Анализ изменения выручки интернет-магазина по результатам A/B-теста

Описание проекта

Проанализировать потери для увеличенной выручки интернет-магазина, запустив A/B-тест и проанализировав результаты.

Оглавление

- Часть 1. Приоритизация гипотез
- Часть 2. Анализ A/B-теста
  - 1. Графики кумулятивных агрегатов по группам
  - 2. Графики кумулятивных метрик по группам
  - 3. Графики кумулятивной выручки по группам
  - 4. Графики кумулятивных конверсий по группам
  - 5. Графики относительных изменений кумулятивных метрик группы B к группе A
  - 6. Точечный график стоимости заказов по пользователям
  - 7. Точечный график распределения заказов по пользователям
  - 8. Расчет 95-го и 99-го перцентилей стоимости заказов. Выбор аномальных заказов пользователей
  - 9. Точечный график количества заказов
  - 10. Расчет 95-го и 99-го перцентилей количества заказов. Выбор аномальных заказов
  - 11. Статистическая значимость различий в конверсии между группами по сырым данным
  - 12. Статистическая значимость различий в среднем чеке между группами по сырым данным
  - 13. Статистическая значимость различий в конверсии между группами по очищенным данным
  - 14. Статистическая значимость различий в среднем чеке между группами по очищенным данным
  - 15. Принятие решения по результатам теста и описание результатов теста
- Общий вывод


```
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt, pd.set
import numpy as np, np.set
import seaborn as sns
import scipy.stats as st
from scipy import stats as st
from scipy.stats import ttest_ind
```

Часть 1. Приоритизация гипотез

Импортируем файлы:

```
hypothesis = pd.read_csv('/datasets/hypothesis.csv')
```

```
hypothesis.head()
```

	Hypothesis	Reach	Impact	Confidence	Efforts
0	Добавить два новых способа оплаты: электронные деньги и банковские карты	3	10	8	6
1	Запустить собственную службу доставки заказов	2	5	4	10
2	Добавить блоки рекомендаций товаров на сайт интернет-магазина, чтобы увеличить количество просмотров разных товаров	3	10	8	6
3	Изменить цвет фона главной страницы, чтобы привлечь больше клиентов	1	3	1	1
4	Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок	8	3	7	2

Применим фреймворк ICE для приоритизации гипотез. Отсортируем по убыванию приоритета с округлением до двух знаков.

```
ice = hypothesis[['Hypothesis', 'Impact', 'Confidence', 'Efforts']].sort_values('Impact', ascending=False)
ice['ICE'] = round(ice['Impact'] * ice['Confidence'] / ice['Efforts'], 2)
ice.sort_values('ICE', ascending=False).style.hide_index()
```

Hypothesis	Impact	Confidence	Efforts	ICE
Запустить акцию, дающую скидку на товар в день рождения	9	4	1	16.00
Добавить два новых способа оплаты: электронные деньги и банковские карты	10	8	6	13.33
Добавить блоки рекомендаций товаров на сайт интернет-магазина, чтобы увеличить количество просмотров разных товаров	10	8	6	13.33
Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок	3	7	2	11.20
Запустить собственную службу доставки заказов	5	4	10	2.00
Изменить цвет фона главной страницы, чтобы привлечь больше клиентов	3	1	1	3.00

Применим фреймворк RICE для приоритизации гипотез. Отсортируем по убыванию приоритета с округлением до двух знаков.

```
rice = hypothesis[['Hypothesis', 'Reach', 'Impact', 'Confidence', 'Efforts']].sort_values('Reach', ascending=False)
rice['RICE'] = round(rice['Reach'] * rice['Impact'] * rice['Confidence'] / rice['Efforts'], 2)
rice.sort_values('RICE', ascending=False).style.hide_index()
```

Hypothesis	Reach	Impact	Confidence	Efforts	RICE
Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок	8	3	7	2	56.00
Добавить блоки рекомендаций товаров на сайт интернет-магазина, чтобы увеличить количество просмотров разных товаров	3	10	8	6	40.00
Добавить два новых способа оплаты: электронные деньги и банковские карты	3	10	8	6	40.00
Запустить акцию, дающую скидку на товар в день рождения	1	9	4	1	16.00
Запустить собственную службу доставки заказов	2	5	4	10	4.00
Изменить цвет фона главной страницы, чтобы привлечь больше клиентов	1	3	1	1	3.00

Применим фреймворк ICE для приоритизации гипотез. Отсортируем по убыванию приоритета гипотезы.

```
px.bar(ice, x='Hypothesis', y='ICE', text='ICE', title='Приоритизация гипотез по фреймворку ICE', width=1000, height=500)
```

Применим фреймворк RICE для приоритизации гипотез. Отсортируем по убыванию приоритета гипотезы.

```
px.bar(rice, x='Hypothesis', y='RICE', text='RICE', title='Приоритизация гипотез по фреймворку RICE', width=1000, height=500)
```

При применении ICE лидируют гипотезы 8 и 0, а фиолетовая гипотеза на первом месте после применения фреймворка RICE. Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок*, которая что-то про доступ/закрытый доступ к каким-то ресурсам, на 2-ом месте. Гипотезы 7 и 2 — на третьем месте по приоритету. Гипотезы 6 и 1 — на последнем месте. Гипотезы 4 и 3 — на предпоследнем месте. Гипотезы 5 — на 5-ом месте. Гипотезы 4 и 3 — на предпоследнем месте. Гипотезы 5 — на 5-ом месте. Для дальнейшей работы оставлю все гипотезы.

К оглавлению

Часть 2. Анализ A/B-теста

Импортируем данные и посмотрим общую информацию.

```
orders = pd.read_csv('/datasets/orders.csv')
```

```
orders.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 5 columns):
 transaction_id  1197 non-null  int64
 visitor_id      1197 non-null  int64
 date            1197 non-null  object
 revenue         1197 non-null  int64
 group           1197 non-null  object
dtypes: int64(4), object(1)
memory usage: 46.8+ KB
```

```
orders.sample(5)
```

Посмотрим полные копии строк таблицы orders, чтобы не пропустить дубликаты.

```
orders[orders.duplicated(keep=False)].sort_values(['transaction_id', 'visitor_id', 'date', 'revenue', 'group'])
```

	transaction_id	visitor_id	date	revenue	group
393	2961555356	4069496402	2019-08-15	10601	B
394	2961555356	4069496402	2019-08-15	10601	B
1016	3941795274	2356004889	2019-08-15	2068	B
1017	3941795274	2356004889	2019-08-15	2068	B

В выборке есть дубликаты.

Посмотрим уникальные переменные в каждой выборке по группам.

```
orders.groupby('group').nunique()
```

group	transaction_id	visitor_id	date	revenue
A	557	358	31	592
B	640	357	31	707

```
visitors = pd.read_csv('/datasets/visitors.csv')
```

```
visitors.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 3 columns):
 date      62 non-null  object
 group     62 non-null  object
 visitors  62 non-null  int64
dtypes: int64(1), object(2)
memory usage: 1.5+ KB
```

```
visitors.sample(5)
```

Пропущенных значений в таблицах нет. Изменим тип данных на datetime.

```
orders['date'] = pd.to_datetime(orders['date'])
visitors['date'] = pd.to_datetime(visitors['date'])
```

Посмотрим полные копии таблицы visitors, чтобы не пропустить дубликаты.

```
visitors[visitors.duplicated(keep=False)].sort_values(['date', 'group', 'visitors'])
```

	date	group	visitors
55	2019-08-27	A	720
56	2019-08-27	A	720
57	2019-08-27	B	650
58	2019-08-27	B	650

В visitors дубликаты.

Посмотрим полные копии строк таблицы orders, чтобы не пропустить дубликаты.

```
orders[orders.duplicated(keep=False)].sort_values(['transaction_id', 'visitor_id', 'date', 'revenue', 'group'])
```

	transaction_id	visitor_id	date	revenue	group
393	2961555356	4069496402	2019-08-15	10601	B
394	2961555356	4069496402	2019-08-15	10601	B
1016	3941795274	2356004889	2019-08-15	2068	B
1017	3941795274	2356004889	2019-08-15	2068	B

Сгруппируем по группам кумулятивные данные о заказах, посчитаем и выручку нарастим. Для удобства назовём столбцы orders.

```
cumulative_orders = orders.groupby(['date', 'group']).agg({'transaction_id': 'nunique', 'visitor_id': 'nunique', 'revenue': 'sum'}).reset_index()
cumulative_orders.columns = ['date', 'group', 'orders', 'buyers', 'revenue']
```

```
cumulative_orders.sample(5)
```

	date	group	orders	buyers	revenue
29	2019-08-15	B	290	267	148579
11	2019-08-06	A	175	165	148687
25	2019-08-14	B	243	225	133494
7	2019-08-04	A	116	109	148579
27	2019-08-15	A	384	343	435241

Сгруппируем по группам кумулятивные данные о посетителях, посчитаем и нарастим метрики visitors, назовём и отсортируем столбцы.

```
cumulative_visitors = visitors.groupby(['date', 'group']).agg({'visitors': 'sum'}).reset_index()
cumulative_visitors.columns = ['date', 'group', 'visitors']
```

```
cumulative_visitors.sample(5)
```

	date	group	visitors
29	2019-08-15	B	17708
11	2019-08-06	A	17708
25	2019-08-14	B	17381
7	2019-08-04	A	11373
27	2019-08-15	A	18736

1. Графики кумулятивных агрегатов по группам

Соберём кумулятивные агрегированные данные о заказах и посетителях магазина. Сгруппируем таблицу cumulative_data по колонкам date и group, посчитаем максимальное значение кумулятивных заказов, посетителей и суммарную выручку. Сделаем индекс датафрейма cumulative_data.

```
cumulative = cumulative_orders.merge(cumulative_visitors, on=['date', 'group'])
cumulative.sample(5)
```

	date	group	orders	buyers	revenue	visitors
29	2019-08-15	B	290	267	148579	17708
11	2019-08-06	A	175	165	148687	17708
25	2019-08-14	B	243	225	133494	17381
7	2019-08-04	A	116	109	148579	11373
27	2019-08-15	A	384	343	435241	18736

```
cumulative.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 62 entries, 0 to 61
Data columns (total 6 columns):
 date      62 non-null  datetime64[ns]
 group     62 non-null  object
 orders    62 non-null  int64
 buyers    62 non-null  int64
 revenue   62 non-null  int64
 visitors  62 non-null  int64
dtypes: datetime64[ns], int64(4), object(1)
memory usage: 3.4+ KB
```

Объединим обе таблицы в одной, сохраним названиями столбцов. Посмотрим случайные строки.

```
cumulative = cumulative.merge(cumulative_visitors, on=['date', 'group'])
cumulative.columns = ['date', 'group', 'orders', 'buyers', 'revenue', 'visitors', 'cum_visitors']
```

```
cumulative.sample(5)
```

	date	group	orders	buyers	revenue	visitors	cum_visitors
29	2019-08-15	B	290	267	148579	18736	17708
11	2019-08-06	A	175	165	148687	18916	17708
25	2019-08-14	B	243	225	133494	18772	17381
7	2019-08-04	A	116	109	148579	11373	11373
27	2019-08-15	A	384	343	435241	18916	18736

Объединим обе таблицы в одной, сохраним названиями столбцов. Посмотрим случайные строки.

```
cumulative = cumulative.merge(cumulative_orders, on=['date', 'group'])
cumulative.columns = ['date', 'group', 'orders', 'buyers', 'revenue', 'visitors', 'cum_visitors', 'cum_orders', 'cum_buyers', 'cum_revenue']
```

```
cumulative.sample(5)
```

	date	group	orders	buyers	revenue	visitors	cum_visitors	cum_orders	cum_buyers	cum_revenue
29	2019-08-15	B	290	267	148579	18736	17708	1548	1404	773828
11	2019-08-06	A	175	165	148687	18916	17708	1068	1020	1020547
25	2019-08-14	B	243	225	133494	18772	17381	1328	1196	711884
7	2019-08-04	A	116	109	148579	11373	11373	716	687	592708
27	2019-08-15	A	384	343	435241	18916	18736	1092	1047	1064962

Построим графики кумулятивных агрегатов по группам. Для каждого графика зададим отдельный размер.

```
# кумулятивные кумулятивные выручка по группам и кумулятивный средний чек по дням и группам
cumulative_revenue = cumulative[['date', 'group', 'cum_revenue']]
cumulative_revenue.sample(5)
```

	date	group	cum_revenue
29	2019-08-15	B	773828
11	2019-08-06	A	1020547
25	2019-08-14	B	711884
7	2019-08-04	A	592708
27	2019-08-15	A	1064962

```
# строим график выручки группы A
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_revenue[cumulative_revenue['group']=='A'], x='date', y='cum_revenue')
```

```
# строим график выручки группы B
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_revenue[cumulative_revenue['group']=='B'], x='date', y='cum_revenue')
```

```
# задаем названия
plt.title('Графики кумулятивных выручек по группам')
plt.xlabel('Дата')
plt.ylabel('Выручка')
plt.legend(['Группа A', 'Группа B'])
```

```
# отключаем экспоненциальное представление
plt.tickformat(1, axis='y')
```

```
# сохраняем
plt.savefig('/datasets/cumulative_revenue.jpg')
```

```
# отображение
plt.show()
```

```
/usr/local/lib/python3.9/site-packages/matplotlib/ticker.py:2448: UserWarning: tickformat not supported for the current backend.
  warnings.warn("tickformat not supported for the "
```

Почему-то не сработало сохранение графика. Сохраню его вручную, выгружу на компьютер и загружу в папку datasets. Сохраню в папку datasets. Что-то опять не сохранилось. Попробую. Что-то опять не сохранилось.

К оглавлению

2. Графики кумулятивных метрик по группам

Построим графики кумулятивных метрик по группам — среднего кумулятивного выручки по кумулятивному числу заказов.

```
# кумулятивные кумулятивные выручка по группам и кумулятивный средний чек по дням и группам
cumulative_revenue = cumulative[['date', 'group', 'cum_revenue']]
cumulative_revenue.sample(5)
```

	date	group	cum_revenue
29	2019-08-15	B	773828
11	2019-08-06	A	1020547
25	2019-08-14	B	711884
7	2019-08-04	A	592708
27	2019-08-15	A	1064962

```
# строим график выручки группы A
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_revenue[cumulative_revenue['group']=='A'], x='date', y='cum_revenue')
```

```
# строим график выручки группы B
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_revenue[cumulative_revenue['group']=='B'], x='date', y='cum_revenue')
```

```
# задаем названия
plt.title('Графики кумулятивных выручек по группам')
plt.xlabel('Дата')
plt.ylabel('Выручка')
plt.legend(['Группа A', 'Группа B'])
```

```
# отключаем экспоненциальное представление
plt.tickformat(1, axis='y')
```

```
# сохраняем
plt.savefig('/datasets/cumulative_revenue.jpg')
```

```
# отображение
plt.show()
```

```
/usr/local/lib/python3.9/site-packages/matplotlib/ticker.py:2448: UserWarning: tickformat not supported for the current backend.
  warnings.warn("tickformat not supported for the "
```

Сохраню не можем сделать. Сохраню вручную. Что-то опять не сохранилось. Попробую. Что-то опять не сохранилось. Сохраню на компьютер. Загружу в папку datasets. Что-то опять не сохранилось. Попробую. Что-то опять не сохранилось. Что-то опять не сохранилось.

К оглавлению

3. Графики кумулятивной выручки по группам

Построим графики кумулятивной выручки по группам — среднего кумулятивного выручки по кумулятивному числу заказов.

```
# кумулятивные кумулятивные выручка по группам и кумулятивный средний чек по дням и группам
cumulative_revenue = cumulative[['date', 'group', 'cum_revenue']]
cumulative_revenue.sample(5)
```

	date	group	cum_revenue
29	2019-08-15	B	773828
11	2019-08-06	A	1020547
25	2019-08-14	B	711884
7	2019-08-04	A	592708
27	2019-08-15	A	1064962

```
# строим график выручки группы A
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_revenue[cumulative_revenue['group']=='A'], x='date', y='cum_revenue')
```

```
# строим график выручки группы B
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_revenue[cumulative_revenue['group']=='B'], x='date', y='cum_revenue')
```

```
# задаем названия
plt.title('Графики кумулятивных выручек по группам')
plt.xlabel('Дата')
plt.ylabel('Выручка')
plt.legend(['Группа A', 'Группа B'])
```

```
# отключаем экспоненциальное представление
plt.tickformat(1, axis='y')
```

```
# сохраняем
plt.savefig('/datasets/cumulative_revenue.jpg')
```

```
# отображение
plt.show()
```

```
/usr/local/lib/python3.9/site-packages/matplotlib/ticker.py:2448: UserWarning: tickformat not supported for the current backend.
  warnings.warn("tickformat not supported for the "
```

Сохраню не можем сделать. Сохраню вручную. Что-то опять не сохранилось. Попробую. Что-то опять не сохранилось. Сохраню на компьютер. Загружу в папку datasets. Что-то опять не сохранилось. Что-то опять не сохранилось. Что-то опять не сохранилось.

К оглавлению

4. Графики кумулятивных конверсий по группам

Построим графики кумулятивной конверсии по группам.

```
# кумулятивные кумулятивные конверсии по группам и кумулятивный средний чек по дням и группам
cumulative_conversion = cumulative[['date', 'group', 'cum_conversion']]
cumulative_conversion.sample(5)
```

	date	group	cum_conversion
29	2019-08-15	B	0.087359
11	2019-08-06	A	0.060316
25	2019-08-14	B	0.076400
7	2019-08-04	A	0.062939
27	2019-08-15	A	0.058232

```
# строим график выручки группы A
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_conversion[cumulative_conversion['group']=='A'], x='date', y='cum_conversion')
```

```
# строим график выручки группы B
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_conversion[cumulative_conversion['group']=='B'], x='date', y='cum_conversion')
```

```
# задаем названия
plt.title('Графики кумулятивных конверсий по группам')
plt.xlabel('Дата')
plt.ylabel('Конверсия')
plt.legend(['Группа A', 'Группа B'])
```

```
# отключаем экспоненциальное представление
plt.tickformat(1, axis='y')
```

```
# сохраняем
plt.savefig('/datasets/cumulative_conversion.jpg')
```

```
# отображение
plt.show()
```

```
/usr/local/lib/python3.9/site-packages/matplotlib/ticker.py:2448: UserWarning: tickformat not supported for the current backend.
  warnings.warn("tickformat not supported for the "
```

Сохраню не можем сделать. Сохраню вручную. Что-то опять не сохранилось. Попробую. Что-то опять не сохранилось. Сохраню на компьютер. Загружу в папку datasets. Что-то опять не сохранилось. Что-то опять не сохранилось. Что-то опять не сохранилось.

К оглавлению

5. Графики относительных изменений кумулятивных метрик группы B к группе A

Построим графики относительных изменений кумулятивных метрик. Добавим горизонтальную ось методом pyplot.axhline().

```
# кумулятивные кумулятивные конверсии по группам и кумулятивный средний чек по дням и группам
cumulative_conversion = cumulative[['date', 'group', 'cum_conversion']]
cumulative_conversion.sample(5)
```

	date	group	cum_conversion
29	2019-08-15	B	0.087359
11	2019-08-06	A	0.060316
25	2019-08-14	B	0.076400
7	2019-08-04	A	0.062939
27	2019-08-15	A	0.058232

```
# отделяем данные группы A
cumulative_conversion_a = cumulative_conversion[cumulative_conversion['group']=='A']
```

```
# отделяем данные группы B
cumulative_conversion_b = cumulative_conversion[cumulative_conversion['group']=='B']
```

```
# строим график выручки группы A
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_conversion_a, x='date', y='cum_conversion')
```

```
# строим график выручки группы B
plt.figure(dpi=300, figure=(15, 5))
sns.lineplot(data=cumulative_conversion_b, x='date', y='cum_conversion')
```

```
# задаем названия
plt.title('Графики кумулятивных конверсий по группам')
plt.xlabel('Дата')
plt.ylabel('Конверсия')
plt.legend(['Группа A', 'Группа B'])
```

```
# отключаем экспоненциальное представление
plt.tickformat(1, axis='y')
```

```
# сохраняем
plt.savefig('/datasets/cumulative_conversion.jpg')
```

```
# отображение
plt.show()
```

```
/usr/local/lib/python3.9/site-packages/matplotlib/ticker.py:2448: UserWarning: tickformat not supported for the current backend.
  warnings.warn("tickformat not supported for the "
```

Сохраню не можем сделать. Сохраню вручную. Что-то опять не сохранилось. Попробую. Что-то опять не сохранилось. Сохраню на компьютер. Загружу в папку datasets. Что-то опять не сохранилось. Что-то опять не сохранилось. Что-то опять не сохранилось.

К оглавлению

6. Точечный график стоимости заказов по пользователям

Построим точечный график стоимости заказов по пользователям.

```
px.scatter(orders, x='transaction_id', y='revenue', color='group', title='Точечный график стоимости заказов по пользователям', width=1000, height=500)
```


К оглавлению

7. Точечный график распределения заказов по пользователям

Посмотрим на распределение заказов по пользователям — построим точечный график.

```
px.scatter(orders, x='transaction_id', y='visitor_id', color='group', title='Точечный график распределения заказов по пользователям', width=1000, height=500)
```


К оглавлению

8. Расчет 95-го и 99-го перцентилей стоимости заказов. Выбор аномальных заказов пользователей

Посчитаем 95-й и 99-й перцентили стоимости заказов.

```
orders.revenue.quantile([0.95, 0.99])
```

```
0.95    53904.000
0.99   148579.000
Name: revenue, dtype: float64
```

Не более 5% заказов имеют стоимость свыше 53904 рублей. Отсеем аномальные заказы стоимостью свыше 148579 рублей.

К оглавлению

9. Точечный график количества заказов по пользователям

Посмотрим на распределение заказов по пользователям — построим точечный график.

```
px.scatter(orders, x='transaction_id', y='visitor_id', color='group', title='Точечный график распределения заказов по пользователям', width=1000, height=500)
```

К оглавлению

10. Расчет 95-го и 99-го перцентилей количества заказов. Выбор аномальных заказов пользователей

Посчитаем 95-й и 99-й перцентили количества заказов.

```
orders.groupby('visitor_id').agg({'transaction_id': 'nunique'}).transaction_id.quantile([0.95, 0.99])
```

```
0.95    2.000
0.99    4.000
Name: transaction_id, dtype: float64
```

Не более 5% пользователей совершали свыше 2 заказов. Отсеем аномальные заказы пользователей, совершившие свыше 4 заказов.

К оглавлению

11. Статистическая значимость различий в конверсии между группами по сырым данным

Посчитаем статистическую значимость различий в конверсии между группами по сырым данным. Сформулируем гипотезы. Нулевая: различий в конверсии между группами нет. Альтернативная: различия в конверсии между группами есть. Зададим уровень статистической значимости в 5%.

```
alpha = 0.05
```

```
# подготовка к расчетам
# количество заказов по пользователям
orders_by_users = orders.groupby('visitor_id').agg({'transaction_id': 'nunique'}).reset_index()
orders_by_users.columns = ['visitor_id', 'orders']
```

```
# объединяем с таблицей visitors
visitors_with_orders = visitors.merge(orders_by_users, on='visitor_id')
```

```
# пользователи группы A
visitors_a = visitors_with_orders[visitors_with_orders['group']=='A']
```

```
# пользователи группы B
visitors_b = visitors_with_orders[visitors_with_orders['group']=='B']
```

```
# расчет p-value
p = st.mannwhitneyu(visitors_a['orders'], visitors_b['orders'])[1]
```

```
# вывод
print('{:.3f}'.format(p))
```

```
0.011
```

```
# расчет относительного различия
print('{:.3f}'.format(visitors_b['orders'].mean()/visitors_a['orders'].mean()-1))
```

```
0.160
```

Гипотеза отвергнута. Статистическая значимость достигнута (значение статистики ниже 0.05). Относительный выигрыш группы B равен 16%.

Теперь мы проверим статистическую значимость различий в среднем чеке между группами. Мы сформулируем нулевую и альтернативную гипотезы. Нулевая: различий в среднем чеке между группами нет. Альтернативная: различия в среднем чеке между группами есть.

```
# расчет p-value
p = st.mannwhitneyu(orders[orders['group']=='A']['revenue'], orders[orders['group']=='B']['revenue'])[1]
```

```
# вывод
print('{:.3f}'.format(p))
```

```
0.829
```

```
# расчет относительного различия
print('{:.3f}'.format(orders[orders['group']=='B']['revenue'].mean()/orders[orders['group']=='A']['revenue'].mean()-1))
```

```
0.287
```

Гипотеза не отвергнута. Статистическая значимость не достигнута (значение статистики выше 0.05). Относительный выигрыш группы B равен 28.7%.

К оглавлению

12. Статистическая значимость различий в среднем чеке между группами по сырым данным

Посчитаем статистическую значимость различий в среднем чеке между группами по сырым данным. Сформулируем гипотезы. Нулевая: различий в среднем чеке между группами нет. Альтернативная: различия в среднем чеке между группами есть. Зададим уровень статистической значимости в 5%.

```
alpha = 0.05
```

```
# расчет p-value
p = st.mannwhitneyu(orders[orders['group']=='A']['revenue'], orders[orders['group']=='B']['revenue'])[1]
```

```
# вывод
print('{:.3f}'.format(p))
```

```
0.829
```

```
# расчет относительного различия
print('{:.3f}'.format(orders[orders['group']=='B']['revenue'].mean()/orders[orders['group']=='A']['revenue'].mean()-1))
```

```
0.287
```

Гипотеза не отвергнута. Статистическая значимость не достигнута (значение статистики выше 0.05). Относительный выигрыш группы B равен 28.7%.

К оглавлению

13. Статистическая значимость различий в конверсии между группами по очищенным данным

Посчитаем статистическую значимость различий в конверсии между группами по очищенным данным. Сформулируем гипотезы. Нулевая: различий в конверсии между группами нет. Альтернативная: различия в конверсии между группами есть. Зададим уровень статистической значимости в 5%.

```
alpha = 0.05
```

```
# подготовка к расчетам
# количество заказов по пользователям
orders_by_users = orders.groupby('visitor_id').agg({'transaction_id': 'nunique'}).reset_index()
orders_by_users.columns = ['visitor_id', 'orders']
```

```
# объединяем с таблицей visitors
visitors_with_orders = visitors.merge(orders_by_users, on='visitor_id')
```

```
# пользователи группы A
visitors_a = visitors_with_orders[visitors_with_orders['group']=='A']
```

```
# пользователи группы B
visitors_b = visitors_with_orders[visitors_with_orders['group']=='B']
```

```
# расчет p-value
p = st.mannwhitneyu(visitors_a['orders'], visitors_b['orders'])[1]
```

```
# вывод
print('{:.3f}'.format(p))
```

```
0.011
```

```
# расчет относительного различия
print('{:.3f}'.format(visitors_b['orders'].mean()/visitors_a['orders'].mean()-1))
```

```
0.160
```

Гипотеза отвергнута. Статистическая значимость достигнута (значение статистики ниже 0.05). Относительный выигрыш группы B равен 16%.

Теперь мы проверим статистическую значимость различий в среднем чеке между группами. Мы сформулируем нулевую и альтернативную гипотезы. Нулевая: различий в среднем чеке между группами нет. Альтернативная: различия в среднем чеке между группами есть.

```
# расчет p-value
p = st.mannwhitneyu(orders[orders['group']=='A']['revenue'], orders[orders['group']=='B']['revenue'])[1]
```

```
# вывод
print('{:.3f}'.format(p))
```

```
0.829
```

```
# расчет относительного различия
print('{:.3f}'.format(orders[orders['group']=='B']['revenue'].mean()/orders[orders['group']=='A']['revenue'].mean()-1))
```

```
0.287
```

Гипотеза не отвергнута. Статистическая значимость не достигнута (значение статистики выше 0.05). Относительный выигрыш группы B равен 28.7%.

К оглавлению

14. Статистическая значимость различий в среднем чеке между группами по очищенным данным

Посчитаем статистическую значимость различий в среднем чеке между группами по очищенным данным. Сформулируем гипотезы. Нулевая: различий в среднем чеке между группами нет. Альтернативная: различия в среднем чеке между группами есть. Зададим уровень статистической значимости в 5%.

```
alpha = 0.05
```

```
# расчет p-value
p = st.mannwhitneyu(orders[orders['group']=='A']['revenue'], orders[orders['group']=='B']['revenue'])[1]
```

```
# вывод
print('{:.3f}'.format(p))
```

```
0.829
```

```
# расчет относительного различия
print('{:.3f}'.format(orders[orders['group']=='B']['revenue'].mean()/orders[orders['group']=='A']['revenue'].mean()-1))
```

```
0.287
```

Гипотеза не отвергнута. Статистическая значимость не достигнута (значение статистики выше 0.05). Относительный выигрыш группы B равен 28.7%.

К оглавлению

15. Принятие решения по результатам теста и описание результатов теста

Вывод:

- **Гипотезы 8 и 0 — на первом месте по приоритету. Гипотезы 7 и 2 — на третьем месте по приоритету. Гипотезы 6 и 1 — на последнем месте. Гипотезы 4 и 3 — на предпоследнем месте. Гипотезы 5 — на 5-ом месте. Для дальнейшей работы оставлю все гипотезы.**
- **Кумулятивные агрегаты по группам — стабильно растут.**
- **Кумулятивные метрики по группам — стабильно растут.**
- **Кумулятивная выручка по группам — стабильно растёт. Группа B — лидер.**
- **Кумулятивные конверсии по группам — стабильно растут. Группа B — лидер.**
- **Относительные изменения кумулятивных метрик группы B к группе A — стабильно растут.**
- **Точечный график стоимости заказов по пользователям — 1 заказ на 1 пользователя. Отсеем аномальные заказы стоимостью свыше 148579 рублей.**
- **Точечный график распределения заказов по пользователям — 1 заказ на 1 пользователя. Отсеем аномальные заказы пользователей, совершившие свыше 4 заказов.**
- **Статистическая значимость различий в конверсии между группами по сырым данным — достигнута. Относительный выигрыш группы B равен 16%.**
- **Статистическая значимость различий в среднем чеке между группами по сырым данным — не достигнута. Относительный выигрыш группы B равен 28.7%.**
- **Статистическая значимость различий в конверсии между группами по очищенным данным — достигнута. Относительный выигрыш группы B равен 16%.**
- **Статистическая значимость различий в среднем чеке между группами по очищенным данным — не достигнута. Относительный выигрыш группы B равен 28.7%.**

Рекомендую остановить тест, зафиксировать победу группы B.

К оглавлению

```
In [29]: plt.figure(figure=None, dpi=None, facecolor='w', edgecolor='k')
plt.plot(df['Customer_ID'], df['Amount'], 'r.')
plt.xlabel('Customer_ID')
plt.ylabel('Amount')
plt.title('Диаграмма рассеяния распределения клиентов')
plt.grid(True)
plt.show()
```

И снова пусть кроме 8 клиентов стоимостью сдачи 5, мало или вовсе нулю и с небольшим числом.

6.отображение

6. Точечный график количества заказов по пользователям

Подготовим количество заказов по пользователям и построим на рисунке:

```
In [30]: df_customer = df.groupby('Customer_ID').count()
df_customer['Amount'].sort_values(ascending=False)
```

```
Out[30]: Customer_ID
         1    10000
         2     5000
         3     3333
         4     2500
         5     2000
         6     1667
         7     1429
         8     1250
         9     1111
         10    1000
         Name: Amount, dtype: int64
```

```
In [31]: plt.figure(figure=None, dpi=None, facecolor='w', edgecolor='k')
plt.plot(df_customer['Amount'], '.')
plt.xlabel('Customer_ID')
plt.ylabel('Amount')
plt.title('Точечный график количества заказов по одному пользователю')
plt.grid(True)
plt.show()
```

Большинство покупателей заказывали более чем один раз, пользователи 1-3 исключение. Построим точечную диаграмму числа заказов на одного пользователя:

```
In [32]: df_customer = df.groupby('Customer_ID').count()
df_customer['Amount'].sort_values(ascending=False)
```

```
Out[32]: Customer_ID
         1    10000
         2     5000
         3     3333
         4     2500
         5     2000
         6     1667
         7     1429
         8     1250
         9     1111
         10    1000
         Name: Amount, dtype: int64
```

```
In [33]: plt.figure(figure=None, dpi=None, facecolor='w', edgecolor='k')
plt.plot(df_customer['Amount'], '.')
plt.xlabel('Customer_ID')
plt.ylabel('Amount')
plt.title('Точечный график количества заказов на одного пользователя')
plt.grid(True)
plt.show()
```

Есть выражение 1-3 исключение

6.отображение

7. 95 и 99 процентиля количества заказов на пользователя. Граница для определения аномальных пользователей

Определим 95 и 99 процентили количества заказов на одного пользователя методом quantile(). Добавляем бибколку:

```
In [34]: df_customer = df.groupby('Customer_ID').count()
df_customer['Amount'].sort_values(ascending=False)
```

```
Out[34]: Customer_ID
         1    10000
         2     5000
         3     3333
         4     2500
         5     2000
         6     1667
         7     1429
         8     1250
         9     1111
         10    1000
         Name: Amount, dtype: int64
```

На более 95 процентилей заказчиков более двух заказов. И 1% пользователей заказывали более четырёх раз. Граница 3 заказа на одного пользователя не слишком сильно высока.

6.отображение

8. Гистограмма стоимостей заказов

Построим гистограмму стоимостей заказов методом hist():

```
In [35]: df['Amount'].hist()
```

```
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8d0a0a0a90>
```

Построим гистограмму стоимостей заказов, ограничив диапазон по оси ординат до 100:000:

```
In [36]: plt.figure(figure=None, dpi=None, facecolor='w', edgecolor='k')
df['Amount'].hist(bins=50)
plt.xlabel('Amount')
plt.ylabel('Count')
plt.title('Гистограмма распределения стоимостей заказов')
plt.grid(True)
plt.show()
```

Ребята стоимостей заказов несколько больше, среднее значение заказов ненадлежащее.

6.отображение

9. 95 и 99 процентиля стоимости заказов. Граница для определения аномальных заказов

Определим 95 и 99 процентили стоимостей заказов:

```
In [37]: df.sort_values(ascending=False, [0], [0])
```

```
Out[37]: Customer_ID  Amount
         1           1   1000
         2           1   1000
         3           1   1000
         4           1   1000
         5           1   1000
         6           1   1000
         7           1   1000
         8           1   1000
         9           1   1000
         10          1   1000
```

Есть больше 5% заказов дороже 20:000 рублей и не более 1% дороже 90:000. Граница 90:000. Не аномально громоздко стоимости заказов.

Отличная работа!

Гистограмма сделана верно. Молодцы, что дополнительно использовали matplotlib.

6.отображение

10. Статистическая значимость различий в конверсии между группами по «сырым» данным

Подготовим отчёт таблицу. data:

```
In [38]: cols=['Customer_ID', 'conversion', 'group']
for i in range(10):
    cols.append('conversion_'+str(i))
cols.append('conversion_mean')
cols.append('conversion_mean_diff')
cols.append('conversion_mean_diff_percent')
cols.append('conversion_pvalue')
cols.append('conversion_pvalue_adjusted')
cols.append('conversion_pvalue_adjusted_reject')
cols.append('conversion_pvalue_adjusted_method')
cols.append('conversion_pvalue_adjusted_alpha')
cols.append('conversion_pvalue_adjusted_power')
cols.append('conversion_pvalue_adjusted_mean')
cols.append('conversion_pvalue_adjusted_mean_diff')
cols.append('conversion_pvalue_adjusted_mean_diff_percent')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_reject')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_method')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_alpha')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_power')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_mean')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_mean_diff')
cols.append('conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_mean_diff_percent')
```

```
In [39]: data = pd.DataFrame(columns=cols)
data['Customer_ID'] = df['Customer_ID']
data['conversion'] = df['Amount']
data['group'] = df['Customer_ID']
data['conversion_0'] = df['Amount']
data['conversion_1'] = df['Amount']
data['conversion_2'] = df['Amount']
data['conversion_3'] = df['Amount']
data['conversion_4'] = df['Amount']
data['conversion_5'] = df['Amount']
data['conversion_6'] = df['Amount']
data['conversion_7'] = df['Amount']
data['conversion_8'] = df['Amount']
data['conversion_9'] = df['Amount']
data['conversion_mean'] = df['Amount']
data['conversion_mean_diff'] = df['Amount']
data['conversion_mean_diff_percent'] = df['Amount']
data['conversion_pvalue'] = df['Amount']
data['conversion_pvalue_adjusted'] = df['Amount']
data['conversion_pvalue_adjusted_reject'] = df['Amount']
data['conversion_pvalue_adjusted_method'] = df['Amount']
data['conversion_pvalue_adjusted_alpha'] = df['Amount']
data['conversion_pvalue_adjusted_power'] = df['Amount']
data['conversion_pvalue_adjusted_mean'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_reject'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_method'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_alpha'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_power'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_mean'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_mean_diff'] = df['Amount']
data['conversion_pvalue_adjusted_mean_diff_percent_pvalue_adjusted_mean_diff_percent'] = df['Amount']
```

```
In [40]: data.sort_values(ascending=False, [0], [0])
```

```
Out[40]:
         Customer_ID  conversion  conversion_0  conversion_1  conversion_2  conversion_3  conversion_4  conversion_5  conversion_6  conversion_7  conversion_8  conversion_9  conversion_mean  conversion_mean_diff  conversion_mean_diff_percent  conversion_pvalue  conversion_pvalue_adjusted  conversion_pvalue_adjusted_reject  conversion_pvalue_adjusted_method  conversion_pvalue_adjusted_alpha  conversion_pvalue_adjusted_power  conversion_pvalue_adjusted_mean  conversion_pvalue_adjusted_mean_diff  conversion_pvalue_adjusted_mean_diff_percent  group
         1           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         2           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         3           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         4           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         5           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         6           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         7           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         8           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         9           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
         10          1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000                                          1000
```

Названия столбцов таблицы:

```
In [41]: data.columns
```

```
Out[41]: Index(['Customer_ID', 'conversion', 'group', 'conversion_0',
                'conversion_1', 'conversion_2', 'conversion_3', 'conversion_4',
                'conversion_5', 'conversion_6', 'conversion_7', 'conversion_8',
                'conversion_9', 'conversion_mean', 'conversion_mean_diff',
                'conversion_mean_diff_percent', 'conversion_pvalue',
                'conversion_pvalue_adjusted', 'conversion_pvalue_adjusted_reject',
                'conversion_pvalue_adjusted_method', 'conversion_pvalue_adjusted_alpha',
                'conversion_pvalue_adjusted_power', 'conversion_pvalue_adjusted_mean',
                'conversion_pvalue_adjusted_mean_diff',
                'conversion_pvalue_adjusted_mean_diff_percent'],
               dtype='object')
```

Сколько строк в таблице:

```
In [42]: data.shape
```

```
Out[42]: (10000, 25)
```

10 000 строк. Сколько уникальных клиентов:

```
In [43]: data['Customer_ID'].nunique()
```

```
Out[43]: 10
```

10 клиентов. Сколько клиентов в каждой группе:

```
In [44]: data.groupby('group').count()
```

```
Out[44]:
         Customer_ID  conversion  conversion_0  conversion_1  conversion_2  conversion_3  conversion_4  conversion_5  conversion_6  conversion_7  conversion_8  conversion_9  conversion_mean  conversion_mean_diff  conversion_mean_diff_percent  conversion_pvalue  conversion_pvalue_adjusted  conversion_pvalue_adjusted_reject  conversion_pvalue_adjusted_method  conversion_pvalue_adjusted_alpha  conversion_pvalue_adjusted_power  conversion_pvalue_adjusted_mean  conversion_pvalue_adjusted_mean_diff  conversion_pvalue_adjusted_mean_diff_percent
         group
         1           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         2           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         3           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         4           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         5           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         6           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         7           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         8           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         9           1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
         10          1   1000          1000          1000          1000          1000          1000          1000          1000          1000          1000          1000             1000                  1000                          1000               1000                        1000                               1000                               1000                              1000                              1000                             1000                                  1000                                          1000
```

По каждому разному в каждом между группами в 1-4 раза. Статистическая значимая разница. Значимая группа 10 на 10% выше

6.отображение

11. Статистическая значимость различий в среднем чеке заказа между группами по «сырым» данным

Сравнение средних чеков:

Уровень значимости, задаваемый при проверке гипотезы. Вероятность ошибки. Возьмем 5% значения:

```
In [45]: alpha = 0.05
```

Проверим критерий о справедливости для каждого из 10 «сырых» данных распределений:

Уровень значимости, задаваемый при проверке гипотезы. Вероятность ошибки. Возьмем 5% значения:

```
In [46]: alpha = 0.05
```

Проверим критерий о справедливости для каждого из 10 «сырых» данных распределений:

```
In [47]: from scipy.stats import ttest_ind
for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)]))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

По каждому разному в среднем чеке между группами в 1-4 раза. Средний чек группы 10 на 10% выше

6.отображение

12. Статистическая значимость различий в среднем чеке заказа между группами по «сгенерированным» данным

Сравним среднее по каждому 1 группе между Группой 1 и остальными. Для каждого между каждой платежкой и остальными. Сравнение проверяем выборки которые мы можем по платежкам по группам по.

```
In [48]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)]))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Анализ 10 значимости показателей

Сравнение среднего по каждому между группами. Сравнение статистической значимости различий в среднем между группами. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [49]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Сравнение среднего по каждому между группами. Сравнение статистической значимости различий в среднем между группами. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [50]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Результаты по каждому приблизительно не совпадаются. Группа 10 по каждому лучше группы 10.

6.отображение

13. Статистическая значимость различий в среднем чеке заказа между группами по «сгенерированным» данным

Сравнение среднего по каждому 1 группе между Группой 1 и остальными. Для каждого между каждой платежкой и остальными. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [51]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Сравнение среднего по каждому между группами. Сравнение статистической значимости различий в среднем между группами. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [52]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

14. Разрывы по результатам теста

Сравнение значимости среднего по каждому между группами. Сравнение результатов разных и каждого в таблицы статистической значимости.

```
In [53]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

15. Статистическая значимость различий в среднем чеке заказа между группами по «сгенерированным» данным

Сравнение среднего по каждому 1 группе между Группой 1 и остальными. Для каждого между каждой платежкой и остальными. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [54]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Сравнение среднего по каждому между группами. Сравнение статистической значимости различий в среднем между группами. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [55]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

16. Разрывы по результатам теста

Сравнение значимости среднего по каждому между группами. Сравнение результатов разных и каждого в таблицы статистической значимости.

```
In [56]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

17. Статистическая значимость различий в среднем чеке заказа между группами по «сгенерированным» данным

Сравнение среднего по каждому 1 группе между Группой 1 и остальными. Для каждого между каждой платежкой и остальными. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [57]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Сравнение среднего по каждому между группами. Сравнение статистической значимости различий в среднем между группами. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [58]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

18. Разрывы по результатам теста

Сравнение значимости среднего по каждому между группами. Сравнение результатов разных и каждого в таблицы статистической значимости.

```
In [59]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

19. Статистическая значимость различий в среднем чеке заказа между группами по «сгенерированным» данным

Сравнение среднего по каждому 1 группе между Группой 1 и остальными. Для каждого между каждой платежкой и остальными. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [60]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Сравнение среднего по каждому между группами. Сравнение статистической значимости различий в среднем между группами. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [61]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

20. Разрывы по результатам теста

Сравнение значимости среднего по каждому между группами. Сравнение результатов разных и каждого в таблицы статистической значимости.

```
In [62]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

21. Статистическая значимость различий в среднем чеке заказа между группами по «сгенерированным» данным

Сравнение среднего по каждому 1 группе между Группой 1 и остальными. Для каждого между каждой платежкой и остальными. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [63]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

Сравнение среднего по каждому между группами. Сравнение статистической значимости различий в среднем между группами. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [64]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

22. Разрывы по результатам теста

Сравнение значимости среднего по каждому между группами. Сравнение результатов разных и каждого в таблицы статистической значимости.

```
In [65]: for i in range(10):
    print('conversion_'+str(i), ttest_ind(data['conversion'], data['conversion_'+str(i)], equal_var=False))
```

```
conversion_0 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_1 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_2 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_3 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_4 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_5 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_6 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_7 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_8 Ttest_indResult(statistic=nan, pvalue=nan)
conversion_9 Ttest_indResult(statistic=nan, pvalue=nan)
```

6.отображение

23. Статистическая значимость различий в среднем чеке заказа между группами по «сгенерированным» данным

Сравнение среднего по каждому 1 группе между Группой 1 и остальными. Для каждого между каждой платежкой и остальными. Сравнение выборки выборки которые мы можем по платежкам по группам по.

```
In [66]: for i in range(10):
    print('conversion_'+str(i),
```

Итоги и обнадеживающие факторы, тот следует отмечать в прошлом или нарушающих. Посмотрим на уже! комментарии, сообщений на экономическую ситуацию

Сегодня в ходе торгов, в ходе торгов, в ходе торгов, в ходе торгов, в ходе торгов, в ходе торгов, Как то, что сообщается. Сегодня различные комментарии между грузовыми перевозками, что результаты сделки 50 лучше прошлых.

Нет статистическая экономия различных на грузовых, что между грузовыми на 50 сделки. Данные, что по сообщениям. Почему грузовые перевозки, что грузовые перевозки, что грузовые перевозки, что грузовые перевозки.

Сегодня существенный выигрыш сделки 50 лучше сделок. Что касается, что сверхприбыльных в основном числе заказов, хейс и основные сплиц, результат заказов выбрасывает.

Сегодня! на заказов сделки 50 в выигрыш выгодным, заказы быстро сделки 50 на заказов. И заказы, заказы хейсовощерков. Сегодня! на заказов сделки 50 в выигрыш выгодным, заказы быстро сделки 50 на заказов.

Разные результаты и различные решения нам необходимости на 50 в сделки. Что заказы.

Сегодня существенный комментарии комментарии, есть выбросы. Существенные комментарии сделки 50 на 50 Что касается комментарии сделки 50 лучше.

И основные числа сделки 50 основные показатели сделки 50, заказы, разные заказы на хейсовощерков.

Экспортеры! Экспортеры! экспортеры! поскорее! поскорее! прогнозы, есть выбросы. Существенные комментарии сделки 50 на 50 Что касается комментарии сделки 50 лучше.

Экспортеры! Экспортеры! экспортеры! поскорее! прогнозы, есть выбросы. Существенные комментарии сделки 50 на 50 Что касается комментарии сделки 50 лучше.

На Всем РТС заказов рынков 200 000 работает на Всем РТС рынке 50 200. Примеры 50 200 на значение прежнего сложности заказов.

Что касается! цветочно-основном между грузовыми 2,4% на что основные основные результаты. Результаты сделки 50 на 50 Что касается! цветочно-основном между грузовыми 2,4% на что основные основные результаты. Результаты сделки 50 на 50 Что касается! сделки 50 лучше.

Сегодня! Разные результаты различной в сделки 50 лучше. Разные заказы 2,4% 50 лучше. Сегодня! Сегодня 50 лучше. Сегодня! Сегодня 50 лучше. Сегодня! Сегодня