Homework 5

Cody

October 19, 2017

Overview

Due anytime Friday 2017-10-20.

Goals:

- Reorder a factor in a principled way based on the data and demonstrate the effect in arranged data and in figures.
- Write some data to file and bring it back into R.
- Remake at least one previously made figure (or make a completely new figure), using new knowledge, such as ability to control the color scheme, empowerment about factor levels, better figure making mechanics, or visualization design principles.
- Write a figure to file explicitly and include it your R Markdown report via `![Alt text](/path/to/img.png)`.
- Clean up and organize your entire repository, to celebrate the completion of STAT 545 and/or to prepare for the glorious future of STAT 547.

Remember the sampler concept. Your homework should serve as your own personal cheatsheet in the future for canonical tasks. Make things nice – your future self will thank you!

Singer data ... or whatever

You can work with Joey's `singer` data or take this chance to play with something else. In which case, you'll have to create comparable tasks for yourself.

Factor management

Drop the 0. Transform the `year` column into a factor. Then, filter the Singer data to remove observations associated with the `year` 0. Additionally, remove the unused factor level associated to 0 and / or relevel everything so to have a level for each year. Provide concrete information on the data before and after removing these rows and the 0 `year`; address the number of rows and the levels of the affected factors.

Reorder the levels of `title` or `artist_name`. Use the `forcats` package to transform the column into factors and change the order of the factor levels, based on a principled summary of one of the quantitative variables. Consider experimenting with a summary statistic beyond the most basic choice of the median. While you're here, practice writing to file and reading back in (see next section).

Characterize the (derived) data before and after your factor re-leveling.

- Explore the effects of `arrange()`. Does merely arranging the data have any effect on, say, a figure?
- Explore the effects of reordering a factor and factor reordering coupled with `arrange()`. Especially, what effect does this have on a figure?

These explorations should involve the data, the factor levels, and some figures.

File I/O

Experiment with one or more of `write_csv()/read_csv()` (and/or TSV friends), `saveRDS()/readRDS()`, `dput()/dget()`. Create something new, probably by filtering or grouped-summarization of Singer. I highly recommend you fiddle with the factor levels, i.e. make them non-alphabetical (see previous section). Explore whether this survives the round trip of writing to file then reading back in.

Visualization design

Remake at least one figure, in light of something you learned in the recent class meetings about visualization design and color. Maybe juxtapose before and after and reflect on the differences. Use the country or continent color scheme that ships with Gapminder. Consult the guest lecture from Tamara Munzner and everything here.

Writing figures to file

Use `ggsave()` to explicitly write a figure to file. Then use `![Alt text](/path/to/img.png)` to embed it in your report. Things to play around with:

- Arguments of `ggsave()`, such as width, height, resolution or text scaling.
- Various graphics devices, e.g. a vector vs. raster format.
- Explicit provision of the plot object `p` via `ggsave(..., plot = p)`. Show a situation in which this actually matters.

Clean up your repo!

You have 6 weeks of R Markdown and GitHub experience now. You've reviewed 4 peer assignments. Surely there are aspects of your current repo organization that could be better. Deal with that. Ideas:

- A nice Table of Contents in top-level README that links to individual pieces of work.
  - Good for future: `hw03 dplyr verbs`
  - Bummer in the future: `hw03`
  - Include a slug with content info!
- Remove all downstream stuff, e.g. figures, html, etc. and re-render everything. It will be nice to not have weird, vestigial files lying around to puzzle you in future.
- Anything that's `Rmd` but that could be `md`? Convert it.

But I want to do more!

Make a deeper exploration of the forcats packages, i.e. try more of the factor level reordering functions.

Revalue a factor

- Pick a handful of countries, each of which you can associate with a stereotypical food (or any other non-controversial thing … sport? hobby? type of music, art or dance? animal? landscape feature?). Create an excerpt of the Gapminder data, filtered to just these countries. Create a new factor – you pick the name! – by mapping the existing country factor levels to the new levels.
  - Examples: Italy –> wine, Germany –> beer, Japan –> sake. (Austria, Germany) –> German, (Mexico, Spain) –> Spanish, (Portugal, Brazil) –> Portuguese. Let your creativity flourish.

Report your process

You're encouraged to reflect on what was hard/easy, problems you solved, helpful tutorials you read, etc. Give credit to your sources, whether it's a blog post, a fellow student, an online tutorial, etc.

Rubric

Check minus: One or more elements are missing or sketchy. Missed opportunities to complement code and numbers with a figure and interpretation. Technical problem that is relatively easy to fix. It's hard to find the report in this crazy repo.

Check: Hits all the elements. No obvious mistakes. Pleasant to read. No heroic detective work required. Well done! *This should be the most typical mark!*

Check plus: Exceeded the requirements in number of dimensions. Developed novel tasks that were indeed interesting and "worked". Impressive use of R – maybe involving functions, packages or workflows that weren't given in class materials. Impeccable organization of repo and report. You learned something new from reviewing their work and you're eager to incorporate it into your work.