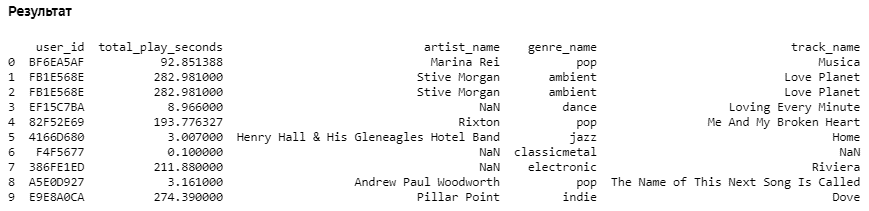
## 5. Обработка пропущенных значений

Теперь, когда названия столбцов нашей таблицы — её берега — обустроены, мы можем пуститься в плавание по самим данным. Вы помните, что столбец, который теперь называется artist\_name, содержит пустые значения (мы употребляем также синоним «пропущенные значения»):

print(df.head(10))



Замены пропущенных значений в DataFrame бывают трёх видов:

1) *Ожидаемые*: **None** или **NaN**. None (англ. *none*, «ничто») — это эквивалент null в других языках программирования: особое значение, указывающее, что в этой ячейке таблицы никакого значения нет. None относится к **NoneType** (от англ. *none type*, «тип никакой»). NaN (сокр. англ. *Not a Number*, «не число») говорит о том, что в ячейке находится «не число». Основное отличие NaN в том, что его можно использовать в математических операциях, так как по типу это число с плавающей запятой.

2) *Странные*: плейсхолдеры (тексты-заполнители) какого-нибудь общепринятого стандарта, иногда неизвестного вам, но которого придерживаются составители. Чаще всего это **n/a**, **na**, **NA** (от англ. *no answer*, «нет ответа»), и **N.N.**либо **NN** (от лат. *Nomen nescio*, «не знаю имени»).

3) *Неожиданные*: например, разработчики решили, что пустые значения в таблице будут заполняться знаками вопроса или нулями. В лучшем случае этот факт укажут в документации, в худшем – придётся просматривать данные самостоятельно. Если какой-нибудь спецсимвол или число встречаются часто, и этому нет внятного объяснения, то высока вероятность, что так передаются пропущенные значения.

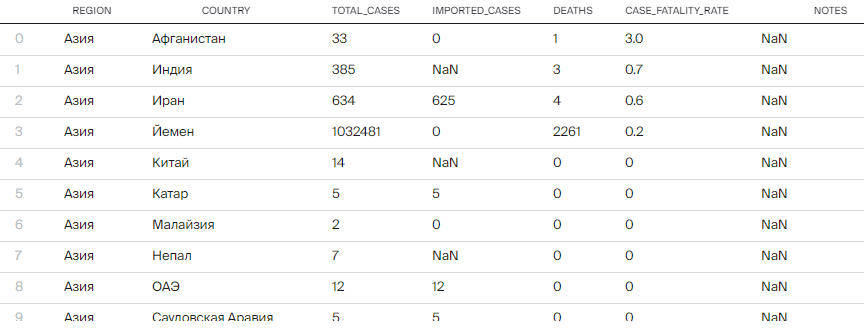
Но будьте осторожны: иногда нули — это действительно нули, как в наборе данных из нашей задачи, где ноль показывает, что трек был пропущен (его слушали 0 секунд).

Рассмотрим методы борьбы с пропущенными значениями на примере данных ВОЗ о заболеваемости холерой в 2017 году:

import pandas as pd

cholera = pd.read\_csv('cholera.csv')

print(cholera)



**Документация**

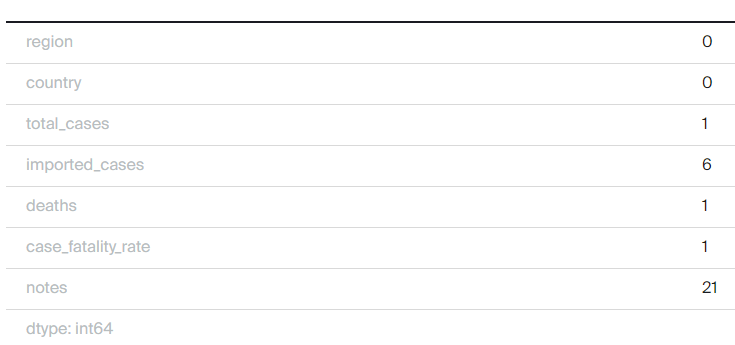
Для разных регионов (столбец *'region'*) и стран (*'country'*) указано общее число случаев заболевания холерой (*'total\_cases'*), в том числе завозные случаи (*'imported\_cases'*) и смертельные (*'deaths'*). Всё это целые числа, поскольку данные приведены с точностью до человека. Столбец *'case\_fatality\_rate'* содержит процент летальных исходов (тип *float*), столбец *'notes'* — строки с примечаниями.

Источник: [*Всемирная организация здравоохранения, Weekly Epidemiological Record), 21 September 2018, vol. 93, 38, pp. 489–500.*](http://apps.who.int/iris/bitstream/handle/10665/274654/WER9338.pdf?ua=1)

<https://apps.who.int/iris/bitstream/handle/10665/274654/WER9338.pdf?ua=1>

Посчитать в каждом столбце отсутствующие значения можно методом **.isnull()** (от англ. is null, «является ничем»). Если значение элемента не существует, .isnull() возвращает True, а иначе — False. Суммируют эти True вызовом метода **sum()** (англ. sum, «сумма»), который в этом случае возвращает общее число элементов без определённых значений.

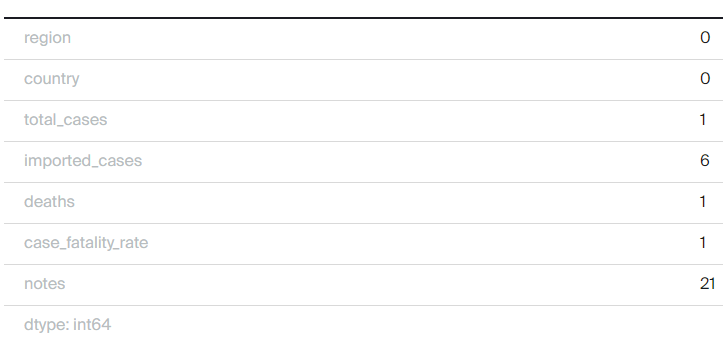
print(cholera.isnull().sum())



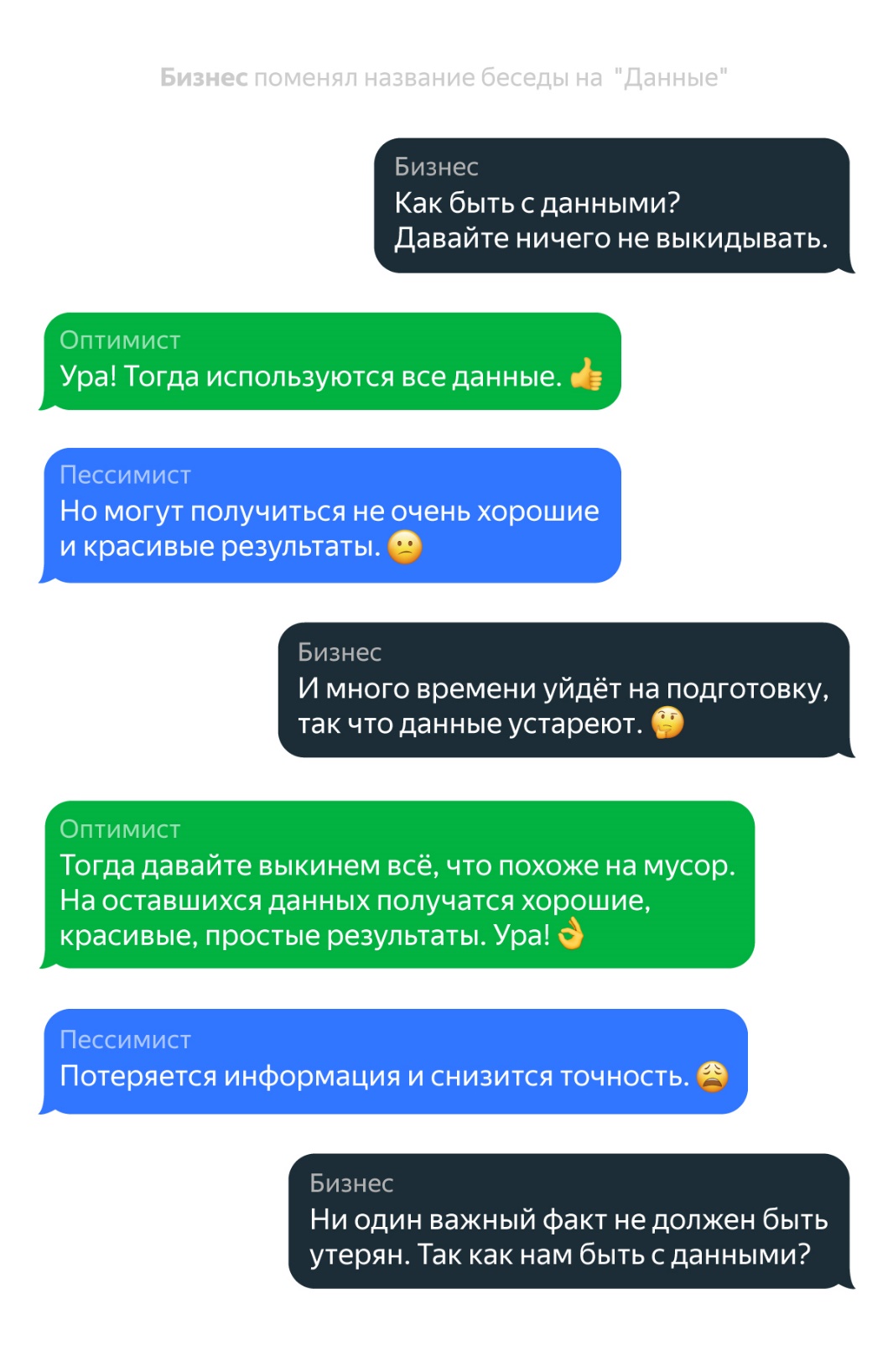
Обратите внимание на цепочку вызовов с точечной нотацией. Мы и прежде вызывали один метод, передавая ему другой метод как аргумент. Например, print(type(df)) . А сейчас результат работы метода isnull()передан методу sum(), и записано это через точку.

Также подойдёт метод **isna()** (от английского is NA, «является NA» — что такое NA, мы уже говорили), подсчитывающий пустые значения. В таблице по холере пропущенные значения качественные, так что этот метод отыщет их все.

print(cholera.isna().sum())



Метод борьбы с пропущенными значениями, который вы выберете, должен учитывать интересы решения конечной задачи бизнеса. В целом есть два пути: **заполнить** пропущенные значения на основе имеющихся данных или **удалить** все строки с пропущенными значениями.



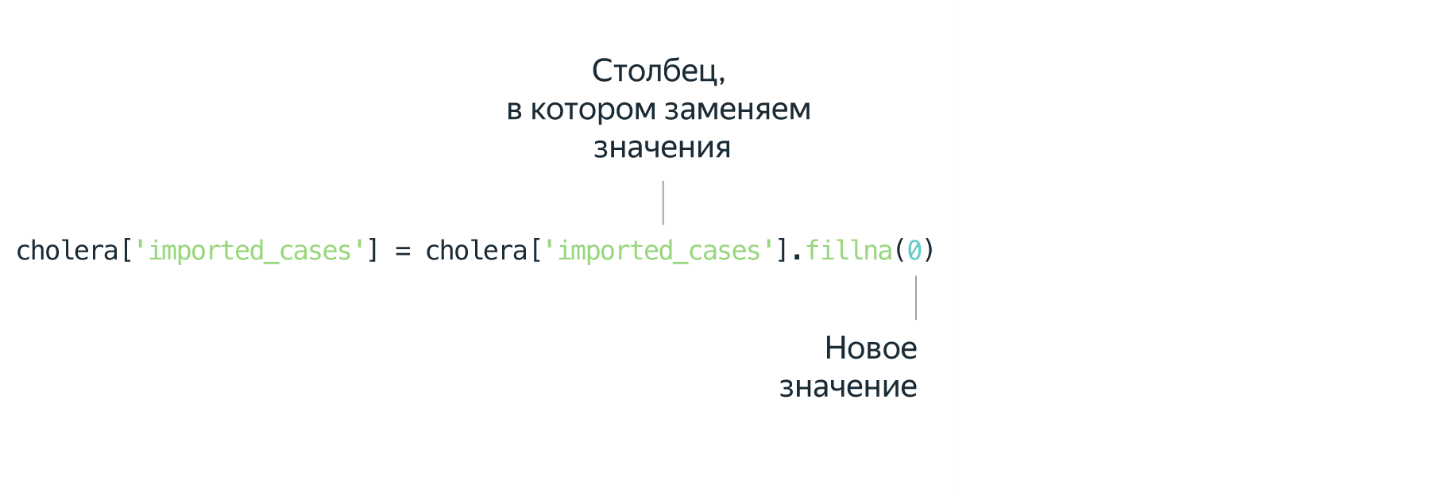
Бизнес разрывается между двумя решениями: как потратить не слишком много времени и как не упустить данные, которые обычно чего-то стоят.

Если сейчас пойти по пути удаления всех строк, где есть пропущенные значения, можно потерять важные данные, например, в Африке было зафиксировано 179835 случаев заболевания, но ни одного завозного случая (это сейчас отмечено NaN). Удалив эту строку, мы потеряем важные данные для статистики.

Другая ситуация с Европой. Вся строка состоит из пропущенных значений. Примечание сообщает, что в 2017 году европейцы холерой не болели. Значит, строку можно удалить и это никак не повлияет на результат.

Чтобы не лишиться строк с важными данными, заполним значения NaN в столбце *'imported\_cases'* нулями.

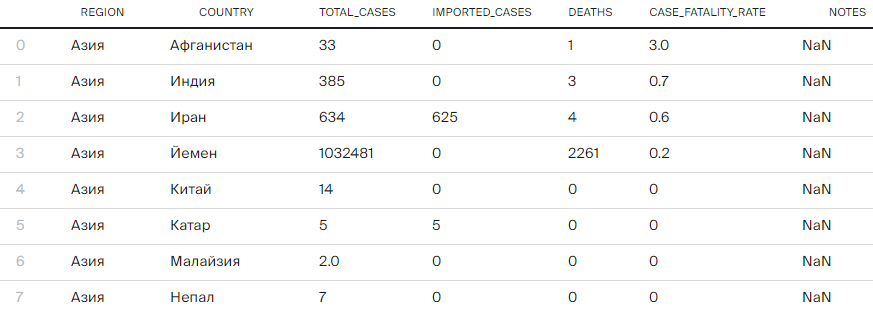
Для этого лучше всего использовать метод **fillna()** (от англ. *fill NA*, «заполни NA»), где в качестве аргумента выступает заменитель отсутствующих значений.



cholera['imported\_cases'] = cholera['imported\_cases'].fillna(0)

Теперь все пропущенные значения в столбце заполнены нулями.

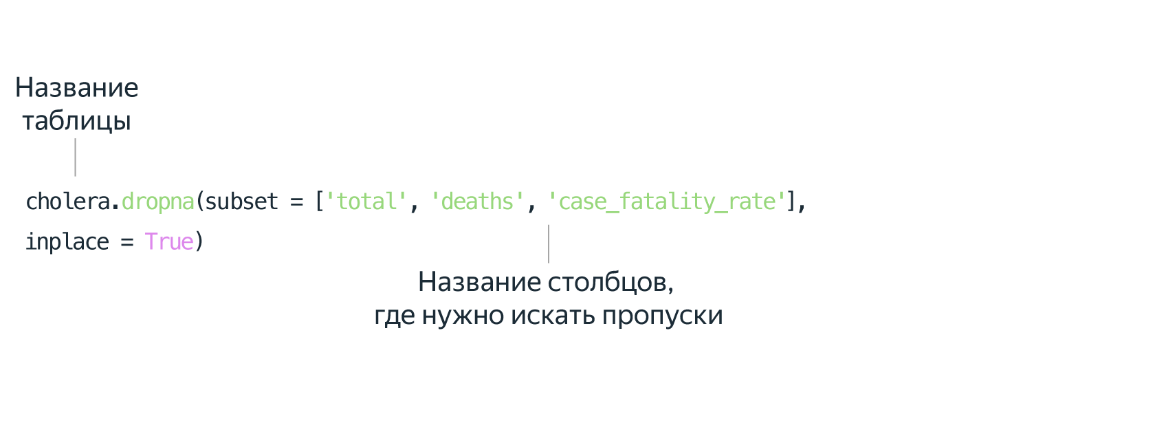
print(cholera)



От строк с нулевыми значениями избавляются методом **dropna()** (от англ. *drop NA*, «сбросить NA»). Он удаляет любую строку, где есть хоть одно отсутствующее значение.

У этого метода есть аргументы:

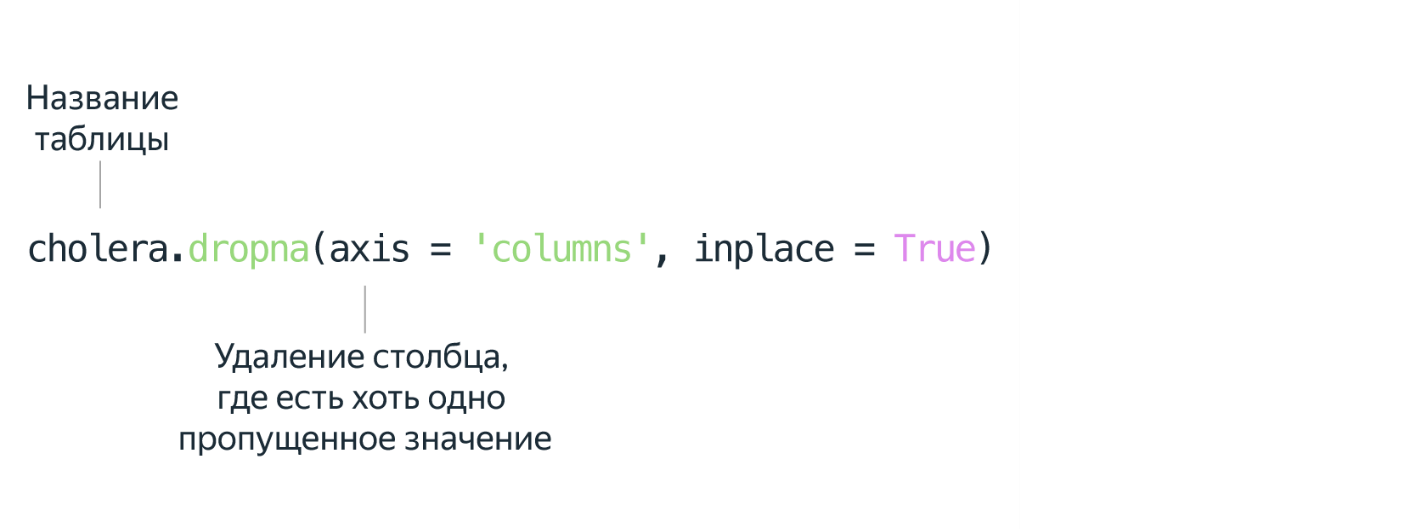
1. subset = [ ] (англ. *subset*, «подмножество»). Его значением указывают названия столбцов, где нужно искать пропуски.
2. Уже знакомый нам *inplace*.



cholera.dropna(subset = ['total\_cases', 'deaths', 'case\_fatality\_rate' ], inplace = True)

print(cholera)

Теперь удалим правый столбец с пропущенными значениями. Снова вызываем метод dropna(). Как и set\_axis(), он имеет ещё и аргумент *axis*. Если этому аргументу присвоить значение 'columns', он удалит любой столбец, где есть хоть один пропуск.



cholera.dropna(axis = 'columns', inplace = True)

print(cholera)

Теперь таблица готова к дальнейшему изучению: ненужные пропуски удалены, но при этом сохранены важные данные.

Пропуски в данных появляются разными путями. Например, пользователь не указал о себе какие-нибудь сведения или автоматизированная система сбора информации дала сбой. Иногда пропуски оставляют умышленно, рассчитывая на автозаполнение спецсимволами.

Давайте разберёмся с пропущенными значениями в данных для задачи Яндекс.Музыки.