Analysis of A/B test results

Test start: March 24, 2017 at 16:00

Control group: users with an even sender_id

Test group: users with odd sender_id

Question: Should the innovation be introduced for everyone or rejected?

In order to evaluate the results of this A / B test, I decided to use how many likes a particular user sent during the test as a metric.

This metric is quantitative, therefore, we will compare samples.

Let's formulate hypotheses:

H0: Changing the heart to a checkmark will not affect the number of likes users give to the app. Changes in the number of likes for groups A (control) and B (test) are not really different and the observed differences are random.

H1: Changing the heart to a checkmark will increase the number of likes users give to the app. The number of likes in group B (test) is higher than in group A (control) and these differences are the result of changes.

Data preprocessing

In [1]:
```
import matplotlib
import pandas as pd
import numpy as np
import seaborn as sns
import datetime
from matplotlib.colors import ListedColormap
from matplotlib import pyplot as plt
cmap = sns.color_palette("rocket")

font = {'family' : 'DejaVu Sans',
        'weight' : 'normal',
        'size'   : 15}

matplotlib.rc('font', **font)
matplotlib.rcParams['axes.titlesize'] = 'medium'
matplotlib.rcParams['axes.labelsize'] = 'medium'
matplotlib.rc('xtick', labelsize=10)
matplotlib.rcParams['figure.dpi'] = 300
matplotlib.rcParams['figure.figsize'] = (16,8)
matplotlib.rc('ytick', labelsize=10)
```

In [2]:
```
df = pd.read_csv("test_results.csv", sep=";")
df.head(10)
```

Out[2]:

	sender_id	platform_id	time_stamp	gender	reg_date
0	3207526951	6	16.03.2017 13:35	m	26.01.2017
1	3207526951	6	16.03.2017 9:09	m	26.01.2017
2	3207526951	6	16.03.2017 9:09	m	26.01.2017
3	3207526951	6	16.03.2017 12:13	m	26.01.2017
4	3207526951	6	15.03.2017 14:01	m	26.01.2017
5	3207526951	6	15.03.2017 12:21	m	26.01.2017
6	3207526951	6	15.03.2017 12:24	m	26.01.2017
7	3207526951	6	15.03.2017 12:31	m	26.01.2017
8	3207526951	6	15.03.2017 12:45	m	26.01.2017
9	3207526951	6	15.03.2017 12:45	m	26.01.2017

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768439 entries, 0 to 768438
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   sender_id    768439 non-null  int64 
 1   platform_id  768439 non-null  int64 
 2   time_stamp   768439 non-null  object
 3   gender       768439 non-null  object
 4   reg_date     768439 non-null  object
dtypes: int64(2), object(3)
memory usage: 29.3+ MB
```

Plan:

1. Bring data types to those needed for analysis
2. Make indexing by groups A / B by even / odd sender_id

In [4]:

```
df['time_stamp'] = pd.to_datetime(df['time_stamp'], format='%d.%m.%Y %H:%M')
df['reg_date'] = pd.to_datetime(df['reg_date'], format='%d.%m.%Y')
```

In [5]:

```
mapping = {6:'desktop', 7:'mobile'}
df = df.replace({'platform_id': mapping})
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768439 entries, 0 to 768438
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype         
---  ------       --------------   -----         
 0   sender_id    768439 non-null  int64         
 1   platform_id  768439 non-null  object        
 2   time_stamp   768439 non-null  datetime64[ns]
 3   gender       768439 non-null  object        
 4   reg_date     768439 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(2)
memory usage: 29.3+ MB
```

In [7]:
```
group_ind = []

for el in df['sender_id']:
    if int(el) % 2:
        group_ind.append('B')
    else:
        group_ind.append('A')
```

In [8]:
```
df.insert(0, 'group', group_ind)
#df.insert(2, 'clicks', np.ones(shape=len(group_ind)))
```

In [9]:
```
df = df.set_index('group')
```

In [10]:
```
# create new dataframes for each of the user groups

group_even_df = df.loc['A']
group_odd_df = df.loc['B']
```

In [11]:
```
group_even_df.head()
```

Out[11]:

group	sender_id	platform_id	time_stamp	gender	reg_date
A	3207528814	desktop	2017-03-13 17:09:00	m	2017-01-26
A	3207528814	desktop	2017-03-13 18:00:00	m	2017-01-26
A	3207528814	desktop	2017-03-13 17:16:00	m	2017-01-26
A	3207528814	desktop	2017-03-13 17:10:00	m	2017-01-26
A	3207528814	desktop	2017-03-13 17:11:00	m	2017-01-26

In [12]:
```
group_odd_df.head()
```

Out[12]:

group	sender_id	platform_id	time_stamp	gender	reg_date
B	3207526951	desktop	2017-03-16 13:35:00	m	2017-01-26
B	3207526951	desktop	2017-03-16 09:09:00	m	2017-01-26
B	3207526951	desktop	2017-03-16 09:09:00	m	2017-01-26
B	3207526951	desktop	2017-03-16 12:13:00	m	2017-01-26
B	3207526951	desktop	2017-03-15 14:01:00	m	2017-01-26

Analysis of the structure of user groups

This is necessary in order to make sure that the differences that will be revealed in the results of the A / B test do not depend on the structure of the user base.

Plan:

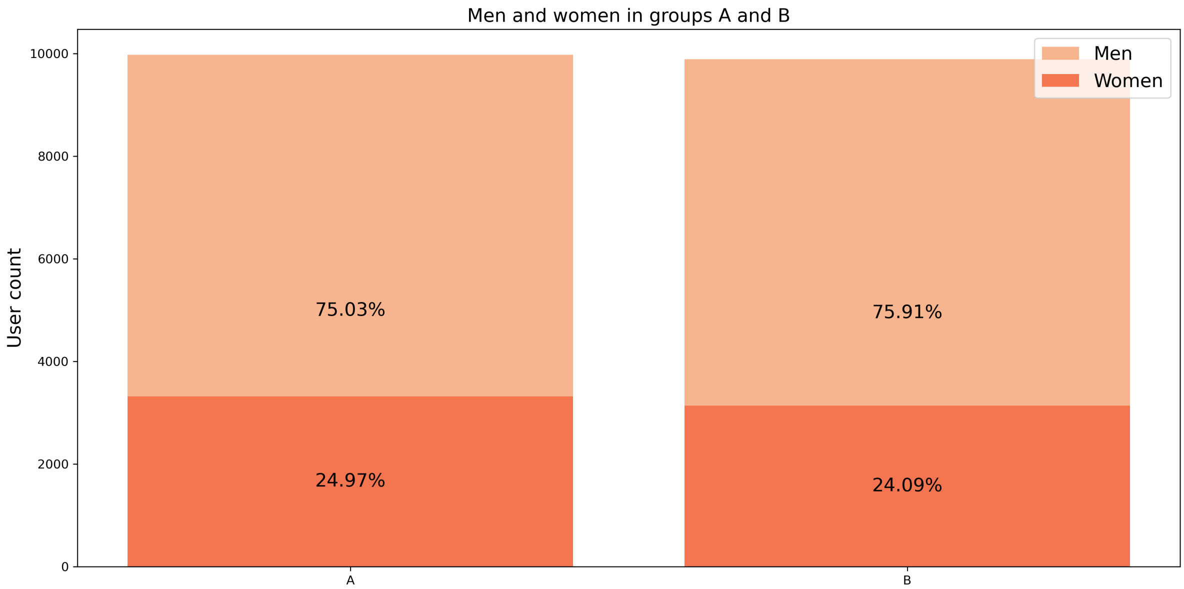
1. Check the structure of two groups according to the following criteria: gender, platform for sending a like;

2. Estimate whether the structure of groups A and B is conditionally the same.

In [13]:
```
# create dataframes of unique users for audience analysis

even_copy = group_even_df.copy(deep='true')
even_copy = even_copy.drop_duplicates(subset='sender_id')

odd_copy = group_odd_df.copy(deep='true')
odd_copy = odd_copy.drop_duplicates(subset='sender_id')
```

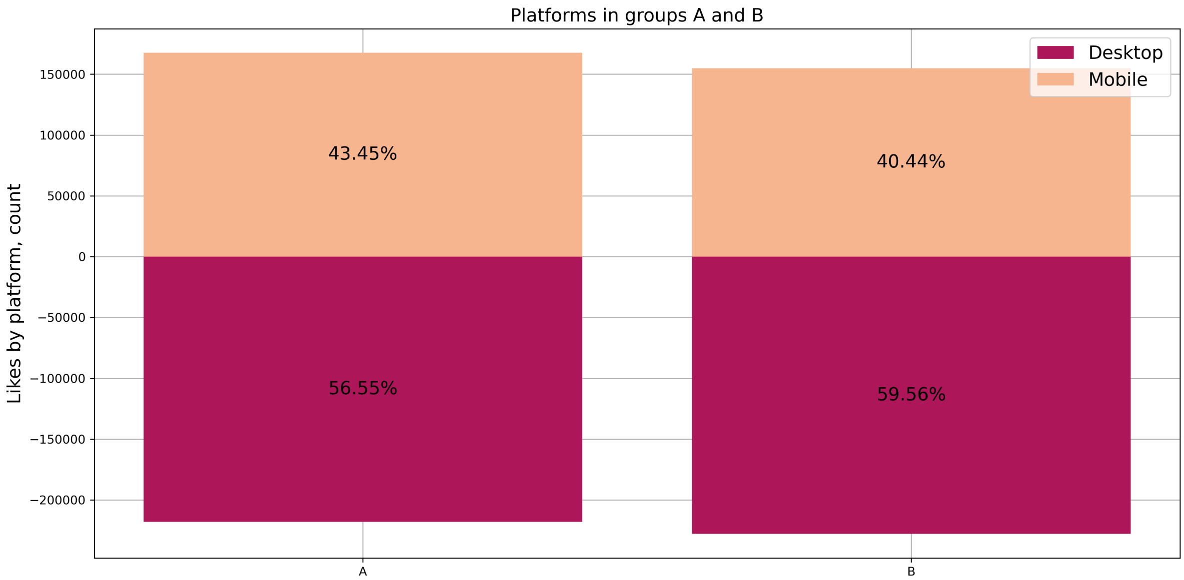
In [14]:
```
genders_even = even_copy['gender'].value_counts()
genders_odd = odd_copy['gender'].value_counts()

men = [genders_even['m'], genders_odd['m']]
women = [genders_even['f'], genders_odd['f']]
fig, ax = plt.subplots()

ax.bar(['A', 'B'], men, label='Men', color=cmap[5])
ax.bar(['A', 'B'], women, label='Women', color=cmap[4])
ax.set_ylabel("User count")
ax.set_title("Men and women in groups A and B")

percents_men = [str(np.round(men[0] / (men[0] + women[0]) * 100, 2))+'%', str(np.r
percents_women = [str(np.round(women[0] / (men[0] + women[0]) * 100, 2))+'%', str(
ax.bar_label(ax.containers[0], labels=percents_men, label_type='center')
ax.bar_label(ax.containers[1], labels=percents_women, label_type='center')

ax.legend()
```

Out[14]: <matplotlib.legend.Legend at 0x22201eba370>



In [15]:
```
platform_even = group_even_df['platform_id'].value_counts()
platform_odd = group_odd_df['platform_id'].value_counts()

desktop = [-platform_even['desktop'], -platform_odd['desktop']]
mobile = [platform_even['mobile'], platform_odd['mobile']]

fig, ax = plt.subplots()

percents_desktop = [str(np.round(-desktop[0] / (-desktop[0] + mobile[0]) * 100, 2)
percents_mobile = [str(np.round(mobile[0] / (-desktop[0] + mobile[0]) * 100, 2))+'

ax.grid()
ax.set_axisbelow(True)
```

```
ax.bar(['A', 'B'], desktop, label='Desktop', color=cmap[2])
ax.bar(['A', 'B'], mobile, label='Mobile', color=cmap[5])
ax.set_ylabel("Likes by platform, count")
ax.set_title("Platforms in groups A and B")
ax.bar_label(ax.containers[0], labels=percents_desktop, label_type='center')
ax.bar_label(ax.containers[1], labels=percents_mobile, label_type='center')
ax.legend()
plt.show()
```



Conclusions:

The structure of users is consistent, we can proceed to the evaluation of the A / B test.

Analysis of test results

Plan:

1. Cut off from groups A and B the data that was obtained before the start of the test (March 24, 2017 16:00)
2. Calculate the number of likes per user in each group
3. Transform the results into two data sets: an array with the number of clicks
4. Compare the resulting samples using the Wilcoxon rank sum test
5. Let's evaluate whether the difference between the samples is statistically significant
6. Conclusions

In [16]:
```
A_group_df = group_even_df[(group_even_df['time_stamp'] >= datetime.datetime(year=
A_group_df.sort_values('time_stamp')
A_group_df.insert(2, 'clicks', np.ones(shape=len(A_group_df)))


A_clicks_df = A_group_df[['sender_id', 'clicks']].copy()
A_clicks_df = A_clicks_df.groupby(['sender_id']).sum()
```

In [17]:
```
B_group_df = group_odd_df[(group_odd_df['time_stamp'] >= datetime.datetime(year=20
B_group_df.sort_values('time_stamp')
B_group_df.insert(2, 'clicks', np.ones(shape=len(B_group_df)))
```

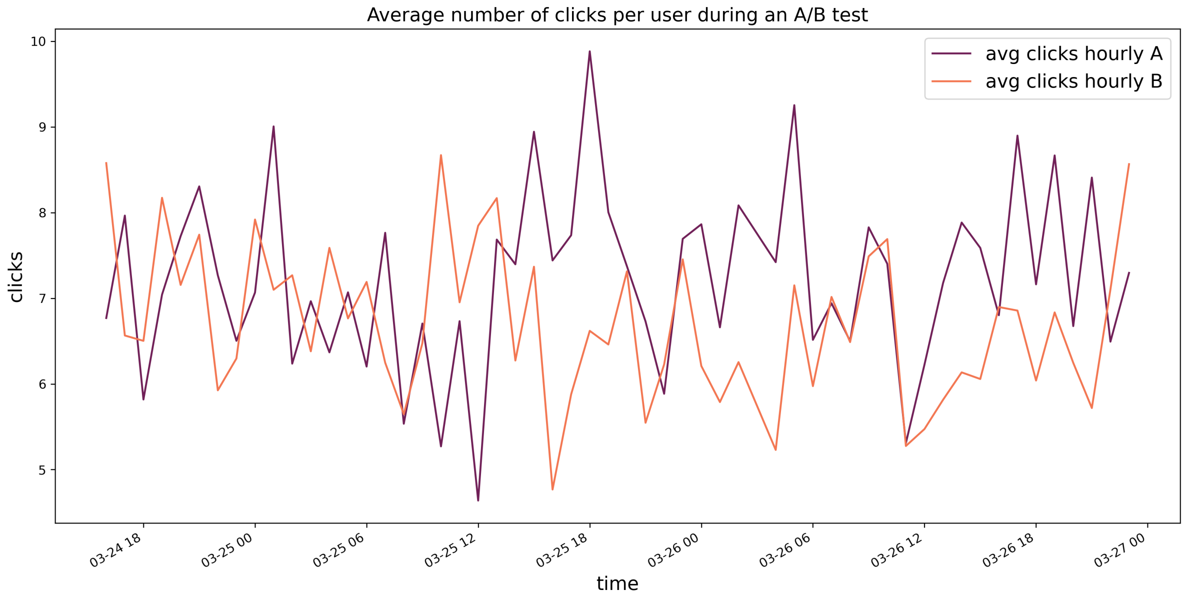
```
B_clicks_df = B_group_df[['sender_id', 'clicks']].copy()
B_clicks_df = B_clicks_df.groupby(['sender_id']).sum()
```

In [18]:
```
B_avg_clicks_by_hour = B_group_df[['sender_id', 'clicks', 'time_stamp']].copy()
B_avg_clicks_by_hour = B_avg_clicks_by_hour.groupby(
    [pd.Grouper(key='time_stamp', freq='H'), 'sender_id']
).sum()
B_avg_clicks_by_hour = B_avg_clicks_by_hour.groupby('time_stamp').mean()[:-1] #sli
```

In [19]:
```
A_avg_clicks_by_hour = A_group_df[['sender_id', 'clicks', 'time_stamp']].copy()
A_avg_clicks_by_hour = A_avg_clicks_by_hour.groupby(
    [pd.Grouper(key='time_stamp', freq='H'), 'sender_id']
).sum()
A_avg_clicks_by_hour = A_avg_clicks_by_hour.groupby('time_stamp').mean()
```

In [20]:
```
fig, ax = plt.subplots()

A_avg_clicks_by_hour.plot(y='clicks', use_index=True, color=cmap[1], label='avg cl
B_avg_clicks_by_hour.plot(y='clicks', use_index=True, color=cmap[4], label='avg cl
ax.set_title("Average number of clicks per user during an A/B test")
ax.set_ylabel("clicks")
ax.set_xlabel("time")
```

Out[20]:
```
Text(0.5, 0, 'time')
```



Comment to the chart:

It's hard to see any change in the number of likes per person. In order to make sure that the differences are not significant or to refute this, you need to analyze the distributions of the average number of likes per user during the A / B test.

Consider p-value relative to 1%, 5% and 10% confidence intervals

With a confidence interval of 1.0%, we accept the main hypothesis. This means that changing the heart to a check mark will not affect the number of likes users give to the app. Changes in the number of likes for groups A (control) and B (test) are not really different and the observed differences are random. With a confidence interval of 5.0%, we accept the main hypothesis. This means that changing the heart to a check mark will not affect the number of likes users give to the app. Changes in the number of likes for groups A (control) and B (test) are not really different and the observed differences are random. With a confidence interval of 10.0%, we accept the main hypothesis. This means that changing the heart to a check mark will not affect the number of likes users give to the app. Changes in the number of likes for groups A (control) and B (test) are not really different and the observed differences are random.

Conclusions:

The innovation should be rejected. Changes in scores for groups A and B are random. In fact, a tick and a heart result in statistically the same number of likes.