

## Библиотеки для визуализации данных

### Топ-10 лучших библиотек визуализации данных Python



#### 1. Matplotlib

На Github более 461 тыс. пользователей, [Matplotlib](#) является самой популярной и часто считающейся лучшей библиотекой визуализации данных Python, используемой специалистами по обработке данных для создания расширенных визуализаций данных. Это будет ваша основная библиотека визуализации данных, которую необходимо освоить при работе над Data science с языком программирования Python. Он также хорошо работает с другими популярными [библиотеками Python data science](#), такими как NumPy, sklearn и pandas. Matplotlib - это пакет для построения графиков на Python, который позволяет создавать статические, интерактивные и динамические представления. Поскольку matplotlib была первой библиотекой для **визуализации данных Python**, в ходе исследований на ее основе было создано или развито несколько других библиотек.



matplotlib, как пакет для построения 2D-графиков на Python, отличается невероятным уровнем гибкости. С помощью matplotlib можно создавать интерактивные 2D-графики, такие как линейные графики, точечные графики, гистограммы и т.д. Хотя matplotlib не включает встроенную поддержку 3D-графиков, он включает альтернативный инструментарий - mplot3d, – который можно использовать для создания 3D-графиков.

#### Ключевые особенности Matplotlib

- Если вы привыкли работать с MATLAB, интерфейс Pyplot Matplotlib покажется вам чрезвычайно знакомым.
- В этом пакете есть несколько серверных частей для рендеринга.
- Некоторые библиотеки, такие как pandas и Seaborn, сводят к минимуму объем кода, который вам приходится писать, тем самым упрощая использование нескольких функций matplotlib.
- Поскольку он существует почти десять лет, у него большая база пользователей.

## Плюсы использования Matplotlib

- Относительно легко понять новичкам.
- Люди, которые раньше использовали Matlab или другие пакеты для построения графиков, найдут его намного проще в использовании. Поскольку он построен аналогично MATLAB, переключение между ними несложно.
- Он предлагает высококачественные фотографии и графики в различных форматах, включая png, pdf и т.д.
- Эта библиотека управляет многочисленными аспектами изображения, включая цвет, размер и т.д.

## Минусы использования Matplotlib

- Matplotlib отлично подходит для создания графиков. Однако это может быть не идеально для данных [временных рядов](#), поскольку для этого требуется импортировать все вспомогательные классы для форматирования года, месяца, недели и дня.
- Это также неудобно при работе с несколькими наборами данных, но преобразовать набор данных в длинный формат и построить по нему график несложно.
- Еще одним существенным недостатком является то, что библиотека низкоуровневая и требует дополнительного кода для создания визуализации.
- Он в значительной степени зависит от других библиотек Python, таких как [NumPy](#).

## 2. Seaborn

Seaborn - еще одна популярная платформа визуализации данных Python на базе Matplotlib с более чем [129 000 пользователей](#) на Github. Это высокоуровневый интерфейс для создания эстетически привлекательных и ценных статистических визуальных элементов, имеющих решающее значение для изучения и понимания данных.

Эта библиотека Python тесно связана со структурами данных [NumPy](#) и pandas. Компания Seaborn стремится сделать визуализацию ключевым компонентом анализа и исследования данных, а ее алгоритмы построения графиков, ориентированные на набор данных, используют фреймы данных, содержащие целые наборы данных.



Линейчатые диаграммы, круговые диаграммы, гистограммы и диаграммы рассеяния - вот некоторые графические элементы библиотеки. Seaborn также предлагает множество инструментов выбора цветовой палитры, которые могут помочь [определить тенденции в данных](#). Seaborn проводит внутреннее семантическое сопоставление и статистическую агрегацию для создания соответствующих диаграмм.

## Ключевые особенности Seaborn

- Seaborn хорошо работает со структурами данных NumPy и Pandas для визуализации одномерных и двумерных данных, построения временных рядов и т.д.
- В нем есть встроенные темы для графического оформления Matplotlib.
- Seaborn использует возможности matplotlib для создания потрясающих диаграмм всего несколькими строками кода.
- Его методы построения графиков, ориентированные на набор данных, работают с фреймами данных и векторами, которые содержат полные наборы данных.
- Seaborn поддерживает внутреннее сопоставление концепций и статистическую агрегацию для создания наглядных графиков.

## Плюсы использования Seaborn

- Seaborn позволяет легко отображать ваши данные на графиках.
- Вы можете использовать Seaborn для визуализации данных, не беспокоясь о внутренних деталях.
- Это позволяет вам просто ввести наш набор данных в функцию `relplot()`, и она вычислит и поместит значение соответствующим образом.
- Свойство 'kind' внутри этой библиотеки позволяет вам переключаться на любой другой формат представления данных.
- Он генерирует динамичный и информативный график для представления ваших данных, упрощая пользователю понимание и просмотр информации в приложении.
- Библиотека seaborn использует статическую агрегацию для генерации графиков для **визуализации данных с помощью Python**.

## Минусы использования Seaborn

- Поскольку она не входит в комплект поставки Python, вам необходимо установить библиотеку Seaborn, выполнив несколько сценариев, прежде чем использовать ее.
- В Seaborn возможности настройки ограничены.
- Интерактивные визуализации в этой библиотеке встречаются редко.
- Иногда пользователям необходимо использовать matplotlib одновременно с seaborn.

## 3. Ggplot

Ggplot - один из лучших пакетов визуализации данных на python с рейтингом [3k + звезд](#) на Github, основанный на реализации ggplot2 для языка программирования R. Используя высокоуровневый API, Ggplot может создавать визуализации данных, такие как столбчатые диаграммы, круговые диаграммы, гистограммы, диаграммы рассеяния, диаграммы ошибок и так далее. Это также позволяет вам объединять многие типы компонентов или слоев визуализации данных в единую визуализацию. После указания того, какие переменные следует сопоставить с определенной эстетикой графика, ggplot позаботится обо всем остальном, позволяя пользователю анализировать визуализации, а не проектировать их. Однако это также означает, что ggplot не позволяет создавать высоконадежные визуализации.



ggplot отличается от Matplotlib тем, что позволяет пользователям складывать компоненты для построения полного графика. Вы можете начать с осей, а затем добавить точки, линию, линию тренда и так далее. Для сильно настраиваемой графики ggplot - не совсем лучший выбор.

### Ключевые особенности ggplot

- ggplot - это еще одна библиотека декларативного стиля, тесно интегрированная с Pandas. Это означает, что вы можете создавать визуализации непосредственно из фрейма данных Pandas.
- ggplot не предназначен для визуализации с кастомной настройкой.
- Использование команды `pip install` в среде Python позволяет установить ggplot, поскольку это модуль с открытым исходным кодом.
- Поскольку ggplot и pandas тесно связаны, хранение ваших данных во фрейме данных важно при использовании ggplot.

### Плюсы использования ggplot

- Если вы переходите с R на Python, вы обнаружите, что использовать ggplot намного проще, чем выполнять ту же задачу с помощью другого пакета Python.
- Для новичков, впервые работающих с ggplot, документация ggplot проста и понятна.
- в ggplot есть метод сохранения, если вам нужно продемонстрировать свои графики или обсудить свои инсайты с другими сотрудниками.

### Минусы использования ggplot

- Некоторые функции, такие как создание карт с помощью `theme_map`, недоступны в ggplot.
- Возможно, вам придется потратить больше времени на изучение документации ggplot, если вы ищете стандартную функциональность, которая не всегда легко преобразуется из R в Python.

### Визуализация данных с помощью ggplot

Иногда использовать ggplot для расширенной визуализации данных на Python может быть непросто, если вы не работали с 'ggplot2' в R. Итак, сначала вам следует попрактиковаться в некоторых базовых примерах использования ggplot на R-

- [Построение круговой диаграммы с помощью ggplot](#)
- [Построение линейной диаграммы с помощью ggplot](#)
- [Создание интерактивной точечной диаграммы с помощью ggplot](#)

Получите **БЕСПЛАТНЫЙ** доступ к [кодам примеров машинного обучения](#) для очистки данных, их обработки и визуализации

#### 4. Plotly- Библиотека Python для 3D-визуализации данных

Plotly - это библиотека Python для **3D-визуализации данных с открытым исходным кодом** с [более чем 50 миллионами пользователей](#) по всему миру. Это веб-инструмент визуализации данных, построенный поверх библиотеки JavaScript Plotly (plotly.js). Точечные диаграммы, гистограммы, линейные диаграммы, столбчатые диаграммы, прямоугольные графики, многоосевые диаграммы, спарклайны, дендрограммы, трехмерные диаграммы и другие виды диаграмм доступны в Plotly. Контурные графики также доступны в Plotly, что немного отличает его от других платформ визуализации данных.



Вы можете использовать Plotly для создания веб-расширенной визуализации данных на Python, представленной в записных книжках Jupyter или веб-приложениях с использованием Dash или сохраненной в виде отдельных HTML-файлов.

##### Ключевые особенности Plotly

- Plotly позволяет вам делиться графиками с общественностью, не раскрывая свой код.
- Синтаксис прост, поскольку все графики используют одни и те же параметры.
- Использование Plotly не требует никаких технических навыков; вы можете создавать визуализации с помощью графического интерфейса пользователя.
- В Plotly доступны 3D-графики с различными интерактивными инструментами.

##### Плюсы использования Plotly

- Возможности инструмента наведения курсора в Plotly позволяют нам выявлять выбросы или аномалии в огромном количестве точек выборки.
- Он обладает эстетичным дизайном, который понравится широкому кругу людей.
- Это позволяет вам персонализировать ваши графики бесконечным количеством способов, делая ваши графики более захватывающими и понятными для всех остальных.

##### Минусы использования Plotly

- Сложно идти в ногу со многими графическими инструментами (Chart Studio, Express и т.д.) и устаревшей документацией.
- Начальная настройка Plotly немного запутанна без онлайн-аккаунта, и приходится писать много кода.
- Хотя Plotly может принимать словари, списки и фреймы данных, нет простых шагов для привязки графиков к одному и тому же исходному набору данных.

#### 5. Geoplotlib

Geoplotlib - одна из лучших библиотек визуализации данных на python для построения географических данных и создания карт. В частности, он позволяет разрабатывать географические карты с использованием различных форматов карт, таких как карты плотности точек, хороплеты и карты символов. Это простой, но эффективный API для визуализации плиток OpenStreetMap. Библиотеки NumPy и SciPy для численных вычислений и модуль Pyglet для графического рендеринга необходимы для выполнения визуализации данных с помощью Geoplotlib. Matplotlib для цветowych карт и pyshp для чтения файлов.shp являются необязательными предварительными условиями.

#### Ключевые особенности Geoplotlib

- Geoplotlib позволяет вам увеличивать и уменьшать масштаб карты, позволяя пользователям видеть больше деталей.
- Эта библиотека без проблем выполняет импорт полного набора данных, картографическую проекцию и перенос фрагментов карты.

#### Плюсы использования Geoplotlib

- В Geoplotlib включено аппаратное ускорение.
- Высокопроизводительный рендеринг больших наборов данных может принести пользу.
- Интерактивность и базовая анимация обеспечиваются с помощью фрагментов карты.

## 6. Vokeh

Vokeh - еще один пример лучших инструментов визуализации данных python с более чем 15 тысячами звездочек на Github. Он генерирует подробные изображения с высоким уровнем взаимодействия для различных наборов данных, больших и малых. Это позволяет красиво и просто создавать адаптируемую графику с высокопроизводительной интерактивностью в больших или потоковых наборах данных. Специалисты по визуализации данных могут использовать vokeh для создания интерактивных графиков для современных веб-браузеров, которые можно использовать в интерактивных онлайн-приложениях, HTML-страницах или объектах JSON.



В Vokeh есть три уровня для создания визуализаций данных. Первый уровень фокусируется исключительно на быстром построении графиков данных, а второй уровень управляет основными компонентами построения графика. Третий уровень предоставляет пользователям полную гибкость при создании диаграмм без каких-либо предустановленных настроек.

#### Ключевые особенности Vokeh

- Vokeh поддерживает активное взаимодействие с графиком, такое как масштабирование, панорамирование, выделение и т.д.
- Он предоставляет низкоуровневый интерфейс с дополнительной гибкостью для настройки диаграмм.
- Vokeh также может конвертировать Matplotlib, ggplot.py и диаграммы и графики seaborn.

- Это помогает создавать интерактивные графики, которые можно сохранять как в форматах PNG, так и SVG.
- Bokeh генерирует выходные данные в различных формах, включая HTML, notebook и т.д.

#### Плюсы использования Bokeh

- Связать графики проще с помощью Bokeh. Изменение, внесенное на одном графике, будет воспроизведено на графике с соответствующей переменной.
- Bokeh можно использовать как высокоуровневый или низкоуровневый интерфейс, что позволяет ему создавать многие из тех же графиков, что и Matplotlib, но с меньшим количеством строк кода и более высоким разрешением.

#### Минусы использования Bokeh

- Поскольку Bokeh - это пакет среднего уровня, часто требуется меньше кода, чем Matplotlib, но больше кода для создания того же графика, что и Seaborn, Altair или Plotly.
- У него пока нет сильной группы поддержки, и он все еще находится на ранних стадиях разработки.
- Он не поддерживает функции 3D-графики и имеет ограниченные возможности интерактивности для работы.
- Прежде чем создавать какой-либо график, вы должны сначала определить режим вывода, включая режимы ноутбука, сервера и веб-браузера.

**Изучите больше проектов в области [науки о данных и машинного обучения](#) для практики. Ускорьте свой карьерный рост с ProjectPro**

#### 7. Folium

Folium - это библиотека визуализации геопространственных данных, написанная на Python. Это простая в использовании библиотека с большими возможностями. Он сочетает в себе мощь Leaflet.js и простоту Python для создания фантастического инструмента построения карт. Folium был создан с акцентом на простоту, производительность и полезность. Он хорошо работает, может быть расширен с помощью различных плагинов и имеет удобный API.



#### Ключевые особенности Folium

Folium упрощает разработчикам создание Google Maps, размещение маркеров и отображение маршрутов, отнимающих много времени. Вы можете просто загрузить несколько библиотек, создать карту и сосредоточиться на вводе и анализе данных в Folium.

#### Плюсы использования Folium



- Folium использует открытую карту улиц, чтобы предоставить вам опыт, подобный Google Map, с меньшим количеством кода.
- Folium упрощает добавление предполагаемых местоположений других пользователей, позволяя пользователям добавлять маркеры.
- Вы можете использовать ряд плагинов Folium для своей карты, включая плагин Altair.

### Минусы использования Folium

- Проблема базовой карты заключается в том, что вам приходится получать шейп-файлы и обрабатывать их, а результатом являются статические изображения (например, графики в matplotlib).
- В интерактивной среде с этой библиотекой просто работать, но из-за нее сложно просто передать результаты своего труда кому-то другому.

## 8. Библиотека Altair Python

Altair - это пакет визуализации статистических данных на Python. Он основан на Vega и Vega-Lite, декларативных языках для создания, сохранения и совместного использования интерактивных проектов визуализации данных. Altair может создавать привлекательные визуализации данных в виде графиков с небольшим количеством сценариев, включая столбчатые диаграммы, круговые диаграммы, гистограммы, диаграммы рассеяния, диаграммы ошибок, базовые диаграммы и многое другое.



В Altair есть такие зависимости, как python 3.6, NumPy, Pandas, которые устанавливаются автоматически в соответствии с процедурами установки Altair. Вы можете использовать Jupyter Notebook или JupyterLab для получения визуализаций данных в Altair.

### Ключевые особенности Altair

- При создании визуализации с помощью Altair вы можете агрегировать данные, и это устраняет различные процедуры, которые обычно выполняются с помощью пакета анализа и манипулирования данными, такого как Pandas.
- Поскольку Altair создает графики в декларативной форме, при использовании этого пакета легко выполнять визуализации и эксперименты.
- Фильтрация данных - еще одна замечательная функция, доступная в Altair, позволяющая создавать более сфокусированные или индивидуальные визуализации.
- Вы также можете использовать Altair для обеспечения динамической фильтрации, и это позволяет вам использовать общий фильтр для соединения многих графиков.

### Плюсы использования Altair

- Он прост в использовании и позволяет создавать более привлекательные и убедительные визуализации.



- Хотя структура кодирования остается прежней, простым изменением атрибута `mark` можно создать несколько графиков.
- Altair helps you to better understand your data by supporting data transformations, such as using the count, min, and max aggregator functions.

#### Cons of using Altair

- Altair - неподходящая библиотека визуализации для вас, если вам нужно создавать 3D-визуализации.
- Altair, как и многие высокоуровневые платформы визуализации, не является полностью настраиваемым,
- В Altair отсутствуют некоторые основные графики, такие как boxplots.

## 9. Pygal

Это один из самых популярных пакетов Python для создания интерактивных графиков, который вы можете встроить в веб-браузер. Это отличная библиотека для работы с небольшими наборами данных, и она генерирует SVG-файлы, что отличает ее от других в списке. Хотя это позволяет получать похвальные результаты с небольшими наборами данных, создание диаграмм с сотнями тысяч точек данных может быть сложной задачей, и в таких случаях может быть трудно выдавать результаты.

# Pygal

Для установки библиотеки Pygal можно использовать `pip`. Pygal упрощает построение графиков и поддерживает различные типы диаграмм, включая линейные, столбчатые, гистограммные, радарные, прямоугольные, древовидные карты и т.д.

#### Ключевые особенности Pygal

- Этот модуль может использоваться для построения динамических и интерактивных графиков на веб-странице с использованием типичных веб-интерфейсов Python, таких как flask и Django.
- Это может упростить работу с презентациями SVG с интерактивными файлами.
- Pygal - отличный выбор для небольших веб-приложений, которым требуются быстрые и эффективные графики.

#### Плюсы использования Pygal

- Pygal может создавать широкий спектр графиков, таких как линейные, столбчатые, гистограммные, XY, круговые, радарные, прямоугольные, точечные и так далее.
- С минимальным количеством кодирования вы можете создавать отличительные и визуально ошеломляющие графики.
- Диаграммы и графики можно экспортировать в различные форматы, включая SVG, PNG, Etree и другие.

#### Минусы использования Pygal

- SVGS будет достаточно, пока вы работаете с небольшими наборами данных. Однако работа с массивными наборами данных, содержащими тысячи точек данных, делает визуализацию сложной и неэффективной.

## 10. Блеск

Glean - это библиотека Python, которая позволяет создавать интерактивные онлайн-визуализации данных без необходимости знать HTML или JS. Glean объединяет все в веб-интерфейс, который позволяет любому пользователю взаимодействовать с вашими данными в режиме реального времени. Вам будет проще объяснять и интерпретировать ваши данные другим пользователям. Пакет Shiny в R послужил источником вдохновения для разработки библиотеки Glean.

Glean используется для создания интерактивных визуализаций, которые включают страницы, панели и кнопки. Эти интерактивные веб-визуализации также полностью интегрированы в Веб, что означает, что они могут быть встроены во что угодно - от веб-сайта до конечной точки.

### Ключевые особенности Glean

- Это позволяет превратить анализ в интерактивные веб-приложения, используя только скрипты на Python, сводя к минимуму необходимость знать другие языки, такие как HTML, CSS или javascript.
- Он работает практически со всеми другими библиотеками визуализации данных Python.
- Вы можете добавлять поля к графику после его создания, чтобы пользователи могли фильтровать и сортировать данные.