An Overview of the Tidyverse

This document is based on the Michael Levy’s presentation at the Davis R-Users’ Group.

* The youtube video of the presentation is available [here.](https://www.youtube.com/watch?v=_rPhSAVhs1A)
* The original github repository of the presentation is [here.](https://github.com/michaellevy/tidyverse_talk)

## What is the tidyverse?

A a suite of R tools that follow a tidy philosophy:

### Tidy Philosophy

Put data in data frames

* Each variable gets a column
* Each observation gets a row
* Each unit of analysis gets a data frame

### Tidy APIs

Functions should be consistent and easily (human) readable

* Take one step at a time
* Connect simple steps with the pipe
* Referential transparency

### Okay but really, what is it?

Suite of ~20 packages that provide consistent, user-friendly, smart-default tools to do most of what most people do in R.

* Core packages: ggplot2, dplyr, tidyr, readr, purrr, tibble
* Data import: DBI, haven, httr, jsonlite, readxl, rvest, xml2
* Specialized data manipulation: hms, stringr, lubridate, forcats
* Modeling: modelr, broom

install.packages(tidyverse) installs all of the above packages.

library(tidyverse) attaches only the core packages.

library(tidyverse)

## tibble

A modern reimagining of a data frame.

tdf <- tibble(x = 1:1e4, y = rnorm(1e4))  
class(tdf)

## [1] "tbl\_df" "tbl" "data.frame"

Tibbles print politely.

tdf

## # A tibble: 10,000 x 2  
## x y  
## <int> <dbl>  
## 1 1 1.56   
## 2 2 0.0664  
## 3 3 0.985   
## 4 4 0.618   
## 5 5 -0.206   
## 6 6 -0.358   
## 7 7 0.515   
## 8 8 -0.285   
## 9 9 -1.47   
## 10 10 -0.740   
## # ... with 9,990 more rows

* Can customize print methods with print(tdf, n = rows, width = cols)
* Set default with options(tibble.print\_max = rows, tibble.width = cols)

Tibbles have some convenient and consistent defaults that are different from base R data.frames.

* In tibbles strings are NOT automatically reconized as factors

Also note that tidyverse import functions, such as readr::read\_csv, default to tibbles and that *this can break existing code*.

## The pipe %>% : Functional composition

Sends the output of the LHS function to the first argument of the RHS function.

piping <- sum(1:8) %>%   
 sqrt()  
piping

## [1] 6

Note that keyboard shortcut for the pipe is cmd + shift + M

## dplyr

A package for data manipulation