

ТИНЬКОФФ КЭШБЭК



→ Хлебникова Мария
Прохорова Нелли
Непорада Анастасия
Заиц Рената



Предварительная обработка

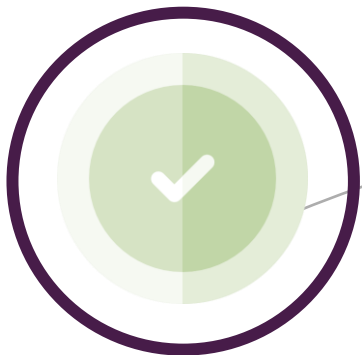
Анализ данных

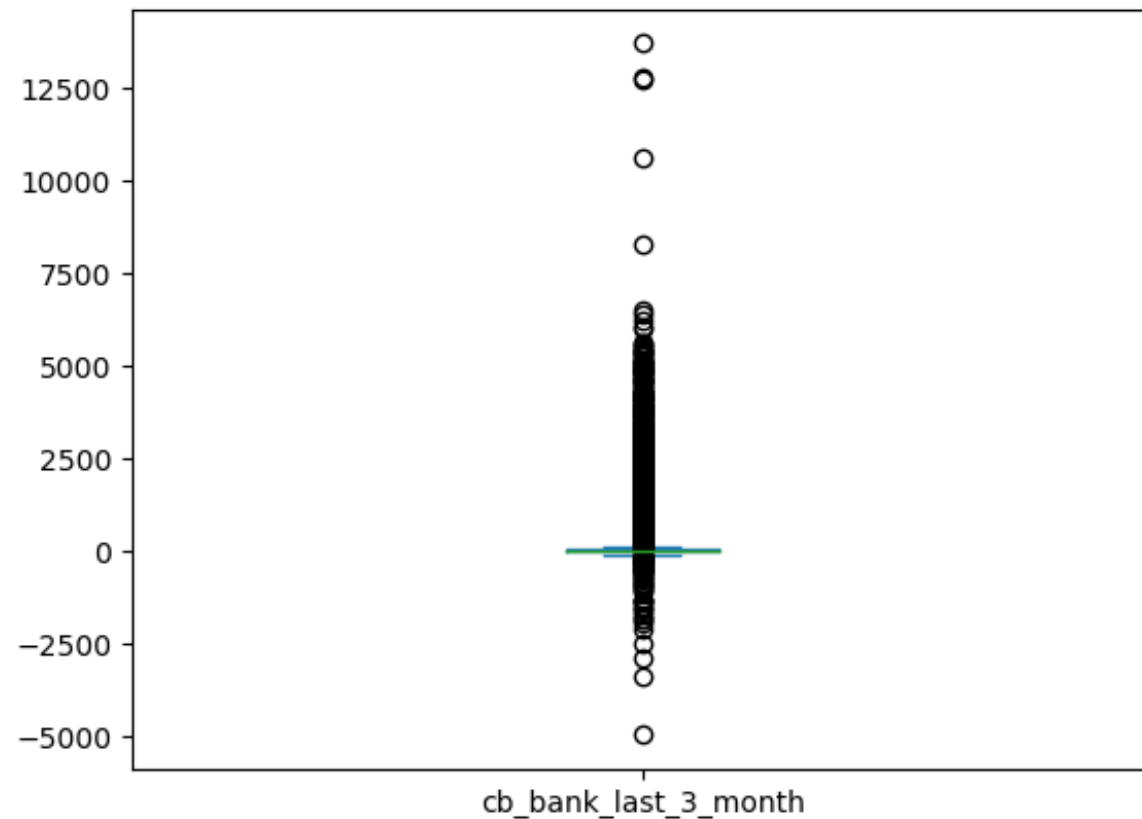
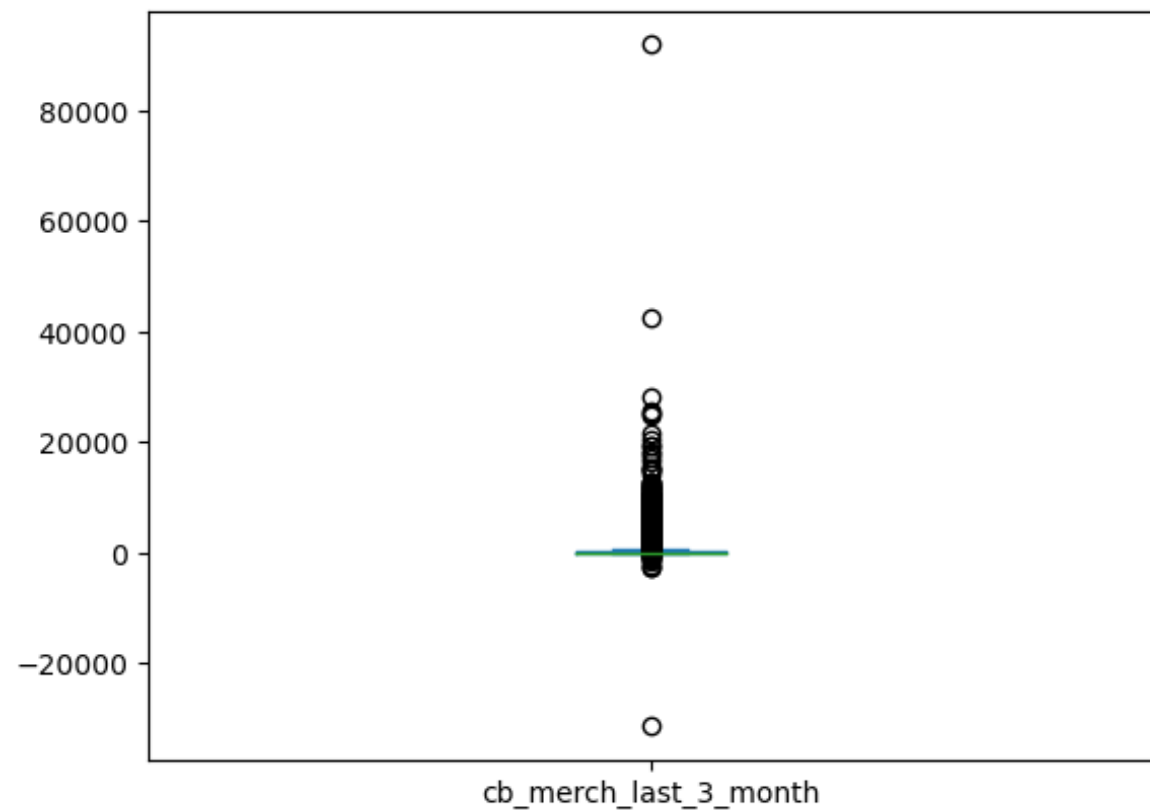
Выводы

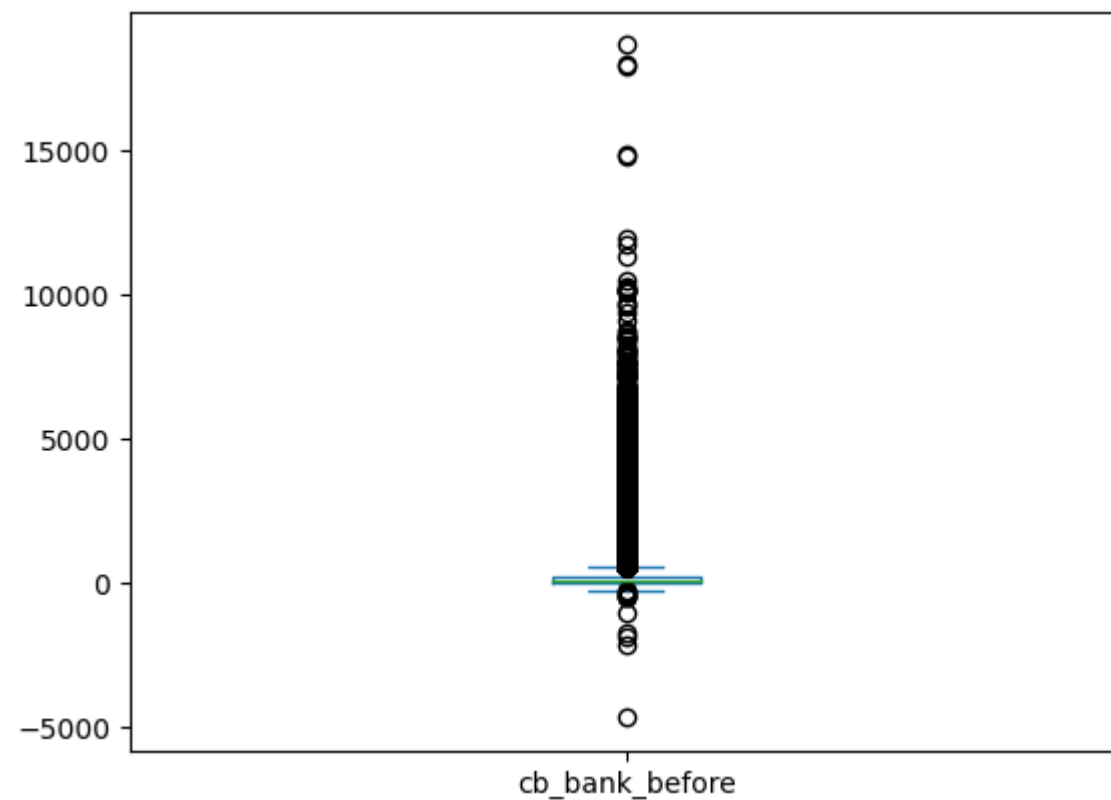
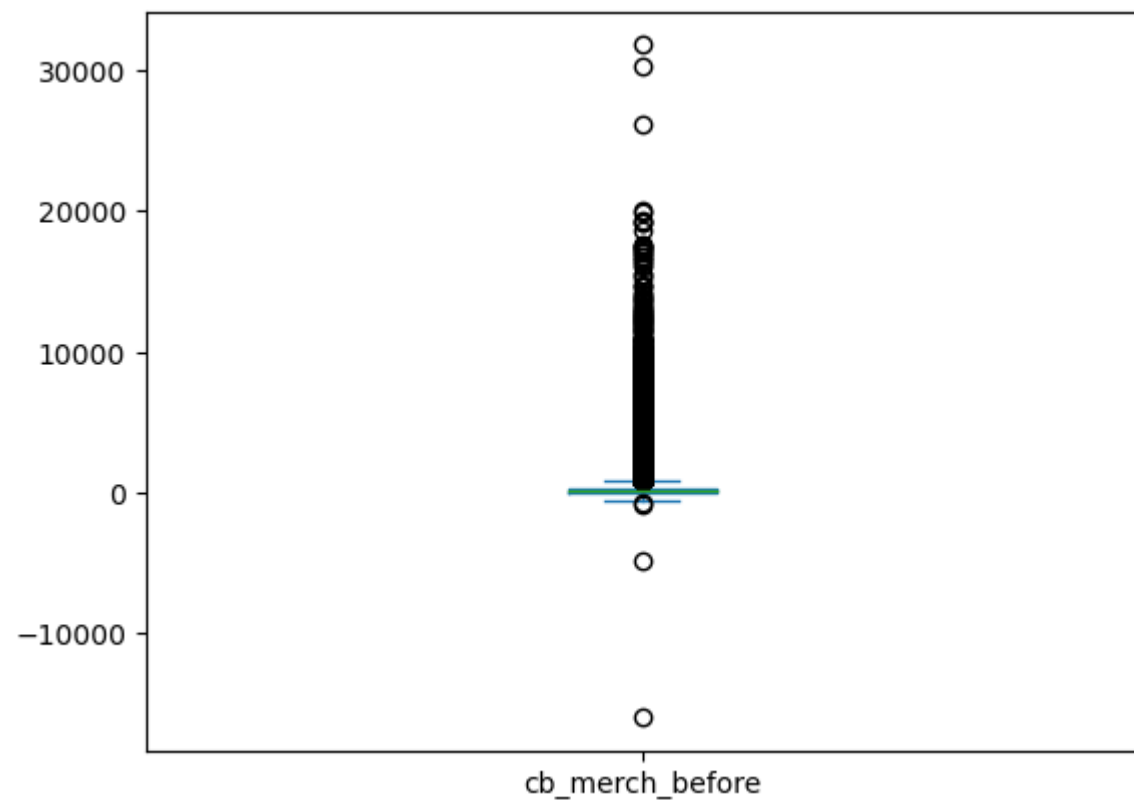
**Данные
приведены к
удобному
виду**

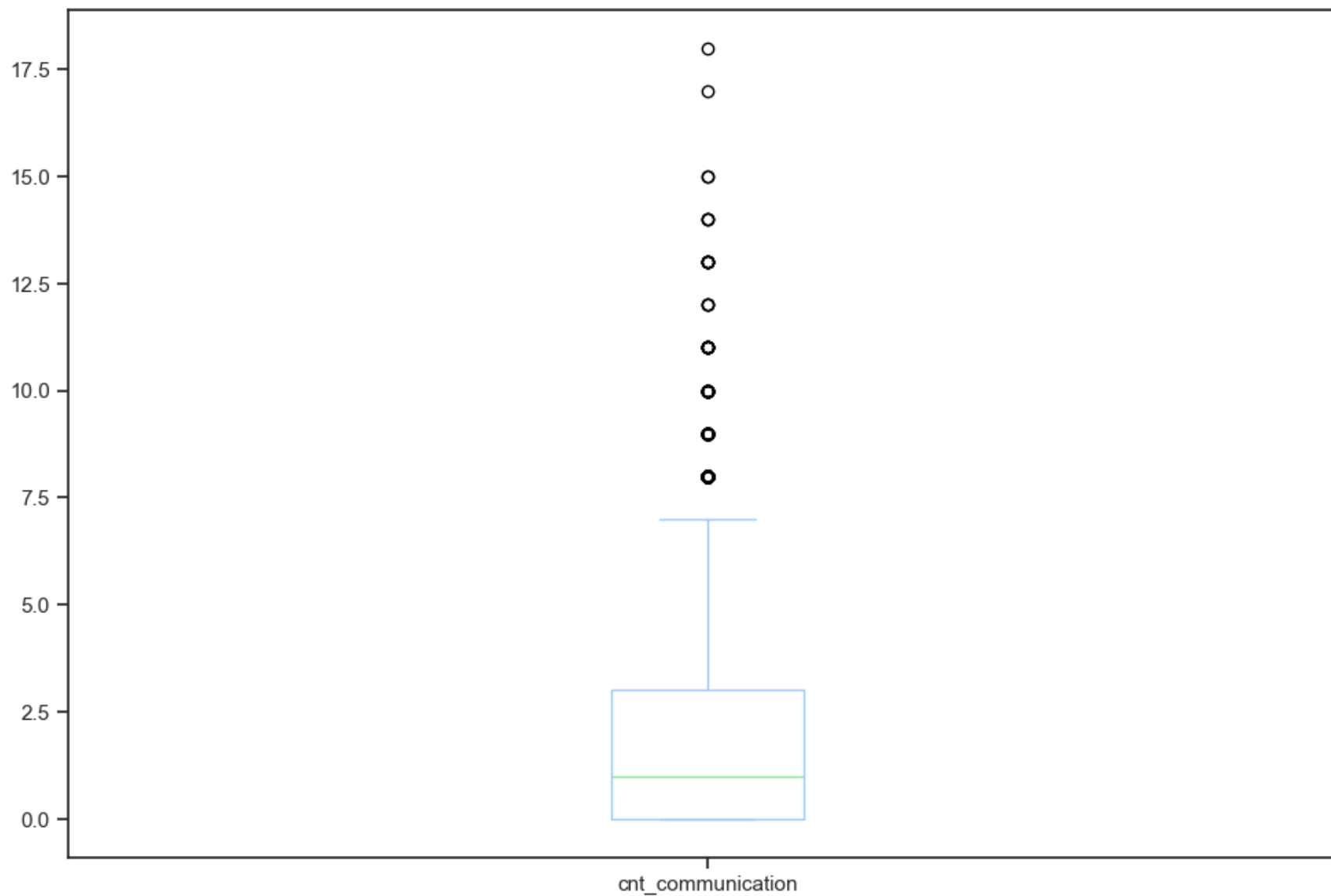
**Проверены
типы данных
и пропуски**

**Исследовано
наличие
дубликатов**







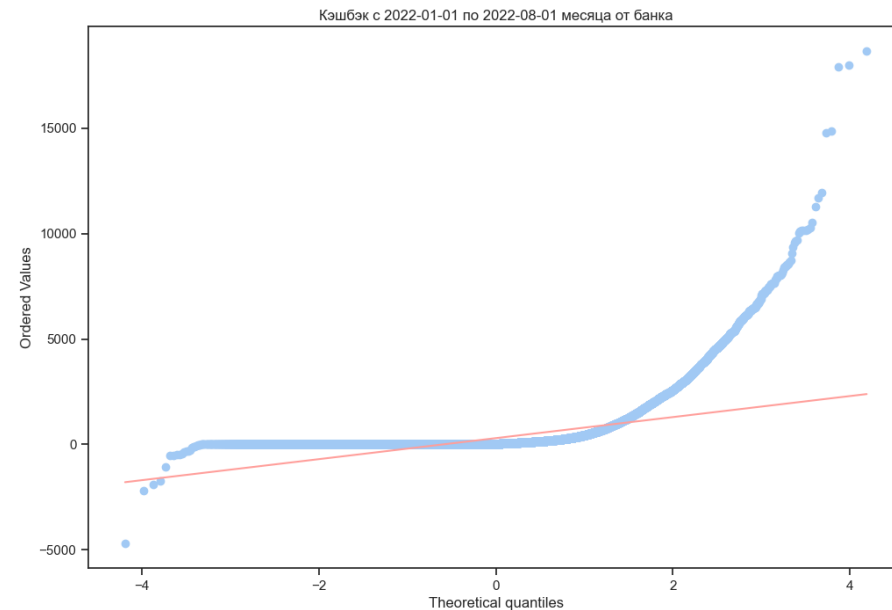
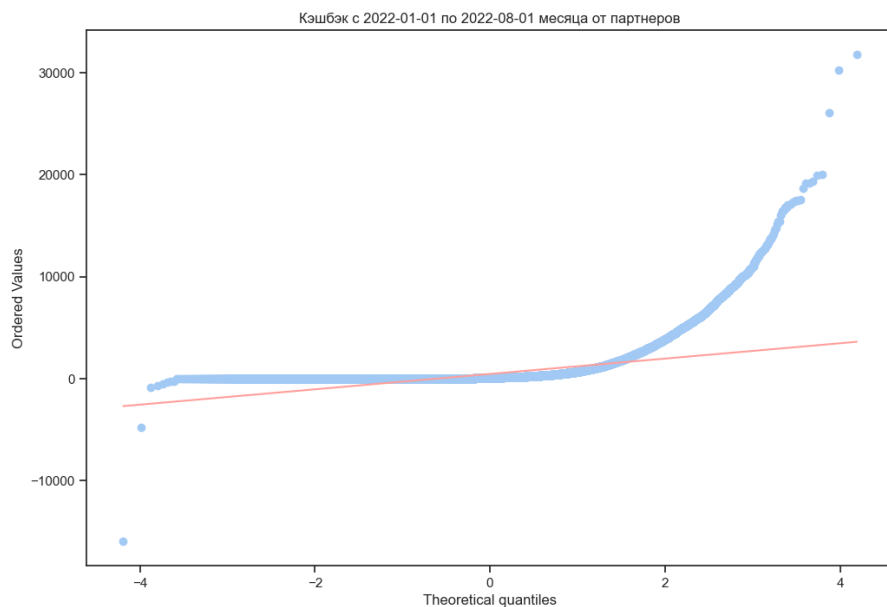
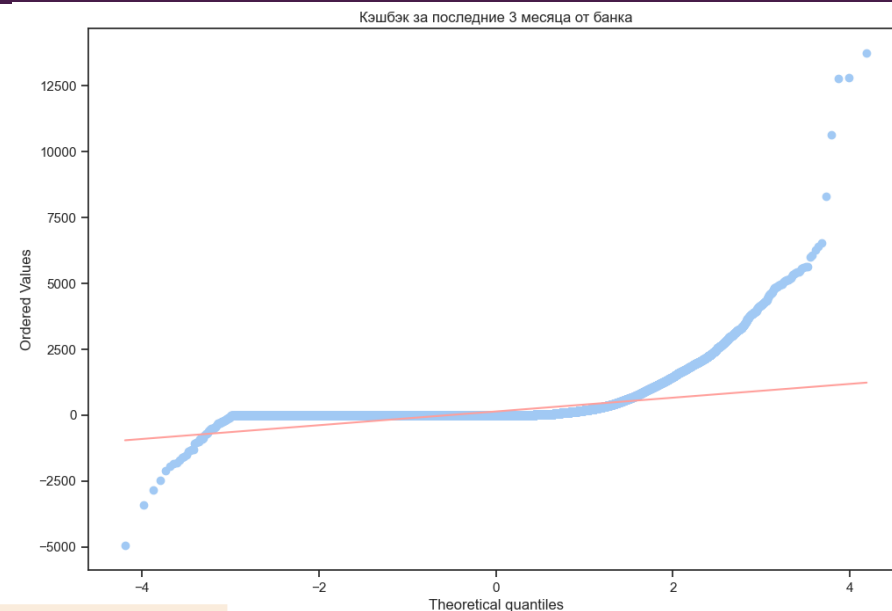
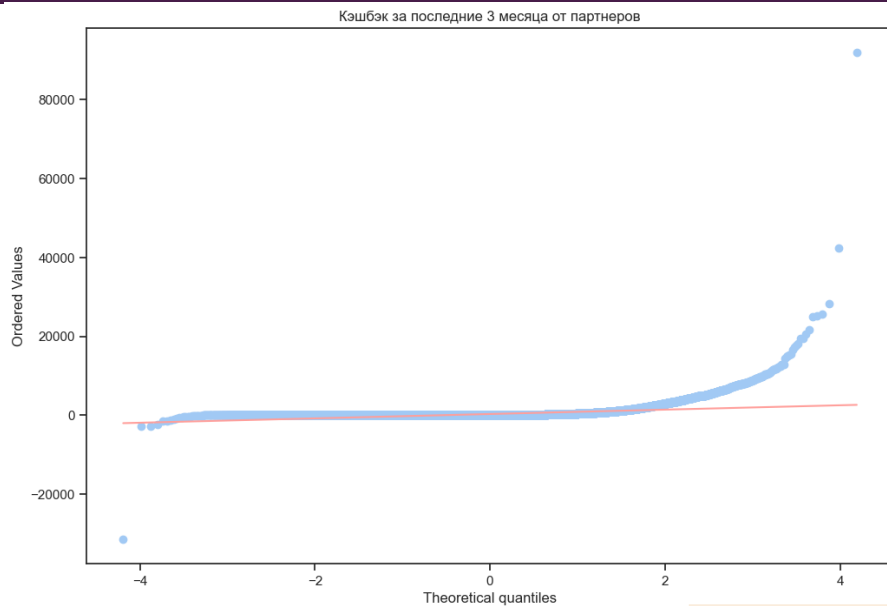




Предварительная обработка

Анализ данных¹

Выводы





КОЭФИЦИЕНТ КОРРЕЛЯЦИИ КЕНДАЛЛА

СОЗДАЛИ НОВЫЕ СТОЛБЦЫ ПО ВРЕМЕННОМУ ПРИЗНАКУ И ПО СУБЪЕКТУ ПОЛУЧЕНИЯ КЭШБЭКА И ПОСТРОИЛИ МАТРИЦУ КОРРЕЛЯЦИЙ

```
df_copy = df.copy()
#Создадим два столбца: первые по временному признаку
df_copy['cb_last_3_month'] = df['cb_merch_last_3_month'] + df['cb_bank_last_3_month'] #Кэшбэк за последние три месяца
df_copy['cb_before'] = df['cb_merch_before'] + df['cb_bank_before'] #Кэшбэк за первые месяцы
#Третий и четвертый по субъекту получения кэшбэка
df_copy['cb_merch'] = df['cb_merch_last_3_month'] + df['cb_merch_before'] #Кэшбэк от партнеров за год
df_copy['cb_bank'] = df['cb_bank_last_3_month'] + df['cb_bank_before'] #Кэшбэк от банка за год
```

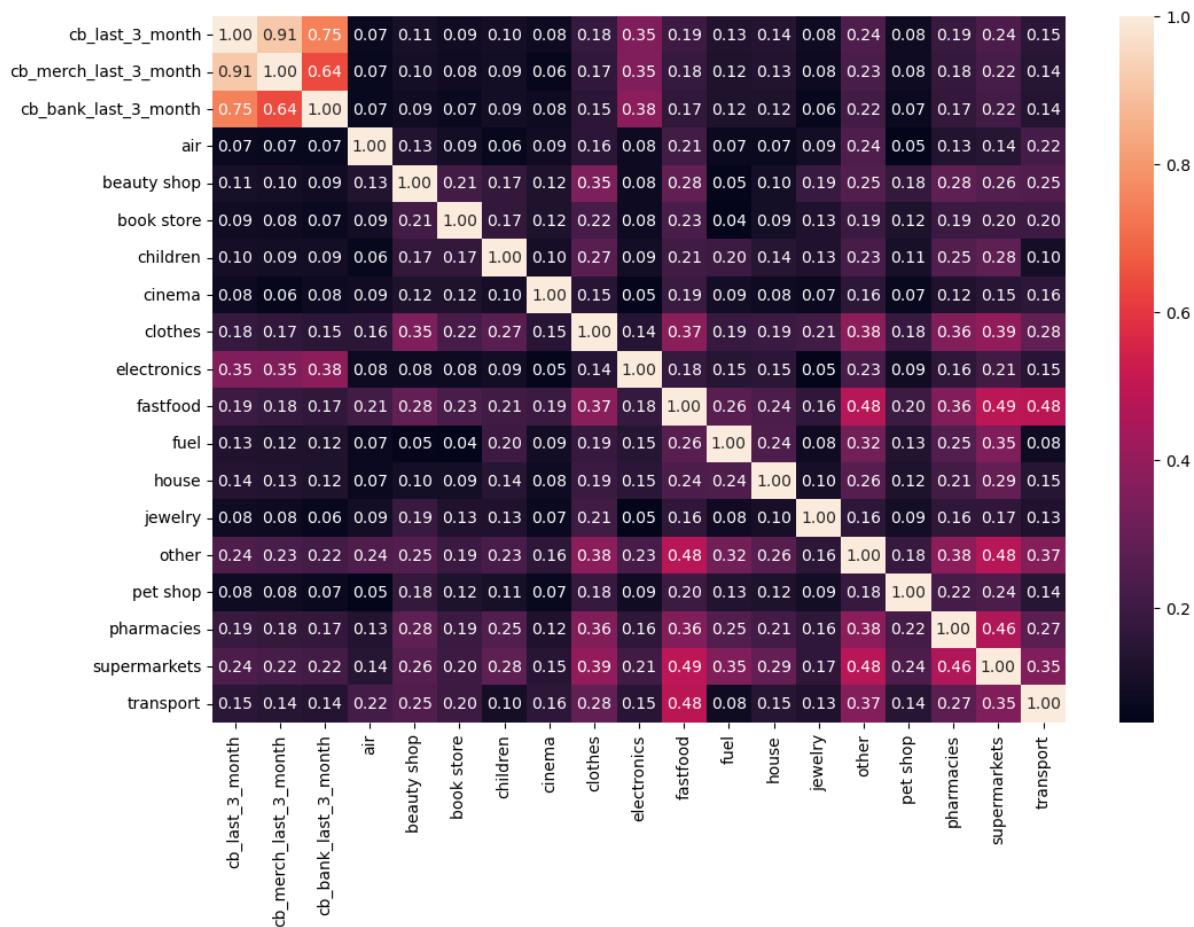




СОЗДАЛИ ДАТАСЕТ С ПОКУПКАМИ ЗА ПОСЛЕДНИЕ ТРИ МЕСЯЦА И ПОСТРОИЛИ МАТРИЦУ КОРРЕЛЯЦИЙ

#Создадим датасет с покупками за последние 3 месяца

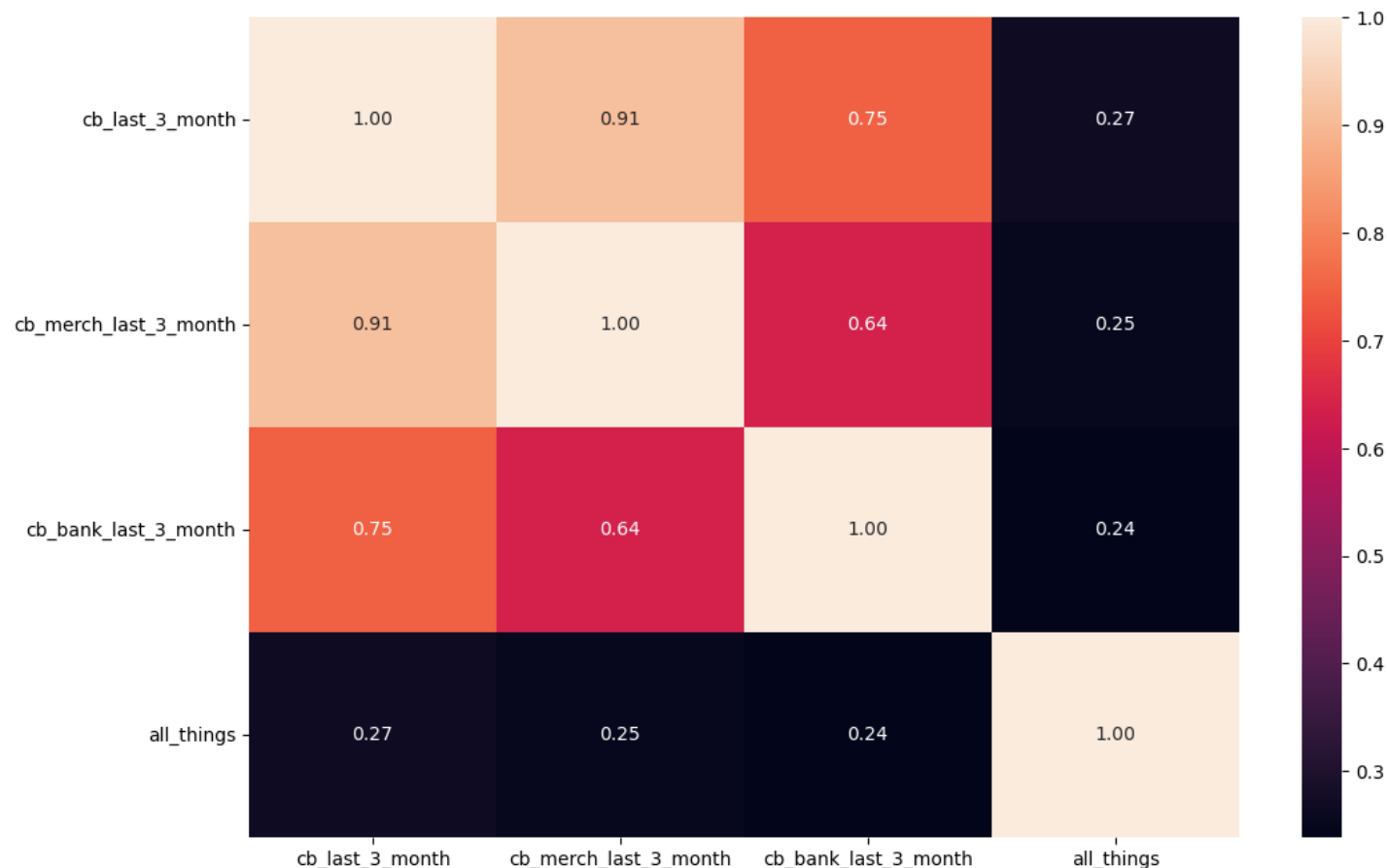
```
df_3 = df_all[['cb_last_3_month', 'cb_merch_last_3_month', 'cb_bank_last_3_month', 'air', 'beauty shop', 'book store',  
              'children', 'cinema', 'clothes', 'electronics', 'fastfood', 'fuel',  
              'house', 'jewelry', 'other', 'pet shop', 'pharmacies', 'supermarkets', 'transport']]  
sns.heatmap(df_3.corr(method='kendall'), annot=True, fmt=".2f", linecolor='white', cmap='BuPu')  
plt.show()
```





СОЗДАЛИ ДАТАСЕТ С ПОКУПКАМИ ЗА ПОСЛЕДНИЕ ТРИ МЕСЯЦА И ПОСТРОИЛИ МАТРИЦУ КОРРЕЛЯЦИЙ

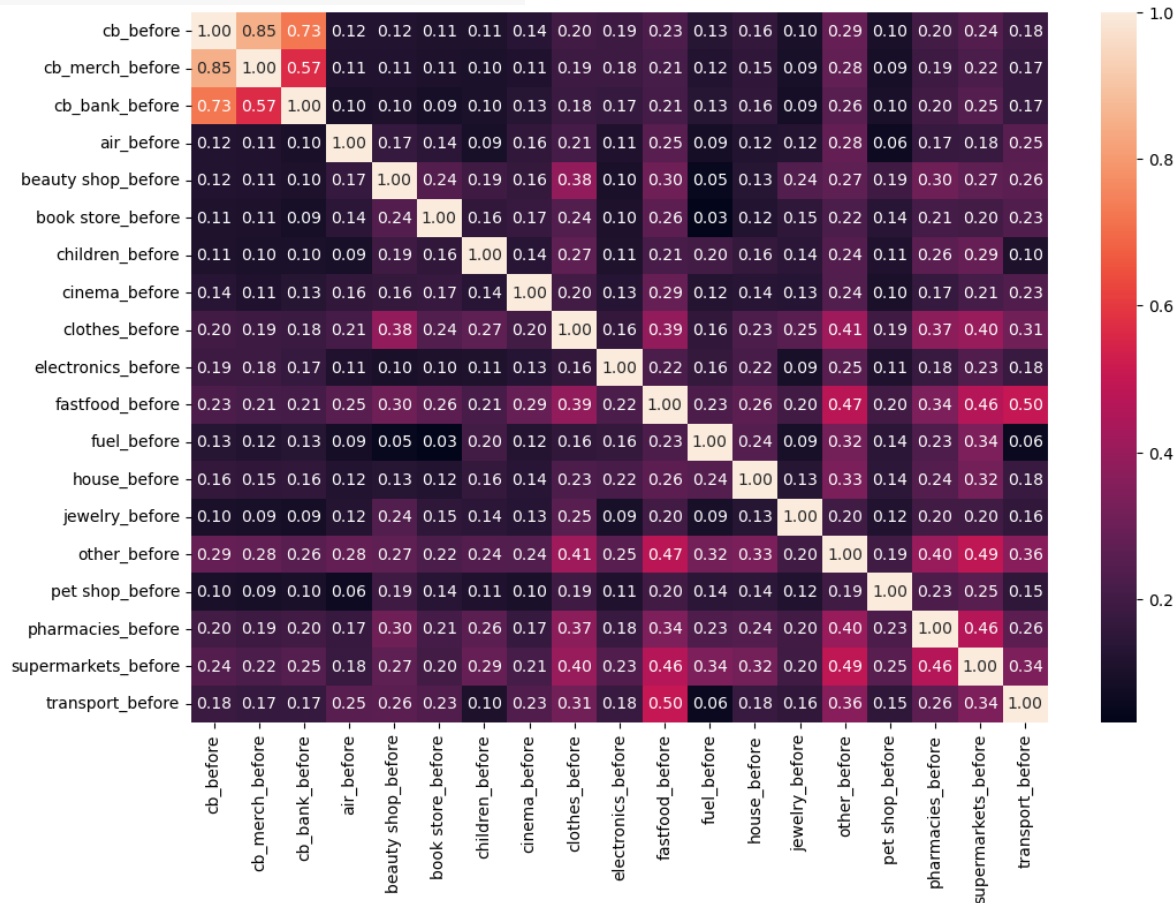
```
#Отдельно посмотрим сумму всех покупок за последние 3 месяца  
df_3_all = df_all[['cb_last_3_month', 'cb_merch_last_3_month', 'cb_bank_last_3_month']]  
df_3_all['all_things'] = df_3['air'] + df_3['beauty shop'] + df_3['book store'] + df_3['children']  
+ df_3['cinema'] + df_3['clothes'] + df_3['electronics'] + df_3['fastfood'] + df_3['fuel'] + df_3['house']  
+ df_3['jewelry'] + df_3['other'] + df_3['pet shop'] + df_3['pharmacies'] + df_3['supermarkets'] + df_3['transport']  
sns.heatmap(df_3_all.corr(method='kendall'), annot=True, fmt=".2f", linecolor='white', cmap='BuPu')  
plt.show()
```





СОЗДАЛИ ДАТАСЕТ С ПОКУПКАМИ ЗА ПЕРВЫЕ МЕСЯЦЫ И ПОСТРОИЛИ МАТРИЦУ КОРРЕЛЯЦИЙ

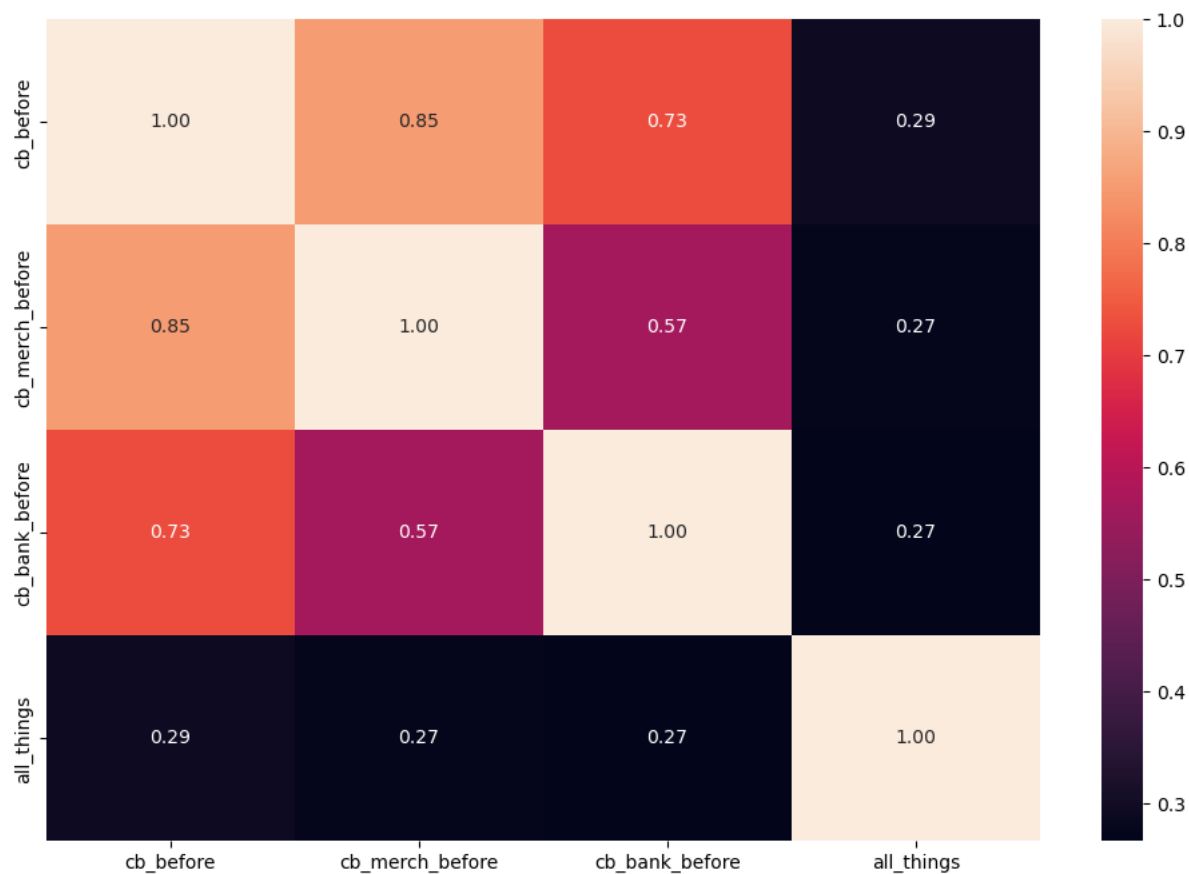
```
#Создадим датасет с покупками за первые месяцы
df_before = df_all[['cb_before', 'cb_merch_before', 'cb_bank_before', 'air_before', 'beauty_shop_before',
                    'book_store_before', 'children_before', 'cinema_before',
                    'clothes_before', 'electronics_before', 'fastfood_before',
                    'fuel_before', 'house_before', 'jewelry_before', 'other_before',
                    'pet_shop_before', 'pharmacies_before', 'supermarkets_before',
                    'transport_before']]
sns.heatmap(df_before.corr(method='kendall'), annot=True, fmt=".2f", linecolor='white', cmap='BuPu')
plt.show()
```





СОЗДАЛИ ДАТАСЕТ С ПОКУПКАМИ ЗА ПЕРВЫЕ МЕСЯЦЫ И ПОСТРОИЛИ МАТРИЦУ КОРРЕЛЯЦИЙ

```
#Отдельно посчитаем сумму всех покупок за первый период
df_before_all = df_all[['cb_before', 'cb_merch_before', 'cb_bank_before']]
df_before_all['all_things'] = df_before['air_before'] + df_before['beauty_shop_before'] + df_before['book_store_before']
+ df_before['children_before'] + df_before['cinema_before'] + df_before['clothes_before'] + df_before['electronics_before']
+ df_before['fastfood_before'] + df_before['fuel_before'] + df_before['house_before'] + df_before['jewelry_before']
+ df_before['other_before'] + df_before['pet_shop_before'] + df_before['pharmacies_before'] + df_before['supermarkets_before']
+ df_before['transport_before']
sns.heatmap(df_before_all.corr(method='kendall'), annot=True, fmt=".2f", linecolor='white', cmap='BuPu')
plt.show()
```

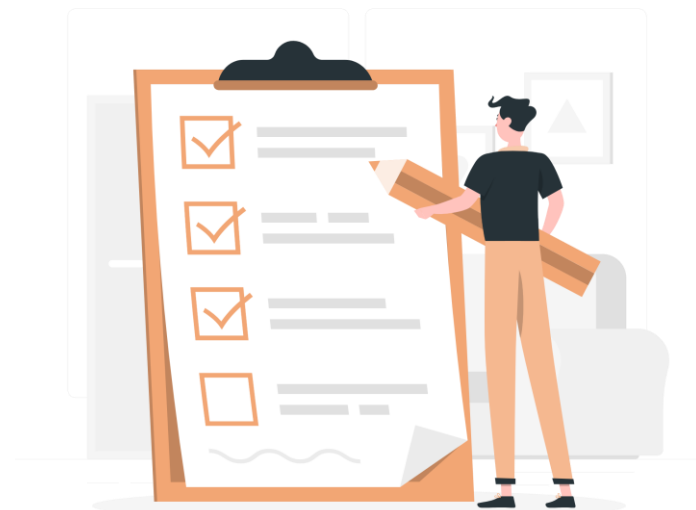




БАКЕТНЫЙ ТЕСТ

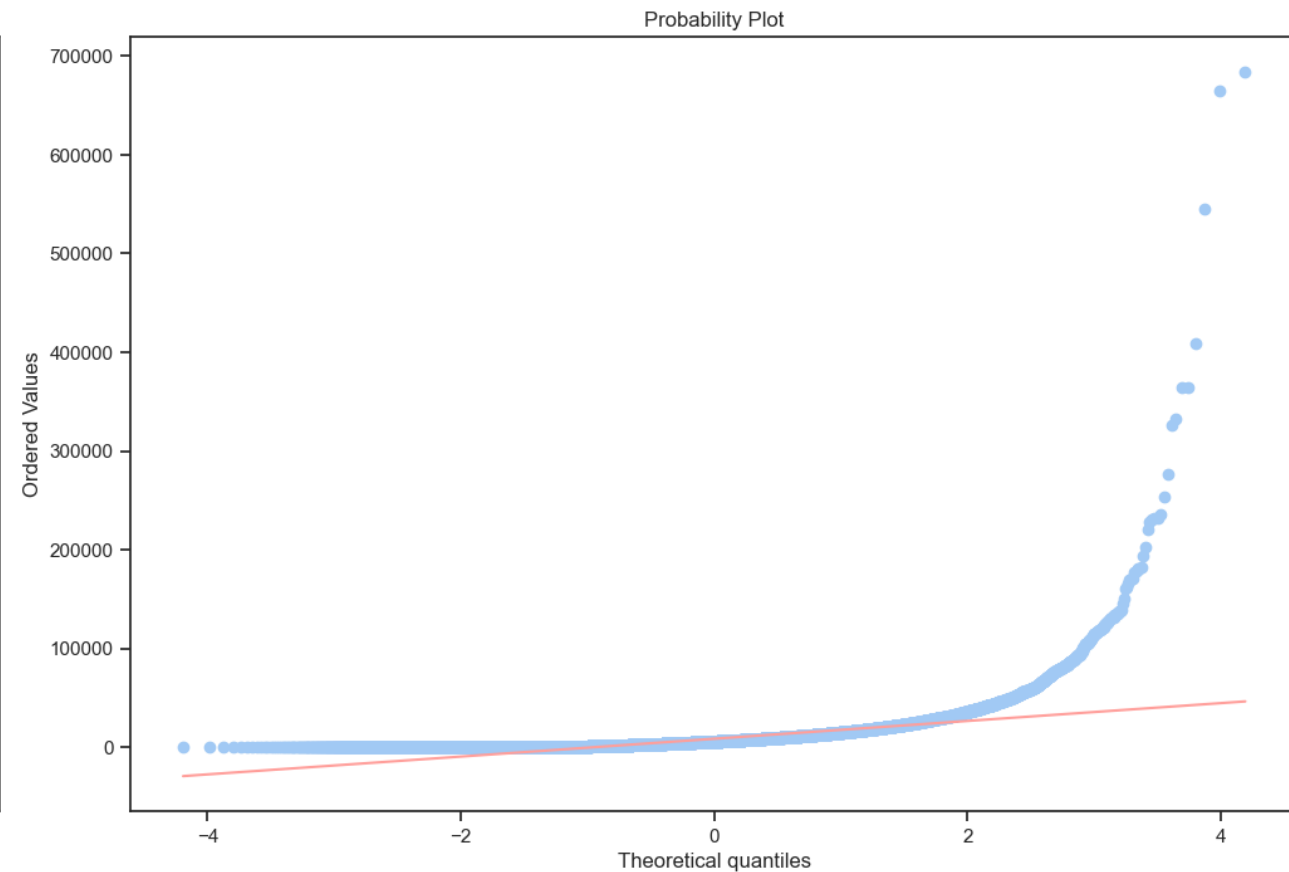
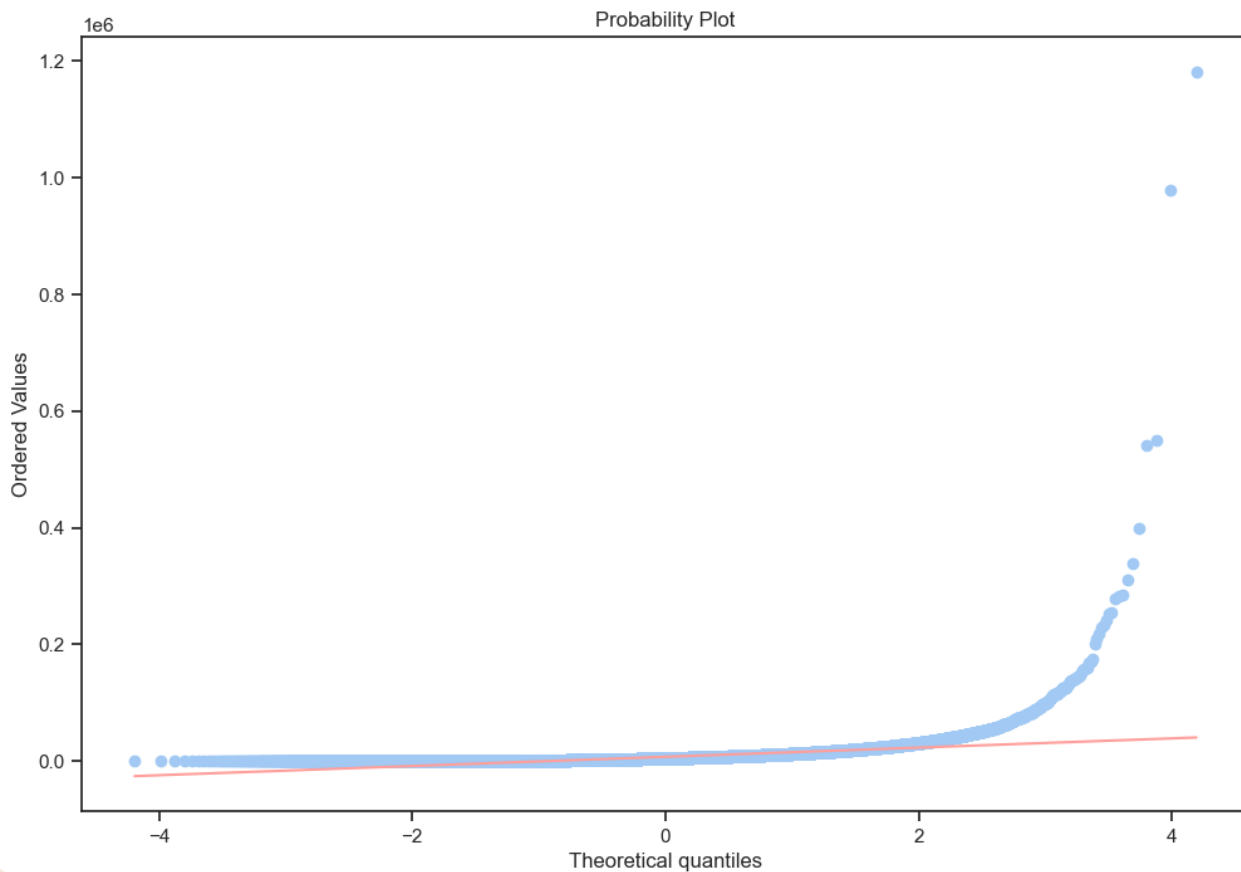
Разделим пользователей по бакетам:

- 50000 измерений
- 100 бакетов
- по 500 пользователей в каждом бакете



H0: различий между выборками нет, обновление не играет роли

H1: различия между выборками есть, обновление повлияло на изменение суммы заказа





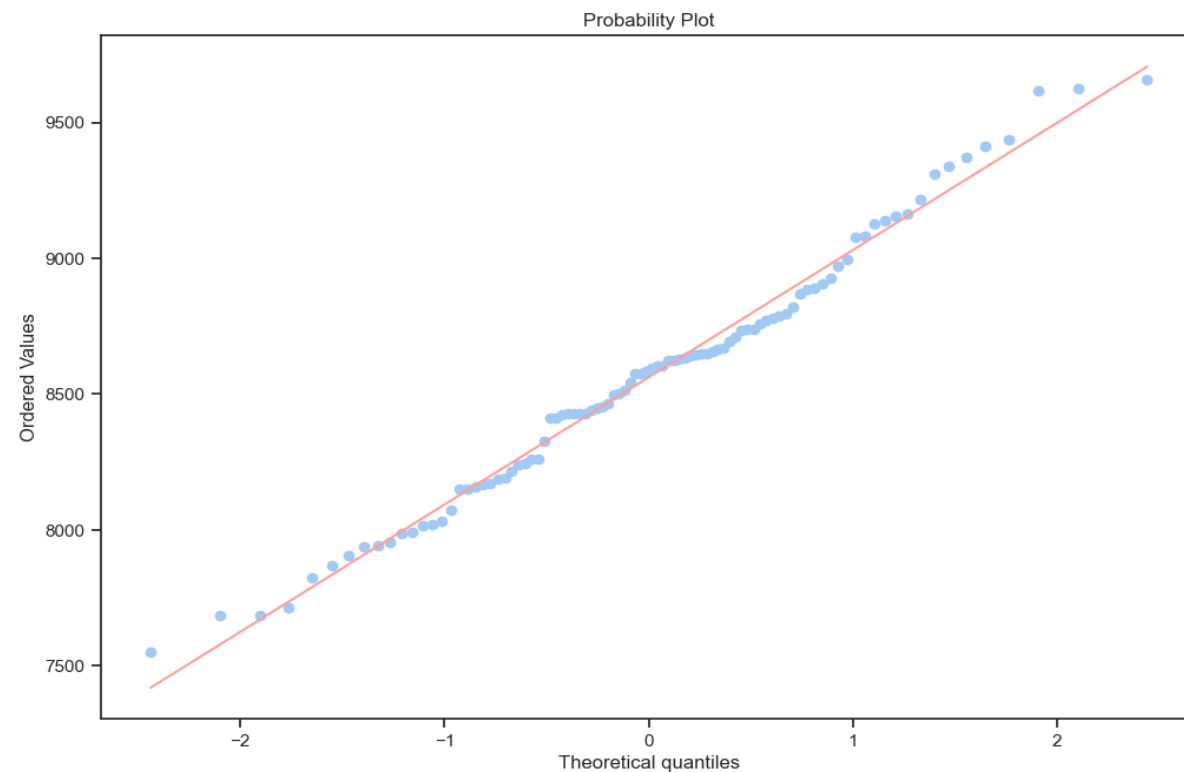
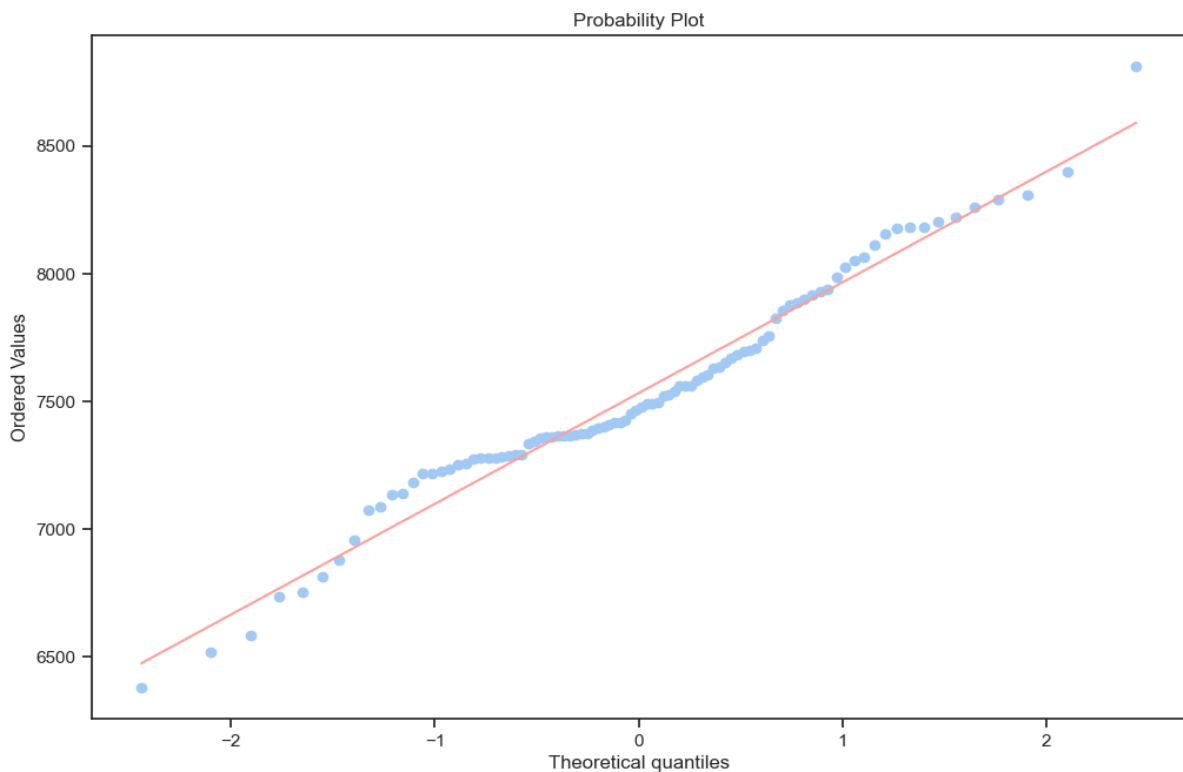
**НЕНОРМАЛЬНОЕ
РАСПРЕДЕЛЕНИЕ**

**ТЕСТ
МАННА-УИТНИ**

**ОТВЕРГАЕМ
ГИПОТЕЗУ О
НЕВАЖНОСТИ РОЛИ
ОБНОВЛЕНИЯ**



С ПОМОЩЬЮ УДАЛЕНИЯ ВЫБРОСОВ ПРИВЕЛИ РАСПРЕДЕЛЕНИЕ К НОРМАЛЬНОМУ



ТЕСТ ШАПИРО-УИЛКА

```
# Наблюдений меньше 5000 - можем применить тест Шапиро-Уилка
stat, p = st.shapiro(mean_metric['metric_before'])

alpha = 0.05
if p < alpha:
    print(p, 'Отвергаем нулевую гипотезу. Выборка НЕ имеет нормального распределения')
else:
    print(p, 'Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение')

0.08686017990112305 Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение

stat, p = st.shapiro(mean_metric['metric_after'])

alpha = 0.05
if p < alpha:
    print(p, 'Отвергаем нулевую гипотезу. Выборка НЕ имеет нормального распределения')
else:
    print(p, 'Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение')

0.4073050022125244 Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение
```



**НОРМАЛЬНОЕ
РАСПРЕДЕЛЕНИЕ**

t-ТЕСТ

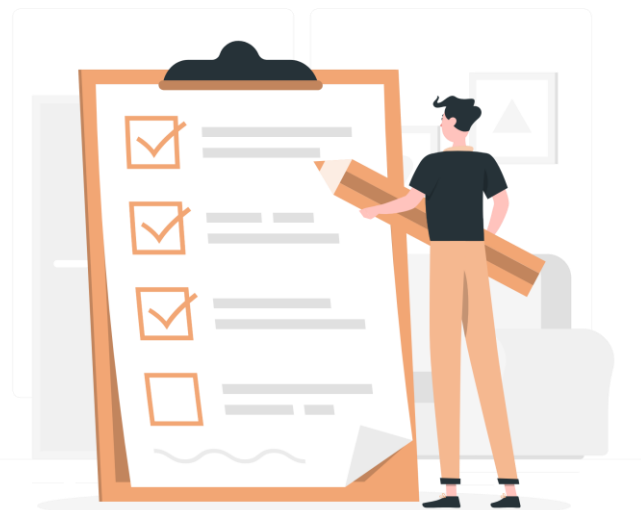
**ОТВЕРГАЕМ
ГИПОТЕЗУ О
НЕВАЖНОСТИ РОЛИ
ОБНОВЛЕНИЯ**



БАКЕТНЫЙ ТЕСТ

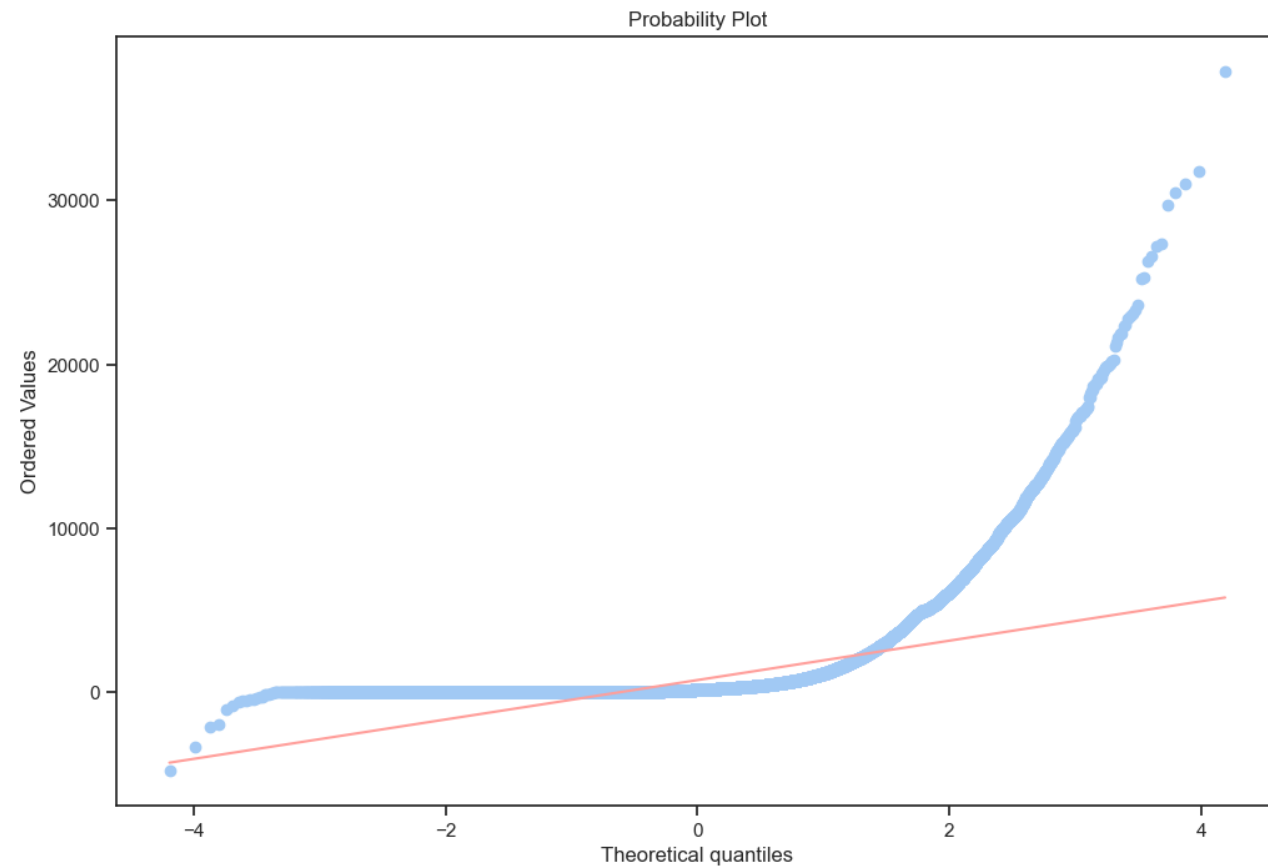
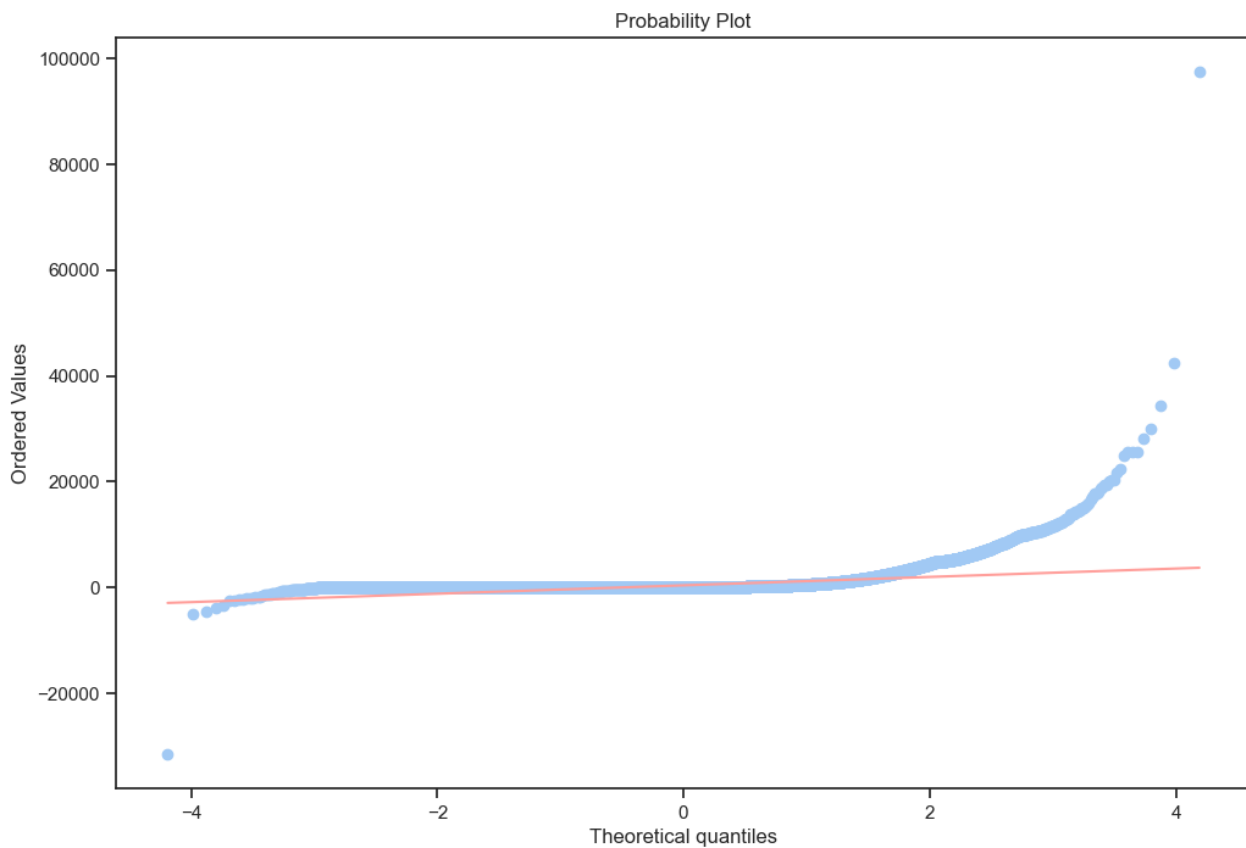
Разделим пользователей по бакетам:

- 50000 измерений
- 100 бакетов
- по 500 пользователей в каждом бакете



H0: обновление не повлияло на общий кэшбэк

H1: обновление повлияло на общий кэшбэк





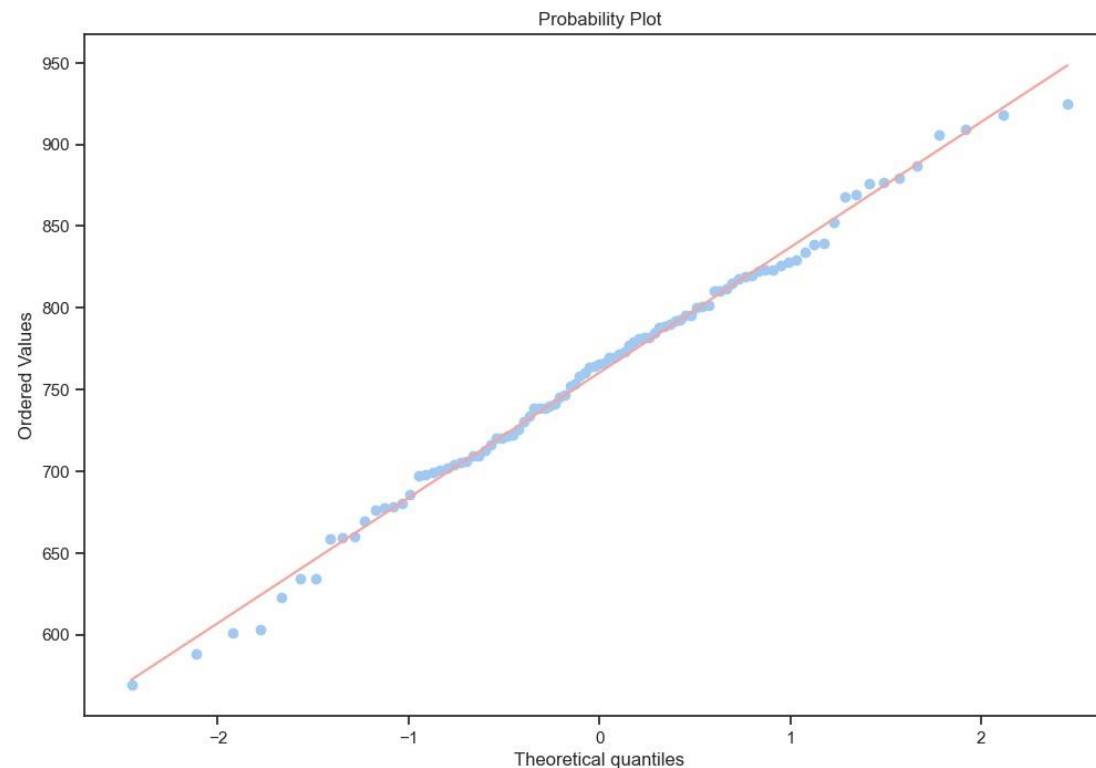
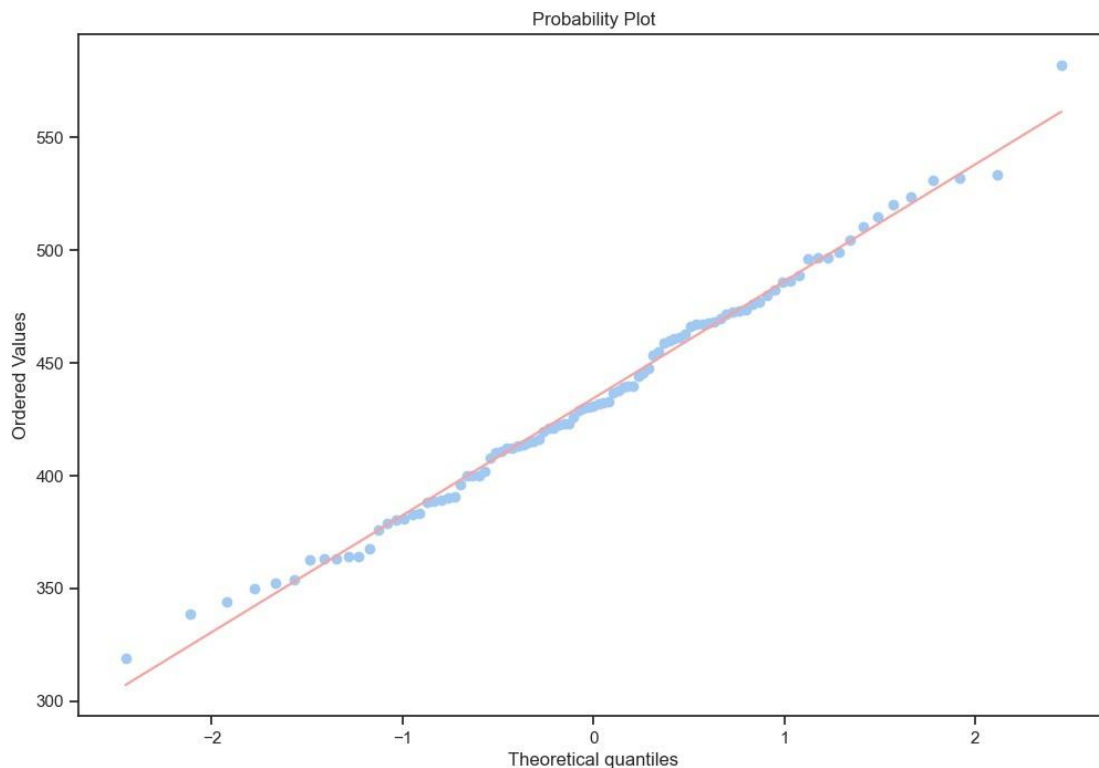
**НЕНОРМАЛЬНОЕ
РАСПРЕДЕЛЕНИЕ**

**ТЕСТ
МАННА-УИТНИ**

**ОТВЕРГАЕМ
ГИПОТЕЗУ О
НЕВАЖНОСТИ РОЛИ
ОБНОВЛЕНИЯ**



С ПОМОЩЬЮ УДАЛЕНИЯ ВЫБРОСОВ ПРИВЕЛИ РАСПРЕДЕЛЕНИЕ К НОРМАЛЬНОМУ



ТЕСТ ШАПИРО-УИЛКА

```
# Наблюдений меньше 5000 - можем применить тест Шапиро-Уилка
stat, p = st.shapiro(mean_metric['cb_last_3_month'])

alpha = 0.05
if p < alpha:
    print(p, 'Отвергаем нулевую гипотезу. Выборка НЕ имеет нормального распределения')
else:
    print(p, 'Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение')

0.8287481069564819 Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение

stat, p = st.shapiro(mean_metric['cb_before'])

alpha = 0.05
if p < alpha:
    print(p, 'Отвергаем нулевую гипотезу. Выборка НЕ имеет нормального распределения')
else:
    print(p, 'Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение')

0.824847400188446 Не получилось отвергнуть нулевую гипотезу. Выборка имеет нормальное распределение
```



**НОРМАЛЬНОЕ
РАСПРЕДЕЛЕНИЕ**

t-ТЕСТ

**ОТВЕРГАЕМ
ГИПОТЕЗУ О
НЕВАЖНОСТИ РОЛИ
ОБНОВЛЕНИЯ**



- 1 СВЯЗЬ МЕЖДУ ВСЕМИ ПОКАЗАТЕЛЯМИ И КОЛИЧЕСТВОМ КОММУНИКАЦИЙ СЛАБАЯ И НЕЛИНЕЙНАЯ
- 2 СВЯЗЬ МЕЖДУ ОБОРОТОМ КЛИЕНТОВ И ВЫПЛАЧЕННЫМ КЭШБЭКОМ СЛАБАЯ И НЕЛИНЕЙНАЯ
- 3 ОБНОВЛЕНИЕ ПОВЛИЯЛО НА СУММУ ПОКУПОК
- 4 ОБНОВЛЕНИЕ ПОВЛИЯЛО НА СУММУ КЭШБЭКА

