Reshaping Data in R with *tidyr*

Sean Hellingman ©

Data Visualization and Manipulation through Scripting (ADSC1010)

shellingman@tru.ca

Fall 2024

THOMPSON RIVERS UNIVERSITY

Topics

Introduction

- Challenging to adjust exactly how the columns and rows are used to represent the data.
- Structuring data frames to properly create visualisations, statistical models, or implement machine learning algorithms is extremely important.
- We can use the *tidyr* package to help with this.

"Tidy" Data

- Need to decide on a desired structure of your data frame.
- Different questions will lead to different requirements in data structures.
- Example: We would like to create time series or econometric models.
- Principles:
  1. Each variable is a column.
  2. Each observation is row.
  3. Each value is a cell.

Package Information

- The *tidyr* package is part of the *tidyverse* collection of R packages.
- We need to install and load the package prior to use:
  - `if(!require(tidyr)) install.packages("tidyr")`
    `library(tidyr)`
- Once the package is loaded we can use the existing functions as we please.

Wide and Long Data

Wide

Class	Variable.1	Variable.2	Variable.3
A	89	55	43
B	88	67	54
C	76	45	77
D	98	45	56

Long

Class	Variable	Value
A	Variable.1	89
A	Variable.2	55
A	Variable.3	43
B	Variable.1	88
B	Variable.2	67
B	Variable.3	54
C	Variable.1	76
C	Variable.2	45
C	Variable.3	77
D	Variable.1	98
D	Variable.2	45
D	Variable.3	56

Columns to Rows `gather()`

- In some cases we need to **gather** all values into the same column.
- The `gather()` function collects data values stored in multiple columns and stores them in a single column.
- It creates two columns representing key-value pairs of the *group* and its value.

`gather()` Main Arguments

1. The first argument is a data frame.
2. `key` = Name of the column(s) listing the gathered values.
3. `value` = Name of the new column for all of the gathered values.
4. column(s) to gather data from (similar to the `select()` function).

Example 1

- Load the *tidyr* package into R.
- Import the TicketPrices.csv file into your working environment.
- Take a moment to examine the data.
- Use the `gather()` function to put the ticket prices into a single column.

Rows to Columns `spread()`

- In some cases we need to **spread** the values into different columns.
- The `spread()` function creates multiple variables from two existing columns.
- Takes arguments similar to the `gather()` function but applies them in the opposite direction.

`spread()` Main Arguments

1. The first argument is a data frame.
2. `key` = Name of the column indicating where to get new variable names.
3. `value` = Name of the column indicating where to get the new values from.

Example 2

- Use the `spread()` function to put the ticket prices into multiple columns for each **competition**.
- Use the `spread()` function to put the ticket prices into multiple columns for each **club**.

Widen Data with `separate()`

- The `separate()` function turns a single character column into multiple columns.
- Sometimes we want to split characters into multiple variables.
- Useful for separating name variables or place variables.
  - `separate(data.frame, col.to.expand, names.of.new.variables, sep = "", remove = TRUE)`

Narrow Data with `unite()`

- The `unite()` function turns two variables into one.
- Sometimes we want to combine variables as characters
- Essentially the opposite of the `separate()` function.
  - `unite(data.frame, new.col.name, col.1, col.2, ..., sep = "_")`

Example 3

- Create a data frame of first and last names.
- Use the `separate()` function to split the name character variable into two variables.
- Use the `unite()` function to merge them again.

Exercise 1

- Run the code to create the data called *dataframe* found under the Exercise 1 heading in the ADSC1010_Reshaping_Data_in_R_with_tidyr_Examples.Rmd file.

- Use the `gather()` function to convert the data to long data.

Exercise 2

- Using the long data you created in Exercise 1, separate the group variable into two separate columns.

Exercise 3

- Convert your long data back into wide data. You may have to account for the new columns you have made.

References & Resources

1. Douglas, A., Roos, D., Mancini, F., Couto, A., & Lusseau, D. (2023). *An introduction to R*. Retrieved from https://intro2r.com/

2. Michael Freeman, Joel Ross, *Programming Skills for Data Science: Start Writing Code to Wrangle, Analyze, and Visualize Data with R*, 2019, ISBN-13: 978-0-13-513310-1

3. Baumer, Kaplan, Horton. *Modern Data Science with R.* Retrieved from https://mdsr-book.github.io/mdsr3e/.

- https://www.tidyverse.org/
- https://tidyr.tidyverse.org/