Analise Weg

Carlos E. Carvalho

1/20/2021

Loading to files about the company WEG:

3t20_dre.csv: file with the company results #https://ri.weg.net/informacoes-financeiras/planilhas/

WEGE3SA.csv: file with stock market prices and volume https://br.financas.yahoo.com/quote/WEGE3.SA/history?period1=1262217600&period2=1611014400&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true

I am trying to see how the stock prices react when the company results are presented. Everything I did here, I've learned in the course Formação Cientista de Dados, at DataScience Academy https://www.datascienceacademy.com.br/bundles?bundle_id=formacao-cientista-de-dados

Carlos Eduardo Carvalho carlos.e.carvalho@gmail.com Linkedin: https://www.linkedin.com/in/carlos-carvalho-93204b13/ https://github.com/CarlosCarvalho1981/Analise_Weg.git

```r
setwd("D:/CIENTISTA_DADOS/WEG")
getwd()
```

```
## [1] "D:/CIENTISTA_DADOS/WEG"
```

Loading packages

```
##
## 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

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

##
## Attaching package: 'reshape2'
## The following objects are masked from ’package:data.table’:
##
##     dcast, melt
```

Loading the file

```
dre <- read_csv("3t20_dre.csv")
```

```
## Warning: Missing column names filled in: ’X1’ [1], ’X2’ [2], ’X3’ [3], ’X4’ [4],
## ’X5’ [5], ’X6’ [6], ’X7’ [7], ’X8’ [8], ’X9’ [9], ’X10’ [10], ’X11’ [11],
## ’X12’ [12], ’X13’ [13], ’X14’ [14], ’X15’ [15], ’X16’ [16], ’X17’ [17],
## ’X18’ [18], ’X19’ [19], ’X20’ [20], ’X21’ [21], ’X22’ [22], ’X23’ [23],
## ’X24’ [24], ’X25’ [25], ’X26’ [26], ’X27’ [27], ’X28’ [28], ’X29’ [29],
## ’X30’ [30], ’X31’ [31], ’X32’ [32], ’X33’ [33], ’X34’ [34], ’X35’ [35],
## ’X36’ [36], ’X37’ [37], ’X38’ [38], ’X39’ [39], ’X40’ [40], ’X41’ [41],
## ’X42’ [42], ’X43’ [43], ’X44’ [44], ’X45’ [45], ’X46’ [46]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character()
## )
## i Use ‘spec()‘ for the full column specifications.
```

```
dre
```

```
## # A tibble: 31 x 46
##    X1    X2    X3    X4    X5    X6    X7    X8    X9    X10   X11   X12   X13
##    <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
##  1 "C\x~ "Des~ 12/3~ 3/31~ 6/30~ 9/30~ 12/3~ 3/31~ 6/30~ 9/30~ 12/3~ 3/31~ 6/30~
##  2 "3.0~ "Rec~ 4,21~ 931,~ 1,01~ 1,18~ 1,25~ 1,12~ 1,27~ 1,31~ 1,46~ 1,36~ 1,52~
##  3 "3.0~ "Cus~ -2,8~ -623~ -703~ -811~ -867~ -815~ -895~ -899~ -1,0~ -977~ -1,0~
##  4 "3.0~ "Res~ 1,35~ 308,~ 309,~ 377,~ 391,~ 310,~ 381,~ 418,~ 445,~ 391,~ 461,~
##  5 "3.0~ "Des~ -733~ -169~ -180~ -210~ -224~ -188~ -216~ -224~ -262~ -243~ -259~
##  6 "3.0~ "Des~ -408~ -93,~ -100~ -121~ -119~ -116~ -123~ -129~ -141~ -142~ -155~
##  7 "3.0~ "Des~ -225~ -57,~ -65,~ -71,~ -68,~ -58,~ -63,~ -66,~ -69,~ -67,~ -75,~
##  8 "3.0~ "Hon~ -13,~ -3,9~ -4,1~ -3,9~ -5,3~ -3,3~ -3,4~ -3,4~ -3,4~ -3,8~ -3,7~
##  9 "3.0~ "Out~ -211~ -53,~ -60,~ -67,~ -63,~ -54,~ -60,~ -62,~ -66,~ -63,~ -71,~
## 10 "3.0~ "Out~ 12,3~ 8,515 2,116 5,541 3,926 8,671 1,995 479   5,927 4,958 8,236
## # ... with 21 more rows, and 33 more variables: X14 <chr>, X15 <chr>,
## #   X16 <chr>, X17 <chr>, X18 <chr>, X19 <chr>, X20 <chr>, X21 <chr>,
## #   X22 <chr>, X23 <chr>, X24 <chr>, X25 <chr>, X26 <chr>, X27 <chr>,
## #   X28 <chr>, X29 <chr>, X30 <chr>, X31 <chr>, X32 <chr>, X33 <chr>,
## #   X34 <chr>, X35 <chr>, X36 <chr>, X37 <chr>, X38 <chr>, X39 <chr>,
## #   X40 <chr>, X41 <chr>, X42 <chr>, X43 <chr>, X44 <chr>, X45 <chr>, X46 <chr>
```

There are some problems with the way the rows and columns are. Let’s see the class of the object

```
class(dre)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Let’s remove the first column that won’t help us in any way.

```
dre$X1 <- NULL
```

Let’s swap the rows and the columns.

```
inv_dre <- t(dre)
```

Now it became a matrix

```
class(inv_dre)
```

```
## [1] "matrix"
```

Let’s turn it into a data frame

```
inv_dre <- as.data.frame(inv_dre)
class(inv_dre)
```

```
## [1] "data.frame"
```

Since the name of some columns have characters that aren’t recognized, let’s rename everything.

```
colnames(inv_dre) <- c("Data", "Receita de Venda de Bens e/ou Servicos", "Custo dos Bens e/ou Servicos
                       "Resultado Bruto" , "Despesas/Receitas Operacionais", "Despesas com Vendas", "De
                       "Honorarios dos Administradores", "Outras Despesas Administrativas", "Outras Rec
                       "Resultado de Equivalencia Patrimonial", "Resultado Antes do Resultado Financeir
                       "Resultado Financeiro", "Receitas Financeiras", "Despesas Financeiras", "Resulta
                       "Imposto de Renda e Contribuicao Social sobre o Lucro", "Corrente", "Diferido",
                       "Lucro/Prejuizo Consolidado por Periodo", "Atribuido a Socios da Empresa Controla
                       "Lucro por Acao - (Reais/Acao", "Lucro Basico por Acao", "ON", "Lucro Diluido po
```

We still have a row that is not usefull. Let’s remove it

```
inv_dre <- inv_dre[-c(1),]
```

And also remove the last 2 columns

```
inv_dre <- inv_dre[,-c(30,31)]
```

Now we have a good data frame! Let’s take the columns Data and Lucro/Prejuizo Consolidado por Periodo.

```
sub_dre <- inv_dre[,c("Data", "Lucro/Prejuizo Consolidado por Periodo")]
```

How does this subset looks like?

```
head(sub_dre)
```

```
##           Data Lucro/Prejuizo Consolidado por Periodo
## X3 12/31/2009                                 559,937
## X4  3/31/2010                                 120,459
## X5  6/30/2010                                 116,956
## X6  9/30/2010                                 149,419
## X7 12/31/2010                                 146,800
## X8  3/31/2011                                 124,259
```

Let’s see the types of the columns

```r
glimpse(sub_dre)
```

```
## Rows: 44
## Columns: 2
## $ Data                                      <fct> 12/31/2009, 3/31/2010, 6/3...
## $ ‘Lucro/Prejuizo Consolidado por Periodo‘ <fct> "559,937", "120,459", "116...
```

Both variables are factors. We’re going to need the variable Data as a date format. Let’s extract a vector to make the conversion

```r
sub_dre1 <- sub_dre$Data
Data2 <- as.Date(sub_dre1, format = "%m/%d/%Y")
glimpse(sub_dre1)
```

```
##  Factor w/ 45 levels "12/31/2009","12/31/2010",..: 1 12 23 34 2 13 24 35 3 14 ...
##  - attr(*, "names")= chr [1:44] "X3" "X4" "X5" "X6" ...
```

And we put it back together

```r
sub_dre <- cbind(sub_dre, Data2)
head(sub_dre)
```

```
##           Data Lucro/Prejuizo Consolidado por Periodo      Data2
## X3 12/31/2009                                 559,937 2009-12-31
## X4  3/31/2010                                 120,459 2010-03-31
## X5  6/30/2010                                 116,956 2010-06-30
## X6  9/30/2010                                 149,419 2010-09-30
## X7 12/31/2010                                 146,800 2010-12-31
## X8  3/31/2011                                 124,259 2011-03-31
```

We have now columns with the same information. Let’s take off the one we don’t need anymore.

```r
sub_dre <- sub_dre[-c(1)]
```

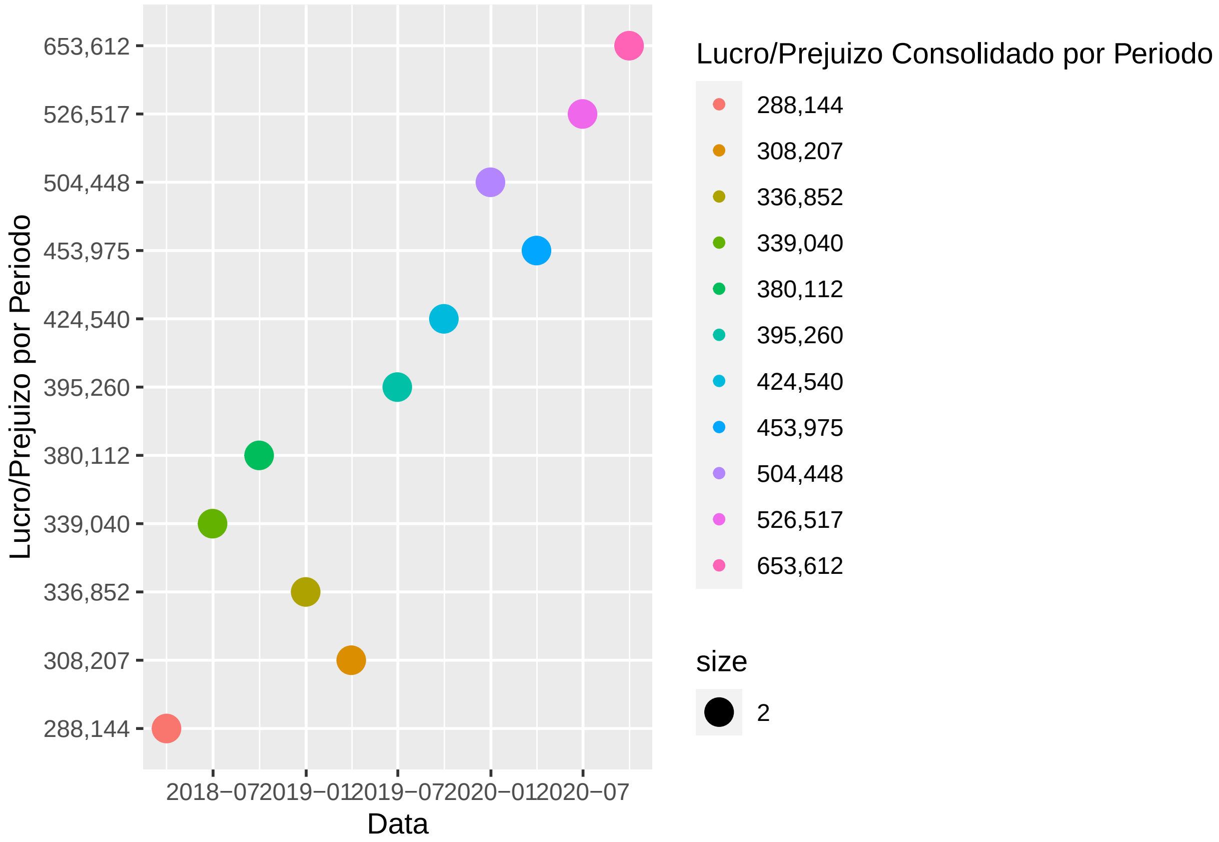
In order to make a plot, let’s take only the last 10 rows.

```r
sub_dre2 <- sub_dre[34:44,]
sub_dre2
```

```
##     Lucro/Prejuizo Consolidado por Periodo      Data2
## X36                                288,144 2018-03-31
## X37                                339,040 2018-06-30
## X38                                380,112 2018-09-30
## X39                                336,852 2018-12-31
## X40                                308,207 2019-03-31
## X41                                395,260 2019-06-30
## X42                                424,540 2019-09-30
## X43                                504,448 2019-12-31
## X44                                453,975 2020-03-31
## X45                                526,517 2020-06-30
## X46                                653,612 2020-09-30
```

We are going to see a scatter plot where the colors of the points are related to the column `Lucro/Prejuizo Consolidado por Periodo`

```
ggplot(data = sub_dre2, aes(x=Data2, y = `Lucro/Prejuizo Consolidado por Periodo`, colour = `Lucro/Preju
  geom_point(aes(size = 2)) + xlab("Data") + ylab("Lucro/Prejuizo por Periodo")
```



Now we're going to load the file with WEG stock prices

```
acoes <- read_csv("WEGE3SA.csv", col_types = list(
  Date = col_date(),
  Open = col_double(),
  High = col_double(),
  Low = col_double(),
  Close = col_double(),
  'Adj Close' = col_double(),
  Volume = col_integer()
))
```

```
## Warning: 12 parsing failures.
## row       col expected actual          file
## 228 Open      a double   null ’WEGE3SA.csv’
## 228 High      a double   null ’WEGE3SA.csv’
## 228 Low       a double   null ’WEGE3SA.csv’
## 228 Close     a double   null ’WEGE3SA.csv’
## 228 Adj Close a double   null ’WEGE3SA.csv’
## ... ......... ........ ...... .............
## See problems(...) for more details.
```

The file has some NULL rows that’s why we’ve got those warnings. Let’s see.

```
acoes
```

```
## # A tibble: 691 x 7
##    Date        Open  High   Low Close ‘Adj Close‘  Volume
##    <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>   <int>
##  1 2018-04-02  17.5  17.5  17.1  17.2        16.6 1794260
##  2 2018-04-03  17.3  17.3  17.0  17.2        16.6 1477190
##  3 2018-04-04  16.9  17.4  16.8  17.2        16.6 2360410
##  4 2018-04-05  17.5  17.5  17.0  17.1        16.5 1966770
##  5 2018-04-06  17.1  17.2  16.7  16.7        16.2 2686190
##  6 2018-04-09  16.8  16.9  16.4  16.4        15.8 2011620
##  7 2018-04-10  16.5  16.7  16.4  16.5        16.0 1948570
##  8 2018-04-11  16.4  16.9  16.4  16.6        16.0 1777880
##  9 2018-04-12  16.8  16.9  16.4  16.8        16.2 2148380
## 10 2018-04-13  16.7  17.1  16.5  17.0        16.4 2572050
## # ... with 681 more rows
```

The NULL rows became NA. Let’s drop those NA rows

```
acoes <- acoes[complete.cases(acoes), ]
```

Let’s get a subset with the last 30 rows, just to see...

```
acoes30 <- acoes[661:691,]
```

```
## Warning: The ‘i‘ argument of ‘‘[.tbl_df‘()‘ must lie in [0, rows] if positive, as of tibble 3.0.0.
## Use ‘NA_integer_‘ as row index to obtain a row full of ‘NA‘ values.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```

#The last 2 are NA to.

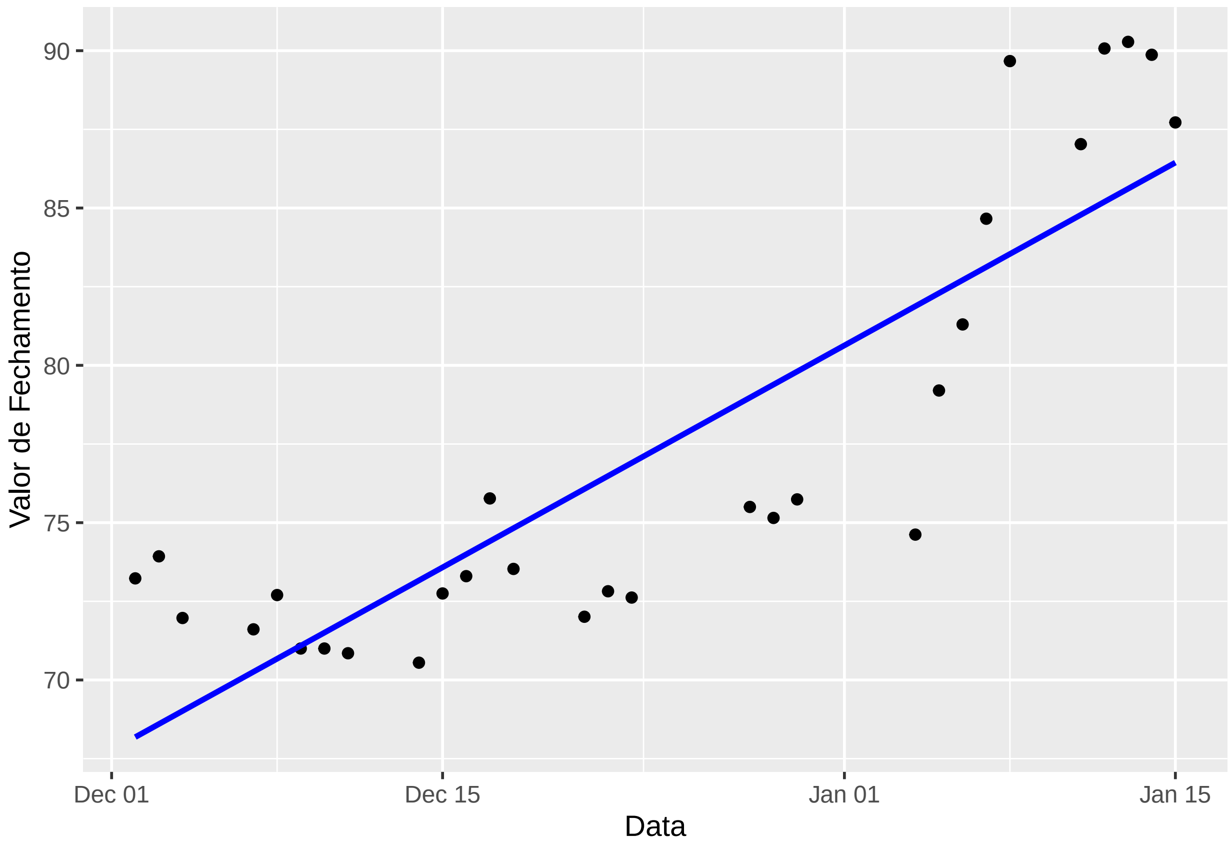
```
acoes30 <- acoes30[complete.cases(acoes30), ]
acoes30
```

```
## # A tibble: 29 x 7
##    Date        Open  High   Low Close `Adj Close`  Volume
##    <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>   <int>
##  1 2020-12-02  74.5  74.5  71.8  73.2        73.2 6208200
##  2 2020-12-03  73.3  75.0  73.2  73.9        73.9 5158100
##  3 2020-12-04  74.1  74.4  71.9  72.0        71.9 4840800
##  4 2020-12-07  72.0  73.5  70.7  71.6        71.6 6428800
##  5 2020-12-08  72.1  72.8  71.3  72.7        72.7 4669300
##  6 2020-12-09  72.7  73.1  70.7  71          71.0 3200100
##  7 2020-12-10  71.2  71.9  69.5  71          71.0 4951300
##  8 2020-12-11  70.8  71.2  70    70.8        70.8 1819200
##  9 2020-12-14  71.4  71.8  70.2  70.6        70.5 2911200
## 10 2020-12-15  70.6  72.8  68.3  72.8        72.7 8882400
## # ... with 19 more rows
```

#We are going to see a scatter plot of the stock prices with a nice linear regression.

```
ggplot(data = acoes30, aes(x = Date, y = Close))+
  geom_point() + xlab("Data") + ylab("Valor de Fechamento") +
  geom_smooth(method = lm, color = "blue", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



We can see that there's a tendency of growing on the prices Now we're going to extract some vectors to work with. We are going to take the dates that are in both data frames: sub_dre2 and acoes. Or, at least, the next dates. Had to do that manually.

```
acoes2 <- acoes[c(1,63,126,185,244,306,371,432,493,554,619),]
acoes2
```

```
## # A tibble: 11 x 7
##    Date        Open  High   Low Close ‘Adj Close‘  Volume
##    <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>   <int>
##  1 2018-04-02  17.5  17.5  17.1  17.2        16.6 1794260
##  2 2018-06-29  16.2  16.4  16.1  16.2        15.7 3207700
##  3 2018-09-28  19.7  19.9  19.6  19.8        19.2 2015800
##  4 2018-12-28  17.2  17.9  17.1  17.5        17.1 3590500
##  5 2019-03-29  18.2  18.3  17.9  18          17.7 2190900
##  6 2019-06-28  21.6  21.8  21.4  21.4        21.0 2387600
##  7 2019-09-30  24.2  24.5  24.1  24.2        24.0 1992200
##  8 2019-12-30  35.4  35.5  34.7  34.7        34.4 2919100
##  9 2020-03-31  34.1  35.4  33.0  33.6        33.4 7063300
## 10 2020-06-30  49.7  51.5  49.5  50.6        50.5 5538000
## 11 2020-09-30  65.2  65.7  64    65.7        65.7 6374500
```

```
Close2 <- acoes2$Close
Volume2 <- acoes2$Volume
Close2
```

```
##  [1] 17.1692 16.2500 19.7500 17.5400 18.0000 21.3600 24.2300 34.6600 33.5700
## [10] 50.6100 65.7000
```

Let's have a column with the range of values instead of the real values. First, looking for the max and min values of the stock prices.

```
maxClose <- max(acoes$Close, na.rm = TRUE)
maxClose
```

```
## [1] 90.28
```

```
minClose <- min(acoes$Close, na.rm = TRUE)
minClose
```

```
## [1] 15.21
```

```
groupClose <- function(fechamento){
  if(fechamento > 15.00 & fechamento <= 30.00){
    return(30)
  }else if(fechamento > 30.00 & fechamento <= 45.00){
    return(45)
  }else if(fechamento > 45.00 & fechamento <= 60.00){
    return(60)
  }else if(fechamento > 60.00 & fechamento <= 75.00){
    return(75)
  }else if(fechamento > 75.00 & fechamento <= 90.00){
    return(90)
  }else{
    return(95)
  }
}

acoes2$CloseGroup <- sapply(acoes2$Close, groupClose)
View(acoes2)

CloseGroup2 <- acoes2$CloseGroup
```

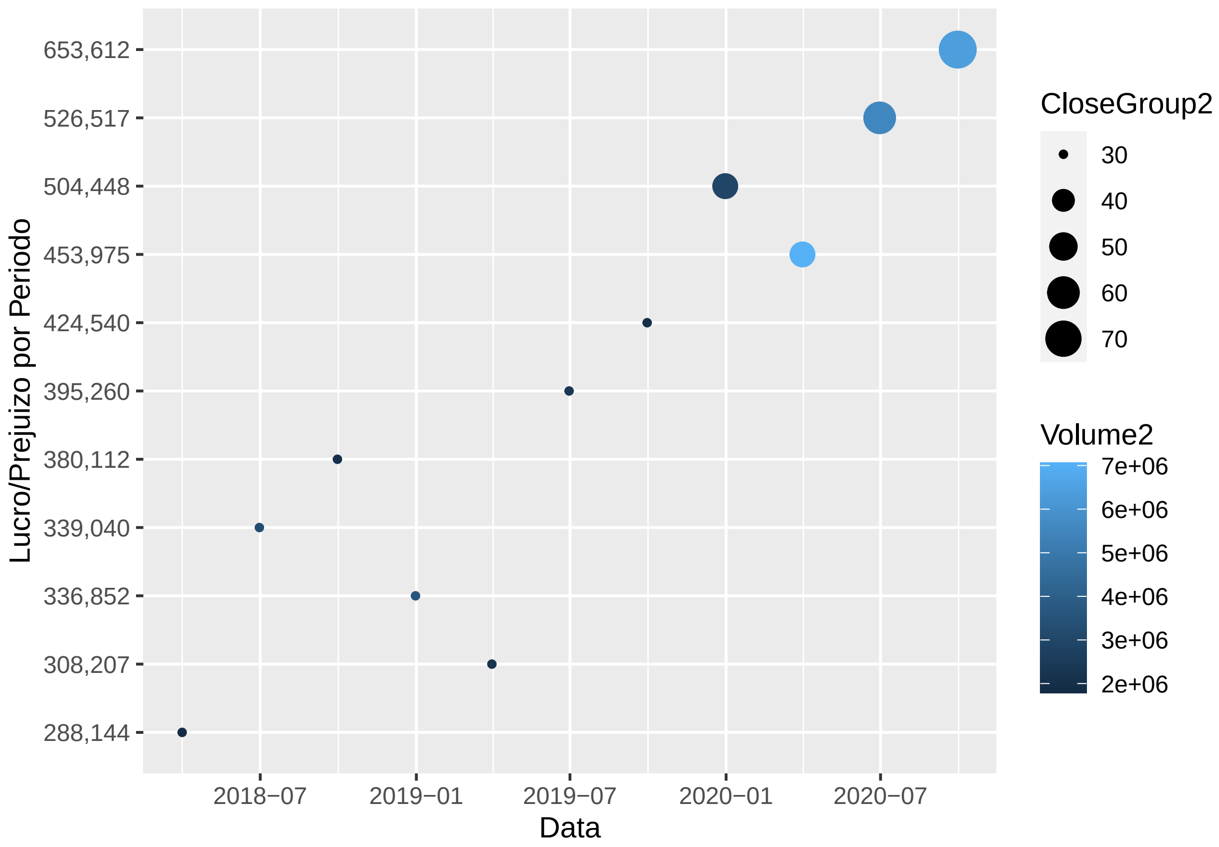
And now we insert the vectors into the data frame

```
sub_dre2 <- cbind(sub_dre2, Close2)
sub_dre2 <- cbind(sub_dre2, Volume2)
sub_dre2 <- cbind(sub_dre2, CloseGroup2)
colnames(sub_dre2) <- c("Lucro/Prejuizo", "Data2", "Close2", "Volume2", "CloseGroup2")
head(sub_dre2)
```

```
##     Lucro/Prejuizo      Data2  Close2 Volume2 CloseGroup2
## X36        288,144 2018-03-31 17.1692 1794260          30
## X37        339,040 2018-06-30 16.2500 3207700          30
## X38        380,112 2018-09-30 19.7500 2015800          30
## X39        336,852 2018-12-31 17.5400 3590500          30
## X40        308,207 2019-03-31 18.0000 2190900          30
## X41        395,260 2019-06-30 21.3600 2387600          30
```

Let's see a scatter plot of Lucro/Prejuizo Consolidado por Periodo, with the size of the dots as the closing stock prices and the color as the volume.

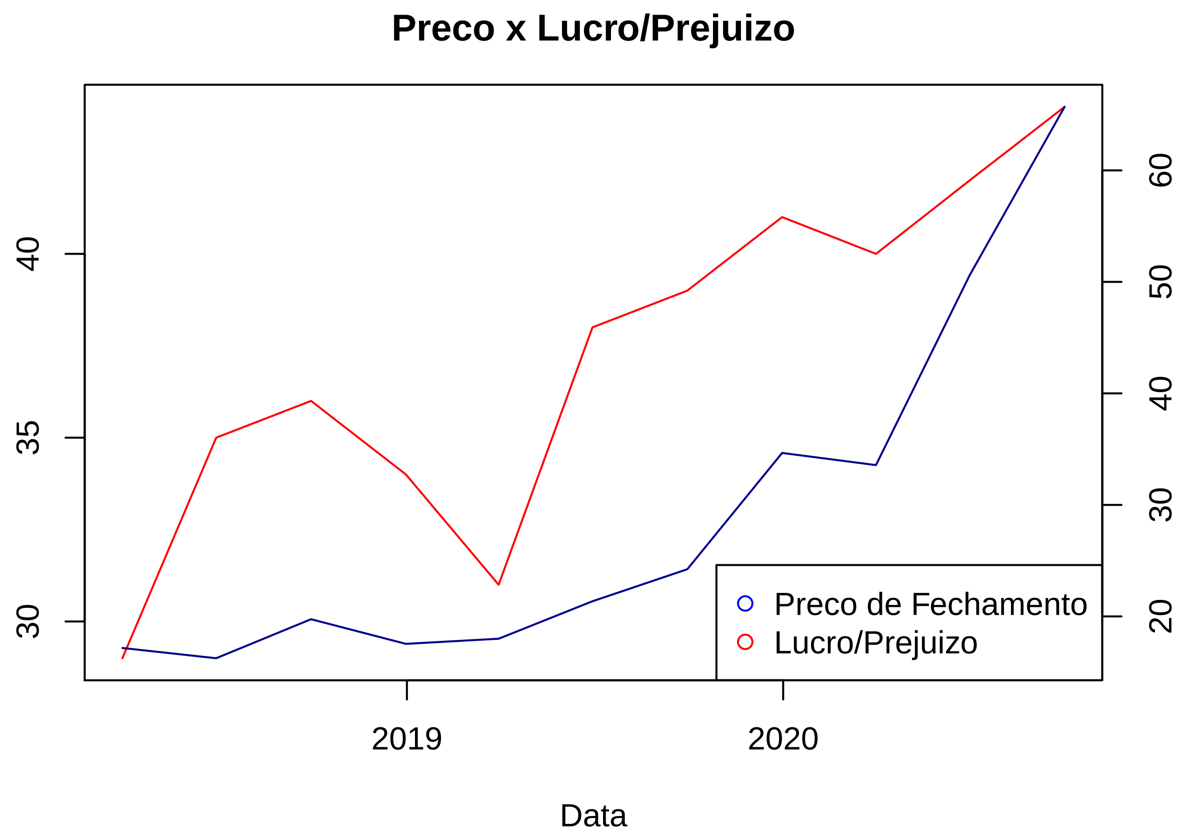
```
ggplot(data = sub_dre2, aes(x=Data2, y = `Lucro/Prejuizo`))+
  geom_point(aes(size = CloseGroup2, color = Volume2)) + xlab("Data") + ylab("Lucro/Prejuizo por Periodo
```



Now we are going to plot two line plots: Lucro/Prejuizo and Preco de Fechamento. We want to see if the behavior of the stock price is the same as the behavior of the results of the company.

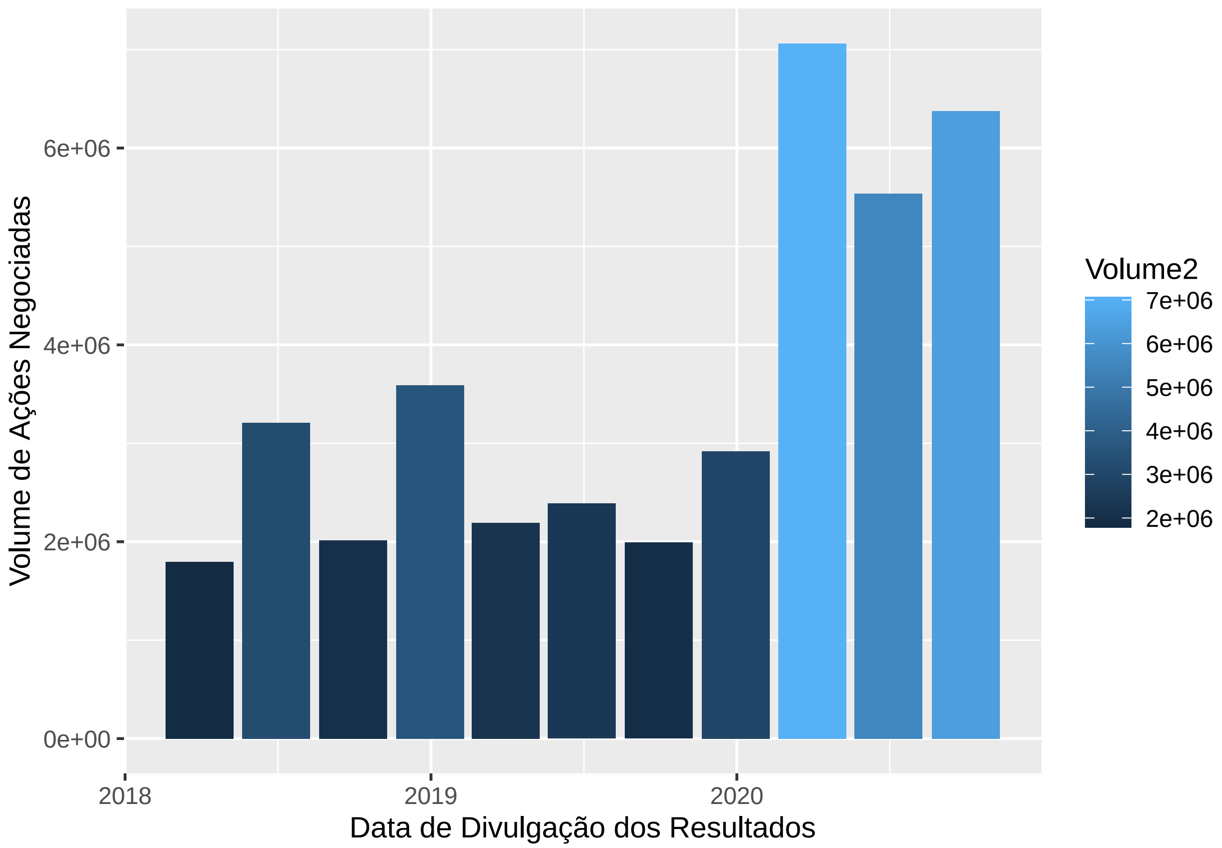
```
par(mar = c(4,3,3,3), col.axis = "black", adj = 0.5)
plot(sub_dre2$Data2, sub_dre2$`Lucro/Prejuizo`, type = "l", col = "red", bty = "o",
     xlab = "Data", ylab = "Lucro/Prejuizo")
par(new = T)

plot(sub_dre2$Data2, sub_dre2$Close2, type = "l", ann = F, axes = F, col = "darkblue", ylab = "Preco")
axis(side = 4)
legend("bottomright", pch = 1, col = c("blue", "red"), legend = c("Preco de Fechamento", "Lucro/Prejuiz
par(mfrow=c(1,1))
title(main = "Preco x Lucro/Prejuizo")
```



Looks like it is ! Now let's look at the volume of the company stocks during the days the results of the company were presented.

```
ggplot(data = sub_dre2, aes(x = Data2, y = Volume2 )) +
  geom_col(aes(fill = Volume2)) +
  ylab("Volume de Ações Negociadas") + xlab("Data de Divulgação dos Resultados")
```



Now let's create another dataset, with dates next to the date when the results of the company were given.

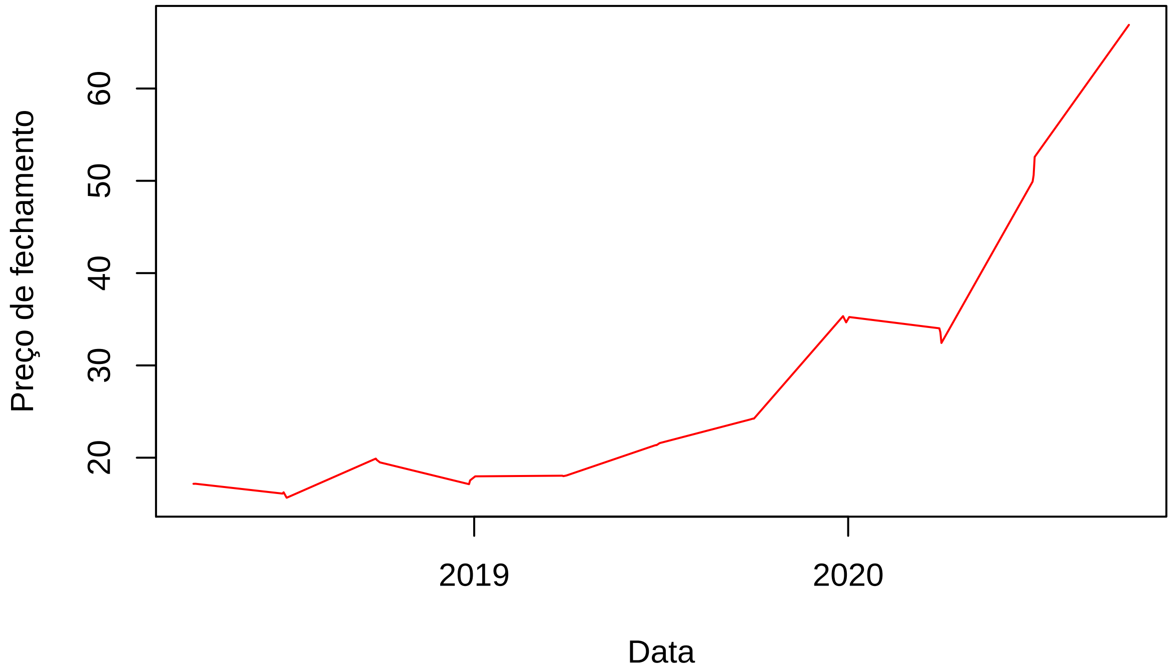
```
acoes3 <- acoes[c(1,2,3,62,63,64,125,126,127,184,185,186,243,244,245,305,306,307,370,371,372,431,432,43
```

Dropping the NA rows.

```
acoes3 <- acoes3[complete.cases(acoes3), ]
View(acoes3)
```

Creating the last plot to see the stock prices near the days when the results came:

```
plot(acoes3$Date, acoes3$Close, type = "l", col = "red", bty = "o",
     xlab = "Data", ylab = "Preço de fechamento")
```


Preço de fechamento
20
30
40
50
60
2019
2020
Data