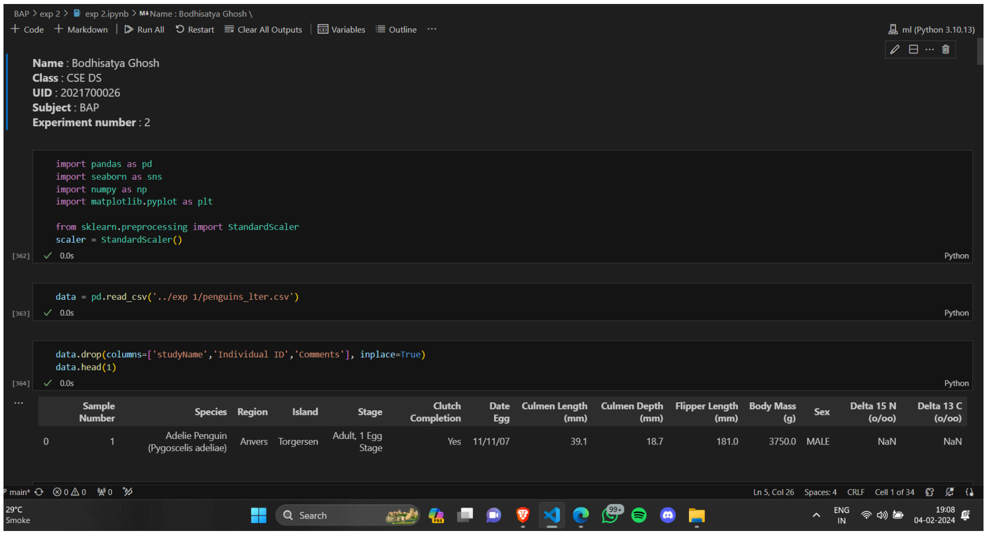
Name	**Bodhisatya Ghosh**
UID no.	**2021700026**

Experiment 2	
HONOUR PLEDGE	**“ I hereby declare that the documentation, code and output attached with this lab experiment has been completed by me in accordance with the highest standards of honesty. I confirm that I have not plagiarized OR used unauthorized materials OR given or received illegitimate help for completing this experiment. I will uphold equity and honesty in the evaluation of my work and if found guilty of plagiarism or dishonesty, will bear the consequences as outlined in the ‘integrity’ section of the lab rubrics. I am doing so in order to maintain a community built around this code of honour”**
PROBLEM STATEMENT :	**Data Cleaning and Preprocessing:** ● Handle missing values by imputing them with mean, median, or mode. Reason out which is more suitable (mean, median or mode) for your dataset and which is not ● Removing outliers based on a specific threshold. Give reasons for your choice of threshold. What do the outliers in your dataset tell you? ● Transform variables using log transformation or standardization. What possibly can go wrong when you do not standardize your data? What are the reasons for using log transformation on your variables and when should you definitely use it? ● Remove duplicate records from a data frame. In case your dataset does not have a exact duplicate rows, can you reason about strategies for identifying and deduplicating your dataset based on a subset of features? ● Standardize date formats across your dataset Is there a certain date-format that you would prefer? why?
THEORY:	

PROGRAM:



BAP > exp 2 > exp 2.ipynb > M↓Name : Bodhisatya Ghosh \
+ Code + Markdown | ▷ Run All ↺ Restart ☰ Clear All Outputs | Variables ☰ Outline ···
ml (Python 3.10.13)

Name : Bodhisatya Ghosh
Class : CSE DS
UID : 2021700026
Subject : BAP
Experiment number : 2

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

[362] ✓ 0.0s Python

```
data = pd.read_csv('../exp 1/penguins_lter.csv')
```

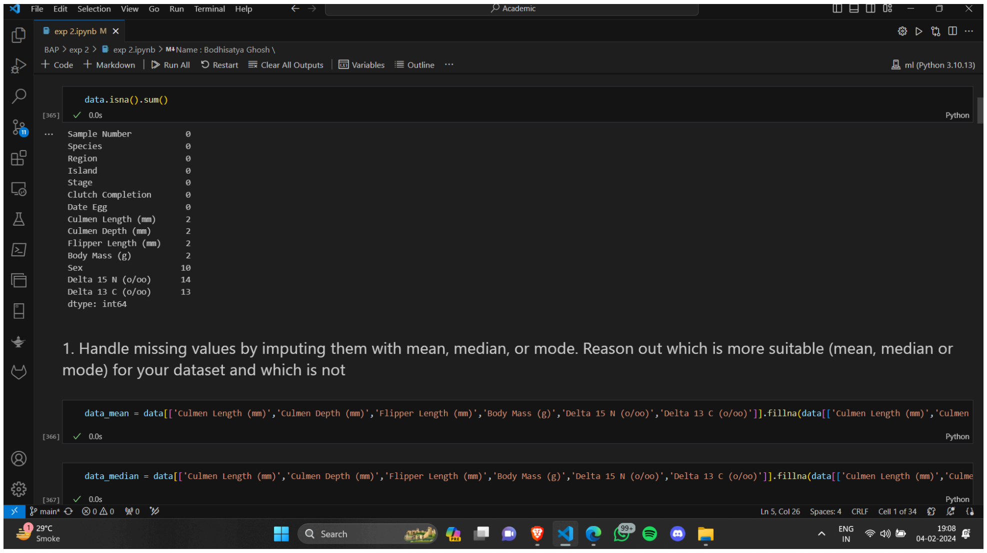
[363] ✓ 0.0s Python

```
data.drop(columns=['studyName','Individual ID','Comments'], inplace=True)
data.head(1)
```

[364] ✓ 0.0s Python

	Sample Number	Species	Region	Island	Stage	Clutch Completion	Date Egg	Culmen Length (mm)	Culmen Depth (mm)	Flipper Length (mm)	Body Mass (g)	Sex	Delta 15 N (o/oo)	Delta 13 C (o/oo)
0	1	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	Yes	11/11/07	39.1	18.7	181.0	3750.0	MALE	NaN	NaN

main* ⊗ 0 ⚠ 0 0
Ln 5, Col 26 Spaces: 4 CRLF Cell 1 of 34
29°C Smoke
Search
ENG IN
19:08 04-02-2024



File Edit Selection View Go Run Terminal Help
Academic
exp 2.ipynb M ×
BAP > exp 2 > exp 2.ipynb > M↓Name : Bodhisatya Ghosh \
+ Code + Markdown | ▷ Run All ↺ Restart ☰ Clear All Outputs | Variables ☰ Outline ···
ml (Python 3.10.13)

```
data.isna().sum()
```

[365] ✓ 0.0s Python

```
Sample Number           0
Species                 0
Region                  0
Island                  0
Stage                   0
Clutch Completion       0
Date Egg                0
Culmen Length (mm)      2
Culmen Depth (mm)       2
Flipper Length (mm)     2
Body Mass (g)           2
Sex                    10
Delta 15 N (o/oo)      14
Delta 13 C (o/oo)      13
dtype: int64
```

1. Handle missing values by imputing them with mean, median, or mode. Reason out which is more suitable (mean, median or mode) for your dataset and which is not

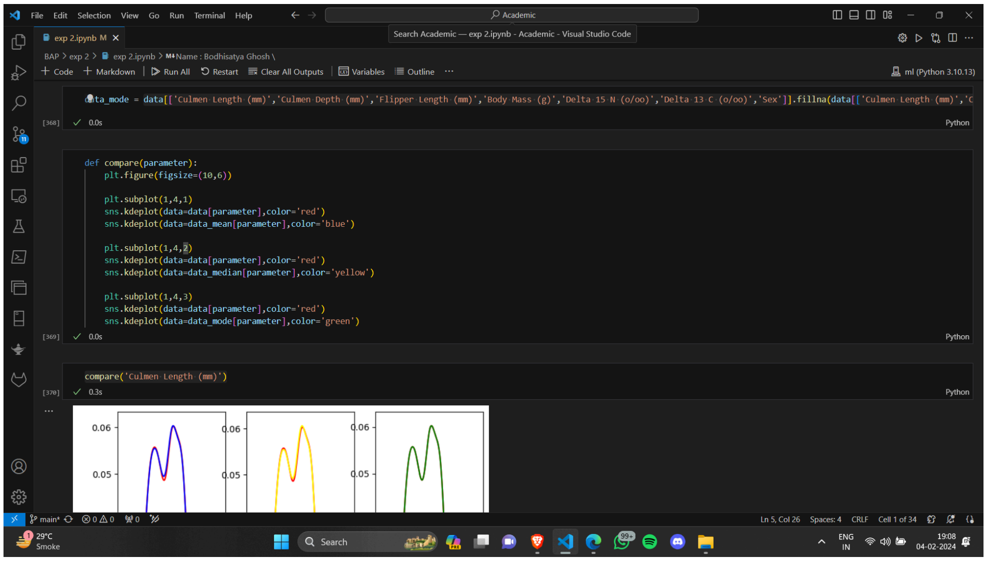
```
data_mean = data[['Culmen Length (mm)','Culmen Depth (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)','Delta 13 C (o/oo)']].fillna(data[['Culmen Length (mm)','Culmen
```

[366] ✓ 0.0s Python

```
data_median = data[['Culmen Length (mm)','Culmen Depth (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)','Delta 13 C (o/oo)']].fillna(data[['Culmen Length (mm)','Culme
```

[367] ✓ 0.0s Python

main* ⊗ 0 ⚠ 0 0
Ln 5, Col 26 Spaces: 4 CRLF Cell 1 of 34
29°C Smoke
Search
ENG IN
19:08 04-02-2024



File Edit Selection View Go Run Terminal Help
Academic
Search Academic — exp 2.ipynb - Academic - Visual Studio Code
exp 2.ipynb M ×
BAP > exp 2 > exp 2.ipynb > M↓Name : Bodhisatya Ghosh \
+ Code + Markdown | ▷ Run All ↺ Restart ☰ Clear All Outputs | Variables ☰ Outline ···
ml (Python 3.10.13)

```
data_mode = data[['Culmen Length (mm)','Culmen Depth (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)','Delta 13 C (o/oo)','Sex']].fillna(data[['Culmen Length (mm)','C
```

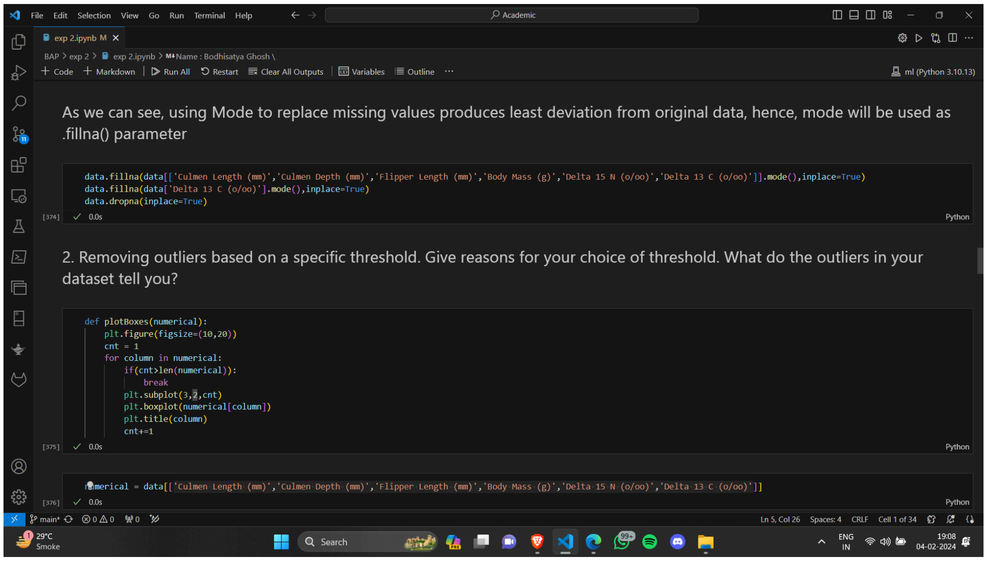
[368] ✓ 0.0s Python

```
def compare(parameter):
    plt.figure(figsize=(10,6))

    plt.subplot(1,4,1)
    sns.kdeplot(data=data[parameter],color='red')
    sns.kdeplot(data=data_mean[parameter],color='blue')

    plt.subplot(1,4,2)
    sns.kdeplot(data=data[parameter],color='red')
    sns.kdeplot(data=data_median[parameter],color='yellow')

    plt.subplot(1,4,3)
    sns.kdeplot(data=data[parameter],color='red')
    sns.kdeplot(data=data_mode[parameter],color='green')
```

[369] ✓ 0.0s Python

```
compare('Culmen Length (mm)')
```

[370] ✓ 0.3s Python



As we can see, using Mode to replace missing values produces least deviation from original data, hence, mode will be used as .fillna() parameter

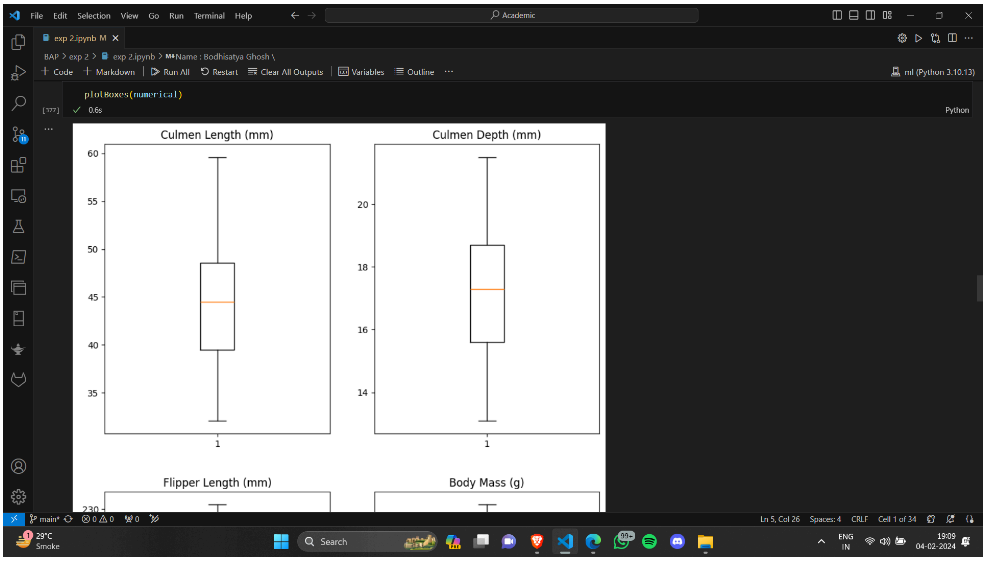
```
data.fillna(data[['Culmen Length (mm)','Culmen Depth (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)','Delta 13 C (o/oo)']].mode(),inplace=True)
data.fillna(data['Delta 13 C (o/oo)'].mode(),inplace=True)
data.dropna(inplace=True)
```

2. Removing outliers based on a specific threshold. Give reasons for your choice of threshold. What do the outliers in your dataset tell you?

```
def plotBoxes(numerical):
    plt.figure(figsize=(10,20))
    cnt = 1
    for column in numerical:
        if(cnt>len(numerical)):
            break
        plt.subplot(3,2,cnt)
        plt.boxplot(numerical[column])
        plt.title(column)
        cnt+=1
```

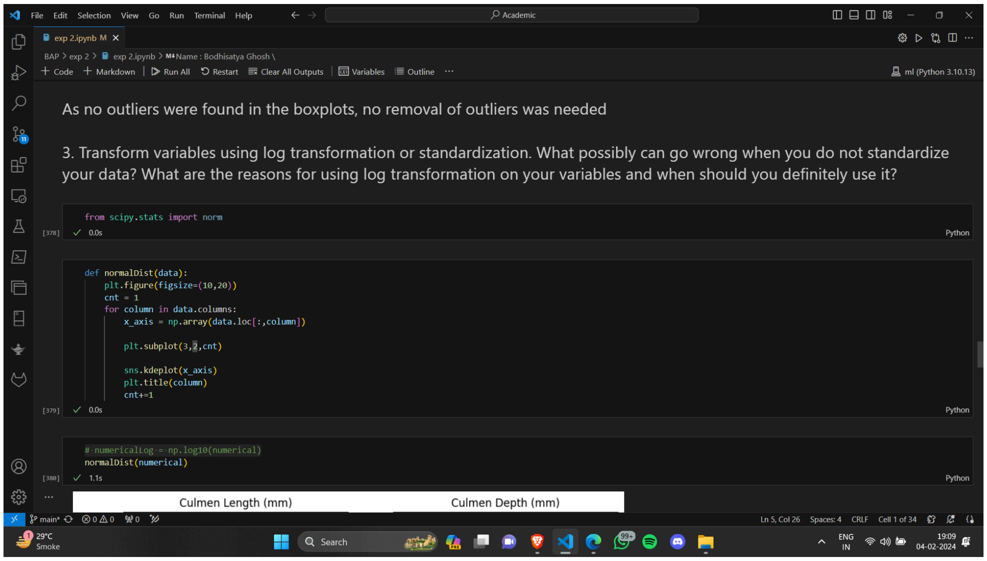
```
numerical = data[['Culmen Length (mm)','Culmen Depth (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)','Delta 13 C (o/oo)']]
```

File Edit Selection View Go Run Terminal Help
Academic
exp 2.ipynb M
BAP > exp 2 > exp 2.ipynb > M↓Name : Bodhisatya Ghosh \
+ Code + Markdown | Run All Restart Clear All Outputs Variables Outline
ml (Python 3.10.13)
[374] 0.0s
[375] 0.0s
[376] 0.0s
Python
main* 0 0 0
Ln 5, Col 26 Spaces: 4 CRLF Cell 1 of 34
29°C Smoke
Search
ENG IN
19:08 04-02-2024



```
plotBoxes(numerical)
```

File Edit Selection View Go Run Terminal Help
Academic
exp 2.ipynb M
BAP > exp 2 > exp 2.ipynb > M↓Name : Bodhisatya Ghosh \
+ Code + Markdown | Run All Restart Clear All Outputs Variables Outline
ml (Python 3.10.13)
[377] 0.6s
Python
Culmen Length (mm)
60
55
50
45
40
35
1
Culmen Depth (mm)
20
18
16
14
1
Flipper Length (mm)
230
Body Mass (g)
main* 0 0 0
Ln 5, Col 26 Spaces: 4 CRLF Cell 1 of 34
29°C Smoke
Search
ENG IN
19:09 04-02-2024



File Edit Selection View Go Run Terminal Help
Academic
exp 2.ipynb M
BAP > exp 2 > exp 2.ipynb > M↓Name : Bodhisatya Ghosh \
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline
ml (Python 3.10.13)

As no outliers were found in the boxplots, no removal of outliers was needed

3. Transform variables using log transformation or standardization. What possibly can go wrong when you do not standardize your data? What are the reasons for using log transformation on your variables and when should you definitely use it?

```
from scipy.stats import norm
```

[378] ✓ 0.0s Python

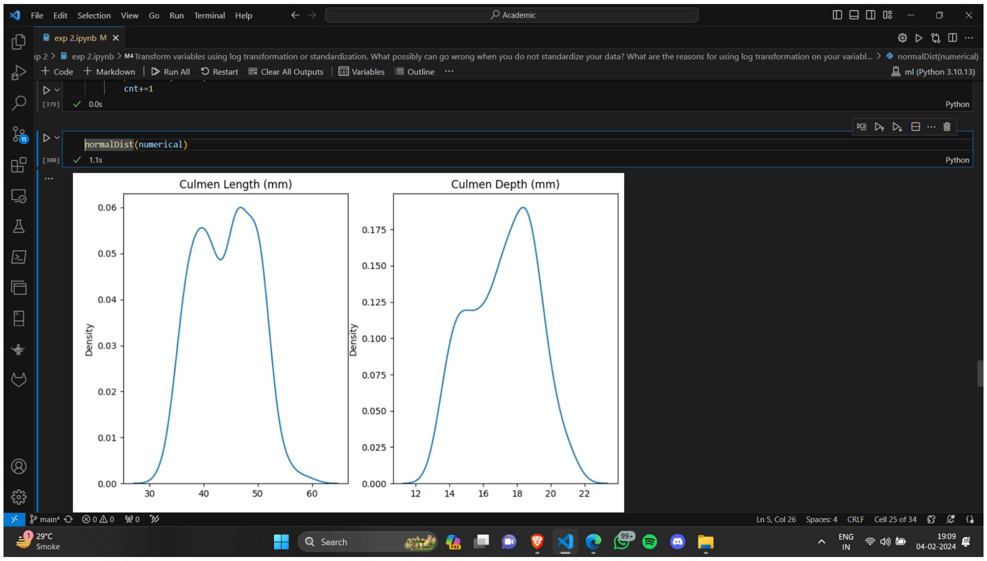
```
def normalDist(data):
    plt.figure(figsize=(10,20))
    cnt = 1
    for column in data.columns:
        x_axis = np.array(data.loc[:,column])

        plt.subplot(3,2,cnt)

        sns.kdeplot(x_axis)
        plt.title(column)
        cnt+=1
```

[379] ✓ 0.0s Python

```
# numericalLog = np.log10(numerical)
normalDist(numerical)
```

[380] ✓ 1.1s Python

Culmen Length (mm)
Culmen Depth (mm)
main* ⊗ 0 ⚠ 0 0
Ln 5, Col 26 Spaces: 4 CRLF Cell 1 of 34
29°C Smoke
Search
ENG IN
19:09 04-02-2024



File Edit Selection View Go Run Terminal Help
Academic
exp 2.ipynb M
xp 2 > exp 2.ipynb > M↓Transform variables using log transformation or standardization. What possibly can go wrong when you do not standardize your data? What are the reasons for using log transformation on your variabl... > normalDist(numerical)
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline
ml (Python 3.10.13)

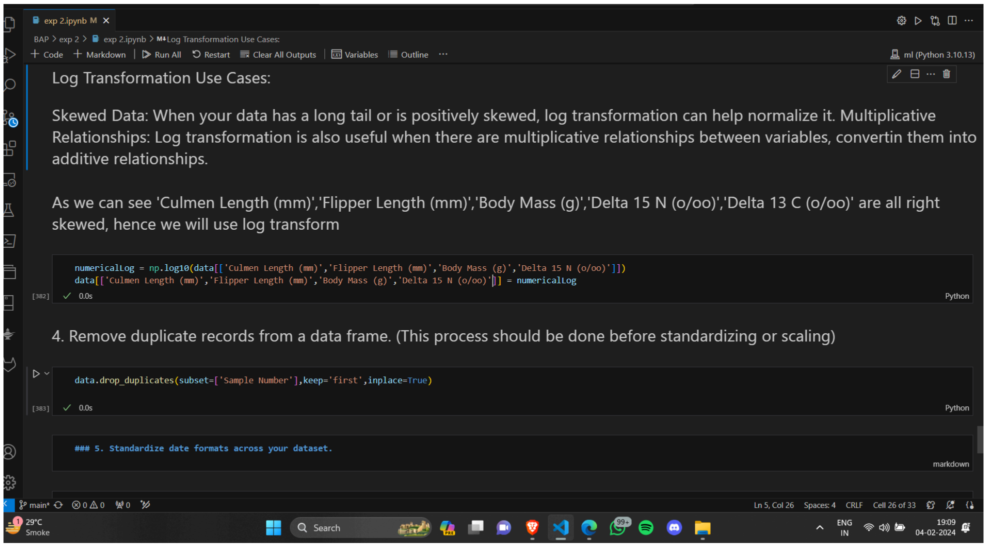
```
        cnt+=1
```

[379] ✓ 0.0s Python

```
normalDist(numerical)
```

[380] ✓ 1.1s Python

Culmen Length (mm)
Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
30
40
50
60
Culmen Depth (mm)
Density
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
12
14
16
18
20
22
main* ⊗ 0 ⚠ 0 0
Ln 5, Col 26 Spaces: 4 CRLF Cell 25 of 34
29°C Smoke
Search
ENG IN
19:09 04-02-2024



exp 2.ipynb M
BAP > exp 2 > exp 2.ipynb > M↓Log Transformation Use Cases:
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline
ml (Python 3.10.13)

Log Transformation Use Cases:

Skewed Data: When your data has a long tail or is positively skewed, log transformation can help normalize it. Multiplicative Relationships: Log transformation is also useful when there are multiplicative relationships between variables, convertin them into additive relationships.

As we can see 'Culmen Length (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)','Delta 13 C (o/oo)' are all right skewed, hence we will use log transform

```
numericalLog = np.log10(data[['Culmen Length (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)']])
data[['Culmen Length (mm)','Flipper Length (mm)','Body Mass (g)','Delta 15 N (o/oo)']] = numericalLog
```

[382] ✓ 0.0s Python

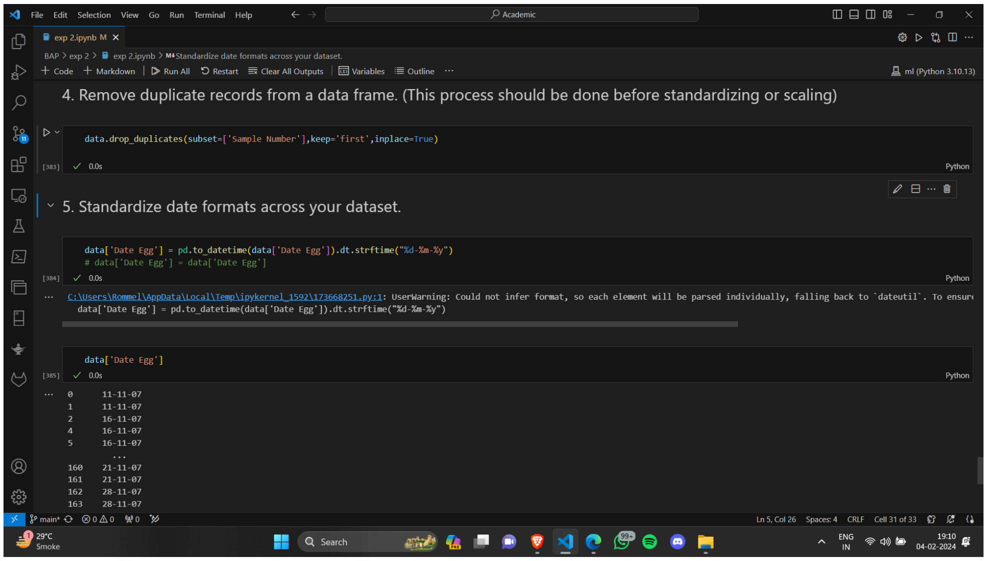
4. Remove duplicate records from a data frame. (This process should be done before standardizing or scaling)

```
data.drop_duplicates(subset=['Sample Number'],keep='first',inplace=True)
```

[383] ✓ 0.0s Python

```
### 5. Standardize date formats across your dataset.
```

markdown


File Edit Selection View Go Run Terminal Help
Academic
exp 2.ipynb M
BAP > exp 2 > exp 2.ipynb > M↓Standardize date formats across your dataset.
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ···
ml (Python 3.10.13)
Ln 5, Col 26 Spaces: 4 CRLF Cell 31 of 33
main* ⊗ 0 ⚠ 0
29°C Smoke
Search
ENG IN
19:10 04-02-2024

4. Remove duplicate records from a data frame. (This process should be done before standardizing or scaling)

```
data.drop_duplicates(subset=['Sample Number'],keep='first',inplace=True)
```

[383] ✓ 0.0s Python

5. Standardize date formats across your dataset.

```
data['Date Egg'] = pd.to_datetime(data['Date Egg']).dt.strftime("%d-%m-%y")
# data['Date Egg'] = data['Date Egg']
```

[384] ✓ 0.0s Python

```
C:\Users\Rommel\AppData\Local\Temp\ipykernel_1592\173668251.py:1: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure
  data['Date Egg'] = pd.to_datetime(data['Date Egg']).dt.strftime("%d-%m-%y")
```

```
data['Date Egg']
```

[385] ✓ 0.0s Python

```
0      11-11-07
1      11-11-07
2      16-11-07
4      16-11-07
5      16-11-07
         ...
160    21-11-07
161    21-11-07
162    28-11-07
163    28-11-07
```

RESULT:

1. From the python notebook, we can see that using mode to fill null values gives us minimum deviation, hence we will be using mode in fillna().
2. From the boxplots, we could see that there are no outliers in the given data, hence we will not remove any data
3. If we do not standardize our data, the numerically larger values may end up dominating the model parameters and hinder in accurate calculations.
   Log transforms are usually used for right skewed data.
4. Duplicates were removed using the Sample number as parameters
5. Dates were standardized to dd-mm-yy format.

References:

[How to change the Pandas datetime format in Python? - GeeksforGeeks]

CONCLUSION: In this experiment, I have learnt how to handle missing data, fill in out of format data, check for outliers and how to deal with them.