Assignment HW_02_EDA

Idris Mahamat

1- Loading of the data set

- To do so first we need to import all our libraries
  - Panda
  - Matplotlib
  - seaborn

HW_02_EDA_Assignment.ipynb ●

HW_02_EDA_Assignment.ipynb > ...

+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ··· Python 3.10.12

Assignment_HW_02_EDA_IdrisMahamat

markdown

```python
import pandas as pd
import matplotlib.pyplot as pt
import seaborn as se
```

[14] ✓ 0.0s Python

```python
# 1- now let's load the dataset  breast-cancer-wisconsin_pfizer05.csv

chemin_fichier = '/home/elcaskerito/Documents/Stevens Jupiter/breast-cancer-wisconsin_pfizer05.csv'
df = pd.read_csv(chemin_fichier)

print(df.info())
print(df.head())
```

[10] Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Sample  35 non-null     int64
 1   F1      35 non-null     int64
 2   F2      35 non-null     int64
 3   F3      35 non-null     int64
 4   F4      35 non-null     int64
 5   F5      35 non-null     int64
 6   F6      34 non-null     float64
 7   F7      35 non-null     int64
 8   F8      35 non-null     int64
 9   F9      35 non-null     int64
 10  Class   35 non-null     int64
dtypes: float64(1), int64(10)
```

I. summarizing each column (min, max, mean)

- To do so we are going to use the method describe() from panda.

	sample	F1	F2	F3	F4	F5	F6	F7	F8	F9	class
mean	1.063904e+06	5.485714	4.000000	4.142857	2.942857	3.771429	4.382353	3.942857	3.685714	2.714286	2.857143
min	1.280590e+05	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000
max	1.369821e+06	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	4.000000

```python
# I. Summarize each column by min,max,mean

detail = df.describe(include='all')
print(detail)
```

[29] ✓ 0.0s

```
                Sample         F1         F2         F3         F4         F5  \
count     3.500000e+01  35.000000  35.000000  35.000000  35.000000  35.000000
mean      1.063904e+06   5.485714   4.000000   4.142857   2.942857   3.771429
std       2.728643e+05   3.211848   3.605551   3.573949   3.262468   2.755590
min       1.280590e+05   1.000000   1.000000   1.000000   1.000000   1.000000
25%       1.033582e+06   2.500000   1.000000   1.000000   1.000000   2.000000
50%       1.137156e+06   5.000000   2.000000   3.000000   1.000000   2.000000
75%       1.233062e+06   8.000000   7.000000   7.000000   3.000000   6.000000
max       1.369821e+06  10.000000  10.000000  10.000000  10.000000  10.000000

                F6         F7         F8         F9      Class
count    34.000000  35.000000  35.000000  35.000000  35.000000
mean      4.382353   3.942857   3.685714   2.714286   2.857143
std       3.915894   2.300164   3.668398   2.936298   1.004193
min       1.000000   1.000000   1.000000   1.000000   2.000000
25%       1.000000   2.000000   1.000000   1.000000   2.000000
50%       1.500000   3.000000   1.000000   1.000000   2.000000
75%       8.750000   5.000000   8.000000   3.000000   4.000000
max      10.000000  10.000000  10.000000  10.000000   4.000000
```

II. Identifying the missing value

- To do so we are going to use the method isnull().sum() from panda.

```
# II. Identifying missing values

ms_val = df.isnull().sum()
print(ms_val)
```

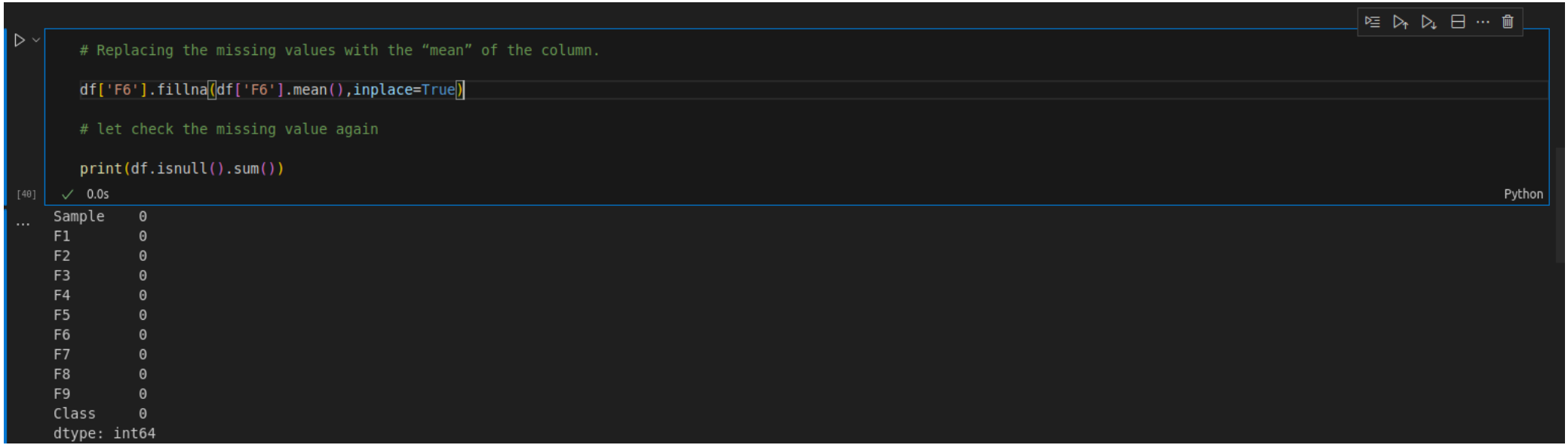
[14] ✓ 0.0s Python

```
Sample    0
F1        0
F2        0
F3        0
F4        0
F5        0
F6        1
F7        0
F8        0
F9        0
Class     0
dtype: int64
```

F6 has 1 missing value

III. Replacing the missing values with the “mean” of the column.

- To do so we are going to specify the column of the missing value then use the function fillna() passing as attribute the mean of the missing value column.



IV. Displaying the frequency table of “Class” vs. F6

- To do so we are going use the function groupby() or crosstab

```
# Displaying the frequency table of “Class” vs. F6 using crosstab
ts = pd.crosstab(df["F6"], df["Class"])
print(ts)

# print(df.info())
```

[80] ✓ 0.0s

```
Class       2  4
F6
1.000000   17  0
2.000000    1  0
3.000000    0  1
4.382353    1  0
5.000000    1  2
8.000000    0  3
9.000000    0  2
10.000000   0  7
```

IV. Displaying the scatter plot of F1 to F6, one pair at a time

- To do so we are going use the function figure(), xlabel(), ylabel(), title() from Matplotlib and scatterplot from seaborn in a Loop for.

```
# v. Displaying the scatter plot of F1 to F6, one pair at a time

fn = ['F1', 'F2', 'F3', 'F4', 'F5', 'F6']

for i in range(len(fn)):
    for j in range(i+1, len(fn)):
        pt.figure(figsize=(6,4))
        se.scatterplot(x=df[fn[i]], y=df[fn[j]])
        pt.xlabel(fn[i])
        pt.ylabel(fn[j])
        pt.title(f'Scatter Plot of {fn[i]} vs {fn[j]}')
        pt.show()
```

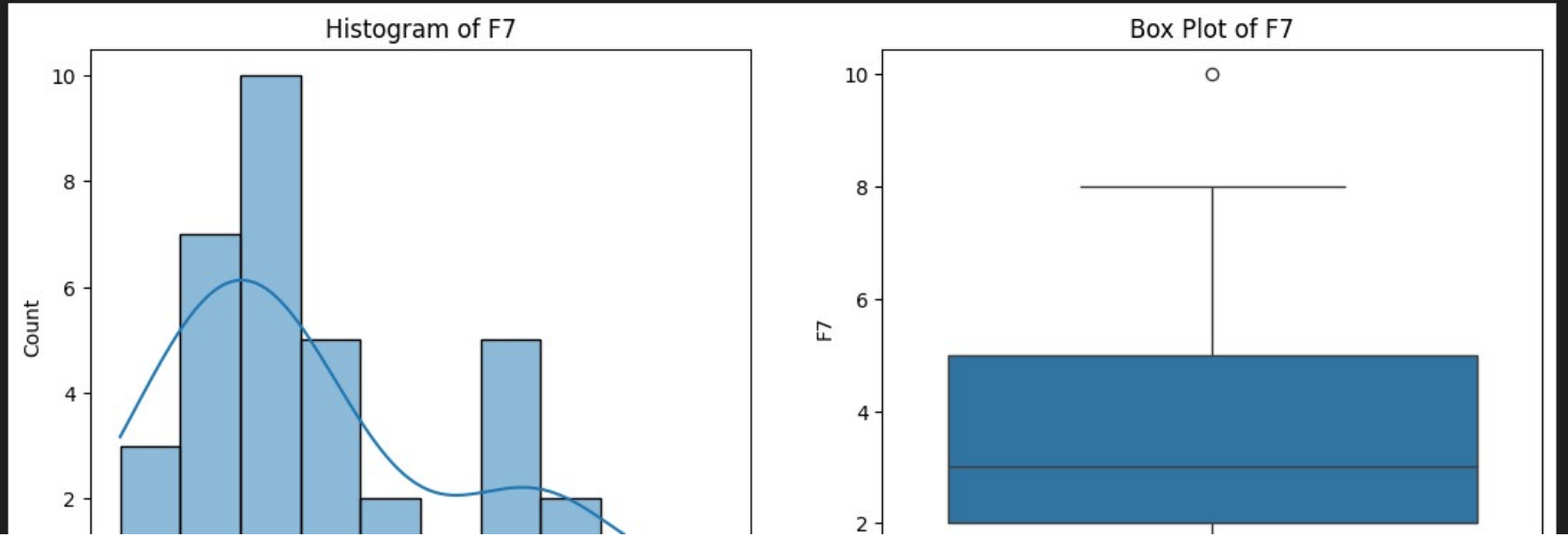
Results is shown in the Jupyter file

VI. Show histogram box plot for columns F7 to F9

- To do so we are going use the function figure(), subplot(), title() from Matplotlib and histplot(), boxplot() from seaborn in a Loop for.

```python
# VI. Show histogram box plot for columns F7 to F9

fn_f7F9 = ['F7', 'F8', 'F9']

for fn in fn_f7F9:
    pt.figure(figsize=(13, 5))

    # Histogram
    pt.subplot(1, 2, 1)
    se.histplot(df[fn], bins=10, kde=True)
    pt.title(f'Histogram of {fn}')

    # Box plot
    pt.subplot(1, 2, 2)
    se.boxplot(y=df[fn])
    pt.title(f'Box Plot of {fn}')

    pt.show()
```

✓ 0.7s



2. Delete all the objects from your R/Python- environment. Reload the “breast-cancer-wisconsin.data.csv” from canvas into R/Python. Remove any row with a missing value in any of the columns.

```python
del df
# Reload the “breast-cancer-wisconsin_pfizer05.csv” into Python
df = pd.read_csv(chemin_fichier)

# Remove all row with a missing value in any columns
df_cleaned = df.dropna()

# Display the cleaned dataset info
print(df_cleaned.info())
print(df_cleaned.head())
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 34 entries, 0 to 34
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Sample  34 non-null     int64
 1   F1      34 non-null     int64
 2   F2      34 non-null     int64
 3   F3      34 non-null     int64
 4   F4      34 non-null     int64
 5   F5      34 non-null     int64
 6   F6      34 non-null     float64
 7   F7      34 non-null     int64
 8   F8      34 non-null     int64
 9   F9      34 non-null     int64
 10  Class   34 non-null     int64
dtypes: float64(1), int64(10)
memory usage: 3.2 KB
None
    Sample  F1  F2  F3  F4  F5    F6  F7  F8  F9  Class
0  1198641  10  10   6   3   3  10.0   4   3   2      4
1  1080233   7   6   6   3   2  10.0   7   1   1      4
3   740492   1   1   1   1   2   1.0   3   1   1      2
4  1120559   8   3   8   3   4   9.0   8   9   8      4
5  1369821  10  10  10  10   5  10.0  10  10   7      4
```

OBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS DEVDB JUPYTER Pyth

```
ase) elcaskerito@elcaskerito-Latitude-5591:~/Documents/Stevens Jupiter$
```