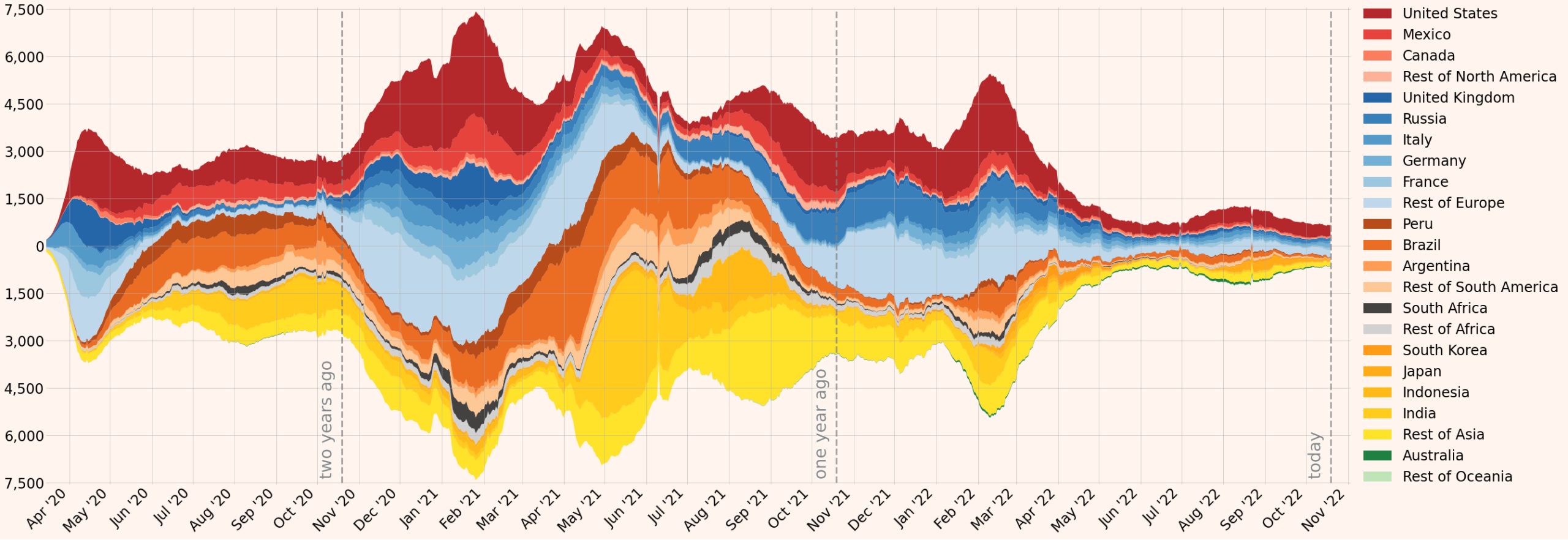
Data Visualization in Python

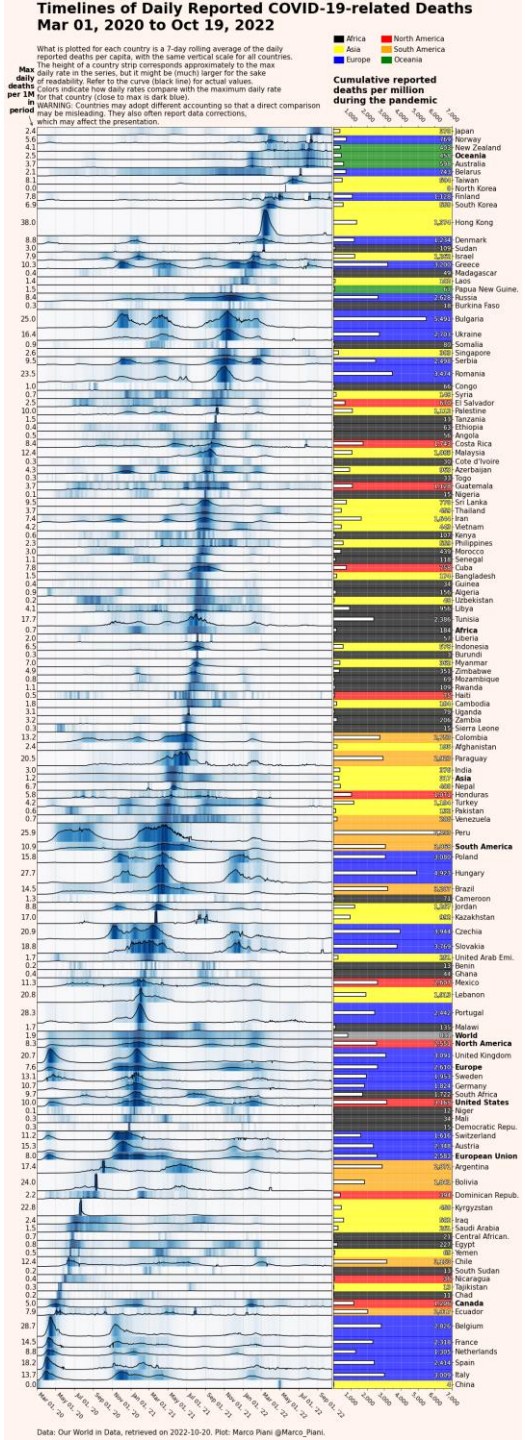
Marco Piani

marcopiani@gmail.com

Daily COVID-19 Reported Deaths by Global Region

7-day rolling average of reported deaths, with reporting criteria and reliability which may vary from country to country. Data are not available for some countries for the last few days.
Data: Our World in Data, as of Oct 20, 2022. Plot by @Marco_Piani





Timelines of Daily Reported COVID-19-related Deaths
Mar 01, 2020 to Oct 19, 2022

What is plotted for each country is a 7-day rolling average of the daily reported deaths per capita, with the same vertical scale for all countries. The height of a country strip corresponds approximately to the max daily rate in the series, but it might be (much) larger for the sake of readability. Refer to the curve (black line) for actual values. Colors indicate how daily rates compare with the maximum daily rate for that country (close to max is dark blue). WARNING: Countries may adopt different accounting so that a direct comparison may be misleading. They also often report data corrections, which may affect the presentation.

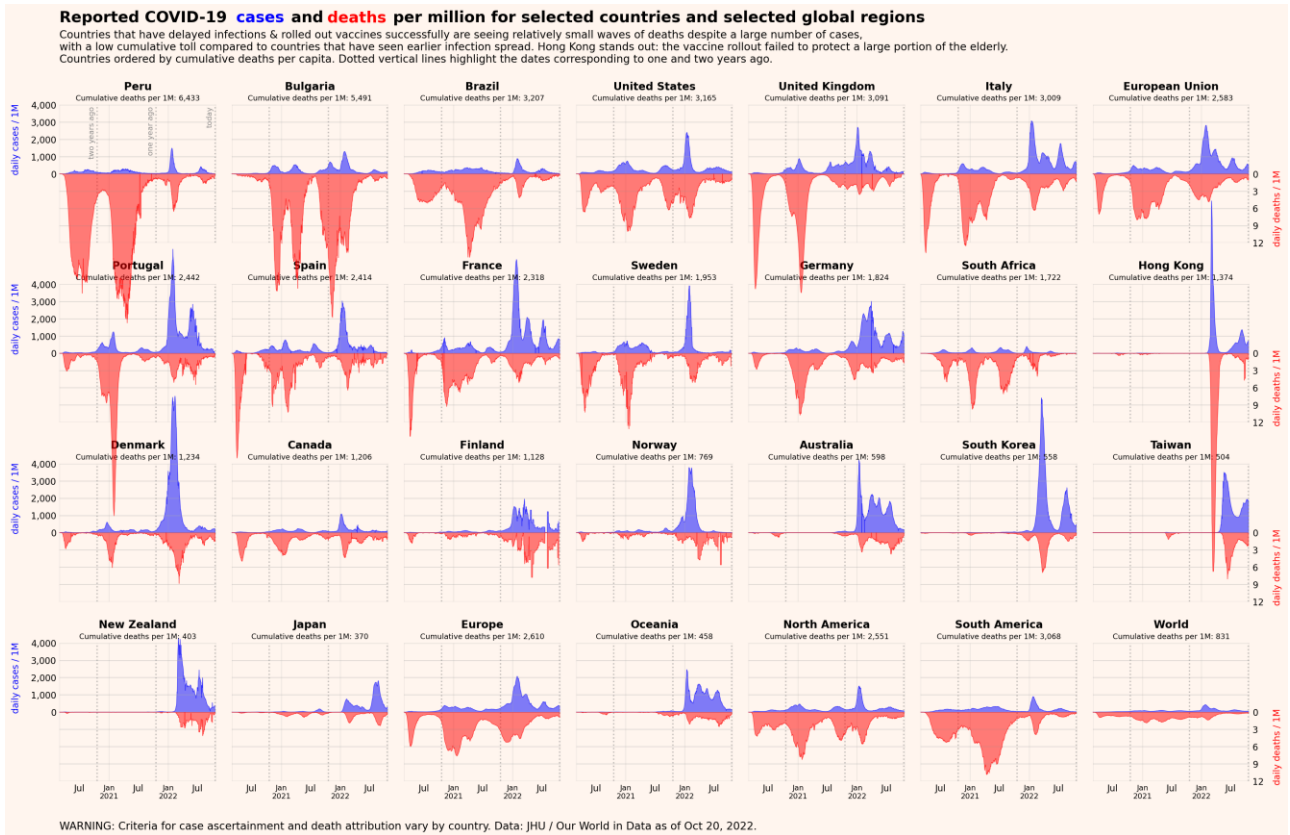
Africa · North America
Asia · South America
Europe · Oceania

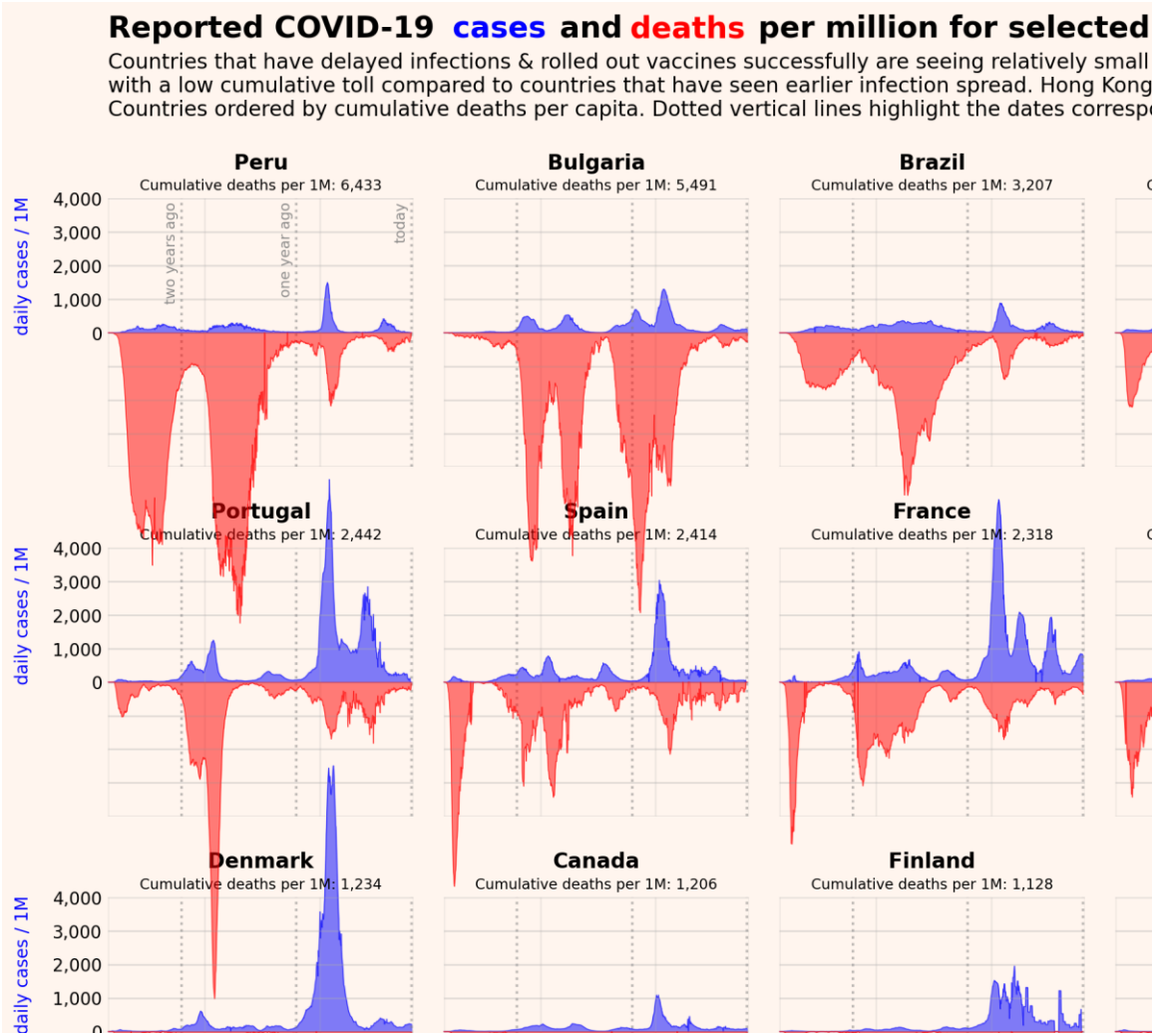
Max daily deaths per 1M in period

Cumulative reported deaths per million during the pandemic

1,000 2,000 3,000 4,000 5,000 6,000 7,000

Max daily deaths per 1M in period	Cumulative reported deaths per million during the pandemic
2.4	370 Japan
5.6	769 Norway
4.1	403 New Zealand
2.5	458 **Oceania**
3.7	598 Australia
2.1	743 Belarus
8.1	504 Taiwan
0.0	0 North Korea
7.8	1,128 Finland
6.9	558 South Korea
38.0	1,374 Hong Kong
8.8	1,234 Denmark
3.0	109 Sudan
7.9	1,262 Israel
10.3	3,200 Greece
0.4	49 Madagascar
1.4	102 Laos
1.5	67 Papua New Guine.
8.4	2,628 Russia
0.3	18 Burkina Faso
25.0	5,491 Bulgaria
16.4	2,703 Ukraine
0.9	80 Somalia
2.6	303 Singapore
9.5	2,498 Serbia
23.5	3,474 Romania
1.0	66 Congo
0.7	148 Syria
2.5	670 El Salvador
10.0	1,112 Palestine
1.5	13 Tanzania
0.4	63 Ethiopia
0.5	56 Angola
8.4	1,743 Costa Rica
12.4	1,085 Malaysia
0.3	30 Cote d'Ivoire
4.3	963 Azerbaijan
0.3	33 Togo
3.7	1,128 Guatemala
0.1	15 Nigeria
9.5	770 Sri Lanka
3.7	459 Thailand
7.4	1,644 Iran
4.2	443 Vietnam

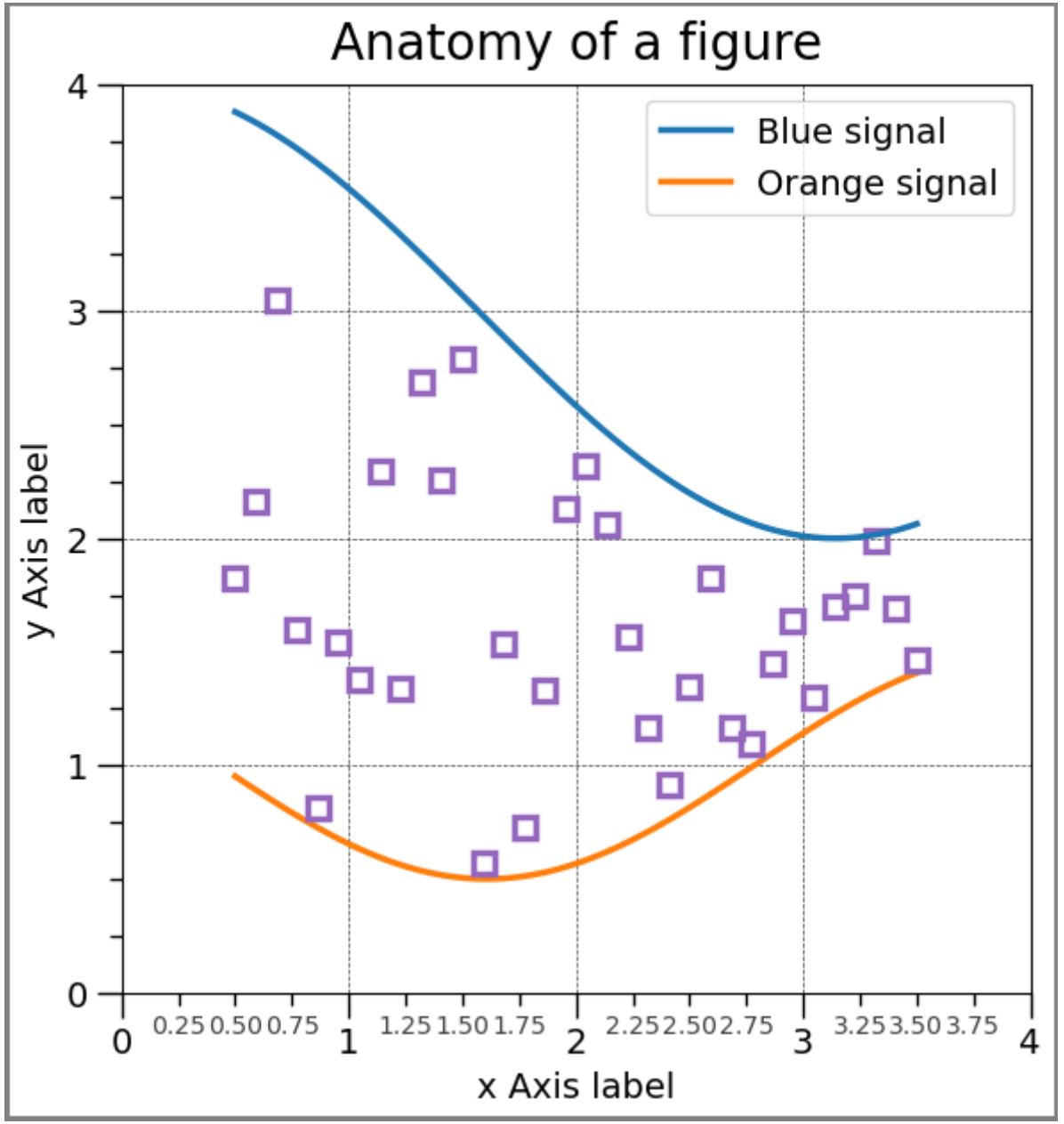

Reported COVID-19 cases and deaths per million for selected countries and selected global regions
Countries that have delayed infections & rolled out vaccines successfully are seeing relatively small waves of deaths despite a large number of cases,
with a low cumulative toll compared to countries that have seen earlier infection spread. Hong Kong stands out: the vaccine rollout failed to protect a large portion of the elderly.
Countries ordered by cumulative deaths per capita. Dotted vertical lines highlight the dates corresponding to one and two years ago.
Peru
Cumulative deaths per 1M: 6,433
Bulgaria
Cumulative deaths per 1M: 5,491
Brazil
Cumulative deaths per 1M: 3,207
United States
Cumulative deaths per 1M: 3,165
United Kingdom
Cumulative deaths per 1M: 3,091
Italy
Cumulative deaths per 1M: 3,009
European Union
Cumulative deaths per 1M: 2,583
Portugal
Cumulative deaths per 1M: 2,442
Spain
Cumulative deaths per 1M: 2,414
France
Cumulative deaths per 1M: 2,318
Sweden
Cumulative deaths per 1M: 1,953
Germany
Cumulative deaths per 1M: 1,824
South Africa
Cumulative deaths per 1M: 1,722
Hong Kong
Cumulative deaths per 1M: 1,374
Denmark
Cumulative deaths per 1M: 1,234
Canada
Cumulative deaths per 1M: 1,206
Finland
Cumulative deaths per 1M: 1,128
Norway
Cumulative deaths per 1M: 769
Australia
Cumulative deaths per 1M: 598
South Korea
Cumulative deaths per 1M: 558
Taiwan
Cumulative deaths per 1M: 504
New Zealand
Cumulative deaths per 1M: 403
Japan
Cumulative deaths per 1M: 370
Europe
Cumulative deaths per 1M: 2,610
Oceania
Cumulative deaths per 1M: 458
North America
Cumulative deaths per 1M: 2,551
South America
Cumulative deaths per 1M: 3,068
World
Cumulative deaths per 1M: 831
daily cases / 1M
daily deaths / 1M
two years ago
one year ago
today
Jul
Jan 2021
Jul
Jan 2022
Jul
WARNING: Criteria for case ascertainment and death attribution vary by country. Data: JHU / Our World in Data as of Oct 20, 2022.


Reported COVID-19 cases and deaths per million for selected
Countries that have delayed infections & rolled out vaccines successfully are seeing relatively small
with a low cumulative toll compared to countries that have seen earlier infection spread. Hong Kong
Countries ordered by cumulative deaths per capita. Dotted vertical lines highlight the dates corresp
Peru
Cumulative deaths per 1M: 6,433
Bulgaria
Cumulative deaths per 1M: 5,491
Brazil
Cumulative deaths per 1M: 3,207
Portugal
Cumulative deaths per 1M: 2,442
Spain
Cumulative deaths per 1M: 2,414
France
Cumulative deaths per 1M: 2,318
Denmark
Cumulative deaths per 1M: 1,234
Canada
Cumulative deaths per 1M: 1,206
Finland
Cumulative deaths per 1M: 1,128
daily cases / 1M
two years ago
one year ago
today

Topics

- Elements of a (Matplotlib) plot
- Matplotlib basics
- Pandas & Seaborn as higher-level options to generate plots
- Examples
  - Small multiples
  - Marginal distributions
- Conclusions

Elements of a (Matplotlib) plot


Anatomy of a figure
Blue signal
Orange signal
y Axis label
x Axis label
0
1
2
3
4
0.25
0.50
0.75
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

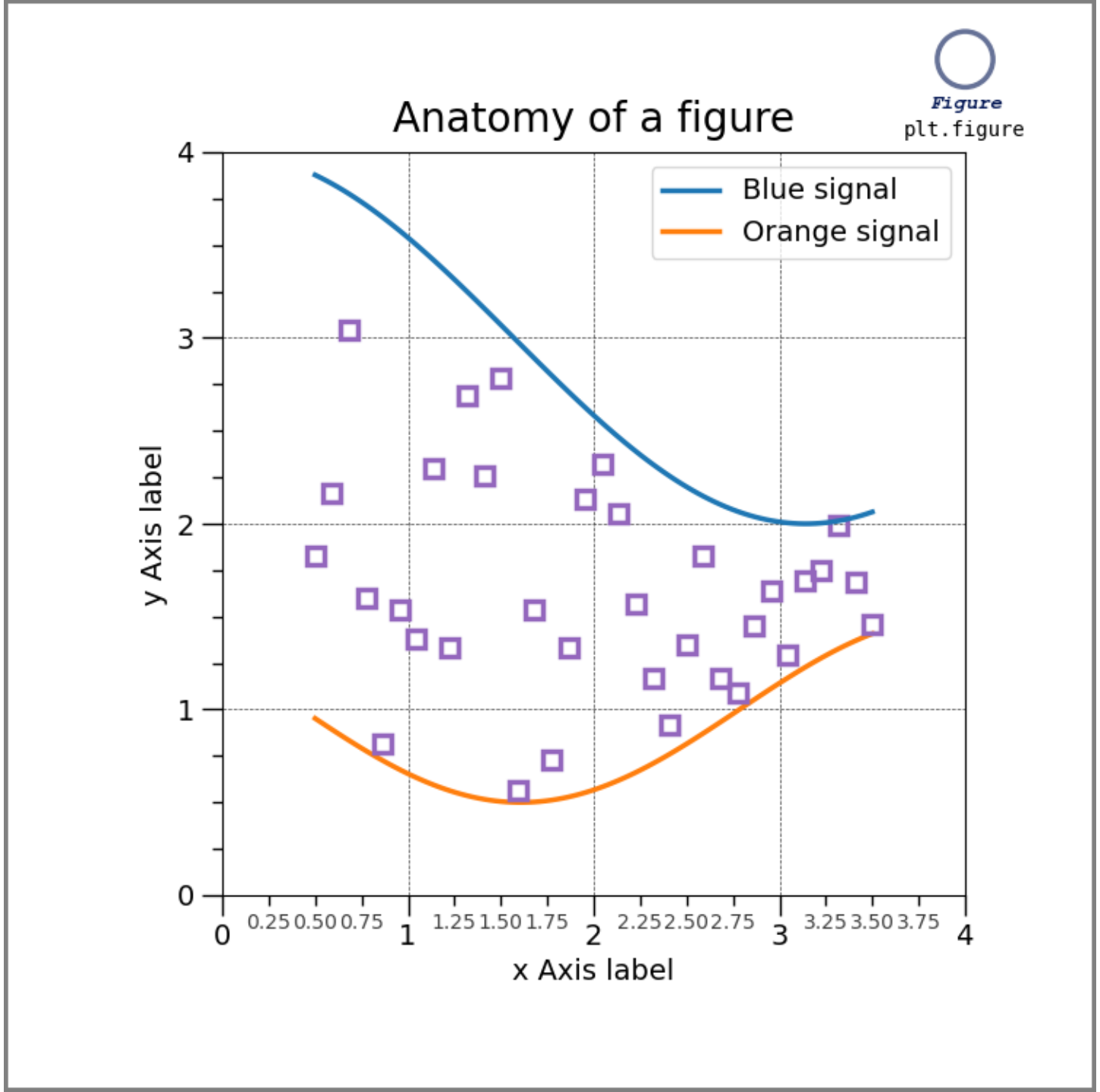
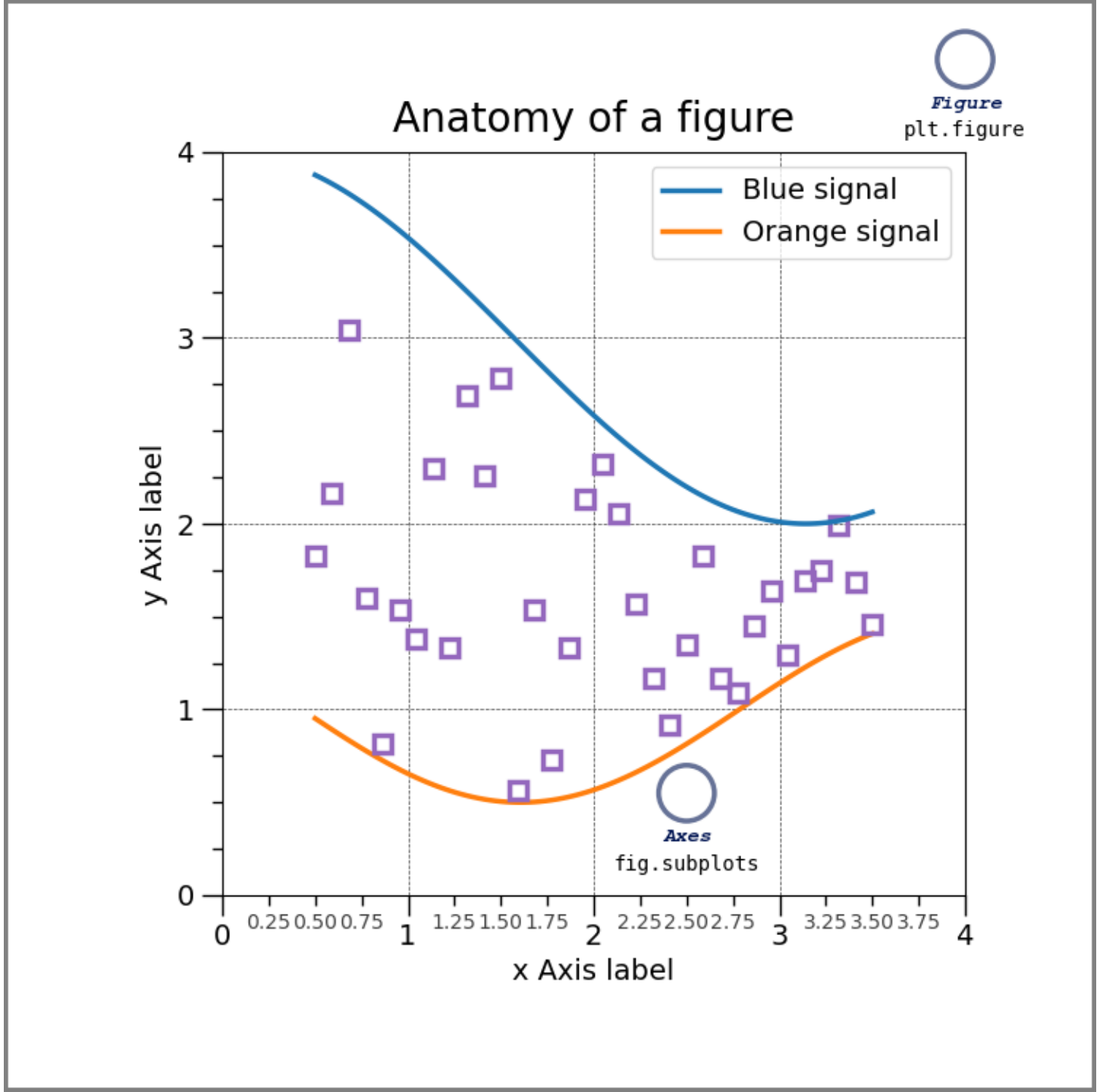
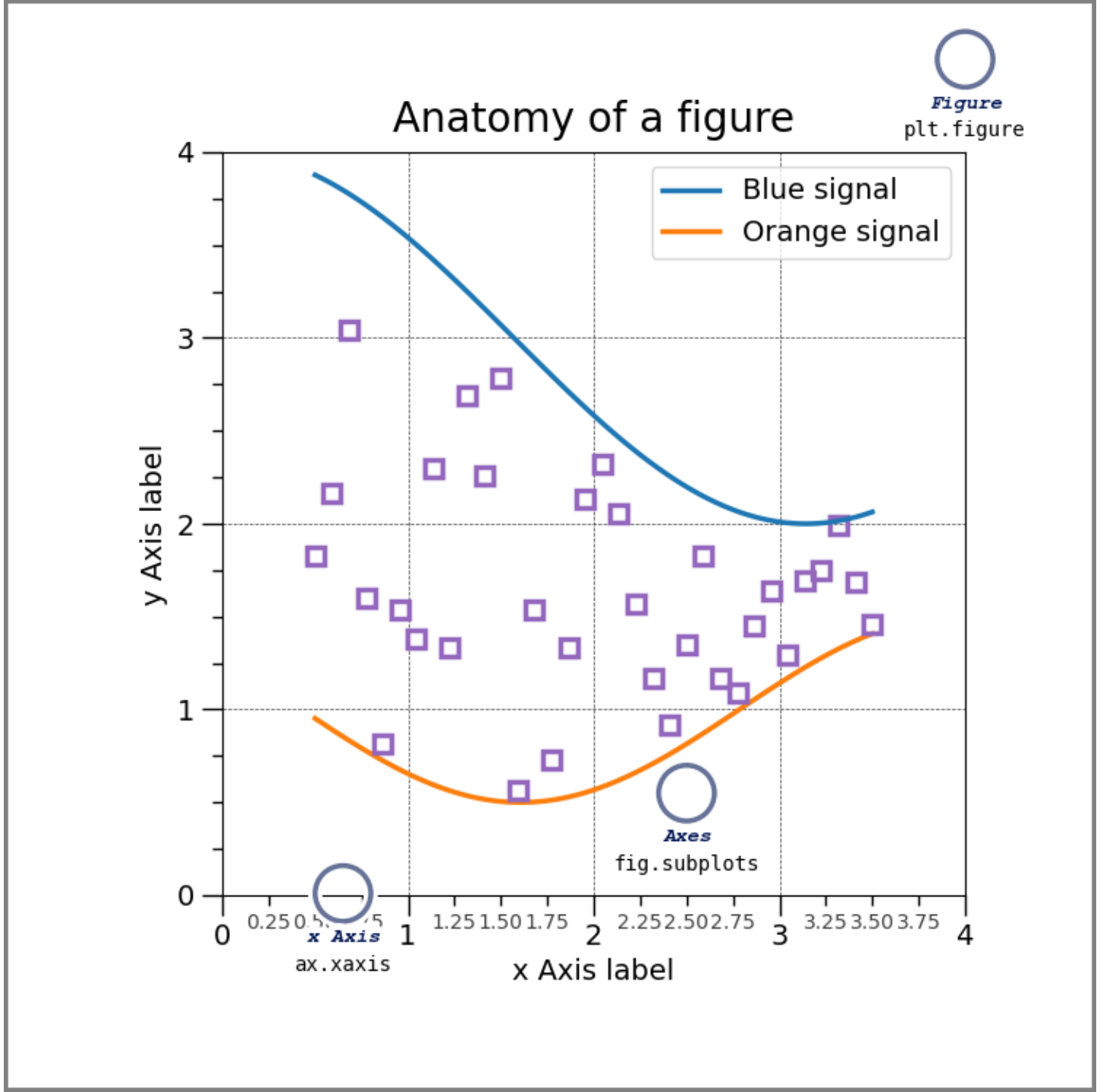
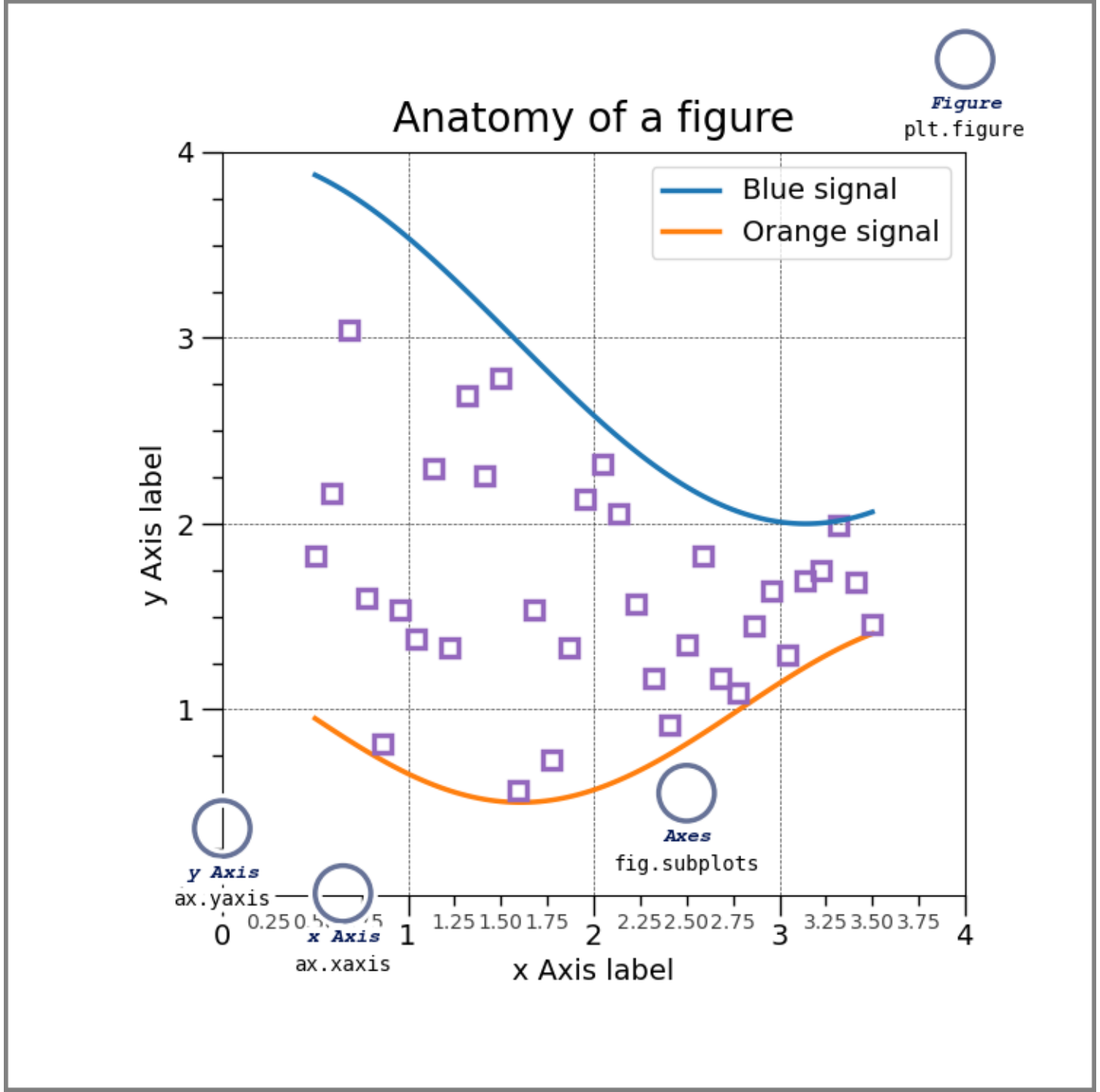
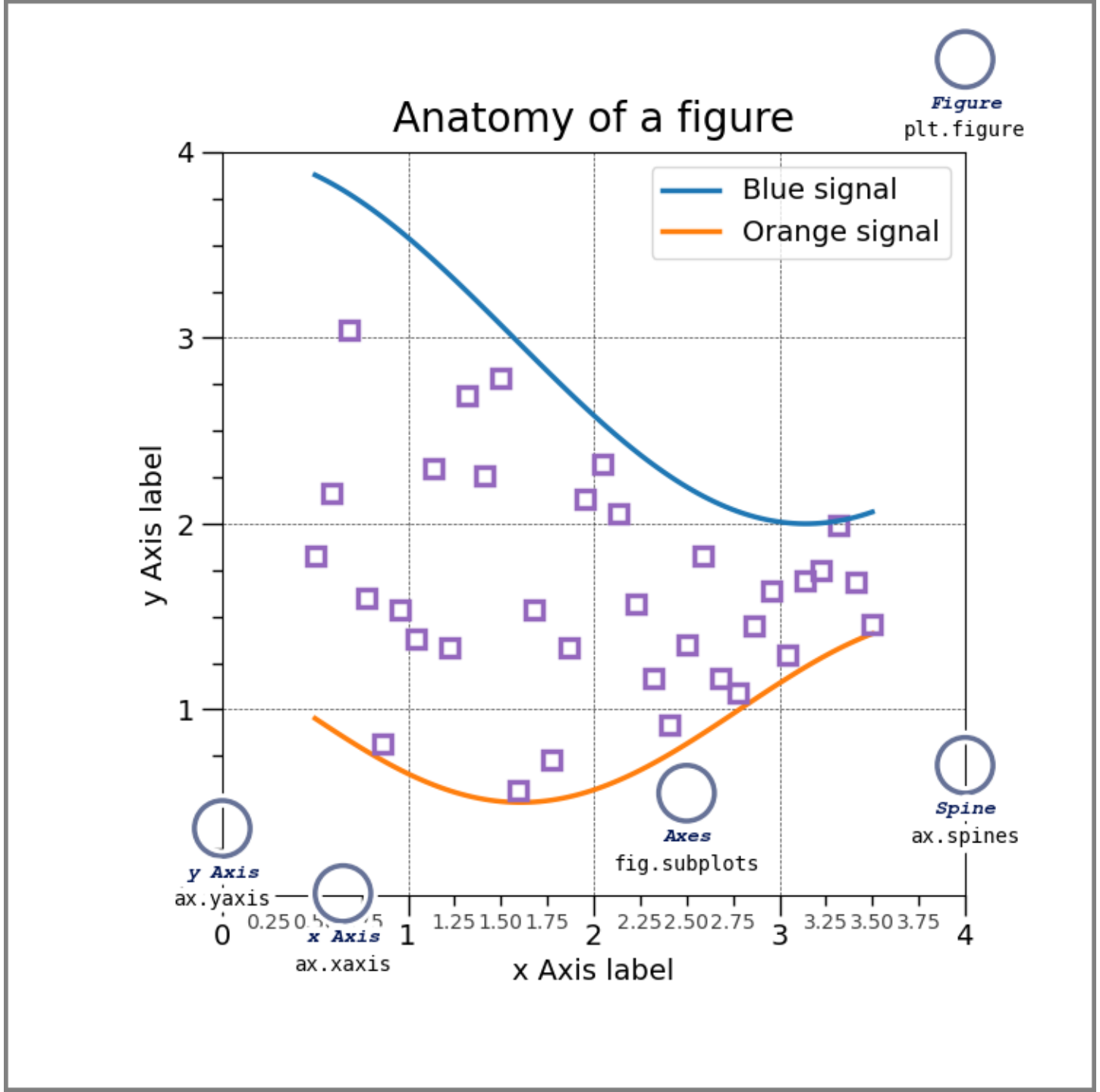


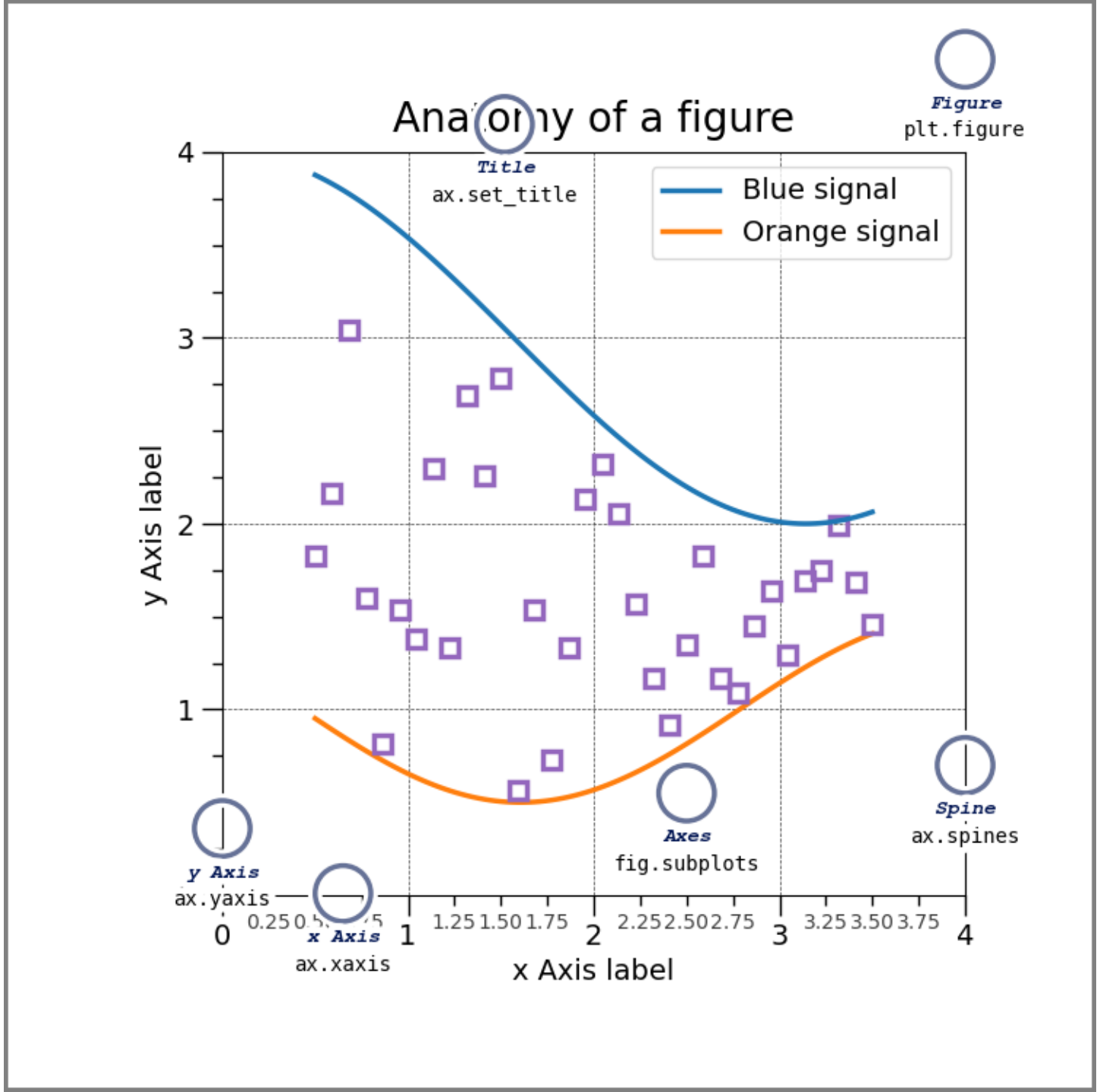
Figure
plt.figure
Anatomy of a figure
Blue signal
Orange signal
y Axis label
x Axis label
0
1
2
3
4
0.25
0.50
0.75
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

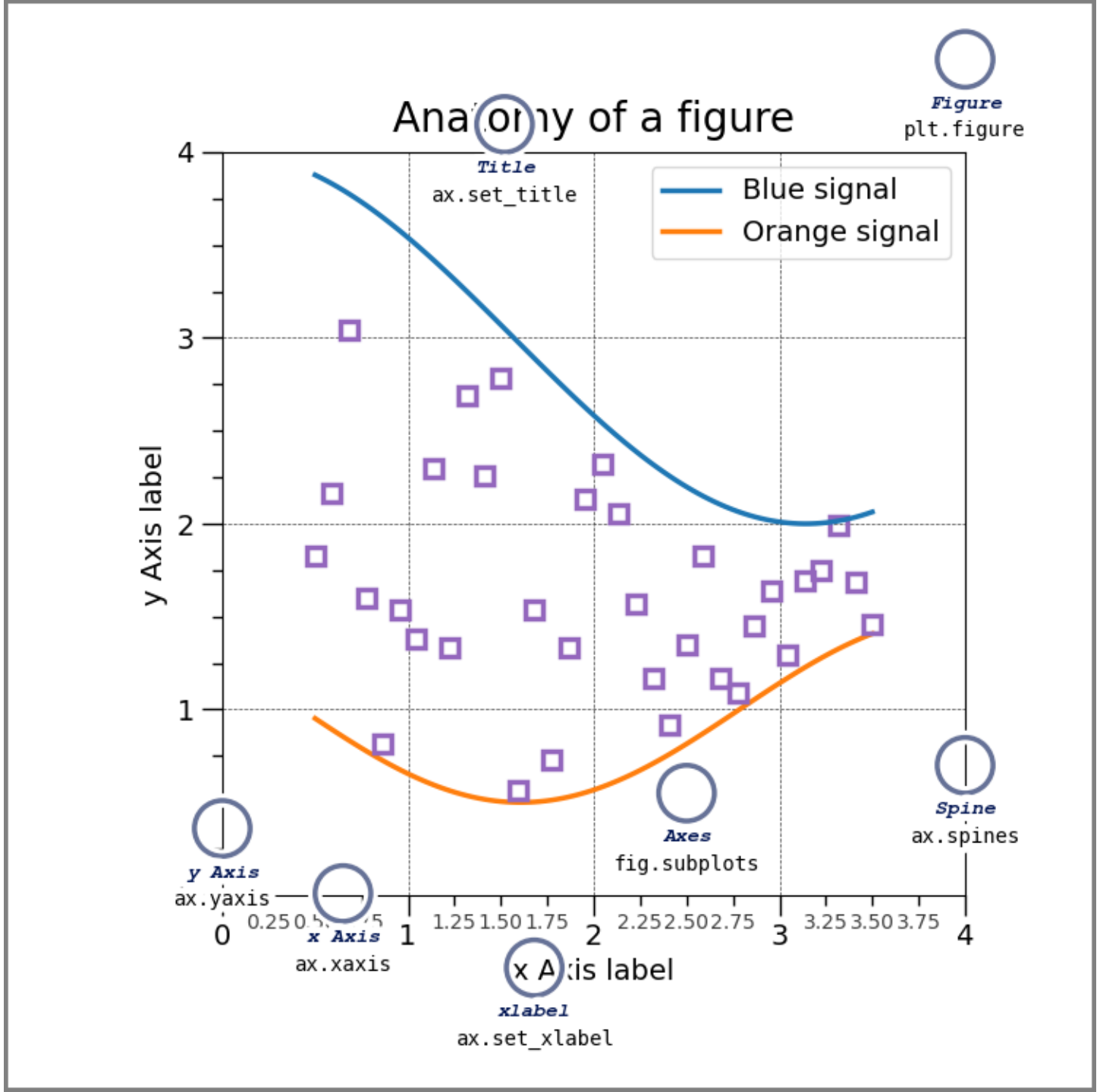

Anatomy of a figure
Figure
plt.figure
Blue signal
Orange signal
Axes
fig.subplots
y Axis label
x Axis label
0
1
2
3
4
0.25
0.50
0.75
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

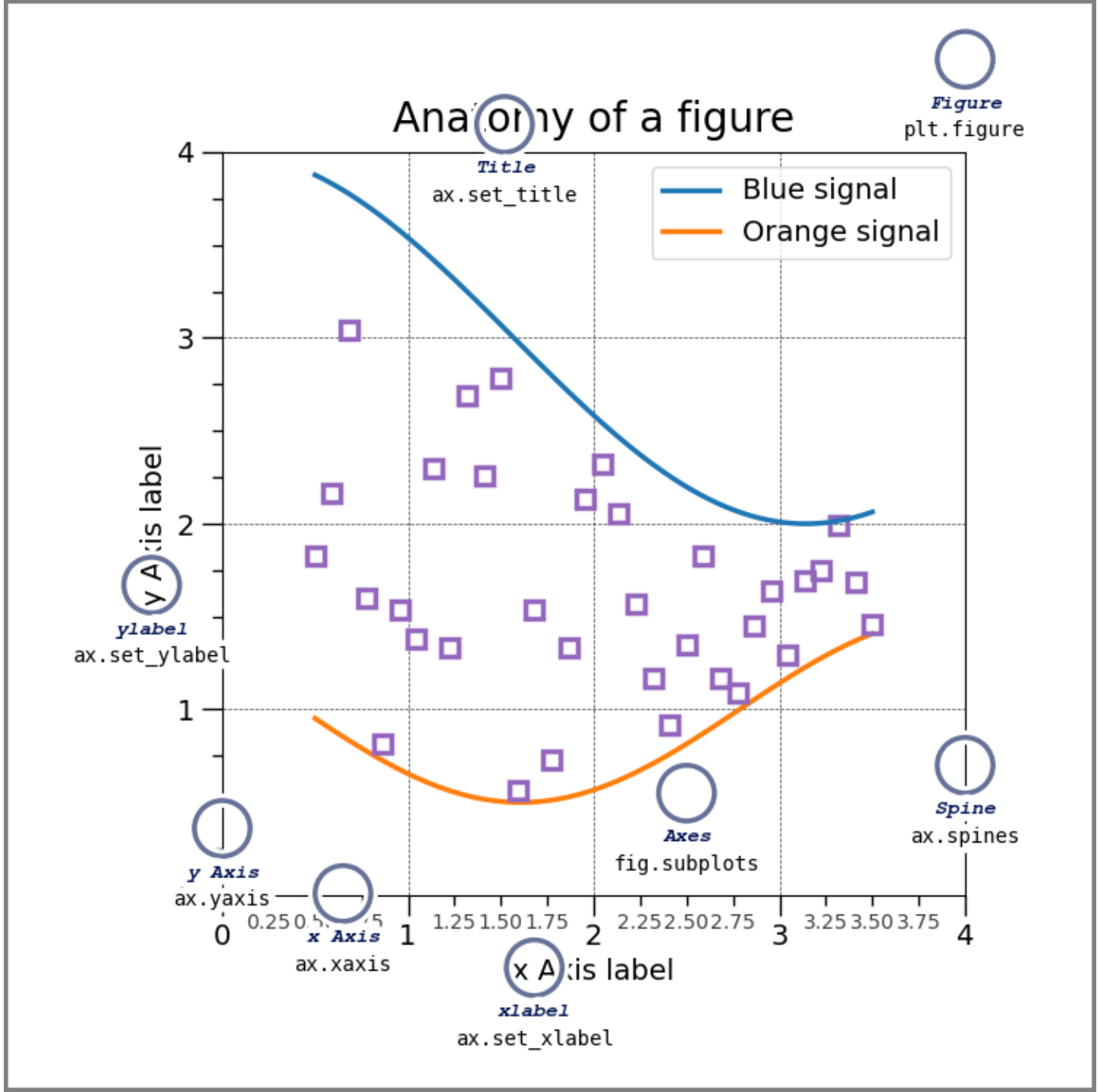

Anatomy of a figure
Figure
plt.figure
Blue signal
Orange signal
y Axis label
x Axis label
Axes
fig.subplots
x Axis
ax.xaxis
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

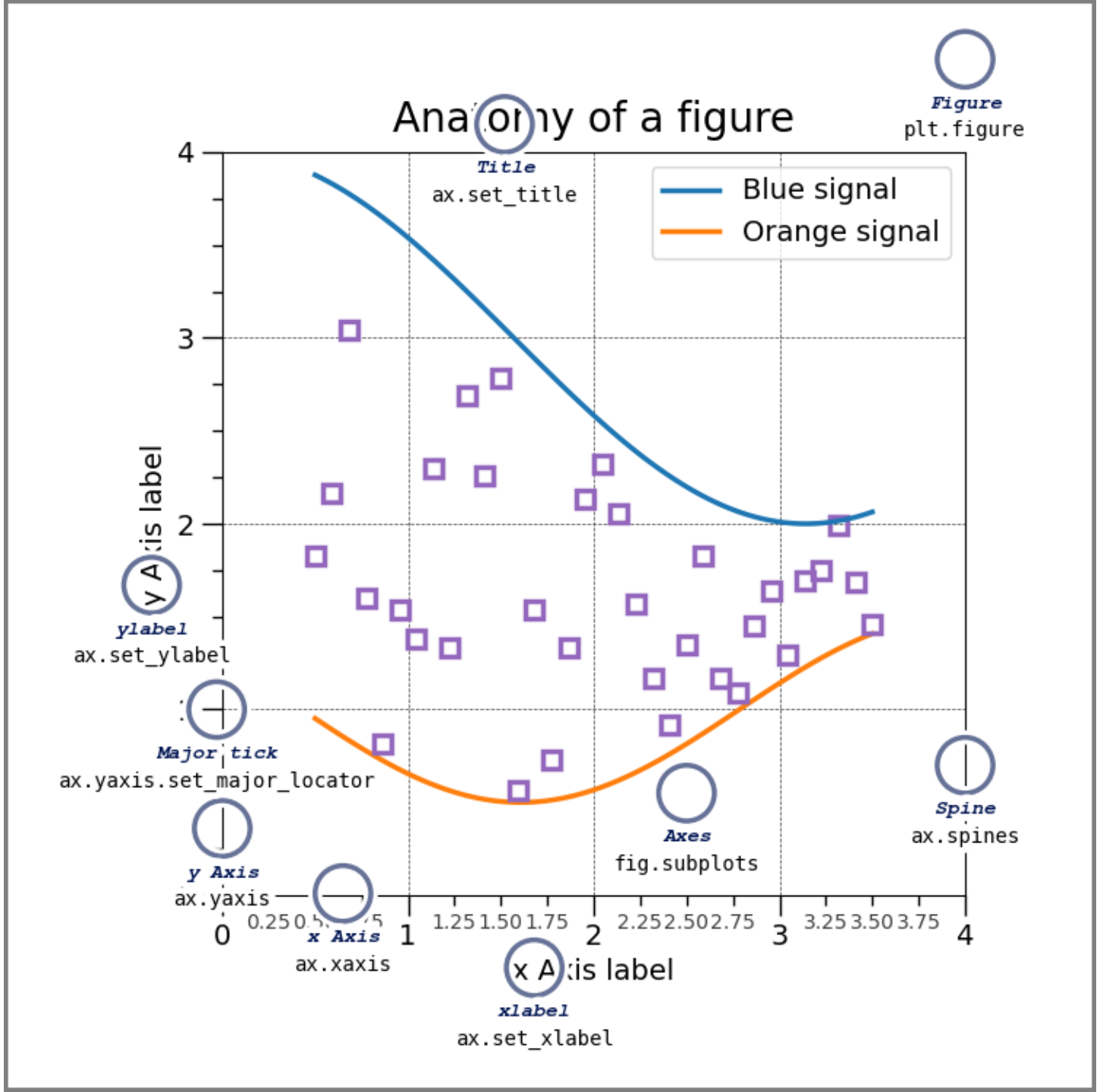

Anatomy of a figure
Figure
plt.figure
Blue signal
Orange signal
y Axis label
x Axis label
y Axis
ax.yaxis
x Axis
ax.xaxis
Axes
fig.subplots
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

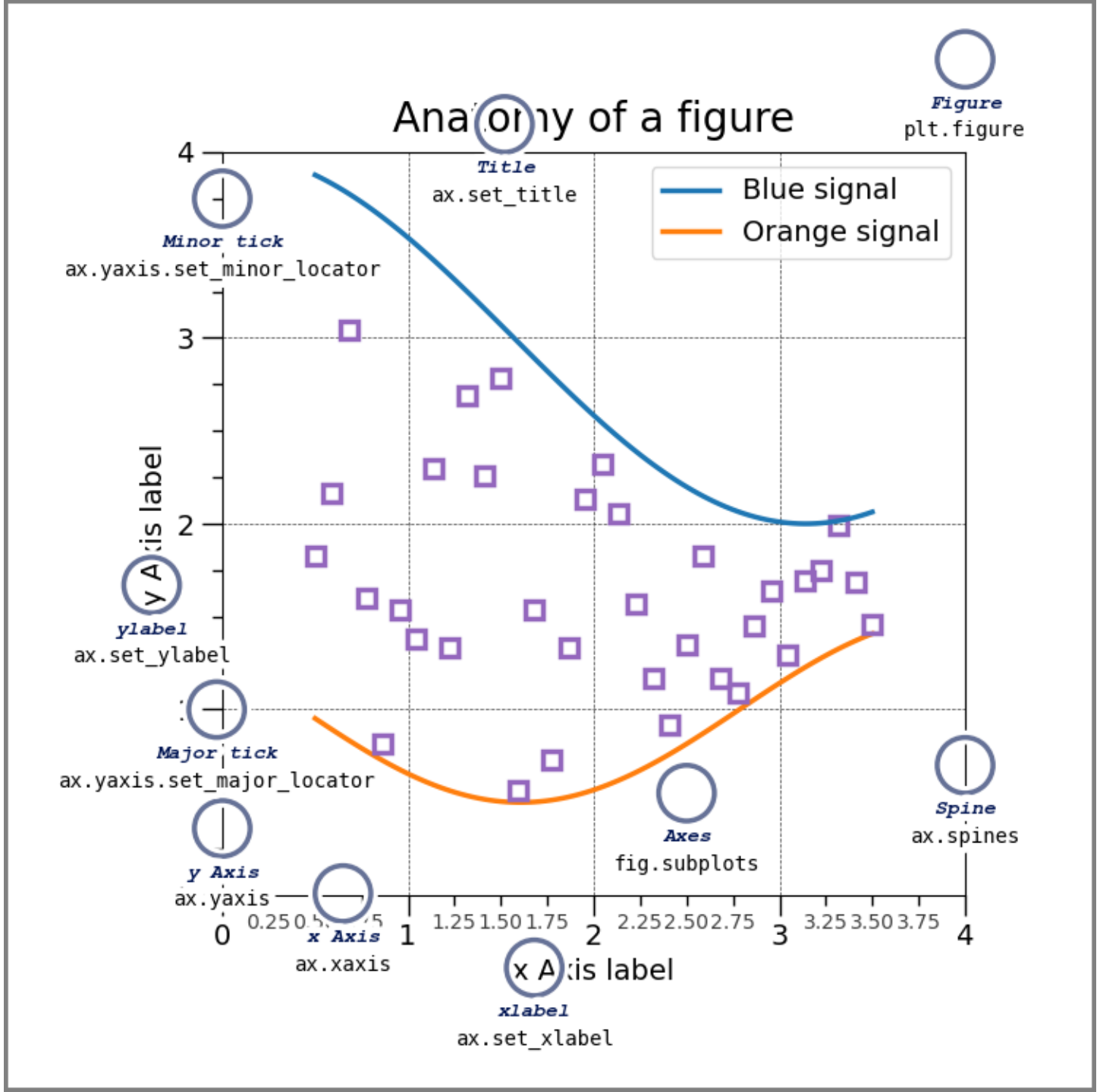

Anatomy of a figure
Figure
plt.figure
Blue signal
Orange signal
y Axis label
x Axis label
y Axis
ax.yaxis
x Axis
ax.xaxis
Axes
fig.subplots
Spine
ax.spines
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

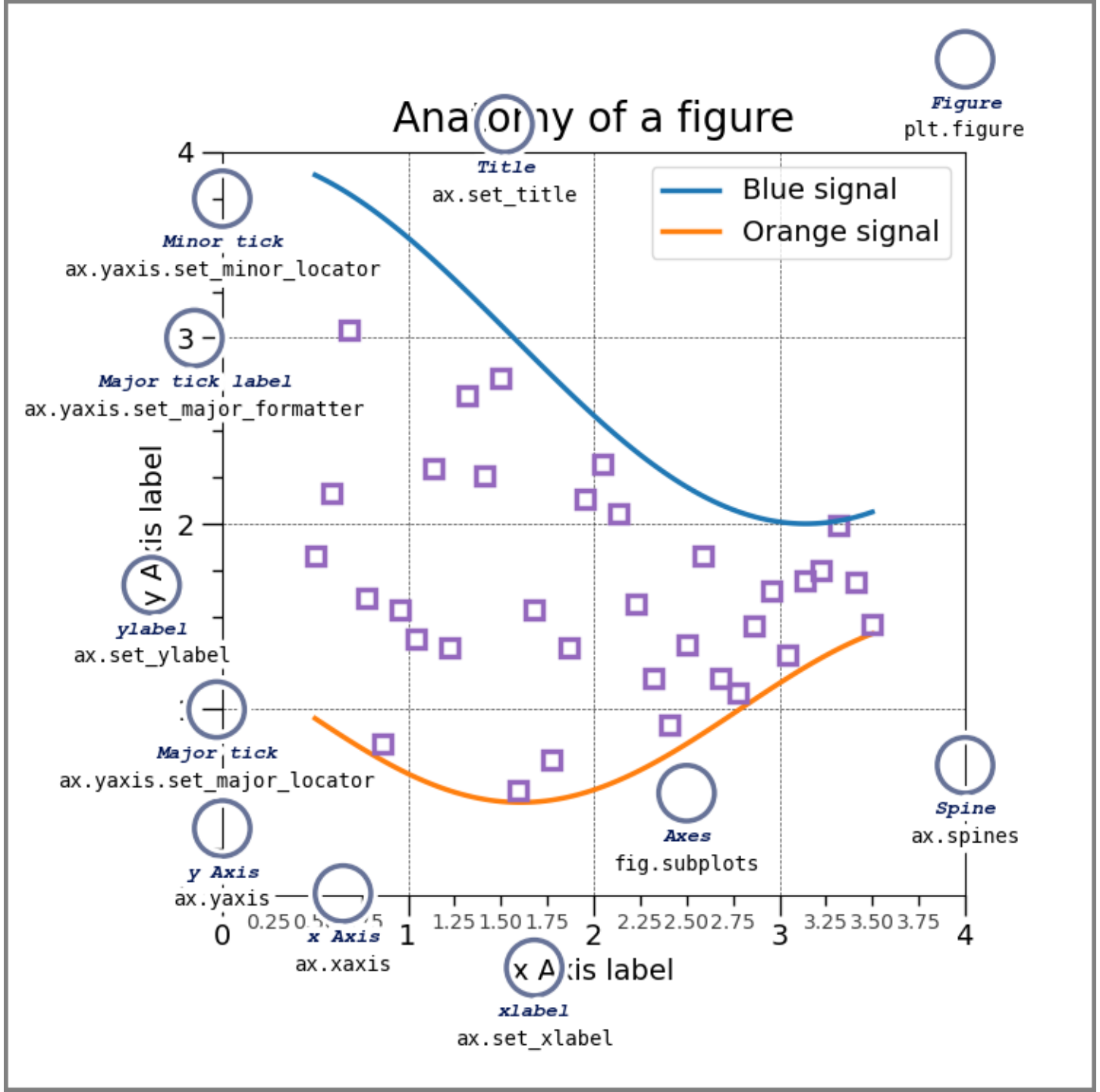

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Blue signal
Orange signal
y Axis label
x Axis label
y Axis
ax.yaxis
x Axis
ax.xaxis
Axes
fig.subplots
Spine
ax.spines
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

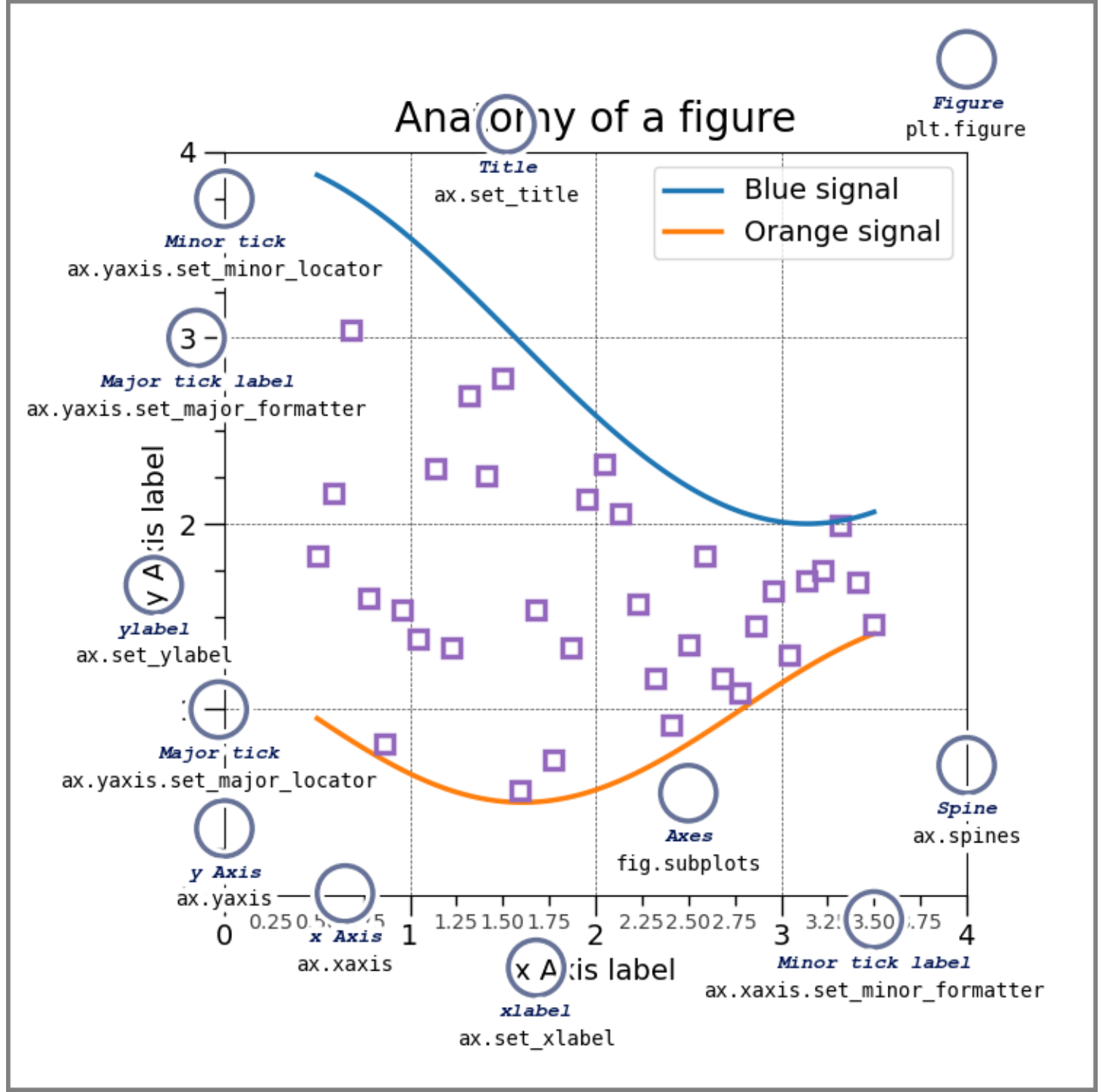

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Blue signal
Orange signal
y Axis label
x Axis label
y Axis
ax.yaxis
x Axis
ax.xaxis
Axes
fig.subplots
Spine
ax.spines
xlabel
ax.set_xlabel

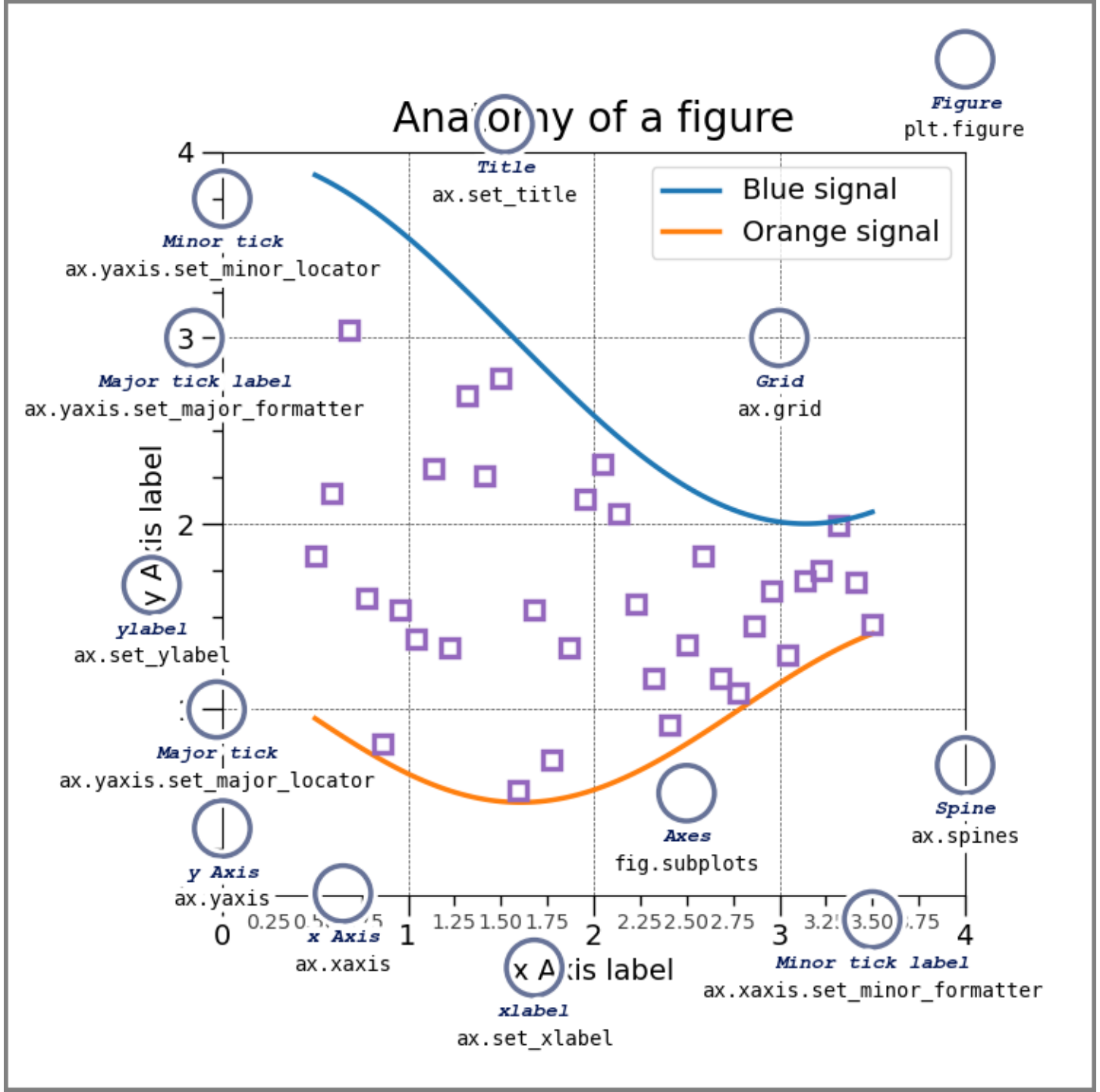

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Blue signal
Orange signal
y Axis label
ylabel
ax.set_ylabel
y Axis
ax.yaxis
x Axis
ax.xaxis
x Axis label
xlabel
ax.set_xlabel
Axes
fig.subplots
Spine
ax.spines
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

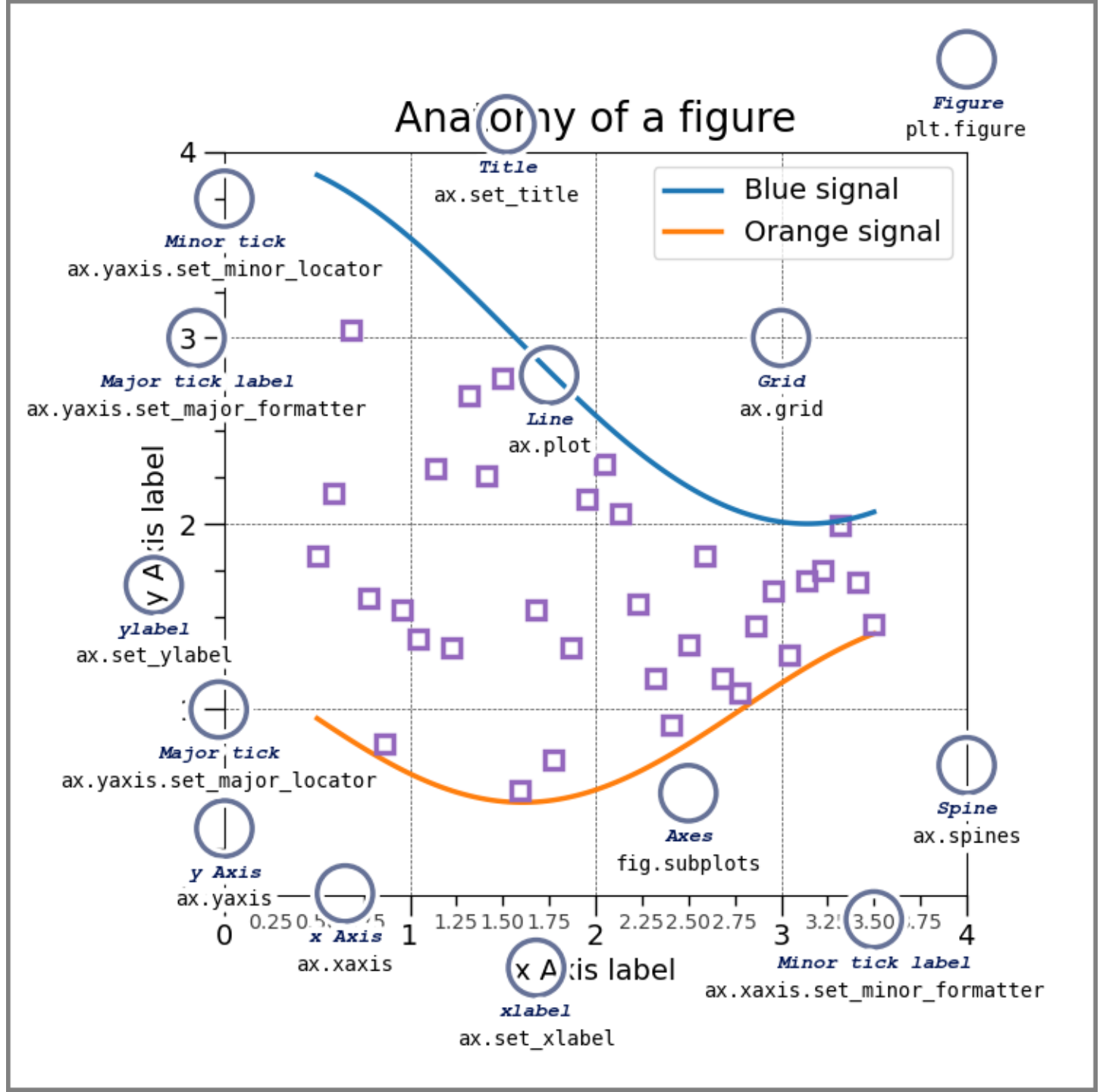

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Blue signal
Orange signal
y Axis label
ylabel
ax.set_ylabel
Major tick
ax.yaxis.set_major_locator
y Axis
ax.yaxis
x Axis
ax.xaxis
Axes
fig.subplots
Spine
ax.spines
x Axis label
xlabel
ax.set_xlabel
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

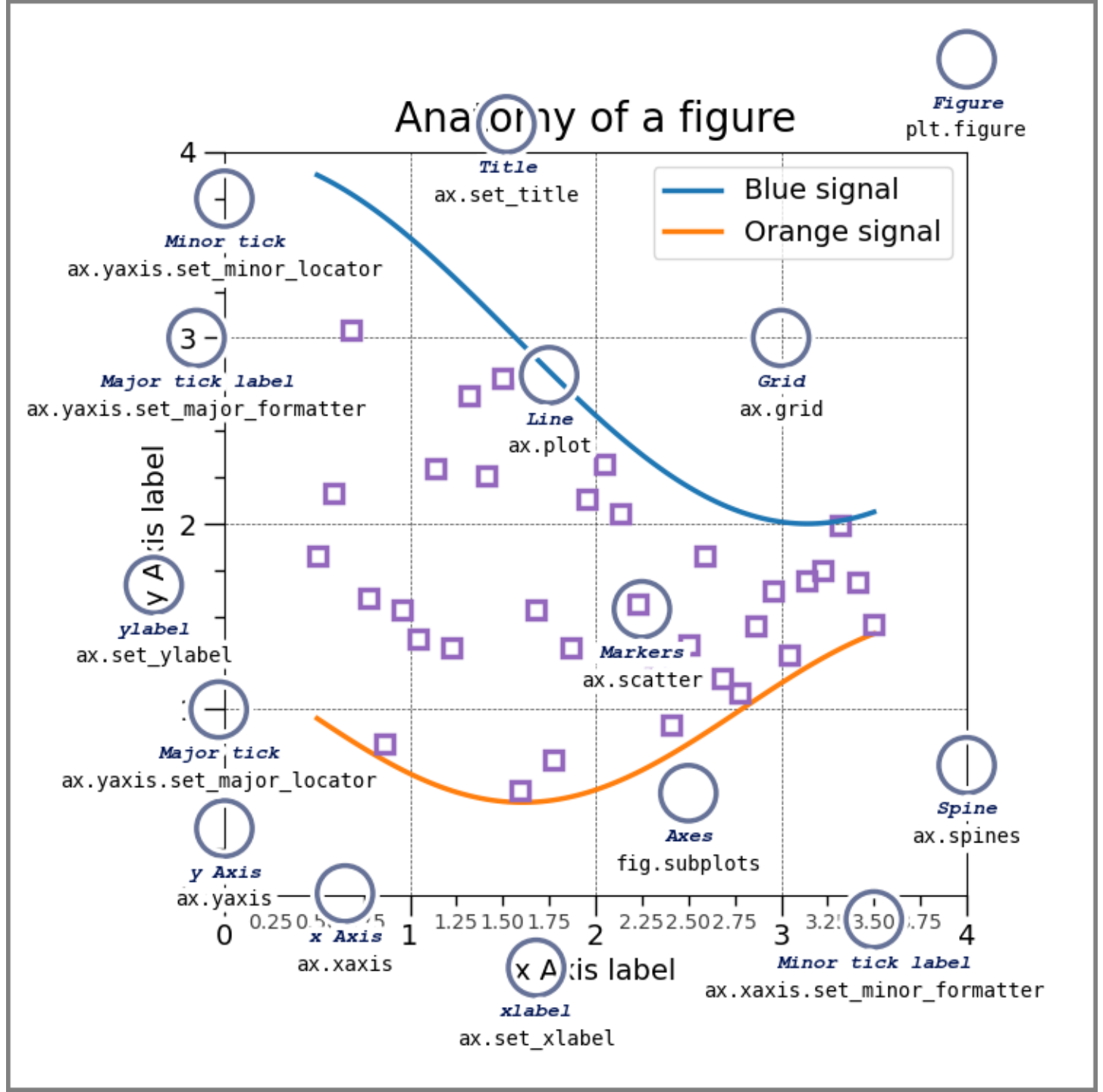

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Minor tick
ax.yaxis.set_minor_locator
Blue signal
Orange signal
y Axis label
ylabel
ax.set_ylabel
Major tick
ax.yaxis.set_major_locator
y Axis
ax.yaxis
Axes
fig.subplots
Spine
ax.spines
x Axis
ax.xaxis
x Axis label
xlabel
ax.set_xlabel
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75

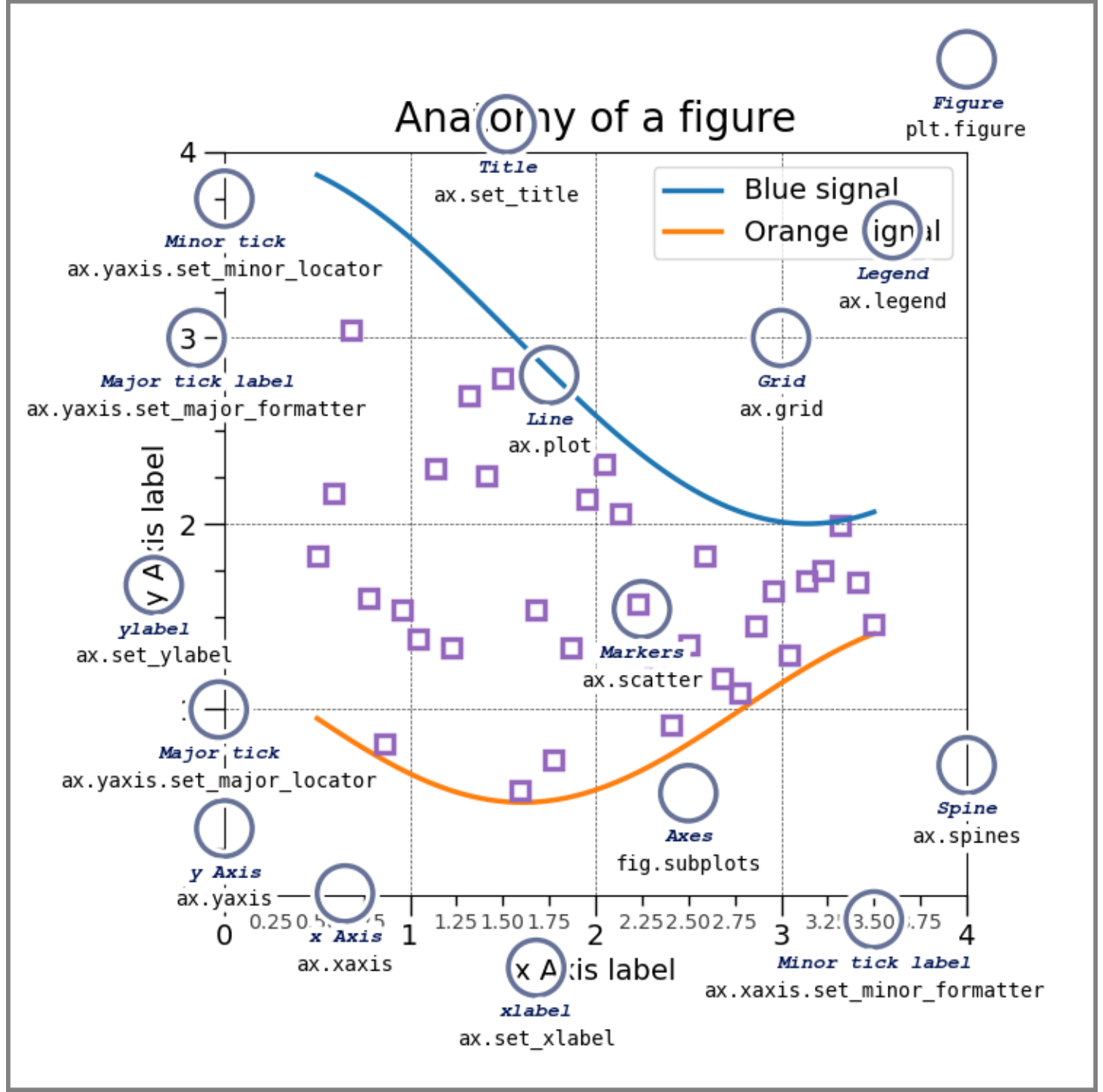

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Minor tick
ax.yaxis.set_minor_locator
Major tick label
ax.yaxis.set_major_formatter
ylabel
ax.set_ylabel
Major tick
ax.yaxis.set_major_locator
y Axis
ax.yaxis
x Axis
ax.xaxis
xlabel
ax.set_xlabel
Axes
fig.subplots
Spine
ax.spines
Blue signal
Orange signal
y Axis label
x Axis label
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.25
3.50
3.75


Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Minor tick
ax.yaxis.set_minor_locator
Major tick label
ax.yaxis.set_major_formatter
ylabel
ax.set_ylabel
Major tick
ax.yaxis.set_major_locator
y Axis
ax.yaxis
x Axis
ax.xaxis
xlabel
ax.set_xlabel
Axes
fig.subplots
Spine
ax.spines
Minor tick label
ax.xaxis.set_minor_formatter
Blue signal
Orange signal
y Axis label
x Axis label
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.50

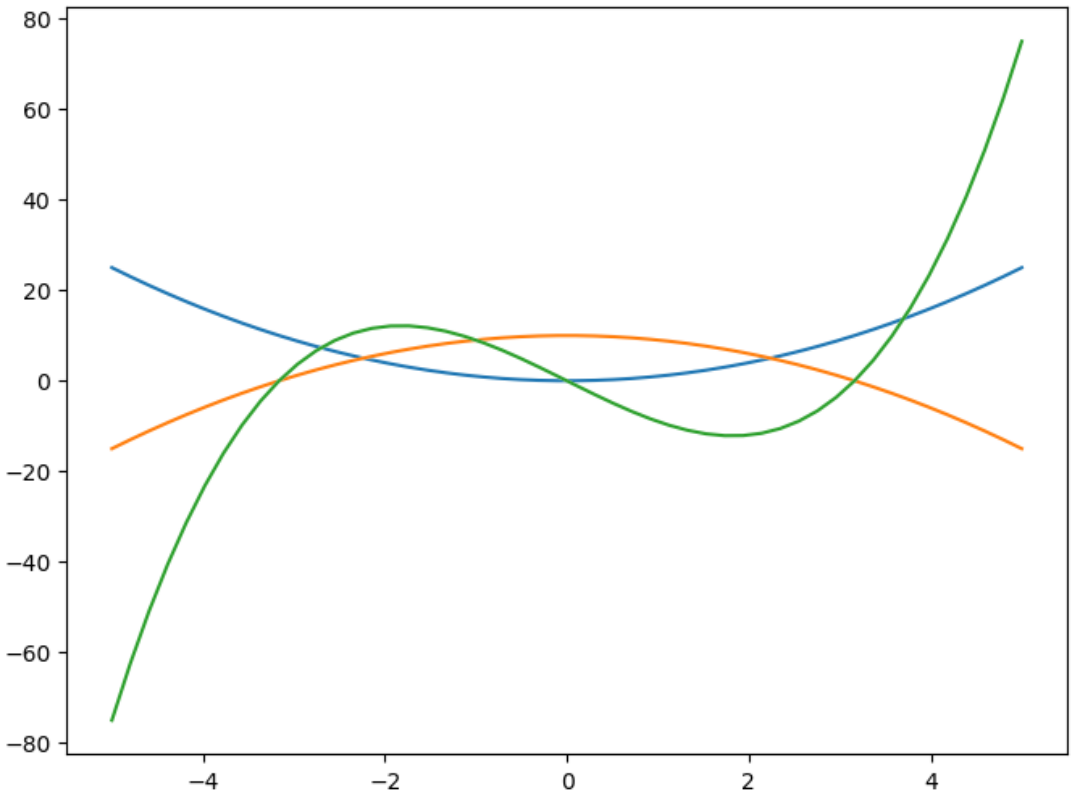

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Minor tick
ax.yaxis.set_minor_locator
Major tick label
ax.yaxis.set_major_formatter
Grid
ax.grid
ylabel
ax.set_ylabel
Major tick
ax.yaxis.set_major_locator
Axes
fig.subplots
Spine
ax.spines
y Axis
ax.yaxis
x Axis
ax.xaxis
xlabel
ax.set_xlabel
Minor tick label
ax.xaxis.set_minor_formatter
Blue signal
Orange signal
y Axis label
x Axis label

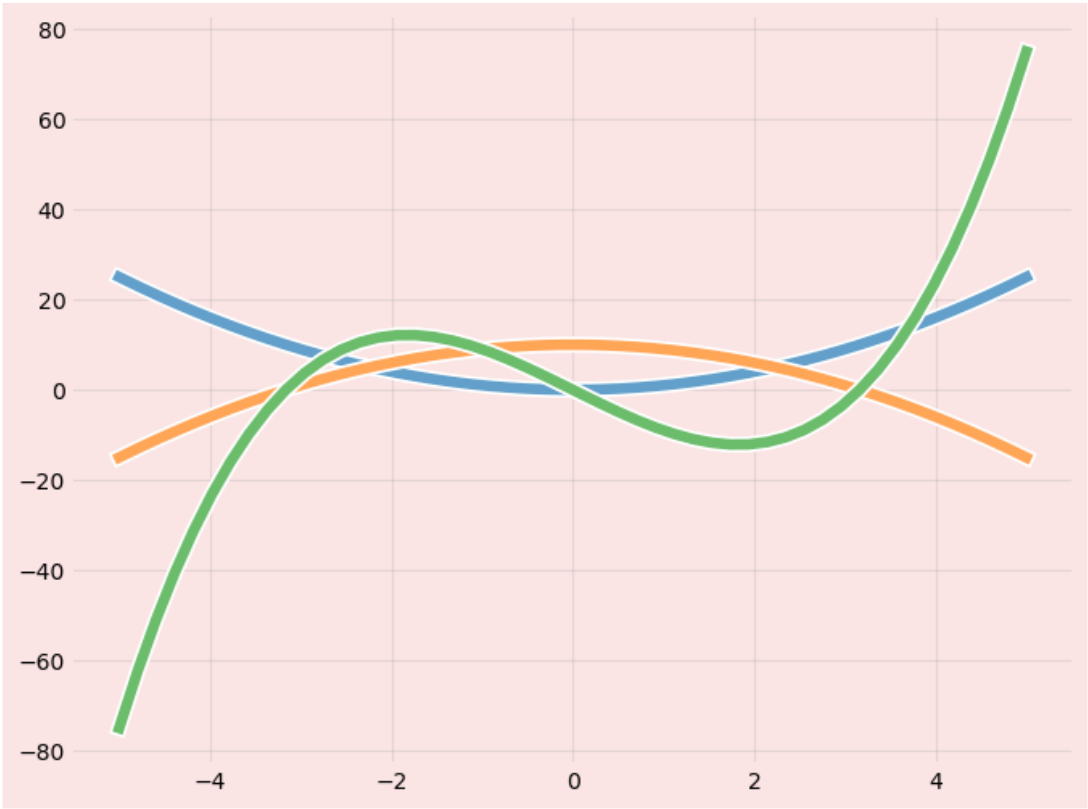

Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Minor tick
ax.yaxis.set_minor_locator
Major tick label
ax.yaxis.set_major_formatter
Line
ax.plot
Grid
ax.grid
ylabel
ax.set_ylabel
Major tick
ax.yaxis.set_major_locator
Axes
fig.subplots
Spine
ax.spines
y Axis
ax.yaxis
x Axis
ax.xaxis
xlabel
ax.set_xlabel
Minor tick label
ax.xaxis.set_minor_formatter
Blue signal
Orange signal
y Axis label
x Axis label


Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Minor tick
ax.yaxis.set_minor_locator
Major tick label
ax.yaxis.set_major_formatter
Line
ax.plot
Grid
ax.grid
Blue signal
Orange signal
ylabel
ax.set_ylabel
Markers
ax.scatter
Major tick
ax.yaxis.set_major_locator
Axes
fig.subplots
Spine
ax.spines
y Axis
ax.yaxis
x Axis
ax.xaxis
xlabel
ax.set_xlabel
Minor tick label
ax.xaxis.set_minor_formatter


Anatomy of a figure
Figure
plt.figure
Title
ax.set_title
Blue signal
Orange signal
Legend
ax.legend
Minor tick
ax.yaxis.set_minor_locator
Major tick label
ax.yaxis.set_major_formatter
Grid
ax.grid
Line
ax.plot
y Axis label
ylabel
ax.set_ylabel
Markers
ax.scatter
Major tick
ax.yaxis.set_major_locator
Axes
fig.subplots
Spine
ax.spines
y Axis
ax.yaxis
x Axis
ax.xaxis
x Axis label
xlabel
ax.set_xlabel
Minor tick label
ax.xaxis.set_minor_formatter
0
1
2
3
4
0.25
1.25
1.50
1.75
2.25
2.50
2.75
3.50

Demo

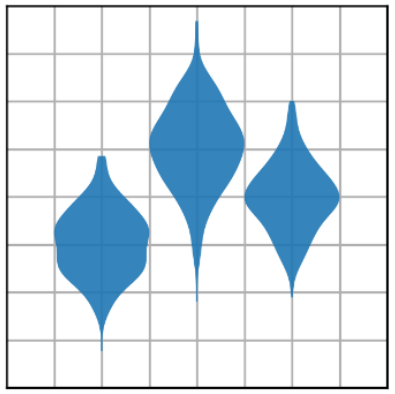




Matplotlib basics

matplotlib

Plot types Examples Tutorials Reference User guide Develop Release notes



violinplot(D)

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

News

September 16, 2022

Matplotlib 3.6.0 Released →

Resources

Be sure to check the Users guide and the API docs. The full text search is a good way to discover the docs including the many examples.

https://matplotlib.org/

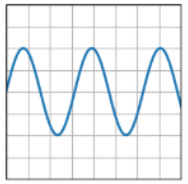
Plot types

Overview of many common plotting commands in Matplotlib.

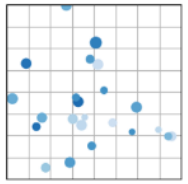
Note that we have stripped all labels, but they are present by default. See the gallery for many more examples and the tutorials page for longer examples.

Basic

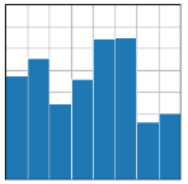
Basic plot types, usually y versus x.



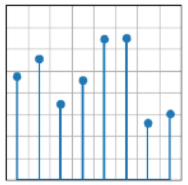
plot(x, y)



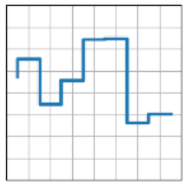
scatter(x, y)



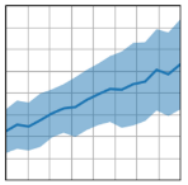
bar(x, height)



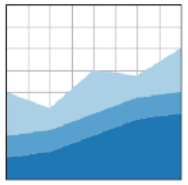
stem(x, y)



step(x, y)



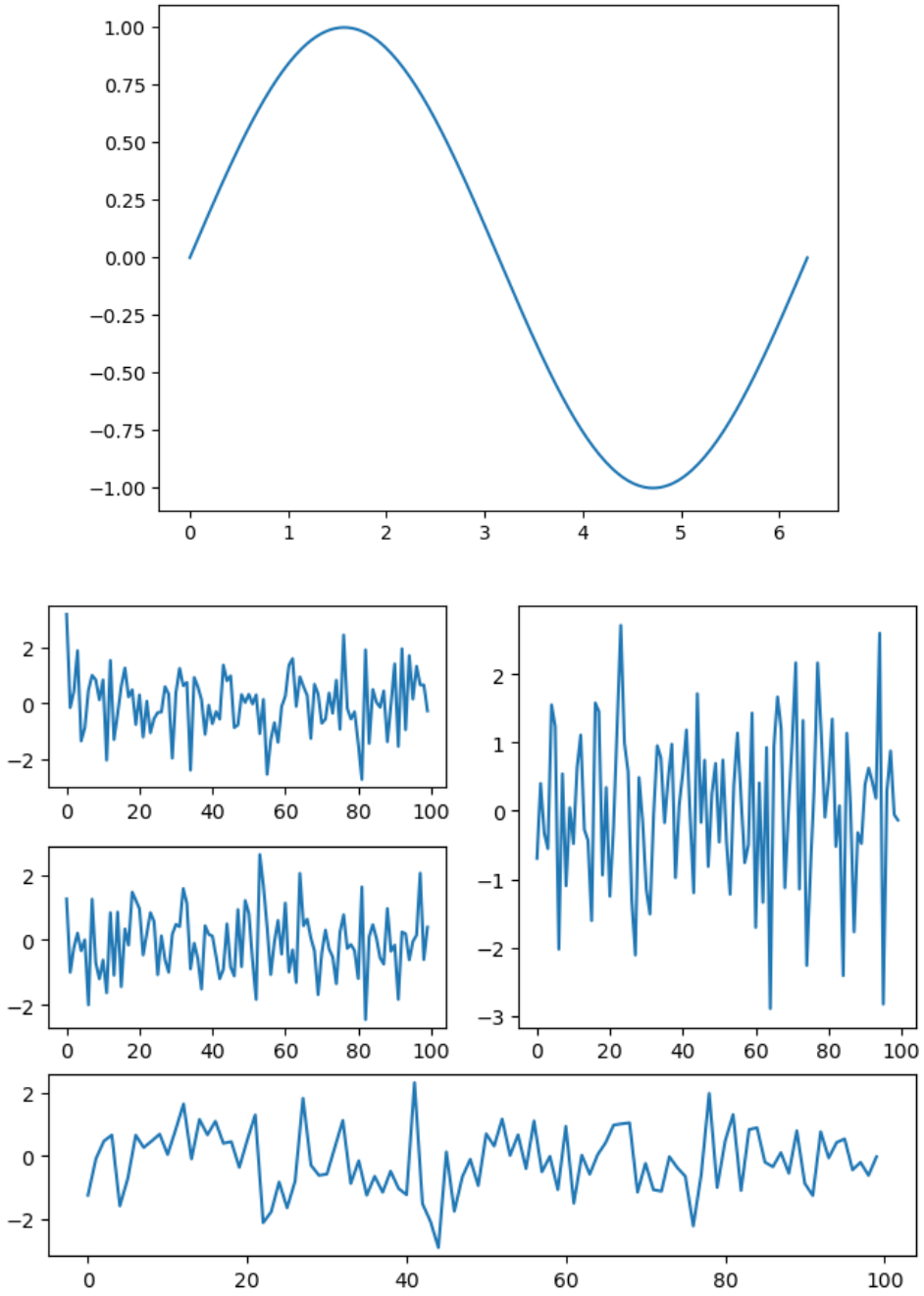
fill_between(x, y1, y2)



stackplot(x, y)

https://matplotlib.org/

- Demo
- Quick start at https://matplotlib.org/stable/tutorials/introductory/quick_start.html



Pandas & Seaborn

https://pandas.pydata.org/

pandas Getting started User Guide API reference Development Release notes

pandas documentation

Date: Oct 19, 2022 **Version**: 1.5.1

Download documentation: Zipped HTML

Previous versions: Documentation of previous pandas versions is available at pandas.pydata.org.

Useful links: Binary Installers | Source Repository | Issues & Ideas | Q&A Support | Mailing List

`pandas` is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Getting started

New to *pandas*? Check out the getting started guides. They contain an introduction to *pandas*' main concepts and links to additional tutorials.

To the getting started guides

User guide

The user guide provides in-depth information on the key concepts of pandas with useful background information and explanation.

To the user guide

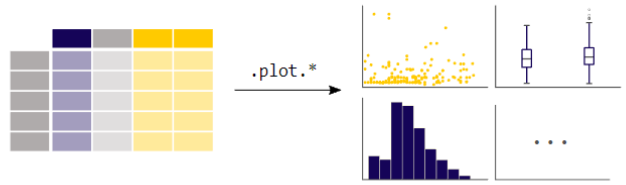
API reference

Developer guide

pandas Getting started User Guide API reference Development Release notes

- Installation
- Package overview
- Getting started tutorials
  - What kind of data does pandas handle?
  - How do I read and write tabular data?
  - How do I select a subset of a `DataFrame`?
  - How do I create plots in pandas?
  - How to create new columns derived from existing columns?
  - How to calculate summary statistics?
  - How to reshape the layout of tables?
  - How to combine data from multiple tables?
  - How to handle time series data with ease?
  - How to manipulate textual data?
- Comparison with other tools
  - Comparison with R / R libraries
  - Comparison with SQL
  - Comparison with spreadsheets
  - Comparison with SAS
  - Comparison with Stata
- Community tutorials

How do I create plots in pandas?




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

In [2]: import matplotlib.pyplot as plt
```

Data used for this tutorial:

Air quality data

```
In [3]: air_quality = pd.read_csv("data/air_quality_no2.csv", index_col=0, parse_dates=True)

In [4]: air_quality.head()
Out[4]:
                     station_antwerp  station_paris  station_london
datetime
2019-05-07 02:00:00              NaN            NaN            23.0
2019-05-07 03:00:00             50.5           25.0            19.0
2019-05-07 04:00:00             45.0           27.7            19.0
2019-05-07 05:00:00              NaN           50.4            16.0
2019-05-07 06:00:00              NaN           61.9             NaN
```

Note

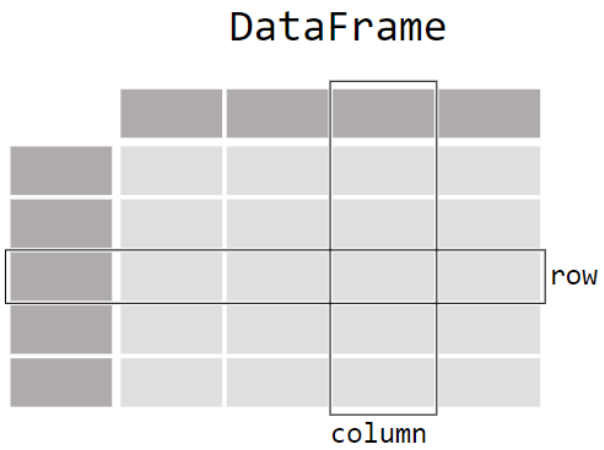
The usage of the `index_col` and `parse_dates` parameters of the `read_csv` function to define the first (0th) column as index of the resulting `DataFrame` and convert the dates in the column to `Timestamp` objects, respectively.

I want a quick visual check of the data.

```
In [5]: air_quality.plot()
Out[5]: <AxesSubplot: xlabel='datetime'>

In [6]: plt.show()
```

pandas data table representation



I want to store passenger data of the Titanic. For a number of passengers, I know the name (characters), age (integers) and sex (male/female) data.

```
In [2]: df = pd.DataFrame(
   ...:     {
   ...:         "Name": [
   ...:             "Braund, Mr. Owen Harris",
   ...:             "Allen, Mr. William Henry",
   ...:             "Bonnell, Miss. Elizabeth",
   ...:         ],
   ...:         "Age": [22, 35, 58],
   ...:         "Sex": ["male", "male", "female"],
   ...:     }
   ...: )
   ...:

In [3]: df
Out[3]:
                        Name  Age     Sex
0    Braund, Mr. Owen Harris   22    male
1  Allen, Mr. William Henry    35    male
2  Bonnell, Miss. Elizabeth    58  female
```

```
1  penguins = sns.load_dataset("penguins")
```
✓ 0.3s

```
1  penguins
```
✓ 0.1s

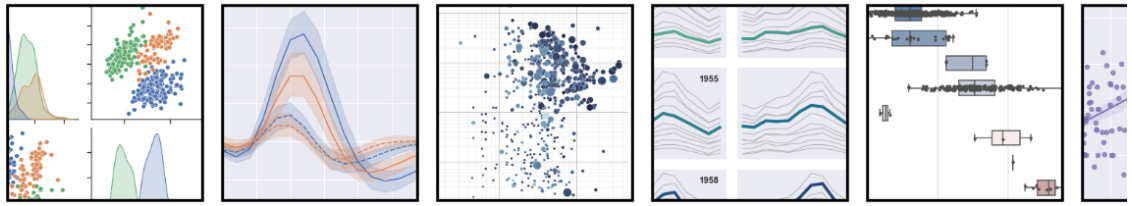
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	Female
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	Male
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	Female
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	Male

344 rows × 7 columns

https://seaborn.pydata.org/

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.

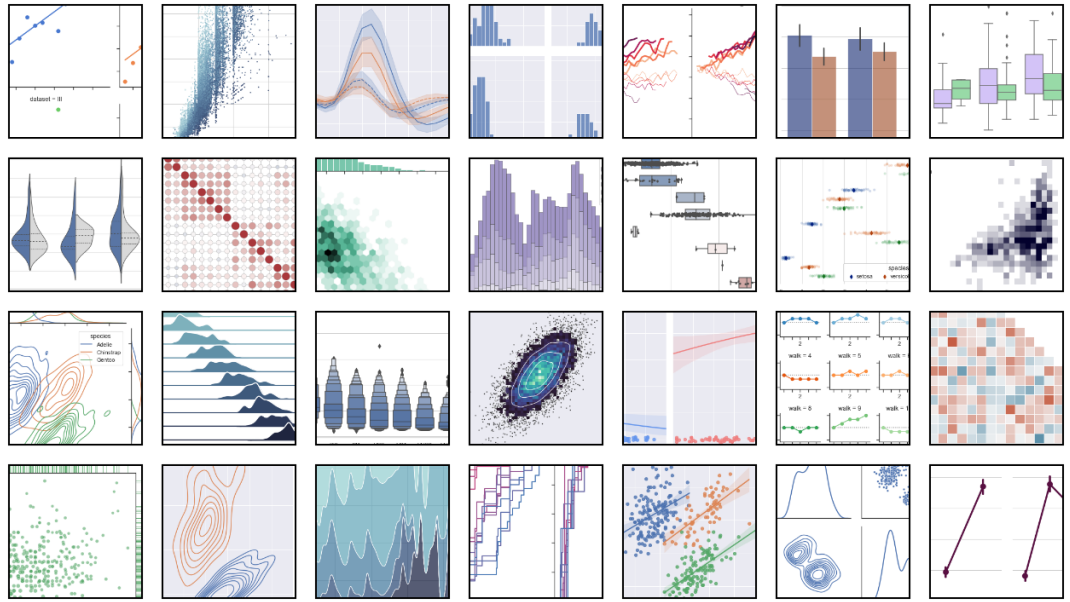
Contents

Features

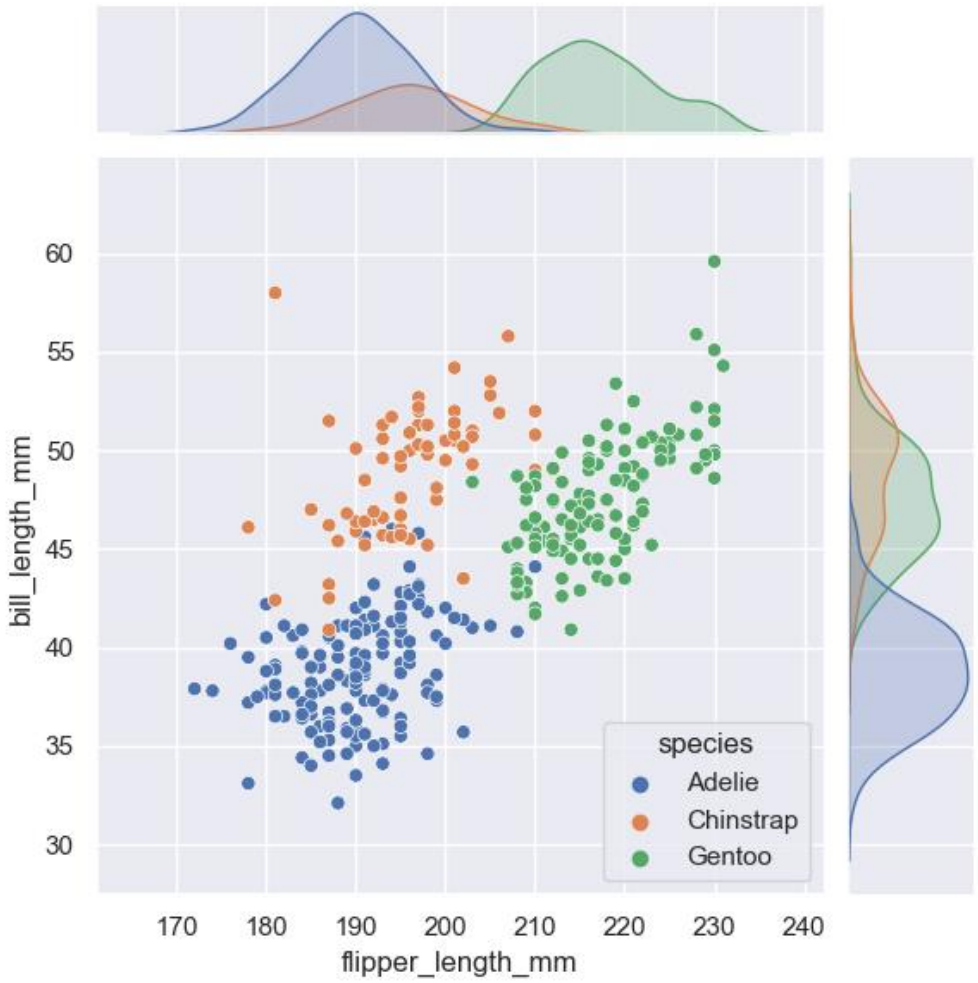
- New Objects: API | Tutorial
- Relational plots: API | Tutorial
- Distribution plots: API | Tutorial
- Categorical plots: API | Tutorial
- Regression plots: API | Tutorial
- Multi-plot grids: API | Tutorial
- Figure theming: API | Tutorial
- Color palettes: API | Tutorial

seaborn Installing Gallery Tutorial API Releases Citing FAQ

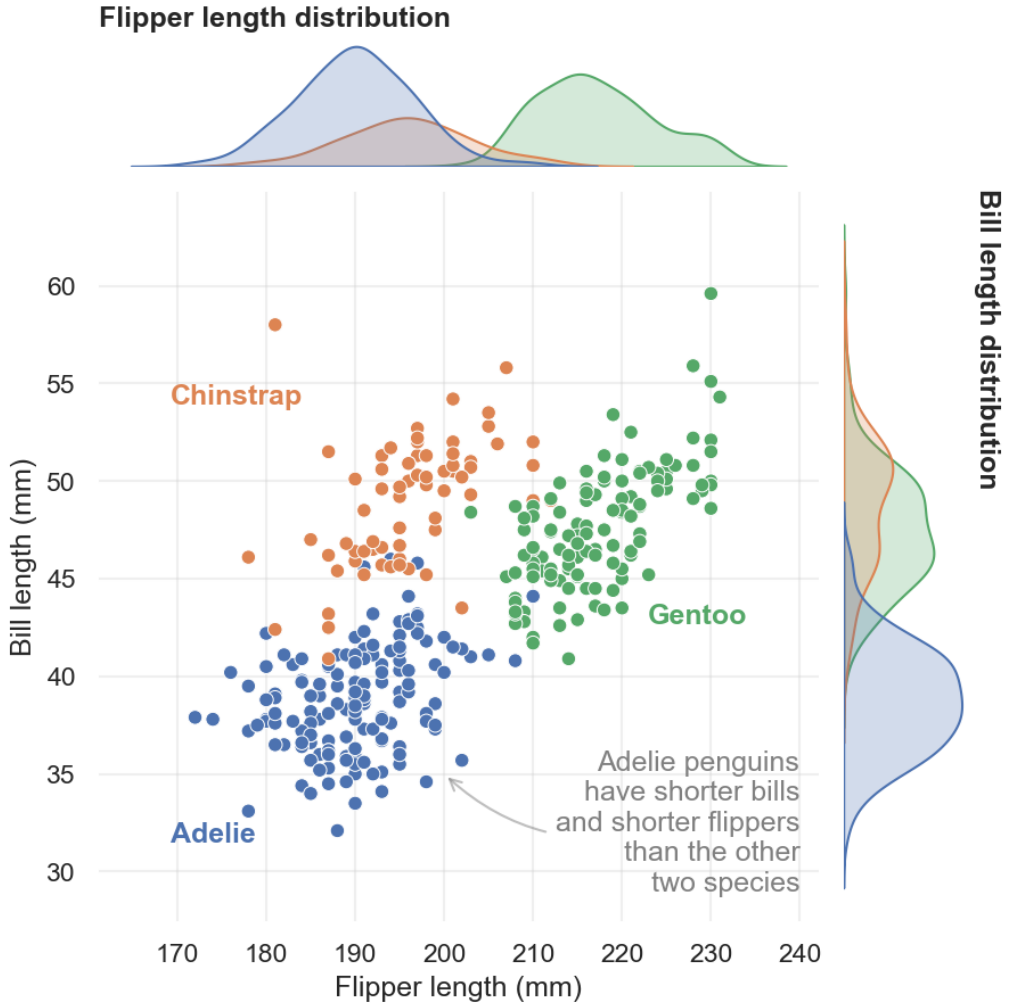
Example gallery



Examples & Hands-on



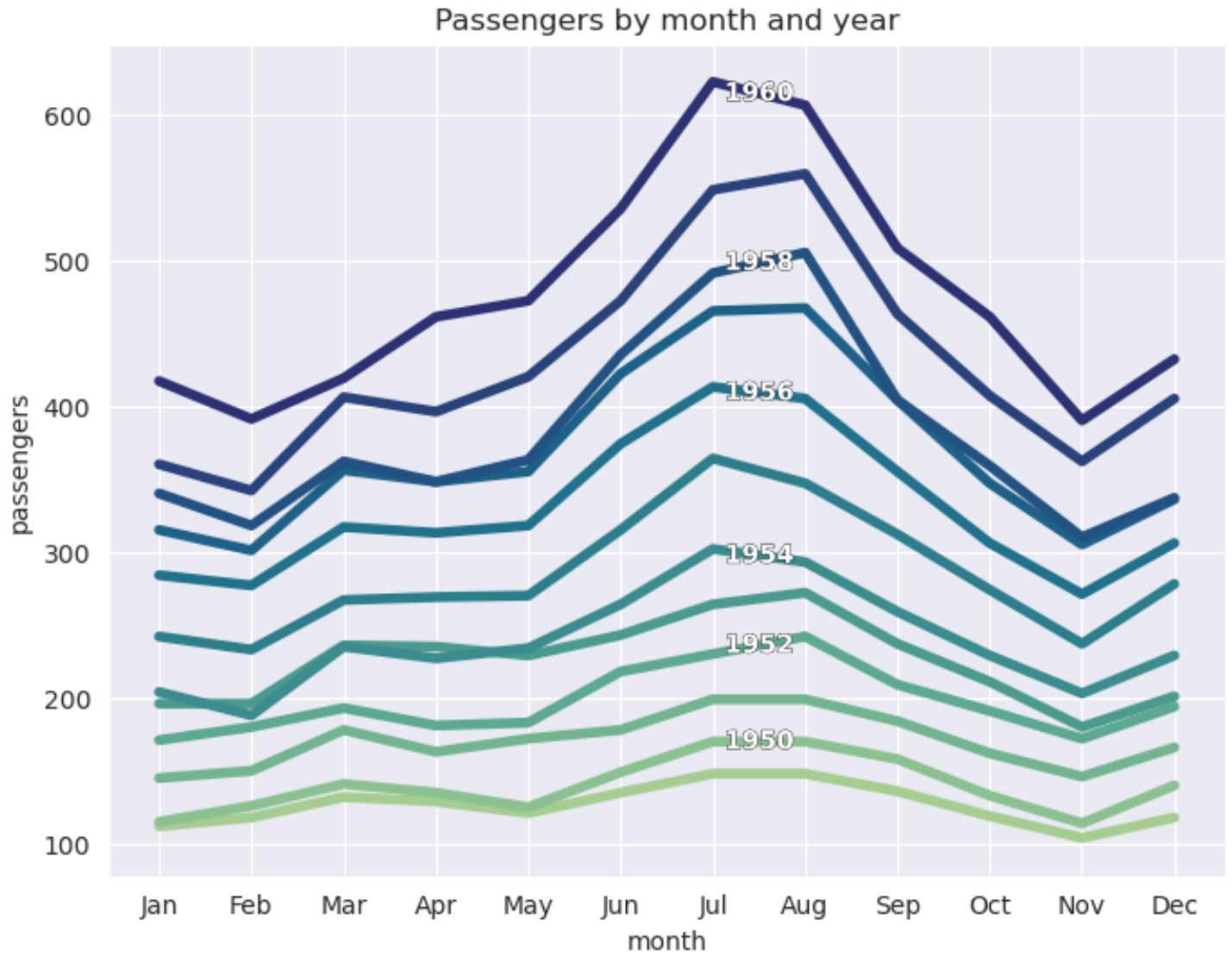
PENGUIN LENGTH RELATIONSHIPS



```
1  flights = sns.load_dataset("flights")
2  flights.head()
```

✓ 0.6s

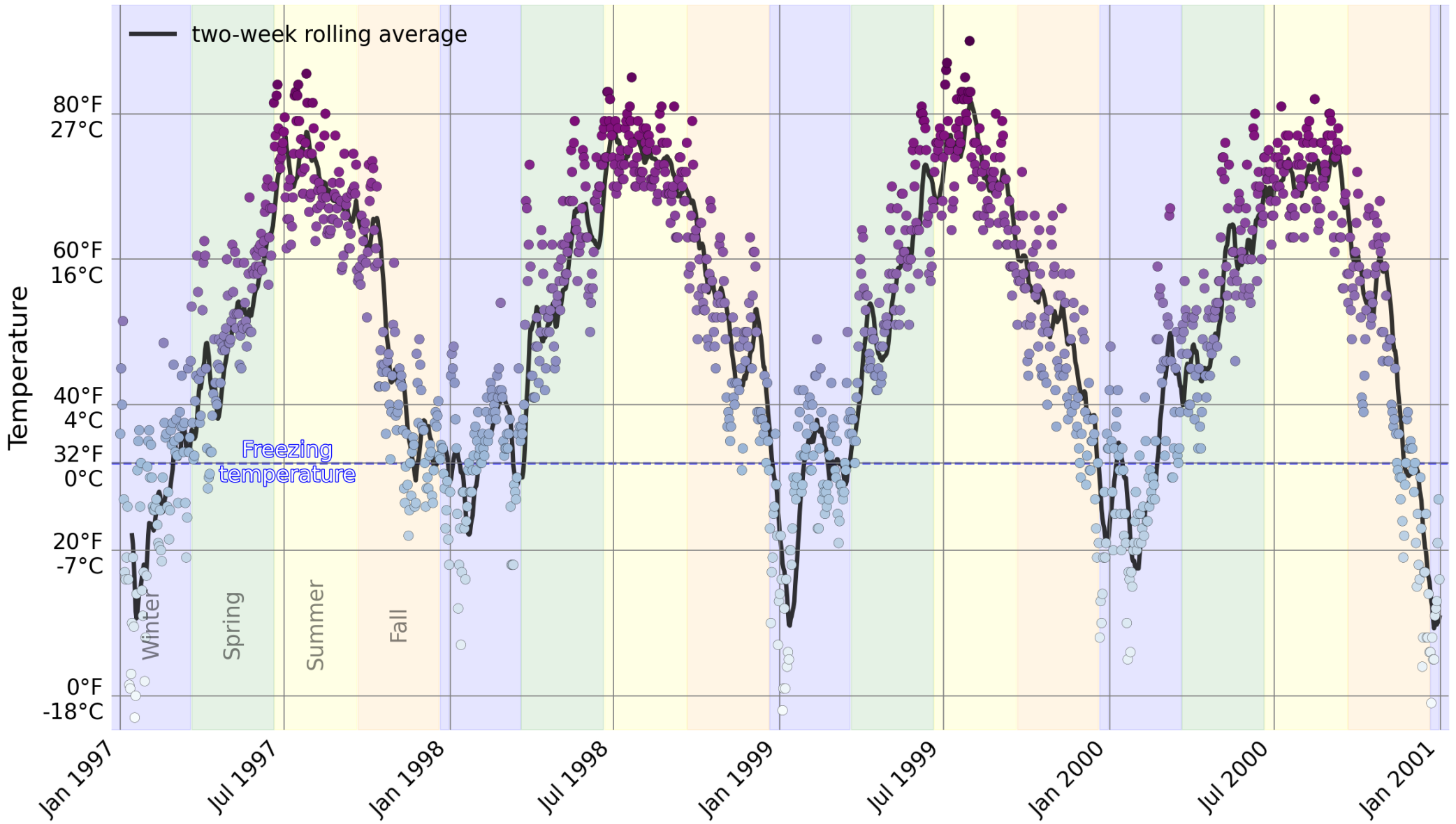
	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121



Conclusions & Remarks

Conclusions

- We have barely scratched the surface of what is available and what is possible
- Working with Python is nice because you can both use it:
  - to do whatever you need – including running complex code
  - AND visualize the outputs/results/relations
- Explore:
  - Check & Test
  - Check what others have done (and how)
    - The visualization community is quite open and happy to help and to provide feedback
    - Many share their code
- Practice


Daily Temperature in Chicago, 1997 to 2000
Data: NNMAPS
two-week rolling average
Temperature
80°F
27°C
60°F
16°C
40°F
4°C
32°F
0°C
20°F
-7°C
0°F
-18°C
Freezing temperature
Winter
Spring
Summer
Fall
Jan 1997
Jul 1997
Jan 1998
Jul 1998
Jan 1999
Jul 1999
Jan 2000
Jul 2000
Jan 2001

THANK YOU!

marcopiani@gmail.com