OR538 Final Project - Step 1: Data Preparation

# Obtaining the data set

The first step is to create the dataset by pulling values and generating the rate of change of interest (i.e., log-returns)

In this case we are going to use the stock prices for two companies in the information technology industry in particular the Data & Processing and outsourced services (i.e., Paycheck inc. and PayPal), one company in the Consumer Staples industry (i.e., Walmart), one market (i.e., S&P 500) and the risk-free rates for 2019 provided by the US Department of Treasury.

We will be analizing the data set obtained in a time-period of 2 years, we will gather data from November 23, 2017 to November 23, 2019. The function getSymbols provided by the quantmod library will be leverage to pull data from Yahoo Finance and the federal reserve.

Below is a table mapping the symbol to the security and the source of data.

|  |  |  |
| --- | --- | --- |
| Symbol | Description | Source |
| PAYX | PayCheck Inc Stock Prices | Yahoo Finance |
| PYPL | Paypal Stock Prices | Yahoo Finance |
| WMT | Walmart Stock Prices | Yahoo Finance |
| SP | S&P 500 Stock Prices | Yahoo Finance |

pacman::p\_load(quantmod, tseries)  
  
symbols <- c("PAYX", "PYPL", "WMT", "SP")  
  
getSymbols(symbols, from = "2017-11-23", to = "2019-11-23")

## '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.

## [1] "PAYX" "PYPL" "WMT" "SP"

# Creating data frames in an extensible time-series data type

getSymbols will provide the stock prices in an extensible time-series format containing the stocks price for each one of the symbols. However, it will be more beneficial to manage data frames containing the necessary transformations that would enable us to perform the best possible analyis of financial time-series information.

The script below will create 3 data frames containing the adjusted price, return and log return for each one of the stocks. Separating this information in individual data frames will simplify the usage of such information for data visualization, model training and prediction.

df\_adj <- data.frame(PAYX=PAYX$PAYX.Adjusted,  
 PYPL=PYPL$PYPL.Adjusted,  
 WMT=WMT$WMT.Adjusted,  
 SP=SP$SP.Adjusted)  
colnames(df\_adj) <- symbols  
  
df\_ret <- data.frame(PAYX=diff(PAYX$PAYX.Adjusted),  
 PYPL=diff(PYPL$PYPL.Adjusted),  
 WMT=diff(WMT$WMT.Adjusted),  
 SP=diff(SP$SP.Adjusted))  
colnames(df\_ret) <- symbols  
  
df\_log\_ret <- data.frame(PAYX=diff(log(PAYX$PAYX.Adjusted)),  
 PYPL=diff(log(PYPL$PYPL.Adjusted)),  
 WMT=diff(log(WMT$WMT.Adjusted)),  
 SP=diff(log(SP$SP.Adjusted)))  
colnames(df\_log\_ret) <- symbols

# Consolidating the dataset

To properly share the information; in this section we will be consolidating all the data set information obtained and generated in a single csv file.

df\_export <- data.frame(PAYX, diff(PAYX$PAYX.Adjusted), diff(log(PAYX$PAYX.Adjusted)),  
 PYPL, diff(PYPL$PYPL.Adjusted), diff(log(PYPL$PYPL.Adjusted)),  
 WMT, diff(WMT$WMT.Adjusted), diff(log(WMT$WMT.Adjusted)),  
 SP, diff(SP$SP.Adjusted), diff(log(SP$SP.Adjusted)))  
  
colnames(df\_export) <- c("PAYX.Open", "PAYX.High", "PAYX.Low", "PAYX.Close",   
 "PAYX.Volume","PAYX.Adjusted", "PAYX.Adj.Ret", "PAYX.Adj.Log.Ret",  
 "PYPL.Open", "PYPL.High", "PYPL.Low", "PYPL.Close",  
 "PYPL.Volume","PYPL.Adjusted", "PYPL.Adj.Ret", "PYPL.Adj.Log.Ret",  
 "WMT.Open", "WMT.High", "WMT.Low", "WMT.Close",   
 "WMT.Volume", "WMT.Adjusted","WMT.Adj.Ret", "WMT.Adj.Log.Ret",  
 "SP.Open", "SP.High", "SP.Low", "SP.Close",  
 "SP.Volume", "SP.Adjusted", "SP.Adj.Ret", "SP.Adj.Log.Ret")  
  
write.csv(df\_export,   
 "/Users/angelsierra/ML\_Project/OR538-FinalPrj/OR538-portfolio-securities.csv")