EX.NO:	EMPLOYEES DATA ANALYSIS TOOL USING THE PANDAS
DATE: / /2025	

AIM

To Create a Employees data analysis tool using the Pandas library in Python.

SOURCE CODE:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import random
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import os
sns.set()
def generate_sample_dataset():
    print("\n No dataset found. Generating sample data (700 employees)...\n")
    data = {
        'EmployeeID': range(1, 701),
        'Name': [f'Employee_{i}' for i in range(1, 701)],
        'Department': random.choices(['HR', 'IT', 'Finance', 'Sales', 'Admin'], k=700),
        'Salary': [random.randint(30000, 120000) for _ in range(700)],
        'YearsExperience': [random.randint(1, 20) for _ in range(700)]
    }
    df = pd.DataFrame(data)
    df.to_csv("generated_employees.csv", index=False)
    print(" Sample dataset saved as 'generated_employees.csv'\n")
    return d
def load_data():
    file_path = input("Enter file path (press Enter to auto-generate sample dataset): ").strip()
    if file_path == "":
        return generate_sample_dataset()
    try:
        if os.path.exists(file_path):
            if file_path.endswith('.csv'):
                return pd.read_csv(file_path)
            elif file_path.endswith(('.xlsx', '.xls')):
                return pd.read_excel(file_path)
            else:
                print(" Unsupported file type. Auto-generating dataset...")
```

```
            return generate_sample_dataset()
        else:
            print(" File not found. Auto-generating dataset...")
            return generate_sample_dataset()
    except Exception as e:
        print(f" Error loading file: {e}")
        return generate_sample_dataset()
def clean_data(df):
    df = df.dropna().drop_duplicates()
    return df
def analyze_data(df):
    analysis = df.describe().T
    analysis.to_csv("salary_analysis_report.csv")
    print("\n Summary Statistics:\n", analysis)
    print("\n Report exported as 'salary_analysis_report.csv'")
    return analysis
def visualize_trend(df):
    if 'YearsExperience' in df.columns and 'Salary' in df.columns:
        plt.figure()
        sns.scatterplot(x='YearsExperience', y='Salary', data=df)
        sns.lineplot(x='YearsExperience', y='Salary', data=df)
        plt.title("Salary Trend vs Years of Experience")
        plt.show()
    else:
        print("⚠ Trend visualization skipped (missing data)")
def visualize_histogram(df):
    if 'Salary' in df.columns:
        plt.figure()
        df['Salary'].hist(bins=20)
        plt.title("Salary Distribution Histogram")
        plt.xlabel("Salary")
        plt.ylabel("Frequency")
        plt.show()
    else:
       print("⚠ Histogram skipped: No 'Salary' column found")
def predictive_model(df):
    if 'Salary' not in df.columns or 'YearsExperience' not in df.columns:
        print("⚠ Prediction skipped: Required columns missing")
        return
    X = df[['YearsExperience']]
    y = df['Salary']
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42
    )
    model = LinearRegression()
    model.fit(X_train, y_train)
```

```
    predictions = model.predict(X_test)
   print("\n Salary Prediction Performance:")
    print("MSE:", mean_squared_error(y_test, predictions))
    print("R2 Score:", r2_score(y_test, predictions))
    plt.figure()
    plt.scatter(y_test, predictions)
    plt.xlabel("Actual Salary")
    plt.ylabel("Predicted Salary")
    plt.title("Actual vs Predicted Salary")
    plt.show()
def main():
    print("\n---- EMPLOYEE SALARY DATA ANALYSIS TOOL (ADVANCED) ----\n")
    df = load_data()
    print("\n Data Preview:\n", df.head())
    df = clean_data(df)
    analyze_data(df)
    visualize_trend(df)
    visualize_histogram(df)
    predictive_model(df
    print("\n Analysis Completed Successfully!")
if __name__ == "__main__":
    main()
```

OUTPUT:

---- EMPLOYEE SALARY DATA ANALYSIS TOOL (ADVANCED)

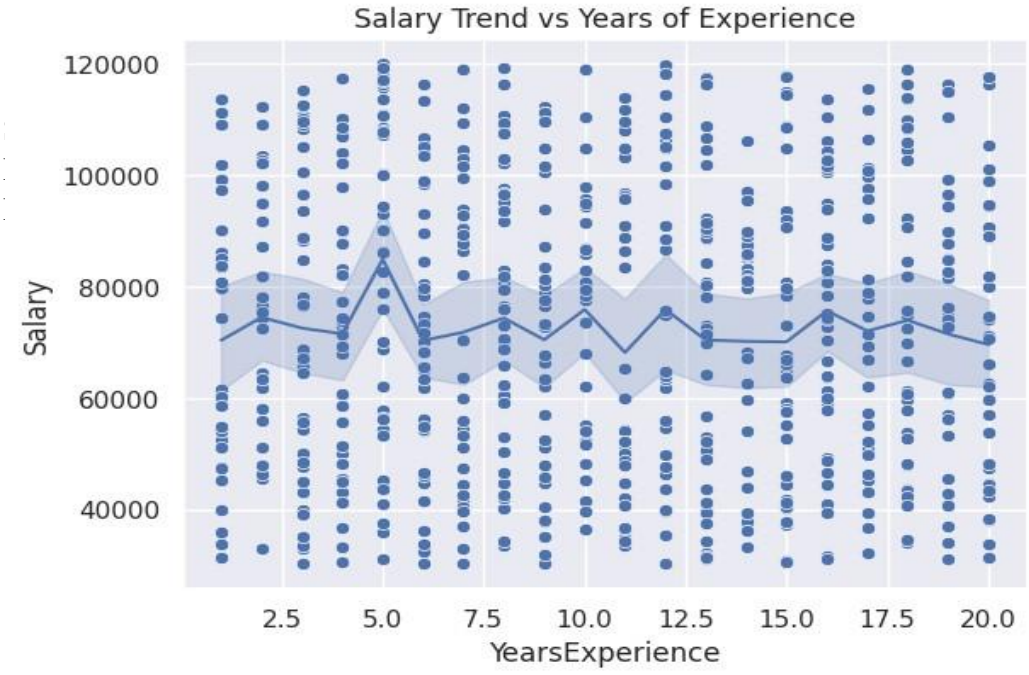
Data Preview:

	EmployeeID	Name	Department	Salary	YearsExperience
0	1	Employee_1	Sales	78392	6
1	2	Employee_2	Finance	59926	8
2	3	Employee_3	Admin	83040	10
3	4	Employee_4	Sales	68837	5
4	5	Employee_5	IT	113746	16

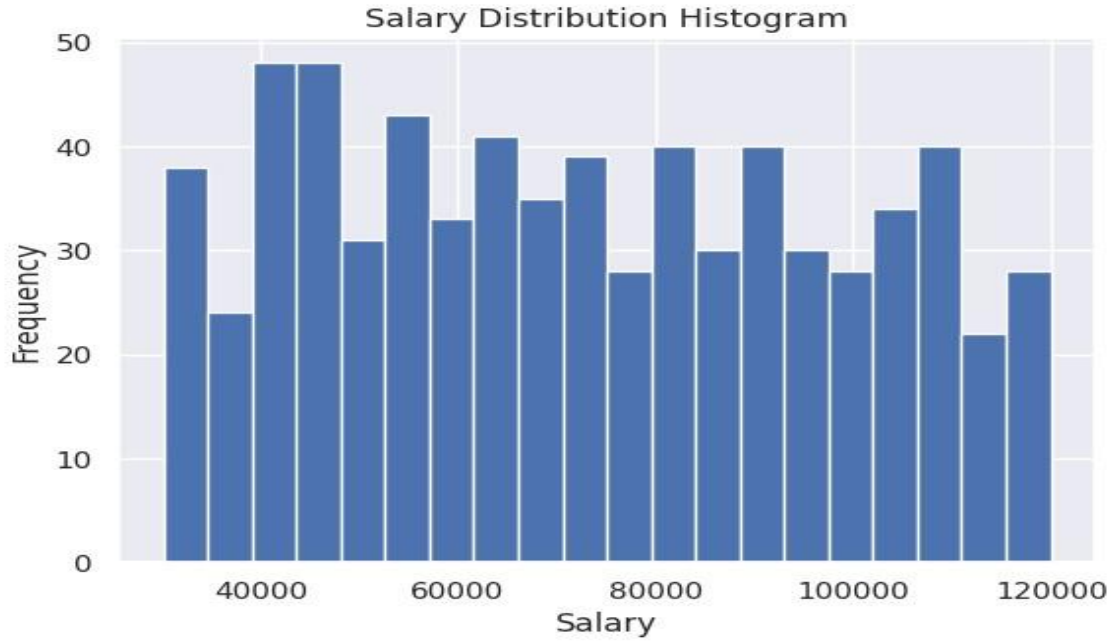
Summary Statistics:

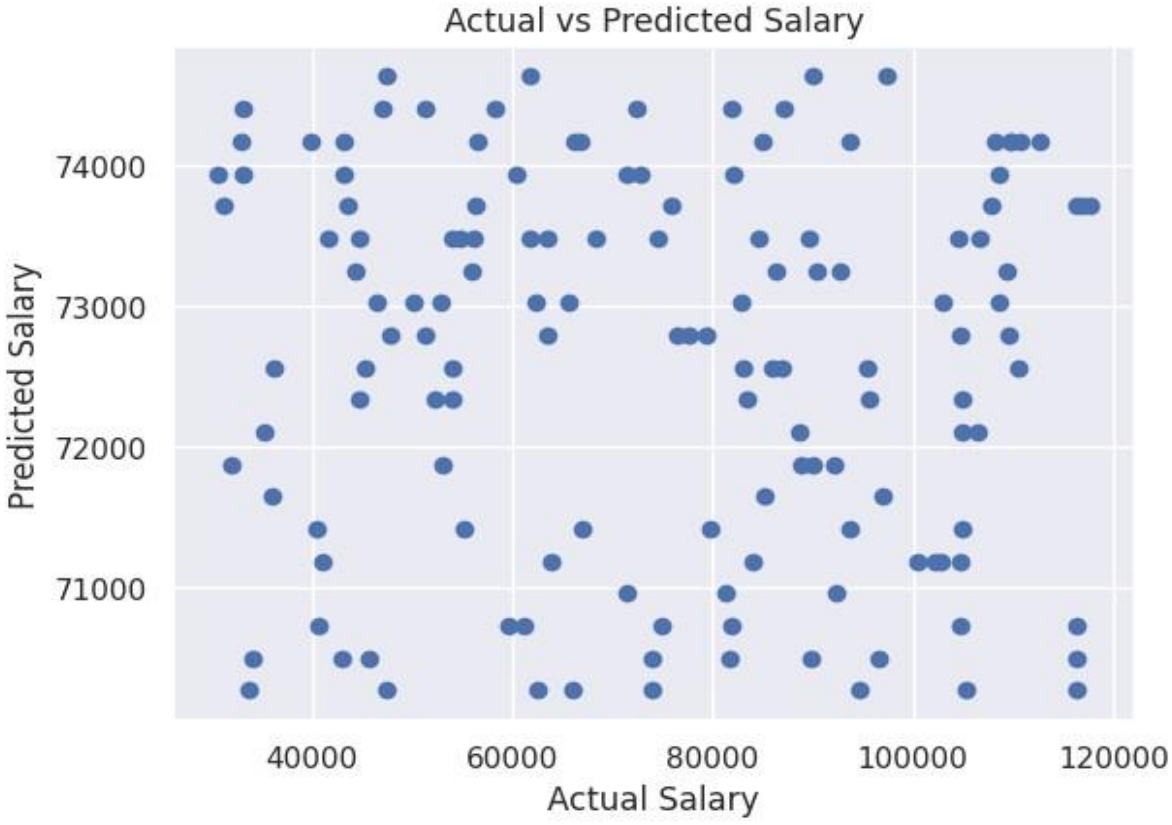
	count	mean	std	min	25%	50%	75%	max
EmployeeID	700.0	350.500000	202.216880	1.0	175.75	350.5	525.25	700.0
Salary	700.0	72853.258571	25405.976817	30175.0	51116.75	72444.0	94080.00	119975.0
YearsExperience	700.0	10.504286	5.667843	1.0	6.00	10.0	16.00	20.0

Report exported as 'salary_analysis_report.csv'



Salary Prediction :
MSE: 648232825.9017441
R2 score : -0.1279633599655





RESULT:

The program has been successfully executed.