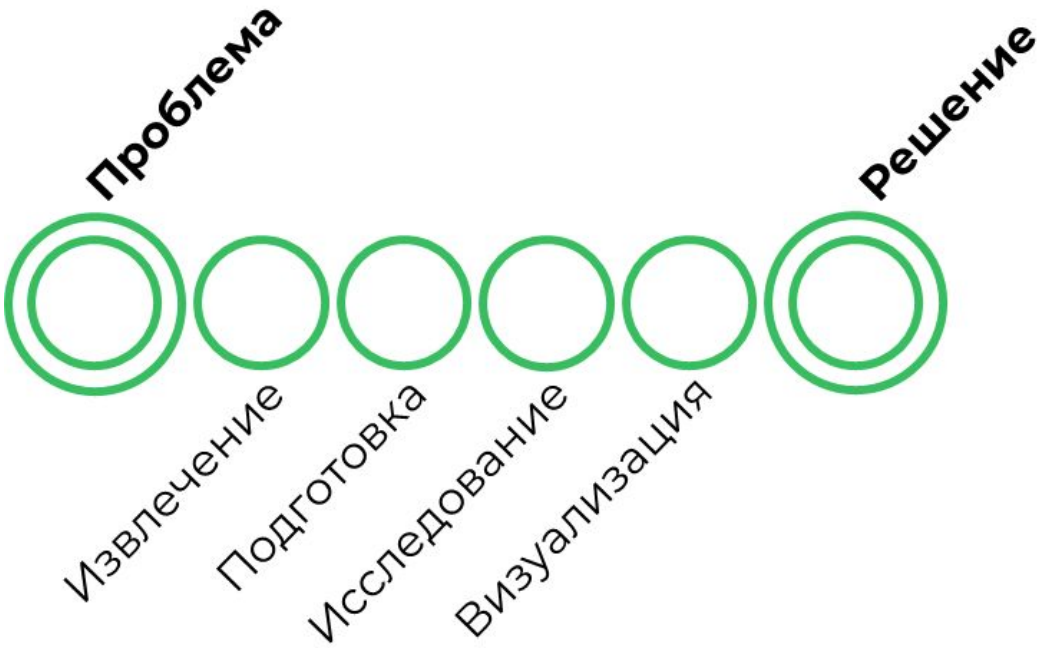
Модуль 3. Урок 5.

Визуализация данных

Ссылка на методичку

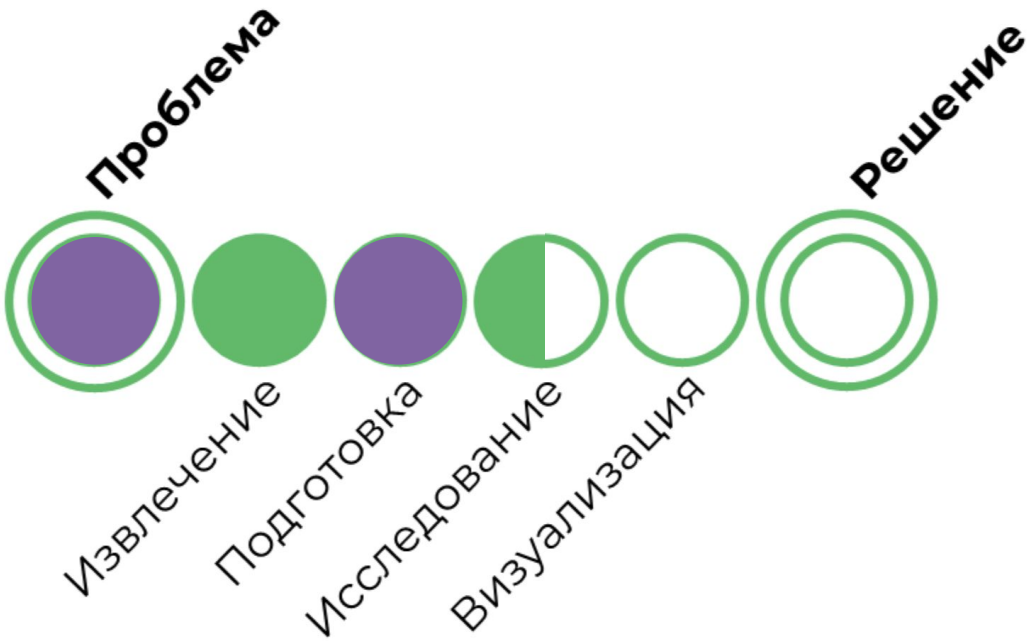
Обсуждение: Использование визуализации

Какие этапы процесса анализа данных вы выполнили на прошлом занятии?



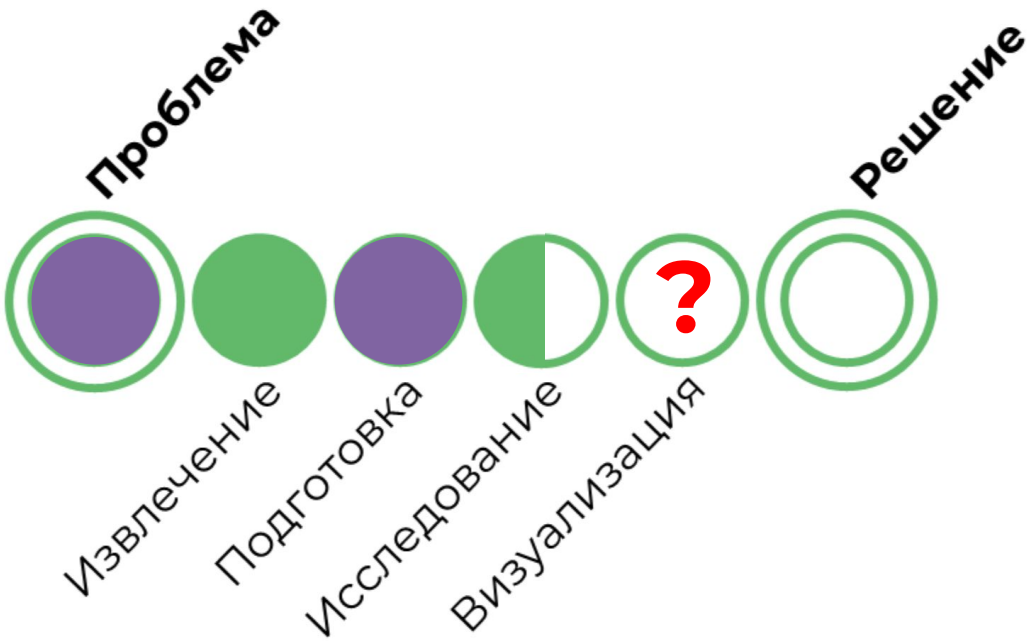
Обсуждение

Этапы процесса анализа данных, которые уже выполнены



Обсуждение

Как можно визуализировать данные?



Обсуждение

Популярные способы визуализации

К данным можно применить:

- ➔ **график**
- ➔ **диаграмма**
- ➔ иллюстрация
- ➔ видео
- ➔ ментальная карта

Обсуждение

Для чего нужен этап визуализации?

Можно ли его пропустить?

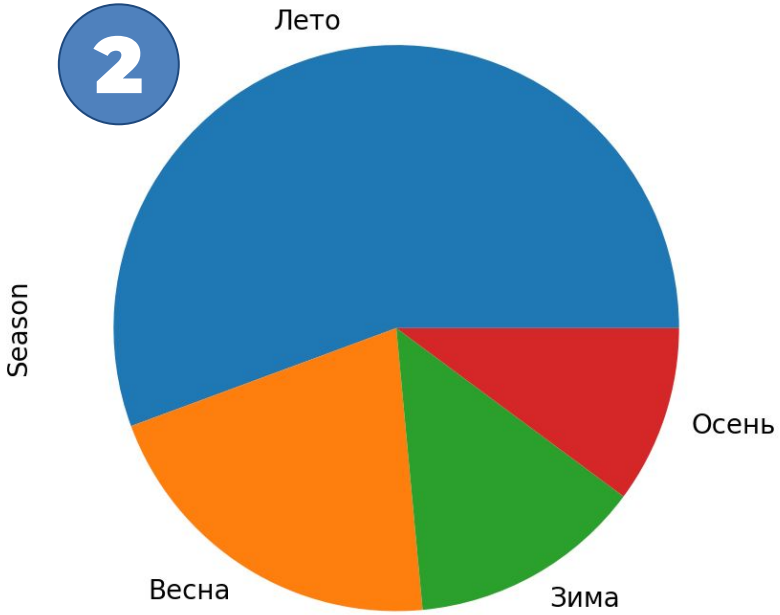
Обсуждение

Какие данные легче воспринимать?

1

Лето	6030
Весна	2261
Зима	1450
Осень	1099

2



Обсуждение

Визуальная информация **лучше воспринимается** и позволяет быстро и эффективно донести основную мысль до аудитории.

Обсуждение

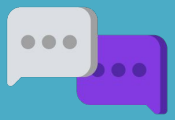
Многочисленные исследования подтверждают, что **90% информации человек получает посредством зрения**.

Для нас это наиболее физиологичный способ восприятия информации.

Обсуждение

Новая тема: Визуализация данных в Pandas

Построение графиков и диаграмм на этапе визуализации — одна из важнейших задач анализа данных.

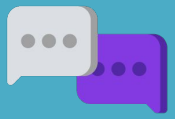


Визуализация данных

Какие виды диаграмм существуют?

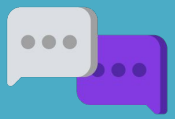
Виды диаграмм

- круговая;
- линейная;
- столбчатая;
- гистограмма;
- «ящик с усами»;
- и многие другие...



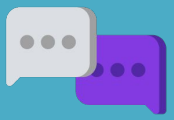
Визуализация данных

Как построить диаграмму на Python?



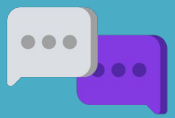
Визуализация данных

Для построения диаграм мы будем использовать **метод plot()** библиотеки **Pandas**.



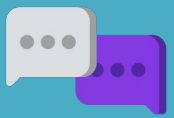
Визуализация данных

Matplotlib — это библиотека на языке программирования Python, с помощью которой мы будем отображать графики на экране.



Визуализация данных

Из Matplotlib мы будем импортировать **модуль pyplot**, для использования **метода show()**.



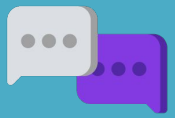
Визуализация данных

```
import pandas as pd
import matplotlib.pyplot as plt
```

Визуализация данных

Метод plot()

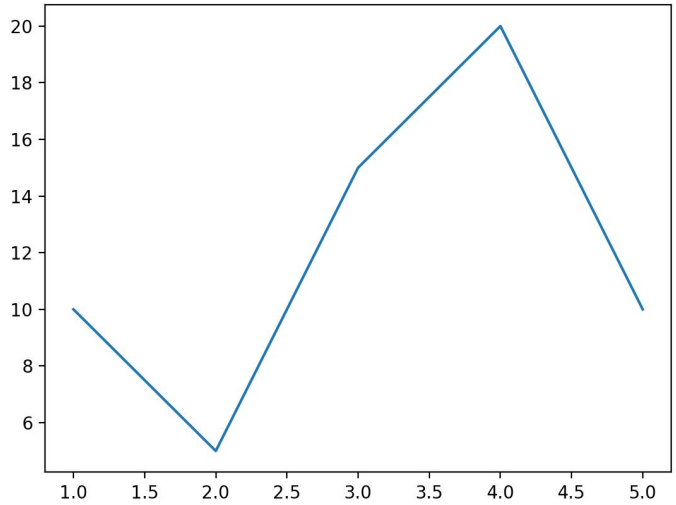
- Метод применяется как к объектам Series, так и к DataFrame.
- Метод строит диаграмму для количественных данных.



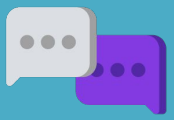
Визуализация данных

По умолчанию метод строит линейный график

```
import pandas as pd
import matplotlib.pyplot as plt
s = pd.Series(data = [10, 5, 15, 20, 10],
  index = [1, 2, 3, 4, 5])
s.plot()
plt.show()
```



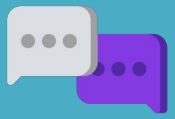
Для отображения каких данных можно построить линейный график?



Визуализация данных

Линейный график

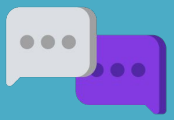
➔ Линейный график используют для наблюдения за изменениями какой-то величины на протяжении определённого периода.



Визуализация данных

Вернёмся к датасету, который хранит данные приложений из Google Play Store.

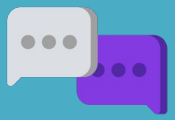
Какую информацию о приложениях мы можем изобразить в виде линейного графика?



Визуализация данных

Без дополнительных действий данные этого датасета нельзя продемонстрировать в динамике.

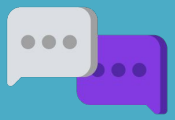
К данным нужно применять фильтрацию и группировку.



Визуализация данных

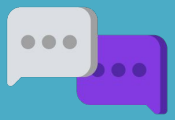
Но это не значит, что в ваших проектах вы не сможете построить линейный график!

Всё будет зависеть от выдвинутых вами гипотез.



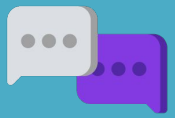
Метод plot() может строить диаграммы разных видов.

Вид диаграммы можно задать с помощью **параметра kind**.



Визуализация данных

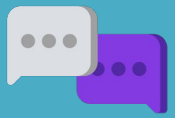
Рассмотрим, какие **значения** может принимать **параметр kind**, и какие диаграммы будут построены в зависимости от этих значений.



Визуализация данных

Параметр kind задаёт тип диаграммы

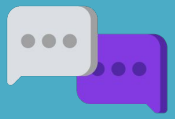
- hist;
- box;
- scatter;
- pie;
- bar;
- barh.



Визуализация данных

Гистограмма

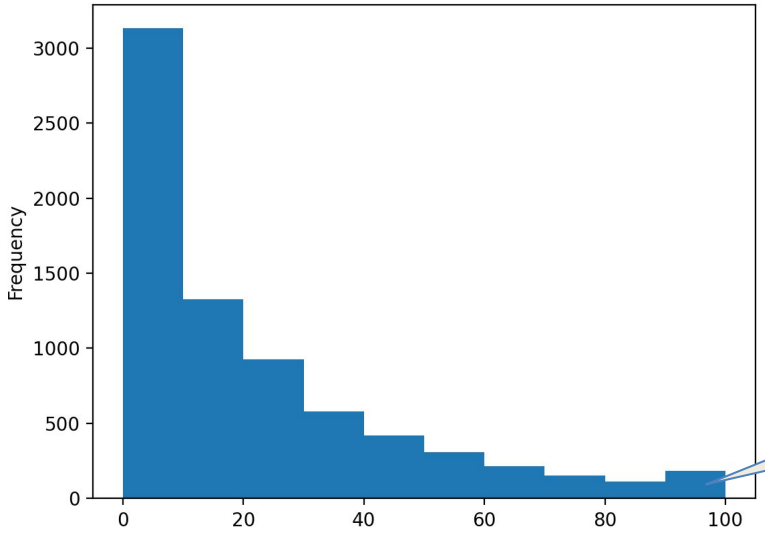
- Диаграмма демонстрирует распределение значений конкретного признака между минимальным и максимальным значениями.
- По умолчанию диапазон распределений разбит на 10 интервалов.
- kind = 'hist'.



Визуализация данных

Гистограмма размера приложений

```
df['Size'].plot(kind = 'hist')
```

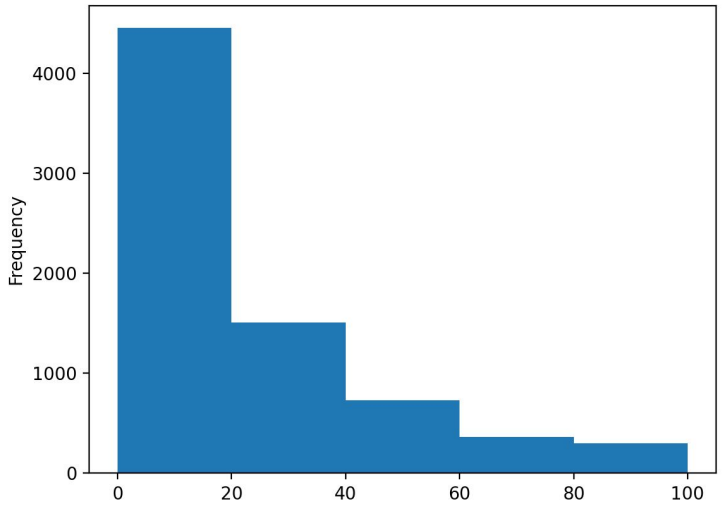


Как изменить количество столбцов?

Визуализация данных

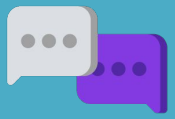
Гистограмма размера приложений

```
df['Size'].plot(kind = 'hist', bins = 5)
```



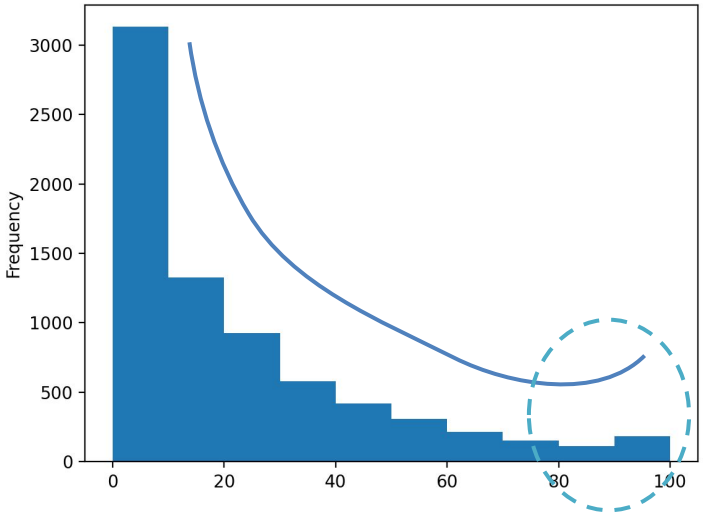
Визуализация данных

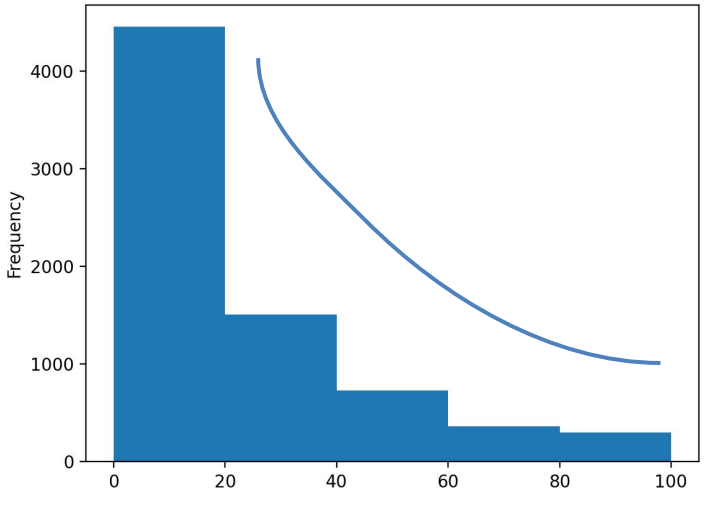
Для чего нужно изменять количество столбцов?



Визуализация данных

Изменяя количество столбцов, можно увидеть колебания значений

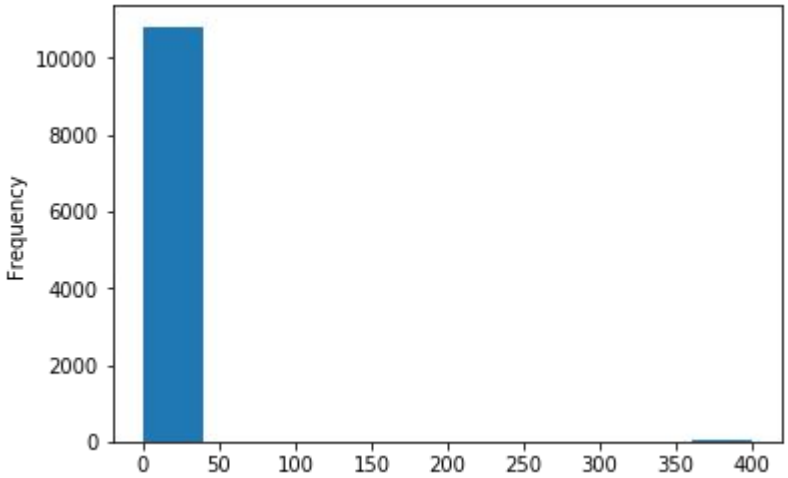




Визуализация данных

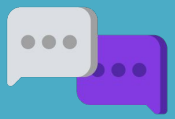
Как оценить распределение значений в этом случае?

```
df[df['Type'] == 'Paid']['Price'].plot(kind = 'hist')
```



Визуализация данных

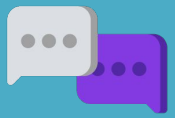
Данные, выделяющиеся среди общей выборки значений, называют **выбросами**.



Визуализация данных

«Ящик с усами»

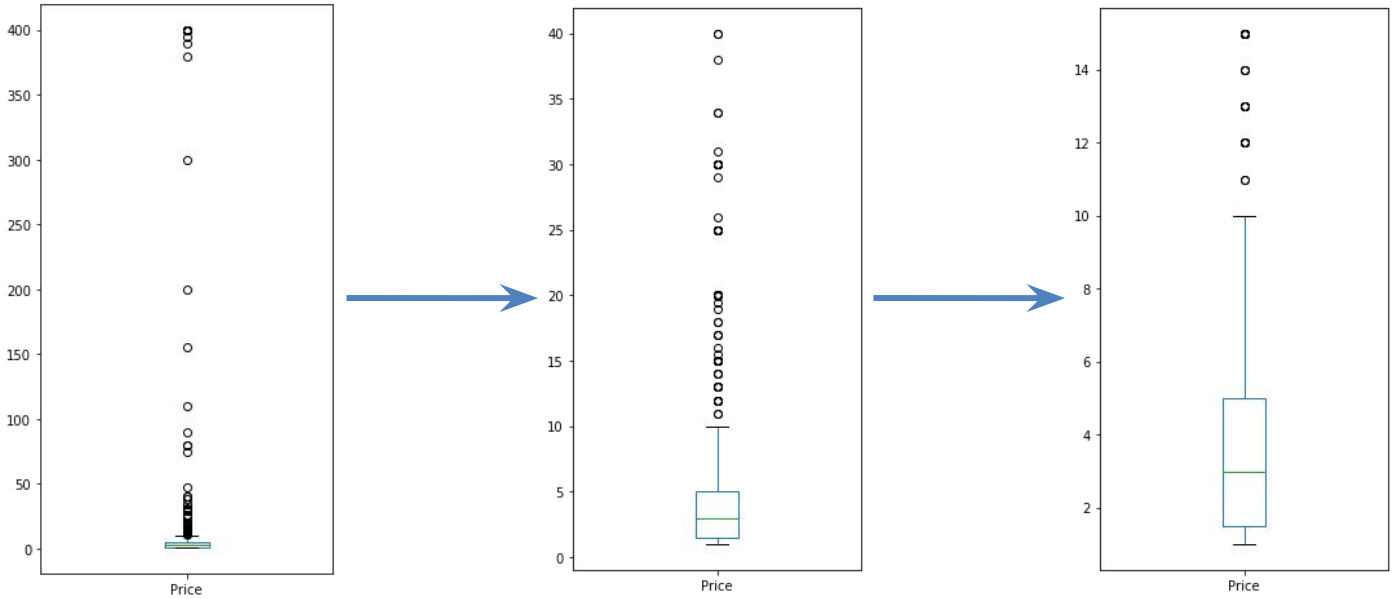
- Диаграмма, которая одновременно отображает медиану, нижний и верхний квартили, минимальное и максимальное значения и выбросы.
- kind = 'box'.



Визуализация данных

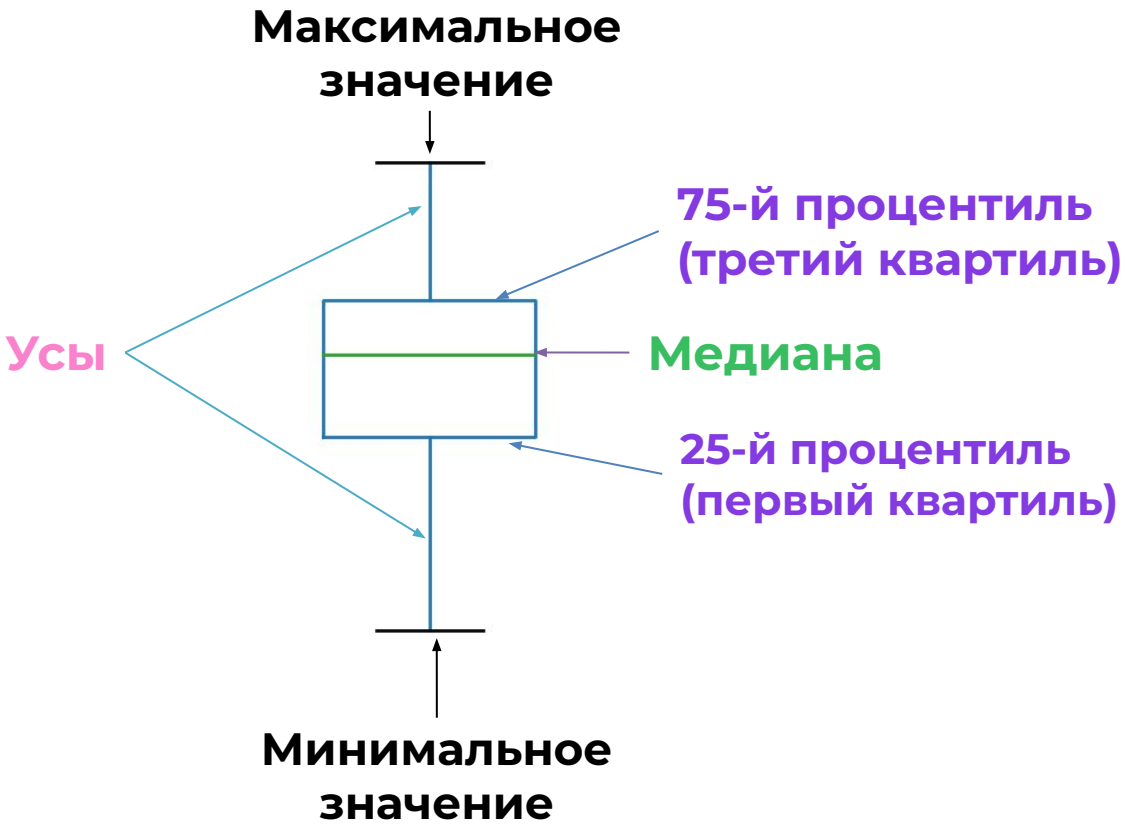
Диаграмма «Ящик с усами» для стоимости приложений

```
df[df['Type'] == 'Paid']['Price'].plot(kind = 'box')
```



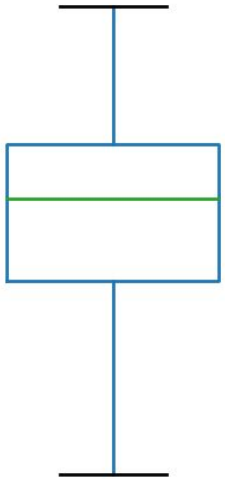
Визуализация данных

Структура диаграммы



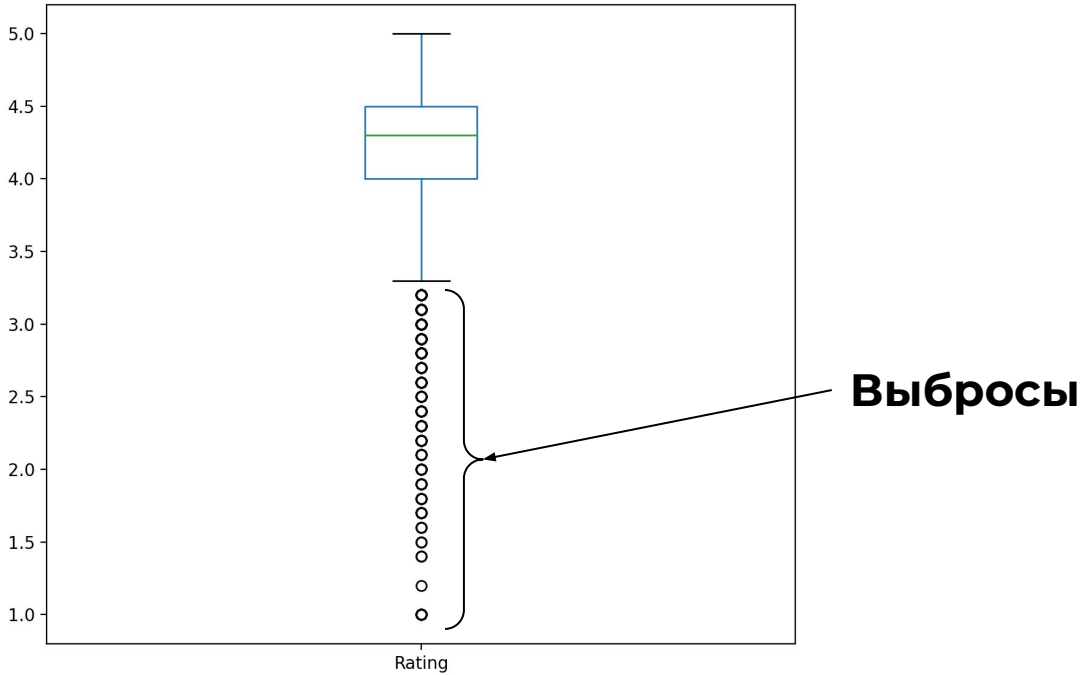
Структура диаграммы

Большая часть значений расположена в ящике.

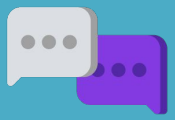


Структура диаграммы

Если значение не характерно для ящика и не попадает в него, то оно становится выбросом.



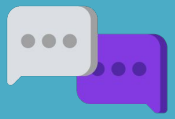
Наличие выбросов — это хорошо, плохо или нейтрально?



Визуализация данных

Опасности выбросов

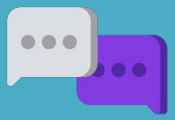
- ➔ Выброс может появиться из-за ошибки ввода данных.
- ➔ Выброс искажает результаты статистических расчётов, например, среднее арифметическое.
- ➔ Наличие выбросов говорит о неоднородности выборки и ставит под сомнение результаты анализа данных.



Визуализация данных

Что делать с выбросами?

- ➔ Проверить, были ли допущены ошибки при вводе данных.
- ➔ Если обнаружены ошибки, исправить их.



Что делать, если данные введены некорректно?

- ➔ Принять решение об удалении из набора данных строк, содержащих выбросы.
- ➔ Решение об удалении строк зависит от цели исследования и количества данных.

Кейс № 1. Мы — команда разработчиков мобильных приложений

Цель анализа данных: изучить особенности рынка, целевую аудиторию, популярные жанры, цены, частоту выпуска обновлений.

Условия работы: чтобы найти свою нишу на рынке, важно иметь данные как о дорогих, так и о бюджетных приложениях.

Вывод: строки с выбросами нужно оставить.

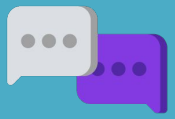
Визуализация данных

Кейс № 2. Мы хотим разместить рекламу своих товаров в приложениях

Цель анализа данных: выбрать подходящие приложения для рекламной интеграции.

Условия работы: наши потенциальные потребители пользуются только бюджетными приложениями.

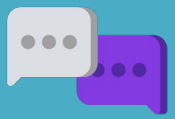
Вывод: строки с выбросами нужно удалить.



Визуализация данных

При выполнении анализа мы изучаем связи между данными.

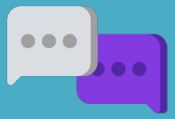
Как визуально отразить связь между двумя показателями?



Визуализация данных

Диаграмма рассеяния

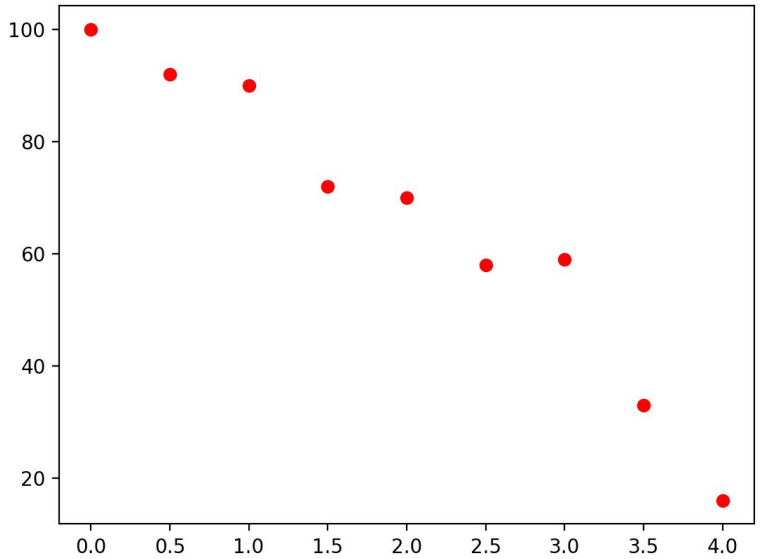
- Диаграмма показывает степень связи между переменными.
- kind = 'scatter'.



Визуализация данных

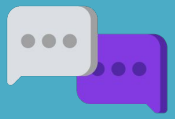
Пример диаграммы рассеяния

Как построить диаграмму этого вида?



Построение диаграммы рассеяния

1. Выбрать два признака, связь между которыми мы хотим визуализировать.
2. Отметить на координатной плоскости точки, координаты которых — это значения выбранных признаков.



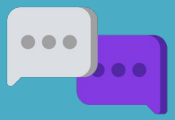
Визуализация данных

Построение диаграммы рассеивания

3. Оценить расположение точек на графике: разбросаны ли они равномерно по координатной плоскости, или собраны вокруг воображаемой линии?
4. Если точки собраны вокруг воображаемой линии, между величинами есть связь, в противном случае — связи нет.

Визуализация данных

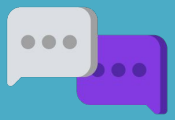
Приведите пример гипотезы, которую можно проверить при помощи диаграммы рассеяния.



Визуализация данных

Пример гипотезы

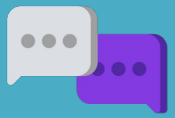
Дорогие приложения имеют меньшее количество установок по сравнению с дешёвыми или бесплатными.



Визуализация данных

Диаграмма рассеяния

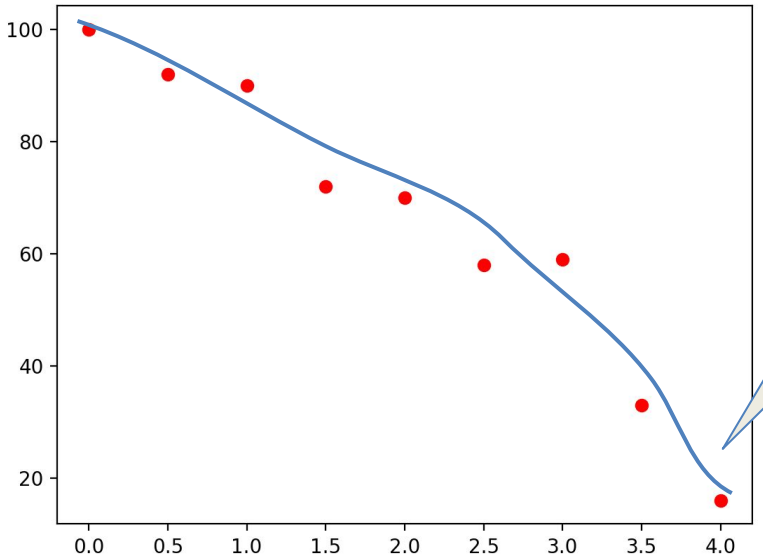
```
df.plot(x = '<имя первого столбца>',
        y = '<имя второго столбца>',
        kind = 'scatter')
```



Визуализация данных

Диаграмма рассеяния

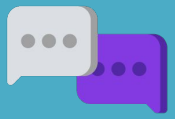
```
df.plot(x = 'Price', y = 'Installs', kind = 'scatter')
```



Взаимосвязь величин подтверждена. Гипотеза верна

Визуализация данных

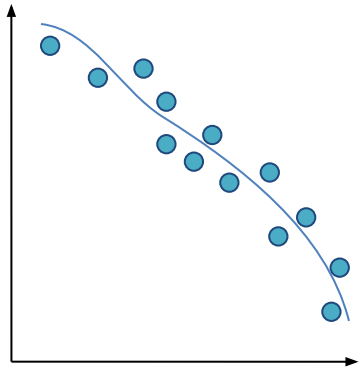
Какие связи между величинами мы можем увидеть на диаграмме?

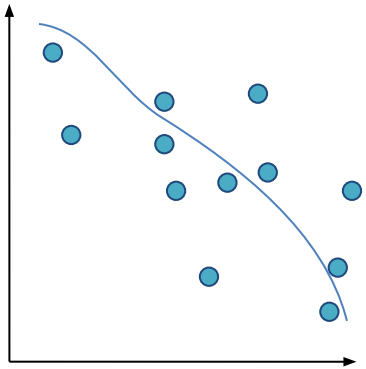


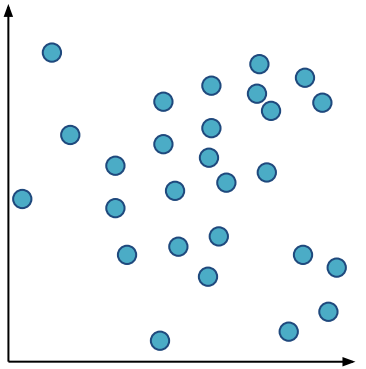
Визуализация данных

По силе

1. Сильная.
2. Слабая.
3. Отсутствует.

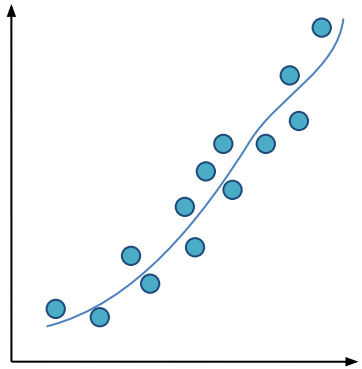


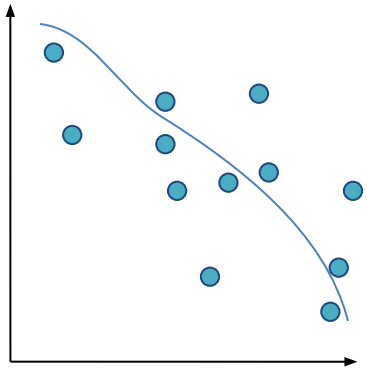


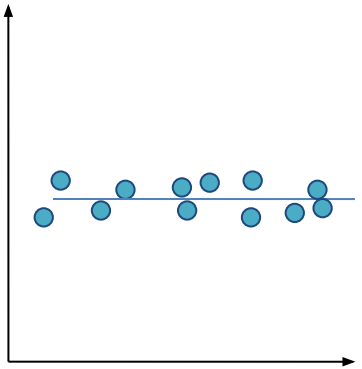


По направленности

1. Положительная.
2. Отрицательная.
3. Нулевая.

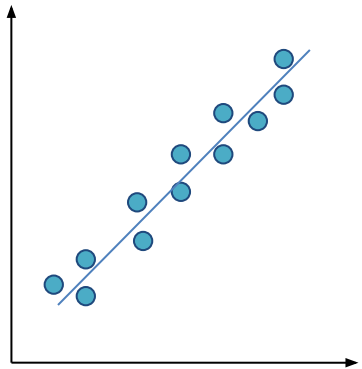


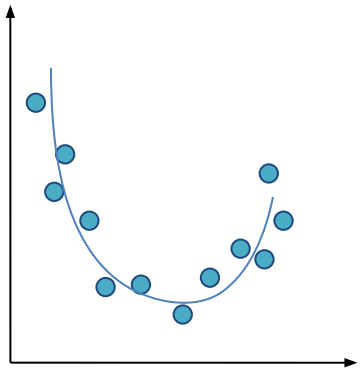




По виду графика

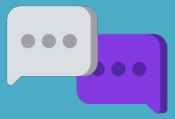
1. Линейная.
2. Нелинейная.





Круговая диаграмма

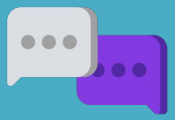
- Диаграмма демонстрирует распределение значений по категориям. Наглядно отображает пропорции и доли.
- kind = 'pie'.



Визуализация данных

Предположим, в DataFrame есть столбец с названием сезона, во время которого было произведено последнее обновление.

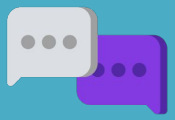
Как построить круговую диаграмму для сезонов, если информация в этом столбце не количественная?



Визуализация данных

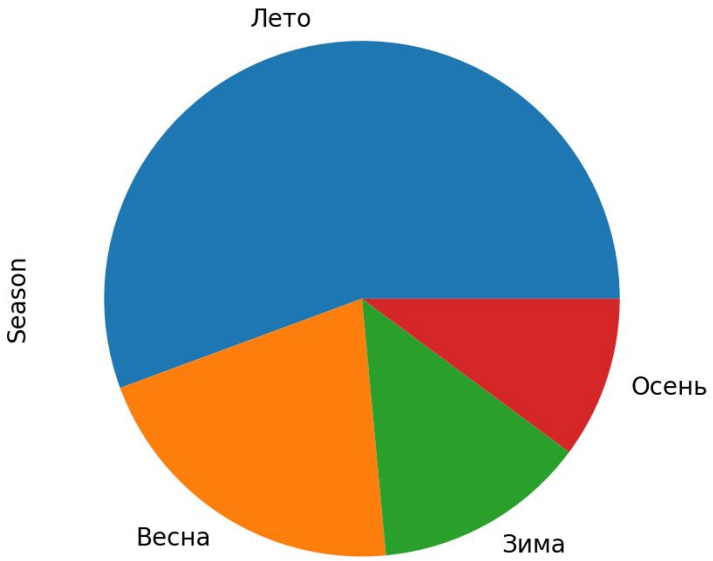
Шаг 1. Посчитать количество уникальных значений столбца при помощи метода value_counts().

Шаг 2. Построить круговую диаграмму на основе полученной Series.

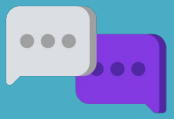


Круговая диаграмма

```
df['Season'].value_counts().plot(kind = 'pie')
```

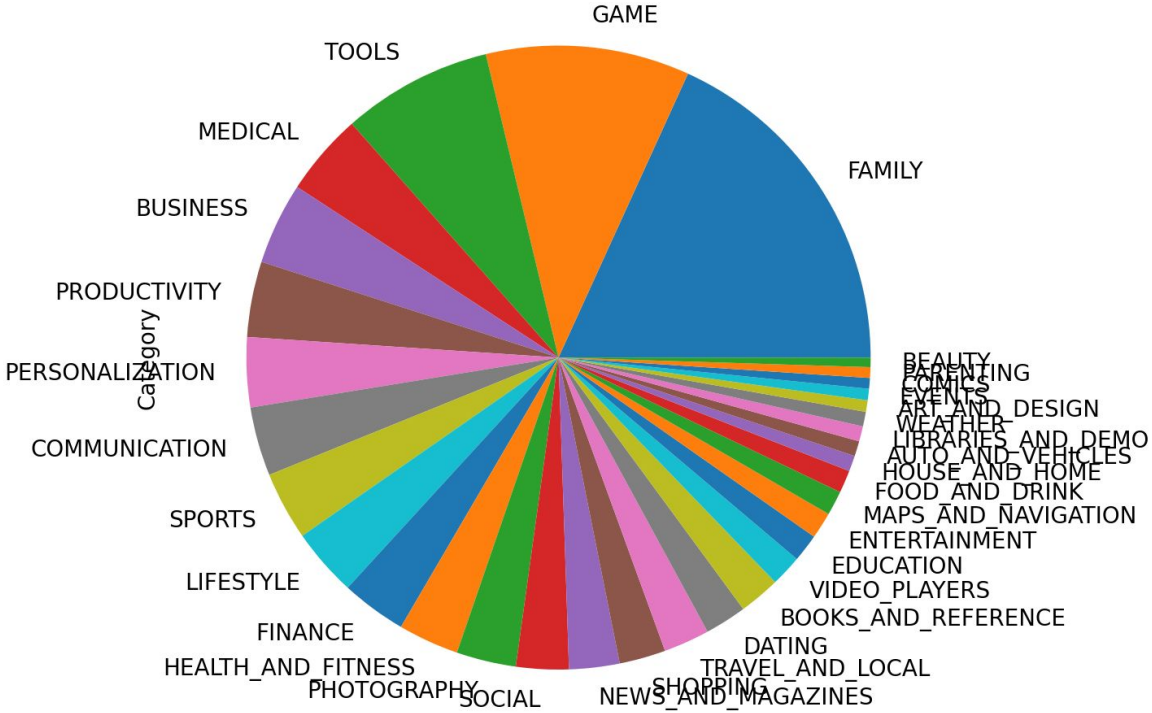


Какое количество долей диаграммы удобно для восприятия?



Визуализация данных

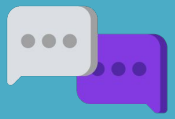
Выбирайте 5–6 категорий, в противном случае диаграмму сложно читать



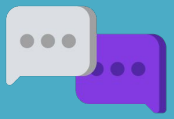
Визуализация данных

Столбчатая диаграмма

- Диаграмма помогает сравнивать значения друг с другом.
- kind = 'bar'. Столбцы расположены вертикально.
- kind = 'barh'. Столбцы расположены горизонтально.



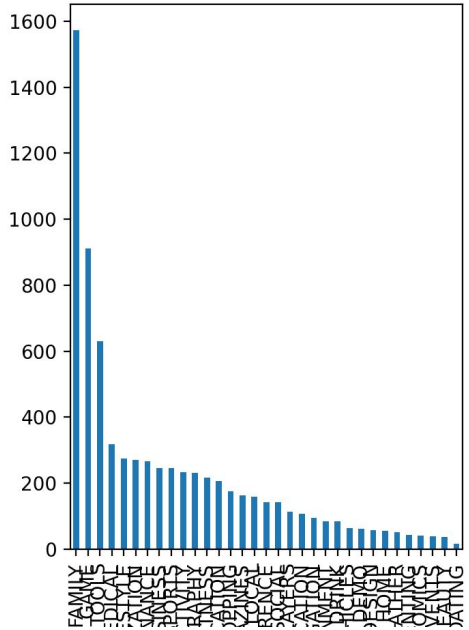
Сравним количество приложений в разных категориях при помощи вертикальной столбчатой диаграммы.



Визуализация данных

Столбчатая диаграмма

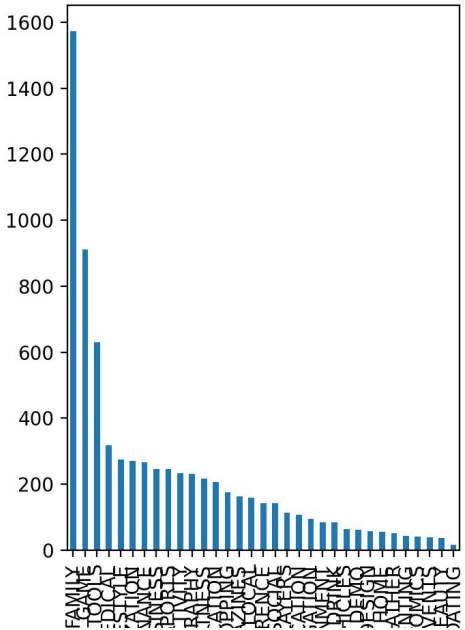
```
df['Category'].value_counts().plot(kind = 'bar')
```



Перечислите минусы этой столбчатой диаграммы

Столбчатая диаграмма

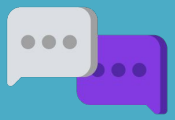
Названия категорий написаны частично и накладываются друг на друга.



Визуализация данных

Изменим тип диаграммы на горизонтальную столбчатую.

Добавим отступы между столбцами диаграммы и подписями.

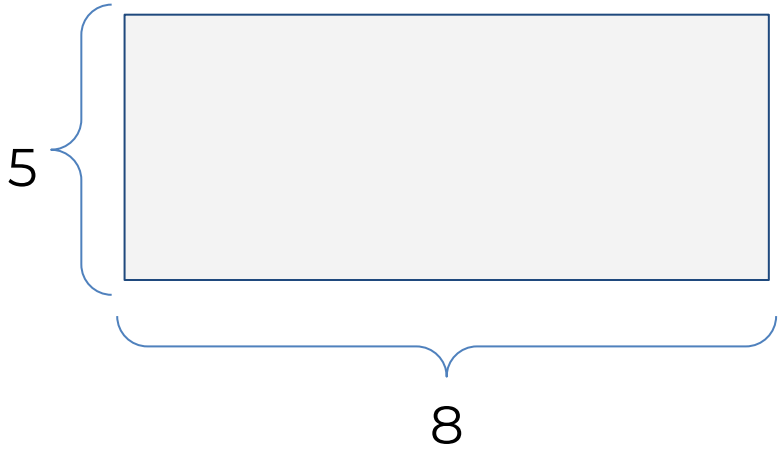


Визуализация данных

Параметр figsize

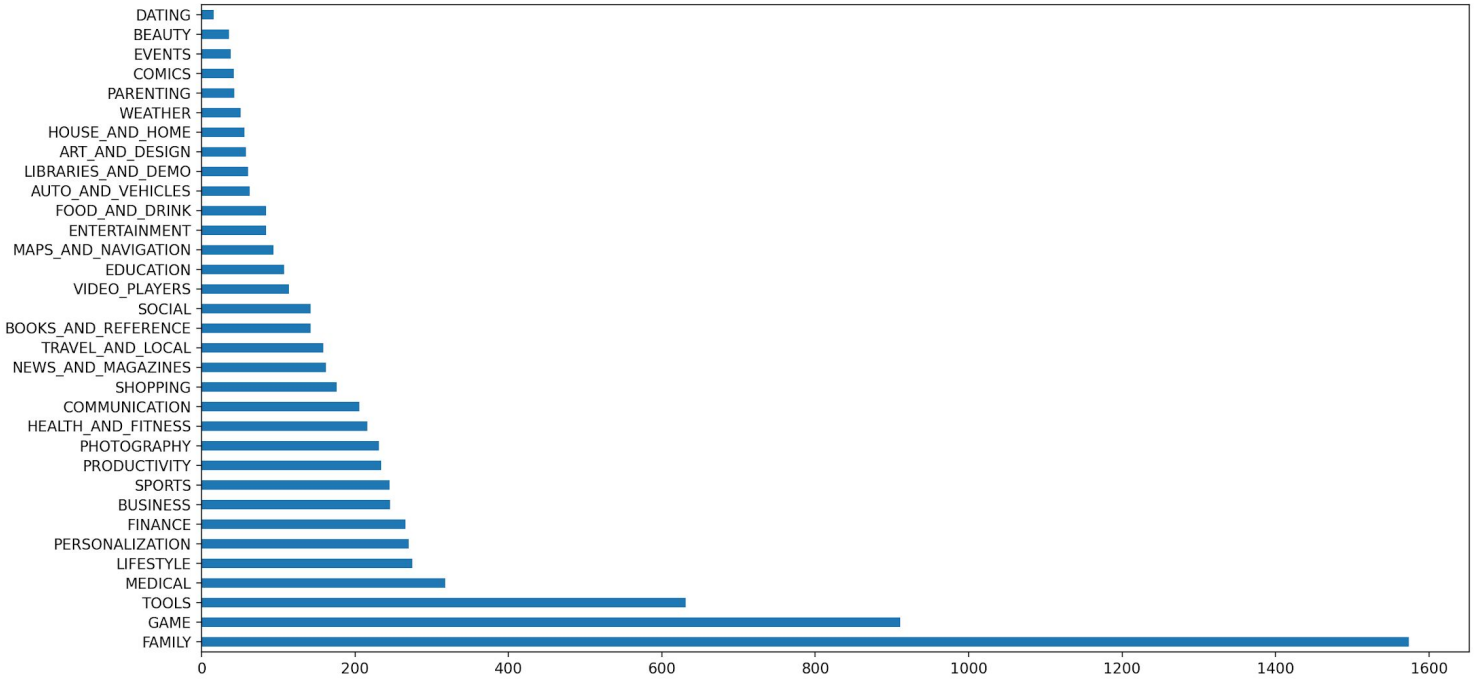
- Позволяет задать ширину и высоту диаграммы.
- Если параметр не указан, то размер по умолчанию (6.4, 4.8).

```
figsize = (8, 5)
```



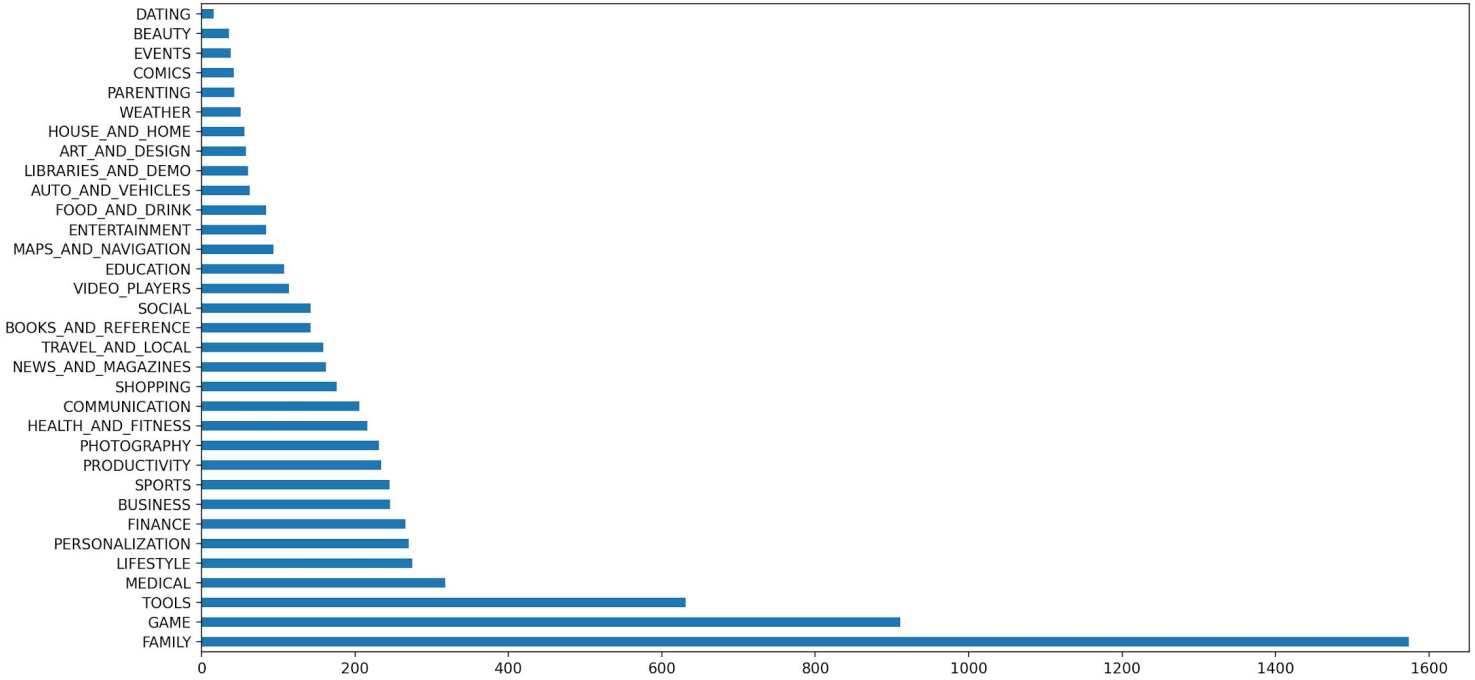
Столбчатая диаграмма

```
df['Category'].value_counts().plot(kind = 'barh', figsize = (8, 5))
```



Визуализация данных

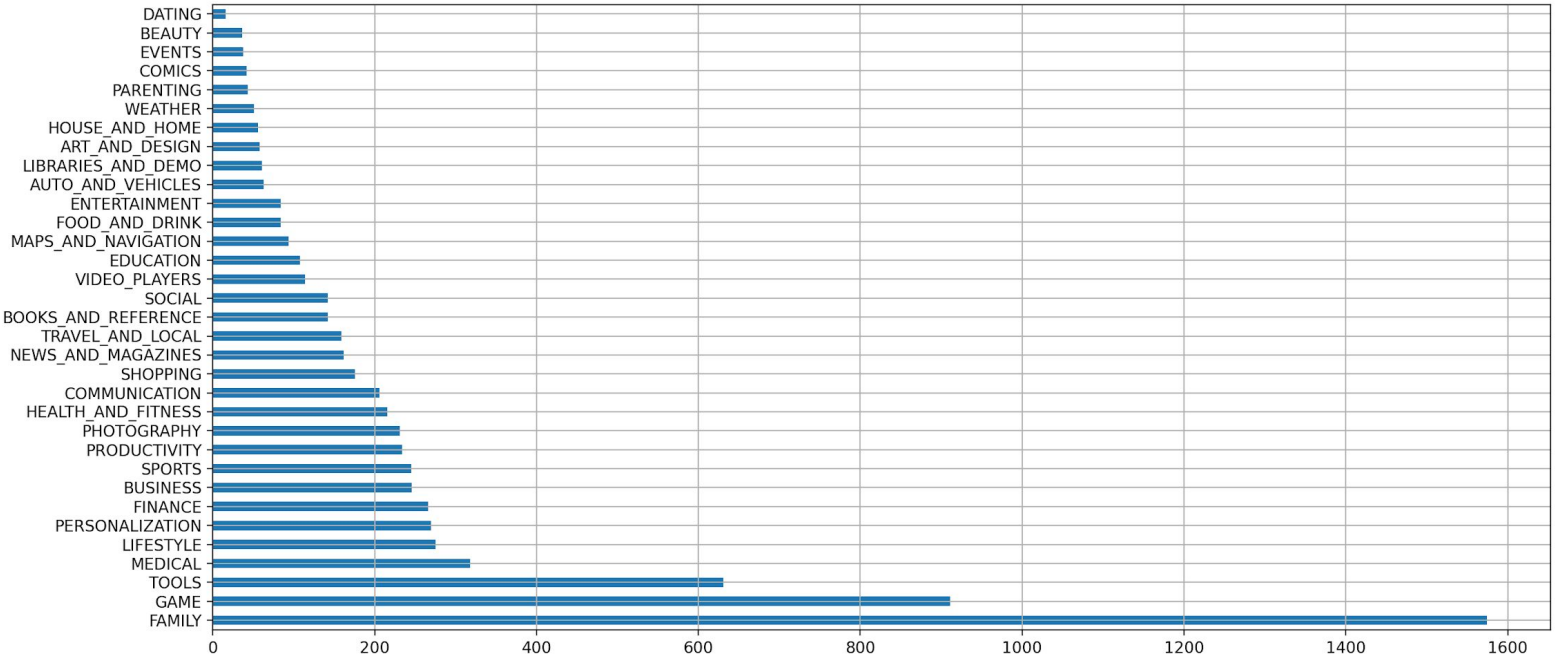
Чего не хватает на диаграмме, чтобы было удобно определять значения категорий?



Визуализация данных

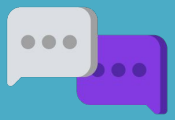
Координатная сетка

```
df['Category'].value_counts().plot(kind = 'barh', figsize = (10, 10), grid = True)
```



Визуализация данных

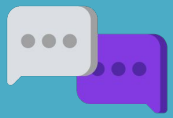
Как можно сравнить, различается ли среднее количество установок в различных целевых аудиториях для платных и бесплатных приложений?



Визуализация данных

Сначала готовим данные, потом строим диаграммы

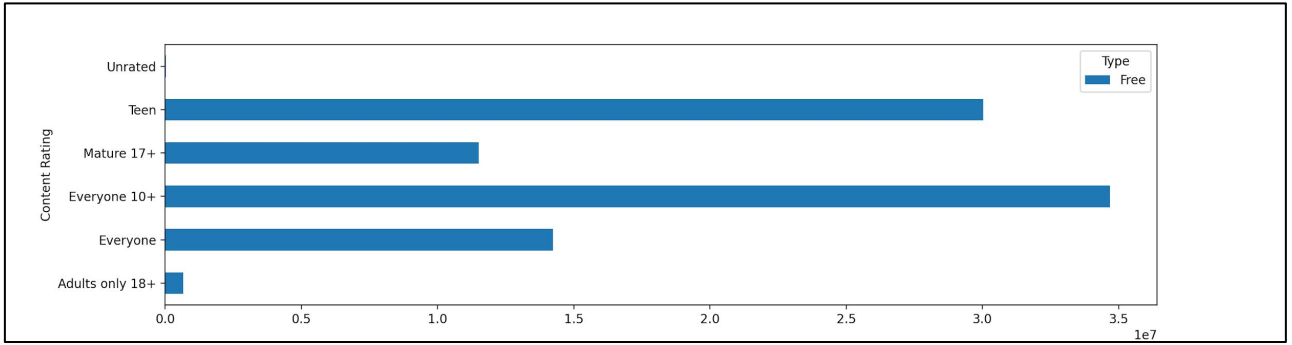
```
d1 = df[df['Type'] == 'Free'].pivot_table(index = 'Content Rating',
                                          columns = 'Type',
                                          values = 'Installs',
                                          aggfunc = 'mean')
d2 = df[df['Type'] == 'Paid'].pivot_table(index = 'Content Rating',
                                          columns = 'Type',
                                          values = 'Installs',
                                          aggfunc = 'mean')

d1.plot(kind = 'barh')
d2.plot(kind = 'barh')
```

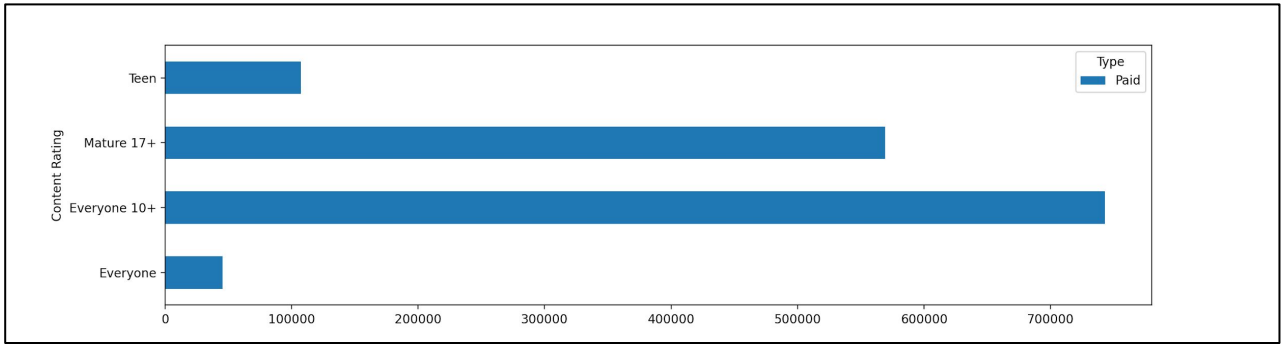


Визуализация данных

Две отдельные диаграммы

Удобно ли сравнивать значения?

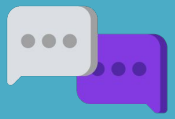




Визуализация данных

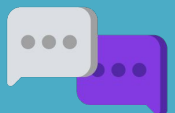
Минусы использования двух отдельных диаграмм

- Одинаковый цвет столбцов.
- Различная цена деления по оси абсцисс.
- Для платных приложений указаны не все категории целевой аудитории.



Сначала готовим данные, потом строим одну диаграмму

```
d = df.pivot_table(index = 'Content Rating',
                   columns = 'Type',
                   values = 'Installs',
                   aggfunc = 'mean')
```

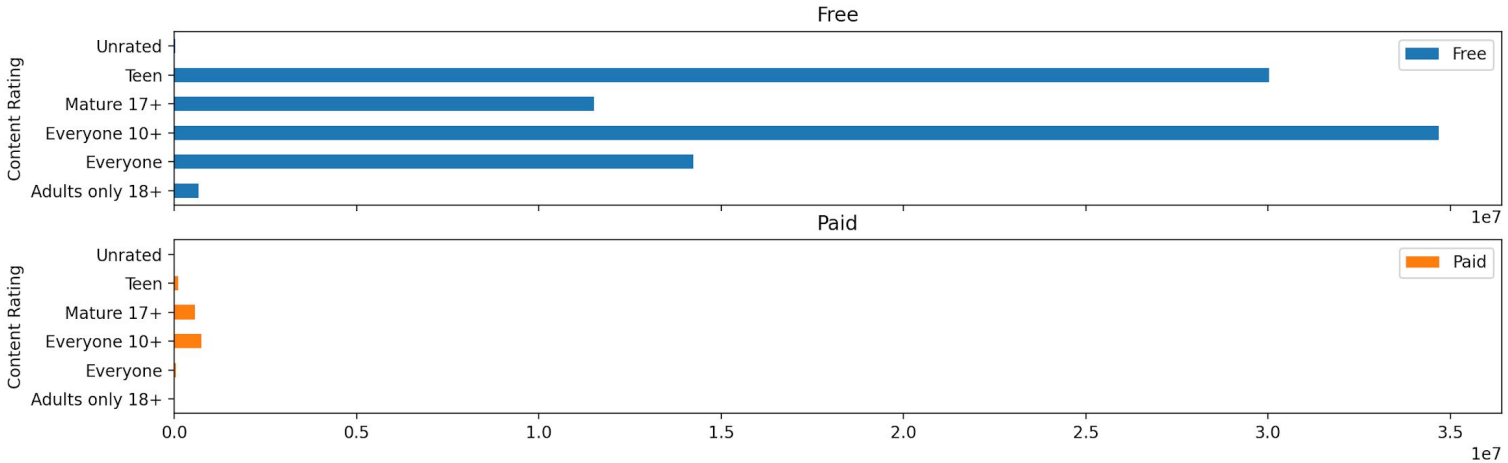


Визуализация данных

Две диаграммы в одной координатной сетке

```
d.plot(kind = 'barh', subplots = True)
```

Позволяет строить два графика друг под другом



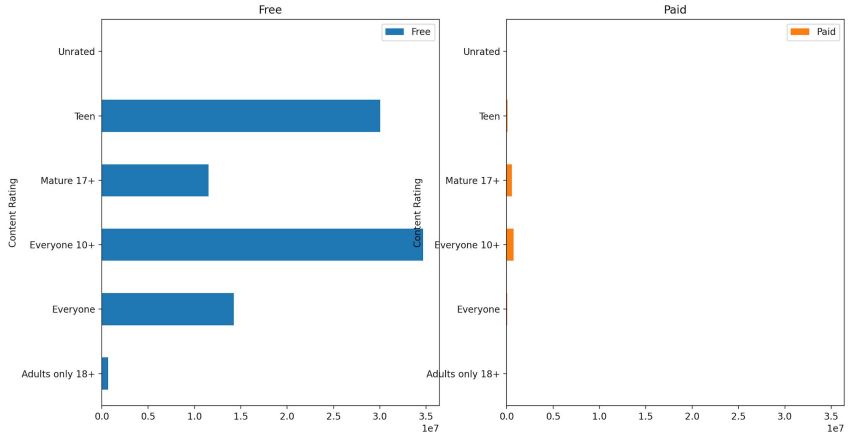
Визуализация данных

Две диаграммы в одной координатной сетке

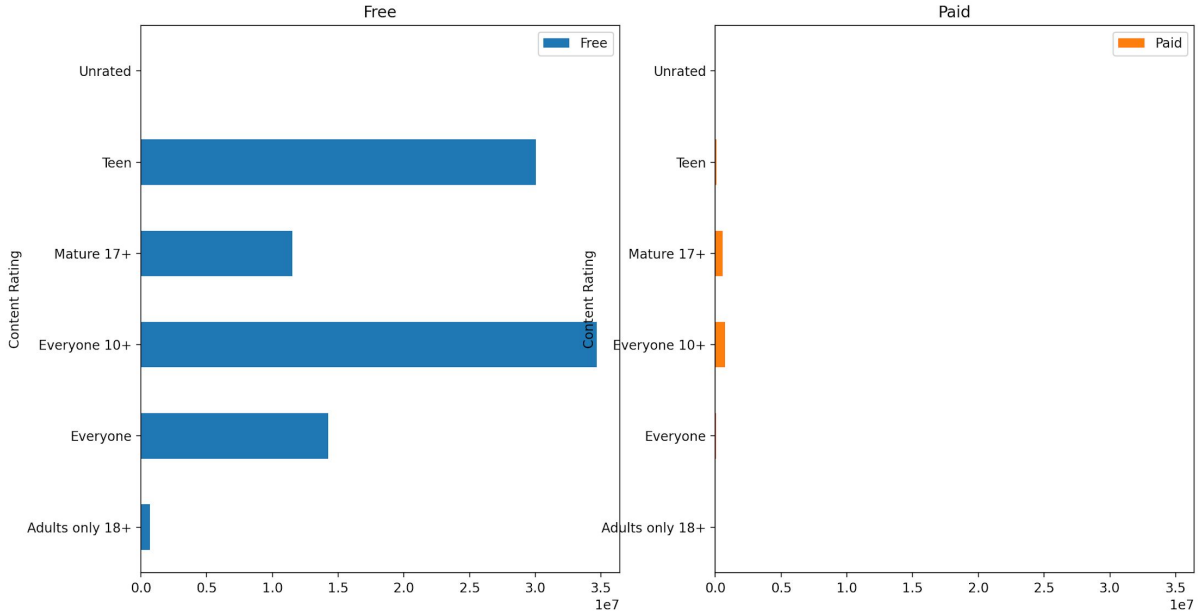
Параметр layout позволяет задать расположение графиков.

- ➔ layout = (1, 2) — два в одну линию.
- ➔ layout = (2, 1) — в две линии по одному.

```
d.plot(kind = 'barh', subplots = True, layout = (1, 2))
```



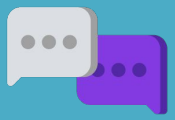
Удобно ли сравнивать значения?



Визуализация данных

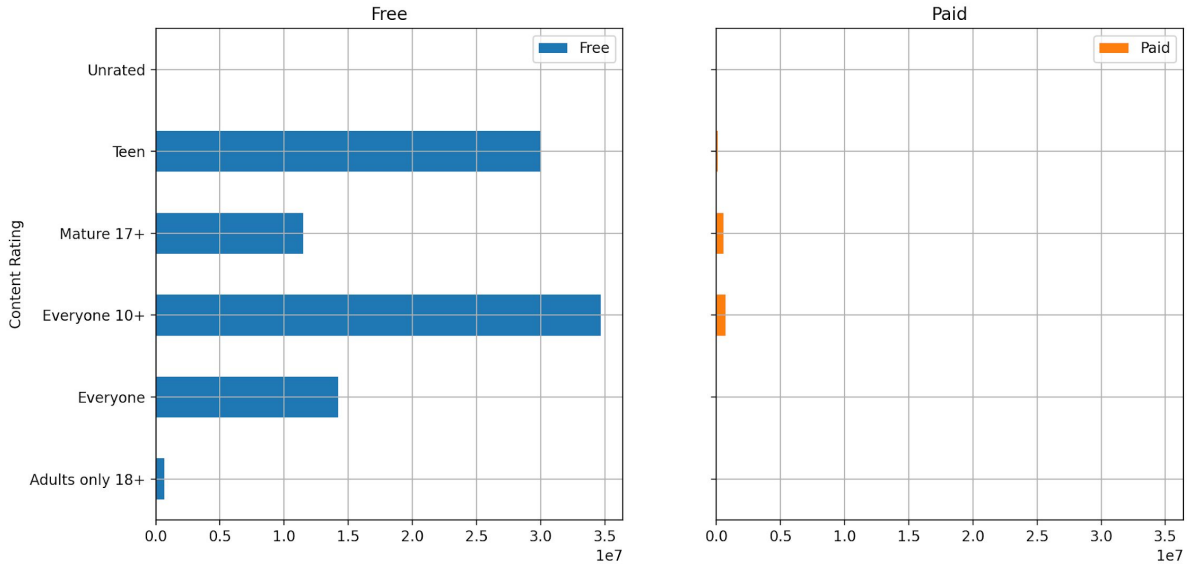
Минусы диаграммы

- Подписи значений по оси ординат второй диаграммы накладываются на первую.
- Нет координатной сетки, чтобы точно определить значения.



Параметр sharey позволяет не дублировать надписи по оси ординат

```
d.plot(kind = 'barh', subplots = True, layout = (1, 2), sharey = True, grid = True)
```



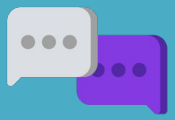
Визуализация данных

Возможно ли совместить эти два графика?

Визуализация данных

Совмещённая столбчатая диаграмма

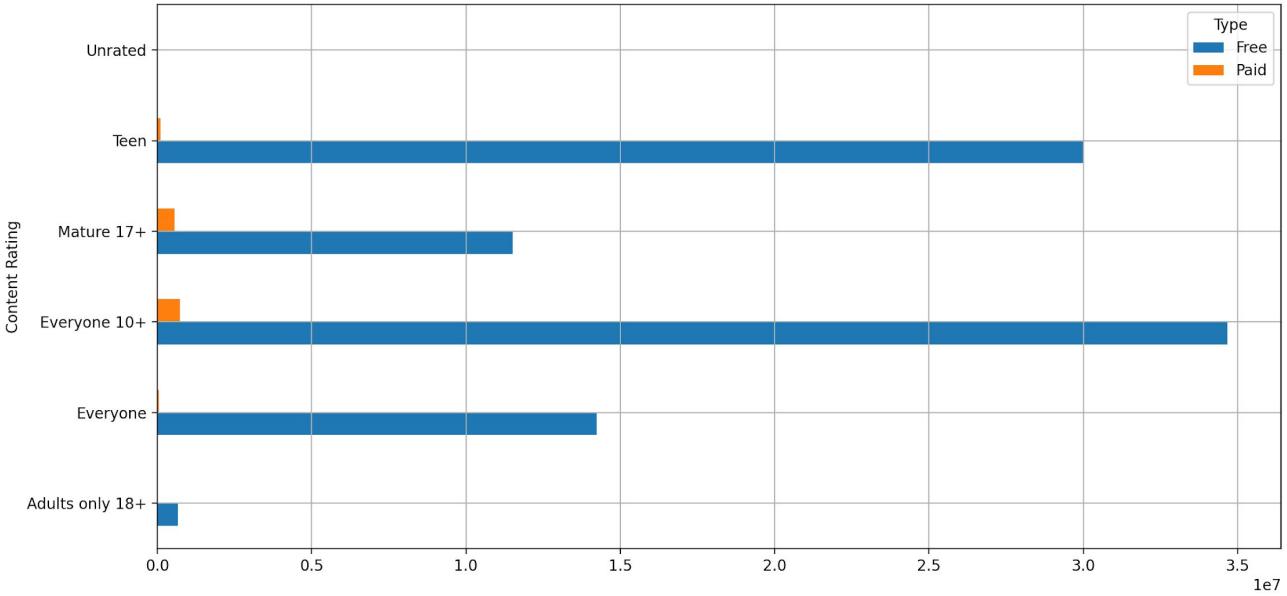
- Диаграмма отражает отношение значений двух и более параметров.



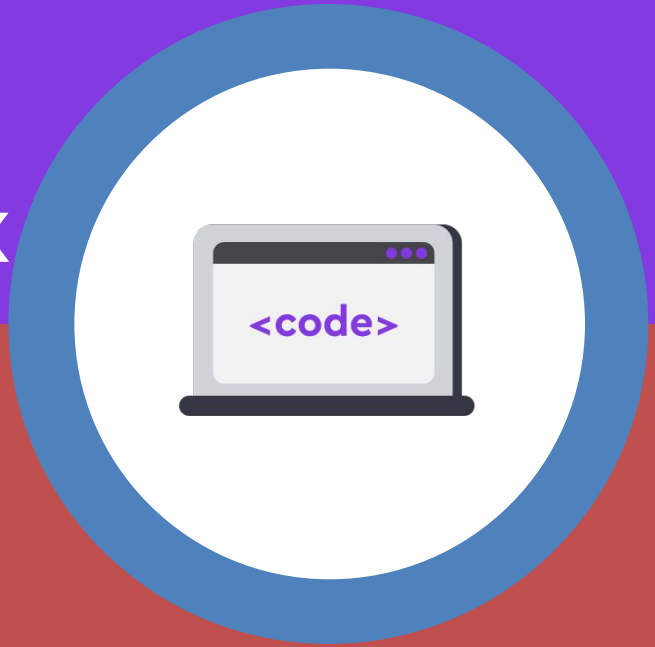
Визуализация данных

Совмещённая столбчатая диаграмма

```
d.plot(kind = 'barh', grid = True)
```



Работа на платформе: Визуализация данных



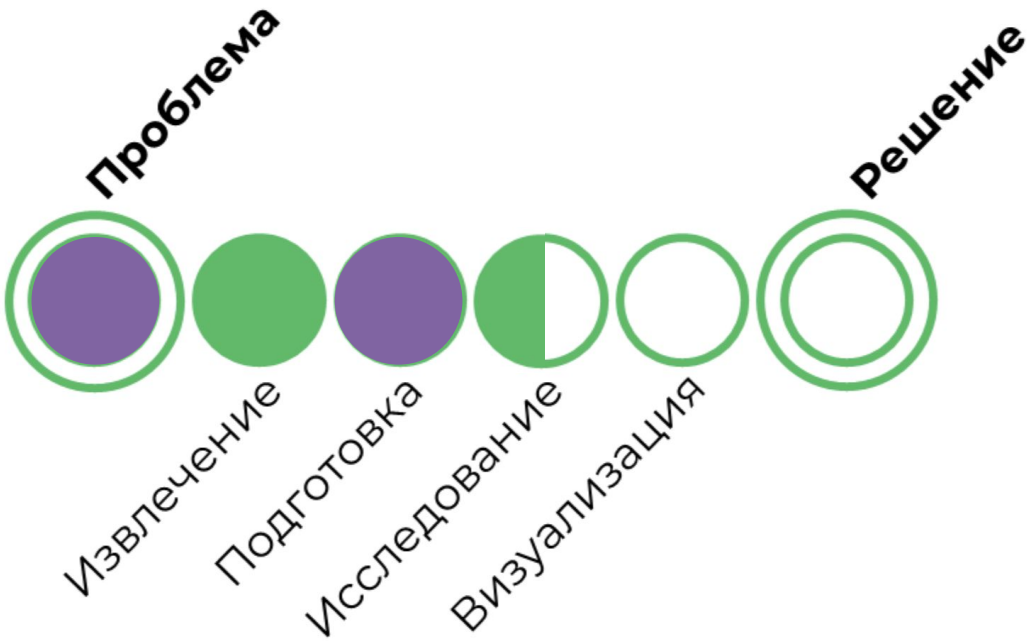
Выполните практическое задание

«Визуализация данных»

Перерыв

Обсуждение:
Работа над индивидуальным проектом

Этапы процесса анализа данных, которые уже выполнены



Обсуждение

Что необходимо сделать сегодня?

Обсуждение

Графики и диаграммы строятся на основе данных, полученных на этапе «Исследование».

Обсуждение

Чек-лист

- Закончить этап «Исследование».
- Выполнить этап «Визуализация».

Результаты, полученные в ходе выполнения этих двух этапов, **на следующем занятии необходимо будет оформить в виде презентации**.

Обсуждение

Не забывайте проводить ревизию чек-листа и ментальной карты, которые вы создали на прошлом занятии.

Обсуждение

Важно!

- В VSC не установлена библиотека Matplotlib.
- Её необходимо будет установить самостоятельно.
- На платформе есть задание с инструкцией.

Обсуждение

Чек-лист

- [ ] Установить библиотеку Matplotlib в среду разработки VSC.
- [ ] Закончить этап «Исследование».
- [ ] Выполнить этап «Визуализация».

Обсуждение

Модуль 3. Урок 5. Визуализация данных

VSC: Работа над индивидуальным проектом

Выполните практическое задание

«VSC: Индивидуальный проект по теме “Анализ данных”»

Завершение урока

Поделитесь мнением

1 мин
для каждого

- Как ты оцениваешь работу, проведённую сегодня?
- Завершил ли ты этап исследования?
- Гипотезы, выдвинутые на прошлом занятии, подверглись изменениям?
- Диаграммы дополняют твоё исследование?
- Что ты планируешь делать дальше?

Завершение урока