**Описание исследуемого датасета**

В документе предоставлены данные по мобильному приложению для пользователей, установивших игру с 21.09.23 по 28.09.23.

Пояснение к таблицe:

1. Лист *ad revenue* (данные по рекламному заработку):

- *Install\_date* — дата установки игры пользователем

- *Event\_date* — дата совершения события

- Event\_revenue — рекламный заработок

- *Platform* — платформа

- *User\_id* — уникальный идентификатор пользователя

2. Лист *installs data* (данные по количеству инсталлов):

- *Install\_date* — дата

- *installs* — количество инсталлов

- *Platform* — платформа

**Цели**

1. Провести исследовательский анализ данных (EDA) для выявления общих закономерностей и распределений.
2. Рассчитать ключевые метрики для когорт:

* *DAU (Daily Active Users)* — ежедневно активные пользователи.
* *Retention (Коэффициент удержания)* — процент пользователей, продолжающих использовать приложение.
* *ARPU (Average Revenue Per User)* — средняя выручка на одного пользователя.
* *Revenue (выручка)* и *Cumulative Revenue* (накопительная выручка).
* *Revenue Growth Rate* (Коэффициент роста выручки).

1. Проверить статистические гипотезы о влиянии размеров когорт и активности пользователей на выручку.
2. Построить прогноз удержания пользователей (*Retention*) и роста выручки (*Revenue Growth Rate*) за первый месяц.

### ****План выполнения задач****

**1. Исследовательский анализ данных (EDA)**

* Импорт данных с листов *ad revenue* и *installs data*.
* Проверка и обработка пропусков, анализ типов данных.
* Исследование распределений ключевых переменных (выручка, активные пользователи, размеры когорт).
* Выявление возможных аномалий и выбросов.

**2. Расчёт, анализ и визуализация ключевых метрик по когортам**

* *DAU (Daily Active Users):*
  + Подсчитайте количество уникальных пользователей (*User\_id*) активных каждый день
  + Визуализируйте динамику DAU по когортам и платформам
* *Retention Rate (Коэффициент удержания):*
  + Рассчитайте процент пользователей, активных в N-й день после установки
  + Постройте тепловую таблицу удержания (cohort retention heatmap) и линейный график по когортам.
* *Revenue (Выручка):*
  + Рассчитать суммарную выручку по дням для каждой когорты
  + Построить тепловую таблицу и график распределения выручки по когортам
* *Cumulative Revenue (Накопительная выручка):*
  + Рассчитать кумулятивную выручку
  + Построить график роста накопительной выручки для когорт
* *Revenue Growth Rate (Коэффициент роста выручки):*
  + Рассчитать прирост выручки по дням для каждой когорты
  + Построить график темпов роста по когортам
* *ARPU (Average Revenue Per User):*
  + Рассчитайте среднюю выручку на пользователя
  + Постройте графики ARPU по когортам и платформам

**3. Проверка гипотез**

* *Гипотеза 1:* Выручка больше зависит от размера когорт, чем от активности единичных пользователей.
  + Корреляционный анализ между размером когорт, Retention Rate и выручкой.
  + Регрессионный анализ для оценки вклада размеров когорт и активности.
* *Гипотеза 2:* Принцип Парето (80/20) соблюдается: 20% пользователей генерируют 80% выручки.
  + Проверьте долю выручки, генерируемую топ-20% пользователей.
* *Гипотеза 3:* Retention Rate различается между Android и iOS.
  + Используйте тест Манна-Уитни или t-тест для сравнения.

**4. Прогнозирование удержания и роста выручки**

* *Прогноз удержания пользователей (Retention):*
  + - Постройте модель удержания для прогнозирования Retention Rate до 30-го дня.
    - Постройте график удержания для когорт за первый месяц.
* *Прогноз роста выручки (Revenue Growth Rate):*
  + - Используйте данные о выручке за первые несколько дней для построения прогнозной модели.
    - Рассчитайте, во сколько раз увеличится доход 1-го и 3-го дня по сравнению с 30-м днём.
    - Постройте график роста выручки за первый месяц.

**1. Исследовательский анализ данных (Exploratory Data Analysis, EDA)**

В рамках исследовательского анализа были выполнены следующие шаги:

* 1. **Подключение библиотек**

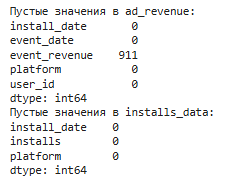
Были подключены следующие библиотеки:

* + Для анализа данных: pandas, numpy.
  + Для визуализации: matplotlib, seaborn.
  + Для прогнозирования: scipy.optimize (метод curve\_fit) и sklearn.metrics (оценка качества модели с использованием mean\_squared\_error и r2\_score).
  1. **Выгрузка данных**

Данные были импортированы с листов *ad revenue* и *installs data*. Таблицы содержат информацию о рекламной выручке (*event revenue*), установках, активных пользователях и других ключевых показателях.

* 1. **Проверка на пропуски**

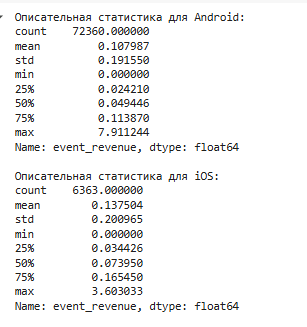
Подсчитано количество нулевых значений в каждом столбце.



Вычислен процент пропущенных данных относительно общего количества строк. Процент пропущенных данных (1.16%) относительно мал, и их заполнение нулями не должно существенно повлиять на общую статистику. Пропущенные значения могут означать, что в эти моменты не было активности, связанной с доходом от рекламы, и поэтому их логично интерпретировать как нулевой доход. Заполнение пропусков нулями позволит избежать проблем при вычислениях метрик (ARPU, DAU и т.д.), так как модели и методы анализа не смогут работать с пропущенными значениями.

* 1. **Анализ дохода за рекламу:**

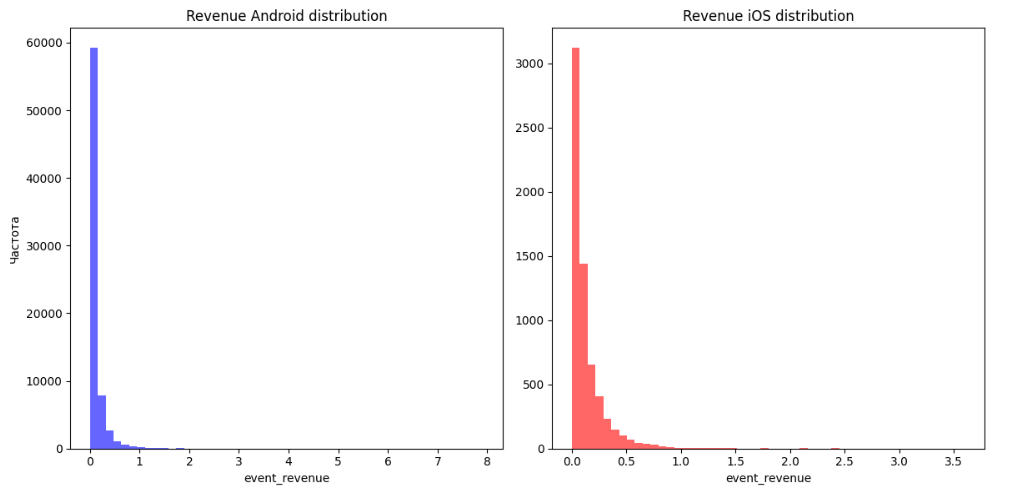
Для столбца *event revenue* были рассчитаны основные статистические показатели по платформам:

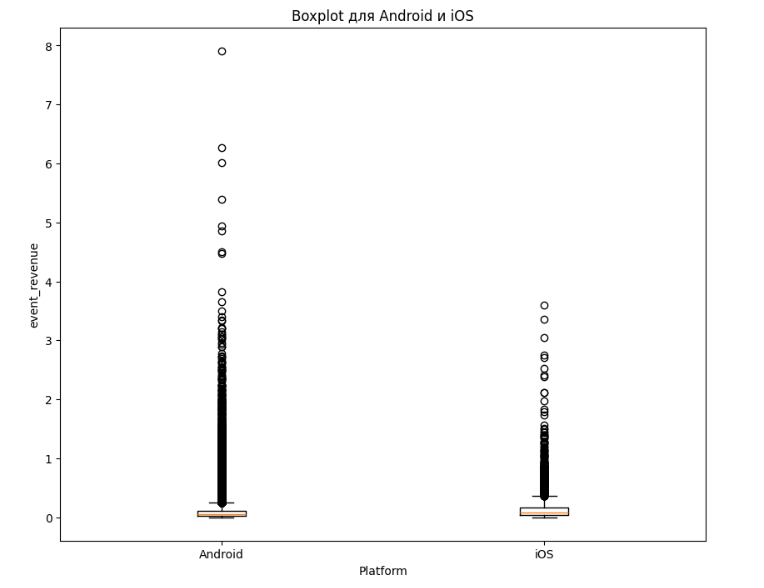


Как показывает описательная статистика, максимальное значение event\_revenue значительно выше 75-го перцентиля (максимум ≈ 7.91, а 75% значений ниже 0.12). Это может говорить о наличии выбросов или значительной асимметрии в данных.

Визуализация данных поможет лучше понять, насколько серьёзны выбросы и насколько они влияют на расчёт метрик.

Если данные сильно асимметричны или содержат выбросы, это может существенно повлиять на метрики, такие как ARPU (Average Revenue Per User). Аномально высокие значения могут исказить средние показатели.





**Выводы о доходе за просмотр рекламы**

Доход за один просмотр рекламы сильно скошен: большинство просмотров приносят низкий доход, но есть редкие события с высокой выручкой.

Средний доход выше на iOS (0.1375) по сравнению с Android (0.1079).

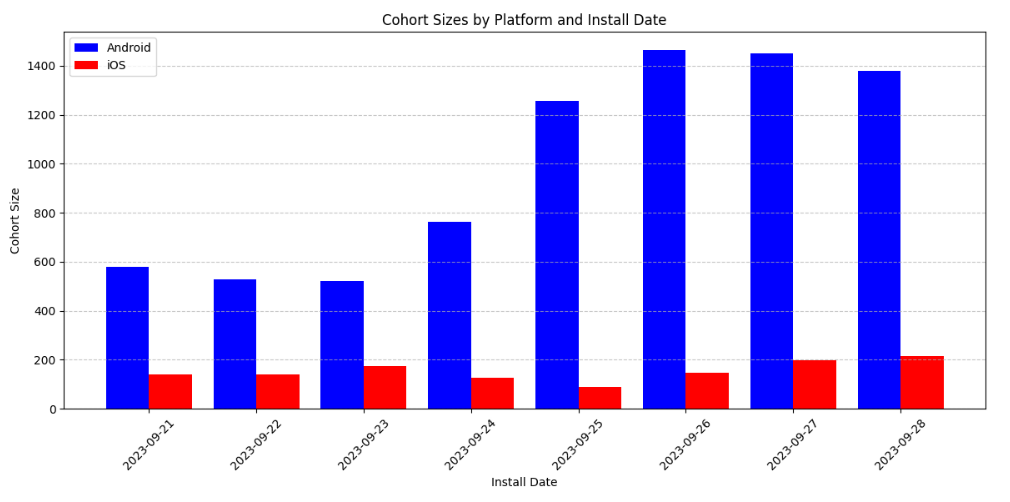
Выбросы:

На обеих платформах есть значительное количество событий с высокой выручкой (выбросы), что увеличивает средние значения.

Влияние на метрики:

* ARPU: Высокие значения дохода за отдельные просмотры могут завышать средний доход на пользователя.
* Revenue Growth Rate: более высокая выручка на iOS может создавать иллюзию быстрого роста доходов.
  1. **Aнализ когорт по платформам**

Данные были сгруппированы по дате установки (install\_date) и платформе (platform). Размер каждой когорты вычислен как сумма установок за конкретную дату и платформу. Использованы агрегированные данные с колонками: install\_date, platform, cohort\_size.



На графике показан размер когорт (количество установок) по датам установки для платформ Android (синие столбики) и iOS (красные столбики). Каждая дата установки (кохорта) представлена двумя столбиками: один для Android, другой для iOS.

**Выводы по анализу когорт:**

* Размеры когорт Android значительно превышают размеры когорт iOS на всех датах установки.
* Начиная с 24 сентября, наблюдается рост размеров когорт Android, достигающий пика 26-27 сентября.
* Для iOS размер когорт стабилен и сравнительно мал (около 200 пользователей), что отражает дисбаланс в выборке.
* Эти наблюдения подтверждают необходимость раздельного анализа платформ из-за значительных различий в количестве пользователей.

1. **Расчёт, анализ и визуализация ключевых метрик для когорт**
   1. **Расчёт метрик**

Для расчёта ключевых метрик была создана таблица daily\_data. Она агрегирует данные на уровне когорт (дата установки и платформа) и ежедневной активности пользователей. Эта таблица позволяет анализировать поведение пользователей, вычислять метрики удержания и дохода, а также строить прогнозы роста выручки и активности.



### ****Описание таблицы**** *daily\_data*

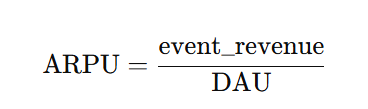
Таблица содержит следующие столбцы:

* *install\_date****:*** Дата установки приложения, определяющая когорту.
* *event\_date****:*** Дата активности пользователей, фиксирующая день взаимодействия с приложением.
* *platform****:*** Платформа (Android или iOS), на которой активны пользователи.
* *DAU****:*** Количество уникальных пользователей, активных в конкретный день.
* *event\_revenue****:*** Суммарная выручка за день активности.
* *days\_since\_install****:*** Количество дней с момента установки:

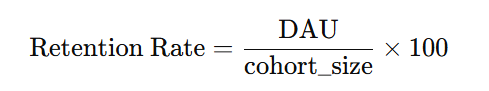


Этот столбец нормализует данные, позволяя анализировать пользователей относительно их дней в приложении.

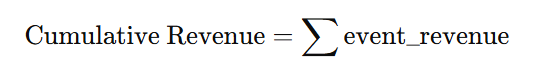
* *cohort\_size****:*** Размер когорты — количество пользователей, установивших приложение в конкретный день.
* *ARPU***:** Средняя выручка на одного активного пользователя за день:



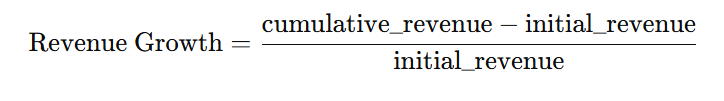
* *retention\_rate***:** Процент активных пользователей от общего размера когорты, отражающий удержание:



* *cumulative\_revenue****:*** Накопительная выручка когорты, рассчитанная как сумма выручки за все предыдущие дни:



* *revenue\_growth****:*** Прирост накопительной выручки относительно первого дня:



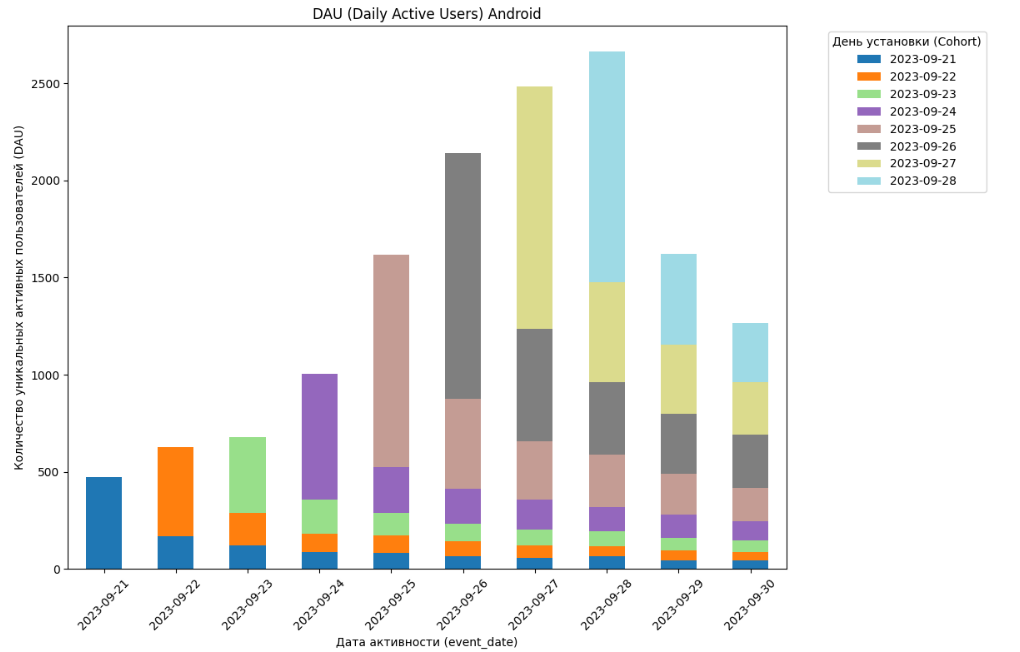
* 1. **Анализ и визуализация активности пользователей (DAU)**

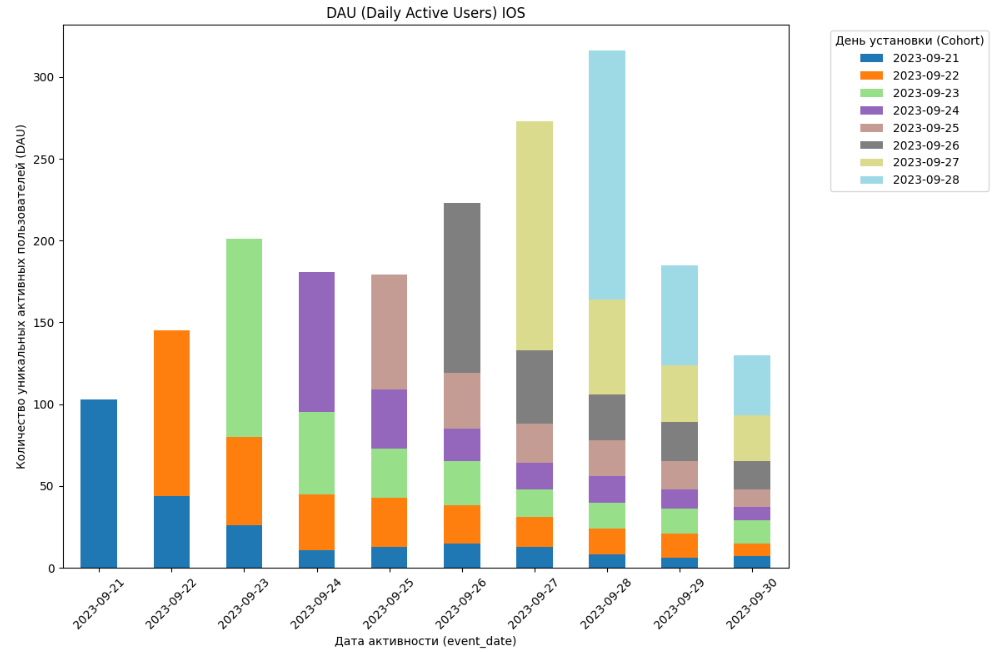
DAU (Daily Active Users) — это метрика, показывающая количество уникальных пользователей, которые взаимодействовали с приложением в течение одного дня.

DAU используется для оценки активности пользователей и популярности приложения на ежедневной основе. Эта метрика помогает понять:

* Как часто пользователи возвращаются в приложение.
* Насколько эффективно приложение удерживает пользователей.
* Какие дни показывают пики активности.

DAU важна для анализа удержания (Retention Rate) и расчёта средней выручки на активного пользователя (ARPU).





На графиках показано распределение DAU по дням активности (event\_date) для каждой платформы (Android и iOS). Столбцы разделены по когортам (день установки приложения). Каждая цветная часть столбца представляет пользователей конкретной когорты. Сложение всех частей даёт общее количество активных пользователей за день.

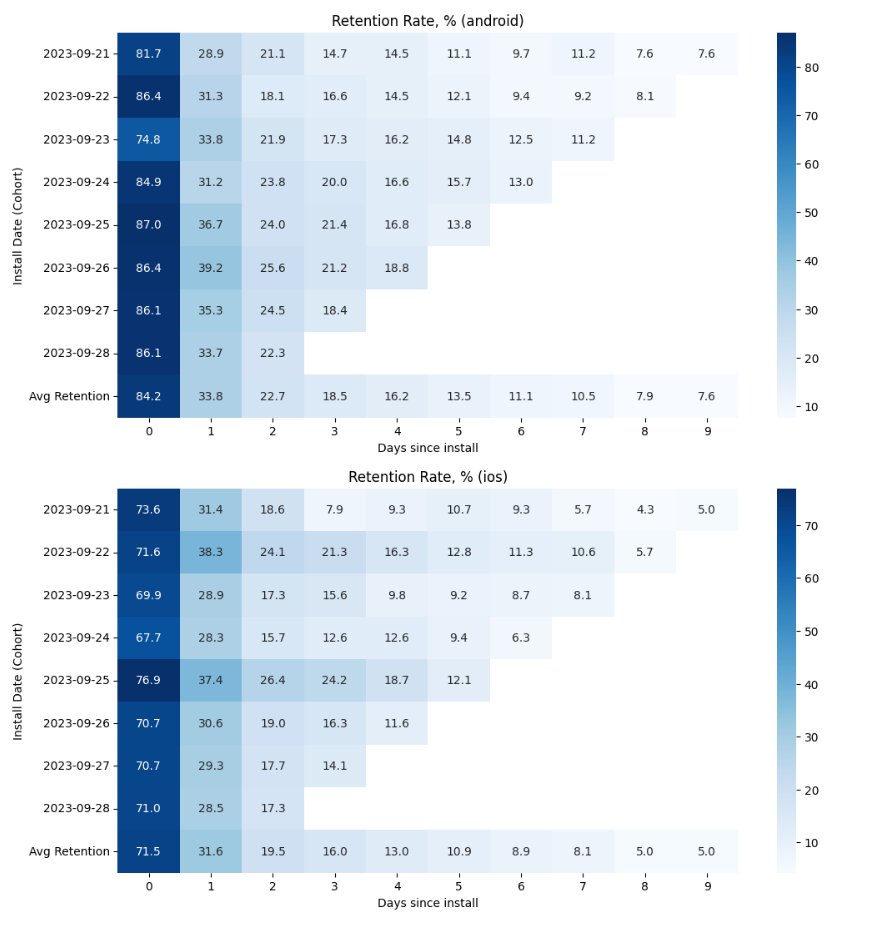
**Общие выводы по метрике**

* Дисбаланс между платформами: Android имеет значительно большую базу пользователей, что сильно влияет на DAU.
* Вклад когорт: Когорты с большими размерами (особенно 26-27 сентября) дают основной вклад в ежедневную активность.
* Удержание: У обеих платформ наблюдается постепенное снижение DAU после пиков активности, что свидетельствует о выбывании пользователей с течением времени.
  1. **Анализ и визуализация удержания пользователей (Retention)**

Retention Rate (Коэффициент удержания) — это процент пользователей, которые продолжают пользоваться приложением через определённое количество дней после установки.

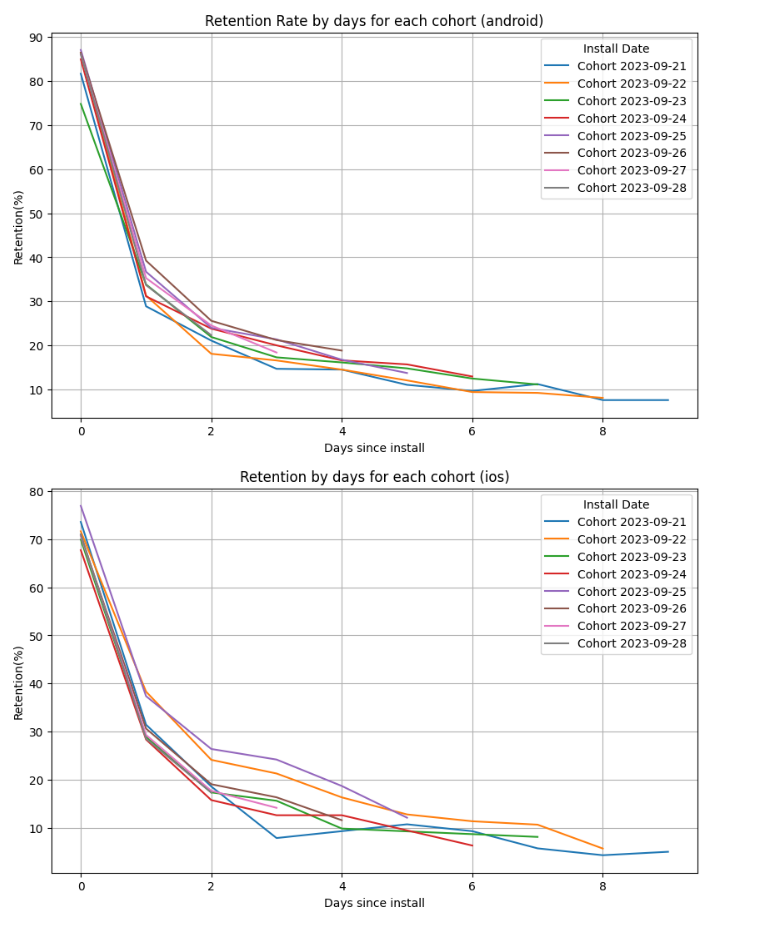
Эта метрика:

* Оценивает способность приложения удерживать пользователей.
* Помогает анализировать поведение когорт с течением времени.
* Используется для определения лояльности пользователей и прогнозирования долгосрочной активности.



Таблицы (тепловые карты) показывают коэффициенты удержания пользователей по дням для каждой когорты (по дате установки).

* Столбцы: количество дней с момента установки (Days since install).
* Строки: когорты по дате установки (Install Date).
* Значения: процент пользователей, которые остались активными на конкретный день.



Линейные графики показывают динамику удержания для каждой когорты отдельно. Каждая линия представляет когортное удержание пользователей с первого дня (Day 0) до девятого дня (Day 9).

**Общие выводы по метрике**

Android:

* Начальный Retention Rate высокий (~84.2% на Day 0), с постепенным снижением до ~7.6% на Day 9.
* Удержание стабилизируется после третьего дня, что характерно для мобильных приложений.

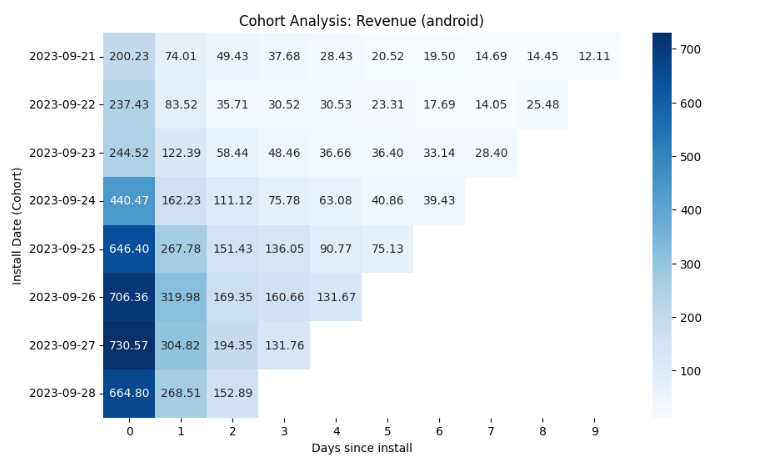
iOS:

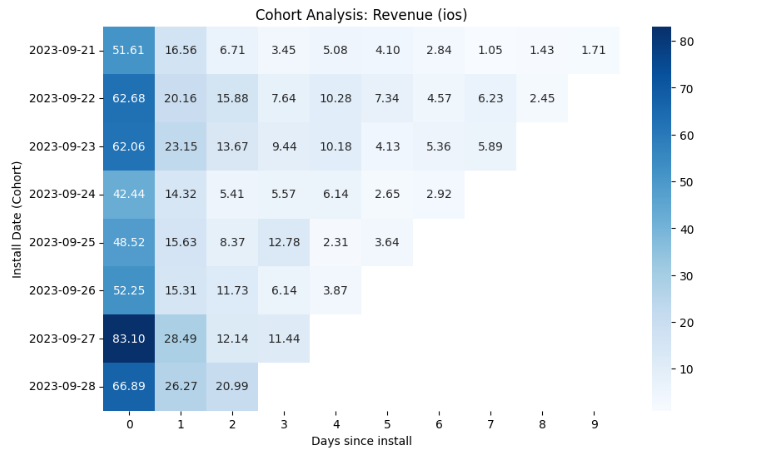
* Начальный Retention Rate ниже (~71.5% на Day 0) и достигает ~5.0% на Day 9.
* Динамика удержания также стабилизируется после первых дней.

Общие выводы:

* Такая динамика удержания типична для мобильных приложений: большинство пользователей теряется в первые несколько дней.
* Разработке следует сосредоточиться на улучшении пользовательского опыта в первые три дня, чтобы повысить Retention Rate.
* Когортный анализ не выявил значительных различий между когортами. Это указывает на стабильную динамику удержания среди всех когорт. Такое поведение говорит о том, что ключевые факторы удержания пользователей одинаково влияют на всех, независимо от даты установки.
* На iOS кривые удержания когорт накладываются меньше, чем на Android. Это связано с меньшим размером выборки когорт, что увеличивает влияние случайных факторов на динамику удержания.
  1. **Анализ выручки (Revenue)**

Revenue: это сумма дохода, полученного за определённый период. В данном случае — ежедневная выручка, сгруппированная по когортам (дата установки и дни с момента установки). Когортный анализ выручки помогает выявить наиболее успешные когорты и дни, приносящие наибольший доход.





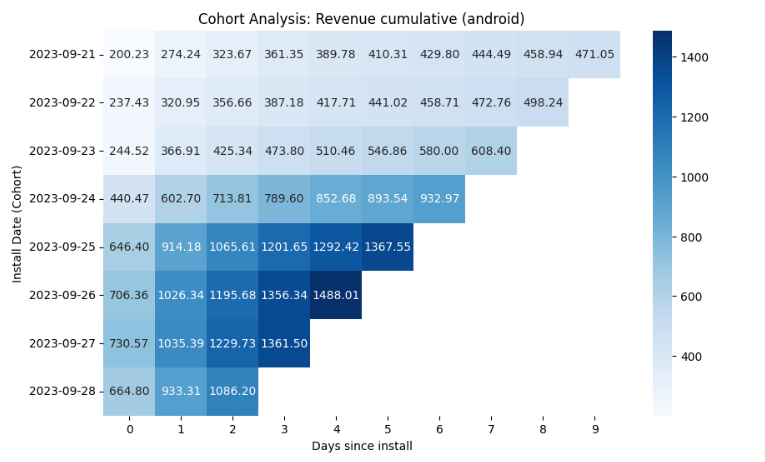
**Общие выводы по метрике**

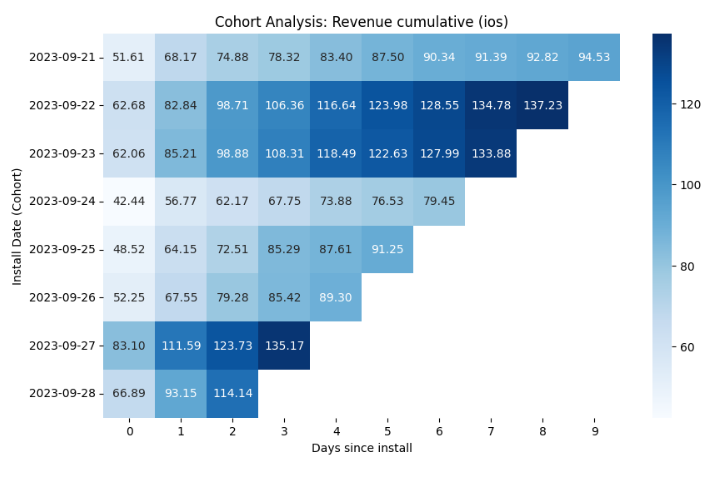
* Android генерирует больше выручки, что связано с большими размерами когорт.
* На iOS выручка с одного пользователя может быть выше, но её абсолютные значения ниже из-за меньшего числа пользователей.
  1. **. Анализ нарастания выручки (Cumulative Revenue)**

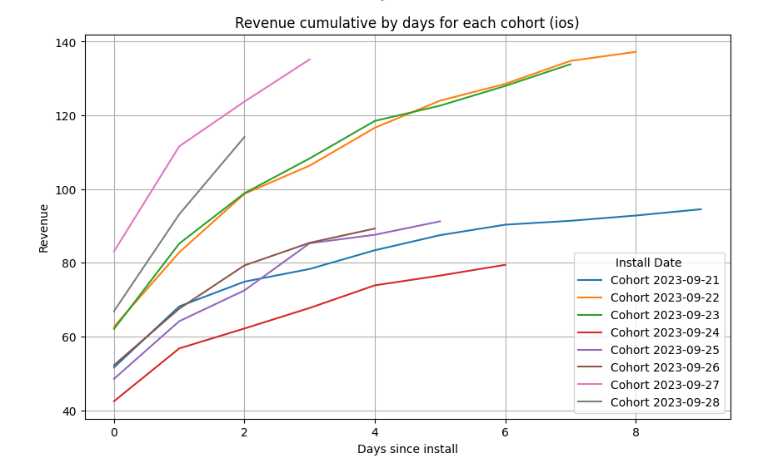
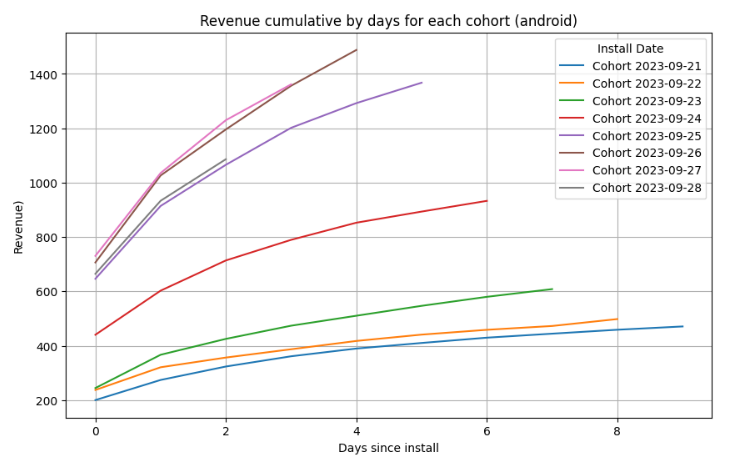
Cumulative Revenue (Кумулятивная выручка): это показатель, отражающий накопленную выручку когорты пользователей с момента их установки. Рассчитывается как сумма ежедневной выручки за весь период.

Зачем нужна:

* Показывает, как растёт доход каждой когорты со временем.
* Позволяет оценить общий вклад каждой когорты в общую выручку.
* Помогает выявить ключевые периоды монетизации (например, первые дни после установки).







**Выводы по кумулятивной выручке (Cumulative Revenue)**

Динамика роста:

* На обеих платформах кумулятивная выручка наиболее активно растёт в первые три дня после установки. После этого наблюдается замедление темпов роста, особенно после Day 5 на Android и Day 6 на iOS.
* Для Android максимальная кумулятивная выручка достигает ~$1488 на Day 9 (когорта 2023-09-26), для iOS — ~$137.23 (когорта 2023-09-22).

Различия между платформами:

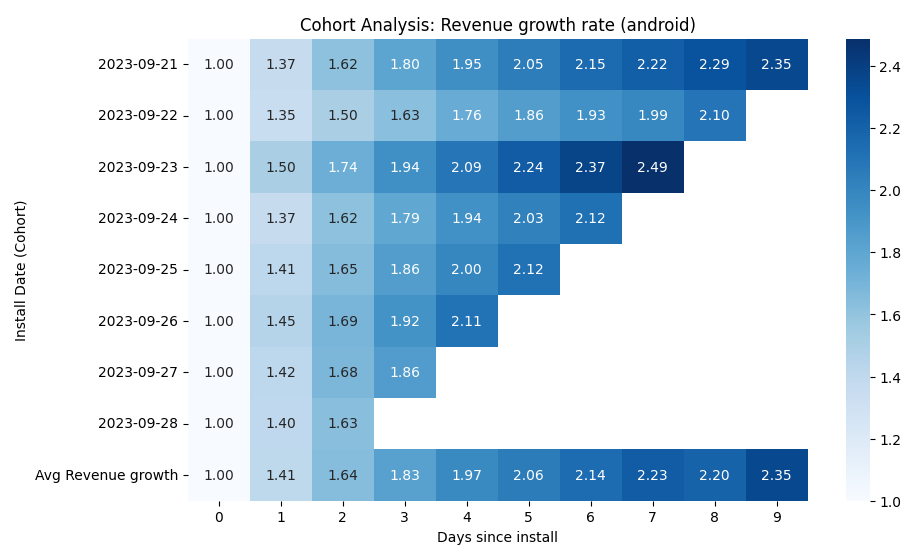
* Android показывает значительно более высокую кумулятивную выручку из-за большего количества пользователей, но рост замедляется на более поздних днях.
* На iOS выручка стабилизируется быстрее, достигая плато к Day 6–7, что связано с меньшим размером когорт.
  1. **Коэффициэнт роста выручки (Revenue Growth Rate)**

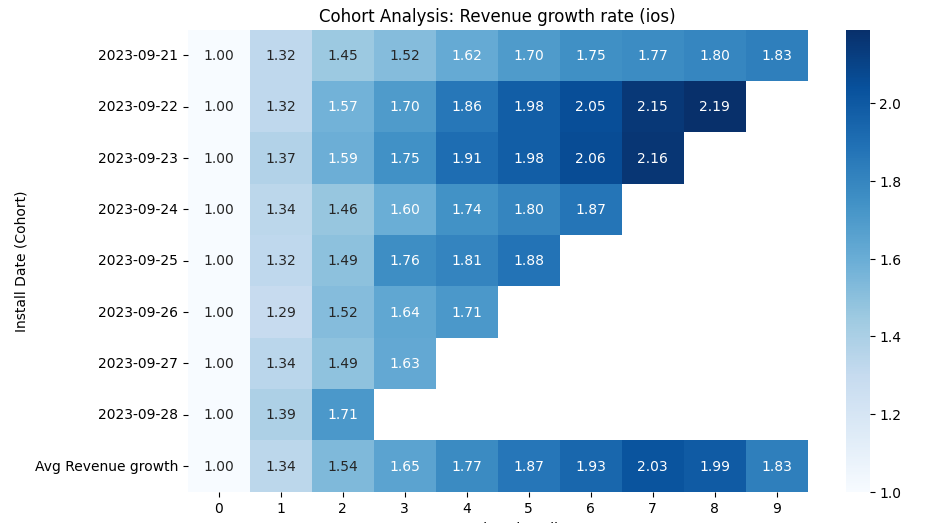
Кумулятивная выручка наглядно показывает общий доход, который приносит каждая когорта за весь период. Однако из-за разного размера когорт (особенно значительного дисбаланса между Android и iOS) сложно объективно оценить, насколько эффективно монетизируется каждая когорта.

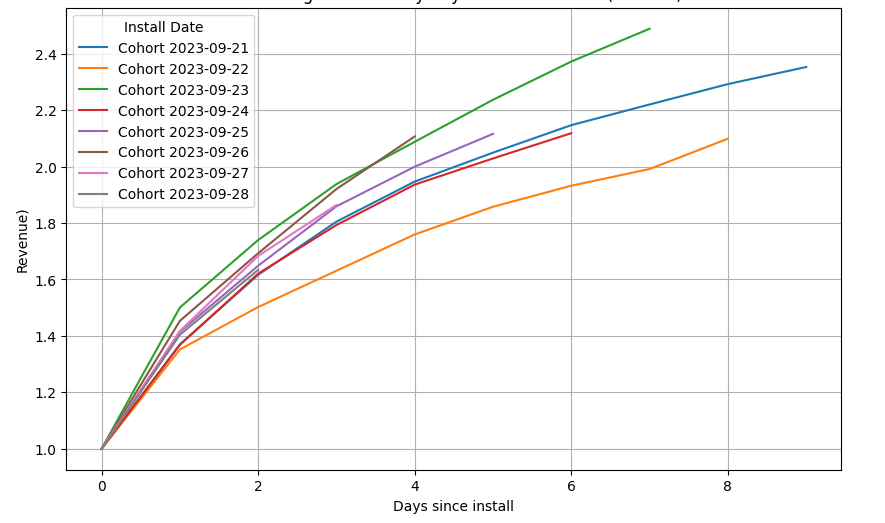
Чтобы нивелировать влияние размера когорт, используется коэффициент роста выручки, который отражает относительный прирост дохода когорт.

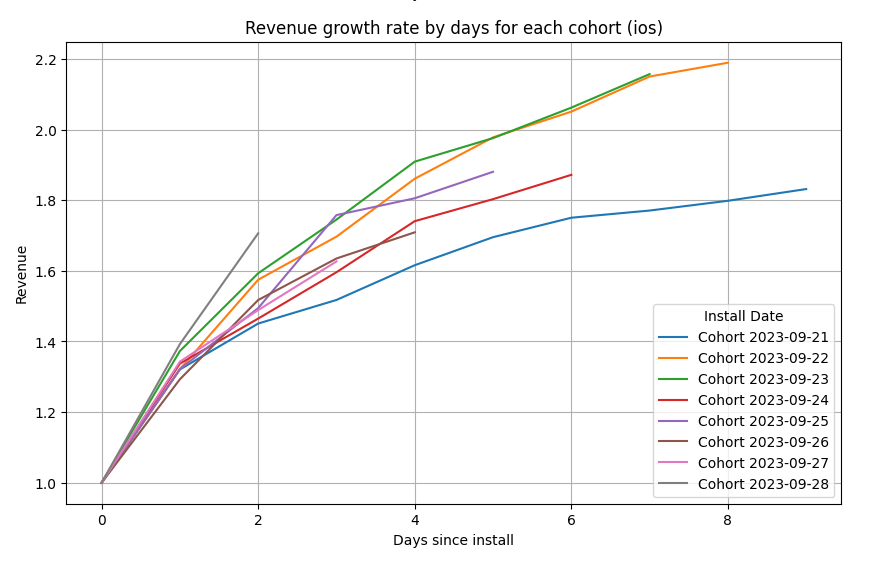
Этот показатель помогает оценить:

* Динамику увеличения дохода в зависимости от времени.
* Сравнительную эффективность монетизации когорт независимо от их размеров.









**Общие выводы по метрике**

Сравнение платформ:

* Android показывает более высокий средний коэффициент роста (~2.35 против ~1.83 на iOS) благодаря большему числу пользователей и более эффективной монетизации.
* На iOS коэффициент роста выручки стабилизируется быстрее, что может говорить о раннем выгорании пользовательской активности.

Ключевые моменты:

* Коэффициент роста выручки позволяет нивелировать эффект различия размеров когорт, предоставляя более объективное понимание эффективности монетизации.
* На обеих платформах наиболее активный рост выручки происходит в первые 3–5 дней, после чего динамика замедляется.
  1. **Средний доход на пользователя**

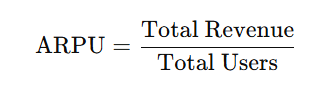
ARPU (Средняя выручка на пользователя): это показатель, отражающий средний доход, генерируемый одним активным пользователем. Рассчитывается как отношение выручки за день к количеству уникальных активных пользователей (DAU).

Зачем нужна:

* ARPU позволяет оценить эффективность монетизации на уровне пользователя.
* Сравнение ARPU между когортами и платформами помогает выявить успешные стратегии монетизации.

**Общий ARPU**

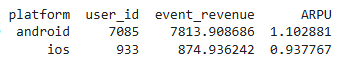
Рассчитан средний доход на пользователя за весь период для каждой платформы. Формула:



Общий ARPU позволяет сравнить общую эффективность монетизации пользователей на разных платформах за весь период. Помогает оценить общий доход на пользователя независимо от динамики по дням.

Результаты:

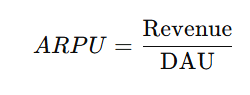
* Android: $1.10 на пользователя за весь период.
* iOS: $0.94 на пользователя за весь период.



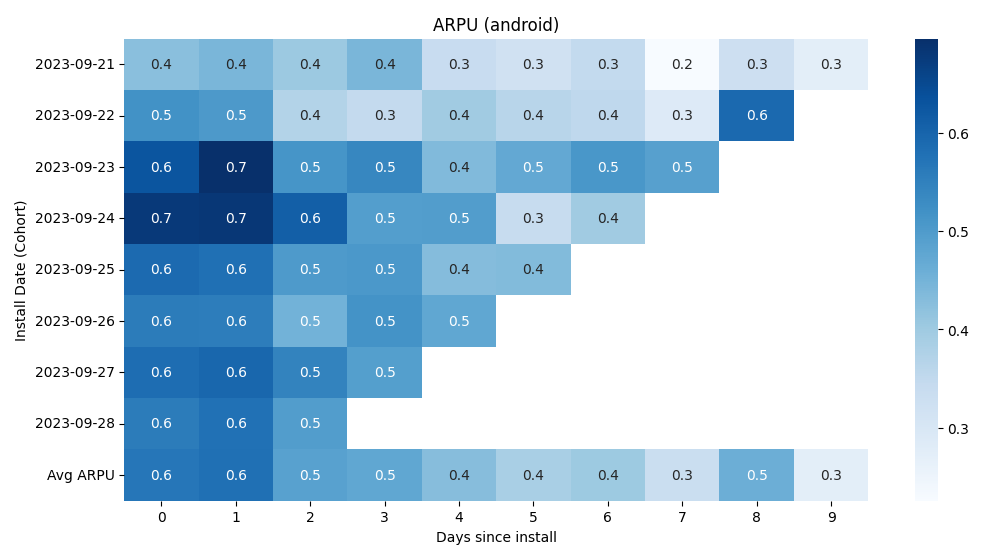
Android демонстрирует более высокий общий ARPU ($1.10 против $0.94 на iOS), что может быть связано с более высокой активностью пользователей или лучшими условиями монетизации.

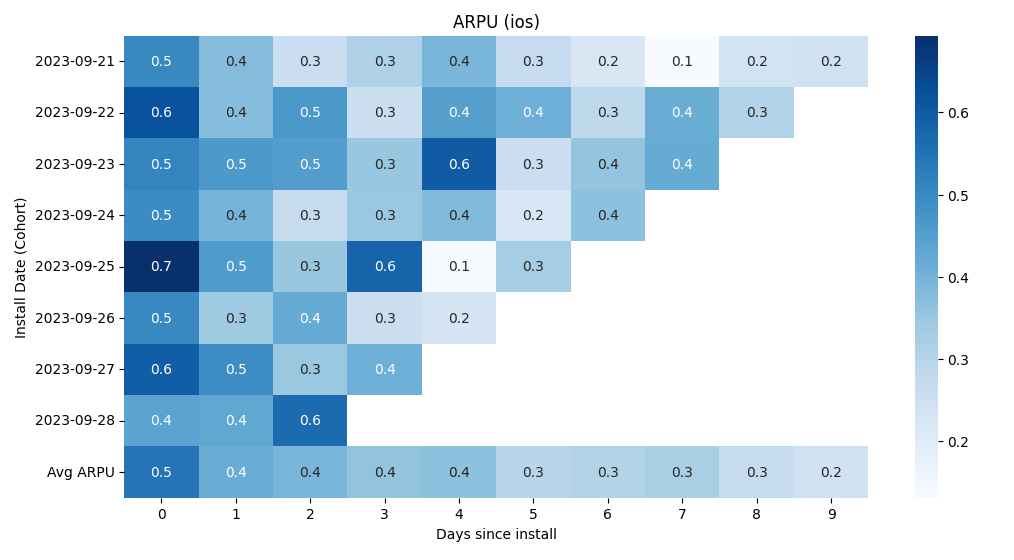
**ARPU по дням**

Формула:



ARPU по дням показывает, как пользователи взаимодействуют с приложением в разные периоды их жизненного цикла. Это особенно важно для определения оптимальных точек монетизации.





**Общие выводы по метрике**

Сравнение платформ:

* Android показывает более высокий ARPU (~$0.6 в первые дни) по сравнению с iOS (~$0.4), что связано с более высокой активностью пользователей.
* На iOS ARPU более изменчивый, но показывает похожую тенденцию снижения в поздние дни.

Ключевые моменты:

* Стабильность ARPU в первые дни указывает на потенциал монетизации на этапе вовлечения.
* Снижение ARPU после первых 4 дней говорит о необходимости дополнительных стратегий монетизации для пользователей в поздние дни.
* Метрика ARPU (Average Revenue Per User) может быть не очень стабильной, особенно если:
  + Маленький размер выборки: на платформах с небольшим количеством активных пользователей (например, iOS) единичные активные пользователи, которые приносят значительный доход, могут сильно искажать общую картину.
  + Редкие, но крупные платежи: если некоторые пользователи делают крупные разовые платежи, это может резко увеличить ARPU за день, но не будет отражать реальную тенденцию для всей когорты.
  + Непостоянная активность: если активность пользователей варьируется по дням, это также может привести к резким изменениям в дневном ARPU.

1. **Проверка гипотез**

#### **Гипотеза 1:**

* **H₀ (нулевая гипотеза):** Выручка не зависит больше от размера когорт, чем от активности единичных пользователей.
* **H₁ (альтернативная гипотеза):** Выручка больше зависит от размера когорт, чем от активности единичных пользователей.

**Корреляционный анализ**

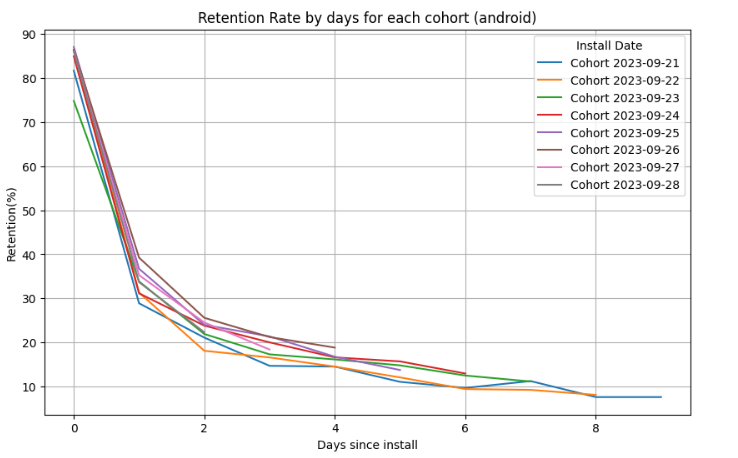
Задача: исследовать взаимосвязь между размером когорты, удержанием (Retention Rate) и выручкой, вычислить коэффициент корреляции.

1. **Прогноз активности пользователей и роста выручки**
   1. **Прогноз удержания пользователей на 30-й день**

График показателя удержания пользователей имеет схожий вид и для веб-приложений, и для мобильных приложений, и для интернет-магазинов, и для многих оффлайн-продуктов:

* в первые дни (недели, месяцы) он резко падает: пользователи не успевают пройти «onboarding» и не остаются в проекте;
* затем те, кто всё же остался, понемногу отпадают, но уже не с такой скоростью, как в самом начале;
* наконец, когда база пользователей уже сформирована, график выходит в плато, которое лишь едва снижается, и чем больше прошло времени, тем больше график похож на горизонтальную линию.

Графически график удержания выглядит примерно так:

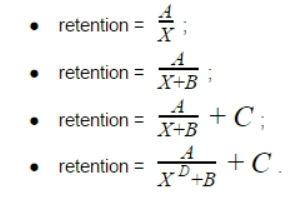


Такую кривую ещё называют «forgetting curve», потому что ею же описывается процесс забывания человеком полученной информации.

Иногда может возникнуть следующая ситуация: у вас есть значения удержания за какие-то фиксированные периоды (1 день, 7 дней, 30 дней), и вы хотите узнать значения показателя за промежуточные периоды (6 дней, 14 дней, 23 дня) или после них (35 дней). Это может пригодиться, если вы хотите прогнозировать «lifetime» или LTV (Lifetime Value), а также просто рассчитать, сколько пользователей из ныне активных будут активны в будущем.

Будем пользоваться механизмом аппроксимации, то есть приближением фактических значений математическими формулами. Делая аппроксимацию, важно, во-первых, выбрать правильную функцию (которая бы изгибалась в нужных местах) и, во-вторых, верно подобрать её коэффициенты, чтобы разница между моделью и фактом была минимальной. В этом случае предстоит смоделировать гиперболу.

Рассмотрим несколько гиперболических уравнений (X в уравнении означает номер периода: дня, недели, или месяца). Начнём с простого уравнения гиперболы A/X, затем будем усложнять его, добавляя различные коэффициенты:



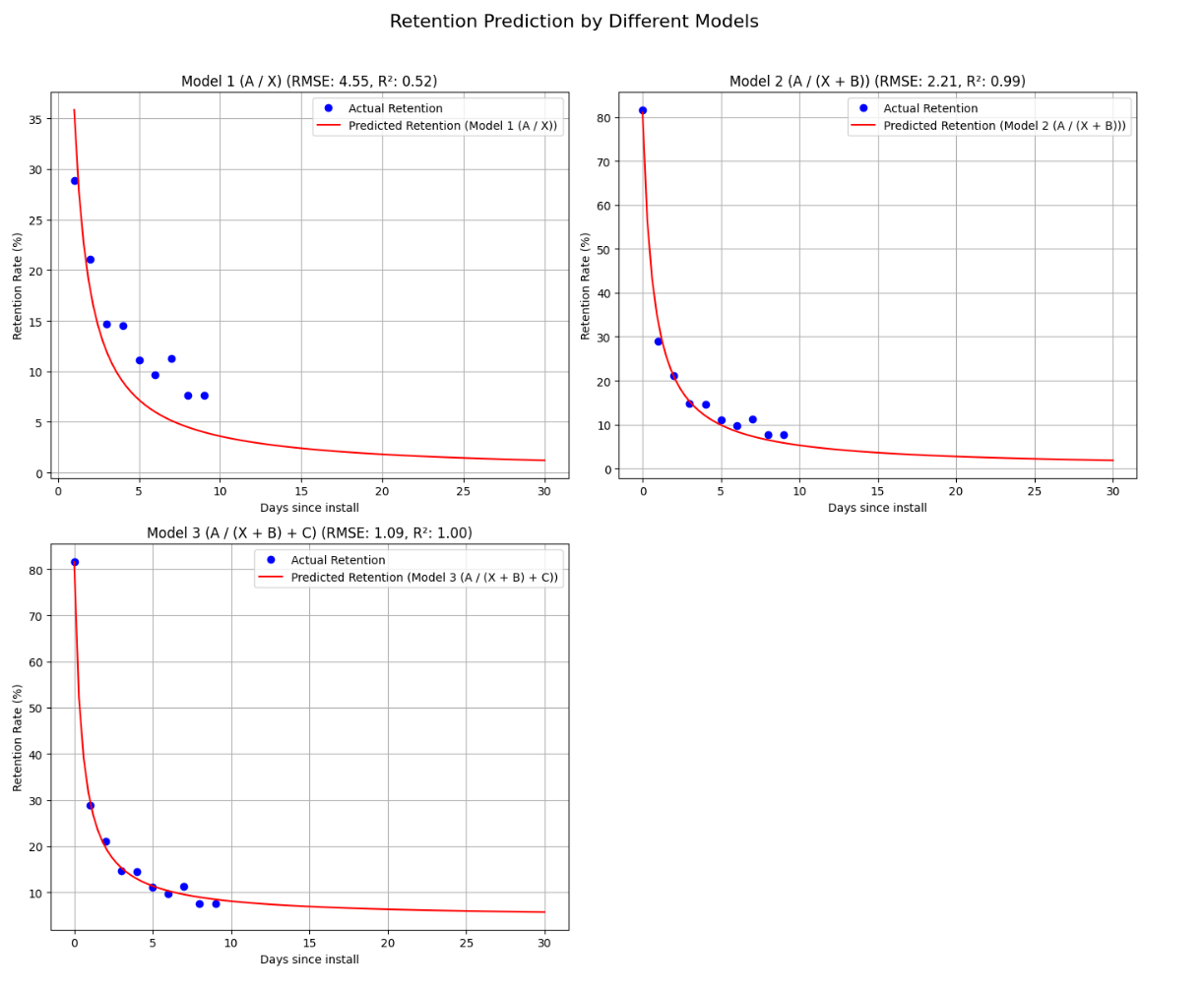
Неизвестные A, B, C – это коэффициенты, которые нам предстоит найти.

Но наша задача состоит не только в этом. Найдя в каждой из формул значения коэффициентов, мы затем должны будем выбрать самое оптимальное из этих уравнений. То есть, итог каждого из уравнений – отдельная кривая. Мы будем сравнивать эту кривую с фактическими значениями и выбирать ту из кривых, которая лучше повторяет факт.

Для оценки качества модели гиперболической аппроксимации использовался метод наименьших квадратов. Это означает, что мы минимизировали сумму квадратов разностей между фактическими значениями удержания пользователей и значениями, предсказанными моделью. В качестве метрик оценки модели использовались среднеквадратическая ошибка (RMSE) и коэффициент детерминации (R²). RMSE отражает среднее отклонение предсказаний модели от фактических данных, а R² показывает, какую долю вариации в данных объясняет модель. Высокое значение R² и низкий RMSE свидетельствуют о том, что модель точно отражает тренды данных и хорошо подходит для предсказания удержания.

Из всех когорт была выбрана первая когорта (21 сентября 2023 года) из-за её максимальной длины (9 дней), что позволяет более точно обучить модель. Модель обучается на данных первой когорты с использованием начальных предположений о коэффициентах (a=80, b=1, c=7). Для минимизации отклонений используется метод наименьших квадратов.

Каждая модель визуализируется на отдельном графике. Заголовок каждого графика включает название модели, RMSE и R2-оценку. выводится название, параметры и метрики оптимальной модели (с минимальным RMSE).



Согласно результатам, **Model 3:**

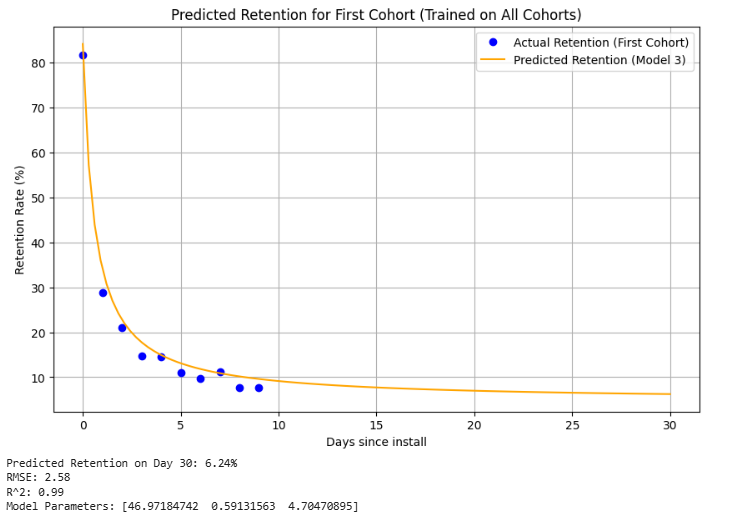


где:

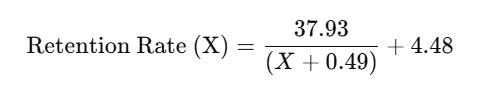
* A — масштабирующий коэффициент (определяет начальный уровень удержания),
* B — смещение по оси X (определяет, как быстро уменьшается удержание),
* C — асимптота, к которой приближается график удержания при увеличении X,
* X — номер периода (день, неделя, месяц и т. д.).

показала наилучший результат с минимальным RMSE (1.09) и R2, близким к 1.00. Это подтверждает, что модель хорошо аппроксимирует данные.

Используя параметры, полученные в результате оптимизации, модель предсказала **удержание пользователей на 30-й день**. Это значение составляет **6.24%** для первой когорты пользователей на платформе Android.



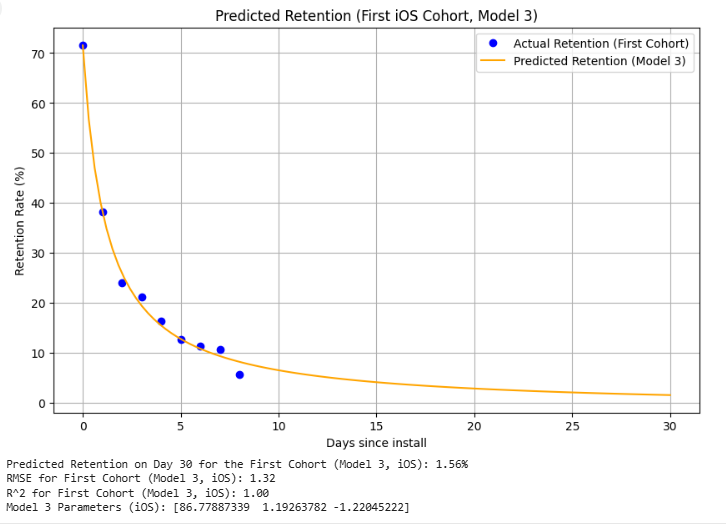
Формула удержания на Android:



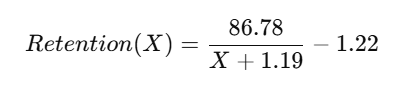
**Результаты оценки модели**

* **RMSE**: 2.58 (среднеквадратическая ошибка, показывающая среднее отклонение предсказанных значений от фактических).
* **R^2**: 0.99 (идеальное объяснение дисперсии фактических значений).

Та же гиперболическая модель была применена для анализа удержания пользователей на платформе iOS. Параметры модели для iOS отличаются от Android, но модель демонстрирует высокую точность и надёжность.



Формула удержания на iOS:



**Результаты**

1. *Android (Первая когорта, 2023-09-21):*
   * Предсказанное удержание на 30-й день: 5.72%.
   * Среднеквадратическая ошибка (RMSE): 1.09.
   * Коэффициент детерминации (R²): 1.00.
   * График показывает, что удержание пользователей резко снижается в первые дни после установки, стабилизируясь ближе к 10-му дню. Модель гиперболической аппроксимации демонстрирует высокую точность, подтвержденную R².
2. *iOS (Первая когорта, 2023-09-21):*
   * Предсказанное удержание на 30-й день: 1.56%.
   * Среднеквадратическая ошибка (RMSE): 1.32.
   * Коэффициент детерминации (R²): 1.00.
   * Удержание на iOS также падает значительно в первые дни, но предсказанное значение на 30-й день значительно ниже, чем на Android. График удержания показывает, что пользователи iOS демонстрируют более низкий уровень возврата.

**Выводы**

* Модель гиперболы показала отличное соответствие данным первой когорты, точно описав снижение удержания в первые дни и его стабилизацию к концу периода.
* Прогнозируемое значение удержания на 30-й день составляет 5.72% для Android и 1.56% для iOS, что соответствует типичным показателям для мобильных приложений.
* Использование самой длинной когорты обосновано, так как она содержит больше точек данных для обучения модели, что повышает её точность.