hw5-1-7

Anatolie Chernyakhovsky

December 8, 2020

Project: Portfolio allocation

The Excel file lecture6p.xlsx contains daily market data for VanEck Vectors Short Muni ETF, Handa Mining Corp, from 12/29/1989 to 8/31/2020, obtained from Yahoo Finance. The file also includes a daily risk-free rate time series from Kenneth French's Data Library. For this problem set, you should calculate time series of weekly returns.

Problem 1:

Construct weekly simple total returns from the price data (use Adj Close to include dividends). Compute and report the weekly and annualized mean and standard deviation for each stock. Compute the correlation matrix.

```
excelAddress = "C:/R/portfolio allocation/lecture6p.xlsx"
MSFT <- read_excel(excelAddress, sheet = "MSFT") %>% select(Date,AdjClose)
INTC <- read_excel(excelAddress, sheet = "INTC") %>% select(Date,AdjClose)
LUV <- read_excel(excelAddress, sheet = "LUV") %>% select(Date,AdjClose)
MCD <- read_excel(excelAddress, sheet = "MCD") %>% select(Date,AdjClose)
JNJ <-read_excel(excelAddress, sheet = "JNJ") %>% select(Date,AdjClose)
MSFT$Date <- as.Date(MSFT$Date)
INTC$Date <- as.Date(MSFT$Date)
LUV$Date <- as.Date(MSFT$Date)
MCD$Date <- as.Date(MSFT$Date)
JNJ$Date <- as.Date(MSFT$Date)
# Form a list including the date of every week
start_date = as.Date("1989-12-29")
end_date = as.Date("2020-08-31")
numOfWeek = as.numeric(end_date-start_date)%/%7
week_time = seq(start_date, end_date, "weeks")

MSFT_weekly = MSFT %>% filter(Date %in% week_time) %>% mutate(AdjClose_back = back(AdjClose)) %>%
  mutate(weeklyReturn =
           (AdjClose-AdjClose_back)/
           AdjClose_back)
INTC_weekly = INTC %>% filter(Date %in% week_time) %>% mutate(AdjClose_back = back(AdjClose)) %>%
  mutate(weeklyReturn = (AdjClose-AdjClose_back)/AdjClose_back)
LUV_weekly = LUV %>% filter(Date %in% week_time) %>% mutate(AdjClose_back = back(AdjClose)) %>%
  mutate(weeklyReturn = (AdjClose-AdjClose_back)/AdjClose_back)
MCD_weekly = MCD %>% filter(Date %in% week_time) %>% mutate(AdjClose_back = back(AdjClose)) %>%
  mutate(weeklyReturn = (AdjClose-AdjClose_back)/AdjClose_back)
```

```r
JNJ_weekly = JNJ %>% filter(Date %in% week_time) %>% mutate(AdjClose_back = back(AdjClose)) %>%
  mutate(weeklyReturn = (AdjClose-AdjClose_back)/AdjClose_back)
# The weekly return and standard deviation of each stock (on weekly basis)
MSFT_wk_rt = sum(MSFT_weekly$weeklyReturn, na.rm = TRUE)/numOfWeek
MSFT_wk_sd = sd(MSFT_weekly$weeklyReturn, na.rm = TRUE)
INTC_wk_rt = sum(INTC_weekly$weeklyReturn, na.rm = TRUE)/numOfWeek
INTC_wk_sd = sd(INTC_weekly$weeklyReturn, na.rm = TRUE)
LUV_wk_rt = sum(LUV_weekly$weeklyReturn, na.rm = TRUE)/numOfWeek
LUV_wk_sd = sd(LUV_weekly$weeklyReturn, na.rm = TRUE)
MCD_wk_rt = sum(MCD_weekly$weeklyReturn, na.rm = TRUE)/numOfWeek
MCD_wk_sd = sd(MCD_weekly$weeklyReturn, na.rm = TRUE)
JNJ_wk_rt = sum(JNJ_weekly$weeklyReturn, na.rm = TRUE)/numOfWeek
JNJ_wk_sd = sd(JNJ_weekly$weeklyReturn, na.rm = TRUE)

# The weekly return and standard deviation on annulized basis
MSFT_wk_rt_annu = MSFT_wk_rt*52
MSFT_wk_sd_annu = MSFT_wk_sd*sqrt(52)
INTC_wk_rt_annu = INTC_wk_rt*52
INTC_wk_sd_annu = INTC_wk_sd*sqrt(52)
LUV_wk_rt_annu = LUV_wk_rt*52
LUV_wk_sd_annu = LUV_wk_sd*sqrt(52)
MCD_wk_rt_annu = MCD_wk_rt*52
MCD_wk_sd_annu = MCD_wk_sd*sqrt(52)
JNJ_wk_rt_annu = JNJ_wk_rt*52
JNJ_wk_sd_annu = JNJ_wk_sd*sqrt(52)

agg_ret = data.frame(
  MSFT_wk_rt, MSFT_wk_rt_annu, MSFT_wk_sd,
  MSFT_wk_sd_annu,
  INTC_wk_rt, INTC_wk_rt_annu, INTC_wk_sd,
  INTC_wk_sd_annu,
  LUV_wk_rt, LUV_wk_rt_annu, LUV_wk_sd,
  LUV_wk_sd_annu,
  MCD_wk_rt, MCD_wk_rt_annu, MCD_wk_sd,
  MCD_wk_sd_annu,
  JNJ_wk_rt, JNJ_wk_rt_annu, JNJ_wk_sd,
  JNJ_wk_sd_annu)

# Combining all weekly return data in a single matrix to compute correlation

stock_data = data.frame(MSFT_weekly$weeklyReturn,INTC_weekly$weeklyReturn,
                        LUV_weekly$weeklyReturn, MCD_weekly$weeklyReturn,
                        JNJ_weekly$weeklyReturn) %>% na.omit()
corr_matrix = cor(stock_data)

# print the data we want:
agg_ret
```

```
##    MSFT_wk_rt MSFT_wk_rt_annu MSFT_wk_sd MSFT_wk_sd_annu  INTC_wk_rt
## 1 0.004828872       0.2511013 0.04160831       0.3000418 0.003937598
##   INTC_wk_rt_annu INTC_wk_sd INTC_wk_sd_annu   LUV_wk_rt LUV_wk_rt_annu
## 1       0.2047551 0.05055023       0.3645229 0.003439815      0.1788704
##    LUV_wk_sd LUV_wk_sd_annu   MCD_wk_rt MCD_wk_rt_annu  MCD_wk_sd
```

```
## 1 0.04791883      0.3455476 0.002880138      0.1497672 0.03210951
##   MCD_wk_sd_annu   JNJ_wk_rt JNJ_wk_rt_annu  JNJ_wk_sd JNJ_wk_sd_annu
## 1      0.2315449 0.002748644      0.1429295 0.02919812      0.2105506
```

```
corr_matrix
```

```
##                            MSFT_weekly.weeklyReturn INTC_weekly.weeklyReturn
## MSFT_weekly.weeklyReturn                  1.0000000                0.5125582
## INTC_weekly.weeklyReturn                  0.5125582                1.0000000
## LUV_weekly.weeklyReturn                   0.3100514                0.2915644
## MCD_weekly.weeklyReturn                   0.2880137                0.2648133
## JNJ_weekly.weeklyReturn                   0.2809817                0.2140466
##                            LUV_weekly.weeklyReturn MCD_weekly.weeklyReturn
## MSFT_weekly.weeklyReturn                 0.3100514               0.2880137
## INTC_weekly.weeklyReturn                 0.2915644               0.2648133
## LUV_weekly.weeklyReturn                  1.0000000               0.3011301
## MCD_weekly.weeklyReturn                  0.3011301               1.0000000
## JNJ_weekly.weeklyReturn                  0.2682334               0.3293791
##                            JNJ_weekly.weeklyReturn
## MSFT_weekly.weeklyReturn                 0.2809817
## INTC_weekly.weeklyReturn                 0.2140466
## LUV_weekly.weeklyReturn                  0.2682334
## MCD_weekly.weeklyReturn                  0.3293791
## JNJ_weekly.weeklyReturn                  1.0000000
```

Problem 2:

Construct the mean-variance frontier for the Intel-Microsoft combination. Indicate the minimum-variance portfolio and the efficient frontier (the efficient frontier is a set of expected returns - risks that you would want to consider investing in).

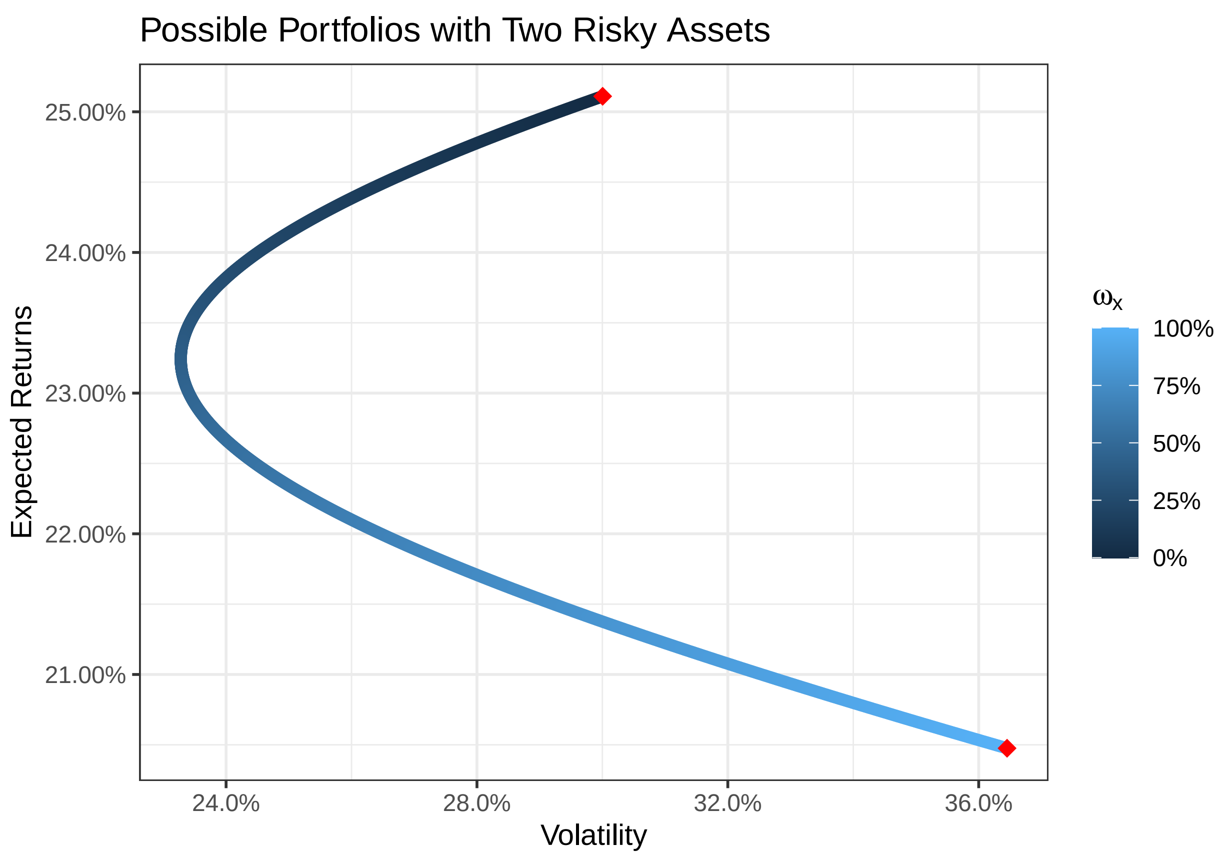
```
# I used "data.table"" package for this - see installation package above to receive.

# er here is 'expected ret'
#er_x = agg_ret$INTC_wk_rt_annu
#er_y = agg_ret$MSFT_wk_rt_annu
er_x = agg_ret$INTC_wk_rt_annu
er_y = agg_ret$MSFT_wk_rt_annu
sd_x = agg_ret$INTC_wk_sd_annu
sd_y = agg_ret$MSFT_wk_sd_annu

cov_xy = cov(na.omit(INTC_weekly$weeklyReturn),
             na.omit(MSFT_weekly$weeklyReturn))
x_weights = seq(from = 0, to = 1,
                length.out = 1000)
two_assets = data.table(wx = x_weights,
                        wy = 1 - x_weights)
# this converts two_assets from being arbitrary weights into useful data using our inputs.
two_assets[, ':=' (er_p = wx * er_x + wy * er_y,
                   sd_p =
                     sqrt(wx^2 * sd_x^2 +
                                 wy^2 * sd_y^2 +
```

```r
                                        2 * wx * (1 - wx) * cov_xy))]

ggplot() +
geom_point(data = two_assets, aes(x = sd_p, y = er_p, color = wx)) +
geom_point(data = data.table(sd = c(sd_x, sd_y), mean = c(er_x, er_y)),
aes(x = sd, y = mean), color = "red", size = 3, shape = 18) +
# Miscellaneous Formatting
theme_bw() + ggtitle("Possible Portfolios with Two Risky Assets") +
xlab("Volatility") + ylab("Expected Returns") +
scale_y_continuous(label = percent) +
scale_x_continuous(label = percent) +
scale_color_continuous(name = expression(omega[x]), labels = percent)
```




```r
# Code citation: https://www.r-bloggers.com/2016/05/a-gentle-introduction-to-finance-using-r-efficient-
```

Problem 3:

Add remaining stocks to the mix. Compute the mean-variance frontier and plot it on the same chart with the one from the previous question. Indicate the minimumvariance portfolio and the efficient frontier. How do they compare to those of the previous question?

```r
# note: correlation matrix is corr_matrix
stdevs = c(agg_ret$MSFT_wk_sd_annu, agg_ret$INTC_wk_sd_annu,
           agg_ret$LUV_wk_sd_annu, agg_ret$MCD_wk_sd_annu,
```

```
          agg_ret$JNJ_wk_sd_annu)
# 'b' is a 5*5 matrix whose generic term is stdev[i]*stdev[j]
b = stdevs %*% t(stdevs)
stockCov = b*corr_matrix
# Code citation: https://stackoverflow.com/questions/18740796/generate-covariance-matrix-from-correlati
colnames(stockCov) = c("MSFT","INTC","LUV",
                       "MCD", "JNJ")
msft_intc = stockCov[1,2]
msft_luv = stockCov[1,3]
msft_mcd = stockCov[1,4]
msft_jnj = stockCov[1,5]
intc_luv = stockCov[2,3]
intc_mcd = stockCov[2,4]
intc_jnj = stockCov[2,5]
luv_mcd = stockCov[3,4]
luv_jnj = stockCov[3,5]
mcd_jnj = stockCov[4,5]
# modifying above approach to fit 5 assets
five_assets = data.table(wx = rep(x_weights, each = length(x_weights)), wy = rep(x_weights, length(x_wei
five_assets[, wz:= 1 - wx - wy]
```