

[Return to Classroom](#)

Communicate Data Findings

REVIEW

CODE REVIEW

HISTORY

Requires Changes

1 specification requires changes

The report asks interesting questions and the analysis reveals important aspects of 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 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.

Code Functionality

Excellent, the code returns what is expected and runs with no errors!

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

You can also examine the python style document here. <https://www.python.org/dev/peps/pep-0008/>

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

Custom Function

It is awesome that you created a custom function that reduces repetitions and simplifies the code. Adding a docstring that occurs as the first statement in a module, function, class, or method definition, becomes the **doc** special attribute of that object.

<https://www.programiz.com/python-programming/docstrings>

```
def LoanStatusRecorder( status ):
    if 'Past Due' in status:
        re_stat = 'Past Due'
    else:
        re_stat = status
    return re_stat

focus_df['LoanStatusRecode'] = focus_df.LoanStatus.apply(lambda x: LoanStatusRecorder(x))
focus_df.LoanStatusRecode.value_counts()
```

Exploratory Data Analysis

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.

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>
- `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.

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 analysis follows a logical flow, the discussion includes reasonings and explanations about the analysis. I really enjoy following the line of your thoughts as you progress with the exploration. It is awesome that you include discussion under the charts, which makes it easier for the readers to follow the analysis.

This rubric is very important since as a data scientist you are expected to communicate your findings and insights from the analysis to colleagues. As the discussion and explanation is more comprehensive you will reach a wider audience,

"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."

I strongly encourage you to show the relevant statistics that quantify the results and insights for each chart. For example,

- When plotting a box plot you can show the mean median and quartiles for each category (box) `".describe()"` <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html> .
- When plotting a scatter plot you can calculate the correlation values <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html>.
- When plotting a bar plot, you can show a table that lists the value for each bar

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) .

The markdown file includes a comprehensive summary of the findings and reflects the analysis done in the report. I strongly encourage you to use bullet points, that will make it easier for the readers to focus on the important items.

For a README file there are some general conventions :

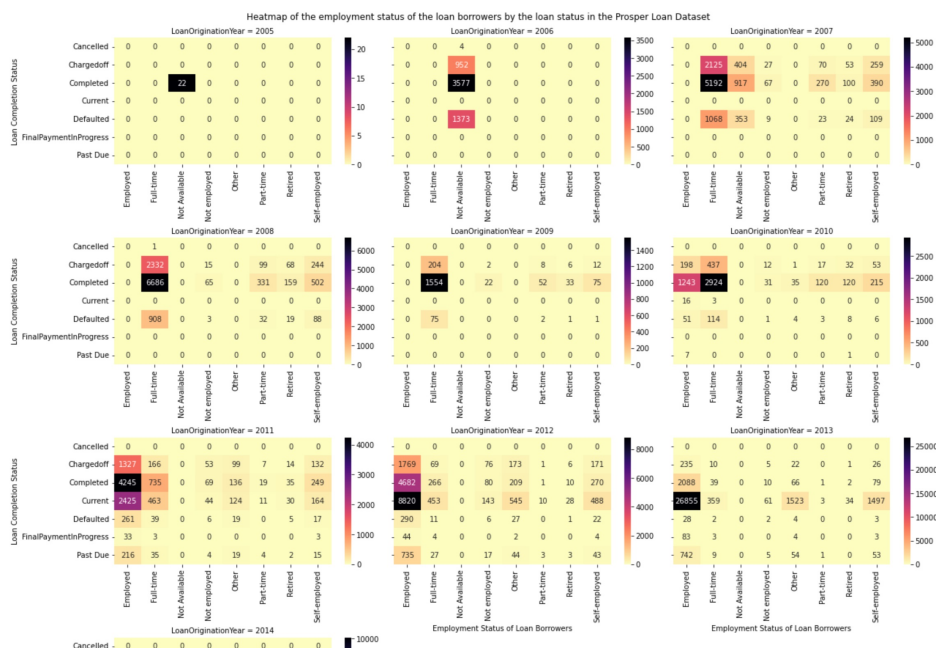
1. General description of the project
2. List of relevant and required software.
3. List of files included in the project.
4. Acknowledgements and credits
5. Bugs

Here is also the GitHub help page on READMEs: <https://help.github.com/articles/about-readmes/>

- A slideshow (HTML file) 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 - 8) 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.

Tip: For Part II notebook, use the `jupyter nbconvert` command to generate the HTML slide show.

Please make sure that the figures fit into the slide and are not larger than the slide.



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.

The slide show is easy to follow and informative! Please consider adding a summary slide in which you list all the findings and explain the shortcomings and limitations of the analysis.

As mentioned above, please consider including more relevant statistics with each figure to quantify the results and support your insights.

 RESUBMIT

 [DOWNLOAD PROJECT](#)



Best practices for your project resubmission

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

 [Watch Video](#) (3:01)

RETURN TO PATH
