

Introdução à Programação e Ciência de Dados para a Gestão Pública

Rebeca de Jesus Carvalho

FGV CEPESP

rebeca.jesus.carvalho@gmail.com

Tópicos da aula

- 1 Introdução
- 2 O pacote *janitor*
- 3 *Factors* em R
- 4 Exportando dados em R
- 5 Gramática de gráficos *ggplot2*
- 6 Laboratório

Conteúdo da aula de hoje

- Vamos seguir com a gramática do *dplyr* para transformar os dados e produzir tabelas. Utilizaremos um pacote chamado *janitor*.
- Também aprenderemos um pouco sobre *factors*, que é a estrutura de dados utilizada em R para variáveis categóricas.
- Por fim, conheceremos a gramática de gráficos do pacote *ggplot2*.

Limpeza e produção de tabelas com *janitor*

- A função `clean_names()` permite que o usuário lide de forma fácil e rápida com nomes de variáveis fora das convenções (espaços, acentos e maiúsculas) utilizadas no R.
- A função `tabyl()` possibilita a produção de tabelas de frequência absoluta e relativa sem auxílio da função `group_by`.

Organização e padronização de tabelas

- No pacote *janitor*, existe um conjunto de funções de prefixo `adorn_*` que serve para alterar a estética das tabelas.
- **Exemplos:**
 - Transforma em percentual: `adorn_pct_formatting()`.
 - Arredonda as casas decimais: `adorn_rounding()`.
 - Adiciona os totais: `adorn_totals()`.
 - Adiciona o nome das variáveis de linha e de coluna em tabelas cruzadas: `adorn_title()`.
- Assim como no *dplyr*, você pode combinar mais de uma função `adorn_*` através do operador **pipe** (`%>%`).

Factors e ordenamento de variáveis categóricas

- *Factors* são uma classe de objetos no R criada para representar as variáveis categóricas numericamente. Essa classe possui um atributo especial: os *levels*. *Levels* são nada mais do que as categorias possíveis de uma variável categórica
- *Levels* são nada mais do que as categorias possíveis de uma variável categórica.
- A função `factor()` possibilita a transformação de uma variável *character* em *factor*. O parâmetro *level* é o que nos permite especificar a ordem das categorias da variável.
- **Exemplo:**
 - `factor(nivel_instrucao,`
 - `levels = c("Fundamental", "Médio", "Superior"))`

Exportando dados em R

- Para exportar arquivos de texto, usamos funções com prefixo *write_**, pertencentes ao pacote visto na aula passada, o *readr*
- Para exportar em formato **.CSV**: `write_csv()` ou `write_csv2()`.
- O primeiro argumento é o objeto a ser exportado e o segundo é o nome do arquivo que será criado.

Visualização de dados com o *ggplot2*

- A gramática de gráficos *ggplot2* é um dos destaques da linguagem R. Além de flexível e aplicável a diversas classes de objetos (*data frames*, objetos de mapa e redes, por exemplo), a qualidade dos gráficos é excepcionalmente boa..
- A essência do *ggplot2* é, basicamente, construir um gráfico camada por camada.
- Em relação aos gráficos do *R base*, o *ggplot2* se destaca pela facilidade de aprendizado, estrutura padronizada dos códigos e fácil personalização dos gráficos.

Sintaxe básica do *ggplot2*

- A estrutura básica de um gráfico do *ggplot2* é:
 - `ggplot(dataframe) +`
 - `geom_*(aes(variavelx, variavely))`
- A função *geom_** indica o tipo de gráfico que será gerado, isto é:
 - Gráfico de barras: `geom_bar()` ou `geom_histogram()`;
 - Gráfico de densidade: `geom_density()`;
 - Boxplots: `geom_boxplot()`;
 - Gráfico de pontos ou linhas: `geom_point()` ou `geom_line()`;
 - Mapas: `geom_map()`; entre outros.
- Usamos o operador `+` para adicionar novas camadas ou componentes ao gráfico.

Outros parâmetros e funções no *ggplot2*

- Define as variáveis que serão utilizadas: `aes()`.
- Define as cores: `color` ou `fill`.
- Define a transparência das cores: `alpha`.
- Define a posição dos gráficos: `position`.
- Separa um gráfico por uma variável discreta: `facet_wrap()`.
- Adiciona título, subtítulo e legendas: `labs()`.
- Altera o tamanho da geometria: `size`.
- Altera o formato da geometria: `shape`.

Laboratório

Agora é o momento de partir para a ação! Temos dois tutoriais agendados para hoje, prontos para serem explorados. Eles estão disponibilizados no GitHub, e você pode acessá-los clicando neste **link**.

Se deixou algum tutorial inacabado dos encontros anteriores, comece por eles. Caso contrário, prossiga.

Dúvidas?