Statistical Methods in Finance Final Project

May 09, 2022

This document is for calculation purposes only and does not represent the final analysis.

Data Processing

```
require(tidyquant)
```

```
## Loading required package: tidyquant

## Warning: package 'tidyquant' was built under R version 4.1.3

## Loading required package: lubridate

##
## Attaching package: 'lubridate'

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

## Loading required package: PerformanceAnalytics

## Warning: package 'PerformanceAnalytics' was built under R version 4.1.3

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.1.3

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

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

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

## Loading required package: quantmod

## Warning: package 'quantmod' was built under R version 4.1.3

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 4.1.3

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

## == Need to Learn tidyquant? ====================================================
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimiza
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

```
require(tidyverse)
```

```
## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x dplyr::first()           masks xts::first()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x dplyr::last()            masks xts::last()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
require(xts)

library(MASS) # for fitdistr() and kde2d() functions
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(copula) # for copula functions
```

```
## Warning: package 'copula' was built under R version 4.1.3

##
## Attaching package: 'copula'

## The following object is masked from 'package:lubridate':
##
##     interval
```

```
library(fGarch) # for standardized t density
```

```
## Warning: package 'fGarch' was built under R version 4.1.3

## Loading required package: timeDate

##
## Attaching package: 'timeDate'

## The following objects are masked from 'package:PerformanceAnalytics':
##
##     kurtosis, skewness

## Loading required package: timeSeries

## Warning: package 'timeSeries' was built under R version 4.1.3

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-

## Loading required package: fBasics

## Warning: package 'fBasics' was built under R version 4.1.3

##
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```
DOWNLOAD.DATA = FALSE
FILENAME = "./data/net_returns_portfolio.csv"
```

```
if(DOWNLOAD.DATA) {
    # Asset symbols that will be used for this analysis
    asset.symbols = c(
        "AMD", "MSFT", "SBUX", "AAPL",
        "ITUB", "FB", "NVDA", "F",
        "BAC", "T", "XOM", "VALE"
    )

    # Download the assets' hisotrical data and load the
    # variables to the environment
    historical.data = getSymbols(
        asset.symbols,
        src = "yahoo",
        from = "2017-05-05",
        to = "2022-05-05",
        periodicity = "monthly"
    )

    # This is a helper function that fetches an asset's
    # data from the environment variabels by using its
    # symbol
    adjusted.price.by.symbol = function(symbol) {
        adj = Ad(get(symbol))
        names(adj) = symbol
        return(adj)
    }

    # Extract the adjusted price for each asset by
    # using its symbol
    adj = asset.symbols %>%
        map(adjusted.price.by.symbol) %>%
        reduce(cbind)

    # Calculate the net return for each asset
    # and save the results to a tibble
    net.returns = (diff(adj) / stats::lag(adj) * 100)[-1, ] %>%
        as_tibble(rownames = "Date")

    # Save the net returns into a csv file
    write_csv(net.returns, FILENAME)
} else {
    net.returns = read_csv(FILENAME)
}
```

```
## Rows: 60 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (12): AMD, MSFT, SBUX, AAPL, ITUB, FB, NVDA, F, BAC, T, XOM, VALE
## date  (1): Date
##
```

```
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(net.returns)
```

```
## # A tibble: 6 x 13
##   Date           AMD   MSFT   SBUX  AAPL  ITUB     FB  NVDA      F    BAC      T
##   <date>       <dbl>  <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1 2017-07-01   9.05   5.47  -7.43   3.27  7.83 12.1   12.4   0.268 -0.577   3.37
## 2 2017-08-01  -4.48   2.85   1.63  10.3   7.27  1.61   4.26 -0.431 -0.954  -2.29
## 3 2017-09-01  -1.92   0.156 -1.65  -5.66  8.43 -0.640  5.60  8.52   6.61    4.56
## 4 2017-10-01 -13.8   11.7    2.10   9.68 -6.46  5.38  15.7   2.51   8.09  -14.1
## 5 2017-11-01  -0.910  1.19   5.43   1.66 -1.99 -1.60  -2.95  3.30   2.85    9.92
## 6 2017-12-01  -5.60   2.14  -0.148 -1.17  3.63 -0.406 -3.53 -0.240  5.24    6.87
## # ... with 2 more variables: XOM <dbl>, VALE <dbl>
```

Summary

Descriptive Statistics

Portfolio Theory

Asset Allocation

Principal Component analysis

Risk Management

Copulas

Conclusion