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

In [2]:
```
df=pd.read_csv("1_2015-1.csv")
```

In [3]:
```
import warnings
warnings.filterwarnings('ignore')
print(df.mean())
```

```
Happiness Rank                  79.493671
Happiness Score                  5.375734
Standard Error                   0.047885
Economy (GDP per Capita)         0.846137
Family                           0.991046
Health (Life Expectancy)         0.630259
Freedom                          0.428615
Trust (Government Corruption)    0.143422
Generosity                       0.237296
Dystopia Residual                2.098977
dtype: float64
```

In [4]:
```
print(df.median())
```

```
Happiness Rank                  79.500000
Happiness Score                  5.232500
Standard Error                   0.043940
Economy (GDP per Capita)         0.910245
Family                           1.029510
Health (Life Expectancy)         0.696705
Freedom                          0.435515
Trust (Government Corruption)    0.107220
Generosity                       0.216130
Dystopia Residual                2.095415
dtype: float64
```

In [5]:
```
print(df.mode())
```

```
         Country             Region  Happiness Rank  Happiness Score  \
0    Afghanistan  Sub-Saharan Africa            82.0            5.192
1        Albania                 NaN             NaN              NaN
2        Algeria                 NaN             NaN              NaN
3         Angola                 NaN             NaN              NaN
4      Argentina                 NaN             NaN              NaN
..           ...                 ...             ...              ...
153    Venezuela                 NaN             NaN              NaN
154      Vietnam                 NaN             NaN              NaN
155        Yemen                 NaN             NaN              NaN
156       Zambia                 NaN             NaN              NaN
157     Zimbabwe                 NaN             NaN              NaN

     Standard Error  Economy (GDP per Capita)   Family  \
0           0.03751                   0.00000  0.00000
1           0.03780                   0.01530  0.13995
2           0.04394                   0.01604  0.30285
3           0.04934                   0.06940  0.35386
4           0.05051                   0.07120  0.38174
..              ...                       ...      ...
153             NaN                   1.45900  1.34043
154             NaN                   1.52186  1.34951
155             NaN                   1.55422  1.36058
156             NaN                   1.56391  1.36948
157             NaN                   1.69042  1.40223

     Health (Life Expectancy)  Freedom  Trust (Government Corruption)  \
0                     0.92356  0.00000                        0.32524
1                         NaN  0.07699                            NaN
2                         NaN  0.09245                            NaN
3                         NaN  0.10081                            NaN
4                         NaN  0.10384                            NaN
..                        ...      ...                            ...
153                       NaN  0.65821                            NaN
154                       NaN  0.65980                            NaN
155                       NaN  0.66246                            NaN
156                       NaN  0.66557                            NaN
157                       NaN  0.66973                            NaN

     Generosity  Dystopia Residual
0       0.00000            0.32858
1       0.00199            0.65429
2       0.02641            0.67042
```

```
3       0.05444           0.67108
4       0.05547           0.89991
..          ...               ...
153     0.51535           3.10712
154     0.51752           3.17728
155     0.51912           3.19131
156     0.57630           3.26001
157     0.79588           3.60214

[158 rows x 12 columns]
```

In [6]:
```
print(df.describe())
```

```
       Happiness Rank  Happiness Score  Standard Error  \
count      158.000000       158.000000      158.000000
mean        79.493671         5.375734        0.047885
std         45.754363         1.145010        0.017146
min          1.000000         2.839000        0.018480
25%         40.250000         4.526000        0.037268
50%         79.500000         5.232500        0.043940
75%        118.750000         6.243750        0.052300
max        158.000000         7.587000        0.136930

       Economy (GDP per Capita)      Family  Health (Life Expectancy)  \
count                158.000000  158.000000                158.000000
mean                   0.846137    0.991046                  0.630259
std                    0.403121    0.272369                  0.247078
min                    0.000000    0.000000                  0.000000
25%                    0.545808    0.856823                  0.439185
50%                    0.910245    1.029510                  0.696705
75%                    1.158448    1.214405                  0.811013
max                    1.690420    1.402230                  1.025250

          Freedom  Trust (Government Corruption)  Generosity  \
count  158.000000                     158.000000  158.000000
mean     0.428615                       0.143422    0.237296
std      0.150693                       0.120034    0.126685
min      0.000000                       0.000000    0.000000
25%      0.328330                       0.061675    0.150553
50%      0.435515                       0.107220    0.216130
75%      0.549092                       0.180255    0.309883
max      0.669730                       0.551910    0.795880

       Dystopia Residual
count         158.000000
mean            2.098977
std             0.553550
min             0.328580
25%             1.759410
50%             2.095415
75%             2.462415
max             3.602140
```

In [7]:
```
print(df.sum())
```

```
Country                          SwitzerlandIcelandDenmarkNorwayCanadaFinlandNe...
Region                           Western EuropeWestern EuropeWestern EuropeWest...
Happiness Rank                                                           12560
Happiness Score                                                        849.366
Standard Error                                                         7.56579
Economy (GDP per Capita)                                             133.68968
Family                                                               156.58526
Health (Life Expectancy)                                              99.58098
Freedom                                                               67.72116
Trust (Government Corruption)                                         22.66065
Generosity                                                            37.49269
Dystopia Residual                                                    331.63833
dtype: object
```

In [8]:
```
print(df.cumsum())
```

```
                                          Country  \
0                                     Switzerland
1                                 SwitzerlandIceland
2                             SwitzerlandIcelandDenmark
3                         SwitzerlandIcelandDenmarkNorway
4                   SwitzerlandIcelandDenmarkNorwayCanada
..                                                ...
153  SwitzerlandIcelandDenmarkNorwayCanadaFinlandNe...
154  SwitzerlandIcelandDenmarkNorwayCanadaFinlandNe...
155  SwitzerlandIcelandDenmarkNorwayCanadaFinlandNe...
156  SwitzerlandIcelandDenmarkNorwayCanadaFinlandNe...
157  SwitzerlandIcelandDenmarkNorwayCanadaFinlandNe...

                                            Region  Happiness Rank  \
0                                   Western Europe               1
1                       Western EuropeWestern Europe               3
2            Western EuropeWestern EuropeWestern Europe               6
3    Western EuropeWestern EuropeWestern EuropeWest...              10
4    Western EuropeWestern EuropeWestern EuropeWest...              15
..                                                 ...             ...
153  Western EuropeWestern EuropeWestern EuropeWest...           11934
154  Western EuropeWestern EuropeWestern EuropeWest...           12089
155  Western EuropeWestern EuropeWestern EuropeWest...           12245
156  Western EuropeWestern EuropeWestern EuropeWest...           12402
157  Western EuropeWestern EuropeWestern EuropeWest...           12560

     Happiness Score  Standard Error  Economy (GDP per Capita)     Family  \
0              7.587         0.03411                   1.39651    1.34951
1             15.148         0.08295                   2.69883    2.75174
2             22.675         0.11623                   4.02431    4.11232
3             30.197         0.15503                   5.48331    5.44327
4             37.624         0.19056                   6.80960    6.76588
..               ...             ...                       ...        ...
153          837.276         7.32523                132.51585  155.20069
154          840.616         7.36179                132.80250  155.55455
155          843.622         7.41194                133.46570  156.02944
156          846.527         7.49852                133.48100  156.44531
157          849.366         7.56579                133.68968  156.58526

     Health (Life Expectancy)   Freedom  Trust (Government Corruption)  \
0                     0.94143   0.66557                        0.41978
1                     1.88927   1.29434                        0.56123
2                     2.76391   1.94372                        1.04480
```

```
3                    3.64912   2.61345                        1.40983
4                    4.55475   3.24642                        1.73940
..                       ...       ...                            ...
153                 98.03156  66.59679                       22.18356
154                 98.35066  67.08129                       22.26366
155                 99.07259  67.23813                       22.45272
156                 99.29655  67.35663                       22.55334
157                 99.58098  67.72116                       22.66065

     Generosity  Dystopia Residual
0       0.29678            2.51738
1       0.73308            5.21939
2       1.07447            7.71143
3       1.42146           10.17674
4       1.87957           12.62850
..          ...                ...
153    36.47422          326.27619
154    36.65682          327.90947
155    37.12861          328.23805
156    37.32588          330.07107
157    37.49269          331.63833

[158 rows x 12 columns]
```

In [9]:
```
print(df.count())
```

```
Country                          158
Region                           158
Happiness Rank                   158
Happiness Score                  158
Standard Error                   158
Economy (GDP per Capita)         158
Family                           158
Health (Life Expectancy)         158
Freedom                          158
Trust (Government Corruption)    158
Generosity                       158
Dystopia Residual                158
dtype: int64
```

In [10]:
```
print(df.max())
```

```
Country                             Zimbabwe
Region                        Western Europe
Happiness Rank                           158
Happiness Score                        7.587
Standard Error                       0.13693
Economy (GDP per Capita)             1.69042
Family                               1.40223
Health (Life Expectancy)             1.02525
Freedom                              0.66973
Trust (Government Corruption)        0.55191
Generosity                           0.79588
Dystopia Residual                    3.60214
dtype: object
```

In [11]:
```
print(df.min())
```

```
Country                                 Afghanistan
Region                     Australia and New Zealand
Happiness Rank                                     1
Happiness Score                                2.839
Standard Error                               0.01848
Economy (GDP per Capita)                         0.0
Family                                           0.0
Health (Life Expectancy)                         0.0
Freedom                                          0.0
Trust (Government Corruption)                    0.0
Generosity                                       0.0
Dystopia Residual                            0.32858
dtype: object
```

In [12]:
```
list(df)
```

Out[12]:
```
['Country',
 'Region',
 'Happiness Rank',
 'Happiness Score',
 'Standard Error',
 'Economy (GDP per Capita)',
 'Family',
 'Health (Life Expectancy)',
 'Freedom',
 'Trust (Government Corruption)',
 'Generosity',
 'Dystopia Residual']
```

In [13]:
```
from numpy import cov
data1=df['Happiness Rank'][0:100]
data2=df['Happiness Score'][0:100]
covariance=cov(data1,data2)
print(covariance)
```

```
[[ 8.41020101e+02 -2.29934817e+01]
 [-2.29934817e+01  6.38401267e-01]]
```

In [14]:
```
from scipy.stats import pearsonr
data1=df['Happiness Rank'][0:100]
data2=df['Happiness Score'][0:100]
corr,_=pearsonr(data1,data2)
print(corr)
```

```
-0.9923267072800683
```

In [15]:
```
from scipy.stats import spearmanr
data1=df['Happiness Rank'][0:100]
data2=df['Happiness Score'][0:100]
corr,_=spearmanr(data1,data2)
print(corr)
```

```
-1.0
```

In [16]:
```
df1=pd.read_csv("3_Fitness.csv")
```

In [17]:
```
print(df1.mean())
```

```
JAN             101.090909
FEB              98.909091
MAR              94.909091
APR             103.818182
MAY              96.727273
JUN              85.090909
TOTAL SALES     579.818182
Unnamed: 8     3193.000000
Unnamed: 9             NaN
Unnamed: 10            NaN
dtype: float64
```

In [18]:
```
print(df1.median())
```

```
JAN              65.0
FEB              54.0
MAR              50.0
APR              59.0
MAY              56.0
JUN              55.0
TOTAL SALES     322.0
Unnamed: 8     3193.0
Unnamed: 9        NaN
Unnamed: 10       NaN
dtype: float64
```

In [19]:
```
print(df1.mode())
```

```
          SALESMAN   JAN   FEB   MAR   APR   MAY   JUN  TOTAL SALES  \
0              ANU  35.0  35.0  49.0  45.0  35.0  25.0        210.0
1             BABU   NaN   NaN   NaN   NaN  56.0  30.0        220.0
2          CHANDRU   NaN   NaN   NaN   NaN   NaN  55.0        247.0
3            DAVID   NaN   NaN   NaN   NaN   NaN  73.0        258.0
4          EINSTEIN   NaN   NaN   NaN   NaN   NaN   NaN        319.0
5            FAROOK   NaN   NaN   NaN   NaN   NaN   NaN        322.0
6          GOWTHAM   NaN   NaN   NaN   NaN   NaN   NaN        351.0
7         HARSHITH   NaN   NaN   NaN   NaN   NaN   NaN        388.0
8           INIYAN   NaN   NaN   NaN   NaN   NaN   NaN        412.0
9             JOHN   NaN   NaN   NaN   NaN   NaN   NaN        462.0
10  MONTHLY SALES   NaN   NaN   NaN   NaN   NaN   NaN       3189.0

    Unnamed: 8  Unnamed: 9  Unnamed: 10  \
0       3193.0         NaN          NaN
1          NaN         NaN          NaN
2          NaN         NaN          NaN
3          NaN         NaN          NaN
4          NaN         NaN          NaN
5          NaN         NaN          NaN
6          NaN         NaN          NaN
7          NaN         NaN          NaN
8          NaN         NaN          NaN
9          NaN         NaN          NaN
10         NaN         NaN          NaN

                                        Unnamed: 11
0                      1. Individual Sales using Sum()
1   2. Find the pattern trend using conditional fo...
2   3. Analyze using Pivot table as column percentage
3                                                33
4                             4. Insert Pivot charts
5   5. Rank() - returns the rank of a given value ...
6                                                NaN
7                                                NaN
8                                                NaN
9                                                NaN
10                                               NaN
```

In [20]:
```
print(df1.describe())
```

```
              JAN         FEB         MAR         APR         MAY         JUN  \
count   11.000000   11.000000   11.000000   11.000000   11.000000   11.000000
mean   101.090909   98.909091   94.909091  103.818182   96.727273   85.090909
std    152.263886  148.884153  142.770763  155.277054  145.500578  128.347540
min     29.000000   25.000000   15.000000   45.000000   25.000000   25.000000
25%     35.000000   40.000000   47.000000   49.000000   40.000000   30.000000
50%     65.000000   54.000000   50.000000   59.000000   56.000000   55.000000
75%     76.000000   75.500000   71.500000   66.500000   69.500000   69.000000
max    556.000000  544.000000  522.000000  571.000000  532.000000  468.000000

       TOTAL SALES  Unnamed: 8  Unnamed: 9  Unnamed: 10
count    11.000000         1.0         0.0          0.0
mean    579.818182      3193.0         NaN          NaN
std     869.142775         NaN         NaN          NaN
min     210.000000      3193.0         NaN          NaN
25%     252.500000      3193.0         NaN          NaN
50%     322.000000      3193.0         NaN          NaN
75%     400.000000      3193.0         NaN          NaN
max    3189.000000      3193.0         NaN          NaN
```

In [21]:
```
print(df1.sum())
```

```
JAN            1112.0
FEB            1088.0
MAR            1044.0
APR            1142.0
MAY            1064.0
JUN             936.0
TOTAL SALES    6378.0
Unnamed: 8     3193.0
Unnamed: 9        0.0
Unnamed: 10       0.0
dtype: float64
```

In [22]:
```
df.dropna()
```

Out[22]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811
...	...	...	...	...	...	...	...	...	...	...	...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681

158 rows × 12 columns

In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Country                        158 non-null    object 
 1   Region                         158 non-null    object 
 2   Happiness Rank                 158 non-null    int64  
 3   Happiness Score                158 non-null    float64
 4   Standard Error                 158 non-null    float64
 5   Economy (GDP per Capita)       158 non-null    float64
 6   Family                         158 non-null    float64
 7   Health (Life Expectancy)       158 non-null    float64
 8   Freedom                        158 non-null    float64
 9   Trust (Government Corruption)  158 non-null    float64
 10  Generosity                     158 non-null    float64
 11  Dystopia Residual              158 non-null    float64
dtypes: float64(9), int64(1), object(2)
memory usage: 14.9+ KB
```

In [24]:
```
print(df1.cumsum())
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\numpy\core\fromnumeric.py:57, in _wrapfunc(obj, method, *args, **
kwds)
     56 try:
---> 57     return bound(*args, **kwds)
     58 except TypeError:
     59     # A TypeError occurs if the object does have such a method in its
     60     # class, but its signature is not identical to that of NumPy's. This
   (...)
     64     # Call _wrapit from within the except clause to ensure a potential
     65     # exception has a traceback chain.

TypeError: can only concatenate str (not "float") to str

During handling of the above exception, another exception occurred:

TypeError                                 Traceback (most recent call last)
Cell In[24], line 1
```

In [26]:
```
print(df1.count())
```

```
SALESMAN       11
JAN            11
FEB            11
MAR            11
APR            11
MAY            11
JUN            11
TOTAL SALES    11
Unnamed: 8      1
Unnamed: 9      0
Unnamed: 10     0
Unnamed: 11     6
dtype: int64
```

In [27]:
```
print(df1.min())
```

```
JAN              29.0
FEB              25.0
MAR              15.0
APR              45.0
MAY              25.0
JUN              25.0
TOTAL SALES     210.0
Unnamed: 8     3193.0
Unnamed: 9        NaN
Unnamed: 10       NaN
dtype: float64
```

In [28]:
```
print(df1.max())
```

```
JAN             556.0
FEB             544.0
MAR             522.0
APR             571.0
MAY             532.0
JUN             468.0
TOTAL SALES    3189.0
Unnamed: 8     3193.0
Unnamed: 9        NaN
Unnamed: 10       NaN
dtype: float64
```

In [29]:
```
print(df1.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   SALESMAN     11 non-null     object 
 1   JAN          11 non-null     float64
 2   FEB          11 non-null     float64
 3   MAR          11 non-null     float64
 4   APR          11 non-null     float64
 5   MAY          11 non-null     float64
 6   JUN          11 non-null     float64
 7   TOTAL SALES  11 non-null     float64
 8   Unnamed: 8   1 non-null      float64
 9   Unnamed: 9   0 non-null      float64
 10  Unnamed: 10  0 non-null      float64
 11  Unnamed: 11  6 non-null      object 
dtypes: float64(10), object(2)
memory usage: 1.3+ KB
None
```

In [30]:
```
df1['JAN'].count()
```

Out[30]: 11

In [31]:
```
from numpy import cov
data1=df1['JAN'][0:10]
data2=df1['MAY'][0:10]
covariance=cov(data1,data2)
print(covariance)
```

```
[[467.37777778 258.64444444]
 [258.64444444 366.17777778]]
```

In [32]:
```
from scipy.stats import pearsonr
data1=df1['JAN'][0:10]
data2=df1['MAY'][0:10]
corr,_=pearsonr(data1,data2)
print(corr)
```

```
0.625205993029724
```

In [33]:
```
from scipy.stats import spearmanr
data1=df1['JAN'][0:10]
data2=df1['MAY'][0:10]
corr,_=spearmanr(data1,data2)
print(corr)
```

```
0.7033672033257948
```

In [34]:
```
df2=pd.read_csv("4_Drug200.csv")
```

In [35]:
```
print(df2.mean())
```

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

In [36]:
```
print(df2.median())
```

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

In [37]:
```
print(df2.mode())
```

```
    Age  Sex    BP Cholesterol  Na_to_K   Drug
0  47.0    M  HIGH        HIGH   12.006  drugY
1   NaN  NaN   NaN         NaN   18.295    NaN
```

In [38]:
```
print(df2.sum())
```

```
Age                                                          8863
Sex            FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...
BP             HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...
Cholesterol    HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...
Na_to_K                                                3216.897
Drug           drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...
dtype: object
```

In [39]:
```
print(df2.cumsum())
```

```
      Age                                                      Sex  \
0      23                                                        F
1      70                                                       FM
2     117                                                      FMM
3     145                                                     FMMF
4     206                                                    FMMFF
..    ...                                                      ...
195  8732  FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...
196  8748  FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...
197  8800  FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...
198  8823  FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...
199  8863  FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...

                                                        BP  \
0                                                     HIGH
1                                                   HIGHLOW
2                                                 HIGHLOWLOW
3                                               HIGHLOWLOWNORMAL
4                                             HIGHLOWLOWNORMALLOW
..                                                       ...
195  HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...
196  HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...
197  HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...
198  HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...
199  HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...

                                          Cholesterol   Na_to_K  \
0                                                HIGH    25.355
1                                             HIGHHIGH    38.448
2                                          HIGHHIGHHIGH    48.562
3                                       HIGHHIGHHIGHHIGH    56.360
4                                    HIGHHIGHHIGHHIGHHIGH    74.403
..                                                  ...       ...
195  HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...  3169.628
196  HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...  3181.634
197  HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...  3191.528
198  HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...  3205.548
199  HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...  3216.897

                                                   Drug
0                                                 drugY
1                                              drugYdrugC
2                                           drugYdrugCdrugC
```

```
3                                  drugYdrugCdrugCdrugX
4                                drugYdrugCdrugCdrugXdrugY
..                                                     ...
195  drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...
196  drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...
197  drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...
198  drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...
199  drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...

[200 rows x 6 columns]
```

In [40]:
```
print(df2.count())
```

```
Age            200
Sex            200
BP             200
Cholesterol    200
Na_to_K        200
Drug           200
dtype: int64
```

In [41]:
```
print(df2.min())
```

```
Age              15
Sex               F
BP             HIGH
Cholesterol    HIGH
Na_to_K       6.269
Drug          drugA
dtype: object
```

In [42]:
```
print(df2.max())
```

```
Age              74
Sex               M
BP           NORMAL
Cholesterol  NORMAL
Na_to_K      38.247
Drug          drugY
dtype: object
```

In [43]:
```
list(df2)
```

Out[43]:
```
['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug']
```

In [44]:
```
print(df2.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          200 non-null    int64  
 1   Sex          200 non-null    object 
 2   BP           200 non-null    object 
 3   Cholesterol  200 non-null    object 
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
None
```