For week 35:

This dashboard groups all data by gender. Top row left has a world choropleth map with selected country highlighted. Top row right is a histogram of user counts, binned by AGE\_RANGE, EDUCATION, or YEARS\_EXPERIENCE. I experimented with Plotly Studio for ideas, but have developed this mostly from scratch.

Bottom row left is a pareto chart showing the top 10 of CS\_LANG, AI\_ASST, or AI\_FEATURE. I made a design decision to place the highest numbered category on the bottom instead of the top to steer clear of the legend. The bottom row right is a barbell chart made with px.line, for direct companion of Male vs Famale responses.

Dataset has 608 columns, but many are part of a multi-column ‘dummies’ format groups, which uses separate columns for each unique answer to the same question. The dummies format is also considered to be a sparse dataframe, and is why the dataset is so wide

Polars concat\_list was used to gather the CS\_LANG, AI\_ASST, and AI\_FEATURE values into 3 columns with datatype of list. Polars support for list and array types is far more extensive and efficient than the pandas equivalent. I use the number of answers from each user by category to weight the values. For example, when users list 4 programming languages, each one is given a weight of ¼. An index is added to give each user a unique ID, before exploding the dataframe to separate the list values into separate rows, similar to a melt or unpivot.

The cleaned dataset is saved as a parquet file, with all datatypes optimized for storage size. String categories are cast as Categorical, float columns are cast as Float32, Integer columns cast as UInt8. The app checks for the parquet file, and it is used as the global dataframe. If not found the app will reads the csv file, cleans it, and saves the data frame as parquet for future runs. This allows me to upload a much smaller parquet file to Plotly Cloud and avoids having to read the csv file and clean it every time.

I hope you enjoy this dashboard and appreciate any feedback or suggestions.

Here is a link to Plotly Cloud hosted dashboard:

Here are a few screenshots:

Here is the code:

.