

1. Найти падение вероятности покупки в зависимости от кол-ва часов после времени регистрации user (предположение, что если пользователь после регистрации в течение 5 часов не купил, он по сути уже и не купит). Важно!: само время заказа берите равное последней дате добавления строчки в баскете по этому заказу.

```
-- создадим облегченную таблицу basket иначе все повиснет очень надолго
```

```
USE orders
```

```
CREATE TABLE basket
```

```
  SELECT * FROM basket16
```

```
  LIMIT 10000;
```

```
-- создадим промежуточную таблицу которая будет содержать id заказа из orders_
```

```
-- и последнюю дату вставки из корзины по такому id заказа
```

```
USE orders
```

```
CREATE TABLE orders_date_ins_max
```

```
  SELECT o.user_id, MAX(b.DATE_INSERT) AS date_ins_max
```

```
  FROM orders_ o
```

```
  LEFT JOIN basket b
```

```
  ON o.id_o = b.ORDER_ID
```

```
  GROUP BY b.ORDER_ID;
```

```
-- из новой таблицы date_ins_max создадим таблицу
```

```
-- которая содержит user_id и первую дату покупки
```

```
USE orders
```

```
CREATE TABLE user_first_order
```

```
  SELECT user_id, MIN(date_ins_max) AS first_order
```

```
  FROM orders_date_ins_max
```

```
  GROUP BY user_id;
```

```
-- сделаем слияние таблиц первого заказа и таблицы
```

```
-- с данными о дате регистрации по столбцу user_id
```

```
-- откидывая даты до 2016 года и запишем в новую таблицу
```

```
USE orders
```

```
CREATE TABLE user_fo_reg
```

```
  SELECT ufo.user_id, ufo.first_order, u.DATE_REGISTER
```

```
  FROM user_first_order ufo
```

```
  LEFT JOIN user16 u
```

```
  ON ufo.user_id = u.u_id
```

```
  WHERE u.DATE_REGISTER > STR_TO_DATE('01.01.2016', '%d.%m.%Y');
```

```
-- сделаем таблицу с разницей между датой заказа и регистрации в часах
```

```
-- и столбцом количества таких записей
```

```
USE orders
```

```
CREATE TABLE diff_hour
```

```
  SELECT HOUR(TIMEDIFF(STR_TO_DATE(first_order, '%d.%m.%Y %H:%i:%s'), DATE_REGISTER)) AS hour_diff, count(*)
```

```
  FROM user_fo_reg
```

```
  GROUP BY hour_diff
```

```
  ORDER BY hour_diff;
```



2. Найти вероятность по дням повторной покупки после совершения первой (предположение: после первой покупки пик вероятности достигает на 10-14 день - так как по идее должна быть регулярность покупок в десткой сфере)

```
-- облегченная таблица orders_
USE orders
CREATE TABLE orders_1
SELECT * FROM orders_
LIMIT 100000;

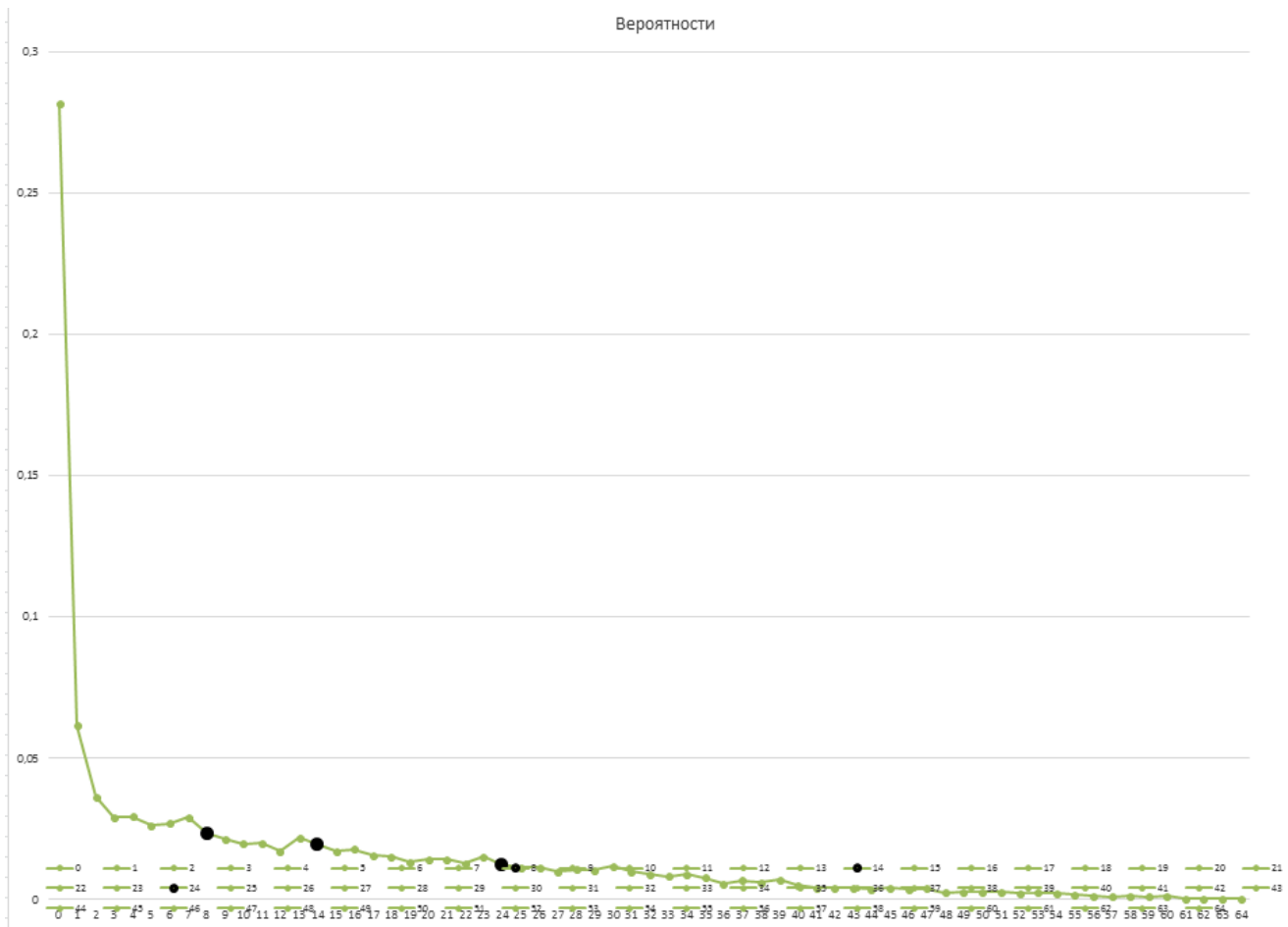
-- создадим таблицу с id пользователя, id заказа и первой датой заказа
-- из облегченной orders
USE orders
CREATE TABLE orders_first_date
SELECT user_id, id_o, MIN(o_date) AS min_date
FROM orders_1
GROUP BY user_id;

-- создадим такую же таблицу но со следующей минимальной датой
USE orders
CREATE TABLE orders_second_date
SELECT user_id, id_o, MIN(o_date) AS min_date
FROM orders_1 o
WHERE id_o NOT IN
(
SELECT id_o
FROM orders_first_date
)
GROUP BY user_id;

-- даты первого и второго заказа
SELECT DATEDIFF (omd2.min_date, omd1.min_date) AS datediff, count(*)
FROM orders_min_date omd1
INNER JOIN orders_second_date omd2
ON omd1.user_id = omd2.user_id
GROUP BY datediff
ORDER BY datediff;
```

datediff	count(*)	
0	4178	0,281555361
1	911	0,061392277
2	536	0,036121032
3	433	0,029179864
4	434	0,029247254
5	386	0,026012535
6	397	0,026753824
7	432	0,029112474
8	347	0,023384325
9	319	0,021497405
10	291	0,019610486
11	296	0,019947436
12	257	0,017319226
13	326	0,021969135
14	289	0,019475706
15	257	0,017319226
16	264	0,017790956
17	231	0,015567087
18	222	0,014960577
19	193	0,013006267
20	214	0,014421457
21	208	0,014017117
22	192	0,012938877
23	223	0,015027967
24	183	0,012332367
25	169	0,011388908
26	168	0,011321518
27	149	0,010041108
28	157	0,010580228
29	153	0,010310668
30	174	0,011725858
31	147	0,009906328
32	131	0,008828088
33	119	0,008019408

34	131	0,008828088
35	114	0,007682458
36	82	0,005525979
37	97	0,006536829
38	87	0,005862929
39	104	0,007008559
40	75	0,005054249
41	63	0,004245569
42	61	0,004110789
43	62	0,004178179
44	51	0,003436889
45	58	0,003908619
46	51	0,003436889
47	56	0,003773839
48	33	0,00222387
49	41	0,002762989
50	38	0,002560819
51	38	0,002560819
52	30	0,0020217
53	34	0,00229126
54	31	0,00208909
55	27	0,00181953
56	20	0,0013478
57	12	0,00080868
58	18	0,00121302
59	14	0,00094346
60	15	0,00101085
61	4	0,00026956
62	2	0,00013478
63	3	0,00020217
64	1	6,739E-05
Bcero:	14839	



3. Найти среднее время создания заказа (посмотреть в рамках таблицы basket на один заказ сколько времени приходится по полю date_insert)

```
-- создадим таблицу
USE orders
CREATE TABLE basket_datediff
SELECT ORDER_ID,
       MIN(STR_TO_DATE(DATE_INSERT, '%d.%m.%Y %H:%i:%s')) AS min_date,
       MAX(STR_TO_DATE(DATE_INSERT, '%d.%m.%Y %H:%i:%s')) AS max_date
FROM basket16
GROUP BY ORDER_ID;

-- среднее время заказа
SELECT AVG(SECOND(TIMEDIFF(max_date, min_date))) AS sec_diff
FROM basket_datediff
```

sec_diff	
DECIMAL(5, 4)	
0,0681	