

## Exercício 2

```
library(tidyverse)

mob_sp <- read_csv(here::here("sp_mobilidade_apple_240420b.csv"),
  col_names = TRUE)

str(mob_sp) # Mostrar a estrutura do conjunto de dados
```

1

Queremos  
Ver a  
Estrutura do  
Conjunto dos  
Dados e um  
Resumo Dele

### Estrutura

- `str()` [Base R]
- `glimpse()` [tidyverse: dentro do pacote `tibble`]
- formato diferente que a `str()`

### Resumo

- `summary()` [Base R]
- `summarytools::dfSummary()` e `summarytools::descr()` [tidyverse]
- `descr` fornece informações sobre os variáveis numéricas; ignora as outras
- `Hmisc::describe()` [Base R]

2

### Exercício 3

Use todas essas ferramentas para ver a estrutura e o resumo dos dados de São Paulo

O que é a média e desvio padrão (*standard deviation* ou *sd*) do índice para São Paulo?

Qual dessas ferramentas você prefere? (A escolha é a sua!)

3

### Exercício 3

```
install.packages("Hmisc", "summarytools")
library(tidyverse, Hmisc, summarytools)

# Estrutura
str(mob_sp)
glimpse(mob_sp)

# Resumos
summary(mob_sp)
summarytools::dfSummary(mob_sp)
summarytools::descr(mob_sp)
Hmisc::describe(mob_sp)
```

4

## Focar no Modalidade de transito - Subsets

- Uma medida de quanto pessoas estão se expondo aos outras e aumentando chances de infecção é a taxa de utilização de transito
- Pessoas estão usando transito mais com o passar do tempo, burlando as regras de distanciamento social?
- Precisamos criar um subconjunto (*subset*) de `mob_sp` para ver a tendência com tempo

5

## Subset - 2 Modalidades

- `filter()` - limitar os casos para aqueles que tem a variável `mode = "transit"` [tidyverse]
- `$` e `[]` anotation [base R]

6

## filter()

- Um dos “verbos” de dplyr, o pacote que trata de manipulação dos dados
  - VSS Veja o *cheatsheet* “Data Transformation”
- `filter()` escolhe as fileiras (casos) que atendam aos critérios lógicos
- Exemplo: `filter(mob_sp, mode == "transit")` selecionará só os casos em que a pessoa levou transito público

7

## Subsets em Base R

- Todos os dataframes, tibbles, etc. tem a estrutura de fileiras e colunas (variáveis) - Parecido com matrizes
- formato: `dados[<fileira>, <coluna>]`
- exemplo: caso 20 e coluna 4 (index) de mobsp

```
mob_sp[20, 4]
## # A tibble: 1 x 1
##   index
##   <dbl>
## 1  113.
```

8

## Pode Pôr Intervalos na Especificação das Fileiras ou Colunas

- Primeiro 5 variáveis do caso 20
- Primeiro 5 casos da variável index

```
mob_sp[20, 1:5]
## # A tibble: 1 x 5
##   region    mode    date      index month
##   <chr>    <chr>  <date>    <dbl> <chr>
## 1 Sao Paulo driving 2020-02-01  113. Feb

mob_sp[1:5, "index"]
## # A tibble: 5 x 1
##   index
##   <dbl>
## 1  100
## 2  104.
## 3  105.
## 4  104.
## 5  110.
```

9

## Simplificação com \$

- Pode anotar as variáveis com \$
- `mob_sp$index` retornará todos os valores da variável index
- Se quisermos só o caso 20, pode combinar os dois métodos: `mob_sp$index[20]`

```
## [1] 15.30 13.64 19.45 16.23
##    23.38 22.44
```

10

## O Pipe

- Função comum entre as linguagens, especialmente vindo da base UNIX
- Símbolo em R diferente daquele da UNIX - R (`%>%`) - UNIX (`|`) - Símbolo de UNIX tem outra função em R
- Definição - Aplique o resultado da operação do lado esquerdo a função do lado direito como 1º argumento
- Permite que fazemos uma cadeia de ações sem precisar criar novos dataframes sem necessidade - Também aumenta a legibilidade dos comandos, funções e scripts
- Uso: `a %>% b` (faça `a`; depois faça `b` usando o resultado de `a` como 1º argumento)

11

## Voltar ao Problema de transit na base de dados

- Agora queremos usar `filter()` e o *pipe* para permite que fazemos um resumo de `transit`

12

## Exercício 4 (último do dia)

- Já mostrei como fazer o `filter()`:  
`filter(mob_sp, mode == "transit")`
- Pegar o resumo de transit com  
`summarytools::descr()`

13

## Exercício 4

```
library(tidyverse)
library(summarytools)
mob_sp %>%
  filter(mode == "transit") %>%
  summarytools::descr()
```

14

## Aula 2

- Expandir *subsetting*
- Visualizações dos dados
- Limpeza dos dados
- Probabilidade e distribuições dos dados

15

---

# Aula 2 - Slides

JAMES HUNTER, PH.D.

<https://rstudio.cloud/project/1181159>

12 de maio de 2020

16



## Temas de Hoje

- Continuar com Análise de Dados Exploratória
- Visualização dos Dados
- Limpeza dos Dados

17

## Onde Terminamos Aula 1 – Subsetting

- Tidyverse vs. Base R
- Verbo de tidyverse  
`dplyr::filter(pais == "Brasil")`
- Anotações de Base R:  
`[<fileiras>, <colunas>]` ou `$`

18

## Análise Exploratória

- Uso de números de resumo estatístico para entender os dados que tem
- Tendências nos números
- *Missing Data* (dados faltando), marcados NA
- Dados com valores errados ou improváveis
- Dados com valores extremos

19

## Análise Exploratória - 2

- Visualizações das variáveis
- Univariadas – uma variável por vez mostrando distribuição
- Multivariadas – relação entre 2 variáveis
- Agrupadas – Valores por nível do grupo - e.g., mode nos dados de mobilidade

20

Como  
Usarmos a  
Análise  
Exploratória

- Tomar decisões sobre quais tipos de limpeza de dados precisamos fazer
- Ver se as unidades e a escala dos dados são adequadas para sua análise
- Tem suficientes valores em todos os subgrupos para ter uma análise útil

21

Como  
Usarmos a  
Análise  
Exploratória - 2

- Decidir quais tipos de análise são possíveis com os dados no formato que você tem
- Paramétricos ou não paramétricos
- Dominar nossos dados
- Entendimento/Compreensão
- Processo muito iterativo

22

1º Dataset  
desta Aula –  
`dplyr::starwars`

- As personagens dos filmes
- Precisa muita limpeza
- Vamos usar as variáveis demográficas
  - `name:species [1:10]`

23

## Visão dos Dados Abrangente

---

- Visão de todos os dados de uma vez
- 1º: olhe na estrutura dos dados com `str()`
- 2º: Olhe nos dados em si
  - `summarytools::dfSummary(x = <df>, graph.col = FALSE)`
- Mostra todas as variáveis/colunas em uma serie das tabelas
- Coluna que mostra um gráfico dos valores difícil interpretar; pode omitir

24

## Exercício 5

- Para quem vai usar RStudio Cloud: <https://rstudio.cloud/project/1181159>
- Passo 1: Carregar os pacotes necessários
- tidyverse, summarytools
- Passo 2: Colocar `starwars` na memória ativa com o nome `sw`
- Pode usar **base R** ou **tidyverse** para limitar as variáveis aquelas que queremos
- Passo 3: Olhar na estrutura dos dados
- Quais são as classes das variáveis?
- `str()`
- Passo 4: Use `dfSummary()` para ver o que são os dados

25

## O Que Diz a `str()`

```
## tibble [87 x 10] (S3: tbl_df/tbl/data.frame)
## $ name      : chr [1:87] "Luke Skywalker" "C-3PO" "R2-D2" "Darth Vader" ...
## $ height    : int [1:87] 172 167 96 202 150 178 165 97 183 182 ...
## $ mass      : num [1:87] 77 75 32 136 49 120 75 32 84 77 ...
## $ hair_color: chr [1:87] "blond" NA NA "none" ...
## $ skin_color: chr [1:87] "fair" "gold" "white, blue" "white" ...
## $ eye_color : chr [1:87] "blue" "yellow" "red" "yellow" ...
## $ birth_year: num [1:87] 19 112 33 41.9 19 52 47 NA 24 57 ...
## $ gender    : chr [1:87] "male" NA NA "male" ...
## $ homeworld : chr [1:87] "Tatooine" "Tatooine" "Naboo" "Tatooine" ...
## $ species   : chr [1:87] "Human" "Droid" "Droid" "Human" ...
```

26

## Dados Categóricos

- Variáveis de classe `chr` e `fctr` são normalmente categóricos
- Usamos elas para agrupar nossos dados
- Criar subsets

27

## Resultado de `dfSummary` para `gender` – Variável `chr`

```
## Data Frame Summary
## sw
## Dimensions: 87 x 1
## Duplicates: 82
##
## -----
## No   Variable      Stats / Values      Freqs (% of Valid)  Missing
## ----
## 1    gender        1. female           19 (22.6%)          3
##      [character]  2. hermaphrodite    1 ( 1.2%)           (3.45%)
##      3. male       62 (73.8%)
##      4. none       2 ( 2.4%)
## -----
```

28

## Resultado de dfSummary para gender

- gender é uma variável categórica
- Exemplo: queremos saber quantos pesam tem os personagens por gênero
- Criaremos grupos para os gêneros diferentes
- Podemos mudar o tipo da variável para um *factor*
- Maneira mais eficiente de guardar em memória
- Esta tela também conta quantos dados missing esta variável tem
- Depois podemos decidir como lidar com os valores faltando

29

## Resultado de dfSummary para mass – Variável num

```
## Data Frame Summary
## sw
## Dimensions: 87 x 1
## Duplicates: 48
##
## -----
## No  Variable  Stats / Values  Freqs (% of Valid)  Missing
## -----
## 1    mass      Mean (sd) : 97.3 (169.5)  38 distinct values  28
##      [numeric] min < med < max:      (32.18%)
##      15 < 79 < 1358
##      IQR (CV) : 28.9 (1.7)
## -----
```

- Aqui temos várias medidas sobre a tendência central e distribuição da variável

30

Exercício 6 - O  
Que Esse  
Distribuição  
Quer Dizer?

- O que é a história da *mass* para as personagens de Star Wars?
- O que devemos fazer com o valor super alto (1358)?
- Este número de missings invalida o uso da variável *mass*?

31

Exercício 6 - *O que é a história da **mass** para as personagens de Star Wars?*

- *Mass* inclui personagens de *species* muito diferentes
- Têm características diferentes
- Precisa respeitar isso
- Tratar eles em grupos de *species*
- O que devemos fazer com o valor super alto (1358)?
- Este número de missings invalida o uso da variável *mass*?

32



## Exercício 6 - O que devemos fazer com o valor super alto (1358)?

### – O velho nojento Jabba o Hutt

```
sw %>%
  filter(mass == max(mass, na.rm = TRUE)) %>%
  select(name, mass, species)

## # A tibble: 1 x 3
##   name                mass species
##   <chr>              <dbl> <chr>
## 1 Jabba Desilijic Tiure 1358 Hutt
```

- Deixe ele fora do data frame? Grupos de species?
- Este número de *missings* invalida o uso da variável `mass`?
- Uma questão de juízo. Com certeza, em publicações baseadas nos dados, precisa explicitar essa porcentagem (32.2%)

33

## Exploração dos Dados em Mais Detalhe

- Univariada - Resumos das variáveis
  - `summarytools::descr()`
  - `Hmisc::describe()`
- Multivariada
  - Pode começar de perguntar coisas sobre os dados
  - e.g.: O que é a diferença em mass para os gêneros diferentes?
  - Usar `filter()`, `select()` e `group_by()` para organizar os subsets
  - Usar `summarytools::descr()` para mostrar o resultado
  - Juntar eles com o *pipe*

34

```
sw >>>
group_by(gender) %>% # dividir dados em 4 grupos
summarytools::descr(mass)

## Descriptive Statistics
## mass by gender
## Data Frame: sw
## N: 19
##
##
```

	female	hermaphrodite	male	none	NA
Mean	54.02	1358.00	81.00	140.00	46.33
Std.Dev	8.37	NA	28.22	NA	24.83
Min	45.00	1358.00	15.00	140.00	32.00
Q1	49.00	1358.00	76.00	140.00	32.00
Median	52.50	1358.00	80.00	140.00	32.00
Q3	56.20	1358.00	87.50	140.00	75.00
Max	75.00	1358.00	159.00	140.00	75.00
MAD	5.34	0.00	8.15	0.00	0.00
IQR	6.65	0.00	10.75	0.00	21.50
CV	0.15	NA	0.35	NA	0.54
Skewness	1.37	NA	0.03	NA	0.38
SE.Skewness	0.69	0.00	0.36	0.00	1.22
Kurtosis	1.13	NA	1.15	NA	-2.33
N.Valid	10.00	1.00	44.00	1.00	3.00
Pct.Valid	52.63	100.00	70.97	50.00	100.00

35

Exemplo: Cor  
de Cabelo

– Para variáveis categóricas, use  
freq() invés de descr()

36

```

sw %>%
  summarytools::freq(hair_color)
## Frequencies
## sw$hair_color
## Type: Character
##
##      Freq  % Valid  % Valid Cum.  % Total  % Total Cum.
## -----
##      auburn      1    1.22        1.22    1.15    1.15
##      auburn, grey  1    1.22        2.44    1.15    2.30
##      auburn, white  1    1.22        3.66    1.15    3.45
##      black     13   15.85       19.51   14.94   18.39
##      blond      3    3.66       23.17    3.45   21.84
##      blonde     1    1.22       24.39    1.15   22.99
##      brown     18   21.95       46.34   20.69   43.68
##      brown, grey  1    1.22       47.56    1.15   44.83
##      grey       1    1.22       48.78    1.15   45.98
##      none      37   45.12       93.90   42.53   88.51
##      unknown    1    1.22       95.12    1.15   89.66
##      white      4    4.88      100.00    4.60   94.25
##      <NA>       5    5.75      100.00    5.75  100.00
##      Total     87  100.00      100.00  100.00  100.00

```

37

### Exercício 7 - Gênero x Altura

- Usando `group_by()` e `summarytools`, mostre o que é o resumo das alturas por os gêneros

38

```
sw %>%
  group_by(gender) %>% # dividir dados em 4 grupos
  summarytools::descr(height)
```

Descriptive Statistics  
height by gender  
Data Frame: sw  
N: 19

	female	hermaphrodite	male	none	NA
Mean	165.47	175.00	179.24	200.00	120.00
Std.Dev	23.03	NA	35.39	NA	40.71
Min	96.00	175.00	66.00	200.00	96.00
Q1	163.00	175.00	173.00	200.00	96.00
Median	166.00	175.00	183.00	200.00	97.00
Q3	178.00	175.00	193.00	200.00	167.00
Max	213.00	175.00	264.00	200.00	167.00
MAD	13.34	0.00	14.83	0.00	1.48
IQR	15.00	0.00	19.00	0.00	35.50
CV	0.14	NA	0.20	NA	0.34
Skewness	-1.08	NA	-1.21	NA	0.38
SE.Skewness	0.55	0.00	0.31	0.00	1.22
Kurtosis	3.00	NA	2.43	NA	-2.33
N.Valid	17.00	1.00	59.00	1.00	3.00
Pct.Valid	89.47	100.00	95.16	50.00	100.00

39

Visualização –  
2º  
Componente  
da Análise  
Exploratória

- Gráficos: ferramentas excelentes para identificar tendências e coisas estranhas nos dados
- Ajuda a gente ver se os dados estão cumprindo expectativas

40

## John Tukey sobre Visualização

*The simple graph has brought more information to the data analyst's mind than any other device.*

O gráfico simples trouxe mais informações à mente do analista dos dados do que qualquer outro dispositivo.

41

## Dados para Visualização

- Conjunto dos dados dentro de R: mpg
- Dados de economia de combustível para 38 modelos diferentes de carro
- Pergunta: O tamanho de motor tem relação com economia nas rodovias
- Tamanho de motor: `displ`
- Economia nas rodovias: `hwy`
- O que seria sua expectativa sobre a relação entre essas variáveis?
  - `hwy` vai aumentar ou diminuir com o aumento no tamanho dos motores?

42

### Variáveis relevantes

- `hwy` medida em *miles per gallon*
- `displ` medida em litros
- `cyl` número de cilindros

43

### Construir um Gráfico com a “Gramática de Gráficos”

- Base do pacote `ggplot2`
- Um sistema racional
- Assemblar elementos de um gráfico para fazer um inteiro
- Elementos:
  - Linhas
  - Eixos
  - Cores
  - Títulos e etiquetas
  - E muitos outros

44

## Formato de `ggplot`

```
ggplot(data = <data> , mapping = aes(x = <var>, y = <var>)) + geom_xxx(mapping = aes(<aes para geom>))
```

- `aes` = estética
  - os elementos principais do gráfico

45

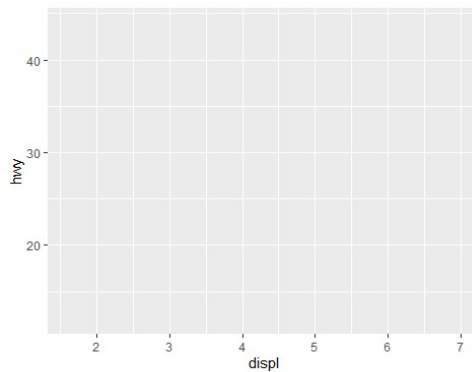
Aplicado ao  
Nosso  
Exemplo

- `data = mpg`      \* conjunto dos dados
- `x = displ`      \* eixo x - tamanho do motor
- `y = hwy`      \* eixo y - economia estradas

46

## O Gráfico Até Agora

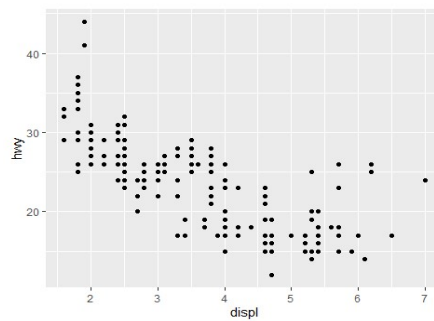
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy))
```



47

## Fazer Dele Um DotPlot

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point()
```



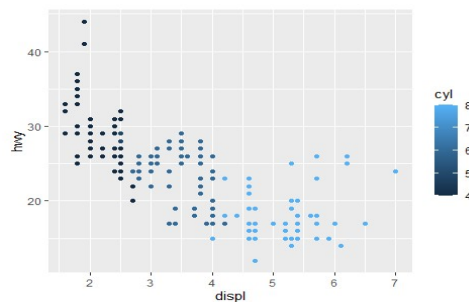
48



## Quero Ver o Efeito do Número de Cilindros na Economia

- Fazer o número de cilindros em uma escala de cores

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, colour = cyl)) + geom_point()
```



49

E Voilà,  
Um  
Gráfico  
Útil

50

## Exercício 8

- Fazer um gráfico do efeito de `displ` sobre a economia nas cidades (`cty`)
- Mostar no mesmo gráfico a influência de tipo do sistema de tração (`drv`)
  - 4 = tração de 4 rodas
  - f = frente
  - r = traseira

51

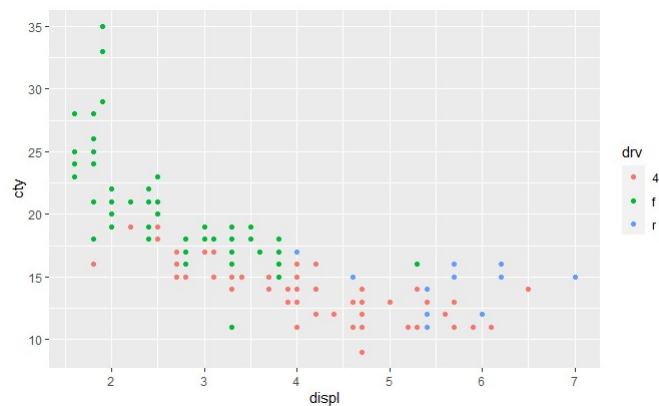
## Exercício 8 - Código

---

```
ggplot(data = mpg, mapping = aes(x = displ, y = cty,  
  colour = drv)) +  
  geom_point()
```

52

## Exercício 8



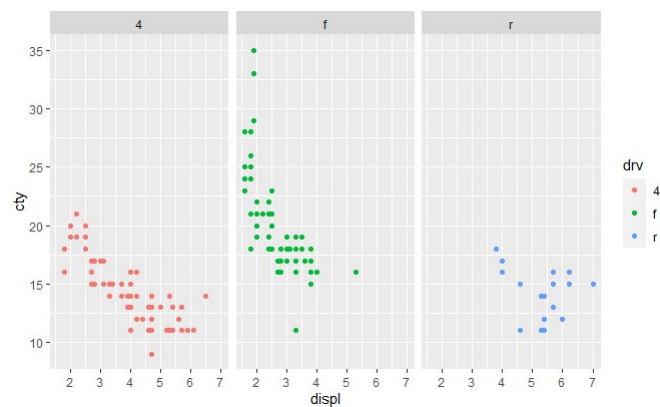
53

Separar  
Gráficos em  
Paneis  
Diferentes

- Nosso gráfico acima é grande; pode pôr cada tipo de veículo no seu próprio painel
- Em ggplot, chamado `facet`
- Forma mais simples é `facet_wrap()`
- Muito flexível com várias opções sobre as escalas das variáveis

54

```
ggplot(data = mpg, mapping = aes(x = displ, y = cty,  
  colour = drv)) +  
  geom_point() +  
  facet_wrap(facets = "drv")
```

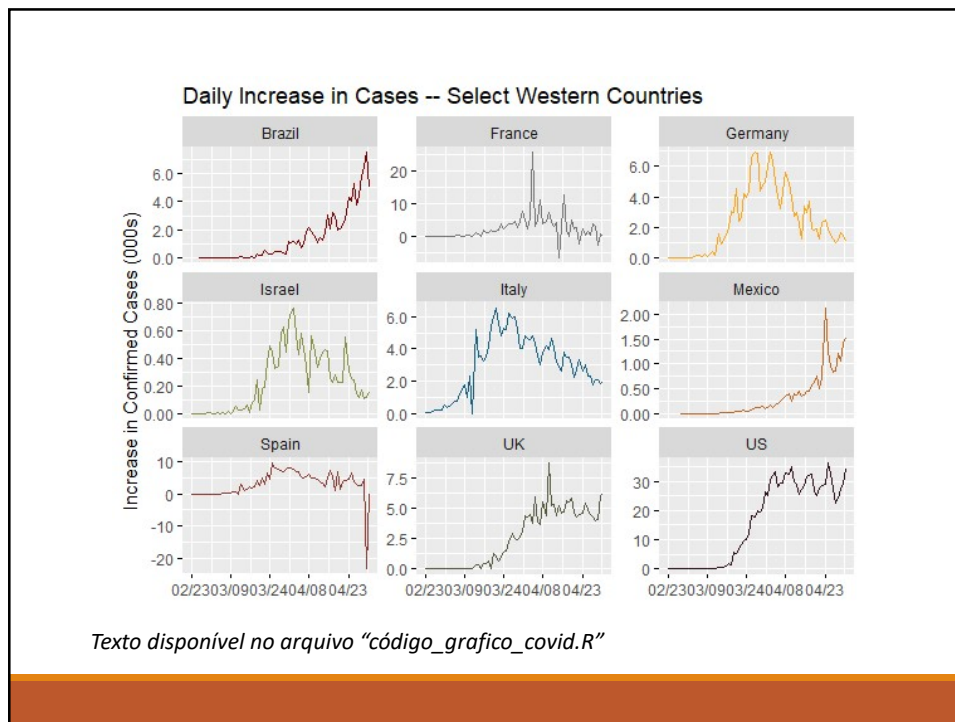


55

Outro Exemplo  
de Facets

– Vem de meu trabalho sobre  
COVID-19

56



57

## Licão de Casa 1

- Este é para vocês
- Não faz parte da nota
- Repita o último gráfico, mas mexe com os seguintes elementos:
- Acrescentar um título e títulos dos eixos
- Mude o caráter dos pontos
- Acrescentar uma curva *smoothing* que mostra a tendência dos dados
- **VSS:** `geom_smooth()`
- Use o ggplot cheatsheet (site de RStudio) como uma guia

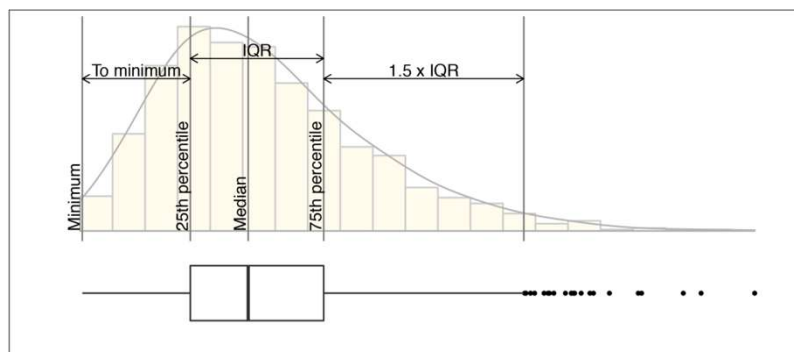
58

## Mais Um Tipo de Gráfico Importante -- boxplot

- Mostra claramente a distribuição dos valores de uma variável
- Baseado no resumo de 5 números
- Ajuda entender os outliers -- valores longe da maioria que
  - Podem ser valores extremos
  - Podem ser erros de entrada de dados
- Essencial para entender como cada variável pode contribuir à análise

59

## Estrutura de Boxplot



Fonte: W. Chang, R Graphics Cookbook, 2nd Ed., 2020, <https://r-graphics.org/>

60

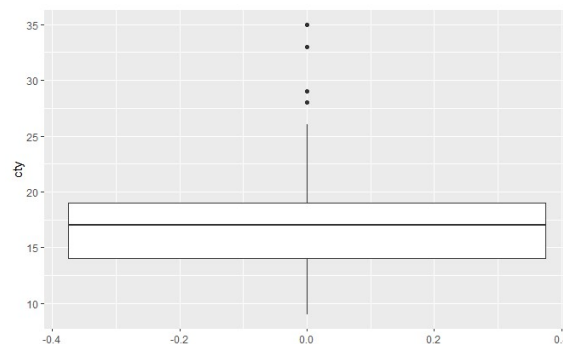
## Construindo um Boxplot

- Queremos ver como `cty` (economia na cidade) está distribuída
- Usamos o mesmo processo que usamos com os dotplots
- Só mudamos o tipo do "geometria" para boxplot

61

## Boxplot Básico

```
ggplot(data = mpg, mapping = aes(y = cty)) +  
  geom_boxplot()
```



- Para esta versão simples, não especifica um eixo X

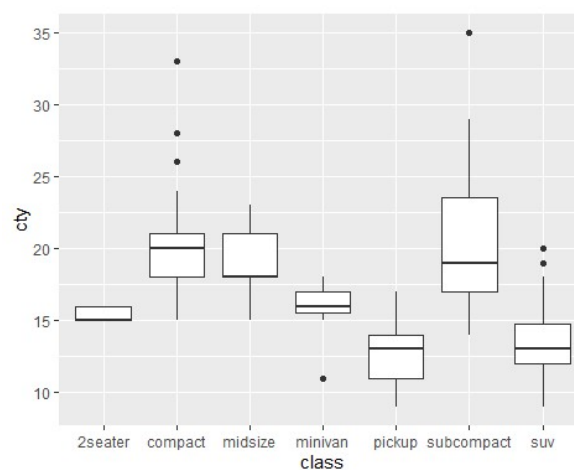
62

Pode mostrar  
diferenças em  
subgrupos dos  
dados

- Aqui especificar um eixo X como a variável que quer usar para agrupamento
- Vamos usar `class` - tipo de veículo

63

```
ggplot(data = mpg, mapping = aes(x = class, y = cty)) +  
  geom_boxplot()
```



64



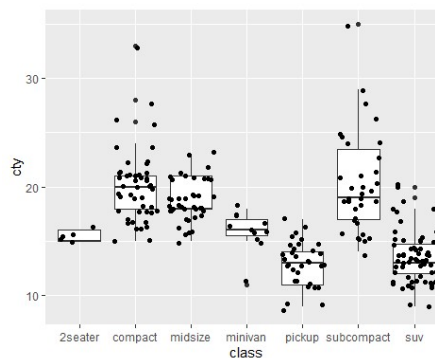
## Boxplot com Ainda Mais Informação

- Quero ver como os pontos individuais dos dados
- Pode fazer isso com uma nova camada `geom_jitter()`
- *jitter* quer dizer mostrar os pontos mas não colocá-los um acima de outro
- Seria como 6 boxplots

65

## Boxplot + Jitter

```
ggplot(data = mpg, mapping = aes(x = class, y = cty)) +  
  geom_boxplot() +  
  geom_jitter()
```



66

## Lição de Casa 2

- Faça um boxplot de `Petal.Length` para todos os 3 espécies (`Species`) de iris no conjunto de dados `iris`
- **VSS:** `iris` faz parte de base R
- Mostre no boxplot a distribuição dos pontos de `Petal.Length`

67

## Tidy Data

- Dados **tidy** são prontos para analisar
- 3 Regras definam tidy data
  1. Todas as colunas devem ser variáveis
  2. Todas as linhas devem ser casos
  3. Cada “celula” deve conter 1 e somente 1 dado

68

Hadley  
Wickham  
sobre Tidy x  
Outros Dados

*Tidy datasets are all alike, but every  
messy dataset is messy in its own way.*

Conjuntos de dados *tidy* parecem  
iguais, mas todo conjunto de dados  
confuso é confuso na sua maneira.

69

## Os 10 Mandamentos das Bases de Dados Bem Formatadas

70

1. Todos seus dados caberá em um dataframe/tibble único

- Não coloque os dados em dataframes múltiplos.
- Este é um habito que se associe com *relational database management*
- Não R e não *tidyverse*

71

2. Você respeitará um estilo de formatação certo

- A tabela deve ser preenchido sem brechas
- Começar na célula para cima a esquerda e descendo sem deixar linhas em branca.

72

### 3. Uma Linha

– Só tem um caso único

73

### 4. Uma Coluna

– Só tem uma variável

74

5. Você não  
codificará  
variáveis  
qualitativas

- Se você tem variáveis para gênero, use
  - homem/mulher,
  - não 1/2

75

5. Você não  
codificará  
variáveis  
qualitativas

- Se você tem variáveis para gênero, use
  - homem/mulher,
  - não 1/2

76

5a. Você não  
codificará  
variáveis com  
cores -- JAMAIS

- R não pode entender cores
- É um artifício preferido dos Excelistas

77

6. A base de  
dados deve  
conter  
somente  
dados

- Formatação artística cria dificuldades na importação dos dados de Excel para R
- Siga o princípio KISS.

78

7.  
Consistente,  
você sempre  
estará

- Não misture nomes diferentes para a mesma coisa
- Sempre homem
- Não uma vez homem, uma vez rapaz ou masculino

79

8. Você  
sempre  
respeitará a  
qualidade  
numérica das  
variáveis  
numéricas

- Exemplo: idade sempre estará um número ('25')
- Não texto e número ('25 anos')
- Datas sempre na forma aceita internacionalmente YYYY-MM-DD

80



9. Proteja o anonimato de seus pacientes/respondentes/clientes

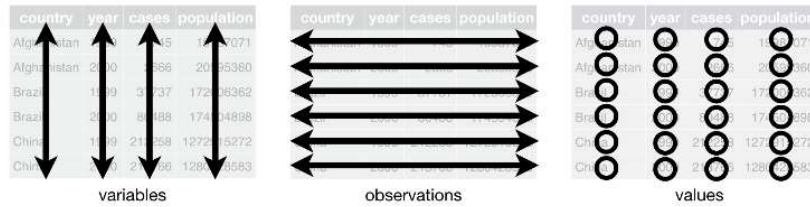
- Use números ou outros identificadores para as pessoas
- Guardar a correspondência num lugar seguro

81

10. Utilize nomes de variáveis compreensíveis pelos seres humanos

- Algum dia no futuro, você ou um outro vai querer entender o que quer dizer o nome de variável `G6`
- Você não vai lembrar que refere a "unidades produzidas em mês 6"

82



Dados São Tidy Quando ...

83

Processo  
de Limpar  
Dados

84

## 1. Fazer os nomes das variáveis “tidy”

- `janitor::clean_names` existe para isso.
- Quando você importar dados em R, deve ser primeira função de limpeza que você usa

85

## 2. Tipos dos dados corretos?

- Números corretamente formados?
- Caracteres vs Fatores (próximo slide)
- Variáveis lógicas tem TRUE e FALSE correto?

86

### 3. Missing Data – NA – Como tratar?

- Tirar o case do conjunto
- Trocar o valor com um valor resumido como a média ou mediana da variável
- Aplicar um algoritmo avançado para criar um valor que não perturbará o resto dos valores e os valores dos resumos
  - **MICE** Multivariate Imputation by Chained Equations

87

### 4. Outliers

- Transformações dos valores
- Erros ou valores corretos?

88

## 5. Verificação Final

- O conjunto obedece as 3 regras de tidy data?
- Os 10 Mandamentos?

89

Para 2  
Últimas  
Aulas

90

### Projeto Individual

- Procure um conjunto de dados que lhe interesse –
  - Pode ser de seu trabalho ou outra parte da sua vida
  - **VSS**: Boa fonte: pacote e site **Gapminder** sobre indicadores sociais e econômicos
  - Tente de trabalhar com 20 até 100 casos e um máximo de 5 variáveis
- Aplique essas técnicas de visualização, exploração dos dados e limpeza dos dados para preparar o conjunto para análise
- Próximas sessões: aprender como tirar conclusões do conjunto. Vai terminar o projeto depois dessas sessões

91

### Projeto em Grupo

- Formar um grupo de um máximo 4 pessoas
- Me avise por email quem fica no grupo
- Identifique um conjunto de dados - Com mais de 100 casos e quantas variáveis que quiser
- Explorar e limpar os dados antes das próximas sessões

92

Ficar  
Atualizado  
durante o  
Intervalo

- Leia o capítulo sobre Recursos na Apostila
  - Siga algumas sessões dos cursos
  - Leia os blogs (esp., R-Bloggers)
  - Download e usar alguns dos livros
- Novo recurso: (Curso gratis) Introduction to R
  - <https://www.quantargo.com/courses/course-r-introduction/>

93

Até o Final  
do Mês!

94