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

ФАКУЛЬТЕТ Информатика, системы управления

КАФЕДРА Системы обработки информации и управления

Отчёт по РК1

По курсу
«Методы машинного обучения»

По теме «Методы обработки данных»

Вариант 9

Выполнила:

Студентка группы ИУ5-22М

Румак Д.П.

Проверил:

Гапанюк Ю.Е.

2025 г.

Варианты заданий

Номер варианта	Номер задачи №1	Номер задачи №2
9	9	29

Дополнительные требования

Для студентов групп ИУ5-22М, ИУ5И-22М - для произвольной колонки данных построить гистограмму.

Задание:

Задача №9

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

Задача №29

Для набора данных проведите удаление константных и псевдоконстантных признаков.

Решение:

Задача №9

Загрузим датасет и проведем первичный анализ. Посмотрим общую информацию о признаках.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data = pd.read_csv(r'uk_universities.csv')
```

```
data.head()
```

	Название университета	Регион	Год основания	Девиз	Национальный ранг	Мировой рейтинг	Оценка мировых рейтингов	Minimum
0	University of Cambridge	East of England	1209	From here, light and sacred draughts	1	4	94.1	
1	University of Oxford	South East England	1096	The Lord is my light	2	2	93.3	
2	University of St Andrews	Scotland	1413	Ever to excel	3	86	75.8	
3	Imperial College London	London	1907	Knowledge is the adornment and safeguard of th...	4	8	86.6	
4	Loughborough University	East Midlands	1966	With Truth, Knowledge and Labour	5	404	72.8	

```
data.shape
```

```
(126, 17)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126 entries, 0 to 125
Data columns (total 17 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Название университета                 126 non-null    object
 1   Регион                                126 non-null    object
 2   Год основания                         126 non-null    int64
 3   Девиз                                 112 non-null    object
 4   Национальный ранг                     126 non-null    int64
 5   Мировой рейтинг                       126 non-null    int64
 6   Оценка мировых рейтингов              82 non-null     float64
 7   Minimum_IELTS_score                   126 non-null    float64
 8   Иностранные студенты                  126 non-null    object
 9   Оценка студентов                      126 non-null    object
 10  Кол-во поступивших студентов (тыс.)   126 non-null    int64
 11  Кол-во преподавательского состава     126 non-null    float64
 12  Тип управления университета           126 non-null    object
 13  Местоположение кампуса                109 non-null    object
 14  Стоимость жизни в год                 126 non-null    int64
 15  Широта                                126 non-null    float64
 16  Долгота                               126 non-null    object
dtypes: float64(4), int64(5), object(8)
memory usage: 16.9+ KB
```

Выведем информацию о пропусках:

```
data.isnull().sum()
```

```
Название университета                 0
Регион                                0
Год основания                         0
Девиз                                14
Национальный ранг                     0
Мировой рейтинг                       0
Оценка мировых рейтингов             44
Minimum_IELTS_score                   0
Иностранные студенты                  0
Оценка студентов                      0
Кол-во поступивших студентов (тыс.)   0
Кол-во преподавательского состава     0
Тип управления университета           0
Местоположение кампуса               17
Стоимость жизни в год                 0
Широта                                0
Долгота                               0
dtype: int64
```

Выведем все числовые признаки и далее найдем числовые признаки, которые имеют пропущенные значения:

```
data.select_dtypes(include=[np.number]).columns.tolist()
```

```
['Год основания',
 'Национальный ранг',
 'Мировой рейтинг',
 'Оценка мировых рейтингов',
 'Minimum_IELTS_score',
 'Кол-во поступивших студентов (тыс.)',
 'Кол-во преподавательского состава',
 'Стоимость жизни в год',
 'Широта']
```

```
# Выберем числовые колонки с пропущенными значениями
num_cols = []
total_count = len(data)  # Общее количество записей

# Цикл по колонкам датасета
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.startswith('int') or dt.startswith('float') or dt == 'number'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print(f'Колонка {col}. Тип данных {dt}. Количество пустых значений {temp_null_count}, {temp_perc}%.')

num_cols
```

```
Колонка Оценка мировых рейтингов. Тип данных float64. Количество пустых значений 44, 34.92%.

['Оценка мировых рейтингов']
```

```
chosen_col = num_cols[0]
chosen_col
```

```
'Оценка мировых рейтингов'
```

Как видим, только один числовой признак содержит пропуски, поэтому будем работать с признаком «Оценка мировых рейтингов».

Перед тем как применять метод "хвоста распределения", проверим, действительно ли распределение асимметричное. Для этого:

1. **Посмотрим на коэффициент асимметрии (skewness)**

— Если $skewness > 1$ или $skewness < -1 \rightarrow$ сильная асимметрия.

— Если $0.5 < skewness < 1$ или $-1 < skewness < -0.5 \rightarrow$ умеренная асимметрия.

— Если $-0.5 < skewness < 0.5 \rightarrow$ распределение почти симметрично.

```
from scipy.stats import skew
# Убираем NaN перед расчетом skewness
col_data = data[chosen_col].dropna()
# Коэффициент асимметрии
skewness = skew(col_data)
print(f"Коэффициент асимметрии (skewness) для {chosen_col}: {skewness:.3f}")
```

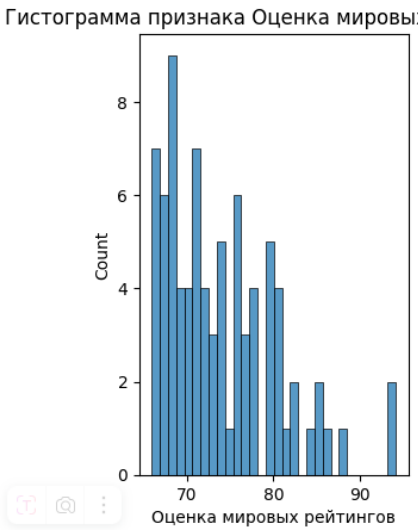
```
Коэффициент асимметрии (skewness) для Оценка мировых рейтингов: 0.965
```

По результатам можем сказать, что небольшая асимметрия есть. Это значит, что есть небольшой хвост.

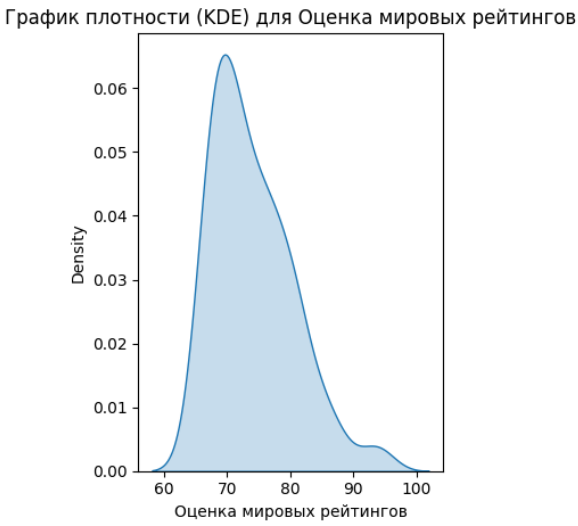
2. **Построим гистограмму и KDE-график, чтобы визуально проверить наличие "хвоста"**

```
# Гистограмма
plt.subplot(1, 2, 1)
sns.histplot(col_data, kde=False, bins=30)
plt.title(f"Гистограмма признака {chosen_col}")
```

```
Text(0.5, 1.0, 'Гистограмма признака Оценка мировых рейтингов')
```




```
 # KDE-график (плотность распределения)
plt.subplot(1, 2, 2)
sns.kdeplot(col_data, fill=True)
plt.title(f"График плотности (KDE) для {chosen_col}")

plt.tight_layout()
plt.show()
```

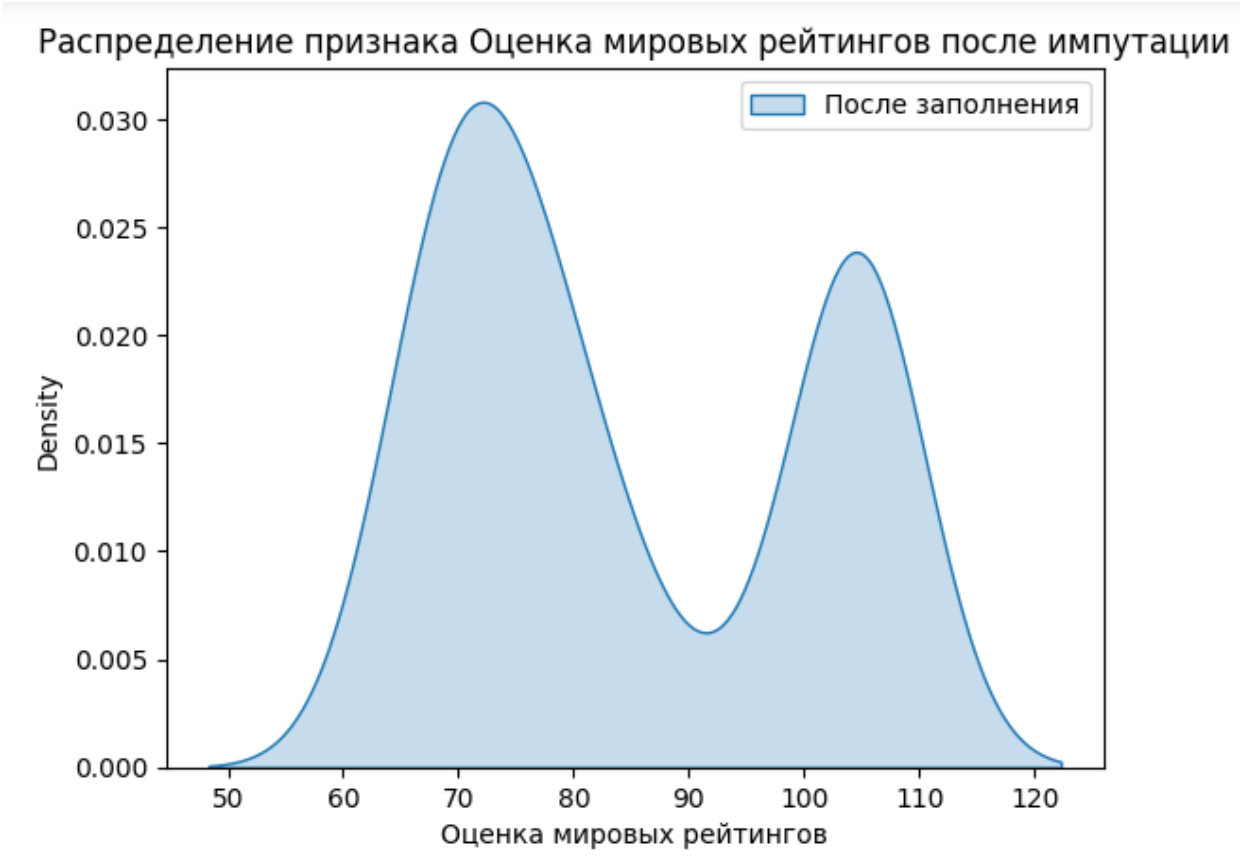


Видим, что все-таки небольшой хвост есть. Поэтому можно использовать метод "хвоста распределения" для заполнения пропусков в столбце «Оценка мировых рейтингов».

```python
# Функция для заполнения пропусков "хвостом распределения"
def impute_with_tail(dataset, column):
    """
    Заполняет пропуски в числовом признаке значением "хвоста распределения"
    """
    # Вычисляем параметры для "хвоста распределения" (метод IQR)
    # Подход для асимметричного распределения:
    Q1 = dataset[column].quantile(0.25)
    Q3 = dataset[column].quantile(0.75)
    IQR = Q3 - Q1
    extreme_value = Q3 + 3 * IQR  # Используем 3*IQR для выделения выбросов

    print(f"Для '{column}': Q1={Q1}, Q3={Q3}, IQR={IQR}, Extreme Value={extreme_value}")

    # Заполняем пропуски этим значением
    imputer = SimpleImputer(strategy='constant', fill_value=extreme_value)
    dataset[column] = imputer.fit_transform(dataset[[column]])

    return dataset
```

```python
# Заполняем пропуски "хвостом распределения"
data = impute_with_tail(data, chosen_col)

# Визуализируем результат
sns.kdeplot(data[chosen_col], fill=True, label="После заполнения")
plt.title(f"Распределение признака {chosen_col} после импутации")
plt.legend()
plt.show()
```

```
Для 'Оценка мировых рейтингов': Q1=68.725, Q3=77.7, IQR=8.975000000000009, Extreme Value=104.62500000
```



Проверим заполнение пропусков. И как видно на рисунке ниже, все пропуски в столбце «Оценка мировых рейтингов» были заполнены.

```
1 data.isnull().sum()
```

```
Название университета                  0
Регион                                 0
Год основания                          0
Девиз                                 14
Национальный ранг                      0
Мировой рейтинг                        0
Оценка мировых рейтингов               0
Minimum_IELTS_score                    0
Иностранные студенты                   0
Оценка студентов                       0
Кол-во поступивших студентов (тыс.)    0
Кол-во преподавательского состава      0
Тип управления университета            0
Местоположение кампуса                17
Стоимость жизни в год                  0
Широта                                 0
Долгота                                0
dtype: int64
```

```
1 data.shape
```

```
(126, 17)
```

Задача № 29

Константные признаки — признаки, которые имеют одно и то же значение во всех строках. Их можно просто удалить, так как они не несут информации.

Псевдоконстантные признаки — признаки с низкой вариативностью (например, если одно значение встречается в 95% случаев, а остальные значения — редкие).

Для удаления константных и псевдоконстантных признаков можно использовать анализ дисперсии признаков. Константные признаки имеют нулевую дисперсию, а псевдоконстантные — дисперсию, близкую к нулю

Удаление константных и псевдоконстантных (почти константных) признаков

- Если признак содержит одинаковые (константные) значения, то он не может внести вклад в построение модели.

- Если признак содержит почти все одинаковые (константные) значения, то скорее всего он мало полезен при построении модели. (При этом нужно быть осторожным, так как данный признак может быть индикатором одного из классов в случае классификации).

Для поиска таких признаков будем использовать дисперсию:

  - У константного признака нулевая дисперсия.
  - У псевдоконстантного списка значение дисперсии очень мало.

```python
from sklearn.feature_selection import VarianceThreshold

# Генерация данных с целыми числами
np.random.seed(42)

# 1. Константный признак (все значения = 1)
constant_feature = np.ones(100, dtype=int)

# 2. Псевдоконстантный признак (95% значений = 5, 5% = 6)
pseudo_constant = np.array([5] * 95 + [6] * 5, dtype=int)
np.random.shuffle(pseudo_constant)  # Перемешиваем

# 3. Нормальные признаки с высокой дисперсией
feature_3 = np.random.randint(0, 5, size=100)  # Случайные целые от 0 до 99
feature_4 = np.random.randint(0, 2, size=100)    # Бинарный признак (0 или 1)
feature_5 = np.random.randint(1, 10, size=100)   # Целевой признак (1-9)

# Создаем DataFrame
df = pd.DataFrame({
    'feature_1': constant_feature,
    'feature_2': pseudo_constant,
    'feature_3': feature_3,
    'feature_4': feature_4,
    'feature_5': feature_5
})

# Вывод первых нескольких строк
df.head()
```

	feature_1	feature_2	feature_3	feature_4	feature_5
0	1	5	2	1	1
1	1	5	0	1	8
2	1	5	2	0	4
3	1	5	2	1	6
4	1	5	0	1	8

Удалим признаки с дисперсией, ниже установленного порога с помощью класса VarianceThreshold из библиотеки Sklearn:

```python
# Применение VarianceThreshold для удаления признаков с низкой дисперсией
selector = VarianceThreshold(threshold=0.15)
selector.fit(df)
```

```
VarianceThreshold(threshold=0.15)
```

```python
# Вывод значений дисперсий для каждого признака
# Значения дисперсий для каждого признака
# Выводим дисперсии в читаемом формате
print("\nДисперсии признаков:")
for col, var in zip(df.columns, selector.variances_):
    # Форматируем вывод: 8 знаков после точки, без экспоненты
    formatted_var = f"{var:.8f}".rstrip('0').rstrip('.') if var != 0 else "0.0"
    print(f"{col}: {formatted_var}")
```

```
Дисперсии признаков:
feature_1: 0.0
feature_2: 0.0475
feature_3: 2.0344
feature_4: 0.2484
feature_5: 6.9139
```

Установили порог threshold=0.15. Удаляются все признаки, где дисперсия **меньше** этого значения:

- $feature\_1$ (0.0) $<$ 0.15 $\rightarrow$ **удален**.
- $feature\_2$ (0.0475) $<$ 0.15 $\rightarrow$ **удален**.
- $\boldsymbol{feature\_3}$ (2.0344) $>$ 0.15 $\rightarrow$ остается.
- $\boldsymbol{feature\_4}$ (0.2484) $>$ 0.15 $\rightarrow$ остается.
- $\boldsymbol{feature\_5}$ (6.9139) $>$ 0.15 $\rightarrow$ остается.

```python
# Преобразование и создание нового DataFrame с оставшимися признаками
df_selected = pd.DataFrame(selector.transform(df), columns=df.columns[selector.get_support()])
```

```python
df_selected.head()
```

	feature_3	feature_4	feature_5
0	2	1	1
1	0	1	8
2	2	0	4
3	2	1	6
4	0	1	8

Дополнительное задание

Для произвольной колонки данных построить гистограмму – выберем колонку «Мировой рейтинг»:

```
# Выбор столбца
column_name = "Мировой рейтинг"

# Построение гистограммы
plt.figure(figsize=(8, 5))
plt.hist(data[column_name].dropna(), bins=15, edgecolor="black", alpha=0.7)
plt.xlabel(column_name)
plt.ylabel("Частота")
plt.title(f"Гистограмма для {column_name}")
plt.grid(True)
plt.show()

```