Lab 8. Testing CAPM

First, down load the data into Splus and start preprocessing: combining the adjusted column data into a matrix

```
> close <- cbind(sp500[1:180,7], ford[1:180,7], ge[1:180,7], jnj[1:180,7])
               #last 180 months for sp500 and 3 stocks
> dimnames(close) <- list(NULL, c("sp500", "ford", "ge", "jnj"))
               #give the names to its column
> close   <-  apply(close,2,rev) #reverse the time order for each column
> close    <- log(close)         #log prices and reverse the order
> tsplot(close)
> title("Monthly closing prices", "(a)")
> returns  <- apply(close,2,diff)#take diference for each column
> tsplot(returns)
> title("Returns of the stocks", "(b)")
> returns <- data.frame(returns)
               #change the object "returns" from a matrix to a data fram
> tb3m  <- tb3m[663:841,2]   #last 179 months yields of 3m T-bill
> tb3m
 [1] 8.82 8.65 8.43 8.15 7.88 7.90 7.75 7.64 7.69 7.63 7.64 7.74 7.90
 [14] 7.77 7.74 7.73 7.62 7.45 7.36 7.17 7.06 6.74 6.22 5.94 5.91 5.65
 [27] 5.46 5.57 5.58 5.33 5.22 4.99 4.56 4.07 3.80 3.84 4.04 3.75 3.63

> returns
               sp500         ford           ge           jnj
  [1,]  0.0488757511  0.032002731  0.095310180  0.0577083176
  [2,]  0.0345343446  0.015625318  0.114410351  0.0764416955
  [3,] -0.0079561892 -0.007782140  0.026668247 -0.0307716587
> tb3m <- tb3m/12     #convert to monthly return
> returns <- returns*100  #covert to percentage
> apply(returns, 2, mean)
>  apply(returns[, 2, mean)
     sp500     ford       ge      jnj
 0.7612089 1.374033 2.621718 1.834552
> returns <- returns - tb3m  #excessive returns
> apply(returns, 2, mean)
    sp500      ford       ge      jnj
 0.00769404 0.6205276 1.868227 1.081055

> rm(sp500,ford,ge,jnj) # remove unnecessary objects from Splus
```

After comging the excess returns above, we are ready to run the least-squares regression. We first run separately and get its residuals.

```
> y <- returns[60:179, 2:4]  #last 120 months e-return
> x <- returns[60:179,1]
>  ls.print(lsfit(x,y[,1]))   #  Ford
Residual Standard Error = 9.605,  Multiple R-Square = 0.247
N = 120,  F-statistic = 38.6999 on 1 and 118 df, p-value = 0

            coef std.err t.stat p.value
Intercept 0.2318  0.8804 0.2633  0.7928
        X 1.1967  0.1924 6.2209  0.0000
>  ls.print(lsfit(x,y[,2]))  #   GE
Residual Standard Error = 5.2554,  Multiple R-Square = 0.4659
N = 120,  F-statistic = 102.9137 on 1 and 118 df, p-value = 0

            coef std.err  t.stat p.value
Intercept 0.9435  0.4817  1.9588  0.0525
        X 1.0678  0.1053 10.1446  0.0000
>  ls.print(lsfit(x,y[,3]))   # Johnson and Johnson
Residual Standard Error = 6.1167,  Multiple R-Square = 0.1402
N = 120,  F-statistic = 19.2361 on 1 and 118 df, p-value = 0

            coef std.err t.stat p.value
Intercept 1.1368  0.5606 2.0277  0.0448
        X 0.5373  0.1225 4.3859  0.0000
> residuals <- resid(lsfit(x,y))
> residuals
            ford           ge           jnj
 60  -0.21715924   7.40057055  -4.764128271
 61   0.54244909  -7.17896355   7.215961721
 62  -2.68489561   2.32423552   7.665585324
 63   5.15326386   0.08377143  -3.024363183
 64   5.16727569   3.58930376   6.191137524
> Sigma <- t(residuals)%*%residuals/120
> alpha <- c(0.2318, 0.9435, 1.1368)
> mreturn <- mean(x)
> msigma <- var(x)
> T0     <-  120/(1+mreturn^2/msigma)* t(alpha) %*% solve(Sigma, alpha)
> 1 - pchisq(T0, 3)
[1] 0.04097483
```

```
> T1     <- (120-3-1)/3/120*T0
> 1 - pf(T1,3,120-3-1)
[1] 0.05140713
```