

In [63]:

```
users = users.merge(orders, on='uid')
```

делаем срезы по источникам для расчета ltv

In [64]:

```
users['source_id'].value_counts()
```

Out[64]:

```
3      13289
4      12895
5       9406
2       6392
1       5223
9       1647
10      1562
7          1
Name: source_id, dtype: int64
```

In [65]:

```
ltv1 = users.query('source_id == 1')
```

In [66]:

```
cohort_size1 = ltv1.groupby('first_order_month').agg({'uid': 'nunique'}).reset_index()
cohort_size1.rename(columns={'uid': 'n_buyers'}, inplace=True)
```

In [67]:

```
cohort_size1
```

Out[67]:

	first_order_month	n_buyers
0	2017-06-01	190
1	2017-07-01	160
2	2017-08-01	113
3	2017-09-01	227
4	2017-10-01	340
5	2017-11-01	324
6	2017-12-01	331
7	2018-01-01	238
8	2018-02-01	275
9	2018-03-01	281
10	2018-04-01	179
11	2018-05-01	241

In [68]:

```
ltv2 = users.query('source_id == 2')
```

In [69]:

```
ltv3 = users.query('source_id == 3')
```

In [70]:

```
ltv4 = users.query('source_id == 4')
```

In [71]:

```
ltv5 = users.query('source_id == 5')
```

In [72]:

```
ltv9 = users.query('source_id == 9')
```

In [73]:

```
ltv10 = users.query('source_id == 10')
```

находим возраст когорты

In [74]:

```
ltv1['age_month'] = ((ltv1['order_month'] - ltv1['first_order_month']) / np.timedelta64(1, 'M')).round()
```

In [75]:

ltv1

Out[75]:

	uid	device	end_ts	source_id	start_ts	session_year	session_
48	21801295586117262	touch	2018-01-26 19:14:19	1	2018-01-26 19:14:00	2018	2018
49	21801295586117262	touch	2018-01-26 19:14:19	1	2018-01-26 19:14:00	2018	2018
50	22538937500067142	touch	2018-02-16 22:33:00	1	2018-02-16 22:06:00	2018	2018
51	22538937500067142	touch	2018-02-16 22:33:00	1	2018-02-16 22:06:00	2018	2018
62	28094489227360733	desktop	2017-06-29 18:01:00	1	2017-06-29 17:41:00	2017	2017
...
50334	18404729752869455706	desktop	2017-11-09 15:31:00	1	2017-11-09 15:14:00	2017	2017
50335	18405234926187729536	touch	2017-07-20 12:26:00	1	2017-07-20 12:22:00	2017	2017
50341	18410655351095696115	desktop	2018-02-16 14:59:00	1	2018-02-16 14:58:00	2018	2018
50373	18423636499440883616	touch	2018-01-30 14:38:00	1	2018-01-30 14:21:00	2018	2018
50399	18436650472890569350	desktop	2017-06-06 16:10:00	1	2017-06-06 16:08:00	2017	2017

5223 rows × 16 columns

In [76]:

```
# считаем выручку на каждого покупателя
cohorts_ltv1 = pd.merge(cohort_size1, ltv1, on = 'first_order_month')
cohorts_ltv1['revenue_per_buyer'] = cohorts_ltv1['revenue'] / cohorts_ltv1['n_buyers']
```

In [77]:

```
# теперь показываем накопительную выручку на покупателя
cohorts_report_ltv1 = cohorts_ltv1.pivot_table(index='first_order_month', columns = 'age_month', values='revenue_per_buyer', aggfunc='sum').cumsum(axis=1)
cohorts_report_ltv1.round(2).fillna('')
```

Out[77]:

	age_month	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
first_order_month												
2017-06-01		7.26	9.44	11.65	15.41	19.68	21.73	24.31	26.83	29.1	30.77	31.97
2017-07-01		7.29	8.39	13.74	15.25	16.13	17.03	17.47	18.12	18.98	19.96	20.89
2017-08-01		7.49	8.69	9.7	10.67	11.7	12.37	13.05	15.71	17.9	18.68	
2017-09-01		6.01	7.82	8.38	9.3	9.34	9.43	9.52	9.83	10.42		
2017-10-01		6.02	7.07	7.32	7.71	7.77	7.88	8.06	8.38			
2017-11-01		7.39	7.96	8.93	9.82	9.99	10.09	10.17				
2017-12-01		4.58	4.9	5.18	5.43	5.46	5.55					
2018-01-01		5.77	5.86	6	6.98	7.03						
2018-02-01		4.69	5.69	5.75	6.12							
2018-03-01		9.34	11.28	12.07								
2018-04-01		4.70	4.97									
2018-05-01		5.22										

In [78]:

```
cac1 = cac.query('source_id == 1')
```

In [79]:

```
cac1
```

Out[79]:

	month	source_id	costs	n_buyers	cac
0	2017-06-01	1	1125.61	401	2.81
7	2017-07-01	1	1072.88	400	2.68
14	2017-08-01	1	951.81	288	3.30
21	2017-09-01	1	1502.01	526	2.86
28	2017-10-01	1	2315.75	833	2.78
35	2017-11-01	1	2445.16	944	2.59
42	2017-12-01	1	2341.20	903	2.59
49	2018-01-01	1	2186.18	638	3.43
56	2018-02-01	1	2204.48	646	3.41
63	2018-03-01	1	1893.09	676	2.80
70	2018-04-01	1	1327.49	391	3.40
77	2018-05-01	1	1467.61	492	2.98

In [80]:

```
cac1 = cac1.drop(columns=['source_id', 'costs', 'n_buyers'])
```

In [81]:

```
cac1
```

Out[81]:

	month	cac
0	2017-06-01	2.81
7	2017-07-01	2.68
14	2017-08-01	3.30
21	2017-09-01	2.86
28	2017-10-01	2.78
35	2017-11-01	2.59
42	2017-12-01	2.59
49	2018-01-01	3.43
56	2018-02-01	3.41
63	2018-03-01	2.80
70	2018-04-01	3.40
77	2018-05-01	2.98

In [82]:

```
cac1.rename(columns={'month': 'first_order_month'}, inplace=True)
```

In [83]:

```
romi1 = pd.merge(cohorts_ltv1, cac1, on='first_order_month')
```

In [84]:

```
romi1
```

Out[84]:

	first_order_month	n_buyers	uid	device	end_ts	source_id	star
0	2017-06-01	190	143427074811278640	desktop	2017-06-20 16:24:00	1	2017-06-20 16:00:00
1	2017-06-01	190	143427074811278640	desktop	2017-06-20 16:24:00	1	2017-06-20 16:00:00
2	2017-06-01	190	148401564840028266	desktop	2017-06-05 19:51:00	1	2017-06-05 19:11:00
3	2017-06-01	190	201308712819916034	desktop	2017-06-14 16:23:00	1	2017-06-14 16:11:00
4	2017-06-01	190	409262394337119652	desktop	2017-06-14 13:03:00	1	2017-06-14 13:00:00
...
5218	2018-05-01	241	17997491778171731375	desktop	2018-05-12 14:14:00	1	2018-05-12 14:14:00
5219	2018-05-01	241	18220171840130521054	desktop	2018-05-19 10:42:00	1	2018-05-19 10:38:00
5220	2018-05-01	241	18220171840130521054	desktop	2018-05-19 10:42:00	1	2018-05-19 10:38:00
5221	2018-05-01	241	18380686934038836742	touch	2018-05-17 21:04:00	1	2018-05-17 21:00:00
5222	2018-05-01	241	18397039008941125340	desktop	2018-05-18 16:23:00	1	2018-05-18 16:20:00

5223 rows × 19 columns

In [85]:

```
romi1['romi'] = romi1['revenue_per_buyer'] / romi1['cac']
```

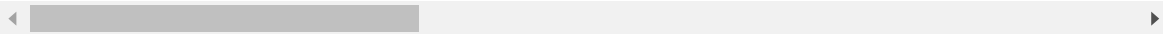
In [86]:

```
romi1.fillna('')
```

Out[86]:

	first_order_month	n_buyers	uid	device	end_ts	source_id	star
0	2017-06-01	190	143427074811278640	desktop	2017-06-20 16:24:00	1	2017-06-20 16:00:00
1	2017-06-01	190	143427074811278640	desktop	2017-06-20 16:24:00	1	2017-06-20 16:00:00
2	2017-06-01	190	148401564840028266	desktop	2017-06-05 19:51:00	1	2017-06-05 19:11:00
3	2017-06-01	190	201308712819916034	desktop	2017-06-14 16:23:00	1	2017-06-14 16:11:00
4	2017-06-01	190	409262394337119652	desktop	2017-06-14 13:03:00	1	2017-06-14 13:00:00
...
5218	2018-05-01	241	17997491778171731375	desktop	2018-05-12 14:14:00	1	2018-05-12 14:14:00
5219	2018-05-01	241	18220171840130521054	desktop	2018-05-19 10:42:00	1	2018-05-19 10:38:00
5220	2018-05-01	241	18220171840130521054	desktop	2018-05-19 10:42:00	1	2018-05-19 10:38:00
5221	2018-05-01	241	18380686934038836742	touch	2018-05-17 21:04:00	1	2018-05-17 21:00:00
5222	2018-05-01	241	18397039008941125340	desktop	2018-05-18 16:23:00	1	2018-05-18 16:20:00

5223 rows × 20 columns



In [88]:

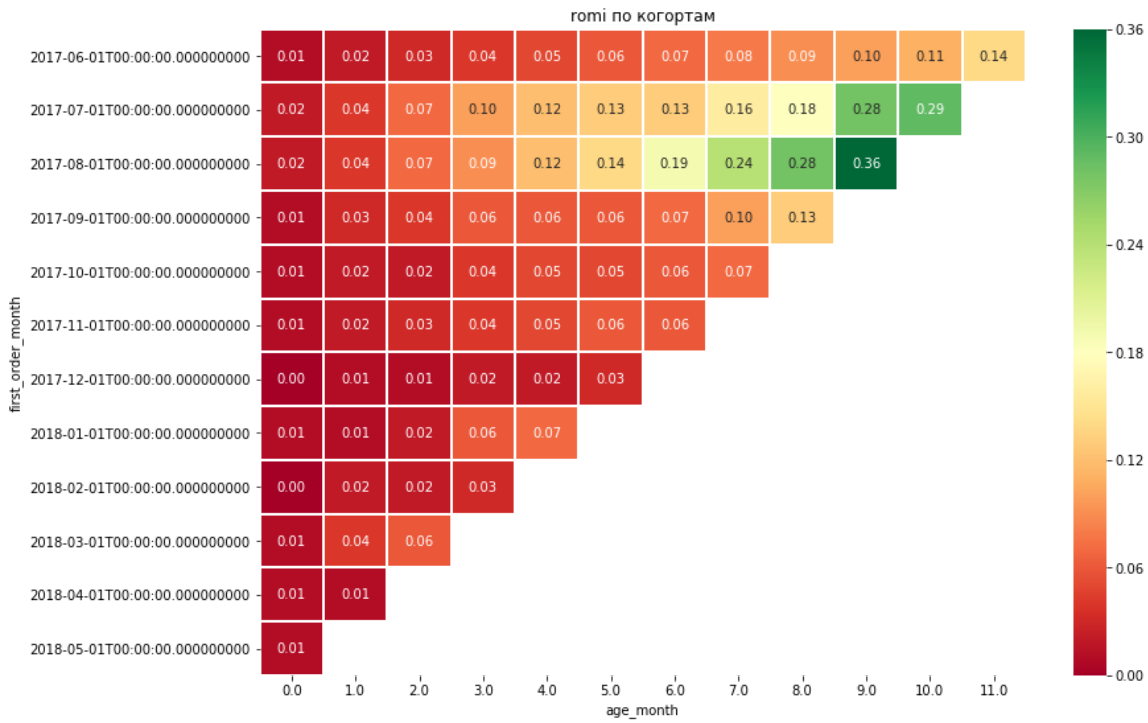
```
romi1 = romi1.pivot_table(index='first_order_month', columns='age_month', values='romi'
, aggfunc='mean').cumsum(axis=1).round(2)
romi1.fillna('')
```

Out[88]:

age_month	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
first_order_month												
2017-06-01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.14
2017-07-01	0.02	0.04	0.07	0.1	0.12	0.13	0.13	0.16	0.18	0.28	0.29	
2017-08-01	0.02	0.04	0.07	0.09	0.12	0.14	0.19	0.24	0.28	0.36		
2017-09-01	0.01	0.03	0.04	0.06	0.06	0.06	0.07	0.1	0.13			
2017-10-01	0.01	0.02	0.02	0.04	0.05	0.05	0.06	0.07				
2017-11-01	0.01	0.02	0.03	0.04	0.05	0.06	0.06					
2017-12-01	0.00	0.01	0.01	0.02	0.02	0.03						
2018-01-01	0.01	0.01	0.02	0.06	0.07							
2018-02-01	0.00	0.02	0.02	0.03								
2018-03-01	0.01	0.04	0.06									
2018-04-01	0.01	0.01										
2018-05-01	0.01											

In [89]:

```
plt.figure(figsize=(13, 9))
plt.title('romi по когортам')
sns.heatmap(romi1, annot=True, fmt='.2f', linewidths=1, cmap="RdYlGn");
```



In []: