SQL Request from Yandex Practicum course project.

Implementation in Python with usage of visualization.

Cohort analysis of retention rate for users who registered in 2019

In [1]:

```
import pandas as pd
%load_ext sql
%sql postgresql://postgres:sqltest123@localhost/test
```

In [ ]:

```
%%sql result <<
WITH
profile AS
  (SELECT u.user_id,
          DATE_TRUNC('month', MIN(event_time))::date AS dt
   FROM tools_shop.users u
   JOIN tools_shop.orders o ON u.user_id = o.user_id
   JOIN tools_shop.events e ON u.user_id = e.user_id
   GROUP BY 1),
sessions AS
  (SELECT p.user_id AS users,
          DATE_TRUNC('month', event_time)::date AS session_dt
   FROM tools_shop.events e
   JOIN profile p ON p.user_id = e.user_id
   GROUP BY 1,2),
cohort_users_cnt AS
  (SELECT dt,
          COUNT(user_id) AS cohort_users_cnt
   FROM profile
   GROUP BY 1)

SELECT p.dt AS cohort_group,
       session_dt AS cohort_session,
       COUNT(p.user_id) AS users_cnt,
       cohort_users_cnt,
       ROUND(COUNT(p.user_id) * 100.0 / cohort_users_cnt, 2)::float AS reten
FROM profile p
JOIN sessions s ON p.user_id = s.users
JOIN cohort_users_cnt AS cuc ON p.dt = cuc.dt
WHERE p.dt >= '2019-01-01'
AND p.dt < '2020-01-01'
GROUP BY 1, 2, 4
ORDER BY 1,2
```

In [7]:

```
#displaying results for SQL request
df = result.DataFrame()
display(df.head(6))
```

	cohort_group	cohort_session	users_cnt	cohort_users_cnt	retention_rate
0	2019-01-01	2019-01-01	306	306	100.00
1	2019-01-01	2019-02-01	62	306	20.26
2	2019-01-01	2019-03-01	63	306	20.59
3	2019-01-01	2019-04-01	42	306	13.73
4	2019-01-01	2019-05-01	40	306	13.07
5	2019-01-01	2019-06-01	29	306	9.48

In [4]:

```python
#preparing table with required data for visualization
cohort_group = list(df['cohort_group'])
cohort_month = list(df['cohort_session'])
retention_rate = list(df['retention_rate'])

ret_r = list(zip(cohort_group, cohort_month, retention_rate))
df2 = pd.DataFrame(ret_r, columns = ['cohort_group', 'cohort_month', 'retent
```

In [8]:

```python
import numpy as np

# function to change cohort months date format into ranks
def cohort_period(df2):
    df2['cohort_month'] = np.arange(len(df2)) + 0
    return df2

cohorts = df2.groupby('cohort_group').apply(cohort_period)
cohorts.head(6)
```

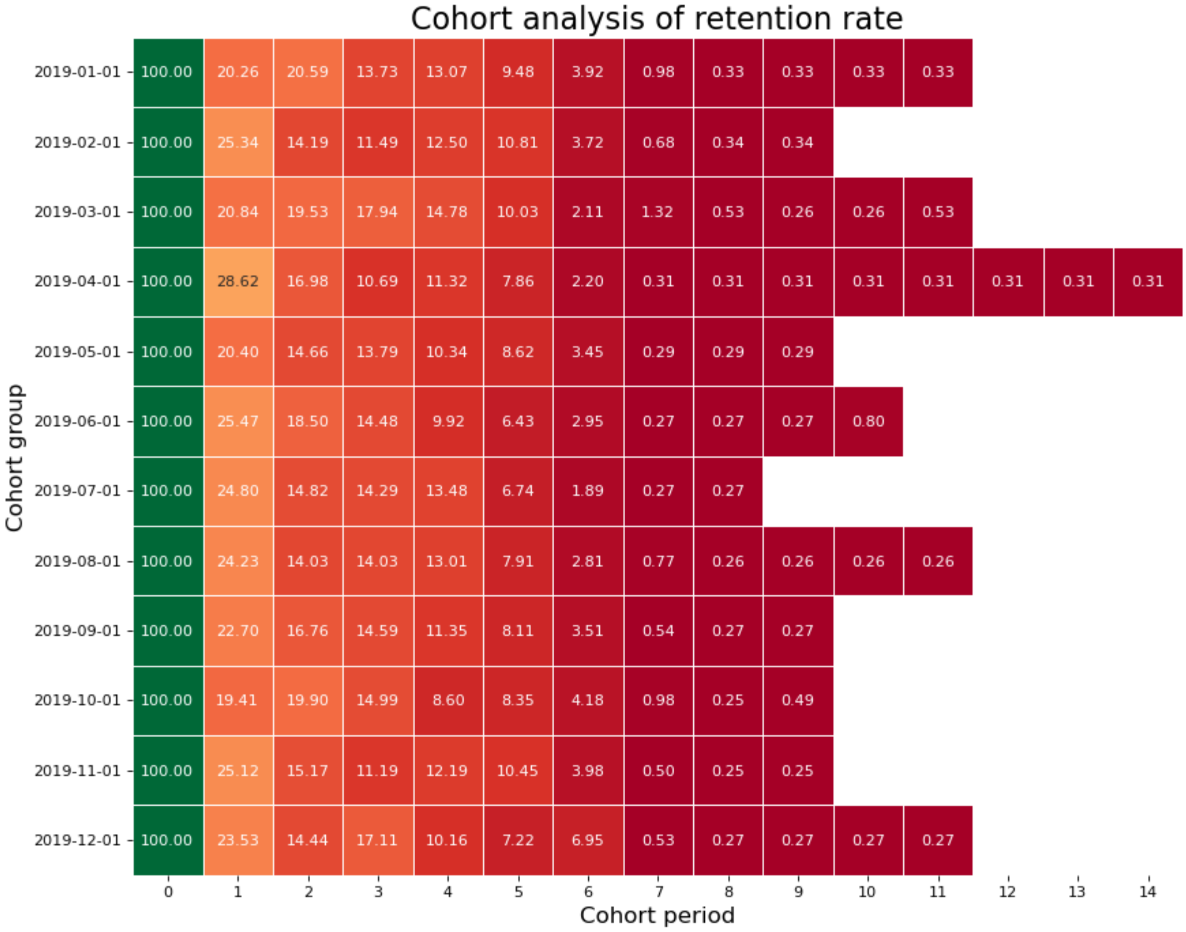
Out[8]:

	cohort_group	cohort_month	retention_rate
0	2019-01-01	0	100.00
1	2019-01-01	1	20.26
2	2019-01-01	2	20.59
3	2019-01-01	3	13.73
4	2019-01-01	4	13.07
5	2019-01-01	5	9.48

In [6]:

```python
import seaborn as sb
import matplotlib.pyplot as plt

df_heatmap = cohorts.pivot('cohort_group', 'cohort_month', 'retention_rate')
plt.figure(figsize=(20,10), dpi=80)
sb.heatmap(df_heatmap,
           annot=True,
           robust=True,
           square=True,
           cmap='RdYlGn',
           fmt=".2f",
           linewidth=.5,
           cbar=False)
plt.ylabel('Cohort group', size=15)
plt.xlabel('Cohort period', size=15)
plt.title('Cohort analysis of retention rate', size=20)
print('Retention rate is the ratio of the number of retained customers to th
Retention rate is the ratio of the number of retained customers to the numbe
r at risk
```



Cohort group	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2019-01-01	100.00	20.26	20.59	13.73	13.07	9.48	3.92	0.98	0.33	0.33	0.33	0.33			
2019-02-01	100.00	25.34	14.19	11.49	12.50	10.81	3.72	0.68	0.34	0.34					
2019-03-01	100.00	20.84	19.53	17.94	14.78	10.03	2.11	1.32	0.53	0.26	0.26	0.53			
2019-04-01	100.00	28.62	16.98	10.69	11.32	7.86	2.20	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
2019-05-01	100.00	20.40	14.66	13.79	10.34	8.62	3.45	0.29	0.29	0.29					
2019-06-01	100.00	25.47	18.50	14.48	9.92	6.43	2.95	0.27	0.27	0.27	0.80				
2019-07-01	100.00	24.80	14.82	14.29	13.48	6.74	1.89	0.27	0.27						
2019-08-01	100.00	24.23	14.03	14.03	13.01	7.91	2.81	0.77	0.26	0.26	0.26	0.26			
2019-09-01	100.00	22.70	16.76	14.59	11.35	8.11	3.51	0.54	0.27	0.27					
2019-10-01	100.00	19.41	19.90	14.99	8.60	8.35	4.18	0.98	0.25	0.49					
2019-11-01	100.00	25.12	15.17	11.19	12.19	10.45	3.98	0.50	0.25	0.25					
2019-12-01	100.00	23.53	14.44	17.11	10.16	7.22	6.95	0.53	0.27	0.27	0.27	0.27			