Communicate Data Findings (Masterschool)

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Congratulations! 🚀🎯

Detailed and accurate work. Your exploratory part and visualizations show that you have extensively analyzed the data set and produced useful observations and interesting key insights. The slide show presentation was also accurate and well-organized. Properly presenting our data to a more general audience is a key skill. For next steps, I recommend picking new and exciting open data sets from the internet that you would like to explore and follow the same principles. It would be a great practice and it would help you to strengthen your skills.

Good luck with your future professional and educational endeavors!

Code Quality

All code is functional (i.e. no errors are thrown by the code). Warnings are okay, as long as they are not a result of poor coding practices.

All the code is functional and runs without errors. Nice. 🎯

A suggestion about warnings in general:

- Although warnings aren't an immediate problem, it's a good practice to eliminate them from our jupyter/python projects. With the evolution of the language, many of them may become errors in the future so, if we attempt to run our projects after, let's say 6 months, the project may no longer run. To make our code future proof and avoid unexpected glitches, it's highly recommended to fix the warnings as well. Very often, following the warning suggestion is all that's needed.

The project uses functions and loops where possible to reduce repetitive code. Comments and docstrings are used as needed to document code functionality.

The project uses code that is clear and well formatted. There are comments in the code cells and written observations. I would also recommended to encapsulate more of the code cells in to functions. In more advanced projects where we have much more code we usually group the blocks in many methods and classes.

Also, when we encapsulate logically similar blocks of code inside a function we use docstrings:

This is an example of an one-liner docstring

```
def kos_root():
    """Return the pathname of the KOS root directory."""
    global _kos_root
    if _kos_root: return _kos_root
    ...
```

- https://www.python.org/dev/peps/pep-0257/

Check also the following links for tips on organizing python code:

- A few useful tips for organizing your python code
- How avoid code repetition.When we eliminate code repetition the project also becomes more manageable.
- DRY - "Don't Repeat Yourself" principles

Exploratory Data Analysis

If one of the provided datasets is *NOT* used, a csv file of the dataset is included in the submission

The "Ford GoBike" dataset is used.

The project (Parts I alone) contains at least 15 visualizations distributed over univariate, bivariate, and multivariate plots to explore many relationships in the data set. Reasoning is used to justify the flow of the exploration.

You have properly used an exploratory notebook to divide the content in to three sections (`univariate` , `bivariate` and `multivariate`) and have included at least 15 visualizations according to the requirements. As a reader I could easily follow the notebook with the provided reasoning that was used to justify the flow of the exploration. Excellent!

Useful links:

- Check this website that contains a range of plots we can use in our analysis. A good summary of the most used plots. https://python-graph-gallery.com
- Sometimes it's not very clear when to use multivariate, univariate or bivariate plots. Read this article in researchoptimus.com to gain a better grasp.
- This article in the topic of Data exploration presents a collection of very practical techniques that you can immediately use as well as information on how to extract key insights from the data and the observations.
- A collection of 5 very common practices in data exploration from this article which I highly recommend taking a closer look.

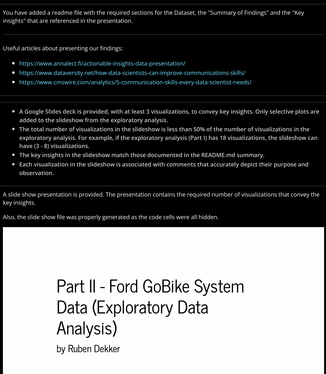
Questions and observations are placed regularly throughout the report, after each plot or set of related plots.

> Tip: Use the ""Question-Visualization-Observations"" framework throughout the exploration.
>
> Tip: For the Part I notebook, use *File > Download as... > HTML or PDF* menu option to generate the HTML/PDF.

The questions and the observations are placed correctly throughout the report.

Also, I recommend to use the "Question-Visualization-Observations" framework throughout your exploration:

1. A question to explore.
2. The actual plot (or group of related plots).
3. The observation(s) from the plot(s).



NOTES

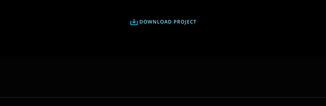
- Many interesting observations were made, like "There are major peaks at 8:00 and 17:00, conventional office hours.",
- Making clear observations in the notebook can help a reader who wants to take a quick look at your work to easily navigate it and judge if it will be useful to them.

"Visualizations made in the project depict the data in an appropriate manner that allows plots to be readily interpreted. This includes choice of appropriate plot type, data encodings, transformations, and formatting (title, axis-labels) as needed."

> Tip: Do not overplot or incorrectly plot ordinal data."

Overall, the visualizations in the project effectively depict the data. The plots are clear and easy to understand. Labels, titles, and legends are used appropriately.

Also, great to see that you're correctly plotting ordinal data:



Explanatory Data Analysis

The README.md must include a summary of main findings that reflects on the steps taken during the data exploration. It should also describes the key insights that are conveyed by the explanatory presentation.

> Tip: The README.md summary is based on the exploration report (Part I notebook) and will guide your explanatory slide deck (Part II notebook) .

You have added a readme file with the required sections for the Dataset, the "Summary of findings" and the "Key insights" that are referenced in the presentation.

Useful articles about presenting our findings:

- https://www.annalect.fi/actionable-insights-data-presentation/
- https://www.dataversity.net/how-data-scientists-can-improve-communications-skills/
- https://www.crmxpress.com/analytics/5-communication-skills-every-data-scientist-needs/

- A Google Slides deck is provided, with at least 3 visualizations, to convey key insights. Only selective plots are added to the slideshow from the exploratory analysis.
- The total number of visualizations in the slideshow is less than 50% of the number of visualizations in the exploratory analysis. For example, if the exploratory analysis (Part I) has 18 visualizations, the slideshow can have (3 - 9) visualizations.
- The key insights in the slideshow match those documented in the README.md summary.
- Each visualization in the slideshow is associated with comments that accurately depict their purpose and observation.

A slide show presentation is provided. The presentation contains the required number of visualizations that convey the key insights.

Also, the slide show file was properly generated as the code cells were all hidden.

All plots in the slideshow are appropriate, meaning the plot type, encodings, and transformations are suitable to the underlying data.

All plots in the slideshow are polished, meaning all plots have a title with labeled axes and legends. Labels include units as needed. In other words, each plot must have - chart title, x/y axis label (with units), x/y ticks, and legend.

All the plots in the presentation have appropriate titles, labeled axes and legends where necessary. Nice. 💫👌

[Suggestions]:

- It would be nice to use fixed plot dimensions for all of your visualizations on the slide show.
  Since most screens have the 16:9 aspect ratio to use the A4 (portrait) page size with the 16/9 aspect ratio we can use:

```
plt.figure(figsize=[14.70, 8.27])
```

  (Place it before the plot call)

  ...

  Also, if the plot has subplots, using:

```
plt.tight_layout()
```

  optimizes the distances between the visualizations.
  Also, if you're using seaborn:

```
sb.set(rc = {'figure.figsize':[14.70,8.27]})
```

```
g.figure.set_size_inches(14.70, 8.27)
```

DOWNLOAD PROJECT