ST5218_Tut6

Name:Zhu Xu
User ID:E0337988
Matriculation ID:A0191344H

26/Mar/2019

1:

Based on the CAPM

$$R_i = r_f + \beta_i(R_m - r_f) + \epsilon_i = (1 - \beta_i)r_f + \beta_i R_m + \epsilon_i$$

Compared with $R_i = a_i + b_i R_M + \epsilon_i$, we get $a_i = (1 - \beta_i)r_i$ and $b_i = \beta_i$

Thus,

$$a_i = (1 - b_i)r_f = r_f - r_f b_i$$

$\Rightarrow \quad c = r_f, d = -r_f$, both of them are constant.

2:

```r
library(tseries)
library(timeSeries)
```

```
## Loading required package: timeDate
```

```r
name0 = ""
name0[1]="AXP"
name0[2]="BA"
name0[3]="CAT"
name0[4]="CSCO"
name0[5]="CVX"
name0[6]="DWDP"
name0[7]="DIS"
name0[8]="GE"
name0[9]="GS"
name0[10]="HD"
name0[11]="IBM"
name0[12]="INTC"
name0[13]="JNJ"
name0[14]="JPM"
name0[15]="KO"
```

```r
name0[16]="MCD"
name0[17]="MMM"
name0[18]="MRK"
name0[19]="MSFT"
name0[20]="NKE"
name0[21]="PFE"
name0[22]="PG"
name0[23]="T"
name0[24]="TRV"
name0[25]="UNH"
name0[26]="UTX"
name0[27]="V"
name0[28]="VZ"
name0[29]="WMT"
name0[30]="XOM"

x = get.hist.quote(instrument = "^DJI",
                   start="2000-01-01",
                   end='2019-03-05',
                   quote = c("AdjClose"),
                   provider ="yahoo",
                   compression = "m")
```

```
## '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).
```

```
## time series ends   2019-03-01
```

```r
for (i in 1:30){
  xi = get.hist.quote(instrument = name0[i],
                      start="2000-01-01",
```

```
                      end='2019-03-05',
                      quote = c("AdjClose"),
                      provider = "yahoo",
                      compression = "m")
  x = merge(x, xi)
}
```

```
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series starts 2008-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
## time series ends   2019-03-01
```

```
R = diff(log(x))
R = na.omit(R)

coeff=matrix(0,30,2)
```

```r
for(i in 1:30){
        coeff[i,1]=lm(R[,1]~R[,i])$coefficients[1]
        coeff[i,2]=lm(R[,1]~R[,i])$coefficients[2]
}
alpha=coeff[,1]
beta=coeff[,2]
```

a)

```r
t.test(alpha)
```

```
##
##  One Sample t-test
##
## data:  alpha
## t = 4.9296, df = 29, p-value = 3.085e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.001106685 0.002676139
## sample estimates:
##   mean of x
## 0.001891412
```

```r
t.test(beta)
```

```
##
##  One Sample t-test
##
## data:  beta
## t = 15.92, df = 29, p-value = 7.169e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.3762402 0.4871642
## sample estimates:
## mean of x
## 0.4317022
```

b)

```r
fm=lm(alpha~beta)
summary(fm)
```

```
##
## Call:
## lm(formula = alpha ~ beta)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0038563 -0.0011808  0.0001864  0.0010237  0.0048040
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.004828   0.001068   4.519 0.000103 ***
## beta        -0.006803   0.002345  -2.902 0.007156 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001875 on 28 degrees of freedom
## Multiple R-squared:  0.2312, Adjusted R-squared:  0.2037
## F-statistic: 8.419 on 1 and 28 DF,  p-value: 0.007156
```

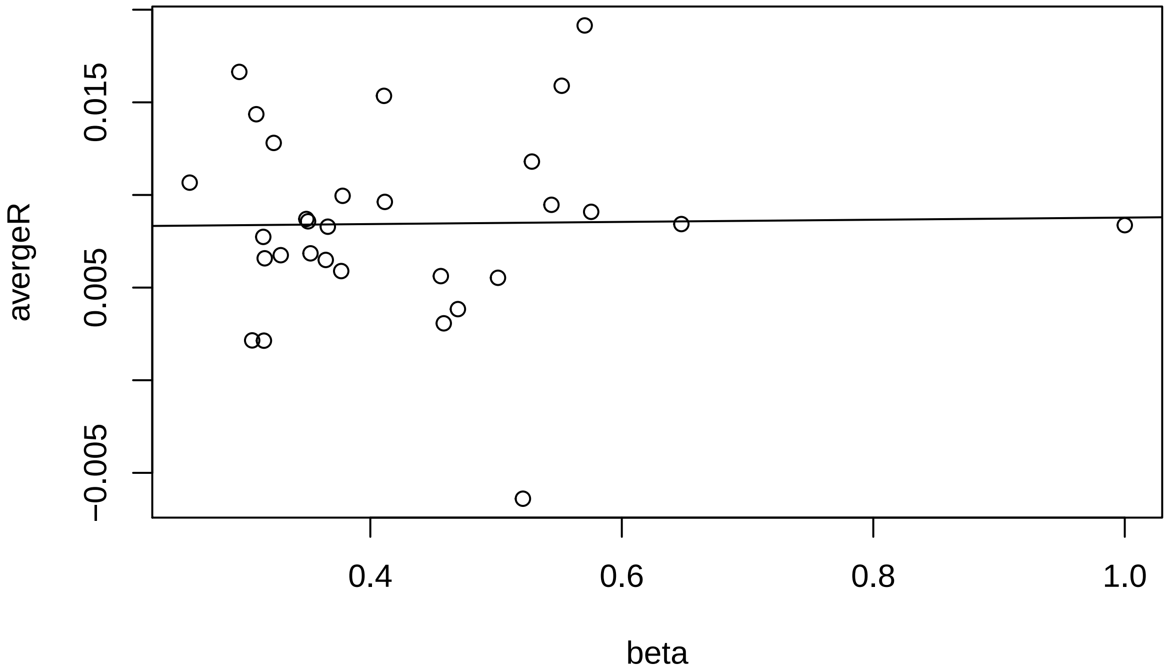
c)

```r
avergeR=apply(R,2,mean)[2:31]
fm2=lm(avergeR~beta)
summary(fm2)
```

```
##
## Call:
## lm(formula = avergeR ~ beta)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0148949 -0.0023713 -0.0001333  0.0021248  0.0106215
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0081959  0.0029413   2.786  0.00946 **
## beta        0.0005832  0.0064540   0.090  0.92864
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005162 on 28 degrees of freedom
## Multiple R-squared:  0.0002916,  Adjusted R-squared:  -0.03541
## F-statistic: 0.008167 on 1 and 28 DF,  p-value: 0.9286
```

d)

```r
plot(beta,avergeR)
abline(a=0.0081959,b=0.0005832)
```




```r
avergeR[which(avergeR>0.0081959+0.0005832*beta)]
```

```
## Adjusted.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.014358659
##        Adjusted.xi.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.010665591
##                 Adjusted.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.016641939
##                         Adjusted.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.009626233
##                                 Adjusted.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.011802251
##                                Adjusted.xi.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.009468615
##                                 Adjusted.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.009089659
##                                Adjusted.xi.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.012809162
##                                         Adjusted.x.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.015350006
##                                        Adjusted.xi.x.x.x.x.x.x.x.x.x.x.x
##                                                            0.008702667
##                                                 Adjusted.x.x.x.x.x.x.x.x
```

```
##                                                     0.009954689
##                                                Adjusted.xi.x.x.x
##                                                     0.015895043
##                                                Adjusted.xi.x.x.x
##                                                     0.019150058
##                                                       Adjusted.x
##                                                     0.008577585
```

3:

$$R_i = r_f + \beta_i(R_m - r_f) + \epsilon_i$$

Where R_i is the net return of asset i during the period $[0, T]$, R_m is the market net return during the same period.

$$E(R_i) = r_f + \beta_i[E(R_m) - r_f] = r_f + \frac{cov(R_i, R_m)}{cov(R_m)}[E(R_m) - r_f]$$

$$E(P_t) = P_0[1 + E(R_i)] = P_0(1 + r_f) + \frac{cov(P_0(1 + R_i), R_m)}{cov(R_m)}[E(R_m)) - r_f]$$

Thus,

$$P_0 = \frac{1}{1 + r_f}[E(P_t) - \frac{cov(P_t, R_m)[E(R_m) - r_f]}{cov(R_m)}]$$

4:

a)

$$cov(R_i, f_k) = cov(r_i + \beta_{i1}f_1 + \cdots + \beta_{ik}f_k + u_i, f_k) = \beta_{ik}cov(f_k)$$

$$\Rightarrow \quad \beta_k = \frac{cov(R_i, f_k)}{cov(f_k)}$$

b)

$$\begin{aligned} cov(R_i) &= cov(r_i + \beta_{i1}f_1 + \cdots + \beta_{ik}f_k + u_i) \\ &= cov(r_i) + cov(\beta_{i1}f_1) + \cdots + cov(\beta_{ik}f_k) + cov(u_i) \\ &= \beta_{i1}^2 cov(f_1) + \cdots + \beta_{ik}^2 cov(f_k) + \sigma_{u_i}^2 \end{aligned}$$

c)

$$\begin{aligned} cov(R_i, R_j) &= cov(r_i, r_j) + cov(\beta_{i1}f_1, \beta_{j1}f_1) + \cdots + cov(\beta_{ik}f_k, \beta_{jk}f_k) + cov(u_i, u_j) \\ &= \beta_{i1}\beta_{j1}cov(f_1) + \cdots + \beta_{ik}\beta_{jk}cov(f_k) \end{aligned}$$

5:

a)

$$R = r_f + 0.7F_1 + 0.5F_2 + \frac{1}{3}(\epsilon_1 + \epsilon_2 + \epsilon_3)$$

$$\Rightarrow \quad Cov(R) = 0.7^2 \times 0.09 + 0.5^2 \times 0.08 + \frac{1}{9}(0.03 + 0.02 + 0.01) = 0.0708$$

b)

$$Cov(R_1) = 0.8^2 \times 0.09 + 0.04^2 \times 0.08 + 0.03 = 0.1004$$

Thus the proportion of total risk of asset 1 is $\frac{0.8^2 \times 0.09}{0.1004} = 57.37\%$

c)

$$Cov(R_i) = \beta_i^2 Cov(F_1) + \gamma_i^2 Cov(F_2) + Cov(\epsilon_i)$$
$$Cov(R_i, R_j) = \beta_i \beta_j Cov(F_1) + \gamma_i \gamma_j Cov(F_2)$$

Thus, the covariance matrix of (R_1, R_2, R_2) is

$$\sum = \begin{bmatrix} cov(R_1) & cov(R_1, R_2) & cov(R_1, R_3) \\ cov(R_2, R_1) & cov(R_2) & cov(R_2, R_3) \\ cov(R_3, R_1) & cov(R_3, R_2) & cov(R_3) \end{bmatrix} = \begin{bmatrix} 0.1004 & 0.0664 & 0.0592 \\ 0.0664 & 0.0841 & 0.0618 \\ 0.0592 & 0.0618 & 0.0712 \end{bmatrix}$$