**Requires Changes**

4 specifications require changes

Dear Student,  
Not a bad start to your project. Good first attempt!  
However, some work is there to be done. There are a few changes to be made before this project can be approved as fully **MEETS SPECIFICATIONS**. Focus on the following sections:

* At least **ONE** re-usable function is required to show knowledge to avoid repetitive code.
* A conclusion is provided but there are no limitations present within it. Ensure to add the limitations.
* Ensure to continue to provide reasoning for all of the analysis including  
  visualizations and coding steps. Focus on the visualizations do not have any after being outputted.
* Finally, ensure to title and label all visualizations. A few are still missing titles/labels.

For the sections that **REQUIRE CHANGES** and sections where necessary comments, recommendations, and suggestions have been provided to help better understand the requirements and assist with helping to pass this project successfully.  
Do not worry or be disheartened this is a learning process I am sure you will do great! Also, we are here to help and guide you in the right direction! If you have any questions/queries feel free to write them in the next review or you can post in the [Knowledge-Forum](https://knowledge.udacity.com/) questions, queries, and follow-ups!

All the best for the next submission!  
Cheers!

**Code Functionality**

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

Good start!  
The following requirements for this section **MEET SPECIFICATIONS!**:

* The code runs well without any errors presented/ outputted.
* The code is properly formatted/documented and is sufficient enough to produce the results described

**RECOMMENDATION**

* For better readability, it is better to remove the **TIPS** and **HEADERS** for better readability.
* Secondly, remove the Udacity instruction comments that are TO DOs

**IMPORTANT NOTE FOR CODING PROJECTS(FRIENDLY REVIEWER ADVICE)**

When programming code, as developers we do encounter the dilemma of "annoying" bugs/errors/issues even indentation and formatting issues  
. Many big companies and professional developers actually look at this and take it heavily into consideration. It is always recommended practice good object-oriented programming skills and learn to debug and format the code after finishing.  
Now, there are many ways to do this project, but it is best suited and recommended by Udacity to do it in the **Jupyter-Notebook** because all code, visualizations, and explanations can be done in one space.  
However, the drawback with Jupyter-Notebook is that sometimes debugging errors(if they occur) can be difficult.  
Luckily, IDEs such as [Spyder-IDE](https://www.spyder-ide.org/) and [PyCharm-IDE](https://www.jetbrains.com/pycharm/) have been updated and can now integrate the **Juptyter Notebook** extension ipynb within them. This makes debugging, formating, and editing a whole lot easier than before!

**NOTE** This is just for the Python programming language. Though it can also be applied to future, Data Analysts, Sciences, and Python projects as well!

* **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.**

Good work! The following requirements for this section **MEETS SPECIFICATIONS!**:

* Project successfully uses Numpy arrays, Pandas Series and Data frames where appropriate
* Practices effeciency in cases where vectorized operations and built in functions are used instead of "for" loops!

**Suggestion**

There are many built-in methods that can be used in the field of Python and Data Analysis/Sciences but here are a few which are commonly used and applicable for these projects:

* [Boolean-Indexing](http://pandas.pydata.org/pandas-docs/stable/indexing.html#boolean-indexing)
* [Group-by](http://pandas.pydata.org/pandas-docs/stable/groupby.html)
* [Value-Counts](https://chrisalbon.com/python/pandas_dataframe_count_values.html)
* [Series.map](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.map.html)
* [Working-with-text-data](https://pandas.pydata.org/pandas-docs/stable/text.html)

**NOTE** The project is not limited only to these methods mentioned above. It all depends on the analysis conducted. There are many other methods that can be used. Ensure to check the official : [Pandas-Data-Analysis-Documentation](https://pandas.pydata.org/)

**RECOMMENDATION (ONE TO CONSIDER!)**

This is a recommendation for this project and potential future projects. It is noticed that there is one or more instances of for loops within the project. Now, the code and tasks being carried out are not wrong they do work successfully which is great!  
However, as time goes by and as a developer, new methods will be learned which will make some for loops appear to be tedious or not efficient. for loops are tricky they are the basic form of iteration to learn but they can be inefficient in certain situations. In data analysis and the field of data sciences and etc, we try to emphasize the usage of built-in methods or depend less on for loops. This is just a note to keep in mind!  
The section below will focus on how to write a function!

* **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.**

Good start! Some of the following requirements for this section do meet the criterion:

* The code is well written with comments and good variables and actually carries out the tasks at hand.
* Good variable names are used in the project

**Required**

The code is not wrong. However, one of the requirements for this section is for at least **ONE** re-usable function to be implemented in the project.

**RECOMMENDATION**

Depending on the student's choice, a function can be placed anywhere in the project. However, a recommendation for where to create a function is for visualization(s).

This is helpful if multiple visualizations are being plotted the function comes in handy instead of writing repetitive code which tends to be considered inefficient coding practice.

**SUGGESTIONS**

Here are some suggestions on how to set up functions using Seaborn methods.

import seaborn as sb

# Function to create violin plot

def violingrid(x, y, \*\*kwargs):

""" Quick hack for creating violin plots with seaborn's PairGrid. """

ax = sb.violinplot(x=x, y=y, color='b', inner='quartile')

ax.tick\_params(rotation=90)

# A function to create a histogram

def hist\_plot\_by(x, xlabel, ylabel):

""" A function to create a Histogram """

ax = plt.subplot(1,1,1)

ax.hist(x)

ax.title('title here')

ax.set\_xlabel(xlabel)

ax.set\_ylabel(ylabel)

plt.show()

# Plot\_bar function to help with plotting bars using Seaborn

def plot\_bar(x, y, height, labels = None, title = None, xlabel = None, ylabel=None, color =

None):

sns.set\_style('darkgrid')

plt.bar(x, y, height, tick\_label = labels, color = color, width = 0.2)

plt.title(title, weight = 'bold', fontsize = 15)

plt.xlabel(xlabel, weight = 'bold', fontsize = 12)

plt.ylabel(ylabel, weight = 'bold', fontsize = 12)

**Comments**

A function can be created from any step of the code and adjusted with the parameters of the visualization. So for example, the function can take seaborn and matplotlib parameters, methods, or attributes!  
The above is just an example. The student is free to use the function examples as a guide and choose how to set up the function according to their project.  
Finally, to add to the point stray for loops are a " *NO-NO* " too! It is more efficient to put them in a function!

**HINT**: In this project, I notice there are lineplots and that they are using for loops. When observed the for loops for the lineplots are similarly structured and perform a similar style to provide the solution. Another idea is why not create a function there to negate the for loop! If the task is tedious feel free to use the examples of functions provided and implement it in the project!

**To Meet specifications**

* Add At least *ONE* reusable function to display knowledge to avoid repetitive code.

**Quality of Analysis**

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

Good work! The following requirements for this section **MEETS SPECIFICATIONS!**:

* The project has an Introductionary Paragraph with the questions that is very appealing to readers. It gives the agenda to the Project and what to expect when reading through the document.

**TIP**

Great Introductionary paragraphs that are properly written provide excellent anticipation for the reader and what to expect. Do this for all projects in the future as it gives insight.

Here are a few more questions that could be asked for this particular dataset.

* Does no-show count have a correlation with ages?
* Does no-show count have a correlation with receiving SMS?
* Does no-show count have a correlation with the living area?
* What is the ratio between genders, the likelihood of somebody showing up, and the age distribution in the dataset?
* Checking if there is any relationship between peoples personal ailment and no-shows

**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! The following requirements for this section **MEETS SPECIFICATIONS!**:

* The project successfully shows the Data Wrangling steps in code which are complete, valid, accurate and consistent.
* The Data Wrangling section is documented in writing and coincides with the coding steps used to wrangle the data.

**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 skewed/biased 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](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.isnull.html) 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](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.dropna.html)could be used to drop the missing rows.
* If there are rows with missing data the [Fillna-method](http://pandas.pydata.org/pandas-docs/version/0.17.0/generated/pandas.DataFrame.fillna.html) 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 or f it is not wanted, it is best to drop the column completely using the [Drop-column-method](http://pandas.pydata.org/pandas-docs/version/0.17.0/generated/pandas.DataFrame.drop.html)

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](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.cut.html) Groups continuous or numerical values into smaller groups or ‘bins’
* [Pandas-Dummies](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html)Transforms categorical data into dummy/indicator variables
* [Working-with-text-data](https://pandas.pydata.org/pandas-docs/stable/text.html)
* Creating new columns

**All these methods can be applied to this project!**

**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.**

Good work! The following requirements for this section **MEETS SPECIFICATIONS!**:

* The project successfully investigates/explores the analysis from multiple angles
* Has at least THREE cases of Univariate explorations (1D). Where only Single Variables are explored
* Has at least THREE cases of Multi-variable explorations (2D). Where Two or More Variables are explored.

**SUGGESTION**

Here is a good refresher to remind us what is Univariate and Multivariable explorations and how to create both simple but efficient univariate and multivariable explorations via code with examples:

1. [Simple-Univariate-Explorations](https://www.kaggle.com/code/lonewolf95/eda-101-univariate-analysis-for-beginners/notebook)
2. [SImple-Multivariable-Explorations](https://www.kaggle.com/code/sanikamal/introduction-to-multivariate-analysis/notebook)

**FUTURE NOTE TO CONSIDER**: Quite a bit of student, even analysts have a misconception that visualization is automatically a multivariable exploration. This is not true! There are different types of visualizations for each type of exploration whether it be univariate or multivariable(Look at the sources for more details). Just keep this in mind when conducting explorations in the project.

* **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.**

Good work! The following requirements for this section **MEETS SPECIFICATIONS!**:

* The project's visualizations are appealing by showing multiple comparisons, trends in different cases, and different variable combinations
* The Project has at least two different types of visualizations for a variety of purposes
* Each visualization appears to be relevant were necessary for making an inference about the data.

**Suggestion**

There are many graphs that can be used to visualize the data in many different ways such as:

* Histograms
* Barcharts
* Count plots
* Box plots
* Line graphs
* Pie charts
* Scatter plots
* Heatmaps

The list goes on...

**SUGGESTION 2**

The student should always feel free to explore these two libraries and their galleries for more variety as they are most efficient with working with Python :

* [Matplotlib-gallery](https://matplotlib.org/gallery.html)
* [Sea-born-gallery](https://seaborn.pydata.org/examples/index.html)
* [Python-Graph-Gallery](https://www.python-graph-gallery.com/)
* [Plotly-DocuementationLibrary](https://plotly.com/python/)

**NOTE**:

* One key thing to understand when plotting visualizations is which visualization is best for the type of data. For example, the main data types are numerical and string/categorical. There are different types of visualizations that can be applicable to both. Keep this in mind when plotting!

**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.**

Good start! The following requirement(s) does meet some of the criteria for this section:

* The Conclusion successfully summarizes the findings from the question(s) asked through the analysis.

**Required**

Almost had it. The reason this section has not met the specifications is that the project has not outlined any limitations within the conclusion section.

**Comments/ Recommendation**

The Limitations subsection can be placed after the conclusion. Limitations are the challenges that were faced during this project or could have been faced in the long run. To give you an idea to address the limitations or challenges you as the student personally faced while doing this project, here are some factors to take into account:

* Was the Missing data too much to handle?
* Was the data sufficient to prove your findings without wrangling much data?
* Was there too much missing data that possibly will skew the analysis
* Were methods difficult to find and implement for this project
* Did you as the student personally find it difficult to achieve certain tasks such as coding steps, writing explanations

These are examples of common limitations, that many persons face while doing this project, but everyone has different experiences explain yours!

REMINDER (ON HOW TO DOCUMENT THE CONCLUSION SECTION)

The proper structure to write a substantial-conclusion is:

1. Summarizing the findings from the questions explored
2. A limitations subsection
3. A reference (OPTIONAL)

**To meet specifications**

* Explain and give justification to the limitations or challenges you personally faced while doing this project.

**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.**

Good work! The following requirements for this section meets most specifications which are:

* Each coding method (i.e.: built-in methods, functions and etc)
* Each exploration for univariate and multivariate outputs.
* Data Wrangling
* A Conclusion

**Required**

However, the report contains visualizations but no reasoning is provided for them after they are being plotted.

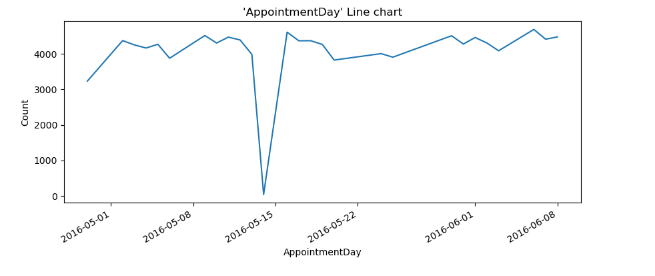
**COMMENTS/ RECOMMENDATION**

The visualizations provided are extremely useful. But there should be some reasoning or explanation given below each visualization. A sentence or two explaining the output of the graph is good enough.  
Remember the graphic visualization provides visualized information about the data, it is just for you to create a worded explanation to make it clear to readers what is really being displayed. Within the worded reasoning you should state numerical values and identify certain points of interest on the graph to enhance the justification and reasoning of the visualization.  
This type of analysis should be done for all of the graphic visualizations provided.

**SUGGESTION**

Before implementing code within the code blocks. Create a Markdown cell and give an explanation of what you are about to do. Then, justify/explain what was done after the execution.  
The main question that always arises is why is code being written and what is it solving?

**An example of what to expect...**

Let us use a visualization from this project and provide reasoning for it:  
[](https://udacity-reviews-uploads.s3.us-west-2.amazonaws.com/_attachments/27924/1676678055/Screenshot_2023-02-17_185353.png)

* *From the plot it can be seen that during the date 2016-05-15, the count was the lowest for appointments. While 2016-06-08 seemed to be the highest*

**JUST ALWAYS REMEMBER THAT...**  
Analyzing data is like writing a story there should be an explanation of why each step is being done. This includes both code cells and visualizations. It is just for you the storyteller to make sure there is a flow to it!  
Try to keep this in mind every time when doing a Data analysis report in fact this is applicable to every project you do within Udacity and the real world!

**To meet specifications**

* Provide Reasoning for each Graphical Visualization provided( The reasoning can be a couple of sentences at minimum)

**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.**

Good start. However, more is still required to meet the critierion.

**Required**

For **ALL** of the visualizations provided within the Notebook/ Report, ensure that they contain the following:

* The x-axis and y-axis are properly labeled in each visualization.
* Each visualization has a title.
* Have a legend if necessary
* All axis data points are readable

Some of the criterion above is missing from some of the visualizations.

**Suggestion**

Here are methods that can help with the requirements and help with code:

* [Setting-up-axes-points](https://plotly.com/python/axes/)
* [Setting-legends](https://plotly.com/python/legend/)
* [Setting-figure-labels](https://plotly.com/python/figure-labels/)
* [Setting-height-width-size-Python](https://plotly.com/python/setting-graph-size/)