

Screen time Analysis

Ever wondered where your phone time goes? Screen time analysis is like a digital fitness tracker, revealing how much time you spend on apps and websites. By analyzing this data, you can gain valuable insights into your tech habits and make informed choices about your digital life.

For the task of screen time analysis, I found an ideal dataset that contains data about:

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

You can download the dataset from [here](#).

Screen time Analysis with Python:

Let's Analyze Your Screen Time with Python: Gear up for a journey into screen time analysis using the power of Python. We'll import essential libraries and explore the data to gain valuable insights.

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

data = pd.read_csv(r'Screentime.csv')
print(data.head())
```

	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

Now let's have a look if the dataset has any null values or not:

```
data.isnull().sum()
```

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

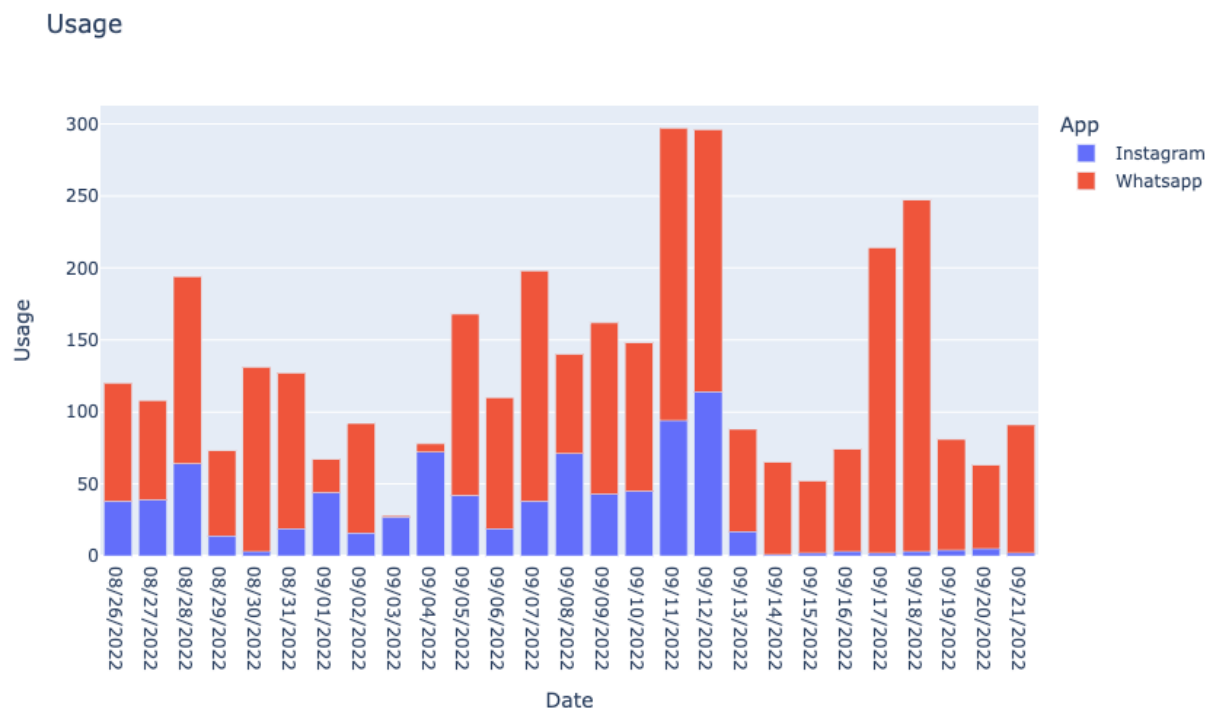
Since there are no missing values, we can proceed with calculating descriptive statistics to understand the data distribution.

```
print(data.describe())
```

	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

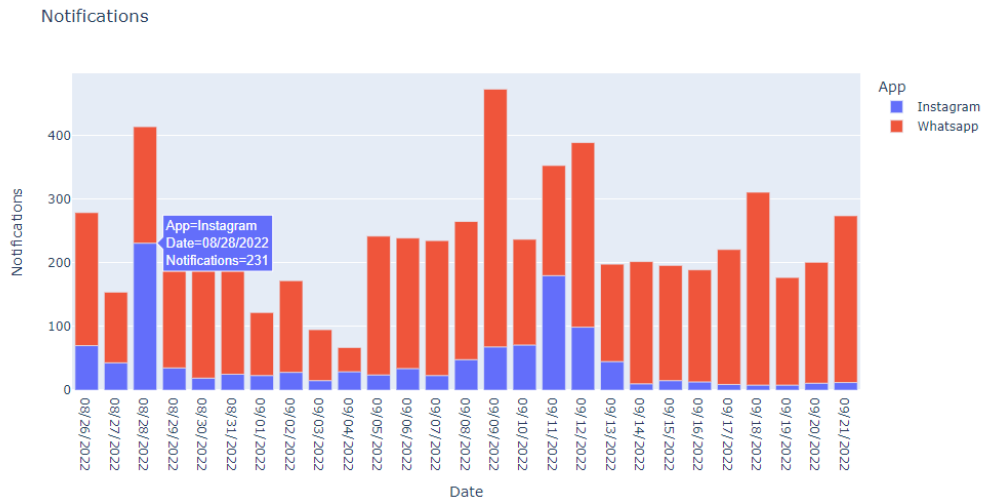
Let's delve into the user's screen time! We'll begin by examining how much time they spend using each app.

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



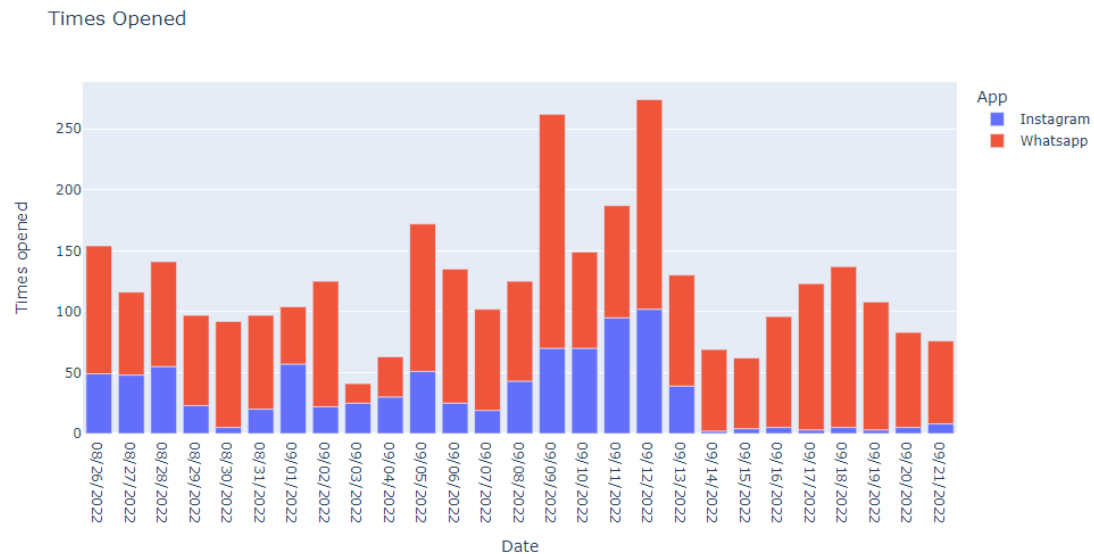
Now let's have a look at the number of notifications from the apps:

```
figure = px.bar(data_frame=data,
                x = "Date",
                y = "Notifications",
                color="App",
                title="Notifications")
figure.show()
```



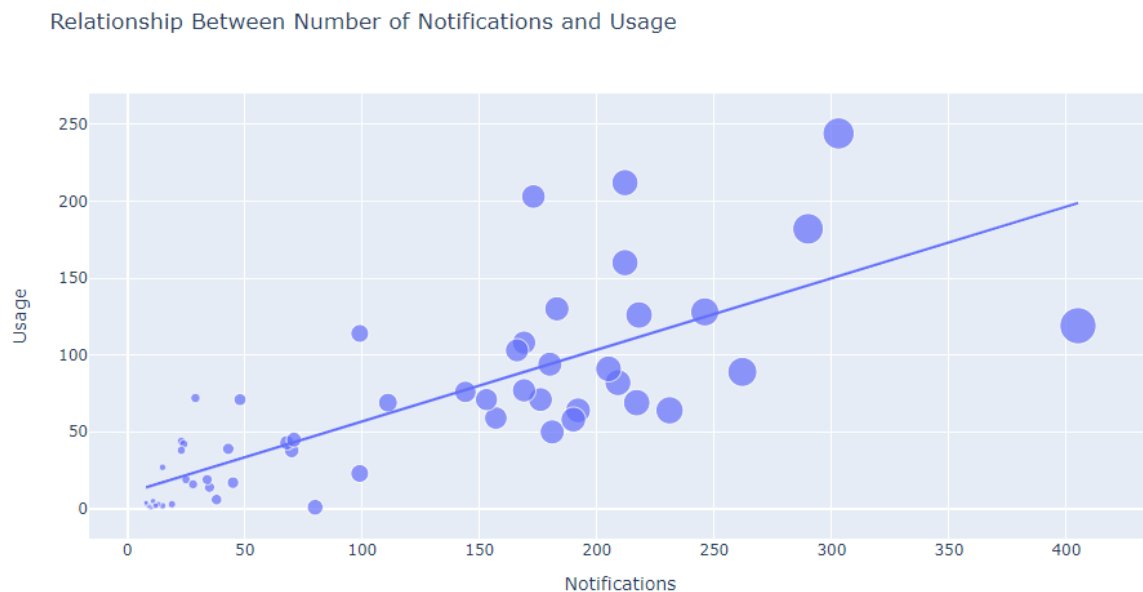
Now let's have a look at the number of times the apps opened:

```
figure = px.bar(data_frame=data,
                x = "Date",
                y = "Times opened",
                color="App",
                title="Times Opened")
figure.show()
```



Intriguingly, notifications often trigger smartphone usage. Let's explore the potential correlation between the number of notifications received and the amount of app usage.

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



There's a linear relationship between the number of notifications and the amount of usage. It means that more notifications result in more use of smartphones.

In Conclusion:

This Python guide has equipped you with the tools to analyze your own screen time! We've explored techniques to uncover app usage patterns, notification influences, and app opening frequency.