>critical_val_norm if using std norm
```

Approach 2: calculate the associated p-value using pt() or pnorm():

```
## Reject H0 if p-value is smaller than 0.05,
## and cannot rejected H0 otherwise
1 - pt(t_stat_2,196)  < 0.05                  # 1 - pnorm(t_stat_2,196) < 0.05 if using std norm
```

```
> 1 - pt(t_stat_2,196)  > 1 - pnorm(t_stat_2)
[1] 0.4112953           [1] 0.4111784

> tmp_123$coefficients
                Estimate Std. Error    t value     Pr(>|t|)
(Intercept)   0.08969882 0.39952307  0.2245147 8.225907e-01
I(r_SPX - r_f) 1.01003596 0.09498343 10.6338125 3.839780e-21

> t_stat_2 == tmp_123$coefficients[1,3]
[1] TRUE
```

(d) Test the null hypothesis that β_2 is zero, against the alternative hypothesis that it is positive, at the 5% level of significance using the reported p-value.

This time we have a two-tail test. Let's name the corresponding test statistic 't_stat_3'

```
beta_3 <- tmp_123$coefficients[2,1]
sd_3 <- tmp_123$coefficients[2,2]
t_stat_3 <- ( beta_3 - 0 ) /  sd_3
t_stat_3 ## the test statistic
```

```
# Alternative you can use tmp_123$coefficients[2,3] as H0: beta_2 = 0
> t_stat_3 == tmp_123$coefficients[2,3]
[1] TRUE
```

Approach 1: Compare the test statistic with critical value. This is a two tailed- test and hence we reject H0 if $|test\ statistic\ t| > t_{c,T-K}\left(\frac{\alpha}{2}\right)$

```
critical_val_t_2_tailed <- qt(0.975 , 196)
# Calculated from the sample to a standard normal distribution
critical_val_norm_2_tailed <- qnorm(0.975)

# reject H0 if below return True, [Do not reject if below statement is False]
abs(t_stat_3) > critical_val_t
```

```
> critical_val_t_2_tailed
[1] 1.972141
> critical_val_norm_2_tailed
[1] 1.959964
```

Abs() return the absolute value of t_stat_3

Approach 2: calculate the associated p-value using pt() or pnorm():

```
##(Equivalently): Reject H0 if p-value is smaller than 0.05%,
## and cannot rejected H0 otherwise
p_val_d = 2 * (1 - pt(t_stat_3, 196) )
p_val_d < 0.05
```