Curso de Introdução ao R para Data Science

Vítor Wilher

1 Introdução

Eu comecei a utilizar o **R** há alguns anos, influenciado por amigos. Minha introdução ao mundo dos programas estatísticos foi através do Eviews, ainda nos tempos da graduação em economia, como provavelmente muitos dos matriculados nesse curso. Ainda que seja possível *programar* no Eviews e em outros pacotes estatísticos fechados (que precisam de licença), as vantagens do **R** são inúmeras, como comentarei mais à frente. Por ora, talvez seja necessário tecer algumas palavras sobre por que afinal é preciso aprender uma linguagem de programação.

Eu poderia falar que o mundo está mudando, que cada vez mais empregos e empresas têm exigido conhecimentos de programação. E isso de fato é verdade, o que por si só gera uma necessidade de saber programação. Mas, aqui entre nós, acho que é meio chato aprender algo por necessidade, não é mesmo?

Minha motivação para aprender a programar foi de fato outra. Eu estava um pouco cansado de *apertar botões* e fazer tarefas repetitivas em pacotes estatísticos como o Eviews, então parecia natural aprender uma forma de *automatizar* as coisas. Essa, afinal, era uma baita motivação para mim e talvez também seja para você. Mas por que o **R**, você pode perguntar.

Essa é de fato uma boa pergunta. Por que não aprender a programar no próprio Eviews? E a resposta é bastante simples. Você já tentou encontrar alguma coisa de programação em Eviews na internet? E sobre **R**? Pois é. Entre as inúmeras vantagens do **R**, posso destacar:

- A existência de uma comunidade grande e bastante entusiasmada, que compartilha conhecimento todo o tempo;
- o **R** é gratuito, *open source*, de modo que você não precisa comprar licenças de software para instalá-lo;
- Tem inúmeras bibliotecas (pacotes) em estatística, *machine learning*, visualização, importação e tratamento de dados;
- Possui uma linguagem estabelecida para *data analysis*;
- Ferramentas poderosas para comunicação dos resultados da sua pesquisa, seja em forma de um website ou em pdf.

Ao aprender **R**, você conseguirá integrar as etapas de coleta, tratamento, análise e apresentação de dados em um único ambiente. Você vai esquecer ter de abrir o excel, algum pacote estatístico, depois o power point ou o word, depois um compilador de pdf para gerar seu relatório. Todas essas etapas serão feitas em um único ambiente. E essa talvez seja a grande motivação para você entrar de cabeça nesse mundo.[1]

Certamente não será simples, tenho de lhe dizer. Haverá muitos momentos em que você pensará em desistir. Um erro inesperado em algum *script* poderá lhe tirar horas do seu tempo. No início, mais especificamente,

[1] Para maiores detalhes sobre esse ponto, ver Grolemund and Wickham (2017).

qualquer pequeno problema pode parecer uma barreira intransponível. Nesses momentos, minha dica é *não se demorar muito nessas dificuldades*. Vá para outro problema ou mesmo descanse. Faça outras coisas. Depois volte mais tranquilo, procure nos canais de ajuda que colocaremos aqui e as coisas irão se acertar. Acredite: passado esse tempo inicial, com persistência, você certamente se beneficiará muito em ter aprendido uma linguagem de programação como o **R**.

Nosso objetivo na Análise Macro, com nossos Cursos Aplicados de R é lhe proporcionar uma experência inovadora. Todos os nossos cursos apresentam sólida teoria, mas sempre recheada de exercícios e exemplos práticos. Nosso objetivo é trazer para o Brasil algo que já é bastante corriqueiro no resto do mundo: *você deve aprender fazendo*. Isso vai tornar o aprendizado mais interessante, mais divertido até.

Isso dito, vamos começar então? Nessa primeira seção, você verá alguns procedimentos básicos, sobre programas e um *overview* do **R**. A partir da seção 02 começa o seu curso propriamente dito. Seja bem vindo a esse mundo e espero que não sai mais dele!

1.1 Instalando os programas

Antes de tudo, é preciso que você tenha os programas que utilizaremos ao longo das aulas instalados em seu computador. Serão três programas: **R**, **RStudio** e **MikTex**. Não se preocupe, posto que são todos programas gratuitos e com *download* seguro. Desse modo, para que não tenhamos problemas, siga a sequência abaixo:

1. Baixe o **R** em http://cran-r.c3sl.ufpr.br/;
2. Baixe o **RStudio** em https://www.rstudio.com/products/rstudio/download/;
3. Baixe o **MikTex** *se você for usuário de Windows* em http://miktex.org/download;
4. Baixe o **MacTex** *se você for usuário de Mac* em http://www.tug.org/mactex/.

Para que você não tenha problemas no futuro, instale a versão do **MikTex** correspondente à versão do seu Windows. Isto é, se o seu Windows for 64 bits, instale a versão 64 bits do **MikTex**, se for 32 bits, instale a versão 32 bits do **MikTex**.

Por fim, alguns pacotes que utilizaremos fazem uso do **JAVA**. Assim, é importante que você o tenha instalado em sua máquina, **na versão correspondente do seu sistema operacional**, assim como o **MikTex**. Para instalar o JAVA, vá em Oracle.

1.2 Trabalhando com o R a partir do RStudio

Ao longo desse curso, nós não trabalharemos diretamente no **R**. Ao invés disso, usaremos o **RStudio**, uma interface mais amigável, que nos permite emular todos os códigos do **R**, visualizar gráficos, ver o histórico de nossas operações, importar dados, criar scripts, etc. Com o **RStudio**, poderemos otimizar o nosso curso, de maneira que o aluno terá mais facilidade para interagir com a linguagem.

A figura acima resume as quatro principais partes de uma tela do **RStudio**. Na parte superior esquerda é onde ficará o nosso *editor de scripts*. Um *script* é uma sequência de comandos com um determinado objetivo. Por exemplo, você pode estar interessado em *construir um modelo univariado para fins de previsão do índice BOVESPA*. Para isso, terá de primeiro importar os dados do Ibovespa, bem como fazer uma análise descritiva inicial dos dados. Depois, com base nessa análise inicial, você terá de decidir entre alguns modelos univariados distintos. De posse da sua decisão, você enfim construirá um modelo de previsão para o índice BOVESPA. Essa sequência de *linhas de comando* pode ficar armazenada em um *script*, com extensão .R, podendo ser acessada posteriormente por você mesmo ou compartilhada com outros colegas de trabalho. Para abrir um novo *script*, vá em *File*, *New File* e clique em *R Script*.

Na parte inferior esquerda, está o *console do R*, onde você poderá executar comandos rápidos, que não queira registrar no seu script, bem como será mostrados os *outputs* dos comandos que você executou no seu script.

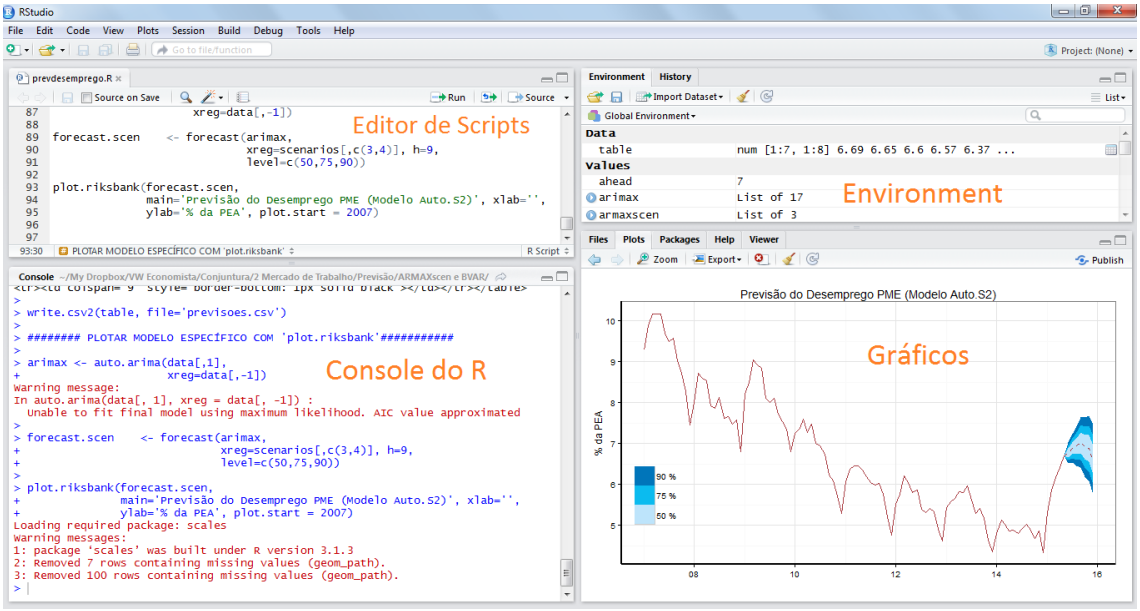

Figure 1: Ambiente do RStudio.

Já na parte superior direita, está o *Environment*, onde ficam mostrados os objetos que você cria ao longo da sua seção no RStudio. Por fim, na parte inferior direita, ficarão os gráficos que você solicitar, bem como pacotes que você instalou, alguma ajuda sobre as funções e os arquivos disponíveis no seu diretório de trabalho.

1.3 Definindo seu diretório de trabalho

Agora que você já instalou os programas e já conhece um pouco do ambiente do **RStudio**, podemos começar a brincar um pouco. Para isso, antes de mais nada é preciso definir o seu *diretório de trabalho* ou a pasta onde ficará salvo o seu *script*. Uma vez definido, você poderá importar arquivos, colocar figuras no seu documento LaTeX, etc. Logo, dois comandos são importantes para isso. O primeiro é o **getwd**, para você ver o seu atual *working directory*. O segundo é o **setwd**, para você *setar* o seu diretório de trabalho[2]

```
getwd()
```

[1] "C:/Users/Vítor Wilher/Dropbox/VW Economista/Análise Macro/01 - Cursos/R for Data Science - Tidyverse/Aulas/Aula01"

```
setwd('C:/Users/Vítor Wilher/Dropbox/VW Economista')
```

Uma vez *setado* o seu diretório de trabalho, você poderá importar dados contidos naquela pasta. Assim, é um ponto importante realizar isso antes de qualquer coisa. Caso já tenha um *script* de R, por suposto, uma vez abrindo-o, o RStudio setará automaticamente o diretório onde o mesmo esteja.

1.4 Uma linguagem orientada a objetos

O **R** é uma linguagem orientada a objetos, de modo que o que você fará dentro do programa será, basicamente, manipulá-los. Seja lidando com objetos criados por terceiros, seja criando seus próprios objetos. As principais

[2]Como você verá ao longo do nosso curso, as funções dentro do **R** são todas em inglês e bastante intuitivas, como *get something* ou *set something*.

estruturas de dados dentro do **R** envolvem *vetores*, *matrizes*, *listas* e *data frames*. Abaixo colocamos um exemplo da estrutura mais simples do **R**: um vetor que exprime a sequência de 1 a 10.

```
vetor <- c(1:10)
vetor
```

[1] 1 2 3 4 5 6 7 8 9 10

Ao longo do nosso curso, aprenderemos na prática a lidar com essas diferentes estruturas de dados. Em particular, aprenderemos a fazer o *subsetting* dessas estruturas, de modo a capturar subelementos das mesmas.

1.5 Milhares de pacotes a sua disposição

O **R** é uma linguagem aberta, onde qualquer pessoa em qualquer parte do mundo pode dar a sua contribuição. Em geral, elas fazem isso através de *pacotes*, que são coleções de funções que fazem algum coisa dentro do **R**. Veremos muitos desses pacotes ao longo do nosso curso. A instalação de pacotes é feita primariamente pelo CRAN, através da seguinte função:

```
install.packages('fBasics')
```

O pacote é instalado corretamente quando aparece a mensagem *package 'fBasics' successfully unpacked and MD5 sums checked* no seu console. Ademais, no processo de instalação de um pacote, pode ser necessário instalar outros pacotes, chamados de *dependentes*, porque uma ou mais funções do pacote que você quer instalar fazem uso de funções de outros pacotes. Assim, **para que a instalação seja efetuada com sucesso, é preciso que todos os pacotes dependentes sejam instalados corretamente.** Por isso, procure verificar as mensagens no console de forma a verificar se o pacote foi instalado corretamente, como mostrado na figura 2.

```
> install.packages('fBasics')
Warning in install.packages :
  cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/src/contrib/PACKAGES.rds'
: HTTP status was '404 Not Found'
Installing package into 'C:/Users/Vitor Wilher/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
Warning in install.packages :
  cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/3.4/P
ACKAGES.rds': HTTP status was '404 Not Found'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/fBasics_3011.87.zi
p'
Content type 'application/zip' length 1559813 bytes (1.5 MB)
downloaded 1.5 MB

package 'fBasics' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Vitor Wilher\AppData\Local\Temp\Rtmp6rMrmM\downloaded_packages
```

Figure 2: Instalando pacotes.

Uma vez instalado, os seus pacotes ficam armazenados na pasta *library* da versão correspondente do seu R.[3] Você pode ver a lista de pacotes naturamente, diretamente no RStudio, através da aba *Packages* no canto inferior direito.[4]

Uma outra forma muito comum de instalar pacotes é através do **GitHub**, uma plataforma bem bacana utilizada por desenvolvedores para compartilhar códigos. Ali ficam armazenados pacotes *em desenvolvimento*,

[3] Toda vez que você instalar uma nova versão do R, uma dica é pegar os pacotes da sua pasta library e copiar os mesmos para a pasta library da nova versão, de modo a não ter que instalar tudo novamente.

[4] Uma lista dos pacotes disponíveis pode ser encontrada aqui.

que ainda não estão disponíveis no CRAN. Para instalar um pacote via GitHub, você deve ter instalado primeiro o pacote *devtools*. O código abaixo exemplifica com a instalação do pacote brasileiro BETS.[5]

```
install.packages('devtools')
require(devtools)
install_github("pedrocostaferreira/BETS")
```

No código acima, nos instalamos o pacote *devtools*, depois **carregamos** o mesmo com a função `require` - também pode ser utilizado a função `library` - e então utilizamos a função `install_github` para instalar um pacote armazenado no GitHub.

Às vezes pode ser necessário instalar uma versão antiga de um pacote, seja porque a versão atual tem algum *bug* - acontece! - seja porque ela é incompatível com o pacote que queremos usar. Para instalar versões antigas, você tem duas opções. Uma é subir o arquivo fonte zipado do pacote diretamente no RStudio, na opção *Tools* e *Install Packages*. Outra é utilizar a função `install_version` do pacote *devtools*, como abaixo.

```
install_version("DBI", version = "0.5", repos = "http://cran.us.r-project.org")
```

É muito comum os alunos receberem o erro de *caminho inválido* para a biblioteca de pacotes. Nesse caso, você pode especificar o caminho da biblioteca com o código abaixo.

```
library(withr) # É instalado e carregado junto com o devtools
withr::with_libpaths(new = "C:/Program Files/R/R-3.4.1/library", install_github("StatsWithR/statsr"))
```

Por fim, vale ressaltar que todo pacote disponível no CRAN tem uma página com suas informações. Veja o exemplo do *devtools* aqui. Lá você pode ver um resumo do pacote, um pdf com as principais funções, versões antigas, etc.

1.6 Obtendo ajuda

Uma parte importante de aprender uma nova linguagem é saber conseguir resolver os pepinos que irão surgir ao longo do caminho. E, acredite: eles serão muitos! Mas não se desespere. Como há uma comunidade incrível trabalhando com **R**, existem inúmeros sites, blogs, tutoriais, apostilas, etc, que podem lhe ajudar. No próprio ambiente do **RStudio**, você pode invocar ajuda com os comandos abaixo.[6]

```
help.start() # Você terá acesso à página de ajuda do R.
help("read.csv") # Uma ajuda sobre a função 'read.csv'
?read.csv # A mesma coisa do comando acima.
```

Caso continue com dúvidas, entretanto, nada melhor do que o bom e velho Google. Recomendo que você jogue sempre sua dúvida lá, antes de qualquer coisa. Uma dica: jogue ela em inglês e verá logo na primeira página um monte de gente com o mesmo problema que você! Em particular, um fórum bastante conhecido é o Stackoverflow, onde pessoas mais experientes procuram ajudar os mais novos. Caso sua dúvida não seja resolvível via google, considere jogá-la nesse fórum. A versão em português do fórum está disponível em https://pt.stackoverflow.com/questions/tagged/r.

Outra coisa que você deve fazer a partir de agora é acompanhar blogs que falam de **R**. Modéstia ao largo, não se esqueça de acompanhar o meu próprio site www.analisemacro.com.br/blog. Lá faço alguns exercícios interessantes de como usar o **R** para resolver nossos problemas diários de análise de dados. No exterior, há uma infinidade de blogs reunidos no famoso R-bloggers.

[5] Para saber mais sobre o BETS, vá em https://github.com/pedrocostaferreira/BETS.

[6] Observe que utilizamos o 'jogo da velha' # para colocar um comentário após a função. Isso é extremamente útil para que você lembre seus passos em um script no **R**, bem como no momento que você compartilhar seu script com outra pessoa.

2 Sobre o Suporte do Curso

Os alunos do Curso de Introdução ao R para Data Science contarão com suporte para dúvidas. Para ter acesso ao suporte, você deve acessar o arquivo `respostas.Rmd`, disponível no arquivo zipado do link *Exercícios*, na seção Primeiros Passos. Ao final, você deve compilar o arquivo `respostas.Rmd`, gerando um pdf, conforme foi ensinado na primeira aula do Curso. Uma vez gerado o pdf, você deve enviá-lo para o professor via o botão azul no canto inferior direito da plataforma, conforme a figura abaixo. Demais dúvidas também devem ser enviadas por esse canal.

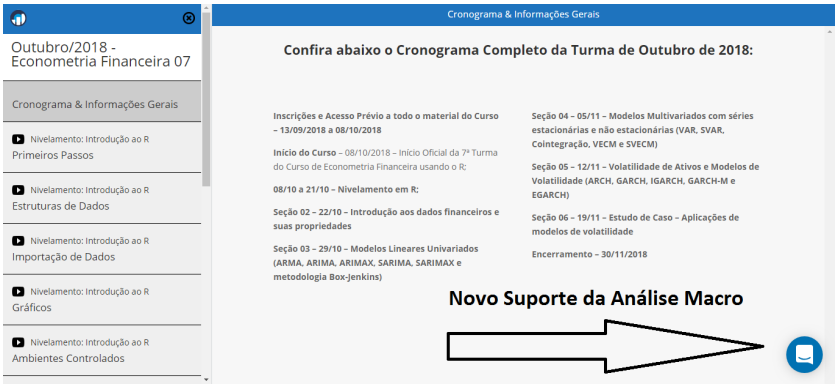

Figure 3: Suporte de Dúvidas da Análise Macro.

Referências

Grolemund, G., and H. Wickham. 2017. *R for Data Science*. O'Reilly Media.