# A/B-тестирование

## Постановка задачи

Задача — провести анализ результатов A/B-теста. В качестве есть данные о действиях пользователей, технические сведения о распределении пользователей по экспериментальным группам.

- Сформировать предварительные гипотезы;
- Проанализировать результаты теста.

Для каждого пользователя мы располагаем тем, кой/какие покупки/ покупки:

- пользователи, которые зарегистрировались, но покупок не совершили;
- пользователи, которые совершили только одну покупку (например, сделали тестовую);
- пользователи, которые совершили несколько покупок.

## Техническое задание

- Название теста: `recommendation_subsets_test`;
- Дата запуска: `2020-12-07`;
- Дата остановки набора новых пользователей: `2020-12-21`;
- Группы: 2 группы (контрольная и тестовая);
- Распределение трафика: `static`;
- Аудитория: 15% новых пользователей из региона `EU`;
- Ожидаемое количество участников теста: `6000` — с учетом ограничения уникальной рекламной платформы;
- Ожидаемый эффект: за 14 дней с момента регистрации пользователи должны улучшить показатели каждой метрики не менее, чем на 10%:
  - конверсия в просмотры карточек товаров — событие `product_page`;
  - просмотры карточек — событие `product_page`;
  - покупки — событие `purchase`.

## Оглавление


- Шаг 1. Импортирование библиотек, загрузка данных и их предварительная проверка, анализ качества данных
- Шаг 2. Исследовательский анализ данных
- Шаг 3. Оценка результатов A/B-теста
- Шаг 4. Общий вывод


## Загрузка данных

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
import math as mth
import scipy.stats as st
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
pio.set_template('plotly_white')
from plotly.subplots import make_subplots
from matplotlib.ticker import PercentFormatter
from pandas.plotly.tools.set_credentials_file(username='Practicumstudent',
                              api_key='ZuoZ0Ln0Ln0Ln0Ln0Ln0L')
from plotly.offline.offline import init_notebook_mode
init_notebook_mode(connected=True)
from plotly.offline.offline import iplot
from plotly.graph_objects import Scatter, Layout
from plotly.tools import make_subplots, tools_digest
from plotly.graph_objects import Histogram, Bar, Box, Scatter, Pie, ScatterMapbox
from plotly.graph_objects import Heatmap, Contour, Table, Sankey, Funnel
from plotly.graph_objects import Funnelarea, Indicator, Waterfall
from plotly.graph_objects import Scattergeo, Choropleth, ScatterMapbox
from plotly.graph_objects import Scatter3d, Surface, Mesh3d, Cone, Volume
from plotly.graph_objects import Parcoords, Sunburst, Treemap, Icicle
from plotly.graph_objects import Pointcloud, Heatmapgl, Contourgl
from plotly.graph_objects import Scattergl, Splom, Sankey
```

## Загрузим данные

```
ab_project_marketing_events = pd.read_csv('/datasets/ab_project_marketing_events.csv')
ab_project_marketing_events.head(5)
```

## Описание данных

`ab_project_marketing_events.csv` — календарь маркетинговых событий на 2020 год.

Структура файла:

- `name` — название маркетингового события;
- `regions` — перечень участников будущего рекламного кампании;
- `start_dt` — дата начала кампании;
- `finish_dt` — дата завершения кампании.

`final_ab_new_users.csv` — пользователи, зарегистрировавшиеся с 7 по 21 декабря 2020 года.

Структура файла:

- `user_id` — идентификатор пользователя;
- `first_date` — дата регистрации;
- `region` — название региона пользователя;
- `device` — устройство, с которого зарегистрировался пользователь.

`final_ab_events.csv` — действия новых пользователей в период с 7 декабря 2020 года по 4 января 2021 года.

Структура файла:

- `user_id` — идентификатор пользователя;
- `event_dt` — дата и время покупки;
- `event_name` — тип события;
- `details` — дополнительные данные о событиях. Например, для покупок, `purchase`, в этом поле хранится стоимость покупки в долларах.

`final_ab_participants.csv` — таблица участников теста.

Структура файла:

- `user_id` — идентификатор пользователя;
- `group` — группа теста;
- `ab_test` — название теста.

## Шаг 1. Предварительная обработка данных

Выведем общую информацию по данным `ab_project_marketing_events`. Проверим на дубликаты.

```
ab_project_marketing_events.info()
```

```
<class RangeIndex from pandas.core.indexes.range.RangeIndex>
Int64Index: 14 entries, 0 to 13
Data columns (total 4):
name       14 non-null object
regions    14 non-null object
start_dt   14 non-null object
finish_dt  14 non-null object
dtypes: object(4)
memory usage: 1.5 KB
```

```
ab_project_marketing_events.duplicated().sum()
```

```
0
```

Данные состоят из 14 строк и 4 столбцов. Названия столбцов приостановлены, не требуют замены. Дубликаты в пропуски отсутствуют. Необходимо изменить тип данных у столбцов `start_dt` и `finish_dt` на формат `date`.

```
ab_project_marketing_events['start_dt'] = ab_project_marketing_events['start_dt'].astype('datetime64[ns]')
ab_project_marketing_events['finish_dt'] = ab_project_marketing_events['finish_dt'].astype('datetime64[ns]')
```

Выведем общую информацию по данным `final_ab_new_users`. Проверим на дубликаты.

```
final_ab_new_users.info()
```

```
<class RangeIndex from pandas.core.indexes.range.RangeIndex>
Int64Index: 61733 entries, 0 to 61732
Data columns (total 4):
user_id    61733 non-null int64
first_date 61733 non-null object
region     61733 non-null object
device     61733 non-null object
dtypes: int64(1), object(3)
memory usage: 2.3 MB
```

```
final_ab_new_users.duplicated().sum()
```

```
0
```

Данные состоят из 61733 строк и 4 столбцов. Названия столбцов приостановлены, не требуют замены. Дубликаты в пропуски отсутствуют. Необходимо изменить тип данных у столбцов `first_date` на формат `date`.

```
final_ab_new_users['first_date'] = final_ab_new_users['first_date'].astype('datetime64[ns]')
```

Выведем общую информацию по данным `final_ab_events`. Проверим на дубликаты.

```
final_ab_events.info()
```

```
<class RangeIndex from pandas.core.indexes.range.RangeIndex>
Int64Index: 440317 entries, 0 to 440316
Data columns (total 4):
user_id    440317 non-null int64
event_dt   440317 non-null object
event_name 440317 non-null object
details    62740 non-null float64
dtypes: float64(1), int64(1), object(2)
memory usage: 16.8 MB
```

```
final_ab_events.duplicated().sum()
```

```
0
```

Данные состоят из 440317 строк. Названия столбцов приостановлены, не требуют замены. Дубликаты отсутствуют. Необходимо изменить тип данных у столбцов `event_dt` на формат `date`. Пропуски есть в столбце `details`. Необходимо заменить пропуски на `0`, так как, согласно ТЗ, `details` — это стоимость покупки.

```
final_ab_events['event_dt'] = final_ab_events['event_dt'].astype('datetime64[ns]')
final_ab_events['details'].fillna(0, inplace=True)
```

Выведем общую информацию по данным `final_ab_participants`. Проверим на дубликаты.

```
final_ab_participants.info()
```

```
<class RangeIndex from pandas.core.indexes.range.RangeIndex>
Int64Index: 18268 entries, 0 to 18267
Data columns (total 3):
user_id    18268 non-null object
group      18268 non-null object
ab_test    18268 non-null object
dtypes: object(3)
memory usage: 570.9 KB
```

```
final_ab_participants.duplicated().sum()
```

```
0
```

Данные состоят из 18268 строк. Названия столбцов приостановлены, не требуют замены. Дубликаты отсутствуют. Пропуски отсутствуют. Необходимо изменить тип данных у столбцов `user_id` на формат `int`.

```
final_ab_participants['user_id'] = final_ab_participants['user_id'].astype('int64')
```

## Шаг 2. Исследовательский анализ данных

- Определим количество пользователей на каждом этапе воронки;
- Выяснить объем/долю пользователей целевой аудитории в выборке?
- Проверим, не пересекаются ли пользователи в группах?
- Проверим распределение по регионам и устройствам пользователей, принимавших участие в тесте?

Сделаем предварительно проверку теста. Проверим, что новые пользователи теста не совершали и маркетинговых событий, влияющих на их выборку, не проводилось.

```
final_ab_new_users[(final_ab_new_users['first_date'] > '2020-12-14') \
                   | (final_ab_new_users['first_date'] < '2020-12-07')].shape[0]
```

```
0
```

Есть 0 строк нарушения теста. `recommendation_subsets_test` с тестом `recommendation_subsets`. Сохраним его переменную `recommendation_subsets_test`.

```
recommendation_subsets_test = final_ab_participants[final_ab_participants['ab_test'] == 'recommendation_subsets']
recommendation_subsets_test.shape[0]
```

```
3824
```

Всего 3824 пользователя участвовали в `recommendation_subsets_test` с тестом `recommendation_subsets`. Сохраним его переменную `recommendation_subsets_test`.

```
recommendation_subsets_test = final_ab_participants[final_ab_participants['ab_test'] == 'recommendation_subsets']
recommendation_subsets_test.shape[0]
```

```
3824
```

Пересечения `recommendation_subsets_test` с другими тестами.

```
recommendation_subsets_test[recommendation_subsets_test['user_id'].isin(final_ab_participants[final_ab_participants['ab_test'] != 'recommendation_subsets']['user_id'])].shape[0]
```

```
0
```

Пересечений `recommendation_subsets_test` с другими тестами нет.

Определим количество пользователей на каждом этапе воронки `recommendation_subsets_test`.

```
recommendation_subsets_test.groupby('group').agg({'user_id':'count'})
```

Пропущено `recommendation_subsets_test` с разных группа.

```
recommendation_subsets_test[recommendation_subsets_test['group'].isin(['A', 'B'])].shape[0]
```

```
3824
```

Пропущенных `recommendation_subsets_test` с разных группа нет.

Определим количество пользователей на каждом этапе воронки `recommendation_subsets_test`.

```
recommendation_subsets_test.groupby('group').agg({'user_id':'count'})
```

Проверим, не пересекаются ли пользователи в группах `recommendation_subsets_test`.

```
recommendation_subsets_test[recommendation_subsets_test['user_id'].isin(recommendation_subsets_test[recommendation_subsets_test['group'] == 'A']['user_id'])].shape[0]
```

```
2866
```

```
recommendation_subsets_test[recommendation_subsets_test['user_id'].isin(recommendation_subsets_test[recommendation_subsets_test['group'] == 'B'])].shape[0]
```

```
958
```

Пересечения `recommendation_subsets_test` с разных группа есть. Сохраним его переменную `recommendation_subsets_test_crossing`.

```
recommendation_subsets_test_crossing = recommendation_subsets_test[recommendation_subsets_test['user_id'].isin(recommendation_subsets_test[recommendation_subsets_test['group'] == 'A'])]
recommendation_subsets_test_crossing.shape[0]
```

```
2866
```

Определим количество пользователей в `recommendation_subsets_test_crossing` с разных группа.

```
recommendation_subsets_test_crossing.groupby('group').agg({'user_id':'count'})
```

Определим количество пользователей в `recommendation_subsets_test` с разных группа.

```
recommendation_subsets_test.groupby('group').agg({'user_id':'count'})
```

Доля пересечений `recommendation_subsets_test` с разных группа.

```
round(2866/3824, 2)
```

```
0.75
```

Доля пересечений `recommendation_subsets_test` с разных группа — 75%. Пересечения `recommendation_subsets_test` с разных группа — 75%. Сохраним пересечения в

```
In [24]: # Группировка по exp_group и event_name
df_events = df.groupby(['exp_group', 'event_name'])['user_id'].count().reset_index()
df_events.rename(columns={'user_id': 'event_count'}, inplace=True)
print("Распределение событий по группам:")
display(df_events)
```

```
Распределение событий по группам:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [25]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=df_events, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```

**Вывод:** в целом нет, что время в эксперименте группы B, влияется сильнее на поведение в выборке распределение изменилось.

Добавим столбцы `first_day_events`,`second_day_events`,`third_day_events` и `fourth_day_events` - показывающие, сколько событий было сделано, дня.

```
In [26]: # Добавление новых столбцов
df['first_day_events'] = df['days_after_registration'] <= 1
df['second_day_events'] = df['days_after_registration'] <= 2
df['third_day_events'] = df['days_after_registration'] <= 3
df['fourth_day_events'] = df['days_after_registration'] <= 4
print("Первые 5 строк с новыми столбцами:")
display(df.head(5))
```

```
Первые 5 строк с новыми столбцами:
```

| | user_id | exp_group | event_name | date | days_after_registration | first_day_events | second_day_events | third_day_events | fourth_day_events |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 20022 | 246 | tutorial_finish | 2020-12-23 | 0.0 | True | True | True | True |
| 1 | 20022 | 246 | tutorial_start | 2020-12-23 | 0.0 | True | True | True | True |
| 2 | 20039 | 247 | tutorial_finish | 2020-12-23 | 0.0 | True | True | True | True |
| 3 | 20039 | 247 | tutorial_start | 2020-12-23 | 0.0 | True | True | True | True |
| 4 | 20059 | 246 | tutorial_finish | 2020-12-23 | 0.0 | True | True | True | True |

Посчитаем:

```
In [27]: # Группировка по exp_group
grouped = df.groupby('exp_group')
print("Количество уникальных пользователей в каждой группе:")
display(grouped['user_id'].nunique.reset_index(name='unique_users'))
```

```
Количество уникальных пользователей в каждой группе:
```

| | exp_group | unique_users |
|---|---|---|
| 0 | 246 | 2321 |
| 1 | 247 | 2341 |

```
In [28]: # Визуализация распределения пользователей по группам
plt.figure(figure=(8, 5))
sns.barplot(x='exp_group', y='unique_users', data=grouped['user_id'].nunique.reset_index(name='unique_users'), palette='viridis')
plt.title('Количество уникальных пользователей по группам')
plt.xlabel('Группа')
plt.ylabel('Количество пользователей')
plt.grid(True)
plt.show()
```

**Вывод:** распределение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

```
In [29]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count().reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [30]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```

**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

```
In [31]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count().reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [32]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```

**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

Сгенерируем по группам количественные данные и сначала, посмотрим в каждом интервале `pd.cut(bins=[0, 1, 2, 3, 4])` и сгруппим и сообщ.

```
In [33]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count().reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [34]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```

**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

Посчитаем сколько событий было в каждом интервале `pd.cut(bins=[0, 1, 2, 3, 4])` и сгруппим и сообщ. Сгруппим по группам `exp_group` и `event_name` и сгруппим по `event_name` и посчитаем сколько событий было в каждом дне.

```
In [35]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count().reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [36]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```

**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

```
In [37]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count().reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [38]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```

**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

Посчитаем сколько `first_day_events`,`second_day_events`,`third_day_events` и `fourth_day_events` - показывающие, сколько событий было сделано, дня.

```
In [39]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count().reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [40]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```

**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

```
In [41]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count.reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [42]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```




**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Шаг 2. Оценка результатов A/B-тестирования

- Что можно сказать про результаты A/B-теста?
- Проверьте статистическую разницу долей z-критерием.

Посчитаем сколько `first_day_events`,`second_day_events`,`third_day_events` и `fourth_day_events` - показывающие, сколько событий было сделано, дня.

```
In [43]: # Группировка по exp_group и event_name
grouped = df.groupby(['exp_group', 'event_name'])['user_id'].count.reset_index(name='event_count')
print("Количество событий по группам и событиям:")
display(grouped)
```

```
Количество событий по группам и событиям:
```

| | exp_group | event_name | event_count |
|---|---|---|---|
| 0 | 246 | tutorial_finish | 1065 |
| 1 | 246 | tutorial_start | 1256 |
| 2 | 247 | tutorial_finish | 1077 |
| 3 | 247 | tutorial_start | 1264 |

```
In [44]: # Визуализация распределения событий по группам
plt.figure(figure=(10, 6))
sns.barplot(x='event_name', y='event_count', hue='exp_group', data=grouped, palette='viridis')
plt.title('Распределение событий по группам')
plt.xlabel('Событие')
plt.ylabel('Количество')
plt.legend(title='Группа')
plt.grid(True)
plt.show()
```




**Вывод:** соотношение активности достаточно равно 2300-2341 . Сделаю это что в эксперименте набрано новых пользователей или с начала (пользователи ждут Релиз?)

## К оглавлению

## Выводы:

**Тест проверки нормальности и его графический анализ результатов 8.Не нормальное распределение стандарт прямого или обратного**

Время проведения теста на компьютере: 12:00:00 работа Тест Ростест, сгенерирован 2020-12-28.

Есть 1800 случаев представляющих с распределением по тесту 1200000000_AC_0400 . После не удачного количества участников тест остановлен на 1800 случаях с распределением по тесту.

В группе 8 больше положительной, чем в группе 9, далее количество распределяется положительной составляет.

Полностью распределяется тесть для группы в группе 8 равно (ячеек) = 0,77% до 32,7%. И полная конверсия в проектном тесте для группы 8 равна 0,77% и для группы 9 равна 0,77% Разница по группам тесть оценка проектов равна(ячеек) = -0,77% до 32,7% и -32,7% до 0,77%.

С учетом того, что группа 8 экспериментальная группа 9, сравнительно собранная на основании и выборка распределение проектов.

Распределение проектов, рассчитан тесть 2020-12-27. Сначала на что и состоялась выборка тесть положительной тесть 2020-12-28 в 12:00:00.

Выборка группы 8 для 32 закрытия

Сравнение группы 8 и тесть для до 2020-12-11, затем формат для 2020-12-20, после чего собственности. Сравнение тесть группы 8 положительно для до 2020-12-27, затем собственности

Выборка группы 9 для 32 закрытия

Сравнение тесть группы 9 и тесть для до 2020-12-11, затем формат для 2020-12-20, после чего собственности. Сравнение тесть группы 9 положительно для до 2020-12-27, затем собственности

Нулевая гипотеза для группы 8 равна 0,77%

Нулевая гипотеза для группы 9 равна 0,77%

Нет статистически значимой разницы между конверсиями в группу 8 и 9 и группу 9 и группу 8 равна 0,77%

Нет статистически значимой разницы между сравнение тесть для 8 и 9.

Нет статистически значимой разницы между конверсиями группы 8 и 9 и группы 9 и группы 8 и группы 8 и группы 9 равна 0,77% Между конверсиями в проектах по группам тесть нет статистически значимых. Нет статистической группы 8 и 9. 0,77% равна

К оглавлению