

Readme.rmd

Sergio Pedro R Oliveira

2023-01-30

Contents

1	Objetivo	1
2	Livro de referência	2
3	Cap 1 - Instalação do R e Rstudio	2
4	Cap 2 - Pacote base e funções estatísticas básicas	2
4.1	Operações matemáticas básicas	2
4.2	Vetor	2
4.3	Tabela de dados (data.frame) e matrizes	3
4.4	Visualizando dados	4
5	Cap 3 - Principais pacotes	6
6	Cap 4 - R Markdown	8
7	Cap 5 - Pacotes do Tidyverse e identificando/mudando tipos de variáveis	10
8	Cap 6 - Pacote data.table	13
9	Cap 7 - Gráficos básicos e pacote ggplot2	14
10	Andamento dos Estudos	17
10.1	Assunto em andamento:	17
10.2	Em andamento:	17
10.3	Vazios:	17
10.4	Finalizando detalhes:	17

1 Objetivo

Estudo dirigido de linguagem R.

2 Livro de referência

Utilizando a Linguagem R.
Editora: ALTA BOOKS EDITORA

3 Cap 1 - Instalação do R e Rstudio

- Download da linguagem R:
<https://www.r-project.org/>
- Download Rstudio IDE:
<https://posit.co/downloads/>

4 Cap 2 - Pacote base e funções estatísticas básicas

4.1 Operações matemáticas básicas

Nome da operação	Operação	Resultado
Adição	5+4	[9]
Subtração	6-2	[4]
Multiplicação	7*3	[21]
Divisão	45/9	[5]
Potência	2^2	[4]
Raiz	sqrt(121)	[11]
Exponencial	exp(0)	[1]
Log na base e	log(1)	[0]
Log na base 10	log10(1)	[0]
Log na base 2	log2(4)	[2]
Log na base 3 ou qualquer outra	log(9,3)	[2]

4.2 Vetor

- Para criar um vetor usamos a função `c()`.
- Os argumentos são separados por virgula dentro do parênteses.
- strings devem estar entre aspas duplas.
Ex.: `c("um", "sete", "nove")`
- Vetores são compostos de elementos todos do mesmo tipo.
- Armazenando vetores em um objeto:
Ex.: `obj_qualquer <- c(1,2,3)`

4.3 Tabela de dados (**data.frame**) e matrizes

4.3.1 **data.frame**

- Uma tabela onde cada coluna é um vetor.
- Como cada coluna é um vetor, cada coluna pode ser de um tipo diferente.
Ex.: `nome_data.frame <- data.frame(vetor_1, vetor_2)`
- Acrescentando uma nova coluna ao **data.frame**.
Ex.: `nome_data.frame <- data.frame(nome_data.frame, vetor_3)`
- Para visualizar um **data.frame** podemos usar a função **View()**.
Ex.: `View(nome_data.frame)`

4.3.2 Matrizes

- A diferença entre **matrizes** e **data.frames**, é que no caso das matrizes todas as colunas e linhas devem ser do mesmo tipo. Enquanto nos **data.frames** as colunas podem ser de tipos diferentes.
- Para adicionar uma coluna numa matriz, usamos a função `cbind()`.
Ex.: `nome_matriz <- cbind(vetor_1, vetor_2, ...)`
- Para adicionar uma linha numa matriz, usamos a função `rbind()`.
Ex.: `nome_matriz <- rbind(vetor_3, vetor_4, ...)`
- Quando inserimos dados (vetor) de naturezas diferentes (tipos) numa matriz, ela converte todos os dados para um único tipo. A princípio *string* (*chr*).

4.4 Visualizando dados

4.4.1 View() - visualização de dados

- Podemos visualizar dados de duas formas:
 - Escrevendo o nome da variável
O valor dela será impressa na tela.
 - Atraves da função **View()**
Ao chamar a função View() e colocar dentro a variavel que queremos ver, será exibido uma nova janela com o valor da variável numa tabela.

4.4.2 str() - estrutura de objetos

- A função “**str()**” retorna a estrutura do objeto do argumento.
- Retorna diversos dados, entre eles:
 - A classe do objeto.
 - Tamanho do objeto.
 - A lista, ou vetor, dos campos com o tipo e tamanho.
- Sintaxe:
str(argumento)

4.4.3 summary() - resumo de variáveis

- A função **summary()** retorna o resumo de variaveis.
- O retorno depende do argumento (se for um vetor, uma lista, um data.frame).
- O retorno para uma matriz ou **data.frame**, vai ser os metodos aplicados a cada campo/coluna.
- O retorno da função, no geral, retorna diversos metodos aplicados aos dados, tais como:
 - valor mínimo
 - 1º quantil
 - valor da mediana
 - valor da media
 - 3º quantil
 - valor máximo
- Sintaxe:
summary(nome_variavel)

4.4.4 `class()` - classe de objetos

- A função “**class()**” retorna a que classe do objeto do argumento pertence.
- Basicamente diz se o objeto é numerico, string, vetor, lista, data.frame, matriz, ...
- Sintaxe:
class(*argumento*)

5 Cap 3 - Principais pacotes

1. Principais pacotes:

- **Rmarkdown**
Produção de relatórios(html, pdf, doc, md).
- **knitr**
Interpretação e compilação do documento rmd.
- **data.table**
Exploração de data.frames.
- **janitor**
Limpeza de dados.
- **DescTools**
Análise descritiva de dados.
- **tidyverse**
conjunto de pacotes.
 - **readr**
Importação e leitura de arquivos de dados.
 - **tibble**
estruturação de data.frame.
 - **dplyr**
Manipulação de data.frame.
 - **tidyr**
Organização de data.frame.
 - **ggplot2**
Visualização de dados, produção de gráficos.
 - **purrr**
Manipulação de vetores e listas.

2. Instalação de pacotes:

- sintaxe de instalação:
install.packages(*“nome do pacote”*)
- sintaxe de variáveis instalações simultâneas:
install.packages(c(*“nome do pacote”*, *“nome do pacote”*, ...), dependencies = **TRUE**)

6 Cap 4 - R Markdown

1. Preâmbulo:

- *title*: “Titulo desejado”
- *author*: “Nome dos autores”
- *date*: “Data do dia da compilação”, para adicionar a data atual pode usar a função através do chunk “r Sys.Date()”
- *output*: o tipo de saída, podem ser:
 - Documentos:
 - * *pdf_document*
 - * *md_document*
 - * *html_document*
 - * *word_document*
 - * *odt_document*
 - * *rtf_document*
 - Apresentação:
 - * *powerpoint_presentation*
 - * *ioslides_presentation*
 - * *beamer_presentation*
 - mais:
 - * *flexdashboard::flex_dashboard*
 - * *github_document*

2. Chunks/codigos embutidos:

3. Titulos e subtítulos:

4. Listas e blocos de citação:

5. Inserir tabelas:

6. Fontes:

7. Hiperlinks e imagens:

- Hiperlinks

- Imagens

8. Letras gregas:

9. Fórmulas:

- Subscritos e superescritos

- Sublinhados, sobrelinhas e vetores

- Frações, matrizes e chavetas

- Expressões

- Sinais e setas

7 Cap 5 - Pacotes do Tidyverse e identificando/mudando tipos de variaveis

1. identificando/mudando tipos de variaveis

- i. identificando
uso do **is**.
- ii. mudando o tipo de variavel:
uso do **as**.

2. pacotes do Tidyverse:

- **readr**
Leitura de dados.
- **tibble**
Tipo de data.frame.
- **magrittr**
Operador pipe '`%>%`', concatena linhas de comando.
- **dplyr**
Manipulação de dados.
 - i. manipulação de dados:
 - *select*
seleciona e retorna as colunas selecionadas da tabela.
 - *pull*
extrai uma coluna de uma tabela de dados e retorna ela como vetor.
 - *filter*
filtra linhas.
 - *distinct*
remove linhas com valores repetidos.
 - *arrange*
reordena ou combina linhas.
 - *mutate*
cria novas colunas.
 - *transmute*
cria novas colunas, mas não adiciona na base de dados.

- *summarise*
sumariza valores.
- *group_by*
permite operações por grupo.
- *add_column*
adiciona novas colunas.
- *add_row*
adiciona novas linhas.
- *rename*
renomeia uma coluna.

ii. combinando tabelas de dados:

- *bind_cols*
Une duas tabelas lado a lado. acrescenta numeração as colunas repetidas.
É necessario que tenha o mesmo numero de linhas nas duas tabelas para fazer essa combinação.
- *bind_rows*
Une duas tabelas sobrepostas.
Quando não há correspondencia o comando retorna **NA**.
- *inner_join*
A tabela final será o resultado da intersecção das duas colunas de x e y, que possuem pelo menos uma coluna em comum, a coluna chave.
Junta duas colunas pela interseção.
- *left_join*
Une duas tabelas, definindo qual será a tabela principal e a unida a esquerda da outra. Esse fator muda a interpretação das linhas/registros correspondentes uma na outra, no caso, a tabela principal e tabela que será colocada a esquerda.
É necessario que tenha pelo menos uma coluna em comum, uma coluna chave.
- *right_join*
Une duas tabelas, definindo qual será a tabela principal e a unida a direita da outra. Esse fator muda a interpretação das linhas/registros correspondentes uma na outra, no caso, a tabela principal e tabela que será colocada a direita.
É necessario que tenha pelo menos uma coluna em comum, uma coluna chave.
- *full_join*
Une duas tabelas. Prestar atenção na junção das linhas/registros que formam novas informações, através da junção de correspondentes.
É necessario que tenha pelo menos uma coluna em comum, uma coluna chave.
- *intersect*
Retorna a interseção entre tabelas.
- *union*
Retorna a união de tabelas.
- *setdiff*

Retorna a diferença entre tabelas.

- *setequal*

Esse comando verifica se duas tabelas de dados possuem linhas com os mesmos valores, independentemente da ordem em que tais valores se apresentem. retorna **TRUE**, se os registros forem iguais, ou **FALSE**, se os registros forem diferentes.

- **tidyr**

Organização de dados.

- *pivot_longer* ou *gather*

Converte a tabela de dados para o formato longo. (larga -> longo)

- *pivot_wider* ou *spread*

Converte a tabela de dados para o formato larga. (longo -> larga)

- *separate*

Separa as respostas que estão em uma unica coluna para diversas colunas.

- *unite*

O comando unite é utilizado para unir duas ou mais colunas em uma unica coluna.

- *complete*

Completa as combinações de duas colunas, se não houver valor completa com *NA*.

- *drop_na*

Elimina as linhas, especificadas ou não, com valor *NA*.

- *replace_na*

Substitui o valor *NA* por outro valor especificado.

8 Cap 6 - Pacote data.table

1. data.table

- Manipulando linhas
- Manipulando colunas
- Sumarizando dados
- Operando um subconjunto de dados
 - *lapply*
- modificando dados com set:
 - *set*
modificando um valor.
 - *setnames*
modificando nome da coluna.
 - *setorder*
modificando ordem das linhas.
 - *setcolorder*
modificando ordem das colunas.

9 Cap 7 - Gráficos basicos e pacote ggplot2

1. Gráficos basicos:

- Gráfico de barras
barplot
- Gráfico circular (pizza)
pie
- Gráfico de linhas
plot
 - Para adicionar mais linhas no grafico.
lines
- Gráfico de dispersão
 - Para obter a correlação.
cor(x,y)
 - Para obter o coeficiente da reta de regressão.
lm(y ~x)\$coef
 - Adiciona a reta tracejada.
abline
- Histograma
hist
- **Boxplot** (diagrama de caixa)

2. Pacote **ggplot2**

- Constroi diversos tipos de graficos a partir da mesma estrutura de componentes:
 - *data*: referente ao banco de dados.
 - *geom_forma*: um rol de tipos possiveis de representação dos dados.
 - *coord_system*: referente ao sistema de coordenadas, que podem ser cartesianas, polares e projeção de mapas.
- i) O que precisa para fazer o grafico?
 - A. Um nome de objeto para guardar o grafico (uma variavel).
 - B. A base de dados que será utilizada para a plotagem.
ggplot(data=nome_da_base)
 - C. Descrever como as variaveis serão utilizadas na plotagem:
aes(x=..., y=..., ...)
 - D. Especificar o tipo de grafico:

geom_forma(...)

E. Utilizar o operador “+” para adicionar camadas ao objeto **ggplot** criado.

F. Pacotes auxiliares como *ggthemes* e *grid*, dentre outros.

ii) Quais formatos podemos utilizar no ggplot2 - geom_forma?

Forma	Tipo de grafico
geom_area ou geom_ribbon	Produce um grafico para visualizar área sob a curva ou entre curvas.
geom_bar ou geom_col	Produce um grafico de colunas do vetor x.
geom_bar+coord_polar	Produce um grafico circular (Pizza).
geom_boxplot	Produce o boxplot de x.
geom_curve	Produce um grafico em curva.
geom_density	Produce um grafico da densidade de x.
geom_dotplot	Produce um grafico de pontos.
geom_histogram	Produce um histograma do vetor x.
geom_line, geom_abline, geom_hline, geom_vline	Produce um grafico de linhas
geom_point	Produce um grafico de dispersão entre x e y.
geom_qq ou geom_qq_line	plota os quantis de x usando como base a curva normal.
geom_tile, geom_rect ou geom_raster	Produce uma grade de retangulos.
geom_violin	Produce um grafico em forma de violino.

iii) Nome dos argumentos para adicionar efeito em graficos do pacote ggplot2.

Função	Efeito no grafico
autoplot	Produce um grafico apropriado para o tipo de variavel.
coord_cartesian	Coordenada cartesiana.
coord_fixed	Coordenada cartesiana com razão entre eixo x e y fixada.
coord_flip	Inverte a posição dos eixos x e y.
coord_polar	Coordenada polar.
geom_blank	Janela em branco.
geom_jitter	Produce um efeito jitter.
geom_smooth	Produce uma curva suavizada.
geom_text	Aplica texto a janela grafica.
scale_fill_(=brewer ou grey ou gradient)	Define a escala de cores.
scale_*_contínuos	Define parametros para o eixo x ou y contínuos.
scale_*_discrete	Define parametros para o eixo x ou y discreto.
scale_*_manual	Define parametros para os eixos manualmente.

- Definindo um tema para o grafico **ggplot**.

- *theme_gray*

- Fundo cinza e linhas grandes brancas.

- *theme_bw*

- O classico preto e branco. Otimo para projetor.

- *theme_linedraw*
Linhas pretas de varias larguras num fundo branco. semelhante ao theme_bw.
- *theme_light*
Semelhante ao theme_linedraw, porem com as linhas mais cinza claro, para dar atenção aos dados.
- *theme_dark*
Versão escura do theme_light, com o fundo escuro, util para criar linhas finas coloridas.
- *theme_minimal*
Um tema minimalista sem anotações de fundo.
- *theme_classic*
Tema classico, com linhas do eixo x e y, sem linhas de grade.
- *theme_void*
Um tema completamente vazio.

10 Andamento dos Estudos

10.1 Assunto em andamento:

Atualmente estou estudando Cap.7, pacote ggplot2.

E revisando Cap.2, acessando valores de posições específicas de objetos.

10.2 Em andamento:

10.3 Vazios:

10.4 Finalizando detalhes: