DATA SCIENCE LAB MANUAL

PRACTICAL 1: INSTALL & EXPLORE PYTHON LIBRARIES

Problem Statement:

Download, install, and explore NumPy, SciPy, Jupyter, Statsmodels, and Pandas.

Instructions:

Use Anaconda / VS Code Prompt:

```
pip install numpy scipy pandas statsmodels seaborn scikit-learn jupyter
```

Python Code (Version Check)

```
import numpy as np
import scipy
import pandas as pd
import statsmodels
import sklearn
import seaborn as sns

print("NumPy:", np.__version__)
print("SciPy:", scipy.__version__)
print("Pandas:", pd.__version__)
print("Statsmodels:", statsmodels.__version__)
print("Scikit-learn:", sklearn.__version__)
print("Seaborn:", sns.__version__)
```

Expected Output (It is showing version so output may differ)

```
NumPy: 1.x.x
SciPy: 1.x.x
Pandas: 1.x.x
Statsmodels: 0.x.x
Scikit-learn: 1.x.x
Seaborn: 0.x.x
```

PRACTICAL 2: NUMPY ARRAY OPERATIONS

Problem Statement: Implement Python programs using NumPy arrays.

Python Code

```
import numpy as np

a = np.array([1,2,3,4])
b = np.arange(10)
c = np.linspace(0,1,5)
d = b.reshape(2,5)

print("Array a:", a)
print("Array b:", b)
print("Linspace c:", c)
print("Reshaped d:\n", d)

print("First element of a:", a[0])
print("Slice of b:", b[2:7])

print("Addition:", a + 10)
print("Multiply:", b * 2)
```

Expected Output

```
Array a: [1 2 3 4]
Array b: [0 1 2 3 4 5 6 7 8 9]
Linspace c: [0.   0.25 0.5  0.75 1. ]
Reshaped d:
 [[0 1 2 3 4]
  [5 6 7 8 9]]
First element: 1
Slice: [2 3 4 5 6]
Addition: [11 12 13 14]
Multiply: [0 2 4 6 8 10 12 14 16 18]
```

PRACTICAL 3: PANDAS INDEXING AND DATAFRAME OPERATIONS
Problem Statement:
Perform indexing, selection, row/column operations, missing data handling, and renaming.

Python Code
```
import pandas as pd

df = pd.DataFrame({
    'Name':['Alice','Bob','Charlie','David'],
    'Age':[25,30,35,40],
    'Salary':[50000,60000,None,80000]
})

print(df)

print("\nRow 0:\n", df.loc[0])
print("\nAge column:\n", df['Age'])

df['Tax'] = df['Salary'].fillna(df['Salary'].median()) * 0.1
print("\nAdded Tax column:\n", df)

subset = df[df['Age'] > 30]
print("\nSubset where Age > 30:\n", subset)

df['Salary_filled'] = df['Salary'].fillna(df['Salary'].median())
print("\nMissing Salary Filled:\n", df)

print("\nRenamed DataFrame:\n", df.rename(columns={'Salary':'MonthlySalary'}))
```

PRACTICAL 4: DESCRIPTIVE ANALYTICS ON IRIS DATASET

Problem Statement:

Read data (text, Excel, web) and perform descriptive analytics on the Iris dataset.

Python Code

```
import seaborn as sns

df = sns.load_dataset('iris')

print(df.head())
print(df.describe())

print("\nSkewness:\n", df.skew())
print("\nKurtosis:\n", df.kurtosis())
```

Expected Output (Sample)

- First 5 rows
- Summary statistics
- Skewness
- Kurtosis

PRACTICAL 5: DIABETES DATASET ANALYSIS

Problem Statement:

Using Diabetes (sklearn) & Pima Indians dataset:
a) Univariate analysis
b) Linear & Logistic Regression
c) Multiple Regression
d) Compare results

Python Code (Sklearn Diabetes)

```
from sklearn.datasets import load_diabetes
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

data = load_diabetes()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)

feature = X['bmi']

print("Mean:", feature.mean())
print("Median:", feature.median())
print("Skew:", feature.skew())
print("Kurtosis:", feature.kurtosis())

X_train, X_test, y_train, y_test = train_test_split(feature.values.reshape(-1,1), y,
test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

print("Coef:", model.coef_)
print("Intercept:", model.intercept_)
print("Score:", model.score(X_test, y_test))
```

Expected Output (Sample)

- Mean, median, variance, SD, skewness, kurtosis
- Regression coefficient
- R^2 score

PRACTICAL 6: VISUALIZATION – PLOT TYPES

Problem Statement:

Create:
a) Normal curves
b) Density & contour plots
c) Scatter plots
d) Histograms
e) 3D plots

Python Code

```
import seaborn as sns
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

iris = sns.load_dataset("iris")

sns.histplot(iris['sepal_length'], kde=True)
plt.show()

sns.kdeplot(iris['sepal_width'], shade=True)
plt.show()

sns.scatterplot(x='sepal_length', y='petal_length', hue='species', data=iris)
plt.show()

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(iris['sepal_length'], iris['sepal_width'], iris['petal_length'])
plt.show()
```

PRACTICAL 7: CAR RENTAL PRICE ANALYTICS

Problem Statement:

Predict car rental price using car age, mileage, model, location, seasonality, and demand.

Python Code Template

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

df = pd.read_csv("car_rentals.csv")

df.fillna(df.median(), inplace=True)

df = pd.get_dummies(df, drop_first=True)

X = df.drop("rental_price", axis=1)
y = df["rental_price"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestRegressor()
model.fit(X_train, y_train)

preds = model.predict(X_test)

print("RMSE:", mean_squared_error(y_test, preds, squared=False))
```

PRACTICAL 8: CREDIT CARD APPROVAL PREDICTION

Problem Statement:

Predict approval (yes/no) using Logistic Regression.

Python Code Template

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

df = pd.read_csv("credit_card.csv")

df.fillna(df.median(), inplace=True)
df = pd.get_dummies(df, drop_first=True)

X = df.drop("approved", axis=1)
y = df["approved"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

model = LogisticRegression(max_iter=500)
model.fit(X_train, y_train)

preds = model.predict(X_test)

print(classification_report(y_test, preds))
```