Correlation

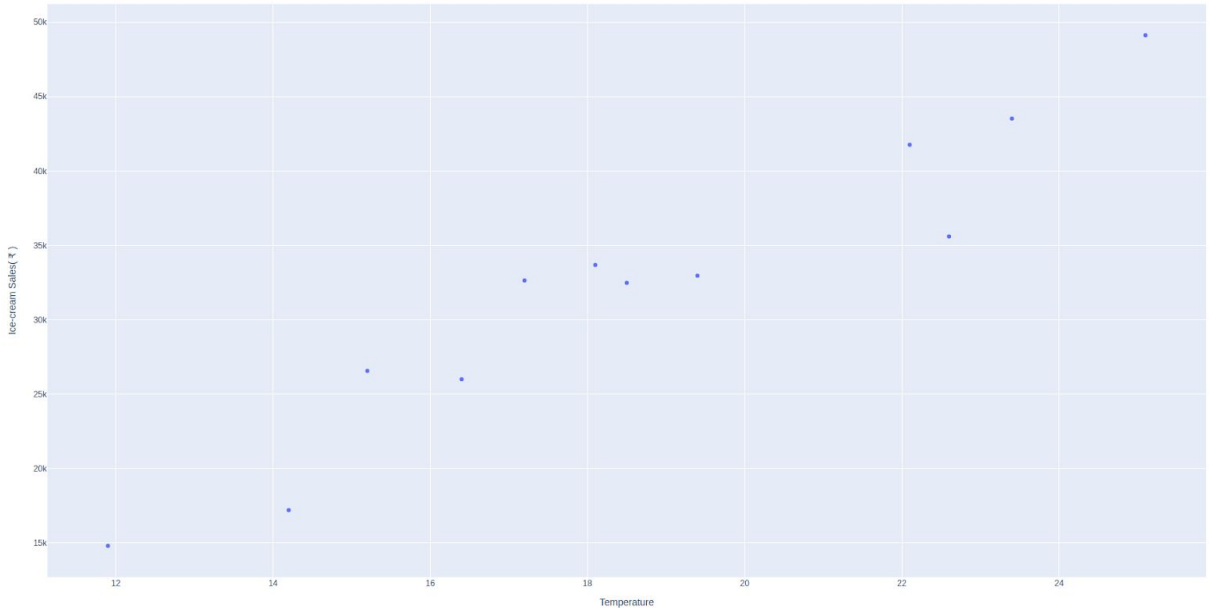
What we did:
In last class we learned about the standard deviation.

In this class we learned about the correlation and methods to find it.

How we did it:

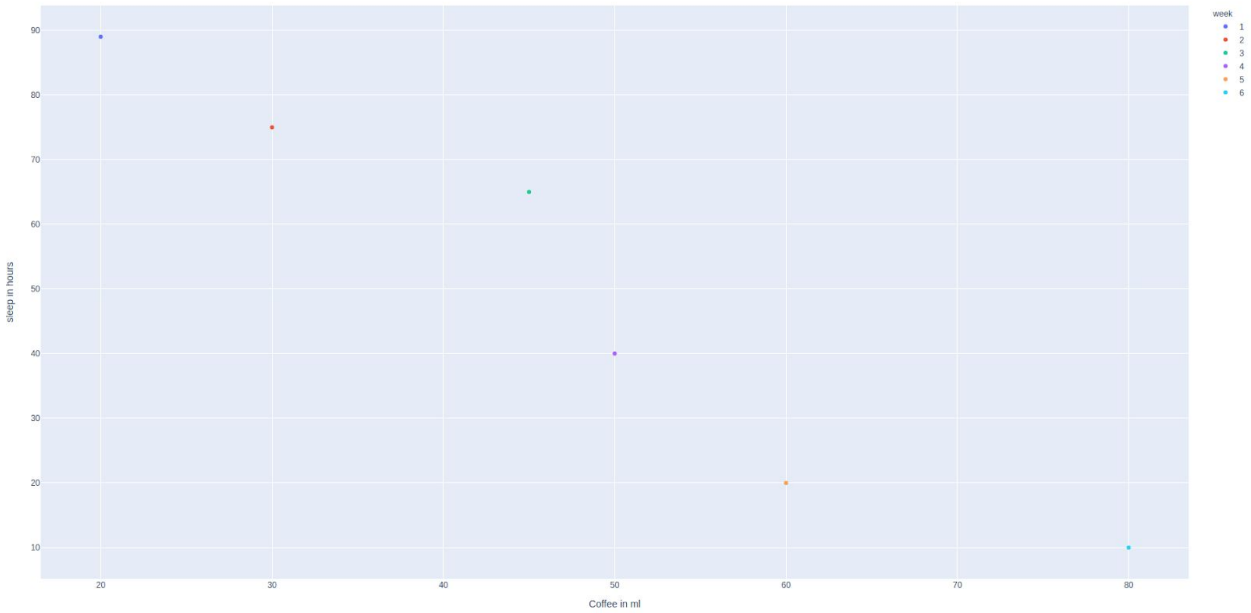
We saw how data is correlated through an example of temperature vs ice-cream sale.
We saw that when the temperature goes up the ice-cream sales go up too.
This type of data is called positive correlated data.

```
import plotly.express as px
import csv

with open("./data/Ice-Cream vs Cold-Drink vs Temperature - Ice Cream Sale vs Temperature data.csv") as csv_file:
      df = csv.DictReader(csv_file)
      fig = px.scatter(df,x="Temperature", y="Ice-cream Sales( ₹ )")
      fig.show()
```



Then we saw the inversely correlated data.
In this data we saw that as the coffee consumption increases the hours of sleep a person gets decreases.

```
import plotly.express as px
import csv

with open("./data/cups of coffee vs hours of sleep.csv") as csv_file:
      df = csv.DictReader(csv_file)
      fig = px.scatter(df,x="Coffee", y="sleep")
      fig.show()
```



We learned that correlation can be calculated as well.
A correlation of 1 means the two data sets are closely correlated. This will be a rising graph where the data points are close to a central line.
A correlation of -1 means that the two data sets are inversely correlated. This will be a falling graph where the data points are close to a central line.
A correlation of 0 means that the two data sets are not correlated at all! The data points will be scattered on the graph.

Correlation always lies in between -1 and 1.

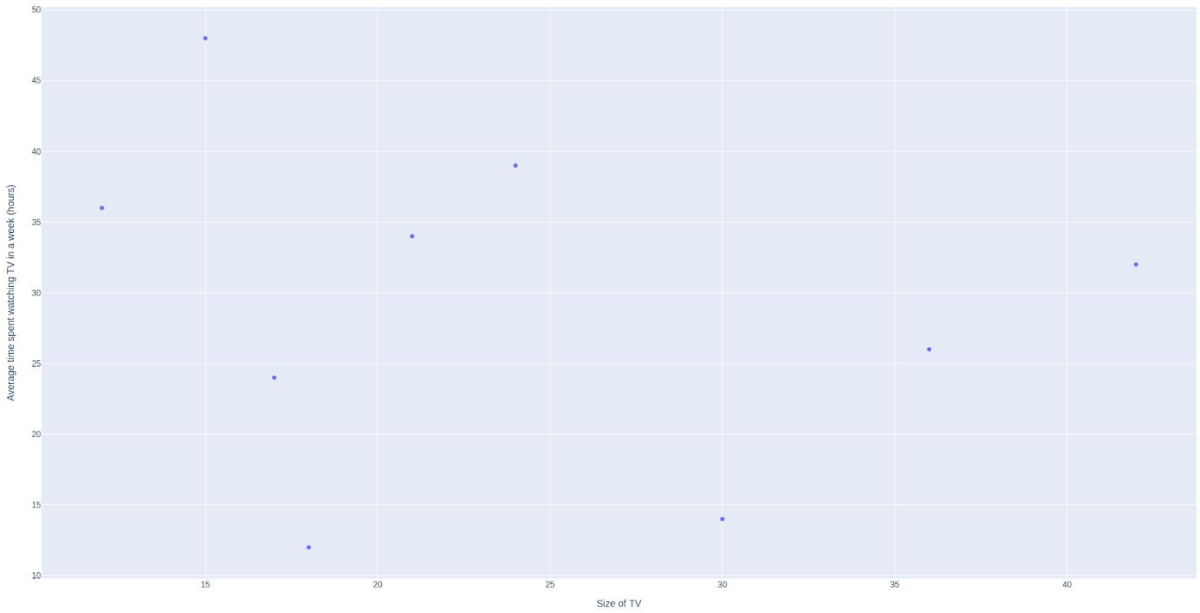
We wrote code to find the correlation between the temperature and ice-cream sales.

```python
import plotly.express as px
import csv
import numpy as np


def getDataSource(data_path):
    ice_cream_sales = []
    cold_drink_sales = []
    with open(data_path) as csv_file:
        csv_reader = csv.DictReader(csv_file)
        for row in csv_reader:
            ice_cream_sales.append(float(row["Temperature"]))
            cold_drink_sales.append(float(row["Ice-cream Sales( ₹ )"]))

    return {"x" : ice_cream_sales, "y": cold_drink_sales}

def findCorrelation(datasource):
    correlation = np.corrcoef(datasource["x"], datasource["y"])
    print("Correlation between Temperature vs Ice Cream Sales :-  \n--->",correlation[0,1])

def setup():
    data_path  = "./data/Ice-Cream vs Cold-Drink vs Temperature - Ice Cream Sale vs Temperature data.csv"

    datasource = getDataSource(data_path)
    findCorrelation(datasource)

setup()
```

```
$ python3 setup.py
Correlation between Temperature vs Ice Cream Sales :-
---> 0.9575066230015955
```

We then plotted a scatter plot for TV watched in a week on average vs the size of television.

```python
import plotly.express as px
import csv

with open("./data/Size of TV,   Average time spent watching TV in a week (hours).csv") as csv_file:
      df = csv.DictReader(csv_file)
      fig = px.scatter(df,x="Size of TV", y="\tAverage time spent watching TV in a week (hours)")
      fig.show()
```



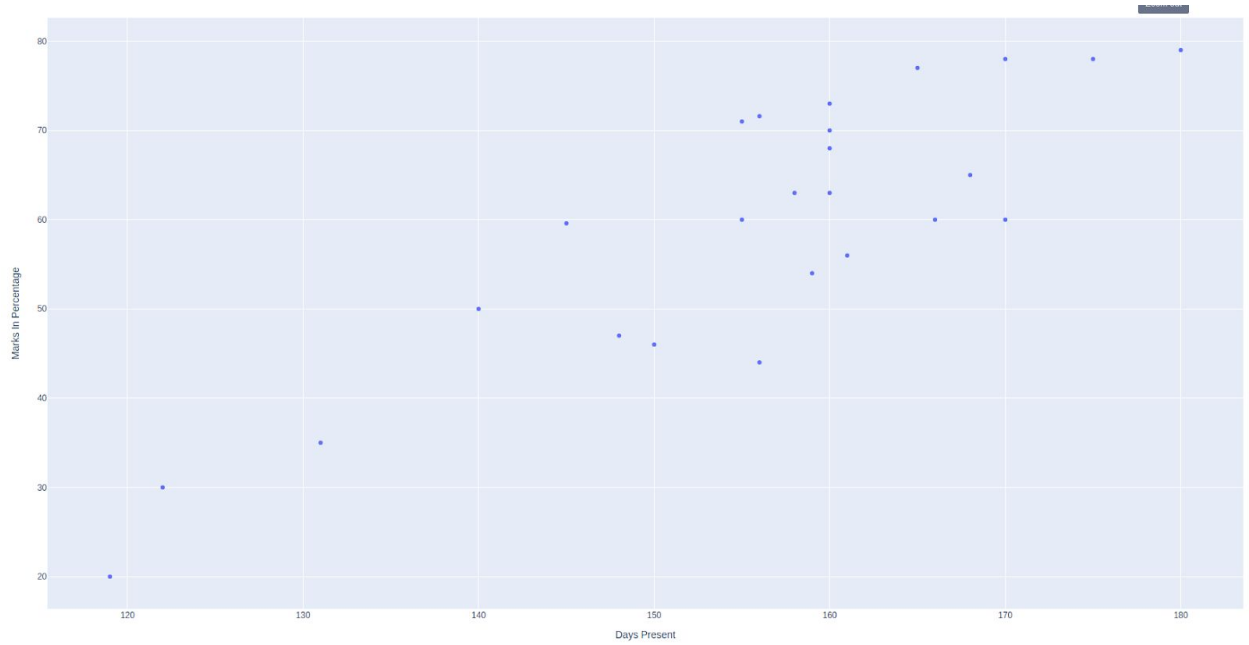
We saw that the points are scattered and dataset is not correlated at all.
We wrote code to calculate the correlation between TV watched in a week on average vs the size of television.

```
import csv
import numpy as np


def getDataSource(data_path):
    size_of_tv = []
    Average_time_spent = []
    with open(data_path) as csv_file:
        csv_reader = csv.DictReader(csv_file)
        for row in csv_reader:
            size_of_tv.append(float(row["Size of TV"]))
            Average_time_spent.append(float(row["\tAverage time spent watching TV in a week (hours)"]))

    return {"x" : size_of_tv, "y": Average_time_spent}

def findCorrelation(datasource):
    correlation = np.corrcoef(datasource["x"], datasource["y"])
    print("Correlation between Size of Tv and Average time spent watching Tv in a week :-  \n--->",correlation[0,1])

def setup():
    data_path  = "./data/Size of TV,    Average time spent watching TV in a week (hours).csv"

    datasource = getDataSource(data_path)
    findCorrelation(datasource)

setup()
```

```
Correlation between Size of Tv and Average time spent watching Tv in a week :-
---> -0.21596489617950243
```

We saw another dataset ,**number of days students attended college vs the marks they scored in their exams.**

```
import plotly.express as px
import csv

with open("./data/Student Marks vs Days Present.csv") as csv_file:
      df = csv.DictReader(csv_file)
      fig = px.scatter(df,x="Days Present", y="Marks In Percentage")
      fig.show()
```



Here we saw that the data points are close to each other and the two data are positively correlated.

Then we wrote code to calculate the correlation .

```python
import csv
import numpy as np


def getDataSource(data_path):
    marks_in_percentage = []
    days_present = []
    with open(data_path) as csv_file:
        csv_reader = csv.DictReader(csv_file)
        for row in csv_reader:
            marks_in_percentage.append(float(row["Marks In Percentage"]))
            days_present.append(float(row["Days Present"]))

    return {"x" : marks_in_percentage, "y": days_present}

def findCorrelation(datasource):
    correlation = np.corrcoef(datasource["x"], datasource["y"])
    print("Correlation between Marks in percentage and Days present :-  \n--->",correlation[0,1])

def setup():
    data_path  = "./data/Student Marks vs Days Present.csv"

    datasource = getDataSource(data_path)
    findCorrelation(datasource)

setup()
```

```
Correlation between Marks in percentage and Days present :-
---> 0.86288947614385
```

What's next?

In the next class, we will learn to find correlation.