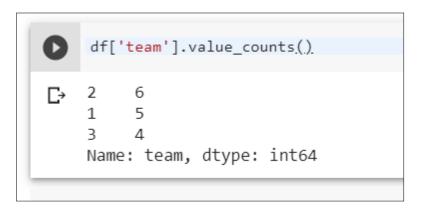
Анализ данных с Python

Лабораторная работа 2 - Pandas DataFrame: группировка данных и сводные таблицы

Создадим новый Jupiter ноутбук $lab_2.ipynb$

Продублируем первую ячейку из предыдущей работы, в которой создается датафрейм:

Подсчитаем сколько у нас строк по каждой бригаде, используя метод $value\ counts$:



Видим что 6 чел. во 2-й бригаде, 5 - во 1-й и 4 в 3-й.

Если хотим выводить по номерам бригад, то вот так:

```
df['team'].value_counts().sort_index()

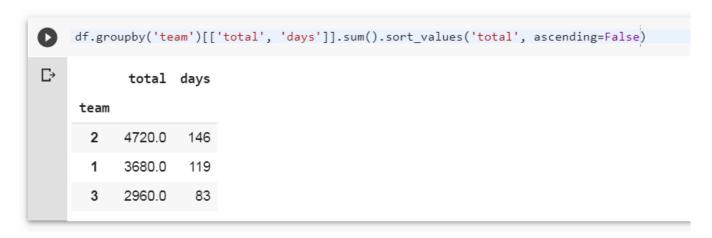
1  5
2  6
3  4
Name: team, dtype: int64
```

Теперь попробуем сгруппировать сумму зарплаты по бригадам. Применим метод **groupby**

Если нас интересует помимо зарплаты еще суммарное количество отработанных дней:

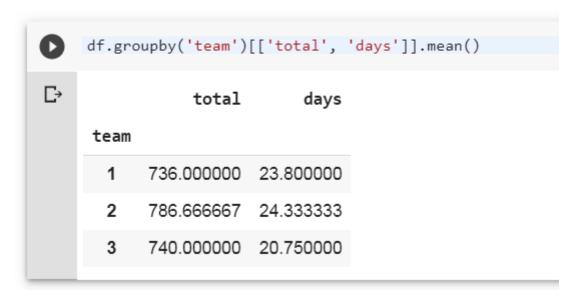


А теперь отсортируем вывод по убыванию зарплаты:



(если по возрастанию, то ascending=True)

Кроме сумм мы может получить средние значения, используя mean() вместо sum():



Теперь перейдем к созданию сводных таблиц (pivot_table).

Допустим, мы хотим получить анализ зарплаты в разрезе «бригада-разряд» в табличном виде, где номера бригады будут в строках таблицы, номера разрядов - в колонках:



То же самое, но с итогами по строкам и колонкам:

```
C→
   category
                     All
         4
              5
                  6
     team
     1
         1100 1620
                960
                    3680.0
     2
         1400 1560 1760
                   4720.0
     3
           0 1080 1880
                    2960.0
    ΑII
         2500 4260 4600 11360.0
```

Поменяем сумму на среднее значение:

0	df.pivot_t	able(\	/alues aggfun	='tot ic='me	al', index=' an', fill_va	team', columns='category' lue=0, margins=True)
₽	category team	4	5	6	All	
	1	550	810	960	736.000000	
	2	700	780	880	786.666667	
	3	0	540	940	740.000000	
	All	625	710	920	757.333333	

```
₽
           sum
                              mean
                        All
                                         All
               5 6
                              4
                                 5
                                     6
   category 4
      team
      1
           1100
               1620
                    960
                         3680.0 550
                                  810
                                     960 736.000000
      2
           1400
               1560
                   1760
                         4720.0 700
                                  780
                                     880
                                         786.666667
      3
               1080
                   1880
                         2960.0
                                  540 940
                                        740.000000
             0
                                0
           2500 4260 4600 11360.0 625 710 920 757.333333
     ΑII
```

Индивидуальное задание

Продолжить исследование своего датафрейма, используя рассмотренные здесь функции count_values, grpupby и pivot_table