Практическая работа 3. Программные средства консолидации данных с использованием Python

Цель работы: освоить практические навыки консолидации данных из различных источников с использованием Python и библиотеки pandas.

Задачи:

1. Загрузить данные из различных источников (CSV, Excel, JSON).
2. Провести предварительную обработку и очистку данных.
3. Объединить данные из разных источников.
4. Выполнить агрегацию и трансформацию данных.
5. Сохранить консолидированные данные в новый файл.

Необходимое программное обеспечение:

- Python 3.x
- Библиотеки: pandas, numpy, matplotlib.

Исходные данные:

- **sales_2022.csv**: данные о продажах за 2022 год.
- **sales_2023.xlsx**: данные о продажах за 2023 год.
- **products.json**: информация о продуктах.

Процесс создания тестовых данных для файлов sales_2022.csv, sales_2023.xlsx и products.json с использованием генератора данных на Python.

```
import pandas as pd
import numpy as np
import json
from datetime import datetime, timedelta
import random
# Функция для генерации случайной даты
def random_date(start, end):
    return start + timedelta(
        seconds=random.randint(0, int((end - start).total_seconds()))
    )
# Генерация данных о продуктах
product_categories = ['Электроника', 'Одежда', 'Книги', 'Продукты питания',
'Мебель']
products = []

for i in range(100):  # Генерируем 100 продуктов
    product = {
        'product_id': f'P{i:03d}',
        'name': f'Продукт {i}',
```

```
            'category': random.choice(product_categories),
            'price': round(random.uniform(10, 1000), 2)
        }
        products.append(product)
# Сохранение данных о продуктах в JSON
    with open('products.json', 'w', encoding='utf-8') as f:
        json.dump(products, f, ensure_ascii=False, indent=4)

    print("Файл products.json создан")
# Генерация данных о продажах за 2022 год
    sales_2022 = []
    start_date = datetime(2022, 1, 1)
    end_date = datetime(2022, 12, 31)

    for _ in range(10000):  # Генерируем 10000 записей о продажах
        product = random.choice(products)
        sale = {
            'date': random_date(start_date, end_date).strftime('%Y-%m-%d'),
            'product_id': product['product_id'],
            'quantity': random.randint(1, 10),
            'sales': round(product['price'] * random.randint(1, 10), 2)
        }
        sales_2022.append(sale)

    df_2022 = pd.DataFrame(sales_2022)
    df_2022.to_csv('sales_2022.csv', index=False)
    print("Файл sales_2022.csv создан")
# Генерация данных о продажах за 2023 год
    sales_2023 = []
    start_date = datetime(2023, 1, 1)
    end_date = datetime(2023, 9, 30)  # Предположим, что данные есть только до
    сентября 2023

    for _ in range(12000):  # Генерируем 12000 записей о продажах (больше, чем
    в 2022)
        product = random.choice(products)
        sale = {
            'date': random_date(start_date, end_date).strftime('%Y-%m-%d'),
            'product_id': product['product_id'],
```

```
            'quantity': random.randint(1, 15),  # Увеличим максимальное количество
            'sales': round(product['price'] * random.randint(1, 15), 2)
        }
        sales_2023.append(sale)

    df_2023 = pd.DataFrame(sales_2023)
    df_2023.to_excel('sales_2023.xlsx', index=False)
    print("Файл sales_2023.xlsx создан")

    print("Все файлы с тестовыми данными успешно созданы.")
```

1. Создание products.json:

- Мы определяем список категорий продуктов.
- Генерируем 100 уникальных продуктов с полями: product_id, name, category и price.
- Каждому продукту присваивается случайная категория и цена.
- Данные сохраняются в формате JSON с отступами для удобства чтения.

2. Создание sales_2022.csv:

- Определяем период продаж с 1 января по 31 декабря 2022 года.
- Генерируем 10000 записей о продажах.
- Для каждой продажи случайным образом выбирается продукт из списка products.
- Генерируется случайная дата продажи в пределах 2022 года.
- Количество проданных единиц выбирается случайно от 1 до 10.
- Сумма продажи рассчитывается как произведение цены продукта на количество.
- Данные сохраняются в CSV формате.

3. Создание sales_2023.xlsx:

- Процесс аналогичен созданию файла за 2022 год, но с некоторыми отличиями:
- Период продаж ограничен с 1 января по 30 сентября 2023 года (предполагая, что данные неполные).
- Генерируется большее количество записей (12000) для имитации роста продаж.
- Максимальное количество проданных единиц увеличено до 15 для создания вариативности.
- Данные сохраняются в формате Excel (xlsx).

Особенности сгенерированных данных:

- Все файлы связаны между собой через поле product_id.
- Данные имеют реалистичную структуру и вариативность.

- В данных присутствуют различные типы полей: строковые (категории, ID), числовые (цены, количество), даты.

Ход работы

#Шаг 1: Подготовка рабочей среды

1. Создайте новый Python-скрипт **data_consolidation.py**.
2. Импортируйте необходимые библиотеки:

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

#Шаг 2: Загрузка данных

1. Загрузите данные из CSV-файла:

```
df_2022 = pd.read_csv('sales_2022.csv')
print("Данные за 2022 год:")
print(df_2022.head())
```

2. Загрузите данные из Excel-файла:

```
df_2023 = pd.read_excel('sales_2023.xlsx')
print("\nДанные за 2023 год:")
print(df_2023.head())
```

3. Загрузите данные из JSON-файла:

```
df_products = pd.read_json('products.json')
print("\nДанные о продуктах:")
print(df_products.head())
```

#Шаг 3: Предварительная обработка и очистка данных

1. Проверьте наличие пропущенных значений:

```
print("\nПропущенные значения:")
print(df_2022.isnull().sum())
print(df_2023.isnull().sum())
print(df_products.isnull().sum())
```

2. Обработайте пропущенные значения (пример для df_2022):

```
df_2022['sales'] = df_2022['sales'].fillna(df_2022['sales'].mean())
```

3. Приведите названия столбцов к единому формату:

```
df_2022.columns = df_2022.columns.str.lower()
df_2023.columns = df_2023.columns.str.lower()
df_products.columns = df_products.columns.str.lower()
```

#Шаг 4: Объединение данных

1. Объедините данные о продажах за 2022 и 2023 годы:

```
df_sales = pd.concat([df_2022, df_2023], ignore_index=True)
print("\nОбъединенные данные о продажах:")
print(df_sales.head())
```

2. Добавьте информацию о продуктах к данным о продажах:

```
df_consolidated = pd.merge(df_sales, df_products, on='product_id', how='left')
print("\nКонсолидированные данные:")
print(df_consolidated.head())
```

#Шаг 5: Агрегация и трансформация данных

1. Рассчитайте общую сумму продаж по категориям продуктов:

```
sales_by_category =
df_consolidated.groupby('category')['sales'].sum().sort_values(ascending=False)
print("\nОбщая сумма продаж по категориям:")
print(sales_by_category)
```

2. Создайте новый столбец с годом продажи:

```
df_consolidated['year'] = pd.to_datetime(df_consolidated['date']).dt.year
```

3. Рассчитайте среднюю сумму продаж по годам:

```
avg_sales_by_year = df_consolidated.groupby('year')['sales'].mean()
print("\nСредняя сумма продаж по годам:")
print(avg_sales_by_year)
```

#Шаг 6: Визуализация данных

1. Создайте график продаж по категориям:

```
plt.figure(figsize=(12, 6))
sales_by_category.plot(kind='bar')
plt.title('Общая сумма продаж по категориям')
plt.xlabel('Категория')
plt.ylabel('Сумма продаж')
plt.tight_layout()
plt.savefig('sales_by_category.png')
```

2. Создайте график средних продаж по годам:

```
plt.figure(figsize=(10, 5))
avg_sales_by_year.plot(kind='line', marker='o')
plt.title('Средняя сумма продаж по годам')
plt.xlabel('Год')
plt.ylabel('Средняя сумма продаж')
plt.tight_layout()
plt.savefig('avg_sales_by_year.png')
```

#Шаг 7: Сохранение консолидированных данных

Сохраните консолидированные данные в CSV-файл:

```
df_consolidated.to_csv('consolidated_sales_data.csv', index=False)
print("\nКонсолидированные данные сохранены в файл
'consolidated_sales_data.csv'")
```

Для всех вариантов студенты должны применить навыки работы с pandas для:

- Чтения данных из разных форматов (`read_csv()`, `read_excel()`, `read_json()`).
- Объединения таблиц (`merge()`, `concat()`).
- Выполнения необходимых аналитических расчётов.
- Представить результаты в виде таблиц или графиков.

Варианты заданий

Вариант 1.

1. Файл CSV: данные о сотрудниках (имя, должность, зарплата).
2. Файл Excel: данные о проектах (название проекта, бюджет, менеджер проекта).
3. Файл JSON: данные о зарплатах по должностям.

Задача: объединить данные и рассчитать среднюю зарплату по каждому проекту, исходя из данных о сотрудниках и их участии в проектах.

Вариант 2.

1. Файл CSV: данные о студентах (имя, курс, средний балл).
2. Файл Excel: данные об успеваемости по предметам (курс, предмет, оценка).
3. Файл JSON: данные о преподавателях и их курируемых курсах.

Задача: объединить данные и посчитать средний балл студентов для каждого преподавателя.

Вариант 3.

1. Файл CSV: данные о клиентах компании (имя, дата рождения, город).
2. Файл Excel: данные о заказах (номер заказа, сумма, дата).
3. Файл JSON: данные о скидках по городам.

Задача: объединить данные и рассчитать общую сумму заказов по каждому городу с учётом скидок.

Вариант 4.

1. Файл CSV: список книг (название, автор, год издания).
2. Файл Excel: данные о читателях (имя, прочитанные книги, дата возврата).
3. Файл JSON: информация об авторах (имя автора, национальность, количество книг).

Задача: объединить данные и рассчитать количество прочитанных книг по авторам.

Вариант 5.

1. Файл CSV: данные о филиалах компании (город, количество сотрудников, выручка).
2. Файл Excel: данные о продажах (филиал, дата продажи, сумма).
3. Файл JSON: данные о квартальных планах продаж по филиалам.

Задача: объединить данные и сравнить фактическую выручку филиалов с плановой.

Вариант 6.

1. Файл CSV: список фильмов (название, режиссёр, год выхода).
2. Файл Excel: данные о просмотрах фильмов (фильм, количество просмотров, страна).
3. Файл JSON: данные о режиссёрах (имя режиссёра, страна).

Задача: объединить данные и рассчитать, какие фильмы какого режиссёра имеют наибольшее количество просмотров по странам.

Вариант 7.

1. Файл CSV: данные о сотрудниках отдела продаж (имя, город, продажи).
2. Файл Excel: данные о ежемесячных планах (город, месяц, план по продажам).
3. Файл JSON: данные о премиях по результатам продаж в различных городах.

Задача: объединить данные и рассчитать премии сотрудников по итогам их продаж за год.

Вариант 8:

1. Файл CSV: данные о спортсменах (имя, вид спорта, результаты).
2. Файл Excel: данные о соревнованиях (соревнование, дата, призы).
3. Файл JSON: данные о тренерах (тренер, вид спорта, спортсмены).

Задача: объединить данные и рассчитать, какие тренеры подготовили больше всего победителей соревнований.

Вариант 9.

1. Файл CSV: список товаров (артикул, категория, цена).
2. Файл Excel: данные о продажах (артикул товара, количество проданных единиц, дата).
3. Файл JSON: данные о скидках на определённые категории товаров.

Задача: объединить данные и рассчитать выручку по каждой категории товаров с учётом скидок.

Вариант 10.

1. Файл CSV: данные о сотрудниках компании (имя, отдел, оклад).
2. Файл Excel: данные о командировках (сотрудник, город, затраты).
3. Файл JSON: данные о затратах по городам (город, суточные, гостиничные расходы).

Задача: объединить данные и рассчитать, насколько покрываются командировочные расходы сотрудников.

Вариант 11.

1. Файл CSV: список студентов (имя, факультет, год поступления).
2. Файл Excel: данные о стипендиях (факультет, год, размер стипендии).
3. Файл JSON: данные о мероприятиях и призах (факультет, мероприятия, премии).

Задача: объединить данные и рассчитать, сколько студенты каждого факультета получают стипендий и призов.

Вариант 12.

1. Файл CSV: список клиентов банка (имя, возраст, кредитный рейтинг).
2. Файл Excel: данные о кредитах (номер кредита, сумма, срок).
3. Файл JSON: данные о процентных ставках для различных кредитных рейтингов.

Задача: объединить данные и рассчитать итоговую выплату по кредитам для каждого клиента с учётом их кредитного рейтинга.

Вариант 13.

1. Файл CSV: данные о производителях автомобилей (бренд, страна).
2. Файл Excel: данные о моделях автомобилей (модель, бренд, цена, год выпуска).
3. Файл JSON: данные о рейтингах автомобилей по странам.

Задача: объединить данные и рассчитать рейтинг каждого бренда в зависимости от страны и моделей автомобилей.

Вариант 14.

1. Файл CSV: данные о пациентах (имя, возраст, диагноз).
2. Файл Excel: данные о лекарствах (лекарство, диагноз, стоимость).
3. Файл JSON: данные о лечении (диагноз, продолжительность лечения).

Задача: объединить данные и рассчитать затраты на лечение для каждого пациента.

Вариант 15.

1. Файл CSV: список преподавателей (имя, кафедра, ставка).
2. Файл Excel: данные о курсах (курс, преподаватель, часы).
3. Файл JSON: данные о ставках за преподавание на разных кафедрах.

Задача: объединить данные и рассчитать зарплату каждого преподавателя с учётом его нагрузки.

Вариант 16.

1. Файл CSV: данные о партнёрах компании (имя партнёра, сумма контракта).
2. Файл Excel: данные о проектах (партнёр, проект, бюджет).
3. Файл JSON: данные о выполнении проектов (партнёр, проект, выполненный объём работы).

Задача: объединить данные и оценить степень выполнения работ по каждому партнёру.

Вариант 17.

1. Файл CSV: список товаров на складе (артикул, категория, количество).
2. Файл Excel: данные о продажах (артикул товара, дата, количество).
3. Файл JSON: данные о спросе на товары по категориям.

Задача: объединить данные и рассчитать прогнозный остаток товаров на складе.

Вариант 18.

1. Файл CSV: данные о сотрудниках отдела маркетинга (имя, уровень квалификации, зарплата).
2. Файл Excel: данные о маркетинговых кампаниях (сотрудник, проект, результат).
3. Файл JSON: данные о бонусах за успешные проекты (уровень квалификации, бонус).

Задача: объединить данные и рассчитать итоговую зарплату каждого сотрудника с учётом бонусов.

Вариант 19.

1. Файл CSV: список акций компаний (название компании, стоимость акции).
2. Файл Excel: данные о сделках на бирже (компания, дата сделки, количество акций).
3. Файл JSON: данные о прогнозах изменения стоимости акций.

Задача: объединить данные и рассчитать ожидаемый доход от купленных акций по каждому портфелю.

Вариант 20.

1. Файл CSV: данные о туристах (имя, страна, возраст).
2. Файл Excel: данные о бронированиях отелей (турист, отель, стоимость).
3. Файл JSON: данные о туристических пакетах (страна, пакет, скидки).

Задача: объединить данные и рассчитать итоговую стоимость путешествия для каждого туриста с учётом скидок на пакеты.

Вариант 21.

1. Файл CSV: данные о фермерских хозяйствах (название, площадь, вид культур).
2. Файл Excel: данные о сборе урожая (фермер, культура, урожай в тоннах).
3. Файл JSON: данные о рыночных ценах на различные культуры.

Задача: объединить данные и рассчитать общий доход фермеров от продажи урожая.

Вариант 22.

1. Файл CSV: данные о покупателях интернет-магазина (имя, город, покупки).
2. Файл Excel: данные о заказах (номер заказа, товар, сумма).
3. Файл JSON: данные о доставке и её стоимости по городам.

Задача: объединить данные и рассчитать общую стоимость заказов для каждого клиента с учётом доставки.

Вариант 23.

1. Файл CSV: данные о сотрудниках и их квалификациях (имя, квалификация, стаж).
2. Файл Excel: данные о проектах (проект, сотрудник, часы работы).
3. Файл JSON: данные о стоимости часа работы в зависимости от квалификации сотрудника.

Задача: объединить данные и рассчитать общую стоимость работы по каждому проекту с учётом квалификации сотрудников.

Вариант 24.

1. Файл CSV: данные о поставщиках продуктов (название, страна, продукты).
2. Файл Excel: данные о заказах продуктов (поставщик, продукт, количество).
3. Файл JSON: данные о ценах на продукты по странам.

Задача: объединить данные и рассчитать затраты на поставки по каждому поставщику.

Вариант 25.

1. Файл CSV: данные о службах такси (название, количество машин).
2. Файл Excel: данные о поездках (служба, количество поездок, средняя цена поездки).
3. Файл JSON: данные о топливных расходах служб такси.

Задача: объединить данные и рассчитать доход и расходы служб такси, а также их рентабельность.