FE582 Group Project

October 16, 2018

1 Problem Statement

Is there a correlation between Cryptocurrencies price and Global Stock Markets or Commodities Markets (Gold/Crude oil)?

2 Objective

Objective of this analysis is to source and compare historical price data for cryptocurrencies with corresponding period prices data for various global stock indices. We also plan to compare cryptocurrencies prices data with commodities like Gold and or Crude oil prices. This analysis will help us understand if there is any correlation between cryptocurrencies prices and selected stock markets or selected commodities.

As correlations are not static, their evolution over time will also be considered.

3 Proposed Data for Analysis

Below is draft list of cryptocurrencies, stock indices and commodities indices we propose to use for the analysis.

Top 10 Cryptocurrencies (by Market Capitalization) - Market Capitalization Data Source (https://goldprice.org/cryptocurrency-price):

- Bitcoin
- Ether
- Ripple

- Bitcoin Cash
- EOS Price
- Stellar
- Litecoin
- Tether
- Cardano
- Monero

Proposed Stock Indices:

- S&P 500 Index
- FTSE 100 Index
- Euro Stoxx 50
- Nikkei 225
- Hang Seng Index
- NIFTY 50
- VIX
- Crypto Mining Stocks
- NVIDIA
- AMD

Proposed Sector Stock Indices:

- S&P 500 GICS Subindices
- Industrials
- Consumer Discretionary
- Consumer Staples
- Financials
- Energy
- IT
- Materials
- Health Care
- Utilities
- Communication Services
- Real Estate

Proposed Stock Indices

- Gold
- Crude Oil

4 Proof Of Concept: Data Sourcing and Initial Exploratory Data Analysis

The following section of this proposal shows exampels of our proposed data sourcing methods and initial analysis as a proof of concept.

4.1 Libraries and Packages

Loading of required libararies and packages.

```
library(quantmod)       #Download of data
library(ggcorrplot)     #Produces correlation maps
library(TTR)            #Functions used to produce rolling correlation coefficients
library(ggplot2)        #Plots
library(tidyquant)      #Used to append rolling correlations results to data frames
 in a tidy fashion
library(gdata)          #Used to read from xls files
library(RColorBrewer)   #Color palettes
library(tidyverse)      #Used to deal with data in tidy fashion
```

4.2 Load Data

4.2.1 Sector Index Data

4.2.1.1 Load S&P500 GICS Sector Indices

Data for sector indices was downloaded to xls files. Data was downloaded from S&P Indices website (spindices.com).

```
dl_sp_sectors<-function(symbol,filename)
  {
  setwd('C:/Users/gordon.garisch/Documents/Projects/Stevens/FE582/Grp_Prjkt/Data/SP
500/')
  df<-read.xls(filename,col.names=c('date',paste0(symbol,'.','Close')))
  return(df)
}

sp500_lst <- list()
sp500_symbols <- c('Industrials',
                   'Consumer Discretionary',
                   'Consumer Staples',
                   'Financials',
                   'Energy',
                   'IT',
                   'Materials',
                   'Health Care',
                   'Utilities',
                   'Communication Services',
                   'Real Estate')

sp500_filenames <- c('industrials.xls',
                   'consumer_discretionary.xls',
                   'Consumer_staples.xls',
                   'financials.xls',
                   'energy.xls',
                   'it.xls',
                   'materials.xls',
                   'health_care.xls',
                   'utilities.xls',
                   'communication_services.xls',
                   'real_estate.xls')

for (i in 1:length(sp500_symbols)) {
  sp500_lst[[i]] <- dl_sp_sectors(sp500_symbols[i],sp500_filenames[i])
}

sp500_sectors<-sp500_lst %>% reduce(left_join, by = 'date')
sp500_sectors$date<-as.Date(sp500_sectors$date)
```

4.2.1.2 Calculate Rate of Change of S&P Sector Indices

```
sp500_sectors<-sp500_sectors %>%
  tq_mutate(select=grep('Close',names(sp500_sectors),value=TRUE),
            mutate_fun = ROC,
            col_rename=gsub("Close","ROC",grep('Close',names(sp500_sectors),value=T
RUE)))
```

4.2.2 Stock and Index Data Avaialable with Quantmod

4.2.2.1 Download Bitcoin and Stock Data from Quantmod

Add symbols to symb_lst list to add more tickers to analysis. More tickers will increase processing time significantly.

```
setSymbolLookup(BTCN=list(name="BTCUSD=X",src="yahoo"))
symb_lst <- c('^GSPC','^VIX','NVDA','AMD','TSM','^FTSE','^N225','^NSEI','BTCN')

stck_lst <- list()

for (i in 1:length(symb_lst)) {
  stck_lst[[i]] <- getSymbols(symb_lst[i], src="yahoo", auto.assign=FALSE, return.c
lass="xts")
}

stcks_xts<-do.call(merge,stck_lst)
```

4.2.2.2 Calculate Rate of Change of Quantmod Downloaded Data

```
stcks_df <- data.frame(date=index(stcks_xts),coredata(stcks_xts))
stcks_df<-stcks_df %>%
  tq_mutate(grep('Adjusted',names(stcks_df)),
            mutate_fun = ROC,
            col_rename=gsub("Adjusted","ROC",x=grep('Adjusted',names(stcks_df),valu
e=TRUE)))
```

4.3 Merge Sector and Quantmod Data

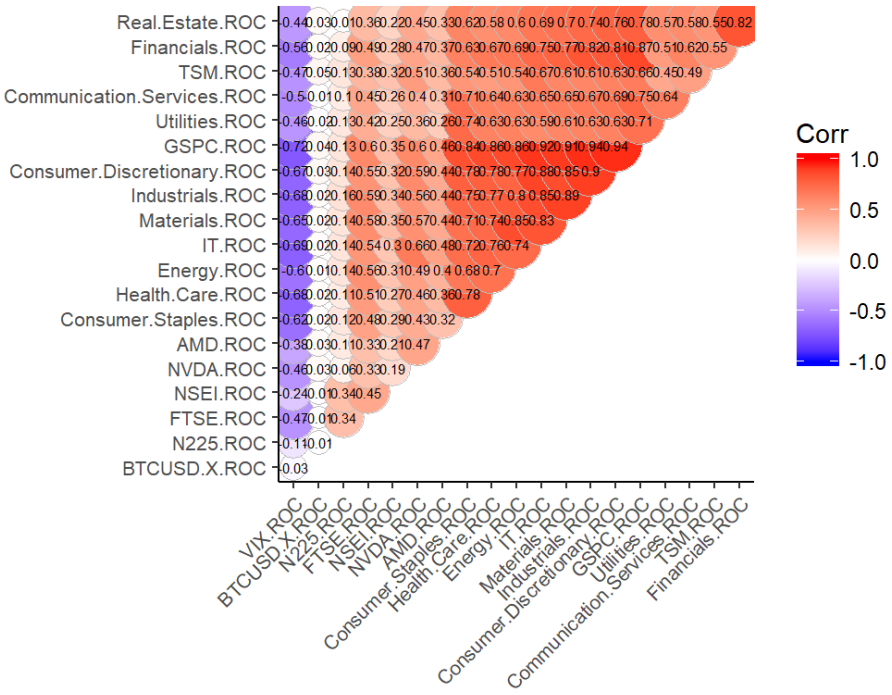
```
all_ROC_df<-merge(subset(sp500_sectors,select=grepl('ROC|date',names(sp500_sector
s))),subset(stcks_df,select=grepl('ROC|date',names(stcks_df))),by='date')
```

4.4 Analysis

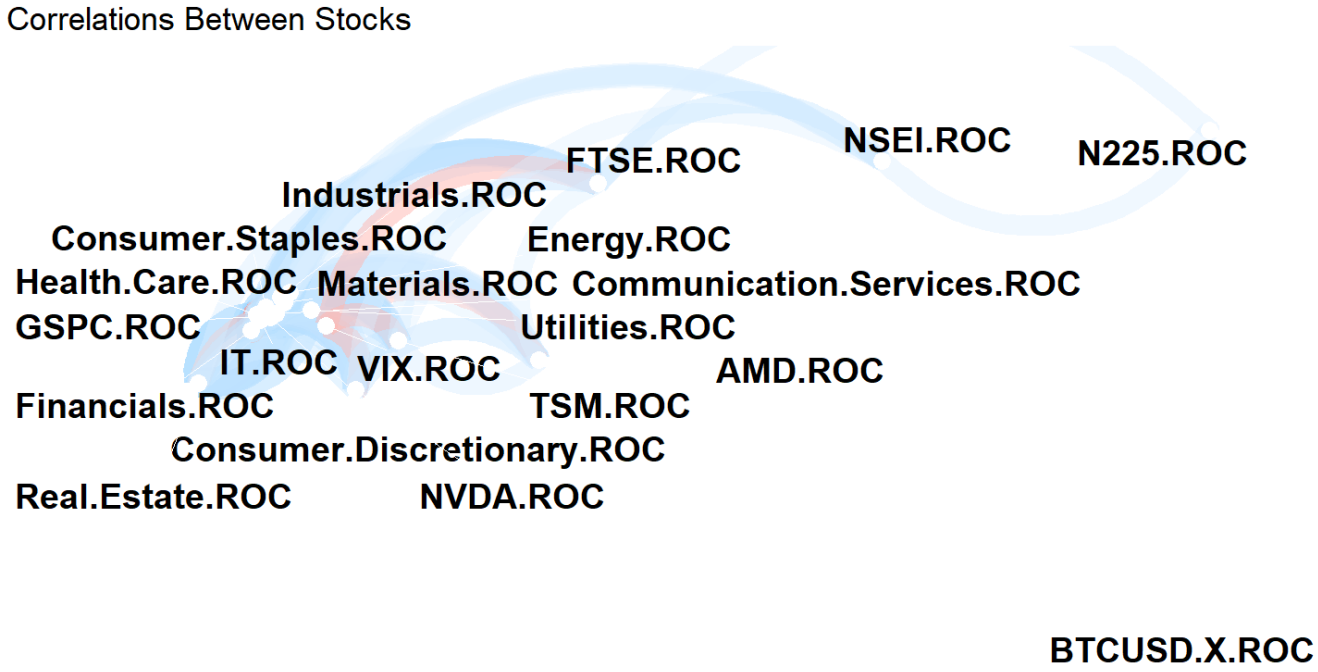
4.4.1 Correlations and Network Plots

Below are correlation maps and network plots of correlations (https://rdrr.io/cran/corrr/man/network_plot.html) of the sample of stocks and indices downloaded above.

```
library(corrr)
corr<-all_ROC_df[,grepl('ROC',names(all_ROC_df))] %>%
  correlate()

ggcorrplot(as_matrix(corr),
           method='circle',
           type='upper',
           hc.order = TRUE,
           ggtheme = ggplot2::theme_classic,
           lab = TRUE,tl.cex=8,lab_size = 2)
```




```
corr %>%
  #network_plot(colours = c(palette_dark()[[1]], palette_dark()[[5]], palette_dark
()[[9]]), legend = TRUE) +
  network_plot(legend = TRUE)+
  #colours = brewer.pal(5,name = 'Accent')
  labs(title = "Correlations Between Stocks") +
  #expand_limits(x = c(-0.75, 0.25), y = c(-0.4, 0.4)) +theme_tq()Z +
  theme(legend.position = "bottom")
```



4.4.2 Rolling Correlations

4.4.2.1 Prepare Data

Correlations change over time. For this reason, we are also analyzing the changes in correlations over time, and the impact of using different rolling window sizes.

The function below adds rolling correlation output to a dataframe.

```
addcorr <- function(df,x,y,n,metric){

  na.omit(df) %>%
  tq_transmute_xy_(x=paste0(x,'.',metric),
                y=paste0(y,'.',metric),
                mutate_fun = 'runCor',
                n=n,
                col_rename='corr')
}
```

Initialize output dataframe and list of symbols

```
stcks_df_corr<-setNames(data.frame(matrix(ncol = 6, nrow = 0)),
                        c('date','x.symbol','y.symbol','metric','window','corr'))

symbols_y<-gsub('.ROC',replacement='',grep(pattern='ROC$',
                names(all_ROC_df),
                value=TRUE))
symbols_x<-'BTCUSD.X'
```

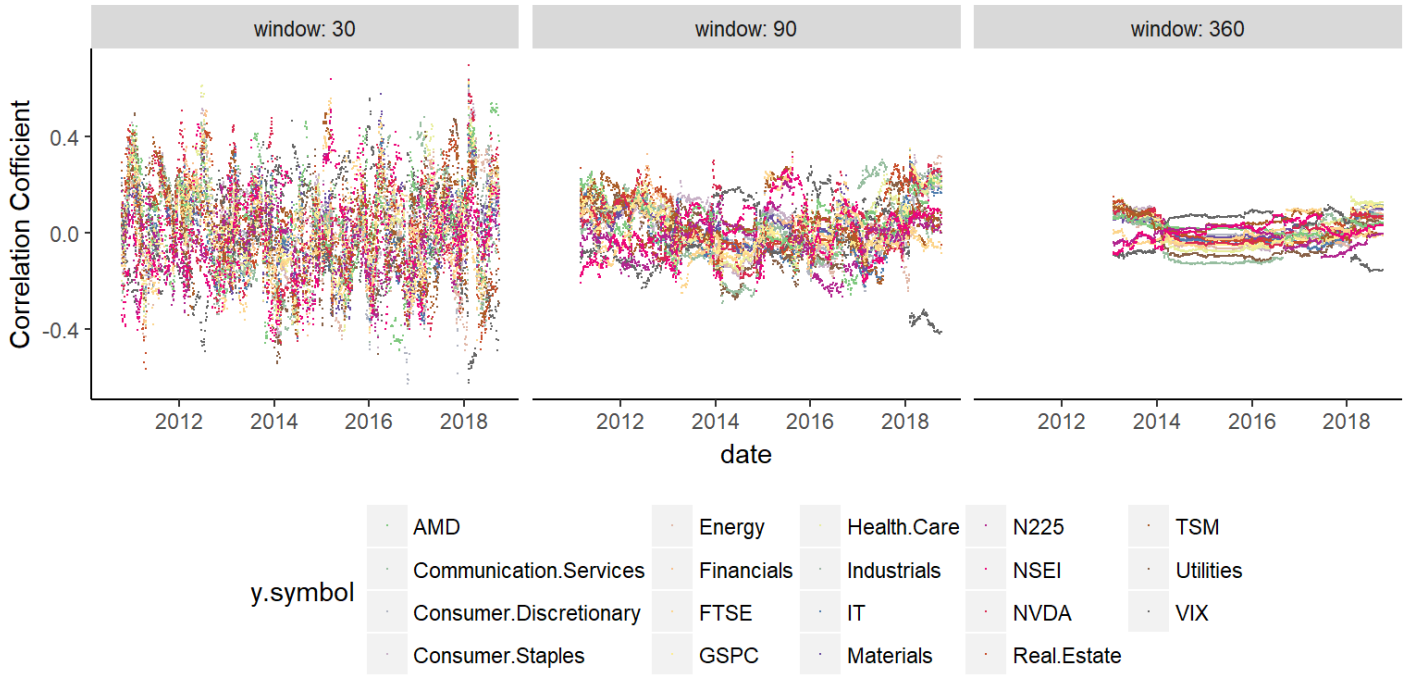
The code below loops through stocks and windows and adds the rolling correlations to the initialized dataframe above.

```
for(x in symbols_x)
  {
  for (y in symbols_y[which(symbols_y!=x)])
    {
    for(i in c(30,90,360))
      {

      if (nrow(subset(stcks_df_corr,x.symbol==y & y.symbol==x & window ==i))==0)
      {
        temp <- addcorr(all_ROC_df,
                                x,
                                y,
                                i,
                                'ROC')
      }
      else
      {
        temp <- subset(stcks_df_corr, x.symbol==y & y.symbol==x & window ==i)
      }
      temp$x.symbol<-x
      temp$y.symbol<-y
      temp$metric<-'ROC'
      temp$window<-i
      stcks_df_corr <-rbind(stcks_df_corr,temp)
      }
    }
  }
```

4.4.2.2 Vizualization

4.4.2.2.1 Plot of Correlations Over Time

```
colorCount <-  length(unique(stcks_df_corr$y.symbol))
getPalette <- colorRampPalette(brewer.pal(19, "Accent"))

ggplot(subset(stcks_df_corr,x.symbol=='BTCUSD.X'))+
  geom_point(aes(x=date,
                 y=corr,color=y.symbol),
             size=0.1)+
  facet_grid(~window,scales='free_x',labeller = label_both)+
  scale_y_continuous(name = "Correlation Cofficient")+
  scale_color_manual(values = getPalette(colorCount))+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank()) +
  theme(legend.position = 'bottom')
```



4.4.2.2.2 Plot of Correlation Distributions

```
ggplot(subset(stcks_df_corr,x.symbol=='BTCUSD.X'))+
  geom_density(aes(x=corr,
                   y=..density..,fill=y.symbol),alpha=0.5)+
  facet_grid(window~.,scales='free_y', labeller=label_both)+
  scale_y_continuous(name = 'Density')+
  scale_x_continuous(name = "Correlation Cofficient")+
  scale_fill_manual(values = getPalette(colorCount))+
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank()) +
  theme(legend.position = 'bottom')
```