
Aula 1 - Slides

JAMES HUNTER, PH.D.

11 de maio de 2020

1

Como Começar o
Curso nesses Tempos
de Pandemia

2

Quem Sou Eu?

- Professor, Escola Paulista de Medicina (UNIFESP)
 - Virologia/Bioinformática
 - Trabalhando agora com COVID-19
- Aulas em Sustentare desde 2009
- Contato com o Professor
 - email: jameshunterbr@gmail.com
 - Twitter: @jimhunterbr
 - cel/WhatsApp: 11-9-5327-5656

3

Materiais Preparatórios

- Instalação de R e RStudio nos seus computadores
- RStudio Cloud
- Apostila/capítulos com material sobre o curso
- Capítulo 02_Recurso
 - Livros, cursos no internet, sites, cheatsheets

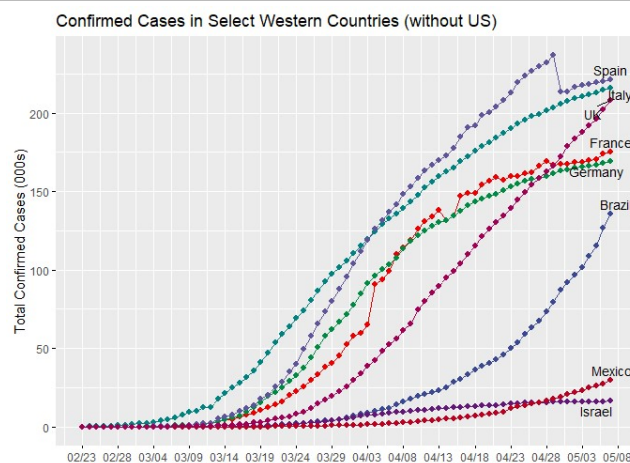
4

VSS: Use os Recursos

- Navarro, **Learning statistics with R: A tutorial for psychology students and other beginners** (<https://learningstatisticswithr.com/book/>)
- Wickham & Grolemund, **R for Data Science** (<http://r4ds.had.co.nz> ou O'Reilly)
- Rstudio Cheatsheets
- **R Bloggers** (<https://www.r-bloggers.com/>)

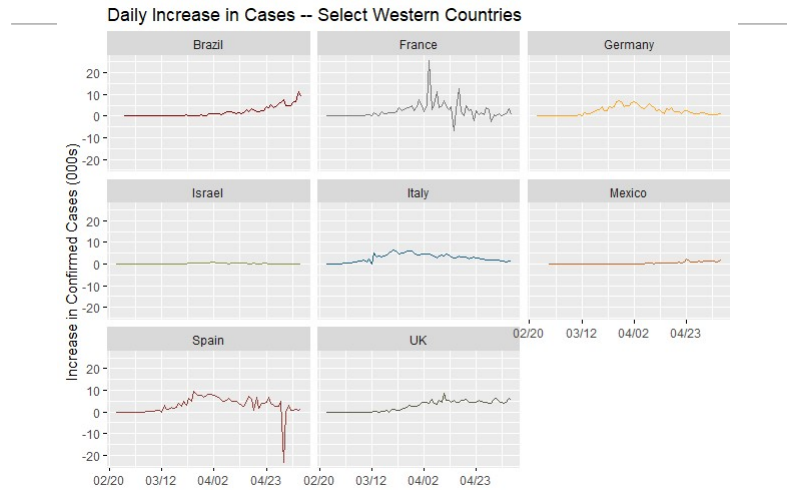
5

Onde Estamos com Covid-19



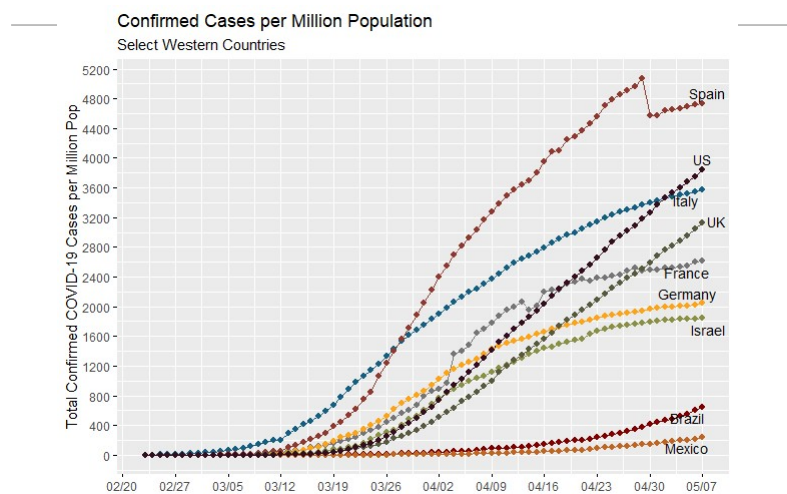
6

Onde Estamos com Covid-19



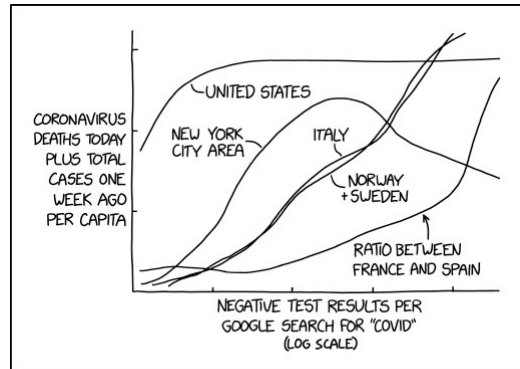
7

Onde Estamos com Covid-19



8

Proliferação dos Gráficos sobre COVID-19



I'M A HUGE FAN OF WEIRD GRAPHS, BUT EVEN I ADMIT SOME OF THESE CORONAVIRUS CHARTS ARE LESS THAN HELPFUL.

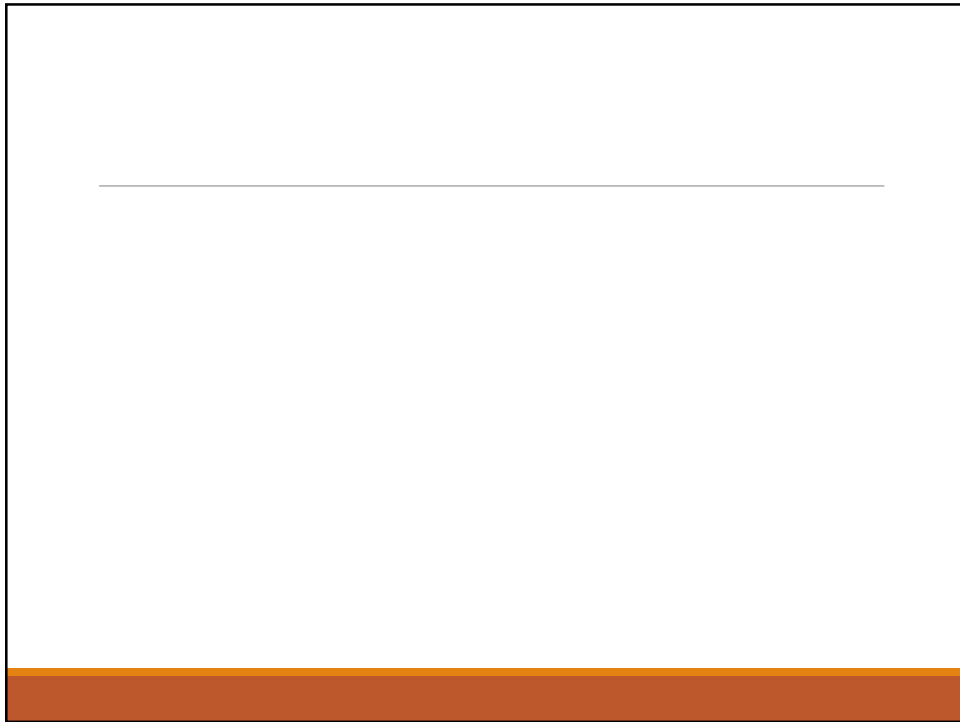
Fonte: Randall Munroe, "Coronavirus Charts", <https://xkcd.com/2294/>

9

Material de Motivação

- COVID-19: o assunto do ano
- Os gráficos analíticos mostrando espalhamento da doença
- Minhas conclusões até agora
- [São só conclusões preliminares]
- Lave suas mãos com sabonete – muito
- Contato público espalha o vírus
- Reabertura rápido vai adoecer muitos
- Vai ter uma segunda onda na primavera

10



11

RStudio Cloud

- Mesmo que o R no seu laptop
- Cadastre-se em <https://rstudio.cloud>
- 4 projetos - um para cada aula
- Faremos exercícios lá

12

RStudio Cloud Projetos

Aula	Project
ADcR Aula 1	https://rstudio.cloud/project/1177204
ADcR Aula 2	https://rstudio.cloud/project/1181159
ADcR Aula 3	https://rstudio.cloud/project/1181172
ADcR Aula 4	https://rstudio.cloud/project/1181165

Localização das Aulas no RStudio Cloud

13

Tarefas do Curso

1. Entrance Quiz
2. Análise Individual - Escolher um dataset (ou da empresa ou dos datasets públicos) - Organizar o dataset - Mostrar tabelas e gráficos para me familiarizar com seus dados - Fazer um modelo de uma relação entre variáveis no dataset (teste de inferência ou modelo de regressão)
3. Projeto em Grupo - Escolher um dataset mais complexo - Organizar - Usar regressão para modelar algum aspecto interessante dos dados - **VSS** Deve ter suficiente trabalho para cada pessoa no grupo pode ter uma participação significativa.

14

Submissão dos Trabalhos

- Os projetos individuais e em grupo devem ser submetidos em formato “pdf” baseados nos arquivos “RMarkdown” (.rmd).
- Se você submete só o arquivo .rmd, vai receber a nota 0 para esse trabalho. PRECISA fazer o knitr, não deixa para eu fazer.

15

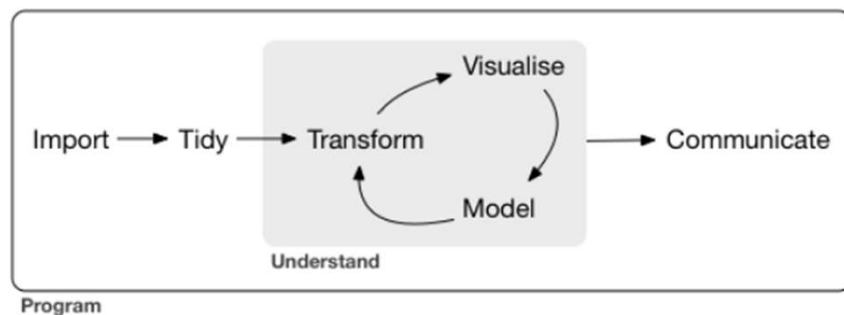
Fluxo de Atividades em Data Science

16

Importância de Planejamento do *Workflow*

- Iniciantes em análise de dados (e muitos experientes) se perdem porque não planejam a sequência das atividades
- Fazem análises improvistas
 - Usando técnicas sem entender
 - As premissas do teste ou modelo
 - Se os resultados são apropriados para a tarefa
 - Onde no computador são os arquivos necessários para a análise
- Melhor
- Planejar o seu projeto do início
- Desenhar uma estrutura das pastas, projetos, e arquivos para guardar seus dados e scripts
- Quando vai escrever seu relatório ou apresentação, vai saber como todas as partes da análise encaixam para criar um total.

17



Fluxo de Trabalho de Hadley Wickham (R for Data Science)

18

3 Fases de Análise

- *Import-Tidy* - Trazer os dados para R e limpar eles
- *Transform-Visualize-Model* - Um processo iterativo em que você põr os dados na forma que o modelo que você quer usar precisa até você tem um resultado adequado
- *Communicate* - relatório, gráficos, apresentações que informa seu público dos resultados e como chegou neles

19

Qual Fase Precisa Qual Proporção do Trabalho?

Fase	Imaginado	Real
Import-Tidy	0.2	0.7
Model	0.6	0.2
Communicate	0.2	0.1

Proporção do Trabalho num Projeto de Análise

20

Como Pôr Um Workflow em Operação

21

De Onde Vêm Os Dados

- Excel (foco aqui)
 - .xls ou .xlsx
 - .csv
- Arquivos de texto
 - .fasta
 - .txt
- Outros formatos

22

Preparando Dados em Excel

- Objetivo é preparar os dados para análise
- **Não** é fazer um relatório graficamente bonito
- KISS
- Lembre que computadores não podem ler toda a formatação
 - Fontes
 - Cores
 - Programas como R (ou Python) odeiam ver linhas em branca

23

Algumas Regras para Uso de Excel

1. Se trabalha com arquivos .csv
 - Guardar dados diferentes em arquivos diferentes
2. Se trabalha com arquivos .xlsx
 - Guardar dados diferentes em abas diferentes
3. Dados devem estar no formato de um bloco
 - Sem linhas brancas
 - Primeira fileira: nomes das variáveis
4. Cada coluna só deve ter uma classe de dados
 - numérica, caráter, lógica, ...
5. Zeros são 0, nunca "-", " " (espaço) ou outro formato
6. Dados faltando são sempre NA, nunca 0, 99 ou outro formato
7. Cada coluna é **variável**
8. Cada fileira é **caso**
9. Nenhuma cor ou desenho

24

Nomes de Arquivos

- Usar somente letras, números e _
 - Não usar espaços entre palavras
 - Não usar acentos
- Fazer os nomes compreensíveis
- Exemplos bons:
 - vendas_regiao_1.csv
 - pac_history.csv
- Exemplos ruins:
 - cv pac 1.csv (espaços)
 - foo.xlsx (???)
 - x_23.csv (???)

25

Iniciar Análise em R - Projetos

26

Projetos

- RStudio oferece capacidade de separar todos seus trabalhos em projetos separados
- Muito útil para você ter dados vindo de 2 matérias e 3 projetos de pesquisa
- Você cria uma pasta em algum lugar de seu disco rígido ou no *cloud*
 - Colocar lá os arquivos de Excel e outros programas, etc.
 - Pode usar isso como o local dos arquivos vai criar como parte da análise
 - Scripts
 - Dados
 - Gráficos
 - Relatórios

27

Algumas consequências de não usar projetos

- Quando você liga R, sempre precisa definir de novo a pasta ativa (*working directory*)
- Como resultado, vai perder arquivos importantes

28

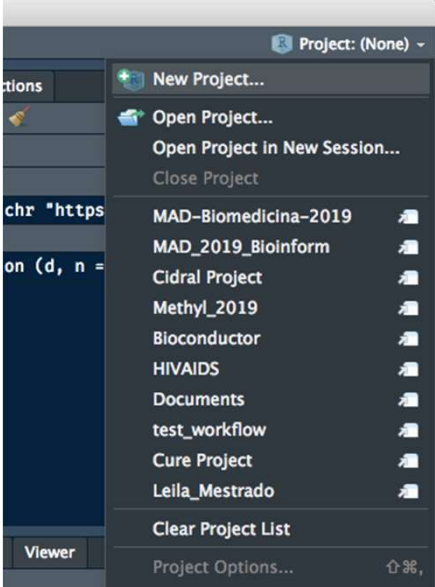
Criar Um Projeto

- No canto da tela a direita para cima, tem um ícone que diz *Project*
- Tem uma pequena seta a direita; clique nela

29



30



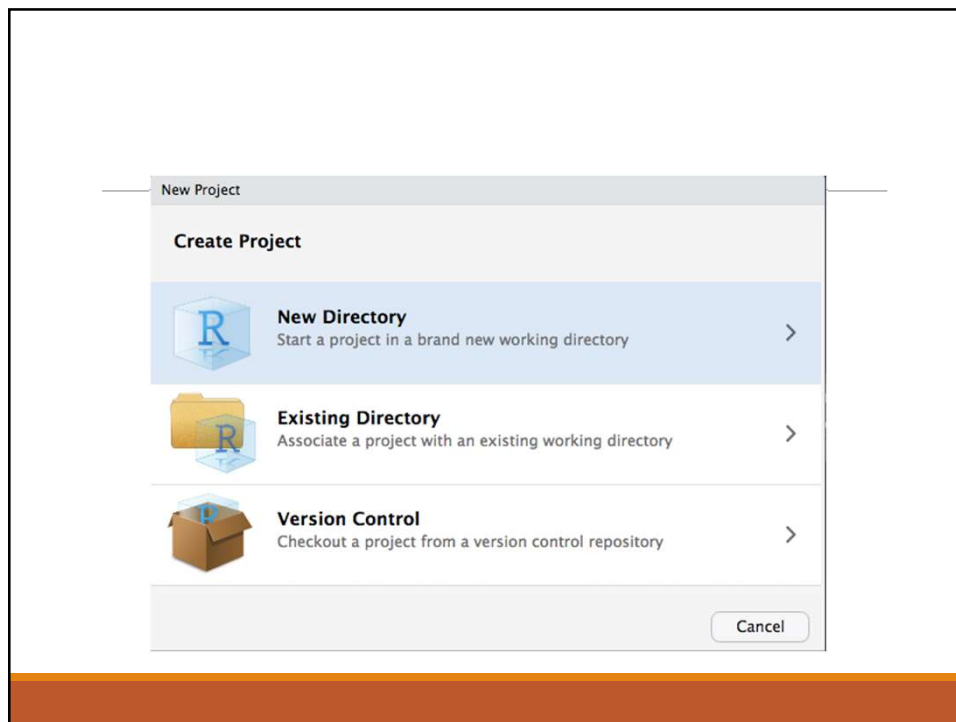
No *drop-down*,
Selecione
New Project

31

Janela de
Projeto Novo

- Você já criou uma pasta para seu projeto
 - 2ª opção *Existing Directory* vai escolher isso
- Por este exemplo, vamos criar uma nova pasta *New Directory*

32

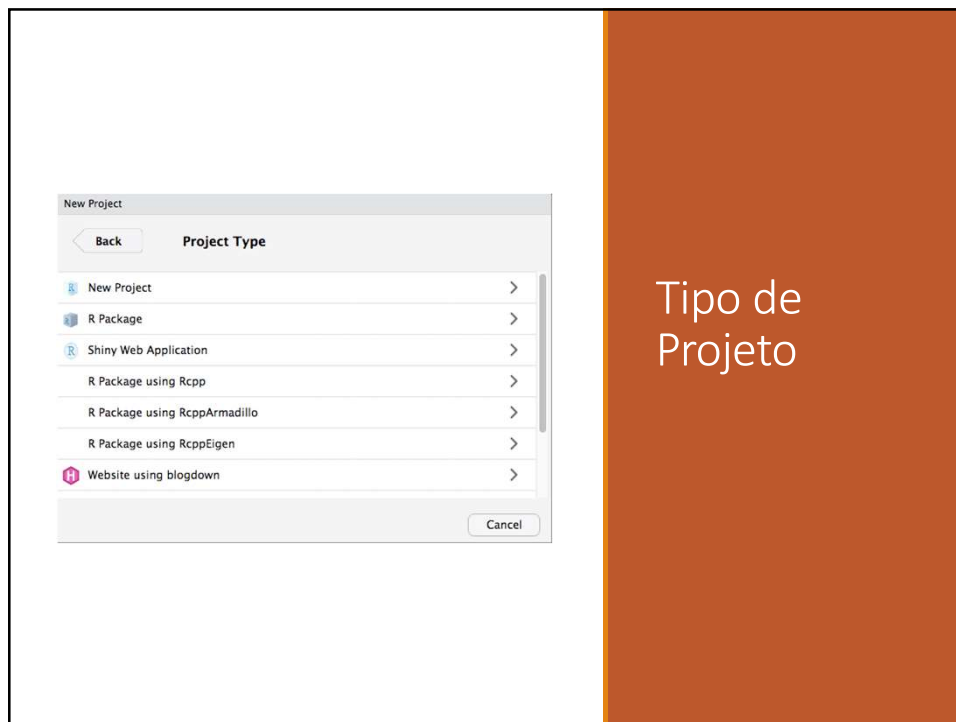


33

Tipos de *New Directory*

- Vários tipos possíveis
- Depende dos pacotes carregados
- Selecione o mais básico - *New Project*

34

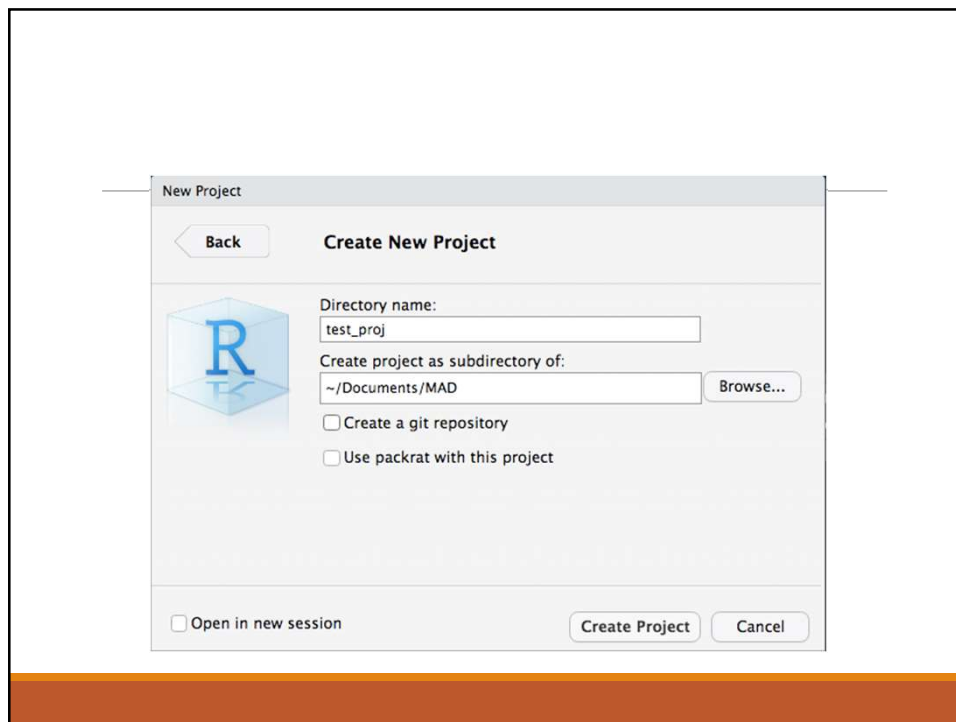


35

Janela de
Create New Project

- Criaremos projeto com nome `test_proj`
- Colocaremos numa nova pasta abaixo de `~/Documents/MAD`
- Deixar as outras caixas vazias

36

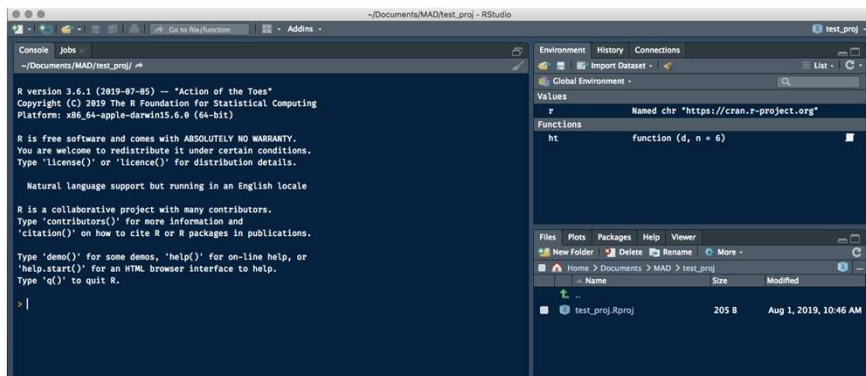


37

Completar a Criação do Projeto

- Clique no *Create Project*

38



Novo Projeto Completo

39

Organizar Sub-Pastas

- Se você ainda não tinha feito antes ...
- Pode criar sub-pastas para organizar os arquivos
- Projetos importantes (TCCs, etc.) tendem ter **MUITOS** arquivos
- Sempre crio uma pasta chamada `data_raw` onde coloco cópias dos dados originais no formato em que recibo eles.
 - Nunca uso esses arquivos, só cópias deles para análise.
 - Assim, sempre tenho uma cópia original que posso usar para referência

40

Onde Estamos, Professor? “We’re HERE”

- Fácil de se perder com as complicações de caminho entre a pasta raiz de seu computador e a pasta de seu projeto
- Nosso projeto no meu computador tem o seguinte *filepath*:
 - `"/Users/jameshunter/Documents/MAD/test_proj"`
- O computador de meu colega Nathalia pode ter o *filepath* seguinte:
 - `"c:._proj"`
- Quando mudo um script de meu computador para aquele da Nathalia, não vai aceitar meu *filepath* de Mac

41

here to the Rescue

- A função `here()` dentro do pacote `here` conta para nos o que é a pasta ativa.
- ```
here::here()
[1]
"C:/Users/james/OneDrive/Documents/
Sustentare/Data_Analysis_R_2020"
```

42

## Como Descubro os Gráficos que Guardei?

- Guardei eles numa sub-pasta chamada graficos
- Posso salvar um *filepath* com a subpasta graficos
- R me avisará o *filepath* desse local
  - Não importa se for no meu computador ou naquele de Nathalia

```
[1]
"C:/Users/james/OneDrive/Documents/
Sustentare/Data_Analysis_R_2020/graficos"
```

43

# R Super-Básico

44

## Console vs. Scripts

- Local de execução dos comandos
- **VSS**: escrever comandos em um script ou *R Markdown*

45

## Comentários

- Num script, utilize muitos comentários explicando o que cada comando faz

```
comentários podem ser escritos dessa forma
usando o hashtag (#)
Não precisa iniciar a linha com o hashtag
x <- 5 + 5 # Qualquer texto depois do hashtag não
será processado
```

46

## Operações Simples - R Como Calculadora

```
5 + 5
[1] 10

36 * 2500000
[1] 9e+07

2^25 # exponenciação
[1] 33554432

25 * (12 + 27) # uso de
parênteses
[1] 975

log10(27587) # função
[1] 4.440704
```

47

## Funções Matemáticas

### Maths Functions

|                           |                                 |                          |                         |
|---------------------------|---------------------------------|--------------------------|-------------------------|
| <code>log(x)</code>       | Natural log.                    | <code>sum(x)</code>      | Sum.                    |
| <code>exp(x)</code>       | Exponential.                    | <code>mean(x)</code>     | Mean.                   |
| <code>max(x)</code>       | Largest element.                | <code>median(x)</code>   | Median.                 |
| <code>min(x)</code>       | Smallest element.               | <code>quantile(x)</code> | Percentage quantiles.   |
| <code>round(x, n)</code>  | Round to n decimal places.      | <code>rank(x)</code>     | Rank of elements.       |
| <code>signif(x, n)</code> | Round to n significant figures. | <code>var(x)</code>      | The variance.           |
| <code>cor(x, y)</code>    | Correlation.                    | <code>sd(x)</code>       | The standard deviation. |

48



### Atribuição/Assignment

- (nome do objeto) <- (definição do objeto)
- definição = os valores que são o conteúdo do objeto.

49

### Estilos de Atribuição

- Funcionam

```
x <- 6
```

```
x <- "olà!"
```
- Funcionam mas não recomendados

```
x = 6
```

```
6 -> x
```
- Produz erro (não pode iniciar um comando com um número)

50

```
> 6 = x
Error in 6 = x : invalid (do_set) left-hand side to assignment
> |
```

51

Atribuição/Nomes  
de Variáveis

```
1ª Versão
peso <- 55 ## Pessoa pesa 55
kg.

2ª Versão
peso_kg <- 55 ## Mais claro

Pode Converter à Libra
peso_lb <- peso_kg * 2.2
peso_lb

[1] 121
```

52

## Exercício 1

1. No console, calcule a seguinte operação matemática:

$$\frac{42 * 95^2 + 6}{16 - 3.5}$$

2. Atribua o valor desta operação à variável `calc`
3. Faça arredondamento do resultado a uma casa decimal

53

## Exercício 1 – Resultado

1. Calcule a seguinte operação matemática:

$$\frac{42 * 95^2 + 6}{16 - 3.5}$$

```
((42 * 95^2) + 6)/(16 - 3.5)
```

```
[1] 30324.48
```

2. Atribua o valor desta operação à variável `calc`

```
calc <- ((42 * 95^2) + 6)/(16 - 3.5)
```

3. Faça o arredondamento do resultado para uma casa decimal

```
round(calc, 1)
```

```
[1] 30324.5
```

54

## Tipos (Classes) de Dados em R

- `<int>` *integer* (número inteiro)
- `<dbl>` *double* (número de duplo tamanho, ou seja, um número real)
- `<chr>` *character* (caráter)
- `<dtm>` *date/time* (data com tempo)
- `<date>` *date* (data)
- `<fctr>` *factor* (fator)
- `<lgl>` *logical* (lógico – TRUE/FALSE)

55

## Tipo Lógico de Dados

- Pode ter um dos dois valores
  - TRUE
  - FALSE
- Pode ser uma atribuição ou resultado de um cálculo lógico
- Atribuição

```
TRUE
```

```
[1] TRUE
```

```
FALSE
```

```
[1] FALSE
```

56

## Cálculos Lógicos

- Operadores Lógicos
  - `==` lado esquerdo é igual ao lado direito
  - `!=` lado esquerdo não é igual ao lado direito
  - `>=` lado esquerdo maior ou igual ao lado direito
  - `<` lado esquerdo menor do lado direito
- Exemplos

```
x <- 6 # dar um valor ao x
x == 6 # testar se x é igual a 6

[1] TRUE

2 < 4

[1] TRUE
```

57

## Carregar e Processar um Conjunto dos Dados

58

## Pronto para Próximo Passo no R

- Capítulo sobre motivação – conjunto de dados da mobilidade da Apple
  - Mede índice de mobilidade por *driving*, *transit* e *walking*
  - Relativa a 13 de janeiro
  - Dados de Brasil, São Paulo e Rio de Janeiro
  - Baseado no número de pedidos de direção no app Apple Maps
  - Formato *csv* (do Excel)
- Carregar esta base de dados
  - Primeiro, só os dados de São Paulo
  - Depois, o conjunto inteiro com os dados do Brasil
- Conduzir análises básicas
  - Subconjuntos (*subsets*)
  - Resumos exploratórios dos dados
  - Combinações das operações com o *pipe*

59

## Passo 1 - Carregar o Tidyverse

- Vamos começar de usar comandos deste sistema
  - `readr::read_csv()`
  - Depois outros

60

## Para carregar um pacote qualquer - `library(tidyverse)`

```

-- Attaching packages ----- tidyverse 1.3.0 --
v ggplot2 3.3.0 v purrr 0.3.4
v tibble 3.0.0 v dplyr 0.8.5
v tidyr 1.0.2 v stringr 1.4.0
v readr 1.3.1 v forcats 0.5.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()

```

61

## Carregar os Dados na Memória Ativa

- Para trabalhar com dados, R tem que ter eles na memória ativa
  - Não no disco rígido ou SSD (como SQL ou C fazem)
- Não é um limite em prática
  - Laptops modernos têm espaço suficiente para gigabytes de dados

62

readr::read\_csv()

```
read_csv(file, col_names = TRUE,
col_types = NULL, locale =
default_locale(), na = c("", "NA"),
quoted_na = TRUE, quote = "\"", comment
="", trim_ws = TRUE, skip = 0, n_max =
Inf, guess_max = min(1000, n_max),
progress = show_progress(),
skip_empty_rows = TRUE)
```

- 1º argumento: nome do arquivo (em aspas):  
file = "br\_mobilidade\_apple\_240420.csv"
- 2º argumento: col\_names
  - se a 1ª fileira tem nomes de variáveis, use TRUE
- 3º argumento: col\_types
  - R tenta entender qual tipo de dados cada coluna tem
  - Se você quer especificar, pode

63


Executar o  
Comando

```
mob_sp <-
read_csv(here::here("sp_mobilidade_apple
_240420b.csv"), col_names = TRUE)

Parsed with column specification:
cols(
region = col_character(),
mode = col_character(),
date = col_date(format = ""),
index = col_double(),
month = col_character(),
weekend = col_logical()
)
```

64





E Voilà –  
Dados

65

Exercício 2

Agora, você faz isso ou no seu  
laptop ou no RStudio Cloud

66

## 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
```

67

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]

68

### 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!)

69

### 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)
```

70

## 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

71

## Subset - 2 Modalidades

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

72

## 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

73

## 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.
```

74

## 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.
```

75

## 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
```

76

## 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)

77

## Voltar ao Problema de transit na base de dados

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

78

## Exercício 4 (último do dia)

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

79

## Exercício 4

---

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

80



## Aula 2

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