

Uso do R para análise da dados da World Values Survey

Coletivo WVSR

2020-07-16

Sumário

Bem Vindo !	5
1 Introdução	7
1.1 O R e o RStudio	7
1.2 O que é o WVS	8
2 Instalação do R	9
2.1 LINUX	9
2.2 Mac OS X	11
2.3 Windows	12
2.4 Criando um projeto no RStudio	13
3 Bibliotecas	15
3.1 Instalação de bibliotecas adicionais	15
3.2 Carregar as bibliotecas	16
4 Importar bases de dados	17
4.1 Com o uso das ferramentas da aba Environment	17
4.2 Com o uso de linhas de comando	18
4.3 Gerar dicionário de variáveis	19
5 Descrever e Analisar os dados	21
5.1 Descrevendo os dados	22
5.2 Estatística descritiva:	23

5.3	Análises bi e multivariadas:	25
5.4	Regressao linear multipla	25
6	Visualizar os dados	27
7	Glossário	31

Bem Vindo !

Este tutorial é resultado de um trabalho conjunto do Coletivo de Análises do WVS em R. Se trata de um documento evolutivo e introdutório ao uso do R nas análises com a base estruturada da Pesquisa Mundial de Valores.

Este tutorial foi preparado em Rmarkdown com a biblioteca bookdown.

Um agradecimento aos colaboradores: ...

Capítulo 1

Introdução

1.1 O R e o RStudio

R é uma linguagem de programação, que é caracterizada como multiparadigma: tem como foco uma programação funcional, flexível, dinâmica e, principalmente, direcionada à manipulação, análise e visualização de dados.

Isso a torna muito útil, por exemplo, para estudos de bases e bancos de dados e pesquisas estatísticas. Essa funcionalidade também pode ser chamada de mineração de dados: o processo de exploração de dados em grande escala com o objetivo de buscar padrões, associações, relacionamentos e sistematização de variáveis.

Já o RStudio é um ambiente computacional de software livre, ou seja, gratuito. E tem compatibilidade com os sistemas operacionais Windows, Linux, Mac e com outros tipos de programas, como planilhas (por exemplo, Microsoft Excel) e aplicativos científicos (por exemplo, SPSS).

O que justifica seu uso nas Ciências Sociais? É possível observar que os métodos quantitativos vem sendo cada vez mais utilizados em perspectivas desse campo de conhecimento. Além disso, especialmente em nível de pesquisas internacionais do tipo survey, a disponibilidade de informações em forma quantitativa vem sendo amplamente aumentada, em diferentes aspectos: realização e englobamento geográfico, novas categorias de análise, questões em diferentes temas políticos e sociais e, também, tendo ampla divulgação de forma livre e acessível virtualmente.

1.2 O que é o WVS

A World Values Survey (WVS) é fruto de uma iniciativa acadêmica que investiga mudanças culturais desde a segunda metade do século XX em mais de cem países. É um projeto que possibilita a comparação de características culturais de diversas sociedades desde a década de 1970 e contribui dentre outros campos, para o debate sobre a relação entre desenvolvimento econômico e mudanças culturais, para o acompanhamento longitudinal de visões sobre as mudanças em curso e ampliar o conhecimento de diferentes áreas do planeta antes de acesso limitado a pesquisadores da região.

A metodologia da WVS sujeita-se à teoria da modernização e do pós-materialismo elaborada por Ronald Inglehart, que sugere que fenômenos como o crescimento do setor de serviços, a melhoria na qualidade de vida e o aumento das oportunidades educacionais nas sociedades industriais avançadas ou pós-industriais têm levado a uma gradual transformação na atividade política em democracias do Ocidente.

A tese articula duas hipóteses para explicar essa mudança: a) a hipótese da escassez: defende que as prioridades da ação humana são resultado do ambiente sócio-econômico vigente, no qual valoriza-se subjetivamente coisas e aspectos da realidade que são escassos; e b) hipótese da socialização: defende que grande parte dos valores básicos de um indivíduo derivam das condições presentes em seu período de formação, anterior à idade adulta.

Assim, a WVS explora a hipótese final de que “as mudanças nos sistemas de crenças de massas têm consequências sociais, políticas e econômicas importantes, [...] esta pesquisa proporciona outras análises a partir de seus resultados, haja vista a qualidade e a diversidade das dimensões e perguntas presentes no questionário.” (de Castro et al., 2015)

“No entanto, essa forte ligação a uma perspectiva teórica não impede que seus dados possam ser úteis para pesquisas que não usem o mesmo referencial teórico.” (de Castro et al., 2015)

Os dados gerados em todos os países integrantes da WVS ficam disponíveis para livre pesquisa na internet, no site www.worldvaluessurvey.org. The survey started in 1981 and [...] consists of nationally representative surveys conducted in almost 100 countries which contain almost 90% of the world’s population, using a common questionnaire and currently including interviews with almost 400,000 respondents.” A WVS usa um questionário de 180 variáveis permitindo a comparação.

Capítulo 2

Instalação do R

Para este tutorial, nós iremos instalar dois softwares: R e RStudio. Para utilizar o RStudio, é necessário primeiro instalar o R.

Assim, siga os passos para instalação de acordo com o sistema operacional de seu computador.

2.1 LINUX

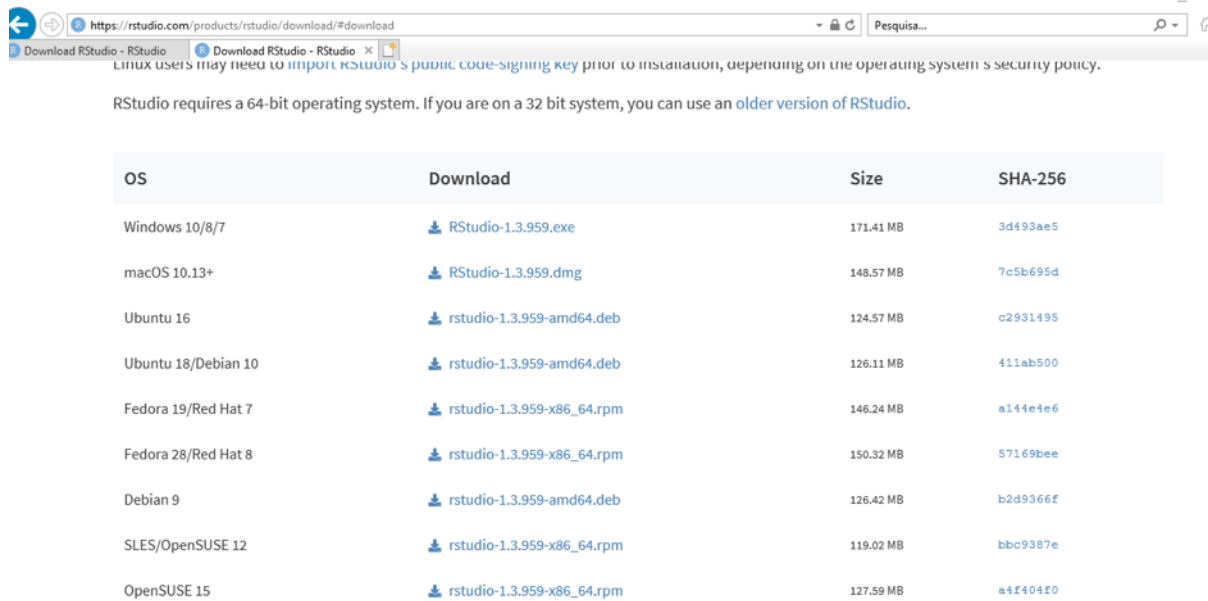
2.1.1 Instalar o R

- Passo 1: Abra o terminal. Se utilizar distribuição Fedora, pressione as teclas Super + T, e no Ubuntu Ctrl + Alt + t;
- Passo 2: Com o terminal aberto digite a seguinte linha de comando:
 - Fedora: `sudo dnf install R`
 - Ubuntu: `sudo apt-get install r-base r-base-core`
- Passo 3: Pressione a tecla Enter para confirmar;
- Passo 4: Colocar a senha do usuário;
- Passo 5: Confirmar. O R estará instalado e pode ser acessado.

Link para eventual consulta: <http://cran-r.c3sl.ufpr.br/bin/linux/>

2.1.2 Instalar o RStudio

- Passo 1: Acesse o site <https://rstudio.com/products/rstudio/download/>
- Passo 2: Encontre na página o local de download gratuito conforme figura abaixo:



The screenshot shows the RStudio download page. At the top, there is a navigation bar with the RStudio logo and a search bar. Below the navigation bar, there is a message: "Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy." Below this message, there is a note: "RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an [older version of RStudio](#)." Below the note, there is a table with four columns: OS, Download, Size, and SHA-256. The table lists download links for various operating systems, including Windows, macOS, Ubuntu, Debian, Fedora, and OpenSUSE.

OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.3.959.exe	171.41 MB	3d493ae5
macOS 10.13+	RStudio-1.3.959.dmg	148.57 MB	7c5b695d
Ubuntu 16	rstudio-1.3.959-amd64.deb	124.57 MB	c2931495
Ubuntu 18/Debian 10	rstudio-1.3.959-amd64.deb	126.11 MB	411ab500
Fedora 19/Red Hat 7	rstudio-1.3.959-x86_64.rpm	146.24 MB	a144e4e6
Fedora 28/Red Hat 8	rstudio-1.3.959-x86_64.rpm	150.32 MB	57169bee
Debian 9	rstudio-1.3.959-amd64.deb	126.42 MB	b2d9366f
SLES/OpenSUSE 12	rstudio-1.3.959-x86_64.rpm	119.02 MB	bbc9387e
OpenSUSE 15	rstudio-1.3.959-x86_64.rpm	127.59 MB	a4f404f0

- Passo 3: Encontre o sistema operacional do seu computador (Ubuntu, Fedora, Debian ou OpenSUSE) e faça download.
- Passo 4: Acesse o terminal na pasta onde foi feito o download e siga as instruções abaixo usando Fedora ou Ubuntu:
 - Fedora: `sudo dnf install nomedo_arquivo_baixado.rpm`; Ex: `sudo dnf install rstudio-1.3.959-x86_64.rpm`
 - Ubuntu: `sudo dpkg -i nomedo_arquivo_baixado.deb`
- Passo 5: Após isso, o RStudio estará instalado no seu computador e pronto para uso.

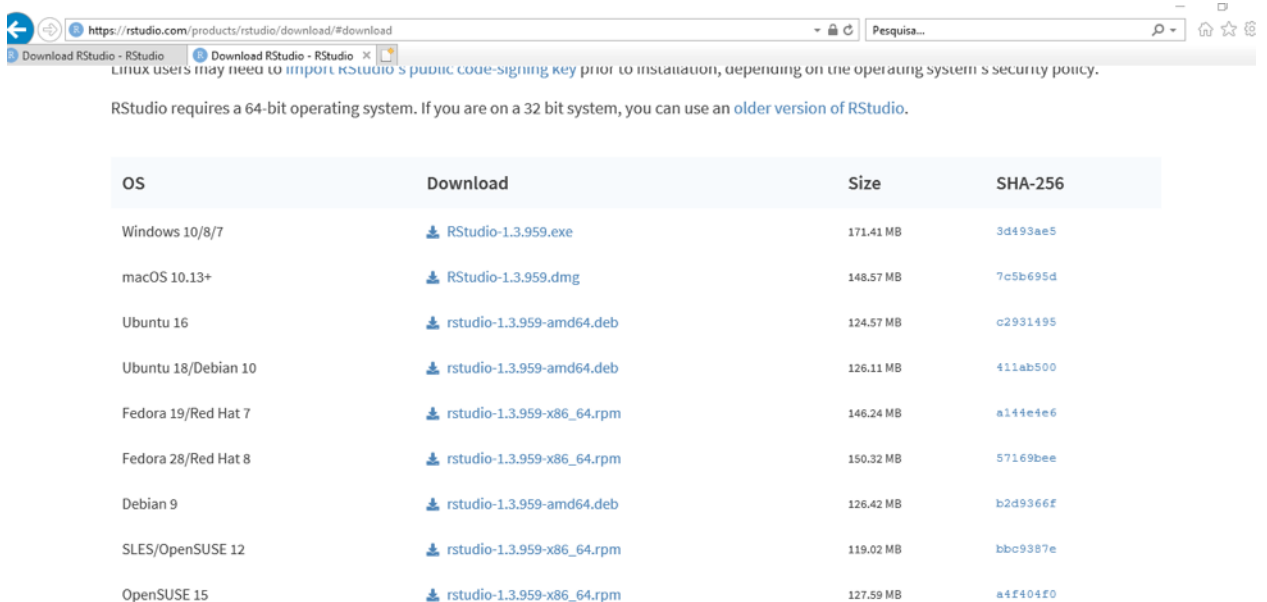
2.2 Mac OS X

2.2.1 Instalar o R

- Passo 1: Abra o site CRAN - <https://cran.r-project.org/>
- Passo 2: Clique em Download de R for (Mac) OS X.
- Passo 3: Clique duas vezes no arquivo depois de baixado que será instalado no seu computador.

2.2.2 Instalar o RStudio

- Passo 1: Acesse o site <https://rstudio.com/products/rstudio/download/>
- Passo 2: Encontre na página o local de download gratuito conforme figura abaixo:



The screenshot shows the RStudio download page. It includes a browser address bar with the URL <https://rstudio.com/products/rstudio/download/#download>. Below the address bar, there is a note: "Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy." and a warning: "RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio." Below this is a table with four columns: OS, Download, Size, and SHA-256.

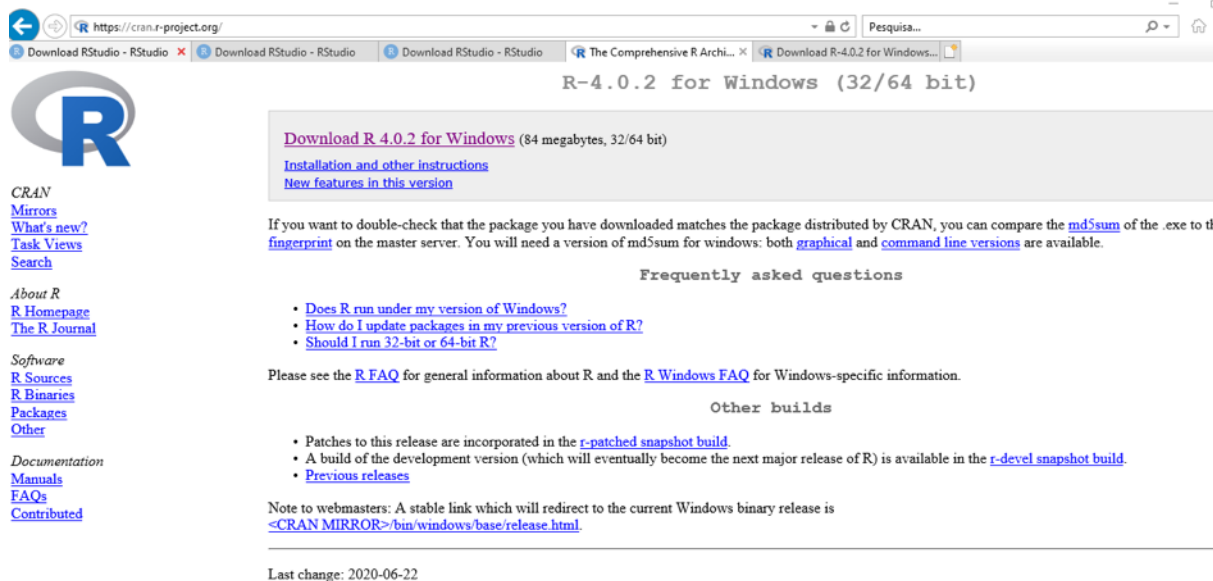
OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.3.959.exe	171.41 MB	3d493ae5
macOS 10.13+	RStudio-1.3.959.dmg	148.57 MB	7c5b695d
Ubuntu 16	rstudio-1.3.959-amd64.deb	124.57 MB	c2931495
Ubuntu 18/Debian 10	rstudio-1.3.959-amd64.deb	126.11 MB	411ab500
Fedora 19/Red Hat 7	rstudio-1.3.959-x86_64.rpm	146.24 MB	a144e4e6
Fedora 28/Red Hat 8	rstudio-1.3.959-x86_64.rpm	150.32 MB	57169bee
Debian 9	rstudio-1.3.959-amd64.deb	126.42 MB	b2d9366f
SLES/OpenSUSE 12	rstudio-1.3.959-x86_64.rpm	119.02 MB	bbc9387e
OpenSUSE 15	rstudio-1.3.959-x86_64.rpm	127.59 MB	a4f404f0

- Passo 3: Encontre o sistema operacional do seu computador (Mac OS) e faça download.
- Passo 4: Depois de baixado, clique duas vezes no arquivo para instalá-lo. Após, estará pronto para uso.

2.3 Windows

2.3.1 Instalar o R

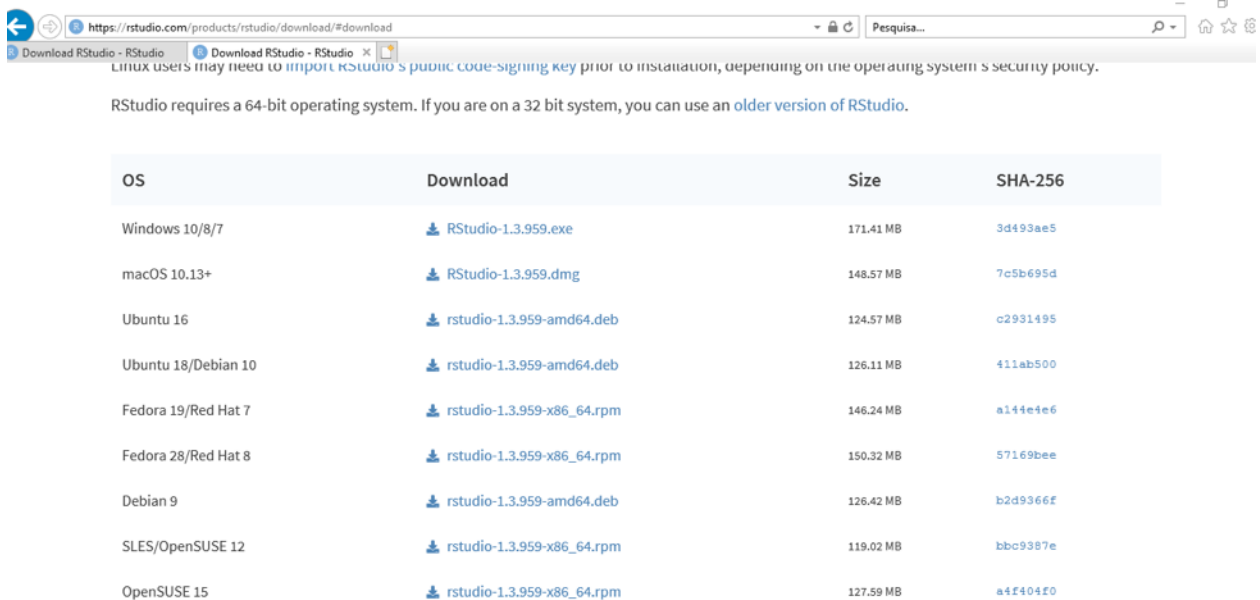
- Passo 1: Clique no seguinte link <https://cran.r-project.org/bin/windows/base/>
- Passo 2: Clique em Download R for Windows (os números que aparecem nesse arquivo de download correspondem à versão do R disponível):



- Passo 3: Clique duas vezes no arquivo depois de baixado, clique em avançar até finalizar a instalação que será instalado no seu computador.

2.3.2 Instalar o RStudio

- Passo 1: Acesse o site <https://rstudio.com/products/rstudio/download/>
- Passo 2: Encontre na página o local de download gratuito conforme figura abaixo:



The screenshot shows the RStudio download page. At the top, there's a navigation bar with the RStudio logo and a search bar. Below the navigation bar, there's a message: "Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy." Below this, a note states: "RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio." The main content is a table with four columns: OS, Download, Size, and SHA-256. The table lists download links for various operating systems including Windows, macOS, Ubuntu, Fedora, Debian, and SLES/OpenSUSE.

OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.3.959.exe	171.41 MB	3d493ae5
macOS 10.13+	RStudio-1.3.959.dmg	148.57 MB	7c5b695d
Ubuntu 16	rstudio-1.3.959-amd64.deb	124.57 MB	c2931495
Ubuntu 18/Debian 10	rstudio-1.3.959-amd64.deb	126.11 MB	411ab500
Fedora 19/Red Hat 7	rstudio-1.3.959-x86_64.rpm	146.24 MB	a144e4e6
Fedora 28/Red Hat 8	rstudio-1.3.959-x86_64.rpm	150.32 MB	57169bee
Debian 9	rstudio-1.3.959-amd64.deb	126.42 MB	b2d9366f
SLES/OpenSUSE 12	rstudio-1.3.959-x86_64.rpm	119.02 MB	bbc9387e
OpenSUSE 15	rstudio-1.3.959-x86_64.rpm	127.59 MB	a4f404f0

- Passo 3: Encontre o sistema operacional do seu computador (Windows) e faça download.
- Passo 4: Depois de baixado, clique duas vezes no arquivo para instalá-lo. Após, estará pronto para uso.

2.4 Criando um projeto no RStudio

Feitas as instalações, podemos começar a trabalhar com o R e RStudio. Uma forma prática de iniciar o trabalho com o RStudio é usando projetos.

Para iniciar um novo projeto, acesse o menu “File > New Project”, ou então o ícone na parte superior direita do RStudio. O RStudio vai perguntar se você quer criar um novo diretório para o projeto, usar um diretório existente, ou ainda clonar um repositório distante (GitHub por exemplo).

Para criar o novo diretório, basta informar o diretório onde será criado o diretório do projeto e o nome do novo diretório onde todos os documentos do projeto serão armazenados.

Ao fechar o RStudio, todos os documentos e objetos no ambiente serão salvos no projeto e você pode retomar o trabalho de onde parou (salvo as bibliotecas, que tem que ser recarregadas a cada nova seção do R!).

Capítulo 3

Bibliotecas

Quando instalamos o R já dispomos de uma grande quantidade de comandos e funções. A lista de funções do R-base pode ser lida aqui.

Para além destes, o ambiente R dispõe de uma grande quantidade de bibliotecas (ou pacotes) disponíveis. No momento em que escrevemos este tutorial, a plataforma de contribuições CRAN lista 16045 bibliotecas disponíveis. As diferentes contribuições são minimamente agrupadas por tópico. Por exemplo, é possível conhecer os pacotes disponíveis para as ciências sociais.

Neste tutorial vamos usar as funções do R-base e algumas bibliotecas adicionais que facilitam algumas operações e já dão conta da maior parte das necessidades de um pesquisador interessado em usar o WVS.

As bibliotecas que são usadas neste tutorial fazem parte da filosofia e gramática tidyverse, entre elas os pacotes `dplyr`, `haven` e `ggplot2`. Mais detalhes podem ser obtidos no livro dos autores disponível neste link. Além do `tidyverse`, usamos a biblioteca `codebook` para visualizar os dados.

3.1 Instalação de bibliotecas adicionais

Como mencionamos acima, quando instalamos o R-base já instalamos uma série de funções e comandos. Dentre eles, existe um comando específico para instalar novas bibliotecas do repositório CRAN.

Para instalar as bibliotecas usa-se o comando `install.packages("nome_da_biblioteca")`. Outra maneira de instalar novas bibliotecas é usar a aba “Packages”. Esta aba permite visualizar as bibliotecas já instaladas e instalar novas usando o botão “Install”.

Entretanto, recomendamos que as bibliotecas que serão usadas nas análises sejam indicadas no início do código R onde são feitas as análises. Assim, o

pesquisador que recebê-lo sabe quais os requisitos necessários para reproduzir as análises.

Para instalar bibliotecas adicionais, seguimos o exemplo abaixo:

```
# para instalar as bibliotecas da gramática tidyverse  
install.packages("tidyverse")  
  
# para instalar a biblioteca codebook para visualizar os dados (ver no capítulo seguinte)  
install.packages("codebook")
```

Quando rodamos o comando, usando **Ctrl + Enter** no script ou **Enter** diretamente no console, o R vai verificar se existem dependências e instalar todas as que forem necessárias para o seu bom funcionamento. **Assim, é importante que se faça a instalação de todas as dependências indicadas pelo R.** Verifique no console, e se o R perguntar se quer fazer a instalação das dependências, diga “Yes”.

3.2 Carregar as bibliotecas

Uma vez instaladas as bibliotecas, é preciso carregar as bibliotecas para que possam ser usadas.

O comando para carregar as bibliotecas é `library("nome_da_biblioteca")`.

```
# carregar o tidyverse  
library(tidyverse)  
  
## carregar a biblioteca haven, que é usada no capítulo seguinte para  
library(codebook)  
  
# carregar o codebook  
library(codebook)
```

Sempre que abrir e fechar o R-Studio é preciso carregar as bibliotecas novamente, pois o R reinicia a sessão somente com os comandos do R-base.

Carregadas as bibliotecas podemos importar os dados que vamos usar.

Capítulo 4

Importar bases de dados

O R importa arquivos salvos em seu próprio formato e de variados outros softwares, como Excell, SPSS e Stata, e em diversos outros formatos, incluindo Text.

Há duas formas de importação de arquivos no R: com o uso das ferramentas do Environment ou com o uso de linhas de comando escritas diretamente no console ou em um script.

4.1 Com o uso das ferramentas da aba Environment

Por esta forma, você usa as ferramentas do R do mesmo jeito que faz com qualquer outro software, como os do pacote Microsoft Office, por exemplo.

4.1.1 Arquivos em formato R

Para arquivos formato R (formato `.rdata`):

- Clique sobre o icone Abrir Arquivos.
- Procure o seu arquivo nos locais do seu computador.
- Clique no arquivo e clique em Open ou clique duas vezes sobre o arquivo.

Pronto! O seu arquivo vai aparecer no Environment.

4.1.2 Arquivos em outros formatos

Para importar os dados em outros formatos, é preciso usar bibliotecas específicas para esse fim. No caso deste tutorial, a biblioteca usada é a **haven**, da gramática Tidyverse.

- Clique sobre o ícone Import Dataset.
- Escolha e clique sobre o formato em que o seu arquivo foi previamente salvo.
- Clique em Browse.
- Procure o seu arquivo nos locais do seu computador.
- Clique no arquivo e clique em Open ou clique duas vezes sobre o arquivo.

Pronto! O seu arquivo vai aparecer no Environment.

4.2 Com o uso de linhas de comando

4.2.1 Arquivos em formato R

Para carregar arquivos no formato nativo do R usando linhas de comando:

- Abra um novo script R e digite: `load("nome_do_arquivo.rdata")`.
 - Dica: Deixe o cursor entre as aspas e tecla TAB. O R vai lhe mostrar os locais do seu computador. Se seu arquivo está em alguma pasta ou subpasta o R também vai lhe mostrar os locais do seu computador.
- Clique no arquivo que você quer importar
- Digite Control + Enter, ou clique em Run (disponível na aba superior do Source Code do seu R).

Pronto! O seu arquivo vai aparecer no Environment.

4.2.2 Arquivos em outros formatos

Para carregar arquivos em outros formatos, usamos a biblioteca **haven** do tidyverse (Certifique-se que ela foi carregada usando o comando `library(haven)`).

Cada tipo de arquivo tem um comando específico no haven. será o mesmo para qualquer origem, com alteração apenas da extensão do arquivo:

Arquivos SPSS: função `read_sav()` Arquivos Excel: função `read_csv()` Arquivos Stata: função `read_dta()` Arquivos SAS: função `read_sas()`

Veja o exemplo com a extensão do SPSS (arquivos formato .sav). Para que a base se transforme em um objeto no ambiente do R, é preciso que se atribua um nome da seguinte forma:

```
# carregar a biblioteca haven
library(haven)

# abrir usando a função e criar um objeto no ambiente R
base_do_wvs <- read_sav("diretório_do_projeto/nome_do_arquivo.sav")
```

Como das outras vezes, deixe o cursor entre as aspas e tecle TAB. O R vai lhe mostrar os locais do seu computador. Em seguida, é só clicar sobre o arquivo que você procura.

- Digite Control + Enter, ou
- Clique em Run (disponível na aba superior do Source Code do seu R).

Pronto! O seu arquivo vai aparecer no Environment.

Com outras extensões você usará, por exemplo, `read_cvs` ou `read_dta`. Para mais informações acesse a ajuda da biblioteca `haven`

4.3 Gerar dicionário de variáveis

Para navegar na base de dados recém importada, reconhecer as variáveis importadas e suas características, existem diversas ferramentas disponíveis.

Uma delas é o pacote `codebook`, que gera um dicionário de variáveis. Nos exemplos abaixo, mostramos como gerar um dicionário de variáveis estático na aba “Preview” e também na forma de um vetor que pode ser visualizado como uma tabela.

```
# carregar a biblioteca codebook
library(codebook)

# gerar um dicionário de variáveis estático navegável na aba "Preview"
label_browser_static(base_de_dados)

# gerar uma tabela com todas as variáveis
dicionario <- codebook_table(base_de_dados)

# visualizar na forma de tabela junto aos scripts R
view(dicionario)
```

Feita a importação da base e com o dicionário nas mãos podemos passar a descrição, análise e visualização dos dados.

Capítulo 5

Descrever e Analisar os dados

Após instalar o *software*, reconhecer seu ambiente de trabalho, instalar e carregar as bibliotecas e importar os bancos, começa a verdadeira diversão: finalmente trabalhar com os dados.

O R permite que se faça desde as operações mais simples, como uma calculadora, passando pela organização e limpeza dos dados, até análises estatísticas mais avançadas.

Um excelente começo nesta empreitada é a descrição dos dados que não só garante o divertimento do pesquisador, como também permite que se façam excelentes análises e conclusões. A partir da estatística descritiva, obtém-se um conjunto de ferramentas que permitem: a) organizar os dados - agrupando, selecionando, filtrando, ordenando, criando variáveis; b) descrever e analisar os dados - resumindo-os com base em sua média, moda, mediana, cruzando e contando as variáveis, criando gráficos e tabelas. A principal função da estatística descritiva é resumir os dados e facilitar a assimilação da informação (Agresti et al., 2012).

Além da descrição dos dados, o R também oferece suporte para que sejam feitos testes estatísticos mais avançados bi e multi-variados, dentre eles correlação e regressão linear multivariada. A correlação tem por função avaliar o comportamento de duas variáveis em função uma da outra, mensurando a interdependência das mesmas e o seu grau de associação. Logo a correlação é apropriada se o seu propósito é analisar se há relação entre a variação de duas dimensões e qual a força desta relação, como por exemplo nas clássicas premissas: nível educacional e renda variam de maneira diretamente proporcional ou apoio à democracia varia de maneira inversamente proporcional a valores autoritários. A regressão linear multivariada permite estabelecer valores estimativos

de uma variável dependente em relação a um conjunto de outras variáveis independentes, isto é resulta no nível de previsibilidade de uma variável em função de outras variáveis explicativas. Deste modo, este tipo de teste se aplica aos casos em que se almeja avaliar e estabelecer potencial previsibilidade de uma variável em função de outras, e não apenas comparar seus comportamentos; por exemplo prevendo a taxa de desconfiança em relação aos partidos com base na quantidade de pessoas que estão satisfeitas com a democracia.

Nesta seção iremos aprender, então, os comandos do R que permitem criar anotações, criar variáveis e vetores, conectar sentenças do código de programação, visualizar os valores máximo e mínimo de uma variável, obter a média, somar os valores de uma variável, filtrar, selecionar, ordenar, contar, cruzar, criar nova coluna, resumir os dados a partir de alguma operação, excluir valores da análise, visualizar variáveis, fazer gráficos, fazer teste de correlação e análise de regressão multivariada. Ou seja, tudo o que se precisa para, em ordem de importância: ter adoráveis momentos com o seu banco de dados e construir conhecimento a partir de análises robustas.

5.1 Descrevendo os dados

5.1.1 Criar variáveis e vetores

Para criar uma variável basta escolher um nome para ela e atribuir através do sinal ‘<-’ um valor para a mesma, lembrando sempre de que o R não aceita caractere do tipo espaço, então se o nome de sua variável tiver mais de um termo é necessário usar o conector underline ‘_’ entre as palavras. Para criar vetores, segue-se a mesma lógica, mas os valores devem ser inseridos entre os parênteses do código: ‘c()’, para indicar que há um conjunto de valores; estes valores devem ser separados por vírgulas. Os valores atribuídos podem ser números simples, podem ser operações como soma, subtração, multiplicação. As variáveis e vetores, após criados podem também ter seus valores submetidos a algum tipo de operação. Exemplos:

```
# Criar um vetor com base em um resultado de cálculo
Meu_resultado <- 1 + 5

# Criar um vetor e multiplicar por um valor
Brasil <- c(1, 3, 10, 12, 19, 21, 28, 30, 37)
Brasil * 3

# Cálculo da idade com base no ano de nascimento
Ano_Nasc <- c(1990, 1984, 1961)

# Calcular as idades
```

```
2020 - Ano_Nasc

# Armazenar como um vetor no ambiente
Idade <- 2020 - Ano_Nasc
```

5.1.2 Funções básicas e comandos úteis

Após carregar seu banco de dados ou criar suas próprias variáveis é possível começar a conhecer e operar os dados observando seus valores máximos, mínimos, média e efetuando operações a partir dos valores dentro da variável. Para executar tais funções, alguns comandos do r-base são os seguintes para a variável “Idade”, por exemplo:

- Valor máximo: `max(Idade)`
- Valor mínimo: `min(Idade)`
- Média: `mean(Idade)`
- Soma: `sum(Idade)`

Na filosofia do tidyverse, para executar mais de um comando é necessário inserir entre eles o conector `%>%`. Além disso, pode ser útil acrescentar anotações que não sejam reconhecidas pelo programa como uma linha de código, para tanto basta inserir no início da frase o caractere jogo da velha `#`.

```
# Criar anotações: #
# Conectar as linhas do código: %>%
# Exemplos (os códigos das variáveis são usados na base longitudinal do WVS: X003 = Idade; S003 = ...

base_do_wvs %>%
  summarise(media = mean(X003, na.rm = TRUE)) %>%
  count(S003, X001)
```

5.2 Estatística descritiva:

Partindo para funções mais sofisticadas, é possível organizar o banco de dados de maneira sistemática e fazer análises resumidas das variáveis. O comando `filter()` filtra as observações (linhas) baseadas em uma condição determinada pelo pesquisador, como por exemplo selecionar apenas determinados países, gênero feminino ou masculino, faixas de renda, idade e escolaridade.

Com o comando `select()` é possível escolher as variáveis específicas (colunas) com que se deseja trabalhar. O comando `mutate` cria uma nova variável (coluna) ou modifica uma já existente a partir de manipulações dos dados.

A função `count()` é muito interessante porque permite que se faça análises dos dados ao apresentar sua contagem (frequência) e cruzar variáveis.

Para agrupar conjuntos específicos de variáveis, utiliza-se o `group()`.

Outra forma de manipular e avaliar os dados é utilizando o comando `summarise()` para resumir os dados em função de alguma operação (média/mean e mediana/median).

O comando `arrange()` ordena os resultados. Outros dois comandos importantes neste contexto é o `na.rm = TRUE` para excluir missing values e o `view()` para fazer uma visualização simples de variáveis específicas.

Para exibir os dados em uma tabela, utiliza-se o comando `table()`. A seguir um resumo com o código que deve ser utilizado para executar cada uma destas funções e exemplos a partir das variáveis da base de dados do WVS:

```
# filter(), exemplo filtrar observações baseado em uma condicao Brasil:
base_do_wvs %>%
  filter(S003 == 76)

# select(), exemplo selecionar somente país e sexo:
base_do_wvs %>%
  select(S003, X001)

#count(), exemplo contar e cruzar as variáveis país e apoio à democracia:
base_do_wvs %>%
  count (S003, E235)

# mutate(), criar uma variável faixas etárias:
base_do_wvs %>%
  mutate(faixa_et = if_else(X003<30, "Jovem", "Não Jovem"))
```

A função 'if_else' divide os valores em dois grupos: os que se encaixam na condição e os que não se encaixam, no caso do exemplo os que possuem idade inferior a 30 ($X003 < 30$) é classificado como Jovem, os que não possuem são classificados como Não Jovem.

```
# summarise(), exemplo resumir os dados pela média e mediana de idade
base_do_wvs %>%
  summarise(media = mean(X003), mediana = median(X003))

# para excluir missing values: na.rm = TRUE
base_do_wvs %>%
  summarise(media = mean(X003, na.rm = TRUE),
            mediana = median(X003))
```



```
# arrange(), exemplo ordenar a variável idade:
base_do_wvs %>%
  arrange(X003)

# view(), exemplo visualizar a variável países:
base_do_wvs %>%
  view(S003)

# table(), exemplo visualizar em uma tabela a variável:
base_do_wvs %>%
  table(variavel)
```

5.3 Análises bi e multivariadas:

As análises descritivas oferecem diversas possibilidades para conhecer o comportamento, distribuição e padrões das diferentes variáveis, contudo quando o objetivo é estabelecer relação entre as mesmas é necessário adotar testes mais avançados bi e multivariados. Para avaliar a taxa de associação entre duas variáveis e a capacidade explicativa de um conjunto de variáveis sobre uma variável dependente específica, devemos utilizar respectivamente correlação e regressão linear multivariada.

5.3.1 Correlação

Para fazer o teste de correlação basta selecionar o par de variáveis em questão e adicionar o comando `cor()` utilizando todas as observações, como no exemplo abaixo:

```
# Selecionar duas variáveis e extrair a correlação
base_do_wvs %>%
  select(X002, X003) %>%
  cor(use = "complete.obs")
```

5.4 Regressão linear múltipla

Para efetuar o teste de regressão linear múltipla basta atribuir a uma variável resultado `<-` a função `lm()`.

Exemplo:

```
lm(variável dependente ~ scale (variável independente 1) + scale
  (variável independente 2) + scale (variável independente n), data
  =.)
```

É necessário sinalizar antes para o programa qual é o banco pelo nome do objeto e o recorte através das funções `group_by()` ou `filter()`. Nos exemplos a seguir é possível visualizar melhor a estrutura do comando:

```
# Exemplos de regressão linear
# Exemplo 1 (usando como variável resposta a renda (X047), idade (X002) e educação (X003))
resultado <- base_do_wvs %>%
  group_by(S003) %>%
  lm(X047 ~ X025R + X002, data = .)

# Neste exemplo o "." indica que a base de dados é enviada na função lm() depois do "e"

# Exemplo 2:
resultado_bol <- base_do_wvs %>%
  filter(S003 == 68) %>%
  lm(X047 ~ scale(X025R) + scale(X002), data = .)

# Neste exemplo as variáveis são normalizadas com a função scale() e foi filtrado apenas
```

Capítulo 6

Visualizar os dados

Para visualizar os dados, usamos a biblioteca `ggplot2()` também da gramática Tidyverse.

Após organizar e manipular os dados e fazer análises descritivas, pode ser uma boa estratégia apresentar as informações em gráficos para evidenciar padrões, tendências e comparações. Conforme visto anteriormente, a biblioteca para plotar gráficos é a `ggplot()`. A biblioteca `ggthemes()` permite aplicar alguns temas já pré-estabelecidos.

O `ggplot` é uma biblioteca que constrói os gráficos em camadas. Primeiramente se define a “folha” com uma base de dados que será usada. Em seguida a geometria que será usada e os demais parâmetros dos gráficos.

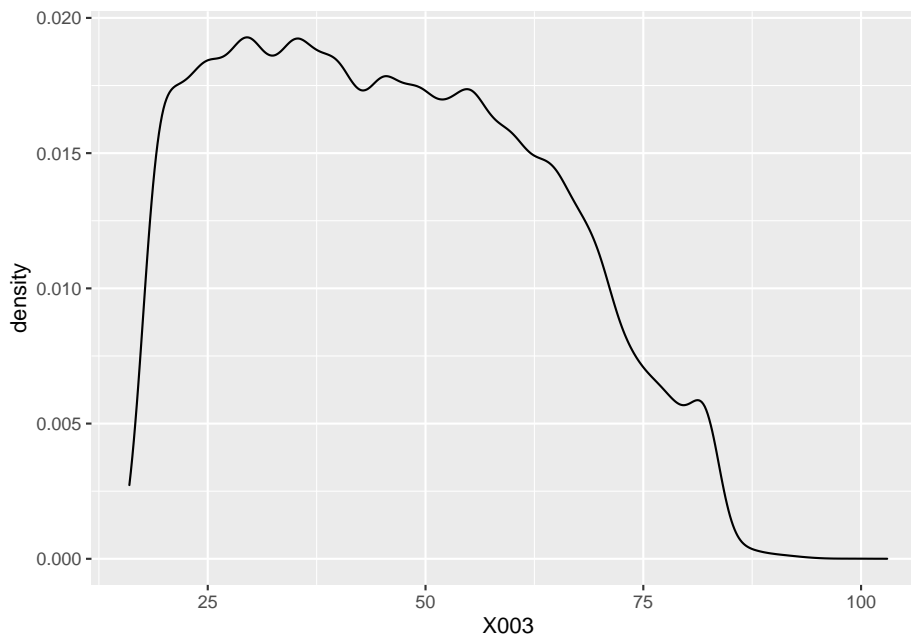
O comando básico do `ggplot` é: `ggplot(data = dados, aes(x = Explicativa, y = Resposta)) + geoma + ...`.

Tipos de Geomas disponíveis:

- Histograma: `geom_hist()`
- Barras: `geom_bar()`
- Pontos: `geom_point()`
- Diagrama de caixa (boxplot): `geom_boxplot()`
- Linhas: `geom_line()`

O guia do `ggplot2` pode ser encontrado aqui. Nesta galeria podem ser encontrados diversos exemplos de gráficos para se inspirar.

```
# Exemplo 1 plotar um gráfico de densidade da variável idade:  
ggplot(df_wvs7, aes(x = X003)) + geom_density()
```



No exemplo abaixo, reproduzimos o mapa cultural do Inglehart (Inglehart and Welzel, 2005) para alguns países. Mas antes é preciso fazer algumas manipulações com os dados com o que aprendemos nos capítulos anteriores.

```
# Criar uma lista com os países que se quer analisar
países <- c(76, 32, 152, 170) # 76=Brasil, 32=Chile, 152=Argentina, 170=Colombia

# Extrair e retrabalhar as variáveis que precisamos
df_y001 <- df_wvs7 %>%
  select(S024, S020, S003, tradrat5, survself) %>% # selecionar as variáveis de interesse
  filter(S003 %in% países) %>% # filtrar os países
  group_by(S024, S020, S003) %>% # agrupar por onda-país, onda e país
  summarise(tradrat5 = mean(tradrat5, na.rm = T),
            survself = mean(survself, na.rm = T)) # agregar os dados pela média
```

Feito isso podemos criar a figura:

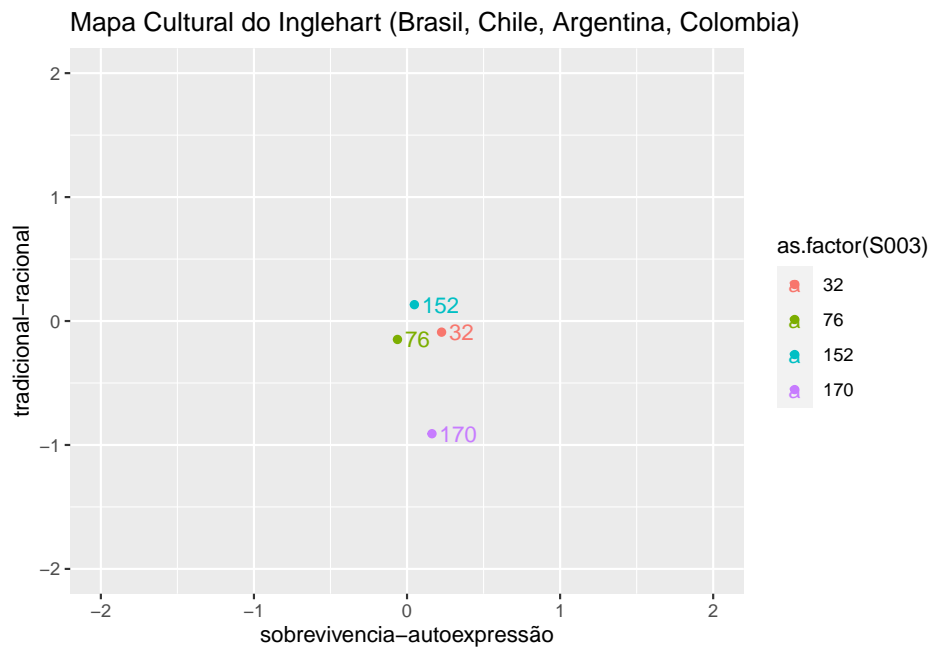
```
# lembrar de carregar a variável ggplot se ela não tiver sido carregada
library(ggplot2)

# Reproduzir o Mapa Cultural Inglehart
ggplot(df_y001, aes(x = survself,
                    y = tradrat5,
                    color = as.factor(S003),
                    label = as.factor(S003))) + # definir os dados e as dimensões
```

```

geom_point() + # definir a geometria principal (pontos)
geom_text(hjust = 0, nudge_x = 0.05, check_overlap = TRUE) + # adicionar os rótulos com a geom
scale_x_continuous(limits = c(-2, 2)) + # restringir o eixo x e y entre -2 e 2
scale_y_continuous(limits = c(-2, 2)) +
labs(x = "sobrevivencia-autoexpressão",
     y = "tradicional-racional",
     title = "Mapa Cultural do Inglehart (Brasil, Chile, Argentina, Colombia)") # adicionar os

```



Capítulo 7

Glossário

- Programação: alguns conceitos relevantes para aprender uma linguagem de programação.
 - Variável: representa um local de armazenamento na memória, com um nome e um valor, mas que não é o valor em si, porque este pode ser constantemente alterado. Cada variável possui diferentes propriedades, por exemplo, pode ser numeric (números) ou string (letras). Inserindo uma metáfora: cada variável seria uma “caixinha” que pode guardar alguma coisa, e tem diferentes características (grande, pequena, quadrada, redonda). `x <- 2` #cria variável "x" e atribuir valor 2
 - Objeto: cada objeto é único, específico, e pode ser sempre referenciado. Seu conteúdo pode ser um atribuído diretamente, ou ser o resultado de uma operação. Cada objeto pode ser armazenado em uma variável. Na metáfora, um objeto pode ser uma carta, uma moeda, um lápis-de-cor. `y <- 5` #criar objeto com nome y e atribuir valor 5
 - Vetor: uma sequência ordenada de elementos. Importante notar que um vetor armazena apenas um “tipo” de informação, por exemplo, só números, o que se chama de unidimensional. Como se fosse uma caixinha com várias cartas (uma de fulano, outra de beltrano, etc), ou uma caixinha com várias moedas (1 euro, 50 centavos de dólar, 25 centavos de real, etc), ou uma caixinha com diferentes lápis-de-cor (rosa, azul, verde). `z <- c(1, 2, 3)` #criar vetor z e atribuir os valores 1, 2 e 3
 - Lista: é como um vetor “avançado”, pois também é uma sequência de elementos mas pode guardar diferentes tipos de dados (números, letras, etc), ou seja, é multidimensional. Assim, pode ser uma mesma caixinha que guarda, tudo junto, cartas, moedas e lápis-de-cor. `lista <- list(idade = 20, "oi", 35, "tchau")` #criar

"lista" e atribuir diferentes conteúdos

- Interfaces: o RStudio possui quatro interfaces, que são as “pequenas telas” ou “janelinhas” que aparecem ao iniciar o programa. Pensando em uma cozinha, o source seria a receita, o passo-a-passo; o console seria o fogão, onde tudo está acontecendo, onde forma o resultado; e o environment seria uma mesa, onde estão os ingredientes disponíveis. Os pacotes e arquivos são o armário com os utensílios de cozinha e ingredientes disponíveis.
 - Source: local onde são inseridos os códigos-fonte, no canto superior esquerdo.
 - Console: também chamado de terminal, onde são mostrados os resultados dos comandos executados pelos códigos, no canto inferior esquerdo.
 - Environment: também chamado de ambiente, onde são mostrados os elementos (por exemplo, vetores, bases de dados, etc) que foram criados, no canto superior direito.
 - Viewer: também chamado files, onde são mostrados os arquivos, pacotes, pastas, entre outros, no canto inferior direito.
- Funções: diferentes “verbos”, tanto do r-base quanto na gramática Tidyverse, realizam diferentes tipos de funções.
 - Count: conta quantas vezes aparece aquela variável, por exemplo, numa base de dados, e mostra seu valor. `count(x)` #contar a quantidade da variável x
 - Filter: filtrar observações (“cadastros”) baseadas em uma condição. `filter(x == 1)` #filtra todos os valores de x e deixar apenas os cadastros em que x tem o valor 1
 - Select: selecionar variáveis - no plural -, ou seja, escolher mais de uma variável. `select(x, y, z)` #selecionar as variáveis x, y e z
 - Group_by: agrupar variáveis, por exemplo, em uma base de dados grande, “juntar” a partir de um determinado quesito. `group_by(país)` #juntar os dados com base nos diferentes valores da variável país
 - View: visualizar a operação realizada, normalmente, o R cria uma tabela com o(s) conteúdo(s), ficando mais “simples” de enxergar os dados. `view(objeto)`
 - Summarise: resumir, sintetizar, os dados baseados em uma operação, ou seja, criando uma espécie de “atalho”. `summarize(media = mean(idade))` #tem a média das idades das pessoas como critério para resumir ou sintetizar os dados
 - Mutate: criar nova informação com base em outras, visualmente, é como se criasse uma nova “coluna”. `view(a = b + c)` #a ‘coluna’ a vai ter o valor da soma de b e c
- Códigos-chave:
 - # para fazer comentários, ou seja, escrever anotações sem que elas sejam executadas como códigos.

- `%>%` para criar um comando no sentido de “e então”, ou seja, especificar que seja realizado o próximo passo do código em seguida, dica: clicar **Ctrl + Shift + M**.
 - **Ctrl + enter**, ou clicar no botão **Run**, para “rodar”, ou seja, executar o código programado/escrito.
- Estatística:
 - Variáveis numéricas: quantitativas, podem ser discretas (contagem, ex.: número de filhos) ou contínuas (mensuração, ex.: peso, altura)
 - Variáveis categóricas: qualitativas, podem ser ordinais (quando há ordem, ex.: grau de instrução) ou nominais (classificação, ex.: sexo, raça)

Referências Bibliográficas

Agresti, A., Finlay, B., and Viali, L. (2012). *Métodos Estatísticos para as Ciências Sociais*. Penso, Porto Alegre, edição: 4 edition.

de Castro, H. C. O., Ranincheski, S., and Capistrano, D. (2015). O conteúdo da globalização para os latino-americanos: Uma análise a partir da Pesquisa Mundial de Valores–WVS.

Inglehart, R. and Welzel, C. (2005). *Modernization, Cultural Change, and Democracy: The Human Development Sequence*. Cambridge University Press, Cambridge, UK ; New York.