

Работа с несколькими источниками данных

1/4

Использование списков, словарей, датафреймов и объектов Series для получения срезов

Выбирает строки датафрейма **data**, содержащие в столбце **column** значения из списка **our_list**

```
our_list = [1, 2, 3, 4]
data.query('column in @our_list')
```

Выбирает строки датафрейма **data**, содержащие в столбце **column** значения из ключей словаря **our_dict**

```
our_dict = {0: 1, 4: 82, 71: 1414}
data.query('column in @our_dict')
```

Выбирает строки датафрейма **data**, содержащие в столбце **column** значения из индексов столбца (**Series**) **our_dataframe**

```
our_series = pd.Series([81, 12, 64])
data.query(
    'column in @our_series.index'
)
```

Выбирает строки датафрейма **data**, содержащие в столбце **column** значения из столбца (**Series**) **our_dataframe**

```
our_series = pd.Series([81, 12, 64])
data.query(
    'column in @our_series'
)
```

Выбирает строки датафрейма **data**, содержащие в столбце **column** значения из индексов датафрейма **our_dataframe**

```
our_dataframe = pd.DataFrame(
    {'column1': [0, 1, 10, 11, 12],
     'column2': [5, 4, 3, 2, 1]}
)
data.query('column in @our_dataframe.index')
```

Работа с несколькими источниками данных

2/4

Проверяет, встречаются ли в столбце одного датафрейма значения из столбца другого

```
our_dataframe = pd.DataFrame(  
    {'column_1': [0, 1, 10, 11, 12],  
     'column_2': [5, 4, 3, 2, 1]  
})  
data.query('column in @our_dataframe.column_2')
```

Построение гистограммы с дополнительными параметрами

```
good_stations_stat.plot(  
    kind='hist',  
    y='time_spent',  
    histtype='step', # Вид гистограммы. Другие виды: 'bar', 'barstacked', 'stepfilled'  
    range=(0, 500),  
    bins=25,  
    linewidth=5, # толщина линий графика в пикселях  
    alpha=0.7, # прозрачность, число от 0 до 1  
    label='filtered', # название линии  
    ax=ax, # оси, на которые наносится график  
    grid=True,  
    legend=True # выводить ли легенду к графику  
)
```

Возврат крайних значений группы

Подробное объяснение, как это работает, есть в уроке «Объединяем данные из двух таблиц».

```
df.pivot_table(  
    index='index_column', values='values_column', aggfunc='first'  
) # первое значение  
df.pivot_table(  
    index='index_column', values='values_column', aggfunc='last'  
) # последнее значение
```

Работа с несколькими источниками данных

Переименование всех столбцов датафрейма

```
df.columns = ['name_1', 'name_2', ...]
```

Объединение датафреймов методами `merge()` и `join()`

```
first_df.merge(  
    second_df,  
    on='column', # столбец, по которому производим объединение  
    how='inner', # режим объединения  
    suffixes=('_первый датафрейм', '_второй датафрейм') # окончания названий столбцов  
)
```

```
first_df.join(  
    second_df,  
    on='column', # название столбца first_df для объединения с индексами second_df  
    how='left', # режим объединения  
    lsuffix='_первый датафрейм',  
    rsuffix='_второй датафрейм' # окончания названий столбцов  
)
```

Работа с несколькими источниками данных

4/4

Режимы объединения методов `merge()` и `join()`:

- Режим `inner` собирает значения, которые есть в обоих датафреймах. По умолчанию установлен в методе `merge()`.
- Режим `outer` собирает все значения, которые есть хотя бы в одном из датафреймов.
- Режим `left` предполагает, что в результат слияния войдут все строки из левого датафрейма. Строки из правого датафрейма сохранятся, только если значения в столбце, по которому происходит объединение, совпадают со значениями в левом датафрейме. По умолчанию установлен в методе `join()`.
- Режим `right` работает зеркально: в результат слияния обязательно войдут все строки из правого датафрейма. Строки из левого датафрейма сохранятся, только если значения в столбце, по которому происходит объединение, совпадают со значениями в правом датафрейме.

```
# Объединяем записи двух учеников о списке литературы. Применяем тип объединения 'outer'
# В результате получим все произведения, даже если автор был только в одном из списков
first_pupil_df.merge(second_pupil_df, on='author', how='outer')
```