2_Cohort_analysis_churn_rate

December 12, 2022

SQL request project Yandex Practicum (personal visuzalization implementation in Python)

Cohort analysis of churn rate

[1]:
```
import pandas as pd
%load_ext sql
%sql postgresql://postgres:sqltest123@localhost/1
```

[2]:
```
%%sql result <<
WITH
prof AS ( SELECT DISTINCT tso.user_id,
                 DATE_TRUNC('month', MIN(tse.event_time))::date AS cohort_start
                 FROM tools_shop.orders AS tso
                 LEFT JOIN tools_shop.events AS tse ON tso.user_id = tse.user_id
                 GROUP BY 1),

sess AS ( SELECT cohort_start,
                 DATE_TRUNC('month', tse.event_time)::date AS session_dt,
                 COUNT(DISTINCT tse.user_id) AS c_users
                 FROM prof
                 LEFT JOIN tools_shop.events AS tse ON prof.user_id = tse.
 ↪user_id
                 GROUP BY 1,2),

fin AS (  SELECT cohort_start::varchar,
                 session_dt::varchar,
                 c_users,
                 LAG(c_users) OVER(PARTITION BY cohort_start ORDER BY␣
 ↪session_dt) AS p_users
                 FROM sess )

SELECT *,
ROUND((1 - c_users::numeric/p_users::numeric)*100*-1 , 2)::numeric AS churn_rate
FROM fin
WHERE cohort_start >= '2019-01-01'
AND cohort_start < '2020-01-01'
ORDER BY 1,2
```

```
 * postgresql://postgres:***@localhost/1
```

```
133 rows affected.
Returning data to local variable result
```

[3]:
```
df = result.DataFrame()
display(df.head(20))
```

	cohort_start	session_dt	c_users	p_users	churn_rate
0	2019-01-01	2019-01-01	306	NaN	None
1	2019-01-01	2019-02-01	62	306.0	-79.74
2	2019-01-01	2019-03-01	63	62.0	1.61
3	2019-01-01	2019-04-01	42	63.0	-33.33
4	2019-01-01	2019-05-01	40	42.0	-4.76
5	2019-01-01	2019-06-01	29	40.0	-27.50
6	2019-01-01	2019-07-01	12	29.0	-58.62
7	2019-01-01	2019-08-01	3	12.0	-75.00
8	2019-01-01	2019-12-01	1	3.0	-66.67
9	2019-01-01	2020-02-01	1	1.0	0.00
10	2019-01-01	2020-08-01	1	1.0	0.00
11	2019-01-01	2021-02-01	1	1.0	0.00
12	2019-02-01	2019-02-01	296	NaN	None
13	2019-02-01	2019-03-01	75	296.0	-74.66
14	2019-02-01	2019-04-01	42	75.0	-44.00
15	2019-02-01	2019-05-01	34	42.0	-19.05
16	2019-02-01	2019-06-01	37	34.0	8.82
17	2019-02-01	2019-07-01	32	37.0	-13.51
18	2019-02-01	2019-08-01	11	32.0	-65.63
19	2019-02-01	2019-09-01	2	11.0	-81.82

Shift churn_rate = None into 0

[10]:
```
cohort_group = df['cohort_start']
cohort_month = df['session_dt']
cohort_users = df['c_users']
churn_rate = df['churn_rate'].fillna(0).astype('float')

churn_r = list(zip(cohort_group, cohort_month, cohort_users, churn_rate))
df2 = pd.DataFrame(churn_r, columns = ['cohort_group', 'cohort_month',␣
↪'cohort_users', 'churn_rate'])
df2
```

[10]:

	cohort_group	cohort_month	cohort_users	churn_rate
0	2019-01-01	2019-01-01	306	0.00
1	2019-01-01	2019-02-01	62	-79.74
2	2019-01-01	2019-03-01	63	1.61
3	2019-01-01	2019-04-01	42	-33.33
4	2019-01-01	2019-05-01	40	-4.76
..	…	…	…	…
128	2019-12-01	2020-07-01	2	-92.31

```
129   2019-12-01   2020-08-01             1      -50.00
130   2019-12-01   2020-10-01             1        0.00
131   2019-12-01   2020-12-01             1        0.00
132   2019-12-01   2021-01-01             1        0.00

[133 rows x 4 columns]
```

[11]:
```
def cohort_period(df2):
    # changing cohort_sessions type into periods
    df2['cohort_month'] = np.arange(len(df2)) + 0
    return df2
```

[12]:
```
import numpy as np
cohorts = df2.groupby('cohort_group').apply(cohort_period)
cohorts.head(20)
```

[12]:

	cohort_group	cohort_month	cohort_users	churn_rate
0	2019-01-01	0	306	0.00
1	2019-01-01	1	62	-79.74
2	2019-01-01	2	63	1.61
3	2019-01-01	3	42	-33.33
4	2019-01-01	4	40	-4.76
5	2019-01-01	5	29	-27.50
6	2019-01-01	6	12	-58.62
7	2019-01-01	7	3	-75.00
8	2019-01-01	8	1	-66.67
9	2019-01-01	9	1	0.00
10	2019-01-01	10	1	0.00
11	2019-01-01	11	1	0.00
12	2019-02-01	0	296	0.00
13	2019-02-01	1	75	-74.66
14	2019-02-01	2	42	-44.00
15	2019-02-01	3	34	-19.05
16	2019-02-01	4	37	8.82
17	2019-02-01	5	32	-13.51
18	2019-02-01	6	11	-65.63
19	2019-02-01	7	2	-81.82

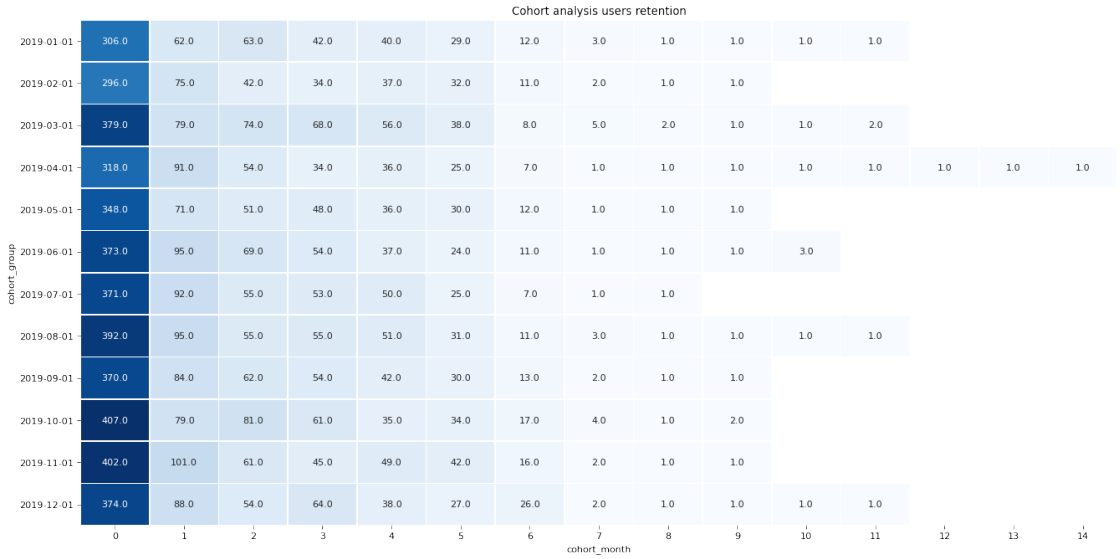
Two graphics for table

[26]:
```
import seaborn as sb
import matplotlib.pyplot as plt

cohort_users = cohorts['cohort_users'].fillna(0)
churn_r2 = list(zip(cohort_group, cohort_month, cohort_users))
df3 = pd.DataFrame(churn_r2, columns = ['cohort_group', 'cohort_month',␣
 ↪'cohort_users'])
```

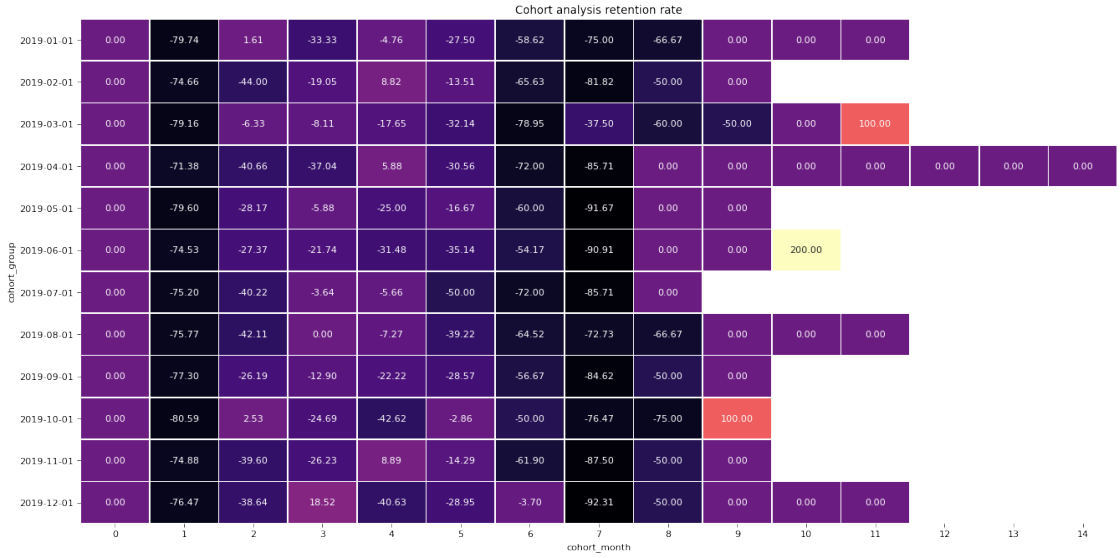
```
df_heatmap = df3.pivot('cohort_group', 'cohort_month', 'cohort_users')
plt.figure(figsize=(20,10), dpi=80)
sb.heatmap(df_heatmap,
           annot=True,
           cmap='Blues',
           fmt='',
           linewidth=.5,
           cbar=False).set(title='Cohort analysis users retention')
```

[26]: [Text(0.5, 1.0, 'Cohort analysis users retention')]



[28]:
```
df_heatmap = df4.pivot('cohort_group', 'cohort_month', 'churn_rate')
plt.figure(figsize=(20,10), dpi=80)
sb.heatmap(df_heatmap,
           annot=True,
           cmap='magma',
           fmt=".2f",
           linewidth=.5,
           cbar=False).set(title='Cohort analysis retention rate')
```

[28]: [Text(0.5, 1.0, 'Cohort analysis retention rate')]


Cohort analysis retention rate
cohort_group
2019-01-01 0.00 -79.74 1.61 -33.33 -4.76 -27.50 -58.62 -75.00 -66.67 0.00 0.00 0.00
2019-02-01 0.00 -74.66 -44.00 -19.05 8.82 -13.51 -65.63 -81.82 -50.00 0.00
2019-03-01 0.00 -79.16 -6.33 -8.11 -17.65 -32.14 -78.95 -37.50 -60.00 -50.00 0.00 100.00
2019-04-01 0.00 -71.38 -40.66 -37.04 5.88 -30.56 -72.00 -85.71 0.00 0.00 0.00 0.00 0.00 0.00 0.00
2019-05-01 0.00 -79.60 -28.17 -5.88 -25.00 -16.67 -60.00 -91.67 0.00 0.00
2019-06-01 0.00 -74.53 -27.37 -21.74 -31.48 -35.14 -54.17 -90.91 0.00 0.00 200.00
2019-07-01 0.00 -75.20 -40.22 -3.64 -5.66 -50.00 -72.00 -85.71 0.00
2019-08-01 0.00 -75.77 -42.11 0.00 -7.27 -39.22 -64.52 -72.73 -66.67 0.00 0.00 0.00
2019-09-01 0.00 -77.30 -26.19 -12.90 -22.22 -28.57 -56.67 -84.62 -50.00 0.00
2019-10-01 0.00 -80.59 2.53 -24.69 -42.62 -2.86 -50.00 -76.47 -75.00 100.00
2019-11-01 0.00 -74.88 -39.60 -26.23 8.89 -14.29 -61.90 -87.50 -50.00 0.00
2019-12-01 0.00 -76.47 -38.64 18.52 -40.63 -28.95 -3.70 -92.31 -50.00 0.00 0.00 0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
cohort_month