Министерство науки и высшего образования Российской Федерации

Российский химико-технологический университет имени Д. И. Менделеева

Факультет цифровых технологий и химического инжиниринга

Кафедра информационных компьютерных технологий

**ОТЧЕТ**

ПО ЗАЧЕТНОЙ РАБОТЕ

**«Информационные технологии»**

**Библиотека Pandas**

**ВЫПОЛНИЛ:** cтудент группы КС-20 Шишко Р.Г.

**ПРОВЕРИЛ:** ст. преподаватель Васецкий А. М.

**Москва**

**2020**

Библиотека Pandas

Цель данной зачетной работы — изучить библиотеку и на практических примерах программного кода проиллюстрировать её использование. Использоваться будет библиотека Pandas. Pandas - библиотека, ориентированная на обработке и анализе данных, поэтому для иллюстрации понадобятся исходные данные для анализа.

Мы будем использовать три датасета для разных целей:

* Данные о пассажирах Титаника, обработав которые через Pandas, в связке с библиотекой sklearn, мы сможем попробовать предсказать вероятность выживания для заданного пассажира, используя машинное обучение
* Данные об исследуемой экосистеме в южной Аризоне, которые мы обработаем и визуализируем в связке с библиотекой matplotlib/seaborn
* Данные о книгах, находящихся в Британской Библиотеке, в которых нужно будет откорректировать некачественные/неверные данные

Краткий список команд, использованных в данной работе:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| read\_csv | head | shape | info | groupby | sum | sort\_values | tail |
| mean | describe | index | loc | value\_counts | sample | to\_datetime | dropna |
| fillna | unique | nunique | crosstab | to\_numeric | isnull | drop | median |

Для импорта данных будет использоваться функция **read\_csv()**, которая принимает путь к файлу, а также может принимать параметры: - *sep* — отвечает за разделитель, используемый в документе - *index\_col* — отвечает за номер колонки, которая будет использоваться для индексации данных - *skip\_blank\_lines* — отвечает за пропуск пустых строк, принимает значение True/False - *encoding* — отвечает за кодировку файла, принимает её имя …и прочие;

*read\_csv ( ), head ( )*

*Сама функция* ***read\_csv()*** *возвращает объект типа dataframe, который является табличной структурой данных (каждая строка/столбец является объектом типа Series), a метод* ***.head()*** *отвечает за вывод шапки исходных данных*

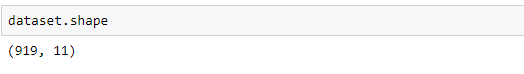
Возьмем первый массив данных

*Этот датасет содержит данные о всех пассажирах Титаника, находящихся на борту во время катастрофы*

|  |
| --- |
|  |
|  |

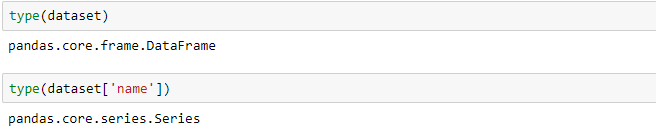
*.shape*

*Возвращает размерность массива данных:*



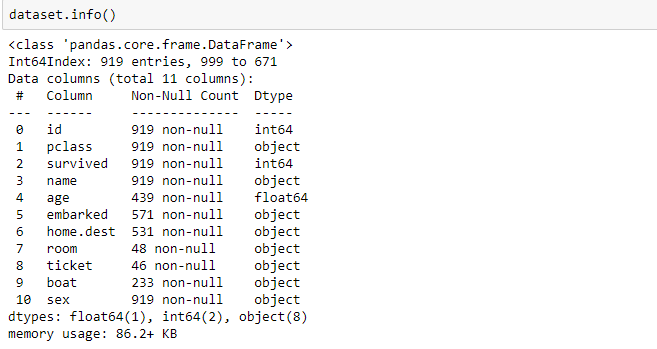
*Датасет является таблицей, состоящей из 919 строк и 11 столбцов*

Мы также можем получить более подробную информацию о типах данных, как всего массива данных, так и конкретных столбцов:



*.info( )*

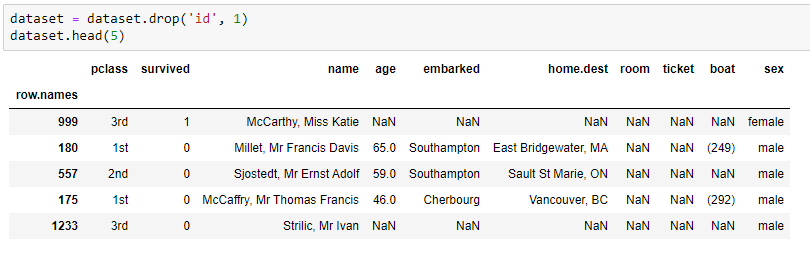
*Возвращает данные по столбцам:*



*.drop( )*

*Удаляет столбец массива:*

***Удалим столбец “id”, дабы оставить только индексы, которые указывают на номер каюты:***



*.groupby( )*

*Группирует данные в таблице по значению индекса:*

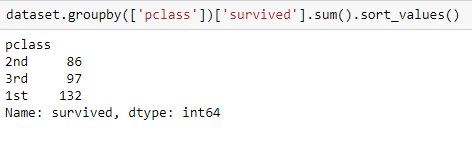
*.sum( )*

*Суммирует данные в таблице по значению индекса:*

*.sort\_values( )*

*Сортирует данные:*

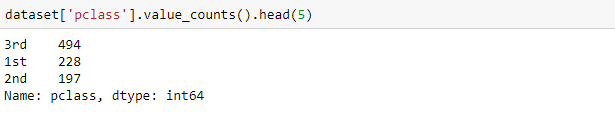
***Посмотрим, какое количество пассажиров выжило в каждом классе:***



*.value\_counts( )*

*Подсчитывает какое количество раз встречалось то или иное значение:*

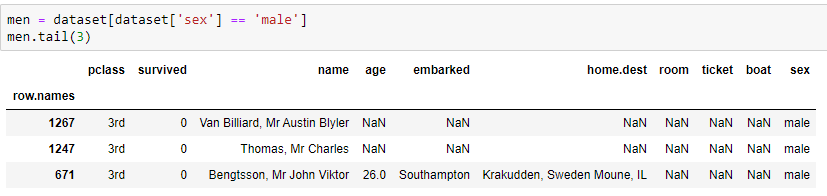
***Выведем общее количество пассажиров по классам***



*.tail( )*

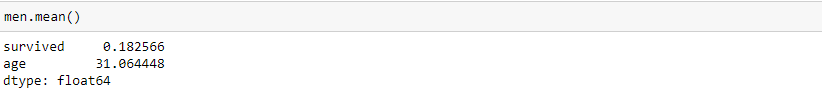
*Возвращает последние элементы массива данных*

***Выделим в отдельный датасет всех мужчин:***



*.mean( )*

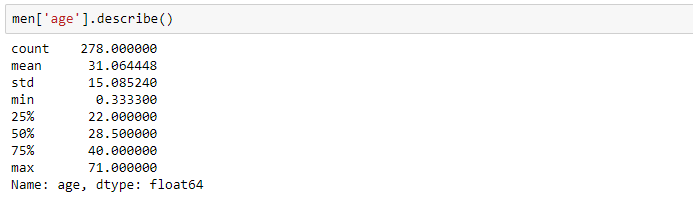
*Возвращает средние значения:*



***Заметим, что выжило всего 18% мужчин, а средний возраст мужчин был 31***

*.describe( )*

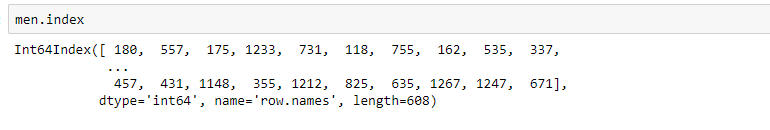
*Возвращает численную статистику массива данных*



***Максимальный возраст был 71 год, а минимальный был 4 месяца***

*.index*

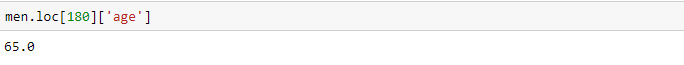
*Возвращает значения, по которым индексирован наш массив данных.*



***Так как наши данные проиндексированы по номеру каюты, можем вывести данные, по этому номеру:***

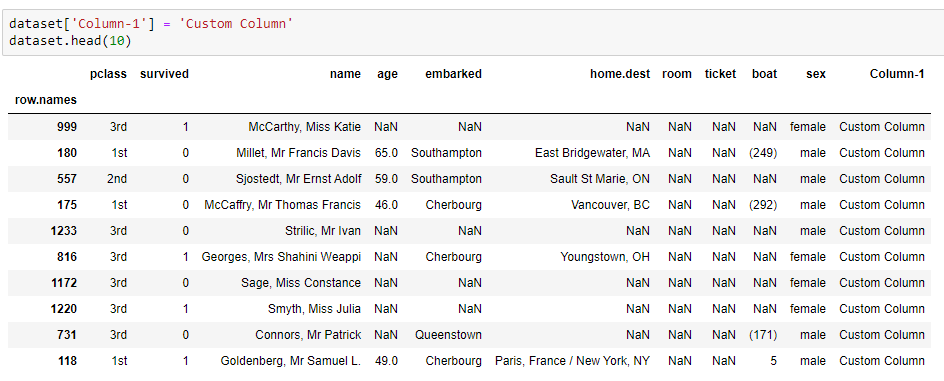
*.loc[ ]*

*Ищет элементы массива данных по индексу*

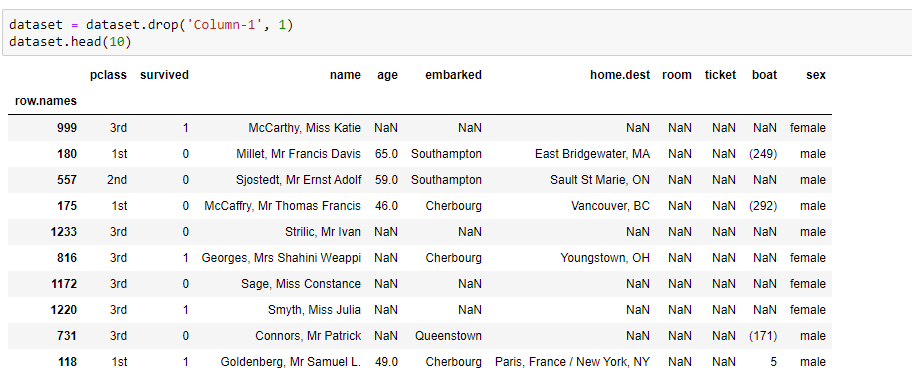


*Как мы видим, пассажиру из 180-ой каюты было 65 лет*

***Также Pandas позволяет как добавлять:***



***…так и удалять колонки:***



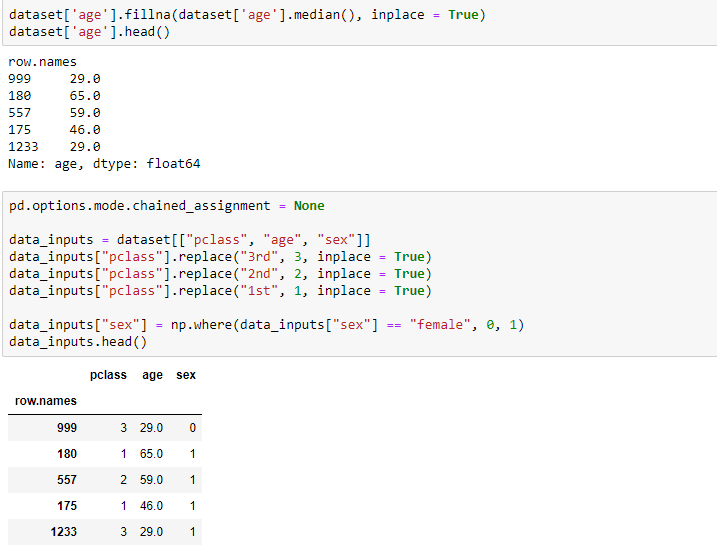
Производим обработку датасета:

*По своей сути, данные делятся на бинарные, качественные и количественные: например, в нашем случае, качественными данными являются: номер каюты, пол пассажира и его имя, а количественным будет его возраст. Для обучения модели нам не понадобится имя пассажира, так как оно никак не влияет на шансы на выживание. Мы будем использовать колонки “класс”, “пол” и “возраст”, однако:*

1. У нас не хватает данных по возрасту, поэтому для пассажиров, у которых не указан возраст, укажем его как медианный для этой выборки
2. Столбец “пол” не является бинарным
3. Столбец “класс” является строчным

*.median( )*

*Возвращает медианное значение.*



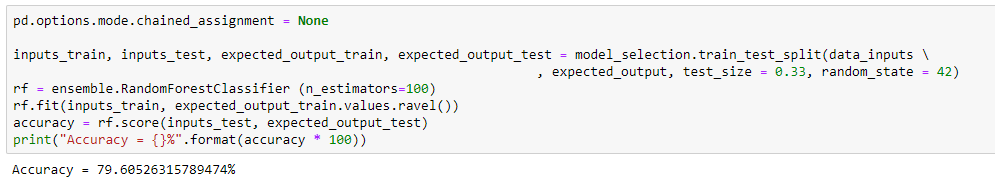
*Теперь мы можем приступать к работе с моделью:*

Использование с sklearn:

***Обозначим как ожидаемый исход параметр, выжил ли пассажир:***



***Опишем и обучим модель:***



Точность модели – около 78 процентов

***Запишем модель в файл:***



***Опишем четырех случайных пассажиров:***

1. 56-летняя женщина из 1-ого класса
2. 31-летняя женщина из 2-ого класса
3. 29-летний мужчина из 2-ого класса
4. 18-летний мужчина из 3-ого класса

***И попробуем предсказать их шансы на выживание:***

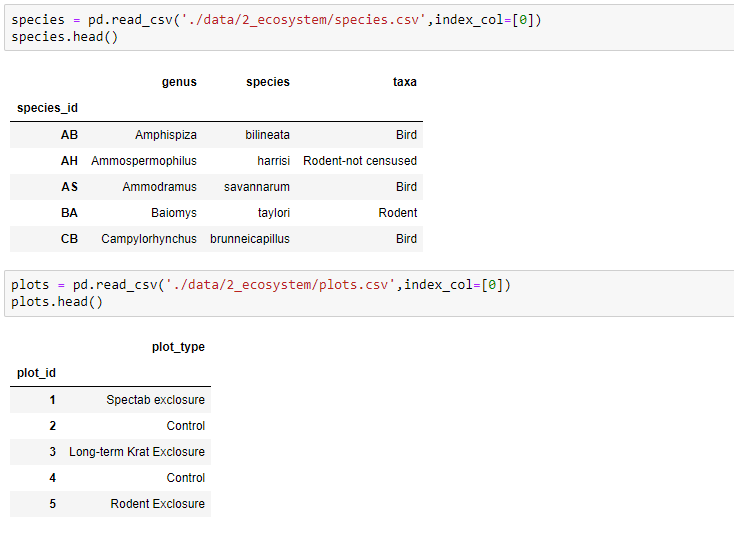


*Исходя из модели, выживут только первые два пассажира.*

Теперь возьмем другой массив данных:

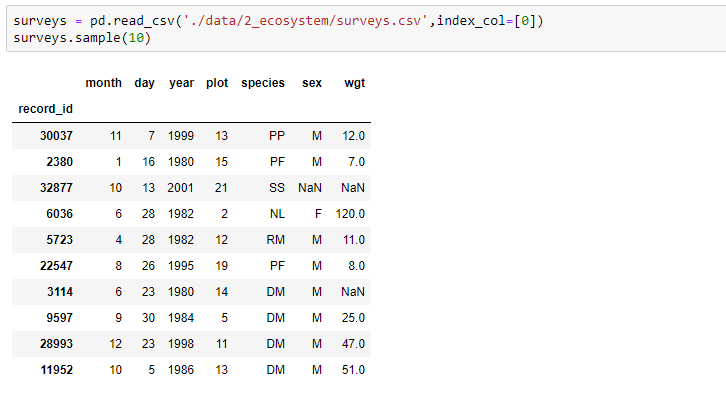
*Этот датасет содержит данные о небольшой экосистеме в южной Аризоне за последние 35 лет. Он является частью большего проекта по изучению влияния различных видов грызунов и насекомых на жизнь растений*

* plots.csv : список участков исследования с их ID и кратким описанием
* species.csv : список с двухсимвольным кодом вида и информации о нем
* surveys.csv : полный список наблюдений над видами на участках



*.sample( )*

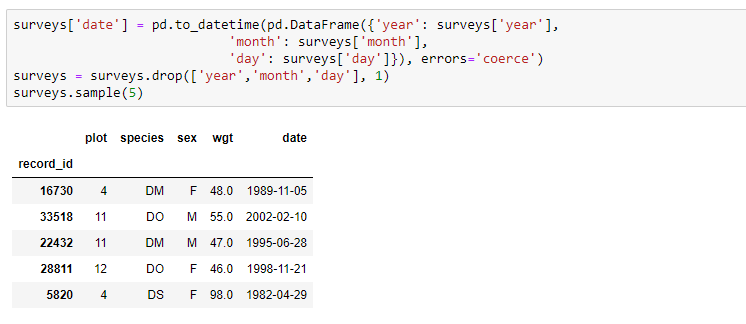
*Возвращает набор случайных данных из массива:*



***Как мы видим, данные очень неоднородные, например не везде указан вид наблюдаемого животного, его пол или вес. Также на разные колонки разбиты даты:***

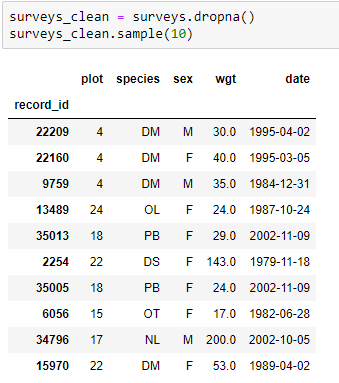
*.to\_datetime( )*

*Возвращает объект типа datetime64:*



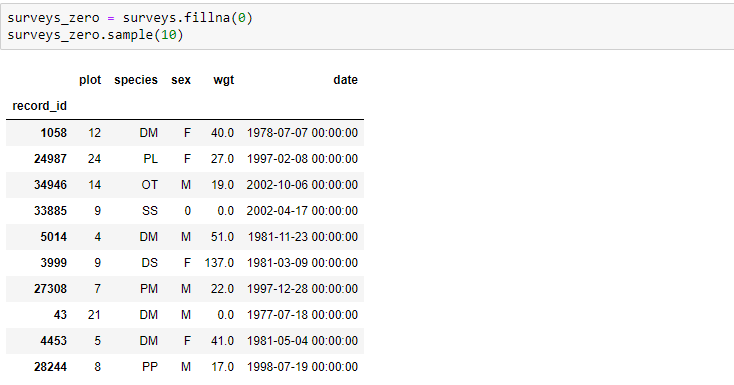
*.dropna( )*

*Удаляет строки с хотя бы одним пропуском:*

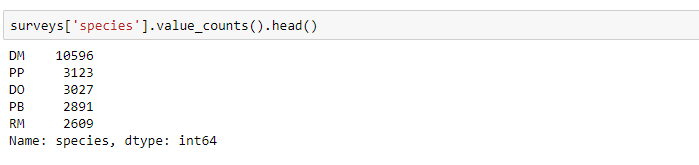


*.fillna( )*

*Заменяет пропуски на какие-то данные:*



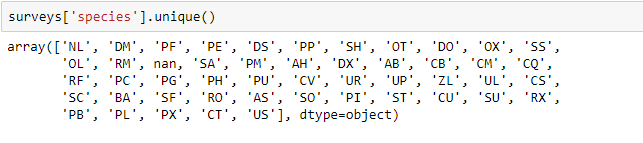
***Так как видов у нас много, можем подсчитать сколько раз встречается тот или иной вид:***



***Либо просто вывести их:***

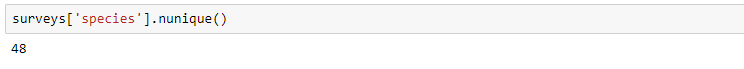
*.unique( )*

*Возвращает объект, содержащий уникальные данные:*



*.nunique( )*

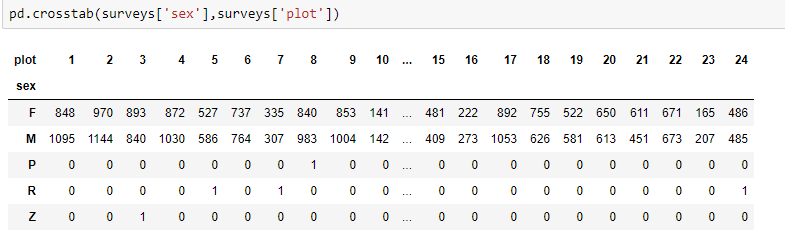
*Возвращает их количество:*



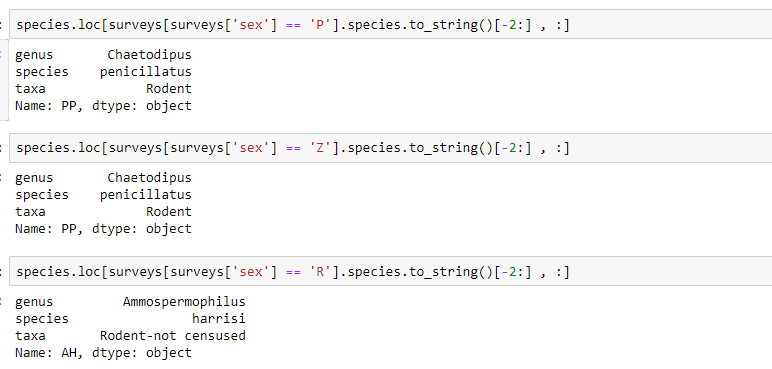
***Pandas позволяет составлять сводные таблицы по двум или нескольким параметрам:***

*.crosstab( )*

*Возвращает сводную таблицу:*

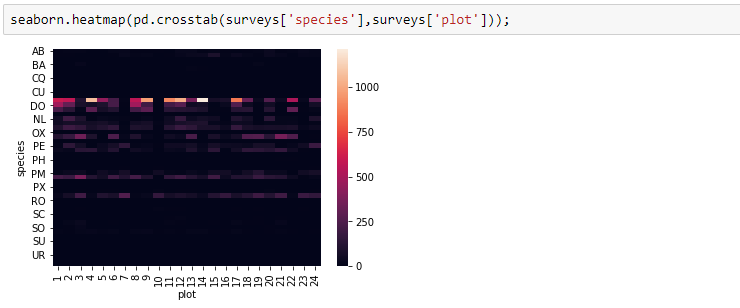


***Заметим, что помимо привычных M/F, в таблице присутствуют еще три строки***



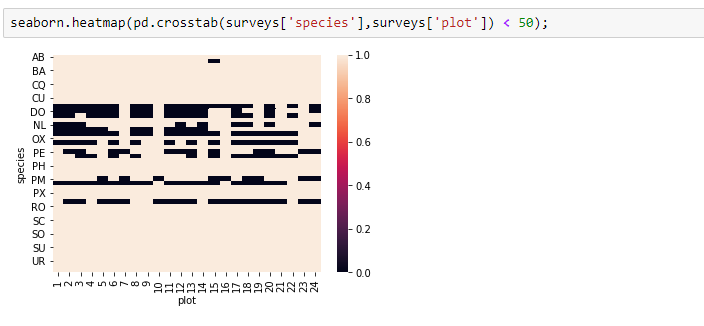
По имени вида существа, мы найдем, что Chaetodipus penicillatus является сумчатым *(pocket)*, а значит P — это тот, кто вынашивает, а Z - тот, кто не вынашивает детеныша в своей сумке. А Ammospermophilus harrisi напротив, практически не имеет внешних половых признаков, а значит ученые просто не смогли определить пол животного.

***Составив тепловую карту (библиотека seaborn) виды/участки, мы заметим, что есть виды, которые встречаются крайне редко на всех участках, а также участки, на которых количество животных заметно ниже, нежели на других:***

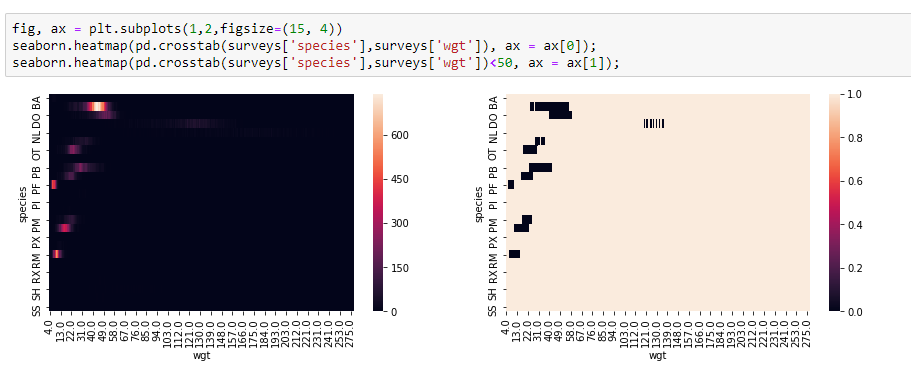


***Можем поставить условие, и показать какие виды встречаются меньше 50 раз:***

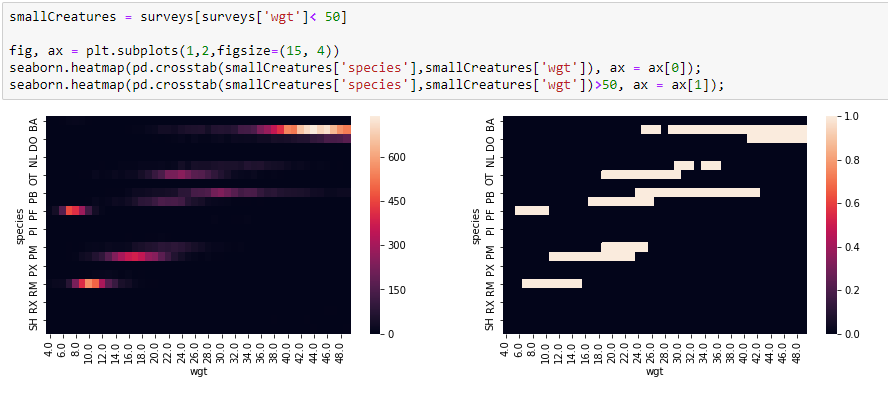
*Т.к. результат будет в формате true/false, белый цвет покажет, что значение истинное, а черный, соответственно, что ложное*



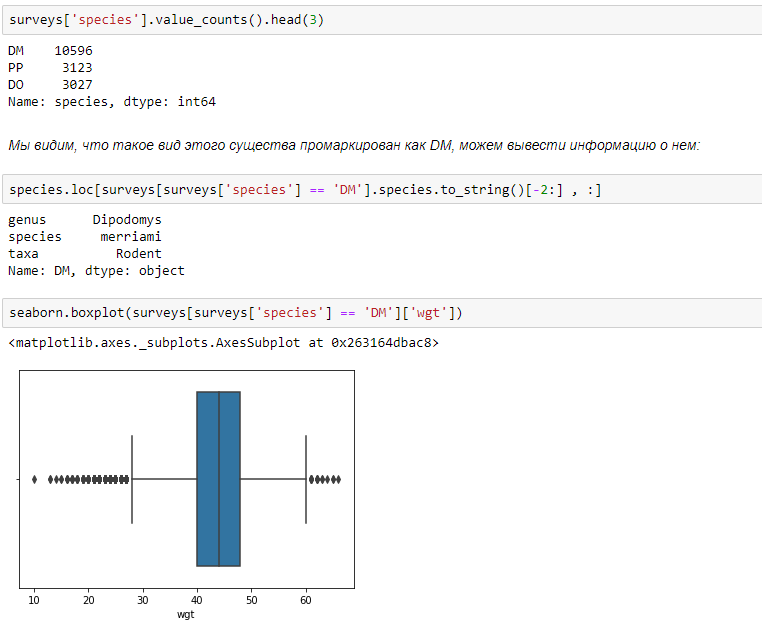
***Проверяя весь датасет на вес существа, увидим, что подавляющая масса зарегистрированных существ весит меньше 50 грамм:***



***Мы можем создать для них отдельный датасет и далее работать с ним:***



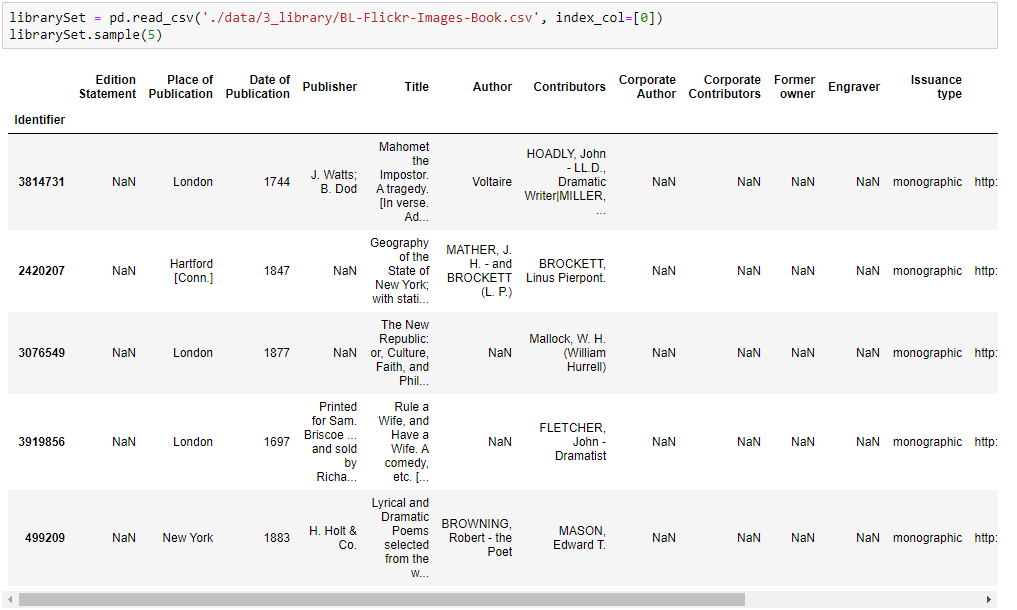
***Построим ящик с усами для веса самого часто наблюдаемого существа***



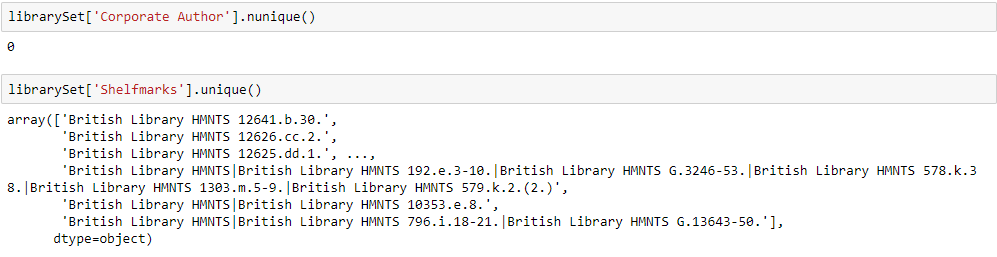
*Мы видим, что медианный вес находится в районе ~43 грамм, а границы ящика примерно на 40 и 47 граммах. Значит примерно четверть замеров показала, что вес меньше, чем 40, и еще четверть больше 47. Усы находятся на 1,5 интерквантильных расстояния (ширины ящика) от медианы, то, что выбивается за эти усы, можем считать выбросами, слишком малыми значениями или слишком большими*

Теперь возьмем третий массив данных:

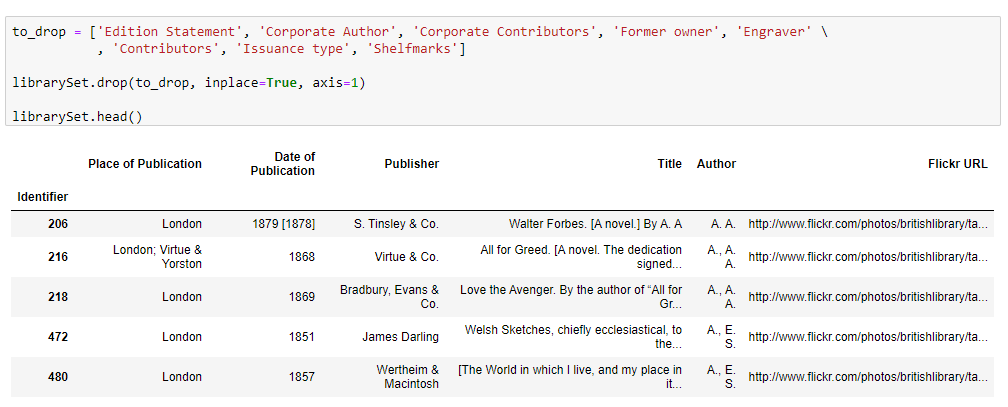
*Этот датасет содержит данные о книгах, находящихся в Британской Библиотеке; он отличается тем, что данные в нем неоднородны и перед их использованием, их нужно “почистить”*



***Видим, что многие столбцы содержат пустые, либо неинформативные***

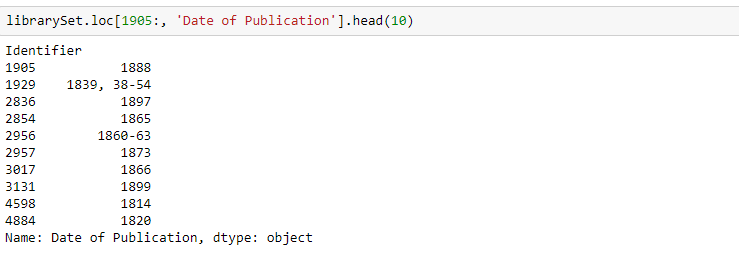


***Удалим колонки с неважными для анализа данными:***



***Как видим, даты тоже местами некорректны:***

Приведем даты к четырехзначному числовому формату *(в данный момент, у дат тип object)* и вычислим какая доля из них не имеет даты вовсе:

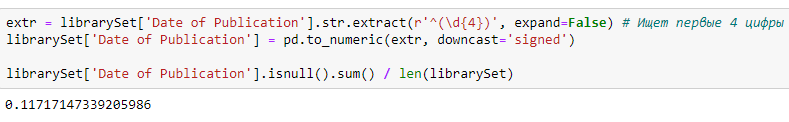


*.to\_numeric( )*

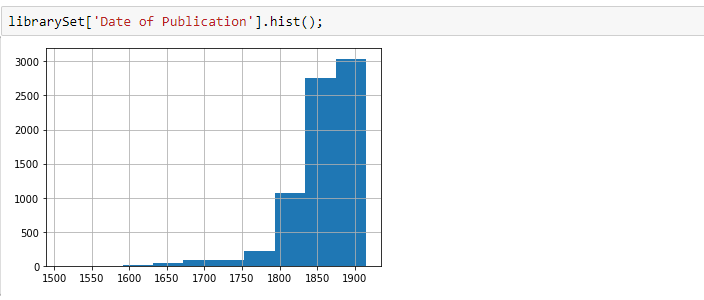
*Проверяет на то, является ли объект числом:*

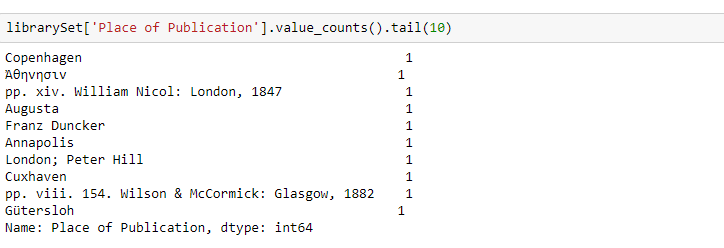
*.isnull( )*

*Проверяет на то, является ли объект нулевым:*



***Каждая десятая книга не имеет даты. Построим график распределения:***



***Так как датасет очень неоднородный, например, места тут зачастую дублируются на разных языках, его нужно чистить от повторяющихся или некачественных записей***

***Покажем на примере городов Лондон и Оксфорд***



Вывод

Библиотека Pandas - очень мощная библиотека для обработки больших массивов данных, их подготовки и непосредственно анализа и моделирования. Пакет прежде всего предназначен для очистки и первичной оценки данных по общим показателям. Статистическим пакетом он в полном смысле не является, однако наборы данных типов DataFrame и Series применяются в качестве входных в большинстве модулей анализа данных и машинного обучения (SciPy, Scikit-Learn и других).