Empirical Study
on
10 IT Stocks in S&P 500

Edited by

Fan,Song 578532
Jinhua,Yang 578309
Wei,Zhang 578473
Yue,Wang 578706

1 Introduction

Based on what we have learned in SLP courses and our interest in financial market, especially the behavior of stock prices, we collected share price data on 10 internet and software stocks, and conducted our empirical studies. In our report, we write four programs, which can help us do statistical analysis of our data. We run panel regression, test for the CAPM theory, and estimate VaR of a price-weighted portfolio.

Asset pricing theory and tine series study of the movement of stock prices are two major focus of finance. Investors use asset pricing theory to search for stocks which are underpriced and make profits by trading these stocks. There are also investors who believe in technical analysis, and they try to figure out the patterns in the movement of stock prices. These people often bet on the trends or momentum they discovered and trade according to their designed strategies.

Stocks in the same industry may also have different performances, some stocks may have excess return compared to the market, while others are not. Assuming stocks in the same industry have the same reaction while market changes(same beta for all stocks), we apply panel regression to do this analysis. The basic model is pooled regression, which is to regress returns of all stocks on the market return and to see whether stocks this industry have excess return.

Considering that the companys own characteristics may influence its stock performance, we also conduct the random- and fixed-effect model to see the effect. As the classical asset pricing theory, CAPM is always of great interest of scholars. If CAPM is true, and the stock prices are correctly priced, the estimated intercept of CAPM should not be significantly different from zero. Using our data, we can run CAPM regressions and use GRS test to see whether the intercepts of our regressions are significantly different from zero or not.

Finally, we use DCC (Dynamic Conditional Correlation) model and calculated VaR for a price weighted portfolio constructed using these 10 stocks. And in order to test the robustness of our result, we apply DW and Engle's ARCH test to see whether the assumptions of our method are satisfied.

2 Theory and Design

2.1 Panel Data Analysis

We assume that the beta of all underlying stocks are the same, and the difference of returns are explained by individual effects, either fixed effects or random.

The basic panel data model (**?**) is a pooled regression, which is:

$$R_{it} = \alpha_i + \beta R_{mt} + e_{it} \tag{1}$$

In (1), R_{it} is the return of stock i at time t;α_i is the idiosyncratic risk premium, β is the part of risk premium that can be explained by the market factor, R_{mt} is the market risk premium at time t, and e_{it} is the residual.

As for the fixed effects model, in which the error term of pooled regression can be divided into two components, one of which is white noise and the other is the individual fixed-effects. The individual fixed-effect is represented by different intercept. This model can be expressed as follows:

$$R_{it} = \alpha_i + \beta R_{mt} + \mu_i + e_{it} \tag{2}$$

Where

$$\varepsilon_{it} = \mu_i + e_{it}, e_{it} \sim (0, \sigma_e^2) \tag{3}$$

e_{it} is a random variable with expectation of 0 and homogeneous variance, and μ_i is the individual fixed-effects. And $E(u_i|R_{mt}) \neq 0$.

When we using individual dummy variable to estimate the model, we simply use the OLS model:

$$R_{it} = com_1 + com_2 + com_3 + com_4 + com_5 + com_6 + com_7 + com_8 + com_9 + com_{10} + \beta R_m + e_{it} \tag{4}$$

Where $com_i = 1$ when the company is i and 0 otherwise and $e_{it} \sim (0, \sigma_e^2)$.
The random effect model is:

$$R_{it} = \alpha_i + \beta R_{mt} + \mu_i + e_{it} \tag{5}$$

Where

$$\varepsilon_{it} = \mu_i + e_{it}, e_{it} \sim (0, \sigma_e^2) \tag{6}$$

In this mode we assume mu_i is a random variable with expectations conditional on R_{it} equal to zero. Since the whole error term ϵ_{it} violates the assumption of OLS, we use GLS to estimate the random effect model.

2.2 Test of CAPM

However, in the test of CAPM, we drop the assumption that all stocks have the same beta, and assume they differ from each other. Firstly, we used the seemingly unrelated regression model to estimate β_i for each company. The model (**??**)can be expressed as follows:

$$Z_{it} = \alpha_i + \beta_i Z_{mt} + \epsilon_{it} \tag{7}$$

Where Z_{it} is the excess return of the stock of company i at time t, which can be calculated as follows:

$$Z_{it} = E(R_i) - R_f \tag{8}$$

At last, we use GRS test to test the null hypothesis that all of the intercepts are jointly equal to zero, i.e.:

$$H_0 : \alpha_1 = \alpha_2 = ... = \alpha_{10} = 0 \tag{9}$$

GRS test (**?**) statistic:

$$\frac{T(T-N-1)}{N(T-2)} \cdot \frac{\hat{\alpha}' \cdot \Sigma^{-1} \cdot \hat{\alpha}}{1+S_m^2} \overset{H_0}{\sim} F_{N,T-N-1} \tag{10}$$

Where S_m is the sharp ratio, which can be calculated as follows:

$$S_m = \frac{E(R_m - R_f)}{\sigma_m} \tag{11}$$

2.3 Estimation of Portfolio VaR

VaR can be computed according to the following formula:

$$VaR = W_0 \cdot (Z \cdot \sigma + \mu) \tag{12}$$

Where W_0 is the initial wealth, Z is the underlying confidence level, σ and μ are the variance and mean of related asset return respectively.

Once we know the variance of the investment asset and its mean, we can easily compute it VaR. In our analysis, we constructed a price-weighted portfolio using the 10 chosen stocks, which means the weight of every stock is proportional to its initial share price. To state differently, in our portfolio, we long one share of every selected stock.

We assume that every stock has autocorrelation and ARCH effect, the empirical variance of portfolio return was not proper to be used to compute VaR, therefore, estimating the variance-covariance matrix is necessary. There are two alternative ways to compute the variance and covariance matrix. The first option is multivariate GARCH models, it can capture volatility clustering and correlation of asset returns. However, multivariate GARCH model suffers from

huge parameter spaces. And traditional correlation models have problem with higher dimensional systems. As a substitute, we used dynamic conditional correlation (DCC) models, which is the trade-off between model feasibility and flexibility. The explicit model is as follows:

$$
\begin{aligned}
h_t &= \alpha + A\epsilon_{t-1}^2 + Bh_{t-1} \\
\epsilon_{i,t} &= h^{1/2}z_{i,t} \\
z_t &\sim ID(0, P_t)
\end{aligned}
\tag{13}
$$

DCC model (**?**) is as follows:

$$
\begin{aligned}
P_t &= (Q_T \bigodot I_N)^{-1/2} Q_T (Q_T \bigodot I_N)^{-1/2} \\
Q_T &= (1 - \alpha - \beta)Q + \alpha Z_{t-1}Z'_{t-1} + \beta Q_{t-1} \\
\alpha + \beta &< 1 \\
\alpha, \beta &> 0
\end{aligned}
\tag{14}
$$

Where Q is a sample covariance matrix of Z_t.

In the end, in order to test whether our time series fulfill the assumption that they have autocorrelation and ARCH effect, we simply used the build-in function of DW-test (**??**) and Engle's ARCH test (**?**).

3 Implementation

3.1 Data Description

Core part of our code:

```
1       summary(Paneldata$Ri)
2       summary(Paneldata$Rm)
3       stargazer(Paneldata)
4       p = ggplot(Paneldata, aes(x = Rm, y = Ri, colours =
            factor(company)))
5       p + geom_point(aes(colour = factor(company))) + stat_
            smooth()
6       p = ggplot(Paneldata, aes(x = Ri))
7       p + geom_histogram(aes(fill = factor(company), y = ..
            density..), alpha = 0.3, colour = "blue") + stat_
            density(geom = "line", position = "identity", size
            = 1.5, aes(colour = factor(company))) + facet_
            wrap(~company, ncol = 1)
8       plotmeans(Ri ~ company, main = "Heterogeineity across
            companies", data = Paneldata)
9       plotmeans(Ri ~ date, n.label = FALSE, main = "
            Heterogeneity across time", data = Paneldata)
```

Firstly we focus on the descriptive analysis of the panel data. We use 'summary()' function in line 1 and 2 and to conclude the descriptive analysis(mean, min, max and other statistics of our data) and use 'stargazer' package to make LaTex table as stated in line 3.

3.2 Panel Analysis

3.2.1 Pooled regression

Core part of our code:

```
1       pooling=plm(Ri~Rm,data=Paneldata,index=c("company","
            date"),model="pooling")
2       plot(Ri ~ Rm, data = Paneldata, main = "relationship
            between market return and stock return",  xlab = "
            market return", ylab = "stock return")
3       model = lm(Ri ~ Rm)
4       abline(model, col = "red")
```

In line 1 we use the 'plm' command to do the pooled regression. Then we use the 'plot' commad to plot out the distribution of the residuals of this regression

and use 'qqnorm' command in line 6 to analyze whether the normality assumption is satisfied in this model.

3.2.2   Fixed and Random Effect Model

Core part of our code:

```
1       xyplot(Ri ~ Rm | com, data = Paneldata, panel =
           function(x, y) {
2       panel.xyplot(x, y)
3       panel.abline(h = median(y), lty = 2, col = "gray")
4       panel.lmline(x, y, col = "red")
5       }, xlab = "market_return", ylab = "stock_return",
           main = "relationship_between_market_return_and_
           stock_return")
6       fixed.dum=lm(Ri~Rm+factor(com)-1,data=Paneldata)
7       yhat=fixed.dum$fitted.values
8       scatterplot(yhat ~ Rm | Paneldata$company, boxplots =
            FALSE, xlab = "Market_price", ylab = "yhat",
           legend.plot = FALSE,
9       main="Fixed_effects_regression_vs._Pooling_Regression
           ",
10      smooth = FALSE)
11      abline(lm(Ri ~ Rm), lwd = 3, col = "Dark_Blue")
12      fixed=plm(Ri~Rm,data=Paneldata,index=c("company","
           date"),model="within")
13      test_fixed = pFtest(fixed, pooling)
14      random=plm(Ri~Rm,data=Paneldata,index=c("company","
           date"),model="random")
15      phtest(fixed,random)
```

First of all, we use 'xyplot' command in line 1 to analyze graphically whether it is obvious that there are individual fixed effects.

Then we use 'lm' command in line 4 to do least squared dummy variable regression. And then from line 5, we analyze graphically the linear regression line for each company and the overall regression using command 'scatterplot'.

And we use 'plm' command to estimate the fixed-effect model and random-effect model by setting the 'model' variable equal to 'within' and 'random'. Then make comparison between pooled model and fixed model by 'pFtest'(F-test), fixed-effect model and random-effect model by 'phtest'(Hausemantest) in line 14 and 16 separately.

3.3 Test of CAPM

3.3.1 Seemingly Unrelated Regression for CAPM

Core part of our code:

```
1   Google = Capm$Google
2   Ebay = Capm$Ebay
3   Facebook = Capm$Facebook
4   Yahoo = Capm$Yahoo
5   WU = Capm$WU
6   Verisign = Capm$Verisign
7   Netflix = Capm$netflix
8   TSS = Capm$TSS
9   FNI = Capm$FNI
10  ADP = Capm$ADP
11  amer = Capm$amer
12  xyplot(aser ~ amer | com, data = Paneldata, panel =
        function(x, y) {
13  panel.xyplot(x, y)
14  panel.abline(h = median(y), lty = 2, col = "gray")
15  panel.lmline(x, y, col = "red")
16  }, xlab = "average_excess_market_return", ylab = "average
        _excess_stock_return", xlim = c(0, 2), ylim = c(0,5),
        main = "Excess_market_return_and_excess_stock_return")
17
18  eq1=Google~amer
19  eq2=Ebay~amer
20  eq3=Facebook~amer
21  eq4=Yahoo~amer
22  eq5=WU~amer
23  eq6=Verisign~amer
24  eq8=TSS~amer
25  eq9=FNI~amer
26  eq10=ADP~amer
27  system=list(eq1=eq1,eq2=eq2,eq3=eq3,eq4=eq4,eq5=eq5,eq6=
        eq6,eq8=eq8,eq9=eq9,eq10=eq10)
28  sur=systemfit(system,method="SUR",data=Capm)
29  summary(sur)
```

From line 1 to line 10, we name the average excess stock return of different companies with the company initials. Line 11 we denote the average market excess return with 'amer'.

In line 12, we use 'xyplot' command to plot out the scatterplot graph of the relationship between average excess stock return and average market return for the 10 companies and add the regression lines to see whether the intercepts of

8

the regression lines are close to 0.

From line 13 to line 26, we define the 9 regression models of the excess return of stocks for each companies on the average excess return of market portfolio. And in line 27, we use 'list' command to define the system of equations and then use 'systemfit' command to estimate the SUR model.

3.3.2   GRS test for CAPM

Core part of our code:

```
1   Capm1 = read.csv("Capm1.csv")
2   sur1 = matrix(sur$coefficients, nrow = 2, ncol = 9)
3   alphas = sur1[1, ]
4   sig.hat=sur$residCov
5   sample.sr = function(x) {mu = mean(x);sg = sd(x);return(
        mu / sg)}
6   amer=Capm1$amer
7   mkt.sr=sample.sr(amer)
8   GRS.stat=t(alphas)%*%solve(sig.hat,(alphas))/(1+mkt.sr^2)
9   T=52
10  N=9
11  F.stat=T*(T−N−1)*GRS.stat/(N*(T−2))
12  p.val=pf(F.stat,N,T−N−1,0,lower.tail = F)
```

Since there is no GRS test package in R yet, we use function command to calculate the statistics and calculate the p-value. Firstly, in line 3 and line 4 we define alphas as the intercepts and 'sig.hat' as the residual covariance matrix from the SUR regression we run previously. After that, we define a function to calculate the sample sharp ratio as stated in line 5. Then we calculate the market sharp ratio use this function in line 7. Finally we define the GRS statistics based on its formula as in line 8 and calculate the corresponding F statistics and the p-value.

3.4   Estimation of Portfolio VaR

3.4.1   VaR Estimation

Core part of our code:

```
1   return = log(price[2, ]/price[1, ])
2   for (j in 2:(T − 1)) {
3   return[j, ] = log(price[j + 1, ]/price[j, ])
4   }
5   for (j in 1:(T − t)) {ts = return[j:(j + t − 1), ]
```

```
 6  for (i in 1:n) {residual[, i] = matrix(residuals(arma(ts
        [, i], order = c(1, 0))))}
 7
 8  for (i in 1:10) {coef[i, ] = matrix(coef(garch(residual[,
        i], order = c(1, 1), series = NULL)))}
 9
10  # DCC–GARCH model estimation
11  a = coef[, 1]
12  A = diag(coef[, 2])
13  B = diag(coef[, 3])
14  dcc.para = c(0.01, 0.97)
15  results = dcc.estimation(inia = a, iniA = A, iniB = B,
        ini.dcc = dcc.para, dvar = residual, model =          "
        diagonal")
16  h = results$h
17  dcc = results$DCC
18  v = sqrt(diag(h[t − 1, ]))
19  R = matrix(data = dcc[1, ], nrow = n, ncol = n)
20  H = v %*% R %*% v
21
22  VaR95[j, ] = sqrt(t(W) %*% H %*% W) * alpha95 * value[j]
23  VaR99[j, ] = sqrt(t(W) %*% H %*% W) * alpha99 * value[j]}
24  PL = value[(t + 1):T, ] − value[t:(T − 1), ]
25  matplot(c(1:(T − t)), bt[, 1:3], type = "l", xlab = "time
        ", ylab = "P&L", xaxt = "n")
26  colnames(bt) = c("PL", "95%VaR", "99%VaR")
27  legend("topright", colnames(bt)[−1], lwd = 1, col = 2:3,
        cex = 0.75)
28  title("Portfolio P&L and estimated VaR")
29  axis(1, at = c(1:(T − t)), las = 0)
```

In order to get the daily log return of each stock, we use line 1 to initialize the return variable and use 'for' command in line 2 to apply the log return formula for every row regardless the columns, thus we can get the daily return for every stock. In line 3, when we fix the length of our rolling screen 't', the number of VaR we estimate for our portfolio will be T-t. Thus we run j from 1 to T-t, to get every estimation of VaR. And for every VaR, we use the stick returns from j to j+t-1 of each stock, thus each stock has t observations for parameter estimation. In line 4, in order to get arma residual for every stock in each rolling screen, we use 'for' command running for each stock and use 'arma' function with order (1,0) with our model being AR(1).

From the above process, we get n*3 coefficients in total, which is the input for DCC model. Through DCC model in line 11, we get the variance as in line 12, correlation coefficients as in line 13. Use the variance of the last day, and compute them with the correlation coefficients and we can get variance-

covariance matrix. Together with the weights of portfolios, we finally obtain the variance of our portfolio, which is the critical input for our estimation of VaR. This whole process is repeated for T-t times, where T is the number of days, t is the number of days in the chosen rolling screen.

Using the estimated coefficient from initial GARCH model, residuals from AR(1) model, together with other initial parameters we use 'dcc.estimation' function and the variance and correlation coefficient can be estimated. 'v' in line 18 is the square root of variance, 'R' in line 19 is the correlation coefficient, then by line 20 we can get the variance of our price weighted portfolio. In line 22 and 23 the 95% and 99% VaRs are estimated.

In line 24, we computed the daily profit and loss of each stock. 'Matplot' function in line 25 is a function for matrix plot, here we use this to plot 3 lines (P&L,VaR95, and VaR99) together at once. And in line 29, we use 'axis' function to create new format for x-axis, and 'at' indicates the tic marks we drawn are related to the number of rolling screen, which is also the number of VaRs we estimate.

3.4.2 DCC Model Assumption

Core part of our code:

```
1  for (i in 1:10) {
2  dw = dwtest(return[2:247, i] ~ return[1:246, i] pvalue_
       dwtest[, i] = dw$p.valuedw_stat[, i] =        dw$
       statistic
3  }
4  pvalue_dwtest = as.data.frame(pvalue_dwtest)
5  dw_stat = as.data.frame(dw_stat)
6
7  return1 = as.list(return)
8  archtest = lapply(return1, ArchTest, lags = 8, demean =
       TRUE)
9  a = as.data.frame(matrix((unlist(archtest)), c(5, 10)))
10 LM_test_ARCH = rbind(a[1, ], a[3, ])
```

To test the two assumptions for DCC model, we used 'lmtest' and 'FintTS' package. From line 1 to 3, we conducted a DW test, and in line 8 we conducted the Engle's ARCH test for all of our time series. In line 1, we used 'for' command to repeat the DW test for our 10 selected stocks, can get the DW statistics and corresponding p value. In line 8, we use 'lapply' function to apply 'ArchTest' to all columns and obtained the corresponding statistics and p values.

4 Empirical Study/Testing

4.1 Data Input

We collect daily stock price and S&P 500 index from http://finance.yahoo.com/ for the following 10 stocks:

Name of the company	Ticker
Google	GOOGL
Ebay	EBAY
Facebook	FB
Yahoo	YHOO
Wester Union	WU
Verisign Inc	VRSN
Netflix Inc.	NFLX
Total System Service	TSS
Fidelity National Information Services	FIS
Automatic Data Processing	ADP

Table 1: Stock List

And we use the Treasury bill rate of US as the risk free rate, which is available at https://www.treasury.gov/resource-center/data-chart-center/interes-trates/Pages/TextView.aspx?data=yield.

The time period for these data is 01/05/2015-25/04/2016. For panel analysis we use weekly return of these stocks, for test of CAPM we use three-month return of these stocks, for the estimation of VaR we use daily return of these stocks.// The raw data we obtain directly from this resources are prices (daily or weekly), and we need to pre-process them by applying the function

$$R_{t+1} = \ln \frac{P_{t+1}}{P_t}$$

to get the return of individual stock and the market.

4.2 Pooled Regression Model

- **Function**

```
1    pooling = plm(Ri ~ Rm, data = Paneldata, index = c("
        company", "date"), model = "pooling")
```

- **Input Description**

Use the data source as stated above, in which Ri is the individual stock return while Rm is the market return. This is panel data, in which 'company' is the 'individual' and 'date' stands for 'time'. Hence the 'index' in the function is to tell R which is individual and which is time.

Besides, we denote the ten companies as the number 1 to 10. For each company there are 52 time periods and corresponding R_i. There are also 52 periods for R_m and for each company R_m is the same.

For pooled regression model, we pool the data together and do OLS.

- **Expected Output**

We expect a positive relation between market portfolio return and the stock return overall.

- **Actual Output**

	Dependent variable:
	Ri
Rm	1.235***
	(0.071)
Constant	0.003**
	(0.001)
Observations	520
R^2	0.368
Adjusted R^2	0.366
F Statistic	301.162***

Table 2: Pooled Regression Results

- **Comments**

The market portfolio return and stock return are actually positive correlated. And the coefficient is significant under the significance level of 1%. According to the F statistic, the model is significant.

13

4.3   Fixed Effect Regression Model

- **Function**

```
1  fixed = plm(Ri ~ Rm, data = Paneldata, index = c("company
     ", "date"), model = "within")
```

- **Input Description**

We employ the panel data as we stated in the pooled regression model. Including fixed effects across companies, we use within estimator approach, which is that we subtract Ri, Rm and also the error term with their corresponding average, and then do the OLS regression. Here for the 'model' part we input 'within'.

- **Expected Output**

According to graphical analysis, we figured that the fixed effects of different companies is not very obvious.

- **Actual Output**

	Dependent variable:
	Ri
Rm	1.235***
	(0.071)
Observations	520
R^2	0.370
Adjusted R^2	0.362
F Statistic	298.774***

Table 3: Fixed Effects Regression Using plm

- **Comments**

The coefficient value and the variance of the coefficient are the same with pooled regression model. It is likely that there is no significant difference between fixed effects model and pooled regression model.

4.4 Testing for Fixed Effects

- **Function**

```
1  test_fixed = pFtest(fixed, pool)
```

- **Input Description**

After finishing the regressions of fixed effects model and pooled model, we use the F-test to test whether the fixed effects across companies are significant. Note that this step is based on the fact that we have already finished the two regressions. Then we use the denoted results, here 'fixed' and 'pool', from the two regressions to do the F-test.

- **Expected Output**

As stated in the last part, we expect no significant fixed effects between companies.

- **Actual Output**

F test for individual effects
data: Ri~Rm
F = 0.54364, df1 = 9, df2 = 509, p-value = 0.8427
Alternative hypothesis: significant effects.

- **Comments**

p-value is much larger than 5%, which means that there is no evidence proving that the Fixed Effects Model is better than Pooled Regression Model.

4.5 Random Effect Regression Model

- **Function**

```
1  random = plm(Ri ~ Rm, data = Paneldata, index = c("
      company", "date"), model = "random")
```

- **Input Description**

Again we use the panel data described in pooled regression model. Here for the 'model' part we input 'random'. The underlying approach used in this function is actually GLS estimation owing to the structure of the error term in random effects model.

- **Expected Output**

Since we expect that the effects is related to the regressor, that is, the important assumption for random effects model,$E(u_i|R_{mt}) \neq 0$, is not satisfied. Therefore we expect random effects model should generate the same outcome as pooled effects model.

- **Actual Output**

	Dependent variable:
	Ri
Rm	1.235***
	(0.071)
Constant	0.003***
	(0.001)
Observations	520
R^2	0.366
Adjusted R^2	0.365
F Statistic	299.657***

Table 4: Random Effects Regression

- **Comments**

Just as we expected, the random effects regression result is almost the same with pooled regression except for a slightly lower R-squared and F-statistic.

4.6 Testing for the Fixed Effects Model and Random Effects Model

- **Function**

```
1  phtest(fixed , random)
```

- **Input Description**

After finishing both fixed effects model regression and random effects model regression, we denote the two models as 'fixed' and 'random'. And then use Hausman test to test whether within estimator or GLS estimator is consistent.

- **Expected Output**

We expect there is random effects model is inconsistent, that is, we expect that we would reject the null hypothesis.

- **Actual Output**

Hausman Test
data: $Ri \sim Rm$
chisq = 4.7193e-25, df = 1, p-value = 1
alternative hypothesis: one model is inconsistent

- **Comments**

We cannot reject null hypothesis, which is that the preferred model is random effects. Thus comparing with Fixed Effects Model, Random Effects Model is better.

4.7   Comparison of Random Effects Model and Pooled Regression Model

- **Function**

```
1        plmtest ( pooling )
```

- **Input Description**

Use the result of the random effects model and pooled regression model.

- **Expected Output**

We expect that we cannot reject the null hypothesis since we expect that the assumption $E(u_i|R_{mt}) \neq 0$ may not be satisfied.

- **Actual Output**

Lagrange Multiplier Test - (Honda)
data: $Ri\tilde{R}m$
normal = -1.14, p-value = 0.2543
Alternative hypothesis: significant effects

- **Comments**

P-value is too high. We cannot reject the null hypothesis. It is better to use Pooled Regression Model.

4.8   CAPM testing-SUR and GRS test

- **Function**

```
1  sur=systemfit(system,method="SUR",data=Capm)
2  sur1=matrix(sur$coefficients,nrow=2,ncol=9)
3  alphas=sur1[1,]
4  sig.hat=sur$residCov
5  sample.sr = function(x) {mu = mean(x);sg = sd(x);return(
      mu / sg)}
6  amer=Capm1$amer
7  mkt.sr=sample.sr(amer)
8  GRS.stat=t(alphas)%*%solve(sig.hat,(alphas))/(1+mkt.sr^2)
9  T=nrow(Capm1)
10 N=ncol(Capm1)−3
11 F.stat=T*(T−N−1)*GRS.stat/(N*(T−2))
12 p.val=pf(F.stat,N,T−N−1,0,lower.tail = F)
```

- **Input Description**

We use the 3-months Treasury bill yield as the risk-free interest rate and calculate out the averaged market excess return and excess stock return (both for 3 months). We put the adjusted data to a file 'Capm', and we execute the SUR and GRS test.

First we should take out all the intercepts of the seemly uncorrelated regressions, which is done by the 2nd and 3rd lines of the functions. If the input of the functions is wrong, then the α we get would be wrong.

Then we should get the variance-covariance matrix of the error terms of the SUR, which is done by the 4th line. And from 5th to 7th lines we get the sample market sharp ratio. The function of the 5th line is merely the sharp ratio function. And in the 6th and 7th lines we insert the averaged market excess return into this function and calculate the sample market sharp ratio. If the function is wrong or what we input is not averaged market excess return, then the result we get would be wrong.
The 8th line is to calculate the GRS statistic. We input the alpha, variance-covariance matrix as well as the sample market sharp ratio into this function and get the GRS statistic. And then based on this, the 9th line get the F statistic and thus the p-value. Any incorrect input for these functions would lead to a wrong output (econometrically speaking).

- **Expected Output**

We expect that we can reject the null hypothesis that all of the intercepts of the SUR are jointly zero

- **Actual Output**

P-value= 0.002696034

- **Comments**

We can reject the null hypothesis that the intercepts are 0. However, this does not necessarily mean that CAPM is a wrong model.

4.9   Estimation of VaR

- **Input Description**

In this sub-program, raw data of daily prices can be directly used as the input. After computing the daily return, we constructed a price weighted portfolio, and then use the return of this portfolio to calculated our 95% and 99% VaR. For a normal case input, we choose rolling screen size t=200.

- **Expected Output**

Through our programming, we expect to compute T-t, which is 248-200=48 95% and 48 99%VaRs, they should be parallel to each other and the 99% VaR has larger absolute value than the 95% VaR.

- **Actual Output**

Portfolio P&L and estimated VaR
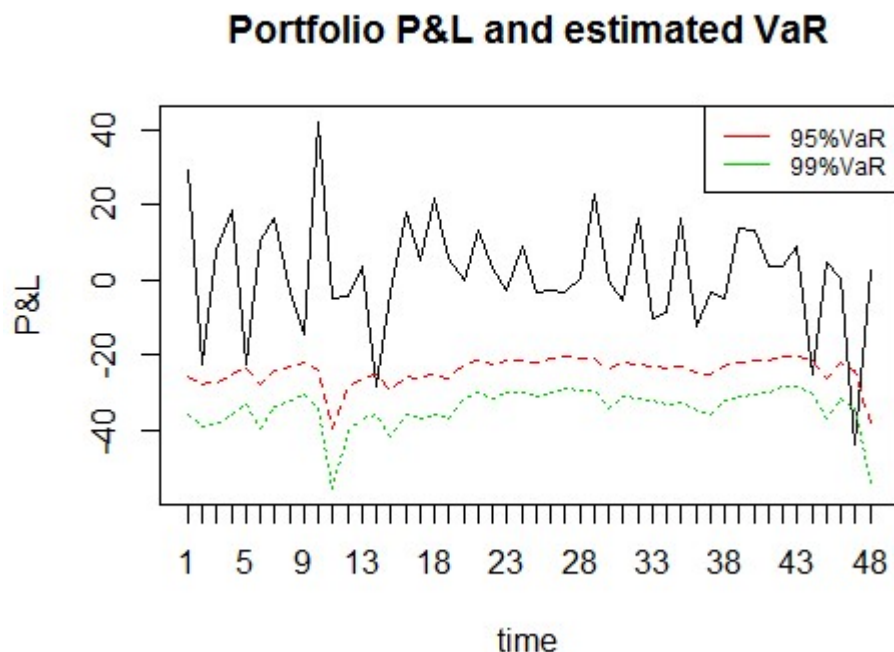
- **Comments**

It is clear that given related data, our program can effectively calculate the 95% and 99% VaR of price weighted portfolio constructed using selected stocks. There are only 3 outliers given 95%VaR estimated, and only 1 outlier given 99%VaR estimated. In the programming, we use DCC GARCH model, which is a generalization of the CCC GARCH model and allows the correlation matrix to depend on time. The DCC GARCH model has clear computational advantages in that the number of parameters to be estimated in the correlation process is independent of the number of series to be correlated. Thus potentially very large correlation matrices can be estimated.

- **Test for Extreme Case t=247**

Expected output: 3 dots
Actual output: no output.
Reason: Due to limitation of 'matrix plot function', it cannot produce any output graph when there is only one row of observation.

- **Test for Special Case t=246**

Expected output: 3 straight lines. When t=246, there are 2 estimations for each portfolio VaR, and with two dots, a straight line can be defined.
Actual output: good output.

Portfolio P&L and estimated VaR
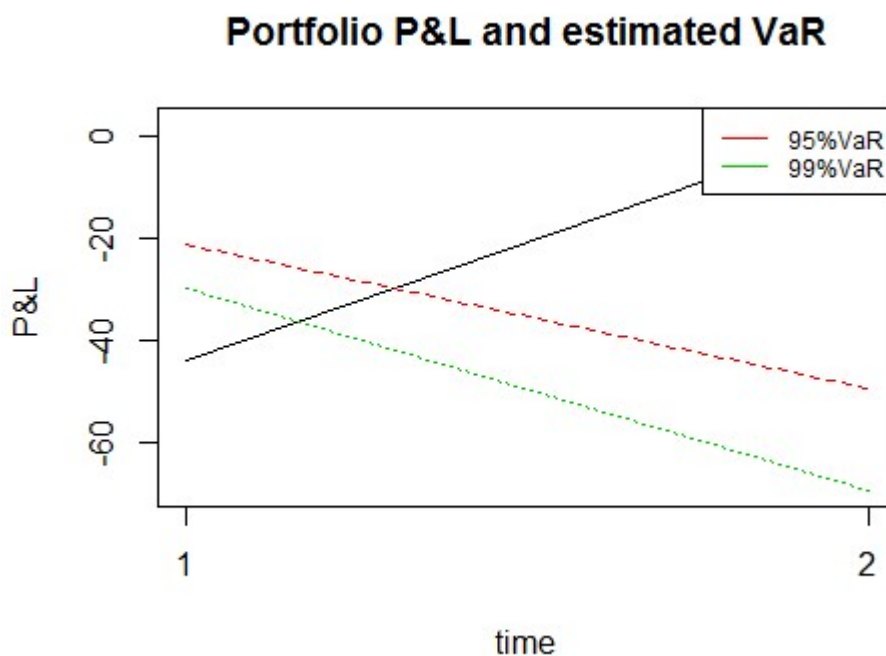
Comments: matrix plot can be implemented with 2 rows.

4.10   DCC Assumptions Test

- **Input Description**

For this sub-program, raw data of daily stock prices can also be directly used as input.

- **Expected Output**

Since our calculation of VaR using DCC model is based on the critical assumptions that the time series have autocorrelation and ARCH effect, we expect the test results to be significantly different from zero, with P-values being small enough (at least ¡10%). Otherwise, our calculation result would not make sense.

- **Actual Output**

	GOOGL	EBAY	FB	YHOO	WU
dw_stat	1.98	1.99	1.99	2.00	1.99
p_value	0.44	0.47	0.48	0.52	0.49

	VRSN	NFLX	TSS	FIS	ADP
dw_stat	1.98	1.99	1.98	2.02	1.99
p_value	0.44	0.47	0.46	0.57	0.49

Table 5: DW-Test

	GOOGL	EBAY	FB	YHOO	WU
LM_stat	1.33	0.42	8.18	5.12	1.19
p_value	0.99	0.99	0.41	0.74	0.99

	VRSN	NFLX	TSS	FIS	ADP
LM_stat	10.21	2.06	2.96	0.92	39.31
P_value	0.25	0.97	0.93	0.99	4.30E-06

Table 6: Engle's ARCH Test

- **Comments**

From the above results, we can see that except for the Engle's-ARCH test for stock of ADP, in both tests, p-values for other stocks are very large, which indicates that the assumptions of DCC model are not fulfilled. And our model of VaR calculation has great limitations, which needs to be improved. But considering that the time span of our data is limited and that we only select stocks from IT industry, the robustness of our model can be improved by including stocks from more industries and adding more observations.

5 Conclusions

To begin with, according to panel data analysis, we can draw the conclusion that the expected returns of stock returns are positively related with the market return. To be exact, using the result of pooled regression model, the share price of IT stocks moves in the same direction as the market. And by test result, we find that the pooled regression is more suitable here using our data set.

To the extent that we only want to get the regression and test results, our programming is already satisfying. However, it would be better that we improve our programming so that it can be more automatic to select a fittest model for a given data set.

As for the test of CAPM, we find that for these 10 companies, using SUR, we reject the null hypothesis that the intercepts are jointly 0, which is in line with our expectation. Because for a single stock, there is a large portion of individual risk. While for CAPM theory, only return of well diversified portfolio has the relationship with the market portfolio. But our program is useful in testing that theory given a constructed well diversified portfolio. Although empirical tests for CAPM are not always successful even with well diversified portfolio, that does not necessarily mean that CAPM is a wrong model for the following reason:

Firstly, there are some technical explanations such as Rolls critique. Roll proves that any mean-variance efficient portfolio satisfies the CAPM equation exactly. This statement is a mathematical fact, requiring no model assumption. Given a proxy for the market portfolio, testing the CAPM equation is equivalent to testing mean-variance efficiency of the portfolio. The CAPM is tautological if the market is assumed to be mean-variance efficient. In other words, if we reject the hypothesis that the intercepts are 0, that may only because they are not mean variance efficient. Additionally, the market portfolio is unobservable. The market portfolio in practice would necessarily include every single possible available asset, including real estate, precious metal and so on. Therefore, merely use the S&P 500 index is probably not enough.

Secondly, CAPM suffers from omitted variables. Here we only consider the variable of expected market return. But there may be time-varying effects included in the error term. Omitting these variables would cause a bias in the coefficient and also the variances and thus our inference becomes invalid.

Thirdly, irrational investors. One of the most important key assumption in most of the finance models is the rational investors. Once the assumption is not valid, for example, investors overreact to some news about certain companies, then the whole model may not be valid anymore.

As for the computation of VaR, we used multivariate GARCH model, explic-

itly DCC GARCH model, instead of multivariate GARCH model or traditional correlation model. The reasons are as follows:

In financial econometrics and management understanding, predicting the dependence in the co-movements of asset returns is important. For example, asset pricing depends on the covariance of the assets in a portfolio. Hence it is important to consider the co-movements in the portfolio.
Financial volatilities move together more or less closely over time across assets and markets.Recognizing this feature through a dynamic model should lead to more relevant empirical models than working with static traditional correlation models.In financial applications, extending from static to dynamic modelling opens the door to better decision tools in various areas such as asset pricing models, portfolio selection, hedging, and Value-at-Risk forecasts.

After the above considerations, we can say in confidence that the model we choose is suitable in theory. However, there are assumptions in our model. To see the effectiveness of the model applying to our data, we then tested whether the assumptions are satisfied. Using established packages, we implemented DW and Engle's ARCH test to see if there is autocorrelation and ARCH effect in our time series for every stock. Unfortunately, the test statistics are not significant, which indicates that the assumptions for our model are not met. The possible explanation might be the limitation of our observations or selected stocks coming from the same industry. Taking these possibilities into consideration, improvement can be made if we enlarge the time span of observations, including more stocks from other industries, or change the length of rolling screen.

To sum up, through this empirical study of IT stocks, we gained a better understanding of the application of R programming, and its usefulness in both academic and practical works. Nevertheless, there are much more to be explored in it, and we will try to gain more expertise in this programming language.

References

Baltagi, B. (2008). *Econometric analysis of panel data.* John Wiley & Sons.

Durbin, J. and Watson, G. S. (1950). Testing for serial correlation in least squares regression: I. *Biometrika*, 37(3/4):409–428.

Durbin, J. and Watson, G. S. (1951). Testing for serial correlation in least squares regression. ii. *Biometrika*, 38(1/2):159–177.

Engle, R. (2002). Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business & Economic Statistics*, 20(3):339–350.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007.

Gibbons, M. R., Ross, S. A., and Shanken, J. (1989). A test of the efficiency of a given portfolio. *Econometrica: Journal of the Econometric Society*, pages 1121–1152.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, 19(3):425–442.

Sharpe, W. F. (1970). *Portfolio theory and capital markets.* McGraw-Hill College.

Appendix

Code: https://github.com/JinhuaY/SPL_SP500int.