WEEK 4 LAB

Rovin Castelino

240957001

PC 46

```
import pandas as pd

temperatures = pd.Series([22, 24, 20, 18, 25, 23, 21],
                         index=["Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday", "Sunday"])

total_temperature = 0
for temp in temperatures:
    total_temperature += temp
average_temperature = total_temperature / len(temperatures)
print(f"Average temperature for the week: {average_temperature}°C")
```

```
Average temperature for the week: 21.857142857142858°C
```

```
max_temp = temperatures[0]
min_temp = temperatures[0]
max_day = "Monday"
min_day = "Monday"

for day, temp in temperatures.items():
    if temp > max_temp:
        max_temp = temp
        max_day = day
    if temp < min_temp:
        min_temp = temp
        min_day = day

print(f"Maximum temperature: {max_temp}°C on {max_day}")
print(f"Minimum temperature: {min_temp}°C on {min_day}")
```

```
Maximum temperature: 25°C on Friday
Minimum temperature: 18°C on Thursday

C:\Users\chira\AppData\Local\Temp\ipykernel_15136\2403270483.py:1:
FutureWarning: Series.__getitem__ treating keys as positions is
deprecated. In a future version, integer keys will always be treated
as labels (consistent with DataFrame behavior). To access a value by
position, use `ser.iloc[pos]`
  max_temp = temperatures[0]
C:\Users\chira\AppData\Local\Temp\ipykernel_15136\2403270483.py:2:
```

```
FutureWarning: Series.__getitem__ treating keys as positions is
deprecated. In a future version, integer keys will always be treated
as labels (consistent with DataFrame behavior). To access a value by
position, use `ser.iloc[pos]`
  min_temp = temperatures[0]
```

```
print("Temperatures greater than 22°C:")
for day, temp in temperatures.items():
    if temp > 22:
        print(f"{day}: {temp}°C")
```

```
Temperatures greater than 22°C:
Tuesday: 24°C
Friday: 25°C
Saturday: 23°C
```

```
print("Temperatures in Fahrenheit:")
for day, temp in temperatures.items():
    fahrenheit = temp * 9/5 + 32
    print(f"{day}: {fahrenheit}°F")
```

```
Temperatures in Fahrenheit:
Monday: 71.6°F
Tuesday: 75.2°F
Wednesday: 68.0°F
Thursday: 64.4°F
Friday: 77.0°F
Saturday: 73.4°F
Sunday: 69.8°F
```

```
print("Days with temperatures above the average:")
for day, temp in temperatures.items():
    if temp > average_temperature:
        print(f"{day}: {temp}°C")
```

```
Days with temperatures above the average:
Monday: 22°C
Tuesday: 24°C
Friday: 25°C
Saturday: 23°C
```

Task 2

```
data = {
    "Roll Number": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    "Name": ["Alice", "Bob", "Charlie", "David", "Eva", "Frank",
"Grace", "Hannah", "Ivy", "Jack"],
    "Gender": ["F", "M", "M", "M", "F", "M", "F", "F", "F", "M"],
```

```
    "Marks1": [85, 72, 64, 50, 90, 55, 79, 88, 92, 76],
    "Marks2": [70, 80, 55, 45, 60, 85, 70, 95, 65, 80],
    "Marks3": [88, 75, 80, 60, 95, 92, 85, 78, 88, 76]
}
students_df = pd.DataFrame(data)

students_df["Total Marks"] = 0
for i in range(len(students_df)):
    students_df.at[i, "Total Marks"] = students_df.at[i, "Marks1"] +
students_df.at[i, "Marks2"] + students_df.at[i, "Marks3"]
print("Student Data with Total Marks:")
print(students_df)
```

```
Student Data with Total Marks:
   Roll Number     Name Gender  Marks1  Marks2  Marks3  Total Marks
0            1    Alice      F      85      70      88          243
1            2      Bob      M      72      80      75          227
2            3  Charlie      M      64      55      80          199
3            4    David      M      50      45      60          155
4            5      Eva      F      90      60      95          245
5            6    Frank      M      55      85      92          232
6            7    Grace      F      79      70      85          234
7            8   Hannah      F      88      95      78          261
8            9      Ivy      F      92      65      88          245
9           10     Jack      M      76      80      76          232
```

```
lowest_marks1 = students_df["Marks1"][0]
for mark in students_df["Marks1"]:
    if mark < lowest_marks1:
        lowest_marks1 = mark
print(f"Lowest marks in Marks1: {lowest_marks1}")
```

```
Lowest marks in Marks1: 50
```

```
highest_marks2 = students_df["Marks2"][0]
for mark in students_df["Marks2"]:
    if mark > highest_marks2:
        highest_marks2 = mark
print(f"Highest marks in Marks2: {highest_marks2}")
```

```
Highest marks in Marks2: 95
```

```
total_marks3 = 0
for mark in students_df["Marks3"]:
    total_marks3 += mark
average_marks3 = total_marks3 / len(students_df["Marks3"])
```

```
print(f"Average marks in Marks3: {average_marks3}")
```

```
Average marks in Marks3: 81.7
```

```
# e. Find student name with highest average marks
highest_avg = 0
best_student = ""
for i in range(len(students_df)):
    average = (students_df.at[i, "Marks1"] + students_df.at[i,
"Marks2"] + students_df.at[i, "Marks3"]) / 3
    if average > highest_avg:
        highest_avg = average
        best_student = students_df.at[i, "Name"]
print(f"Student with the highest average marks: {best_student}")
```

```
Student with the highest average marks: Hannah
```

```
# f. Find how many students failed in Marks2 (<40)
failed_students = 0
for mark in students_df["Marks2"]:
    if mark < 40:
        failed_students += 1
print(f"Number of students who failed in Marks2: {failed_students}")
```

```
Number of students who failed in Marks2: 0
```