Introducing R/Tidyverse to Clinical Statistical Programming

MBSW 2018

Freeman Wang, @freestatman

2018-05-15

Slides available at https://bit.ly/2KNKALU

BioStat Solutions Incorporated

Where are my biases

- Biomarker Statistician
- Genomic Data Scientist and Bioinformatician
- Visualization Engineer
- R/Shiny Developer
- Long time Linux/HPC/Vim user

Where are my biases

- Biomarker Statistician
- Genomic Data Scientist and Bioinformatician
- Visualization Engineer
- R/Shiny Developer
- Long time Linux/HPC/Vim user
- SAS Certified Base and Advanced Programmer

Disclaimer

1. All the data and info in this talk are public (Twitter, GitHub).
    - CDISC example data were downloaded from: GitHub
2. This talk represents my own views, not those of BSSI.
    - BSSI does not have an opinion of which tool you should use: e.g. SAS vs R, or R/base vs R/Tidyverse.

Why? Why so popular (1/2)

- **Not** about the good-looking plots, or the fancy manipulation functions
- Content-driven and communication-focused workflow (logic-flow)
- Concisely expresses human logic as R code
  - Fast human logic I/O
  - Yourself ↔ team / customer
  - Past you → present you
- Seamlessly align multiple layers of logic, across analysis objective, programming, and output

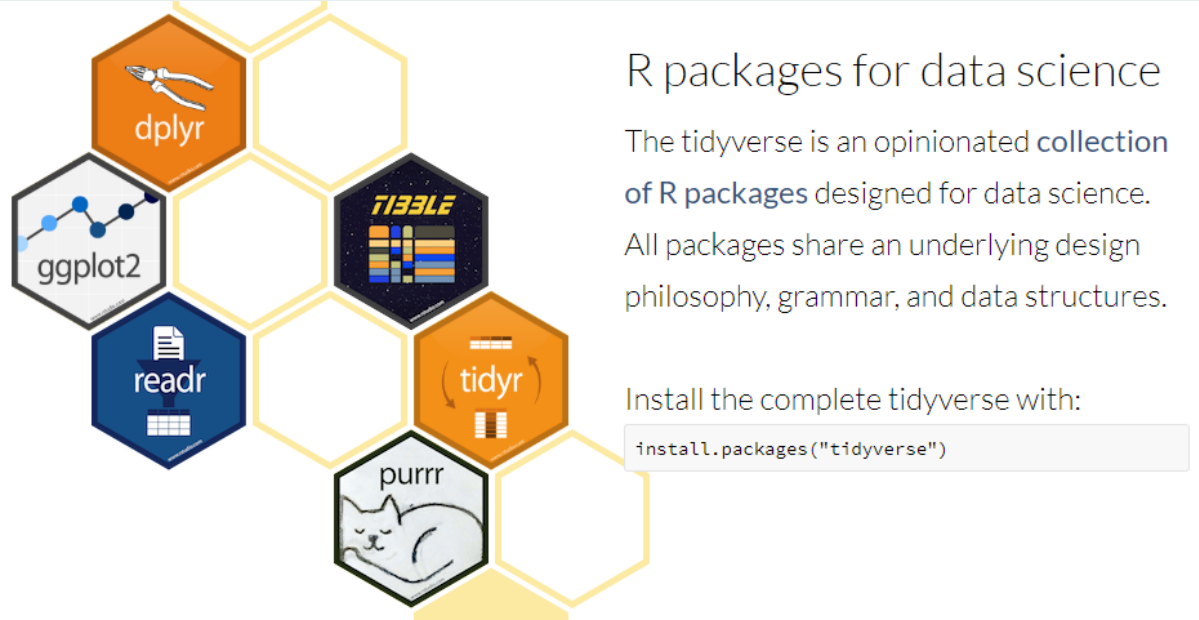
Why? Why so popular (2/2)

- structured domains of workflow, and well-defined verb/vocabulary within each domain
  - grammer of data manipulation (dplyr)
  - grammer of data visualizaiton (ggplot2)
  - grammer of statistics (not mature yet... SAS is the standard.)
- consistent design:
  - **learn it once, use it everywhere**

How? Tidy principles

1. Tidy data (Shared data structures)
2. Tidy programming API (Compose simple pieces)
3. The pipe! (functional programming for Human logic)
4. Tidy statistics

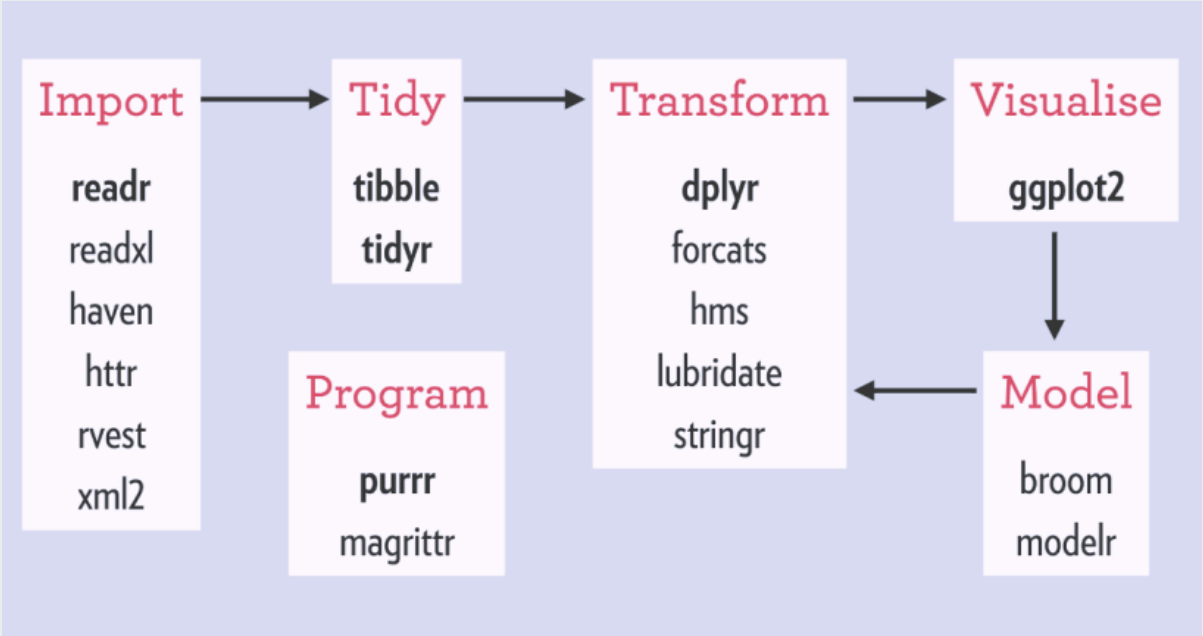
Tidyverse: core packages



https://www.tidyverse.org/

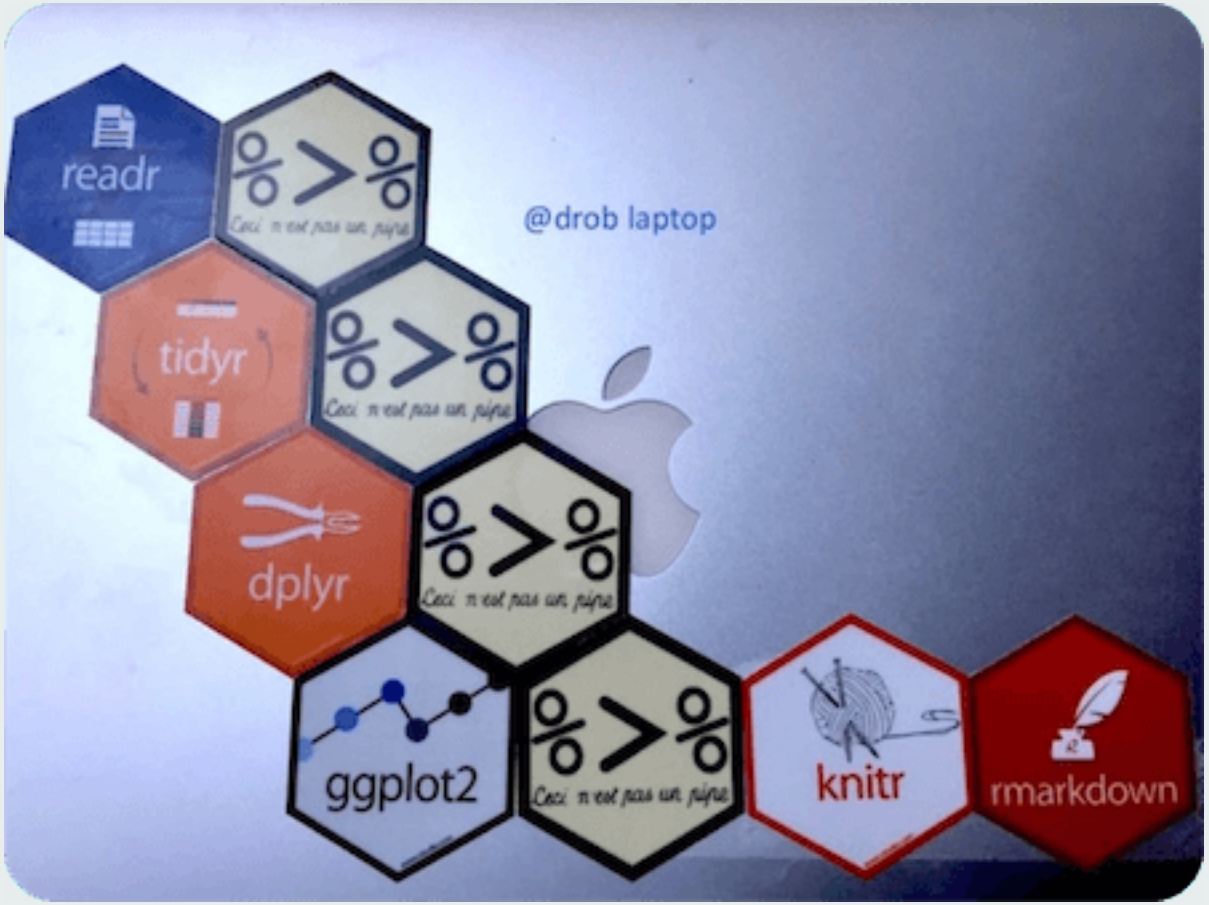
Tidyverse: more packages

Clinical programming is one of Data Science



http://r4ds.had.co.nz/

A "Real" Tidyverse Workflow



What? Tidy data

- Each row is an observation
- Each column is a variable
- Clnical data
  - Long-format is commonly seen in data storage, e.g. SDTM/ADaM
  - Wide-format is commonly seen for DEA, modeling, and visualization
  - Align manipulaiton, statistical and visualization logic with tidy data

What? Grammer of data manipulation

- `dplyr`, key verbs
  - `select` (common verb in SQL)
  - `mutate` (e.g. case_when)
  - `filter`
  - `group_by`
  - `summarize`
  - `arrange`
- Translatable to SQL
- Cheatsheet

Example of Why, How&What

What? Tidyverse extended families

From the community

- ggplot2 extention packages
  - survminer, cowplot, etc
- plotly
- summarytools
- janitor
- tidyversity
- jsmisc
- More bioconductor packages buys in!

Example

```r
library(haven)
library(tidyverse)
iris <- haven::read_sas('data/iris.sas7bdat')
adsl <- Hmisc::sasxport.get("data/adam/cdisc/adsl.xpt")
```

```
## Processing SAS dataset ADSL      ..
```

```r
adsl %>%
    select(usubjid, contains('trt')) %>%
    DT::datatable(options = list(pageLength = 3))
```

Tidy programming API: Compose simple pieces

- Tidyverse vs Base R
  - Reduce unnecessary intermediate objects
  - Keep data in relational formate as much as possible, e.g. data.frame

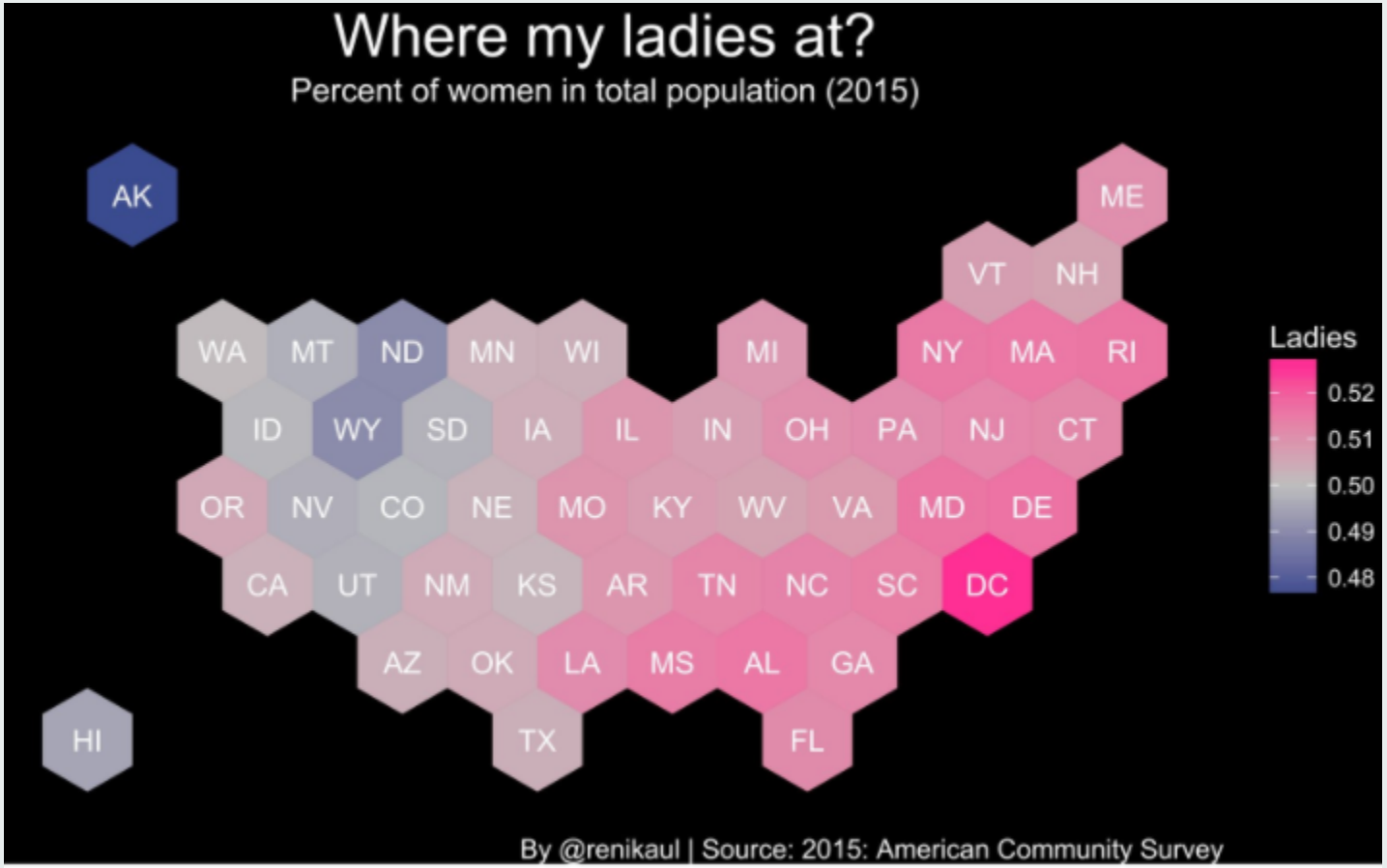
```
# base R
mtcars$pounds <- mt$wt * 1000
mtcars[["pounds"]] <- mtcars[["wt"]] / 1000
mtcars[, "pounds"] <- mtcars[, "wt"] / 1000

# Tidyverse R
mtcars <- mtcars %>%
  mutate(pounds = wt / 1000)
```

The pipe! %>%

- Conceptually the same with Unix pipe snytax
  - Push the LHS output into the 1st argument of the RHS function
- Natural representation of human logic
  - Each layer of process is a function
  - Enbrace Functional programming
- Similar philosophy to ggplot2
  - Grammar of Graphics

#TidyTuesday

#TidyTuesday



#TidyTuesday

```r
# plot inspired by @DaveBloom11

library(tidyverse)
library(geojsonio)
library(broom)
library(rgeos)

acs  <- read_csv("data/acs2015_county_data.csv")

# import hexbin map
# see blog: https://www.r-graph-gallery.com/328-hexbin-map-of-the-usa/
spdf <- geojson_read("data/us_states_hexgrid.geojson", what = "sp")

# mush data into format to link to data
spdf@data = spdf@data %>% mutate(google_name = gsub(" \\(United States\\)", "", google_name))
spdf_fortified <- tidy(spdf, region = "google_name")
# calculate center of each hex to add the label
centers <-
  cbind.data.frame(data.frame(gCentroid(spdf, byid = TRUE), id = spdf@data$iso3166_2))


hexPlot <- acs %>%
  group_by(State) %>%
  summarise(Ladies = sum(Women) / sum(TotalPop)) %>%
  right_join(spdf_fortified, by = c("State" = "id")) %>%
  ggplot() +
  geom_polygon(aes(fill = Ladies, x = long, y = lat, group = group)) +
  scale_fill_gradient2( midpoint = 0.5, low = "royalblue4", high = "deeppink", mid = "grey") +
  geom_text(data = centers, aes(x = x, y = y, label = id), color = "white") +
  theme_void() +
  coord_map() +
  labs(title = "Where my ladies at?",
       subtitle = "Percent of women in total population (2015)"
```

Tidy Statistics

`library(broom)` turns tidy output of model objects that are suited to further analysis, manipulation, and visualization.

Discussion

- R/Tidyverse is fast growing
  - Adpoting new idea
  - Some rare API change caused some pain for R package developers (OK for general users)
- Enviroment/Namespace control is a common R problem
  - Loaded functions may be over-writen by loading other packages
  - More robust usage is to add package namespace: `dplyr::select()`

Thanks for attending

Special thanks to

- Statistical Inference: A Tidy Approach @old_man_chester
- tidyverse 101: Simplifying life for useRs
- Slides created via the R package xaringan by Yihui Xie
- HTML document created via the R package rmarkdown by RStudio
- Slides and source code are available at https://github.com/freestatman/MBSW_2018_Tidyverse