

# R para usuários de Python

Patrícia de Siqueira Ramos

UNIFAL-MG, *campus* Varginha

22 de Abril de 2019

# Dicas sobre R

- comparação python × R
- dicas de R
- tutorial de R
- rstudio cloud (online)
- r for data science (site)
- google
- stackoverflow

# Diferenças entre R e Python - I

função	python	R
criar variáveis	<code>a = 23</code>	<code>a = 23</code> <del><code>a ← 23</code></del>
ler dataframes	<code>pd.read_csv('restaurante1.csv')</code>	<code>read.csv('restaurante1.csv')</code>
ver tamanho do dataframe	<code>restaurante1.shape</code>	<code>dim(restaurante1)</code>
criar comentários	<code># linha comentada</code>	<code># linha comentada</code>
subselecionar dataframe por linhas	<code>restaurante1.query('receitas &lt; 10000')</code>	<code>restaurante1 %&gt;% filter(receitas &lt; 10000)</code>
subselecionar dataframe por colunas	<code>restaurante1.receitas</code>	<code>restaurante1\$receitas</code>
criar vetor	<code>a = np.array([3, 4, 5])</code>	<code>a &lt;- c(3, 4, 5)</code>
subselecionar vetor	<code>a[1:2]</code>	<code>a[1:2]</code>
regressão	<code>smf.ols('receita ~ despesas + empresa', data=dois_restaurantes).fit()</code>	<code>lm('receita ~ despesas + empresa', data=dois_restaurantes)</code>
histograma	<code>restaurante1.receitas.hist()</code>	<code>ggplot(restaurante1, aes(receitas)) + geom_histogram()</code>
valores booleanos	<code>True/False</code>	<code>T/F, TRUE/FALSE</code>
primeira posição de um vetor	<code>minha_lista[0]</code>	<code>minha_lista[1]</code>
não use ponto no nome de objetos	<code>novo_objeto = 23</code>	<code>novo_objeto = 23</code> <del><code>novo.objeto = 23</code></del>
potenciação	<code>2**3</code>	<code>2**3</code> <del><code>2^3</code></del>

## Diferenças entre R e Python - II

executar comandos	shift+enter	ctrl+enter
delimitação de funções e estruturas	indentação	chaves
interface	JupyterLab	JupyterLab <b>REStudio</b>
carregar pacotes	import pandas as pd	library(dplyr)
criar funções	def soma(a, b): return a + b	soma = function(a, b){ return(a + b) }
for loop	for i in range(1, 6): print(i)	for(i in 1:5){ print(i) }
while	while i < 5: print(i) i += 1	while(i < 5){ print(i) i = i + 1 }
if else	if x >= 0: print('não negativo') else: print('negativo')	if (x >= 0){ print('não negativo') } else { print('negativo') }

# Importar arquivos

- Os códigos abaixo assumem que os arquivos estão na sua pasta de trabalho definida em `setwd()` ou CTRL+SHIFT+H
- Para consultar qual a pasta de trabalho: `getwd()`

# Importar arquivos

- Os códigos abaixo assumem que os arquivos estão na sua pasta de trabalho definida em `setwd()` ou CTRL+SHIFT+H
- Para consultar qual a pasta de trabalho: `getwd()`
- ler dados tabulares

```
read.table(arquivo, header = FALSE, sep = , dec = ".")
```

- ler csv (*comma separated value*)

```
read.csv(arquivo, header = TRUE, sep = ",", dec = ".", ...)
```

- ler csv de outra forma - para valores decimais com vírgula e separador ';'

```
read.csv2(arquivo, header = TRUE, sep = ";", dec = ",", ...)
```

# Importar arquivos

- ler arquivos separados por TAB

```
read.delim(arquivo, header = TRUE, sep = '\t', dec = '.', ...)  
read.delim2(arquivo, header = TRUE, sep = '\t', dec = ',', ...)
```

- ler arquivo a partir de um site

```
read.delim('http://www.sthda.com/upload/boxplot_format.txt')
```

- outra opção: carregar de forma interativa

```
read.csv(file.choose())
```

- carregar rdata

```
load('arquivo.RData')
```

## Subseleção do *dataframe*

Quando precisamos selecionar partes do *dataframe* temos três opções:

```
# ler arquivo
```

```
rest = read.csv('restaurante1.csv')
```

```
rest
```

```
# 1 usando subset
```

```
subset(rest, mes == 'agosto') # dois sinais de igual
```

```
# 2 usando dplyr
```

```
rest %>% filter(mes == 'agosto')
```

```
# 3 usando apenas índices - lembrar da vírgula [ , ]
```

```
rest[rest$mes == 'agosto', ]
```



# Gráficos

## Opções:

- R básico
- ggplot:
  - segue uma “gramática de gráficos estatísticos” (explicações e exemplos aqui)
  - no começo parece um pouco diferente essa forma de construir gráficos.

# Gráficos no ggplot

## 'Gramática dos gráficos':

- Um gráfico estatístico é um mapeamento dos dados para propriedades estéticas (cor, forma, tamanho) e geométricas (pontos, linhas, barras) da tela
- O gráfico também pode conter transformações estatísticas
- Todas essas camadas formam o gráfico
- Os gráficos no *ggplot* são construídos por meio da adição de camadas:
  - uma base de dados (*dataframe*);
  - atributos estéticos (*aesthetics*);
  - objetos geométricos;
  - transformações estatísticas;
  - demais ajustes.