Рубежный контроль №1

Технологии разведочного анализа и обработки данных. Вариант 12

Выполнил Плешаков Владислав, РТ5-61Б

Задача: 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?

Датасет: https://www.kaggle.com/noriuk/us-education-datasets-unification-project

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer
```

Общая информация о данных

In [2]:
```
data = pd.read_csv('data/states_all.csv', sep=',')
```

In [3]:
```
data.head()
```

Out[3]:

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_R
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	1659028.0	
1	1992_ALASKA	ALASKA	1992	NaN	1049591.0	106780.0	720711.0	
2	1992_ARIZONA	ARIZONA	1992	NaN	3258079.0	297888.0	1369815.0	1
3	1992_ARKANSAS	ARKANSAS	1992	NaN	1711959.0	178571.0	958785.0	
4	1992_CALIFORNIA	CALIFORNIA	1992	NaN	26260025.0	2072470.0	16546514.0	7

5 rows × 25 columns

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

Out[4]:

	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTAL
count	1715.000000	1.224000e+03	1.275000e+03	1.275000e+03	1.275000e+03	1.275000e+03	
mean	2002.075219	9.175416e+05	9.102045e+06	7.677799e+05	4.223743e+06	4.110522e+06	
std	9.568621	1.066514e+06	1.175962e+07	1.146992e+06	5.549735e+06	5.489562e+06	
min	1986.000000	4.386600e+04	4.656500e+05	3.102000e+04	0.000000e+00	2.209300e+04	
25%	1994.000000	2.645145e+05	2.189504e+06	1.899575e+05	1.165776e+06	7.151210e+05	
50%	2002.000000	6.499335e+05	5.085826e+06	4.035480e+05	2.537754e+06	2.058996e+06	

75%	2010.000000	1.010532e+06	1.084516e+07	8.279320e+05	5.055548e+06	4.755293e+06
max	2019.000000	6.307022e+06	8.921726e+07	9.990221e+06	5.090457e+07	3.610526e+07

8 rows × 23 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
```

Заполнение пропусков

Посмотрим процент пропусков для каждой из колонок

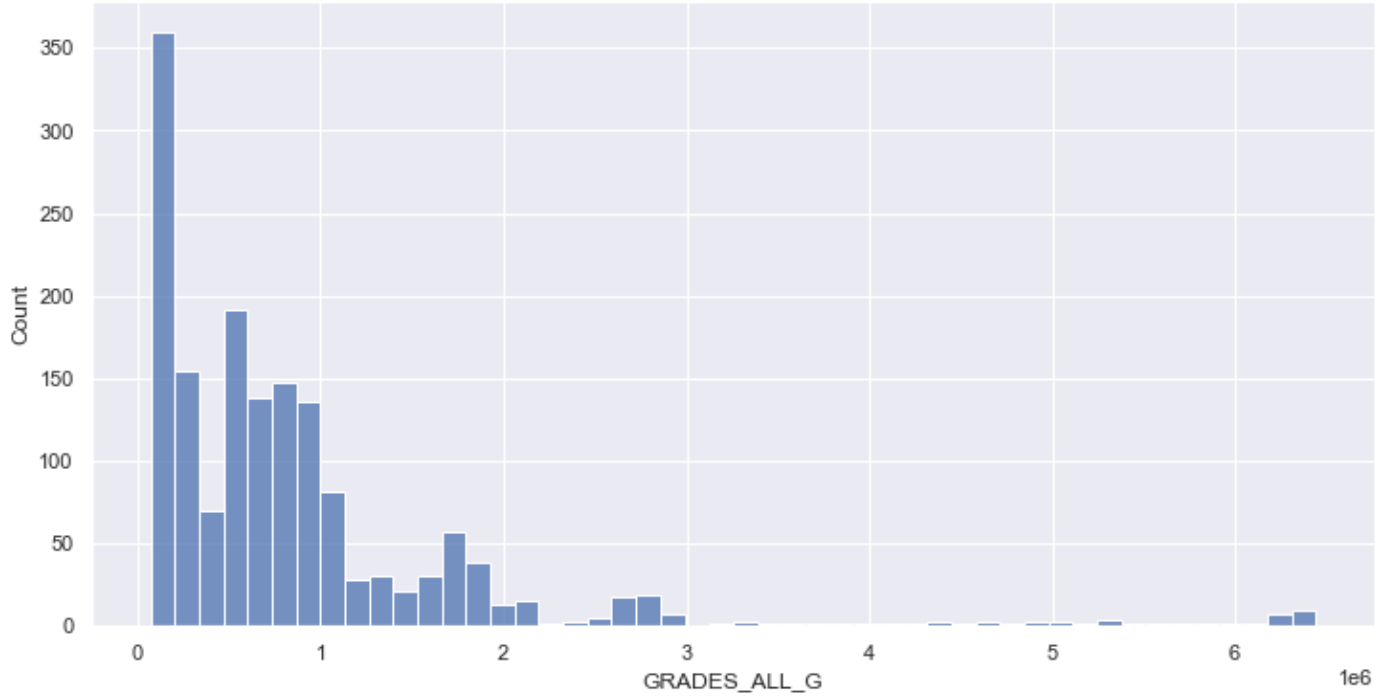
In [7]:
```
total_rows = data.shape[0]
for col in data.columns:
    null_count = data[data[col].isnull()].shape[0]
    col_type = str(data[col].dtype)
    print(f'Колонка {col}, тип {col_type}, процент пропусков {null_count / total_rows *
```

```
Колонка PRIMARY_KEY, тип object, процент пропусков 0.00%
Колонка STATE, тип object, процент пропусков 0.00%
Колонка YEAR, тип int64, процент пропусков 0.00%
Колонка ENROLL, тип float64, процент пропусков 28.63%
Колонка TOTAL_REVENUE, тип float64, процент пропусков 25.66%
Колонка FEDERAL_REVENUE, тип float64, процент пропусков 25.66%
Колонка STATE_REVENUE, тип float64, процент пропусков 25.66%
Колонка LOCAL_REVENUE, тип float64, процент пропусков 25.66%
Колонка TOTAL_EXPENDITURE, тип float64, процент пропусков 25.66%
Колонка INSTRUCTION_EXPENDITURE, тип float64, процент пропусков 25.66%
Колонка SUPPORT_SERVICES_EXPENDITURE, тип float64, процент пропусков 25.66%
Колонка OTHER_EXPENDITURE, тип float64, процент пропусков 28.63%
Колонка CAPITAL_OUTLAY_EXPENDITURE, тип float64, процент пропусков 25.66%
Колонка GRADES_PK_G, тип float64, процент пропусков 10.09%
Колонка GRADES_KG_G, тип float64, процент пропусков 4.84%
Колонка GRADES_4_G, тип float64, процент пропусков 4.84%
Колонка GRADES_8_G, тип float64, процент пропусков 4.84%
Колонка GRADES_12_G, тип float64, процент пропусков 4.84%
Колонка GRADES_1_8_G, тип float64, процент пропусков 40.52%
Колонка GRADES_9_12_G, тип float64, процент пропусков 37.55%
Колонка GRADES_ALL_G, тип float64, процент пропусков 4.84%
Колонка AVG_MATH_4_SCORE, тип float64, процент пропусков 67.06%
Колонка AVG_MATH_8_SCORE, тип float64, процент пропусков 64.90%
Колонка AVG_READING_4_SCORE, тип float64, процент пропусков 62.10%
Колонка AVG_READING_8_SCORE, тип float64, процент пропусков 67.23%
```

Пропусков в категориальных колонках нет. Для заполнения пропусков возьмем колонку GRADES_ALL_G. Сначала построим гистограмму данной колонки

In [8]:
```
sns.set(rc={"figure.figsize":(12, 6)})
sns.histplot(data=data['GRADES_ALL_G'])
```

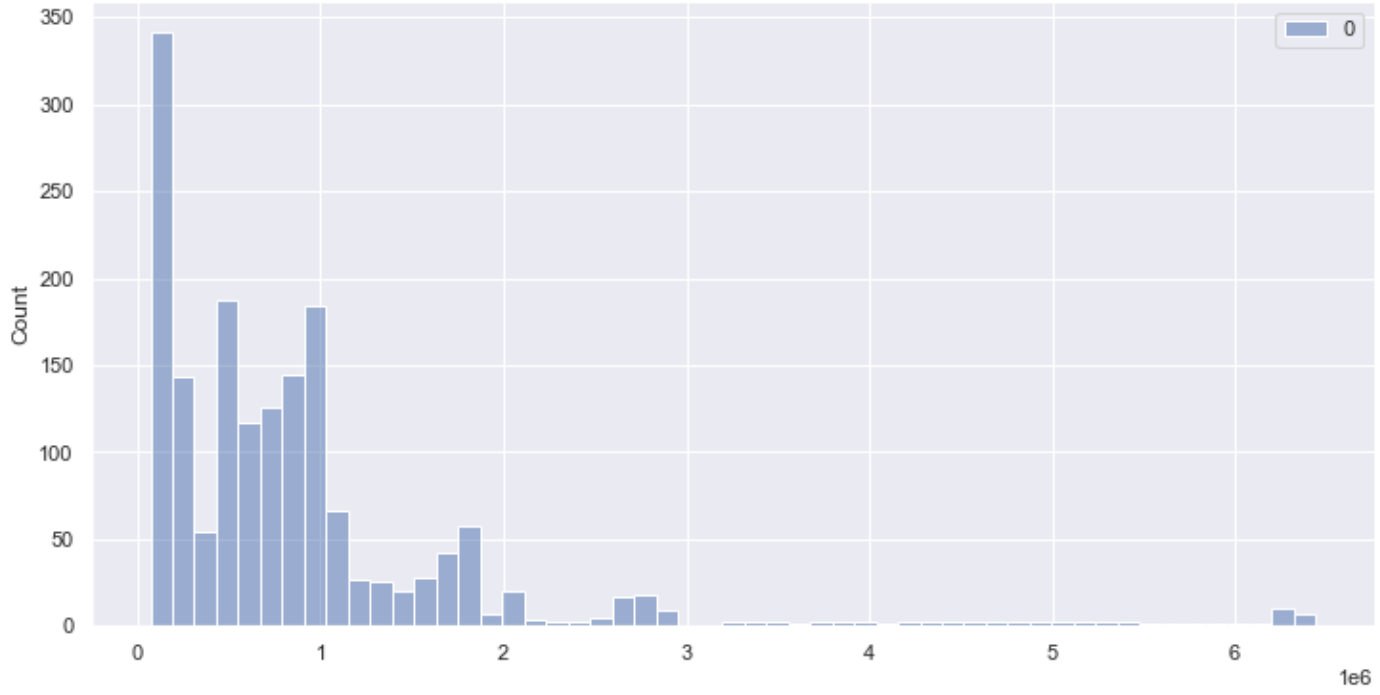
Out[8]:
```
<AxesSubplot:xlabel='GRADES_ALL_G', ylabel='Count'>
```



Заполним ее с применением различных стратегий

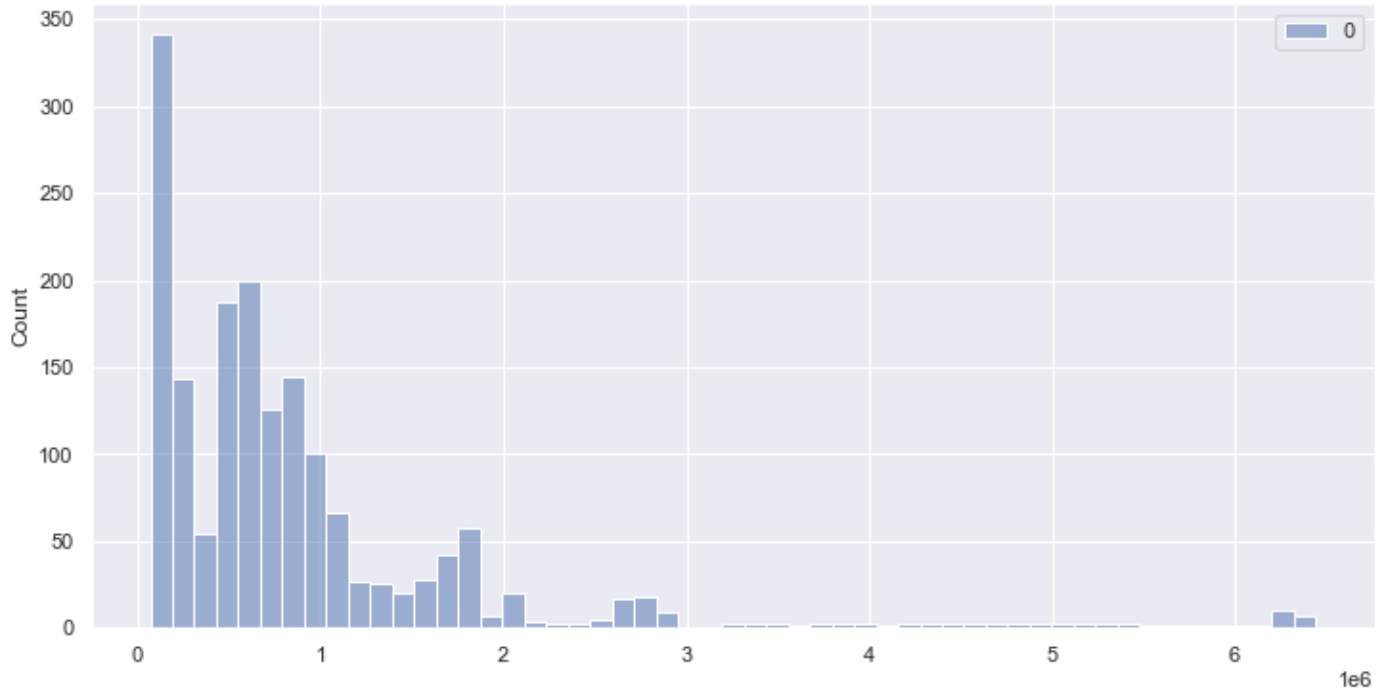
In [9]:
```
# Заполнение средним
mean_imp = SimpleImputer(strategy='mean')
tot_exp_mean = mean_imp.fit_transform(data[['GRADES_ALL_G']])
sns.histplot(data=tot_exp_mean)
```

Out[9]:
```
<AxesSubplot:ylabel='Count'>
```



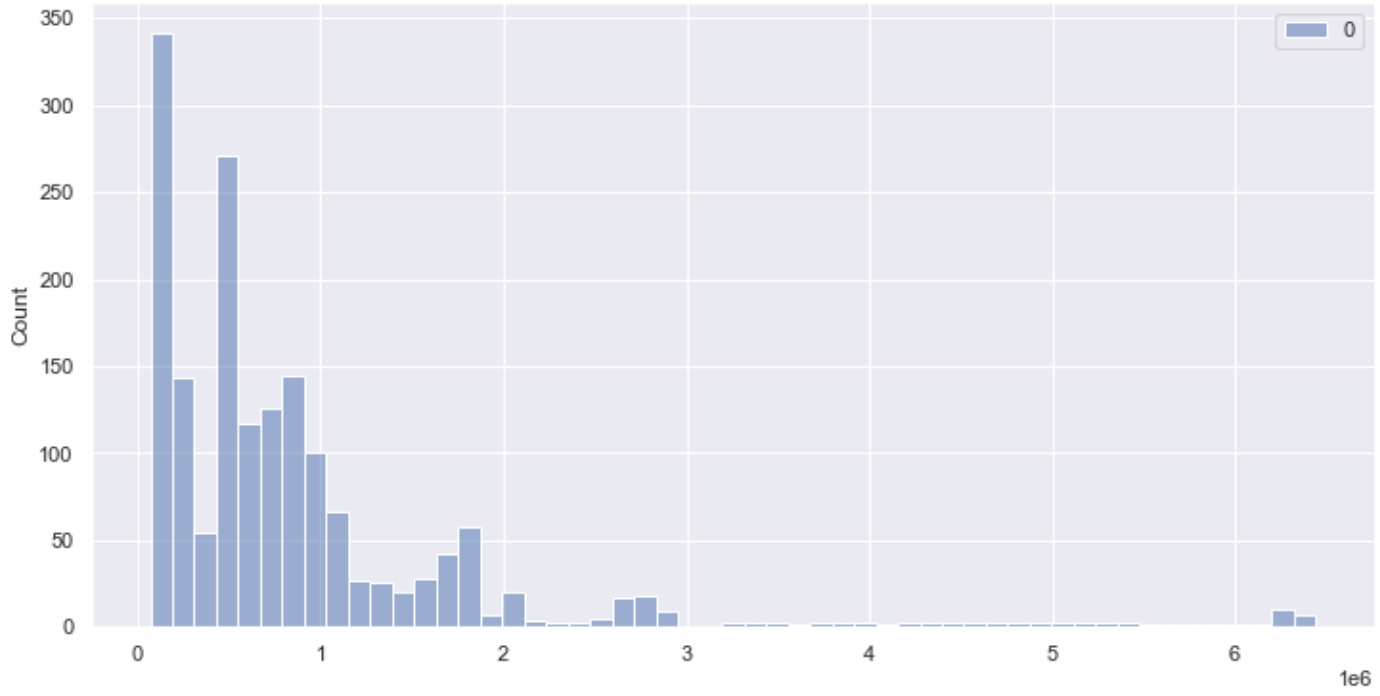
In [10]:
```
# Заполнение медианой
median_imp = SimpleImputer(strategy='median')
tot_exp_mean = median_imp.fit_transform(data[['GRADES_ALL_G']])
sns.histplot(data=tot_exp_mean)
```

Out[10]:
```
<AxesSubplot:ylabel='Count'>
```



In [11]:
```
# Заполнение модой
most_freq_imp = SimpleImputer(strategy='most_frequent')
tot_exp_mean = most_freq_imp.fit_transform(data[['GRADES_ALL_G']])
sns.histplot(data=tot_exp_mean)
```

Out[11]:
```
<AxesSubplot:ylabel='Count'>
```



Для обработки пропусков был использован класс SimpleImputer и рассмотрены три стратегии, которые он реализует: заполнение средним, медианой и модой. Для колонки GRADES_ALL_G, исходя из гистограмм, лучшего всего подходит заполнение средним, т.к. не так сильно влияет на плотность вероятности распределения.

Для заполнения пропусков в категориальных признаках также используется класс SimpleImputer, только в этом случае он реализует стратегии most frequent (заполнение самым часто встречаемым значением) и constant (заполнение некторой константой).

Для дальнейшего построения модели точно следует исключить признаки AVG_MATH_4_SCORE, AVG_MATH_8_SCORE, AVG_READING_4_SCORE и AVG_READING_8_SCORE, т.к. они имеют слишком много пропусков. Следует оставить колонки GRADES_PK_G, GRADES_KG_G, GRADES_4_G, GRADES_8_G, GRADES_12_G и GRADES_ALL_G т.к. в каждой из них меньше 5 процентов пропусков