Nuevas Perspectivas - Master de Agrobiologia Ambiental

Introdução ao R: Noções básicas de estatística e dados visualização

Adrian Gracia Romero, estudante de doutorado. ✉ a.graciaromero@ub.edu

Grupo Integrativo de Ecofisiologia de Culturas, Universidade de Barcelona

Conteúdo

1. Configuração de R e RStudio

Para baixar o R, vá para CRAN, a rede abrangente de arquivos R. CRAN é composto por um conjunto de servidores espelho distribuídos ao redor do mundo e é usado para distribuir pacotes R e R.

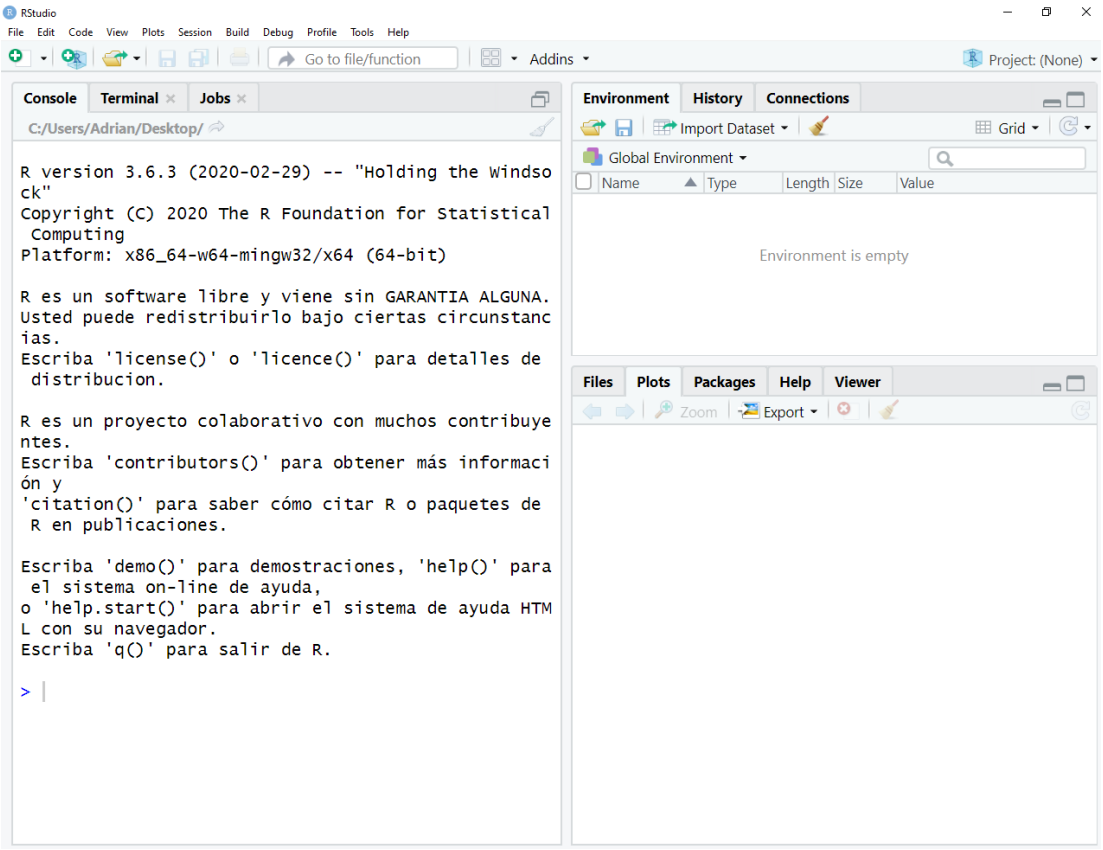
Janelas: https://cran.r-project.org/bin/windows/base/
Mac OS: https://cran.r-project.org/bin/macosx/

RStudio é um ambiente de desenvolvimento integrado, ou IDE, para programação R.

Janelas: https://www.rstudio.com/products/rstudio/download/#download
Mac OS : https://www.rstudio.com/products/rstudio/download/#download

2. Conceitos básicos

Ao iniciar o RStudio, você verá duas regiões principais na interface:



A interface do RStudio é simples. Você digita o código R na linha inferior do painel do console RStudio e clica em Enter para executá-lo. O código que você digita é chamado de comando, porque ele comandará seu computador para fazer algo por você. A linha em que você digita é chamada de linha de comando.

Quando você digita um comando no prompt e pressiona Enter, o computador executa o comando e mostra os resultados. Em seguida, o RStudio exibe um novo prompt para seu próximo comando. Por exemplo, se você digitar 1 + 1 e pressionar Enter, o RStudio exibirá:

```
> 1 + 1
[1] 2
```

Você notará que um [1] aparece próximo ao seu resultado. R está apenas informando que esta linha começa com o primeiro valor em seu resultado. Alguns comandos retornam mais de um valor e seus resultados podem preencher várias linhas. Por exemplo, o comando 100: 130 retorna 31 valores; ele cria uma sequência de inteiros de 100 a 130. Observe que novos números entre colchetes aparecem no início da segunda e da terceira linhas de saída. Esses números significam apenas que a segunda linha começa com o 14º valor no resultado e a terceira linha começa com o 25º valor. Você pode ignorar principalmente os números que aparecem entre colchetes:

```
> 100: 130
[1] 100 101 102 103 104 105 106 107 108 109 110 111 112 [14] 113 114 115
116 117 118 119 120 121 122 123 124 125 [25] 126 127 128 129 130
```

Se você digitar um comando incompleto e pressionar Enter, R exibirá um prompt +, o que significa que está esperando que você digite o resto do comando. Conclua o comando ou pressione Escape para recomeçar:

```
> 5 -
+
+ 1
[1] 4
```

R permite salvar dados armazenando-os dentro de um objeto R. O que é um objeto? Apenas um nome que você pode usar para acessar os dados armazenados. Por exemplo, você pode salvar dados em um objeto como a ou b. Quando você cria um objeto, ele aparecerá no painel de ambiente do RStudio. Sempre que R encontrar o objeto, ele irá substituí-lo pelos dados salvos nele, da seguinte forma:

```
a <- 1
uma
[1] 1

a + 2
[1] 3
```

Para criar um objeto R, escolha um nome e use o símbolo menor que, <, seguido de um sinal de menos, -, para salvar os dados nele. Esta combinação se parece com uma seta, <-. R fará um objeto, dará a ele seu nome e armazenará nele tudo o que segue a seta. Quando você pergunta a R o que está acontecendo
a, ele informa na próxima linha. Você também pode usar seu objeto em novos comandos R. Como o valor 1 foi armazenado anteriormente, agora você está adicionando 1 a 2. Portanto, para outro exemplo, o código a seguir criaria um objeto denominado die que contém os números de um a seis. Para ver o que está armazenado em um objeto, basta digitar o nome do próprio objeto:

```
a <- 1: 6
uma
[1] 1 2 3 4 5 6

a <- c (1,3,4,7)
uma
[1] 1 3 4 7
```

3. Pacotes e funções

Um pacote R é uma coleção de funções, dados e documentação que estende os recursos da base R. Você pode instalar o pacote completo com uma única linha de código:

```
install.packages ("NOME DO PACOTE")
```

Em seu próprio computador, digite essa linha de código no console e pressione Enter para executá-lo. R baixará os pacotes do CRAN e os instalará em seu computador. Se tiver problemas de instalação, certifique-se de que está conectado à Internet e de que https://cloud.rproject.org/ não está bloqueado por seu firewall ou proxy. Você não será capaz de usar as funções, objetos e arquivos de ajuda em um pacote até carregá-lo com biblioteca (): su

```
biblioteca (NOME DO PACOTE) #Sem “”
```

Cada função em R tem três partes básicas: um nome, um corpo de código e um conjunto de argumentos.

Já existem algumas funções em R. Um exemplo é matriz() , essa é uma função para criar a matriz de dados. A sintaxe comum desta função é matriz (dados, nrow, ncol, byrow = F) . Dados são os valores que irão integrar a matriz. A dimensão da matriz pode ser definida passando o valor apropriado para os argumentos agora e ncol . Podemos ver que a matriz é preenchida em colunas. Isso pode ser revertido para o preenchimento de linha passando TRUE para o argumento Byrow . Esses nomes podem ser acessados ou alterados com duas funções úteis colnames () e rownames () .

```
> matriz (1: 6)
     [, 1]
[1,] 1
[2,] 2
[3,] 3
[4,] 4
[5,] 5
[6,] 6

> matriz (1: 6, nrow = 2)
     [, 1] [, 2] [, 3]
[1,] 1 3 5
[2,] 2 4 6

> matriz (1: 6, nrow = 2, byrow = T)
     [, 1] [, 2] [, 3]
[1,] 1 2 3
[2,] 4 5 6
```

4. Crie um novo script

Um script é simplesmente um arquivo de texto contendo um conjunto de comandos e comentários. O script pode ser salvo e usado posteriormente para reexecutar os comandos salvos. O script também pode ser editado para que você possa executar uma versão modificada dos comandos. É fácil criar um novo script no RStudio. Você pode abrir um novo script vazio clicando no ícone Novo arquivo no canto superior esquerdo da barra de ferramentas principal do RStudio. Este ícone se parece com um quadrado branco com um sinal de mais branco em um círculo verde. Clicar no ícone abre o menu Novo arquivo. Clique na opção de menu R Script e o editor de scripts será aberto com um script vazio.

5. Antes de começar: Diretório de trabalho

O diretório de trabalho é a pasta onde os dados serão salvos. Podemos escolher nosso diretório de trabalho usando a função setwd ( ) Para verificar se o diretório de trabalho definido é o correto, use a função getwd () . Se você quiser verificar o que está dentro do diretório de trabalho, use a função
list.files ().

```
setwd ("C: / Users / Adrian / Documents / Projectes")
getwd ()
list.files ()
```

6. Importe os dados de um documento excel

Carregar arquivo excel com o pacote readxl e a função read_excel ().
https://cran.project.org/web/packages/readxl/readxl.pdf

read_excel () chama excel_format () para determinar se o caminho é xls ou xlsx, com base na extensão do arquivo e no próprio arquivo, nessa ordem. Caminho para o arquivo xls / xlsx. Folha ler. Ou uma string (o nome

de uma folha) ou um número inteiro (a posição da folha). Ignorado se a planilha for especificada por intervalo. Se nenhum dos argumentos especificar a planilha, o padrão é a primeira planilha. col_names TRUE para usar a primeira linha como nomes de coluna, FALSE para obter nomes padrão ou um vetor de caracteres que fornece um nome para cada coluna. Se o usuário fornecer col_types como um vetor, col_names pode ter uma entrada por coluna,
ou seja, têm o mesmo comprimento que col_types ou uma entrada por coluna não ignorada.

```
install.packages ("readxl")
biblioteca (readxl)
mydata <- read_excel (path, sheet = NULL, col_names = T) View (mydata)
```

Outras funções úteis

dim () : informa as dimensões de uma matriz.
colnames () : nomes das colunas de uma matriz.
rownames () : nomes das linhas de uma matriz.
str () : exibe a estrutura interna de um objeto R
comprimento() : comprimento de uma matriz
Aplique() : Aplica uma função às colunas ou linhas de uma matriz
cbind () : Adicionar uma nova coluna
rbind () : Adicionar uma nova linha

A interpretação de um fator depende tanto dos códigos quanto do atributo "níveis". Tenha cuidado apenas para comparar fatores com o mesmo conjunto de níveis (na mesma ordem). Em particular,
as.numeric () aplicado a um fator não tem sentido e pode acontecer por coerção implícita. Para transformar uma coluna em valores categóricos, use as.factor () .

7. Manipulação de dados

Os pacotes dplyr e tidyr são úteis para manipular facilmente os dados. Usar filtro() para filtrar os dados em grupos, select () para selecionar colunas, subconjunto () para criar novos dados usando apenas algumas colunas ...

```
# # Pacotes de manipulação de dados "dplyr" e "tidyr"
# # https://cran.r-project.org/web/packages/dplyr/dplyr.pdf install.packages
("dplyr")

biblioteca (dplyr)
# # https://cran.r-project.org/web/packages/tidyr/tidyr.pdf install.packages
("tidyr")

biblioteca (tidyr)
mydata.A <- filter (mydata, Locality == "Aranjuez")
    mydata.AI <- filter (mydata.A, Treatment == "Irrigação") mydata.AR <-
  filter (mydata.A, Treatment == "Rainfed") mydata.V <- filter (mydata,
  Locality == "Valladolid ")
    mydata.VI <- filter (mydata.A, Treatment == "Irrigação") mydata.VR <-
    filter (mydata.A, Treatment == "Rainfed")


mydata.factors <- subconjunto (mydata,, -c (NDVI.avg_field: harvest.index))
```

8. Noções básicas de estatística

Para o cálculo das principais estatísticas básicas, podemos usar a função summarySE () , e vamos precisar dos pacotes treliça , plyr e Rmisc . Usando o argumento mesurevar vamos indicar o parâmetro que queremos estudar e usando groupvars vamos indicar a variável de agrupamento.

```
# # Fornece contagem, média, desvio padrão, erro padrão da média e intervalo de confiança
# # https://cran.r-project.org/web/packages/Rmisc/Rmisc.pdf install.packages ("Rmisc")

install.packages ("lattice")
# # se não funcionar, feche e abra a biblioteca do programa (lattice)

biblioteca (plyr)
biblioteca (Rmisc)

resumoSE (mydata, mensevar = "GY", groupvars = c ("Localidade", "Tratamento"))
```

Outra opção é usar a função describeBy () dentro do pacote “Psicológico” .

```
# # Relatório de estatísticas básicas de resumo por uma variável de agrupamento
# # https://cran.r-project.org/web/packages/psych/psych.pdf install.packages ("psych")

biblioteca (psicologia)
describeBy (mydata.A, group = mydata.A $ Treatment)
```

9. ANOVA

A análise ANOVA pode ser feita usando a função anova () .

```
anova (lm (GY ~ Genotype, mydata.AI))
```

10. Correlações

Para o cálculo de uma correlação entre os parâmetros, podemos usar a função cor.test ().

```
cor.test (mydata.AI $ GA.g, mydata.AI $ GY)
```

11. Visualização de dados com “ggplot2”

Um dos pacotes mais úteis para visualização de dados é ggplot2 . Todos os gráficos ggplot2 começam com uma chamada para ggplot () , fornecendo dados padrão e mapeamentos estéticos, especificados por aes () . Em seguida, você adiciona camadas, escalas, coordenadas e facetas com +. Para salvar um gráfico no disco, use ggsave () .

Seguindo as páginas abaixo, você terá sucesso fazendo quase qualquer gráfico.

https://ggplot2.tidyverse.org/reference/
https://www.r-graph-gallery.com/index.html