Introducing **R/Tidyverse** to Clinical Statistical Programming

MBSW 2018

Freeman Wang, @freestatman

2018-05-15

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:

```r
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 ↔ 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
Clinical programming is one type 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? dplyr: Grammar of data manipulation

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

- Translatable to SQL code,

- Cheatsheet
What? Tidyverse extended families

Tidyverse community goodies making life nice and clean

- ggplot2 extension packages
  - survminer, cowplot, ggpubr, etc
- plotly
- summarytools
- janitor
- tidyversity
- jsmisc
- More bioconductor packages buy in!
# Clinical Programming Example 1/3

```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 ..

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

<table>
<thead>
<tr>
<th>usubjid</th>
<th>trt01p</th>
<th>trt01pn</th>
<th>trtsdt</th>
<th>trtedt</th>
<th>trtdur</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-701-1015</td>
<td>Placebo</td>
<td>0</td>
<td>2014-01-02</td>
<td>2014-07-02</td>
<td>182</td>
</tr>
<tr>
<td>01-701-1023</td>
<td>Placebo</td>
<td>0</td>
<td>2012-08-05</td>
<td>2012-09-01</td>
<td>28</td>
</tr>
<tr>
<td>01-701-1028</td>
<td>Xanomeline</td>
<td>81</td>
<td>2013-07-19</td>
<td>2014-01-14</td>
<td>180</td>
</tr>
</tbody>
</table>
```r
ads1 %>%
  group_by(trt01p) %>% # treatment-wise manipulation
  summarize(ave_trtdur = mean(trtdur, na.rm = TRUE),
            n = n()) %>%
  knitr::kable(format = 'html')
```

<table>
<thead>
<tr>
<th>trt01p</th>
<th>ave_trtdur</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placebo</td>
<td>149.06977</td>
<td>86</td>
</tr>
<tr>
<td>Xanomeline High Dose</td>
<td>99.39286</td>
<td>84</td>
</tr>
<tr>
<td>Xanomeline Low Dose</td>
<td>99.02381</td>
<td>84</td>
</tr>
</tbody>
</table>
Clinical Programming Example 2b/3

```r
ae_max <- adae %>%
group_by(usubjid) %>%
arrange(aesev) %>%
slice(n()) %>%
ungroup() %>%
select(usubjid, aesev)

ae_max %>%
DT::datatable(options = list(pageLength = 3))
```

<table>
<thead>
<tr>
<th>usubjid</th>
<th>aesev</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-701-1015</td>
<td>MILD</td>
</tr>
<tr>
<td>01-701-1023</td>
<td>MODERATE</td>
</tr>
<tr>
<td>01-701-1028</td>
<td>MILD</td>
</tr>
</tbody>
</table>

Showing 1 to 3 of 225 entries
Clinical Programming Example 3a/3

```r
adsl %>%
  left_join(ae_max) %>%
  filter(!is.na(aesev)) %>%
  group_by(aesev) %>%
  summarize(ave_trtdur = mean(trtdur, na.rm = TRUE), n = n()) %>%
  ggpubr::ggbarplot(x = 'aesev', y='ave_trtdur', fill='aesev', palette='RdYlGn')
```
Clinical Programming Example 3b/3

```r
ads1 %>%
  left_join(ae_max) %>%
  filter(!is.na(aesev)) %>%
  group_by(aesev, trt01p) %>%
  summarize(ave_trtdur = mean(trtdur, na.rm = TRUE), n = n()) %>%
  ggpubr::ggbarplot(x = 'aesev', y = 'ave_trtdur', fill = 'aesev', palette = 'Set1',
                    facet_grid(~trt01p) +
                    theme(axis.text.x = element_text(size = 8)))
```
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 argument values, e.g. `stringsAsFactors=FALSE` in `tibble` package.
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 argument values, e.g. `stringsAsFactors=FALSE` in `tibble` package.

*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 3/3

```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 = "deppink", 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)",
       caption = "Source: acs2015_county_data.csv")
```

```
Tidy Statistics

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

- R/Tidyverse ecosystem is fast growing
  - Keep adopting new ideas, e.g. non-standard evaluation (NSE)
  - Some deep-level API change may cause pain for R package developers (OK for general users)
- Function name collision is a common R problem
  - Functions may be over-written by later loaded packages (or the packages' dependency packages...)
  - More robust function calling 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