EKT-816 Lecture 1

Introduction to R (1)

Jesse Naidoo

University of Pretoria

Visualizing Distributions

- most basic thing to do: `view()` to inspect the raw data!
- assumes data is a “tibble”
- this is a tidyverse object, basically like a `data.frame` in base R
- will show you the data type (int, double, chr) for each column (variable)
- univariate distributions:
- `geom_freq`: like “kdensity” in STATA
- `geom_histogram`; try altering the bin width to spot unusual spikes
    - this occurs frequently in South African income data
    - see Wittenberg (2017)
- joint distributions:
- `geom_point()` generates scatterplots
- `geom_smooth()` plots a nonparametric estimate of the conditional mean
- can use `geom_quantile()` for conditional quantiles
- for conditional distributions, `geom_bar()` can be a good option (esp. with categorical variables)

Examining Metadata and Summary Statistics

- if your data is a “tibble”, just typing its name will
- show you the first few observations
- display variable names and data types for each
- number of missing values:
- `skim()` from `skimr` package will display this as well as other summary stats
- interacts well with `group_by()`
- `n_distinct`: number of distinct values (takes vector argument)
- categorical variables (“factors” in R): see ch. 15 of “R for Data Science”

Scripts

- a *script* (a `.R` file) is just a list of instructions that R executes in order
- can become complicated, if you need:
- call other scripts
- load, manipulate, and save data
- generate and save plots or estimation output
- an RStudio "project" is a built-in implementation of the idea that everything should be
- local (no absolute filepaths) and portable
- the `.Rproj` file reminds R to save certain configurations
  - e.g. related to the treatment of certain data types (strings, factors)
  - command history and loaded libraries
- it's often good practice to:
- hit Ctrl-Shift-F10 to restart R
- then rerun your script (to make sure it works as expected)

Good Housekeeping

- *never* have R save your workspace and data at exit
- put `rm(ls())` and `gc()` at the head of every script you write
- do all your work in scripts
- prototype by running scripts line-by-line
- this will prevent you from making costly mistakes
    - e.g. you forgot that you had some other data in memory; months later, your code breaks
    - even worse, the code does *not* break, but the results change - why?
- consult style.tidyverse.org for "good enough" practices

Further Reading

- Gentzkow and Shapiro's "Code and Data for the Social Sciences" handbook
- see NBER SI talk here (link)
- a lot of these practices are useful even if your data are not "big"
- automation (scripting); version control
- good folder structure; code style
- data management
- "R for Data Science" is extremely useful
- StackOverflow usually has the answer to your questions
- Google the error message!
- Tyler Ransom's "Data Science for Economists" course: github.com/tyleransom/DScourseS18
- for R Markdown
- `rmarkdown.rstudio.com`
- Steven Miller's blog has lots of useful advice and customization tips: svmiller.com/blog

Table of Contents