Final Project Step 2

Mahaldar Arti

11/4/2021

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

## Load the `data/r4ds/heights.csv` to  
#install.packages("readxl")  
library("readxl")  
  
#the excel sheet below represent Crypto Current Market Cap Data  
  
cryptodata <- read\_excel('C:/Users/Sandeep Raina/Documents/dsc520/data/Cryptocurrency.xlsx')  
  
#What are different ways you could look at this data to answer the questions you want to answer?  
### Functions to clean and examine the data frame  
dim(cryptodata)

## [1] 535168 8

str(cryptodata)

## tibble [535,168 x 8] (S3: tbl\_df/tbl/data.frame)  
## $ Currencyname: chr [1:535168] "0x" "0x" "0x" "0x" ...  
## $ Date : POSIXct[1:535168], format: "2017-08-16" "2017-08-17" ...  
## $ MarketCap : num [1:535168] 6.70e+07 1.34e+08 1.23e+08 1.77e+08 2.83e+08 ...  
## $ Close : num [1:535168] 0.224 0.207 0.293 0.479 0.424 ...  
## $ Open : num [1:535168] 0.112 0.223 0.206 0.295 0.471 ...  
## $ High : num [1:535168] 0.28 0.239 0.35 0.544 0.475 ...  
## $ Low : num [1:535168] 0.104 0.207 0.206 0.284 0.403 ...  
## $ Volume : num [1:535168] 5232600 2752410 12793800 52677500 16016500 ...

summary(cryptodata)

## Currencyname Date MarketCap   
## Length:535168 Min. :2013-12-27 00:00:00 Min. :0.000e+00   
## Class :character 1st Qu.:2015-09-27 00:00:00 1st Qu.:1.715e+04   
## Mode :character Median :2016-10-01 00:00:00 Median :1.081e+05   
## Mean :2016-07-14 05:40:24 Mean :7.169e+07   
## 3rd Qu.:2017-06-15 00:00:00 3rd Qu.:9.701e+05   
## Max. :2017-11-24 00:00:00 Max. :1.374e+11   
## NA's :13496 NA's :13496   
## Close Open High Low   
## Min. : 0.0 Min. : 0.0 Min. : 0.0 Min. : 0.0   
## 1st Qu.: 0.0 1st Qu.: 0.0 1st Qu.: 0.0 1st Qu.: 0.0   
## Median : 0.0 Median : 0.0 Median : 0.0 Median : 0.0   
## Mean : 88.5 Mean : 90.1 Mean : 102.3 Mean : 77.7   
## 3rd Qu.: 0.1 3rd Qu.: 0.1 3rd Qu.: 0.1 3rd Qu.: 0.1   
## Max. :793273.0 Max. :1013620.0 Max. :1146320.0 Max. :732467.0   
## NA's :13496 NA's :13496 NA's :13496 NA's :13496   
## Volume   
## Min. :0.000e+00   
## 1st Qu.:2.200e+01   
## Median :3.160e+02   
## Mean :2.111e+06   
## 3rd Qu.:5.952e+03   
## Max. :8.957e+09   
## NA's :13496

colnames(cryptodata)

## [1] "Currencyname" "Date" "MarketCap" "Close" "Open"   
## [6] "High" "Low" "Volume"

### The head() and tail() functions default to 6 rows, but we can adjust the number of rows using the "n = " argument  
  
head(cryptodata, n = 12)

## # A tibble: 12 x 8  
## Currencyname Date MarketCap Close Open High Low Volume  
## <chr> <dttm> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0x 2017-08-16 00:00:00 67034800 0.224 0.112 0.280 0.104 5232600  
## 2 0x 2017-08-17 00:00:00 133813000 0.207 0.223 0.239 0.207 2752410  
## 3 0x 2017-08-18 00:00:00 123335000 0.293 0.206 0.350 0.206 12793800  
## 4 0x 2017-08-19 00:00:00 176747000 0.479 0.295 0.544 0.284 52677500  
## 5 0x 2017-08-20 00:00:00 282777000 0.424 0.471 0.475 0.403 16016500  
## 6 0x 2017-08-21 00:00:00 253364000 0.352 0.422 0.444 0.328 15013800  
## 7 0x 2017-08-22 00:00:00 175773000 0.365 0.352 0.376 0.316 8045240  
## 8 0x 2017-08-23 00:00:00 182005000 0.387 0.364 0.411 0.364 8663860  
## 9 0x 2017-08-24 00:00:00 191353000 0.385 0.383 0.393 0.372 6227850  
## 10 0x 2017-08-25 00:00:00 193716000 0.362 0.387 0.394 0.346 6146620  
## 11 0x 2017-08-26 00:00:00 181011000 0.356 0.362 0.378 0.324 10377400  
## 12 0x 2017-08-27 00:00:00 177403000 0.352 0.355 0.357 0.334 4536030

tail(cryptodata, n = 6)

## # A tibble: 6 x 8  
## Currencyname Date MarketCap Close Open High Low  
## <chr> <dttm> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 zurcoin 2017-11-19 00:00:00 131769 0.00152 0.00153 0.00175 0.00151  
## 2 zurcoin 2017-11-20 00:00:00 131532 0.00156 0.00152 0.00157 0.00151  
## 3 zurcoin 2017-11-21 00:00:00 134581 0.00145 0.00156 0.00169 0.00144  
## 4 zurcoin 2017-11-22 00:00:00 125544 0.00189 0.00145 0.00330 0.00145  
## 5 zurcoin 2017-11-23 00:00:00 163521 0.00169 0.00189 0.00264 0.00169  
## 6 zurcoin 2017-11-24 00:00:00 145940 0.00197 0.00169 0.00238 0.00166  
## # ... with 1 more variable: Volume <dbl>

#Do you plan to slice and dice the data in different ways, create new variables, or join separate data frames to create new summary information? Explain.  
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

step\_1\_df <- select(cryptodata, -Volume)  
dim(step\_1\_df)

## [1] 535168 7

head(step\_1\_df, n = 10)

## # A tibble: 10 x 7  
## Currencyname Date MarketCap Close Open High Low  
## <chr> <dttm> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0x 2017-08-16 00:00:00 67034800 0.224 0.112 0.280 0.104  
## 2 0x 2017-08-17 00:00:00 133813000 0.207 0.223 0.239 0.207  
## 3 0x 2017-08-18 00:00:00 123335000 0.293 0.206 0.350 0.206  
## 4 0x 2017-08-19 00:00:00 176747000 0.479 0.295 0.544 0.284  
## 5 0x 2017-08-20 00:00:00 282777000 0.424 0.471 0.475 0.403  
## 6 0x 2017-08-21 00:00:00 253364000 0.352 0.422 0.444 0.328  
## 7 0x 2017-08-22 00:00:00 175773000 0.365 0.352 0.376 0.316  
## 8 0x 2017-08-23 00:00:00 182005000 0.387 0.364 0.411 0.364  
## 9 0x 2017-08-24 00:00:00 191353000 0.385 0.383 0.393 0.372  
## 10 0x 2017-08-25 00:00:00 193716000 0.362 0.387 0.394 0.346

#Arrange  
  
step\_2\_df <-step\_1\_df %>% arrange(step\_1\_df$MarketCap, step\_1\_df$CryptoCurrencyname)

## Warning: Unknown or uninitialised column: `CryptoCurrencyname`.

## Warning: Unknown or uninitialised column: `CryptoCurrencyname`.  
head(step\_2\_df)

## # A tibble: 6 x 7  
## Currencyname Date MarketCap Close Open High Low  
## <chr> <dttm> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 dixasset 2017-04-18 00:00:00 0 0.0000006 0.00000061 6.1e-7 5.8e-7  
## 2 dixasset 2017-04-19 00:00:00 0 0.00000075 0.00000074 7.9e-7 7.3e-7  
## 3 dixasset 2017-04-20 00:00:00 0 0.00000072 0.00000075 8 e-7 6.7e-7  
## 4 dixasset 2017-04-21 00:00:00 0 0.00000062 0.00000072 7.2e-7 6.1e-7  
## 5 dixasset 2017-04-22 00:00:00 0 0.00000064 0.00000063 6.6e-7 6.3e-7  
## 6 dixasset 2017-04-23 00:00:00 0 0.00000064 0.00000065 6.5e-7 6.3e-7

#How could you summarize your data to answer key questions?  
filter\_df <- cryptodata %>% filter(Currencyname == "dixasset")  
filter\_df

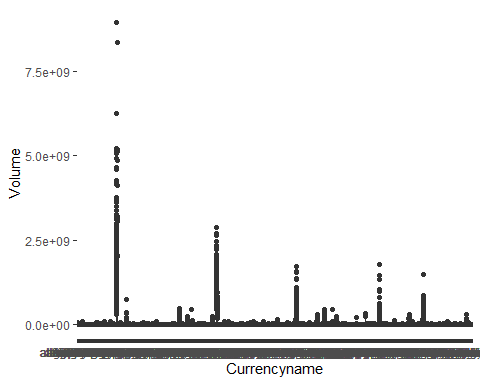
## # A tibble: 293 x 8  
## Currencyname Date MarketCap Close Open High Low Volume  
## <chr> <dttm> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 dixasset 2017-01-15 00:00:00 2554 1.66 1.7 1.73 1.56 180  
## 2 dixasset 2017-01-16 00:00:00 2497 1.47 1.66 1.68 1.46 213  
## 3 dixasset 2017-01-17 00:00:00 2190 1.44 1.46 1.5 1.41 209  
## 4 dixasset 2017-01-18 00:00:00 2160 1.44 1.44 1.44 1.37 407  
## 5 dixasset 2017-01-19 00:00:00 2145 1.48 1.43 1.51 1.41 419  
## 6 dixasset 2017-01-20 00:00:00 2222 1.43 1.48 1.49 1.41 140  
## 7 dixasset 2017-01-21 00:00:00 2135 1.42 1.42 1.45 1.39 139  
## 8 dixasset 2017-01-22 00:00:00 2134 1.42 1.42 1.45 1.39 1  
## 9 dixasset 2017-01-23 00:00:00 2131 1.33 1.42 1.42 1.31 123  
## 10 dixasset 2017-01-24 00:00:00 1998 1.3 1.33 1.34 1.27 121  
## # ... with 283 more rows

Mean\_Market\_cap\_df <- summarize(filter\_df,mean(MarketCap,na.rm=TRUE))  
  
cryptodata\_Group <- group\_by(cryptodata,Currencyname)  
summarize(cryptodata\_Group,avg= mean(MarketCap,na.rm=TRUE))

## # A tibble: 1,056 x 2  
## Currencyname avg  
## <chr> <dbl>  
## 1 0x 116992351.  
## 2 1337coin 415734.  
## 3 2give 944053.  
## 4 300token 69380.  
## 5 42coin 78702.  
## 6 808coin 239983.  
## 7 8bit 79521.  
## 8 abjcoin 7371621.  
## 9 abncoin 18156.  
## 10 achain 32667383.  
## # ... with 1,046 more rows

#What types of plots and tables will help you to illustrate the findings to your questions? Ensure #that all graph plots have axis titles, legend if necessary, scales are appropriate, appropriate #geoms used, etc.).  
library(ggplot2)  
ggplot(data = cryptodata\_Group, aes(x=Currencyname,y=Volume)) + geom\_boxplot()

## Warning: Removed 13496 rows containing non-finite values (stat\_boxplot).



#  
#Machine Learning technique that helps in answering the question and the group to which the data belongs to is Classification.  
  
#Classification in machine learning is a process of identifying to which of the set a group of experimental observations belongs.  
  
#This is mainly done with the help of training set which contains pre-installed data and the those data are classified on the basis of some categories.  
  
#Classification is very important for solving complex mathematical and statistical problems.

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot. References install.packages(“knitr”)