Описание проекта

В ходе проекта получен подготовленный список гипотез для увеличения выручки. Нужно приоритизировать гипотезы, запустить A/B-тест и проанализировать результаты. Проект состоит из 2-х частей.

В первой части нужно приоритизировать гипотезы по увеличению выручки интернет-магазина с применением фреймворков ICE и RICE. Указать, как изменилась приоритизация гипотез при применении RICE вместо ICE и почему.

Во второй части нужно провести анализ A/B-теста. Для этого: Построить графики: кумулятивной выручки по группам, кумулятивного среднего чека по группам, относительного изменения кумулятивного среднего чека группы B к группе A, кумулятивной конверсии по группам, относительного изменения кумулятивной конверсии группы B к группе A, график количества заказов по пользователям, график стоимостей заказов. Посчитать: 95-й и 99-й перцентили количества заказов на пользователя. Выбрать границу для определения аномальных пользователей, 95-й и 99-й перцентили стоимости заказов. Выбрать границу для определения аномальных заказов, статистическую значимость различий в конверсии между группами по «сырым» данным, статистическую значимость различий в среднем чеке заказа между группами по «сырым» данным, статистическую значимость различий в конверсии между группами по «очищенным» данным, статистическую значимость различий в среднем чеке заказа между группами по «очищенным» данным.

```
На каждом этапе сделать выводы и предположения.
Принять решение по результатам теста и объяснить его.
Варианты решений: 1. Остановить тест, зафиксировать победу одной из групп.
                  2. Остановить тест, зафиксировать отсутствие различий между
группами.
                  3. Продолжить тест.
```

# Принятие решений в бизнесе на основе данных.

## Загрузка данных и подготовка их к анализу

Загрузим данные о визитах, заказах и гипотезах в переменные. Оптимизируем данные для анализа. Убедимся, что тип данных в каждой колонке — правильный. Путь к файлам:

```
/datasets/hypothesis.csv.
/datasets/orders.csv.
/datasets/visitors.csv.
```

### Загрузка данных

In [1]:

```
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
from matplotlib import pyplot as plt
import scipy.stats as stats
from pandas.plotting import register_matplotlib_converters
%config InlineBackend.figure_format = 'retina'
import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_colwidth',1000)
```

In [2]:
```
hypothes, orders, visits = (
                                pd.read_csv('/datasets/hypothesis.csv'),# журнал гипотез
                                pd.read_csv('/datasets/orders.csv'),    # журнал покупок
                                pd.read_csv('/datasets/visitors.csv')) # журнал посещений
```

## Предобработка данных

In [3]:
```
#получаем информацию
hypothes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Hypothesis  9 non-null      object
 1   Reach       9 non-null      int64 
 2   Impact      9 non-null      int64 
 3   Confidence  9 non-null      int64 
 4   Efforts     9 non-null      int64 
dtypes: int64(4), object(1)
memory usage: 488.0+ bytes
```

In [4]:
```
#получаем информацию
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 [5]:
```
#получаем информацию
visits.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
```

Видим, что пропусков нет.Но время указано в формате object и названия столбцов указаны неправильно. Исправим.

In [6]:
```
# приведем названия столбцов к нижнему регистру и исправим названия:
hypothes.columns = hypothes.columns.str.lower().str.replace(' ', '_')
```

In [7]:
```
hypothes.columns# проверка результатов - перечень названий столбцов
```

Out[7]:
```
Index(['hypothesis', 'reach', 'impact', 'confidence', 'efforts'], dtype='object')
```

In [8]:
```
# приведем названия столбцов к нижнему регистру и исправим названия:
```

```
visits.columns = visits.columns.str.lower().str.replace(' ', '_')
```

In [9]:
```
visits.columns# проверка результатов - перечень названий столбцов
```

Out[9]:
```
Index(['date', 'group', 'visitors'], dtype='object')
```

In [10]:
```
# приведем названия столбцов к нижнему регистру и исправим названия:
orders = orders.rename(columns={'transactionId':'transaction_Id', 'visitorId':'visitor_Id'})
orders.columns = map(str.lower, orders.columns)
```

In [11]:
```
orders.columns# проверка результатов - перечень названий столбцов
```

Out[11]:
```
Index(['transaction_id', 'visitor_id', 'date', 'revenue', 'group'], dtype='object')
```

In [12]:
```
# проверим на дубликаты
hypothes.duplicated().sum()
```

Out[12]:
```
0
```

In [13]:
```
# проверим на дубликаты
visits.duplicated().sum()
```

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

In [14]:
```
# проверим на дубликаты
orders.duplicated().sum()
```

Out[14]:
```
0
```

In [15]:
```
# преобразование данных о времени
visits['date'] = pd.to_datetime(visits['date'])
orders['date'] = pd.to_datetime(orders['date'])
```

В процессе предобработки данных были выявлены следующие ошибки: время указано в формате object, названия столбцов указаны неправильно. Привели названия столбцов к нижнему регистру и перемеиновали. Преобразовали данные о времени. Проверили на дубликаты. Дубликатов не обнаружено.

### Проверим количество пользователей в группах.

In [16]:
```
data_new = (
    visits.groupby('group', as_index=False)
    .agg({'visitors': 'sum'})
)

print(data_new)
```

```
  group  visitors
0     A     18736
1     B     18916
```

Группы равны, можно проводить тест.

### Проверим,есть ли пользователи, которые попали в две группы теста одновременно.

In [17]:
```
visitors_2 = orders.groupby('visitor_id').agg({'group':'nunique'}).query('group > 1')#Пользовал
visitors_2.count()
```

Out[17]:
```
group    58
dtype: int64
```

58 пользователей находятся в двух группах одновременно. Это меньше 1% пользователей, можно пренебречь.

### Начало и конец теста

In [18]:
```
visits['date'].min()
```

Out[18]:
```
Timestamp('2019-08-01 00:00:00')
```

In [19]:
```
visits['date'].max()
```

Out[19]:
```
Timestamp('2019-08-31 00:00:00')
```

# Приоритизация гипотез.

In [20]:
```
#В файле /datasets/hypothesis.csv 9 гипотез по увеличению выручки интернет-магазина с указанным
hypothes.style
```

Out[20]:

| | 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 |

In [21]:
```
#Применим фреймворк ICE для приоритизации гипотез. Отсортируем их по убыванию приоритета.
hypothes['ICE'] = hypothes['impact'] * hypothes['confidence'] / hypothes['efforts']
```

In [22]:
```
print(hypothes[['ICE']].sort_values(by='ICE', ascending=False).round(3))
```

```
      ICE
8  16.200
0  13.333
7  11.200
6   8.000
2   7.000
```

```
1   2.000
5   1.333
3   1.125
4   1.000
```

В приоритете 8, 0 и 7 гипотезы: Запустить акцию, дающую скидку на товар в день рождения. Добавить два новых канала привлечения трафика, что позволит привлекать на 30% больше пользователей. Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок

In [23]:
```
# Применим фреймворк RICE для приоритизации гипотез. Отсортируем их по убыванию приоритета.
hypothes['RICE'] = hypothes['reach'] * hypothes['impact'] * hypothes['confidence'] / hypothes['
```

In [24]:
```
print(hypothes.sort_values(by='RICE', ascending=False)[['RICE']])
```

```
    RICE
7  112.0
2   56.0
0   40.0
6   40.0
8   16.2
3    9.0
1    4.0
5    4.0
4    3.0
```

In [25]:
```
print(hypothes[['ICE']].sort_values(by='ICE', ascending=False).round(3))
print(hypothes.sort_values(by='RICE', ascending=False)[['RICE']])
```

```
      ICE
8  16.200
0  13.333
7  11.200
6   8.000
2   7.000
1   2.000
5   1.333
3   1.125
4   1.000
    RICE
7  112.0
2   56.0
0   40.0
6   40.0
8   16.2
3    9.0
1    4.0
5    4.0
4    3.0
```

С большим отрывом 7 гипотеза. Следом 2, 0 и 6: Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок. Добавить блоки рекомендаций товаров на сайт интернет магазина, чтобы повысить конверсию и средний чек заказа. Добавить два новых канала привлечения трафика, что позволит привлекать на 30% больше пользователей. Показать на главной странице баннеры с актуальными акциями и распродажами, чтобы увеличить конверсию.

Приоритизация гипотез при применении RICE вместо ICE изменилась. С большим отрывом идет 7 гипотеза - "Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок".Так получилось потому, что ее параметр reach равен 10 - больше, чем у других гипотез.У 8 гипотезы -"Запустить акцию, дающую скидку на товар в день рождения" параметр reach равен 1,поэтому она ушла на 5 место.Охват аудитории имеет огромное значение.

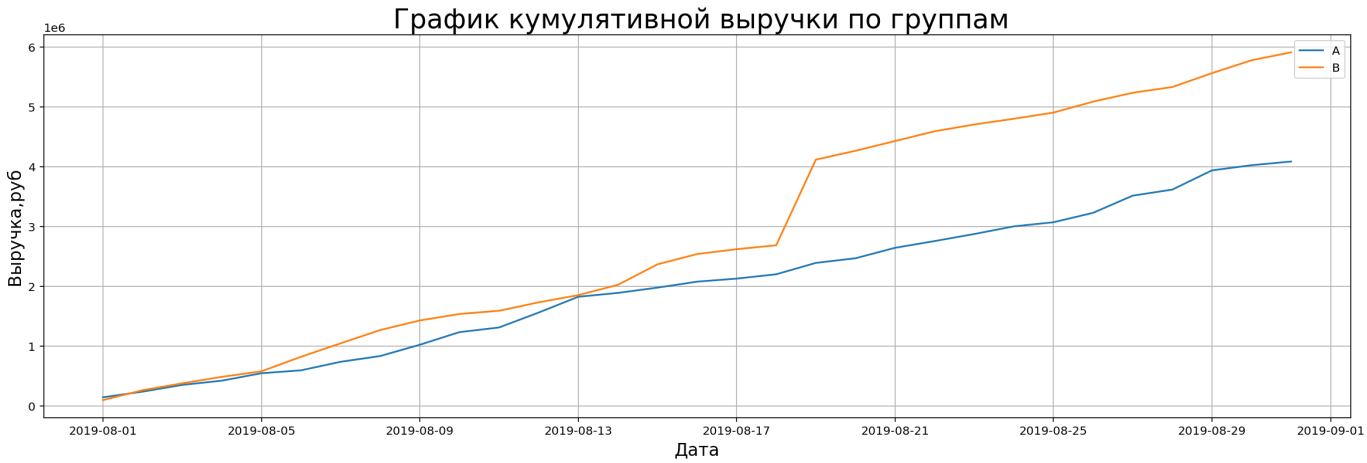
# Анализ A/B-теста

Результаты A/B-теста описаны в файлах /datasets/orders.csv и /datasets/visitors.csv.

## График кумулятивной выручки по группам

In [26]:

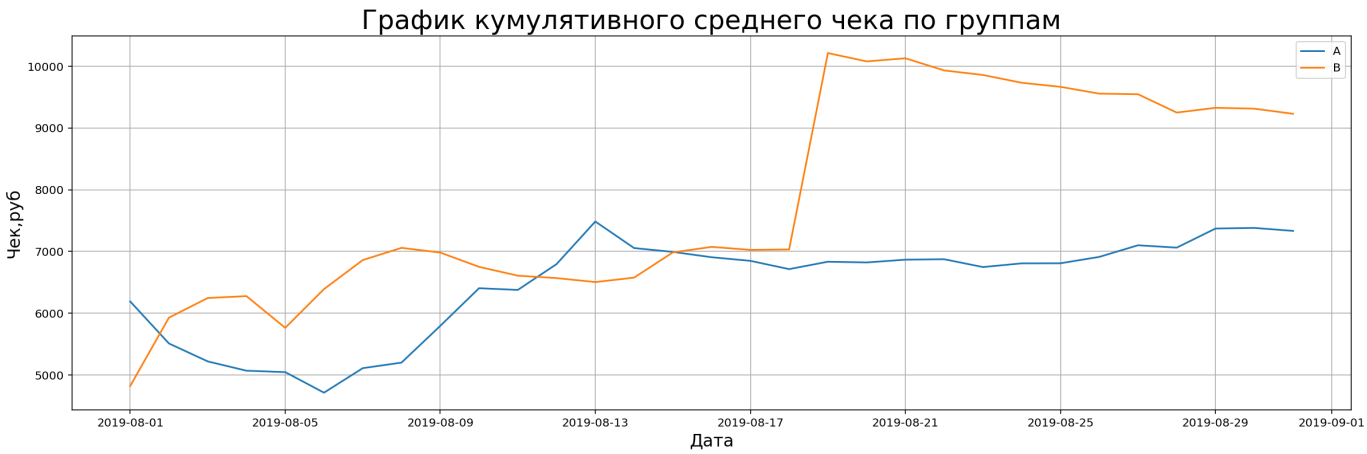
```
datesGroups = orders[['date', 'group']].drop_duplicates()
ordersAggregated = datesGroups.apply(
    lambda x: orders[
        np.logical_and(
            orders['date'] <= x['date'], orders['group'] == x['group']
        )
    ].agg(
        {
            'date': 'max',
            'group': 'max',
            'transaction_id': 'nunique',
            'visitor_id': 'nunique',
            'revenue': 'sum',
        }
    ),
    axis=1,
).sort_values(by=['date', 'group'])

visitorsAggregated = datesGroups.apply(
    lambda x: visits[
        np.logical_and(
            visits['date'] <= x['date'], visits['group'] == x['group']
        )
    ].agg({'date': 'max', 'group': 'max', 'visitors': 'sum'}),
    axis=1,
).sort_values(by=['date', 'group'])

cumulativeData = ordersAggregated.merge(
    visitorsAggregated, left_on=['date', 'group'], right_on=['date', 'group']
)
cumulativeData.columns = [
    'date',
    'group',
    'orders',
    'buyers',
    'revenue',
    'visitors',
]
cumulativeRevenueA = cumulativeData[cumulativeData['group']=='A'][['date','revenue', 'orders']]
cumulativeRevenueB = cumulativeData[cumulativeData['group']=='B'][['date','revenue', 'orders']]
plt.figure(figsize = (20,6))
plt.plot(cumulativeRevenueA['date'], cumulativeRevenueA['revenue'], label='A')
plt.plot(cumulativeRevenueB['date'], cumulativeRevenueB['revenue'], label='B')
#plt.annotate('скачок', xy=('2019-08-17', 2800000), xytext=('2019-08-10', 3500000),
#             arrowprops=dict(facecolor='g'))
#plt.annotate('отрыв', xy=('2019-08-05', 800000), xytext=('2019-08-05', 2000000),
#             arrowprops=dict(facecolor='g'))
plt.title(' График кумулятивной выручки по группам',size = 23)
plt.xlabel('Дата',size =15)
plt.ylabel('Выручка,руб',size =15)
plt.grid()
plt.legend()
plt.show()
```




В середине теста сегмент B вырвался вперед и продолжил лидировать до конца теста.Похоже, что это аномально большие заказы влияют на результаты.Впоследствии их нужно удалить.

## График кумулятивного среднего чека по группам

In [27]:
```
plt.figure(figsize = (20,6))
plt.plot(cumulativeRevenueA['date'], cumulativeRevenueA['revenue']/cumulativeRevenueA['orders']
plt.plot(cumulativeRevenueB['date'], cumulativeRevenueB['revenue']/cumulativeRevenueB['orders']
#plt.annotate('скачок', xy=('2019-08-18', 7000),xytext=('2019-08-10', 9000),
#             arrowprops=dict(facecolor='g'))
plt.title('График кумулятивного среднего чека по группам',size = 23)
plt.xlabel('Дата',size =15)
plt.ylabel('Чек,руб',size =15)
plt.legend()
plt.grid()
plt.show()
```




Кумулятивное значение среднего чека по сегментам продолжает колебаться. Принимать решение по этой метрике нельзя. Или придется анализировать выбросы, сильно влияющие на результаты.

## График относительного изменения кумулятивного среднего чека группы B к группе A.

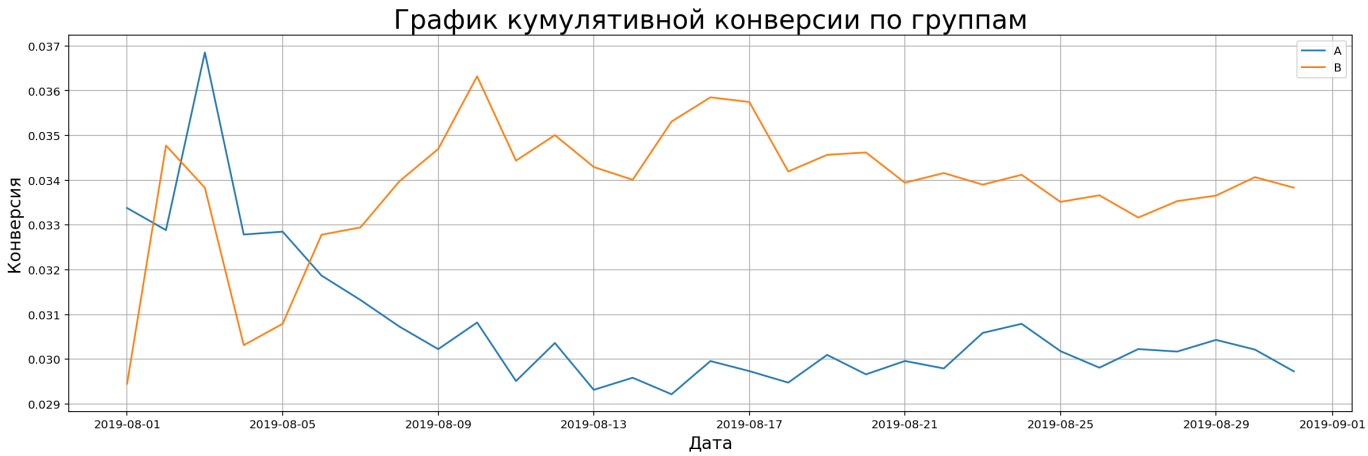
In [28]:
```
plt.figure(figsize = (20,6))
mergedCumulativeRevenue = cumulativeRevenueA.merge(cumulativeRevenueB, left_on='date', right_or
plt.plot(mergedCumulativeRevenue['date'], (mergedCumulativeRevenue['revenueB']/mergedCumulative
plt.title('График относительного изменения кумулятивного среднего чека группы B к группе A',siz
plt.xlabel('Дата',size =15)
plt.ylabel('Отношение B к A',size =15)
plt.axhline(y=0, color='black', linestyle='--')
plt.grid()
plt.show()
```


График относительного различия в среднем чеке между группами. Результаты теста значительно и резко менялись несколько раз по датам.Видимо,именно тогда были совершены аномальные выбросы.

## График кумулятивной конверсии по группам

In [29]:

```
plt.figure(figsize = (20,6))
cumulativeData['conversion'] = cumulativeData['orders']/cumulativeData['visitors']
cumulativeDataA = cumulativeData[cumulativeData['group']=='A']
cumulativeDataB = cumulativeData[cumulativeData['group']=='B']
plt.plot(cumulativeDataA['date'], cumulativeDataA['conversion'], label='A')
plt.plot(cumulativeDataB['date'], cumulativeDataB['conversion'], label='B')
plt.title('График кумулятивной конверсии по группам',size = 23)
plt.xlabel('Дата',size =15)
plt.ylabel('Конверсия',size =15)
plt.legend()
#plt.annotate('отрыв', xy=('2019-08-06', 0.033), xytext=('2019-08-05', 0.035),
#             arrowprops=dict(facecolor='g'))
#plt.axis(figsize = (0,1))
plt.ylim()
plt.grid()
plt.show()
```




В начале сегмент A имел большую конверсию, но затем сегмент B выравнялся и зафиксировал большее значение относительно сегмента A.

## График относительного изменения кумулятивной конверсии группы B к группе A.

In [30]:

```
cumulativeData['conversion'] = (
    cumulativeData['orders'] / cumulativeData['visitors']
)
plt.figure(figsize = (20,6))
cumulativeDataA = cumulativeData[cumulativeData['group'] == 'A']
cumulativeDataB = cumulativeData[cumulativeData['group'] == 'B']
mergedCumulativeConversions = cumulativeDataA[['date','conversion']].merge(cumulativeDataB[['da
```

```
plt.plot(mergedCumulativeConversions['date'], mergedCumulativeConversions['conversionB']/mergec
plt.title('График относительного изменения кумулятивной конверсии группы B к группе A.',size =
plt.xlabel('Дата',size =15)
plt.ylabel('Отношение B к A',size =15)
plt.axhline(y=0, color='black', linestyle='--')
plt.axhline(y=0.2, color='grey', linestyle='--')
plt.grid()
plt.show()
```

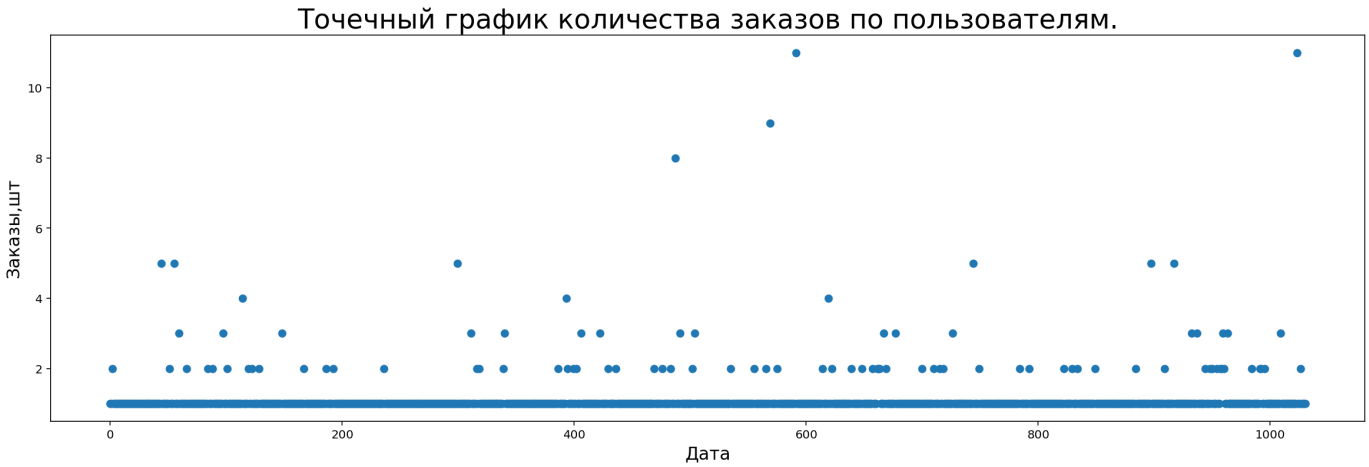

График относительного различия конверсии между группами. Почти с самого начала теста группа B лидирует по конверсии. Был спад в начале, снижение в середине теста. К концу теста наметился подъем до показателя 13% относительно группы A.

## Точечный график количества заказов по пользователям.

In [31]:

```
plt.figure(figsize = (20,6))
ordersByUsers = (
    orders.groupby('visitor_id', as_index=False)
    .agg({'transaction_id': pd.Series.nunique})
)
ordersByUsers.columns = ['visitor_id', 'orders']
print(ordersByUsers.sort_values(by='orders', ascending=False).head(10))
plt.title('Точечный график количества заказов по пользователям.',size = 23)
plt.xlabel('Дата',size =15)
plt.ylabel('Заказы,шт',size =15)
x_values = pd.Series(range(0, len(ordersByUsers)))
plt.scatter(x_values, ordersByUsers['orders'])
plt.show()
```

```
      visitor_id  orders
1023  4256040402      11
591   2458001652      11
569   2378935119       9
487   2038680547       8
44     199603092       5
744   3062433592       5
55     237748145       5
917   3803269165       5
299   1230306981       5
897   3717692402       5
```




Пользователей, заказавших более 2 раз,немного. А заказавших более 3 раз совсем мало. Они вполне могут быть аномальными.

## Подсчет 95-й и 99-й перцентили количества заказов на пользователя.

In [32]:
```
print(ordersByUsers.sort_values(by='orders', ascending=False).head(10))
print(np.percentile(ordersByUsers['orders'], [90, 95, 99]))
```

```
      visitor_id  orders
1023  4256040402      11
591   2458001652      11
569   2378935119       9
487   2038680547       8
44     199603092       5
744   3062433592       5
55     237748145       5
917   3803269165       5
299   1230306981       5
897   3717692402       5
[1. 2. 4.]
```

Не более 5% пользователей совершили 2 заказа, и не более 1% - более 4 заказов.

## Точечный график стоимостей заказов

In [33]:
```
print(orders.sort_values(by='revenue', ascending=False).head(10))
plt.figure(figsize = (20,6))
x_values = pd.Series(range(0, len(orders['revenue'])))
plt.scatter(x_values, orders['revenue'])
plt.title('Точечный график стоимостей заказов.',size = 23)
plt.xlabel('Дата',size =15)
plt.ylabel('Заказы,руб',size =15)
plt.show()
```

```
      transaction_id  visitor_id       date  revenue group
425        590470918  1920142716 2019-08-19  1294500     B
1196      3936777065  2108080724 2019-08-15   202740     B
858        192721366  1316129916 2019-08-27    93940     A
1136       666610489  1307669133 2019-08-13    92550     A
744       3668308183   888512513 2019-08-27    86620     B
682       1216533772  4266935830 2019-08-29    78990     B
662       1811671147  4266935830 2019-08-29    78990     A
743       3603576309  4133034833 2019-08-09    67990     A
1103      1348774318  1164614297 2019-08-12    66350     A
1099       316924019   148427295 2019-08-12    65710     A
```

Точечный график стоимостей заказов.


Заказов болеее 100000 очень мало. Есть выбросы в районе 1200000. Это аномалии.

## Посдчет 95-й и 99-й перцентили стоимости заказов

In [34]:
```
print(orders.head(10))
print(np.percentile(orders['revenue'], [90, 95, 99]))
```

```
   transaction_id  visitor_id       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
5       182168103   935554773 2019-08-15     2210     B
6       398296753  2900797465 2019-08-15     1860     B
7      2626614568    78758296 2019-08-15     1044     A
8      1576988021   295230930 2019-08-15    13710     A
9      1506739906  1882260405 2019-08-15     1855     B
[18168.  28000.  58233.2]
```

Не более 5% чеков дороже 28000 и не более 1% - дороже 58233.

Построим гистрограмму распределения количества заказов на одного пользователя.

In [35]:
```
plt.figure(figsize = (20,6))
plt.hist(ordersByUsers['orders'])
plt.title('Гистограмма распределения количества заказов на одного пользователя.',size = 23)
plt.xlabel('Заказы,шт',size =15)
plt.ylabel('Пользователи,чел',size =15)
plt.show()
```

Гистограмма распределения количества заказов на одного пользователя.


Большинство покупателей заказывали только один раз. Однако доля пользователей с 2-4 заказами тоже значительна.

Посчитаем выборочные перцентили количества заказов на одного пользователя:

In [36]:

```
print(np.percentile(ordersByUsers['orders'], [90, 95, 99]))
```

```
[1. 2. 4.]
```

Нижнюю границу для определения аномальных заказов установим в 4 заказа на одного пользователя.

И отсеем аномальных пользователей по ней.

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

Сформулируем гипотезы:

1.Нулевая гипотеза: Статистически значимых различий в конверсии между группами нет.

2.Обратная гипотеза:Статистически значимые различия в конверсии между группами есть.

In [48]:

```
visitorsADaily = visits[visits['group'] == 'A'][['date', 'visitors']]
visitorsADaily.columns = ['date', 'visitorsPerDateA']

visitorsACummulative = visitorsADaily.apply(
    lambda x: visitorsADaily[visitorsADaily['date'] <= x['date']].agg(
        {'date': 'max', 'visitorsPerDateA': 'sum'}
    ),
    axis=1,
)
visitorsACummulative.columns = ['date', 'visitorsCummulativeA']

visitorsBDaily = visits[visits['group'] == 'B'][['date', 'visitors']]
visitorsBDaily.columns = ['date', 'visitorsPerDateB']

visitorsBCummulative = visitorsBDaily.apply(
    lambda x: visitorsBDaily[visitorsBDaily['date'] <= x['date']].agg(
        {'date': 'max', 'visitorsPerDateB': 'sum'}
    ),
    axis=1,
)
visitorsBCummulative.columns = ['date', 'visitorsCummulativeB']

ordersADaily = (
    orders[orders['group'] == 'A'][['date', 'transaction_id', 'visitor_id', 'revenue']]
    .groupby('date', as_index=False)
    .agg({'transaction_id': pd.Series.nunique, 'revenue': 'sum'})
)
ordersADaily.columns = ['date', 'ordersPerDateA', 'revenuePerDateA']

ordersACummulative = ordersADaily.apply(
    lambda x: ordersADaily[ordersADaily['date'] <= x['date']].agg(
        {'date': 'max', 'ordersPerDateA': 'sum', 'revenuePerDateA': 'sum'}
    ),
    axis=1,
).sort_values(by=['date'])
ordersACummulative.columns = [
    'date',
    'ordersCummulativeA',
    'revenueCummulativeA',
]

ordersBDaily = (
    orders[orders['group'] == 'B'][['date', 'transaction_id', 'visitor_id', 'revenue']]
    .groupby('date', as_index=False)
    .agg({'transaction_id': pd.Series.nunique, 'revenue': 'sum'})
)
ordersBDaily.columns = ['date', 'ordersPerDateB', 'revenuePerDateB']

ordersBCummulative = ordersBDaily.apply(
    lambda x: ordersBDaily[ordersBDaily['date'] <= x['date']].agg(
        {'date': 'max', 'ordersPerDateB': 'sum', 'revenuePerDateB': 'sum'}
```

```
    ),
    axis=1,
).sort_values(by=['date'])
ordersBCummulative.columns = [
    'date',
    'ordersCummulativeB',
    'revenueCummulativeB',
]

data = (
    ordersADaily.merge(
        ordersBDaily, left_on='date', right_on='date', how='left'
    )
    .merge(ordersACummulative, left_on='date', right_on='date', how='left')
    .merge(ordersBCummulative, left_on='date', right_on='date', how='left')
    .merge(visitorsADaily, left_on='date', right_on='date', how='left')
    .merge(visitorsBDaily, left_on='date', right_on='date', how='left')
    .merge(visitorsACummulative, left_on='date', right_on='date', how='left')
    .merge(visitorsBCummulative, left_on='date', right_on='date', how='left')
)
data

ordersByUsersA = (
    orders[orders['group'] == 'A']
    .groupby('visitor_id', as_index=False)
    .agg({'transaction_id': pd.Series.nunique})
)
ordersByUsersA.columns = ['visitor_id', 'orders']
ordersByUsersB = (
    orders[orders['group'] == 'B']
    .groupby('visitor_id', as_index=False)
    .agg({'transaction_id': pd.Series.nunique})
)
ordersByUsersB.columns = ['visitor_id', 'orders']
sampleA = pd.concat([ordersByUsersA['orders'],pd.Series(0, index=np.arange(data['visitorsPerDat
sampleB = pd.concat([ordersByUsersB['orders'],pd.Series(0, index=np.arange(data['visitorsPerDat
print("{0:.3f}".format((data['ordersPerDateB'].sum()/data['visitorsPerDateB'].sum())/(data['ord
print("{0:.5f}".format(stats.mannwhitneyu(sampleA, sampleB, alternative = 'two-sided')[1]))
```

```
0.138
0.01679
```

P-value значительно меньше 0.05, поэтому нулевую гипотезу отвергаем. Анализ "сырых" данных говорит, что в конверсиях между группами есть статистически значимые различия. Относительный прирост конверсии группы B к конверсии группы A равен 13.8%.

### Автоматическая проверка двухсторонней гипотезы.

In [38]:
```
print("{0:.5f}".format(stats.mannwhitneyu(sampleA, sampleB, alternative = 'two-sided')[1]))
```

```
0.01679
```

Проверка двусторонней гипотезы подтвердила результат.Анализ "сырых" данных говорит, что в конверсиях между группами есть статистически значимые различия.

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

In [39]:
```
print('{0:.3f}'.format(stats.mannwhitneyu(orders[orders['group']=='A']['revenue'], orders[order
print('{0:.3f}'.format(orders[orders['group']=='B']['revenue'].mean()/orders[orders['group']=='
```

```
0.729
0.259
```

p- value больше 0.05 - нулевую гипотезу о том, что статистически значимых различий в среднем чеке заказа между группами по «сырым» данным нет, не отвергаем. Относительный прирост среднего чека

сегмента B к сегменту A почти 26%.

### Автоматическая проверка двухсторонней гипотезы.

In [40]:
```
print('{0:.3f}'.format(stats.mannwhitneyu(orders[orders['group']=='A']['revenue'], orders[order
```

```
0.729
```

p- value больше 0.05 - нулевую гипотезу о том, что статистически значимых различий в среднем чеке заказа между группами по «сырым» данным нет, не отвергаем.

## Подготовка очищенных от аномалий данных.

Напомним, что 95-й и 99-й перцентили средних чеков равны 28000 и 58233.2 рублям. А 95-й и 99-й перцентили числа заказов на одного пользователя равны 2 и 4 заказам на пользователя.Примем за аномальных пользователей тех, кто совершил 4 заказа и более, или совершил заказ на сумму свыше 30000 рублей.Так мы уберём 1% пользователей с наибольшим числом заказов и от 1% до 5% заказов с наибольшей стоимостью

In [41]:
```
ordersByUsersB.columns = ['visitor_id', 'orders']
usersWithManyOrders = pd.concat(
    [
       ordersByUsersA[ordersByUsersA['orders'] > 3]['visitor_id'],
    ordersByUsersB[ordersByUsersB['orders'] > 3]['visitor_id'],],
    axis=0,
)
usersWithExpensiveOrders = orders[orders['revenue'] > 30000]['visitor_id']
abnormalUsers = (
pd.concat([usersWithManyOrders, usersWithExpensiveOrders], axis=0)
    .drop_duplicates()
    .sort_values()
)
print(abnormalUsers.head(5))
print(abnormalUsers.shape)
```

```
1099    148427295
18      199603092
928     204675465
23      237748145
684     358944393
Name: visitor_id, dtype: int64
(57,)
```

Получили 57 аномальных пользователей.Узнаем, как их действия повлияли на результаты теста. После их удаления нужно посчитать очищенные данные.

## Статистическая значимость различий в конверсии между группами по «очищенным» данным.

In [42]:
```
sampleAFiltered = pd.concat(
    [
        ordersByUsersA[
            np.logical_not(ordersByUsersA['visitor_id'].isin(abnormalUsers))
        ]['orders'],
        pd.Series(
            0,
            index=np.arange(
                data['visitorsPerDateA'].sum() - len(ordersByUsersA['orders'])
            ),
            name='orders',
        ),
    ],
    axis=0,
)
```

```
sampleBFiltered = pd.concat(
    [
        ordersByUsersB[
            np.logical_not(ordersByUsersB['visitor_id'].isin(abnormalUsers))
        ]['orders'],
        pd.Series(
            0,
            index=np.arange(
                data['visitorsPerDateB'].sum() - len(ordersByUsersB['orders'])
            ),
            name='orders',
        ),
    ],
    axis=0,
)
print('{0:.5f}'.format(stats.mannwhitneyu(sampleAFiltered, sampleBFiltered)[1]))
print('{0:.3f}'.format(sampleBFiltered.mean()/sampleAFiltered.mean()-1))
```

```
0.01832
0.148
```

Результаты по конверсии практически не изменились. p-value значительно меньше 0.05, как и в случае с "сырыми" данными. Статистическая значимость достигнута, сегмент B значительно лучше сегмента A на 14.8%.

### Автоматическая проверка двухсторонней гипотезы.

In [43]:
```
print('{0:.5f}'.format(stats.mannwhitneyu(sampleAFiltered, sampleBFiltered, alternative = 'two-
```

```
0.01832
```

Результаты по конверсии практически не изменились. p-value значительно меньше 0.05, как и в случае с "сырыми" данными.

## Статистическая значимость различий в среднем чеке заказа между группами по «очищенным» данным

In [44]:
```
print(
    '{0:.3f}'.format(
        stats.mannwhitneyu(
            orders[
                np.logical_and(
                    orders['group'] == 'A',
                    np.logical_not(orders['visitor_id'].isin(abnormalUsers)),

                )
            ]['revenue'],
            orders[
                np.logical_and(
                    orders['group'] == 'B',
                    np.logical_not(orders['visitor_id'].isin(abnormalUsers)),
                )
            ]['revenue'],
        )[1]
    )
)

print(
    "{0:.3f}".format(
        orders[
            np.logical_and(
                orders['group'] == 'B',
                np.logical_not(orders['visitor_id'].isin(abnormalUsers)),
            )
        ]['revenue'].mean()
        / orders[
```

```
            np.logical_and(
                orders['group'] == 'A',
                np.logical_not(orders['visitor_id'].isin(abnormalUsers)),
            )
        ]['revenue'].mean()
        - 1
    )
)
```

```
0.958
-0.020
```

После удаления аномалий картина не поменялась. p-value = 0.479 больше 0.05. Значит,нулевую гипотезу о том, что статистически значимых различий в конверсии между группами нет, не отвергаем. Относительный проигрыш группы B равен 2% . Это может быть просто "шум".

### Автоматическая проверка двухсторонней гипотезы

In [45]:

```
print(
    '{0:.3f}'.format(
        stats.mannwhitneyu(
            orders[
                np.logical_and(
                    orders['group'] == 'A',
                    np.logical_not(orders['visitor_id'].isin(abnormalUsers)),

                )
            ]['revenue'],
            orders[
                np.logical_and(
                    orders['group'] == 'B',
                    np.logical_not(orders['visitor_id'].isin(abnormalUsers)),
                )
            ]['revenue'], alternative = 'two-sided'
        )[1]
    )
)
```

```
0.958
```

После удаления аномалий картина не поменялась. p-value = 0.958 больше 0.05. Значит,нулевую гипотезу о том, что статистически значимых различий в конверсии между группами нет, не отвергаем.

In [46]:

```
cumulativeData['conversion'] = (
    cumulativeData['orders'] / cumulativeData['visitors']
)
plt.figure(figsize = (20,6))
cumulativeDataA = cumulativeData[cumulativeData['group'] == 'A']
cumulativeDataB = cumulativeData[cumulativeData['group'] == 'B']
mergedCumulativeConversions = cumulativeDataA[['date','conversion']].merge(cumulativeDataB[['da

plt.plot(mergedCumulativeConversions['date'], mergedCumulativeConversions['conversionB']/merged
plt.title('График относительного изменения кумулятивной конверсии группы B к группе A.',size =
plt.xlabel('Дата',size =15)
plt.ylabel('Отношение B к A',size =15)
plt.axhline(y=0, color='black', linestyle='--')
plt.axhline(y=0.2, color='grey', linestyle='--')
plt.grid()
plt.show()
```


In [47]:

```
mergedCumulativeRevenue = cumulativeRevenueA.merge(cumulativeRevenueB, left_on='date', right_or
plt.figure(figsize = (20,6))
plt.plot(mergedCumulativeRevenue['date'], (mergedCumulativeRevenue['revenueB']/mergedCumulative
plt.title('График относительного изменения кумулятивного среднего чека группы B к группе A',siz
plt.xlabel('Дата',size =15)
plt.ylabel('Отношение B к A',size =15)
plt.axhline(y=0, color='black', linestyle='--')
plt.grid()
plt.show()
```


Есть статистически значимое различие по конверсии между группами как по сырым данным, так и после фильтрации аномалий. Нет статистически значимого различия по среднему чеку между группами как по сырым данным, так и после фильтрации аномалий. График различия конверсии между группами сообщает, что результаты группы B лучше группы A: имеют тенденцию к росту. Сейчас данные группы B лучше на почти 15 %. График различия среднего чека колеблется: он-то и позволил вам найти аномалии.

Исходя из обнаруженных фактов,остановить тест, признать его успешным и перейти к проверке следующей гипотезы.

# Выводы.

В ходе проекта обработал полученные данные крупного интернет-магазина. Первым делом проведена предобработка данных на наличие пропусков, дубликатов. Определил и изучил пропущенные значения. Там, где это необходимо, заменил типы данных на необходимые для удобной работы. Проверил,есть ли пользователи, которые попали в две группы теста одновременно. Таких 58 человек. Это мало, можно пренебречь.

В первой части проекта изучил список представленных гипотез для увеличения выручки,приоритизировал их. Для приоритизации гипотез применил фреймворки ICE и RICE.Указал, как изменилась приоритизация гипотез при применении RICE вместо ICE.Так если вначале приоритетны были гипотезы: "Запустить акцию, дающую скидку на товар в день рождения" и

"Добавить два новых канала привлечения трафика, что позволит привлекать на 30% больше пользователей", то после - "Добавить форму подписки на все основные страницы, чтобы собрать базу клиентов для email-рассылок" и "Добавить блоки рекомендаций товаров на сайт интернет магазина, чтобы повысить конверсию и средний чек заказа". Это стало возможным из-за большего охвата аудитории.

Во второй части проанализировал A/B-тест: Построил график кумулятивной выручки по группам. Построил график кумулятивного среднего чека по группам. Построил график относительного изменения кумулятивного среднего чека группы B к группе A. Построил график кумулятивной конверсии по группам. Построил график относительного изменения кумулятивной конверсии группы B к группе A. Построил точечный график количества заказов по пользователям.

```
Посчитал 95-й и 99-й перцентили количества заказов на пользователя. Выбрал
границу для определения аномальных пользователей.
Построил точечный график стоимостей заказов.
Посчитал 95-й и 99-й перцентили стоимости заказов. Выбрал границу для
определения аномальных заказов.
Посчитал статистическую значимость различий в конверсии между группами по
«сырым» данным.
Посчитал статистическую значимость различий в среднем чеке заказа между группами
по «сырым» данным.
Посчитал статистическую значимость различий в конверсии между группами по
«очищенным» данным.
Посчитал статистическую значимость различий в среднем чеке заказа между группами
по «очищенным» данным.

Результаты по "сырым" данным:
по конверсии p- value меньше заданного, значит нулевую гипотезу не подтверждаем,
статистически значимые различия есть. Прирост группы B относительно группы A
13.8%. По чекам p- value больше заданного, значит, нулевую гипотезу не
отвергаем. Различий нет.Прирост группы B относительно группы A 26%.

После удаления аномальных значений картина не поменялась. По конверсии p- value
меньше заданного,статистически значимые различия есть.Прирост группы B
относительно группы A 14.8%.
По чекам p- value больше заданного, значит, нулевую гипотезу не отвергаем.
Отношение группы B к группе A 2%. Это скорее всего, "шум".

Принял решение по результатам теста:
   Метрики стабилизировались, поэтому можно остановить тест, признать его
успешным.
   Есть статистически значимые отличия по конверсии между группами. Группа B
опережает группу A.
   По среднему чеку статистически значимых отличий между группами нет.
   Можно выдать интернет-магазину рекомендации :
         конверсия тестовой группа лучше платит, внедряем тестируемые изменения
на весь продукт.
```

https://pandas.pydata.org/pandas-docs/stable/user_guide/style.html