‹ Return to Classroom

Investigate a Dataset

REVIEW

HISTORY

Meets Specifications

Very impressive submission! I can see your hard work reflected in your project 🏆 Congratulations on achieving this and good luck on your way to master data analysis 😃

Code Functionality

✓

All code is functional and produces no errors when run. The code given is sufficient to reproduce the results described.

✓

The project uses NumPy arrays and Pandas Series and DataFrames where appropriate rather than Python lists and dictionaries. Where possible, vectorized operations and built-in functions are used instead of loops.

✓

The code makes use of functions to avoid repetitive code. The code contains good comments and variable names, making it easy to read.

Excellent job! solid code and well documented 😃

Quality of Analysis

✓

The project clearly states one or more questions, then addresses those questions in the rest of the analysis.

Data Wrangling Phase

✓

The project documents any changes that were made to clean the data, such as merging multiple files, handling missing values, etc.

Good work in implementing a Data Wrangling Phase
Suggestion

The most important aspect of Data Wrangling is to clean or transform the data preparing it for analysis.

One main issue is having missing data while conducting analysis, which can provide skew/bias results. Luckily there are a few methods that Pandas provide to deal with these issues:

- The first thing to do is to always Identify the missing values within the dataset. The few steps after this explain how to deal with the missing data
- If there are columns with a few rows of missing data the Dropna method could be used to drop the missing rows.
- If there are rows with missing data the Fillna-method can be used instead of dropping them completely (This method can vary with the data and the project)
- The final option is if there are way too many missing values within a column it is best to drop the column completely using the Drop-column-method

Data Wrangling does not only involve Identifying and dealing with missing values but also involves in transforming the data to a more effective state to target the analysis. Here are other wrangling methods:

- Binning or Cutting Groups continuous or numerical values into smaller groups or ‘bins’
- Pandas-Dummies Transforms categorical data into dummy/indicator variables

Exploration Phase

✓

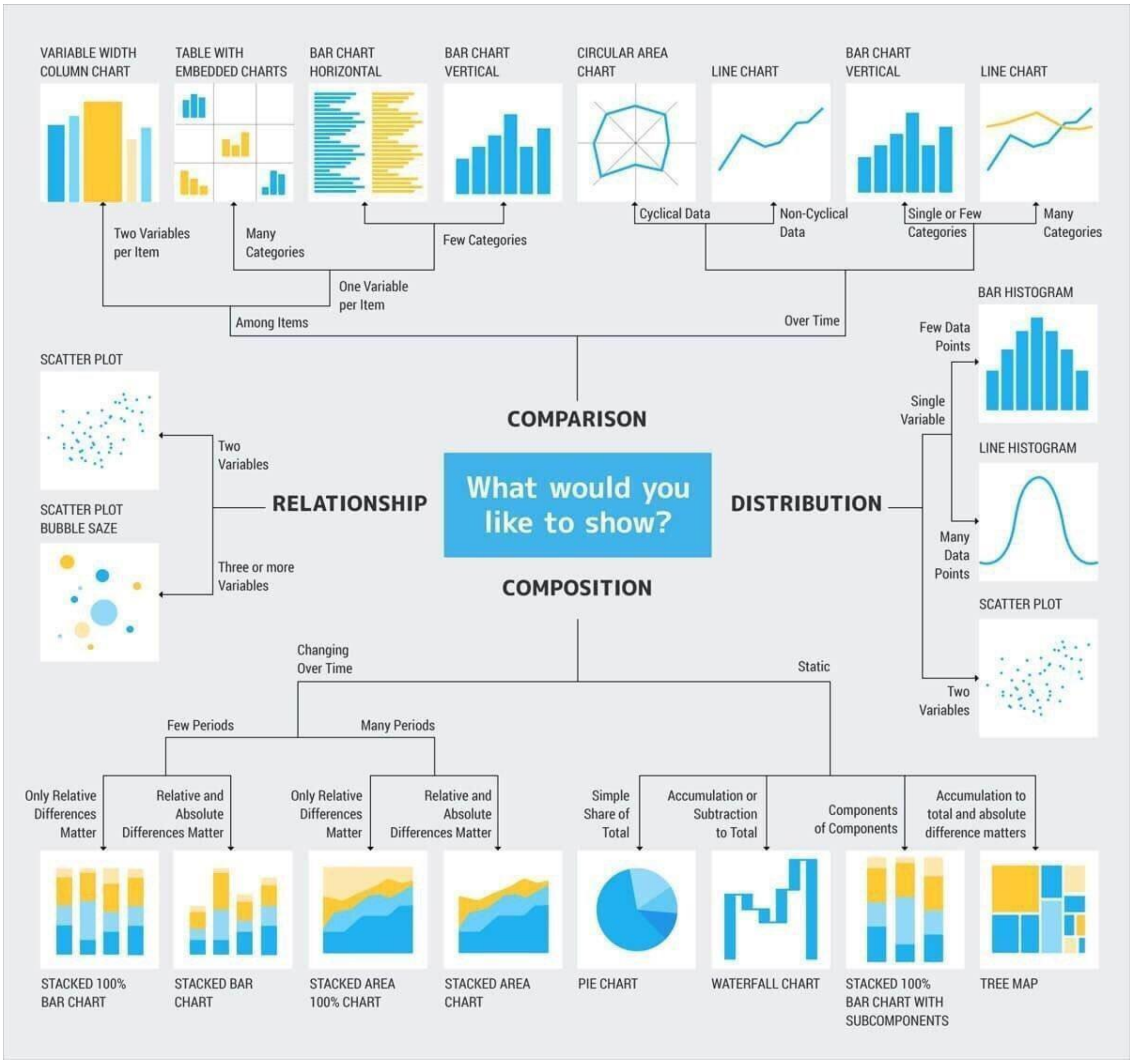
The project investigates the stated question(s) from multiple angles. At least three variables are investigated using both single-variable (1d) and multiple-variable (2d) explorations.

✓

The project's visualizations are varied and show multiple comparisons and trends. Relevant statistics are computed throughout the analysis when an inference is made about the data.

At least two kinds of plots should be created as part of the explorations.

Very good ! for future projects let me recommend you these tools to choose your visualizations



Conclusions Phase

✓

The results of the analysis are presented such that any limitations are clear. The analysis does not state or imply that one change causes another based solely on a correlation.

Congratulations, your project is super impressive 😃

Communication

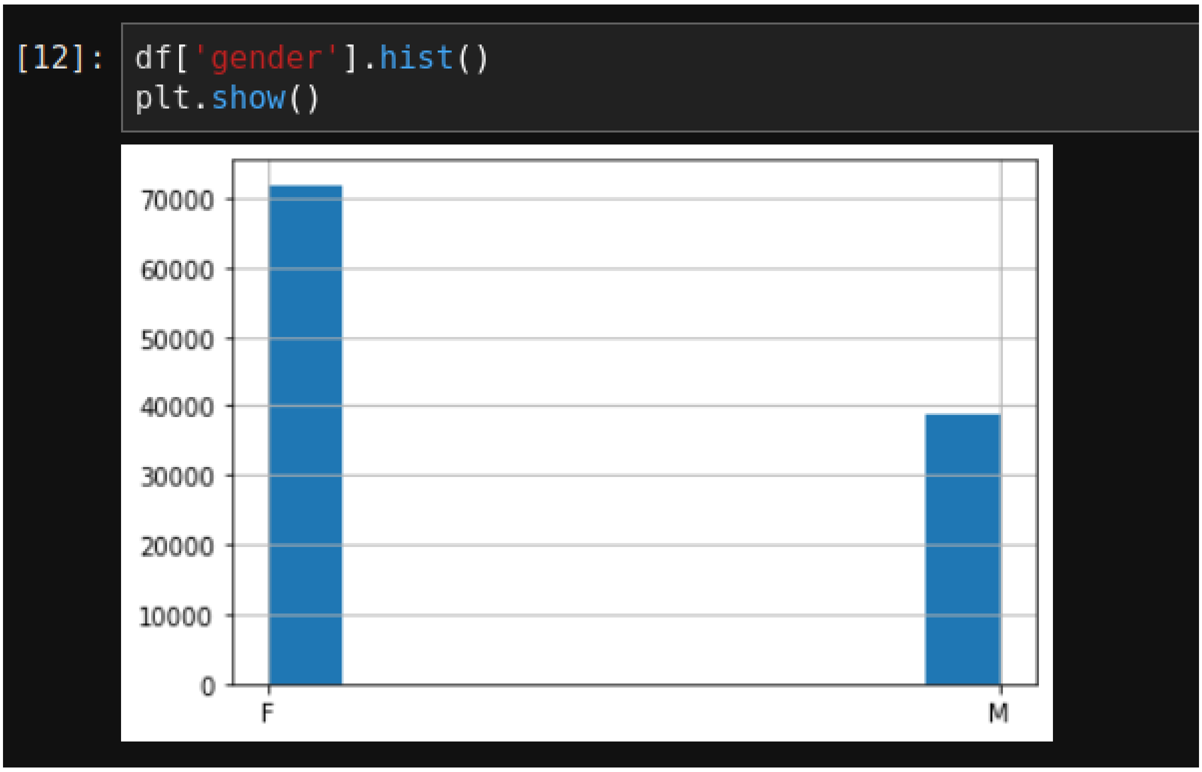
✓

Reasoning is provided for each analysis decision, plot, and statistical summary.

Fantastic 👏

✓

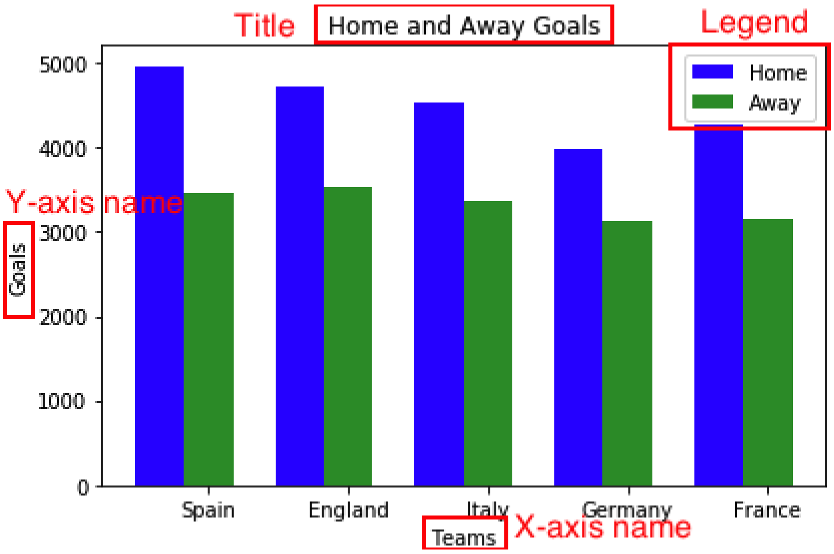
Visualizations made in the project depict the data in an appropriate manner that allows plots to be readily interpreted.



```
[12]: df['gender'].hist()
      plt.show()
```

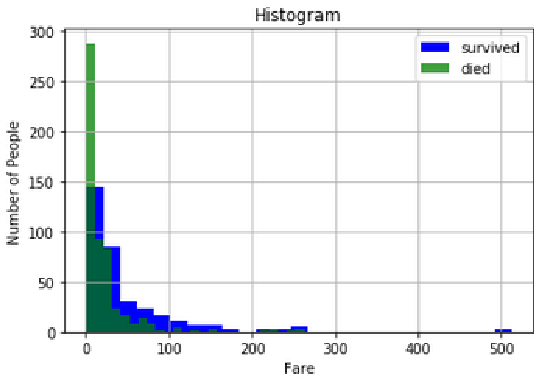
Please make sure that each graph has the following three characteristics:

1. A title
2. Names of the axes (in the X and Y axis)
3. Labels



Here are a Few samples of how to do it:

```
In [19]: plt.hist(df.Fare[df.Survived == True], 25, facecolor='b', alpha=1, label='survived');
         plt.hist(df.Fare[df.Survived == False], 25, facecolor='g', alpha=0.75, label='died');
         plt.legend()
         plt.xlabel('Fare')
         plt.ylabel('Number of People')
         plt.title('Histogram')
         plt.grid(True)
```



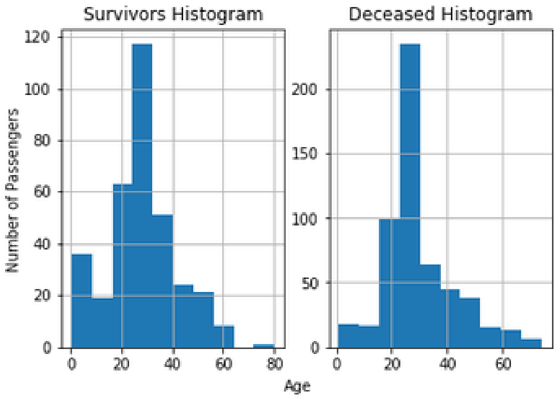
Is passenger age associated with survival?

```
In [15]: fig, axes = plt.subplots(1, 2)

         df.Age[df.Survived == True].hist(label='survived', ax=axes[0])
         df.Age[df.Survived == False].hist(label='survived', ax=axes[1])

         axes[0].set_title('Survivors Histogram')
         axes[1].set_title('Deceased Histogram');

         fig.text(0.5, 0.02, 'Age', ha='center');
         fig.text(0.04, 0.5, 'Number of Passengers', va='center', rotation='vertical');
```



DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START