Лабораторная работ 2.
Предварительная обработка данных.

Цель: научиться пользоваться библиотекой Pandas и ее возможностями для предварительной обработки данных.
Задачи:

1. Загрузить датасет в Pandas DataFrame;
2. Переименовать столбцы;
3. Определить пустые строки и удалить;
4. Определить пустые ячейки и заменить на среднее значение;
5. Определить дубликаты и удалить;
6. Привести столбцы с числами к числовому формату (int).

Пример выполнения лабораторной работы.

Импортируем необходимые библиотеки, загружаем данные и выводим DataFrame.

```
1. import pandas as pd
2. import numpy as np
3. df = pd.read_csv('lab1.csv',encoding='cp1251')
4. print(df)
```

Результат:

```
     номер     пол  возраст  доход  рейтинг трат
0      1.0    Male     19.0   15.0          39.0
1      2.0    Male     21.0   15.0          81.0
2      3.0  Female     20.0   16.0           6.0
3      4.0  Female     23.0   16.0          77.0
4      5.0  Female     31.0   17.0          40.0
..     ...     ...      ...    ...           ...
195  195.0  Female     47.0  120.0          16.0
196  197.0  Female     45.0  126.0          28.0
197  198.0    Male     32.0  126.0          74.0
198  199.0    Male     32.0  137.0          18.0
199  200.0    Male     30.0  137.0          83.0

[200 rows x 5 columns]
```

Посмотрим сводную информацию какие столбцы, какие типы данных, какие они принимают значения.

```
1.df = pd.read_csv('lab1.csv',encoding='cp1251')
2.print(df.info())
```

Из полученной информации мы узнаем, что в таблице 200 строк, 5 столбцов, видим типы данных для каждого столбца и количество ненулевых значений.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Номер                   199 non-null    float64
 1   Пол                     199 non-null    object
 2   Возраст                 197 non-null    float64
 3   Доход $                 197 non-null    float64
 4   Рейтинг трат (1-100)    199 non-null    float64
dtypes: float64(4), object(1)
memory usage: 7.9+ KB
```

Переименуем название столбцов и сохраним, для дальнейшей работы.

```
1.df = pd.read_csv('1.csv',encoding='cp1251')
2.col_name = ['id', 'gender', 'age', 'income','spending_rating' ]
3.df.set_axis(col_name, axis = 'columns', inplace = True)
4.df.to_csv('processed_1.csv',index=False, header=True)
5.print(df.info())
```

В результате получим:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               199 non-null    float64
 1   gender           199 non-null    object
 2   age              197 non-null    float64
 3   income           197 non-null    float64
 4   spending_rating  199 non-null    float64
dtypes: float64(4), object(1)
memory usage: 7.9+ KB
```

Удалим все строки, в которых пропущено значение в столбце «id».

```
1.df = pd.read_csv('1.csv',encoding='cp1251')
2.col_name = ['id', 'gender', 'age', 'income','spending_rating' ]
3.df.set_axis(col_name, axis = 'columns', inplace = True)
4.df.dropna(subset = ['id'], inplace=True)
5.print(df.info())
```

В результате получим:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 199 entries, 0 to 199
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               199 non-null    float64
 1   gender           199 non-null    object
 2   age              197 non-null    float64
 3   income           197 non-null    float64
 4   spending_rating  199 non-null    float64
dtypes: float64(4), object(1)
memory usage: 9.3+ KB
```

Посмотрим уникальные значения в столбце возраста «age».

```
1.df = pd.read_csv('processed_2.csv',encoding='cp1251')
2.print(df.age.unique())
```

Результат:

```
[19. 21. 20. 23. 31. 32. 35. nan 64. 25. 67. 58. 24. 22. 52. 46. 54. 29.
 45. 40. 60. 53. 18. 49. 42. 30. 36. 65. 48. 50. 27. 33. 59. 47. 51. 69.
 70. 63. 43. 68. 26. 57. 38. 55. 34. 66. 39. 44. 28. 37. 56. 41.]
```

Выведем на экран срез данных с пустыми значениями в столбце возраста «age».

```
1.df = pd.read_csv('processed_2.csv',encoding='cp1251')
2.print(df[df['age'].isnull() == True])
```

Результат:

	id	gender	age	income	spending_rating
7	8.0	Female	NaN	18.0	94.0
14	15.0	Male	NaN	20.0	13.0

Определим среднее значение возраста отдельно для мужчин и для женщин.

```
1.df = pd.read_csv('processed_2.csv',encoding='cp1251')
2.male_age = round(df[(df['gender'] == 'Male')]['age'].mean())
3.female_age = round(df[(df['gender'] == 'Female')]['age'].mean())
4.print(male_age,female_age)
```

Результат:

```
40 38
```

А теперь заменим пропущенный значения возраста полученными результатами и сохраним файл.

```
1.df = pd.read_csv('processed_2.csv',encoding='cp1251')
2.male_age = round(df[(df['gender'] == 'Male')]['age'].mean())
3.female_age = round(df[(df['gender'] == 'Female')]['age'].mean())
4.df['age'] = df['age'].fillna(male_age)
5.df['age'] = df['age'].fillna(female_age)
6.df.to_csv('processed_3.csv',index=False, header=True)
```

Тоже самое проделаем для колонки «incame». Выведем на экран срез данных с пустыми значениями.

```
1.df = pd.read_csv('processed_3.csv',encoding='cp1251')
2.print(df[df['income'].isnull() == True])
```

Результат:

	id	gender	age	income	spending_rating
6	7.0	Female	35.0	NaN	6.0
15	16.0	Male	22.0	NaN	79.0

Подсчитаем среднее значение дохода в зависимости от пола.

```
1.df = pd.read_csv('processed_3.csv',encoding='cp1251')
2.male_income = round(df[(df['gender'] == 'Male')]['income'].mean())
3.female_income = round(df[(df['gender'] == 'Female')]['income'].mean())
4.print(male_income,female_income)
```

Результат:

```
63 60
```

Заменим пропущенный значения дохода полученными результатами и сохраним файл.

```
df = pd.read_csv('processed_3.csv',encoding='cp1251')
male_income = round(df[(df['gender'] == 'Male')]['income'].mean())
female_income = round(df[(df['gender'] == 'Female')]['income'].mean())
df['income'] = df['income'].fillna(male_income)
df['income'] = df['income'].fillna(female_income)
df.to_csv('processed_4.csv',index=False, header=True)
```

Теперь необходимо проверить файл на дубликаты и узнать их количество.

```
df = pd.read_csv('processed_4.csv',encoding='cp1251')
print(df[df.duplicated() == True])
print(df.duplicated().sum())
```

Результат:

```
       id  gender   age  income  spending_rating
194  195.0  Female  47.0   120.0             16.0
1
```

Удаляем дубликат, сохраняем файл.

```
df = pd.read_csv('processed_4.csv',encoding='cp1251')
df = df.drop_duplicates().reset_index(drop=True)
df.to_csv('processed_5.csv',index=False, header=True)
```

Проверяем новый файл на дубликат.

```
df = pd.read_csv('processed_5.csv',encoding='cp1251')
print(df.duplicated().sum())
```

Результат:

```
0
```

Посмотрим, как выглядят данные и какой тип данных у каждого столбца.

```
df = pd.read_csv('processed_5.csv',encoding='cp1251')
print(df.head(5))
print(df.dtypes)
```

Результат:

```
    id  gender   age  income  spending_rating
0  1.0    Male  19.0    15.0             39.0
1  2.0    Male  21.0    15.0             81.0
2  3.0  Female  20.0    16.0              6.0
3  4.0  Female  23.0    16.0             77.0
4  5.0  Female  31.0    17.0             40.0
id                float64
gender             object
age               float64
income            float64
spending_rating   float64
dtype: object
```

Приведем столбцы с числами («id», «age», «income» и «spending_rating») к формату int64. Изменим тип данных у всех этих столбцов и сохраним изменения.

```
df = pd.read_csv('processed_5.csv',encoding='cp1251')
df['id'] = df['id'].astype('int64')
df['age'] = df['age'].astype('int64')
df['income'] = df['income'].astype('int64')
df['spending_rating'] = df['spending_rating'].astype('int64')
df.to_csv('processed_6.csv',index=False, header=True)
```

Посмотрим, как теперь выглядят данные и проверим сводную информацию.

```
df = pd.read_csv('processed_5.csv',encoding='cp1251')
print(df.head(5))
print(df.info())
```

Результат:

```
   id  gender  age  income  spending_rating
0   1    Male   19      15               39
1   2    Male   21      15               81
2   3  Female   20      16                6
3   4  Female   23      16               77
4   5  Female   31      17               40
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 198 entries, 0 to 197
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               198 non-null    int64
 1   gender           198 non-null    object
 2   age              198 non-null    int64
 3   income           198 non-null    int64
 4   spending_rating  198 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

В ходе лабораторной работы мы познакомились с Pandas DataFrame и произвели первичную обработку данных.

Задание.

1.Найти датасет содержавший поля с разными типами данных;
2.Загрузить датасет в Pandas DataFrame и посмотреть сводную информацию о датасете;
3. Переименовать одну или все колонки;
4. Определить пустые строки и удалить;
5. Определить пустые ячейки, вывести на экран и заменить средним значением колонки, средним значением с учетом параметра другой колонки.
6. Найти дубликаты, вывести на экран и удалить;

7. Изменить типы данных для одной или всех колонок;
8. Сохранить исправленный датасет для следующей лабораторной работы.

Формат отчета.

Протокол лабораторной работы в формате PDF, который должен содержать, поэтапное выполнение всех задач с текстовым описанием ваших действий и экранными формами, отображающими данные действия.