FINC-780 Final Exam

Tayo Obafaiye

03/12/2020

Introduction

The first question has to do with Portfolio Optimization. We'll be using the Portfolio Analytics Package in solving this question. The said package is useful for solving portfolio optimization either mathematically or by simulation.

The focus in the second question is Ordinary Least Squares (OLS) Regression. A typical use case of regression models in finance is the Single Index Model/Market Model which connects stock returns with market returns. In this question, we will be exploring the relationship between stock returns and earnings by estimating a regression.

```
library(PortfolioAnalytics)
```

```
## Loading required package: zoo
```

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

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

```
## Loading required package: xts
```

```
## Loading required package: foreach
```

```
## Loading required package: PerformanceAnalytics
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(quantmod)
```

```
## Loading required package: TTR
```

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

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(reshape2)
```

Question 1

The portfolio in this question is formed with six stocks. Our aim/objective is to optimize our portfolio along two dimensions. The first is to minimize variance and the second is to minimize expected tail loss (ETL). What is ETL?

There usually has to be at least one constraint as we frame the problem. This portfolio is set with two constraints, one being weights/leverage adds up to one (which suggests the portfolio is fully invested) and the other being a box constraint minimum of 5% and maximum of 30% for each asset (stock). Applied in this question is the best practice of min weight sum of 0.99 and max. of 1.01 to allow for wiggle room.

We access daily stock return data for the six assets using the quantmod package. The optimization method used in this question is ROI which is the numerical method of the Portfolio Analytics package. We also independently calculate objective measure to verify the optimal portfolio output.

```r
stocks <- c("CBRL", "XRX", "IBM", "NFLX", "CAT", "LRCX")
stocks <- as.matrix(stocks)
r <- lapply(stocks, function(x) dailyReturn(na.omit(getSymbols(x, auto.assign=FALSE))))
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
returns <- do.call(merge, r)
returns <- returns['2017/2018']
colnames(returns) <- c(stocks)

myport <- portfolio.spec(assets = stocks)
myport <- add.constraint(myport, type="weight_sum",
                         min_sum=0.99, max_sum=1.01)
myport <- add.constraint(myport, type="box", min=0.05, max=0.3)
#myport
minvar <- add.objective(portfolio=myport, type="risk", name="var")
opt_minvar <- optimize.portfolio(R=returns, portfolio=minvar,
                                 optimize_method="ROI",trace=TRUE)
```

```
## Registered S3 method overwritten by 'ROI':
##   method           from
##   print.constraint PortfolioAnalytics
```

```
#opt_minvar
opt_minvar$weights
```

```
##      CBRL       XRX       IBM      NFLX       CAT      LRCX
## 0.3000000 0.1420778 0.3000000 0.0500000 0.1479222 0.0500000
```

```
opt_minvar$objective_measures$StdDev
```

```
##     StdDev
## 0.00964179
```

```
# StdDev Objective Measure
var_port <- as.matrix(returns) %*% as.matrix(opt_minvar$weights)
sd(var_port)
```

```
## [1] 0.00964179
```

```
etl <- add.objective(portfolio=myport, type="risk", name="ETL")
opt_etl <- optimize.portfolio(R=returns, portfolio=etl,
                              optimize_method="ROI", trace=TRUE)
#opt_etl
opt_etl$weights
```

```
##      CBRL       XRX       IBM      NFLX       CAT      LRCX
## 0.3000000 0.1032536 0.3000000 0.0500000 0.1867464 0.0500000
```

```
opt_etl$objective_measures$ETL
```

```
## [1] 0.02473619
```

```
# ETL Objective Measure
etl_port <- as.matrix(returns) %*% as.matrix(opt_etl$weights)
etl_port <- as.data.frame(etl_port)
etl_port <- etl_port[order(etl_port$V1), , drop = FALSE]
subetl <- etl_port[1:ceiling(nrow(etl_port)*.05),]
mean(subetl)
```

```
## [1] -0.02453594
```

Discussion of Results

From the first optimization that minimizes variance, we get optimal weights “CBRL” (30%), “XRX” (14.21%), “IBM” (30%), “NFLX” (5%), “CAT” (14.79%) and “LRCX” (5%). Thus 30% of the portfolio should be in the “CBRL” asset as well as 30% in “IBM”. 5% of the portfolio each for “NFLX” and “LRCX” and 14.79% of the portfolio in “CAT”. The objective measure output is standard deviation with a value of 0.009642.

From the second optimization that minimizes ETL, the minimum values for the box are in securities NFLX and LRCX. CBRL and IBM have a lower risk level than the other securities. The objective measure output is ETL with a value of 0.02474. We can expect a loss or negative returns of 0.02474.

Question 2

We start by reading in “Compustat data small 2000-17.csv” file. We then use the unique function to get an array of the unique tickers in the dataset. Again we use the quantmod package to obtain the levels of the tickers in our data set, and to calculate the returns associated with those levels. In this case we calculate the yearly returns and subset the return data for years 2014 to 2016 to obtain a time series object with observations on the twelve stocks. We extract the Year from the index and convert the “xts” object to a data frame. Finally we convert the data frame from wide form to long form ahead of merging the data. The long form data frame consists of the Year (fyear), tickers(tic) and Stock Return values.

From the Compustat data set, we create a new data frame subset with dplyr by selecting columns “fyear”, “tic” and “ebit”, renaming “ebit” as “Earnings” and filtering “fyear” for 2014 - 2016.

Now that we have two data frames, we proceed to merge them both by “tic” and by “fyear”. At this point we have our data points for the regression model. We use the lm() function to run the regreesion. Our regression formula is of the form, Stock_Return ~ Earnings. From the compustat dataset, the column ebit, Earnings Before Interest and Taxes, is used as Earnings in the formula.

```
myfile <- "C:/Users/Tayo Obafaiye/Desktop/FINC780/FINC 780 Data/Compustat data small 2000-17.
csv"
mydat <- read.csv(myfile, header = T, sep = ",")
ticks <- unique(mydat$tic)
retout <- NULL
retout <- xts(retout)
for(i in 1:length(ticks)){
  prices = getSymbols(ticks[i], auto.assign = F)
  returns <- periodReturn(prices, period = "yearly",
                          type = "arithmetic")
  retout <- merge.xts(retout, returns)
}
colnames(retout) <- ticks
retout <- retout['2014/2016']
retout <- na.omit(retout)
retout$Year  <- format(index(retout),"%Y")
df1 <- as.data.frame(retout)
long_df1 <- melt(df1, id.vars = "Year")
colnames(long_df1) <- c("fyear", "tic", "Stock_Return")

df2 <- mydat %>% select(fyear, tic, ebit) %>% rename("Earnings" = "ebit") %>% filter(fyear ==
2014 | fyear == 2015 | fyear == 2016) #df2 <- na.omit(df2)

finaldf <- merge(long_df1, df2, by = c('tic', 'fyear'))
head(finaldf)
```

```
##    tic fyear Stock_Return Earnings
## 1 AAPL  2014   0.37724144    52503
## 2 AAPL  2015  -0.04638514    71230
## 3 AAPL  2016   0.10032297    59476
## 4 AMAT  2014   0.40950226     1598
## 5 AMAT  2015  -0.25080257     1692
## 6 AMAT  2016   0.72844135     2159
```

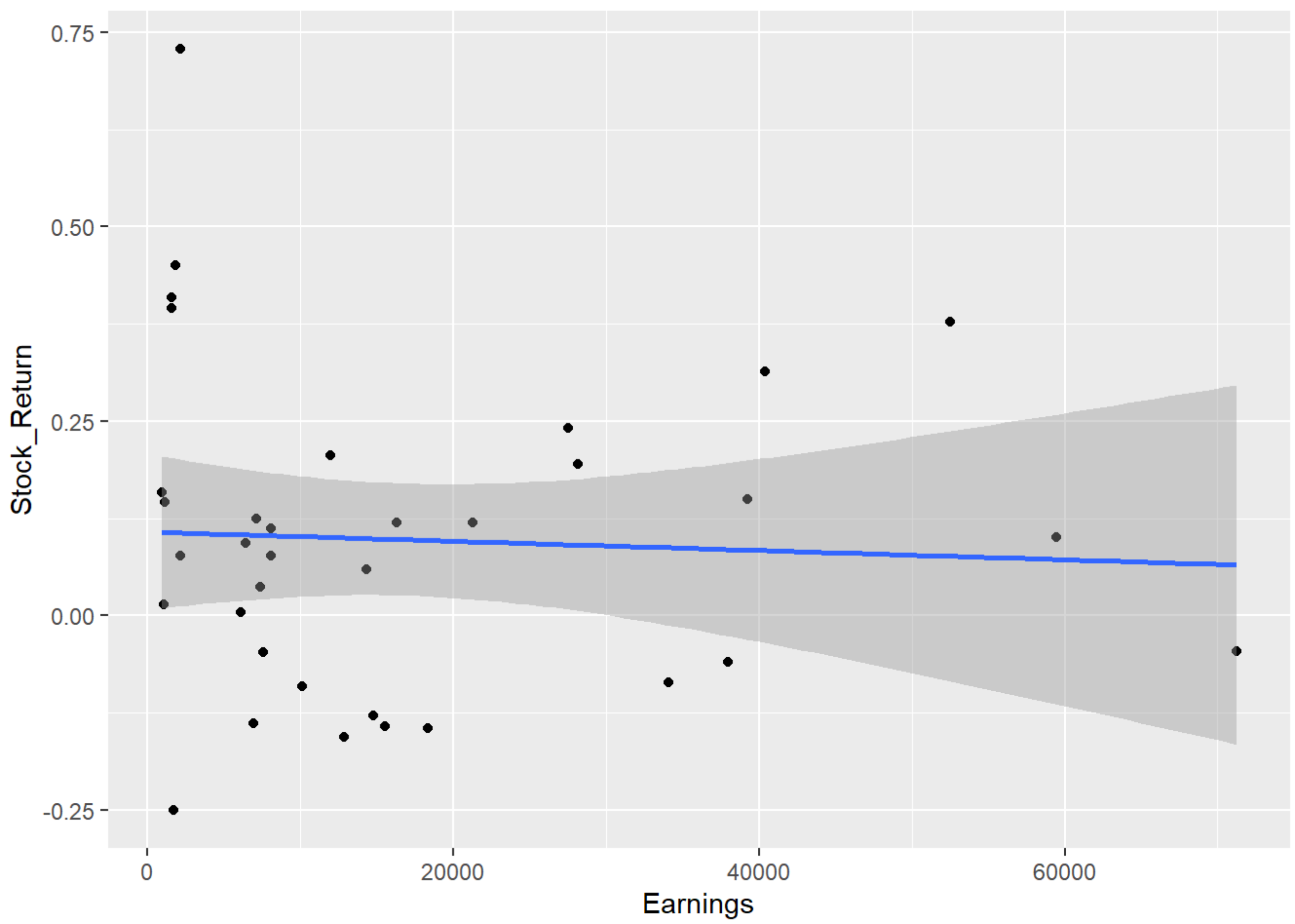
```
regression <- lm(Stock_Return ~ Earnings, data = finaldf)
summary(regression)
```

```
##
## Call:
## lm(formula = Stock_Return ~ Earnings, data = finaldf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35749 -0.14760 -0.01029  0.08411  0.62203
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.077e-01  4.914e-02   2.192   0.0356 *
## Earnings    -5.960e-07  1.993e-06  -0.299   0.7668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2095 on 33 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.002702,	Adjusted R-squared:  -0.02752
## F-statistic: 0.08942 on 1 and 33 DF,  p-value: 0.7668
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



Discussion of Result

The intercept is equal to 1.077e-01 and the slope is 5.960e-07. The model demonstrates a very low R-squared value of 0.27%. This could be due to the noise associated with market data. The p-value is not significant which may hint that the regression model does not work. It turns out from the data of stock returns and corresponding earnings that the two variables have a negative linear relationship.

Stock Return = 1.077e-01 - 5.960e-07 * Earnings