Проверка результатов A/B теста

SUMENKOV ILYA

Я начинающий (Junior Data Analyst). И выбрал данный pet-project для того, чтобы улучшить свои навыки Анализа данных. Целью данного пет-проекта является выявление эффективности нового дизайна в сравнении со старым. И будут ли изменения работать лучше для компании или нет.

1. ПОДГОТОВКА

Источник данных

Используемые данные были взяты из следующего общедоступного набора данных: A/B testing

Данные общедоступны на Kaggle и хранятся в 1 CSV-файле.

Собранные данные включают

User ID - Уникальный идентификатор пользователя.

TimeStamp - Время начала сеанса для пользователя.

Group - Содержит 2 разных значения в качестве контроля и обработки.

Landing Page - Содержит 2 разных значения как 'old_page' и 'new_page'.

Converted - Представляет поведение пользователя: совершил ли пользователь покупку (1) или нет (0).

2. ПРОЦЕСС

Подготовка рабочей среды

Я буду использовать Python для очистки, преобразования и визуализации данных. Установлены следующие библиотеки:

In [37]:
```
import statsmodels.stats.api as sms
from statsmodels.stats.proportion import proportions_ztest
from statsmodels.stats.proportion import proportion_confint
import scipy.stats as stats
from math import ceil
```

In [38]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import seaborn as sns
%matplotlib inline
```

Импорт набора данных

In [39]:
```
data = pd.read_csv("C:/Users/sumen/OneDrive/Рабочий стол/Аналитика/Analytics projects/AB testing/ab_data.csv")
```

In [40]:
```
num_observations = len(data)
```

In [41]:
```
num_observations
```

Out[41]:
```
294478
```

Просмотр данных

In [42]:
```
data.head()
```

Out[42]:

	user_id	timestamp	group	landing_page	converted
0	851104	2017-01-21 22:11:48.556739	control	old_page	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0
4	864975	2017-01-21 01:52:26.210827	control	old_page	1

In [43]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294478 entries, 0 to 294477
Data columns (total 5 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   user_id       294478 non-null  int64 
 1   timestamp     294478 non-null  object
 2   group         294478 non-null  object
 3   landing_page  294478 non-null  object
 4   converted     294478 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 11.2+ MB
```

In [44]:
```
data.describe()
```

Out[44]:

	user_id	converted
count	294478.000000	294478.000000
mean	787974.124733	0.119659
std	91210.823776	0.324563
min	630000.000000	0.000000
25%	709032.250000	0.000000
50%	787933.500000	0.000000
75%	866911.750000	0.000000
max	945999.000000	1.000000

In [45]:
```
group_counts = data['group'].value_counts()
```

In [46]:
```
group_counts
```

Out[46]:
```
group
treatment    147276
control      147202
Name: count, dtype: int64
```

In [47]:
```
data['landing_page'].value_counts()
```

Out[47]:
```
landing_page
old_page    147239
new_page    147239
Name: count, dtype: int64
```

In [48]:
```
conversion_rate = data['converted'].mean()
```

In [49]:
```
conversion_rate
```

Out[49]:
```
0.11965919355605512
```

In [50]:
```
control_group = data[data['group'] == 'control']
treatment_group = data[data['group'] == 'treatment']
```

In [51]:
```
control_group
```

Out[51]:

	user_id	timestamp	group	landing_page	converted
0	851104	2017-01-21 22:11:48.556739	control	old_page	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0
4	864975	2017-01-21 01:52:26.210827	control	old_page	1
5	936923	2017-01-10 15:20:49.083499	control	old_page	0
7	719014	2017-01-17 01:48:29.539573	control	old_page	0
...	...	...	...	...	...
294471	718310	2017-01-21 22:44:20.378320	control	old_page	0
294473	751197	2017-01-03 22:28:38.630509	control	old_page	0
294474	945152	2017-01-12 00:51:57.078372	control	old_page	0
294475	734608	2017-01-22 11:45:03.439544	control	old_page	0
294476	697314	2017-01-15 01:20:28.957438	control	old_page	0

147202 rows × 5 columns

In [52]:
```
treatment_group
```

Out[52]:

	user_id	timestamp	group	landing_page	converted
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0
6	679687	2017-01-19 03:26:46.940749	treatment	new_page	1
8	817355	2017-01-04 17:58:08.979471	treatment	new_page	1
9	839785	2017-01-15 18:11:06.610965	treatment	new_page	1
...	...	...	...	...	...
294462	677163	2017-01-03 19:41:51.902148	treatment	new_page	0
294465	925675	2017-01-07 20:38:26.346410	treatment	new_page	0
294468	643562	2017-01-02 19:20:05.460595	treatment	new_page	0
294472	822004	2017-01-04 03:36:46.071379	treatment	new_page	0
294477	715931	2017-01-16 12:40:24.467417	treatment	new_page	0

147276 rows × 5 columns

In [53]:
```
conversion_rate_control = control_group['converted'].mean()
conversion_rate_treatment = treatment_group['converted'].mean()
```

In [54]:
```
conversion_rate_control
```

Out[54]:
```
0.12039917935897611
```

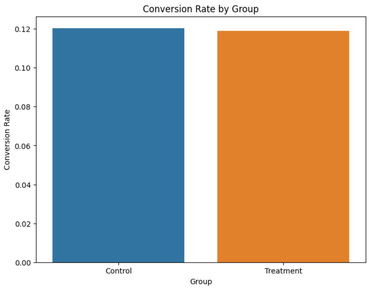
In [55]:
```
conversion_rate_treatment
```

Out[55]:
```
0.11891957956489856
```

In [56]:
```
print('Процент конверсии группы control_grou',conversion_rate_control)
print('Процент конверсии группы treatment_group',conversion_rate_treatment)
```

```
Процент конверсии группы control_grou 0.12039917935897611
Процент конверсии группы treatment_group 0.11891957956489856
```

In [57]:
```
plt.figure(figsize=(8, 6))
sns.barplot(x=['Control', 'Treatment'], y=[conversion_rate_control, conversion_rate_treatment])
plt.xlabel('Group')
plt.ylabel('Conversion Rate')
plt.title('Conversion Rate by Group')
plt.show()
```



In [58]:
```
# Проверим статистическую значимость различий между группами

z_score, p_value = proportions_ztest([control_group['converted'].sum(), treatment_group['converted'].sum()],
                                     [len(control_group), len(treatment_group)])
if p_value < 0.05:
    print('Различие между группами статистически значимо.')
else:
    print('Различие между группами не является статистически значимым.')
```

```
Различие между группами не является статистически значимым.
```

In [59]:
```
# Рассчитаем доверительный интервал для каждой группы
ci_control = proportion_confint(control_group['converted'].sum(), len(control_group), alpha=0.05)
ci_treatment = proportion_confint(treatment_group['converted'].sum(), len(treatment_group), alpha=0.05)

print('Доверительный интервал для контрольной группы:', ci_control)
print('Доверительный интервал для контрольной группы:', ci_treatment)
```

```
Доверительный интервал для контрольной группы: (0.11873674000172378, 0.12206161871622843)
Доверительный интервал для контрольной группы: (0.11726641320754189, 0.12057274592225523)
```

In [60]:
```
# Подведем итоги и сделаем выводы:
if conversion_rate_treatment > conversion_rate_control:
    print('Новый дизайн сайта приводит к более высокому проценту конверсии.')
else:
    print('Новый дизайн сайта не приводит к более высокому проценту конверсии.')
```

```
Новый дизайн сайта не приводит к более высокому проценту конверсии.
```

Вывод:

Различие между группами не является статистически значимым. Новый дизайн сайта не приводит к более высокому проценту конверсии.