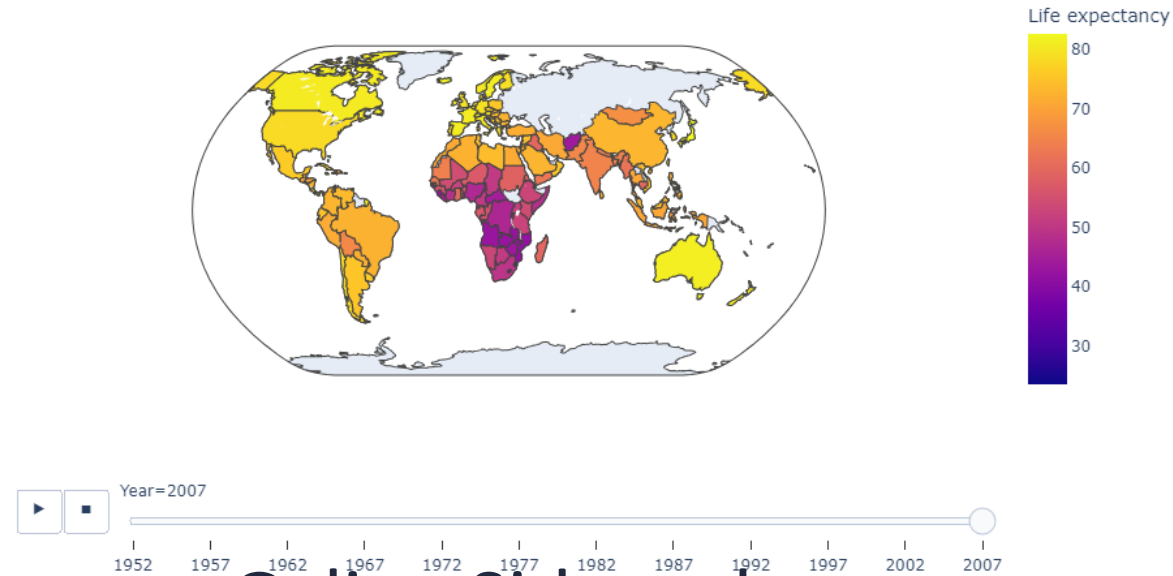










Plotly - Open Source Graphing Library for Python



Galina Sidorenko
March 2024

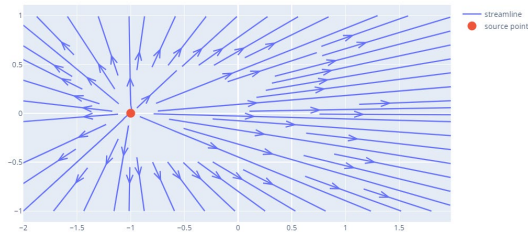
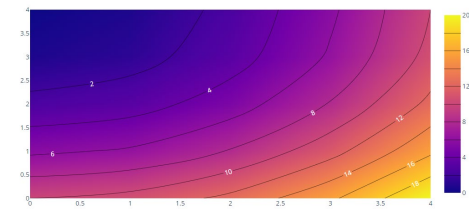
Plotly - Open Source Graphing Library for Python

<https://plotly.com/>

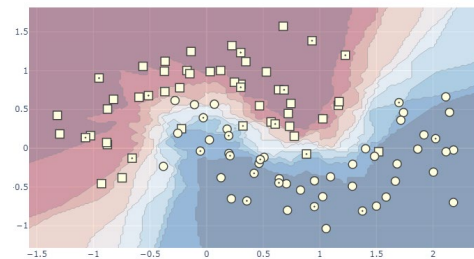
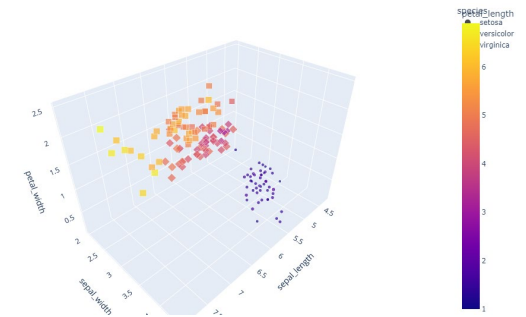
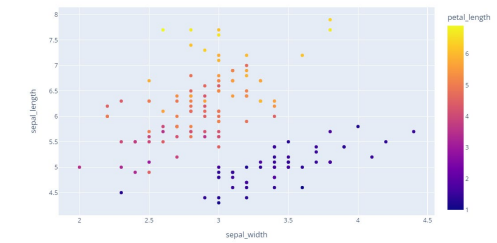
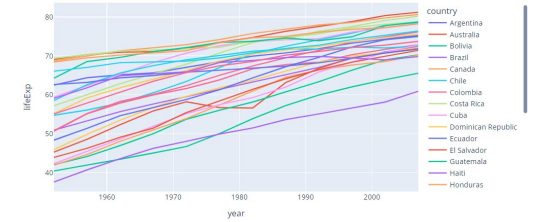
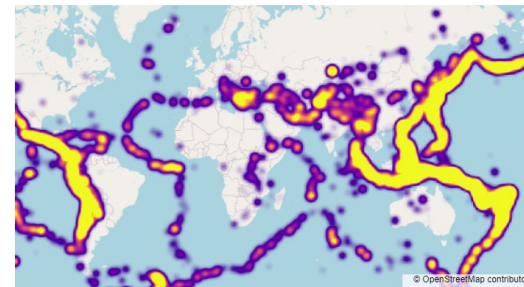
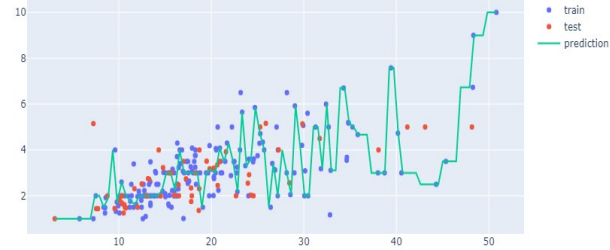
 <p>Plotly Python Open Source Graphing Library</p> <p>Star 14,972</p>	 <p>Plotly R Open Source Graphing Library</p> <p>Star 2,480</p>	 <p>Plotly Julia Open Source Graphing Library</p> <p>Star 403</p>	 <p>Plotly Javascript Open Source Graphing Library</p> <p>Star 16,419</p>
 <p>Plotly ggplot2 Open Source Graphing Library</p> <p>Star 2,480</p>	 <p>Plotly F# Open Source Graphing Library</p> <p>Star 541</p>	 <p>Plotly MATLAB® Open Source Graphing Library</p> <p>Star 361</p>	 <p>Plotly Dash Open Source Analytical App Framework</p> <p>Star 20,242</p>

Plotly: an overview

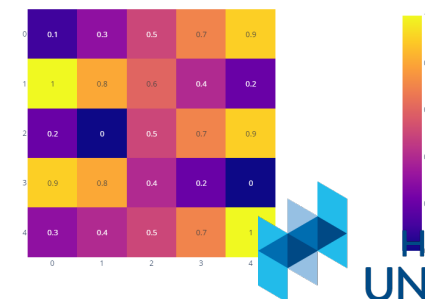
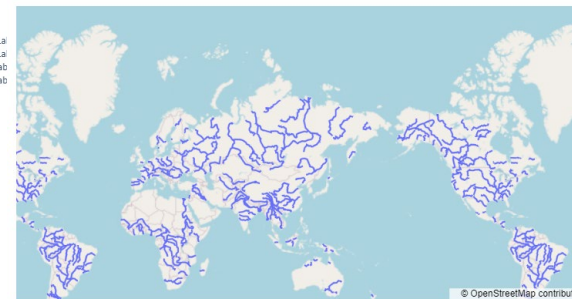
- Interactive, open-source library
- Supports over 40 unique chart types



<https://plotly.com/python/>



□ Train Split, La
○ Train Split, La
□ Test Split, Lab
○ Test Split, Lab



Plotly: an overview

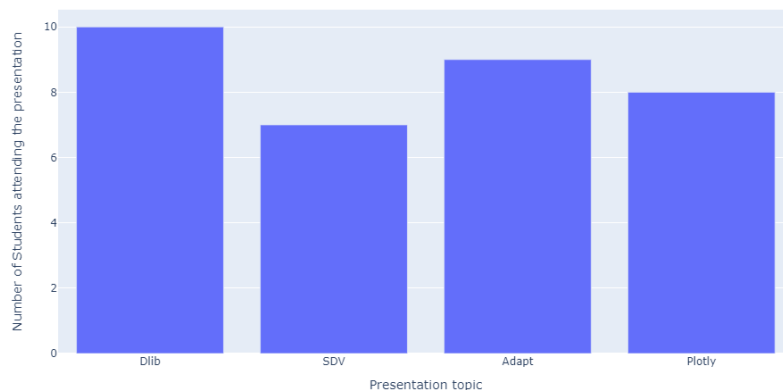
- Interactivity: zooming, panning, hovering to display tooltips, and toggling visibility of specific data traces.
- Customization: colors, markers, lines, fonts, axis labels, annotations
- Output:
 - Displayed in Jupyter Notebook
 - Saved to static images (e.g., PNG, JPEG)
 - Saved to HTML files
 - Used in developing Python-build web applications using Dash

Plotly: an overview

- Built on top of the Plotly JavaScript library ([plotly.js](#))
- Graphical figures are represented by data structures, which are commonly referred to as figures. Alternatively, figures can be represented as Python dictionaries.
- Plotly figures are serialized into JavaScript Object Notation (JSON) format before being passed to the plotly.js library

```
In [110]: pres_names = ['Dlib', 'SDV', 'Adapt', 'Plotly']
n_stud = [10, 7, 9, 8]

fig = px.bar(x=pres_names, y=n_stud, labels={'x': 'Presentation topic', 'y': 'Number of Students attending the presentation'})
fig.show() #displaying figure
print(fig) #printing figure in JSON-format
```



```
Figure({
  'data': [{ 'alignmentgroup': 'True',
    'hovertemplate': ('Presentation topic=%{x}<br>Num ... 'esentation=%{y}<extra></extra>'),
    'legendgroup': '',
    'marker': { 'color': '#636efa', 'pattern': { 'shape': '' } },
    'name': '',
    'offsetgroup': '',
    'orientation': 'v',
    'showlegend': False,
    'textposition': 'auto',
    'type': 'bar',
    'x': array(['Dlib', 'SDV', 'Adapt', 'Plotly'], dtype=object),
    'xaxis': 'x',
    'y': array([10, 7, 9, 8], dtype=int64),
    'yaxis': 'y' }],
  'layout': { 'barmode': 'relative',
    'legend': { 'tracegroupgap': 0 },
    'margin': { 't': 60 },
    'template': '...',
    'xaxis': { 'anchor': 'y', 'domain': [0.0, 1.0], 'title': { 'text': 'Presentation topic' } },
    'yaxis': { 'anchor': 'x',
    'domain': [0.0, 1.0],
    'title': { 'text': 'Number of Students attending the presentation' } } } })
```

Plotly: Displaying Figures in Python

```
fig = px.bar(x=pres_names, y=n_stud, labels={'x':'Presentation topic', 'y':'Number of Students attending the presentation'})
```

- Output displayed Using The renderers Framework:
 - `fig.show()`
 - `plotly.io.show(fig)`
 - `fig`
- It is possible to specify renderer:
 - `fig.show(renderer="png")`
 - `fig.show(renderer="pdf")`

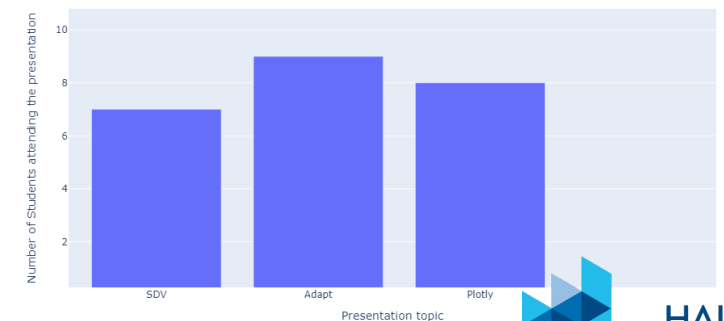
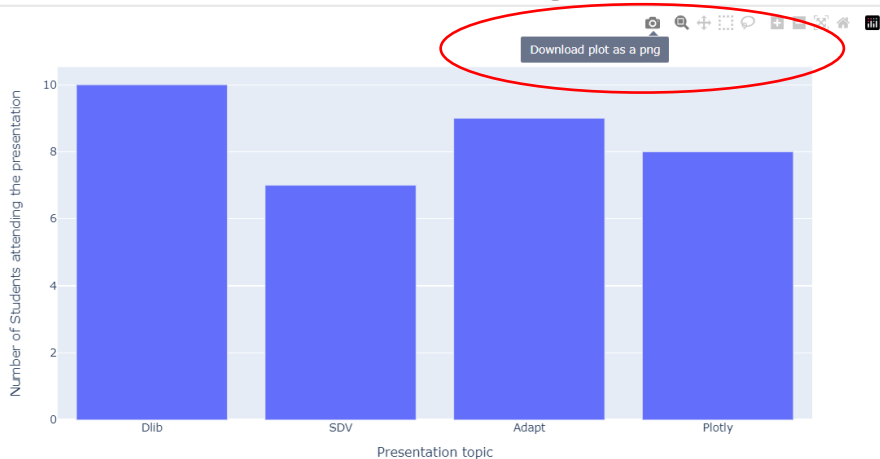
and even customize renderer:

```
import plotly.io as pio
png_renderer = pio.renderers["png"]
png_renderer.width = 500
png_renderer.height = 500
pio.renderers.default = "png"
```

Plotly: Export as a static file format

```
fig = px.bar(x=pres_names, y=n_stud, labels={'x':'Presentation topic', 'y':'Number of Students attending the presentation'})
```

- Static Format (PNG, JPEG, SVG or PDF):
 - `fig.write_image("fig1.jpeg")`
 - `fig.write_image("fig1.png",width=600, height=350)`
 - `fig.write_image("fig1.pdf")`
- Kaleido is default rendering engine being used by Plotly. Kaleido is a high-performance rendering engine developed by Plotly. It is designed to efficiently convert Plotly figures into various static image formats



Plotly: Export as an interactive HTML file

```
fig = px.bar(x=pres_names, y=n_stud, labels={'x': 'Presentation topic', 'y': 'Number of Students attending the presentation'})
```

- HTML Format:
 - `fig.write_html("file.html")`
- It can be uploaded to a web server, shared via email or other file-sharing mechanisms.
- However, contains an inlined copy of the Plotly.js library required to make the figure interactive

Power Point:



Add-Ins: Power BI?
Web viewer?

Plotly sub-modules

- Plotly Graph Objects: low level interface to figures, traces and layouts
- Plotly Express: high-level wrapper
- Plotly Subplots
- Plotly IO for saving Plotly figures in different formats, displaying Plotly figures in different environments, configuring various settings related to plot rendering and display

Plotly sub-modules: Plotly Graph Objects

- Plotly Graph Objects provides a lower-level API that gives users more control and flexibility over the creation of plots.
- It allows users to create plots by explicitly defining traces, layouts, and other components of the plot.
- It offers fine-grained control over every aspect of the plot, allowing users to customize the plot to their exact specifications.
- It's suitable for more complex visualizations, customization, and advanced plotting requirements.

Plotly sub-modules: Plotly Graph Objects

- Plotly Express is a high-level API that provides a simpler interface for creating various types of plots quickly.
- It offers a wide range of functions to create different types of plots (e.g., scatter plots, bar charts, line charts, etc.) with minimal code.
- It automatically handles the creation of figure objects and layouts, making it easier for users to create plots without worrying about low-level details.
- It's suitable for rapid prototyping, exploratory data analysis, and creating simple visualizations with ease.
- Plotly Express is the recommended entry-point into the plotly package

Using plotly.graph_objects

```
#Import required packages
import plotly.graph_objects as go
import plotly.express as px
import numpy as np

#Creating data
np.random.seed(10)
x=np.arange(12)
y=np.random.randint(100, 1000, size=12)

#Creating figure with plotly.graph_objects
fig=go.Figure(data=go.Scatter(x=x,y=y))
fig.update_layout(title='Simple Line Plot', xaxis_title='Month', yaxis_title='Sales')
fig.show()
```

Using plotly.express

```
#Import required packages
import plotly.graph_objects as go
import plotly.express as px
import numpy as np

#Creating data
np.random.seed(10)
x=np.arange(12)
y=np.random.randint(100, 1000, size=12)

#Creating figure with plotly.express
fig=px.line(x=x, y=y, title='Simple Line Plot', labels=dict(x='Month', y='Sales'))
fig.show()
```

Plotly: Customization

- **Titles**
- **Legends**
- **Axis Range and Scaling**
- **Marker and Line Styles**
- **Hover Formatting**
- **Fonts and Text Styling**
- **Layout and Margins**
- **Colorbar**
- **...**

Plotly vs Matplotlib

- Plotly suits better for interactive plots, for web visualization, exploring datasets, fast creating good-looking figures just with few line of code
- Matplotlib suits better for professional scientific plots with a lot of features be customized in a more complex way

Practice