Use any 5 datasets from drive and display the following results

In [1]:
```
import numpy as np
import pandas as pd
```

a) Find mean, median, mode and describe

b) Find sum(), cumsum(), count, min and max values

c) Find covariance and correlation (spearman and pearsons)

D1

In [2]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\1_fiat500_VehicleSelection_Dataset.csv
df
```

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	l
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.6115598
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.241889
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.417
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.634609
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.495650
...	...	...	...	...	...	...	...	
1544	NaN	NaN	NaN	NaN	NaN	NaN	NaN	leng
1545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	con
1546	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Null valu
1547	NaN	NaN	NaN	NaN	NaN	NaN	NaN	fi
1548	NaN	NaN	NaN	NaN	NaN	NaN	NaN	sear

1549 rows × 11 columns

In [159]:
```
df1=df.fillna(value=0)
dat = df1[["ID","engine_power"]]
data=df1[["ID","engine_power"]]
datt=data[0:2]
```

In [119]:
```
print(dat.mean())
```

```
ID              764.035507
engine_power     51.535830
dtype: float64
```

In [120]:
```
print(dat.median())
```

```
ID              764.0
engine_power     51.0
dtype: float64
```

In [121]:
```
print(dat.mode())
```

```
    ID  engine_power
0  0.0          51.0
```

In [122]:
```
print(dat.describe())
```

```
                ID  engine_power
count  1549.000000    1549.00000
mean    764.035507      51.53583
std     447.241129       5.89909
min       0.000000       0.00000
25%     377.000000      51.00000
50%     764.000000      51.00000
75%    1151.000000      51.00000
max    1538.000000      77.00000
```

In [123]:
```
print(dat.sum())
```

```
ID              1183491.0
engine_power      79829.0
dtype: float64
```

In [124]:
```
print(dat.min())
```

```
ID              0.0
engine_power    0.0
dtype: float64
```

In [125]:
```
print(dat.max())
```

```
ID              1538.0
engine_power      77.0
dtype: float64
```

In [126]:
```
print(dat.count())
```

```
ID              1549
engine_power    1549
dtype: int64
```

In [127]:
```
print(dat.cumsum())
```

```
             ID  engine_power
0           1.0          51.0
1           3.0         102.0
2           6.0         176.0
3          10.0         227.0
4          15.0         300.0
...         ...           ...
1544  1183491.0       79829.0
1545  1183491.0       79829.0
1546  1183491.0       79829.0
1547  1183491.0       79829.0
1548  1183491.0       79829.0

[1549 rows x 2 columns]
```

In [154]:
```
from scipy.stats import spearmanr
print(spearmanr(dat))
```

```
SignificanceResult(statistic=0.018476888614293814, pvalue=0.4674243860814524)
```

D2

In [18]:
```
df2=pd.read_csv(r"C:\Users\user\Downloads\2_2015.csv")
df2
```

Out[18]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Fre
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.
...	...	...	...	...	...	...	...	...	
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.

158 rows × 12 columns

In [128]:
```
dat1=df2[["Happiness Rank","Happiness Score"]]
```

In [129]:
```
print(dat1.mean())
```

```
Happiness Rank     79.493671
Happiness Score     5.375734
dtype: float64
```

In [130]:
```
print(dat1.median())
```

```
Happiness Rank     79.5000
Happiness Score     5.2325
dtype: float64
```

In [131]:
```
print(dat1.mode())
```

```
   Happiness Rank  Happiness Score
0              82            5.192
```

In [132]:
```
print(dat1.describe())
```

```
       Happiness Rank  Happiness Score
count      158.000000       158.000000
mean        79.493671         5.375734
std         45.754363         1.145010
min          1.000000         2.839000
25%         40.250000         4.526000
50%         79.500000         5.232500
75%        118.750000         6.243750
max        158.000000         7.587000
```

In [133]:
```
print(dat1.min())
```

```
Happiness Rank     1.000
Happiness Score    2.839
dtype: float64
```

In [134]:
```
print(dat1.max())
```

```
Happiness Rank     158.000
Happiness Score      7.587
dtype: float64
```

In [135]:
```
print(dat1.count())
```

```
Happiness Rank     158
Happiness Score    158
dtype: int64
```

In [136]:
```
print(dat1.sum())
```

```
Happiness Rank     12560.000
Happiness Score      849.366
dtype: float64
```

In [87]:
```
print(dat1.cumsum())
```

```
     Happiness Rank  Happiness Score
0                 1            7.587
1                 3           15.148
2                 6           22.675
3                10           30.197
4                15           37.624
..              ...              ...
153           11934          837.276
154           12089          840.616
155           12245          843.622
156           12402          846.527
157           12560          849.366

[158 rows x 2 columns]
```

In [161]:
```
from scipy.stats import spearmanr
print(spearmanr(dat1))
```

```
SignificanceResult(statistic=-0.9999999999999999, pvalue=0.0)
```

D3

In [32]:
```
df3=pd.read_csv(r"C:\Users\user\Downloads\3_Fitness-1.csv")
df3
```

Out[32]:

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	A	5.62%	7.73%	6.16%	75
1	B	4.21%	17.27%	19.21%	160
2	C	9.83%	11.60%	5.17%	101
3	D	2.81%	21.91%	7.88%	127
4	E	25.28%	10.57%	11.82%	179
5	F	8.15%	16.24%	18.47%	167
6	G	18.54%	8.76%	17.49%	171
7	H	25.56%	5.93%	13.79%	170
8	Grand Total	100.00%	100.00%	100.00%	1150

In [41]:
```
print(df3.mean())
```

```
Sum of Total Sales    255.555556
dtype: float64

C:\Users\user\AppData\Local\Temp\ipykernel_6096\392603381.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.mean is deprecated. In a fu
ture version, it will default to False. In addition, specifying 'numeric_only
=None' is deprecated. Select only valid columns or specify the value of numer
ic_only to silence this warning.
  print(df3.mean())
```

In [40]:
```
print(df3.median())
```

```
Sum of Total Sales    167.0
dtype: float64

C:\Users\user\AppData\Local\Temp\ipykernel_6096\1320291346.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.median is deprecated. In a
future version, it will default to False. In addition, specifying 'numeric_on
ly=None' is deprecated. Select only valid columns or specify the value of num
eric_only to silence this warning.
  print(df3.median())
```

In [39]:
```
print(df3.mode())
```

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	A	100.00%	10.57%	100.00%	75
1	B	18.54%	100.00%	11.82%	101
2	C	2.81%	11.60%	13.79%	127
3	D	25.28%	16.24%	17.49%	160
4	E	25.56%	17.27%	18.47%	167
5	F	4.21%	21.91%	19.21%	170
6	G	5.62%	5.93%	5.17%	171
7	Grand Total	8.15%	7.73%	6.16%	179
8	H	9.83%	8.76%	7.88%	1150

In [38]:
```
print(df3.describe())
```

	Sum of Total Sales
count	9.000000
mean	255.555556
std	337.332963
min	75.000000
25%	127.000000
50%	167.000000
75%	171.000000
max	1150.000000

In [88]:
```
print(df3.min())
```

```
Row Labels                 A
Sum of Jan            100.00%
Sum of Feb             10.57%
Sum of Mar            100.00%
Sum of Total Sales         75
dtype: object
```

In [89]:
```
print(df3.max())
```

```
Row Labels                 H
Sum of Jan             9.83%
Sum of Feb             8.76%
Sum of Mar             7.88%
Sum of Total Sales      1150
dtype: object
```

In [90]:
```
print(df3.count())
```

```
Row Labels            9
Sum of Jan            9
Sum of Feb            9
Sum of Mar            9
Sum of Total Sales    9
dtype: int64
```

In [91]:
```
print(df3.sum())
```

```
Row Labels                                          ABCDEFGHGrand Total
Sum of Jan            5.62%4.21%9.83%2.81%25.28%8.15%18.54%25.56%100...
Sum of Feb            7.73%17.27%11.60%21.91%10.57%16.24%8.76%5.93%1...
Sum of Mar            6.16%19.21%5.17%7.88%11.82%18.47%17.49%13.79%1...
Sum of Total Sales                                                  2300
dtype: object
```

In [162]:
```
from scipy.stats import spearmanr
print(spearmanr(df3))
```

```
SignificanceResult(statistic=nan, pvalue=nan)
```

D4

In [37]:
```
df4=pd.read_csv(r"C:\Users\user\Downloads\4_drug200.csv")
df4
```

Out[37]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [141]:
```
dat3=df4[["Age","Na_to_K"]]
```

In [142]:
```
print(dat3.mean())
```

```
Age        44.315000
Na_to_K    16.084485
dtype: float64
```

In [143]:
```
print(dat3.mode())
```

```
    Age  Na_to_K
0  47.0   12.006
1   NaN   18.295
```

In [144]:
```
print(dat3.median())
```

```
Age        45.0000
Na_to_K    13.9365
dtype: float64
```

In [145]:
```
print(dat3.describe())
```

```
              Age     Na_to_K
count  200.000000  200.000000
mean    44.315000   16.084485
std     16.544315    7.223956
min     15.000000    6.269000
25%     31.000000   10.445500
50%     45.000000   13.936500
75%     58.000000   19.380000
max     74.000000   38.247000
```

In [146]:
```
print(dat3.min())
```

```
Age        15.000
Na_to_K     6.269
dtype: float64
```

In [147]:
```
print(dat3.max())
```

```
Age        74.000
Na_to_K    38.247
dtype: float64
```

In [148]:
```
print(dat3.count())
```

```
Age        200
Na_to_K    200
dtype: int64
```

In [149]:
```
print(dat3.sum())
```

```
Age        8863.000
Na_to_K    3216.897
dtype: float64
```

In [150]:
```
print(dat3.cumsum())
```

```
      Age   Na_to_K
0      23    25.355
1      70    38.448
2     117    48.562
3     145    56.360
4     206    74.403
..    ...       ...
195  8732  3169.628
196  8748  3181.634
197  8800  3191.528
198  8823  3205.548
199  8863  3216.897

[200 rows x 2 columns]
```

In [163]:
```
from scipy.stats import spearmanr
print(spearmanr(dat3))
```

```
SignificanceResult(statistic=-0.047273882688479915, pvalue=0.506220058138741
8)
```

D5

In [46]:
```
df5=pd.read_csv(r"C:\Users\user\Downloads\6_Salesworkload1.csv")
df5
```

Out[46]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLe
0	10.2016	1.0	United Kingdom	88253.0	London (I)	1.0	Dry	3184.764	
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	
...	...	...	...	...	...	...	...	...	
7653	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	
7654	06.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	
7655	06.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	
7656	06.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	
7657	06.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	

7658 rows × 14 columns

In [153]:
```
print(df5.mean())
```

```
Time index     5.000000e+00
StoreID        6.199522e+04
Dept_ID        9.470588e+00
HoursLease     2.203608e+01
Sales units    1.076471e+06
Turnover       3.721393e+06
Customer                NaN
dtype: float64

C:\Users\user\AppData\Local\Temp\ipykernel_6096\3502034520.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.mean is deprecated. In a fu
ture version, it will default to False. In addition, specifying 'numeric_only
=None' is deprecated. Select only valid columns or specify the value of numer
ic_only to silence this warning.
  print(df5.mean())
```

In [49]:
```
print(df5.median())
```

```
Time index          5.0
StoreID         75400.5
Dept_ID             9.0
HoursLease          0.0
Sales units    293230.0
Turnover       931957.5
Customer            NaN
dtype: float64

C:\Users\user\AppData\Local\Temp\ipykernel_6096\130194914.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.median is deprecated. In a
future version, it will default to False. In addition, specifying 'numeric_on
ly=None' is deprecated. Select only valid columns or specify the value of num
eric_only to silence this warning.
  print(df5.median())
```

In [50]:
```
print(df5.mode())
```

```
   MonthYear  Time index         Country  StoreID             City  Dept_ID
\
0    01.2017         1.0          France  12227.0      Aalborg (I)      1.0
1    02.2017         2.0         Germany  15552.0     Aalborg (II)      2.0
2    03.2017         3.0  United Kingdom  16927.0        Amsterdam      3.0
3    04.2017         4.0             NaN  17647.0          Antwerp      4.0
4    05.2017         5.0             NaN  18808.0    Barcelona (I)      5.0
5    06.2017         6.0             NaN  19000.0   Barcelona (II)      6.0
6    10.2016         7.0             NaN  19340.0       Berlin (I)      7.0
7    11.2016         8.0             NaN  19769.0      Berlin (II)      8.0
8    12.2016         9.0             NaN  20166.0           Bilbao      9.0
9        NaN         NaN             NaN  20891.0       Birmingham     11.0
10       NaN         NaN             NaN  22117.0          Bologna     12.0
11       NaN         NaN             NaN  23623.0         Bordeaux     13.0
12       NaN         NaN             NaN  29650.0             Brno     14.0
13       NaN         NaN             NaN  32949.0     Brussels (I)     15.0
14       NaN         NaN             NaN  34378.0    Brussels (II)     16.0
15       NaN         NaN             NaN  38560.0          Cologne     17.0
16       NaN         NaN             NaN  38976.0   Copenhagen (I)     18.0
```

In [51]:
```
print(df5.median())
```

```
Time index          5.0
StoreID         75400.5
Dept_ID             9.0
HoursLease          0.0
Sales units    293230.0
Turnover       931957.5
Customer            NaN
dtype: float64

C:\Users\user\AppData\Local\Temp\ipykernel_6096\130194914.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.median is deprecated. In a
future version, it will default to False. In addition, specifying 'numeric_on
ly=None' is deprecated. Select only valid columns or specify the value of num
eric_only to silence this warning.
  print(df5.median())
```

In [52]:
```
print(df5.describe())
```

```
        Time index       StoreID      Dept_ID   HoursLease   Sales units  \
count  7650.000000   7650.000000  7650.000000  7650.000000  7.650000e+03
mean      5.000000  61995.220000     9.470588    22.036078  1.076471e+06
std       2.582158  29924.581631     5.337429   133.299513  1.728113e+06
min       1.000000  12227.000000     1.000000     0.000000  0.000000e+00
25%       3.000000  29650.000000     5.000000     0.000000  5.457125e+04
50%       5.000000  75400.500000     9.000000     0.000000  2.932300e+05
75%       7.000000  87703.000000    14.000000     0.000000  9.175075e+05
max       9.000000  98422.000000    18.000000  3984.000000  1.124296e+07

           Turnover  Customer
count  7.650000e+03       0.0
mean   3.721393e+06       NaN
std    6.003380e+06       NaN
min    0.000000e+00       NaN
25%    2.726798e+05       NaN
50%    9.319575e+05       NaN
75%    3.264432e+06       NaN
max    4.271739e+07       NaN
```

In [111]:
```
print(df5.min())
```

```
MonthYear       - - - -
Time index            1.0
StoreID           12227.0
Dept_ID               1.0
HoursLease            0.0
Sales units           0.0
Turnover              0.0
Customer              NaN
dtype: object

C:\Users\user\AppData\Local\Temp\ipykernel_6096\1508775488.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.min is deprecated. In a fut
ure version, it will default to False. In addition, specifying 'numeric_only=
None' is deprecated. Select only valid columns or specify the value of numeri
c_only to silence this warning.
  print(df5.min())
```

In [112]:
```
print(df5.max())
```

```
MonthYear         12.2016
Time index            9.0
StoreID           98422.0
Dept_ID              18.0
HoursLease         3984.0
Sales units    11242955.0
Turnover       42717390.0
Customer              NaN
dtype: object

C:\Users\user\AppData\Local\Temp\ipykernel_6096\657196608.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.max is deprecated. In a fut
ure version, it will default to False. In addition, specifying 'numeric_only=
None' is deprecated. Select only valid columns or specify the value of numeri
c_only to silence this warning.
  print(df5.max())
```

In [113]:
```
print(df5.count())
```

```
MonthYear        7658
Time index       7650
Country          7650
StoreID          7650
City             7650
Dept_ID          7650
Dept. Name       7650
HoursOwn         7650
HoursLease       7650
Sales units      7650
Turnover         7650
Customer            0
Area (m2)        7650
Opening hours    7650
dtype: int64
```

In [114]:
```
print(df5.sum())
```

```
MonthYear      10.201610.201610.201610.201610.201610.201610.2...
Time index                                                38250.0
StoreID                                                474263433.0
Dept_ID                                                    72450.0
HoursLease                                                168576.0
Sales units                                            8235000965.0
Turnover                                              28468656015.0
Customer                                                     0.0
dtype: object

C:\Users\user\AppData\Local\Temp\ipykernel_6096\1783206477.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.sum is deprecated. In a fut
ure version, it will default to False. In addition, specifying 'numeric_only=
None' is deprecated. Select only valid columns or specify the value of numeri
c_only to silence this warning.
  print(df5.sum())
```