

+ Код + Текст

✓ ОЗУ | Диск | Редактирование

▼ Подготовка

```
[217] 1 # imports
      2 import numpy as np
      3 import pandas as pd
      4 import matplotlib
      5 import plotly.express as px
      6 from sqlalchemy import create_engine
      7 from datetime import datetime, timedelta
      8 from plotly.subplots import make_subplots
```

```
[218]: 1 # connection
 2 HOST = '37.139.42.145'
 3 DBNAME = 'game-analytics'
 4 USER = 'analytics'
 5 PASSWORD = 'BRtTaqYiJyr29WXN'
 6 TABLE_SCHEMA = 'data_viz_1068.project_dataset'
 7 engine = create_engine(f'postgresql://{{USER}}:{{PASSWORD}}@{{HOST}}/{{DBNAME}}')
```

```
[219] 1 # getting table from database
2 project_dataset = pd.read_sql(f"""
3     SELECT *
4     FROM {TABLE_SCHEMA}
5     """, con=engine)
6
7 # converting some types to datetime
8 project_dataset['event_date'] = pd.to_datetime(project_dataset['event_date'])
9 project_dataset['cohort_date'] = pd.to_datetime(project_dataset['cohort_date'])
10
11 # dropping full duplicates (if they exists)
12 project_dataset.drop_duplicates()
13
14 # printing
15 project_dataset.head(2)
```

	event_time	event_date	event_name	revenue_usd	region	country	device_type	platform	cohort_date	user_id	user_type	content_id	current_step
0	2021-04-11 04:40:35	2021-04-11	LaunchApp	0.0	EU	RU	iPad8,11	ios	2021-03-31	227648	organic	None	No
1	2021-04-11 04:40:59	2021-04-11	LaunchApp	0.0	AS	KR	iPhone 11 Pro Max	ios	2021-03-29	227316	organic	None	No



```
[220] 1 project dataset = project dataset.dropna(subset=['event time'])
```

▼ Дополнение основного датасета

```
[221] 1 project_dataset['region'] = (np.where(project_dataset['country'] == 'CIS', project_dataset['region']))
```

```
[222] 1 # adding install datetime
2 install_times = project_dataset[project_dataset['event_name']=='FirstLaunchApp']
3 install_times = install_times.drop_duplicates(subset=['user_id'])
4 install_times = install_times[['user_id', 'event_time']]
5 install_times = install_times.rename(columns={'event_time' : 'install_time'})
6 table_24 = project_dataset.merge(install_times, on='user_id')
7 table_24.head(1)
```

	event_time	event_date	event_name	revenue_usd	region	country	device_type	platform	cohort_date	user_id	user_type	content_id	current_page
0	2021-04-11 04:40:35	2021-04-11	LaunchApp	0.0	CIS	RU	iPad8,11	ios	2021-03-31	227648	organic	None	No



```
[223] 1 # adding living times
2 table_24 = table_24[table_24['event_time'] >= table_24['install_time']]
3 table_24['lifehour'] = ((table_24['event_time'] - table_24['install_time']).dt.total_seconds() / 3600).astype('int')
4 table_24['lifeday'] = (table_24['event_time'] - table_24['install_time']).dt.days
5 table_24.head(1)
6 #table_24 = table_24[table_24['event_time'] <= table_24['install_time'] + timedelta(hours=24)]
```

```
[224] 1 refined_df = table_24.copy()
2 refined_df['install_month'] = refined_df.install_time.dt.to_period('M')
3 display(refined_df.head(1))
```

	event_time	event_date	event_name	revenue_usd	region	country	device_type	platform	cohort_date	user_id	user_type	content_id	current
0	2021-04-11 04:40:35	2021-04-11	LaunchApp	0.0	CIS	RU	iPad8,11	ios	2021-03-31	227648	organic	None	No



1. Исследование процента платящих для новых, старых и очень старых игроков относительно конкретного месяца для каждой платформы, регионов NA и CIS

```
[225] 1 def CalculatePayersPercent(df):
2     if(df.shape[0] == 0): return 0
3     payers = df[df['revenue_usd'] > 0]['user_id'].nunique()
4     users = df['user_id'].nunique()
5     result = (payers / users) * 100
6     return result
7
8 def CalculateForAll(df, install_time_name):
9     result = pd.DataFrame([])
10    df = df.copy()
11    df['install_month'] = df[install_time_name].dt.to_period('M')
12    df = df.sort_values('install_month')
13    for initial_month in df['install_month'].unique():
14        line = pd.DataFrame(index=[initial_month])
15
16        new = df[df['install_month'] == initial_month]
17        old = df[df['install_month'] == initial_month - 1]
18        aged = df[df['install_month'] == initial_month - 2]
19
20        line['and_n'] = CalculatePayersPercent(new[new['platform'] == 'android'])
21        line['and_o'] = CalculatePayersPercent(old[old['platform'] == 'android'])
22        line['and_a'] = CalculatePayersPercent(aged[aged['platform'] == 'android'])
23
24        line['ios_n'] = CalculatePayersPercent(new[new['platform'] == 'ios'])
25        line['ios_o'] = CalculatePayersPercent(old[old['platform'] == 'ios'])
26        line['ios_a'] = CalculatePayersPercent(aged[aged['platform'] == 'ios'])
27
28        line['na_n'] = CalculatePayersPercent(new[new['region'] == 'NA'])
29        line['na_o'] = CalculatePayersPercent(old[old['region'] == 'NA'])
30        line['na_a'] = CalculatePayersPercent(aged[aged['region'] == 'NA'])
31
32        line['cis_n'] = CalculatePayersPercent(new[new['region'] == 'CIS'])
33        line['cis_o'] = CalculatePayersPercent(old[old['region'] == 'CIS'])
34        line['cis_a'] = CalculatePayersPercent(aged[aged['region'] == 'CIS'])
35
36        result = result.append(line)
37    return result
```

```
1 def CalculateForAllSimplier(df, install_time_name):
2     result = pd.DataFrame([])
3     df = df.copy()
4     df['install_month'] = df[install_time_name].dt.to_period('M')
5     df = df.sort_values('install_month')
6     for initial_month in df['install_month'].unique():
7         line = pd.DataFrame(index=[initial_month])
8
9         new = df[df['install_month'] == initial_month]
10
11         line['and_n'] = CalculatePayersPercent(new[new['platform'] == 'android'])
12         line['ios_n'] = CalculatePayersPercent(new[new['platform'] == 'ios'])
13         line['na_n'] = CalculatePayersPercent(new[new['region'] == 'NA'])
14         line['cis_n'] = CalculatePayersPercent(new[new['region'] == 'CIS'])
```



```
15
16     result = result.append(line)
17
18
```

✓ [227] 1 final_df = CalculateForAll(refined_df, 'install_time')
2 pd.options.display.precision = 2

✓ [228] 1 simpplier_df = CalculateForAllSimplifier(refined_df, 'install_time')

✓ [229] 1 display(final_df.style.background_gradient(cmap='PuBu', axis=None))

	and_n	and_o	and_a	ios_n	ios_o	ios_a	na_n	na_o	na_a	cis_n	cis_o	cis_a
2020-10	18.18	0.00	0.00	4.66	0.00	0.00	6.26	0.00	0.00	3.22	0.00	0.00
2020-11	4.44	18.18	0.00	5.84	4.66	0.00	8.45	6.26	0.00	3.20	3.22	0.00
2020-12	4.20	4.44	18.18	7.12	5.84	4.66	10.42	8.45	6.26	3.26	3.20	3.22
2021-01	4.38	4.20	4.44	5.82	7.12	5.84	8.33	10.42	8.45	3.11	3.26	3.20
2021-02	3.20	4.38	4.20	5.38	5.82	7.12	8.52	8.33	10.42	2.17	3.11	3.26
2021-03	3.62	3.20	4.38	4.26	5.38	5.82	8.35	8.52	8.33	3.21	2.17	3.11
2021-04	3.74	3.62	3.20	3.42	4.26	5.38	8.45	8.35	8.52	2.60	3.21	2.17
2021-05	1.72	3.74	3.62	4.17	3.42	4.26	5.73	8.45	8.35	2.11	2.60	3.21

Среди всех когорт ярко выделяется процент платящих у android аудитории. Посмотрим сколько это человек относительно следующего по величине процента платящих (NA, 2020-12)

✓ [230] 1 refined_df[(refined_df['install_month'].astype('str')=='2020-10') & (refined_df['platform']=='android')['user_id'].nunique()]
11

✓ [231] 1 refined_df[(refined_df['install_month'].astype('str')=='2020-12') & (refined_df['region']=='NA')['user_id'].nunique()]
2111

Значение отличается на 2 порядка. Кажется что в 2020-10 на андроид платформе было слишком мало пользователей.

А если вспомнить что всплеск андроид пользователей был только в 2020-11.

То можно предположить что эти 11 человек были тестерами и можно отбросить данные по ним чтобы они не искали градиентное колорирование других ячеек.

✓ [232] 1 final_df.loc[['2020-10'], ['and_n']] = 0
2 final_df.loc[['2020-11'], ['and_o']] = 0
3 final_df.loc[['2020-12'], ['and_a']] = 0
4
5 simpplier_df.loc[['2020-10'], ['and_n']] = 0

▼ Процент платящих для всех когорт для всех месяцев (n=new, o=old, a=aged) (там где нули - данных нет/мало)

✓ [239] 1 from IPython.display import display_html
2
3 df1_styler = final_df.style.background_gradient(cmap='coolwarm', axis=None)
4 df2_styler = simpplier_df.style.background_gradient(cmap='coolwarm', axis=None).set_table_attributes("style='display:inline'")
5 df3_styler = simpplier_df.style.background_gradient(cmap='coolwarm', axis=1).set_table_attributes("style='display:inline'")
6 df4_styler = simpplier_df.style.background_gradient(cmap='coolwarm', axis=0).set_table_attributes("style='display:inline'")
7
8 display_html(df1_styler._repr_html_()+df2_styler._repr_html_()+df3_styler._repr_html_()+df4_styler._repr_html_(), raw=True)

	and_n	and_o	and_a	ios_n	ios_o	ios_a	na_n	na_o	na_a	cis_n	cis_o	cis_a
2020-10	0.00	0.00	0.00	4.66	0.00	0.00	6.26	0.00	0.00	3.22	0.00	0.00
2020-11	4.44	0.00	0.00	5.84	4.66	0.00	8.45	6.26	0.00	3.20	3.22	0.00
2020-12	4.20	4.44	0.00	7.12	5.84	4.66	10.42	8.45	6.26	3.26	3.20	3.22
2021-01	4.38	4.20	4.44	5.82	7.12	5.84	8.33	10.42	8.45	3.11	3.26	3.20
2021-02	3.20	4.38	4.20	5.38	5.82	7.12	8.52	8.33	10.42	2.17	3.11	3.26
2021-03	3.62	3.20	4.38	4.26	5.38	5.82	8.35	8.52	8.33	3.21	2.17	3.11
2021-04	3.74	3.62	3.20	3.42	4.26	5.38	8.45	8.35	8.52	2.60	3.21	2.17
2021-05	1.72	3.74	3.62	4.17	3.42	4.26	5.73	8.45	8.35	2.11	2.60	3.21

	and_n	ios_n	na_n	cis_n	and_n	ios_n	na_n	cis_n	and_n	ios_n	na_n	cis_n		
2020-10	0.00	4.66	6.26	3.22	2020-10	0.00	4.66	6.26	3.22	2020-10	0.00	4.66	6.26	3.22

2020-11	4.44	5.84	8.45	3.20	2020-11	4.44	5.84	8.45	3.20
2020-12	4.20	7.12	10.42	3.26	2020-12	4.20	7.12	10.42	3.26
2021-01	4.38	5.82	8.33	3.11	2021-01	4.38	5.82	8.33	3.11
2021-02	3.20	5.38	8.52	2.17	2021-02	3.20	5.38	8.52	2.17
2021-03	3.62	4.26	8.35	3.21	2021-03	3.62	4.26	8.35	3.21
2021-04	3.74	3.42	8.45	2.60	2021-04	3.74	3.42	8.45	2.60
2021-05	1.72	4.17	5.73	2.11	2021-05	1.72	4.17	5.73	2.11

NA

Лидером среди всех когорт является NA аудитория особенно в 2020-12 (рождество повлияло?), да и вообще среди всех месяцев NA колонки самые яркие и стабильные. 2020-10 метрика была почти самой маленькой. (почему? это был старт проекта и монетизация еще плохо работала?). В Мае метрика просела еще хуже.

iOS

Далее идет iOS платформа. Пик - 2020-12 и один из минимумов в 2020-10. Наблюдается кореляция между пиками и минимумами на iOS и у NA аудитории. Самым неудачным для iOS был 2021-04. Да и Май тоже плох.

Android

Тут метрика была самой высокой в Ноябре-Январе (запуск, Новый Год). Дальше метрика плавномерно опускалась. И свелась к минимуму в Мае. В целом у андроида картина похожа на CIS.

CIS

Для CIS процент платящих самый низкий и достиг пика 2020-012 (новый год повлиял). Однако уж в Феврале обнаружился резкий спад почти до минимума. Потом подъем. Далее спад. Май опять самый низкий.

Минимум у всех платформ - 2021-05 (у iOS 2021-04). Пришла Весна?. Или разработчики уже забили на игру?.

Почему у более старых игроков процент платящих отличается от более новых?

Потому что каждый месяц не похож на предыдущий. (Праздники, время года и т.д)

Потому что с каждым новым месяцем игра стареет. В игру приходит когорта которая уже менее заинтересована в игре и с меньшей вероятностью будет платить.

Потому что в одном из месяцев мог быть какой то новый апдейт который что то меняет что затрагивает процент платящих в этом или в последующем месяце.