Pandas practice set 1 solution

1Load the dataset and print the first 5 rows.

Hint: Use the function `pd.read_csv()` and then call `.head()`.

2Show the last 3 rows of the dataset.

Hint: Use `.tail(3)` instead of `.tail()`.

3Print the total number of rows and columns in the dataset.

Hint: Use `.shape` → it returns `(rows, columns)`.

---

4 Display only the column names of the dataset.

Hint: Use `.columns`.

5Select only the “Temperature” column and print it.

Hint: Use `df["Temperature"]`.

---

6 Show all rows where Temperature is greater than 30°C.

Hint: Use filtering —
`df[df["Temperature"] > 30]`.

---

7 Show only the rows from index 5 to index 10 (inclusive).

Hint: Use `.loc[5:10]`.

8 Add a new column named `Temp_Fahrenheit` based on Temperature.

Formula:
F = (C × 9/5) + 32
Hint:
`df["Temp_Fahrenheit"] = df["Temperature"]*9/5 + 32.`

9 Replace missing humidity values with the value 50.

Hint:
`df["Humidity"].fillna(50).`

10. Drop the column named "Air_Quality" from the dataset.

Hint:
`df.drop("Air_Quality", axis=1, inplace=True).`

11. Show the details of all days where the Temperature is higher than the average Temperature of the first 5 rows.

Hint:

- Use `df.head()` to get the first 5 rows
- Calculate average manually → check values
- Filter using:
  `df[df["Temperature"] > some_value]`

12. Display only those rows where BOTH Temperature > 25 AND Air_Quality is missing (NaN).

Hint:
Use:

```
df[(df["Temperature"] > 25) & (df["Air_Quality"].isnull())]
```

---

13. Add a new column called “TempPlusHumidity” = Temperature + Humidity, and display the *last 5* updated rows.

Hint:

```
df["TempPlusHumidity"] = df["Temperature"] + df["Humidity"]
```

Then → `df.tail()`

---

14. Select rows from index 3 to index 10 but show ONLY the columns Temperature and Humidity.

Hint:

```
df.loc[3:10, ["Temperature", "Humidity"]]
```

---

15. Update the Humidity of the 2nd row to 50, then print the updated row.

Hint:

```
df.loc[1, "Humidity"] = 50
```

Then:

```
df.loc[1]
```

---

16. Drop the row where Temperature is the lowest in the dataset.

Hint:

- First check smallest value using `df.sort_values("Temperature")` is NOT allowed
- So instead:
    - Use `df.describe()` → find Min temperature

  - Drop using:
    `df.drop(index_of_min, inplace=True)`

(You manually identify the index of min temperature)

---

17. Show all rows where any column has a missing value.

Hint:

```
df[df.isnull().any(axis=1)]
```

---

18. Replace the missing values in the Humidity column with 0, then print the first 10 rows.

Hint:
`df["Humidity"].fillna(0)`
Then → `df.head(10)`

---

19. Create a new column named “Category” such that:

- If Temperature > 30 → Category = "HOT"
- Otherwise → Category = "NORMAL"
  Hint:
  You cannot use apply or np.where.
  So:

```
df.loc[df["Temperature"] > 30, "Category"] = "HOT"
df.loc[df["Temperature"] <= 30, "Category"] = "NORMAL"
```

---

20. Remove the column with the most missing values from the dataset.

Hint:

- Use `df.isnull().sum()` → manually see which column is highest
- Drop using:
  `df.drop("ColumnName", axis=1, inplace=True)`