Investigate a Dataset

REVIEW | HISTORY

Requires Changes

1 specification requires changes

This is a great report, you state interesting questions that address important aspects of the data and your analysis allows you to produce excellent insights from the data. As you continue with the program forward, I encourage you to post more questions in the knowledge forum, which will help you and other students.

Please see my comments inside the review. If you have any further questions, please do not hesitate to post them 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.

Code Functionality & Readability

The code is well-formatted and appropriately commented. That make
to follow the analysis steps and identify a specific functional operatio
documented and easy-to-follow code will save you a lot of time when
to debug it. If you like you can examine the python style document.
https://www.python.org/dev/peps/pep-0008/

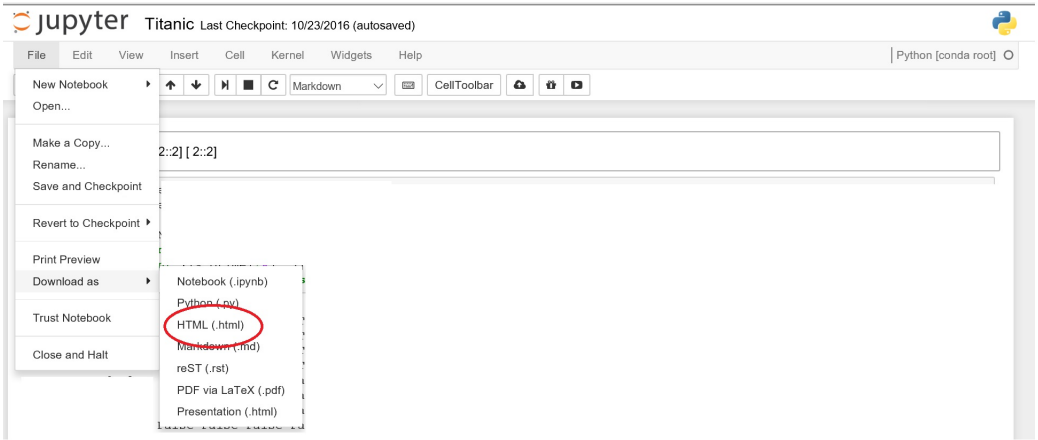
Rules for Python variables Names,

I would like to encourage you to look into this link below, which discusses Rules and conventions for Python variable Names. This is important because that will allow other programmers, to better understand and follow your code. In many cases, you will be part of a team that will appreciate the clarity that these Rules and conventions provide.
https://www.w3schools.com/python/gloss_python_variable_names.asp

Python Comments

You can also look into this link that includes a discussion about python convention for code comments https://www.w3schools.com/python/python_comments.asp

Please include *also an HTML* version of the report with the submission. That is useful not just for us, as project reviewers, but also for any other reader who wants to see your analysis but does not have the technical means to access and open the notebook (ipynb) file.



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 both single and multiple variable explorations to investigate different features and the relations between these features in the dataset. It is awesome that you demonstrate the use of pandas and NumPy, these libraries are widely used for data analysis and data manipulation.

Built In functions

It is awesome that you make use of the functions `.info()` and `.desc` examine the structure of the entire data, identify missing values, and summary statistics for the numerical features.

- `DataFrame.groupby`: Allows you to aggregate the data according specific categories: http://pandas.pydata.org/pandas-docs/stable

/groupby.html
For example, here I calculate different statistics for each category

```
df.groupby(['Sex' ])['Age'].median()
```

```
Sex
female    27.0
male      29.0
Name: Age, dtype: float64
```

```
df.groupby(['Sex' ])['Age'].mean()
```

```
Sex
female    27.915709
male      30.726645
Name: Age, dtype: float64
```

```
df.groupby(['Sex' ])['Age'].std()
```

```
Sex
female    14.110146
male      14.678201
Name: Age, dtype: float64
```

- `DataFrame.value_counts` : Return a Series containing counts of unique rows in the DataFrame https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.value_counts.html
- `pandas.cut` : This allows you to easily cut continuous variables into segments. https://pandas.pydata.org/docs/reference/api/pandas.cut.html
  Here I cut the age to several selected ranges and add a new column with this new information

```
[13]: df['Age_range' ] = pd.cut(x=df['Age' ], bins=[0, 20, 40, 60, 80, 100])
      df['Age_range' ].sample(4)

Out[13]: 592    (40, 60]
         562    (20, 40]
         599    (40, 60]
         724    (20, 40]
         Name: Age_range, dtype: category
         Categories (5, interval[int64, right]): [(0, 20] < (20, 40] < (40, 60] < (60, 80] < (80, 100]]
```

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.
Usually, the documentation should be inside the function, which allows you to use help and look into the docstring.
https://www.programiz.com/python-programming/docstrings
https://pythonprogramminglanguage.com/functions/

```
# no-show graph
repeats = ['repeats','uniques']
# include unique duplicates and unique solos
noshows = [no_showDupCnt,no_showUCnt]
nomiss = [no_missDupCnt,no_missUCnt]

def func(pct, allvals):
    absolute = int(round(pct/100.*np.sum(allvals)))
    return "{:.1f}%\n({:d})".format(pct, absolute)
```

Quality of Analysis

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

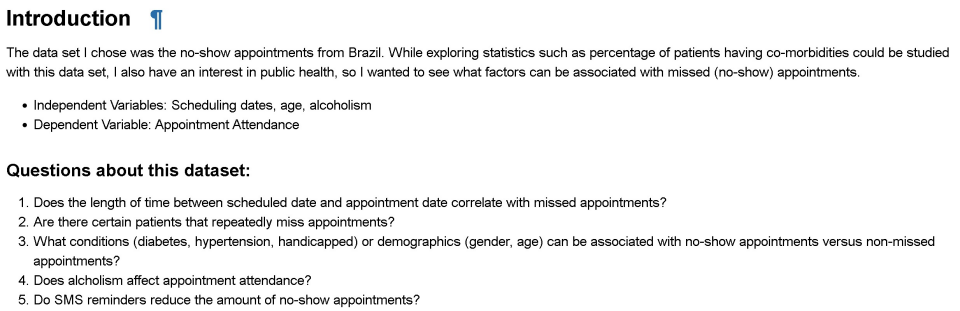
Rate this review

START

Project Introduction

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

It will be very useful for your readers if you expand the introduction to discuss the analysis that you intend to implement.



Introduction ¶

The data set I chose was the no-show appointments from Brazil. While exploring statistics such as percentage of patients having co-morbidities could be studied with this data set, I also have an interest in public health, so I wanted to see what factors can be associated with missed (no-show) appointments.

- Independent Variables: Scheduling dates, age, alcoholism
- Dependent Variable: Appointment Attendance

Questions about this dataset:

1. Does the length of time between scheduled date and appointment date correlate with missed appointments?
2. Are there certain patients that repeatedly miss appointments?
3. What conditions (diabetes, hypertension, handicapped) or demographics (gender, age) can be associated with no-show appointments versus non-missed appointments?
4. Does alcholism affect appointment attendance?
5. Do SMS reminders reduce the amount of no-show appointments?

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 - Be Comprehensive

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.

Diversified visualizations

The report uses 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 differen types and the relevant descriptive statistics.

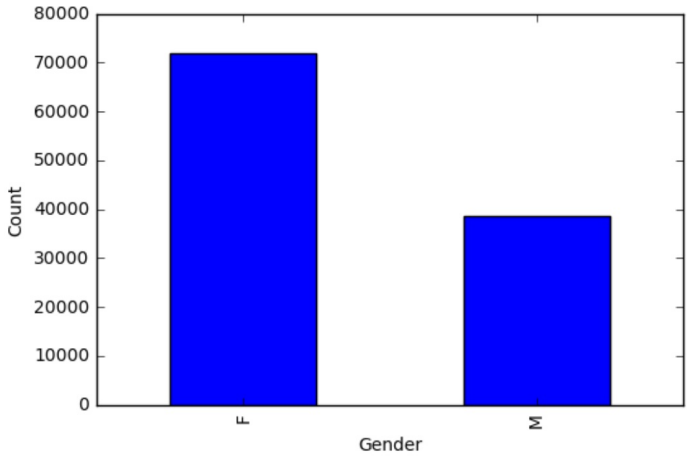
Single variable bar plot depict the count distribution for categorical va

Rate this review

START

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

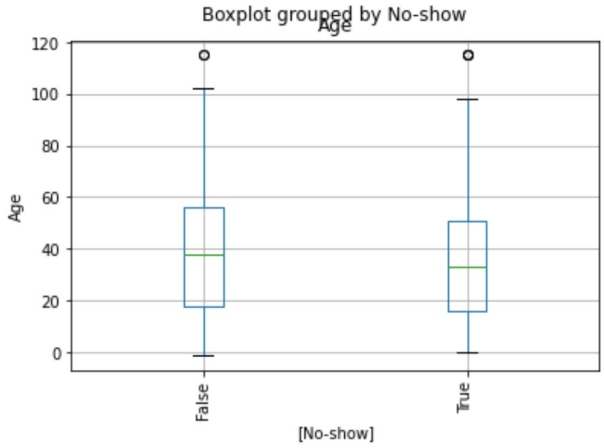
	PatientId
Gender	
F	71840
M	38687



A simple box plot allows 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']])
```

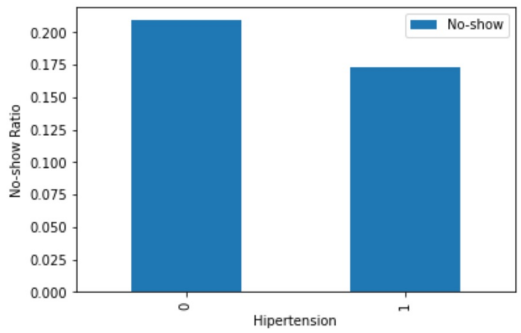
	mean	std
No-show		
False	37.790064	23.338878
True	34.317667	21.965941



Bivariate bar plot allows 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()
```

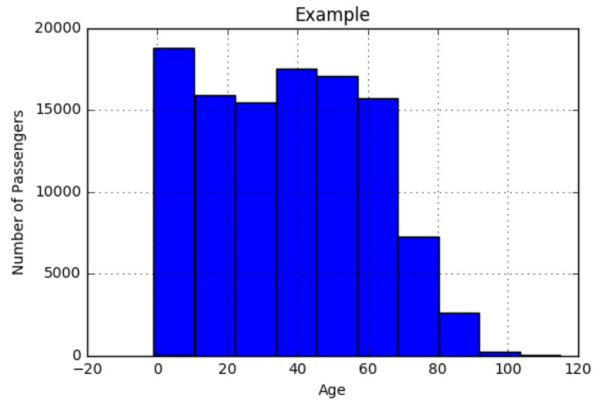
Hipertension	No-show
0	0.209037
1	0.173020



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



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 - Be detailed and descriptive!

Well Done for reporting the missing values in the dataset and docume
changes made in the dataset. This is important because it makes it pc

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.

Important

Please add markdown cells (instead of the comments inside the code), to describe each step of the wrangling section.

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 Shortcomings & Data Limitations

Excellent! The report includes a discussion about the limitations and 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.

Visualization Clarity

The charts in the report should be clear and easy to interpret. please add the x and the y axis for each chart, please add a descriptive title for each ch
Please add a short discussion under each chart explaining what the ch
depicts and your insight.

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 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, which 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.

RESUBMIT PROJECT

DOWNLOAD PROJECT

Rate this review

START

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

Rate this review

START