

Matplotlib: Визуализация данных для Data Analysis



Зачем нужна визуализация данных?

График стоит тысячи строк кода – он позволяет почти мгновенно увидеть то, что невозможно заметить в таблице.

🔍 Обнаружение паттернов

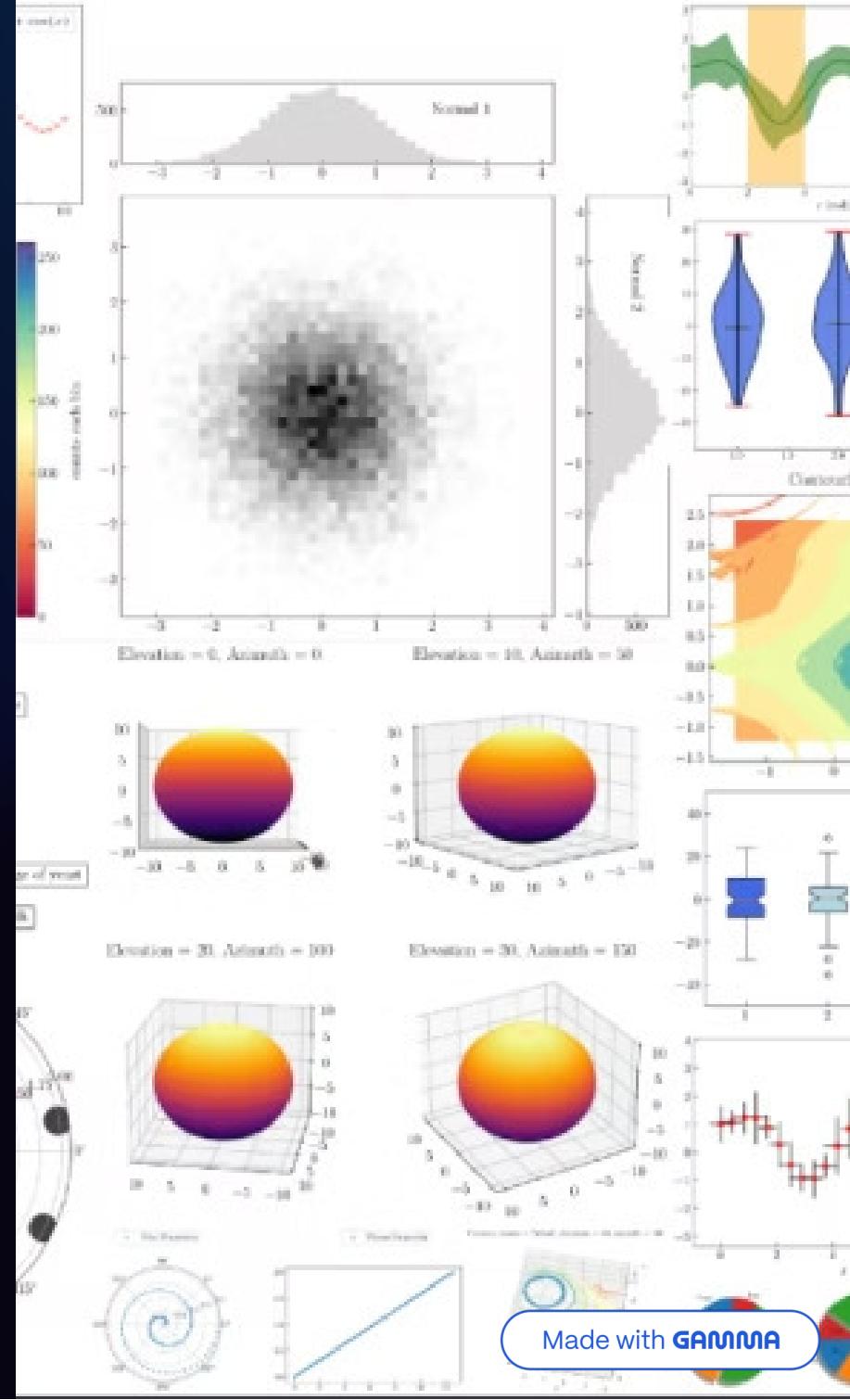
Видеть тренды, аномалии и выбросы, которые скрыты в табличных данных.

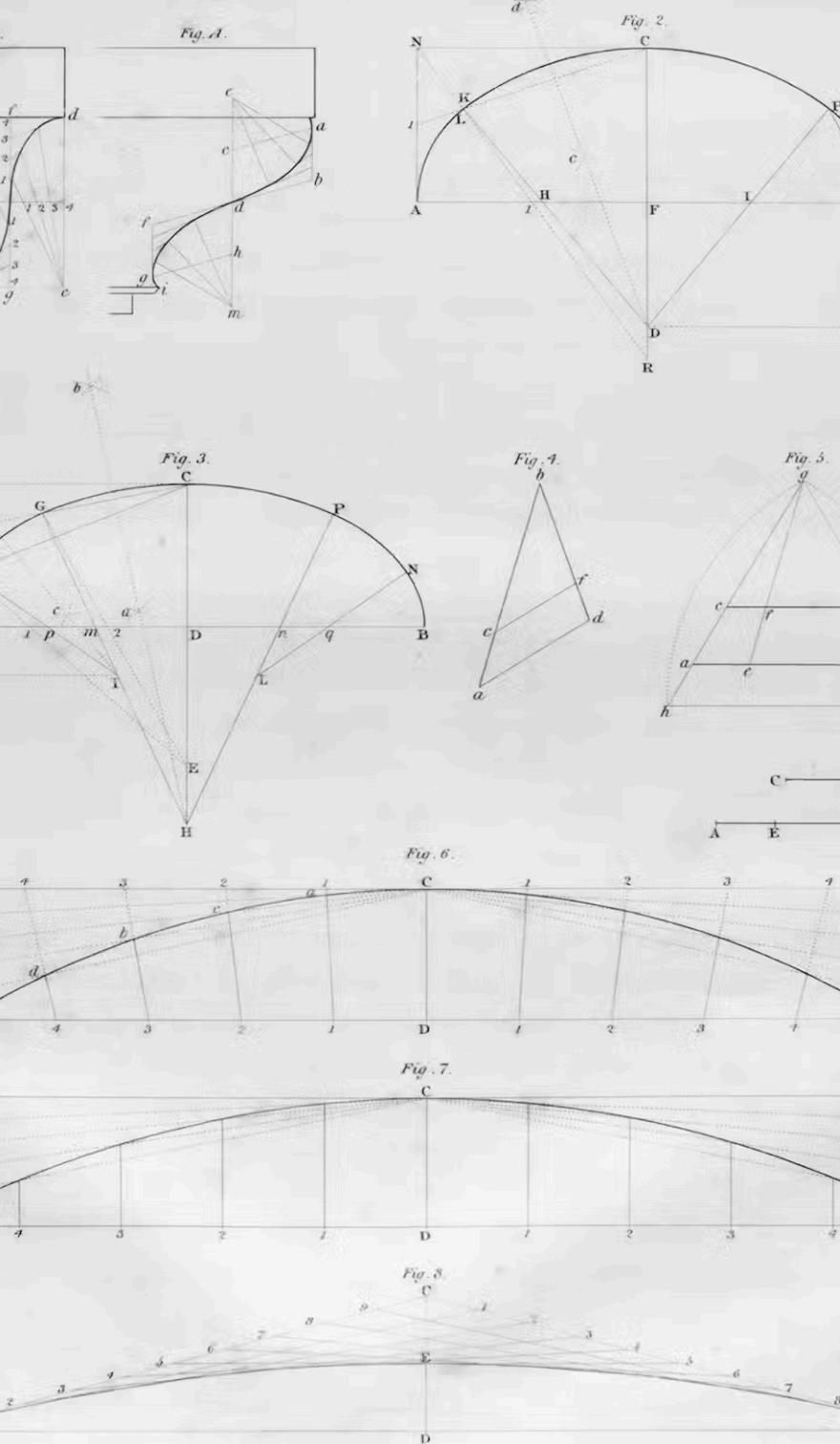
💬 Коммуникация

Эффективно доносить инсайты до коллег и заказчиков с помощью ясной визуализации.

✓ Проверка гипотез

Визуально подтверждать или опровергать выводы, полученные через анализ в Pandas.





Анатомия Matplotlib

Понимание базовых компонентов – первый шаг к созданию эффективных графиков.

Figure (Холст)

Весь контейнер, в котором находятся все элементы графика. Это как пустой лист бумаги, на котором вы будете рисовать.

Axes (Область построения)

Фактическая область, где отображаются ваши данные. Это важный объект, с которым вы будете работать каждый день.

pyplot (plt)

Модуль для быстрого построения графиков.
Используйте: `import matplotlib.pyplot as plt`

Линейные графики (plot)

Когда использовать

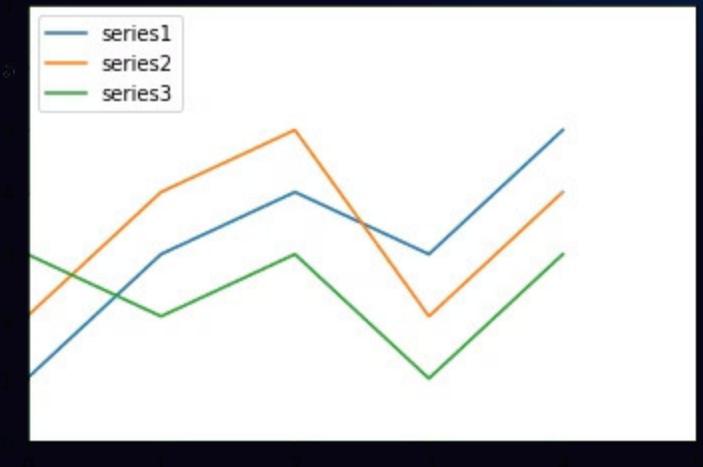
Линейные графики идеальны для отображения трендов во времени или выявления взаимосвязи между двумя непрерывными переменными. Они помогают увидеть направление изменения данных.

Пример кода

```
plt.plot(x_data, y_data)  
plt.xlabel('Время')  
plt.ylabel('Значение')  
plt.title('Тренд во времени')  
plt.show()
```

Практическое применение

Визуализация пассажиропотока по часам дня помогает выявить пиковые периоды и оптимизировать расписание.



Столбчатые диаграммы (bar)

Когда использовать

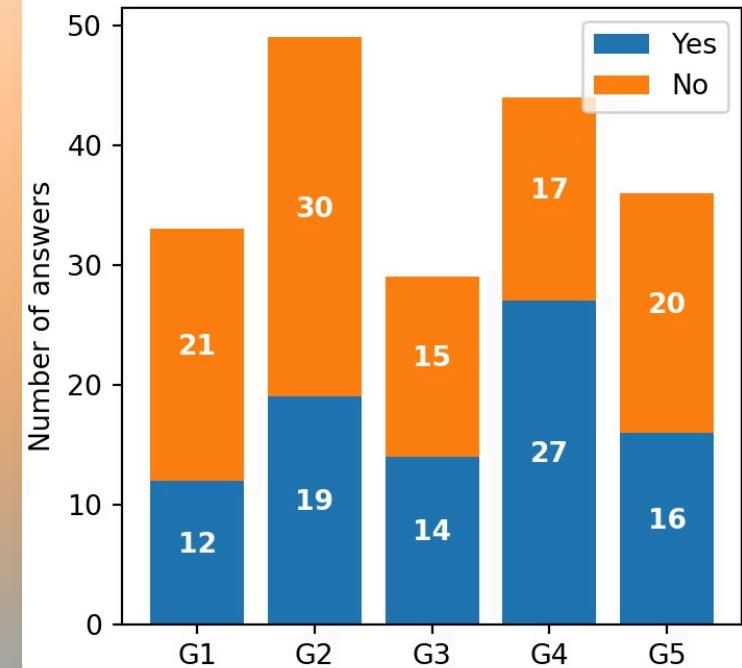
Столбчатые диаграммы превосходны для сравнения категориальных данных. Они позволяют быстро сравнить значения между разными категориями и выделить наибольшее и наименьшее значения.

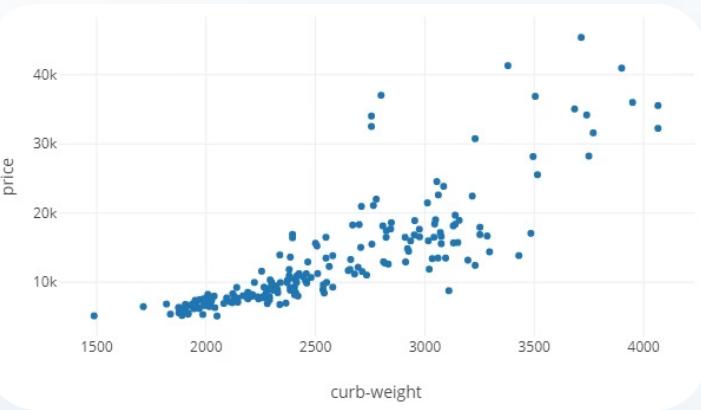
Пример кода

```
plt.bar(категории, значения)  
plt.xlabel('Категория')  
plt.ylabel('Количество')  
plt.title('Сравнение по категориям')  
plt.show()
```

Практическое применение

Сравнение потока пассажиров между прямым и обратным направлениями или анализ способов оплаты.





Диаграммы рассеяния (scatter)

Когда использовать

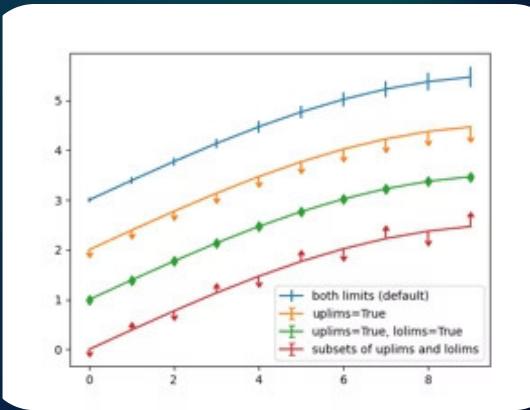
Диаграммы рассеяния позволяют исследовать корреляцию между двумя переменными. Каждая точка представляет одно наблюдение, что помогает выявить закономерности и выбросы.

Пример кода

```
plt.scatter(x, y, alpha=0.5)
plt.xlabel('Переменная X')
plt.ylabel('Переменная Y')
plt.title('Взаимосвязь переменных')
plt.show()
```

Практическое применение

Построение зависимости суммы платежа от времени суток выявляет закономерности поведения клиентов.



Кастомизация I: Добавление контекста

Хороший график всегда имеет четкие подписи. Контекст делает ваш график самодостаточным и легко интерпретируемым для любого зрителя.

01

`plt.title()`

Ясный заголовок, описывающий суть графика. Помогает зрителю сразу понять, что он видит.

02

`plt.xlabel()` и `plt.ylabel()`

Подписи осей должны включать единицы измерения. Это избегает путаницы и облегчает интерпретацию.

03

`plt.legend()`

Легенда необходима, если на графике несколько линий или серий данных. Она идентифицирует каждый элемент.

Кастомизация II: Стили и визуальный дизайн

Визуальная привлекательность и читаемость – ключевые элементы профессионального графика.
Правильные стили помогают выделить главное и облегчить восприятие.



figsize=(width, height)

Контролируйте размер холста функцией plt.figure(). Большой размер улучшает читаемость.



color, linestyle, marker

Выбирайте цвета разумно, используйте пунктир для различия линий, добавляйте маркеры для выделения точек.



plt.grid(True)

Сетка значительно облегчает оценку точных значений на графике, особенно для новичков.

Интеграция Matplotlib и Pandas

Pandas использует Matplotlib "под капотом" и предоставляет удобный метод `.plot()`, экономя ваше время и строки кода.

Агрегация в Pandas

Используйте `df.groupby().sum()` для подготовки данных. Это основа для красивой визуализации.

Прямое построение графика

Вызовите `df_aggregated.plot(kind='bar')` или `df_aggregated.plot(kind='line')` напрямую из DataFrame.

Экономия времени

Не нужно вручную передавать данные в `plt.bar()` или `plt.plot()`. Pandas справляется с этим автоматически.