Lab Assignment: Chapter 15 - Reshaping Data in the Tidyverse

Michael Beebe

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tibble)  
library(readr)  
library(tidyr)

## 1. Binding Rows and Columns

Create 2 tibbles:

* tbl1 will have the numbers 1-5 as IDs, and the first names of 5 characters from your favorite TV show
* tbl2 will have the numbers 1-4, & 6 as IDs, and the last names of the characters 1-4 above, and the last name of a different character that doesn’t match from tbl1

tibble1 <- tibble(  
 ID = c(1:5),  
 F\_Names = c("Michael", "Dwight", "Jim", "Pam", "Toby")  
)  
  
tibble2 <- tibble(  
 ID = c(1:4, 6),  
 L\_Names = c("Scott", "Schrute", "Halpert", "Beasley", "Malone")  
)  
  
tibble1

## # A tibble: 5 x 2  
## ID F\_Names  
## <int> <chr>   
## 1 1 Michael  
## 2 2 Dwight   
## 3 3 Jim   
## 4 4 Pam   
## 5 5 Toby

tibble2

## # A tibble: 5 x 2  
## ID L\_Names  
## <dbl> <chr>   
## 1 1 Scott   
## 2 2 Schrute  
## 3 3 Halpert  
## 4 4 Beasley  
## 5 6 Malone

Join the rows and print the results to the screen

tibble3 <- full\_join(tibble1, tibble2)

## Joining, by = "ID"

tibble3

## # A tibble: 6 x 3  
## ID F\_Names L\_Names  
## <dbl> <chr> <chr>   
## 1 1 Michael Scott   
## 2 2 Dwight Schrute  
## 3 3 Jim Halpert  
## 4 4 Pam Beasley  
## 5 5 Toby <NA>   
## 6 6 <NA> Malone

## 2. Joins with dplyr

Add a line or lines to your script that will join the tables from step 1 above into 1 table

* Use the ID field to join the tables
* Use a left join and a take a screen shot of your results
* Use a right join and take a screen shot of your results

left <- left\_join(  
 tibble1,  
 tibble2,  
 by = "ID"  
)  
  
left

## # A tibble: 5 x 3  
## ID F\_Names L\_Names  
## <dbl> <chr> <chr>   
## 1 1 Michael Scott   
## 2 2 Dwight Schrute  
## 3 3 Jim Halpert  
## 4 4 Pam Beasley  
## 5 5 Toby <NA>

right <- right\_join(  
 tibble1,  
 tibble2,  
 by = "ID"  
)  
  
right

## # A tibble: 5 x 3  
## ID F\_Names L\_Names  
## <dbl> <chr> <chr>   
## 1 1 Michael Scott   
## 2 2 Dwight Schrute  
## 3 3 Jim Halpert  
## 4 4 Pam Beasley  
## 5 6 <NA> Malone

## 3. Converting Data Formats

Use the read\_tsv file function to read the “reaction.txt” file from the book site

* Store the data into a variable “emotion”
* Print emotion to the screen

emotion <- read\_tsv("../../Data/reaction.txt")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## ID = col\_double(),  
## Test = col\_double(),  
## Age = col\_double(),  
## Gender = col\_character(),  
## BMI = col\_double(),  
## React = col\_double(),  
## Regulate = col\_double()  
## )

head(emotion)

## # A tibble: 6 x 7  
## ID Test Age Gender BMI React Regulate  
## <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 1 1 9.69 F 14.7 4.17 3.15  
## 2 1 2 12.3 F 14.6 3.89 2.55  
## 3 2 1 15.7 F 19.5 4.39 4.41  
## 4 2 2 17.6 F 20.0 2.03 2.2   
## 5 3 1 9.52 F 20.9 3.38 2.65  
## 6 3 2 11.8 F 24.0 4 3.63

Gather the data with a key = Type and value = Measurement

emotion %>%  
 gather(  
 key = Type,  
 value = Measurement  
 )

## # A tibble: 693 x 2  
## Type Measurement  
## <chr> <chr>   
## 1 ID 1   
## 2 ID 1   
## 3 ID 2   
## 4 ID 2   
## 5 ID 3   
## 6 ID 3   
## 7 ID 4   
## 8 ID 4   
## 9 ID 5   
## 10 ID 5   
## # … with 683 more rows

Arrange the data from the step above by ID

emotion %>%  
 arrange(ID)

## # A tibble: 99 x 7  
## ID Test Age Gender BMI React Regulate  
## <dbl> <dbl> <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 1 1 9.69 F 14.7 4.17 3.15  
## 2 1 2 12.3 F 14.6 3.89 2.55  
## 3 2 1 15.7 F 19.5 4.39 4.41  
## 4 2 2 17.6 F 20.0 2.03 2.2   
## 5 3 1 9.52 F 20.9 3.38 2.65  
## 6 3 2 11.8 F 24.0 4 3.63  
## 7 4 1 16.3 M 25.1 3.15 3.59  
## 8 4 2 18.8 M 28.0 3.02 3.54  
## 9 5 1 15.8 M 28.4 3.08 2.64  
## 10 5 2 18.2 M 19.6 3.17 2.29  
## # … with 89 more rows