

Converter um objeto xts zoo num tibble

Ariel Levy

7/27/2018

Antes de iniciar recomendo a leitura até o fim.

Carregar os pacotes

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

Obter os dados do exemplo:

```
B3<-getSymbols("^BVSP", from = "1999-01-01", to = "2018-03-30", src = "yahoo")
```

Neste exemplo estamos utilizando o índice Bovespa da B3 disponível no Yahoo com o ticker “^BVSP”, obtido com a função `getSymbols()` do pacote `quantmod`. Observe que `B3` é uma variável caracter e `BVSP` foi criada com classe `xts zoo`.

```
head(BVSP)
```

```
##           BVSP.Open BVSP.High BVSP.Low BVSP.Close BVSP.Volume
## 1999-01-04      6787      7052      6787      6942           0
## 1999-01-05      6942      7147      6843      7111           0
## 1999-01-06      7111      7334      7110      7331           0
## 1999-01-07      7161      7161      6841      6954           0
## 1999-01-08      6955      7100      6706      6781           0
## 1999-01-11      6784      6784      6268      6403           0
##           BVSP.Adjusted
## 1999-01-04           6942
## 1999-01-05           7111
## 1999-01-06           7331
## 1999-01-07           6954
## 1999-01-08           6781
## 1999-01-11           6403
```

```
class(BVSP)
```

```
## [1] "xts" "zoo"
```

Converter para tibble

Parece trivial, não? Basta utilizar a função `as_tibble()`

```
BVSP_tb<-as_tibble(BVSP)
head(BVSP_tb)
```

```
## # A tibble: 6 x 6
##   BVSP.Open BVSP.High BVSP.Low BVSP.Close BVSP.Volume BVSP.Adjusted
```

```
##      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1      6787      7052      6787      6942          0      6942
## 2      6942      7147      6843      7111          0      7111
## 3      7111      7334      7110      7331          0      7331
## 4      7161      7161      6841      6954          0      6954
## 5      6955      7100      6706      6781          0      6781
## 6      6784      6784      6268      6403          0      6403
```

Entretanto não aparecem as Index. Observe que no xts elas eram os nomes das linhas, e não uma coluna. Será que sumiram?

```
head(rownames(BVSP_tb))
```

```
## [1] "1999-01-04" "1999-01-05" "1999-01-06" "1999-01-07" "1999-01-08"
## [6] "1999-01-11"
```

```
class(rownames(BVSP_tb))
```

```
## [1] "character"
```

Elas ainda existem e num formato character. Será necessário transformá-las para o formato date, utilizando a função `as.Date()`. No código abaixo utilizei também `%<>%`, um tipo diferente de pipe cujo uso permite assinalar de volta ao objeto o produto da sequência do pipe.

```
Index<-rownames(BVSP_tb)
```

```
Index %<>% as.Date()
```

```
head(Index,4)
```

```
## [1] "1999-01-04" "1999-01-05" "1999-01-06" "1999-01-07"
```

```
class(Index)
```

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

Pronto agora bastará criar uma nova coluna com `dplyr::mutate` e arrumar com `dplyr::select`. Observe que os parenteses externos equivalem a solicitar que o resultado seja exposto.

```
(BVSP_tb %<>% mutate(Index = Index) %>%
  select(Index, BVSP.Open:BVSP.Adjusted))
```

```
## # A tibble: 4,861 x 7
##   Index      BVSP.Open BVSP.High BVSP.Low BVSP.Close BVSP.Volume
##   <date>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 1999-01-04      6787      7052      6787      6942          0
## 2 1999-01-05      6942      7147      6843      7111          0
## 3 1999-01-06      7111      7334      7110      7331          0
## 4 1999-01-07      7161      7161      6841      6954          0
## 5 1999-01-08      6955      7100      6706      6781          0
## 6 1999-01-11      6784      6784      6268      6403          0
## 7 1999-01-12      6408      6408      5792      5916          0
## 8 1999-01-13      5657      5742      5278      5617          0
## 9 1999-01-14      5702      5857      5050      5057          0
## 10 1999-01-15      5039      6782      4797      6747          0
## # ... with 4,851 more rows, and 1 more variable: BVSP.Adjusted <dbl>
```

Você aprendeu bastante:

Sobre como carregar dados financeiros do Yahoo (pesquise mais sobre o pacote `quantmod`).

Sobre datas, nome de linhas.

Sobre pipples e o pacote magrittr.

Sobre o uso do mutate e select do pacote dplyr.

Sobre o uso de parenteses externos para expor um resultado.

Usando o ggplot2

Contudo, há uma forma mais simples de obter o mesmo resultado, de posse do objeto BVSP de classe xts com a função fortify.zoo do pacote ggplot2.

```
(BVSP_tb2<-fortify.zoo(BVSP) %>% as_tibble())
```

```
## # A tibble: 4,861 x 7
##   Index      BVSP.Open BVSP.High BVSP.Low BVSP.Close BVSP.Volume
##   <date>      <dbl>    <dbl>    <dbl>    <dbl>      <dbl>
## 1 1999-01-04    6787      7052      6787      6942         0
## 2 1999-01-05    6942      7147      6843      7111         0
## 3 1999-01-06    7111      7334      7110      7331         0
## 4 1999-01-07    7161      7161      6841      6954         0
## 5 1999-01-08    6955      7100      6706      6781         0
## 6 1999-01-11    6784      6784      6268      6403         0
## 7 1999-01-12    6408      6408      5792      5916         0
## 8 1999-01-13    5657      5742      5278      5617         0
## 9 1999-01-14    5702      5857      5050      5057         0
## 10 1999-01-15    5039      6782      4797      6747         0
## # ... with 4,851 more rows, and 1 more variable: BVSP.Adjusted <dbl>
```

Você pensava que o ggplot2 só servia para gráficos?

Aprenda sempre as funções dos pacotes elas podem ajudar muito.

E lembre sempre a prática leva a perfeição então pratique o que aprendeu.

Referências

WICKHAM, Hardley; GROLEMUND Garrett - R for Data Science Orelly 2016

GONÇALVES, Guilherme A., How-transforming-xts-object-on-data-frame-in-r-quantmod-package disponível em 28/07/2018:

<https://stackoverflow.com/questions/49469964/how-transforming-xts-object-on-data-frame-in-r-quantmod-package/>