

Introdução à Análise de Ações com R



Introdução à Análise de Ações com R

A análise de ações e investimentos é um tema que pode ser muito bem explorado na programação.

Isso inclui a linguagem R, que já possui vasta literatura, pacotes e funções desenvolvidas no tema.

Nesta aula faremos uma breve introdução ao assunto utilizando os pacotes `quantmod` e `ggplot2`.

Como será nossa análise

A ação analisada será a da Petrobras, com dados extraídos do site [Yahoo Finance](https://finance.yahoo.com/) (<https://finance.yahoo.com/>), por meio do pacote [quantmod](https://cran.r-project.org/web/packages/quantmod/index.html) (<https://cran.r-project.org/web/packages/quantmod/index.html>), que é bastante utilizado para a modelagem quantitativa de dados financeiros.

Além disso, o pacote [ggplot2](https://cran.r-project.org/web/packages/ggplot2/index.html) (<https://cran.r-project.org/web/packages/ggplot2/index.html>) será utilizado para a visualização dos dados.

Preparando o ambiente

Primeiramente, instalamos e carregamos os pacotes necessários.

```
In [80]: rm(list=ls())
install.packages("quantmod")
install.packages('Rcpp', dependencies = TRUE)
install.packages('ggplot2', dependencies = TRUE)
install.packages("vctrs", dependencies = TRUE)
```

Warning message:

"package 'quantmod' is in use and will not be installed"

Installing package into 'C:/Users/domin/OneDrive/Documentos/R/win-library/4.0'
(as 'lib' is unspecified)

package 'Rcpp' successfully unpacked and MD5 sums checked

Warning message:

"cannot remove prior installation of package 'Rcpp'"

Warning message in file.copy(savedcopy, lib, recursive = TRUE):

"problem copying C:\Users\domin\OneDrive\Documentos\R\win-library\4.0\00LOCK\Rcpp\libs\x64\Rcpp.dll to C:\Users\domin\OneDrive\Documentos\R\win-library\4.0\Rcpp\libs\x64\Rcpp.dll: Permission denied"

Warning message:

"restored 'Rcpp'"

The downloaded binary packages are in

C:\Users\domin\AppData\Local\Temp\RtmpsjKU1M\downloaded_packages

Warning message:

"package 'ggplot2' is in use and will not be installed"

Installing package into 'C:/Users/domin/OneDrive/Documentos/R/win-library/4.0'
(as 'lib' is unspecified)

package 'vctrs' successfully unpacked and MD5 sums checked

Warning message:

"cannot remove prior installation of package 'vctrs'"

```
Warning message in file.copy(savedcopy, lib, recursive = TRUE):  
"problem copying C:\Users\domin\OneDrive\Documentos\R\win-library\4.0\00LOCK\vctr\li  
bs\x64\vctr.dll to C:\Users\domin\OneDrive\Documentos\R\win-library\4.0\vctr\libs\x  
64\vctr.dll: Permission denied"  
Warning message:  
"restored 'vctr'"  
The downloaded binary packages are in  
  C:\Users\domin\AppData\Local\Temp\RtmpsjKU1M\downloaded_packages
```

Agora vamos carregar os pacotes instalados

```
In [81]: library(quantmod)
library(ggplot2)
```


Depois de termos os pacotes instalados e carregados, vamos baixar a série de preços da ação e tratar os dados para que fiquem da melhor forma possível para a realização das análises. Isso é feito por meio do comando `getSymbols`.

```
In [144]: pbr <- getSymbols("PBR", src = "yahoo", from = "2013-01-01", to = "2020-06-01", auto.assign = FALSE)
```

O primeiro argumento `PBR` é o símbolo do ativo que queremos analisar no Yahoo Finance. Você pode procurar pelo símbolo de outra ação ou índice que desejar analisar clicando [aqui \(https://finance.yahoo.com/lookup/\)](https://finance.yahoo.com/lookup/).

O argumento `src = "yahoo"` indica a fonte dos dados. Também podemos utilizar outras fontes como o Google Finance, FRED, Oanda, bancos de dados locais, CSVs e vários outros.

O terceiro e quarto argumentos indicam o período no qual os dados serão extraídos, com a data no formato “aaaa-mm-dd”.

Por último, o argumento `auto.assign = FALSE` nos permite nomear o dataset com o

nome que quisermos. Caso seja TRUE, o nome será automaticamente o símbolo que estamos buscando, ou seja, o primeiro argumento.

Adendo

Caso a API do Yahoo Finance não esteja funcionando, é possível baixar a base de preços diretamente do site.

Basta buscar pelo tick da ação ou pelo nome da empresa. Na página do ativo, basta clicar em “Historical Data”, conforme imagem abaixo:

Petróleo Brasileiro S.A. - Petrobras (PBR)

NYSE - NYSE Delayed Price. Currency in USD

☆ Add to watchlist

8.74 +0.16 (+1.86%) **8.70** -0.05 (-0.57%)

At close: 4:02PM EDT

After hours: 4:11PM EDT

Summary

Conversations

Statistics

Profile

Financials

Options

Holders

Historical Data

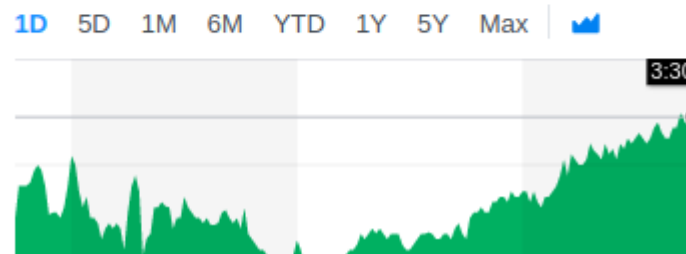
Analysts

Previous Close **8.58** Market Cap **56.942B**

Open **8.66** Beta **2.75**

Bid **8.65 x 3200** PE Ratio (TTM) **-20.61**

Ask **8.74 x 200** EPS (TTM) **-0.42**



Depois, basta selecionar o período desejado e clicar em “Download Data”. O arquivo baixado estará no formato csv.

Agora é só colocá-lo no seu working directory, que pode ser descoberto utilizando o comando `getwd()` .

```
pbr <- read.csv("PBR.csv")  
pbr[,1] <- as.Date(pbr[,1])  
pbr <- xts(pbr)  
pbr <- pbr[, -1]
```

Visualização de Preços

Agora vamos dar uma breve olhada nos dados, apenas para conhecê-los melhor. Rode as linhas a seguir e analise cada output.

In [145]: `head(pbr)`

	AMZN.Open	AMZN.High	AMZN.Low	AMZN.Close	AMZN.Volume	AMZN.Adjusted
2013-01-02	256.08	258.10	253.26	257.31	3271000	257.31
2013-01-03	257.27	260.88	256.37	258.48	2750900	258.48
2013-01-04	257.58	259.80	256.65	259.15	1874200	259.15
2013-01-07	262.97	269.73	262.67	268.46	4910000	268.46
2013-01-08	267.07	268.98	263.57	266.38	3010700	266.38
2013-01-09	268.17	269.50	265.40	266.35	2265600	266.35

In [146]: `tail(pbr)`

	AMZN.Open	AMZN.High	AMZN.Low	AMZN.Close	AMZN.Volume	AMZN.Adjusted
2020-05-21	2500.00	2525.45	2442.54	2446.74	5114400	2446.74
2020-05-22	2455.01	2469.85	2430.13	2436.88	2860900	2436.88
2020-05-26	2458.00	2462.00	2414.06	2421.86	3568200	2421.86
2020-05-27	2404.99	2413.58	2330.00	2410.39	5056900	2410.39
2020-05-28	2384.33	2436.97	2378.23	2401.10	3190200	2401.10
2020-05-29	2415.94	2442.37	2398.20	2442.37	3529300	2442.37

In [147]: `summary(pbr)`

Index	AMZN.Open	AMZN.High	AMZN.Low
Min. :2013-01-02	Min. : 248.9	Min. : 252.9	Min. : 245.8
1st Qu.:2014-11-06	1st Qu.: 366.4	1st Qu.: 371.4	1st Qu.: 363.3
Median :2016-09-14	Median : 764.4	Median : 768.7	Median : 758.9
Mean :2016-09-14	Mean : 952.5	Mean : 961.9	Mean : 942.0
3rd Qu.:2018-07-23	3rd Qu.:1623.5	3rd Qu.:1639.4	3rd Qu.:1600.9
Max. :2020-05-29	Max. :2500.0	Max. :2525.4	Max. :2467.3

AMZN.Close	AMZN.Volume	AMZN.Adjusted
Min. : 248.2	Min. : 881300	Min. : 248.2
1st Qu.: 366.4	1st Qu.: 2654800	1st Qu.: 366.4
Median : 764.0	Median : 3452800	Median : 764.0
Mean : 952.5	Mean : 4073160	Mean : 952.5
3rd Qu.:1626.0	3rd Qu.: 4730000	3rd Qu.:1626.0
Max. :2497.9	Max. :23856100	Max. :2497.9

In [148]: `str(pbr)`

```
An 'xts' object on 2013-01-02/2020-05-29 containing:  
  Data: num [1:1865, 1:6] 256 257 258 263 267 ...  
- attr(*, "dimnames")=List of 2  
  ..$ : NULL  
  ..$ : chr [1:6] "AMZN.Open" "AMZN.High" "AMZN.Low" "AMZN.Close" ...  
Indexed by objects of class: [Date] TZ: UTC  
xts Attributes:  
List of 2  
 $ src      : chr "yahoo"  
 $ updated: POSIXct[1:1], format: "2020-11-23 18:16:37"
```

Pelos comandos `head()` e `tail()` podemos ver as primeiras e últimas 6 linhas da base.

São 6 colunas com: preço de abertura, preços máximo e mínimo do dia, preço de fechamento, volume de transações e preço ajustado.

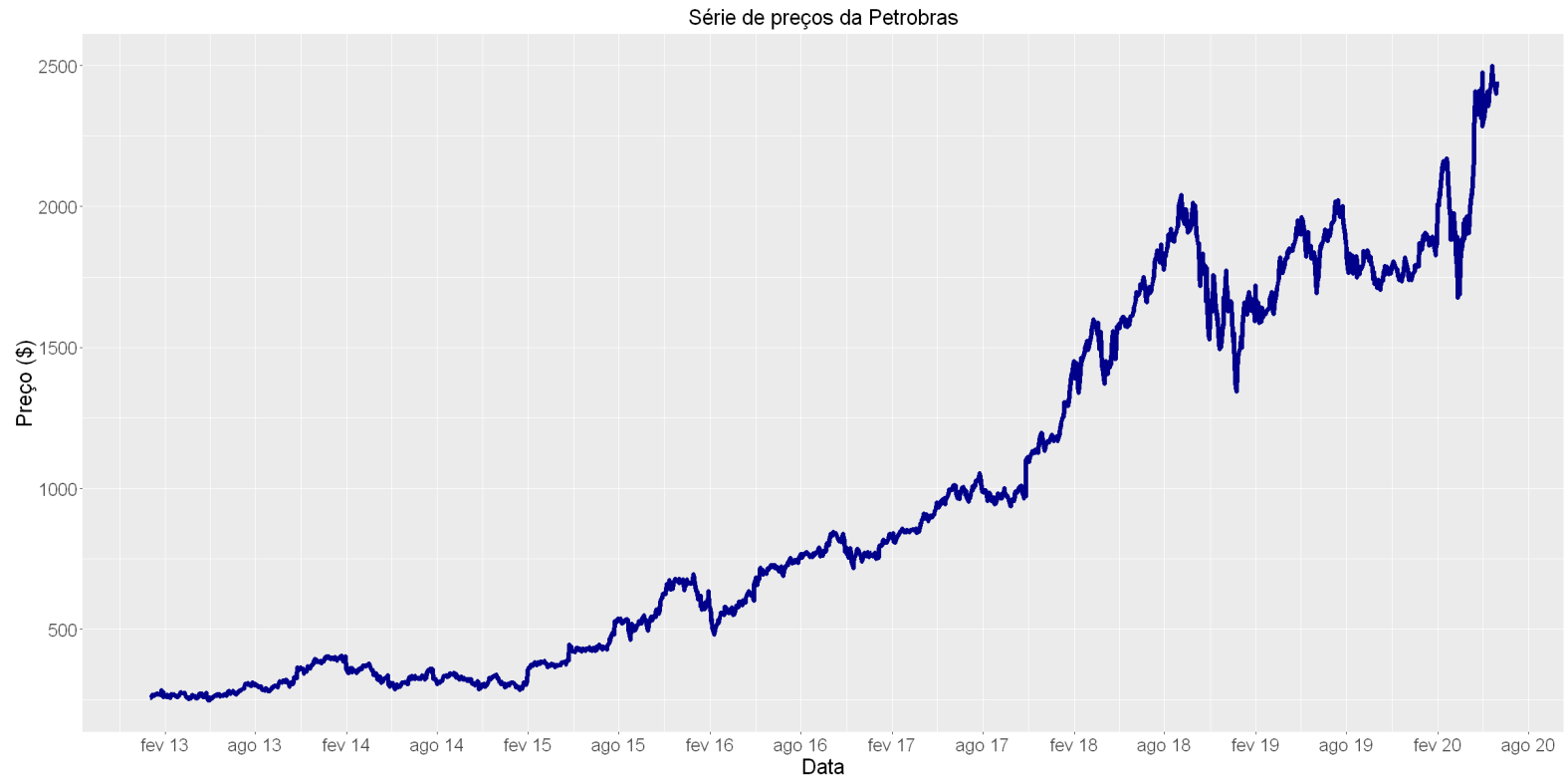
Pelo comando `summary` verificamos as estatísticas descritivas de cada série de preços e volume.

Já o comando `str` fornece a estrutura do objeto. Neste caso, é um objeto [xts](https://cran.r-project.org/web/packages/xts/vignettes/xts.pdf) (<https://cran.r-project.org/web/packages/xts/vignettes/xts.pdf>), uma série temporal.

Vamos agora plotar os preços diários, utilizando a coluna de Preço Ajustado, visto que ela incorpora eventos como splits e distribuição de dividendos, que podem afetar a série.

```
In [164]: library(repr)
options(repr.plot.width = 20, repr.plot.height = 10)
ggplot(pbr, aes(x = index(pbr), y = pbr[,6])) + geom_line(color = "darkblue", size=2) +
ggtitle("Série de preços da Petrobras") +
xlab("Data") + ylab("Preço ($)") +
theme(plot.title = element_text(hjust = 0.5, size=20)) +
theme(text = element_text(size = 20))+
scale_x_date(date_labels = "%b %y", date_breaks = "6 months")
```

Don't know how to automatically pick scale for object of type xts/zoo. Defaulting to continuous.



Criamos esse gráfico usando o comando `ggplot`.

Antes de tudo carregamos o pacote `repr` e com o comando `options(repr.plot.width = 20, repr.plot.height = 10)` ajustamos o tamanho do gráfico para 20 x 10

Primeiro utilizamos o objeto `pbr` como a série a ser plotada.

Depois indicamos quais elementos serão os eixos: `index(pbr)`, a data no eixo x, e a coluna de preço ajustado, `pbr[,6]`, no eixo y.

Em seguida, adicionamos o elemento a ser plotado, no caso, uma linha azul: `geom_line(color = "darkblue")`.

Depois, acrescentamos o título e nomes dos eixos, com os comandos `ggtitle("Série de preços da Petrobras")`, `xlab("Data")`, `ylab("Preço ($)")`.

Por padrão, o título do gráfico fica alinhado à esquerda. Para centralizá-lo, utilizamos `theme(plot.title = element_text(hjust = 0.5))`.

Usamos `theme(text = element_text(size = 20))` para ajustar o tamanho da fonte para 20

Por último, para deixar o eixo temporal com mais informações, colocamos o tick de data a cada seis meses e na forma de mmm aa utilizando `scale_x_date(date_labels = "%b %y", date_breaks = "6 months")`.

Em Análise Técnica (https://pt.wikipedia.org/wiki/An%C3%A1lise_t%C3%A9cnica) de ações, uma técnica muito utilizada é a plotagem de médias móveis nos gráficos de preços. Uma média móvel simples é a média aritmética dos últimos q dias de uma série x_t no período de tempo t . Assim, a média móvel MM_{qt} é dada por:

$$MM_t^q = \frac{1}{q} \sum_{i=0}^{q-1} x_{t-i}$$

Esse indicador é interessante pois ajuda a identificar tendências e suaviza ruídos dos preços.

Isto é, quanto maior a janela de dias para o cálculo da média, menor é a responsividade da MM à variação no preço.

Quanto menor a janela, mais rápido a média móvel se ajusta às mudanças.

Vamos então calcular agora duas médias móveis para a série de preços da Petrobras, uma com janela de 10 dias e outra com janela de 30 dias:

```
In [150]: pbr_mm <- subset(pbr, index(pbr) >= "2016-01-01")

pbr_mm10 <- rollmean(pbr_mm[,6], 10, fill = list(NA, NULL, NA), align = "right")
pbr_mm30 <- rollmean(pbr_mm[,6], 30, fill = list(NA, NULL, NA), align = "right")

pbr_mm$mm10 <- coredata(pbr_mm10)
pbr_mm$mm30 <- coredata(pbr_mm30)
```

Primeiro dividimos a base para dados a partir de 2016 utilizando a função `subset()`.

Em seguida, usamos a função `rollmean()`, que toma como argumento a série x_t , no caso o preço ajustado; a janela de períodos (q); um argumento opcional de preenchimento, que é utilizado para completar os dias em que ainda não é possível calcular a média móvel, ou seja, quando não se passaram dias suficiente para completar a janela desejada; e, por último, o argumento `align`, que indica se a média móvel deve ser calculada usando os períodos à esquerda, ao centro ou à direita do dia t da série.

Por último, acrescentamos as médias móveis a duas novas colunas na base inicial.

Em nosso caso, calculamos utilizando 10 e 30 dias de janela, preenchendo os valores com NA e utilizando os períodos à esquerda do dia em questão.

Em seguida, podemos plotar ambas as séries no mesmo gráfico de preços para identificar tendências. Uma teoria existente na Análise Técnica é a de que quando duas médias móveis de janelas de curto e longo prazo se cruzam, há uma indicação de compra ou de venda.

Caso a MM de menor período cruze de baixo para cima a MM de maior período, há uma indicação de compra. Se o cruzamento for o contrário, é um sinal de **venda**.

Plotando a série de preços e as médias móveis para todos os dias a partir de 2016, temos:

```
In [169]: ggplot(pbr_mm, aes(x = index(pbr_mm))) + geom_line(aes(y = pbr_mm[,6], color = "PBR"), size=1) +
  ggtitle("Série de preços da Petrobras") +
  geom_line(aes(y = pbr_mm$mm10, color = "MM10"), size=1) +
  geom_line(aes(y = pbr_mm$mm30, color = "MM30"), size=1) +
  xlab("Data") + ylab("Preço ($)") +
  theme(plot.title = element_text(hjust = 0.5), panel.border = element_blank()) +
  theme(text = element_text(size = 24)) +
  scale_x_date(date_labels = "%b %y", date_breaks = "3 months") +
  scale_colour_manual("Séries", values=c("PBR"="gray40", "MM10"="firebrick4", "MM30"="dark cyan"))
```

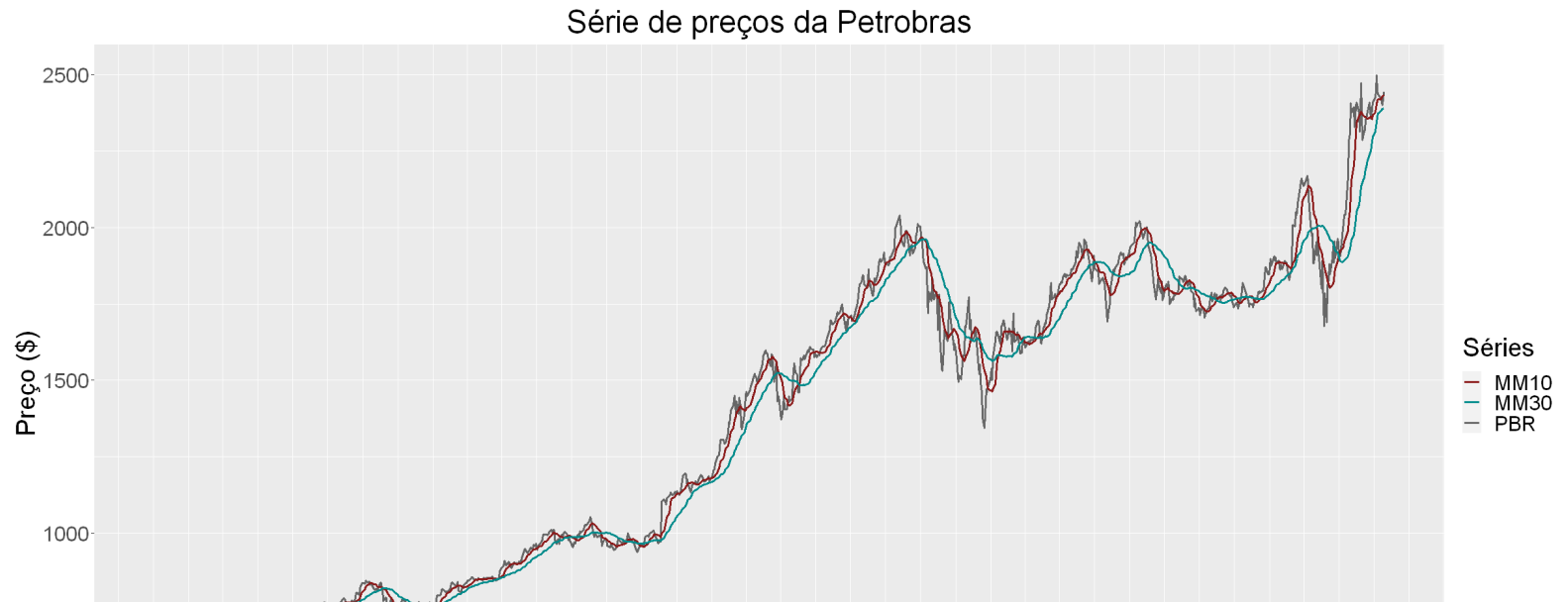
Don't know how to automatically pick scale for object of type xts/zoo. Defaulting to continuous.

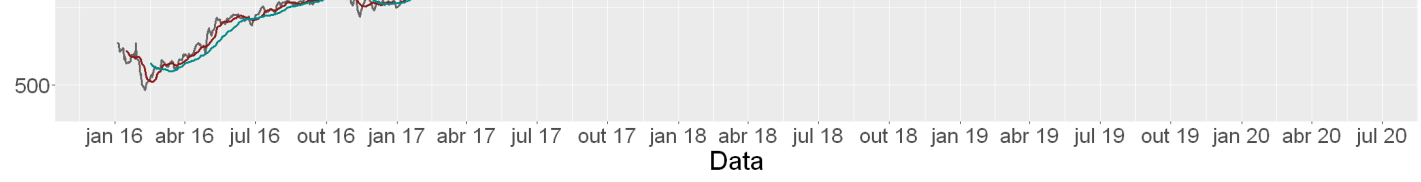
Warning message:

"Removed 9 row(s) containing missing values (geom_path)."

Warning message:

"Removed 29 row(s) containing missing values (geom_path)."





Para criar o gráfico, plotamos a linha de preços e as linhas das médias móveis.

Nesta ocasião, utilizamos de forma diferente a plotagem de cada linha, criando uma espécie de apelido para as cores de cada uma.

Em seguida, adicionamos a linha `scale_colour_manual`, indicando a cor de cada apelido para que as cores constassem na legenda de cada série.

Retornos!

Vimos como o preço da ação tem variado ao longo do tempo. Vamos agora verificar como se comportou o retorno da ação no mesmo período.

Para isso, precisamos primeiro criar um novo objeto com os retornos calculados, utilizando também a coluna de preço ajustado:

```
In [152]: pbr_ret <- diff(log(pbr[,6]))  
pbr_ret <- pbr_ret[-1,]
```

O que fizemos aqui foi utilizar as propriedades do logaritmo para calcular o retorno logarítmico da ação. Ou seja, fizemos:

$$r_t = \ln(1 + R_t) = \ln\left(\frac{P_t}{P_{t-1}}\right) = \ln(P_t) - \ln(P_{t-1}) \approx R_t$$

O comando `diff` calcula a diferença de todos os valores de algum vetor ou elemento. Com isso, apenas aplicamos a diferença aos logaritmos naturais dos preços da ação.

Além desse modo, é possível calcular os retornos de forma diferente. O pacote quantmod possui algumas funções interessantes para isso.

Primeiramente, é bastante fácil selecionar apenas uma coluna de preços de cada ação, por exemplo:

```
In [153]: head(Op(pbr))
```

	AMZN.Open
2013-01-02	256.08
2013-01-03	257.27
2013-01-04	257.58
2013-01-07	262.97
2013-01-08	267.07
2013-01-09	268.17

```
In [154]: head(Cl(pbr))
```

	AMZN.Close
2013-01-02	257.31
2013-01-03	258.48
2013-01-04	259.15
2013-01-07	268.46
2013-01-08	266.38
2013-01-09	266.35

```
In [155]: head(Ad(pbr))
```

	AMZN.Adjusted
2013-01-02	257.31
2013-01-03	258.48
2013-01-04	259.15
2013-01-07	268.46
2013-01-08	266.38
2013-01-09	266.35

Todos esses comandos tornam a análise e os cálculos muito mais rápidos e fáceis de serem feitos.

Vamos agora verificar algumas estatísticas básicas dos retornos da Petrobras:


```
In [156]: summary(pbr_ret)
```

	Index	AMZN.Adjusted
Min.	:2013-01-03	Min. : -0.116503
1st Qu.	:2014-11-06	1st Qu.: -0.007426
Median	:2016-09-14	Median : 0.001196
Mean	:2016-09-14	Mean : 0.001207
3rd Qu.	:2018-07-23	3rd Qu.: 0.010734
Max.	:2020-05-29	Max. : 0.132178

```
In [157]: sd(pbr_ret)
```

```
0.0190142323742705
```

```
In [158]: mean(pbr_ret)
```

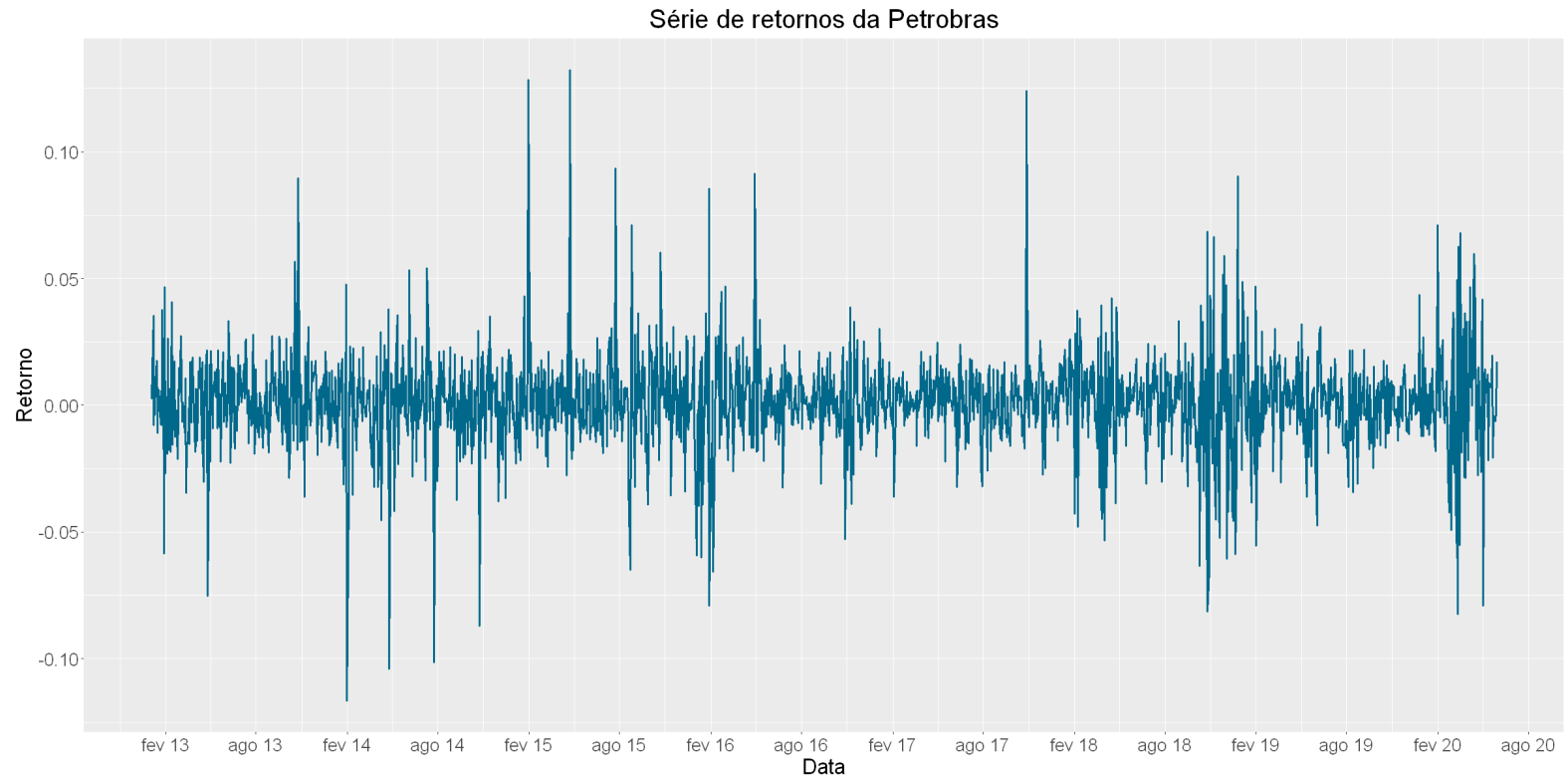
```
0.00120731900769929
```

Ou seja, em média a ação não tem tido um bom desempenho.

Podemos agora plotar um gráfico dos retornos e ver como eles desempenharam ao longo do tempo:

```
In [170]: ggplot(pbr_ret, aes(x = index(pbr_ret), y = pbr_ret)) + geom_line(color = "deepskyblue4",
, size=1) +
ggtitle("Série de retornos da Petrobras") + xlab("Data") + ylab("Retorno") +
theme(plot.title = element_text(hjust = 0.5)) +
theme(text = element_text(size = 20))+
scale_x_date(date_labels = "%b %y", date_breaks = "6 months")
```

Don't know how to automatically pick scale for object of type xts/zoo. Defaulting to continuous.



Para plotar este último gráfico, utilizamos os mesmos parâmetros do gráfico de preços, alterando apenas a cor da linha.

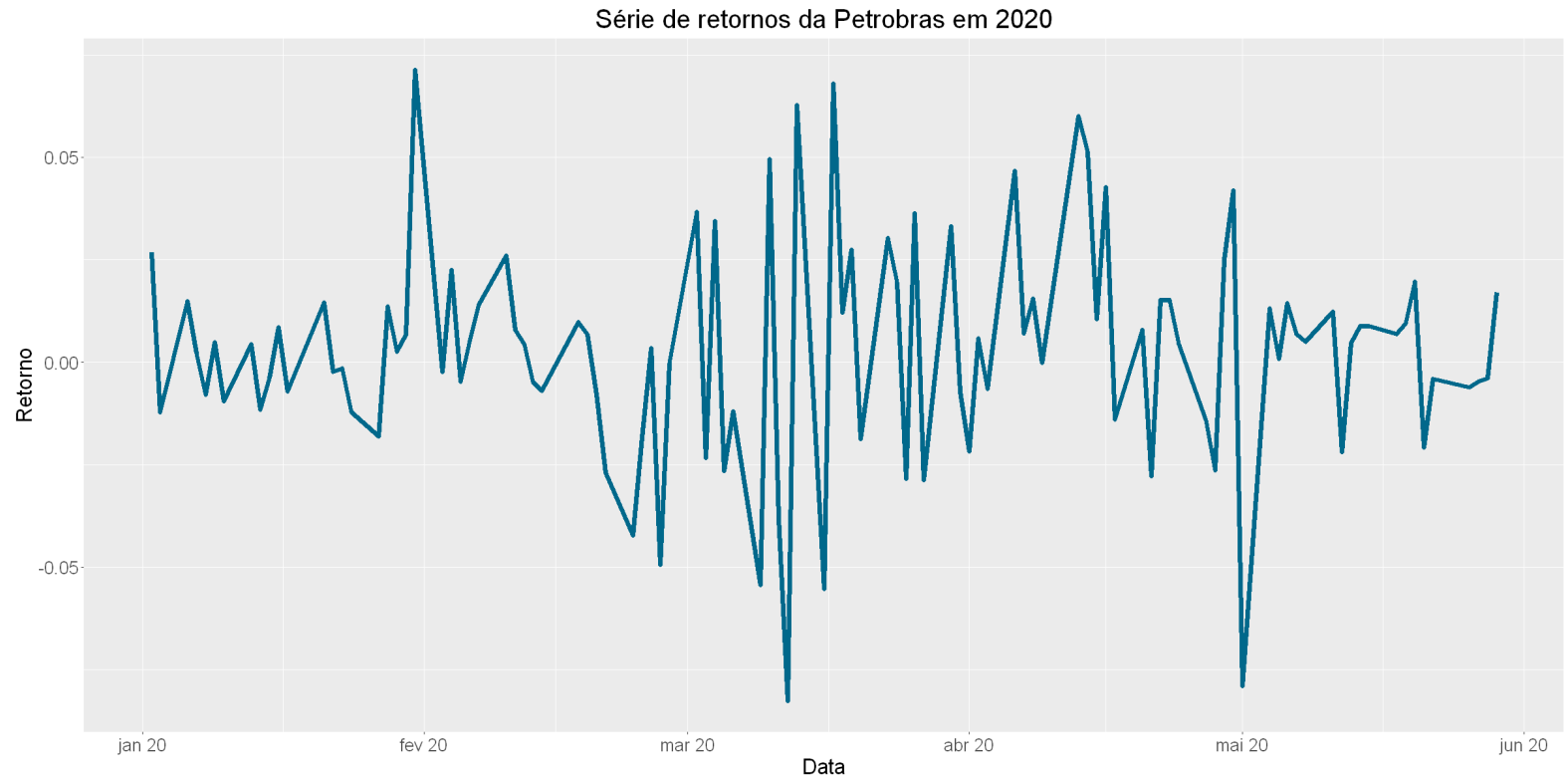
Dando uma breve analisada no gráfico, é possível ver que o menor retorno da série ocorreu este ano entre fevereiro e agosto.

Vamos fazer agora uma breve checagem dos retornos da ação em 2020:

```
In [171]: pbr_ret20 <- subset(pbr_ret, index(pbr_ret) > "2020-01-01")

ggplot(pbr_ret20, aes(x = index(pbr_ret20), y = pbr_ret20)) + geom_line(color = "deepsky
blue4", size=2) +
ggtitle("Série de retornos da Petrobras em 2020") + xlab("Data") + ylab("Retorno") +
theme(plot.title = element_text(hjust = 0.5)) +
theme(text = element_text(size = 20))+
scale_x_date(date_labels = "%b %y", date_breaks = "1 months")
```

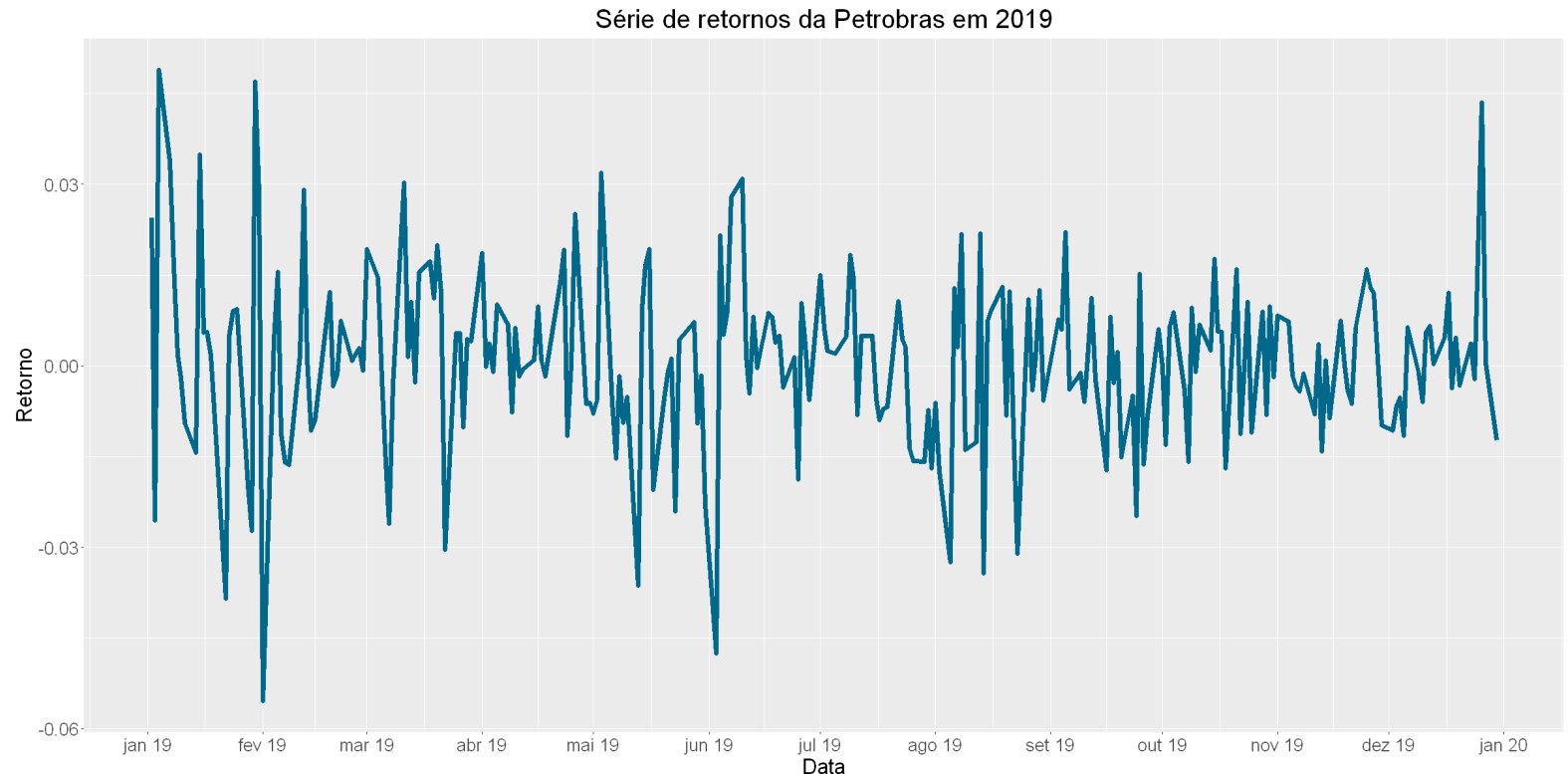
Don't know how to automatically pick scale for object of type xts/zoo. Defaulting to continuous.



```
In [172]: pbr_ret19 <- subset(pbr_ret, (index(pbr_ret) > "2019-01-01") & (index(pbr_ret) < "2019-12-31"))

ggplot(pbr_ret19, aes(x = index(pbr_ret19), y = pbr_ret19)) +
  geom_line(color = "deepskyblue4", size=2) +
  ggtitle("Série de retornos da Petrobras em 2019") + xlab("Data") + ylab("Retorno") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text = element_text(size = 20)) +
  scale_x_date(date_labels = "%b %y", date_breaks = "1 months")
```

Don't know how to automatically pick scale for object of type xts/zoo. Defaulting to continuous.



In [162]: `summary(pbr_ret20)`

`mean(pbr_ret20)`

`sd(pbr_ret20)`

	Index	AMZN.Adjusted
Min.	:2020-01-02	Min. : -0.082535
1st Qu.	:2020-02-08	1st Qu.: -0.010540
Median	:2020-03-17	Median : 0.004478
Mean	:2020-03-16	Mean : 0.002708
3rd Qu.	:2020-04-22	3rd Qu.: 0.014650
Max.	:2020-05-29	Max. : 0.071196

0.00270826753672114

0.0270156005554484

In [163]: `summary(pbr_ret19)`

`mean(pbr_ret19)`

`sd(pbr_ret19)`

Index	AMZN.Adjusted
Min. :2019-01-02	Min. :-0.0553212
1st Qu.:2019-04-02	1st Qu.: -0.0066713
Median :2019-07-02	Median : 0.0012141
Mean :2019-07-01	Mean : 0.0008236
3rd Qu.:2019-09-30	3rd Qu.: 0.0087527
Max. :2019-12-30	Max. : 0.0488511

0.00082360793252136

0.0144595060589382

In []: