



НЕТОЛОГИЯ
групп

ЗАНЯТИЕ 7

ФУНКЦИИ И ГРУППИРОВКИ В PANDAS

КОНСТАНТИН БАШЕВОЙ

Яндекс

Habr

@kpi_maker



ФУНКЦИИ

Встроенных методов не всегда достаточно

Например, в датафрейме надо достать кампанию из ссылки:

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

Вариант 1 – к столбцу

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

```
def watcher(a):  
    """Мне только посмотреть"""  
    if a == 0:  
        return True  
  
    return False
```

Вариант 1 – к столбцу

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

```
def watcher(a):  
    """Мне только посмотреть"""  
    if a == 0:  
        return True  
  
    return False
```

Вариант 1 – к столбцу

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

```
def watcher(a):  
    """Мне только посмотреть"""  
    if a == 0:  
        return True  
  
    return False
```

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



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

а – значения столбца orders

Вариант 1 – к столбцу

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

Метод `apply` – аналог цикла, который проходит по всем строкам датафрейма и применяет к каждой функцию `watcher`

```
def watcher(a):  
    """Мне только посмотреть"""  
    if a == 0:  
        return True  
  
    return False
```


Вариант 2 – к строке

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

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

Вариант 2 – к строке

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

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

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

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

а – вся строка

Вариант 2 – к строке

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

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

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

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

а – вся строка

—

ГРУППИРОВКИ

Группировка

Для каждой страны считаем определенный показатель

Номер заказа	Страна	Стоимость
1	Россия	100
2	Китай	80
3	Китай	90
4	Россия	140
5	Россия	90



Страна	Метрика
Россия	...
Китай	...

Группировка

Для каждой страны считаем сумму продаж

Номер заказа	Страна	Стоимость
1	Россия	100
2	Китай	80
3	Китай	90
4	Россия	140
5	Россия	90



Страна	Сумма
Россия	330
Китай	170

Группировка

Для каждой страны считаем среднюю стоимость заказа

Номер заказа	Страна	Стоимость
1	Россия	100
2	Китай	80
3	Китай	90
4	Россия	140
5	Россия	90



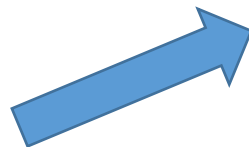
Страна	Среднее
Россия	110
Китай	85

—

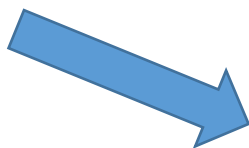
КАК ЭТО СЧИТАЕТСЯ

Каждому значению свой датафрейм

Номер заказа	Страна	Стоимость
1	Россия	100
2	Китай	80
3	Китай	90
4	Россия	140
5	Россия	90



Номер заказа	Страна	Стоимость
1	Россия	100
4	Россия	140
5	Россия	90



Номер заказа	Страна	Стоимость
2	Китай	80
3	Китай	90

О ЧЕМ ЭТО МЫ



Применяем функцию к каждому датафрейму

Номер заказа	Страна	Стоимость
1	Россия	100
4	Россия	140
5	Россия	90

Номер заказа	Страна	Стоимость
2	Китай	80
3	Китай	90

Применяем функцию к каждому датафрейму

Номер заказа	Страна	Стоимость
1	Россия	100
4	Россия	140
5	Россия	90

apply



Страна	Сумма
Россия	330

Номер заказа	Страна	Стоимость
2	Китай	80
3	Китай	90

apply



Страна	Сумма
Китай	170

Применяем функцию к каждому датафрейму

Номер заказа	Страна	Стоимость
1	Россия	100
4	Россия	140
5	Россия	90

apply



Страна	Сумма
Россия	330



Страна	Сумма
Россия	330
Китай	170

Номер заказа	Страна	Стоимость
2	Китай	80
3	Китай	90

apply



Страна	Сумма
Китай	170

