DATA CLEANING, ANALYSIS AND VISUALIZATION OF NESTED CSV FILE DATASETS WITH MERGING

A Dissertation Submitted To

BIG LEARN TRAINING INSTITUTE

TIRUCHIRAPPALLI

In Partial Fulfillment Of The Requirement

For The Award Of The Degree Of

DATA SCIENCE

Guided By:

MUTHU KARUPPAN K

By:

UKESH A

INTRODUCTION:

This Python script is designed to analyze and visualize data from two distinct datasets, referred to as "Dataset 1" and "Dataset 2." The primary goal is to explore the statistical characteristics of individual datasets and compare them to the combined dataset obtained through merging. Additionally, the program utilizes popular data manipulation and visualization libraries such as NumPy, Pandas, Seaborn, and Matplotlib.

OBJECTIVE:

The objective of this program seems to be the analysis and comparison of statistical measures (mean and standard deviation) for specific columns in two datasets and their merged form. Additionally, the program performs data visualization using seaborn and matplotlib libraries to create line plots, bar plots, and histograms for the datasets.

DATA SOURCE:

Data Analysis and Visualization.

METHODOLOGY:

- ✓ **Data Loading:**
  - Two datasets (Dataset 1 and Dataset 2) are loaded into Pandas DataFrames using pd.read_csv().
  - data = pd.read_csv('C:\\Users\\User\\Desktop\\Dataset 1.csv').
  - data1 = pd.read_csv('C:\\Users\\User\\Desktop\\Dataset 2.csv').
- ✓ **Descriptive Satistics:**
  - The script calculates the mean and standard deviation for specific columns in each dataset.
  - The calculations are performed separately for each dataset.

```
# Example for Dataset 1:

  column_name = "CD-ID"

  column_mode = data[column_name].mode()

  print(column_mode)
  column_name = "CD86-1"

  column_mean1 = data[column_name].mean()

  print(column_mean1)

# Similar calculations are performed for other columns.
```

✓ **Data Merging:**

- The two datasets are merged into a new DataFrame named data2 using pd.concat().
- data2 = pd.concat([data, data1], ignore_index=True).
- data2.to_csv("C:\\Users\\User\\Desktop\\merged_file.csv", index=False).

✓ **Descriptive Statistics for Merged Data:**

- The script calculates and prints the mean and standard deviation for specific columns in the merged dataset.

```
# Example for Merged Dataset:

  column_name = "CD-ID"

  column_mode = data2[column_name].mode()

  print(column_mode)


  column_name = "CD86-1"

  column_mean19 = data2[column_name].mean()

  print(column_mean19)

# Similar calculations are performed for other columns
```

✓ **Overall Analysis:**

- The script calculates the overall mean and standard deviation values for each dataset and the merged dataset.

```
  fst_dtst = (column_mean1 + column_mean2 + ... + column_mean9)

  snd_dtst = (column_mean10 + ... + column_mean18)

  mrg_dtst = (column_mean19 + ... + column_mean27)
```

# Similar calculations are performed for standard deviations

✓ **Data Visualization:**

- The script uses Seaborn and Matplotlib to create line plots, bar plots, and histograms for the three datasets.

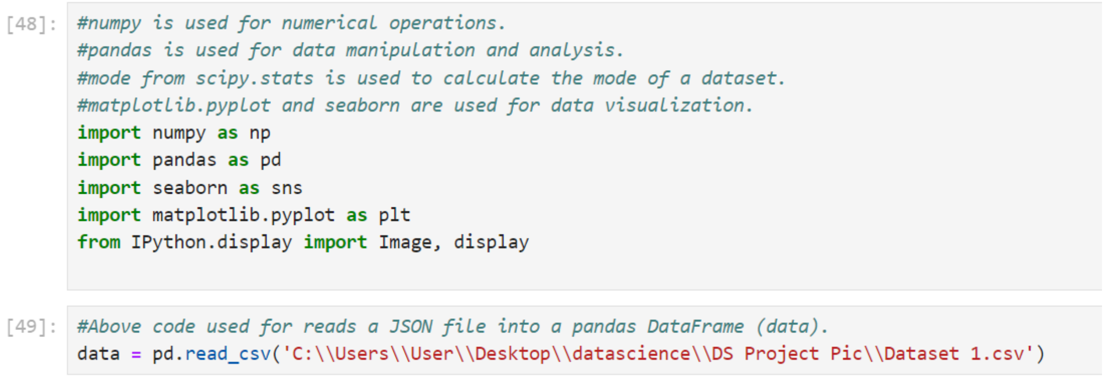
sns.lineplot(data)

plt.show()

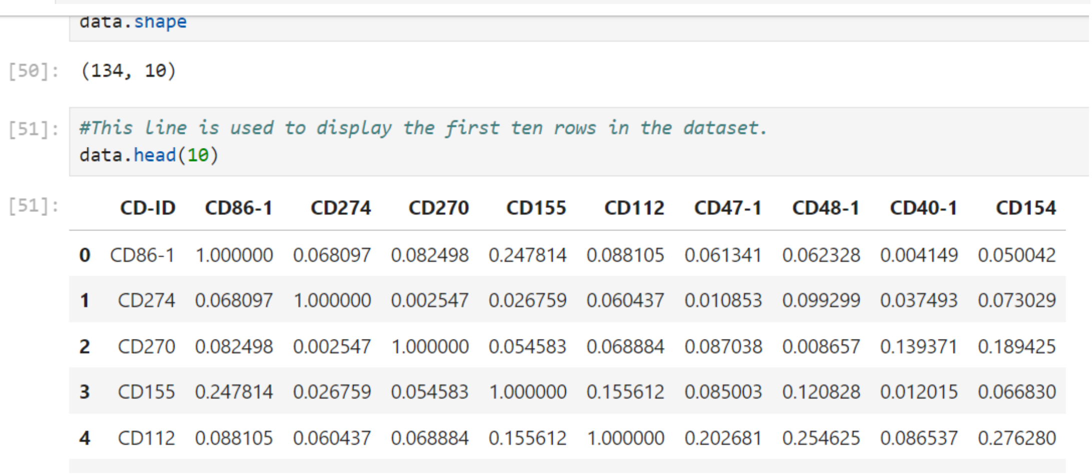
# Similar plots are created for other datasets

The script seems to perform exploratory data analysis, combining statistics and visualization techniques to understand and compare the characteristics of the datasets.

PROJECT CODE:

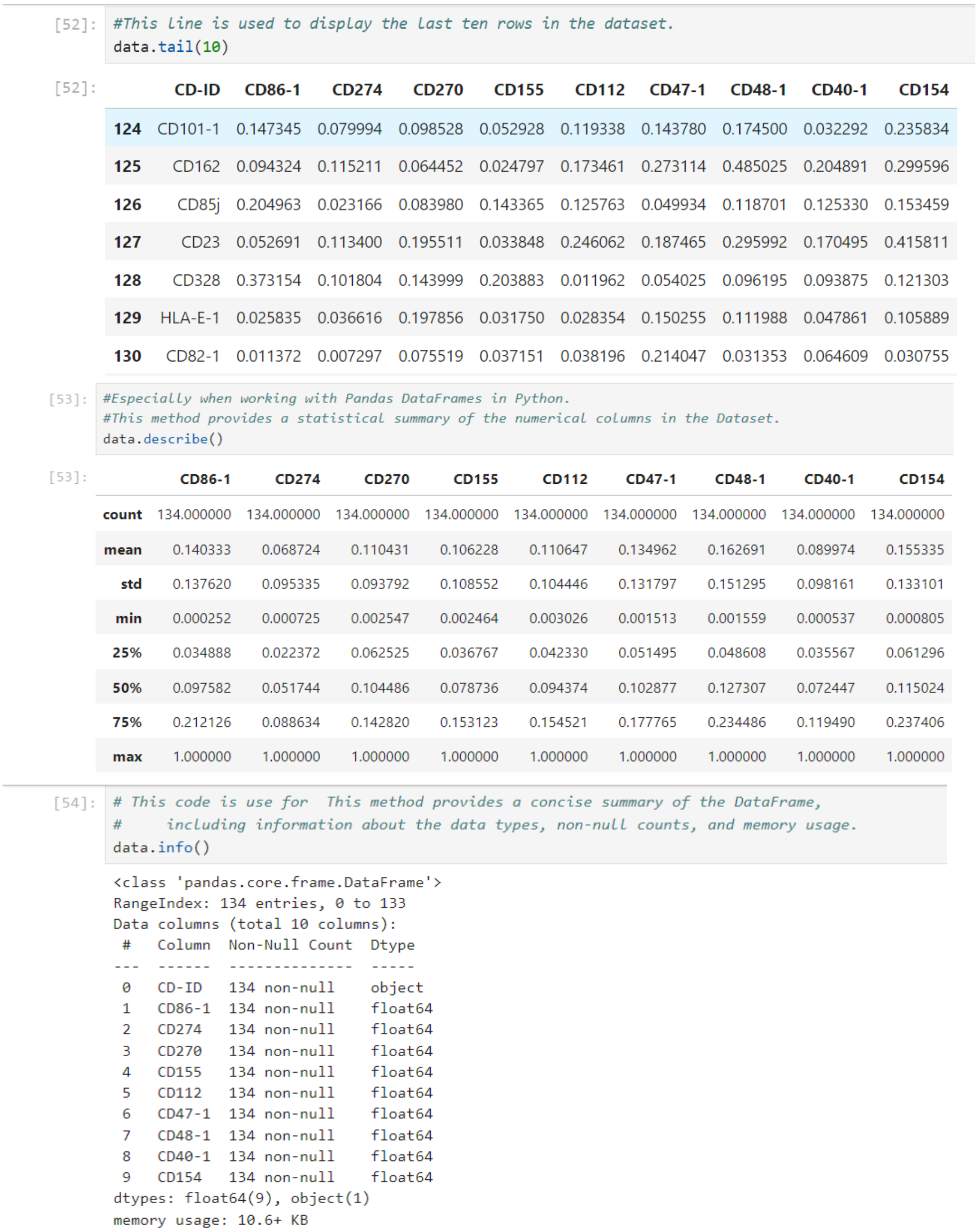


```
[48]: #numpy is used for numerical operations.
      #pandas is used for data manipulation and analysis.
      #mode from scipy.stats is used to calculate the mode of a dataset.
      #matplotlib.pyplot and seaborn are used for data visualization.
      import numpy as np
      import pandas as pd
      import seaborn as sns
      import matplotlib.pyplot as plt
      from IPython.display import Image, display

[49]: #Above code used for reads a JSON file into a pandas DataFrame (data).
      data = pd.read_csv('C:\\Users\\User\\Desktop\\datascience\\DS Project Pic\\Dataset 1.csv')
```



```
      data.shape

[50]: (134, 10)

[51]: #This line is used to display the first ten rows in the dataset.
      data.head(10)

[51]:
```

	CD-ID	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
0	CD86-1	1.000000	0.068097	0.082498	0.247814	0.088105	0.061341	0.062328	0.004149	0.050042
1	CD274	0.068097	1.000000	0.002547	0.026759	0.060437	0.010853	0.099299	0.037493	0.073029
2	CD270	0.082498	0.002547	1.000000	0.054583	0.068884	0.087038	0.008657	0.139371	0.189425
3	CD155	0.247814	0.026759	0.054583	1.000000	0.155612	0.085003	0.120828	0.012015	0.066830
4	CD112	0.088105	0.060437	0.068884	0.155612	1.000000	0.202681	0.254625	0.086537	0.276280



[52]:

```
#This line is used to display the last ten rows in the dataset.
data.tail(10)
```

[52]:

	CD-ID	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
124	CD101-1	0.147345	0.079994	0.098528	0.052928	0.119338	0.143780	0.174500	0.032292	0.235834
125	CD162	0.094324	0.115211	0.064452	0.024797	0.173461	0.273114	0.485025	0.204891	0.299596
126	CD85j	0.204963	0.023166	0.083980	0.143365	0.125763	0.049934	0.118701	0.125330	0.153459
127	CD23	0.052691	0.113400	0.195511	0.033848	0.246062	0.187465	0.295992	0.170495	0.415811
128	CD328	0.373154	0.101804	0.143999	0.203883	0.011962	0.054025	0.096195	0.093875	0.121303
129	HLA-E-1	0.025835	0.036616	0.197856	0.031750	0.028354	0.150255	0.111988	0.047861	0.105889
130	CD82-1	0.011372	0.007297	0.075519	0.037151	0.038196	0.214047	0.031353	0.064609	0.030755

[53]:

```
#Especially when working with Pandas DataFrames in Python.
#This method provides a statistical summary of the numerical columns in the Dataset.
data.describe()
```

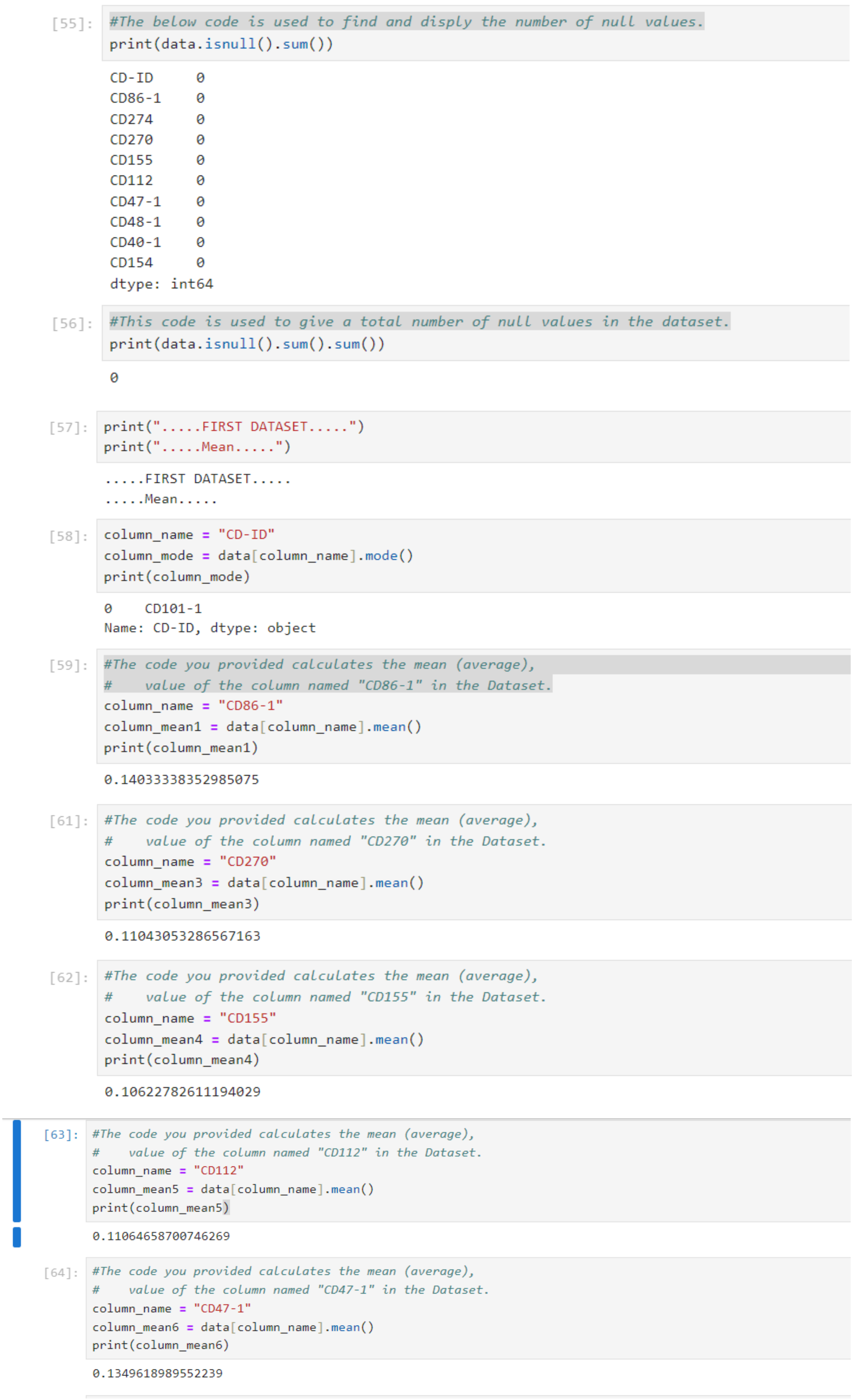
[53]:

	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
count	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000
mean	0.140333	0.068724	0.110431	0.106228	0.110647	0.134962	0.162691	0.089974	0.155335
std	0.137620	0.095335	0.093792	0.108552	0.104446	0.131797	0.151295	0.098161	0.133101
min	0.000252	0.000725	0.002547	0.002464	0.003026	0.001513	0.001559	0.000537	0.000805
25%	0.034888	0.022372	0.062525	0.036767	0.042330	0.051495	0.048608	0.035567	0.061296
50%	0.097582	0.051744	0.104486	0.078736	0.094374	0.102877	0.127307	0.072447	0.115024
75%	0.212126	0.088634	0.142820	0.153123	0.154521	0.177765	0.234486	0.119490	0.237406
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

[54]:

```
# This code is use for  This method provides a concise summary of the DataFrame,
#     including information about the data types, non-null counts, and memory usage.
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134 entries, 0 to 133
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   CD-ID   134 non-null    object 
 1   CD86-1  134 non-null    float64
 2   CD274   134 non-null    float64
 3   CD270   134 non-null    float64
 4   CD155   134 non-null    float64
 5   CD112   134 non-null    float64
 6   CD47-1  134 non-null    float64
 7   CD48-1  134 non-null    float64
 8   CD40-1  134 non-null    float64
 9   CD154   134 non-null    float64
dtypes: float64(9), object(1)
memory usage: 10.6+ KB
```



```
[55]: #The below code is used to find and disply the number of null values.
      print(data.isnull().sum())
```

```
CD-ID     0
CD86-1    0
CD274     0
CD270     0
CD155     0
CD112     0
CD47-1    0
CD48-1    0
CD40-1    0
CD154     0
dtype: int64
```

```
[56]: #This code is used to give a total number of null values in the dataset.
      print(data.isnull().sum().sum())
```

```
0
```

```
[57]: print(".....FIRST DATASET.....")
      print(".....Mean.....")
```

```
.....FIRST DATASET.....
.....Mean.....
```

```
[58]: column_name = "CD-ID"
      column_mode = data[column_name].mode()
      print(column_mode)
```

```
0    CD101-1
Name: CD-ID, dtype: object
```

```
[59]: #The code you provided calculates the mean (average),
      #    value of the column named "CD86-1" in the Dataset.
      column_name = "CD86-1"
      column_mean1 = data[column_name].mean()
      print(column_mean1)
```

```
0.14033338352985075
```

```
[61]: #The code you provided calculates the mean (average),
      #    value of the column named "CD270" in the Dataset.
      column_name = "CD270"
      column_mean3 = data[column_name].mean()
      print(column_mean3)
```

```
0.11043053286567163
```

```
[62]: #The code you provided calculates the mean (average),
      #    value of the column named "CD155" in the Dataset.
      column_name = "CD155"
      column_mean4 = data[column_name].mean()
      print(column_mean4)
```

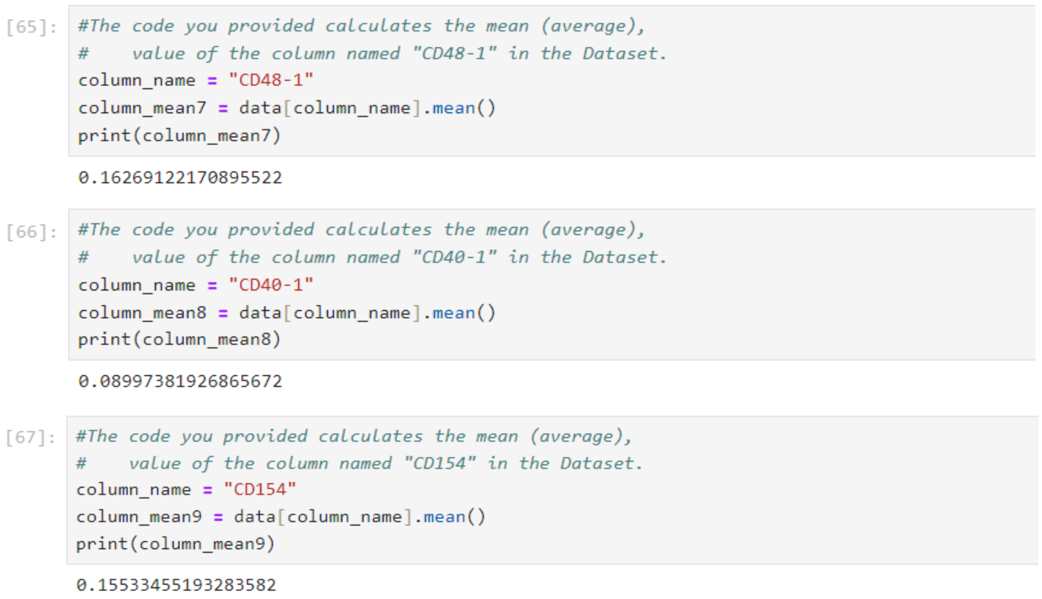
```
0.10622782611194029
```

```
[63]: #The code you provided calculates the mean (average),
      #    value of the column named "CD112" in the Dataset.
      column_name = "CD112"
      column_mean5 = data[column_name].mean()
      print(column_mean5)
```

```
0.11064658700746269
```

```
[64]: #The code you provided calculates the mean (average),
      #    value of the column named "CD47-1" in the Dataset.
      column_name = "CD47-1"
      column_mean6 = data[column_name].mean()
      print(column_mean6)
```

```
0.1349618989552239
```



[65]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD48-1" in the Dataset.
column_name = "CD48-1"
column_mean7 = data[column_name].mean()
print(column_mean7)
```

```
0.16269122170895522
```

[66]:

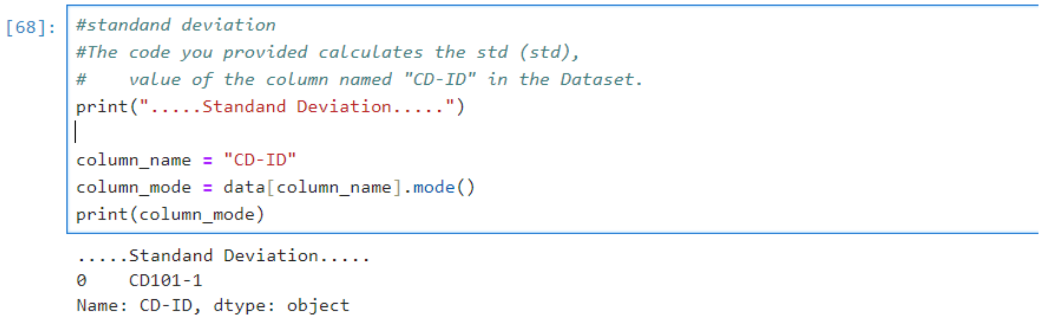
```
#The code you provided calculates the mean (average),
#    value of the column named "CD40-1" in the Dataset.
column_name = "CD40-1"
column_mean8 = data[column_name].mean()
print(column_mean8)
```

```
0.08997381926865672
```

[67]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD154" in the Dataset.
column_name = "CD154"
column_mean9 = data[column_name].mean()
print(column_mean9)
```

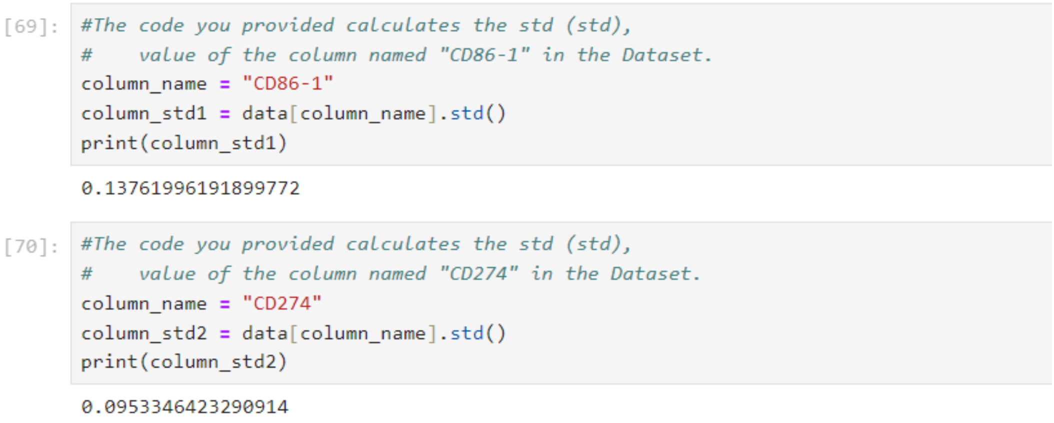
```
0.15533455193283582
```



[68]:

```
#standand deviation
#The code you provided calculates the std (std),
#    value of the column named "CD-ID" in the Dataset.
print(".....Standand Deviation.....")

column_name = "CD-ID"
column_mode = data[column_name].mode()
print(column_mode)
```

```
.....Standand Deviation.....
0    CD101-1
Name: CD-ID, dtype: object
```



[69]:

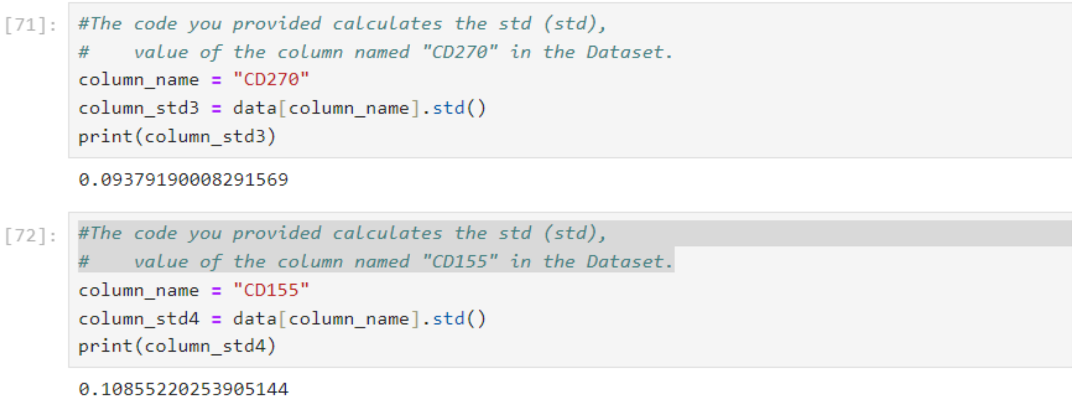
```
#The code you provided calculates the std (std),
#    value of the column named "CD86-1" in the Dataset.
column_name = "CD86-1"
column_std1 = data[column_name].std()
print(column_std1)
```

```
0.13761996191899772
```

[70]:

```
#The code you provided calculates the std (std),
#    value of the column named "CD274" in the Dataset.
column_name = "CD274"
column_std2 = data[column_name].std()
print(column_std2)
```

```
0.0953346423290914
```



[71]:

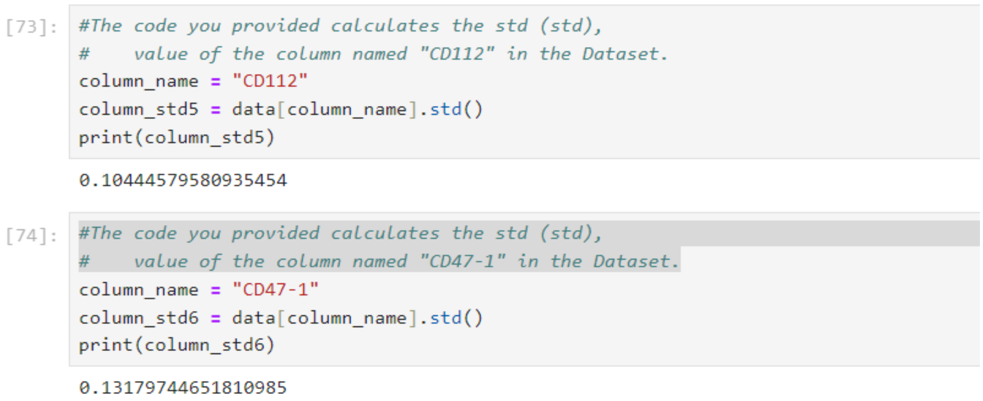
```
#The code you provided calculates the std (std),
#    value of the column named "CD270" in the Dataset.
column_name = "CD270"
column_std3 = data[column_name].std()
print(column_std3)
```

```
0.09379190008291569
```

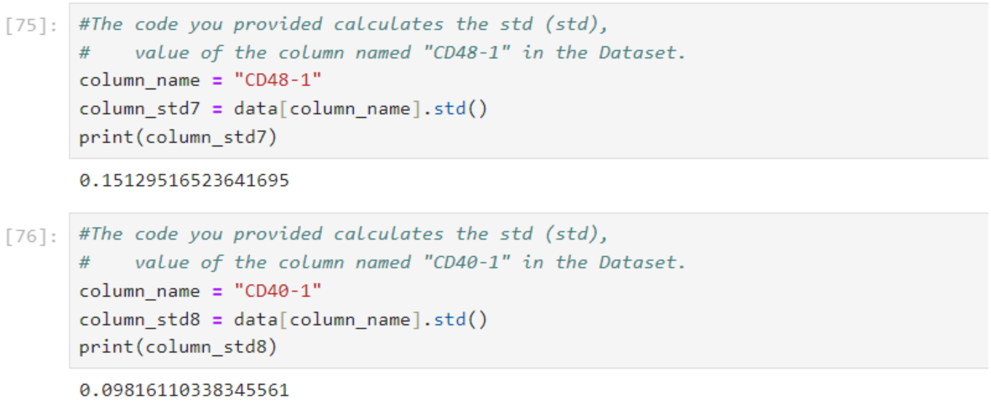
[72]:

```
#The code you provided calculates the std (std),
#    value of the column named "CD155" in the Dataset.
column_name = "CD155"
column_std4 = data[column_name].std()
print(column_std4)
```

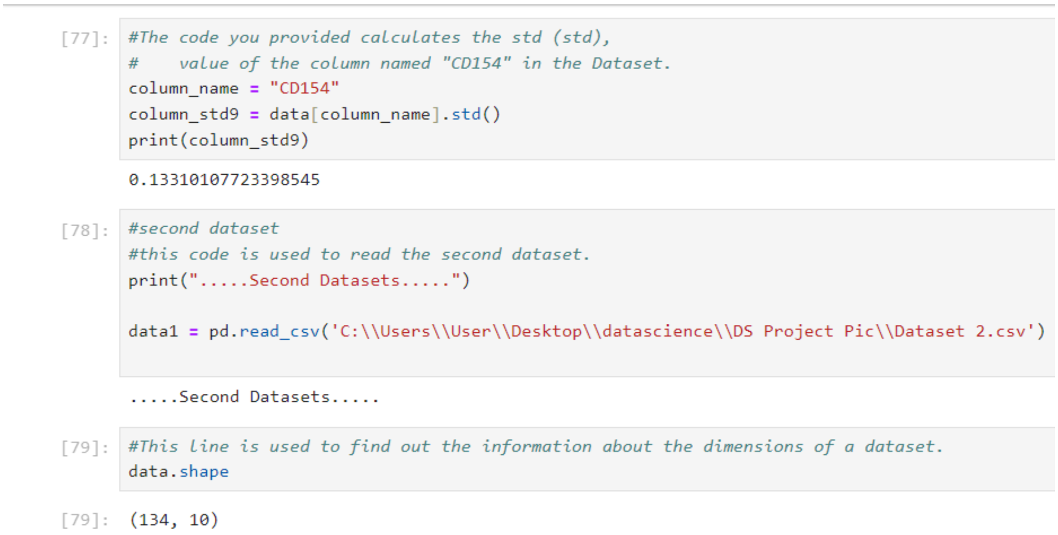
```
0.10855220253905144
```



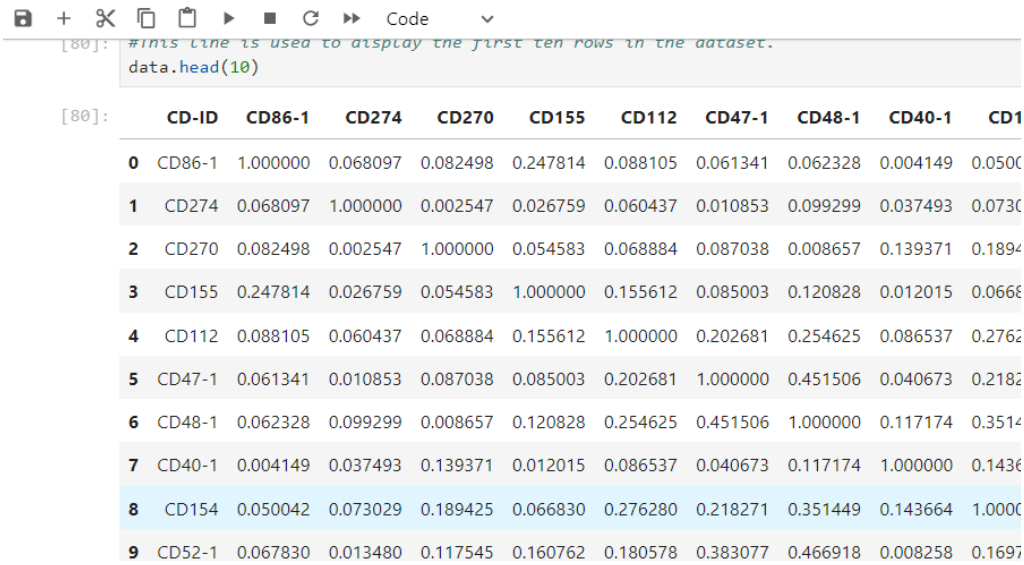
```
[73]: #The code you provided calculates the std (std),
      #    value of the column named "CD112" in the Dataset.
      column_name = "CD112"
      column_std5 = data[column_name].std()
      print(column_std5)

      0.10444579580935454

[74]: #The code you provided calculates the std (std),
      #    value of the column named "CD47-1" in the Dataset.
      column_name = "CD47-1"
      column_std6 = data[column_name].std()
      print(column_std6)

      0.13179744651810985
```



```
[75]: #The code you provided calculates the std (std),
      #    value of the column named "CD48-1" in the Dataset.
      column_name = "CD48-1"
      column_std7 = data[column_name].std()
      print(column_std7)

      0.15129516523641695

[76]: #The code you provided calculates the std (std),
      #    value of the column named "CD40-1" in the Dataset.
      column_name = "CD40-1"
      column_std8 = data[column_name].std()
      print(column_std8)

      0.09816110338345561
```



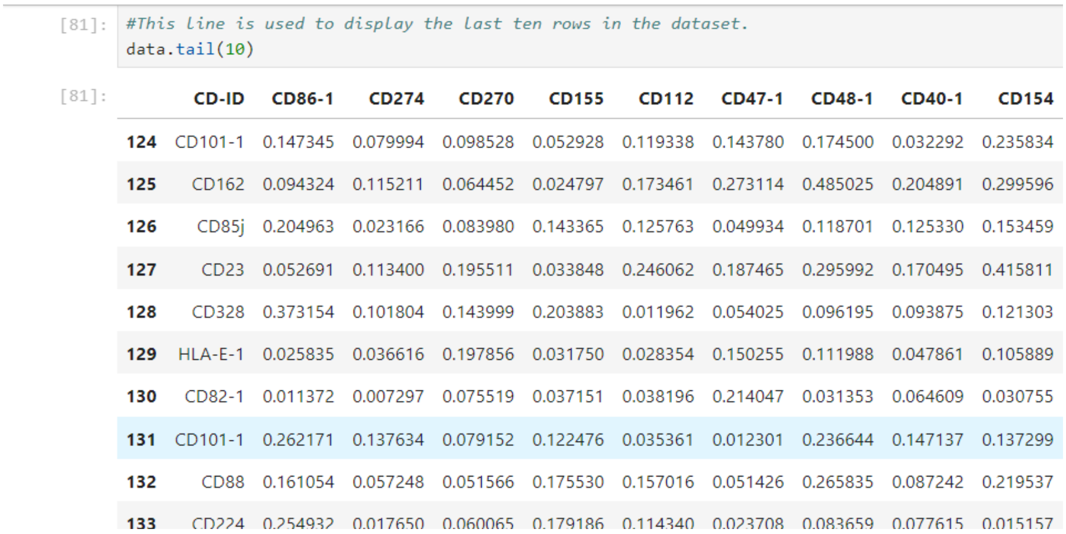
```
[77]: #The code you provided calculates the std (std),
      #    value of the column named "CD154" in the Dataset.
      column_name = "CD154"
      column_std9 = data[column_name].std()
      print(column_std9)

      0.13310107723398545

[78]: #second dataset
      #this code is used to read the second dataset.
      print(".....Second Datasets.....")

      data1 = pd.read_csv('C:\\Users\\User\\Desktop\\datascience\\DS Project Pic\\Dataset 2.csv')

      .....Second Datasets.....

[79]: #This line is used to find out the information about the dimensions of a dataset.
      data.shape

[79]: (134, 10)
```



```
[80]: #This line is used to display the first ten rows in the dataset.
      data.head(10)
```

[80]:

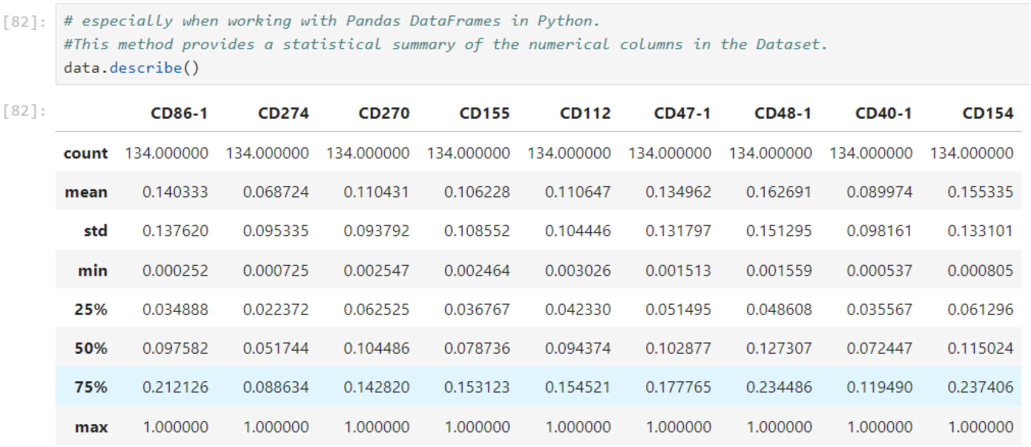
	CD-ID	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD1
0	CD86-1	1.000000	0.068097	0.082498	0.247814	0.088105	0.061341	0.062328	0.004149	0.0500
1	CD274	0.068097	1.000000	0.002547	0.026759	0.060437	0.010853	0.099299	0.037493	0.0730
2	CD270	0.082498	0.002547	1.000000	0.054583	0.068884	0.087038	0.008657	0.139371	0.1894
3	CD155	0.247814	0.026759	0.054583	1.000000	0.155612	0.085003	0.120828	0.012015	0.0668
4	CD112	0.088105	0.060437	0.068884	0.155612	1.000000	0.202681	0.254625	0.086537	0.2762
5	CD47-1	0.061341	0.010853	0.087038	0.085003	0.202681	1.000000	0.451506	0.040673	0.2182
6	CD48-1	0.062328	0.099299	0.008657	0.120828	0.254625	0.451506	1.000000	0.117174	0.3514
7	CD40-1	0.004149	0.037493	0.139371	0.012015	0.086537	0.040673	0.117174	1.000000	0.1436
8	CD154	0.050042	0.073029	0.189425	0.066830	0.276280	0.218271	0.351449	0.143664	1.0000
9	CD52-1	0.067830	0.013480	0.117545	0.160762	0.180578	0.383077	0.466918	0.008258	0.1697



```
[81]: #This line is used to display the last ten rows in the dataset.
      data.tail(10)
```

[81]:

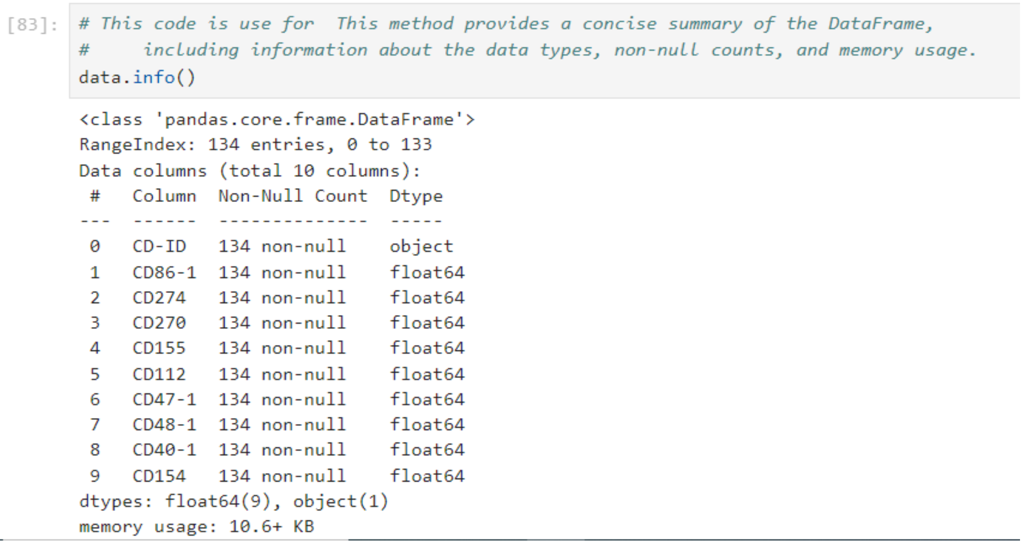
	CD-ID	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
124	CD101-1	0.147345	0.079994	0.098528	0.052928	0.119338	0.143780	0.174500	0.032292	0.235834
125	CD162	0.094324	0.115211	0.064452	0.024797	0.173461	0.273114	0.485025	0.204891	0.299596
126	CD85j	0.204963	0.023166	0.083980	0.143365	0.125763	0.049934	0.118701	0.125330	0.153459
127	CD23	0.052691	0.113400	0.195511	0.033848	0.246062	0.187465	0.295992	0.170495	0.415811
128	CD328	0.373154	0.101804	0.143999	0.203883	0.011962	0.054025	0.096195	0.093875	0.121303
129	HLA-E-1	0.025835	0.036616	0.197856	0.031750	0.028354	0.150255	0.111988	0.047861	0.105889
130	CD82-1	0.011372	0.007297	0.075519	0.037151	0.038196	0.214047	0.031353	0.064609	0.030755
131	CD101-1	0.262171	0.137634	0.079152	0.122476	0.035361	0.012301	0.236644	0.147137	0.137299
132	CD88	0.161054	0.057248	0.051566	0.175530	0.157016	0.051426	0.265835	0.087242	0.219537
133	CD224	0.254932	0.017650	0.060065	0.179186	0.114340	0.023708	0.083659	0.077615	0.015157



```
[82]: # especially when working with Pandas DataFrames in Python.
      #This method provides a statistical summary of the numerical columns in the Dataset.
      data.describe()
```

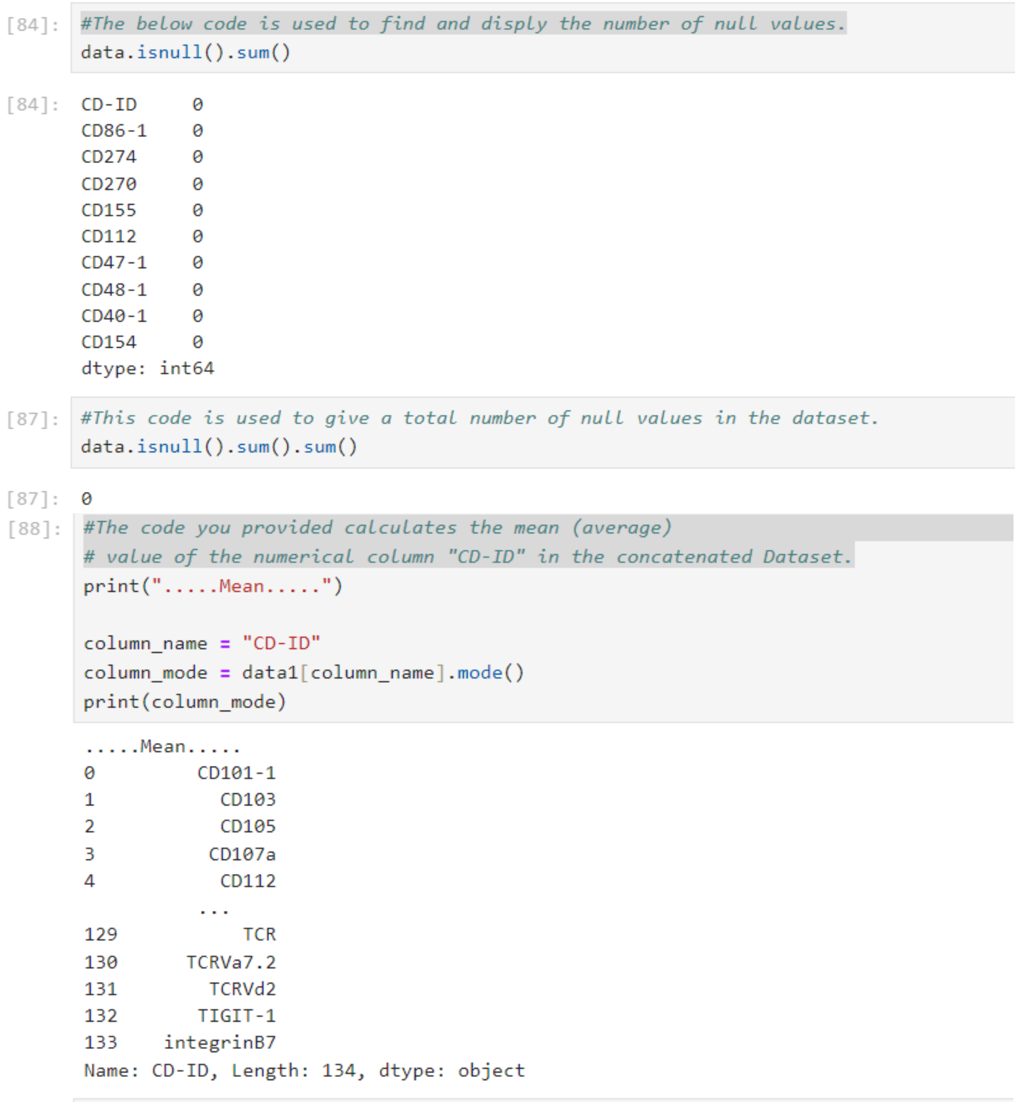
[82]:

	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
count	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000
mean	0.140333	0.068724	0.110431	0.106228	0.110647	0.134962	0.162691	0.089974	0.155335
std	0.137620	0.095335	0.093792	0.108552	0.104446	0.131797	0.151295	0.098161	0.133101
min	0.000252	0.000725	0.002547	0.002464	0.003026	0.001513	0.001559	0.000537	0.000805
25%	0.034888	0.022372	0.062525	0.036767	0.042330	0.051495	0.048608	0.035567	0.061296
50%	0.097582	0.051744	0.104486	0.078736	0.094374	0.102877	0.127307	0.072447	0.115024
75%	0.212126	0.088634	0.142820	0.153123	0.154521	0.177765	0.234486	0.119490	0.237406
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000



```
[83]: # This code is use for  This method provides a concise summary of the DataFrame,
      #     including information about the data types, non-null counts, and memory usage.
      data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134 entries, 0 to 133
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   CD-ID   134 non-null    object
 1   CD86-1  134 non-null    float64
 2   CD274   134 non-null    float64
 3   CD270   134 non-null    float64
 4   CD155   134 non-null    float64
 5   CD112   134 non-null    float64
 6   CD47-1  134 non-null    float64
 7   CD48-1  134 non-null    float64
 8   CD40-1  134 non-null    float64
 9   CD154   134 non-null    float64
dtypes: float64(9), object(1)
memory usage: 10.6+ KB
```



```
[84]: #The below code is used to find and disply the number of null values.
      data.isnull().sum()
```

```
[84]: CD-ID     0
      CD86-1    0
      CD274     0
      CD270     0
      CD155     0
      CD112     0
      CD47-1    0
      CD48-1    0
      CD40-1    0
      CD154     0
      dtype: int64
```

```
[87]: #This code is used to give a total number of null values in the dataset.
      data.isnull().sum().sum()
```

```
[87]: 0
```

```
[88]: #The code you provided calculates the mean (average)
      # value of the numerical column "CD-ID" in the concatenated Dataset.
      print(".....Mean.....")

      column_name = "CD-ID"
      column_mode = data1[column_name].mode()
      print(column_mode)
```

```
.....Mean.....
0          CD101-1
1            CD103
2            CD105
3           CD107a
4            CD112
          ...
129            TCR
130       TCRVa7.2
131         TCRVd2
132        TIGIT-1
133    integrinB7
Name: CD-ID, Length: 134, dtype: object
```

[89]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD86-1" in the concatenated Dataset.
column_name = "CD86-1"
column_mean10 = data1[column_name].mean()
print(column_mean10)
```

```
0.21335994158208957
```

[90]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD274" in the concatenated Dataset.
column_name = "CD274"
column_mean11 = data1[column_name].mean()
print(column_mean11)
```

```
0.08302227674626865
```

[91]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD270" in the concatenated Dataset.
column_name = "CD270"
column_mean12 = data1[column_name].mean()
print(column_mean12)
```

```
0.15032986518656716
```

[92]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD155" in the concatenated Dataset.
column_name = "CD155"
column_mean13 = data1[column_name].mean()
print(column_mean13)
```

```
0.15494947644029852
```

[93]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD112" in the concatenated Dataset.
column_name = "CD112"
column_mean14 = data1[column_name].mean()
print(column_mean14)
```

```
0.13282018396268658
```

[94]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD47-1" in the concatenated Dataset.
column_name = "CD47-1"
column_mean15 = data1[column_name].mean()
print(column_mean15)
```

```
0.1992478441492537
```

[95]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD48-1" in the concatenated Dataset.
column_name = "CD48-1"
column_mean16 = data1[column_name].mean()
print(column_mean16)
```

```
0.24271059612686569
```

[96]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD40-1" in the concatenated Dataset.
column_name = "CD40-1"
column_mean17 = data1[column_name].mean()
print(column_mean17)
```

```
0.15718121084328357
```

[97]:

```
#The code you provided calculates the mean (average)
# value of the numerical column "CD154" in the concatenated Dataset.
column_name = "CD154"
column_mean18 = data1[column_name].mean()
print(column_mean18)
```

```
0.19984196713432834
```

[98]:

```
#standand deviation
#The code you provided calculates the std (std),
#    value of the column named "CD-ID" in the Dataset.
print(".....Standand Deviation.....")

column_name = "CD-ID"
column_mode = data1[column_name].mode()
print(column_mode)
```

```
.....Standand Deviation.....
0         CD101-1
1           CD103
2           CD105
3          CD107a
4           CD112
          ...
129           TCR
130      TCRVa7.2
131        TCRVd2
132       TIGIT-1
133     integrinB7
Name: CD-ID, Length: 134, dtype: object
```

[99]:

```
#The code you provided calculates the std (std),
#    value of the column named "CD86-1" in the Dataset.
column_name = "CD86-1"
column_std10 = data1[column_name].std()
print(column_std10)
```

```
0.17010699556283507
```

[100]:

```
#The code you provided calculates the std (std),
#    value of the column named "CD274" in the Dataset.
column_name = "CD274"
column_std11 = data1[column_name].std()
print(column_std11)
```

```
0.10317798790512335
```

[101]:

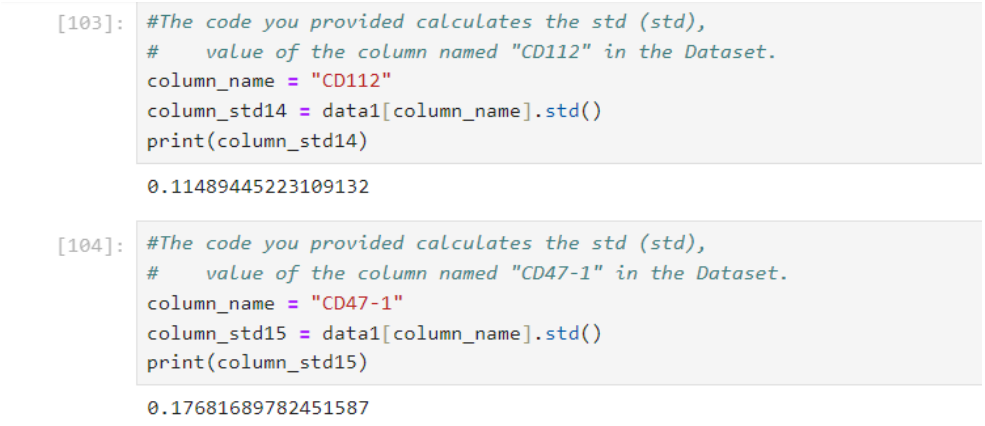
```
#The code you provided calculates the std (std),
#    value of the column named "CD270" in the Dataset.
column_name = "CD270"
column_std12 = data1[column_name].std()
print(column_std12)
```

```
0.1086895614600121
```

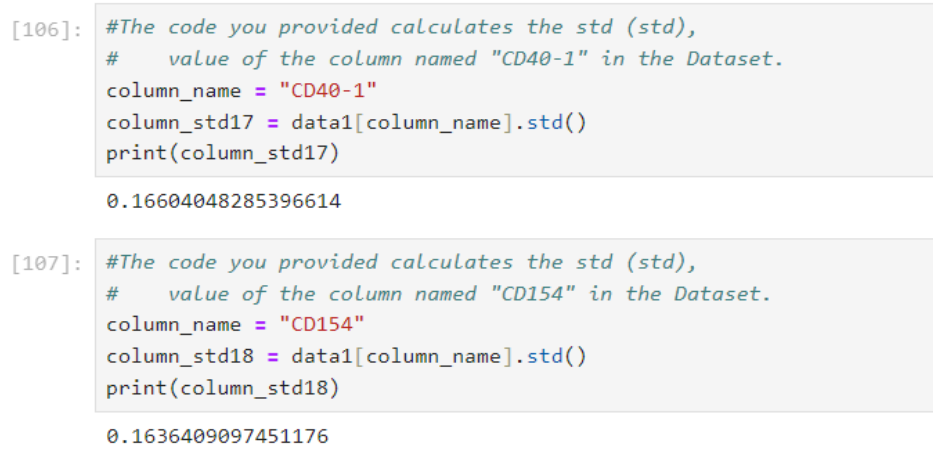
[102]:

```
#The code you provided calculates the std (std),
#    value of the column named "CD155" in the Dataset.
column_name = "CD155"
column_std13 = data1[column_name].std()
print(column_std13)
```

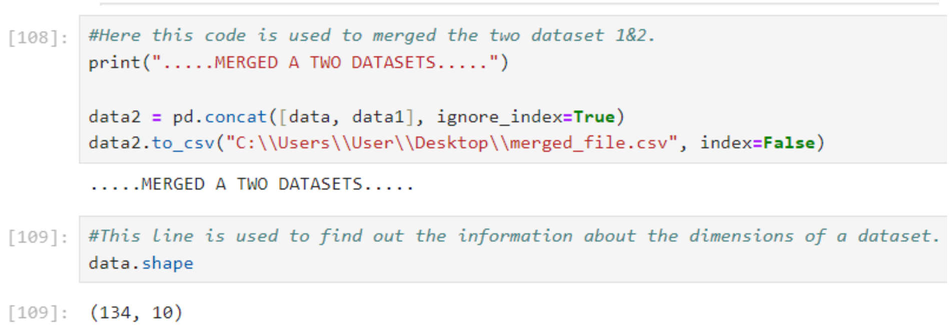
```
0.12832355711367385
```



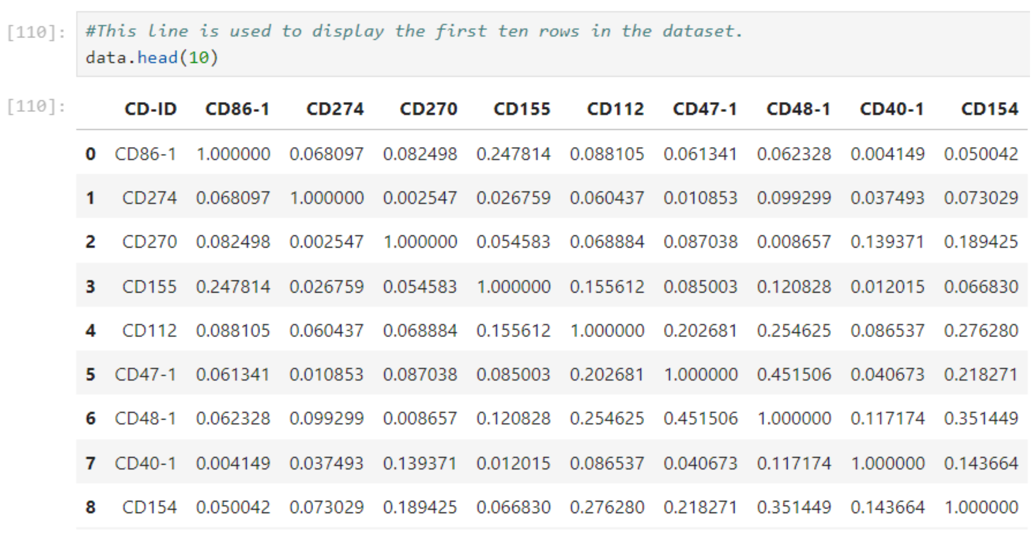
```
[103]: #The code you provided calculates the std (std),
       #    value of the column named "CD112" in the Dataset.
       column_name = "CD112"
       column_std14 = data1[column_name].std()
       print(column_std14)

       0.11489445223109132

[104]: #The code you provided calculates the std (std),
       #    value of the column named "CD47-1" in the Dataset.
       column_name = "CD47-1"
       column_std15 = data1[column_name].std()
       print(column_std15)

       0.17681689782451587
```



```
[106]: #The code you provided calculates the std (std),
       #    value of the column named "CD40-1" in the Dataset.
       column_name = "CD40-1"
       column_std17 = data1[column_name].std()
       print(column_std17)

       0.16604048285396614

[107]: #The code you provided calculates the std (std),
       #    value of the column named "CD154" in the Dataset.
       column_name = "CD154"
       column_std18 = data1[column_name].std()
       print(column_std18)

       0.1636409097451176
```



```
[108]: #Here this code is used to merged the two dataset 1&2.
       print(".....MERGED A TWO DATASETS.....")

       data2 = pd.concat([data, data1], ignore_index=True)
       data2.to_csv("C:\\Users\\User\\Desktop\\merged_file.csv", index=False)

       .....MERGED A TWO DATASETS.....

[109]: #This line is used to find out the information about the dimensions of a dataset.
       data.shape

[109]: (134, 10)
```



```
[110]: #This line is used to display the first ten rows in the dataset.
       data.head(10)

[110]:
```

	CD-ID	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
0	CD86-1	1.000000	0.068097	0.082498	0.247814	0.088105	0.061341	0.062328	0.004149	0.050042
1	CD274	0.068097	1.000000	0.002547	0.026759	0.060437	0.010853	0.099299	0.037493	0.073029
2	CD270	0.082498	0.002547	1.000000	0.054583	0.068884	0.087038	0.008657	0.139371	0.189425
3	CD155	0.247814	0.026759	0.054583	1.000000	0.155612	0.085003	0.120828	0.012015	0.066830
4	CD112	0.088105	0.060437	0.068884	0.155612	1.000000	0.202681	0.254625	0.086537	0.276280
5	CD47-1	0.061341	0.010853	0.087038	0.085003	0.202681	1.000000	0.451506	0.040673	0.218271
6	CD48-1	0.062328	0.099299	0.008657	0.120828	0.254625	0.451506	1.000000	0.117174	0.351449
7	CD40-1	0.004149	0.037493	0.139371	0.012015	0.086537	0.040673	0.117174	1.000000	0.143664
8	CD154	0.050042	0.073029	0.189425	0.066830	0.276280	0.218271	0.351449	0.143664	1.000000

[111]:

```
#This line is used to display the last ten rows in the dataset.
data.tail(10)
```

[111]:

	CD-ID	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
124	CD101-1	0.147345	0.079994	0.098528	0.052928	0.119338	0.143780	0.174500	0.032292	0.235834
125	CD162	0.094324	0.115211	0.064452	0.024797	0.173461	0.273114	0.485025	0.204891	0.299596
126	CD85j	0.204963	0.023166	0.083980	0.143365	0.125763	0.049934	0.118701	0.125330	0.153459
127	CD23	0.052691	0.113400	0.195511	0.033848	0.246062	0.187465	0.295992	0.170495	0.415811
128	CD328	0.373154	0.101804	0.143999	0.203883	0.011962	0.054025	0.096195	0.093875	0.121303
129	HLA-E-1	0.025835	0.036616	0.197856	0.031750	0.028354	0.150255	0.111988	0.047861	0.105889
130	CD82-1	0.011372	0.007297	0.075519	0.037151	0.038196	0.214047	0.031353	0.064609	0.030755
131	CD101-1	0.262171	0.137634	0.079152	0.122476	0.035361	0.012301	0.236644	0.147137	0.137299
132	CD88	0.161054	0.057248	0.051566	0.175530	0.157016	0.051426	0.265835	0.087242	0.219537

[112]:

```
# especially when working with Pandas DataFrames in Python.
#This method provides a statistical summary of the numerical columns in the Dataset.
data.describe()
```

[112]:

	CD86-1	CD274	CD270	CD155	CD112	CD47-1	CD48-1	CD40-1	CD154
count	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000	134.000000
mean	0.140333	0.068724	0.110431	0.106228	0.110647	0.134962	0.162691	0.089974	0.155335
std	0.137620	0.095335	0.093792	0.108552	0.104446	0.131797	0.151295	0.098161	0.133101
min	0.000252	0.000725	0.002547	0.002464	0.003026	0.001513	0.001559	0.000537	0.000805
25%	0.034888	0.022372	0.062525	0.036767	0.042330	0.051495	0.048608	0.035567	0.061296
50%	0.097582	0.051744	0.104486	0.078736	0.094374	0.102877	0.127307	0.072447	0.115024
75%	0.212126	0.088634	0.142820	0.153123	0.154521	0.177765	0.234486	0.119490	0.237406
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

[113]:

```
# This code is use for  This method provides a concise summary of the DataFrame,
#     including information about the data types, non-null counts, and memory usage.
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134 entries, 0 to 133
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CD-ID    134 non-null    object 
 1   CD86-1   134 non-null    float64
 2   CD274    134 non-null    float64
 3   CD270    134 non-null    float64
 4   CD155    134 non-null    float64
 5   CD112    134 non-null    float64
 6   CD47-1   134 non-null    float64
 7   CD48-1   134 non-null    float64
 8   CD40-1   134 non-null    float64
 9   CD154    134 non-null    float64
dtypes: float64(9), object(1)
memory usage: 10.6+ KB
```

[114]:

```
#The below code is used to find and disply the number of null values.
data.isnull().sum()
```

[114]:

```
CD-ID     0
CD86-1    0
CD274     0
CD270     0
CD155     0
CD112     0
CD47-1    0
CD48-1    0
CD40-1    0
CD154     0
dtype: int64
```

[115]:

```
#This code is used to give a total number of null values in the dataset.
data.isnull().sum().sum()
```

[115]:

```
0
```

[116]:

```

print(".....Mean.....")

column_name = "CD-ID"
column_mode = data2[column_name].mode()
print(column_mode)
```

```
.....Mean.....
0    CD101-1
Name: CD-ID, dtype: object
```

[117]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD86-1" in the Dataset.
column_name = "CD86-1"
column_mean19 = data2[column_name].mean()
print(column_mean19)
```

```
0.17684666255597015
```

[118]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD274" in the Dataset.
column_name = "CD274"
column_mean20 = data2[column_name].mean()
print(column_mean20)
```

```
0.07587295282462685
```

[119]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD270" in the Dataset.
column_name = "CD270"
column_mean21 = data2[column_name].mean()
print(column_mean21)
```

```
0.1303801990261194
```

[120]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD155" in the Dataset.
column_name = "CD155"
column_mean22 = data2[column_name].mean()
print(column_mean22)
```

```
0.1305886512761194
```

[121]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD112" in the Dataset.
column_name = "CD112"
column_mean23 = data2[column_name].mean()
print(column_mean23)
```

```
0.12173338548507462
```

[122]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD47-1" in the Dataset.
column_name = "CD47-1"
column_mean24 = data2[column_name].mean()
print(column_mean24)
```

```
0.16710487155223883
```

[123]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD48-1" in the Dataset.
column_name = "CD48-1"
column_mean25 = data2[column_name].mean()
print(column_mean25)
```

```
0.20270090891791046
```

[124]:

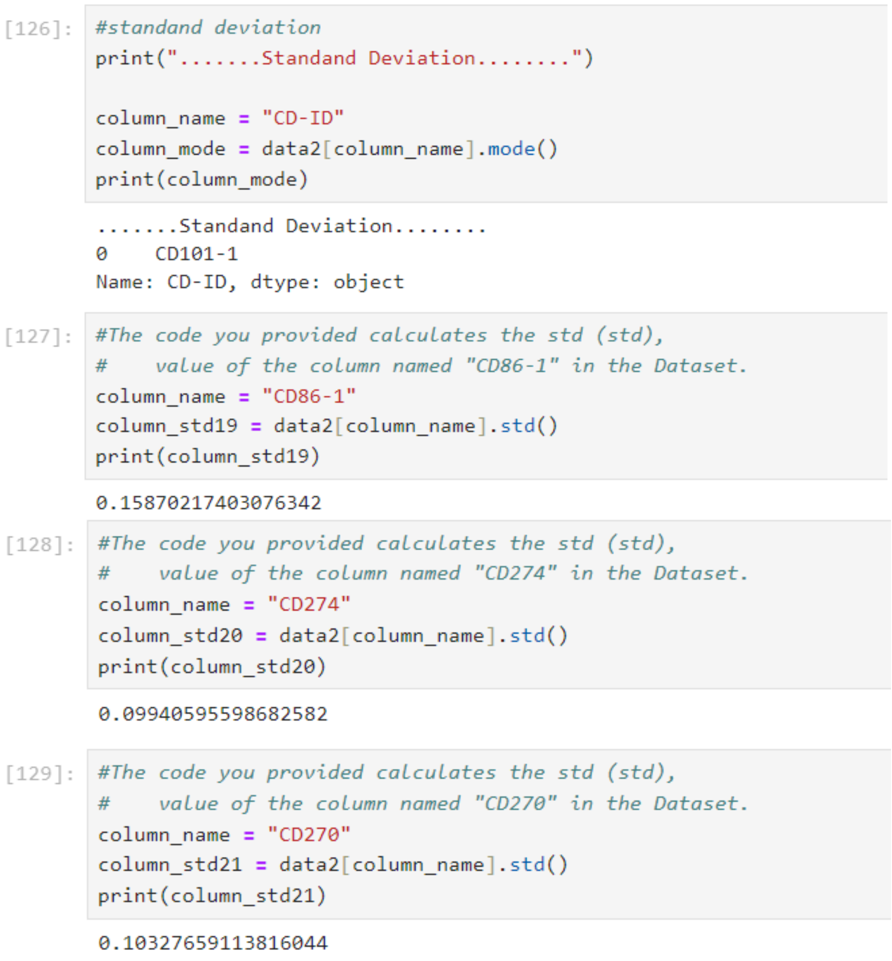
```
#The code you provided calculates the mean (average),
#    value of the column named "CD40-1" in the Dataset.
column_name = "CD40-1"
column_mean26 = data2[column_name].mean()
print(column_mean26)
```

```
0.12357751505597016
```

[125]:

```
#The code you provided calculates the mean (average),
#    value of the column named "CD154" in the Dataset.
column_name = "CD154"
column_mean27 = data2[column_name].mean()
print(column_mean27)
```

```
0.1775882595335821
```



```
[126]: #standand deviation
       print(".......Standand Deviation........")

       column_name = "CD-ID"
       column_mode = data2[column_name].mode()
       print(column_mode)
```

```
.......Standand Deviation........
0    CD101-1
Name: CD-ID, dtype: object
```

```
[127]: #The code you provided calculates the std (std),
       #    value of the column named "CD86-1" in the Dataset.
       column_name = "CD86-1"
       column_std19 = data2[column_name].std()
       print(column_std19)
```

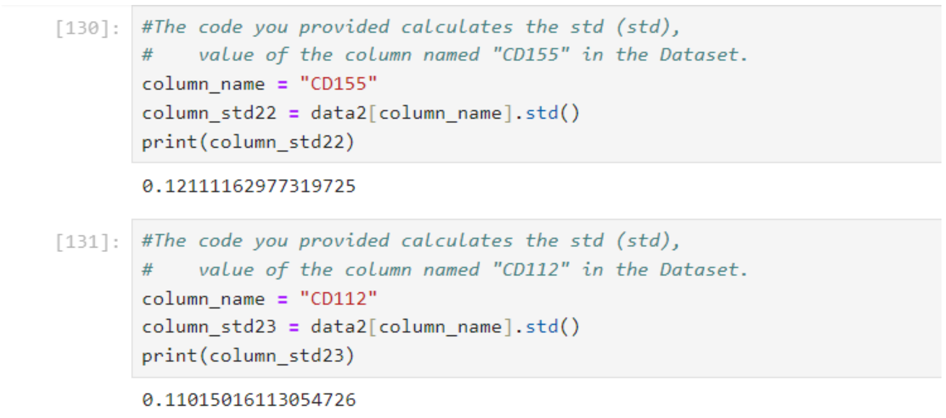
```
0.15870217403076342
```

```
[128]: #The code you provided calculates the std (std),
       #    value of the column named "CD274" in the Dataset.
       column_name = "CD274"
       column_std20 = data2[column_name].std()
       print(column_std20)
```

```
0.09940595598682582
```

```
[129]: #The code you provided calculates the std (std),
       #    value of the column named "CD270" in the Dataset.
       column_name = "CD270"
       column_std21 = data2[column_name].std()
       print(column_std21)
```

```
0.10327659113816044
```

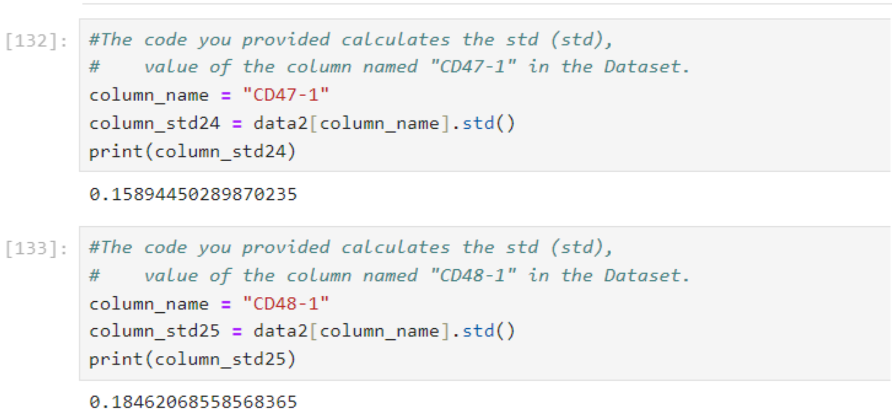


```
[130]: #The code you provided calculates the std (std),
       #    value of the column named "CD155" in the Dataset.
       column_name = "CD155"
       column_std22 = data2[column_name].std()
       print(column_std22)
```

```
0.12111162977319725
```

```
[131]: #The code you provided calculates the std (std),
       #    value of the column named "CD112" in the Dataset.
       column_name = "CD112"
       column_std23 = data2[column_name].std()
       print(column_std23)
```

```
0.11015016113054726
```

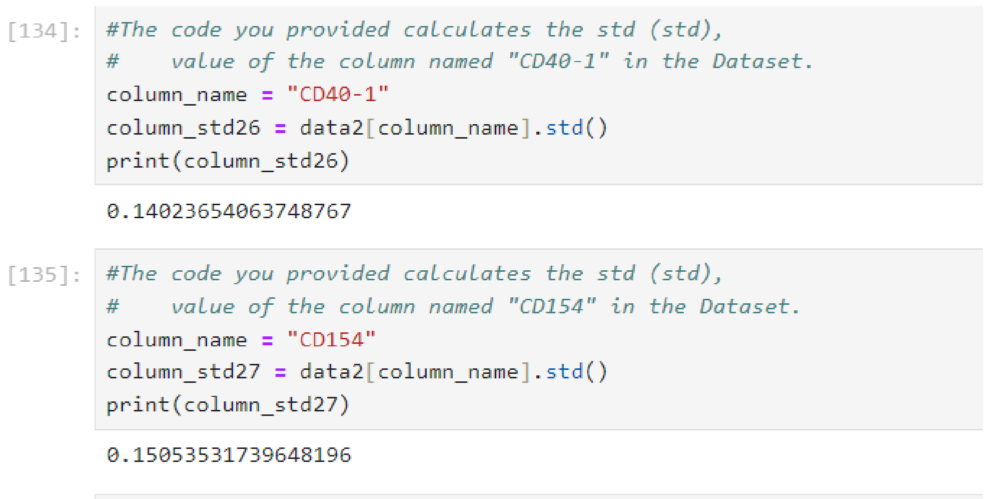


```
[132]: #The code you provided calculates the std (std),
       #    value of the column named "CD47-1" in the Dataset.
       column_name = "CD47-1"
       column_std24 = data2[column_name].std()
       print(column_std24)
```

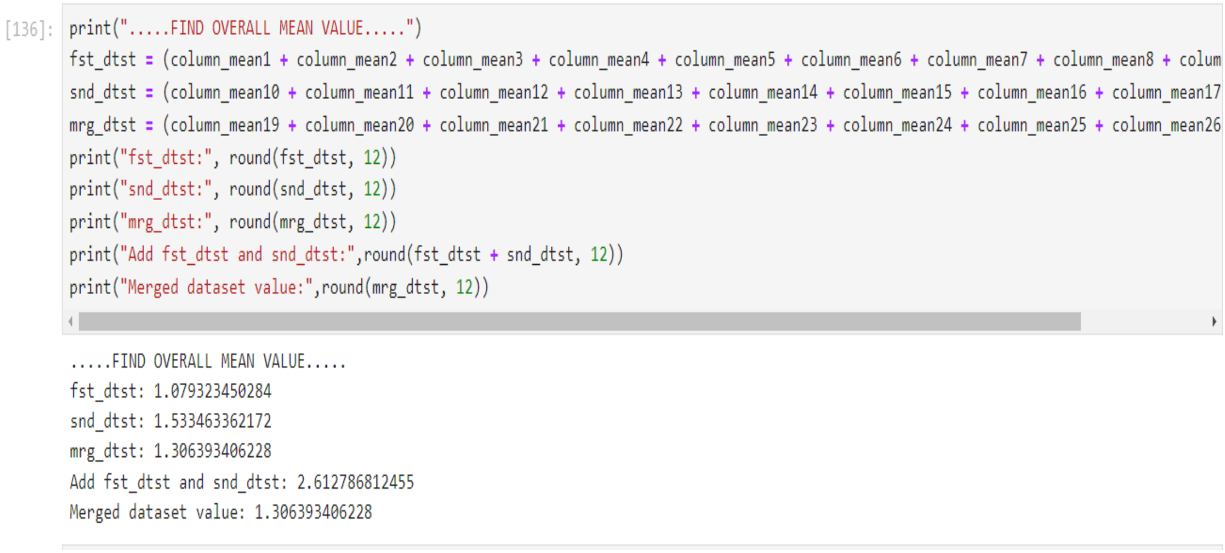
```
0.15894450289870235
```

```
[133]: #The code you provided calculates the std (std),
       #    value of the column named "CD48-1" in the Dataset.
       column_name = "CD48-1"
       column_std25 = data2[column_name].std()
       print(column_std25)
```

```
0.18462068558568365
```

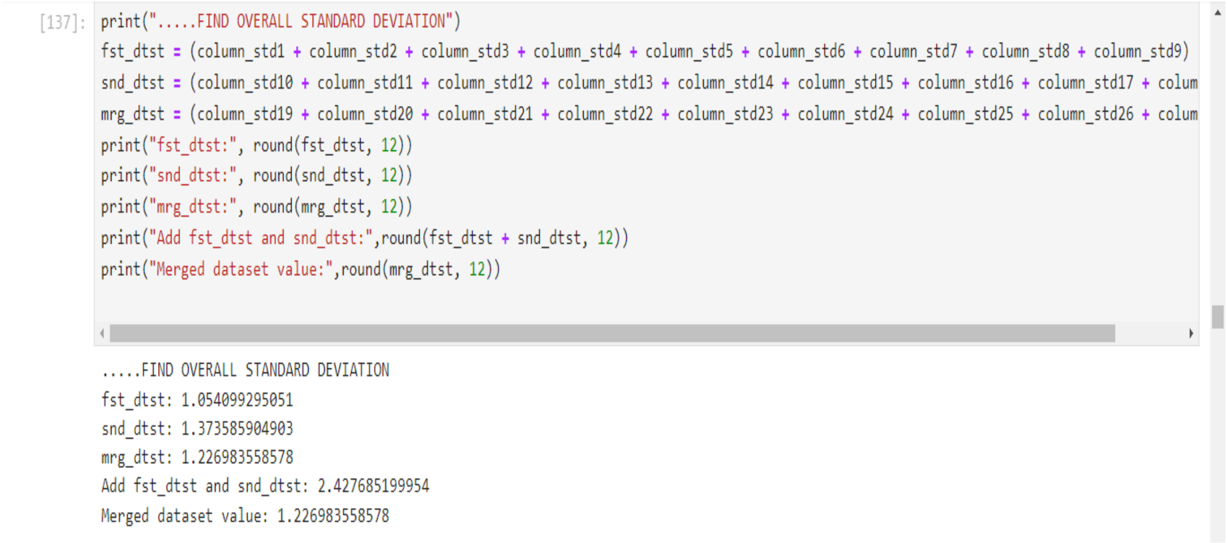


```
[134]: #The code you provided calculates the std (std),
       #    value of the column named "CD40-1" in the Dataset.
       column_name = "CD40-1"
       column_std26 = data2[column_name].std()
       print(column_std26)

       0.14023654063748767

[135]: #The code you provided calculates the std (std),
       #    value of the column named "CD154" in the Dataset.
       column_name = "CD154"
       column_std27 = data2[column_name].std()
       print(column_std27)

       0.15053531739648196
```



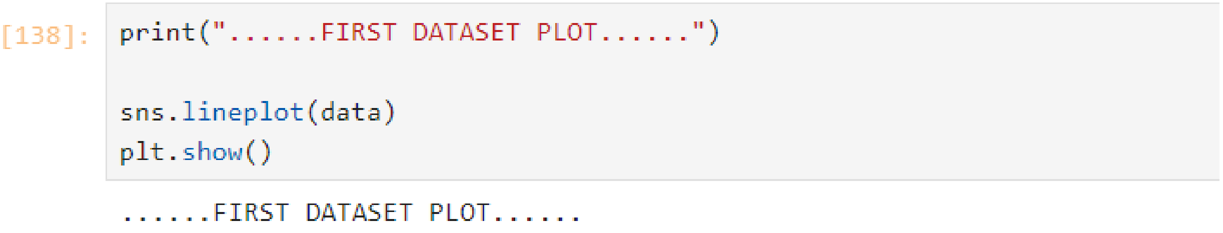
```
[136]: print(".....FIND OVERALL MEAN VALUE.....")
       fst_dtst = (column_mean1 + column_mean2 + column_mean3 + column_mean4 + column_mean5 + column_mean6 + column_mean7 + column_mean8 + colum
       snd_dtst = (column_mean10 + column_mean11 + column_mean12 + column_mean13 + column_mean14 + column_mean15 + column_mean16 + column_mean17
       mrg_dtst = (column_mean19 + column_mean20 + column_mean21 + column_mean22 + column_mean23 + column_mean24 + column_mean25 + column_mean26
       print("fst_dtst:", round(fst_dtst, 12))
       print("snd_dtst:", round(snd_dtst, 12))
       print("mrg_dtst:", round(mrg_dtst, 12))
       print("Add fst_dtst and snd_dtst:",round(fst_dtst + snd_dtst, 12))
       print("Merged dataset value:",round(mrg_dtst, 12))

       .....FIND OVERALL MEAN VALUE.....
       fst_dtst: 1.079323450284
       snd_dtst: 1.533463362172
       mrg_dtst: 1.306393406228
       Add fst_dtst and snd_dtst: 2.612786812455
       Merged dataset value: 1.306393406228
```

Here A represents value dataset and B represents value dataset and C represents linked dataset.Here dataset1 (A) and dataset2 (B) must be equal to the mean value of the combined dataset (C).A+B=2.612 and c=1.306, where two different mean values are not equal because the formula (sum of values)%(of total values Count) Dataset(A), Dataset(B) Total count of each dataset is half of dataset(c), so here we get different mean values.This is also same as standard Deviation method.



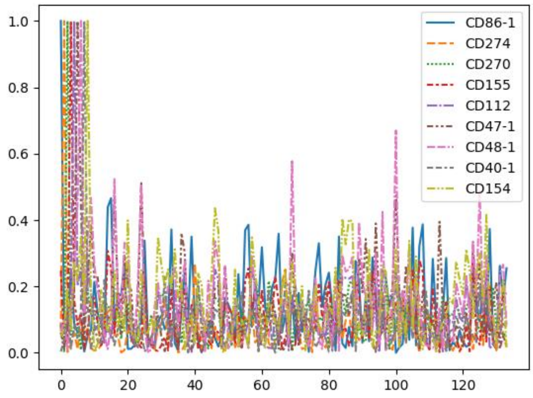
```
[137]: print(".....FIND OVERALL STANDARD DEVIATION")
       fst_dtst = (column_std1 + column_std2 + column_std3 + column_std4 + column_std5 + column_std6 + column_std7 + column_std8 + column_std9)
       snd_dtst = (column_std10 + column_std11 + column_std12 + column_std13 + column_std14 + column_std15 + column_std16 + column_std17 + colum
       mrg_dtst = (column_std19 + column_std20 + column_std21 + column_std22 + column_std23 + column_std24 + column_std25 + column_std26 + colum
       print("fst_dtst:", round(fst_dtst, 12))
       print("snd_dtst:", round(snd_dtst, 12))
       print("mrg_dtst:", round(mrg_dtst, 12))
       print("Add fst_dtst and snd_dtst:",round(fst_dtst + snd_dtst, 12))
       print("Merged dataset value:",round(mrg_dtst, 12))
```

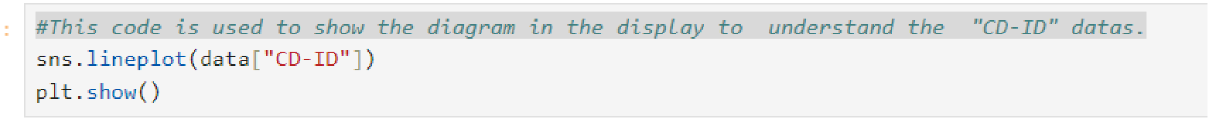
```
.....FIND OVERALL STANDARD DEVIATION
fst_dtst: 1.054099295051
snd_dtst: 1.373585904903
mrg_dtst: 1.226983558578
Add fst_dtst and snd_dtst: 2.427685199954
Merged dataset value: 1.226983558578
```



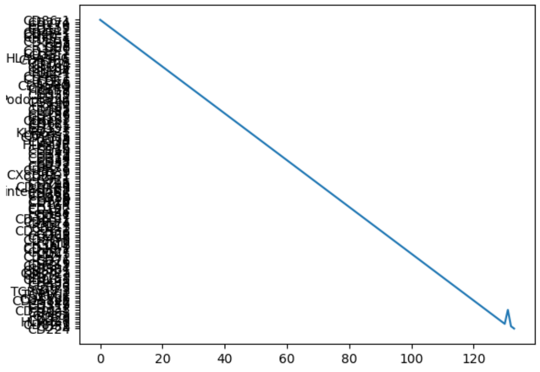
```
[138]: print("......FIRST DATASET PLOT......")

       sns.lineplot(data)
       plt.show()
```

```
......FIRST DATASET PLOT......
```



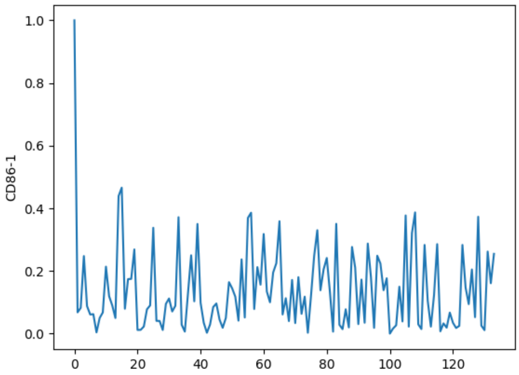


```
: #This code is used to show the diagram in the display to  understand the  "CD-ID" datas.
  sns.lineplot(data["CD-ID"])
  plt.show()
```



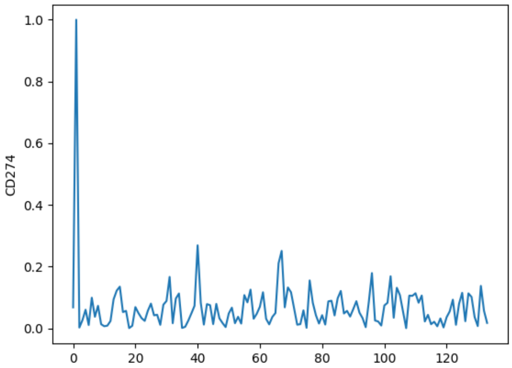
140]:

```
#This code is used to show the diagram in the display to  understand the  "CD86-1" datas.
sns.lineplot(data["CD86-1"])
plt.show()
```



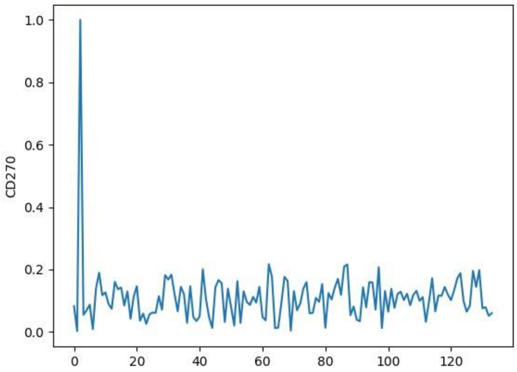
41]:

```
#This code is used to show the diagram in the display to  understand the  "CD274" datas.
sns.lineplot(data["CD274"])
plt.show()
```



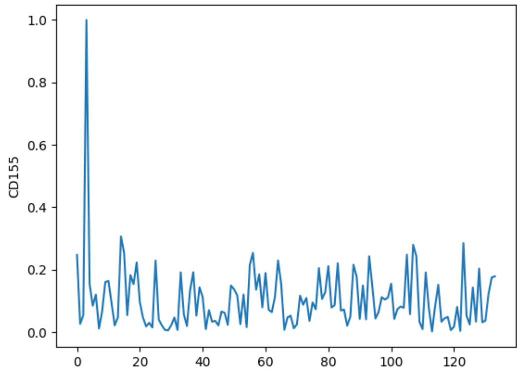
42]:

```
#This code is used to show the diagram in the display to  understand the  "CD270" datas.
sns.lineplot(data["CD270"])
plt.show()
```



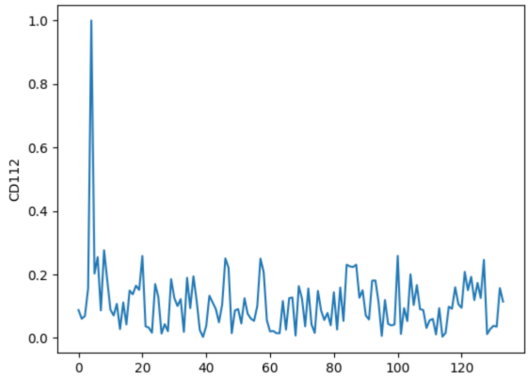
43]:

```
#This code is used to show the diagram in the display to  understand the  "CD155" datas.
sns.lineplot(data["CD155"])
plt.show()
```



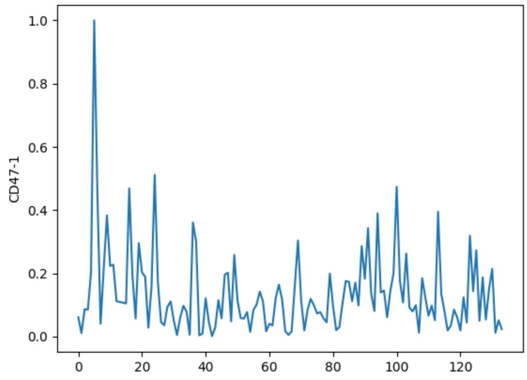
4]:

```
#This code is used to show the diagram in the display to  understand the  "CD112" datas.
sns.lineplot(data["CD112"])
plt.show()
```

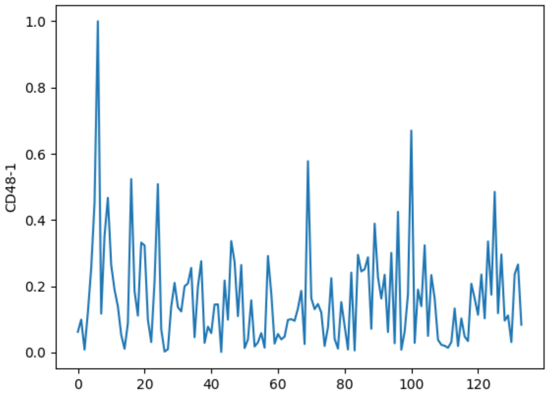


45]:

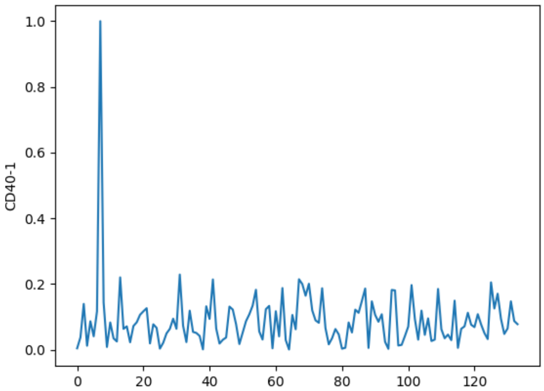
```
#This code is used to show the diagram in the display to  understand the  "CD47-1" datas.
sns.lineplot(data["CD47-1"])
plt.show()
```



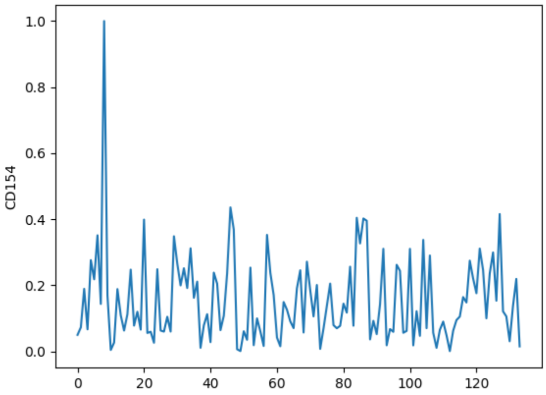
```
6]: #This code is used to show the diagram in the display to  understand the  "CD-481" datas.
    sns.lineplot(data["CD48-1"])
    plt.show()
```



```
7]: #This code is used to show the diagram in the display to  understand the  "CD40-1" datas.
    sns.lineplot(data["CD40-1"])
    plt.show()
```



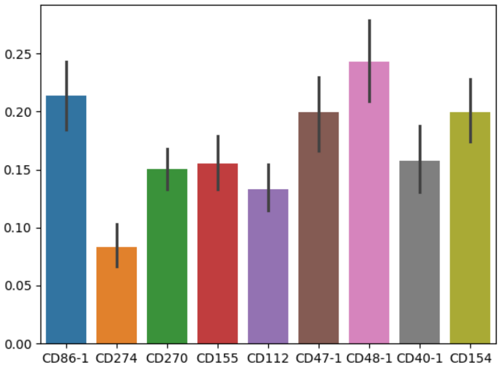
```
48]: #This code is used to show the diagram in the display to  understand the  "CD154" datas.
     sns.lineplot(data["CD154"])
     plt.show()
```



49]:

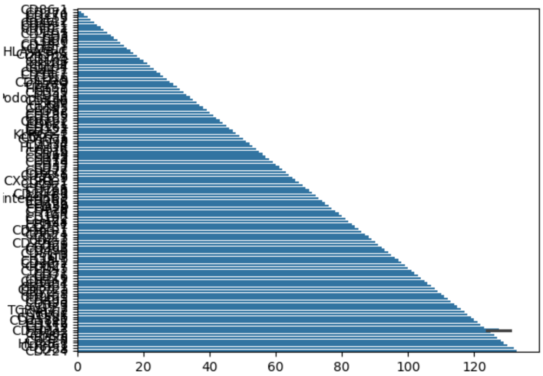
```
#This code is used to show the diagram in the display to  understand the  data1 dataset.
print("......SECOND DATASET PLOT......")

sns.barplot(data1)
plt.show()
```

```
SECOND DATASET PLOT
```



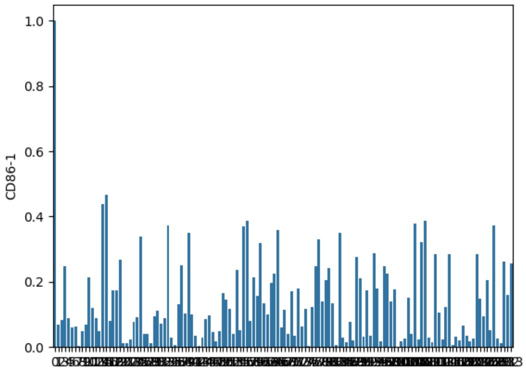
]:

```
#This code is used to show the diagram in the display to  understand the  "CD-ID" datas.
sns.barplot(data["CD-ID"])
plt.show()
```

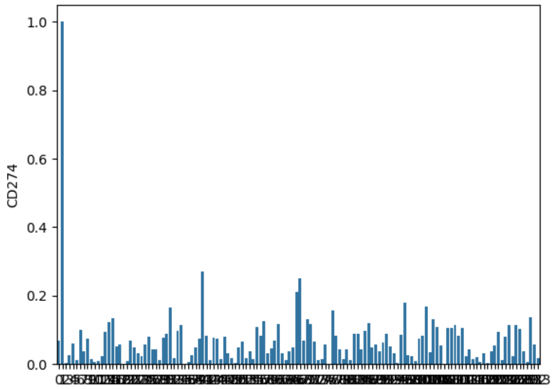


1]:

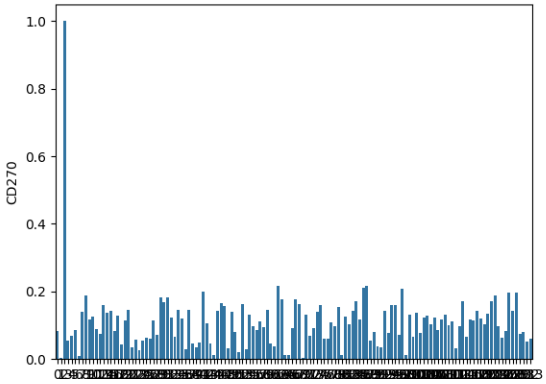
```
#This code is used to show the diagram in the display to  understand the  "CD86-1" datas.
sns.barplot(data["CD86-1"])
plt.show()
```



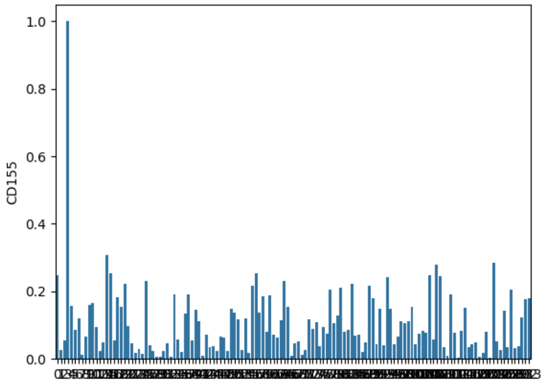
```
2]: #This code is used to show the diagram in the display to  understand the  "CD274" datas.
    sns.barplot(data["CD274"])
    plt.show()
```



```
]: #This code is used to show the diagram in the display to  understand the  "CD270" datas.
   sns.barplot(data["CD270"])
   plt.show()
```

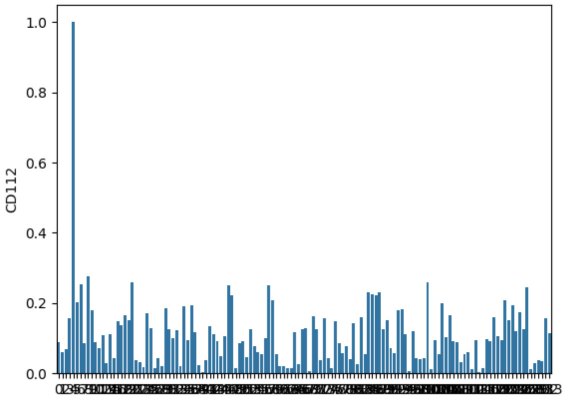


```
4]: #This code is used to show the diagram in the display to  understand the  "CD155" datas.
    sns.barplot(data["CD155"])
    plt.show()
```



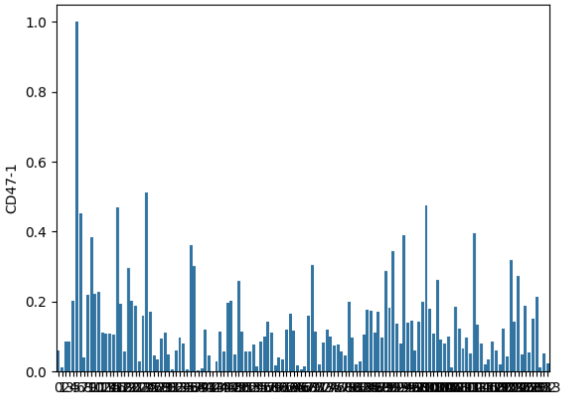
5]:

```
#This code is used to show the diagram in the display to  understand the  "CD112" datas.
sns.barplot(data["CD112"])
plt.show()
```



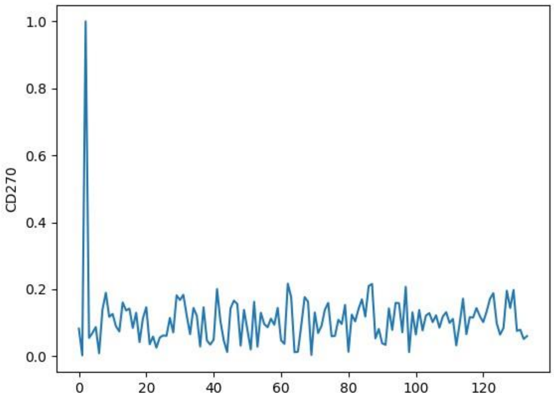
]:

```
#This code is used to show the diagram in the display to  understand the  "CD447-1" datas.
sns.barplot(data["CD47-1"])
plt.show()
```



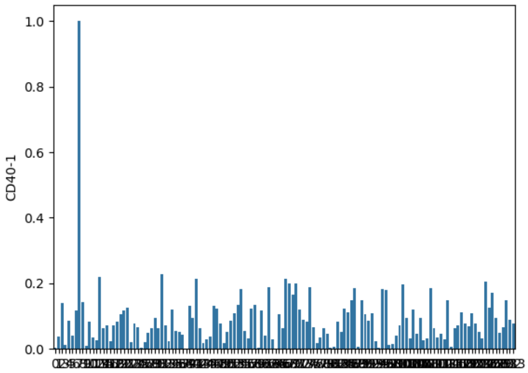
57]:

```
#This code is used to show the diagram in the display to  understand the  "CD448-1" datas.
sns.barplot(data["CD48-1"])
plt.show()
```



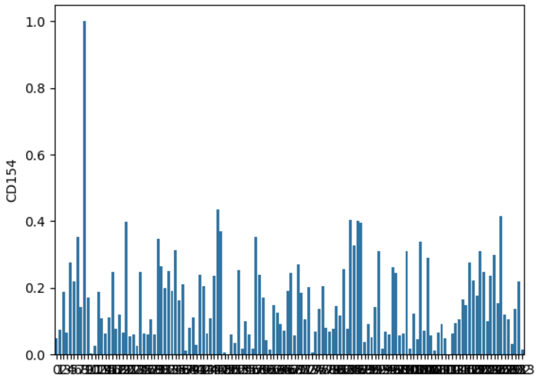
]:

```
#This code is used to show the diagram in the display to  understand the  "CD40-1" datas.
sns.barplot(data["CD40-1"])
plt.show()
```



]:

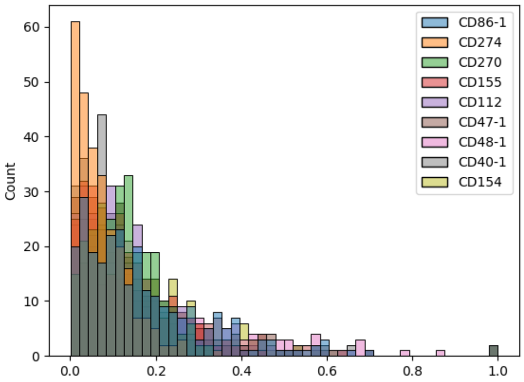
```
#This code is used to show the diagram in the display to  understand the  "CD154" datas.
sns.barplot(data["CD154"])
plt.show()
```



]:

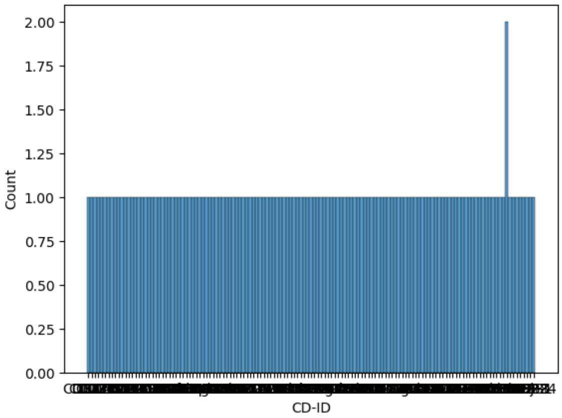
```
#This code is used to show the diagram in the display to  understand the  merged data dataset.
print("......MERGED DATASET PLOT......")

sns.histplot(data2)
plt.show()
```

```
......MERGED DATASET PLOT......
```



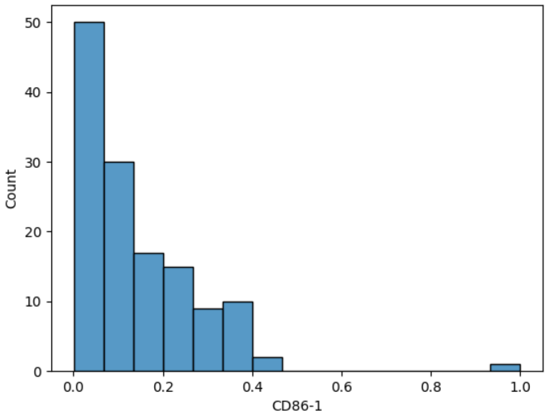
1]:

```
#This code is used to show the diagram in the display to  understand the  "CD-ID" datas.
sns.histplot(data["CD-ID"])
plt.show()
```



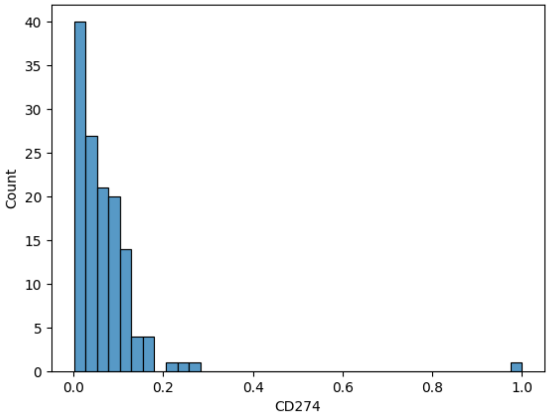
2]:

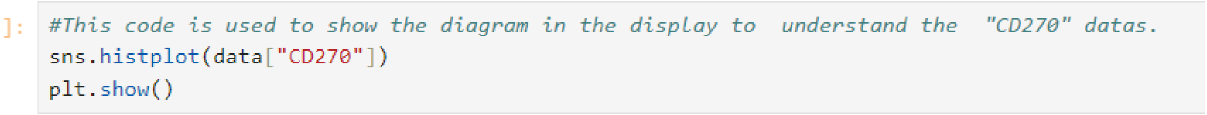
```
#This code is used to show the diagram in the display to  understand the  "CD86-1" datas.
sns.histplot(data["CD86-1"])
plt.show()
```



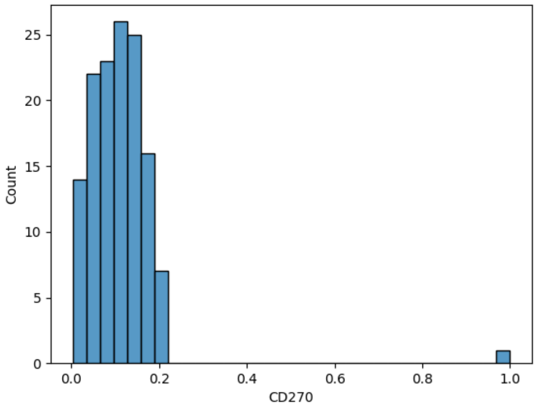
3]:

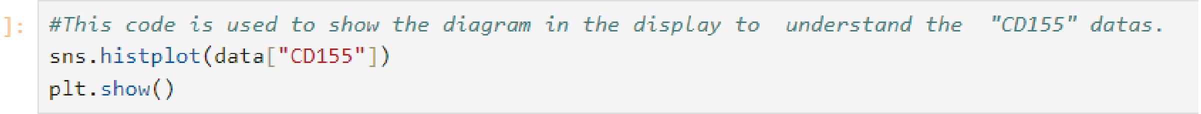
```
#This code is used to show the diagram in the display to  understand the  "CD274" datas.
sns.histplot(data["CD274"])
plt.show()
```



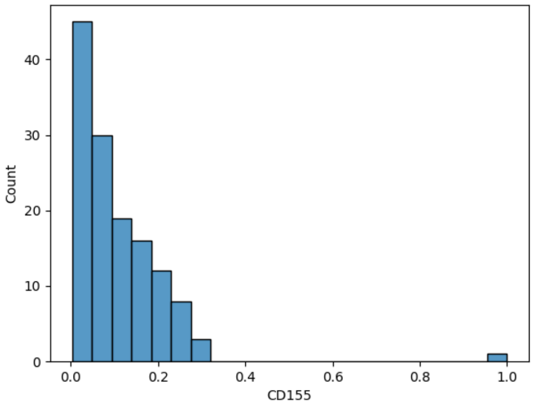


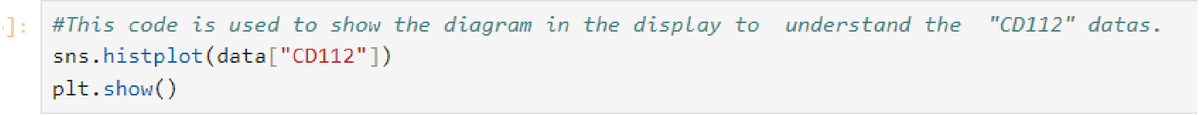
```
]: #This code is used to show the diagram in the display to  understand the  "CD270" datas.
   sns.histplot(data["CD270"])
   plt.show()
```



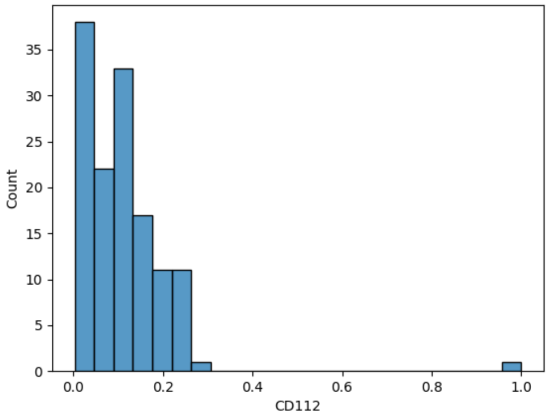


```
]: #This code is used to show the diagram in the display to  understand the  "CD155" datas.
   sns.histplot(data["CD155"])
   plt.show()
```

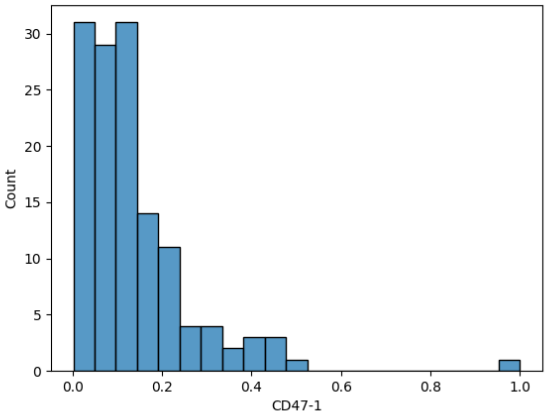




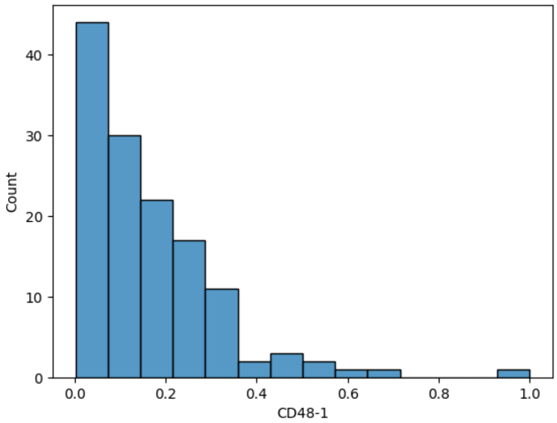
```
]: #This code is used to show the diagram in the display to  understand the  "CD112" datas.
   sns.histplot(data["CD112"])
   plt.show()
```



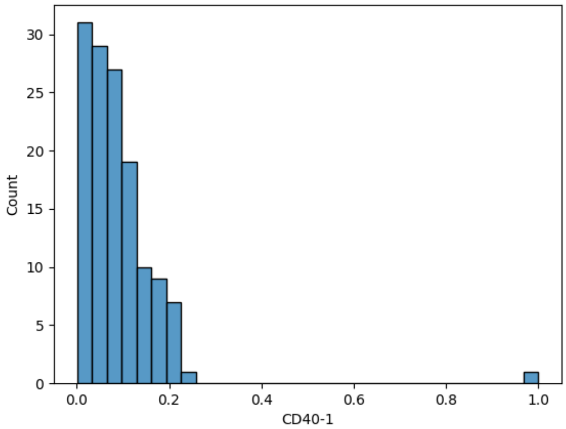
```
]: #This code is used to show the diagram in the display to  understand the  "CD447-1" datas.
   sns.histplot(data["CD47-1"])
   plt.show()
```

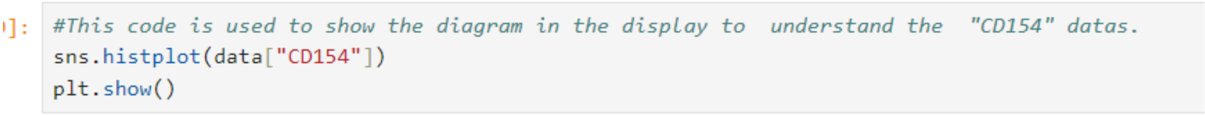


```
]: #This code is used to show the diagram in the display to  understand the  "CD48-1" datas.
   sns.histplot(data["CD48-1"])
   plt.show()
```

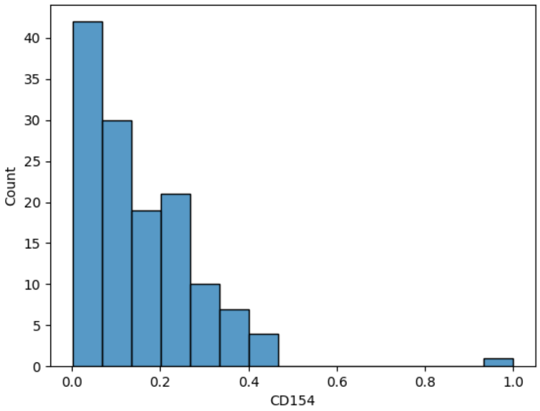


```
9]: #This code is used to show the diagram in the display to  understand the  "CD40-1" datas.
    sns.histplot(data["CD40-1"])
    plt.show()
```





```
]: #This code is used to show the diagram in the display to  understand the  "CD154" datas.
   sns.histplot(data["CD154"])
   plt.show()
```



CONCLUSION:

In conclusion, this project successfully explored, analyzed, and visualized the provided datasets, providing valuable insights into the underlying patterns and characteristics of the data.