day-6

July 26, 2023

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

1 Dataset 1

[2]:
```
data = pd.read_csv(r"C:\Users\SHREYAS\Downloads\dataset.csv")
data
```

[2]:
```
         Country                           Region  Happiness Rank  \
0    Switzerland                   Western Europe               1
1        Iceland                   Western Europe               2
2        Denmark                   Western Europe               3
3         Norway                   Western Europe               4
4         Canada                    North America               5
..           …                              …             …
153       Rwanda               Sub-Saharan Africa             154
154        Benin               Sub-Saharan Africa             155
155        Syria  Middle East and Northern Africa             156
156      Burundi               Sub-Saharan Africa             157
157         Togo               Sub-Saharan Africa             158

     Happiness Score  Standard Error  Economy (GDP per Capita)   Family  \
0              7.587         0.03411                   1.39651  1.34951
1              7.561         0.04884                   1.30232  1.40223
2              7.527         0.03328                   1.32548  1.36058
3              7.522         0.03880                   1.45900  1.33095
4              7.427         0.03553                   1.32629  1.32261
..               …             …                       …      …
153            3.465         0.03464                   0.22208  0.77370
154            3.340         0.03656                   0.28665  0.35386
155            3.006         0.05015                   0.66320  0.47489
156            2.905         0.08658                   0.01530  0.41587
157            2.839         0.06727                   0.20868  0.13995

     Health (Life Expectancy)  Freedom  Trust (Government Corruption)  \
0                     0.94143  0.66557                        0.41978
1                     0.94784  0.62877                        0.14145
```

```
2                     0.87464  0.64938                              0.48357
3                     0.88521  0.66973                              0.36503
4                     0.90563  0.63297                              0.32957
..                        …        …                                    …
153                   0.42864  0.59201                              0.55191
154                   0.31910  0.48450                              0.08010
155                   0.72193  0.15684                              0.18906
156                   0.22396  0.11850                              0.10062
157                   0.28443  0.36453                              0.10731

     Generosity  Dystopia Residual
0       0.29678            2.51738
1       0.43630            2.70201
2       0.34139            2.49204
3       0.34699            2.46531
4       0.45811            2.45176
..          …                  …
153     0.22628            0.67042
154     0.18260            1.63328
155     0.47179            0.32858
156     0.19727            1.83302
157     0.16681            1.56726

[158 rows x 12 columns]
```

Mean, Median, Mode, Describe

[4]:
```
df = data[["Happiness Score","Standard Error"]]
df
```

[4]:
```
     Happiness Score  Standard Error
0              7.587         0.03411
1              7.561         0.04884
2              7.527         0.03328
3              7.522         0.03880
4              7.427         0.03553
..               …               …
153            3.465         0.03464
154            3.340         0.03656
155            3.006         0.05015
156            2.905         0.08658
157            2.839         0.06727

[158 rows x 2 columns]
```

[5]:
```
print(df.mean())
```

```
Happiness Score    5.375734
```

```
Standard Error     0.047885
dtype: float64
```

[7]:
```
print(df.median())
```

```
Happiness Score    5.23250
Standard Error     0.04394
dtype: float64
```

[8]:
```
print(df.mode())
```

```
   Happiness Score  Standard Error
0            5.192         0.03751
1              NaN         0.03780
2              NaN         0.04394
3              NaN         0.04934
4              NaN         0.05051
```

[9]:
```
print(df.describe())
```

```
       Happiness Score  Standard Error
count       158.000000      158.000000
mean          5.375734        0.047885
std           1.145010        0.017146
min           2.839000        0.018480
25%           4.526000        0.037268
50%           5.232500        0.043940
75%           6.243750        0.052300
max           7.587000        0.136930
```

[10]:
```
print(df.sum())
```

```
Happiness Score    849.36600
Standard Error       7.56579
dtype: float64
```

[11]:
```
print(df.cumsum())
```

```
     Happiness Score  Standard Error
0              7.587         0.03411
1             15.148         0.08295
2             22.675         0.11623
3             30.197         0.15503
4             37.624         0.19056
..               …             …
153          837.276         7.32523
154          840.616         7.36179
155          843.622         7.41194
156          846.527         7.49852
157          849.366         7.56579
```

```
[158 rows x 2 columns]
```

```
[12]: print(df.count())
```

```
Happiness Score    158
Standard Error     158
dtype: int64
```

```
[13]: print(df.max())
```

```
Happiness Score    7.58700
Standard Error     0.13693
dtype: float64
```

```
[14]: print(df.min())
```

```
Happiness Score    2.83900
Standard Error     0.01848
dtype: float64
```

```
[25]: from numpy import cov
      from scipy.stats import spearmanr,pearsonr
```

```
[17]: print(df.cov())
```

```
                 Happiness Score  Standard Error
Happiness Score         1.311048       -0.003480
Standard Error         -0.003480        0.000294
```

```
[23]: print(spearmanr(df))
```

```
SignificanceResult(statistic=-0.21519846171732626, pvalue=0.006619286429972024)
```

```
[30]: print(pearsonr(data["Happiness Score"],data["Standard Error"]))
```

```
PearsonRResult(statistic=-0.17725380900494767, pvalue=0.025878684792533208)
```

2 Dataset 2

```
[31]: data1 = pd.read_csv(r"C:\Users\SHREYAS\Downloads\dataset2.csv")
      data1
```

```
[31]:      ID   model  engine_power  age_in_days        km  previous_owners  \
0     1.0  lounge          51.0        882.0   25000.0              1.0
1     2.0     pop          51.0       1186.0   32500.0              1.0
2     3.0   sport          74.0       4658.0  142228.0              1.0
3     4.0  lounge          51.0       2739.0  160000.0              1.0
4     5.0     pop          73.0       3074.0  106880.0              1.0
…     …       …             …            …         …                …
```

```
1554  NaN     NaN           NaN          NaN       NaN                NaN
1555  NaN     NaN           NaN          NaN       NaN                NaN
1556  NaN     NaN           NaN          NaN       NaN                NaN
1557  NaN     NaN           NaN          NaN       NaN                NaN
1558  NaN     NaN           NaN          NaN       NaN                NaN

            lat          lon       price  Unnamed: 9  Unnamed: 10 Unnamed: 11  \
0     44.907242  8.611559868        8900         NaN          NaN         NaN
1     45.666359  12.24188995        8800         NaN          NaN         NaN
2     45.503300     11.41784        4200         NaN          NaN         NaN
3     40.633171  17.63460922        6000         NaN          NaN         NaN
4     41.903221  12.49565029        5700         NaN          NaN         NaN
…          …          …         …         …          …         …
1554        NaN    averageif       44028         NaN          NaN         NaN
1555        NaN       counta        1538         NaN          NaN         NaN
1556        NaN         left         lou         NaN          NaN         NaN
1557        NaN        right         ort         NaN          NaN         NaN
1558        NaN         date  26-11-2002         NaN          NaN         NaN

     Unnamed: 12
0            NaN
1            NaN
2            NaN
3            NaN
4            NaN
…          …
1554         NaN
1555         NaN
1556         NaN
1557         NaN
1558         NaN

[1559 rows x 13 columns]
```

[64]:

```
df1 = data1[["engine_power","ID",]]
df
```

[64]:

```
     Happiness Score  Standard Error
0              7.587         0.03411
1              7.561         0.04884
2              7.527         0.03328
3              7.522         0.03880
4              7.427         0.03553
..               …             …
153            3.465         0.03464
154            3.340         0.03656
155            3.006         0.05015
```

```
156              2.905         0.08658
157              2.839         0.06727

[158 rows x 2 columns]
```

```
[83]: df1.dropna()
```

```
[83]:      engine_power      ID
      0              51.0     1.0
      1              51.0     2.0
      2              74.0     3.0
      3              51.0     4.0
      4              73.0     5.0
      …                 …       …
      1533           51.0  1534.0
      1534           74.0  1535.0
      1535           51.0  1536.0
      1536           51.0  1537.0
      1537           51.0  1538.0

      [1538 rows x 2 columns]
```

```
[66]: print(df1.mean())
```

```
engine_power     51.904421
ID              769.500000
dtype: float64
```

```
[67]: print(df1.median())
```

```
engine_power     51.0
ID              769.5
dtype: float64
```

```
[68]: print(df1.mode())
```

```
      engine_power      ID
0             51.0     1.0
1              NaN     2.0
2              NaN     3.0
3              NaN     4.0
4              NaN     5.0
…                …       …
1533           NaN  1534.0
1534           NaN  1535.0
1535           NaN  1536.0
1536           NaN  1537.0
1537           NaN  1538.0
```

```
[1538 rows x 2 columns]
```

[69]:
```
print(df1.cumsum())
```

```
      engine_power    ID
0             51.0   1.0
1            102.0   3.0
2            176.0   6.0
3            227.0  10.0
4            300.0  15.0
…              …   …
1554           NaN   NaN
1555           NaN   NaN
1556           NaN   NaN
1557           NaN   NaN
1558           NaN   NaN

[1559 rows x 2 columns]
```

[70]:
```
print(df1.describe())
```

```
       engine_power           ID
count   1538.000000  1538.000000
mean      51.904421   769.500000
std        3.988023   444.126671
min       51.000000     1.000000
25%       51.000000   385.250000
50%       51.000000   769.500000
75%       51.000000  1153.750000
max       77.000000  1538.000000
```

[71]:
```
print(df.sum())
```

```
Happiness Score    849.36600
Standard Error       7.56579
dtype: float64
```

[72]:
```
print(df1.count())
```

```
engine_power    1538
ID              1538
dtype: int64
```

[73]:
```
print(df1.min())
```

```
engine_power    51.0
ID               1.0
dtype: float64
```

[74]:
```
print(df1.max())
```

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

[75]:
```
print(df1.cov())
```

```
              engine_power             ID
engine_power     15.904327     -60.325634
ID              -60.325634  197248.500000
```

[86]:
```
print(spearmanr(df1))
```

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

[88]:
```
print(pearsonr(df1,))
```

```
  Cell In[88], line 1
    print(pearsonr(df1,09))
                       ^
SyntaxError: leading zeros in decimal integer literals are not permitted; use a
 ↪0o prefix for octal integers
```

3 Dataset 3

[89]:
```
data2 = pd.read_csv(r"C:\Users\SHREYAS\Downloads\3_Fitness-1.csv")
data2
```

[89]:
```
    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
```

[92]:
```
df2 = data2["Sum of Total Sales"]
df2
```

[92]:
```
0      75
1     160
2     101
3     127
4     179
```

```
5     167
6     171
7     170
8    1150
Name: Sum of Total Sales, dtype: int64
```

```
[93]: print(df2.mean())
```

```
255.55555555555554
```

```
[94]: print(df2.median())
```

```
167.0
```

```
[95]: print(df2.mode())
```

```
0      75
1     101
2     127
3     160
4     167
5     170
6     171
7     179
8    1150
Name: Sum of Total Sales, dtype: int64
```

```
[96]: print(df2.describe())
```

```
count       9.000000
mean      255.555556
std       337.332963
min        75.000000
25%       127.000000
50%       167.000000
75%       171.000000
max      1150.000000
Name: Sum of Total Sales, dtype: float64
```

```
[97]: print(df2.sum())
```

```
2300
```

```
[98]: print(df2.cumsum())
```

```
0      75
1     235
2     336
3     463
4     642
```

```
5     809
6     980
7    1150
8    2300
Name: Sum of Total Sales, dtype: int64
```

```
[99]: print(df2.count())
```

```
9
```

```
[100]: print(df2.min())
```

```
75
```

```
[101]: print(df2.max())
```

```
1150
```

```
[105]: print(cov(df2,df2))
```

```
[[113793.52777778 113793.52777778]
 [113793.52777778 113793.52777778]]
```

```
[107]: print(spearmanr(df2,df2))
```

```
SignificanceResult(statistic=1.0, pvalue=0.0)
```

```
[109]: print(pearsonr(df2,df2))
```

```
PearsonRResult(statistic=1.0, pvalue=0.0)
```

4 Dataset 4

```
[111]: data3 = pd.read_csv(r"C:\Users\SHREYAS\Downloads\4_drug200.csv")
       data3
```

```
[111]:     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 x 6 columns]
```

[112]:

```python
df3 = data3[["Age","Na_to_K"]]
df3
```

[112]:

```
     Age  Na_to_K
0     23   25.355
1     47   13.093
2     47   10.114
3     28    7.798
4     61   18.043
..   …       …
195   56   11.567
196   16   12.006
197   52    9.894
198   23   14.020
199   40   11.349

[200 rows x 2 columns]
```

[113]:

```python
print(df3.mean())
```

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

[114]:

```python
print(df3.median())
```

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

[115]:

```python
print(df3.mode())
```

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

[116]:

```python
print(df3.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
```

[117]:

```python
print(df3.count())
```

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

```
[118]: print(df3.sum())
```

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

```
[119]: print(df3.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]
```

```
[120]: print(df3.min())
```

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

```
[121]: print(df3.max())
```

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

```
[122]: print(df3.cov())
```

```
                Age    Na_to_K
Age      273.714347  -7.543752
Na_to_K   -7.543752  52.185533
```

```
[124]: print(spearmanr(df3))
```

```
SignificanceResult(statistic=-0.047273882688479915, pvalue=0.5062200581387418)
```

```
[129]: print(pearsonr(df3["Age"],df3["Na_to_K"]))
```

```
PearsonRResult(statistic=-0.0631194972677259, pvalue=0.37457563990343007)
```

5 Dataset 5

[131]:
```
data4 = pd.read_csv(r"C:\Users\SHREYAS\Downloads\6_Salesworkload1.csv")
data4
```

[131]:
```
     MonthYear  Time index         Country  StoreID        City  Dept_ID  \
0      10.2016         1.0  United Kingdom  88253.0  London (I)      1.0
1      10.2016         1.0  United Kingdom  88253.0  London (I)      2.0
2      10.2016         1.0  United Kingdom  88253.0  London (I)      3.0
3      10.2016         1.0  United Kingdom  88253.0  London (I)      4.0
4      10.2016         1.0  United Kingdom  88253.0  London (I)      5.0
…          …         …             …      …        …       …
7653   06.2017         9.0          Sweden  29650.0  Gothenburg     12.0
7654   06.2017         9.0          Sweden  29650.0  Gothenburg     16.0
7655   06.2017         9.0          Sweden  29650.0  Gothenburg     11.0
7656   06.2017         9.0          Sweden  29650.0  Gothenburg     17.0
7657   06.2017         9.0          Sweden  29650.0  Gothenburg     18.0

               Dept. Name  HoursOwn  HoursLease  Sales units    Turnover  \
0                     Dry  3184.764         0.0     398560.0   1226244.0
1                  Frozen  1582.941         0.0      82725.0    387810.0
2                   other    47.205         0.0     438400.0    654657.0
3                    Fish  1623.852         0.0     309425.0    499434.0
4      Fruits & Vegetables  1759.173         0.0     165515.0    329397.0
…                       …         …          …         …           …
7653             Checkout  6322.323         0.0    3886530.0  14538825.0
7654    Customer Services  4270.479         0.0        245.0         0.0
7655             Delivery         0         0.0          0.0         0.0
7656               others  2224.929         0.0        245.0         0.0
7657                  all   39652.2         0.0    3886530.0  15056214.0

      Customer Area (m2) Opening hours
0          NaN    953.04        Type A
1          NaN    720.48        Type A
2          NaN    966.72        Type A
3          NaN   1053.36        Type A
4          NaN   1053.36        Type A
…         …       …           …
7653       NaN       #NV        Type A
7654       NaN       #NV        Type A
7655       NaN       #NV        Type A
7656       NaN       #NV        Type A
7657       NaN       #NV        Type A

[7658 rows x 14 columns]
```

[140]:

```
df4 = data4[["Dept_ID","Sales units"]]
df4
```

[140]:

```
      Dept_ID  Sales units
0         1.0     398560.0
1         2.0      82725.0
2         3.0     438400.0
3         4.0     309425.0
4         5.0     165515.0
…         …          …
7653     12.0    3886530.0
7654     16.0        245.0
7655     11.0          0.0
7656     17.0        245.0
7657     18.0    3886530.0

[7658 rows x 2 columns]
```

[141]:

```
print(df4.mean())
```

```
Dept_ID        9.470588e+00
Sales units    1.076471e+06
dtype: float64
```

[142]:

```
print(df4.median())
```

```
Dept_ID             9.0
Sales units    293230.0
dtype: float64
```

[136]:

```
print(df4.mode())
```

```
    Dept_ID HoursOwn
0       1.0   47.205
1       2.0      NaN
2       3.0      NaN
3       4.0      NaN
4       5.0      NaN
5       6.0      NaN
6       7.0      NaN
7       8.0      NaN
8       9.0      NaN
9      11.0      NaN
10     12.0      NaN
11     13.0      NaN
12     14.0      NaN
13     15.0      NaN
14     16.0      NaN
15     17.0      NaN
```

```
16     18.0      NaN
```

[143]:
```
print(df4.mode())
```

```
    Dept_ID  Sales units
0       1.0          0.0
1       2.0          NaN
2       3.0          NaN
3       4.0          NaN
4       5.0          NaN
5       6.0          NaN
6       7.0          NaN
7       8.0          NaN
8       9.0          NaN
9      11.0          NaN
10     12.0          NaN
11     13.0          NaN
12     14.0          NaN
13     15.0          NaN
14     16.0          NaN
15     17.0          NaN
16     18.0          NaN
```

[144]:
```
print(df4.describe())
```

```
           Dept_ID   Sales units
count  7650.000000  7.650000e+03
mean      9.470588  1.076471e+06
std       5.337429  1.728113e+06
min       1.000000  0.000000e+00
25%       5.000000  5.457125e+04
50%       9.000000  2.932300e+05
75%      14.000000  9.175075e+05
max      18.000000  1.124296e+07
```

[145]:
```
print(df4.sum())
```

```
Dept_ID        7.245000e+04
Sales units    8.235001e+09
dtype: float64
```

[146]:
```
print(df4.cumsum())
```

```
      Dept_ID   Sales units
0         1.0  3.985600e+05
1         3.0  4.812850e+05
2         6.0  9.196850e+05
3        10.0  1.229110e+06
4        15.0  1.394625e+06
…       …           …
```

```
7653  72388.0  8.231114e+09
7654  72404.0  8.231114e+09
7655  72415.0  8.231114e+09
7656  72432.0  8.231114e+09
7657  72450.0  8.235001e+09

[7658 rows x 2 columns]
```

[147]:
```
print(df4.count())
```

```
Dept_ID        7650
Sales units    7650
dtype: int64
```

[148]:
```
print(df4.min())
```

```
Dept_ID        1.0
Sales units    0.0
dtype: float64
```

[149]:
```
print(df4.max())
```

```
Dept_ID              18.0
Sales units    11242955.0
dtype: float64
```

[151]:
```
print(df4.cov())
```

```
                  Dept_ID   Sales units
Dept_ID      2.848815e+01  2.645877e+06
Sales units  2.645877e+06  2.986375e+12
```

[152]:
```
print(spearmanr(df4))
```

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

[160]:
```
print(pearsonr(df2,df2))
```

```
PearsonRResult(statistic=1.0, pvalue=0.0)
```

[ ]:
```

```