UDACITY

Return to Classroom

# Communicate Data Findings

REVIEW

HISTORY

## Meets Specifications

## Congratulations 🎓

You have met all of the requirements for this project! This is an excellent submission.

- The commentary was engaging (and very, very detailed).
- The analysis was focused and thorough.
- You take a complex dataset and reveal some interesting, and unusual, patterns *(for example: "Term showed the most variation as a third variable when applied to Rate to APR scatterplot. In this case, the 60 month loans were almost completely linear and the 36 month loans were incredibly varied.")*.

Very nice work!!

For data analysts, one of the most difficult parts of the job is explaining the implications of their studies to non-statisticians. With your choice of visualizations, and your data wrangling to generate those visualizations, you have highlighted that you have the skill to do this with ease. It is a skill that translates to any *data exploration*. You should look forward with confidence to applying what you have learned here, to any of the interesting data analyses that you will face in your career.

*(I have made some suggestions below. These suggestions don't detract from your work, they are just included so that you can add the finishing touches to your excellent report).*

Congrats again and best wishes for your next project (or your next endeavor, if this is your last project)!!

p.s. If you are interested in developing your skills in data analysis, this free text is a valuable resource (the link is to the authors github version of the text) *(**Note:** The first 4 chapters complement the contents of this course (they*

the authors github version of the text) *(**Note:** The first 4 chapters complement the contents of this course (they begin with concepts that you have learned, and then delve a little deeper to areas that could not be covered in this course). Later chapters involve more advanced areas of analysis - they will give you a good introduction to some of the directions that data analysis/science can take)*

## 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.**

The code in both of your notebooks evaluate as expected (without errors). Nice work!

ADDITIONAL NOTES

- To help develop your skills using python, I highly recommend working through the examples in this free online text

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

- pandas is used for most data wrangling tasks, so vectorized operations rather than loops are used. Nice work.
- The comments in your code, and in your Markdown that relate to the code, makes it easy to follow your code. Very nice work!
- Excellent work using a function to avoid repetitive coding

ADDITIONAL NOTES

- In data analysis, in a work environment, commenting code, so that your colleagues understand the intent of the code that you write, is a basic necessity.
    - *(It is always more difficult to read other people's code).*
    - Commenting code is a good habit to develop, because it communicates to colleagues, or to yourself at some future time, the intent of the code that you have written (in a succinct way that avoids you having to examine the code line by line). It is difficult to overstate the importance of clear code commenting, in a work environment, in data analysis.
    - Here is a summary of good commenting etiquette

Again, nice work!!

- If you want to explore functions in python:
  - This free online text/tutorial will help you to develop this area of your coding skills
  - Here is a simple overview of functions

## Exploratory Data Analysis

**The project appropriately uses univariate, bivariate, and multivariate plots to explore many relationships in the data set. Reasoning is used to justify the flow of the exploration.**

Your plot selection is excellent (as are your aesthetic choices for those plots). Also, your data wrangling to generate those plots is exceptional (which helps capture patterns that would otherwise have not been so evident).

You have a wide range of visualizations to choose from for your slide deck/show. Very nice work!

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

This is the most detailed commentary that I have seen for this project. Kudos!

Your commentary makes it easy to follow your thought processes as you work through your data exploration. It is clear, well organized and engaging. The questions that you answer throughout your report uncovers interesting patterns in the data. Excellent work!

---

**FYI**: This information is not available in the data description:

> "On November 24, 2008, the SEC found Prosper to be in violation of the Securities Act of 1933. As a result of these findings, the SEC imposed a cease and desist order on Prosper ... In July 2009, Prosper reopened their website for lending ("investing") and borrowing after having obtained SEC registration for its loans ("notes"). After the relaunch, bidding on loans was restricted to residents of 28 U.S. states and the District of Columbia. Borrowers may reside in any of 47 states, with residents of three states (Iowa, Maine, and North Dakota) not permitted to borrow through Prosper".

Source: Wikipedia

**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 labels as needed.**

As noted above, you include an excellent selection of visualizations. However, there are some issues that, while they don't violate any of the project rubric items, should be addressed.

---

TIPS

---

SAMPLE SIZE

For some of your visualizations a sample size of 350 is used.

For some of those plots, multiple categories are grouped by.

If, say, there are 10 categories, that will be, on average 35 observations per category.

For data this complete, use at least 10000 observations in your visualizations *(there will be very little effect on runtime and, using the correct visualizations, only positive effects on your visualizations)*

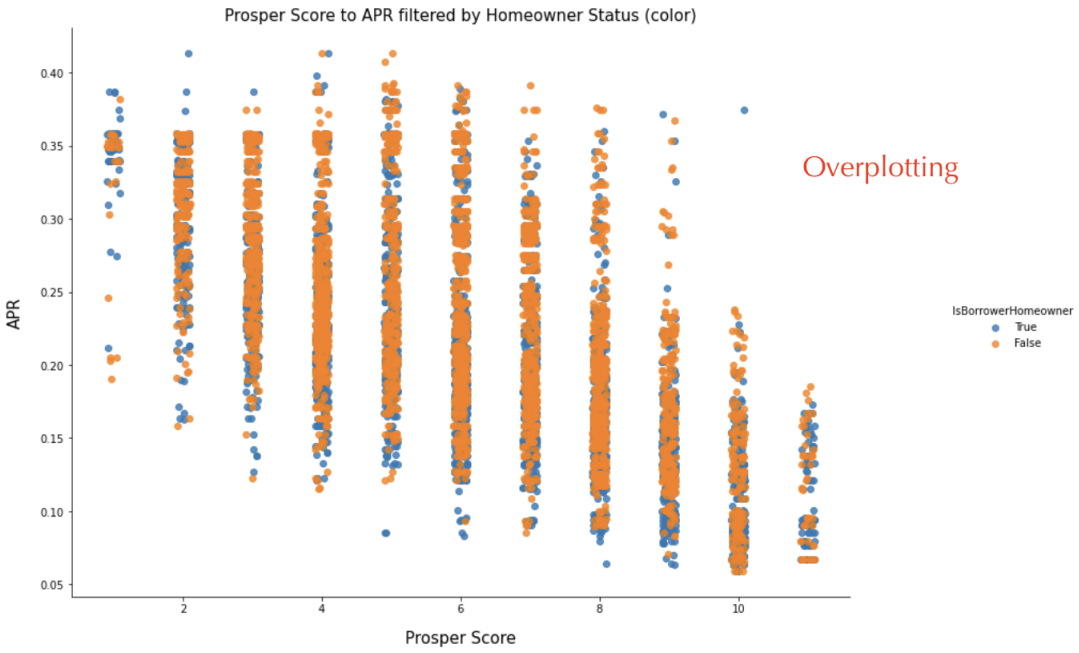
---

OVERPLOTTING

Many of your plots are overplotted.

Seaborn has a method `.stripplot()` that allows you to overcome this issue:

```
plt.ylabel("APR", fontsize=15, labelpad=15);
```




```
In [172]:
"""Graph a Scatterplot with 3rd Variable Added in Seaborn"""

# testing for a Qualitative nominal 3rd variable "...Homeowner" using color

# random sample of 350
#loan_data_subset_1 = loan_data.sample(350, random_state=42)

g = sb.FacetGrid(data = loan_data_subset_1,
                 height = 7, # This increases plot height size
                 aspect = 1.0 # increases width 50% longer than it is tall
                );

```

```
g.map(sb.stripplot, 'ProsperScore','BorrowerAPR', 'IsBorrowerHomeowner',
      hue_order = [True, False],
      jitter=0.35, dodge=True);
g.add_legend()  #adds legend to plot
## Set Plot Dimensions - FIGURE LEVEL
g.fig.set_size_inches(14.70, 8.27);

plt.title("Prosper Score to APR filtered by Homeowner Status (color)", fontsize=15)
plt.xlabel("Prosper Score", fontsize=15, labelpad=15)
plt.ylabel("APR", fontsize=15, labelpad=15);
```


*Click On Images To Enlarge Them*

---

OUTLIERS

For data with outliers, include two versions of the visualizations:

1. With all of the data
2. Excluding outliers

*(which you did for some of your visualization)*

For example, here is a sign of an outlier:

```
In [146]: loan_data_subset_4.DebtToIncomeRatio.describe()
```

```
count    914.000000
mean       0.269972
std        0.466360
min        0.010000
25%        0.140000
50%        0.215000
75%        0.317500
max       10.010000
Name: DebtToIncomeRatio, dtype: float64
```

Let's take a closer look:

```
In [147]: 1 loan_data_subset_4.DebtToIncomeRatio.describe(np.arange(0,1.05,0.05))
```

```
count    914.000000
mean       0.269972
std        0.466360
min        0.010000
0%         0.010000
5%         0.056500
10%        0.080000
15%        0.100000
20%        0.120000
25%        0.140000
30%        0.160000
35%        0.170000
40%        0.190000
45%        0.200000
50%        0.215000
55%        0.230000
60%        0.250000
65%        0.270000
70%        0.290000
75%        0.317500
80%        0.350000
85%        0.380000
90%        0.420000
95%        0.500000
100%      10.010000
max       10.010000
Name: DebtToIncomeRatio, dtype: float64
```

And again, a closer look:

```
[176]: 1 loan_data_subset_4.DebtToIncomeRatio.describe(np.arange(0.9,1.01,0.01))
```

```
count    9262.000000
mean        0.285242
std         0.610444
min         0.000000
50%         0.220000
90%         0.420000
91%         0.430000
92%         0.450000
93%         0.470000
94%         0.490000
95%         0.510000
96%         0.540000
97%         0.580000
98%         0.650000
99%         0.983900
100%       10.010000
max        10.010000
Name: DebtToIncomeRatio, dtype: float64
```

So, above the 99th percentile, there are extreme observations. Even above the 95th percentile, there are large

jumps.

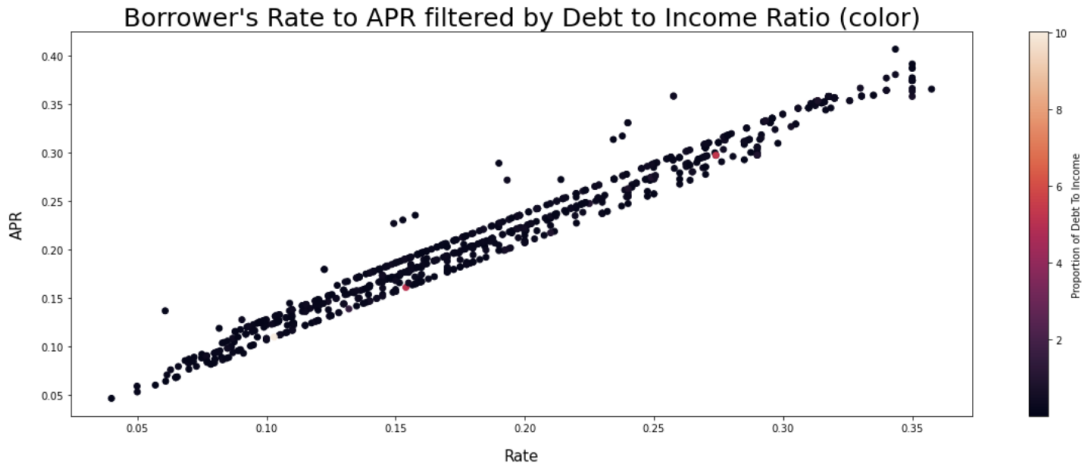
We can take a look at:

1. All of the data
2. Data below the 99th percentile
3. Data below the 95th percentile

for debt to income ratios

```
plt.scatter(data = loan_data_subset_4, x = 'BorrowerRate', y = 'BorrowerAPR',
                c = 'DebtToIncomeRatio', #this is the 3rd variable what will get colored
                                #you lose the height and aspect ability though
                cmap = 'rocket'  #this will set a sequential color palette
                                # 'rocket' or 'mako' too. default 'viridis_r'
               );

plt.colorbar(label = 'Proportion of Debt To Income') #adds color and legend  to 3rd variable
plt.title("Borrower's Rate to APR filtered by Debt to Income Ratio (color)", fontsize=25)
plt.xlabel("Rate", fontsize=15, labelpad=15) #offsets label
plt.ylabel("APR", fontsize=15, labelpad=15); #offsets label
```

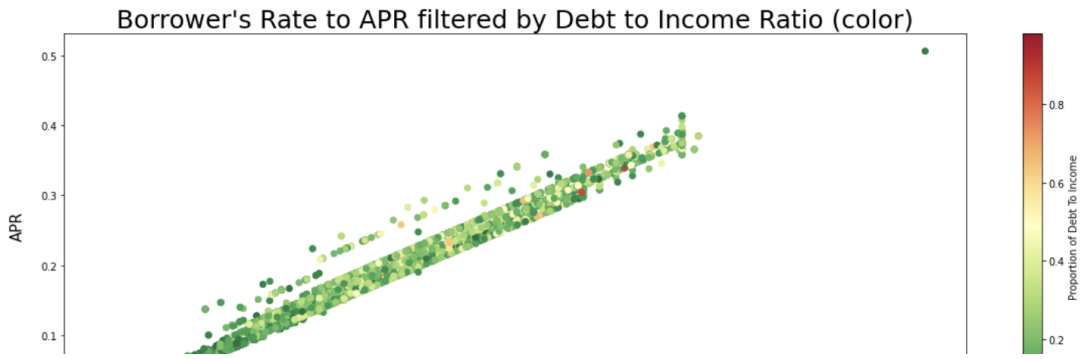


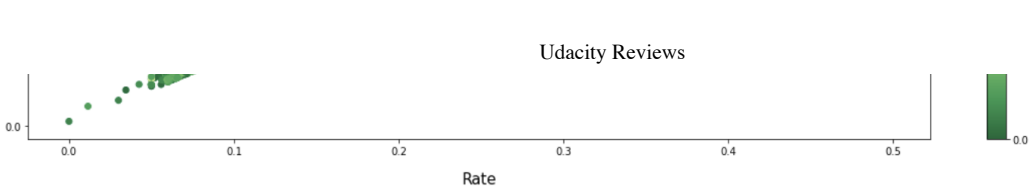

[174]:

```
"""Graph a Scatterplot (with 3rd Variable Added) in Matplotlib"""

# testing for color palette with a Quantitative 3rd variable "...DebtToIncomeRatio"
# use matplotlib scatterplot instead of seaborn's FacetGrid
# set a larger figure size for subplots (width, height (inches); default 6.4, 4.8)
plt.figure(figsize = [20, 7])
plt.scatter(data = loan_data_subset_4.query("DebtToIncomeRatio < DebtToIncomeRatio.quantile(0.99)") ,
            x = 'BorrowerRate', y = 'BorrowerAPR',
                c = 'DebtToIncomeRatio', #this is the 3rd variable what will get colored
                                #you lose the height and aspect ability though
                cmap = 'RdYlGn_r'  #this will set a sequential color palette
                                # 'rocket' or 'mako' too. default 'viridis_r'
               );

plt.colorbar(label = 'Proportion of Debt To Income') #adds color and legend  to 3rd variable
plt.title("Borrower's Rate to APR filtered by Debt to Income Ratio (color)", fontsize=25)
plt.xlabel("Rate", fontsize=15, labelpad=15) #offsets label
plt.ylabel("APR", fontsize=15, labelpad=15); #offsets label
```



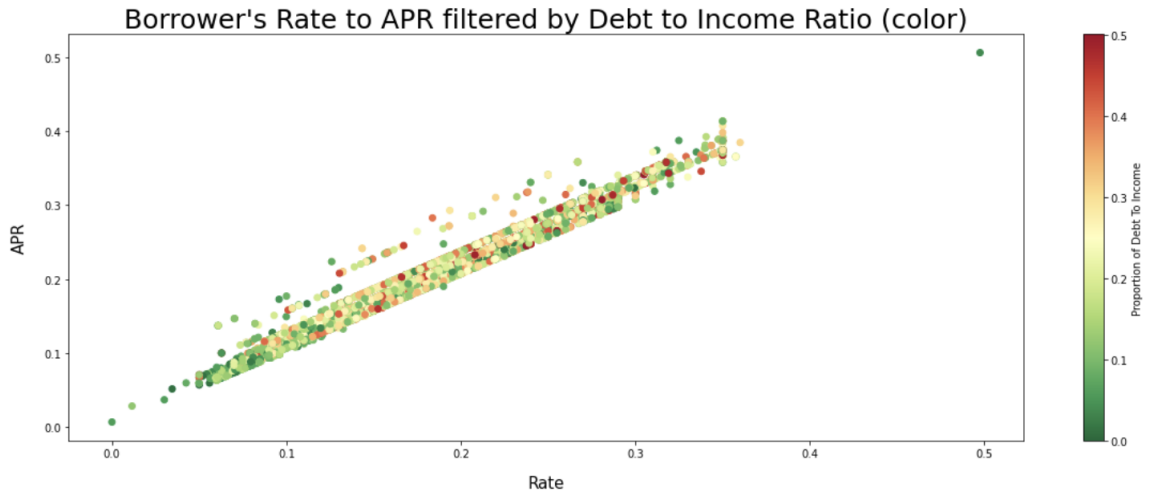



AND

```
In [175]:
"""Graph a Scatterplot (with 3rd Variable Added) in Matplotlib"""

# testing for color palette with a Quantitative 3rd variable "...DebtToIncomeRatio"
# use matplotlib scatterplot instead of seaborn's FacetGrid
# set a larger figure size for subplots (width, height (inches); default 6.4, 4.8)
plt.figure(figsize = [20, 7])
plt.scatter(data = loan_data_subset_4.query("DebtToIncomeRatio < DebtToIncomeRatio.quantile(0.95)") ,
            x = 'BorrowerRate', y = 'BorrowerAPR',
                 c = 'DebtToIncomeRatio', #this is the 3rd variable what will get colored
                                          #you lose the height and aspect ability though
                 cmap = 'RdYlGn_r'  #this will set a sequential color palette
                                    # 'rocket' or 'mako' too. default 'viridis_r'
                );

plt.colorbar(label = 'Proportion of Debt To Income') #adds color and legend  to 3rd variable
plt.title("Borrower's Rate to APR filtered by Debt to Income Ratio (color)", fontsize=25)
plt.xlabel("Rate", fontsize=15, labelpad=15) #offsets label
plt.ylabel("APR", fontsize=15, labelpad=15); #offsets label
```




REGRESSION LINES

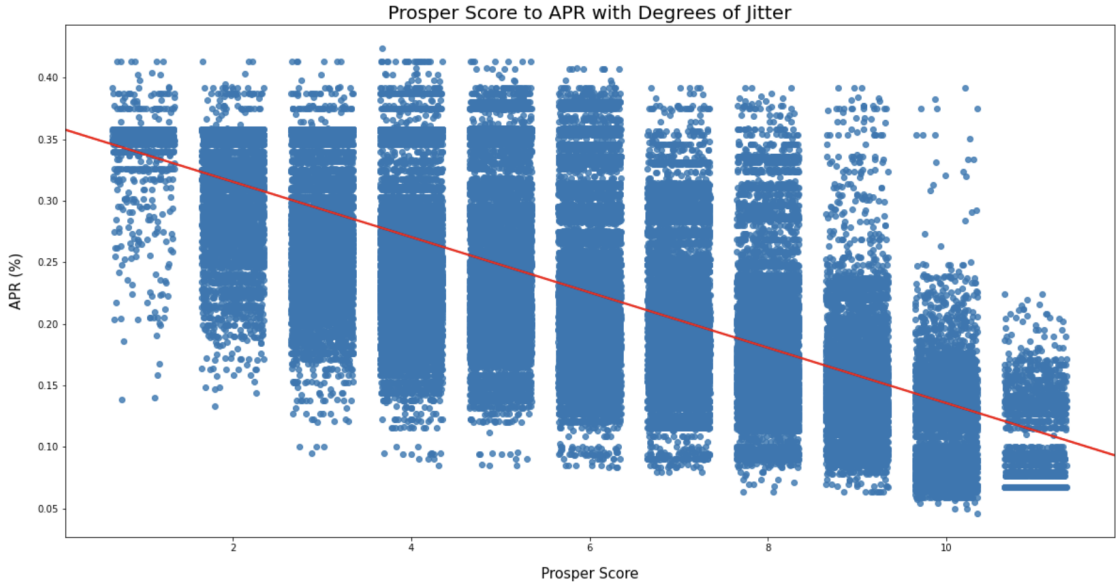
In Seaborn, you can color the regression line use line keywords: `line_kws={"color":"red"}`

```
"""Graph a Scatterplot in Seaborn"""

# In seaborn with a regressionline

plt.figure(figsize = [20, 10])
# In seaborn with a regressionline
sb.regplot(data = loan_data, x = 'ProsperScore', y = 'BorrowerAPR',
           truncate=False, x_jitter=0.35, line_kws={"color":"red"}
           );

plt.title("Prosper Score to APR with Degrees of Jitter", fontsize=20)
plt.xlabel("Prosper Score", fontsize=15, labelpad=15)
plt.ylabel("APR (%)", fontsize=15, labelpad=15);
#plt.figure(figsize = [20, 10])
```




## Explanatory Data Analysis

**A section in the submitted materials includes a summary of main findings that reflects on the steps taken during the data exploration. The section also describes the key insights that are conveyed by the explanatory presentation.**

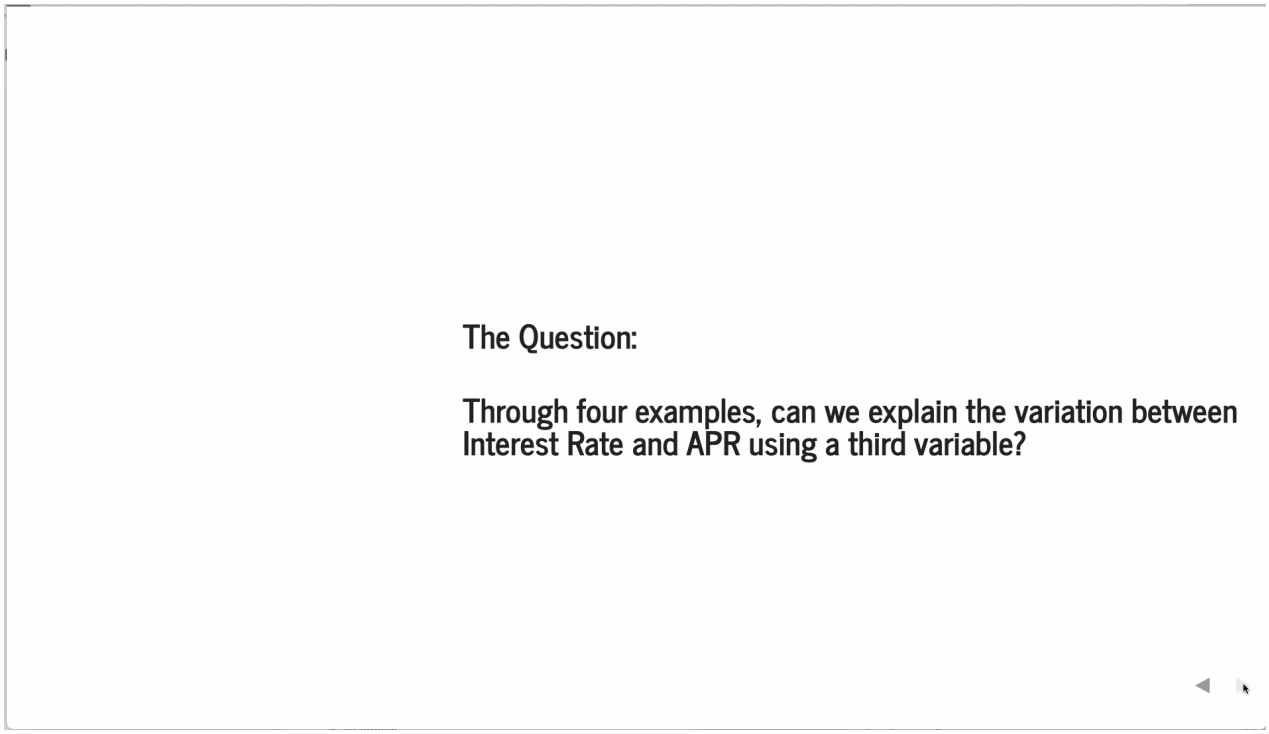
You summarize the contents of your project files in your `readme.md` file.

After reading the readme file, the reader should have a clear sense of what to expect in your *data exploration* and slide deck/show. Your summary is excellent is this regard. Very nice work!

**A slideshow is provided, with at least three visualizations used in the presentation to convey key insights. These key insights match those documented in the summary. Each visualization is associated with**

**comments that accurately depict their purpose.**

You include a slide show notebook and a slide show *HTML* file in your submission.




Your plot selection, and commentary, are excellent. Very nice work!

**All plots in the presentation have an appropriate title with labeled axes and legends. Labels include units as needed. Plot type, encodings, and transformations are all appropriate.**

All plots are titled, and axes are labeled (and units of measurement are included). Nice work!

---

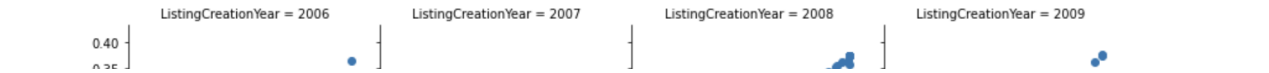
TIP

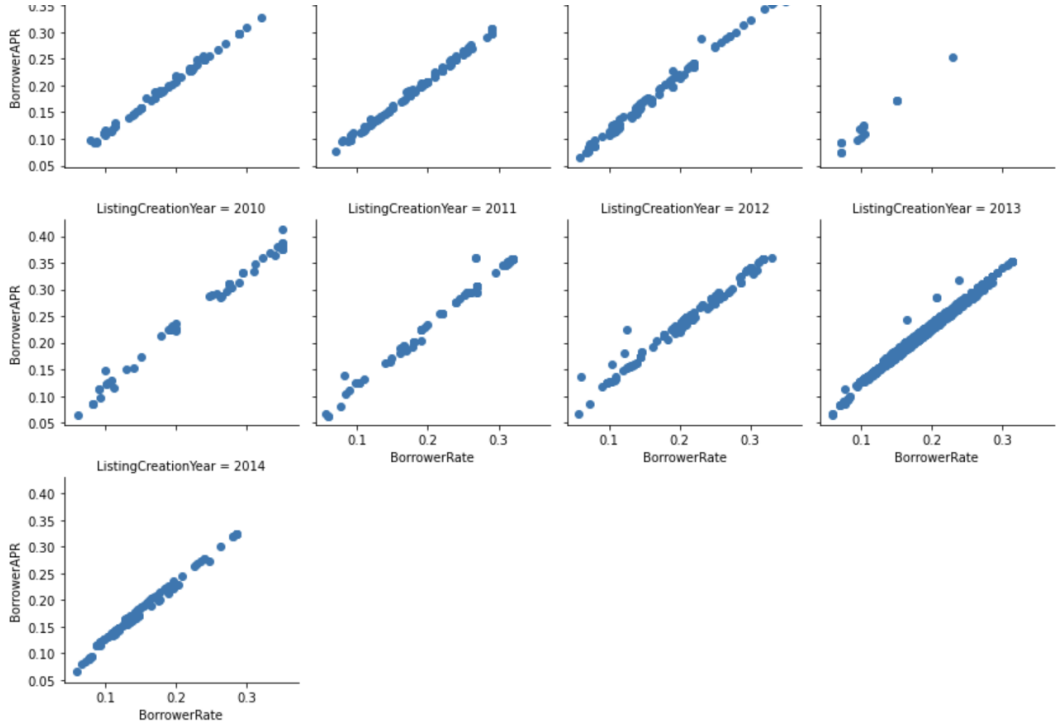
---

For visualizations with subplots, always include a main title, you can use `plt.suptitle()` to add a main title to these plots.

Also, for Seaborn figure level plots, there are specific options that control the plot's text elements - which are explained in code comments here:

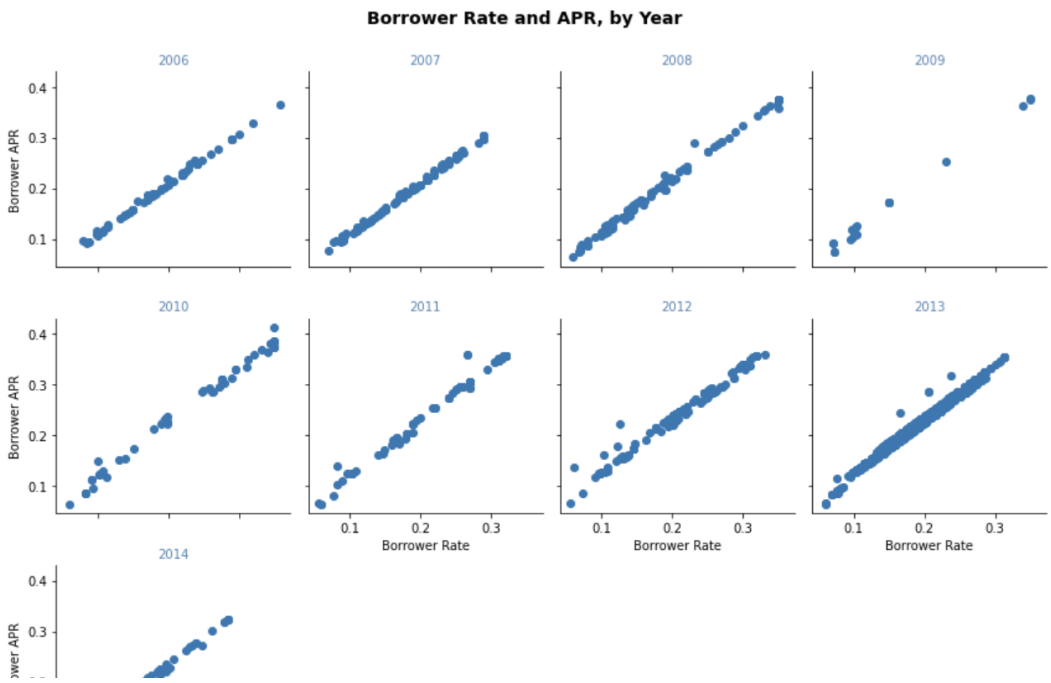
```
141]:
"""Graph Faceted Scatterplots in Seaborn"""
g = sb.FacetGrid(data = loan_data_subset_5, col = 'ListingCreationYear', col_wrap =4, height =3)
g.map(plt.scatter, 'BorrowerRate', 'BorrowerAPR');
```






140]:
```
"""Graph Faceted Scatterplots in Seaborn"""
g = sb.FacetGrid(data = loan_data_subset_5, col = 'ListingCreationYear', col_wrap =4, height =3)
g.map(plt.scatter, 'BorrowerRate', 'BorrowerAPR');

# Add Main Title
plt.suptitle("Borrower Rate and APR, by Year", y = 1,
             fontsize = 14, weight = "bold")
# Include legible axes labels
g.set_axis_labels(x_var="Borrower Rate", y_var="Borrower APR")
# only include categories as plot titles
g.set_titles('{col_name}', color='steelblue');
# optimize distance between subplots
plt.tight_layout();
```




DOWNLOAD PROJECT

RETURN TO PATH