Day 3 - Statistics - Covariance and Correlation_by_Virat Tiwari

October 29, 2023

1 Covariance and Correlation

```
[1]: # Seaborn is a well-known Python library for data visualization that offers a␣
     ↪user-friendly interface for producing visually appealing and informative␣
     ↪statistical graphics.

     import seaborn as sns
```

```
[3]: # sns.load_dataset ( ) function is used importing built in dataset

     df=sns.load_dataset("healthexp")
     df.head()
```

```
[3]:    Year        Country  Spending_USD  Life_Expectancy
     0  1970        Germany       252.311             70.6
     1  1970         France       192.143             72.2
     2  1970  Great Britain       123.993             71.9
     3  1970          Japan       150.437             72.0
     4  1970            USA       326.961             70.9
```

```
[8]: # NOTE - We can not find the covariance of cateegorical data so we should first␣
     ↪import numpy as np for getting the covariance

     # NumPy stands for numeric python which is a python package for the computation␣
     ↪and processing of the multidimensional and single dimensional array elements

     import numpy as np
```

```
[6]: # cov-  cov stands for the covariance
     # This case or df.cov ( ) function is used for getting the covariance

     df.cov()
```

```
/tmp/ipykernel_370/1545644723.py:1: FutureWarning: The default value of
numeric_only in DataFrame.cov is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.cov()
```

```
[6]:                    Year   Spending_USD  Life_Expectancy
     Year            201.098848   2.571883e+04        41.915454
     Spending_USD  25718.827373   4.817761e+06      4166.800912
     Life_Expectancy   41.915454   4.166801e+03        10.733902
```

```
[10]: # Pearson correlation cofficient
      # This case or df.corr (method=" " ) function is used for getting the pearson␣
       ↪correlation coffcient

      df.corr(method="pearson")
```

/tmp/ipykernel_370/2979612414.py:3: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.corr(method="pearson")

```
[10]:                    Year   Spending_USD  Life_Expectancy
     Year             1.000000      0.826273         0.902175
     Spending_USD     0.826273      1.000000         0.579430
     Life_Expectancy  0.902175      0.579430         1.000000
```

```
[17]: # This is another way to find the correlation

      df.corr(method="spearman")
```

/tmp/ipykernel_370/2145751946.py:3: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.corr(method="spearman")

```
[17]:                    Year   Spending_USD  Life_Expectancy
     Year             1.000000      0.931598         0.896117
     Spending_USD     0.931598      1.000000         0.747407
     Life_Expectancy  0.896117      0.747407         1.000000
```

```
[18]: df1=sns.load_dataset("tips")
      df1.head()
```

```
[18]:    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
```

```
[19]: df.cov()
```

/tmp/ipykernel_370/1545644723.py:1: FutureWarning: The default value of
numeric_only in DataFrame.cov is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.cov()

```
[19]:                       Year  Spending_USD  Life_Expectancy
      Year             201.098848  2.571883e+04        41.915454
      Spending_USD   25718.827373  4.817761e+06      4166.800912
      Life_Expectancy    41.915454  4.166801e+03        10.733902
```

```
[20]: df.corr(method="spearman")
```

/tmp/ipykernel_370/3994303778.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.corr(method="spearman")

```
[20]:                      Year  Spending_USD  Life_Expectancy
      Year             1.000000      0.931598         0.896117
      Spending_USD     0.931598      1.000000         0.747407
      Life_Expectancy  0.896117      0.747407         1.000000
```

THANK YOU SO MUCH !!

VIRAT TIWARI :)