Wizualizacja FFT przy użyciu Dash Plotly

Filip Marciniak

Politechnika Poznańska

2023

Plan prezentacji

1. Krótkie omówienie języka Python
2. Narzędzia do wizualizacji danych w Pythonie
3. Plotly oraz Dash
4. FFT
5. Wizualizacja FFT przy użyciu Dash
6. Podsumowanie

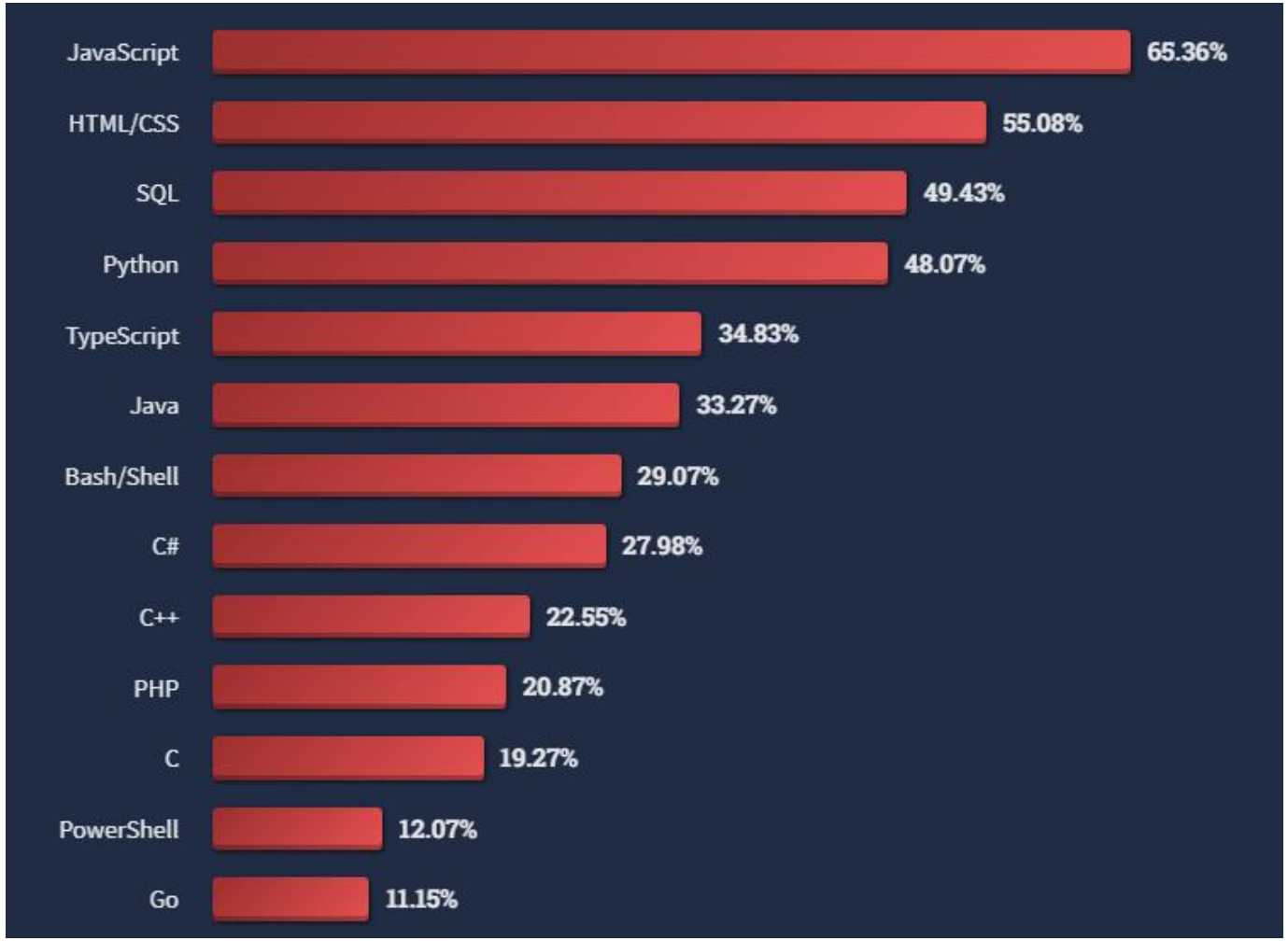
Język programowania Python

1. Prosty i łatwy do nauki
2. Interpretowany i dynamiczny
3. Zorientowany obiektowo
4. Wysoce rozszerzalny
5. Budowany również przez społeczność

Przykładowy program w języku Python

```
print("Hello! What is your name?")
name = input()

if name == "Alice" or name == "Bob":
    print("Nice to meet you, " + name + "! You have a great name.")
else:
    print("Nice to meet you, " + name + "!")

num_letters = 0
for letter in name:
    if letter.isalpha():
        num_letters += 1

print("Your name has " + str(num_letters) + " letters.")
```

Python vs inne języki programowania



Stack Overflow 2022 Developer Survey
https://survey.stackoverflow.co/2022/#overview

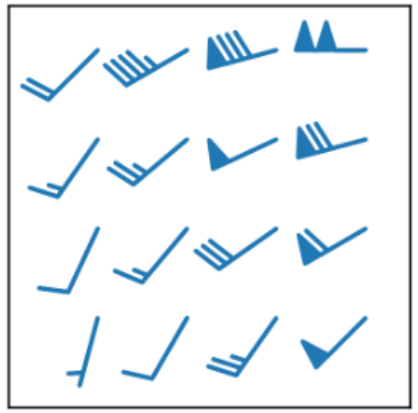
Narzędzia do wizualizacji danych w Pythonie

Do najpopularniejszych bibliotek służących do wizualizacji danych z użyciem Pythona można zaliczyć np. matplotlib lub seaborn

Matplotlib

matplotlib

Plot types Examples Tutorials Reference User guide Develop Release notes



barbs(X, Y, U, V)

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

Try Matplotlib (on Binder) →

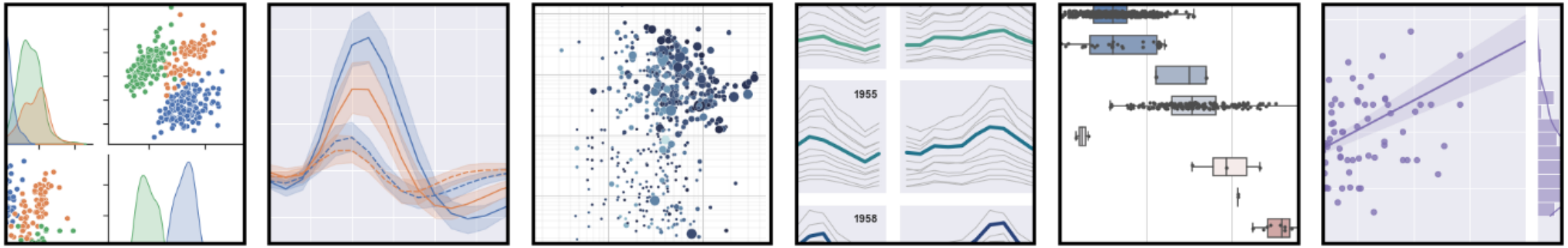
Getting Started Examples Reference Cheat Sheets Documentation

Seaborn

seaborn

Installing Gallery Tutorial API Releases Citing FAQ

seaborn: statistical data visualization



Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Visit the installation page to see how you can download the package and get started with it. You can browse the example gallery to see some of the things that you can do with seaborn, and then check out the tutorials or API reference to find out how.

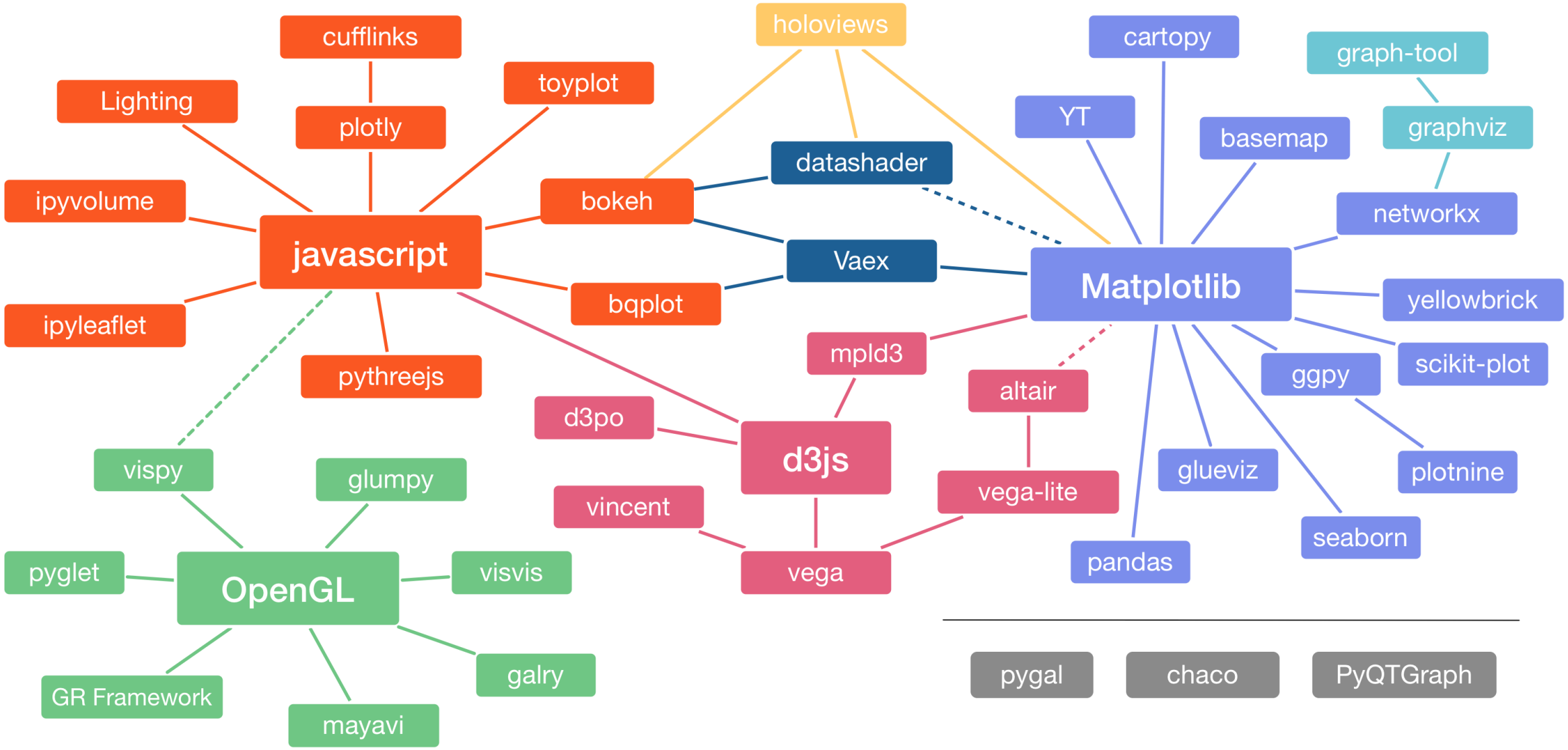
To see the code or report a bug, please visit the GitHub repository. General support questions are most at home on stackoverflow, which has a dedicated channel for seaborn.

Contents

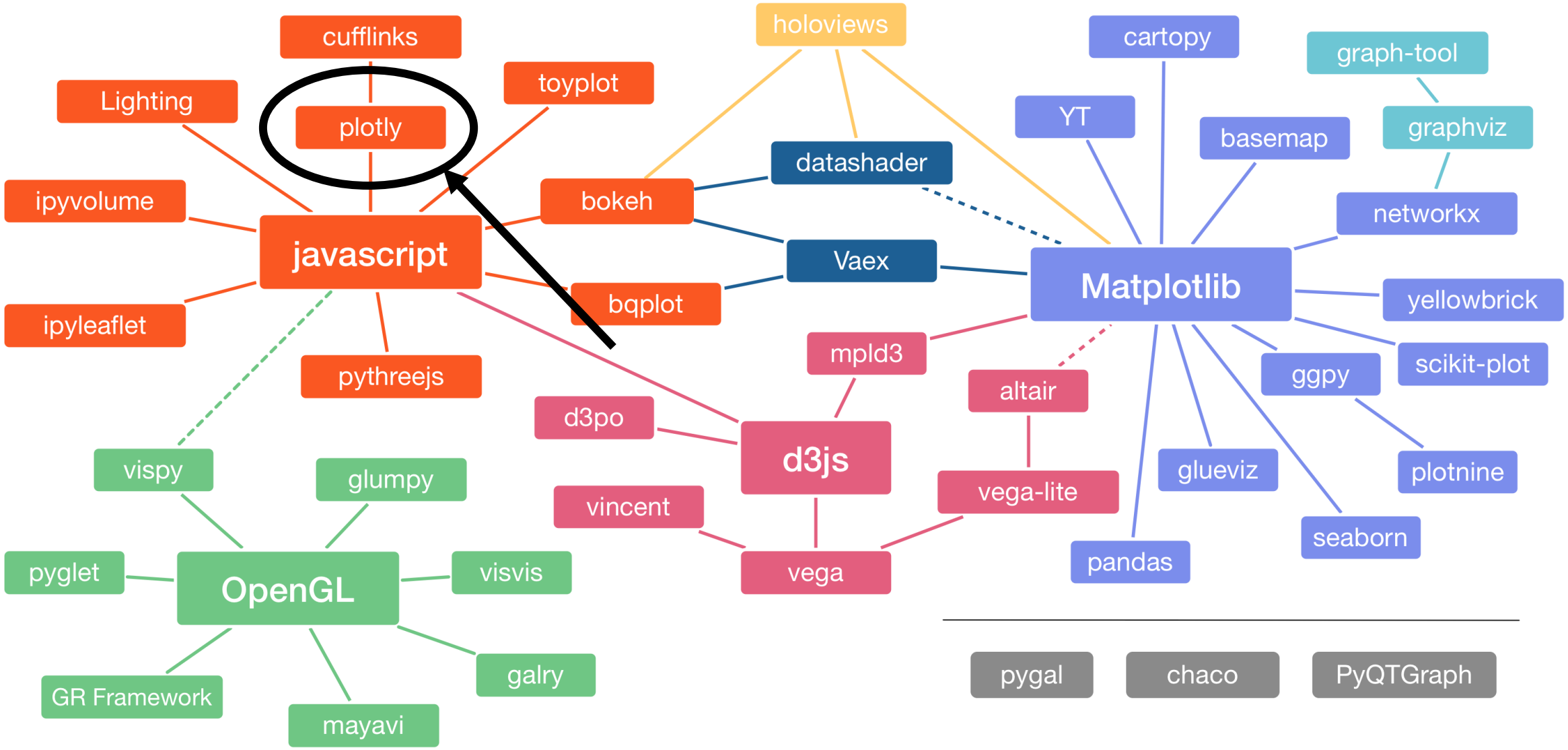
Features

- New Objects: API | Tutorial
- Relational plots: API | Tutorial
- Distribution plots: API | Tutorial
- Categorical plots: API | Tutorial
- Regression plots: API | Tutorial
- Multi-plot grids: API | Tutorial
- Figure theming: API | Tutorial
- Color palettes: API | Tutorial

Graf bibliotek do wizualizacji danych



Graf bibliotek do wizualizacji danych



Plotly

Plotly Open Source Graphing Libraries

Interactive charts and maps for Python, R, Julia, Javascript, ggplot2, F#, MATLAB®, and Dash.

Plotly Python
Open Source
Graphing Library
Star 12,938

Plotly R
Open Source
Graphing Library
Star 2,324

Plotly Julia
Open Source
Graphing Library
Star 362

Plotly Javascript
Open Source
Graphing Library
Star 15,413

Plotly ggplot2
Open Source
Graphing Library
Star 2,324

Plotly F#
Open Source
Graphing Library
Star 395

Plotly MATLAB®
Open Source
Graphing Library
Star 331

Plotly Dash
Open Source
Analytical App Framework
Star 18,105

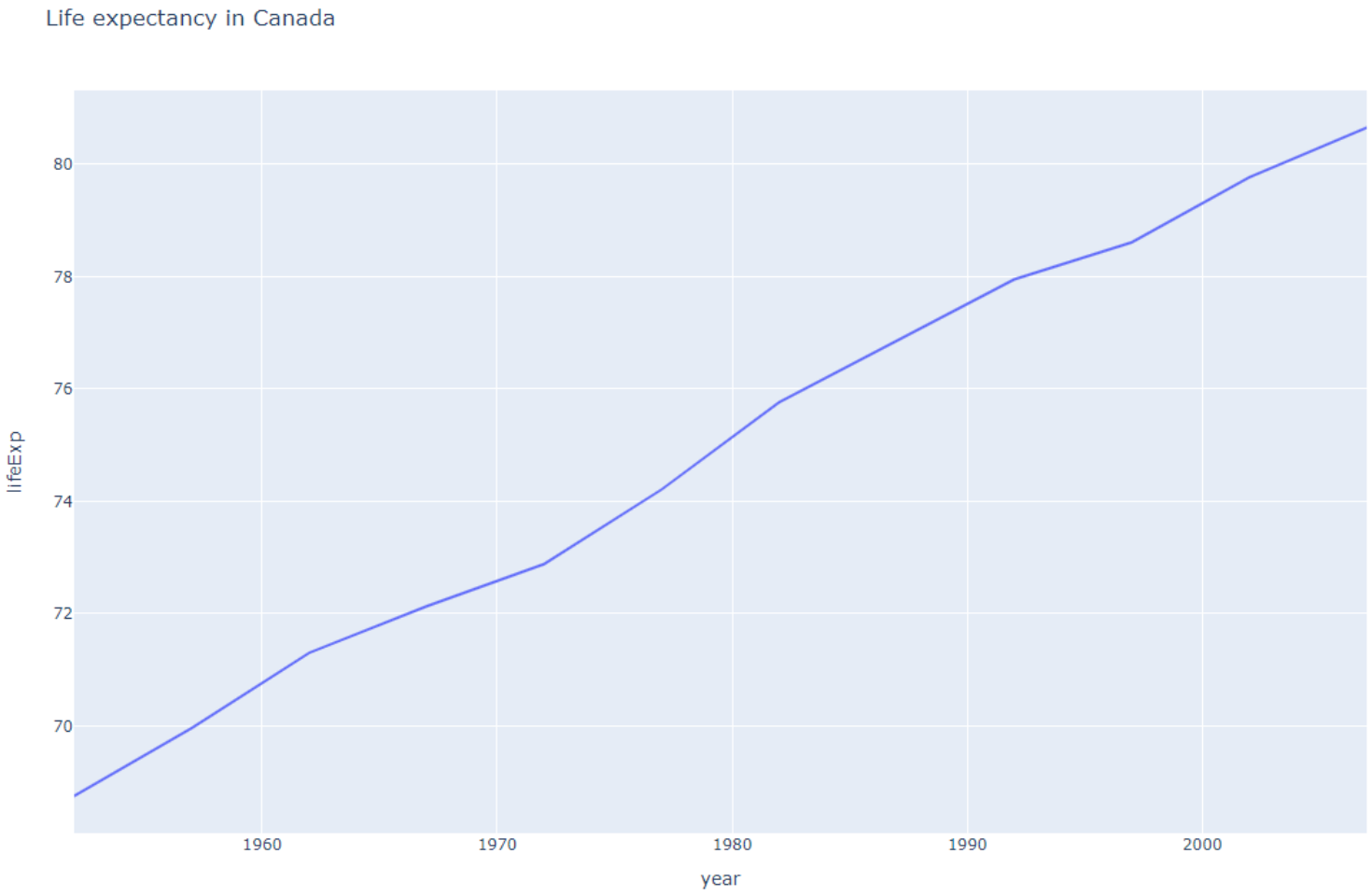
Plotly Python

Plotly is a Python library for creating interactive and publication-quality graphs, charts, and visualizations that can be easily embedded in web applications or exported as image files.

Przykład numer 1 – wykres liniowy

```
import plotly.express as px

df = px.data.gapminder().query("country=='Canada'")
fig = px.line(df, x="year", y="lifeExp", title='Life expectancy in Canada')
fig.show()
```

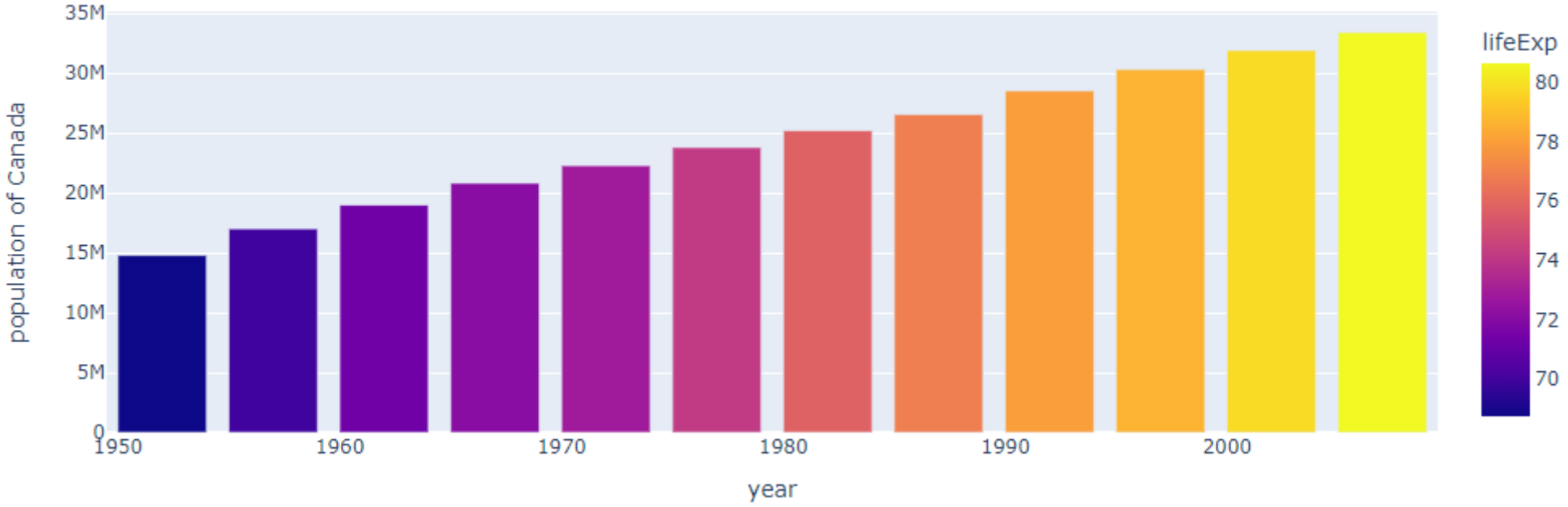
Rezultat przykładu numer 1



Przykład numer 2 – wykres słupkowy

```
import plotly.express as px

df = px.data.gapminder().query("country == 'Canada'")
fig = px.bar(df, x='year', y='pop',
             hover_data=['lifeExp', 'gdpPercap'], color='lifeExp',
             labels={'pop': 'population of Canada'}, height=400)
fig.show()
```

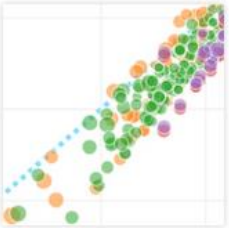
Rezultat przykładu numer 2



Różne typy wykresów w Plotly

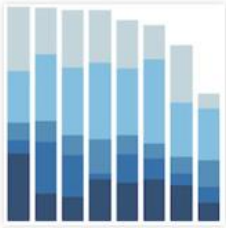
Basic Charts

More Basic Charts »

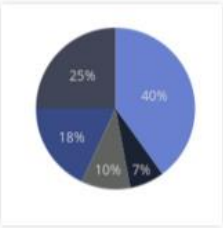


Scatter Plots

Line Charts



Bar Charts

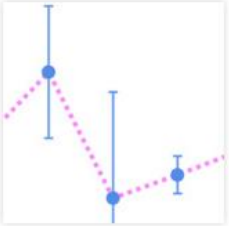


Pie Charts

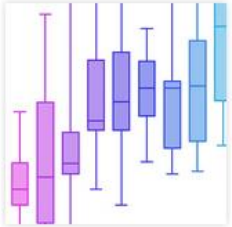
Bubble Charts

Statistical Charts

More Statistical Charts »

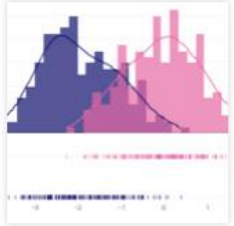


Error Bars

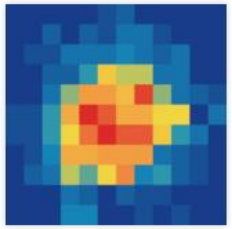


Box Plots

Histograms



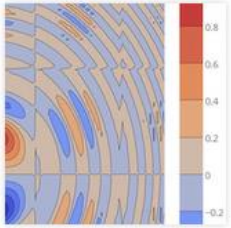
Distplots



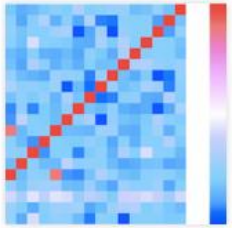
2D Histograms

Scientific Charts

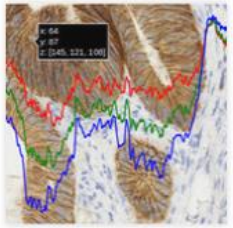
More Scientific Charts »



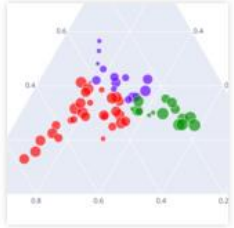
Contour Plots



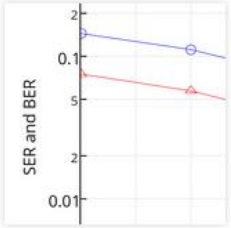
Heatmaps



Imshow



Ternary Plots

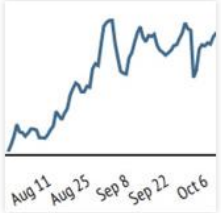


Log Plots

Różne typy wykresów w Plotly

Financial Charts

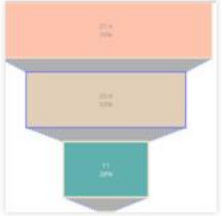
More Financial Charts »



Time Series and Date Axes

Candlestick Charts

Waterfall Charts



Funnel Chart

OHLC Charts

Maps

More Maps »

Mapbox Choropleth Maps

Lines on Mapbox

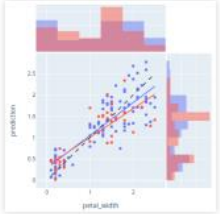
Filled Area on Maps

Bubble Maps

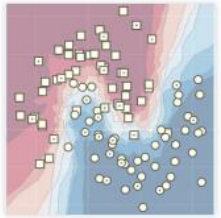
Mapbox Density Heatmap

Artificial Intelligence and Machine Learning

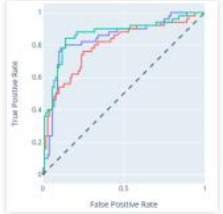
More AI and ML »



ML Regression



kNN Classification



ROC and PR Curves

PCA Visualization

AI/ML Apps with Dash

Dash

Dash is the original low-code framework for rapidly building data apps in Python, R, Julia, and F# (experimental).

Written on top of Plotly.js and React.js, Dash is ideal for building and deploying data apps with customized user interfaces. It's particularly suited for anyone who works with data.

plotly | Dash

Flask
web development,
one drop at a time

Przykład programu z użyciem Dash

```
# visit http://127.0.0.1:8050/ in your web browser.

from dash import Dash, html, dcc
import plotly.express as px
import pandas as pd

app = Dash(__name__)

# assume you have a "long-form" data frame
# see https://plotly.com/python/px-arguments/ for more options
df = pd.DataFrame({
    "Fruit": ["Apples", "Oranges", "Bananas", "Apples", "Oranges", "Bananas"],
    "Amount": [4, 1, 2, 2, 4, 5],
    "City": ["SF", "SF", "SF", "Montreal", "Montreal", "Montreal"]
})
```

Przykład programu z użyciem Dash

```
fig = px.bar(df, x="Fruit", y="Amount", color="City", barmode="group")

app.layout = html.Div(children=[
    html.H1(children='Hello Dash'),

    html.Div(children='''
        Dash: A web application framework for your data.
    '''),

    dcc.Graph(
        id='example-graph',
        figure=fig
    )
])
```

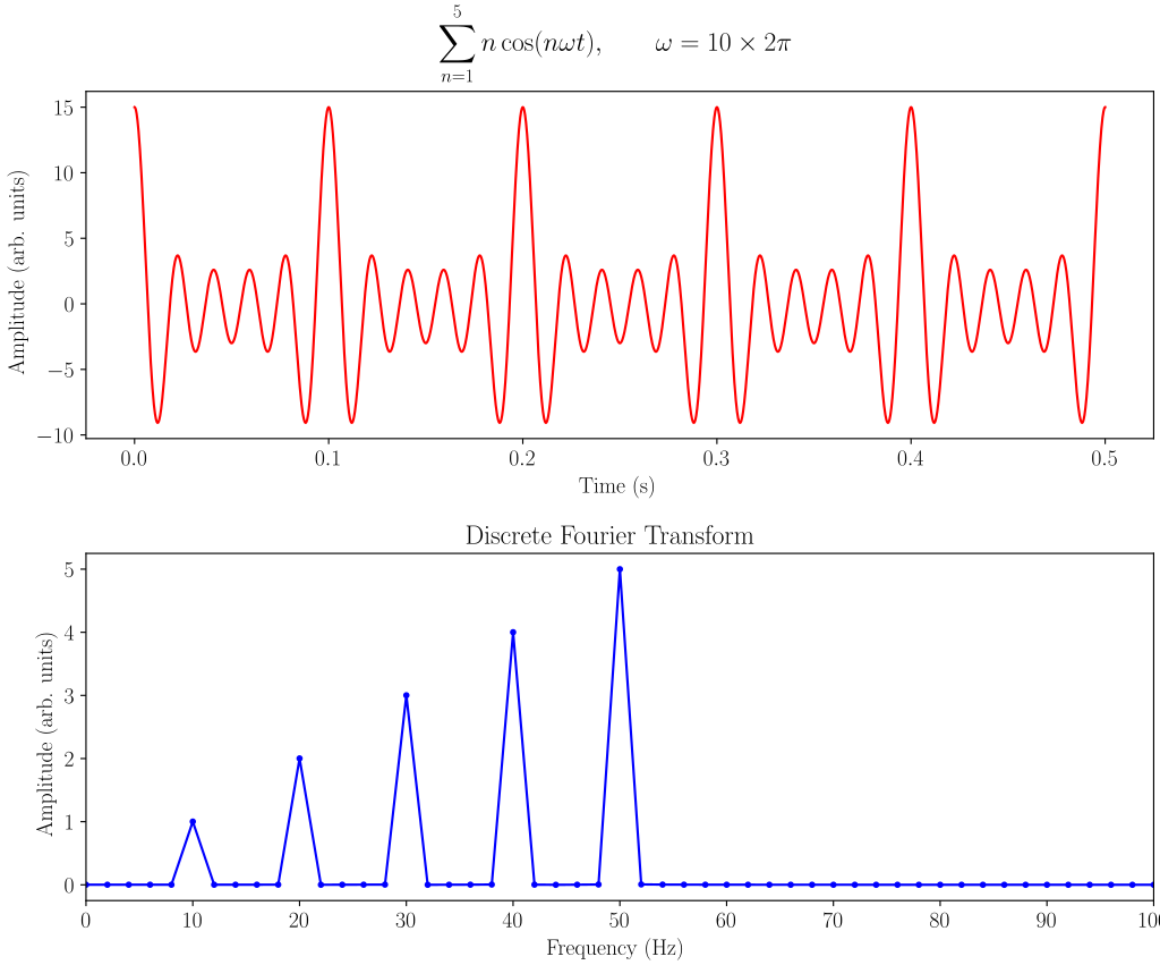
Przykład użycia callback

```
from dash import Dash, dcc, html, Input, Output

app = Dash(__name__)

app.layout = html.Div([
    html.H6("Change the value in the text box to see callbacks in action!"),
    html.Div([
        "Input: ",
        dcc.Input(id='my-input', value='initial value', type='text')
    ]),
    html.Br(),
    html.Div(id='my-output'),

])
```

Przykład użycia callback

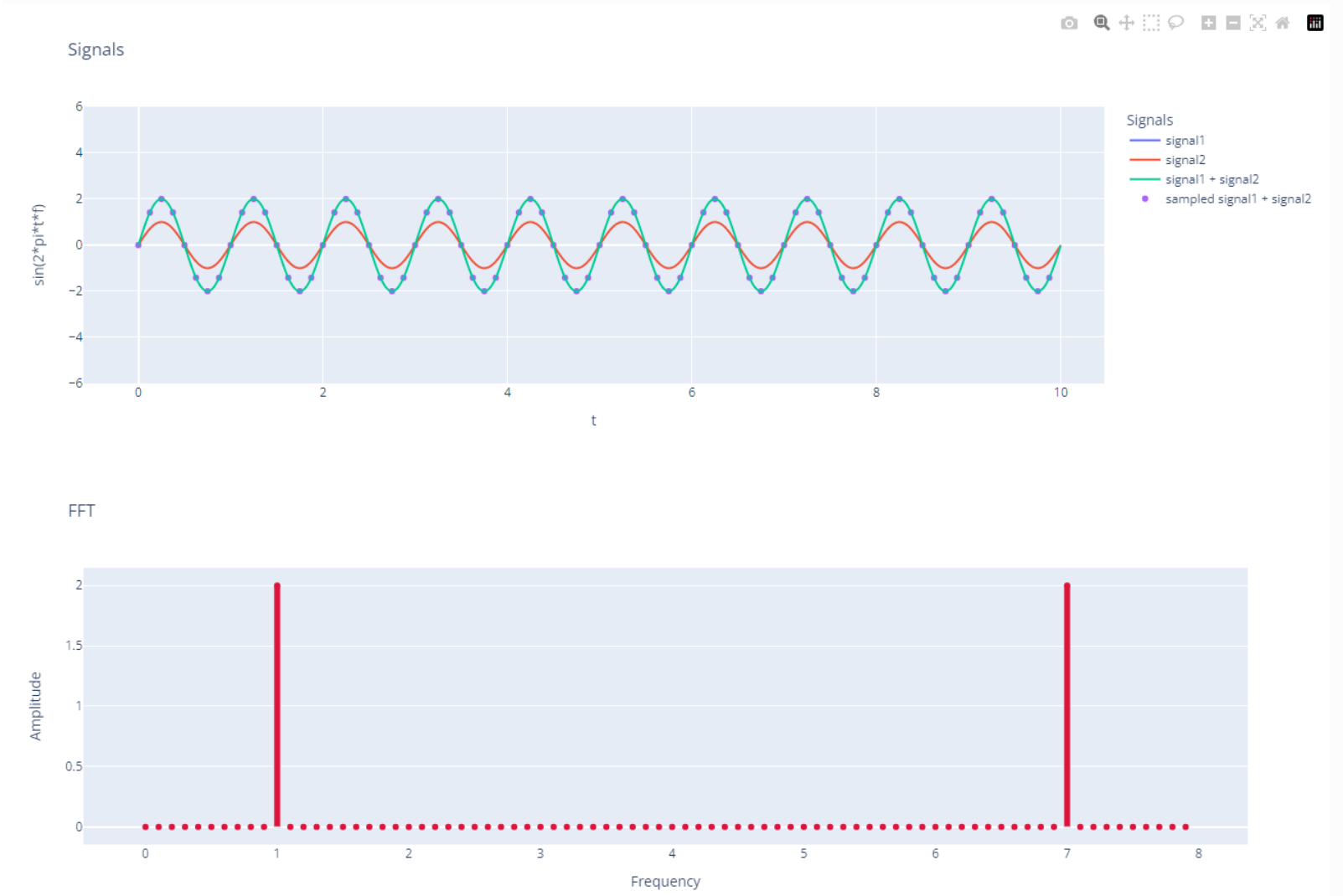
```
@app.callback(
    Output(component_id='my-output', component_property='children'),
    Input(component_id='my-input', component_property='value')
)
def update_output_div(input_value):
    return f'Output: {input_value}'


if __name__ == '__main__':
    app.run_server(debug=True)
```

FFT

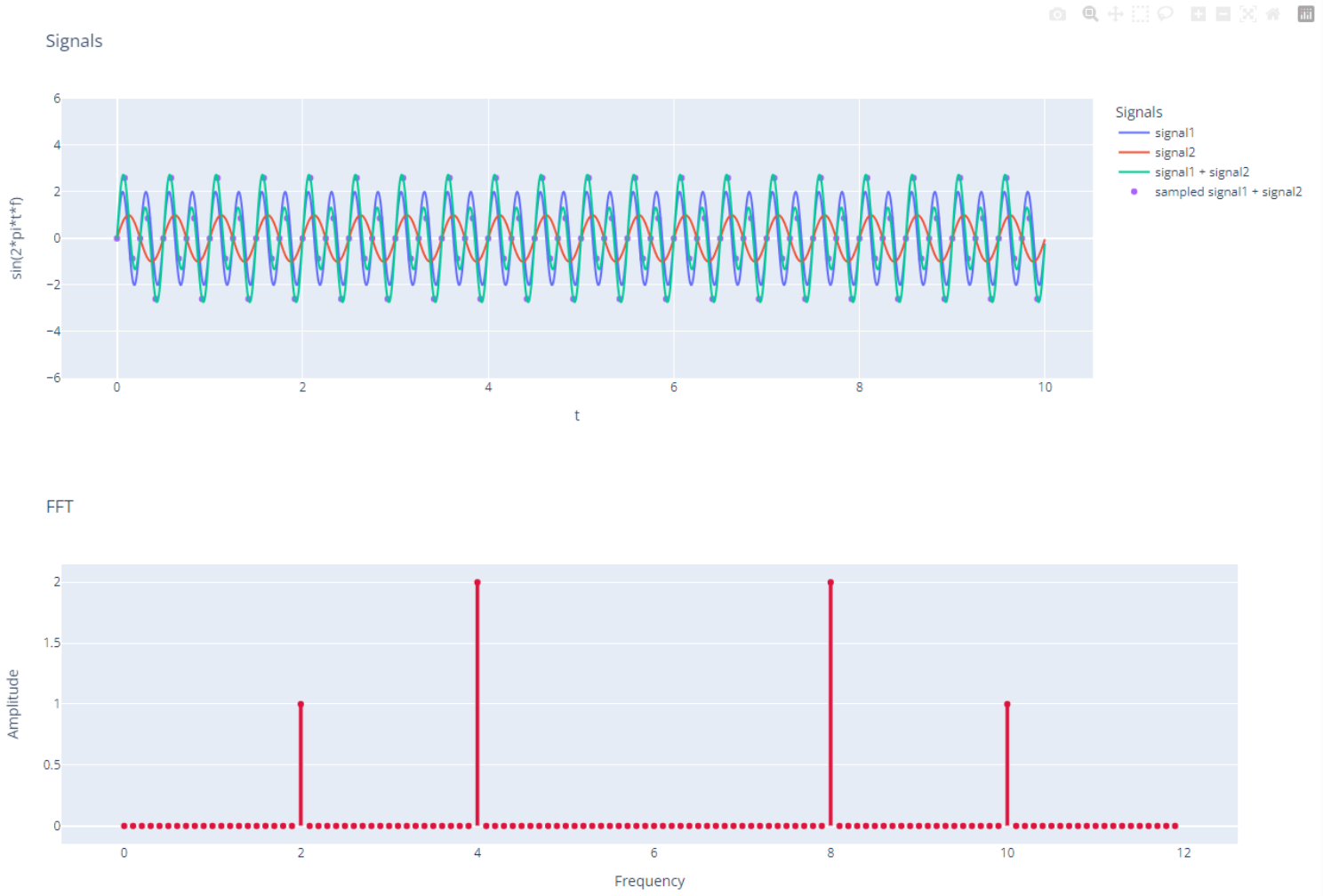


Language	Command/Method	Pre-requisites
R	stats::fft(x)	None
Octave/MATLAB	fft(x)	None
Python	fft.fft(x)	numpy scipy
Mathematica	Fourier[x]	None
Fortran	fftw_one(plan,in,out)	FFTW
Julia	fft(A [,dims])	FFTW
Rust	fft.process(&mut x);	rustfft
Haskell	dft x	fft

Wizualizacja FFT z użyciem Dash Plotly



Wizualizacja FFT z użyciem Dash Plotly



Podsumowanie

- Wizualizacja danych pozwala na lepsze zrozumienie otaczającego nas świata
- Dash Plotly umożliwia interaktywną wizualizację zagadnień cyfrowego przetwarzania sygnałów, przykładowo wizualizację Dyskretnej Transformacji Fouriera

Literatura

- https://www.python.org/
- https://matplotlib.org/
- https://dash.plotly.com/
- https://www.youtube.com/@CharmingData

Warto zobaczyć również:

- https://dash.gallery/Portal/

Materiały

Materiały z prezentacji dostępne są w repozytorium GitHub:
https://github.com/anras5/FFT-with-Dash

Aplikacja z wizualizacją FFT dostępna jest pod linkiem:
https://fft-with-dash.onrender.com/