UDACITY

‹ Return to "Data Analyst Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

Investigate a Dataset

REVIEW

HISTORY

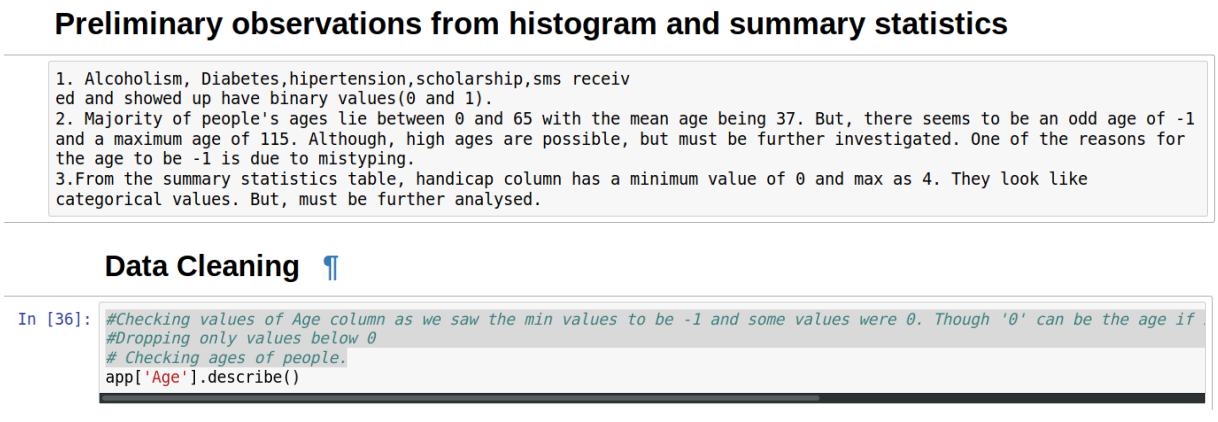
Meets Specifications

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.

One important thing you have to do is to only comment to document your code, any type of analysis should be included in a markdown cell (just go to a normal cell, click in the menu "Cell", then "Cell Type" and finally select "Markdown")



Preliminary observations from histogram and summary statistics

```
1. Alcoholism, Diabetes,hipertension,scholarship,sms receiv
ed and showed up have binary values(0 and 1).
2. Majority of people's ages lie between 0 and 65 with the mean age being 37. But, there seems to be an odd age of -1
and a maximum age of 115. Although, high ages are possible, but must be further investigated. One of the reasons for
the age to be -1 is due to mistyping.
3.From the summary statistics table, handicap column has a minimum value of 0 and max as 4. They look like
categorical values. But, must be further analysed.
```

Data Cleaning ¶

```
In [36]: #Checking values of Age column as we saw the min values to be -1 and some values were 0. Though '0' can be the age if
         #Dropping only values below 0
         # Checking ages of people.
         app['Age'].describe()
```

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.

I suggest you make a brief summary at the end of the data cleaning section, list the steps you took and explain why they are the best for this dataset (do not forget to use a markdown cell for this)

Now that my data is loaded, I'll make the following changes:

- Correct spelling of the 'Hipertension' & 'Handcap' column headers.
- Standardize column formatting with all lowercase letters.
- Change 'scheduledday' and 'appointmentday' to pandas datetime format.
- Remove 'appointmentid' column since the info it contains isn't useful.
- Create a column indicating whether or not the appointment date is within 7 days of when it was scheduled. I anticipate that appointments made within a short time frame (7 days) will have higher turnout.
- Create another column indicating whether or not the appointment date is within 1 day of when it was scheduled.
- Filter the data to only include patients older than 18 years old, but also less than 100 years old. This also removes the patient with an age of -1.
- Change 'patientid' column type to int64 to fix exponential formatting.
- Create age_group column to group by age.
- To minimize confusion, change 'no-show' column name to 'attended' and make the column's formatting binary like the other yes/no columns. A '1' indicates the patient attended, a '0' indicates they were a no-show. New column will be dtype int64.

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)

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

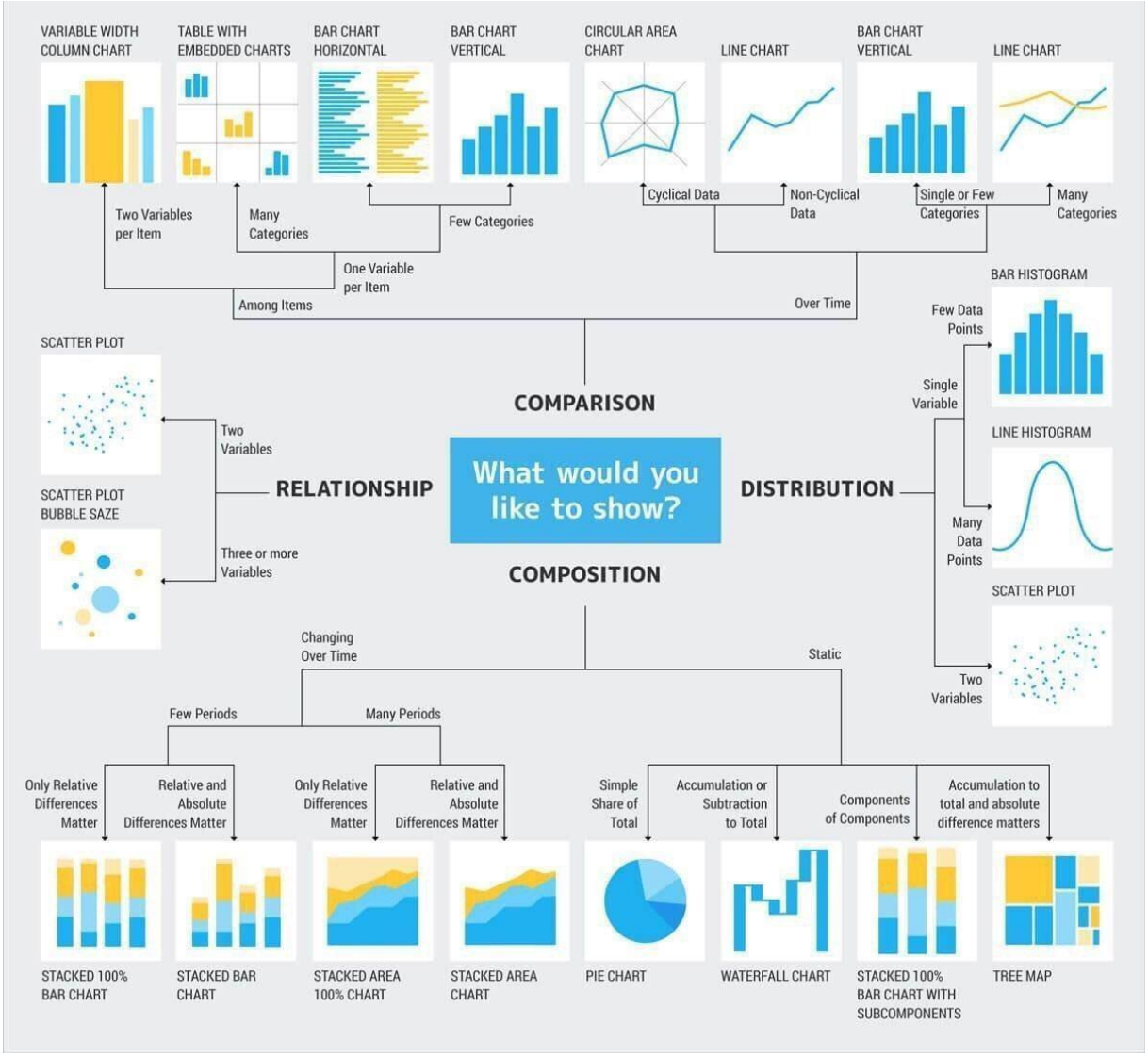
Exploration Phase

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.

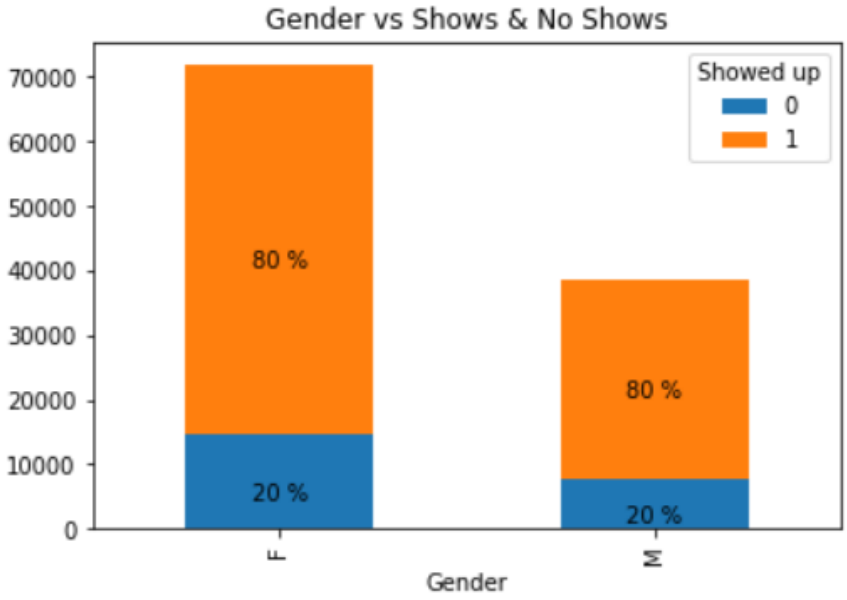
Congratulations, your project is super impressive 😃

Communication

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

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

```
Out[64]: Text(0.5, 1.0, 'Gender vs Shows & No Shows')
```

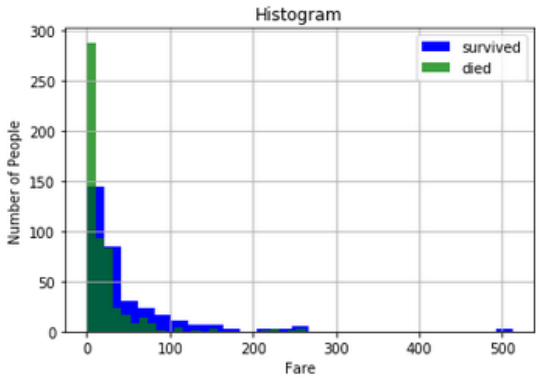


Please make sure that each graph has the following three characteristics:

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)
```



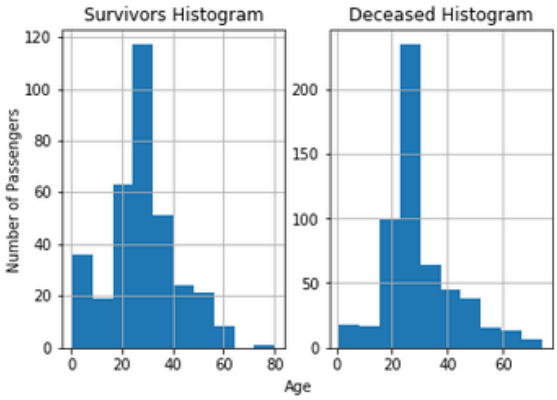
Is passenger age associated with survival?

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