Online Academic Data Analysis Bootcamp Using Open-Access Program R: Essentials

Patrick Njage

Aug 17th 2023

# The leadership data

A researcher studied how men and women differ in the ways they lead their organizations. Typical questions might be:

* Do men and women in management positions differ in the degree to which they defer to superiors?
* Does this vary from country to country, or are these gender differences universal?

One way to address these questions is to have bosses in multiple countries rate their managers on deferential behavior, using questions like the following: *This manager asks my opinion before making personnel decisions.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| strongly disagree | disagree | neither agree nor disagree | agree | strongly agree |

In the resulting data, each row represents the ratings given to a manager by his or her boss. Here, each manager is rated by their boss on five statements (q1 to q5) related to deference to authority. For example, manager 1 is a 32-year-old male working in the US and is rated deferential by his boss, while manager 5 is a female of unknown age (99 probably indicates missing) working in the UK and is rated low on deferential behavior. The date column captures when the ratings were made. Although a dataset might have dozens of variables and thousands of observations, we’ve only included 10 columns and 5 rows to simplify the examples. Additionally, we’ve limited the number of items pertaining to the managers’ deferential behavior to 5. In a real-world study, you’d probably use 10–20 such items to improve the reliability and validity of the results. You can create a data frame containing the data using the following code.

# Creating the leadership data frame

manager <- c(1, 2, 3, 4, 5)  
date <- c("10/24/08", "10/28/08", "10/1/08", "10/12/08", "5/1/09")  
country <- c("US", "US", "UK", "UK", "UK")  
gender <- c("M", "F", "F", "M", "F")  
age <- c(32, 45, 25, 39, 99)  
q1 <- c(5, 3, 3, 3, 2)  
q2 <- c(4, 5, 5, 3, 2)  
q3 <- c(5, 2, 5, 4, 1)  
q4 <- c(5, 5, 5, NA, 2)  
q5 <- c(5, 5, 2, NA, 1)  
leadership <- data.frame(manager, date, country, gender, age,   
 q1, q2, q3, q4, q5, stringsAsFactors=FALSE)

In order to address the questions of interest, we must first address several data management issues. Here’s a partial list:

* The five ratings (q1 to q5) will need to be combined, yielding a single mean deferential score from each manager.
* In surveys, respondents often skip questions. For example, the boss rating manager 4 skipped questions 4 and 5. We’ll need a method of handling incomplete data. We’ll also need to recode values like 99 for age to missing.
* There may be hundreds of variables in a dataset, but we may only be interested in a few. To simplify matters, we’ll want to create a new dataset with only the variables of interest.
* Past research suggests that leadership behavior may change as a function of themanager’s age. To examine this, we may want to recode the current values of ageinto a new categorical age grouping (for example, young, middle-aged, elder).
* Leadership behavior may change over time. We might want to focus on deferen- tial behavior during the recent global financial crisis. To do so, we may want to limit the study to data gathered during a specific period of time (say, January 1, 2009 to December 31, 2009).

# Exercise 1

Let’s say that you want to recode the ages of the managers in our leadership dataset from the continuous variable *age* to the categorical variable *agecat* (Young, Middle Aged, Elder).

1. First, recode the value 99 for age to missing with NA.
2. Change the variables *manager* to *managerID* and *date* to *testDate*.
3. The date is coded as a character variable in mm/dd/yy format. Please replace it in the data frame as a date variable.
4. Create a new dataset containing rows sorted:

* from youngest manager to oldest manager.
* into female followed by male, and youngest to oldest within each gender.
* by gender, and then from oldest to youngest manager within each gender.

1. Select variables q1, q2, q3, q4, and q5 from the leadership data frame and save them to the data frame *newdata*.

# Exercise 2

Selecting or excluding observations (rows) is typically a key aspect of successful data preparation and analysis.

1. Select all *men over 30* from the leadership dataframe.
2. Select all rows that have a value of age greater than or equal to 35 or age less than 24 and keep the variables q1 through q4.
3. Select all men over the age of 25 and keep variables gender through q4 (gender, q4, and all columns between them). *Hint*: the colon operator from:to provides all variables in a data frame between the *from* variable and the *to* variable, inclusive.
4. Take a random sample of size 3 from the leadership dataset without replacement.
5. Transpose the leadership dataframe so that the column names (variable names) become the row names.
6. Aggregate the *mtcars* data by number of cylinders (cyl) and gears (gear), returning means on each of the numeric variables. The mtcars data is part of the R package (run help(mtcars) to view its description)

# Exercise 3

Consider the provided “state\_income” data which contains hypothetical income generated by US states from year 2002 to 2015. For this exercise, please use use the package Dplyr. For a *cheatsheet* visit <https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf> or for direct access in Rstudio, please click Help -> Cheatsheets -> Data Transformation with Dplyr

1. Suppose you are asked to select only a few variables. Selects variables “Index”, columns from “State” to “Y2008”. *Hint:* use select() function.
2. Select (call this mydata2) and then drop variables (call this mydata3) starts with ‘Y’. *Hint:* The starts\_with() function is used to select variables starts with an alphabet. Adding a negative sign before starts\_with() implies dropping the variables starts with the alphabet.
3. Suppose you need to subset the data. Please filter rows and retain only those values in which Index is equal to A. *Hint:* use filter() function.
4. Calculat the mean and median for the variable Y2015. *Hint:* use summarise() function.
5. Calculating count and mean of variables Y2011 and Y2012 by variable Index.

# Exercise 4: combining datasets using join() function

Generate the two datasets by running the following codes:

df1 = data.frame(ID = c(1, 2, 3, 4, 5),  
w = c('a', 'b', 'c', 'd', 'e'),  
x = c(1, 1, 0, 0, 1),  
y=rnorm(5),  
z=letters[1:5])  
df2 = data.frame(ID = c(1, 7, 3, 6, 8),  
a = c('z', 'b', 'k', 'd', 'l'),  
b = c(1, 2, 3, 0, 4),  
c =rnorm(5),  
d =letters[2:6])

1. Merge df1 and df2 with ID as common variable (primary key). *Hint:* use inner\_join
2. Select all rows from the left table (df1), even if there are no matches in the right table (df2). *Hint:* use left\_join
3. Include rows of df1 that match df2 but only keep the columns from df1. *Hint:* use semi\_join()
4. Select the opposite of what was selected in c) above. *Hint:* use anti\_join()