UDACITY

‹ Return to Classroom

Investigate a Dataset

REVIEW

HISTORY

Meets Specifications

Excellent!

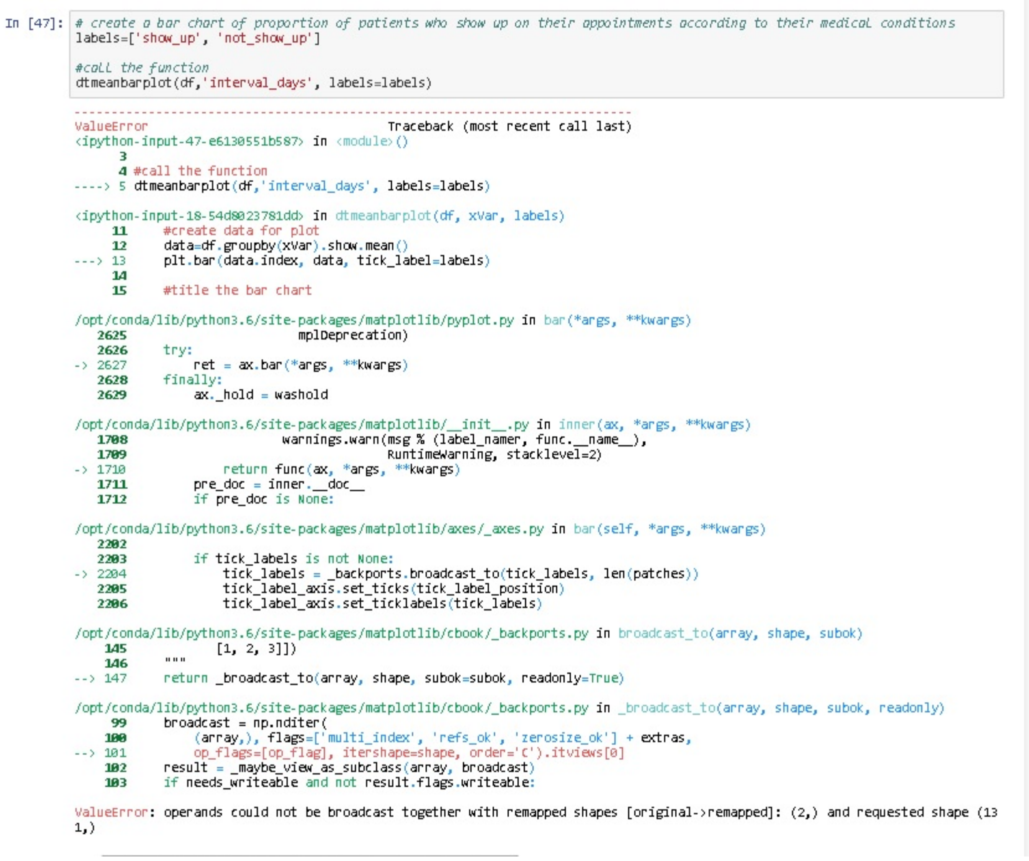
This is a great report, you have some interesting questions that address important aspects of the data and your analysis allows us to derive a good insight from the data. As you continue with the program forward, please do not hesitate to post questions in the knowledge forum if you have any.

Please see my comments inside the review. If you have any further questions please do not hesitate to post a question in the knowledge forum.

Code Functionality

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

The code returns an error, please make sure all the code returns what is expected with no error,
In this case, it seems that the dimension of the input does not fit the operation inside the function.



```
In [47]: # create a bar chart of proportion of patients who show up on their appointments according to their medical conditions
         labels=['show_up', 'not_show_up']

         #call the function
         dtmeanbarplot(df,'interval_days', labels=labels)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-47-e6130551b587> in <module>()
      3 
      4 #call the function
----> 5 dtmeanbarplot(df,'interval_days', labels=labels)

<ipython-input-18-54d8023781dd> in dtmeanbarplot(df, xVar, labels)
     11     #create data for plot
     12     data=df.groupby(xVar).show.mean()
---> 13     plt.bar(data.index, data, tick_label=labels)
     14 
     15     #title the bar chart

/opt/conda/lib/python3.6/site-packages/matplotlib/pyplot.py in bar(*args, **kwargs)
   2625                       mplDeprecation)
   2626     try:
-> 2627         ret = ax.bar(*args, **kwargs)
   2628     finally:
   2629         ax._hold = washold

/opt/conda/lib/python3.6/site-packages/matplotlib/__init__.py in inner(ax, *args, **kwargs)
   1708                     warnings.warn(msg % (label_namer, func.__name__),
   1709                                   RuntimeWarning, stacklevel=2)
-> 1710             return func(ax, *args, **kwargs)
   1711         pre_doc = inner.__doc__
   1712         if pre_doc is None:

/opt/conda/lib/python3.6/site-packages/matplotlib/axes/_axes.py in bar(self, *args, **kwargs)
   2202 
   2203         if tick_labels is not None:
-> 2204             tick_labels = _backports.broadcast_to(tick_labels, len(patches))
   2205             tick_label_axis.set_ticks(tick_label_position)
   2206             tick_label_axis.set_ticklabels(tick_labels)

/opt/conda/lib/python3.6/site-packages/matplotlib/cbook/_backports.py in broadcast_to(array, shape, subok)
    145            [1, 2, 3]])
    146     """
--> 147     return _broadcast_to(array, shape, subok=subok, readonly=True)

/opt/conda/lib/python3.6/site-packages/matplotlib/cbook/_backports.py in _broadcast_to(array, shape, subok, readonly)
     99     broadcast = np.nditer(
    100         (array,), flags=['multi_index', 'refs_ok', 'zerosize_ok'] + extras,
--> 101         op_flags=[op_flag], itershape=shape, order='C').itviews[0]
    102     result = _maybe_view_as_subclass(array, broadcast)
    103     if needs_writeable and not result.flags.writeable:

ValueError: operands could not be broadcast together with remapped shapes [original->remapped]: (2,) and requested shape (13
1,)
```

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.

Pandas and Numpy Operators

The analysis makes use of the NumPy and Pandas libraries, vector operators are employed instead of loops and lists.

It is awesome, you use the function `.info()` and `.describe()` to examine the structure of the entire data, identify missing values, and the summary statistics for the numerical features.

- Group-by: http://pandas.pydata.org/pandas-docs/stable/groupby.html
- Value-Counts: https://chrisalbon.com/python/data_wrangling/pandas_dataframe_count_values/

The code makes use of at least 1 function to avoid repetitive code. The code contains good variable names that have meaning. Comments and docstrings are used as needed to document code functionality making it easy to read.

It is awesome that you created a custom function that reduces repetitions and simplifies the code.

Quality of Analysis

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

Project Introduction

The report states clear and relevant questions that are being addressed by the following 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.

Documenting Data Preparation

Well Done for reporting the missing values in the dataset and documenting the changes made in the dataset. This is important because it makes it possible for the readers to repeat your analysis if needed. Please note that for some of the columns, a major portion of the data is missing. That might affect the result of the analysis. Think about other ways to handle missing values.

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.

Single and Multiple Variable Explorations

The analysis makes use of both single and multiple variable explorations to investigate different features and the relations between these features in the dataset.

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.

Visualization and Relevant Statistics

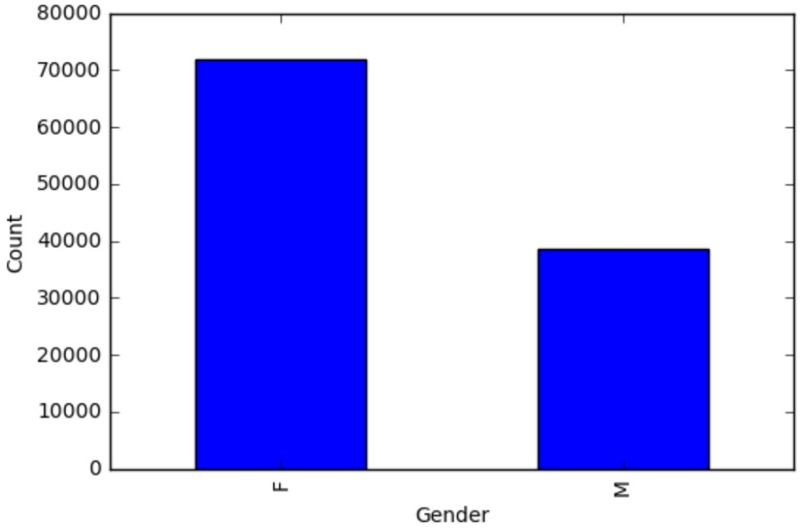
The report makes use of different chart type to explore and depict the insights and the results of the analysis. I strongly encourage you to include the relevant statistics next to each figure. Below I show a few

examples of different chart types and the relevant descriptive statistics.

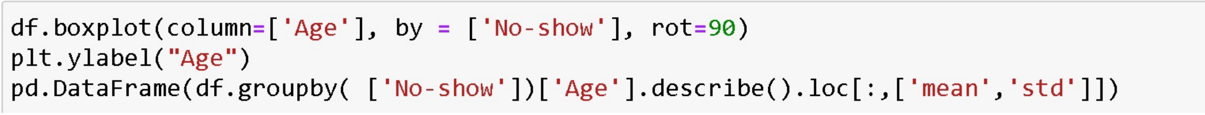
Single variable bar plot depict the count distribution for categorical variable

```
df.groupby(['Gender'])['PatientId'].count().plot(kind='bar').set_ylabel('Count')
df.groupby(['Gender' ])[['PatientId']].count()
```

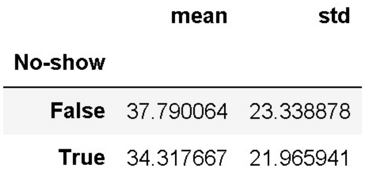
	PatientId
Gender	
F	71840
M	38687



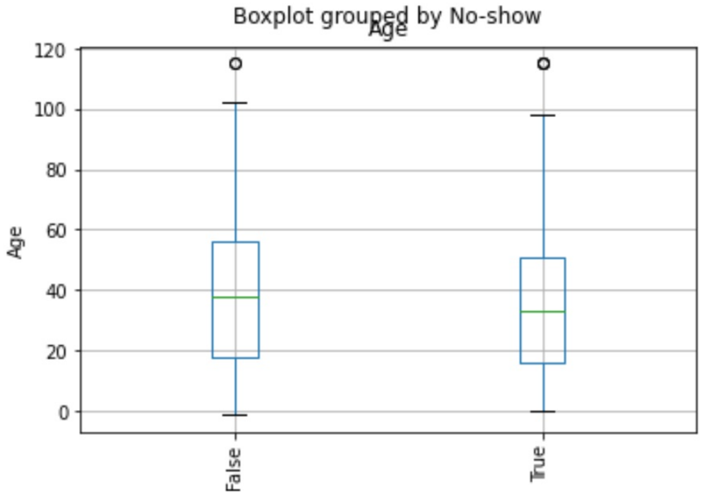
A simple box plot allow you to depict the distribution of a continuous feature for different categories,

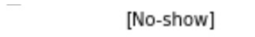


```
df.boxplot(column=['Age'], by = ['No-show'], rot=90)
plt.ylabel("Age")
pd.DataFrame(df.groupby( ['No-show'])['Age'].describe().loc[:,['mean','std']])
```



	mean	std
No-show		
False	37.790064	23.338878
True	34.317667	21.965941

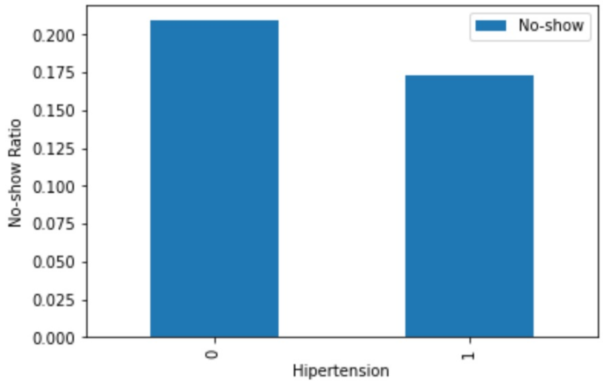




Bivariate bar plot allow you to depict the ratio of one feature in different categories

```
df.groupby(['Hipertension'])[['No-show']].mean().plot(kind='bar').set_ylabel('No-show Ratio')
df.groupby(['Hipertension'])[['No-show']].mean()
```

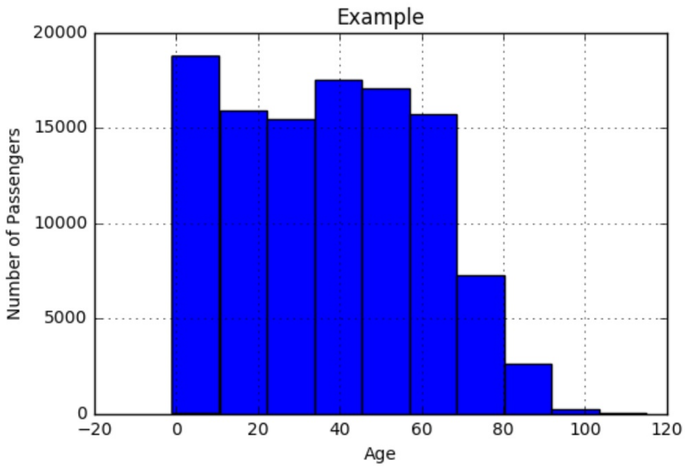
	No-show
Hipertension	
0	0.209037
1	0.173020



Histograms depict the distribution of continuous features.

```
ax = df['Age'].hist()
ax.set_ylabel('Number of Passengers')
ax.set_xlabel('Age')
ax.set_title('Example')
pd.DataFrame(df['Age'].describe())
```

	Age
count	110527.000000
mean	37.088874
std	23.110205
min	-1.000000
25%	18.000000
50%	37.000000
75%	55.000000
max	115.000000



Conclusions Phase

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.

Analysis Shortcoming & Data Limitations

Excellent! The report includes a discussion about the limitations and the shortcomings of the analysis and the dataset.

Communication

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

Analysis Description

The analysis follows a logical flow, the discussion includes reasonings, explanations about the analysis and relevant statistics to quantify the results and insights.

Only for Project Reviewers (No student work needed)

This rubric is ungraded. The reviewer will provide the student a code review.

This rubric will be ungraded. The reviewer will brief the students about the concepts learned in this section of the Nanodegree program.

I would just like to mention the data analysis process, usually, we start by obtaining the data and reading that into the computer.
The first step is the cleaning of the data, I will use the function info to identify the missing values. And then next I will use a histogram or bar plot to examine the structure of the data.

After we are familiar with the data we will start looking for relations in the data, that should be pursued first by visualization and next using statistical tests to verify if the changes we see in the visualizations are indeed significant.

As we go over these steps it is important to document and explain each step. don't assume that your reader knows python, instead explain in plain English each step of the analysis the methods that you are using and the results that you obtained.

This rubric is ungraded. If the learner has asked a question pertaining to the implementation of the project, the reviewer will provide an answer along with links to any helpful resources.

DOWNLOAD PROJECT