MEGA MILLIONS DATA ANALYSIS

Mis 5400 Summer 2019

Dane Mariott & Shelby Fullbright

Documentation/Presentation

**INTRODUCTION:**

We chose Mega Millions lottery as a bit of an abstraction because it's basically a generic, medium-sized, information-rich dataset that could represent any number of business applications. The Mega Millions page of the site contains a dynamically generated data table in a scroll window that updates bi-weekly, an .xlsx direct download of the table data, and no API. Our intention is to explore variables that might marginally increase the probability of winning.

**DATA ACQUISITION:**

We provided the code to import the .xlsx but intentionally scraped the table with pandas to exhibit the ability to do so and familiarize ourselves with the pandas library. Once scraped, we also wrote the data to a .csv in the event that it might become necessary (like we if failed to persist our data).

**DATA PERSISTENCE:**

At this point, we created our Microsoft Azure SQL database, cataloged our credentials to view the DB in our respective SQL Server Management programs, and began writing code to persist our data. We have chosen an SQL database hosted by Azure because these appear to be both strong and common industry standards for data management. While we will only be working with one data table at a time, we will be using a traditional SQL relational database table schema for its ease of access, spreadsheet structure, and because it already mirrors the shape of our data. We also immediately realized that the example code for the connection string was not working for us and spent days googling alternative options. This revelation led to the addition of SQLalchemy, urllib, and a system-specific PYODBC "wheel" file installation to create our connection string.

**DATA EXPOSURE:**

*Part 1.*

Again, because our connection string was so specific, we needed to create entirely different code to interact with our API; most notably the way pandas processes SQL. As we populated the API, we considered cleaning the data first but opted to leave the entirety of our dataset because we wanted a large variety of options to assess value for visualizations.

*Part 2.*

Here, we created our Jupyter Notebook file and imported the data from our API to a pandas dataframe for analysis and value addition. We verified the data by number and visual inspection. We then munged the 'Draw date' column down to a single 'month' variable to cross-examine with 'Jackpot Winners' which needed to be reduced to 'wins/losses' in boolean "true/false" fashion, in order to show which months in the 9 years of drawings had the highest wins and highest losses over that span. This would allow a player to avoid high-loss months and indulge during high-win months. Next, we wanted to show the most frequently drawn Megaball numbers as winning just the Megaball number alone allows a player to break even on their bet. Pandas displays columns as "series" and generally casts them as objects or floats, which meant our Megaball "number" displayed as "NaN" and had to be cast as "int" before it could be displayed for analysis. As seen, we have provided bar chart plots for both aspect examinations in our analysis.

**DOCUMENTATION/PRESENTATION:** ^^^^^^^^^^^^^^^^^ \*

**\*NOTE:** *More in-depth documentation of our step by step process is included in the code of the project files themselves.*