Stats_C183_Project_6_Charles_Liu

Charles Liu (304804942)

5/28/2020

Set up the Data:

```
library(readr)
# Loading the data for all stocks:
a <- read.csv("C:/Users/cliuk/Documents/UCLA Works/UCLA Spring 2020/Stats C183/Project/stockData_2012-20

# Calculate the returns
r <- (a[-1,3:ncol(a)]-a[-nrow(a),3:ncol(a)])/a[-nrow(a),3:ncol(a)]
```

a)

```
# Compute the Means, Standard Deviation, Average Correlations for Industries
means <- colMeans(r[-1])
covmat <- var(r[-1])
stdev <- diag(covmat)^.5
cormat <- cor(r[-1])

# Compute the average correlation matrix based on industry:
rho_bar <- matrix(NA, nrow = 5, ncol = 5)
for(i in 1:5){
for(j in 1:5){
temp <- cormat[(6*(i - 1) + 1):(6*i), (6*(j - 1) + 1):(6*j)]
if(i == j){
rho_bar[i,j] <- mean(temp[upper.tri(temp)])
} else {
rho_bar[i,j] <- mean(temp)
}
}
}

# Compute matrix A:
A <- matrix(NA, nrow = 5, ncol = 5)
for(i in 1:5){
for(j in 1:5){
if(i == j){
A[i,j] <- 1 + (6*rho_bar[i,i] / (1 - rho_bar[i,i]))
} else {
A[i,j] <- 6*rho_bar[i,j] / (1 - rho_bar[i,i])
}
}
```

```r
}

# Choose a Rf value:
rf <- 0.005

# Compute matrix C:
C <- rep(NA, 5)
for(i in 1:5){
C[i] <- sum((means[(6*(i - 1) + 1):(6*i)] - rf) / (stdev[(6*(i - 1) + 1):(6*i)]*(1 - rho_bar[i,i])))
}

# Compute matrix phi:
phi <- solve(A) %*% C

# Calculate the cut-off points C*:
C_star <- rho_bar %*% phi

# Calculate Z:
z <- rep(NA, 30)
for(i in 1:30){
k <- ceiling(i/6)
z[i] <- (1 / (stdev[i]*(1 - rho_bar[k,k])))*(((means[i] - rf) / stdev[i]) - C_star[k])
}

# Calculate the composition X:
x <- z/sum(z)
names(x) <- names(means)

# The composition under MGM:
x
```

```
##           BHP          GOLD          VALE          GOOG             T
## -0.3794669116 -0.1503740661 -0.1673797386  0.4640076254  0.3183253332
##          NFLX          AMZN           MCD          TSLA           WMT
##  0.1804806840  0.3145764079 -0.3784027479  0.1252040211 -0.5058271878
##            KO          COST           XOM           CVX           TRP
## -0.3467354321  0.3845250599 -0.8833920783 -0.4936786760 -0.5144474274
##         BRK.B             V           JPM           JNJ          AMGN
##  0.5480234790  1.0063042779  0.2241384535  0.3111563709  0.1302698591
##           CVS           UNP            BA            GE           DLR
##  0.1123208417  0.1269996909  0.1851872779  0.0545075516 -0.0560078525
##           BXP             O          MSFT          AAPL          NVDA
## -0.4321580493  0.0363899421  0.2073462121  0.0007036288  0.5774034504
```

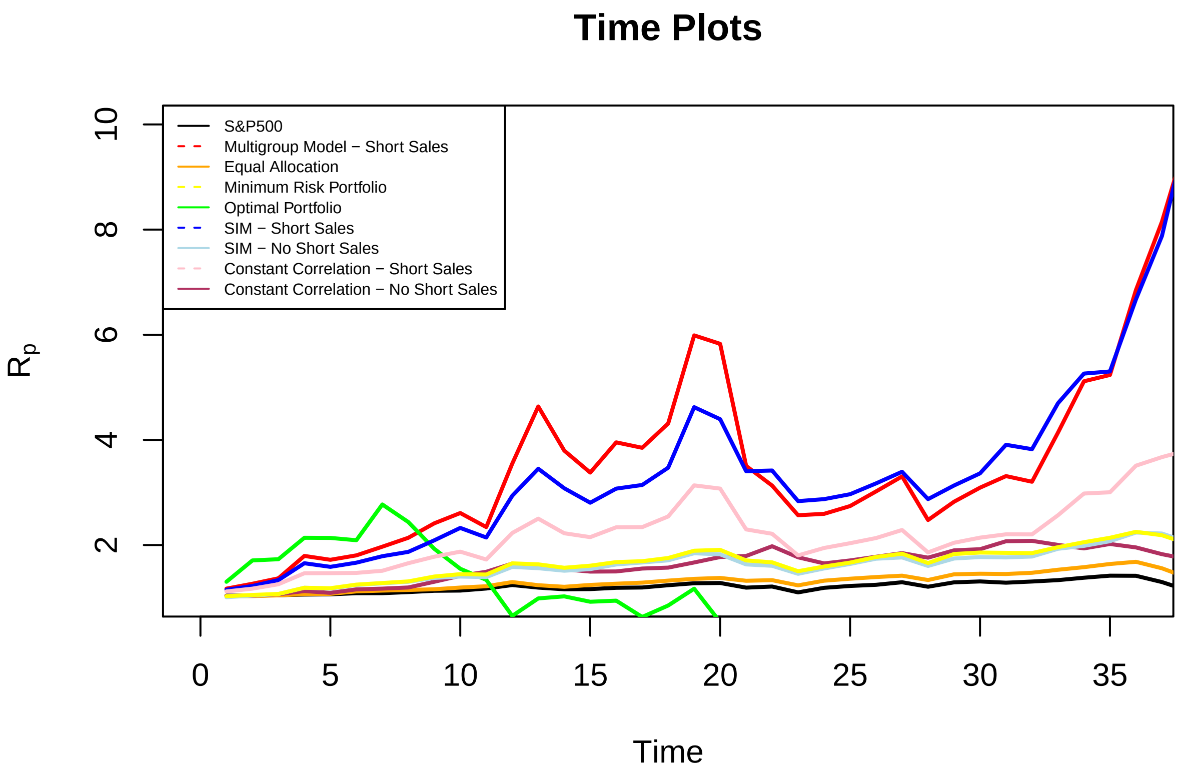
b(i))

```r
# Read stock csv file for the period 2017-01-01 to 2020-03-01:
a2 <- read.csv("C:/Users/cliuk/Documents/UCLA Works/UCLA Spring 2020/Stats C183/Project/stockData_2017-2
# Convert adjusted close prices into returns:
r2 <- (a2[-1,3:ncol(a2)] - a2[-nrow(a2),3:ncol(a2)]) / a2[-nrow(a2),3:ncol(a2)]
# Convert S&P500 into returns:
r_sp500 <- (a2[-1,2] - a2[-nrow(a2),2]) / a2[-nrow(a2),2]
# Compute variance covariance matrix:
```

```r
covmat2 <- cov(r2)
# Compute the vector of standard deviations:
stdev2 <- sqrt(diag(covmat2))
# Multigroup Model Optimal Portfolio - from (a)
rp_mgm <- as.matrix(r2) %*% x
sd_mgm <- sqrt(t(x) %*% covmat2 %*% x)
# Equal Allocation Portfolio
x_equal <- rep(1/30, 30)
rp_equal <- as.matrix(r2) %*% x_equal
sd_equal <- sqrt(t(x_equal) %*% covmat2 %*% x_equal)
# Minimum Risk Portfolio
ones <- rep(1, 30)
x_MRP <- (solve(covmat) %*% ones) / as.numeric(t(ones) %*% solve(covmat) %*% ones)
rp_MRP <- as.matrix(r2) %*% x_MRP
sd_MRP <- sqrt(t(x_MRP) %*% covmat2 %*% x_MRP)
# Optimal Portfolio
R <- means - rf
z_optimal <- solve(covmat) %*% R
x_optimal <- z_optimal / sum(z_optimal)
rp_optimal <- as.matrix(r2) %*% x_optimal
sd_optimal <- sqrt(t(x_optimal) %*% covmat2 %*% x_optimal)
# Single Index Model Optimal Portfolio
r_SIM <- (a[-1, 3:ncol(a)] - a[-nrow(a), 3:ncol(a)]) / a[-nrow(a),3:ncol(a)]

beta <- rep(0, 30)
alpha <- rep(0, 30)
sigma_e2 <- rep(0, 30)
Ribar <- rep(0, 30)
Ratio <- rep(0, 30)
stock <- rep(0, 30)
for(i in 1:30){
  q <- lm(data = r_SIM, formula = r_SIM[,i+1] ~ r_SIM[,1])
  beta[i] <- q$coefficients[2]
  alpha[i] <- q$coefficients[1]
  sigma_e2[i] <- summary(q)$sigma^2
  Ribar[i] <- q$coefficients[1] + q$coefficients[2] * mean(r_SIM[,1])
  Ratio[i] <- (Ribar[i] - rf)/beta[i]
  stock[i] <- i
}
xx <- (cbind(stock, alpha, beta, Ribar, sigma_e2, Ratio))
A <- xx[order(-xx[,6]),]
col1 <- rep(0, nrow(A))
col2 <- rep(0, nrow(A))
col3 <- rep(0, nrow(A))
col4 <- rep(0, nrow(A))
col5 <- rep(0, nrow(A))
col1 <- (A[,4] - rf)*A[,3] / A[,5]
col3 <- A[,3]^2 / A[,5]

for(i in(1:nrow(A))){
  col2[i] <- sum(col1[1:i])
  col4[i] <- sum(col3[1:i])
}
```

```r
for(i in (1:nrow(A))){
  col5[i] <- var(r_SIM$X.GSPC)*col2[i] / (1 + var(r_SIM$X.GSPC)*col4[i])
}
#SHORT SALES ALLOWED:
z_short <- (A[,3]/A[,5])*(A[,6]-col5[nrow(A)])
x_sim_short <- z_short/sum(z_short)
x_sim_short2 <- x_sim_short[order(A[,1])]
rp_sim_short <- as.matrix(r2) %*% x_sim_short2
sd_sim_short <- sqrt(t(x_sim_short2) %*% covmat2 %*% x_sim_short2)

# SHORT SALES NOT ALLOWED:
table1 <- cbind(A, col1, col2, col3, col4, col5)
table2 <- table1[1:which(col5 == max(col5)), ]
z_no_short <- (table2[,3]/table2[,5])*(table2[,6]-max(col5))
x_sim_no_short <- z_no_short/sum(z_no_short)
x_sim_no_short2 <- x_sim_no_short[order(A[,1])]
x_sim_no_short2 <- ifelse(is.na(x_sim_no_short2), 0, x_sim_no_short2)
rp_sim_no_short <- as.matrix(r2) %*% x_sim_no_short2
sd_sim_no_short <- sqrt(t(x_sim_no_short2) %*% covmat2 %*% x_sim_no_short2)

# Constant Correlation Model Optimal Portfolio
stock <- 1:30
diff <- means - rf
ratio <- diff / stdev
avg_cor <- (sum(cormat) - 30) / (30*29)
xx <- (cbind(stock, means, diff, stdev, ratio))
A <- xx[order(-xx[,5]),]
col1 <- rep(0, nrow(A))
col2 <- rep(0, nrow(A))
col3 <- rep(0, nrow(A))
col1 <- avg_cor / (1 - avg_cor + stock*avg_cor)
for(i in(1:nrow(A))){
  col2[i] <- sum(A[1:i, 5])
}
col3 <- col1 * col2

# SHORT SALES ALLOWED:
z_short <- (A[,5] - col3[nrow(A)]) / ((1 - avg_cor)*A[,4])
x_cc_short <- z_short/sum(z_short)
x_cc_short2 <- x_cc_short[order(A[,1])]
rp_cc_short <- as.matrix(r2) %*% x_cc_short2
sd_cc_short <- sqrt(t(x_cc_short2) %*% covmat2 %*% x_cc_short2)

# SHORT SALES NOT ALLOWED:
table1 <- cbind(A, col1, col2, col3)
table2 <- table1[1:which(col3 == max(col3)), ]
z_no_short <- (table2[,5] - col3[nrow(table2)]) / ((1 - avg_cor)*table2[,4])
x_cc_no_short <- z_no_short/sum(z_no_short)
x_cc_no_short2 <- x_cc_no_short[order(A[,1])]
x_cc_no_short2 <- ifelse(is.na(x_cc_no_short2), 0, x_cc_no_short2)
rp_cc_no_short <- as.matrix(r2) %*% x_cc_no_short2
sd_cc_no_short <- sqrt(t(x_cc_no_short2) %*% covmat2 %*% x_cc_no_short2)
```

```r
# Market (S&P500) performance for the period 2017-01-01 to 2020-03-01:
plot(cumprod(1 + r_sp500), lwd = 2, type = "l", xlim = c(0, 36), ylim = c(1, 10),
main = "Time Plots", ylab = expression(R[p]), xlab = "Time")

# Multigroup Model Optimal portfolio performance for the period 2017-01-01 to 2020-03-01 - from (a):
points(cumprod(1 + rp_mgm), col="red", lwd = 2, type = "l")

# Equal Allocation portfolio performance for the period 2017-01-01 to 2020-03-01:
points(cumprod(1 + rp_equal), col="orange", lwd = 2, type = "l")

# Minimum Risk portfolio performance for the period 2017-01-01 to 2020-03-01:
points(cumprod(1 + rp_MRP), col="maroon", lwd = 2, type = "l")

# Optimal portfolio performance for the period 2017-01-01 to 2020-03-01:
points(cumprod(1 + rp_optimal), col="green", lwd = 2, type = "l")

# SIM Optimal portfolio performance for the period 2017-01-01 to 2020-03-01:
points(cumprod(1 + rp_sim_short), col="blue", lwd = 2, type = "l")
points(cumprod(1 + rp_sim_no_short), col="lightblue", lwd = 2, type = "l")

# Constant Correlation Model Optimal portfolio performance for the period 2017-01-01 to 2020-03-01:
points(cumprod(1 + rp_cc_short), col="pink", lwd = 2, type = "l")
points(cumprod(1 + rp_cc_no_short), col="yellow", lwd = 2, type = "l")

legend(x = "topleft",lty = 1:2,
legend = c("S&P500","Multigroup Model - Short Sales", "Equal Allocation",
"Minimum Risk Portfolio", "Optimal Portfolio", "SIM - Short Sales",
"SIM - No Short Sales", "Constant Correlation - Short Sales",
"Constant Correlation - No Short Sales"),
col = c("black", "red", "orange", "yellow", "green", "blue", "lightblue",
"pink", "maroon"), cex=0.53)
```



From my portfolios, it would see that the best performing model would be under CCM (Constant Correlation Model) with Short Sales Allowed and Risk Free Asset equal to 0.005. We can see the all our models follow the general trend for the market, except for CCM, Optimal Portfolio, and SIM (Single Index Model) with Short Sales Allowed and Risk Free Asset equal to 0.005.

b(ii))

```r
# Market mean and standard deviation:
mean_sp500 <- mean(r_sp500)
sd_sp500 <- sd(r_sp500)
# Combine Compositions:
X <- cbind(x, x_equal, x_MRP, x_optimal, x_sim_short2, x_sim_no_short2, x_cc_short2,
x_cc_no_short2)
# Beta of stocks:
covmat_all <- cov(r_SIM)
beta <- covmat_all[1,-1] / covmat_all[1,1]
# All portfolios:
portfolios <- c("Multigroup Model - Short Sales", "Equal Allocation", "Minimum Risk Portfolio",
"Optimal Portfolio", "SIM - Short Sales", "SIM - No Short Sales",
"Constant Correlation - Short Sales", "Constant Correlation - No Short Sales")
portfolio_mean <- c(mean(rp_mgm), mean(rp_equal), mean(rp_MRP),
                    mean(rp_optimal),mean(rp_sim_short), mean(rp_sim_no_short),
                    mean(rp_cc_short), mean(rp_cc_no_short))

portfolio_sd <- c(sd_mgm, sd_equal, sd_MRP, sd_optimal, sd_sim_short,
```

```r
                   sd_sim_no_short, sd_cc_short, sd_cc_no_short)

portfolio_beta <- t(X) %*% beta

Sharpe <- (portfolio_mean - rf) / portfolio_sd
differential <- portfolio_mean -
  (rf + ((mean_sp500 - rf) / sd_sp500)*portfolio_sd)
Treynor <- (portfolio_mean - rf) / portfolio_beta
Jensen <- portfolio_mean - (rf + (mean_sp500 - rf)*portfolio_beta)

df <- data.frame(Sharpe, differential, Treynor, Jensen)
row.names(df) <- portfolios

#
df
```

```
##                                          Sharpe differential     Treynor
## Multigroup Model - Short Sales        0.4025431  0.075622080 0.034272022
## Equal Allocation                      0.1052051  0.004935854 0.004792019
## Minimum Risk Portfolio                0.2074661  0.011747918 0.029761215
## Optimal Portfolio                     0.1836664  0.084382308 0.043664003
## SIM - Short Sales                     0.4914838  0.067827111 0.065589642
## SIM - No Short Sales                  0.2871448  0.016495078 0.015852231
## Constant Correlation - Short Sales    0.3364335  0.038539177 0.021752131
## Constant Correlation - No Short Sales 0.2751939  0.016135534 0.013867481
##                                            Jensen
## Multigroup Model - Short Sales        0.074093385
## Equal Allocation                      0.004899772
## Minimum Risk Portfolio                0.011118628
## Optimal Portfolio                     0.078463305
## SIM - Short Sales                     0.066274158
## SIM - No Short Sales                  0.016291824
## Constant Correlation - Short Sales    0.037915989
## Constant Correlation - No Short Sales 0.015999593
```

b(iii))

```r
mean_sim_no_short <- mean(rp_sim_no_short)
beta_2prime <- sd_sim_no_short / sd_sp500
mean_2prime <- rf + (mean_sp500 - rf)*beta_2prime
beta_sim_no_short <- t(x_sim_no_short2) %*% beta
mean_prime <- rf + (mean_sp500 - rf)*beta_sim_no_short
net_selectivity <- mean_sim_no_short - mean_2prime
net_selectivity
```

```
##            [,1]
## [1,] 0.01649508
```

```r
diversification <- mean_2prime - mean_prime
diversification
```

```
##               [,1]
## [1,] -0.0002032538
```

```r
plot(c(beta_sim_no_short, beta_2prime, 1, 0), c(mean_prime, mean_2prime, mean_sp500, rf),
type = "o", pch = 19,
main = "Fama Decomposition - SIM when short sales not allowed",
xlab = "Beta",
ylab = "Expected Return",
ylim = c(0,0.0105))
points(c(beta_sim_no_short, beta_sim_no_short), c(-1, mean_sim_no_short), type = "o",
pch = 19, lty = 2)
points(c(beta_2prime, beta_2prime), c(-1, mean_sim_no_short), type = "l", lty = 2)
points(c(-1, beta_2prime), c(mean_sim_no_short, mean_sim_no_short), type = "l", lty = 2)
points(c(-1, beta_2prime), c(mean_prime, mean_prime), type = "l", lty = 2)
points(c(beta_sim_no_short, beta_2prime), c(mean_2prime, mean_2prime), type = "l", lty = 2)
text(c(beta_sim_no_short, beta_2prime, 1, 0, beta_sim_no_short) - 0.02,
c(mean_prime, mean_2prime, mean_sp500, rf, mean_sim_no_short) + 0.0005,
labels = c("A'", "A''", "M", "Rf", "A"))
```