

### Укрупнённый алгоритм процесса взаимодействия дата-сайентиста с заказчиком

- получены от заказчика ссылки на источники данных;
- проведен анализ структуры данных;
- формализована структура данных;
- определён список инструментов для работы;
- создан код в *Python* для загрузки данных и создания *DataFrame*;
- данные обработаны с использованием *Python/pandas*;
- получен результат обработки;
- результат преобразован в формат отчета, определённый заказчиком;
- заказчику передан отчёт.

Данные для анализа поступают в разных форматах.  
Большие по объёму наборы данных упакованы в архив.  
Загрузка данных возможна по URL-ссылке.

### Основные проблемы при выгрузке данных из файлов в *DataFrame*:

- Первая строка данных интерпретируется как строка заголовков
- Данные считываются в неправильной кодировке
- Файл данных упакован в архив

### Функции чтения данных из файлов разных форматов в *DataFrame*

Функция	Значение
<code>pd.read_csv(path)</code>	Чтение файла .csv. В качестве разделителя по умолчанию используется запятая (символ).
<code>pd.read_table(path)</code>	Чтение файла .txt. В качестве разделителя по умолчанию используется символ табуляции (' <code>\t</code> ').
<code>pd.read_excel(path)</code>	Чтение файла .xlsx и .xls

```
pd.read_json(path)
```

Чтение файла .json

## Основные параметры функций `read_csv()` и `read_table()`

Параметр	Описание
<code>path</code>	Первый параметр, адрес файла, можно передавать адрес в интернете
<code>header = None</code>	Не использовать информацию из первой строки файла в качестве имён столбцов
<code>index_col = ИндексСтолбца</code>	Использовать в качестве индекса строк данные в указанном столбце, например <code>index_col = 0</code>
<code>name = [НаименованияСтолбцов]</code>	Задать имена столбцов, например <code>name = ["country", "population"]</code>
<code>sep = ";"</code>	Использовать в качестве разделителя символ ";"
<code>encoding = "ТипКодировки"</code>	Указать тип кодировки считываемого файла, например <code>encoding = "koi8-r"</code>

## Основные параметры функции `read_excel()`

Параметр	Значение
<code>path</code>	Первый параметр, адрес файла, можно передавать адрес в интернете
<code>sheet_name</code>	Ссылка на лист/листы в Excel-файле, например <code>sheetname = [0, 1, "Sheet5"]</code>
<code>na_values</code>	Список значений, которые будут считаться пропусками

## Функции выгрузки данных из DataFrame в файлы разных форматов

Функция	Значение
<code>pd.to_csv(path)</code>	Выгрузка файла .csv
<code>pd.to_excel(path)</code>	Выгрузка файла .xlsx

## Чтение данных из архива

Механизм, используемый в функции `read_csv()`, позволяет проводить чтение текстового файла из архива, не распаковывая его.

**Ограничение** — в **zip**-архиве должен быть один файл.

```
data = pd.read_csv('students_performance.zip')
```

## Запись данных в архив

Механизм, используемый в функции `to_csv()`, позволяет проводить упаковку CSV-файлов, например в **zip**-архив.

```
compression_opts = dict(method='zip', archive_name='out.csv') #  
Определяем параметры архивирования — метод сжатия, имя файл в архиве  
  
data.to_csv('out.zip', index=False, compression=compression_opts) #  
Записываем данные в архив
```

## Считывание данных по интернет-ссылке

При чтении данных из интернета достаточно выбрать необходимую функцию и вместо пути до файла указать ссылку на файл.

```
data =  
pd.read_excel('https://github.com/asaydn/test/raw/master/january.xlsx'  
)
```

## Применение конструкции *with ... as ...* для чтения/записи файла

Конструкция `with ... as ...` гарантирует закрытие файла `filename`, связанного с объектом `f`.

```
with open('path/filename') as f: # Открываем файл и связываем его с
    # Работа с файлом...
    # ...не забываем про отступ...
    # ...
# Нет отступа = работа с файлом закончена, файл filename закрыт
```

## Используемые модули

Библиотека	Описание
<code>import json</code>	Модуль для работы с JSON-файлами
<code>from pprint import pprint</code>	Модуль для вывода информации в структурированном виде
<code>import xml.etree.ElementTree as ET</code>	Модуль для работы со структурами XML