

**B. V. Patel Institute of Computer Science**

**Subject: Data Analysis Using Python**

**Mini Project**

**Title :**

**phone usage data with data analysis**

**Submitted by:**

**Feny Doshi : 202307100510166**

**Soniya Valeja : 202307100510194**

**Jasvi Patel : 202307100510057**

**Guided By:**

**Ms. Urvi Panchal**

# 1. Introduction:

The file `phone_usage_india.csv` contains a dataset that provides detailed information about mobile phone usage patterns among individuals in India. The dataset includes various attributes related to users' demographics, phone specifications, and usage habits. Here is an overview of the key columns in the dataset:

**User ID (U00001, U00002, etc.):** A unique identifier for each user.

**Age:** The age of the user.

**Gender:** The gender of the user (Male, Female, Other).

**Location:** The city or region where the user resides (e.g., Mumbai, Delhi, Bangalore).

**Phone Brand:** The brand of the phone used by the user (e.g., Vivo, Samsung, Apple).

**OS:** The operating system of the phone (e.g., Android, iOS).

**Screen Time (hrs/day):** The average number of hours the user spends on their phone per day.

**Data Usage (GB/month):** The average amount of mobile data consumed by the user per month.

**Calls Duration (mins/day):** The average duration of phone calls made by the user per day.

**Number of Apps Installed:** The total number of apps installed on the user's phone.

**Social Media Time (hrs/day):** The average time spent on social media apps per day.

**E-commerce Spend (INR/month):** The average monthly expenditure on e-commerce platforms.

**Streaming Time (hrs/day):** The average time spent on streaming services (e.g., video, music) per day.

**Gaming Time (hrs/day):** The average time spent on mobile gaming per day.

**Monthly Recharge Cost (INR):** The average monthly cost of mobile recharge.

**Primary Use:** The primary purpose of phone usage (e.g., Education, Entertainment, Social Media, Work).

This dataset can be used to analyze various aspects of mobile phone usage, such as the relationship between demographics and phone usage patterns, the popularity of different phone brands and operating systems, and the

impact of mobile usage on different activities like social media, gaming, and e-commerce.

- Loading the CSV file

```
import pandas as pd
df=pd.read_csv("phone_usage_india.csv")
print(df)
```

Output :

```
8 rows x 10 columns]
PS C:\Users\doshi\Downloads\feny\feny\fff> & C:/Users/doshi/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/doshi/Downloads/feny/feny/fff/1.py
   U00001  Age  Gender  Location  ...  Streaming Time (hrs/day)  Gaming Time (hrs/day)  Monthly Recharge Cost (INR)  Primary Use
0   U00002  53   Male   Mumbai   ...                5.2                4.1                803             Education
1   U00003  60   Other   Delhi    ...                5.1                0.4                1526            Gaming
2   U00004  37  Female  Ahmedabad ...                1.7                2.9                1619            Entertainment
3   U00005  32   Male    Pune    ...                3.2                0.3                1560            Entertainment
4   U00006  16   Male   Mumbai   ...                3.4                2.3                742             Social Media
...      ...    ...    ...      ...                ...                ...                ...             ...
1494  U01496  32   Male   Delhi    ...                5.9                3.1                1617            Gaming
1495  U01497  59   Other  Bangalore ...                7.1                0.8                192             Education
1496  U01498  33  Female  Kolkata   ...                7.9                4.3                1546            Social Media
1497  U01499  55  Female   Mumbai   ...                0.6                4.5                680            Entertainment
1498  U01500  51   Male   Kolkata   ...                7.9                0.5                1894             Work
```

- First ten lines using head

```
# First ten lines
head = df.head(10)
print("First 10 lines : ")
print(head)
```

Output :

```
1498  U01500  51  Male  Kolkata  ...  7.9  0.5  1894  Work

[1499 rows x 16 columns]
First 10 lines :
  U00001  Age  Gender  Location  ...  Streaming Time (hrs/day)  Gaming Time (hrs/day)  Monthly Recharge Cost (INR)  Primary Use
0  U00002  53  Male  Mumbai  ...  5.2  4.1  803  Education
1  U00003  60  Other  Delhi  ...  5.1  0.4  1526  Gaming
2  U00004  37  Female  Ahmedabad  ...  1.7  2.9  1619  Entertainment
3  U00005  32  Male  Pune  ...  3.2  0.3  1560  Entertainment
4  U00006  16  Male  Mumbai  ...  3.4  2.3  742  Social Media
5  U00007  21  Male  Jaipur  ...  0.6  4.8  1749  Entertainment
6  U00008  57  Female  Lucknow  ...  2.9  2.3  1073  Social Media
7  U00009  56  Other  Kolkata  ...  5.2  5.0  1136  Entertainment
8  U00010  46  Female  Kolkata  ...  6.1  2.8  1253  Entertainment
9  U00011  44  Other  Kolkata  ...  7.6  0.4  168  Education

Ln 5, Col 1 (79 selected)  Spaces: 4  UTF-8  CRLF  {} Python  3.11.0  ENG
```

- Last ten lines using tail

```
# Last ten lines
tail = df.tail(10)
print("Last ten lines : ")
print(tail)
```

Output :

```
[10 rows x 16 columns]
Last ten lines :
  U00001  Age  Gender  Location  ...  Streaming Time (hrs/day)  Gaming Time (hrs/day)  Monthly Recharge Cost (INR)  Primary Use
1489  U01491  16  Female  Kolkata  ...  3.0  3.2  1759  Social Media
1490  U01492  21  Male  Jaipur  ...  4.3  1.5  579  Education
1491  U01493  58  Male  Delhi  ...  6.1  3.3  943  Education
1492  U01494  59  Female  Hyderabad  ...  7.7  1.1  529  Education
1493  U01495  59  Female  Delhi  ...  3.0  1.7  888  Social Media
1494  U01496  32  Male  Delhi  ...  5.9  3.1  1617  Gaming
1495  U01497  59  Other  Bangalore  ...  7.1  0.8  192  Education
1496  U01498  33  Female  Kolkata  ...  7.9  4.3  1546  Social Media
1497  U01499  55  Female  Mumbai  ...  0.6  4.5  680  Entertainment
1498  U01500  51  Male  Kolkata  ...  7.9  0.5  1894  Work

[10 rows x 16 columns]

Ln 10, Col 1 (77 selected)  Spaces: 4  UTF-8  CRLF  {} Python  3.11.0  ENG
```

- Dataset shape

```
# Shape
shape = df.shape
print("Shaping :")
print(shape)
```

Output :

```
[10 rows x 16 columns]
Shaping :
(1499, 16)
```

- Checking NULL values

```
# Finding if they have any missing value
null = df.isnull().sum()
print("Finding if the dataset has any missing  
value : ")
print(null)
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Finding if the dataset has any missing value :
U00001          0
Age             0
Gender          0
Location        0
Phone Brand     0
OS              0
Screen Time (hrs/day)  0
Data Usage (GB/month)  0
Calls Duration (mins/day)  0
Number of Apps Installed  0
Social Media Time (hrs/day)  0
E-commerce Spend (INR/month)  0
Streaming Time (hrs/day)  0
Gaming Time (hrs/day)  0
Monthly Recharge Cost (INR)  0
```

- Checking if the dataset has duplicated value

```
# Finding the duplicated value
dup = df.duplicated().sum()
print("Finding if the dataset has any
duplicated value : ")
print(dup)
```

Output :

```
21
Finding if the dataset has any duplicated value :
0
```

- Description of the dataset

```
# Description of the dataset
des =df.describe()
print("Description of the dataset : ")
print(des)
```

Output :

```
Description of the dataset :
      Age  Screen Time (hrs/day)  Data Usage (GB/month)  ...  Streaming Time (hrs/day)  Gaming Time (hrs/day)  Monthly Recharge Cost (INR)
count  1499.000000             1499.000000             1499.000000  ...             1499.000000             1499.000000             1499.000000
mean    37.877252              6.665043              25.697932  ...              4.307405              2.508139             1026.304870
std     13.430914              3.180136              14.162906  ...              2.149289              1.444542             541.781953
min     15.000000              1.000000              1.000000  ...              0.500000              0.000000             102.000000
25%     26.000000              3.900000              13.500000  ...              2.500000              1.300000             572.000000
50%     39.000000              6.800000              25.600000  ...              4.300000              2.500000             1007.000000
75%     50.000000              9.400000              38.000000  ...              6.100000              3.800000             1482.000000
max     60.000000             12.000000              50.000000  ...              8.000000              5.000000             2000.000000

[8 rows x 10 columns]
Columns :
```

- Showing all the columns

```
# Columns
col =df.columns
print("Columns : ")
print(col)
```



Output :

```
Columns :
Index(['U00001', 'Age', 'Gender', 'Location', 'Phone Brand', 'OS',
      'Screen Time (hrs/day)', 'Data Usage (GB/month)',
      'Calls Duration (mins/day)', 'Number of Apps Installed',
      'Social Media Time (hrs/day)', 'E-commerce Spend (INR/month)',
      'Streaming Time (hrs/day)', 'Gaming Time (hrs/day)',
      'Monthly Recharge Cost (INR)', 'Primary Use'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
```

- Information about dataset

```
# Dataset's information
info = df.info()
print("Dataset's information : ")
print(info)
```

Output :

```
Data columns (total 16 columns):
#      Column                                     Non-Null Count  Dtype
---  -
0      U00001                                         1499 non-null   object
1      Age                                           1499 non-null   int64
2      Gender                                       1499 non-null   object
3      Location                                    1499 non-null   object
4      Phone Brand                                1499 non-null   object
5      OS                                           1499 non-null   object
6      Screen Time (hrs/day)                       1499 non-null   float64
7      Data Usage (GB/month)                       1499 non-null   float64
8      Calls Duration (mins/day)                   1499 non-null   float64
9      Number of Apps Installed                     1499 non-null   int64
10     Social Media Time (hrs/day)                  1499 non-null   float64
11     E-commerce Spend (INR/month)                  1499 non-null   int64
12     Streaming Time (hrs/day)                      1499 non-null   float64
13     Gaming Time (hrs/day)                        1499 non-null   float64
14     Monthly Recharge Cost (INR)                   1499 non-null   int64
15     Primary Use                                   1499 non-null   object
dtypes: float64(6), int64(4), object(6)
memory usage: 187.5+ KB
Dataset's information :
None
PS C:\Users\doshi\Downloads\feny\feny\fff> 
```

# Graphs

## ➤ Top 10 Most Popular Phone Brands :

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

```
# Load the dataset
```

```
df = pd.read_csv("phone_usage_india.csv")
```

```
# Plot Top 10 Phone Brands
```

```
plt.figure(figsize=(10, 5))
```

```
df["Phone Brand"].value_counts().head(10).plot(kind="bar",  
color="skyblue")
```

```
plt.title(" Top 10 Most Popular Phone Brands")
```

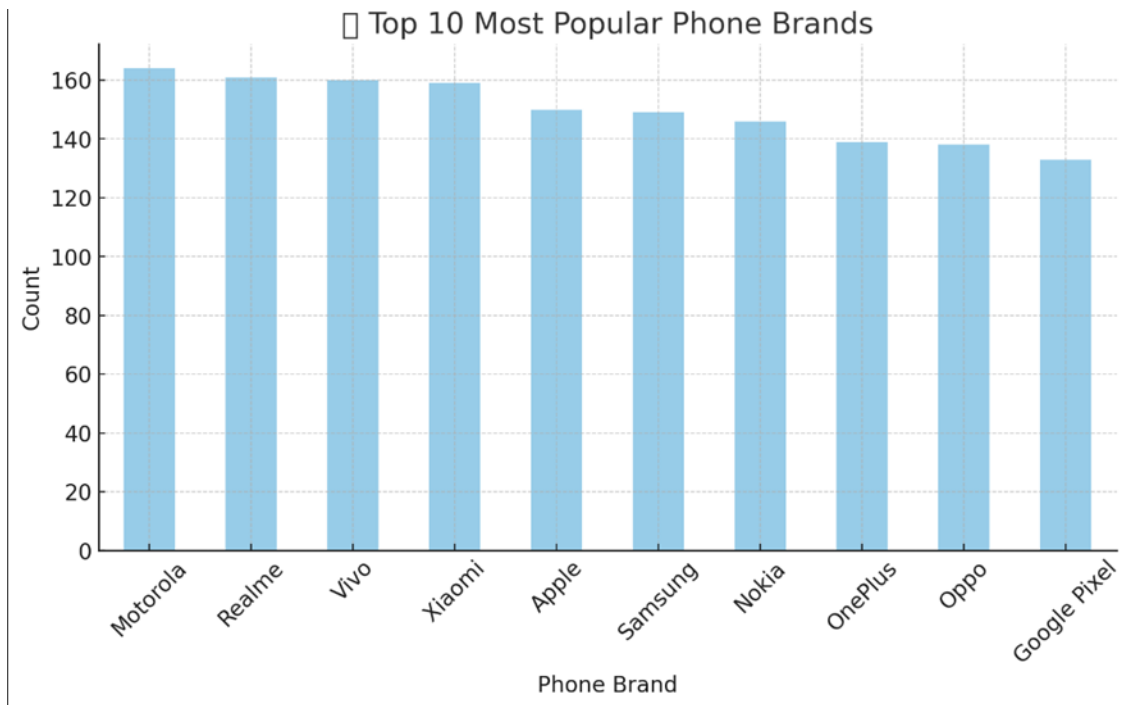
```
plt.xlabel("Phone Brand")
```

```
plt.ylabel("Count")
```

```
plt.xticks(rotation=45)
```

```
plt.show()
```

Output :



➤ **Average Data Usage by Age Group:**

```
import matplotlib.pyplot as plt
```

```
# Plot Data Usage by Age
```

```
plt.figure(figsize=(8, 5))
```

```
df.groupby("Age")["Data Usage  
(GB/month)"].mean().plot(kind="line", marker="o", color="green")
```

```
plt.title("Average Data Usage by Age Group")
```

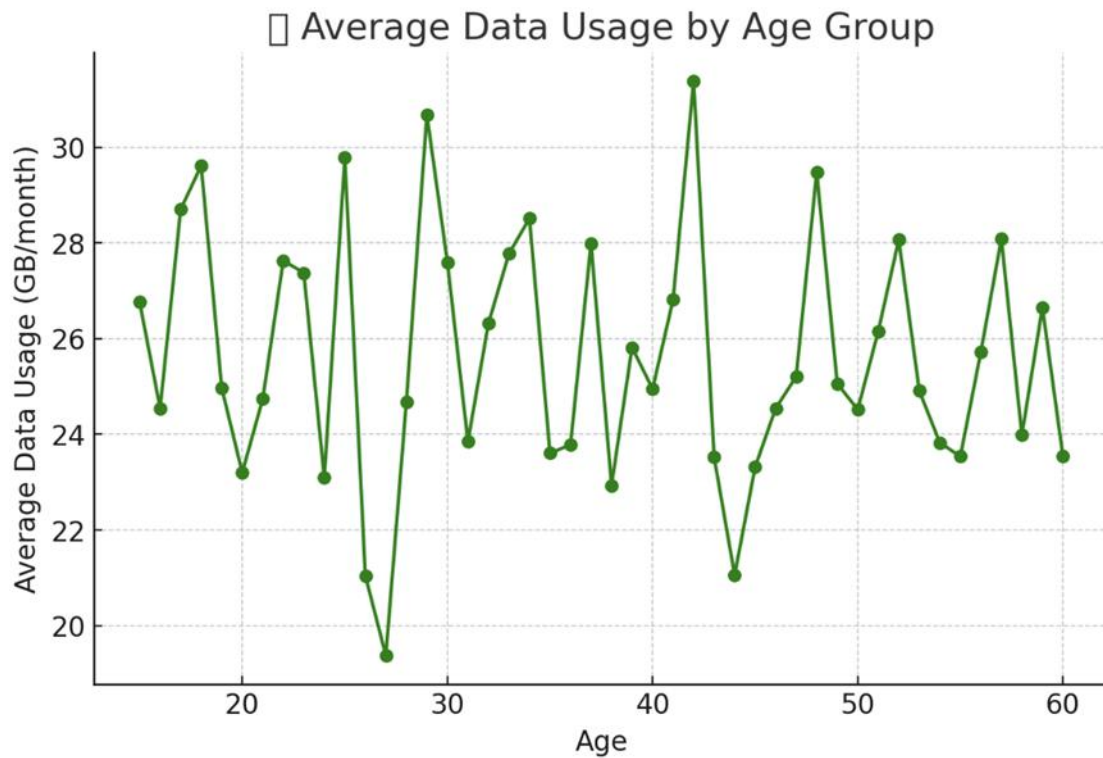
```
plt.xlabel("Age")
```

```
plt.ylabel("Average Data Usage (GB/month)")
```

```
plt.grid(True)
```

```
plt.show()
```

Output :



➤ **Monthly Recharge Cost Distribution:**

```
import matplotlib.pyplot as plt
```

```
# Plot Monthly Recharge Cost Distribution
```

```
plt.figure(figsize=(8, 5))
```

```
df["Monthly Recharge Cost (INR)"].plot(kind="hist", bins=20,  
color="purple", edgecolor="black")
```

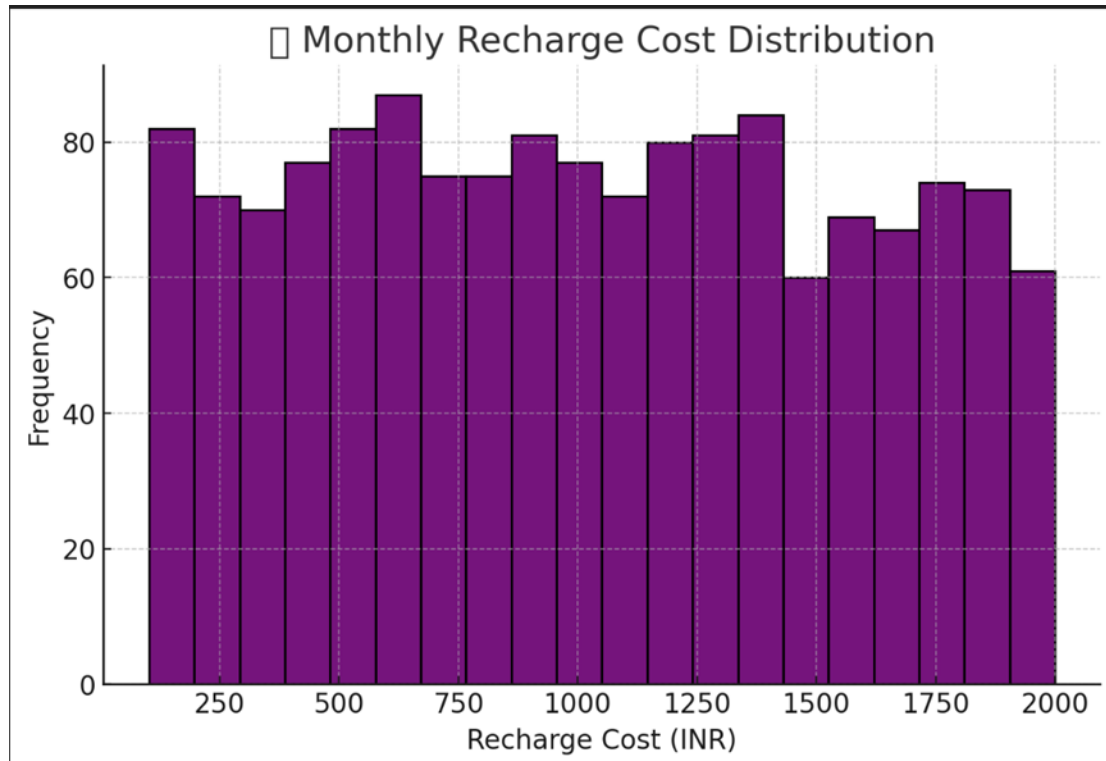
```
plt.title(" Monthly Recharge Cost Distribution")
```

```
plt.xlabel("Recharge Cost (INR)")
```

```
plt.ylabel("Frequency")
```

```
plt.show()
```

Output :

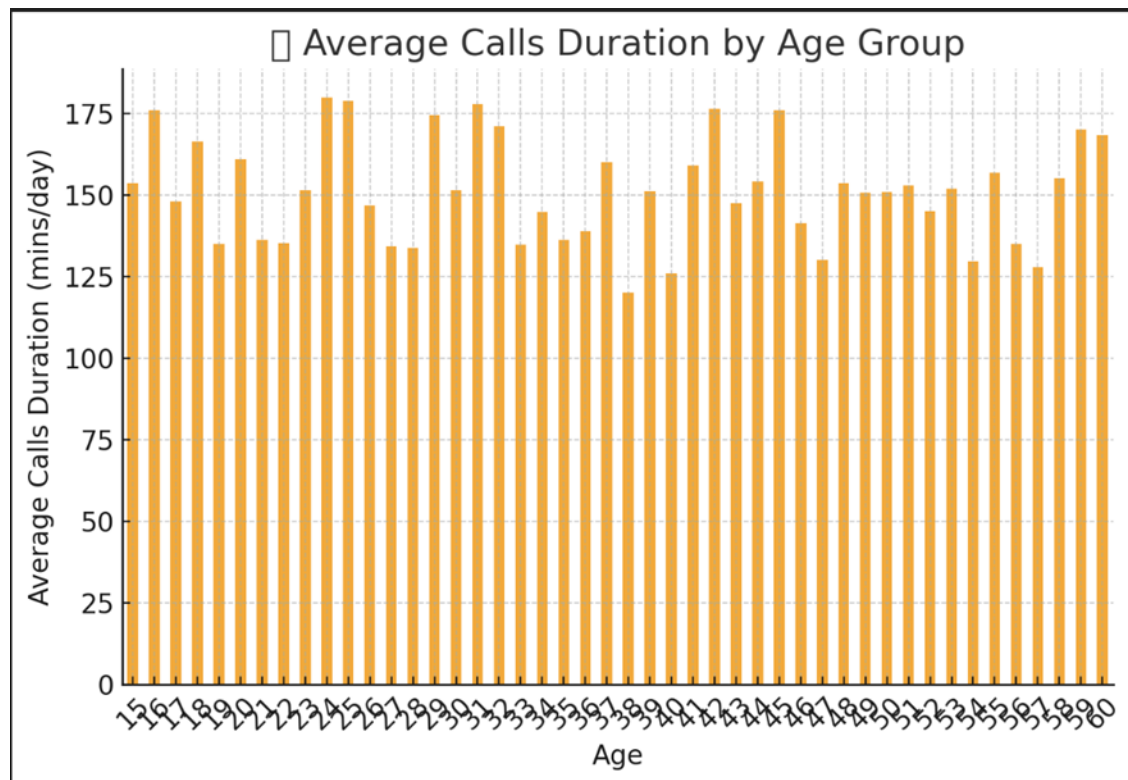


➤ Average Calls Duration by Age Group:

```
import matplotlib.pyplot as plt

# Plot Calls Duration by Age
plt.figure(figsize=(8, 5))
df.groupby("Age")["Calls Duration (mins/day)"].mean().plot(kind="bar", color="orange")
plt.title("Average Calls Duration by Age Group")
plt.xlabel("Age")
plt.ylabel("Average Calls Duration (mins/day)")
plt.xticks(rotation=45)
plt.show()
```

Output :



### ➤ Average Screen Time by Gender:

```
import matplotlib.pyplot as plt
import pandas as pd

Load the dataset

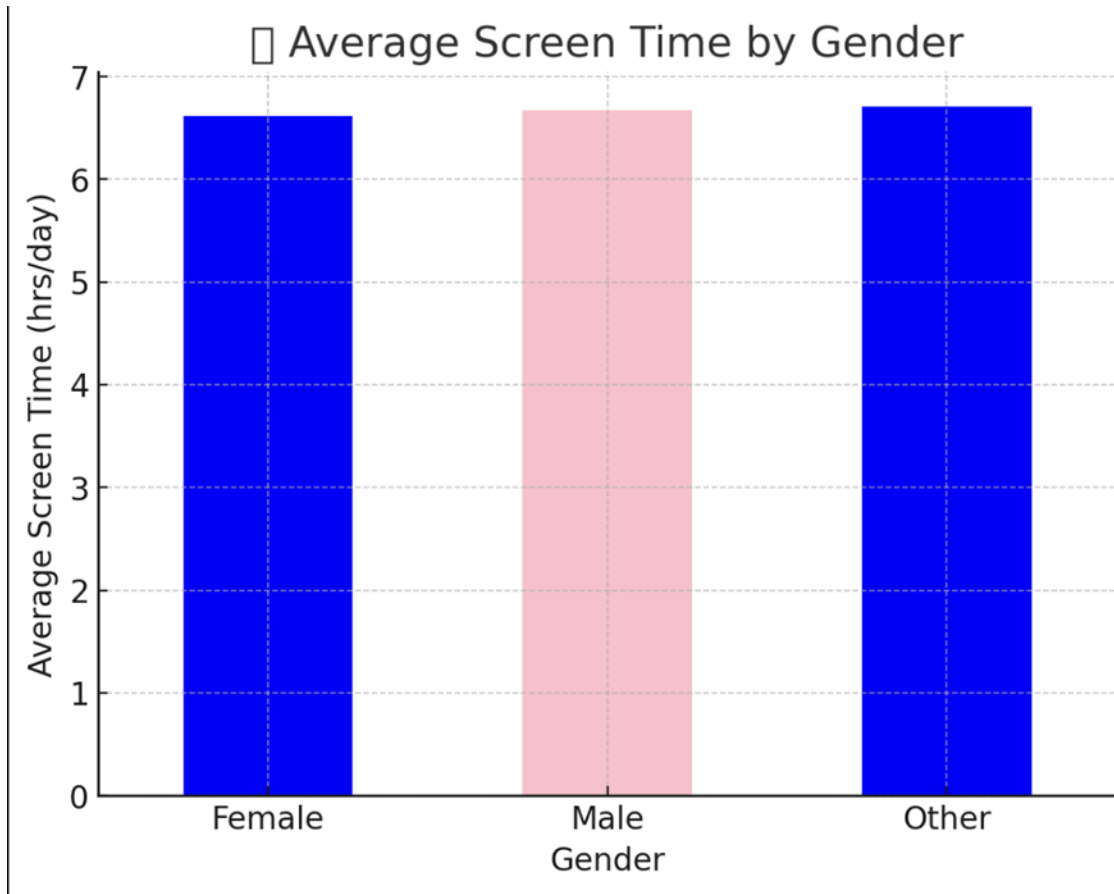
df = pd.read_csv("phone_usage_india.csv")

# Calculate average screen time by gender
avg_screen_time = df.groupby("Gender")["Screen Time
(hrs/day)"].mean()

# Plot
avg_screen_time.plot(kind="bar", color=["blue", "red"])
plt.title(" Average Screen Time by Gender")
plt.xlabel("Gender")
plt.ylabel("Average Screen Time (hrs/day)")
plt.show()
```



Output :



# Screen Time

```
project.py x
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4
5 # Load dataset
6 file_path = "phone_usage_india.csv"
7 df = pd.read_csv(file_path)
8
9 top_locations = df.groupby("Location")["Screen Time (hrs/day)"].mean().sort_values(ascending=False).reset_index().head(10)
10
11 print(top_locations)
```

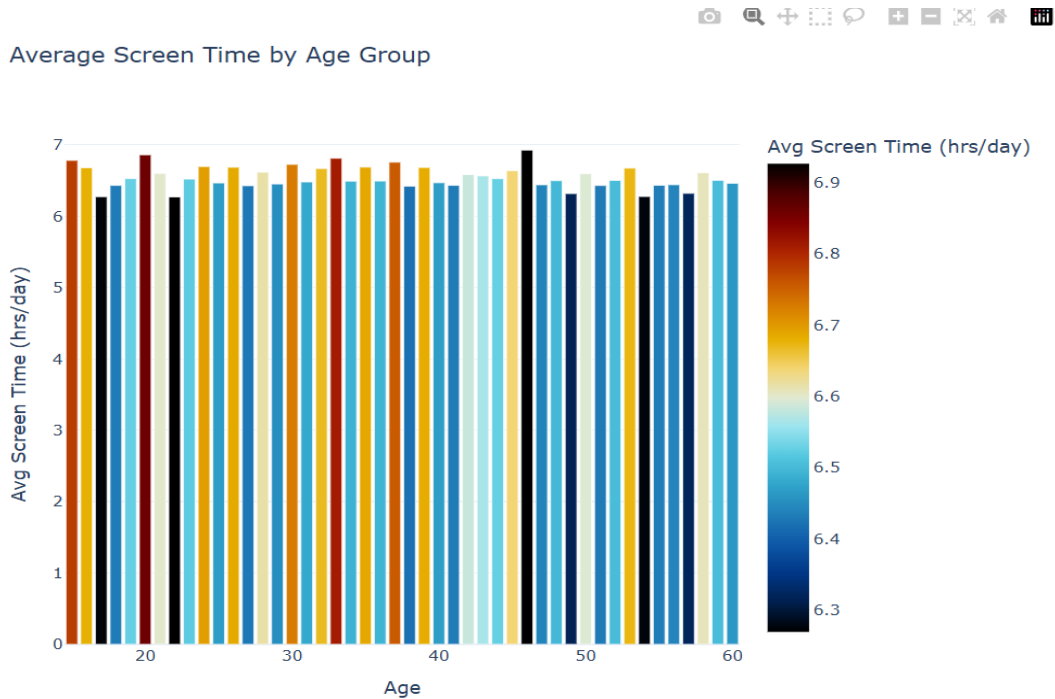
Output :

```
PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
PS C:\python>
PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
  Location  Screen Time (hrs/day)
0      Pune          6.665124
1    Jaipur          6.653509
2  Hyderabad          6.595706
3    Lucknow          6.578824
4  Bangalore          6.572508
5  Ahmedabad          6.563866
6    Chennai          6.527877
7      Delhi          6.468000
8     Mumbai          6.421022
9    Kolkata          6.412353
PS C:\python> |
```

⇒ Average screen time by Age group

```
project.py ×
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv"
6 df = pd.read_csv(file_path)
7
8 # Group data by Age and calculate average screen time
9 df_grouped = df.groupby("Age")["Screen Time (hrs/day)"].mean().reset_index()
10
11 # Create a bar chart
12 fig = px.bar(df_grouped, x="Age", y="Screen Time (hrs/day)",
13             title="Average Screen Time by Age Group",
14             labels={"Age": "Age", "Screen Time (hrs/day)": "Avg Screen Time (hrs/day)"},
15             color="Screen Time (hrs/day)", color_continuous_scale="icefire")
16
17 # Adjust layout
18 fig.update_layout(xaxis_title="Age", yaxis_title="Avg Screen Time (hrs/day)")
19 fig.update_layout(template="plotly_white", height=600, width=800)
20
21 # Show figure
22 fig.show()
23
24
```

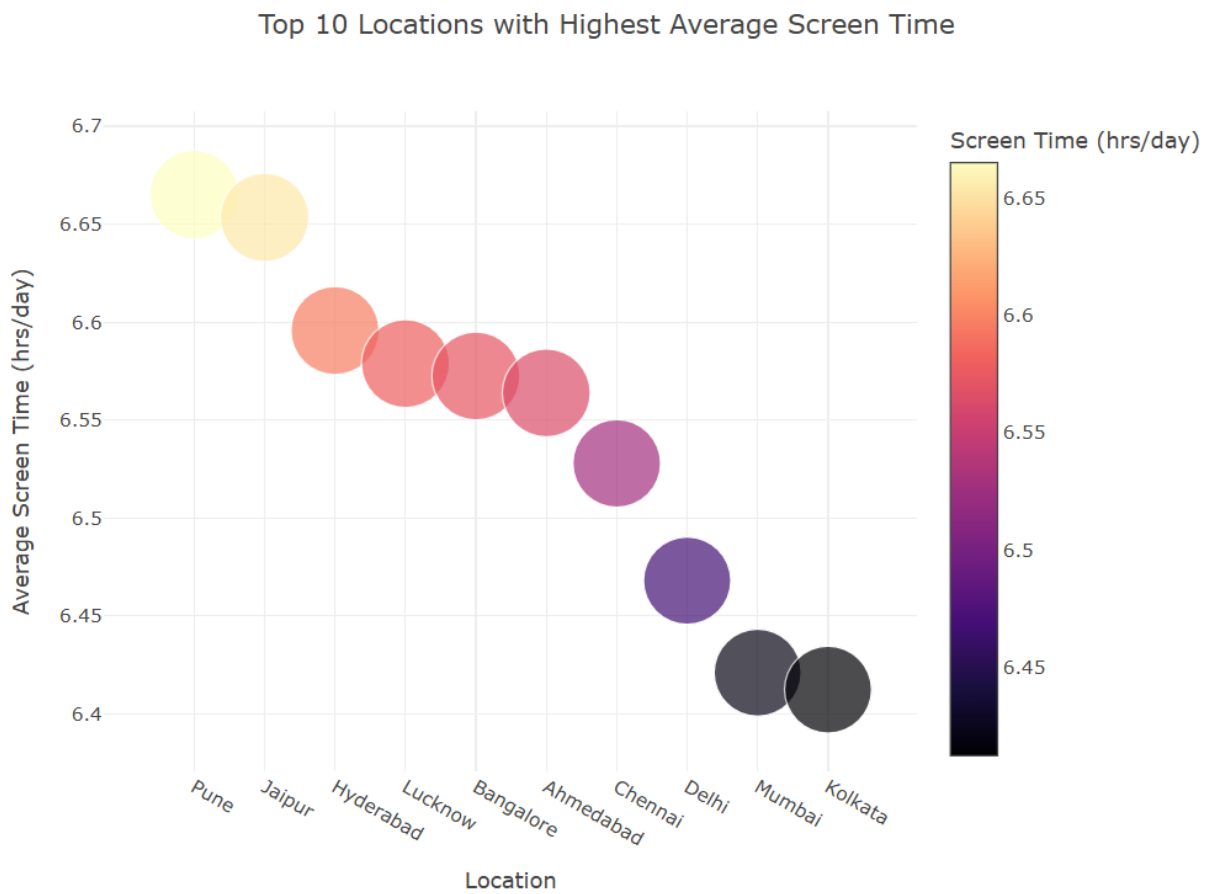
Output :



⇒ Top 10 Location with Highest Average Screen time

```
... project.py X
project.py > ...
1  import plotly.express as px
2  import pandas as pd
3
4  # Load dataset
5  file_path = "phone_usage_india.csv"
6  df = pd.read_csv(file_path)
7
8  # Group by 'Location' and calculate the average screen time
9  top_locations = (
10     df.groupby("Location")["Screen Time (hrs/day)"]
11     .mean()
12     .sort_values(ascending=False)
13     .reset_index()
14     .head(10)
15 )
16
17 # Plot the data as a scatter plot
18 fig = px.scatter(
19     top_locations,
20     x="Location",
21     y="Screen Time (hrs/day)",
22     size="Screen Time (hrs/day)",
23     color="Screen Time (hrs/day)",
24     title="Top 10 Locations with Highest Average Screen Time",
25     color_continuous_scale='Magma',
26     size_max=40
27 )
28
29 fig.update_layout(
30     height=600,
31     width=800,
32     xaxis_title="Location",
33     yaxis_title="Average Screen Time (hrs/day)",
34     showlegend=True,
35 )
36 fig.update_layout(template='gridon')
37 fig.show()
38
```

Output :

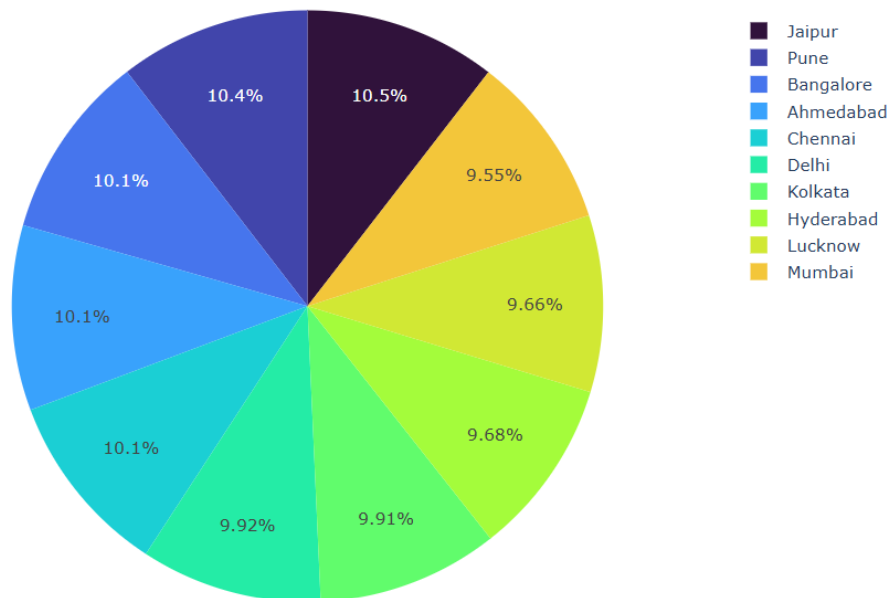


⇒ Total screen time distribution by location

```
project.py X
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv" # Ensure this file exists in the working directory
6 df = pd.read_csv(file_path)
7
8 # Group by Location and calculate total screen time
9 location_screen_time = df.groupby("Location")["Screen Time (hrs/day)"].sum().reset_index()
10
11 print(location_screen_time)
12
13 # Create a pie chart
14 fig = px.pie(location_screen_time, names="Location", values="Screen Time (hrs/day)",
15             title="Total Screen Time Distribution by Location",
16             (parameter) color_discrete_sequence: Any | None }": "Total Screen Time (hrs/day)",
17             color_discrete_sequence=px.colors.sequential.Turbo)
18
19 # Adjust layout
20 fig.update_layout(height=600, width=800, template="plotly_white")
21
22 # Show figure
23 fig.show()
24
25
```

Output:

Total Screen Time Distribution by Location

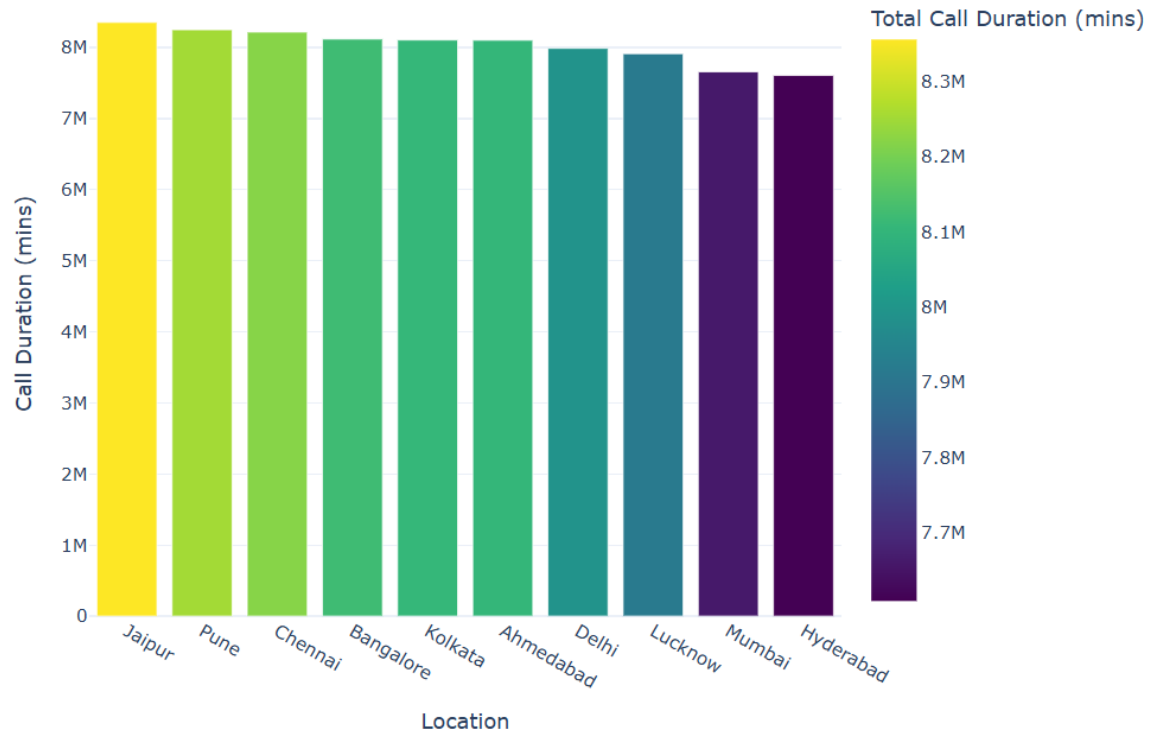


# Call Duration

```
... project.py X
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv"
6 df = pd.read_csv(file_path)
7
8 # Estimate total monthly call duration
9 df["Call Duration (mins/month)"] = df["Calls Duration (mins/day)"] * 30
10
11 # Group by 'Location' and calculate the total call duration
12 top_call_duration = (
13     df.groupby("Location")["Call Duration (mins/month)"]
14     .sum()
15     .sort_values(ascending=False)
16     .reset_index()
17     .head(10)
18 )
19
20 print(top_call_duration)
21
22 # Plot the data as a bar chart
23 fig = px.bar(
24     top_call_duration,
25     x='Location',
26     y='Call Duration (mins/month)',
27     title='Total Call Duration by Top 10 Locations',
28     labels={'Location': 'Location', 'Call Duration (mins/month)': 'Total Call Duration (mins)'},
29     color='Call Duration (mins/month)',
30     color_continuous_scale='viridis'
31 )
32
33 fig.update_layout(xaxis_title='Location', yaxis_title='Call Duration (mins)')
34 fig.update_layout(template='plotly_white', height=600, width=800)
35 fig.show()
36
```

Output :

Total Call Duration by Top 10 Locations

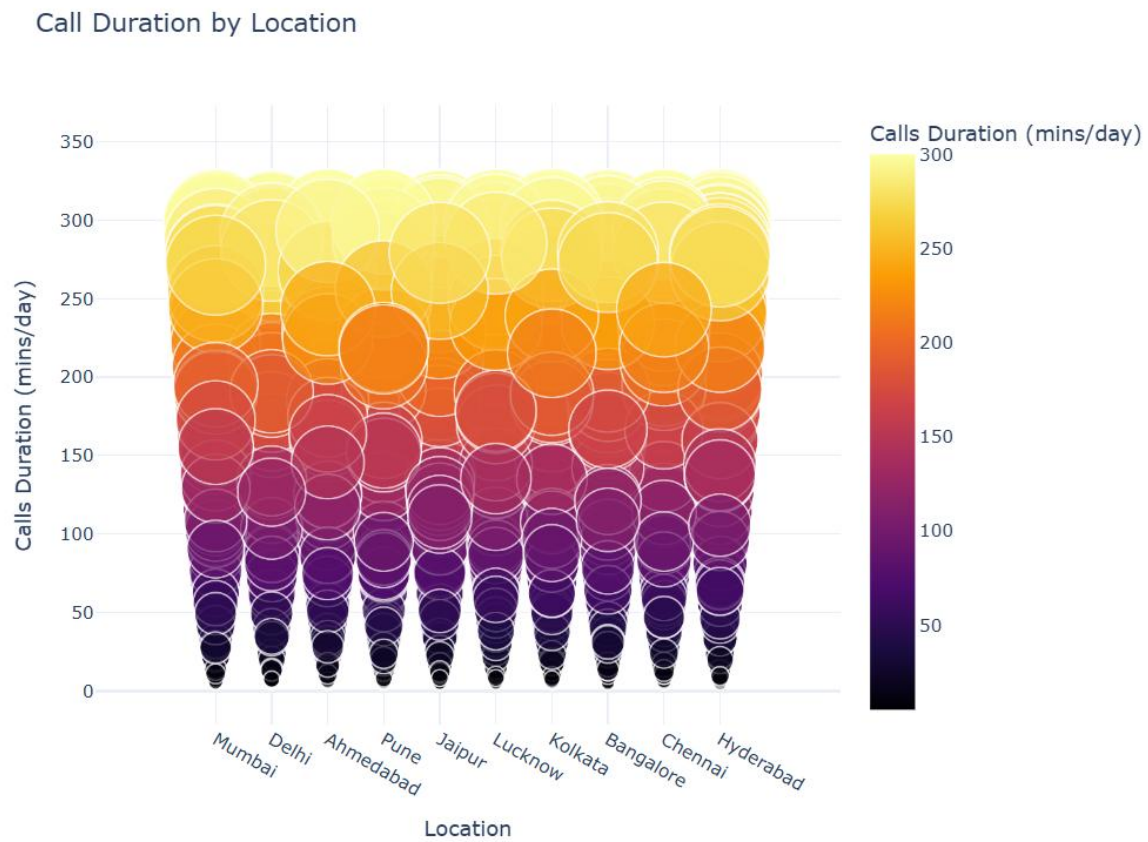




⇒ Call duration by location

```
... project.py X
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv" # Ensure correct file path
6 df = pd.read_csv(file_path)
7
8 # Create scatter plot using Location and Calls Duration
9 fig = px.scatter(
10     df,
11     x="Location",
12     y="Calls Duration (mins/day)",
13     size="Calls Duration (mins/day)",
14     color="Calls Duration (mins/day)",
15     title="Call Duration by Location",
16     color_continuous_scale='Inferno',
17     size_max=70
18 )
19
20 fig.update_layout(
21     height=600,
22     width=800,
23     xaxis_title="Location",
24     yaxis_title="Calls Duration (mins/day)",
25     showlegend=True,
26 )
27 fig.update_layout(template='plotly_white')
28 fig.show()
29 |
```

Output :



⇒ Total call duration by top 10 locations

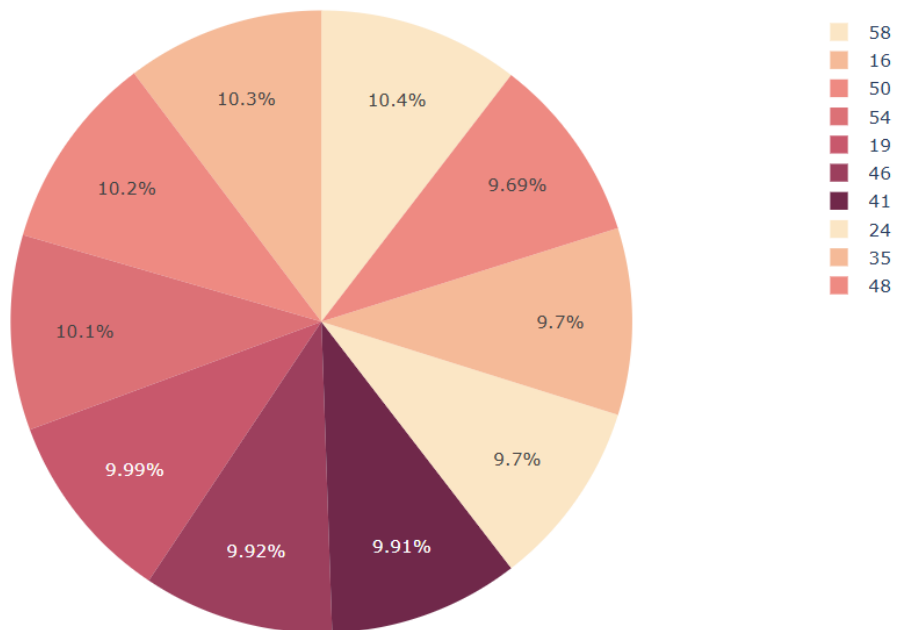
```
... project.py X
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv"
6 df = pd.read_csv(file_path)
7
8 # Estimate total monthly call duration
9 df["Call Duration (mins/month)"] = df["Calls Duration (mins/day)"] * 30
10
11 # Group by 'Location' and calculate the total call duration
12 top_call_counts = (
13     df.groupby("Age")["Call Duration (mins/month)"]
14     .sum()
15     .sort_values(ascending=False)
16     .reset_index()
17     .head(10)
18 )
19
20 print(top_call_counts)
21
22 # Plot the data as a pie chart
23 fig = px.pie(
24     top_call_counts,
25     names='Age',
26     values='Call Duration (mins/month)',
27     title='Total Call Duration by Top 10 Locations',
28     color='Age',
29     color_discrete_sequence=px.colors.sequential.Burgyl
30 )
31
32 fig.update_layout(height=600, width=800)
33 fig.update_layout(template='plotly_white')
34 fig.show()
35
```

## Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Age  Call Duration (mins/month)
0   58      1973112.0
1   16      1944807.0
2   50      1927803.0
3   54      1914141.0
4   19      1886892.0
5   46      1873668.0
6   41      1871085.0
7   24      1832697.0
8   35      1832250.0
9   48      1830357.0
PS C:\python>
```

Total Call Duration by Top 10 Age

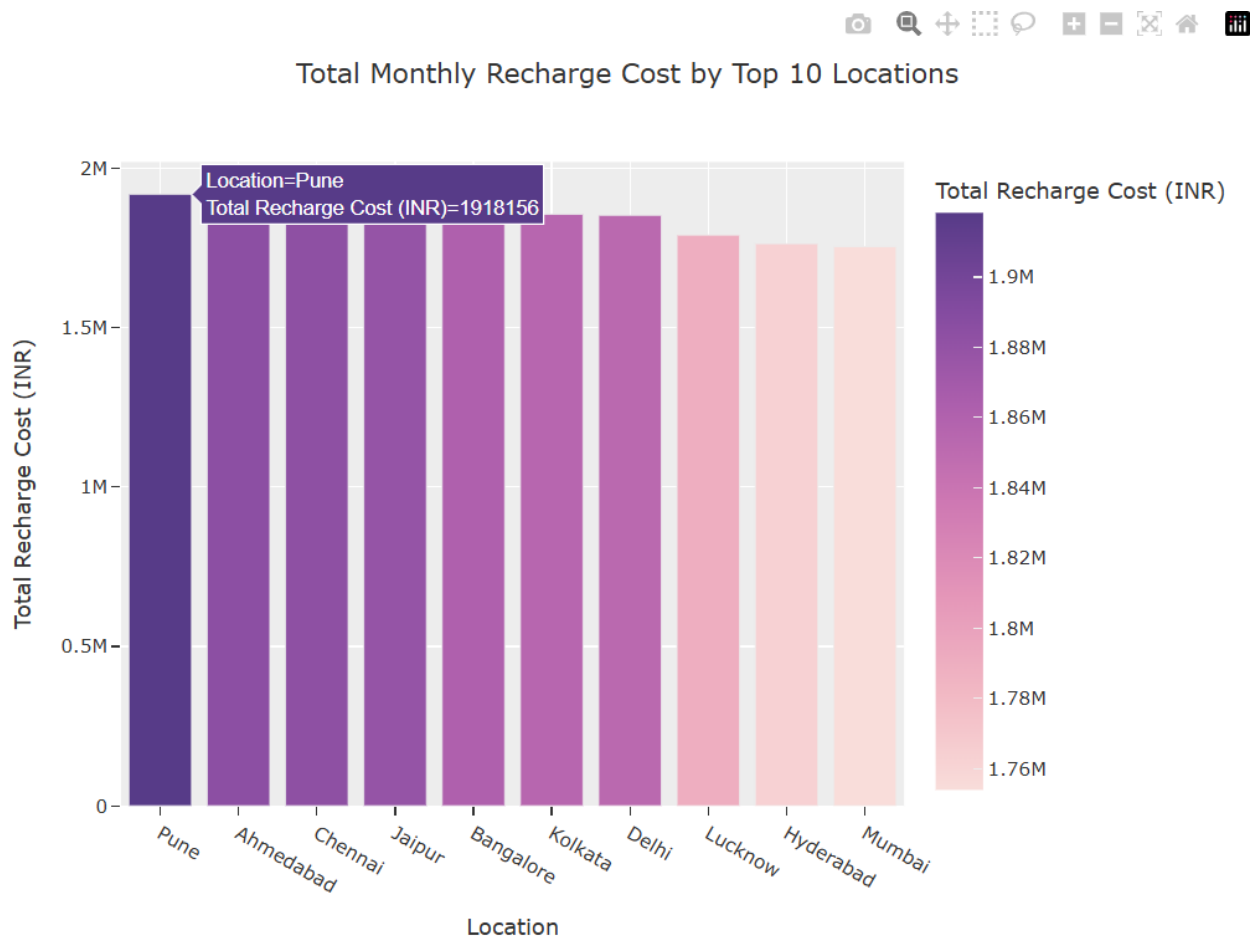


# Monthly Recharge Cost (INR)

```
project.py X
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv"
6 df = pd.read_csv(file_path)
7
8 # Group by 'Location' and calculate the total monthly recharge cost
9 top_recharge_cost = (
10     df.groupby("Location")["Monthly Recharge Cost (INR)"]
11     .sum()
12     .sort_values(ascending=False)
13     .reset_index()
14     .head(10)
15 )
16 print(top_recharge_cost)
17
18 # Plot the data as a bar chart
19 fig = px.bar(
20     top_recharge_cost,
21     x='Location',
22     y='Monthly Recharge Cost (INR)',
23     title='Total Monthly Recharge Cost by Top 10 Locations',
24     labels={'Location': 'Location', 'Monthly Recharge Cost (INR)': 'Total Recharge Cost (INR)'},
25     color='Monthly Recharge Cost (INR)',
26     color_continuous_scale='purpor'
27 )
28
29 fig.update_layout(xaxis_title='Location', yaxis_title='Total Recharge Cost (INR)')
30 fig.update_layout(template='ggplot2')
31 fig.update_layout(height=600, width=800)
32 fig.show()
```

Output :

	Location	Monthly Recharge Cost (INR)
0	Pune	1918156
1	Ahmedabad	1884848
2	Chennai	1883733
3	Jaipur	1879095
4	Bangalore	1861931
5	Kolkata	1855423
6	Delhi	1852209
7	Lucknow	1790224
8	Hyderabad	1763194
9	Mumbai	1753889

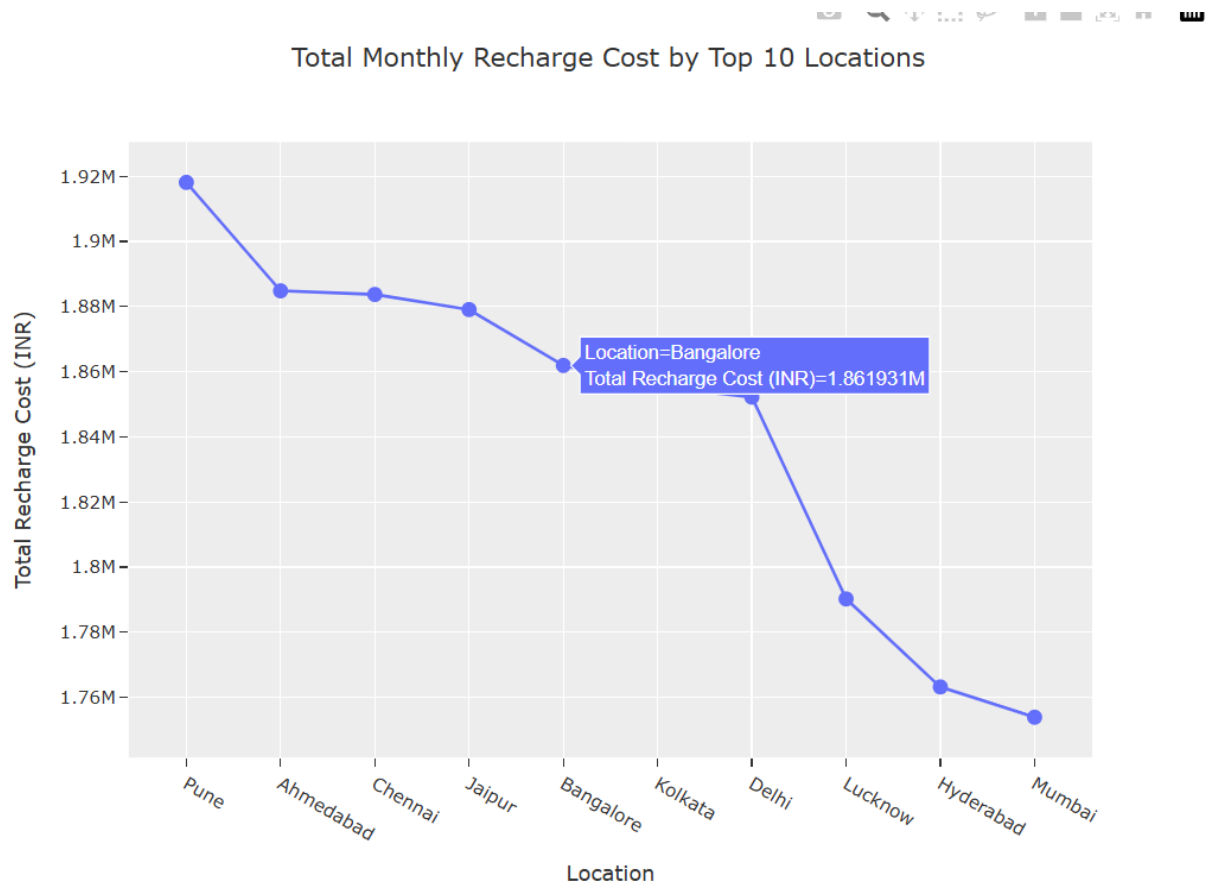


⇒ Total monthly recharge cost by top 10 location

```
project.py X
project.py > top_recharge_cost
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv"
6 df = pd.read_csv(file_path)
7
8 # Group by 'Location' and calculate the total monthly recharge cost
9 top_recharge_cost = (
10     df.groupby("Location")["Monthly Recharge Cost (INR)"]
11     .sum()
12     .sort_values(ascending=False)
13     .reset_index()
14     .head(10)
15 )
16 print(top_recharge_cost)
17
18 # Plot the data as a line chart
19 fig = px.line(
20     top_recharge_cost,
21     x='Location',
22     y='Monthly Recharge Cost (INR)',
23     title='Total Monthly Recharge Cost by Top 10 Locations',
24     labels={'Location': 'Location', 'Monthly Recharge Cost (INR)': 'Total Recharge Cost (INR)'},
25     line_shape="linear",
26     markers=True
27 )
28
29 fig.update_layout(xaxis_title='Location', yaxis_title='Total Recharge Cost (INR)')
30 fig.update_layout(template='ggplot2')
31 fig.update_layout(height=600, width=800)
32 fig.update_traces(marker=dict(size=10))
33 fig.show()
34
35
```

Output :

```
PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Location Monthly Recharge Cost (INR)
0 Pune 1918156
1 Ahmedabad 1884848
2 Chennai 1883733
3 Jaipur 1879095
4 Bangalore 1861931
5 Kolkata 1855423
6 Delhi 1852209
7 Lucknow 1790224
8 Hyderabad 1763194
9 Mumbai 1753889
PS C:\python>
```





⇒ Total monthly recharge cost by top 10 locations

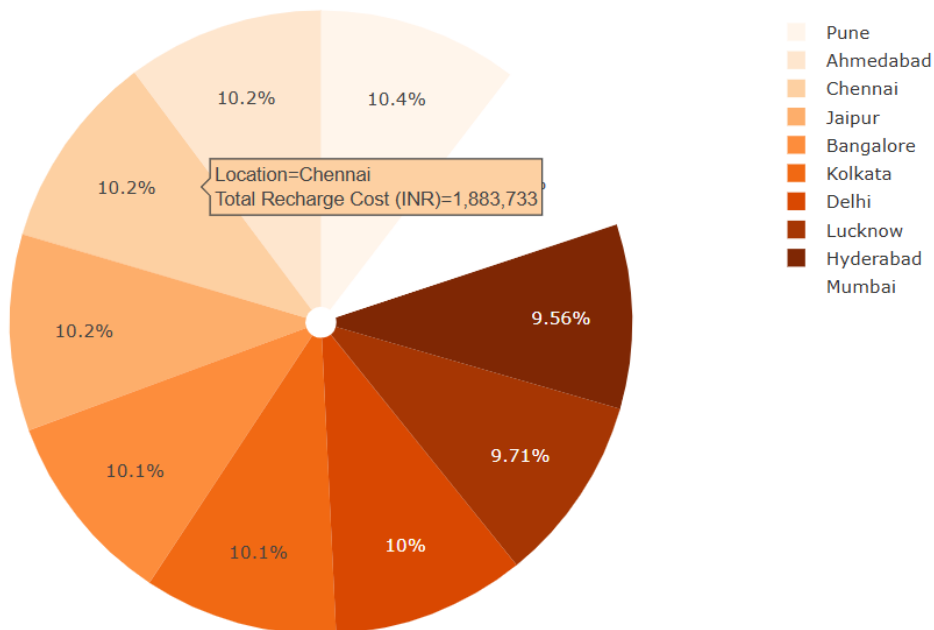
```
project.py X
project.py > ...
1  import plotly.express as px
2  import pandas as pd
3
4  # Load dataset
5  file_path = "phone_usage_india.csv"
6  df = pd.read_csv(file_path)
7
8  # Group by 'Location' and sum Monthly Recharge Cost
9  top_recharge_cost = (
10     df.groupby("Location")["Monthly Recharge Cost (INR)"]
11         .sum()
12         .sort_values(ascending=False)
13         .reset_index()
14         .head(10)
15 )
16 print(top_recharge_cost)
17
18 # Plot the data as a pie chart
19 fig = px.pie(
20     top_recharge_cost,
21     names='Location',
22     values='Monthly Recharge Cost (INR)',
23     title='Total Monthly Recharge Cost by Top 10 Locations',
24     labels={'Location': 'Location', 'Monthly Recharge Cost (INR)': 'Total Recharge Cost (INR)'},
25     color_discrete_sequence=px.colors.sequential.Oranges,
26     hole=0.05
27 )
28
29 fig.update_layout(template='ggplot2')
30 fig.update_layout(height=600, width=800)
31 fig.show()
```

## Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Location Monthly Recharge Cost (INR)
0 Pune 1918156
1 Ahmedabad 1884848
2 Chennai 1883733
3 Jaipur 1879095
4 Bangalore 1861931
5 Kolkata 1855423
6 Delhi 1852209
7 Lucknow 1790224
8 Hyderabad 1763194
9 Mumbai 1753889
PS C:\python>
```

Total Monthly Recharge Cost by Top 10 Locations

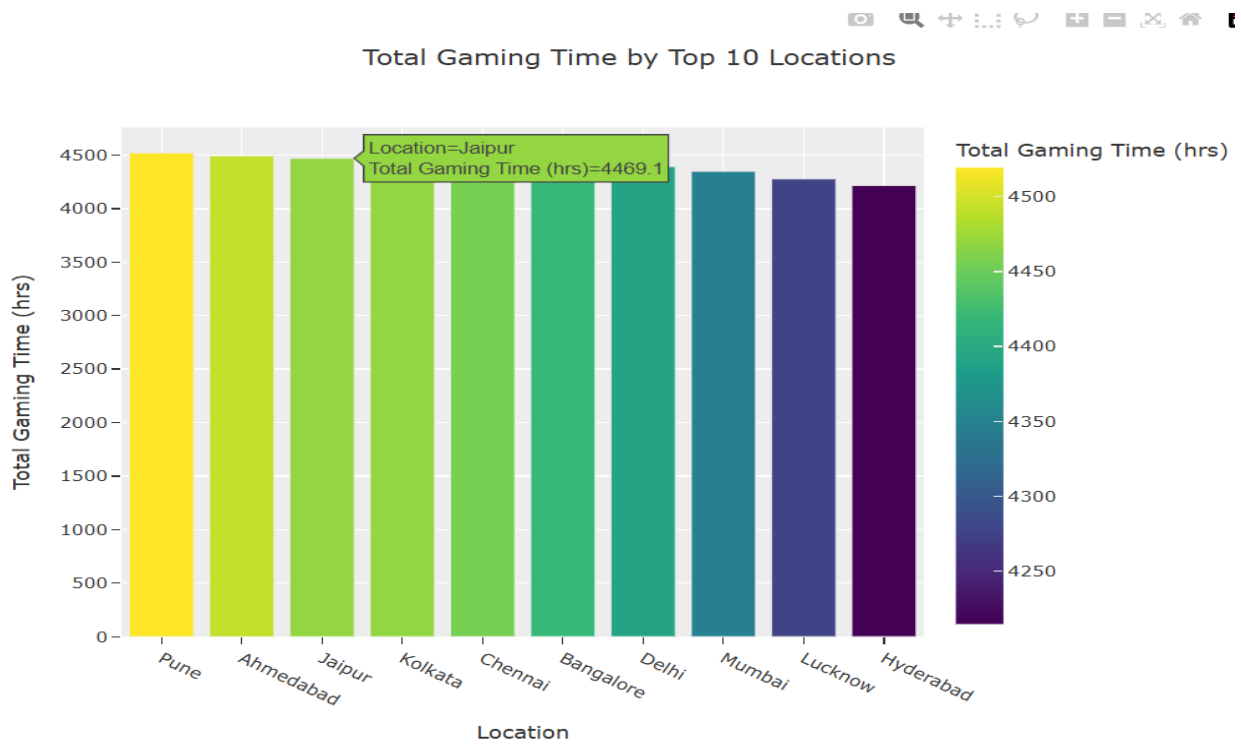


# Gaming Time (hrs/day)

```
project.py X
project.py > ...
1  import plotly.express as px
2  import pandas as pd
3
4  # Load dataset
5  file_path = "phone_usage_india.csv"
6  df = pd.read_csv(file_path)
7
8  # Group by 'Location' and calculate the total gaming time
9  top_gaming_time = (
10     df.groupby("Location")["Gaming Time (hrs/day)"]
11         .sum()
12         .sort_values(ascending=False)
13         .reset_index()
14         .head(10)
15 )
16
17 print(top_gaming_time)
18
19 # Plot the data as a bar chart
20 fig = px.bar(
21     top_gaming_time,
22     x='Location',
23     y='Gaming Time (hrs/day)',
24     title='Total Gaming Time by Top 10 Locations',
25     labels={'Location': 'Location', 'Gaming Time (hrs/day)': 'Total Gaming Time (hrs)'},
26     color='Gaming Time (hrs/day)',
27     color_continuous_scale='Viridis'
28 )
29
30 fig.update_layout(xaxis_title='Location', yaxis_title='Total Gaming Time (hrs)')
31 fig.update_layout(template='ggplot2')
32 fig.update_layout(height=600, width=800)
33 fig.show()
34
35
```

## Output :

```
PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Location Gaming Time (hrs/day)
0 Pune 4519.2
1 Ahmedabad 4491.4
2 Jaipur 4469.1
3 Kolkata 4468.6
4 Chennai 4456.2
5 Bangalore 4419.1
6 Delhi 4391.7
7 Mumbai 4347.3
8 Lucknow 4276.4
9 Hyderabad 4214.6
```

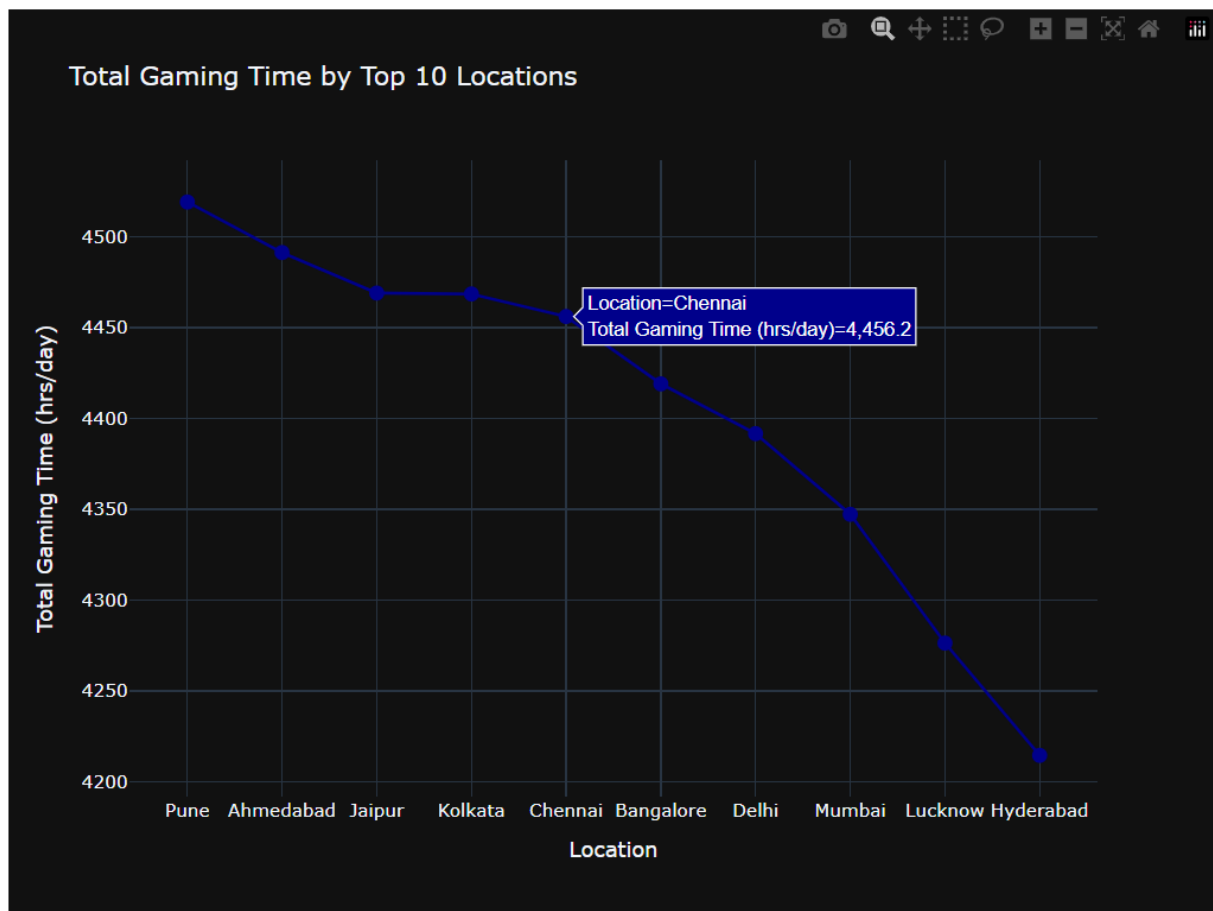


⇒ Total gaming time by 10 locations

```
project.py X
project.py > ...
1  import plotly.express as px
2  import pandas as pd
3
4  # Load dataset
5  file_path = "phone_usage_india.csv"
6  df = pd.read_csv(file_path)
7
8  # Group by 'Location' and calculate the total gaming time
9  top_gaming_time = (
10     df.groupby("Location")["Gaming Time (hrs/day)"]
11     .sum()
12     .sort_values(ascending=False)
13     .reset_index()
14     .head(10)
15 )
16 print(top_gaming_time)
17
18 # Plot the data as a line chart
19 fig = px.line(
20     top_gaming_time,
21     x='Location',
22     y='Gaming Time (hrs/day)',
23     title='Total Gaming Time by Top 10 Locations',
24     labels={'Location': 'Location', 'Gaming Time (hrs/day)': 'Total Gaming Time (hrs/day)'},
25     line_shape="linear",
26     markers=True,
27     color_discrete_sequence=['darkblue']
28 )
29
30 fig.update_layout(xaxis_title='Location', yaxis_title='Total Gaming Time (hrs/day)')
31 fig.update_layout(template='plotly_dark')
32 fig.update_layout(height=600, width=800)
33 fig.update_traces(marker=dict(size=10))
34 fig.show()
```

Output :

```
PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Location Gaming Time (hrs/day)
0 Pune 4519.2
1 Ahmedabad 4491.4
2 Jaipur 4469.1
3 Kolkata 4468.6
4 Chennai 4456.2
5 Bangalore 4419.1
6 Delhi 4391.7
7 Mumbai 4347.3
8 Lucknow 4276.4
9 Hyderabad 4214.6
PS C:\python>
```



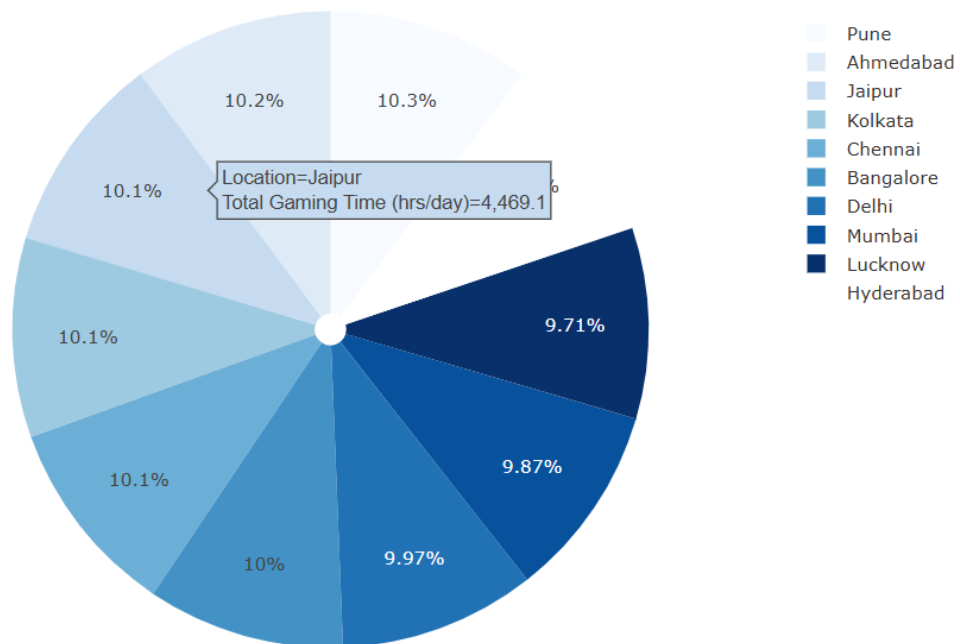
⇒ Total gaming time by 10 locations

```
project.py X
project.py > ...
1  import plotly.express as px
2  import pandas as pd
3
4  # Load dataset
5  file_path = "phone_usage_india.csv"
6  df = pd.read_csv(file_path)
7
8  # Group by 'Location' and sum Gaming Time
9  top_gaming_time = (
10     df.groupby("Location")["Gaming Time (hrs/day)"]
11     .sum()
12     .sort_values(ascending=False)
13     .reset_index()
14     .head(10)
15 )
16 print(top_gaming_time)
17
18 # Plot the data as a pie chart
19 fig = px.pie(
20     top_gaming_time,
21     names='Location',
22     values='Gaming Time (hrs/day)',
23     title='Total Gaming Time by Top 10 Locations',
24     labels={'Location': 'Location', 'Gaming Time (hrs/day)': 'Total Gaming Time (hrs/day)'},
25     color_discrete_sequence=px.colors.sequential.Blues,
26     hole=0.05
27 )
28
29 fig.update_layout(template='ggplot2')
30 fig.update_layout(height=600, width=800)
31 fig.show()
```

## Output :

```
9      Mumbai      1753889
PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Location Gaming Time (hrs/day)
0      Pune      4519.2
1      Ahmedabad  4491.4
2      Jaipur     4469.1
3      Kolkata    4468.6
4      Chennai    4456.2
5      Bangalore  4419.1
6      Delhi      4391.7
7      Mumbai     4347.3
8      Lucknow    4276.4
9      Hyderabad  4214.6
PS C:\python>
```

Total Gaming Time by Top 10 Locations





# Primary Use

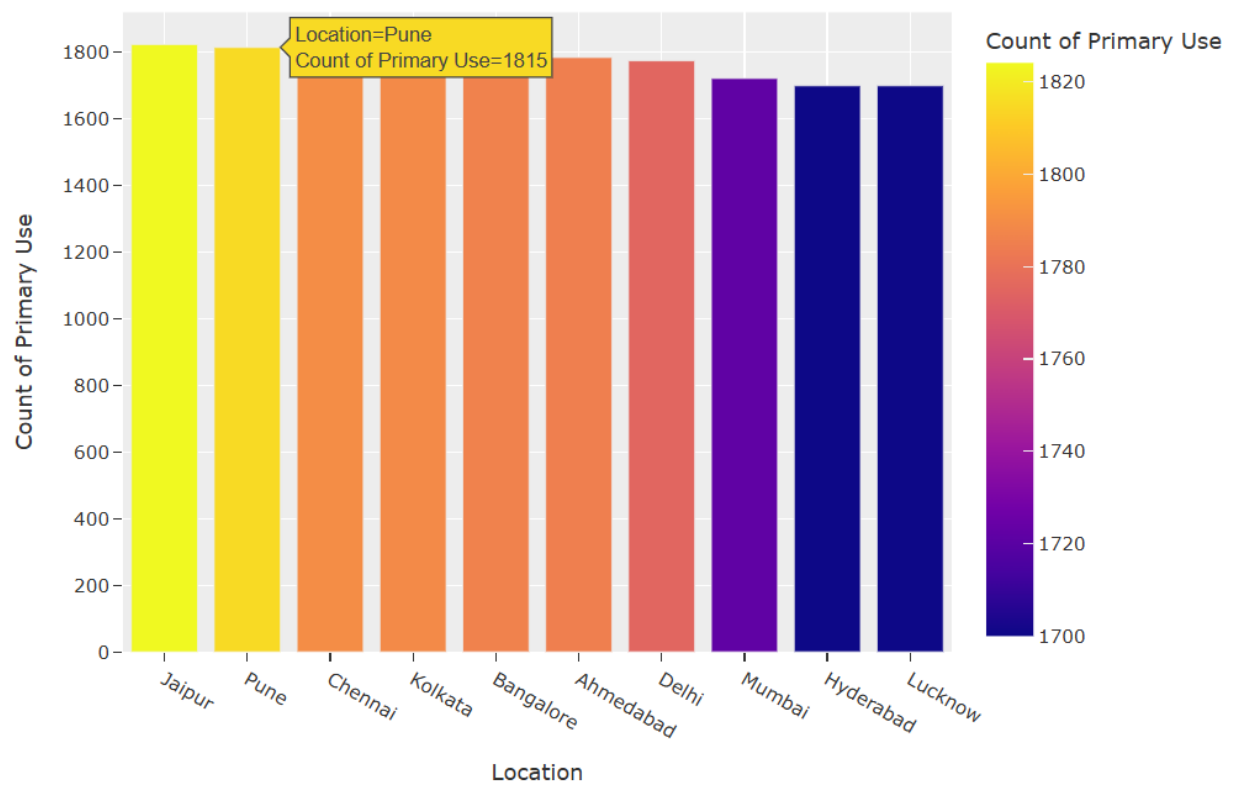
```
project.py X
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv"
6 df = pd.read_csv(file_path)
7
8 # Group by 'Location' and calculate the total primary use occurrences
9 top_primary_use = (
10     df.groupby("Location")["Primary Use"]
11     .count()
12     .sort_values(ascending=False)
13     .reset_index()
14     .head(10)
15 )
16 print(top_primary_use)
17
18 # Plot the data as a bar chart
19 fig = px.bar(
20     top_primary_use,
21     x='Location',
22     y='Primary Use',
23     title='Top 10 Locations by Primary Use',
24     labels={'Location': 'Location', 'Primary Use': 'Count of Primary Use'},
25     color='Primary Use',
26     color_continuous_scale='Plasma'
27 )
28
29 fig.update_layout(xaxis_title='Location', yaxis_title='Count of Primary Use')
30 fig.update_layout(template='ggplot2')
31 fig.update_layout(height=600, width=800)
32 fig.show()
```

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Location Primary Use
0 Jaipur 1824
1 Pune 1815
2 Chennai 1790
3 Kolkata 1789
4 Bangalore 1786
5 Ahmedabad 1785
6 Delhi 1775
7 Mumbai 1722
8 Hyderabad 1700
9 Lucknow 1700
PS C:\python>
```

Top 10 Locations by Primary Use

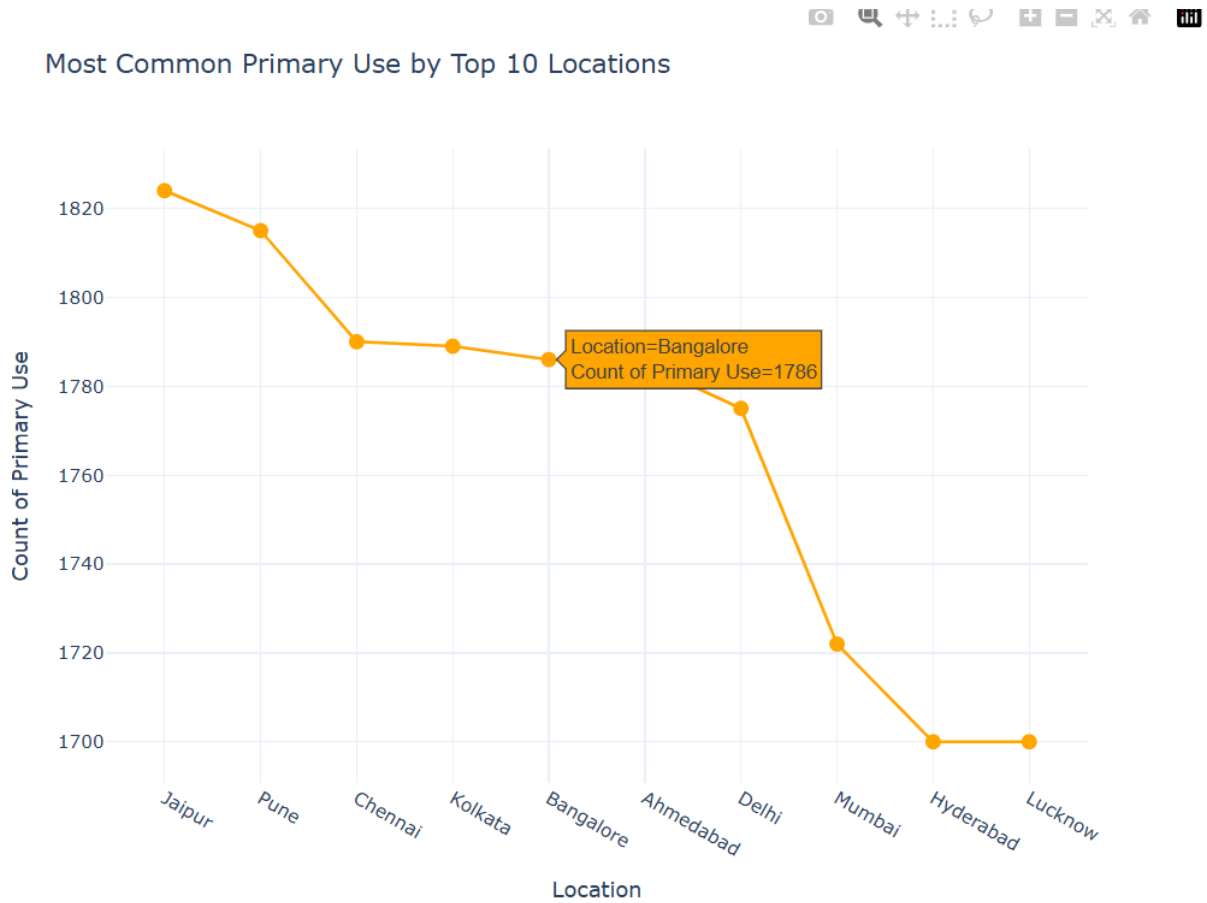


⇒ Most common primary use by top 10 locations

```
project.py X
project.py > ...
1 import plotly.express as px
2 import pandas as pd
3
4 # Load dataset
5 file_path = "phone_usage_india.csv"
6 df = pd.read_csv(file_path)
7
8 # Group by 'Location' and count occurrences of Primary Use
9 top_primary_use = (
10     df.groupby("Location")["Primary Use"]
11     .count()
12     .sort_values(ascending=False)
13     .reset_index()
14     .head(10)
15 )
16 print(top_primary_use)
17
18 # Plot the data as a line chart
19 fig = px.line(
20     top_primary_use,
21     x='Location',
22     y='Primary Use',
23     title='Most Common Primary Use by Top 10 Locations',
24     labels={'Location': 'Location', 'Primary Use': 'Count of Primary Use'},
25     line_shape="linear",
26     markers=True,
27     color_discrete_sequence=['orange']
28 )
29
30 fig.update_layout(xaxis_title='Location', yaxis_title='Count of Primary Use')
31 fig.update_layout(template='plotly_white')
32 fig.update_layout(height=600, width=800)
33 fig.update_traces(marker=dict(size=10))
34 fig.show()
35
```

Output :

```
PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Location Primary Use
0 Jaipur 1824
1 Pune 1815
2 Chennai 1790
3 Kolkata 1789
4 Bangalore 1786
5 Ahmedabad 1785
6 Delhi 1775
7 Mumbai 1722
8 Hyderabad 1700
9 Lucknow 1700
PS C:\python>
```



⇒ Top 10 primary use of phones

```
project.py ×
project.py > ...
1  import plotly.express as px
2  import pandas as pd
3
4  # Load dataset
5  file_path = "phone_usage_india.csv"
6  df = pd.read_csv(file_path)
7
8  # Group by 'Primary Use' and count occurrences
9  top_primary_use = (
10     df["Primary Use"].value_counts()
11     .reset_index()
12     .rename(columns={"index": "Primary Use", "Primary Use": "Count"})
13     .head(10)
14 )
15 print(top_primary_use)
16
17 # Plot the data as a pie chart
18 fig = px.pie(
19     top_primary_use,
20     names='Primary Use',
21     values='Count',
22     title='Top 10 Primary Uses of Phones',
23     labels={'Primary Use': 'Primary Use', 'Count': 'Usage Count'},
24     color_discrete_sequence=px.colors.sequential.Plasma,
25     hole=0.05
26 )
27
28 fig.update_layout(template='ggplot2')
29 fig.update_layout(height=600, width=800)
30 fig.show()
31
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\python> & C:/Users/jaspa/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/python/project.py
Primary Use Count
0 Education 3601
1 Gaming 3576
2 Work 3557
3 Social Media 3501
4 Entertainment 3451
PS C:\python>
```

Top 10 Primary Uses of Phones

