Изучение срезов данных

Проверка наличия элементов списка lst в столбце

```
In  data['column'].isin(lst)
```

Работа с датой и временем

```
In  # Получить ...
    data['datetime'].dt.date    #дату
    data['datetime'].dt.year    #год
    data['datetime'].dt.weekday #день нед.
```

Быстрое получение срезов данных

```
In  data.query('column != "value"')
    data.query('column < column.mean()')
```

```
In  variable = 2
    data.query('column > @variable')
```

Поставить значение года на первое место

```
In  pd.to_datetime(df['datetime'],
                   yearfirst = True)
```

Сдвиг даты и времени

```
In  data['shifted_dt'] = data['datetime'] + pd.Timedelta(hours=10) # добавить 10 часов
```

Округление времени

```
In  data['datetime'] = data['datetime'].dt.round('1H')  # округлить до 1 часа
    data['datetime'] = data['datetime'].dt.round('1D')  # округлить до 1 дня
    data['datetime'] = data['datetime'].dt.round('5T')  # округлить до 5 минут
    data['datetime'] = data['datetime'].dt.round('10S') # округлить до 10 секунд
    data['datetime'] = data['datetime'].dt.floor('1H')  # округлять в меньшую сторону
    data['datetime'] = data['datetime'].dt.ceil('1H')   # округлять в большую сторону
```

Построение графиков по датафрейму

```
In  data.plot(x='column1',    # столбец значений для горизонтальной оси
              y='column2',    # столбец значений для вертикальной оси
              style='o-',     # стиль заполнения:'o'(точечный) или  'o-'(точечно-линейный)
              xlim=(0, 30),   # границы по оси X
              ylim=(30, 0),   # границы по оси Y
              figsize=(4, 5), # размеры картинки: (x_size, y_size)
              grid=True)      # отображать сетку или нет
```

Словарь

Баг-репорт (сообщение об ошибке)
сообщение, содержащее полную информацию об ошибке в программе, на сайте или в системе: *суть ошибки и где, когда, при каких условиях она была обнаружена*

Срез данных
часть данных из предоставленного набора, отобранная по определённым условиям