```
#Pandas
import pandas as pd
```

```
#Series pd.series(column,index)
A = ['SAKULJI',"DILEEP","SHANKAR","SHUBHAS","SHIVA","PRABHAS"]
index = [1,2,3,4,5,6]
B = pd.Series(A,index)
print(B)
```

```
1    SAKULJI
2     DILEEP
3    SHANKAR
4    SHUBHAS
5      SHIVA
6    PRABHAS
dtype: object
```

```
#loading csv
dia = pd.read_csv("/content/diabetcsv.csv")
print(dia)
```

```
     preg  plas  pres  skin  insu  mass   pedi  age            class
0       6   148    72    35     0  33.6  0.627   50  tested_positive
1       1    85    66    29     0  26.6  0.351   31  tested_negative
2       8   183    64     0     0  23.3  0.672   32  tested_positive
3       1    89    66    23    94  28.1  0.167   21  tested_negative
4       0   137    40    35   168  43.1  2.288   33  tested_positive
..    ...   ...   ...   ...   ...   ...    ...  ...              ...
763    10   101    76    48   180  32.9  0.171   63  tested_negative
764     2   122    70    27     0  36.8  0.340   27  tested_negative
765     5   121    72    23   112  26.2  0.245   30  tested_negative
766     1   126    60     0     0  30.1  0.349   47  tested_positive
767     1    93    70    31     0  30.4  0.315   23  tested_negative

[768 rows x 9 columns]
```

```
grad=pd.read_csv("/content/diabetcsv.csv")
grad
```

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	tested_negative
764	2	122	70	27	0	36.8	0.340	27	tested_negative
765	5	121	72	23	112	26.2	0.245	30	tested_negative
766	1	126	60	0	0	30.1	0.349	47	tested_positive
767	1	93	70	31	0	30.4	0.315	23	tested_negative

768 rows × 9 columns

```
dia = pd.read_csv("/content/demodt.txt")
print(dia)
```

```
  State  Literacy  Cleanliness  Crime_Rate  Good
0     A        92           90          54     0
1     B        56           67          50     1
2     C        78           85          62     0
3     D        63           72          48     1
4     E        85           79          55     0
5     F        71           68          58     0
6     G        80           83          51     0
7     H        67           74          47     1
8     I        89           88          53     0
```

```
9      J       58          65          49    1
10     K       82          81          60    0
11     L       75          78          57    0
12     M       69          70          46    1
13     N       87          86          52    0
14     O       61          63          45    1
15     P       93          91          56    0
16     Q       55          66          61    0
17     R       76          77          59    0
18     S       84          82          44    1
19     T       70          69          50    1
20     U       94          92          57    0
21     V       59          64          52    0
22     W       83          80          43    1
23     X       74          76          63    0
24     Y       68          73          41    1
25     Z       88          84          47    1
```

```
grad=pd.read_csv("/content/demodt.txt")
grad
```

	State	Literacy	Cleanliness	Crime_Rate	Good
0	A	92	90	54	0
1	B	56	67	50	1
2	C	78	85	62	0
3	D	63	72	48	1
4	E	85	79	55	0
5	F	71	68	58	0
6	G	80	83	51	0
7	H	67	74	47	1
8	I	89	88	53	0
9	J	58	65	49	1
10	K	82	81	60	0
11	L	75	78	57	0
12	M	69	70	46	1
13	N	87	86	52	0
14	O	61	63	45	1
15	P	93	91	56	0
16	Q	55	66	61	0
17	R	76	77	59	0
18	S	84	82	44	1
19	T	70	69	50	1
20	U	94	92	57	0
21	V	59	64	52	0
22	W	83	80	43	1
23	X	74	76	63	0
24	Y	68	73	41	1
25	Z	88	84	47	1

```
dia = pd.read_excel("/content/diabetes.xlsx")
print(dia)
```

```
     preg  plas  pres  skin  insu  mass   pedi  age            class
0       6   148    72    35     0  33.6  0.627   50  tested_positive
1       1    85    66    29     0  26.6  0.351   31  tested_negative
2       8   183    64     0     0  23.3  0.672   32  tested_positive
3       1    89    66    23    94  28.1  0.167   21  tested_negative
4       0   137    40    35   168  43.1  2.288   33  tested_positive
..    ...   ...   ...   ...   ...   ...    ...  ...              ...
763    10   101    76    48   180  32.9  0.171   63  tested_negative
```

```
764     2   122    70    27     0  36.8  0.340   27  tested_negative
765     5   121    72    23   112  26.2  0.245   30  tested_negative
766     1   126    60     0     0  30.1  0.349   47  tested_positive
767     1    93    70    31     0  30.4  0.315   23  tested_negative

[768 rows x 9 columns]
```

```
grad=pd.read_excel("/content/diabetes.xlsx")
grad
```

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	tested_negative
764	2	122	70	27	0	36.8	0.340	27	tested_negative
765	5	121	72	23	112	26.2	0.245	30	tested_negative
766	1	126	60	0	0	30.1	0.349	47	tested_positive
767	1	93	70	31	0	30.4	0.315	23	tested_negative

768 rows × 9 columns

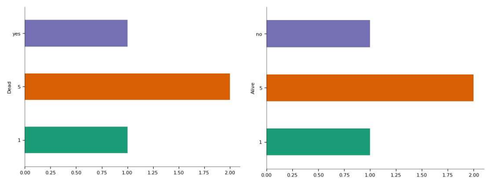
```
diaxl1=pd.read_excel("/content/diabetes.xlsx",sheet_name="dora")
diaxl1
```

	Dead	Alive
0	yes	no
1	yes	no
2	yes	no
3	yes	no
4	yes	no

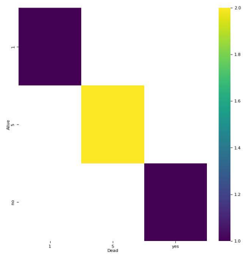
```
diaxl1.describe()
```

	Dead	Alive
count	5	5
unique	1	1
top	yes	no
freq	5	5

Categorical distributions



2-d categorical distributions



```
diaxl1.describe()
```

	Dead	Alive
count	5	5
unique	1	1
top	yes	no
freq	5	5

```
print(dia.shape)#Return rows and columns
print(dia.shape[0])#only rows
print(dia.shape[1])#only columns
```

```
(768, 9)
768
9
```

```
dia1=pd.read_csv("/content/grades_withnulls.csv")
dia1
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	NaN	9.1	9.9	A+	1
9	Rajini	M	NaN	9.1	9.3	A	0
10	Kiran	V	NaN	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
grad=pd.read_csv("/content/grades_withnulls.csv")
grad
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	NaN	9.1	9.9	A+	1
9	Rajini	M	NaN	9.1	9.3	A	0
10	Kiran	V	NaN	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
print(df2.drop_duplicates())#changes are not reflected
print(df2)
df2.drop_duplicates(inplace=True)#changes reflected
print(df2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-9cfc545544f7> in <cell line: 1>()
----> 1 print(df2.drop_duplicates())#changes are not reflected
      2 print(df2)
      3 df2.drop_duplicates(inplace=True)#changes reflected
      4 print(df2)

NameError: name 'df2' is not defined
```

```
df2.tail()#by default value is 5
df2.head()#by default value is 5
df2.head(2)
df2.tail(2)
```

```
#columns
col=list(df2.columns)
print(col)
```

```
dem=pd.read_excel("/content/diabetes.xlsx",sheet_name="dora")
gradescsv=pd.read_csv("/content/grades_withnulls.csv")
dem.describe()
print(dem.shape)#returns no of rows and columns
print(dem.shape[0])#returns no of rows if 0
print(dem.shape[1])#returns no of columns if 1
print(gradescsv.isnull())#returns where null values are present. NaN is nothing but not a number
#syntax is variable.isnull()
print(gradescsv.isnull().sum())#returns the total no of true(true in the sense null) values in every column
```

```
#data cleaning is nothing but removing null values
#it can be done in two ways
#save the returned data in another file or use inplace=True
print(gradescsv.dropna())#returns the file with deleted rows of NaN values but the new file is not replaced with the original one
print(gradescsv)
gc=gradescsv.dropna()
print(gc)
gradescsv.dropna(inplace=True)
print(gradescsv)
```

```
     Names Initials  SEM1  SEM2  SEM3 Grade  Placed
0      Joe        K   9.8  10.0   9.9    A+       1
1   Rajesh        M   8.9   9.1   9.3     A       1
2   Kissan        V   9.9   9.8  10.0     A       0
4     Jeen        K   9.8   9.1   9.9    A+       1
5      Raj        M   8.9   9.1   9.3     A       1
6   Hassan        V   9.9   9.0   9.2     A       1
7     Mari        N   7.7   8.0   7.1     B       1
11    Maya        N   7.7   8.0   7.1     B       0
12   Jolin        K   9.8   9.1   9.9    A+       1
13  Rajesh        M   8.9   9.1   9.3     A       1
14    Riya        M   9.3   9.9  10.0     A       1
15    Sana        V   9.9   9.3   9.2     A       0
16    Mark        N   7.7   8.0   7.0     B       0
     Names Initials  SEM1  SEM2  SEM3 Grade  Placed
0      Joe        K   9.8  10.0   9.9    A+       1
1   Rajesh        M   8.9   9.1   9.3     A       1
2   Kissan        V   9.9   9.8  10.0     A       0
3     Mary        N   7.7   8.0   NaN     B       0
4     Jeen        K   9.8   9.1   9.9    A+       1
5      Raj        M   8.9   9.1   9.3     A       1
6   Hassan        V   9.9   9.0   9.2     A       1
7     Mari        N   7.7   8.0   7.1     B       1
8     Jess        K   NaN   9.1   9.9    A+       1
9   Rajini        M   NaN   9.1   9.3     A       0
10   Kiran        V   NaN   9.3   9.2     A       0
11    Maya        N   7.7   8.0   7.1     B       0
12   Jolin        K   9.8   9.1   9.9    A+       1
13  Rajesh        M   8.9   9.1   9.3     A       1
14    Riya        M   9.3   9.9  10.0     A       1
15    Sana        V   9.9   9.3   9.2     A       0
16    Mark        N   7.7   8.0   7.0     B       0
     Names Initials  SEM1  SEM2  SEM3 Grade  Placed
0      Joe        K   9.8  10.0   9.9    A+       1
1   Rajesh        M   8.9   9.1   9.3     A       1
2   Kissan        V   9.9   9.8  10.0     A       0
4     Jeen        K   9.8   9.1   9.9    A+       1
5      Raj        M   8.9   9.1   9.3     A       1
```

```
6   Hassan        V   9.9   9.0   9.2     A       1
7     Mari        N   7.7   8.0   7.1     B       1
11    Maya        N   7.7   8.0   7.1     B       0
12   Jolin        K   9.8   9.1   9.9    A+       1
13  Rajesh        M   8.9   9.1   9.3     A       1
14    Riya        M   9.3   9.9  10.0     A       1
15    Sana        V   9.9   9.3   9.2     A       0
16    Mark        N   7.7   8.0   7.0     B       0
     Names Initials  SEM1  SEM2  SEM3 Grade  Placed
0      Joe        K   9.8  10.0   9.9    A+       1
1   Rajesh        M   8.9   9.1   9.3     A       1
2   Kissan        V   9.9   9.8  10.0     A       0
4     Jeen        K   9.8   9.1   9.9    A+       1
5      Raj        M   8.9   9.1   9.3     A       1
6   Hassan        V   9.9   9.0   9.2     A       1
7     Mari        N   7.7   8.0   7.1     B       1
11    Maya        N   7.7   8.0   7.1     B       0
12   Jolin        K   9.8   9.1   9.9    A+       1
13  Rajesh        M   8.9   9.1   9.3     A       1
14    Riya        M   9.3   9.9  10.0     A       1
```

```
df1=pd.read_csv("/content/grades_withnulls.csv")
df1.fillna(555,inplace=True)#Changes are saved
df1
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	555.0	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	555.0	9.1	9.9	A+	1
9	Rajini	M	555.0	9.1	9.3	A	0
10	Kiran	V	555.0	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
df1=pd.read_csv("/content/grades_withnulls.csv")
mv=df1['SEM1'].mean()
print(mv)
df1.fillna(mv,inplace=True)
df1
```

```
8.992857142857144
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.800000	10.0	9.900000	A+	1
1	Rajesh	M	8.900000	9.1	9.300000	A	1
2	Kissan	V	9.900000	9.8	10.000000	A	0
3	Mary	N	7.700000	8.0	8.992857	B	0
4	Jeen	K	9.800000	9.1	9.900000	A+	1
5	Raj	M	8.900000	9.1	9.300000	A	1
6	Hassan	V	9.900000	9.0	9.200000	A	1
7	Mari	N	7.700000	8.0	7.100000	B	1
8	Jess	K	8.992857	9.1	9.900000	A+	1
9	Rajini	M	8.992857	9.1	9.300000	A	0
10	Kiran	V	8.992857	9.3	9.200000	A	0
11	Maya	N	7.700000	8.0	7.100000	B	0
12	Jolin	K	9.800000	9.1	9.900000	A+	1
13	Rajesh	M	8.900000	9.1	9.300000	A	1
14	Riya	M	9.300000	9.9	10.000000	A	1
15	Sana	V	9.900000	9.3	9.200000	A	0
16	Mark	N	7.700000	8.0	7.000000	B	0

```
#Access The Data
#iloc-integer location,index
#loc-fields name,index
#dfname.loc[index]-->rows
#dfname.loc[st::stop]-->range of rows
#dfname.loc[row_index,col_index]-->rows and columns
mydf=pd.read_csv("/content/grades_withnulls.csv")
```

```
mydf.loc[5]
```

```
Names       Raj
Initials      M
SEM1        8.9
SEM2        9.1
SEM3        9.3
Grade         A
Placed        1
Name: 5, dtype: object
```

```
#range of records
#df.loc[i,j] range of records
mydf.loc[0:5]#first five record
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1

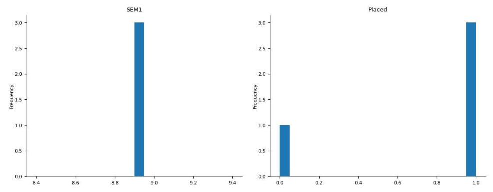
```
mydf.iloc[5:9,0:4]
```

	Names	Initials	SEM1	SEM2
5	Raj	M	8.9	9.1
6	Hassan	V	9.9	9.0
7	Mari	N	7.7	8.0
8	Jess	K	NaN	9.1

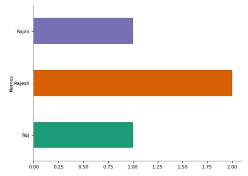
```
mydf[mydf.SEM3==9.3]
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
1	Rajesh	M	8.9	9.1	9.3	A	1
5	Raj	M	8.9	9.1	9.3	A	1
9	Rajini	M	NaN	9.1	9.3	A	0
13	Rajesh	M	8.9	9.1	9.3	A	1

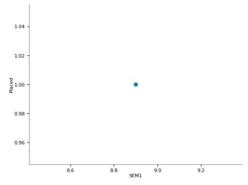
Distributions



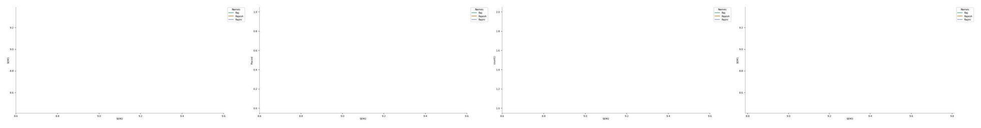
Categorical distributions



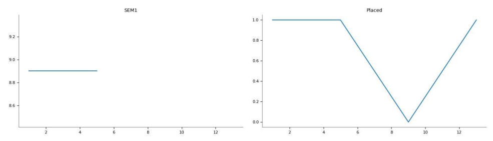
2-d distributions



Time series



Values



Faceted distributions

```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `leg

                              <string>:5: FutureWarning:
```

```
#DISPLAYING THE DATA BASED UPON THE CONDITION
mydf.loc[mydf.SEM1>9.5,'Names']
```

```
0        Joe
2     Kissan
4       Jeen
6     Hassan
12     Jolin
15      Sana
Name: Names, dtype: object
```

```
#print the grades of the students who scored more than 9 in SEM3
import pandas as pd
mydf=pd.read_csv("/content/grades_withnulls.csv")
mydf.loc[mydf.SEM3>9,"Grade"]
```

```
0     A+
1      A
2      A
4     A+
5      A
6      A
8     A+
9      A
10     A
12    A+
13     A
14     A
15     A
Name: Grade, dtype: object
```

```
mydf.drop_duplicates
mydf
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	NaN	9.1	9.9	A+	1
9	Rajini	M	NaN	9.1	9.3	A	0
10	Kiran	V	NaN	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
mydf.head()#top five
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1

```
mydf.tail()#last five
```

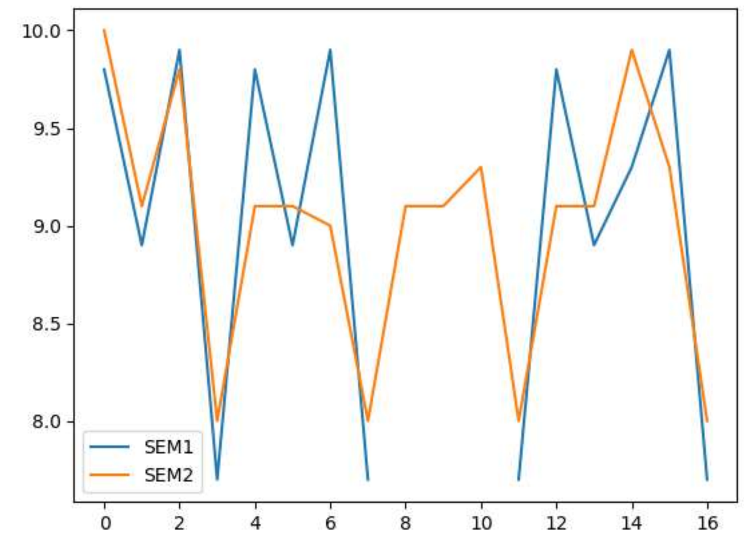
```
     Names  Initials  SEM1  SEM2  SEM3  Grade  Placed
12   Jolin         K   9.8   9.1   9.9     A+       1
13  Rajesh         M   8.9   9.1   9.3      A       1
14    Riya         M   9.3   9.9  10.0      A       1
15    Sana         V   9.9   9.3   9.2      A       0
16    Mark         N   7.7   8.0   7.0      B       0
```

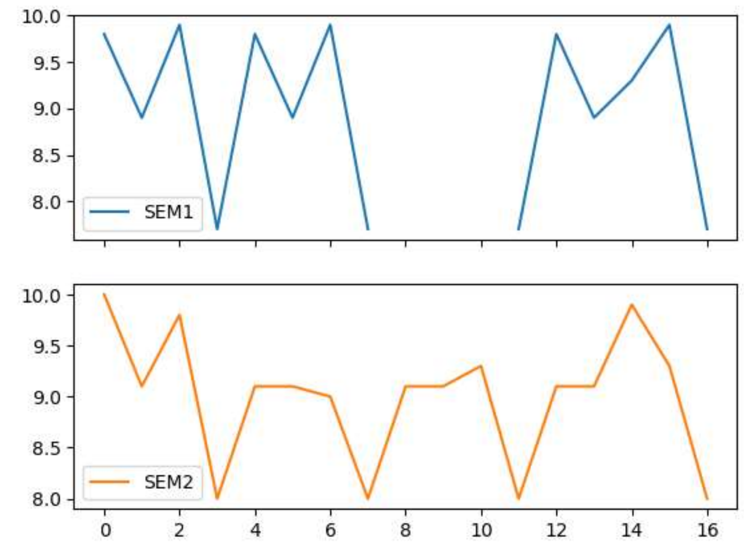
```
#Create new column with average values of three sum
mydf["Average"]=(mydf['SEM1']+mydf['SEM2']+mydf['SEM3'])/3
mydf
```

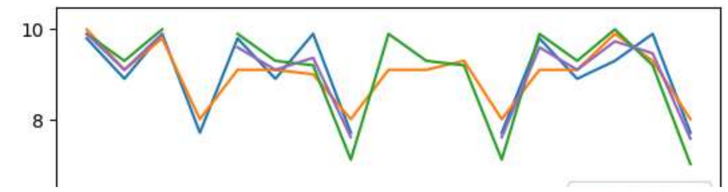
```
     Names  Initials      SEM1  SEM2       SEM3  Grade  Placed   Average
0      Joe         K  9.800000  10.0   9.900000     A+       1  9.900000
1   Rajesh         M  8.900000   9.1   9.300000      A       1  9.100000
2   Kissan         V  9.900000   9.8  10.000000      A       0  9.900000
3     Mary         N  7.700000   8.0   8.992857      B       0  8.230952
4     Jeen         K  9.800000   9.1   9.900000     A+       1  9.600000
5      Raj         M  8.900000   9.1   9.300000      A       1  9.100000
6   Hassan         V  9.900000   9.0   9.200000      A       1  9.366667
7     Mari         N  7.700000   8.0   7.100000      B       1  7.600000
8     Jess         K  8.992857   9.1   9.900000     A+       1  9.330952
9   Rajini         M  8.992857   9.1   9.300000      A       0  9.130952
10   Kiran         V  8.992857   9.3   9.200000      A       0  9.164286
11    Maya         N  7.700000   8.0   7.100000      B       0  7.600000
12   Jolin         K  9.800000   9.1   9.900000     A+       1  9.600000
13  Rajesh         M  8.900000   9.1   9.300000      A       1  9.100000
14    Riya         M  9.300000   9.9  10.000000      A       1  9.733333
15    Sana         V  9.900000   9.3   9.200000      A       0  9.466667
16    Mark         N  7.700000   8.0   7.000000      B       0  7.566667
```

```
mydf["Conduct"]="Good"
mydf["Average"]=(mydf["SEM1"]+mydf["SEM2"]+mydf["SEM3"])/3
mydf[['SEM1','SEM2']].plot.line()
mydf[['SEM1','SEM2']].plot.line(subplots=True)
mydf.plot.line()
mydf.plot.line(subplots=True)
```

```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >], dtype=object)
```







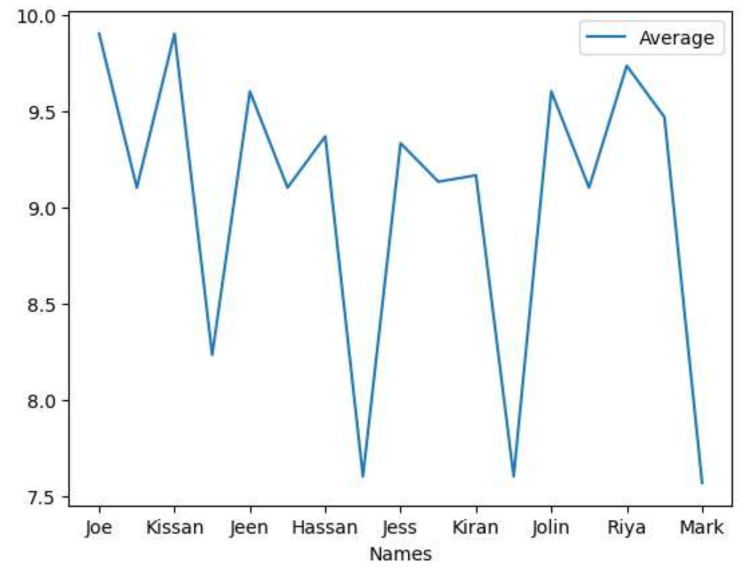
```
mydf=pd.read_csv("/content/grades_withnulls.csv")
mv=mydf['SEM1'].mean()
print(mv)
mydf.fillna(mv,inplace=True)
mydf
```

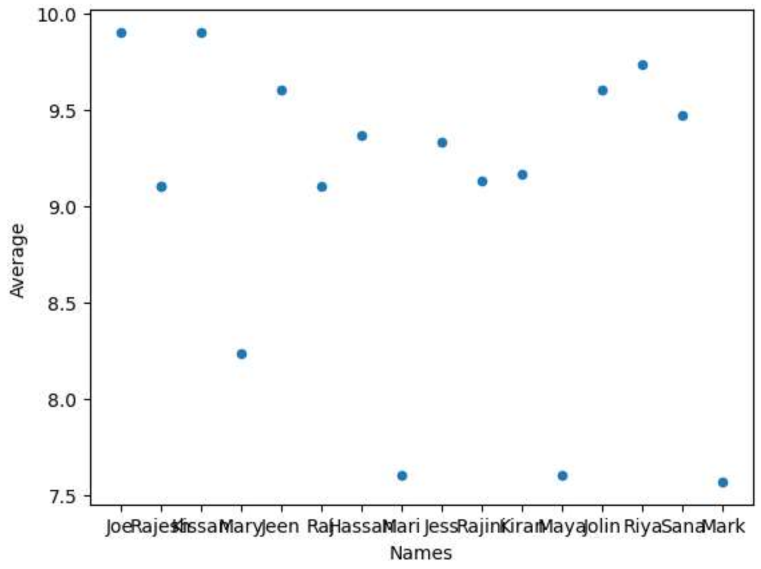
```
8.992857142857144
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.800000	10.0	9.900000	A+	1
1	Rajesh	M	8.900000	9.1	9.300000	A	1
2	Kissan	V	9.900000	9.8	10.000000	A	0
3	Mary	N	7.700000	8.0	8.992857	B	0
4	Jeen	K	9.800000	9.1	9.900000	A+	1
5	Raj	M	8.900000	9.1	9.300000	A	1
6	Hassan	V	9.900000	9.0	9.200000	A	1
7	Mari	N	7.700000	8.0	7.100000	B	1
8	Jess	K	8.992857	9.1	9.900000	A+	1
9	Rajini	M	8.992857	9.1	9.300000	A	0
10	Kiran	V	8.992857	9.3	9.200000	A	0
11	Maya	N	7.700000	8.0	7.100000	B	0
12	Jolin	K	9.800000	9.1	9.900000	A+	1
13	Rajesh	M	8.900000	9.1	9.300000	A	1
14	Riya	M	9.300000	9.9	10.000000	A	1
15	Sana	V	9.900000	9.3	9.200000	A	0
16	Mark	N	7.700000	8.0	7.000000	B	0

```
mydf.plot(kind="line",x="Names",y="Average")
mydf.plot(kind="scatter",x="Names",y="Average")
```

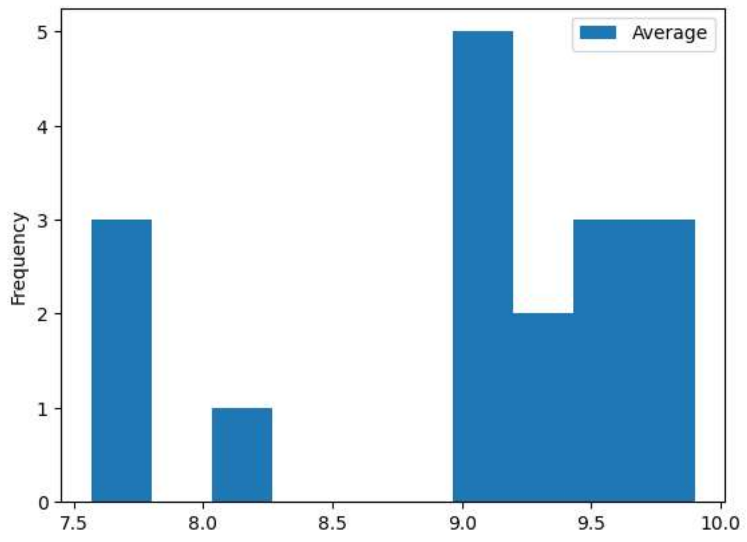
```
<Axes: xlabel='Names', ylabel='Average'>
```

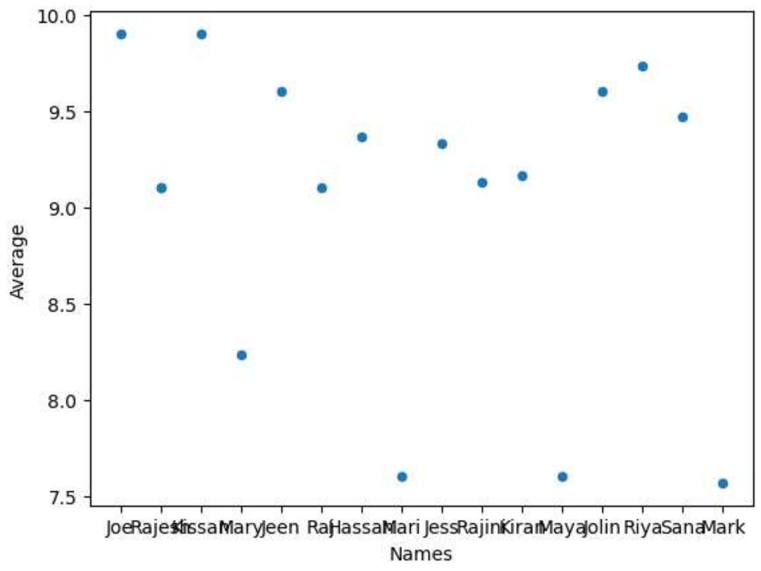




```
mydf.plot(kind="hist",x="Names",y="Average")
mydf.plot(kind="scatter",x="Names",y="Average")
```

```
<Axes: xlabel='Names', ylabel='Average'>
```

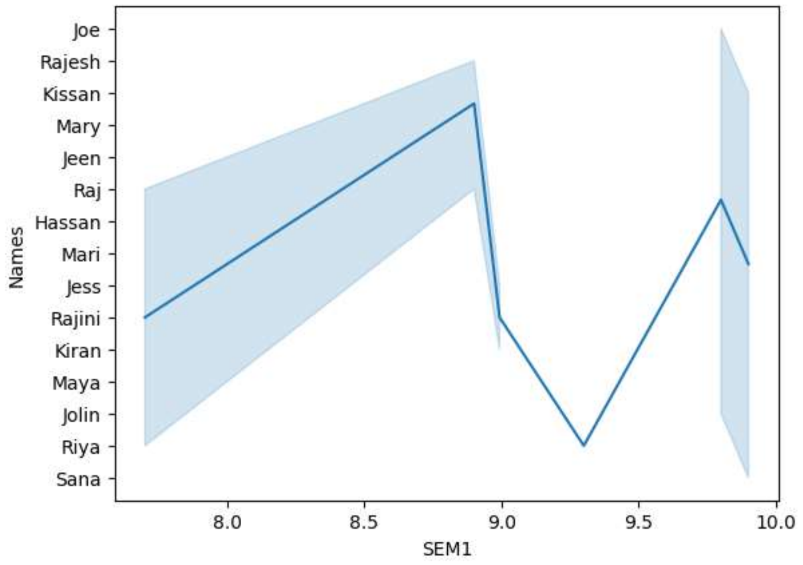




```
mydf.to_csv("mydf.csv")
```

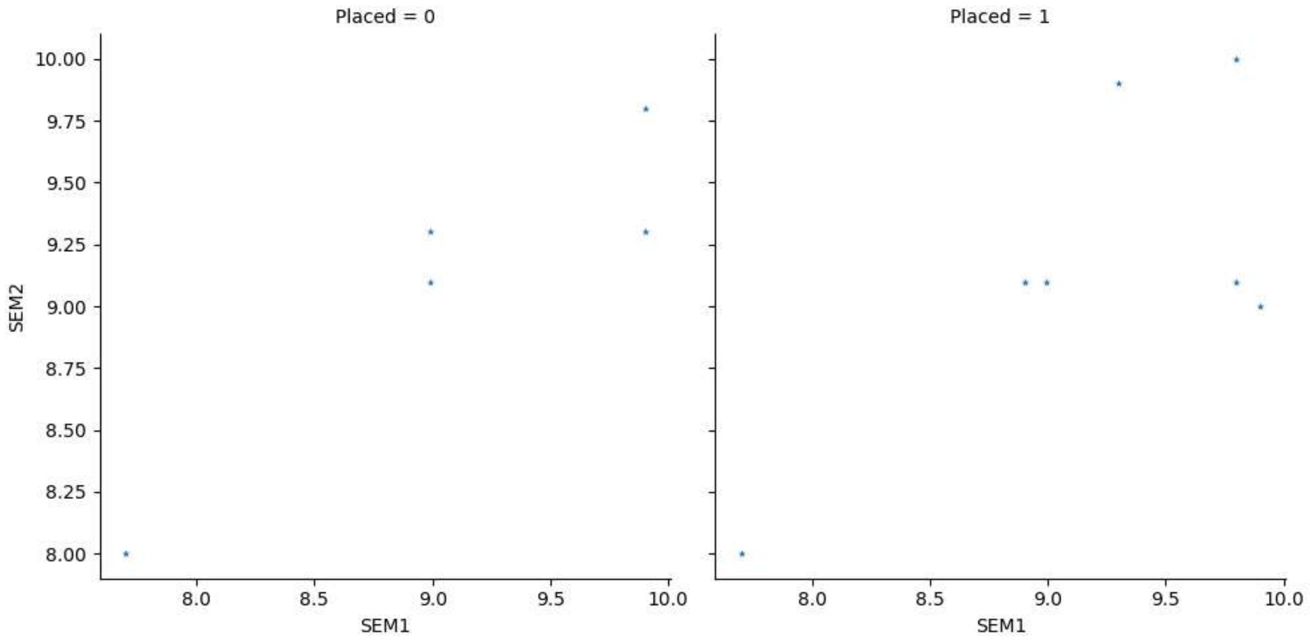
```
df=pd.read_csv("/content/mydf.csv")
```

```
import seaborn as sns
p1=sns.lineplot(y="Names",x="SEM1",data=df)
```



```
#col-->Graphs are separeted based on this col
sns.relplot(data=df,x="SEM1",y="SEM2",col="Placed",marker='*')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9f80e87700>
```



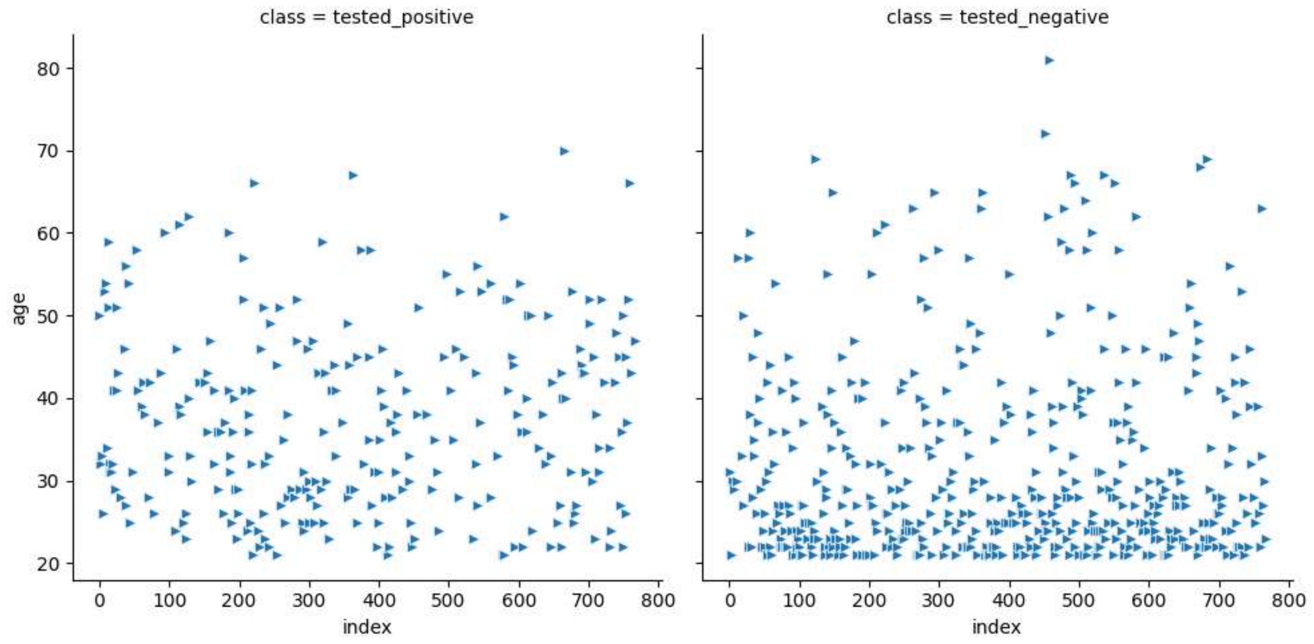
```
#create a relational plot using age as y axis and index as x axis separated by class
df1=pd.read_excel("/content/diabetes.xlsx")
df1['index']=range(0,768)
df1
```

	preg	plas	pres	skin	insu	mass	pedi	age	class	index
0	6	148	72	35	0	33.6	0.627	50	tested_positive	0
1	1	85	66	29	0	26.6	0.351	31	tested_negative	1
2	8	183	64	0	0	23.3	0.672	32	tested_positive	2
3	1	89	66	23	94	28.1	0.167	21	tested_negative	3
4	0	137	40	35	168	43.1	2.288	33	tested_positive	4
...	...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	tested_negative	763
764	2	122	70	27	0	36.8	0.340	27	tested_negative	764
765	5	121	72	23	112	26.2	0.245	30	tested_negative	765
766	1	126	60	0	0	30.1	0.349	47	tested_positive	766
767	1	93	70	31	0	30.4	0.315	23	tested_negative	767

768 rows × 10 columns

```
sns.relplot(data=df1,x="index",y="age",col="class",marker=">")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9fc8b6a320>
```



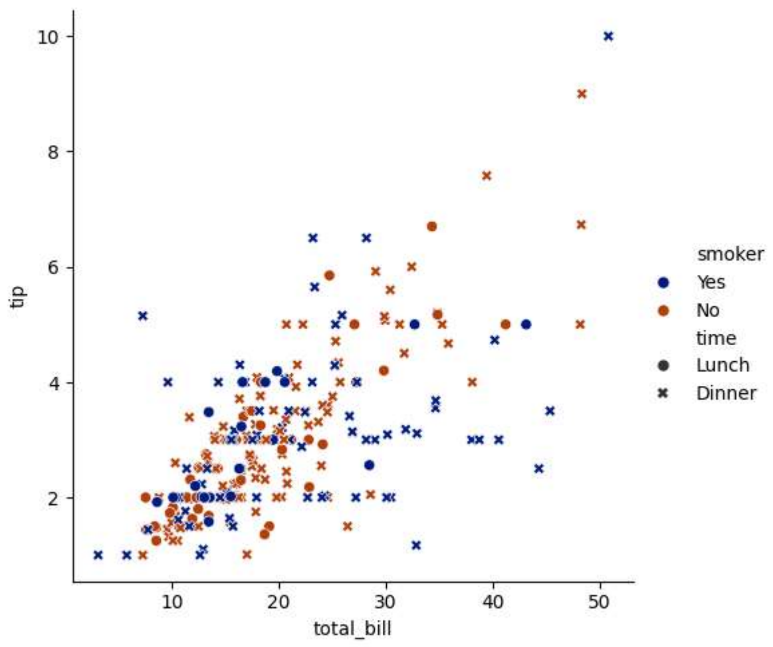
```
sns.load_dataset("tips")
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

```
tips=sns.load_dataset("tips")
sns.relplot(data=tips,x="total_bill",y="tip",hue="smoker",style="time",palette="dark")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f9f7af619f0>
```



```
dj=sns.load_dataset("dowjones")
dj
```

	Date	Price
0	1914.12.01	55.00

```
sns.relplot(data=dj,x="Date",y="Price",kind="line")
```