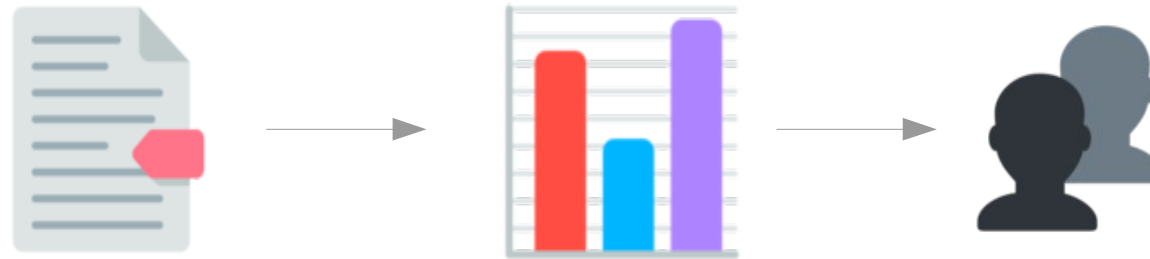


A tour of data viz in Python



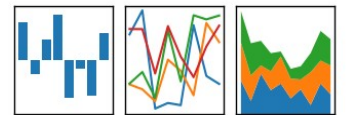
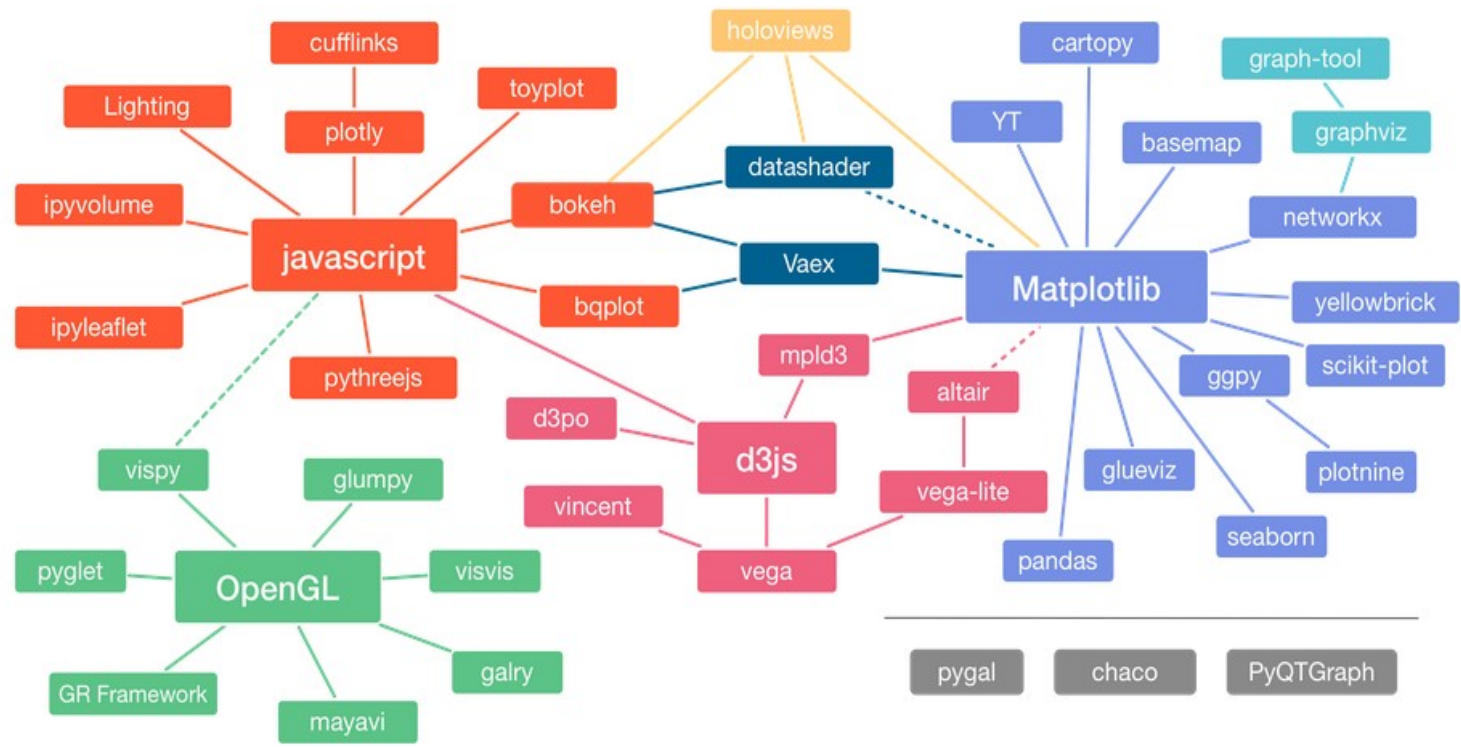
Éléonore Mayola





Different needs for different use cases

- yourself
- your colleagues
- your manager
- your clients
- exploring a dataset
- writing an internal report
- writing a client report
- writing a research article

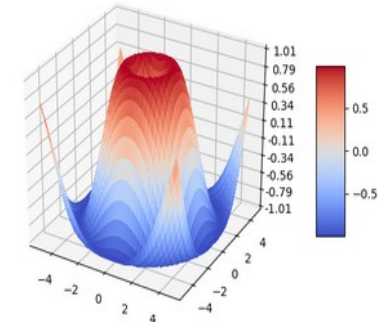


Adaptation of Jake VanderPlas graphic about the Python visualization landscape, by Nicolas P. Rougier

Source: <https://pyviz.org/overviews/index.html>

matplotlib.org

“Matplotlib tries to make easy things easy and hard things possible.”



The **pyplot** module provides a MATLAB-like interface

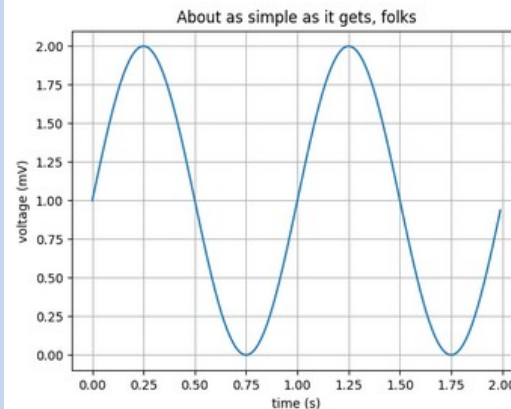
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
```

```
# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)
```

```
fig, ax = plt.subplots()
ax.plot(t, s)
```

```
ax.set(xlabel='time (s)', ylabel='voltage (mV)',
       title='About as simple as it gets, folks')
ax.grid()
```

```
fig.savefig("test.png")
plt.show()
```

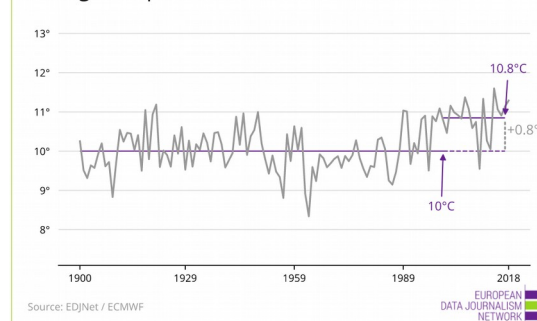


Customisable labels, lines and annotations...

```
matplotlib.axes.Axes.axhline
ax.axhline(y=10yrs_average, color='#6f2c91',
           linewidth=1.5)
```

```
matplotlib.pyplot.annotate
ax.annotate(f'{temperature}°C',
            xy=(3, 1), xycoords='data',
            xytext=(0.8, 0.95),
            arrowprops=dict(facecolor='#6f2c91'),
            horizontalalignment='right',
            verticalalignment='top')
```

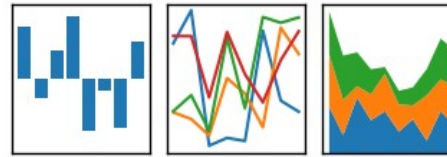
Average temperature in and around London



onedegreewarmer.eu

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

**pandas.pydata.org**

“[...] high-performance, easy-to-use data structures and data analysis tools for the Python programming language.”

Uses Matplotlib for plotting

```
import matplotlib.pyplot as plt
```

```
DataFrame.plot(x=None, y=None, kind='line', ax=None, subplots=False, sharex=None, sharey=False, layout=None, figsize=None, use_index=True, title=None, grid=None, legend=True, style=None, logx=False, logy=False, loglog=False, xticks=None, yticks=None, xlim=None, ylim=None, rot=None, fontsize=None, colormap=None, table=False, yerr=None, xerr=None, secondary_y=False, sort_columns=False, **kwargs)
```

```
In [1]: %matplotlib inline
```

```
In [2]: import matplotlib
import matplotlib.pyplot as plt
matplotlib.style.use('seaborn')
import numpy as np
import pandas as pd
```

```
In [3]: df = pd.DataFrame(np.random.randn(1000, 4),
                           columns=list('ABCD'))

df = df.cumsum()

plt.figure()

df.plot()
plt.show()
```

<Figure size 576x396 with 0 Axes>





plot.ly/python



Plotly Python Open Source Graphing Library

Plotly's Python graphing library makes interactive, publication-quality graphs. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, polar charts, and bubble charts.

```
In [1]: import plotly.express as px

In [2]: Gapminder = px.data.gapminder().query("continent=='Oceania'")
fig = px.line(Gapminder, x="year", y="lifeExp", color='country')
fig.show()
```



plotly.py

github.com/plotly/plotly.py

`pip install plotly==4.1.0`

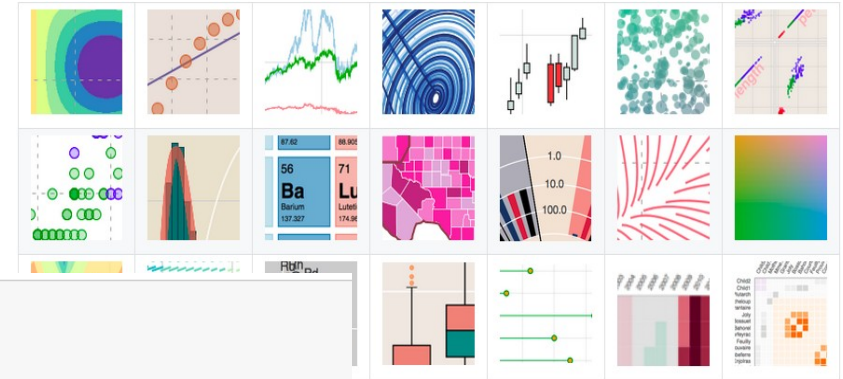
Plotly Express

github.com/plotly/plotly_express

`pip install plotly_express==0.4.1`

bokeh.org**github.com/bokeh/bokeh**

“Bokeh is an **interactive** visualization library for Python that enables beautiful and meaningful visual presentation of data in modern web browsers.”



```
In [1]: from bokeh.io import output_notebook, show
        from bokeh.plotting import figure

        # Install bokeh sample data (the first time you run it)
        # import bokeh
        # bokeh.sampledata.download()

        # Glucose sample data requires Pandas to be installed
        from bokeh.sampledata.glucose import data

In [2]: output_notebook()

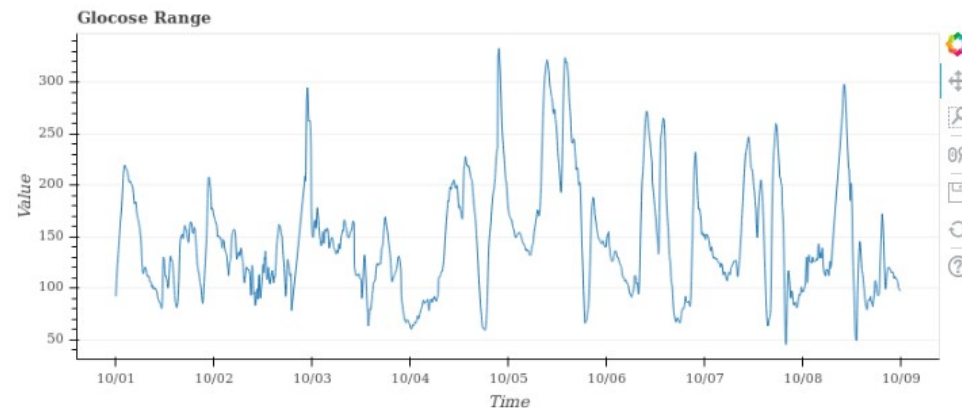
        BokehJS 1.3.4 successfully loaded.

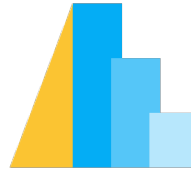
In [3]: # reduce data size to one week
        week = data.loc['2010-10-01':'2010-10-08']

        p = figure(x_axis_type="datetime", title="Glucose Range", plot_height=350, plot_width=800)
        p.xgrid.grid_line_color=None
        p.ygrid.grid_line_alpha=0.5
        p.xaxis.axis_label = 'Time'
        p.yaxis.axis_label = 'Value'

        p.line(week.index, week.glucose)

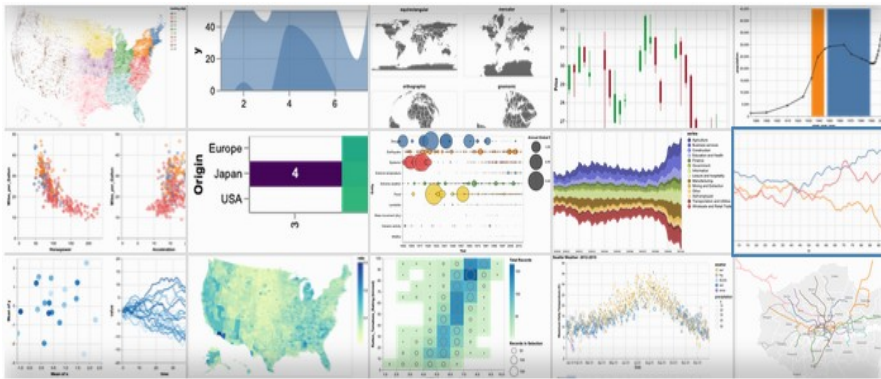
        show(p)
```





altair-viz.github.io

“Altair is a declarative statistical visualization library for Python, based on Vega and Vega-Lite.”



vega.github.io/vega

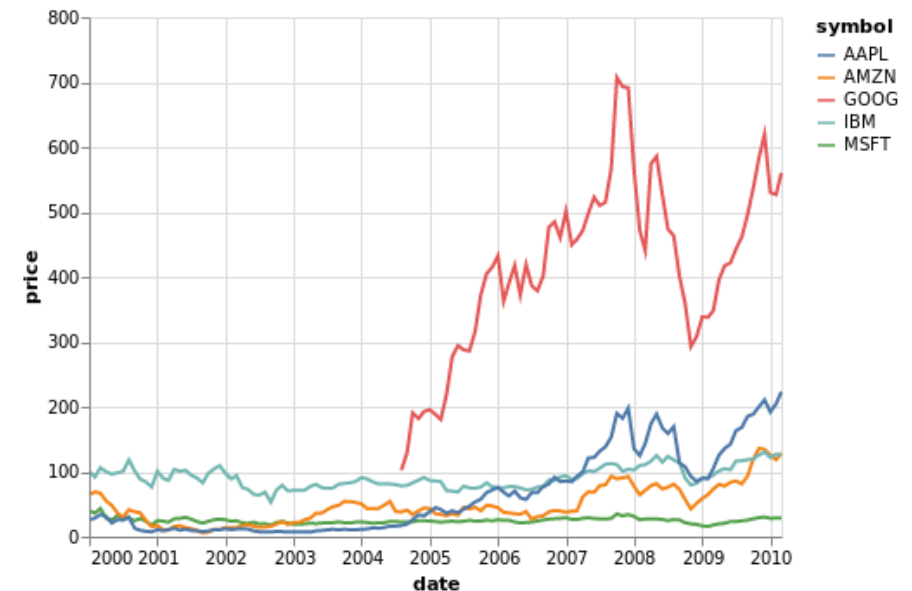
“Vega is a visualization grammar, a declarative language for creating, saving, and sharing **interactive** visualization designs.”

```
In [1]: import altair as alt
alt.renderers.enable('notebook')

from vega_datasets import data
```

```
In [2]: source = data.stocks()

alt.Chart(source).mark_line().encode(
    x='date',
    y='price',
    color='symbol'
)
```



Out[2]:

Best for

Challenges

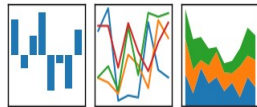


Complex or customised plots

Syntax can become tricky

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Plotting during data analysis

Not the most aesthetic
→ try Seaborn



Notebook or plain html report

Maintained by a private company



Interactive plots or dashboards

Tricky to create dashboard apps

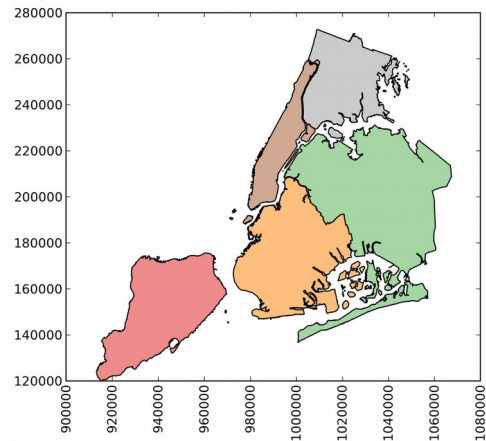


Interactive plots and maps

One main maintainer

GeoPandas

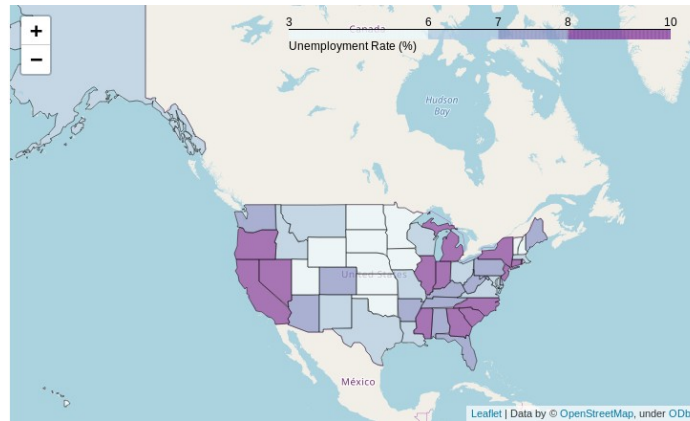
<https://github.com/geopandas/geopandas>



Folium



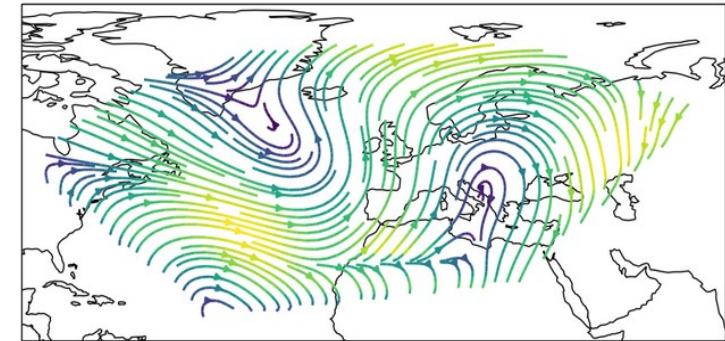
<https://github.com/python-visualization/folium>



CartoPy

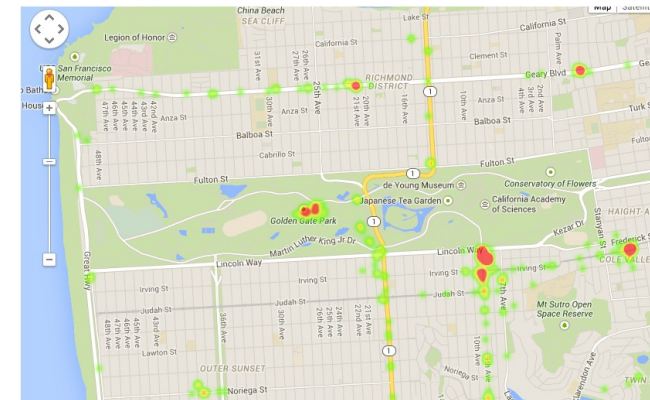


<https://github.com/SciTools/cartopy>



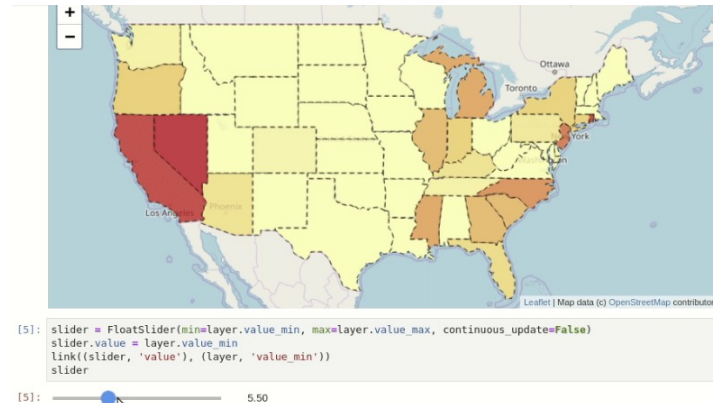
gmplot

<https://github.com/vgm64/gmplot>



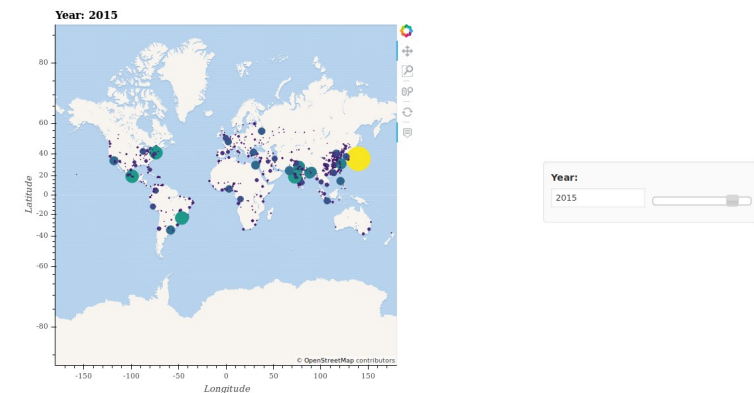
ipyleaflet

<https://github.com/jupyter-widgets/ipyleaflet>



geoviews

<https://github.com/pyviz/geoviews>



Thank you



github.com/Eleonore9/tour_dataviz_python

- Exhaustive list of Python tools for data viz:
pyviz.org/tools.html
- Libraries mentioned:

matplotlib.org

pandas.pydata.org

plot.ly

bokeh.org

altair-viz.github.io

 **matplotlib**



pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

