Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра информационной безопасности

Направление подготовки **09.03.01 Информатика и вычислительная техника**

Направленность(профиль) **Автоматизированные системы обработки информации**

Факультет **Физики, математики, информатики**

Дисциплина **Системы искусственного интеллекта**

Форма обучения **очная**

**ОТЧЕТ**

по лабораторной работе №1

Выполнил: студент 2 курса, 209 группы Красников Я.И.  
Проверил: ассистент кафедры ИБ Юрьев И. А.

Курск 2025

**Лабораторная работа №1**

1.Загрузите датасет в Pandas DataFrame и переименуйте столбцы. (рис.1)

import pandas as pd

df = pd.read\_csv('1.csv')

from IPython.display import display

display(df)

display(df.head(2))

display(df.tail(2))

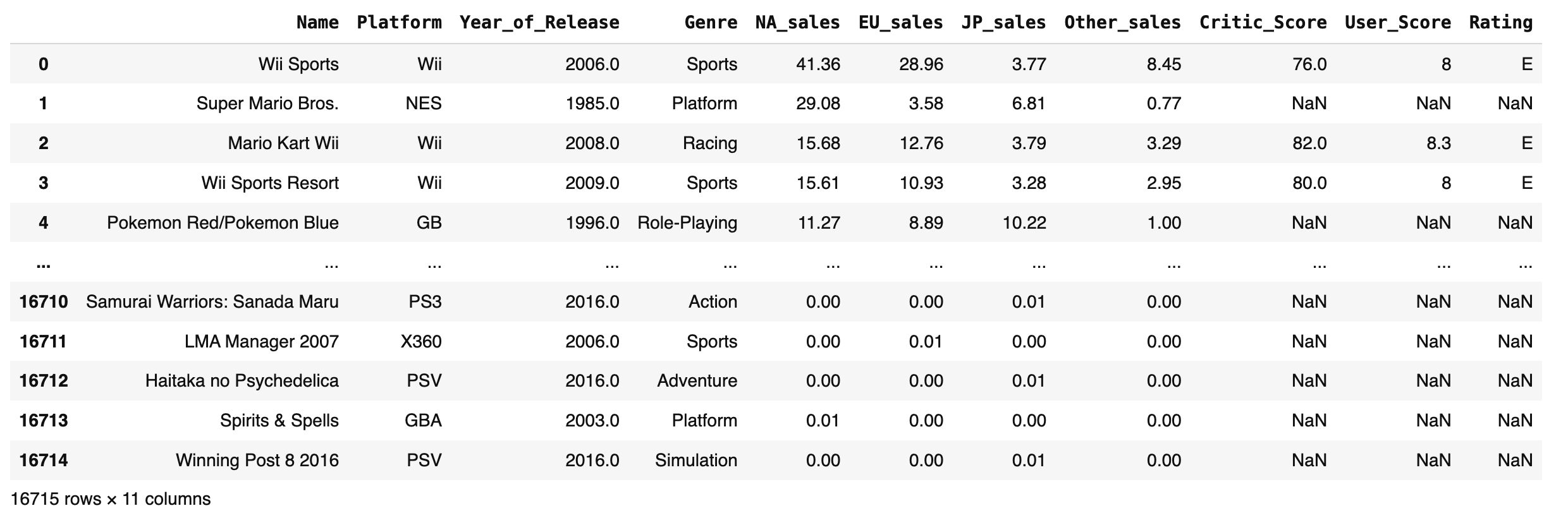


Рисунок 1. Загрузка датасета в Pandas DataFrame и переименование столбцов

2. Категоризируйте данные по признаку с использованием apply() и создайте новый столбец с категориями. (рис.2)

df.columns = ['Names', 'Platforms', 'Years','Genre','NAsales','EAsales','JPsales','Othersales','Criticscore','userscore','Rating']

display(df)

display(df.head(2))

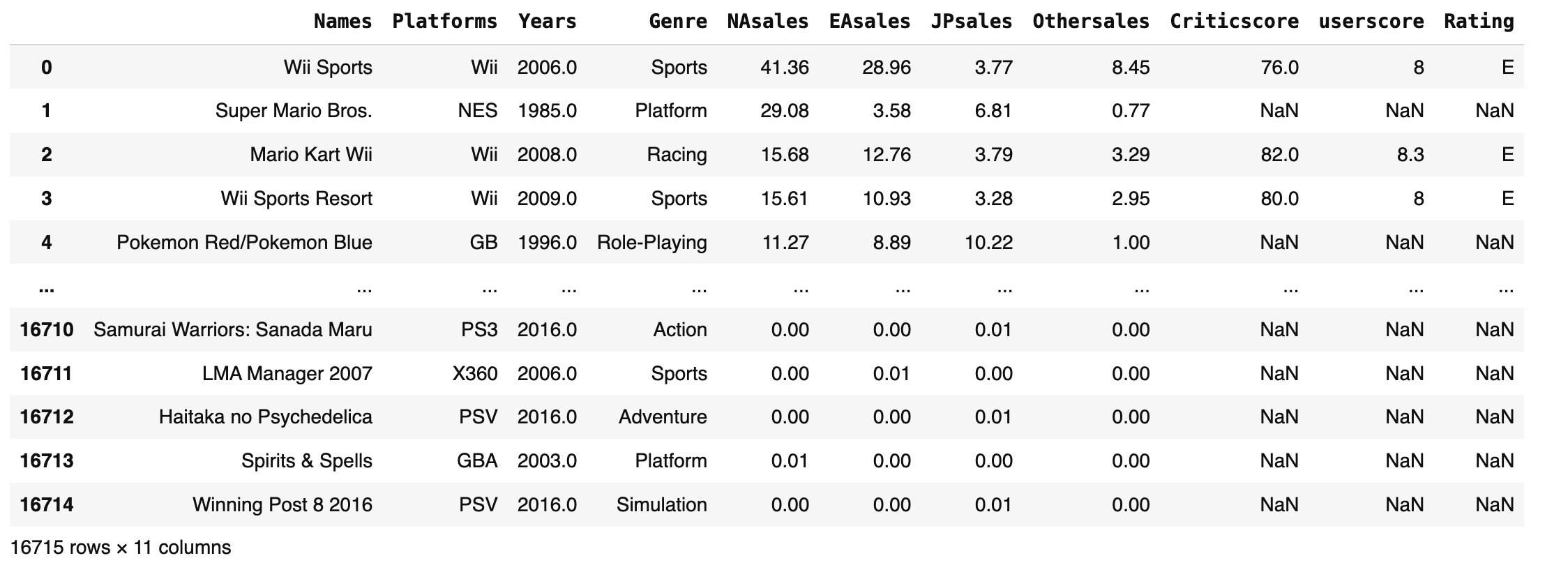


Рисунок 2. Категоризация данных с использованием apply() и создание нового столбца с категориями

3. Создайте срез данных, выбрав столбцы на ваш выбор и строки по условиям. Используйте два различных метода для создания срезов. (рис.3.1-3.3)

def categoriases(value):

if value > 2003:

return 'new games'

else :

return 'old games'

df['category'] = df['Years'].apply(categoriases)

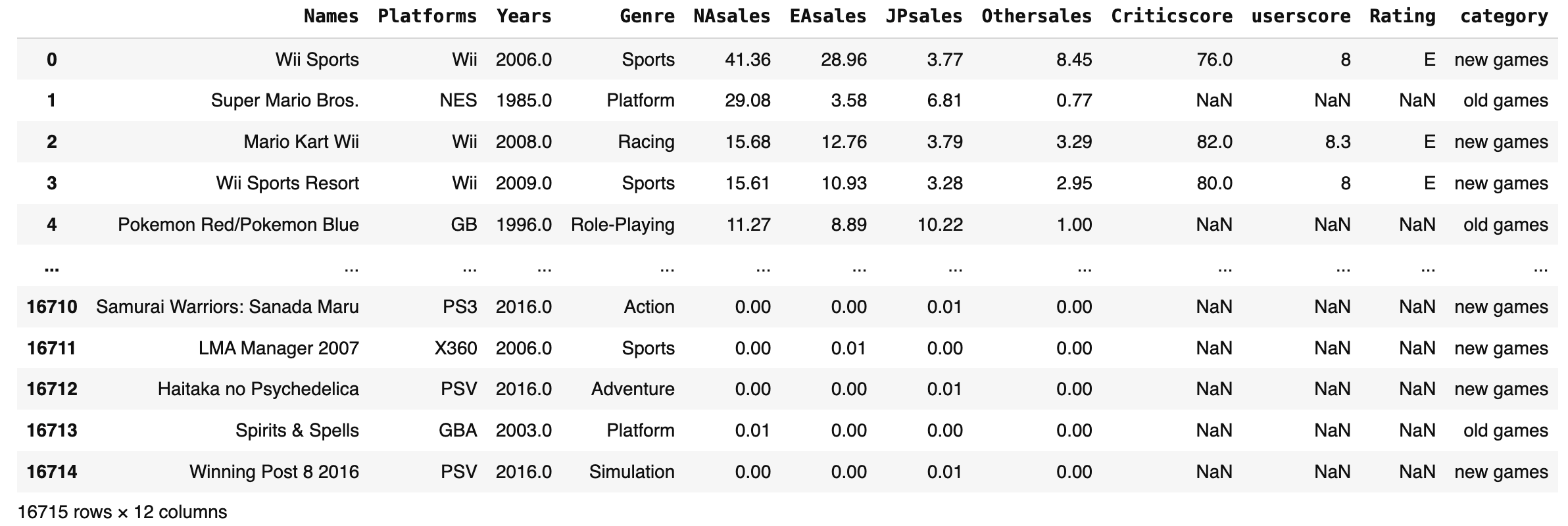
display(df) 

Рисунок 3.1. Создание срезов данных

srez = df.loc[df['category'] == 'old games', [ 'Years']]

print (srez)

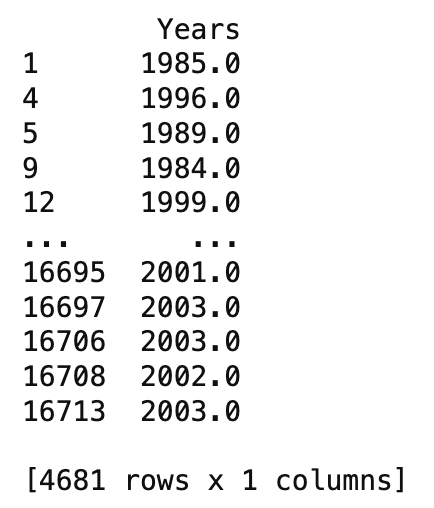


Рисунок 3.2. Первый метод

srez\_2 = df.iloc[0:10, [0, 1, 2]]

print(srez\_2)



Рисунок 3.3. Второй метод

4. Проведите анализ данных: Постройте график зависимости числового признака от любого другого признака.(рис.4)

import matplotlib.pyplot as plt

plt.scatter(df['Years'], df['Platforms'])

plt.xlabel('Years')

plt.ylabel('Platforms')

plt.title('график зависящий от платформ и годов выпуска игр')

plt.show()

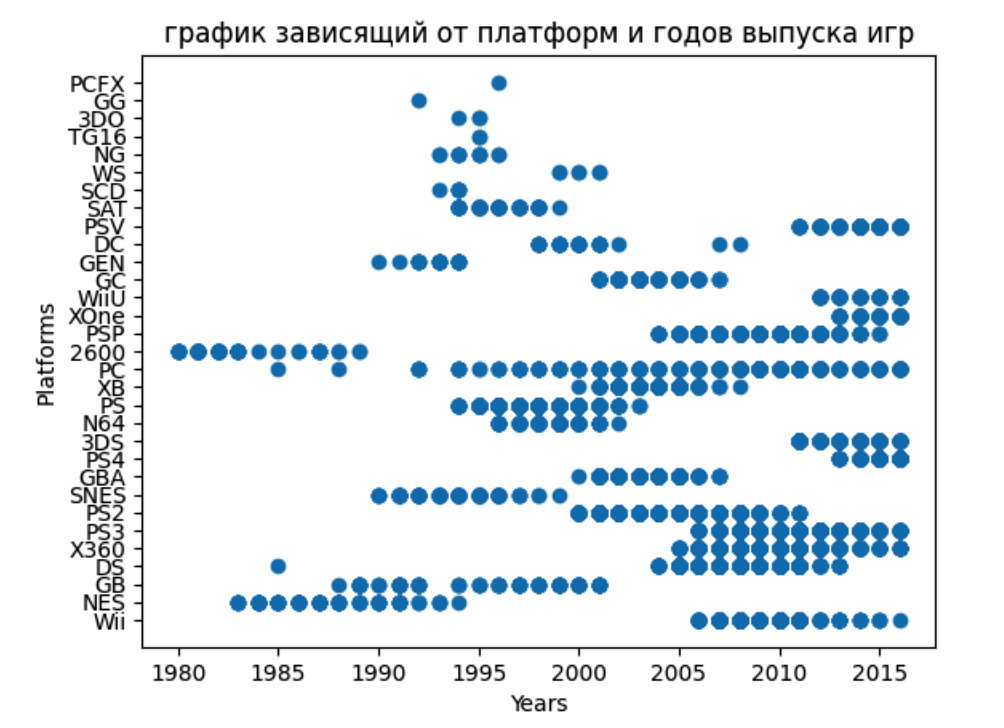


Рисунок 4. Построение графика зависимости числового признака

5. Выполните группировку данных по категориям и произведите агрегацию для числовых признаков (например, среднего, медианы, минимума, максимума)(рис.5).

grouped = df.groupby('category').agg({

'NAsales': ['mean', 'median', 'min', 'max'],

'EAsales': ['mean', 'median', 'min', 'max']

})

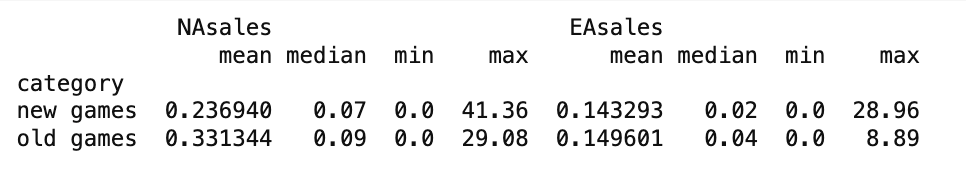
print(grouped) 

Рисунок 5. Группировка данных по категориям и агрегация числовых признаков

6. Работа с датой и временем:

Преобразуйте столбец с датой в формат datetime и произведите анализ данных, учитывая информацию о времени.(рис.6)

df['Years'] = pd.to\_datetime(df['Years'], format='%Y')

df['Year'] = df['Years'].dt.year

yearly\_sales = df.groupby('Year')['NAsales'].sum()

print(yearly\_sales)

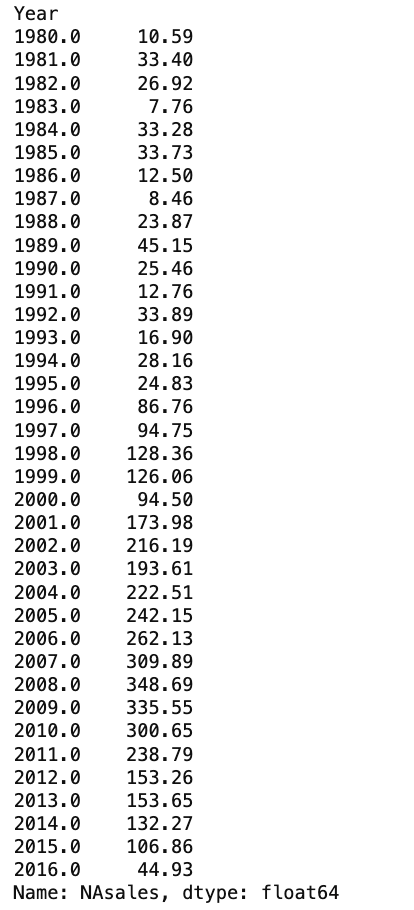


Рисунок 6. Работа с датой и временем

7. Определите количество выбросов.(рис.7)

Q1 = df['NAsales'].quantile(0.25)

Q3 = df['NAsales'].quantile(0.75)

IQR = Q3 - Q1

outliers = df[(df['NAsales'] < (Q1 - 1.5 \* IQR)) | (df['NAsales'] > (Q3 + 1.5 \* IQR))]

print(f"Количество выбросов: {len(outliers)}")

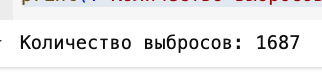


Рисунок 7. Определение количества выбросов

Ссылка на github: