PYTHON BASICS FOR DATA SCIENCE

S. No	Name of the Topic	Page
1	Install Jupyter Notebook Libraries (Seaborn, Matplotlib, Pandas, SciPy, Scikit-Learn, NumPy)	4
2	NumPy Learnings	4-9
	2.A. Data Types of a column or entire data tale	4
	2.B. Addition, Subtraction, Multiplication	5
	2.C. Maximum, Minimum, Mean, Median, Mode, Standard Deviation, Variance, Percentile (90^{th}, 15^{th}, 50^{th}) values.	6
	2.D. Create a Range. How to assess total number of values within the range	7
	2.E. Select specific 'Row' based on its Serial Number or Index Value	7
	2.F. Evaluate existing format of data (for example 'Even & Odd Numbers') in a column.	9
	2.G. Retrieve the numerical/String at specific position of a row under single column.	8
	2.H. Estimate all values which is greater than 5 or less than 2. Then multiply them with its square, cube.	8
	2.I. Calculate Sin, Cos, Square root values for given integers.	9
	2.J. Convert 1D array to 2D array.	9
	2.K. How to Merge 2 different labels listed in 1-Dimensional array.	9
	2.L. Change Dimensions of arrays from 1D to 3D (3 rows, 2 columns)	9
	3. Pandas Library	10-21
	3.A. Create any specific range of numbers. Then Retrieve the value located in specific row of a column.	10
	3.B. Pull entire 'Even' or 'Odd numbers' from above range (3.A). Then 'Convert' integers to Strings.	10
	3.C. Multiply 'Odd Numbers' in above list with its square.	11
	3.D How to 'Remove', 'Rename (Fill/Update)' Null values within given range.	12
	3.E. How to 'Replace' the values in a column.	12
	3.F. How to 'Rename' existent similar values/names in a single column or double column with different ones.	13
	3.G. Create a short data series and pull data from different columns.	14
	3.H. How to Load data from CSV/Excel/Text file and pull all column names in the dataset	15
	3.I. Show top 5 & top 2 (heads) rows along with bottom 5 & bottom 3 (tails) rows of the data table.	16
	3.J. Calculate the number of patients whose Age is greater than 25	18

	3.K. Set 'Adverse Event Onset Date' as Index column	19
	3.L. Select 'Specific Row(s)' by displaying 'Description of Adverse Event' associated with different 'VAERS-ID's	21
4.	**Statistics**	22-23
	A. Calculate Mean for given Numerical array	22
	B. Calculate Mean for a specific column in a data table.	23
	C. Calculate Covariance for 2 different variables in the data table.	23
5.	**Data Wrangling**	24-37
	5.A. Identify Unique Elements in a column	24
	5.B. Estimate 'Correlation' between 2 variables	25
	5.C. Select only 'Age, Number of days' (show only these 2) columns out of total 53 columns in a data frame	25
	5.D. Identify which columns have 'Missing/Null Values' in a data table	26
	5.E. Remove 'Missing/Null values' in 'Age' column. Display total rows in table without null values in 'Age' column.	27
	5.F. Detect 'Outliers' in 'Age' column	28
	5.G. How to 'Remove Outliers' in above scenario.	31
	5.H. 'Merge' multiple files (3 CSV) to create a single data frame and display box	32
	5.I. 'Group By/Sort' single or multiple columns	36
	5.J. 'Combine/Concatenate' 2 different data tables without any similarity between data/unrelated tables	37
	5.K. 'Combine/Merge' 2 data tables with at least single matching column (SQL 'Joins' concept)	37
6.	**Data Visualization with Python**	39-45
	6.A. Histogram/Bar Chart	39
	6.B. Scatter Plot	40
	6.C. Line Chart	41
	6.D. Pie Chart	42
	6.E. Box Plot	43
	6.F. 3D Visualization using 3 Axis	44
	6.G. Heatmap	45
7.	**Write a Demo Script**	
	Replace existing values in 'Number of Events' with defined parameters and create a pair plot visualization for script.	46

Python Basics

Enter jupyter notebook in command prompt to launch Jupyter Notebook.

1. Install Libraries in Jupyter Notebook

A. *Code to install & Import Seaborn:*

```
pip install seaborn
import seaborn as sns
import seaborn.objects as so
```

B. *Code to install & Import Matplotlib:*

```
pip install matplotlib
import matplotlib as mpl
import matplotlib.pyplot as plt
```

C. *Code to install & Import Pandas:*

```
! pip install pandas (it downloads numpy as default)
pip install openpyxl
import numpy as np
import pandas as pd
```

D. *Code to install & Import SciPy:*

```
pip install scipy
from scipy import linalg
```

E. *Code to install & Import Scikit-Learn:*

```
! pip install scikit-learn
from sklearn import datasets
```

F. *Install Numpy:*

```
import numpy as np
```

2. Numpy Library

2.A. View Data Type of a column or entire data tables:

```
df= pd.read_csv(r"Path of Data Tracker/Name of Tracker.csv")
df
```

i. Formula for entire data table data type:

```
print(df.dtypes)
```

ii. Formula to view Specific Column data type:

```
print(df['AGE_YRS'].dtypes)
```

```
print(df.dtypes)
```

```
VAERS_ID              int64
RECVDATE             object
STATE                object
AGE_YRS             float64
CAGE_YR             float64
CAGE_MO             float64
SEX                  object
RPT_DATE            float64
SYMPTOM_TEXT         object
DIED                 object
DATEDIED             object
L_THREAT             object
ER_VISIT            float64
HOSPITAL             object
HOSPDAYS            float64
X_STAY               object
DISABLE              object
RECOVD               object
VAX_DATE             object
ONSET_DATE           object
NUMDAYS             float64
LAB_DATA             object
V_ADMINBY            object
V_FUNDBY            float64
```

3560	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Computerised tomogram head	25.1	COV
3561	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Meningitis cryptococcal	25.1	COV
3562	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Vital signs measurement	25.1	COV
3563	2552104	06-01-2023	NaN	NaN	NaN	NaN	M	NaN	covid 19; This spontaneous case was reported b...	NaN	...	NaN	NaN	COV

3564 rows × 53 columns

```
In [10]: RawData['AGE_YRS'].dtypes
Out[10]: dtype('float64')
```

2.B. Addition, Subtraction, Multiplication: -

Define data frame:

A=np.array([1,2,3,4])

B=np.array([5,6,7,8])

Addition: np.add(A,B)

Subtraction: np.subtract(A,B)

Multiplication: np.multiply(A,B)

Division: np.divide(A,B)

```
In [36]: A=np.array([1,2,3,4])
         B=np.array([5,6,7,8])
         np.add(A,B)
Out[36]: array([ 6,  8, 10, 12])

In [37]: np.subtract(B,A)
Out[37]: array([4, 4, 4, 4])

In [38]: np.multiply(A,B)
Out[38]: array([ 5, 12, 21, 32])

In [39]: np.divide(A,B)
Out[39]: array([0.2       , 0.33333333, 0.42857143, 0.5       ])
```

2.C. Identify Maximum, Minimum, Mean, Median, Mode, Standard Deviation, Variance, Percentile (90^{th}, 15^{th}, 50^{th}) of integers

H=np.array([1,3,5,6,8,9])

np.max(H)

```
H=np.array([1,3,5,6,8,9])
```

```
In [35]: np.max(H)
Out[35]: 9

In [36]: np.min(H)
Out[36]: 1

In [37]: np.mean(H)
Out[37]: 5.333333333333333

In [40]: np.median(H)
Out[40]: 5.5

In [42]: np.std(H)
Out[42]: 2.748737083745107

In [43]: np.var(H)
Out[43]: 7.5555555555555545
```

```
In [44]: np.percentile(H,90)
Out[44]: 8.5

In [45]: np.percentile(H,50)
Out[45]: 5.5

In [46]: np.percentile(H,15)
Out[46]: 2.5
```

2.D. Create a range of numerical and assess total number of values existing in that range

Define data frame:

threeD_Array = np.arange(15)
threeD_Array

Pull total numbers within the range:

df=threeD_Array
df.shape

```
In [98]: threeD_Array = np.arange(15)
         threeD_Array
Out[98]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])

In [99]:
    df=threeD_Array
df.shape
Out[99]: (15,)
```

2.E. Select a specific row/ specific value in a row based on its Serial number or Index

Define Data frame:

L=np.array([21,11,50,8,-6,0,-2,73])

Pull value from Serial Number 1:

L[1]

```
L=np.array([21,11,50,8,-6,0,-2,73])

In [54]: L[0]
Out[54]: 21

In [55]: L[1]
Out[55]: 11

In [56]: L[2]
Out[56]: 50

In [57]: L[4]
Out[57]: -6
```

2.D. Identify existent format of data (for example 'Even & Odd Numbers') in a column.

H=np.array([1,3,5,6,8,9])

np.where(H%2==0, "Even","Odd")

```
In [30]: H=np.array([1,3,5,6,8,9])

In [31]: np.where(H%2==0, "Even","Odd")
Out[31]: array(['Odd', 'Odd', 'Odd', 'Even', 'Even', 'Odd'], dtype='<U4')
```

2.E. Identify any numerical which is greater than or less than predefined number. Then multiply the identified values with specific numbers

H=np.array([1,3,5,6,8,9])
conditionlist=([H>5,H<5])
choicelist=([H**2, H**3])
np.select(conditionlist,choicelist,default=H)

```
H=np.array([1,3,5,6,8,9])
```

Identify the numbers greater than and less than 5. Then multiply them with 2 (greater than 5) and 3 (less than 5) respectively. Then combine above 2 logics: -

```
conditionlist=([H>5,H<5])
choicelist=([H**2, H**3])
```

```
[34]: np.select(conditionlist,choicelist,default=H)
[34]: array([ 1, 27,  5, 36, 64, 81])
```

2.F. Calculate Sin, Cos, Square root values for given integers

np.cos(Integer)
np.sin(Integer)
np.sqrt(Integer)

```
In [6]: np.cos(16)
Out[6]: -0.9576594803233847

In [7]: np.sin(9)
Out[7]: 0.4121184852417566

In [16]: np.sqrt([49,36,64]),
Out[16]: (array([7., 6., 8.]),)
```

2.G. Convert 1D array to 2D array:

np.array([[25,3,101],[40,16,1]])

```
In [23]: Convert Columns into rows:-
         np.array([[25,3,101],[40,16,1]])
Out[23]: array([[ 25,   3, 101],
                [ 40,  16,   1]])
```

2.H. How to Merge 2 different arrays listed in 1-Dimension:

Define data frame (df) with 2 labels in 1D array
df.ravel()

```
df=np.array([[25,3,101],[40,16,1]])
df.ravel()
Out[24]: array([ 25,   3, 101,  40,  16,   1])
```

2.I. Change Dimensions of arrays from 1Dimensional to 3Dimensional (3 rows, 2 columns)

Since total number of numericals are 6, we could create either 3D (3 rows, 2 columns or 2 rows, 3columns array)

df=np. array([25, 3, 101, 40, 16, 1])

df=df.ravel()
df.reshape(3,2)

```
df=df.ravel()
df.reshape(3,2)
```

```
Out[29]: array([[ 25,   3],
                [101,  40],
                [ 16,   1]])
```

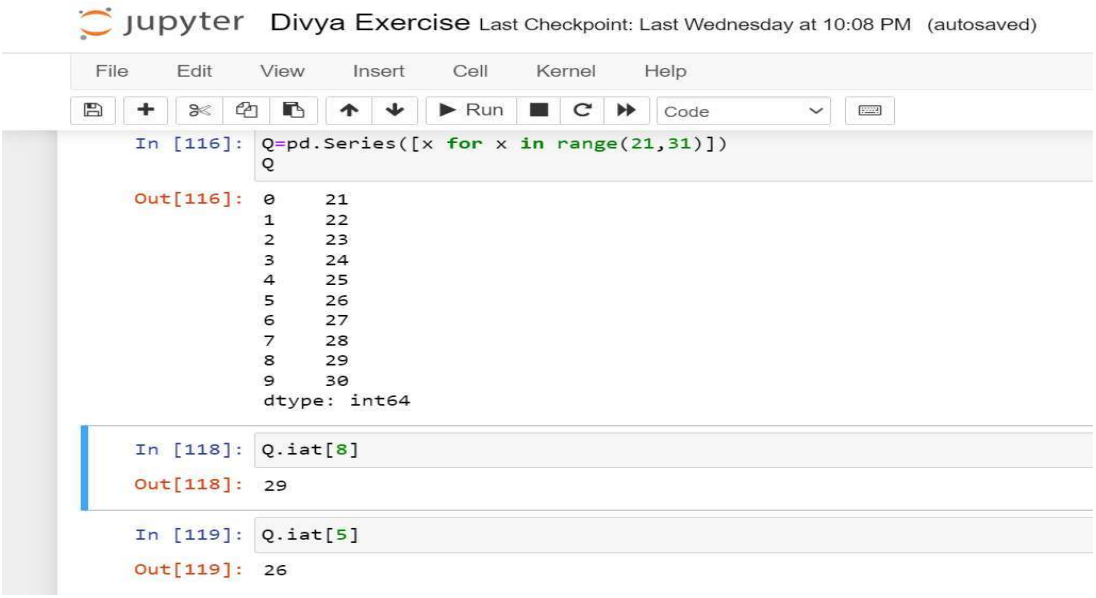
3. Pandas Library

3.A. Create a range of numbers with an upper limit and lower limit. Then retrieve data any specific row.

Q=pd.Series([x for x in range(21,31)])

Q

Q.iat[Position of row]



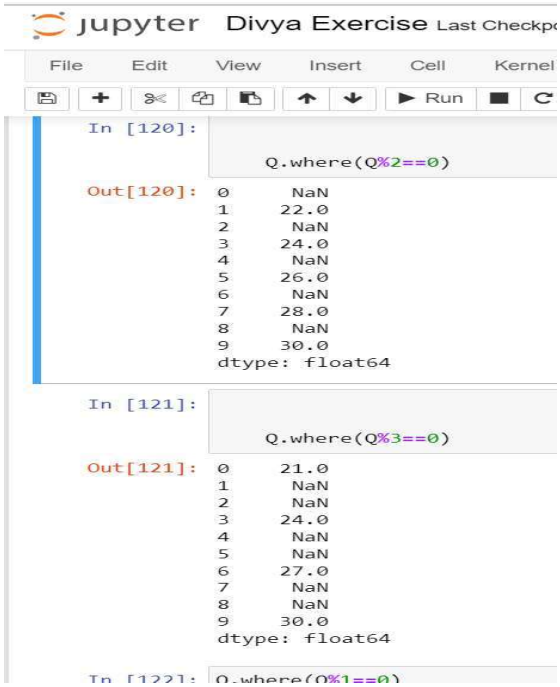
3.B. Pull entire list of either 'Even' or 'Odd numbers' from above range. Then 'Convert' integers to Strings.

To find only Even Numbers: Q.where(Q%2==0)

To find only Odd Numbers: Q.where(Q%3==0)

Convert all Odd number numerical to String of Odd Number: Q.where(Q%2==0,'Odd Number')

Convert all Even number numerical to String of Even Number: Q.where(Q%3==0,'Even Number')




```
In [124]: Q.where(Q%2==0,'Odd Number')
Out[124]: 0    Odd Number
          1            22
          2    Odd Number
          3            24
          4    Odd Number
          5            26
          6    Odd Number
          7            28
          8    Odd Number
          9            30
          dtype: object
In [125]: Q.where(Q%3==0,'Even Number')
Out[125]: 0            21
          1    Even Number
          2    Even Number
          3            24
          4    Even Number
          5    Even Number
          6            27
          7    Even Number
          8    Even Number
          9            30
          dtype: object
```

3.C. Multiply 'Odd Numbers' in above list with its square.

Define Data frame series as Q: Q=pd.Series([x for x in range(21,31)])

To find only Odd Numbers: Q.where(Q%3==0)

Q.where(Q%3==0,Q**2)

```
Q.where(Q%3==0,Q**2)
```

```
Out[126]: 0     21
          1    484
          2    529
          3     24
          4    625
          5    676
          6     27
          7    784
          8    841
          9     30
          dtype: int64
```

3.D. How to 'Remove' or 'Rename (Refill/Update)' Null values within given range.

Define Data frame series as Q: - Q=pd.Series([x for x in range(21,31)])

Remove all Null values: - Q.dropna()

Rename all Null values (Filling is best feature to update Null values instead of using Replace feature. After using this formula, there won't be any Null/NaN values in a column or entire data frame): - Q.fillna('Optimised')

```
In [131]: #Drop all NA values in data series and project only numericals without any null values:-

          Q.dropna()

Out[131]: 1    22.0
          3    24.0
          5    26.0
          7    28.0
          9    30.0
          dtype: float64

In [132]: #Fill all NA values with any Specific word and pull entire data series:-

          Q.fillna('Optimised')

Out[132]: 0    Optimised
          1         22.0
          2    Optimised
          3         24.0
          4    Optimised
          5         26.0
          6    Optimised
          7         28.0
          8    Optimised
          9         30.0
          dtype: object
```

3.E. How to 'Replace' the values in a column

import pandas as pd

Vaccinedose = pd.DataFrame(RawData['VAX_DOSE_SERIES'])

Vaccinedose.replace('UNK','0', regex=True)

```
Vaccinedose = pd.DataFrame(RawData['VAX_DOSE_SERIES'])
Vaccinedose.replace('UNK','0', regex=True)
```

Out[139]:

	VAX_DOSE_SERIES
0	0
1	0
2	0
3	3
4	4
...	...
3559	1
3560	2
3561	2
3562	2
3563	2

3564 rows × 1 columns

3.F. How to 'Rename' existent values/names with new values/names

i. Rename data within single column:

ModifiedData =pd.read_csv(r"Path for Data Source\Name of Data Source.csv",parse_dates=True)

ModifiedData

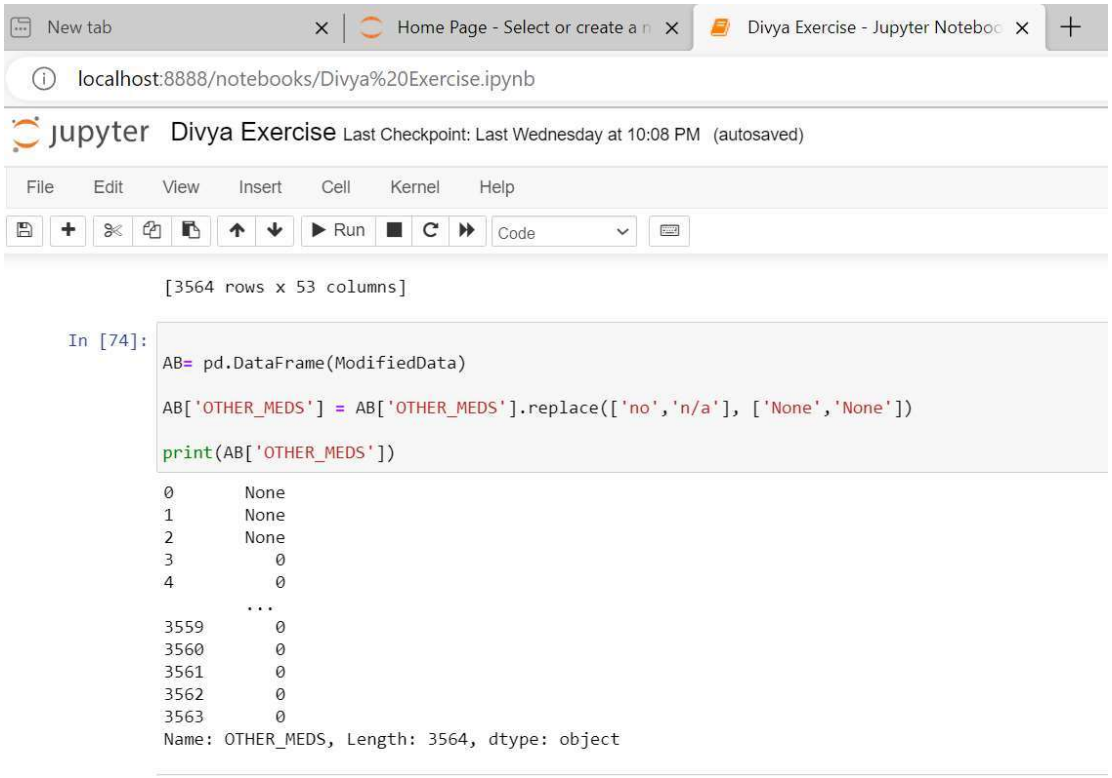
Change no, n/a data of 'Other Meds' column to 'None' to maintain uniformity of data which falls under same scenario throughout the column.

AB= pd.DataFrame(ModifiedData)

AB['OTHER_MEDS'] = AB['OTHER_MEDS'].replace(['no','n/a'], ['None','None'])

print(AB['OTHER_MEDS'])

ii. Rename data in 2 columns which has similar data in different rows:
Look for insights in this Google Link



3.G. Create a short data series and pull data from different columns.

Define Data frame series as Z: - Z = {'A':101, 'B':102, 'C':103,'F':119}

Single Column: Z['A']

Double columns: pd.Series(Z,index=['A','B'])

Existent and Non-existent columns: pd.Series(Z,index=['A','B','D'])

jupyter Divya Exercise Last Checkpoint: Last Wednesday at 10:08 PM (unsaved chan

File Edit View Insert Cell Kernel Help

Run Code

```
In [98]: #Define data series:-

             Z={'A':101, 'B':102, 'C':103,'F':119}

In [107]: Z['A']
Out[107]: 101

In [111]: #Pull specific list from data series:-

              pd.Series(Z,index=['A','B'])
Out[111]: A    101
          B    102
          dtype: int64

In [112]: #Try to Define new index which is not existent in data series:-

              pd.Series(Z,index=['A','B','D'])
Out[112]: A    101.0
          B    102.0
          D      NaN
          dtype: float64
```

3.H. Load data from CSV/Excel/Text file and pull all column names in the data set:

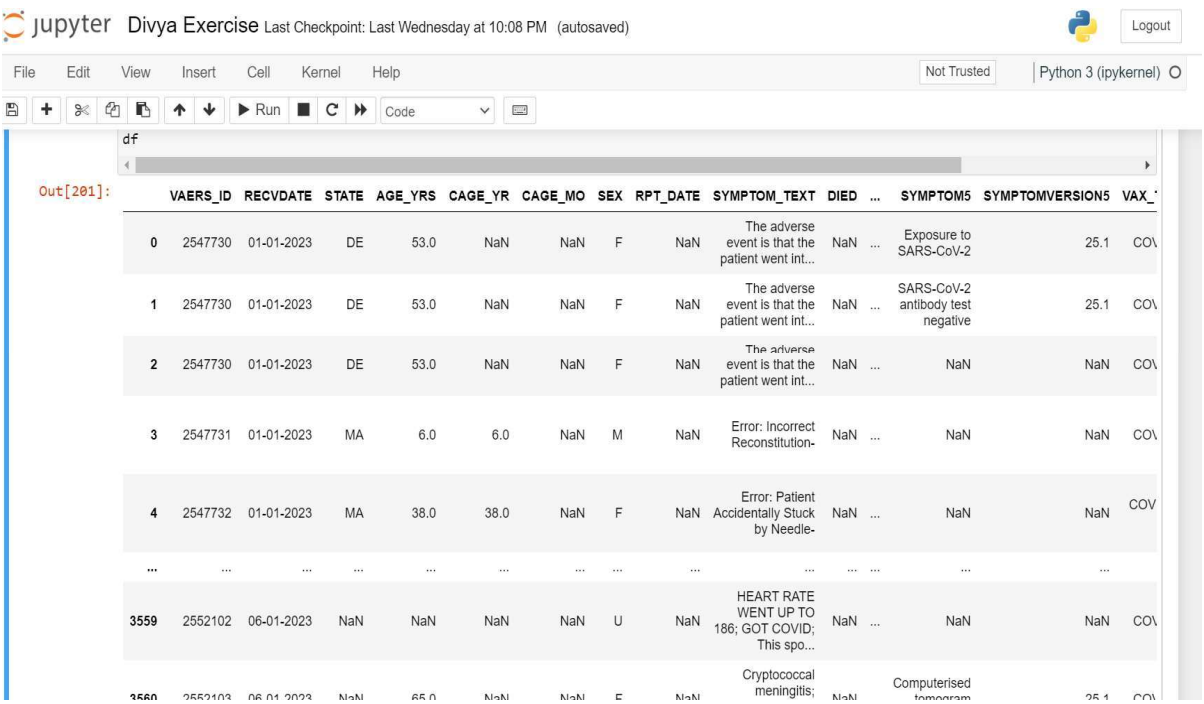
df=pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

df=pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

df=pd.read_text(r"Path for Data Source File\Name of the data Source File.txt",parse_dates=True)

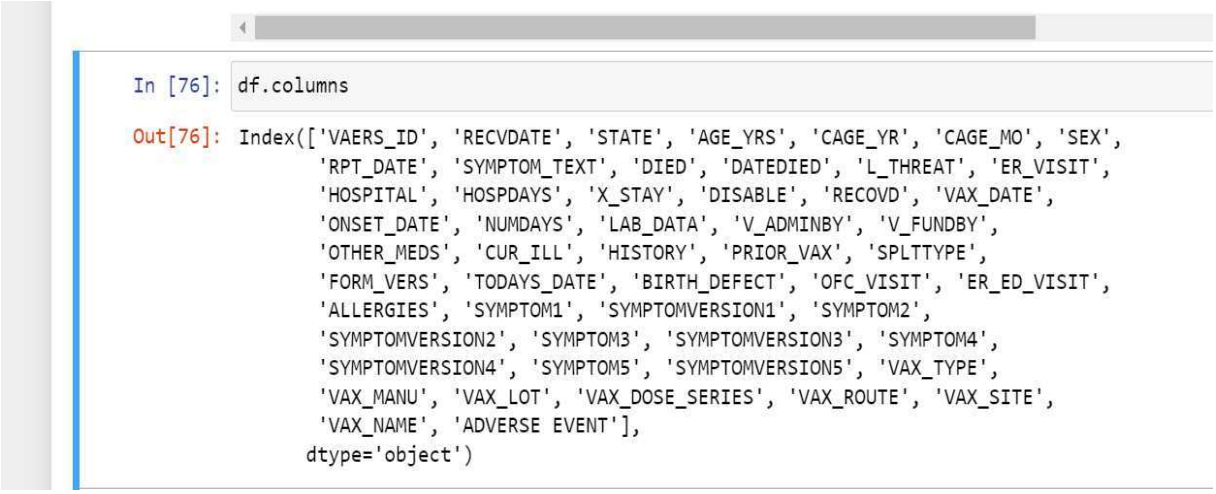
For defining Columns:

df.columns



	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION5	VAX_
0	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	Exposure to SARS-CoV-2	25.1	COV
1	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	SARS-CoV-2 antibody test negative	25.1	COV
2	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	NaN	NaN	COV
3	2547731	01-01-2023	MA	6.0	6.0	NaN	M	NaN	Error: Incorrect Reconstitution-	NaN	...	NaN	NaN	COV
4	2547732	01-01-2023	MA	38.0	38.0	NaN	F	NaN	Error: Patient Accidentally Stuck by Needle-	NaN	...	NaN	NaN	COV
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
3559	2552102	06-01-2023	NaN	NaN	NaN	NaN	U	NaN	HEART RATE WENT UP TO 186; GOT COVID; This spo...	NaN	...	NaN	NaN	COV
3560	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis;	NaN		Computerised tomogram	25.1	COV

Result after defining columns



```
In [76]: df.columns

Out[76]: Index(['VAERS_ID', 'RECVDATE', 'STATE', 'AGE_YRS', 'CAGE_YR', 'CAGE_MO', 'SEX',
                'RPT_DATE', 'SYMPTOM_TEXT', 'DIED', 'DATEDIED', 'L_THREAT', 'ER_VISIT',
                'HOSPITAL', 'HOSPDAYS', 'X_STAY', 'DISABLE', 'RECOVD', 'VAX_DATE',
                'ONSET_DATE', 'NUMDAYS', 'LAB_DATA', 'V_ADMINBY', 'V_FUNDBY',
                'OTHER_MEDS', 'CUR_ILL', 'HISTORY', 'PRIOR_VAX', 'SPLTTYPE',
                'FORM_VERS', 'TODAYS_DATE', 'BIRTH_DEFECT', 'OFC_VISIT', 'ER_ED_VISIT',
                'ALLERGIES', 'SYMPTOM1', 'SYMPTOMVERSION1', 'SYMPTOM2',
                'SYMPTOMVERSION2', 'SYMPTOM3', 'SYMPTOMVERSION3', 'SYMPTOM4',
                'SYMPTOMVERSION4', 'SYMPTOM5', 'SYMPTOMVERSION5', 'VAX_TYPE',
                'VAX_MANU', 'VAX_LOT', 'VAX_DOSE_SERIES', 'VAX_ROUTE', 'VAX_SITE',
                'VAX_NAME', 'ADVERSE EVENT'],
               dtype='object')
```

3.I. Show top 5 & top 2 (heads) and bottom 5 & bottom 2 (tails) rows in the datatable:

df=pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

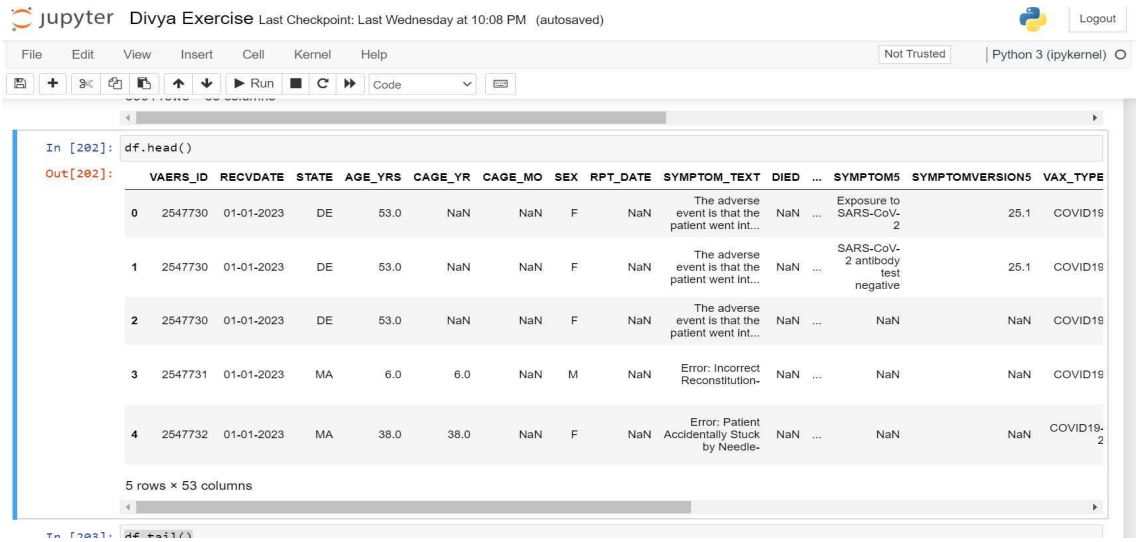
Top 5 rows would be pulled as default: df.head()

Bottom 5 rows would be pulled as default: df.tail()

Top 2 rows would be pulled: df.head(2)

Bottom 3 rows would be pulled: df.tail(3)

Head (top 5):-



jupyter Divya Exercise Last Checkpoint: Last Wednesday at 10:08 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3 (ipykernel)

Run Code

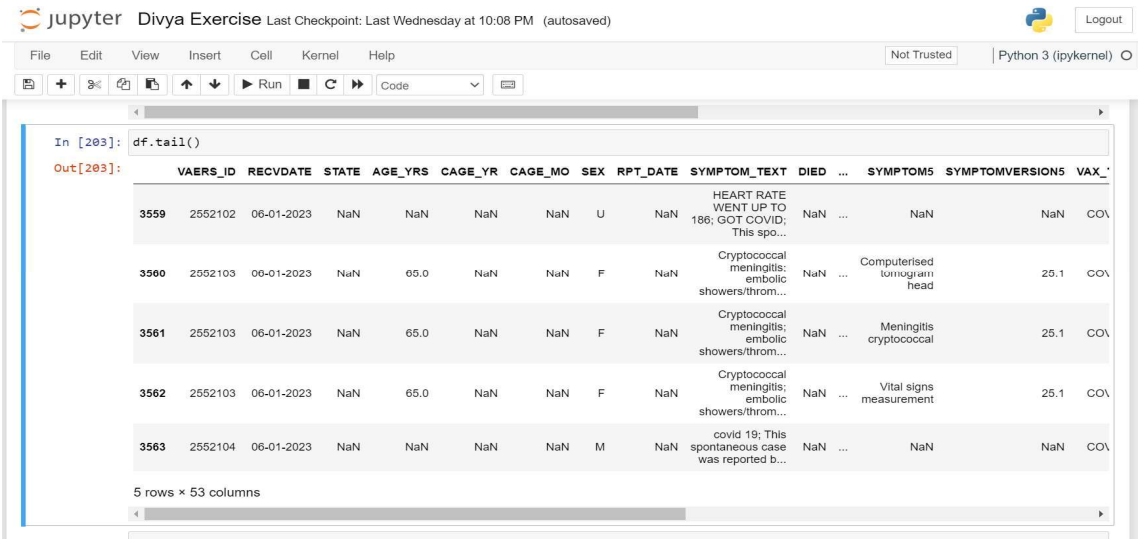
```
In [202]: df.head()
```

Out[202]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION5	VAX_TYPE
0	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	Exposure to SARS-CoV-2	25.1	COVID19
1	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	SARS-CoV-2 antibody test negative	25.1	COVID19
2	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	NaN	NaN	COVID19
3	2547731	01-01-2023	MA	6.0	6.0	NaN	M	NaN	Error: Incorrect Reconstitution-	NaN	...	NaN	NaN	COVID19
4	2547732	01-01-2023	MA	38.0	38.0	NaN	F	NaN	Error: Patient Accidentally Stuck by Needle-	NaN	...	NaN	NaN	COVID19-2

5 rows × 53 columns

Tail (bottom5)



jupyter Divya Exercise Last Checkpoint: Last Wednesday at 10:08 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3 (ipykernel)

Run Code

```
In [203]: df.tail()
```

Out[203]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION5	VAX_T
3559	2552102	06-01-2023	NaN	NaN	NaN	NaN	U	NaN	HEART RATE WENT UP TO 186; GOT COVID; This spo...	NaN	...	NaN	NaN	COV
3560	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Computerised tomogram head	25.1	COV
3561	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Meningitis cryptococcal	25.1	COV
3562	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Vital signs measurement	25.1	COV
3563	2552104	06-01-2023	NaN	NaN	NaN	NaN	M	NaN	covid 19; This spontaneous case was reported b...	NaN	...	NaN	NaN	COV

5 rows × 53 columns

Top 2(head) and Bottom 3 (tail):

```
In [204]: df.head(2)
```

Out[204]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION5	VAX_TYPE
0	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	Exposure to SARS-CoV-2	25.1	COVID19
1	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	SARS-CoV-2 antibody test negative	25.1	COVID19

2 rows × 53 columns

```
In [205]: df.tail(3)
```

Out[205]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION5	VAX_T
3561	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Meningitis cryptococcal	25.1	COV
3562	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Vital signs measurement	25.1	COV
3563	2552104	06-01-2023	NaN	NaN	NaN	NaN	M	NaN	covid 19; This spontaneous case was reported b...	NaN	...	NaN	NaN	COV

3 rows × 53 columns

3.J. Calculate the number of patients whose Age is greater than 25:

df=pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

Formula to calculate number of patients based on age:

df=df[df['AGE_YRS']>25.0]

df

Edge of this data frame shows 2600 rows x 53 columns in which 2600 is total number of patients with >25 yrs age.

```
In [209]: df=df[df['AGE_YRS']>25.0]
          df
```

Out[209]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION5	VAX_
0	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	Exposure to SARS-CoV-2	25.1	COV
1	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	SARS-CoV-2 antibody test negative	25.1	COV
2	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	NaN	NaN	COV
4	2547732	01-01-2023	MA	38.0	38.0	NaN	F	NaN	Error: Patient Accidentally Stuck by Needle-	NaN	...	NaN	NaN	COV
5	2547733	01-01-2023	CA	63.0	63.0	NaN	M	NaN	Error: Dose in Series Given Too Early-	NaN	...	NaN	NaN	COV
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3555	2552078	06-01-2023	AL	54.0	54.0	NaN	F	NaN	Irregular heartbeat , severe & frequent premat...	NaN	...	Heart rate irregular	25.1	COV
3556	2552078	06-01-2023	AL	54.0	54.0	NaN	F	NaN	Irregular heartbeat , severe & frequent premat...	NaN	...	NaN	NaN	COV
3560	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Computerised tomogram head	25.1	COV
3561	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Meningitis cryptococcal	25.1	COV
3562	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Vital signs measurement	25.1	COV

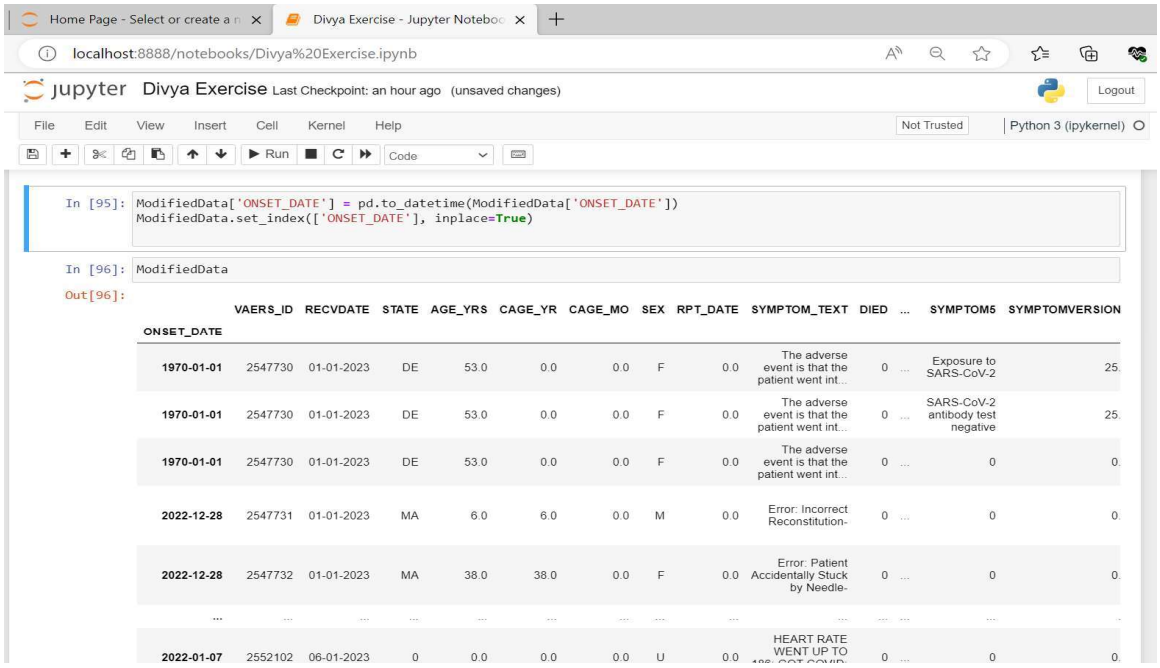
2600 rows × 53 columns

3.K. ***Set 'AE Onset date' as index***

i. set 'Onset Date' as Index:

```
ModifiedData['ONSET_DATE'] = pd.to_datetime(ModifiedData['ONSET_DATE'])

ModifiedData.set_index(['ONSET_DATE'], inplace=True)
```

ii. Then Execute the data Table name or with heads/tails to pull the final output
ModifiedData or ModifiedData.head()



Outcome with entire data table with 3564 rows and 52 Columns:

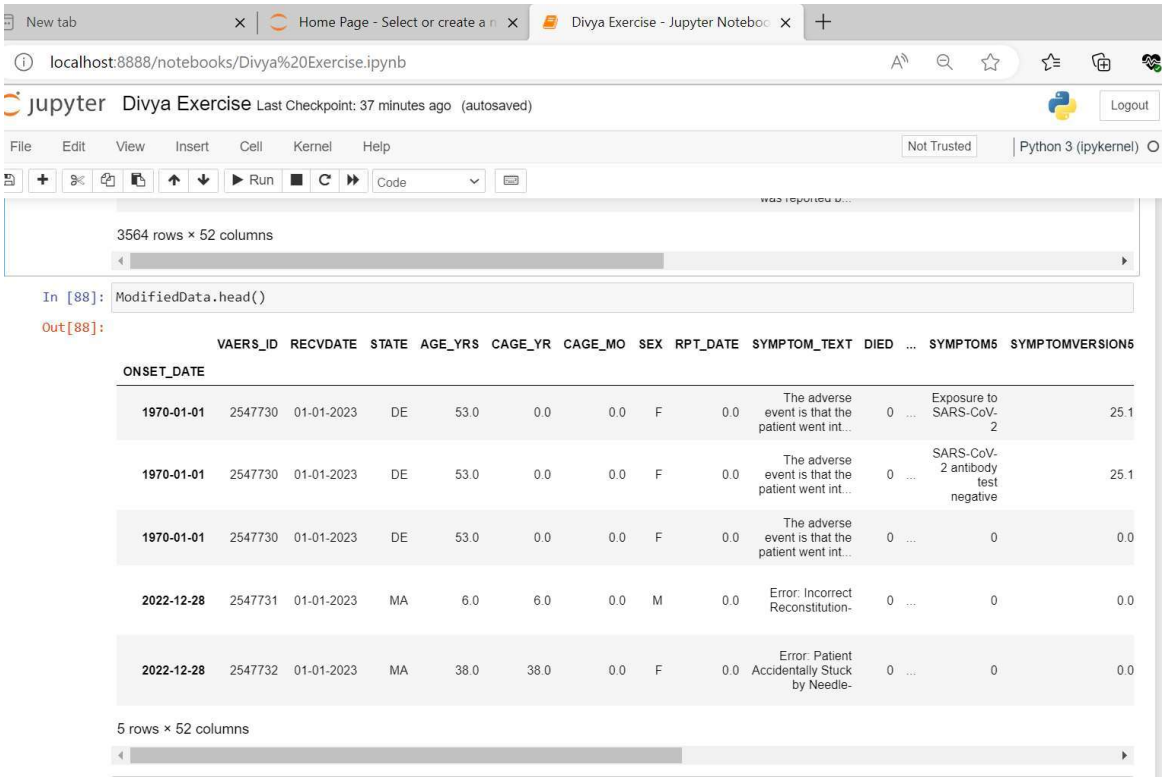
In [85]:
```
ModifiedData
```

Out[85]:

ONSET_DATE	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION
1970-01-01	2547730	01-01-2023	DE	53.0	0.0	0.0	F	0.0	The adverse event is that the patient went int...	0	...	Exposure to SARS-CoV-2	25
1970-01-01	2547730	01-01-2023	DE	53.0	0.0	0.0	F	0.0	The adverse event is that the patient went int...	0	...	SARS-CoV-2 antibody test negative	25
1970-01-01	2547730	01-01-2023	DE	53.0	0.0	0.0	F	0.0	The adverse event is that the patient went int...	0	...	0	0
2022-12-28	2547731	01-01-2023	MA	6.0	6.0	0.0	M	0.0	Error: Incorrect Reconstitution-	0	...	0	0
2022-12-28	2547732	01-01-2023	MA	38.0	38.0	0.0	F	0.0	Error: Patient Accidentally Stuck by Needle-	0	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2022-01-07	2552102	06-01-2023	0	0.0	0.0	0.0	U	0.0	HEART RATE WENT UP TO 188; GOT COVID; This spo...	0	...	0	0
2022-03-02	2552103	06-01-2023	0	65.0	0.0	0.0	F	0.0	Cryptococcal meningitis; embolic showers/throm...	0	...	Computerised tomogram head	25
2022-03-02	2552103	06-01-2023	0	65.0	0.0	0.0	F	0.0	Cryptococcal meningitis; embolic showers/throm...	0	...	Meningitis cryptococcal	25
2022-03-02	2552103	06-01-2023	0	65.0	0.0	0.0	F	0.0	Cryptococcal meningitis; embolic showers/throm...	0	...	Vital signs measurement	25
2023-02-01	2552104	06-01-2023	0	0.0	0.0	0.0	M	0.0	covid 19; This spontaneous case was reported b...	0	...	0	0

3564 rows × 52 columns

Pull top 5 rows of entire data table (5Rowsx15columns)



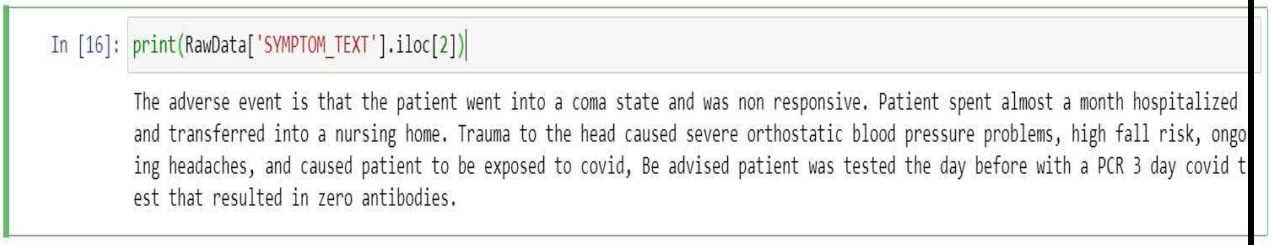
3.I. Select specific row(s) by displaying 'Description of Adverse Event' in row 2 of data table:

RawData =pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

RawData

i. Select single specific row in data table:

print(RawData['SYMPTOM_TEXT'].iloc[2])



ii. Select multiple rows in a data table within expected range:

print(RawData['SYMPTOM_TEXT'].iloc[1:8])

print(RawData['SYMPTOM_TEXT'].iloc[15:28])

```
In [19]: print(RawData['SYMPTOM_TEXT'].iloc[1:8])

1    The adverse event is that the patient went int...
2    The adverse event is that the patient went int...
3                    Error: Incorrect Reconstitution-
4         Error: Patient Accidentally Stuck by Needle-
5              Error: Dose in Series Given Too Early-
6    Systemic: Confusion-Mild, Systemic: Fainting /...
7            Systemic: Fainting / Unresponsive-Severe
Name: SYMPTOM_TEXT, dtype: object
```

```
In [21]: print(RawData['SYMPTOM_TEXT'].iloc[15:28])

15              Error: Dose in Series Given Too Early-
16    Upon receiving the Covid booster I started wit...
17    Upon receiving the Covid booster I started wit...
18    Upon receiving the Covid booster I started wit...
19    Upon receiving the Covid booster I started wit...
20    Upon receiving the Covid booster I started wit...
21    Upon receiving the Covid booster I started wit...
22    Cold like symptoms, fever 101.3 runny nose los...
23    Cold like symptoms, fever 101.3 runny nose los...
24    Cold like symptoms, fever 101.3 runny nose los...
25    Cold like symptoms, fever 101.3 runny nose los...
26    Cold like symptoms, fever 101.3 runny nose los...
27    Cold like symptoms, fever 101.3 runny nose los...
Name: SYMPTOM_TEXT, dtype: object
```

4. Statistics:

import statistics

4.A. Calculate Mean for given numericals:-

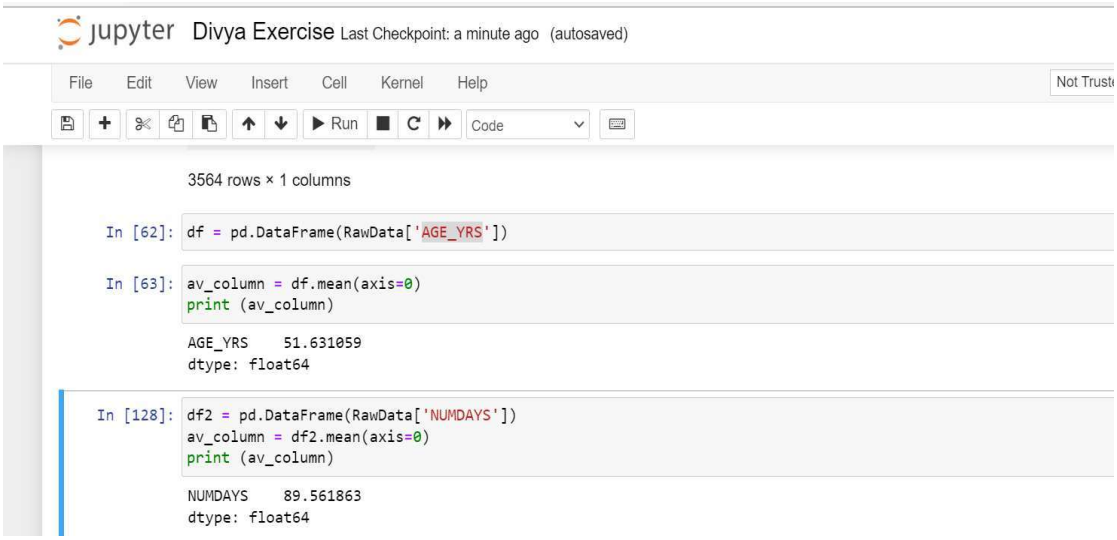
statistics.mean([5, 12, 26, 2])

```
In [45]: statistics.mean([5, 12, 26, 2])

Out[45]: 11.25
```

4.B. Calculate Mean for a specific column in the data table:-

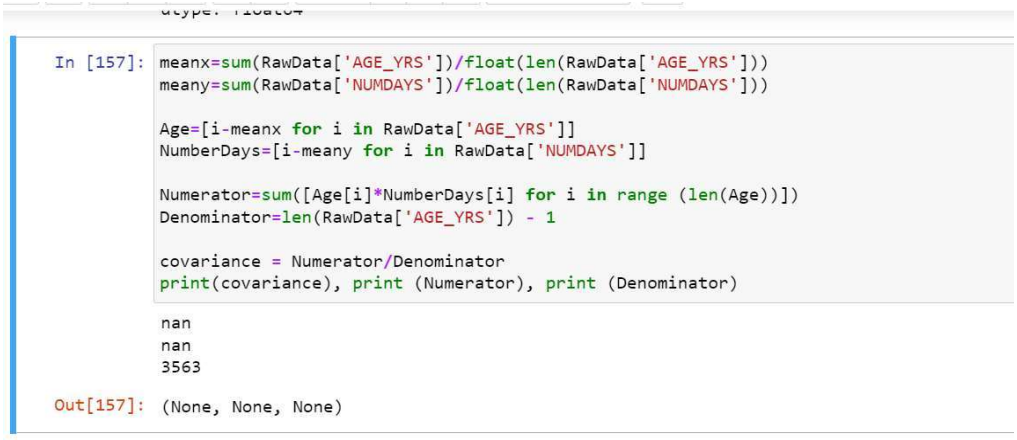
RawData=pd.read_csv(r"Source Tracker Path/Source Tracker Name.csv")
RawData

df = pd.DataFrame(RawData['NUMDAYS'])
av_column = df.mean(axis=0)
print (av_column)



4.C. Calculate Covariance for 2 different variables in the data table: -

```
meanx=sum(RawData['AGE_YRS'])/float(len(RawData['AGE_YRS']))
meany=sum(RawData['NUMDAYS'])/float(len(RawData['NUMDAYS']))

Age=[i-meanx for i in RawData['AGE_YRS']]
NumberDays=[i-meany for i in RawData['NUMDAYS']]

Numerator=sum([Age[i]*NumberDays[i] for i in range (len(Age))])
Denominator=len(RawData['AGE_YRS']) - 1

covariance = Numerator/Denominator
print(covariance), print (Numerator), print (Denominator)
```



4.D. Calculate Standard Deviation for 2 different variables in the data table

```
StandarddeviationX=RawData[['AGE_YRS']].std(ddof=0)
StandarddeviationX
```

```
In [175]: StandarddeviationX=RawData[['AGE_YRS']].std(ddof=0)
          StandarddeviationX

Out[175]: AGE_YRS    23.50829
          dtype: float64

In [176]: StandarddeviationY=RawData[['NUMDAYS']].std(ddof=0)
          StandarddeviationY

Out[176]: NUMDAYS    161.528879
          dtype: float64
```

4.E. Calculate Standard Deviation for 2 different variables in the data table

Correlation=covariance/StandarddeviationX*StandarddeviationY
print(Correlation)

```
In [178]: Correlation=covariance/StandarddeviationX*StandarddeviationY
          print(Correlation)

          AGE_YRS    NaN
          NUMDAYS    NaN
          dtype: float64
```

5. Data Wrangling

5.A. Identify unique elements in a column:

RawData =pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

RawData['VAX_MANU'].unique()

```
In [22]: RawData['VAX_MANU'].unique()

Out[22]: array(['JANSSEN', 'PFIZER\\BIONTECH', 'MODERNA', 'UNKNOWN MANUFACTURER',
                'BAVARIAN NORDIC', 'SEQIRUS, INC.', 'SANOFI PASTEUR',
                'GLAXOSMITHKLINE BIOLOGICALS', 'NOVAVAX', 'MERCK & CO. INC.',
                'PFIZER\\WYETH', 'MEDIMMUNE VACCINES, INC.',
                'PROTEIN SCIENCES CORPORATION'], dtype=object)
```

5.B. Estimate 'Correlation' between two variables:

Frame=pd.read_excel(r"Path of Data Source/Data Source Name.xlsx")

Frame[['2021Salary','2022Salary']].corr()

```
Frame
```

Out[5]:

	EmployeeID	Department	2021Salary	2022Salary
0	1	Finance	40000	100000
1	2	R&D	55000	80000
2	3	Sales	50000	50000
3	4	HR	42000	80000
4	5	General Maintainace	35000	40000
5	6	IT	80000	40000
6	7	Other	100000	35000

```
In [50]: Frame[['2021Salary','2022Salary']].corr()
```

Out[50]:

	2021Salary	2022Salary
2021Salary	1.000000	-0.558372
2022Salary	-0.558372	1.000000

5.C. Show only 'Age, Number of days' columns out of total 53 columns in data frame:

RawData =pd.read_excel(r"Path of Data Source/Data Source Name.xlsx")

i. Selecting 2 columns with Title & Data Frame Name

```
ModifiedData=print("Select specific columns:")
ModifiedData=print(RawData[['AGE_YRS', 'NUMDAYS']])
```

ii. Selecting 3 columns without Title & Data Frame Name:

```
print(RawData[['LAB_DATA', 'OTHER_MEDS','CUR_ILL']])
```

3564 rows × 53 columns

```
In [66]: ModifiedData=print("Select specific columns:")
         ModifiedData=print(RawData[['AGE_YRS', 'NUMDAYS']])

Select specific columns:
      AGE_YRS  NUMDAYS
0        53.0      NaN
1        53.0      NaN
2        53.0      NaN
3         6.0      0.0
4        38.0      0.0
...       ...      ...
3559      NaN      NaN
3560     65.0     94.0
3561     65.0     94.0
3562     65.0     94.0
3563      NaN      NaN

[3564 rows x 2 columns]
```

```
In [67]: print(RawData[['LAB_DATA', 'OTHER_MEDS','CUR_ILL']])

                                                  LAB_DATA OTHER_MEDS  \
0     Hospilization 4/17/2021 - Lab work, MRI, Catsc...          no
1     Hospilization 4/17/2021 - Lab work, MRI, Catsc...          no
2     Hospilization 4/17/2021 - Lab work, MRI, Catsc...          no
3                                                   NaN        NaN
4                                                   NaN        NaN
...                                                 ...        ...
3559  Test Name: HEART RATE; Result Unstructured Dat...        NaN
3560  Test Name: Chest x-ray; Result Unstructured Da...        NaN
3561  Test Name: Chest x-ray; Result Unstructured Da...        NaN
3562  Test Name: Chest x-ray; Result Unstructured Da...        NaN
3563                                                NaN        NaN

          CUR_ILL
0        diabetic
1        diabetic
2        diabetic
3             NaN
4             NaN
...           ...
3559          NaN
3560  Hypertension
3561  Hypertension
3562  Hypertension
3563          NaN

[3564 rows x 3 columns]
```

5.D. identify which columns has 'Null/Missing values':

RawData =pd.read_excel(r"Path of Data Source/Data Source Name.xlsx")

RawData.isna().any()

Output of 'True': Missing values exists

Output of 'False': No null or Missing values exists

```
In [52]: RawData.isna().any()

Out[52]: VAERS_ID           False
         RECVDATE           False
         STATE               True
         AGE_YRS             True
         CAGE_YR             True
         CAGE_MO             True
         SEX                False
         RPT_DATE            True
         SYMPTOM_TEXT        True
         DIED                True
         DATEDIED            True
         L_THREAT            True
         ER_VISIT            True
         HOSPITAL            True
         HOSPDAYS            True
         X_STAY              True
         DISABLE             True
         RECOVD              True
         VAX_DATE            True
         ONSET_DATE          True
         NUMDAYS             True
         LAB_DATA            True
         V_ADMINBY          False
         V_FUNDBY            True
         OTHER_MEDS          True
         CUR_ILL             True
         HISTORY             True
         PRIOR_VAX           True
         SPLTTYPE            True
         FORM_VERS          False
         TODAYS_DATE         True
         BIRTH_DEFECT        True
         OFC_VISIT           True
         ER_ED_VISIT         True
         ALLERGIES           True
         SYMPTOM1           False
         SYMPTOMVERSION1    False
         SYMPTOM2            True
         SYMPTOMVERSION2     True
         SYMPTOM3            True
         SYMPTOMVERSION3     True
         SYMPTOM4            True
         SYMPTOMVERSION4     True
         SYMPTOM5            True
         SYMPTOMVERSION5     True
         VAX_TYPE           False
         VAX_MANU           False
         VAX_LOT             True
         VAX_DOSE_SERIES     True
         VAX_ROUTE           True
         VAX_SITE            True
         VAX_NAME           False
         ADVERSE_EVENT      False
         dtype: bool
```

5.E. Filter 'Missing/Null values' in 'Age' column and display rest of the rows without null values:

RawData =pd.read_excel(r"Path of Data Source/Data Source Name.xlsx")

```
bool_series = pd.notnull(RawData['AGE_YRS'])
RawData[bool_series]
```

Result = 3136 rows has data without missing/null values out of 3564 rows. It means 428 null values are existent under 'Age' column

```
In [79]: bool_series = pd.notnull(RawData['AGE_YRS'])
         RawData[bool_series]
```

Out[79]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	SYMPTOM5	SYMPTOMVERSION5	VAX_
0	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	Exposure to SARS-CoV-2	25	COV
1	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	SARS-CoV-2 antibody test negative	25	COV
2	2547730	01-01-2023	DE	53.0	NaN	NaN	F	NaN	The adverse event is that the patient went int...	NaN	...	NaN	NaN	COV
3	2547731	01-01-2023	MA	6.0	6.0	NaN	M	NaN	Error: Incorrect Reconstitution-	NaN	...	NaN	NaN	COV
4	2547732	01-01-2023	MA	38.0	38.0	NaN	F	NaN	Error: Patient Accidentally Stuck by Needle-	NaN	...	NaN	NaN	COV
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
3555	2552078	06-01-2023	AL	54.0	54.0	NaN	F	NaN	Irregular heartbeat , severe & frequent premat...	NaN	...	Heart rate irregular	25	COV
3556	2552078	06-01-2023	AL	54.0	54.0	NaN	F	NaN	Irregular heartbeat , severe & frequent premat...	NaN	...	NaN	NaN	COV
3560	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Computerised tomogram head	25	COV
3561	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Meningitis cryptococcal	25	COV
3562	2552103	06-01-2023	NaN	65.0	NaN	NaN	F	NaN	Cryptococcal meningitis; embolic showers/throm...	NaN	...	Vital signs measurement	25	COV

3136 rows × 53 columns

5.F. Detect 'Outliers' in a dataset and remove them:

DataFrame1=pd.read_excel(r"Path of Data Source/Data Source Name.xlsx")

i. Create Boxplot to check outliers:

```
import seaborn as sns

sns.boxplot(x=DataFrame1['AGE_YRS'])
```

ii. Calculate Interquartile Range (IQR) to know distance between each quartile:

```
q1=DataFrame1['AGE_YRS'].quantile(0.25)
q3=DataFrame1['AGE_YRS'].quantile(0.75)

IQR=q3-q1
print(IQR)
```

iii. Calculate upper limit and lower limit for outlier values:

```
low_lim = q1 - 1.5 * IQR
up_lim = q3 + 1.5 * IQR
print('low_limit is', low_lim)
print('up_limit is', up_lim)
```

iv. Calculate existence of outlier numbers within data set of given column:

```
outlier =[]
for x in DataFrame1['AGE_YRS']:
   if ((x> up_lim) or (x<low_lim)):
       outlier.append(x)
print(' outlier in the dataset is', outlier)
```

v. Numerical Value calculation for outlier:

```
from scipy import stats
IQR = stats.iqr(DataFrame1['AGE_YRS'], interpolation = 'midpoint')
IQR
```

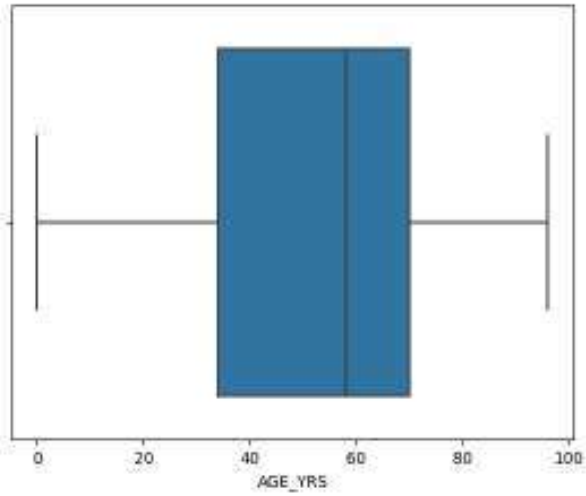
vi. Plot with an outlier value

```
sns.boxplot(DataFrame1['AGE_YRS'])
```

Result: In this scenario, there is no outlier existed in ‘Age’ column. Hence outlier output value was 0.

In [111]:
```
import seaborn as sns
sns.boxplot(x=DataFrame1['AGE_YRS'])
```

Out[111]: <Axes: xlabel='AGE_YRS'>



In [104]:
```
q1=DataFrame1['AGE_YRS'].quantile(0.25)
q3=DataFrame1['AGE_YRS'].quantile(0.75)

IQR=q3-q1
print(IQR)
```

```
36.0
```

In [112]:
```
low_lim = q1 - 1.5 * IQR
up_lim = q3 + 1.5 * IQR
print('low_limit is', low_lim)
print('up_limit is', up_lim)
```

```
low_limit is -20.0
up_limit is 124.0
```

In [114]:
```
outlier =[]
for x in DataFrame1['AGE_YRS']:
    if ((x> up_lim) or (x<low_lim)):
        outlier.append(x)
print(' outlier in the dataset is', outlier)
```

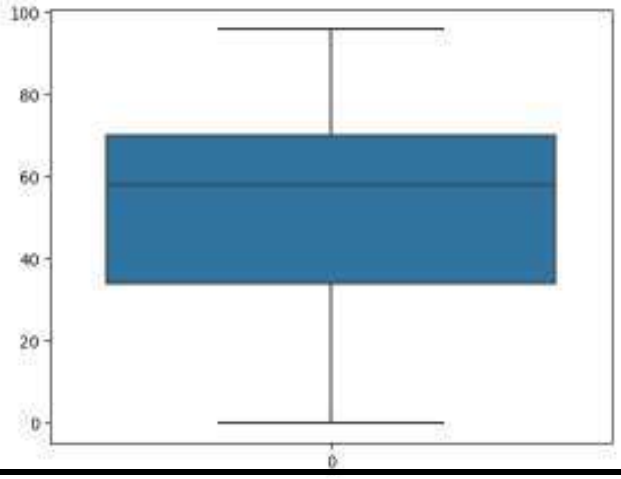
```
 outlier in the dataset is []
```

In [116]:
```
from scipy import stats
IQR = stats.iqr(DataFrame1['AGE_YRS'], interpolation = 'midpoint')
IQR
```

Out[116]: nan

In [117]:
```
sns.boxplot(DataFrame1['AGE_YRS'])
```

Out[117]: <Axes: >



5.G. How to 'Remove Outliers' in above scenario:

DataFrame1=pd.read_excel(r"Path of Data Source/Data Source Name.xlsx")

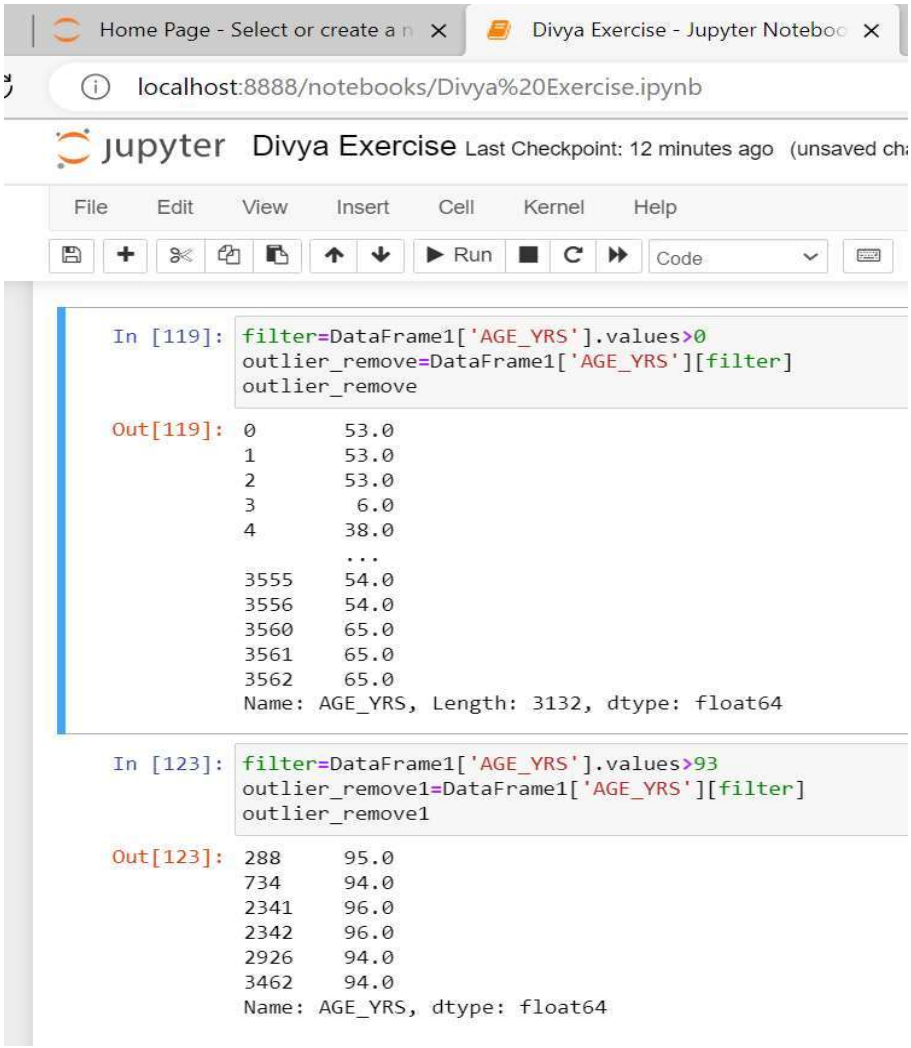
Result: In above scenario, there is no outlier existed in 'Age' column. Hence outlier output value was 0.

i. Total number of rows after Removal of outliers:

```
filter=DataFrame1['AGE_YRS'].values>0
outlier_remove=DataFrame1['AGE_YRS'][filter]
outlier_remove
```

ii. Let's consider outlier value as 93 as example and outliers from data:

```
filter=DataFrame1['AGE_YRS'].values>93
outlier_remove=DataFrame1['AGE_YRS'][filter]
outlier_remove
```



5.H. 'Merge' multiple files (3 CSV) to create a single data frame and display box plots:

```
from sklearn.datasets import load_diabetes

dataset = load_diabetes()

dataset

dataset.data

dataset.target

dataset['feature_names']

import pandas as pd

import numpy as np
```

Create data frame with required rows & columns: -

```
Output= pd.DataFrame(data=np.c_[dataset['data'],dataset['target']],
columns=dataset['feature_names']+['target'])

Output
```

Check if any null/missing values exist: -

```
Output.isnull().any()
```

Create box plot for all columns in the data frame:

```
%matplotlib inline

import matplotlib.pyplot as plt


for column in Output:

    plt.figure()

    Output.boxplot([column])
```

jupyter Divya Exercise Last Checkpoint: 2 hours ago (autosaved)

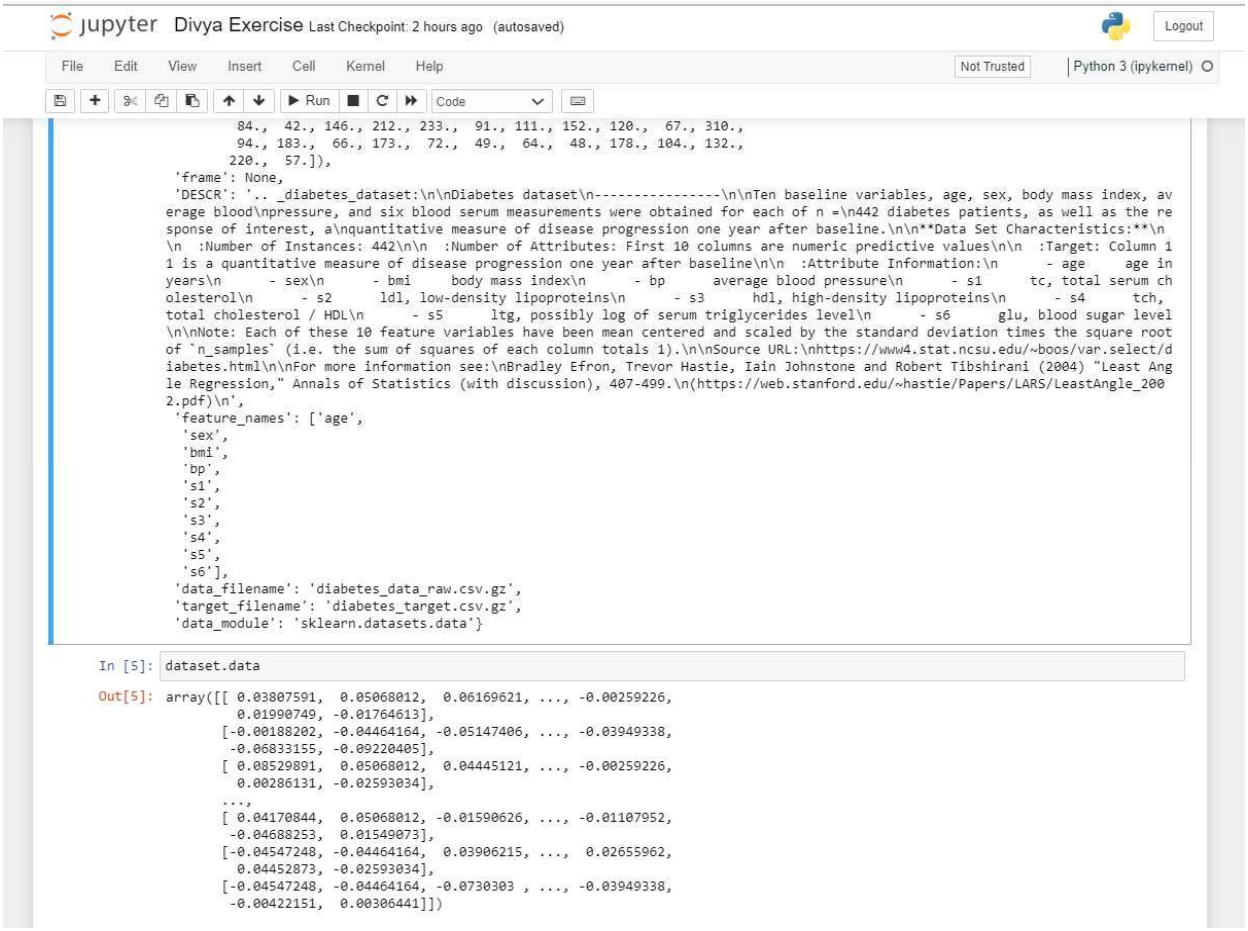
File Edit View Insert Cell Kernel Help

Run Code

In [1]:
```
from sklearn.datasets import load_diabetes
```

In [3]:
```
dataset = load_diabetes()
dataset
```

Out[3]:
```
{'data': array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
         0.01990749, -0.01764613],
       [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
        -0.06833155, -0.09220405],
       [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
         0.00286131, -0.02593034],
       ...,
       [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
        -0.04688253,  0.01549073],
       [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
         0.04452873, -0.02593034],
       [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
        -0.00422151,  0.00306441]]),
 'target': array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 310., 101.,
        69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
        68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
        87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
       259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
       128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
       150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
       200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
        42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
        83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
       104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
       173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
       107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
        60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
       197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
        59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
       237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
       143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
       142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
        77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
        78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
       154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
        71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
       150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
       145.,  74.,  45., 115., 264.,  87., 202., 127., 182., 241.,  66.,
        94., 283.,  64., 102., 200., 265.,  94., 230., 181., 156., 233.,
        60., 219.,  80.,  68., 332., 248.,  84., 200.,  55.,  85.,  89.,
        31., 129.,  83., 275.,  65., 198., 236., 253., 124.,  44., 172.,
       114., 142., 109., 180., 144., 163., 147.,  97., 220., 190., 109.,
```


jupyter Divya Exercise Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Not Trusted
Python 3 (ipykernel)
Run
Code
84., 42., 146., 212., 233., 91., 111., 152., 120., 67., 310.,
94., 183., 66., 173., 72., 49., 64., 48., 178., 104., 132.,
220., 57.]),
'frame': None,
'DESCR': '.. _diabetes_dataset:\n\nDiabetes dataset\n----------\n\nTen baseline variables, age, sex, body mass index, average blood\npressure, and six blood serum measurements were obtained for each of n =\n442 diabetes patients, as well as the response of interest, a\nquantitative measure of disease progression one year after baseline.\n\n**Data Set Characteristics:**\n\n :Number of Instances: 442\n\n :Number of Attributes: First 10 columns are numeric predictive values\n\n :Target: Column 11 is a quantitative measure of disease progression one year after baseline\n\n :Attribute Information:\n - age age in years\n - sex\n - bmi body mass index\n - bp average blood pressure\n - s1 tc, total serum cholesterol\n - s2 ldl, low-density lipoproteins\n - s3 hdl, high-density lipoproteins\n - s4 tch, total cholesterol / HDL\n - s5 ltg, possibly log of serum triglycerides level\n - s6 glu, blood sugar level\n\nNote: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times the square root of `n_samples` (i.e. the sum of squares of each column totals 1).\n\nSource URL:\nhttps://www4.stat.ncsu.edu/~boos/var.select/diabetes.html\n\nFor more information see:\nBradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regression," Annals of Statistics (with discussion), 407-499.\n(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)\n',
'feature_names': ['age',
'sex',
'bmi',
'bp',
's1',
's2',
's3',
's4',
's5',
's6'],
'data_filename': 'diabetes_data_raw.csv.gz',
'target_filename': 'diabetes_target.csv.gz',
'data_module': 'sklearn.datasets.data'}
In [5]:
dataset.data
Out[5]:
array([[0.03807591, 0.05068012, 0.06169621, ..., -0.00259226,
0.01990749, -0.01764613],
[-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
-0.06833155, -0.09220405],
[0.08529891, 0.05068012, 0.04445121, ..., -0.00259226,
0.00286131, -0.02593034],
...,
[0.04170844, 0.05068012, -0.01590626, ..., -0.01107952,
-0.04688253, 0.01549073],
[-0.04547248, -0.04464164, 0.03906215, ..., 0.02655962,
0.04452873, -0.02593034],
[-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
-0.00422151, 0.00306441]])

jupyter Divya Exercise Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Not Trusted

Run Code

In [7]:
```
dataset['feature_names']
```

Out[7]:
```
['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']
```

In [8]:
```
import pandas as pd
import numpy as np
```

In [10]:
```
Output= pd.DataFrame(data=np.c_[dataset['data'],dataset['target']], columns=dataset['feature_names']+['target'])
Output
```

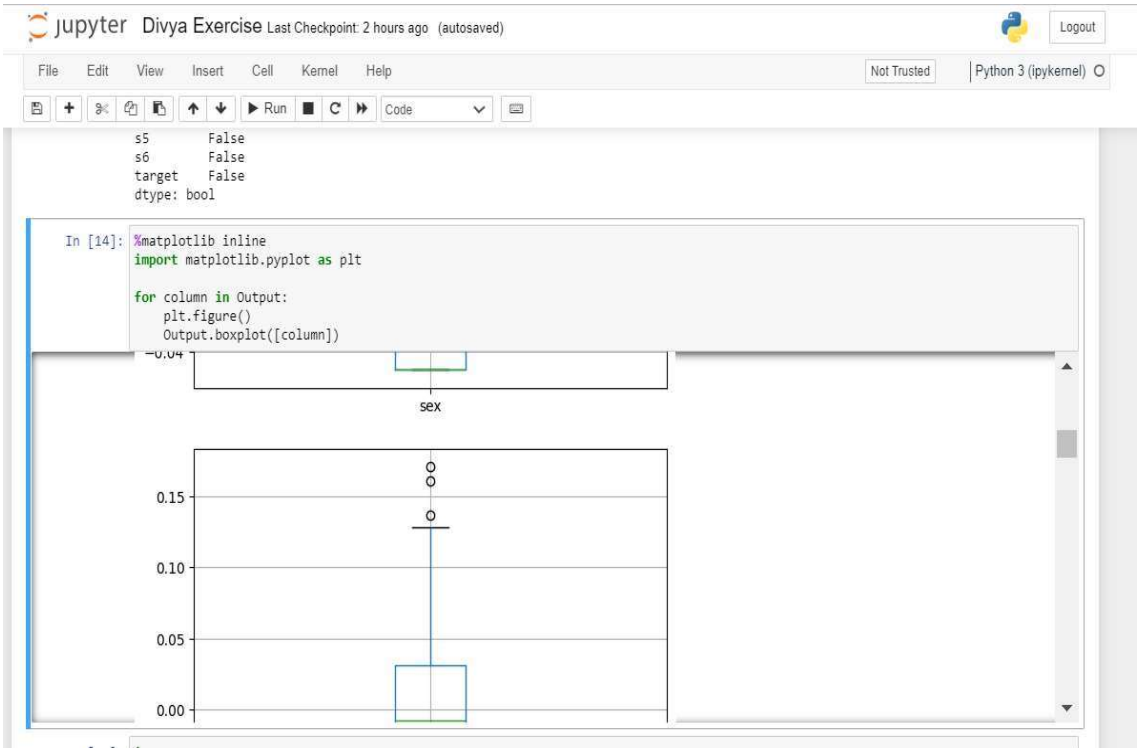
Out[10]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0
...	...	...	...	...	...	...	...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018114	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.046883	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044529	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.004222	0.003064	57.0

442 rows × 11 columns

In [12]:
```
Output.isnull().any()
```

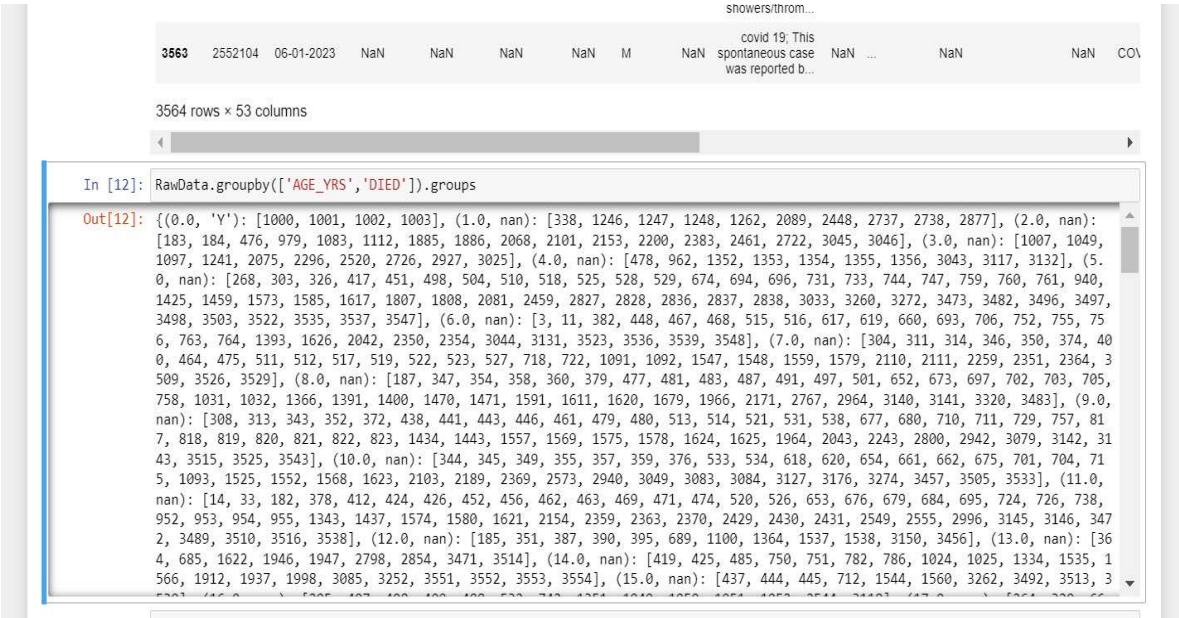
Out[12]:
```
age       False
sex       False
bmi       False
bp        False
s1        False
s2        False
s3        False
s4        False
s5        False
s6        False
target    False
dtype: bool
```



5.I. 'Group By' columns data.

RawData =pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

RawData.groupby(['AGE_YRS','DIED']).groups



5.J. 'Concatenate' 2 different data tables which are unrelated and without any similarity between data/entirely different data:

RawData =pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

Table1=pd.DataFrame(RawData)

Table2=pd.DataFrame(DiabetesOutput)

pd.concat([RawData,DiabetesOutput],axis=1)

```
In [28]: pd.concat([RawData,DiabetesOutput],axis=1)
```

Out[28]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	sex	bmi	bp	s1
0	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	0.050680	0.061696	0.021872	-0.044223
1	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	-0.044642	-0.051474	-0.026328	-0.008449
2	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	0.050680	0.044451	-0.005670	-0.045599
3	2547731	01-01-2023	MA	6.0	6.0	0	M	0	Error: Incorrect Reconstitution-	0	...	-0.044642	-0.011595	-0.036656	0.012191
4	2547732	01-01-2023	MA	38.0	38.0	0	F	0	Error: Patient Accidentally Stuck by Needle-	0	...	-0.044642	-0.036385	0.021872	0.003935
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3559	2552102	06-01-2023	0	0	0	0	U	0	HEART RATE WENT UP TO 186; GOT COVID; This spo...	0	...	NaN	NaN	NaN	NaN
3560	2552103	06-01-2023	0	65.0	0	0	F	0	Cryptococcal meningitis; embolic showers/throm...	0	...	NaN	NaN	NaN	NaN
3561	2552103	06-01-2023	0	65.0	0	0	F	0	Cryptococcal meningitis; embolic showers/throm...	0	...	NaN	NaN	NaN	NaN
3562	2552103	06-01-2023	0	65.0	0	0	F	0	Cryptococcal meningitis; embolic showers/throm...	0	...	NaN	NaN	NaN	NaN
3563	2552104	06-01-2023	0	0	0	0	M	0	covid 19; This spontaneous case was reported b...	0	...	NaN	NaN	NaN	NaN

3564 rows × 64 columns

```
In [ ]:
```

5.J. 'Merging' of 2 different data tables with a single matching column like SQL joins:

RawData =pd.read_csv(r"Path for Data Source File\Name of the data Source File.csv",parse_dates=True)

ForecastData =pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

Table1=pd.DataFrame(RawData)

Table2=pd.DataFrame(ForecastData)

pd.merge(RawData,ForecastData,on='SEX')

```
ForecastData
```

Out[64]:

	EmployeeID	Department	SEX	2021Salary	2022Salary
0	1	Finance	F	40000	100000
1	2	R&D	M	55000	80000
2	3	Sales	F	50000	50000
3	4	HR	M	42000	80000
4	5	General Maintainace	M	35000	40000
5	6	IT	M	80000	40000
6	7	Other	F	100000	35000

In [68]:
```
Table1=pd.DataFrame(RawData)
Table2=pd.DataFrame(ForecastData)
pd.merge(RawData,ForecastData,on='SEX')
```

Out[68]:

	VAERS_ID	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	VAX_LOT	VAX_DOSE_SERIES	VAX_RO
0	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	1808982	UNK	
1	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	1808982	UNK	
2	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	1808982	UNK	
3	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	1808982	UNK	
4	2547730	01-01-2023	DE	53.0	0	0	F	0	The adverse event is that the patient went int...	0	...	1808982	UNK	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
10262	2552073	05-01-2023	VA	14.0	14.0	0	M	0	7/19/2022 - first occurance of rapid heart rat...	0	...	EW0181	2	
10263	2552104	06-01-2023	0	0	0	0	M	0	covid 19; This spontaneous case was reported b...	0	...	ASKU	2	
10264	2552104	06-01-2023	0	0	0	0	M	0	covid 19; This spontaneous case was reported b...	0	...	ASKU	2	
10265	2552104	06-01-2023	0	0	0	0	M	0	covid 19; This spontaneous case was reported b...	0	...	ASKU	2	
10266	2552104	06-01-2023	0	0	0	0	M	0	covid 19; This spontaneous case was reported b...	0	...	ASKU	2	

10267 rows × 57 columns

```
ForecastData
```

Out[64]:

	EmployeeID	Department	SEX	2021Salary	2022Salary
0	1	Finance	F	40000	100000
1	2	R&D	M	55000	80000
2	3	Sales	F	50000	50000
3	4	HR	M	42000	80000
4	5	General Maintainace	M	35000	40000
5	6	IT	M	80000	40000
6	7	Other	F	100000	35000

In [68]:
```
Table1=pd.DataFrame(RawData)
Table2=pd.DataFrame(ForecastData)
pd.merge(RawData,ForecastData,on='SEX')
```

Out[68]:

TEXT	DIED	...	VAX_LOT	VAX_DOSE_SERIES	VAX_ROUTE	VAX_SITE	VAX_NAME	ADVERSE EVENT	EmployeeID	Department	2021Salary	2022Salary
lverse at the it int...	0	...	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	Blood pressure orthostatic abnormalCOVID-19Com...	1	Finance	40000	100000
lverse at the it int...	0	...	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	Blood pressure orthostatic abnormalCOVID-19Com...	3	Sales	50000	50000
lverse at the it int...	0	...	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	Blood pressure orthostatic abnormalCOVID-19Com...	7	Other	100000	35000
lverse at the it int...	0	...	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	Head injuryHeadacheLaboratory testMagnetic res...	1	Finance	40000	100000
lverse at the it int...	0	...	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	Head injuryHeadacheLaboratory testMagnetic res...	3	Sales	50000	50000
...	...	...	...	...	...	...	...	...	...	...	...	...
- first nce of t rat...	0	...	EW0181	2	SYR	LA	COVID19 (COVID19 (PFIZER-BIONTECH))	Supraventricular tachycardia	6	IT	80000	40000
l; This case ad b...	0	...	ASKU	2	OT	0	COVID19 (COVID19 (MODERNA))	COVID-19	2	R&D	55000	80000
l; This case ad b...	0	...	ASKU	2	OT	0	COVID19 (COVID19 (MODERNA))	COVID-19	4	HR	42000	80000
l; This case ad b...	0	...	ASKU	2	OT	0	COVID19 (COVID19 (MODERNA))	COVID-19	5	General Maintainace	35000	40000
l; This case ad b...	0	...	ASKU	2	OT	0	COVID19 (COVID19 (MODERNA))	COVID-19	6	IT	80000	40000

6. Data Visualization with Python

6.A. Histogram/Bar Chart:

Vaccinesdata =pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

x_axis = Vaccinesdata.Number_of_Adverse_Events

y_axis = Vaccinesdata.AGE_YRS

plt.figure(figsize=(5,5))

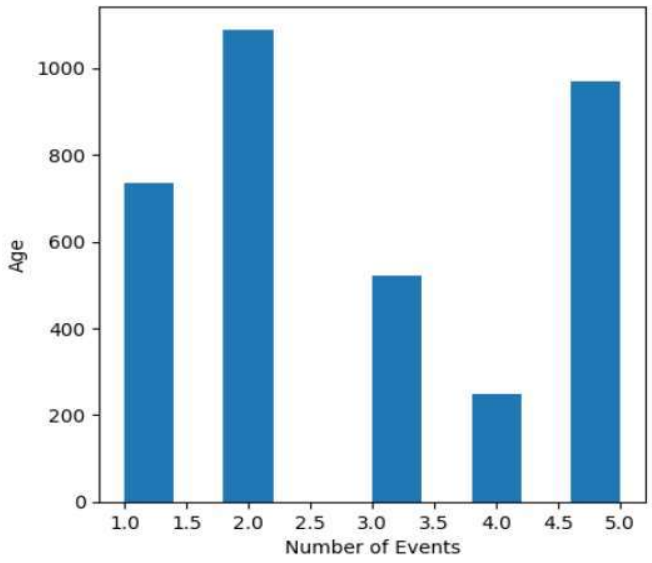
plt.hist(x_axis)

plt.xlabel('Number of Events')

plt.ylabel('Age')

plt.show()

```
In [122]: x_axis = Vaccinesdata.Number_of_Adverse_Events

In [102]: y_axis = Vaccinesdata.AGE_YRS

In [123]: plt.figure(figsize=(5,5))
          plt.hist(x_axis)
          plt.xlabel('Number of Events')
          plt.ylabel('Age')
          plt.show()
```



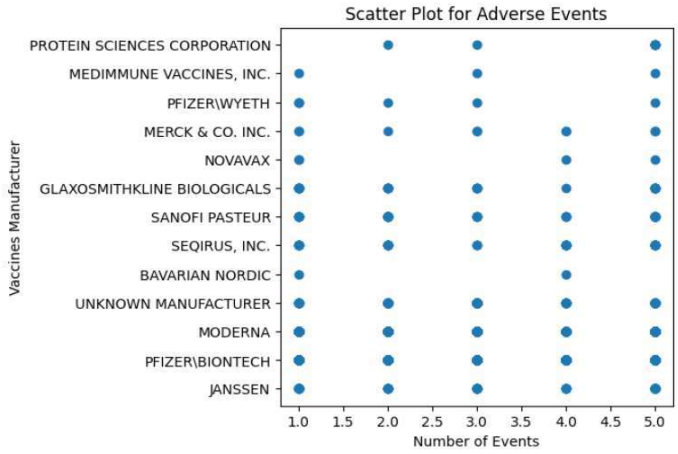
6.B. Scatter Plot:

Vaccinesdata =pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

i. Events based on manufacturer:

```
plt.figure(figsize=(5,5))

plt.title('Scatter Plot for Adverse Events')

plt.scatter(x=Vaccinesdata.Number_of_Adverse_Events,y=Vaccinesdata.VAX_MANU)

plt.xlabel('Number of Events')

plt.ylabel('Vaccines Manufacturer')

plt.show()
```

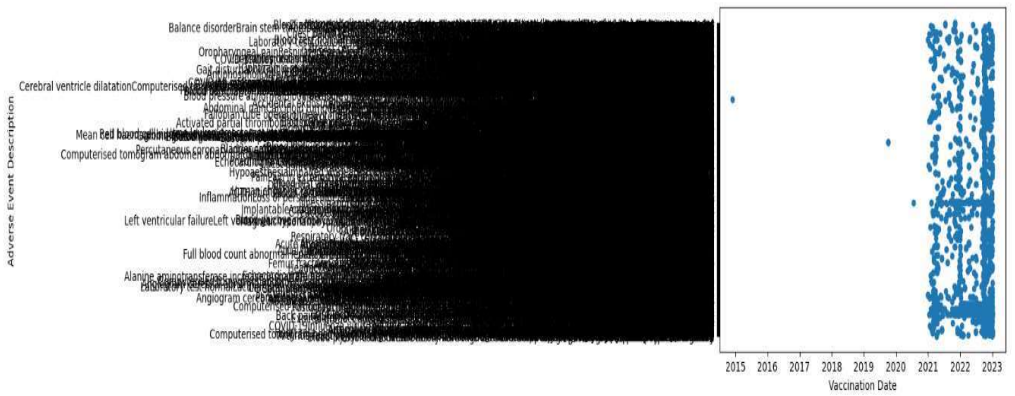
```
In [128]: plt.figure(figsize=(5,5))
plt.title('Scatter Plot for Adverse Events')
plt.scatter(x=Vaccinesdata.Number_of_Adverse_Events,y=Vaccinesdata.VAX_MANU)
plt.xlabel('Number of Events')
plt.ylabel('Vaccines Manufacturer')
plt.show()
```



ii. Events based on vaccines administered date scatter plot:

Vaccinesdata.plot(x='VAX_DATE',y='ADVERSE_EVENT',kind='scatter')
plt.xlabel('Vaccination Date')
plt.ylabel('Adverse Event Description')

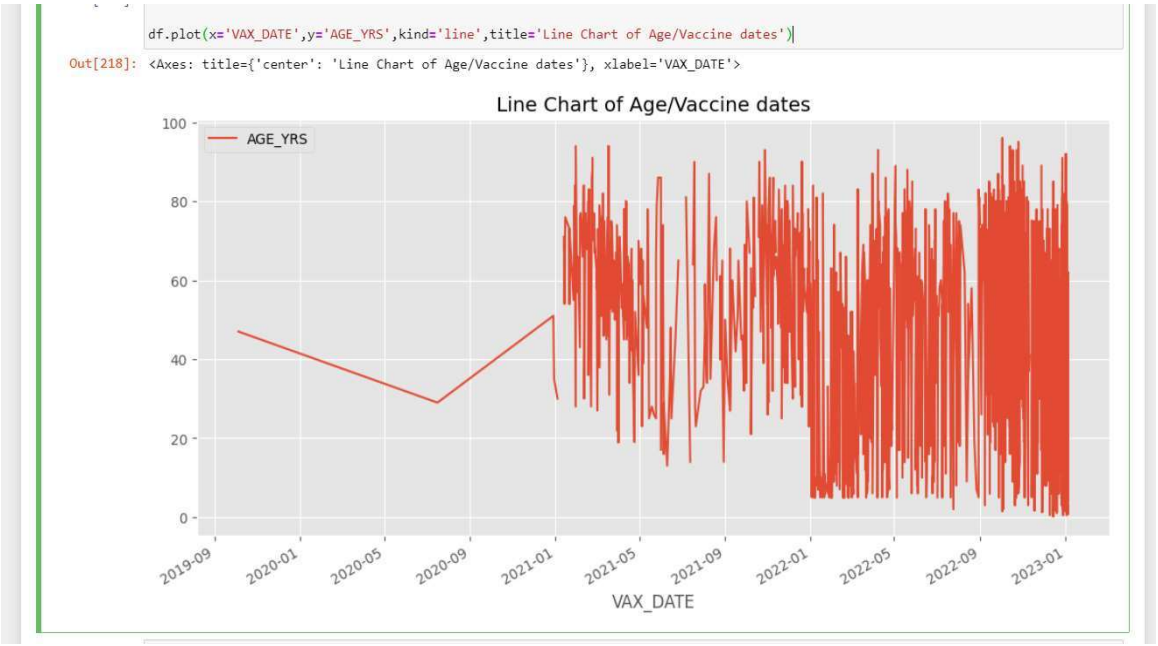
```
51]: Vaccinesdata.plot(x='VAX_DATE',y='ADVERSE_EVENT',kind='scatter')
plt.xlabel('Vaccination Date')
plt.ylabel('Adverse Event Description')

51]: Text(0, 0.5, 'Adverse Event Description')
```



6.C. Line Chart:

df =pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

df.plot(x='VAX_DATE',y='AGE_YRS',kind='line',title='Line Chart of Age/Vaccine dates')



6.D. Pie Chart:

df =pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

plt.figure(figsize=(5,5))

SEX = 'M','F','U'

percentile = [60,30,10]

explode=(0.05,0,0)

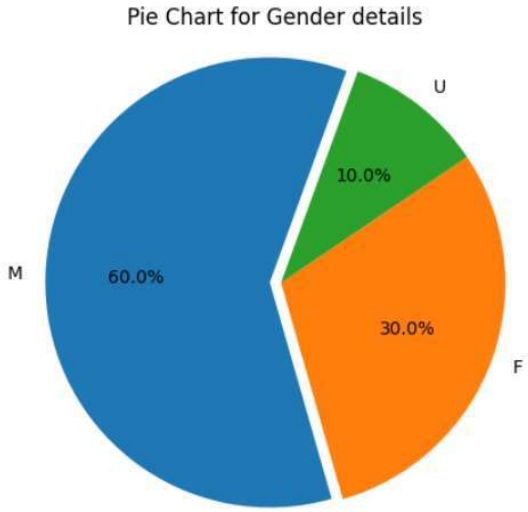
plt.pie(percentile,labels=SEX,explode=explode,autopct='%1.1f%%',startangle=70)

plt.axis('equal')

plt.title('Pie Chart for Gender details')

plt.show()

```
In [144]: plt.figure(figsize=(5,5))
          SEX = 'M','F','U'
          percentile = [60,30,10]
          explode=(0.05,0,0)
          plt.pie(percentile,labels=SEX,explode=explode,autopct='%1.1f%%',startangle=70)
          plt.axis('equal')
          plt.title('Pie Chart for Gender details')
          plt.show()
```

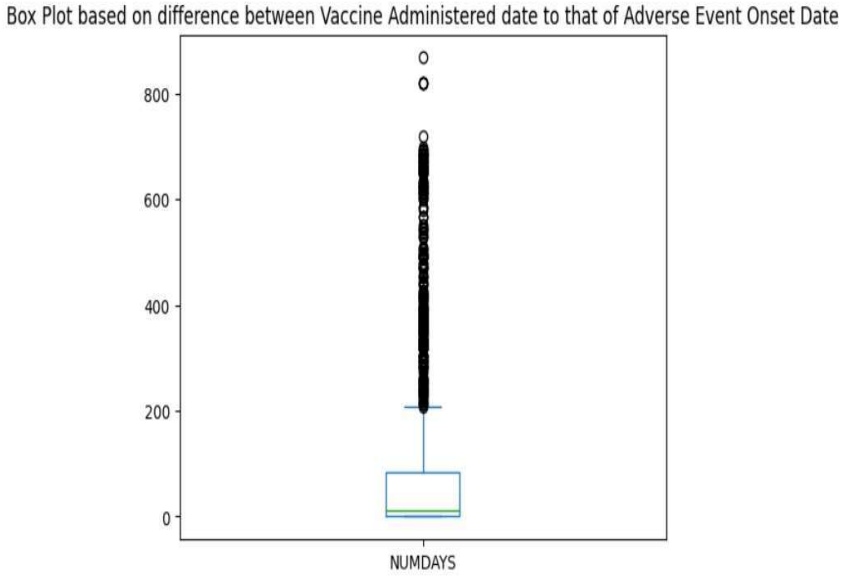


6.E. Box Plot (with distinct outliers):

Vaccinesdata =pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

Vaccinesdata['NUMDAYS'].plot(kind='box',title='Box Plot based on difference between Vaccine Administered date to that of Adverse Event Onset Date')

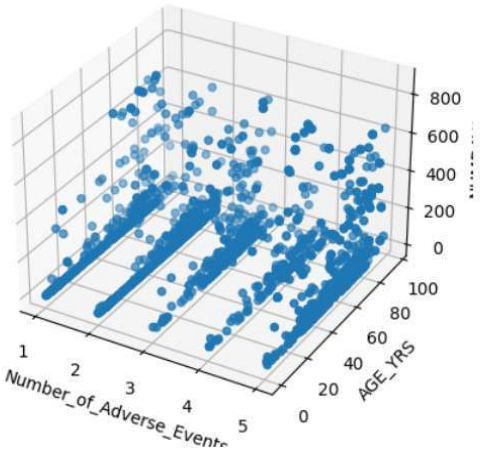
```
In [154]: Vaccinesdata['NUMDAYS'].plot(kind='box',title='Box Plot based on difference between Vaccine Administered date to that of Adverse
Out[154]: <Axes: title={'center': 'Box Plot based on difference between Vaccine Administered date to that of Adverse Event Onset Date'}>
```



6.F. 3D Visualization using 3axis:

```
Vaccinesdata =pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)


ax=plt.axes(projection='3d')

ax.scatter(Vaccinesdata.Number_of_Adverse_Events,Vaccinesdata.AGE_YRS,Vaccinesdata.NUMDAYS)

ax.set_xlabel('Number_of_Adverse_Events')

ax.set_ylabel('AGE_YRS')

ax.set_zlabel('NUMDAYS')

plt.show()
```

```
In [159]: #3D visualization
ax=plt.axes(projection='3d')
ax.scatter(Vaccinesdata.Number_of_Adverse_Events,Vaccinesdata.AGE_YRS,Vaccinesdata.NUMDAYS)
ax.set_xlabel('Number_of_Adverse_Events')
ax.set_ylabel('AGE_YRS')
ax.set_zlabel('NUMDAYS')
plt.show()
```




```
In [ ]:
```

6.G. Heatmap:

Heatmapdataset1=pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

Heatmapdataset1=(Trialdata.HOSPDAYS,Trialdata.Number_of_Adverse_Events)

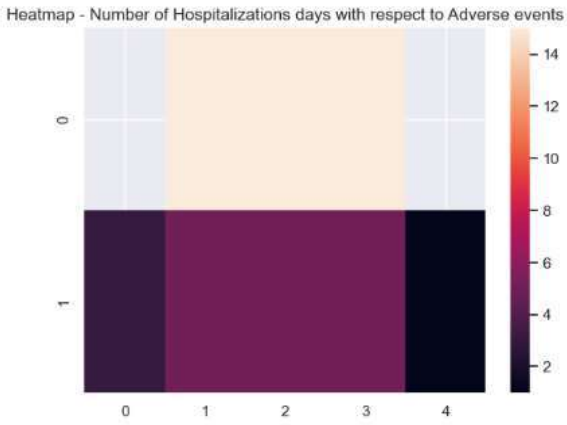
Heatmapdataset1

ax = sns.heatmap(Heatmapdataset1)

plt.title("Heatmap - Number of Hospitalizations days with respect to Adverse events")

```
In [229]: Heatmapdataset1=(Trialdata.HOSPDAYS,Trialdata.Number_of_Adverse_Events)
          Heatmapdataset1

Out[229]: (3559     NaN
           3560    15.0
           3561    15.0
           3562    15.0
           3563     NaN
           Name: HOSPDAYS, dtype: float64,
           3559    3
           3560    5
           3561    5
           3562    5
           3563    1
           Name: Number_of_Adverse_Events, dtype: int64)
```

```
In [235]: ax = sns.heatmap(Heatmapdataset1)
          plt.title("Heatmap - Number of Hospitalizations days with respect to Adverse events")

Out[235]: Text(0.5, 1.0, 'Heatmap - Number of Hospitalizations days with respect to Adverse events')
```



7. Write a Demo Script

7.A. Demo script which replaces existing values in 'Number of Events' column with defined parameters:

Heatmapdataset=pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

```
def Number_of_Adverse_Events(num):
    if num==5:
        return 'Highly Severe'
    if num==4:
        return 'Moderately Severe'
    if num==3:
        return 'Mild in Severity'
    if num==2:
        return 'Severe with impact'
    if num==1:
```

return 'Severe with no impact'

HeatmapDataset['Number_of_Adverse_Events'] =
HeatmapDataset['Number_of_Adverse_Events'].apply(Number_of_Adverse_Events)

HeatmapDataset

```
In [257]: def Number_of_Adverse_Events(num):
              if num==5:
                  return 'Highly Severe'
              if num==4:
                  return 'Moderately Severe'
              if num==3:
                  return 'Mild in Severity'
              if num==2:
                  return 'Severe with impact'
              if num==1:
                  return 'Severe with no impact'

          HeatmapDataset['Number_of_Adverse_Events'] = HeatmapDataset['Number_of_Adverse_Events'].apply(Number_of_Adverse_Events)
```

```
In [259]: HeatmapDataset

Out[259]:
```

VAX_TYPE	VAX_MANU	VAX_LOT	VAX_DOSE_SERIES	VAX_ROUTE	VAX_SITE	VAX_NAME	ADVERSE_EVENT	Number_of_Adverse_Events
COVID19	JANSSEN	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	Blood pressure orthostatic abnormalCOVID-19Com...	Highly Severe
COVID19	JANSSEN	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	Head injuryHeadacheLaboratory testMagnetic res...	Highly Severe
COVID19	JANSSEN	1808982	UNK	SYR	AR	COVID19 (COVID19 (JANSSEN))	SARS-CoV-2 test positiveUnresponsive to stimul...	Mild in Severity
COVID19	PFIZER\BIONTECH	GK1657	3	IM	LA	COVID19 (COVID19 (PFIZER-BIONTECH))	Product preparation issue	Severe with no impact
COVID19-2	MODERNA	141H22A	4	IM	LA	COVID19 (COVID19 (MODERNA BIVALENT))	Injury associated with device	Severe with no impact
...	...	...	...	...	...	...	...	...
COVID19	JANSSEN	NaN	1	NaN	NaN	COVID19 (COVID19 (JANSSEN))	COVID-19Heart rateHeart rate increased	Mild in Severity
COVID19	MODERNA	NaN	2	OT	NaN	COVID19 (COVID19 (MODERNA))	AtelectasisCOVID-19Chest X-rayComa scaleComput...	Highly Severe
COVID19	MODERNA	NaN	2	OT	NaN	COVID19 (COVID19 (MODERNA))	Diastolic dysfunctionEchocardiogramEjection fr...	Highly Severe
COVID19	MODERNA	NaN	2	OT	NaN	COVID19 (COVID19 (MODERNA))	MicroembolismPolymerase chain reactionSARS-CoV...	Highly Severe
COVID19	MODERNA	ASKU	2	OT	NaN	COVID19 (COVID19 (MODERNA))	COVID-19	Severe with no impact

7.B. Create pair plot visualization for above script:

Heatmapdataset=pd.read_excel(r"Path for Data Source File\Name of the data Source File.xlsx",parse_dates=True)

sns.pairplot(HeatmapDataset[['Number_of_Adverse_Events','HOSPDAYS','AGE_YRS']],hue='Number_of_Adverse_Events',size=4)

In [260]:
```
sns.pairplot(HeatmapDataset[['Number_of_Adverse_Events','HOSPDAYS','AGE_YRS']],hue='Number_of_Adverse_Events',size=4)
```

```
C:\Users\Divya\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:2095: UserWarning: The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
C:\Users\Divya\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[260]: <seaborn.axisgrid.PairGrid at 0x2425a2ec0d0>