# Intro to Tidyverse!


Janani Ravi

jananiravi.github.io | @janani137 | janani@msu.edu

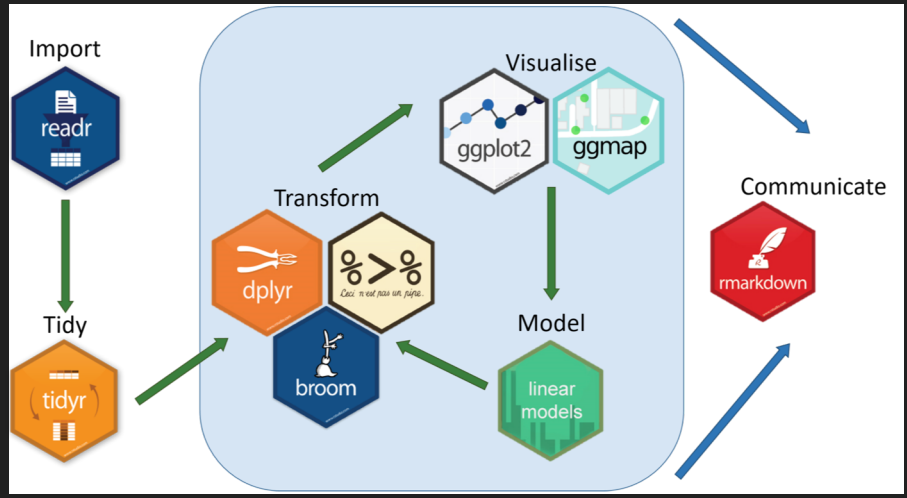
# Intro

- **Familiar w/ R and ggplot? → Shuffle**
- **Introduce yourself to your neighbor**
  - Who you are | Name, affiliation
  - Do you have the same version of
    - R (3.1+), RStudio & Tidyverse?
    - NO? Installation time!

**Need help? Red sticky! All set? Green!**

# Today!

- **Workshop**
  - Intro to Tidyverse



# Welcome to *tidyverse*

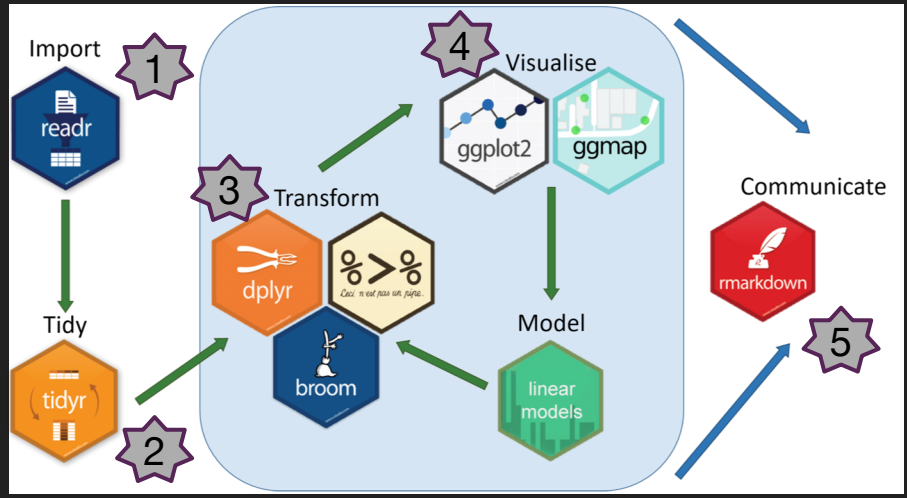
**P1:** Getting Started w/ readr

**P2:** Reshaping data w/ tidyr

**P3:** Data wrangling w/ dplyr

**P4:** DataViz w/ ggplot

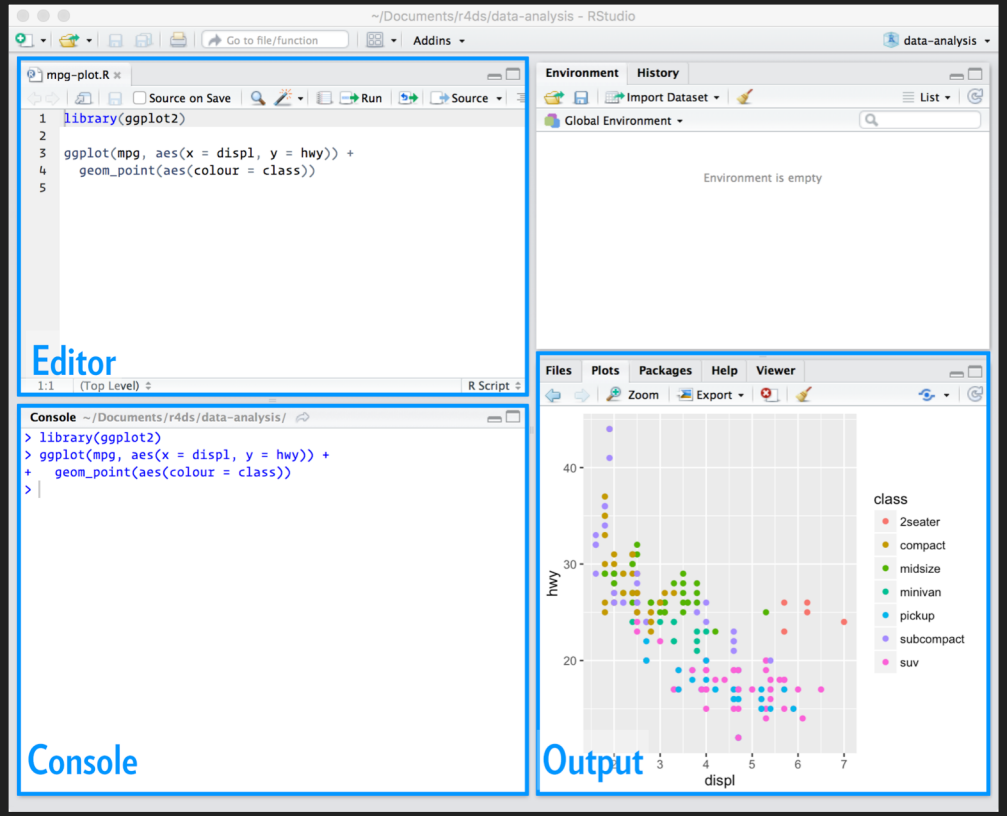
**P5:** Wrap-up w/ RMarkdown



# Part 1: Getting Started: Environment

- ✓ Installing RStudio, R
- ✓ Installing tidyverse

```
> library(tidyverse)
── Attaching packages ─────────────────────── tidyverse 1.2.1 ──
✔ ggplot2 3.0.0     ✔ purrr   0.2.5
✔ tibble  1.4.2     ✔ dplyr   0.7.6
✔ tidyr   0.8.1     ✔ stringr 1.3.1
✔ readr   1.1.1     ✔ forcats 0.3.0
── Conflicts ────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```



# Getting Started: Data, your data

1. Import your data

```
library(tidyverse)
read_csv(file="my_data.csv",
                  col_names=T)       # comma-separated values
read_delim(file="my_data.txt", col_names=T,
                     delim="//")     # any delimitter
# Other useful packages
# readxl by Jenny Bryan
read_excel(path="path/to/excel.xls",
                    sheet=1,
                    range="A1:D50",
                    col_names=T)
```

# Getting Started: Today's Dataset

## A resource of ribosomal RNA-depleted RNA-Seq data from different normal adult and fetal human tissues

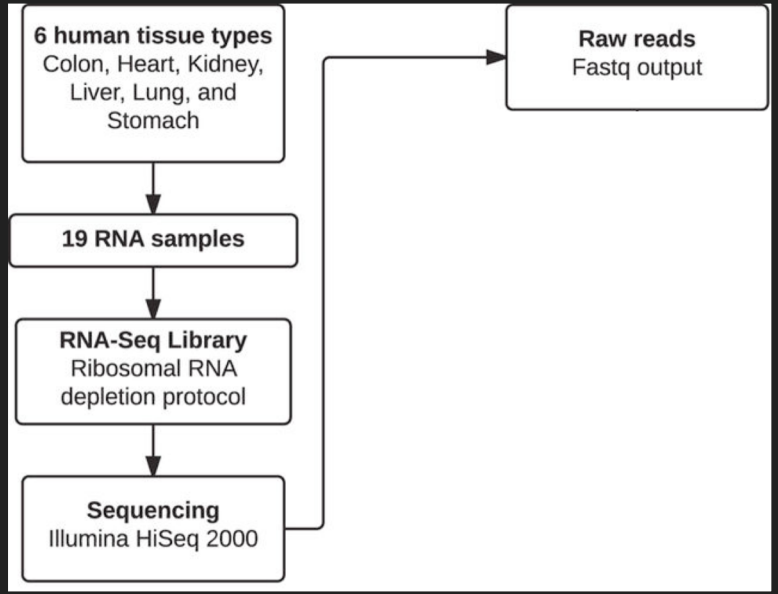
Jocelyn Y.H. Choy, Priscilla L.S. Boon, Nicolas Bertin & Melissa J. Fullwood

*Scientific Data* **2**, Article number: 150063 (2015)
doi:10.1038/sdata.2015.63
Download Citation

Development | RNA sequencing | Transcriptomics

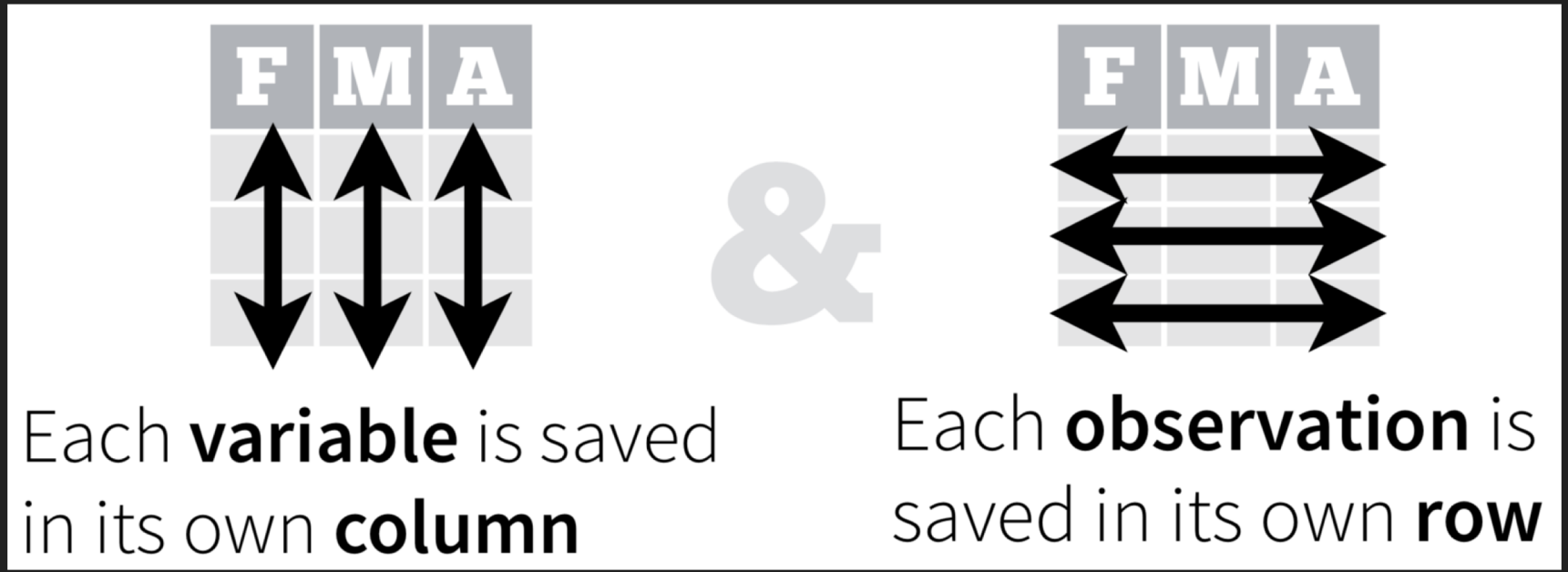
Received: 10 June 2015
Accepted: 07 October 2015
Published online: 10 November 2015

| | |
|---|---|
| **Design Type(s)** | parallel group design • replicate design • organism development design |
| **Measurement Type(s)** | transcription profiling assay |
| **Technology Type(s)** | next generation sequencing |
| **Factor Type(s)** | tissue specimen • life cycle stage |
| **Sample Characteristic(s)** | Homo sapiens • colon • stomach • heart • kidney • liver • lung |



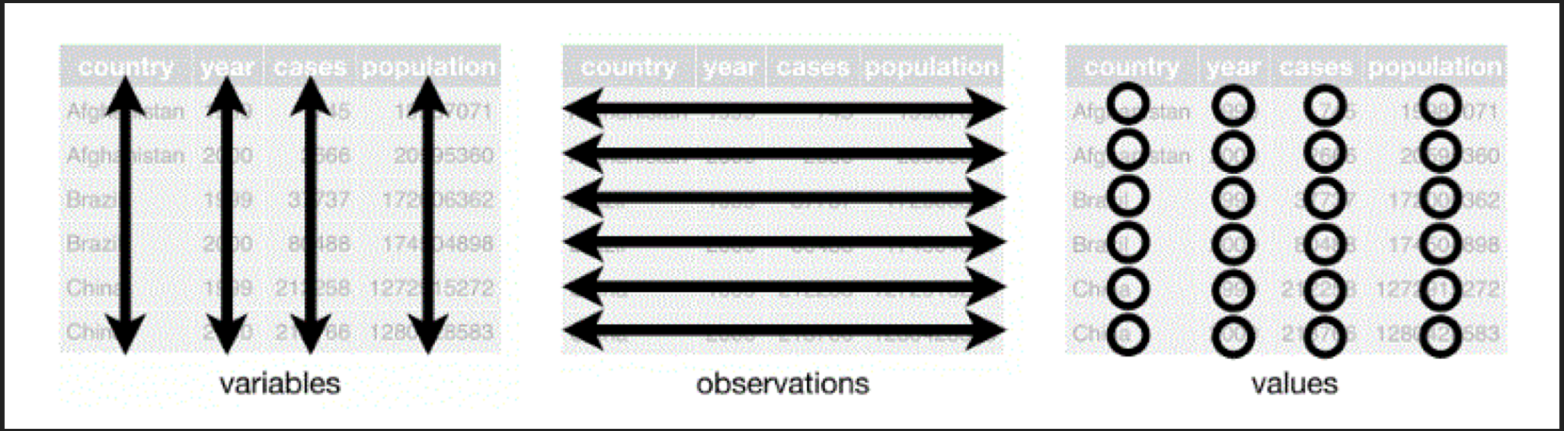

https://www.nature.com/articles/sdata201563

# Part 2: Reshaping data w/ *tidyr*



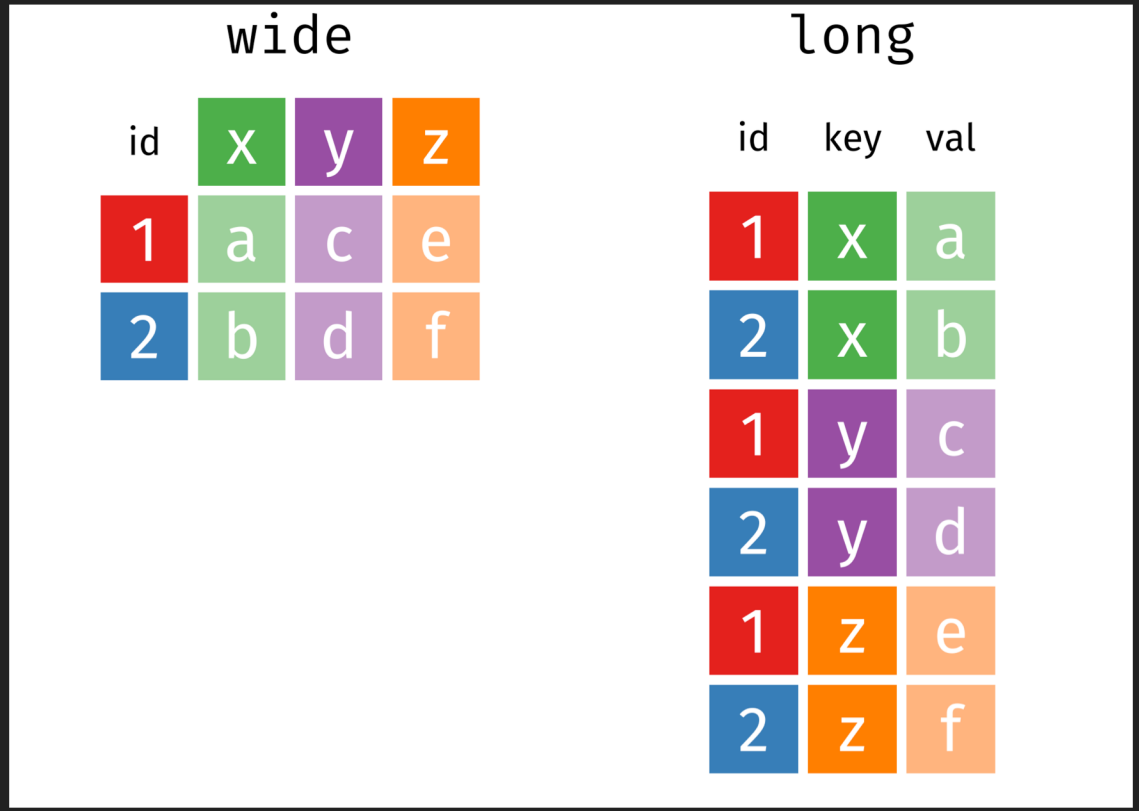

Tidy Data … In a ‘Tidy’ dataset, …

# Part 2: What is tidy data?




- Each variable in the data set is placed in its own column.
- Each observation is placed in its own row.
- Each value is placed in its own cell.

# Part 2: In a ‘Tidy’ dataset, …



# Part 2: Untidy data

| Name | MaleSex | FemaleSex | BlueEyes | BlackEyes | BrownEyes |
|---|---|---|---|---|---|
| John | 1 | 0 | 1 | 0 | 0 |
| Jane | 0 | 1 | 0 | 1 | 0 |
| Sally | 0 | 1 | 0 | 0 | 1 |

# Part 2: Tidy data

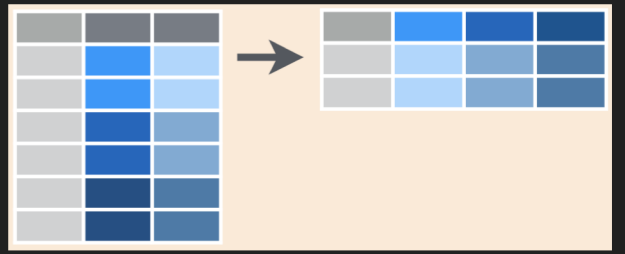
| Name | Sex | Eye color |
| --- | --- | --- |
| John | M | Blue |
| Jane | F | Black |
| Sally | F | Brown |

# Part 2: Reshaping data w/ *tidyr*

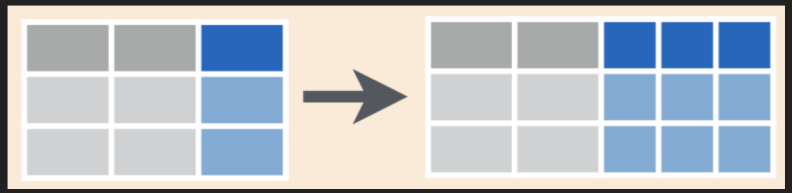
```
gather()     # Gather COLUMNS -> ROWS
spread()     # Spread ROWS -> COLUMNS
separate()   # Separate 1 COLUMN -> many COLUMNS
unite()      # Unite several COLUMNS -> 1 COLUMN
```

# Part 2: Reshaping data w/ *tidyr*

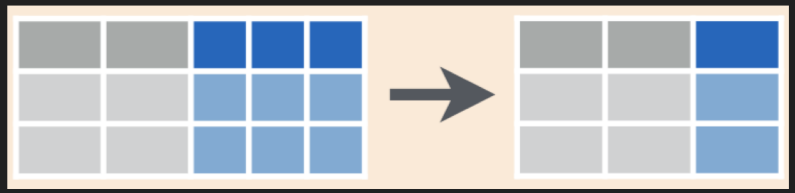
gather



spread



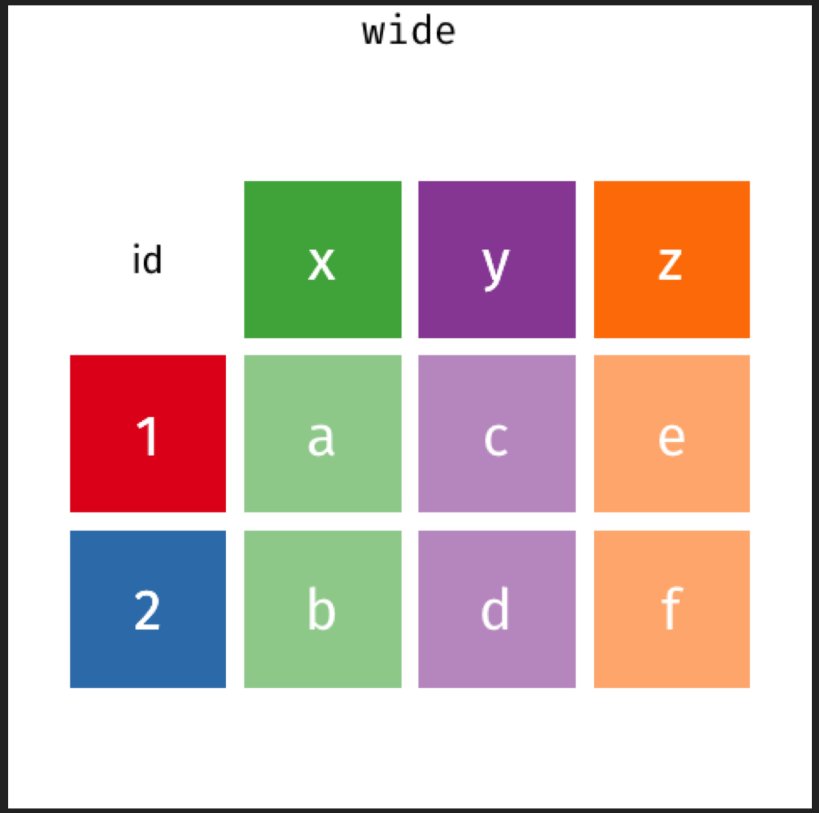
separate



unite

# Part 2: Reshaping data w/ *tidyr*

## Tidy data

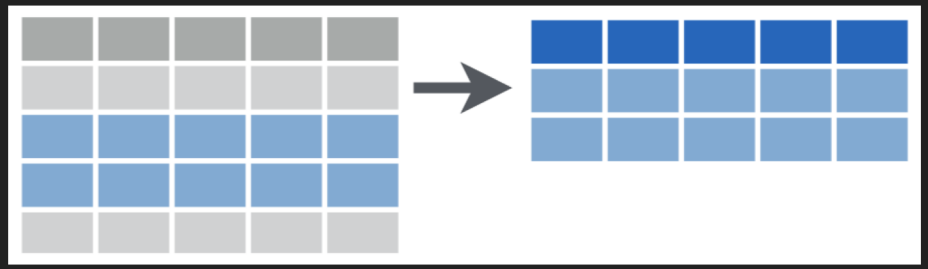



gadenbuie/tidyexplain

# Part 3: Wrangling data w/ *dplyr*

```
filter()    # PICK observations by their values | ROWS
select()    # PICK variables by their names | COLUMNS
mutate()    # CREATE new variables w/ functions of existing variables | COLUMNS
transmute() # COMPUTE 1 or more COLUMNS but drop original columns
arrange()   # REORDER the ROWS
summarize() # COLLAPSE many values to a single SUMMARY
group_by()  # GROUP data into rows with the same value of variable (COLUMN)
```

# Part 3: Wrangling data w/ *dplyr*



filter

select

# Part 3: Wrangling data w/ *dplyr*

summarise

group_by

# Part 3: Wrangling data w/ *dplyr*

## Mutating joins

a

| x1 | x2 |
|---|---|
| A | 1 |
| B | 2 |
| C | 3 |

+

b

| x1 | x3 |
|---|---|
| A | T |
| B | F |
| D | T |

=

left_join()

| x1 | x2 | x3 |
|---|---|---|
| A | 1 | T |
| B | 2 | F |
| C | 3 | NA |

right_join()

| x1 | x3 | x2 |
|---|---|---|
| A | T | 1 |
| B | F | 2 |
| D | T | NA |

inner_join()

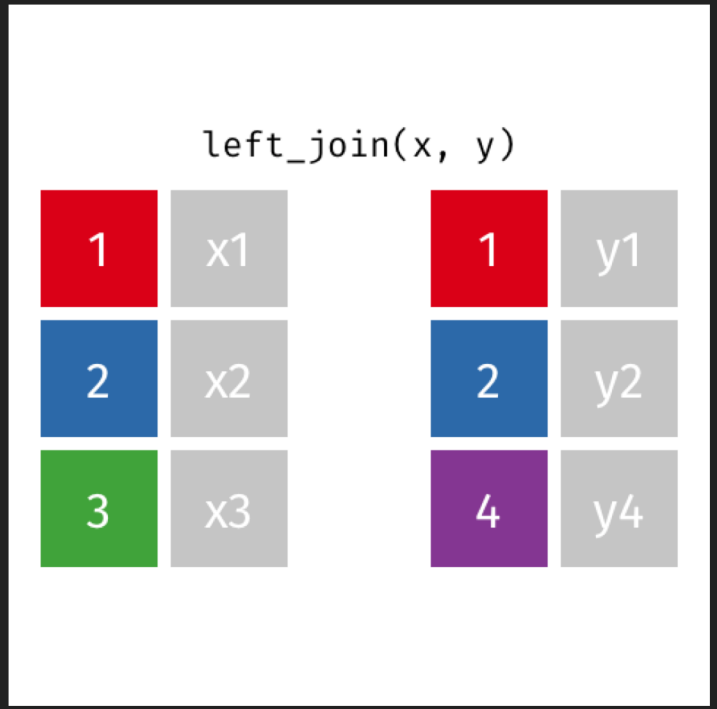
| x1 | x2 | x3 |
|---|---|---|
| A | 1 | T |
| B | 2 | F |

outer_join()

| x1 | x2 | x3 |
|---|---|---|
| A | 1 | T |
| B | 2 | F |
| C | 3 | NA |
| D | NA | T |

# Part 3: Wrangling data w/ *dplyr*

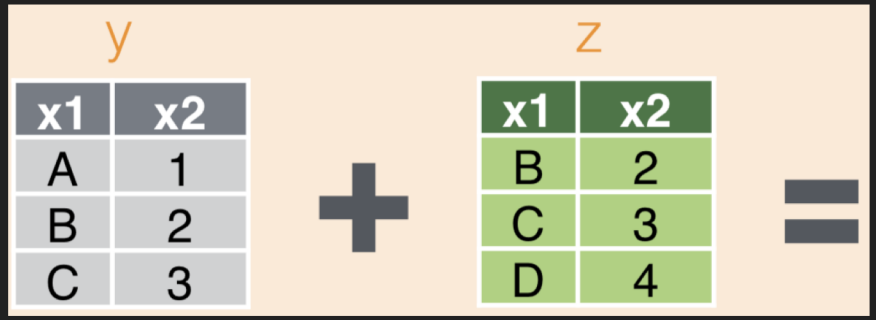
## Mutating joins




gadenbuie/tidyexplain

# Part 3: Wrangling data w/ *dplyr*

## Set operations




intersect()

| x1 | x2 |
|---|---|
| B | 2 |
| C | 3 |

union()

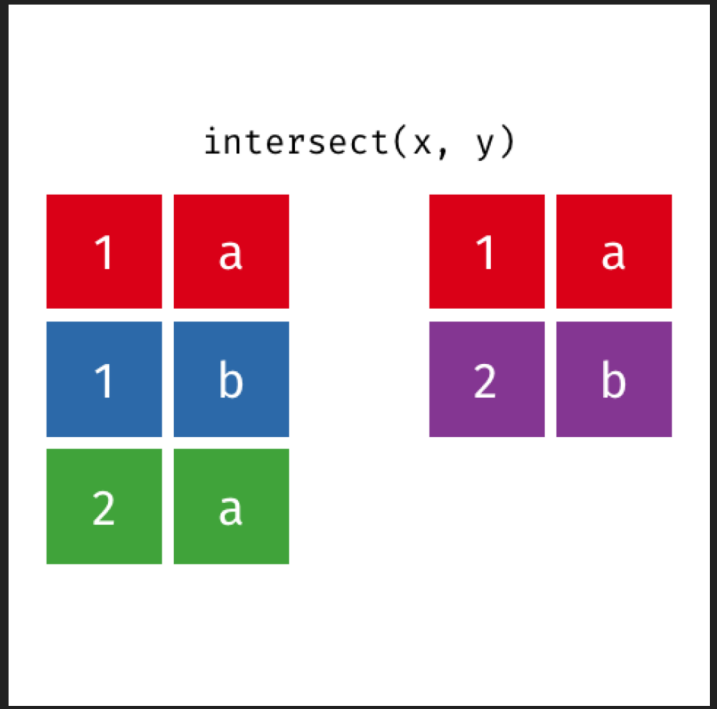
| x1 | x2 |
|---|---|
| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |

setdiff()

| x1 | x2 |
|---|---|
| A | 1 |
| D | 4 |

# Part 3: Wrangling data w/ *dplyr*

## Set operations




gadenbuie/tidyexplain

# Part 3: Wrangling data w/ *dplyr*

## Binding

y

| x1 | x2 |
|---|---|
| A | 1 |
| B | 2 |
| C | 3 |

+

z

| x1 | x2 |
|---|---|
| B | 2 |
| C | 3 |
| D | 4 |

=

bind_rows()

| x1 | x2 |
|---|---|
| A | 1 |
| B | 2 |
| C | 3 |
| B | 2 |
| C | 3 |
| D | 4 |

bind_cols()

| x1 | x2 | x1 | x2 |
|---|---|---|---|
| A | 1 | B | 2 |
| B | 2 | C | 3 |
| C | 3 | D | 4 |

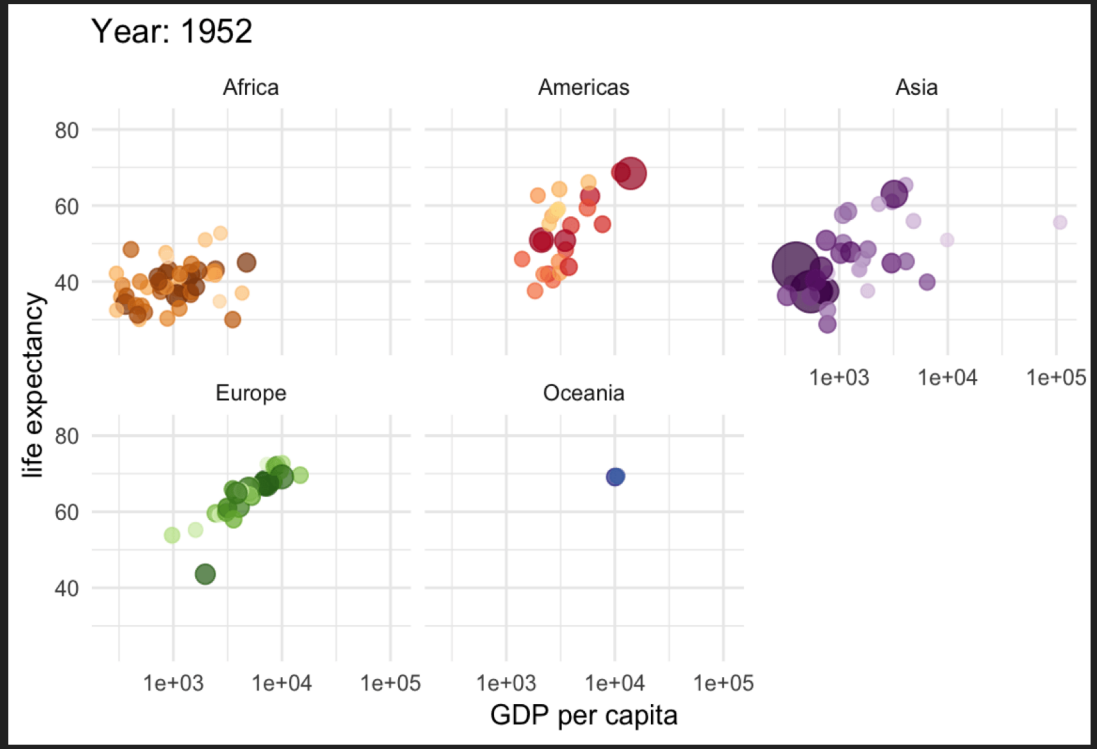
# Part 4: Visualizing tidy data w/ ggplot2

## Geometry of graphics

- **data**: Must be a data frame
- **aes**thetics: How your data are represented visually
  - x, y, color, size, shape, etc.
- **geom**etry: Geometries of plotted objects
  - points, lines, boxplot, polygons, etc.
- and *other customizations*

# Part 4: Visualizing tidy data w/ ggplot2

## gganimate: aha!

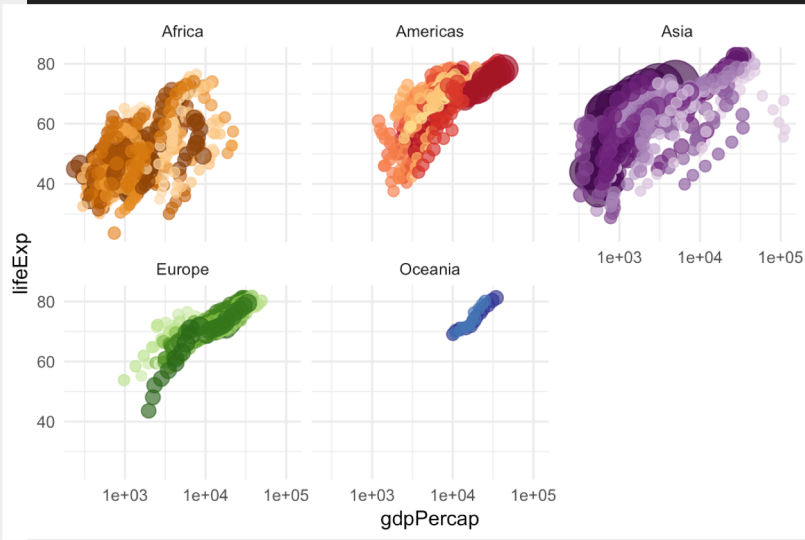



thomasp85/
gganimate

# Part 4: Visualizing tidy data w/ ggplot2

## gapminder: static plot

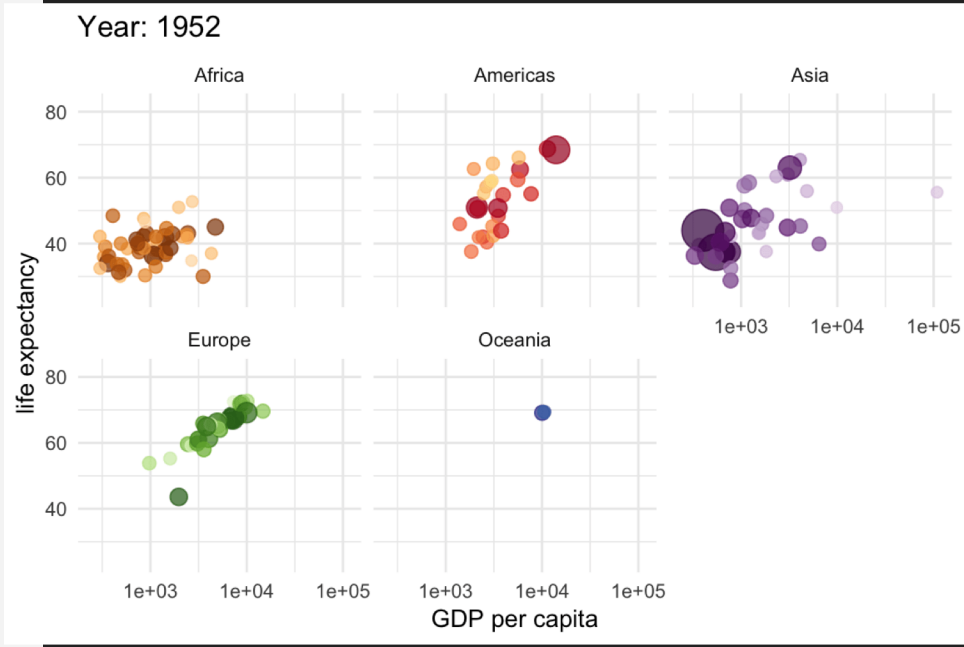
```
library(gapminder)

ggplot(gapminder,
       aes(gdpPercap, lifeExp,

           size=pop,colour=country)) +
  geom_point(alpha = 0.7,
                          show.legend = FALSE)
 +
  scale_colour_manual(values=country_colors) +
  scale_size(range=c(2, 12)) +
  scale_x_log10() +
  facet_wrap(~continent) +
  theme_minimal()
```



# Part 4: Visualizing tidy data w/ ggplot2

## gapminder: dynamic plot

```
ggplot(gapminder,
       aes(gdpPercap, lifeExp,
           size=pop, colour=country)) +
  … … … +
  theme_minimal() +
  # Here comes the gganimate part!
  labs(title = 'Year: {frame_time}',
       x = 'GDP per capita',
       y = 'life expectancy') +
  transition_time(year) +
  ease_aes('linear')
```



# Part 5: Export & Wrap-up

ggsave — Save your plots

write_delim — Save your data

# Tidyverse Recap

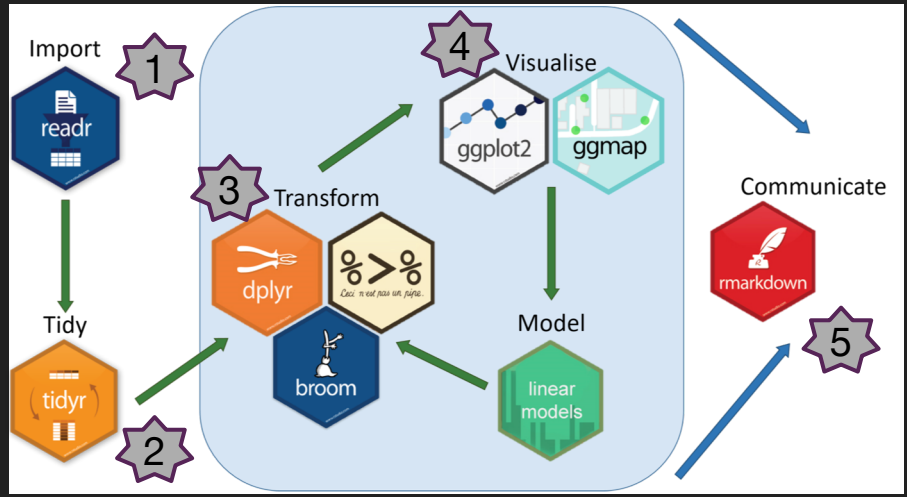
**P1:** Getting Started w/ readr ✓

**P2:** Reshaping data w/ tidyr ✓

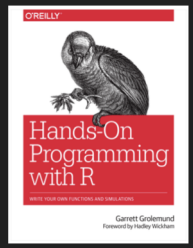
**P3:** Data wrangling w/ dplyr ✓

**P4:** DataViz w/ ggplot ✓

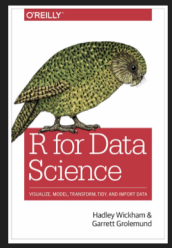
**P5:** Wrap-up w/ RMarkdown ✓



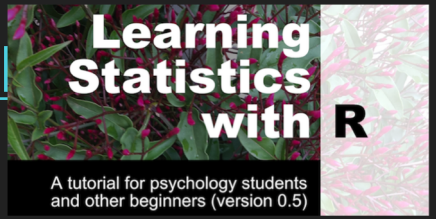
# Resources




- **Hands-On Programming with R**: *Grolemund* **#HOPR**
  - https://rstudio-education.github.io/hopr/
- **R for Data Science:** *Wickham & Grolemund* **#R4DS**
  - https://r4ds.had.co.nz
- R Programming for Data Science: Peng
  - https://leanpub.com/rprogramming
- Learning Statistics with R: Navarro
  - https://learningstatisticswithr.com/book/







# More Resources

- **#TidyTuesday** challenges 
- **gganimate**: thomasp85/gganimate 
- **tidyexplain**: gadenbuie/tidyexplain 
- **Pipes (%>%)**: datacamp.com/community/tutorials/pipe-r-tutorial
- **Radix (theme for RMarkdown)**: https://rstudio.github.io/radix/
- Google & https://stackoverflow.com/ are your best friends!


# Acknowledgements


- Arjun Krishnan, CMSE & BMB, MSU
- R-Ladies EL & my previous talks!
- The Krishnan Lab
- The R&DS books
- The R-Ladies Global community


East Lansing

# Questions? Comments?

janani@msu.edu

jananiravi.github.io

github.com/jananiravi

twitter.com/janani137

CVM, G304, MSU

East Lansing

rladies-eastlansing.github.io

JOIN US!