Jackson Rudoff

Modern Data Structures

November 25th, 2022

**Final Project Proposal**

Political science – as with other social sciences – have begun migrating to Python for conducting statistical research. This tracks with the parallel developments in packages like Pandas, which has made the language more hospital to data analysis and hypothesis-based research. However, many datasets which have been adapted to R-specific data-forms or packages have not seen a similar conversion to Python. Datasets like the GSS, for example, are accessible as easily ingestible .rds files, which can be quickly and efficiently converted to tibbles with correctly formatted variables. My proposed final project would be to make an oft-used dataset – the [American National Election Studies (ANES)](https://electionstudies.org/data-center/) – into a package format, which would provide users with helper functions that will streamline analysis of the data. It would include graphical and statistical methods that will make public opinion research more user-friendly, particularly for those just getting familiar with using Python in social science research.

ANES data is, to put it simply, messy. It is the most widely used public opinion dataset, collected every two years during presidential and midterm election cycles by a rotating team of political scientists and survey methodologists. The nature of public opinion data, however, means that there is little natural synergy between the variables. Questions offered to respondents s will range from demographic information to self-assessments on a Likert scale to numeric “feelings thermometers” of the respondent’s opinions on a given political issue. The sheer number of variables also means you must work with an extensive and at times confusing codebook. This makes the pre-processing stage overwhelming and strenuous, especially for those with limited coding experience. But the consistent structure of the ANES dataset means that there is some consistency in the sorts of transformations you would perform for each time series.

The main advantage of the package is that it bypasses the step of importing the data, immediately providing the user with the ANES samples in the form of class objects. Though the package by default will provide the last 4 editions of the time series. However, users could also instantiate new ANES objects by providing new Excel sheets taken from the ANES website. These class objects would have several built-in methods and functions, the exact inventory of which I am still working out. At present, these are the built-ins I intend to include [\*\* = I am doubting if I can complete it on time]:

1. .clean\_junk(ANES\_object)
   1. Performs the common cleaning operations that ANES objects require, such as filtering out variables that are solely clerical or for the interviewer.
2. .generate\_sample(n\_respondents = 1000, preview\_length = 10, variables)
   1. Method that slices a random sample of *n* respondents (defaults to 1000). *Preview\_length* indicates how much of the resulting DataFrame to print to console. *Variables* takes a list of column names to slice from the entire dataset.
3. .what\_is\_this(variable)
   1. Outputs the type, mean, sd, description from the codebook, and question asked to elicit a response for a given variable.
4. .time\_check(years, variables)
   1. Checks if the given variables are in the given years in case user wants to merge different samples along the time series.
5. .save\_ANES\_data(df\_in, out\_path, format = ‘.csv’)
   1. Allows you to save the objects you create so that they can be shared or used in another environment. Defaults to .csv, can also work with .xlsx. Leverages the Pandas functions but condenses the process to be more like R.
6. thermometer\_plot(x, y)
   1. Seaborn-based plotting method that allows for easy plotting of the feelings thermometer variables that make up a significant portion of the ANES data.
7. .model\_vars(df\_in, response, explanatory, type = ‘ols’)\*\*
   1. Helper function that runs a simple linear model and outputs test statistics based on the given response and explanatory variables in the DataFrame. Defaults to OLS, can work with logistic regression as well.
8. .find\_problematic\_vars(ANES\_object, issue)\*\*
   1. Returns a DataFrame ordering the variables by a selected potential problematic quality from the list [%NaN, skew, max/min z-score].
9. .find\_possible\_predictors(response, n\_respondents = 1000)\*\*
   1. Iterates through the data to find which variables significantly predict a linear effect for a given response variable. Intended to help find possible avenues of inquiry, not conclusive results.

I subtract from this list given time constraints, but I believe these constitute the most useful functions for a typical analyst of ANES data. The most important contribution of this project, in my opinion, would be that it transforms recent ANES datasets to an easily manipulable format in Python, improving its accessibility for users of varying familiarity with Python and ANES data.

My package will have a few core dependencies. These include Pandas, NumPy, Seaborn, SciPy, sklearn, and statistics. Pandas is particularly important to the functioning of this package, as it is essential to transforming the Excel/csv based ANES datasets. The remaining packages will assist with the various helper functions/methods that I detail below. Seaborn, for example, is what my proposed graphical method would be based on, as I prefer it for social science-oriented plotting over Matplotlib.