eduViz Data-map

CIM 545
Nicolas Aguirre

Table of Contents

Overview

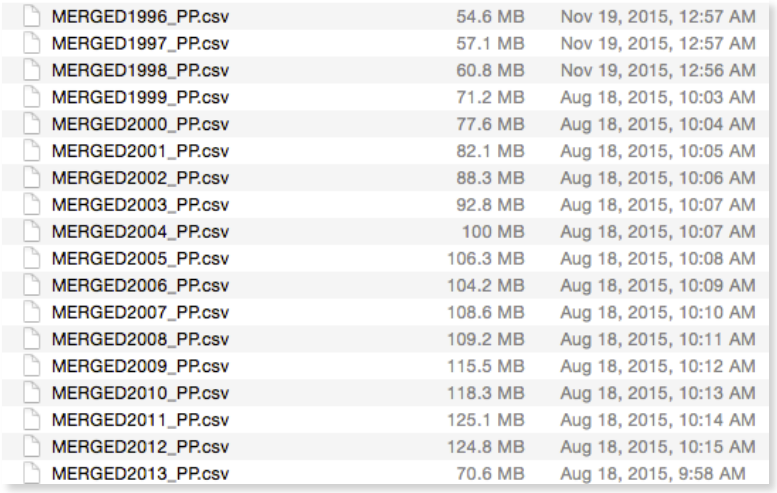
This document will serve to inform the design of the single-page web app, EduViz. Given the criticality and immenseness of the dataset, this project merits a document to serve as a guide. While the app's interface is integral to satisfying usability, the single-page application is ultimately a data-driven endeavor. This is scarcely a project of UI design; the story revealed from the data depends squarely on an effective analysis. This document will also serve as a dataset codebook, functional development log, as well as an effective data analysis tutorial.

In the same way that a site map aims to show the extent of the application in logical sequence; this data map strives to construct a reproducible procedure for manipulating, cleaning, and analyzing data in the interest of a meaningful and informative narrative. In accordance with *A Project Guide to UX Design* (Ungers and Chandler), hierarchy and dominance obligate the project to focus on a few, focal elements of the interface and eschew extraneous or minute functionality. Analogously, the hierarchy and dominance of the dataset will facilitate a meaningful user interaction with the visualization and narrative.

Given that the project is overwhelmingly back-end in nature, design is more relevant to data and related computational procedures, i.e., creating reliable scripting to parse the dataset (especially in the interest that others may utilize the data for other projects). In early stages of data exploration, it became evident that this project is hardly about visualization, and more about engineering a wieldy dataset for visualization.

Dataset Profile

The dataset is considered "non-trivial[1]", about 1.67 gigabytes in total, comprised of 18 individual (yearly) datasets. Each year of data contains in excess of 5000 rows, and about 1700 columns.



MERGED1996_PP.csv	54.6 MB	Nov 19, 2015, 12:57 AM
MERGED1997_PP.csv	57.1 MB	Nov 19, 2015, 12:57 AM
MERGED1998_PP.csv	60.8 MB	Nov 19, 2015, 12:56 AM
MERGED1999_PP.csv	71.2 MB	Aug 18, 2015, 10:03 AM
MERGED2000_PP.csv	77.6 MB	Aug 18, 2015, 10:04 AM
MERGED2001_PP.csv	82.1 MB	Aug 18, 2015, 10:05 AM
MERGED2002_PP.csv	88.3 MB	Aug 18, 2015, 10:06 AM
MERGED2003_PP.csv	92.8 MB	Aug 18, 2015, 10:07 AM
MERGED2004_PP.csv	100 MB	Aug 18, 2015, 10:07 AM
MERGED2005_PP.csv	106.3 MB	Aug 18, 2015, 10:08 AM
MERGED2006_PP.csv	104.2 MB	Aug 18, 2015, 10:09 AM
MERGED2007_PP.csv	108.6 MB	Aug 18, 2015, 10:10 AM
MERGED2008_PP.csv	109.2 MB	Aug 18, 2015, 10:11 AM
MERGED2009_PP.csv	115.5 MB	Aug 18, 2015, 10:12 AM
MERGED2010_PP.csv	118.3 MB	Aug 18, 2015, 10:13 AM
MERGED2011_PP.csv	125.1 MB	Aug 18, 2015, 10:14 AM
MERGED2012_PP.csv	124.8 MB	Aug 18, 2015, 10:15 AM
MERGED2013_PP.csv	70.6 MB	Aug 18, 2015, 9:58 AM

Important to note is that the number of columns precludes the use of certain services such as *MySQL* to construct a database. Data parsing will likely be done with the use of scripting as opposed to a query-based approach.

[1] In mathematics, the adjective *trivial* is frequently used for objects (for example, groups or topological spaces) that have a very simple structure… The antonym *nontrivial* is commonly used by engineers and mathematicians to indicate a statement or theorem that is not obvious or easy to prove. (Wikipedia – Triviality (Mathematics))

Tools

- Python
    - Python is a high-level programming language with extensive documentation and libraries, many of which assist in data manipulation. Python has become a favorite of data scientists for parsing very large datasets.

- Jupyter
    - Jupyter is a web application that allows programmers to create and share documents containing live code, equations, visualizations and explanatory text. It proves invaluable in multi-phasic statistical modeling and data cleaning. Because it supports multiple programming languages, especially Python and terminal-based commands, it doubles as an effective documentation tool. In the interest of having a reproducible procedure for data parsing, Jupyter will provide a step-by-step approach to the data analysis for anyone interested in following along. Tools such as this one greatly improve the transparency of the project.

- RStudio
    - While the majority if scripting will take place in Python, R has an exhaustive set of tools for data exploration and visualization. Some tools in particular allow for the dynamic generation of HTML pages containing visualizations; this may prove useful creating explanatory visualizations.

- Conda environment
    - Conda is a package manager application that allows for an isolated python-based environment. This, in turn, allows the programmer to ensure that a stringent development environment exists for execution of Python scripting, which will be critical in data analysis.

- Bash scripting
    - Given that the dataset consists of multiple files, command-line scripting *a la* bash will give small scripts that can be used to organize and manipulate the dataset. The challenge comes from having a multitude of data files, rather than one, and shell scripting will prove invaluable in generating reproducible code

- csvkit
    - Command line tool for data manipulation. Csvkit allows for easy file type conversion (e.g. Excel to CSV), reorder columns of data, exploring subsets of the data and more.

- Tableau
    - Tableau is a data-exploration and data visualization suite. With data that has been properly prepared, Tableau allows for a quick, virtually programming-free environment for visualization. This will prove useful in the exploratory phase of data analysis.

- D3.js
    - D3 has become the golden standard for web visualization- it integrates nicely with other JavaScript libraries and provides flexible, informative, and interactive visualizations. A concerning limitation of D3 is that the dataset must already be in a maneuverable state (this is where the other tools will prove essential)