

Manipulação de dados

Instrutor: Daniel Pagotto

Realização



Daniel Pagotto

- Mestre em Administração (UFG) e bacharel em administração (UnB)
- Coordenador Adjunto do Laboratório de Pesquisa em Empreendedorismo e Inovação da UFG (LAPEI-UFG)
- Consultoria em órgãos públicos federais



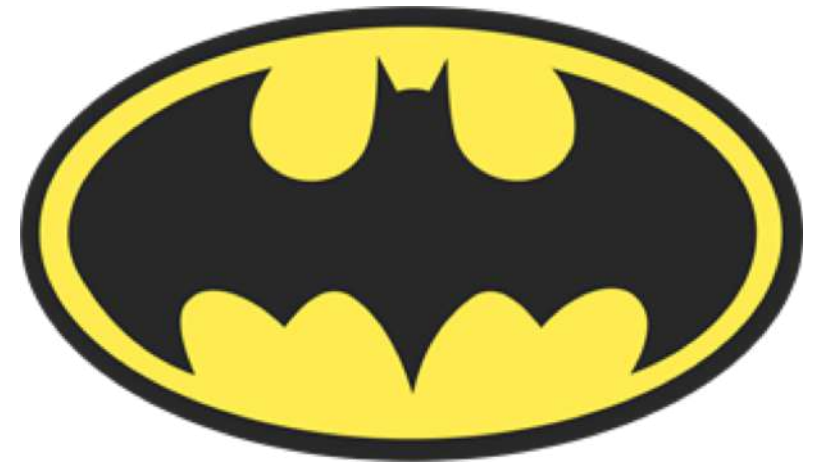
Objetivos do treinamento

- Nivelar conhecimentos básicos
- Manipular dados em um dataframe
- Demonstrar princípios de visualização de dados



A linguagem R

- Linguagem de programação e ambiente computacional estatístico mantido por um conjunto de colaboradores do mundo todo
- 15639 pacotes (*packages*)
 - dplyr, ggplot2, rtweet, tm, qdap, dendextend, tidyverse



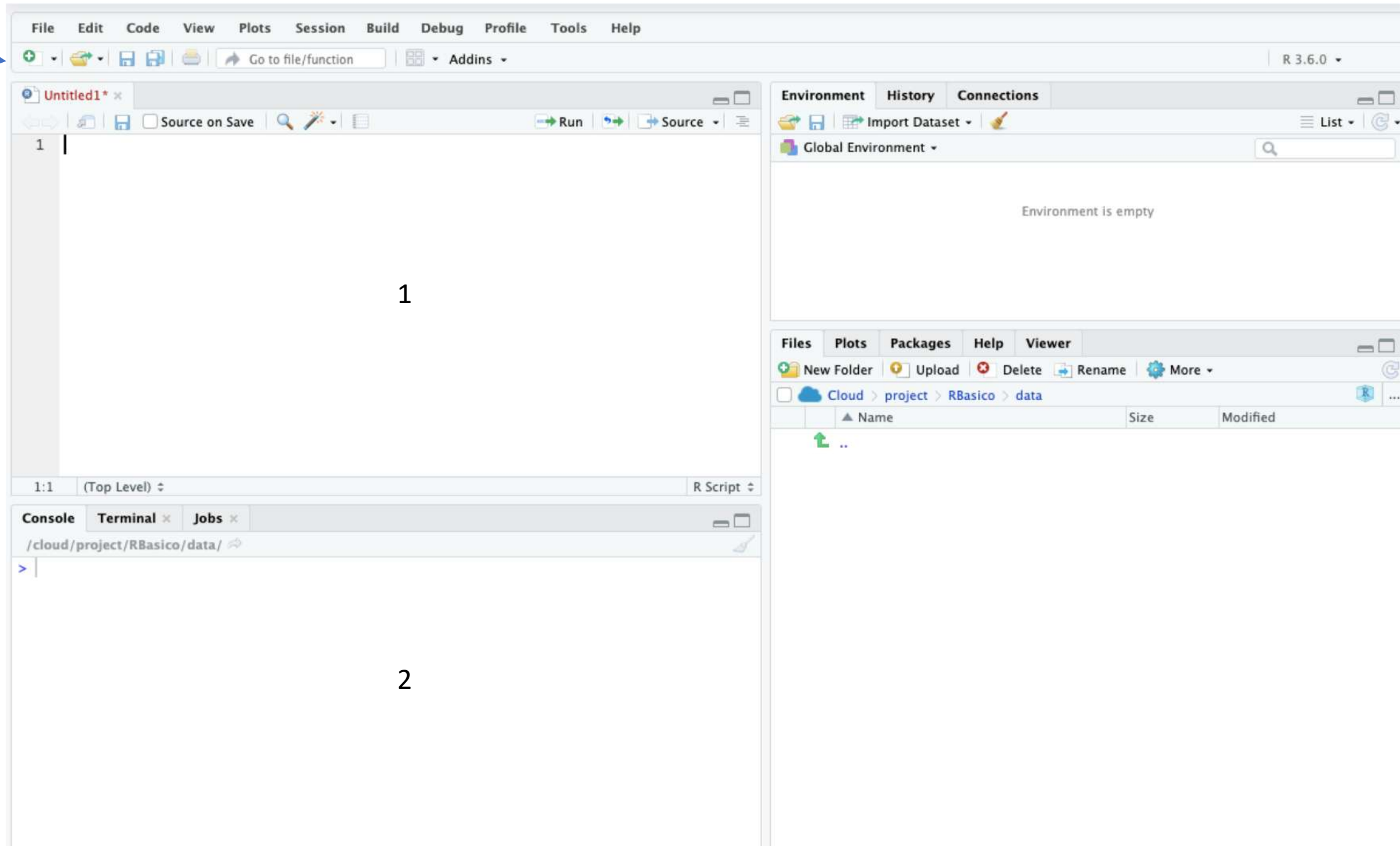
Fontes: <https://www.r-project.org/contributors.html>
https://cran.r-project.org/web/packages/available_packages_by_name.html

Toques antes de iniciar

- Principalmente no começo, é muito comum acontecer pequenos erros de digitação. Portanto, fiquem atentos. (ex.: `roud(number)`)
- O R é *case sensitive*! (ex.: `pesoDaniel` \neq `PesoDaniel`)
- Os comandos do R são baseados em palavras ao inglês.
- *Take a deep breath!*



Novo script



The screenshot displays the RStudio Cloud web interface. At the top, the title bar reads "Your Workspace / Untitled Project" with a link to "Click to name your project". The user's name, "Daniel do Prado Pagotto", is visible in the top right corner. The main menu includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for creating a new file, saving, and other actions. A blue arrow points to the "New script" button (represented by a green plus icon). The central editor area shows a single line of code: "1". The right sidebar contains the "Environment" panel, which is currently empty, and the "Files" panel, which shows a directory structure: "Cloud > project > RBasico > data". The bottom panel is the "Console", which shows the current directory path: "/cloud/project/RBasico/data/".

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins R 3.6.0

Untitled1* x

Source on Save Run Source

1

Environment History Connections

Global Environment

Environment is empty

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Cloud > project > RBasico > data

Name Size Modified

..

Console Terminal x Jobs x

/cloud/project/RBasico/data/

>

Rstudio e
Rstudio Cloud

O R é uma calculadora

- O R vai executar os comandos que você instruir. Portanto, para começarmos nossa jornada, o R é uma calculadora.

```
# Operações básicas  
5 + 5  
10 - 6  
10*2  
5/2  
5**2  
sqrt(16)  
5*(50-45)
```

- Observe que, assim como na matemática, você pode usar parênteses para priorizar a ordem de um cálculo
- # são usados para fazer comentários.
- Digite em cada linha de script exemplos de cálculos usando as operações básicas e execute-os usando Ctrl + Enter (Cmd + Enter) posicionado sobre a linha

Atribuição de variáveis

- O resultado das operações que você criou pode ser armazenado em variáveis. Para isso use a notação (<-)

```
# Operações básicas e atribuições  
x <- 5 + 5  
y <- 10 - 16  
a <- 9  
soma <- a + x  
nome <- "daniel"  
certo <- TRUE
```



Eu executei esse comando aí, mas não aconteceu nada...

Vamos criar um programa que calcula IMC

- Vamos criar duas variáveis: peso e altura

```
# Operações básicas  
pesoDaniel <- 79  
alturaDaniel <- 1.78  
  
imcDaniel <- pesoDaniel/alturaDaniel**2
```

IMC	Situação
<16	Subpeso Severo
16 a 19,9	Subpeso
20 a 24,9	Normal
25 a 29,9	Sobrepeso
30 a 39,9	Obeso
>40	Obeso Mórbido



Agora tente fazer o IMC de Pedro (peso: 85, altura: 1.69) e Maria (peso: 69, altura: 1.60).



Agora tente fazer o IMC desse pessoal aí:

Nome	Peso	Altura
Alice	65	1.60
Gilmar	95	1.78
Cecília	75	1.80
Bianca	77	1.68
Valentina	80	1.72
Augusto	68	1.65



Agora tente fazer o IMC desse pessoal aí:

Nome	Peso	Altura
Alice	65	1.60
Gilmar	95	1.78
Cecília	75	1.80
Bianca	77	1.68
Valentina	80	1.72
Augusto	68	1.65

Vamos usar um tipo de **objeto** chamado **vetor**

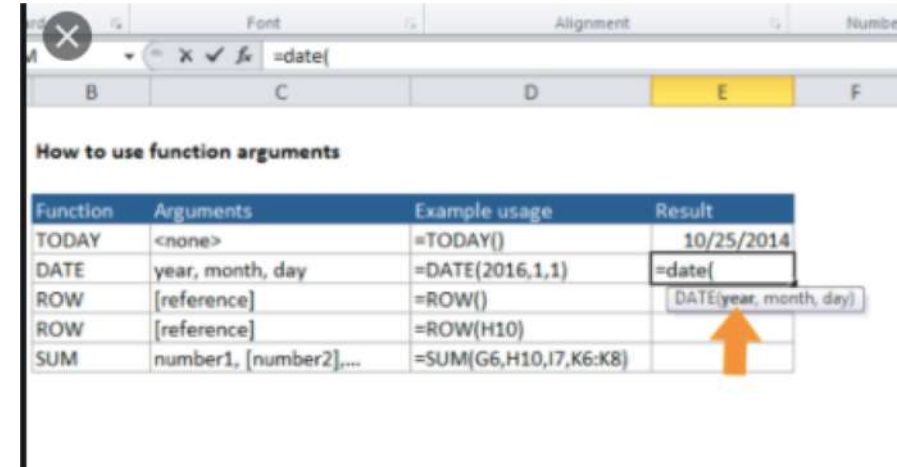
```
# trabalhando com vetores
pesos <- c(65, 95, 75, 77, 80, 68)
alturas <- c(1.60, 1.78, 1.80, 1.68, 1.72, 1.65)
imc <- pesos/alturas**2
imc
```

```
help(round)
```

```
round(imc, 2)
```

```
imc <- round(imc, 2) #estou sobrescrevendo um vetor  
#arredondado sobre ele mesmo
```

```
imc
```



Agora que entendemos o que é um vetor, vou apresentar outro tipo de **objeto** chamado **matriz**

```
Matriz<-cbind(pesos,alturas,imc)
```


```
Matriz
```

```
rownames(Matriz)<-c("Alice","Gilmar","Cecilia",  
                    "Bianca","Valentina","Augusto")
```

```
Matriz
```

$$A = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 2 & 1 & 1 & 1 \\ 2 & 3 & 0 & 1 \\ -1 & 1 & 2 & 2 \end{bmatrix}$$

Existe um objeto chamado **lista**, porém, exige um nível de abstração um pouco maior.

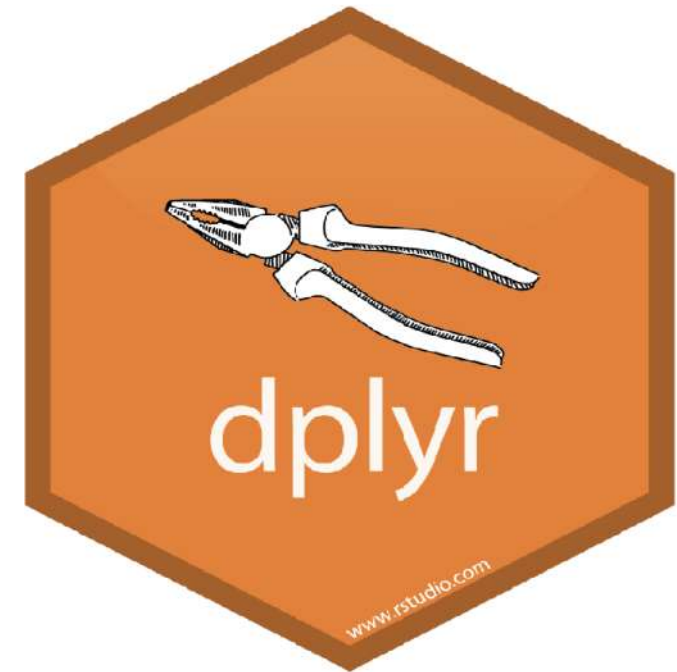
afinn	list [9] (S3: gg, ggplot)	List of length 9
data	list [48015 x 28] (S3: data.fra	A data.frame wit...
layers	list [1]	List of length 1
scales	environment [2] (S3: ScalesLis	<environment: 0...
mapping	list [2] (S3: uneval)	List of length 2
theme	list [0]	List of length 0
coordinates	environment [5] (S3: CoordCa	<environment: 0...
clip	character [1]	'on'
default	logical [1]	TRUE
expand	logical [1]	TRUE
limits	list [2]	List of length 2
super	function (S3: ggproto_methoc	function(...) { ... }
facet	environment [2] (S3: FacetNul	<environme... 
plot_env	environment [110]	<environment: R...
labels	list [2]	List of length 2

Agora vamos para o **dataframe**, um dos **objetos** mais importantes para a manipulação dos nossos dados.

Fonte dos dados: <https://cran.r-project.org/web/packages/gapminder/index.html>

Manipulando dados

- Pertence ao conjunto tidyverse
- Possui um conjunto de funções que permite manipular um dataframe de modo eficiente e intuitivo
- Select, filter, group_by, arrange, top_n, mutate, summarise, join



Instalando o dplyr

- Você pode instalar o dplyr unicamente ou o tidyverse que automaticamente insere todos os pacotes do conjunto

```
# Instalando
install.packages("tidyverse")
library(tidyverse)

install.packages("dplyr")
library(dplyr)
```

O R consegue ler arquivos das mais variadas extensões. Mas uma das formas mais recomendadas é o csv (*comma separated values*).

Como adicionar a arquivos basePaíses.csv e paísesIDH.csv?

Se você estiver usando **Rstudio Cloud**

Studio Cloud

Spaces

Your Workspace

New Space

Learn

Guide

What's New

Primers

Cheat Sheets

Feedback and Questions

Info

Terms and Conditions

System Status

Your Workspace / Untitled Project

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

Addins

Run Source

Source on Save

1:1 (Top Level)

R Script

Console Terminal Jobs

/cloud/project/RBasico/data/

Environment History Connections

Import Dataset

Global Environment

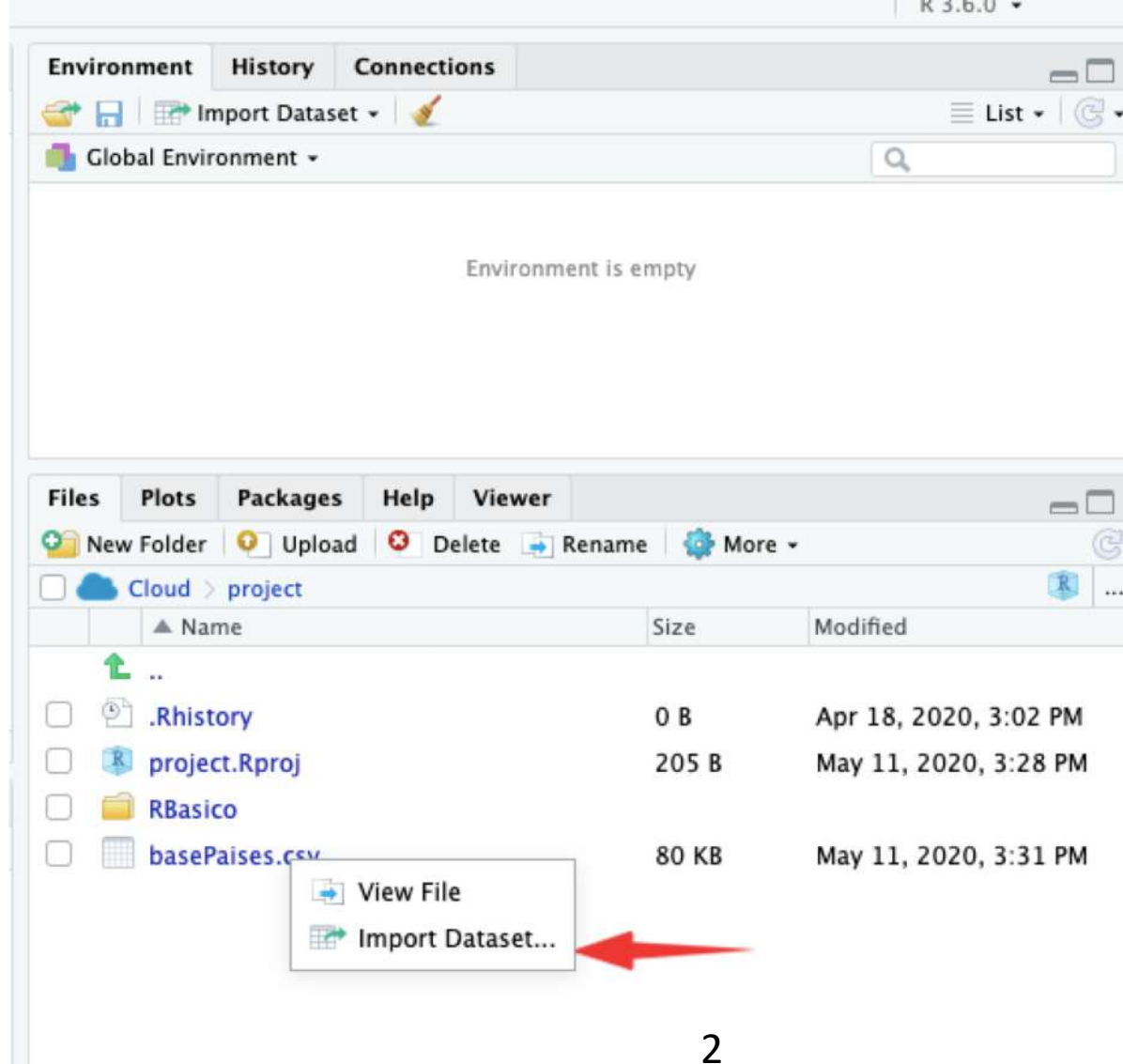
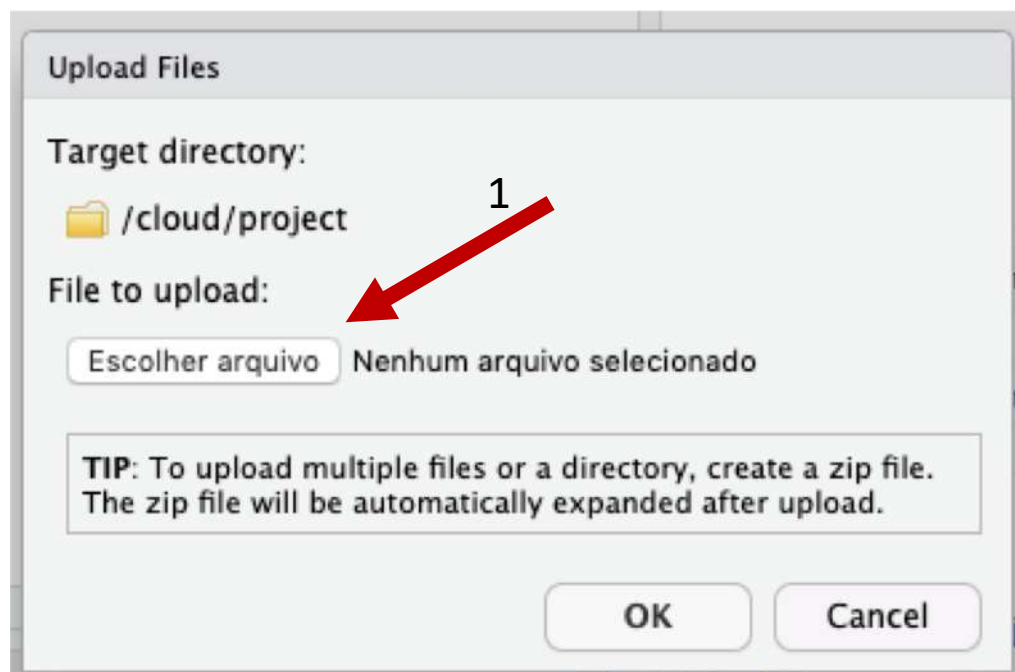
Environment is empty

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Cloud project

	Name	Size	Modified
	..		
	.Rhistory	0 B	Apr 18, 2020, 3:02 PM
	project.Rproj	205 B	May 11, 2020, 3:28 PM
	RBasico		



File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins R 3.6.0

Import Text Data

File/URL: /cloud/project/basePaises.csv Update

Data Preview:

country (character)	continent (character)	year (double)	lifeExp (double)	pop (double)	gdpPercap (double)
Afghanistan	Asia	1952	28.801	8425333	779.4453
Afghanistan	Asia	1957	30.332	9240934	820.8530
Afghanistan	Asia	1962	31.997	10267083	853.1007
Afghanistan	Asia	1967	34.020	11537966	836.1971
Afghanistan	Asia	1972	36.088	13079460	739.9811
Afghanistan	Asia	1977	38.438	14880372	786.1134
Afghanistan	Asia	1982	39.854	12881816	978.0114
Afghanistan	Asia	1987	40.822	13867957	852.3959
Afghanistan	Asia	1992	41.674	16317921	649.3414
Afghanistan	Asia	1997	41.763	22227415	635.3414

Previewing first 50 entries.

Import Options:

Name: basePaises ☒ First Row as Names Delimiter: **Tab** Escape: None

Skip: 0 ☒ Trim Spaces Quotes: Default Comment: Default

☒ Open Data Viewer Locale: Configure... NA: Default

Code Preview:

```
library(readr)
basePaises <- read_delim("/cloud/project/basePaises.csv",
  "\t", escape_double = FALSE,
  trim_ws = TRUE)
View(basePaises)
```

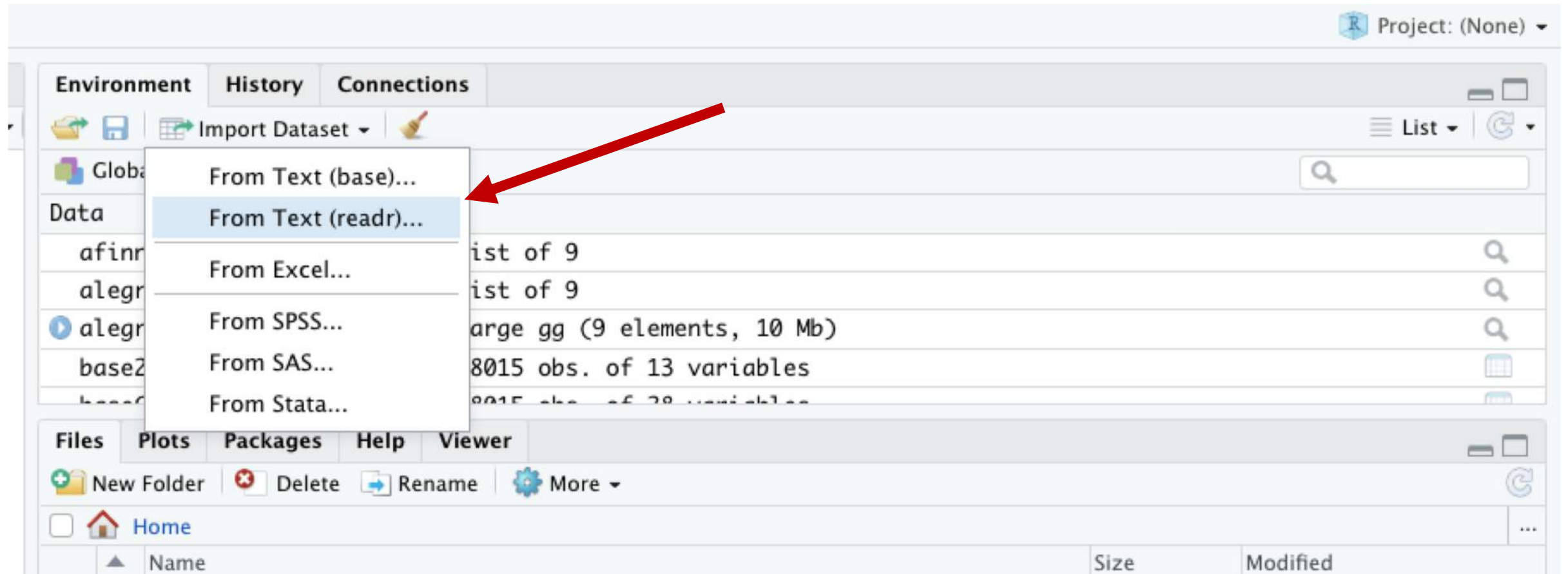
1 Mudar para tab

2

Import Cancel

? Reading rectangular data using readr

Para quem tiver com o Rstudio instalado no **computador**



Import Text Data

File/URL:

Browse...

1

Data Preview:

Aqui vai abrir uma janela na qual você deverá localizar seu arquivo

Mudar aqui para **tab**

2

Import Options:

Name: dataset

Skip: 0

☒ First Row as Names

☒ Trim Spaces

☒ Open Data Viewer

Delimiter:

Comma

Quotes:

Default

Locale:

Configure...

Escape:

None

Comment:

Default

NA:

Default

Code Preview:

```
library(readr)
dataset <- read_csv(NULL)
View(dataset)
```

? Reading rectangular data using readr

Import

Cancel

3

Pausa de 10 minutos

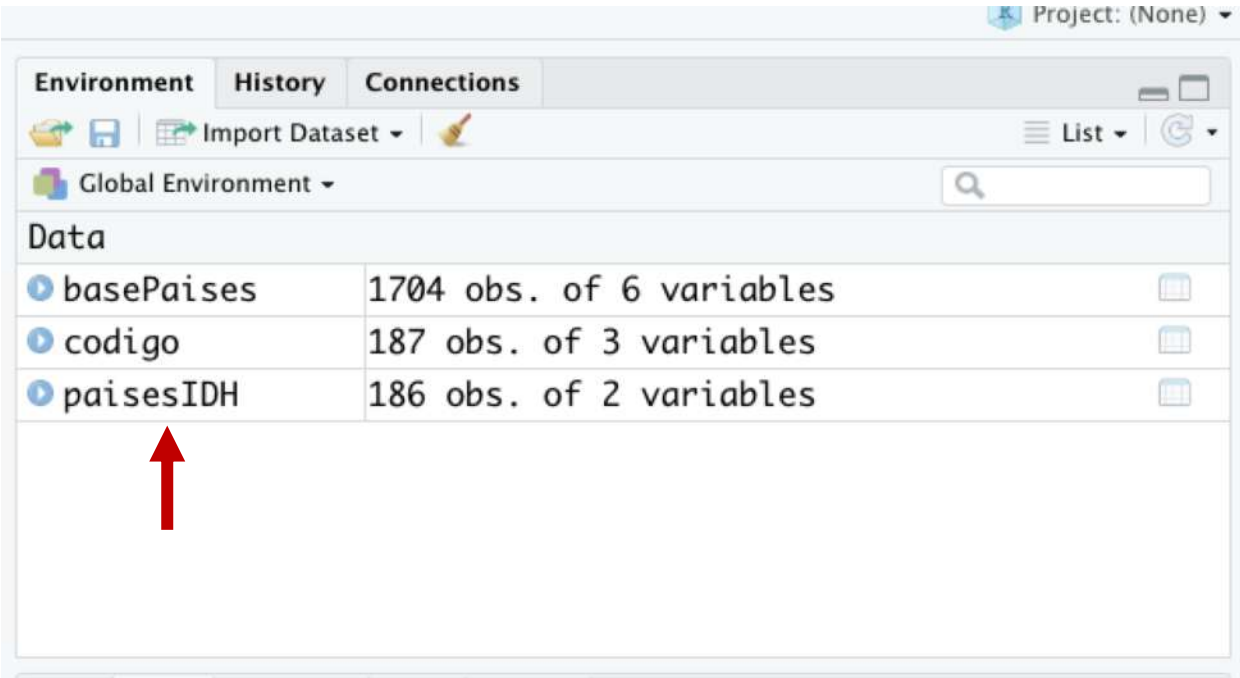
Inspeccionando a base e pacote dplyr

```
# caso não tenha conseguido subir a
#base, use o comando a seguir
1 install.packages("gapminder")
2 library(gapminder)
3 basePaises <- gapminder
4 codigo <- country_codes

5 str(basePaises)
6 head(basePaises)
7 tail(basePaises, n = 10)

8 glimpse(basePaises)
9 View(basePaises)

10 basePaises%>%
    distinct(continent)
```



Project: (None)

Environment History Connections

Import Dataset

Global Environment

Data

basePaises	1704 obs. of 6 variables
codigo	187 obs. of 3 variables
paisesIDH	186 obs. of 2 variables

Funções importantes: filter e select

E se eu quiser pegar só países que estão no continente Asiático?



```
1 basePaises%>%  
  filter(continent == "Asia")  
  
2 basePaises%>%  
  filter(continent == "Americas" & year>1990)  
  
3 basePaises%>%  
  filter(continent != "Oceania")  
  
# O mais legal é que você pode armazenar essas consultas em novos objetos  
# basta usar uma atribuição  
  
4 baseAsia <- basePaises%>%  
  filter(continent == "Asia")
```


Funções importantes: select

Vamos supor agora que eu precise só de quatro variáveis da minha tabela: ano, país e PIB per capita



```
1 basePaises%>%  
  select(year, country, gdpPercap)  
  
2 basePaises%>%  
  select(-lifeExp)  
  
3 basePaises%>%  
  filter(continent == "Americas" & year>1990)%>%  
  select(year, country, gdpPercap)
```

Funções importantes: mutate

E se eu quiser adicionar uma nova variável na tabela?
Como posso fazer?



```
1 basePaises <- basePaises%>%  
  mutate(GDP = gdpPercap * pop)  
  
2 base2007 <- basePaises%>%  
  filter(year == 2007) %>%  
  mutate(porte = if_else(pop>median(pop), "G", "P"))  
  
3 base2007<-base2007%>%  
  mutate(classGPC = case_when(  
    gdpPercap < 1625 ~ "Baixo",  
    gdpPercap >= 1625 & gdpPercap <18008 ~ "Medio",  
    gdpPercap >= 18008 ~ "Alto")) # til
```

Funções importantes: group_by e summarize

```
1 base2007%>%  
  group_by(classGPC)%>%  
  count()  
  
2 basePaises%>%  
  group_by(country)%>%  
  summarize(meanLE=mean(lifeExp),meanPop=mean(pop),  
            meanGpc=mean(gdpPercap))  
  
3 baseContinentes <- basePaises %>%  
  group_by(continent,year)%>%  
  summarize(meanLE=mean(lifeExp),meanPop=mean(pop),  
            meanGpc=mean(gdpPercap))
```

Juntando duas bases: joins

```
#Adicione a base codigo e idh ao global environment

1 basePaisCod <- basePaises%>%
    inner_join(codigo,by="country")

2 baseCompleta <- basePaisCod %>%
    inner_join(paisesIDH,by=c("country"="paises"))

3 Paisesfora <- paisesIDH%>%
    anti_join(basePaises,by=c("paises"="country"))

4 Paisesfora <-base2007%>%
    anti_join(basePaises,by=c("country"="paises"))
```



Exercícios de manipulação

1. Filtre os países das Américas ou Europa
2. Filtre países que possuem uma população menor que 100 milhões de habitantes em 2007
3. Crie uma nova variável chamada *gdpRound* que é um arredondamento de 2 casas da variável *gdpPercap*
4. Faça um agrupamento por continente e ano e depois uma resumo considerando cálculos de média expectativa de vida, população e PIB per capta. Salvar o resultado em um novo dataframe chamado baseContinente
5. Filtre os anos de 1957 e 2007 para comparar o 20 maiores PIB dos dois anos. Faça o mesmo para identificar os 20 menores.

Pausa de 10 minutos

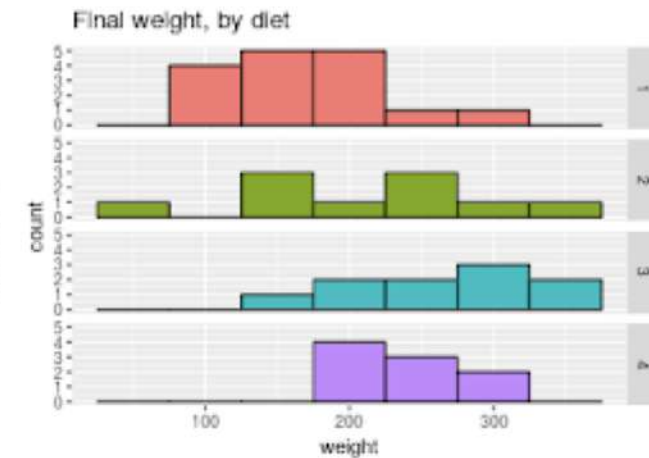
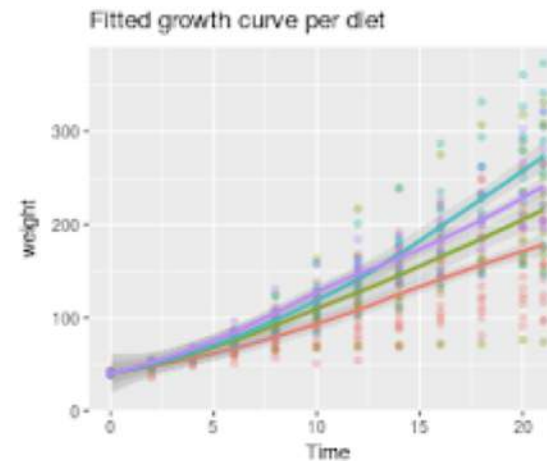
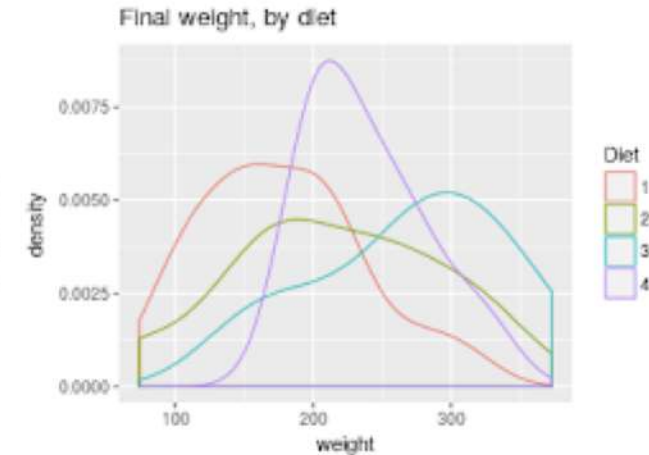
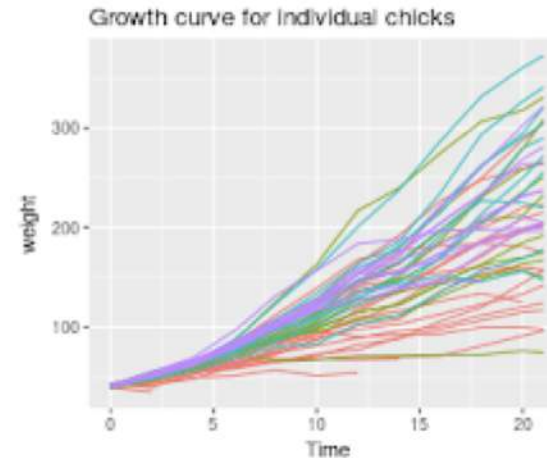
Vamos preparar algumas bases para nosso próximo tópico, visualização de dados

```
1 Base2007 <- basePaises%>%  
    filter(year==2007)  
  
2 baseProx <-basePaises%>%  
    filter(country=="Brazil" | country == "Chile" |  
    country == "Argentina" | country == "Uruguay")
```

Usando o ggplot2

- O ggplot2 é um dos pacotes de visualização mais populares do R
- Camadas
- Alto grau de customização

```
install.packages("ggplot2")  
library(ggplot2)
```



O ggplot2 funciona por camadas. A primeira camada possui a seguinte estrutura:

`ggplot(nome da base, aes(x = variável do eixo x, y = variável do eixo y)) +` ... (demais configurações/customizações)

```
1  grafico1<-ggplot(base2007,aes(x=continent,y=lifeExp))
2  ggplot(base2007,aes(x=continent,y=lifeExp)) + geom_boxplot()
3  ggplot(base2007,aes(x=continent,y=lifeExp)) + geom_point()
4  ggplot(base2007,aes(x=continent,y=lifeExp)) + geom_violin()

5  ggplot(base2007,aes(x=continent, y=lifeExp))+geom_boxplot()+
    ggtitle("Boxplot da expectativa de vida")+xlab("Continentes")+
    ylab("Expectativa de vida")

6  ggplot(baseProx,aes(x=year, y=gdpPercap))+geom_point()
7  ggplot(baseProx,aes(x=year, y=gdpPercap))+geom_line()

8  ggplot(baseProx,aes(x=year, y=gdpPercap, col = country))+geom_point()
9  ggplot(baseProx,aes(x=year, y=gdpPercap, col = country))+geom_line()
10 ggplot(baseProx,aes(x=year, y=gdpPercap, col = country))+geom_line(size=1.2)
```

Vamos visualizar agora a associação entre duas variáveis quantitativas: PIB per capita e expectativa de vida

```
1 ggplot(base2007,aes(x=gdpPercap,y=lifeExp))+ geom_point()  
2 ggplot(base2007,aes(x=gdpPercap,y=lifeExp))+ geom_point() + geom_smooth()  
3 ggplot(base2007,aes(x=gdpPercap,y=lifeExp,col=continent))+ geom_point()  
4 ggplot(base2007,aes(x=gdpPercap,y=lifeExp,col=continent,size=pop))+ geom_point()  
5 ggplot(base2007,aes(x=gdpPercap,y=lifeExp,color=continent))+ geom_point() +  
  geom_smooth(method='lm')  
7 ggplot(base2007,aes(x=gdpPercap,y=lifeExp,col=continent))+ geom_point() +  
  geom_smooth(method='lm') + facet_wrap(~continent)
```

Vamos visualizar o histograma da expectativa de vida

```
1 ggplot(base2007,aes(x=lifeExp))+ geom_histogram()  
2 ggplot(base2007,aes(x=lifeExp))+ geom_density()  
  
3 ggplot(base2007,aes(x=lifeExp, col=continent))+ geom_density()  
4 ggplot(base2007,aes(x=lifeExp, col=continent, fill = continent))+  
  geom_density()  
5 ggplot(base2007,aes(x=lifeExp, col=continent, fill = continent))+  
  geom_density(alpha=0.5)  
  
6 ggplot(base2007,aes(x=lifeExp, col=continent))+ geom_density(alpha=0.5)+  
  facet_wrap(~continent)  
  
7 ggplot(base2007,aes(x=lifeExp, fill=continent))+ geom_histogram(alpha=0.5)+  
  facet_wrap(~continent)  
  
8 base2007%>%  
  filter(continent!="Oceania")%>%  
  ggplot(aes(x=lifeExp, col=continent, fill = continent))+  
  geom_density(alpha=0.5)
```

Dicas a mais

- Aprender R é como aprender um novo idioma: exige prática constante e quanto mais você estuda, mais vocabulário você tem
- O R possui muitas funcionalidades. Enquanto preparava esse treinamento surgiram outros conteúdos que podem ser explorados: listas, funções, estruturas condicionais, conectando tabelas, manipulando estruturas de dados, etc
- O R possui muitas aplicações na ciências sociais aplicadas, tanto na perspectiva acadêmica, quanto de mercado.

