databricks
https://databricks.com/

VOIZ FM COHORT ANALYSIS (Python)

Import notebook

VOIZ FM COHORT ANALYSIS

(01/2024 - 06/2024)

1. INTRODUCTION

This project examines the first purchase behavior of subscription users in the Voiz FM app from January 2024 to June 2024 using the cohort analysis method.

The project focuses on:

1. Evaluating retention rates by genders and platforms among the defined cohorts.
2. Analyzing the total revenue and the average revenue per user(ARPU) by genders and the total ARPU within these cohorts.

Before diving into the analysis, it is important to understand some key concepts used in this project.

Firstly, a subscription cohort is a group of users who started their subscription within the same time period, allowing businesses to track and compare their behavior and retention over time.

Secondly, retention rate is the percentage of subscribers who continue their subscriptions over a defined period. It reflects how well the business maintains its customers by keeping them engaged and satisfied enough to renew their subscription, rather than canceling it.

Thirdly, average revenue per user (ARPU) is the average income generated from each active subscriber during a specific period. It helps measure the monetization efficiency by indicating how much revenue, on average, each user contributes, including both new and renewing subscribers.

;

2. DATA EXPLORATION

First, all essential libraries are imported for processing and visualising the data.

```
                                                    8

   # Importing "pandas" library for reading the dataset and working with it.
   import pandas as pd

   # Importing seaborn library for visualization of the dataset(heatmap)
   import seaborn as sns

   #Importing "matlotlib.pyplot" library for visualization of the dataset.
   import matplotlib.pyplot as plt

   # #Import dateutil.relativedelta library for calculating the difference between two dates
   from dateutil.relativedelta import relativedelta
```

The data are read by pandas and we can get initial overview.

```
   user = pd.read_csv('/Volumes/voiz_fm/default/voizfm/users.csv')
   plan_orders = pd.read_csv('/Volumes/voiz_fm/default/voizfm/plan_orders.csv')
```

▸ plan_orders: pandas.core.frame.DataFrame = [user_id: int64, orderable_type: object ... 4 more fields]
▸ user: pandas.core.frame.DataFrame = [id: int64, birthday: object ... 2 more fields]

We check the general information of the data.

```
   user.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21351 entries, 0 to 21350
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   id          21351 non-null  int64
 1   birthday    11708 non-null  object
 2   gender      12228 non-null  float64
 3   created_at  21351 non-null  object
```

```
dtypes: float64(1), int64(1), object(2)
memory usage: 667.3+ KB
```

The user table has 4 different fields and 21351 records. birthday and gender fields have missing values because the non-null counts are less than the number of entries.

```
plan_orders.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 127946 entries, 0 to 127945
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype 
---  ------           --------------   ----- 
 0   user_id          127946 non-null  int64 
 1   orderable_type   127946 non-null  object
 2   orderable_id     127946 non-null  int64 
 3   amount           127946 non-null  int64 
 4   payment_gateway  127946 non-null  object
 5   created_at       127946 non-null  object
dtypes: int64(3), object(3)
memory usage: 5.9+ MB
```

The plan orders table has 5 different fields and 127946 records. There is no missing value in these fields because the non-null counts are equal to the number of entries.

We check the first few records in each table.

```
user.head()
```

	id	birthday	gender	created_at
0	162	NaN	NaN	2019-01-15 22:44:33.677058
1	316	NaN	0.0	2019-01-22 17:01:54.747598
2	1723	NaN	NaN	2019-03-16 18:46:21.233027
3	1997	NaN	1.0	2019-03-22 15:33:30.189282
4	2179	1999-01-01	0.0	2019-03-25 23:53:59.099162

The user table contains some missing values in the birthday and gender fields due to input errors caused by users not fully completing the survey form. We will add an age field for an ease in inspection instead of birthday field.

```
user['age'] = 2025 - pd.to_datetime(user['birthday'], errors = 'coerce').dt.year
print(user)
```

```
            id    birthday  gender                 created_at   age
0          162         NaN     NaN  2019-01-15 22:44:33.677058   NaN
1          316         NaN     0.0  2019-01-22 17:01:54.747598   NaN
2         1723         NaN     NaN  2019-03-16 18:46:21.233027   NaN
3         1997         NaN     1.0  2019-03-22 15:33:30.189282   NaN
4         2179  1999-01-01     0.0  2019-03-25 23:53:59.099162  26.0
...        ...         ...     ...                         ...   ...
21346  1826145  2024-10-16     NaN  2024-06-30 12:17:32.265631   1.0
21347  1826386         NaN     NaN  2024-06-30 15:12:26.046764   NaN
21348  1826479         NaN     NaN  2024-06-30 16:30:32.084677   NaN
21349  1837455  1995-01-01     1.0  2024-07-06 16:57:33.554793  30.0
21350  2029390         NaN     NaN   2024-09-23 01:25:28.62144   NaN

[21351 rows x 5 columns]
```

```
plan_orders.head()
```

	user_id	orderable_type	orderable_id	amount	payment_gateway	created_at
0	162	PlanPackage	57	99000	appstore	2024-01-19 15:07:18.95099
1	316	PlanPackage	1	49000	appstore	2019-11-06 02:54:57.938836
2	316	PlanPackage	1	49000	appstore	2019-12-05 17:24:50.592796
3	316	PlanPackage	1	49000	appstore	2019-10-04 11:12:08.858265
4	316	PlanPackage	25	9000	playstore	2020-08-19 05:06:47.577726

Most of the fields in the plan orders table are well-structured. We will perform further verification by examining specific columns.

First, we will check the distinct types of values present in the orderable_type field.

```
plan_orders['orderable_type'].unique()
```

```
array(['PlanPackage', 'PlanCoin', 'Playlist', 'ComboPackage'],
      dtype=object)
```

There are 4 types of packages.

- PlanPackage: User purchased subscription including one-month, three-month, and one-year packages.

- PlanCoin: User purchased Voiz FM's currency for products in the app.
- Playlist: User purchased the list of audios.
- ComboPackage: User purchased a combo of books.

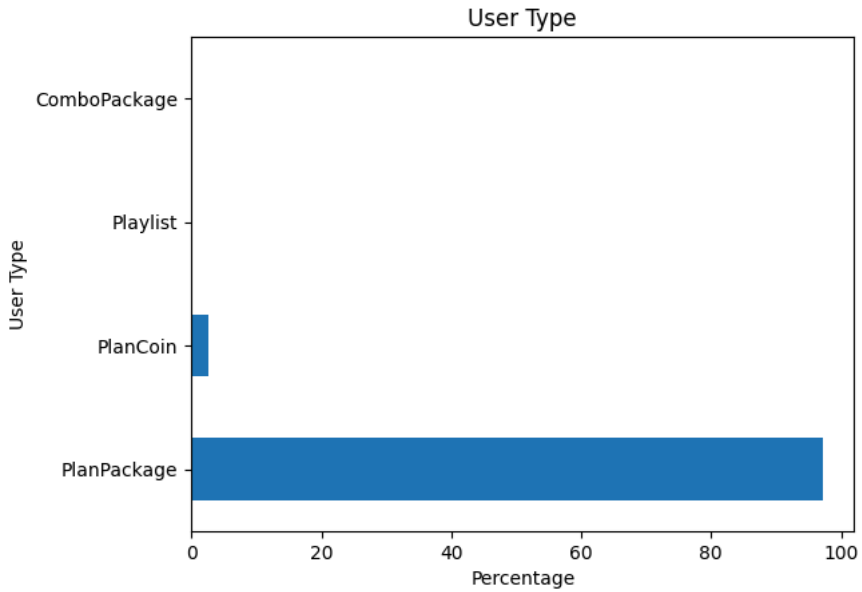
We check the contribution of each package type.

```
# User's type analysis
user_type = plan_orders['orderable_type'].value_counts(normalize=True)*100
print(user_type)
```

```
PlanPackage     97.148797
PlanCoin         2.640176
Playlist         0.147719
ComboPackage     0.063308
Name: orderable_type, dtype: float64
```

The package types are visualised.

```
# User's type chart
user_type.plot.barh()
plt.title('User Type')
plt.xlabel('Percentage')
plt.ylabel('User Type')
plt.show()
```



PlanPackage contributes over 95% in the total users. Therefore, we will focus on analysing PlanPackage.

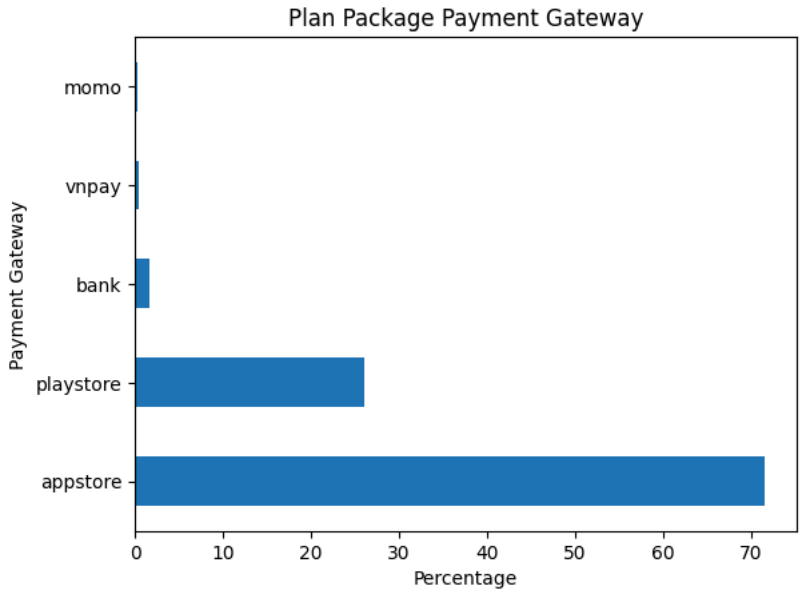
With PlanPackage, we will check its payment gateways.

```
# Plan package type
plan_package = plan_orders[plan_orders['orderable_type'] == 'PlanPackage']['payment_gateway'].value_counts(normalize=True)*100
print("Payment types contribution of plan packages:")
print(plan_package)
```

```
Payment types contribution of plan packages:
appstore     71.590854
playstore    26.104201
bank          1.626736
vnpay         0.393409
momo          0.284799
Name: payment_gateway, dtype: float64
```

Let's visualise the payment gateways for a better assessment.

```
plan_package.plot.barh()
plt.title('Plan Package Payment Gateway')
plt.xlabel('Percentage')
plt.ylabel('Payment Gateway')
plt.show()
```



The AppStore is the most popular payment gateway for PlanPackage, which aligns with the iPhone's strong market presence, holding around a 40% share of the mobile device market in Vietnam. Given the significant contributions from both the Playstore and AppStore, further analysis will focus primarily on these two payment gateways.

3. DATA PREPARTION

3.1. Check duplicates

Based on information from the business manager, some records were created due to users' network connection errors, resulting in multiple entries where all fields are identical except for the created_at timestamp, which differs by minutes. To identify and handle these duplicate records, the created_at field in the plan orders table will be converted to a date-only format for duplicate checking.

```
    # Change datetime to date for plan orders table and check duplicated values
    plan_orders.created_at = pd.to_datetime(plan_orders.created_at).dt.date

    print("Number of fully duplicated rows in plan order table:", plan_orders.duplicated().sum())
```

```
Number of fully duplicated rows in plan order table: 581
```

We remove all duplicated records in the table plan orders.

```
    # drop duplicated rows
    plan_orders = plan_orders.drop_duplicates()
```

▸ plan_orders: pandas.core.frame.DataFrame = [user_id: int64, orderable_type: object ... 4 more fields]

```
    user.created_at = pd.to_datetime(user.created_at).dt.date
    print("Number of fully duplicated rows in user table:", plan_orders.duplicated().sum())
```

```
Number of fully duplicated rows in user table: 0
```

There are no duplicated values in user table.

3.2. Check for missing values

As shown in data exploration, plan orders and plan subscriptions have no missing value in their field. User table is obtained through survey, so missing values are understandable.

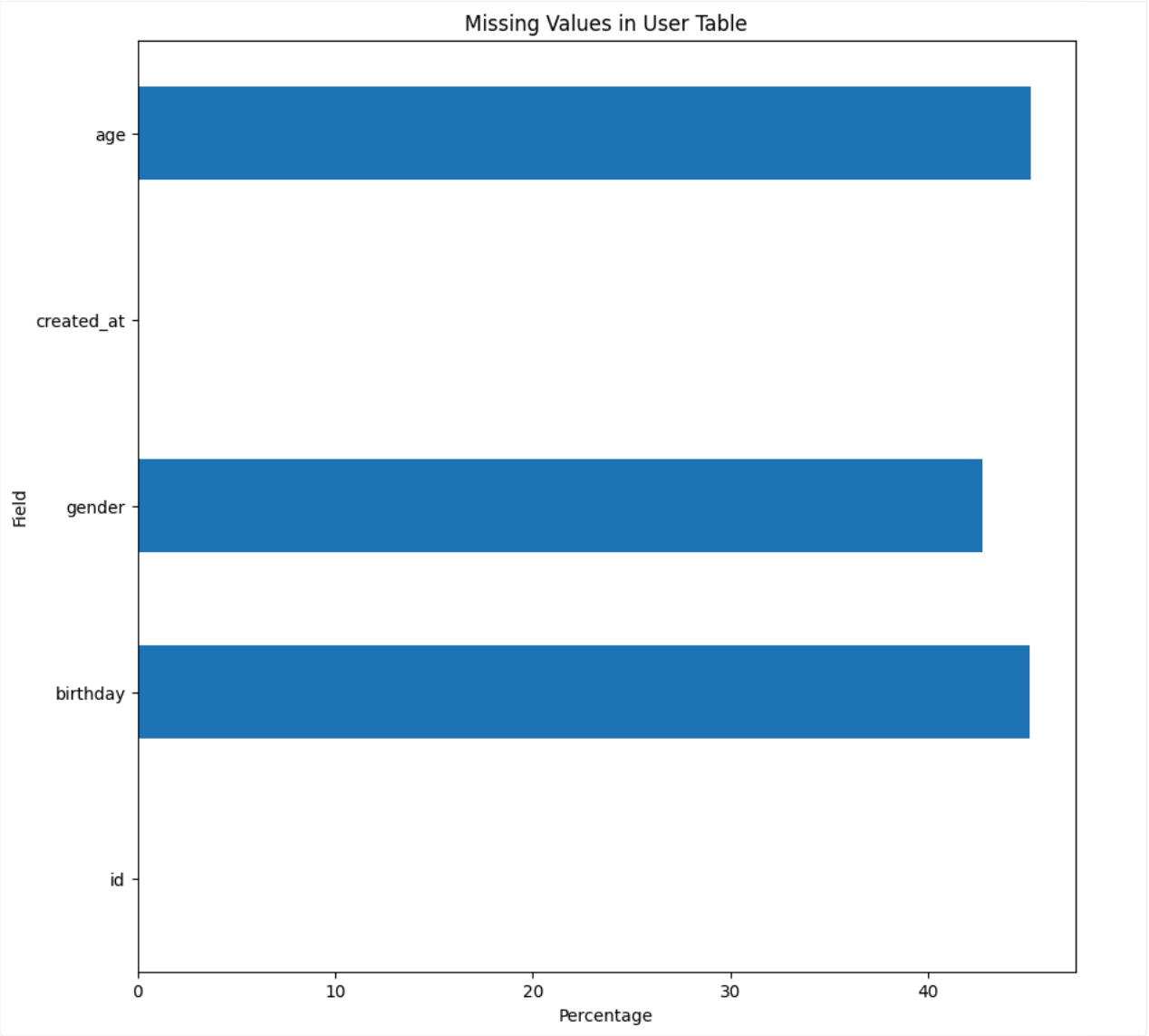
```
    # Total missing values for each column of user table
    user.isnull().sum()
```

```
id              0
birthday      9643
gender        9123
created_at       0
age           9646
dtype: int64
```

```
    # Total users
    total_users = len(user)

    # Number of missing values
    missing_values = user.isnull().sum()

    # Percentage of missing values
    missing_percentage = (missing_values / total_users) * 100

    print(missing_percentage)
```

```
id             0.000000
birthday      45.164161
gender        42.728678
created_at     0.000000
age           45.178212
dtype: float64
```

```
    # VIsualise missing values
    plt.figure(figsize=(10, 10))
    missing_percentage.plot.barh()
    plt.title('Missing Values in User Table')
    plt.xlabel('Percentage')
    plt.ylabel('Field')
    plt.show()
```



There is around 40% of missing values for all birthday, gender, and age.

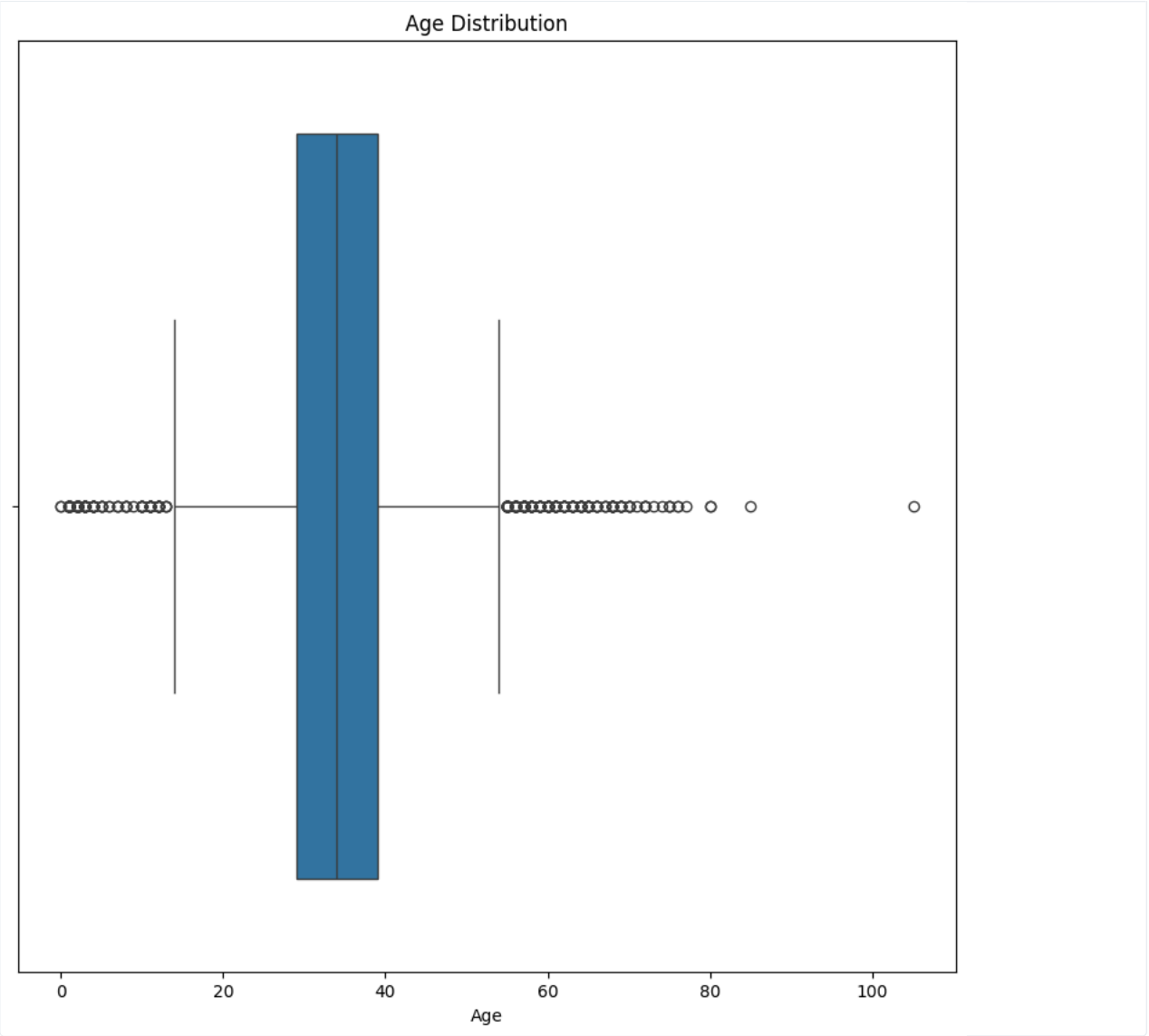
For gender, we will keep null value as a type of gender for further analysis because there are users using subscription without

personal information.

3.3. Check age distribution

For birthday, we will use age to analyse the boxplot to check outliers; however, for age, the analysis is limited to examining the distribution of users across age groups. Boxplots will still be processed to show high distribution range for future evaluations if needed.

```
# Boxplot to detect outliers
plt.figure(figsize=(10, 10))
sns.boxplot(x='age', data=user)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.show()
```



We will check age distribution by groups.

Age will be divided into different groups including:

1. Under 18

2. 18-22
3. 23-29
4. 30-34
5. 35-39
6. 40-44
7. Over 45

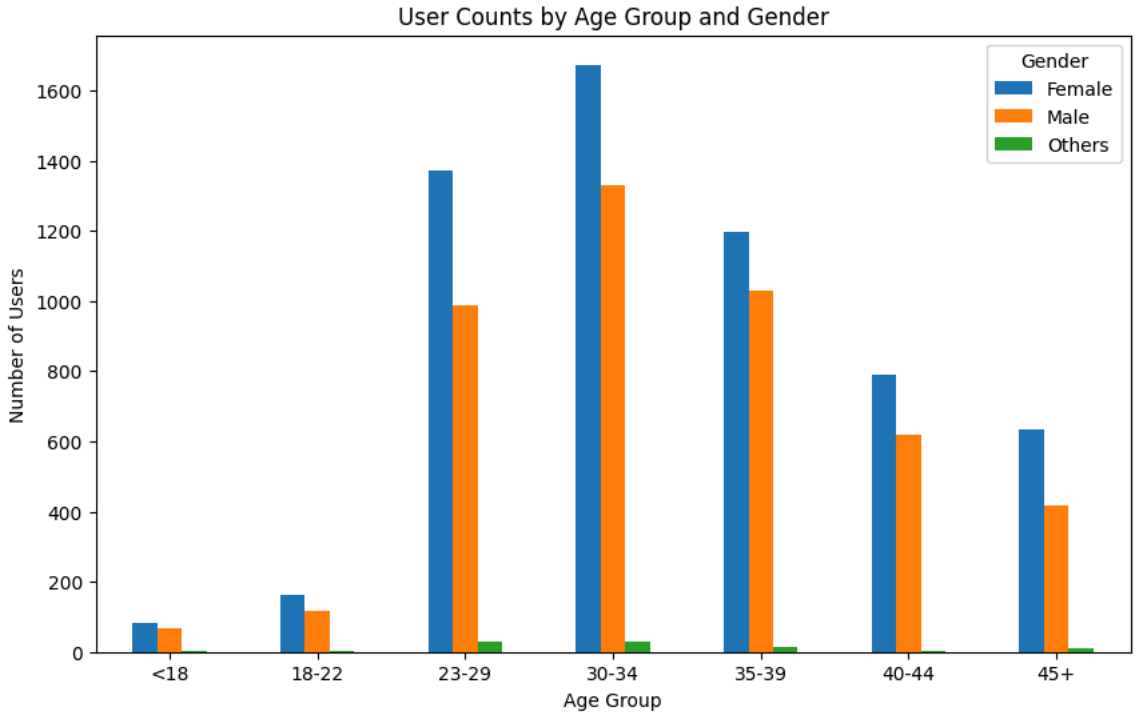
```
# Define age bins and labels optimally
bins = [0, 17, 22, 29, 34, 39, 44, float('inf')]
labels = ['<18', '18-22', '23-29', '30-34', '35-39', '40-44', '45+']

# Use pd.cut to assign age groups efficiently
user['age_group'] = pd.cut(user['age'], bins=bins, labels=labels, right=True, include_lowest=True)

# Group by age_group and gender, count unique user IDs
counts = user.groupby(['age_group', 'gender'])['id'].nunique().reset_index()

# Build dictionary with age groups as keys mapping to {gender: count}
gender_and_age = {}

for _, row in counts.iterrows():
    age_group = row['age_group']
    gender_key = str(row['gender'])
    count = row['id']

    if age_group not in gender_and_age:
        gender_and_age[age_group] = {}

    gender_and_age[age_group][gender_key] = count

print(gender_and_age)
```

▸ counts: pandas.core.frame.DataFrame = [age_group: category, gender: float64 ... 1 more field]

```
{'<18': {'0.0': 81, '1.0': 66, '2.0': 1}, '18-22': {'0.0': 163, '1.0': 115, '2.0': 3}, '23-29': {'0.0': 1370, '1.0': 986, '2.0': 29}, '30-34': {'0.0': 1673, '1.0': 1329, '2.0': 29}, '35-39': {'0.0': 1195, '1.0': 1030, '2.0': 16}, '40-44': {'0.0': 789, '1.0': 618, '2.0': 3}, '45+': {'0.0': 635, '1.0': 416, '2.0': 10}}
```

```
# Visualise the age group distribution
# Convert nested dict to DataFrame
gender_and_age_df = pd.DataFrame(gender_and_age).T.fillna(0)

# Rename gender columns for clarity
gender_and_age_df.columns = ['Female', 'Male', 'Others']

# Plot grouped bar chart
ga = gender_and_age_df.plot(kind='bar', figsize=(10,6))

ga.set_xlabel('Age Group')
ga.set_ylabel('Number of Users')
ga.set_title('User Counts by Age Group and Gender')
plt.xticks(rotation=0)
plt.legend(title='Gender')

plt.show()
```

▸ gender_and_age_df: pandas.core.frame.DataFrame = [Female: int64, Male: int64 ... 1 more field]



In all genders, users are concentrated in the 23-29 and 30-34 age groups. This can be explained by the fact that these groups tend to have a stable income and established media consumption habits. The chart does not display null values for gender. Gender null values are not shown in the main chart; however, in the exploratory data section, there are 12,228 non-null gender entries out of 21,351 total records. Null gender entries are classified separately as “non-identified,” while the “others” gender category is excluded from further analysis due to its negligible representation.

3.5. Function establishments

The data will be analysed by genders and platforms. To streamline the process and avoid redundancy, we will develop reusable functions to express the data efficiently.

Plan orders and user are merged to prepare for data processing.

```
plan_orders = plan_orders.merge(user, how = 'inner', left_on='user_id', right_on='id')
```

▸ plan_orders: pandas.core.frame.DataFrame = [user_id: int64, orderable_type: object ... 10 more fields]

A cohort consists of users who made their first subscription purchase within each defined period—in this analysis, each period is one month, spanning from January 2024 to June 2024. This function is used to define cohorts from January 2024 to June 2024.

```
# Initialize the dictionary outside
cohorts = {}

def define_cohort(plan_orders, cohort_start, cohort_end, cohort_month, cohort_year, cohorts_dict):
    plan_orders['created_at_x'] = pd.to_datetime(plan_orders['created_at_x'])

    existing_users = plan_orders[plan_orders['created_at_x'] < cohort_start].user_id.drop_duplicates()
    new_users = plan_orders[~plan_orders.user_id.isin(existing_users)]

    cohort_df = new_users[
        (new_users['created_at_x'] >= cohort_start) &
        (new_users['created_at_x'] <= cohort_end)
    ]

    # Use cohort_month and cohort_year as key, e.g. '2024-01'
    cohort_key = f"{cohort_year}-{cohort_month:02d}"

    # Store the cohort dataframe in the dictionary
    cohorts_dict[cohort_key] = cohort_df
```

Cohorts are processed by the function define_cohort and stored in a dictionary named cohorts, where each key corresponds to a specific cohort month. This structure allows easy access and analysis of user groups based on the month they made their first subscription purchase.

```
define_cohort(plan_orders, '2024-01-01', '2024-01-31', 1, 2024, cohorts)
define_cohort(plan_orders, '2024-02-01', '2024-02-28', 2, 2024, cohorts)
define_cohort(plan_orders, '2024-03-01', '2024-03-31', 3, 2024, cohorts)
define_cohort(plan_orders, '2024-04-01', '2024-04-30', 4, 2024, cohorts)
define_cohort(plan_orders, '2024-05-01', '2024-05-31', 5, 2024, cohorts)
define_cohort(plan_orders, '2024-06-01', '2024-06-30', 6, 2024, cohorts)
```

From the cohort details, we divide users into three smaller cohort types based on subscription duration for each specific month: one-month, three-month, and one-year cohorts.

```
def define_renewal_groups(cohort_df):
    one_month = cohort_df[
        ((cohort_df['amount'] >= 49000) & (cohort_df['amount'] <= 99000)) | (cohort_df['amount'] == 185000)
    ]
    three_month = cohort_df[
        ((cohort_df['amount'] >= 199000) & (cohort_df['amount'] <= 345000)) | (cohort_df['amount'] == 159000)
    ]
    one_year = cohort_df[
        cohort_df['amount'] >= 499000
    ]
    return {
        'one_month': one_month,
        'three_month': three_month,
        'one_year': one_year
    }
```

All three cohort types are stored in 1 dictionary cohort with month.

```
cohort_2024_01 = define_renewal_groups(cohorts['2024-01'])
cohort_2024_02 = define_renewal_groups(cohorts['2024-02'])
cohort_2024_03 = define_renewal_groups(cohorts['2024-03'])
cohort_2024_04 = define_renewal_groups(cohorts['2024-04'])
cohort_2024_05 = define_renewal_groups(cohorts['2024-05'])
cohort_2024_06 = define_renewal_groups(cohorts['2024-06'])
print(cohort_2024_06)
```

```
[275 rows x 12 columns]}
```

All cohorts will be assessed over a one-year period. For example, the January 2024 cohort will be evaluated for renewal activity from February 2024 through December 2024. This function tracks renewal activity across all months following the cohort start.

```
def get_renewals_by_cohort_month_gender(plan_orders, cohort_user_ids, cohort_year, cohort_month, months_to_check):
    plan_orders['created_at_x'] = pd.to_datetime(plan_orders['created_at_x'])
    results = {}

    start_date = pd.Timestamp(year=cohort_year, month=cohort_month, day=1)

    for i in range(1, months_to_check + 1):
        renewal_date = start_date + relativedelta(months=i)
        year = renewal_date.year
        month = renewal_date.month
        renewal_key = f"{year}-{str(month).zfill(2)}"

        filtered = plan_orders[(plan_orders['created_at_x'].dt.year == year) &
                               (plan_orders['created_at_x'].dt.month == month)]

         # Amount condition: include orders with amount in [49000, 99000] or exactly 185000
        filtered = filtered[
            (((filtered['amount'] >= 49000) & (filtered['amount'] <= 99000)) | (filtered['amount'] == 185000))
        ]

        # Filter cohort users
        renewal_users = filtered[filtered['user_id'].isin(cohort_user_ids)]


        # Separate by gender: male=1, female=0, non-identified=NaN
        male = renewal_users[renewal_users['gender'] == 1]
        female = renewal_users[renewal_users['gender'] == 0]
        others = renewal_users[renewal_users['gender'] == 2]
        non_identified = renewal_users[renewal_users['gender'].isna()]

        results[renewal_key] = {
            'male': male,
            'female': female,
            'others': others,
            'non_identified': non_identified
        }

    return results
```

All renewal activities are stored within a single dictionary, where the primary keys represent the renewal months, and each primary key contains nested keys corresponding to user genders. This structure allows organized tracking of renewals by month and gender.

```
renewals_2024_01_1_month = get_renewals_by_cohort_month_gender(plan_orders, cohort_2024_01['one_month'].user_id, 2024, 1 ,
11)
renewals_2024_02_1_month = get_renewals_by_cohort_month_gender(plan_orders, cohort_2024_02['one_month'].user_id, 2024, 2 ,
11)
renewals_2024_03_1_month = get_renewals_by_cohort_month_gender(plan_orders, cohort_2024_03['one_month'].user_id, 2024, 3 ,
11)
renewals_2024_04_1_month = get_renewals_by_cohort_month_gender(plan_orders, cohort_2024_04['one_month'].user_id, 2024, 4 ,
11)
renewals_2024_05_1_month = get_renewals_by_cohort_month_gender(plan_orders, cohort_2024_05['one_month'].user_id, 2024, 5 ,
11)
renewals_2024_06_1_month = get_renewals_by_cohort_month_gender(plan_orders, cohort_2024_06['one_month'].user_id, 2024, 6 ,
11)
print(renewals_2024_06_1_month)
```

```
47083    671160    PlanPackage            57  ...    2022-06-02   NaN       NaN
90104   1300457    PlanPackage            57  ...    2023-07-09   NaN       NaN
96727   1401442    PlanPackage            57  ...    2023-09-10   NaN       NaN
108301  1545989    PlanPackage            57  ...    2023-12-26   NaN       NaN
120856  1700263    PlanPackage            57  ...    2024-04-10   NaN       NaN
125389  1776632    PlanPackage            57  ...    2024-06-02   NaN       NaN
125643  1783293    PlanPackage            57  ...    2024-06-05   NaN       NaN
125705  1784897    PlanPackage            57  ...    2024-06-06  49.0       45+
125794  1787444    PlanPackage            57  ...    2024-06-08   NaN       NaN
125808  1787566    PlanPackage            57  ...    2024-06-08   NaN       NaN
125881  1788988    PlanPackage            57  ...    2024-06-09   NaN       NaN
125902  1789428    PlanPackage            57  ...    2024-06-09   NaN       NaN
125988  1791271    PlanPackage            57  ...    2024-06-10   NaN       NaN
126019  1791596    PlanPackage            57  ...    2024-06-10   NaN       NaN
126063  1791846    PlanPackage            57  ...    2024-06-10   NaN       NaN
126241  1794924    PlanPackage            57  ...    2024-06-12   NaN       NaN
126640  1804731    PlanPackage            57  ...    2024-06-18   NaN       NaN
126731  1807142    PlanPackage            57  ...    2024-06-20   NaN       NaN
126901  1812606    PlanPackage            57  ...    2024-06-23   NaN       NaN
127060  1816541    PlanPackage            57  ...    2024-06-25   NaN       NaN
127226  1821551    PlanPackage            57  ...    2024-06-28   NaN       NaN
```

This function tracks the platform data which is stored in a single dictionary.

```
def get_renewals_by_cohort_month_platform(plan_orders, cohort_user_ids, cohort_year, cohort_month, months_to_check):
    plan_orders['created_at_x'] = pd.to_datetime(plan_orders['created_at_x'])
    results = {}

    start_date = pd.Timestamp(year=cohort_year, month=cohort_month, day=1)

    for i in range(1, months_to_check + 1):
        renewal_date = start_date + relativedelta(months=i)
        year = renewal_date.year
        month = renewal_date.month
        renewal_key = f"{year}-{str(month).zfill(2)}"

        filtered = plan_orders[
            (plan_orders['created_at_x'].dt.year == year) &
            (plan_orders['created_at_x'].dt.month == month)
        ]

        filtered = filtered[
            (((filtered['amount'] >= 49000) & (filtered['amount'] <= 99000)) | (filtered['amount'] == 185000))
        ]

        renewal_users = filtered[filtered['user_id'].isin(cohort_user_ids)]

        # Separate by platform
        appstore = renewal_users[renewal_users['payment_gateway'] == 'appstore']
        playstore = renewal_users[renewal_users['payment_gateway'] == 'playstore']

        results[renewal_key] = {
            'appstore': appstore,
            'playstore': playstore
        }

    return results
```

All renewal activities are stored within a single dictionary, where the primary keys represent the renewal months, and each primary key contains nested keys corresponding to platform. This structure allows organized tracking of renewals by month and platform.

```
renewals_2024_01_1_month_platform = get_renewals_by_cohort_month_platform(plan_orders, cohort_2024_01['one_month'].user_id,
2024, 1 , 11)
renewals_2024_02_1_month_platform = get_renewals_by_cohort_month_platform(plan_orders, cohort_2024_02['one_month'].user_id,
2024, 2 , 11)
renewals_2024_03_1_month_platform = get_renewals_by_cohort_month_platform(plan_orders, cohort_2024_03['one_month'].user_id,
2024, 3 , 11)
renewals_2024_04_1_month_platform = get_renewals_by_cohort_month_platform(plan_orders, cohort_2024_04['one_month'].user_id,
2024, 4 , 11)
renewals_2024_05_1_month_platform = get_renewals_by_cohort_month_platform(plan_orders, cohort_2024_05['one_month'].user_id,
2024, 5 , 11)
renewals_2024_06_1_month_platform = get_renewals_by_cohort_month_platform(plan_orders, cohort_2024_06['one_month'].user_id,
2024, 6 , 11)
print(renewals_2024_06_1_month_platform)
```

```
127226  1821551    PlanPackage            57  ...    2024-06-28   NaN       NaN
127278  1823778    PlanPackage            57  ...    2024-06-29   NaN       NaN

[21 rows x 12 columns], 'playstore':         user_id orderable_type  orderable_id  ...  created_at_y   age age_group
92395   1338925    PlanPackage            57  ...    2023-07-30  33.0     30-34
96727   1401442    PlanPackage            57  ...    2023-09-10   NaN       NaN
125548  1780898    PlanPackage            57  ...    2024-06-04  24.0     23-29
125643  1783293    PlanPackage            57  ...    2024-06-05   NaN       NaN
125705  1784897    PlanPackage            57  ...    2024-06-06  49.0       45+
125794  1787444    PlanPackage            57  ...    2024-06-08   NaN       NaN
125902  1789428    PlanPackage            57  ...    2024-06-09   NaN       NaN
125988  1791271    PlanPackage            57  ...    2024-06-10   NaN       NaN
126019  1791596    PlanPackage            57  ...    2024-06-10   NaN       NaN
126043  1791779    PlanPackage            57  ...    2024-06-10  20.0     18-22
126063  1791846    PlanPackage            57  ...    2024-06-10   NaN       NaN
127060  1816541    PlanPackage            57  ...    2024-06-25   NaN       NaN
127315  1825769    PlanPackage            57  ...    2024-06-30   NaN       NaN
127328  1825792    PlanPackage            57  ...    2024-06-30   NaN       NaN
127342  1825946    PlanPackage            57  ...    2024-06-30   NaN       NaN

[15 rows x 12 columns]}}
```

These functions enable counting users for each renewal month efficiently, avoiding repetitive processing.

```
def count_unique_users(renewals_dict, month_key, gender):
    if month_key in renewals_dict and gender in renewals_dict[month_key]:
        return len(renewals_dict[month_key][gender]['user_id'].drop_duplicates())
    else:
        return 0
```

```
def count_unique_users_by_gateway(renewals_dict, month_key, gateway):
    if month_key in renewals_dict and gateway in renewals_dict[month_key]:
        return len(renewals_dict[month_key][gateway]['user_id'].drop_duplicates())
    else:
        return 0
```

Similar functions and calculations used for one-month subscription users will be applied to the three-month subscription users with relevant adjustments to account for differences in subscription duration.

```
# 3-month function
def get_renewals_by_cohort_quarterly(plan_orders, cohort_user_ids, cohort_year, cohort_month, periods):
    plan_orders['created_at_x'] = pd.to_datetime(plan_orders['created_at_x'])
    results_3 = {}
    start_date = pd.Timestamp(year=cohort_year, month=cohort_month, day=1)

    for i in range(1, periods + 1):
        renewal_date = start_date + relativedelta(months=3 * i)
        year = renewal_date.year
        month = renewal_date.month
        renewal_key = f"{year}-{str(month).zfill(2)}"

        filtered = plan_orders[
            (plan_orders['created_at_x'].dt.year == year) &
            (plan_orders['created_at_x'].dt.month == month)
        ]

        filtered = filtered[
            (((filtered['amount'] >= 199000) & (filtered['amount'] <= 345000)) | (filtered['amount'] == 159000))
        ]

        renewal_users = filtered[filtered['user_id'].isin(cohort_user_ids)]

        male = renewal_users[renewal_users['gender'] == 1]
        female = renewal_users[renewal_users['gender'] == 0]
        others = renewal_users[renewal_users['gender'] == 2]
        non_identified = renewal_users[renewal_users['gender'].isna()]

        results_3[renewal_key] = {
            'male': male,
            'female': female,
            'others': others,
            'non_identified': non_identified
        }

    return results_3
```

```
renewals_2024_01_3_month = get_renewals_by_cohort_quarterly(plan_orders, cohort_2024_01['three_month']['user_id'], 2024, 1 , 3)
renewals_2024_02_3_month = get_renewals_by_cohort_quarterly(plan_orders, cohort_2024_02['three_month']['user_id'], 2024, 2 , 3)
renewals_2024_03_3_month = get_renewals_by_cohort_quarterly(plan_orders, cohort_2024_03['three_month']['user_id'], 2024, 3 , 3)
renewals_2024_04_3_month = get_renewals_by_cohort_quarterly(plan_orders, cohort_2024_04['three_month']['user_id'], 2024, 4 , 3)
renewals_2024_05_3_month = get_renewals_by_cohort_quarterly(plan_orders, cohort_2024_05['three_month']['user_id'], 2024, 5 , 3)
renewals_2024_06_3_month = get_renewals_by_cohort_quarterly(plan_orders, cohort_2024_06['three_month']['user_id'], 2024, 6 , 3)
```

```
def get_renewals_by_cohort_quarterly_payment_gateway(plan_orders, cohort_user_ids, cohort_year, cohort_month, periods):
    plan_orders['created_at_x'] = pd.to_datetime(plan_orders['created_at_x'])
    results_3 = {}
    start_date = pd.Timestamp(year=cohort_year, month=cohort_month, day=1)

    for i in range(1, periods + 1):
        renewal_date = start_date + relativedelta(months=3 * i)
        year = renewal_date.year
        month = renewal_date.month
        renewal_key = f"{year}-{str(month).zfill(2)}"

        filtered = plan_orders[
            (plan_orders['created_at_x'].dt.year == year) &
            (plan_orders['created_at_x'].dt.month == month)
        ]

        filtered = filtered[
            (((filtered['amount'] >= 199000) & (filtered['amount'] <= 345000)) | (filtered['amount'] == 159000))
        ]

        renewal_users = filtered[filtered['user_id'].isin(cohort_user_ids)]

        appstore = renewal_users[renewal_users['payment_gateway'] == 'appstore']
        playstore = renewal_users[renewal_users['payment_gateway'] == 'playstore']

        results_3[renewal_key] = {
            'appstore': appstore,
            'playstore': playstore
        }

    return results_3
```

```
renewals_2024_01_3_month_payment_gateway = get_renewals_by_cohort_quarterly_payment_gateway(plan_orders,
cohort_2024_01['three_month']['user_id'], 2024, 1 , 3)
renewals_2024_02_3_month_payment_gateway = get_renewals_by_cohort_quarterly_payment_gateway(plan_orders,
cohort_2024_02['three_month']['user_id'], 2024, 2 , 3)
renewals_2024_03_3_month_payment_gateway = get_renewals_by_cohort_quarterly_payment_gateway(plan_orders,
cohort_2024_03['three_month']['user_id'], 2024, 3 , 3)
renewals_2024_04_3_month_payment_gateway = get_renewals_by_cohort_quarterly_payment_gateway(plan_orders,
cohort_2024_04['three_month']['user_id'], 2024, 4 , 3)
renewals_2024_05_3_month_payment_gateway = get_renewals_by_cohort_quarterly_payment_gateway(plan_orders,
cohort_2024_05['three_month']['user_id'], 2024, 5 , 3)
renewals_2024_06_3_month_payment_gateway = get_renewals_by_cohort_quarterly_payment_gateway(plan_orders,
cohort_2024_06['three_month']['user_id'], 2024, 6 , 3)
```

Similar functions and calculations will be applied for one-year subscription users with sufficient adjustments to account for differences in duration.

```
def get_renewals_by_cohort_yearly(plan_orders, cohort_user_ids, cohort_year, cohort_month, periods):
    plan_orders['created_at_x'] = pd.to_datetime(plan_orders['created_at_x'])
    results_yearly = {}
    start_date = pd.Timestamp(year=cohort_year, month=cohort_month, day=1)

    for i in range(1, periods + 1):
        # Renewal date every 12 months * i
        renewal_date = start_date + relativedelta(months=12 * i)
        year = renewal_date.year
        month = renewal_date.month
        renewal_key = f"{year}-{str(month).zfill(2)}"

        filtered = plan_orders[
            (plan_orders['created_at_x'].dt.year == year) &
            (plan_orders['created_at_x'].dt.month == month)
        ]

        filtered = filtered[
            (((filtered['amount'] >= 499000)))
        ]

        renewal_users = filtered[filtered['user_id'].isin(cohort_user_ids)]

        male = renewal_users[renewal_users['gender'] == 1]
        female = renewal_users[renewal_users['gender'] == 0]
        others = renewal_users[renewal_users['gender'] == 2]
        non_identified = renewal_users[renewal_users['gender'].isna()]

        results_yearly[renewal_key] = {
            'male': male,
            'female': female,
            'others': others,
            'non_identified': non_identified
        }

    return results_yearly
```

```
renewals_2024_01_1_year = get_renewals_by_cohort_yearly(plan_orders, cohort_2024_01['one_year'].user_id, 2024, 1, 1)
renewals_2024_02_1_year = get_renewals_by_cohort_yearly(plan_orders, cohort_2024_02['one_year'].user_id, 2024, 2, 1)
renewals_2024_03_1_year = get_renewals_by_cohort_yearly(plan_orders, cohort_2024_03['one_year'].user_id, 2024, 3, 1)
renewals_2024_04_1_year = get_renewals_by_cohort_yearly(plan_orders, cohort_2024_04['one_year'].user_id, 2024, 4, 1)
renewals_2024_05_1_year = get_renewals_by_cohort_yearly(plan_orders, cohort_2024_05['one_year'].user_id, 2024, 5, 1)
renewals_2024_06_1_year = get_renewals_by_cohort_yearly(plan_orders, cohort_2024_06['one_year'].user_id, 2024, 6, 1)
```

```
def get_renewals_by_cohort_yearly_payment_gateway(plan_orders, cohort_user_ids, cohort_year, cohort_month, periods):
    plan_orders['created_at_x'] = pd.to_datetime(plan_orders['created_at_x'])
    results_yearly = {}
    start_date = pd.Timestamp(year=cohort_year, month=cohort_month, day=1)

    for i in range(1, periods + 1):
        # Renewal date every 12 months * i
        renewal_date = start_date + relativedelta(months=12 * i)
        year = renewal_date.year
        month = renewal_date.month
        renewal_key = f"{year}-{str(month).zfill(2)}"

        filtered = plan_orders[
            (plan_orders['created_at_x'].dt.year == year) &
            (plan_orders['created_at_x'].dt.month == month)
        ]

        filtered = filtered[
            (filtered['amount'] >= 499000)
        ]

        renewal_users = filtered[filtered['user_id'].isin(cohort_user_ids)]

        appstore = renewal_users[renewal_users['payment_gateway'] == 'appstore']
        playstore = renewal_users[renewal_users['payment_gateway'] == 'playstore']

        results_yearly[renewal_key] = {
            'appstore': appstore,
            'playstore': playstore
        }

    return results_yearly
```

```
renewals_2024_01_1_year_payment_gateway = get_renewals_by_cohort_yearly_payment_gateway(plan_orders, cohort_2024_01['one_year'].
renewals_2024_02_1_year_payment_gateway = get_renewals_by_cohort_yearly_payment_gateway(plan_orders, cohort_2024_02['one_year'].
renewals_2024_03_1_year_payment_gateway = get_renewals_by_cohort_yearly_payment_gateway(plan_orders, cohort_2024_03['one_year'].
renewals_2024_04_1_year_payment_gateway = get_renewals_by_cohort_yearly_payment_gateway(plan_orders, cohort_2024_04['one_year'].
renewals_2024_05_1_year_payment_gateway = get_renewals_by_cohort_yearly_payment_gateway(plan_orders, cohort_2024_05['one_year'].
renewals_2024_06_1_year_payment_gateway = get_renewals_by_cohort_yearly_payment_gateway(plan_orders, cohort_2024_06['one_year'].
```

4. ANALYSIS

4.1. Retention rate analysis

Retention rate is calculated by:

$$\text{Retention Rate} = \frac{\text{Total number of users in renewal periods}}{\text{Total initial users at the start of the cohort}} \times 100$$

The total number of initial users and retention rate calculations will be performed separately for genders and platforms. They will be repeated for cohorts from February 2024 through June 2024.

4.1.1. Retention rates calculations by genders

Functions in Section 3.5 will be used to calculate retention rates, which represent lists of users who renew their subscriptions after a specific period. For example, retention rates for one-month subscription users will be calculated monthly and recorded accordingly.

4.1.1.1. One-month subscription users

For each cohort, the total number of initial users at the start will first be calculated by gender, followed by the calculation of retention rates for each gender within that cohort.

```
# One-month male users of Cohort January 2024
male_user_of_cohort_2024_01_one_month = len(cohort_2024_01['one_month'][cohort_2024_01['one_month']['gender'] == 1]
['user_id'].drop_duplicates())
# One-month female users of Cohort January 2024
female_user_of_cohort_2024_01_one_month = len(cohort_2024_01['one_month'][cohort_2024_01['one_month']['gender'] == 0]
['user_id'].drop_duplicates())
# One-month non-identified users of Cohort January 2024
non_identified_user_of_cohort_2024_01_one_month = len(cohort_2024_01['one_month'][cohort_2024_01['one_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
## Retention rate for 1-month subscription male users Cohort 2024-01
retention_male_2024_01 = [100,
                        round(count_unique_users(renewals_2024_01_1_month, '2024-02', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-03', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-04', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-05', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-06', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-07', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-08', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-09', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-10', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-11', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-12', 'male') /
male_user_of_cohort_2024_01_one_month * 100, 2)]
print('Retention rate of one-month subscription male users in Cohort 2024-01:')
print(retention_male_2024_01)

retention_female_2024_01 = [100,
                        round(count_unique_users(renewals_2024_01_1_month, '2024-02', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-03', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-04', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-05', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-06', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-07', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-08', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-09', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-10', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-11', 'female') /
female_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-12', 'female') /
```

```
female_user_of_cohort_2024_01_one_month * 100, 2)]
print('Retention rate of one-month subscription female users in Cohort 2024-01:')
print(retention_female_2024_01)

retention_non_identified_2024_01 = [100,
                        round(count_unique_users(renewals_2024_01_1_month, '2024-02', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100, 2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-03', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-04', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-05', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-06', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-07', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-08', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-09', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-10', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-11', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100,2),
                        round(count_unique_users(renewals_2024_01_1_month, '2024-12', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_month * 100, 2)]
print('Retention rate of one-month subscription non_identified users in Cohort 2024-01:')
print(retention_non_identified_2024_01)
```

```
Retention rate of one-month subscription male users in Cohort 2024-01:
[100, 55.98, 31.52, 23.37, 19.57, 17.93, 12.5, 10.33, 9.24, 9.24, 9.78, 7.07]
Retention rate of one-month subscription female users in Cohort 2024-01:
[100, 51.33, 37.0, 29.33, 21.67, 15.67, 11.67, 11.33, 10.33, 8.67, 9.67, 7.0]
Retention rate of one-month subscription non_identified users in Cohort 2024-01:
[100, 53.52, 41.85, 29.78, 23.34, 19.52, 15.49, 13.08, 12.27, 10.87, 9.46, 8.45]
```

```
# One-month male users of Cohort February 2024
male_user_of_cohort_2024_02_one_month = len(cohort_2024_02['one_month'][cohort_2024_02['one_month']['gender'] == 1]
['user_id'].drop_duplicates())
# One-month female users of Cohort February 2024
female_user_of_cohort_2024_02_one_month = len(cohort_2024_02['one_month'][cohort_2024_02['one_month']['gender'] == 0]
['user_id'].drop_duplicates())
# One-month non-identified users of Cohort February 2024
non_identified_user_of_cohort_2024_02_one_month = len(cohort_2024_02['one_month'][cohort_2024_02['one_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Male retention rates for Cohort 2024-02
retention_male_2024_02 = [100,
    round(count_unique_users(renewals_2024_02_1_month, '2024-03', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-04', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-05', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-06', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-07', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-08', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-09', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-10', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-11', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-12', 'male') / male_user_of_cohort_2024_02_one_month * 100,
2),
    round(count_unique_users(renewals_2024_02_1_month, '2025-01', 'male') / male_user_of_cohort_2024_02_one_month * 100, 2)
]
print('Retention rate of male users in Cohort 2024-02:')
print(retention_male_2024_02)


# Female retention rates for Cohort 2024-02
retention_female_2024_02 = [100,
    round(count_unique_users(renewals_2024_02_1_month, '2024-03', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-04', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-05', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-06', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-07', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-08', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-09', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-10', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-11', 'female') / female_user_of_cohort_2024_02_one_month *
100, 2),
    round(count_unique_users(renewals_2024_02_1_month, '2024-12', 'female') / female_user_of_cohort_2024_02_one_month *
```

```
    100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2025-01', 'female') / female_user_of_cohort_2024_02_one_month *
    100, 2)
    ]
    print('Retention rate of female users in Cohort 2024-02:')
    print(retention_female_2024_02)

    # Non-identified retention rates for Cohort 2024-02
    retention_non_identified_2024_02 = [100,
        round(count_unique_users(renewals_2024_02_1_month, '2024-03', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-04', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-05', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-06', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-07', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-08', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-09', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-10', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-11', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2024-12', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2),
        round(count_unique_users(renewals_2024_02_1_month, '2025-01', 'non_identified') /
    non_identified_user_of_cohort_2024_02_one_month * 100, 2)
    ]

    print('Retention rate of non_identified users in Cohort 2024-02:')
    print(retention_non_identified_2024_02)
```

```
Retention rate of male users in Cohort 2024-02:
[100, 53.46, 34.59, 29.56, 24.53, 13.84, 15.09, 12.58, 12.58, 10.06, 6.29, 6.29]
Retention rate of female users in Cohort 2024-02:
[100, 58.88, 34.58, 25.23, 18.22, 15.42, 14.02, 12.62, 10.28, 7.94, 7.01, 7.01]
Retention rate of non_identified users in Cohort 2024-02:
[100, 51.72, 36.03, 27.45, 21.81, 17.89, 14.71, 13.48, 11.76, 9.8, 8.58, 8.82]
```

```
# One-month male users of Cohort March 2024
male_user_of_cohort_2024_03_one_month = len(cohort_2024_03['one_month'][cohort_2024_03['one_month']['gender'] == 1]
['user_id'].drop_duplicates())
# One-month female users of Cohort March 2024
female_user_of_cohort_2024_03_one_month = len(cohort_2024_03['one_month'][cohort_2024_03['one_month']['gender'] == 0]
['user_id'].drop_duplicates())
# One-month non-identified users of Cohort March 2024
non_identified_user_of_cohort_2024_03_one_month = len(cohort_2024_03['one_month'][cohort_2024_03['one_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Male retention rates Cohort 2024-03
retention_male_2024_03 = [100,
    round(count_unique_users(renewals_2024_03_1_month, '2024-04', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-05', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-06', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-07', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-08', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-09', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-10', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-11', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-12', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2025-01', 'male') / male_user_of_cohort_2024_03_one_month * 100,
2),
    round(count_unique_users(renewals_2024_03_1_month, '2025-02', 'male') / male_user_of_cohort_2024_03_one_month * 100, 2)
]
print('Retention rate of male users in Cohort 2024-03:')
print(retention_male_2024_03)


# Female retention rates Cohort 2024-03
retention_female_2024_03 = [100,
    round(count_unique_users(renewals_2024_03_1_month, '2024-04', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-05', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-06', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-07', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-08', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-09', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-10', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-11', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2024-12', 'female') / female_user_of_cohort_2024_03_one_month *
100, 2),
    round(count_unique_users(renewals_2024_03_1_month, '2025-01', 'female') / female_user_of_cohort_2024_03_one_month *
```

```
    100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2025-02', 'female') / female_user_of_cohort_2024_03_one_month *
    100, 2)
    ]
    print('Retention rate of female users in Cohort 2024-03:')
    print(retention_female_2024_03)


    # Non-identified retention rates Cohort 2024-03
    retention_non_identified_2024_03 = [100,
        round(count_unique_users(renewals_2024_03_1_month, '2024-04', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-05', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-06', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-07', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-08', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-09', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-10', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-11', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2024-12', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2025-01', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users(renewals_2024_03_1_month, '2025-02', 'non_identified') /
    non_identified_user_of_cohort_2024_03_one_month * 100, 2)
    ]
    print('Retention rate of non_identified users in Cohort 2024-03:')
    print(retention_non_identified_2024_03)
```

```
Retention rate of male users in Cohort 2024-03:
[100, 55.77, 42.95, 33.33, 26.28, 21.79, 19.23, 16.03, 14.1, 11.54, 9.62, 7.05]
Retention rate of female users in Cohort 2024-03:
[100, 47.66, 36.92, 28.04, 21.96, 17.29, 12.15, 10.75, 8.88, 9.35, 7.94, 5.61]
Retention rate of non_identified users in Cohort 2024-03:
[100, 51.63, 37.09, 26.82, 19.55, 16.04, 12.28, 10.03, 9.77, 9.52, 7.52, 6.52]
```

```
# One-month male users of Cohort April 2024
male_user_of_cohort_2024_04_one_month = len(cohort_2024_04['one_month'][cohort_2024_04['one_month']['gender'] == 1]
['user_id'].drop_duplicates())
# One-month female users of Cohort April 2024
female_user_of_cohort_2024_04_one_month = len(cohort_2024_04['one_month'][cohort_2024_04['one_month']['gender'] == 0]
['user_id'].drop_duplicates())
# One-month non-identifed users of Cohort April 2024
non_identified_user_of_cohort_2024_04_one_month = len(cohort_2024_04['one_month'][cohort_2024_04['one_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Male retention rates Cohort 2024-04
retention_male_2024_04 = [100,
    round(count_unique_users(renewals_2024_04_1_month, '2024-05', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-06', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-07', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-08', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-09', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-10', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-11', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-12', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2025-01', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2025-02', 'male') / male_user_of_cohort_2024_04_one_month * 100,
2),
    round(count_unique_users(renewals_2024_04_1_month, '2025-03', 'male') / male_user_of_cohort_2024_04_one_month * 100, 2)
]
print('Retention rate of male users in Cohort 2024-04:')
print(retention_male_2024_04)


# Female retention rates Cohort 2024-04
retention_female_2024_04 = [100,
    round(count_unique_users(renewals_2024_04_1_month, '2024-05', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-06', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-07', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-08', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-09', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-10', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-11', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2024-12', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2025-01', 'female') / female_user_of_cohort_2024_04_one_month *
100, 2),
    round(count_unique_users(renewals_2024_04_1_month, '2025-02', 'female') / female_user_of_cohort_2024_04_one_month *
```

```
    100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2025-03', 'female') / female_user_of_cohort_2024_04_one_month *
    100, 2)
    ]
    print('Retention rate of female users in Cohort 2024-04:')
    print(retention_female_2024_04)


    # Non-identified retention rates Cohort 2024-04
    retention_non_identified_2024_04 = [100,
        round(count_unique_users(renewals_2024_04_1_month, '2024-05', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2024-06', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2024-07', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2024-08', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2024-09', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2024-10', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2024-11', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2024-12', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2025-01', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2025-02', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users(renewals_2024_04_1_month, '2025-03', 'non_identified') /
    non_identified_user_of_cohort_2024_04_one_month * 100, 2)
    ]
    print('Retention rate of non_identified users in Cohort 2024-04:')
    print(retention_non_identified_2024_04)
```

```
Retention rate of male users in Cohort 2024-04:
[100, 53.6, 37.6, 24.8, 22.4, 17.6, 16.0, 9.6, 9.6, 4.8, 5.6, 6.4]
Retention rate of female users in Cohort 2024-04:
[100, 50.28, 35.75, 26.82, 18.99, 15.64, 15.08, 12.29, 12.85, 11.73, 10.61, 9.5]
Retention rate of non_identified users in Cohort 2024-04:
[100, 50.15, 35.4, 27.14, 20.35, 16.52, 15.63, 12.39, 12.39, 10.03, 9.14, 7.67]
```

```
# One-month male users of Cohort May 2024
male_user_of_cohort_2024_05_one_month = len(cohort_2024_05['one_month'][cohort_2024_05['one_month']['gender'] == 1]
['user_id'].drop_duplicates())
# One-month female users of Cohort May 2024
female_user_of_cohort_2024_05_one_month = len(cohort_2024_05['one_month'][cohort_2024_05['one_month']['gender'] == 0]
['user_id'].drop_duplicates())
# One-month non-identifed users of Cohort May 2024
non_identified_user_of_cohort_2024_05_one_month = len(cohort_2024_05['one_month'][cohort_2024_05['one_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Male retention rates Cohort 2024-05
retention_male_2024_05 = [100,
    round(count_unique_users(renewals_2024_05_1_month, '2024-06', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-07', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-08', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-09', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-10', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-11', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-12', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2025-01', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2025-02', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2025-03', 'male') / male_user_of_cohort_2024_05_one_month * 100,
2),
    round(count_unique_users(renewals_2024_05_1_month, '2025-04', 'male') / male_user_of_cohort_2024_05_one_month * 100, 2)
]
print('Retention rate of male users in Cohort 2024-05:')
print(retention_male_2024_05)


# Female retention rates Cohort 2024-05
retention_female_2024_05 = [100,
    round(count_unique_users(renewals_2024_05_1_month, '2024-06', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-07', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-08', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-09', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-10', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-11', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2024-12', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2025-01', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2025-02', 'female') / female_user_of_cohort_2024_05_one_month *
100, 2),
    round(count_unique_users(renewals_2024_05_1_month, '2025-03', 'female') / female_user_of_cohort_2024_05_one_month *
```

```
    100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2025-04', 'female') / female_user_of_cohort_2024_05_one_month *
    100, 2)
    ]
    print('Retention rate of female users in Cohort 2024-05:')
    print(retention_female_2024_05)


    # Non-identified retention rates Cohort 2024-05
    retention_non_identified_2024_05 = [100,
        round(count_unique_users(renewals_2024_05_1_month, '2024-06', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2024-07', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2024-08', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2024-09', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2024-10', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2024-11', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2024-12', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2025-01', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2025-02', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2025-03', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users(renewals_2024_05_1_month, '2025-04', 'non_identified') /
    non_identified_user_of_cohort_2024_05_one_month * 100, 2)
    ]
    print('Retention rate of non_identified users in Cohort 2024-05:')
    print(retention_non_identified_2024_05)
```

```
Retention rate of male users in Cohort 2024-05:
[100, 55.24, 39.86, 31.47, 22.38, 18.18, 16.08, 14.69, 10.49, 9.09, 9.09, 7.69]
Retention rate of female users in Cohort 2024-05:
[100, 47.59, 34.22, 27.27, 19.79, 17.11, 12.83, 10.7, 10.7, 9.09, 7.49, 7.49]
Retention rate of non_identified users in Cohort 2024-05:
[100, 53.03, 35.45, 27.67, 22.19, 18.16, 15.85, 13.26, 11.53, 10.95, 9.51, 8.07]
```

```
# One-month male users of Cohort June 2024
male_user_of_cohort_2024_06_one_month = len(cohort_2024_06['one_month'][cohort_2024_06['one_month']['gender'] == 1]
['user_id'].drop_duplicates())
# One-month female users of Cohort May 2024
female_user_of_cohort_2024_06_one_month = len(cohort_2024_06['one_month'][cohort_2024_06['one_month']['gender'] == 0]
['user_id'].drop_duplicates())
# One-month non-identified users of Cohort May 2024
non_identified_user_of_cohort_2024_06_one_month = len(cohort_2024_06['one_month'][cohort_2024_06['one_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Male retention rates Cohort 2024-06
retention_male_2024_06 = [100,
    round(count_unique_users(renewals_2024_06_1_month, '2024-07', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-08', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-09', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-10', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-11', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-12', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-01', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-02', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-03', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-04', 'male') / male_user_of_cohort_2024_06_one_month * 100,
2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-05', 'male') / male_user_of_cohort_2024_06_one_month * 100, 2)
]
print('Retention rate of male users in Cohort 2024-06:')
print(retention_male_2024_06)


# Female retention rates Cohort 2024-06
retention_female_2024_06 = [100,
    round(count_unique_users(renewals_2024_06_1_month, '2024-07', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-08', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-09', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-10', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-11', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2024-12', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-01', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-02', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-03', 'female') / female_user_of_cohort_2024_06_one_month *
100, 2),
    round(count_unique_users(renewals_2024_06_1_month, '2025-04', 'female') / female_user_of_cohort_2024_06_one_month *
```

```
    100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2025-05', 'female') / female_user_of_cohort_2024_06_one_month *
    100, 2)
    ]
    print('Retention rate of female users in Cohort 2024-06:')
    print(retention_female_2024_06)


    # Non-identified retention rates Cohort 2024-06
    retention_non_identified_2024_06 = [100,
        round(count_unique_users(renewals_2024_06_1_month, '2024-07', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2024-08', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2024-09', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2024-10', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2024-11', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2024-12', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2025-01', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2025-02', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2025-03', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2025-04', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users(renewals_2024_06_1_month, '2025-05', 'non_identified') /
    non_identified_user_of_cohort_2024_06_one_month * 100, 2)
    ]
    print('Retention rate of non_identified users in Cohort 2024-06:')
    print(retention_non_identified_2024_06)
```

```
Retention rate of male users in Cohort 2024-06:
[100, 47.62, 35.24, 22.86, 17.14, 13.33, 10.48, 5.71, 6.67, 5.71, 5.71, 4.76]
Retention rate of female users in Cohort 2024-06:
[100, 53.99, 32.52, 25.15, 22.09, 16.56, 14.11, 10.43, 9.2, 9.82, 6.75, 3.68]
Retention rate of non_identified users in Cohort 2024-06:
[100, 60.22, 42.01, 32.71, 24.91, 21.19, 19.7, 16.73, 14.5, 11.15, 10.41, 9.29]
```

```
# One-month male subscription users
retentions_male = pd.DataFrame({
    '2024-01':retention_male_2024_01,
    '2024-02':retention_male_2024_02,
    '2024-03':retention_male_2024_03,
    '2024-04':retention_male_2024_04,
    '2024-05':retention_male_2024_05,
    '2024-06':retention_male_2024_06
})

# One-month female subscription users
retentions_female = pd.DataFrame({
    '2024-01':retention_female_2024_01,
    '2024-02':retention_female_2024_02,
    '2024-03':retention_female_2024_03,
    '2024-04':retention_female_2024_04,
    '2024-05':retention_female_2024_05,
    '2024-06':retention_female_2024_06
})

# One-month non-identified subscription users
retentions_non_identified = pd.DataFrame({
    '2024-01':retention_non_identified_2024_01,
    '2024-02':retention_non_identified_2024_02,
    '2024-03':retention_non_identified_2024_03,
    '2024-04':retention_non_identified_2024_04,
    '2024-05':retention_non_identified_2024_05,
    '2024-06':retention_non_identified_2024_06
})
```

▸ retentions_female: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
▸ retentions_male: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
▸ retentions_non_identified: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

4.1.1.2. Three-month subscription users

```
# Cohort 2024-01
male_user_of_cohort_2024_01_three_month = len(cohort_2024_01['three_month'][cohort_2024_01['three_month']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_01_three_month = len(cohort_2024_01['three_month'][cohort_2024_01['three_month']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_01_three_month = len(cohort_2024_01['three_month'][cohort_2024_01['three_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
## Retention rate for 3-month subscription male users Cohort 2024-01
retention_male_2024_01_three_month = [100,
    round(count_unique_users(renewals_2024_01_3_month, '2024-04', 'male') / male_user_of_cohort_2024_01_three_month * 100,
2),
    round(count_unique_users(renewals_2024_01_3_month, '2024-07', 'male') / male_user_of_cohort_2024_01_three_month * 100,
2),
    round(count_unique_users(renewals_2024_01_3_month, '2024-10', 'male') / male_user_of_cohort_2024_01_three_month * 100,
2)]
print('Retention rate of three-month subscription male users in Cohort 2024-01:')
print(retention_male_2024_01_three_month)


# Retention rate for 3-month subscription female users Cohort 2024-01
retention_female_2024_01_three_month = [100,
    round(count_unique_users(renewals_2024_01_3_month, '2024-04', 'female') / female_user_of_cohort_2024_01_three_month *
100, 2),
    round(count_unique_users(renewals_2024_01_3_month, '2024-07', 'female') / female_user_of_cohort_2024_01_three_month *
100, 2),
    round(count_unique_users(renewals_2024_01_3_month, '2024-10', 'female') / female_user_of_cohort_2024_01_three_month *
100, 2)
]
print('Retention rate of three-month subscription female users in Cohort 2024-01:')
print(retention_female_2024_01_three_month)

# Retention rate for 3-month subscription non-identified users Cohort 2024-01
retention_non_identified_2024_01_three_month = [100,
    round(count_unique_users(renewals_2024_01_3_month, '2024-04', 'non_identified') /
non_identified_user_of_cohort_2024_01_three_month * 100, 2),
    round(count_unique_users(renewals_2024_01_3_month, '2024-07', 'non_identified') /
non_identified_user_of_cohort_2024_01_three_month * 100, 2),
    round(count_unique_users(renewals_2024_01_3_month, '2024-10', 'non_identified') /
non_identified_user_of_cohort_2024_01_three_month * 100, 2)
]
print('Retention rate of three-month subscription non-identified users in Cohort 2024-01:')
print(retention_non_identified_2024_01_three_month)
```

```
Retention rate of three-month subscription male users in Cohort 2024-01:
[100, 46.27, 23.88, 16.42]
Retention rate of three-month subscription female users in Cohort 2024-01:
[100, 47.83, 26.09, 11.96]
Retention rate of three-month subscription non-identified users in Cohort 2024-01:
[100, 49.49, 27.27, 19.19]
```

```
# Cohort 2024-02
male_user_of_cohort_2024_02_three_month = len(cohort_2024_02['three_month'][cohort_2024_02['three_month']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_02_three_month = len(cohort_2024_02['three_month'][cohort_2024_02['three_month']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_02_three_month = len(cohort_2024_02['three_month'][cohort_2024_02['three_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Retention rate for 3-month subscription male users Cohort 2024-02
retention_male_2024_02_three_month = [100,
    round(count_unique_users(renewals_2024_02_3_month, '2024-05', 'male') / male_user_of_cohort_2024_02_three_month * 100,
2),
    round(count_unique_users(renewals_2024_02_3_month, '2024-08', 'male') / male_user_of_cohort_2024_02_three_month * 100,
2),
    round(count_unique_users(renewals_2024_02_3_month, '2024-11', 'male') / male_user_of_cohort_2024_02_three_month * 100,
2)
]
print('Retention rate of three-month subscription male users in Cohort 2024-02:')
print(retention_male_2024_02_three_month)


# Retention rate for 3-month subscription female users Cohort 2024-02
retention_female_2024_02_three_month = [100,
    round(count_unique_users(renewals_2024_02_3_month, '2024-05', 'female') / female_user_of_cohort_2024_02_three_month *
100, 2),
    round(count_unique_users(renewals_2024_02_3_month, '2024-08', 'female') / female_user_of_cohort_2024_02_three_month *
100, 2),
    round(count_unique_users(renewals_2024_02_3_month, '2024-11', 'female') / female_user_of_cohort_2024_02_three_month *
100, 2)
]
print('Retention rate of three-month subscription female users in Cohort 2024-02:')
print(retention_female_2024_02_three_month)


# Retention rate for 3-month subscription non-identified users Cohort 2024-02
retention_non_identified_2024_02_three_month = [100,
    round(count_unique_users(renewals_2024_02_3_month, '2024-05', 'non_identified') /
non_identified_user_of_cohort_2024_02_three_month * 100, 2),
    round(count_unique_users(renewals_2024_02_3_month, '2024-08', 'non_identified') /
non_identified_user_of_cohort_2024_02_three_month * 100, 2),
    round(count_unique_users(renewals_2024_02_3_month, '2024-11', 'non_identified') /
non_identified_user_of_cohort_2024_02_three_month * 100, 2)
]
print('Retention rate of three-month subscription non-identified users in Cohort 2024-02:')
print(retention_non_identified_2024_02_three_month)
```

```
Retention rate of three-month subscription male users in Cohort 2024-02:
[100, 37.3, 11.9, 7.94]
Retention rate of three-month subscription female users in Cohort 2024-02:
[100, 42.67, 28.0, 16.67]
Retention rate of three-month subscription non-identified users in Cohort 2024-02:
[100, 41.67, 22.06, 14.22]
```

```
# Cohort 2024-03
male_user_of_cohort_2024_03_three_month = len(cohort_2024_03['three_month'][cohort_2024_03['three_month']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_03_three_month = len(cohort_2024_03['three_month'][cohort_2024_03['three_month']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_03_three_month = len(cohort_2024_03['three_month'][cohort_2024_03['three_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Retention rate for 3-month subscription male users Cohort 2024-03
retention_male_2024_03_three_month = [100,
    round(count_unique_users(renewals_2024_03_3_month, '2024-06', 'male') / male_user_of_cohort_2024_03_three_month * 100,
2),
    round(count_unique_users(renewals_2024_03_3_month, '2024-09', 'male') / male_user_of_cohort_2024_03_three_month * 100,
2),
    round(count_unique_users(renewals_2024_03_3_month, '2024-12', 'male') / male_user_of_cohort_2024_03_three_month * 100,
2)
]
print('Retention rate of three-month subscription male users in Cohort 2024-03:')
print(retention_male_2024_03_three_month)


# Retention rate for 3-month subscription female users Cohort 2024-03
retention_female_2024_03_three_month = [100,
    round(count_unique_users(renewals_2024_03_3_month, '2024-06', 'female') / female_user_of_cohort_2024_03_three_month *
100, 2),
    round(count_unique_users(renewals_2024_03_3_month, '2024-09', 'female') / female_user_of_cohort_2024_03_three_month *
100, 2),
    round(count_unique_users(renewals_2024_03_3_month, '2024-12', 'female') / female_user_of_cohort_2024_03_three_month *
100, 2)
]
print('Retention rate of three-month subscription female users in Cohort 2024-03:')
print(retention_female_2024_03_three_month)


# Retention rate for 3-month subscription non-identified users Cohort 2024-03
retention_non_identified_2024_03_three_month = [100,
    round(count_unique_users(renewals_2024_03_3_month, '2024-06', 'non_identified') /
non_identified_user_of_cohort_2024_03_three_month * 100, 2),
    round(count_unique_users(renewals_2024_03_3_month, '2024-09', 'non_identified') /
non_identified_user_of_cohort_2024_03_three_month * 100, 2),
    round(count_unique_users(renewals_2024_03_3_month, '2024-12', 'non_identified') /
non_identified_user_of_cohort_2024_03_three_month * 100, 2)
]
print('Retention rate of three-month subscription non-identified users in Cohort 2024-03:')
print(retention_non_identified_2024_03_three_month)
```

```
Retention rate of three-month subscription male users in Cohort 2024-03:
[100, 37.5, 25.0, 11.46]
Retention rate of three-month subscription female users in Cohort 2024-03:
[100, 42.48, 23.01, 14.16]
Retention rate of three-month subscription non-identified users in Cohort 2024-03:
[100, 49.09, 30.3, 21.21]
```

```
# Cohort 2024-04
male_user_of_cohort_2024_04_three_month = len(cohort_2024_04['three_month'][cohort_2024_04['three_month']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_04_three_month = len(cohort_2024_04['three_month'][cohort_2024_04['three_month']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_04_three_month = len(cohort_2024_04['three_month'][cohort_2024_04['three_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Retention rate for 3-month subscription male users Cohort 2024-04
retention_male_2024_04_three_month = [100,
    round(count_unique_users(renewals_2024_04_3_month, '2024-07', 'male') / male_user_of_cohort_2024_04_three_month * 100,
2),
    round(count_unique_users(renewals_2024_04_3_month, '2024-10', 'male') / male_user_of_cohort_2024_04_three_month * 100,
2),
    round(count_unique_users(renewals_2024_04_3_month, '2025-01', 'male') / male_user_of_cohort_2024_04_three_month * 100,
2)
]
print('Retention rate of three-month subscription male users in Cohort 2024-04:')
print(retention_male_2024_04_three_month)


# Retention rate for 3-month subscription female users Cohort 2024-04
retention_female_2024_04_three_month = [100,
    round(count_unique_users(renewals_2024_04_3_month, '2024-07', 'female') / female_user_of_cohort_2024_04_three_month *
100, 2),
    round(count_unique_users(renewals_2024_04_3_month, '2024-10', 'female') / female_user_of_cohort_2024_04_three_month *
100, 2),
    round(count_unique_users(renewals_2024_04_3_month, '2025-01', 'female') / female_user_of_cohort_2024_04_three_month *
100, 2)
]
print('Retention rate of three-month subscription female users in Cohort 2024-04:')
print(retention_female_2024_04_three_month)


# Retention rate for 3-month subscription non-identified users Cohort 2024-04
retention_non_identified_2024_04_three_month = [100,
    round(count_unique_users(renewals_2024_04_3_month, '2024-07', 'non_identified') /
non_identified_user_of_cohort_2024_04_three_month * 100, 2),
    round(count_unique_users(renewals_2024_04_3_month, '2024-10', 'non_identified') /
non_identified_user_of_cohort_2024_04_three_month * 100, 2),
    round(count_unique_users(renewals_2024_04_3_month, '2025-01', 'non_identified') /
non_identified_user_of_cohort_2024_04_three_month * 100, 2)
]
print('Retention rate of three-month subscription non-identified users in Cohort 2024-04:')
print(retention_non_identified_2024_04_three_month)
```

```
Retention rate of three-month subscription male users in Cohort 2024-04:
[100, 48.42, 24.21, 16.84]
Retention rate of three-month subscription female users in Cohort 2024-04:
[100, 40.2, 22.55, 12.75]
Retention rate of three-month subscription non-identified users in Cohort 2024-04:
[100, 49.71, 29.48, 18.5]
```

```
# Cohort 2024-05
male_user_of_cohort_2024_05_three_month = len(cohort_2024_05['three_month'][cohort_2024_05['three_month']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_05_three_month = len(cohort_2024_05['three_month'][cohort_2024_05['three_month']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_05_three_month = len(cohort_2024_05['three_month'][cohort_2024_05['three_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Retention rate for 3-month subscription male users Cohort 2024-05
retention_male_2024_05_three_month = [100,
    round(count_unique_users(renewals_2024_05_3_month, '2024-08', 'male') / male_user_of_cohort_2024_05_three_month * 100,
2),
    round(count_unique_users(renewals_2024_05_3_month, '2024-11', 'male') / male_user_of_cohort_2024_05_three_month * 100,
2),
    round(count_unique_users(renewals_2024_05_3_month, '2025-01', 'male') / male_user_of_cohort_2024_05_three_month * 100,
2)
]
print('Retention rate of three-month subscription male users in Cohort 2024-05:')
print(retention_male_2024_05_three_month)


# Retention rate for 3-month subscription female users Cohort 2024-05
retention_female_2024_05_three_month = [100,
    round(count_unique_users(renewals_2024_05_3_month, '2024-08', 'female') / female_user_of_cohort_2024_05_three_month *
100, 2),
    round(count_unique_users(renewals_2024_05_3_month, '2024-11', 'female') / female_user_of_cohort_2024_05_three_month *
100, 2),
    round(count_unique_users(renewals_2024_05_3_month, '2025-01', 'female') / female_user_of_cohort_2024_05_three_month *
100, 2)
]
print('Retention rate of three-month subscription female users in Cohort 2024-05:')
print(retention_female_2024_05_three_month)


# Retention rate for 3-month subscription non-identified users Cohort 2024-05
retention_non_identified_2024_05_three_month = [100,
    round(count_unique_users(renewals_2024_05_3_month, '2024-08', 'non_identified') /
non_identified_user_of_cohort_2024_05_three_month * 100, 2),
    round(count_unique_users(renewals_2024_05_3_month, '2024-11', 'non_identified') /
non_identified_user_of_cohort_2024_05_three_month * 100, 2),
    round(count_unique_users(renewals_2024_05_3_month, '2025-01', 'non_identified') /
non_identified_user_of_cohort_2024_05_three_month * 100, 2)
]
print('Retention rate of three-month subscription non-identified users in Cohort 2024-05:')
print(retention_non_identified_2024_05_three_month)
```

```
Retention rate of three-month subscription male users in Cohort 2024-05:
[100, 42.11, 22.37, 0.0]
Retention rate of three-month subscription female users in Cohort 2024-05:
[100, 47.06, 25.49, 0.0]
Retention rate of three-month subscription non-identified users in Cohort 2024-05:
[100, 44.44, 24.79, 0.0]
```

```
# Cohort 2024-06
male_user_of_cohort_2024_06_three_month = len(cohort_2024_06['three_month'][cohort_2024_06['three_month']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_06_three_month = len(cohort_2024_06['three_month'][cohort_2024_06['three_month']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_06_three_month = len(cohort_2024_06['three_month'][cohort_2024_06['three_month']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
# Retention rate for 3-month subscription male users Cohort 2024-06
retention_male_2024_06_three_month = [100,
    round(count_unique_users(renewals_2024_06_3_month, '2024-09', 'male') / male_user_of_cohort_2024_06_three_month * 100,
2),
    round(count_unique_users(renewals_2024_06_3_month, '2024-12', 'male') / male_user_of_cohort_2024_06_three_month * 100,
2),
    round(count_unique_users(renewals_2024_06_3_month, '2025-02', 'male') / male_user_of_cohort_2024_06_three_month * 100,
2)
]
print('Retention rate of three-month subscription male users in Cohort 2024-06:')
print(retention_male_2024_06_three_month)


# Retention rate for 3-month subscription female users Cohort 2024-06
retention_female_2024_06_three_month = [100,
    round(count_unique_users(renewals_2024_06_3_month, '2024-09', 'female') / female_user_of_cohort_2024_06_three_month *
100, 2),
    round(count_unique_users(renewals_2024_06_3_month, '2024-12', 'female') / female_user_of_cohort_2024_06_three_month *
100, 2),
    round(count_unique_users(renewals_2024_06_3_month, '2025-02', 'female') / female_user_of_cohort_2024_06_three_month *
100, 2)
]
print('Retention rate of three-month subscription female users in Cohort 2024-06:')
print(retention_female_2024_06_three_month)


# Retention rate for 3-month subscription non-identified users Cohort 2024-06
retention_non_identified_2024_06_three_month = [100,
    round(count_unique_users(renewals_2024_06_3_month, '2024-09', 'non_identified') /
non_identified_user_of_cohort_2024_06_three_month * 100, 2),
    round(count_unique_users(renewals_2024_06_3_month, '2024-12', 'non_identified') /
non_identified_user_of_cohort_2024_06_three_month * 100, 2),
    round(count_unique_users(renewals_2024_06_3_month, '2025-02', 'non_identified') /
non_identified_user_of_cohort_2024_06_three_month * 100, 2)
]
print('Retention rate of three-month subscription non-identified users in Cohort 2024-06:')
print(retention_non_identified_2024_06_three_month)
```

```
Retention rate of three-month subscription male users in Cohort 2024-06:
[100, 44.19, 25.58, 0.0]
Retention rate of three-month subscription female users in Cohort 2024-06:
[100, 43.16, 23.16, 0.0]
Retention rate of three-month subscription non-identified users in Cohort 2024-06:
[100, 49.07, 34.16, 0.0]
```

```
male_retention_rate_three_month = pd.DataFrame({
    '2024-01': retention_male_2024_01_three_month,
    '2024-02': retention_male_2024_02_three_month,
    '2024-03': retention_male_2024_03_three_month,
    '2024-04': retention_male_2024_04_three_month,
    '2024-05': retention_male_2024_05_three_month,
    '2024-06': retention_male_2024_06_three_month
})

female_retention_rate_three_month = pd.DataFrame({
    '2024-01': retention_female_2024_01_three_month,
    '2024-02': retention_female_2024_02_three_month,
    '2024-03': retention_female_2024_03_three_month,
    '2024-04': retention_female_2024_04_three_month,
    '2024-05': retention_female_2024_05_three_month,
    '2024-06': retention_female_2024_06_three_month
})

non_identified_retention_rate_three_month = pd.DataFrame({
    '2024-01': retention_non_identified_2024_01_three_month,
    '2024-02': retention_non_identified_2024_02_three_month,
    '2024-03': retention_non_identified_2024_03_three_month,
    '2024-04': retention_non_identified_2024_04_three_month,
    '2024-05': retention_non_identified_2024_05_three_month,
    '2024-06': retention_non_identified_2024_06_three_month
})
```

- female_retention_rate_three_month: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
- male_retention_rate_three_month: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
- non_identified_retention_rate_three_month: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

4.1.1.3. One-year subscription users

```
male_user_of_cohort_2024_01_one_year = len(cohort_2024_01['one_year'][cohort_2024_01['one_year']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_01_one_year = len(cohort_2024_01['one_year'][cohort_2024_01['one_year']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_01_one_year = len(cohort_2024_01['one_year'][cohort_2024_01['one_year']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
## Retention rate for 1-year subscription male users Cohort 2024-01
retention_rate_male_2024_01_one_year = [100,
    round(count_unique_users(renewals_2024_01_1_year, '2025-01', 'male') / male_user_of_cohort_2024_01_one_year * 100, 2)]
print("Retention rate for 1-year subscription male users Cohort 2024-01: ")
print(retention_rate_male_2024_01_one_year)
## Retention rate for 1-year subscription female users Cohort 2024-01
retention_rate_female_2024_01_one_year = [100,
    round(count_unique_users(renewals_2024_01_1_year, '2025-01', 'female') / female_user_of_cohort_2024_01_one_year * 100,
2)]
print("Retention rate for 1-year subscription female users Cohort 2024-01: ")
print(retention_rate_female_2024_01_one_year)
## Retention rate for 1-year subscription non-identified users Cohort 2024-01
retention_rate_non_identified_2024_01_one_year = [100,
    round(count_unique_users(renewals_2024_01_1_year, '2025-01', 'non_identified') /
non_identified_user_of_cohort_2024_01_one_year * 100, 2)]
print('Retention rate for 1-year subscription non-identified users Cohort 2024-01: ')
print(retention_rate_non_identified_2024_01_one_year)
```

```
Retention rate for 1-year subscription male users Cohort 2024-01:
[100, 28.79]
Retention rate for 1-year subscription female users Cohort 2024-01:
[100, 31.11]
Retention rate for 1-year subscription non-identified users Cohort 2024-01:
[100, 31.82]
```

```
male_user_of_cohort_2024_02_one_year = len(cohort_2024_02['one_year'][cohort_2024_02['one_year']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_02_one_year = len(cohort_2024_02['one_year'][cohort_2024_02['one_year']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_02_one_year = len(cohort_2024_02['one_year'][cohort_2024_02['one_year']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
## Retention rate for 1-year subscription male users Cohort 2024-02
retention_rate_male_2024_02_one_year = [100,
    round(count_unique_users(renewals_2024_02_1_year, '2025-02', 'male') / male_user_of_cohort_2024_02_one_year * 100, 2)]
print("Retention rate for 1-year subscription male users Cohort 2024-02: ")
print(retention_rate_male_2024_02_one_year)

## Retention rate for 1-year subscription female users Cohort 2024-02
retention_rate_female_2024_02_one_year = [100,
    round(count_unique_users(renewals_2024_02_1_year, '2025-02', 'female') / female_user_of_cohort_2024_02_one_year * 100,
2)]
print("Retention rate for 1-year subscription female users Cohort 2024-02: ")
print(retention_rate_female_2024_02_one_year)

## Retention rate for 1-year subscription non-identified users Cohort 2024-02
retention_rate_non_identified_2024_02_one_year = [100,
    round(count_unique_users(renewals_2024_02_1_year, '2025-02', 'non_identified') /
non_identified_user_of_cohort_2024_02_one_year * 100, 2)]
print('Retention rate for 1-year subscription non-identified users Cohort 2024-02: ')
print(retention_rate_non_identified_2024_02_one_year)
```

```
Retention rate for 1-year subscription male users Cohort 2024-02:
[100, 31.96]
Retention rate for 1-year subscription female users Cohort 2024-02:
[100, 31.06]
Retention rate for 1-year subscription non-identified users Cohort 2024-02:
[100, 33.33]
```

```
male_user_of_cohort_2024_03_one_year = len(cohort_2024_03['one_year'][cohort_2024_03['one_year']['gender'] == 1]['user_id'].drop
female_user_of_cohort_2024_03_one_year = len(cohort_2024_03['one_year'][cohort_2024_03['one_year']['gender'] == 0]['user_id'].dr
non_identified_user_of_cohort_2024_03_one_year = len(cohort_2024_03['one_year'][cohort_2024_03['one_year']['gender'].isna()]['us
```

```
## Retention rate for 1-year subscription male users Cohort 2024-03
retention_rate_male_2024_03_one_year = [100,
    round(count_unique_users(renewals_2024_03_1_year, '2025-03', 'male') / male_user_of_cohort_2024_03_one_year * 100, 2)]
print("Retention rate for 1-year subscription male users Cohort 2024-03: ")
print(retention_rate_male_2024_03_one_year)

## Retention rate for 1-year subscription female users Cohort 2024-03
retention_rate_female_2024_03_one_year = [100,
    round(count_unique_users(renewals_2024_03_1_year, '2025-03', 'female') / female_user_of_cohort_2024_03_one_year * 100,
2)]
print("Retention rate for 1-year subscription female users Cohort 2024-03: ")
print(retention_rate_female_2024_03_one_year)

## Retention rate for 1-year subscription non-identified users Cohort 2024-03
retention_rate_non_identified_2024_03_one_year = [100,
    round(count_unique_users(renewals_2024_03_1_year, '2025-03', 'non_identified') /
non_identified_user_of_cohort_2024_03_one_year * 100, 2)]
print('Retention rate for 1-year subscription non-identified users Cohort 2024-03: ')
print(retention_rate_non_identified_2024_03_one_year)
```

```
Retention rate for 1-year subscription male users Cohort 2024-03:
[100, 38.33]
Retention rate for 1-year subscription female users Cohort 2024-03:
[100, 38.36]
Retention rate for 1-year subscription non-identified users Cohort 2024-03:
[100, 33.86]
```

```
male_user_of_cohort_2024_04_one_year = len(cohort_2024_04['one_year'][cohort_2024_04['one_year']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_04_one_year = len(cohort_2024_04['one_year'][cohort_2024_04['one_year']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_04_one_year = len(cohort_2024_04['one_year'][cohort_2024_04['one_year']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
## Retention rate for 1-year subscription male users Cohort 2024-04
retention_rate_male_2024_04_one_year = [100,
    round(count_unique_users(renewals_2024_04_1_year, '2025-04', 'male') / male_user_of_cohort_2024_04_one_year * 100, 2)]
print("Retention rate for 1-year subscription male users Cohort 2024-04: ")
print(retention_rate_male_2024_04_one_year)

## Retention rate for 1-year subscription female users Cohort 2024-04
retention_rate_female_2024_04_one_year = [100,
    round(count_unique_users(renewals_2024_04_1_year, '2025-04', 'female') / female_user_of_cohort_2024_04_one_year * 100, 2)]
print("Retention rate for 1-year subscription female users Cohort 2024-04: ")
print(retention_rate_female_2024_04_one_year)

## Retention rate for 1-year subscription non-identified users Cohort 2024-04
retention_rate_non_identified_2024_04_one_year = [100,
    round(count_unique_users(renewals_2024_04_1_year, '2025-04', 'non_identified') / non_identified_user_of_cohort_2024_04_one_y
print('Retention rate for 1-year subscription non-identified users Cohort 2024-04: ')
print(retention_rate_non_identified_2024_04_one_year)
```

```
Retention rate for 1-year subscription male users Cohort 2024-04:
[100, 34.25]
Retention rate for 1-year subscription female users Cohort 2024-04:
[100, 43.48]
Retention rate for 1-year subscription non-identified users Cohort 2024-04:
[100, 32.82]
```

```
male_user_of_cohort_2024_05_one_year = len(cohort_2024_05['one_year'][cohort_2024_05['one_year']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_05_one_year = len(cohort_2024_05['one_year'][cohort_2024_05['one_year']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_05_one_year = len(cohort_2024_05['one_year'][cohort_2024_05['one_year']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
## Retention rate for 1-year subscription male users Cohort 2024-05
retention_rate_male_2024_05_one_year = [100,
    round(count_unique_users(renewals_2024_05_1_year, '2025-05', 'male') / male_user_of_cohort_2024_05_one_year * 100, 2)]
print("Retention rate for 1-year subscription male users Cohort 2024-05: ")
print(retention_rate_male_2024_05_one_year)

## Retention rate for 1-year subscription female users Cohort 2024-05
retention_rate_female_2024_05_one_year = [100,
    round(count_unique_users(renewals_2024_05_1_year, '2025-05', 'female') / female_user_of_cohort_2024_05_one_year * 100,
2)]
print("Retention rate for 1-year subscription female users Cohort 2024-05: ")
print(retention_rate_female_2024_05_one_year)

## Retention rate for 1-year subscription non-identified users Cohort 2024-05
retention_rate_non_identified_2024_05_one_year = [100,
    round(count_unique_users(renewals_2024_05_1_year, '2025-05', 'non_identified') /
non_identified_user_of_cohort_2024_05_one_year * 100, 2)]
print('Retention rate for 1-year subscription non-identified users Cohort 2024-05: ')
print(retention_rate_non_identified_2024_05_one_year)
```

```
Retention rate for 1-year subscription male users Cohort 2024-05:
[100, 33.33]
Retention rate for 1-year subscription female users Cohort 2024-05:
[100, 23.08]
Retention rate for 1-year subscription non-identified users Cohort 2024-05:
[100, 33.06]
```

```
male_user_of_cohort_2024_06_one_year = len(cohort_2024_06['one_year'][cohort_2024_06['one_year']['gender'] == 1]
['user_id'].drop_duplicates())
female_user_of_cohort_2024_06_one_year = len(cohort_2024_06['one_year'][cohort_2024_06['one_year']['gender'] == 0]
['user_id'].drop_duplicates())
non_identified_user_of_cohort_2024_06_one_year = len(cohort_2024_06['one_year'][cohort_2024_06['one_year']
['gender'].isna()]['user_id'].drop_duplicates())
```

```
## Retention rate for 1-year subscription male users Cohort 2024-06
retention_rate_male_2024_06_one_year = [100,
    round(count_unique_users(renewals_2024_06_1_year, '2025-06', 'male') / male_user_of_cohort_2024_06_one_year * 100, 2)]
print("Retention rate for 1-year subscription male users Cohort 2024-06: ")
print(retention_rate_male_2024_06_one_year)

## Retention rate for 1-year subscription female users Cohort 2024-06
retention_rate_female_2024_06_one_year = [100,
    round(count_unique_users(renewals_2024_06_1_year, '2025-06', 'female') / female_user_of_cohort_2024_06_one_year * 100,
2)]
print("Retention rate for 1-year subscription female users Cohort 2024-06: ")
print(retention_rate_female_2024_06_one_year)

## Retention rate for 1-year subscription non-identified users Cohort 2024-06
retention_rate_non_identified_2024_06_one_year = [100,
    round(count_unique_users(renewals_2024_06_1_year, '2025-06', 'non_identified') /
non_identified_user_of_cohort_2024_06_one_year * 100, 2)]
print('Retention rate for 1-year subscription non-identified users Cohort 2024-06: ')
print(retention_rate_non_identified_2024_06_one_year)
```

```
Retention rate for 1-year subscription male users Cohort 2024-06:
[100, 49.06]
Retention rate for 1-year subscription female users Cohort 2024-06:
[100, 39.34]
Retention rate for 1-year subscription non-identified users Cohort 2024-06:
[100, 30.38]
```

```
male_retention_rate_one_year = pd.DataFrame ({
    '2024-01': retention_rate_male_2024_01_one_year,
    '2024-02': retention_rate_male_2024_02_one_year,
    '2024-03': retention_rate_male_2024_03_one_year,
    '2024-04': retention_rate_male_2024_04_one_year,
    '2024-05': retention_rate_male_2024_05_one_year,
    '2024-06': retention_rate_male_2024_06_one_year
})

female_retention_rate_one_year = pd.DataFrame ({
    '2024-01': retention_rate_female_2024_01_one_year,
    '2024-02': retention_rate_female_2024_02_one_year,
    '2024-03': retention_rate_female_2024_03_one_year,
    '2024-04': retention_rate_female_2024_04_one_year,
    '2024-05': retention_rate_female_2024_05_one_year,
    '2024-06': retention_rate_female_2024_06_one_year
})

non_identified_retention_rate_one_year = pd.DataFrame ({
    '2024-01': retention_rate_non_identified_2024_01_one_year,
    '2024-02': retention_rate_non_identified_2024_02_one_year,
    '2024-03': retention_rate_non_identified_2024_03_one_year,
    '2024-04': retention_rate_non_identified_2024_04_one_year,
    '2024-05': retention_rate_non_identified_2024_05_one_year,
    '2024-06': retention_rate_non_identified_2024_06_one_year
})
```

▸ female_retention_rate_one_year: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
▸ male_retention_rate_one_year: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
▸ non_identified_retention_rate_one_year: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

4.1.2. Retention rates by platforms

Similar to genders, the same structures are applied for retention rates by platforms.

4.1.2.1. One-month retention rates

```
# One-month subscription users in Cohort 2024-01 on 2 platforms
appstore_user_of_cohort_2024_01_one_month = len(cohort_2024_01['one_month'][cohort_2024_01['one_month']['payment_gateway']
== 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_01_one_month = len(cohort_2024_01['one_month'][cohort_2024_01['one_month']['payment_gateway']
== 'playstore']['user_id'].drop_duplicates())
```

```
# Retention rates for one-month subscription users on Appstore in Cohort 2024-01
retention_2024_01_one_month_appstore = [100,
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-02', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-03', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-04', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-05', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-06', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_01_one_month * 100, 2)
    ]
print('Retention rate of one-month subscription users on Appstore in Cohort 2024-06:')
print(retention_2024_01_one_month_appstore)

# Retention rates for one-month subscription users on Playstore in Cohort 2024-01
retention_2024_01_one_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-02', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-03', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-04', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-05', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-06', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-08', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-09', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-10', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
```

```
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-11', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_1_month_platform, '2024-12', 'playstore') /
playstore_user_of_cohort_2024_01_one_month * 100, 2)
]

print('Retention rate of one-month subscription users on Playstore in Cohort 2024-01:')
print(retention_2024_01_one_month_playstore)
```

```
Retention rate of one-month subscription users on Appstore in Cohort 2024-06:
[100, 58.42, 42.76, 30.13, 22.9, 19.19, 15.82, 14.14, 13.3, 10.77, 10.27, 9.26]
Retention rate of one-month subscription users on Playstore in Cohort 2024-01:
[100, 53.21, 37.31, 29.66, 25.08, 18.96, 12.84, 10.4, 9.48, 10.4, 10.09, 6.73]
```

```
# Calculate one-month cohort user counts for 2024-02 on 2 platforms
appstore_user_of_cohort_2024_02_one_month = len(cohort_2024_02['one_month'][cohort_2024_02['one_month']['payment_gateway']
== 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_02_one_month = len(cohort_2024_02['one_month'][cohort_2024_02['one_month']['payment_gateway']
== 'playstore']['user_id'].drop_duplicates())

# Retention rates for Appstore, Cohort 2024-02 (March 2024 to January 2025)
retention_2024_02_one_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-03', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-04', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-05', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-06', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2025-01', 'appstore') /
appstore_user_of_cohort_2024_02_one_month * 100, 2)
]

print('Retention rate of one-month subscription users on Appstore in Cohort 2024-02:')
print(retention_2024_02_one_month_appstore)

# Retention rates for Playstore, Cohort 2024-02 (March 2024 to January 2025)
retention_2024_02_one_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-03', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-04', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-05', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-06', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
```

```
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-08', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-09', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-10', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-11', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2024-12', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_1_month_platform, '2025-01', 'playstore') /
playstore_user_of_cohort_2024_02_one_month * 100, 2)
]

print('Retention rate of one-month subscription users on Playstore in Cohort 2024-02:')
print(retention_2024_02_one_month_playstore)
```

```
Retention rate of one-month subscription users on Appstore in Cohort 2024-02:
[100, 59.65, 40.78, 30.37, 23.86, 18.22, 16.05, 14.97, 13.45, 10.2, 8.68, 8.89]
Retention rate of one-month subscription users on Playstore in Cohort 2024-02:
[100, 54.48, 33.21, 26.87, 20.9, 16.42, 14.93, 12.31, 10.07, 9.7, 7.46, 7.46]
```

```
# Calculate one-month cohort user counts for 2024-03 on 2 platforms
appstore_user_of_cohort_2024_03_one_month = len(cohort_2024_03['one_month'][cohort_2024_03['one_month']['payment_gateway']
== 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_03_one_month = len(cohort_2024_03['one_month'][cohort_2024_03['one_month']['payment_gateway']
== 'playstore']['user_id'].drop_duplicates())

# Retention rates for Appstore, Cohort 2024-03 (April 2024 to February 2025)
retention_2024_03_one_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-04', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-05', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-06', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2025-01', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2025-02', 'appstore') /
appstore_user_of_cohort_2024_03_one_month * 100, 2)
]

print('Retention rate of one-month subscription users on Appstore in Cohort 2024-03:')
print(retention_2024_03_one_month_appstore)


# Retention rates for Playstore, Cohort 2024-03 (April 2024 to February 2025)
retention_2024_03_one_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-04', 'playstore') /
playstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-05', 'playstore') /
playstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-06', 'playstore') /
playstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_03_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-08', 'playstore') /
```

```
    playstore_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-09', 'playstore') /
    playstore_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-10', 'playstore') /
    playstore_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-11', 'playstore') /
    playstore_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2024-12', 'playstore') /
    playstore_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2025-01', 'playstore') /
    playstore_user_of_cohort_2024_03_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_03_1_month_platform, '2025-02', 'playstore') /
    playstore_user_of_cohort_2024_03_one_month * 100, 2)
    ]

    print('Retention rate of one-month subscription users on Playstore in Cohort 2024-03:')
    print(retention_2024_03_one_month_playstore)
```

```
Retention rate of one-month subscription users on Appstore in Cohort 2024-03:
[100, 57.52, 43.57, 32.68, 24.18, 18.95, 14.38, 11.33, 11.11, 10.02, 8.06, 7.41]
Retention rate of one-month subscription users on Playstore in Cohort 2024-03:
[100, 52.63, 38.46, 28.34, 23.08, 20.24, 16.19, 14.57, 11.74, 12.15, 9.72, 6.07]
```

```
# Calculate one-month cohort user counts for 2024-04 on 2 platforms
appstore_user_of_cohort_2024_04_one_month = len(cohort_2024_04['one_month'][cohort_2024_04['one_month']['payment_gateway']
== 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_04_one_month = len(cohort_2024_04['one_month'][cohort_2024_04['one_month']['payment_gateway']
== 'playstore']['user_id'].drop_duplicates())

# Retention rates for Appstore, Cohort 2024-04 (May 2024 to March 2025)
retention_2024_04_one_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-05', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-06', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2025-01', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2025-02', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2025-03', 'appstore') /
appstore_user_of_cohort_2024_04_one_month * 100, 2)
]

print('Retention rate of one-month subscription users on Appstore in Cohort 2024-04:')
print(retention_2024_04_one_month_appstore)


# Retention rates for Playstore, Cohort 2024-04 (May 2024 to March 2025)
retention_2024_04_one_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-05', 'playstore') /
playstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-06', 'playstore') /
playstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-08', 'playstore') /
playstore_user_of_cohort_2024_04_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-09', 'playstore') /
```

```
    playstore_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-10', 'playstore') /
    playstore_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-11', 'playstore') /
    playstore_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2024-12', 'playstore') /
    playstore_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2025-01', 'playstore') /
    playstore_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2025-02', 'playstore') /
    playstore_user_of_cohort_2024_04_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_04_1_month_platform, '2025-03', 'playstore') /
    playstore_user_of_cohort_2024_04_one_month * 100, 2)
    ]

    print('Retention rate of one-month subscription users on Playstore in Cohort 2024-04:')
    print(retention_2024_04_one_month_playstore)
```

```
Retention rate of one-month subscription users on Appstore in Cohort 2024-04:
[100, 54.83, 38.64, 27.94, 22.45, 18.02, 16.71, 12.27, 13.32, 10.97, 9.92, 7.83]
Retention rate of one-month subscription users on Playstore in Cohort 2024-04:
[100, 56.37, 39.71, 30.39, 21.57, 17.65, 17.16, 14.22, 12.75, 9.31, 9.31, 10.29]
```

```
# Calculate one-month cohort user counts for 2024-05 on 2 platforms
appstore_user_of_cohort_2024_05_one_month = len(cohort_2024_05['one_month'][cohort_2024_05['one_month']['payment_gateway']
== 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_05_one_month = len(cohort_2024_05['one_month'][cohort_2024_05['one_month']['payment_gateway']
== 'playstore']['user_id'].drop_duplicates())

# Retention rates for Appstore, Cohort 2024-05 (June 2024 to April 2025)
retention_2024_05_one_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-06', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-01', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-02', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-03', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-04', 'appstore') /
appstore_user_of_cohort_2024_05_one_month * 100, 2)
]

print('Retention rate of one-month subscription users on Appstore in Cohort 2024-05:')
print(retention_2024_05_one_month_appstore)


# Retention rates for Playstore, Cohort 2024-05 (June 2024 to April 2025)
retention_2024_05_one_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-06', 'playstore') /
playstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-08', 'playstore') /
playstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-09', 'playstore') /
playstore_user_of_cohort_2024_05_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-10', 'playstore') /
```

```
    playstore_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-11', 'playstore') /
    playstore_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2024-12', 'playstore') /
    playstore_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-01', 'playstore') /
    playstore_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-02', 'playstore') /
    playstore_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-03', 'playstore') /
    playstore_user_of_cohort_2024_05_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_05_1_month_platform, '2025-04', 'playstore') /
    playstore_user_of_cohort_2024_05_one_month * 100, 2)
    ]

    print('Retention rate of one-month subscription users on Playstore in Cohort 2024-05:')
    print(retention_2024_05_one_month_playstore)
```

```
Retention rate of one-month subscription users on Appstore in Cohort 2024-05:
[100, 60.37, 43.35, 34.31, 27.39, 22.34, 19.41, 15.69, 13.3, 13.03, 11.17, 10.11]
Retention rate of one-month subscription users on Playstore in Cohort 2024-05:
[100, 52.12, 33.05, 26.27, 18.22, 15.68, 12.29, 11.86, 11.02, 8.05, 8.05, 6.36]
```

```
# Calculate one-month cohort user counts for 2024-06 on 2 platforms
appstore_user_of_cohort_2024_06_one_month = len(cohort_2024_06['one_month'][cohort_2024_06['one_month']['payment_gateway']
== 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_06_one_month = len(cohort_2024_06['one_month'][cohort_2024_06['one_month']['payment_gateway']
== 'playstore']['user_id'].drop_duplicates())

# Retention rates for Appstore, Cohort 2024-06 (July 2024 to May 2025)
retention_2024_06_one_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-01', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-02', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-03', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-04', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-05', 'appstore') /
appstore_user_of_cohort_2024_06_one_month * 100, 2)
]

print('Retention rate of one-month subscription users on Appstore in Cohort 2024-06:')
print(retention_2024_06_one_month_appstore)


# Retention rates for Playstore, Cohort 2024-06 (July 2024 to May 2025)
retention_2024_06_one_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-08', 'playstore') /
playstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-09', 'playstore') /
playstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-10', 'playstore') /
playstore_user_of_cohort_2024_06_one_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-11', 'playstore') /
```

```
    playstore_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2024-12', 'playstore') /
    playstore_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-01', 'playstore') /
    playstore_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-02', 'playstore') /
    playstore_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-03', 'playstore') /
    playstore_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-04', 'playstore') /
    playstore_user_of_cohort_2024_06_one_month * 100, 2),
        round(count_unique_users_by_gateway(renewals_2024_06_1_month_platform, '2025-05', 'playstore') /
    playstore_user_of_cohort_2024_06_one_month * 100, 2)
    ]

    print('Retention rate of one-month subscription users on Playstore in Cohort 2024-06:')
    print(retention_2024_06_one_month_playstore)
```

```
Retention rate of one-month subscription users on Appstore in Cohort 2024-06:
[100, 65.52, 41.38, 32.29, 26.02, 20.06, 16.3, 12.85, 12.85, 10.34, 9.09, 6.58]
Retention rate of one-month subscription users on Playstore in Cohort 2024-06:
[100, 51.67, 38.89, 27.78, 22.22, 20.0, 19.44, 15.0, 11.11, 10.56, 8.89, 8.33]
```

```
    # Appstore retention rates
    appstore_retention_rates_1_month = pd.DataFrame({
        '2024-01': retention_2024_01_one_month_appstore,
        '2024-02': retention_2024_02_one_month_appstore,
        '2024-03': retention_2024_03_one_month_appstore,
        '2024-04': retention_2024_04_one_month_appstore,
        '2024-05': retention_2024_05_one_month_appstore,
        '2024-06': retention_2024_06_one_month_appstore
    })

    # Playstore retention rates
    playstore_retention_rates_1_month = pd.DataFrame({
        '2024-01': retention_2024_01_one_month_playstore,
        '2024-02': retention_2024_02_one_month_playstore,
        '2024-03': retention_2024_03_one_month_playstore,
        '2024-04': retention_2024_04_one_month_playstore,
        '2024-05': retention_2024_05_one_month_playstore,
        '2024-06': retention_2024_06_one_month_playstore
    })
```

▸ appstore_retention_rates_1_month: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
▸ playstore_retention_rates_1_month: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

4.1.2.2. Three-month retention rates

```
# Calculate three-month cohort user counts for 2024-01 on 2 platforms
appstore_user_of_cohort_2024_01_three_month = len(cohort_2024_01['three_month'][cohort_2024_01['three_month']
['payment_gateway'] == 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_01_three_month = len(cohort_2024_01['three_month'][cohort_2024_01['three_month']
['payment_gateway'] == 'playstore']['user_id'].drop_duplicates())

#Retention rates of three-month subscription users on Appstore
retention_2024_01_three_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_01_3_month_payment_gateway, '2024-04', 'appstore') /
appstore_user_of_cohort_2024_01_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_3_month_payment_gateway, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_01_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_3_month_payment_gateway, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_01_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Appstore in Cohort 2024-01:')
print(retention_2024_01_three_month_appstore)

#Retention rates of three-month subscription users on Playstore
retention_2024_01_three_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_01_3_month_payment_gateway, '2024-04', 'playstore') /
playstore_user_of_cohort_2024_01_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_3_month_payment_gateway, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_01_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_01_3_month_payment_gateway, '2024-10', 'playstore') /
playstore_user_of_cohort_2024_01_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Playstore in Cohort 2024-01:')
print(retention_2024_01_three_month_playstore)
```

```
Retention rate of three-month subscription users on Appstore in Cohort 2024-01:
[100, 50.85, 28.81, 17.51]
Retention rate of three-month subscription users on Playstore in Cohort 2024-01:
[100, 56.67, 28.33, 16.67]
```

```
# Calculate three-month cohort user counts for 2024-02 on 2 platforms
appstore_user_of_cohort_2024_02_three_month = len(cohort_2024_02['three_month'][cohort_2024_02['three_month']
['payment_gateway'] == 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_02_three_month = len(cohort_2024_02['three_month'][cohort_2024_02['three_month']
['payment_gateway'] == 'playstore']['user_id'].drop_duplicates())

# Retention rates for three-month plan users on Appstore, Cohort 2024-02
retention_2024_02_three_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_02_3_month_payment_gateway, '2024-05', 'appstore') /
appstore_user_of_cohort_2024_02_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_3_month_payment_gateway, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_02_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_3_month_payment_gateway, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_02_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Appstore in Cohort 2024-02:')
print(retention_2024_02_three_month_appstore)

# Retention rates for three-month plan users on Playstore, Cohort 2024-02
retention_2024_02_three_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_02_3_month_payment_gateway, '2024-05', 'playstore') /
playstore_user_of_cohort_2024_02_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_3_month_payment_gateway, '2024-08', 'playstore') /
playstore_user_of_cohort_2024_02_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_02_3_month_payment_gateway, '2024-11', 'playstore') /
playstore_user_of_cohort_2024_02_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Playstore in Cohort 2024-02:')
print(retention_2024_02_three_month_playstore)
```

```
Retention rate of three-month subscription users on Appstore in Cohort 2024-02:
[100, 48.36, 24.48, 15.22]
Retention rate of three-month subscription users on Playstore in Cohort 2024-02:
[100, 43.59, 25.64, 16.67]
```

```
# Calculate three-month cohort user counts for 2024-03 on 2 platforms
appstore_user_of_cohort_2024_03_three_month = len(cohort_2024_03['three_month'][cohort_2024_03['three_month']
['payment_gateway'] == 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_03_three_month = len(cohort_2024_03['three_month'][cohort_2024_03['three_month']
['payment_gateway'] == 'playstore']['user_id'].drop_duplicates())

# Retention rates for three-month plan users on Appstore, Cohort 2024-03
retention_2024_03_three_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_03_3_month_payment_gateway, '2024-06', 'appstore') /
appstore_user_of_cohort_2024_03_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_3_month_payment_gateway, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_03_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_3_month_payment_gateway, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_03_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Appstore in Cohort 2024-03:')
print(retention_2024_03_three_month_appstore)

# Retention rates for three-month plan users on Playstore, Cohort 2024-03
retention_2024_03_three_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_03_3_month_payment_gateway, '2024-06', 'playstore') /
playstore_user_of_cohort_2024_03_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_3_month_payment_gateway, '2024-09', 'playstore') /
playstore_user_of_cohort_2024_03_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_03_3_month_payment_gateway, '2024-12', 'playstore') /
playstore_user_of_cohort_2024_03_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Playstore in Cohort 2024-03:')
print(retention_2024_03_three_month_playstore)
```

```
Retention rate of three-month subscription users on Appstore in Cohort 2024-03:
[100, 45.78, 28.9, 18.18]
Retention rate of three-month subscription users on Playstore in Cohort 2024-03:
[100, 55.26, 23.68, 13.16]
```

```
# Calculate three-month cohort user counts for 2024-04 on 2 platforms
appstore_user_of_cohort_2024_04_three_month = len(cohort_2024_04['three_month'][cohort_2024_04['three_month']
['payment_gateway'] == 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_04_three_month = len(cohort_2024_04['three_month'][cohort_2024_04['three_month']
['payment_gateway'] == 'playstore']['user_id'].drop_duplicates())

# Retention rates for three-month plan users on Appstore, Cohort 2024-04
retention_2024_04_three_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_04_3_month_payment_gateway, '2024-07', 'appstore') /
appstore_user_of_cohort_2024_04_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_3_month_payment_gateway, '2024-10', 'appstore') /
appstore_user_of_cohort_2024_04_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_3_month_payment_gateway, '2025-01', 'appstore') /
appstore_user_of_cohort_2024_04_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Appstore in Cohort 2024-04:')
print(retention_2024_04_three_month_appstore)

# Retention rates for three-month plan users on Playstore, Cohort 2024-04
retention_2024_04_three_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_04_3_month_payment_gateway, '2024-07', 'playstore') /
playstore_user_of_cohort_2024_04_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_3_month_payment_gateway, '2024-10', 'playstore') /
playstore_user_of_cohort_2024_04_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_04_3_month_payment_gateway, '2025-01', 'playstore') /
playstore_user_of_cohort_2024_04_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Playstore in Cohort 2024-04:')
print(retention_2024_04_three_month_playstore)
```

```
Retention rate of three-month subscription users on Appstore in Cohort 2024-04:
[100, 50.91, 28.73, 18.55]
Retention rate of three-month subscription users on Playstore in Cohort 2024-04:
[100, 47.89, 25.35, 12.68]
```

```
# Calculate three-month cohort user counts for 2024-05 on 2 platforms
appstore_user_of_cohort_2024_05_three_month = len(cohort_2024_05['three_month'][cohort_2024_05['three_month']
['payment_gateway'] == 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_05_three_month = len(cohort_2024_05['three_month'][cohort_2024_05['three_month']
['payment_gateway'] == 'playstore']['user_id'].drop_duplicates())

# Retention rates for three-month plan users on Appstore, Cohort 2024-05
retention_2024_05_three_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_05_3_month_payment_gateway, '2024-08', 'appstore') /
appstore_user_of_cohort_2024_05_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_3_month_payment_gateway, '2024-11', 'appstore') /
appstore_user_of_cohort_2024_05_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_3_month_payment_gateway, '2025-02', 'appstore') /
appstore_user_of_cohort_2024_05_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Appstore in Cohort 2024-05:')
print(retention_2024_05_three_month_appstore)

# Retention rates for three-month plan users on Playstore, Cohort 2024-05
retention_2024_05_three_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_05_3_month_payment_gateway, '2024-08', 'playstore') /
playstore_user_of_cohort_2024_05_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_3_month_payment_gateway, '2024-11', 'playstore') /
playstore_user_of_cohort_2024_05_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_05_3_month_payment_gateway, '2025-02', 'playstore') /
playstore_user_of_cohort_2024_05_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Playstore in Cohort 2024-05:')
print(retention_2024_05_three_month_playstore)
```

```
Retention rate of three-month subscription users on Appstore in Cohort 2024-05:
[100, 49.58, 27.31, 13.87]
Retention rate of three-month subscription users on Playstore in Cohort 2024-05:
[100, 41.18, 17.65, 11.76]
```

```
# Calculate three-month cohort user counts for 2024-06 on 2 platforms
appstore_user_of_cohort_2024_06_three_month = len(cohort_2024_06['three_month'][cohort_2024_06['three_month']
['payment_gateway'] == 'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_06_three_month = len(cohort_2024_06['three_month'][cohort_2024_06['three_month']
['payment_gateway'] == 'playstore']['user_id'].drop_duplicates())

# Retention rates for three-month plan users on Appstore, Cohort 2024-06
retention_2024_06_three_month_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_06_3_month_payment_gateway, '2024-09', 'appstore') /
appstore_user_of_cohort_2024_06_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_3_month_payment_gateway, '2024-12', 'appstore') /
appstore_user_of_cohort_2024_06_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_3_month_payment_gateway, '2025-03', 'appstore') /
appstore_user_of_cohort_2024_06_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Appstore in Cohort 2024-06:')
print(retention_2024_06_three_month_appstore)

# Retention rates for three-month plan users on Playstore, Cohort 2024-06
retention_2024_06_three_month_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_06_3_month_payment_gateway, '2024-09', 'playstore') /
playstore_user_of_cohort_2024_06_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_3_month_payment_gateway, '2024-12', 'playstore') /
playstore_user_of_cohort_2024_06_three_month * 100, 2),
    round(count_unique_users_by_gateway(renewals_2024_06_3_month_payment_gateway, '2025-03', 'playstore') /
playstore_user_of_cohort_2024_06_three_month * 100, 2)
]
print('Retention rate of three-month subscription users on Playstore in Cohort 2024-06:')
print(retention_2024_06_three_month_playstore)
```

```
Retention rate of three-month subscription users on Appstore in Cohort 2024-06:
[100, 47.84, 28.24, 19.61]
Retention rate of three-month subscription users on Playstore in Cohort 2024-06:
[100, 48.48, 36.36, 27.27]
```

```
# Appstore retention rates for three-month subscription users
appstore_retention_rates_3_month = pd.DataFrame({
    '2024-01': retention_2024_01_three_month_appstore,
    '2024-02': retention_2024_02_three_month_appstore,
    '2024-03': retention_2024_03_three_month_appstore,
    '2024-04': retention_2024_04_three_month_appstore,
    '2024-05': retention_2024_05_three_month_appstore,
    '2024-06': retention_2024_06_three_month_appstore
})

# Playstore retention rates for three-month subscription users
playstore_retention_rates_3_month = pd.DataFrame({
    '2024-01': retention_2024_01_three_month_playstore,
    '2024-02': retention_2024_02_three_month_playstore,
    '2024-03': retention_2024_03_three_month_playstore,
    '2024-04': retention_2024_04_three_month_playstore,
    '2024-05': retention_2024_05_three_month_playstore,
    '2024-06': retention_2024_06_three_month_playstore
})
```

▸ appstore_retention_rates_3_month: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
▸ playstore_retention_rates_3_month: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

4.1.2.3. One-year retention rates

```
# Calculate one-year cohort user counts for 2024-01 on 2 platforms
appstore_user_of_cohort_2024_01_one_year = len(cohort_2024_01['one_year'][cohort_2024_01['one_year']['payment_gateway'] ==
'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_01_one_year = len(cohort_2024_01['one_year'][cohort_2024_01['one_year']['payment_gateway'] ==
'playstore']['user_id'].drop_duplicates())

#Retention rates of one-year subscription users on Appstore
retention_2024_01_one_year_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_01_1_year_payment_gateway, '2025-01', 'appstore') /
appstore_user_of_cohort_2024_01_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Appstore in Cohort 2024-01:')
print(retention_2024_01_one_year_appstore)

# Retention rates of one-year subscription users on Playstore, Cohort 2024-01
retention_2024_01_one_year_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_01_1_year_payment_gateway, '2025-01', 'playstore') /
playstore_user_of_cohort_2024_01_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Playstore in Cohort 2024-01:')
print(retention_2024_01_one_year_playstore)
```

```
Retention rate of one-year subscription users on Appstore in Cohort 2024-01:
[100, 39.56]
Retention rate of one-year subscription users on Playstore in Cohort 2024-01:
[100, 27.27]
```

```
# Calculate one-year cohort user counts for 2024-02 on 2 platforms
appstore_user_of_cohort_2024_02_one_year = len(cohort_2024_02['one_year'][cohort_2024_02['one_year']['payment_gateway'] ==
'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_02_one_year = len(cohort_2024_02['one_year'][cohort_2024_02['one_year']['payment_gateway'] ==
'playstore']['user_id'].drop_duplicates())

# Retention rates of one-year subscription users on Appstore, Cohort 2024-02
retention_2024_02_one_year_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_02_1_year_payment_gateway, '2025-02', 'appstore') /
appstore_user_of_cohort_2024_02_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Appstore in Cohort 2024-02:')
print(retention_2024_02_one_year_appstore)

# Retention rates of one-year subscription users on Playstore, Cohort 2024-02
retention_2024_02_one_year_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_02_1_year_payment_gateway, '2025-02', 'playstore') /
playstore_user_of_cohort_2024_02_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Playstore in Cohort 2024-02:')
print(retention_2024_02_one_year_playstore)
```

```
Retention rate of one-year subscription users on Appstore in Cohort 2024-02:
[100, 39.37]
Retention rate of one-year subscription users on Playstore in Cohort 2024-02:
[100, 30.7]
```

```
# Calculate one-year cohort user counts for 2024-03 on 2 platforms
appstore_user_of_cohort_2024_03_one_year = len(cohort_2024_03['one_year'][cohort_2024_03['one_year']['payment_gateway'] ==
'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_03_one_year = len(cohort_2024_03['one_year'][cohort_2024_03['one_year']['payment_gateway'] ==
'playstore']['user_id'].drop_duplicates())

# Retention rates of one-year subscription users on Appstore, Cohort 2024-03
retention_2024_03_one_year_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_03_1_year_payment_gateway, '2025-03', 'appstore') /
appstore_user_of_cohort_2024_03_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Appstore in Cohort 2024-03:')
print(retention_2024_03_one_year_appstore)

# Retention rates of one-year subscription users on Playstore, Cohort 2024-03
retention_2024_03_one_year_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_03_1_year_payment_gateway, '2025-03', 'playstore') /
playstore_user_of_cohort_2024_03_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Playstore in Cohort 2024-03:')
print(retention_2024_03_one_year_playstore)
```

```
Retention rate of one-year subscription users on Appstore in Cohort 2024-03:
[100, 40.99]
Retention rate of one-year subscription users on Playstore in Cohort 2024-03:
[100, 35.9]
```

```
# Calculate one-year cohort user counts for 2024-04 on 2 platforms
appstore_user_of_cohort_2024_04_one_year = len(cohort_2024_04['one_year'][cohort_2024_04['one_year']['payment_gateway'] ==
'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_04_one_year = len(cohort_2024_04['one_year'][cohort_2024_04['one_year']['payment_gateway'] ==
'playstore']['user_id'].drop_duplicates())

# Retention rates of one-year subscription users on Appstore, Cohort 2024-04
retention_2024_04_one_year_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_04_1_year_payment_gateway, '2025-04', 'appstore') /
appstore_user_of_cohort_2024_04_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Appstore in Cohort 2024-04:')
print(retention_2024_04_one_year_appstore)

# Retention rates of one-year subscription users on Playstore, Cohort 2024-04
retention_2024_04_one_year_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_04_1_year_payment_gateway, '2025-04', 'playstore') /
playstore_user_of_cohort_2024_04_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Playstore in Cohort 2024-04:')
print(retention_2024_04_one_year_playstore)
```

```
Retention rate of one-year subscription users on Appstore in Cohort 2024-04:
[100, 42.71]
Retention rate of one-year subscription users on Playstore in Cohort 2024-04:
[100, 30.67]
```

```
# Calculate one-year cohort user counts for 2024-05 on 2 platforms
appstore_user_of_cohort_2024_05_one_year = len(cohort_2024_05['one_year'][cohort_2024_05['one_year']['payment_gateway'] ==
'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_05_one_year = len(cohort_2024_05['one_year'][cohort_2024_05['one_year']['payment_gateway'] ==
'playstore']['user_id'].drop_duplicates())

# Retention rates of one-year subscription users on Appstore, Cohort 2024-05
retention_2024_05_one_year_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_05_1_year_payment_gateway, '2025-05', 'appstore') /
appstore_user_of_cohort_2024_05_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Appstore in Cohort 2024-05:')
print(retention_2024_05_one_year_appstore)

# Retention rates of one-year subscription users on Playstore, Cohort 2024-05
retention_2024_05_one_year_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_05_1_year_payment_gateway, '2025-05', 'playstore') /
playstore_user_of_cohort_2024_05_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Playstore in Cohort 2024-05:')
print(retention_2024_05_one_year_playstore)
```

```
Retention rate of one-year subscription users on Appstore in Cohort 2024-05:
[100, 36.96]
Retention rate of one-year subscription users on Playstore in Cohort 2024-05:
[100, 26.32]
```

```
# Calculate one-year cohort user counts for 2024-06 on 2 platforms
appstore_user_of_cohort_2024_06_one_year = len(cohort_2024_06['one_year'][cohort_2024_06['one_year']['payment_gateway'] ==
'appstore']['user_id'].drop_duplicates())
playstore_user_of_cohort_2024_06_one_year = len(cohort_2024_06['one_year'][cohort_2024_06['one_year']['payment_gateway'] ==
'playstore']['user_id'].drop_duplicates())

# Retention rates of one-year subscription users on Appstore, Cohort 2024-06
retention_2024_06_one_year_appstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_06_1_year_payment_gateway, '2025-06', 'appstore') /
appstore_user_of_cohort_2024_06_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Appstore in Cohort 2024-06:')
print(retention_2024_06_one_year_appstore)

# Retention rates of one-year subscription users on Playstore, Cohort 2024-06
retention_2024_06_one_year_playstore = [
    100,
    round(count_unique_users_by_gateway(renewals_2024_06_1_year_payment_gateway, '2025-06', 'playstore') /
playstore_user_of_cohort_2024_06_one_year * 100, 2)
]
print('Retention rate of one-year subscription users on Playstore in Cohort 2024-06:')
print(retention_2024_06_one_year_playstore)
```

```
Retention rate of one-year subscription users on Appstore in Cohort 2024-06:
[100, 37.89]
Retention rate of one-year subscription users on Playstore in Cohort 2024-06:
[100, 41.27]
```

```
# Appstore retention rates for one-year subscription users
appstore_retention_rates_1_year = pd.DataFrame({
  '2024-01': retention_2024_01_one_year_appstore,
  '2024-02': retention_2024_02_one_year_appstore,
  '2024-03': retention_2024_03_one_year_appstore,
  '2024-04': retention_2024_04_one_year_appstore,
  '2024-05': retention_2024_05_one_year_appstore,
  '2024-06': retention_2024_06_one_year_appstore
})

# Playstore retention rates for one-year subscription users
playstore_retention_rates_1_year = pd.DataFrame({
  '2024-01': retention_2024_01_one_year_playstore,
  '2024-02': retention_2024_02_one_year_playstore,
  '2024-03': retention_2024_03_one_year_playstore,
  '2024-04': retention_2024_04_one_year_playstore,
  '2024-05': retention_2024_05_one_year_playstore,
  '2024-06': retention_2024_06_one_year_playstore
})
```

▸ appstore_retention_rates_1_year: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]
▸ playstore_retention_rates_1_year: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

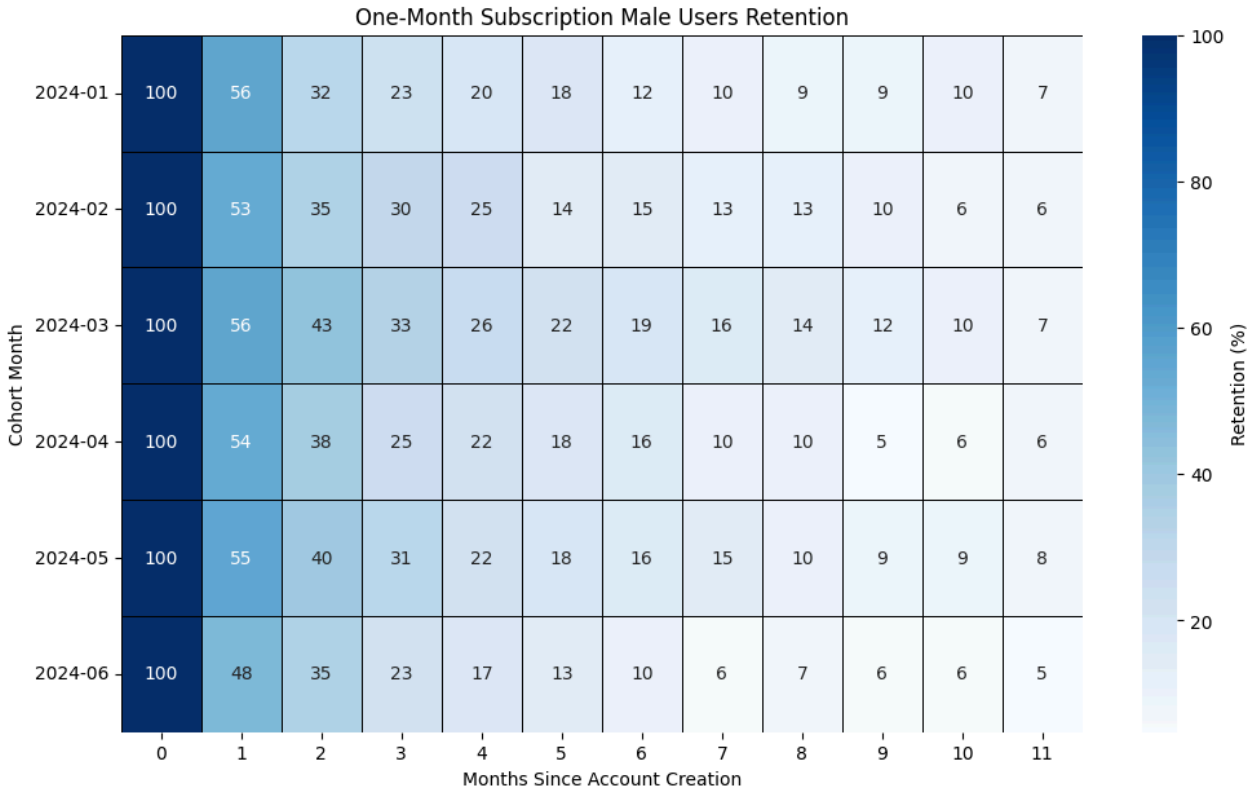
4.1.3. Retention rates visualisations

Retention rates will be visualized as a heatmap, showing user renewal patterns from the initial purchase month through to the twelfth month. This heatmap will clearly illustrate retention trends over the 12-month period.

4.1.3.1. One-month retention rates by genders

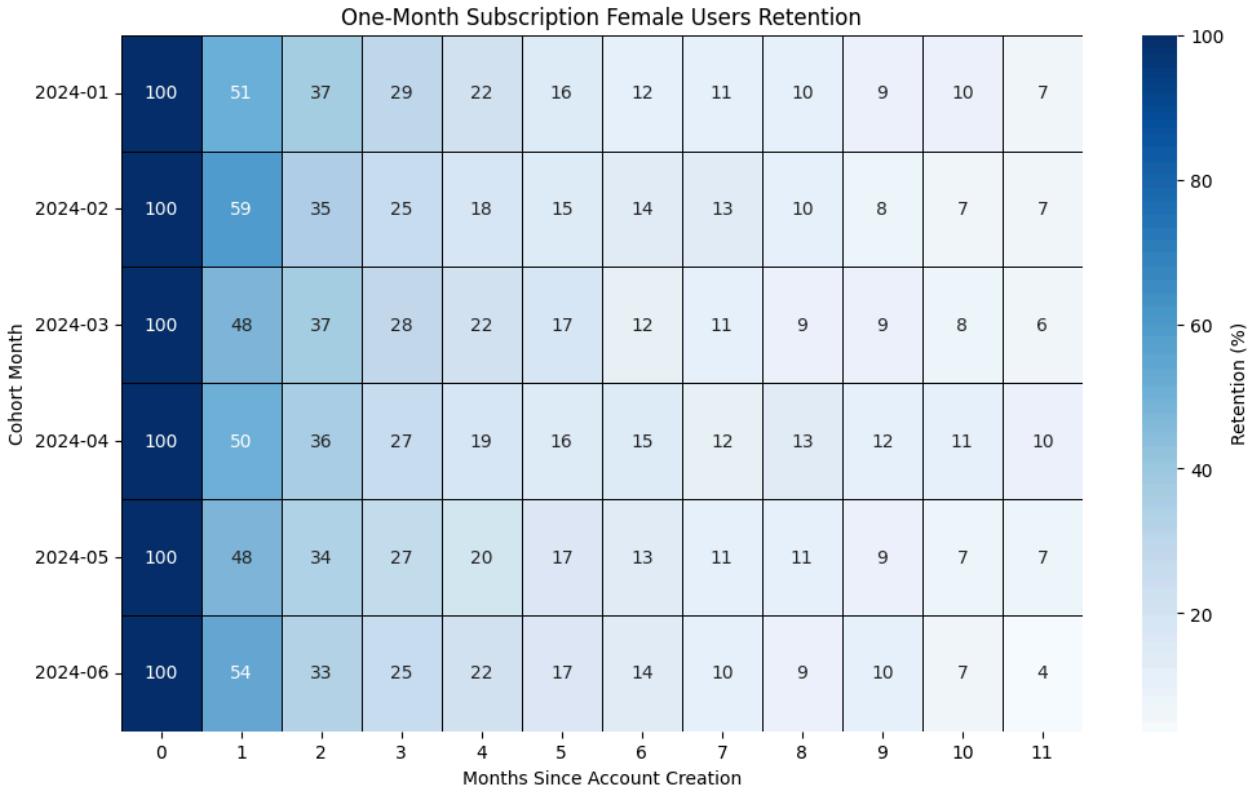
```
# Transpose so cohorts are rows and months are columns
retentions_male_t = retentions_male.T

plt.figure(figsize=(12, 7))
sns.heatmap(retentions_male_t, annot=True, fmt='.0f', cmap='Blues', cbar_kws={'label': 'Retention (%)'}, linewidths=0.5,
linecolor='black')

plt.title('One-Month Subscription Male Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ retentions_male_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 10 more fields]



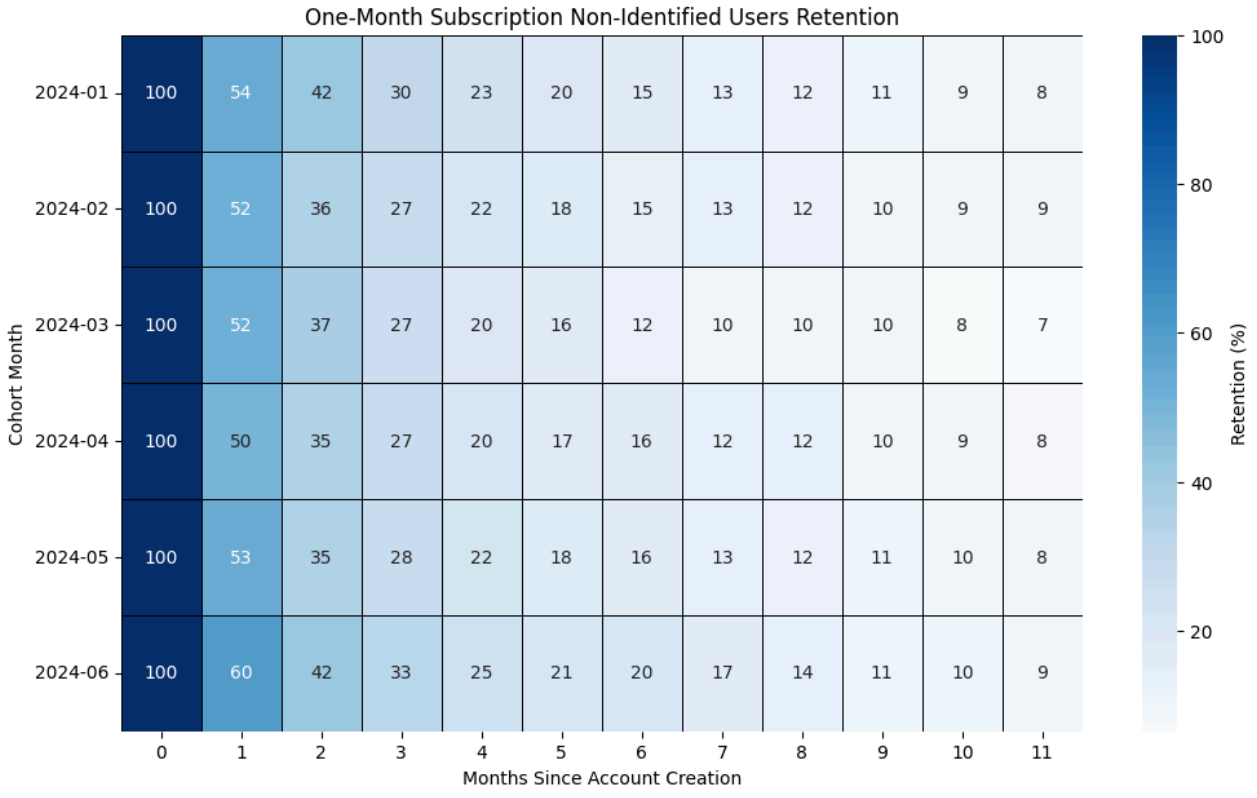
```
retentions_female_t = retentions_female.T

plt.figure(figsize=(12, 7))
sns.heatmap(retentions_female_t, annot=True, fmt='.0f', cmap='Blues', cbar_kws={'label': 'Retention (%)'}, linewidths=0.5,
linecolor='black')

plt.title('One-Month Subscription Female Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

retentions_female_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 10 more fields]



```
retentions_non_identified_t = retentions_non_identified.T

plt.figure(figsize=(12, 7))
sns.heatmap(retentions_non_identified_t, annot=True, fmt='.0f', cmap='Blues', cbar_kws={'label': 'Retention (%)'},
linewidths=0.5, linecolor='black')

plt.title('One-Month Subscription Non-Identified Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

retentions_non_identified_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 10 more fields]



The retention rates of one-month male users are higher in the first month comparing to the others. However, the retention rates

among genders reduce gradually over time.

4.1.3.2. Three-month retention rates by genders

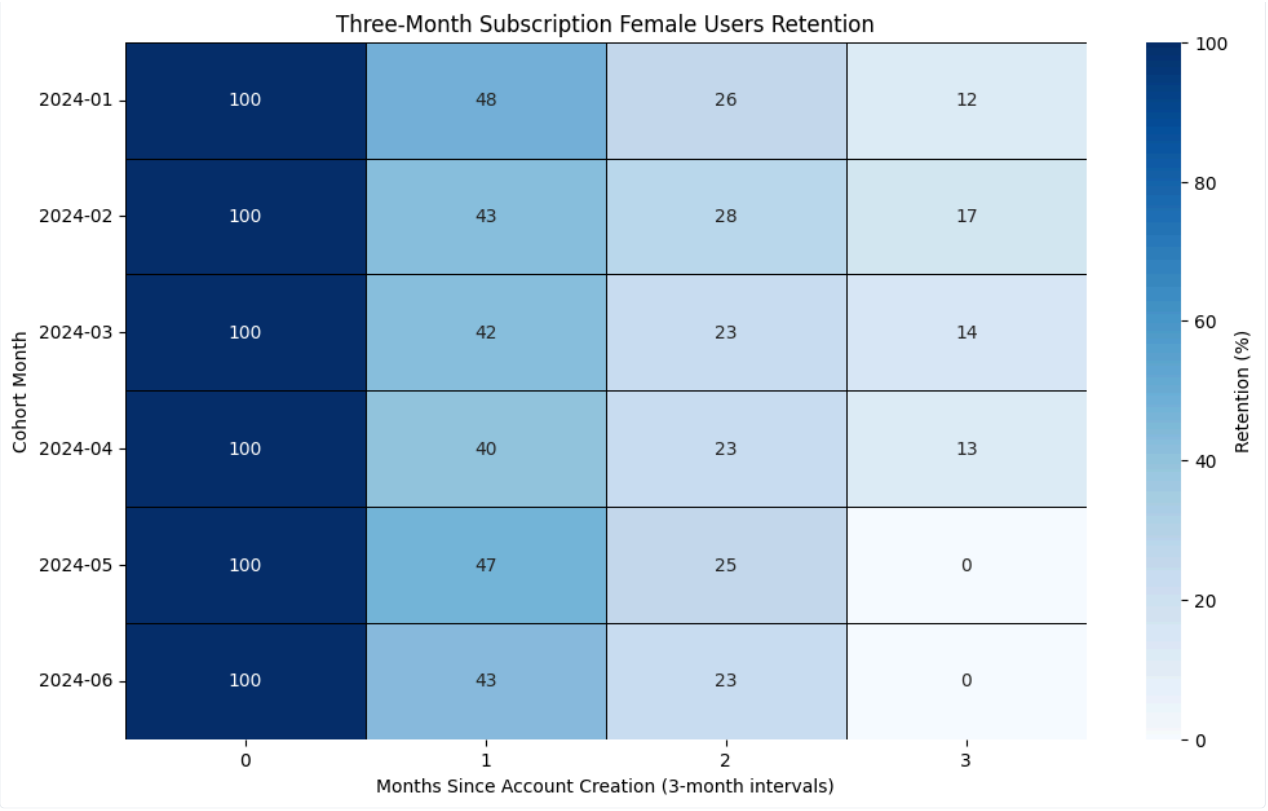
```
male_retention_rate_three_month_t = male_retention_rate_three_month.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    male_retention_rate_three_month_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('Three-Month Subscription Male Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (3-month intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ male_retention_rate_three_month_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 2 more fields]


Three-Month Subscription Male Users Retention
Cohort Month
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
100
46
24
16
100
37
12
8
100
38
25
11
100
48
24
17
100
42
22
0
100
44
26
0
0
1
2
3
Months Since Account Creation (3-month intervals)
Retention (%)
100
80
60
40
20
0

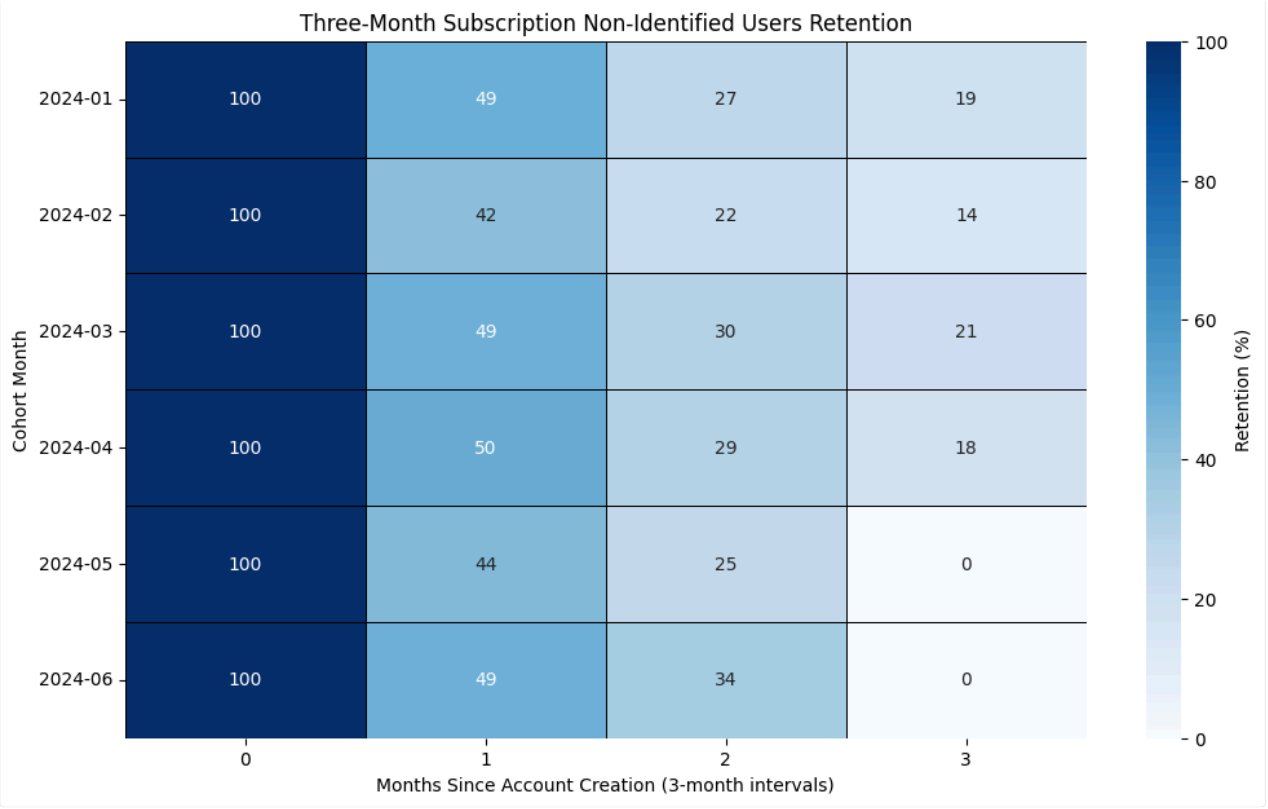
```
female_retention_rate_three_month_t = female_retention_rate_three_month.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    female_retention_rate_three_month_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('Three-Month Subscription Female Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (3-month intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ female_retention_rate_three_month_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 2 more fields]


Three-Month Subscription Female Users Retention
Cohort Month
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
100
48
26
12
100
43
28
17
100
42
23
14
100
40
23
13
100
47
25
0
100
43
23
0
0
1
2
3
Months Since Account Creation (3-month intervals)
Retention (%)
100
80
60
40
20
0

```
non_identified_retention_rate_three_month_t = non_identified_retention_rate_three_month.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    non_identified_retention_rate_three_month_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('Three-Month Subscription Non-Identified Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (3-month intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ non_identified_retention_rate_three_month_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 2 more fields]

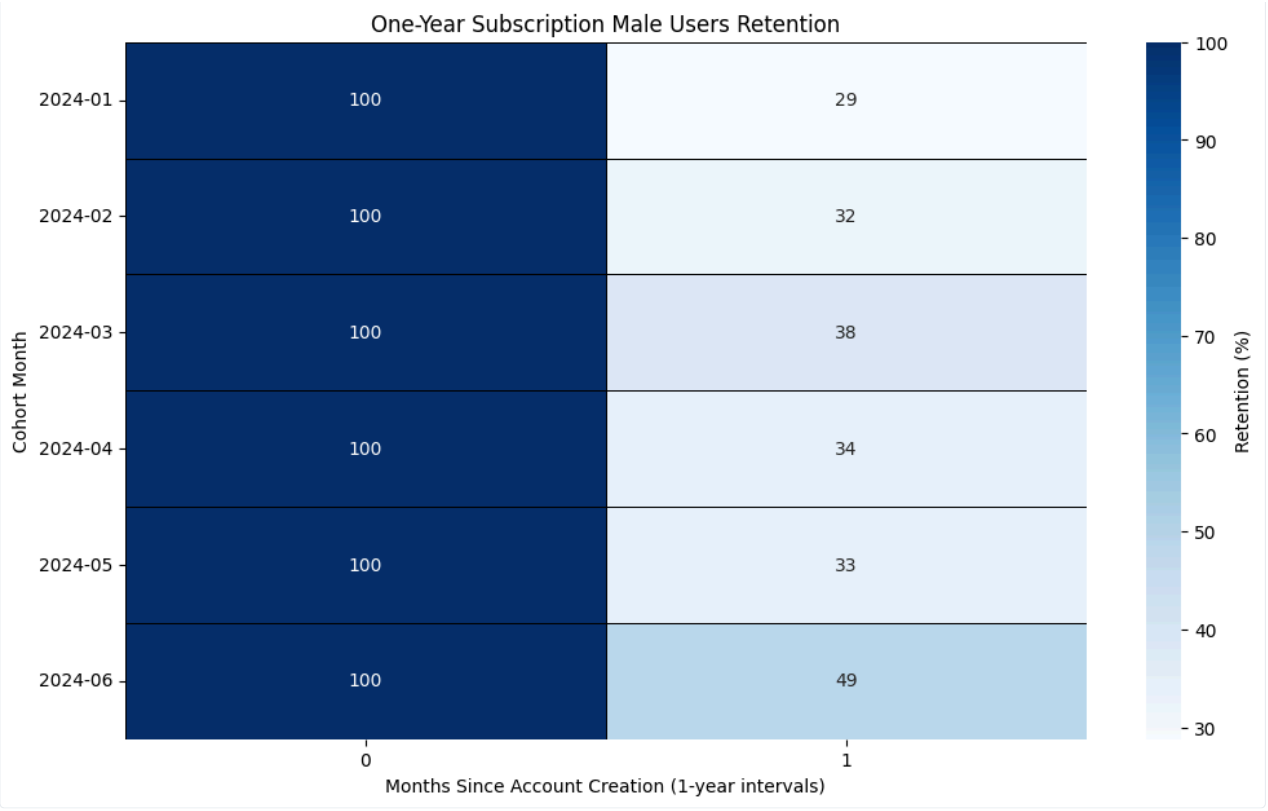


There are significant variations in first-month retention rates for three-month subscription male users across different cohorts, while female and non-identified users display more stable retention trends. Additionally, for all genders, there are no users remaining at the third renewal point in cohorts 2024-05 and 2024-06.

4.1.3.3 - One-year retention rates by genders

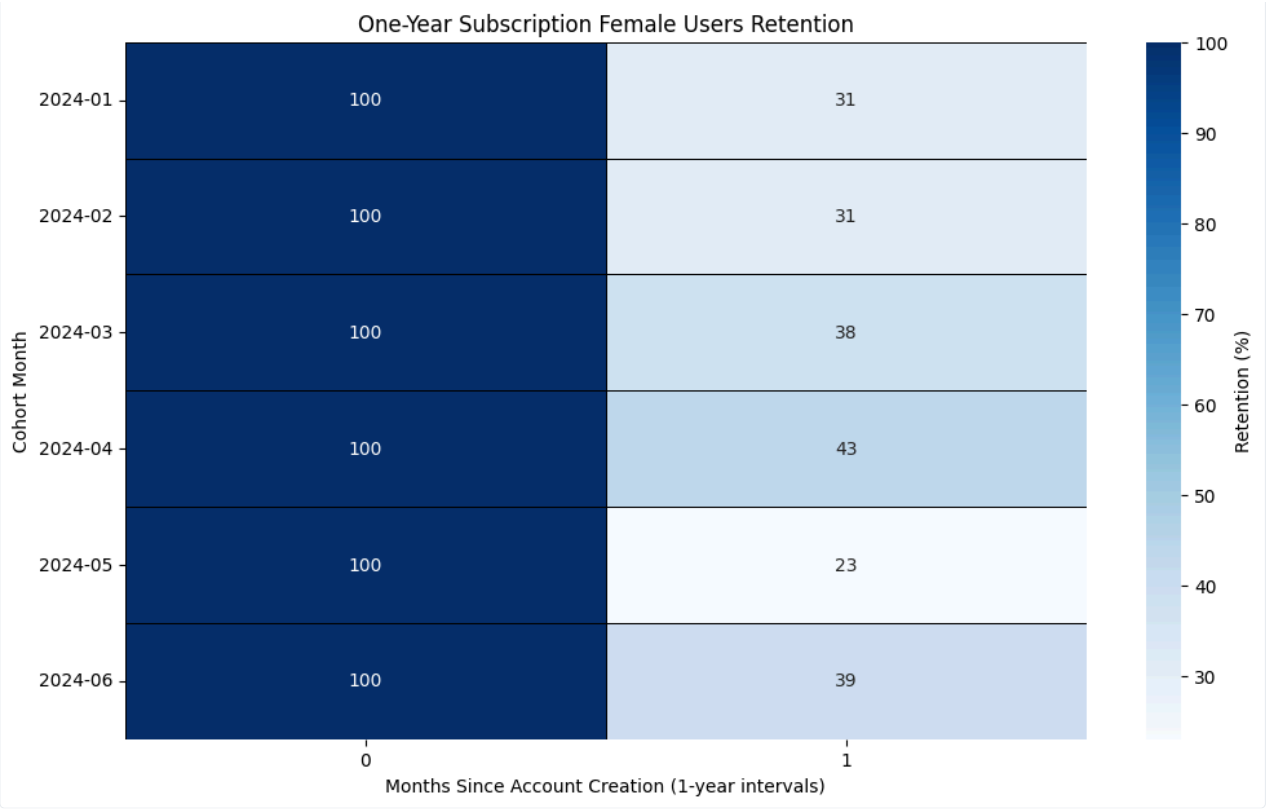
```
male_retention_rate_one_year_t = male_retention_rate_one_year.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    male_retention_rate_one_year_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('One-Year Subscription Male Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (1-year intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ male_retention_rate_one_year_t: pandas.core.frame.DataFrame = [0: float64, 1: float64]


One-Year Subscription Male Users Retention
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
Cohort Month
100
29
100
32
100
38
100
34
100
33
100
49
0
1
Months Since Account Creation (1-year intervals)
Retention (%)
100
90
80
70
60
50
40
30

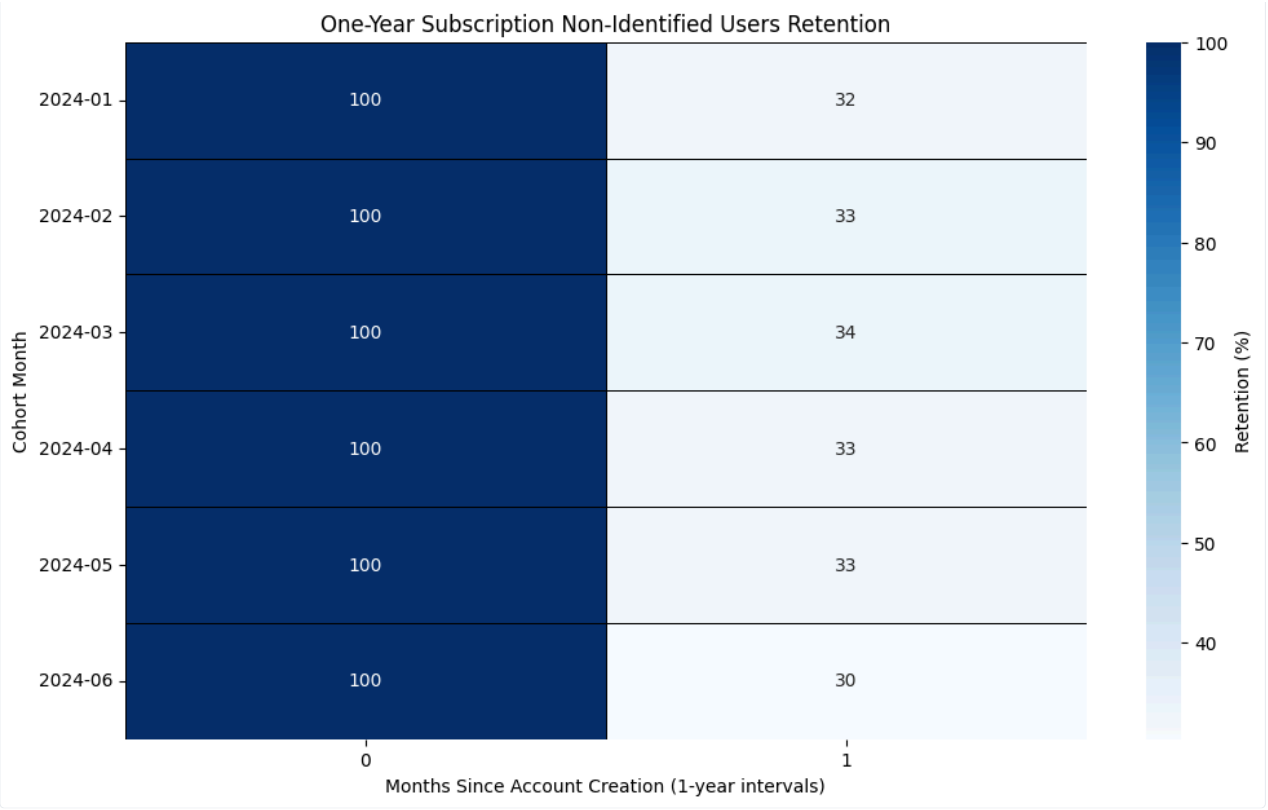
```
female_retention_rate_one_year_t = female_retention_rate_one_year.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    female_retention_rate_one_year_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('One-Year Subscription Female Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (1-year intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ female_retention_rate_one_year_t: pandas.core.frame.DataFrame = [0: float64, 1: float64]


One-Year Subscription Female Users Retention
Cohort Month
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
100
31
100
31
100
38
100
43
100
23
100
39
0
1
Months Since Account Creation (1-year intervals)
Retention (%)
100
90
80
70
60
50
40
30

```
non_identified_retention_rate_one_year_t = non_identified_retention_rate_one_year.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    non_identified_retention_rate_one_year_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('One-Year Subscription Non-Identified Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (1-year intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ non_identified_retention_rate_one_year_t: pandas.core.frame.DataFrame = [0: float64, 1: float64]

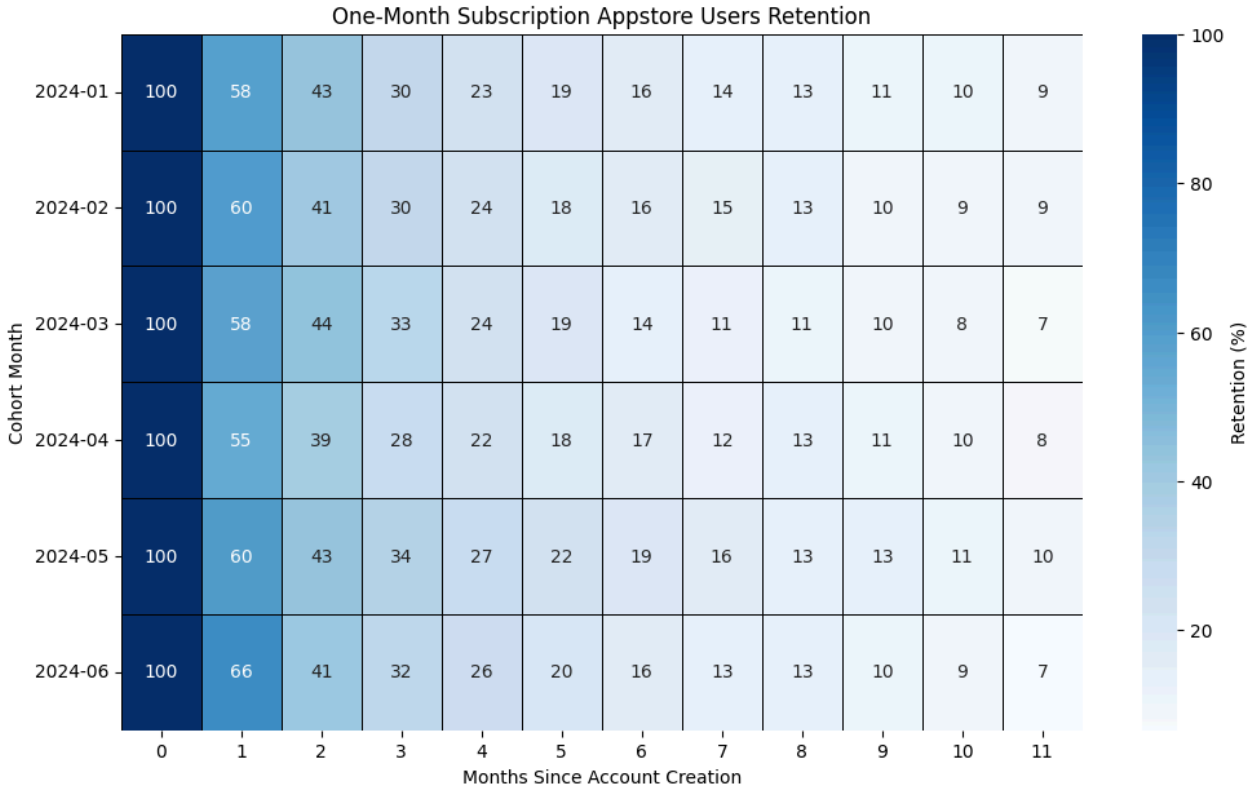


The renewal rates of one-year subscriptions for female and male users are highly variable across cohorts, with notable changes like the drop to 23% for females in cohort 2024-05 and a peak of 49% for males in cohort 2024-06. In contrast, non-identified users demonstrate a very stable renewal trend (around 33%) regardless of cohort.

4.1.3.4. One-month retention rates by platforms

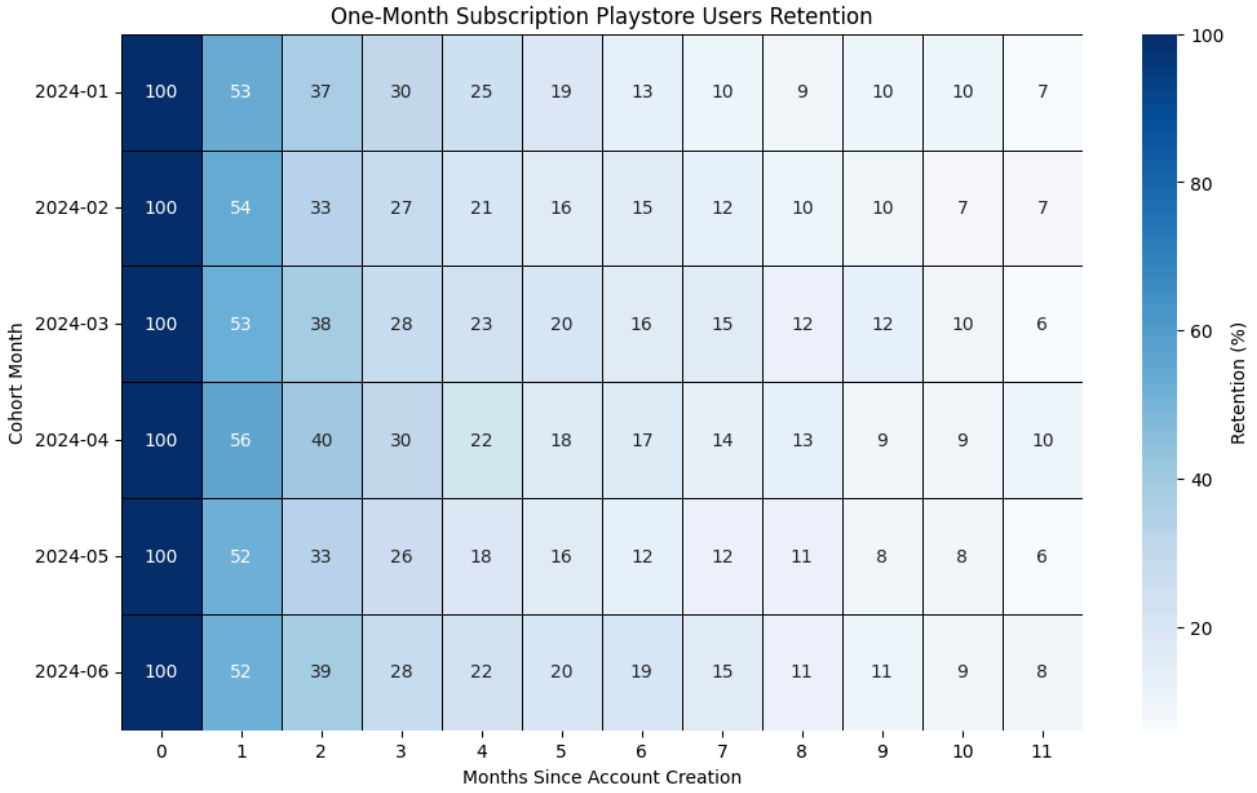
```
appstore_retention_rates_1_month_t = appstore_retention_rates_1_month.T

plt.figure(figsize=(12, 7))
sns.heatmap(appstore_retention_rates_1_month_t, annot=True, fmt='.0f', cmap='Blues', cbar_kws={'label': 'Retention (%)'},
linewidths=0.5, linecolor='black')

plt.title('One-Month Subscription Appstore Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ appstore_retention_rates_1_month_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 10 more fields]



```
playstore_retention_rates_1_month_t = playstore_retention_rates_1_month.T

plt.figure(figsize=(12, 7))
sns.heatmap(playstore_retention_rates_1_month_t, annot=True, fmt='.0f', cmap='Blues', cbar_kws={'label': 'Retention (%)'},
linewidths=0.5, linecolor='black')

plt.title('One-Month Subscription Playstore Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ playstore_retention_rates_1_month_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 10 more fields]

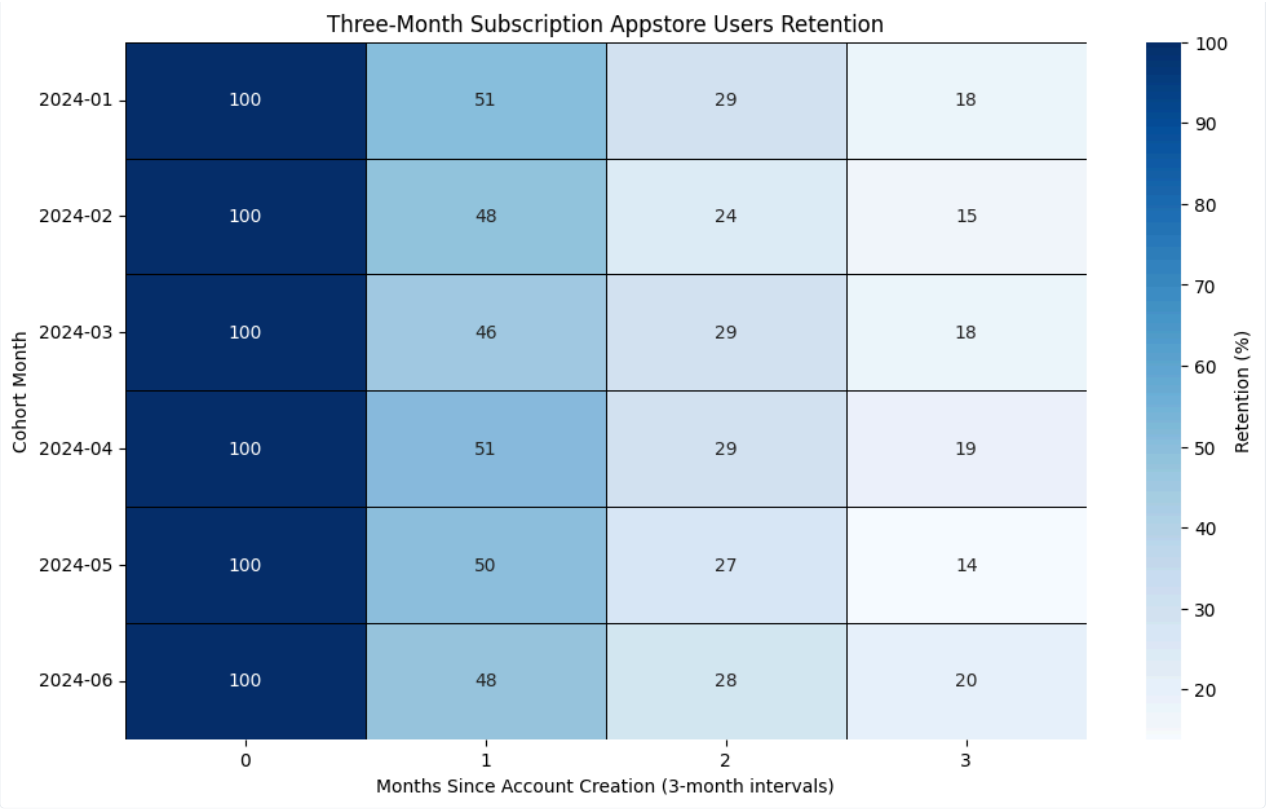


Retention rates for one-month subscription users are consistently higher for App Store users compared to Play Store users across all months, with significant differences of around 10%, such as in month 1 for the 2024-06 and 2024-05 cohorts. After the first month, both platforms show a gradual decline in retention over time, with the differences between them becoming less pronounced.

4.1.3.5. Three-month retention rates by platforms

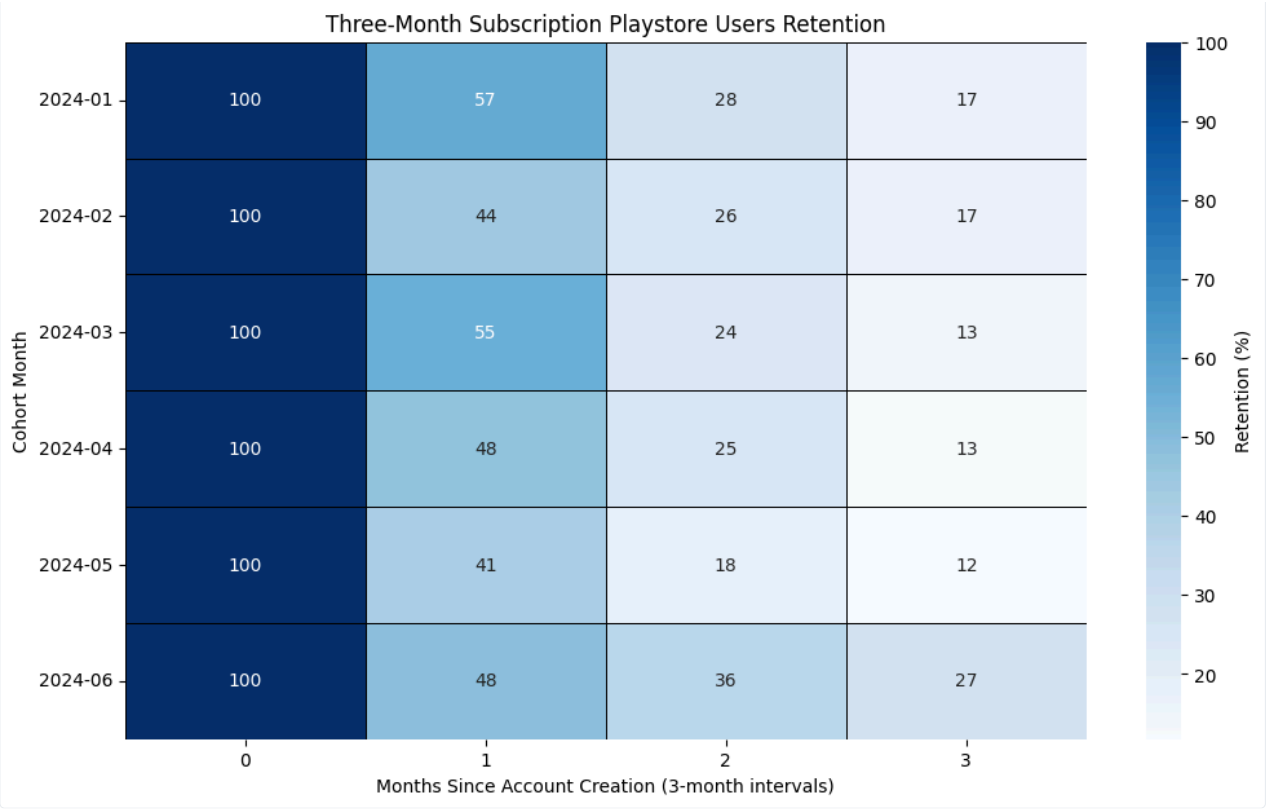
```
appstore_retention_rates_3_month_t = appstore_retention_rates_3_month.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    appstore_retention_rates_3_month_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('Three-Month Subscription Appstore Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (3-month intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ appstore_retention_rates_3_month_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 2 more fields]


Three-Month Subscription Appstore Users Retention
Cohort Month
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
100
51
29
18
100
48
24
15
100
46
29
18
100
51
29
19
100
50
27
14
100
48
28
20
0
1
2
3
Months Since Account Creation (3-month intervals)
Retention (%)
100
90
80
70
60
50
40
30
20

```
playstore_retention_rates_3_month_t = playstore_retention_rates_3_month.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    playstore_retention_rates_3_month_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('Three-Month Subscription Playstore Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (3-month intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ playstore_retention_rates_3_month_t: pandas.core.frame.DataFrame = [0: float64, 1: float64 ... 2 more fields]

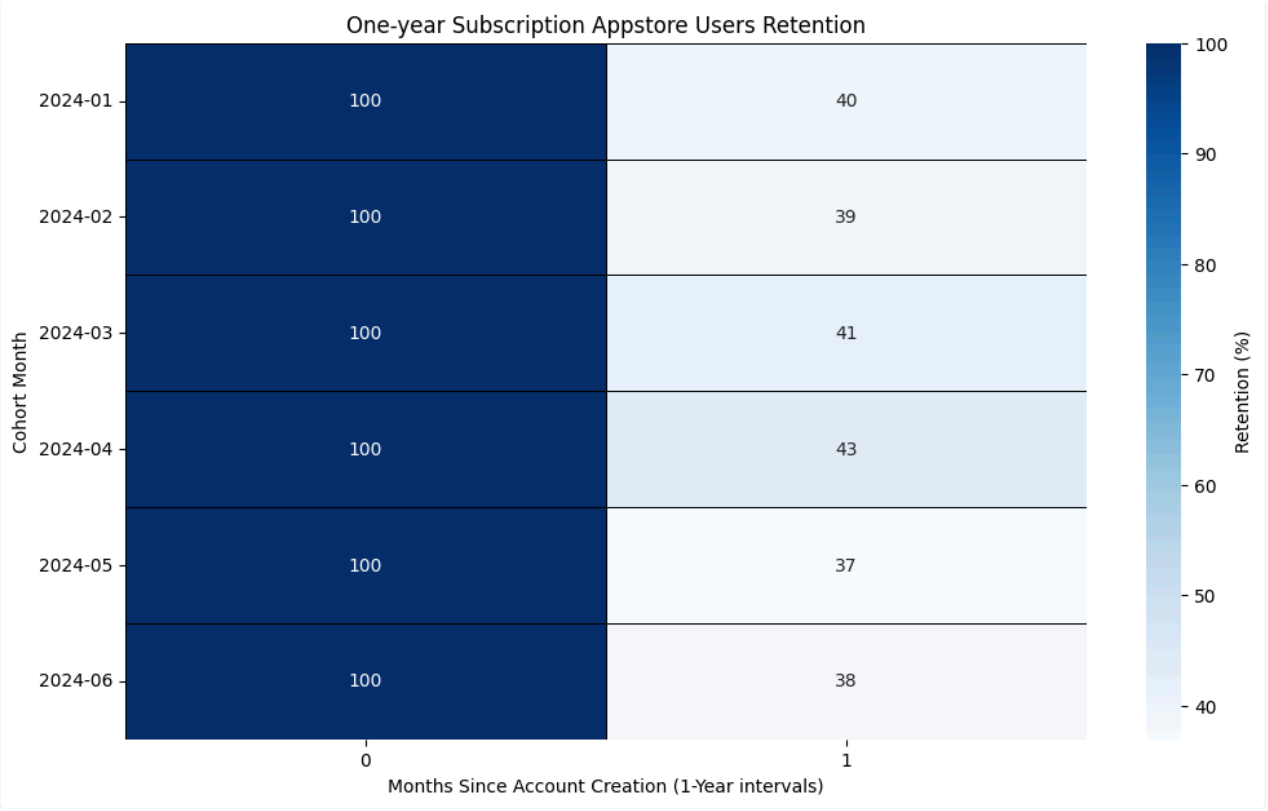


The retention rates for three-month subscription users on the App Store remain stable across different cohorts and over time, indicating consistent user engagement. In contrast, the Play Store displays noticeable fluctuations in retention trends, suggesting varying levels of user loyalty and engagement from cohort to cohort. This highlights the App Store's stronger consistency in retaining three-month subscribers compared to the Play Store.

4.1.3.6. One-year retention rates by platforms

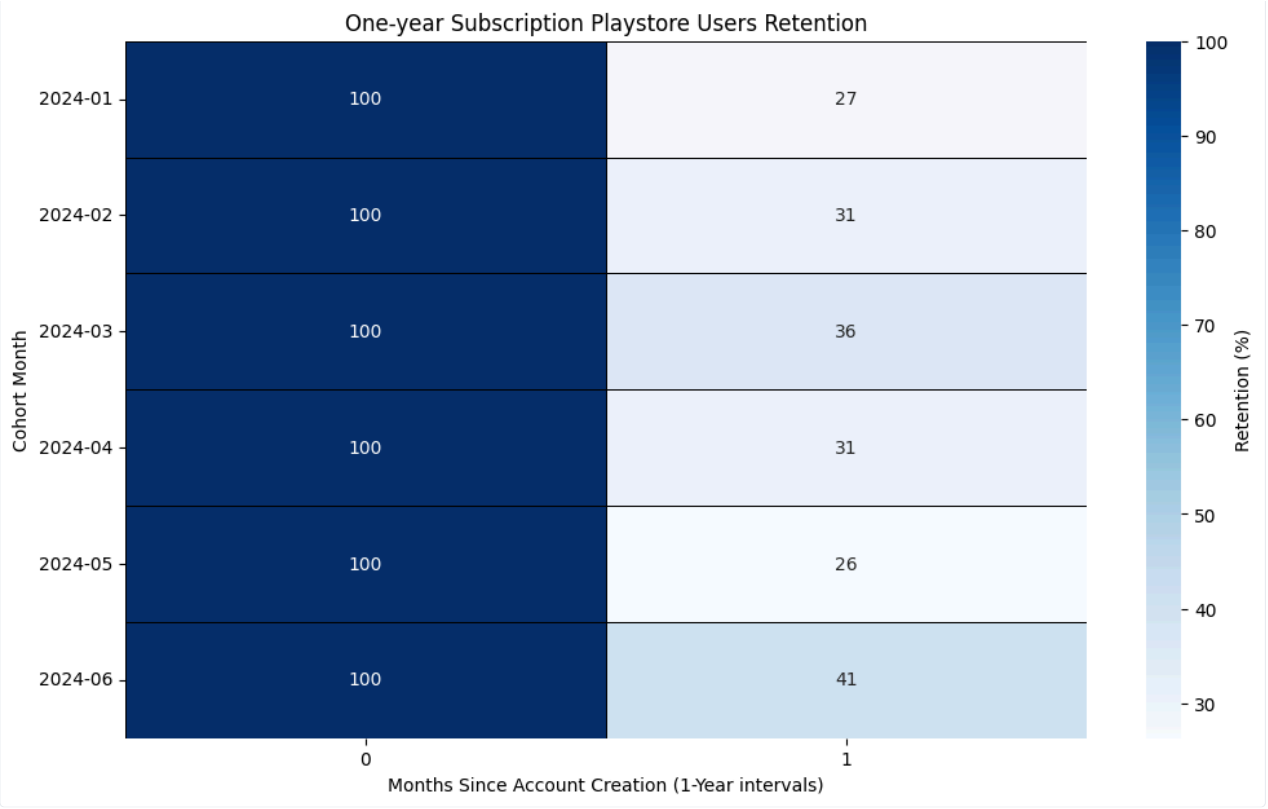
```
appstore_retention_rates_1_year_t = appstore_retention_rates_1_year.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    appstore_retention_rates_1_year_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('One-year Subscription Appstore Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (1-Year intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ appstore_retention_rates_1_year_t: pandas.core.frame.DataFrame = [0: float64, 1: float64]


One-year Subscription Appstore Users Retention
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
100
40
100
39
100
41
100
43
100
37
100
38
Cohort Month
0
1
Months Since Account Creation (1-Year intervals)
Retention (%)
100
90
80
70
60
50
40

```
playstore_retention_rates_1_year_t = playstore_retention_rates_1_year.T

plt.figure(figsize=(12, 7))
sns.heatmap(
    playstore_retention_rates_1_year_t,
    annot=True,
    fmt='.0f',
    cmap='Blues',
    cbar_kws={'label': 'Retention (%)'},
    linewidths=0.5,
    linecolor='black'
)

plt.title('One-year Subscription Playstore Users Retention')
plt.ylabel('Cohort Month')
plt.xlabel('Months Since Account Creation (1-Year intervals)')
plt.xticks(rotation=0)
plt.yticks(rotation=0)

plt.show()
```

▸ playstore_retention_rates_1_year_t: pandas.core.frame.DataFrame = [0: float64, 1: float64]



The retention rates for one-year subscription users are clearly higher for the App Store than for the Play Store after the first renewal period as well as a stable percentage of renewal users across all cohorts.

4.1.4. Retention rates conclusion

After reviewing retention rates by gender and platform, both approaches reveal some inconsistent trends and notable gaps. However, further analysis is recommended to focus on platforms because platform-based retention rates exhibit smaller gaps between cohorts within each user type, while still highlighting clear differences between the two main platforms: App Store and Play Store. This recommendation will give business managers clearer insights for identifying issues and devising effective solutions. Subsequent calucaltions such as total revenue and average revenue per user will also be performed with a platform-based focus.

4.2. Total revenue of each cohort by platforms

Total revenue for a cohort is typically calculated as the sum of income from the initial subscription purchase plus all subsequent renewal fees paid by users in that cohort

4.2.1. One-month total revenue

```
# Total revenue 2024-01 1-month on Appstore
appstore_revenue_2024_01_1_month = (
  renewals_2024_01_1_month_platform['2024-02']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-03']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-04']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-05']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-06']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-07']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-08']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-09']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-10']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-11']['appstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-12']['appstore'].amount.sum()
  + cohort_2024_01['one_month'][cohort_2024_01['one_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-01 1-month on Appstore: {appstore_revenue_2024_01_1_month:,} VND')

# Total revenue 2024-01 1-month on Playstore
playstore_revenue_2024_01_1_month = (
  renewals_2024_01_1_month_platform['2024-02']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-03']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-04']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-05']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-06']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-07']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-08']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-09']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-10']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-11']['playstore'].amount.sum()
  + renewals_2024_01_1_month_platform['2024-12']['playstore'].amount.sum()
  + cohort_2024_01['one_month'][cohort_2024_01['one_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-01 1-month on playstore: {playstore_revenue_2024_01_1_month:,} VND')
```

```
Total revenue 2024-01 1-month on Appstore: 204,077,000 VND
Total revenue 2024-01 1-month on playstore: 105,086,000 VND
```

```
# Total revenue 2024-02 1-month on Appstore
appstore_revenue_2024_02_1_month = (
    renewals_2024_02_1_month_platform['2024-03']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-04']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-05']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-06']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-07']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-08']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-09']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-10']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-11']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-12']['appstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2025-01']['appstore'].amount.sum()
    + cohort_2024_02['one_month'][cohort_2024_02['one_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-02 1-month on Appstore: {appstore_revenue_2024_02_1_month:,} VND')


# Total revenue 2024-02 1-month on Playstore
playstore_revenue_2024_02_1_month = (
    renewals_2024_02_1_month_platform['2024-03']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-04']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-05']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-06']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-07']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-08']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-09']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-10']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-11']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2024-12']['playstore'].amount.sum()
    + renewals_2024_02_1_month_platform['2025-01']['playstore'].amount.sum()
    + cohort_2024_02['one_month'][cohort_2024_02['one_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-02 1-month on Playstore: {playstore_revenue_2024_02_1_month:,} VND')
```

```
Total revenue 2024-02 1-month on Appstore: 157,155,000 VND
Total revenue 2024-02 1-month on Playstore: 82,905,000 VND
```

```
# Total revenue 2024-03 1-month on Appstore
appstore_revenue_2024_03_1_month = (
    renewals_2024_03_1_month_platform['2024-04']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-05']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-06']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-07']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-08']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-09']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-10']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-11']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-12']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2025-01']['appstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2025-02']['appstore'].amount.sum()
    + cohort_2024_03['one_month'][cohort_2024_03['one_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-03 1-month on Appstore: {appstore_revenue_2024_03_1_month:,} VND')


# Total revenue 2024-03 1-month on Playstore
playstore_revenue_2024_03_1_month = (
    renewals_2024_03_1_month_platform['2024-04']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-05']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-06']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-07']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-08']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-09']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-10']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-11']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2024-12']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2025-01']['playstore'].amount.sum()
    + renewals_2024_03_1_month_platform['2025-02']['playstore'].amount.sum()
    + cohort_2024_03['one_month'][cohort_2024_03['one_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-03 1-month on Playstore: {playstore_revenue_2024_03_1_month:,} VND')
```

```
Total revenue 2024-03 1-month on Appstore: 154,585,000 VND
Total revenue 2024-03 1-month on Playstore: 81,425,000 VND
```

```
# Total revenue 2024-04 1-month on Appstore
appstore_revenue_2024_04_1_month = (
    renewals_2024_04_1_month_platform['2024-05']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-06']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-07']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-08']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-09']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-10']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-11']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-12']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2025-01']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2025-02']['appstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2025-03']['appstore'].amount.sum()
    + cohort_2024_04['one_month'][cohort_2024_04['one_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-04 1-month on Appstore: {appstore_revenue_2024_04_1_month:,} VND')


# Total revenue 2024-04 1-month on Playstore
playstore_revenue_2024_04_1_month = (
    renewals_2024_04_1_month_platform['2024-05']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-06']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-07']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-08']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-09']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-10']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-11']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2024-12']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2025-01']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2025-02']['playstore'].amount.sum()
    + renewals_2024_04_1_month_platform['2025-03']['playstore'].amount.sum()
    + cohort_2024_04['one_month'][cohort_2024_04['one_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-04 1-month on Playstore: {playstore_revenue_2024_04_1_month:,} VND')
```

```
Total revenue 2024-04 1-month on Appstore: 126,071,000 VND
Total revenue 2024-04 1-month on Playstore: 68,356,000 VND
```

```
# Total revenue 2024-05 1-month on Appstore
appstore_revenue_2024_05_1_month = (
    renewals_2024_05_1_month_platform['2024-06']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-07']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-08']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-09']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-10']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-11']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-12']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-01']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-02']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-03']['appstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-04']['appstore'].amount.sum()
    + cohort_2024_05['one_month'][cohort_2024_05['one_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-05 1-month on Appstore: {appstore_revenue_2024_05_1_month:,} VND')


# Total revenue 2024-05 1-month on Playstore
playstore_revenue_2024_05_1_month = (
    renewals_2024_05_1_month_platform['2024-06']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-07']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-08']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-09']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-10']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-11']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2024-12']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-01']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-02']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-03']['playstore'].amount.sum()
    + renewals_2024_05_1_month_platform['2025-04']['playstore'].amount.sum()
    + cohort_2024_05['one_month'][cohort_2024_05['one_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-05 1-month on Playstore: {playstore_revenue_2024_05_1_month:,} VND')
```

```
Total revenue 2024-05 1-month on Appstore: 137,902,000 VND
Total revenue 2024-05 1-month on Playstore: 70,632,000 VND
```

```
# Total revenue 2024-06 1-month on Appstore
appstore_revenue_2024_06_1_month = (
    renewals_2024_06_1_month_platform['2024-07']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-08']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-09']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-10']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-11']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-12']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-01']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-02']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-03']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-04']['appstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-05']['appstore'].amount.sum()
    + cohort_2024_06['one_month'][cohort_2024_06['one_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-06 1-month on Appstore: {appstore_revenue_2024_06_1_month:,} VND')


# Total revenue 2024-06 1-month on Playstore
playstore_revenue_2024_06_1_month = (
    renewals_2024_06_1_month_platform['2024-07']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-08']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-09']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-10']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-11']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2024-12']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-01']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-02']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-03']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-04']['playstore'].amount.sum()
    + renewals_2024_06_1_month_platform['2025-05']['playstore'].amount.sum()
    + cohort_2024_06['one_month'][cohort_2024_06['one_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-06 1-month on Playstore: {playstore_revenue_2024_06_1_month:,} VND')
```

```
Total revenue 2024-06 1-month on Appstore: 111,322,000 VND
Total revenue 2024-06 1-month on Playstore: 59,449,000 VND
```

4.2.2. Three-month total revenue

```
# Total revenue 2024-01 3-month on Appstore
appstore_revenue_2024_01_3_month = (
    renewals_2024_01_3_month_payment_gateway['2024-04']['appstore'].amount.sum()
    + renewals_2024_01_3_month_payment_gateway['2024-07']['appstore'].amount.sum()
    + renewals_2024_01_3_month_payment_gateway['2024-10']['appstore'].amount.sum()
    + cohort_2024_01['three_month'][cohort_2024_01['three_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-01 3-month on Appstore: {appstore_revenue_2024_01_3_month:,} VND')

# Total revenue 2024-01 3-month on Playstore
playstore_revenue_2024_01_3_month = (
    renewals_2024_01_3_month_payment_gateway['2024-04']['playstore'].amount.sum()
    + renewals_2024_01_3_month_payment_gateway['2024-07']['playstore'].amount.sum()
    + renewals_2024_01_3_month_payment_gateway['2024-10']['playstore'].amount.sum()
    + cohort_2024_01['three_month'][cohort_2024_01['three_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-01 3-month on Playstore: {playstore_revenue_2024_01_3_month:,} VND')
```

```
Total revenue 2024-01 3-month on Appstore: 82,079,000 VND
Total revenue 2024-01 3-month on Playstore: 28,379,000 VND
```

```
# Total revenue 2024-02 3-month on Appstore
appstore_revenue_2024_02_3_month = (
    renewals_2024_02_3_month_payment_gateway['2024-05']['appstore'].amount.sum()
    + renewals_2024_02_3_month_payment_gateway['2024-08']['appstore'].amount.sum()
    + renewals_2024_02_3_month_payment_gateway['2024-11']['appstore'].amount.sum()
    + cohort_2024_02['three_month'][cohort_2024_02['three_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-02 3-month on Appstore: {appstore_revenue_2024_02_3_month:,} VND')


# Total revenue 2024-02 3-month on Playstore
playstore_revenue_2024_02_3_month = (
    renewals_2024_02_3_month_payment_gateway['2024-05']['playstore'].amount.sum()
    + renewals_2024_02_3_month_payment_gateway['2024-08']['playstore'].amount.sum()
    + renewals_2024_02_3_month_payment_gateway['2024-11']['playstore'].amount.sum()
    + cohort_2024_02['three_month'][cohort_2024_02['three_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-02 3-month on Playstore: {playstore_revenue_2024_02_3_month:,} VND')
```

```
Total revenue 2024-02 3-month on Appstore: 136,040,000 VND
Total revenue 2024-02 3-month on Playstore: 31,514,000 VND
```

```
# Total revenue 2024-03 3-month on Appstore
appstore_revenue_2024_03_3_month = (
    renewals_2024_03_3_month_payment_gateway['2024-06']['appstore'].amount.sum()
    + renewals_2024_03_3_month_payment_gateway['2024-09']['appstore'].amount.sum()
    + renewals_2024_03_3_month_payment_gateway['2024-12']['appstore'].amount.sum()
    + cohort_2024_03['three_month'][cohort_2024_03['three_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-03 3-month on Appstore: {appstore_revenue_2024_03_3_month:,} VND')

# Total revenue 2024-03 3-month on Playstore
playstore_revenue_2024_03_3_month = (
    renewals_2024_03_3_month_payment_gateway['2024-06']['playstore'].amount.sum()
    + renewals_2024_03_3_month_payment_gateway['2024-09']['playstore'].amount.sum()
    + renewals_2024_03_3_month_payment_gateway['2024-12']['playstore'].amount.sum()
    + cohort_2024_03['three_month'][cohort_2024_03['three_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-03 3-month on Playstore: {playstore_revenue_2024_03_3_month:,} VND')
```

```
Total revenue 2024-03 3-month on Appstore: 133,238,000 VND
Total revenue 2024-03 3-month on Playstore: 17,177,000 VND
```

```
# Total revenue 2024-04 3-month on Appstore
appstore_revenue_2024_04_3_month = (
    renewals_2024_04_3_month_payment_gateway['2024-07']['appstore'].amount.sum()
    + renewals_2024_04_3_month_payment_gateway['2024-10']['appstore'].amount.sum()
    + renewals_2024_04_3_month_payment_gateway['2025-01']['appstore'].amount.sum()
    + cohort_2024_04['three_month'][cohort_2024_04['three_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-04 3-month on Appstore: {appstore_revenue_2024_04_3_month:,} VND')

# Total revenue 2024-04 3-month on Playstore
playstore_revenue_2024_04_3_month = (
    renewals_2024_04_3_month_payment_gateway['2024-07']['playstore'].amount.sum()
    + renewals_2024_04_3_month_payment_gateway['2024-10']['playstore'].amount.sum()
    + renewals_2024_04_3_month_payment_gateway['2025-01']['playstore'].amount.sum()
    + cohort_2024_04['three_month'][cohort_2024_04['three_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-04 3-month on Playstore: {playstore_revenue_2024_04_3_month:,} VND')
```

```
Total revenue 2024-04 3-month on Appstore: 114,381,000 VND
Total revenue 2024-04 3-month on Playstore: 28,602,000 VND
```

```
# Total revenue 2024-05 3-month on Appstore
appstore_revenue_2024_05_3_month = (
    renewals_2024_05_3_month_payment_gateway['2024-08']['appstore'].amount.sum()
    + renewals_2024_05_3_month_payment_gateway['2024-11']['appstore'].amount.sum()
    + renewals_2024_05_3_month_payment_gateway['2025-02']['appstore'].amount.sum()
    + cohort_2024_05['three_month'][cohort_2024_05['three_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-05 3-month on Appstore: {appstore_revenue_2024_05_3_month:,} VND')

# Total revenue 2024-05 3-month on Playstore
playstore_revenue_2024_05_3_month = (
    renewals_2024_05_3_month_payment_gateway['2024-08']['playstore'].amount.sum()
    + renewals_2024_05_3_month_payment_gateway['2024-11']['playstore'].amount.sum()
    + renewals_2024_05_3_month_payment_gateway['2025-02']['playstore'].amount.sum()
    + cohort_2024_05['three_month'][cohort_2024_05['three_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-05 3-month on Playstore: {playstore_revenue_2024_05_3_month:,} VND')
```

```
Total revenue 2024-05 3-month on Appstore: 102,305,000 VND
Total revenue 2024-05 3-month on Playstore: 13,492,000 VND
```

```
# Total revenue 2024-06 3-month on Appstore
appstore_revenue_2024_06_3_month = (
    renewals_2024_06_3_month_payment_gateway['2024-09']['appstore'].amount.sum()
    + renewals_2024_06_3_month_payment_gateway['2024-12']['appstore'].amount.sum()
    + renewals_2024_06_3_month_payment_gateway['2025-03']['appstore'].amount.sum()
    + cohort_2024_06['three_month'][cohort_2024_06['three_month']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-06 3-month on Appstore: {appstore_revenue_2024_06_3_month:,} VND')

# Total revenue 2024-06 3-month on Playstore
playstore_revenue_2024_06_3_month = (
    renewals_2024_06_3_month_payment_gateway['2024-09']['playstore'].amount.sum()
    + renewals_2024_06_3_month_payment_gateway['2024-12']['playstore'].amount.sum()
    + renewals_2024_06_3_month_payment_gateway['2025-03']['playstore'].amount.sum()
    + cohort_2024_06['three_month'][cohort_2024_06['three_month']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-06 3-month on Playstore: {playstore_revenue_2024_06_3_month:,} VND')
```

```
Total revenue 2024-06 3-month on Appstore: 107,662,000 VND
Total revenue 2024-06 3-month on Playstore: 30,556,000 VND
```

4.2.3. One-year total revenue

```
# Total revenue 2024-01 1-year on Appstore
appstore_revenue_2024_01_1_year = (
    cohort_2024_01['one_year'][cohort_2024_01['one_year']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-01 one-year on Appstore: {appstore_revenue_2024_01_1_year:,} VND')

# Total revenue 2024-01 1-year on Playstore
playstore_revenue_2024_01_1_year = (
    cohort_2024_01['one_year'][cohort_2024_01['one_year']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-01 one-year on Playstore: {playstore_revenue_2024_01_1_year:,} VND')
```

```
Total revenue 2024-01 one-year on Appstore: 136,618,000 VND
Total revenue 2024-01 one-year on Playstore: 53,849,000 VND
```

```
# Total revenue 2024-02 1-year on Appstore
appstore_revenue_2024_02_1_year = (
     cohort_2024_02['one_year'][cohort_2024_02['one_year']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-02 one-year on Appstore: {appstore_revenue_2024_02_1_year:,} VND')


# Total revenue 2024-02 1-year on Playstore
playstore_revenue_2024_02_1_year = (
     cohort_2024_02['one_year'][cohort_2024_02['one_year']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-02 one-year on Playstore: {playstore_revenue_2024_02_1_year:,} VND')
```

```
Total revenue 2024-02 one-year on Appstore: 194,511,000 VND
Total revenue 2024-02 one-year on Playstore: 76,254,000 VND
```

```
# Total revenue 2024-03 1-year on Appstore
appstore_revenue_2024_03_1_year = (
     cohort_2024_03['one_year'][cohort_2024_03['one_year']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-03 one-year on Appstore: {appstore_revenue_2024_03_1_year:,} VND')


# Total revenue 2024-03 1-year on Playstore
playstore_revenue_2024_03_1_year = (
     cohort_2024_03['one_year'][cohort_2024_03['one_year']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-03 one-year on Playstore: {playstore_revenue_2024_03_1_year:,} VND')
```

```
Total revenue 2024-03 one-year on Appstore: 120,039,000 VND
Total revenue 2024-03 one-year on Playstore: 54,127,000 VND
```

```
# Total revenue 2024-04 1-year on Appstore
appstore_revenue_2024_04_1_year = (
    cohort_2024_04['one_year'][cohort_2024_04['one_year']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-04 one-year on Appstore: {appstore_revenue_2024_04_1_year:,} VND')


# Total revenue 2024-04 1-year on Playstore
playstore_revenue_2024_04_1_year = (
    cohort_2024_04['one_year'][cohort_2024_04['one_year']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-04 one-year on Playstore: {playstore_revenue_2024_04_1_year:,} VND')
```

```
Total revenue 2024-04 one-year on Appstore: 130,799,000 VND
Total revenue 2024-04 one-year on Playstore: 47,988,000 VND
```

```
# Total revenue 2024-05 1-year on Appstore
appstore_revenue_2024_05_1_year = (
    cohort_2024_05['one_year'][cohort_2024_05['one_year']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-05 one-year on Appstore: {appstore_revenue_2024_05_1_year:,} VND')


# Total revenue 2024-05 1-year on Playstore
playstore_revenue_2024_05_1_year = (
    cohort_2024_05['one_year'][cohort_2024_05['one_year']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-05 one-year on Playstore: {playstore_revenue_2024_05_1_year:,} VND')
```

```
Total revenue 2024-05 one-year on Appstore: 101,261,000 VND
Total revenue 2024-05 one-year on Playstore: 53,855,000 VND
```

```
# Total revenue 2024-06 1-year on Appstore
appstore_revenue_2024_06_1_year = (
    cohort_2024_06['one_year'][cohort_2024_06['one_year']['payment_gateway'] == 'appstore'].amount.sum()
)
print(f'Total revenue 2024-06 one-year on Appstore: {appstore_revenue_2024_06_1_year:,} VND')


# Total revenue 2024-06 1-year on Playstore
playstore_revenue_2024_06_1_year = (
    cohort_2024_06['one_year'][cohort_2024_06['one_year']['payment_gateway'] == 'playstore'].amount.sum()
)
print(f'Total revenue 2024-06 one-year on Playstore: {playstore_revenue_2024_06_1_year:,} VND')
```

```
Total revenue 2024-06 one-year on Appstore: 127,010,000 VND
Total revenue 2024-06 one-year on Playstore: 41,236,000 VND
```

```
# Appstore revenue 1-month from Cohort 1/24 to 6/24
appstore_total_revenue = pd.DataFrame({
    '2024-01': {
        'One month': appstore_revenue_2024_01_1_month,
        'Three month': appstore_revenue_2024_01_3_month,
        'One year': appstore_revenue_2024_01_1_year
    },
    '2024-02': {
        'One month': appstore_revenue_2024_02_1_month,
        'Three month': appstore_revenue_2024_02_3_month,
        'One year': appstore_revenue_2024_02_1_year
    },
    '2024-03': {
        'One month': appstore_revenue_2024_03_1_month,
        'Three month': appstore_revenue_2024_03_3_month,
        'One year': appstore_revenue_2024_03_1_year
    },
    '2024-04': {
        'One month': appstore_revenue_2024_04_1_month,
        'Three month': appstore_revenue_2024_04_3_month,
        'One year': appstore_revenue_2024_04_1_year
    },
    '2024-05': {
        'One month': appstore_revenue_2024_05_1_month,
        'Three month': appstore_revenue_2024_05_3_month,
        'One year': appstore_revenue_2024_05_1_year
    },
    '2024-06': {
        'One month': appstore_revenue_2024_06_1_month,
        'Three month': appstore_revenue_2024_06_3_month,
        'One year': appstore_revenue_2024_06_1_year
    }
})
```

▸ appstore_total_revenue: pandas.core.frame.DataFrame = [2024-01: int64, 2024-02: int64 ... 4 more fields]

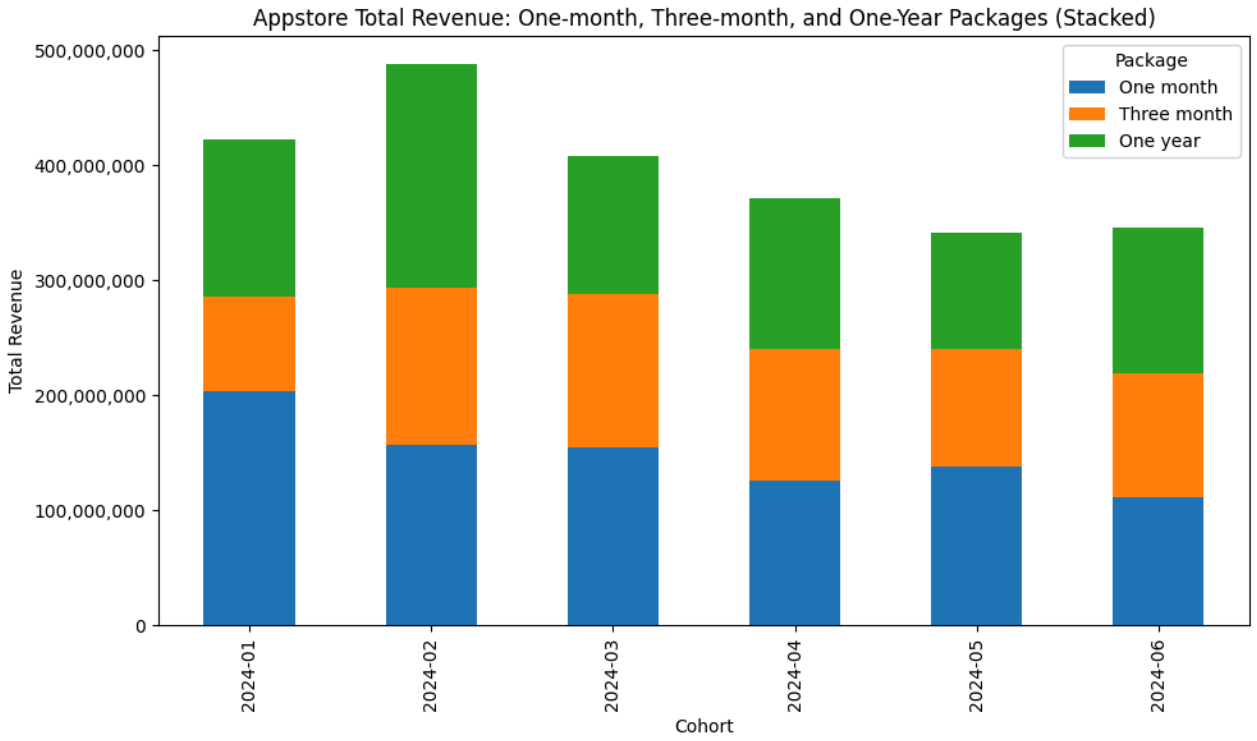
```
# Playstore revenue 1-month from Cohort 1/24 to 6/24
playstore_total_revenue = pd.DataFrame({
    '2024-01': {
        'One month': playstore_revenue_2024_01_1_month,
        'Three month': playstore_revenue_2024_01_3_month,
        'One year': playstore_revenue_2024_01_1_year
    },
    '2024-02': {
        'One month': playstore_revenue_2024_02_1_month,
        'Three month': playstore_revenue_2024_02_3_month,
        'One year': playstore_revenue_2024_02_1_year
    },
    '2024-03': {
        'One month': playstore_revenue_2024_03_1_month,
        'Three month': playstore_revenue_2024_03_3_month,
        'One year': playstore_revenue_2024_03_1_year
    },
    '2024-04': {
        'One month': playstore_revenue_2024_04_1_month,
        'Three month': playstore_revenue_2024_04_3_month,
        'One year': playstore_revenue_2024_04_1_year
    },
    '2024-05': {
        'One month': playstore_revenue_2024_05_1_month,
        'Three month': playstore_revenue_2024_05_3_month,
        'One year': playstore_revenue_2024_05_1_year
    },
    '2024-06': {
        'One month': playstore_revenue_2024_06_1_month,
        'Three month': playstore_revenue_2024_06_3_month,
        'One year': playstore_revenue_2024_06_1_year
    }
})
```

▸ playstore_total_revenue: pandas.core.frame.DataFrame = [2024-01: int64, 2024-02: int64 ... 4 more fields]

4.2.2. Appstore total revenue visualisation

```
# Reformat data for plotting (cohorts as x-axis)
df = appstore_total_revenue.T  # transpose so cohorts are rows

# Plot stacked bar chart
ax = df.plot(kind='bar', stacked=True, figsize=(10,6))

# Add labels and title
ax.set_xlabel('Cohort')
ax.set_ylabel('Total Revenue')
ax.set_title('Appstore Total Revenue: One-month, Three-month, and One-Year Packages (Stacked)')
ax.legend(title='Package')

# Format y-axis with commas
import matplotlib.ticker as mticker
ax.yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))

plt.tight_layout()
plt.show()
```

▸ df: pandas.core.frame.DataFrame = [One month: int64, Three month: int64 ... 1 more field]

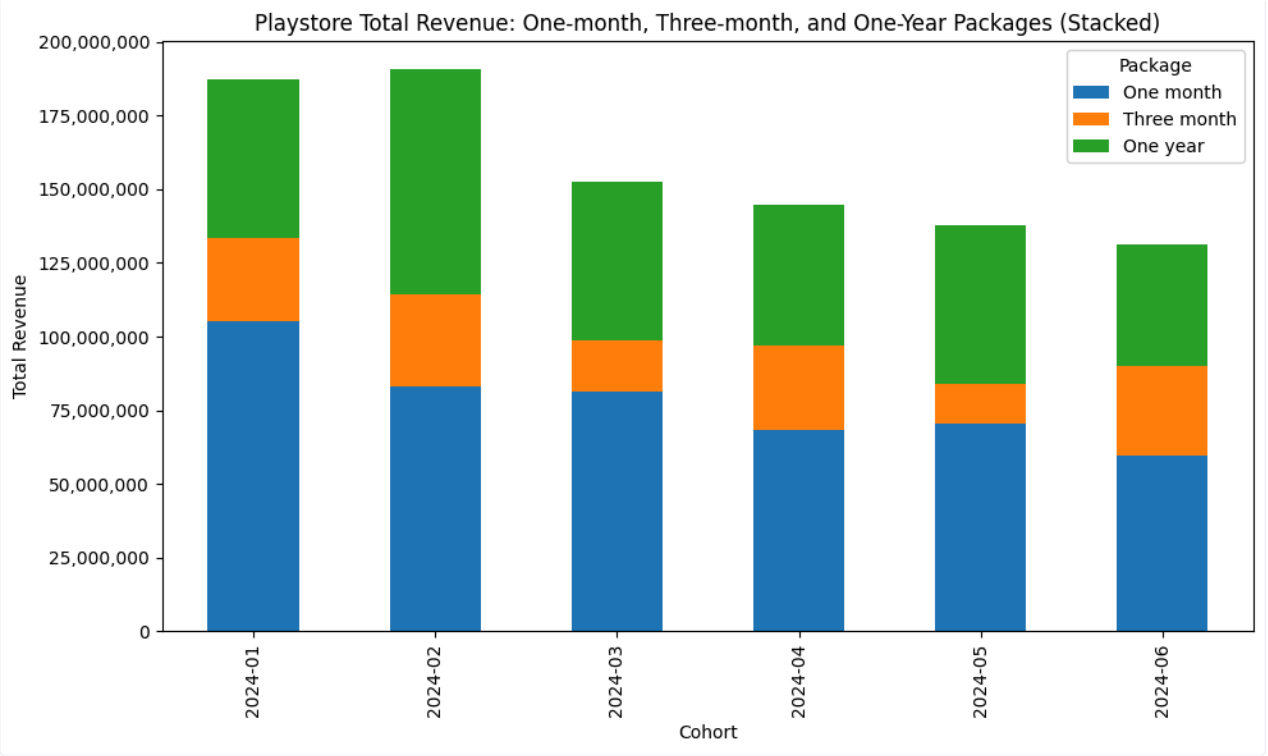


Overall, the three subscription user types contribute a relatively even amount of revenue, except for cohort 2024-01, where one-month users stand out, and cohort 2024-02, where one-year users generate an outstanding share.

4.2.3. Playstore total revenue visualisation

```
# Reformat data for plotting (cohorts as x-axis)
playstore_total_revenue_t = playstore_total_revenue.T

# Plot stacked bar chart
ax = playstore_total_revenue_t.plot(kind='bar', stacked=True, figsize=(10,6))

# Add labels and title
ax.set_xlabel('Cohort')
ax.set_ylabel('Total Revenue')
ax.set_title('Playstore Total Revenue: One-month, Three-month, and One-Year Packages (Stacked)')
ax.legend(title='Package')

# Format y-axis with commas
import matplotlib.ticker as mticker
ax.yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))

plt.tight_layout()
plt.show()
```

▸ playstore_total_revenue_t: pandas.core.frame.DataFrame = [One month: int64, Three month: int64 ... 1 more field]



On the Play Store, three-month subscription users consistently generate the lowest revenue share across all cohorts. Meanwhile, one-month users typically contribute the largest portion of revenue, with a notable increase in revenue from one-year users in cohort 2024-02

When comparing between platforms, Appstore consistently generates higher total revenue than Playstore for all six cohorts and across each subscription package type (one-month, three-month, and one-year). The revenue gap is especially pronounced for one-year and three-month subscriptions, suggesting that Appstore users prefer and purchase longer-term packages more frequently than Playstore users.

While both platforms experience a decline in total revenue over the cohorts, the rate of decline appears similar, and Appstore maintains a much higher revenue baseline throughout. This pattern suggests stronger long-term customer value and engagement for Appstore subscriptions compared to Playstore.

4.3. Average revenue per user

Average revenue per user is calculated by:

$$\text{ARPU} = \frac{\text{Total revenue from first purchases + Total revenue from renewals in each month}}{\text{Total number of first purchase users}}$$

4.3.1. One-month ARPU

```
# ARPU 2024-01 1-month on Appstore
appstore_arpu_2024_01_1_month = round( appstore_revenue_2024_01_1_month / len(cohort_2024_01['one_month']
[cohort_2024_01['one_month']['payment_gateway'] == 'appstore']), 2)

print('Average revenue per user of one-month subscription appstore users in Cohort 2024-01:')
print(f"{appstore_arpu_2024_01_1_month:,} VND")

# ARPU 2024-01 1-month on Playstore
playstore_arpu_2024_01_1_month = round(playstore_revenue_2024_01_1_month / len(cohort_2024_01['one_month']
[cohort_2024_01['one_month']['payment_gateway'] == 'playstore']), 2)

print('Average revenue per user of one-month subscription playstore users in Cohort 2024-01:')
print(f"{playstore_arpu_2024_01_1_month:,} VND" )
```

```
Average revenue per user of one-month subscription appstore users in Cohort 2024-01:
338,998.34 VND
Average revenue per user of one-month subscription playstore users in Cohort 2024-01:
320,384.15 VND
```

```
# ARPU 2024-02 1-month on Appstore
appstore_arpu_2024_02_1_month = round(
    appstore_revenue_2024_02_1_month / len(cohort_2024_02['one_month'][cohort_2024_02['one_month']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-month subscription appstore users in Cohort 2024-02:')
print(f"{appstore_arpu_2024_02_1_month:,} VND")

# ARPU 2024-02 1-month on Playstore
playstore_arpu_2024_02_1_month = round(
    playstore_revenue_2024_02_1_month / len(cohort_2024_02['one_month'][cohort_2024_02['one_month']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-month subscription playstore users in Cohort 2024-02:')
print(f"{playstore_arpu_2024_02_1_month:,} VND")
```

```
Average revenue per user of one-month subscription appstore users in Cohort 2024-02:
339,427.65 VND
Average revenue per user of one-month subscription playstore users in Cohort 2024-02:
305,922.51 VND
```

```
# ARPU 2024-03 1-month on Appstore
appstore_arpu_2024_03_1_month = round(
    appstore_revenue_2024_03_1_month / len(cohort_2024_03['one_month'][cohort_2024_03['one_month']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-month subscription appstore users in Cohort 2024-03:')
print(f"{appstore_arpu_2024_03_1_month:,} VND")

# ARPU 2024-03 1-month on Playstore
playstore_arpu_2024_03_1_month = round(
    playstore_revenue_2024_03_1_month / len(cohort_2024_03['one_month'][cohort_2024_03['one_month']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-month subscription playstore users in Cohort 2024-03:')
print(f"{playstore_arpu_2024_03_1_month:,} VND")
```

```
Average revenue per user of one-month subscription appstore users in Cohort 2024-03:
332,440.86 VND
Average revenue per user of one-month subscription playstore users in Cohort 2024-03:
329,655.87 VND
```

```
# ARPU 2024-04 1-month on Appstore
appstore_arpu_2024_04_1_month = round(
    appstore_revenue_2024_04_1_month / len(cohort_2024_04['one_month'][cohort_2024_04['one_month']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-month subscription appstore users in Cohort 2024-04:')
print(f"{appstore_arpu_2024_04_1_month:,} VND")

# ARPU 2024-04 1-month on Playstore
playstore_arpu_2024_04_1_month = round(
    playstore_revenue_2024_04_1_month / len(cohort_2024_04['one_month'][cohort_2024_04['one_month']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-month subscription playstore users in Cohort 2024-04:')
print(f"{playstore_arpu_2024_04_1_month:,} VND")
```

```
Average revenue per user of one-month subscription appstore users in Cohort 2024-04:
328,309.9 VND
Average revenue per user of one-month subscription playstore users in Cohort 2024-04:
335,078.43 VND
```

```
# ARPU 2024-05 1-month on Appstore
appstore_arpu_2024_05_1_month = round(
    appstore_revenue_2024_05_1_month / len(cohort_2024_05['one_month'][cohort_2024_05['one_month']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-month subscription appstore users in Cohort 2024-05:')
print(f"{appstore_arpu_2024_05_1_month:,} VND")

# ARPU 2024-05 1-month on Playstore
playstore_arpu_2024_05_1_month = round(
    playstore_revenue_2024_05_1_month / len(cohort_2024_05['one_month'][cohort_2024_05['one_month']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-month subscription playstore users in Cohort 2024-05:')
print(f"{playstore_arpu_2024_05_1_month:,} VND")
```

```
Average revenue per user of one-month subscription appstore users in Cohort 2024-05:
363,857.52 VND
Average revenue per user of one-month subscription playstore users in Cohort 2024-05:
296,773.11 VND
```

```
# ARPU 2024-06 1-month on Appstore
appstore_arpu_2024_06_1_month = round(
    appstore_revenue_2024_06_1_month / len(cohort_2024_06['one_month'][cohort_2024_06['one_month']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-month subscription appstore users in Cohort 2024-06:')
print(f"{appstore_arpu_2024_06_1_month:,} VND")

# ARPU 2024-06 1-month on Playstore
playstore_arpu_2024_06_1_month = round(
    playstore_revenue_2024_06_1_month / len(cohort_2024_06['one_month'][cohort_2024_06['one_month']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-month subscription playstore users in Cohort 2024-06:')
print(f"{playstore_arpu_2024_06_1_month:,} VND")
```

```
Average revenue per user of one-month subscription appstore users in Cohort 2024-06:
347,881.25 VND
Average revenue per user of one-month subscription playstore users in Cohort 2024-06:
330,272.22 VND
```

4.3.2. Three-month ARPU

```
# ARPU 2024-01 3-month on Appstore
appstore_arpu_2024_01_3_month = round(
    appstore_revenue_2024_01_3_month / len(cohort_2024_01['three_month'][cohort_2024_01['three_month']['payment_gateway']
== 'appstore']),
    2
)
print('Average revenue per user of three-month subscription appstore users in Cohort 2024-01:')
print(f"{appstore_arpu_2024_01_3_month:,} VND")

# ARPU 2024-01 3-month on Playstore
playstore_arpu_2024_01_3_month = round(
    playstore_revenue_2024_01_3_month / len(cohort_2024_01['three_month'][cohort_2024_01['three_month']['payment_gateway']
== 'playstore']),
    2
)
print('Average revenue per user of three-month subscription playstore users in Cohort 2024-01:')
print(f"{playstore_arpu_2024_01_3_month:,} VND")
```

```
Average revenue per user of three-month subscription appstore users in Cohort 2024-01:
461,117.98 VND
Average revenue per user of three-month subscription playstore users in Cohort 2024-01:
472,983.33 VND
```

```
    # ARPU 2024-02 3-month on Appstore
    appstore_arpu_2024_02_3_month = round(
        appstore_revenue_2024_02_3_month / len(cohort_2024_02['three_month'][cohort_2024_02['three_month']['payment_gateway']
    == 'appstore']),
        2
    )
    print('Average revenue per user of three-month subscription appstore users in Cohort 2024-02:')
    print(f"{appstore_arpu_2024_02_3_month:,} VND")

    # ARPU 2024-02 3-month on Playstore
    playstore_arpu_2024_02_3_month = round(
        playstore_revenue_2024_02_3_month / len(cohort_2024_02['three_month'][cohort_2024_02['three_month']['payment_gateway']
    == 'playstore']),
        2
    )
    print('Average revenue per user of three-month subscription playstore users in Cohort 2024-02:')
    print(f"{playstore_arpu_2024_02_3_month:,} VND")
```

```
Average revenue per user of three-month subscription appstore users in Cohort 2024-02:
406,089.55 VND
Average revenue per user of three-month subscription playstore users in Cohort 2024-02:
404,025.64 VND
```

```
    # ARPU 2024-03 3-month on Appstore
    appstore_arpu_2024_03_3_month = round(
        appstore_revenue_2024_03_3_month / len(cohort_2024_03['three_month'][cohort_2024_03['three_month']['payment_gateway']
    == 'appstore']),
        2
    )
    print('Average revenue per user of three-month subscription appstore users in Cohort 2024-03:')
    print(f"{appstore_arpu_2024_03_3_month:,} VND")

    # ARPU 2024-03 3-month on Playstore
    playstore_arpu_2024_03_3_month = round(
        playstore_revenue_2024_03_3_month / len(cohort_2024_03['three_month'][cohort_2024_03['three_month']['payment_gateway']
    == 'playstore']),
        2
    )
    print('Average revenue per user of three-month subscription playstore users in Cohort 2024-03:')
    print(f"{playstore_arpu_2024_03_3_month:,} VND")
```

```
Average revenue per user of three-month subscription appstore users in Cohort 2024-03:
432,590.91 VND
Average revenue per user of three-month subscription playstore users in Cohort 2024-03:
452,026.32 VND
```

```
    # ARPU 2024-04 3-month on Appstore
    appstore_arpu_2024_04_3_month = round(
        appstore_revenue_2024_04_3_month / len(cohort_2024_04['three_month'][cohort_2024_04['three_month']['payment_gateway']
    == 'appstore']),
        2
    )
    print('Average revenue per user of three-month subscription appstore users in Cohort 2024-04:')
    print(f"{appstore_arpu_2024_04_3_month:,} VND")

    # ARPU 2024-04 3-month on Playstore
    playstore_arpu_2024_04_3_month = round(
        playstore_revenue_2024_04_3_month / len(cohort_2024_04['three_month'][cohort_2024_04['three_month']['payment_gateway']
    == 'playstore']),
        2
    )
    print('Average revenue per user of three-month subscription playstore users in Cohort 2024-04:')
    print(f"{playstore_arpu_2024_04_3_month:,} VND")
```

```
Average revenue per user of three-month subscription appstore users in Cohort 2024-04:
415,930.91 VND
Average revenue per user of three-month subscription playstore users in Cohort 2024-04:
402,845.07 VND
```

```
# ARPU 2024-05 3-month on Appstore
appstore_arpu_2024_05_3_month = round(
    appstore_revenue_2024_05_3_month / len(cohort_2024_05['three_month'][cohort_2024_05['three_month']['payment_gateway']
== 'appstore']),
    2
)
print('Average revenue per user of three-month subscription appstore users in Cohort 2024-05:')
print(f"{appstore_arpu_2024_05_3_month:,} VND")

# ARPU 2024-05 3-month on Playstore
playstore_arpu_2024_05_3_month = round(
    playstore_revenue_2024_05_3_month / len(cohort_2024_05['three_month'][cohort_2024_05['three_month']['payment_gateway']
== 'playstore']),
    2
)
print('Average revenue per user of three-month subscription playstore users in Cohort 2024-05:')
print(f"{playstore_arpu_2024_05_3_month:,} VND")
```

```
Average revenue per user of three-month subscription appstore users in Cohort 2024-05:
429,852.94 VND
Average revenue per user of three-month subscription playstore users in Cohort 2024-05:
396,823.53 VND
```

```
# ARPU 2024-06 3-month on Appstore
appstore_arpu_2024_06_3_month = round(
    appstore_revenue_2024_06_3_month / len(cohort_2024_06['three_month'][cohort_2024_06['three_month']['payment_gateway']
== 'appstore']),
    2
)
print('Average revenue per user of three-month subscription appstore users in Cohort 2024-06:')
print(f"{appstore_arpu_2024_06_3_month:,} VND")

# ARPU 2024-06 3-month on Playstore
playstore_arpu_2024_06_3_month = round(
    playstore_revenue_2024_06_3_month / len(cohort_2024_06['three_month'][cohort_2024_06['three_month']['payment_gateway']
== 'playstore']),
    2
)
print('Average revenue per user of three-month subscription playstore users in Cohort 2024-06:')
print(f"{playstore_arpu_2024_06_3_month:,} VND")
```

```
Average revenue per user of three-month subscription appstore users in Cohort 2024-06:
420,554.69 VND
```

```
Average revenue per user of three-month subscription playstore users in Cohort 2024-06:
462,969.7 VND
```

4.3.3. One-year ARPU

```
# ARPU 2024-01 1-year on Appstore
appstore_arpu_2024_01_1_year = round(
    appstore_revenue_2024_01_1_year / len(cohort_2024_01['one_year'][cohort_2024_01['one_year']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-year subscription appstore users in Cohort 2024-06:')
print(f"{appstore_arpu_2024_01_1_year:,} VND")

# ARPU 2024-01 1-year on Playstore
playstore_arpu_2024_01_1_year = round(
    playstore_revenue_2024_01_1_year / len(cohort_2024_01['one_year'][cohort_2024_01['one_year']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-year subscription playstore users in Cohort 2024-06:')
print(f"{playstore_arpu_2024_01_1_year:,} VND")
```

```
Average revenue per user of one-year subscription appstore users in Cohort 2024-06:
750,648.35 VND
Average revenue per user of one-year subscription playstore users in Cohort 2024-06:
699,337.66 VND
```

```
# ARPU 2024-02 1-year on Appstore
appstore_arpu_2024_02_1_year = round(
    appstore_revenue_2024_02_1_year / len(cohort_2024_02['one_year'][cohort_2024_02['one_year']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-year subscription appstore users in Cohort 2024-02:')
print(f"{appstore_arpu_2024_02_1_year:,} VND")

# ARPU 2024-02 1-year on Playstore
playstore_arpu_2024_02_1_year = round(
    playstore_revenue_2024_02_1_year / len(cohort_2024_02['one_year'][cohort_2024_02['one_year']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-year subscription playstore users in Cohort 2024-02:')
print(f"{playstore_arpu_2024_02_1_year:,} VND")
```

```
Average revenue per user of one-year subscription appstore users in Cohort 2024-02:
673,048.44 VND
Average revenue per user of one-year subscription playstore users in Cohort 2024-02:
668,894.74 VND
```

```
# ARPU 2024-03 1-year on Appstore
appstore_arpu_2024_03_1_year = round(
    appstore_revenue_2024_03_1_year / len(cohort_2024_03['one_year'][cohort_2024_03['one_year']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-year subscription appstore users in Cohort 2024-03:')
print(f"{appstore_arpu_2024_03_1_year:,} VND")

# ARPU 2024-03 1-year on Playstore
playstore_arpu_2024_03_1_year = round(
    playstore_revenue_2024_03_1_year / len(cohort_2024_03['one_year'][cohort_2024_03['one_year']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-year subscription playstore users in Cohort 2024-03:')
print(f"{playstore_arpu_2024_03_1_year:,} VND")
```

```
Average revenue per user of one-year subscription appstore users in Cohort 2024-03:
745,583.85 VND
Average revenue per user of one-year subscription playstore users in Cohort 2024-03:
693,935.9 VND
```

```
# ARPU 2024-04 1-year on Appstore
appstore_arpu_2024_04_1_year = round(
    appstore_revenue_2024_04_1_year / len(cohort_2024_04['one_year'][cohort_2024_04['one_year']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-year subscription appstore users in Cohort 2024-04:')
print(f"{appstore_arpu_2024_04_1_year:,} VND")

# ARPU 2024-04 1-year on Playstore
playstore_arpu_2024_04_1_year = round(
    playstore_revenue_2024_04_1_year / len(cohort_2024_04['one_year'][cohort_2024_04['one_year']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-year subscription playstore users in Cohort 2024-04:')
print(f"{playstore_arpu_2024_04_1_year:,} VND")
```

```
Average revenue per user of one-year subscription appstore users in Cohort 2024-04:
650,741.29 VND
Average revenue per user of one-year subscription playstore users in Cohort 2024-04:
639,840.0 VND
```

```
# ARPU 2024-05 1-year on Appstore
appstore_arpu_2024_05_1_year = round(
    appstore_revenue_2024_05_1_year / len(cohort_2024_05['one_year'][cohort_2024_05['one_year']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-year subscription appstore users in Cohort 2024-05:')
print(f"{appstore_arpu_2024_05_1_year:,} VND")

# ARPU 2024-05 1-year on Playstore
playstore_arpu_2024_05_1_year = round(
    playstore_revenue_2024_05_1_year / len(cohort_2024_05['one_year'][cohort_2024_05['one_year']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-year subscription playstore users in Cohort 2024-05:')
print(f"{playstore_arpu_2024_05_1_year:,} VND")
```

```
Average revenue per user of one-year subscription appstore users in Cohort 2024-05:
728,496.4 VND
```

```
Average revenue per user of one-year subscription playstore users in Cohort 2024-05:
708,618.42 VND
```

```
# ARPU 2024-06 1-year on Appstore
appstore_arpu_2024_06_1_year = round(
    appstore_revenue_2024_06_1_year / len(cohort_2024_06['one_year'][cohort_2024_06['one_year']['payment_gateway'] ==
'appstore']),
    2
)
print('Average revenue per user of one-year subscription appstore users in Cohort 2024-06:')
print(f"{appstore_arpu_2024_06_1_year:,} VND")

# ARPU 2024-06 1-year on Playstore
playstore_arpu_2024_06_1_year = round(
    playstore_revenue_2024_06_1_year / len(cohort_2024_06['one_year'][cohort_2024_06['one_year']['payment_gateway'] ==
'playstore']),
    2
)
print('Average revenue per user of one-year subscription playstore users in Cohort 2024-06:')
print(f"{playstore_arpu_2024_06_1_year:,} VND")
```

```
Average revenue per user of one-year subscription appstore users in Cohort 2024-06:
668,473.68 VND
Average revenue per user of one-year subscription playstore users in Cohort 2024-06:
644,312.5 VND
```

```
# Appstore ARPU
appstore_arpu = pd.DataFrame({
    '2024-01': {
        'One month': appstore_arpu_2024_01_1_month,
        'Three month': appstore_arpu_2024_01_3_month,
        'One year': appstore_arpu_2024_01_1_year
    },
    '2024-02': {
        'One month': appstore_arpu_2024_02_1_month,
        'Three month': appstore_arpu_2024_02_3_month,
        'One year': appstore_arpu_2024_02_1_year
    },
    '2024-03': {
        'One month': appstore_arpu_2024_03_1_month,
        'Three month': appstore_arpu_2024_03_3_month,
        'One year': appstore_arpu_2024_03_1_year
    },
    '2024-04': {
        'One month': appstore_arpu_2024_04_1_month,
        'Three month': appstore_arpu_2024_04_3_month,
        'One year': appstore_arpu_2024_04_1_year
    },
    '2024-05': {
        'One month': appstore_arpu_2024_05_1_month,
        'Three month': appstore_arpu_2024_05_3_month,
        'One year': appstore_arpu_2024_05_1_year
    },
    '2024-06': {
        'One month': appstore_arpu_2024_06_1_month,
        'Three month': appstore_arpu_2024_06_3_month,
        'One year': appstore_arpu_2024_06_1_year
    }
})
```

▸ appstore_arpu: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

```
# Playstore ARPU
playstore_arpu = pd.DataFrame({
    '2024-01': {
        'One month': playstore_arpu_2024_01_1_month,
        'Three month': playstore_arpu_2024_01_3_month,
        'One year': playstore_arpu_2024_01_1_year
    },
    '2024-02': {
        'One month': playstore_arpu_2024_02_1_month,
        'Three month': playstore_arpu_2024_02_3_month,
        'One year': playstore_arpu_2024_02_1_year
    },
    '2024-03': {
        'One month': playstore_arpu_2024_03_1_month,
        'Three month': playstore_arpu_2024_03_3_month,
        'One year': playstore_arpu_2024_03_1_year
    },
    '2024-04': {
        'One month': playstore_arpu_2024_04_1_month,
        'Three month': playstore_arpu_2024_04_3_month,
        'One year': playstore_arpu_2024_04_1_year
    },
    '2024-05': {
        'One month': playstore_arpu_2024_05_1_month,
        'Three month': playstore_arpu_2024_05_3_month,
        'One year': playstore_arpu_2024_05_1_year
    },
    '2024-06': {
        'One month': playstore_arpu_2024_06_1_month,
        'Three month': playstore_arpu_2024_06_3_month,
        'One year': playstore_arpu_2024_06_1_year
    }
})
```

▸ playstore_arpu: pandas.core.frame.DataFrame = [2024-01: float64, 2024-02: float64 ... 4 more fields]

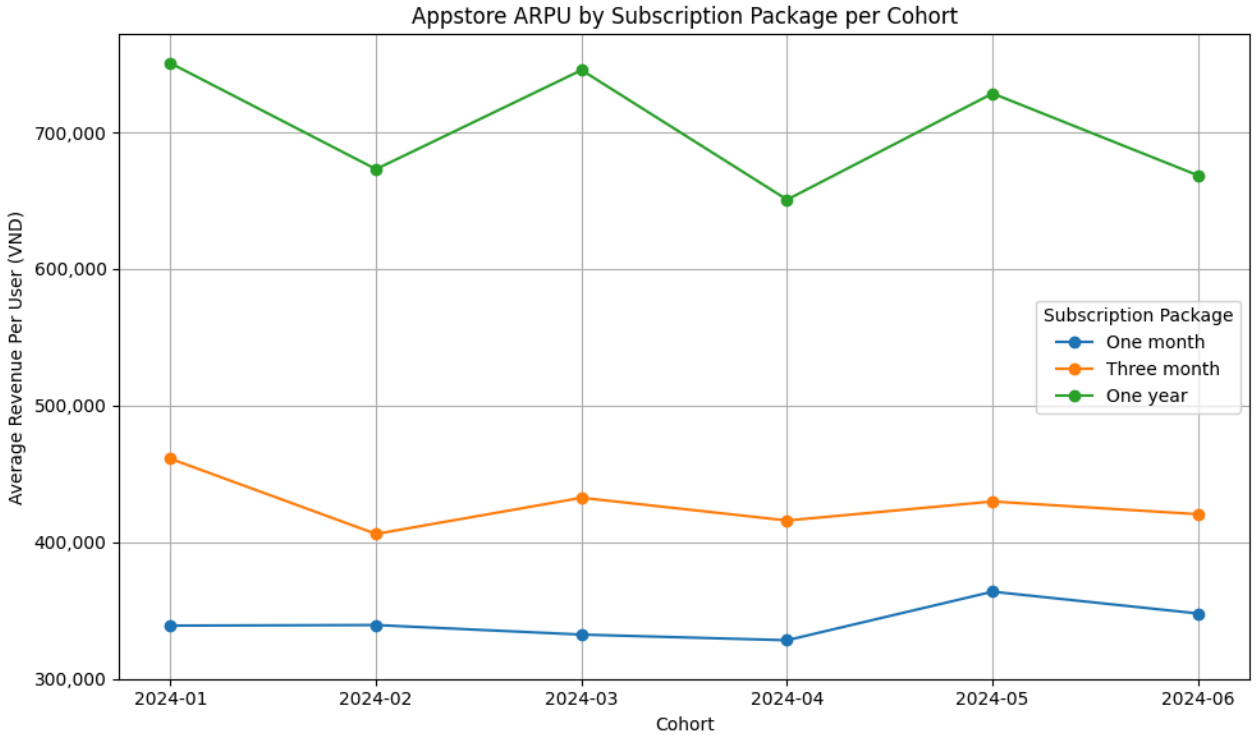
4.3.4. ARPU visualisation

Appstore ARPU by packages

```
appstore_arpu_t = appstore_arpu.T

# Plot line chart
ax = appstore_arpu_t.plot(kind='line', marker='o', figsize=(10,6))

# Set labels and title
ax.set_xlabel('Cohort')
ax.set_ylabel('Average Revenue Per User (VND)')
ax.set_title('Appstore ARPU by Subscription Package per Cohort')
ax.grid(True)
ax.set_ylim(bottom=300000)
ax.yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))
plt.tight_layout()
ax.legend(title='Subscription Package')
```

▸ appstore_arpu_t: pandas.core.frame.DataFrame = [One month: float64, Three month: float64 ... 1 more field]

```
<matplotlib.legend.Legend at 0xfff3c42b4650>
```

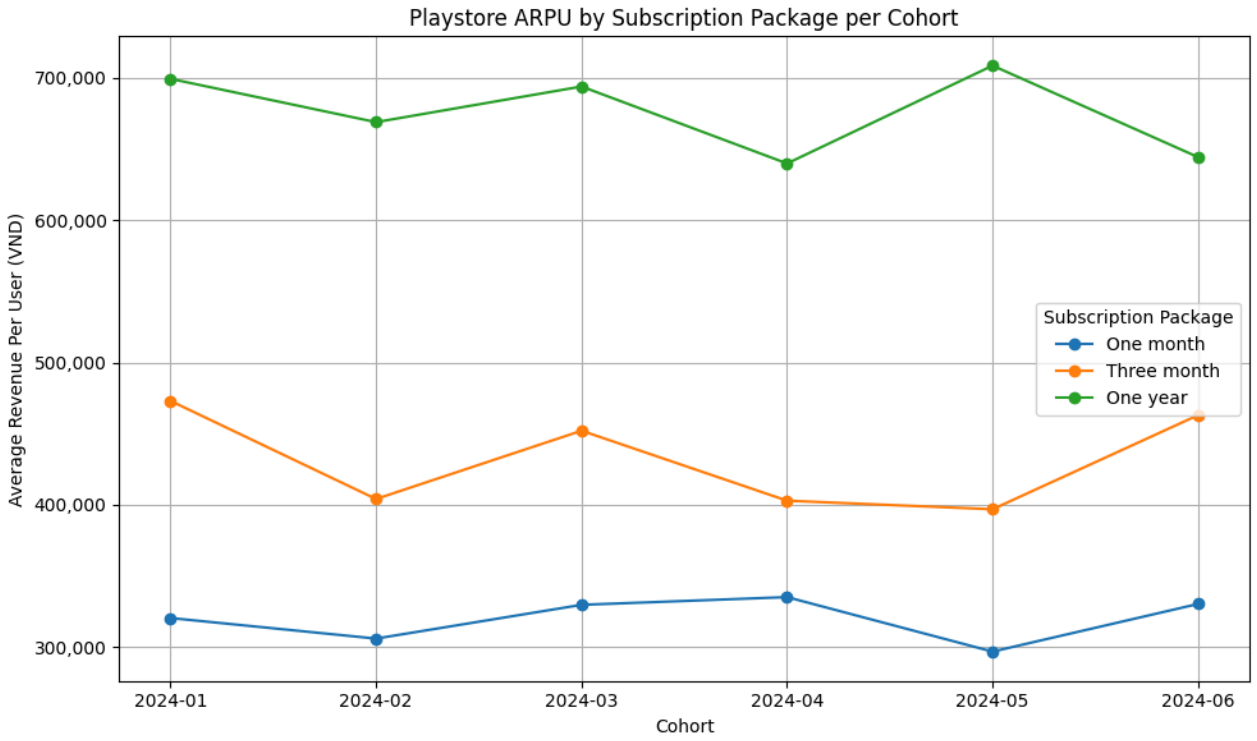


Playstore ARPU by packages

```
playstore_arpu_t = playstore_arpu.T

# Plot line chart
ax = playstore_arpu_t.plot(kind='line', marker='o', figsize=(10,6))

# Set labels and title
ax.set_xlabel('Cohort')
ax.set_ylabel('Average Revenue Per User (VND)')
ax.set_title('Playstore ARPU by Subscription Package per Cohort')
ax.grid(True)
ax.yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))
plt.tight_layout()
ax.legend(title='Subscription Package')
```

▸ playstore_arpu_t: pandas.core.frame.DataFrame = [One month: float64, Three month: float64 ... 1 more field]

```
<matplotlib.legend.Legend at 0xfff3bf1af500>
```



Appstore users generate higher Average Revenue Per User (ARPU) compared to Playstore users across all subscription packages and cohorts. The one-year subscription package shows the highest ARPU on both platforms due to the higher fees and one renewal period only, but this gap is more pronounced on the Appstore, indicating stronger long-term user spending there.

The three-month and one-month packages show variability in ARPU across cohorts but maintain the trend of higher ARPU for Appstore users. This suggests that while shorter-term packages fluctuate, Appstore users maintain higher per-user revenue overall.

5. CONCLUSION

- Retention Rate

Retention rates by gender and platform reveal considerable gaps and inconsistent trends across cohorts and user types, but platform-based analysis is preferred due to more stable trends and smaller internal fluctuations. App Store users consistently show higher retention rates, particularly in long-term subscriptions, indicating stronger user loyalty and lifetime value.

- Average Revenue Per User (ARPU) and Total Revenue

The Appstore delivers superior revenue performance both per user and in total subscription revenue across all package types compared to Playstore. Based on business manager insights, the business becomes profitable when user ARPU exceeds 400,000 VND. Both the one-year and three-month subscription users consistently exceed this profitability threshold, demonstrating their critical contribution to business revenue.

In contrast, the one-month subscription users, while valuable, fall below this profit level on average, highlighting a need to improve their lifetime value to enhance overall profitability. Targeted promotions or incentives for one-month subscribers on both platforms could extend their subscription duration and increase their ARPU.