Проект по метрикам

Описание задачи: Вы работаете продуктовым аналитиком в сервисе по доставке продуктов на дом (приложение на ios и на android).

Вы настроили фронтовую аналитику в AppMetrica, в конце квартала маркетинг-менеджер попросил вас проанализировать поведение пользователей и оценить эффективность каналов их привлечения.

Выгрузите данные из AppMetrica за период с 1 января по 31 марта 2020, только по пользователям, зарегистрированным позднее 1 января 2020.

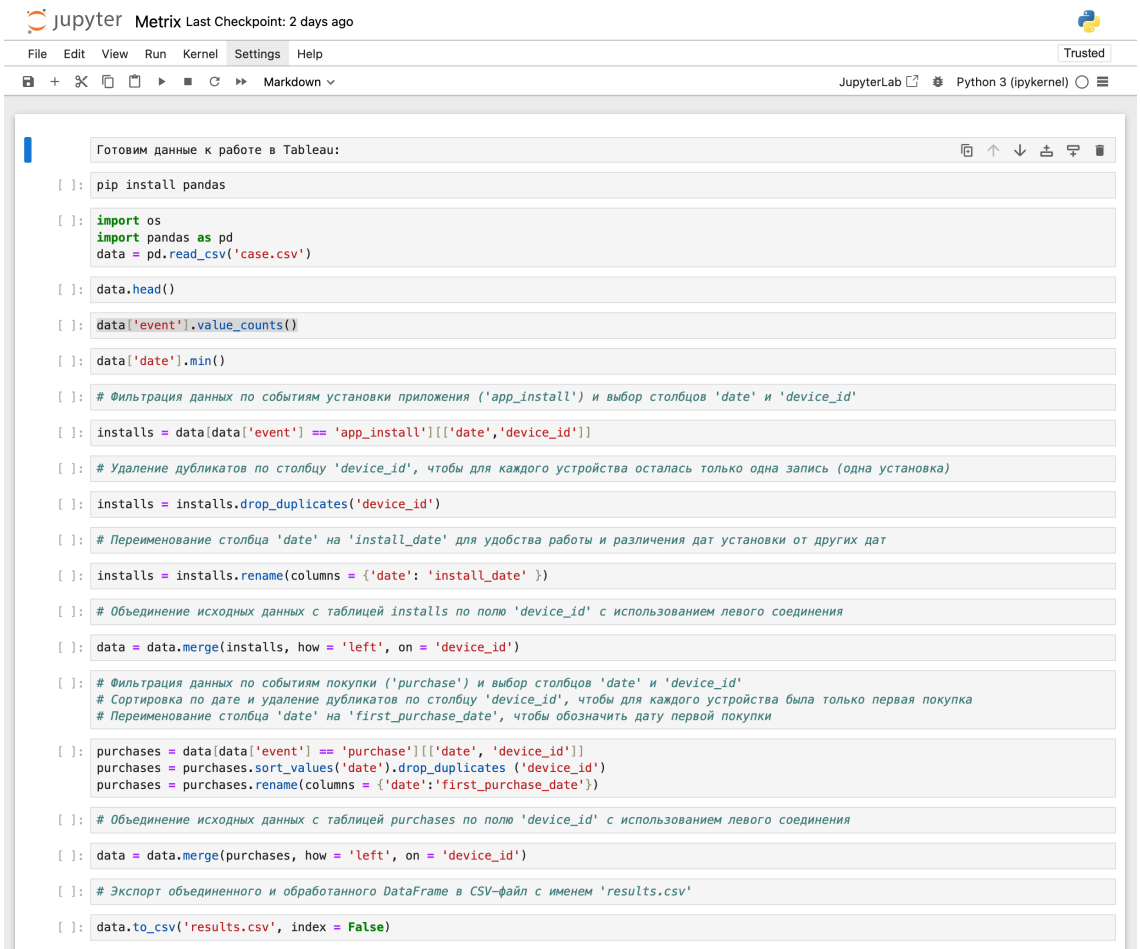
Инструменты: Python3, Tableau

Решение:

1. Выгрузили данные в формате csv и открыли **Jupiter** для преобразования данных с помощью **Python3.**

2. Вычислим **DAU января, февраля и марта.**

2.1. Преобразуем данные в Jupiter для дальнейшей работы в Tableau



Всё преобразилось и выполняется:

[1]:
```
pip install pandas
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in /Users/anyashatkova/Library/Python/3.9/lib/python/site-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /Users/anyashatkova/Library/Python/3.9/lib/python/site-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/anyashatkova/Library/Python/3.9/lib/python/site-packages (from pandas) (2.9.
0.post0)
Requirement already satisfied: pytz>=2020.1 in /Users/anyashatkova/Library/Python/3.9/lib/python/site-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /Users/anyashatkova/Library/Python/3.9/lib/python/site-packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in /Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.
9/site-packages (from python-dateutil>=2.8.2->pandas) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

[2]:
```
import os
import pandas as pd
data = pd.read_csv('case.csv')
```

[3]:
```
data.head()
```

[3]:

	date	event	purchase_sum	os_name	device_id	gender	city	utm_source
0	2020-01-01	app_start	NaN	android	669460	female	Moscow	-
1	2020-01-01	app_start	NaN	ios	833621	male	Moscow	vk_ads
2	2020-01-01	app_start	NaN	android	1579237	male	Saint-Petersburg	referal
3	2020-01-01	app_start	NaN	android	1737182	female	Moscow	facebook_ads
4	2020-01-01	app_start	NaN	ios	4029024	female	Moscow	facebook_ads

[4]:
```
data['event'].value_counts()
```

[4]:
```
event
app_start      748705
search         708639
choose_item    538669
tap_basket     377665
app_install    154597
purchase       141383
register        78310
Name: count, dtype: int64
```

[5]:
```
data['date'].min()
```

[5]:
```
'2020-01-01'
```

[6]:
```
# Фильтрация данных по событиям установки приложения ('app_install') и выбор столбцов 'date' и 'device_id'
```

[7]:
```
installs = data[data['event'] == 'app_install'][['date','device_id']]
```

[8]:
```
# Удаление дубликатов по столбцу 'device_id', чтобы для каждого устройства осталась только одна запись (одна установка)
```

[9]:
```
installs = installs.drop_duplicates('device_id')
```

[10]:
```
# Переименование столбца 'date' на 'install_date' для удобства работы и различения дат установки от других дат
```

[11]:
```
installs = installs.rename(columns = {'date': 'install_date' })
```

[12]:
```
# Объединение исходных данных с таблицей installs по полю 'device_id' с использованием левого соединения
```

[13]:
```
data = data.merge(installs, how = 'left', on = 'device_id')
```

[14]:
```
# Фильтрация данных по событиям покупки ('purchase') и выбор столбцов 'date' и 'device_id'
# Сортировка по дате и удаление дубликатов по столбцу 'device_id', чтобы для каждого устройства была только первая покупка
# Переименование столбца 'date' на 'first_purchase_date', чтобы обозначить дату первой покупки
```

[15]:
```
purchases = data[data['event'] == 'purchase'][['date', 'device_id']]
purchases = purchases.sort_values('date').drop_duplicates ('device_id')
purchases = purchases.rename(columns = {'date':'first_purchase_date'})
```

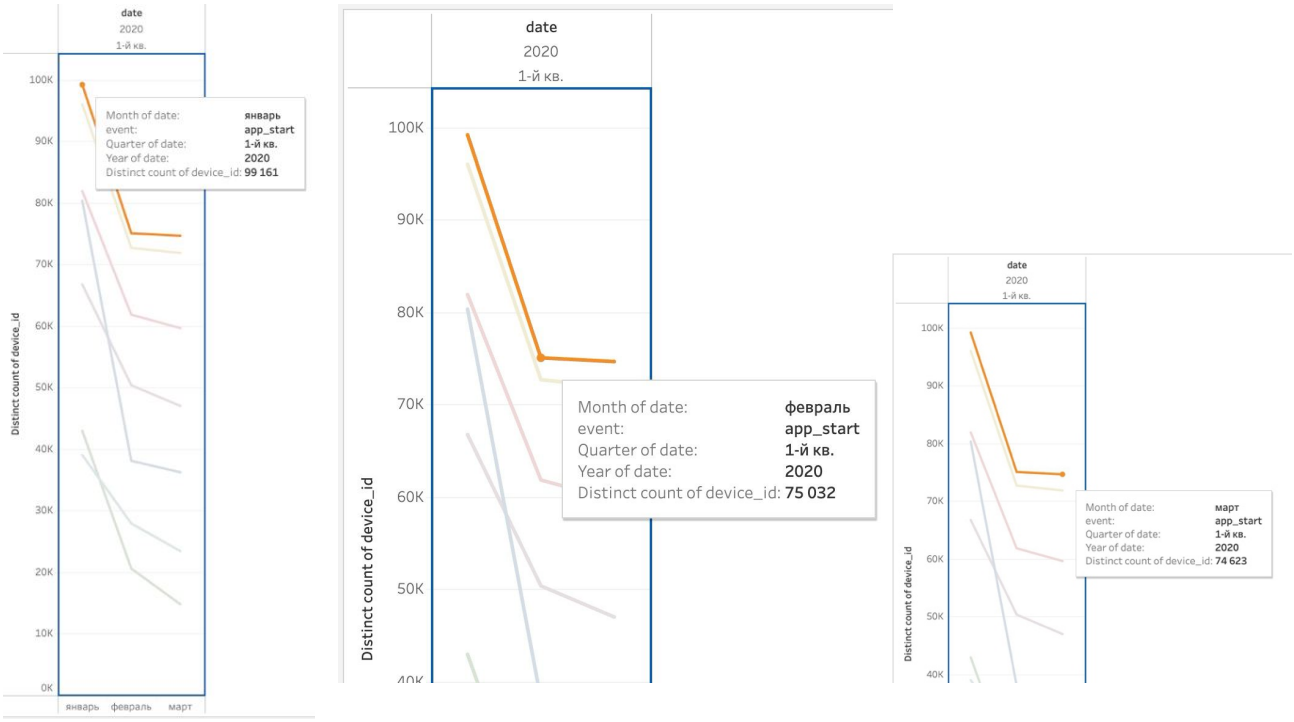
[16]:
```
# Объединение исходных данных с таблицей purchases по полю 'device_id' с использованием левого соединения
```

[17]:
```
data = data.merge(purchases, how = 'left', on = 'device_id')
```

[18]:
```
# Экспорт объединенного и обработанного DataFrame в CSV-файл с именем 'results.csv'
```

[19]:
```
data.to_csv('results.csv', index = False)
```

2.2. Переходим в **Tableau**:



Анализ данных:

1. Январь (99 191 пользователей):

Январь показывает наибольшее количество активных пользователей. Это связано с новогодними праздниками, когда многие пользователи активно пользуются доставкой продуктов, чтобы справиться с увеличенными потребностями в еде для праздников или из-за предпочтения удобств в праздники.

2. Февраль (75 032 пользователей):

В феврале наблюдается значительное снижение активности пользователей по сравнению с январем - около 24%. Это может быть связано с тем, что январь — это обычно "пиковый" месяц для покупок, а в феврале наблюдается снижение активности, когда люди возвращаются к обычному ритму жизни.

3. Март (74 623 пользователей):

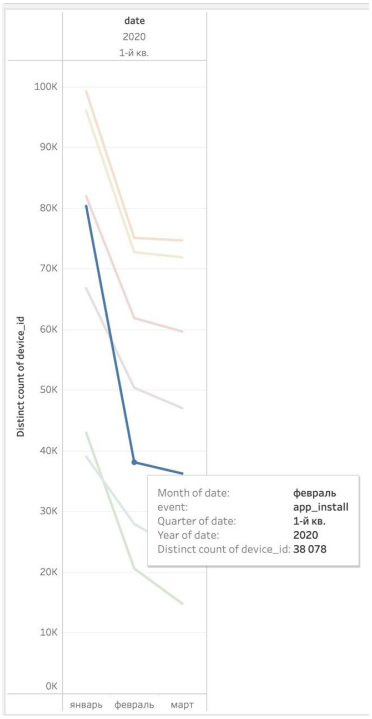
В марте активность стабилизируется, но на уровне, который значительно ниже январского пика.

Выводы:

Сезонные колебания: Январский пик активности связан с сезонностью, а в феврале и марте наступает спад после праздников.

Потребность в улучшении маркетинга и удержания: Резкое падение в феврале и стабилизация на низком уровне могут быть сигналом для усиления маркетинговых кампаний или внедрения программ лояльности, чтобы мотивировать пользователей продолжать пользоваться сервисом даже после праздников.

3. Вычислим количество установок



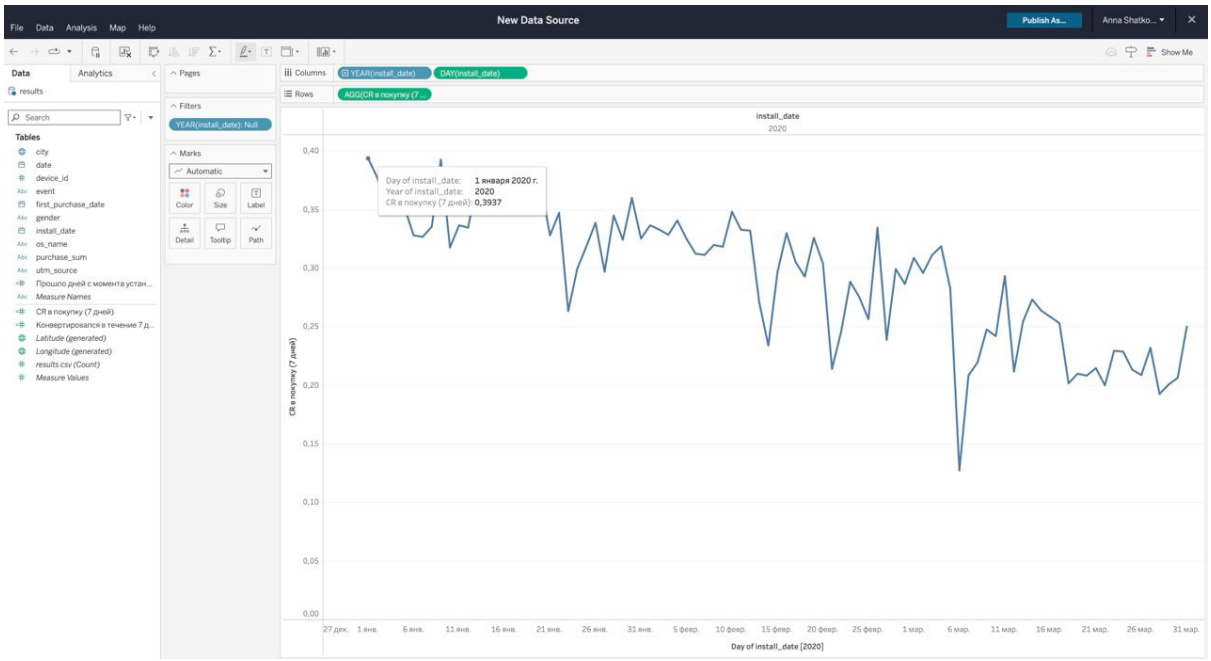
Наблюдается корреляция с данными MAU:

1. Установок приложения в феврале значительно меньше, чем в январе, что и отражается в резком падении MAU.
2. В марте количество установок продолжает снижаться, но в меньшей степени, что также можно увидеть в данных по MAU, где активные пользователи остаются практически на том же уровне.

Выводы:

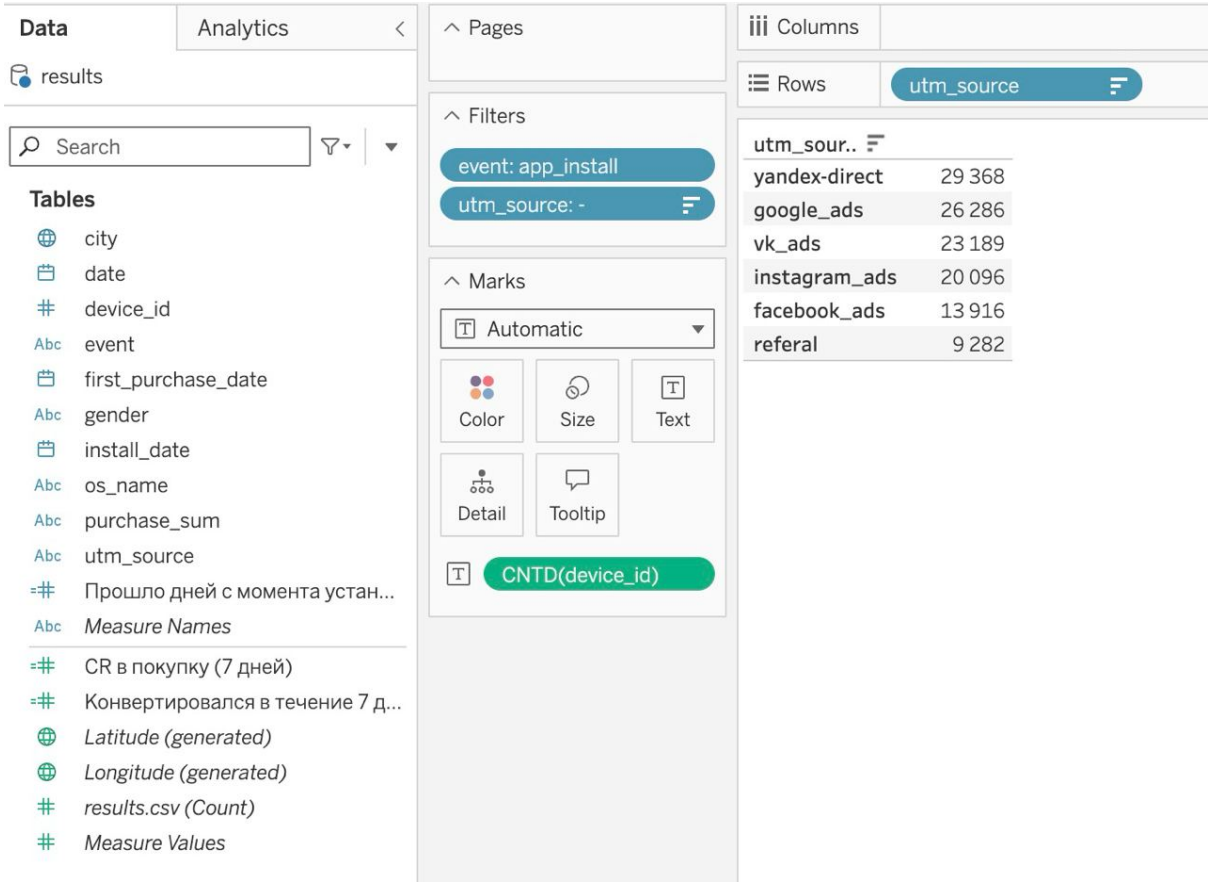
1. **Причина падения MAU**: Снижение MAU в феврале может быть связано с уменьшением количества новых установок, что подтверждается графиком. Это говорит о том, что новые пользователи играют важную роль в поддержании общего числа активных пользователей.
2. **Необходимость улучшения привлечения:** Снижение MAU и установок сигнализирует о необходимости улучшения стратегии привлечения новых пользователей, маркетинговых кампаний или других факторов, которые могли бы стимулировать рост базы пользователей.

4. Присвоим пользователям **когорты по дню установки приложения** и посчитаем для них **конверсию из установки в покупку в течение 7 дней.**



Самая высокая конверсия была в первый день (1 января).
Возможные причины:
1. Начало года и праздники
2. Релиз приложения
3. Маркетинг

5. Выясним **с какого платного маркетингового канала пришло больше всего новых пользователей.**



Большее кол-во пользователей пришло с канала Yandex.direct, а наименьшее с реферальной программы.

Это связано с мощным охватом **Yandex.direct**, более активными маркетинговыми кампаниями, таргетированной рекламой и более широким распространением рекламы. Реферальная программа зависит от активности существующих пользователей, что объясняет её меньший вклад в привлечение новых клиентов.

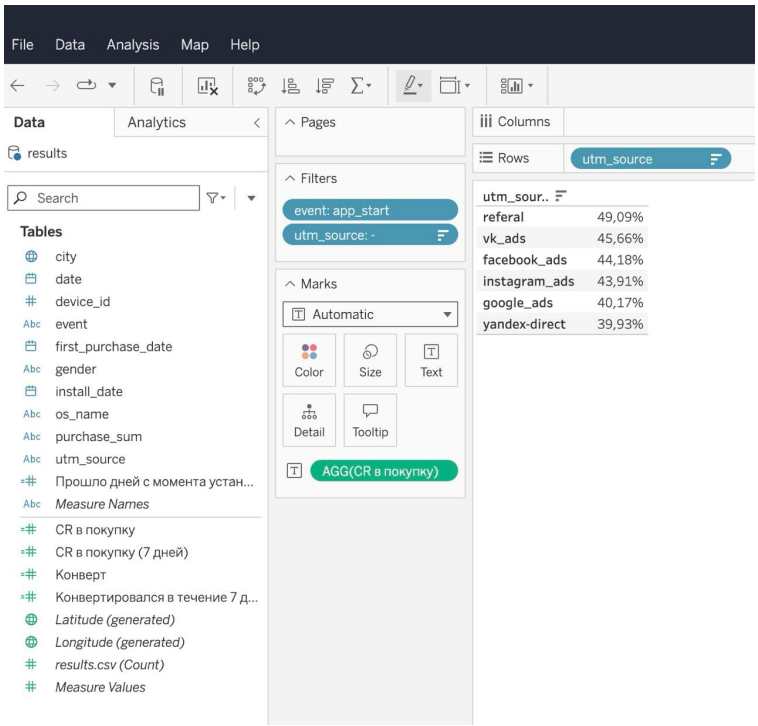
6. Проанализируем **на каком этапе воронки отваливается бОльшая часть клиентов.** Посмотрим отдельно сценарии для зарегистрированных и для незарегистрированных пользователей.

Анализ конверсии на этапах воронки

```
[20]: # Определение первой регистрации устройства и добавление даты первой регистрации в исходные данные

[21]: first_registration = data[data['event']=='register'].sort_values('date').drop_duplicates('device_id')
      data ['first_registration'] = data['device_id'].map(first_registration.set_index('device_id')['date'])

[22]: # Фильтрация уже зарегистрированных устройств:

[23]: already_reg = data[data['first_registration'] < data['date']]

[24]: # Подсчет уникальных устройств по событиям:

[25]: already_reg.groupby('event')['device_id'].nunique()

[25]: event
      app_start      40991
      choose_item    37926
      purchase       24880
      search         40482
      tap_basket     34517
      Name: device_id, dtype: int64

[26]: print ("Конверсия в поиск из открытия", round ((40482/40991)*100,2))

      Конверсия в поиск из открытия 98.76

[27]: print ("Конверсия в добавление товара из поиска", round ((37926/40482) *100,2))

      Конверсия в добавление товара из поиска 93.69

[28]: print ("Конверсия в переход в корзину из добавления товара", round((34517/37926) * 100,2))

      Конверсия в переход в корзину из добавления товара 91.01

[29]: print ("Конверсия в покупку из перехода в корзину", round( (24880/34517) * 100,2))

      Конверсия в покупку из перехода в корзину 72.08
```

Конверсия хорошая. Наименьшая на этапе в покупку из перехода в корзину, что обычно.

7. Выясним пользователи, пришедшие **с каких каналов**, показали **самую низкую конверсию** в первую покупку.



Вывод: Yandex.direct показал наименьшую конверсию, а реферальная программа - наибольшую.

Пользователи, пришедшие по рекомендациям, более вероятно становятся клиентами из-за их более высокого уровня доверия к продукту.
На основе данных можно принять решение о перераспределении бюджета между различными каналами. Например, вложение большего бюджета в реферальную программу может принести более высокую отдачу в долгосрочной перспективе, несмотря на меньший объем трафика.

8. Выясним пользователи, пришедшие **с какого канала, имеют медианный первый чек выше?** (учитываются только первые покупки пользователей)

Медианный первый чек:

```
[30]: purchases = data[data['event'] == 'purchase'][['date','device_id','purchase_sum','utm_source']]
      purchases = purchases.sort_values('date').drop_duplicates('device_id')

[31]: # Применяем median только к числовому столбцу 'purchase_sum', группируя по 'utm_source'
      purchases.groupby('utm_source')['purchase_sum'].median()

[31]: utm_source
      -                398.5
      facebook_ads     389.0
      google_ads       390.5
      instagram_ads    393.5
      referal          395.5
      vk_ads           393.0
      yandex-direct    392.5
      Name: purchase_sum, dtype: float64
```

Вывод: **Реферальная программа** имеет медианный первый чек выше, что подтверждает необходимость перераспределения бюджета в её пользу.

9. Выясним **какой платный канал привлечения имеет самый высокий ROMI?**

Самый высокий ROMI:

[32]:
```
data.groupby('utm_source')['purchase_sum'].sum()
```

[32]:
```
utm_source
-                21449749.5
facebook_ads     12249901.0
google_ads       12868276.0
instagram_ads    14546969.0
referal           8837044.5
vk_ads           16389652.5
yandex-direct    13915368.0
Name: purchase_sum, dtype: float64
```

[33]:
```
print ("ROMI Facebook:", round ((12249901/8590498-1)*100,2),"%")
print ("ROMI Гугл:", round ((12868276/10534878-1)*100,2), "%")
print ("ROMI Яндекс:", round ((13915368/10491707-1)*100,2),"%")
print ("ROMI Инстаграм:", round ((14546969/8561626-1)*100,2), "%")
print ("ROMI ВК:", round((16389652/9553531-1)*100,2), "%")
```

```
ROMI Facebook: 42.6 %
ROMI Гугл: 22.15 %
ROMI Яндекс: 32.63 %
ROMI Инстаграм: 69.91 %
ROMI ВК: 71.56 %
```

Самая высокая метрика отдачи вложенных инвестиций в маркетинг у VK. То есть VK самый оправданный по расходам канал маркетинга. На каждые вложеные 1 у.е. получаем 71 у.е..

Аналитические выводы

1. Существуют сезонные колебания
2. Каналы привлечения пользователей:

Yandex.Direct привлек больше пользователей, но имеет низкую конверсию и меньший медианный чек.

Реферальная программа привлекает меньше пользователей, но показывает высокую конверсию и медианный чек.

VK демонстрирует наивысшую отдачу вложенных инвестиций, что делает его наиболее эффективным каналом для маркетинга с точки зрения ROI (возврат на инвестиции).

Рекомендации для Повышения Метрик и Устойчивого Роста:

1. Улучшение маркетинга и удержания пользователей:

- **Внедрение программ лояльности, чтобы мотивировать пользователей делать заказы чаще.** Это может быть система бонусов, скидок на повторные заказы или эксклюзивные предложения для постоянных клиентов.
- **Усиление маркетинговых кампаний в не сезон, чтобы поддержать интерес.** Это может включать таргетированную рекламу, сезонные предложения или скидки.

2. Оптимизация каналов привлечения:

- **Инвестирование в реферальную программу.** Т.к. реферальная программа показывает высокую конверсию и медианный чек, стоит увеличить инвестиции в этот канал. Реферальные пользователи имеют тенденцию быть более ценными, так как они приходят по рекомендациям довольных клиентов (для довольности внедряем программы лояльности, скидки).
- **Анализ и улучшение в Yandex.Direct**. Оптимизация рекламы в Yandex.Direct. Улучшение таргетинга, тестирования рекламы, добавление прогрессивных промокодов на скидки (чем выше чек, тем выше скидка). Это будет стимулировать людей увеличивать медианный чек.

3. Оптимизация процесса покупки:

- **Улучшение UX/UI на этапе оформления заказа.** Анализ и оптимизация процесса оформления заказа, чтобы уменьшить количество брошенных корзин. Упростите процесс оплаты, обеспечьте удобные методы доставки и улучшите коммуникацию с клиентами.

Эти рекомендации помогут повысить метрики, улучшить привлечение и удержание пользователей, сделать сервис более стабильным и привлекательным в течение всего года.