python-stats

April 15, 2023

```
[1]: data= [23,24,32,45,12,43,67,45,32,56,32]
```

```
[2]: data
```

```
[2]: [23, 24, 32, 45, 12, 43, 67, 45, 32, 56, 32]
```

```
[3]: import pandas as pd
     data2= pd.read_csv("https://raw.githubusercontent.com/sunnysavita10/
      ↪Statistics-With-Python-TheCompleteGuide/main/Iris.csv")
```

```
[4]: data2
```

```
[4]:       Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
     0      1            5.1           3.5            1.4           0.2
     1      2            4.9           3.0            1.4           0.2
     2      3            4.7           3.2            1.3           0.2
     3      4            4.6           3.1            1.5           0.2
     4      5            5.0           3.6            1.4           0.2
     ..   …              …             …              …             …
     145  146            6.7           3.0            5.2           2.3
     146  147            6.3           2.5            5.0           1.9
     147  148            6.5           3.0            5.2           2.0
     148  149            6.2           3.4            5.4           2.3
     149  150            5.9           3.0            5.1           1.8

                 Species
     0       Iris-setosa
     1       Iris-setosa
     2       Iris-setosa
     3       Iris-setosa
     4       Iris-setosa
     ..              …
     145  Iris-virginica
     146  Iris-virginica
     147  Iris-virginica
     148  Iris-virginica
     149  Iris-virginica
```

```
[150 rows x 6 columns]
```

[5]:

```
data3= pd.read_csv("https://raw.githubusercontent.com/sunnysavita10/
 ↪Statistics-With-Python-TheCompleteGuide/main/Titanic.csv")
```

[6]:

```
data3
```

[6]:

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …         …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …   …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q
```

```
[891 rows x 12 columns]
```

```
[7]: data
```

```
[7]: [23, 24, 32, 45, 12, 43, 67, 45, 32, 56, 32]
```

```
[8]: data_copy = data.copy()
```

```
[9]: data_copy.sort()
```

```
[10]: #pandas
      #numpy
      #matplotlin and seaborn
      #scipy
      #statsmodel
      #statistics
```

```
[11]: data
```

```
[11]: [23, 24, 32, 45, 12, 43, 67, 45, 32, 56, 32]
```

```
[12]: data_copy = data.copy()
```

```
[13]: data_copy.sort()
```

```
[14]: data_copy
```

```
[14]: [12, 23, 24, 32, 32, 32, 43, 45, 45, 56, 67]
```

```
[15]: import numpy as np
      np.mean(data)
```

```
[15]: 37.36363636363637
```

```
[16]: np.median(data)
```

```
[16]: 32.0
```

```
[17]: np.mean(data2['SepalLengthCm'])
```

```
[17]: 5.843333333333334
```

```
[18]: import  statistics
```

```
[19]: statistics.mode(data)
```

```
[19]: 32
```

[20]:

```
## how to calculate mean with code
```

[21]:

```
def mean(data):
    sum = 0
    for i in data:
        sum = sum+i
    mean = sum/len(data)
    return mean
```

[22]:

```
from scipy import stats as st
st.mode(data)
```

```
/tmp/ipykernel_78/3794622683.py:2: FutureWarning: Unlike other reduction
functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically
preserves the axis it acts along. In SciPy 1.11.0, this behavior will change:
the default value of `keepdims` will become False, the `axis` over which the
statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  st.mode(data)
```

[22]:

```
ModeResult(mode=array([32]), count=array([3]))
```

[23]:

```
data_copy.append(150)
```

[24]:

```
data_copy
```

[24]:

```
[12, 23, 24, 32, 32, 32, 43, 45, 45, 56, 67, 150]
```

[25]:

```
np.mean(data_copy)
```

[25]:

```
46.75
```

[26]:

```
data_copy2 = data.copy()
```

[27]:

```
data_copy2.append(75)
```

[28]:

```
data_copy2
```

[28]:

```
[23, 24, 32, 45, 12, 43, 67, 45, 32, 56, 32, 75]
```

[29]:

```
np.mean(data_copy2)
```

[29]:

```
40.5
```

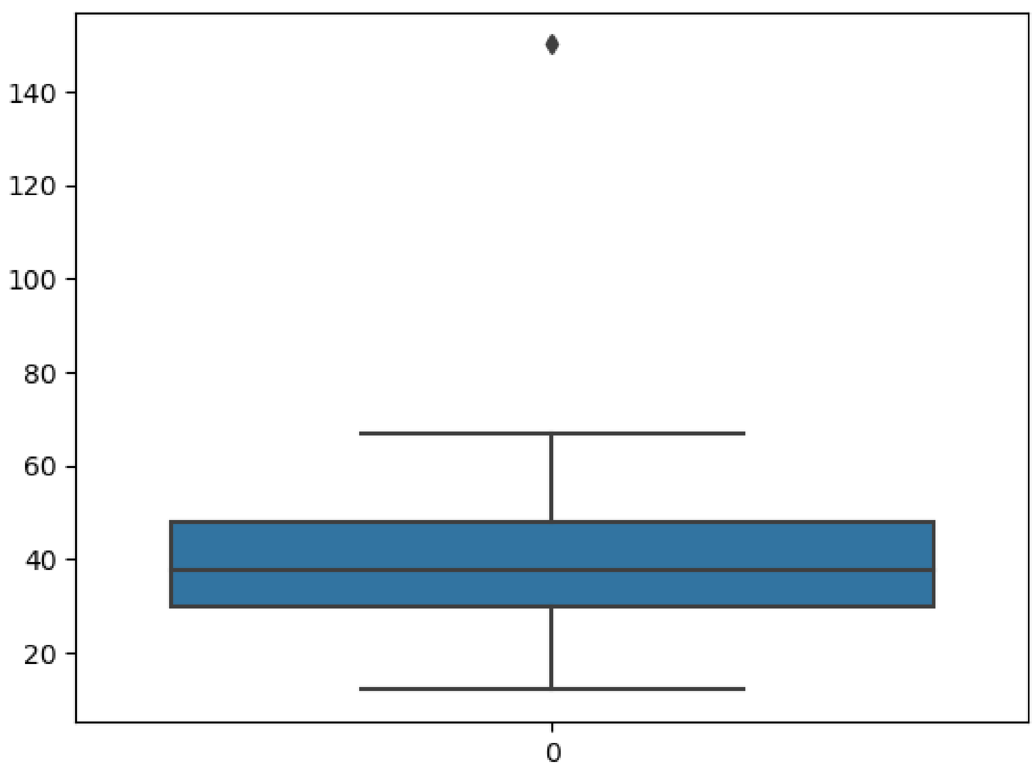
[30]:

```
np.median(data_copy)
```

[30]:

```
37.5
```

```
[31]: np.median(data_copy2)

[31]: 37.5

[ ]:

[32]: # Dispersion of data

[33]: np.percentile(data,[25])

[33]: array([28.])

[34]: np.percentile(data,[50])

[34]: array([32.])

[35]: data_copy.pop()

[35]: 150

[36]: np.percentile(data,[25,50,75,100])

[36]: array([28., 32., 45., 67.])

[37]: ## q1,q2,q3,q4  min & max

      ## TQR = q3-q1

[38]: data_copy.append(150)

[39]: data_copy

[39]: [12, 23, 24, 32, 32, 32, 43, 45, 45, 56, 67, 150]

[40]: import seaborn as sns
      sns.boxplot(data_copy)

[40]: <AxesSubplot: >
```



```
[41]: # q1,q2,q3,q4

      # IQR = Q3-Q1

      # LOWER FENCE= q1-IQR*1.5
      # UPPER FENCE= q3+IQR*1.5
```

```
[44]: data= [23,24,32,45,12,43,67,45,32,56,32]
```

```
[43]: data
```

```
[43]: [23, 24, 32, 45, 12, 43, 67, 45, 32, 56, 32]
```

```
[46]: # varience
      np.var(data)
```

```
[46]: 226.23140495867773
```

```
[47]: # standar deviation
      np.std(data)
```

```
[47]: 15.040990823701666
```

```
[49]: np.random.choice(data) ## Finding random variable
```

```
[49]: 45
```

```
[50]: np.random.choice(data,size=3)
```

```
[50]: array([23, 45, 43])
```

```
[51]: ## Find out 5 sampling technique of the sampling and implement it with the help␣
       ↪of python?
```

```
[52]: data2
```

```
[52]:        Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
      0       1            5.1           3.5            1.4           0.2
      1       2            4.9           3.0            1.4           0.2
      2       3            4.7           3.2            1.3           0.2
      3       4            4.6           3.1            1.5           0.2
      4       5            5.0           3.6            1.4           0.2
      ..    …            …          …           …          …
      145   146            6.7           3.0            5.2           2.3
      146   147            6.3           2.5            5.0           1.9
      147   148            6.5           3.0            5.2           2.0
      148   149            6.2           3.4            5.4           2.3
      149   150            5.9           3.0            5.1           1.8

                  Species
      0       Iris-setosa
      1       Iris-setosa
      2       Iris-setosa
      3       Iris-setosa
      4       Iris-setosa
      ..              …
      145  Iris-virginica
      146  Iris-virginica
      147  Iris-virginica
      148  Iris-virginica
      149  Iris-virginica

      [150 rows x 6 columns]
```

```
[53]: data2.sample()
```

```
[53]:     Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
      34  35            4.9           3.1            1.5           0.1  Iris-setosa
```

```
[54]: data2.sample(15)
```

[54]:

```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
13    14            4.3           3.0            1.1           0.1
109  110            7.2           3.6            6.1           2.5
70    71            5.9           3.2            4.8           1.8
107  108            7.3           2.9            6.3           1.8
148  149            6.2           3.4            5.4           2.3
94    95            5.6           2.7            4.2           1.3
52    53            6.9           3.1            4.9           1.5
37    38            4.9           3.1            1.5           0.1
139  140            6.9           3.1            5.4           2.1
6      7            4.6           3.4            1.4           0.3
138  139            6.0           3.0            4.8           1.8
56    57            6.3           3.3            4.7           1.6
18    19            5.7           3.8            1.7           0.3
67    68            5.8           2.7            4.1           1.0
100  101            6.3           3.3            6.0           2.5

             Species
13       Iris-setosa
109   Iris-virginica
70   Iris-versicolor
107   Iris-virginica
148   Iris-virginica
94   Iris-versicolor
52   Iris-versicolor
37       Iris-setosa
139   Iris-virginica
6        Iris-setosa
138   Iris-virginica
56   Iris-versicolor
18       Iris-setosa
67   Iris-versicolor
100   Iris-virginica
```

[58]:

```python
## python code for varience

def var(data):
    n=len(data)
    mean= sum(data)/n
    deviation=[(x-mean)** 2 for x in data]
    var = sum(deviation)/n-1
    return var
```

[59]:

```python
var(data)
```

[59]:

```
225.23140495867773
```

[61]:

```
def var(data):
    n=len(data)
    mean= sum(data)/n
    deviation=[(x-mean)** 2 for x in data]
    var = sum(deviation)/n
    return var
```

[62]:

```
var(data)
```

[62]:

```
226.23140495867773
```

[63]:

```
np.var(data)
```

[63]:

```
226.23140495867773
```

[66]:

```
import statistics
statistics.variance(data)
```

[66]:

```
248.85454545454544
```

[67]:

```
statistics.pvariance(data)
```

[67]:

```
226.23140495867767
```

[68]:

```
import math

math.sqrt(statistics.variance(data))
```

[68]:

```
15.775124261144361
```

[70]:

```
len(data)
```

[70]:

```
11
```

[71]:

```
len(data)-1
```

[71]:

```
10
```

[72]:

```
## correlation and  covarience
```

[80]:

```
import seaborn as sns
df=sns.load_dataset('tips')
```

[74]:

```
df.head()
```

[74]:

```
   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
```

[75]:

```
df.corr()
```

```
/tmp/ipykernel_78/1134722465.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()
```

[75]:

```
            total_bill       tip      size
total_bill    1.000000  0.675734  0.598315
tip           0.675734  1.000000  0.489299
size          0.598315  0.489299  1.000000
```

[76]:

```
df.info()
```

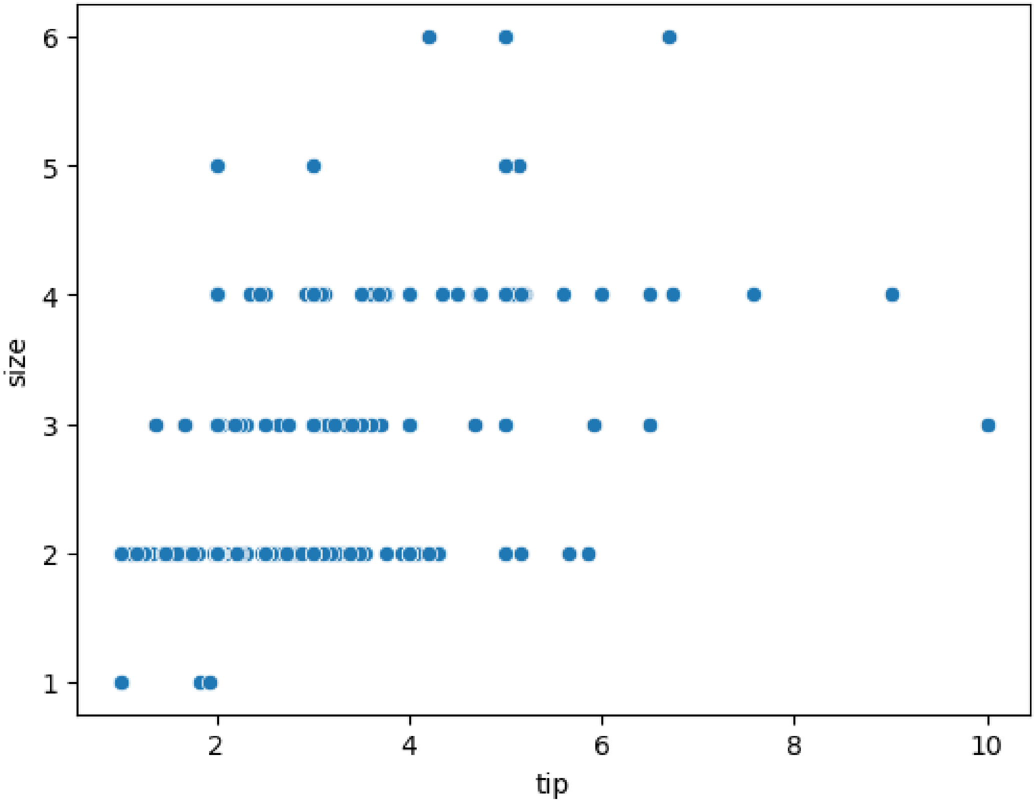
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   total_bill  244 non-null    float64
 1   tip         244 non-null    float64
 2   sex         244 non-null    category
 3   smoker      244 non-null    category
 4   day         244 non-null    category
 5   time        244 non-null    category
 6   size        244 non-null    int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.4 KB
```

[81]:

```
sns.scatterplot(x=df['tip'],y=df['size'])
```

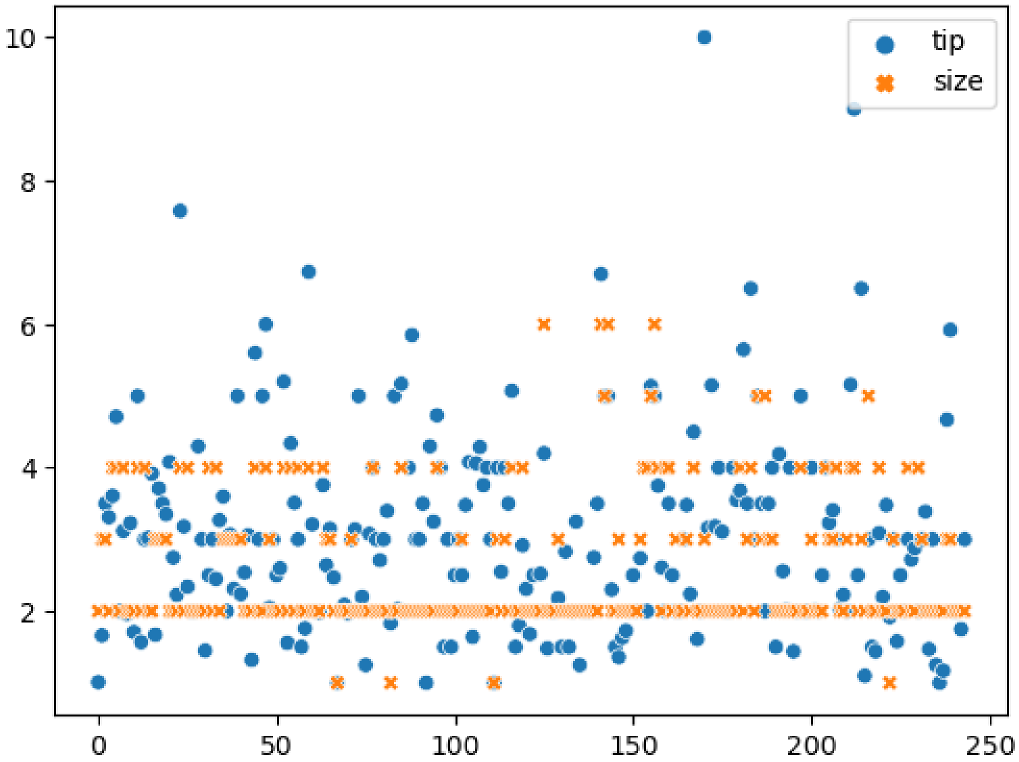
[81]:

```
<AxesSubplot: xlabel='tip', ylabel='size'>
```



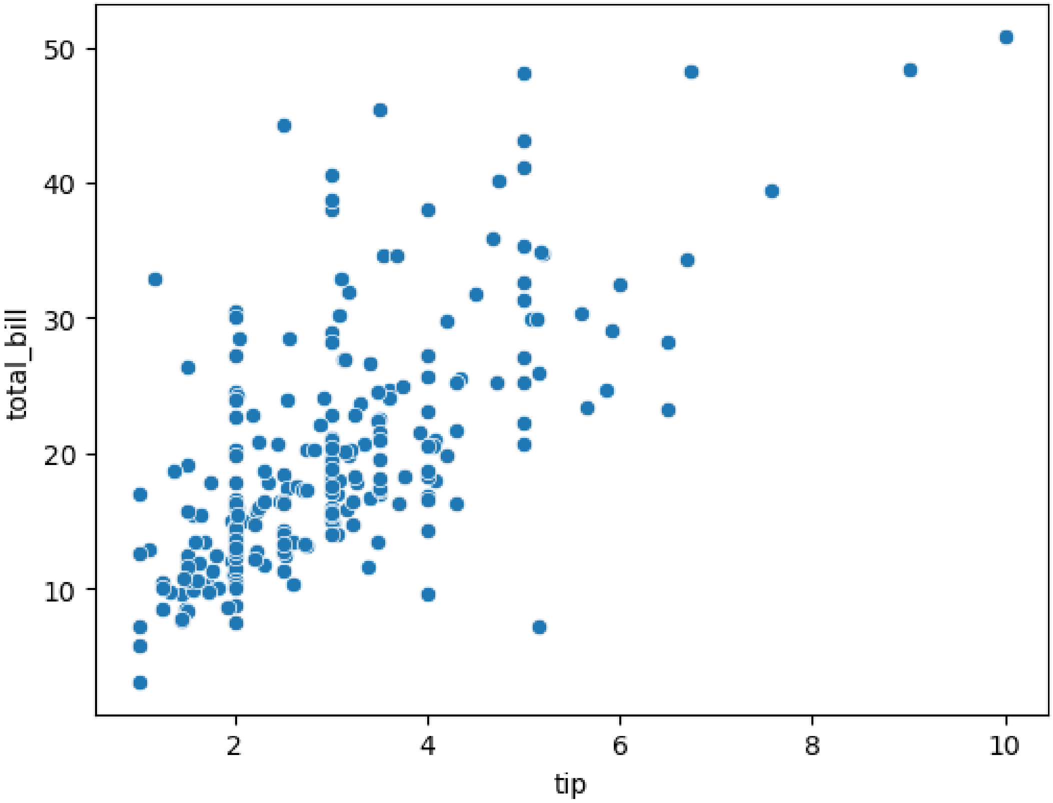
```
[84]: sns.scatterplot(df[['tip','size']])
```

```
[84]: <AxesSubplot: >
```



[85]:
```
sns.scatterplot(x=df['tip'],y=df['total_bill'])
```

[85]:
```
<AxesSubplot: xlabel='tip', ylabel='total_bill'>
```



[86]:
```
df.cov()
```

```
/tmp/ipykernel_78/1545644723.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()
```

[86]:
```
             total_bill       tip      size
total_bill    79.252939  8.323502  5.065983
tip            8.323502  1.914455  0.643906
size           5.065983  0.643906  0.904591
```

[ ]:
```

```