Annex 1

Alejandro Vaca

3/3/2018

#portfolio  
#install.packages('readr')  
library(readr)  
temp = list.files(pattern="\*.csv")  
myfiles = lapply(temp, read.csv)  
names = c("")  
  
# list all csv files from the current directory  
list.files(pattern=".csv$") # use the pattern argument to define a common pattern for import files with regex. Here: .csv

## [1] "bat.csv" "btc.csv" "dash.csv" "eos.csv" "eth.csv" "iot.csv"   
## [7] "lsk.csv" "ltc.csv" "maid.csv" "neo.csv" "xem.csv" "xmr.csv"   
## [13] "xrp.csv" "zec.csv"

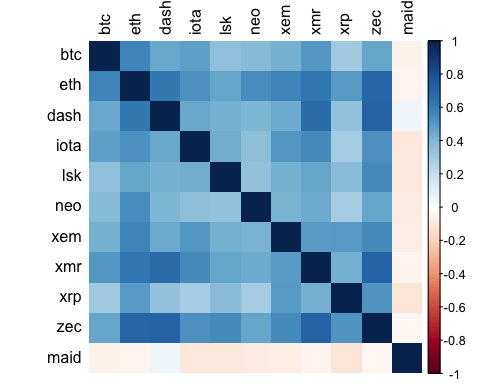
# create a list from these files  
list.filenames<-list.files(pattern=".csv$")  
list.filenames

## [1] "bat.csv" "btc.csv" "dash.csv" "eos.csv" "eth.csv" "iot.csv"   
## [7] "lsk.csv" "ltc.csv" "maid.csv" "neo.csv" "xem.csv" "xmr.csv"   
## [13] "xrp.csv" "zec.csv"

# create an empty list that will serve as a container to receive the incoming files  
list.data<-list()  
  
# create a loop to read in your data  
for (i in 1:length(list.filenames))  
{  
 list.data[[i]]<-read.csv(list.filenames[i])  
}  
  
# add the names of your data to the list  
names(list.data)<-list.filenames  
da <- list.data  
bat = data.frame(da[1])  
btc = data.frame(da[2])  
dash = data.frame(da[3])  
#eos = data.frame(da[4])  
eth = data.frame(da[5])  
iota = data.frame(da[6])  
lsk = data.frame(da[7])  
ltc = data.frame(da[8])  
maid = data.frame(da[9])  
neo = data.frame(da[10])  
xem = data.frame(da[11])  
xmr = data.frame(da[12])  
xrp = data.frame(da[13])  
zec = data.frame(da[14])  
  
  
#close\_prices = data.frame(lapply(da, '[[', 3))   
  
bat = bat[ , 2:5]  
btc = btc[ , 2:5]  
dash = dash[ , 2:5]  
#eos = eos[ , 1:2]  
eth = eth[ , 2:5]  
iota = iota[ , 2:5]  
lsk = lsk[ , 2:5]  
ltc = ltc[ , 2:5]  
neo = neo[ , 2:5]  
xem = xem[ , 2:5]  
xmr = xmr[ , 2:5]  
xrp = xrp[ , 2:5]  
zec = zec[ , 2:5]  
maid = maid[ , 2:5]  
  
  
bat$volatility = bat[,3] - bat[,4]  
btc$volatility = btc[,3] - btc[,4]  
dash$volatility = dash[,3] - dash[,4]  
#we cannot use eos because there is no data about its volatility (high -low)  
eth$volatility = eth[,3] - eth[,4]  
iota$volatility = iota[,3] - iota[,4]  
lsk$volatility = lsk[,3] - lsk[,4]  
ltc$volatility = ltc[,3] - ltc[,4]  
neo$volatility = neo[,3] - neo[,4]  
xem$volatility = xem[,3] - xem[,4]  
xmr$volatility = xmr[,3] - xmr[,4]  
xrp$volatility = xrp[,3] - xrp[,4]  
zec$volatility = zec[,3] - zec[,4]  
maid$volatility = maid[,3] - maid[,4]  
  
#for different reasons, we will not use nor eos nor bat; the first one because it is still on ICO and most importantly there is no data for USD High and USD Low daily.  
#for BAT we do not have sufficient data...  
  
#the one with less data is neo, with 197 observations. Therefore, we should take the last 197 observations of all coins  
n = 196  
  
btc = btc[(nrow(btc)-n):nrow(btc),]  
dash = dash[(nrow(dash)-n):nrow(dash),]  
eth = eth[(nrow(eth)-n):nrow(eth),]  
iota = iota[(nrow(iota)-n):nrow(iota),]  
lsk = lsk[(nrow(lsk)-n):nrow(lsk),]  
ltc = ltc[(nrow(ltc)-n):nrow(ltc),]  
xem = xem[(nrow(xem)-n):nrow(xem),]  
xmr = xmr[(nrow(xmr)-n):nrow(xmr),]  
xrp = xrp[(nrow(xrp)-n):nrow(xrp),]  
zec = zec[(nrow(zec)-n):nrow(zec),]  
maid = maid[(nrow(maid)-n):nrow(maid),]  
  
btc$ret = c(NA, diff(log(btc$btc.csv.close)))  
dash$ret = c(NA, diff(log(dash$dash.csv.close)))  
eth$ret = c(NA, diff(log(eth$eth.csv.close)))  
iota$ret = c(NA, diff(log(iota$iot.csv.close)))  
lsk$ret = c(NA, diff(log(lsk$lsk.csv.close)))  
ltc$ret = c(NA, diff(log(ltc$ltc.csv.close)))  
neo$ret = c(NA, diff(log(neo$neo.csv.close)))  
xem$ret = c(NA, diff(log(xem$xem.csv.close)))  
xmr$ret = c(NA, diff(log(xmr$xmr.csv.close)))  
xrp$ret = c(NA, diff(log(xrp$xrp.csv.close)))  
zec$ret = c(NA, diff(log(zec$zec.csv.close)))  
maid$ret = c(NA, diff(log(maid$maid.csv.close)))  
  
btc = btc[2:nrow(btc), ]  
eth = eth[2:nrow(eth), ]  
dash = dash[2:197, ]  
iota = iota[2:197, ]  
lsk = lsk[2:197, ]  
ltc = ltc[2:197, ]  
neo = neo[2:197, ]  
xem = xem[2:197, ]  
xmr = xmr[2:197, ]  
xrp = xrp[2:197, ]  
zec = zec[2:197, ]  
maid = maid[2:197, ]  
  
returns = cbind(btc$ret, eth$ret, dash$ret, iota$ret, lsk$ret, neo$ret, xem$ret, xmr$ret, xrp$ret, zec$ret, maid$ret)  
returns = data.frame(returns)  
colnames(returns) = c("btc", "eth", "dash", "iota", "lsk", "neo", "xem", "xmr", "xrp", "zec", "maid")  
  
cor\_mat = cor(returns)  
  
#install.packages("corrplot")  
library(corrplot)

## corrplot 0.84 loaded

corrplot(cor\_mat, method='shade', type='full', shade.col=NA, tl.col='black')



#install.packages("TSclust")  
library(TSclust)

## Loading required package: wmtsa

## Loading required package: pdc

## Loading required package: cluster

D1 <- diss(returns, "COR")  
summary(D1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.7483 0.9707 1.0530 1.0914 1.1438 1.4930

average\_return\_btc = mean(btc$ret)  
  
#install.packages("dplyr")  
library(dplyr)

##   
## Attaching package: 'dplyr'

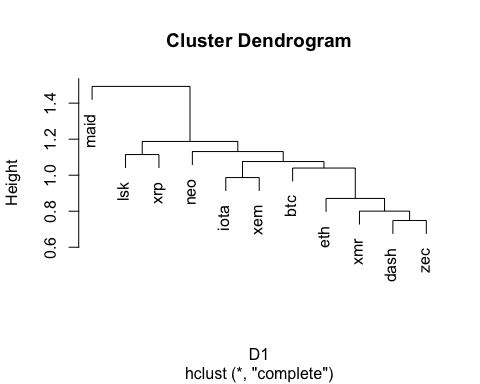
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

summary\_ret = returns %>%   
 summarize(av\_ret\_btc = mean(btc),  
 av\_ret\_eth = mean(eth),   
 av\_ret\_dash = mean(dash),   
 av\_ret\_iota = mean(iota),  
 av\_ret\_lsk = mean(lsk),   
 av\_ret\_neo = mean(neo),  
 av\_ret\_xem = mean(xem),   
 av\_ret\_xmr = mean(xmr),   
 av\_ret\_xrp = mean(xrp),   
 av\_ret\_zec = mean(zec),   
 av\_ret\_maid = mean(maid))  
#here we see the average return for the period selected of the cryptocurrencies under consideration.   
  
C1 <- hclust(D1)  
C1

##   
## Call:  
## hclust(d = D1)  
##   
## Cluster method : complete   
## Number of objects: 11

plot(C1)



#D2 <- diss(returns, "FRECHET")  
#the calculation takes too long for nothing; it's not worth running it.

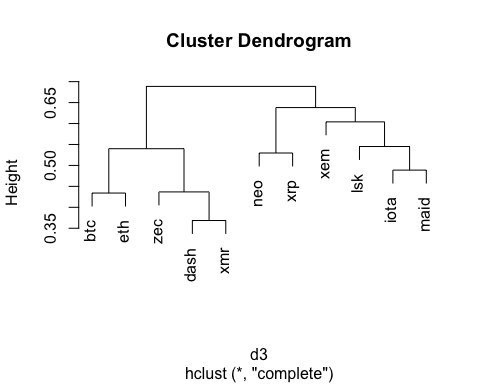
#c2 = hclust(D2)  
#c2  
#plot(c2)  
  
#we are going to try more methods

d3 <- diss(returns, "ACF")

c3 = hclust(d3)  
c3

##   
## Call:  
## hclust(d = d3)  
##   
## Cluster method : complete   
## Number of objects: 11

plot(c3)  
  
#install.packages("ggdendro")  
library(ggplot2)



library(ggdendro)  
  
ggdendrogram(c3, rotate = FALSE, theme\_dendro = FALSE, segments = TRUE, labels = TRUE ) +  
 ggtitle('HIERARCHICAL CLUSTER CRYPTOCURRENCIES')

