

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("C:\\Users\\ishaan\\Desktop\\proj\\python\\screen metrics and monitoring analysis\\Screentime-Details.csv")
data.head(15)

Out[2]:
```

	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
5	08/31/2022	19	25	20	Instagram
6	09/01/2022	44	23	57	Instagram
7	09/02/2022	16	28	22	Instagram
8	09/03/2022	27	15	25	Instagram
9	09/04/2022	72	29	30	Instagram
10	09/05/2022	42	24	51	Instagram
11	09/06/2022	19	34	25	Instagram
12	09/07/2022	38	23	19	Instagram
13	09/08/2022	71	48	43	Instagram
14	09/09/2022	43	68	70	Instagram

```
In [35]: ds = pd.DataFrame(data)
print(ds.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54 entries, 0 to 53
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  --
0   Date             54 non-null    object
1   Usage            54 non-null    int64
2   Notifications     54 non-null    int64
3   Times opened     54 non-null    int64
4   App              54 non-null    object
dtypes: int64(3), object(2)
memory usage: 2.2+ KB
None

In [12]: #to check if there is any null values in any column of the dataset.

data.isnull().sum()

Out[12]:
Date            0
Usage           0
Notifications    0
Times opened    0
App             0
dtype: int64

In [5]: # to see descriptive stats of our data

data.describe()

Out[5]:
```

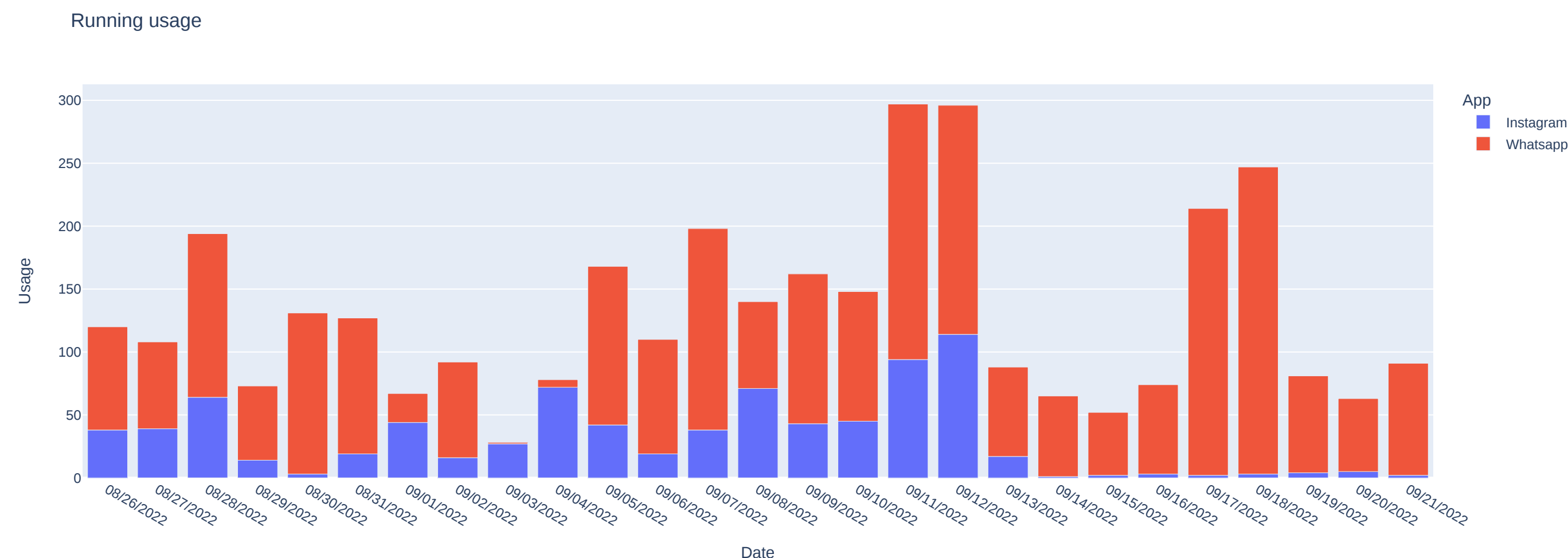
	Usage	Notifications	Times opened
count	54.000000	54.000000	54.000000
mean	65.037037	117.703704	61.481481
std	58.317272	97.017530	43.836635
min	1.000000	8.000000	2.000000
25%	17.500000	25.750000	23.500000
50%	58.500000	99.000000	62.500000
75%	90.500000	188.250000	90.000000
max	244.000000	405.000000	192.000000

Now we start analyzing and plotting screen usage of users for different applications for various dates

```
In [3]: figure = px.bar(data_frame = data,
                      y= "Usage",
                      x= "Date",
                      color= "App",
                      title = "Running usage")

figure.show()

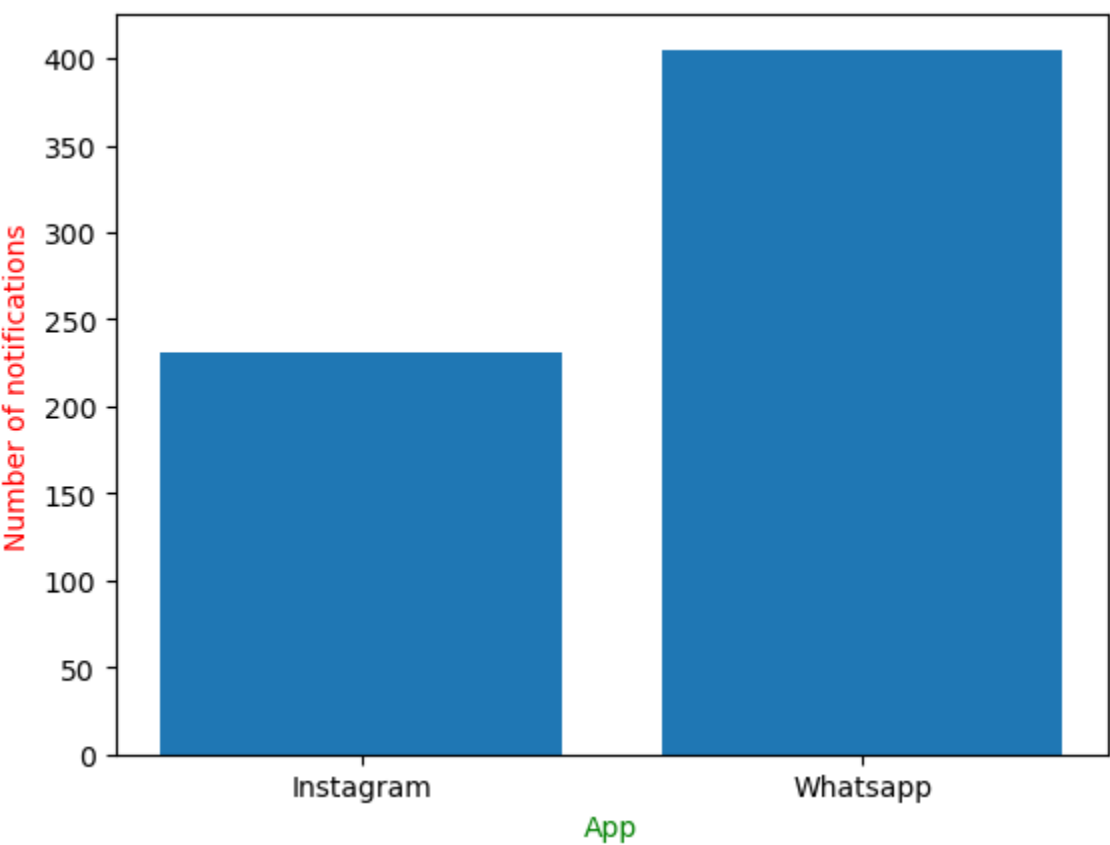
#observation : Whatsapp is more likely used by the user based on screen time usage.
```



```
In [7]: x = data['App']
y = data['Notifications']
plt.bar(x,y)
plt.xlabel("App",c="g")
plt.ylabel("Number of notifications",c='r')

#observation : Whatsapp again being the most used app based on number of notifications.

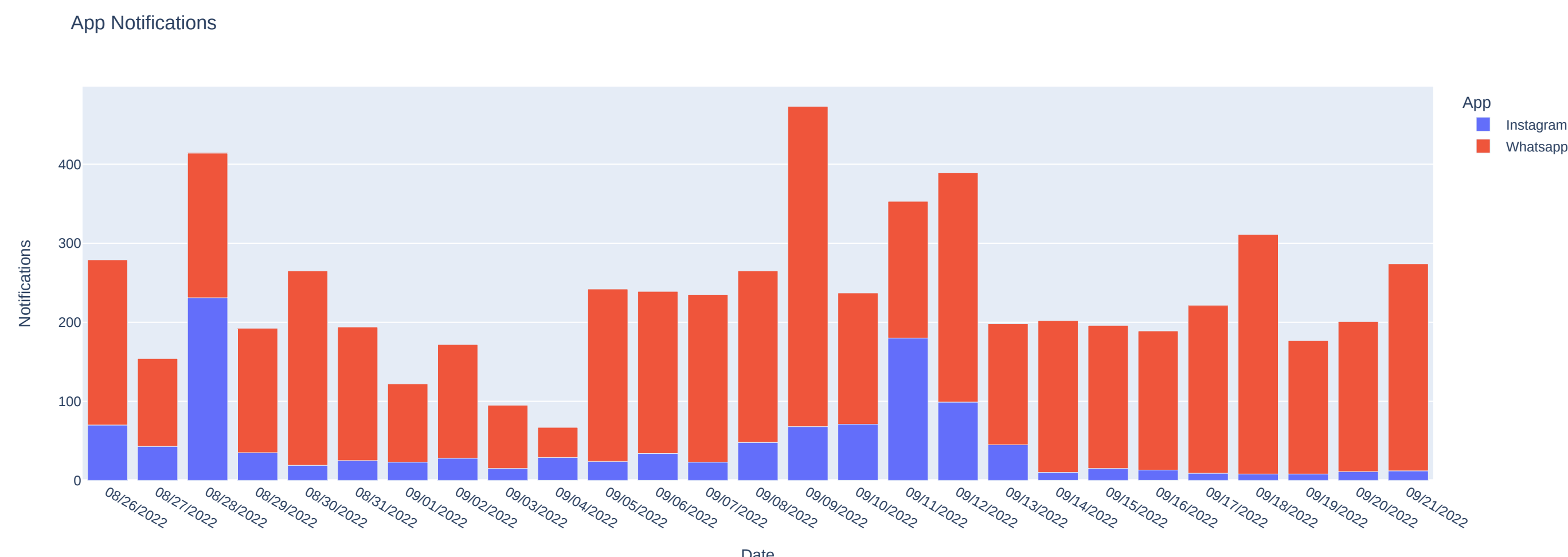
Out[7]: Text(0, 0.5, 'Number of notifications')
```



Number of notifications from the apps for various dates

```
In [4]: figure = px.bar(data_frame=data,
                      x = "Date",
                      y = "Notifications",
                      color="App",
                      title="App Notifications")

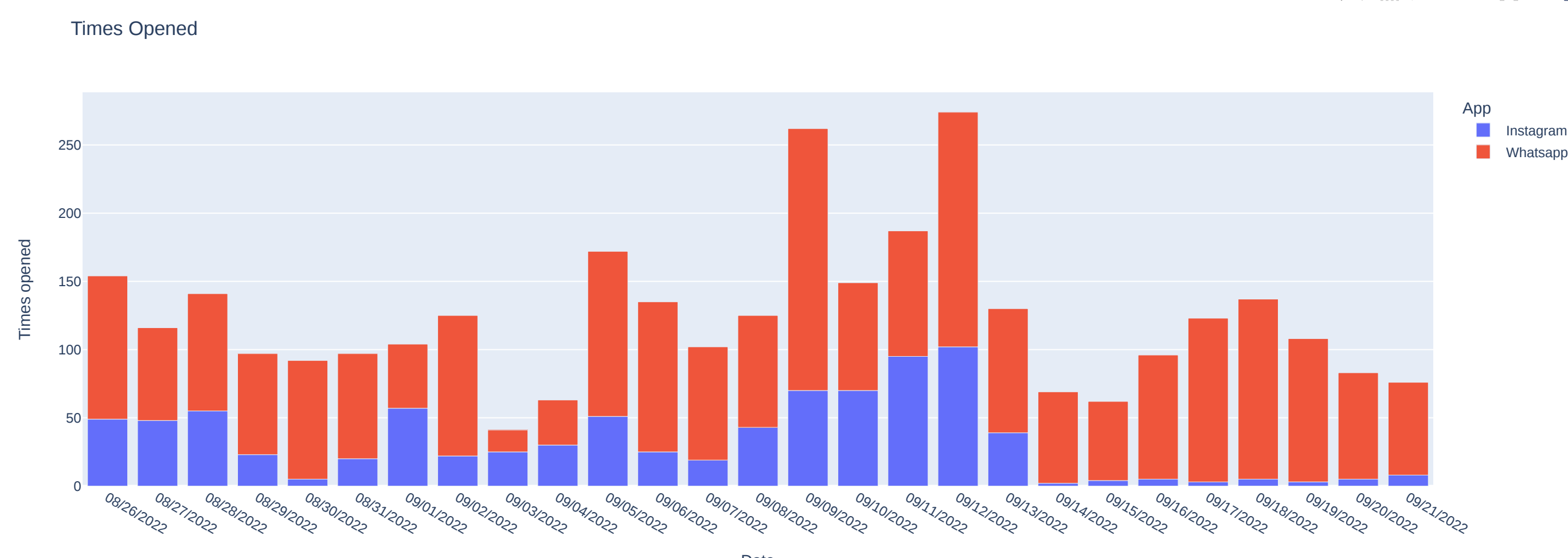
figure.show()
```



Number of times the apps opened

```
In [5]: figure = px.bar(data_frame=data,
                      x = "Date",
                      y = "Times opened",
                      color="App",
                      title="Times Opened")

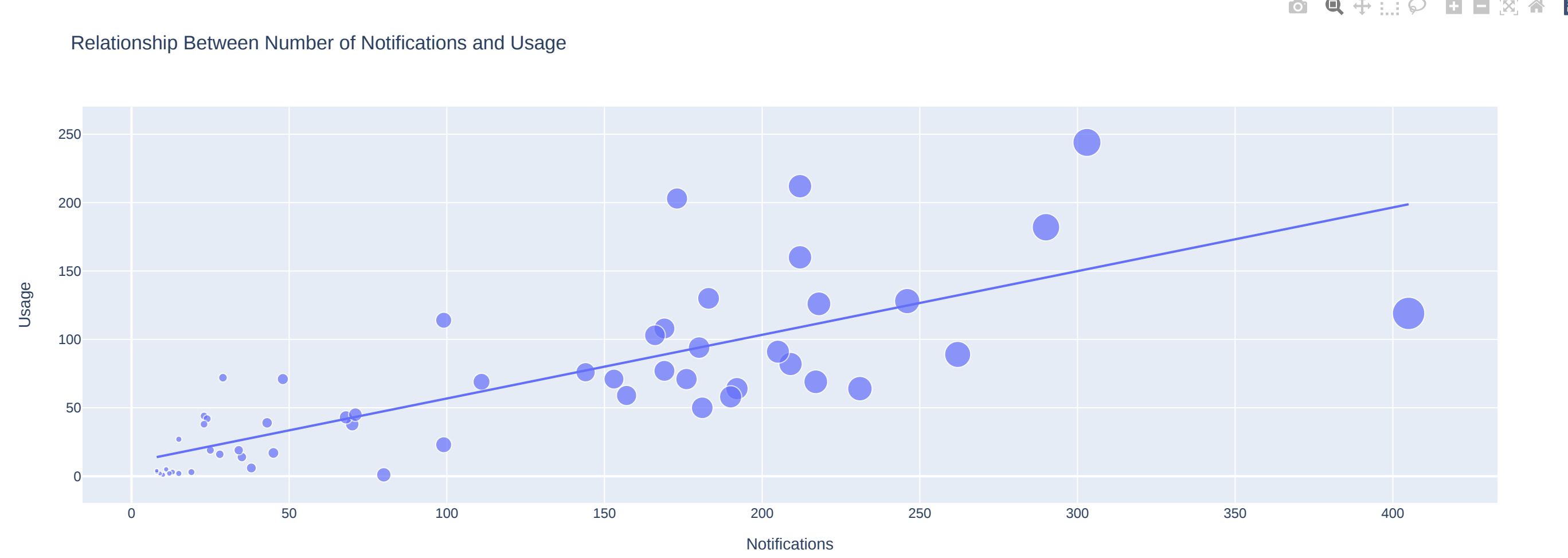
figure.show()
```



The usage of smartphones is often triggered by app notifications. Let's examine the correlation between notification frequency and usage amount.

```
In [6]: figure = px.scatter(data_frame = data,
                          x="Notifications",
                          y="Usage",
                          size="Notifications",
                          trendline="ols",
                          title = "Relationship Between Number of Notifications and Usage")

figure.show()
```



Observation from above scatter plot :
The correlation between the frequency of notifications and smartphone utilization is direct. This implies that a higher number of notifications leads to an increase in smartphone usage. Hence, showing a linear relationship.

Summary

By utilizing the Python programming language and it's libraries like numpy,pandas,matplotlib and plotly making it more interactive and user friendly , we can perform a Screen Time Analysis to determine the usage patterns of a user. This analysis involves generating a report on the applications and websites accessed by the user and the amount of time spent on each. I hope this Screen Time Analysis using Python was informative and insightful.