Рубежный контроль №1

Варианты заданий

Вариант	Номер задачи №1	Номер задачи №2
8	8	28

Задача №8.

Для набора данных проведите устранение пропусков для одного (произвольного) числового признака с использованием метода заполнения модой.

Задача №28.

Для набора данных для одного (произвольного) числового признака проведите обнаружение и замену (найденными верхними и нижними границами) выбросов на основе межквартильного размаха.

Доп задание

Для студентов групп ИУ5-21М, ИУ5И-21М - для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Dataset:https://www.kaggle.com/alexanderklarge/london-westminster-hourly-pollution-2010july-2020

Context

UK Government data from the London Westminster air pollution sensor. Data source can be found here. Note the data actually goes back to 2001. Interactive map of UK DEFRA air pollution sensors.

Content

2010 to July 2020 air pollution data. Metrics like ozone and sulphur were measured in the early years but have since been abandoned. Nitric oxide and Nitrogen dioxide are still measured.

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df = pd.read_csv('Westminster.csv')
df
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2718:
DtypeWarning: Columns (14,15,22,23) have mixed types.Specify dtype option on
```

```
import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[ ]:

	Unnamed: 0	Carbon monoxide	Daily measured PM_{2.5} (uncorrected)	Date	Nitric oxide	Nitrogen dioxide	Nitrogen oxides as nitrogen dioxide	Ozone	P (
0	0	NaN	NaN	2020-01-01	15.24691	46.18613	69.56440	NaN	
1	1	NaN	NaN	2020-01-01	17.63596	44.73371	71.77513	NaN	
2	2	NaN	NaN	2020-01-01	18.69909	44.28266	72.95419	NaN	
3	3	NaN	NaN	2020-01-01	16.54894	41.27891	66.65360	NaN	
4	4	NaN	NaN	2020-01-01	2.75219	30.76489	34.98485	NaN	
...	...	...	...	...	...	...	...	...	
92515	8755	0.9	NaN	2010-12-31	106.00000	92.00000	254.00000	4.0	
92516	8756	0.9	NaN	2010-12-31	44.00000	74.00000	141.00000	6.0	
92517	8757	0.8	NaN	2010-12-31	24.00000	63.00000	99.00000	8.0	
92518	8758	0.7	NaN	2010-12-31	19.00000	52.00000	80.00000	12.0	
92519	8759	0.3	NaN	2010-12-31	18.00000	46.00000	73.00000	20.0	

92520 rows × 25 columns

Ищем данные с пропусками

In [ ]:
```
df.isna().sum()
```

Out[ ]:
```
Unnamed: 0                                                  0
Carbon monoxide                                         71018
Daily measured PM<sub>2.5</sub> (uncorrected)           92520
Date                                                        0
Nitric oxide                                             8501
Nitrogen dioxide                                         8503
Nitrogen oxides as nitrogen dioxide                      8502
Ozone                                                   58512
PM<sub>2.5</sub> particulate matter (Hourly measured)   73502
Sulphur dioxide                                         71122
status                                                  12557
status.1                                                 8503
status.2                                                 8502
status.3                                                39280
status.4                                                67176
status.5                                                71122
status.6                                                92520
time                                                        0
unit                                                    12557
unit.1                                                   8503
unit.2                                                   8502
unit.3                                                  39280
unit.4                                                  67176
unit.5                                                  71122
```

```
unit.6                                                    92520
dtype: int64
```

Заполним пропуски в Nitric oxide с помощью моды

In [ ]:
```
print('Nitric oxide mode:', df['Nitric oxide'])
df['Nitric oxide'].fillna(df['Nitric oxide'].mode(), inplace=True)
```

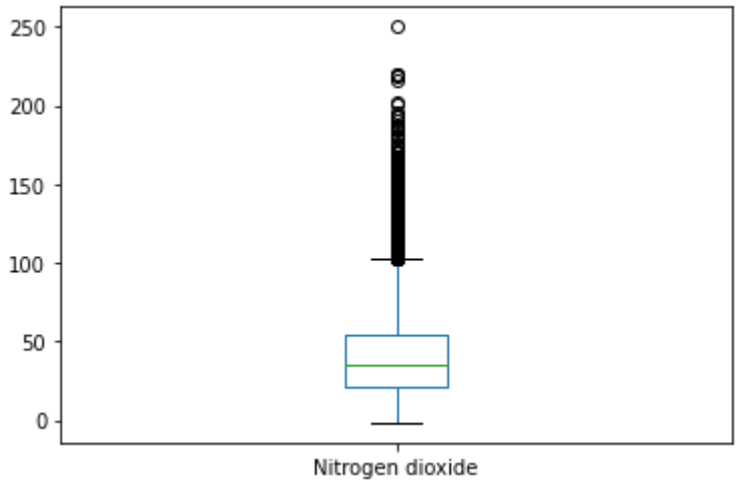
```
Nitric oxide mode: 0         15.24691
1          17.63596
2          18.69909
3          16.54894
4           2.75219
             ...
92515    106.00000
92516     44.00000
92517     24.00000
92518     19.00000
92519     18.00000
Name: Nitric oxide, Length: 92520, dtype: float64
```

Заменим выбросы в числовом признаке с использованием Nitrogen dioxide квартильного размаха

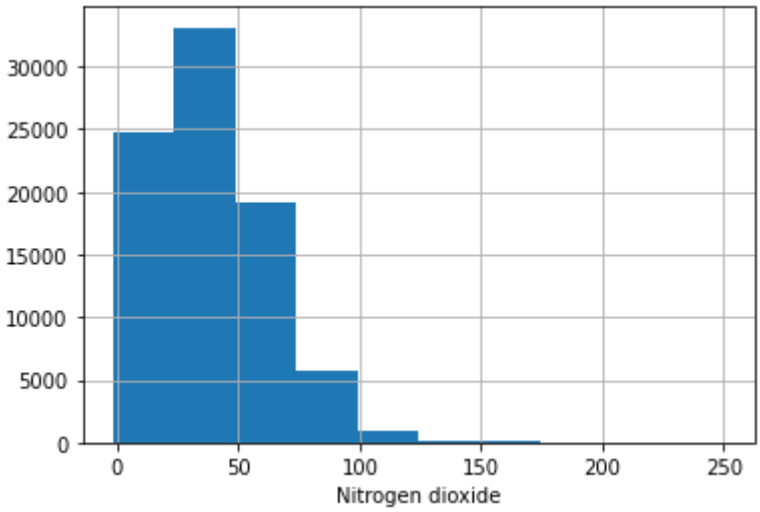
In [ ]:
```
q25, q75 = np.nanpercentile(df['Nitrogen dioxide'], [25 ,75])
iqr = q75 - q25
print(f'q25:{q25}, q75:{q75}, iqr: {iqr}')

high_clip_value = q75 + 1.5*iqr
low_clip_value = q25 - 1.5*iqr
print(low_clip_value, high_clip_value)
```

```
q25:21.0, q75:53.70587, iqr: 32.70587
-28.058804999999992 102.76467499999998
```

In [ ]:
```
df['Nitrogen dioxide'].plot(kind='box')
plt.show()
df['Nitrogen dioxide'].hist()
plt.xlabel('Nitrogen dioxide')
```

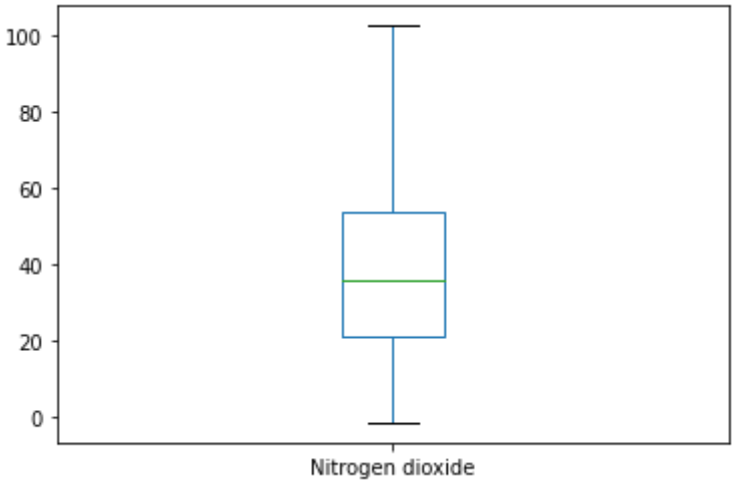


Out[ ]:
```
Text(0.5, 0, 'Nitrogen dioxide')
```



In [ ]:
```
clipped = df['Nitrogen dioxide'].clip(lower=low_clip_value, upper=high_clip_v
clipped.plot(kind='box')
```

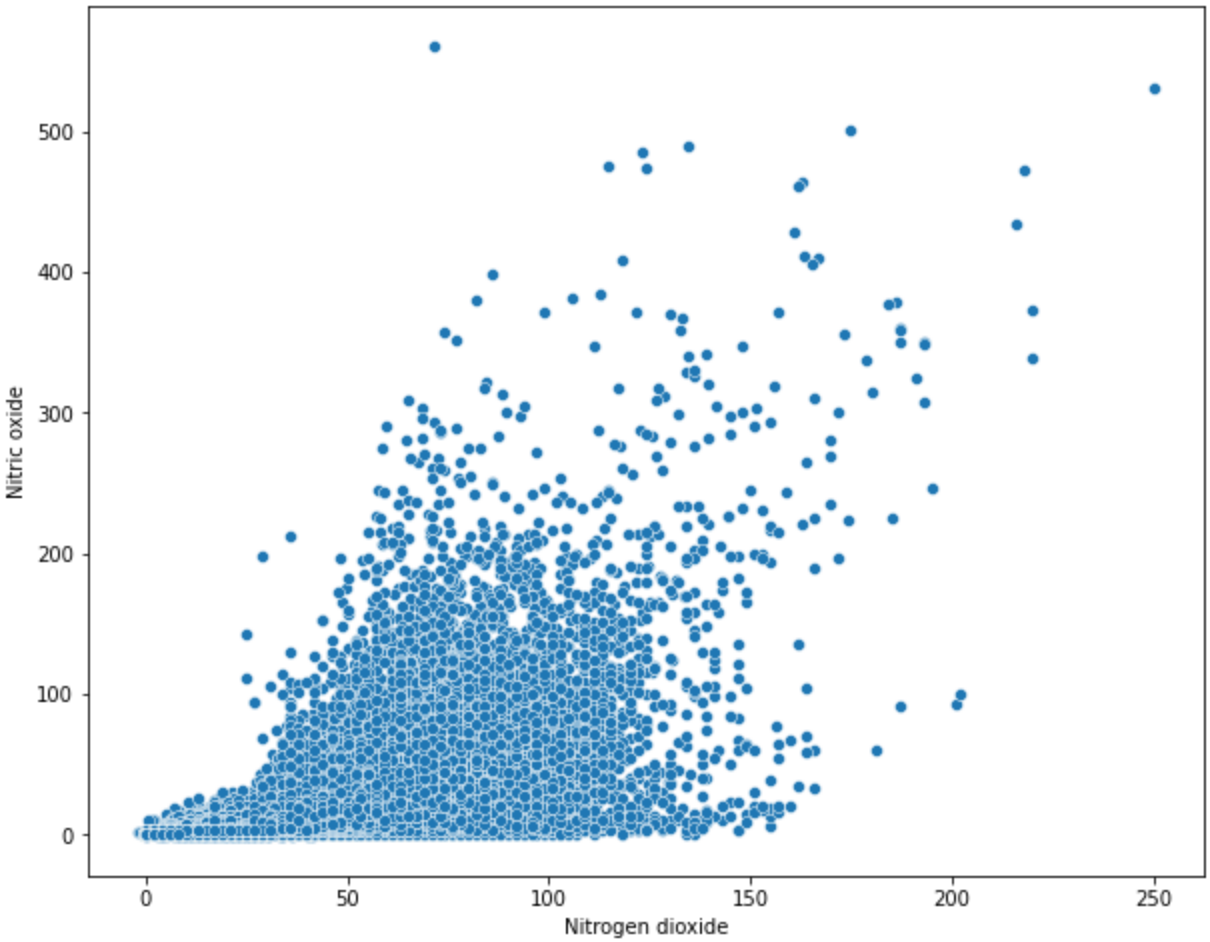
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f94ce835910>



In [ ]:
```
# from sklearn.preprocessing import RobustScaler

# scaler_with_iqr_using = RobustScaler()
# scaled = scaler_with_iqr_using.fit_transform(np.array(df['Nitrogen dioxide'
# pd.Series(scaled.reshape(df.shape[0])).plot(kind='box')
```

Построим диаграмму рассеяния для двух калонок Nitrogen dioxide и Nitric oxide

In [ ]:
```
plt.figure(figsize=(10,8))
sns.scatterplot(data=df, x="Nitrogen dioxide", y="Nitric oxide")
plt.show()
```


Nitric oxide
Nitrogen dioxide
0
100
200
300
400
500
0
50
100
150
200
250