Introduction to the R Statistical Computing Environment
Data in R: Exercises

John Fox
(McMaster University)
ICPSR

2021

1. Read data from various sources into R data frames:

- Directly from the keyboard.
- Using the data editor `fix()`.
- From a text file in which the data values are delimited by white space.
- From a SAS, SPSS, or Stata data file, using the `Import()` function in the **car** package.
- * From an Excel spreadsheet using the `Import()` function in the **car** package. (The file `Prestige.xlsx` is supplied on the website for the lectures.)

2. Explore the properties of various kinds of objects:

- Create a character vector, a numeric vector, a logical vector, a character matrix, a numeric matrix, a factor, a data frame, a tibble, a list, and a function.
- Apply each of the following functions to these objects: `length()`, `class()`, `mode()`, `typeof()`, and `attributes()`.
- Look at the help files for each of these functions – e.g., `?length`.
- What did you learn?

3. R has a number of "coercion" functions, prefixed with `as.`, and a number of "predicate" functions, prefixed with `is.`: for example, `as.matrix` and `is.matrix`.

- Get a complete list of these functions via the commands `apropos("^as\\.")` and `apropos("^is\\.")`. *Note*: The quoted arguments to `apropos()` are "regular expressions" — a powerful notation for searching text that will be familiar to Unix users; see `?regex` and section 2.4 of the *R Companion* for how regular expressions are used in R.
- Using the objects created in the previous exercise, experiment with (for example) the coercion functions `as.matrix` , `as.vector`, and `as.character`, and with the predicates `is.vector` and `is.character`. What did you learn?

4. * *Merging:* A common operation in data management is to *merge* data from two or more sources into a rectangular data set. There are many functions in the standard R distribution and in contributed CRAN packages for performing merges. Consider the following example:

The `MathAchieve` data frame[1] in the **nlme** package contains data on 7185 high-school students in 160 high schools, and includes the variable `School`, which is an ordered factor giving the school ID number for each student. The separate data frame `MathAchSchool` contains data on the 160 schools, and also contains the factor (*not* ordered factor) `School` with the (same) school IDs. See `?MathAchieve` and `?MathAchSchool` for more information on these data sets.

- Using the standard R `merge()` function (consult `?merge`), merge the two data sets so that the appropriate school-level data in `MathAchSchool` is associated with each student in the individual-level data in `MathAchieve`.
- The resulting merged data set should have two versions of the `MEANSES` variable—`MEANSES.x` originating from the `MathAchieve` data set and `MEANSES.y` from the `MathAchSchool` data set. If you performed the merge correctly, these two variables should have identical values (check it!). Delete one copy of the variable from the data frame and rename the other as `"MEANSES"`.
- Optionally repeat the merge using the `left_join()` function in the **dplyr** Tidyverse package (see `?left_join`). *Hint:* Before performing the merge, you'll have to convert the `School` variable in one of the data sets so that it's of the same class as in the other data set (or just convert both to character variables).
- The variable `MEANSES` is a *compositional variable*,[2] that is a variable describing a higher-level unit (here, schools) that is aggregated from properties of individuals (here, students in each school). Optionally recompute school-mean SES for each school using the individual-level variable SES and merge recomputed school-mean SES (give it another name, like `MEANSES2`) with the individual-level data. Finally, compare the original `MEANSES` variable with your recomputed `MEANSES2`,. Are they the same? If not, how do they differ?

[1] `MathAchieve` is actually an object of compound class `c("nfnGroupedData", "nfGroupedData" "groupedData", "data.frame")`; it consequently "inherits" from the `"data.frame"` class and can be treated as a data frame.

[2] A variable like this is sometimes called a *contextual variable*, a term that I prefer to reserve for direct properties of higher-level units, such as the variable `Sector` (Catholic or Public) in the school-level `MathAchSchool` data set.