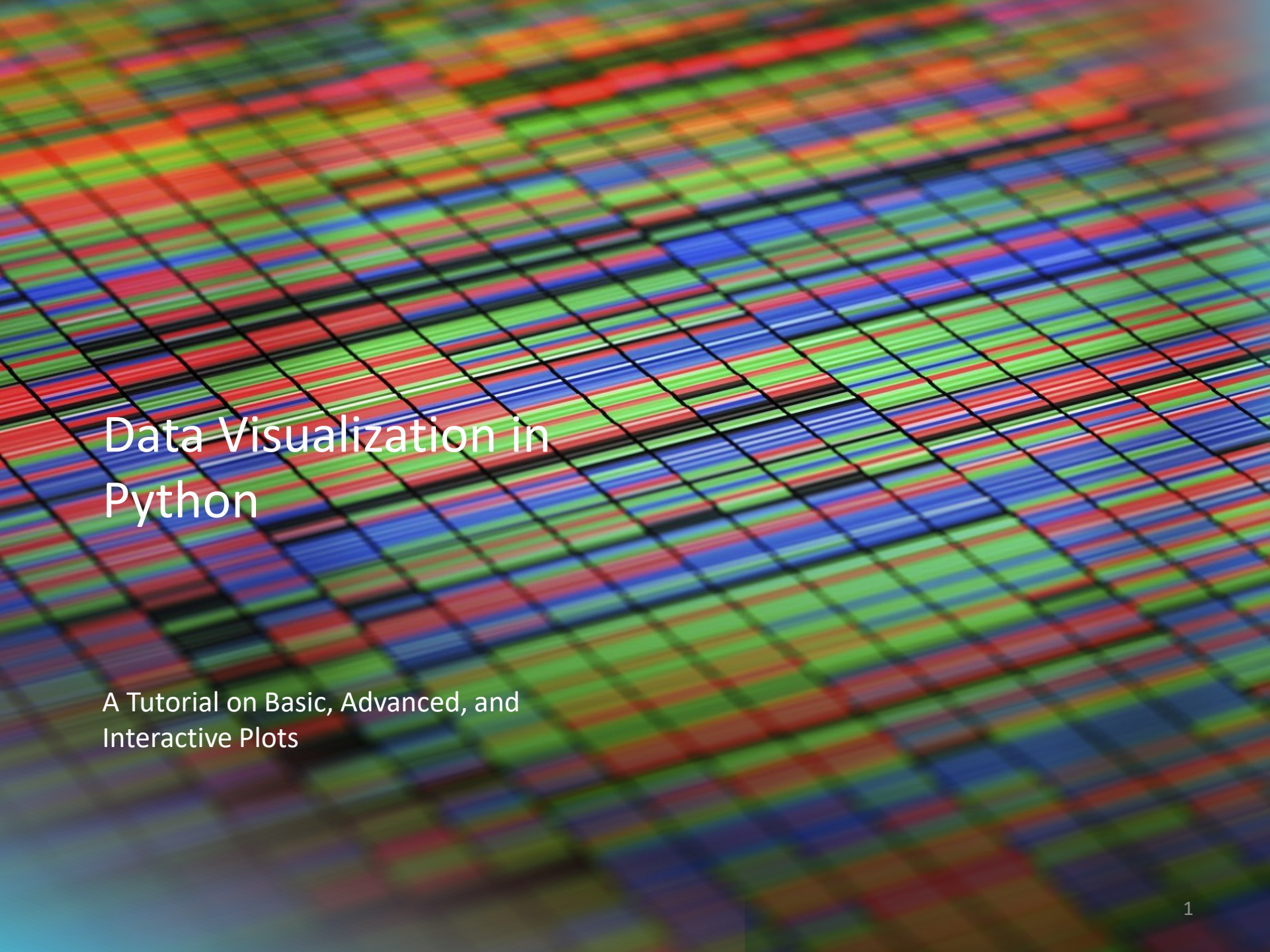
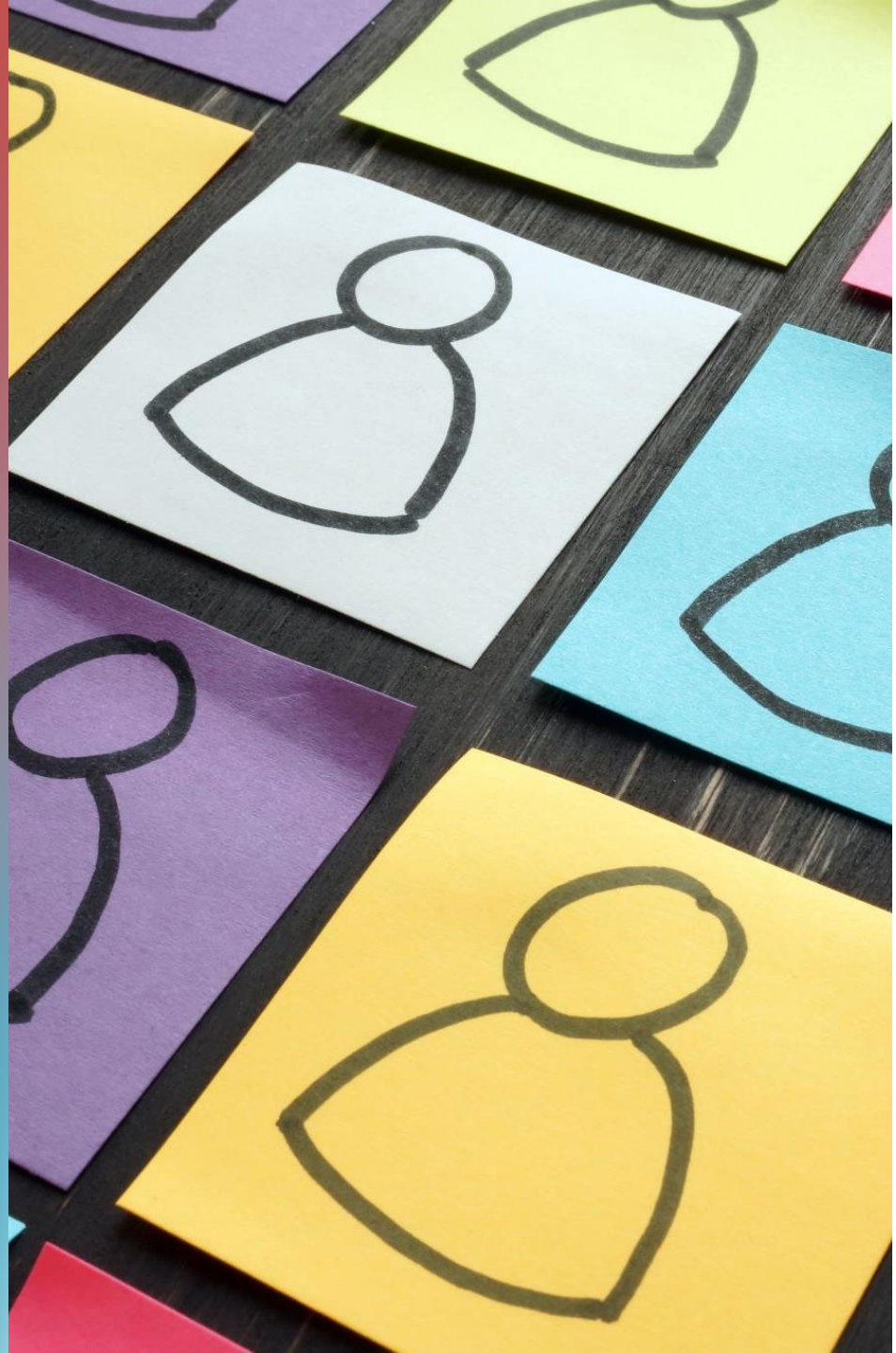

Data Visualization in Python
A Tutorial on Basic, Advanced, and Interactive Plots

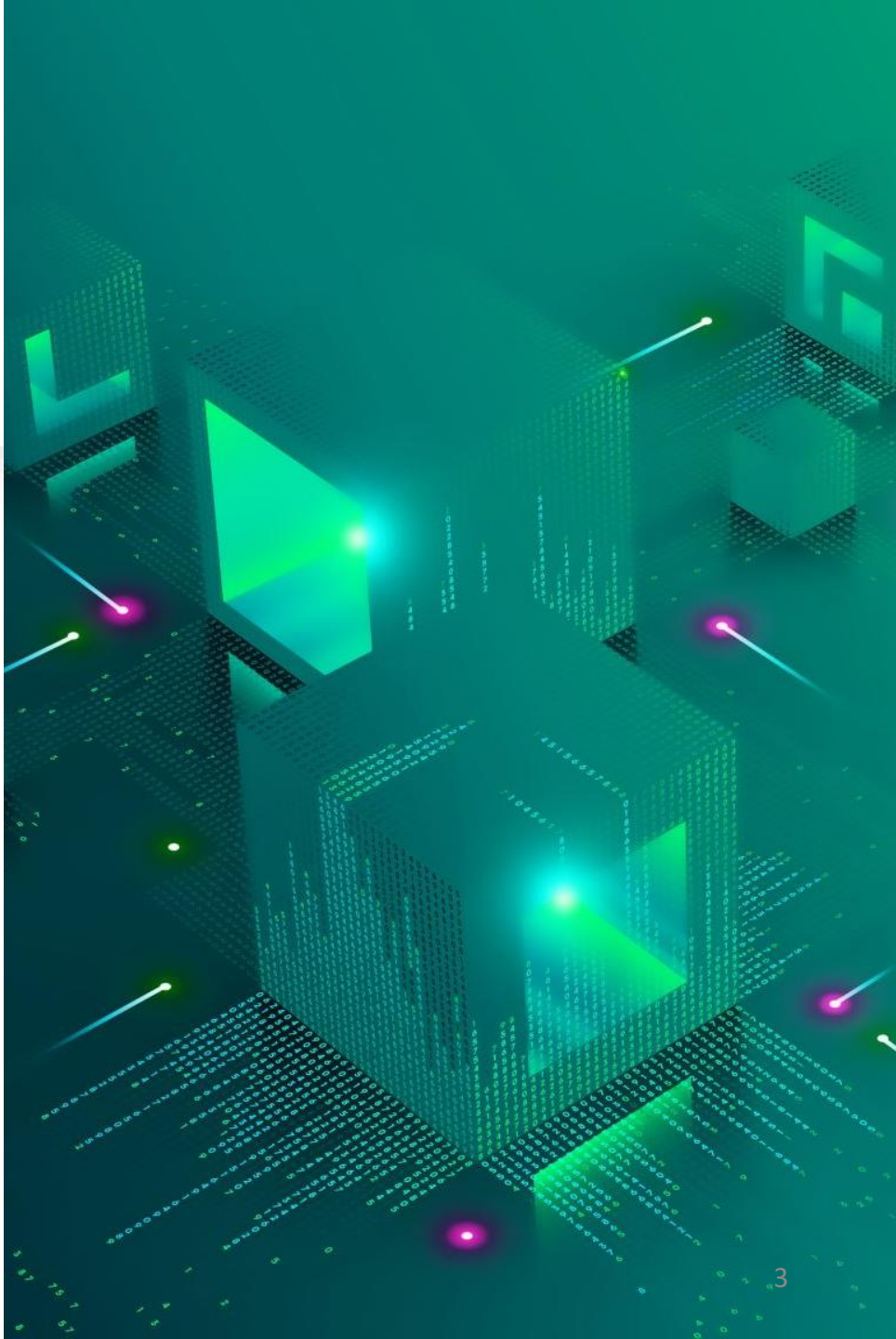


Session Objectives

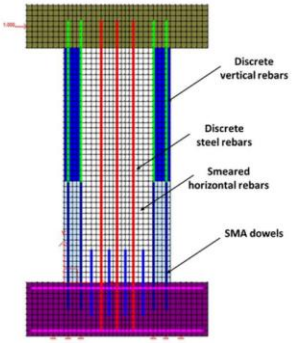
- Understand the key **principles and best practices** of data visualization.
- Learn to use popular **Python libraries** (Matplotlib, Seaborn, Plotly) for creating various types of visualizations.
- Develop the ability to **choose** the right type of chart for different data scenarios.
- Gain **practical experience** through hands-on exercises and examples.

Contents

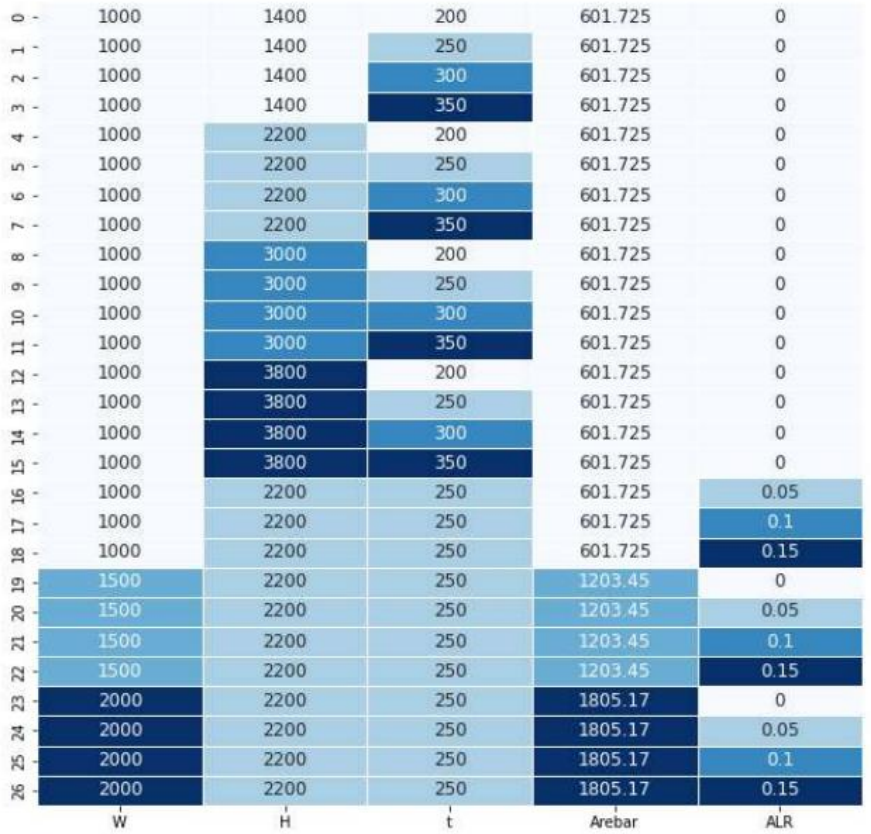
1. Introduction to Data Visualization in Python
2. Setting Up Your Environment
3. Basic Visualization with Matplotlib
4. Advanced Visualization with Seaborn
5. Interactive Visualization with Plotly
6. Tips for Effective Data Visualization



Input data

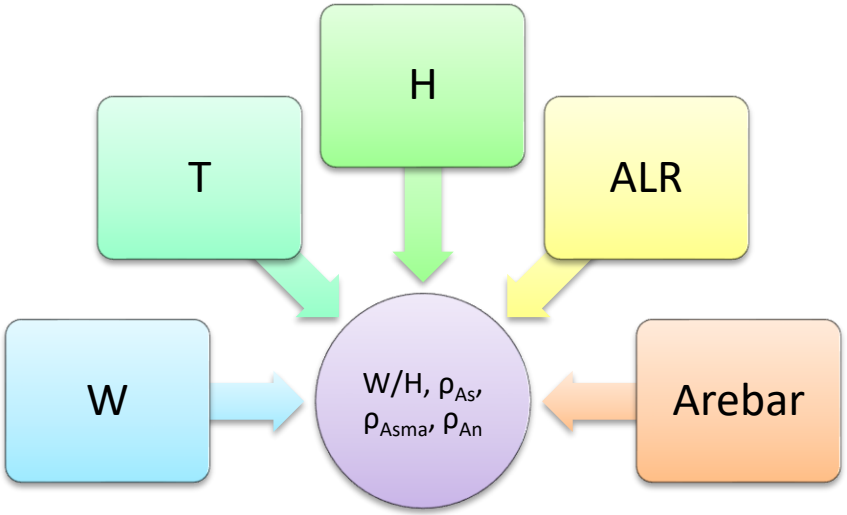


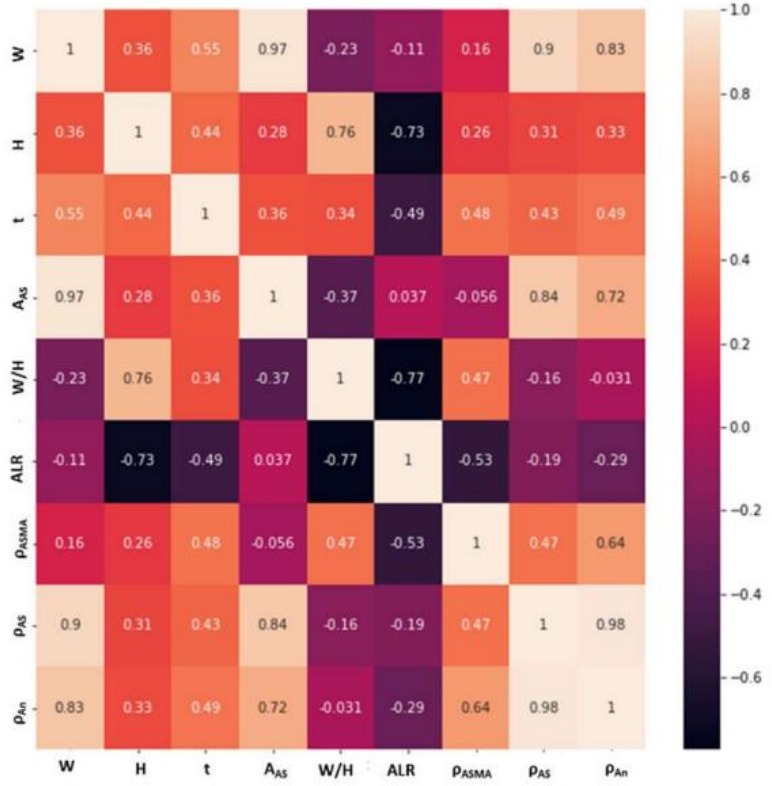
W	H	t	Arebar	ALR
1000	3000	250	601.725	0
1000	2200	300	601.725	0.1
1000	3800	200	601.725	0
1500	2200	200	1203.45	0
1000	1400	200	601.725	0
1500	2200	300	1203.45	0.1
1000	2200	350	601.725	0
1000	3800	250	601.725	0
1000	2200	200	601.725	0
2000	2200	300	1805.17	0.1
2000	2200	250	1805.17	0.05
1000	2200	250	601.725	0.05
1000	2200	200	601.725	0
1000	1400	250	601.725	0
1500	2200	350	1203.45	0.15
2000	2200	350	1805.17	0.15
1000	2200	300	601.725	0
1000	3000	350	601.725	0
1000	3000	200	601.725	0
1000	3800	300	601.725	0
2000	2200	200	1805.17	0
1000	2200	350	601.725	0.15
1000	1400	300	601.725	0
1000	3000	300	601.725	0
1500	2200	250	1203.45	0.05
1000	1400	350	601.725	0
1000	3800	350	601.725	0
1000	2200	250	601.725	0



Main and dependent variables

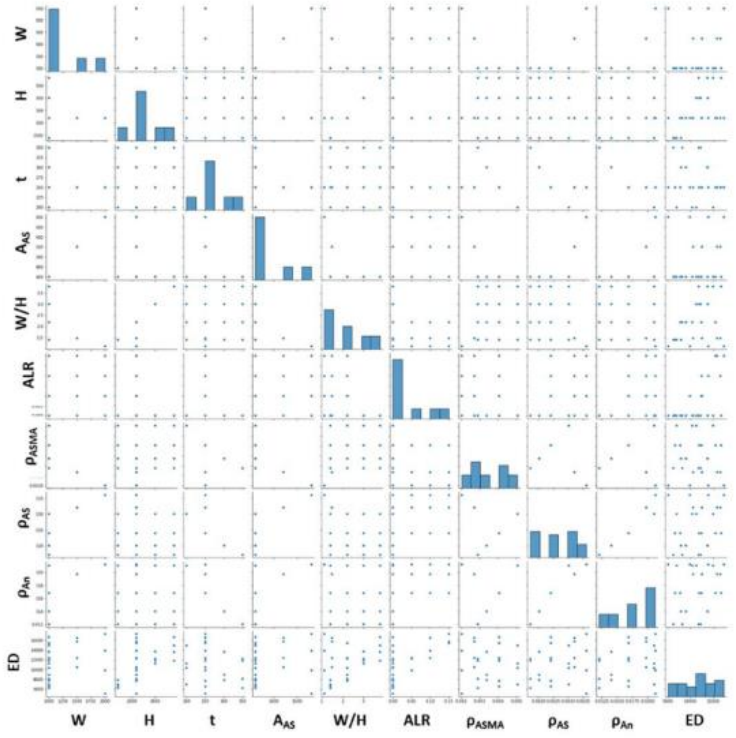
$$\rho_{An} = A_s + nA_{ASMA}/A_c$$



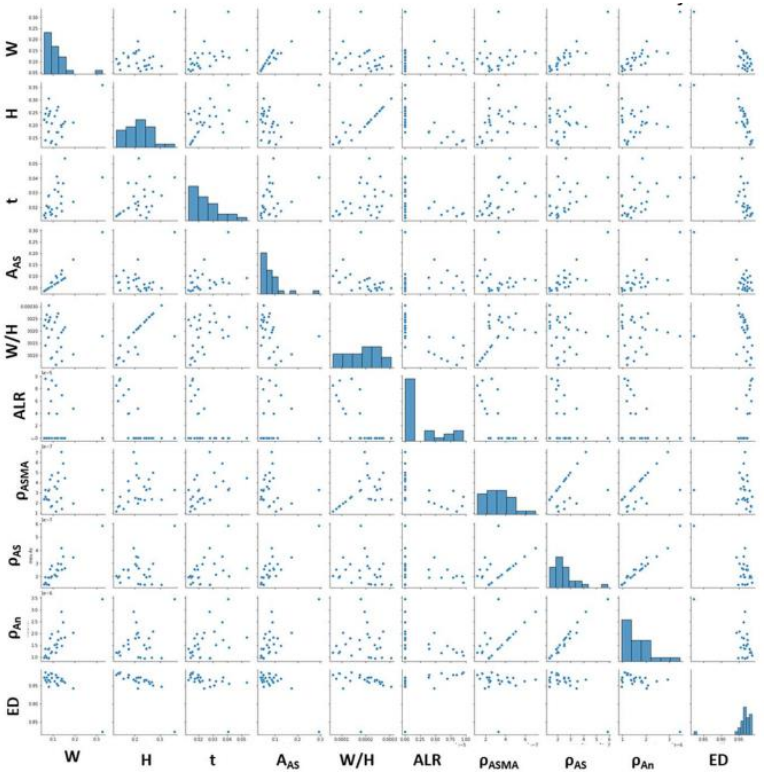


Correlation matrix for the main and dependent variables

Dataset VS output

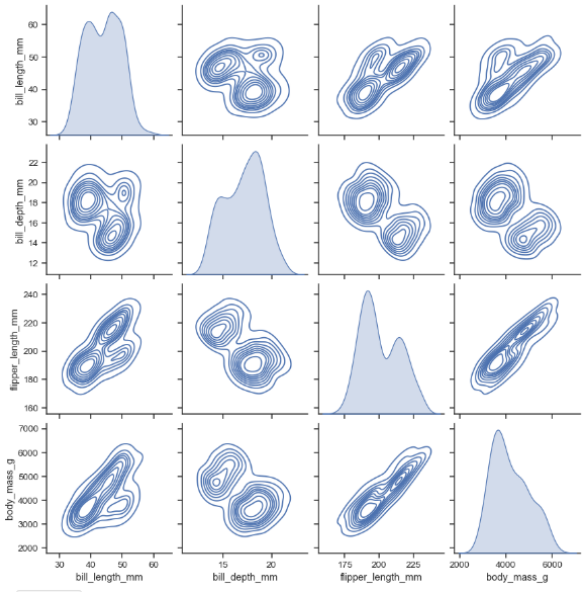


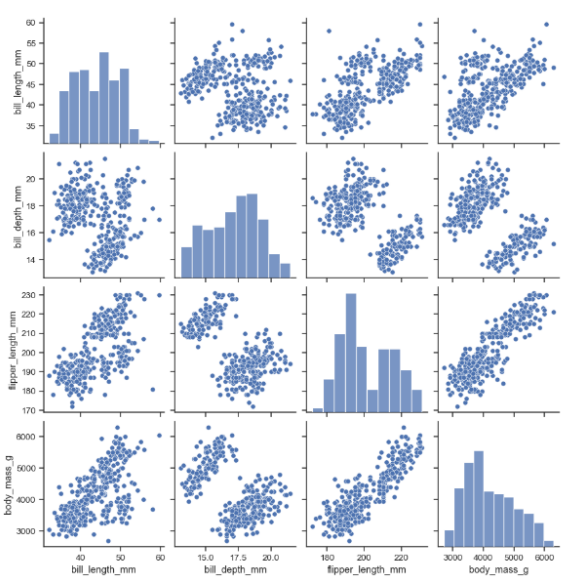
Scatter pair plot for the raw data

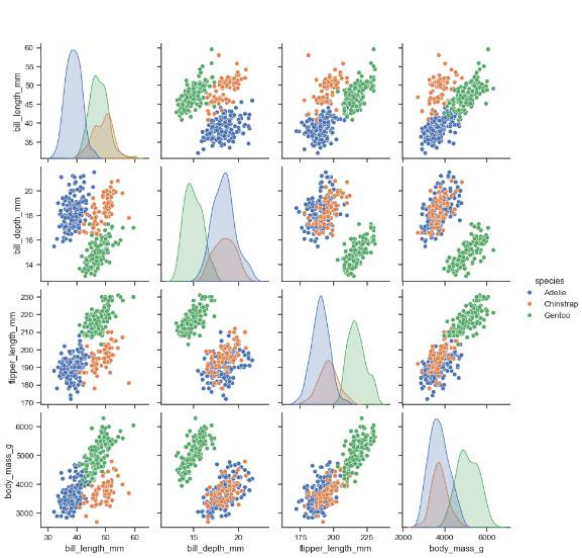


Scatter pair plot for the normalized data

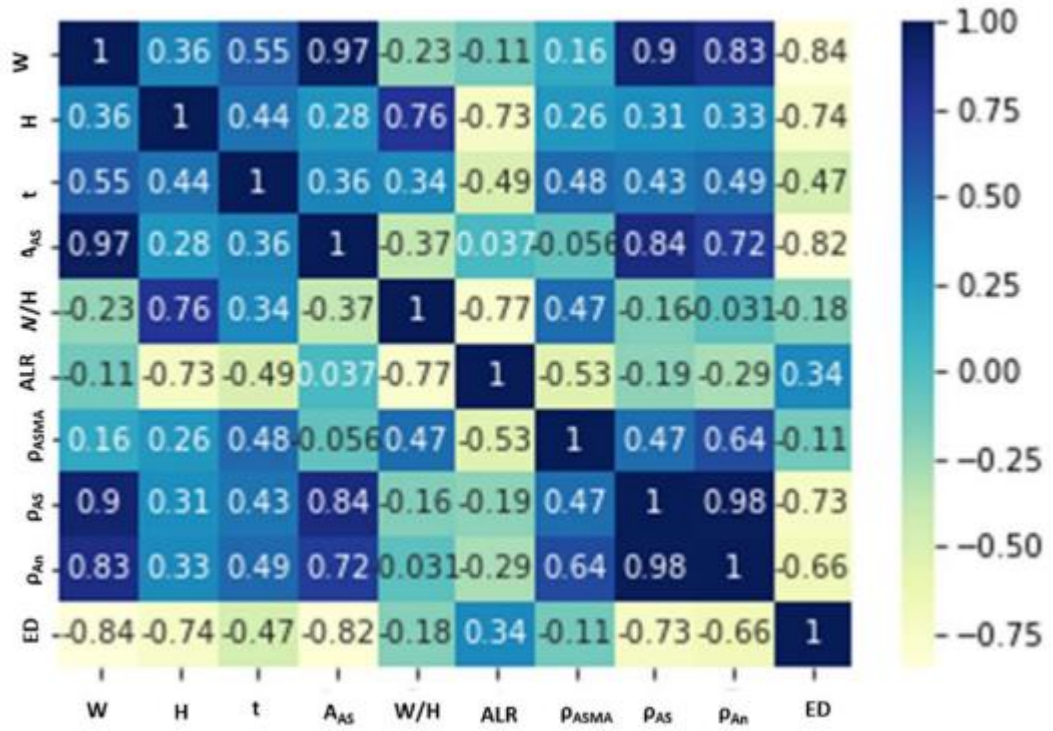
Examples on scatter pair plots







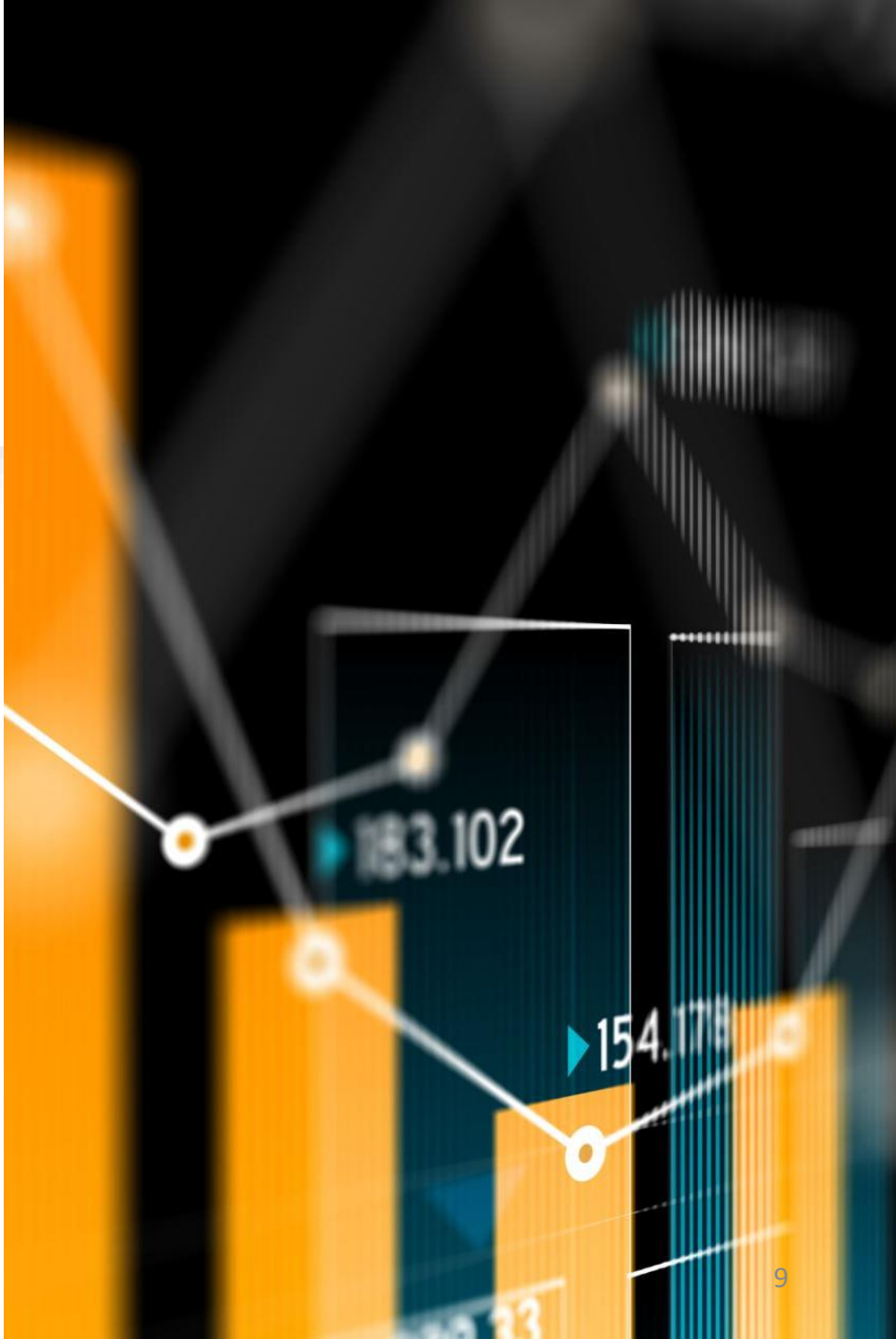
Dataset VS output



Correlation matrix between the input and output

Introduction to Data Visualization in Python

- Data visualization is essential for understanding and interpreting data. Python offers several powerful libraries for creating a wide range of static, animated, and interactive plots. The most common libraries are:
  - **Matplotlib:** A versatile library for basic plots like line charts, bar charts, and scatter plots.
  - **Seaborn:** Built on top of Matplotlib, provides a high-level interface for drawing attractive statistical graphics.
  - **Plotly:** Allows for creating interactive plots easily and is great for dashboards.



Setting Up Your Environment

- Before you start, ensure you have Python and the required libraries installed. You can install these libraries using pip:

```
pip install matplotlib seaborn plotly pandas numpy
```

Basic Visualization with Matplotlib

Line plot:

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

# Generate sample data
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Create a line plot
plt.figure(figsize=(8, 5))
plt.plot(x, y, label='Sine Wave')
plt.title('Line Plot Example')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.grid(True)
plt.show()
```

Basic Visualization with Matplotlib

Scatter plot:

```
import matplotlib.pyplot as plt

# Sample data
x = np.random.rand(50)
y = np.random.rand(50)
colors = np.random.rand(50)
sizes = 1000 * np.random.rand(50)

# Create a scatter plot
plt.figure(figsize=(8, 5))
plt.scatter(x, y, c=colors, s=sizes, alpha=0.5,
cmap='viridis')
plt.title('Scatter Plot Example')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.colorbar()  # Show color scale
plt.show()
```

Basic Visualization with Matplotlib

Bar Chart:

```
import matplotlib.pyplot as plt

# Sample data
categories = ['A', 'B', 'C', 'D', 'E']
values = [5, 7, 3, 8, 4]

# Create a bar chart
plt.figure(figsize=(8, 5))
plt.bar(categories, values, color='skyblue')
plt.title('Bar Chart Example')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.show()
```

Basic Visualization with Matplotlib

Pair plot:

```
import seaborn as sns
import pandas as pd

# Load sample data
df = sns.load_dataset('iris')

# Create a pair plot
sns.pairplot(df, hue='species')
plt.show()
```

Basic Visualization with Matplotlib

Heat Map:

```
import seaborn as sns
import numpy as np

# Sample data
data = np.random.rand(10, 12)

# Create a heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(data, annot=True, cmap='coolwarm')
plt.title('Heatmap Example')
plt.show()
```

Basic Visualization with Matplotlib

Correlation matrix heat map:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Load sample data
df = sns.load_dataset('iris')

# Calculate the correlation matrix
corr_matrix = df.corr()

# Create a correlation matrix heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='viridis',
fmt='.2f')
plt.title('Correlation Matrix Heatmap')
plt.show()
```

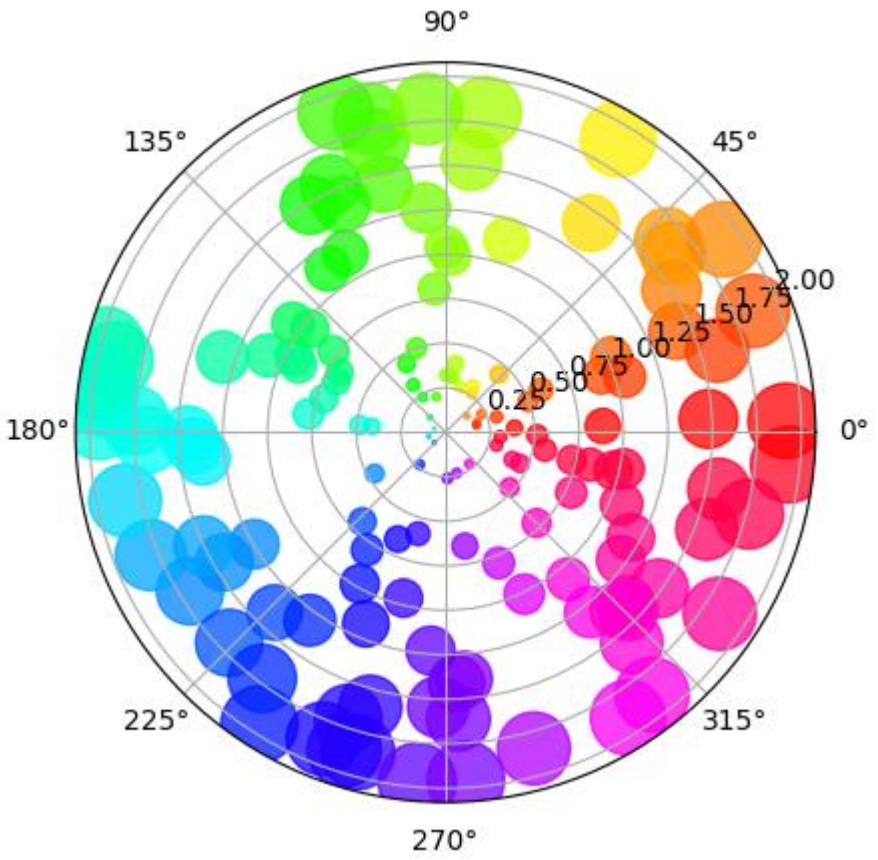
Scatter plot on polar axis

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

# Fixing random state for reproducibility
np.random.seed(19680801)

# Compute areas and colors
N = 150
r = 2 * np.random.rand(N)
theta = 2 * np.pi * np.random.rand(N)
area = 200 * r**2
colors = theta

fig = plt.figure()
ax = fig.add_subplot(projection='polar')
c = ax.scatter(theta, r, c=colors, s=area,
cmap='hsv', alpha=0.75)
```



Interactive Plots with Plotly Express

Interactive line chart:

```
import plotly.express as px
import numpy as np

# Sample data
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Create interactive line plot
fig = px.line(x=x, y=y, title='Interactive Line
Plot Example')
fig.show()
```

Interactive Plots with Plotly Express

Interactive scatter chart:

```
import plotly.express as px
import pandas as pd

# Sample data
df = px.data.iris()

# Create interactive scatter plot
fig = px.scatter(df, x='sepal_width',
y='sepal_length', color='species',
title='Interactive Scatter Plot Example')
fig.show()
```

Interactive Plots with Plotly Express

3D scatter plot:

```
import plotly.graph_objects as go
import pandas as pd

# Sample data
df = px.data.iris()

# Create a 3D scatter plot
fig = go.Figure(data=[go.Scatter3d(
    x=df['sepal_length'],
    y=df['sepal_width'],
    z=df['petal_length'],
    mode='markers',
    marker=dict(size=5, color=df['petal_width'], colorscale='Viridis', opacity=0.8))])

fig.update_layout(title='Interactive 3D Scatter Plot')
fig.show()
```