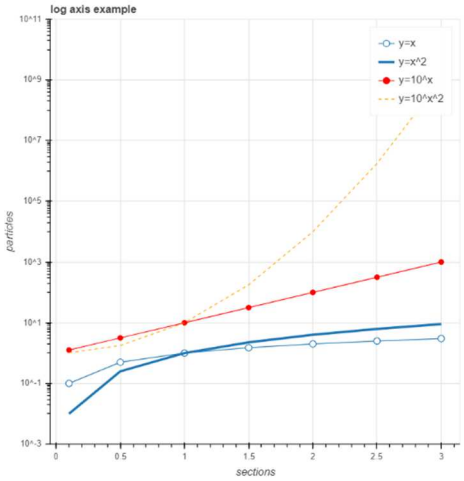
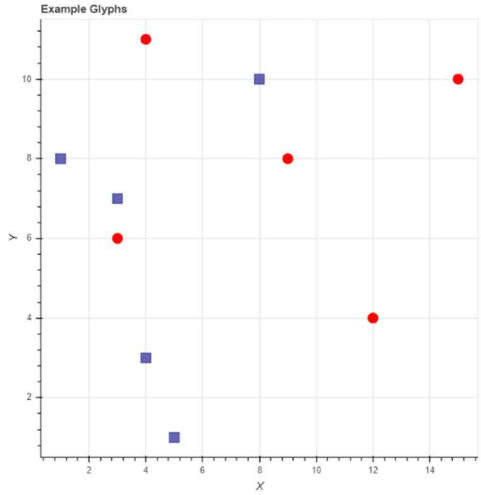
Bokeh

Several Line Plots




```
from bokeh.plotting import figure, output_file, show

# prepare some data
x = [0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0]
y0 = [i**2 for i in x]
y1 = [10**i for i in x]
y2 = [10**(i**2) for i in x]

# create a new plot
p = figure(
    tools="pan,box_zoom,reset,save",
    y_axis_type="log", y_range=[0.001, 10**11], title="log axis example",
    x_axis_label='sections', y_axis_label='particles'
)

# add some renderers
p.line(x, x, legend="y=x")
p.circle(x, x, legend="y=x", fill_color="white", size=8)
p.line(x, y0, legend="y=x^2", line_width=3)
p.line(x, y1, legend="y=10^x", line_color="red")
p.circle(x, y1, legend="y=10^x", fill_color="red", line_color="red", size=6)
p.line(x, y2, legend="y=10^x^2", line_color="orange", line_dash="4 4")

# show the results
show(p)
```

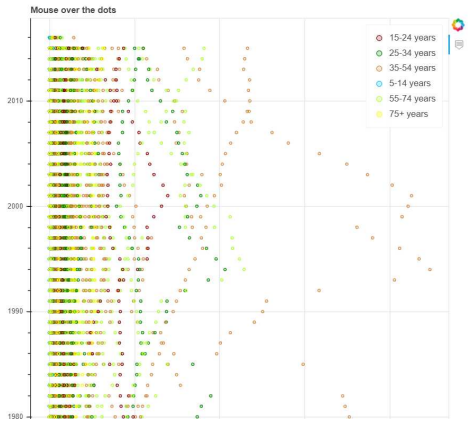
ScatterPlot




```
# bokeh basics
# Create a blank figure with labels
p = figure(plot_width = 600, plot_height = 600,
                    title = 'Example Glyphs',
                    x_axis_label = 'X', y_axis_label = 'Y')

# Example data
squares_x = [1, 3, 4, 5, 8]
squares_y = [8, 7, 3, 1, 10]
circles_x = [9, 12, 4, 3, 15]
circles_y = [8, 4, 11, 6, 10]

# Add squares glyph
p.square(squares_x, squares_y, size = 12, color = 'navy', alpha = 0.6)
# Add circle glyph
p.circle(circles_x, circles_y, size = 12, color = 'red')

# Set to output the plot in the notebook
output_notebook()
# Show the plot
show(p)
```

```
from bokeh.plotting import figure, output_file, show, ColumnDataSource
from bokeh.models import HoverTool, CategoricalColorMapper
```

```
from bokeh.io import output_notebook

output_notebook()

source = ColumnDataSource(
            data=wh
      )

hover = HoverTool(
            tooltips=[
                  ("year", "@year"),
                  ("sex", "@sex"),
                  ("country", "@country"),
                  ("suicides_no","@suicides_no")
            ]
      )

p = figure(plot_width=700, plot_height=700, tools=[hover],
                  title="Mouse over the dots")
#https://www.kaggle.com/kanncaa1/visualization-bokeh-tutorial-part-1
#https://github.com/bokeh/bokeh/issues/5112
#factors = list(wh.age.unique())
factors = ['5-14 years','15-24 years','25-34 years','35-54 years','55-74 years','75+ years']
colors = ["deepskyblue","maroon","green","peru","greenyellow","yellow"]
mapper = CategoricalColorMapper(factors = factors,palette = colors)
p.circle('suicides_no', 'year', size=4, source=source,
      legend='age', fill_alpha=0.2, color = {"field":"age","transform":mapper})

show(p)
```



WordCloud for text

https://www.datacamp.com/community/tutorials/wordcloud-python

```
text = " ".join(review for review in df.description)
print ("There are {} words in the combination of all review.".format(len(text)))

# Create stopword list:
stopwords = set(STOPWORDS)
stopwords.update(["drink", "now", "wine", "flavor", "flavors"])

# Generate a word cloud image
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)

# Display the generated image:
```

```
# the matplotlib way:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Pandas Bokeh

Libraries

!pip install pandas-bokeh

```
import numpy as np
import pandas as pd
import pandas_bokeh
pandas_bokeh.output_notebook()
pd.set_option('plotting.backend', 'pandas_bokeh')
# Create Bokeh-Table with DataFrame:
from bokeh.models.widgets import DataTable, TableColumn
from bokeh.models import ColumnDataSource
```

Line Graph / Line Graph (rangetools) / Step Plot

```
# Normal Line Graphs
df_line = pd.DataFrame({"Northern Region": df["NR"].values,
                        "Southern Region": df["SR"].values,
                        "Eastern Region": df["ER"].values,
                        "Western Region": df["WR"].values,
                        "North Eastern Region": df["NER"].values},index=df.Date)

df_line.plot_bokeh(kind="line",title ="India - Power Consumption Regionwise",
                   figsize =(1000,800),
                   xlabel = "Date",
                   ylabel="MU(millions of units)"
                   )
```

```
# Line graph using Range Tools
df_line = pd.DataFrame({"Northern Region": df["NR"].values,
                        "Southern Region": df["SR"].values,
                        "Eastern Region": df["ER"].values,
                        "Western Region": df["WR"].values,
                        "North Eastern Region": df["NER"].values},index=df.Date)

df_line.plot_bokeh(kind="line",title ="India - Power Consumption Regionwise",
                   figsize =(1000,800),
                   xlabel = "Date",
                   ylabel="MU(millions of units)",rangetool=True
                   )
```

 # Step plot

```
df_line.plot_bokeh.step(
    x=df.Date,
    xticks=range(-1, 1),
    colormap=["#009933", "#ff3399","#ae0399","#220111","#890300"],
    title="Step Plot - India Power Consumption",
```

```
        figsize=(1000,800),
        fontsize_title=20,
        fontsize_label=20,
        fontsize_ticks=20,
        fontsize_legend=8,
        )
```

Bar Graph

```
# Bar Graph (with x = 'time series / date')
df_line.plot_bokeh(kind="bar",title ="India - Power Consumption Regionwise",figsize =(1000,800),xlabel = "Date",ylabel="MU(millions of units)")

# Bar Graph (with x = 'categorical variable')
data = {
        'Cars':
        ['Maruti Suzuki', 'Honda', 'Toyota', 'Hyundai', 'Benz', 'BMW'],
        '2018': [20000, 15722, 4340, 38000, 2890, 412],
        '2019': [19000, 13700, 340, 31200, 290, 234],
        '2020': [23456, 15891, 440, 36700, 890, 417]
}
df = pd.DataFrame(data).set_index("Cars")

p_bar = df.plot_bokeh.bar(
        ylabel="Price per Unit",
        title="Car Units sold per Year",
        alpha=0.6)

# Stacked Bar Plot
data = {
        'Cars':
        ['Maruti Suzuki', 'Honda', 'Toyota', 'Hyundai', 'Benz', 'BMW'],
        '2018': [20000, 15722, 4340, 38000, 2890, 412],
        '2019': [19000, 13700, 340, 31200, 290, 234],
        '2020': [23456, 15891, 440, 36700, 890, 417]
}
df = pd.DataFrame(data).set_index("Cars")

stacked_bar = df.plot_bokeh.bar(
        ylabel="Price per Unit",
        title="Car Units sold per Year",
        stacked=True,
        alpha=0.6)

# Horizontal Bar Plot
(same df as previous bar plot example of cars)
#Reset index, such that "Cars" is now a column of the DataFrame:
df.reset_index(inplace=True)

#Create horizontal bar (via kind keyword):
p_hbar = df.plot_bokeh(
        kind="barh",
        x="Cars",
        ylabel="Price per Unit",
        title="Car Units sold per Year",
```

```
        alpha=0.6,
        legend = "bottom_right",
        show_figure=False)

#Create stacked horizontal bar (via barh accessor):
stacked_hbar = df.plot_bokeh.barh(
        x="Cars",
        stacked=True,
        ylabel="Price per Unit",
        title="Car Units sold per Year",
        alpha=0.6,
        legend = "bottom_right",
        show_figure=False)
```

Scatter / Point Plot

```
# Pointplot 1
df_line.plot_bokeh(kind="point",title ="India - Power Consumption Regionwise",figsize =(1000,800),xlabel =
"Date",ylabel="MU(millions of units)")

# PointPlot 2
df_line.plot_bokeh.point(
        x=df.Date,
        xticks=range(0,1),
        size=5,
        colormap=["#009933", "#ff3399","#ae0399","#220111","#890300"],
        title=" Point Plot - India Power Consumption",
        fontsize_title=20,
        marker="x",figsize =(1000,800))

# Scatter
df = pd.read_csv("../input/iris/Iris.csv")
df = df.sample(frac=1)

data_table = DataTable(
        columns=[TableColumn(field=Ci, title=Ci) for Ci in df.columns],
        source=ColumnDataSource(df),
        height=300,
)

# Create Scatterplot:
p_scatter = df.plot_bokeh.scatter(
        x="PetalLengthCm",
        y="SepalWidthCm",
        category="Species",
        title="Iris DataSet Visualization",
        show_figure=False
)

# Combine Table and Scatterplot via grid layout:
pandas_bokeh.plot_grid([[data_table, p_scatter]], plot_width=400, plot_height=350)
```

Histogram

```
df_line.plot_bokeh(kind="hist",title ="India - Power Consumption Regionwise",
                                figsize =(1000,800),
                                xlabel = "Date",
```

```
                ylabel="MU(millions of units)"
            )
```