


# MATIKS GAMING ANALYSIS



SIGN IN

PLAY AS GUEST

POLICY

MATIKS

MAKING MATH A SPORT

Matiks is a community for mathletes, turning math into an accessible exciting sport for everyone.


LET'S PLAY ...→

+

-

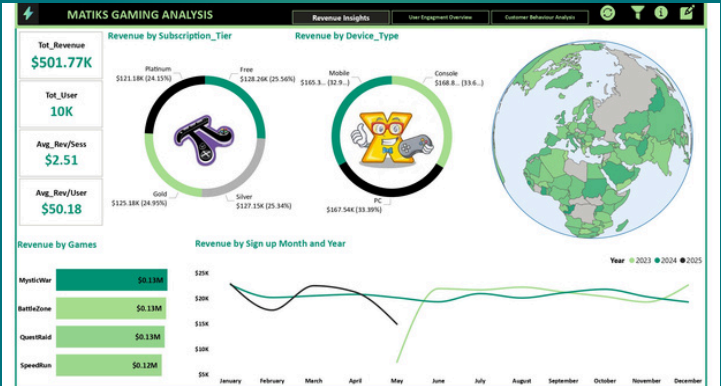
x

=



HI!

YOU'RE FAST. BUT HOW FAST? LET'S FIND OUT.



MATIKS\_GAMING\_ANALYSIS

Installing Necessary Modules

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

data = pd.read_csv('Matiks - Data Analyst Data - Sheet1.csv')
```

DATA\_PREVIEW

```
data.shape
```

```
(10000, 20)
```

```
data.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  object
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  object
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: float64(3), int64(4), object(13)
memory usage: 1.5+ MB
```

```
data.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	15-Dec-2024	Austria	22	Other	Mobile	MysticWar	
1	23c48d4f-f5d0-4ff4-ba0f-2007441b9b57	riverachristian	masonmelissa@hotmail.com	07-Mar-2024	Gabon	22	Other	PC	QuestRaid	
2	cf8d530c-c137-4346-a78b-e76e36d45e2a	brownchris	mnichols@mcmillan.net	19-Oct-2023	Ireland	36	Female	PC	QuestRaid	
3	47fcbe87-a1c1-40c3-b450-1b5692f61538	christopher90	ttaylor@gmail.com	28-Sep-2023	Belarus	23	Other	PC	QuestRaid	
4	0b620a32-9e77-4b4a-9931-f0b654bef095	vfreeman	amanda80@gmail.com	09-Aug-2024	Slovenia	26	Other	PC	QuestRaid	

DATA TYPE CONVERSION

```
data['Signup_Date'] = pd.to_datetime(data['Signup_Date'], format='%d-%b-%Y')
data['Last_Login'] = pd.to_datetime(data['Last_Login'], format='%d-%b-%Y')
```

```
obj_cols = data.select_dtypes(include='object').columns
data[obj_cols] = data[obj_cols].astype('string')
```

data.dtypes



	0
User_ID	string[python]
Username	string[python]
Email	string[python]
Signup_Date	datetime64[ns]
Country	string[python]
Age	int64
Gender	string[python]
Device_Type	string[python]
Game_Title	string[python]
Total_Play_Sessions	int64
Avg_Session_Duration_Min	float64
Total_Hours_Played	float64
In_Game_Purchases_Count	int64
Total_Revenue_USD	float64
Last_Login	datetime64[ns]
Subscription_Tier	string[python]
Referral_Source	string[python]
Preferred_Game_Mode	string[python]
Rank_Tier	string[python]
Achievement_Score	int64

dtype: object

Double-click (or enter) to edit

Multiple\_Entry\_Breakdown :

```
data[data.duplicated()]
```



User_ID	Username	Email	Signup_Date	Country	Age	Gender	Device_Type	Game_Title	Total_Play_Sessions	Avg_Session_Duration_Mi
---------	----------	-------	-------------	---------	-----	--------	-------------	------------	---------------------	-------------------------

```
data['Username'].nunique(),data['Email'].nunique(),data['User_ID'].nunique()
```

(9431, 9945, 10000)

```
data[['Username','Email']].value_counts()
```



		count
Username	Email	
zshort	rachel50@gmail.com	1
zshepard	kristen01@poole-martinez.org	1
zscott	austinodom@green-kelly.org	1
	angelaclark@rodgers.info	1
zsandoval	norrissarah@robinson.net	1
...	...	...
aaron36	nancy43@gmail.com	1
aaron12	janicecarlson@smith.info	1
aandrade	riverathomas@yahoo.com	1
aanderson	rebeccaharris@hotmail.com	1
	nataliebarr@gonzales-delgado.com	1

10000 rows × 1 columns

dtype: int64

```
data[['User_ID', 'Username']].value_counts()
```



		count
User_ID	Username	
ffa88a1e-9b7c-421f-b6bc-49c4f66fc9f4	mwilliams	1
ffa81832-9e60-4047-95e8-3fbbf0474a20	jeffrey57	1
ff9d871a-461b-4fca-8aa6-4d22421850ef	baileyjennifer	1
ff9a3972-f2e1-4b58-bb44-7c057d4a1823	amandajames	1
ff886b26-0181-425d-841c-f47c52961f2c	martin99	1
...	...	...
0018bf27-0fb0-482d-a2ed-647938e36d11	johnbarnes	1
0016b26e-c453-474f-bde4-3082018b608f	allentanya	1
0012ade9-4b67-46b8-b5a9-c1271550aa63	qmartin	1
00126f0c-8304-48c5-ad45-de9ce57f4dc9	ihughes	1
00079902-13c1-4102-9288-a404ecc1133e	rachelsantos	1

10000 rows × 1 columns

dtype: int64

```
# Group by Username and count unique emails
multi_email_users = data.groupby('Username')['Email'].nunique()
multi_email_users = multi_email_users[multi_email_users > 1].reset_index()
multi_email_users['Username'].count()
```



```
np.int64(478)
```

```
# Group by Email and count unique usernames
multi_username_emails = data.groupby('Email')['Username'].nunique()
multi_username_emails = multi_username_emails[multi_username_emails > 1].reset_index()
multi_username_emails['Email'].count()
```



```
np.int64(55)
```

## ✓ Feature\_Engineering

```
## USER_LIFESPAN IN DAYS
```

```
data['Retention_Days'] = (data['Last_Login'] - data['Signup_Date']).dt.days
data.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	

5 rows × 21 columns

```
data['Retention_Days'].describe()
```

	Retention_Days
count	10000.000000
mean	348.648100
std	211.711214
min	-29.000000
25%	165.000000
50%	348.000000
75%	529.000000
max	729.000000
dtype:	float64

```
# Negative Retention_Days --Last_Login happened before their Signup_Date,
# which is illogical in real-world behavior

neg_retention = data[data['Retention_Days'] < 0]
print(neg_retention[['User_ID','Username','Email','Signup_Date','Last_Login','Retention_Days']])
```

	User_ID	Username \	Email	Signup_Date	Last_Login	Retention_Days
211	6827e3e9-8406-44d6-88c9-dccc678d7120	kimberlyconway	kathleensalas@johnson-bowen.org	2025-05-09	2025-04-24	-15
262	17dff8af-2102-460f-a95f-b109992644ae	justin00	jacobslarry@hotmail.com	2025-05-04	2025-05-02	-2
277	839385d8-174f-4fe5-88f5-3c44b0cba598	gardnerdouglas	teresaward@yahoo.com	2025-05-13	2025-04-22	-21
290	ee7ed4e3-1fa4-4482-bae5-8431c9db5a66	nsmith	wolfdiane@yahoo.com	2025-05-13	2025-05-05	-8
325	783b1e0f-aa50-44f3-969b-02c9587d18a0	owright	joshuawilliams@yahoo.com	2025-04-27	2025-04-23	-4
...	...	...	...	...	...	...
9780	6de1c318-23db-48e7-9d21-6dc4b036883e	haynespatrick	mclaughlinjodi@webb.biz	2025-05-14	2025-05-03	-11
9838	825fa983-264f-4e2e-90f4-b82f4cb8cf6b	suarezedward	thompsoncaitlin@thomas.com	2025-05-21	2025-05-10	-11
9848	061337aa-3188-4158-8dce-cfd51e16bca6	davidsdaniel	jmartinez@yahoo.com	2025-05-15	2025-04-24	-21
9855	e458c381-b03d-47a6-b304-5edc2461539e	ysloan	kara82@martin.net	2025-05-12	2025-05-01	-11
9902	a165d6aa-8b6f-4f1b-b48f-436db95320b5	qmyers	goodwinjames@lynch.info	2025-05-18	2025-05-14	-4

[206 rows x 6 columns]

```
neg_retention[['User_ID','Username','Email','Signup_Date','Last_Login','Retention_Days']].head()
```

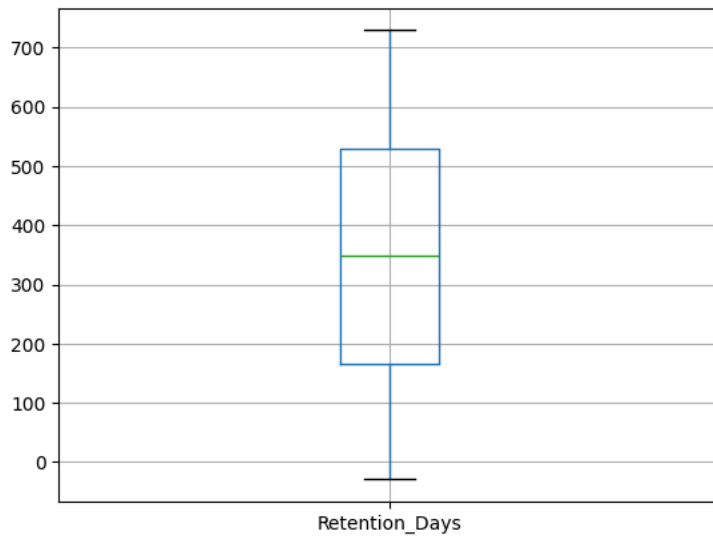


	User_ID	Username	Email	Signup_Date	Last_Login	Retention_Days
211	6827e3e9-8406-44d6-88c9-dccc678d7120	kimberlyconway	kathleensalas@johnson-bowen.org	2025-05-09	2025-04-24	-15
262	17dff8af-2102-460f-a95f-b109992644ae	justin00	jacobsLarry@hotmail.com	2025-05-04	2025-05-02	-2
277	839385d8-174f-4fe5-88f5-3c44b0cba598	gardnerdouglas	teresaward@yahoo.com	2025-05-13	2025-04-22	-21
290	ee7ed4e3-1fa4-4482-bae5-8431c9db5a66	nsmith	wolfdiane@yahoo.com	2025-05-13	2025-05-05	-8
325	783b1e0f-aa50-44f3-969b-02c9587d18a0	owright	joshuawilliams@yahoo.com	2025-04-27	2025-04-23	-4

```
data.boxplot(column=['Retention_Days'])
```



<Axes: >



```
## RETENTION DAYS SEGMENTATION BASED ON LIFESPAN IN DAYS
```

```
def retention_star_balanced(days):  
    if days < 0:  
        return '❌ Invalid'  
    elif days <= 90:  
        return '1★'  
    elif days <= 150:  
        return '2★'  
    elif days <= 250:  
        return '3★'  
    elif days <= 400:  
        return '4★'  
    else:  
        return '5★'
```

```
# Create the column in your dataset  
data['Retention_Segment'] = data['Retention_Days'].apply(retention_star_balanced)
```

```
data.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	browncris	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	

5 rows × 22 columns

```
data['Last_Login'].describe()
```

	Last_Login
count	10000
mean	2025-05-06 09:51:59.040000256
min	2025-04-22 00:00:00
25%	2025-04-29 00:00:00
50%	2025-05-06 00:00:00
75%	2025-05-14 00:00:00
max	2025-05-21 00:00:00

dtype: object

```
##CHURN SEGMENTATION

# Set "today" as one day after the max login
latest_login = data['Last_Login'].max()
reference_date = latest_login + pd.Timedelta(days=1)

# Calculate inactivity gap
data['Days_Since_Last_Login'] = (reference_date - data['Last_Login']).dt.days

# Classify churn status
def churn_status(days):
    if days <= 7:
        return 'Active'
    elif days <= 14:
        return 'At Risk'
    elif days <= 30:
        return 'Likely Churned'
    else:
        return 'Churned'

data['Churn_Status'] = data['Days_Since_Last_Login'].apply(churn_status)

data.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	browncris	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	

5 rows × 26 columns

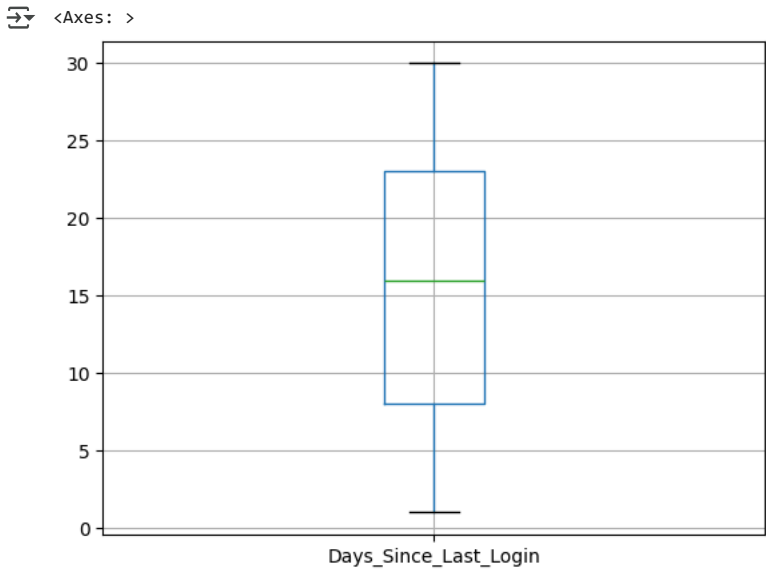
```
data['Days_Since_Last_Login'].describe()
```

↗

Days_Since_Last_Login	
count	10000.000000
mean	15.588900
std	8.653183
min	1.000000
25%	8.000000
50%	16.000000
75%	23.000000
max	30.000000

dtype: float64

```
data.boxplot(column=['Days_Since_Last_Login'])
```



```
data['Age'].describe()
```





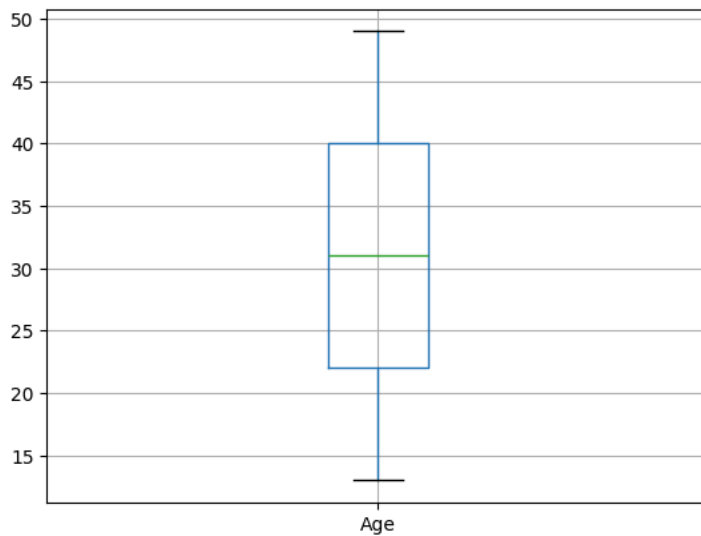
	Age
count	10000.000000
mean	31.063700
std	10.687547
min	13.000000
25%	22.000000
50%	31.000000
75%	40.000000
max	49.000000

dtype: float64

```
data.boxplot(column=['Age'])
```



<Axes: >



```
def age_bucket_custom(age):  
    if 11 <= age <= 15:  
        return 'Age 11-15'  
    elif 16 <= age <= 20:  
        return 'Age 16-20'  
    elif 21 <= age <= 25:  
        return 'Age 21-25'  
    elif 26 <= age <= 30:  
        return 'Age 26-30'  
    elif 31 <= age <= 35:  
        return 'Age 31-35'  
    elif 36 <= age <= 40:  
        return 'Age 36-40'  
    elif 41 <= age <= 45:  
        return 'Age 41-45'  
    elif 46 <= age <= 50:  
        return 'Age 46-50'  
    else:  
        return 'Unknown'
```

```
data['Age_Group_Bucket'] = data['Age'].apply(age_bucket_custom)
```

## ✓ CHECKLIST

# 1. Analyze Churn vs. Retention Segments

```
churn_vs_retention = data.groupby(['Retention_Segment', 'Churn_Status'])['User_ID'].count().reset_index()  
churn_vs_retention
```



	Retention_Segment	Churn_Status	User_ID
0	1★	Active	306
1	1★	At Risk	303
2	1★	Likely Churned	663
3	2★	Active	195
4	2★	At Risk	181
5	2★	Likely Churned	428
6	3★	Active	317
7	3★	At Risk	322
8	3★	Likely Churned	754
9	4★	Active	405
10	4★	At Risk	499
11	4★	Likely Churned	1133
12	5★	Active	1071
13	5★	At Risk	994
14	5★	Likely Churned	2223

```
# 2. Segment by Revenue
revenue_bins = pd.qcut(data['Total_Revenue_USD'], q=4, labels=['Low', 'Mid', 'High', 'Top'])
data['Revenue_Segment'] = revenue_bins

data.groupby(['Revenue_Segment', 'Churn_Status'])['User_ID'].count()
```



User\_ID



	Retention_Segment	Churn_Status	User_ID
0	1★	Active	306
1	1★	At Risk	303
2	1★	Likely Churned	663
3	2★	Active	195
4	2★	At Risk	181
5	2★	Likely Churned	428
6	3★	Active	317
7	3★	At Risk	322
8	3★	Likely Churned	754
9	4★	Active	405
10	4★	At Risk	499
11	4★	Likely Churned	1133
12	5★	Active	1071
13	5★	At Risk	994
14	5★	Likely Churned	2223

```
# 2. Segment by Revenue
revenue_bins = pd.qcut(data['Total_Revenue_USD'], q=4, labels=['Low', 'Mid', 'High', 'Top'])
data['Revenue_Segment'] = revenue_bins

data.groupby(['Revenue_Segment', 'Churn_Status'])['User_ID'].count()
```



		User_ID
Revenue_Segment	Churn_Status	
Low	Active	570
	At Risk	590
	Likely Churned	1341
Mid	Active	560
	At Risk	583
	Likely Churned	1357
High	Active	573
	At Risk	572
	Likely Churned	1354
Top	Active	600
	At Risk	588
	Likely Churned	1312

dtype: int64

## FILE CONVERSION

```
data.to_csv('matiks_gaming_dataset.csv', index=False)
```

Start coding or [generate](#) with AI.