Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Радиотехнический»

КАФЕДРА ИУ-5 «Системы обработки информации и управления»

Отчет по лабораторной работе № 1 по курсу
Технологии машинного обучения

Тема работы: "Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных**"**

Выполнил: Мирсонов В. А.
Группа: РТ5-61Б

Дата выполнения: «» 2021 г.

Подпись:_____

Проверил: Гапанюк Ю. Е.

Дата проверки: «» 2021 г.

Подпись:_____

Москва, 2021 г.

Содержание

Цель лабораторной работы - изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Описание задания

1. Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
2. Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:
    - обработку пропусков в данных;
    - кодирование категориальных признаков;
    - масштабирование данных.

Ход выполнения лабораторной работы

Лабораторная работа №2

Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.

Цель лабораторной работы: изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Задание:

1. Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
2. Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:
   - обработку пропусков в данных;
   - кодирование категориальных признаков;
   - масштабирование данных.

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Загрузка и первичный анализ данных

Используем данные информации об измерениях загрязнения воздуха в Сеуле (Корея).

In [2]:

```
# Будем использовать только обучающую выборку
data = pd.read_csv('Measurement_summary.csv', sep=",")
```

In [3]:

```
# размер набора данных
data.shape
```

Out[3]:

```
(647511, 11)
```

In [4]:

```
# типы колонок
data.dtypes
```

Out[4]:

```
Measurement date      object
Station code           int64
Address               object
Latitude             float64
Longitude            float64
SO2                  float64
NO2                  float64
O3                   float64
CO                   float64
PM10                 float64
PM2.5                float64
dtype: object
```

In [5]:

```
# проверим есть ли пропущенные значения
data.isnull().sum()
```

Out[5]:

```
Measurement date      0
Station code          0
Address               0
Latitude              0
Longitude             0
SO2                   0
NO2                   0
O3                    0
CO                    0
PM10                329
PM2.5                 0
dtype: int64
```

In [6]:

```
# Первые 5 строк
датасета data.head()
```

Out[6]:

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.059	0.002	1.2	73.0	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.058	0.002	1.2	71.0	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	59.0
3	2017-01-01 03:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	58.0
4	2017-01-01 04:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.003	0.051	0.002	1.2	69.0	61.0

In [7]:

```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 647511
```

1.Обработка пропусков в данных

1.1. Простые стратегии - удаление или заполнение нулями

In [8]:

```
# Удаление колонок, содержащих пустые
значения data_new_1 = data.dropna(axis=1,
how='any') (data.shape, data_new_1.shape)
```

Out[8]:

```
((647511, 11), (647511, 10))
```

In [9]:

```
#Проверим колонки после удаления нужных
data_new_1.head()
```

Out[9]:

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.059	0.002	1.2	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.058	0.002	1.2	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	59.0
3	2017-01-01 03:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	58.0
4	2017-01-01 04:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.003	0.051	0.002	1.2	61.0

In [10]:

```
# Удаление строк, содержащих пустые
значения data_new_2 = data.dropna(axis=0,
how='any') (data.shape, data_new_2.shape)
```

Out[10]:

```
((647511, 11), (647182, 11))
```

In [11]:

```
#Проверим строки после удаления нужных
data_new_2.head()
```

Out[11]:

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.059	0.002	1.2	73.0	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.058	0.002	1.2	71.0	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	59.0
3	2017-01-01 03:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	58.0
4	2017-01-01 04:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.003	0.051	0.002	1.2	69.0	61.0

In [12]:

```
# Заполнение всех пропущенных значений нулями
# Однако, в данном случае так потсупать -
# некорректно, так как нулями заполняются в том числе
категориальные колонки data_new_3 = data.fillna(0)
data_new_3.head()
```

Out[12]:

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.059	0.002	1.2	73.0	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.058	0.002	1.2	71.0	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	59.0
3	2017-01-01 03:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	58.0
4	2017-01-01 04:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.003	0.051	0.002	1.2	69.0	61.0

1.2. "Внедрение значений" - импьютация (imputation)

1.2.1. Обработка пропусков в числовых данных

In [13]:

```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col, dt, temp_null_coun
```

```
Колонка PM10. Тип данных float64. Количество пустых значений 329, 0.05%.
```

In [14]:

```
# Фильтр по колонкам с пропущенными
значениями data_num = data[num_cols]
data_num
```

Out[14]:

	PM10
0	73.0
1	71.0
2	70.0
3	70.0
4	69.0
5	70.0
6	66.0
7	71.0
8	72.0
9	74.0
10	76.0
11	83.0

	PM10
12	93.0
13	94.0
14	93.0
15	87.0
16	87.0
17	91.0
18	91.0
19	92.0
20	94.0
21	93.0
22	89.0
23	91.0
24	93.0
25	92.0
26	90.0
27	92.0
28	92.0
29	92.0
...	...
647481	54.0
647482	47.0
647483	40.0
647484	35.0
647485	28.0
647486	30.0
647487	43.0
647488	36.0
647489	38.0
647490	43.0
647491	42.0
647492	31.0
647493	28.0
647494	25.0
647495	25.0
647496	20.0
647497	20.0
647498	18.0
647499	19.0
647500	22.0
647501	23.0
647502	24.0
647503	27.0
647504	27.0
647505	24.0
647506	23.0
647507	25.0
647508	24.0
647509	25.0
647510	27.0

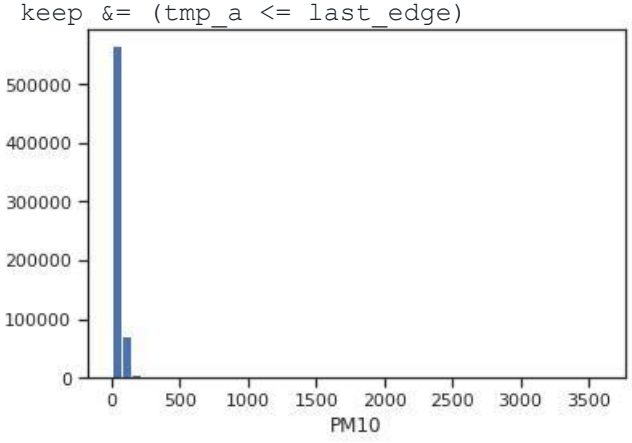
PM10

647511 rows × 1 columns

In [15]:

```
# Гистограмма по признакам
for col in data_num:
    plt.hist(data[col],
    50) plt.xlabel(col)
    plt.show()
```

```
/home/mark/.local/lib/python3.7/site-packages/numpy/lib/histograms.py:824:
RuntimeWarning: invalid value encou ntered in greater_equal
  keep = (tmp_a >= first_edge)
/home/mark/.local/lib/python3.7/site-packages/numpy/lib/histograms.py:825: RuntimeWarning: invalid value encou
ntered in less_equal
  keep &= (tmp_a <= last_edge)
```



In [16]:

```
# Фильтр по пустым значениям поля
MasVnrArea data[data['PM10'].isnull()]
```

Out[16]:

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
33421	2017-11-10 04:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.057	0.002	0.8	NaN	29.0
33422	2017-11-10 05:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.059	0.002	0.8	NaN	29.0
33423	2017-11-10 06:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.065	0.002	1.0	NaN	30.0
33424	2017-11-10 07:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.065	0.002	0.9	NaN	29.0
33425	2017-11-10 08:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.006	0.070	0.003	1.2	NaN	36.0
33426	2017-11-10 09:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.006	0.071	0.002	1.2	NaN	34.0
33427	2017-11-10 10:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.006	0.073	0.004	1.0	NaN	37.0
33428	2017-11-10 11:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.045	0.017	0.7	NaN	26.0
33429	2017-11-10 12:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.039	0.022	0.6	NaN	28.0
33430	2017-11-10 13:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.028	0.5	NaN	20.0
33431	2017-11-10 14:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.027	0.4	NaN	17.0
33432	2017-11-10 15:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.030	0.021	0.5	NaN	16.0
33433	2017-11-10 16:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.026	0.020	0.6	NaN	23.0
33434	2017-11-10 17:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.019	0.025	0.5	NaN	18.0
33435	2017-11-10 18:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.022	0.023	0.5	NaN	16.0

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
33436	2017-11-10 19:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.030	0.016	0.5	NaN	16.0
33437	2017-11-10 20:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.023	0.020	0.4	NaN	12.0
33438	2017-11-10 21:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.014	0.027	0.3	NaN	9.0
33439	2017-11-10 22:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.015	0.024	0.3	NaN	9.0
33440	2017-11-10 23:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.014	0.024	0.3	NaN	9.0
33441	2017-11-11 00:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.014	0.024	0.3	NaN	9.0
33442	2017-11-11 01:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.012	0.024	0.3	NaN	8.0
33443	2017-11-11 02:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.010	0.025	0.3	NaN	7.0
33444	2017-11-11 03:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.011	0.023	0.3	NaN	8.0
33445	2017-11-11 04:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.010	0.022	0.3	NaN	6.0
33446	2017-11-11 05:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.013	0.019	0.3	NaN	7.0
33447	2017-11-11 06:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.017	0.015	0.3	NaN	9.0
33448	2017-11-11 07:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.002	0.023	0.010	0.3	NaN	9.0
33449	2017-11-11 08:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.002	0.020	0.013	0.4	NaN	10.0
33450	2017-11-11 09:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.002	0.014	0.019	0.3	NaN	12.0
...	...	...	...	...	...	...	...	...	...	...	...
33720	2017-11-22 15:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.029	0.029	0.7	NaN	52.0
33721	2017-11-22 16:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.028	0.024	0.5	NaN	42.0
33722	2017-11-22 17:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.018	0.5	NaN	24.0
33723	2017-11-22 18:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.023	0.020	0.4	NaN	22.0
33724	2017-11-22 19:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.017	0.023	0.3	NaN	11.0
33725	2017-11-22 20:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.019	0.021	0.4	NaN	8.0
33726	2017-11-22 21:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.019	0.020	0.4	NaN	10.0
33727	2017-11-22 22:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.019	0.020	0.4	NaN	9.0
33728	2017-11-22 23:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.018	0.020	0.4	NaN	10.0
33729	2017-11-23 00:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.017	0.021	0.4	NaN	7.0
33730	2017-11-23 01:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.015	0.022	0.4	NaN	10.0
33731	2017-11-23 02:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.012	0.025	0.4	NaN	11.0
33732	2017-11-23 03:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.012	0.024	0.4	NaN	12.0
33733	2017-11-23 04:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.013	0.023	0.4	NaN	11.0
33734	2017-11-23 05:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.008	0.5	NaN	10.0
33735	2017-11-23 06:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic	37.564263	126.974676	0.004	0.038	0.002	0.5	NaN	12.0

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
33736	2017-11-23 07:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.041	0.002	0.7	NaN	11.0
33737	2017-11-23 08:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.040	0.003	0.6	NaN	12.0
33738	2017-11-23 09:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.040	0.004	0.7	NaN	14.0
33739	2017-11-23 10:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.040	0.005	0.6	NaN	17.0
33740	2017-11-23 11:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.016	0.023	0.3	NaN	15.0
33741	2017-11-23 12:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.009	0.030	0.3	NaN	10.0
33742	2017-11-23 13:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.009	0.029	0.3	NaN	10.0
33743	2017-11-23 14:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.011	0.028	0.3	NaN	11.0
33744	2017-11-23 15:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.013	0.026	0.3	NaN	10.0
33745	2017-11-23 16:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.016	0.022	0.3	NaN	10.0
33746	2017-11-23 17:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.027	0.013	0.3	NaN	8.0
33747	2017-11-23 18:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.031	0.009	0.4	NaN	7.0
33748	2017-11-23 19:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.033	0.007	0.4	NaN	8.0
33749	2017-11-23 20:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.034	0.006	0.4	NaN	9.0

329 rows × 11 columns

In [17]:

```
# Запоминаем индексы строк с пустыми
значениями flt_index =
data[data['PM10'].isnull()].index flt_index
```

Out[17]:

```
Int64Index([33421, 33422, 33423, 33424, 33425, 33426,
            33427, 33428, 33429, 33430,
            ...
            33740, 33741, 33742, 33743, 33744, 33745, 33746, 33747, 33748,
            33749],
           dtype='int64', length=329)
```

In [18]:

```
# Проверяем что выводятся нужные строки
data[data.index.isin(flt_index)]
```

Out[18]:

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
33421	2017-11-10 04:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.057	0.002	0.8	NaN	29.0
33422	2017-11-10 05:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.059	0.002	0.8	NaN	29.0
33423	2017-11-10 06:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.065	0.002	1.0	NaN	30.0
33424	2017-11-10 07:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.065	0.002	0.9	NaN	29.0
33425	2017-11-10 08:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.006	0.070	0.003	1.2	NaN	36.0
33426	2017-11-10 09:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.006	0.071	0.002	1.2	NaN	34.0
33427	2017-11-10 10:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.006	0.073	0.004	1.0	NaN	37.0
33428	2017-11-10 11:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.045	0.017	0.7	NaN	26.0

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
33429	2017-11-10 12:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.039	0.022	0.6	NaN	28.0
33430	2017-11-10 13:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.028	0.5	NaN	20.0
33431	2017-11-10 14:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.027	0.4	NaN	17.0
33432	2017-11-10 15:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.030	0.021	0.5	NaN	16.0
33433	2017-11-10 16:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.026	0.020	0.6	NaN	23.0
33434	2017-11-10 17:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.019	0.025	0.5	NaN	18.0
33435	2017-11-10 18:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.022	0.023	0.5	NaN	16.0
33436	2017-11-10 19:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.030	0.016	0.5	NaN	16.0
33437	2017-11-10 20:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.023	0.020	0.4	NaN	12.0
33438	2017-11-10 21:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.014	0.027	0.3	NaN	9.0
33439	2017-11-10 22:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.015	0.024	0.3	NaN	9.0
33440	2017-11-10 23:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.014	0.024	0.3	NaN	9.0
33441	2017-11-11 00:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.014	0.024	0.3	NaN	9.0
33442	2017-11-11 01:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.012	0.024	0.3	NaN	8.0
33443	2017-11-11 02:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.010	0.025	0.3	NaN	7.0
33444	2017-11-11 03:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.011	0.023	0.3	NaN	8.0
33445	2017-11-11 04:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.010	0.022	0.3	NaN	6.0
33446	2017-11-11 05:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.013	0.019	0.3	NaN	7.0
33447	2017-11-11 06:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.017	0.015	0.3	NaN	9.0
33448	2017-11-11 07:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.002	0.023	0.010	0.3	NaN	9.0
33449	2017-11-11 08:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.002	0.020	0.013	0.4	NaN	10.0
33450	2017-11-11 09:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.002	0.014	0.019	0.3	NaN	12.0
...	...	...	...	...	...	...	...	...	...	...	...
33720	2017-11-22 15:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.029	0.029	0.7	NaN	52.0
33721	2017-11-22 16:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.005	0.028	0.024	0.5	NaN	42.0
33722	2017-11-22 17:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.018	0.5	NaN	24.0
33723	2017-11-22 18:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.023	0.020	0.4	NaN	22.0
33724	2017-11-22 19:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.017	0.023	0.3	NaN	11.0
33725	2017-11-22 20:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.019	0.021	0.4	NaN	8.0
33726	2017-11-22 21:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.019	0.020	0.4	NaN	10.0
33727	2017-11-22 22:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.019	0.020	0.4	NaN	9.0

	date	code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
33728	2017-11-22 23:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic	37.564263	126.974676	0.003	0.018	0.020	0.4	NaN	10.0
33729	2017-11-23 00:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.017	0.021	0.4	NaN	7.0
33730	2017-11-23 01:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.015	0.022	0.4	NaN	10.0
33731	2017-11-23 02:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.012	0.025	0.4	NaN	11.0
33732	2017-11-23 03:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.012	0.024	0.4	NaN	12.0
33733	2017-11-23 04:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.013	0.023	0.4	NaN	11.0
33734	2017-11-23 05:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.029	0.008	0.5	NaN	10.0
33735	2017-11-23 06:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.038	0.002	0.5	NaN	12.0
33736	2017-11-23 07:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.041	0.002	0.7	NaN	11.0
33737	2017-11-23 08:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.040	0.003	0.6	NaN	12.0
33738	2017-11-23 09:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.040	0.004	0.7	NaN	14.0
33739	2017-11-23 10:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.040	0.005	0.6	NaN	17.0
33740	2017-11-23 11:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.016	0.023	0.3	NaN	15.0
33741	2017-11-23 12:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.004	0.009	0.030	0.3	NaN	10.0
33742	2017-11-23 13:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.009	0.029	0.3	NaN	10.0
33743	2017-11-23 14:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.011	0.028	0.3	NaN	11.0
33744	2017-11-23 15:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.013	0.026	0.3	NaN	10.0
33745	2017-11-23 16:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.016	0.022	0.3	NaN	10.0
33746	2017-11-23 17:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.027	0.013	0.3	NaN	8.0
33747	2017-11-23 18:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.031	0.009	0.4	NaN	7.0
33748	2017-11-23 19:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.033	0.007	0.4	NaN	8.0
33749	2017-11-23 20:00	102	15, Deoksugung-gil, Jung-gu, Seoul, Republic o...	37.564263	126.974676	0.003	0.034	0.006	0.4	NaN	9.0

329 rows × 11 columns

In [19]:

```
# фильтр по колонке
data_num[data_num.index.isin(flt_index)]['PM10']
```

Out[19]:

```
33421  NaN
33422 NaN
33423 NaN
33424 NaN
33425 NaN
33426 NaN
33427 NaN
33428 NaN
33429 NaN
33430 NaN
33431 NaN
33432 NaN
33433 NaN
33434 NaN
33435 NaN
33436 NaN
33437 NaN
33438 NaN
33439 NaN
33440 NaN
33441 NaN
33442 NaN
33443 NaN
33444 NaN
33445 NaN
33446 NaN
33447 NaN
33448 NaN
33449 NaN
33450 NaN
       ..
33720 NaN
33721 NaN
33722 NaN
33723 NaN
33724 NaN
33725 NaN
33726 NaN
33727 NaN
33728 NaN
33729 NaN
33730 NaN
33731 NaN
33732 NaN
33733 NaN
33734 NaN
33735 NaN
33736 NaN
33737 NaN
33738 NaN
33739 NaN
33740 NaN
33741 NaN
33742 NaN
33743 NaN
33744 NaN
33745 NaN
33746 NaN
33747 NaN
33748 NaN
33749 NaN
Name: PM10, Length: 329, dtype: float64
```

Будем использовать встроенные средства импьютации библиотеки scikit-learn - https://scikit-learn.org/stable/modules/impute.html#impute

In [20]:

```
data_num_PM10 = data_num[['PM10']]
data_num_PM10.head()
```

Out[20]:

	PM10
0	73.0
1	71.0
2	70.0
3	70.0
4	69.0

In [21]:

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

In [22]:

```
# Фильтр для проверки заполнения пустых
значений indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_PM10)
mask_missing_values_only
```

Out[22]:

```
array([[False],
       [False],
       [False],
        ...,
        [False],
        [False],
        [False]])
```

С помощью класса SimpleImputer можно проводить импьютацию различными показателями центра распределения

In [23]:

```
strategies=['mean', 'median','most_frequent']
```

In [24]:

```
def test_num_impute(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_num_PM10)
    return data_num_imp[mask_missing_values_only]
```

In [25]:

```
strategies[0], test_num_impute(strategies[0])
```

Out[25]:

```
('mean',
 array([43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206, 43.71126206,
        43.71126206, 43.71126206, 43.71126206, 43.71126206]))
```

In [26]:

```
strategies[1], test_num_impute(strategies[1])
```

Out[26]:

```
('median',
 array([35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
   35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
              35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35., 35.,
        35., 35., 35., 35.]))
```

In [27]:

```
strategies[2], test_num_impute(strategies[2])
```

Out[27]:

```
('most_frequent',
 array([27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
   27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
              27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27., 27.,
        27., 27., 27., 27.]))
```

In [28]:

```
# Более сложная функция, которая позволяет задавать колонку и вид
импьютации def test_num_impute_col(dataset, column, strategy_param):
    temp_data = dataset[[column]]
    indicator = MissingIndicator()
    mask_missing_values_only = indicator.fit_transform(temp_data)
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(temp_data)
    filled_data = data_num_imp[mask_missing_values_only]

    return column, strategy_param, filled_data.size, filled_data[0], filled_data[filled_data.size-1]
```

In [29]:

```
data[['PM10']].describe()
```

Out[29]:

	PM10
count	647182.000000
mean	43.711262
std	71.153913
min	-1.000000
25%	22.000000
50%	35.000000
75%	53.000000
max	3586.000000

In [30]:

```
test_num_impute_col(data, 'PM10', strategies[0])
```

Out[30]:

```
('PM10', 'mean', 329, 43.711262056114045, 43.711262056114045)
```

In [31]:

```
test_num_impute_col(data, 'PM10', strategies[1])
```

Out[31]:

```
('PM10', 'median', 329, 35.0, 35.0)
```

In [32]:

```
test_num_impute_col(data, 'PM10', strategies[2])
```

Out[32]:

```
('PM10', 'most_frequent', 329, 27.0, 27.0)
```

1.2.2. Обработка пропусков в категориальных данных

Для обработки пропусков в категоральных данных будем исопльзовать другой датасет (battles.csv), содержащий информацию о битвах из серии книг "Песнь льда и пламени" (Сериал "Игра престолов")

In [35]:

```
# Будем использовать только обучающую выборку
data = pd.read_csv('battles.csv', sep=",")
```

In [60]:

```
data.head()
```

Out[60]:

	name	year	battle_number	attacker_king	defender_king	attacker_1	attacker_2	defender_1	attacker_outcome	battle_type	attacker
0	Battle of the Golden Tooth	298	1	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	Tully	1	pitched battle	Ja
1	Battle at the Mummer's Ford	298	2	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	Baratheon	1	ambush	G
2	Battle of Riverrun	298	3	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	Tully	1	pitched battle	Ja
3	Battle of the Green Fork	298	4	Robb Stark	Joffrey/Tommen Baratheon	Stark	NaN	Lannister	0	pitched battle	Roose Man
4	Battle of the Whispering Wood	298	5	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully	Lannister	1	ambush	Robb S

In [36]:

```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
```

```
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col, dt, temp_null_coun
```

```
Колонка attacker_king. Тип данных object. Количество пустых значений 2, 0.0%.
Колонка defender_king. Тип данных object. Количество пустых значений 3, 0.0%.
Колонка attacker_2. Тип данных object. Количество пустых значений 28, 0.0%.
Колонка defender_1. Тип данных object. Количество пустых значений 1, 0.0%.
Колонка battle_type. Тип данных object. Количество пустых значений 18, 0.0%.
Колонка attacker_commander. Тип данных object. Количество пустых значений 1, 0.0%.
Колонка location. Тип данных object. Количество пустых значений 1, 0.0%.
```

Какие из этих колонок Вы бы выбрали или не выбрали для построения модели?

Для категориальных признаков со стратегиями "most_frequent" или "constant", будем использовать класс SimpleImputer.

In [37]:

```
cat_temp_data = data[['battle_type']]
cat_temp_data.head()
```

Out[37]:

	battle_type
0	pitched battle
1	ambush
2	pitched battle
3	pitched battle
4	ambush

In [38]:

```
cat_temp_data['battle_type'].unique()
```

Out[38]:

```
array(['pitched battle', 'ambush', 'siege', nan, 'razing'], dtype=object)
```

In [39]:

```
cat_temp_data[cat_temp_data['battle_type'].isnull()].shape
```

Out[39]:

```
(18, 1)
```

In [40]:

```
# Импьютация наиболее частыми значениями
imp2 = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
data_imp2 = imp2.fit_transform(cat_temp_data)
data_imp2
```

Out[40]:

```
array([['pitched battle'],
       ['ambush'],
       ['pitched battle'],
       ['pitched battle'],
       ['ambush'],
       ['ambush'],
       ['pitched battle'],
       ['pitched battle'],
       ['siege'],
       ['ambush'],
       ['pitched battle'],
       ['ambush'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['pitched battle'],
       ['razing'],
       ['siege'],
       ['siege'],
       ['siege'],
       ['siege']], dtype=object)
```

In [41]:

```
# Пустые значения отсутствуют
np.unique(data_imp2)
```

Out[41]:

```
array(['ambush', 'pitched battle', 'razing', 'siege'], dtype=object)
```

In [42]:

```
# Импьютация константой
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value='!!!')
data_imp3 = imp3.fit_transform(cat_temp_data)
data_imp3
```

Out[42]:

```
array([['pitched battle'],
       ['ambush'],
       ['pitched battle'],
       ['pitched battle'],
       ['ambush'],
       ['ambush'],
       ['pitched battle'],
       ['pitched battle'],
       ['siege'],
       ['ambush'],
       ['pitched battle'],
       ['ambush'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['!!!!'],
       ['pitched battle'],
       ['pitched battle'],
       ['razing'],
       ['siege'],
       ['siege'],
       ['siege'],
       ['siege']], dtype=object)
```

In [43]:

```
np.unique(data_imp3)
```

Out[43]:

```
array(['!!!!', 'ambush', 'pitched battle', 'razing', 'siege'], dtype=object)
```

In [44]:

```
data_imp3[data_imp3=='!!!!'].size
```

Out[44]:

```
18
```

2. Преобразование категориальных признаков в числовые

In [45]:

```
cat_enc = pd.DataFrame({'c1':data_imp2.T[0]})
cat_enc
```

Out[45]:

	c1
0	pitched battle
1	ambush
2	pitched battle
3	pitched battle
4	ambush
5	ambush
6	pitched battle
7	pitched battle
8	siege
9	ambush
10	pitched battle
11	ambush
12	pitched battle
13	pitched battle
14	pitched battle
15	pitched battle
16	pitched battle
17	pitched battle
18	pitched battle
19	pitched battle
20	pitched battle
21	pitched battle
22	pitched battle
23	pitched battle
24	pitched battle
25	pitched battle
26	pitched battle
27	pitched battle
28	pitched battle
29	pitched battle
30	pitched battle
31	pitched battle
32	razing
33	siege
34	siege
35	siege
36	siege

2.1. Кодирование категорий целочисленными значениями - label encoding

In [46]:

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

In [47]:

```
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['c1'])
```

In [48]:

```
cat_enc['c1'].unique()
```

Out[48]:

```
array(['pitched battle', 'ambush', 'siege', 'razing'], dtype=object)
```

In [49]:
```
np.unique(cat_enc_le)
```

Out[49]:
```
array([0, 1, 2, 3])
```

In [50]:
```
le.inverse_transform([0, 1, 2, 3])
```

Out[50]:
```
array(['ambush', 'pitched battle', 'razing', 'siege'], dtype=object)
```

2.2. Кодирование категорий наборами бинарных значений - one-hot encoding

In [51]:
```
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['c1']])
```

In [52]:
```
cat_enc.shape
```

Out[52]:
```
(37, 1)
```

In [53]:
```
cat_enc_ohe.shape
```

Out[53]:
```
(37, 4)
```

In [54]:
```
cat_enc_ohe
```

Out[54]:
```
<37x4 sparse matrix of type '<class 'numpy.float64'>'
 with 37 stored elements in Compressed Sparse Row format>
```

In [55]:
```
cat_enc_ohe.todense()[0:10]
```

Out[55]:
```
matrix([[0., 1., 0., 0.],
    [1., 0., 0., 0.],
        [0., 1., 0., 0.],
        [0., 1., 0., 0.],
        [1., 0., 0., 0.],
        [1., 0., 0., 0.],
        [0., 1., 0., 0.],
        [0., 1., 0., 0.],
        [0., 0., 0., 1.],
        [1., 0., 0., 0.]])
```

In [56]:
```
cat_enc.head(10)
```

Out[56]:

	c1
0	pitched battle
1	ambush
2	pitched battle
3	pitched battle
4	ambush
5	ambush
6	pitched battle
7	pitched battle
8	siege
9	ambush

2.3. Pandas get_dummies - быстрый вариант one-hot кодирования

In [57]:
```
pd.get_dummies(cat_enc).head()
```

Out[57]:

	c1_ambush	c1_pitched battle	c1_razing	c1_siege
0	0	1	0	0
1	1	0	0	0
2	0	1	0	0
3	0	1	0	0
4	1	0	0	0

In [58]:

```
pd.get_dummies(cat_temp_data, dummy_na=True).head()
```

Out[58]:

	battle_type_ambush	battle_type_pitched battle	battle_type_razing	battle_type_siege	battle_type_nan
0	0	1	0	0	0
1	1	0	0	0	0
2	0	1	0	0	0
3	0	1	0	0	0
4	1	0	0	0	0

In [ ]:

3.Масштабирование данных

Термины "масштабирование" и "нормализация" часто используются как синонимы. Масштабирование предполагает изменение диапазона измерения величины, а нормализация - изменение распределения этой величины.

Если признаки лежат в различных диапазонах, то необходимо их нормализовать. Как правило, применяют два подхода:

- MinMax масштабирование: $$ x_{новый} = \frac{x_{старый} - min(X)}{max(X)-min(X)} $$ В этом случае значения лежат в диапазоне от 0 до 1.
- Масштабирование данных на основе Z-оценки: $$ x_{новый} = \frac{x_{старый} - AVG(X)}{\sigma(X)} $$ В этом случае большинство значений попадает в диапазон от -3 до 3.

где X - матрица объект-признак, $AVG(X)$ - среднее значение, σ - среднеквадратичное отклонение.

In [59]:

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

3.1. MinMax масштабирование

Для минимаксного масштабирования будем использовать датасет, уже рассматриваемый ранее при обработке пропусков данных.

In [67]:

```
data = pd.read_csv('Measurement_summary.csv', sep=",")
data.head()
```

Out[67]:

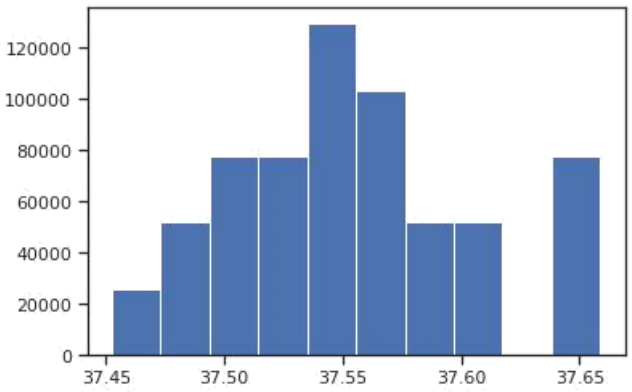
	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.059	0.002	1.2	73.0	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.058	0.002	1.2	71.0	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	59.0
3	2017-01-01 03:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.004	0.056	0.002	1.2	70.0	58.0
4	2017-01-01 04:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005007	0.003	0.051	0.002	1.2	69.0	61.0

In [68]:

```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['PM2.5']])
```

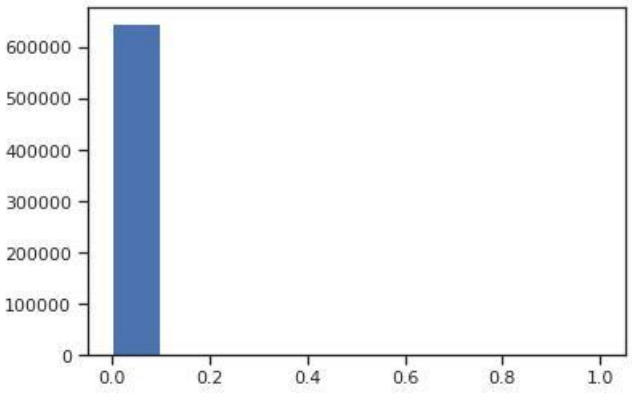
In [76]:

```
plt.hist(data['Latitude'], 10)
plt.show()
```



In [77]:

```
plt.hist(sc1_data, 10)
plt.show()
```



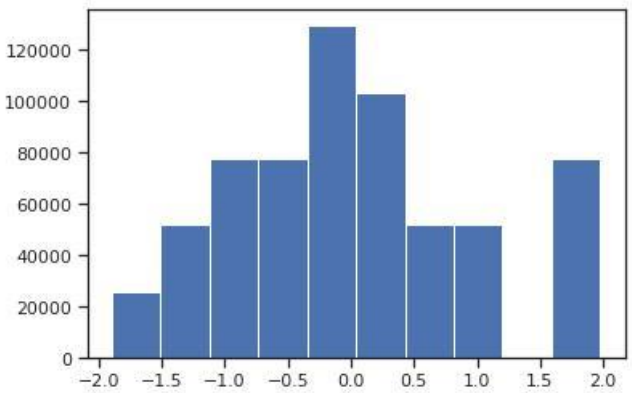
3.2. Масштабирование данных на основе Z-оценки - StandardScaler

In [81]:

```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['Latitude']])
```

In [82]:

```
plt.hist(sc2_data, 10)
plt.show()
```



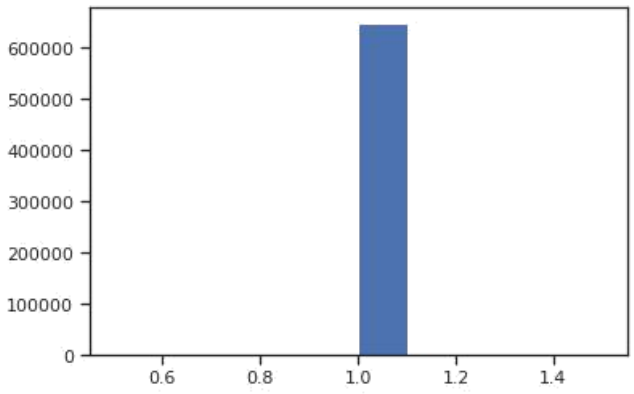
3.3. Нормализация данных

In [83]:

```
sc3 = Normalizer()
sc3_data = sc3.fit_transform(data[['Latitude']])
```

In [84]:

```
plt.hist(sc3_data, 10)
plt.show()
```


600000
500000
400000
300000
200000
100000
0
0.6
0.8
1.0
1.2
1.4