UDACITY

‹ Return to Classroom

Investigate a Dataset

REVIEW

HISTORY

Meets Specifications

Excellent!

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 makes it easy to follow the analysis steps and identify a specific functional operation. 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 that discusses Rules for Python variable Names, when

using the python convention, you make sure that the code is easy to follow by other programmers.
https://www.w3schools.com/python/gloss_python_variable_names.asp

Python Comments

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

- **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.**

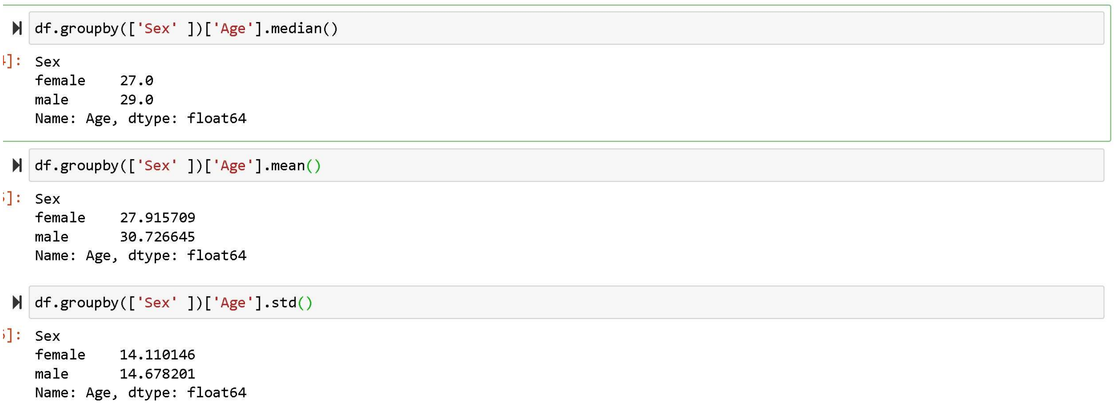
Pandas and Numpy Operators

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

Built In functions

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

- `DataFrame.groupby` : Allows you to aggregate the data according to 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
```

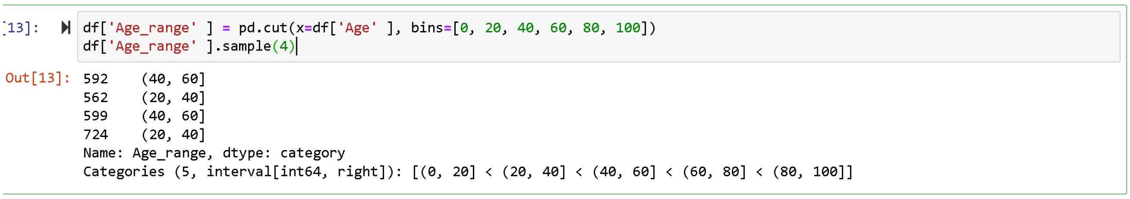
```
df.groupby(['Sex' ])['Age'].mean()
```

```
]: 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

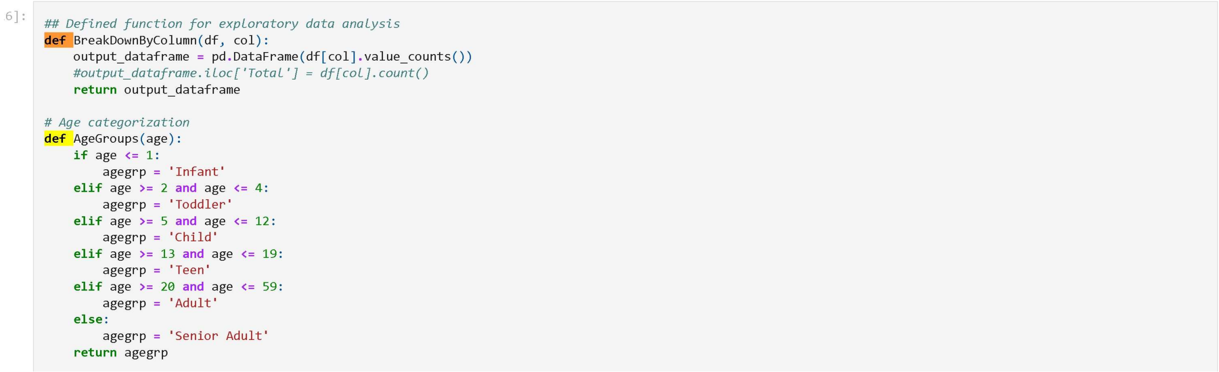


```
[13]:   df['Age_range' ] = pd.cut(x=df['Age' ], bins=[0, 20, 40, 60, 80, 100])
        df['Age_range' ].sample(4)
```

```
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 comments and meaningful variable names, 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, that allows you to use help, and look into the docstring.
https://www.programiz.com/python-programming/docstrings
https://pythonprogramminglanguage.com/functions/



6]:

```
## Defined function for exploratory data analysis
def BreakDownByColumn(df, col):
    output_dataframe = pd.DataFrame(df[col].value_counts())
    #output_dataframe.iloc['Total'] = df[col].count()
    return output_dataframe

# Age categorization
def AgeGroups(age):
    if age <= 1:
        agegrp = 'Infant'
    elif age >= 2 and age <= 4:
        agegrp = 'Toddler'
    elif age >= 5 and age <= 12:
        agegrp = 'Child'
    elif age >= 13 and age <= 19:
        agegrp = 'Teen'
    elif age >= 20 and age <= 59:
        agegrp = 'Adult'
    else:
        agegrp = 'Senior Adult'
    return agegrp
```

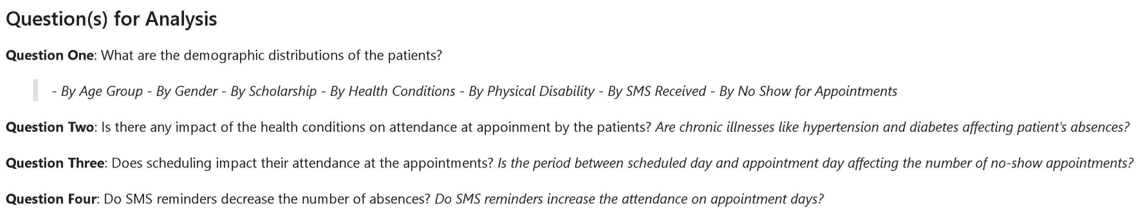
Quality of Analysis

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.

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



Question(s) for Analysis

Question One: What are the demographic distributions of the patients?

> *- By Age Group - By Gender - By Scholarship - By Health Conditions - By Physical Disability - By SMS Received - By No Show for Appointments*

Question Two: Is there any impact of the health conditions on attendance at appoinment by the patients? *Are chronic illnesses like hypertension and diabetes affecting patient's absences?*

Question Three: Does scheduling impact their attendance at the appointments? *Is the period between scheduled day and appointment day affecting the number of no-show appointments?*

Question Four: Do SMS reminders decrease the number of absences? *Do SMS reminders increase the attendance on appointment days?*

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 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.**
- **The project explores at least three variables in relation to the primary question. This can be an exploratory relationship between three variables of interest, or looking at how two independent variables relate to a single dependent variable of interest.**
- **The project performs both single-variable (1d) and multiple-variable (2d) explorations.**

Single and Multiple Variable Explorations - Be Comprehensive

It is important to perform a single variable analysis ***for each feature that is included in the report/analysis***. That will allow you to appreciate the distribution and perhaps outliers for these variables.

For that you can use visualization to examine each feature which is more informative and will help you understand the data much better. You can use a simple histogram (continuous features) or barplot (categorical features).

- **The project's visualizations are varied and show multiple comparisons and trends.**
- **At least two kinds of plots should be created as part of the explorations.**
- **Relevant statistics are computed throughout the analysis when an inference is made about the data.**

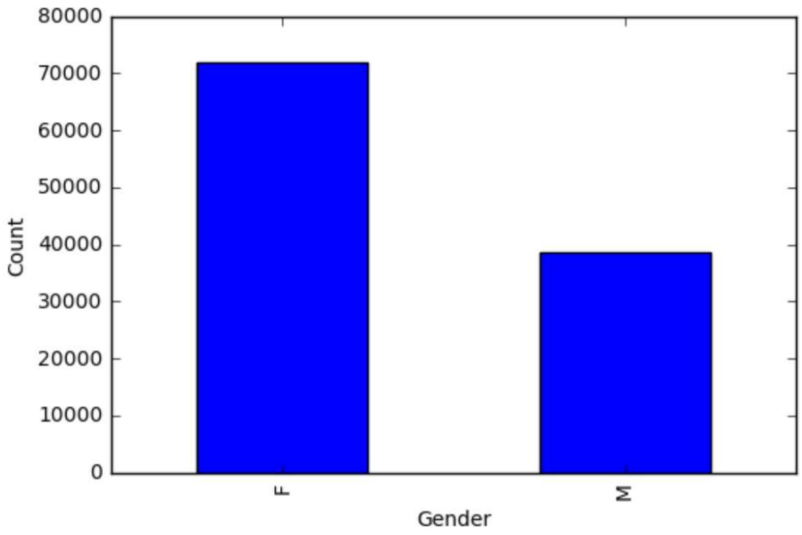
Diversified visualizations

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 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()
```

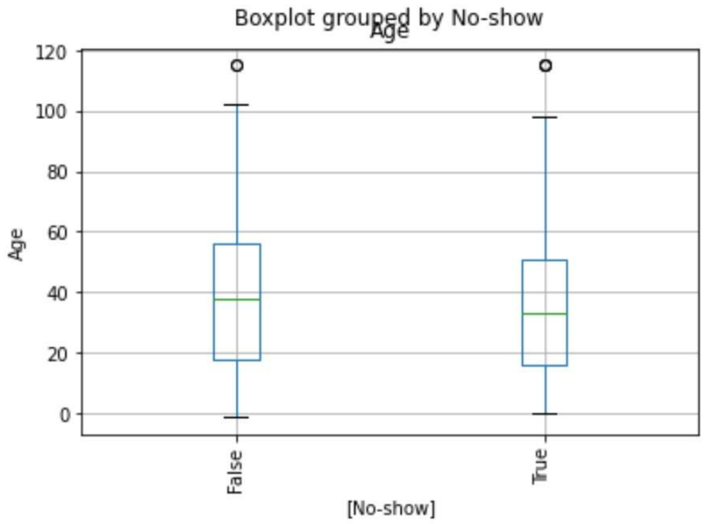
	PatientId
Gender	
F	71840
M	38687



A simple box plot 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']])
```

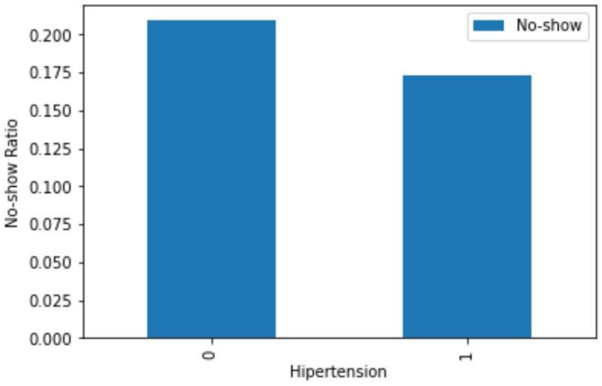
	mean	std
No-show		
False	37.790064	23.338878
True	34.317667	21.965941



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()
```

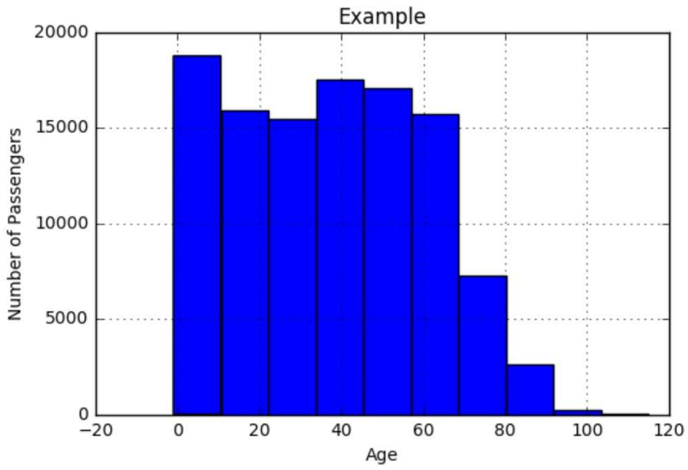
	No-show
Hipertension	
0	0.209037
1	0.173020



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 Conclusions have reflected on the steps taken during the data exploration.

- The Conclusions have summarized the main findings in relation to the question(s) provided at the beginning of the analysis accurately.
- The project has pointed out where additional research can be done or where additional information could be useful.
- The conclusion should have at least 1 limitation explained clearly.
- 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 shortcomings of the analysis and the dataset.

Communication

- The code should have ideally the following sections: Introduction; Questions; Data Wrangling; Exploratory Data Analysis; Conclusions, Limitation.
- Reasoning is provided for each analysis decision, plot, and statistical summary.
- Interpretation of plots and application of statistical tests should be correct and without error.
- Comments are used within the code cells.
- Documented the flow of analysis in the mark-down cells.

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.

Visualizations made in the project depict the data in an appropriate manner (i.e., has appropriate labels, scale, legends, and plot type) that allows plots to be readily interpreted.

All the charts are clear and easy to interpret.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START