

# Introducing R/Tidyverse to Clinical Statistical Programming

MBSW 2018

Freeman Wang, @freestatman

2018-05-15

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

# 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.

# Tidyverse




R packages for data science

The tidyverse is an opinionated **collection** of **R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("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  $\leftrightarrow$  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
  - Grammar of data manipulation (dplyr)
  - Grammar of data visualization (ggplot2)
- consistent design:
  - **learn it once, use it everywhere**

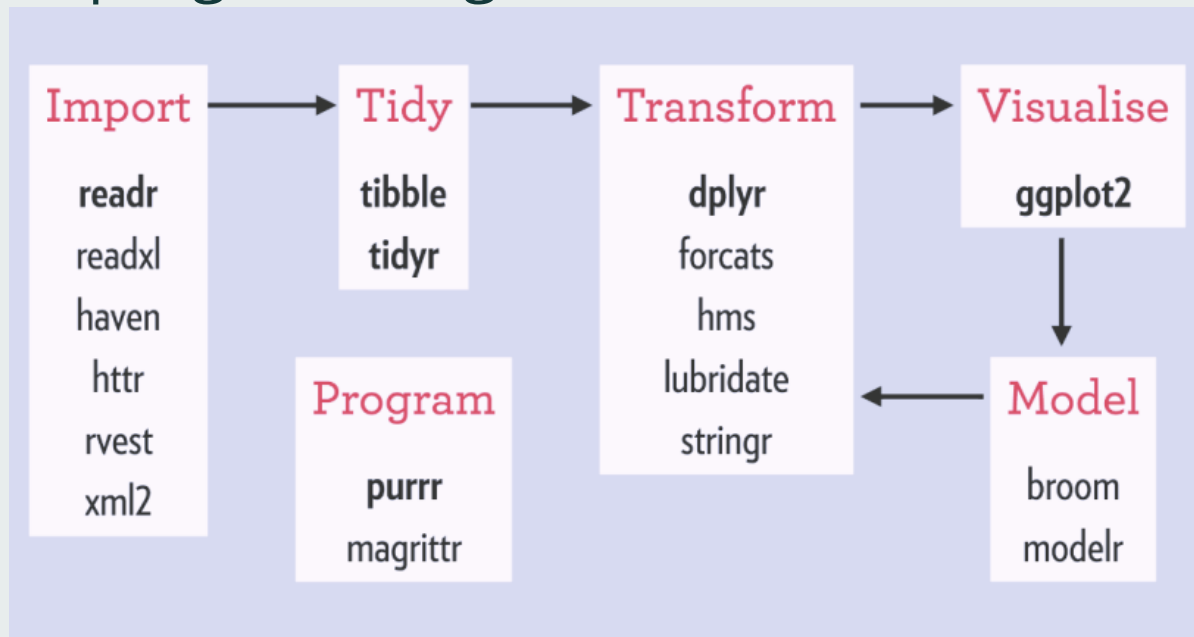
# How? Tidy principles

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

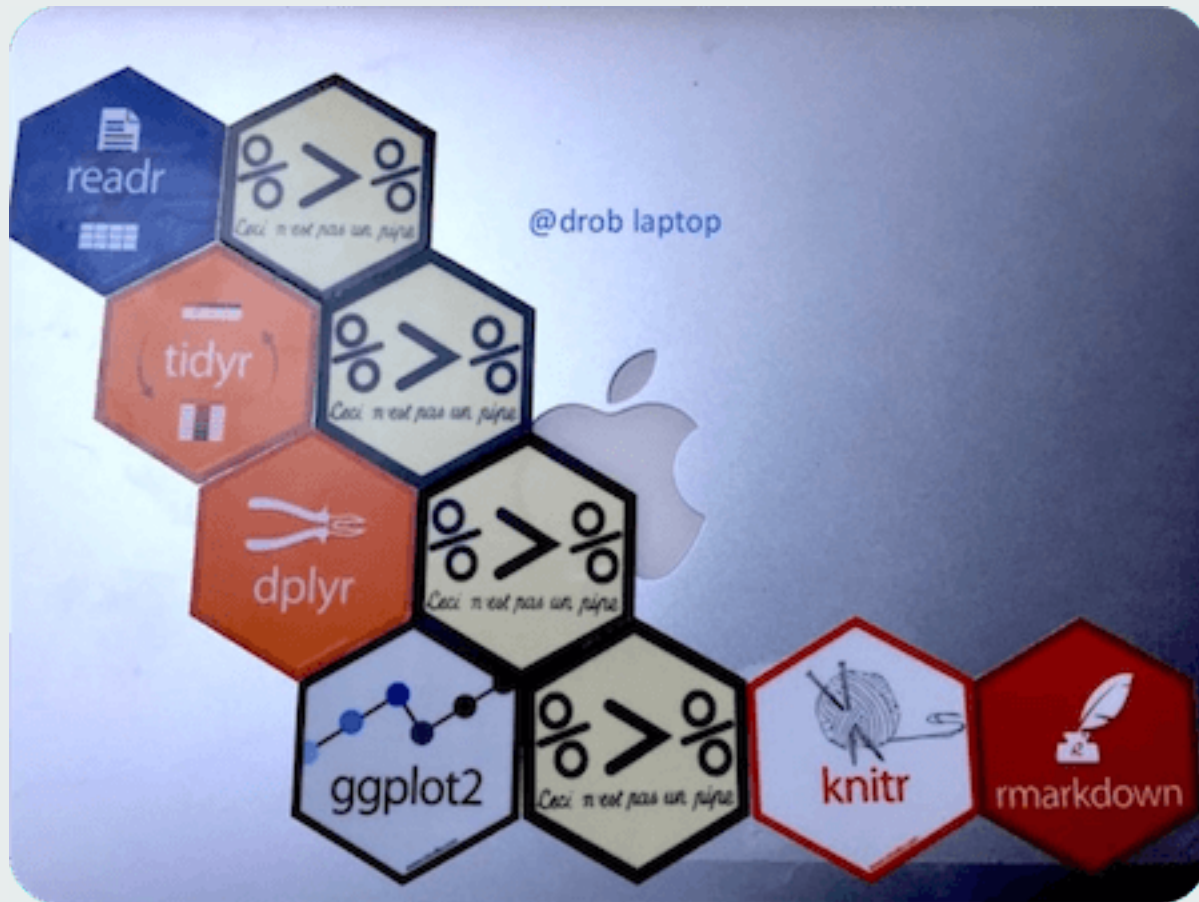


# Tidyverse: EDA (Exploratory Data Analysis) Workflow

Clinical programming is one of Data Science



# A "Real" Tidyverse Workflow



# What? Tidy data

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

# What? Grammar 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 Tidyverse vs Base R

# What? Tidyverse extended families

Tidyverse community goodies making life nice and clean

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

# Clinical Programming Example 1/3

```
library(haven)
library(tidyverse)
iris <- haven::read_sas('data/iris.sas7bdat')
ads1 <- Hmisc::sasxport.get("data/adam/cdisc/ads1.xpt")
```

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

```
ads1 %>%
  select(usubjid, contains('trt'), -starts_with('trt01a')) %>%
  DT::datatable(options = list(pageLength = 3))
```

Show  entries

Search:

	usubjid	trt01p	trt01pn	trtsdt	trtedt	trtdur
1	01-701-1015	Placebo	0	2014-01-02	2014-07-02	182
2	01-701-1023	Placebo	0	2012-08-05	2012-09-01	28
3	01-701-1028	Xanomeline High Dose	81	2013-07-19	2014-01-14	180

# Clinical Programming Example 2a/3

```
ads1 %>%  
  dplyr::group_by(trt01p) %>% # treatment-wise manipulation  
  dplyr::summarize(ave_trtdur = mean(trtdur, na.rm = TRUE),  
                  n = n()) %>%  
  knitr::kable(format = 'html')
```

trt01p	ave_trtdur	n
Placebo	149.06977	86
Xanomeline High Dose	99.39286	84
Xanomeline Low Dose	99.02381	84



# Clinical Programming Example 2b/3

```
ae_max <- adae %>%  
  group_by(usubjid) %>%      # group by patient: patient-wise manipulat  
  dplyr::arrange(aesev) %>% # sort AE severity from low to high, withi  
  slice(n()) %>%           # take the most severe level, within patie  
  ungroup() %>%  
  dplyr::select(usubjid, aesev)  
  
ae_max %>%  
  DT::datatable(options = list(pageLength = 3))
```

Show  entries

Search:

	usubjid	aesev
1	01-701-1015	MILD
2	01-701-1023	MODERATE
3	01-701-1028	MILD

Showing 1 to 3 of 225 entries

Previous

1

2

3

4

5

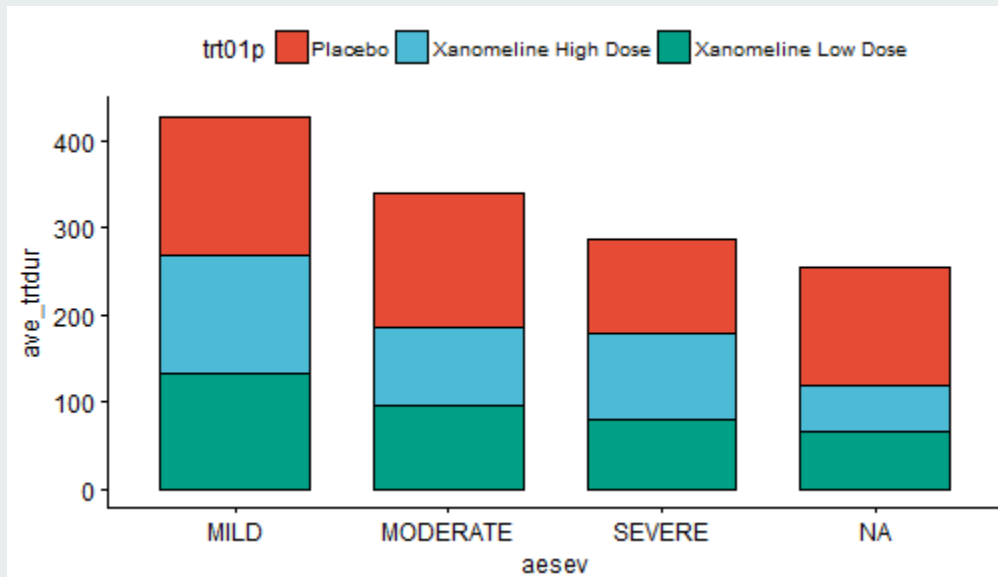
...

75

Next

# Clinical Programming Example 3/3

```
ads1 %>%  
left_join(ae_max) %>%  
group_by(trt01p, aesev) %>%  
dplyr::summarize(ave_trtdur = mean(trtdur, na.rm = TRUE), n = n()) %>%  
ggpubr::ggbarplot(x = 'aesev', y = 'ave_trtdur', fill = 'trt01p', palette = 'nord')
```



# Tidy programming API: Compose simple pieces

- Tidyverse vs Base R
  - Reduce unnecessary intermediate objects (e.g. index, dummy variables), save your brain memory bandwidth
  - Data lives in structured data.frame, instead of, individual value or vector
  - Better default values, e.g.  
`stringsAsFactors=FALSE, drop=FALSE`

**R Inferno:** An essential guide to the trouble spots and oddities of R

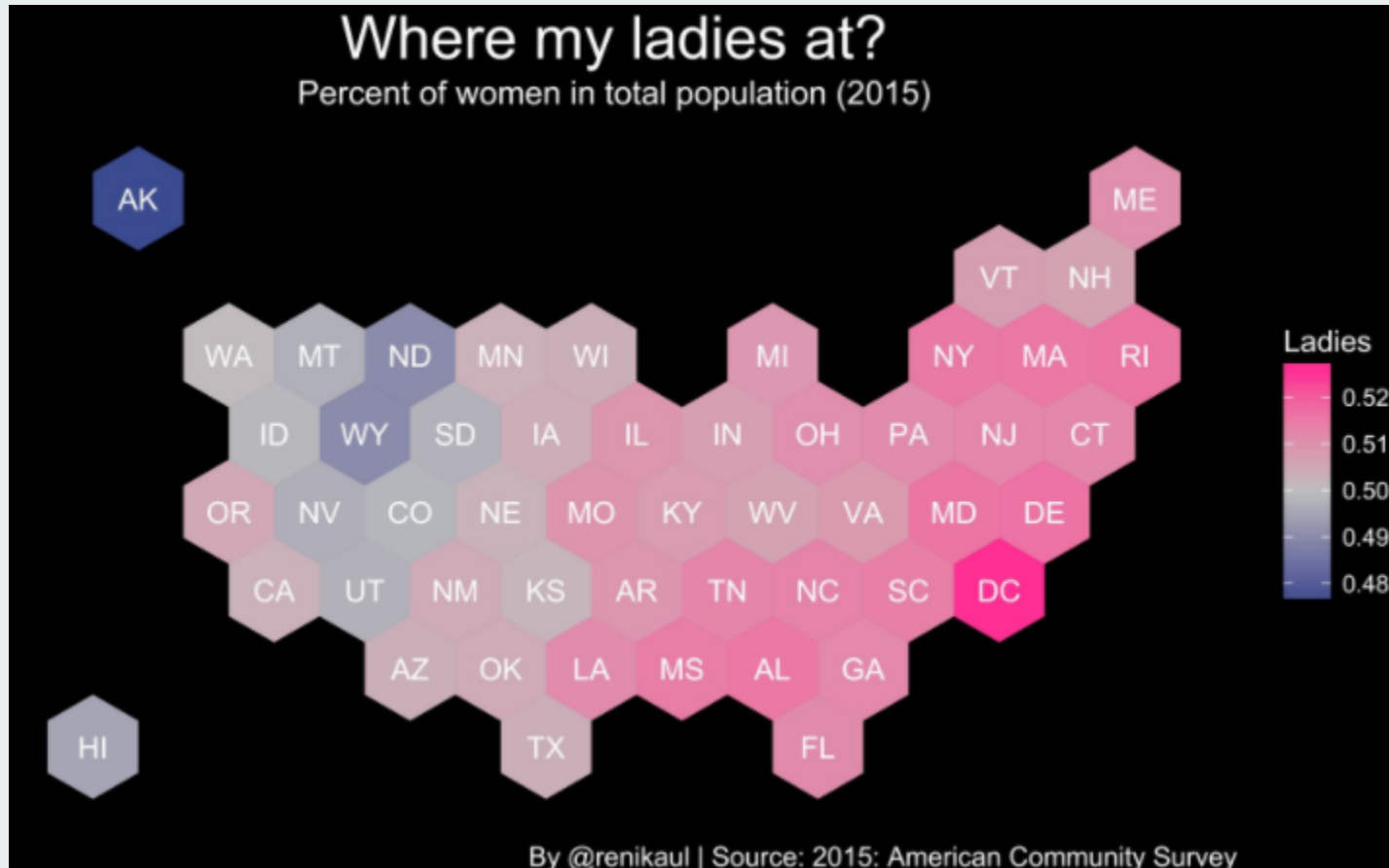
# The pipe! %>%

- Conceptually the same with Unix pipe syntax
  - Push the LHS output into the 1st argument of the RHS function
- Natural representation of human logic
  - Each processing layer is a function
  - Embrace functional programming
- Similar philosophy to `ggplot2`
  - Grammar of Graphics

# #TidyTuesday 1/3



# #TidyTuesday 2/3



# #TidyTuesday 3/3

```
# 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
  - Keep adopting new ideas
  - Some rare API change caused pain for R package developers (OK for general users)
- Environment/Namespace control is a common R problem
  - Loaded functions may be over-written by later loaded packages (or the packages' dependency packages...)
  - More robust usage is to add package namespace: `dp1yr::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](https://github.com/freestatman/MBSW_2018_Tidyverse)

