РК ИУ5-63Б Киреев Андрей Вариант № 12

Условие задачи:

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
data = pd.read_csv('states_all.csv', sep=',')
```

In [4]:
```
data.head()
```

Out[4]:

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	ST
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	
1	1992_ALASKA	ALASKA	1992	NaN	1049591.0	106780.0	
2	1992_ARIZONA	ARIZONA	1992	NaN	3258079.0	297888.0	
3	1992_ARKANSAS	ARKANSAS	1992	NaN	1711959.0	178571.0	
4	1992_CALIFORNIA	CALIFORNIA	1992	NaN	26260025.0	2072470.0	

5 rows × 25 columns

In [5]:
```
data.dtypes
```

Out[5]:
```
PRIMARY_KEY                      object
STATE                            object
YEAR                              int64
ENROLL                          float64
TOTAL_REVENUE                   float64
FEDERAL_REVENUE                 float64
STATE_REVENUE                   float64
LOCAL_REVENUE                   float64
TOTAL_EXPENDITURE               float64
INSTRUCTION_EXPENDITURE         float64
SUPPORT_SERVICES_EXPENDITURE    float64
OTHER_EXPENDITURE               float64
CAPITAL_OUTLAY_EXPENDITURE      float64
GRADES_PK_G                     float64
GRADES_KG_G                     float64
GRADES_4_G                      float64
GRADES_8_G                      float64
GRADES_12_G                     float64
GRADES_1_8_G                    float64
GRADES_9_12_G                   float64
GRADES_ALL_G                    float64
AVG_MATH_4_SCORE                float64
AVG_MATH_8_SCORE                float64
AVG_READING_4_SCORE             float64
AVG_READING_8_SCORE             float64
dtype: object
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
PRIMARY_KEY                      0
STATE                            0
YEAR                             0
ENROLL                         491
TOTAL_REVENUE                  440
FEDERAL_REVENUE                440
STATE_REVENUE                  440
LOCAL_REVENUE                  440
TOTAL_EXPENDITURE              440
INSTRUCTION_EXPENDITURE        440
SUPPORT_SERVICES_EXPENDITURE   440
OTHER_EXPENDITURE              491
CAPITAL_OUTLAY_EXPENDITURE     440
GRADES_PK_G                    173
GRADES_KG_G                     83
GRADES_4_G                      83
GRADES_8_G                      83
GRADES_12_G                     83
GRADES_1_8_G                   695
GRADES_9_12_G                  644
GRADES_ALL_G                    83
AVG_MATH_4_SCORE              1150
AVG_MATH_8_SCORE              1113
AVG_READING_4_SCORE           1065
AVG_READING_8_SCORE           1153
dtype: int64
```

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1715 entries, 0 to 1714
Data columns (total 25 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   PRIMARY_KEY                     1715 non-null   object 
 1   STATE                           1715 non-null   object 
 2   YEAR                            1715 non-null   int64  
 3   ENROLL                          1224 non-null   float64
 4   TOTAL_REVENUE                   1275 non-null   float64
 5   FEDERAL_REVENUE                 1275 non-null   float64
 6   STATE_REVENUE                   1275 non-null   float64
 7   LOCAL_REVENUE                   1275 non-null   float64
 8   TOTAL_EXPENDITURE               1275 non-null   float64
 9   INSTRUCTION_EXPENDITURE         1275 non-null   float64
 10  SUPPORT_SERVICES_EXPENDITURE    1275 non-null   float64
 11  OTHER_EXPENDITURE               1224 non-null   float64
 12  CAPITAL_OUTLAY_EXPENDITURE      1275 non-null   float64
 13  GRADES_PK_G                     1542 non-null   float64
 14  GRADES_KG_G                     1632 non-null   float64
 15  GRADES_4_G                      1632 non-null   float64
 16  GRADES_8_G                      1632 non-null   float64
 17  GRADES_12_G                     1632 non-null   float64
 18  GRADES_1_8_G                    1020 non-null   float64
 19  GRADES_9_12_G                   1071 non-null   float64
 20  GRADES_ALL_G                    1632 non-null   float64
 21  AVG_MATH_4_SCORE                565 non-null    float64
 22  AVG_MATH_8_SCORE                602 non-null    float64
 23  AVG_READING_4_SCORE             650 non-null    float64
 24  AVG_READING_8_SCORE             562 non-null    float64
dtypes: float64(22), int64(1), object(2)
memory usage: 335.1+ KB
```

Удаляем ненужные столбцы

In [9]:
```
data.drop(['INSTRUCTION_EXPENDITURE', 'YEAR'], axis=1, inplace=True)
```

In [10]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1715 entries, 0 to 1714
Data columns (total 23 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   PRIMARY_KEY                     1715 non-null   object 
 1   STATE                           1715 non-null   object 
 2   ENROLL                          1224 non-null   float64
 3   TOTAL_REVENUE                   1275 non-null   float64
 4   FEDERAL_REVENUE                 1275 non-null   float64
 5   STATE_REVENUE                   1275 non-null   float64
 6   LOCAL_REVENUE                   1275 non-null   float64
 7   TOTAL_EXPENDITURE               1275 non-null   float64
 8   SUPPORT_SERVICES_EXPENDITURE    1275 non-null   float64
 9   OTHER_EXPENDITURE               1224 non-null   float64
 10  CAPITAL_OUTLAY_EXPENDITURE      1275 non-null   float64
 11  GRADES_PK_G                     1542 non-null   float64
 12  GRADES_KG_G                     1632 non-null   float64
 13  GRADES_4_G                      1632 non-null   float64
 14  GRADES_8_G                      1632 non-null   float64
 15  GRADES_12_G                     1632 non-null   float64
 16  GRADES_1_8_G                    1020 non-null   float64
 17  GRADES_9_12_G                   1071 non-null   float64
 18  GRADES_ALL_G                    1632 non-null   float64
 19  AVG_MATH_4_SCORE                565 non-null    float64
 20  AVG_MATH_8_SCORE                602 non-null    float64
 21  AVG_READING_4_SCORE             650 non-null    float64
 22  AVG_READING_8_SCORE             562 non-null    float64
dtypes: float64(21), object(2)
memory usage: 308.3+ KB
```

In [19]:

```
data['TOTAL_REVENUE'] = data['TOTAL_REVENUE'].replace(0, np.nan)
data['TOTAL_REVENUE'] = data['TOTAL_REVENUE'].fillna(data['TOTAL_REVENUE'].
```

In [26]:

```
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 / data.shape[0]) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {

data['STATE'] = data.fillna('Nane')
data.head()
```

Out[26]:

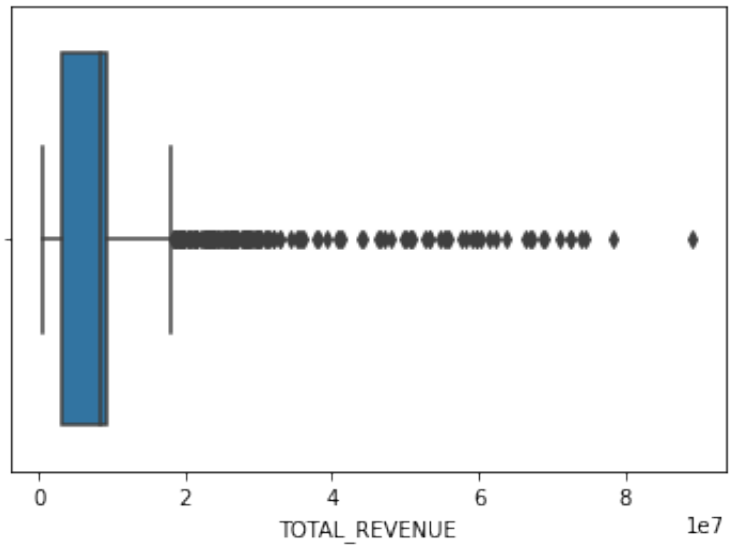
	PRIMARY_KEY	STATE	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STAT
0	1992_ALABAMA	1992_ALABAMA	NaN	2678885.0	304177.0	
1	1992_ALASKA	1992_ALASKA	NaN	1049591.0	106780.0	
2	1992_ARIZONA	1992_ARIZONA	NaN	3258079.0	297888.0	
3	1992_ARKANSAS	1992_ARKANSAS	NaN	1711959.0	178571.0	
4	1992_CALIFORNIA	1992_CALIFORNIA	NaN	26260025.0	2072470.0	

5 rows × 23 columns

Ящик с усами

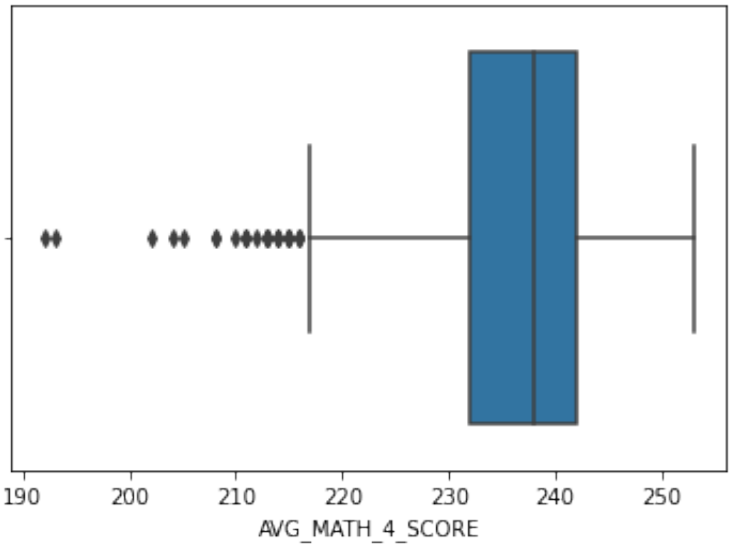
In [33]:
```
sns.boxplot(x=data['TOTAL_REVENUE'])
```

Out[33]:
```
<AxesSubplot:xlabel='TOTAL_REVENUE'>
```



In [35]:
```
sns.boxplot(x=data['AVG_MATH_4_SCORE'])
```

Out[35]: <AxesSubplot:xlabel='AVG_MATH_4_SCORE'>



In [ ]: