Język R część 2 - zadania projektowe

Ola Jaglińska

Zadanie 1

Plik crypto.rds zawiera notowania wybranych kryptowalut. 1.Wczytaj dane crypto.rds do R i zapoznaj się z nimi.

```
kryptowaluty <- readRDS('./crypto.rds')
saveRDS(kryptowaluty, file = './crypto.rds')
str(kryptowaluty)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    83153 obs. of  8 variables:
##  $ Currency  : chr  "0x" "0x" "0x" "0x" ...
##  $ Date      : chr  "Feb 11, 2018" "Feb 10, 2018" "Feb 09, 2018" "Feb 08, 2018" ...
##  $ Open      : num  1.09 1.14 1.08 0.989 1.01 ...
##  $ High      : num  1.09 1.2 1.15 1.13 1.16 1.03 1.16 1.39 1.5 1.61 ...
##  $ Low       : num  0.934 0.986 1.01 0.989 0.902 ...
##  $ Close     : num  0.979 1.1 1.14 1.07 0.993 ...
##  $ Volume    : chr  "4,888,770" "10,828,700" "5,979,420" "12,992,800" ...
##  $ Market.Cap: chr  "555,363,000" "576,535,000" "545,842,000" "501,142,000" ...
```

2.Wybierz z danych tylko te wiersze, które dotyczą Bitcoina.

```
search()
```

```
## [1] ".GlobalEnv"        "package:stats"     "package:graphics"
## [4] "package:grDevices" "package:utils"     "package:datasets"
## [7] "package:methods"   "Autoloads"         "package:base"
```

```
install.packages('tidyverse')
```

```
## Installing package into '/home/mion/s/284/ojaglins/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library(magrittr)
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------- tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```
library(dplyr)
```

```
kryptowaluty %>% dplyr::select(tidyselect::contains("bitcoin"))
```

```
## # A tibble: 83,153 x 0
```

3.Pozostaw w danych tylko kolumny Date i Close.

```
kryptowaluty %>% select(Date, Close)
```

```
## # A tibble: 83,153 x 2
##    Date         Close
##    <chr>        <dbl>
##  1 Feb 11, 2018 0.979
##  2 Feb 10, 2018 1.1
##  3 Feb 09, 2018 1.14
##  4 Feb 08, 2018 1.07
##  5 Feb 07, 2018 0.993
##  6 Feb 06, 2018 1.02
##  7 Feb 05, 2018 0.817
##  8 Feb 04, 2018 1.15
##  9 Feb 03, 2018 1.39
## 10 Feb 02, 2018 1.3
## # ... with 83,143 more rows
```

4.Popraw kolumnę Date w taki sposób, aby była typu Date.

```
kryptowaluty$Date <- as.Date(kryptowaluty$Date, format = "%b%d,%Y")
class(kryptowaluty$Date)
```

```
## [1] "Date"
```

5.Stwórz kolumnę Rate na podstawie kolumny Close zgodnie z następującą definicją()

```
definition_Rate <- (kryptowaluty$Close - (kryptowaluty$Close-1))/(kryptowaluty$Close-1)
kryptowaluty_new <- kryptowaluty %>% mutate(.data = kryptowaluty, Rate = definition_Rate, .after = Close
```

6.Posortuj dane według kolumny Rate w porządku malejącym.

```
kryptowaluty_new %>% arrange(desc(Rate))
```

```
## # A tibble: 83,153 x 9
##    Currency     Date        Open  High   Low Close  Rate Volume      Market.Cap
##    <chr>        <date>     <dbl> <dbl> <dbl> <dbl> <dbl> <chr>       <chr>
##  1 aelf         2017-12-31 0.878  1.03 0.854     1   Inf 125,966,000 219,551,000
##  2 bitcoindark  2015-04-12 1.04   1.06 0.984     1   Inf 1,545       1,264,660
##  3 bitcoindark  2015-04-04 1.09   1.11 0.953     1   Inf 3,658       1,322,910
##  4 bitcoindark  2015-02-03 1.1    1.2  0.936     1   Inf 11,129      1,327,400
##  5 civic        2018-01-10 1.11   1.11 0.953     1   Inf 26,788,600  380,067,000
##  6 counterparty 2015-11-12 0.926  1.02 0.868     1   Inf 2,015       2,442,170
##  7 edgeless     2018-02-03 0.863  1.04 0.803     1   Inf 2,520,960   70,787,000
##  8 ethereum     2015-10-28 0.871  1.06 0.808     1   Inf 2,373,050   64,676,600
##  9 ethos        2017-08-24 0.984  1.07 0.787     1   Inf 1,022,480   69,657,800
## 10 factom       2016-02-26 1.04   1.05 0.975     1   Inf 168,408     9,117,650
## # ... with 83,143 more rows
```