Fachhochschule Kiel
University of Applied Sciences

Tools and Programming Languages for Data Science
Introduction to Visualization

Study Program Data Science
Prof. Dr. Tillmann Schwörer

Our course agenda

- **Introduction and overview**
- **NumPy**: Basic data handling with Numpy arrays
- **Pandas**
  - Exploratory data analysis
  - Data consolidation
  - Data cleaning
- **Data visualization using Matplotlib and Seaborn**
- **Interacting with APIs**
- **Interacting with SQL databases**
- **Version Control with Git and GitHub**
- **Advanced Python**

Python foundations

Data types

Operators

Functions

Control flow and iterators

Programming concepts & paradigms

See also Precourse Programming

Tooling

Installation

Visual Studio Code

Jupyter Notebooks

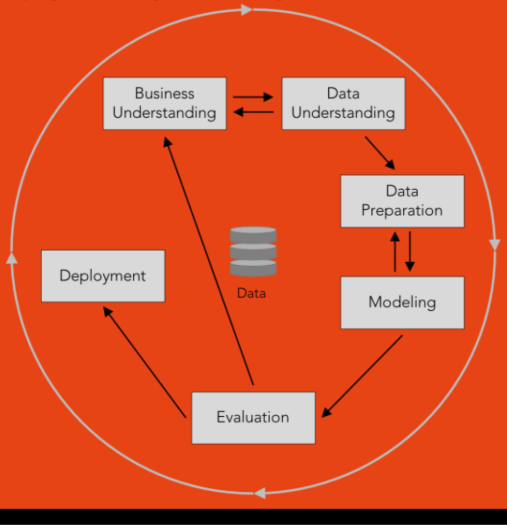
Packages

Virtual Environments

Git and Github

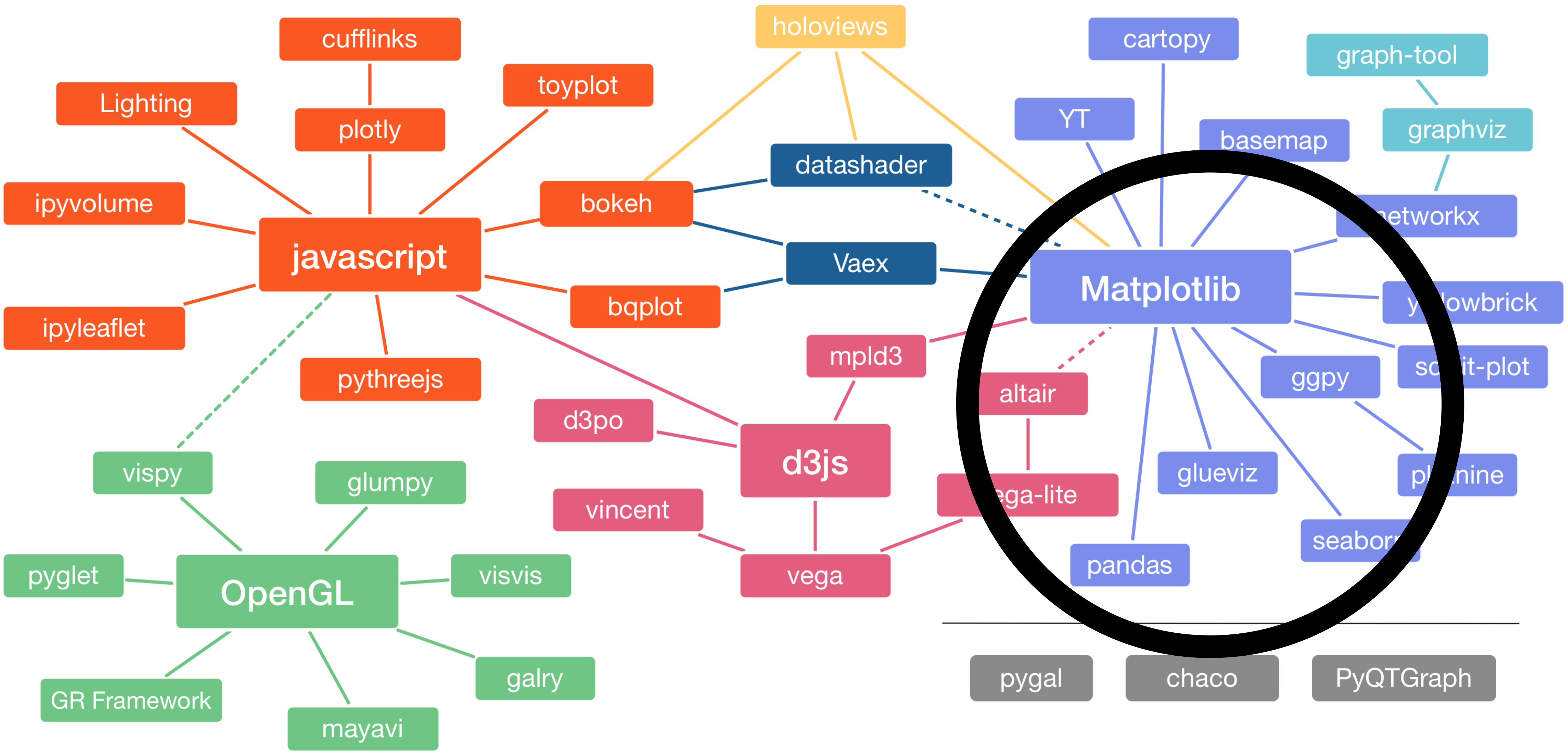
Python

Data Science Workflow



NumPy pandas matplotlib

Python Visualization Landscape



https://pyviz.org/overviews/

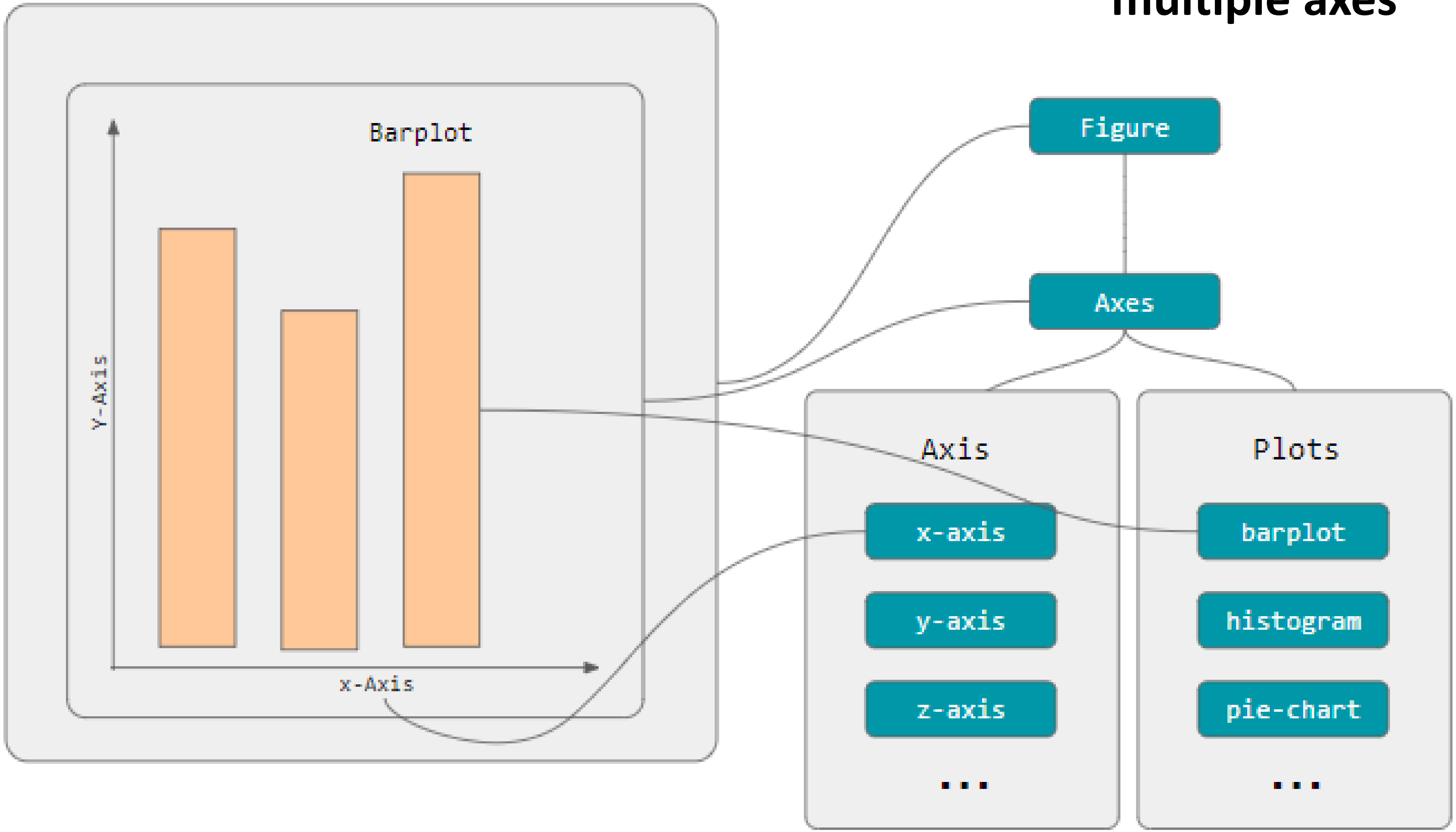
Matplotlib, Seaborn and Pandas

Library	Description	Key Features
Matplotlib	Core plotting library	• Low-level control of all aspects of a visualization
Seaborn	Statistical visualization library built on top off Matplotlib	• User-friendly functions • Complex visualizations and statistical plotting made easy • Works naturally with DataFrames and Series • Improved aesthetics compared to Matplotlib • Interoperable with Matplotlib
Pandas	Data analysis library with basic visualization functionalities	• Convenient plot methods for Pandas Series and Dataframes • Plot methods for grouped data • Interoperable with Matplotlib

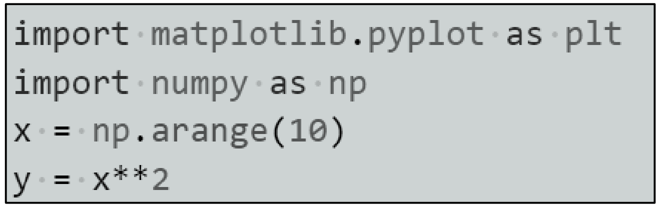
Matplotlib

Matplotlib: Figure, Axes, and Axis

Note: one figure can contain multiple axes

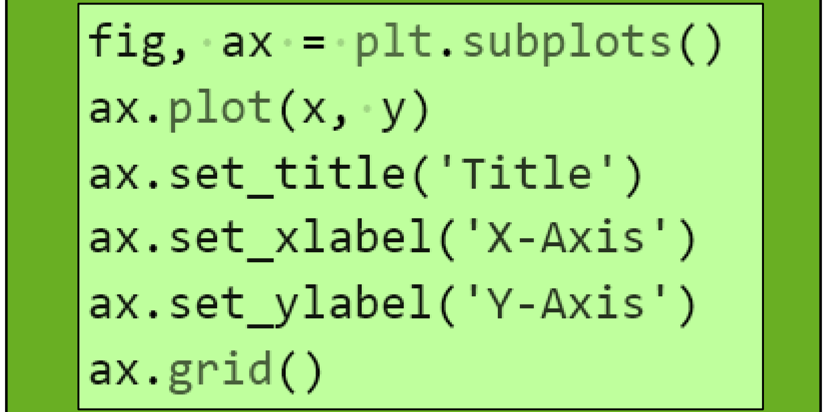


Two Matplotlib Interfaces



```python
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(10)
y = x**2
```

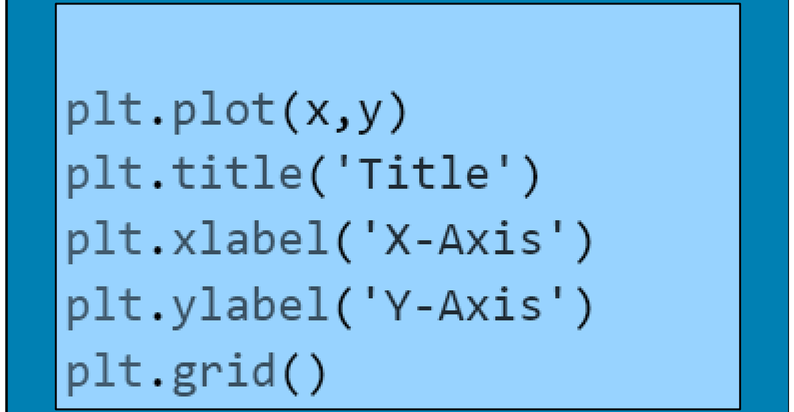
Explicit „Axes“ Interface



```python
fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_title('Title')
ax.set_xlabel('X-Axis')
ax.set_ylabel('Y-Axis')
ax.grid()
```

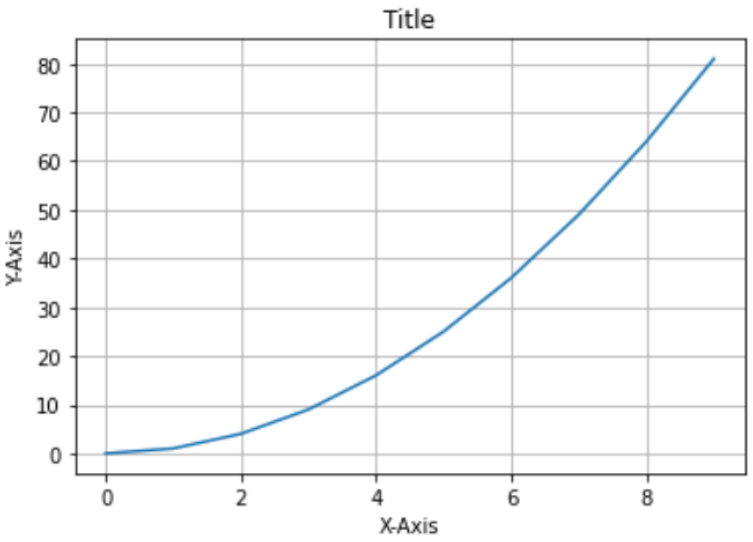
- Better for complex figures
- More explicit and readable
- Order of execution does not matter

Implicit „Pyplot“ Interface



```python
plt.plot(x,y)
plt.title('Title')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.grid()
```

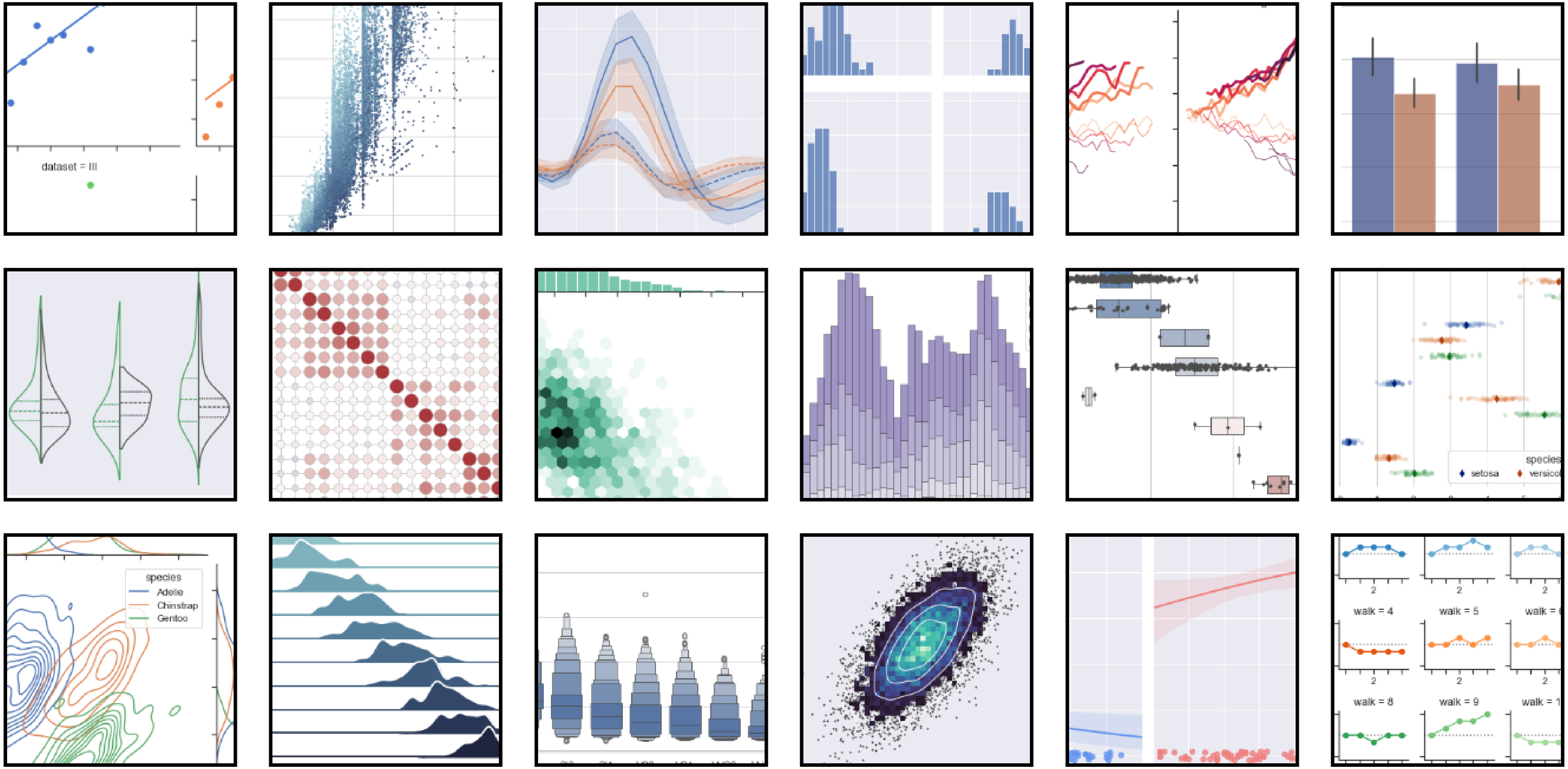
- Wrapper around the explicit interface
- Convenient for simple plots due to shorter code
- State-based / Matlab-like



See also: https://matplotlib.org/stable/users/explain/figure/api_interfaces.html

Seaborn

Seaborn

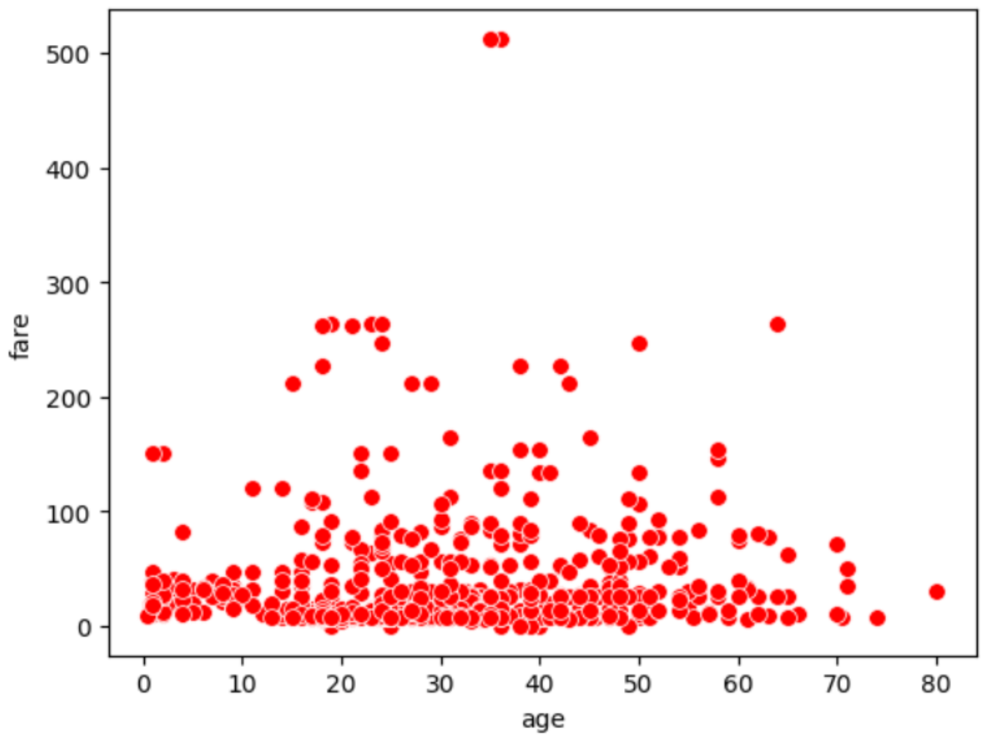

dataset = III
species
Adelie
Chinstrap
Gentoo
species
setosa
versicol
2
walk = 4
walk = 5
walk = 8
walk = 9

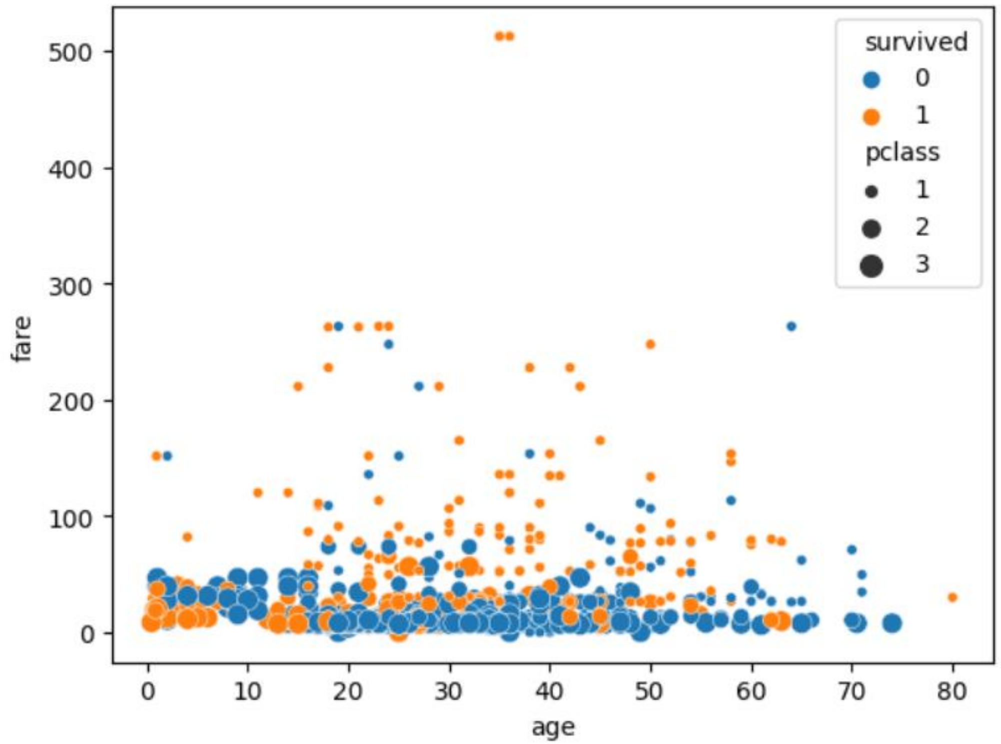
Advantages of Seaborn

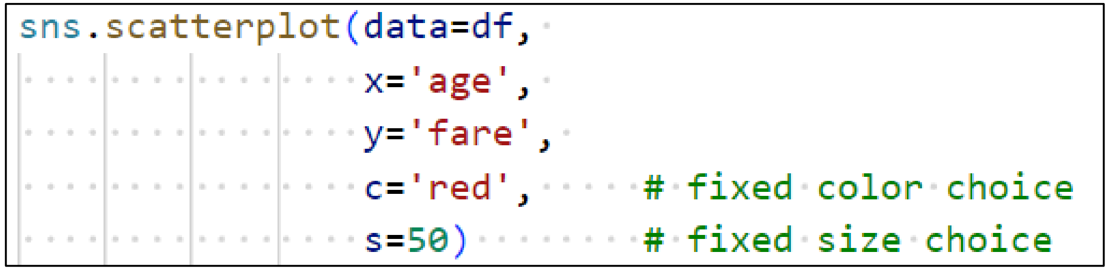
Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

- **High-level interface:**
  - Common plots are easily accessible
  - Few code lines for complex plots
- **Statistical Plotting**
  - Estimation and plotting of regression lines
  - Heatmaps, violin plots, …
- Works well with Pandas DataFrames and Series
- Better Default Aesthetics
- Comes with built-in datasets

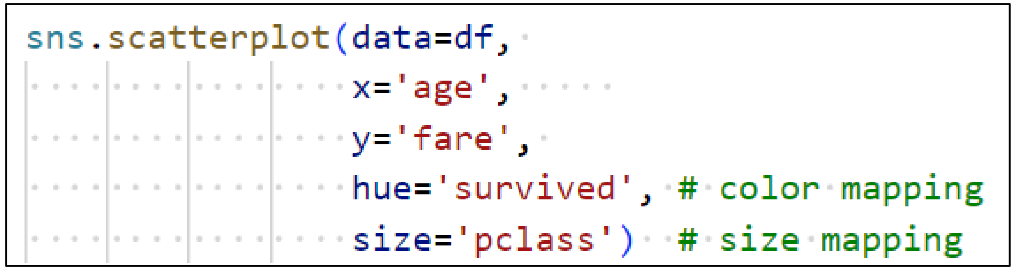
Fixed visual properties vs. data mapping





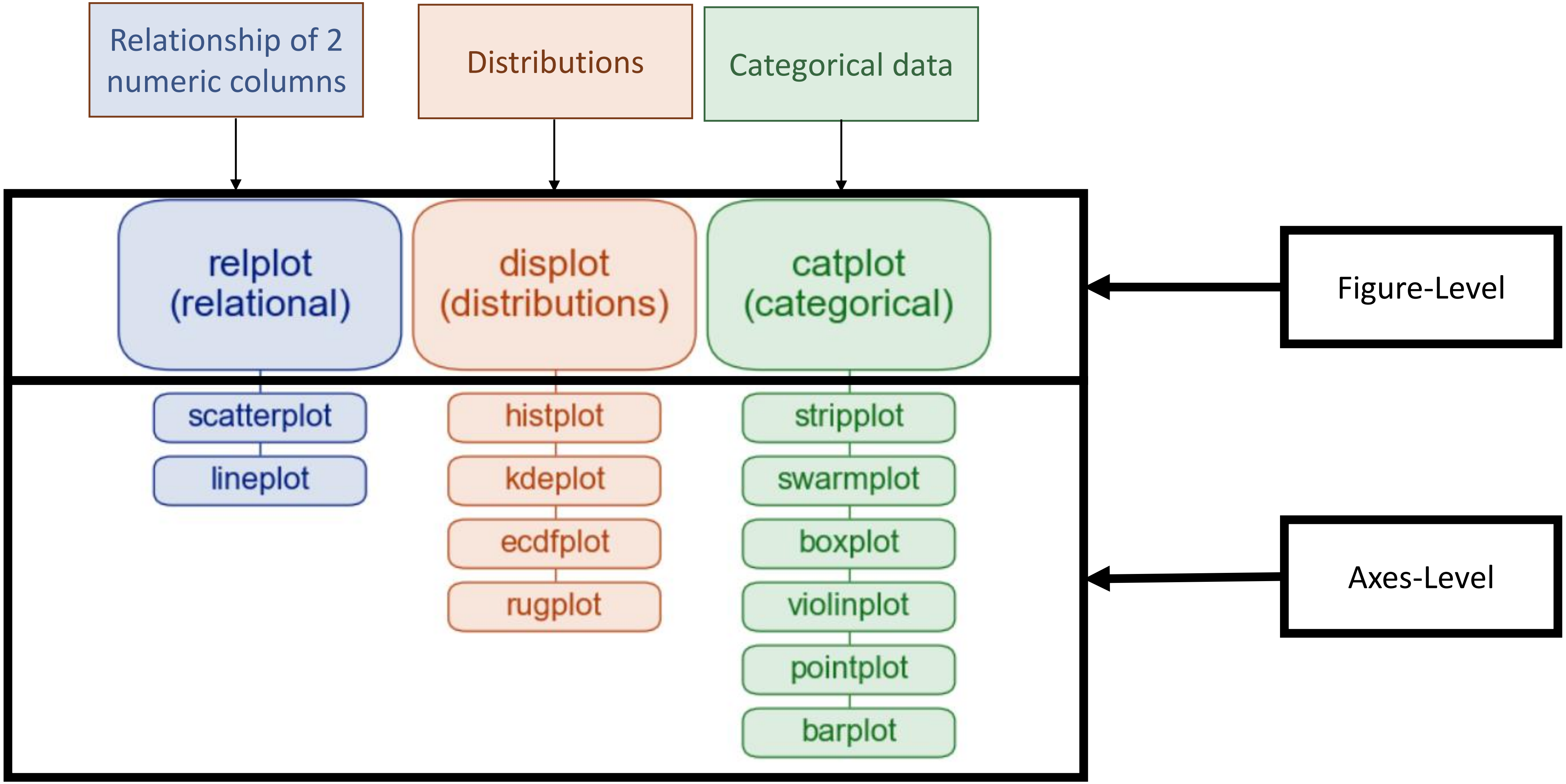


```
sns.scatterplot(data=df,
                x='age',
                y='fare',
                c='red',       # fixed color choice
                s=50)          # fixed size choice
```



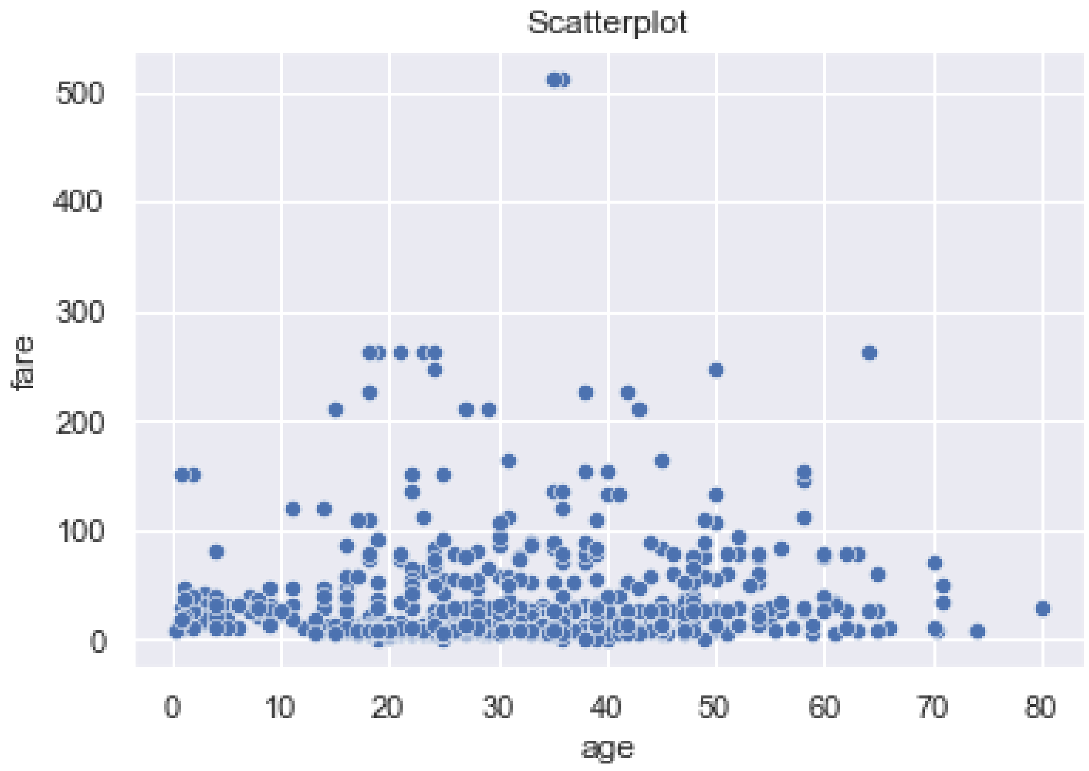
```
sns.scatterplot(data=df,
                x='age',
                y='fare',
                hue='survived', # color mapping
                size='pclass')  # size mapping
```

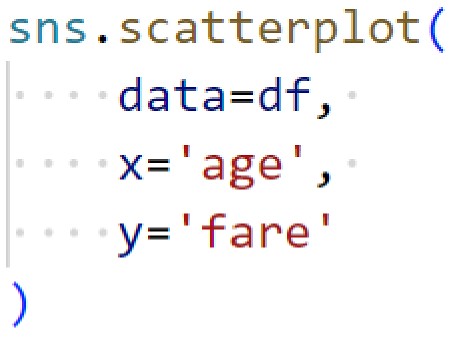
Figure-level vs. axes-level plotting



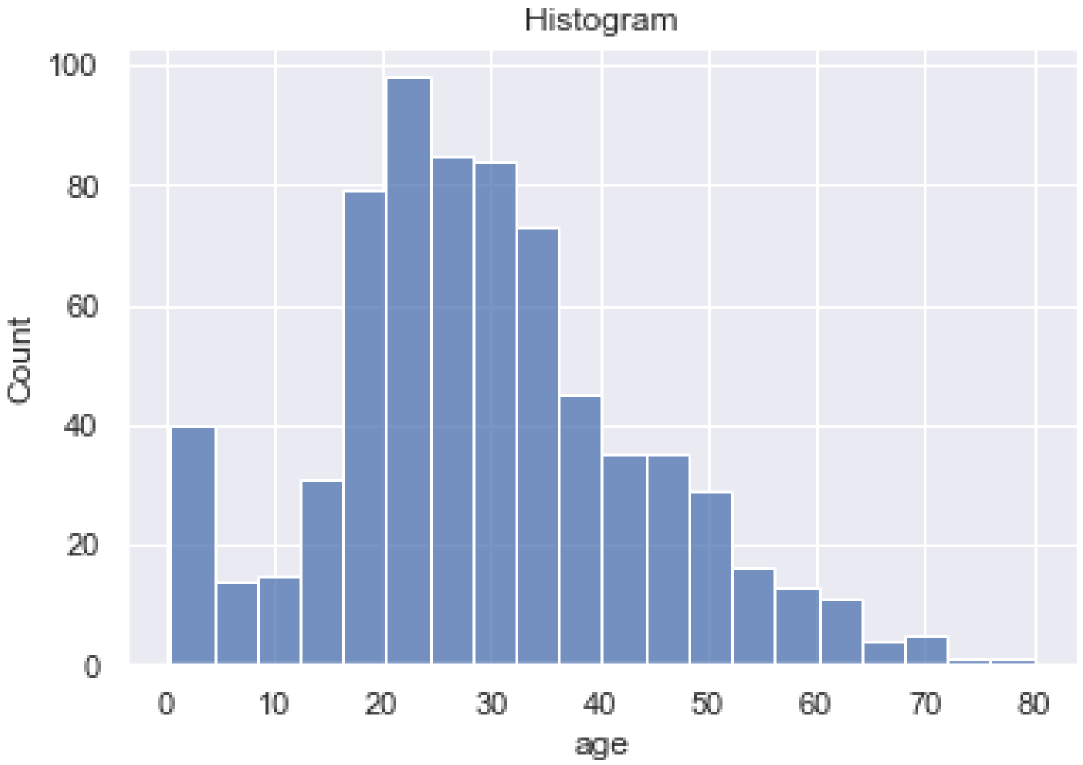
See: https://seaborn.pydata.org/tutorial/function_overview.html#figure-level-vs-axes-level-functions

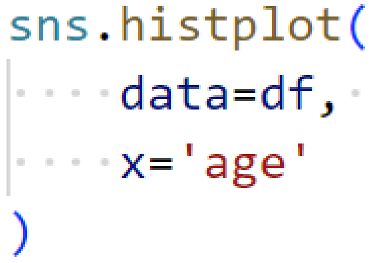
Axes Level Plots: Examples





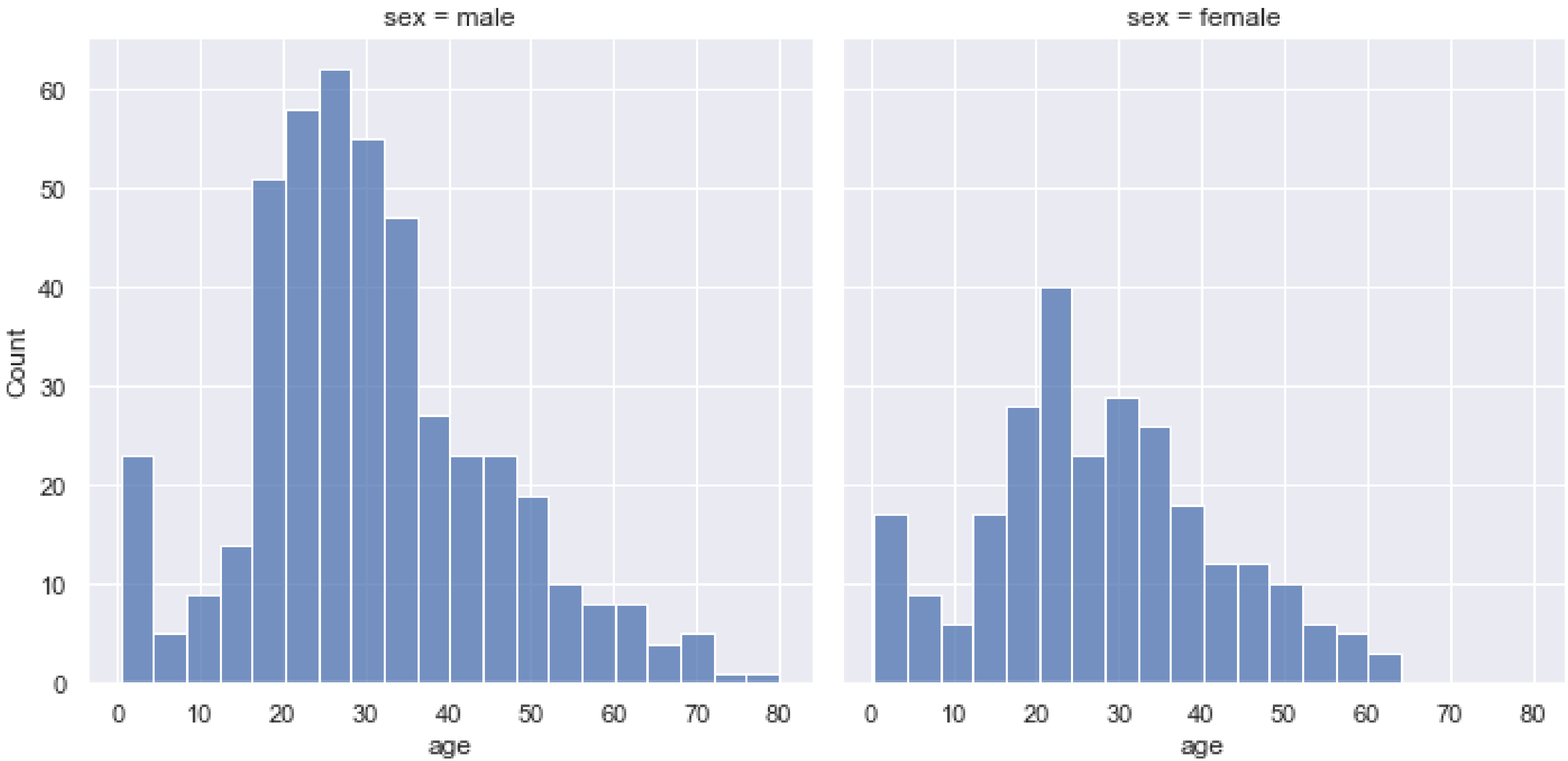
```
sns.scatterplot(
    data=df, 
    x='age', 
    y='fare'
)
```





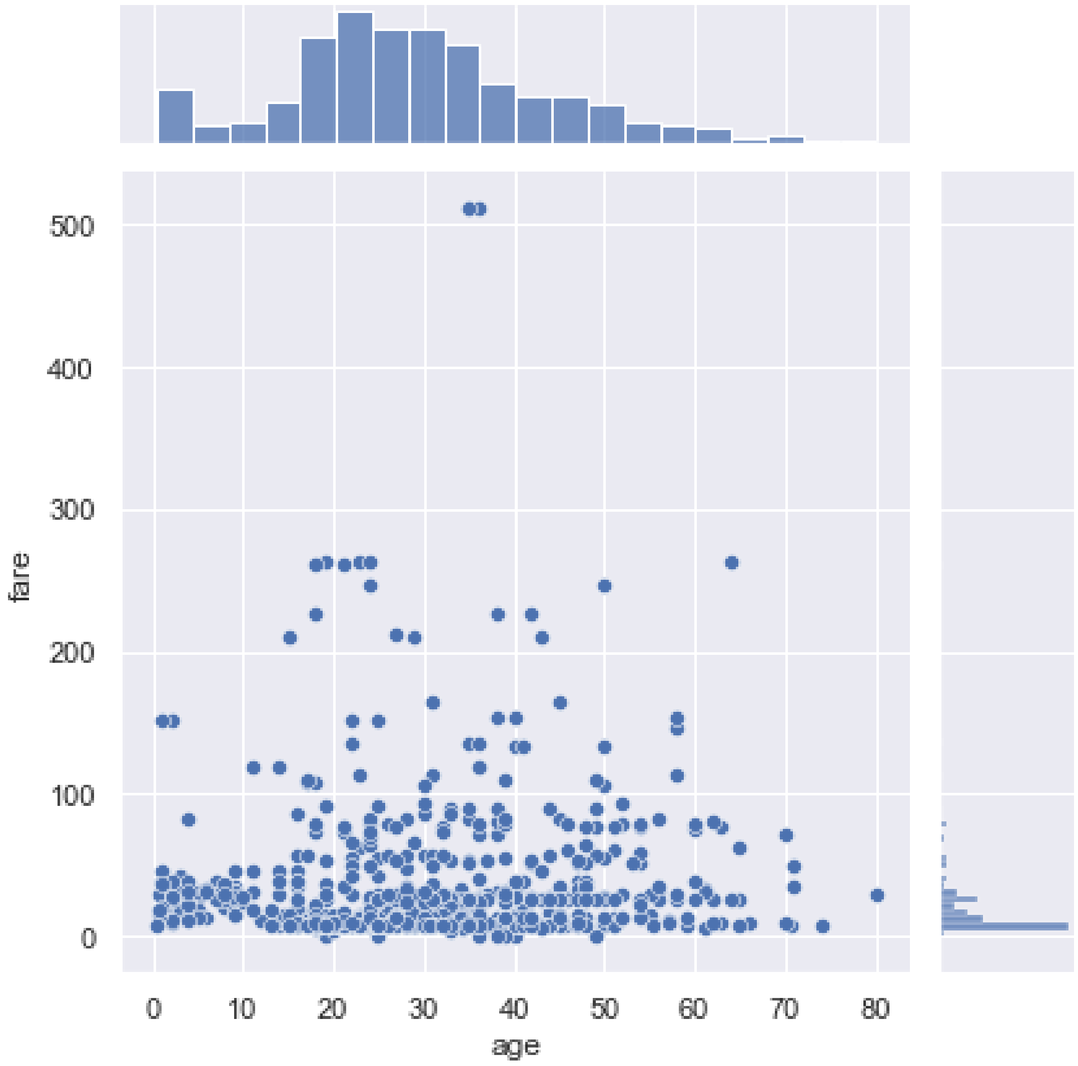
```
sns.histplot(
    data=df, 
    x='age'
)
```

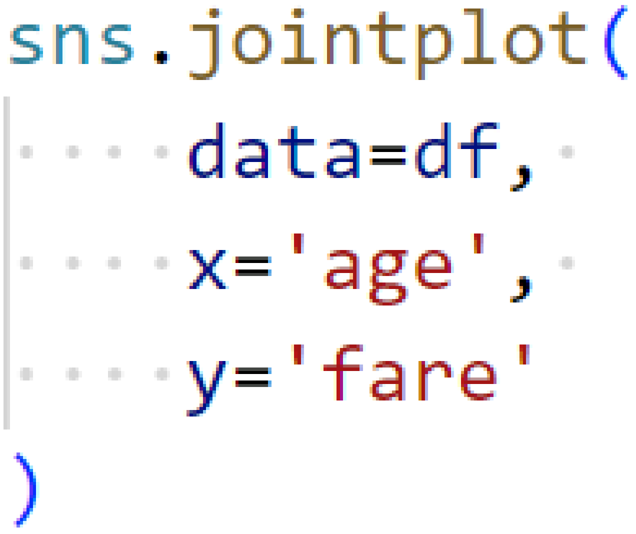
Figure Level Plot: Example 1



```
sns.displot(data=df, x='age', col='sex', kind='hist')
```

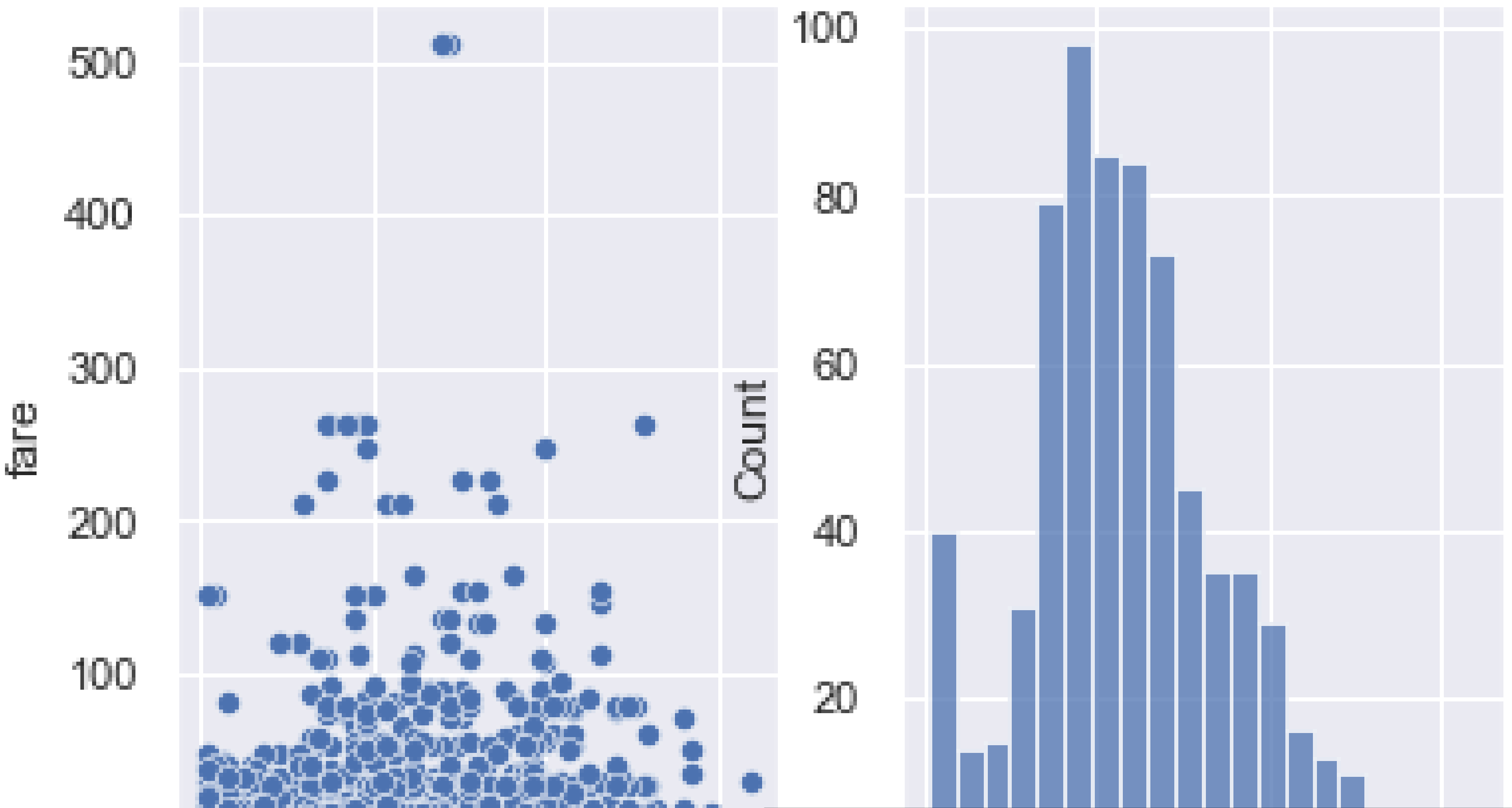
Figure Level Plot: Example 2

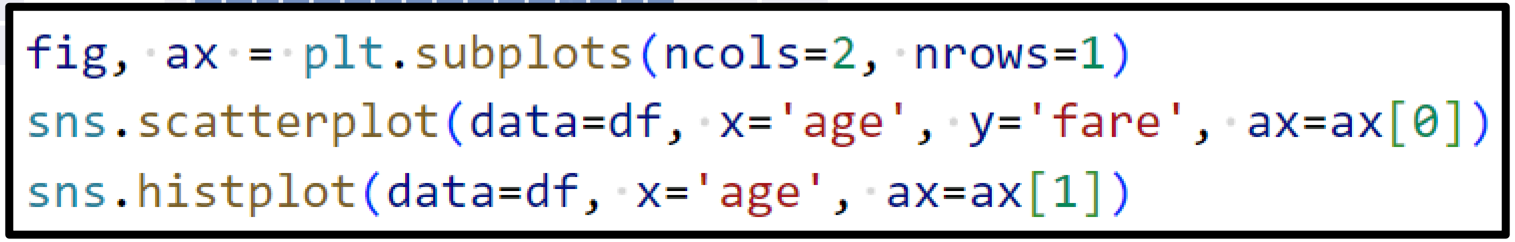




```
sns.jointplot(
    data=df,
    x='age',
    y='fare'
)
```

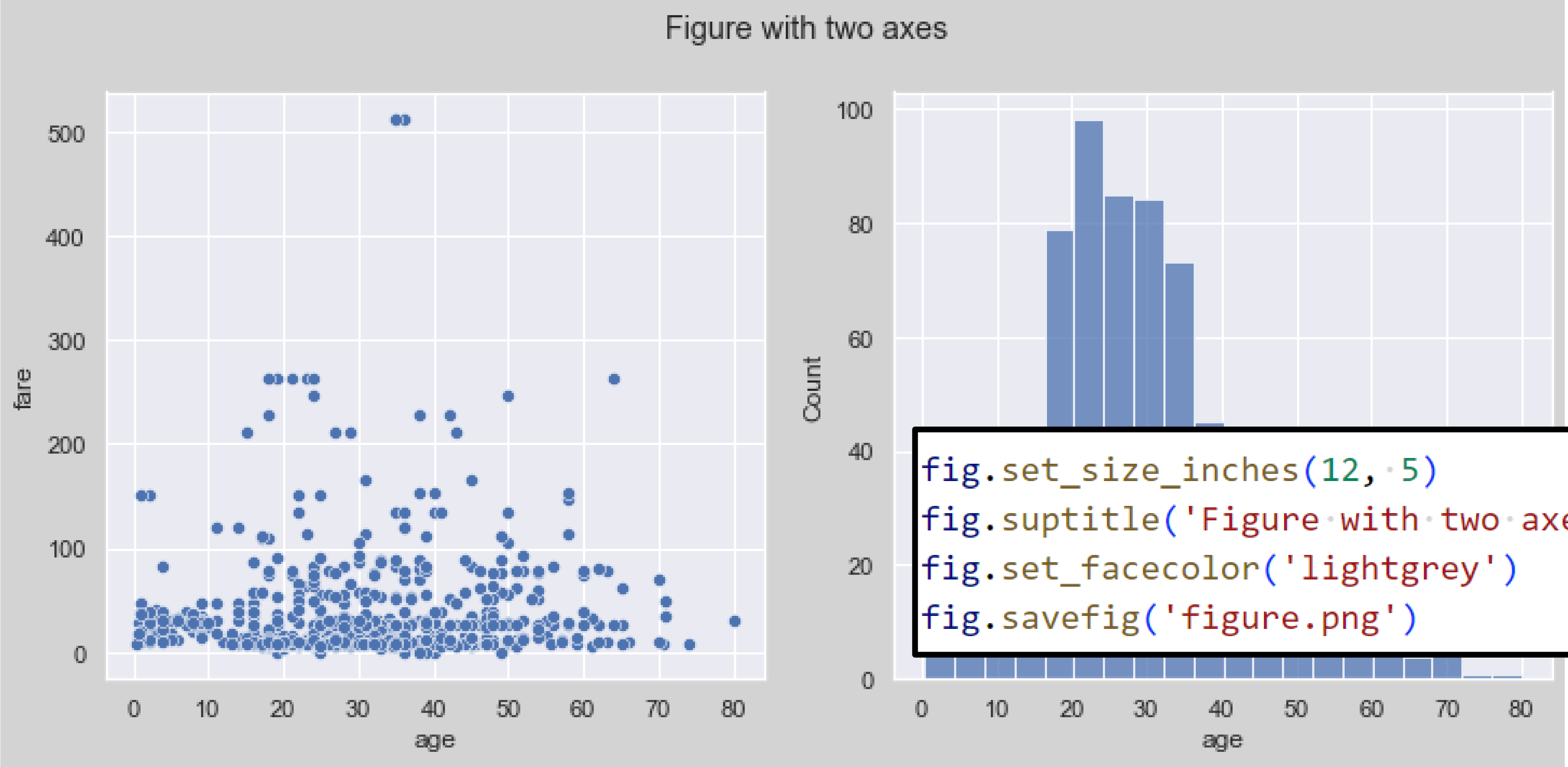
Combining Seaborn‘s axes level plots into 1 figure





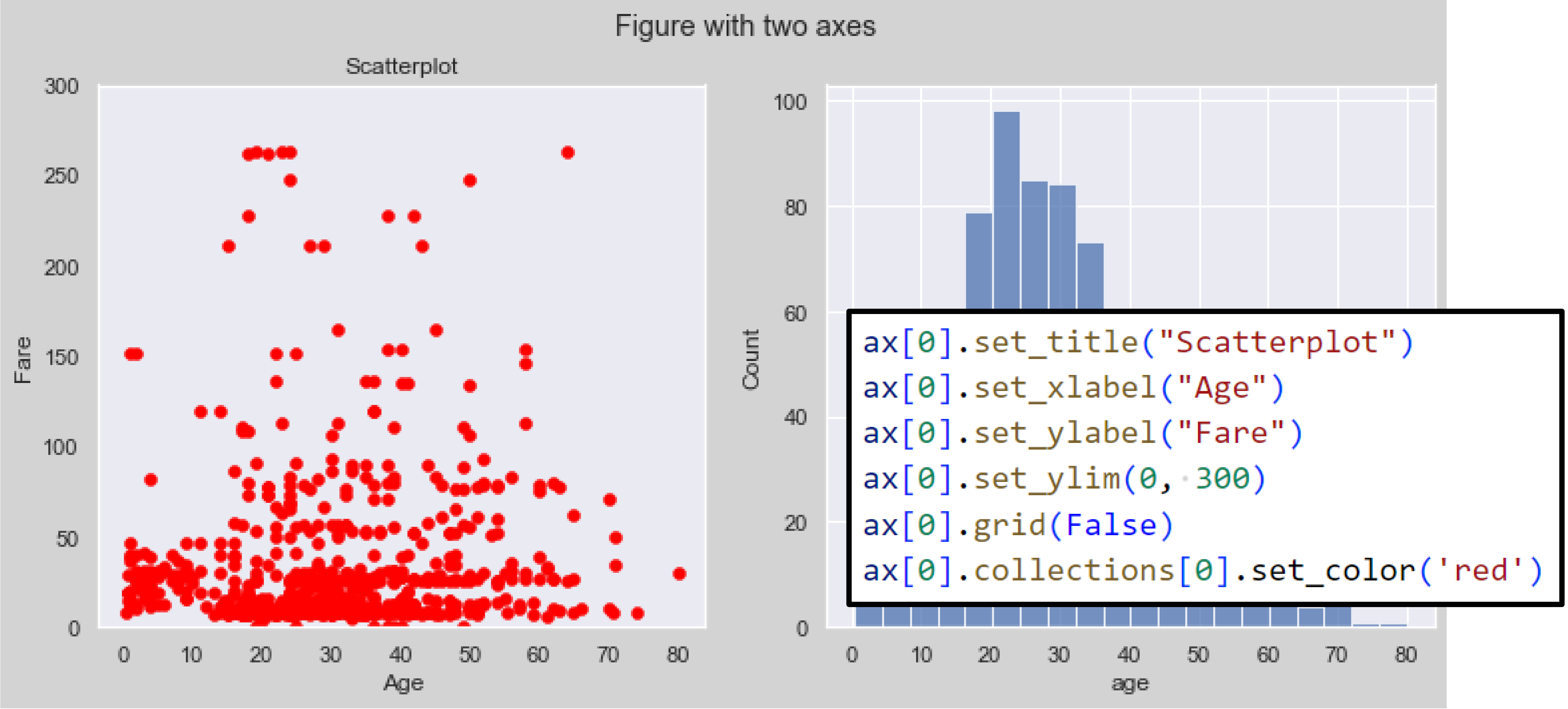
```
fig, ax = plt.subplots(ncols=2, nrows=1)
sns.scatterplot(data=df, x='age', y='fare', ax=ax[0])
sns.histplot(data=df, x='age', ax=ax[1])
```

Figure-level properties



```
fig.set_size_inches(12, 5)
fig.suptitle('Figure with two axes')
fig.set_facecolor('lightgrey')
fig.savefig('figure.png')
```

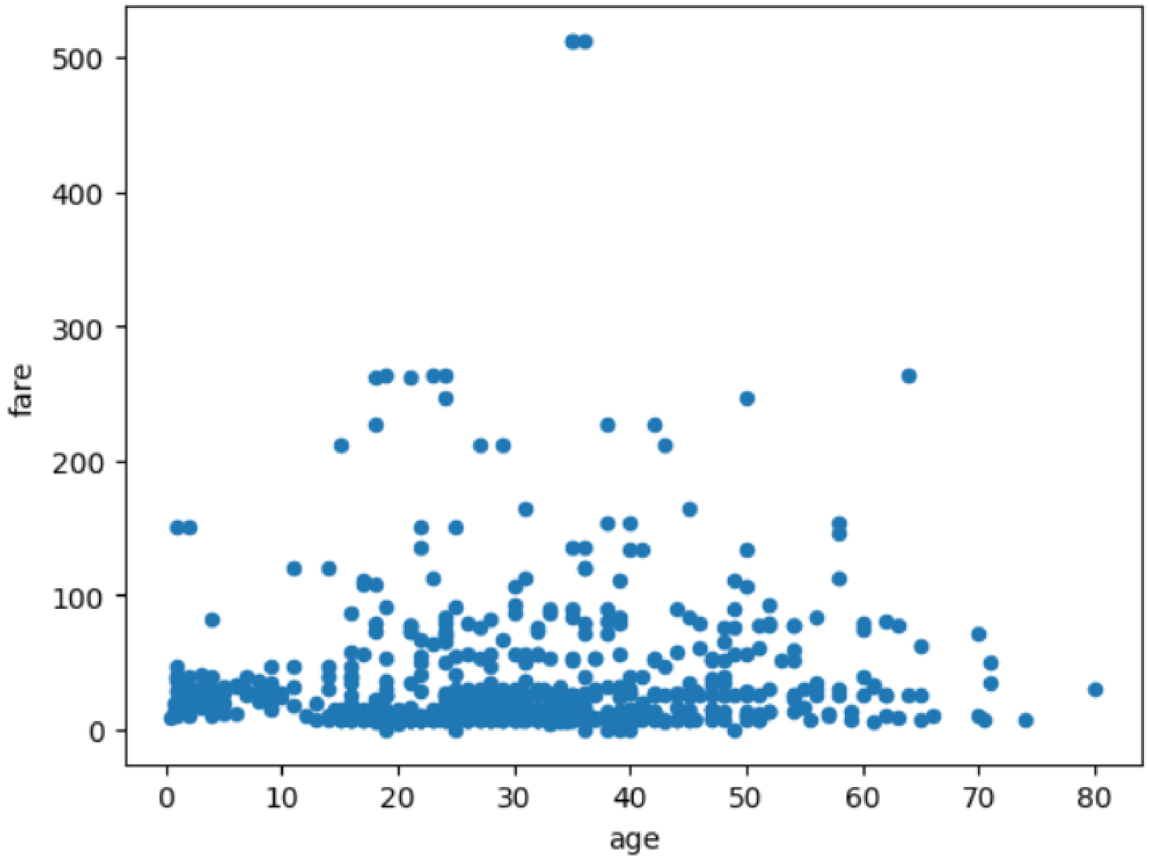
Axes-level properties

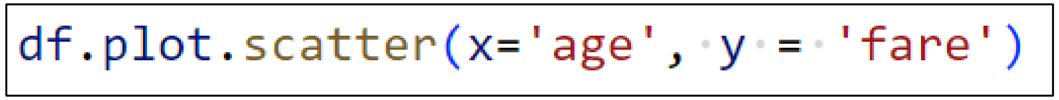


```
ax[0].set_title("Scatterplot")
ax[0].set_xlabel("Age")
ax[0].set_ylabel("Fare")
ax[0].set_ylim(0, 300)
ax[0].grid(False)
ax[0].collections[0].set_color('red')
```

Pandas

Plot methods of DataFrames and Series

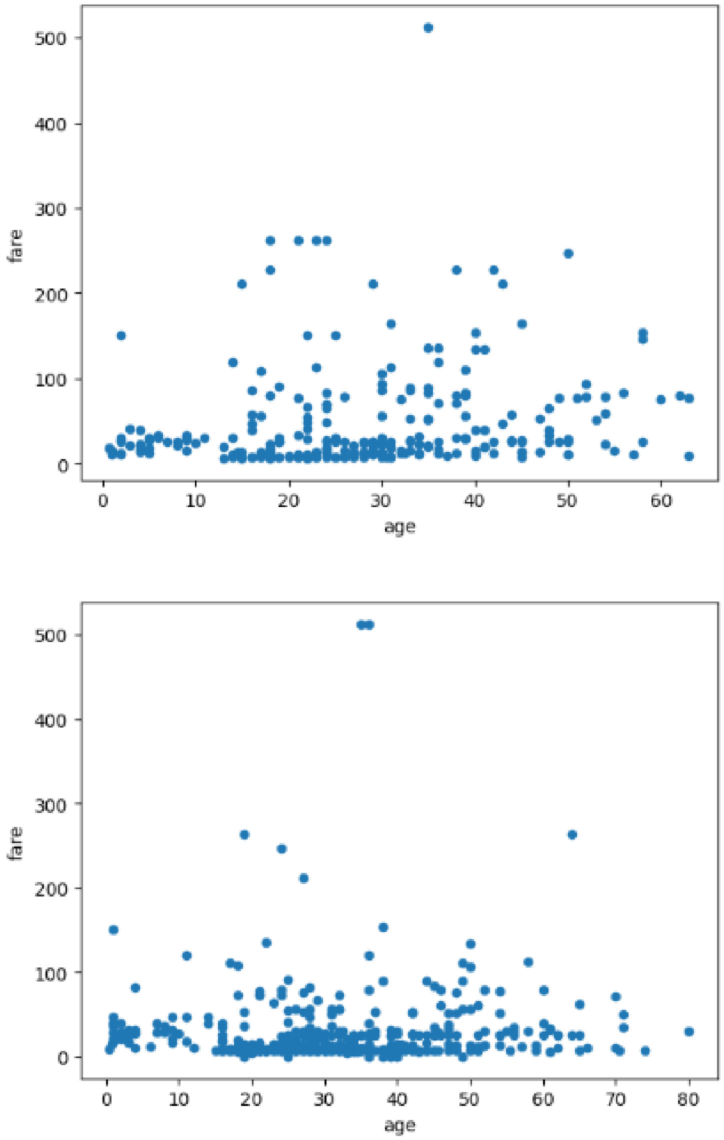




```
df.plot.scatter(x='age', y = 'fare')
```

- **Plot methods directly operating on Pandas DataFrame or Pandas Series**
- **Types**: line, bar, hist, box, kde, area, pie, scatter, hexbin

Plot methods of DataFrames and Series



```
df.groupby('sex').plot.scatter(x='age', y='fare')
```

- **Plot methods directly operating on grouped data**
- Produces a figure containing <u>subplots for each individual group</u>

More resources

- **Packages Overview**
  - https://matplotlib.org/stable/tutorials/index
  - https://seaborn.pydata.org/tutorial.html
  - https://pandas.pydata.org/docs/user_guide/visualization.html

- **Colors**
  - https://matplotlib.org/stable/tutorials/colors/colors.html
  - https://matplotlib.org/stable/tutorials/colors/colormaps.html
  - https://seaborn.pydata.org/tutorial/color_palettes.html