Sabneet Bains

Data Visualization – EN.605.662

Project #4: Interactive Visualization using Python

Introduction

Project #4 is an exploratory assignment to acquaint Python's open-source data visualization libraries by developing three discrete sample visualizations with inherent interactivity to enable data exploration and advanced visualization techniques, either unavailable or hard-to-attain within off-the-shelf applications.

Thus, three discrete datasets with at least 3 variables and 100 rows were chosen from the online data science community Kaggle and individually visualized through Python as "Sample01", "Sample02", and "Sample03." Different visualization approaches were taken to each dataset's nature, but the overall libraries were kept relatively consistent.

In the end, three Python scripts were produced to analyze each dataset and create interactive visualization dashboards.

Dataset

Keeping with the requirements for the datasets as outlined in the Introduction, the first dataset representing "Sample01" can be found here Electric & Alternative Fuel Vehicles US [2022] | Kaggle. In essence, the dataset lists specifications of all Electric Vehicles (EVs) and Alternative Fuel Vehicles (AFVs) available in the US as of July 2022.

However, for this project, only the All-Electric variants were analyzed, and their individual variable descriptions are as follows,

Variable	*Data Type*	*Description*
Category	Nominal	Vehicle type: Sedan/Wagon, SUV, Pickup
Model	Nominal	Vehicle model/name by the manufacturer
Model Year	Interval	Production year of the model
Manufacturer	Nominal	Vehicle manufacturer

The Second dataset for “Sample02” is available here Most Subscribed YouTube Channels | Kaggle. The dataset lists 7 attributes about the top YouTube channels accrording to subscriber share, and the individual variable descriptions for these attributes are as follows,

Variable	*Data Type*	*Description*
Rank	Ordinal	The rank of the channel by subscribers
YouTuber	Nominal	Channel official name

Subscribers	Ratio	# of subscribers
Video Views	Ratio	# of total video views
Video Count	Ratio	# of video uploaded by channel
Category	Nominal	Genre/category of channel
Started	Interval	Origin year of channel

Likewise, the third dataset for "Sample03" is found here Student's Scores | Kaggle. This dataset includes student subject scores in different data science skills and then links them to the job recruitment status of each student; the individual variable descriptions are as follows,

Variable	***Data Type***	***Description***
Python	Ratio	Student's subject score in Python
SQL	Ratio	Student's subject score in SQL
ML	Ratio	Student's subject score in ML

Tableau	Ratio	Student’s subject score in Tableau
Excel	Ratio	Student’s subject score in Excel
Student Placed	Boolean	The recruitment status of a student

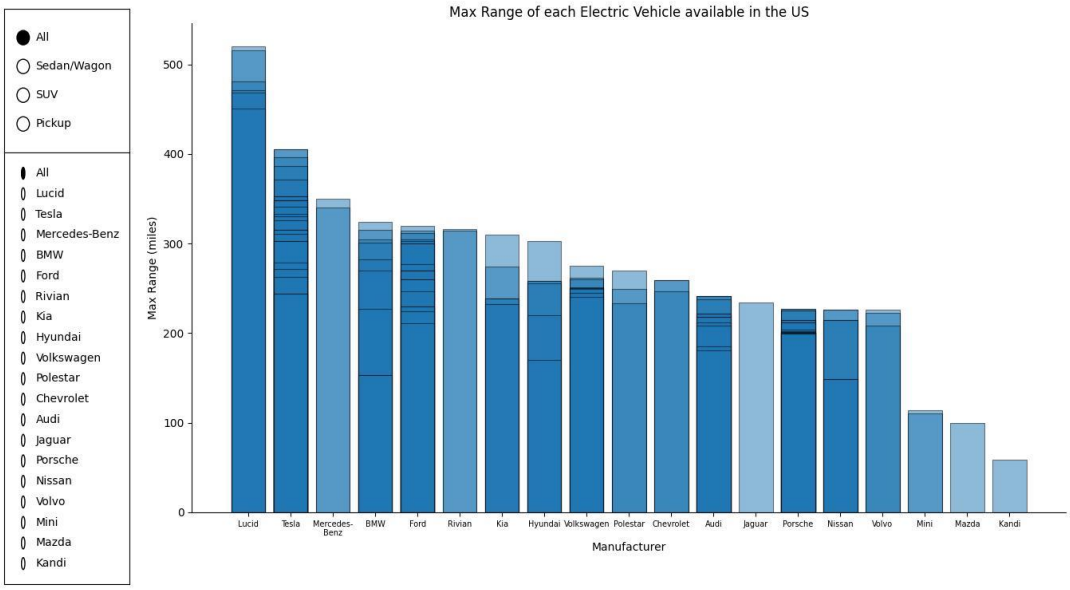
Approach

To create individual visualizations for each of the datasets, I chose the following libraries,

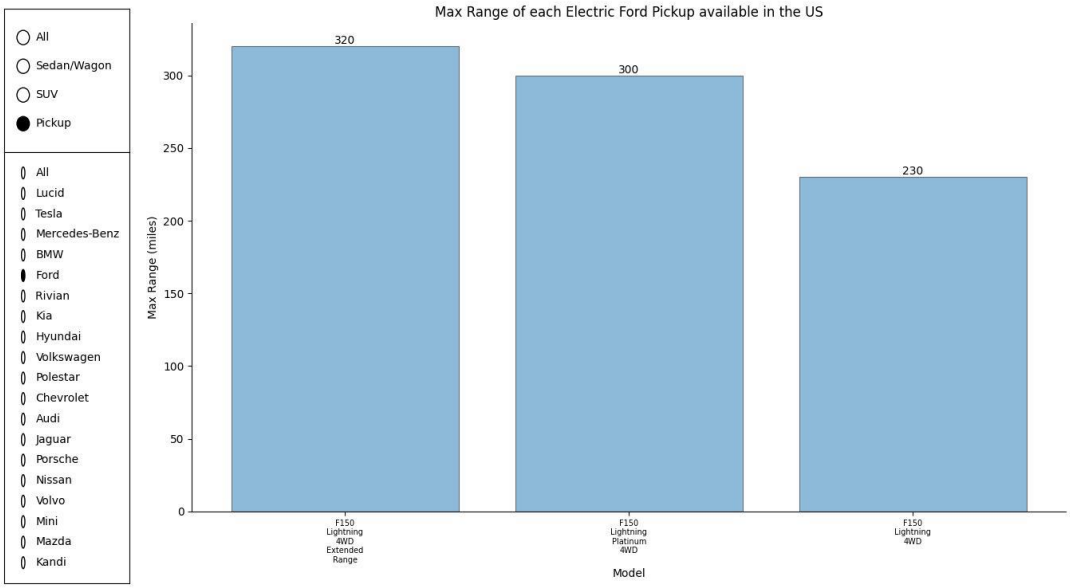
Library	***Purpose***
Matplotlib	Plotting library with object-oriented APIs for embedding plots into general-purpose graphical applications
NumPy	Adds support for large multi-dimensional arrays, matrices and other high-level mathematical operations
Pandas	For data manipulation and analysis of data structures and operations for manipulating numerical tables and time series
Textwrap	For wrapping and formatting plain text and long sentences

Visualization #1

For “Sample01,” I chose to visualize the data through a bar chart with the default view representing all the vehicle categories and manufacturers in a single plot. This view is excellent for condensing a lot of data, and different bar transparencies help distinguish separate models under each manufacturer.



However, this is a double-edged sword as the model names can be pretty extensive, and labeling each model/bar height is cumbersome; hence a simple approach was to create a dual filter system where the user can filter by both the category and manufacturer at the same time, which means one can end up with a plot like this,

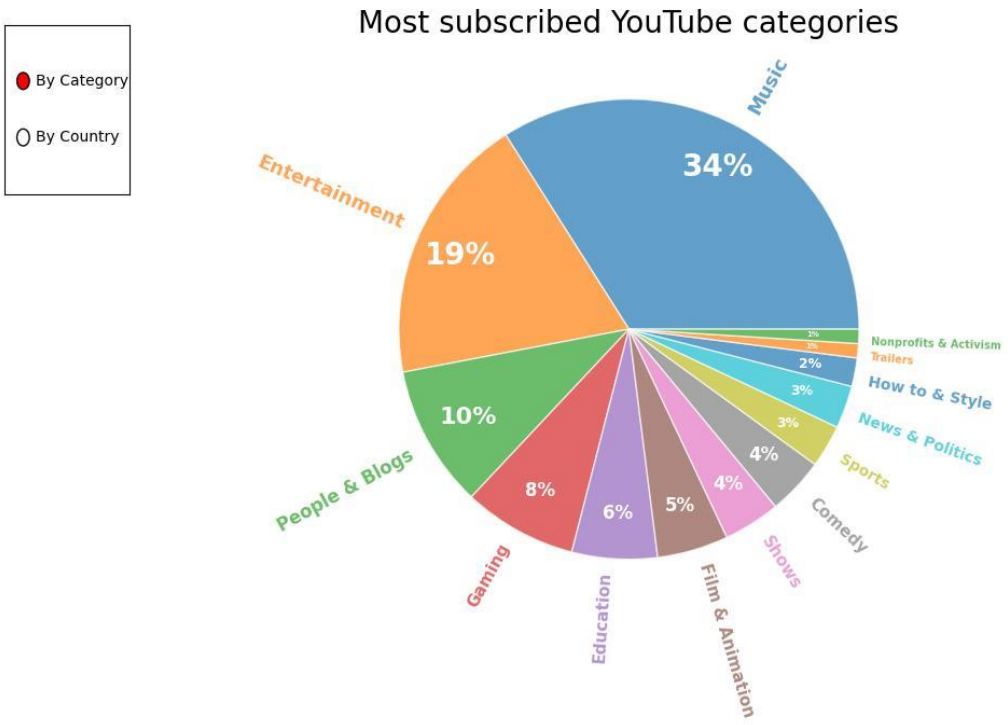


Here, while keeping data exploration thriving, the user can drill down on only pickup trucks while filtering by Ford as the sole manufacturer. The resulting plot, therefore, shows all the models by Ford that are all electric pickup trucks and their respective max ranges in miles.

Visualization #2

For “Sample02,” I chose to visualize the data through a pie chart with the default view representing the most subscribed YouTube categories worldwide. The visualization is laid out as a pseudo extension of radial plots with category labels radiating outwards at different font sizes respective to their share of the pie.

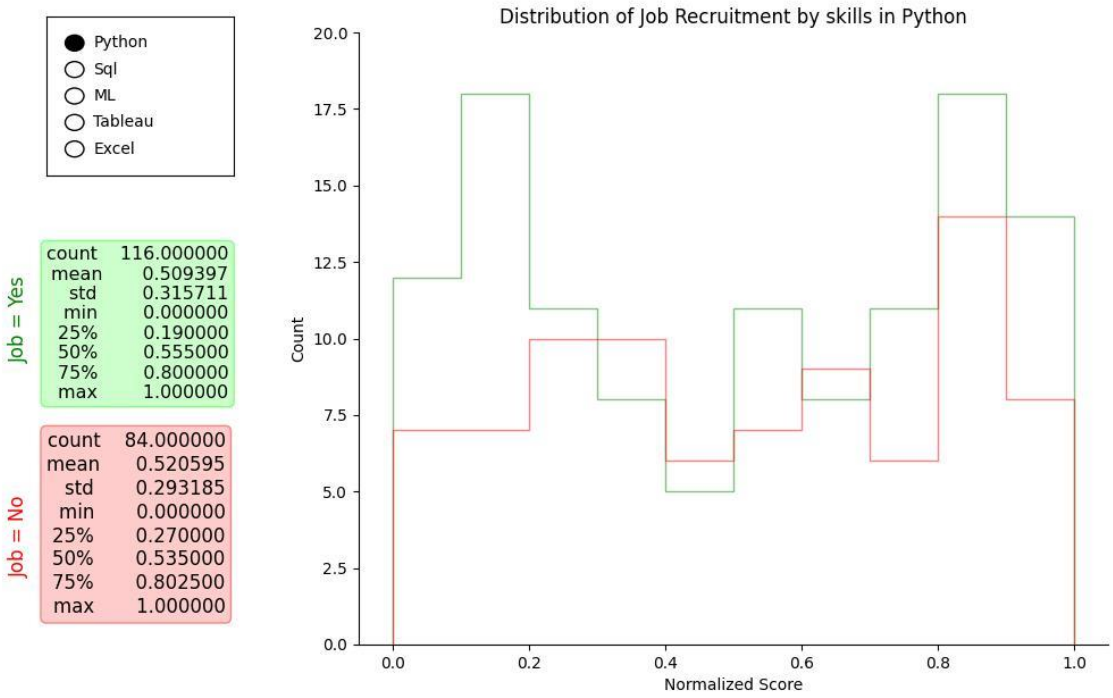
Likewise, the filter is quite simple in this case, as one can switch between the most subscribed categories or the countries with the most subscribing individuals.



Visualization #3

For “Sample03,” I chose to visualize the data through a histogram with the default view showing the distribution of job recruitment by skills in Python while also presenting the descriptive statistics of each recruitment boolean (yes/no). The visualization features unfilled bins to correctly see the distribution of “yes” and “no” over the ranges of the normalized test scores.

Likewise, the filters are pretty simple, as one can switch between any subject and drill down on the distribution of recruitment status and the corresponding statistics.



Conclusion

Throughout the three visualizations in this project, I could leverage the Python libraries as described in the Approach (see the source code for more information).

These visualizations were distinct and had inherent interactivity to enable data exploration and advanced visualization techniques, either unavailable or hard-to-attain within off-the-shelf applications.

The source code provided as three separate Python scripts is constructed in a pseudo-object-oriented fashion as it is modular and easily modifiable according to future needs. Hence, the main objective of exploration was accomplished while utilizing visualization skills learned from previous projects.

References

docs.python.org. (n.d.). *textwrap — Text wrapping and filling — Python 3.10.0 documentation*. [online] Available at: https://docs.python.org/3/library/textwrap.html.

matplotlib.org. (n.d.). *Gallery — Matplotlib 3.4.2 documentation*. [online] Available at: https://matplotlib.org/stable/gallery/index.html.

numpy.org. (n.d.). *NumPy tutorials — NumPy Tutorials*. [online] Available at: https://numpy.org/numpy-tutorials/ [Accessed 1 Aug. 2022].

pandas.pydata.org. (n.d.). *Getting started — pandas 1.0.1 documentation*. [online] Available at: https://pandas.pydata.org/docs/getting_started/index.html.

www.kaggle.com. (n.d.). *Electric & Alternative Fuel Vehicles US [2022]*. [online] Available at: https://www.kaggle.com/datasets/saketpradhan/alternative-fuel-vehicles-in-the-us [Accessed 1 Aug. 2022].

www.kaggle.com. (n.d.). *Most Subscribed YouTube Channels*. [online] Available at: https://www.kaggle.com/datasets/surajjha101/top-youtube-channels-data [Accessed 1 Aug. 2022].

www.kaggle.com. (n.d.). *Student's Scores*. [online] Available at: https://www.kaggle.com/datasets/samarsaeedkhan/scores.