1. DataFrame Creation and Basic Operations

Question: Create a Pandas DataFrame with the following data:

Employee Department Salary Age John IT 60000 30 Alice HR 55000 28 Bob Finance 70000 35 Emma IT 72000 32 Perform the following operations:

Display the first two rows of the DataFrame.

Add a new column "Experience" with values [5, 3, 7, 6].

Find the average salary of all employees.

b)Perform the following tasks: Create dataset of students with name and 3 subjects

1)display all the students who scored more than 80 in math 2)sort the dataframe in descending order based on the Science score 3)Find the student with highest English score

```
import pandas as pd

# Create the DataFrame
data = {'Employee': ['John', 'Alice', 'Bob', 'Emma'],
        'Department': ['IT', 'HR', 'Finance', 'IT'],
        'Salary': [60000, 55000, 70000, 72000],
        'Age': [30, 28, 35, 32]}
df = pd.DataFrame(data)

# Display the first two rows
print("First two rows:")
print(df.head(2))

# Add 'Experience' column
df['Experience'] = [5, 3, 7, 6]
print("\nDataFrame with Experience column:")
print(df)

# Calculate average salary
average_salary = df['Salary'].mean()
print("\nAverage salary:", average_salary)
```

```
First two rows:
  Employee Department  Salary  Age
0     John         IT   60000   30
1    Alice         HR   55000   28

DataFrame with Experience column:
  Employee Department  Salary  Age  Experience
0     John         IT   60000   30           5
1    Alice         HR   55000   28           3
2      Bob    Finance   70000   35           7
3     Emma         IT   72000   32           6
```

```
Average salary: 64250.0
```

```
import pandas as pd

# Create the student dataset
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
        'Math': [85, 70, 92, 78, 88],
        'Science': [90, 82, 75, 88, 95],
        'English': [78, 85, 90, 76, 82]}
df = pd.DataFrame(data)

# 1) Display students who scored more than 80 in Math
print("Students who scored more than 80 in Math:")
print(df[df['Math'] > 80])

# 2) Sort the DataFrame by Science score in descending order
print("\nDataFrame sorted by Science score (descending):")
print(df.sort_values(by=['Science'], ascending=False))

# 3) Find the student with the highest English score
print("\nStudent with the highest English score:")
print(df.loc[df['English'].idxmax()])
```

```
Students who scored more than 80 in Math:
      Name  Math  Science  English
0    Alice    85       90       78
2  Charlie    92       75       90
4      Eve    88       95       82

DataFrame sorted by Science score (descending):
      Name  Math  Science  English
4      Eve    88       95       82
0    Alice    85       90       78
3    David    78       88       76
1      Bob    70       82       85
2  Charlie    92       75       90

Student with the highest English score:
Name       Charlie
Math            92
Science         75
English         90
Name: 2, dtype: object
```

Lab2: Suppose you want to track and analyze your household expenses for a month.

You have recorded the expenses for various categories, such as groceries, utilities,

rent, transportation, and entertainment. You can represent this expense data using a

Pandas Series.

Input:

Expense categories

categories = ['Groceries', 'Utilities', 'Rent', 'Transportation', 'Entertainment']

Monthly expense data (example data in USD)

expenses = [500, 200, 1200, 300, 150]

```
import pandas as pd

# Expense data
categories = ['Groceries', 'Utilities', 'Rent', 'Transportation',
'Entertainment']
expenses = [500, 200, 1200, 300, 150]

# Create Pandas Series
expenses_series = pd.Series(expenses, index=categories)

# Display the Series
print(expenses_series)
```

```
Groceries          500
Utilities          200
Rent              1200
Transportation     300
Entertainment      150
dtype: int64
```

Lab3: Suppose you want to track and analyze the monthly energy consumption in your home. You have recorded the monthly energy usage for electricity and gas over a year, and you want to represent this data using Pandas Series.

Input:

Months in a year

months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']

Monthly energy consumption data (example data in kilowatt-hours for electricity and

therms for gas)

electricity_usage = [350, 320, 310, 330, 340, 370, 380, 360, 350, 330, 320, 330]

gas_usage = [20, 18, 16, 15, 12, 10, 8, 9, 12, 15, 17, 19]

```
import pandas as pd

# Energy consumption data
months = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December']
electricity_usage = [350, 320, 310, 330, 340, 370, 380, 360, 350, 330,
320, 330]
gas_usage = [20, 18, 16, 15, 12, 10, 8, 9, 12, 15, 17, 19]

# Create Pandas Series
electricity_series = pd.Series(electricity_usage, index=months)
gas_series = pd.Series(gas_usage, index=months)

# Display the Series
print("Electricity Usage:")
print(electricity_series)
print("\nGas Usage:")
print(gas_series)
```

```
Electricity Usage:
January      350
February     320
March        310
April        330
May          340
June         370
July         380
August       360
September    350
October      330
November     320
December     330
dtype: int64

Gas Usage:
January      20
February     18
March        16
April        15
May          12
```

```
June          10
July           8
August         9
September     12
October       15
November      17
December      19
dtype: int64
```

Lab4:Suppose you are managing a website and want to analyze the monthly revenue generated from advertising. You have recorded the monthly revenue for the past year, and you want to represent this data using a Pandas Series.

Input:

Months in a year

months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']

Monthly advertising revenue data (example data in USD)

revenue = [5000, 5200, 4800, 5400, 5600, 5800, 6100, 5900, 6200, 6500, 7000, 6900]

```
import pandas as pd

# Revenue data
months = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December']
revenue = [5000, 5200, 4800, 5400, 5600, 5800, 6100, 5900, 6200, 6500,
7000, 6900]

# Create Pandas Series
revenue_series = pd.Series(revenue, index=months)

# Display the Series
print(revenue_series)
```

```
January      5000
February     5200
March        4800
April        5400
```

```
May          5600
June         5800
July         6100
August       5900
September    6200
October      6500
November     7000
December     6900
dtype: int64
```