Handaling Different types of Datasets.

In [24]:

```
my_dict={'Name':["a","b","c","d","e","f","g"],
         'age':[20,27,40,34,56,78,89],
         'designation':["VP","CEO","CER","DER","ERT","DER","ERT"]}

import pandas as pd
import numpy  as np
import seaborn as sns
df=pd.DataFrame(my_dict)
df
```

Out[24]:

	Name	age	designation
0	a	20	VP
1	b	27	CEO
2	c	40	CER
3	d	34	DER
4	e	56	ERT
5	f	78	DER
6	g	89	ERT

In [6]:

```
df.to_csv('Csv example')
df
```

Out[6]:

	Name	age	designation
0	a	20	VP
1	b	27	CEO
2	c	40	CER
3	d	34	DER
4	e	56	ERT
5	f	78	DER
6	g	89	ERT

In [7]:

```
df.head()
```

Out[7]:

	Name	age	designation
0	a	20	VP
1	b	27	CEO
2	c	40	CER
3	d	34	DER
4	e	56	ERT

In [8]:

```
df.tail()
```

Out[8]:

	Name	age	designation
2	c	40	CER
3	d	34	DER
4	e	56	ERT
5	f	78	DER
6	g	89	ERT

In [9]:

```
print(df)
```

```
  Name  age designation
0    a   20          VP
1    b   27         CEO
2    c   40         CER
3    d   34         DER
4    e   56         ERT
5    f   78         DER
6    g   89         ERT
```

In [10]:

```
df.shape
```

Out[10]:

```
(7, 3)
```

In [11]:

```
df.size
```

Out[11]:

```
21
```

In [14]:

```
df.describe()
```

Out[14]:

	age
count	7.000000
mean	49.142857
std	26.207051
min	20.000000
25%	30.500000
50%	40.000000
75%	67.000000
max	89.000000

In [16]:

```
df.columns
```

Out[16]:

```
Index(['Name', 'age', 'designation'], dtype='object')
```

In [17]:

```
df.nunique()
```

Out[17]:

```
Name           7
age            7
designation    5
dtype: int64
```

In [18]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Name         7 non-null      object
 1   age          7 non-null      int64 
 2   designation  7 non-null      object
dtypes: int64(1), object(2)
memory usage: 296.0+ bytes
```

In [19]:

```
df.min()
```

Out[19]:

```
Name            a
age            20
designation    CEO
dtype: object
```

In [20]:

```
df.max()
```

Out[20]:

```
Name            g
age            89
designation    VP
dtype: object
```

In [21]:

```
df.mode()
```

Out[21]:

	Name	age	designation
0	a	20	DER
1	b	27	ERT
2	c	34	NaN
3	d	40	NaN
4	e	56	NaN
5	f	78	NaN
6	g	89	NaN

In [22]:

```
df.mean()
```

```
C:\Users\Admin\AppData\Local\Temp/ipykernel_8248/3698961737.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_onl
y=None') is deprecated; in a future version this will raise TypeError.  Sele
ct only valid columns before calling the reduction.
  df.mean()
```

Out[22]:

```
age    49.142857
dtype: float64
```

In [23]:

```
df.median()
```

```
C:\Users\Admin\AppData\Local\Temp/ipykernel_8248/530051474.py:1: FutureWarni
ng: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only
=None') is deprecated; in a future version this will raise TypeError.  Selec
t only valid columns before calling the reduction.
  df.median()
```

Out[23]:

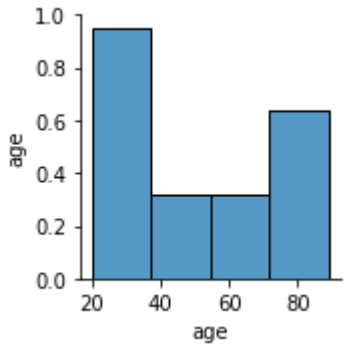
```
age    40.0
dtype: float64
```

In [25]:

```
sns.pairplot(df)
```

Out[25]:

```
<seaborn.axisgrid.PairGrid at 0x19d3c31cbe0>
```



In [41]:

```
import os
import pandas as pd
import numpy  as  np
import matplotlib .pyplot as plt
%matplotlib inline
import seaborn as sns
Location = "C:/Users/Admin/Downloads.csv"
```

In [40]:

```
import os
import pandas as pd
Location = "student-mat.csv"
```

In [35]:

```
df.head()
```

Out[35]:

	Name	age	designation
0	a	20	VP
1	b	27	CEO
2	c	40	CER
3	d	34	DER
4	e	56	ERT

In [42]:

```
df.tail()
```

Out[42]:

	Name	age	designation
2	c	40	CER
3	d	34	DER
4	e	56	ERT
5	f	78	DER
6	g	89	ERT

In [45]:

```
import os
import pandas as pd
import numpy  as  np
import matplotlib .pyplot as plt
%matplotlib inline
import seaborn as sns
Location = "student-mat - student-mat.csv"
```

In [46]:

```
df.head()
```

Out[46]:

	Name	age	designation
0	a	20	VP
1	b	27	CEO
2	c	40	CER
3	d	34	DER
4	e	56	ERT

In [72]:

```
import pandas as pd
Location = "student-mat - student-mat.csv"

#Loading the data and adding the header

df=pd.read_csv(Location, names=['Roll no','Names','Grades'])

#to add headers to a dataframe

df.columns=['Roll no','Names','Grades']
df.head()
```

Out[72]:

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardia
GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mothe
		17	U	GT3	T	1	1	at_home	other	course	fathe
		15	U	LE3	T	1	1	at_home	other	other	mothe
				GT3	T	4	2	health	services	home	mothe

In [74]:

```
df.tail()
```

Out[74]:

MS	M	20	U	LE3	A	2	2	services	services	course	other	1	2	2	no	yes	yes	no
		17	U	LE3	T	3	1	services	services	course	mother	2	1	0	no	no	no	no
		21	R	GT3	T	1	1	other	other	course	other	1	1	3	no	no	no	no
		18	R	LE3	T	3	2	services	other	course	mother	3	1	0	no	no	no	no
		19	U	LE3	T	1	1	other	at_home	course	father	1	1	0	no	no	no	no

In [75]:

```
print(df)
```

```
Roll no  \
school sex age address famsize Pstatus Medu Fedu Mjob     Fjob     reason gu
ardian traveltime studytime failures schoolsup famsup paid activities nurser
y higher internet romantic famrel freetime goout Dalc Walc health absences
G1
GP     F   18  U       GT3     A       4    4    at_home  teacher  course mo
ther   2          2         0        yes       no     no   no         yes
yes    no       no       4      3        4     1    1    3      6
5
           17  U       GT3     T       1    1    at_home  other    course fa
ther   1          2         0        no        yes    no   no         no
yes    yes      no       5      3        3     1    1    3      4
5
           15  U       LE3     T       1    1    at_home  other    other  mo
ther   1          2         3        yes       no     yes  no         yes
yes    yes      no       4      3        2     2    3    3      10
7
                       GT3     T       4    2    health   services home   mo
ther   1          3         0        no        yes    yes  yes        yes
yes    yes      yes      3      2        2     1    1    5      2
15
...
...
MS     M   20  U       LE3     A       2    2    services services course ot
her    1          2         2        no        yes    yes  no         yes
yes    no       no       5      5        4     4    5    4      11
9
           17  U       LE3     T       3    1    services services course mo
ther   2          1         0        no        no     no   no         no
yes    yes      no       2      4        5     3    4    2      3
14
           21  R       GT3     T       1    1    other    other    course ot
her    1          1         3        no        no     no   no         no
yes    no       no       5      5        3     3    3    3      3
10
           18  R       LE3     T       3    2    services other    course mo
ther   3          1         0        no        no     no   no         no
yes    yes      no       4      4        1     3    4    5      0
11
           19  U       LE3     T       1    1    other    at_home  course fa
ther   1          1         0        no        no     no   no         yes
yes    yes      no       3      2        3     3    3    5      5
8


Names  \
school sex age address famsize Pstatus Medu Fedu Mjob     Fjob     reason gu
ardian traveltime studytime failures schoolsup famsup paid activities nurser
y higher internet romantic famrel freetime goout Dalc Walc health absences
G2
GP     F   18  U       GT3     A       4    4    at_home  teacher  course mo
ther   2          2         0        yes       no     no   no         yes
yes    no       no       4      3        4     1    1    3      6
6
           17  U       GT3     T       1    1    at_home  other    course fa
ther   1          2         0        no        yes    no   no         no
```

```
yes    yes      no       5      3        3     1    1    3      4
5
           15  U       LE3     T       1    1    at_home  other    other  mo
ther   1          2         3        yes       no     yes  no         yes
yes    yes      no       4      3        2     2    3    3      10
8
                       GT3     T       4    2    health   services home   mo
ther   1          3         0        no        yes    yes  yes        yes
yes    yes      yes      3      2        2     1    1    5      2
14
...
...
MS     M   20  U       LE3     A       2    2    services services course ot
her    1          2         2        no        yes    yes  no         yes
yes    no       no       5      5        4     4    5    4      11
9
           17  U       LE3     T       3    1    services services course mo
ther   2          1         0        no        no     no   no         no
yes    yes      no       2      4        5     3    4    2      3
16
           21  R       GT3     T       1    1    other    other    course ot
her    1          1         3        no        no     no   no         no
yes    no       no       5      5        3     3    3    3      3
8
           18  R       LE3     T       3    2    services other    course mo
ther   3          1         0        no        no     no   no         no
yes    yes      no       4      4        1     3    4    5      0
12
           19  U       LE3     T       1    1    other    at_home  course fa
ther   1          1         0        no        no     no   no         yes
yes    yes      no       3      2        3     3    3    5      5
9


Grades
school sex age address famsize Pstatus Medu Fedu Mjob     Fjob     reason gu
ardian traveltime studytime failures schoolsup famsup paid activities nurser
y higher internet romantic famrel freetime goout Dalc Walc health absences
G3
GP     F   18  U       GT3     A       4    4    at_home  teacher  course mo
ther   2          2         0        yes       no     no   no         yes
yes    no       no       4      3        4     1    1    3      6
6
           17  U       GT3     T       1    1    at_home  other    course fa
ther   1          2         0        no        yes    no   no         no
yes    yes      no       5      3        3     1    1    3      4
6
           15  U       LE3     T       1    1    at_home  other    other  mo
ther   1          2         3        yes       no     yes  no         yes
yes    yes      no       4      3        2     2    3    3      10
10
                       GT3     T       4    2    health   services home   mo
ther   1          3         0        no        yes    yes  yes        yes
yes    yes      yes      3      2        2     1    1    5      2
15
...
...
MS     M   20  U       LE3     A       2    2    services services course ot
her    1          2         2        no        yes    yes  no         yes
yes    no       no       5      5        4     4    5    4      11
9
```

```
           17  U       LE3     T       3    1    services services course mo
ther   2          1         0       no        no     no   no         no
yes    yes      no       2      4        5     3    4    2      3
16
           21  R       GT3     T       1    1    other    other    course ot
her    1          1         3       no        no     no   no         no
yes    no       no       5      5        3     3    3    3      3
7
           18  R       LE3     T       3    2    services other    course mo
ther   3          1         0       no        no     no   no         no
yes    yes      no       4      4        1     3    4    5      0
10
           19  U       LE3     T       1    1    other    at_home  course fa
ther   1          1         0       no        no     no   no         yes
yes    yes      no       3      2        3     3    3    5      5
9

[396 rows x 3 columns]
```

In [76]:

```
df.shape
```

Out[76]:

```
(396, 3)
```

In [77]:

```
df.describe()
```

Out[77]:

	Roll no	Names	Grades
count	396	396	396
unique	18	18	19
top	10	9	10
freq	51	50	56

In [78]:

```
df.columns
```

Out[78]:

```
Index(['Roll no', 'Names', 'Grades'], dtype='object')
```

In [79]:

```
df.nunique()
```

Out[79]:

```
Roll no    18
Names      18
Grades     19
dtype: int64
```

In [85]:

```
import pandas as pd
names=['sushant','raj','ram','arvind','akash']
grades=   [98,67,78,35,89]
bsdegrees=[1,2,3,4,5]
msdegrees=[0,1,2,3,4]
phddegrees=[2,3,5,6,0]

Degrees = zip(names,grades,bsdegrees,msdegrees,phddegrees)
columns = ['Name','Grades','Bs','Ms','Phd']

df=pd.DataFrame(data=Degrees,columns=columns)
df
```

Out[85]:

	Name	Grades	Bs	Ms	Phd
0	sushant	98	1	0	2
1	raj	67	2	1	3
2	ram	78	3	2	5
3	arvind	35	4	3	6
4	akash	89	5	4	0

In [86]:

```
# loading data from excel file and changing columns name

import pandas as pd
Location="gradedata.xlsx"
df=pd.read_excel(Location)

#changing columns names
df.columns =['first','last','sex','age','exer','hrs','grd','addr']
df.head()
```

Out[86]:

	first	last	sex	age	exer	hrs	grd	addr
0	Marcia	Pugh	female	17.0	3.0	10.0	82.4	7379 Highland Rd. , Dublin, GA 31021
1	Kadeem	Morrison	male	18.0	4.0	4.0	78.2	8 Bayport St. , Honolulu, HI 96815
2	Nash	Powell	male	18.0	5.0	9.0	79.3	Encino, CA 91316, 3 Lilac Street
3	Noelani	Wagner	female	14.0	2.0	7.0	83.2	Riverview, FL 33569, 9998 North Smith Dr.
4	Noelani	Cherry	female	18.0	4.0	15.0	87.4	97 SE. Ocean Street , Bethlehem, PA 18015

In [87]:

```
import pandas as pd
names=['sushant','raj','ram','arvind','akash']
grades=   [98,67,78,35,89]
GradeList =zip(names,grades)
df=pd.DataFrame(data=GradeList,columns=['Name','Grades'])

writer=pd.ExcelWriter('dataframe.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name="Sheet1")
writer.save()
```

In [ ]:

```

```