

Оценка эффективности рекламных компаний

Используемые библиотеки

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Загрузка данных

```
items = pd.read_csv(r'D:\down2\Тестовое задание для аналитика в отдел
Закупок - Товары.csv')
df
```

	Номер доставки	Номер заказа	Товар	Цена за шт	Количество
\					
0	ZBP240603022	104П	1019843177275	37	1000
1	BZK240626169	104G	1019109333952	39	3000
2	ZBP240603022	106П	1019912111340	37	500
3	ZBP240603022	106П	1019885674196	37	1000
4	ZBP240603022	106П	1019912234617	37	1000
..
814	K472-LB350-16	147П	1019566284619	52	200
815	K472-LB350-16	147П	1019566284066	52	300
816	K472-LB350-16	148П	1019793358135	41	2000
817	Z25241115990	148П	1021912235812	33	3000
818	Z25241115990	150П	1019285887327	25	1000

	Вес товара гр
0	9,8
1	29
2	16,4
3	14,9
4	13,2
..	...
814	3,7

815	3,7
816	18,7
817	45,9
818	25,9

[819 rows x 6 columns]

Пропущенные id

```
registrations.loc[registrations['account_id'].isna(), 'account_id'] =
3842380.0 + registrations.index[registrations['account_id'].isna()]
registrations['account_id'] =
registrations['account_id'].astype('int')
```

```
persents = pd.read_csv(r'D:\down2\testovoe\persents.csv', index_col=0)
persents
```

	Payment_types	Share, %
0	Payment_system_1	20.7
1	Payment_system_2	15.0
2	Payment_system_3	58.5
3	Payment_system_4	0.0
4	Payment_system_5	24.0

```
payments = pd.read_csv(r'D:\down2\testovoe\payments.csv')
payments
```

	payment_date	Payment_types	real_cost	account_id
0	2021-01-01 00:00:56	Payment_system_5	9.650000	2907221
1	2021-01-01 00:01:48	Payment_system_5	0.955872	3228373
2	2021-01-01 00:03:42	Payment_system_5	3.026930	318552
3	2021-01-01 00:04:27	Payment_system_2	99.990000	3832817
4	2021-01-01 00:06:41	Payment_system_5	1.432550	7229767
...
303609	2021-12-31 23:56:20	Payment_system_5	2.165630	4333575
303610	2021-12-31 23:57:05	Payment_system_5	47.111000	1129267
303611	2021-12-31 23:57:09	Payment_system_5	47.111000	583846
303612	2021-12-31 23:58:09	Payment_system_2	30.100000	4595502
303613	2021-12-31 23:58:46	Payment_system_1	47.111000	3405166

[303614 rows x 4 columns]

```
budget = pd.read_excel(r'D:\down2\testovoe\Рекламный бюджет январь
2021.xlsx', sheet_name=0) # Загрузка первого листа
budget
```

	media_source	Campaign_type	Target	Plan, USD	Installs
0	Media_source_1	install	Brazil	100	10460
1	NaN	purchase	Brazil	400	1081

2	NaN	install	Russia	200	11894
3	NaN	purchase	Russia	500	1441
4	NaN	install	Ukraine	100	6424
5	NaN	purchase	Ukraine	500	242
6	Media_source_2	install	Brazil	100	152
7	NaN	install	English_speaking	100	371
8	Total	NaN	NaN	2000	32065

	Spend, USD	Deviation, USD
0	99.12	0.88
1	398.67	1.33
2	246.30	-46.30
3	616.77	-116.77
4	99.43	0.57
5	348.13	151.87
6	81.15	18.85
7	99.95	0.05
8	1989.52	10.48

Оптимальный срок окупаемости рекламных кампаний

Рассчитаем сколько принесли клиенты за январь с момента регистрации.

```
payments.rename(columns={'account_id': 'account_id_payments'},
inplace=True)
registrations.rename(columns={'account_id':
'account_id_registrations'}, inplace=True)

merged_df = pd.merge(payments, registrations,
left_on='account_id_payments', right_on='account_id_registrations',
how='inner')
merged_df = pd.merge(merged_df, persents, how='left',
on='Payment_types')

merged_df['created_date'] = pd.to_datetime(merged_df['created_date'])
merged_df['payment_date'] = pd.to_datetime(merged_df['payment_date'])
```

Остались только пользователи зарегистрированные в январе, которые совершили платежи.

```
merged_df['period'] = (merged_df['payment_date'] -
merged_df['created_date']).dt.days
```

```
merged_df['profit'] = merged_df['real_cost']*(1-merged_df['Share, %']*0.01)
merged_df
```

	payment_date	Payment_types	real_cost
account_id_payments \			
0	2021-01-01 05:50:37	Payment_system_5	3.026930
3843059			
1	2021-01-01 05:54:59	Payment_system_5	0.955872
3843059			
2	2021-01-01 05:55:11	Payment_system_5	0.955872
3843059			
3	2021-01-03 15:08:59	Payment_system_5	0.955872
3843059			
4	2021-01-11 10:47:13	Payment_system_5	0.955872
3843059			
...
...			
7335	2021-12-28 11:27:51	Payment_system_4	0.440000
3850701			
7336	2021-12-28 21:31:15	Payment_system_5	3.370000
3878621			
7337	2021-12-28 21:37:50	Payment_system_5	1.100000
3878621			
7338	2021-12-30 03:16:35	Payment_system_4	0.090000
3845607			
7339	2021-12-31 16:58:15	Payment_system_5	99.990000
3918249			

	account_id_registrations	created_date	campaign	Share, %	period
\					
0	3843059	2021-01-01	NaN	24.0	0
1	3843059	2021-01-01	NaN	24.0	0
2	3843059	2021-01-01	NaN	24.0	0
3	3843059	2021-01-01	NaN	24.0	2
4	3843059	2021-01-01	NaN	24.0	10
...
7335	3850701	2021-01-03	NaN	0.0	359
7336	3878621	2021-01-12	NaN	24.0	350
7337	3878621	2021-01-12	NaN	24.0	350
7338	3845607	2021-01-01	NaN	0.0	363

7339		3918249	2021-01-26	NaN	24.0	339
------	--	---------	------------	-----	------	-----

	profit
0	2.300467
1	0.726463
2	0.726463
3	0.726463
4	0.726463
...	...
7335	0.440000
7336	2.561200
7337	0.836000
7338	0.090000
7339	75.992400

[7340 rows x 10 columns]

Ошибка в данных, дата первого платежа до регистрации. (Может и не ошибка, но объединил я правильно айди сопадают..., не ясно)

```
merged_df.drop(merged_df[merged_df['created_date']>merged_df['payment_date']].index, inplace=True)
```

Оставим только клиентов пришедших с рекламных компаний

```
merged_df = merged_df[merged_df['campaign'].isna()==False]
merged_df
```

	payment_date	Payment_types	real_cost
account_id_payments \			
22 3861811	2021-03-26 17:35:51	Payment_system_5	23.53140
23 3861811	2021-04-05 13:49:08	Payment_system_5	5.87268
24 3861811	2021-04-05 13:55:38	Payment_system_5	2.34201
25 3861811	2021-09-04 00:17:51	Payment_system_5	2.30912
26 3861811	2021-09-16 11:42:51	Payment_system_5	23.19560
...
...			
7246 3852582	2021-11-05 01:48:07	Payment_system_4	0.22000
7247 3852582	2021-11-14 12:45:58	Payment_system_4	2.62000
7287 3922486	2021-11-13 23:40:26	Payment_system_4	0.17000

```
7289 2021-11-15 00:56:30 Payment_system_4 0.09000
3914543
7307 2021-11-23 02:27:40 Payment_system_5 4.51369
3861124
```

	account_id_registrations	created_date	campaign	Share,
% \				
22	3861811	2021-01-07	RUS_MS1_purchase	
24.0				
23	3861811	2021-01-07	RUS_MS1_purchase	
24.0				
24	3861811	2021-01-07	RUS_MS1_purchase	
24.0				
25	3861811	2021-01-07	RUS_MS1_purchase	
24.0				
26	3861811	2021-01-07	RUS_MS1_purchase	
24.0				
...
.				
7246	3852582	2021-01-04	UKR_MS1_install	
0.0				
7247	3852582	2021-01-04	UKR_MS1_install	
0.0				
7287	3922486	2021-01-28	RUS_MS1_purchase	
0.0				
7289	3914543	2021-01-24	UKR_MS1_install	
0.0				
7307	3861124	2021-01-06	BRA_MS2_install	
24.0				

	period	profit
22	78	17.883864
23	88	4.463237
24	88	1.779928
25	240	1.754931
26	252	17.628656
...
7246	305	0.220000
7247	314	2.620000
7287	289	0.170000
7289	295	0.090000
7307	321	3.430404

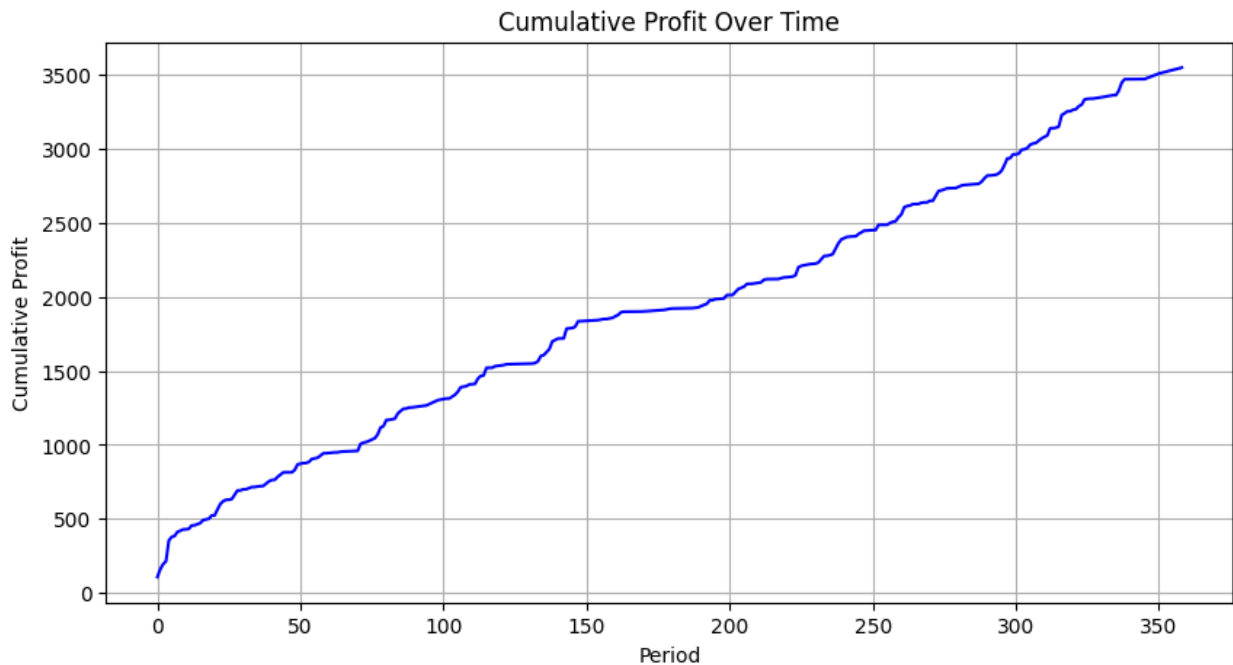
```
[693 rows x 10 columns]
```

```
cumulative_payments = merged_df.groupby('period')
['profit'].sum().cumsum().reset_index()
```

```
# Построение графика
plt.figure(figsize=(10, 5))
```

```
plt.plot(cumulative_payments['period'], cumulative_payments['profit'],
color='b')

# Добавление заголовка и меток осей
plt.title('Cumulative Profit Over Time')
plt.xlabel('Period')
plt.ylabel('Cumulative Profit')
plt.grid()
```



Новые пользователи со всех каналов довольно быстро начинают приносить деньги. С течением компании количество пользователей стабильно растёт.

budget

	media_source	Campaign_type	Target	Plan, USD	Installs
0	Media_source_1	install	Brazil	100	10460
1	NaN	purchase	Brazil	400	1081
2	NaN	install	Russia	200	11894
3	NaN	purchase	Russia	500	1441
4	NaN	install	Ukraine	100	6424
5	NaN	purchase	Ukraine	500	242
6	Media_source_2	install	Brazil	100	152

7	NaN	install	English_speaking	100	371
8	Total	NaN	NaN	2000	32065

	Spend, USD	Deviation, USD
0	99.12	0.88
1	398.67	1.33
2	246.30	-46.30
3	616.77	-116.77
4	99.43	0.57
5	348.13	151.87
6	81.15	18.85
7	99.95	0.05
8	1989.52	10.48

```
cumulative_payments[cumulative_payments['profit']>1989.52]
```

	period	profit
147	199	2010.945704
148	201	2011.565301
149	203	2051.939543
150	205	2067.784631
151	206	2085.987619
..
259	338	3468.335270
260	344	3468.955081
261	345	3469.576687
262	350	3507.219287
263	358	3546.896826

[117 rows x 2 columns]

Затраты на все рекламные компании суммированы начинают окупаться на 199 день, это до вычета налогов на прибыль, поэтому округлим до 7 месяцев.

Наиболее успешные рекламные компании на момент оценки окупаемости

```
campaign_profit =
merged_df[merged_df['period']<212].groupby('campaign')
['profit'].sum().reset_index().sort_values(by='profit',ascending=False)
campaign_profit
```

	campaign	profit
6	RUS_MS1_purchase	516.041879
8	UKR_MS1_purchase	453.236576
1	BRA_MS1_purchase	444.513184


```

5    RUS_MS1_install  325.681555
0    BRA_MS1_install  200.040209
7    UKR_MS1_install   67.437764
4    PR_youtube_blogger 53.108910
3    ENG_MS2_install   25.626667
2    BRA_MS2_install   11.774035

```

```

budget['campaign'] = ['BRA_MS1_install', 'BRA_MS1_purchase',
'BUS_MS1_install', 'BUS_MS1_purchase', 'UKR_MS1_install',
'UKR_MS1_purchase', 'BRA_MS2_install', 'ENG_MS2_install',
campaign_profit['profit'].sum())
campaign = pd.merge(budget, campaign_profit, on='campaign')
campaign['revenue'] = campaign['profit'] - campaign['Spend, USD']
campaign.sort_values(by='revenue', ascending=False)

```

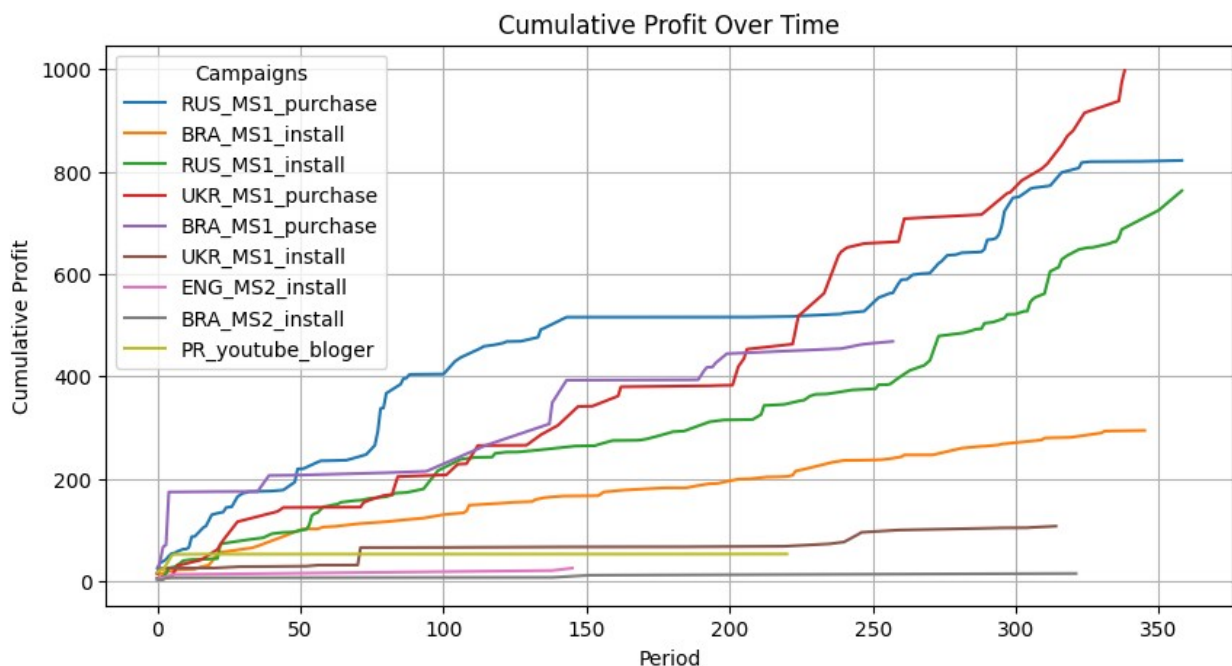
	media_source	Campaign_type	Target	Plan, USD	Installs
5	NaN	purchase	Ukraine	500	242
0	Media_source_1	install	Brazil	100	10460
2	NaN	install	Russia	200	11894
1	NaN	purchase	Brazil	400	1081
4	NaN	install	Ukraine	100	6424
6	Media_source_2	install	Brazil	100	152
7	NaN	install	English-speaking	100	371
3	NaN	purchase	Russia	500	1441

	Spend, USD	Deviation, USD	campaign	profit
revenue				
5	348.13	151.87	UKR_MS1_purchase	453.236576
105.106576				
0	99.12	0.88	BRA_MS1_install	200.040209
100.920209				
2	246.30	-46.30	RUS_MS1_install	325.681555
79.381555				
1	398.67	1.33	BRA_MS1_purchase	444.513184
45.843184				
4	99.43	0.57	UKR_MS1_install	67.437764
-31.992236				
6	81.15	18.85	BRA_MS2_install	11.774035
-69.375965				
7	99.95	0.05	ENG_MS2_install	25.626667
-74.323333				

```
3      616.77      -116.77  RUS_MS1_purchase  516.041879 -  
100.728121
```

Видно что некоторые компании уже начали приносить прибыль, остальные же пока немного отстают, необходимо понять причину и проследить динамику.

```
plt.figure(figsize=(10, 5))  
for comp in merged_df['campaign'].unique():  
    cp1 = merged_df[merged_df['campaign'] == comp].groupby('period')  
    ['profit'].sum().cumsum().reset_index()  
    plt.plot(cp1['period'], cp1['profit'], label=comp) # Добавляем  
метку для легенды  
  
# Добавление заголовка и меток осей  
plt.title('Cumulative Profit Over Time')  
plt.xlabel('Period')  
plt.ylabel('Cumulative Profit')  
plt.grid()  
plt.legend(title='Campaigns');
```



Подводя итоги можно сделать вывод о том, что наиболее успешными были рекламные компании UKR_MS1_purchase, BRA_MS1_install, RUS_MS1_install. Различные типы рекламных компаний показывают похожую эффективность, а наиболее прибыльной рекламной сетью стала MS1.

Average revenue per user

```
successful_comp =
merged_df[merged_df['campaign'].isin(['UKR_MS1_purchase', 'BRA_MS1_inst
all', 'RUS_MS1_install', 'BRA_MS1_purchase'])]
successful_comp
```

	payment_date	Payment_types	real_cost
account_id_payments \			
58 2021-01-01 17:16:09	Payment_system_5	9.624570	
3844647			
59 2021-01-01 17:17:45	Payment_system_5	9.624570	
3844647			
168 2021-03-06 09:06:49	Payment_system_5	4.760000	
3922908			
169 2021-03-27 09:13:53	Payment_system_5	4.710000	
3922908			
170 2021-05-01 12:53:49	Payment_system_5	4.650000	
3922908			
...
...			
7220 2021-10-25 16:33:42	Payment_system_4	43.610000	
3915834			
7227 2021-10-31 00:00:21	Payment_system_4	0.170000	
3920629			
7238 2021-11-03 18:00:07	Payment_system_5	2.658150	
3910603			
7239 2021-11-03 18:01:35	Payment_system_5	0.839416	
3910603			
7240 2021-11-15 10:33:49	Payment_system_5	4.873850	
3910603			

	account_id_registrations	created_date	campaign	Share, %
\				
58	3844647	2021-01-01	BRA_MS1_install	24.0
59	3844647	2021-01-01	BRA_MS1_install	24.0
168	3922908	2021-01-28	RUS_MS1_install	24.0
169	3922908	2021-01-28	RUS_MS1_install	24.0
170	3922908	2021-01-28	RUS_MS1_install	24.0
...
7220	3915834	2021-01-25	RUS_MS1_install	0.0
7227	3920629	2021-01-27	RUS_MS1_install	0.0
7238	3910603	2021-01-23	RUS_MS1_install	24.0

7239	3910603	2021-01-23	RUS_MS1_install	24.0
7240	3910603	2021-01-23	RUS_MS1_install	24.0

	period	profit
58	0	7.314673
59	0	7.314673
168	37	3.617600
169	58	3.579600
170	93	3.534000
...
7220	273	43.610000
7227	277	0.170000
7238	284	2.020194
7239	284	0.637956
7240	296	3.704126

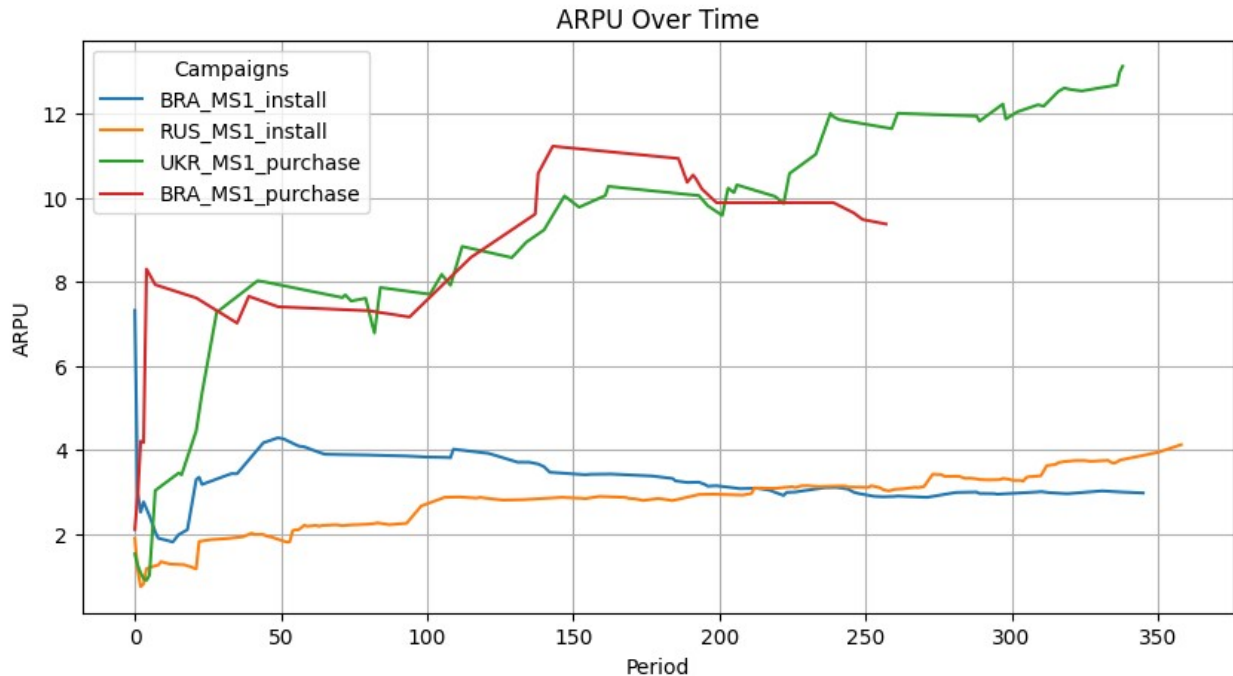
[410 rows x 10 columns]

```
plt.figure(figsize=(10, 5))
for comp in successful_comp['campaign'].unique():
    cum_val = successful_comp[successful_comp['campaign'] ==
comp].groupby('period')['profit'].sum().cumsum().reset_index()
    cum_val['cum_count'] =successful_comp[successful_comp['campaign']
== comp].groupby('period')['profit'].count().cumsum().reset_index()
    ['profit']
    cum_val['ARPU'] = cum_val['profit']/cum_val['cum_count']
    plt.plot(cum_val['period'], cum_val['ARPU'], label=comp) #
```

Добавляем метку для легенды

Добавление заголовка и меток осей

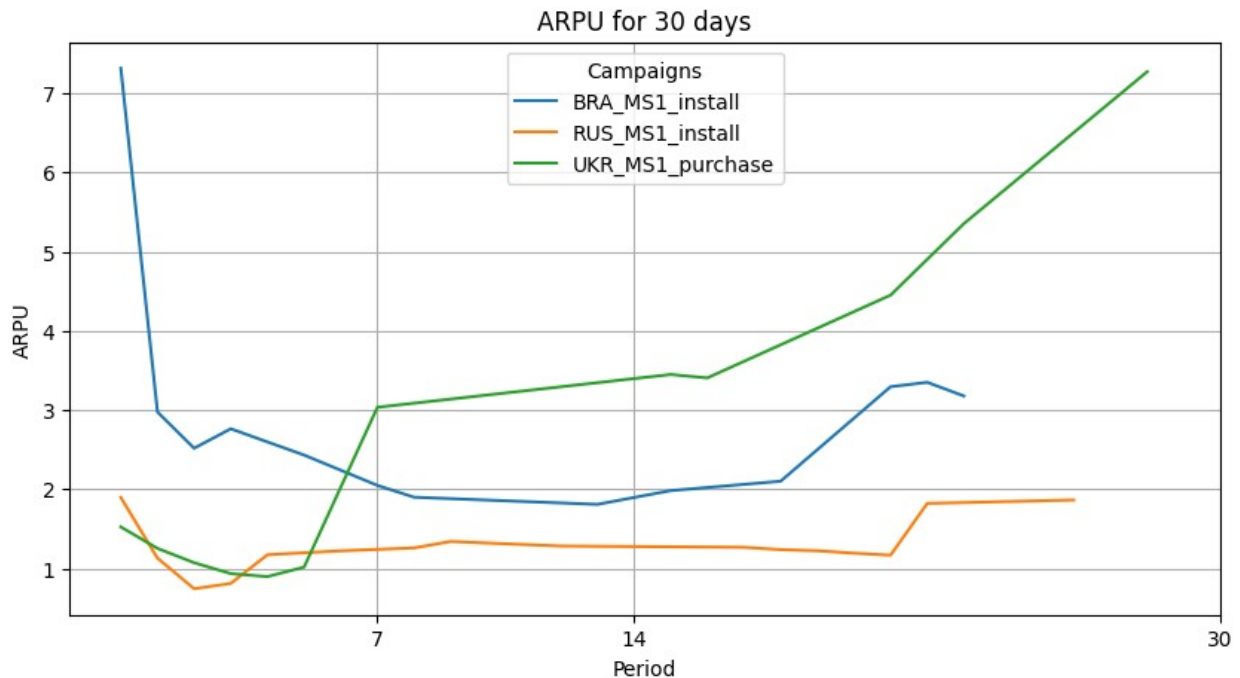
```
plt.title('ARPU Over Time')
plt.xlabel('Period')
plt.ylabel('ARPU')
plt.grid()
plt.legend(title='Campaigns');
```



Как видно по графику не во всех рекламных компаниях с увеличением срока жизни аккаунта увеличивается прибыль с пользователя. В начале срока жизни аккаунтов наблюдается резкий скачек ARPU, однако далее ARPU выравнивается. Дальнейшее поведение пользователей разных регионов и компаний различается и должно быть проанализировано. Рассмотрим промежутки 7-14-30 дней с момента регистрации.

```
plt.figure(figsize=(10, 5))
for comp in successful_comp['campaign'].unique():
    cum_val = successful_comp[(successful_comp['campaign'] ==
comp)&(successful_comp['period']<=31)].groupby('period')
['profit'].sum().cumsum().reset_index()
    cum_val['cum_count'] =
successful_comp[(successful_comp['campaign'] ==
comp)&(successful_comp['period']<=30)].groupby('period')
['profit'].count().cumsum().reset_index()['profit']
    cum_val['ARPU'] = cum_val['profit']/cum_val['cum_count']
    plt.plot(cum_val['period'], cum_val['ARPU'], label=comp) #
Добавляем метку для легенды

# Добавление заголовка и меток осей
plt.title(f'ARPU for 30 days')
plt.xlabel('Period')
plt.ylabel('ARPU')
plt.xticks([7, 14, 30])
plt.grid()
plt.legend(title='Campaigns');
```



Количество пользователей в начале графика невелико, это может значительно влиять на устойчивость графика ARPU. В целом наблюдается увеличение значения ARPU к 30 дню.

Cost per install

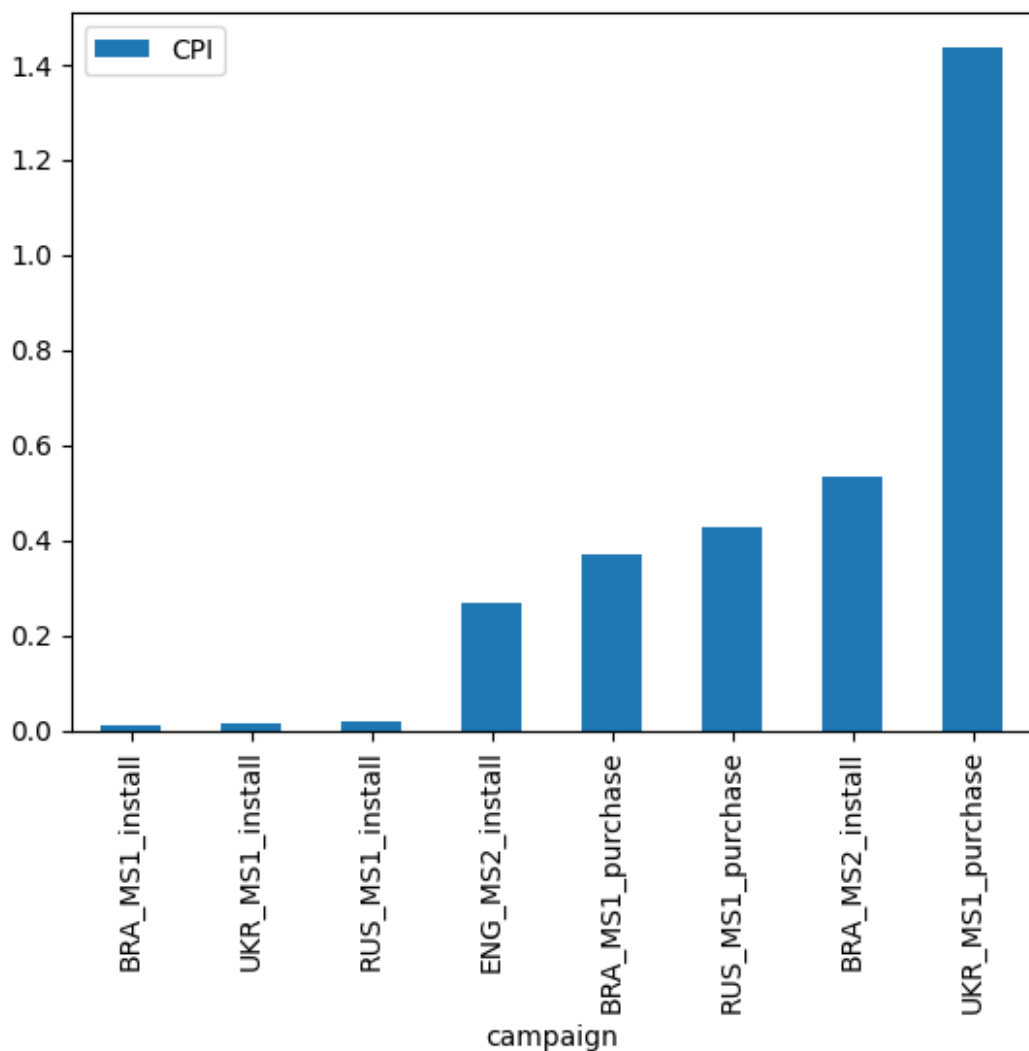
```
# Рассчитываем CPI для каждой кампании и добавляем его в DataFrame
campaign['CPI'] = campaign['Spend, USD'] / campaign['Installs']
campaign.sort_values(by='CPI', ascending=False)
```

	media_source	Campaign_type	Target	Plan, USD	Installs
5	NaN	purchase	Ukraine	500	242
6	Media_source_2	install	Brazil	100	152
3	NaN	purchase	Russia	500	1441
1	NaN	purchase	Brazil	400	1081
7	NaN	install	English_speaking	100	371
2	NaN	install	Russia	200	11894
4	NaN	install	Ukraine	100	6424
0	Media_source_1	install	Brazil	100	10460
	Spend, USD revenue	Deviation, USD	campaign	profit	

5	348.13	151.87	UKR_MS1_purchase	453.236576	
105.106576					
6	81.15	18.85	BRA_MS2_install	11.774035	-
69.375965					
3	616.77	-116.77	RUS_MS1_purchase	516.041879	-
100.728121					
1	398.67	1.33	BRA_MS1_purchase	444.513184	
45.843184					
7	99.95	0.05	ENG_MS2_install	25.626667	-
74.323333					
2	246.30	-46.30	RUS_MS1_install	325.681555	
79.381555					
4	99.43	0.57	UKR_MS1_install	67.437764	-
31.992236					
0	99.12	0.88	BRA_MS1_install	200.040209	
100.920209					
	CPI				
5	1.438554				
6	0.533882				
3	0.428015				
1	0.368797				
7	0.269407				
2	0.020708				
4	0.015478				
0	0.009476				

Больше всего пользователей по низкой цене получилось привлечь из Бразилии, России и Украины MS1. MS2 привлекла значительно меньше пользователей по гораздо более высокой цене. Компании purchase оказались наиболее дорогими.

```
campaign.sort_values(by='CPI').plot.bar(y='CPI', x='campaign');
```



Вывод

Затраты на рекламу быстро начинают приносить стабильную прибыль которая окупает затраты в среднем чуть больше чем за пол года, однако каждый регион имеет свою динамику развития, важно вовремя уловить момент и понять слабые места той или иной маркетинговой стратегии. Наиболее прибыльными оказался канал продвижения MS1, на момент оценки (7 месяцев) 4 компании (UKR_MS1_purchase, BRA_MS1_install, RUS_MS1_install, BRA_MS1_purchase) начали приносить прибыль которая постепенно увеличивается к концу года. Наибольшее ARPU приносят компании purchase, поскольку нацелены на пользователей лояльных к транзакциям, однако не всегда это стоит потраченных денег если цель быстрое извлечение прибыли, скорее всего на более длительном промежутке такие компании принесут больше прибыли. Компании purchase привлекают меньшее количество пользователей по более высокой цене за одного, однако такие пользователи приносят значительную часть дохода компании. Компании install логично привлекают больше пользователей и стоят дешевле, и за счет этого быстрее окупаются. В целом выбор той или иной компании будет зависеть от маркетинговой стратегии организации.