


Metodologias Informacionais com R

Módulo I: Introdução à Linguagem R

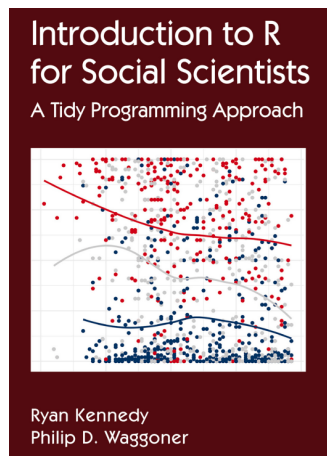
Telmo dos Santos Klipp (telmo.klipp@inpe.br)

Informações Gerais sobre o Curso

- Materiais disponibilizados via **Classroom**;
- O aprendizado requer a prática que será constante nas aulas;
- Será cobrado ao menos uma atividade semanal .

Bibliografia Básica:

- Kennedy, R., & Waggoner, P. D. (2021). Introduction to r for social scientists: a tidy programming approach. CRC Press.



Bibliografia Complementar:

- Wickham, H., Çetinkaya-Rundel, M., & Golemund, G. (2023). R for data science (2e): import, tidy, transform, visualize, and model data. "O'Reilly Media, Inc.". Disponível em: <https://r4ds.hadley.nz/>. Acesso em: 14 de junho, 2023. (Online)
- Damiani, A. et. al., (2022). Ciência de Dados em R. Curso-R. Disponível em: <https://livro.curso-r.com>. Acesso em: 12 de maio, 2023. (Online)
- de Aquino, J. A. (2014). R para cientistas sociais. Editora da UESC (editus). Disponível em: <http://www.uesc.br/editora/>. Acesso em: 12 de maio, 2023.
- de Oliveira, P. F., Guerra, S., McDonnell, R. (2018). Ciência de Dados com R: Introdução. Editora IBPAD. Disponível em: <https://cdr.ibpad.com.br/index.html>. Acesso em: 12 de maio, 2023. (Online)

Nas últimas aulas vimos:

- Representação de dados no R (objetos).
- Persistência de dados na memória.
- Como declarar funções no R.
- Como e porque inserir comentários.

Fundamentos do R – session

Literalmente o ambiente de execução do R. Todos os objetos que criamos, bibliotecas que importamos, comandos que usamos acontecem no contexto de uma *R session*. O RStudio, por sua vez, inicia uma *R session* ao ser executado e passa a interagir com a mesma. Podemos salvar objetos e o histórico de comandos da sessão R em arquivos, conforme:

```
save.image(file = ".RData")      # salva todos os objetos
load(file = ".RData")            # carrega todos os objetos
savehistory(file = ".Rhistory")  # salva o histórico de comandos
loadhistory(file = ".Rhistory")  # carrega o histórico de comandos
```

Como configurar para salvar esses arquivos no RStudio?

No menu principal: *Tools* → *Global Options* (menu *General*, aba *Basic*).

Isso nos mantém organizados e seguros sobre nossos dados e atividades no R?

Fundamentos do R – projeto

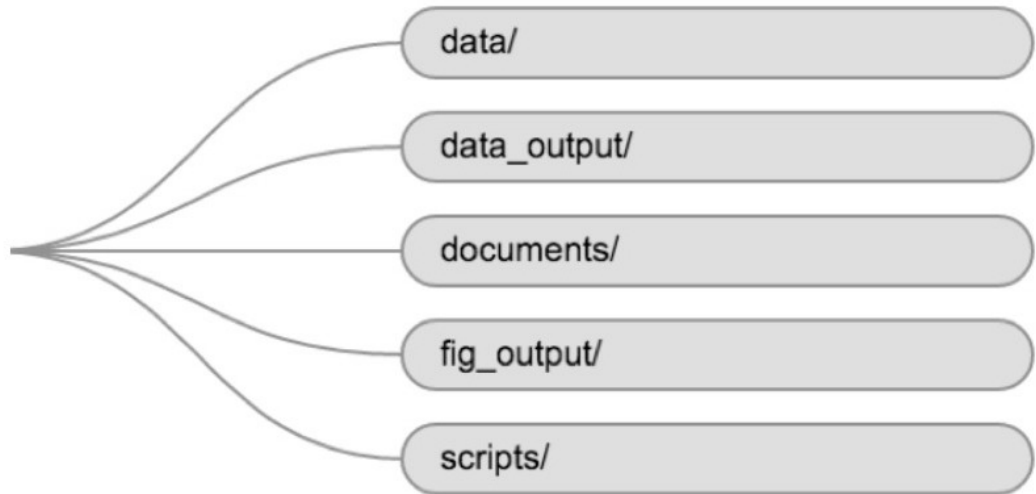
Projetos permitem organização, seja em algum trabalho, seja na vida no geral. No R isso não é diferente. **O que pretendemos organizar?**

- Diferentes tipos de arquivos (ex: .R, .csv, .png ...).
- Diferentes tipos de dados, salvos em arquivos.
- Arquivos especiais do R (ex: .RData e .RHistory ...).
- Outros arquivos com informações relativas ao projeto.

Ganhos?

- Gestão e controle das etapas do projeto.
- Gestão e controle dos recursos do projeto (ex: dados, *R scripts*).
- Produtividade e gerenciamento de tempo.

Fundamentos do R – projeto



Créditos: [Erin Becker \(2023\)](#).

Fundamentos do R – projeto

Por onde começar?

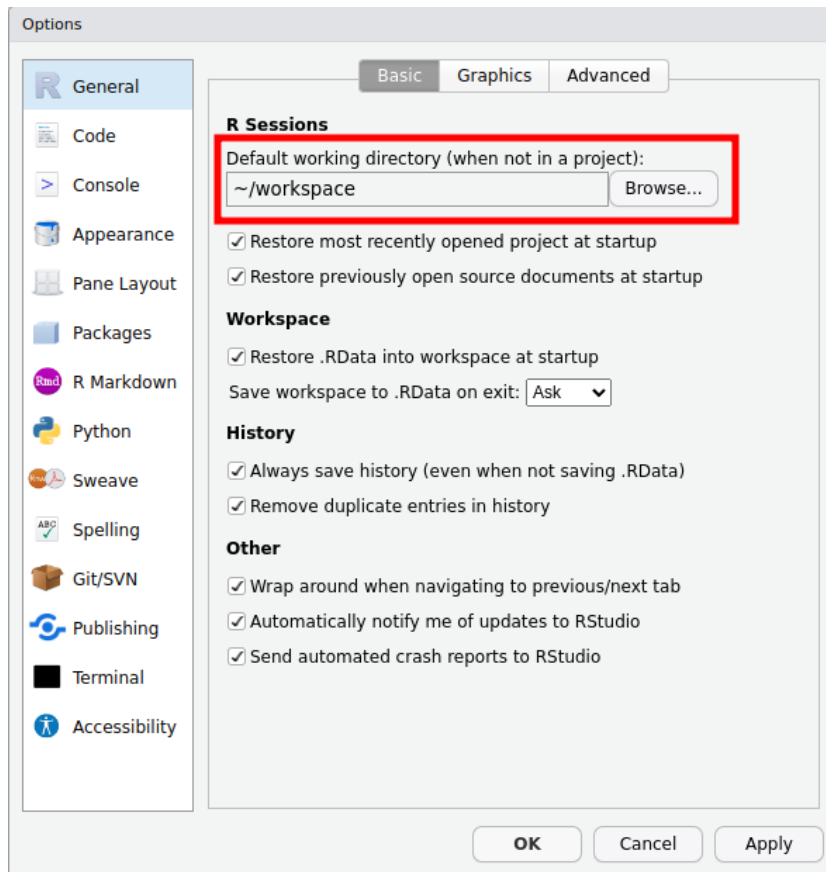
- Escolher um diretório de trabalho/pasta para o projeto.

Diretórios? Subdivisões de um sistema de arquivos que permitem organização hierárquica. Diretórios podem ter:

- Caminhos absolutos
 - C:/Users/username/meus_programas/... ; D:/meus_dados/ ... (Windows)
 - /home/usuario/meus_dados/ ... (linux)
- Caminhos relativos
 - ~/projeto1
 - ~/projeto2
 - .
 - .
 - .
 - ~/projetoN

Fundamentos do R – projeto

O RStudio permite a configuração de um diretório de trabalho padrão.



Fundamentos do R – projeto

Criar um projeto no R é uma boa prática. Para isso é importante entender sobre manipulação e definição do diretório de trabalho. Considere as funções:

```
getwd()      # Retorna (imprime) o diretório de trabalho atual
setwd()      # Muda o diretório atual para o indicado no argumento
dir.create() # Cria um diretório ou apenas uma pasta
list.files() # Lista as pastas e arquivos contidas no diretório de
              # trabalho atual
setwd(choose.dir()) # Escolha por interface gráfica
setwd("meu_diretorio/pasta_do_meu_projeto") # Um exemplo de uso
```

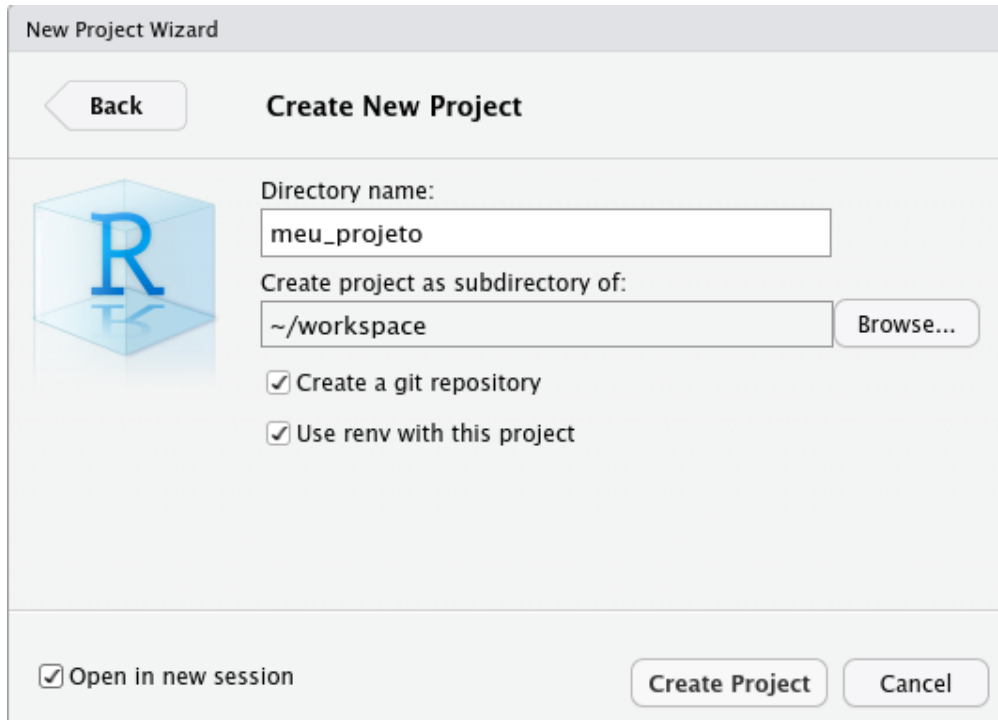
Obs: `setwd("../")` dir. anterior \leftarrow \rightarrow `setwd("./")` dir. atual

Na aba principal do RStudio: Session \rightarrow Set Working Directory \rightarrow Choose Directory (ou Ctrl + Shift + H).

Mãos a obra! Escolha um diretório onde ficaram seus R Project.

Fundamentos do R – projeto

No menu principal do RStudio: File → New Project → New Directory → New Project.



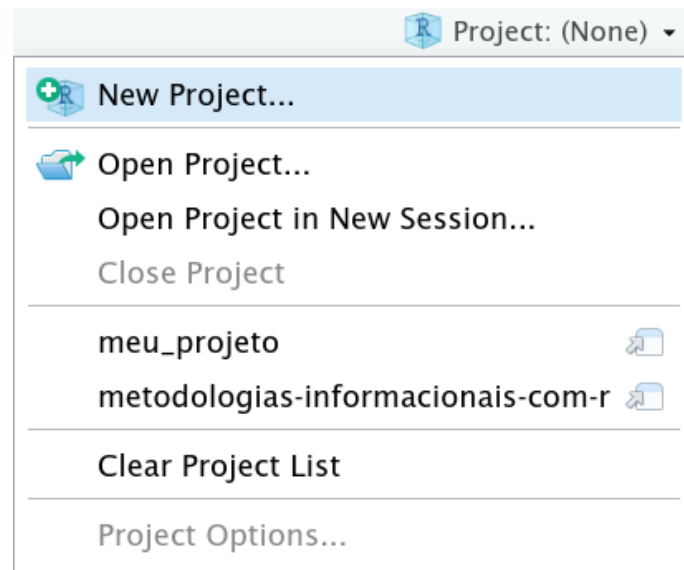
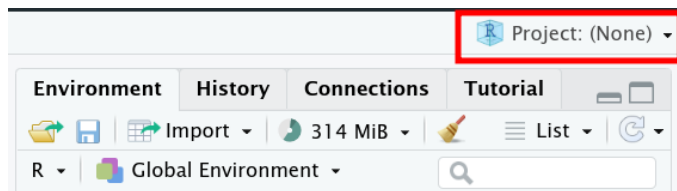
The screenshot shows the 'New Project Wizard' dialog box in RStudio. The title bar reads 'New Project Wizard'. Inside, there is a 'Back' button on the left and a 'Create New Project' title. On the left side of the main area is the R logo. The main area contains the following fields and options:

- Directory name:** A text box containing 'meu_projeto'.
- Create project as subdirectory of:** A text box containing '~/workspace' with a 'Browse...' button to its right.
- ☒ Create a git repository
- ☒ Use renv with this project

At the bottom, there is a checkbox for ☒ Open in new session, and two buttons: 'Create Project' and 'Cancel'.

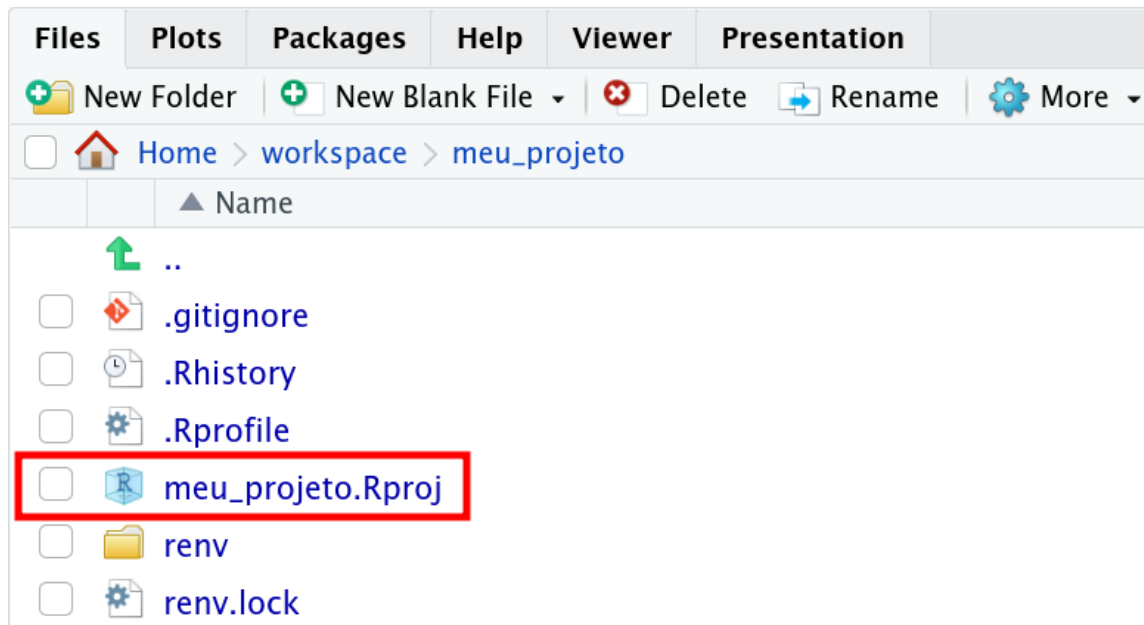
Fundamentos do R – projeto

No menu secundário do RStudio, canto superior direito e acima do painel contendo a aba Ambiente, há um menu para projetos.



Fundamentos do R – projeto

O arquivo principal do R Project tem como extensão `.Rproj`.



Certo! E agora? 

Fundamentos do R – pacotes

Essenciais no R, **pacotes** podem conter **conjunto de outros pacotes, biblioteca de funções e estruturas de dados**. Pacotes permitem customização (ao gosto de cada um), assim como instalar apenas o que é necessário. Pacotes são formas de distribuir soluções (funcionalidades) que outras pessoas produziram.

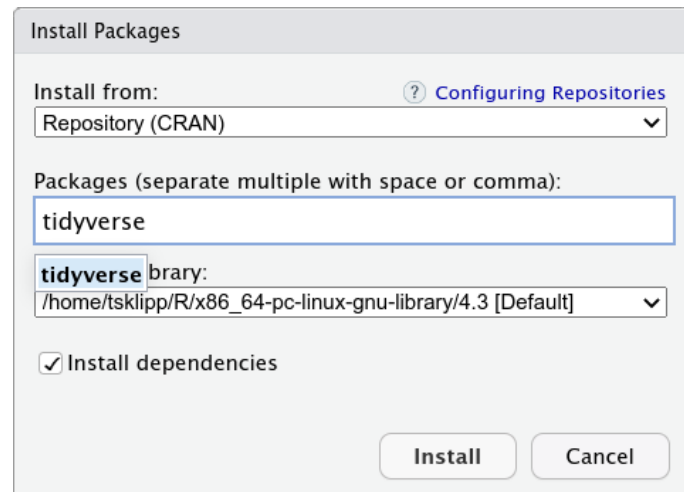
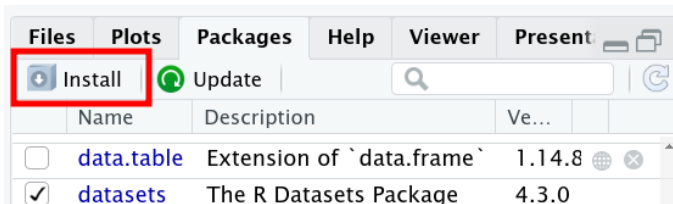
- Existem inúmeros pacotes no repositório oficial **CRAN**.
- Existem inúmeros pacotes em repositórios não oficiais (ex: Github).

Por onde começar? Instalar um pacote desejado segundo alguma opção. É comum proceder por linha de comando (no console ou terminal) usando a função **`install.packages()`**, conforme:

```
install.packages("tidyverse")
```

Fundamentos do R – pacotes

Por onde começar? Na aba Pacotes (*Packages*) do RStudio (ao lado da aba Ajuda) clicar em *Install* e depois procurar/digitar o nome do pacote desejado.



Ao longo do curso, trabalharemos com alguns pacotes, mas vamos instalá-los conforme forem necessários.

Fundamentos do R – pacotes

Após a instalação do pacote é possível carregá-lo na sessão do R atual, conforme:

```
library(tidyverse)           # carrega o pacote
require(tidyverse)          # também carrega o pacote
library(help = tidyverse)    # documentação do pacote
detach("package:tidyverse") # retira o pacote da sessão atual
```

Isso fará com que o conjunto de funções e dados (ou até mesmo outros pacotes) instalados fique disponível para uso.

Em toda nova sessão do R, será necessário carregar o pacote.

A instalação do pacote é realizada apenas uma vez, embora possa ser necessário atualizações eventuais.

Fundamentos do R – pacotes

Os objetos contidos nos pacotes (ex: funções e estruturas de dados) podem ser acessados usando o operador *Double Colon* (`::`), conforme:

```
tidyverse::tidyverse_packages()
```

O uso de `::` representa uma boa prática. **Motivo?**

- Mascaramento de funções;

```
tidyverse::tidyverse_conflicts()
```

- Clareza de código;
- Reprodutibilidade;

Fundamentos do R – pacotes

O RStudio tem várias funcionalidades interessantes. Por exemplo, é possível rodar os exemplos do uso de uma função, disponíveis na aba Ajuda. No entanto, essa funcionalidade requer o pacote **knitr**.

Faça uma consulta usando *help* ao pacote **datasets**, na forma:

```
help(package = "datasets")
```

Procure e clique no conjunto de dados **sunspots** e rode o exemplo apresentado clicando no *link* [Run examples](#).

Muito provavelmente você não terá o pacote **knitr** instalado. Portanto, instale esse pacote e depois tente rodar o exemplo novamente.

Você acaba de instalar um pacote versátil que pode ser muito útil para usuários de R.

Fundamentos do R – pacotes

Pacotes tem ampla documentação, sendo um dos requerimentos aos desenvolvedores para tornar um pacote disponível no repositório oficial. Alguns exemplos:

- Manual do pacote no [CRAN](#).
- *Vignettes* – são documentos *html* mostrando usos específicos de um pacote e suas funções, muitas vezes direcionado a solução de um problema.

```
browseVignettes()
```

- *Cheatsheets* – são documentos com notas curtas e exemplos de uso para consultas rápidas. Ex: [Data Wrangling Cheatsheet](#).

Questões e Prática

- Quais pacotes compõem o tidyverse?
- O tidyverse possui uma lista de pacotes, porém somente alguns (*core packages*) são carregados automaticamente. Experimente carregar o pacote **readxl**. O que este pacote faz?
- Após carregar o tidyverse, desejamos produzir gráficos com **ggplot()**. Para isso, devemos carregar o pacote **ggplot2** com **library()**?
- Instale o pacote **patchwork**.
- Instale o pacote **basedosdados**.

Fundamentos do R – salvar/carregar

Existem diversas maneiras para salvar/carregar dados, sendo comum proceder por linha de comando. São exemplos de funções do pacote **base**:

```
saveRDS(object = obj, file = "my_object.rds")  # salva um objeto
readRDS(file = "my_object.rds")                # carrega um objeto
save.image(file = ".RData")                    # salva objetos
load(file = ".RData")                          # carrega objetos
writeLines(text = "Abacate!", con = "file.txt") # salva texto
readLines(con = "file.txt")                    # lê o texto do arquivo linha a linha
```

Vá em frente e descubra opções no console:

- **base::rea** ...
- **base::write** ...

Fundamentos do R – salvar/carregar

Para formatos do tipo tabular/retangular, o pacote **utils** possui uma família de funções do tipo `write.tipo-do-arquivo()` e `read.tipo-do-arquivo()`.

```
write.table; read.table()    # salva/lê arquivos do tipo tabular
write.csv(); read.csv()     # salva/lê arquivos do tipo CSV
write.csv2; read.csv2()     # salva/lê arquivos do tipo CSV
```

Comma-separated values (CSV): Arquivo do tipo texto simples (*plain text*) cujos valores são separados por vírgula. Cada linha é um registro e cada coluna é um campo, portanto, os dados terão formato tabular. É comum usar outros separadores para delimitação de campos como ponto e vírgula, espaço e tab.

`read.table()` considera espaçamentos regulares (espaço, tab) como separador, enquanto `read.csv()` a vírgula (,) e `read.csv2()` o ponto e vírgula (;). No entanto, essas funções possuem o parâmetro `sep`, sendo possível informar qual o separador do arquivo.

Fundamentos do R – salvar/carregar

Para formatos do tipo tabular/retangular, o pacote **readr** tem uma família de funções do tipo `write_tipo-do-arquivo()` e `read_tipo-do-arquivo()`.

```
write_delim(); read_delim() # salva/lê arquivos do tipo tabular
write_tsv(); read_tsv()    # salva/lê arquivos com espaçamento
write_csv(); read_csv()    # salva/lê arquivos do tipo CSV
write_csv2(); read_csv2()  # salva/lê arquivos do tipo CSV
```

Somente `read_delim()` possui um parâmetro (`delim`) que permite informar qual o separador do arquivo.

Bom! Agora, enfim, começa a parte divertida!



Questões e Prática

As funções `dir.create()` e `download.file()` permitem criar diretórios/pastas e baixar arquivos da internet, respectivamente.

- Crie uma pasta de dados no seu **R project** e faça o *download* de dois arquivos contendo séries históricas de desocupação (Fonte: PNAD contínua), conforme:

```
dir.create("dados")
download.file(url = "link 1", destfile = "dados/file-name.csv")
# Use mode = "wd" para baixar o arquivo excel no Windows
download.file(url = "link 2",
              destfile = "dados/file-name.xlsx", mode = "wd")
```

Links:

```
https://tsklipp.github.io/metodologias-informacionais-com-
r/dados/taxa_desocupacao_sexo_2012-2013.csv
https://tsklipp.github.io/metodologias-informacionais-com-
r/dados/taxa_desocupacao_idade_2012-2013.xlsx
```

Questões e Prática

- Leia o arquivo `.csv` salvo usando as funções `utils::read.csv2()` e `readr::read_csv2()` armazenando os dados em dois objetos diferentes – considere usar o parâmetro `skip = 1`. Os objetos criados são da mesma classe (tipos iguais)?
- Leia o arquivo `.xlsx` salvo usando a função `readxl::read_excel()` e armazene os dados em um objeto. Qual o tipo desse novo objeto?
- Considerando a estruturação atual dos dados, é possível executar diferentes operações sobre os dados e tarefas de análise com facilidade?

[illegible]

		1º	2º	3º	4º	1º	2º
Sexo		trimestre	trimestre	trimestre	trimestre	trimestre	trimestre
		2012	2012	2012	2012	2013	2013
Homens	Brasil	6.2	6.1	5.8	5.7	6.5	6.0
Mulheres	Brasil	10.5	9.6	9.0	8.5	10.2	9.5
Total	Brasil	8.0	7.6	7.1	6.9	8.1	7.5

Fundamentos do R – tidy data

- Cada coluna é um atributo (variável)
- Cada linha é uma observação (instância)
- Cada célula possui um único valor

Créditos: Wickham, et al (2023).

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174004898
China	1999	212258	1272015272
China	2000	216766	128042583

variables

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174004898
China	1999	212258	1272015272
China	2000	216766	128042583

observations

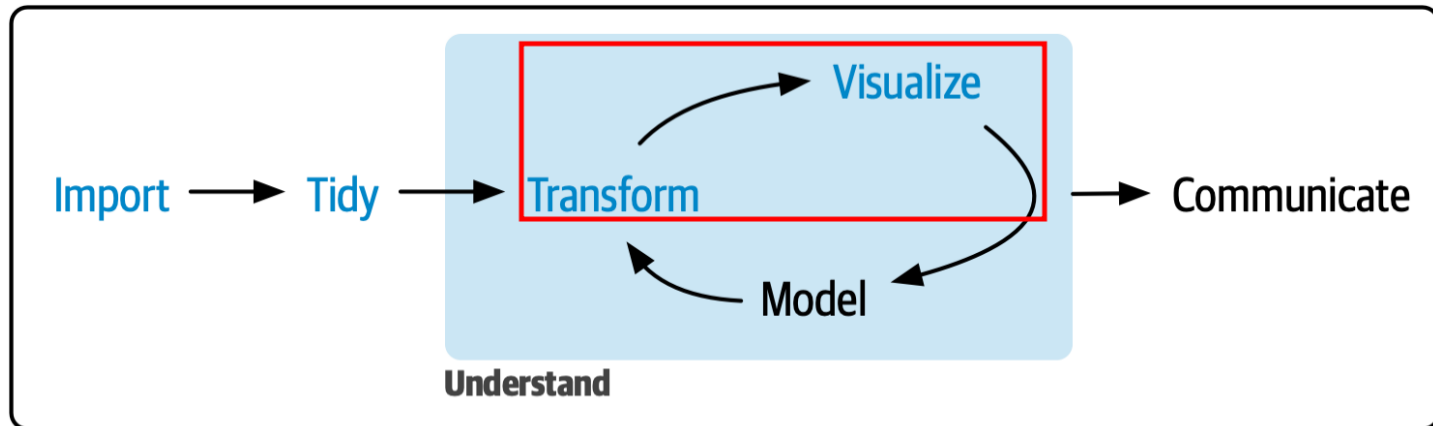
country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174004898
China	1999	212258	1272015272
China	2000	216766	128042583

values

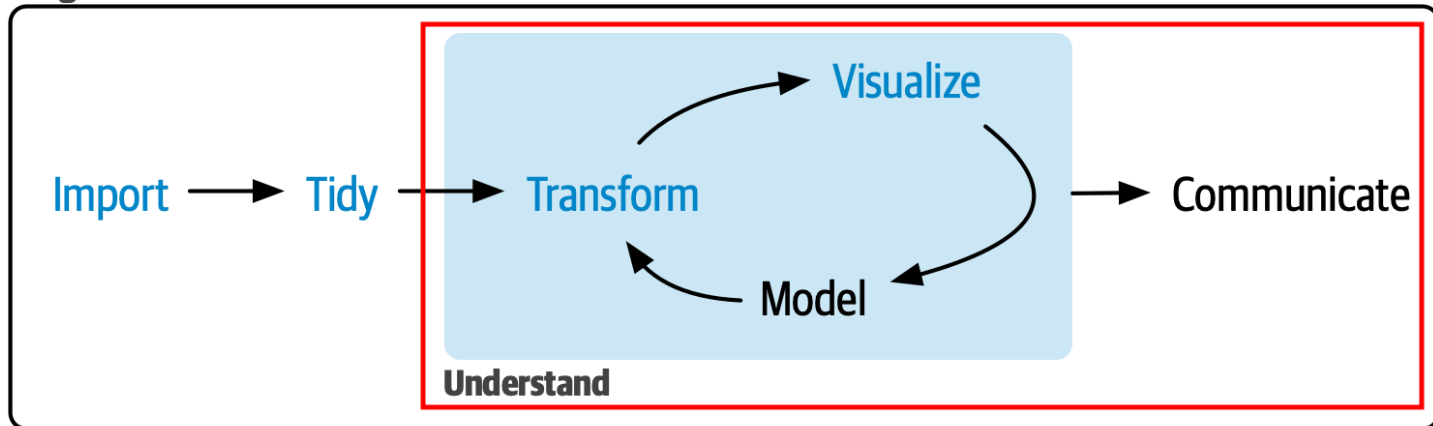
"Conjunto de dados organizados (tidy) são todos iguais, mas todo conjunto bagunçado é bagunçado à sua maneira."

— Hadley Wickham

Ciclo da ciência dos dados – "whole game"



Program



Program

Créditos: Adaptado de [Wickham, et al \(2023\)](#).

Referências

Wickham, H., M. Çetinkaya-Rundel, and G. Grolemund (2023). *R for data science (2e): import, tidy, transform, visualize, and model data*. Disponível em: <https://r4ds.hadley.nz/>, Acesso em: 14 de junho de 2023. O'Reilly Media, Inc.

Becker, E. (2023). *R for Social Scientists*. Data Carpentry. Disponível em: <https://datacarpentry.org/r-socialsci/index.html>, Acesso em: 14 de junho de 2023.

Metotologias Informacionais com 

Muito Obrigado pela Atenção!