

R para Ciência de Dados

Uma introdução ao tidyverse

Ítallo Silva

Quem sou eu?



Me chamo Ítallo!

Atualmente sou mestrando em
Computação aqui na UFCG!
Minha linha de pesquisa é na
área de Sistemas de
Recomendação e LLMs.

Trabalho com análise de dados e
aprendizagem de máquina desde
2020.

Você pode entrar em contato via
itallo@copin.ufcg.edu.br

Relembrando nosso título
R para Ciência de Dados

Quem sou eu?



Me chamo Ítallo!

Atualmente sou mestrando em
Computação aqui na UFCG!
Minha linha de pesquisa é na
área de Sistemas de
Recomendação e LLMs.

Trabalho com análise de dados e
aprendizagem de máquina desde
2020.

Você pode entrar em contato via
itallo@copin.ufcg.edu.br

Relembrando nosso título
R para Ciência de Dados

Quem sou eu?



Me chamo Ítallo!

Atualmente sou mestrando em
Computação aqui na UFCG!
Minha linha de pesquisa é na
área de Sistemas de
Recomendação e LLMs.

Trabalho com análise de dados e
aprendizagem de máquina desde
2020.

Você pode entrar em contato via
itallo@copin.ufcg.edu.br

Relembrando nosso título
R para Ciência de Dados

Quem sou eu?



Me chamo Ítallo!

Atualmente sou mestrando em
Computação aqui na UFCG!
Minha linha de pesquisa é na
área de Sistemas de
Recomendação e LLMs.

Trabalho com análise de dados e
aprendizagem de máquina desde
2020.

Você pode entrar em contato via
itallo@copin.ufcg.edu.br

Relembrando nosso título
R para Ciência de Dados

Leitura e processamento de dados

Momento expositivo I

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Leitura de dados com readr

readr

- ▶ Oferece uma forma rápida e fácil de ler dados tabulares de arquivos delimitados (CSV, TSV, etc).
- ▶ Isso pode ser feito através da família de funções `read_*`. A seguir vemos o uso da função `read_csv`.

```
library(readr)  
data <- read_csv("data/students.csv")
```

- ▶ Além da função `read_csv`, a biblioteca oferece ainda:
`read_delim`, `read_csv2`, `read_tsv`, `read_log` e `read_fwf`.

readr

As funções `read_*` compartilham alguns parâmetros entre si que merecem destaque:

- ▶ `file`
- ▶ `col_names`
- ▶ `col_types`
- ▶ `col_select`
- ▶ `na`

Organizando dados com tidy tidy

O objetivo do tidy é ajudá-lo a criar **tidy data**. **tidy data** são dados onde:

1. Cada coluna é variável.

tidy data descrevem uma maneira padrão de armazenar dados que é usada sempre que possível em todo o tidyverse. Vejamos alguns exemplos de dados.

Caso 1: É tidy data?

```
# A tibble: 12 x 4
```

	country	year	type	count
	<chr>	<dbl>	<chr>	<dbl>
1	Afghanistan	1999	cases	745
2	Afghanistan	1999	population	19987071
3	Afghanistan	2000	cases	2666
4	Afghanistan	2000	population	20595360
5	Brazil	1999	cases	37737
6	Brazil	2000	cases	100000000
7	Brazil	2000	population	180000000
8	China	1999	cases	100000000
9	China	1999	population	1200000000
10	China	2000	cases	100000000
11	China	2000	population	1200000000
12	China	2000	population	1200000000

Organizando dados com tidy

tidy

O objetivo do tidy é ajudá-lo a criar **tidy data**. **tidy data** são dados onde:

1. Cada coluna é variável.
2. Cada linha é uma observação.

tidy data descrevem uma maneira padrão de armazenar dados que é usada sempre que possível em todo o tidyverse. Vejamos alguns exemplos de dados.

Caso 1: É tidy data?

```
# A tibble: 12 x 4
```

	country	year	type	count
	<chr>	<dbl>	<chr>	<dbl>
1	Afghanistan	1999	cases	745
2	Afghanistan	1999	population	19987071
3	Afghanistan	2000	cases	2666
4	Afghanistan	2000	population	20595360
5	Brazil	1999	cases	37737
6	Brazil	2000	cases	100000000
7	Brazil	2000	population	180000000
8	China	1999	cases	100000000
9	China	1999	population	1200000000
10	China	2000	cases	100000000
11	China	2000	population	1200000000
12	China	2000	population	1200000000

Organizando dados com tidyR

tidyR

O objetivo do tidyR é ajudá-lo a criar **tidy data**. **tidy data** são dados onde:

1. Cada coluna é variável.
2. Cada linha é uma observação.
3. Cada célula é um valor único.

tidy data descrevem uma maneira padrão de armazenar dados que é usada sempre que possível em todo o tidyverse. Vejamos alguns exemplos de dados.

Caso 1: É tidy data?

```
# A tibble: 12 x 4
```

	country	year	type	count
	<chr>	<dbl>	<chr>	<dbl>
1	Afghanistan	1999	cases	745
2	Afghanistan	1999	population	19987071
3	Afghanistan	2000	cases	2666
4	Afghanistan	2000	population	20595360
5	Brazil	1999	cases	37737
6	Brazil	1999	population	188435300
7	Brazil	2000	cases	100000000
8	Brazil	2000	population	196587800
9	China	1999	cases	100000000
10	China	1999	population	1253730000
11	China	2000	cases	100000000
12	China	2000	population	1295330000

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.
- ▶ `rename`: permite mudar o nome das variáveis.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.
- ▶ `rename`: permite mudar o nome das variáveis.
- ▶ `mutate`: adiciona novas variáveis que são funções de variáveis existentes.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.
- ▶ `rename`: permite mudar o nome das variáveis.
- ▶ `mutate`: adiciona novas variáveis que são funções de variáveis existentes.
- ▶ `filter`: escolhe casos com base em seus valores.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.
- ▶ `rename`: permite mudar o nome das variáveis.
- ▶ `mutate`: adiciona novas variáveis que são funções de variáveis existentes.
- ▶ `filter`: escolhe casos com base em seus valores.
- ▶ `arrange`: altera a ordem das linhas.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.
- ▶ `rename`: permite mudar o nome das variáveis.
- ▶ `mutate`: adiciona novas variáveis que são funções de variáveis existentes.
- ▶ `filter`: escolhe casos com base em seus valores.
- ▶ `arrange`: altera a ordem das linhas.
- ▶ `summarise`: reduz vários valores a um único resumo.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.
- ▶ `rename`: permite mudar o nome das variáveis.
- ▶ `mutate`: adiciona novas variáveis que são funções de variáveis existentes.
- ▶ `filter`: escolhe casos com base em seus valores.
- ▶ `arrange`: altera a ordem das linhas.
- ▶ `summarise`: reduz vários valores a um único resumo.
- ▶ `group_by`: permite realizar qualquer operação “por grupo”.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Transformando dados com dplyr

dplyr

dplyr é uma gramática de manipulação de dados, fornecendo um conjunto consistente de verbos que ajudam a resolver os desafios mais comuns de manipulação de dados:

```
library(dplyr)
```

- ▶ `select`: escolhe variáveis com base em seus nomes.
- ▶ `rename`: permite mudar o nome das variáveis.
- ▶ `mutate`: adiciona novas variáveis que são funções de variáveis existentes.
- ▶ `filter`: escolhe casos com base em seus valores.
- ▶ `arrange`: altera a ordem das linhas.
- ▶ `summarise`: reduz vários valores a um único resumo.
- ▶ `group_by`: permite realizar qualquer operação “por grupo”.
- ▶ Além disso ela também fornece operações de `join`, porém não iremos trabalhar com elas nesse workshop.

`mtcars`

Para os exemplos a seguir utilizaremos o seguinte conjunto de

Leitura e processamento de dados

Momento prático I

Motivando a atividade prática

Motivando a atividade prática

- ▶ Considere uma empresa bem estabelecida que opera no setor de varejo de alimentos. Atualmente, eles têm cerca de várias centenas de milhares de clientes registrados e atendem quase um milhão de consumidores por ano.

Motivando a atividade prática

Motivando a atividade prática

- ▶ Considere uma empresa bem estabelecida que opera no setor de varejo de alimentos. Atualmente, eles têm cerca de várias centenas de milhares de clientes registrados e atendem quase um milhão de consumidores por ano.
- ▶ Eles vendem produtos de 5 categorias principais: vinhos, produtos de carne rara, frutas exóticas, peixes especialmente preparados e produtos doces. Esses produtos podem ser ainda divididos em produtos gold e produtos regulares. Os clientes podem fazer pedidos e adquirir produtos por meio de 3 canais de venda: lojas físicas, catálogos e o site da empresa.

Motivando a atividade prática

Motivando a atividade prática

- ▶ Considere uma empresa bem estabelecida que opera no setor de varejo de alimentos. Atualmente, eles têm cerca de várias centenas de milhares de clientes registrados e atendem quase um milhão de consumidores por ano.
- ▶ Eles vendem produtos de 5 categorias principais: vinhos, produtos de carne rara, frutas exóticas, peixes especialmente preparados e produtos doces. Esses produtos podem ser ainda divididos em produtos gold e produtos regulares. Os clientes podem fazer pedidos e adquirir produtos por meio de 3 canais de venda: lojas físicas, catálogos e o site da empresa.
- ▶ Globalmente, a empresa teve receitas sólidas e uma saúde financeira sólida nos últimos 3 anos, mas as perspectivas de crescimento dos lucros para os próximos 3 anos não são promissoras... Por essa razão, várias iniciativas estratégicas estão sendo consideradas para reverter essa situação. Uma delas é melhorar o desempenho das atividades de marketing, com um foco especial em campanhas de marketing.

Motivando a atividade prática

Motivando a atividade prática

- ▶ Considere uma empresa bem estabelecida que opera no setor de varejo de alimentos. Atualmente, eles têm cerca de várias centenas de milhares de clientes registrados e atendem quase um milhão de consumidores por ano.
- ▶ Eles vendem produtos de 5 categorias principais: vinhos, produtos de carne rara, frutas exóticas, peixes especialmente preparados e produtos doces. Esses produtos podem ser ainda divididos em produtos gold e produtos regulares. Os clientes podem fazer pedidos e adquirir produtos por meio de 3 canais de venda: lojas físicas, catálogos e o site da empresa.
- ▶ Globalmente, a empresa teve receitas sólidas e uma saúde financeira sólida nos últimos 3 anos, mas as perspectivas de crescimento dos lucros para os próximos 3 anos não são promissoras... Por essa razão, várias iniciativas estratégicas estão sendo consideradas para reverter essa situação. Uma delas é melhorar o desempenho das atividades de marketing, com um foco especial em campanhas de marketing.

Visualização de dados

Momento expositivo II

Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

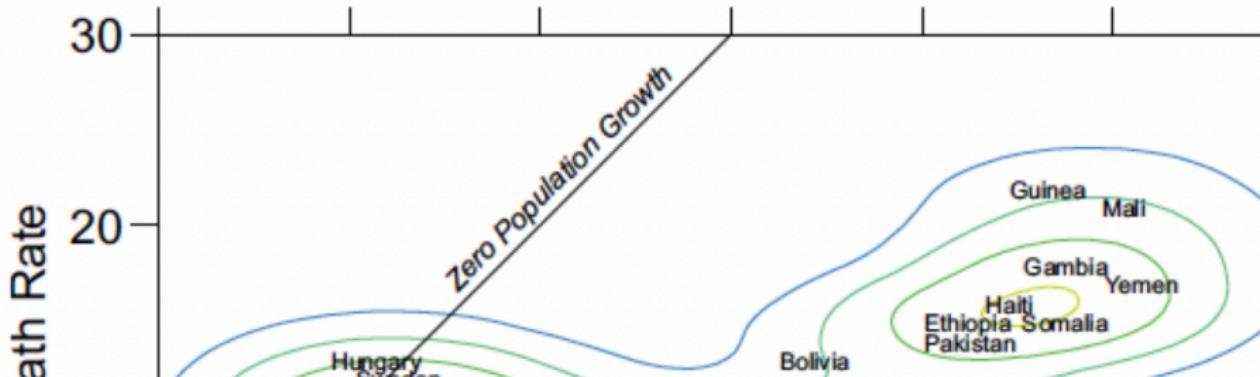
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot

ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

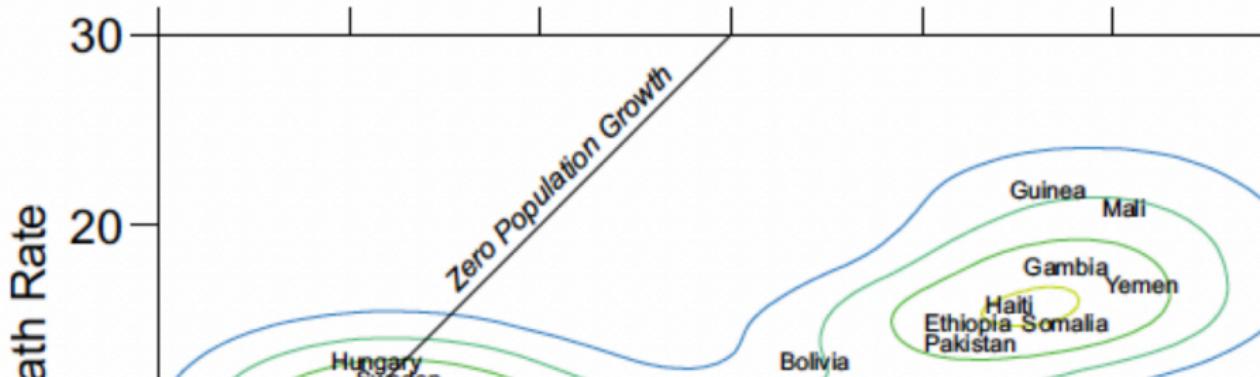
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),`
`color.hue()))`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

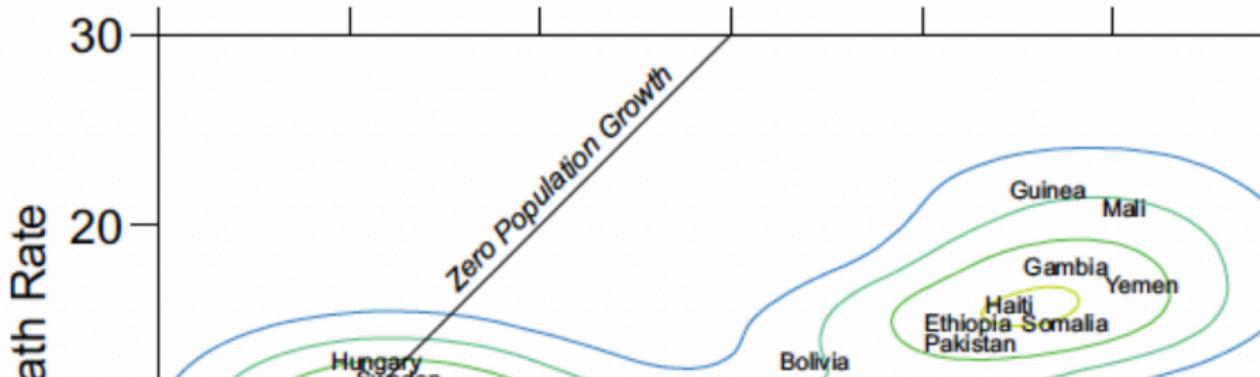
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

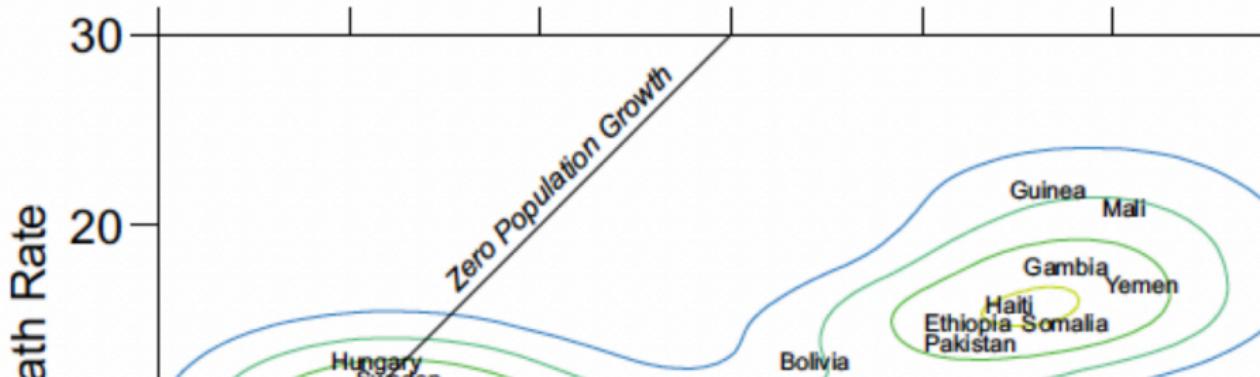
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

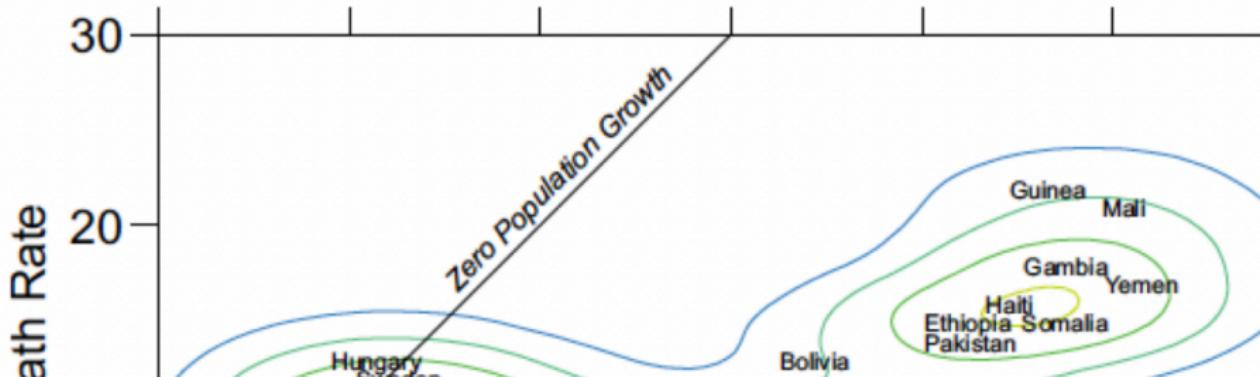
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

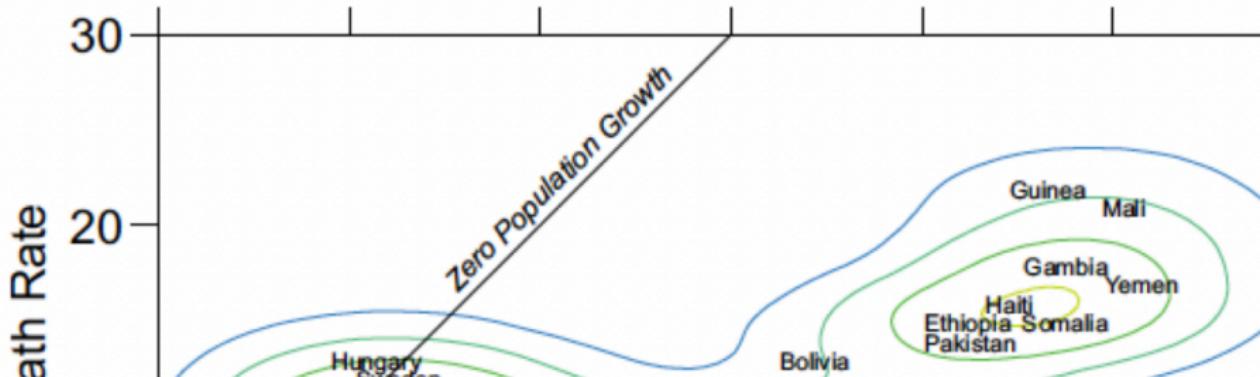
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

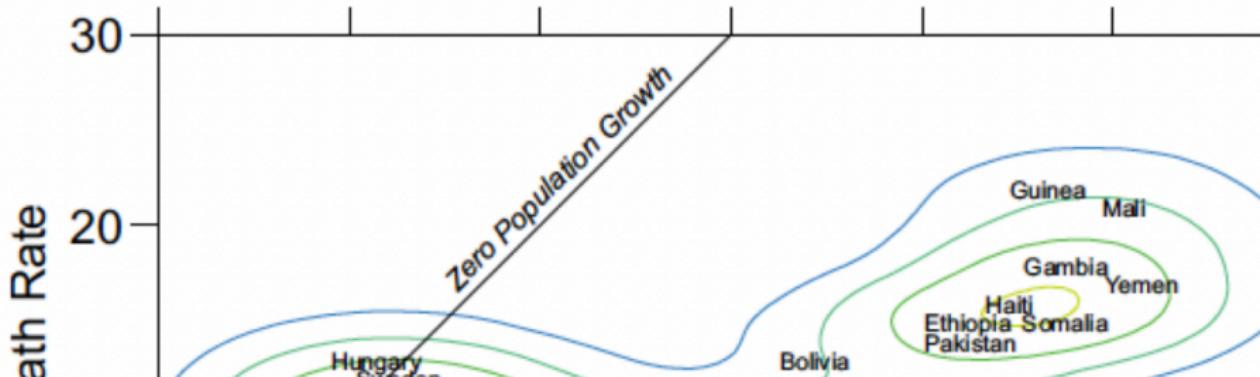
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

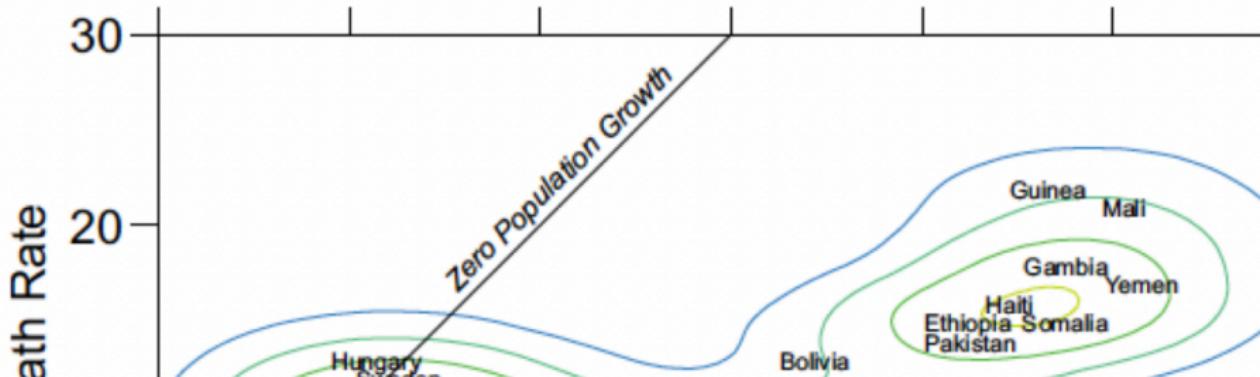
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

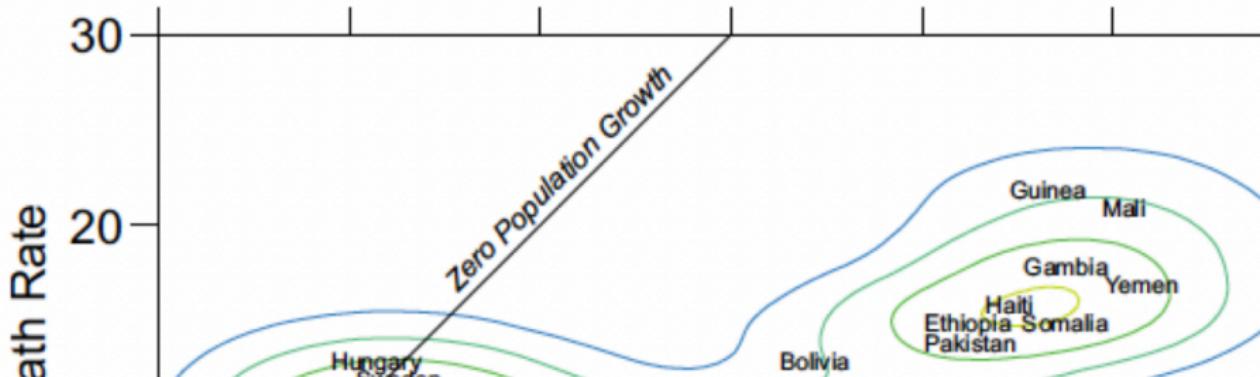
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

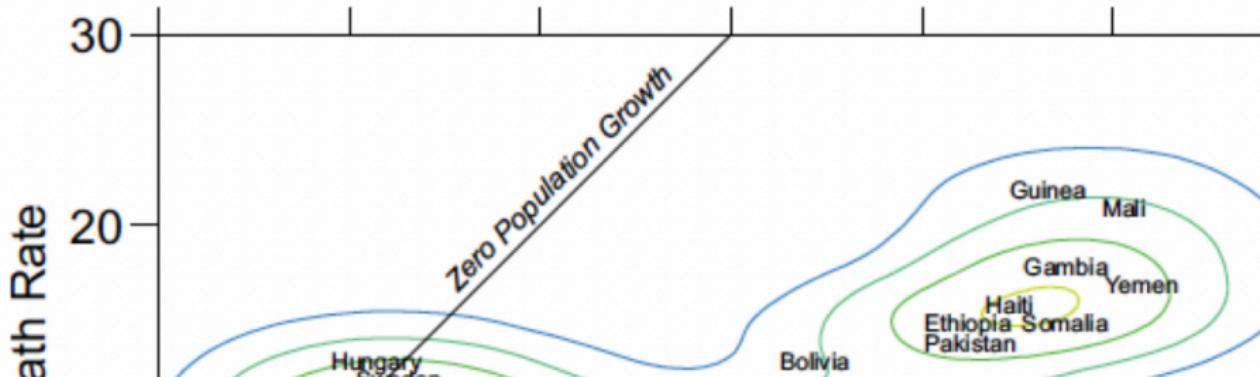
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

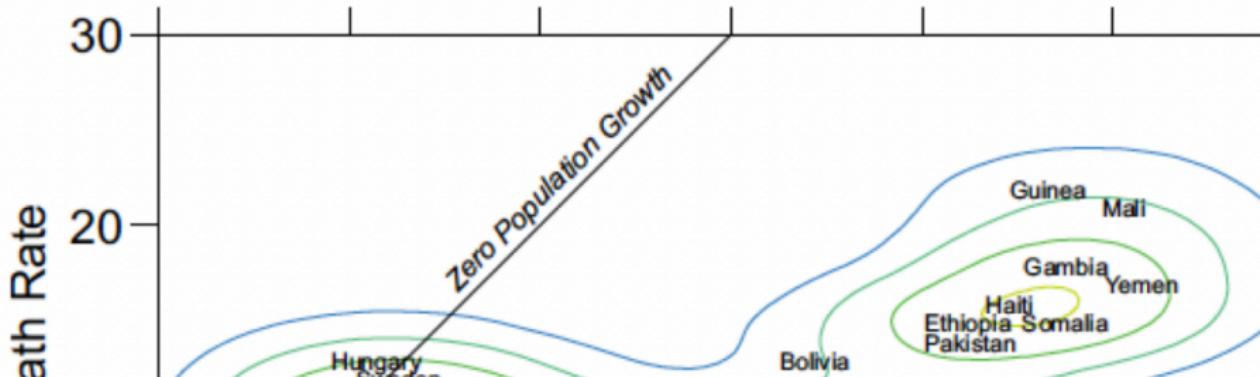
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados com ggplot ggplot

ggplot2 é um sistema para criação de gráficos declarativamente, baseado em The Grammar of Graphics. Esse livro descreve uma gramática que permite sair de declarações para gráficos.

ELEMENT: `point(position(birth*death), size(0), label(country))`

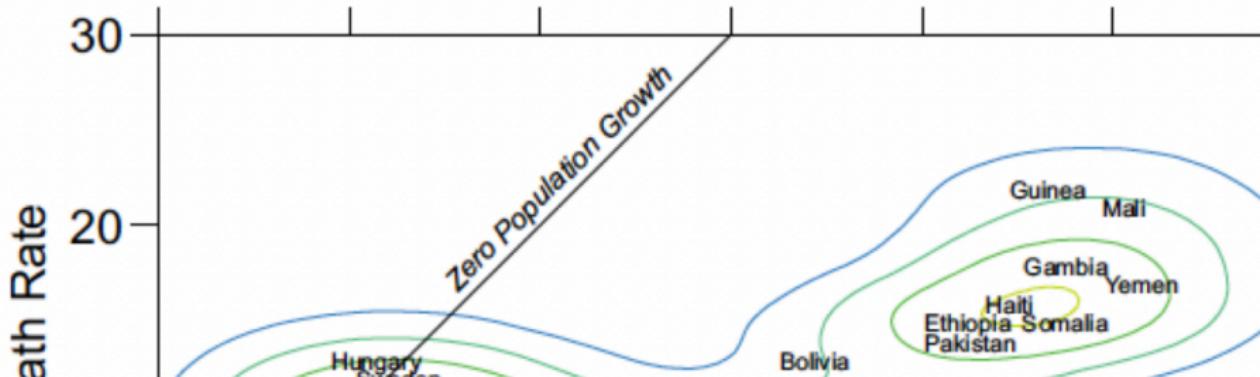
ELEMENT: `contour(position(`

`smooth.density.kernel.epanechnikov.joint(birth*death)),
color.hue())`

GUIDE: `form.line(position((0,0),(30,30)), label("Zero Population Growth"))`

GUIDE: `axis(dim(1), label("Birth Rate"))`

GUIDE: `axis(dim(2), label("Death Rate"))`



Visualização de dados

Momento prático II

Motivando a atividade prática

Após seu sucesso em suas primeiras tarefas, a empresa pediu a você parar gerar algumas visualizações para ela.

5ª Tarefa

Com o objetivo de focar melhor na publicidade da próxima campanha, a equipe de marketing deseja saber qual a distribuição das vendas por canal de venda do público-alvo da campanha (clientes que tem uma criança ou um adolescente em casa). Eles desejam ver essa informação em um gráfico de barra.

6ª Tarefa

O time de marketing deseja visualizar a relação entre a renda dos clientes e o total gasto em compras. Eles desejam ver essa informação em um gráfico de dispersão.

Como aprender mais?

Como aprender mais?

- ▶ Curso de R via swirl
- ▶ Cursos de Data Science Foundations with R no Coursera
- ▶ Livro online R for Data Science
- ▶ Cursando as disciplinas do curso de Ciência de Dados

Fim!

Obrigado pela participação!