

Функции в Pandas

Примеры из презентации

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.DataFrame({'user_id': [1, 2, 3], 'clicks': [163, 130, 97], 'orders': [2, 4, 0]})  
df
```

Out[2]:

	user_id	clicks	orders
0	1	163	2
1	2	130	4
2	3	97	0

In [3]:

```
df = pd.DataFrame({'user_id': [1, 2, 3], 'clicks': [163, 130, 97], 'orders': [2, 4, 0], 'calculated': [False, False, True]})  
df[['user_id', 'clicks', 'orders', 'calculated']]
```

Out[3]:

	user_id	clicks	orders	calculated
0	1	163	2	False
1	2	130	4	False
2	3	97	0	True

In [4]:

```
def watcher(param):  
    """Мне только посмотреть"""  
    return param == 0
```

In [5]:

```
df['watcher'] = df['orders'].apply(watcher)  
df
```

Out[5]:

	user_id	clicks	orders	calculated	watcher
0	1	163	2	False	False
1	2	130	4	False	False
2	3	97	0	True	True

Применяем метод apply к одному столбцу. Сейчас в переменную функции передаются значения одного столбца

In [6]:

```
def conversion(row):  
    """Подсчет конверсии переходов в покупки"""  
    return row['orders'] / row['clicks']
```

Как потестировать функцию со значениями одной строки

In [7]:

```
row1 = pd.DataFrame({'clicks': [163],  
                     'orders': [2],  
                     'user_id': [1]})  
conversion(row1)
```

Out[7]:

```
0    0.01227  
dtype: float64
```

In [11]:

row1

Out[11]:

	clicks	orders	user_id
0	163	2	1

Применяем метод apply к датафрейму. В переменную функции передаются строки целиком

In [8]:

```
df = pd.DataFrame({'user_id': [1, 2, 3], 'clicks': [163, 130, 97], 'orders': [2, 4, 0]})  
df[['user_id', 'clicks', 'orders']]
```

Out[8]:

	user_id	clicks	orders
0	1	163	2
1	2	130	4
2	3	97	0

In [9]:

```
df['conversion'] = df.apply(conversion, axis=1)
```

In [10]:

```
df[['user_id', 'clicks', 'orders', 'conversion']]
```

Out[10]:

	user_id	clicks	orders	conversion
0	1	163	2	0.012270
1	2	130	4	0.030769
2	3	97	0	0.000000

Анализ рекламных кампаний

In [12]:

```
import pandas as pd
```

In [13]:

```
stats = pd.read_excel('ad_campaigns.xlsx')
stats.head()
```

Out[13]:

	Название группы	Фраза (с минус-словами)	Продуктивность	ID объявления	Заголовок	Текст
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. МР в дни обращения Консультаци: вр..
1	мрт цао	"мрт менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. МР в дни обращения Консультаци: вр..
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	MPT на Менделеевской от 2000Р	24 часа. МР в дни обращения Консультаци: вр..
3	мрт цао	"!мрт !цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. МР в дни обращения Консультаци: вр..
4	мрт сао	"мрт цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. МР в дни обращения Консультаци: вр..

In [14]:

```
stats.columns = ['group', 'phrase', 'effect', 'ad_id', 'title', 'text', 'link']
stats.head()
```

Out[14]:

	group	phrase	effect	ad_id	title	text
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр... https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр... https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр... https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр... https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр... https://a utm_sourc

Lambda-функции

Хотим посчитать распределение количества слов в столбце с фразами

In [15]:

```
phrase = 'MPT на Менделеевской от 2000'
```

In [16]:

```
phrase.split(' ')
```

Out[16]:

```
['MPT', 'на', 'Менделеевской', 'от', '2000']
```

In [17]:

```
len(phrase.split(' '))
```

Out[17]:

5

In [18]:

```
def word_count(phrase):
    return len(phrase.split(' '))
```

In [19]:

```
word_count('МРТ на Менделеевской от 2000')
```

Out[19]:

5

In []:

```
word_count = lambda phrase: len(phrase.split(' '))
```

In [20]:

```
stats['word_count'] = stats['phrase'].apply(lambda word: len(word.split(' ')))
stats.head()
```

Out[20]:

	group	phrase	effect	ad_id	title	text	
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	МРТ на Менделеевской от 2000Р	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	МРТ на Менделеевской от 2000Р	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	МРТ на Менделеевской от 2000Р	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	МРТ в ЦАО от 2000Р. Звоните!	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	МРТ в ЦАО от 2000Р. Звоните!	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc

In [21]:

```
# вариант с передачей всей строки функции
# тут надо обязательно указать параметр axis = 1

stats['word_count'] = stats.apply(lambda x: len(x['phrase'].split(' ')), axis=1)
stats.head()
```

Out[21]:

	group	phrase	effect	ad_id	title	text	
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	МРТ на Менделеевской от 2000Р	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	МРТ на Менделеевской от 2000Р	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	МРТ на Менделеевской от 2000Р	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	МРТ в ЦАО от 2000Р. Звоните!	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	МРТ в ЦАО от 2000Р. Звоните!	24 часа. МРТ в день обращения. Консультация вр...	https://a utm_sourc

Упражнение

Напишите отдельную функцию `word_count`, которая считает количество слов в столбце `phrase`. Функцию можно применять как к столбцу `phrase`, так и к строке целиком.

In []:

In []:

In []:

Произвольные функции

В URL кампаний есть названия. С этим надо что-то делать

In [23]:

```
# обращение к индексу и значениям Series
res = stats.word_count.value_counts()
res
```

Out[23]:

```
3      1129
4       529
2       189
5       137
6        84
7        38
8        30
9         12
10         6
12         2
21         1
17         1
11         1
Name: word_count, dtype: int64
```

In []:

```
res.index[0]
```

In []:

```
res.values[0]
```

In [22]:

```
# пример ссылки
url = stats.loc[0, 'link']
url
```

Out[22]:

```
'https://awesome-site.ru/?utm_source=yandex&utm_medium=cpc&utm_campaign=a825749b87&utm_content=dev_{device_type}'
```

In [24]:

```
from urllib import parse
```

In [25]:

```
parse.urlsplit('https://ya.ru/news/sport?search=footbal#abc')
```

Out[25]:

```
SplitResult(scheme='https', netloc='ya.ru', path='/news/sport', query='search=footbal', fragment='abc')
```

In [26]:

```
parse.urlsplit('https://ya.ru/news/sport?search=footbal#abc').netloc
```

Out[26]:

```
'ya.ru'
```

In [27]:

```
parsed = parse.urlsplit(url)
parsed
```

Out[27]:

```
SplitResult(scheme='https', netloc='awesome-site.ru', path='/', query='utm_source=yandex&utm_medium=cpc&utm_campaign=a825749b87&utm_content=dev_{device_type}', fragment='')
```

In []:

```
# можно конечно вручную
parsed.query.split('&')[2].split('=')[1]
```

In []:

```
# как доставать значения

parsed.netloc
```

In [28]:

```
params = parse.parse_qs(parsed.query)
params
```

Out[28]:

```
{'utm_source': ['yandex'],
 'utm_medium': ['cpc'],
 'utm_campaign': ['a825749b87'],
 'utm_content': ['dev_{device_type}']}
```

In [29]:

```
# вот и кампании

params['utm_campaign'][0]
```

Out[29]:

```
'a825749b87'
```


In [30]:

```
# зачем тут везде списки?

url_with_doubles = 'https://awesome-site.ru/?a=1&a=2&a=3'

parsed = parse.urlsplit(url_with_doubles)
parse.parse_qs(parsed.query)
```

Out[30]:

```
{'a': ['1', '2', '3']}
```

In [31]:

```
# оборачиваем все в функцию
# в качестве аргумента будет строка датафрейма

def campaign_name(row):
    """Получение названия кампании из ссылки внутри строки row"""

    parsed = parse.urlsplit(row['link'])
    params_dict = parse.parse_qs(parsed.query)

    return params_dict['utm_campaign'][0]
```

In [32]:

```
# проверяем датафрейм
stats.head()
```

Out[32]:

	group	phrase	effect	ad_id	title	text	
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc

In [33]:

```
stats['campaign'] = stats.apply(campaign_name, axis=1)
stats.head()
```

Out[33]:

	group	phrase	effect	ad_id	title	text	
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc

Как передать в функцию несколько аргументов

In [34]:

```
# как передать несколько аргументов

def power_up(row, n):
    """Возводит значение столбца effect в степень n"""
    return row['effect'] ** n
```

In [36]:

```
stats['power_up'] = stats.apply(power_up, n=2, axis=1)
stats.head()
```

Out[36]:

	group	phrase	effect	ad_id	title	text	
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc

Упражнение

В наших данных есть много объявлений с услугой MPT (в столбце group есть слово 'мрт') круглосуточно (в тексте объявления text есть '24 часа'). Отфильтруйте строки, в которых НЕ упоминается MPT, но прием идет круглосуточно. Сколько таких строк в датасете?

In [37]:

```
def mrt(row):
    if ('мрт' not in row['group']) and ('24 часа' in row['text']):
        return True

    return False
```

In [38]:

```
stats['mrt'] = stats.apply(mrt, axis=1)
```

In [39]:

stats.head()

Out[39]:

	group	phrase	effect	ad_id	title	text	
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc

In [40]:

stats.mrt.value_counts()

Out[40]:

```
False    2120
True       39
Name: mrt, dtype: int64
```

Группировки

In [41]:

```
df = pd.DataFrame({'order_id': [1, 2, 3, 4, 5], 'country': ['Россия', 'Китай', 'Китай', 'Ро',  
                  'category': ['Электроника', 'Авто', 'Электроника', 'Авто', 'Авто'],  
                  'amount': [100, 80, 90, 140, 90]})  
df
```

Out[41]:

	order_id	country	category	amount
0	1	Россия	Электроника	100
1	2	Китай	Авто	80
2	3	Китай	Электроника	90
3	4	Россия	Авто	140
4	5	Россия	Авто	90

Создадим датафрейм df_russia, в котором оставим заказы из России. И аналогично df_china (заказы из Китая).

In [42]:

```
df_russia = df[df.country == 'Россия']  
df_russia
```

Out[42]:

	order_id	country	category	amount
0	1	Россия	Электроника	100
3	4	Россия	Авто	140
4	5	Россия	Авто	90

In [43]:

```
df_china = df[df.country == 'Китай']  
df_china
```

Out[43]:

	order_id	country	category	amount
1	2	Китай	Авто	80
2	3	Китай	Электроника	90

In []:

Посчитаем для df_russia и df_china:

- среднюю стоимость заказа
- разницу между максимальной и минимальной стоимостью заказа

In [44]:

```
df_russia.amount.mean()
```

Out[44]:

110.0

In [45]:

```
df_china.amount.mean()
```

Out[45]:

85.0

In [46]:

```
df_russia.amount.max() - df_russia.amount.min()
```

Out[46]:

50

In [47]:

```
df_china.amount.max() - df_china.amount.min()
```

Out[47]:

10

Объединим процесс разбиения на датафреймы

In [48]:

```
def groupby_function(data):  
    return data.amount.max() - data.amount.min()
```

In [49]:

```
df.groupby('country').apply(groupby_function)
```

Out[49]:

```
country  
Китай      10  
Россия     50  
dtype: int64
```

Вернемся к статистике рекламных кампаний

In [50]:

```
# раньше использовали value_counts  
stats['campaign'].value_counts().head()
```

Out[50]:

```
7f35591a28    456  
8e77a0e565    348  
2cc2e7d770    240  
f6d2ae1e3d    220  
e90f4db55a    218  
Name: campaign, dtype: int64
```

In [51]:

```
# более универсальный способ  
stats.groupby('campaign').count().head()
```

Out[51]:

	group	phrase	effect	ad_id	title	text	link	word_count	power_up	mrt
campaign										
034ada41b6	3	3	3	3	3	3	3	3	3	3
03fcaecd1d	2	2	2	2	2	2	2	2	2	2
0697a81555	2	2	2	2	2	2	2	2	2	2
08cdcb57a3	3	3	3	3	3	3	3	3	3	3
0f0ba311fb	2	2	2	2	2	2	2	2	2	2

In [53]:

stats.head()

Out[53]:

	group	phrase	effect	ad_id	title	text	
0	мрт менделеевская	"!мрт ! менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
1	мрт цао	"мрт менделеевская"	4.5	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
2	мрт цао	мрт менделеевская -головы -позвоночника -сдела...	7.4	2101704995	MPT на Менделеевской от 2000Р	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
3	мрт цао	"!мрт !цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc
4	мрт сао	"мрт цао"	5.0	2101704999	MPT в ЦАО от 2000Р. Звоните!	24 часа. MPT в день обращения. Консультация вр...	https://a utm_sourc

In [52]:

stats.groupby('campaign').count()[['group', 'effect']].head()

Out[52]:

	group	effect
campaign		
034ada41b6	3	3
03fcaecd1d	2	2
0697a81555	2	2
08cdcb57a3	3	3
0f0ba311fb	2	2

Как вернуть столбец из индекса - метод `reset_index()`

In [54]:

```
stats.groupby('campaign').count().reset_index().head()
```

Out[54]:

	campaign	group	phrase	effect	ad_id	title	text	link	word_count	power_up	mrt
0	034ada41b6	3	3	3	3	3	3	3	3	3	3
1	03fcaecd1d	2	2	2	2	2	2	2	2	2	2
2	0697a81555	2	2	2	2	2	2	2	2	2	2
3	08cdcb57a3	3	3	3	3	3	3	3	3	3	3
4	0f0ba311fb	2	2	2	2	2	2	2	2	2	2

К группировке можно применять разные функции таким образом:

In [55]:

```
obj = stats.groupby('campaign')
```

In [56]:

```
type(obj)
```

Out[56]:

```
pandas.core.groupby.generic.DataFrameGroupBy
```

In [57]:

```
obj.max().head()
```

Out[57]:

	group	phrase	effect	ad_id	title	text
campaign						
034ada41b6	в недорого москва мрт	в москве открытого типа адрес мрт -цена	9.0	2428554779	Адрес МРТ открытого типа в Москве	24 часа. От 2 000Р. Консультация врача бесплат...
03fcaecd1d	сделать мрт сущевский вал	сделать мрт стромынка	6.5	2101705759	Сделайте МРТ на ул. Стромынка!	24 часа. От 2 000Р. Консультация врача бесплат...
0697a81555	мрт позвоночника сущевский вал	мрт позвоночника стромынка	5.5	2101706415	МРТ позвоночника на ул. Стромынка	24 часа. От 2 000Р. Консультация врача бесплат...

In [58]:

```
obj.mean().head()
```

Out[58]:

	effect	ad_id	word_count	power_up	mrt
campaign					
034ada41b6	7.200000	2.428555e+09	6.333333	56.686667	0.0
03fcaecd1d	5.250000	2.101706e+09	3.000000	29.125000	0.0
0697a81555	4.750000	2.101706e+09	3.000000	23.125000	0.0
08cdcb57a3	6.133333	2.101705e+09	3.666667	38.420000	0.0
0f0ba311fb	5.400000	2.101706e+09	3.000000	30.160000	0.0

Несколько функций в группировках

In [59]:

```
# задаем несколько функций сразу
```

```
stats.groupby('campaign').agg(['min', 'max'])['effect'].head()
```

Out[59]:

	min	max
campaign		
034ada41b6	4.1	9.0
03fcaecd1d	4.0	6.5
0697a81555	4.0	5.5
08cdcb57a3	5.5	7.4
0f0ba311fb	4.4	6.4

In [60]:

разные метрики для разных столбцов

```
results = stats.groupby('campaign').agg({'effect': ['min', 'max'], 'power_up': 'mean'})
results.head()
```

Out[60]:

	effect		power_up
	min	max	mean
campaign			
034ada41b6	4.1	9.0	56.686667
03fcaecd1d	4.0	6.5	29.125000
0697a81555	4.0	5.5	23.125000
08cdcb57a3	5.5	7.4	38.420000
0f0ba311fb	4.4	6.4	30.160000

Как обращаться к вложенным столбцам

In [61]:

```
results['effect']['max'].head()
```

Out[61]:

```
campaign
034ada41b6    9.0
03fcaecd1d    6.5
0697a81555    5.5
08cdcb57a3    7.4
0f0ba311fb    6.4
Name: max, dtype: float64
```

In [62]:

или даже так

```
stats.groupby('campaign').agg({'effect': ['min', 'max'], 'power_up': 'mean'})['effect']['ma
```

Out[62]:

```
campaign
034ada41b6    9.0
03fcaecd1d    6.5
0697a81555    5.5
08cdcb57a3    7.4
0f0ba311fb    6.4
Name: max, dtype: float64
```

In [63]:

группировка по нескольким столбцам

stats.groupby(['group', 'campaign']).count().head()

Out[63]:

		phrase	effect	ad_id	title	text	link	word_count	power_up	mrt
group	campaign									
1 5 москва мрт тесла	777160ba51	1	1	1	1	1	1	1	1	1
	8e77a0e565	1	1	1	1	1	1	1	1	1
	f0079349af	1	1	1	1	1	1	1	1	1
1 градская мрт	7f35591a28	3	3	3	3	3	3	3	3	3
1 мрт нагатинский	8e77a0e565	1	1	1	1	1	1	1	1	1

In [64]:

максимальное число объявлений в одной группе

stats.groupby(['group', 'campaign']).count().sort_values('phrase', ascending=False).head()

Out[64]:

		phrase	effect	ad_id	title	text	link	word_count	power_up	mrt
group	campaign									
мрт на алексеевской кулаков переулок 13	7f35591a28	3	3	3	3	3	3	3	3	3
мрт каширское шоссе	7f35591a28	3	3	3	3	3	3	3	3	3
мрт кутузовский	7f35591a28	3	3	3	3	3	3	3	3	3
мрт куркинское шоссе 30	7f35591a28	3	3	3	3	3	3	3	3	3
мрт куркино	7f35591a28	3	3	3	3	3	3	3	3	3