Screen Time Analysis Project

Screen Time Analysis is the task of analyzing and creating a report on which applications and websites are used by the user for how much time.

For the task of screen time analysis, dataset that contains:

1. Date
2. Usage of Applications
3. Number of Notifications from Applications
4. Number of times apps opened

Import neccesary modules:

[1]:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
```

pandas (pd):

Pandas is a powerful library for data manipulation and analysis in Python. It provides data structures such as DataFrame and Series, which allow for easy handling and manipulation of structured data.

numpy (np):

NumPy is a fundamental package for scientific computing with Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

plotly.express (px):

Plotly Express is a high-level interface for creating expressive and interactive plots using Plotly. It offers a simple syntax for generating a wide range of plot types, including scatter plots, line plots, bar charts, and more, with minimal code.

plotly.graph_objects (go):

Plotly Graph Objects provides a low-level interface for creating customizable and interactive plots using Plotly. It allows for more fine-grained control over plot elements, making it suitable for creating complex and customized visualizations.

Load the data:

[3]:

```
data = pd.read_csv("C:/Users/hp/OneDrive/Desktop/Projects/Project 1/Screentime-App-Deta
```

[4]:

```
print(data.head(5)) # display the top 5 rows of the database
```

```
         Date  Usage  Notifications  Times opened        App
0  08/26/2022     38             70            49  Instagram
1  08/27/2022     39             43            48  Instagram
2  08/28/2022     64            231            55  Instagram
3  08/29/2022     14             35            23  Instagram
4  08/30/2022      3             19             5  Instagram
```

Data cleaning:

[5]:

```
data.isnull() # check if there are null values in the data set
```

[5]:

	Date	Usage	Notifications	Times opened	App
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	False	False	False
11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	False	False
15	False	False	False	False	False

Continue…

[6]:

```
data.isnull().sum() # sum of the null values in all the columns
```

[6]:

```
Date             0
Usage            0
Notifications    0
Times opened     0
App              0
dtype: int64
```

Statistical Analysis:

[7]:

```
print(data.descriabe())
```

```
             Usage  Notifications  Times opened
count    54.000000      54.000000     54.000000
mean     65.037037     117.703704     61.481481
std      58.317272      97.017530     43.836635
min       1.000000       8.000000      2.000000
25%      17.500000      25.750000     23.500000
50%      58.500000      99.000000     62.500000
75%      90.500000     188.250000     90.000000
max     244.000000     405.000000    192.000000
```

Data Visualization:

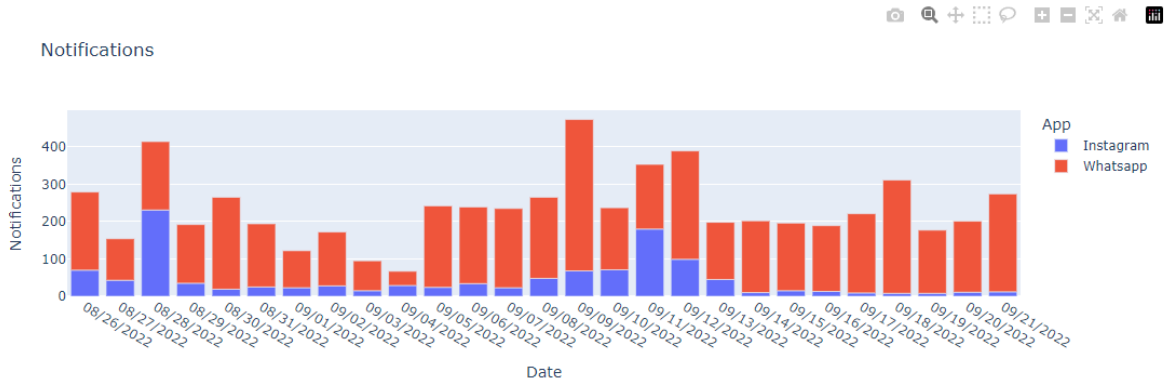
Usage of Applications

[16]:

```
figure = px.bar(data_frame=data,
                x= "Date",
                y= "Usage",
                color="App",
                title="Usage of Screen")
```

The code snippet utilizes Plotly Express to create a bar chart visualizing the usage of different applications on the screen over time.

```
[11]: figure.show()
```

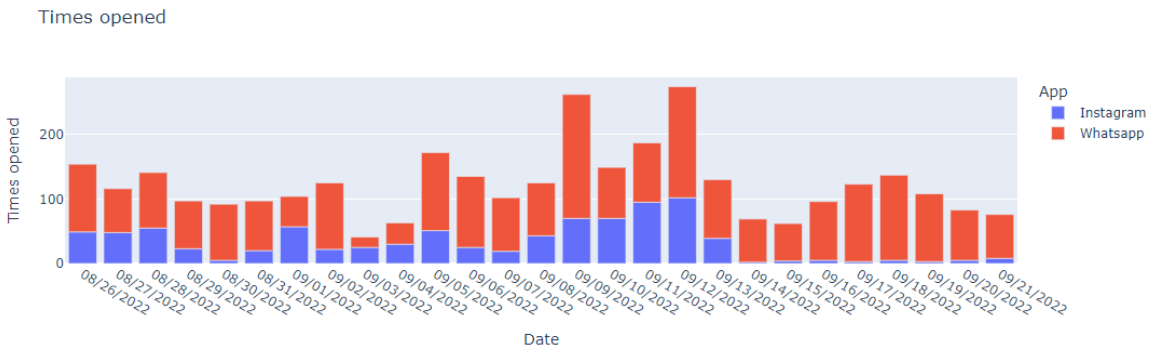


How many time the application opened:

```
[12]:
figure = px.bar(data_frame=data,
                x= "Date",
                y= "Times opened",
                color="App",
                title="Times opened")
```

The code creates a bar chart with Plotly Express, illustrating the frequency of opening different applications over time.

```
[13]: figure.show()
```

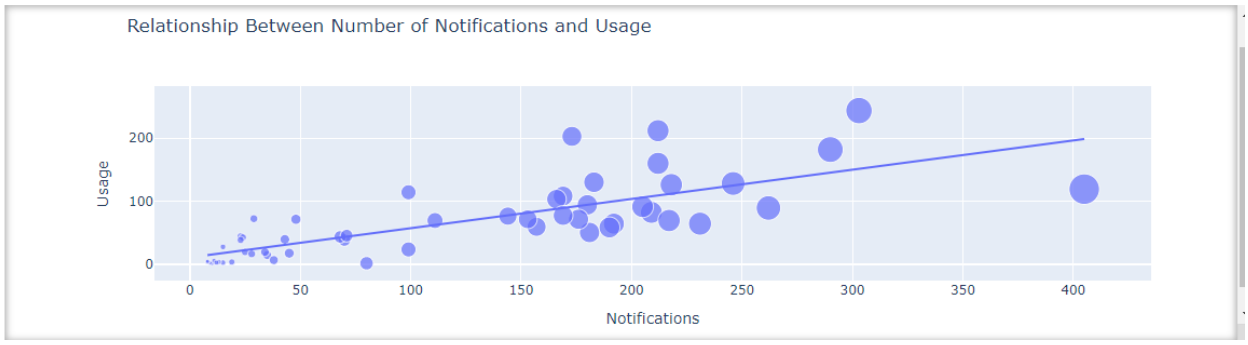


Relationship Between Number of Notifications and Usage

```
[14]:
figure = px.scatter(data_frame = data,
                    x="Notifications",
                    y="Usage",
                    size="Notifications",
                    trendline="ols",
                    title = "Relationship Between Number of Notifications and Usage")
```

The code generates a scatter plot using Plotly Express to visualize the relationship between the number of notifications and usage of applications, with a trendline fitted using ordinary least squares regression.

```
[15]: figure.show()
```



This Python project focuses on analyzing screen time data to gain insights into the usage patterns of different applications and websites by users. Screen Time Analysis involves examining usage metrics such as the duration of app usage, the number of notifications received, and the frequency of app openings. By leveraging Python libraries like pandas for data manipulation and Plotly for visualization, this project aims to provide a comprehensive report on screen time behavior.