

# Introduction to R for Biologists

Intro to R and the Tidyverse Ecosystem

Developed by Rachael Cox and Alex Lukasiewicz

# Welcome to Class!!



**Alexandra Lukasiewicz**  
(they/them)  
PhD Candidate  
Contreras Lab

# Course Format

## **Concept Slides and Code Along Activities**

- **Lecture -> Coding -> Lecture -> Coding**
- Check email for link to my Github
  - Download zip file for this workshop
  - [https://github.com/ajlukasiewicz/Intro\\_to\\_R\\_Workshop/](https://github.com/ajlukasiewicz/Intro_to_R_Workshop/)

# Course Format

 **Intro\_to\_R\_Workshop** Public

 main  1 Branch  0 Tags

Go to file  Add file  Code

**ajlukasiewicz** Add files via upload

 Intro_R_markdown.Rmd	Add files via upload
 README.md	Create README.md
 data-transformation_cheatsheet.pdf	Add files via upload
 frog_apms_dnai2.csv	Add files via upload
 frog_apms_heatr2.csv	Add files via upload
 frog_enog_annotations.csv	Add files via upload
 mushrooms.csv	Add files via upload
 rmarkdown-cheatsheet.pdf	Add files via upload

**Local** **Codespaces**

 **Clone** 

**HTTPS** **SSH** **GitHub CLI**

[https://github.com/ajlukasiewicz/Intro\\_to\\_I](https://github.com/ajlukasiewicz/Intro_to_I) 

Clone using the web URL.

 **Open with GitHub Desktop**

 **Download ZIP** 

 **README**  

**Welcome to Intro to R for Biologists**

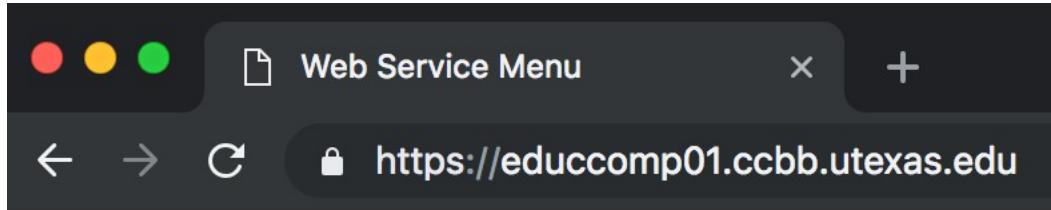
# Workshop Outline

1. How to get set up using R (On the POD and Locally)
2. How and why to use RStudio & R Markdown (.Rmd)
3. Basics of programming
  - Data types
  - Functions
  - Troubleshooting
4. Intro to the Tidyverse
  - Tidy vs untidy data
  - Tidyverse-specific functions

# Access R Studio through your web browser

1. <https://gsafcomp01.ccbb.utexas.edu/>
2. <https://gsafcomp02.ccbb.utexas.edu/>

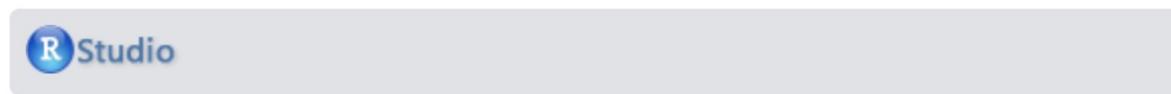
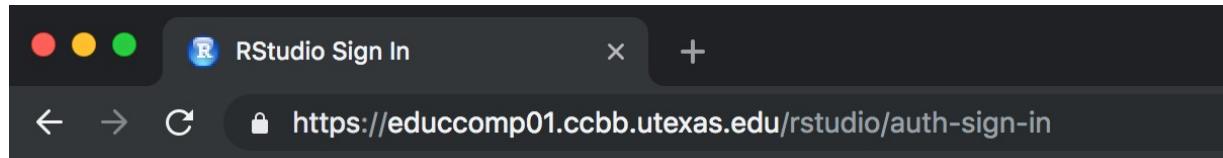
# Select RStudio



Please choose one of the following applications:

- [RStudio](#) ←—————
- [Jupyterhub](#)

# Sign in with your student# and password



Refer to “Student Accounts” file for your username

A close-up of the RStudio sign-in form. A red arrow points to the "Username:" field, which is circled in red. The form includes fields for "Username", "Password", a "Stay signed in" checkbox, and a "Sign In" button.

# R: The premier data analysis and visualization platform

Step 1: install R

<https://cran.r-project.org/>



## The Comprehensive R Archive Network

### Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

### Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper

# R Studio: A nice user interface for R

## Step 2: Install RStudio

<https://www.rstudio.com/products/rstudio/download/>



The screenshot shows the R Studio interface. On the left, the code editor displays an R Markdown file named 'class1.Rmd'. The code includes R global options, library imports (knitr), and a header for an in-class worksheet. It also contains a date entry and two sections of explanatory text. The middle pane, titled 'Environment', shows that the global environment is currently empty. At the bottom right, there's a 'R Resources' sidebar with links to various R-related support and documentation sites.

Code in class1.Rmd:

```
1  ```{r global_options, include=FALSE}
2  library(knitr)
3  opts_chunk$set(fig.align="center", fig.height=4, fig.width=4)
4  ```
5  ##In-class worksheet 1
6
7  **Jan 17, 2017**
8
9
10 Much of the work in this class will be done via R Markdown
11 documents. R Markdown documents are documents that combine text, R
12 code, and R output, including figures. They are a great way to
13 produce self-contained and documented statistical analyses.
14 In this first worksheet, you will learn how to do some basic
15 markdown editing. After you have made a change to the document,
press "Knit HTML" in R Studio and see what kind of a result you
get.
16
17 Edit only below this line.
```

Environment is empty

R Resources

- Learning R Online
- CRAN Task Views
- R on StackOverflow
- Getting Help with R

RStudio

- RStudio IDE Support
- RStudio Cheat Sheets
- RStudio Tip of the Day
- RStudio Packages
- RStudio Products

# Note about local installation

**Installing R locally will require you to install additional packages, such as "Tidyverse" and "DESeq"**

**I will add the installation code for these in the markdown files**

# R Markdown

# R Markdown: Open the markdown

The screenshot shows the RStudio interface running in a web browser at [gsafcomp01.ccbb.utexas.edu/rstudio/](https://gsafcomp01.ccbb.utexas.edu/rstudio/). The browser's address bar and various tabs are visible at the top.

**Console Pane:**

```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(ggthemes)
Session restored from your saved work on 2023-May-25 21:29:05 UTC (3 hours ago)
> |
```

**Environment Pane:**

Environment is empty

**Files Pane:**

Name	Size	Modified
R		

# R Markdown: Open the markdown

The screenshot shows the RStudio interface running in a web browser. The top navigation bar includes links for Function reference, Binary classification, Keitz Chamber, Morpheus, UT Account Information, Grep, Learn to purrr, Bike Rides in Texas, and Other Bookmarks. The main menu bar has options for File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The left sidebar contains tabs for Console, Terminal, and Background Jobs, showing R version 3.6.3 startup messages. The right sidebar contains tabs for Environment, History, Connections, and Tutorial, with the Environment tab active. The Environment pane displays the Global Environment as empty. The bottom pane, titled 'Files', shows a list of files including 'Home' and 'R'. The 'Upload' button in the Files toolbar is highlighted with a red box.

gsafcomp01.ccbb.utexas.edu/rstudio/

Function reference... Binary classification Keitz Chamber Morpheus UT Account Information Grep Learn to purrr Bike Rides in Texas Other Bookmarks

File Edit Code View Plots Session Build Debug Profile Tools Help student48

Console Terminal Background Jobs

R 3.6.3 · ~/

R version 3.6.3 (2020-02-29) -- "Holding the Windsock"  
Copyright (C) 2020 The R Foundation for Statistical Computing  
Platform: x86\_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

> `library(ggthemes)`  
Session restored from your saved work on 2023-May-25 21:29:05 UTC (3 hours ago)  
>

Environment History Connections Tutorial

Import Dataset 170 MiB

Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

New Folder New Blank File Upload Delete Rename More

Home

Name	Size	Modified
R		

# R Markdown: Open the markdown

The screenshot shows the RStudio interface running in a web browser. The title bar indicates the URL is `gsafcomp01.ccbb.utexas.edu/rstudio/`. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top right corner shows a user profile for "student48" and a "Project: (None)" dropdown. The main workspace consists of several panes: a left pane with "Console", "Terminal", and "Background Jobs"; a top-right pane with tabs for "Environment", "History", "Connections", and "Tutorial"; and a bottom-right pane for file management. A central modal dialog box titled "Upload Files" is open, prompting for a "Target directory" (set to "~") and a "File to upload" (with a "Choose File" button and a message "No file chosen"). A tip at the bottom of the dialog says: "TIP: To upload multiple files or a directory, create a zip file. The zip file will be automatically expanded after upload." At the bottom of the dialog are "OK" and "Cancel" buttons. The console pane contains R startup messages, including the R version 3.6.3 license notice and a warning about no warranty. It also shows a command to load ggthemes and a message about session restoration.

```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(ggthemes)
Session restored from your saved work on 2023-May-25 21:29:05 UTC (3 hours ago)
>
```

# R Markdown: Open the markdown

The screenshot shows the RStudio interface running in a web browser window. The title bar indicates the URL is `gsafcomp01.ccbb.utexas.edu/rstudio/`. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The left sidebar has tabs for Console, Terminal, and Background Jobs, with the Console tab active. The main workspace shows R version 3.6.3 starting up. A modal dialog box titled "Upload Files" is open in the center. It has fields for "Target directory:" (set to "~") and "File to upload:" with a "Choose File" button highlighted by a red box. Below the file input field is a tip: "TIP: To upload multiple files or a directory, create a zip file. The zip file will be automatically expanded after upload." At the bottom of the dialog are "OK" and "Cancel" buttons. The right side of the screen shows the Environment, History, Connections, and Tutorial panes, and a Global Environment view.

R version 3.6.3 (2020-02-29) -- "Holding the Windsock"  
Copyright (C) 2020 The R Foundation for Statistical Computing  
Platform: x86\_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

```
> library(ggthemes)
Session restored from your saved work on 2023-May-25 21:29:05 UTC (3 hours ago)
>
```

# R Markdown: Open the markdown

## Demonstration Time!!

The screenshot shows the RStudio interface with the following details:

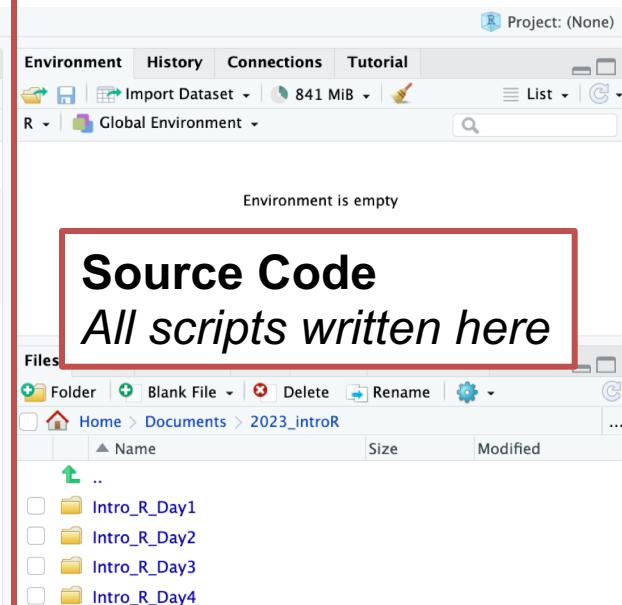
- Header Bar:** Shows the URL [gsafcomp01.cccb.utexas.edu/rstudio/](https://gsafcomp01.cccb.utexas.edu/rstudio/), a search bar, and various bookmarks.
- Toolbar:** Includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins menus.
- Console Tab:** Displays R version 3.6.3 startup messages and a command history.
- Environment Tab:** Shows the Global Environment with an empty list.
- File Menu:** Contains options like New Blank File, Upload, Delete, Rename, More, and a dropdown for Size and Modified.
- Central Area:** An "Upload Files" dialog box is open, prompting for a target directory and a file to upload. The "Choose File" button is highlighted with a red box.
- Bottom:** A command line interface with the prompt > |.

# R Markdown: Relevant Sections

```
## Developed by Rachael Cox
``{r global_options, include=FALSE}
library(knitr)
library(tidyverse)
opts_chunk$set(fig.align="center", fig.height=4, fig.width=4)
```
## Day 1: Introduction to R
### In-class worksheet
**May 31st, 2024**

Computational analyses require methods and notes to be recorded the same way you would for wet lab experiments. An excellent way to do this is via R Markdown documents. R Markdown documents are documents that combine text, R code, and R code output, and figures. They are a great way to produce self-contained and documented statistical analyses.

In this first worksheet, you will learn how to do some basic markdown editing in addition to the basic use of variables and functions in R. After you have made a change to the document, press "Knit HTML" in R Studio and see what kind of a result
```



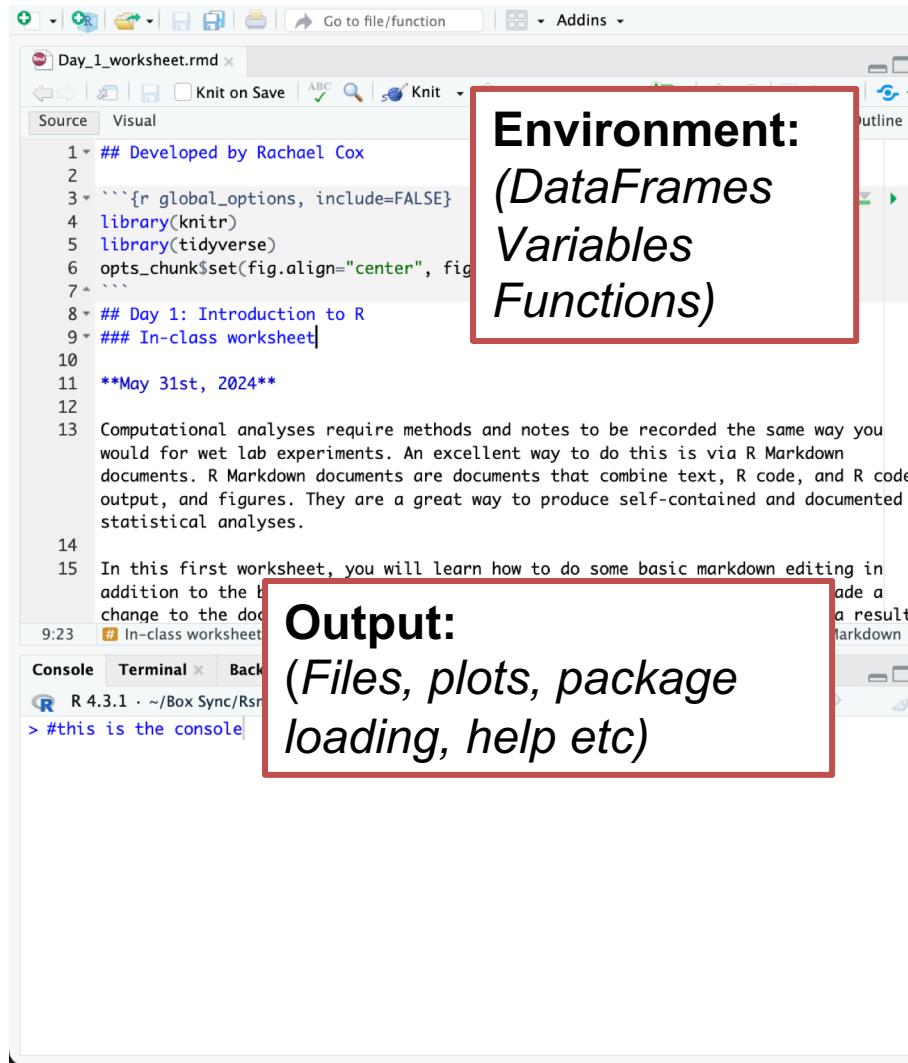
Console Terminal Background Jobs

R 4.3.1 · ~/Box Sync/RsmA biophysical model manuscript and documents/Other Pathogen Modeling/

```
> #this is the console
```

**Console**  
*Output and history  
of executed code*

# R Markdown: Relevant Sections



```
## Developed by Rachael Cox
```{r global_options, include=FALSE}
library(knitr)
library(tidyverse)
opts_chunk$set(fig.align="center", fig.width=5, fig.height=5)
```
## Day 1: Introduction to R
### In-class worksheet

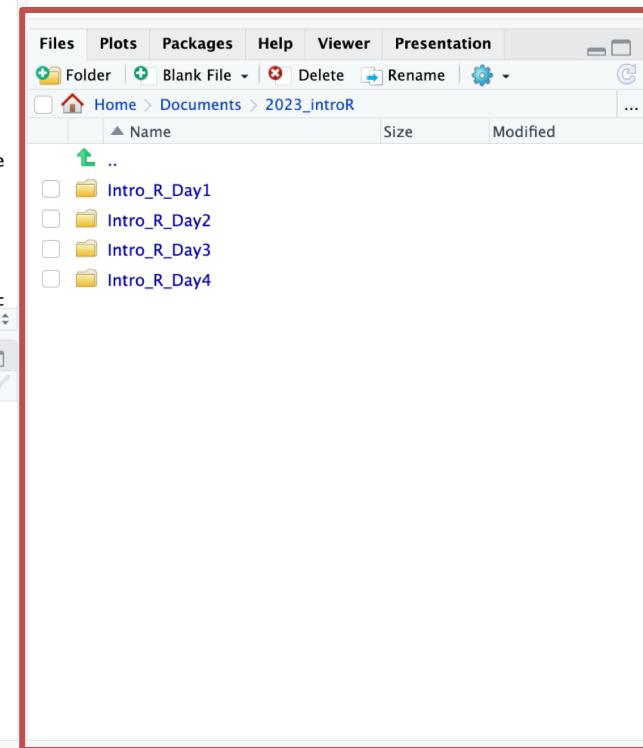
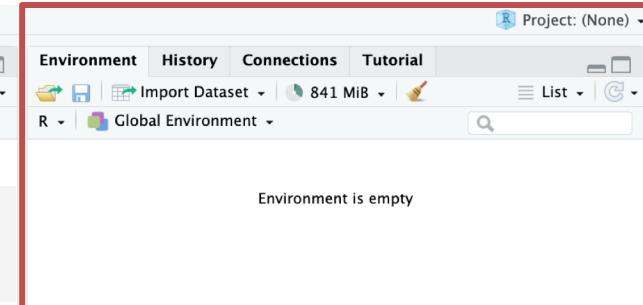
**May 31st, 2024**

Computational analyses require methods and notes to be recorded the same way you would for wet lab experiments. An excellent way to do this is via R Markdown documents. R Markdown documents are documents that combine text, R code, and R code output, and figures. They are a great way to produce self-contained and documented statistical analyses.

In this first worksheet, you will learn how to do some basic markdown editing in addition to the basic R code editing. You can change to the document tab to see how the changes appear in the document.

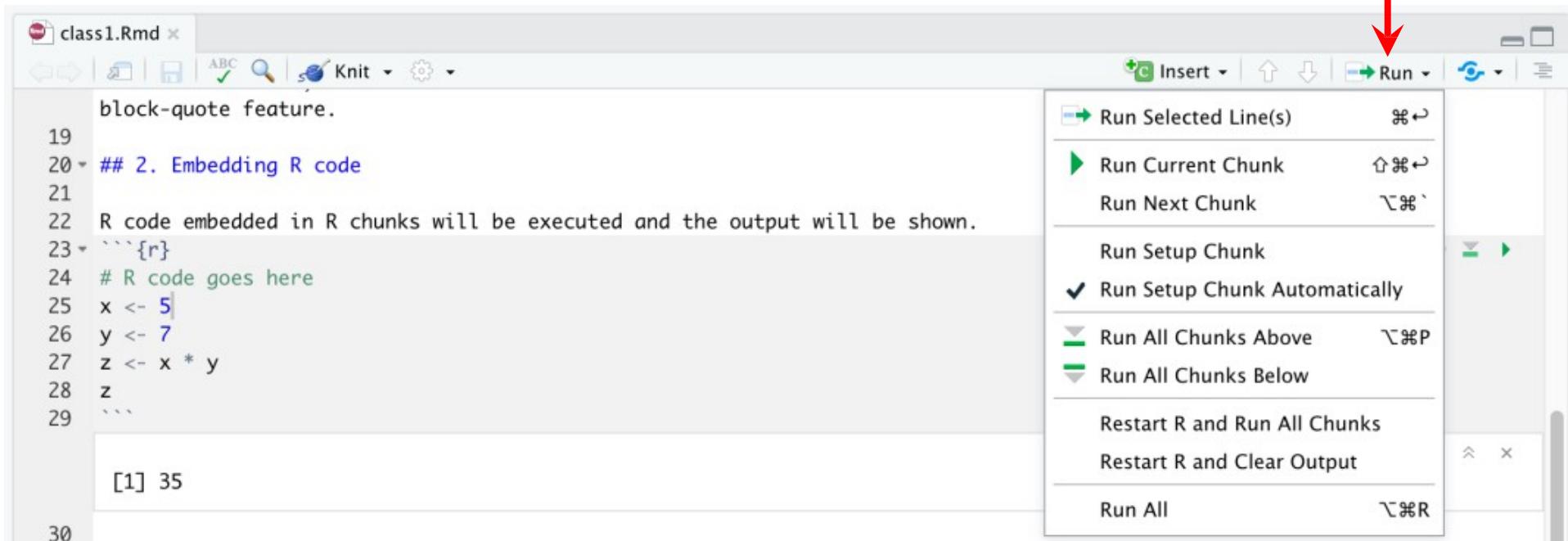
# Environment:
(DataFrames  
Variables  
Functions)
```

Output:  
(Files, plots, package loading, help etc)



# Different ways to execute code in RStudio

# Option 1: Press the “Run” button



## Option 2: Highlight code you want to execute and press ctrl+Enter (cmd+Enter on Macs)

R code embedded in R chunks will be executed and the output will be shown.

```
```{r}
# R code goes here
x <- 5
y <- 7
z <- x * y
z
...``
```



Console

Terminal ×

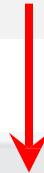
Jobs ×

~/Desktop/projects/ ↗

```
> x <- 5
> y <- 7
> z <- x * y
> z
[1] 35
> |
```

Option 3: Place pointer on line of code you want to execute, press ctrl+Enter (cmd+Enter on Macs)

```
R code embedded in R chunks will be executed and the output will be shown.  
```{r}  
# R code goes here  
x <- 5  
y <- 7  
z <- x * y  
z  
```
```



The screenshot shows an R console interface with three tabs: 'Console', 'Terminal x', and 'Jobs x'. The 'Console' tab is active, displaying the path ' ~/Desktop/projects/' followed by a file icon. Below the path, two lines of R code are shown: '> z <- x \* y' and '> |'. A vertical cursor is positioned at the end of the second line. A red arrow from the previous image points to the center of the console window, indicating where the user should click to execute the selected line of code.

```
> z <- x * y  
> |
```

# Use **ctrl+Shift+Enter** (**cmd+Shift+Enter** on Macs) to execute an entire code chunk

```
R code embedded in R chunks will be executed and the output will be shown.  
```{r}  
# R code goes here  
x <- 5  
y <- 7  
z <- x * y  
z  
```
```



```
Console Terminal × Jobs ×  
~/Desktop/projects/ ↵  
> x <- 5  
> y <- 7  
> z <- x * y  
> z  
[1] 35  
> |
```

# Shortcuts for coding

- **Ctrl+Shift+C** (Cmd+Shift+C on Macs) will comment/uncomment a line or multiple lines
- **Tab** and **Shift+Tab** will indent and un-indent lines, respectively
- **Ctrl+Shift+M** (Cmd+Shift+M on Macs) produces a pipe operator `%>%` (will be used within tidyverse)

# R Programming Basics

# Variable assignments and objects

```
> x <- 5
```

Assign **number** 5 to **variable** x

```
> x
```

```
[1] 5
```

```
> 5*x^2+7
```

Calculate  $5 \cdot x^2 + 7$

```
[1] 132
```

```
> y <- c(1, 2, 3, 4, 5)
```

Create object (**a vector**),  
assign to **variable** y

```
> y
```

```
[1] 1 2 3 4 5
```

```
> x*y
```

Multiply each element  
in **vector** y with the **number** in x

```
[1] 5 10 15 20 25
```

# Strings

A **string** contains text:

```
> name <- "Alex L"  
> name  
[1] "Alex L"
```

A **vector of strings**:

```
> animals <- c("cat", "mouse", "mouse",  
"cat", "rabbit")  
> animals  
[1] "cat"      "mouse"    "mouse"    "cat"  
"rabbit"
```

# Factors

**Factors** keep track of distinct categories (levels) in a vector:

```
> animals  
[1] "cat"      "mouse"    "mouse"    "cat"  
"rabbit"  
  
> factor(animals)  
[1] cat       mouse     mouse     cat       rabbit  
Levels: cat mouse rabbit
```

# Data frames

We use **data frames** to store data sets with multiple variables:

```
> pets <- data.frame(  
  family = c(1, 2, 3, 4, 5),  
  pet = animals  
)  
  
> pets  
family      pet  
1       1     cat  
2       2   mouse  
3       3   mouse  
4       4     cat  
5       5 rabbit
```

# Data frames

We access individual columns in a data frame with \$ + the column name:

```
> pets$family  
[1] 1 2 3 4 5
```

```
> pets$pet  
[1] cat      mouse    mouse    cat      rabbit  
Levels: cat mouse rabbit
```

# Demonstration Time!

Work on Section #1

# Data frames

R has many built-in data frames:

```
> cars
```

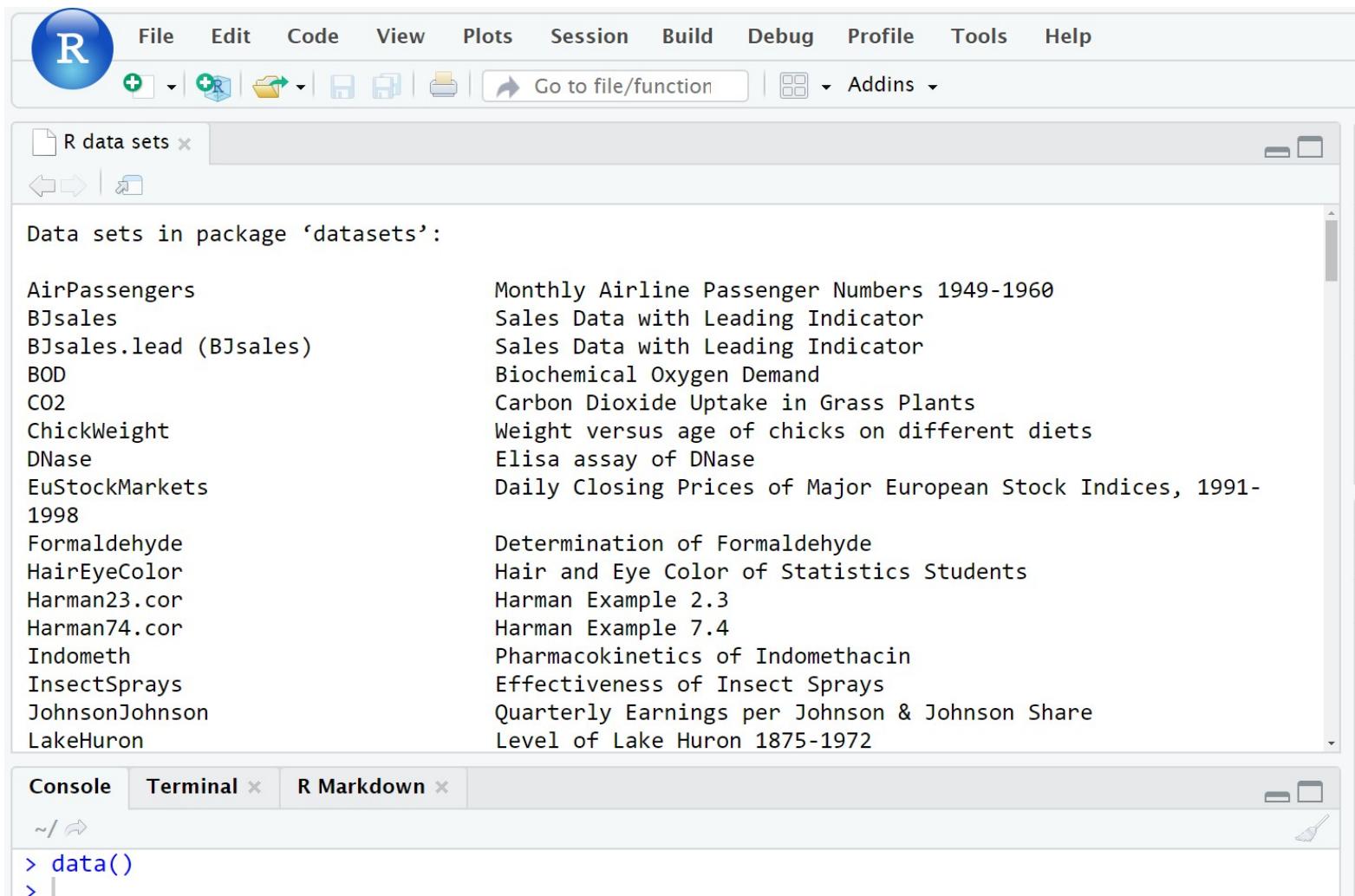
|   | speed | dist |
|---|-------|------|
| 1 | 4     | 2    |
| 2 | 4     | 10   |
| 3 | 7     | 4    |
| 4 | 7     | 22   |
| 5 | 8     | 16   |
| 6 | 9     | 10   |
| 7 | 10    | 18   |
| 8 | 10    | 26   |
| 9 | 10    | 34   |

```
> chickwts
```

|     | weight | feed      |
|-----|--------|-----------|
| 1   | 179    | horsebean |
| 2   | 160    | horsebean |
| 3   | 136    | horsebean |
| 4   | 227    | horsebean |
| ... | ...    | ...       |
| 11  | 309    | linseed   |
| 12  | 229    | linseed   |
| 13  | 181    | linseed   |
| 14  | 141    | linseed   |

# Data frames

Available built-in datasets can be accessed with `data()`



The screenshot shows the RStudio interface with the 'R data sets' browser panel open. The panel lists various built-in datasets from the 'datasets' package, each with a brief description. The 'Console' tab at the bottom shows the command `> data()` being run.

| Data set               | Description   |
|------------------------|---|
| AirPassengers          | Monthly Airline Passenger Numbers 1949-1960                 |
| BJSales                | Sales Data with Leading Indicator                           |
| BJSales.lead (BJSales) | Sales Data with Leading Indicator                           |
| BOD                    | Biochemical Oxygen Demand                                   |
| CO2                    | Carbon Dioxide Uptake in Grass Plants                       |
| ChickWeight            | Weight versus age of chicks on different diets              |
| DNase                  | Elisa assay of DNase  |
| EuStockMarkets         | Daily Closing Prices of Major European Stock Indices, 1991- |
| 1998                   | 1998  |
| Formaldehyde           | Determination of Formaldehyde                               |
| HairEyeColor           | Hair and Eye Color of Statistics Students                   |
| Harman23.cor           | Harman Example 2.3  |
| Harman74.cor           | Harman Example 7.4  |
| Indometh               | Pharmacokinetics of Indomethacin                            |
| InsectSprays           | Effectiveness of Insect Sprays                              |
| JohnsonJohnson         | Quarterly Earnings per Johnson & Johnson Share              |
| LakeHuron              | Level of Lake Huron 1875-1972                               |

```
> data()
```

# Data frames

Data set information can be accessed with `?dataset`

The screenshot shows the RStudio interface. On the left, the Environment pane lists various datasets: Quarterly Time Series of the Number of Australian Residents, beaver1 (beavers), Body Temperature Series of Two Beavers, beaver2 (beavers), Body Temperature Series of Two Beavers, cars, chickwts, co2, Mauna Loa Atmospheric CO2 Concentration, and crimtab. The 'cars' dataset is highlighted with a red circle and has a red arrow pointing from it towards the Help pane. In the Help pane, the 'Packages' tab is selected (also circled in red). The search bar shows 'R: Speed and Stopping Distances of Cars'. The main content area displays the documentation for the 'cars' dataset, which includes:

## Speed and Stopping Distances of Cars

### Description

The data give the speed of cars and the distances taken to stop. Note that the data were recorded in the 1920s.

### Usage

`cars`

### Format

A data frame with 50 observations on 2 variables.

[,1] speed numeric Speed (mph)  
[,2] dist numeric Stopping distance (ft)

### Source

Ezekiel, M. (1930) *Methods of Correlation Analysis*. Wiley.

```
> data()
> ?data
> ?cars
> |
```

# Data frames

The `head()` function shows the first few lines of a data frame:

```
> head(cars)
  speed dist
1     4     2
2     4    10
3     7     4
4     7    22
5     8    16
6     9    10
>
```

# Functions

Functions are called in the format `function(argument)`

```
> head(cars)
```

Function name

First argument



# Functions

Functions can have any number of required arguments or optional arguments

```
> head(cars, 8)
```

Function name      First argument  
(required)      Second argument  
(optional; default = 6)

# Functions

`head(cars, 8)` will show the first 8 lines of the data frame instead of the default 6:

```
> head(cars, 8)
```

|   | speed | dist |
|---|-------|------|
| 1 | 4     | 2    |
| 2 | 4     | 10   |
| 3 | 7     | 4    |
| 4 | 7     | 22   |
| 5 | 8     | 16   |
| 6 | 9     | 10   |
| 7 | 10    | 18   |
| 8 | 10    | 26   |

# Functions

More information about what a function does and/or requires can be accessed with `?function`

The screenshot shows the RStudio interface with the following details:

- Environment Tab:** Shows a data frame with columns `speed` and `dist`. The data is as follows:

|   | speed | dist |
|---|-------|------|
| 1 | 4     | 2    |
| 2 | 4     | 10   |
| 3 | 7     | 4    |
| 4 | 7     | 22   |
| 5 | 8     | 16   |
| 6 | 9     | 10   |
| 7 | 10    | 18   |
| 8 | 10    | 23   |

- Help Tab:** The `head` function is selected. The title is **Return the First or Last Part of an Object**.
- Description:** Returns the first or last parts of a vector, matrix, table, data frame or function. Since `head()` and `tail()` are generic functions.
- Usage:** The code for the `head` function is displayed.

```
head(x, ...)  
## Default S3 method:  
head(x, n = 6L, ...)  
## S3 method for class 'data.frame'  
head(x, n = 6L, ...)  
## S3 method for class 'matrix'  
head(x, n = 6L, ...)  
## S3 method for class 'ftable'  
head(x, n = 6L, ...)  
## S3 method for class 'table'  
head(x, n = 6L, ...)  
## S3 method for class 'function'  
head(x, n = 6L, ...)  
  
tail(x, ...)  
## Default S3 method:  
tail(x, n = 6L, ...)
```

- Console:** The command `> ?head` is highlighted with a red circle and has a red arrow pointing from it to the title of the help page.

# Functions

?function has argument information

The screenshot shows the RStudio interface with the following components:

- Environment View:** Displays a data frame with columns "speed" and "dist". The first seven rows are shown, with values: 1, 4; 2, 4; 3, 7; 4, 7; 5, 8; 6, 9; 7, 10.
- Help View:** The title bar says "R: Return the First or Last Part of an Object". The main content is the source code for the `tail` function, which is an S3 method for various classes. Below the code, the word "Arguments" is highlighted with a red oval.
- Console View:** Shows the command `> ?head` entered by the user.

The "Arguments" section of the help page is circled in red, and a red arrow points from the circled text in the console to this section.

**Arguments**

- x an object
- n a single integer. If positive, size for the resulting object: number of elements for a vector (including lists), rows for a matrix or data frame or lines for a function. If negative, all but the n last/first number of elements of x.
- addrownums if there are no row names, create them from the row numbers.
- ... arguments to be passed to or from other methods.

# Functions

We can implicitly or explicitly pass arguments

```
> head(cars, 8)
```

|  | speed | dist |
|--|-------|------|
|--|-------|------|

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
|---|---|---|

|   |   |    |
|---|---|----|
| 2 | 4 | 10 |
|---|---|----|

|   |   |   |
|---|---|---|
| 3 | 7 | 4 |
|---|---|---|

|   |   |    |
|---|---|----|
| 4 | 7 | 22 |
|---|---|----|

|   |   |    |
|---|---|----|
| 5 | 8 | 16 |
|---|---|----|

|   |   |    |
|---|---|----|
| 6 | 9 | 10 |
|---|---|----|

|   |    |    |
|---|----|----|
| 7 | 10 | 18 |
|---|----|----|

|   |    |    |
|---|----|----|
| 8 | 10 | 26 |
|---|----|----|

```
>
```

```
> head(x=cars, n=8)
```

|  | speed | dist |
|--|-------|------|
|--|-------|------|

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
|---|---|---|

|   |   |    |
|---|---|----|
| 2 | 4 | 10 |
|---|---|----|

|   |   |   |
|---|---|---|
| 3 | 7 | 4 |
|---|---|---|

|   |   |    |
|---|---|----|
| 4 | 7 | 22 |
|---|---|----|

|   |   |    |
|---|---|----|
| 5 | 8 | 16 |
|---|---|----|

|   |   |    |
|---|---|----|
| 6 | 9 | 10 |
|---|---|----|

|   |    |    |
|---|----|----|
| 7 | 10 | 18 |
|---|----|----|

|   |    |    |
|---|----|----|
| 8 | 10 | 26 |
|---|----|----|

```
>
```

# Reading and writing data

`write_csv(variable, "filename.csv")` is a package-specific function that allows you write R data frames to a comma-separated table

`read_csv("filename.csv")` is a package-specific function that allows you to import a file in your local environment:

```
> read_csv("mushrooms_tiny.csv")
```

\*note: if you are running the workbook locally and have not installed tidyverse use:  
`write.csv()` and `read.csv()` with the same arguments

# Demonstration Time!

Work on Sections 2 and 3

# Troubleshooting

# Ask RStudio for help

Type `?function` into console

The screenshot shows the RStudio interface with the 'Console' tab active, displaying the command `> ?t.test`. A red arrow points from the console output to the title 't.test {stats}' in the help documentation pane. The documentation pane includes tabs for 'Files', 'Plots', 'Packages', 'Help', and 'Viewer'. The 'Help' tab is selected, showing the topic 'R: Student's t-Test'. The page title is 'Student's t-Test'.

**t.test {stats}**

**Student's t-Test**

**Description**

Performs one and two sample t-tests on vectors of data.

**Usage**

```
t.test(x, ...)
```

## Default S3 method:

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

## S3 method for class 'formula'

```
t.test(formula, data, subset, na.action, ...)
```

# Ask Google for help

how do i run a t test in r

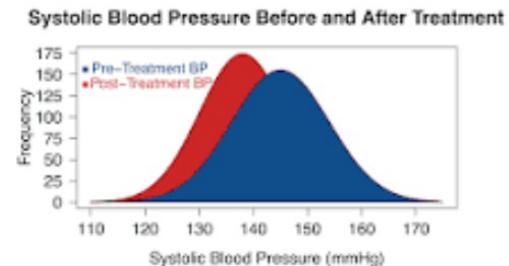
X |

All Videos Images News Shopping More Settings Tools

About 2,780,000,000 results (0.64 seconds)

How to Perform T-tests in R. To conduct a one-sample t-test in R, we use the syntax `t.test(y, mu = 0)` where x is the name of our variable of interest and mu is set equal to the mean specified by the null hypothesis.

Aug 17, 2015



[datascienceplus.com › t-tests](http://datascienceplus.com/t-tests/)

[How to Perform T-tests in R | DataScience+](#)

G Error in `t.test.default(x, y)` : not enough 'x' observations

Q Error in `t.test.default(x, y)` : not enough 'x' observations - Google Search

# Ask StackOverflow for help

stack**overflow** Products  Log in Sign up

Home PUBLIC  Stack Overflow Tags Users Jobs TEAMS What's this?

## Rotating and spacing axis labels in ggplot2

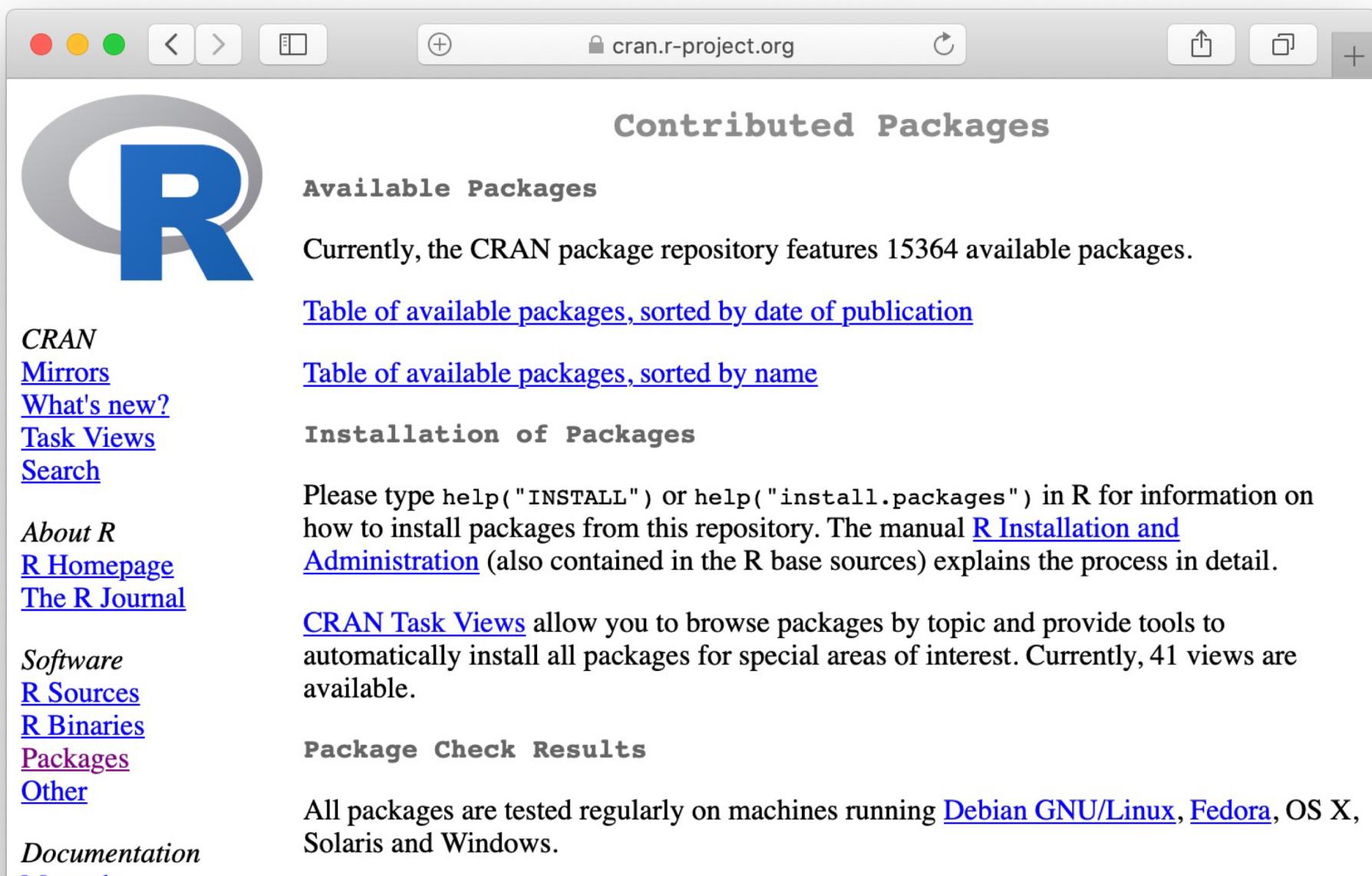
Asked 10 years, 10 months ago Active 18 days ago Viewed 768k times

 680 

I have a plot where the x-axis is a factor whose labels are long. While probably not an ideal visualization, for now I'd like to simply rotate these labels to be vertical. I've figured this part out with the code below, but as you can see, the labels aren't totally visible.

Extending R through packages:  
There's a package for everything

# R packages are available on CRAN (Comprehensive R Archive Network)



The screenshot shows a web browser window with the URL `cran.r-project.org` in the address bar. The page content is as follows:

**Contributed Packages**

**Available Packages**

Currently, the CRAN package repository features 15364 available packages.

[Table of available packages, sorted by date of publication](#)

[Table of available packages, sorted by name](#)

**Installation of Packages**

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository. The manual [R Installation and Administration](#) (also contained in the R base sources) explains the process in detail.

**CRAN Task Views** allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 41 views are available.

**Package Check Results**

All packages are tested regularly on machines running [Debian GNU/Linux](#), [Fedora](#), OS X, Solaris and Windows.

**CRAN**  
[Mirrors](#)  
[What's new?](#)  
[Task Views](#)  
[Search](#)

**About R**  
[R Homepage](#)  
[The R Journal](#)

**Software**  
[R Sources](#)  
[R Binaries](#)  
[Packages](#)  
[Other](#)

**Documentation**

# Bio-specific R packages are available on Bioconductor



Search:

[Home](#) [Install](#) [Help](#) [Developers](#) [About](#)

## About Bioconductor

*Bioconductor* provides tools for the analysis and comprehension of high-throughput genomic data.

*Bioconductor* uses the R statistical programming language, and is open source and open development. It has two releases each year, and an active user community. *Bioconductor* is also available as an [AMI](#) (Amazon Machine Image) and [Docker](#) images.

## News

- See our [google calendar](#) for events, conferences, meetings, forums, etc. Add your event with email to events at [bioconductor.org](#).
- *Bioconductor 3.11* is available.
- Nominate an outstanding community member for a *Bioconductor Award!* See the [support site](#) for more information.
- Registration open for [BioC2020](#).
- Core team **job opportunities** available, contact Martin.Morgan at RoswellPark.org
- *Bioconductor F1000 Research Channel* is

## BioC 2020

Get the latest updates on the [BioC 2020 Conference](#)!

- BioC 2020 is going virtual July 27 - July 31. Please see the [Registration Page](#) for more information.
- Nominate an outstanding *Bioconductor* community member for a *Bioconductor Award!* See [posting](#) for more information.
- Call for birds-of-feather, hack-a-thon, and how-to sections. Please see [posting](#) for more information.
- Registration is now open. [Register today](#).

## Install »

- Discover [1903 software packages](#) available in *Bioconductor* release 3.11.

Get started with *Bioconductor*

- [Install \*Bioconductor\*](#)
- [Get support](#)
- [Latest newsletter](#)
- [Follow us on twitter](#)
- [Install R](#)

## Learn »

Master *Bioconductor* tools

- [Courses](#)
- [Support site](#)
- [Package vignettes](#)
- [Literature citations](#)
- [Common work flows](#)
- [FAQ](#)
- [Community resources](#)
- [Videos](#)

Break Time  
Feel free to ask questions!

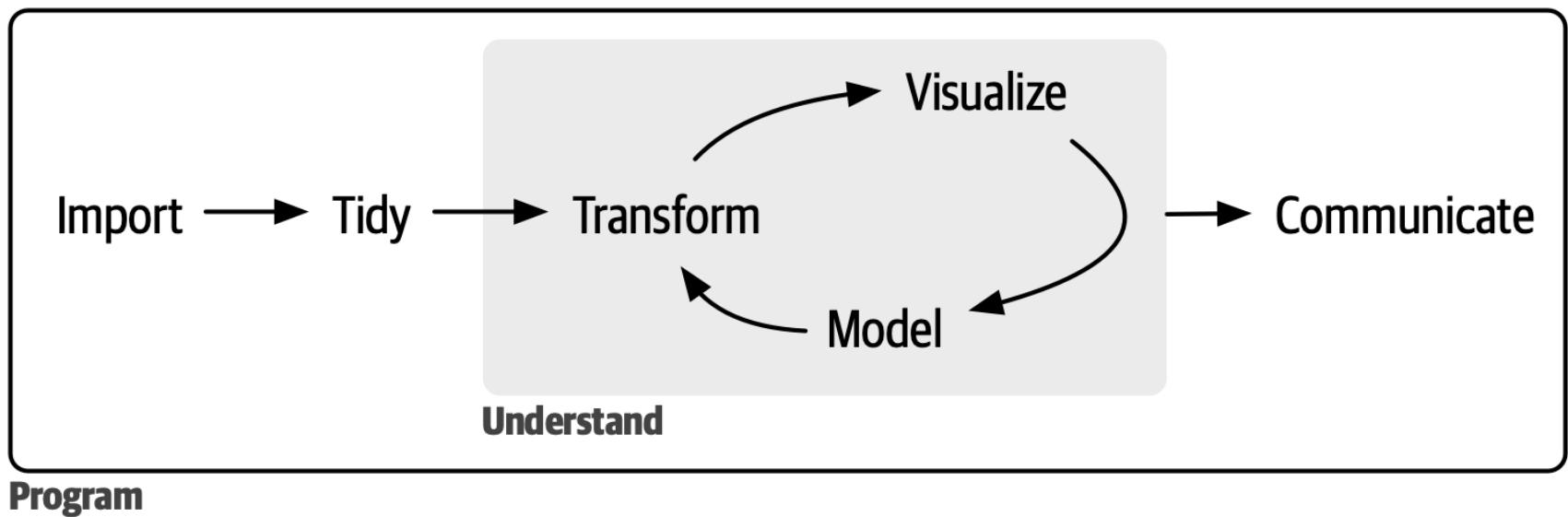
# Workshop Outline

1. How to get set up using R
2. How and why to use RStudio & R Markdown (.Rmd)
3. Basics of programming
  - Data types
  - Functions
  - Troubleshooting
4. Intro to the Tidyverse
  - Tidy vs untidy data
  - Tidyverse-specific functions

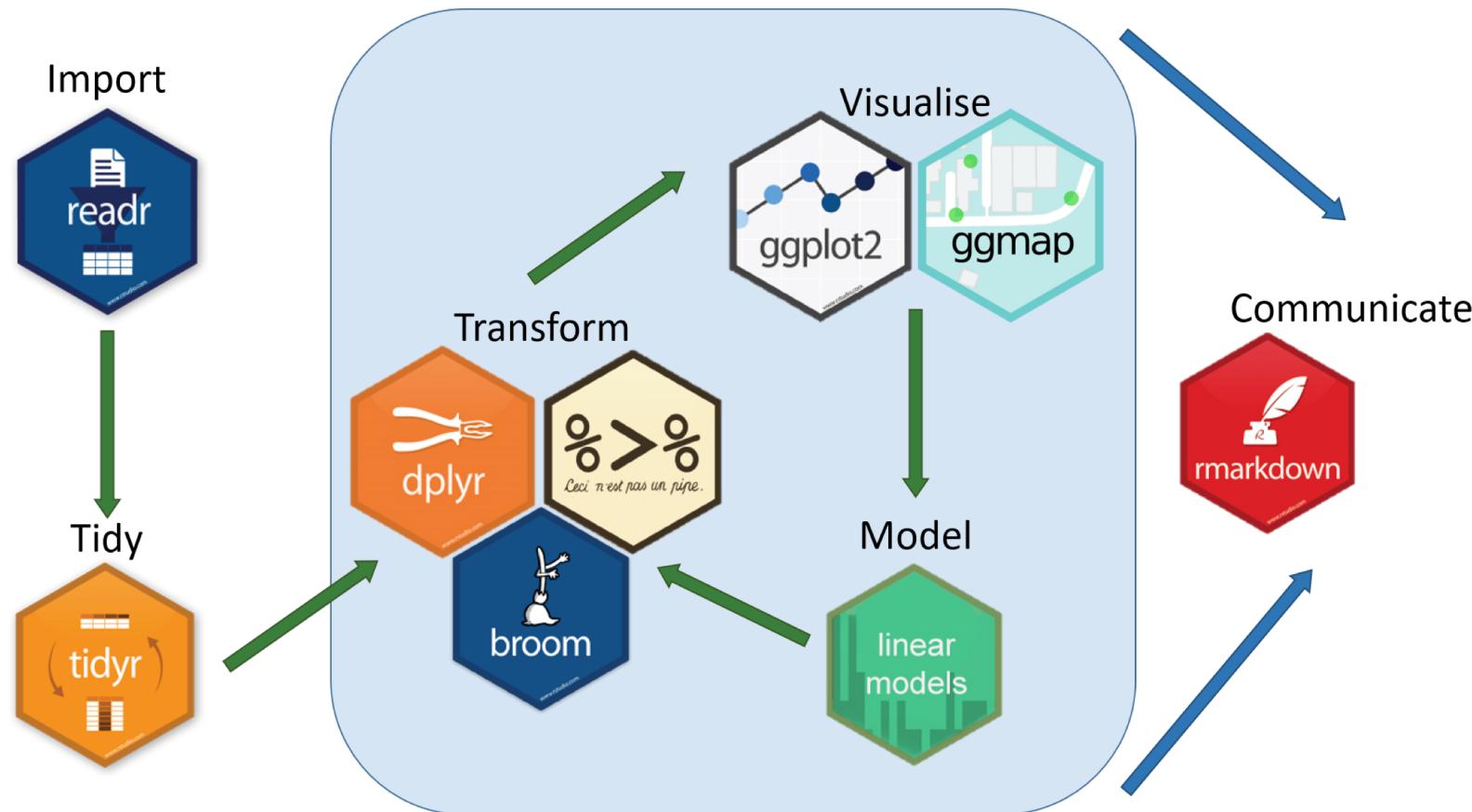
# Workshop Outline

1. How to get set up using R
2. How and why to use RStudio & R Markdown (.Rmd)
3. Basics of programming
4. Intro to the Tidyverse
  - Tidy vs untidy data
  - Tidyverse-specific functions

Especially in biology, we work with many different types of data which we collect, analyze, and communicate



In R, the collection of programs that make up the Tidyverse make this process easier and more reproducible



# Tidy data

“Tidy datasets are all alike but every messy dataset is messy in its own way” — Hadley Wickham

# Tidy data

| country     | year | cases | population |
|-------------|------|-------|------------|
| Afghanistan | 1999 | 745   | 16937071   |
| Afghanistan | 2000 | 2666  | 20595360   |
| Brazil      | 1999 | 3737  | 172006362  |
| Brazil      | 2000 | 80488 | 174504898  |
| China       | 1999 | 21258 | 1272015272 |
| China       | 2000 | 21366 | 128042583  |

| country     | year | cases | population |
|-------------|------|-------|------------|
| Afghanistan | 1999 | 745   | 16937071   |
| Afghanistan | 2000 | 2666  | 20595360   |
| Brazil      | 1999 | 3737  | 172006362  |
| Brazil      | 2000 | 80488 | 174504898  |
| China       | 1999 | 21258 | 1272015272 |
| China       | 2000 | 21366 | 128042583  |

| country     | year | cases | population |
|-------------|------|-------|------------|
| Afghanistan | 1999 | 745   | 16937071   |
| Afghanistan | 2000 | 2666  | 20595360   |
| Brazil      | 1999 | 3737  | 172006362  |
| Brazil      | 2000 | 80488 | 174504898  |
| China       | 1999 | 21258 | 1272015272 |
| China       | 2000 | 21366 | 128042583  |

Three rules:

1. Each variable has its own column
2. Each observation has its own row
3. Each value has its own cell

# Example: Contingency table

|                | <b>survived</b> | <b>died</b> |          |
|----------------|-----------------|-------------|----------|
| <b>drug</b>    | 15              | 3           | not tidy |
| <b>placebo</b> | 4               | 12          |          |

# Example: Contingency table

|                | <b>survived</b> | <b>died</b> | ← Y variable, outcome<br>not tidy |
|----------------|-----------------|-------------|-----------------------------------|
| <b>drug</b>    | 15              | 3           |                                   |
| <b>placebo</b> | 4               | 12          |                                   |

↑  
X variable,  
treatment

|      | X                | Y              | count |
|------|------------------|----------------|-------|
|      | <b>treatment</b> | <b>outcome</b> |       |
| tidy | drug             | survived       | 15    |
|      | drug             | died           | 3     |
|      | placebo          | survived       | 4     |
|      | placebo          | died           | 12    |

# Example: Contingency table, extended

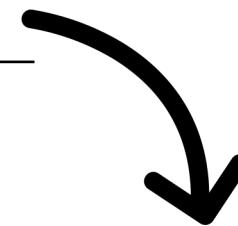
|                | <b>survived</b> | <b>died</b> |                 |
|----------------|-----------------|-------------|-----------------|
| <b>drug</b>    | 15              | 3           | <b>not tidy</b> |
| <b>placebo</b> | 4               | 12          |                 |

| tidy | <b>patient</b> | <b>treatment</b> | <b>outcome</b> |
|------|----------------|------------------|----------------|
|      |                |                  | survived       |
|      | 1              | drug             | survived       |
|      | 2              | drug             | died           |
|      | 3              | drug             | survived       |
|      | 4              | placebo          | died           |
|      |                | •                | •              |
|      |                | •                | •              |

# tidyverse library provides functions for transforming tables

|         | survived | died |
|---------|----------|------|
| drug    | 15       | 3    |
| placebo | 4        | 12   |

`pivot_longer()`



`pivot_wider()`

| patient | treatment | outcome  |
|---------|-----------|----------|
| 1       | drug      | survived |
| 2       | drug      | died     |
| 3       | drug      | survived |
| 4       | placebo   | died     |

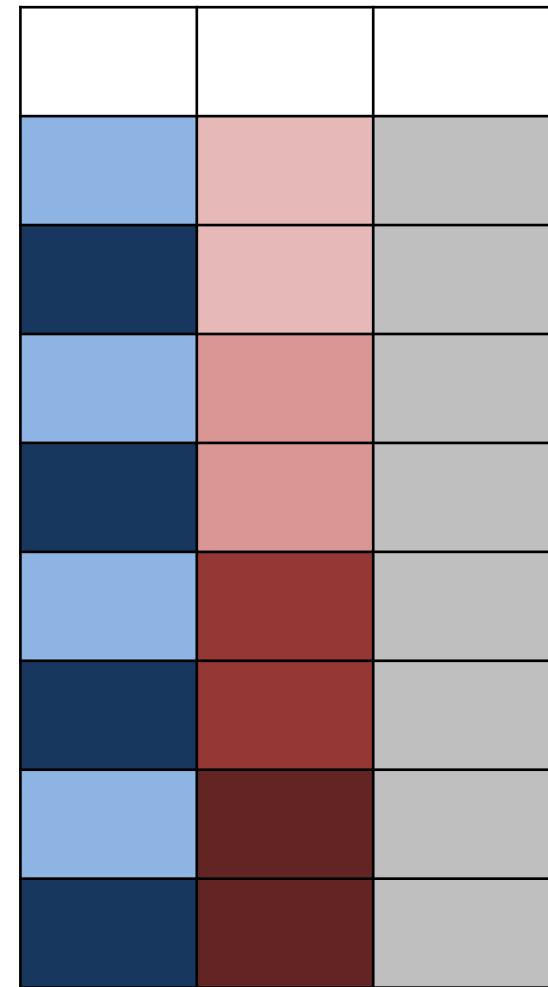
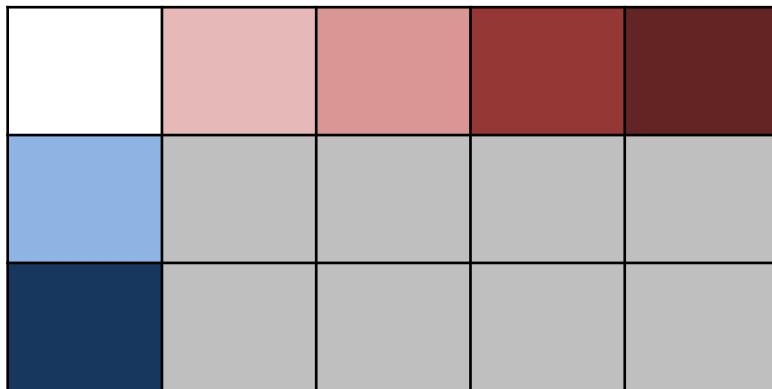
⋮  
⋮

# Making data sets longer or wider

We'll be discussing two functions:

- `pivot_longer()` — make a wide table long
- `pivot_wider()` — make a long table wide

## pivot\_longer()



# pivot\_longer()

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

```
data %>%  
  pivot_longer(cols, names_to = "A", values_to = "B")
```

# pivot\_longer()

|  |  |  |  |  |  |
|--|--|--|--|--|--|
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

↑  
↑  
↑  
↑  
**columns**



|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

```
data %>%  
  pivot_longer(cols, names_to = "A", values_to = "B")
```

# pivot\_longer()

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

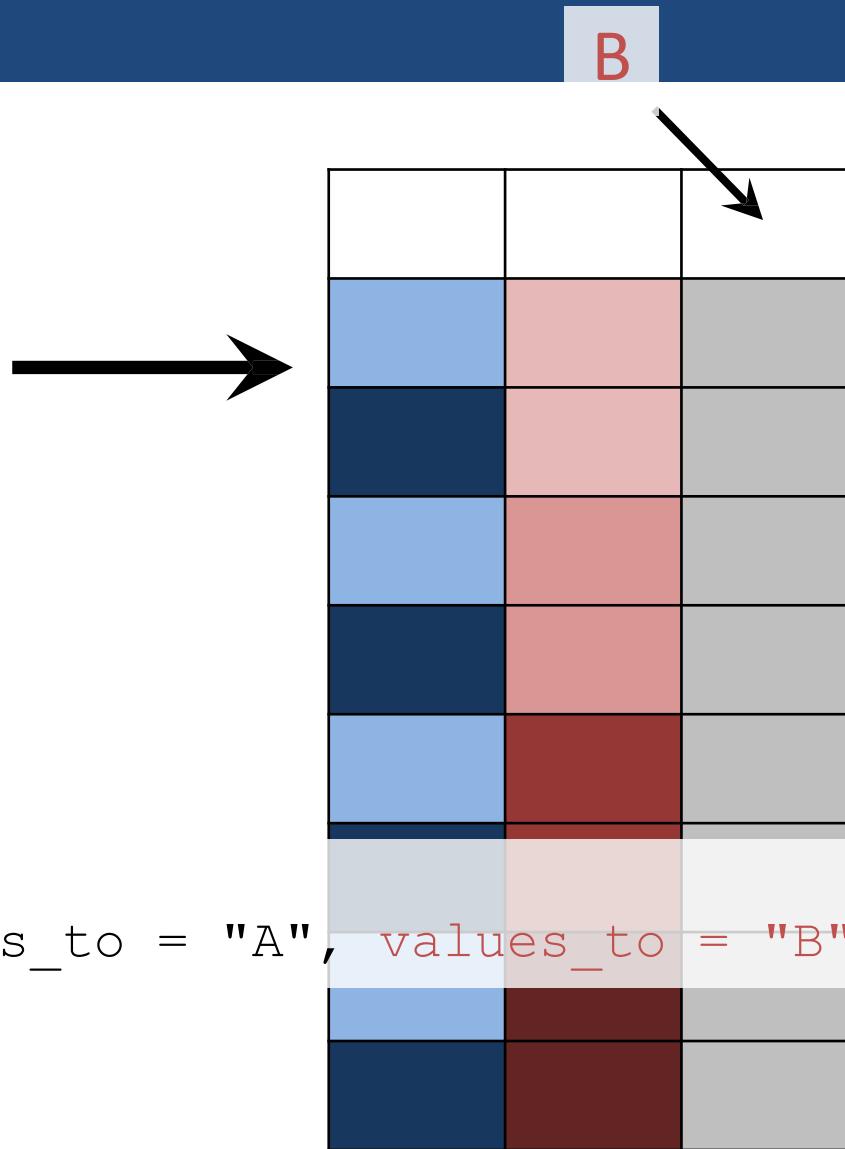
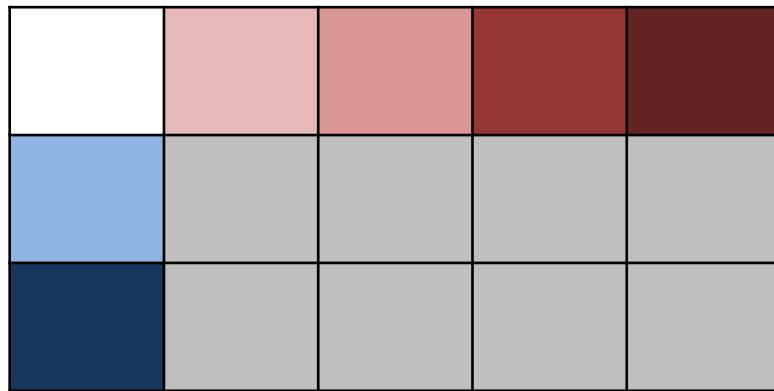


A

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

```
data %>%  
  pivot_longer(cols, names_to = "A", values_to = "B")
```

## pivot\_longer()



```
data %>%  
  pivot_longer(cols, names_to = "A", values_to = "B")
```

# Example dataset

```
> head(sitka_wide)
```

|   | tree | treat | t152 | t174 | t201 | t227 | t258 |
|---|------|-------|------|------|------|------|------|
| 1 | 1    | ozone | 4.51 | 4.98 | 5.41 | 5.90 | 6.15 |
| 2 | 2    | ozone | 4.24 | 4.20 | 4.68 | 4.92 | 4.96 |
| 3 | 3    | ozone | 3.98 | 4.36 | 4.79 | 4.99 | 5.03 |
| 4 | 4    | ozone | 4.36 | 4.77 | 5.10 | 5.30 | 5.36 |
| 5 | 5    | ozone | 4.34 | 4.95 | 5.42 | 5.97 | 6.28 |
| 6 | 6    | ozone | 4.59 | 5.08 | 5.36 | 5.76 | 6.00 |

Is this data tidy? Why or why not?

# Example: Tidying dataset using pivot\_longer()

```
> head(sitka_wide)
  tree treat t152 t174 t201 t227 t258
1   1 ozone 4.51 4.98 5.41 5.90 6.15
2   2 ozone 4.24 4.20 4.68 4.92 4.96
3   3 ozone 3.98 4.36 4.79 4.99 5.03
4   4 ozone 4.36 4.77 5.10 5.30 5.36
5   5 ozone 4.34 4.95 5.42 5.97 6.28
6   6 ozone 4.59 5.08 5.36 5.76 6.00
```

**Data Frame object**

sitka\_wide %>% **Pipe command (being "sent to")**

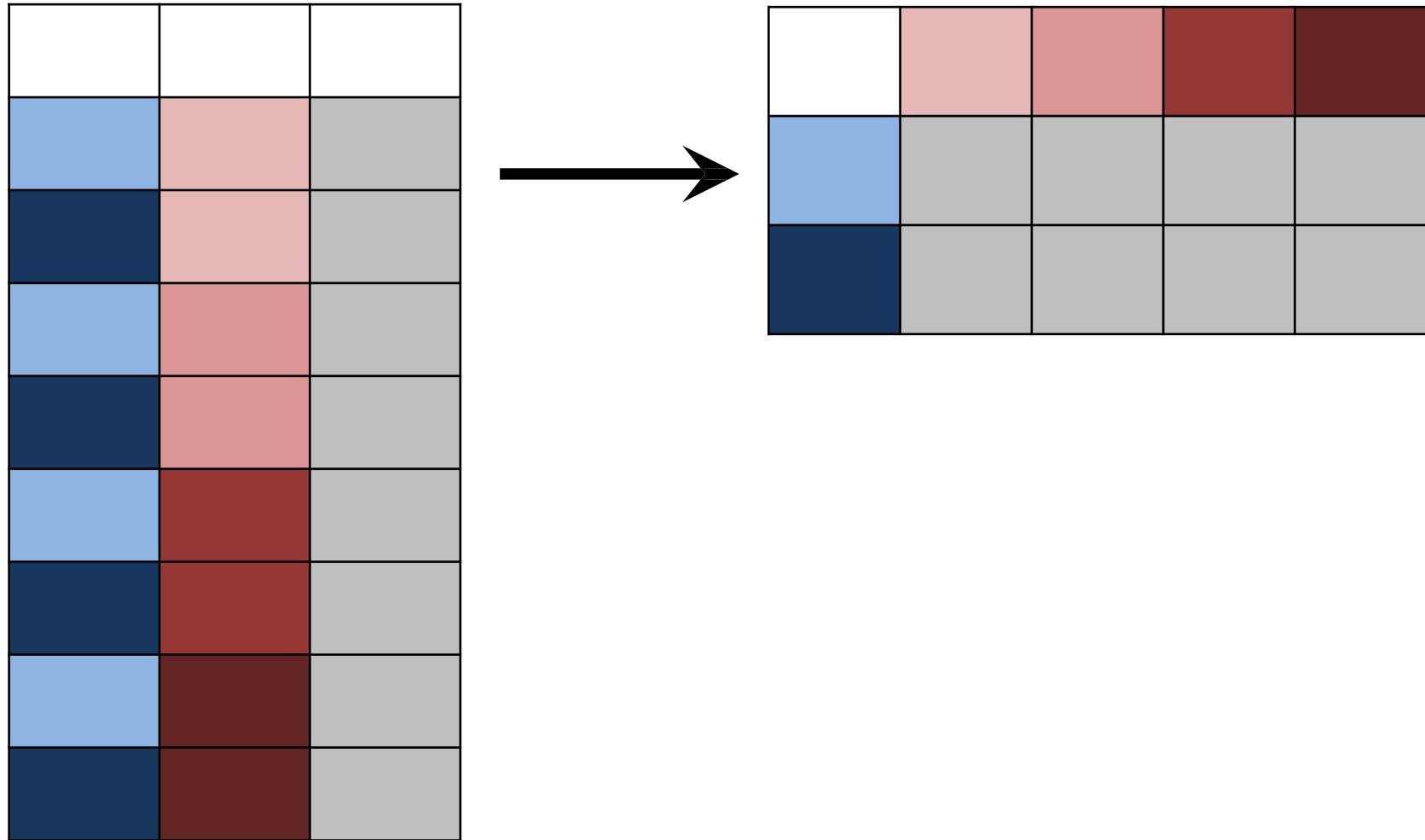
```
pivot_longer(
  t152:t258, names_to = "time", values_to = "size"
)
```

**pivot\_longer() function**

# Example: Tidying dataset using pivot\_longer()

```
> sitka_wide %>%
  pivot_longer(
    t152:t258, names_to = "time", values_to = "size"
  )
# A tibble: 395 x 4
  tree treat time   size
  <int> <fct> <chr> <dbl>
1     1 ozone t152   4.51
2     1 ozone t174   4.98
3     1 ozone t201   5.41
4     1 ozone t227   5.9 
5     1 ozone t258   6.15
6     2 ozone t152   4.24
7     2 ozone t174   4.2 
8     2 ozone t201   4.68
9     2 ozone t227   4.92
10    2 ozone t258   4.96
# ... with 385 more rows
```

# pivot\_wider()



# pivot\_wider()

|           |     |      |
|-----------|-----|------|
|           |     |      |
| Blue      | Red | Grey |
| Dark Blue | Red | Grey |
| Blue      | Red | Grey |
| Dark Blue | Red | Grey |



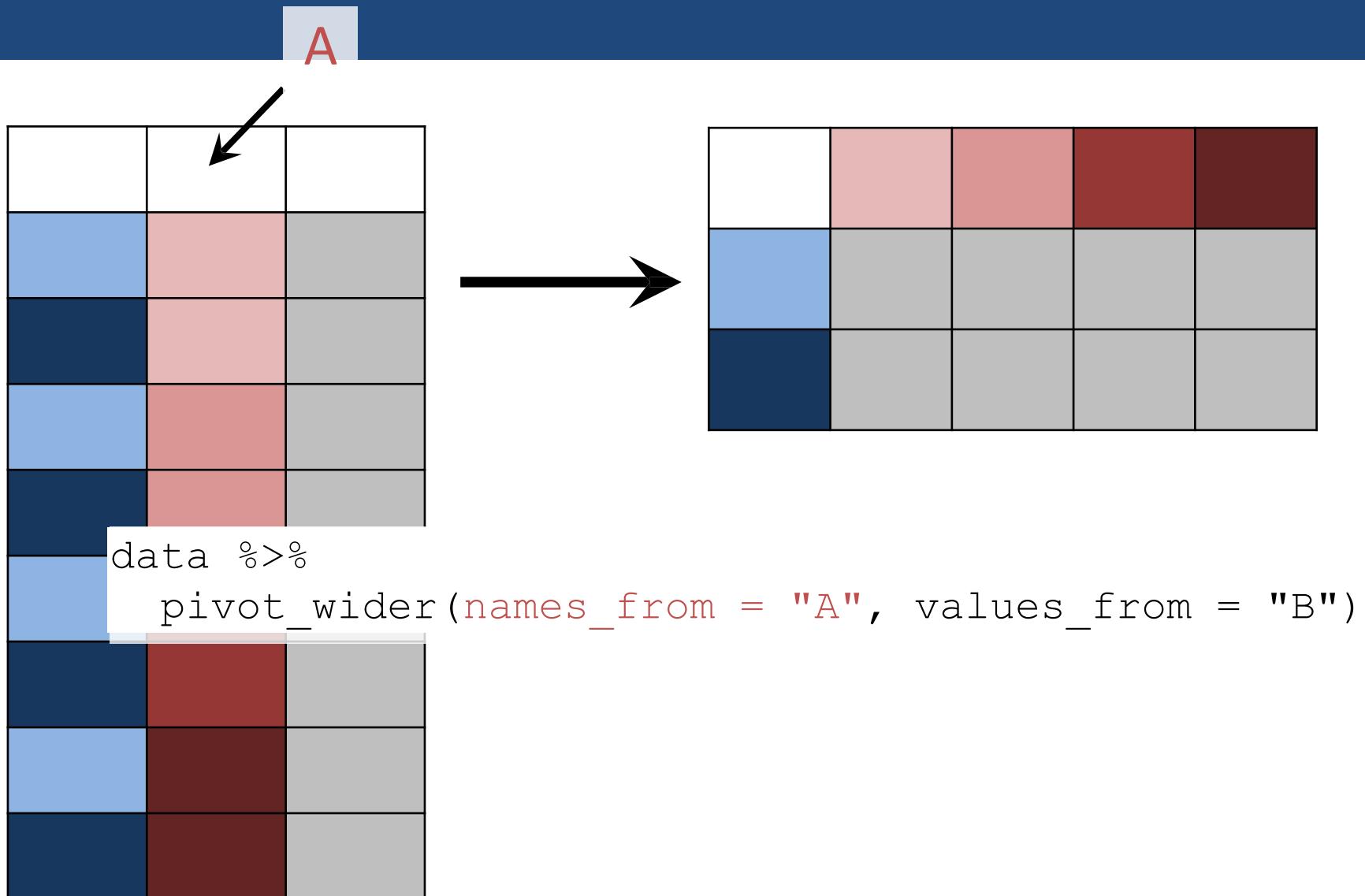
|           |     |     |          |          |
|-----------|-----|-----|----------|----------|
|           | Red | Red | Dark Red | Dark Red |
| Blue      |     |     |          |          |
| Dark Blue |     |     |          |          |
| Blue      |     |     |          |          |
| Dark Blue |     |     |          |          |

```
data %>%
```

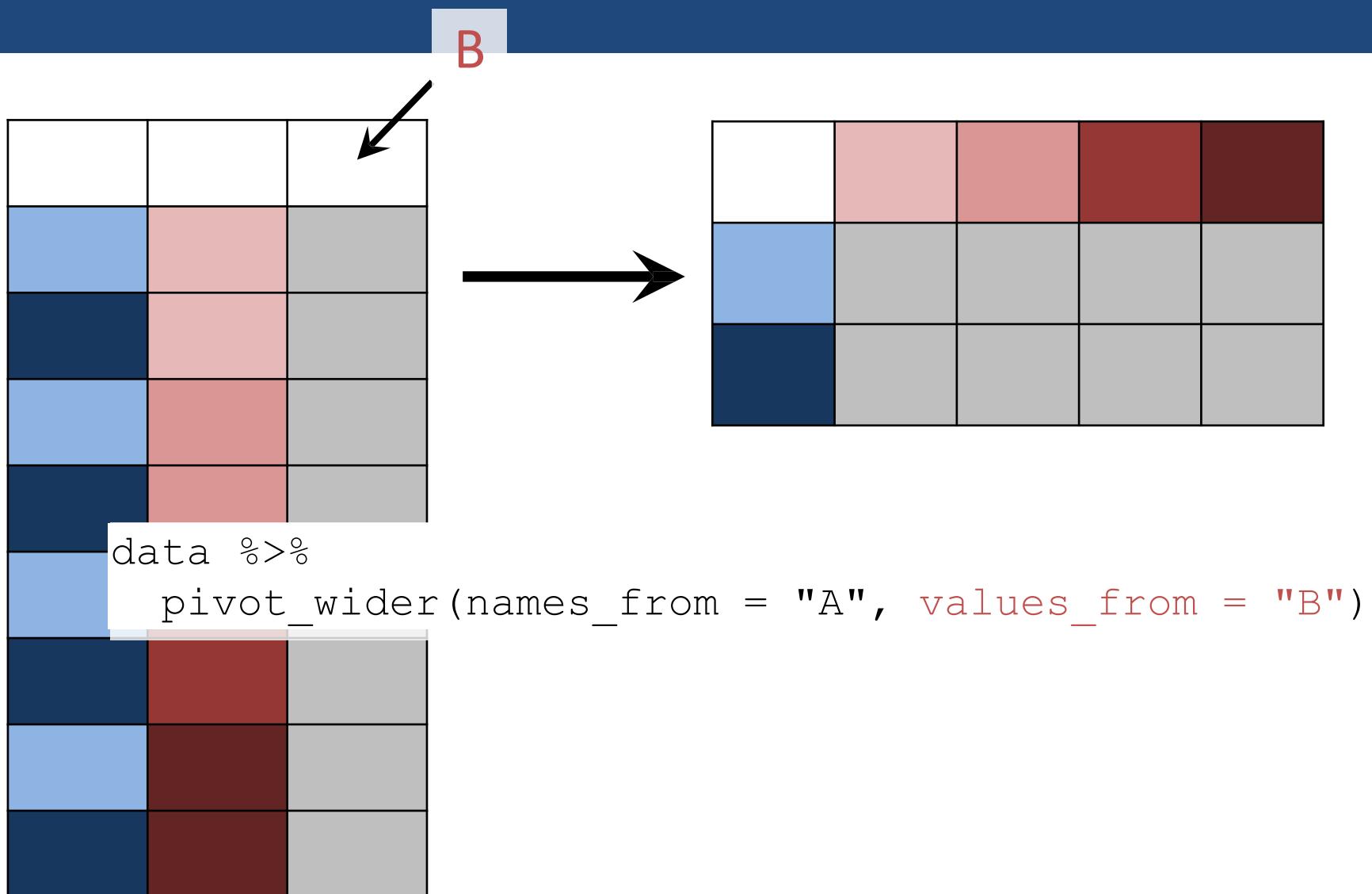
```
  pivot_wider(names_from = "A", values_from = "B")
```

|           |          |      |
|-----------|----------|------|
|           | Dark Red | Grey |
| Dark Blue |          |      |
| Blue      |          |      |
| Dark Blue |          |      |

# pivot\_wider()



# pivot\_wider()



# Example: Let's turn the sitka data into a wide table

```
> head(sitka)
  size Time tree treat
1 4.51   152     1 ozone
2 4.98   174     1 ozone
3 5.41   201     1 ozone
4 5.90   227     1 ozone
5 6.15   258     1 ozone
6 4.24   152     2 ozone

sitka %>%
  pivot_wider(names_from="Time", values_from="size")
```

# Example: Let's turn the Sitka data into a wide table

```
> sitka %>%
  pivot_wider(names_from="Time", values_from="size")

# A tibble: 79 x 7
  tree treat `152` `174` `201` `227` `258`
  <int> <fct> <dbl> <dbl> <dbl> <dbl> <dbl>
1     1 ozone   4.51   4.98   5.41   5.9    6.15
2     2 ozone   4.24   4.2    4.68   4.92   4.96
3     3 ozone   3.98   4.36   4.79   4.99   5.03
4     4 ozone   4.36   4.77   5.1    5.3    5.36
5     5 ozone   4.34   4.95   5.42   5.97   6.28
6     6 ozone   4.59   5.08   5.36   5.76   6
7     7 ozone   4.41   4.56   4.95   5.23   5.33
8     8 ozone   4.24   4.64   4.95   5.38   5.48
9     9 ozone   4.82   5.17   5.76   6.12   6.24
10    10 ozone  3.84   4.17   4.67   4.67   4.8
# ... with 69 more rows
```

# Working with tidy data in R: tidyverse

## Fundamental actions on data tables:

- make new columns — `mutate()`
- combine tables, adding columns — `left_join()`
- combine tables, adding rows — `bind_rows()`
- choose rows — `filter()`
- choose columns — `select()`
- arrange rows — `arrange()`
- calculate summary statistics — `summarize()`
- work on groups of data — `group_by()`

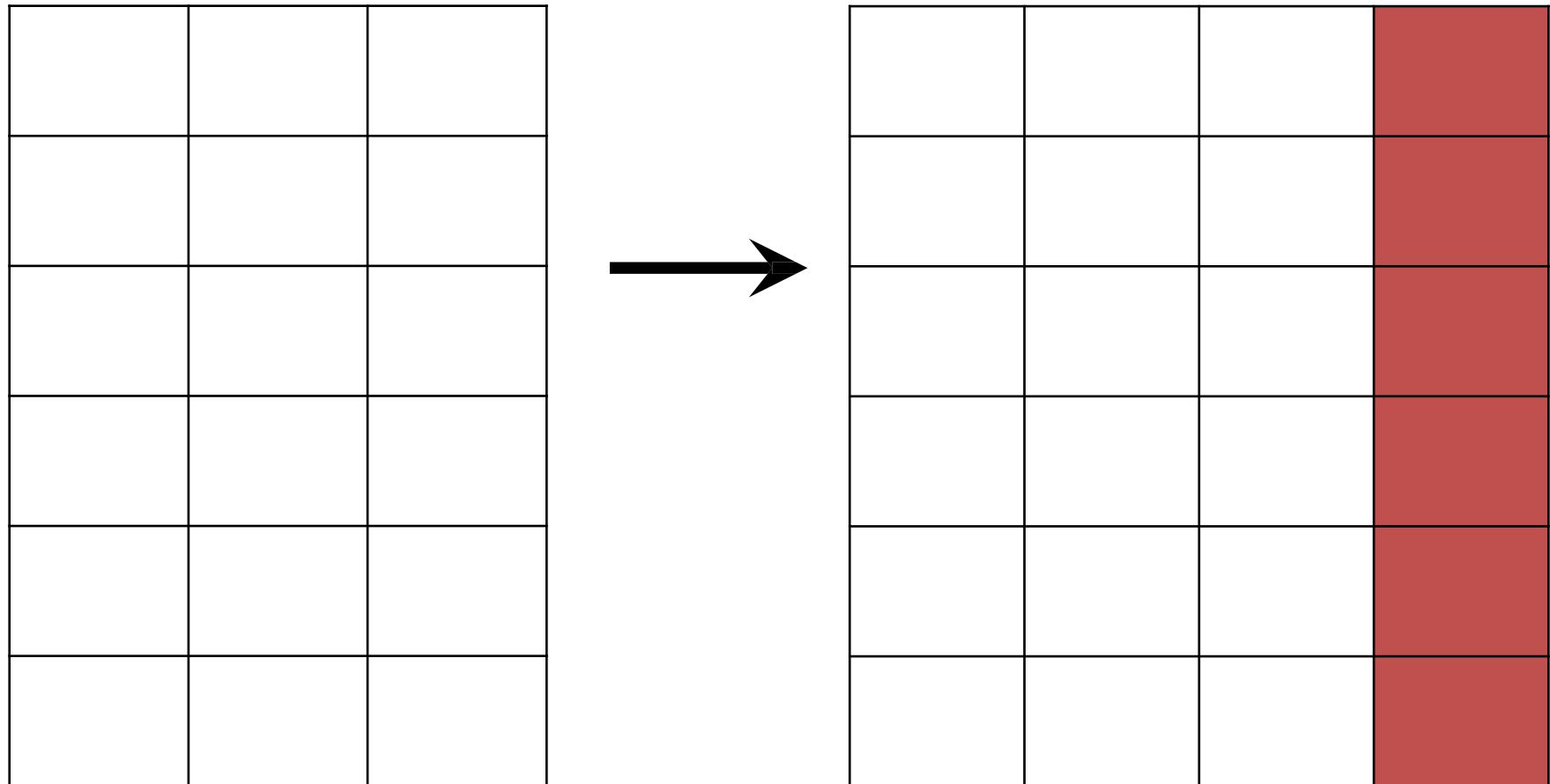
# Working with tidy data in R: tidyverse

## Fundamental actions on data tables:

- make new columns — `mutate()`
- combine tables, adding columns — `left_join()`
- combine tables, adding rows — `bind_rows()`
- choose rows — `filter()`
- choose columns — `select()`
- arrange rows — `arrange()`
- calculate summary statistics — `summarize()`
- work on groups of data — `group_by()`

# mutate () : make new columns

# mutate () : make new columns



# Make new column with ratio of Sepal.Length to Sepal.Width

```
> mutate(iris, sepal_length_to_width = Sepal.Length/Sepal.Width)
```

# Make new column with ratio of Sepal.Length to Sepal.Width

```
> mutate(iris, sepal_length_to_width = Sepal.Length/Sepal.Width)
```

|    | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species | sepal_length_to_width |
|----|--------------|-------------|--------------|-------------|---------|-----------------------|
| 1  | 5.1          | 3.5         | 1.4          | 0.2         | setosa  | 1.457143              |
| 2  | 4.9          | 3.0         | 1.4          | 0.2         | setosa  | 1.633333              |
| 3  | 4.7          | 3.2         | 1.3          | 0.2         | setosa  | 1.468750              |
| 4  | 4.6          | 3.1         | 1.5          | 0.2         | setosa  | 1.483871              |
| 5  | 5.0          | 3.6         | 1.4          | 0.2         | setosa  | 1.388889              |
| 6  | 5.4          | 3.9         | 1.7          | 0.4         | setosa  | 1.384615              |
| 7  | 4.6          | 3.4         | 1.4          | 0.3         | setosa  | 1.352941              |
| 8  | 5.0          | 3.4         | 1.5          | 0.2         | setosa  | 1.470588              |
| 9  | 4.4          | 2.9         | 1.4          | 0.2         | setosa  | 1.517241              |
| 10 | 4.9          | 3.1         | 1.5          | 0.1         | setosa  | 1.580645              |
| 11 | 5.4          | 3.7         | 1.5          | 0.2         | setosa  | 1.459459              |
| 12 | 4.8          | 3.4         | 1.6          | 0.2         | setosa  | 1.411765              |
| 13 | 4.8          | 3.0         | 1.4          | 0.1         | setosa  | 1.600000              |
| 14 | 4.3          | 3.0         | 1.1          | 0.1         | setosa  | 1.433333              |
| 15 | 5.8          | 4.0         | 1.2          | 0.2         | setosa  | 1.450000              |
| 16 | 5.7          | 4.4         | 1.5          | 0.4         | setosa  | 1.295455              |
| 17 | 5.4          | 3.9         | 1.3          | 0.4         | setosa  | 1.384615              |
| 18 | 5.1          | 3.5         | 1.4          | 0.3         | setosa  | 1.457143              |
| 19 | 5.7          | 3.8         | 1.7          | 0.3         | setosa  | 1.500000              |
| 20 | 5.1          | 3.8         | 1.5          | 0.3         | setosa  | 1.342105              |

# rbind() or bind\_rows()

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |

|      |  |  |
|------|--|--|
| ID_4 |  |  |
| ID_5 |  |  |
| ID_6 |  |  |

# rbind() or bind\_rows(): Stack tables

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |



|      |  |  |
|------|--|--|
| ID_4 |  |  |
| ID_5 |  |  |
| ID_6 |  |  |



|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |
| ID_4 |  |  |
| ID_5 |  |  |
| ID_6 |  |  |

# `left_join()`: combine two tables

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |

# left\_join () : combine two tables

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |



|      |  |  |  |  |
|------|--|--|--|--|
| ID_1 |  |  |  |  |
| ID_2 |  |  |  |  |
| ID_3 |  |  |  |  |

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |



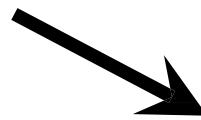
`left_join()`: missing values in 2<sup>nd</sup> table  
are set to NA

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |

|      |  |  |
|------|--|--|
| ID_2 |  |  |
|------|--|--|

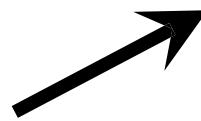
`left_join()`: missing values in 2<sup>nd</sup> table  
are set to NA

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |
| ID_3 |  |  |



|      |  |  |    |    |
|------|--|--|----|----|
| ID_1 |  |  | NA | NA |
| ID_2 |  |  |    |    |
| ID_3 |  |  | NA | NA |

|      |  |  |
|------|--|--|
| ID_2 |  |  |
|------|--|--|



`left_join()`: values from 2<sup>nd</sup> table are duplicated where necessary

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_1 |  |  |
| ID_2 |  |  |
| ID_2 |  |  |

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |

`left_join()`: values from 2<sup>nd</sup> table are duplicated where necessary

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_1 |  |  |
| ID_2 |  |  |
| ID_2 |  |  |

|      |  |  |
|------|--|--|
| ID_1 |  |  |
| ID_2 |  |  |



|      |  |  |  |  |
|------|--|--|--|--|
| ID_1 |  |  |  |  |
| ID_1 |  |  |  |  |
| ID_2 |  |  |  |  |
| ID_2 |  |  |  |  |

# Example: Joining tables

Let's extract two tables from msleep:

# Example: Joining tables

Let's extract two tables from msleep:

```
> order_table <- select(msleep, name, order)  
> order_table
```

|    |                            | name              | order        |
|----|----------------------------|-------------------|--------------|
| 1  |                            | Cheetah           | Carnivora    |
| 2  |                            | Owl monkey        | Primates     |
| 3  |                            | Mountain beaver   | Rodentia     |
| 4  | Greater short-tailed shrew |                   | Soricomorpha |
| 5  |                            | Cow               | Artiodactyla |
| 6  |                            | Three-toed sloth  | Pilosa       |
| 7  |                            | Northern fur seal | Carnivora    |
| 8  |                            | Vesper mouse      | Rodentia     |
| 9  |                            | Dog               | Carnivora    |
| 10 |                            | Roe deer          | Artiodactyla |

# Example: Joining tables

Let's extract two tables from msleep:

```
> awake_table <- select(msleep, name, awake)
> awake_table
  name   awake
1 Cheetah 11.90
2 Owl monkey 7.00
3 Mountain beaver 9.60
4 Greater short-tailed shrew 9.10
5 Cow 20.00
6 Three-toed sloth 9.60
7 Northern fur seal 15.30
8 Vesper mouse 17.00
9 Dog 13.90
10 Roe deer 21.00
```

# Example: Joining tables

And put them back together:

```
> left_join(order_table, awake_table)
```

# Example: Joining tables

And put them back together:

```
> left_join(order_table, awake_table)
```

Joining by: "name"

|    |                            | name              | order        | awake |
|----|----------------------------|-------------------|--------------|-------|
| 1  |                            | Cheetah           | Carnivora    | 11.90 |
| 2  |                            | Owl monkey        | Primates     | 7.00  |
| 3  |                            | Mountain beaver   | Rodentia     | 9.60  |
| 4  | Greater short-tailed shrew |                   | Soricomorpha | 9.10  |
| 5  |                            | Cow               | Artiodactyla | 20.00 |
| 6  |                            | Three-toed sloth  | Pilosa       | 9.60  |
| 7  |                            | Northern fur seal | Carnivora    | 15.30 |
| 8  |                            | Vesper mouse      | Rodentia     | 17.00 |
| 9  |                            | Dog               | Carnivora    | 13.90 |
| 10 |                            | Roe deer          | Artiodactyla | 21.00 |

# Several different join functions are available

- `left_join()`
- `right_join()`
- `inner_join()`
- `semi_join()`
- `full_join()`
- `anti_join()`

# Demonstration Time!

Refer to workbook Section 5

# Working with tidy data in R: tidyverse

Fundamental actions on data tables:

- make new columns — `mutate()`
- combine tables, adding columns — `left_join()`
- combine tables, adding rows — `bind_rows()`
- choose rows — `filter()`
- choose columns — `select()`
- arrange rows — `arrange()`
- calculate summary statistics — `summarize()`
- work on groups of data — `group_by()`

# filter () : pick rows

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# filter(): pick rows

|       |       |       |
|-------|-------|-------|
| Red   | Red   | Red   |
| White | White | White |
| Red   | Red   | Red   |
| Red   | Red   | Red   |
| White | White | White |
| Red   | Red   | Red   |



|     |     |     |
|-----|-----|-----|
| Red | Red | Red |

# Choose rows with Sepal.Width > 4

```
> filter(iris, Sepal.Width > 4)
```

# Choose rows with Sepal.Width > 4

```
> filter(iris, Sepal.Width > 4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.7       4.4         1.5       0.4   setosa
2          5.2       4.1         1.5       0.1   setosa
3          5.5       4.2         1.4       0.2   setosa
```

# select () : pick columns

# select () : pick columns

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# select () : pick columns



|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Choose the two columns Species and Sepal.Width

```
> select(iris, Species, Sepal.Width)
```

# Choose the two columns Species and Sepal.Width

```
> select(iris, Species, Sepal.Width)
```

|    | Species | Sepal.Width |
|----|---------|-------------|
| 1  | setosa  | 3.5         |
| 2  | setosa  | 3.0         |
| 3  | setosa  | 3.2         |
| 4  | setosa  | 3.1         |
| 5  | setosa  | 3.6         |
| 6  | setosa  | 3.9         |
| 7  | setosa  | 3.4         |
| 8  | setosa  | 3.4         |
| 9  | setosa  | 2.9         |
| 10 | setosa  | 3.1         |
| 11 | setosa  | 3.7         |
| 12 | setosa  | 3.4         |
| 13 | setosa  | 3.0         |
| 14 | setosa  | 3.0         |

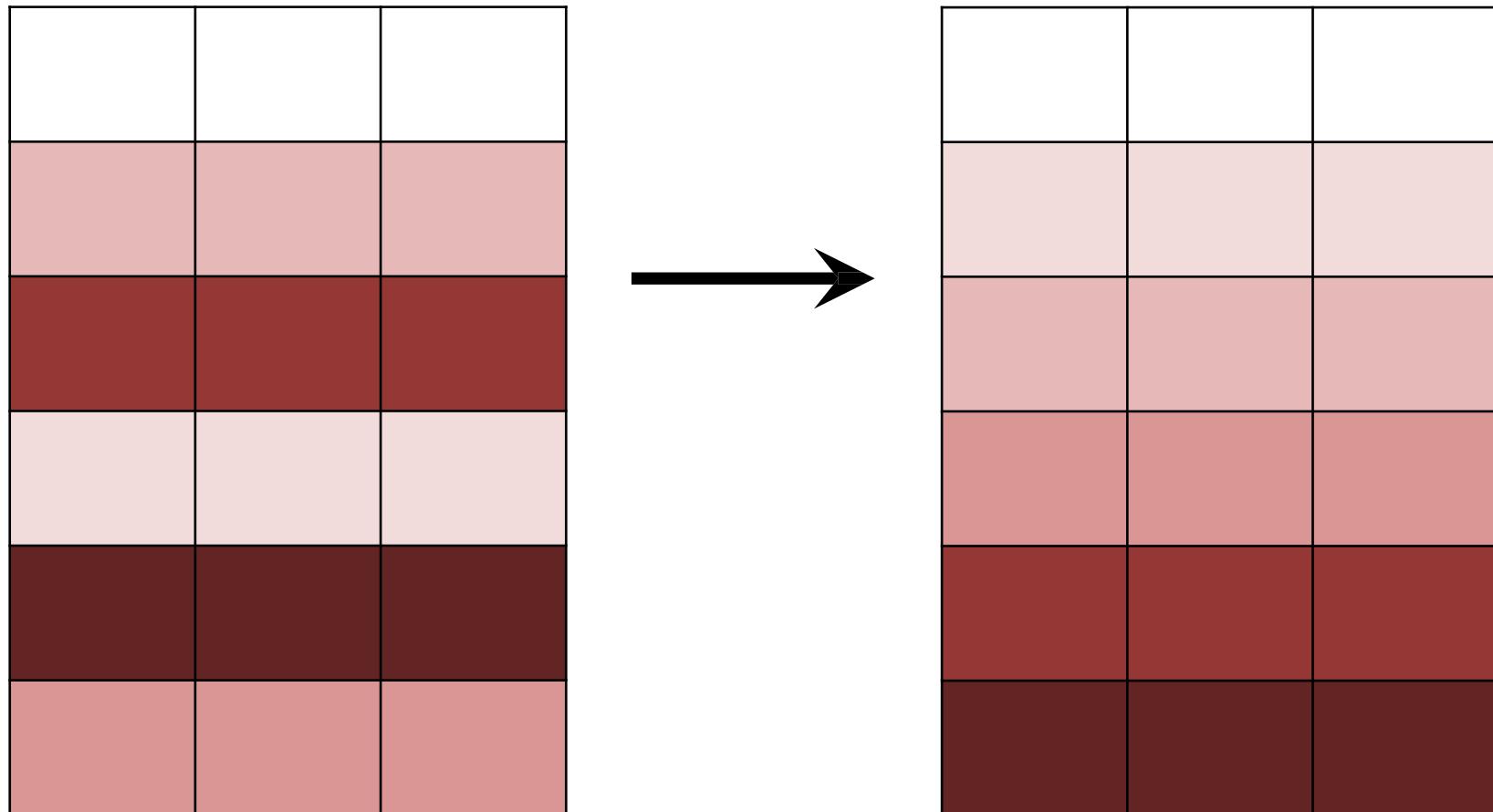
# Demonstration Time!

Refer to Section #6

# arrange ( ) : change row order

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# arrange ( ) : change row order



# Sort by increasing order of Sepal.Width

```
> arrange(iris, Sepal.Width)
```

# Sort by increasing order of Sepal.Width

```
> arrange(iris, Sepal.Width)
```

|    | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species    |
|----|--------------|-------------|--------------|-------------|------------|
| 1  | 5.           | 2.0         | 3.5          | 1.0         | versicolor |
| 2  | 6.           | 2.2         | 4.0          | 1.0         | versicolor |
| 3  | 6.2          | 2.2         | 4.5          | 1.5         | versicolor |
| 4  | 6.0          | 2.2         | 5.0          | 1.5         | virginica  |
| 5  | 4.5          | 2.3         | 1.3          | 0.3         | setosa     |
| 6  | 5.5          | 2.3         | 4.0          | 1.3         | versicolor |
| 7  | 6.3          | 2.3         | 4.4          | 1.3         | versicolor |
| 8  | 5.0          | 2.3         | 3.3          | 1.0         | versicolor |
| 9  | 4.9          | 2.4         | 3.3          | 1.0         | versicolor |
| 10 | 5.5          | 2.4         | 3.8          | 1.1         | versicolor |
| 11 | 5.           | 2.4         | 3.7          | 1.0         | versicolor |
| 12 | 6.           | 2.5         | 3.9          | 1.1         | versicolor |

# Sort by decreasing order of Sepal.Length

```
> arrange(iris, desc(Sepal.Length))
```

# Sort by decreasing order of Sepal.Length

```
> arrange(iris, desc(Sepal.Length))
```

|    | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species   |
|----|--------------|-------------|--------------|-------------|-----------|
| 1  | 7.9          | 3.8         | 6.4          | 2.0         | virginica |
| 2  | 7.7          | 3.8         | 6.7          | 2.2         | virginica |
| 3  | 7.7          | 2.6         | 6.9          | 2.3         | virginica |
| 4  | 7.7          | 2.8         | 6.7          | 2.0         | virginica |
| 5  | 7.7          | 3.0         | 6.1          | 2.3         | virginica |
| 6  | 7.6          | 3.0         | 6.6          | 2.1         | virginica |
| 7  | 7.4          | 2.8         | 6.1          | 1.9         | virginica |
| 8  | 7.3          | 2.9         | 6.3          | 1.8         | virginica |
| 9  | 7.2          | 3.6         | 6.1          | 2.5         | virginica |
| 10 | 7.2          | 3.2         | 6.0          | 1.8         | virginica |
| 11 | 7.2          | 3.0         | 5.8          | 1.6         | virginica |
| 12 | 7.1          | 3.0         | 5.9          | 2.1         | virginica |

# Working with tidy data in R: tidyverse

Fundamental actions on data tables:

- make new columns — `mutate()`
- combine tables, adding columns — `left_join()`
- combine tables, adding rows — `bind_rows()`
- choose rows — `filter()`
- choose columns — `select()`
- arrange rows — `arrange()`
- calculate summary statistics — `summarize()`
- work on groups of data — `group_by()`

# summarize () : collapse multiple rows

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# summarize () : collapse multiple rows

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |

# Calculate mean and standard deviation of Sepal.Length

```
> summarize(iris, mean_sepal_length = mean(Sepal.Length),  
           sd_sepal_length     = sd(Sepal.Length))
```

# Calculate mean and standard deviation of Sepal.Length

```
> summarize(iris, mean_sepal_length = mean(Sepal.Length),  
           sd_sepal_length     = sd(Sepal.Length))  
mean_sepal_length sd_sepal_length  
1                 5.843333          0.8280661
```

# group\_by(): set up groupings

|   |  |  |
|---|--|--|
| A |  |  |
| B |  |  |
| A |  |  |
| A |  |  |
| B |  |  |
| B |  |  |

# group\_by(): set up groupings

|   |  |  |
|---|--|--|
| A |  |  |
| B |  |  |
| A |  |  |
| A |  |  |
| B |  |  |
| B |  |  |



|   |  |  |
|---|--|--|
| A |  |  |
| A |  |  |
| A |  |  |
| B |  |  |
| B |  |  |
| B |  |  |

# Calculate mean and standard deviation of Sepal.Length, grouped by Species

```
> summarize(group_by(iris, Species),  
           mean_sepal_length = mean(Sepal.Length),  
           sd_sepal_length   = sd(Sepal.Length))
```

# Calculate mean and standard deviation of Sepal.Length, grouped by Species

```
> summarize(group_by(iris, Species),  
           mean_sepal_length = mean(Sepal.Length),  
           sd_sepal_length   = sd(Sepal.Length))
```

Source: local data frame [3 x 3]

|   | Species    | mean_sepal_length | sd_sepal_length |
|---|------------|-------------------|-----------------|
| 1 | setosa     | 5.006             | 0.3524897       |
| 2 | versicolor | 5.936             | 0.5161711       |
| 3 | virginica  | 6.588             | 0.6358796       |

# Pipe example 1: count how many herbivores of different orders there are in msleep

|   | name                       | genus      | vore  | order     | conse... <sup>1</sup> | sleep... <sup>2</sup> | sleep... <sup>3</sup> | sleep... <sup>4</sup> | awake | brainwt | bodywt |
|---|----------------------------|------------|-------|-----------|-----------------------|-----------------------|-----------------------|-----------------------|-------|---------|--------|
|   | <chr>                      | <chr>      | <chr> | <chr>     | <dbl>                 | <dbl>                 | <dbl>                 | <dbl>                 | <dbl> | <dbl>   | <dbl>  |
| 1 | Cheetah                    | Acinonyx   | carni | Carniv... | lc                    | 12.1                  | NA                    | NA                    | 11.9  | NA      | 50     |
| 2 | Owl monkey                 | Aotus      | omni  | Primat... | NA                    | 17                    | 1.8                   | NA                    | 7     | 0.0155  | 0.48   |
| 3 | Mountain beaver            | Aplodontia | herbi | Rodent... | nt                    | 14.4                  | 2.4                   | NA                    | 9.6   | NA      | 1.35   |
| 4 | Greater short-tailed shrew | Blarina    | omni  | Sorico... | lc                    | 14.9                  | 2.3                   | 0.133                 | 9.1   | 0.00029 | 0.019  |
| 5 | Cow                        | Bos        | herbi | Artiod... | domest...             | 4                     | 0.7                   | 0.667                 | 20    | 0.423   | 600    |
| 6 | Three-toed sloth           | Bradypus   | herbi | Pilosa    | NA                    | 14.4                  | 2.2                   | 0.767                 | 9.6   | NA      | 3.85   |

# with abbreviated variable names: <sup>1</sup>conservation, <sup>2</sup>sleep total, <sup>3</sup>sleep rem, <sup>4</sup>sleep cycle

# Pipe example 1: count how many herbivores of different orders there are in msleep

```
msleep %>%  
  filter(vore == "herbi")
```

# Pipe example 1: count how many herbivores of different orders there are in msleep

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order)
```

# Pipe example 1: count how many herbivores of different orders there are in msleep

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order) %>%
  summarize(count = n())
```

# Pipe example 1: count how many herbivores of different orders there are in msleep

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

# Pipe example 1: count how many herbivores of different orders there are in msleep

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

|   | order          | count |
|---|----------------|-------|
| 1 | Rodentia       | 16    |
| 2 | Artiodactyla   | 5     |
| 3 | Perissodactyla | 3     |
| 4 | Hyracoidea     | 2     |
| 5 | Proboscidea    | 2     |
| 6 | Diprotodontia  | 1     |
| 7 | Lagomorpha     | 1     |
| 8 | Pilosa         | 1     |
| 9 | Primates       | 1     |

Pipe example 2: What is the median awake time of different orders in msleep?

# Pipe example 2: What is the median awake time of different orders in msleep?

```
msleep %>%  
  group_by(order)
```

# Pipe example 2: What is the median awake time of different orders in msleep?

```
msleep %>%  
  group_by(order) %>%  
  summarize(med_awake = median(awake))
```

# Pipe example 2: What is the median awake time of different orders in msleep?

```
msleep %>%
  group_by(order) %>%
  summarize(med_awake = median(awake)) %>%
  arrange(med_awake)
```

# Pipe example 2: What is the median awake time of different orders in msleep?

```
msleep %>%
  group_by(order) %>%
  summarize(med_awake = median(awake)) %>%
  arrange(med_awake)
```

|    | order           | med_awake |
|----|-----------------|-----------|
| 1  | Chiroptera      | 4.20      |
| 2  | Didelphimorphia | 5.30      |
| 3  | Cingulata       | 6.25      |
| 4  | Afrosoricida    | 8.40      |
| 5  | Pilosa          | 9.60      |
| 6  | Rodentia        | 11.10     |
| 7  | Diprotodontia   | 11.60     |
| 8  | Soricomorpha    | 13.70     |
| 9  | Carnivora       | 13.75     |
| 10 | Erinaceomorpha  | 13.80     |

# Demonstration Time!

Refer to Section #7