

[◀ Return to Classroom](#)

Communicate Data Findings

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Hi Udacity Learner,
Dorothy Kunth.

- Congratulations on completing the project! 🎉
- You have done outstanding work on this project. It was very easy for me to navigate through your work as everything was well explained.

ADDITIONAL LINKS  TO READ IN FREE TIME

- [7 Fundamental Steps to Complete a Data Analytics Project](#)
- [A Comprehensive Guide to Data Exploration](#)

Sure you have learned a lot and we encourage you to keep up with this hard work. Have a nice day and good luck forward. 🙌

What issues did you face in the project?

How long did you take to complete this project?

Any suggestions or ideas you may have on the project.

- Don't forget to rate my work as a project reviewer! Your detailed feedback is very helpful and appreciated - thank you!.
- I'll look forward to reading from you. Thanks a lot! 🙌

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 runs without any error. Good job.

Suggestion.

- There were warnings in the exploration report. Try to fix warnings as they do not come under good coding practice. You can learn about warnings [through this link](#)

```
# convert ProsperRating (Alpha), IncomeRange, and LoanStatus into ordered categorical types
ordinal_var_dict = {'ProsperRating (Alpha)': ['HR', 'E', 'D', 'C', 'B', 'A', 'AA'],
                    'IncomeRange': ['Not displayed', 'Not employed', '$0', '$1-24,999', '$25,000-49,999',
                                     '$50,000-74,999', '$75,000-99,999', '$100,000+'],
                    'LoanStatus': ['Cancelled', 'Defaulted', 'Chargedoff', 'Past Due (>120 days)', 'Past Due (91-120 d
ays)',
                                   'Past Due (61-90 days)', 'Past Due (31-60 days)', 'Past Due (16-30 days)',
                                   'Past Due (1-15 days)', 'FinalPaymentInProgress', 'Current', 'Completed']}

for var in ordinal_var_dict:
    ordered_var = pd.api.types.CategoricalDtype(ordered = True,
                                                categories = ordinal_var_dict[var])
    df_prosper[var] = df_prosper[var].astype(ordered_var)
```

/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>
if sys.path[0] == '':

- You can also ignore the warning

```
# at the top of the file, before other imports
import warnings

warnings.filterwarnings('ignore')

# no warnings will be printed from now on.
```

ADDITIONAL LINKS

- If you want to learn more about python programming you can [follow this link](#)
- [15 Python tips and tricks to master Data Science and Machine Learning](#)
- [The ultimate guide to writing better Python code](#)

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

- Great job doing data wrangling task before the actual exploration.
- Comments and docstrings are used to document the code functionality.
- Function and loops are used to avoid code repetition.

```
# plot matrix of numeric feature LoanOriginalAmount against categorical features.
def boxgrid(x, y, **kwargs):
    """ Quick hack for creating box plots with seaborn's PairGrid. """
    default_color = sb.color_palette()[0]
    sb.boxplot(x = x, y = y, color = default_color)

plt.figure(figsize = [10, 10])
g = sb.PairGrid(data = df_prosper_rating, y_vars = ['LoanOriginalAmount'],
               x_vars = ['ProsperRating (Alpha)', 'EmploymentStatus', 'IncomeRange'], size = 3, aspect = 1.5,)
g.map(boxgrid)
# rotate the x axes for plots 2 and 3 using flat method
for ax in g.axes.flat[1:3]:
    ax.tick_params(axis = 'x', rotation = 45)
plt.show();

# convert ProsperRating (Alpha), IncomeRange, and LoanStatus into ordered categorical types
ordinal_var_dict = {'ProsperRating (Alpha)': ['HR', 'E', 'D', 'C', 'B', 'A', 'AA'],
                    'IncomeRange': ['Not displayed', 'Not employed', '$0', '$1-24,999', '$25,000-49,999',
                                     '$50,000-74,999', '$75,000-99,999', '$100,000+'],
                    'LoanStatus': ['Cancelled', 'Defaulted', 'Chargedoff', 'Past Due (>120 days)', 'Past Due (91-120 d
ays)',
                                   'Past Due (61-90 days)', 'Past Due (31-60 days)', 'Past Due (16-30 days)',
                                   'Past Due (1-15 days)', 'FinalPaymentInProgress', 'Current', 'Completed']}

for var in ordinal_var_dict:
    ordered_var = pd.api.types.CategoricalDtype(ordered = True,
                                                categories = ordinal_var_dict[var])
    df_prosper[var] = df_prosper[var].astype(ordered_var)
```

ADDITIONAL LINKS WHICH YOU CAN READ IN YOUR FREE TIME.

- [Ten Good Coding Practices for Data Scientists](#)
- [Six steps to more professional data science code](#)
- [Good coding practices - Describing your code](#)

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.

- Great job defining three different sections in your exploration report for univariate, bivariate, and multivariate plots. The flow of your exploration was very smooth as you started from univariate and gradually moved on to bivariate and multivariate plots. Also, proper reasoning was used to justify the flow of the exploration.
- Also, as per the rubrics requirement, there were more than 15 visualizations distributed among the three sections.

ADDITIONAL LINKS

[How to use Python Seaborn for Exploratory Data Analysis](#)

[Univariate, Bivariate, and Multivariate](#)

[This link contains](#) some in-depth information about Matplotlib.

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.

- great job placing the question in between your exploration. it really helps in going through the flow of the exploration. I was able to follow along, and understand what was going on in your head while you were trying to answer those questions!

- You have added observation after the plots which really helped.

Great job framing question and answering them based on findings

Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

- For the borrower rate, most loans have borrower rate between 5 and 40%. After adjusting the bins, It looks unimodal with high peak between 15-16% then there is also a very high peak at 31-32%.

Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

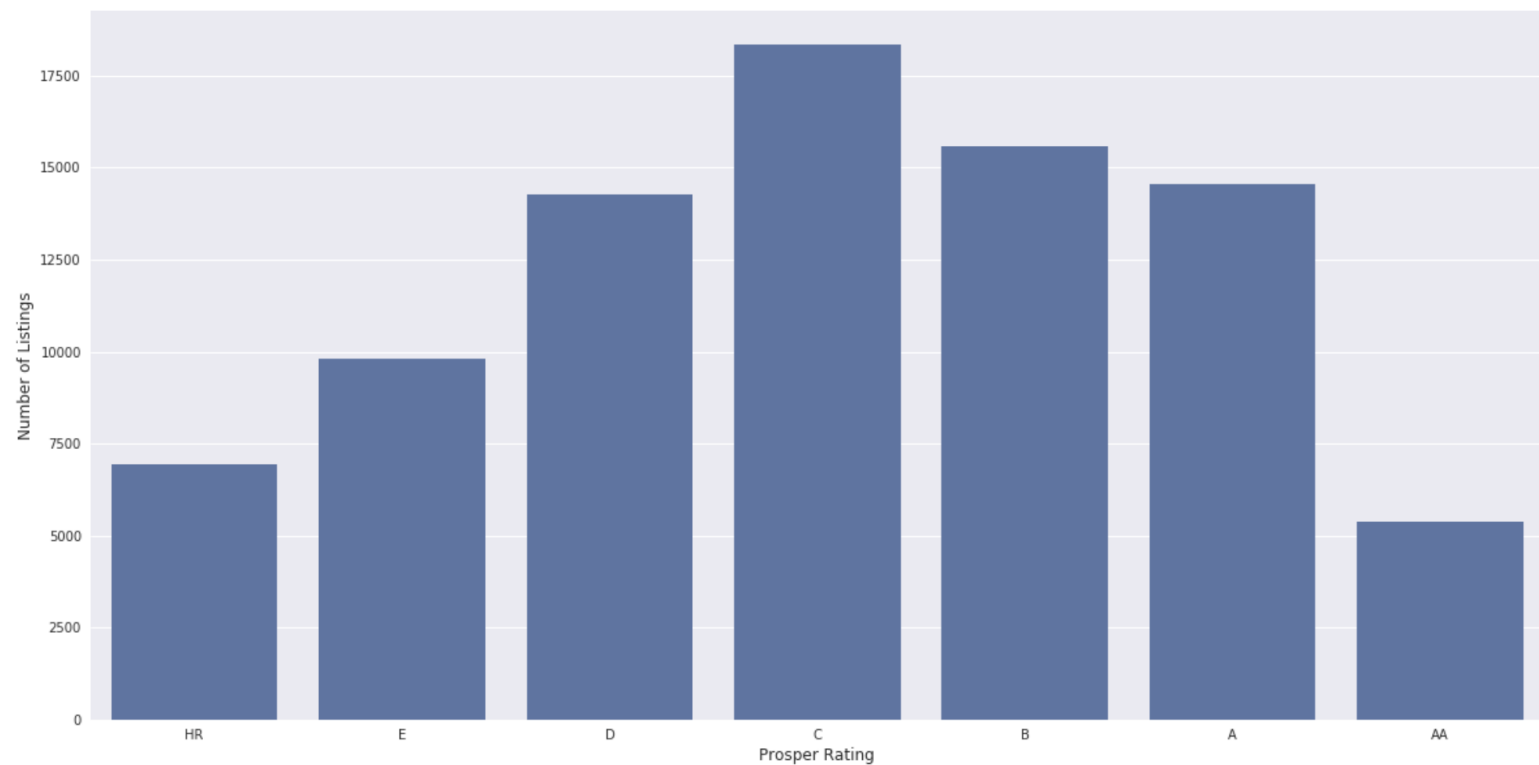
- For the loan amount, after adjusting the bins, it showed that the top 3 loan amounts are USD 4,000.00, USD 15,000.00 and USD 10,000.00 respectively.
- For the Prosper rating distribution, I needed to plot the distribution of data with only non-null values in Prosper Rating column only. With 29084 null values, it would just distort the analysis and provide poor visualization.
- For the distribution of loans based on origination year, I created a new column LoanOriginationYear to store the extracted year from loan origination date.
- There is a significant number of loans which has the lowest rating HR (high risk) which I find unusual. If I am the investor I would definitely not invest in loans listed as high risk rating.
- A little above 1% of the loans are taken by the borrowers who are unemployed and with income range of USD 0.
- It's surprising to see that about a hundred of loans are delinquent in the amounts of over USD 100,000.00

"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."

- You have included a nice selection of univariate, bivariate, and multivariate plots.
- Your aesthetic and labeling choices have made the plots readily interpretable.

- ordinal available like ProsperRating is correctly ordered.



Prosper Rating is correctly ordered

ADDITIONAL LINK

- [Fundamentals of Data Visualization](#)
- [How to avoid overplotting with python](#)

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) .

- Great job summarizing your project in the readme file.
- You have included a summary of the main findings and key insights.

ADDITIONAL LINKS

- [You can go through this link](#) to know the answers to the questions.

Why is a README File necessary?


What is a README File?


How to write a good README File?


How to make your README file more interesting?

- 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.

 Great job, submitting the slideshow. The format of your slideshow is correct and it meets all the rubrics requirements.

A slideshow is provided. 

At least three visualizations are used in the presentation to convey key insights. 

Each visualization is associated with comments that accurately depict their purpose. ✓

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 are well polished that is they have an appropriate title with labeled axes and legends.

WHY PLOTS SHOULD BE WELL POLISHED?

- A data visualization's purpose is to convey information and make a point. To reliably achieve this goal when preparing visualizations, we have to place the data into context and provide accompanying titles, captions, and other annotations.

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

Rate this review

START