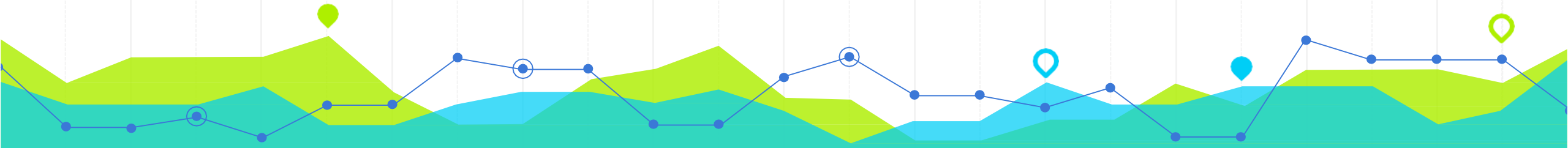
Introduction to Urban Data Science

CRP 4680/5680 Spring 2025

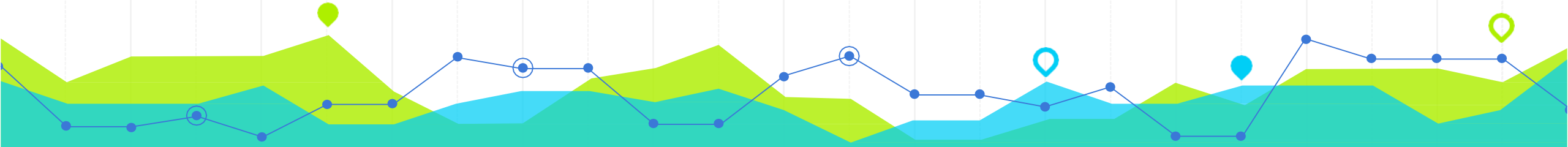


Week 4 Data Visualization

Wenzheng Li
Hazel (Yujin) Lee
02/10/2025

OUTLINE

- Review
- Basics of Data Visualization
- Matplotlib and Seaborn



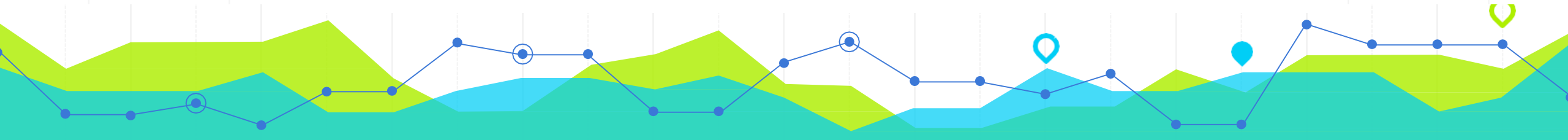
Review 0

Pandas dtype

- Each column/row in a Pandas (and GeoPandas) DataFrame has a data type, called *dtype* attribute.

- Here is the mapping between Pandas dtypes and python data types.
- Note that the object dtype means that the column is a mix of types or it's a string.

Pandas dtype	Python type	Usage
object	str or mixed	Text or mixed numeric and non-numeric values
int64	int	Integer numbers
float64	float	Floating point numbers
bool	bool	True/False values
datetime64	NA	Date and time values
timedelta[ns]	NA	Differences between two datetimes
category	NA	Finite list of text values



Null values

- None means a missing entry, but it's not a numeric type. It is of type **object** and is often found in columns that contain strings or mixed data types.
- NaN (Not a Number) used by Pandas for representing missing data in numeric columns.
- Na is Pandas' newer, more flexible missing data indicator that can be used across different data types.)

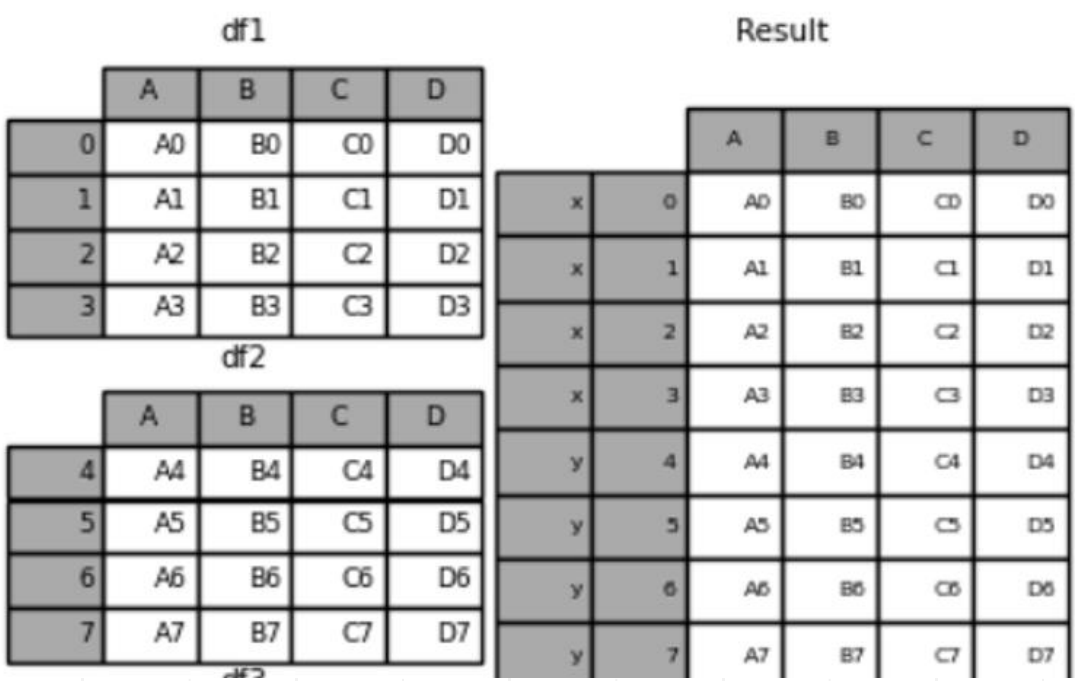
	Column_None	Column_NaN	Column_String	Column_NA
0	1	1.1	apple	1
1	NaN	NaN	banana	NA
2	3	3.3	None	3
3	4	4.4	cherry	4

NaN for Missing Value

- Removing data:
  - If it’s an important cell, we might remove the entire row the cell belongs to.
- Imputing data:
  - We might want to replace it with:
    - The most frequent value (mode), if we think that there's some default value
    - The median value (if you think there are outliers in the sample that might be skewing the mean)
    - The average value (if you don’t want the replaced data to influence your regression values).
  - Fill forward or backward: Fill missing values with the previous or next value (useful in time series data).
  - Or if you have more knowledge of the substantive topic (for ex: body temperature of mammals might typically be XX, but this species, it might be YY)
- Indicate that the data is missing in a new column
- Use linear regression or machine learning to predict the missing value.

Concatenating multiple Dataframes along the row axis (axis = 0)

- concatenating along the rows: joining df2 to df1 vertically using pd.concat(axis=0)
- this means stacking your DataFrames on top of one another. If columns share the same names, they’re combined into a single column; if not, new columns are created and filled with missing values.



	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

		A	B	C	D
x	0	A0	B0	C0	D0
x	1	A1	B1	C1	D1
x	2	A2	B2	C2	D2
x	3	A3	B3	C3	D3
y	4	A4	B4	C4	D4
y	5	A5	B5	C5	D5
y	6	A6	B6	C6	D6
y	7	A7	B7	C7	D7

Merging Dataframes along the column

- **Merging along the columns** means merging DF B to DF A horizontally based on a **merge key** (the column (or set of columns) whose values are used to match rows across the two DFs.).
- Function: pd.merge( )
- **pd.concat()** can also be used to merge along columns by changing the argument axis = 1; **pd.merge()** can **ONLY** be used to merge along the columns.

left

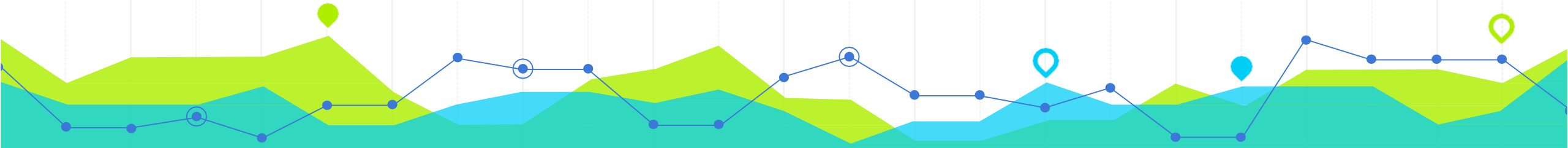
	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

right

	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K2	C2	D2
3	K3	C3	D3

Result

	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3



Basics of Data Visualization (nonspatial)

1

Why Pictures?

What should you do first with data to try to understand it? 3 rules of thumb of data analysis:

- Make a picture—a display of your data will reveal things you are not likely to see in a table of numbers and will help you think clearly about the interesting findings
- Make a picture – a well-designed display will show the important features and patterns and help you find things you did not expect to see– including errors.
- Make a picture- the best way to tell others about your data is with a well-chosen picture

A picture is worth a thousand words is a key theme of doing statistics well!

Types of variables and, thus, types of data

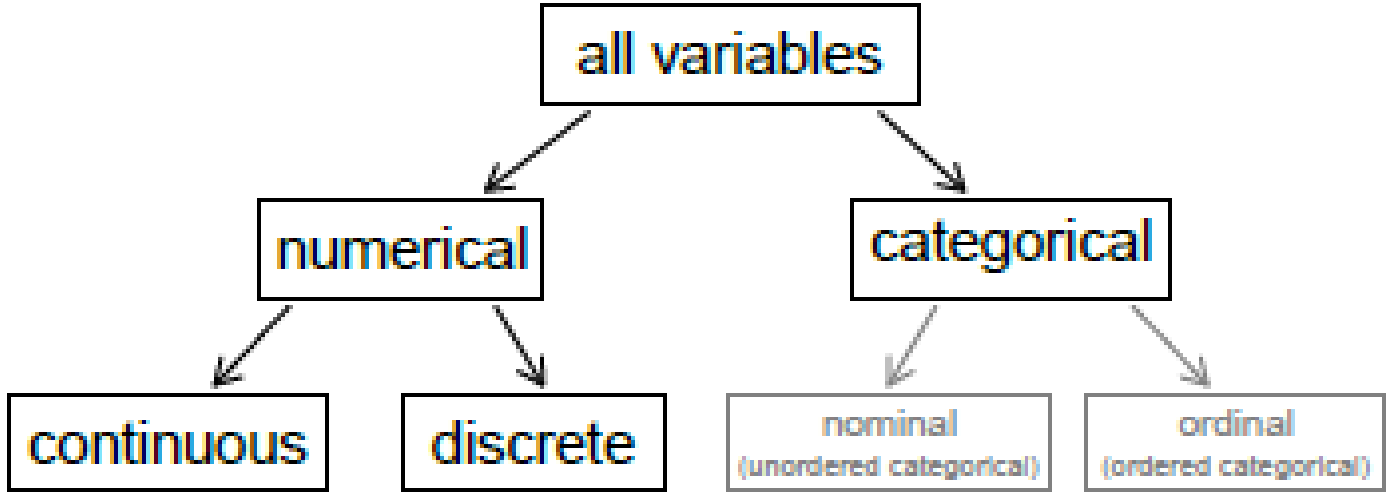


Figure 1.7: Breakdown of variables into their respective types.

It is important to understand the type of data you have or need and its level of measurement because that determines the statistical tools and visualization method you can use for that data type.

Data Types and their levels of measurement

Measurement levels determine the mathematical relations that can be established between values of each data type.

1. Nominal data (also known as: categorical data/qualitative data): Data identifies the group or category. The only relationship that can be meaningfully established is same or different - cannot rank or calculate these categories because they do not have an inherent order or magnitude.

Examples: how do you commute to work?, what is your occupation, where do you live?

2. ***Ordinal data*** – categorical data that has an ordering so we can also use.

Examples: Highest education degree, Likert scales.

Data Types and their levels of measurement

Numerical data/quantitative data – data contains measured numerical values with measurement units. The measurement units provide the meaning for the numbers. We can now add and subtract in a meaningful way.

Numerical data can be categorized as

- Discrete – finite countable number of possible values.

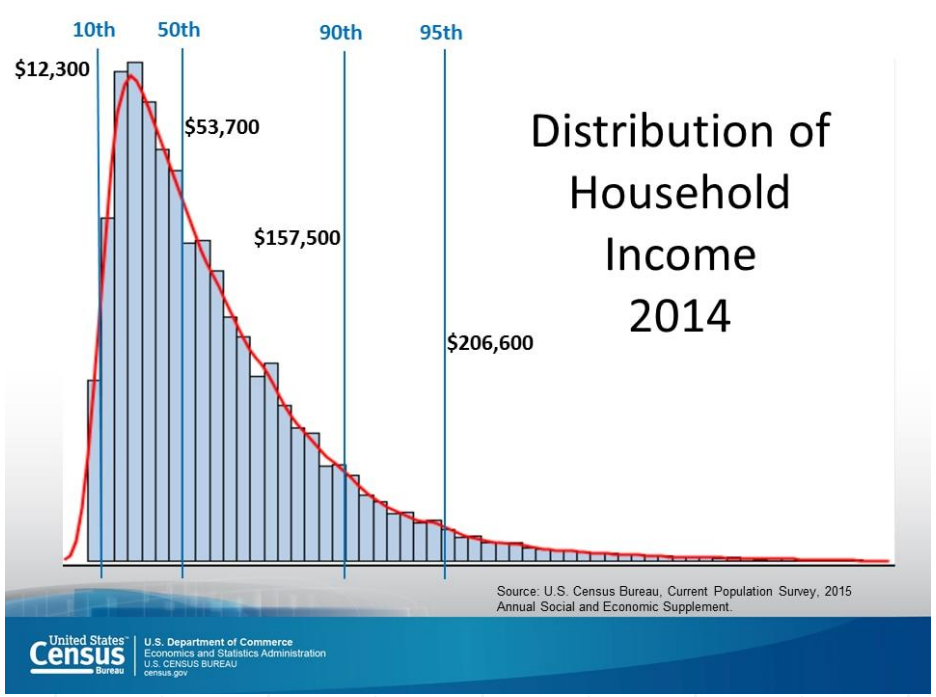
  examples: years of education, number of people unemployed in a neighborhood, number of people in a classroom

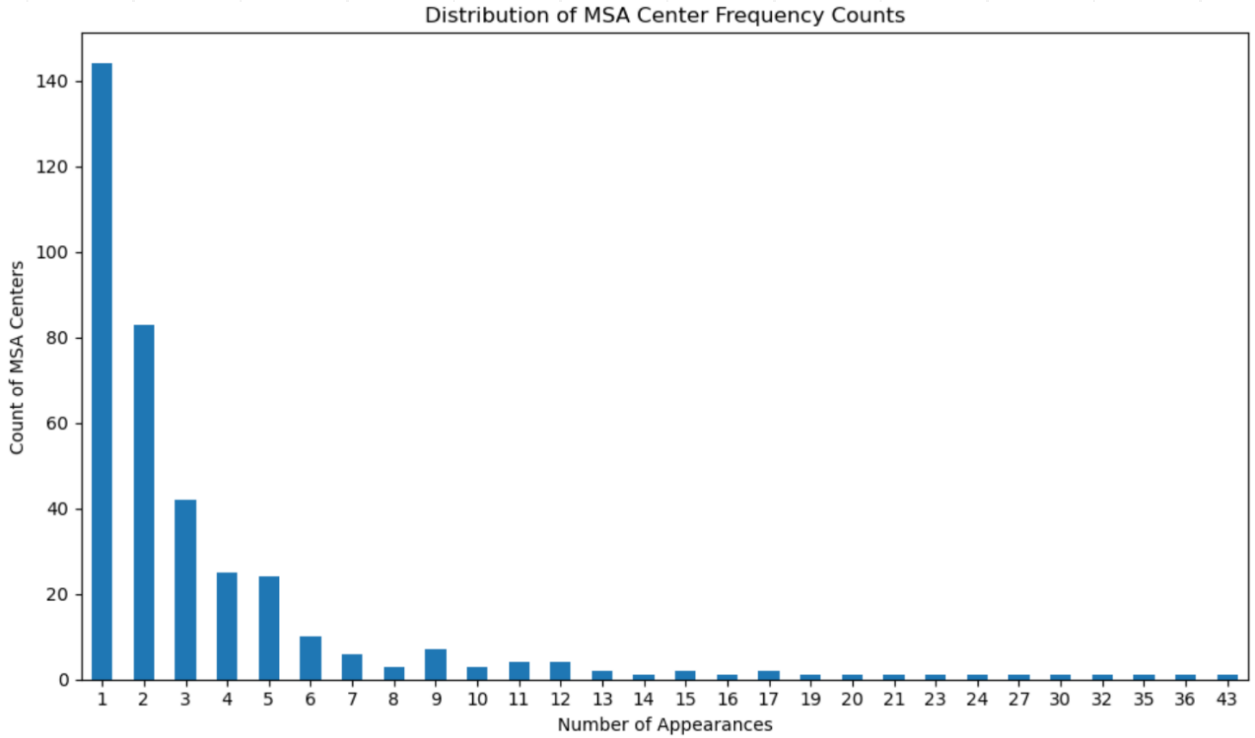
- Continuous – infinite number of values are possible as data can take any value within a range.

  examples: rainfall, height, poverty rates, income, number of people unemployed if it is a large population

Histogram

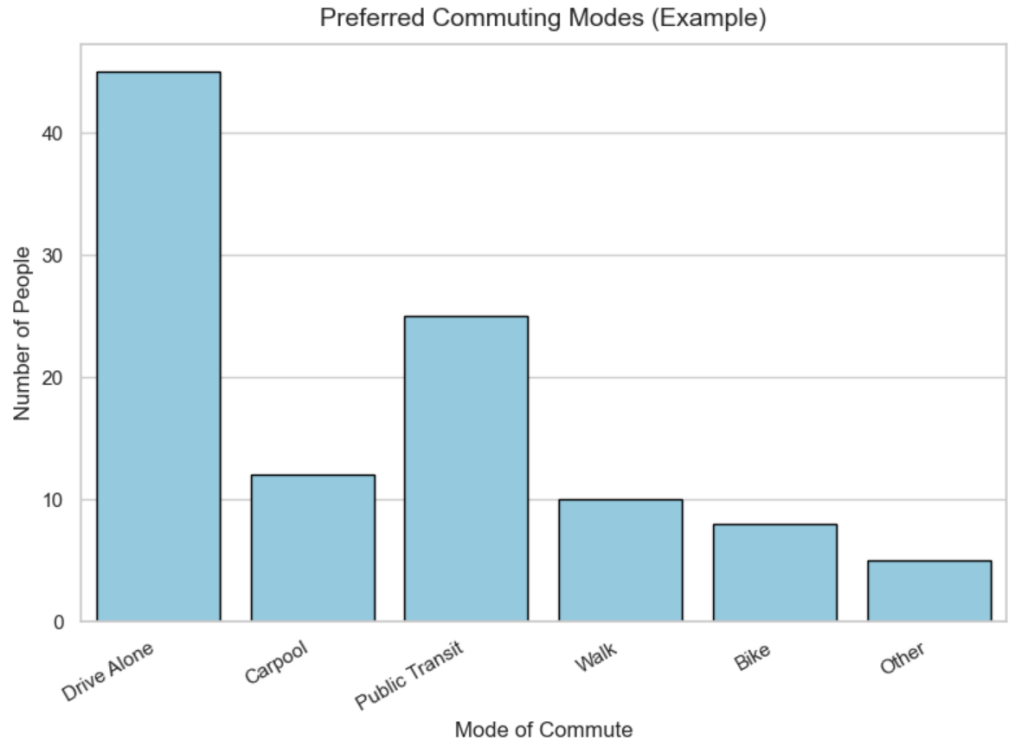
Histograms are primarily used for numerical data, which can be further categorized into continuous and discrete data.

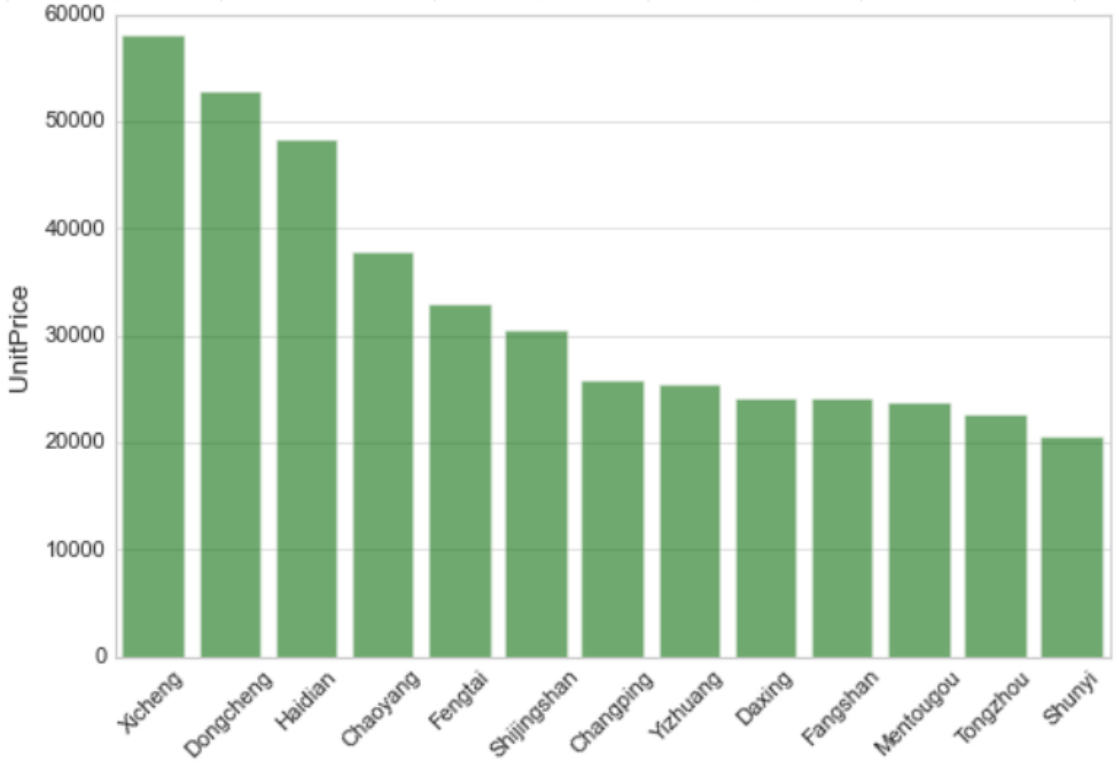




Bar Chart vs. Histogram

A bar chart represents ***categorical data,*** such as different commuting modes.

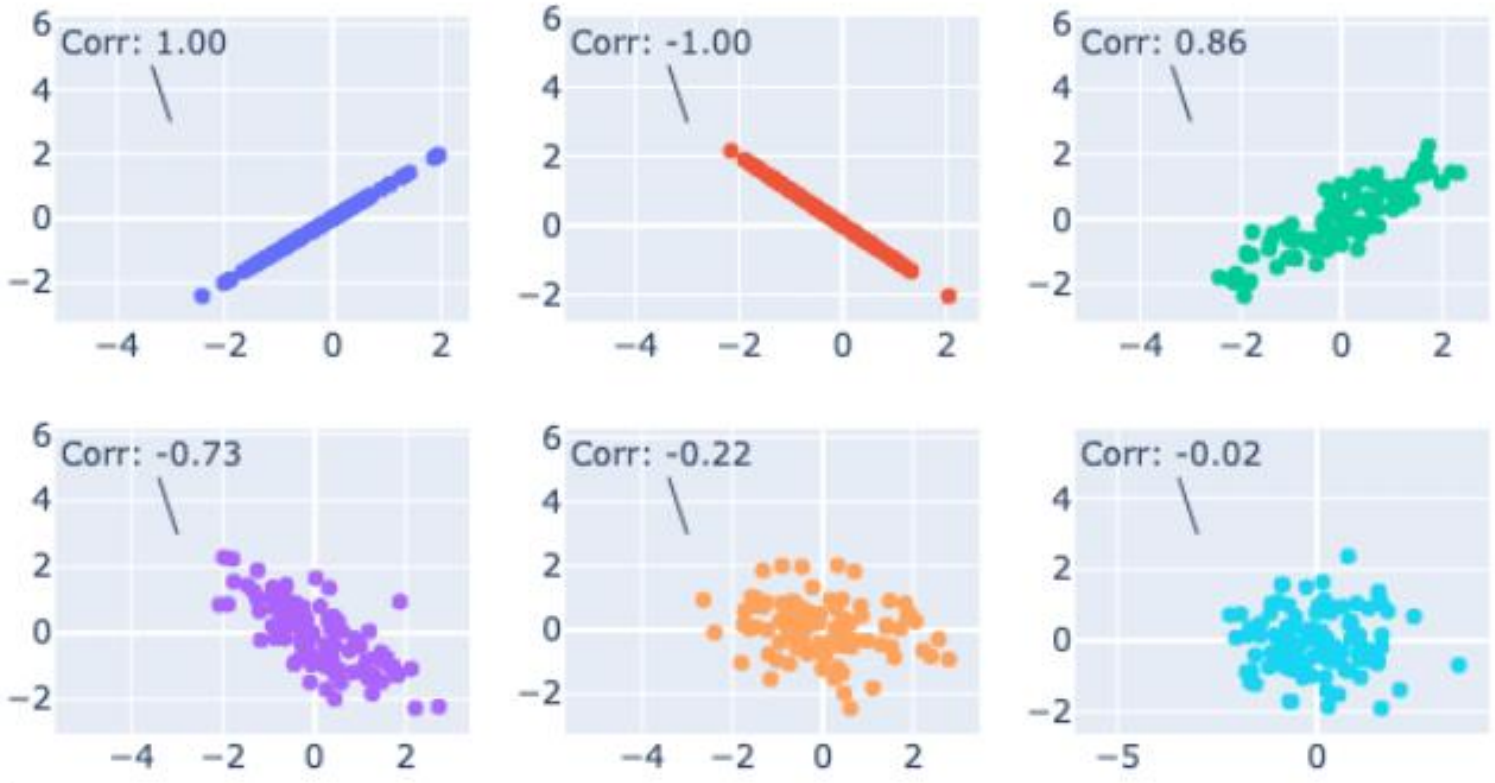




Scatterplot

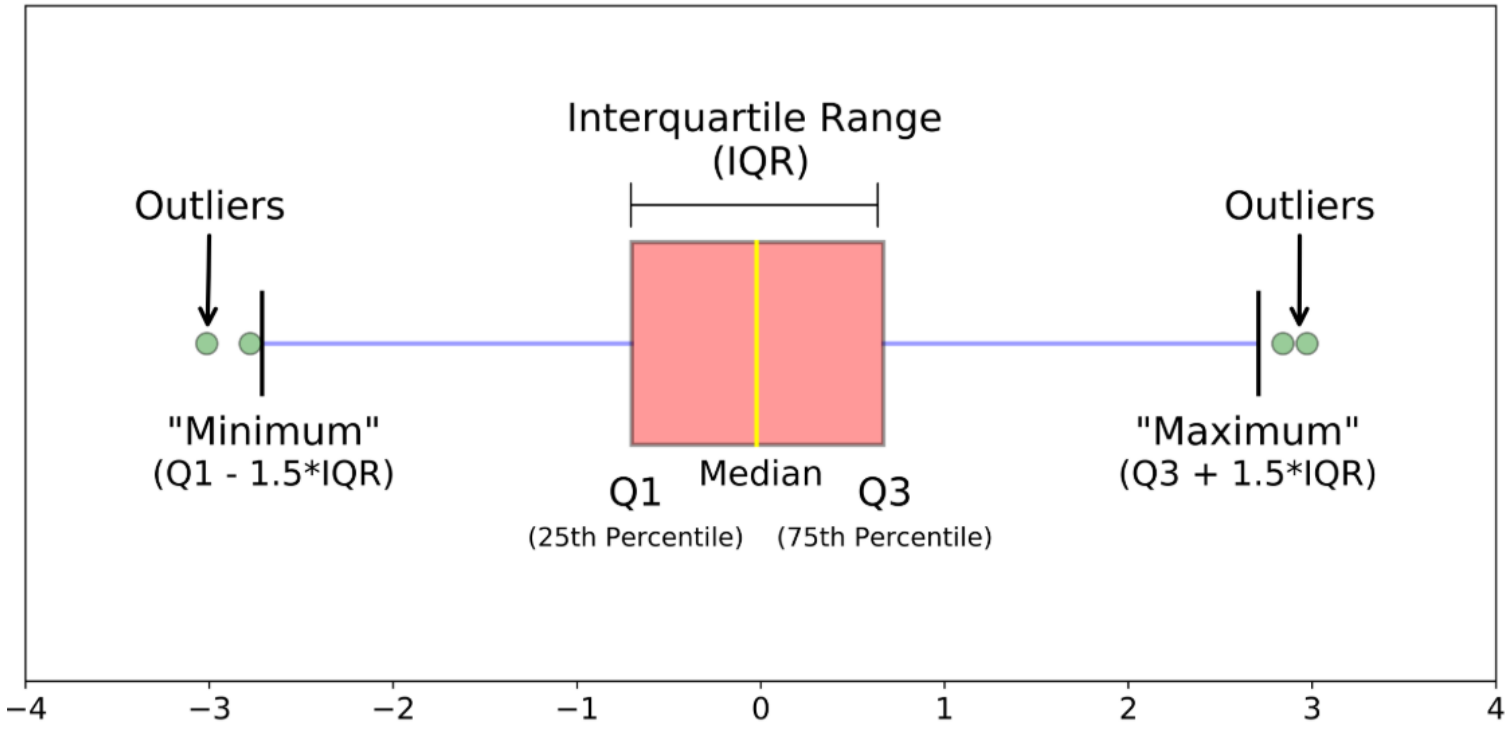
A type of plot that shows the relationship between two **numerical variables**.

- **Points**: Each point represents an observation in the dataset.
- **Axes**: X-axis represents the independent variable; Y-axis represents the dependent variable.



Boxplot

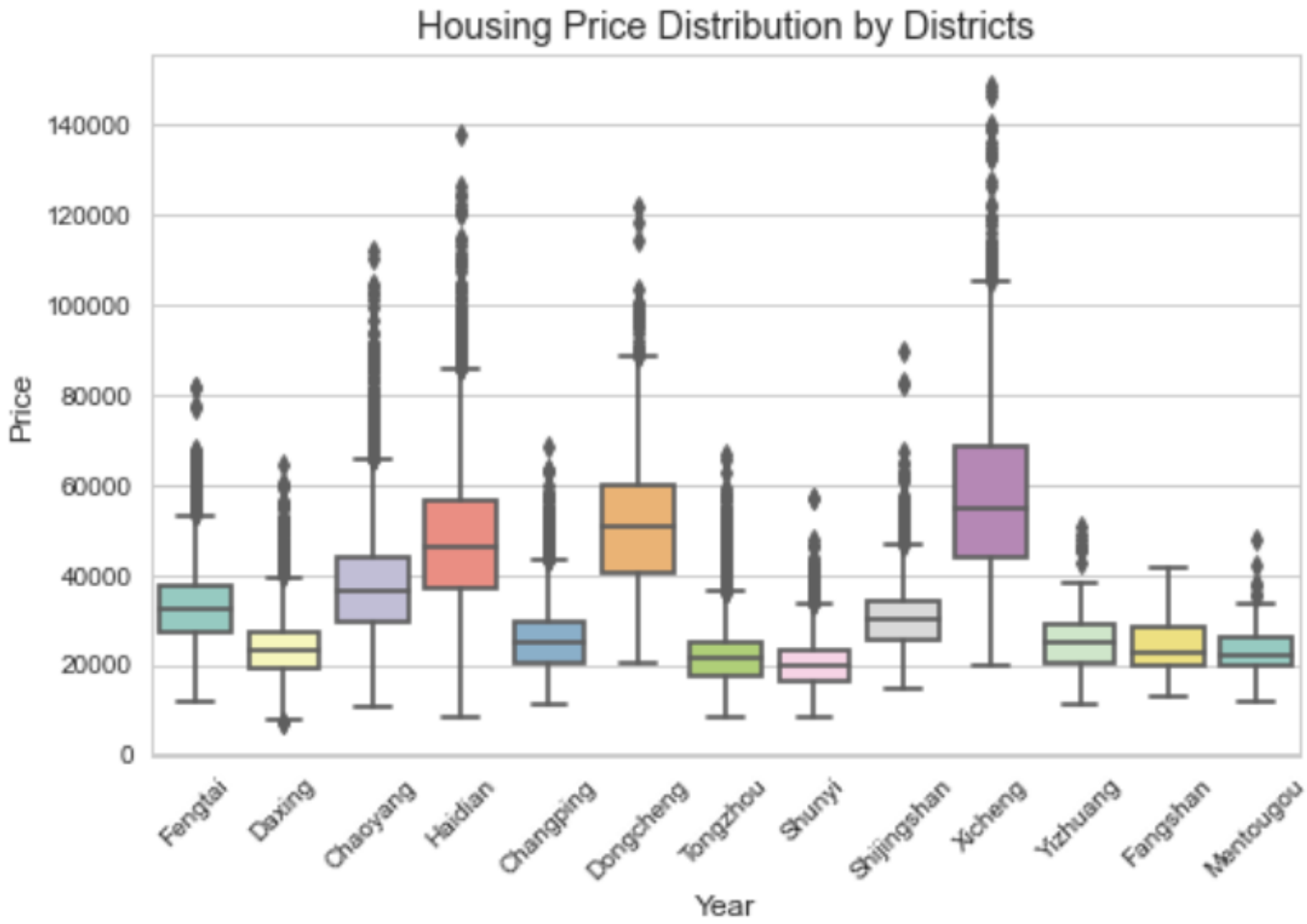
Understanding the distribution and spread of data; Comparing distributions across different groups or categories; Identifying Outliers



Different parts of a boxplot

Interquartile Range (IQR): The range between Q1 and Q3, representing the middle 50% of the data.

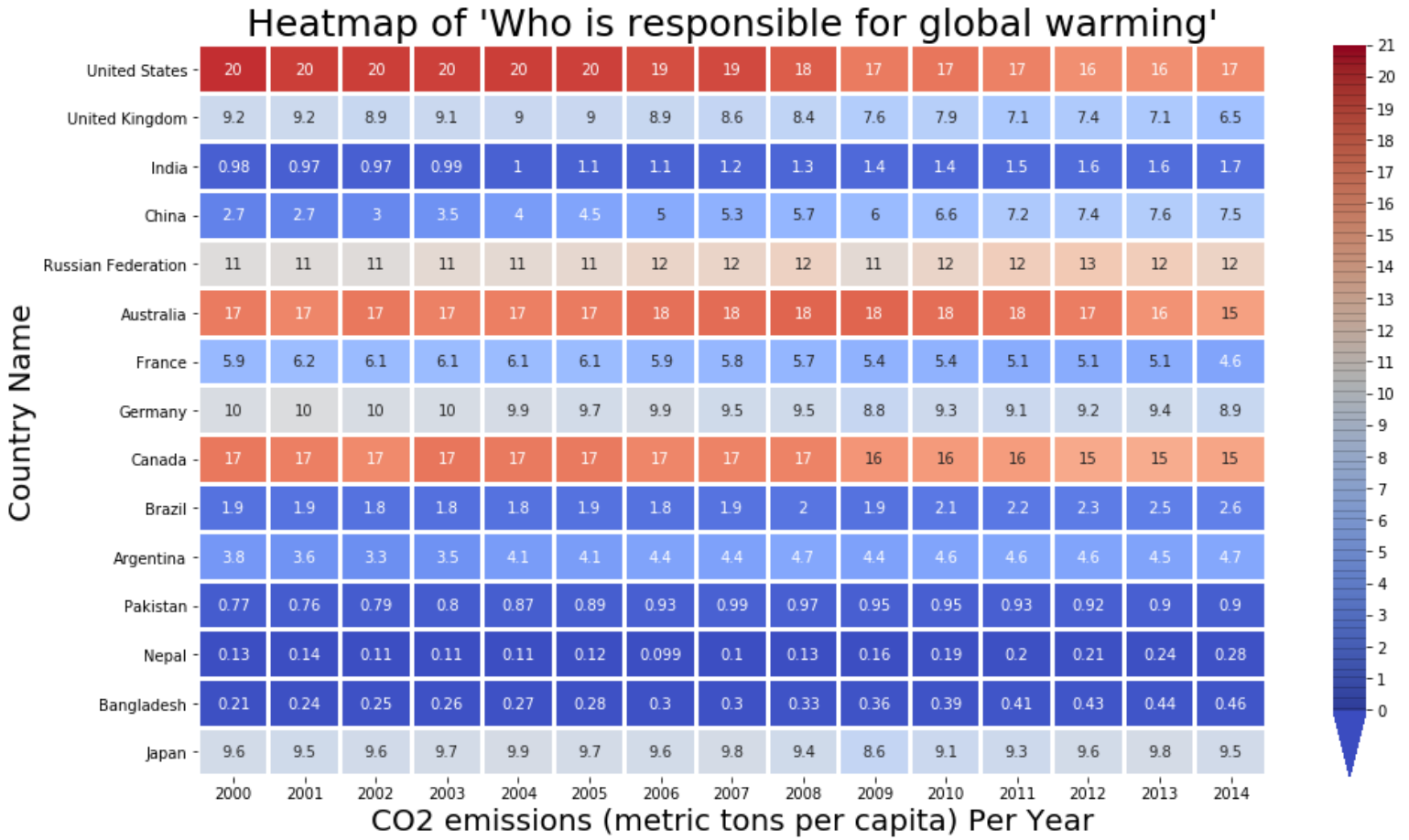
Boxplot

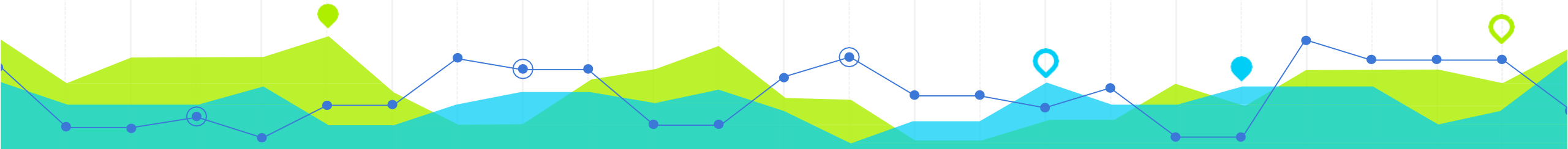

Housing Price Distribution by Districts
Price
140000
120000
100000
80000
60000
40000
20000
0
Fengtai
Daxing
Chaoyang
Haidian
Changping
Dongcheng
Tongzhou
Shunyi
Shijingshan
Xicheng
Yizhuang
Fangshan
Mentougou
Year

Heatmap

A heatmap represents data values as colors in a matrix format.

- **Purpose**: Used to quickly identify patterns, correlations, and anomalies in data.
- **Components**:
  - **Color Scale**: Represents the range of data values. Each color corresponds to a specific data value.
  - **Matrix**: Grid of cells where each cell's color represents a data value.





Python Visualization libraries 2

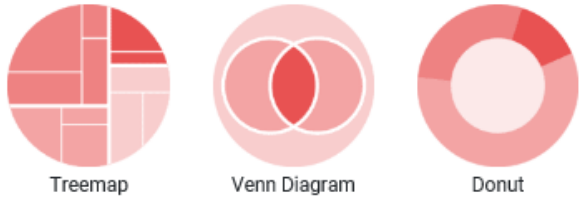
Python Graph Gallery (https://python-graph-gallery.com/)

Distribution

Correlation

Ranking

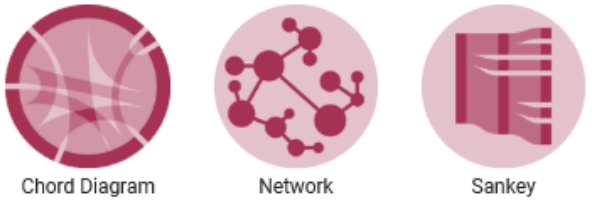
Part Of A Whole



Evolution

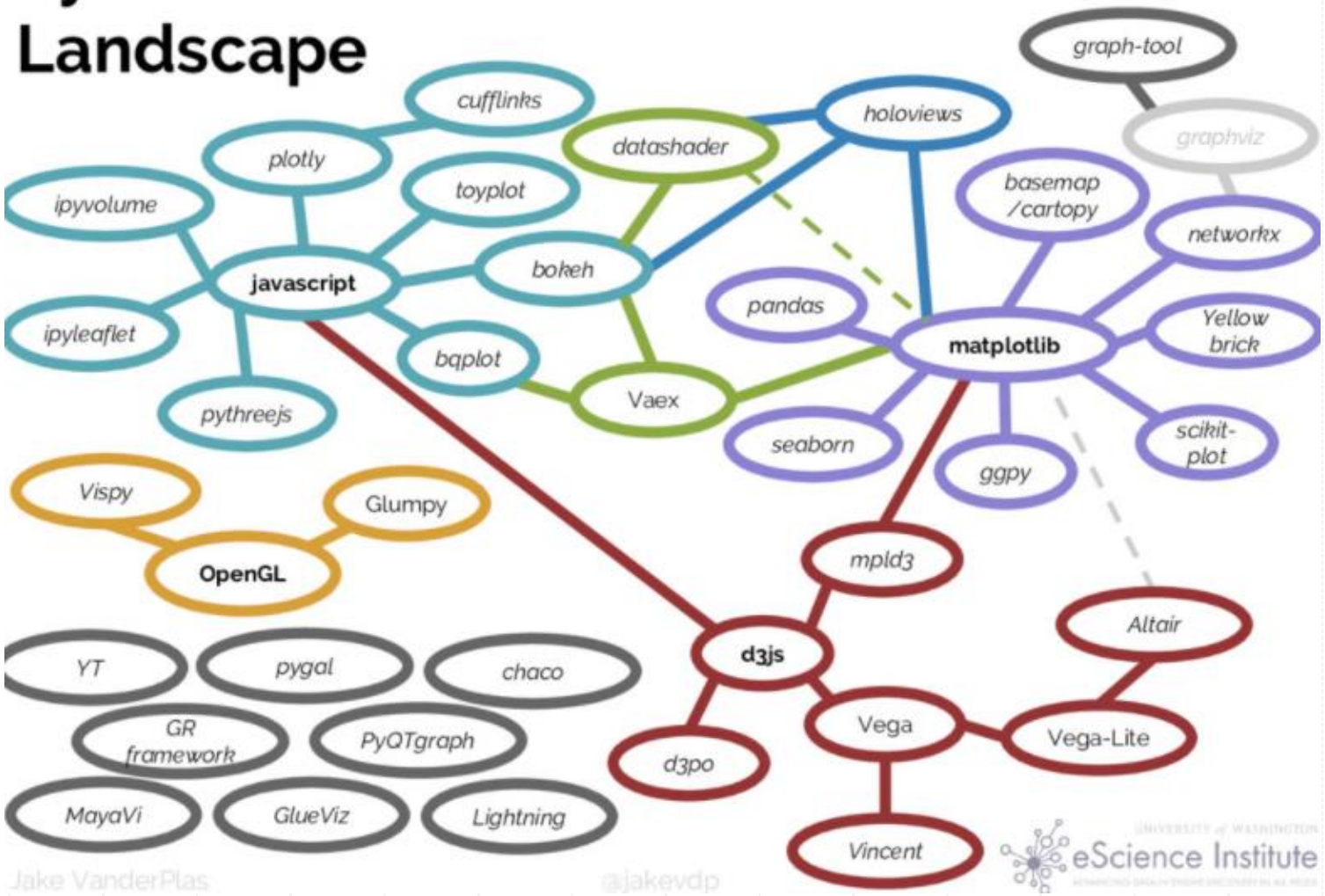
Map

Flow



General Knowledge

Python's Visualization Landscape



Matplotlib is at the core of the Python visualization. Many libraries are built upon matplotlib…

JavaScript: the backbone for interactive and web-based visualization libraries

d3.js: a JavaScript library that powers interactive, web-based visualizations.

OpenGL: leverage GPU power for rendering complex, large-scale visualizations.

https://python-graph-gallery.com/best-dataviz-packages/

Python Visualization libraries

Core packages


Matplotlib


Plotly

Plotnine

Python's ggplot2


Seaborn

aesthetically pleasing statistical charts

Interactivity


Plotly


Bokeh


Vega-Altair

Geospatial packages


Geopandas


Geoplot

Cartopy


Basemap


Folium

interactive maps

Specific chart types


NetworkX

Wordcloud


PyWaffle

https://python-graph-gallery.com/best-dataviz-packages/

Resources

Matplotlib gallery - a vast collection of plots created with Matplotlib, from basic line graphs to advanced visualizations.
https://matplotlib.org/stable/gallery/index.html

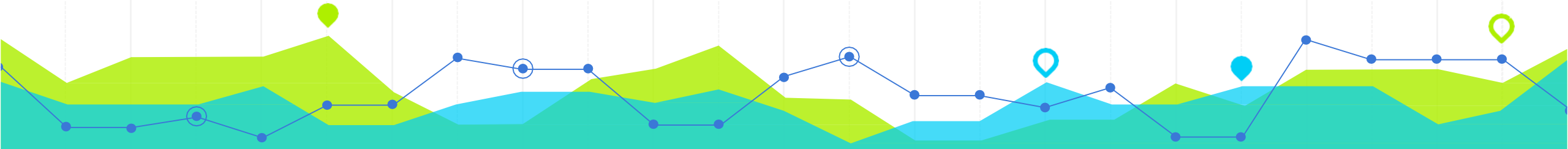
Seaborn - Focused on statistical visualization https://seaborn.pydata.org/examples/index.html

Plotly - interactive plotting, include maps. https://plotly.com/python/

Bokeh - interactive and web-based plots. It's ideal for dynamic and visually appealing charts.
https://docs.bokeh.org/en/latest/docs/gallery.html

NetworkX - network visualizations, useful for visualizing relationships and connections.
https://networkx.github.io/documentation/stable/auto_examples/index.html

Kaggle - https://www.kaggle.com/models

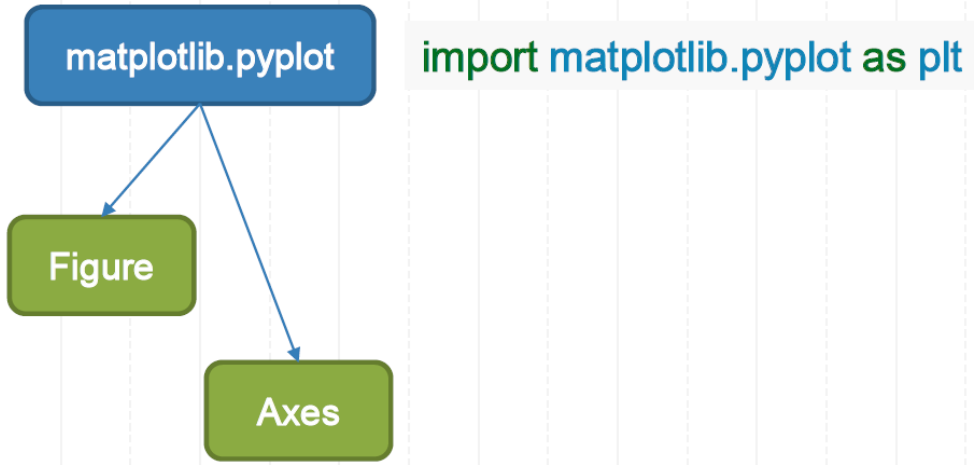


Python Visualization - matplotlib

3

Introduction of Matplotlib

- Everything in matplotlib is organized in a **hierarchy.**




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

A basic canvas generated by the matplotlib library consists of the following two components:

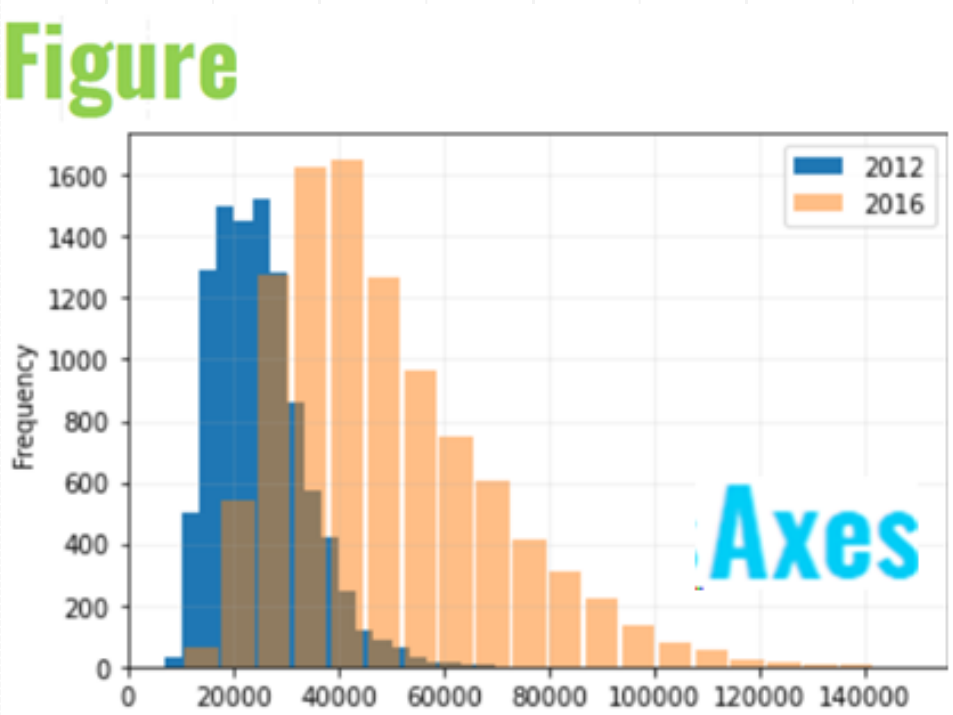
figure:

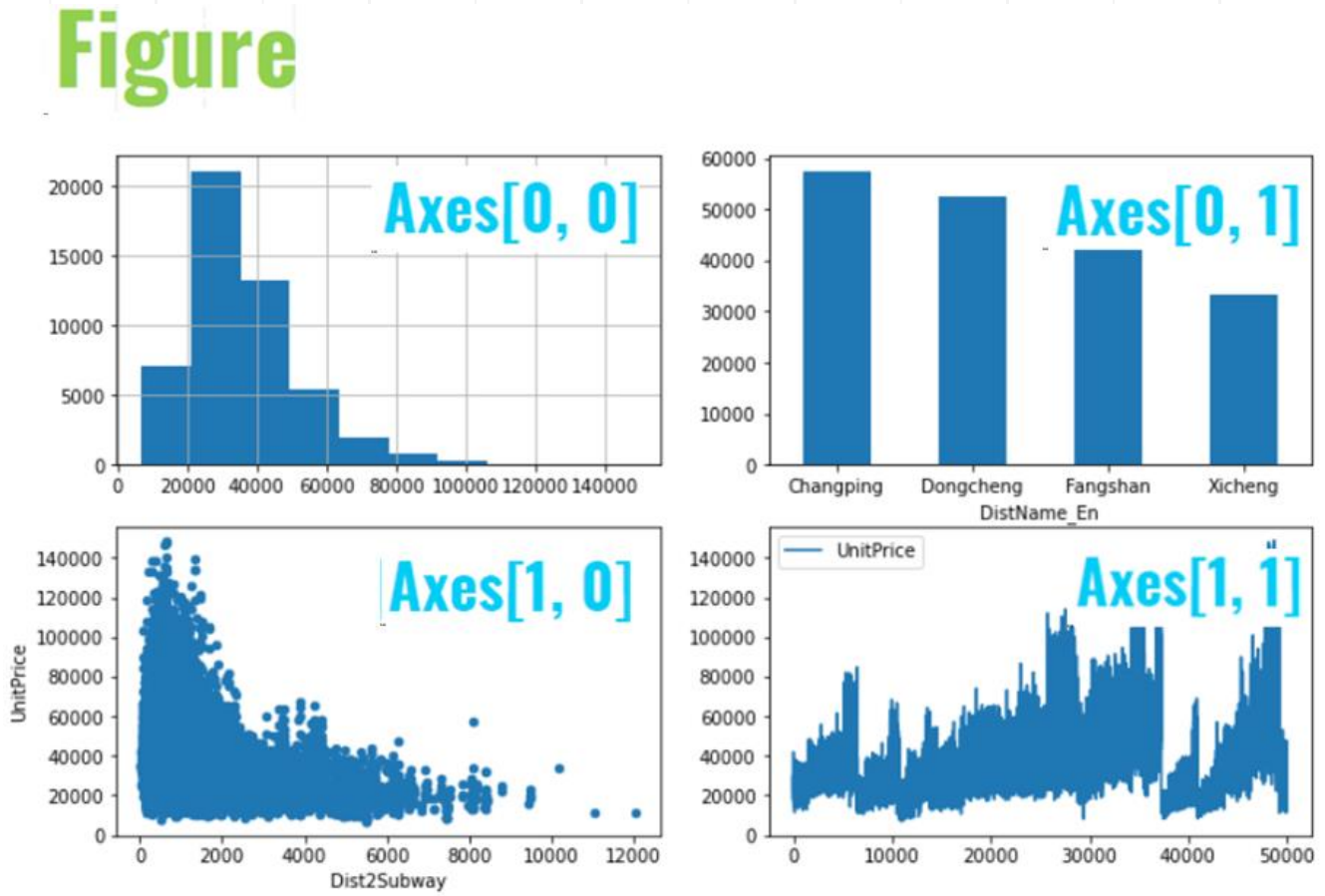
- Represents the entire canvas for the visualization.
- Can contain one or more **Axes** (subplots), along with additional elements like titles and legends.

Axes:

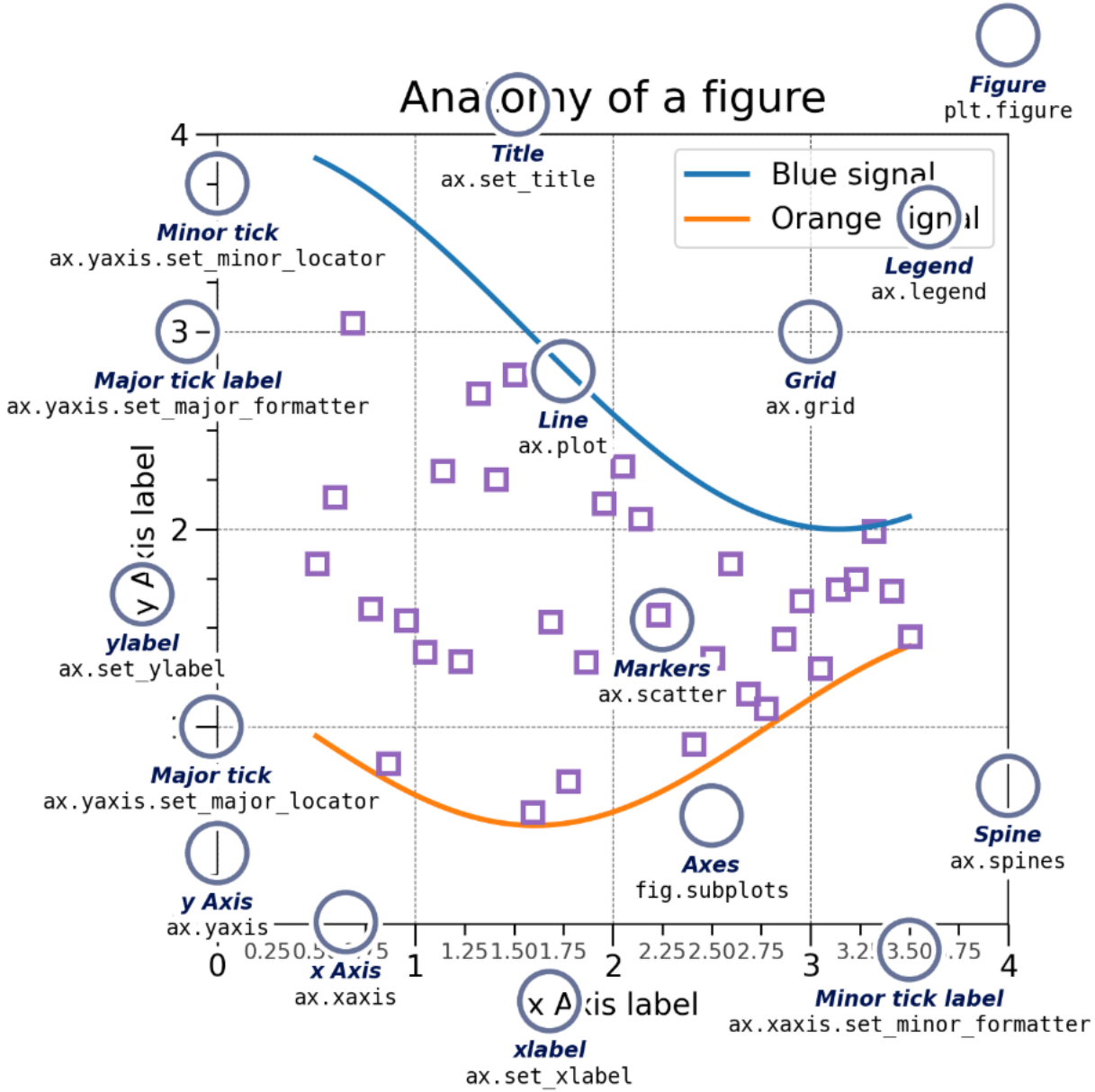
- This is what you think of as ‘a plot’, it is the region of the image with the data space.
- a given Axes object can only be in one Figure.
- an Axes contains two (or three in the case of 3D) Axis objects, along with labels and title.

Figure and axes





Elements in a Figure



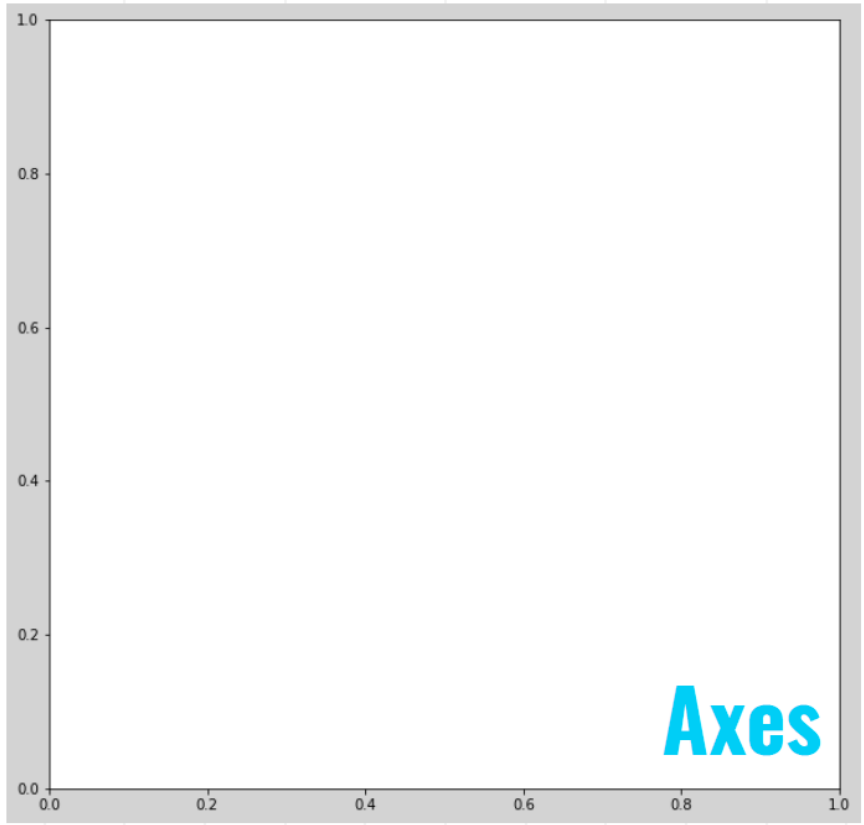
A Figure with one Axes

To create a basic plot, we need to define the **figure** and **axes**:

```
fig, ax = plt.subplots(1, 1, facecolor = 'lightgrey')
```

Figure

Axes



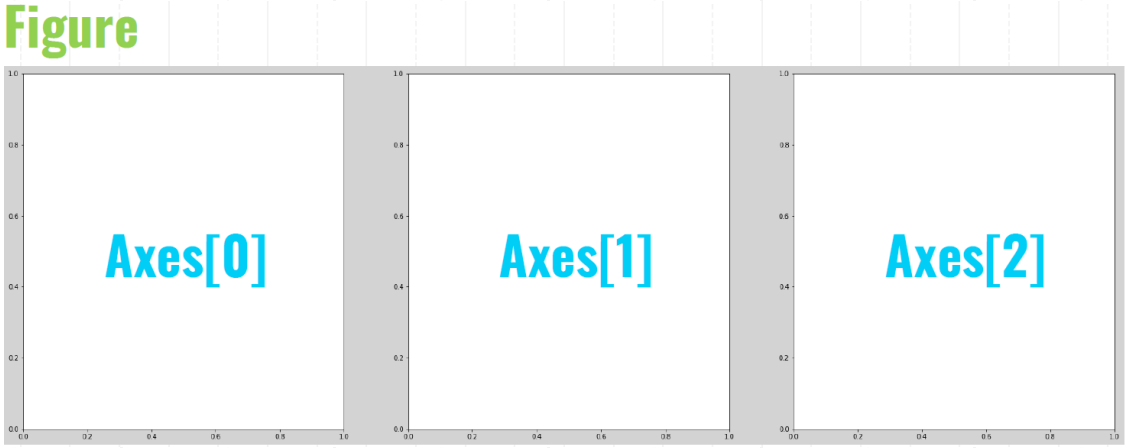
Figure

A combined Figure includes multiple Axes

To create a basic plot, we need to define the **figure** and **axes**:

```
fig, ax = plt.subplots()  # a figure with a single Axes
fig, axs = plt.subplots(1, 3)  # a figure with a 1x3 grid of Axes (1 row, 3 columns just like the figure one)
fig, axs = plt.subplot(2,2) # a figure with a 2x2 grid of Axes (2 row, 2 columns just like the figure two)
```

The whole figure consists of three axes:



The whole figure consists of four axes:

Help Materials:

- The codes for various plotting examples: https://matplotlib.org/gallery/index.html
- For matplotlib cheat sheet: https://github.com/matplotlib/cheatsheets/blob/master/cheatsheets.pdf
- The Python Graph Gallery: https://python-graph-gallery.com/
- Seaborn gallery: https://seaborn.pydata.org/tutorial.html
- Kaggle: https://www.kaggle.com/

Data Insights & Stories

- Census Bureau Infographics & Visualizations Gallery: https://www.census.gov/library/visualizations.html