STOCK MARKET CASE:

MV PORTFOLIO OPTIMIZATION

Group D3

Alice Im
Melissa Tran
Sandeep Kumar Prasad
Venkat Naren Pagadala

ISDS 570
Dr. Pawel Kalczynski
California State University, Fullerton

Table of Contents

Introduction

In the stock market case, our portfolio consists of the 12 following tickers traded in 3 markets:

***Table 1**: Selected Tickers*

Market	**Tickers**
NASDAQ	IDXX, IRMD, PLAY, PLUS, PLUG, PZZA, TWIN, TOWN, TURN
NYSE	PWR, PLOW
OTCMKTS	IFJPY

Some of these tickers were chosen randomly, but most were chosen for their easily pronounceable acronyms. Our tickers were verified to have sufficient historical data by excluding tickers with incomplete data before making our choices. To optimize a portfolio that would outperform the SP500, we utilized PG Admin and R Studio to examine the historical data from 2016 to 2020 that we had available. This data was run through an ETL process, and the results are obtained from R Studio.

ETL Process

Extraction

The Quandl wiki database was utilized for this project; it provides stock prices and investment returns for publicly traded companies. The data for the SP500TR was obtained from finance.yahoo.com. The raw data was first downloaded from the Quandl wiki and then transferred to PostgreSQL. The data was then cleaned before being transferred to R as a clean file with daily references and a custom calendar. The data was then downloaded in .csv format once we picked the time frame.

Data transformation

We enforce data cleansing, quality control, and data transformation into standard formats in this stage. All required information regarding the SP500TR and stock market tickers and a custom calendar is accessible. We now import the data into RStudio, where we will do our data transformations such as making sure tickers with missing data are not used. We utilize the RStudio packages such as 'PerformanceAnalytics' and 'PortfolioAnalytics' to import all the data to create authenticated connections with PostgreSQL. The SP500TR and stock ticker data are then combined.

Historical Data

With all the information on daily returns from each ticker, we created a cumulative return chart to view how our selected stocks performed during the historical period from January 2016 to December 2020. In reviewing the cumulative returns, our diverse choices across various industries offered us a wide range of returns. PLUG had the highest cumulative return of 15.07 while PLAY had the lowest return at -0.28, and since we have some stocks with negative returns, we will be opting to shorten a portion of our portfolio.

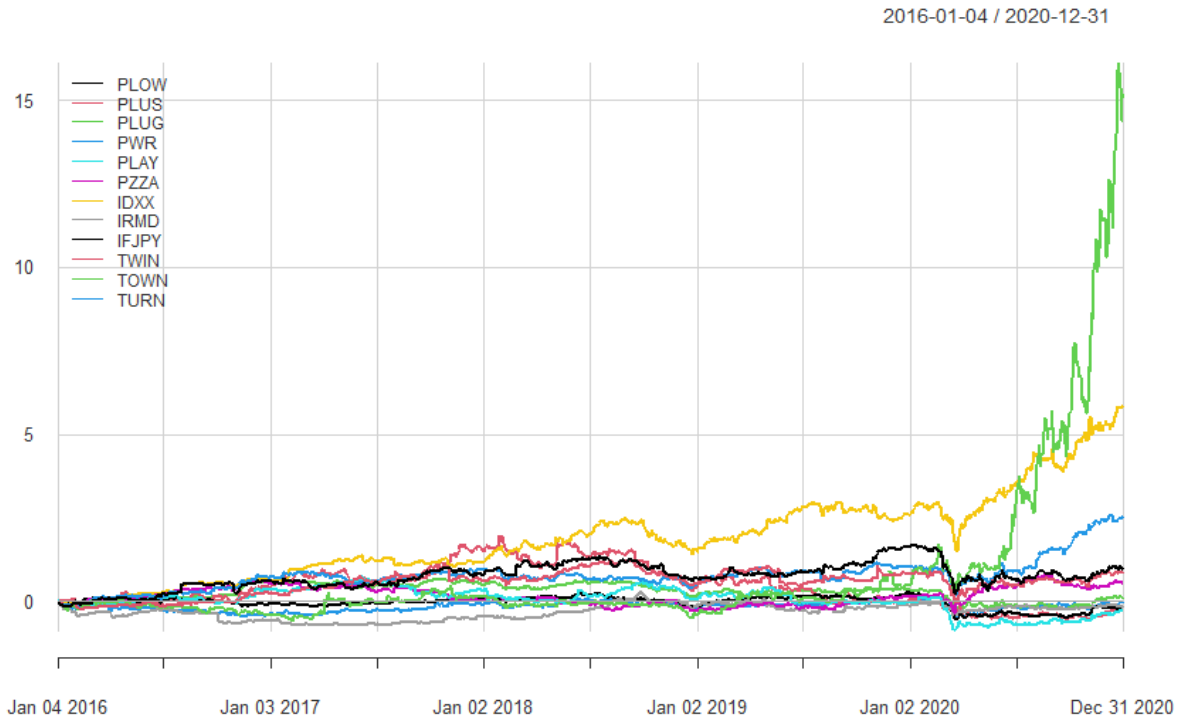

***Figure 1**: Cumulative return line chart 2016 – 2020*

	PLOW	PLUS	PLUG	PWR	PLAY	PZZA	IDXX	IRMD	IFJPY	TWIN
Cumulative Return	1.0299	0.8861248	15.07109	2.556543	-0.2807858	0.5187041	5.855047	-0.1865858	-0.1915181	-0.2538023

	TOWN	TURN
Cumulative Return	0.1250599	0.009090909

***Figure 2**: Cumulative return values 2016 – 2020*

Optimization

In creating our portfolio, we needed to choose the optimal weights to assign to each ticker. By applying functions from the Portfolio Analytics package of R, we were able to determine the optimal weights listed here. We should shorten 4 stocks i.e., PLAY, PLUS, PLUG, and TWIN because their values were predicted to fall. The sum of these optimal weights is 1 because they are percentages in our portfolio.

```
> opt_w
        PLOW         PLUS         PLUG          PWR         PLAY         PZZA         IDXX         IRMD        IFJPY
 0.050921107 -0.003316408 -0.014465073  0.091781315 -0.087608400  0.177642322  0.209130854  0.056606397  0.174164049
        TWIN         TOWN         TURN
-0.015929826  0.179977458  0.181096204
> sum(opt_w)
[1] 1
```

***Figure 3**: Optimal weights of the tickers*

Performance

Now that we have the tickers and their applied weights, we developed a forecast to predict how our portfolio would perform and constructed a new cumulative chart to compare it against the SP500. Since they appear to perform similarly in the chart, we will look at the numerical data next.

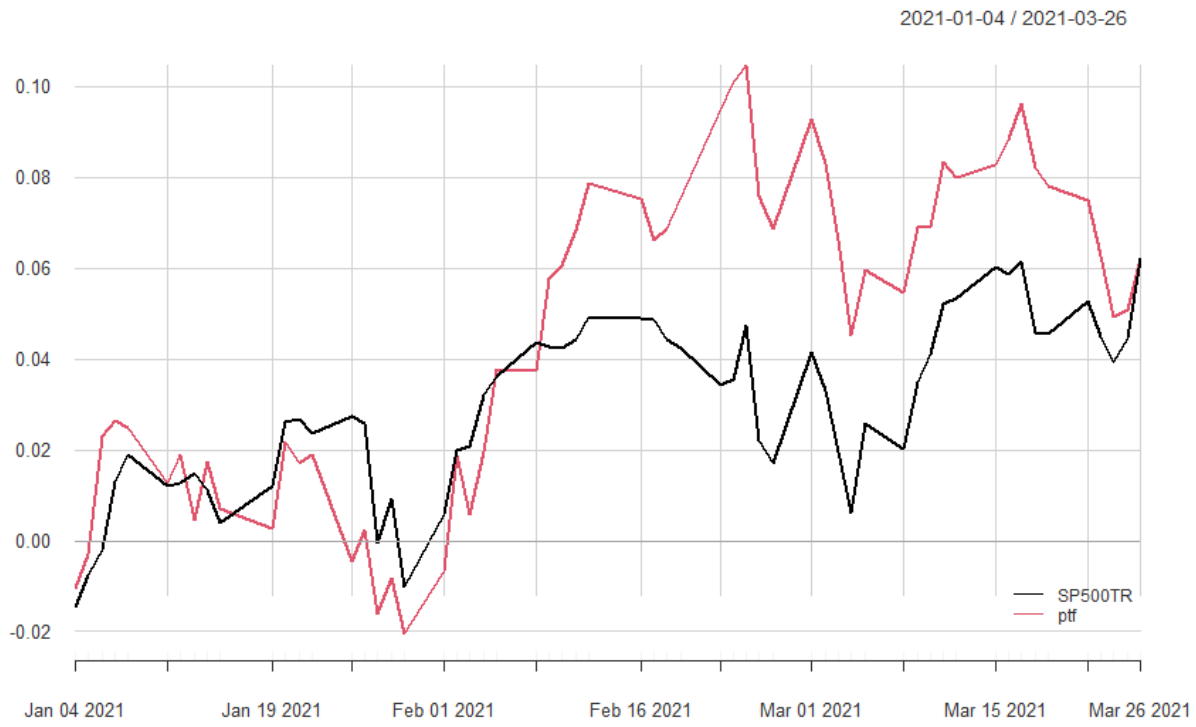

***Figure 4**: Cumulative return chart for your optimized portfolio (01 Jan 2021 - 26th March 2021) and SP500TR*

```
> Return.cumulative(Rp)
                    SP500TR       ptf
Cumulative Return 0.06199359 0.06229073
```

***Figure 5**: Cumulative return for optimized portfolio and SP500TR*

```
                              SP500TR    ptf
Annualized Return              0.2987 0.3002
Annualized Std Dev             0.1623 0.1960
Annualized Sharpe (Rf=0%)      1.8399 1.5315
```

***Figure 6**: Annualized return for SP500TR and Portfolio (01 Jan 2021 - 26th March 2021)*

On taking a closer look at the numbers, our portfolio's cumulative return of 0.0623 and annualized return of 0.3002 is just above the SP500's returns. However, it has greater variability due to its higher standard deviation. While the Sharpe ratio is lower, it is still high enough to be deemed as an acceptable Sharpe ratio.

```
> table.AnnualizedReturns(cbind(Ra_testing,Rb_testing),scale=252)
                           PLOW   PLUS   PLUG    PWR    PLAY   PZZA    IDXX   IRMD  IFJPY   TWIN   TOWN   TURN
Annualized Return        0.3969 0.9210 0.0155 1.2887  7.4794 0.1961 -0.1599 0.1531 0.2012 0.8376 2.5111 0.5529
Annualized Std Dev       0.4143 0.3654 1.4066 0.3692  0.6036 0.4735  0.3601 0.4353 0.3795 0.6541 0.3736 0.2522
Annualized Sharpe (Rf=0%) 0.9580 2.5205 0.0110 3.4902 12.3917 0.4141 -0.4440 0.3518 0.5302 1.2805 6.7220 2.1926
                         SP500TR
Annualized Return         0.2987
Annualized Std Dev        0.1623
Annualized Sharpe (Rf=0%) 1.8399
> |
```

Figure 7: *Annualized return for SP500TR and individually selected tickers (01 Jan 2021 - 26th March 2021)*

Appendix

R Codes:

```
rm(list=ls(all=T))
require(RPostgres) # require package installation
require(DBI)
conn <- dbConnect(RPostgres::Postgres()
                  ,user="stockmarketreader"
                  ,password="read123"
                  ,host="localhost"
                  ,port=5432
                  ,dbname="stockmarket"
)

#Retrieve data from custom calendar
qry<-"SELECT * FROM custom_calendar WHERE date BETWEEN '2015-12-31'
AND '2021-03-26'ORDER by date"
ccal<-dbGetQuery(conn,qry)
#eod prices and indices
qry1="SELECT symbol,date,adj_close FROM eod_indices WHERE date BETWEEN
'2015-12-31' AND '2021-03-26'"
qry2="SELECT ticker,date,adj_close FROM eod_quotes WHERE date BETWEEN
'2015-12-31' AND '2021-03-26'"
eod<-dbGetQuery(conn,paste(qry1,'UNION',qry2))
dbDisconnect(conn)
rm(conn)

head(eod[which(eod$symbol=='SP500TR'),])
tail(eod[which(eod$symbol=='SP500TR'),])
```

```
# Use Calendar ------------------------------------------------
tdays<-ccal[which(ccal$trading==1),,drop=F]
head(tdays)
nrow(tdays)-1

# Completeness ------------------------------------------------
# Percentage of completeness
pct<-table(eod$symbol)/(nrow(tdays)-1)
selected_symbols_daily<-names(pct)[which(pct>=0.99)]
eod_complete<-eod[which(eod$symbol %in%
selected_symbols_daily),,drop=F]

# Transform (Pivot) -------------------------------------------
require(reshape2) #did you install this package?
eod_pvt<-dcast(eod_complete, date ~
symbol,value.var='adj_close',fun.aggregate = mean, fill=NULL)

# Merge with Calendar -----------------------------------------
eod_pvt_complete<-
merge.data.frame(x=tdays[,'date',drop=F],y=eod_pvt,by='date',all.x=T)

#use dates as row names and remove the date column
rownames(eod_pvt_complete)<-eod_pvt_complete$date
eod_pvt_complete$date<-NULL #remove the "date" column


# Missing Data Imputation -------------------------------------
# We can replace a few missing (NA or NaN) data items with previous
data
# Let's say no more than 3 in a row...
require(zoo)
eod_pvt_complete<-
na.locf(eod_pvt_complete,na.rm=F,fromLast=F,maxgap=3)

# Calculating Returns -----------------------------------------
require(PerformanceAnalytics)
eod_ret<-CalculateReturns(eod_pvt_complete)
#remove the first row
eod_ret<-tail(eod_ret,-1) #use tail with a negative value

# Check for extreme returns -----------------------------------
# There is colSums, colMeans but no colMax so we need to create it
colMax <- function(data) sapply(data, max, na.rm = TRUE)
# Apply it
max_daily_ret<-colMax(eod_ret)
max_daily_ret[1:10] #first 10 max returns
# And proceed just like we did with percentage (completeness)
selected_symbols_daily<-
names(max_daily_ret)[which(max_daily_ret<=1.00)]
length(selected_symbols_daily)
#subset eod_ret
```

```
eod_ret<-eod_ret[,which(colnames(eod_ret) %in%
selected_symbols_daily),drop=F]

# Tabular Return Data Analytics ---------------------------------
# We will select 'SP500TR' and 12 TICKERS
random12<-
c("PLOW","PLUS","PLUG","PWR","PLAY","PZZA","IDXX","IRMD","IFJPY","TWIN
","TOWN","TURN")
# We need to convert data frames to xts (extensible time series)
Ra<-as.xts(eod_ret[,random12,drop=F])
Rb<-as.xts(eod_ret[,'SP500TR',drop=F]) #benchmark
Ra1<-Ra[1:1259,]
nrow(Ra)
nrow(Ra1)

# And now we can use the analytical package...
# Stats
table.Stats(Ra1)

# Distributions
table.Distributions(Ra1)

# Returns
table.AnnualizedReturns(cbind(Rb,Ra1),scale=252) # note for monthly
use scale=12

# Accumulate Returns
acc_Ra<-Return.cumulative(Ra1)
acc_Rb<-Return.cumulative(Rb)

# Capital Assets Pricing Model
table.CAPM(Ra1,Rb)

# Graphical Return Data Analytics -------------------------
# Cumulative returns chart
chart.CumReturns(Ra1,legend.loc = 'topleft')
chart.CumReturns(Rb,legend.loc = 'topleft')
chart.Drawdown(Ra,legend.loc = 'bottomleft')

# MV Portfolio Optimization --------------------------------
# withhold the last 253 trading days
Ra_training<-head(Ra,-58)
Rb_training<-head(Rb,-58)

# use the last 253 trading days for testing
Ra_testing<-tail(Ra,58)
Rb_testing<-tail(Rb,58)

#optimize the MV (Markowitz 1950s) portfolio weights based on training
table.AnnualizedReturns(Rb_training)
table.AnnualizedReturns(Ra_training)
mar<-mean(Rb_training) #we need daily minimum acceptable return
```

```
require(PortfolioAnalytics)
require(ROI) # make sure to install it
require(ROI.plugin.quadprog)  # make sure to install it
pspec<-portfolio.spec(assets=colnames(Ra_training))
pspec<-add.objective(portfolio=pspec,type="risk",name='StdDev')
pspec<-add.constraint(portfolio=pspec,type="full_investment")
pspec<-add.constraint(portfolio=pspec,type="return",return_target=mar)

#optimize portfolio
opt_p<-
optimize.portfolio(R=Ra_training,portfolio=pspec,optimize_method =
'ROI')

#extract weights (negative weights means shorting)
opt_w<-opt_p$weights

#apply weights to test returns
Rp<-Rb_testing # easier to apply the existing structure
#define new column that is the dot product of the two vectors
Rp$ptf<-Ra_testing %*% opt_w

#Compare basic metrics
table.AnnualizedReturns(Rp)

# Chart Hypothetical Portfolio Returns
chart.CumReturns(Rp,legend.loc = 'bottomright')
```