

Анализ результатов A/B тестирования и приоритизация гипотез

Описание проекта

Вы — аналитик крупного интернет-магазина. Вместе с отделом маркетинга вы подготовили список гипотез для увеличения выручки.

Приоритизируйте гипотезы, запустите A/B-тест и проанализируйте результаты.

Описание данных

Данные представлены в файле `/datasets/hypothesis.csv`

- `Hypothesis` — краткое описание гипотезы;
- `Reach` — охват пользователей по 10-балльной шкале;
- `Impact` — влияние на пользователей по 10-балльной шкале;
- `Confidence` — уверенность в гипотезе по 10-балльной шкале;
- `Efforts` — затраты ресурсов на проверку гипотезы по 10-балльной шкале. Чем больше значение `Efforts`, тем дороже проверка гипотезы.

Данные для второй части в файле `/datasets/orders.csv`

- `transactionId` — идентификатор заказа;
- `visitorId` — идентификатор пользователя, совершившего заказ;
- `date` — дата, когда был совершён заказ;
- `revenue` — выручка заказа;
- `group` — группа A/B-теста, в которую попал заказ.

Файл `/datasets/visitors.csv`

- `date` — дата;
- `group` — группа A/B-теста;
- `visitors` — количество пользователей в указанную дату в указанной группе A/B-теста

Оглавление

- [Часть 1. Приоритизация гипотез](#)
- [Часть 2. Анализ A/B теста](#)
- [Часть 3. Общий вывод и рекомендации](#)

Часть 1. Приоритизация гипотез.

В файле `/datasets/hypothesis.csv` 9 гипотез по увеличению выручки интернет-магазина с указанными параметрами `Reach`, `Impact`, `Confidence`, `Effort`.

```
In [1]: # For better figure's quality
%config InlineBackend.figure_format = 'retina'
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import mannwhitneyu, shapiro
```

```
In [2]: # workaround to use praktikum file system as well as local windows system
try:
    hypothesis = pd.read_csv('/datasets/hypothesis.csv')
except:
    hypothesis = pd.read_csv('datasets/hypothesis.csv')
```

Применим фреймворки ICE и RICE для приоритизации гипотез и сравним результаты.

Для полного отображения текста каждой гипотезы изменим опцию ограничения ширины столбца.

```
In [3]: pd.set_option('display.max_colwidth', 0)
```

```
In [4]: hypothesis
```

```
Out[4]:
```

	Hypothesis	Reach	Impact	Confidence	Efforts
0	Добавить два новых канала привлечения трафика, что позволит привлекать на 30% больше пользователей	3	10	8	6
1	Запустить собственную службу доставки, что сократит срок доставки заказов	2	5	4	10
2	Добавить блоки рекомендаций товаров на сайт интернет магазина, чтобы повысить конверсию и средний чек заказа	8	3	7	3
3	Изменить структура категорий, что увеличит конверсию, т.к. пользователи быстрее найдут нужный товар	8	3	3	8
4	Изменить цвет фона главной страницы, чтобы увеличить вовлеченность пользователей	3	1	1	1
5	Добавить страницу отзывов клиентов о магазине, что позволит увеличить количество заказов	3	2	2	3
6	Показать на главной странице баннеры с актуальными акциями и распродажами, чтобы увеличить конверсию	5	3	8	3
7	Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок	10	7	8	5
8	Запустить акцию, дающую скидку на товар в день рождения	1	9	9	5

В загруженной таблице уже посчитаны нужные параметры. Рассчитаем показатели ICE и RICE .

Показатели рассчитывают по следующим формулам:

$$ICE = \frac{Impact * Confidence}{Efforts}$$

$$RICE = \frac{Reach * Impact * Confidence}{Efforts}$$

```
In [5]: hypothesis['ICE'] = hypothesis['Impact'] * hypothesis['Confidence'] / hypothesis['Efforts']
```

```
In [6]: hypothesis['RICE'] = hypothesis['Reach'] * hypothesis['Impact'] * hypothesis['Confidence'] / hypothesis['Efforts']
```

```
In [7]: hypothesis
```

```
Out[7]:
```

	Hypothesis	Reach	Impact	Confidence	Efforts	ICE	RICE
0	Добавить два новых канала привлечения трафика, что позволит привлекать на 30% больше пользователей	3	10	8	6	13.333333	40.0
1	Запустить собственную службу доставки, что сократит срок доставки заказов	2	5	4	10	2.000000	4.0
2	Добавить блоки рекомендаций товаров на сайт интернет магазина, чтобы повысить конверсию и средний чек заказа	8	3	7	3	7.000000	56.0
3	Изменить структура категорий, что увеличит конверсию, т.к. пользователи быстрее найдут нужный товар	8	3	3	8	1.125000	9.0
4	Изменить цвет фона главной страницы, чтобы увеличить вовлеченность пользователей	3	1	1	1	1.000000	3.0
5	Добавить страницу отзывов клиентов о магазине, что позволит увеличить количество заказов	3	2	2	3	1.333333	4.0
6	Показать на главной странице баннеры с актуальными акциями и распродажами, чтобы увеличить конверсию	5	3	8	3	8.000000	40.0
7	Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок	10	7	8	5	11.200000	112.0
8	Запустить акцию, дающую скидку на товар в день рождения	1	9	9	5	16.200000	16.2

Отсортируем гипотезы по ICE по убыванию.

```
In [8]: hypothesis.sort_values('ICE', ascending=False)
```

Out[8]:

	Hypothesis	Reach	Impact	Confidence	Efforts	ICE	RICE
8	Запустить акцию, дающую скидку на товар в день рождения	1	9	9	5	16.200000	16.2
0	Добавить два новых канала привлечения трафика, что позволит привлекать на 30% больше пользователей	3	10	8	6	13.333333	40.0
7	Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок	10	7	8	5	11.200000	112.0
6	Показать на главной странице баннеры с актуальными акциями и распродажами, чтобы увеличить конверсию	5	3	8	3	8.000000	40.0
2	Добавить блоки рекомендаций товаров на сайт интернет магазина, чтобы повысить конверсию и средний чек заказа	8	3	7	3	7.000000	56.0
1	Запустить собственную службу доставки, что сократит срок доставки заказов	2	5	4	10	2.000000	4.0
5	Добавить страницу отзывов клиентов о магазине, что позволит увеличить количество заказов	3	2	2	3	1.333333	4.0
3	Изменить структура категорий, что увеличит конверсию, т.к. пользователи быстрее найдут нужный товар	8	3	3	8	1.125000	9.0
4	Изменить цвет фона главной страницы, чтобы увеличить вовлеченность пользователей	3	1	1	1	1.000000	3.0

In [9]:

hypothesis.sort_values('RICE', ascending=False)

Out[9]:

	Hypothesis	Reach	Impact	Confidence	Efforts	ICE	RICE
7	Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок	10	7	8	5	11.200000	112.0
2	Добавить блоки рекомендаций товаров на сайт интернет магазина, чтобы повысить конверсию и средний чек заказа	8	3	7	3	7.000000	56.0
0	Добавить два новых канала привлечения трафика, что позволит привлекать на 30% больше пользователей	3	10	8	6	13.333333	40.0
6	Показать на главной странице баннеры с актуальными акциями и распродажами, чтобы увеличить конверсию	5	3	8	3	8.000000	40.0
8	Запустить акцию, дающую скидку на товар в день рождения	1	9	9	5	16.200000	16.2
3	Изменить структура категорий, что увеличит конверсию, т.к. пользователи быстрее найдут нужный товар	8	3	3	8	1.125000	9.0
1	Запустить собственную службу доставки, что сократит срок доставки заказов	2	5	4	10	2.000000	4.0
5	Добавить страницу отзывов клиентов о магазине, что позволит увеличить количество заказов	3	2	2	3	1.333333	4.0
4	Изменить цвет фона главной страницы, чтобы увеличить вовлеченность пользователей	3	1	1	1	1.000000	3.0

Расчет ICE вывел в лидеры гипотезы 8 и 0 с наибольшим влиянием (Impact), но низким охватом (Reach). С учетом охвата метрика RICE вывела на первое место гипотезу 7 с наибольшим охватом, достаточно большим влиянием и не очень большими затратами на внедрение (Efforts). На втором месте по расчету RICE оказалась гипотеза 2 также с высоким охватом, но небольшим влиянием, зато с большой уверенностью (Confidence) и небольшими затратами.

Таким образом, для правильной приоритизации гипотез следует учитывать все основные параметры, влияющие на принятие решения, как например RICE в данном случае.

Часть 2. Анализ A/B-теста.

Мы провели A/B-тест и получили результаты, которые описаны в файлах /datasets/orders.csv и /datasets/visitors.csv.

Загрузим данные и проверим общие параметры.

In [10]:

```
# workaround to use praktikum file system as well as local windows system
try:
    orders = pd.read_csv('/datasets/orders.csv')
except:
    orders = pd.read_csv('datasets/orders.csv')
```

In [11]:

orders.head()

```
Out[11]:
```

	transactionId	visitorId	date	revenue	group
0	3667963787	3312258926	2019-08-15	1650	B
1	2804400009	3642806036	2019-08-15	730	B
2	2961555356	4069496402	2019-08-15	400	A
3	3797467345	1196621759	2019-08-15	9759	B
4	2282983706	2322279887	2019-08-15	2308	B

```
In [12]: orders.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   transactionId    1197 non-null   int64
1   visitorId        1197 non-null   int64
2   date             1197 non-null   object
3   revenue          1197 non-null   int64
4   group            1197 non-null   object
dtypes: int64(3), object(2)
memory usage: 46.9+ KB
```

```
In [13]: orders.duplicated().sum()
```

```
Out[13]: 0
```

Дубликатов и пропусков нет, проверим распределение данных в численных показателях. Дата хранится в строке, преобразуем для удобства в форматы pandas даты-времени.

```
In [14]: orders['date'] = pd.to_datetime(orders['date'], format='%Y-%m-%d')
```

```
In [15]: orders['date'].min()
```

```
Out[15]: Timestamp('2019-08-01 00:00:00')
```

```
In [16]: orders['date'].max()
```

```
Out[16]: Timestamp('2019-08-31 00:00:00')
```

```
In [17]: orders['revenue'].describe()
```

```
Out[17]: count    1.197000e+03
mean      8.348006e+03
std       3.919113e+04
min       5.000000e+01
25%      1.220000e+03
50%      2.978000e+03
75%      8.290000e+03
max      1.294500e+06
Name: revenue, dtype: float64
```

```
In [18]: orders['group'].value_counts()
```

```
Out[18]: B    640
A     557
Name: group, dtype: int64
```

В таблице `visitors` содержится информация о покупках в августе 2019 года с привязкой к идентификатору посетителя, выручкой и разбиением на группы А/В. Группы А и В неравномерно распределены, в группе В на 90 покупателей больше, но с учетом большой базы не будем удалять лишние данные, лишь учтем это при дальнейшем анализе.

```
In [19]: # workaround to use praktikum file system as well as local windows system
try:
    visitors = pd.read_csv('/datasets/visitors.csv')
except:
    visitors = pd.read_csv('datasets/visitors.csv')
```

```
In [20]: visitors.head()
```

```
Out[20]:
```

	date	group	visitors
--	------	-------	----------

	date	group	visitors
0	2019-08-01	A	719
1	2019-08-02	A	619
2	2019-08-03	A	507
3	2019-08-04	A	717
4	2019-08-05	A	756

In [21]: `visitors.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0    date      62 non-null    object
1    group      62 non-null    object
2    visitors  62 non-null    int64
dtypes: int64(1), object(2)
memory usage: 1.6+ KB
```

In [22]: `visitors.duplicated().sum()`

Out[22]: 0

Дубликатов и пропусков в таблице `visitors` нет. Преобразуем дату в формат даты-времени и посмотрим на распределения.

In [23]: `visitors['date'] = pd.to_datetime(visitors['date'], format='%Y-%m-%d')`

In [24]: `visitors['date'].min()`

Out[24]: `Timestamp('2019-08-01 00:00:00')`

In [25]: `visitors['date'].max()`

Out[25]: `Timestamp('2019-08-31 00:00:00')`

In [26]: `visitors['group'].value_counts()`

Out[26]:

B	31
A	31

Name: group, dtype: int64

In [27]: `visitors['visitors'].describe()`

Out[27]:

count	62.000000
mean	607.290323
std	114.400560
min	361.000000
25%	534.000000
50%	624.500000
75%	710.500000
max	770.000000

Name: visitors, dtype: float64

В таблице `visitors` на каждый день содержится информация о том, сколько посетителей было в каждой из групп в течение августа 2019 года. Результаты A/B теста готовы к анализу.

Проанализируем результаты A/B теста.

1. Исследование кумулятивной выручки по группам

In [28]:

```
orders.pivot_table(
    index='date',
    columns='group',
    values='revenue',
    aggfunc='sum').cumsum(axis=0).plot(grid=True,
                                       figsize=(13,5),
                                       title='Кумулятивная выручка по группам по дням')
```

```
);
plt.xlabel('Дата');
plt.ylabel('Кумулятивная выручка');
```



```
In [29]: orders.groupby('group')['revenue'].sum()
```

```
Out[29]: group
A      4084803
B      5907760
Name: revenue, dtype: int64
```

Как видно на графике 19 августа 2019 года в группе В произошла крупная покупка, а далее покупки в группах происходили примерно одинаково. В результате общая выручка группы В превысила выручку группы А на 1.8 млн.

Проверим, что за покупка была в группе В 19 августа.

```
In [30]: orders[(orders['group'] == 'B') & (orders['date'] == '2019-08-19')]
```

```
Out[30]:
```

	transactionId	visitorId	date	revenue	group
315	1939633950	157579733	2019-08-19	2330	B
319	2414788465	2247745159	2019-08-19	1435	B
320	1652782757	2763027084	2019-08-19	1460	B
322	3154167297	2091902649	2019-08-19	250	B
363	2096847394	4251115836	2019-08-19	2500	B
367	1033377712	3860669668	2019-08-19	3990	B
368	3268879337	2536252163	2019-08-19	20026	B
423	4161654914	990904712	2019-08-19	11249	B
424	2609443403	2406916810	2019-08-19	470	B
425	590470918	1920142716	2019-08-19	1294500	B
426	19467429	2397843657	2019-08-19	2630	B
427	3019510679	2397843657	2019-08-19	1460	B
429	2436528253	1060341621	2019-08-19	1100	B
459	863448794	2738601405	2019-08-19	50	B
461	2683113224	154576532	2019-08-19	26550	B
463	4170537404	290022471	2019-08-19	16480	B
486	752294429	2758621772	2019-08-19	1130	B
523	1739748874	522292794	2019-08-19	4880	B
526	1654949825	1335154194	2019-08-19	500	B
527	1570513684	2837914161	2019-08-19	33405	B

	transactionId	visitorId	date	revenue	group
529	90109939	2075769885	2019-08-19	3525	B

В группе В произошла очень крупная транзакция 590470918, которая принесла выручки на 1,3 млн., что и сказалось на результате теста.

```
In [31]: orders.query('transactionId != "590470918"]').pivot_table(
    index='date',
    columns='group',
    values='revenue',
    aggfunc='sum').cumsum(axis=0).plot(grid=True,
                                       figsize=(13,5),
                                       title='Кумулятивная выручка по группам по дням'
                                       );

plt.xlabel('Дата');
plt.ylabel('Кумулятивная выручка');
```



Если убрать эту транзакцию, кумулятивная выручка по группе В также лидирует, но уже не так значительно, продажи по двум группам шли в одном темпе, кроме начала месяца, когда группа В увеличила выручку. **Если судить только по кумулятивной выручке группа В показывает лучшие результаты.**

2. Исследование кумулятивного среднего чека по группам

Для расчета среднего чека рассчитаем кумулятивную выручку и число заказов, а затем разделим одно на другое.

```
In [32]: revenues_cum = orders.pivot_table(
    index='date',
    columns='group',
    values='revenue',
    aggfunc='sum').cumsum(axis=0)
```

```
In [33]: n_orders_cum = orders.pivot_table(
    index='date',
    columns='group',
    values='transactionId',
    aggfunc='nunique').cumsum(axis=0)
```

```
In [34]: revenues_cum.head()
```

```
Out[34]:
```

	group	A	B
	date		
	2019-08-01	148579	101217
	2019-08-02	242401	266748
	2019-08-03	354874	380996
	2019-08-04	425699	489567

group	A	B
date		
2019-08-05	549917	581995

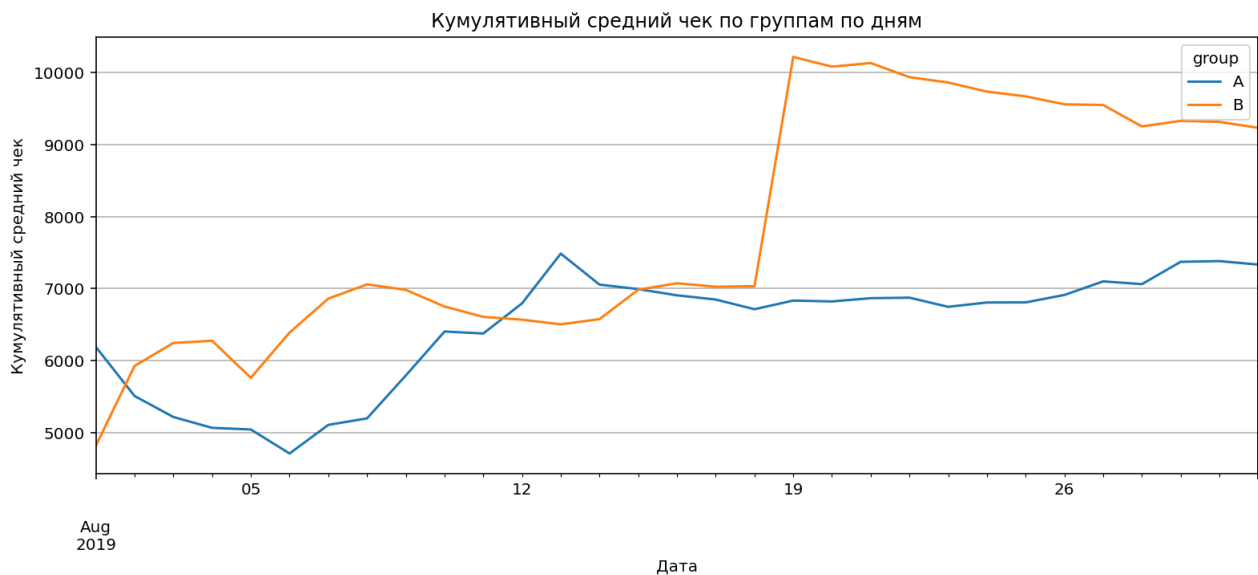
```
In [35]: n_orders_cum.head()
```

```
Out[35]:
```

group	A	B
date		
2019-08-01	24	21
2019-08-02	44	45
2019-08-03	68	61
2019-08-04	84	78
2019-08-05	109	101

```
In [36]: mean_bill = revenues_cum.divide(n_orders_cum)
```

```
In [37]: mean_bill.plot(grid=True, figsize=(13,5),
                        title='Кумулятивный средний чек по группам по дням');
plt.xlabel('Дата');
plt.ylabel('Кумулятивный средний чек');
```



По графику кумулятивного среднего чека также виден выброс в виде очень дорогой покупки 19 августа, который сместил все дальнейшие результаты. В то же время, даже на таком графике видно, что кумулятивный средний чек для группы В начинает снижаться к концу месяца, тогда как группа А постепенно растет. **По графику изменения кумулятивного среднего чека можно сделать вывод, что с учетом выброса группа В лидирует.**

3. Исследование относительного изменения кумулятивного среднего чека группы В к группе А.

Для расчета среднего чека рассчитаем кумулятивную выручку и число заказов, а затем разделим одно на другое.

```
In [38]: mean_bill['rel'] = mean_bill['B'] / mean_bill['A'] - 1
```

```
In [39]: mean_bill.head()
```

```
Out[39]:
```

group	A	B	rel
date			
2019-08-01	6190.791667	4819.857143	-0.221447
2019-08-02	5509.113636	5927.733333	0.075987
2019-08-03	5218.735294	6245.836066	0.196810

group	A	B	rel
date			
2019-08-04	5067.845238	6276.500000	0.238495
2019-08-05	5045.110092	5762.326733	0.142161

```
In [40]: mean_bill['rel'].plot(grid=True, figsize=(13,5),
                                title='Относительный кумулятивный средний чек В по отношению к А по дням');
plt.xlabel('Дата');
plt.ylabel('Относительный кумулятивный средний чек В к А -1');
plt.axhline(y=0.0, color='black', linestyle='--');
```



По графику относительного кумулятивного среднего чека также виден выброс в виде очень дорогой покупки 19 августа, который сместил все дальнейшие результаты. В то же время, показатель группы В при любых условиях после 19 августа уменьшается к концу месяца. **По графику изменения относительного кумулятивного среднего чека можно сделать вывод, что с учетом выброса группа В лидирует, но имеет тенденцию к снижению относительно группы А.**

4. Исследование кумулятивной конверсии по группам

```
In [41]: n_orders_cum.head()
```

```
Out[41]:
```

group	A	B
date		
2019-08-01	24	21
2019-08-02	44	45
2019-08-03	68	61
2019-08-04	84	78
2019-08-05	109	101

```
In [42]: visitors_cum = visitors.pivot_table(
            index='date',
            columns='group',
            values='visitors',
            aggfunc='sum').cumsum(axis=0)
```

```
In [43]: visitors_cum.head()
```

```
Out[43]:
```

group	A	B
date		
2019-08-01	719	713
2019-08-02	1338	1294

group	A	B
date		
2019-08-03	1845	1803
2019-08-04	2562	2573
2019-08-05	3318	3280

```
In [44]: conversion_cum = n_orders_cum.div(visitors_cum)
```

```
In [45]: conversion_cum.plot(grid=True, figsize=(13,5),
                                title='Кумулятивная конверсия по группам по дням',
                                ylim=(0, 0.040)
                                );

plt.xlabel('Дата');
plt.ylabel('Кумулятивная конверсия');
```

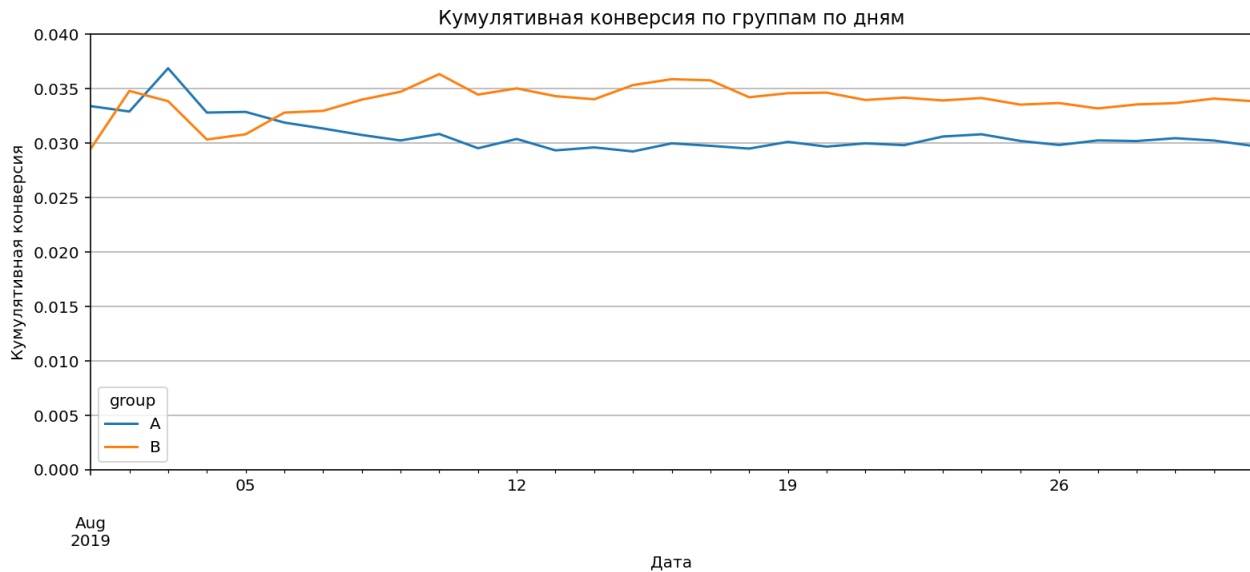


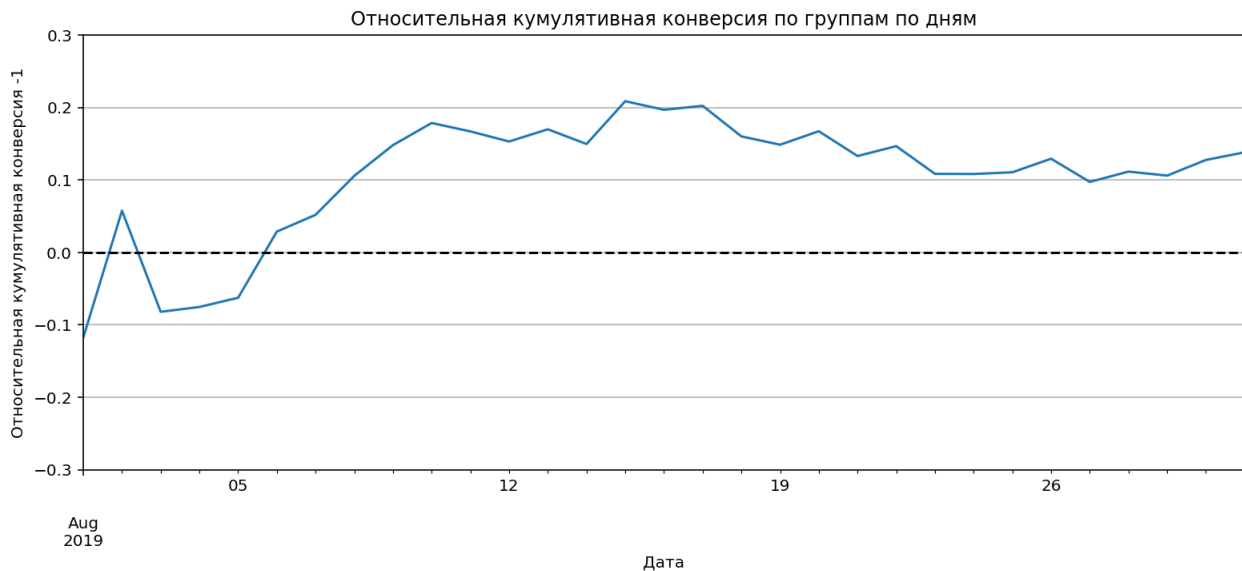
График кумулятивной конверсии для групп установился на стабильных уровнях к концу месяца, причем группа В показала немногом лучший результат по сравнению с группой А (на 0.5 процентных пунктов). Интересно отметить, что группа В опередила группу А уже с 6-7 августа, а крупная покупка 19 августа не оказала действия на конверсию, так как не она не зависит от значений выручки. Зависимости стабилизировались, поэтому можно сказать, что времени для сбора данных для теста достаточно. **При сравнении кумулятивной конверсии группа В показывает лучший результат по сравнению с группой А на 0.5 процентных пункта.**

5. Исследование относительного изменения кумулятивной конверсии группы В к группе А

```
In [46]: conversion_cum['rel'] = conversion_cum['B'] / conversion_cum['A'] - 1
```

```
In [47]: conversion_cum['rel'].plot(grid=True, figsize=(13,5),
                                    title='Относительная кумулятивная конверсия по группам по дням',
                                    ylim=(-0.3, 0.3));

plt.xlabel('Дата');
plt.ylabel('Относительная кумулятивная конверсия -1');
plt.axhline(y=0.0, color='black', linestyle='--');
```



6-7 августа показатель группы В вырвался вперед, но немного снизился после 19 августа, но в последних числах месяца наметилась тенденция на рост. **В целом группа В показывает лучшие результаты, чем группа А, но показатель относительной конверсии все еще колеблется около стабильного значения, что может говорить о том, что требуется больше времени для А/В теста.**

6. Исследование количества заказов по пользователям

Посчитаем из исходных данных количество заказов у каждого пользователя.

```
In [48]: orders.head()
```

```
Out[48]:
```

	transactionId	visitorId	date	revenue	group
0	3667963787	3312258926	2019-08-15	1650	B
1	2804400009	3642806036	2019-08-15	730	B
2	2961555356	4069496402	2019-08-15	400	A
3	3797467345	1196621759	2019-08-15	9759	B
4	2282983706	2322279887	2019-08-15	2308	B

```
In [49]: n_orders_users = orders.groupby('visitorId')['transactionId'].nunique().reset_index()
```

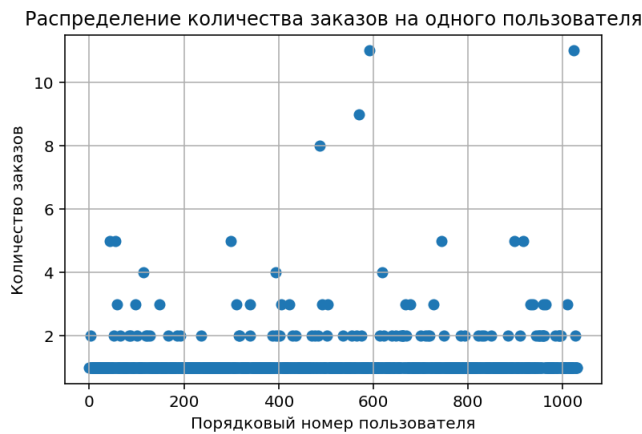
```
In [50]: n_orders_users.head()
```

```
Out[50]:
```

	visitorId	transactionId
0	5114589	1
1	6958315	1
2	8300375	2
3	11685486	1
4	39475350	1

Для наглядной визуализации с помощью точечного графика используем индекс в таблице как значения по оси X.

```
In [51]: plt.scatter(x=n_orders_users.index, y=n_orders_users['transactionId']);
plt.grid(True);
plt.title('Распределение количества заказов на одного пользователя');
plt.ylabel('Количество заказов');
plt.xlabel('Порядковый номер пользователя');
```



По точечному графику видно, что часть пользователей имеют выбросы в виде большого количества заказов, что оказывает влияние на конверсию (в числителе как раз число заказов). **Большинство покупателей совершают около 2 заказов, но некоторые совершают 8 и более.**

7. Поиск аномальных пользователей

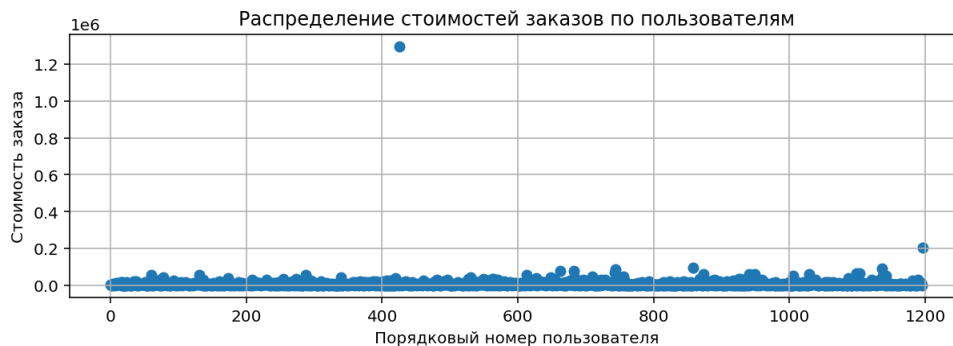
```
In [52]: print(np.percentile(n_orders_users['transactionId'], [95, 99]))
```

[2. 4.]

Не более 5% пользователей делали более 2 заказов и не более 1% делали 4 покупок. **Для устранения выбросов логично выбрать границу в виде 2 заказов для отделения необычных пользователей из набора данных.**

8. Исследование стоимостей заказов

```
In [53]: plt.figure(figsize=(10,3));
plt.scatter(x=orders.index, y=orders['revenue']);
plt.grid(True);
plt.title('Распределение стоимостей заказов по пользователям');
plt.ylabel('Стоимость заказа');
plt.xlabel('Порядковый номер пользователя');
```



По точечному графику сразу стали видны пользователи, которые совершили очень дорогую покупку, очень сильно превышающую обычный уровень ниже 100 тыс. Такие выбросы оказывают влияние на расчеты среднего чека и другие метрики, включающие в себя значения выручки, а значит оказывают влияние на анализ A/B теста. **Большинство покупателей совершают покупки дешевле 100 тыс., но некоторые совершили очень дорогие покупки более 1 млн. Для чистоты анализа A/B теста следует устранить выбросы.**

9. Поиск аномальных заказов

```
In [54]: print(np.percentile(orders['revenue'], [95, 99]))
```

[28000. 58233.2]

Не более 5% заказов совершено на сумму более 28 тыс. и не более 1% на сумму более 58 тыс. **Для устранения выбросов логично выбрать границу в виде суммы выручки не более 28 тыс. для определения аномальных заказов в наборе данных.**

10. Исследование статистической значимости различий в конверсии между группами по «сырым» данным

Для статистического теста Манна-Уитни требуется собрать выборку из всех пользователей, сделавшими, либо не сделавшими заказ в группах. для этого нужно рассчитать количество заказов для каждого покупателя и добавить нули до размера выборки по всем посетителям.

```
In [55]: orders_users_a = orders[orders['group']=='A'].groupby('visitorId')['transactionId'] \
        .nunique().reset_index()
orders_users_b = orders[orders['group']=='B'].groupby('visitorId')['transactionId'] \
        .nunique().reset_index()
```

```
In [56]: orders_users_a.head()
```

```
Out[56]:
```

	visitorId	transactionId
0	8300375	1
1	11685486	1
2	54447517	1
3	66685450	1
4	78758296	1

```
In [57]: orders_users_b.head()
```

```
Out[57]:
```

	visitorId	transactionId
0	5114589	1
1	6958315	1
2	8300375	1
3	39475350	1
4	47206413	1

```
In [58]: visitors.head()
```

```
Out[58]:
```

	date	group	visitors
0	2019-08-01	A	719
1	2019-08-02	A	619
2	2019-08-03	A	507
3	2019-08-04	A	717
4	2019-08-05	A	756

```
In [59]: visitors.groupby('group')['visitors'].sum().reset_index()
```

```
Out[59]:
```

	group	visitors
0	A	18736
1	B	18916

```
In [60]: visitors_a = visitors.query('group == "A"').groupby('group')['visitors'].sum()['A']
```

```
In [61]: visitors_b = visitors.query('group == "B"').groupby('group')['visitors'].sum()['B']
```

```
In [62]: visitors_a
```

```
Out[62]: 18736
```

```
In [63]: visitors_b
```

```
Out[63]: 18916
```

Для каждой группы получили количество посетителей.

```
In [64]: orders_a = orders_users_a[['transactionId']]
```

```
In [65]: orders_a.head()
```

```
Out[65]: transactionId
```

0	1
1	1
2	1
3	1
4	1

```
In [66]: len(orders_a)
```

```
Out[66]: 503
```

Все совершили заказы 503 покупателя из 18736 посетителей для группы A.

```
In [67]: zeros_a = pd.DataFrame(0, index=np.arange(visitors_a - len(orders_a)), columns=orders_a.columns)
```

```
In [68]: zeros_a
```

```
Out[68]: transactionId
```

0	0
1	0
2	0
3	0
4	0
...	...
18228	0
18229	0
18230	0
18231	0
18232	0

18233 rows × 1 columns

Получили список из недостающих нулей для посетителей без покупок. Объединим таблицы.

```
In [69]: sample_orders_a = pd.concat([orders_a, zeros_a], axis=0)
```

```
In [70]: sample_orders_a
```

```
Out[70]: transactionId
```

0	1
1	1
2	1
3	1
4	1
...	...
18228	0
18229	0
18230	0
18231	0
18232	0

18736 rows × 1 columns

```
In [71]: len(sample_orders_a)
```

```
Out[71]: 18736
```

Получили готовый набор данных. Аналогично для группы В.

```
In [72]: orders_b = orders_users_b[['transactionId']]
```

```
In [73]: zeros_b = pd.DataFrame(0, index=np.arange(visitors_b - len(orders_b)), columns=orders_b.columns)
```

```
In [74]: sample_orders_b = pd.concat([orders_b, zeros_b], axis=0)
```

```
In [75]: len(sample_orders_b)
```

```
Out[75]: 18916
```

Количества элементов сошлись, все в порядке.

Для статистически точной оценки различий между наборами данных применяются различные статистические тесты, параметрические (зависят от абсолютных значений величин в выборке, например тест Стьюдента) и непараметрические (сравнивают ранги значений вместо самих значений, например тест Манна-Уитни).

Хотя многие реальные процессы описываются нормальным распределением, количество заказов им не описывается, так как подавляющее большинство пользователей совершают малое число заказов.

Для формальной проверки выборки на нормальность можно применить критерий Шапиро-Уилка. Нулевой гипотезой в таком случае будет ситуация, что выборка нормальная, а альтернативной - что выборка нормальной не является. Критерий значимости примем за **0.05**.

```
In [76]: results_shapiro = shapiro(sample_orders_a)
```

```
c:\users\sergey\appdata\local\programs\python\python38\lib\site-packages\scipy\stats\morestats.py:1681: UserWarning:
p-value may not be accurate for N > 5000.
  warnings.warn("p-value may not be accurate for N > 5000.")
```

```
In [77]: results_shapiro[1]
```

```
Out[77]: 0.0
```

Получившийся $pvalue = 0$ показывает, что нулевую гипотезу о том, что распределение выборки нормальное, нужно отбросить. Таким образом тест Стьюдента здесь неприменим, поэтому используем непараметрический тест Манна-Уитни.

В нашем случае нулевой гипотезой будет предположение, что выборки структурно равны, а альтернативной - что различны. Выберем **критический уровень значимости 0.05** и установим параметр альтернативной гипотезы **two-sided** для расчета различий в целом.

```
In [78]: pvalue = mannwhitneyu(sample_orders_a, sample_orders_b, alternative='two-sided')[1]
```

```
In [79]: pvalue
```

```
Out[79]: 0.016792355056752608
```

Для сравнения конверсий достаточно сравнить среднее арифметическое по выборкам, так как сумма по столбцу в получившейся таблице дает количество заказов, а количество строк - число посетителей.

```
In [80]: rel_conversion = sample_orders_b.mean()[0]/sample_orders_a.mean()[0] - 1
```

```
In [81]: rel_conversion
```

```
Out[81]: 0.13807884655320146
```

В результате с выбранным критерием значимости 0.05 получившийся $pvalue = 0.016$ его не превышает, что позволяет сделать вывод, что различия в выборках есть и они статистически значимы. Группа В имеет конверсию на 14% выше, чем группа А по сырым данным.

11. Исследование статистической значимости различий в среднем чеке заказа между группами по «сырым» данным

Для подготовки наборов данных для теста достаточно выбрать значения выручки для групп.

```
In [82]: sample_revenue_a = orders.query('group == "A"')['revenue']
```

```
In [83]: sample_revenue_b = orders.query('group == "B"')['revenue']
```

```
In [84]: sample_revenue_a.head()
```

```
Out[84]: 2      400
         7     1044
         8    13710
        10     4008
        12     7370
         Name: revenue, dtype: int64
```

```
In [85]: sample_revenue_b.head()
```

```
Out[85]: 0      1650
         1       730
         3     9759
         4     2308
         5     2210
         Name: revenue, dtype: int64
```

Выберем критерий значимости также 0.05.

```
In [86]: pvalue_revenue = mannwhitneyu(sample_revenue_a, sample_revenue_b)[1]
```

```
In [87]: pvalue_revenue
```

```
Out[87]: 0.3646454927716229
```

Значение *pvalue* превышает выбранный критерий, поэтому статистически значимой разницы в выручке по группам нет.

Проверим также относительную разницу среднего чека.

```
In [88]: sample_revenue_b.mean()/sample_revenue_a.mean()-1
```

```
Out[88]: 0.2587136699126005
```

Так как тест Манна-Уитни непараметрический, то ему безразличны выбросы значений в выборках, поэтому на "грязных" данных с выбросами он дает верный результат, даже если сравнение средних метрик дает противоположный результат.

Как видно, средний чек группы B превышает средний чек группы A на 26%, но статистической разницы между выборками нет.

12. Исследование статистической значимости различий в конверсии между группами по «очищенным» данным

Для очистки выборки от аномальных пользователей выберем критерии **не более 2 заказов и выручка не более 28 тыс.**

Вычленим идентификаторы таких пользователей из каждой группы и объединим в один столбец.

```
In [89]: users_many_orders = pd.concat([orders_users_a.query('transactionId > 2')['visitorId'],
                                     orders_users_b.query('transactionId > 2')['visitorId']],
                                     axis=0)
```

```
In [90]: len(users_many_orders)
```

```
Out[90]: 24
```

Таких покупателей всего 24.

```
In [91]: users_expensive_orders = orders.query('revenue > 28000')['visitorId']
```

```
In [92]: len(users_expensive_orders)
```

```
Out[92]: 60
```

Пользователей с дорогими покупками всего 60 из всего числа. Объединим таких пользователей в один столбец и удалим возможные дубликаты из-за пересечений множеств.

```
In [93]: abnormal_users = pd.concat([users_many_orders, users_expensive_orders], axis=0).drop_duplicates()
```



```
In [94]: abnormal_users.head()
```

```
Out[94]: 18      199603092
        23      237748145
        68      611059232
        146     1230306981
        189     1614305549
        Name: visitorId, dtype: int64
```

```
In [95]: len(abnormal_users)
```

```
Out[95]: 74
```

В результате получили 74 идентификатора пользователей, который совершили необычные покупки.

Подготовим новые выборки для теста Манна-Уитни.

```
In [96]: orders_users_a_filtered = orders_users_a[~orders_users_a['visitorId'].isin(abnormal_users)]
```

```
In [97]: orders_users_b_filtered = orders_users_b[~orders_users_b['visitorId'].isin(abnormal_users)]
```

```
In [98]: zeros_a_filtered = pd.DataFrame(0, index=np.arange(visitors_a - len(orders_users_a_filtered['transactionId'])),
        columns=['transactionId'])
```

```
In [99]: zeros_b_filtered = pd.DataFrame(0, index=np.arange(visitors_b - len(orders_users_b_filtered['transactionId'])),
        columns=['transactionId'])
```

```
In [100... sample_orders_a_filtered = pd.concat([orders_users_a_filtered[['transactionId']], zeros_a_filtered], axis=0)
```

```
In [101... sample_orders_b_filtered = pd.concat([orders_users_b_filtered[['transactionId']], zeros_b_filtered], axis=0)
```

Для расчета статистической значимости также применим критерий Манна-Уитни с выбранным **критерием значимости 0.05** и двусторонней гипотезой. Нулевая гипотеза состоит в том, что структурной разницы в выборках нет, а альтернативная - что есть.

```
In [102... pvalue_filtered = mannwhitneyu(sample_orders_a_filtered, sample_orders_b_filtered,
        alternative='two-sided')[1]
```

```
In [103... pvalue_filtered
```

```
Out[103... 0.013063114767059361
```

```
In [104... rel_conversion_filtered = sample_orders_b_filtered.mean()[0]/sample_orders_a_filtered.mean()[0] - 1
```

```
In [105... rel_conversion_filtered
```

```
Out[105... 0.17266633792572206
```

В отфильтрованном от аномальных покупателей наборах *pvalue* = 0.013 стал еще ниже и все также меньше выбранного критического значени 0.05, а значит нулевую гипотезу нужно отбросить. Это позволяет сделать вывод, что различия в выборках есть и они статистически значимы также и в отфильтрованных данных. Конверсия для группы В даже увеличилась и стала на 17% выше, чем в группе А.

13. Исследование статистической значимости различий в среднем чеке заказа между группами по «очищенным» данным

```
In [106... sample_revenue_a_filtered = orders[~orders['visitorId'].isin(abnormal_users)].query('group == "A"')['revenue']
```

```
In [107... sample_revenue_b_filtered = orders[~orders['visitorId'].isin(abnormal_users)].query('group == "B"')['revenue']
```

Для расчета статистической значимости также применим критерий Манна-Уитни с выбранным **критерием значимости 0.05** и двусторонней гипотезой. Нулевая гипотеза состоит в том, что структурной разницы в выборках нет, а альтернативная - что есть.

```
In [108... pvalue_revenue_filtered = mannwhitneyu(sample_revenue_a_filtered, sample_revenue_b_filtered,
        alternative='two-sided')[1]
```

```
In [109... pvalue_revenue_filtered
```

```
0.738481751314565
```

Out[109...

Значение *pvalue* превышает выбранный критерий, поэтому нулевую гипотезу нельзя отбросить и значит статистически значимой разницы в выручке по группам нет.

Проверим также относительную разницу среднего чека.

```
In [110... sample_revenue_b_filtered.mean()/sample_revenue_a_filtered.mean()-1
```

Out[110... -0.019624288331982598

В отфильтрованных от аномальных пользователей выборках статистической разницы все также нет (*pvalue* = 0.74). В то же время, так как мы избавились от выбросов, разница в средних чеках стала всего лишь 2%, причем группа В хуже группы А.

14. Принимаем решение по результатам теста

В процессе анализа результатов А/В теста обнаружили следующие факты:

- Есть статистически значимые различия по конверсии между группами по «сырым» и отфильтрованным от аномалий данным, группа В показывает в среднем на 17% лучшую конверсию, чем группа А;
- Нет статистически значимого различия по среднему чеку между группами ни по «сырым» данным, ни по данным после фильтрации аномалий;
- График различия конверсии между группами сообщает, что результаты группы В практически зафиксировались на уровне на 17% лучше, чем у группы А;
- График различия среднего чека говорит о том, что результаты группы В ухудшаются к концу теста и сейчас почти на 2% хуже группы А, тенденция отрицательная.

Исходя из обнаруженных фактов, тест следует остановить и признать, что сегмент В показывает лучшие результаты по конверсии, чем сегмент А, но хуже, чем сегмент А по среднему чеку. Продолжать смысла нет, так как основные кумулятивные метрики выравнивались и наметились тенденции.

Часть 3. Общий вывод и рекомендации

В результате работы было сделано следующее:

- сделан рейтинг гипотез по методиками ICE и RICE
- проанализированы результаты А/В теста, зафиксирована победа сегмента В по конверсии, но неудача по части среднего чека.

Как итог, можно предположить, что результаты А/В теста можно использовать в случае, если компании важно увеличить конверсию. Но, по-видимому, изменения, примененные в сегменте В вызывают приток новых покупателей, которые не делают большие заказы, а наоборот снижают метрику среднего чека. В совокупности увеличение конверсии приводит к увеличению общей выручки компании, несмотря на незначительное уменьшение среднего чека.

Так как в изначальной задаче исследования предполагалась цель по увеличению общей выручки в конечном итоге, посмотрим на распределение кумулятивной выручки по группам без аномальных покупателей.

```
In [111... orders_filtered = orders[~orders['visitorId'].isin(abnormal_users)]
```

```
In [112... orders_filtered.pivot_table(
    index='date',
    columns='group',
    values='revenue',
    aggfunc='sum').cumsum(axis=0).plot(grid=True,
                                     figsize=(13,5),
                                     title='Кумулятивная выручка по группам по дням без аномальных пользователей'
    );
plt.xlabel('Дата');
plt.ylabel('Кумулятивная выручка');
```



```
In [113... orders_filtered.groupby('group')['revenue'].sum().reset_index()
```

```
Out[113...
group revenue
0      A  2443806
1      B  2836522
```

Как видно по графику и общей выручке за месяц, несмотря на незначительную разницу в средних чеках, увеличенная конверсия сегмента В дала большой прирост в общей выручке за время А/В теста. **В сумме сегмент В заработал почти на 400 тыс. больше (14%), чем сегмент А.** Этот факт позволяет заявить об успешности А/В теста и выигрыше группы В по основной цели исследования, направленной на увеличение общей выручки.