```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

```
df = pd.read_excel("/content/Matiks - Data Analyst Data.xlsx")
```

```
df.head()
```

| | User_ID | Username | Email | Signup_Date | Country | Age | Gender | Device_Type | Game_Title | Total_Play_Ses |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7280e6c4-6f7c-45dd-a8fc-c58389ea8e07 | geoffreyanderson | haleymitchell@gmail.com | 2024-12-15 | Austria | 22 | Other | Mobile | MysticWar | |
| **1** | 23c48d4f-f5d0-4ff4-ba0f-2007441b9b57 | riverachristian | masonmelissa@hotmail.com | 2024-03-07 | Gabon | 22 | Other | PC | QuestRaid | |
| **2** | cf8d530c-c137-4346-a78b-e76e36d45e2a | brownchris | mnichols@mcmillan.net | 2023-10-19 | Ireland | 36 | Female | PC | QuestRaid | |
| **3** | 47fcbe87-a1c1-40c3-b450-1b5692f61538 | christopher90 | ttaylor@gmail.com | 2023-09-28 | Belarus | 23 | Other | PC | QuestRaid | |
| **4** | 0b620a32-9e77-4b4a-9931-f0b654bef095 | vfreeman | amanda80@gmail.com | 2024-08-09 | Slovenia | 26 | Other | PC | QuestRaid | |

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.describe()
```

| | Signup_Date | Age | Total_Play_Sessions | Avg_Session_Duration_Min | Total_Hours_Played | In_Game_Purchases_Count | Tota |
|---|---|---|---|---|---|---|---|
| **count** | 10000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.00000 | |
| **mean** | 2024-05-22 18:18:43.200000 | 31.063700 | 19.993900 | 30.035238 | 100.216981 | 4.97600 | |
| **min** | 2023-05-22 00:00:00 | 13.000000 | 6.000000 | -5.640000 | -89.730000 | 0.00000 | |
| **25%** | 2023-11-24 00:00:00 | 22.000000 | 17.000000 | 23.210000 | 66.490000 | 3.00000 | |
| **50%** | 2024-05-23 00:00:00 | 31.000000 | 20.000000 | 29.860000 | 100.435000 | 5.00000 | |
| **75%** | 2024-11-22 00:00:00 | 40.000000 | 23.000000 | 36.902500 | 133.900000 | 6.00000 | |
| **max** | 2025-05-21 00:00:00 | 49.000000 | 42.000000 | 75.620000 | 283.260000 | 17.00000 | |
| **std** | NaN | 10.687547 | 4.492314 | 10.062647 | 49.642141 | 2.23623 | |

```
df.isnull().sum()
```

| | 0 |
|---|---|
| User_ID | 0 |
| Username | 0 |
| Email | 0 |
| Signup_Date | 0 |
| Country | 0 |
| Age | 0 |
| Gender | 0 |
| Device_Type | 0 |
| Game_Title | 0 |
| Total_Play_Sessions | 0 |
| Avg_Session_Duration_Min | 0 |
| Total_Hours_Played | 0 |
| In_Game_Purchases_Count | 0 |
| Total_Revenue_USD | 0 |
| Last_Login | 0 |
| Subscription_Tier | 0 |
| Referral_Source | 0 |
| Preferred_Game_Mode | 0 |
| Rank_Tier | 0 |
| Achievement_Score | 0 |

dtype: int64

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   User_ID                   10000 non-null  object
 1   Username                  10000 non-null  object
 2   Email                     10000 non-null  object
 3   Signup_Date               10000 non-null  datetime64[ns]
 4   Country                   10000 non-null  object
 5   Age                       10000 non-null  int64
 6   Gender                    10000 non-null  object
 7   Device_Type               10000 non-null  object
 8   Game_Title                10000 non-null  object
 9   Total_Play_Sessions       10000 non-null  int64
 10  Avg_Session_Duration_Min  10000 non-null  float64
 11  Total_Hours_Played        10000 non-null  float64
 12  In_Game_Purchases_Count   10000 non-null  int64
 13  Total_Revenue_USD         10000 non-null  float64
 14  Last_Login                10000 non-null  datetime64[ns]
 15  Subscription_Tier         10000 non-null  object
 16  Referral_Source           10000 non-null  object
 17  Preferred_Game_Mode       10000 non-null  object
 18  Rank_Tier                 10000 non-null  object
 19  Achievement_Score         10000 non-null  int64
dtypes: datetime64[ns](2), float64(3), int64(4), object(11)
memory usage: 1.5+ MB
```

## Prepare Date Fields

```
df['Last_Login_Date'] = df['Last_Login'].dt.date
df['Week'] = df['Last_Login'].dt.to_period('W').apply(lambda r: r.start_time)
df['Month'] = df['Last_Login'].dt.to_period('M').dt.to_timestamp()
```

```
df['Last_Login_Date']
```

| | Last_Login_Date |
|---|---|
| **0** | 2025-05-19 |
| **1** | 2025-05-12 |
| **2** | 2025-05-03 |
| **3** | 2025-05-08 |
| **4** | 2025-04-24 |
| **...** | ... |
| **9995** | 2025-05-13 |
| **9996** | 2025-04-29 |
| **9997** | 2025-04-22 |
| **9998** | 2025-05-19 |
| **9999** | 2025-05-02 |

10000 rows × 1 columns

**dtype:** object

```
df['Week']
```

| | Week |
|---|---|
| **0** | 2025-05-19 |
| **1** | 2025-05-12 |
| **2** | 2025-04-28 |
| **3** | 2025-05-05 |
| **4** | 2025-04-21 |
| **...** | ... |
| **9995** | 2025-05-12 |
| **9996** | 2025-04-28 |
| **9997** | 2025-04-21 |
| **9998** | 2025-05-19 |
| **9999** | 2025-04-28 |

10000 rows × 1 columns

**dtype:** datetime64[ns]

```
df['Month'].head()
```

| | Month |
|---|---|
| **0** | 2025-05-01 |
| **1** | 2025-05-01 |
| **2** | 2025-05-01 |
| **3** | 2025-05-01 |
| **4** | 2025-04-01 |

**dtype:** datetime64[ns]

## Calculate DAU/WAU/MAU

```
dau = df.groupby('Last_Login_Date')['User_ID'].nunique().reset_index(name='DAU')
wau = df.groupby('Week')['User_ID'].nunique().reset_index(name='WAU')
mau = df.groupby('Month')['User_ID'].nunique().reset_index(name='MAU')
```

```
print(dau)
print(wau)
print(mau)
```

```
    Last_Login_Date  DAU
0        2025-04-22  343
1        2025-04-23  323
```

```
2       2025-04-24  403
3       2025-04-25  322
4       2025-04-26  311
5       2025-04-27  347
6       2025-04-28  325
7       2025-04-29  334
8       2025-04-30  320
9       2025-05-01  318
10      2025-05-02  308
11      2025-05-03  342
12      2025-05-04  353
13      2025-05-05  333
14      2025-05-06  327
15      2025-05-07  355
16      2025-05-08  346
17      2025-05-09  350
18      2025-05-10  333
19      2025-05-11  315
20      2025-05-12  338
21      2025-05-13  318
22      2025-05-14  333
23      2025-05-15  341
24      2025-05-16  345
25      2025-05-17  324
26      2025-05-18  335
27      2025-05-19  301
28      2025-05-20  325
29      2025-05-21  332
        Week   WAU
0 2025-04-21  2049
1 2025-04-28  2300
2 2025-05-05  2359
3 2025-05-12  2334
4 2025-05-19   958
       Month   MAU
0 2025-04-01  3028
1 2025-05-01  6972
```

## Revenue Trend Over Time

```
df['Month'] = df['Last_Login'].dt.to_period('M').dt.to_timestamp()
monthly_revenue = df.groupby('Month')['Total_Revenue_USD'].sum().reset_index()
```

```
monthly_revenue
```

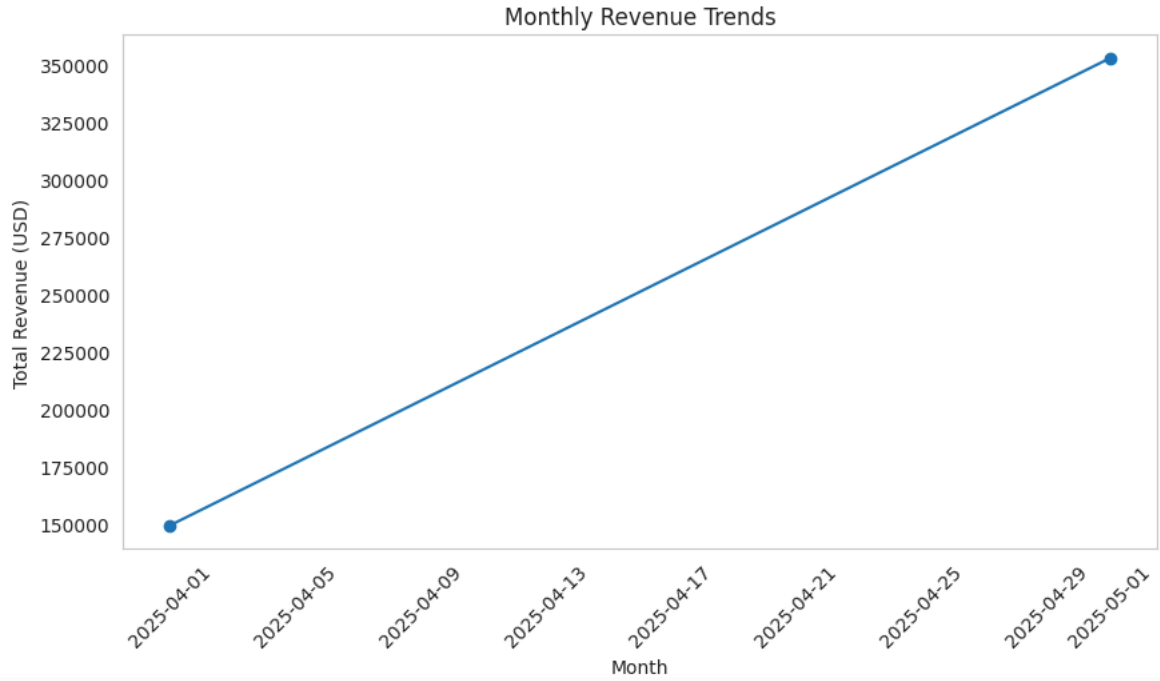
| | Month | Total_Revenue_USD |
|---|---|---|
| 0 | 2025-04-01 | 149031.48 |
| 1 | 2025-05-01 | 352743.18 |

Next steps: Generate code with monthly_revenue | View recommended plots | New interactive sheet

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
plt.plot(monthly_revenue['Month'], monthly_revenue['Total_Revenue_USD'], marker='o', linestyle='-')

plt.xlabel("Month")
plt.ylabel("Total Revenue (USD)")
plt.title("Monthly Revenue Trends")
plt.xticks(rotation=45)
plt.grid()
plt.show()
```




```
df['Week'] = df['Last_Login'].dt.to_period('W').apply(lambda r: r.start_time)
weekly_revenue = df.groupby('Week')['Total_Revenue_USD'].sum().reset_index()
```

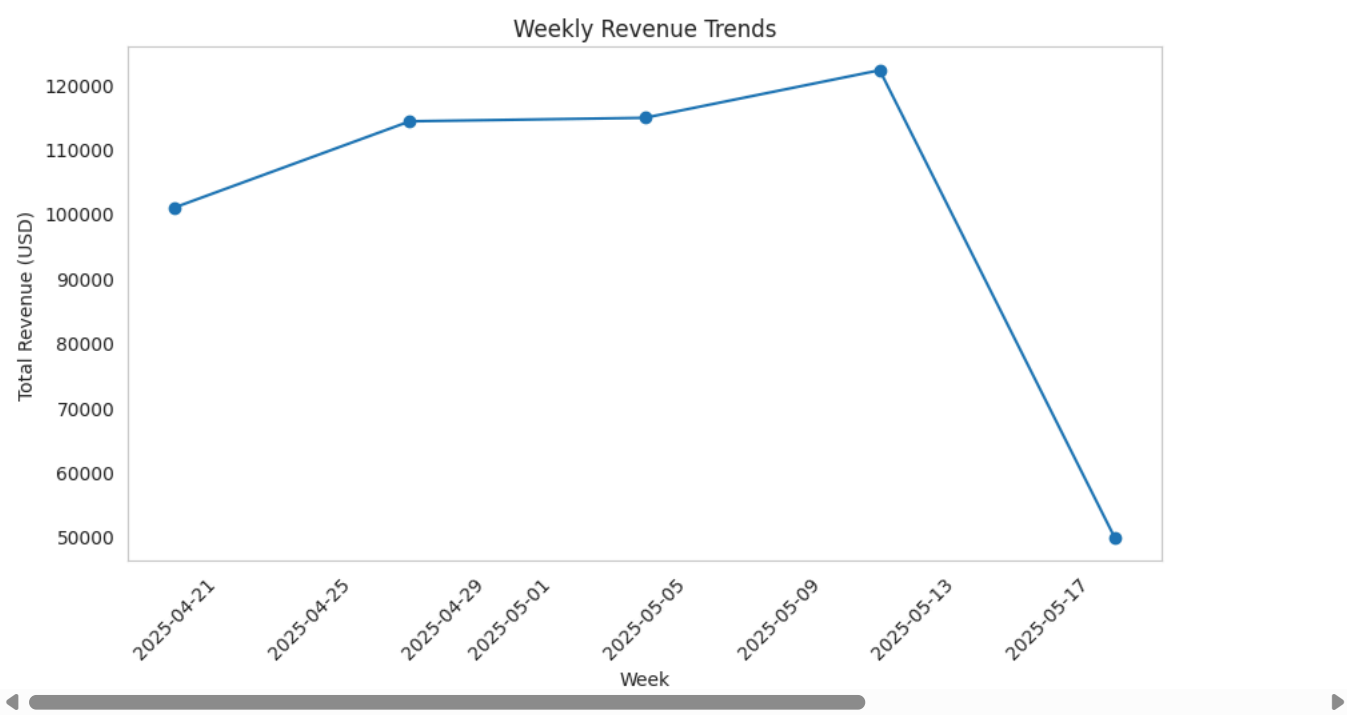
```
weekly_revenue
```

| | Week | Total_Revenue_USD |
|---|---|---|
| 0 | 2025-04-21 | 100885.25 |
| 1 | 2025-04-28 | 114212.05 |
| 2 | 2025-05-05 | 114751.66 |
| 3 | 2025-05-12 | 122117.40 |
| 4 | 2025-05-19 | 49808.30 |

Next steps: Generate code with weekly_revenue | View recommended plots | New interactive sheet

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
plt.plot(weekly_revenue['Week'], weekly_revenue['Total_Revenue_USD'], marker='o', linestyle='-')
plt.xlabel("Week")
plt.ylabel("Total Revenue (USD)")
plt.title("Weekly Revenue Trends")
plt.xticks(rotation=45)
plt.grid()
plt.show()
```




## Revenue Breakdown

### Device Type

```
revenue_by_device = df.groupby('Device_Type')['Total_Revenue_USD'].sum().reset_index()
```

```
print(revenue_by_device)
```

```
  Device_Type  Total_Revenue_USD
0     Console          168884.50
1      Mobile          165350.09
2          PC          167540.07
```

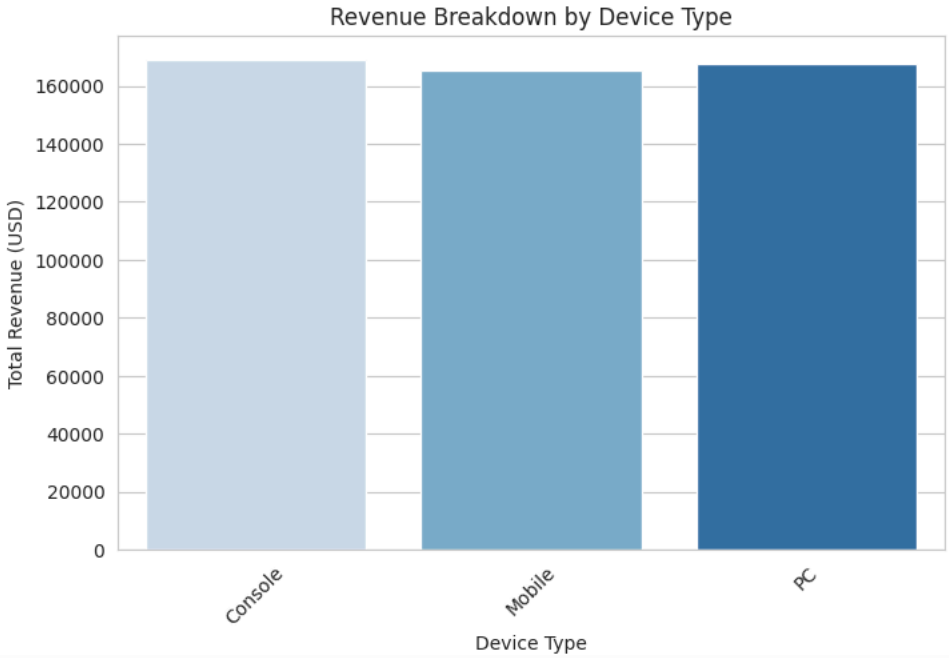
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_style("whitegrid")
plt.figure(figsize=(8,5))

# Creating the bar plot
sns.barplot(x="Device_Type", y="Total_Revenue_USD", data=revenue_by_device, hue="Device_Type", palette="Blues", legend=False)

# Customizing the labels and title
plt.xlabel("Device Type")
plt.ylabel("Total Revenue (USD)")
plt.title("Revenue Breakdown by Device Type")
plt.xticks(rotation=45)

plt.show()
```



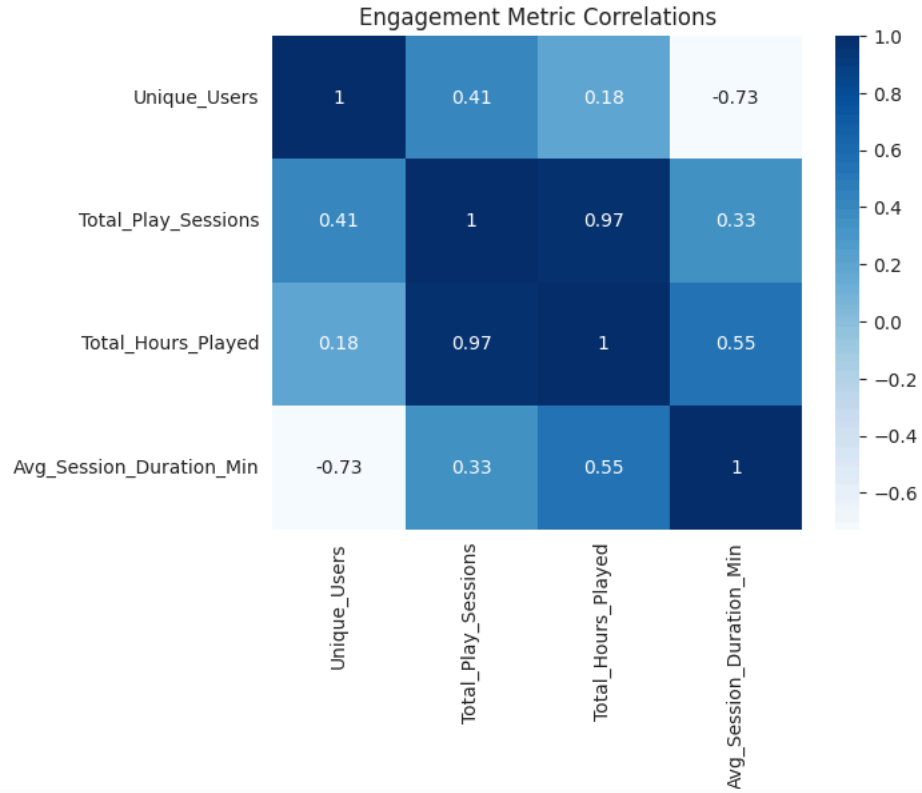

```
engagement_by_device = df.groupby('Device_Type').agg({
    'User_ID': 'nunique',
    'Total_Play_Sessions': 'mean',
    'Total_Hours_Played': 'mean',
    'Avg_Session_Duration_Min': 'mean'
}).reset_index().rename(columns={'User_ID': 'Unique_Users'})
```

```
print(engagement_by_device)
```

```
  Device_Type  Unique_Users  Total_Play_Sessions  Total_Hours_Played  \
0     Console          3395            20.015611          100.379090
1      Mobile          3301            19.936383           99.195341
2          PC          3304            20.029056          101.071120

   Avg_Session_Duration_Min
0                 29.925676
1                 30.006238
2                 30.176792
```

```
sns.heatmap(engagement_by_device.drop(columns=["Device_Type"]).corr(), annot=True, cmap="Blues")
plt.title("Engagement Metric Correlations")
plt.show()
```




## User Segment

```
revenue_by_tier = df.groupby('Subscription_Tier')['Total_Revenue_USD'].sum().reset_index()
```

```
revenue_by_tier
```

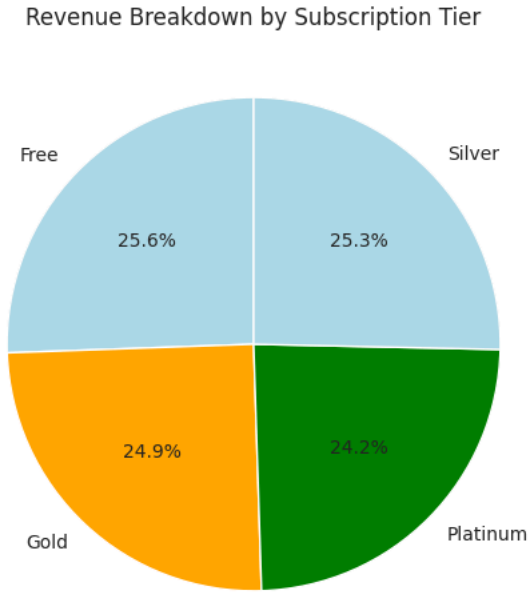
| | Subscription_Tier | Total_Revenue_USD |
|---|---|---|
| 0 | Free | 128255.21 |
| 1 | Gold | 125182.02 |
| 2 | Platinum | 121182.66 |
| 3 | Silver | 127154.77 |

Next steps: Generate code with revenue_by_tier | View recommended plots | New interactive sheet

```
import matplotlib.pyplot as plt

labels = revenue_by_tier["Subscription_Tier"]
sizes = revenue_by_tier["Total_Revenue_USD"]

plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, autopct="%1.1f%%", colors=["lightblue", "orange", "green"], startangle=90)
plt.title("Revenue Breakdown by Subscription Tier")
plt.show()
```




```
revenue_by_rank = df.groupby('Rank_Tier')['Total_Revenue_USD'].sum().reset_index()
```

```
revenue_by_rank
```

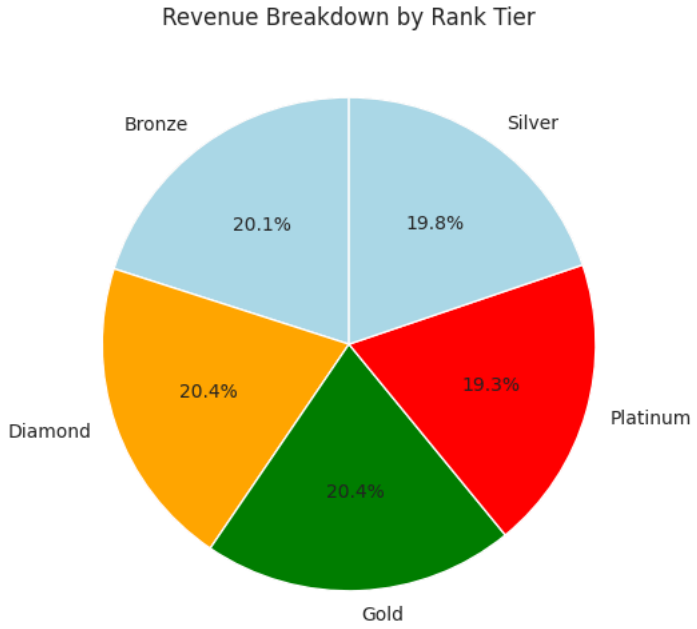
| | Rank_Tier | Total_Revenue_USD |
|---|---|---|
| 0 | Bronze | 100623.72 |
| 1 | Diamond | 102591.19 |
| 2 | Gold | 102490.25 |
| 3 | Platinum | 96764.17 |
| 4 | Silver | 99305.33 |

Next steps: Generate code with revenue_by_rank | View recommended plots | New interactive sheet

```
import matplotlib.pyplot as plt

labels = revenue_by_rank["Rank_Tier"]
sizes = revenue_by_rank["Total_Revenue_USD"]

plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, autopct="%1.1f%%", colors=["lightblue", "orange", "green", "red"], startangle=90)
plt.title("Revenue Breakdown by Rank Tier")
plt.show()
```




## Preferred Game Mode

```
revenue_by_mode = df.groupby('Preferred_Game_Mode')['Total_Revenue_USD'].sum().reset_index()
```

```
revenue_by_mode
```

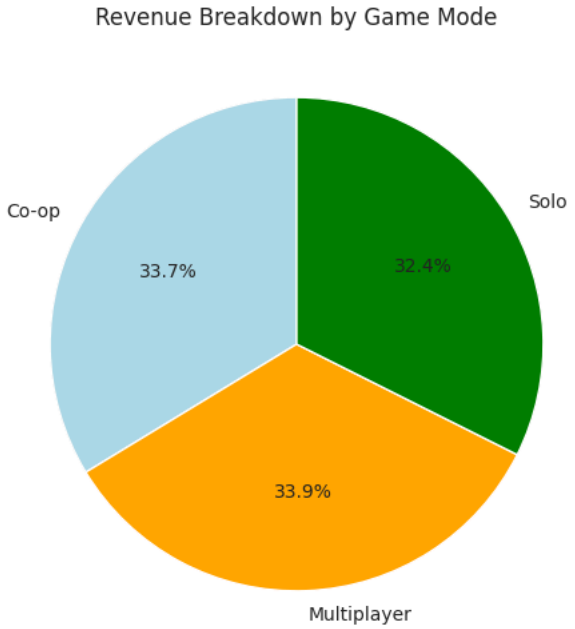
| | Preferred_Game_Mode | Total_Revenue_USD |
|---|---|---|
| 0 | Co-op | 169020.38 |
| 1 | Multiplayer | 170228.97 |
| 2 | Solo | 162525.31 |

Next steps: Generate code with revenue_by_mode | View recommended plots | New interactive sheet

```
import matplotlib.pyplot as plt

labels = revenue_by_mode["Preferred_Game_Mode"]
sizes = revenue_by_mode["Total_Revenue_USD"]

plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, autopct="%1.1f%%", colors=["lightblue", "orange", "green", "red"], startangle=90)
plt.title("Revenue Breakdown by Game Mode")
plt.show()
```




```
engagement_by_mode = df.groupby('Preferred_Game_Mode').agg({
    'User_ID': 'nunique',
    'Total_Play_Sessions': 'mean',
    'Total_Hours_Played': 'mean',
    'Avg_Session_Duration_Min': 'mean'
}).reset_index().rename(columns={'User_ID': 'Unique_Users'})
```

```
engagement_by_mode
```

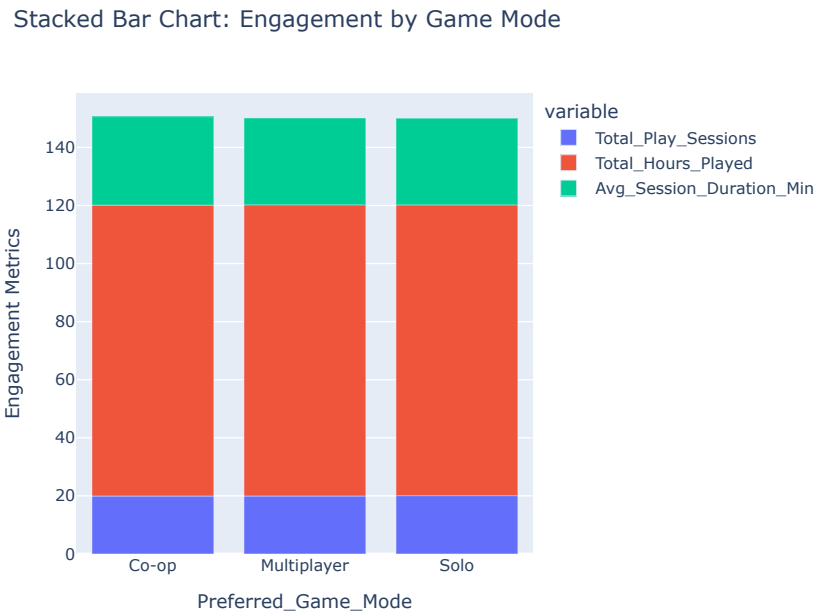
| | Preferred_Game_Mode | Unique_Users | Total_Play_Sessions | Total_Hours_Played | Avg_Session_Duration_Min |
|---|---|---|---|---|---|
| 0 | Co-op | 3351 | 19.943599 | 100.188729 | 30.441098 |
| 1 | Multiplayer | 3381 | 19.956818 | 100.300819 | 29.903392 |
| 2 | Solo | 3268 | 20.083843 | 100.159214 | 29.755474 |

Next steps: Generate code with engagement_by_mode | View recommended plots | New interactive sheet

```
import plotly.express as px

# Create stacked bar chart
fig = px.bar(
    engagement_by_mode,
    x="Preferred_Game_Mode",
    y=["Total_Play_Sessions", "Total_Hours_Played", "Avg_Session_Duration_Min"],
    labels={"value": "Engagement Metrics"},
    title="Stacked Bar Chart: Engagement by Game Mode",
    barmode="stack"
)

fig.show()
```




## Key Behavioral Metrics

```
df['Estimated_Active_Days'] = df['Total_Hours_Played'] / (df['Avg_Session_Duration_Min'] / 60)
```

```
df['Estimated_Active_Days'].head()
```

| | Estimated_Active_Days |
|---|---|
| 0 | 398.719723 |
| 1 | 237.971104 |
| 2 | 174.013921 |
| 3 | 167.983075 |
| 4 | 295.065230 |

dtype: float64

## Usage Frequency

```
df['Days_Since_Signup'] = (df['Last_Login'] - df['Signup_Date']).dt.days
df['Sessions_per_Day'] = df['Total_Play_Sessions'] / df['Days_Since_Signup']
```

```
df['Days_Since_Signup'].head()
```

| | Days_Since_Signup |
|---|---|
| 0 | 155 |
| 1 | 431 |
| 2 | 562 |
| 3 | 588 |
| 4 | 258 |

dtype: int64

```
df['Sessions_per_Day'].head()
```

| | Sessions_per_Day |
|---|---|
| **0** | 0.135484 |
| **1** | 0.051044 |
| **2** | 0.021352 |
| **3** | 0.032313 |
| **4** | 0.069767 |

dtype: float64

## Session Consistency

```
df['Play_Intensity'] = df['Total_Hours_Played'] / df['Total_Play_Sessions']
```

```
df['Play_Intensity']
```

| | Play_Intensity |
|---|---|
| **0** | 1.829048 |
| **1** | 5.864545 |
| **2** | 5.208333 |
| **3** | 4.178947 |
| **4** | 4.816667 |
| **...** | ... |
| **9995** | 6.357273 |
| **9996** | 3.662400 |
| **9997** | 4.955357 |
| **9998** | 4.252500 |
| **9999** | 6.481667 |

10000 rows × 1 columns

dtype: float64

## Churn Indicators

Long Gaps Since Signup

```
df['Days_Active'] = (df['Last_Login'] - df['Signup_Date']).dt.days
```

```
df['Days_Active'].tail()
```

| | Days_Active |
|---|---|
| **9995** | 712 |
| **9996** | 301 |
| **9997** | 403 |
| **9998** | 161 |
| **9999** | 603 |

dtype: int64

Low Session Volume

```
df[df['Total_Play_Sessions'] < 5]
```

| User_ID | Username | Email | Signup_Date | Country | Age | Gender | Device_Type | Game_Title | Total_Play_Sessions | ... | Rank_Tier | Achievem |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

0 rows × 28 columns

Short Average Session Duration

```
df[df['Avg_Session_Duration_Min'] < 5].head()
```

| | User_ID | Username | Email | Signup_Date | Country | Age | Gender | Device_Type | Game_Title | Total_Play_Se |
|---|---|---|---|---|---|---|---|---|---|---|
| **28** | b0a921b2-1156-4567-be03-564e1faf54ef | thomas50 | sheliachambers@hotmail.com | 2024-03-16 | Afghanistan | 20 | Male | Console | SpeedRun | |
| **38** | 134189bd-335e-46d6-ba03-6fa8e6b42ff2 | tracy41 | robert00@yahoo.com | 2024-10-10 | Namibia | 37 | Other | PC | BattleZone | |
| **114** | 7806e6fe-187c-4b18-980e-e3d7b72467a0 | michelle43 | johnny91@woods.com | 2025-03-22 | Egypt | 32 | Female | PC | MysticWar | |
| **130** | bc431b43-0acb-4281-b03d-5bfc8b75ae32 | james11 | martha30@jones.org | 2024-09-18 | Tuvalu | 39 | Other | PC | QuestRaid | |
| **305** | c9fde89a-db7c-49a9-a73b-33c2ce15d057 | grayemily | powelljames@adams.biz | 2023-07-14 | Guadeloupe | 44 | Female | Console | MysticWar | |

5 rows × 32 columns

## Recent Inactivity

```
df['Days_Since_Last_Login'] = (pd.Timestamp.today() - df['Last_Login']).dt.days
```

```
df['Days_Since_Last_Login']
```

| | Days_Since_Last_Login |
|---|---|
| **0** | 6 |
| **1** | 13 |
| **2** | 22 |
| **3** | 17 |
| **4** | 31 |
| **...** | ... |
| **9995** | 12 |
| **9996** | 26 |
| **9997** | 33 |
| **9998** | 6 |
| **9999** | 23 |

10000 rows × 1 columns

**dtype:** int64

## High-Value User Characteristics

```
high_value = df[df['Total_Revenue_USD'] > df['Total_Revenue_USD'].quantile(0.90)]
```

```
high_value.head()
```

| | User_ID | Username | Email | Signup_Date | Country | Age | Gender | Device_Type | Game_Title | Total_Play_Se |
|---|---|---|---|---|---|---|---|---|---|---|
| **3** | 47fcbe87-a1c1-40c3-b450-1b5692f61538 | christopher90 | ttaylor@gmail.com | 2023-09-28 | Belarus | 23 | Other | PC | QuestRaid | |
| **41** | 525986ce-3f37-47ed-95eb-5dadff517593 | michaelnicholson | stonecarl@ponce-russell.com | 2023-08-31 | Gambia | 16 | Other | PC | SpeedRun | |
| **45** | 51a4c311-58e8-4aa9-94b8-9046e74b3dd7 | angela96 | schmidtjoshua@davis.com | 2023-06-23 | South Georgia and the South Sandwich Islands | 13 | Female | Console | MysticWar | |
| **53** | f0d9a6c7-e478-4c52-8ec7-8238cfc949d7 | cherylsalinas | sextondawn@whitaker.net | 2023-07-26 | Montserrat | 34 | Other | Mobile | SpeedRun | |
| **67** | 9d829379-19ec-46bc-944e-d6f557799011 | timothy15 | anthony30@smith-kramer.com | 2024-05-06 | Aruba | 45 | Male | Console | MysticWar | |

5 rows × 20 columns