Ex. No.: 4 Write a Program for Correlation Coefficient

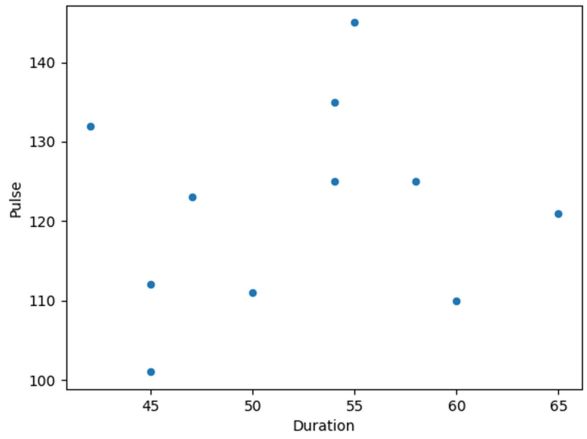
AIM:

To write a Python program to calculate the Correlation coefficient by using the given series and csv files.

(i) Program (By using the csv file):

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("Book.csv", header=0, sep=",")
df.plot(x='Duration',y='Pulse',kind='scatter')
plt.show()
```

Output:

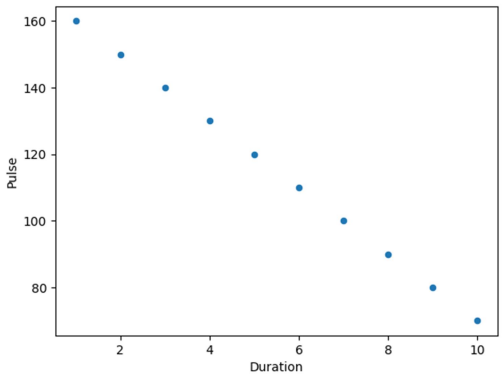


The Correlation coefficient is zero. Hence there is no linear relationship between the two variables.

(ii) Program (By using the given series):

```
import pandas as pd
import matplotlib.pyplot as plt
Correlation={'Duration':[10,9,8,7,6,5,4,3,2,1],'Pulse':[70,80,90,100,110,120,130
,140,150,160]}
Correlation=pd.DataFrame(data=Correlation)
Correlation.plot(x='Duration',y='Pulse',kind='scatter')
plt.show()
```

Output:



The Correlation coefficient is -1. Hence, if the duration is longer hours, we tend to have lower pulse.

(ii) Program (By using the csv file):

```
import pandas as pd
df=pd.read_csv("Book.csv", header=0, sep=",")
Correlation=round(df.corr(),2)
print(Correlation)
```

Output:

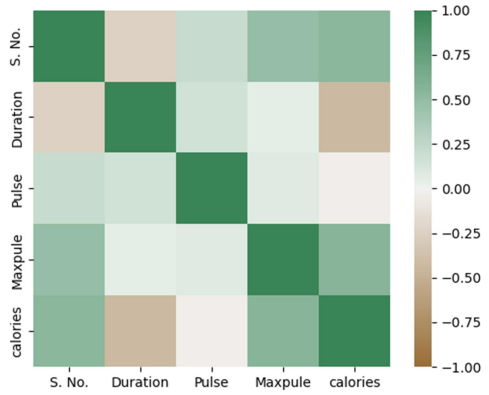
	S. No.	Duration	Pulse	Maxpule	calories
S. No.	1.00	-0.25	0.22	0.49	0.54
Duration	-0.25	1.00	0.16	0.05	-0.44
Pulse	0.22	0.16	1.00	0.08	-0.03
Maxpule	0.49	0.05	0.08	1.00	0.55
calories	0.54	-0.44	-0.03	0.55	1.00

(iii) Program (By using the csv file and heatmap):[We can use the Seaborn library to create a correlation heat map. Seaborn is a visualization library based on matplotlib]

```
import pandas as pd
import matplotlib.pyplot as plt
importseaborn as sns
df=pd.read_csv("Book.csv", header=0, sep=",")
correlation=df.corr()
axis_corr=sns.heatmap(correlation, vmin=-1, vmax=1, center=0,
cmap=sns.diverging_palette(50,500,n=500), square=True)
plt.show()
```

Import the library seaborn as sns. Use sns.heatmap() to tell Python that we want a heatmap to visualize the correlation matrix. Use the correlation matrix. Define the maximal and minimal values of the heatmap. Define that 0 is the center. Define the colors with sns.diverging_palette. n=500 means that we want 500 types of color in the same color palette. square = True means that we want to see squares.

Output:

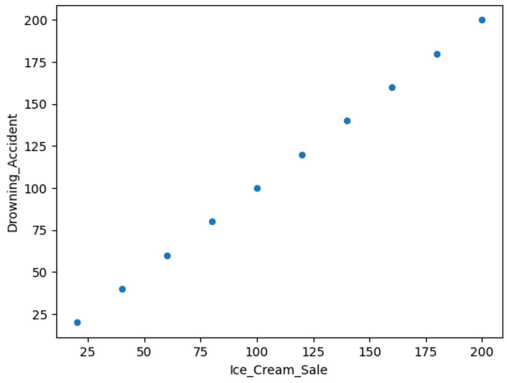


The closer the correlation coefficient is to 1, the greener the squares get.
The closer the correlation coefficient is to -1, the browner the squares get.

(iv) Program (By using the given series):

```
import pandas as pd
import matplotlib.pyplot as plt
Beach = {"Drowning_Accident": [20,40,60,80,100,120,140,160,180,200],
"Ice_Cream_Sale": [20,40,60,80,100,120,140,160,180,200]}
Drowning = pd.DataFrame(data=Beach)
Drowning.plot(x="Ice_Cream_Sale", y="Drowning_Accident", kind="scatter")
plt.show()
```

Output:



The number varies from -1 to 1.1 means that there is a 1 to 1 relationship (a perfect correlation), and for this data set, each time a value went up in the first column, the other one went up as well.0.9 is also a good relationship, and if you increase one value, the other will probably increase as well.-0.9 would be just as good relationship as 0.9, but if you increase one value, the other will probably go down.0.2 means NOT a good relationship, meaning that if one value goes up does not mean that the other will.

Correlation measures the numerical relationship between two variables. A high correlation coefficient (close to 1), does not mean that we can for sure conclude an actual relationship between two variables.
In other words: can we use ice cream sale to predict drowning accidents? The answer is - Probably not. It is likely that these two variables are accidentally correlating with each other.
What causes drowning? Unskilled swimmers, Waves, Cramp, Seizure disorders, Lack of supervision, Alcohol (mis)use, etc.
There is an important difference between correlation and causality:

- Correlation is a number that measures how closely the data are related
- Causality is the conclusion that x causes y.

Tip: Always critically reflect over the concept of causality when doing predictions!

```
import pandas as pd
importmatplotlib.pyplot as plt
df = pd.read_csv('Book.csv')
df.plot()
plt.show()
```