

**Data Analysis for Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age Dataset. As a Data Scientist i want to calculate the mean of 'Total Deaths', the median of 'COVID-19 Deaths', the mode of the 'Sex' with higher death occurrence, range of 'Influenza Deaths', the variance of 'Influenza Death' spread of the dataset, the Standard Deviation of 'Influenza Death' deviation score, the PMF(probability mass function) of 'Influenza Death' mass function, the PDF(probability density function) of 'Influenza Death' and visualize the distributions of the dataset to derive useful insight from 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset.**

## **DataSet Overview.**

All data come from one source which was csv file shared and contains details in columns as following.

index: just index.

Data As Of: Initial data point.

Start Date: Start date.

End Date: End date.

Group: The group order.

Year: The year.

Month: The month.

State: The state.

Sex: The gender(Sex).

Age Group: The age group order.

COVID-19 Deaths: The COVID-19 Deaths scores.

Total Deaths: Total death scores.

Pneumonia Deaths: The Pneumonia Deaths scores.

Pneumonia and COVID-19 Deaths: The Pneumonia and COVID-19 Deaths scores.

Influenza Deaths: The Influenza Deaths scores.

Pneumonia, Influenza, or COVID-19 Deaths: The Pneumonia, Influenza, or COVID-19 Deaths scores.

Footnote: The Footnote records.

## **Table of Content**

We will go through several tasks to achieve our goals and purposes:

Task1: Problem statement(goals and purpose) for Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age Dataset for Analysis with an overview of the whole Dataset.

Task2: Importing libraries.

Task3: Loading data from my google drive, i used 'COLAB' for my python programming.

Task4: Viewing the data, the head view of the 1st 5 rows of the data.

Task5: Viewing the data, the bottom view of the 1st 5 rows of the train data.

Task6: To view the shape of the data.

Task7: Checking for NAN/NULL values(missing values) of the data.

Task8: To get the details of the data columns nature information.

Task9: To get the full descriptive statistics chart table for 'Year' column from the data with missing values.

Task10: Computing mean value Year to the missing values in Year column

Task11: Computing mean value Month to the missing values in Month column

Task12: Computing mean value COVID-19 Deaths to the missing values in COVID-19 Deaths column

Task13: Computing mean value Total Deaths to the missing values in Total Deaths column

Task14: Computing mean value Pneumonia Deaths to the missing values in Pneumonia Deaths column

Task15: Computing mean value Pneumonia and COVID-19 Deaths to the missing values in Pneumonia and COVID-19 Deaths column

Task16: Computing mean value Influenza Deaths to the missing values in Influenza Deaths column

Task17: Computing mean value Pneumonia, Influenza, or COVID-19 Deaths to the missing values in Pneumonia, Influenza, or COVID-19 Deaths column

Task18: To calculate the mean of Total Deaths and the mean report Total Deaths.

Task19: To calculate for median of COVID-19 Deaths column and report

Task20: Calculating for the mode of the Sex with higher death occurrence and report.

Task21: To calculate the range of Influenza Deaths in the dataset and report.

Task22: To calculate the variance of Influenza Death spread of the dataset and report.

Task23: To calculate the Standard Deviation of Influenza Death , Deviation score the dataset and report.

Task24: To calculate the PMF of Influenza Death , the mass function value and report.

Task25: To calculate the PDF of Influenza Death , the distribution influenza death function values and report.

References: The sources i took the code from

### **Task1**

#### **Problem Statement(Aim):**

Data Analysis for Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age. As a Data Scientist i want to calculate the mean of 'Total Deaths', the median of 'COVID-19 Deaths', the mode of the 'Sex' with higher death occurrence, range of 'Influenza Deaths', the variance of 'Influenza Death' spread of the dataset, the Standard Deviation of 'Influenza Death' Deviation score, the PMF(probability mass function) of 'Influenza Death' mass function, the PDF(probability density function) of 'Influenza Death' and visualize the distributions of the dataset to derive useful insight from 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' data.

### **Task2:**

Importing libraries

```
[ ] import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import statistics
```

### Task3:

Loading data from my google drive, i used 'COLAB' for my python programming.

```
[ ] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] df= pd.read_csv(r'/content/drive/MyDrive/mathematics dataset/Provisional_COVID-19_Deaths_by_Sex_and_Age.csv')
```

### Task4:

Viewing the data, the head view of the 1st 5 rows of the data.

```
[ ] df.head()
```

	Data As Of	Start Date	End Date	Group	Year	Month	State	Sex	Age Group	COVID-19 Deaths	Total Deaths	Pneumonia Deaths	Pneumonia and COVID- 19 Deaths	Influenza Deaths	Pneumonia, Influenza, or COVID-19 Deaths	Footnote
0	11/09/2022	01/01/2020	11/05/2022	By Total	NaN	NaN	United States	All Sexes	All Ages	1067539.0	9521311.0	987383.0	540649.0	12621.0	1525057.0	NaN
1	11/09/2022	01/01/2020	11/05/2022	By Total	NaN	NaN	United States	All Sexes	Under 1 year	380.0	55203.0	740.0	62.0	43.0	1099.0	NaN
2	11/09/2022	01/01/2020	11/05/2022	By Total	NaN	NaN	United States	All Sexes	0-17 years	1350.0	98109.0	2043.0	343.0	270.0	3306.0	NaN
3	11/09/2022	01/01/2020	11/05/2022	By Total	NaN	NaN	United States	All Sexes	1-4 years	202.0	10536.0	434.0	48.0	89.0	673.0	NaN
4	11/09/2022	01/01/2020	11/05/2022	By Total	NaN	NaN	United States	All Sexes	5-14 years	423.0	16461.0	569.0	121.0	110.0	974.0	NaN

### Task5:

Viewing the data, the bottom view of the 1st 5 rows of the train data.

df.tail()

	Data As Of	Start Date	End Date	Group	Year	Month	State	Sex	Age Group	COVID-19 Deaths	Total Deaths	Pneumonia Deaths	Pneumonia and COVID-19 Deaths	Influenza Deaths	Pneumonia, Influenza, or COVID-19 Deaths	Footnote
107401	11/09/2022	11/01/2022	11/05/2022	By Month	2022.0	11.0	Puerto Rico	Female	50-64 years	0.0	NaN	0.0	0.0	0.0	0.0	One or more data cells have counts between 1-9...
107402	11/09/2022	11/01/2022	11/05/2022	By Month	2022.0	11.0	Puerto Rico	Female	55-64 years	0.0	NaN	0.0	0.0	0.0	0.0	One or more data cells have counts between 1-9...
107403	11/09/2022	11/01/2022	11/05/2022	By Month	2022.0	11.0	Puerto Rico	Female	65-74 years	0.0	12.0	NaN	0.0	0.0	NaN	One or more data cells have counts between 1-9...
107404	11/09/2022	11/01/2022	11/05/2022	By Month	2022.0	11.0	Puerto Rico	Female	75-84 years	NaN	15.0	NaN	0.0	0.0	NaN	One or more data cells have counts between 1-9...
107405	11/09/2022	11/01/2022	11/05/2022	By Month	2022.0	11.0	Puerto Rico	Female	85 years and over	NaN	11.0	NaN	NaN	NaN	NaN	One or more data cells have counts between 1-9...

## Task6:

To view the shape of the data.

```
df.shape
```

```
(107406, 16)
```

The data contains 107406 files and 16 colums

## Task7:

Checking for NAN/NULL values(missing values) of the data.



df.isnull().sum()

Data As Of	0
Start Date	0
End Date	0
Group	0
Year	2754
Month	11016
State	0
Sex	0
Age Group	0
COVID-19 Deaths	28933
Total Deaths	15500
Pneumonia Deaths	33185
Pneumonia and COVID-19 Deaths	27787
Influenza Deaths	18726
Pneumonia, Influenza, or COVID-19 Deaths	32475
Footnote	34228
dtype: int64	

There are 2754 missing values in Year column, 11016 missing values in Month, 28933 missing values in COVID-19 Deaths, 15500 missing values in Total Deaths, 33185 missing values in Pneumonia Deaths, 27787 missing values in Pneumonia and COVID-19 Deaths, 18726 missing values in Influenza Deaths, 32475 missing values in Pneumonia, Influenza, or COVID-19 Deaths, 34228 missing values in Footnote. I have to fill the missing values using mean method for numerical missing values and mode method for categorical values.

### Task8:

To get the details of the data columns nature information.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
#   Column                                          Non-Null Count  Dtype
---  -
0   Data As Of                                    107406 non-null  object
1   Start Date                                    107406 non-null  object
2   End Date                                       107406 non-null  object
3   Group                                          107406 non-null  object
4   Year                                           104652 non-null  float64
5   Month                                          96390 non-null   float64
6   State                                          107406 non-null  object
7   Sex                                            107406 non-null  object
8   Age Group                                     107406 non-null  object
9   COVID-19 Deaths                             78473 non-null   float64
10  Total Deaths                                91906 non-null   float64
11  Pneumonia Deaths                           74221 non-null   float64
12  Pneumonia and COVID-19 Deaths               79619 non-null   float64
13  Influenza Deaths                           88680 non-null   float64
14  Pneumonia, Influenza, or COVID-19 Deaths    74931 non-null   float64
15  Footnote                                      73178 non-null   object
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

There are 107406 entries, ranges from 0 to 107405 Data columns (total 16 columns) dtypes is float64(8), object(8) of the data. I will convert the "Object" to "Categorical" for proper data cleaning order.

### Task9:

To get the full descriptive statistics chart table for 'Year' column from the data with missing values.

```
df['Year'].describe()

count      104652.000000
mean        2020.973684
std          0.810684
min         2020.000000
25%         2020.000000
50%         2021.000000
75%         2022.000000
max         2022.000000
Name: Year, dtype: float64
```

Year is numerical column so i fill it with Mean Imputation. To remove null values in Year by computing the mean value of the column since Year is a numerical values.

### Task10:

Computing mean value Year to the missing values in Year column.

```
[ ] df['Year'].fillna(df['Year'].mean(),inplace=True)
```

To check if there is null values in Year column

```
df.isnull().sum()
```

Data As Of	0
Start Date	0
End Date	0
Group	0
Year	0
Month	11016
State	0
Sex	0
Age Group	0
COVID-19 Deaths	28933
Total Deaths	15500
Pneumonia Deaths	33185
Pneumonia and COVID-19 Deaths	27787
Influenza Deaths	18726
Pneumonia, Influenza, or COVID-19 Deaths	32475
Footnote	34228
dtype: int64	

We can see that all the missing values in year column has been filled with the mean value of year column.

### Task11:

Computing mean value Month to the missing values in Month column.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
#   Column                                          Non-Null Count  Dtype
---  -
0   Data As Of                                    107406 non-null object
1   Start Date                                    107406 non-null object
2   End Date                                      107406 non-null object
3   Group                                          107406 non-null object
4   Year                                           107406 non-null float64
5   Month                                          96390 non-null  float64
6   State                                          107406 non-null object
7   Sex                                            107406 non-null object
8   Age Group                                     107406 non-null object
9   COVID-19 Deaths                             78473 non-null  float64
10  Total Deaths                                91906 non-null  float64
11  Pneumonia Deaths                             74221 non-null  float64
12  Pneumonia and COVID-19 Deaths                 79619 non-null  float64
13  Influenza Deaths                             88680 non-null  float64
14  Pneumonia, Influenza, or COVID-19 Deaths      74931 non-null  float64
15  Footnote                                       73178 non-null  object
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

Month is numerical column so i fill it with Mean Inputation. To remove null values in Month by computing the mean value of the column since Month is a numerical values.

```
[ ] df['Month'].fillna(df['Month'].mean(),inplace=True)

To check if there is null values in Month column

df.isnull().sum()

Data As Of      0
Start Date      0
End Date        0
Group           0
Year            0
Month           0
State           0
Sex             0
Age Group       0
COVID-19 Deaths 28933
Total Deaths    15500
Pneumonia Deaths 33185
Pneumonia and COVID-19 Deaths 27787
Influenza Deaths 18726
Pneumonia, Influenza, or COVID-19 Deaths 32475
Footnote        34228
dtype: int64
```

We can see that all the missing values in Month column has been filled with the mean value of Month column.

## Task12:

Computing mean value COVID-19 Deaths to the missing values in COVID-19 Deaths column.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   Data As Of                               107406 non-null object
 1   Start Date                               107406 non-null object
 2   End Date                                 107406 non-null object
 3   Group                                    107406 non-null object
 4   Year                                     107406 non-null float64
 5   Month                                   107406 non-null float64
 6   State                                    107406 non-null object
 7   Sex                                      107406 non-null object
 8   Age Group                               107406 non-null object
 9   COVID-19 Deaths                         78473 non-null float64
10   Total Deaths                             91906 non-null float64
11   Pneumonia Deaths                         74221 non-null float64
12   Pneumonia and COVID-19 Deaths            79619 non-null float64
13   Influenza Deaths                         88680 non-null float64
14   Pneumonia, Influenza, or COVID-19 Deaths 74931 non-null float64
15   Footnote                                 73178 non-null object
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

COVID-19 Deaths is numerical column so i fill it with Mean Imputation. To remove null values in COVID-19 Deaths by computing the mean value of the column since COVID-19 Deaths is a numerical values.

```
[ ] df['COVID-19 Deaths'].fillna(df['COVID-19 Deaths'].mean(),inplace=True)
```

To check if there is null values in COVID-19 Deaths column

```
df.isnull().sum()
```

Data As Of	0
Start Date	0
End Date	0
Group	0
Year	0
Month	0
State	0
Sex	0
Age Group	0
COVID-19 Deaths	0
Total Deaths	15500
Pneumonia Deaths	33185
Pneumonia and COVID-19 Deaths	27787
Influenza Deaths	18726
Pneumonia, Influenza, or COVID-19 Deaths	32475
Footnote	34228
dtype: int64	

We can see that all the missing values in COVID-19 Deaths column has been filled with the COVID-19 Deaths value of COVID-19 column.

### Task13:

Computing mean value Total Deaths to the missing values in Total Deaths column.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Data As Of                               107406 non-null object
1   Start Date                               107406 non-null object
2   End Date                                 107406 non-null object
3   Group                                    107406 non-null object
4   Year                                     107406 non-null float64
5   Month                                    107406 non-null float64
6   State                                    107406 non-null object
7   Sex                                      107406 non-null object
8   Age Group                               107406 non-null object
9   COVID-19 Deaths                         78473 non-null  float64
10  Total Deaths                             91906 non-null  float64
11  Pneumonia Deaths                         74221 non-null  float64
12  Pneumonia and COVID-19 Deaths            79619 non-null  float64
13  Influenza Deaths                         88680 non-null  float64
14  Pneumonia, Influenza, or COVID-19 Deaths 74931 non-null  float64
15  Footnote                                  73178 non-null  object
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

Total Deaths is numerical column so i fill it with Mean Imputation. To remove null values in Total Deaths by computing the mean value of the column since Total Deaths is a numerical values.

```
[ ] df['Total Deaths'].fillna(df['Total Deaths'].mean(),inplace=True)
```

To check if there is null values in Total Deaths column

```
df.isnull().sum()
```

Data As Of	0
Start Date	0
End Date	0
Group	0
Year	0
Month	0
State	0
Sex	0
Age Group	0
COVID-19 Deaths	0
Total Deaths	0
Pneumonia Deaths	33185
Pneumonia and COVID-19 Deaths	27787
Influenza Deaths	18726
Pneumonia, Influenza, or COVID-19 Deaths	32475
Footnote	34228
dtype: int64	

We can see that all the missing values in Total Deaths column has been filled with the Total Deaths value of Total Deaths column.

#### Task14:

Computing mean value Pneumonia Deaths to the missing values in Pneumonia Deaths column.



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
 #   Column                                          Non-Null Count  Dtype
---  -
 0   Data As Of                                    107406 non-null object
 1   Start Date                                    107406 non-null object
 2   End Date                                       107406 non-null object
 3   Group                                          107406 non-null object
 4   Year                                           107406 non-null float64
 5   Month                                          107406 non-null float64
 6   State                                          107406 non-null object
 7   Sex                                            107406 non-null object
 8   Age Group                                     107406 non-null object
 9   COVID-19 Deaths                             107406 non-null float64
10   Total Deaths                                107406 non-null float64
11   Pneumonia Deaths                             74221 non-null  float64
12   Pneumonia and COVID-19 Deaths                 79619 non-null  float64
13   Influenza Deaths                             88680 non-null  float64
14   Pneumonia, Influenza, or COVID-19 Deaths      74931 non-null  float64
15   Footnote                                       73178 non-null  object
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

Pneumonia Deaths is numerical column so i fill it with Mean Imputation. To remove null values in Pneumonia Deaths by computing the mean value of the column since Pneumonia Deaths is a numerical values.

```
[ ] df['Pneumonia Deaths'].fillna(df['Pneumonia Deaths'].mean(),inplace=True)
```

To check if there is null values in Pneumonia Deaths column

```
df.isnull().sum()
```

```
Data As Of                                0
Start Date                                0
End Date                                  0
Group                                      0
Year                                       0
Month                                      0
State                                      0
Sex                                        0
Age Group                                 0
COVID-19 Deaths                          0
Total Deaths                             0
Pneumonia Deaths                          0
Pneumonia and COVID-19 Deaths             27787
Influenza Deaths                         18726
Pneumonia, Influenza, or COVID-19 Deaths 32475
Footnote                                  34228
dtype: int64
```

We can see that all the missing values in Pneumonia Deaths column has been filled with the Pneumonia Deaths value of Pneumonia Deaths column.



## Task15:

Computing mean value Pneumonia and COVID-19 Deaths to the missing values in Pneumonia and COVID-19 Deaths column.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Data As Of                               107406 non-null  object
1   Start Date                               107406 non-null  object
2   End Date                                 107406 non-null  object
3   Group                                    107406 non-null  object
4   Year                                     107406 non-null  float64
5   Month                                    107406 non-null  float64
6   State                                    107406 non-null  object
7   Sex                                       107406 non-null  object
8   Age Group                               107406 non-null  object
9   COVID-19 Deaths                        107406 non-null  float64
10  Total Deaths                            107406 non-null  float64
11  Pneumonia Deaths                        107406 non-null  float64
12  Pneumonia and COVID-19 Deaths           79619 non-null   float64
13  Influenza Deaths                        88680 non-null   float64
14  Pneumonia, Influenza, or COVID-19 Deaths 74931 non-null   float64
15  Footnote                                 73178 non-null   object
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

Pneumonia and COVID-19 Deaths is numerical column so i fill it with Mean Imputation. To remove null values in Pneumonia and COVID-19 Deaths by computing the mean value of the column since Pneumonia and COVID-19 Deaths is a numerical values.

```
[ ] df['Pneumonia and COVID-19 Deaths'].fillna(df['Pneumonia and COVID-19 Deaths'].mean(),inplace=True)

To check if there is null values in Pneumonia and COVID-19 Deaths column

df.isnull().sum()

Data As Of          0
Start Date          0
End Date            0
Group               0
Year                0
Month               0
State               0
Sex                 0
Age Group           0
COVID-19 Deaths    0
Total Deaths       0
Pneumonia Deaths   0
Pneumonia and COVID-19 Deaths 0
Influenza Deaths   18726
Pneumonia, Influenza, or COVID-19 Deaths 32475
Footnote            34228
dtype: int64
```

We can see that all the missing values in Pneumonia and COVID-19 Deaths column has been filled with the Pneumonia and COVID-19 Deaths value of Pneumonia and COVID-19 Deaths column.

### Task16:

Computing mean value Influenza Deaths to the missing values in Influenza Deaths column.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Data As Of                               107406 non-null object
1   Start Date                               107406 non-null object
2   End Date                                 107406 non-null object
3   Group                                    107406 non-null object
4   Year                                     107406 non-null float64
5   Month                                   107406 non-null float64
6   State                                    107406 non-null object
7   Sex                                      107406 non-null object
8   Age Group                               107406 non-null object
9   COVID-19 Deaths                        107406 non-null float64
10  Total Deaths                           107406 non-null float64
11  Pneumonia Deaths                       107406 non-null float64
12  Pneumonia and COVID-19 Deaths          107406 non-null float64
13  Influenza Deaths                       88680 non-null  float64
14  Pneumonia, Influenza, or COVID-19 Deaths 74931 non-null  float64
15  Footnote                                73178 non-null  object
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

Influenza Deaths is numerical column so i fill it with Mean Imputation. To remove null values in Pneumonia and Influenza Deaths by computing the mean value of the column since Pneumonia and Influenza Deaths is a numerical values.

```
[ ] df['Influenza Deaths'].fillna(df['Influenza Deaths'].mean(),inplace=True)
```

To check if there is null values in Influenza Deaths column

```
[ ] df.isnull().sum()
```

```
Data As Of      0
Start Date      0
End Date        0
Group           0
Year            0
Month           0
State           0
Sex             0
Age Group       0
COVID-19 Deaths 0
Total Deaths    0
Pneumonia Deaths 0
Pneumonia and COVID-19 Deaths 0
Influenza Deaths 0
Pneumonia, Influenza, or COVID-19 Deaths 32475
Footnote        34228
dtype: int64
```

We can see that all the missing values in Influenza Deaths column has been filled with the Influenza Deaths value of Influenza Deaths column.

### Task17:

Computing mean value Pneumonia, Influenza, or COVID-19 Deaths to the missing values in Pneumonia, Influenza, or COVID-19 Deaths column.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107406 entries, 0 to 107405
Data columns (total 16 columns):
```

#	Column	Non-Null Count	Dtype
0	Data As Of	107406 non-null	object
1	Start Date	107406 non-null	object
2	End Date	107406 non-null	object
3	Group	107406 non-null	object
4	Year	107406 non-null	float64
5	Month	107406 non-null	float64
6	State	107406 non-null	object
7	Sex	107406 non-null	object
8	Age Group	107406 non-null	object
9	COVID-19 Deaths	107406 non-null	float64
10	Total Deaths	107406 non-null	float64
11	Pneumonia Deaths	107406 non-null	float64
12	Pneumonia and COVID-19 Deaths	107406 non-null	float64
13	Influenza Deaths	107406 non-null	float64
14	Pneumonia, Influenza, or COVID-19 Deaths	74931 non-null	float64
15	Footnote	73178 non-null	object

```
dtypes: float64(8), object(8)
memory usage: 13.1+ MB
```

Pneumonia, Influenza, or COVID-19 Deaths is numerical column so i fill it with Mean Imputation. To remove null values in Pneumonia, Influenza, or COVID-19 Deaths by computing the mean value of the column since Pneumonia, Influenza, or COVID-19 Deaths is a numerical values.

```
[ ] df['Pneumonia, Influenza, or COVID-19 Deaths'].fillna(df['Pneumonia, Influenza, or COVID-19 Deaths'].mean(),inplace=True)
```

To check if there is null values in Pneumonia, Influenza, or COVID-19 Deaths colum

```
[ ] df.isnull().sum()
```

Data As Of	0
Start Date	0
End Date	0
Group	0
Year	0
Month	0
State	0
Sex	0
Age Group	0
COVID-19 Deaths	0
Total Deaths	0
Pneumonia Deaths	0
Pneumonia and COVID-19 Deaths	0
Influenza Deaths	0
Pneumonia, Influenza, or COVID-19 Deaths	0
Footnote	34228
dtype: int64	

We can see that all the missing values in Pneumonia, Influenza, or COVID-19 Deaths colum has been filled with the Pneumonia, Influenza, or COVID-19 Deaths value of Pneumonia, Influenza, or COVID-19 Deaths column.

Our data is totally clean and contains no missing value for the colums features that we are using for our mathematical calculations and reports.

```
[40] df.shape
```

```
(107406, 16)
```

The total values of the dataset are 107406 with 16 colums

**Task18:**

To calculate the mean of Total Deaths and the mean report Total Deaths.

```
[ ] df_TotalDeaths = 'Total Deaths'

[ ] TotalDeaths_mean = df[df_TotalDeaths].mean()

[ ] print(TotalDeaths_mean)

2830.7697103562336
```

**Report:**

The 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset reveals that the mean of the Total Deaths is approximately 2830.77. This statistical measure provides insight into the average mortality across different demographic groups, emphasizing the significance of understanding COVID-19's impact on various age and gender categories.

**Task19:**

To calculate for median of COVID-19 Deaths column and report.

```
[ ] df_COVID_Deaths = 'COVID-19 Deaths'

[ ] COVID_Deaths_median = df[df_COVID_Deaths].median()

[ ] print(COVID_Deaths_median)

38.0
```

**Report:**

In the 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset, the median for 'COVID-19 Deaths' is 38.0. This central tendency measure provides a representative value, indicating the middle point of the distribution. Understanding the median contributes to a more comprehension of mortality patterns across different demographics.

**Task20:**

Calculating for the mode of the Sex with higher death occurrence and report.

```
[ ] df_Sex = 'Sex'

[ ] Sex_mode = df[df_Sex].mode()[0]

[ ] print(Sex_mode)

All Sexes
```

**Report:**

The 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset reveals that the mode for 'Sex' is "All Sexes." This statistical mode indicates the most frequently occurring category, emphasizing the predominant representation of data related to all sexes in the context of COVID-19 deaths.

### Task21:

To calculate the range of Influenza Deaths in the dataset and report.

```
[ ] df_InfluenzaDeaths = 'Influenza Deaths'

[ ] InfluenzaDeaths_range = df[df_InfluenzaDeaths].max() - df[df_InfluenzaDeaths].min()

[ ] print(InfluenzaDeaths_range)

12621.0
```

### Report:

The 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset shows a range score of 12621.0 for Influenza Deaths. This range, representing the difference between the maximum and minimum values, highlights the variability in Influenza Deaths across different demographic categories, underscoring the dataset's diverse nature.

### Task22:

To calculate the variance of Influenza Death spread of the dataset and report.

```
[ ] df_InfluenzaDeaths = 'Influenza Deaths'

[ ] InfluenzaDeaths_variance = df[df_InfluenzaDeaths].var()

[ ] print(InfluenzaDeaths_variance)

4752.604192810497
```

### Report:

The 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset indicates a variance score of 4752.60 for Influenza Deaths. This statistical measure quantifies the spread or dispersion of Influenza Deaths data, providing valuable insights into the dataset's variability across diverse demographic categories.

### Task23:

To calculate the Standard Deviation of Influenza Death , Deviation score the dataset and report.

```
[ ] df_InfluenzaDeaths = 'Influenza Deaths'

[ ] InfluenzaDeaths_std_dev = df[df_InfluenzaDeaths].std()

[ ] print(InfluenzaDeaths_std_dev)

68.93913397200821
```

### Report:

For the 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset, the Standard Deviation score is 68.94 for Influenza Deaths. This measure of dispersion showcases the extent to which Influenza Deaths deviate from the mean, providing a understanding of the dataset's variability across demographic categories.

### Task24:

To calculate the PMF of Influenza Death , the mass function value and report.

```
[ ] df_InfluenzaDeaths = 'Influenza Deaths'

[ ] pmf = df[df_InfluenzaDeaths].value_counts(normalize=True)

[ ] print(pmf)

0.000000    0.779817
3.525722    0.174348
10.000000    0.002728
11.000000    0.002411
1.000000    0.002346
...
1803.000000    0.000009
4545.000000    0.000009
187.000000    0.000009
448.000000    0.000009
468.000000    0.000009
Name: Influenza Deaths, Length: 379, dtype: float64
```



### Report:

The probability mass function for Influenza Deaths in the 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset provides insights into the likelihood of different death counts. The probabilities range from 0.779817 for a count of 0 to smaller probabilities for higher counts. This distribution reveals the statistical likelihood of encountering specific Influenza Death counts, contributing to a comprehensive understanding of the dataset's variability and potential patterns. It serves as a valuable tool for assessing the probability of observing different outcomes related to Influenza Deaths within diverse demographic categories.

### Task25:

To calculate the PDF of Influenza Death , the distribution influenza death function values and report.

```
[ ] from scipy.stats import norm  
  
[ ] df_InfluenzaDeaths = 'Influenza Deaths'  
  
[ ] pdf = norm.fit(df[df_InfluenzaDeaths])  
  
[ ] print(pdf)  
  
(3.5257216959855655, 68.93881304350126)
```

The probability density function (PDF) for 'Influenza Deaths' in the 'Provisional\_COVID-19\_Deaths\_by\_Sex\_and\_Age' dataset is defined by a mean of 3.53 and a standard deviation of 68.94. This PDF provides a statistical distribution of Influenza Deaths, indicating that the majority of values are expected to lie within one standard deviation of the mean. The mean and standard deviation serve as central

parameters, offering valuable insights into the dataset's central tendency and spread. Understanding the PDF aids in gauging the likelihood of observing specific Influenza Death values, contributing to interpretation of the dataset.

#### **References:**

- 1) <https://github.com/pydeveloperashish/BigMart-Sales-Prediction-With-Deployment/blob/main/BigMart%20Sales%20Prediction%20-%20Updated.ipynb>
- 2) <https://github.com/TahsinNakibTalukder/Python-Sales-data-analysis-using-pandas/blob/main/Sales%20analysis.ipynb>
- 3) Data\_Analytics\_lecture\_note  
<https://moodle.roehampton.ac.uk/mod/resource/view.php?id=1568523>
- 4) Machine\_Learning\_Lecture\_Note  
<https://moodle.roehampton.ac.uk/course/view.php?id=15797>
- 5) Mathematics for Data Science Lecture Note