Data Science Python Libraries - Examples and Explanations

This PDF contains explanations, Python examples with comments, and expected outputs for key Python libraries used in Data Science.

---

1. NumPy

```python
import numpy as np

# Create a numpy array
arr = np.array([1, 2, 3])
# Multiply each element by 2
result = arr * 2
print(result)  # Output: [2 4 6]
```

---

2. Pandas

```python
import pandas as pd

# Create a DataFrame
data = pd.DataFrame({"A": [1, 2, 3], "B": [4, 5, 6]})
print(data)
# Output:
#    A  B
# 0  1  4
# 1  2  5
# 2  3  6
```

---

3. Matplotlib

```python
import matplotlib.pyplot as plt

# Create a simple line plot
plt.plot([1, 2, 3], [4, 5, 6])
plt.title('Simple Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()  # Output: A line plot will appear
```

---

4. Seaborn

```python
import seaborn as sns
import pandas as pd

# Sample DataFrame
data = pd.DataFrame({"A": [1, 2, 3], "B": [4, 5, 6]})

# Heatmap of correlation
sns.heatmap(data.corr(), annot=True)
plt.show()  # Output: Heatmap showing correlation values
```

---

5. SciPy

```python
from scipy import stats

# Calculate Z-scores for a list
z_scores = stats.zscore([1, 2, 3, 4, 5])
print(z_scores)  # Output: standardized values
```

---

6. scikit-learn

```python
from sklearn.linear_model import LinearRegression
import numpy as np

# Sample data
X = np.array([[1], [2], [3]])
y = np.array([2, 4, 6])

# Train linear regression model
model = LinearRegression()
model.fit(X, y)
print(model.predict([[4]]))  # Output: [8.0]
```

---

7. Jupyter Notebook

* Run `jupyter notebook` in terminal or Anaconda Prompt
* Opens browser where you can run cells interactively
* Example cell content:

```python
import pandas as pd
data = pd.DataFrame({"A": [1,2], "B": [3,4]})
```

```
data.head()
```

---

8. TensorFlow

```python
import tensorflow as tf

# Simple neural network layer
model = tf.keras.Sequential([tf.keras.layers.Dense(1)])
print(model.summary())  # Output: Summary of model layers
```

---

9. PyTorch (torch)

```python
import torch

# Create a tensor
t = torch.tensor([1.0, 2.0])
# Multiply each element by 2
result = t * 2
print(result)  # Output: tensor([2., 4.])
```

---

10. Statsmodels

```python
import statsmodels.api as sm
import numpy as np

# Sample data
X = np.array([1, 2, 3])
y = np.array([2, 4, 6])
X = sm.add_constant(X)  # Adds constant term for regression
model = sm.OLS(y, X).fit()
print(model.summary())  # Output: Regression statistics
```

---

11. Plotly

```python
import plotly.express as px

# Simple scatter plot
fig = px.scatter(x=[1, 2, 3], y=[4, 5, 6], title="Scatter Plot")
```

```
fig.show()  # Output: Interactive scatter plot in browser
```

All code snippets include **comments** to explain each step. Run them in Python to see the outputs interactively.