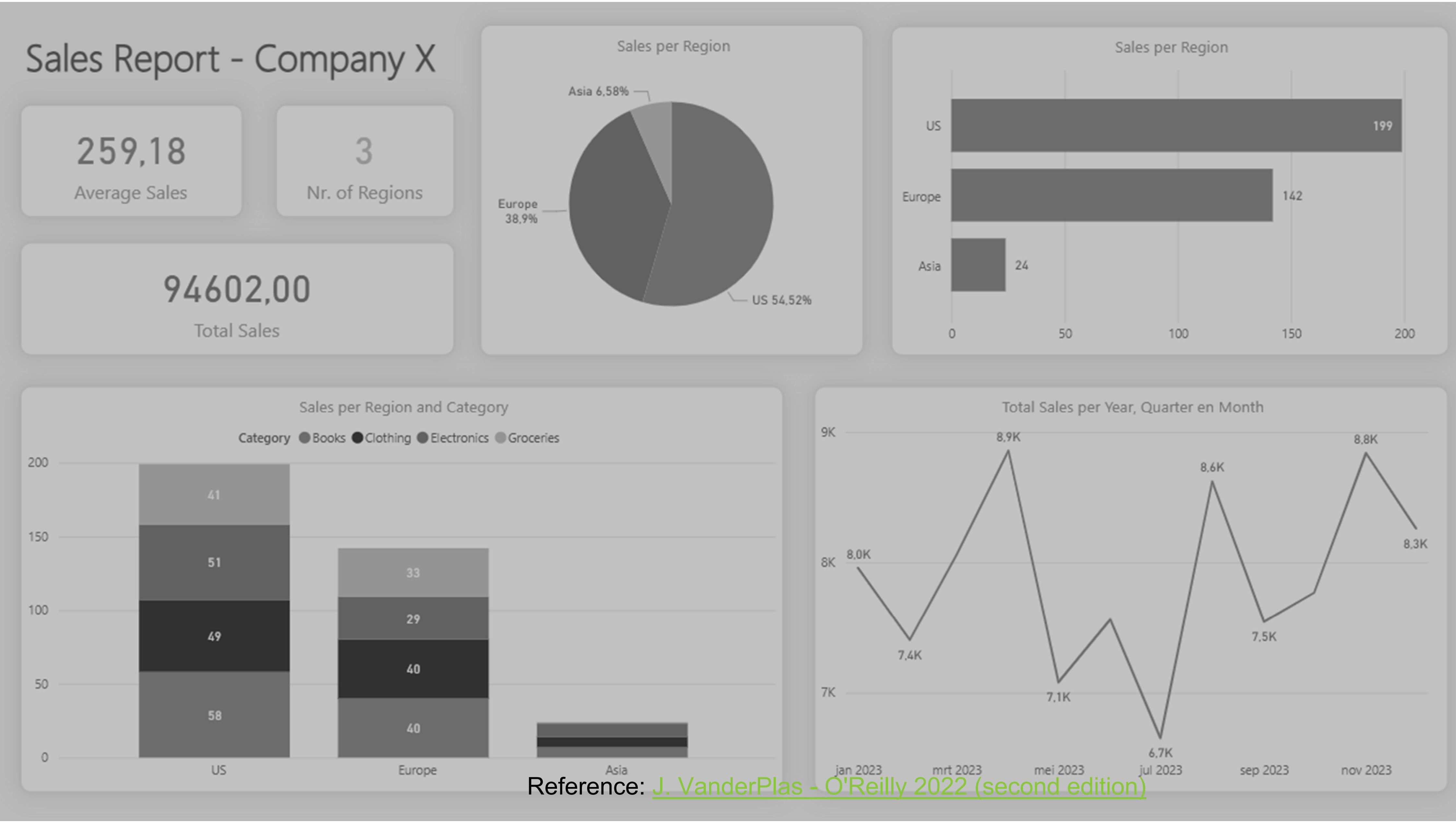


Sales Report - Company X

259,18
Average Sales

3
Nr. of Regions

94602,00
Total Sales

Reference: J. VanderPlas - O'Reilly 2022 (second edition)

Agenda

1. Introduction to Data visualization
2. Introduction to Matplotlib
3. Introduction to Seaborn
4. Types of Data Visualizations
5. Which Data Visualization to Choose

1

Introduction to data visualization

Intro to Data Visualization

- Purpose of Data Visualization:
  - Transforming data into a visual context.
  - Making data accessible, understandable, and actionable.
- Importance in Data Analysis:
  - Identifies trends, patterns, and outliers.
  - Simplifies complex data sets.

2

Introduction to matplotlib

Intro to Matplotlib

- What is Matplotlib?
  - A powerful plotting library in Python.
  - Provides comprehensive control over plot elements.

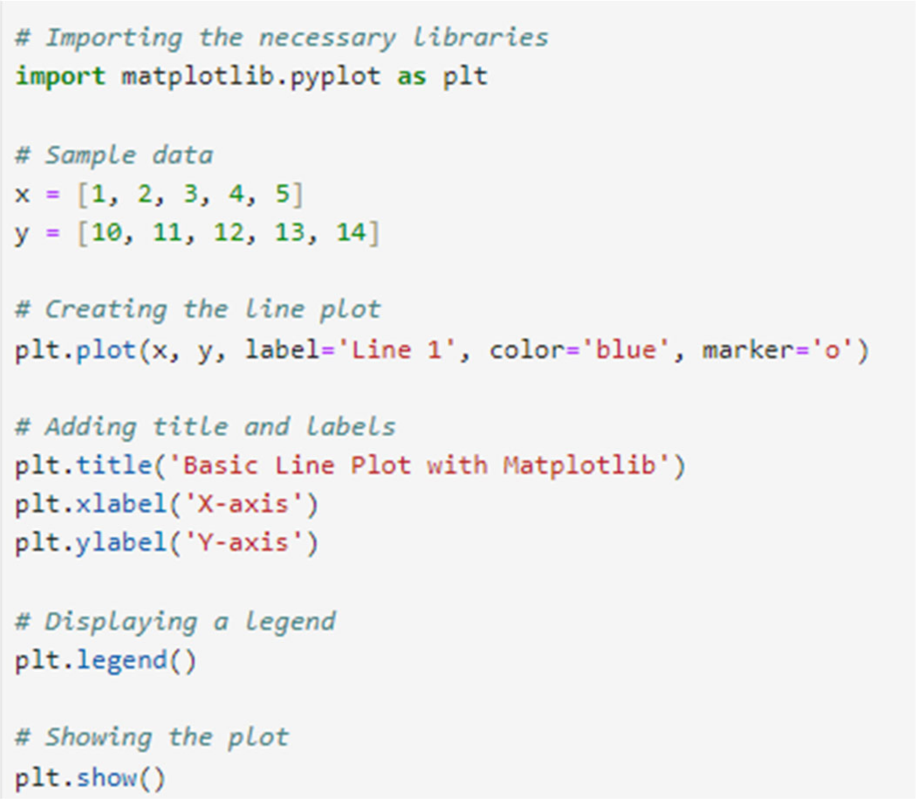
- Key Features:
  - Highly customizable plots.
  - Large variety of plot types.

Intro to Matplotlib

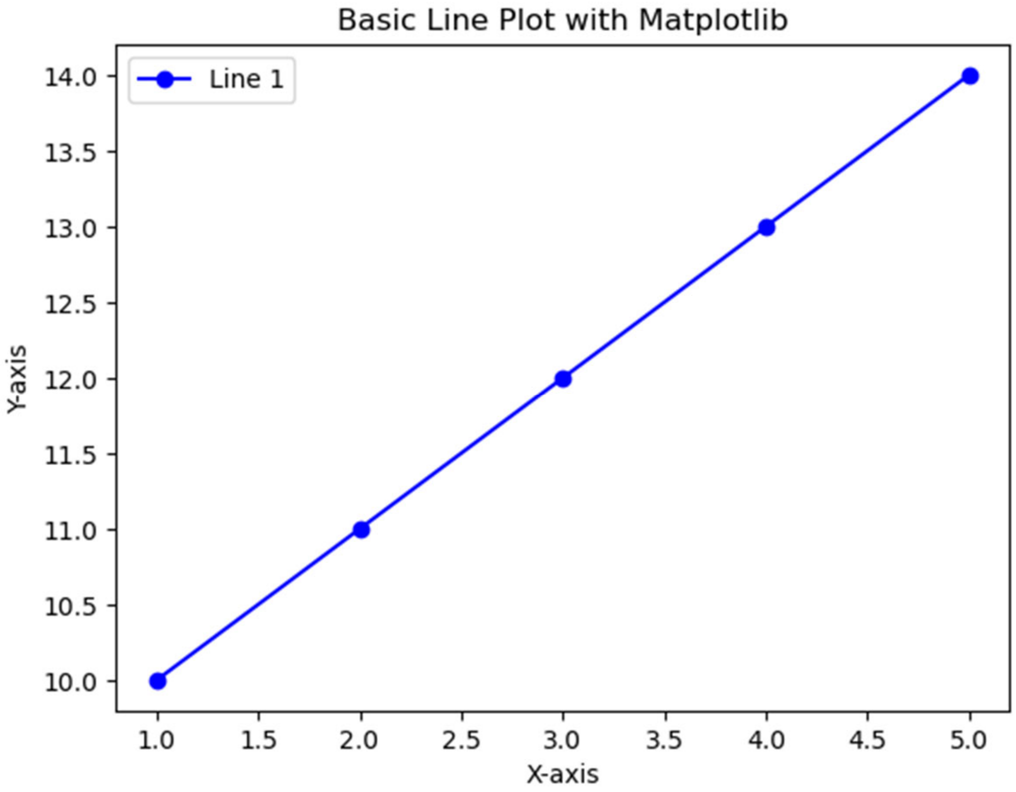
Setting up Matplotlib

- Installation: `pip install matplotlib`
- Import: `import matplotlib.pyplot as plt`

Intro to Matplotlib



```
# Importing the necessary libraries
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [10, 11, 12, 13, 14]

# Creating the line plot
plt.plot(x, y, label='Line 1', color='blue', marker='o')

# Adding title and labels
plt.title('Basic Line Plot with Matplotlib')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Displaying a legend
plt.legend()

# Showing the plot
plt.show()
```



3

Introduction to seaborn

Intro to Seaborn

- What is Seaborn?
  - Built on top of Matplotlib, focusing on statistical data visualization.
  - Provides an aesthetically pleasing interface and advanced plots.

- Key Features:
  - Themes for styling plots.
  - Simplified syntax for complex visualizations.

Matplotlib vs. Seaborn

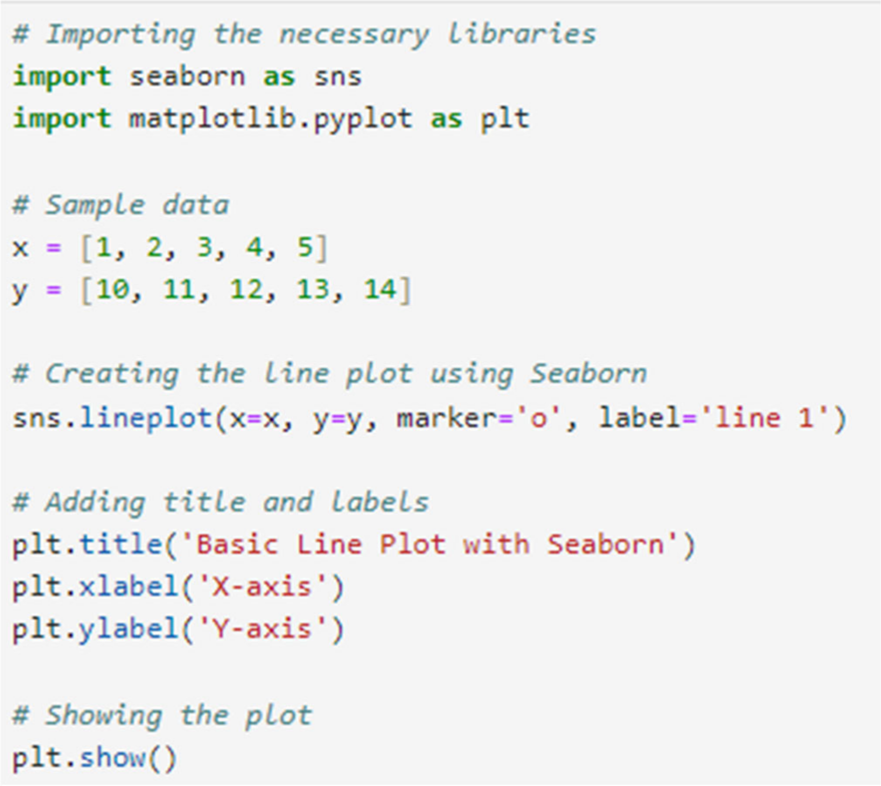
- Matplotlib:
  - More control,
  - detailed customization.
- Seaborn:
  - Better default aesthetics,
  - easier to use for statistical plots.

Intro to Seaborn

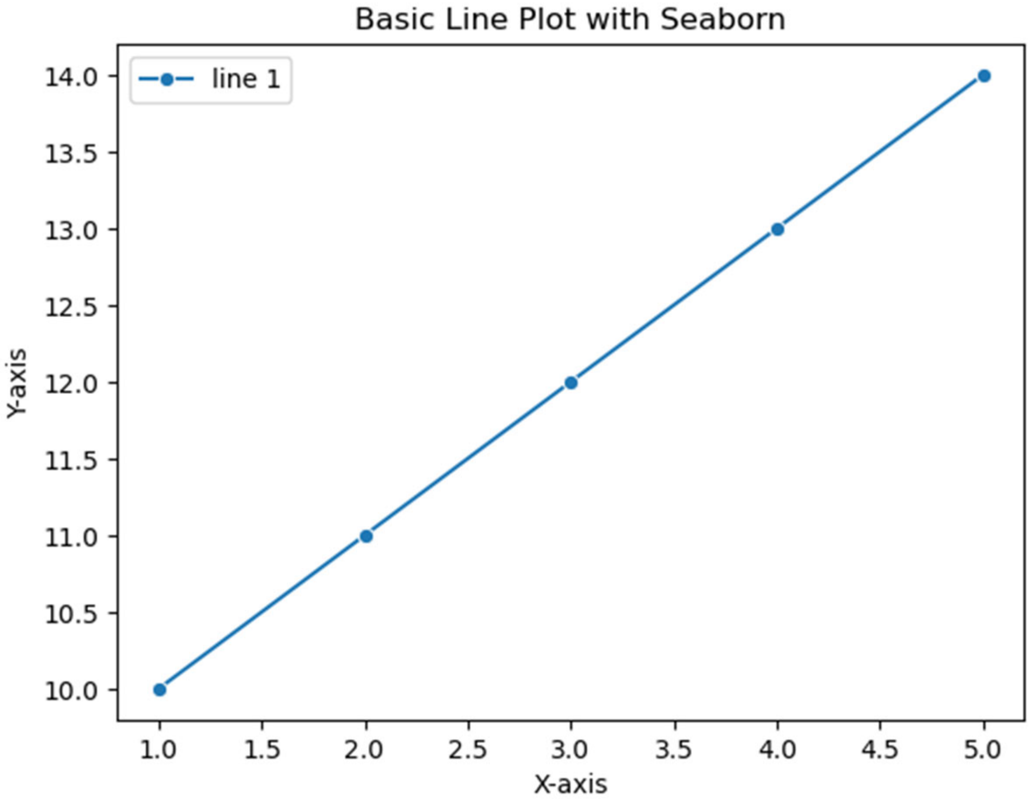
Setting up Seaborn:

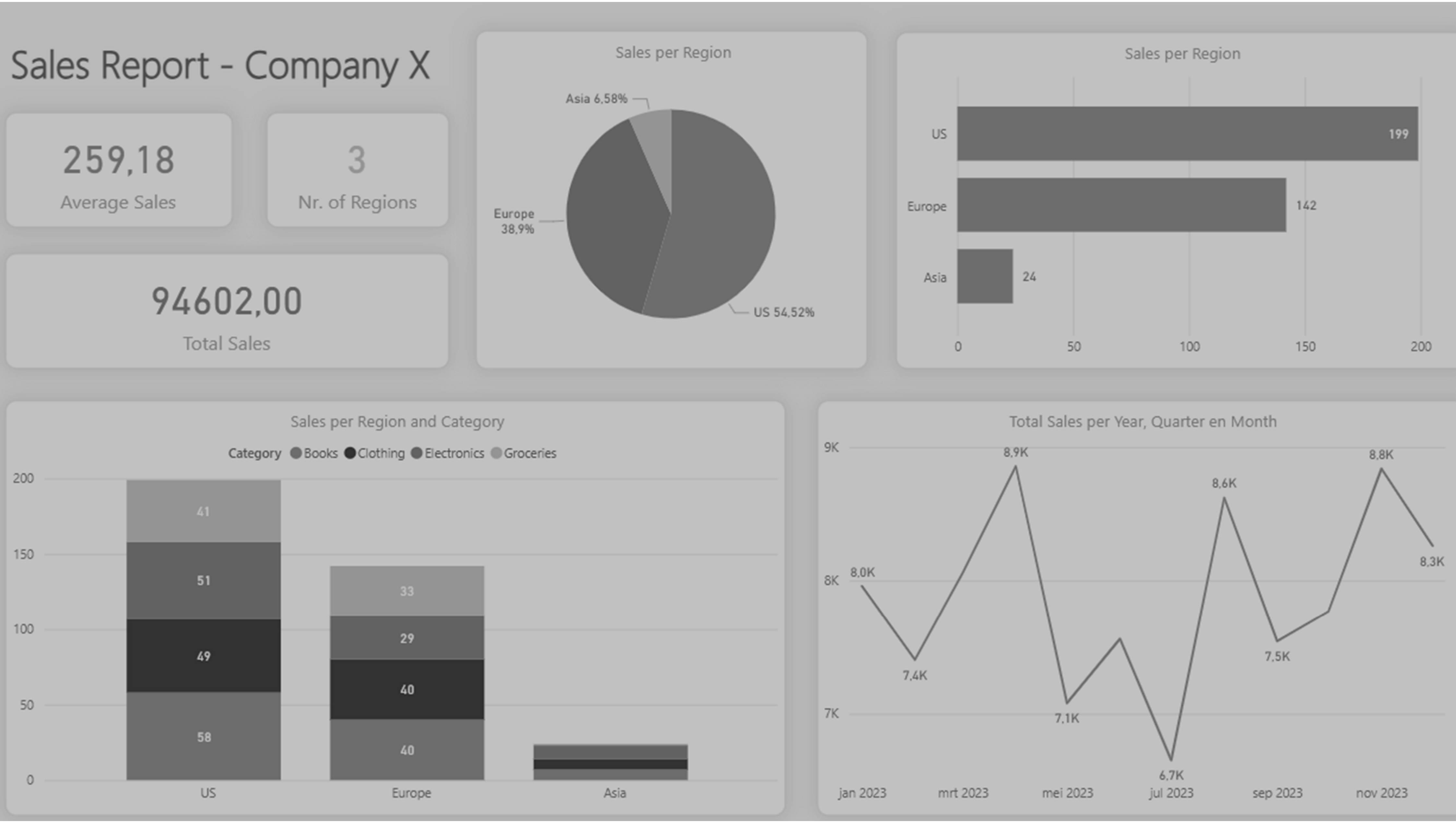
- Installation: pip install seaborn
- Import: import seaborn as sns

Intro to Seaborn



```
# Importing the necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [10, 11, 12, 13, 14]

# Creating the line plot using Seaborn
sns.lineplot(x=x, y=y, marker='o', label='line 1')

# Adding title and labels
plt.title('Basic Line Plot with Seaborn')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Showing the plot
plt.show()
```





Sales Report - Company X

259,18
Average Sales

3
Nr. of Regions

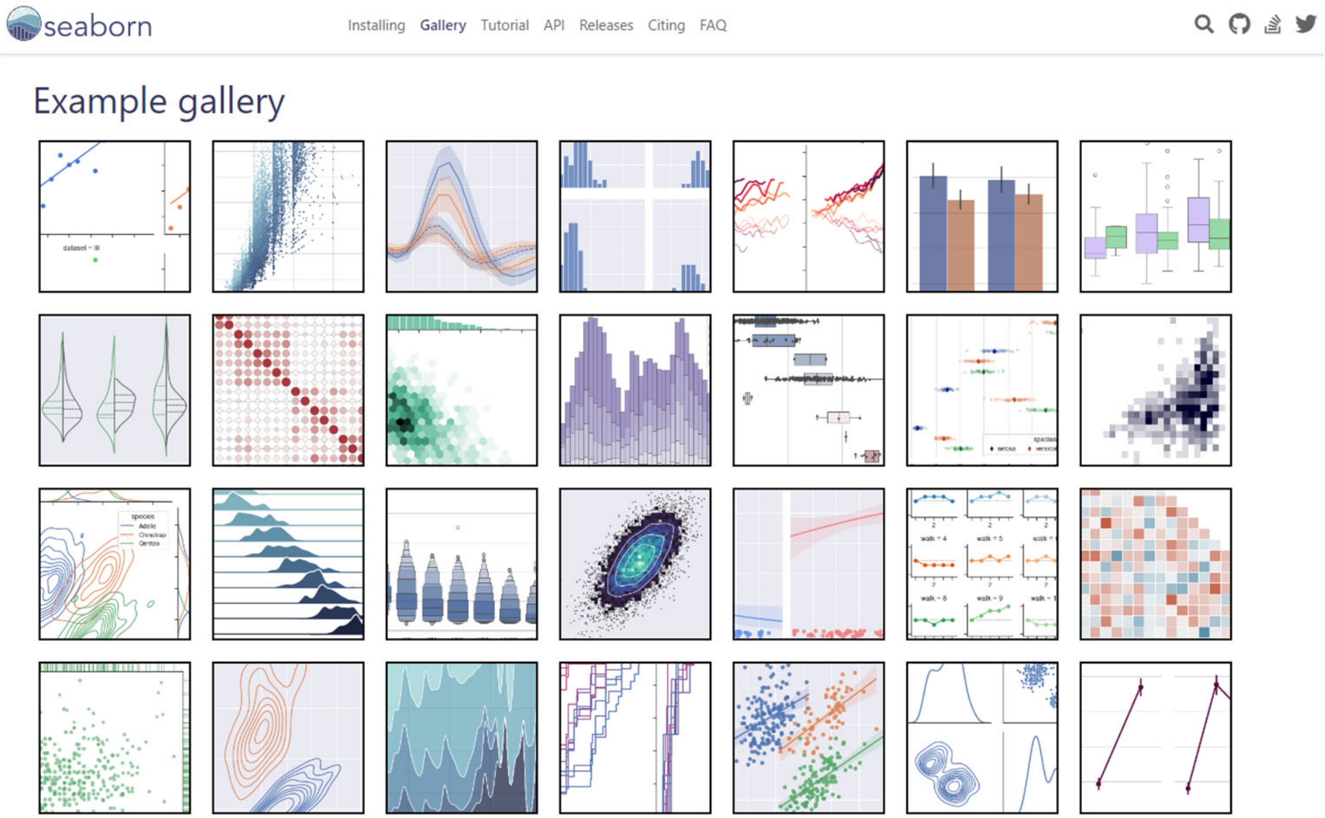
94602,00
Total Sales

Types of Data visualizations

- Line chart (Matplotlib)
- Scatter plot (Matplotlib)
- Histogram (Matplotlib)
- Bar chart/Stacked Barchart (Seaborn)
- Boxplot (Seaborn)
- Pie chart (No implementation needed in this course)

Types of Data visualizations

- Line chart
- Scatter plot
- Histogram
- Bar chart
- Boxplot
- Pie chart



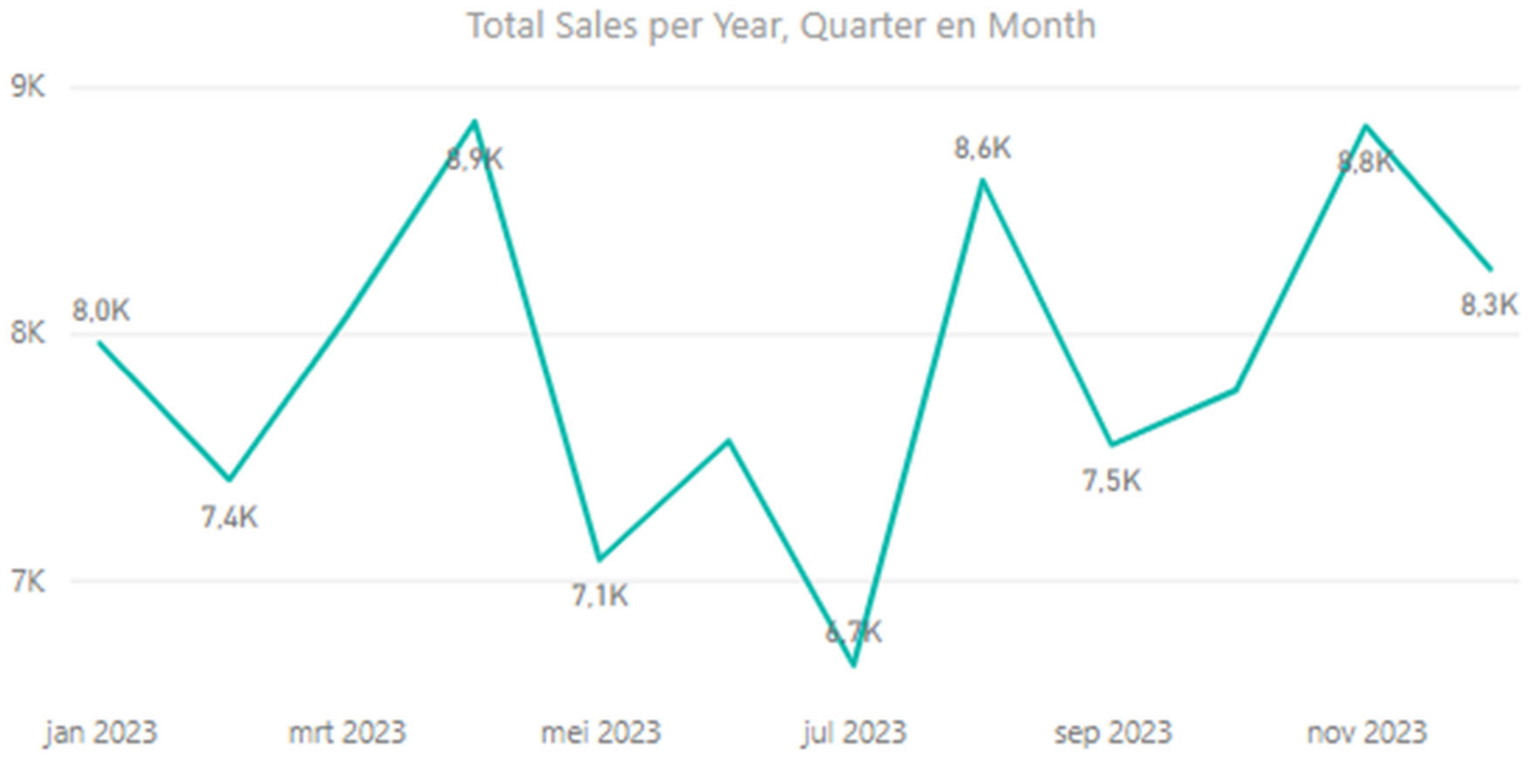
https://seaborn.pydata.org/examples/index.html

Graphs – Line Chart

- What is it?
  - A line that shows the relation between two variables, mostly between time and another variable
- When to use it?
  - Showing trends over time (timeseries)
  - Continuously monitoring data (for example: Temperature changes)
- Example:
  - Sales over the months of a year

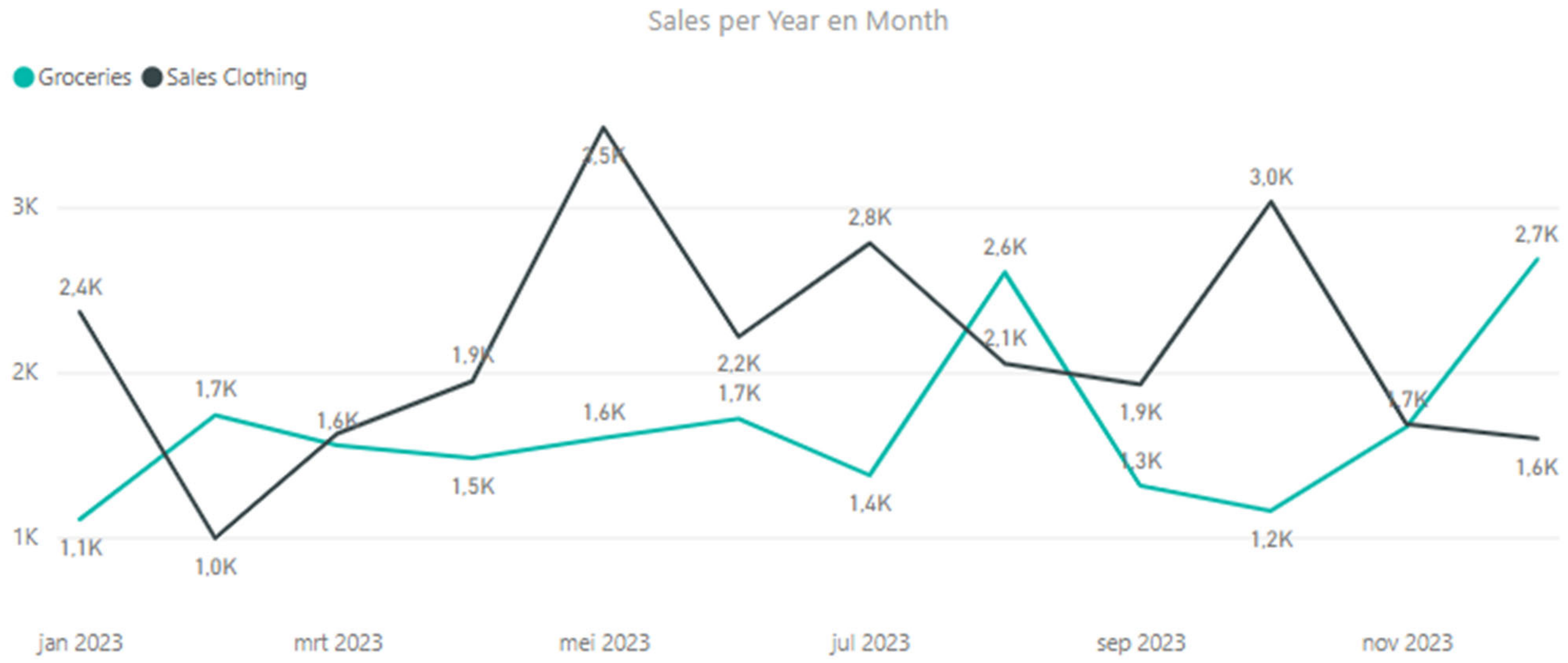
Graphs – Line Chart

Correct use:



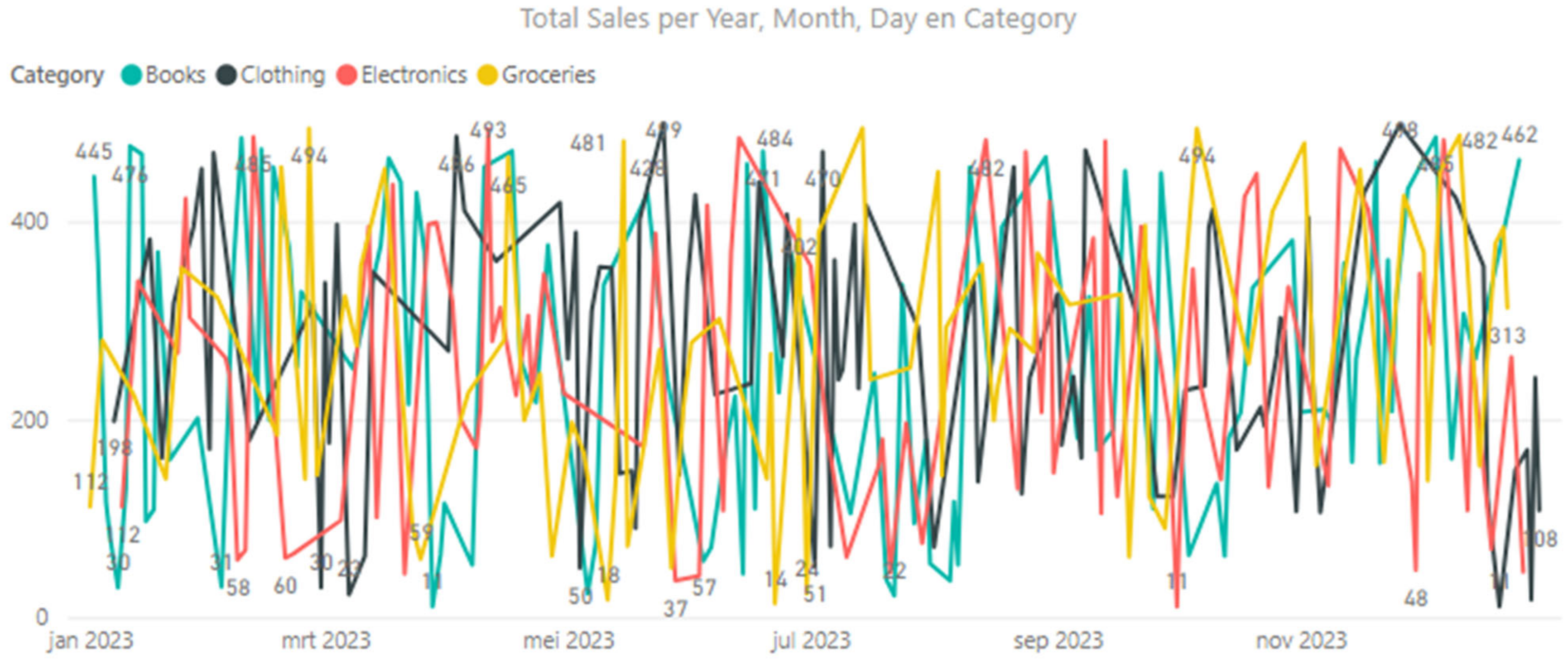
Graphs – Line Chart

Correct use:



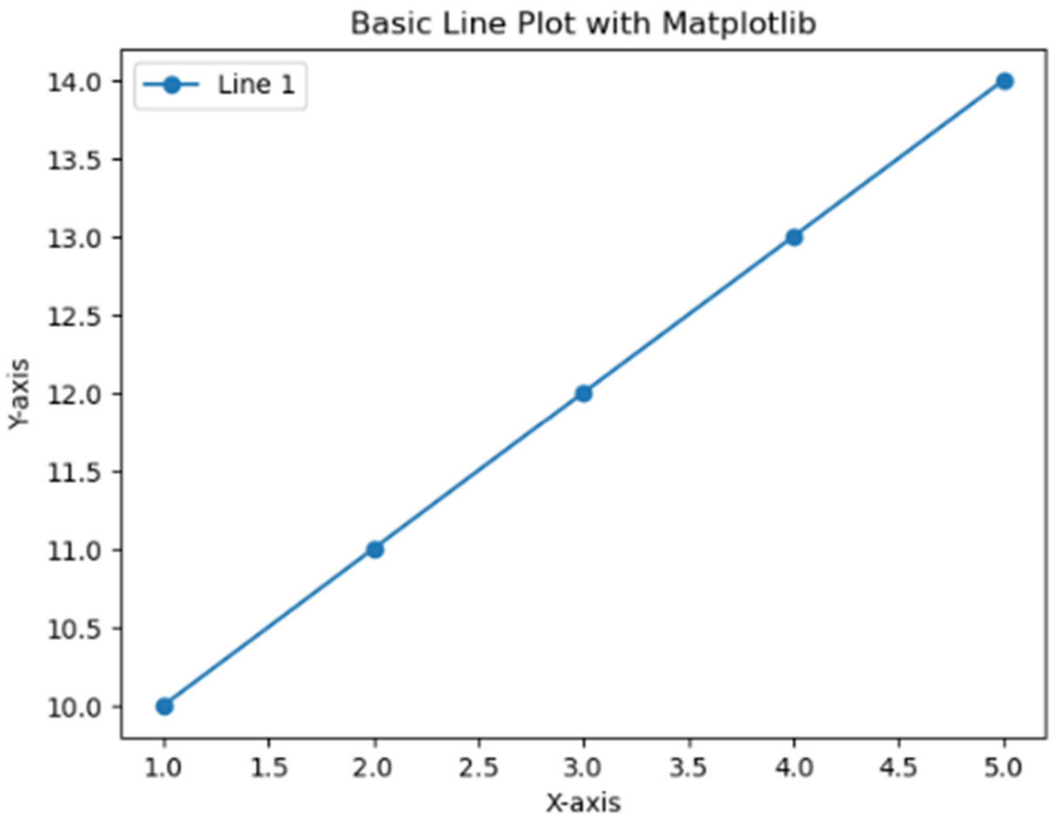
Graphs – Line Chart

Incorrect use:



Line Chart with Matplotlib

```
# Importing necessary libraries
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [10, 11, 12, 13, 14]

# Creating the line plot using Matplotlib
plt.plot(x, y, marker='o', label='Line 1')

# Adding title and labels
plt.title('Basic Line Plot with Matplotlib')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Adding legend
plt.legend()

# Showing the plot
plt.show()
```



Line Chart with Matplotlib

Notebook

See 04.01-Simple-Line-Plots.ipynb

Exercises

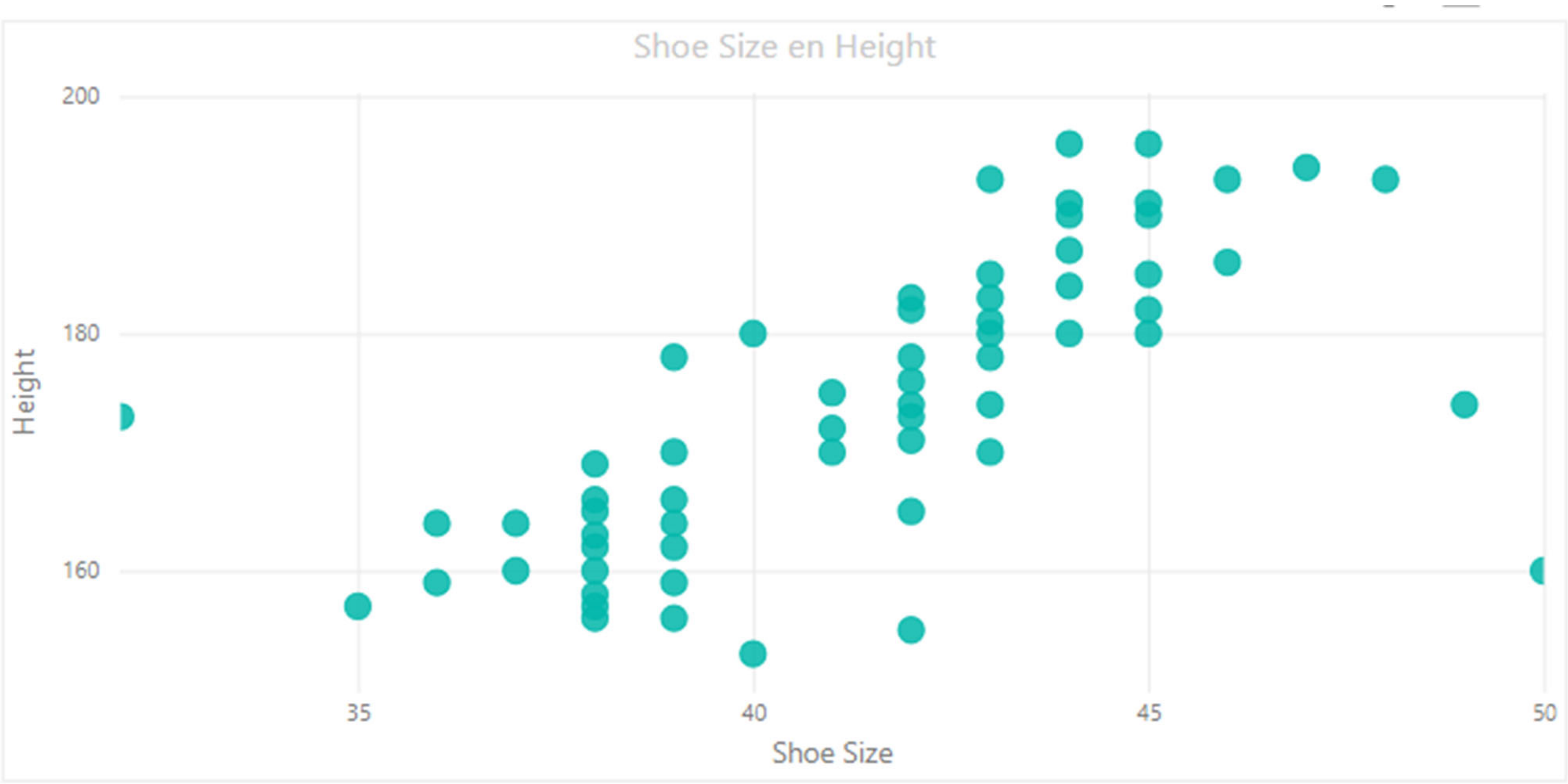
04.01_EX.ipynb

Seaborn Scatter plot:
https://seaborn.pydata.org/examples/different_scatter_variables.html

Graphs – Scatter Plot

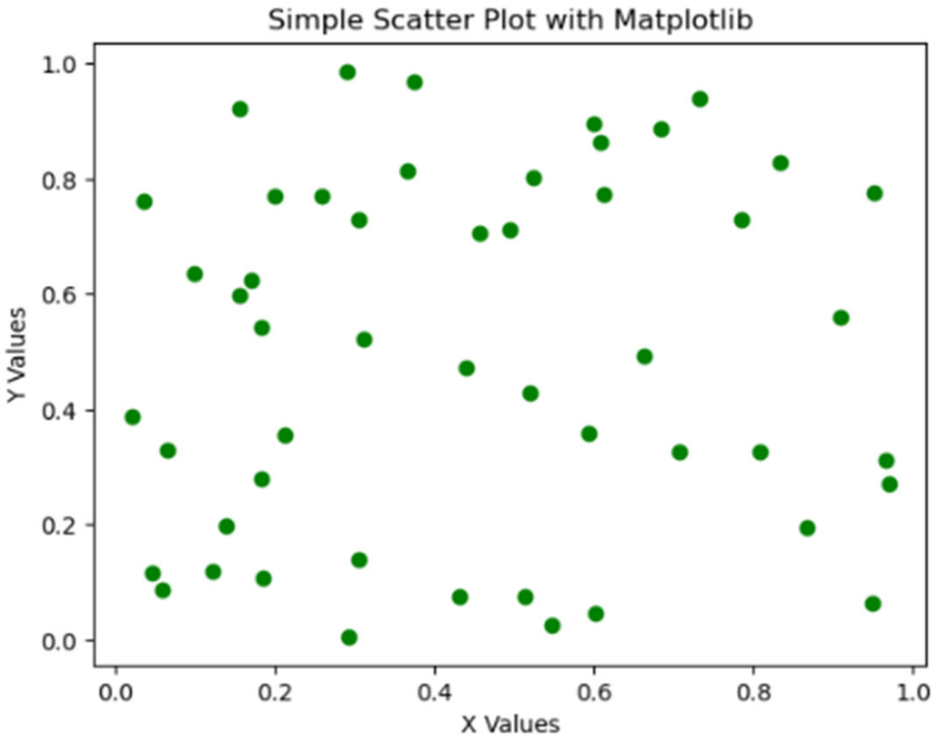
- What is it?:
  - Show relationship between two variables using dots.
- When to us it?
  - Relationship or correlation between two variables
  - Shows patterns like linear or non-linear relationships
- Example:
  - Relationship between length and weight.

Graphs – Scatter Plot



Scatter Plot with Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

# Generate random data
np.random.seed(42)  # For reproducibility
x = np.random.rand(50)
y = np.random.rand(50)

# Create the scatter plot
plt.scatter(x, y, color='green', marker='o')

# Add title and labels
plt.title('Simple Scatter Plot with Matplotlib')
plt.xlabel('X Values')
plt.ylabel('Y Values')

# Display the plot
plt.show()
```



Scatter Plot with Matplotlib

Notebook

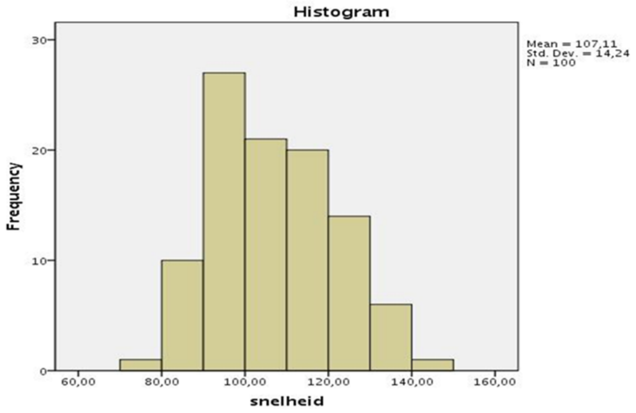
See 04.02-Simple-Scatter-Plots.ipynb

Exercises

04.02_EX.ipynb

Seaborn Scatter plot:
https://seaborn.pydata.org/examples/different_scatter_variables.html

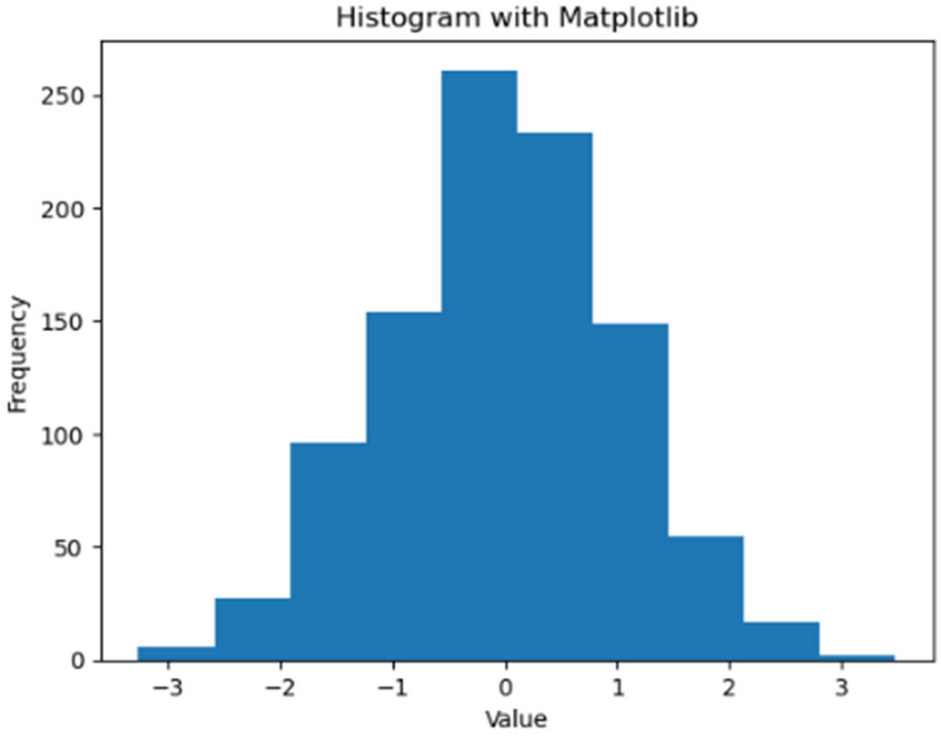
Graphs – Histogram



- What is it?
  - A graph that displays the frequencies of values in certain intervals
- When to use it?:
  - Visualization of the distribution of continuous variables
  - Suitable for interval- or ratio-data
- Visualization = Bar chart
  - But with the bars connected to each other!
- Example:
  - Distribution of ages in a certain population

Histogram with Matplotlib

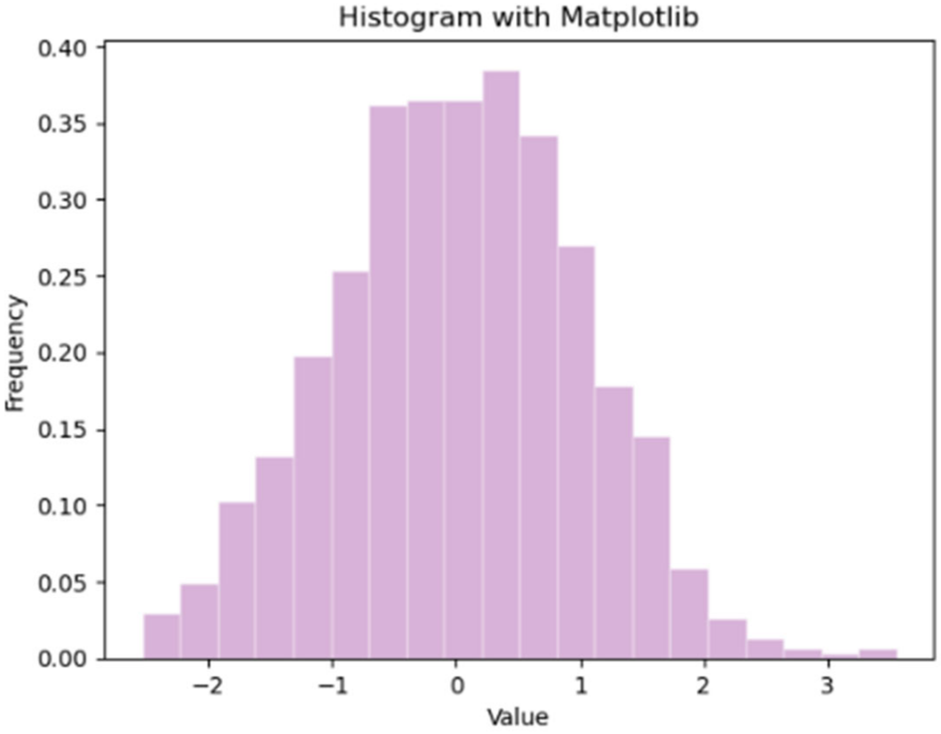
```
# Importing necessary libraries
import matplotlib.pyplot as plt
import numpy as np

# Generating random data
data = np.random.randn(1000)

# Creating the histogram using Matplotlib
plt.hist(data)

# Adding title and labels
plt.title('Histogram with Matplotlib')
plt.xlabel('Value')
plt.ylabel('Frequency')

# Showing the plot
plt.show()
```



Histogram with Matplotlib

```
# Creating the histogram using Matplotlib
plt.hist(x=data, bins=20, density=True, color='purple', alpha=0.3, edgecolor='white’)
```

- bins = 20 -> Set the number of bins to 20
- density = True -> Display Relative Frequencies
- color = ‘Purple’ -> Change color to purple
- alpha = 0.3 -> Set transparency to 30%
- edgecolor = ‘White’ -> Set borders to white



Histogram with Matplotlib

Notebook

See 04.05-Histograms-and-Binnings.ipynb

Exercises

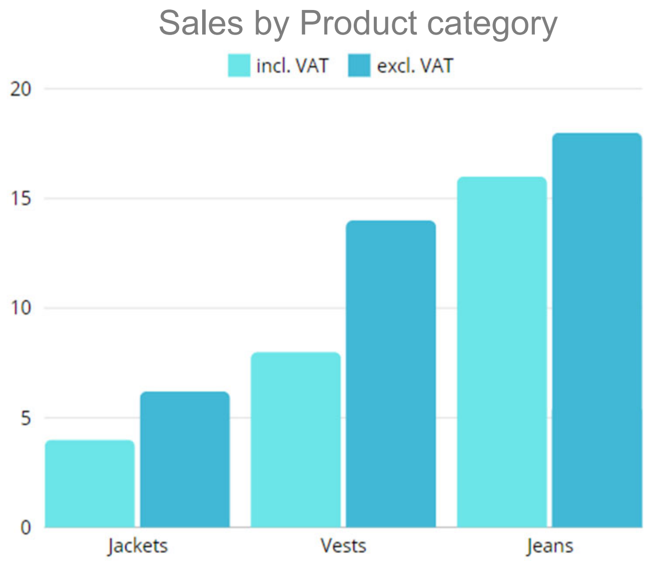
04.05_EX.ipynb

Seaborn Histogram:

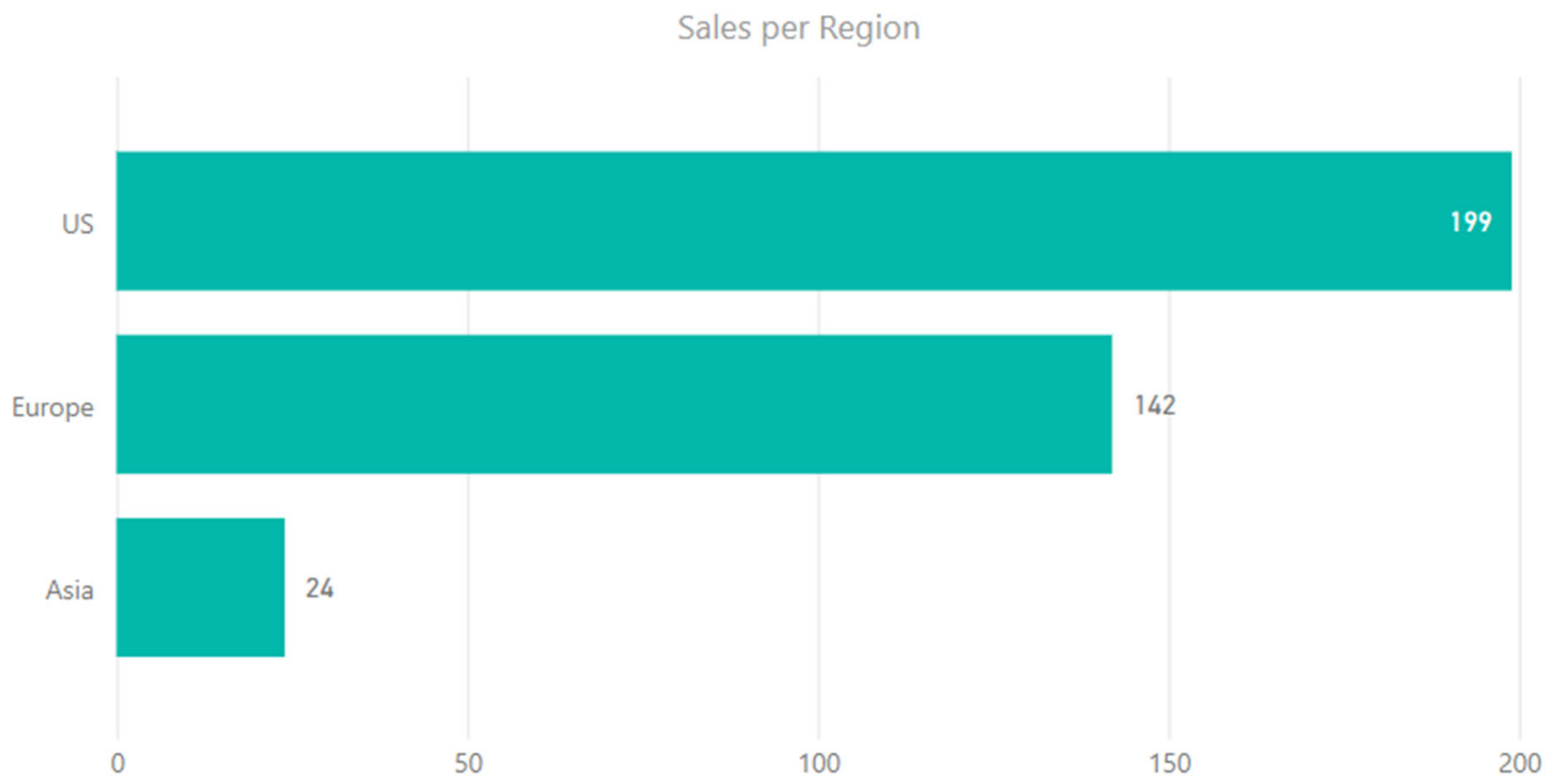
https://seaborn.pydata.org/examples/histogram_stacked.html
https://seaborn.pydata.org/examples/faceted_histogram.html

Graphs – Bar Chart

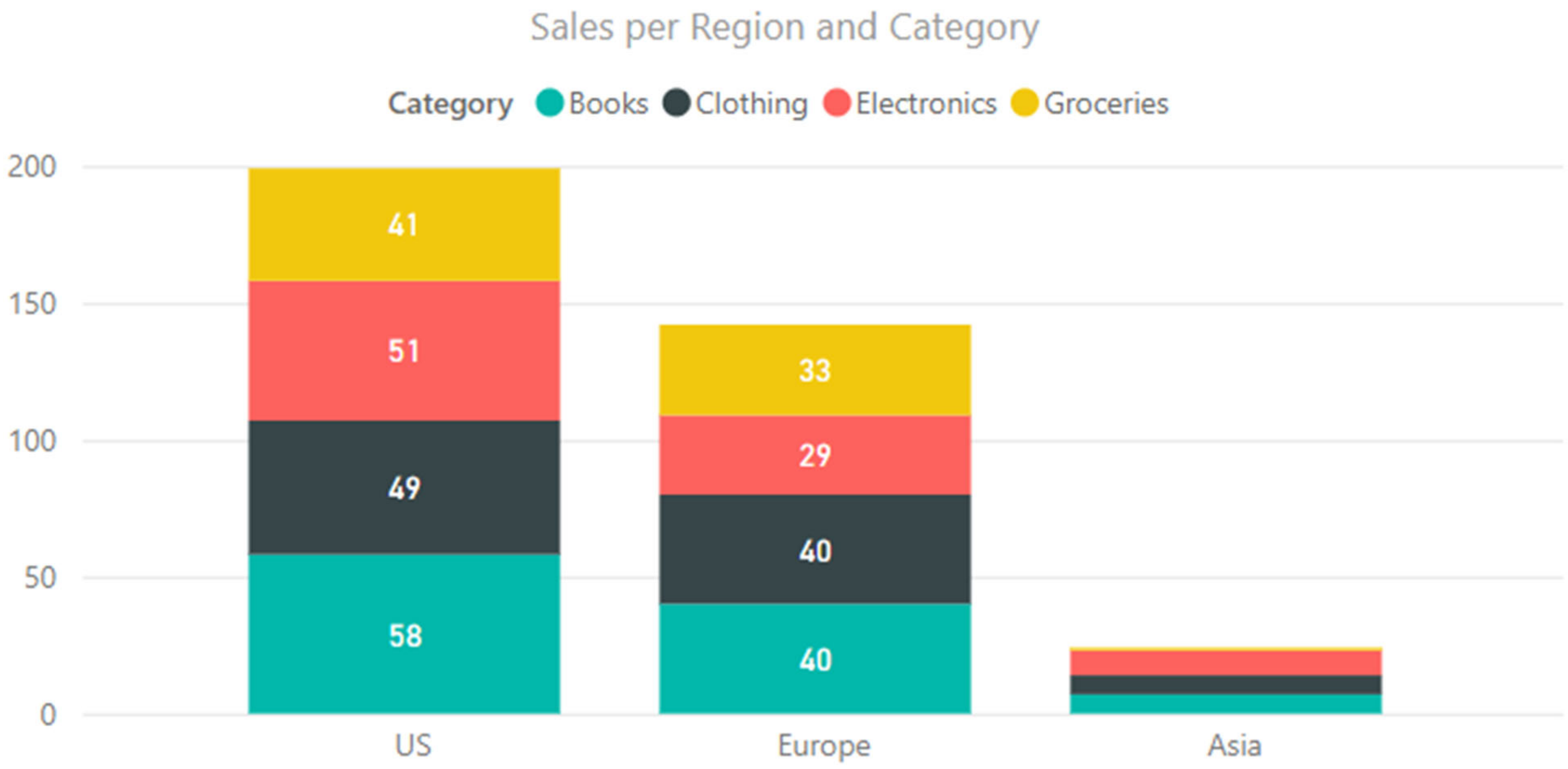
- What is it?
  - Shows data as bars that represent the height of categories
- When to use it?
  - Comparing discrete categories (for example sales per product)
  - Suited for nominal and ordinal data
- Example:
  - Number of sold products per product category.



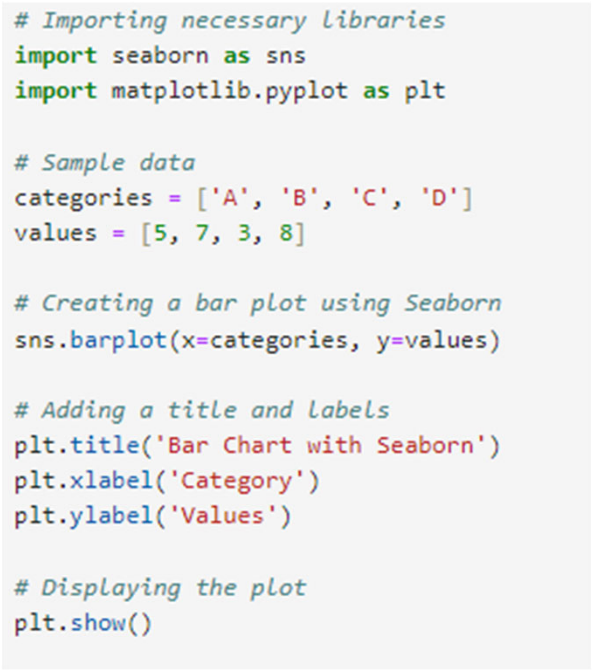
Graphs – Bar Chart



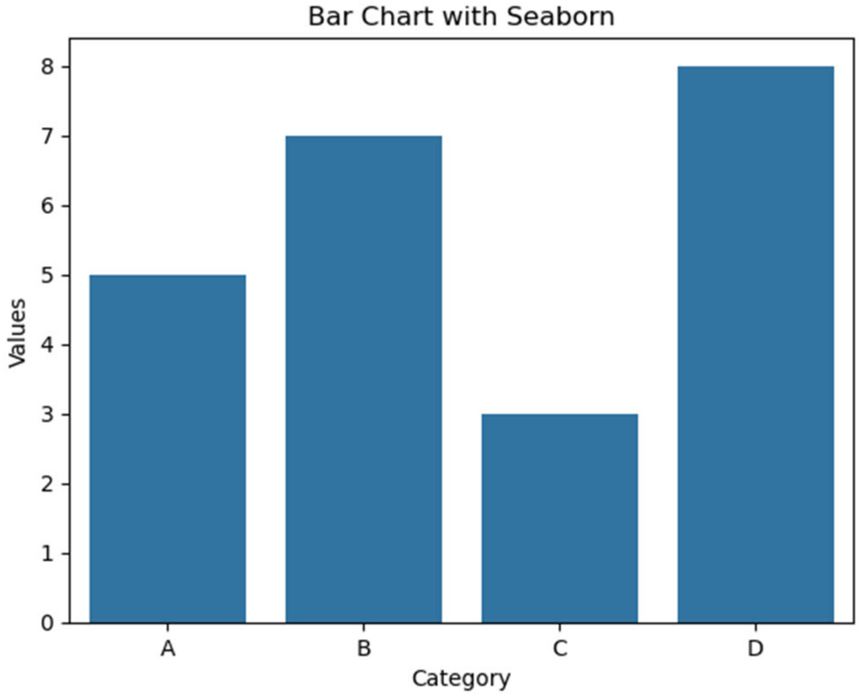
Graphs – Stacked Bar Chart



Bar Chart with Seaborn



```
# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Sample data
categories = ['A', 'B', 'C', 'D']
values = [5, 7, 3, 8]

# Creating a bar plot using Seaborn
sns.barplot(x=categories, y=values)

# Adding a title and labels
plt.title('Bar Chart with Seaborn')
plt.xlabel('Category')
plt.ylabel('Values')

# Displaying the plot
plt.show()
```



https://seaborn.pydata.org/examples/part_whole_bars.html

Bar Chart with Seaborn

Notebook

See 04.14-Visualization-with-Seaborn.ipynb

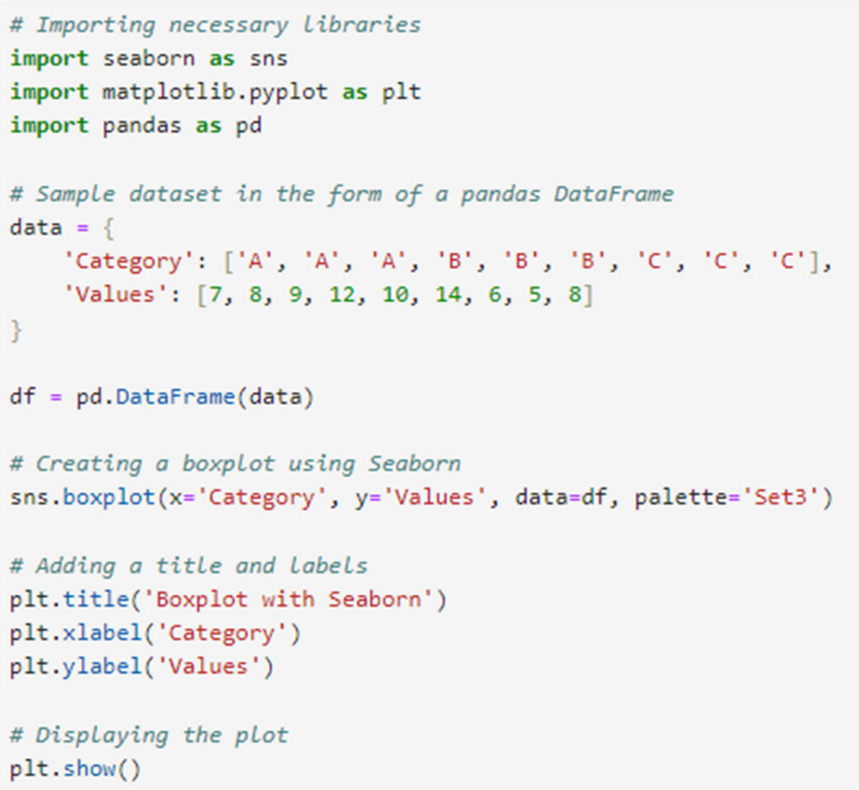
Exercises

04.14_EX.ipynb

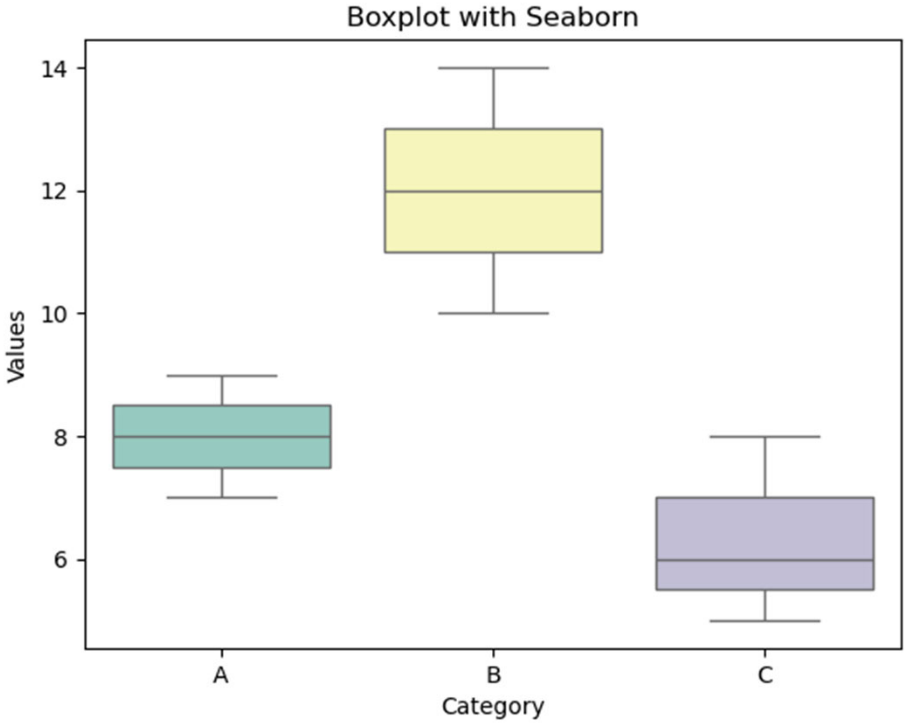
Graphs – Boxplot

- What is it?
  - Shows dispersion of a dataset using the min, Q1, median, Q3, max values.
- When to use it?
  - Visualisation of dispersion and outliers
  - To compare different datasets.
- Example:
  - Compare the dispersion of salaries between different sectors.

Boxplot with Seaborn



```python
# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Sample dataset in the form of a pandas DataFrame
data = {
    'Category': ['A', 'A', 'A', 'B', 'B', 'B', 'C', 'C', 'C'],
    'Values': [7, 8, 9, 12, 10, 14, 6, 5, 8]
}

df = pd.DataFrame(data)

# Creating a boxplot using Seaborn
sns.boxplot(x='Category', y='Values', data=df, palette='Set3')

# Adding a title and labels
plt.title('Boxplot with Seaborn')
plt.xlabel('Category')
plt.ylabel('Values')

# Displaying the plot
plt.show()
```



https://seaborn.pydata.org/examples/grouped_boxplot.html

Boxplot with Seaborn

Notebook

See 04.14-Visualization-with-Seaborn.ipynb

Exercises

04.14_EX.ipynb

Graphs - Pie Chart

- What is it?
  - Shows parts of a whole in circlesegments.
- When to use it?
  - Showing distributions as part of a whole (100%)
  - Suited for relatively few categories (ideally maximum 5)

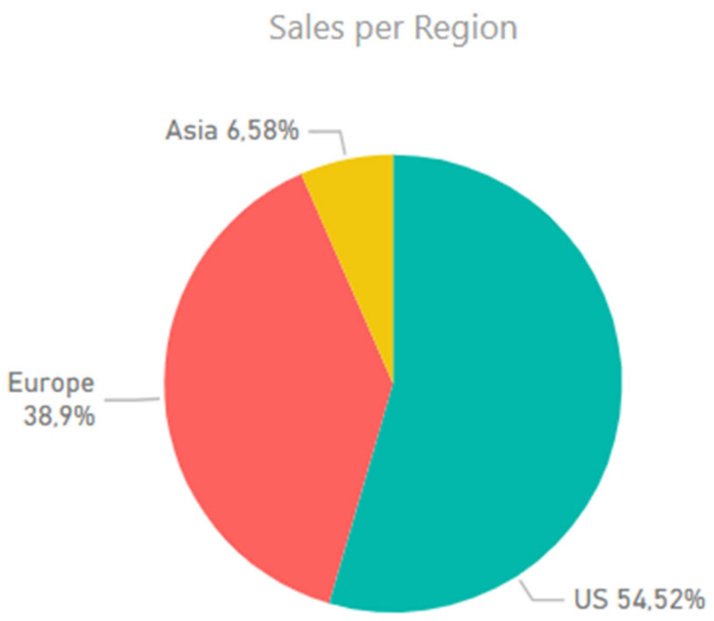
Example:

  - Marketshare of 3 different companies.

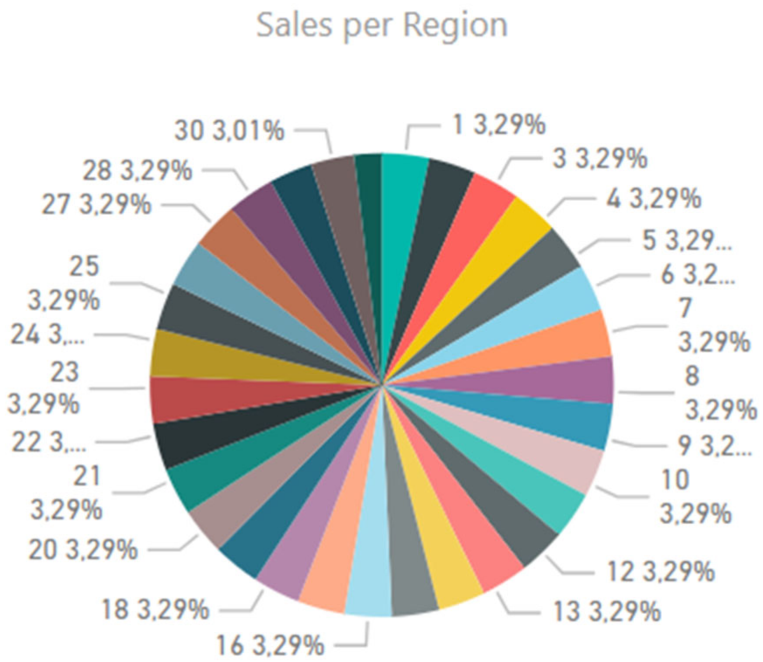
- **Remark**: Not suited for many categories or if the exact differences between categories are important!

Graphs - Pie Chart

Correct use:

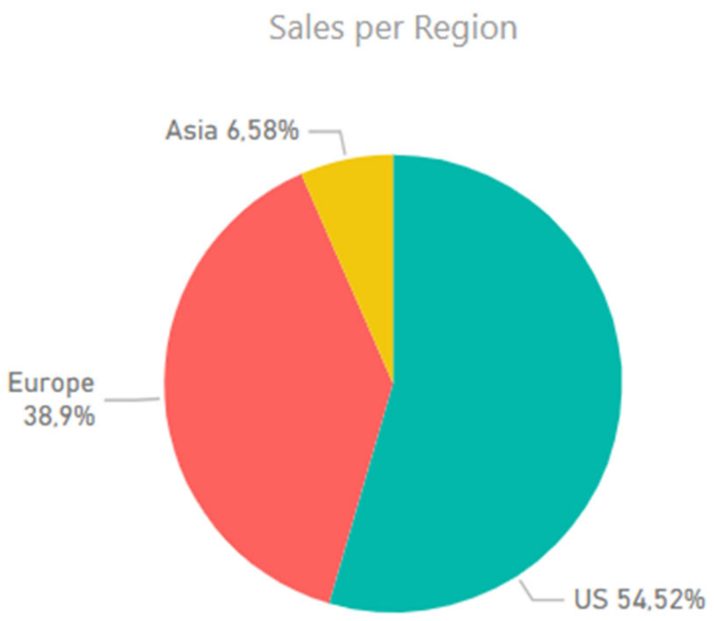


Incorrect use:

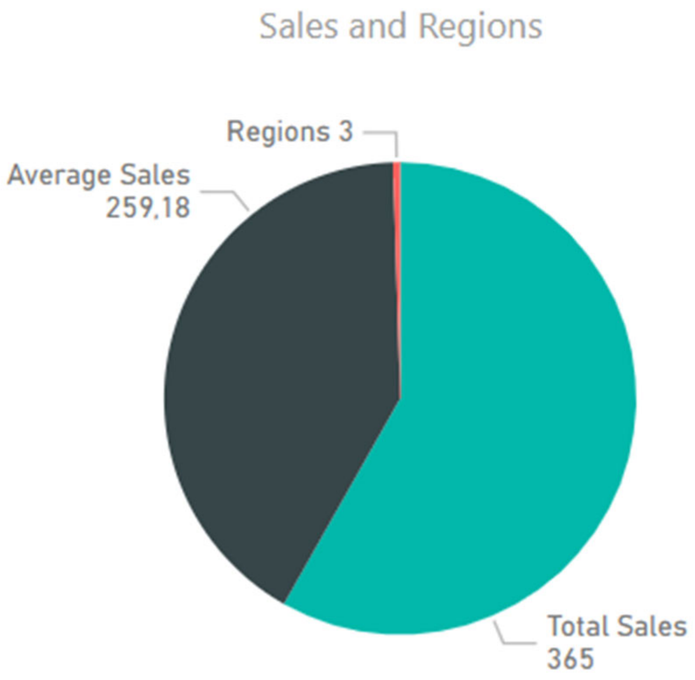


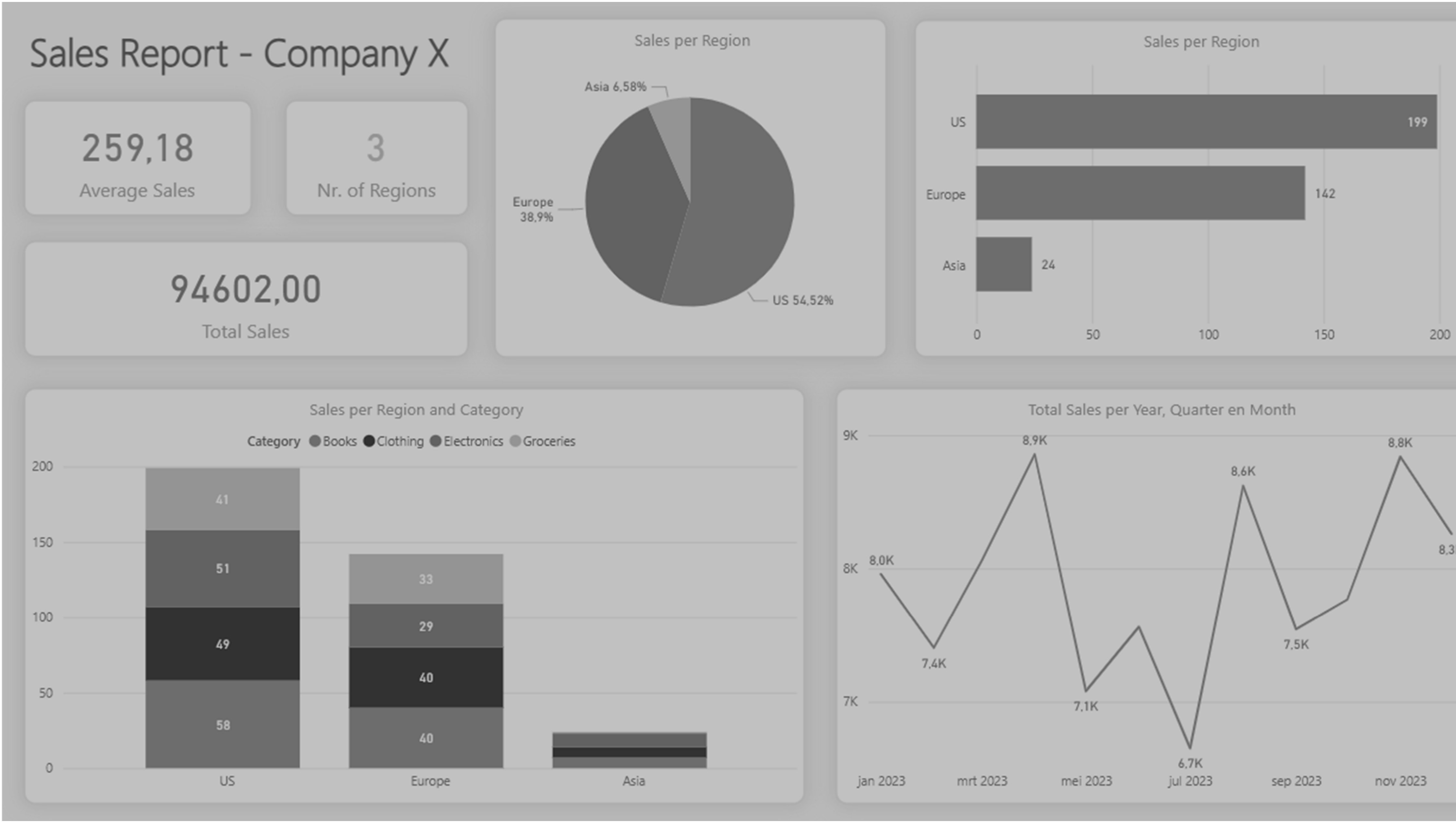
Graphs - Pie Chart

Correct use:



Incorrect use:





Sales Report - Company X

259,18
Average Sales

3
Nr. of Regions

94602,00
Total Sales

Graphs – frequent mistakes

- Too many categories in a pie or line chart.
- Wrongly chosen axes (manipulating the Y-axis)
- Wrong graph for the type of data

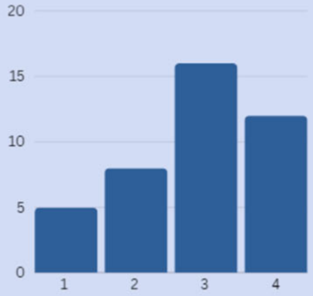
Tips to make graphs better and more effective:

- Only show the necessary information
- Don’t put too much information in 1 graph, it is better to spread the information over multiple graphs instead.

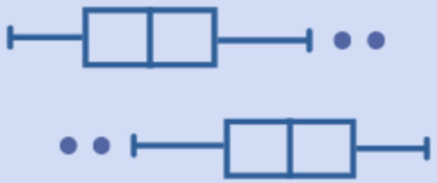
What do you want to show?

How the data is distributed

Frequency distribution: Histogram

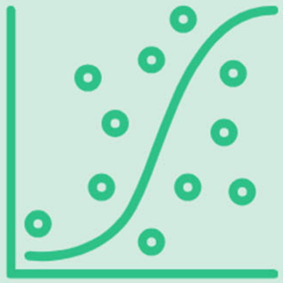


Statistical dispersion: Boxplot



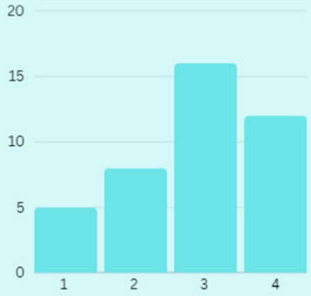
How values relate to each other

Scatterplot

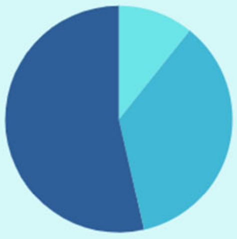


How values compare to each other

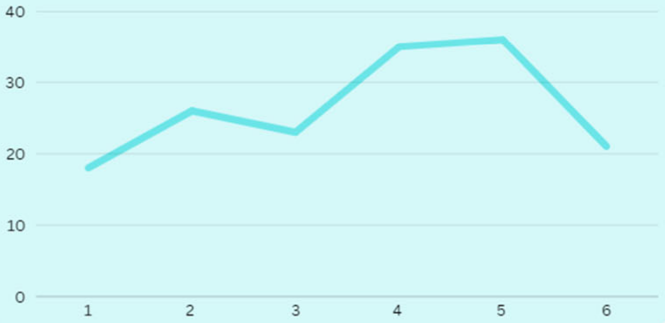
Bar chart



Pie chart

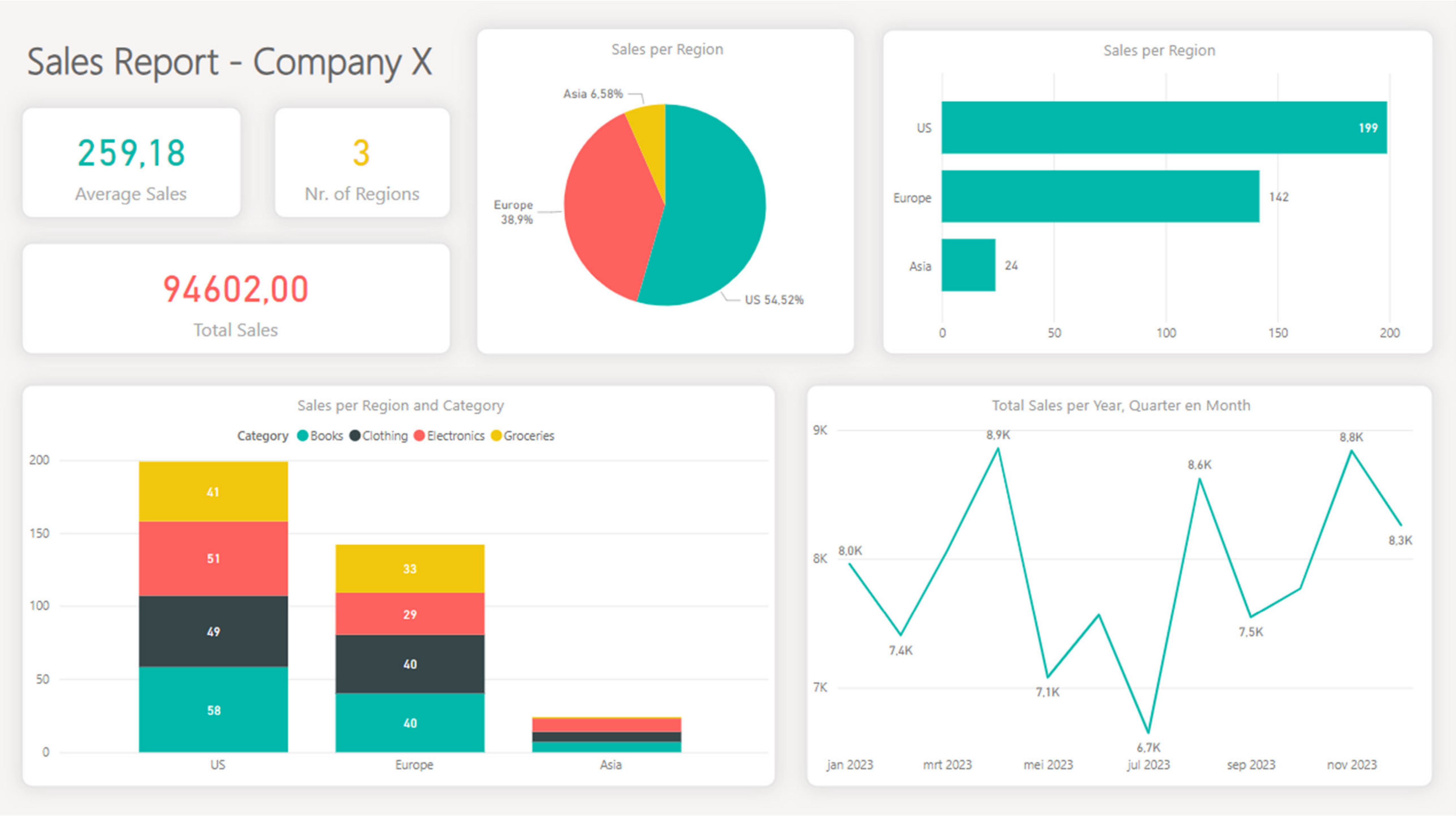


Over time: Line chart



Graphs – When to use which one?

- Bar chart vs. Pie chart:
  - Use the bar chart when it is important to see exact differences between categories.
  - Use Piecharts only for up to 5 different categories
- Linechart vs. histogram:
  - use histogram for distributions, linecharts for trends over time (time series).
- Boxplot vs. scatter plot:
  - Use a boxplot for dispersion and a scatterplot for relationships between to variables.



Sales Report - Company X

259,18
Average Sales

3
Nr. of Regions

94602,00
Total Sales

Graphs – Conclusion

- Know when to use which graph
- Graphs should make it easier to understand data
- Avoid making graphs with high complexity