Computer Lab Session 2: Simple Regression

EF 3450

February 5, 2018

Example: Capital Asset Pricing Model

We will use CAPM in this exercise as it is an application of simple linear regression.

Just to brush up on the theory side,

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

where

1. $r_{security,t}$: return of security at period t
2. $r_{mkt,t}$: market return at period t
3. $r_{f,t}$: risk free rate at period t
4. e_t: error term at period t
5. β_2 estimate the sensitivity of a security in comparison of the market, and
6. β_1 (Intercept) captures abnormal return if it is statistically significant from 0.

Know your data

Now we are given a csv file **CAPM_dataset_3.csv**. This file consists the average monthly price of the following assets ranging from Nov 2000 to May 2017

```
- closing price of IBM (risky security),
- closing price of SP500 index (market), and
- 3 month T-bill (risk free asset).
```

Display the data

Now let's load the data and have a look on it's structure:

```
setwd('C:\\Users\\dmtwong\\Desktop')
data_ex2 <- read.csv('CAPM_dataset_3.csv',
                     stringsAsFactors = F)
str(data_ex2)
```

```
## 'data.frame':    199 obs. of  4 variables:
##  $ Date : chr  "11/1/2000" "12/1/2000" "1/1/2001" "2/1/2
##  $ 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 ...
```

Data processing

As you can see, this dataset need to be further processed before starting the analysis:

```
1. Date is read as 'character' type ( result of
stringsAsFactors = F ): Convert it to Date-Time
class such as 'Date' and 'POSIXlt' type

2. CAPM model use asset return:
   Transform from price series to returns
```

Convert to ‘Date’ Class

From csv, ‘Date’ is stored as: **MM/DD/YYYY** The corresponding conversion specification is **%m/%d/%Y** (capital sensitive).

`Note: If date/time stored in different format, read the R Documentation of strptime (run ?strptime)`

```r
data_ex2 <- transform(data_ex2,
                      Date = as.Date(Date, "%m/%d/%Y"))
str(data_ex2)
```

```
## 'data.frame':    199 obs. of  4 variables:
##  $ Date : Date, format: "2000-11-01" "2000-12-01" ...
##  $ 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 ...
```

Create a empty data frame

```
list_name <- c('p_IBM', 'p_SPX')
ret_name  <- c('r_IBM', 'r_SPX')
ret_df <- data.frame(matrix(ncol = length(ret_name) + 1 ,
                            nrow = nrow(data_ex2) - 1 ))
```

Then insert first column using 'Date' from 'data_ex2' and set column names

```
ret_df[,1] <- data_ex2[-1,1]
colnames(ret_df) <- c('Date', ret_name)
head(ret_df,1); tail(ret_df,1)
```

```
##         Date r_IBM r_SPX
## 1 2000-12-01    NA    NA

##           Date r_IBM r_SPX
## 198 2017-05-01    NA    NA
```

2: Calculate return

```r
ret_df[, -1] <- sapply(data_ex2[,-1],
                       function(x){
                         diff(x)/x[-length(x)]*100
                       }
)
str(ret_df);head(ret_df,1); tail(ret_df,1)
```

```
## 'data.frame':    198 obs. of  3 variables:
##  $ Date : Date, format: "2000-12-01" "2001-01-01" ...
##  $ 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 ...
```

```
##         Date     r_IBM     r_SPX
## 1 2000-12-01 -9.090909 0.4053386
```

```
##           Date     r_IBM    r_SPX
## 198 2017-05-01 -4.778838 1.157621
```

Check first return of IBM (optional)

```r
head(data_ex2, 2)
```

```
##         Date p_IBM   p_SPX       r_f
## 1 2000-11-01  93.5 1314.95 0.5110175
## 2 2000-12-01  85.0 1320.28 0.4789658
```

```r
head(ret_df, 1)
```

```
##         Date     r_IBM     r_SPX
## 1 2000-12-01 -9.090909 0.4053386
```

```r
ret_df[1, 2] == diff(data_ex2[1:2, 2])/data_ex2[1,2]*100 #
```

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

Check last return of IBM (optional)

```
tail(ret_df, 1)
```

```
##           Date     r_IBM    r_SPX
## 198 2017-05-01 -4.778838 1.157621
```

```
tail(data_ex2, 2)
```

```
##           Date  p_IBM  p_SPX        r_f
## 198 2017-04-01 160.29 2384.2 0.06448000
## 199 2017-05-01 152.63 2411.8 0.07687833
```

```
ret_df[nrow(ret_df), 2] == diff(data_ex2[198:199, 2])/
  data_ex2[198,2]*100 # last return of IBM
```

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

Note: These test cases are far from comprehensive, and it does not guarantee what we did is correct

Now add the risk free rate to ret_df as well

```
ret_df$r_f <- data_ex2[-1, 'r_f']
str(ret_df)
```

```
## 'data.frame':    198 obs. of  4 variables:
##  $ Date : Date, format: "2000-12-01" "2001-01-01" ...
##  $ 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 ...
```

```
head(ret_df, 1)
```

```
##         Date     r_IBM     r_SPX       r_f
## 1 2000-12-01 -9.090909 0.4053386 0.4789658
```

```
tail(ret_df, 1)
```

```
##           Date     r_IBM    r_SPX        r_f
## 198 2017-05-01 -4.778838 1.157621 0.07687833
```

(Optional) Generate excess return

To be specific, CAPM regress ‘excess returns of asset against excess returns on market We could perform arithmetic operation with I() for formula of ’lm’ function, but we could also generate the excess return in the data frame explicitly

```
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)
```

```
## 'data.frame':    198 obs. of  6 variables:
##  $ Date     : Date, format: "2000-12-01" "2001-01-01" ...
##  $ 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
```

Regression

```
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) #Altern
```

summary(reg_CAPM)

```
##
## Call:
## lm(formula = I(r_IBM - r_f) ~ I(r_SPX - r_f), data = ret
##
## 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.
##
## 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
```

summary(reg_CAPM2) would be the same

```
##
## Call:
## lm(formula = ex_r_IBM ~ ex_r_SPX, 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
## ex_r_SPX     1.01004    0.09498  10.634   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.
##
## Residual standard error: 5.609 on 196 degrees of freedo
## Multiple R-squared:  0.3659, Adjusted R-squared:  0.362
## F-statistic: 113.1 on 1 and 196 DF,  p-value: < 2.2e-16
```

hypothesis testing using t-distribution (Small sample)

Hypothesis Testing 1: Can we describe IBM as an aggressive asset?
Approach 1: Compare with critical value (using t distribution)
Step 1: Compute the test statistic first

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

```
## [1] 1.010036
```

```
## [1] 0.09498343
```

```
## [1] 0.1056601
```

Can we describe IBM as an aggressive asset? (cont)

Step 2: Compare with critical value using t distribution

```
critical_val_t <- qt(1-0.05 , 196)
critical_val_normal <- qnorm(1-0.05)
# As sample size grows, df is larger and t is closer to st
c(critical_val_normal, critical_val_t)
```

```
## [1] 1.644854 1.652665
```

```
t_stat_1 > critical_val_t
```

```
## [1] FALSE
```

```
# Conclusion: Not an aggressive asset
```

Hypothesis Testing 2: Does IBM outperform the market?

Approach 2: Compare p-value with significance level (say 5%)
Step 1: Again compute the test statistic first

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

```
## [1] 0.08969882
```

```
## [1] 0.3995231
```

```
## [1] 0.2245147
```

Does IBM outperform the market? (cont)

Step 2: Compute p(X $<=$ t statistic) following t distribution

```
pt(t_stat_2, df=196, lower.tail = T)
```

```
## [1] 0.5887047
```

```
pt(t_stat_2, df= 196, lower.tail = T) > 0.05
```

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

```
# Conclusion: Does not outperform the market
```

Hypothesis Testing 3: Does performance of IBM really related to business cycle??

Step 1: t-stat

```
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
```

```
## [1] 10.63381
```

Does IBM really related to business cycle?? (cont)

```
critical_val_3_norm <- qnorm( 1- 0.05/2 ) #0.975 quantile
critical_val_3_t <- qt( 1-0.05/2 , 196) # 2 tail test, eac
c(critical_val_3_norm, critical_val_3_t) # very similar as
```

```
## [1] 1.959964 1.972141
```

```
abs(t_stat_3) > critical_val_3_t
```

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

```
2 * ( 1- pt(t_stat_3, df = 196 ) )
```

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

```
# p-value is so small close to 0 reject the H0: beta = 0
# Conclusion: Does related to business cycle
```

hypothesis testing using normal-distribution (large sample)

As sample size grows, coefficient estimates are asymptotically normal distributed Let's revisit the hypothesis testing using normal distribution

Hypothesis Testing 1: Can we describe IBM as an aggressive asset?
Approach 1: Compare with critical value (using t and normal distribution)
Step 1: Compute the test statistic first

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

```
## [1] 1.010036

## [1] 0.09498343

## [1] 0.1056601
```

Can we describe IBM as an aggressive asset? (cont)

Step 2: Compare with critical value using normal distribution

```
critical_val_t <- qt(1-0.05 , 196)
critical_val_normal <- qnorm(1-0.05)
# with large sample size, df is large and t is close to st
c(critical_val_normal, critical_val_t)
```

```
## [1] 1.644854 1.652665
```

```
t_stat_1 > critical_val_normal
```

```
## [1] FALSE
```

```
# Conclusion: Not an aggressive asset
```

Hypothesis Testing 2: Does IBM outperform the market?

Approach 2: Compare p-value with significance level (say 5%)
Step 1: Again compute the test statistic first

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

```
## [1] 0.08969882
```

```
## [1] 0.3995231
```

```
## [1] 0.2245147
```

Does IBM outperform the market? (cont)

Step 2: Compute p(X $<=$ t statistic) following normal distribution

```
pnorm(t_stat_2, lower.tail = T)
```

```
## [1] 0.5888216
```

```
pnorm(t_stat_2, lower.tail = T) > 0.05
```

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

```
# Conclusion: Does not outperform the market
```

Hypothesis Testing 3: Does performance of IBM really related to business cycle??

Step 1: t-stat

```
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
```

```
## [1] 10.63381
```

Does IBM really related to business cycle?? (cont)

```
critical_val_3_norm <- qnorm( 1- 0.05/2 ) #0.975 quantile
critical_val_3_t <- qt( 1-0.05/2 , 196) # 2 tail test, each
c(critical_val_3_norm, critical_val_3_t) # very similar as
```

```
## [1] 1.959964 1.972141
```

```
abs(t_stat_3) > critical_val_3_norm
```

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

```
2* (1- pnorm(t_stat_3, lower.tail = TRUE))
```

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

```
# p-value is so small and close to 0 and reject the H0: be
# Conclusion: Does related to business cycle
```

Optional: Rounding result

```
round(tmp_123$coefficients, 4)
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.0897     0.3995  0.2245   0.8226
## I(r_SPX - r_f)     1.0100     0.0950 10.6338   0.0000
```