National College of Ireland

Configuration Manual

MSc Research Project
MSc Fintech

Nachiket Kaore
Student ID:x19107978

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nachiket Kaore
Student ID:	x19107978
Programme:	MSc Fintech **Year:** 2020
Module:	Research Project
Lecturer:	Noel Cosgrave
Submission Due Date:	28th September 2020
Project Title:	A Comparative Study on the Impact of Portfolio Diversity.
Word Count:	2089 **Page Count:** 12

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: 28th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Nachiket Kaore
Student ID: x19107978

1 Introduction

In this manual, all the tools and systems will be discussed in detail which were used in this research. First, the system used to run the codes will be discussed followed by the packages used and the code itself.

2 System Configuration

Below is a description of the system used for this research.

Category	Information
OS Name	Microsoft Windows 10 Home Single Language
Version	10.0.19041 Build 19041
System Model	HP Pavilion x360 Convertible 14-dh0xxx
System Type	x64-based PC
Processor	Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz, 1800 Mhz, 4 Core(s), 8 Logical Processor(s)
BIOS Version/Date	Insyde F.07, 03-07-2019
Memory	8GB RAM + 2GB Optane, 1TB HDD, 256GB SSD

3 R Studio (Version and Packages)

All of the performance metrics have been tested using R Version 3.6.1 and RStudio version 1.2.1335. Below is the list of libraries and packages used for conducting this research.

Table 1 : List of all packages used

Sr. No.	Library	Version	Requirement
1	quantmod	0.4.17	Data collection & Visualization
2	PerformanceAnalytics	2.0.4	Statistical Evaluation
3	corrplot	0.84	Visualization
4	reshape	3.6.1	Data transformation
5	ggplot	3.2.1	Visualization
6	PortfolioAnalytics	1.1.0	Statistical Evaluation
7	ROI	0.3.3	Statistical Evaluation

4 Data Collection and Portfolio Creation

Data collection for this research is straightforward. Using the below code, data can be directly downloaded from yahoo finance.

```
## Creating Porfolio 1 & 2
tickers1 <-
c("VTI","AMZN","GOOGL","FB","URI","NVDA","NFLX","DIS","SHOP","MSFT")
tickers2 <- c("NVDA","URI")

## Downloading Prices
portfolioPrices1 <- NULL
for (Ticker1 in tickers1)
  portfolioPrices1 <- cbind(portfolioPrices1,
                            getSymbols.yahoo(Ticker1, from="2017-01-01",
                            periodicity = "daily", auto.assign=FALSE)[,4])

portfolioPrices2 <- NULL
for (Ticker2 in tickers2)
  portfolioPrices2 <- cbind(portfolioPrices2,
                            getSymbols.yahoo(Ticker2, from="2017-01-01",
                            periodicity = "daily", auto.assign=FALSE)[,4])

## Market Index - S&P 500
Marketindex <- getSymbols.yahoo("^GSPC", from="2017-01-01", periodicity =
"daily", auto.assign=FALSE)[,4]
```

5 Plotting the Trend

For plotting the trend of each stock in the portfolio, below code was used.

```
## Getting individual stock data

getSymbols("VTI",
           from = "2017-01-01",
           to = "2020-01-29",
           src="yahoo")

getSymbols("AMZN",
           from = "2017-01-01",
           to = "2020-01-29",
           src="yahoo")

getSymbols("GOOGL",
           from = "2017-01-01",
           to = "2020-01-29",
           src="yahoo")

getSymbols("FB",
           from = "2017-01-01",
           to = "2020-01-29",
           src="yahoo")

getSymbols("URI",
           from = "2017-01-01",
           to = "2020-01-29",
           src="yahoo")

getSymbols("NVDA",
           from = "2017-01-01",
           to = "2020-01-29",
           src="yahoo")

getSymbols("NFLX",
```

```
            from = "2017-01-01",
            to = "2020-01-29",
            src="yahoo")

getSymbols("DIS",
            from = "2017-01-01",
            to = "2020-01-29",
            src="yahoo")

getSymbols("SHOP",
            from = "2017-01-01",
            to = "2020-01-29",
            src="yahoo")

getSymbols("MSFT",
            from = "2017-01-01",
            to = "2020-01-29",
            src="yahoo")

getSymbols("^GSPC",
            from = "2017-01-01",
            to = "2020-01-29",
            src="yahoo")


chartSeries(x=VTI,TA='addBBands();addVo()',name = "Vanguard",theme =
chartTheme('white'))
chartSeries(x=AMZN,TA='addBBands();addVo()',name="Amazon",theme =
chartTheme('white'))
chartSeries(x=GOOGL,TA='addBBands();addVo()',name = "Alphabet",theme =
chartTheme('white'))
chartSeries(x=FB,TA='addBBands();addVo()',name="Facebook",theme =
chartTheme('white'))
chartSeries(x=URI,TA='addBBands();addVo()',name="United Rentals",theme =
chartTheme('white'))
chartSeries(x=NVDA,TA='addBBands();addVo()', name = "Nvidia",theme =
chartTheme('white'))
chartSeries(x=NFLX,TA='addBBands();addVo()', name = "Netflix",theme =
chartTheme('white'))
chartSeries(x=DIS,TA='addBBands();addVo()', name = "Walt Disney",theme =
chartTheme('white'))
chartSeries(x=SHOP,TA='addBBands();addVo()', name = "Shopify",theme =
chartTheme('white'))
chartSeries(x=MSFT,TA='addBBands();addVo()', name = "Microsoft",theme =
chartTheme('white'))
chartSeries(x=GSPC,TA='addBBands();addVo()', name = "S&P 500",theme =
chartTheme('white'))
```

6 Stock selection and Diversification

Correlation analysis was used to pick individual stocks for the portfolio, by based on the code below.

```
## Calculating Returns

> portfolio1_roc <- na.omit(ROC(portfolioPrices1))
> portfolio2_roc <- na.omit(ROC(portfolioPrices2))
> Marketchange <- na.omit(ROC(Marketindex))
> portfolio1_returns <- Return.portfolio(portfolio1_roc)
> portfolio2_returns <- Return.portfolio(portfolio2_roc)
> Cor_Porfolio_1=as.data.frame(cor(portfolio1_roc))
> Cor_Porfolio_1
             VTI.Close AMZN.Close GOOGL.Close  FB.Close URI.Close NVDA.Close NFLX.Close
VTI.Close    1.0000000  0.6058297   0.7825270 0.6422115 0.7324799  0.6655887  0.5100804
AMZN.Close   0.6058297  1.0000000   0.6843045 0.6123282 0.4129121  0.5785700  0.6432643
GOOGL.Close  0.7825270  0.6843045   1.0000000 0.6970479 0.5517233  0.6324322  0.5692671
FB.Close     0.6422115  0.6123282   0.6970479 1.0000000 0.4487419  0.5577121  0.4969532
URI.Close    0.7324799  0.4129121   0.5517233 0.4487419 1.0000000  0.5086404  0.3772806
NVDA.Close   0.6655887  0.5785700   0.6324322 0.5577121 0.5086404  1.0000000  0.5254943
NFLX.Close   0.5100804  0.6432643   0.5692671 0.4969532 0.3772806  0.5254943  1.0000000
DIS.Close    0.7213410  0.3735669   0.5347980 0.4311576 0.6008903  0.3982939  0.3020975
SHOP.Close   0.5178793  0.5426237   0.5206330 0.4668732 0.3067410  0.5160848  0.4764839
MSFT.Close   0.8445256  0.7106406   0.7989881 0.6453178 0.5455528  0.6620873  0.5853688
             DIS.Close SHOP.Close MSFT.Close
VTI.Close    0.7213410  0.5178793  0.8445256
AMZN.Close   0.3735669  0.5426237  0.7106406
GOOGL.Close  0.5347980  0.5206330  0.7989881
FB.Close     0.4311576  0.4668732  0.6453178
URI.Close    0.6008903  0.3067410  0.5455528
NVDA.Close   0.3982939  0.5160848  0.6620873
NFLX.Close   0.3020975  0.4764839  0.5853688
DIS.Close    1.0000000  0.2818391  0.5508799
SHOP.Close   0.2818391  1.0000000  0.5887681
MSFT.Close   0.5508799  0.5887681  1.0000000

> cor(portfolio1_returns,Marketchange)
                  GSPC.Close
portfolio.returns  0.8025568


> cor(portfolio1_roc,Marketchange)
             GSPC.Close
VTI.Close     0.9962787
AMZN.Close    0.6144666
GOOGL.Close   0.7899806
FB.Close      0.6494819
URI.Close     0.7153064
NVDA.Close    0.6669896
NFLX.Close    0.5155873
DIS.Close     0.7133014
SHOP.Close    0.5155930
MSFT.Close    0.8570539


> cor(portfolio2_returns,Marketchange)
                  GSPC.Close
portfolio.returns  0.7833927
```

```
#For Vizualization

Stock_Correlation <-cor(portfolio1_roc)
Stock_Correlation_Market <-cor(portfolio1_roc, Marketchange)

corrplot(Stock_Correlation,method = "color",
                 outline = T,
                 addCoef.col = "black")

corrplot(Stock_Correlation_Market,
                 method = "color",
                 outline = T,
                 addCoef.col = "black")
```

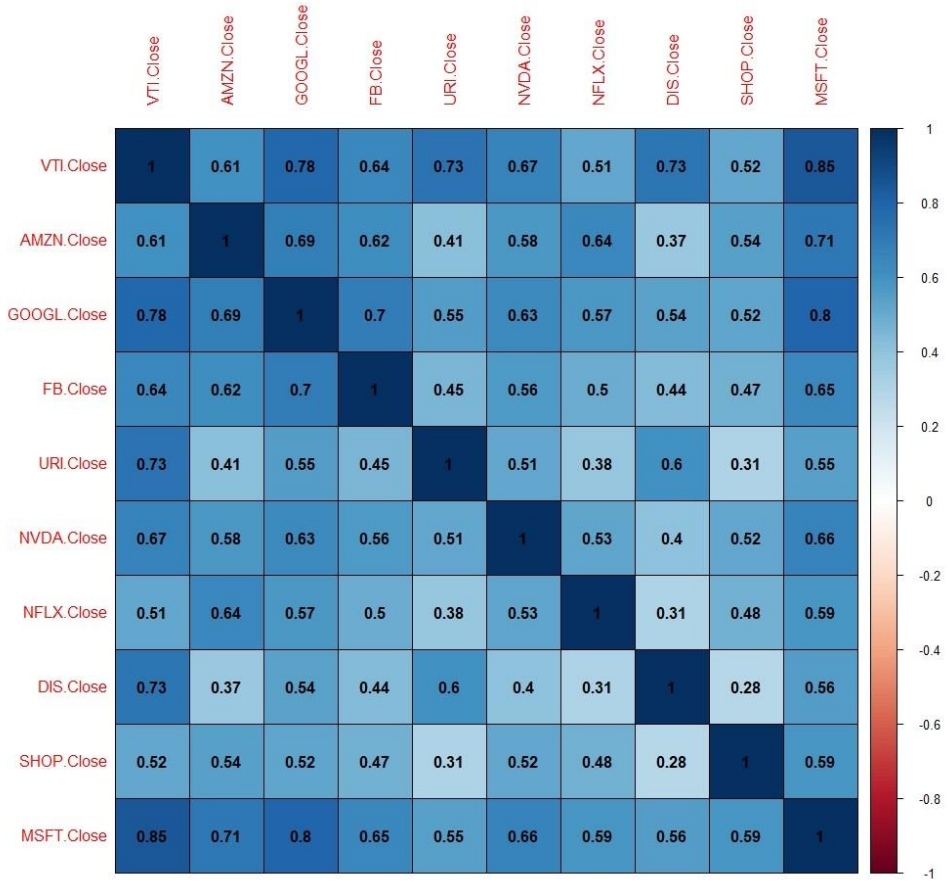

Figure 1: Correlation Plot

7 Performance Metrics

For gauging the performance of the 2 portfolios, below code was used.

7.1 Beta

```
## Calculating Beta (Volatility)

> CAPM.beta(portfolio1_returns,Marketchange)
[1] 1.147437

> CAPM.beta(portfolio2_returns,Marketchange)
[1] 1.545113

> CAPM.beta(portfolio1_roc,Marketchange)
                 VTI.Close AMZN.Close GOOGL.Close FB.Close URI.Close
NVDA.Close
Beta: GSPC.Close 0.9936791  0.8811664    1.043002 1.053221  1.597897
1.536727
                 NFLX.Close DIS.Close SHOP.Close MSFT.Close
Beta: GSPC.Close  0.9621104 0.9826741   1.295328   1.180504

> CAPM.beta(portfolio2_roc,Marketchange)
                 NVDA.Close URI.Close
Beta: GSPC.Close   1.536727  1.597897
```

7.2 Jensen's Alpha

```
## Calculating Alpha (Excess returns)

> CAPM.jensenAlpha(portfolio1_returns,Marketchange)
[1] 0.3019918

> CAPM.jensenAlpha(portfolio2_returns,Marketchange)
[1] 0.06571587

> CAPM.jensenAlpha(portfolio1_roc,Marketchange)
                                  VTI.Close AMZN.Close GOOGL.Close
FB.Close  URI.Close
Jensen's Alpha (Risk free = 0) -0.004042047  0.3447723  0.04715124
0.09171385 -0.1282045
                                NVDA.Close NFLX.Close  DIS.Close SHOP.Close
MSFT.Close
Jensen's Alpha (Risk free = 0)  0.2000365  0.2630446 -0.0752272   1.000913
0.2394432

> CAPM.jensenAlpha(portfolio2_roc,Marketchange)
                                NVDA.Close  URI.Close
Jensen's Alpha (Risk free = 0)  0.2000365 -0.1282045
```

7.3 Sharpe Ratio

```
##Sharpe Ratio (Risk-Return)

> SharpeRatio(portfolio1_returns,annualize = T,FUN = "StdDev")
                                              portfolio.returns
Annualized StdDev Sharpe (Rf=0%, p=95%):          1.361184
> SharpeRatio(portfolio2_returns,annualize = T,FUN = "StdDev")
                                              portfolio.returns
Annualized StdDev Sharpe (Rf=0%, p=95%):          0.5027603
```

7.4 Sortino Ratio

```
## SortinoRatio (Downside Risk-Return)

> SortinoRatio(portfolio1_returns)
                           portfolio.returns
Sortino Ratio (MAR = 0%)          0.1125077
> SortinoRatio(portfolio2_returns)
                           portfolio.returns
Sortino Ratio (MAR = 0%)         0.05729606
```

7.5 Annualized Returns

```
##Annualized Returns

> table.AnnualizedReturns(portfolio1_returns)
                           portfolio.returns
Annualized Return                     0.4073
Annualized Std Dev                    0.2992
Annualized Sharpe (Rf=0%)             1.3612

> table.AnnualizedReturns(portfolio2_returns)
                           portfolio.returns
Annualized Return                     0.2075
Annualized Std Dev                    0.4128
Annualized Sharpe (Rf=0%)             0.5028

> table1 <-table.CalendarReturns(portfolio1_returns)
> table2 <-table.CalendarReturns(portfolio2_returns)

> table1
      Jan  Feb  Mar Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec portfolio.returns
2017 -0.5 -0.8 -0.1 1.0 -0.2  0.0 -1.1  1.0  0.7 -1.7  0.7 -0.9              -1.9
2018  0.4 -0.7  2.7 0.1  0.3 -0.5 -0.7 -0.1 -0.1  4.0  1.1  1.7               8.4
2019  2.1 -0.4  0.9 0.8 -1.8  0.1 -1.0 -0.7  1.0 -0.8 -0.8  0.2              -0.4
2020 -1.3  2.8 -1.1 0.3  1.6  2.5 -0.3 -2.0   NA   NA   NA   NA
2.3
> table2
      Jan  Feb Mar  Apr  May  Jun  Jul  Aug Sep  Oct  Nov  Dec portfolio.returns
2017 -1.3 -1.4 0.2 -0.6 -0.9  0.3 -0.9  0.9 1.2  0.3  1.7 -1.5              -2.1
2018  0.5 -1.8 3.5 -1.8 -0.9 -1.0  1.0  0.5 1.8  3.8  2.6  0.4               8.8
2019  1.3 -0.8 1.4  0.5 -3.5  0.6 -3.4  0.6 0.7 -1.3 -1.3  0.3              -4.9
2020 -4.0  5.1 1.5 -1.4  3.5  2.5 -0.3 -0.1  NA   NA   NA   NA               6.8
```

7.6 Value at Risk and Conditional Value at Risk

```
#VaR and CVaR using equal weights
> VaR(portfolio1_roc,p=0.95,weights = equal_weights1 ,portfolio_method =
"component",
+     method = "modified")
$MVaR
[1] 0.02785963

$contribution
  VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close  NVDA.Close
NFLX.Close
0.001774610 0.002507749 0.002394311 0.002681094 0.003255408 0.004125288
0.003297825
  DIS.Close  SHOP.Close  MSFT.Close
0.001588477 0.003908895 0.002325970

$pct_contrib_MVaR
  VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close  NVDA.Close
NFLX.Close
 0.06369828  0.09001372  0.08594196  0.09623582  0.11685036  0.14807407
0.11837291
  DIS.Close  SHOP.Close  MSFT.Close
 0.05701718  0.14030679  0.08348892
```

```
> CVaR(portfolio1_roc,p=0.95,weights = equal_weights1 ,portfolio_method =
"component",
+      method = "modified",)
$MES
[1] 0.06608681

$contribution
  VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close  NVDA.Close
NFLX.Close
0.006040385 0.003581552 0.005523330 0.006818387 0.006939975 0.009737300
0.006911173
  DIS.Close  SHOP.Close  MSFT.Close
0.005171345 0.009135666 0.006227693

$pct_contrib_MES
  VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close  NVDA.Close
NFLX.Close
 0.09140078  0.05419465  0.08357689  0.10317319  0.10501301  0.14734107
0.10457720
  DIS.Close  SHOP.Close  MSFT.Close
 0.07825079  0.13823737  0.09423505

>
> VaR(portfolio2_roc,p=0.95,weights = equal_weights2 ,portfolio_method =
"component",
+     method = "modified")
$MVaR
[1] 0.04172283

$contribution
NVDA.Close  URI.Close
0.02206543 0.01965740

$pct_contrib_MVaR
NVDA.Close  URI.Close
 0.5288575  0.4711425

> CVaR(portfolio2_roc,p=0.95,weights = equal_weights2 ,portfolio_method =
"component",
+      method = "modified")
$MES
[1] 0.08398267

$contribution
NVDA.Close  URI.Close
0.04760731 0.03637536

$pct_contrib_MES
NVDA.Close  URI.Close
 0.5668707  0.4331293

>
> # VaR and CVaR using optimal weights
> VaR(portfolio1_roc,p=0.95,weights = OptimalWt_1 ,portfolio_method =
"component",
+     method = "modified")
$MVaR
[1] 0.03560943

$contribution
   VTI.Close   AMZN.Close  GOOGL.Close     FB.Close    URI.Close
NVDA.Close
0.0007980489 0.0011613398 0.0011055873 0.0012200573 0.0012954871
0.0131925855
  NFLX.Close    DIS.Close   SHOP.Close   MSFT.Close
0.0015278284 0.0006450495 0.0135365985 0.0011268464

$pct_contrib_MVaR
```

```
  VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close  NVDA.Close
NFLX.Close
 0.02241117  0.03261327  0.03104760  0.03426220  0.03638045  0.37048012
0.04290516
  DIS.Close  SHOP.Close  MSFT.Close
 0.01811457  0.38014085  0.03164461

> CVaR(portfolio1_roc,p=0.95,weights = OptimalWt_1 ,portfolio_method =
"component",
+     method = "modified",)
$MES
[1] 0.07831111

$contribution
  VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close  NVDA.Close
NFLX.Close
0.003148453 0.001615596 0.003047609 0.003712038 0.003340813 0.029939373
0.003199952
  DIS.Close  SHOP.Close  MSFT.Close
0.002720298 0.023986262 0.003600715

$pct_contrib_MES
  VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close  NVDA.Close
NFLX.Close
 0.04020442  0.02063048  0.03891669  0.04740117  0.04266078  0.38231323
0.04086204
  DIS.Close  SHOP.Close  MSFT.Close
 0.03473706  0.30629450  0.04597962

>
> VaR(portfolio2_roc,p=0.95,weights = OptimalWt_2 ,portfolio_method =
"component",
+     method = "modified")
$MVaR
[1] 0.04456381

$contribution
 NVDA.Close   URI.Close
0.036657118 0.007906696

$pct_contrib_MVaR
NVDA.Close  URI.Close
 0.8225759  0.1774241

> CVaR(portfolio2_roc,p=0.95,weights = OptimalWt_2 ,portfolio_method =
"component",
+      method = "modified")
$MES
[1] 0.09379995

$contribution
NVDA.Close  URI.Close
0.08057893 0.01322102

$pct_contrib_MES
NVDA.Close  URI.Close
 0.8590509  0.1409491
```

7.6.1 Value at Risk visualization and comparison

In order to compare the VaR of both portfolios, visual aid can be the best option. Using the below code, this research creates a barplot for data visualization.

```
### Compare VaR Calcualation Method of each stocks

> #Portfolio 1
>
> ## Calculation of VaR using historical, modified and gaussian methods
> Historical_Var1<-VaR(portfolio1_roc,p=0.95,method = "historical",weights
= NULL,portfolio_method = "single")
> Modified_Var1<-VaR(portfolio1_roc,p=0.95,method = "modified",weights =
NULL,portfolio_method = "single")
> Gaussian_Var1<-VaR(portfolio1_roc,p=0.95,method = "gaussian",weights =
NULL,portfolio_method = "single")
>
> ## Creating a Data frame for plotting
> All_VAR1<-data.frame(rbind(Historical_Var1,Modified_Var1,Gaussian_Var1))
> row.names(All_VAR1)<-c("Historical","Modified","Gaussian")
> head(All_VAR1)
             VTI.Close  AMZN.Close GOOGL.Close    FB.Close   URI.Close
NVDA.Close
Historical -0.01811983 -0.02866067 -0.02618986 -0.02930171 -0.04240688 -
0.04556490
Modified   -0.01993622 -0.02688825 -0.02755428 -0.03518316 -0.04688256 -
0.05001805
Gaussian   -0.02119862 -0.02949525 -0.02793393 -0.03422879 -0.04790638 -
0.04830202
             NFLX.Close   DIS.Close  SHOP.Close  MSFT.Close
Historical -0.04053015 -0.02449161 -0.05601070 -0.02671026
Modified   -0.03793167 -0.02444290 -0.05270972 -0.02545931
Gaussian   -0.03894695 -0.02963534 -0.05091380 -0.02850341
>
> ## Calculating of VaR in each Method for Portfolio with Optimal weights
> port_Var_Historical1<-VaR(portfolio1_roc,p=0.95,method =
"historical",weights = OptimalWt_1,portfolio_method =
"component")$hVaR[["hVaR 95%"]]
> port_Var_Modified1<-VaR(portfolio1_roc,p=0.95,method =
"modified",weights = OptimalWt_1,portfolio_method = "component")$MVaR[1]
> port_Var_Gaussian1<-VaR(portfolio1_roc,p=0.95,method =
"gaussian",weights = OptimalWt_1,portfolio_method = "component")$VaR[1]
>
> ##Combining them all into one Data Frame
> All_VAR1$Portfolio<-c(0)
> All_VAR1$Portfolio<-
c(port_Var_Historical1,port_Var_Modified1,port_Var_Gaussian1)
> All_VAR1<-abs(All_VAR1)
> All_VAR1$Type<-c("Historical","Modified","Gaussian")
> All_VAR1
             VTI.Close AMZN.Close GOOGL.Close   FB.Close  URI.Close
NVDA.Close NFLX.Close
Historical 0.01811983 0.02866067  0.02618986 0.02930171 0.04240688
0.04556490 0.04053015
Modified   0.01993622 0.02688825  0.02755428 0.03518316 0.04688256
0.05001805 0.03793167
Gaussian   0.02119862 0.02949525  0.02793393 0.03422879 0.04790638
0.04830202 0.03894695
             DIS.Close SHOP.Close MSFT.Close  Portfolio       Type
Historical 0.02449161 0.05601070 0.02671026 0.04182751 Historical
Modified   0.02444290 0.05270972 0.02545931 0.03560943   Modified
Gaussian   0.02963534 0.05091380 0.02850341 0.03373157   Gaussian
>
> ## Ploting the Value at Risk
> plot_all_VAR1<-melt(All_VAR1,variable.name = "Stocks",value.name ="VaR"
)
Using Type as id variables
> ggplot(plot_all_VAR1,aes(x = Type,y = VaR, fill=Stocks))+geom_bar(stat =
"identity",position = "dodge")
>
```

```
>
> # Portfolio 2
> ## Calculation of VaR using historical, modified and gaussian methods
> Historical_Var2<-VaR(portfolio2_roc,p=0.95,method = "historical",weights
= NULL,portfolio_method = "single")
> Modified_Var2<-VaR(portfolio2_roc,p=0.95,method = "modified",weights =
NULL,portfolio_method = "single")
> Gaussian_Var2<-VaR(portfolio2_roc,p=0.95,method = "gaussian",weights =
NULL,portfolio_method = "single")
>
> ## Creating a Data frame for plotting
> All_VAR2<-data.frame(rbind(Historical_Var2,Modified_Var2,Gaussian_Var2))
> row.names(All_VAR2)<-c("Historical","Modified","Gaussian")
> head(All_VAR2)
            NVDA.Close   URI.Close
Historical -0.04556490 -0.04240688
Modified   -0.05001805 -0.04688256
Gaussian   -0.04830202 -0.04790638
>
> ## Calculating of VaR in each Method for Portfolio with Optimal weights
> port_Var_Historical2<-VaR(portfolio2_roc,p=0.95,method =
"historical",weights = OptimalWt_2,portfolio_method =
"component")$hVaR[["hVaR 95%"]]
> port_Var_Modified2<-VaR(portfolio2_roc,p=0.95,method =
"modified",weights = OptimalWt_2,portfolio_method = "component")$MVaR[1]
> port_Var_Gaussian2<-VaR(portfolio2_roc,p=0.95,method =
"gaussian",weights = OptimalWt_2,portfolio_method = "component")$VaR[1]
>
> ##Combining them all into one Data Frame
> All_VAR2$Portfolio<-c(0)
> All_VAR2$Portfolio<-
c(port_Var_Historical2,port_Var_Modified2,port_Var_Gaussian2)
> All_VAR2<-abs(All_VAR2)
> All_VAR2$Type<-c("Historical","Modified","Gaussian")
> All_VAR2
            NVDA.Close  URI.Close  Portfolio       Type
Historical 0.04556490 0.04240688 0.04199612 Historical
Modified   0.05001805 0.04688256 0.04456381   Modified
Gaussian   0.04830202 0.04790638 0.04350827   Gaussian
>
> ## Ploting the Value at Risk
> plot_all_VAR2<-melt(All_VAR2,variable.name = "Stocks",value.name ="VaR"
)
Using Type as id variables
> ggplot(plot_all_VAR2,aes(x = Type,y = VaR, fill=Stocks))+geom_bar(stat =
"identity",position = "dodge")
```

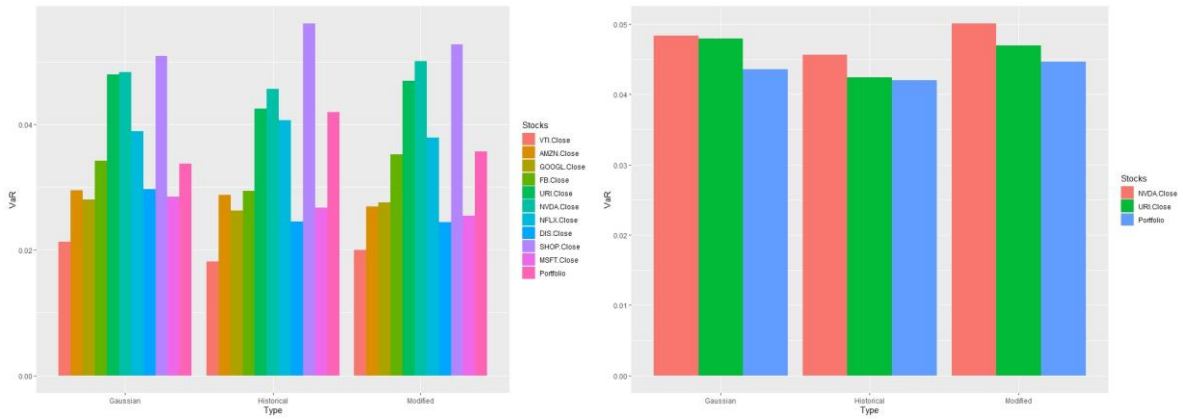

Figure 2: VaR comparison

8 Efficient Frontier and Optimization

Portfolio optimization is necessary for creating an efficient frontier plot. For optimization, 4 conditions have been introduced. The code for creating an efficient frontier plot and portfolio optimization is given below.

```
tickers <-
c("VTI","AMZN","GOOGL","FB","URI","NVDA","NFLX","DIS","SHOP","MSFT")
>
> portfolioPrices <- NULL
> for(ticker in tickers) {
+   portfolioPrices <- cbind(portfolioPrices,
+                            getSymbols.yahoo(ticker, from='2016-01-03',
periodicity = 'daily', auto.assign=FALSE)[,4])
+ }
>
> portfolioReturns <- na.omit(ROC(portfolioPrices))
> portf <- portfolio.spec(colnames(portfolioReturns))
>
> ## Adding Constraints and Objectives
> portf <- add.constraint(portf, type="weight_sum", min_sum=1, max_sum=1)
> portf <- add.constraint(portf, type="box", min=.05, max=.30)
> portf <- add.objective(portf, type="return", name="mean")
> portf <- add.objective(portf, type="risk", name="StdDev")
>
> ##Optimization
> optPort <- optimize.portfolio(portfolioReturns, portf, optimize_method =
"ROI", trace=TRUE)
> chart.Weights(optPort, neighbors = NULL,
+               main = "Portfolio A", las = 3, xlab = NULL, cex.lab = 1,
+               element.color = "black", cex.axis = 0.8, colorset = NULL,
+               legend.loc ="topleft" , cex.legend = 0.0, plot.type =
"barplot")
Error in strwidth(legend, units = "user", cex = cex) :
  invalid 'cex' value
 > ##Efficient Frontier
> ef <- extractEfficientFrontier(optPort, match.col = "StdDev",
n.portfolios = 25,
+                                risk_aversion = NULL)
>
> chart.EfficientFrontier(ef,
+                         match.col = "StdDev", n.portfolios = 25, xlim =
NULL, ylim = NULL,
+                         cex.axis = 0.8, element.color = "black", main =
"Portfolio A",
+                         RAR.text = "SR", rf = 0, tangent.line = TRUE,
cex.legend = 0.8,
+                         chart.assets = TRUE, labels.assets = TRUE,
pch.assets = 21,
+                         cex.assets = 0.8)
```