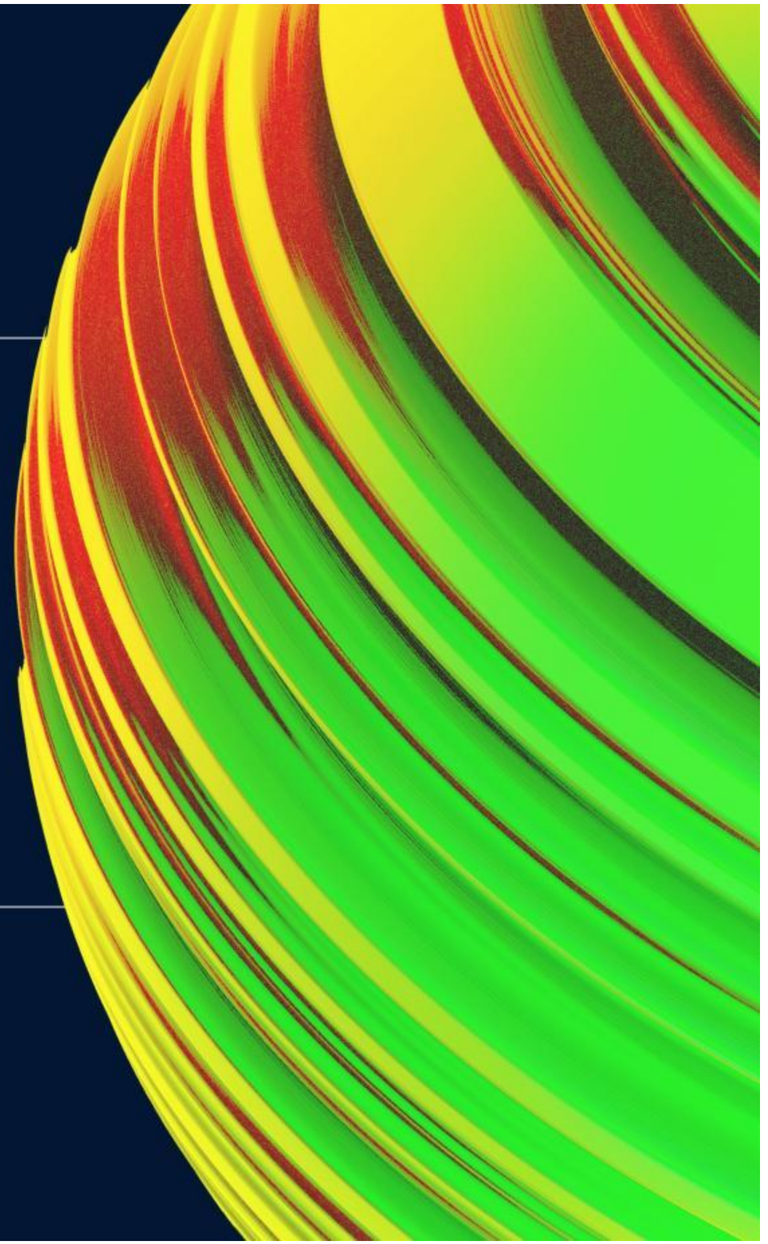
УНИВЕРСИТЕТ ИННОПОЛИС

Инструменты Python для обработки данных. Pandas



План занятия (лекция + семинар)

- ★ Series и DataFrame
- ★ Загрузка данных из файла
- ★ Исследовательский анализ данных (EDA) с библиотекой pandas:
  - ○ Загрузка данных
  - ○ Информация о таблице
  - ○ Пропуски и заполнение пропусков
  - ○ Квантиль, квартиль, персентиль
  - ○ Понятие выбросов
  - ○ Типы столбцов: категориальные, интервальные
  - ○ Фильтрация данных в pandas
  - ○ Примеры исследования на тестовых данных

Библиотека pandas

Pandas - высокоуровневая Python библиотека для анализа данных

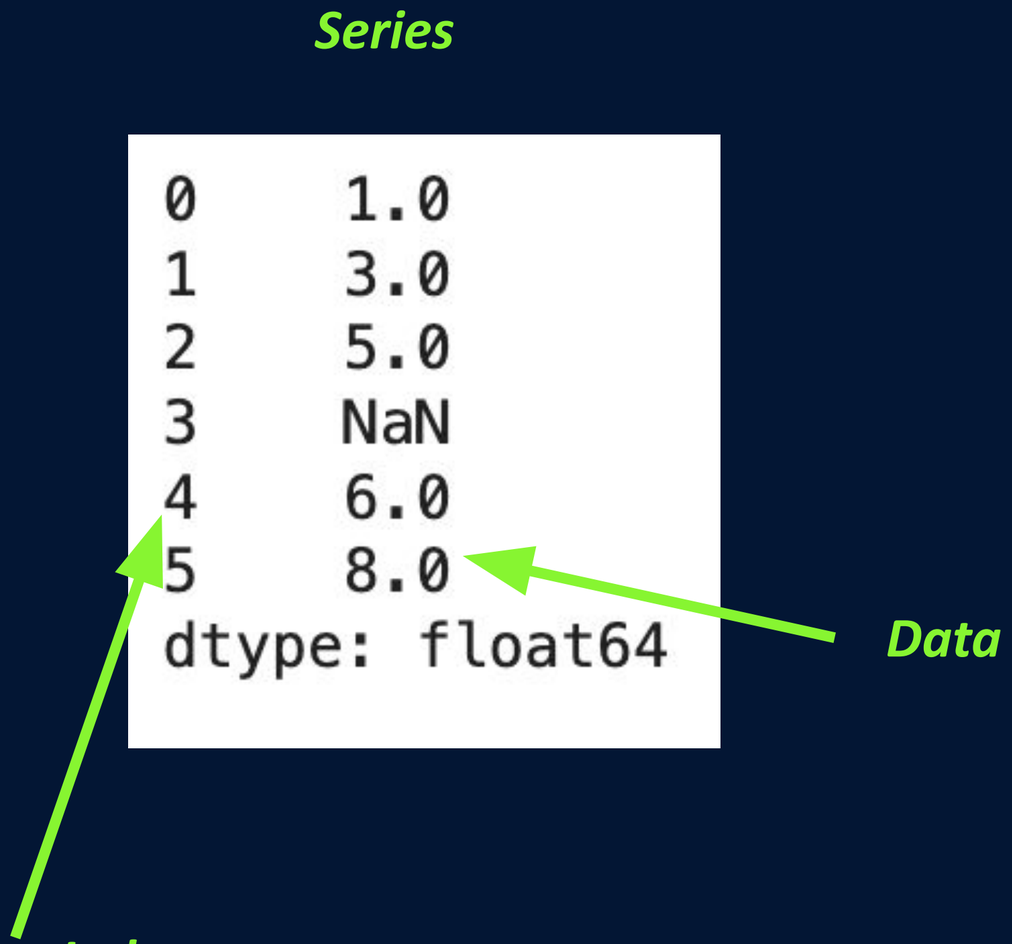
Построена над более низкоуровневой библиотеки ***NumPy*** (написана на Си)

Главные структуры данных библиотеки: ***DataFrame*** и ***Series***

Область применения

- сбор данных
- очистка данных
- анализ
- моделирование

Series



Series *— объект библиотеки pandas, спроектированный для представления одномерных структур данных, похожих на массивы*

Series и DataFrame

Series + *Series* = *Dataframe*

player_id	pts_per_game
201939	27.3
201142	26.0
202691	21.5
202326	16.3

+

player_id	reb_per_game
201939	5.3
201142	6.4
202691	3.8
202326	8.2

=

player_id	pts_per_game	reb_per_game
201939	27.3	5.3
201142	26.0	6.4
202691	21.5	3.8
202326	16.3	8.2

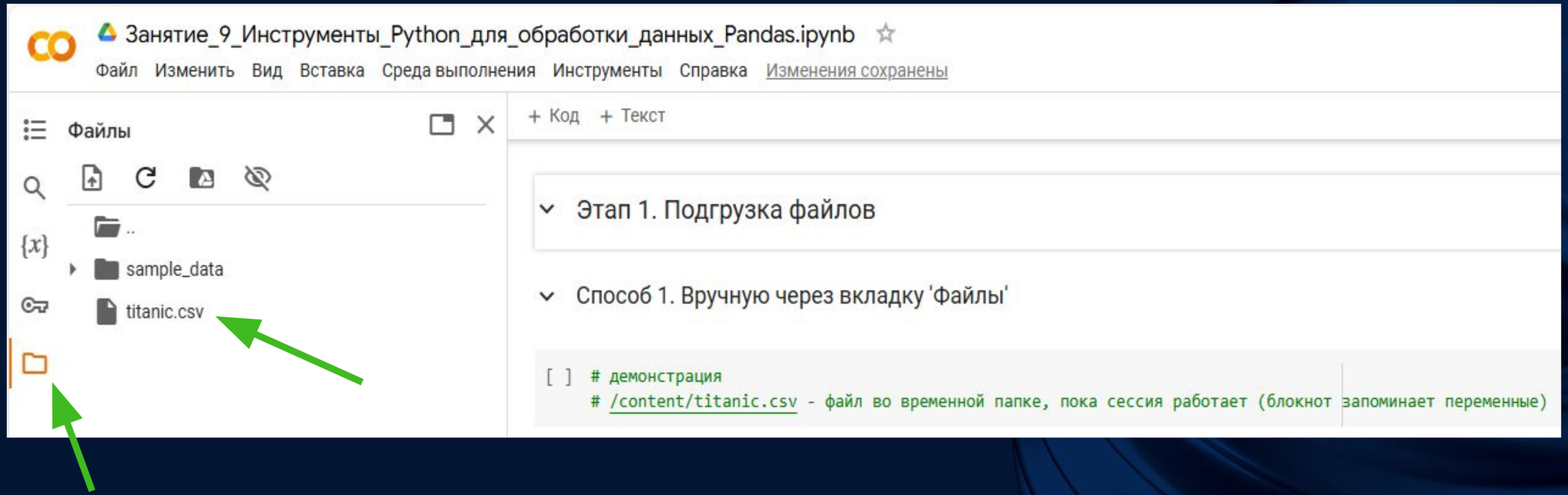
Index

***Series** — объект библиотеки pandas, спроектированный для представления одномерных структур данных, похожих на массивы, но с дополнительными возможностями («одномерный ndarray»)*

***Dataframe** состоит из колонок и строк. У колонок есть имена, а у строк — индексы. Табличная структура данных, напоминает таблицы из Microsoft Excel.*

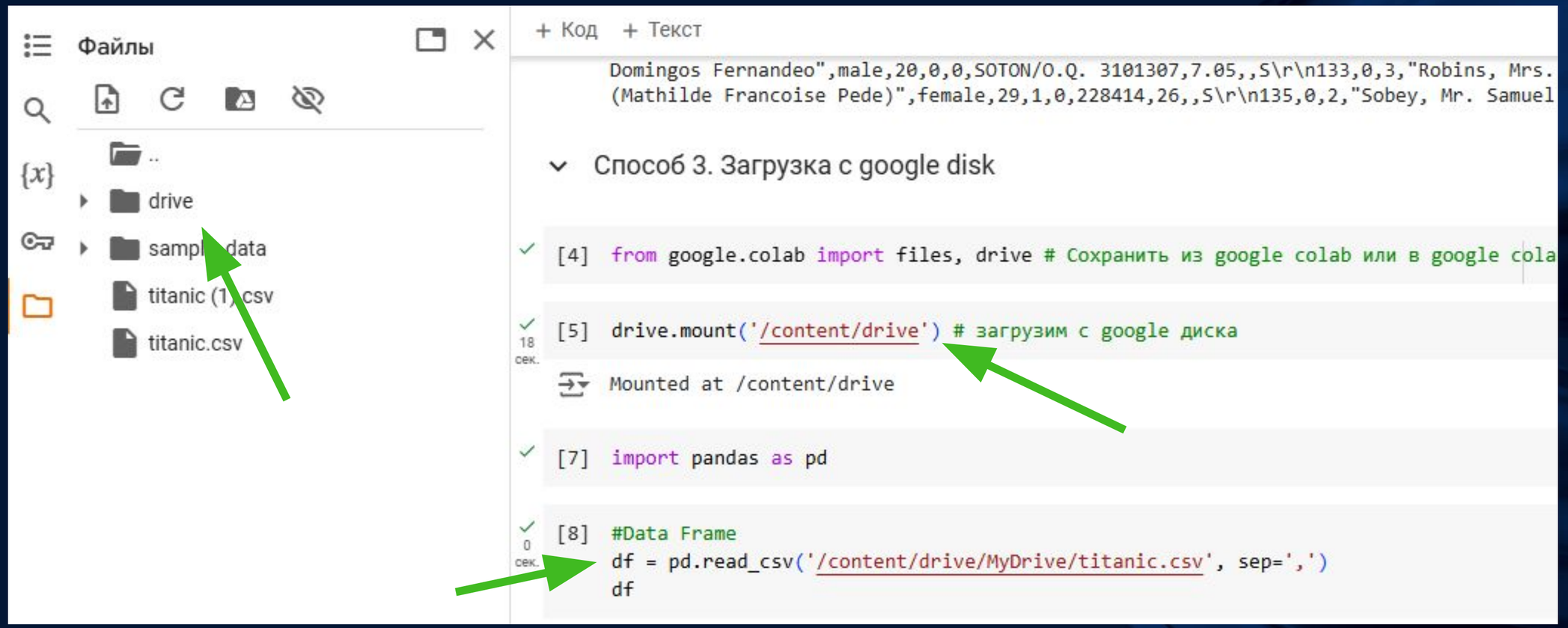
Подгрузка данных из файла. Вручную через вкладку «Файлы»

После загрузки файл .csv оказывается в сессионном хранилище в папке под названием /content/.



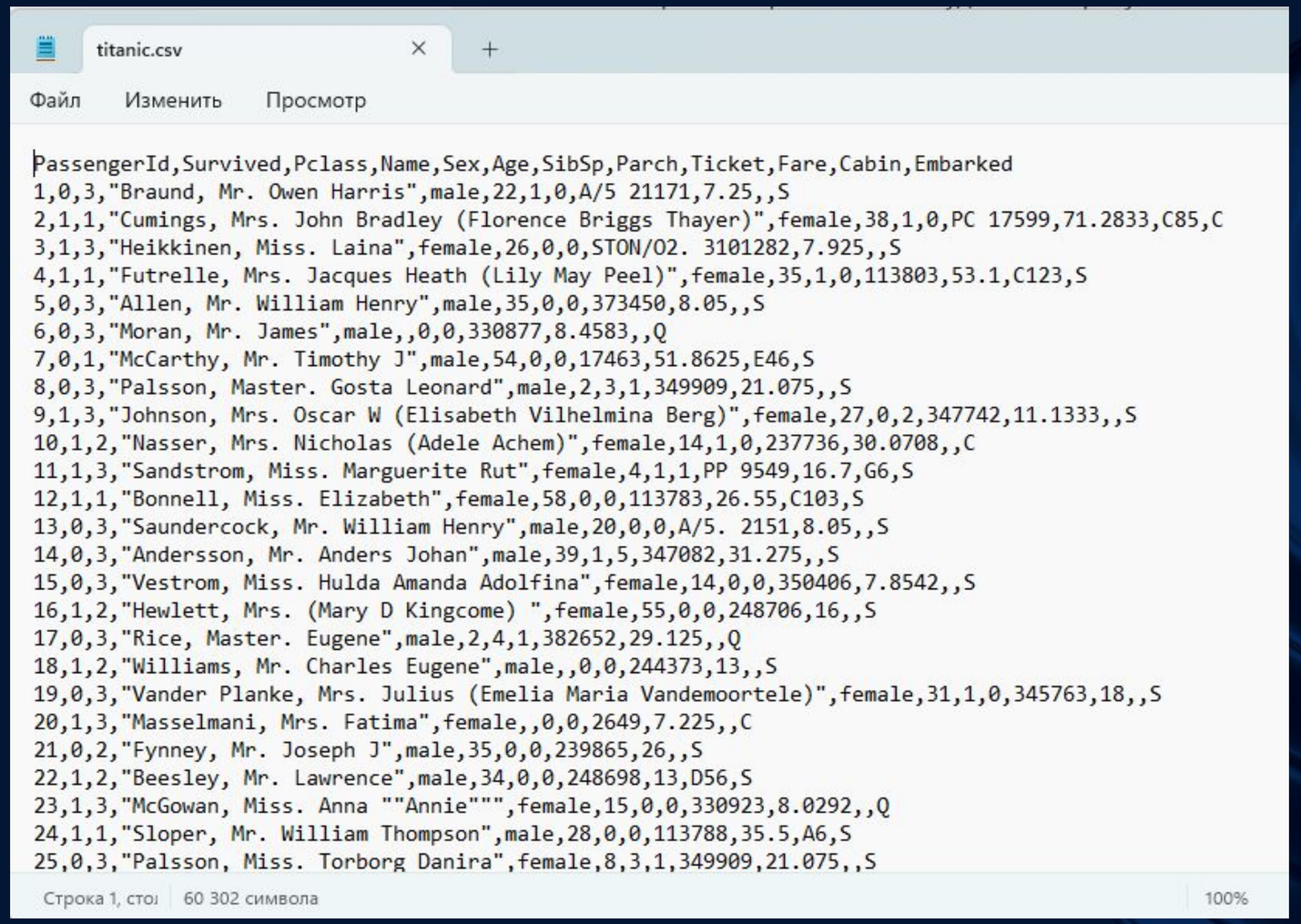
Подгрузка данных из файла. С подключением Google.Диск

После загрузки файл .csv оказывается на диске Google. Перед подключением к диску необходимо согласовать работу с персональными данными (вход в Google аккаунт обязателен).
Данные остаются на диске после завершении сессии блокнота Colab.



Что такое файлы CSV?

Файл CSV – это особый вид файла, который позволяет структурировать большие объемы данных. Пример CSV файла, где в качестве разделителя используется запятая:



Функция pandas.read_csv

Используется для чтения данных из файла CSV и преобразования их в DataFrame.

Основные параметры этой функции:

- **encoding:** позволяет указать кодировку файла, которую необходимо использовать при чтении данных (по умолчанию utf-8).
- **filepath_or_buffer**: путь к файлу с данными.
- **sep**: разделитель столбцов в файле (по умолчанию запятая ',').
- **header**: указывает, какая строка считается заголовком (по умолчанию header=0, но можно поменять на header=None, если заголовка в файле нет).
- **names**: измененный список названий столбцов.
- **index_col**: столбец-индекс DataFrame. Тип данных: число (индексация по столбцу), строка (имя столбца), дата.
- **dtype**: Типы данных для каждого столбца, при изменении указать словарь (key - имена столбцов, value - типы данных).
- **on_bad_numbers**: чтение строк с недопустимыми числами. Возможные значения: 'raise', 'skip', 'warn'.

Функция pandas.read_csv (2)

Используется для чтения данных из файла CSV и преобразования их в DataFrame.

Основные параметры этой функции:

- **na_values**: значения, которые должны интерпретироваться как отсутствующие значения (NaN).
- **keep_default_na**: если установлено значение False, то не будет использоваться набор стандартных значений для отсутствующих значений.
- **decimal**: символ, используемый в качестве десятичного разделителя (по умолчанию точка).
- **skip_blank_lines**: если установлено значение True, то пустые строки будут пропущены.
- **max_rows**: максимальное количество строк для чтения. Если достигнут этот лимит, чтение прекращается.
- **max_cols**: Максимальное количество столбцов для чтения. Если достигнут этот лимит, чтение прекращается.
- **dayfirst**: чтение дат, где день должен идти перед месяцем. Вместе с параметром **date_format (**позволяет указать формат даты для столбцов**)**.

Пропущенные значения

Пропущенные значения – незаполненные, «пустые» ячейки в данных.

Виды пропусков:

- **Полностью случайные пропуски (missing completely at random, MCAR)** предполагают, что вероятность появления пропуска никак не связана с данными.
- **Случайные пропуски (missing at random, MAR)** — вероятность появления пропуска зависит от некоторой известной нам переменной.
- **Неслучайные пропуски (missing not at random, MNAR)** — вероятность появления пропуска зависит, в том числе, от фактора, о котором мы ничего не знаем.

Пропущенные значения

Источник данных	Причины	Вероятность возникновения	Как восстановить данные?
Социальные сети / Веб-страницы	Заполняемость людьми	Высокая	Автоматическая/сбор/отсутствие возможности восстановления
СМИ и журналистика	Разные годы издания/отличные способы оформления	Низкая, обычно стандартное наполнение	Дополнительный сбор / недоступность
Статистика/опросы	Заполняемость людьми	Высокая	Через автоматическую обработку
Государственные данные	Заполненность/отсутствие информации о чем-либо	Средняя	Дополнительный сбор / Удаление
Административные данные	Заполненность/отсутствие информации о чем-либо	Средняя	Дополнительный сбор / Удаление
Известные компании: Google, Яндекс и т. д	Нарушение структуры	Низкая	Через автоматическую обработку / Удаление

Что делать с пропусками?

1. определить количество пропущенных значений в каждом столбце;
2. понять, случайны они или нет;
3. оценить возможность их восстановления через дополнительный сбор данных;
4. применить один из методов обработки пропущенных значений.

Метод	Когда используем?
Удаление наблюдений	много наблюдений и мало пропусков или делаем предварительный анализ
Присвоение пропускам специальной категории	необходимо отметить пропуски, но значения NaN, null, 0 для исследования не подходят
Замена на основании распределения заполненных значений в столбце	много наблюдений и мало пропусков, плохо подходит для значений со строковыми переменными
Заполнение на основании других столбцов	когда можем прогнозировать пропущенные значения на основе доступных значений в столбце

Что делать с пропусками?

Решение:
заполнить средним значением из распределения или другие статистики из распределения
экстраполировать/интерполировать значения
удалить наблюдения с пропусками (делать ОСТОРОЖНО)

например, мы можем рассчитать среднее = 1198 и пропуски заменить на 1198

ID магазина	площадь	количество этажей	в ТЦ?	доход от магазина
1	1000	1	1	1000000
2	1569	2	0	200000
3	870	1	0	300000
4	2000	2	0	500000
5	900	1	1	600000
6	850	1	1	1000000
7	1700	2	1	200000
8		2	1	300000
9		2	0	500000
10	700	1	0	600000

ID магазина	площадь	количество этажей	в ТЦ?	доход от магазина
1	1000	1	1	1000000
2	1569	2	0	200000
3	870	1	0	300000
4	2000	2	0	500000
5	900	1	1	600000
6	850	1	1	1000000
7	1700	2	1	200000
8	1198	2	1	300000
9	1198	2	0	500000
10	700	1	0	600000

Что делать с пропусками?

Решение:
построить модель/формулу, которая восстанавливает значение на основании других характеристик

То есть строим регрессию на наблюдениях с 1 по 7 и 10. Целевая переменная y=площадь магазина, независимые признаки: x1 = “количество этажей” и x2=”в ТЦ”
y = 700 + 300*x1+100*x2

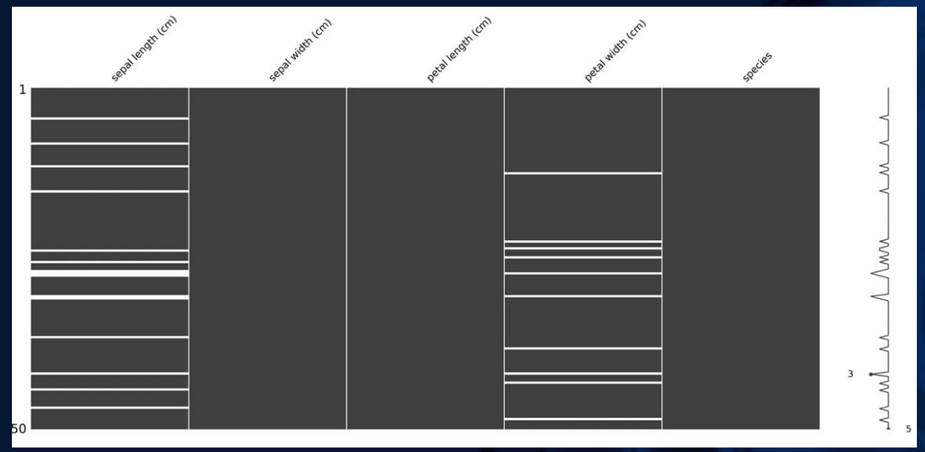
ID магазина	площадь	количество этажей	в ТЦ?	доход от магазина
1	1000	1	1	1000000
2	1569	2	0	200000
3	870	1	0	300000
4	2000	2	0	500000
5	900	1	1	600000
6	850	1	1	1000000
7	1700	2	1	200000
8		2	1	300000
9		2	0	500000
10	700	1	0	600000

ID магазина	площадь	количество этажей	в ТЦ?	доход от магазина
1	1000	1	1	1000000
2	1569	2	0	200000
3	870	1	0	300000
4	2000	2	0	500000
5	900	1	1	600000
6	850	1	1	1000000
7	1700	2	1	200000
8	1400	2	1	300000
9	1300	2	0	500000
10	700	1	0	600000

Что делать с пропусками?

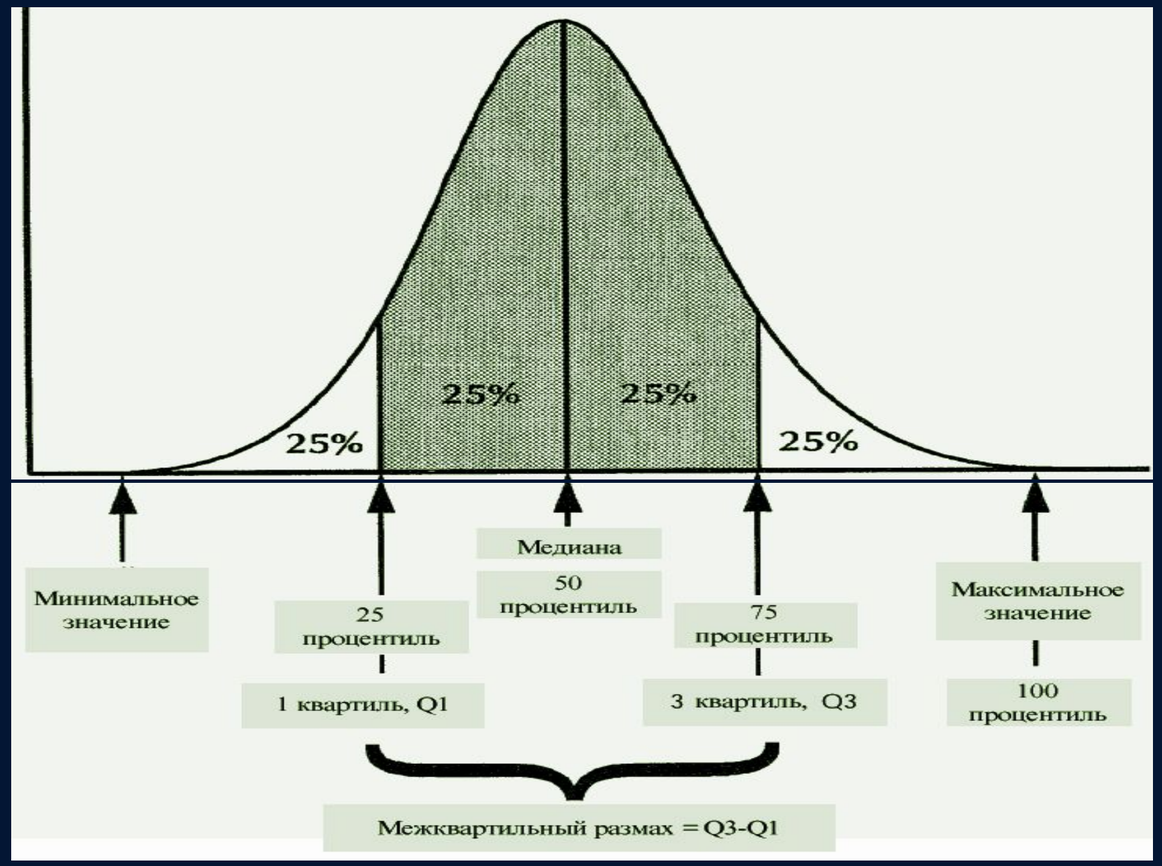
На практике:
Сначала необходимо обнаружить пропуски:
либо методы describe, либо isna().sum()
смотрим комплексно

```
import missingno as msno
msno.matrix(df_miss, figsize=(10, 6))
```



Далее заменяем:
просто методы Pandas fillna()
методы sklearn.impute
https://scikit-learn.org/stable/modules/impute.html#univariate-vs-multivariate-imputation
Datawig, Fancyimpute и MissForest и так далее

Квантиль, квартиль, перцентиль, межквартильный размах (IQR)



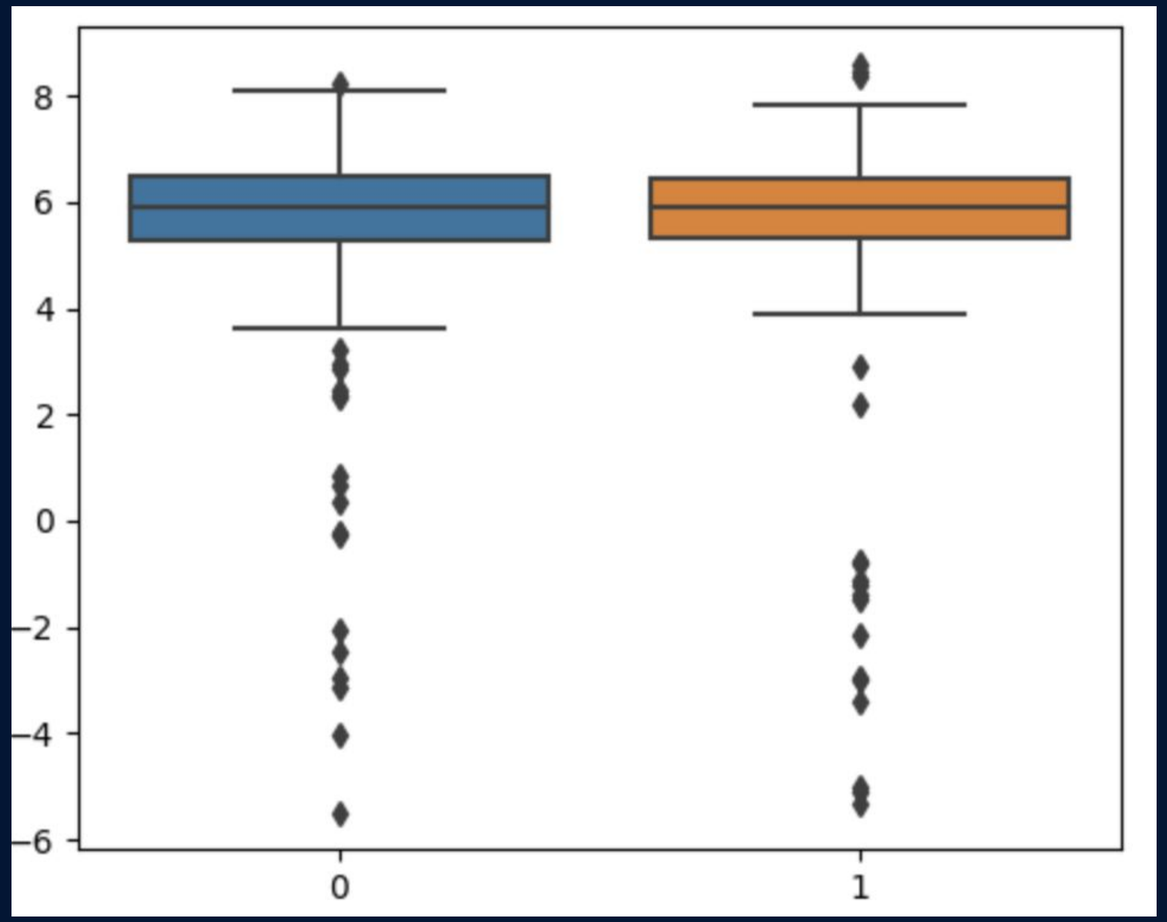
Что такое квантиль? Квантиль задает значение, ниже которого находится определенная доля данных в распределении.

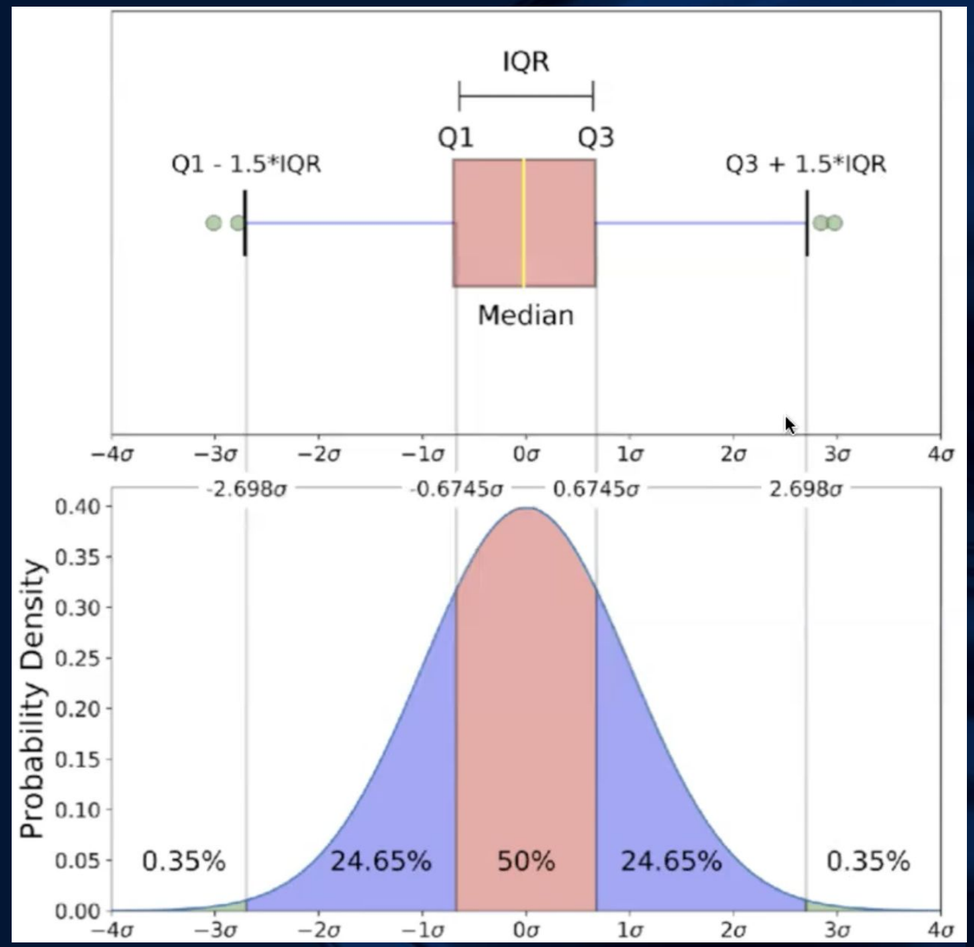
отсортируем ряд
разделим ряд на две части. Точка, которой мы делим - это и есть медиана. То есть ниже медианы 50% данных
Если мы ряд разделим на 3 равные части - то получим квартили. Ниже 1 квартиля 25% всех данных, ниже 2 квартиля - 50%, ниже 3 квартиля 75%

Получается, Квантиль задает значение, ниже которого находится определенная доля данных в распределении.
Медиана = 0.5 квантиль = 50% персентиль = 2 квартиль - это значение,

Понятие выбросов в данных

Выбросы — это данные, которые сильно отличаются от общего распределения.



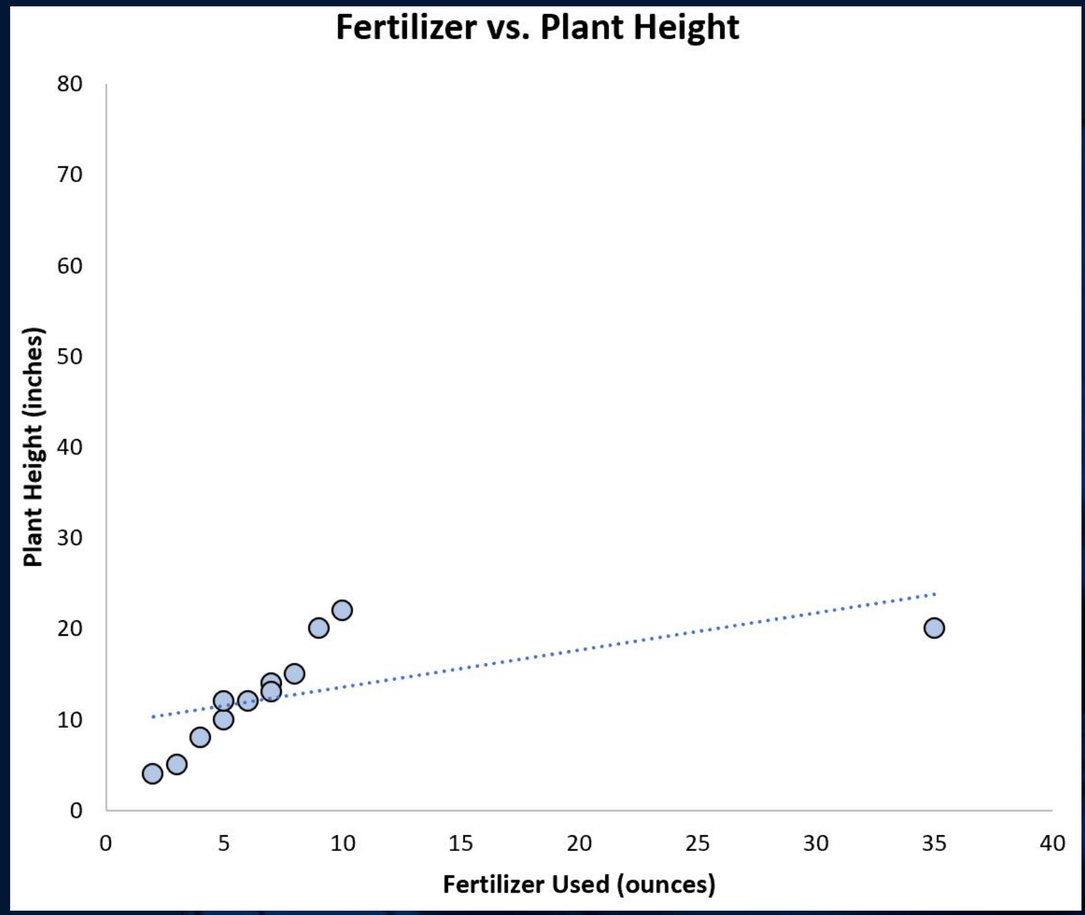


Понятие выбросов в данных

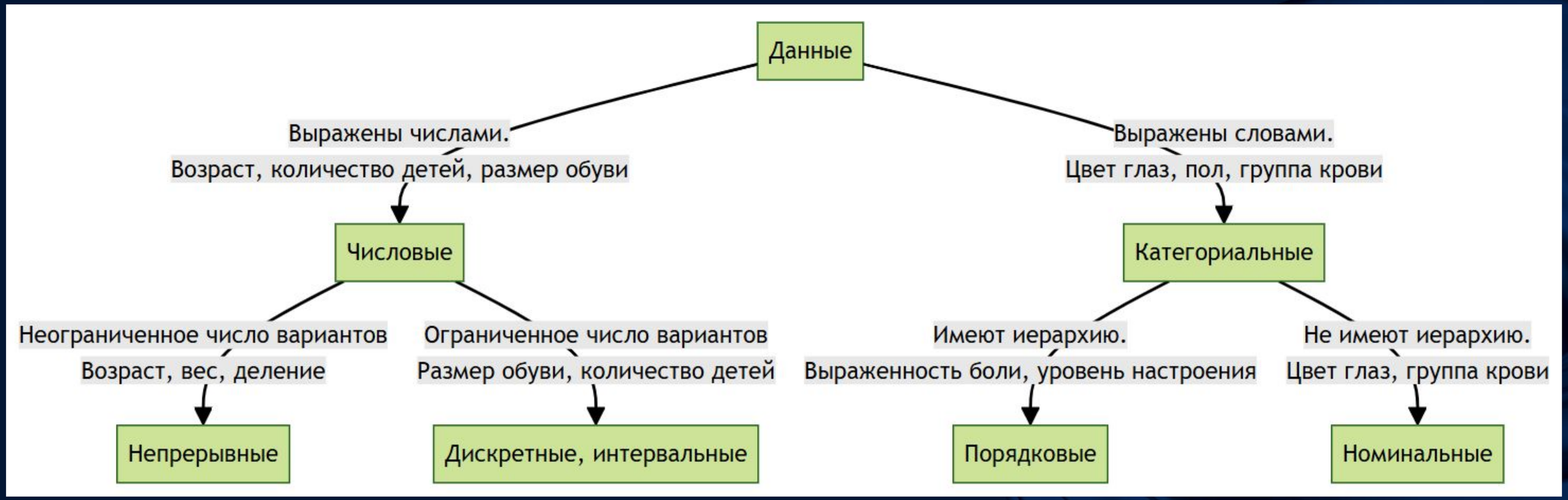
Выбросы — это данные, которые сильно отличаются от общего распределения.

- ***ошибочно возникающие выбросы:***
  - человеческий фактор, ошибка ввода данных
  - погрешности измерения
  - ошибка эксперимента (например, шум при записи голоса)
  - ошибка обработки
  - ошибки получения выборки (sampling error)

- ***естественные выбросы:***
  - высокий человек, разовая большая покупка
  - аномально низкая цена на отдельный объект недвижимости



Типы столбцов: категориальные, интервальные



Фильтрация данных в pandas

Фильтр по одному условию

```
In [4]: cars[cars>300]

Out[4]: Ferarri         350.0
        Lamborghini     380.0
        Bugatti         460.0
        dtype: float64
```

```
In [5]: cars[cars.isnull()]

Out[5]: Lada   NaN
        dtype: float64
```

```
In [7]: cars[cars.notna()]

Out[7]: BMW             230.0
        Mercedes        250.0
        Ferarri         350.0
        Lamborghini     380.0
        Bugatti         460.0
        dtype: float64
```

```
In [6]: cars[cars.notnull()]

Out[6]: BMW             230.0
        Mercedes        250.0
        Ferarri         350.0
        Lamborghini     380.0
        Bugatti         460.0
        dtype: float64
```

Фильтрация данных в pandas

Фильтр по нескольким условиям

```
In [8]: cars[(cars>300)&(cars<400)]

Out[8]: Ferarri         350.0
        Lamborghini     380.0
        dtype: float64
```

```
In [9]: cars[(cars<300)|(cars>400)]

Out[9]: BMW          230.0
        Mercedes     250.0
        Bugatti      460.0
        dtype: float64
```

Фильтрация данных в pandas

Фильтрация Series в Pandas при помощи цикла

```
In [10]: for index, value in cars.items():
             if value<300:
                 print(index)

BMW
Mercedes
```

Какие методы доступны в Pandas для работы с данными?

Метод	Когда используем?
read_csv(), read_excel()	Чтение данных
write_csv(), to_csv(), to_excel()	Запись данных
fillna(), replace()	для замены пропущенных значений, удаления строк или столбцов
drop(), dropna()	для удаления строк или столбцов
rename(), reindex()	для переименования столбцов, изменения индекса
sort_values(), groupby()	сортировки данных и группировки по одному или нескольким столбцам
sum(), mean(), min(), max(), count()	для подсчета суммы, среднего значения, минимального/максимального значения и количества элементов в группе (агрегация данных)
query(), loc[], iloc[], filter()	для фильтрации данных по условию, выборки по индексу или позиции и применения условий к столбцам

Какие методы доступны в Pandas для работы с данными?

Метод	Когда используем?
corr(), cov()	для расчета корреляций между переменными, оценки ковариации
describe()	для получения описательной статистики
plot(), hist(), boxplot(), scatter()	для создания графиков, гистограмм, боксовых диаграмм и рассеяния данных
apply(), map(), merge(), join()	для применения пользовательских функций к данным, сопоставления значений и объединения данных

Pandas также поддерживает множество других функций для работы с временными рядами, категориальными данными, географическими данными и др.

ДЕМОНСТРАЦИЯ

Примеры исследования на тестовых данных

Краткий словарь программиста

Pandas — высокоуровневая Python библиотека для анализа данных.

Series — объект библиотеки pandas, одномерные структуры данных с дополнительными возможностями («одномерный ndarray»).

DataFrame в Pandas — двумерная структура данных, таблица с возможностью хранения разных типов (числа, строки, boolean и др.). DataFrame позволяет манипулировать данными, выполнять агрегирующие операции, фильтрацию, сортировку, визуализацию данных.

Файл CSV – это особый вид файла, который позволяет структурировать большие объемы данных.

Пропущенные значения – незаполненные, «пустые» ячейки в данных.

Выбросы — это данные, которые сильно отличаются от общего распределения.

Полезные ссылки

1. Шпаргалка по pandas. URL: https://habr.com/ru/companies/ruvds/articles/494720/
2. Панды Упражнения, Практика, Решение. URL: http://kodesource.top/python-exercises/pandas/index.php
3. Наборы данных для анализа. URL: https://www.kaggle.com/datasets
4. Пропущенные значения. URL: https://www.dmitrymakarov.ru/data-analysis/nan-06/?ysclid=lx2eh242lk94041346#12-udalenie-strok
5. Кодирование категориальных переменных. URL: https://www.dmitrymakarov.ru/data-analysis/encoding-10/
6. Выбросы в данных. URL: https://www.dmitrymakarov.ru/data-analysis/outliers-09/
7. Обработка пропущенных значений. URL: https://education.yandex.ru/handbook/data-analysis/article/obrabotka-propushennyh-znachenij

УНИВЕРСИТЕТ
ИННОПОЛИС

ВОПРОСЫ И ОТВЕТЫ