**Исследование данных популярного сервиса аренды самокатов GoFast.**

Описание проекта.

Я аналитик популярного сервиса аренды самокатов GoFast. Мне передали данные о некоторых пользователях из нескольких городов, а также об их поездках. Необходимо проанализировать данные и проверить некоторые гипотезы, которые могут помочь бизнесу вырасти. Чтобы совершать поездки по городу, пользователи сервиса GoFast пользуются мобильным приложением. Сервисом можно пользоваться:

1) без подписки

* абонентская плата отсутствует;
* стоимость одной минуты поездки — 8 рублей;
* стоимость старта (начала поездки) — 50 рублей;

2) с подпиской Ultra

* абонентская плата — 199 рублей в месяц;
* стоимость одной минуты поездки — 6 рублей;
* стоимость старта — бесплатно.

Исследование включает в себя следующие этапы:

* загрузка данных;
* предобработка данных;
* исследовательский анализ данных;
* объединение данных;
* подсчёт выручки;
* проверка гипотез.

Ввод [1]:

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**from** scipy **import** stats **as** st

​

​

**1  Шаг 1. Загрузка данных**

Ввод [2]:

users **=** pd.read\_csv('/datasets/users\_go.csv')

rides **=** pd.read\_csv('/datasets/rides\_go.csv')

subscriptions **=** pd.read\_csv('/datasets/subscriptions\_go.csv')

​

Ввод [3]:

users.head()

Out[3]:

|  | **user\_id** | **name** | **age** | **city** | **subscription\_type** |
| --- | --- | --- | --- | --- | --- |
| **0** | 1 | Кира | 22 | Тюмень | ultra |
| **1** | 2 | Станислав | 31 | Омск | ultra |
| **2** | 3 | Алексей | 20 | Москва | ultra |
| **3** | 4 | Константин | 26 | Ростов-на-Дону | ultra |
| **4** | 5 | Адель | 28 | Омск | ultra |

Ввод [4]:

rides.head()

Out[4]:

|  | **user\_id** | **distance** | **duration** | **date** |
| --- | --- | --- | --- | --- |
| **0** | 1 | 4409.919140 | 25.599769 | 2021-01-01 |
| **1** | 1 | 2617.592153 | 15.816871 | 2021-01-18 |
| **2** | 1 | 754.159807 | 6.232113 | 2021-04-20 |
| **3** | 1 | 2694.783254 | 18.511000 | 2021-08-11 |
| **4** | 1 | 4028.687306 | 26.265803 | 2021-08-28 |

Ввод [5]:

subscriptions.head()

Out[5]:

|  | **subscription\_type** | **minute\_price** | **start\_ride\_price** | **subscription\_fee** |
| --- | --- | --- | --- | --- |
| **0** | free | 8 | 50 | 0 |
| **1** | ultra | 6 | 0 | 199 |

Ввод [6]:

users.info()

rides.info()

subscriptions.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1565 entries, 0 to 1564

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 user\_id 1565 non-null int64

1 name 1565 non-null object

2 age 1565 non-null int64

3 city 1565 non-null object

4 subscription\_type 1565 non-null object

dtypes: int64(2), object(3)

memory usage: 61.3+ KB

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 18068 entries, 0 to 18067

Data columns (total 4 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 user\_id 18068 non-null int64

1 distance 18068 non-null float64

2 duration 18068 non-null float64

3 date 18068 non-null object

dtypes: float64(2), int64(1), object(1)

memory usage: 564.8+ KB

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2 entries, 0 to 1

Data columns (total 4 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 subscription\_type 2 non-null object

1 minute\_price 2 non-null int64

2 start\_ride\_price 2 non-null int64

3 subscription\_fee 2 non-null int64

dtypes: int64(3), object(1)

memory usage: 192.0+ bytes

Ввод [7]:

subscriptions.isna().sum()

Out[7]:

subscription\_type 0

minute\_price 0

start\_ride\_price 0

subscription\_fee 0

dtype: int64

Ввод [8]:

users.isna().sum()

Out[8]:

user\_id 0

name 0

age 0

city 0

subscription\_type 0

dtype: int64

Ввод [9]:

rides.isna().sum()

Out[9]:

user\_id 0

distance 0

duration 0

date 0

dtype: int64

При беглом осмотре, можно сказать, что в данных отсутствуют пропуски и проблемы с типами данных; нам удалось получить 3 небольших датафрейма, колонки в которых пересекаются по значениям (например по user\_id и subscription\_type).

**2  Шаг 2. Предобработка данных**

Приведем столбец date к типу даты pandas для удобства. Создаю новый столбец month для дальнейших рассчетов с ним.

Ввод [10]:

rides['date'] **=** pd.to\_datetime(rides['date'])

rides['month'] **=** rides['date'].dt.month

*#rides*

rides['date'].min(), rides['date'].max()

Out[10]:

(Timestamp('2021-01-01 00:00:00'), Timestamp('2021-12-30 00:00:00'))

Мы имеем данные за календарный 2021 год. Ранее установлено отсутствие пропусков в данных. Проверю их на наличие дубликатов:

Ввод [11]:

users.duplicated().sum()

​

​

Out[11]:

31

Ввод [12]:

rides.duplicated().sum()

Out[12]:

0

Ввод [13]:

subscriptions.duplicated().sum()

Out[13]:

0

Ввод [14]:

users **=** users.drop\_duplicates()

Дубликаты найдены только в датафрейме users (31строка) и удалены. в остальных датафреймах ноль дубликатов.

**3  Шаг 3. Исследовательский анализ данных**

Ввод [15]:

user\_counts **=** pd.pivot\_table(users, index**=**'city', values**=**'user\_id', aggfunc**=**'count')

user\_counts **=** user\_counts.sort\_values(by**=**'user\_id', ascending**=False**)

user\_counts

Out[15]:

|  | **user\_id** |
| --- | --- |
| **city** |  |
| **Пятигорск** | 219 |
| **Екатеринбург** | 204 |
| **Ростов-на-Дону** | 198 |
| **Краснодар** | 193 |
| **Сочи** | 189 |
| **Омск** | 183 |
| **Тюмень** | 180 |
| **Москва** | 168 |

Ввод [16]:

*# код ревьюера*

users['city'].value\_counts()

Out[16]:

Пятигорск 219

Екатеринбург 204

Ростов-на-Дону 198

Краснодар 193

Сочи 189

Омск 183

Тюмень 180

Москва 168

Name: city, dtype: int64

Ввод [17]:

user\_counts **=** pd.pivot\_table(users, index**=**'city', values**=**'user\_id', aggfunc**=**'count')

​

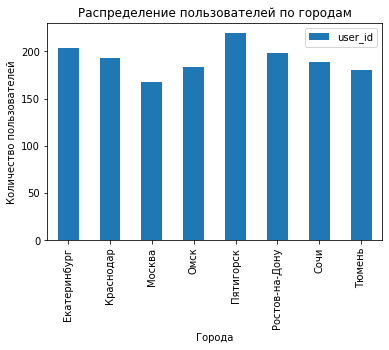
user\_counts.plot(kind**=**'bar')

plt.title('Распределение пользователей по городам')

plt.xlabel('Города')

plt.ylabel('Количество пользователей')

plt.show()



В соответствии со сводной таблицей наибольшее количество пользователей приложения проживает в таких городах как Пятигорск (наибольшее количество пользователей приложения тут), Екатеринбург (как житель этого города могу подтвердить, что количество пользователей самокатов бросается в глаза), Ростов-на-Дону, Краснодар, Сочи, Омск, Тюмень, Москва.

Сравним далее данные по их значениям.

Ввод [18]:

subscription\_counts **=** pd.pivot\_table(users, index**=**'subscription\_type', values**=**'user\_id', aggfunc**=**'count')

​

subscription\_counts.plot(kind**=**'bar')

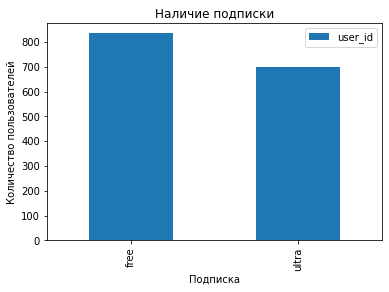
plt.title('Наличие подписки')

plt.xlabel('Подписка')

plt.ylabel('Количество пользователей')

plt.show()

​



Как мы можем видеть на графике кол-во пользователей, не оформивших подписку, больше - около 800, чем пользователей приложения, оформивших ее.

Ввод [19]:

age\_counts **=** pd.pivot\_table(users, index**=**'age', values**=**'user\_id', aggfunc**=**'count')

​

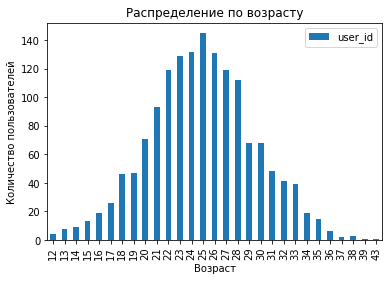
age\_counts.plot(kind**=**'bar')

plt.title('Распределение по возрасту')

plt.xlabel('Возраст')

plt.ylabel('Количество пользователей')

plt.show()



Самая многочисленная группа пользователей находится в возрасте 25 лет, самому молодому - 12, а самому старшему - 43 года (при желании можно разделить аудиторию на Подростков и Взрослых).

Ввод [20]:

ax **=** rides['distance'].plot(kind**=**'hist', bins**=**100, title**=**'Расстояние, которое пользователь преодолел за одну поездку', grid**=True**)

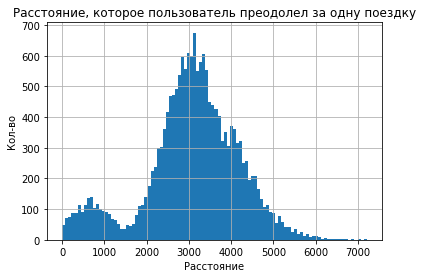
ax.set\_xlabel('Расстояние')

ax.set\_ylabel('Кол-во')

​

Out[20]:

Text(0, 0.5, 'Кол-во')



Чаще всего (в соответствии с гистограммой) пользователь преодолевает на самокате расстояние чуть больше 3 км (это значение далее будет фигурировать в проверке одной из гепотез).

Ввод [21]:

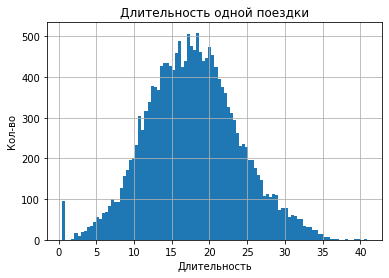
ax **=** rides['duration'].plot(kind**=**'hist', bins**=**100, title**=**'Длительность одной поездки', grid**=True**)

ax.set\_xlabel('Длительность')

ax.set\_ylabel('Кол-во')

Out[21]:

Text(0, 0.5, 'Кол-во')



На графике можно видеть небольшой пик со значение 1 мин (за его исключением распределение имеет нормальный вид). Самая распрострененная длительность поездки составила около 16 мин. Но в выборке так же есть любители "покататься" и свыше 40 мин.

**4  Шаг 4. Объединение данных**

Ввод [22]:

merged\_df **=** pd.merge(users, rides, on**=**'user\_id')

merged\_df **=** pd.merge(merged\_df, subscriptions, on**=**'subscription\_type')

merged\_df

Out[22]:

|  | **user\_id** | **name** | **age** | **city** | **subscription\_type** | **distance** | **duration** | **date** | **month** | **minute\_price** | **start\_ride\_price** | **subscription\_fee** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Кира | 22 | Тюмень | ultra | 4409.919140 | 25.599769 | 2021-01-01 | 1 | 6 | 0 | 199 |
| **1** | 1 | Кира | 22 | Тюмень | ultra | 2617.592153 | 15.816871 | 2021-01-18 | 1 | 6 | 0 | 199 |
| **2** | 1 | Кира | 22 | Тюмень | ultra | 754.159807 | 6.232113 | 2021-04-20 | 4 | 6 | 0 | 199 |
| **3** | 1 | Кира | 22 | Тюмень | ultra | 2694.783254 | 18.511000 | 2021-08-11 | 8 | 6 | 0 | 199 |
| **4** | 1 | Кира | 22 | Тюмень | ultra | 4028.687306 | 26.265803 | 2021-08-28 | 8 | 6 | 0 | 199 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **18063** | 1534 | Альберт | 25 | Краснодар | free | 3781.098080 | 19.822962 | 2021-11-04 | 11 | 8 | 50 | 0 |
| **18064** | 1534 | Альберт | 25 | Краснодар | free | 2840.423057 | 21.409799 | 2021-11-16 | 11 | 8 | 50 | 0 |
| **18065** | 1534 | Альберт | 25 | Краснодар | free | 3826.185507 | 18.435051 | 2021-11-18 | 11 | 8 | 50 | 0 |
| **18066** | 1534 | Альберт | 25 | Краснодар | free | 2902.308661 | 16.674362 | 2021-11-27 | 11 | 8 | 50 | 0 |
| **18067** | 1534 | Альберт | 25 | Краснодар | free | 2371.711192 | 15.893917 | 2021-12-29 | 12 | 8 | 50 | 0 |

18068 rows × 12 columns

Создаем ещё два датафрейма из общего merged\_df: c данными о пользователях без подписки и с данными о пользователях с подпиской.

Ввод [23]:

users\_free **=** merged\_df.loc[merged\_df['subscription\_type'] **==** 'free']

​

users\_ultra **=** merged\_df.loc[merged\_df['subscription\_type'] **==** 'ultra']

users\_ultra

Out[23]:

|  | **user\_id** | **name** | **age** | **city** | **subscription\_type** | **distance** | **duration** | **date** | **month** | **minute\_price** | **start\_ride\_price** | **subscription\_fee** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Кира | 22 | Тюмень | ultra | 4409.919140 | 25.599769 | 2021-01-01 | 1 | 6 | 0 | 199 |
| **1** | 1 | Кира | 22 | Тюмень | ultra | 2617.592153 | 15.816871 | 2021-01-18 | 1 | 6 | 0 | 199 |
| **2** | 1 | Кира | 22 | Тюмень | ultra | 754.159807 | 6.232113 | 2021-04-20 | 4 | 6 | 0 | 199 |
| **3** | 1 | Кира | 22 | Тюмень | ultra | 2694.783254 | 18.511000 | 2021-08-11 | 8 | 6 | 0 | 199 |
| **4** | 1 | Кира | 22 | Тюмень | ultra | 4028.687306 | 26.265803 | 2021-08-28 | 8 | 6 | 0 | 199 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **6495** | 699 | Оксана | 22 | Москва | ultra | 3973.046497 | 24.671163 | 2021-08-29 | 8 | 6 | 0 | 199 |
| **6496** | 699 | Оксана | 22 | Москва | ultra | 3288.702563 | 15.892886 | 2021-09-12 | 9 | 6 | 0 | 199 |
| **6497** | 699 | Оксана | 22 | Москва | ultra | 3076.159348 | 14.389525 | 2021-09-26 | 9 | 6 | 0 | 199 |
| **6498** | 699 | Оксана | 22 | Москва | ultra | 4707.845501 | 15.839579 | 2021-10-17 | 10 | 6 | 0 | 199 |
| **6499** | 699 | Оксана | 22 | Москва | ultra | 3202.890770 | 25.217345 | 2021-12-03 | 12 | 6 | 0 | 199 |

6500 rows × 12 columns

Ввод [24]:

plt.hist(users\_free['distance'], bins**=**20, alpha**=**0.5, label**=**'Free')

plt.hist(users\_ultra['distance'], bins**=**20, alpha**=**0.5, label**=**'Ultra')

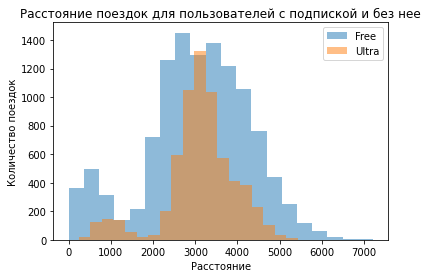
plt.legend(loc**=**'upper right')

plt.title('Расстояние поездок для пользователей с подпиской и без нее')

plt.xlabel('Расстояние')

plt.ylabel('Количество поездок')

plt.show()



Из графиков видно, что пользователи с платной подпиской ездят на самокатах на меньшее расстояние и время, чем пользователи без подписки. Возможно, это связано с тем, что пользователи с платной подпиской чаще используют самокаты для краткосрочных поездок в городе, например, для передвижения между деловыми встречами или походами в магазины, в то время как пользователи без подписки могут использовать самокаты для более длительных прогулок или спортивных тренировок. Также возможно, что пользователи с платной подпиской более ограничены в количестве времени на поездку, что приводит к более коротким поездкам на самокатах. Эту информацию мы сможем проверить ниже используя методы статистического анализа.

Ввод [25]:

plt.hist(users\_free['duration'], bins**=**20, alpha**=**0.5, label**=**'Free')

plt.hist(users\_ultra['duration'], bins**=**20, alpha**=**0.5, label**=**'Ultra')

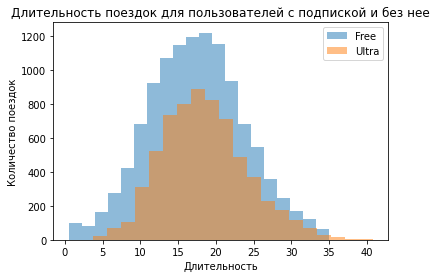
plt.legend(loc**=**'upper right')

plt.title('Длительность поездок для пользователей с подпиской и без нее')

plt.xlabel('Длительность')

plt.ylabel('Количество поездок')

plt.show()



На данном же графике мы видим, что пользователи с подпиской демонстрируют самые максимальные значения по длительности поездок (свыше 40 мин), в то время как самая максимальная длительность поездки у пользователя без подписки - около 35 мин.

**5  Шаг 5. Подсчёт выручки**

Ввод [26]:

merged\_df['duration'] **=** np.ceil(merged\_df['duration'])

​

avg\_merged\_df **=** merged\_df.pivot\_table(index**=**['user\_id', 'month'], values**=**['distance','duration'], aggfunc**=**['sum', 'count'])

**del** avg\_merged\_df['count', 'duration']

​

avg\_merged\_df.columns **=** ['distance\_monthly', 'duration\_monthly', 'rides\_number']

df **=** avg\_merged\_df.reset\_index()

*#df*

​

Создаю датафрейм avg\_merged\_df с агрегированными данными о поездках за месяц на основе датафрейма merged\_df и округляю продолжительность каждой поездки вверх, избавляюсь от старых индексов.

Ввод [27]:

subscription\_mapping **=** dict(merged\_df[['user\_id', 'subscription\_type']].values)

​

df['subscription\_type'] **=** df['user\_id'].map(subscription\_mapping)

*#df*

Создаем словарь для отображения user\_id и subscription\_type. Добавляем столбец subscription\_type в датасет df с использованием метода map().

Ввод [28]:

**def** calculate\_value(row):

**if** row['subscription\_type'] **==** 'ultra':

**return** row['duration\_monthly'] **\*** 6 **+** 199

**else**:

**return** row['duration\_monthly'] **\*** 8 **+** 50 **\*** row['rides\_number']

​

​

df['expenses'] **=** df.apply(calculate\_value, axis**=**1)

​

​

display(df)

|  | **user\_id** | **month** | **distance\_monthly** | **duration\_monthly** | **rides\_number** | **subscription\_type** | **expenses** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 1 | 7027.511294 | 42.0 | 2 | ultra | 451.0 |
| **1** | 1 | 4 | 754.159807 | 7.0 | 1 | ultra | 241.0 |
| **2** | 1 | 8 | 6723.470560 | 46.0 | 2 | ultra | 475.0 |
| **3** | 1 | 10 | 5809.911100 | 32.0 | 2 | ultra | 391.0 |
| **4** | 1 | 11 | 7003.499363 | 56.0 | 3 | ultra | 535.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **11326** | 1534 | 6 | 3409.468534 | 26.0 | 2 | free | 308.0 |
| **11327** | 1534 | 8 | 7622.453034 | 48.0 | 2 | free | 484.0 |
| **11328** | 1534 | 9 | 4928.173852 | 23.0 | 1 | free | 234.0 |
| **11329** | 1534 | 11 | 13350.015305 | 78.0 | 4 | free | 824.0 |
| **11330** | 1534 | 12 | 2371.711192 | 16.0 | 1 | free | 178.0 |

11331 rows × 7 columns

Создаю функцию для подсчета выручки по типам подписки и добавляю ее значение в новый столбец expenses.

**6  Шаг 6. Проверка гипотез**

**Важно понять, тратят ли пользователи с подпиской больше времени на поездки? Если да, то пользователи с подпиской могут быть «выгоднее» для компании. Проверьте гипотезу. Используйте исходные данные о продолжительности каждой сессии — отдельно для подписчиков и тех, у кого нет подписки.**

**H0: Нет различий в наблюдаемых средних между выборками тех кто на платной подписке и тех кто на бесплатной**

**H1: Пользователи с платной подпиской тратят больше времени на поездки**

Ввод [29]:

*# Разделение данных на две выборки*

x1 **=** users\_ultra['duration']

x2 **=** users\_free['duration']

​

*# Проверка равенства дисперсий*

st.levene(x1, x2)

​

*#Если p-value > 0.05, то можно считать, что дисперсии равны. В противном случае, можно использовать параметр equal\_var=False в функции ttest\_ind():*

*# Проверка гипотезы о равенстве средних*

​

t\_stat, p\_value **=** st.ttest\_ind(x1, x2, equal\_var**=False**, alternative**=**'greater')

​

​

*# Вывод результатов*

print("t-статистика:", t\_stat)

print("p-value:", p\_value)

​

**if** p\_value **<** 0.05:

print("Отвергаем нулевую гипотезу: пользователи с платной подпиской тратят больше времени на поездки")

**else**:

print("Не отвергаем нулевую гипотезу: нет оснований полагать, что пользователи с платной подпиской тратят больше времени на поездки")

t-статистика: 12.683043751946238

p-value: 5.6757813771289775e-37

Отвергаем нулевую гипотезу: пользователи с платной подпиской тратят больше времени на поездки

p-value < 0.05, можно отвергнуть нулевую гипотезу и сделать вывод о том, что пользователи с платной подпиской тратят больше времени на поездки.

**Расстояние одной поездки в 3130 метров — оптимальное с точки зрения износа самоката. Можно ли сказать, что расстояние, которое проезжают пользователи с подпиской за одну поездку, не превышает 3130 метров? Проверьте гипотезу и сделайте выводы.**

**H0: Нет различий между наблюдаемым средним для выборки тех кто на платной подписке по переменной "расстояние одной поездки" и 3130 метрами**

**H1: Расстояние, которое проезжают пользователи с подпиской за одну поездку, больше 3130 метров.**

Ввод [30]:

distance **=** users\_ultra['distance']

optimum\_value **=** 3130

alpha **=** 0.05

​

t\_stat, p\_value **=** st.ttest\_1samp(distance, optimum\_value, alternative**=**'greater')

​

print("T-статистика:", t\_stat)

print("p-value:", p\_value)

​

T-статистика: -1.402123166869486

p-value: 0.9195368847849785

Да, так сказать можно: статистически значимых доказательств того что среднее расстояние поездки пользователей с подпиской не равно 3130 метрам не найдено.

p-value < 0.05 (уровня статистической значимости), можно отвергнуть нулевую гипотезу и сделать вывод о том, что пользователи с платной подпиской тратят больше времени на поездки. Расстояние, которое проезжают пользователи с подпиской за одну поездку, не превышает 3130метров, таким образом не привышая оптимальное с точки зрения износа самоката, значение, что тоже является значительным плюсом этой группы пользователей.

**Проверьте гипотезу о том, будет ли помесячная выручка от пользователей с подпиской по месяцам выше, чем выручка от пользователей без подписки.**

**H0: Нет различий между помесячной выручкой от тех кто на платной подписке по переменной "расходы" и от тех, кто не имеет подписки**

**H1: Выручка по месяцам от пользователей с платной подпиской выше.**

Ввод [31]:

t\_statistic, p\_value **=** st.ttest\_ind(df[(df['subscription\_type'] **==** 'ultra')]['expenses'], df[(df['subscription\_type'] **==** 'free')]['expenses'], equal\_var**=False**, alternative **=** 'greater')

print("T-статистика:", t\_statistic)

print("p-value:", p\_value)

print("Выручка для пользователей без абонентской платы:",

df[(df['subscription\_type'] **==** 'free')]['expenses'].round(2))

print("Выручка для пользователей с абонентской платой:",

df[(df['subscription\_type'] **==** 'ultra')]['expenses'].round(2))

T-статистика: 13.201754888325674

p-value: 8.874140893976874e-40

Выручка для пользователей без абонентской платы: 4533 170.0

4534 962.0

4535 202.0

4536 178.0

4537 332.0

...

11326 308.0

11327 484.0

11328 234.0

11329 824.0

11330 178.0

Name: expenses, Length: 6798, dtype: float64

Выручка для пользователей с абонентской платой: 0 451.0

1 241.0

2 475.0

3 391.0

4 535.0

...

4528 301.0

4529 469.0

4530 385.0

4531 295.0

4532 355.0

Name: expenses, Length: 4533, dtype: float64

Да, на исследованной выборке месячная выручка с пользователя на абонентской оплате значимо выше чем месячная выручка с пользователя без абонентской платы. При уровне значимости 5% различия между двумя выборками являются значимыми. При этом, для марта p-value равно 1%: это единственный месяц для которого различия не будут значимыми на уровне значимости 1%.

**Задание 6.4: Представьте такую ситуацию: техническая команда сервиса обновила сервера, с которыми взаимодействует мобильное приложение. Она надеется, что из-за этого количество обращений в техподдержку значимо снизилось. Некоторый файл содержит для каждого пользователя данные о количестве обращений до обновления и после него. Какой тест вам понадобился бы для проверки этой гипотезы?**

**Ответ: Вероятно, сравнение выборочных средних при помощи теста Стьюдента для связанных выборок stats.ttest\_rel(data1, data2)**

Общий вывод по проекту. В ходе работы мною были описаны полученные данные, изменен тип данных там, где это было необходимо, созданы новые столбцы для дальнейших рассчетов, удалены явные дубликаты, проведен исследовательский анализ данных при помощи методов визуализации, объединены данные и разных датафреймов, подсчитана выручка по группам, имеющим подписку и нет, проверены все заявленные гипотезы.

Выводы после проведенной проверки гипотез. Пользователи с подпиской тратят больше времени на поездки: среднее время поездки для пользователей без абонентской платы составляет 17.53, в то время как среднее время поездки для пользователей с абонентской платой составляет 18.54. Таким образом, на основании результатов теста Стьюдента можно сделать вывод о том, что среднее время поездки для пользователей с абонентской платой статистически значимо отличается от среднего времени поездки для пользователей без абонентской платы. Среднее расстояние которое проезжают пользователи с подпиской равно 3130 метров. Статистически значимых доказательств того что среднее расстояние поездки пользователей с подпиской равно 3130 метрам не найдено. Будет ли помесячная выручка от пользователей с подпиской по месяцам выше, чем выручка от пользователей без подписки? Да, на исследованной выборке средняя месячная выручка с пользователя на абонентской оплате значимо выше чем средняя месячная выручка с пользователя без абонентской платы. При уровне значимости 5% различия между двумя выборками являются значимыми.

После всей проведенной работы могу рекомендовать запланировать и реализовать дополнительные акции или мероприятия для привлечения новых клиентов группу имеющих подписку в приложении, тк они потребляют предоставляемую услугу в большем объему, при этом не изнашивают оборудование, что является несомненным плюсом для бизнеса в стадии роста.