Чтение данных из файлов

Таблица payments_CG1.csv

In [126]:

```
from scipy.stats import norm
import pandas as pd
import numpy as np
payments_CG1 = pd.read_csv('~/TZ_TopfaceMedia/TopfaceMedia/Data/payments_CG1.csv', delimiter=';')
payments_CG1 = payments_CG1.dropna(how='all')
payments_CG1
```

Out[126]:

	user_id	registration_date	payment_date	revenue	payment_service_id	place	country	product_type
0	891319275.0	05.11.2017 19:57	12.11.2017 21:57	12727.0	147.0	TrialVipPopup	FR	trial
1	892421826.0	14.10.2017 12:46	23.10.2017 12:54	14586.0	147.0	TrialVipPopup	IL	trial
2	891822480.0	26.10.2017 22:27	12.11.2017 20:59	1911.0	19.0	none	IT	coins
3	891367521.0	05.11.2017 2:37	06.11.2017 19:15	1456.0	127.0	sympathy_access	IT	other
4	892207959.0	18.10.2017 18:30	12.11.2017 12:03	1261.0	19.0	EmptyLikes	FR	coins
...	...	...	...	...	...	...	...	...
372	891556971.0	01.11.2017 5:07	01.11.2017 5:16	2886.0	68.0	none	FR	prem
373	892052106.0	22.10.2017 1:38	29.10.2017 3:44	12883.0	147.0	TrialVipPopup	ES	trial
374	892145004.0	20.10.2017 4:25	01.11.2017 2:56	9919.0	147.0	TrialVipPopup	CL	trial
375	891458229.0	03.11.2017 12:12	04.11.2017 3:43	6305.0	19.0	Menu	ES	coins
376	891466989.0	03.11.2017 7:08	03.11.2017 7:56	6292.0	147.0	AutoReplyMessage	US	prem

377 rows × 8 columns

Таблица payments_CG2.csv

In [127]:

```
import pandas as pd
import numpy as np
payments_CG2 = pd.read_csv('~/TZ_TopfaceMedia/TopfaceMedia/Data/payments_CG2.csv', delimiter=';')
payments_CG2 = payments_CG2.dropna(how='all')
payments_CG2
```

Out[127]:

	user_id	registration_date	payment_date	revenue	payment_service_id	place	country	product_type
0	891266616	06.11.2017 15:25	10.11.2017 17:35	6305	19	autorefill	AR	coins
1	892186737	19.10.2017 6:03	04.11.2017 8:53	4732	147	promo.key31	CL	prem
2	891959004	24.10.2017 0:12	31.10.2017 2:12	10153	147	TrialVipPopup	US	trial
3	892115478	20.10.2017 20:39	26.10.2017 6:26	2366	67	EmptyLikes	ES	other
4	891592941	31.10.2017 12:40	03.11.2017 16:25	6292	147	promo.key31	CH	prem
...	...	...	...	...	...	...	...	...
323	891985866	23.10.2017 13:19	30.10.2017 15:35	16640	147	TrialVipPopup	BE	trial
324	891491145	02.11.2017 18:13	09.11.2017 20:32	12727	147	SendMessage	US	trial
325	892053750	22.10.2017 0:46	28.10.2017 8:17	1261	19	autorefill	US	coins
326	891668253	29.10.2017 23:08	15.11.2017 0:57	9997	147	TrialVipPopup	CL	trial
327	891459291	03.11.2017 11:43	03.11.2017 11:57	6292	147	AutoReplyMessage	ES	prem

328 rows × 8 columns

Таблица payments_TG.csv

In [128]:

```
import pandas as pd
import numpy as np
payments_TG = pd.read_csv('~/TZ_TopfaceMedia/TopfaceMedia/Data/payments_TG.csv', delimiter=';')
payments_TG = payments_TG.dropna(how='all')
# paments_TGy[['user_id']]
payments_TG
```

Out[128]:

```
       user_id  registration_date      payment_date  revenue  payment_service_id          place  country  product_type
  0  891345942   05.11.2017 13:00  13.11.2017 15:04    12909                 147  TrialVipPopup       IT         trial
  1  892054251    22.10.2017 0:33    30.10.2017 1:37    13923                 147  TrialVipPopup       US         trial
  2  892236423    18.10.2017 1:09    23.10.2017 0:15     3783                  67           Menu       US         other
  3  892236423    18.10.2017 1:09  27.10.2017 22:38     3783                  67           Menu       US         other
  4  892168170   19.10.2017 17:10  27.10.2017 19:10     9087                 147  TrialVipPopup       US         trial
...        ...                ...               ...      ...                 ...            ...      ...           ...
268  891419625    04.11.2017 5:58   11.11.2017 8:33    14352                 147  TrialVipPopup       IL         trial
269  891447105   03.11.2017 17:05   11.11.2017 4:23     7319                 147           Fans       GB          prem
270  891497073   02.11.2017 15:33   05.11.2017 4:06     2730                  67     EmptyLikes       US         other
271  891406224   04.11.2017 12:56  04.11.2017 13:35     2743                  67     EmptyLikes       IL         other
272  891499359   02.11.2017 14:25  02.11.2017 14:30     1599                  19           none       DE         coins

273 rows × 8 columns
```

Таблица portrait_CG1.csv

In [129]:

```
import pandas as pd
import numpy as np
portrait_CG1 = pd.read_csv('~/TZ_TopfaceMedia/TopfaceMedia/Data/portrait_CG1.csv', delimiter=';')
portrait_CG1 = portrait_CG1.dropna(how='all')
portrait_CG1
```

Out[129]:

```
        user_id  age  attraction_distribution  coins  country                         retention_days  sex  user_dating.age_start  user_dating.age_end  view_count
   0  892319115   25                      435    NaN       US                            1,2,3,4,5,6    0                     22                   32         982
   1  891248523   29                      500    NaN       US                                    1,2    1                     26                   32          12
   2  891670932   33                      800    NaN       FR                  1,2,3,5,6,8,9,11,13,16    1                     30                   36           5
   3  891060786   26                        0    NaN       AR                              1,2,3,4,5    1                     23                   29           0
   4  892006554   35                      294    NaN       US         1,2,3,5,6,7,8,9,10,12,15,16,17,19    1                     30                   40          17
 ...        ...  ...                      ...    ...      ...                                    ...  ...                    ...                  ...         ...
4335  892022304   26                      227    NaN       US                                    NaN    0                     23                   29         427
4336  891905022   38                      200    NaN       AR  1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,21    1                     20                   50           5
4337  891304722   49                      411    NaN       US                                      1    1                     46                   52          17
4338  891302328   27                        0    NaN       US                                  1,3,5    1                     24                   30           0
4339  891182532   31                     1000    NaN       US                                      1    1                     19                   34           2

4340 rows × 13 columns
```

Таблица portrait_CG2.csv

In [130]:

```
import pandas as pd
import numpy as np
portrait_CG2 = pd.read_csv('~/TZ_TopfaceMedia/TopfaceMedia/Data/portrait_CG2.csv', delimiter=';')
portrait_CG2 = portrait_CG2.dropna(how='all')
portrait_CG2
```

Out[130]:

	user_id	age	attraction_distribution	coins	country	retention_days	sex	user_dating.age_start	user_dating.age_end	view_count	was_s
0	892035504	37	137	NaN	IL	1,2,3,4,5,6,7,8,9,11,13	1	30	40	51	
1	891782112	57	0	NaN	IT	1	1	54	60	0	
2	891110337	30	769	NaN	FR	1,2,3,4,5	1	27	33	13	
3	891796320	48	750	NaN	FR	1,4,6,8,10,11,12,14,15,16,18	1	45	51	12	
4	891880212	54	638	NaN	US	1	1	35	53	94	
...	...	...	...	...	...	...	...	...	...	...	
4259	891937401	63	242	NaN	IT	1,2	1	50	66	33	
4260	891692844	35	500	NaN	FR	1,2,3,4,5,6,8,9,10,11,12,14,16	1	24	45	2	
4261	891096429	41	333	NaN	US	NaN	1	38	44	6	
4262	891697584	48	210	NaN	GB	2,3,5,7,9,10,12,14,15	1	45	51	19	
4263	891095634	22	0	NaN	IT	NaN	1	19	25	0	

4264 rows × 13 columns

Таблица portrait_TG.csv

In [131]:

```
import pandas as pd
import numpy as np
portrait_TG = pd.read_csv('~/TZ_TopfaceMedia/TopfaceMedia/Data/portrait_TG.csv', delimiter=';')
portrait_TG = portrait_TG.dropna(how='all')
portrait_TG
```

Out[131]:

	user_id	age	attraction_distribution	coins	country	retention_days	sex	user_dating.age_start	user_dating.age_end	vie
0	892309896	27	685	NaN	US	1,2,3,4,5,6,7,8,9,10,11,12,16,17	1	24	30	
1	892044516	27	0	NaN	DE	NaN	1	24	30	
2	892185708	45	44	NaN	IL	1,3,4,6,7,8,10,11,12,13,14,15,16,17,18,19,20,2...	1	42	48	
3	892130292	32	0	NaN	US	1,2,3,4,5,6,7,8,9,10,11,12	1	29	35	
4	891406035	27	1000	NaN	FR	NaN	1	24	30	
...	...	...	...	...	...	...	...	...	...	
4303	891078639	17	895	NaN	US	1,2,3,4,5	1	16	20	
4304	891188733	27	866	NaN	US	1,2,3	1	18	41	
4305	891293718	48	0	NaN	AR	NaN	1	47	51	
4306	891192357	32	0	NaN	AU	NaN	1	16	49	
4307	891668910	35	575	NaN	AE	1	1	16	99	

4308 rows × 13 columns

Решение

1. Процент пользователей CG1 с премиум подпиской

In [132]:

```
p1 = payments_CG1[payments_CG1.product_type == 'prem'].count()[0]/payments_CG1.count()[0]*100
p1
```

Out[132]:

```
28.116710875331563
```

Процент пользователей CG2 с премиум подпиской

In [133]:

```
p2 = payments_CG2[payments_CG2.product_type == 'prem'].count()[0]/payments_CG2.count()[0]*100
```

In [134]:

```
p2
```

Out[134]:

```
32.6219512195122
```

Процент пользователей TG с премиум подпиской

In [135]:

```
p3 = payments_TG[payments_TG.product_type == 'prem'].count()[0]/payments_TG.count()[0]*100
```

In [136]:

```
p3
```

Out[136]:

```
27.83882783882784
```

In [137]:

```
p = (payments_CG2[payments_CG2.product_type == 'prem'].count()[0] + payments_TG[payments_TG.product_type == 'prem'].count()[0])/(payments_CG2.c
```

In [138]:

```
z = (p3-p2)/(p*(1-p)*((1/payments_CG2.count()[0]*100)+(1/payments_TG.count()[0]*100))**0.5)
```

In [139]:

```
p_value = norm.cdf(z)
```

In [140]:

```
print(['{:.12f}'.format(a) for a in (abs(z), p_value*2)])
```

```
['1923.270573579465', '0.000000000000']
```

Примерно те же самые значения будут при проверке CG1 и TG. Однако, далее отметим:

p_value<0.001 и количество пользователей с премиум подпиской уменьшилось на 5%(было 32.6, стало 27.8). Таким образом, значение метрики кол-ва пользователей с премиум подпиской уменьшилось

2. Процент пользователей с пробной премиум подпиской

In [141]:

```
# Решение


# Процент пользователей CG2 с пробной премиум подпиской

p2 = payments_CG2[payments_CG2.product_type == 'trial'].count()[0]/payments_CG2.count()[0]*100

print(p2)

# Процент пользователей TG с премиум подпиской

p3 = payments_TG[payments_TG.product_type == 'trial'].count()[0]/payments_TG.count()[0]*100

print(p3)

p = (payments_CG2[payments_CG2.product_type == 'trial'].count()[0] + payments_TG[payments_TG.product_type == 'trial'].count()[0])/(payments_CG2.cou

z = (p3-p2)/(p*(1-p)*((1/payments_CG2.count()[0]*100)+(1/payments_TG.count()[0]*100))**0.5)

p_value = norm.cdf(z)

print(['{:.12f}'.format(a) for a in (abs(z), p_value*2)])
```

```
16.158536585365855
22.71062271062271
['4187.637182522381', '2.000000000000']
```

3. Процент пользователей с подпиской other

In [142]:

```
# Решение


# Процент пользователей CG2 с пробной премиум подпиской
```

```
p2 = payments_CG2[payments_CG2.product_type == 'other'].count()[0]/payments_CG2.count()[0]*100

print(p2)

# Процент пользователей TG с премиум подпиской

p3 = payments_TG[payments_TG.product_type == 'other'].count()[0]/payments_TG.count()[0]*100

print(p3)

p = (payments_CG2[payments_CG2.product_type == 'other'].count()[0] + payments_TG[payments_TG.product_type == 'other'].count()[0])/(payments_CG2.c

z = (p3-p2)/(p*(1-p)*((1/payments_CG2.count()[0]*100)+(1/payments_TG.count()[0]*100))**0.5)

p_value = norm.cdf(z)

print(['{:.12f}'.format(a) for a in (abs(z), p_value*2)])
```

```
13.414634146341465
10.622710622710622
['2809.079123923951', '0.000000000000']
```

Выручка в среднем на одного пользователя(ARPU)

In [143]:

```
## ДО повышения CG1
pay1 = payments_CG1['revenue'].sum()/payments_CG1.count()[4]
print('CG1:',pay1)

## ДО повышения CG2
pay2 = payments_CG2['revenue'].sum()/payments_CG2.count()[4]
print('CG2:',pay2)

## После повышения
pay3 = payments_TG['revenue'].sum()/payments_TG.count()[4]
print('TG:',pay3)
```

```
CG1: 6929.206896551724
CG2: 5854.993902439024
TG: 8589.380952380952
```

Видно, что после повышения тарифа выручка на одного пользователя заметно выросла. Это является существенным показателем

Количество посещений приложения

In [144]:

```
retention1 = portrait_CG1[['retention_days']].dropna(how='all') #Удалены строки со значениями NAN

df_marks1 = []

for i, row in retention1.iterrows():
    cmp = len(row['retention_days'].split(','))
    df_marks1.append(cmp)

sum1 = sum(df_marks1)
print('retention_CG1:', sum1)
```

```
retention_CG1: 20148
```

In [145]:

```
retention2 = portrait_CG2[['retention_days']].dropna(how='all') #Удалены строки со значениями NAN

df_marks2 = []

for i, row in retention2.iterrows():
    cmp = len(row['retention_days'].split(','))
    df_marks2.append(cmp)

sum2 = sum(df_marks2)
print('retention_CG2:', sum2)
```

```
retention_CG2: 18958
```

In [146]:

```
retention3 = portrait_TG[['retention_days']].dropna(how='all') #Удалены строки со значениями NAN
```

```
df_marks3 = []

for i, row in retention3.iterrows():
    cmp = len(row['retention_days'].split(','))
    df_marks3.append(cmp)

sum3 = sum(df_marks3)
print('retention_TG:', sum3)
```

```
retention_TG: 19754
```

Из этих данных можно сказать, что количество посещений приложения практический не изменилось

In [ ]: