С помощью следующих кодов можно: удалить ненужную информацию из набора данных с помощью функции drop(); установить индекс для набора данных, чтобы можно было легко ссылаться на элементы в нем; очищать поля объекта с помощью метода доступа .str(); очищать весь набор данных с помощью метода applymap(); пропускать строки в файле CSV; переименовывать столбцы с помощью метода rename().

Знать и понимать то, как можно использовать Pandas и NumPy для очистки наборов данных, очень полезно, потому что это большая часть науки о данных.

**import pandas as pd**

**import numpy as np**

**#Удаление столбцов в DataFrame**

**df = pd.read\_csv('Datasets/BL-Flickr-Images-Book.csv')**

**print('Вывод загруженного csv файла:')**

**print(df.head())**

**to\_drop = ['Edition Statement',**

**'Corporate Author',**

**'Corporate Contributors',**

**'Former owner',**

**'Engraver',**

**'Contributors',**

**'Issuance type',**

**'Shelfmarks']**

**df.drop(to\_drop, inplace=True, axis=1)**

**print('Вывод csv файла с удаленными столбцами:')**

**print(df.head())**

**#Изменение индекса фрейма данных**

**df = df.set\_index('Identifier')**

**print(' Замена существующего индекса столбцом Identifier:')**

**print(df.head())**

**print('Получение доступа к каждой записи:')**

**print(df.loc[206])**

**#Очистка полей в данных**

**print('Вывод поля даты публикации для того, чтобы мы могли выполнять вычисления в будущем')**

**print(df.loc[1905:, 'Date of Publication'].head(10))**

**extr = df['Date of Publication'].str.extract(r'^(\d{4})', expand=False)**

**print('Модернизированные поля даты публикации:')**

**print(extr.head())**

**df['Date of Publication'] = pd.to\_numeric(extr)**

**print(f"Получение числовой версии столбца: {df['Date of Publication'].dtype}")**

**#Объединение методов str с NumPy для очистки столбцов**

**print('Вывод содержимого столбца Place of Publication')**

**print(df['Place of Publication'].head(10))**

**print('Вывод информации о двух конкретных записях:')**

**print(df.loc[4157862])**

**print(df.loc[4159587])**

**pub = df['Place of Publication']**

**london = pub.str.contains('London')**

**print('Вывод очищенной колонки:')**

**print(london[:5])**

**oxford = pub.str.contains('Oxford')**

**df['Place of Publication'] = np.where(london, 'London',**

**np.where(oxford, 'Oxford',**

**pub.str.replace('-', ' ')))**

**print('Объединение с помощью np.where')**

**print(df['Place of Publication'].head())**

**#Очистка всего набора данных с помощью функции applymap**

**university\_towns = []**

**with open('Datasets/university\_towns.txt') as file:**

**for line in file:**

**if '[edit]' in line:**

**state = line**

**else:**

**university\_towns.append((state, line))**

**print('Вывод созданного списка, преобразованного в DataFrame:')**

**print(university\_towns[:5])**

**towns\_df = pd.DataFrame(university\_towns,**

**columns=['State', 'RegionName'])**

**print('Вывод результирующего DataFrame:')**

**print(towns\_df.head())**

**#Переименование столбцов и пропуск строк**

**olympics\_df = pd.read\_csv('Datasets/olympics.csv')**

**print('Вывод olympics.csv:')**

**print(olympics\_df.head())**

**olympics\_df = pd.read\_csv('Datasets/olympics.csv', header=1)**

**print('Вывод olympics.csv без 0 строки:')**

**print(olympics\_df.head())**

**new\_names = {'Unnamed: 0': 'Country',**

**'? Summer': 'Summer Olympics',**

**'01 !': 'Gold',**

**'02 !': 'Silver',**

**'03 !': 'Bronze',**

**'? Winter': 'Winter Olympics',**

**'01 !.1': 'Gold.1',**

**'02 !.1': 'Silver.1',**

**'03 !.1': 'Bronze.1',**

**'? Games': '# Games',**

**'01 !.2': 'Gold.2',**

**'02 !.2': 'Silver.2',**

**'03 !.2': 'Bronze.2'}**

**olympics\_df.rename(columns=new\_names, inplace=True)**

**print('Вывод нового olympics.csv: ')**

**print(olympics\_df.head())**