Final Portfolio

Due: May 11, 2018 @ 11:59pm

Overview

In lieu of a written final exam, you will construct a personal course portfolio containing three elements:

1. An annotated list of R functions
2. Reflection and further learning response
3. Comparative discussion of related simulations

You will be provided with a starter repository on Github that must be used for putting together your individual portfolio, following the instructions provided below. The portfolio must be submitted by the end of the day on **May 11, 2018** by issuing a *Pull Request* in the usual way.

In addition to the portfolio, students will also meet with the course instructor for a 5–10 minute final interview on **May 15, 2018** during the scheduled final exam period. If you have an unavoidable conflict with this time slot, you must inform the instructor and schedule another time during the final exams period that you will do the interview.

Grading

The assessment of your portfolio and final interview will be combined into a single grade, with the portfolio worth 70% and the final interview worth 30%. As per the syllabus, the combined grade is worth 25% of your overall course grade.

Portfolio guidelines

Consider the following guidelines as you get started with putting together your portfolio, **but please note that this list is not exhaustive**. It is important that you fulfill the basic requirements for each element, but organization, thoroughness, and creativity are encouraged. Grades will reflect the overall quality of the portfolio.

Annotated list of R functions

Purpose — A significant portion of the course is dedicated to learning how to use R and the `tidyverse` ecosystem to wrangle and explore data, and then analyze it using statistics. Remembering the different functions, how they are used, and what problems they solve requires consistent practice and review. This is why it's important to have personal notes that explain how a function works in your own words, which can be referenced long after you've completed this course.

For your portfolio — Put together an annotated list of R functions for your portfolio by adding information to the template file in your starter repository. The template file comes with a pre-filled example for the `ggplot2` syntax and how to use `geom_histogram()`, as well as listing other sections that are expected to be part of your notes. Use these examples as inspiration for how to put together notes on the other functions in R.

Unless told otherwise, when writing notes for a function, please include the following:

- The function's name
- The important inputs
  - If the input was used in class or for a homework assignment, then it's important
  - For `ggplot2` functions, include a separate subsection for important aesthetic mappings (see example in your template file)

- A summary — 1 or 2 sentences — of what the function does
- An example showing how to use the function
  - **Your example cannot use the same dataset as the examples found in the help documentation or from class.** For example, when explaining the `dplyr` function you should not use the `presidential` dataset, but you *can* use a dataset from one of the homework assignments.

Your notes should discuss the following packages and functions:

- `base` R
  - `c()` and `nrow()`
- The `ggplot2` package
  - `geom_histogram()`, `geom_point()`, `geom_bar()`, `geom_col()`, `geom_qq()`, `geom_smooth()`, `facet_wrap()`, `facet_grid()`
  - Basic instructions on how to change the axes labels and give the plot a title
- The `readr` package
  - `read_csv()`
- The `dplyr` package
  - `select()`, `slice()`, `rename()`, `arrange()`, `filter()`, `mutate()`, `group_by()`, `summarize()`, `count()`
- The `tibble` package
  - Instructions on manually creating a `tibble` using `data_frame()`
- The `tidyr` package
  - `gather()`, `separate()`
- The `rvest` package
  - Instead of notes for each function, describe the procedure for using the SelectorGadget extension with `read_html()`, `html_nodes()`, and `html_text()` to perform basic webscraping tasks
- Statistics functions
  - Summary statistics functions: `mean()`, `median()`, `sd()`, `IQR()`, `min()`, and `max()`
  - Percentiles: `quantile()`
  - Instructions on extracting the **values** of the cumulative distribution function using `ggplot_build()` and `stat_ecdf()` from the `ggplot2` package
  - Linear modeling: `lm()`
- The `infer` package
  - Describe the inputs for the four functions: `specify()`, `hypothesize()`, `generate()`, and `calculate()`
  - Instructions **with an example** on how you use `infer` to conduct a hypothesis test: how you simulate the null distribution, and how you calculate the p-value
  - Instructions **with an example** on how you use `infer` to find a confidence interval using bootstrapping, including how you would find different size intervals, for example a 90% interval or a 95% interval
- The `modelr` package

  - Instructions on how to use `data_grid()`, `add_residuals()`, `add_predictions()`, `seq_range()`, and `geom_ref_line()` to obtain the predictions and residuals of a model, as well as create visualizations that allow you to inspect the quality of the model

You are welcome to include notes on additional functions not listed here. Remember, these notes are ultimately for you, so think about what you would find helpful when you refer back to these.

Reflection and further learning response

Purpose — You've practiced and reviewed a lot of different skills this semester during in-class demos, group work, posts about the readings, homework assignments, and the midterm project. You've learned a lot! As with any discipline, there's many more topics to learn about in the data science field that we did not have enough time to cover here, and there is far more depth and complexity to the various topics we did cover. Imagine for a moment that an official sequel to this course (let's call it CDS-103) was being developed, and you have been asked to provide input that will help in designing the topic list and schedule. Reflect on what you've learned in this course, the skills you've learned that you've found to be the most valuable, which data science topics we discussed that you now want to learn even more about, and the topics you wish we covered in CDS-101 but did not. For this part of the portfolio, you will turn these thoughts into two concrete suggestions:

1. A topic that we introduced in CDS-101 that deserves more in-depth discussion and practice exercises
2. A data science topic that we did not cover in CDS-101 that should be covered in a follow-up course

For your portfolio — Your response should be at least two paragraphs in length — one paragraph per suggestion minimum — with at least 3 sentences per paragraph. While there is an element of opinion here, your suggestions must be **substantive** and explained/supported by referring to content from the course and additional information you look up on your own. If you refer to content from this course, mention the assignment, reading, or class we covered it in (see the calendar on http://spring18.cds101.com). If you refer to ideas you found elsewhere, provide a citation.

Consider the following guidelines and suggestions when completing your write-up:

- Keep in mind that the only prerequisite for a hypothetical CDS-103 class would be completing CDS-101, so while it would be a lot of fun to have in-depth coursework on neural networks, the math and programming skills necessary to cover this kind of topic properly is too high for a 100-level class
- When suggesting a topic for more in-depth coverage, explain your reasoning for why it should receive more attention. What aspects of the topic should be focused on, and how will this improve your understanding of the topic? Be precise!
- When suggesting a new topic that should be covered, provide a reference to where you heard about it. Also explain why this is a good topic to cover, and why it should be considered as a top choice over other possible topics. In your justification, you may want to explain how would learning about this topic be useful to you. For example, if you are not a CDS major, does it intersect with your chosen major, and if so, how? If you are a CDS major, will it better prepare you for a later class and is this topic covered in any of the other classes provided by the CDS department?
- While programming is an important aspect of data science, it is ultimately just a tool and is not an end unto itself. This means that suggestions should **not** be of the form "learn how to X using R". Suggestions that primarily focus on a programming topic need to be justified by citing data science methods that require them.
- Do not suggest that we cover the mechanism-driven simulations (this includes agent-based simulations) described in the next section, **Comparative discussion of simulations**. As discussed, these are approaches that generate their own data and predictions without the use of datasets, and so this is generally beyond the scope of a follow-up CDS-103 course. Additionally, the CDS department has several courses that touch on the methods and concepts behind these types of simulations, including CDS-201, CDS-205, CDS-230, and CDS-411.

- If you do not know anything about the topic you are suggesting, you are expected to take the time to look up some basic information so that you can describe it competently. This is not a blog post or journal entry, so you will have to look up information in order to produce a write-up that is eligible for full credit.

Comparative discussion of simulations

Purpose — The field of computational and data sciences extends beyond the topics focused on in this course. From the CDS-101 perspective, we've used the terms **model** and **simulation** as shorthand for *data-driven* models and simulations. Yet, there is an alternative approach to model and simulation building that works in the opposite direction and is an indispensable tool in the natural sciences, engineering, and computational social sciences. This class of models and simulations *generate* predictions and data without using an underlying dataset as input. To distinguish these from their *data-driven* counterparts, we will refer to them as follows:

- A *microscopic* or *mechanism-driven* model or simulation is based on the known laws of nature. An example is deriving equations of motion for the planets in our solar system using Newton's law of universal gravitation.

After building this kind of model or simulation, the researcher will scan the model's parameter space and look for trends in the predictions and outputs, which are then compared against experimental data (if available). If the model or simulation generates predictions or outputs that accord with the experimental data, then the proposed mechanism can be regarded as a plausible explanation for observed trends. However, if the predictions or outputs fail to agree with the experimental data, then the model or simulation is falsified and the proposed mechanism is rejected.

For your portfolio — You will visit the following two webpages containing short summaries of two simulations that share a common lineage:

- Ising Model (https://jkglasbrenner.github.io/ising-model-js/)
- Schelling's Model of Segregation (https://jkglasbrenner.github.io/schelling-model-js/)

Each page also contains the simulation itself implemented in Javascript, which runs inside the web browser. The simulations are visual and interactive, allowing you to change a small set of parameters using a simple dashboard. After becoming familiar with the different simulations and developing a basic intuition for how each one behaves, you will then compare and contrast them in a short write-up. Your comparative discussion must be at least 2 paragraphs in length (a minimum of one paragraph per simulation) and include the following:

- At least three ways in which the simulations are similar to one another
- For each simulation, at least one way it is *different* from the other one
- Pick one of the simulations and suggest a feature or rule that you could add to it that would change its outputs and predictions. You only need to do this using plain language, you are not expected to write any code for this. Be sure to hypothesize what you think the changes will do and what they might mean. For example, how do you anticipate that your proposed change will simulate different physical mechanisms or human behavior?