Time-Series Visualization using bokeh

Bingyi Li

```
In [1]: import pandas as pd

        from bokeh.layouts import gridplot
        from bokeh.plotting import figure, show, output_file
        from bokeh.palettes import YlOrRd
        from bokeh.transform import factor_cmap
        from bokeh.io import output_notebook
        from bokeh.models import (
            ColumnDataSource,
            HoverTool,
            LinearColorMapper,
            BasicTicker,
            PrintfTickFormatter,
            ColorBar,
        )
```

```
In [2]: output_notebook()
```

BokehJS 0.12.15 successfully loaded.
(https://bokeh.pydata.org)

```
In [3]: data = pd.read_csv('/Users/libingyi/Documents/MSAN/MSAN622/data/Monthly_Property_Crime_2005_to_2015.csv', par
        se_dates=['Date'])
        data['Year'] = data['Date'].dt.year
        data['Month'] = data['Date'].dt.month
        data.head()
```

Out[3]:

	Date	Category	IncidntNum	Year	Month
0	2014-02-01	BURGLARY	506	2014	2
1	2007-02-01	VANDALISM	531	2007	2
2	2012-07-01	BURGLARY	522	2012	7
3	2013-07-01	LARCENY/THEFT	3318	2013	7
4	2010-08-01	VANDALISM	694	2010	8

This dataset contains information of monthly property crimes in San Francisco from 2005 to 2015. First, one can use a heatmap to get an overall sense of the incidences.

```
In [4]:  # Heatmap
         data_h = data.groupby(['Year','Month'])['IncidntNum'].sum().reset_index()
         data_h['Year'] = data_h['Year'].astype(str)
         data_h['Month'] = data_h['Month'].astype(str)

         years = list(data_h.Year.unique())
         months = list(data_h.Month.unique())

         colors = YlOrRd[9]
         mapper = LinearColorMapper(palette=YlOrRd[9][::-1], low=data_h.IncidntNum.min(), high=data_h.IncidntNum.max
         ())

         source = ColumnDataSource(data_h)

         TOOLS = "hover,save,pan,box_zoom,reset,wheel_zoom"

         p = figure(title="San Francisco Property Crime ({0} - {1})".format(years[0], years[-1]),
                    x_range=years, y_range=list(reversed(months)),
                    x_axis_location="above", plot_width=900, plot_height=400,
                    tools=TOOLS, toolbar_location='below')

         p.axis.axis_line_color = None
         p.axis.major_tick_line_color = None
         p.axis.major_label_text_font_size = "5pt"
         p.axis.major_label_standoff = 0
         p.xaxis.major_label_orientation = 1.2

         p.rect(x="Year", y="Month", width=1, height=1,
                source=source,
                fill_color={'field': 'IncidntNum', 'transform': mapper},
                line_color=None)

         color_bar = ColorBar(color_mapper=mapper, major_label_text_font_size="5pt",
                              ticker=BasicTicker(desired_num_ticks=len(colors)),
                              formatter=PrintfTickFormatter(format="%d%%"),
                              label_standoff=6, border_line_color=None, location=(0, 0))
         p.add_layout(color_bar, 'right')

         p.select_one(HoverTool).tooltips = [
             ('date', '@Month @Year'),
             ('Number of Incident', '@IncidntNum'),
         ]

         show(p)
```
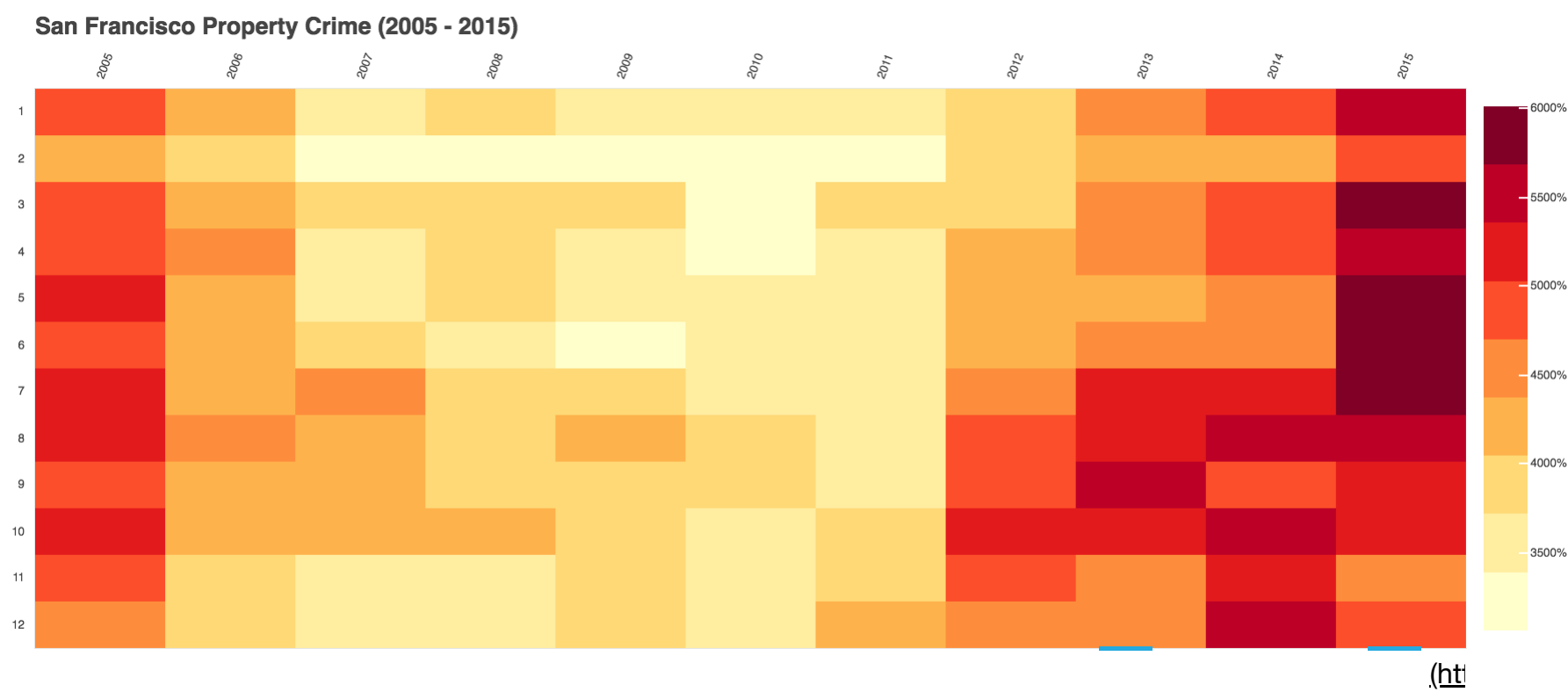

According to the heatmap, it is clear to see that the crime rate is at the lowest point in around 2010, and reaches the highest point in May, June and July of 2015.

Then one may be interested in the overall trend of crime rate in San Francisco. A line chart can be used to visualize this.

```
In [5]:  # Line Chart
         sum_data = pd.DataFrame(data.groupby('Date')['IncidntNum'].sum()).reset_index()
         p1 = figure(x_axis_type="datetime", title="Monthly Property Crime", plot_width=800,plot_height=600)
         p1.grid.grid_line_alpha=0.3
         p1.xaxis.axis_label = 'Date'
         p1.yaxis.axis_label = 'Number of Crime'

         p1.line(sum_data['Date'], sum_data['IncidntNum'], color='#A6CEE3')

         show(p1)
```

Monthly Property Crime
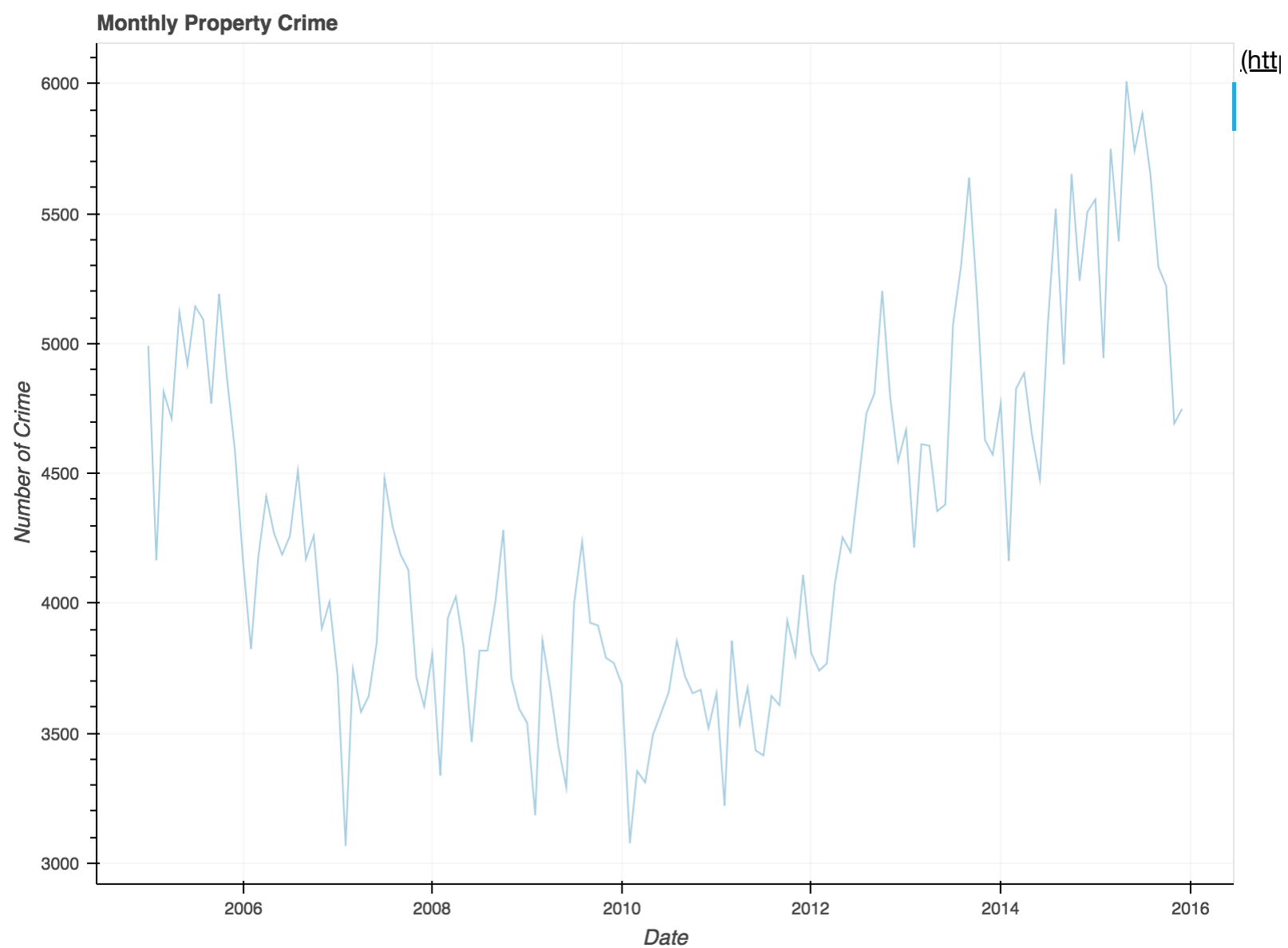
According to the diagram, one can see that the number of monthly property crime decreased from 2006 to 2010, and then increased after 2010.

Besides the trend of overall property crime, one also need to know the trends for different types of crimes respectively. A multiple line chart can be used to visualize this.

```
In [6]:  data.Category.unique()

Out[6]:  array(['BURGLARY', 'VANDALISM', 'LARCENY/THEFT', 'VEHICLE THEFT',
                'STOLEN PROPERTY', 'ARSON'], dtype=object)
```

```
In [7]:  # Multi Line Chart
         burg = data[data['Category']=='BURGLARY']
         burg = pd.DataFrame(burg.groupby('Date')['IncidntNum'].sum()).reset_index()

         vand = data[data['Category']=='VANDALISM']
         vand = pd.DataFrame(vand.groupby('Date')['IncidntNum'].sum()).reset_index()

         larc = data[data['Category']=='LARCENY/THEFT']
         larc = pd.DataFrame(larc.groupby('Date')['IncidntNum'].sum()).reset_index()

         vehi = data[data['Category']=='VEHICLE THEFT']
         vehi = pd.DataFrame(vehi.groupby('Date')['IncidntNum'].sum()).reset_index()

         stol = data[data['Category']=='STOLEN PROPERTY']
         stol = pd.DataFrame(stol.groupby('Date')['IncidntNum'].sum()).reset_index()

         arso = data[data['Category']=='ARSON']
         arso = pd.DataFrame(arso.groupby('Date')['IncidntNum'].sum()).reset_index()

         p2 = figure(x_axis_type="datetime", title="Monthly Property Crime", plot_width=800,plot_height=600)
         p2.grid.grid_line_alpha=0.3
         p2.xaxis.axis_label = 'Date'
         p2.yaxis.axis_label = 'Number of Crime'

         p2.line(burg['Date'], burg['IncidntNum'], color='red', legend='BURGLARY')
         p2.line(vand['Date'], vand['IncidntNum'], color='orange', legend='VANDALISM')
         p2.line(larc['Date'], larc['IncidntNum'], color='blue', legend='LARCENY/THEFT')
         p2.line(vehi['Date'], vehi['IncidntNum'], color='green', legend='VEHICLE THEFT')
         p2.line(stol['Date'], stol['IncidntNum'], color='brown', legend='STOLEN PROPERTY')
         p2.line(arso['Date'], arso['IncidntNum'], color='purple', legend='ARSON')

         p2.legend.location = "top_left"
         p2.legend.label_text_font_size = "7pt"
         show(p2)
```
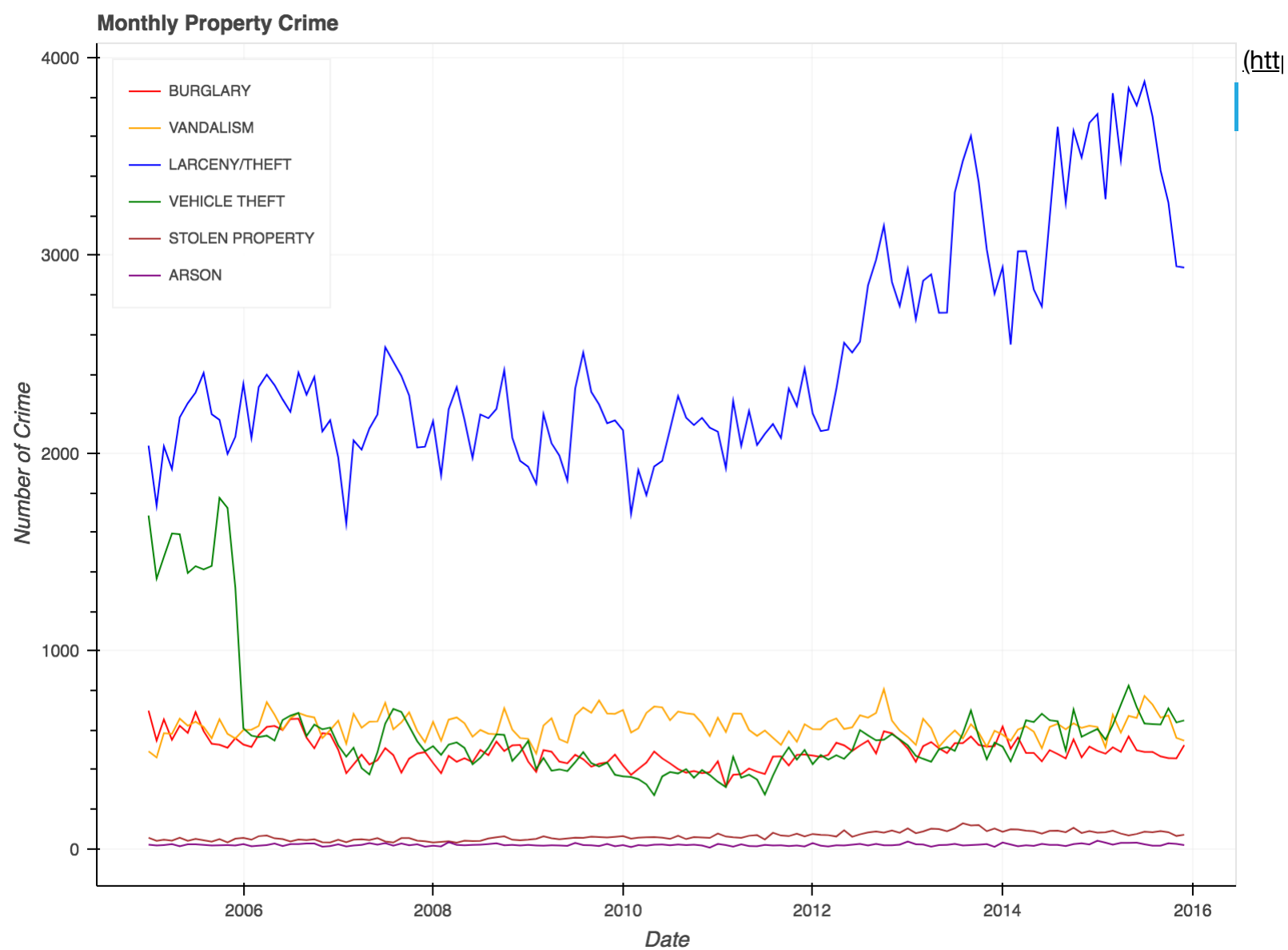

According to the diagram, one can see that larceny and theft is the most frequent crime and has been increasing throughout the time. Also there was a sharp decrease of vehicle theft in 2006. As for the other types of crime, they tend to be stationary and have no particular trend.

Since larceny and theft has the most number of occurence, it will be good to know it's yearly average number of occurence. A bar plot can be used to visualize this.

```
In [8]:  # Bar Chart
         larc = data[data['Category']=='LARCENY/THEFT']
         larc.Year = larc.Year.astype(str)
         group = larc.groupby('Year')

         source = ColumnDataSource(group)

         yr_cmap = factor_cmap('Year',palette=['blue']*11, factors=sorted(larc.Year.unique()))

         p3 = figure(plot_height=350, x_range=group, title="Average Number of Larceny/Theft",
                  toolbar_location=None, tools="")

         p3.vbar(x='Year', top='IncidntNum_mean',width=0.9, source=source, alpha=0.5,
              line_color=yr_cmap, fill_color=yr_cmap)

         p3.y_range.start = 0
         p3.xgrid.grid_line_color = None
         p3.xaxis.axis_label = "Year"
         p3.xaxis.major_label_orientation = 1.2
         p3.x_range.range_padding = 0.1

         p3.add_tools(HoverTool(tooltips=[("IncidntNum", "@IncidntNum_mean"), ("Year", "@Year")]))

         show(p3)
```
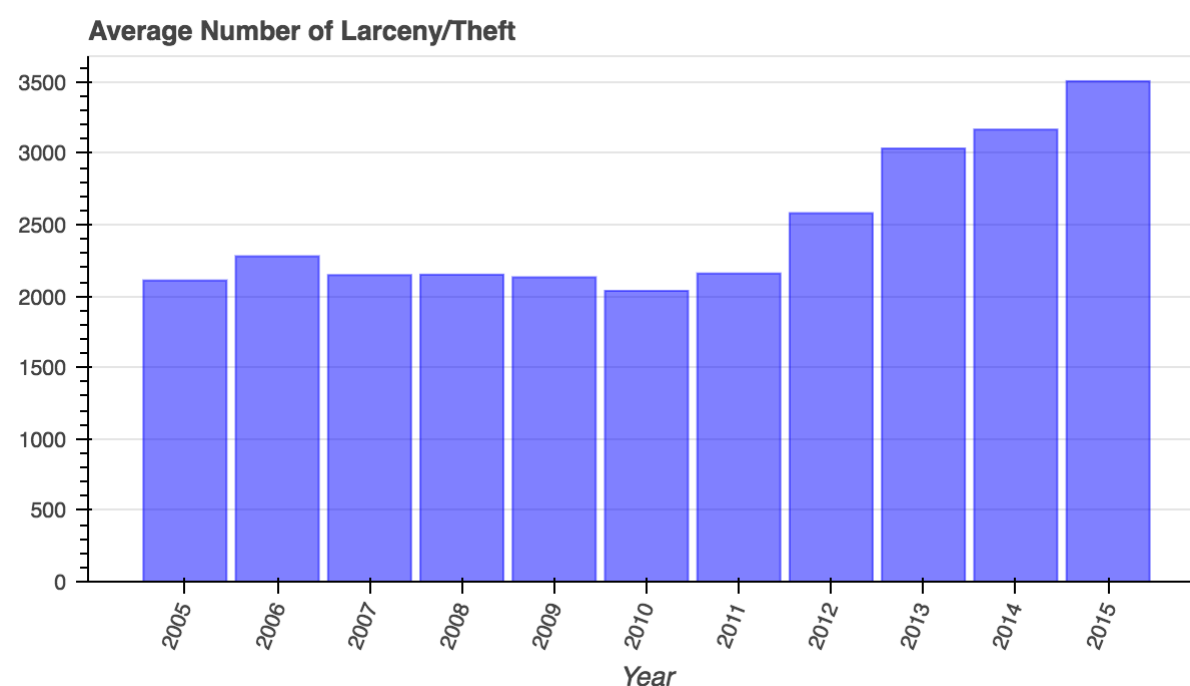
/Users/libingyi/anaconda/envs/python36/lib/python3.6/site-packages/pandas/core/generic.py:3110: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-vie
w-versus-copy
  self[name] = value

Average Number of Larceny/Theft

We can see that in 2010, the average number of larceny/theft is 2037, which is the lowest. And in 2015 the average number is 3506, which is the highest.