Assignment 2: Mean-Variance Optimal Risky Portfolio Using Simulation

Dana DiVincenzo
6/20/21

Introduction

After securities are selected for a portfolio, a portfolio manager needs to perform portfolio optimization. Following Modern Portfolio Theory, investors should construct portfolios to maximize expected returns based off of a given level of market risk. Thanks to technological advancements, it is possible to construct hundreds, thousands, or millions of potential portfolios in order to discover which ones are optimal given constraints.

In this assignment, I use the quantmod package in order to get data on 7 securities: GE, XOM, GBX, SBUX, PFE, HMC, and NVDA. Based on this data, along with data constraints of 1/1/2014 to 12/31/2017, and a risk rate constraint of 0, I create a function which creates hundreds of possible portfolios. The function then finds the most mean-variance optimal of those portfolios, and returns them, along with supporting information. Using the Sharpe Ratio, I am able to find which of those portfolios is the most optimal of all. Finally, I graphically display this information and discuss the results.

Step 1: Creating the Mean-Variance Portfolio Optimizing Function

```r
library(quantmod)
library(ggplot2)
library(dplyr)
library(knitr)
library(RColorBrewer)
```

```
# this function takes securities, begin and end dates, and a risk free rate
# it outputs the stock means, the covariance matrix of the stocks, and a dataframe of mean-variance
# optimal portfolios with supporting information
myMeanVarPort <- function(ticks, begin, end, risk_free_rate) {
  retout <- NULL
  retout <- xts(retout)

  # for each tick gets montly returns
  for(i in 1:length(ticks)){
    prices = getSymbols(ticks[i], auto.assign = F)
    returns <- periodReturn(prices, period = "monthly", type = "arithmetic")
    retout <- merge.xts(retout, returns)
  }
  # set col names
  colnames(retout) <- ticks
  # set time period
  retout <- retout[paste(begin, '/', end, sep = '')]
  # omit NAs
  retout <- na.omit(retout)

  # get stock means
  meanret <- colMeans(retout,na.rm = T)
  x1 = (round(meanret, 5))

  # get covariance matrix
  covar <- var(retout)
  x2 = (round(covar, 8))

  # creating random portfolio weights
  set.seed(12)
  niter <- 100*length(ticks)   # Set the number of iterations
  randomnums <- data.frame(replicate(length(ticks), runif(niter, 1, 10)))

  # normalize the weights
  wt_sim <- randomnums / rowSums(randomnums)

  # initialize weight and results variables
  weight <- matrix(data = NA, nrow = length(ticks), ncol = 1)
  Results <- matrix(data = NA, nrow = niter, ncol = length(ticks) + 2)

  # doing portfolio calculations for each simulated portfolio
  for (i in 1:niter){
      # inner loop places weights into Results
      for (k in 1:length(ticks)) {
               Results[i,k] = weight[k,1] = wt_sim[i,k]
      }
      Results[i,length(ticks) + 1] <- t(weight) %*% meanret                # portfolio mean
      Results[i,length(ticks) + 2] <- sqrt(t(weight) %*% covar %*% weight) # portfolio sigma
  }

  # gives us dataframe with each generated portfolios, showing weight of each portfolio, portfolio mean, and
 portfolio sigma
  colnames(Results) <- c(ticks, "PortMean", "PortSigma")
  Results <- as.data.frame(Results)
  Results$portfolio_ID <- 1:nrow(Results)

  # optimizing Results dataframe based on mean an variance
  minmret = min(Results$PortMean)
  maxmret = max(Results$PortMean)
  seqmret = seq(round(minmret,3)-.001, maxmret+.001, .001)

  optim <- Results %>% mutate(portnumber = index(Results)) %>%
      mutate(ints = cut(PortMean ,breaks = seqmret),
             lower = as.numeric( sub("\\((.+),.*", "\\1", ints) )) %>%
      group_by(ints) %>%
```

```
    summarise(portfolio_ID=portnumber[which.min(PortSigma)])

  optim <- as.data.frame(optim)

  # joining the optimized df with the results dataframe so weights are shown with optomized portfolios
  opt_info <- Results %>% inner_join(optim)
  opt_info <- opt_info %>% select (-c(ints))

  # calculating the sharpe ratio of each optimized portfolio and adding it as a feature in the optomized df
  opt_info$sharpe_ratio <- (opt_info$PortMean - risk_free_rate) / opt_info$PortSigma

  opt_info <- as.data.frame(opt_info)

  # the output of this function is a vector of the stock means, the covariance matrix, and the optimized resu
lts df
  list(stock_means = x1, cov_matrix = x2, opt_info = opt_info)
}
```

Step 2: Running the Function with Required Inputs and Showing Results

```
# list of ticks used
ticks <- c('GE','XOM','GBX','SBUX','PFE','HMC','NVDA')
final_list <- myMeanVarPort(ticks, 20140101, 20171231, 0)
```

```
summary(final_list)
```

```
##             Length Class      Mode
## stock_means  7     -none-     numeric
## cov_matrix  49     -none-     numeric
## opt_info    11     data.frame list
```

```
final_list$stock_means
```

```
##       GE      XOM      GBX     SBUX      PFE      HMC     NVDA
## -0.00837 -0.00314  0.01536  0.00902  0.00442 -0.00246  0.05817
```

```
final_list$cov_matrix
```

```
##              GE        XOM        GBX       SBUX        PFE        HMC
## GE   0.00294255 0.00103470 0.00121121 0.00080728 0.00064653 0.00062168
## XOM  0.00103470 0.00168517 0.00162533 0.00023608 0.00053446 0.00020432
## GBX  0.00121121 0.00162533 0.01036163 0.00006399 0.00157675 0.00212741
## SBUX 0.00080728 0.00023608 0.00006399 0.00211533 0.00052364 0.00068028
## PFE  0.00064653 0.00053446 0.00157675 0.00052364 0.00188741 0.00083543
## HMC  0.00062168 0.00020432 0.00212741 0.00068028 0.00083543 0.00316065
## NVDA 0.00135432 0.00109161 0.00318265 0.00036711 0.00036748 0.00116384
##            NVDA
## GE   0.00135432
## XOM  0.00109161
## GBX  0.00318265
## SBUX 0.00036711
## PFE  0.00036748
## HMC  0.00116384
## NVDA 0.01112564
```

```
head(final_list$opt_info)
```

```
##           GE       XOM        GBX      SBUX       PFE        HMC       NVDA
## 1 0.05626487 0.2246457 0.06018307 0.1510301 0.2730653 0.06200091 0.17281000
## 2 0.03739068 0.1444033 0.04193202 0.1974942 0.2627637 0.07678379 0.23923234
## 3 0.24824290 0.2337927 0.04790202 0.1992185 0.1159321 0.10767234 0.04723949
## 4 0.10386410 0.2108330 0.02436894 0.1618174 0.2042215 0.11997669 0.17491844
## 5 0.11208103 0.1911952 0.04432895 0.2524876 0.1777203 0.13497125 0.08721570
## 6 0.09270160 0.2071430 0.08273455 0.1791743 0.2124571 0.12015798 0.10563141
##      PortMean  PortSigma portfolio_ID sharpe_ratio
## 1 0.012215574 0.03574835           20   0.34171017
## 2 0.016546006 0.03863194           48   0.42829864
## 3 0.002715558 0.03275405           76   0.08290753
## 4 0.011083293 0.03502610           89   0.31642954
## 5 0.006945810 0.03204797          190   0.21673167
## 6 0.008247531 0.03387608          195   0.24346175
```

Step 3: Explaining Function Output

The function above outputs a list of: (1) a vector of the stock means, (2) the covaraince matrix of the stocks, and (3) the optimized portfolio dataframe.

1. The vector of stock means tells us the average returns of each stock over the given time period
2. The covariance matrix tells us how strongly (positively or negatively) each stock relates to one another
3. The optimized portfolio dataframe holds information on the subset of hypothetical portfolios which best maximize returns and limit risk
    - The weight of each stock
    - The portfolio mean
    - The portfolio risk
    - The Sharpe Ratio (higher value = more optimized)
    - The portfolio ID

Step 4: Finding the Optimal Portfolio

The portfolio with the highest Sharpe Ratio is the most optimal. The Sharpe Ratio was calculated for each portfolio in the myMeanVarPort function. The Sharpe Ratio is defined as the difference between the portfolio return and risk free rate, divided by the portfolio sigma.

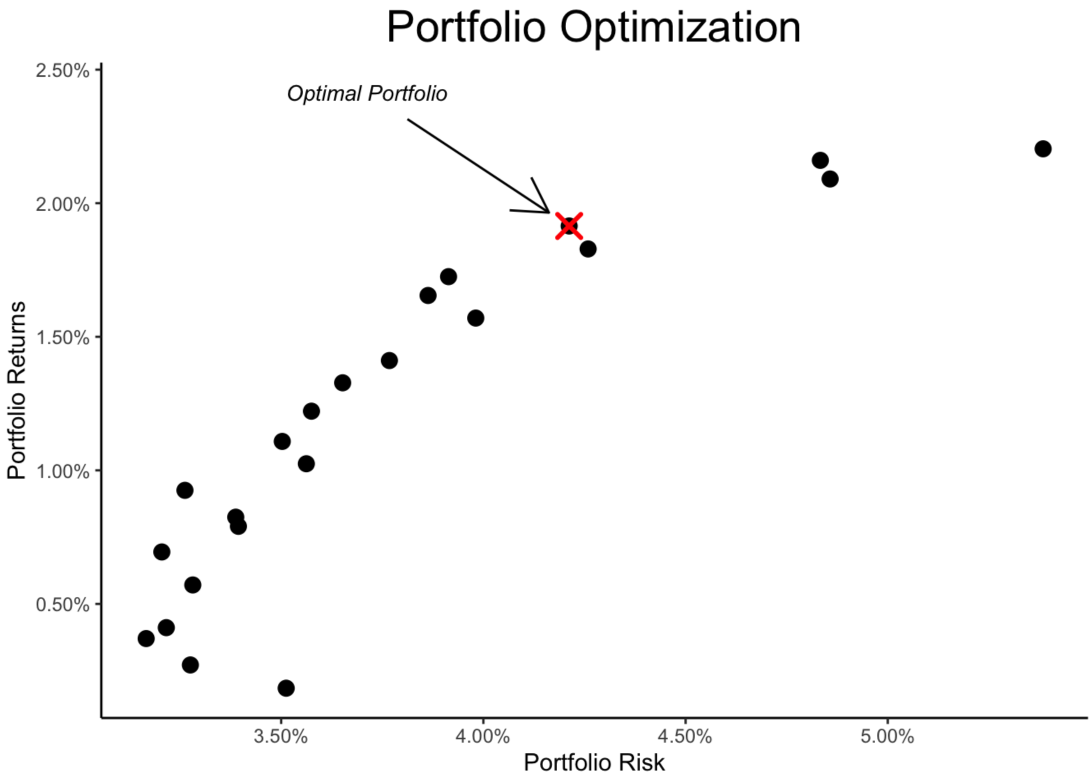
```
# pulling only the optimized portfolio info out of the results from the above function
opt <- final_list$opt_info

# pulling the portfolio with the highest sharpe ratio out
max_sharpe <- opt[which.max(opt$sharpe_ratio),]
```

Step 5: Visualizing Results

The graph below shows a plot of the most optimal portfolios. The portfolio which had the highest Sharpe Ratio is pointed out. Data about that portfolio is also displayed below the graph.

```
# plotting the output, emphasizing the portfolio with the highest sharpe ratio
  ggplot(data = opt , aes(x = PortSigma, y = PortMean)) +
    geom_point(colour = "black", size = 3) +
    annotate('text', x = max_sharpe$PortSigma - .005, y = max_sharpe$PortMean + .005,
             label = "Optimal Portfolio", fontface = 'italic', size = 3.5) +
    annotate(geom = 'segment', x = max_sharpe$PortSigma - .004, xend = max_sharpe$PortSigma - .0005,  y = max
_sharpe$PortMean + .004,
             yend = max_sharpe$PortMean + .0005, color = 'black', arrow = arrow(type = "open")) +
    labs(x = 'Portfolio Risk',
       y = 'Portfolio Returns',
       title = "Portfolio Optimization") +
    theme(plot.title = element_text(hjust = 0.5, size = 20), panel.grid.major = element_blank(), panel.grid.m
inor = element_blank(),
          panel.background = element_blank(), axis.line = element_line(colour = "black")) +
    scale_y_continuous(labels = scales::percent) +
    scale_x_continuous(labels = scales::percent) +
    geom_point(data = max_sharpe, aes(x = PortSigma, y = PortMean), shape = 4, colour = "red", size = 4, stro
ke = 1.5)
```



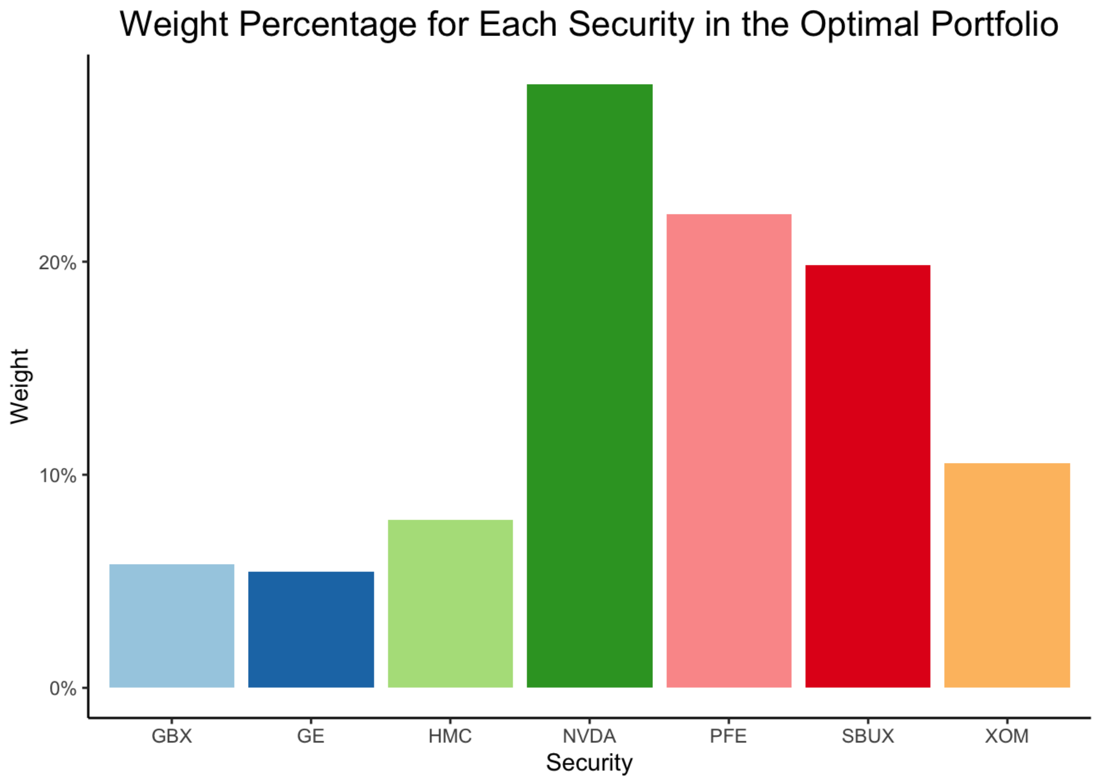

```
kable(max_sharpe, align = "lllll", caption = "Optimal Portfolio Data.")
```

Optimal Portfolio Data.

	GE	XOM	GBX	SBUX	PFE	HMC	NVDA	PortMean	PortSigma	portfolio_ID	sharpe_ratio
12	0.0544097	0.1053346	0.057912	0.1985004	0.2221315	0.0786257	0.283086	0.0191477	0.0421212	356	0.4545844

```
# prepping optimal portfolio data for ggplot barplot
max_sharpe_t <- max_sharpe %>% select (-c('PortMean', 'PortSigma',  'portfolio_ID', 'sharpe_ratio'))
max_sharpe_t <- t(max_sharpe_t)
max_sharpe_t <- as.data.frame(max_sharpe_t)
max_sharpe_t <- cbind(name = rownames(max_sharpe_t), max_sharpe_t)
colnames(max_sharpe_t) <- c('Security', 'Weight')
```

```
ggplot(data = max_sharpe_t) + geom_col(aes(x = Security, y =Weight, fill = Security)) +
    scale_fill_brewer(palette="Paired") +
    labs(title = "Weight Percentage for Each Security in the Optimal Portfolio") +
    theme(legend.position="none", plot.title = element_text(hjust = 0.5, size = 16), panel.grid.major = eleme
nt_blank(), panel.grid.minor = element_blank(),
          panel.background = element_blank(), axis.line = element_line(colour = "black")) +
    scale_y_continuous(labels = scales::percent)
```



Step 6: Analyzing Results

The portfolio with the highest Sharpe Ratio has a mean return of 1.91%, and a risk of 4.21%. Looking at its place on the graph relative to the other optimized portfolios, this portfolio has one of the highest return rates, but also one of the highest risk rates. We can see that there are possible portfolios further right on the graph which would have given us higher returns. However, selecting those means that we would have to be comfortable with a higher level of risk.

Below the graph we can also see the weight breakdown of each security in the optimal portfolio, both in a table and in a bar chart. In descending order by weight, the securities in this portfolio are NVDA, PFE, SBUX, XOM, HMC, GBX, and GE. Let's look at the context around NVDA, PFE, and SBUX, the 3 stocks with the highest weights.

NVDA, or NVIDIA is a visual computing technology company. They are worldwide leaders in this sector, and their stock has shown steady positive growth over the years. Between the dates 1/1/2014 and 12/31/2017, the stock increased by about 1130%, and had an ending value of $34.31. This is huge relative growth, and therefore it is no surprise that NVDA is the stock with the highest weight here. PFE, or Pfizer, has had recent fame during the COVID19 crisis. However, before COVID, Pfizer was still a successful pharmaceudical company, which showed stock growth of about 21% between 1/1/2014 and 12/31/2017, and had an ending value of $193.50. In that same time period, SBUX, or Starbucks, showed growth of about 59%, and had an ending value of $58.29.

Next, we can look at the stock with the smallest weight, GE, or General Electric, a multinational conglomerate operating in many different industries. From 1/1/2014 to 12/31/2017, GE's stock value decreased about 35.15%, and its ending value was $17.82. This decrease in value may be why this stock has the lowest weight in this portfolio.

Although stock analysis is not as simple as looking at these 2 values, it gives us a good idea of the success or failures of these stocks during this time.
* *Stock information in this section was gathered from Google Finance.*

Conclusions

Although the portfolio optimization method shown here is simple, it effectively found the mean-variable optimal portfolio, and captured important information related to that portfolio. Additionally, the function used here could use any number of securities, with different beginning and end dates, and varying risk rates. Therefore, this function could be used over and over again to find optimal portfolios. The function was designed to create 100 * (N = Number of ticks) portfolios before optimizing, however, with a few simple tweaks it could be used to find 1000N or 1000000N portfolios, and the only difference would be processing time. This would give the user more portfolio options to analyze. Other optimization constrains could also be added in by a more financially experienced analyst, to make this function more powerful.