File Edit View Run Kernel Tabs Settings Help

picture_perfect.ipynb

Code Python 3

Picture Perfect

Perfect your visualizations by adding in some finishing touches. Use the hvPlot options and customization attributes to change the look and feel of each visualization, paying particular attention to the legibility of axis labels and the availability of titles and labels.

[1]:
```python
import pandas as pd
import hvplot.pandas
from pathlib import Path
```

Prep the data

[2]:
```python
# Read in loan data
loan_data = pd.read_csv(Path("../../Resources/state_loan_data.csv"))

# Group data by state for state-level analysis
loan_data = loan_data.groupby('State Code').sum()
loan_data.head()
```

[2]:

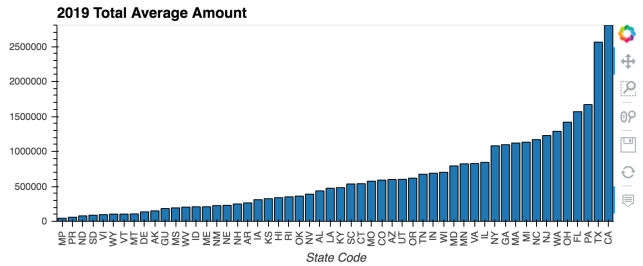
State Code	Total Active Loans	Total Average Loan Amount	2015 - 2016	2010 - 2014	Self Help Loan	Leveraged Loan
AK	897	148811.39	189583.49	172371.39	96	450
AL	7455	435054.53	686066.79	648663.44	18	605
AR	6186	263212.27	355445.70	351562.42	861	567
AZ	3982	597591.94	507532.91	599688.64	1718	1057
CA	8875	2805653.85	3008866.40	2881741.23	4134	2231

Rotate x axis labels and format y axis values using yformatter

[3]:
```python
# Slice for Total Average Loan Amount
loan_data_state = loan_data["Total Average Loan Amount"]
loan_data_avg_grp = loan_data_state.sort_values()

# Use rot and yformatter attributes/options
plot_state_avgs = loan_data_avg_grp.hvplot.bar(
    label="2019 Total Average Amount", rot=90
).opts(yformatter="%.0f")
plot_state_avgs
```

[3]:

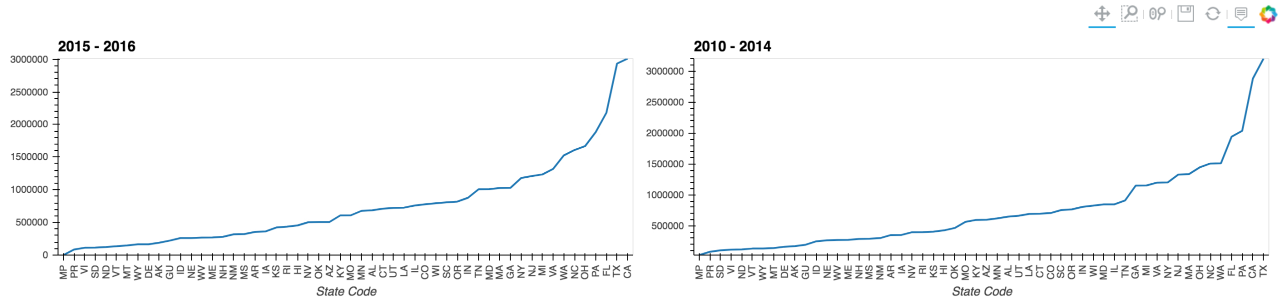


[4]:
```python
# Slice data for Total Average Loan Amount by 2015-2016 and 2010-2014 date ranges
loan_data_range_1 = loan_data["2015 - 2016"]
loan_data_range_2 = loan_data["2010 - 2014"]
loan_data_range_grp = loan_data_range_1.sort_values()
loan_data_range_grp_2 = loan_data_range_2.sort_values()

# Use rot argument and yformatter option to customize formatting
plot_2015_2016 = loan_data_range_grp.hvplot(label="2015 - 2016", rot=90).opts(
    yformatter="%.0f"
)
plot_2010_2014 = loan_data_range_grp_2.hvplot(label="2010 - 2014", rot=90).opts(
    yformatter="%.0f"
)

plot_2015_2016 + plot_2010_2014
```

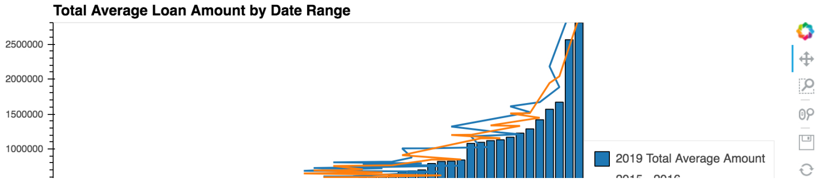
[4]:

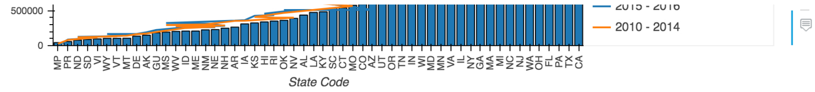


Use title option to customize title

[5]:
```python
# Use opts function to customize title and width
(plot_state_avgs * plot_2015_2016 * plot_2010_2014).opts(
    title="Total Average Loan Amount by Date Range", width=900
)
```

[5]:

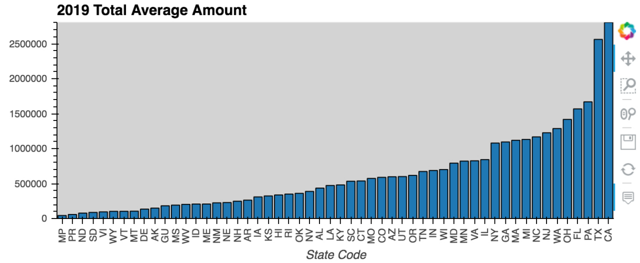




Use the bgcolor option to change background color

[6]:
```
# Use bgcolor option to change plot background
plot_state_avgs.opts(bgcolor="lightgray")
```

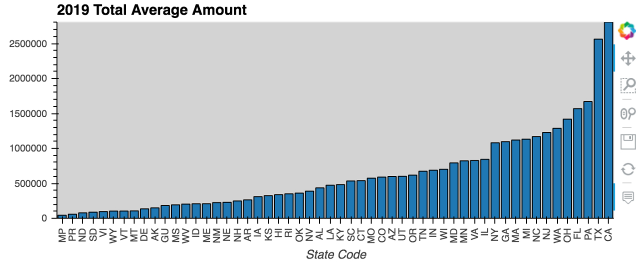
[6]:



Use the hover_line_color option to change data point color when hovering

[7]:
```
# Add color changing hover effect wit hover_line_color option
plot_state_avgs.opts(hover_line_color="red")
```

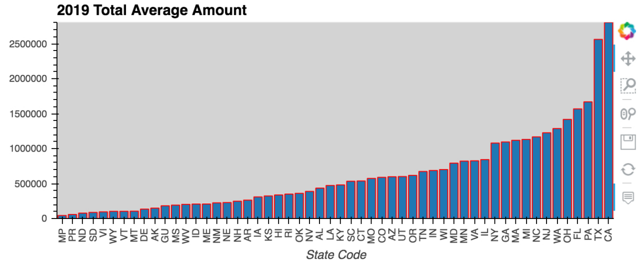
[7]:



Use the line_color option to change line color or bar outline color

[8]:
```
# Customize bar chart with bar outline color
plot_state_avgs.opts(line_color='red')
```

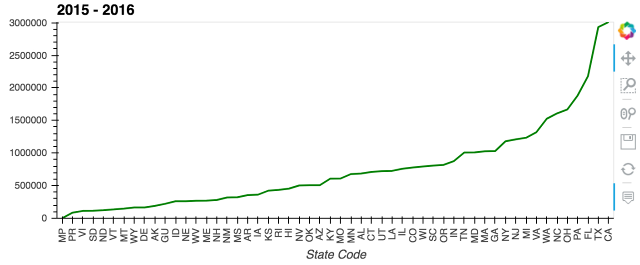
[8]:



Change color of line chart to green

[9]:
```
# Customize line chart line color using line_color option
plot_2015_2016.opts(line_color='green')
```

[9]:


MP PR VI SD ND VT MT WY DE AK GU NE WV NH NM MS AR IA KS HI RI NV OK AZ KY MO AL CT LA IL UT CO WI SC OR TN MD MA NY GA NJ MI WA VA NC OH FL PA TX CA
State Code