Factor Models

This is sourced from STAD70 course practice questions and sample R code taught by professor Sotos. If you have any questions/concerns/comments feel free to email me: cristal.wang111@gmail.com (mailto:cristal.wang111@gmail.com).

1.Regression - Mutual Funds Sector Attribution and Performance Analysis

For this problem you will use regression to identify the composition of various mutual funds.

1.1.Data and Return for Mutual Funds

Download the adjusted daily closing prices from Jan 1 2020 to Dec 31 2022 for the 5 mutual funds below (use `tseries::get.hist.quote()` for each ticker):

- FCNTX: Fidelity Contrafund
- PIODX: Pioneer A
- AIVSX: American Funds Invmt Co of Amer A
- PRBLX: Parnassus Core Equity Investor
- VFIAX: Vanguard 500 Index Admiral

Note that each of these funds has at least 90% of their weight in the US stocks market. You can actually check the composition of the investment over different stock sectors from Yahoo Finance, under the fund's holdings tab; e.g. for FCNTX at https://finance.yahoo.com/quote/FCNTX/holdings (https://finance.yahoo.com/quote/FCNTX/holdings).

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
MF.names=c('FCNTX','PIODX','AIVSX','PRBLX','VFIAX')
N.MF=length(MF.names)
S=list()
for(i in 1:N.MF){
  S[[i]]=get.hist.quote(MF.names[i], start='2022-01-01', end='2022-12-31', quote='AdjClos
e', quiet = TRUE)
}
R=lapply(S, FUN = function(x){ diff(x) /lag(x,-1) }) # MF net returns
logReturn = lapply(S, FUN = function(x){ diff(log(x)) })
RY=matrix(unlist(R),ncol=N.MF) # bind returns in a matrix
colnames(RY)=MF.names
```

- **`lapply(...)`** : Applies the function to each time series of prices in list **`S`** .
- **`diff(x)`** : Calculates the difference between consecutive days (i.e., P_t - P_{t-1}).
- **`lag(x, -1)`** : Shifts the prices forward by one time unit, i.e. P_{t-1}
- MF net returns: $Return_t = \frac{P_t - P_{t-1}}{P_{t-1}}$
- `diff(log(x))` : Computes the difference of log prices, which gives you:

  $$LogReturn_t = r_t = log(P_t) - log(P_{t-1}) = \log\left(\frac{P_t}{P_{t-1}}\right) = log(1 + R_t)$$

1.2.Regression - Sector Attribution on Sector ETFs

Assume you do not have any information about the investment strategy of the funds. Download the daily prices and calculate returns of the following EFTs, which track different sectors of the economy:

- XLB: Basic Materials | XLY: Consumer Cyclical | XLF: Financial Services | VNQ: Real Estate |

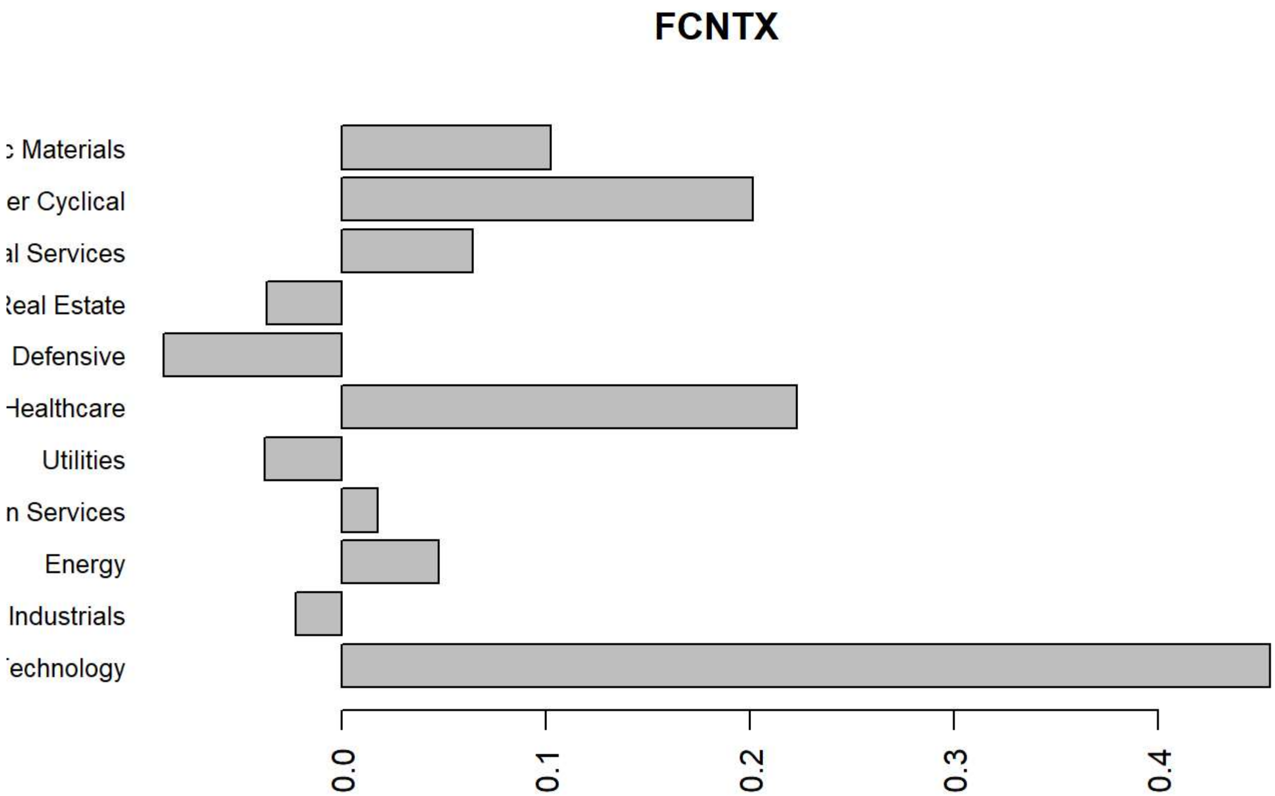
  XLP: Consumer Defensive | XLV: Healthcare | XLU: Utilities | XTL: Communication Services |

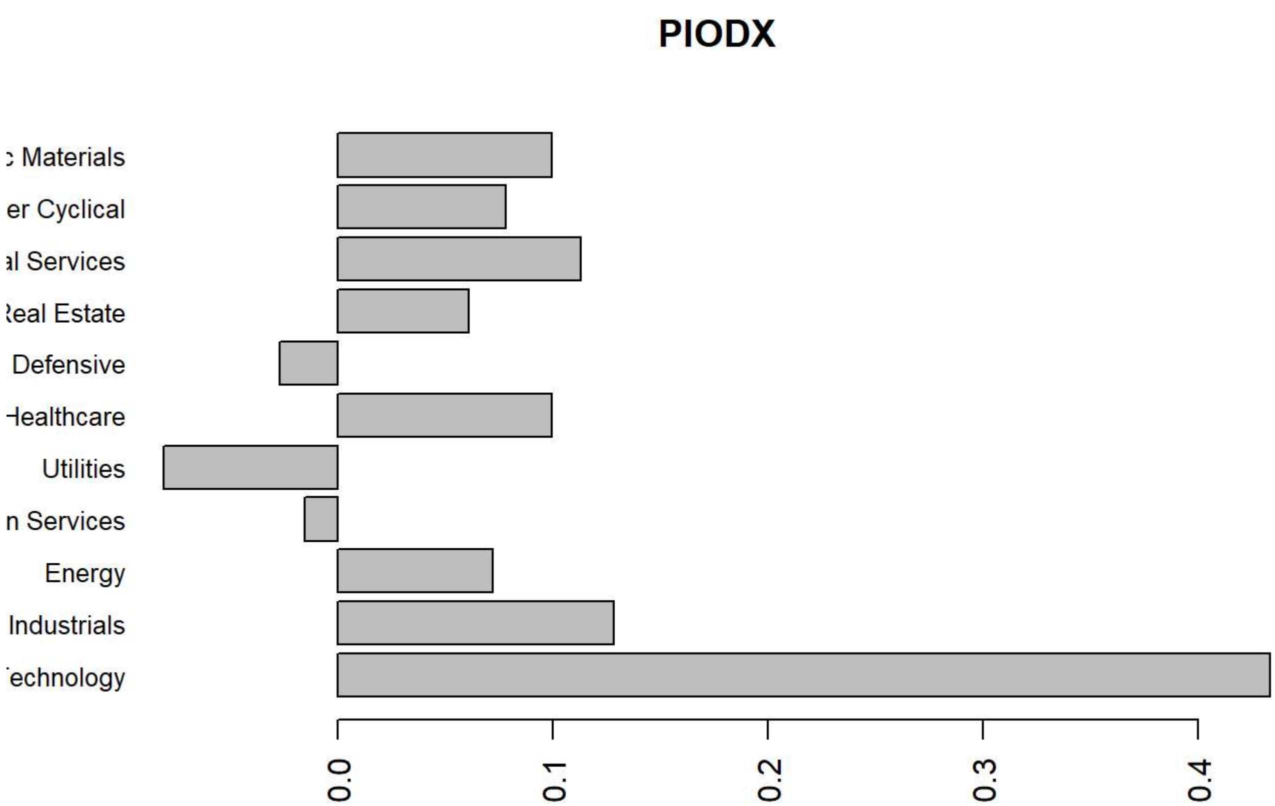
  XLE: Energy | XLI: Industrials | XLK: Technology

Regress each of the mutual fund returns on the above ETF returns and create barplots of the estimated beta coefficients. Do these accurately reflect the allocation over the different sectors (as described in Yahoo Finance)?

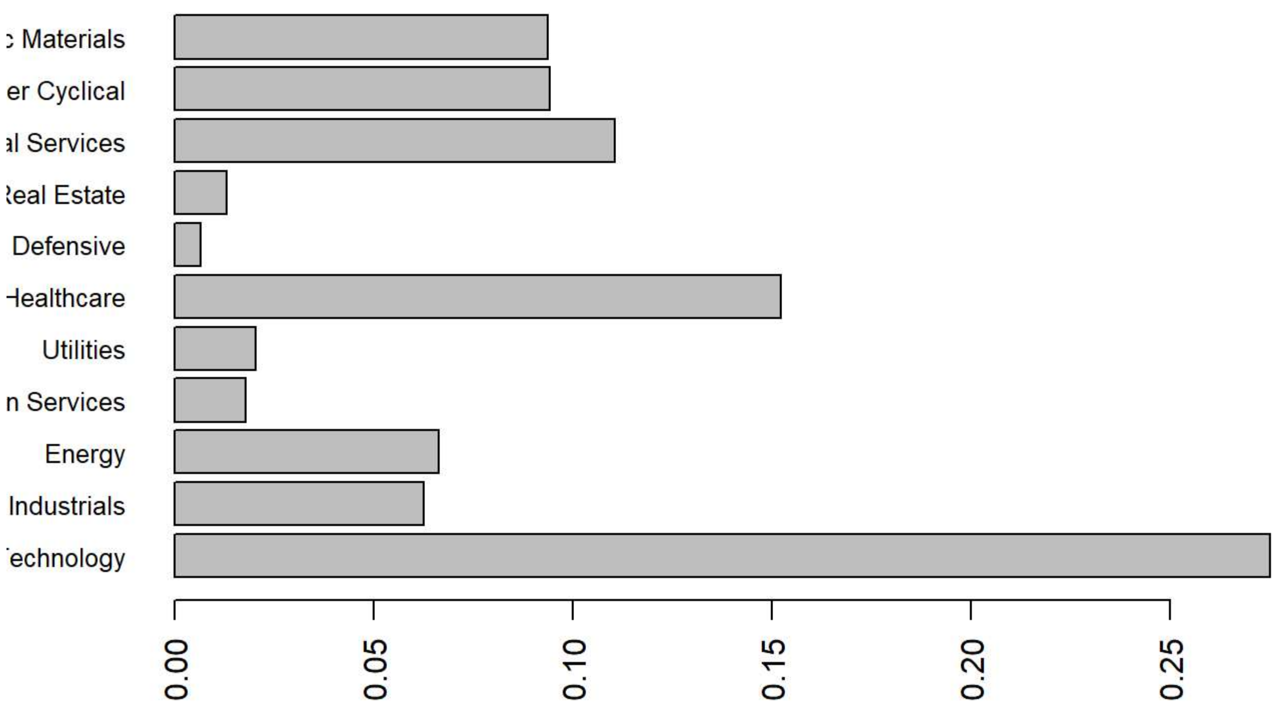
```
ETF.names=c('XLB','XLY','XLF','VNQ','XLP','XLV','XLU','XTL','XLE','XLI','XLK')
sectors = c("Basic Materials","Consumer Cyclical","Financial Services",
            "Real Estate","Consumer Defensive", "Healthcare", "Utilities",
            "Communication Services", "Energy", "Industrials", "Technology")
N.ETF=length(ETF.names)
S=list()
for(i in 1:N.ETF){
  S[[i]]=get.hist.quote(ETF.names[i], start='2022-01-01',                        end='2022
-12-31', quote='AdjClose', quiet = TRUE)
}
R=lapply(S, FUN = function(x){ diff(x) / lag(x,-1) }) # calculate ETF returns
logReturn = lapply(S, FUN = function(x){ diff(log(x)) })
RX=matrix(unlist(R),ncol=N.ETF) # bind returns in a matrix
colnames(RX)=ETF.names

out=list();
for(i in 1:N.MF){
  out[[i]]=lm(RY[,i] ~ RX)
  weights = out[[i]]$coef[-1]
  barplot( rev(weights), names.arg = rev(sectors), main = MF.names[i],
          horiz = TRUE, las = 2, cex.names = .8)
}
```

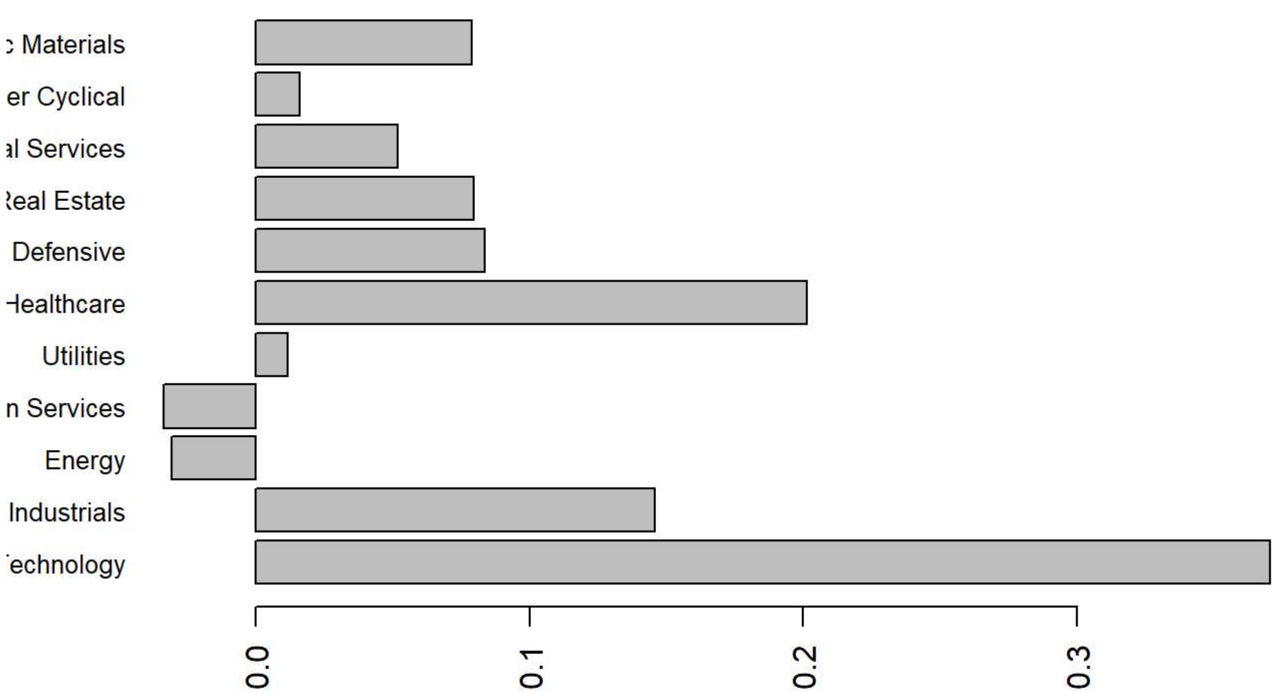

FCNTX
: Materials
er Cyclical
al Services
Real Estate
Defensive
Healthcare
Utilities
n Services
Energy
Industrials
Technology
0.0
0.1
0.2
0.3
0.4

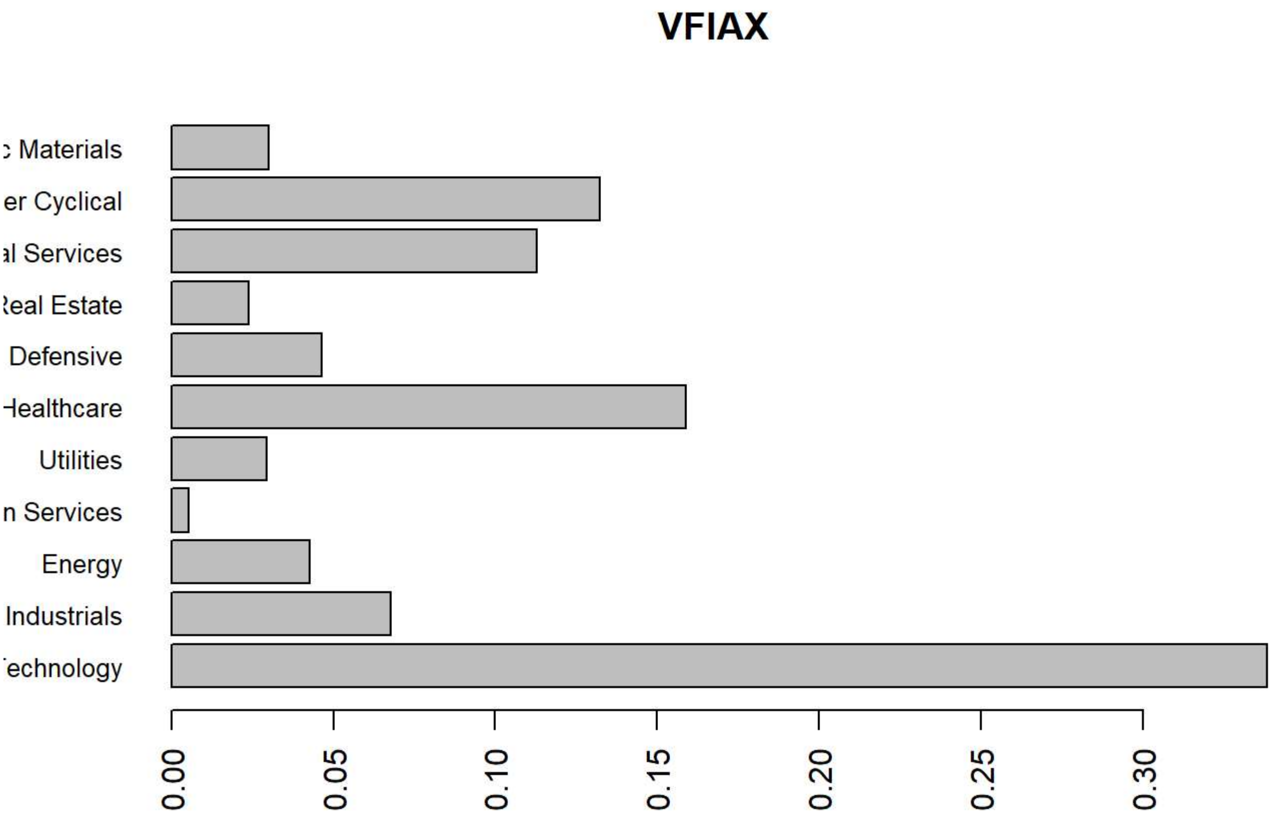

PIODX
: Materials
er Cyclical
al Services
Real Estate
Defensive
Healthcare
Utilities
n Services
Energy
Industrials
Technology
0.0
0.1
0.2
0.3
0.4

AIVSX


c Materials
er Cyclical
al Services
Real Estate
Defensive
Healthcare
Utilities
n Services
Energy
Industrials
Technology
0.00
0.05
0.10
0.15
0.20
0.25

PRBLX


c Materials
er Cyclical
al Services
Real Estate
Defensive
Healthcare
Utilities
n Services
Energy
Industrials
Technology
0.0
0.1
0.2
0.3



The barplots of the regression coefficients (betas) **roughly** follow the sector weightings for each fund. Nevertheless, they are not always close in actual value (e.g. in some cases the betas are negative, even though weightings are positive). The differences can be due to the fact that we use ETFs as *proxies* for a sector, but the actual holding of the fund within the sector might be different. Moreover, there will be estimation error in our regression model, which is only based on the last year's returns.

Nevertheless, it is quite impressing that we can (approximately) identify the strategy of a fund, without knowing anything beyond its past returns. This approach works because of the linear formula for net portfolio returns:

$$R_p = w_1 R_1 + \cdots + w_N R_N$$

Regressing portfolio returns on other assets, we can estimate the weights, assuming the portfolio composition is constant.

1.3.Jensen's Alpha and Fund Performance

Compare the performance of the mutual funds to that of a portfolio of ETFs by reporting the value of **Jensen's alpha** (based on the regressions from the previous part) and its corresponding **p-value**.

```
alpha=p.val=rep(0,N.MF)
for(i in 1:N.MF){
  alpha[i]=out[[i]]$coefficients[1]*250 # annualized Jensen alpha
  p.val[i]=summary(out[[i]])$coefficients[1,4]
}
cbind(alpha, p.val)
```

```
##            alpha      p.val
## [1,] -0.10324434 0.07786015
## [2,] -0.03384690 0.37016892
## [3,] -0.03855314 0.24977650
## [4,] -0.02484763 0.46805825
## [5,] -0.03037423 0.03986191
```

All the funds' alphas are negative, although their p-values are not very small. A likely cause for this is that funds charge a *fee* which consistently eats up some of the returns of their constituent assets. ETFs have typically lower fees than mutual funds, but our regression does not account for **transaction costs** (it is more costly to buy multiple assets than a single one), so the comparison is more nuanced.

Note that you can find Jensen alphas and other performance measures (e.g., Sharpe & Treynor Ratios) for assets in Yahoo! Finance under the risk tab (https://finance.yahoo.com/quote/FCNTX/risk?p=FCNTX). These metrics are based on the CAPM/Market factor model, by regressing the asset's returns on a proxy for the market return (e.g., S&P500).

2.Factor Analysis and Simulation

Use the following R code to download daily prices of 10 ETFs, from Jan 1, 2018 to Dec 31, 2019.

```
library(zoo)
library(tseries)
tickers = c("DVEM", "EXT", "HYEM", "LTPZ", "SCHP",
            "EDV",  "SPMB", "TLT", "GOVT")
S=list()
for(i in 1:length(tickers)){
  S[[i]] = get.hist.quote(tickers[i], start='2018-01-01', end='2019-12-31', quote='AdjClos
e', drop = TRUE)
}
```

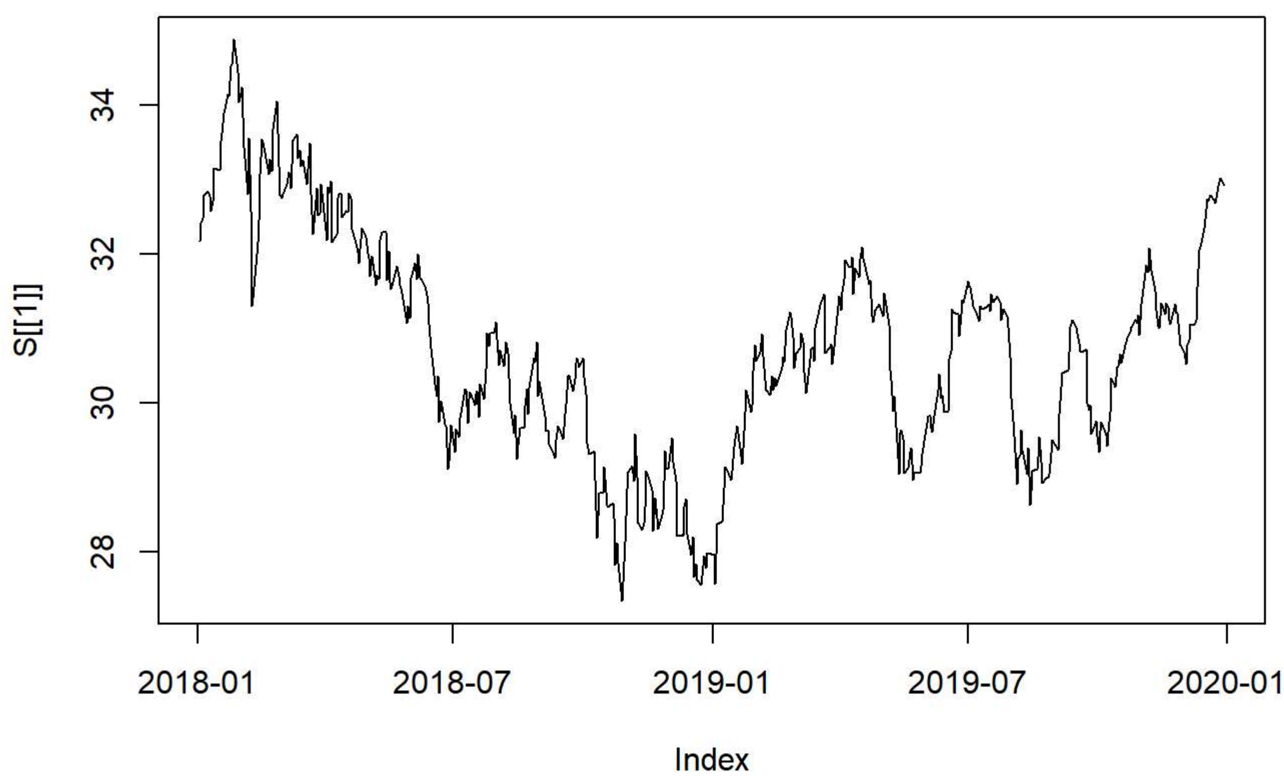
```
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
## time series starts 2018-01-02
## time series ends   2019-12-30
```

2.1.Return, LogReturn and plot

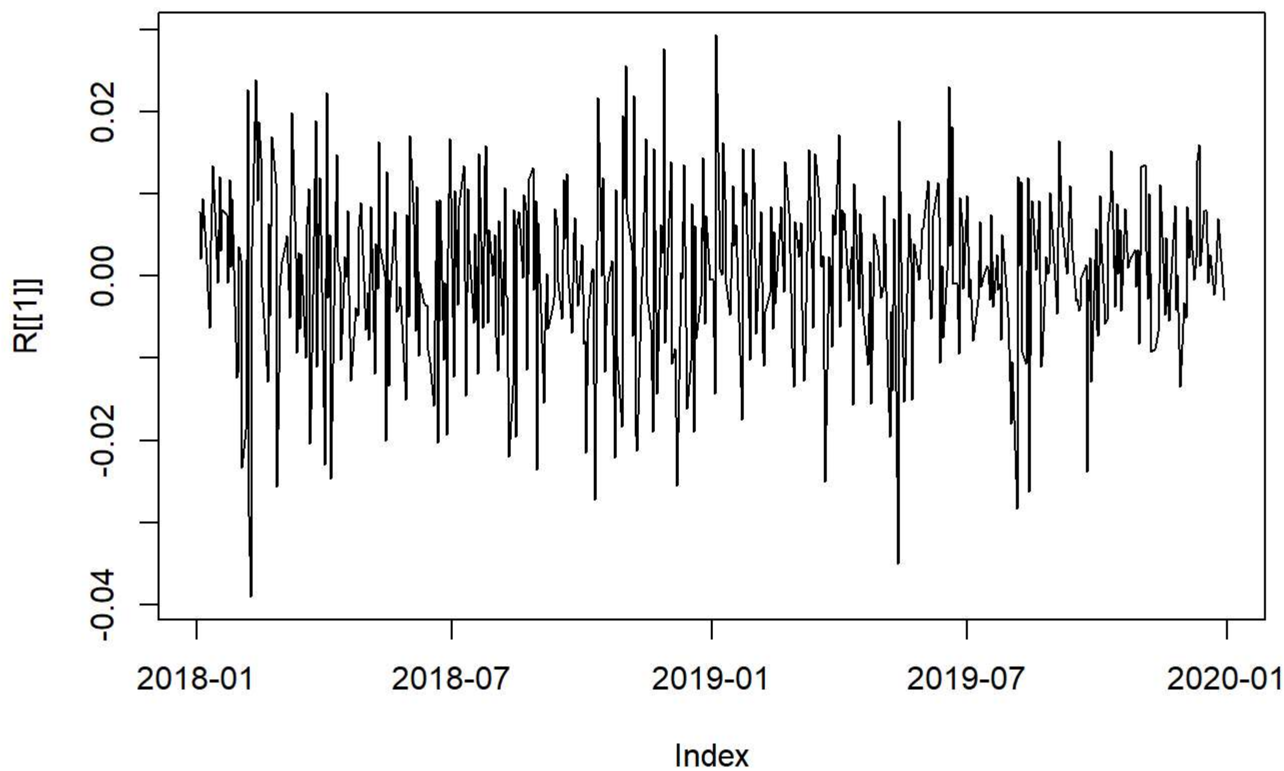
Calculate the log-returns of the ETFs, and plot the price and return series for the first ETF (DVEM)

```
r = lapply(S, FUN=function(x){diff(log(x))})
R = lapply(r, FUN=function(x){exp(x)-1})

plot(S[[1]])
```




```
plot(R[[1]])
```



2.2.Fact Analysis

Use `factanal()` to fit a 2-factor model to the correlation matrix of the returns. Report the factor loadings and idiosyncratic variances of you model.

```
Rmat = simplify2array(R) # Simplify list of matrices into a single array
fmod = factanal( Rmat, factors = 2,  lower = 0.005)

( b = fmod$loadings )
```

```
##
## Loadings:
##       Factor1 Factor2
##  [1,] -0.102   0.846
##  [2,] -0.251   0.788
##  [3,]          0.464
##  [4,]  0.881
##  [5,]  0.815
##  [6,]  0.979  -0.173
##  [7,]  0.544
##  [8,]  0.981  -0.183
##  [9,]  0.904  -0.214
##
##                Factor1 Factor2
## SS loadings      4.549   1.663
## Proportion Var   0.505   0.185
## Cumulative Var   0.505   0.690
```

```
( v = fmod$uniquenesses )
```

```
## [1] 0.27299172 0.31626790 0.78396875 0.22347432 0.33530372 0.01078944 0.70305519
## [8] 0.00500000 0.13730272
```

Or can do :

```
RX=matrix(unlist(R),ncol=length(tickers))
fact = factanal(RX, factors = 2, rotation = "none")
#print(fact)
```

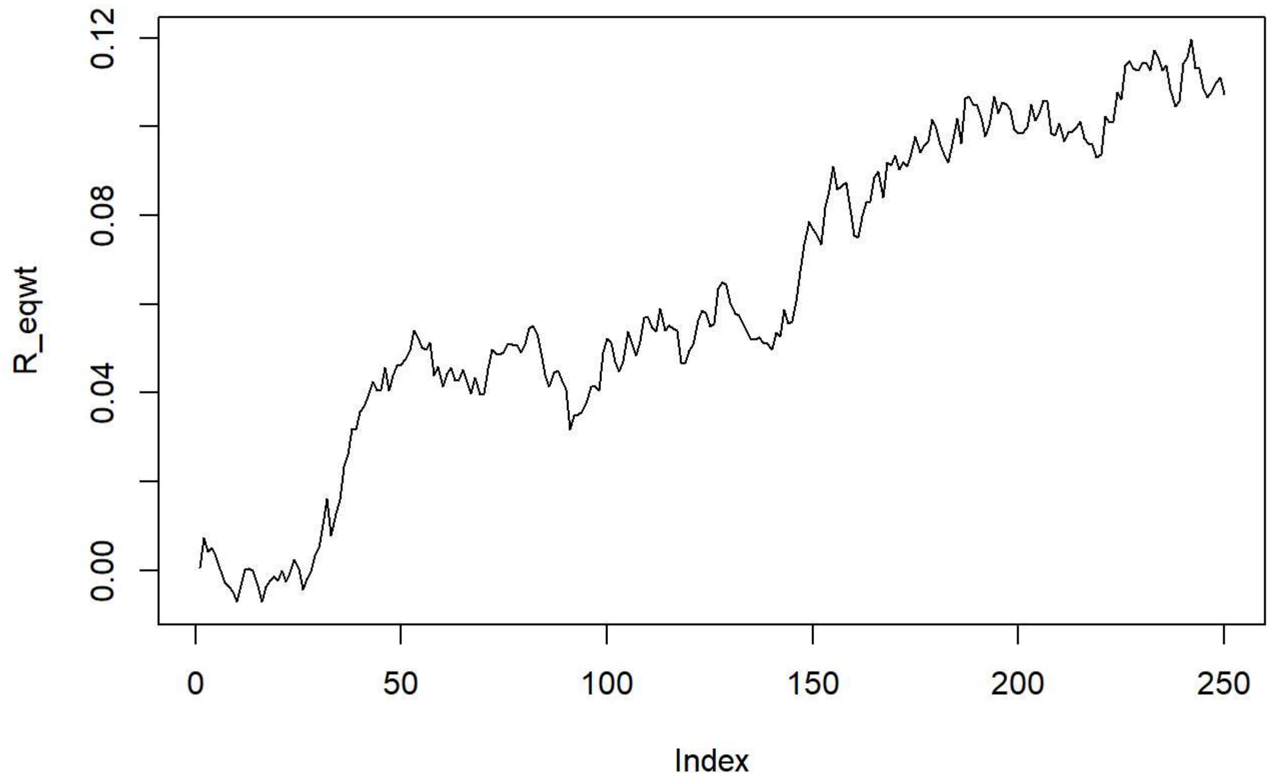
2.3.Multivariate normal simulation using Factor-model correlations

Simulate 250 daily log-returns using a **multivariate Normal distribution** with parameters given by the sample means and variances of the ETFs, and **correlation matrix given by the previous factor model**. Calculate and plot the cumulative net-returns of an equally weighted portfolio over the 10 ETFs.

```
MU = sapply(R, mean)
SD = sapply(R, sd)
VC = ( b%*%t(b) + diag(v) ) * (SD %*% t(SD))

library(mvtnorm)
Rsim = mvtnorm::rmvnorm(250, MU, VC)

R_eqwt = rowMeans( exp( apply(Rsim,2,cumsum) ) - 1 )
plot(R_eqwt, type = "l");
```



3.Factor Models of Asset Returns

In this section, we will start with the **one-factor CAPM model** of Chap. 17 and then extend this model to the **three-factor Fama–French model**.

- We will use the dataset Stock_Bond_2004_to_2005.csv on the book's website, which contains stock prices and other financial time series for the years 2004 and 2005. Data on the Fama–French factors are available at Prof. Kenneth French's website, where `RF` is the **risk-free rate** and `Mkt.RF`, `SMB`,and `HML` are the **Fama–French factors.**

Go to Prof. French's website and get the daily values of `RF`, `Mkt.RF`, `SMB`, and `HML` for the years 2004–2005. It is assumed here that you've put the data in a text file FamaFrenchDaily.txt. Returns on this website are expressed as **percentages**.

Model 1: One-Factor CAPM

Now fit the CAPM to the four stocks using the `lm` command. This code fits a linear regression model separately to the four responses. In each case, the independent variable is `Mkt.RF`.

We apply a linear regression model where excess returns of four selected stocks (GM, Ford, UTX, Merck) are regressed on the market excess return (`Mkt.RF`).

```
# Uses daily data 2004-2005
stocks = read.csv("Stock_FX_Bond_2004_to_2005.csv",header=T)
attach(stocks)

FF_data = read.table("FamaFrenchDaily.txt", header = TRUE)
FF_data = FF_data[-1, ] # delete first row since stocks_diff

stocks_subset = as.data.frame(cbind(GM_AC, F_AC, UTX_AC, MRK_AC))
# Calculating Log Returns and Differencing
stocks_diff = as.data.frame(100 * apply(log(stocks_subset),2, diff)- FF_data$RF)
names(stocks_diff) = c("GM", "Ford", "UTX", "Merck")


# lost a row due to differencing
fit1 = lm(as.matrix(stocks_diff) ~ FF_data$Mkt.RF)
summary(fit1)
```

```
## Response GM :
##
## Call:
## lm(formula = GM ~ FF_data$Mkt.RF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8316  -0.7213   0.0241   0.8083  15.2480
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.22902    0.08646  -2.649  0.00833 **
## FF_data$Mkt.RF  1.25000    0.12730   9.819  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.937 on 501 degrees of freedom
## Multiple R-squared:  0.1614, Adjusted R-squared:  0.1597
## F-statistic: 96.41 on 1 and 501 DF,  p-value: < 2.2e-16
##
##
## Response Ford :
##
## Call:
## lm(formula = Ford ~ FF_data$Mkt.RF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3512 -0.8506  0.0183  0.7905  9.1818
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.18347    0.06757  -2.715  0.00685 **
## FF_data$Mkt.RF  1.31952    0.09950  13.262  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 501 degrees of freedom
## Multiple R-squared:  0.2598, Adjusted R-squared:  0.2584
## F-statistic: 175.9 on 1 and 501 DF,  p-value: < 2.2e-16
##
##
## Response UTX :
##
## Call:
## lm(formula = UTX ~ FF_data$Mkt.RF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3531 -0.5407 -0.0075  0.5370  3.6128
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.002193   0.038835   0.056    0.955
## FF_data$Mkt.RF 0.919322   0.057183  16.077   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.87 on 501 degrees of freedom
## Multiple R-squared:  0.3403, Adjusted R-squared:  0.339
## F-statistic: 258.5 on 1 and 501 DF,  p-value: < 2.2e-16
##
##
## Response Merck :
##
## Call:
## lm(formula = Merck ~ FF_data$Mkt.RF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -31.1555  -0.4810   0.0587   0.7090  12.3073
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.08883    0.09203  -0.965    0.335
## FF_data$Mkt.RF  0.62545    0.13552   4.615 4.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.062 on 501 degrees of freedom
## Multiple R-squared:  0.04078,    Adjusted R-squared:  0.03887
## F-statistic:  21.3 on 1 and 501 DF,  p-value: 4.994e-06
```

(a).Testing Intercepts

The CAPM predicts that all four intercepts will be zero. For each stock, using α =0.025, can you accept the null hypothesis that its intercept is zero? Why or why not? Include the p-values with your work.

- The intercepts p-value is below 0.025 for General motors (GM) and Ford, but not for United Technologies Incorporatted (UTX) and Merck. Of course, a p-value only shows statistical significance, not the size of an effect.
- However, estimated intercepts for GM and Ford are -0.23 and -0.18 and these are reasonably large in magnitude. Since they are negative, this suggests than these two stocks were overpriced.

(b).Residual Correlations

The CAPM also predicts that the four sets of residuals will be uncorrelated. What is the correlation matrix of the residuals? Give a 95% confidence interval for each of the six correlations. Can you accept the hypothesis that all six correlations are zero?

- The correlation matrix is below. All correlations are reasonably close to 0 (less than 0.1 in magnitude) except the correlation between GM and Ford residuals.

- That correlation is 0.52 and has a very small p-value. The correlation between GM and Merck residuals is -0.0878 and is statistically significant at 0.05 but might be too small to be of practical significance.

```
cor(residuals(fit1))
```

```
##                GM         Ford          UTX        Merck
## GM     1.00000000  0.520164695 -0.010045234 -0.087756009
## Ford   0.52016469  1.000000000 -0.023761716 -0.009579221
## UTX   -0.01004523 -0.023761716  1.000000000 -0.005502672
## Merck -0.08775601 -0.009579221 -0.005502672  1.000000000
```

```
res = residuals(fit1)
cor.test(res[,"GM"],res[,"Ford"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "GM"] and res[, "Ford"]
## t = 13.632, df = 501, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4533535 0.5811635
## sample estimates:
##       cor
## 0.5201647
```

```
cor.test(res[,"GM"],res[,"UTX"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "GM"] and res[, "UTX"]
## t = -0.22485, df = 501, p-value = 0.8222
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09738817  0.07745125
## sample estimates:
##         cor
## -0.01004523
```

```
cor.test(res[,"GM"],res[,"Merck"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "GM"] and res[, "Merck"]
## t = -1.9719, df = 501, p-value = 0.04918
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.173850629 -0.000330075
## sample estimates:
##         cor
## -0.08775601
```

```
cor.test(res[,"Ford"],res[,"UTX"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "Ford"] and res[, "UTX"]
## t = -0.53201, df = 501, p-value = 0.595
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.11095967  0.06379929
## sample estimates:
##         cor
## -0.02376172
```

```
cor.test(res[,"Ford"],res[,"Merck"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "Ford"] and res[, "Merck"]
## t = -0.21442, df = 501, p-value = 0.8303
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09692651  0.07791450
## sample estimates:
##          cor
## -0.009579221
```

```
cor.test(res[,"UTX"],res[,"Merck"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "UTX"] and res[, "Merck"]
## t = -0.12317, df = 501, p-value = 0.902
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09288645  0.08196523
## sample estimates:
##          cor
## -0.005502672
```

(c).Covariance Matrix CAPM Estimation

Regardless of your answer to Problem 6, assume for now that the residuals are **uncorrelated**. Then use the CAPM to estimate the **covariance matrix** of the excess returns on the four stocks. Compare this estimate with the **sample covariance matrix** of the excess returns. Do you see any large discrepancies between the two estimates of the covariance matrix?

- We see below that the estimated covariance matrix using the CAPM is similar to the sample covariance matrix, with the exception of the covariance between GM and Ford. Since these two stocks have a high residual correlation and the CAPM assumes that the residual correlation is 0, it is not surprising that the CAPM estimated covariance matrix severely underestimates the correlation between GM and Ford.

```
attach(FF_data)
sigF = var(Mkt.RF)
bbeta = as.matrix(fit1$coef)
bbeta = bbeta[-1,] # delete intercepts so bbeta has the four slopes
n=dim(stocks_diff)[1]
sigeps = as.matrix((var(as.matrix(res))))

sigeps_ind = diag(as.matrix(sigeps))
sigeps_ind = diag(sigeps_ind,nrow=4)
cov_equities = sigF* bbeta %*% t(bbeta) + sigeps_ind
cov_equities
```

```
##             GM      Ford       UTX     Merck
## [1,] 4.4640945 0.7605160 0.5298576 0.3604834
## [2,] 0.7605160 3.0896783 0.5593293 0.3805342
## [3,] 0.5298576 0.5593293 1.1450430 0.2651212
## [4,] 0.3604834 0.3805342 0.2651212 4.4228337
```

```
cov(stocks_diff)
```

```
##                GM      Ford       UTX      Merck
## GM    4.46409450 2.2824942 0.5129656 0.01075292
## Ford  2.28249418 3.0896783 0.5280993 0.35069689
## UTX   0.51296558 0.5280993 1.1450430 0.25527074
## Merck 0.01075292 0.3506969 0.2552707 4.42283371
```

If use the true residual correlation.

```
bbeta_mat <- matrix(bbeta, nrow = 1)
cov_equities = t(bbeta_mat) %*% sigF %*% (bbeta_mat) + sigeps_ind
cov_equities
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 4.4640945 0.7605160 0.5298576 0.3604834
## [2,] 0.7605160 3.0896783 0.5593293 0.3805342
## [3,] 0.5298576 0.5593293 1.1450430 0.2651212
## [4,] 0.3604834 0.3805342 0.2651212 4.4228337
```

```
cov(stocks_diff)
```

```
##                GM      Ford       UTX      Merck
## GM    4.46409450 2.2824942 0.5129656 0.01075292
## Ford  2.28249418 3.0896783 0.5280993 0.35069689
## UTX   0.51296558 0.5280993 1.1450430 0.25527074
## Merck 0.01075292 0.3506969 0.2552707 4.42283371
```

Model 2: Fama–French Three-Factor Model

Next, you will fit the Fama–French three-factor model. Run the following R code, which is much like the previous code except that the regression model has two additional predictor variables, SMB and HML.

```
fit2 = lm(as.matrix(stocks_diff) ~ FF_data$Mkt.RF +
FF_data$SMB + FF_data$HML)
summary(fit2)
```

```
## Response GM :
##
## Call:
## lm(formula = GM ~ FF_data$Mkt.RF + FF_data$SMB + FF_data$HML)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7591  -0.7271  -0.0326   0.7751  15.0074
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.25114    0.08647  -2.904  0.00384 **
## FF_data$Mkt.RF  1.38891    0.15166   9.158  < 2e-16 ***
## FF_data$SMB    -0.25044    0.22364  -1.120  0.26334
## FF_data$HML     0.60056    0.26427   2.272  0.02348 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.927 on 499 degrees of freedom
## Multiple R-squared:  0.1733, Adjusted R-squared:  0.1683
## F-statistic: 34.86 on 3 and 499 DF,  p-value: < 2.2e-16
##
##
## Response Ford :
##
## Call:
## lm(formula = Ford ~ FF_data$Mkt.RF + FF_data$SMB + FF_data$HML)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3444 -0.8324 -0.0099  0.7933  9.1975
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.19508    0.06787  -2.874  0.00422 **
## FF_data$Mkt.RF  1.35115    0.11905  11.349  < 2e-16 ***
## FF_data$SMB    -0.01570    0.17556  -0.089  0.92876
## FF_data$HML     0.34122    0.20745   1.645  0.10064
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.513 on 499 degrees of freedom
## Multiple R-squared:  0.264,  Adjusted R-squared:  0.2595
## F-statistic: 59.65 on 3 and 499 DF,  p-value: < 2.2e-16
##
##
## Response UTX :
##
## Call:
## lm(formula = UTX ~ FF_data$Mkt.RF + FF_data$SMB + FF_data$HML)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -3.2694 -0.5369  0.0082  0.4931  3.4288 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    -3.991e-06  3.878e-02   0.000  0.99992    
## FF_data$Mkt.RF  1.029e+00  6.803e-02  15.121  < 2e-16 ***
## FF_data$SMB    -2.927e-01  1.003e-01  -2.918  0.00369 ** 
## FF_data$HML    -9.595e-04  1.185e-01  -0.008  0.99354    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.8643 on 499 degrees of freedom
## Multiple R-squared:  0.3516, Adjusted R-squared:  0.3477 
## F-statistic: 90.18 on 3 and 499 DF,  p-value: < 2.2e-16
## 
## 
## Response Merck :
## 
## Call:
## lm(formula = Merck ~ FF_data$Mkt.RF + FF_data$SMB + FF_data$HML)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -30.6271  -0.4536   0.1062   0.6905  12.3105 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    -0.05983    0.09148  -0.654  0.51337    
## FF_data$Mkt.RF  0.70927    0.16045   4.420 1.21e-05 ***
## FF_data$SMB    -0.41740    0.23661  -1.764  0.07832 .  
## FF_data$HML    -0.95592    0.27959  -3.419  0.00068 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.039 on 499 degrees of freedom
## Multiple R-squared:  0.06604,    Adjusted R-squared:  0.06042 
## F-statistic: 11.76 on 3 and 499 DF,  p-value: 1.863e-07
```

(a).Testing Additional Factor Relevance

The CAPM predicts that for each stock, the slope (beta) for SMB and HML will be zero. Explain why the CAPM makes this prediction. Do you accept this null hypothesis? Why or why not?

(NO SOLUTION)

(b).Residual Correlations

If the Fama–French model explains all covariances between the returns, then the correlation matrix of the residuals should be diagonal. What is the estimated correlations matrix? Would you accept the hypothesis that the correlations are all zero?

(NO SOLUTION)

```
cor(residuals(fit2))
```

```
##                GM          Ford         UTX         Merck
## GM     1.00000000  0.5170146848 -0.01852131 -0.0778186342
## Ford   0.51701468  1.0000000000 -0.02577070  0.0007289032
## UTX   -0.01852131 -0.0257707016  1.00000000 -0.0135962856
## Merck -0.07781863  0.0007289032 -0.01359629  1.0000000000
```

```
res = residuals(fit2)
cor.test(res[,"GM"],res[,"Ford"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "GM"] and res[, "Ford"]
## t = 13.519, df = 501, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4499236 0.5783028
## sample estimates:
##       cor
## 0.5170147
```

```
cor.test(res[,"GM"],res[,"UTX"])
```

```
##
##  Pearson's product-moment correlation
##
## data:  res[, "GM"] and res[, "UTX"]
## t = -0.41463, df = 501, p-value = 0.6786
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.10577849  0.06901892
## sample estimates:
##         cor
## -0.01852131
```

```
# cor.test(res[,"GM"],res[,"Merck"])
# cor.test(res[,"Ford"],res[,"UTX"])
# cor.test(res[,"Ford"],res[,"Merck"])
# cor.test(res[,"UTX"],res[,"Merck"])
```

(c).Model Comparison CAPM vs Fama-French

Which model, CAPM or Fama–French, has the smaller value (better) of AIC? Which has the smaller value of BIC? What do you conclude from this?

(NO SOLUTION)

The `AIC()` and `BIC()` functions need models with a **single response** (like just `y ~ x`). For multivariate models, `logLik()` (which AIC/BIC rely on) isn't defined by default.

```
fit_AICBIC=fit1
# Residual Sum of Squares
rss <- sum(residuals(fit_AICBIC)^2)

# Number of observations and parameters
n=dim(stocks_diff)[1]
k <- length(coef(fit_AICBIC))  # or use length(fit$coefficients) * number of responses

# AIC and BIC (based on Gaussian likelihood approximation)
(aic <- n * log(rss / n) + 2 * k)
```

```
## [1] 1222.434
```

```
(bic <- n * log(rss / n) + log(n) * k)
```

```
## [1] 1256.199
```

(d).Covariance Matrix

Factor Covariance Matrix

What is the covariance matrix of the three Fama–French factors?

```
#attach(FF_data)
SIGMA_F = cov(FF_data[,c(2,3,4)])
SIGMA_F
```

```
##             Mkt.RF         SMB         HML
## Mkt.RF  0.46108683  0.17229574 -0.03480511
## SMB     0.17229574  0.21464312 -0.02904749
## HML    -0.03480511 -0.02904749  0.11023817
```

Covariance Matrix of Return - by Fama–French Model

```
fit2 = lm(as.matrix(stocks_diff) ~ FF_data$Mkt.RF+FF_data$SMB + FF_data$HML)
# summary(fit2)
```

```
res = residuals(fit2)

FF3=FF_data[,c(2,3,4)]
SIGMA_F = cov(FF3)
bbeta = as.matrix(fit2$coef)
bbeta = bbeta[-1,] # delete intercepts so bbeta has the four slopes
n=dim(stocks_diff)[1]
sigeps = as.matrix((var(as.matrix(res))))
sigeps_ind = diag(as.matrix(sigeps))
sigeps_ind = diag(sigeps_ind,nrow=4)
cov_equities_indeps = t(bbeta) %*% SIGMA_F %*% (bbeta) + sigeps_ind

cov_equities = t(bbeta) %*% SIGMA_F %*%(bbeta) + sigeps
cov_equities_indeps;cov_equities;cov(stocks_diff)
```

```
##              GM      Ford       UTX     Merck
## GM    4.4640945 0.7846809 0.5436251 0.3145936
## Ford  0.7846809 3.0896783 0.5615865 0.3484629
## UTX   0.5436251 0.5615865 1.1450430 0.2790819
## Merck 0.3145936 0.3484629 0.2790819 4.4228337
```

```
##                GM      Ford       UTX      Merck
## GM    4.46409450 2.2824942 0.5129656 0.01075292
## Ford  2.28249418 3.0896783 0.5280993 0.35069689
## UTX   0.51296558 0.5280993 1.1450430 0.25527074
## Merck 0.01075292 0.3506969 0.2552707 4.42283371
```

```
##                GM      Ford       UTX      Merck
## GM    4.46409450 2.2824942 0.5129656 0.01075292
## Ford  2.28249418 3.0896783 0.5280993 0.35069689
## UTX   0.51296558 0.5280993 1.1450430 0.25527074
## Merck 0.01075292 0.3506969 0.2552707 4.42283371
```

(e).Predicting Variance and Covariance for New Stocks

In this problem, Stocks 1 and 2 are two stocks, not necessarily in the Stock_FX_Bond_2004_to_2005.csv data set. Suppose that Stock 1 has betas of 0.5,0.4,and −0.1 with respect to the three factors in the Fama–French model and a residual variance of 23.0. Suppose also that Stock 2 has betas of 0.6, 0.15, and 0.7 with respect to the three factors and a residual variance of 37.0. Regardless of your answer to Problem 9, when doing this problem, assume that the three factors do account for all covariances.

a. Use the Fama–French model to estimate the variance of the excess return on Stock 1.

```
# factor covariance matrix
Sigma_F <- cov(FF_data[,3:5],FF_data[,3:5])  # columns: Mkt.RF, SMB, HML


# Define betas and residual variance
beta1 <- c(0.5, 0.4, -0.1)
resid_var1 <- 23.0


# Calculate the factor-related variance
var_factor <- t(beta1) %*% Sigma_F %*% beta1

# Total variance
(var_stock1 <- var_factor + resid_var1)
```

```
##          [,1]
## [1,] 23.05968
```

b. Use the Fama–French model to estimate the variance of the excess return on Stock 2. And use the Fama–French model to estimate the covariance between the excess returns on Stock 1 and Stock 2.

```
# Define betas for Stock 2 and residual variance
beta2 <- c(0.6, 0.15, 0.7)
resid_var2 <- 37.0

# Calculate the variance for Stock 2
var_factor2 <- t(beta2) %*% Sigma_F %*% beta2
(var_stock2 <- var_factor2 + resid_var2)
```

```
##          [,1]
## [1,] 37.07452
```

```
# Calculate the covariance between Stock 1 and Stock 2
(cov12 <- t(beta1) %*% Sigma_F %*% beta2)
```

```
##            [,1]
## [1,] 0.06184519
```

(NO SOLUTION)

Model 3: Statistical Factor Analysis

This section applies statistical factor analysis to the log returns of 10 stocks in the data set Stock_FX_Bond.csv. The data set contains adjusted closing (AC) prices of the stocks, as well as daily volumes and other information that we will not use here.

(a).Exploratory Factor Modeling

The following R code will read the data, compute the log returns, and fit a two-factor model. Note that `factanal` works with the correlation matrix or, equivalently, with standardized variables.

```
dat = read.csv("Stock_FX_Bond.csv")
stocks_ac = dat[ , c(3, 5, 7, 9, 11, 13, 15, 17)]
n = length(stocks_ac[ , 1])
stocks_returns = log(stocks_ac[-1, ] / stocks_ac[-n, ])

fact = factanal(stocks_returns, factors = 2, rotation = "none")
print(fact)
```

```
##
## Call:
## factanal(x = stocks_returns, factors = 2, rotation = "none")
##
## Uniquenesses:
##   GM_AC    F_AC  UTX_AC  CAT_AC  MRK_AC  PFE_AC  IBM_AC MSFT_AC
##   0.399   0.423   0.718   0.714   0.519   0.410   0.760   0.749
##
## Loadings:
##         Factor1 Factor2
## GM_AC    0.693  -0.348
## F_AC     0.692  -0.313
## UTX_AC   0.531
## CAT_AC   0.529
## MRK_AC   0.551   0.421
## PFE_AC   0.574   0.511
## IBM_AC   0.490
## MSFT_AC  0.499
##
##                Factor1 Factor2
## SS loadings      2.643   0.666
## Proportion Var   0.330   0.083
## Cumulative Var   0.330   0.414
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 564.66 on 13 degrees of freedom.
## The p-value is 2.6e-112
```

Loadings less than the parameter cutoff are not printed. The default value of cutoff is 0.1, but you can change it as in `print(fact,cutoff = 0.01)` or `print(fact, cutoff = 0)`.

(b).Interpreting Factor Loadings

What are the factor loadings? What are the variances of the unique risks for Ford and General Motors?

- The factor loadings and uniqueness are in the output below. We see that the uniquenesses for Ford and GM are 0.423 and 0.399, respectively.

(c).Number of Factors Needed

Does the likelihood ratio test suggest that two factors are enough? If not, what is the minimum number of factors that seems sufficient?

- The results of the likelihood ratio tests below strongly suggest that there are more than two factors. It seems that 4 factors are sufficient, but not 3.

```
Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 564.66 on 13 degrees of freedom.
The p-value is 2.6e-112

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 162.29 on 7 degrees of freedom.
The p-value is 1.06e-31

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 0.3 on 2 degrees of freedom.
The p-value is 0.86
```

(d).Estimating Correlations from Factor Model

Regardless of your answer to Problem, use the two-factor model to estimate the correlation of the log returns for Ford and IBM.

The output below contains the **estimated correlation matrix using the factor model** after line 4. For comparison, the **sample correlation matrix** is also printed.

```
loadings = matrix(as.numeric(loadings(fact)), ncol = 2)
unique = as.numeric(fact$unique)
options(digits=2)
loadings %*% t(loadings) + diag(unique)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 1.00 0.59 0.38 0.39 0.24 0.22 0.34 0.33
## [2,] 0.59 1.00 0.38 0.39 0.25 0.24 0.34 0.33
## [3,] 0.38 0.38 1.00 0.28 0.28 0.29 0.26 0.26
## [4,] 0.39 0.39 0.28 1.00 0.26 0.26 0.26 0.26
## [5,] 0.24 0.25 0.28 0.26 1.00 0.53 0.27 0.29
## [6,] 0.22 0.24 0.29 0.26 0.53 1.00 0.28 0.31
## [7,] 0.34 0.34 0.26 0.26 0.27 0.28 1.00 0.24
## [8,] 0.33 0.33 0.26 0.26 0.29 0.31 0.24 1.00
```

```
cor(stocks_returns)
```

```
##         GM_AC F_AC UTX_AC CAT_AC MRK_AC PFE_AC IBM_AC MSFT_AC
## GM_AC    1.00 0.62   0.35   0.36   0.25   0.23   0.32    0.31
## F_AC     0.62 1.00   0.35   0.37   0.26   0.25   0.30    0.30
## UTX_AC   0.35 0.35   1.00   0.40   0.26   0.28   0.29    0.28
## CAT_AC   0.36 0.37   0.40   1.00   0.24   0.25   0.30    0.29
## MRK_AC   0.25 0.26   0.26   0.24   1.00   0.55   0.24    0.27
## PFE_AC   0.23 0.25   0.28   0.25   0.55   1.00   0.26    0.29
## IBM_AC   0.32 0.30   0.29   0.30   0.24   0.26   1.00    0.41
## MSFT_AC  0.31 0.30   0.28   0.29   0.27   0.29   0.41    1.00
```