

Шпаргалка по теме «Работа с несколькими источниками данных»

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

Проверяет, есть ли значение в списке `our_list`

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

Проверяет, есть ли значение среди ключей словаря `our_dict`

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

Проверяет, есть ли значение среди индексов объекта `Series`

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

Проверяет, есть ли значение среди значений объекта `Series`

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

Проверяет, есть ли значение среди индексов датафрейма

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

Проверяет, есть ли значение среди значений колонки датафрейма

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

Построение гистограммы с дополнительными параметрами

```
data.plot(  
    kind='hist',  
    y='column',  
    histtype='step', # тип гистограммы  
    range=(y_min, y_max),  
    bins=n_bins,  
    linewidth=our_linewidth, # толщина линий графика в пикселях  
    alpha=our_alpha, # густота заливки, число от 0 до 1  
    label='label', # название линии  
    ax=our_ax, # оси, по которым строится график  
    grid=True,  
    legend=True # выводить ли легенду к графику  
)
```

Возврат крайних значений группы

```
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
```

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

Метод `merge()` объединяет датафреймы по столбцам, а `join()` — по индексам.

```
first_df.merge(
    second_df,
    on='column', # название столбца, по которому объединять, передают в параметре on
    how='inner', # режим объединения передают в параметре how
    suffixes=('_первый датафрейм', '_второй датафрейм') # окончания названий столбцов зада
ют в параметре suffixes
)
```

```
first_df.join(
    second_df,
    on='column', # название столбца, по индексам которого объединять, передают в параметре
on
    how='left', # режим объединения передают в параметре how
    lsuffix='_первый датафрейм',
    rsuffix='_второй датафрейм' # окончания названий столбцов задают в параметре suffixes
)
```

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 обязательно войдут все строки из левого датафрейма. Строки из правого датафрейма сохранятся только в том случае, если значения в столбце, по которому происходит объединение, совпадают со значениями в левом датафрейме. По умолчанию установлен в методе `join()`.
- Режим `right` предполагает, что в результат слияния обязательно войдут все строки из левого датафрейма. Строки из правого датафрейма сохранятся только в том случае, если значения в столбце, по которому происходит объединение, совпадают со значениями в левом датафрейме.