

Um tutorial sobre o pacote `xts` do R

William Suzuki - USP

Este documento é um tutorial sobre o pacote `xts` do R apresentado no grupo de estudos da pós graduação em economia da FEA RP USP.

O `xts` apresenta uma série de ferramentas úteis para manipulação de séries de tempo. Quem trabalha com macroeconometria ou finanças pode se beneficiar muito em aprender alguns dos seus instrumentos.

O R tem vários tipos de classes de objetos: vetores numéricos, characters, matrizes, data frames; entre outros. Aqui vamos usar objetos da classe `xts` e `zoo`, este que é também um outro pacote importante para manipulação de séries de tempo.

```
#carregar pacotes
library(zoo)
library(xts)
library(readxl)
```

Objetos `xts` são basicamente matrizes com um indicador de tempo em cada linha.

Em um primeiro exemplo vamos construir uma série com ruído branco de uma Gaussiana padrão:

```
#criar vetor com valores aleatórios
nobs <- 50
set.seed(9876)
data <- rnorm(nobs)

#Classe 'Date', criar vetores de 'Date'
dates <- seq(as.Date("2010-01-01"), length = nobs, by = "day")
#no by= experimente com "week" "month" "quarter"

# Crie o objeto xts
ruído <- xts(x = data, order.by = dates)

#veja a classe do objeto
class(ruído)

#plote o gráfico da série temporal
plot(ruído)
```

Para saber mais como criar vetores de `Date` veja:

<https://stat.ethz.ch/R-manual/R-devel/RHOME/library/base/html/seq.Date.html>

Objetos do tipo `xts` podem ser pensados como uma junção de uma matriz e um índice temporal. Chamamos a matrix de “core” e o índice de “index”.

Vamos aprender como extrair esses dois objetos:

```
# extrair o core do ruído
ruído_core <- coredata(ruído)

# Qual a classe do ruído_core
class(ruído_core)

# extrair o index do ruído
ruído_index <- index(ruído)
```

```
# Qual a classe do ruido_index  
class(ruido_index)
```

POSIXct cria objetos em formato de data com horas, minutos e segundos.

```
# criar dados de índice de tempo  
dates <- as.Date("2016-01-01") + 0:10  
dates  
  
#aplicar o POSIXct  
as.POSIXct(dates)
```

Como um exemplo, vamos importar e manipular a série de preços do CSN.

```
#onde é o working directory?  
getwd()  
  
#coloque as bases de dados no working directory  
#importe os dados xlsx  
sid_nacional <- read_excel("economica 190515 sid nacional.xlsx")  
  
#classe do objeto sid_nacional  
class(sid_nacional)  
  
#mudar os nomes das variaveis  
names(sid_nacional) <- c('date','price')  
  
#ver a dimensão da base  
dim(sid_nacional)  
  
#use head() para ver as primeiras 5 observações de um objeto  
head(sid_nacional$date)  
head(sid_nacional$price)  
  
#verificar a classe dos objetos  
class(sid_nacional$date)  
class(sid_nacional$price)  
  
#mudar a classe dos objetos  
price <- as.numeric(sid_nacional$price)  
date <- as.character(sid_nacional$date)  
  
#verifique as novas classes  
head(date); class(date)  
head(price); class(price)  
  
#transformar character em Date  
date <- as.Date(date,"%Y-%m-%d")  
  
#construir o xts  
sid_nacional <- xts(price,date)  
  
#plotar o time series  
plot(sid_nacional)
```

Como extrair um período da série de tempo?

```
#extrair de 1 de janeiro de 2018 para frente
sid_nacional_2 <- sid_nacional['20180101/']
plot(sid_nacional_2)

#extrair apenas o ano de 2017
sid_nacional_3 <- sid_nacional['20170101/20171231']
plot(sid_nacional_3)
```

Agora, vamos importar dados intraday da ExxonMobil no NYSE. Os dados foram padronizados para mostrar em minutos.

```
#verificar o working directory
getwd()

#certifique de colocar o arquivo US1.XOM_180101_190515.csv no workig directory
#assim podemos importar os dados intraday

#vamos criar uma função que importa os dados para nós:
intra_exxon <- function(){
  irreg <- read.csv("US1.XOM_180101_190515.csv") #ler o csv
  irreg <- irreg[,-c(1,2,6:9)] #dropar colunas não importantes
  names(irreg) <- c('date','time','open') #mudar nome das variáveis
  irreg$time <- as.character(irreg$time) #mudar time para character
  irreg$date <- as.character(irreg$date) #mudar date em character
  year <- substr(irreg$date,1,4) #pegar ano
  month <- substr(irreg$date,5,6) #pegar mês
  day <- substr(irreg$date,7,8) #pegar dia
  #i <-1
  for (i in 1:dim(irreg)[1]){ #change time because 9h's have smaller character sizes
    if (substr(irreg$time[i],1,1)=='9') {
      irreg$time[i] <- paste('0', irreg$time[i], sep='') #put a zero in front of it
    }
  }
  hours <- substr(irreg$time,1,2) #pegar horas
  seconds <- substr(irreg$time,5,6) #pegar segundos
  minutes <- substr(irreg$time,3,4) #pegar minutos
  date_time <- paste(year,'-',month,'-',day,' ',hours,':',minutes,':',seconds,sep='') #juntar todos em
  irreg$date_time <- as.POSIXct(date_time, format='%Y-%m-%d %H:%M:%OS') #transformar o objeto em POSIXct
  irreg <- irreg[,c(4,3)] #Exxon Mobil shares
  irreg <- xts(x = irreg$open ,order.by = irreg$date_time) ##criar o xts
  names(irreg) <- c('price') #change name of irreg
  irreg
}
```

```
exxon <- intra_exxon() #pegar o output da função que criamos e colocar em 'exxon'

#vamos ver as 5 primeiras observações da Exxon
head(exxon)

#plotar Exxon
plot(exxon)
```

Como extrair dados de todos os dias, mas apenas de certos períodos durante o dia.

```
#extrair dados da abertura 9:30 até às 12h.  
exxon2 <- exxon["T09:30/T12:00"]  
plot(exxon2)
```

Substituir dados dentro de um xts

```
exxon3 <- exxon
```

```
#você pode especificar certas data e atribuir vaores para elas  
#aqui vamos chamar de dates como todos os períodos depois de 2019  
dates <- '2019/'
```

```
#substituir todos os dados de 2019 em diante como NA  
exxon3[dates] <- NA  
plot(exxon)
```

```
plot(exxon3)
```

```
# substituir todos os valores de agosto de 2018 como 0  
exxon3["20180801/20180830"] <- 0  
plot(exxon3)
```

```
# substituir todos os valores das 13h até 16h como 0  
exxon3["T13:00/T16:00"] <- 50  
plot(exxon3)
```

Tutorial baseado em:

<https://rpubs.com/mohammadshadan/288218>