## 3. Группировка данных

Слово «анализ» означает разбор, рассмотрение с разных сторон. Анализ данных начинают с разделения их на группы по какому-нибудь признаку. Эта операция называется **группировка данных**. Она помогает изучить материал более подробно, чтобы затем перейти к поиску взаимосвязей между отдельными группами.

Группировка оправданна, если данные чётко делятся по значимому признаку, а полученные группы близки к теме задачи. Например, когда есть данные обо всех покупках в супермаркете, можно смело заниматься группировкой. Так можно установить время наплыва покупателей и решить проблему пиковых нагрузок. Или посчитать средний чек — обычно для магазинов это ключевая метрика.

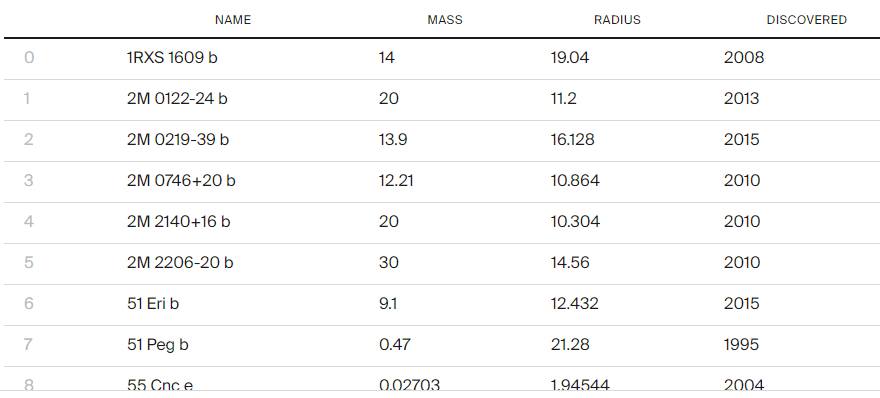
Стадии группировки хорошо укладываются в словесную формулу split-apply-combine (англ. split, apply, combine, «разделить, применить, объединить»):

* разделить, split — разбиение на группы по определённому критерию;
* применить, apply — применение какого-либо метода к каждой группе в отдельности, например, подсчёт численности группы методом count() или суммирование вызовом sum();
* объединить, combine — сведение результатов в новую структуру данных, в зависимости от условий разделения и выполнения метода это бывает DataFrame и Series.

В библиотеке Pandas есть отличные инструменты группировки. Рассмотрим обращение с ними на примере анализа данных о планетах за пределами Солнечной системы, или экзопланетах. Орбитальные обсерватории засекли уже тысячи таких небесных тел. Их выявляют на снимках космических телескопов наши коллеги, аналитики данных. Поищем среди экзопланет похожие на Землю. Возможно, это наши будущие колонии, или там уже обитают разумные существа, с которыми однажды предстоит установить контакт.

DataFrame с данными по нескольким тысячам экзопланет сохранён в переменной exoplanet. Посмотрим на первые 30 строк таблицы:

print(exoplanet.head(30))



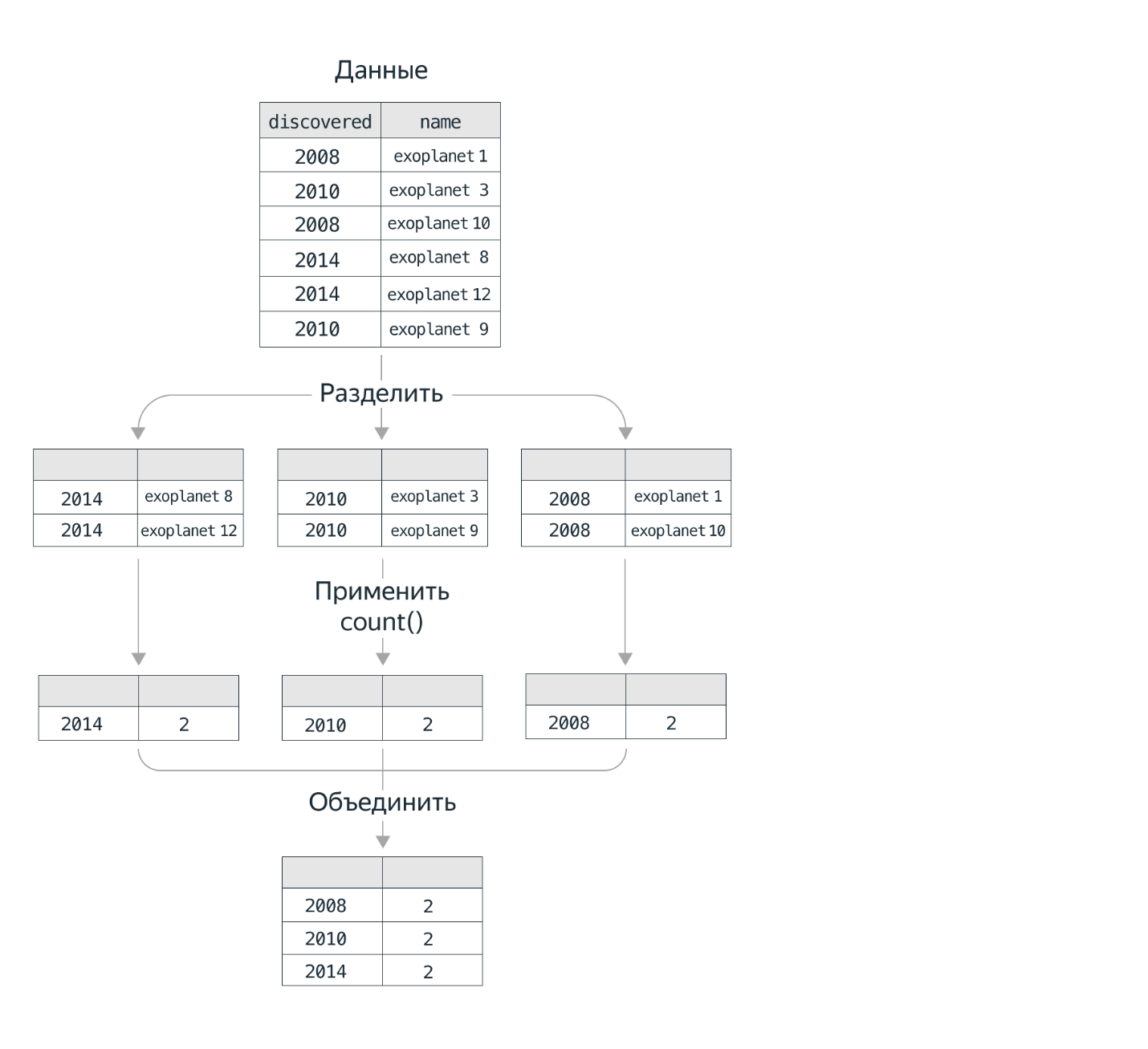
**Документация**

Столбцы:

* name: название экзопланеты;
* mass: масса в массах планеты Юпитер;
* radius: радиус, пересчитанный в радиусах Земли;
* discovered: год открытия экзопланеты.

Источник: каталог экзопланет на портале [**exoplanet.eu**](http://exoplanet.eu/catalog/) **http://exoplanet.eu/catalog/**

На картинке изображен принцип split-apply-combine для таблицы с экзопланетами. Посмотрим, как вообще идут дела с поиском экзопланет. Сначала данные делят по группам, где каждая группа — это год. Потом метод count() подсчитывает численность каждой группы. В итоге получаем новую структуру данных с группами, где каждая содержит год и число открытых за этот год экзопланет.



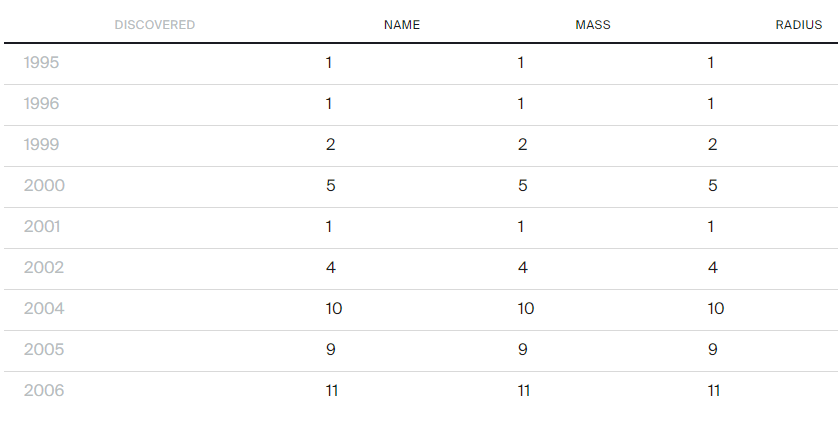
В Рandas для группировки данных есть метод groupby() (от англ. group by, «сгруппировать по»). Он принимает как аргумент название столбца, по которому нужно группировать. В случае с делением экзопланет по годам открытия:

print(exoplanet.groupby('discovered'))

<pandas.core.groupby.DataFrameGroupBy object at 0x7fc1e1ca3400>

Применение метода groupby() к объекту типа DataFrame приводит к созданию объекта особого типа — DataFrameGroupBy. Это сгруппированные данные. Если применить к ним какой-нибудь метод Pandas, они станут новой структурой данных типа DataFrame или Series.  
Подсчитаем сгруппированные по годам экзопланеты методом count():

print(exoplanet.groupby('discovered').count())



Результат выполнения кода exoplanet.groupby('discovered').count() — это уже новая структура данных, типа DataFrame. И с первого взгляда на этот DataFrame заметна тенденция: количество открытых экзопланет почти ежегодно растёт.

Если нужно сравнить наблюдения по одному показателю, метод применяют к DataFrameGroupBy с указанием на один столбец. Нас в первую очередь интересует радиус экзопланет: мы ищем другую Землю. Давайте получим таблицу с единственным столбцом 'radius':

exo\_number = exoplanet.groupby('discovered')['radius'].count()

print(exo\_number)

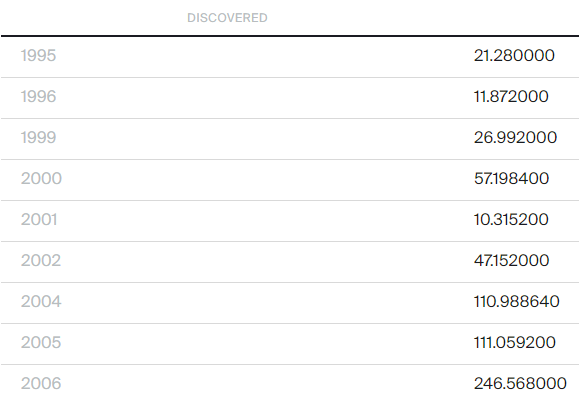


Получили Series, где по годам открытия расписано количество экзопланет, для которых удалось установить радиус.

Посмотрим, как меняется средний радиус открытых экзопланет год от года. Для этого надо сложить радиусы планет, открытых за определённый год, и поделить на их количество (которое мы уже нашли).

Сумма радиусов считается методом sum():

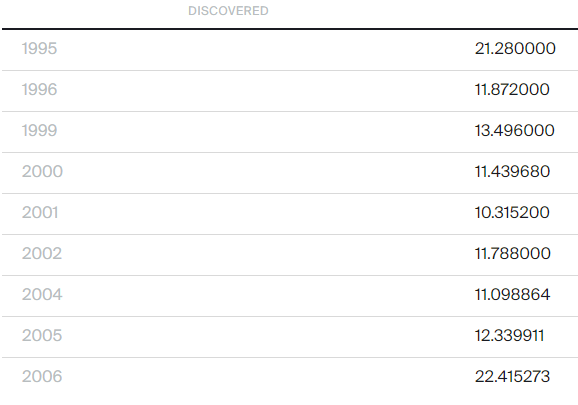
exo\_radius\_sum = exoplanet.groupby('discovered')['radius'].sum() print(exo\_radius\_sum)



Очень кстати, что объекты Series можно делить друг на друга. Это позволит нам разделить перечень сумм радиусов на перечень количеств экзопланет без перебора в цикле:

exo\_radius\_mean = exo\_radius\_sum/exo\_number

print(exo\_radius\_mean)



Точность наших приборов растёт, и новые экзопланеты по размерам всё ближе к Земле. За 24 года средний радиус обнаруженных планет снизился втрое.

Тем же методом groupby(), которым мы ищем новую Землю, можно поискать и необыкновенного человека в данных Яндекс.Музыки. Тем более, что без этого не выполнить поставленной менеджером задачи.

Прежде, чем рассчитывать метрику *happiness*, нужно изучить пользователей, чьё «счастье» мы собираемся оценить. Какие они, эти люди, которые слушают действительно много музыки? Есть ли у них особые предпочтения, или они потребляют всё подряд?