Простые способы визуализации данных на Python.

Визуализация данных — это большая часть работы специалистов в области data science. На ранних стадиях развития проекта часто необходимо выполнять разведочный анализ данных (РАД, Exploratory data analysis (EDA)), чтобы выявить закономерности, которые обнаруживают данные. Визуализация данных помогает представить большие и сложные наборы данных в простом и наглядном виде. На этапе окончания проекта важно суметь отчитаться о его результатах так, чтобы даже непрофессионалам, не обладающим техническими знаниями, всё стало ясно и понятно.

Matplotlib — это популярная библиотека для визуализации данных, написанная на языке Python. Хоть пользоваться ей очень просто, настройка данных, параметров, графиков и отрисовки для каждого нового проекта — сложное занятие. Разберем простые способов визуализации данных и напишем быстрые и простые функции для их реализации с помощью питоновской библиотеки Matplotlib.

Диаграммы рассеяния (Scatter Plots)

Используются если нужно показать связь между двумя переменными, так как они позволяют отображать грубое распределение данных. На нем также можно показать соотношение между различными группами данных за счет окрашивания их разными цветами. Если Нужно показать взаимосвязь между тремя переменным добавляем дополнительные параметры, такие как размер точек, чтобы закодировать эту третью переменную, как это сделано на втором графике снизу.

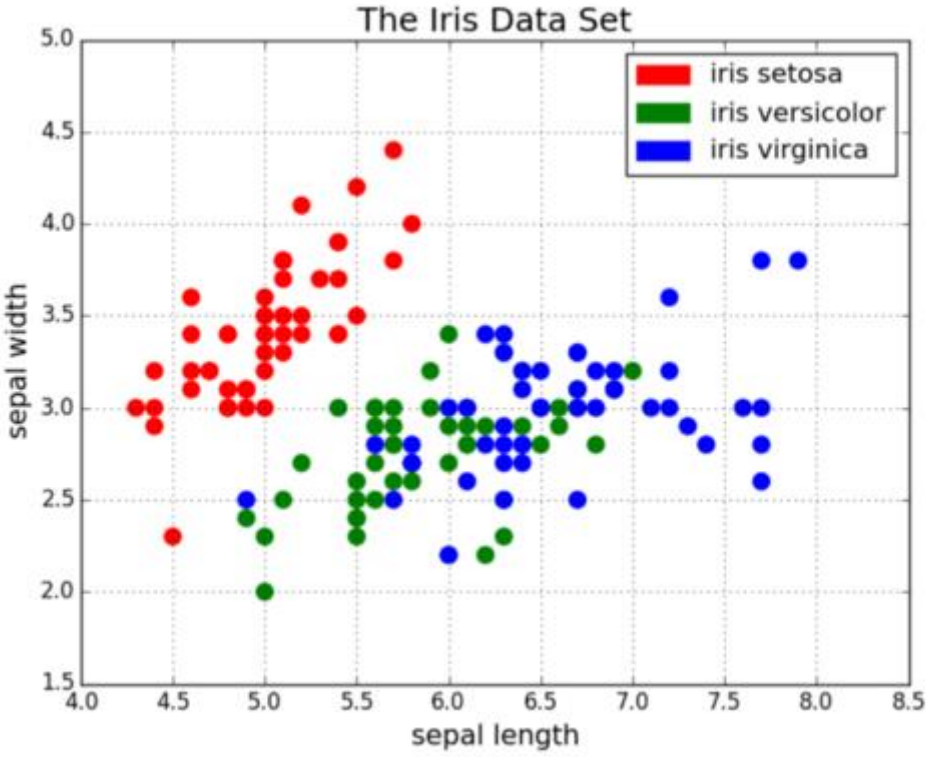


Диаграмма рассеяния с группировкой по цветам

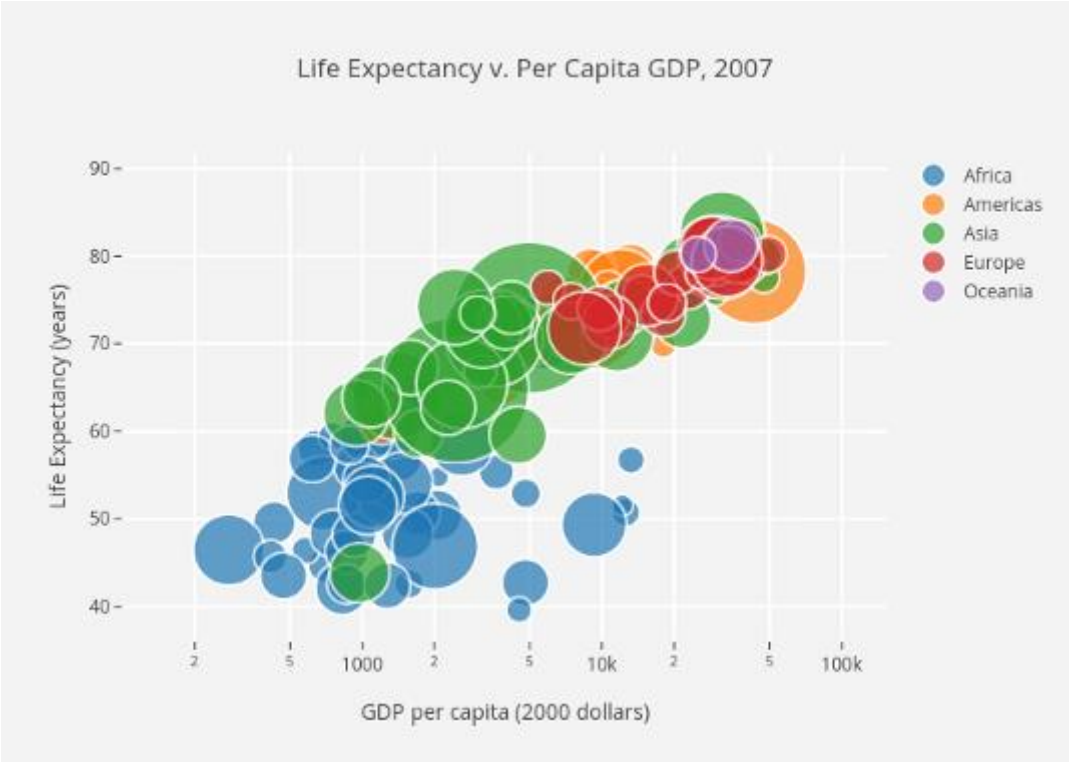
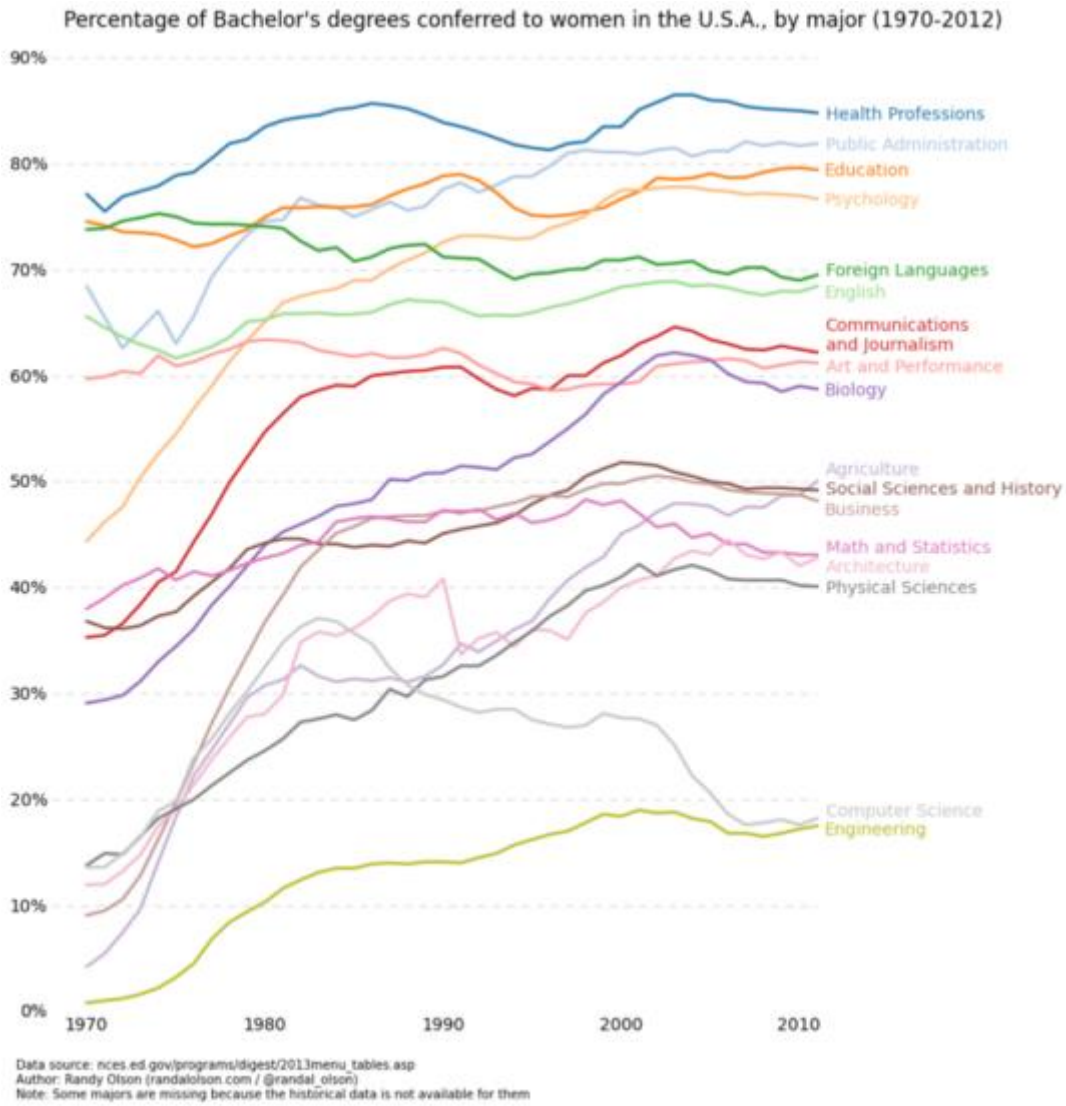


Диаграмма рассеяния с группировкой по цветам и по размерам для отображения 3-го параметра — размера страны

Графики

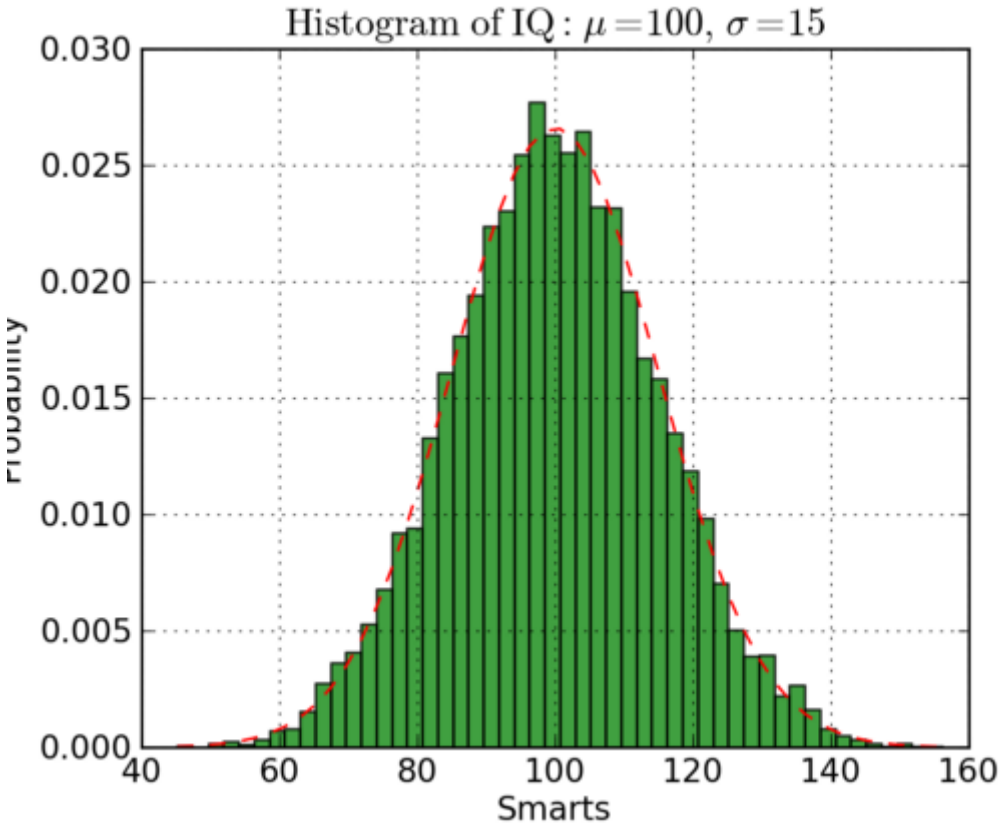
Графики лучше всего использовать тогда, когда одна переменная сильно варьируется в зависимости от другой, другими словами, когда у них высокая ковариация. График наглядно показывает большой разброс процентного соотношения за указанный промежуток времени. Если бы эти же данные мы представили в виде диаграммы рассеяния, она была бы чрезвычайно загроможденной и сложной, что затрудняло бы понимание и визуальное отображение рассматриваемой зависимости. Графики, выполненные в виде линий, идеально подходят для этой ситуации, потому что они показывают ковариации двух переменных (в данном случае процент и временной промежуток). Как и в предыдущем примере, мы можем использовать группировку с помощью цвета.



Пример графика. На нем показано процент степеней бакалавра, присвоенных женщинам в США (1970–2012)

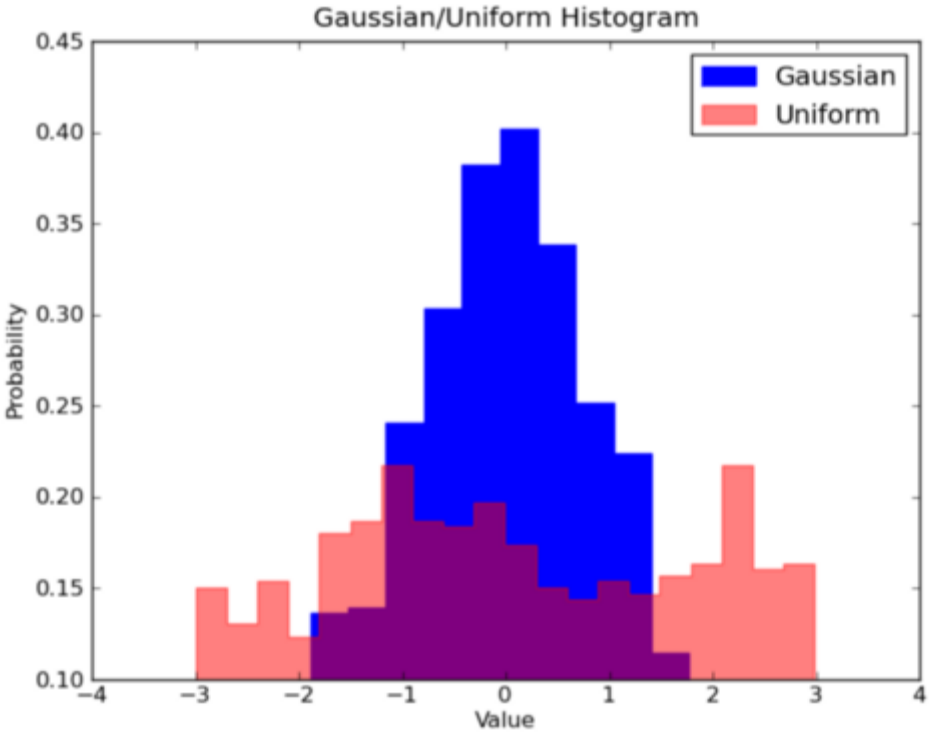
Гистограммы (Histograms)

Гистограммы полезны для представления (или даже выявления) распределения данных. Посмотрите на пример ниже, где мы построили гистограмму частоты IQ. Мы легко можем заметить концентрацию ближе к центру, а также отчетливо прослеживается медиана значений. Мы также видим, что оно подчиняется гауссовскому распределению. Использование столбцов (а не точек рассеивания, например) действительно дает нам четкую визуализацию относительной разницы между частотой каждого интервала. Использование полос (интервалов = дискретизация) действительно помогает нам увидеть «целостную картину». Если эти же данные представить в виде отдельных точек, без выделения интервалов, то на диаграмме появится слишком много шума, что затруднит понимание тенденции, которая иллюстрируется с помощью этих данных.



Пример гистограммы

Теперь представьте себе, что мы хотим сравнить распределение двух переменных в наших данных. Первая мысль, которая приходит в голову — это сделать две отдельные гистограммы и расположить их рядом, для наглядности. Но на самом деле есть способ лучше: мы можем накладывать гистограммы с различной прозрачностью. Посмотрите на рисунок, представленный ниже. Равномерное распределение имеет прозрачность 0,5, чтобы мы могли видеть, что расположено за ним. Это позволяет одновременно отобразить два распределения на одном рисунке.



Наложение двух гистограмм: гауссовского и равномерного распределения

Столбчатые диаграммы (Bar Plots)

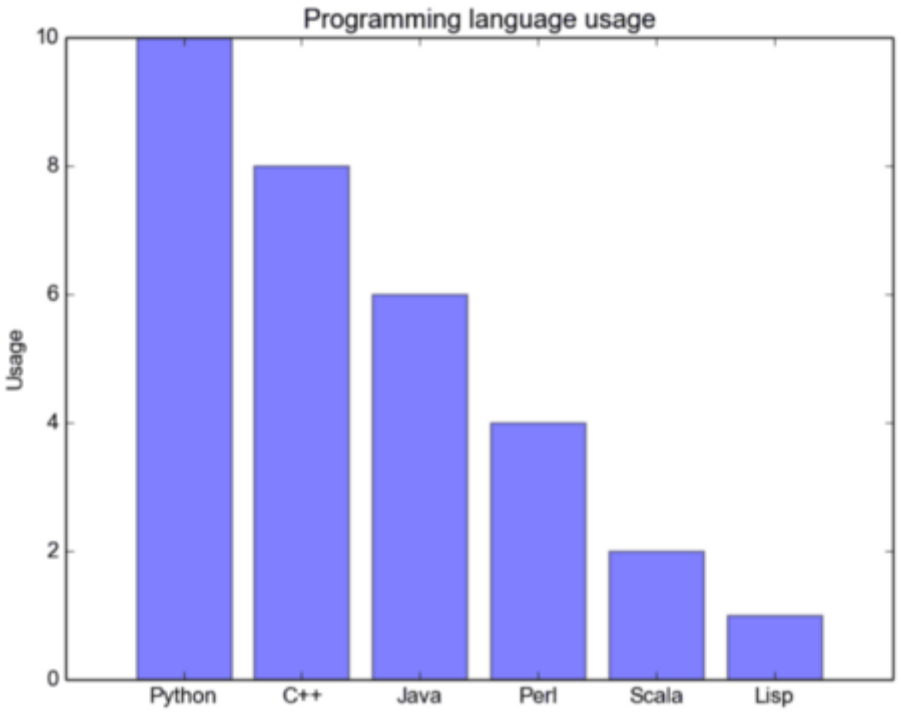
Столбчатые диаграммы наиболее эффективны тогда, когда вам необходимо визуализировать данные в виде категорий, если их число не превышает 10. Если у нас слишком много категорий, то столбцы будут сильно загромождать график, и его трудно будет понять. Они хороши для данных, разделенных по категориям, потому что вы можете легко увидеть разницу между категориями в зависимости от размера столбца (например, величины); категории также легко можно сформировать и выделить цветом. Есть три разных типа столбчатых диаграмм, которые мы будем рассматривать далее: обычные, сгруппированные и составные. Каждый из этих типов мы рассмотрим по порядку.

Обычная столбчатая диаграмма находится на первом рисунке снизу. В функции barplot() x_data задает метки на оси x, а y_data задает высоту столбца по оси y. Строка ошибки представляет собой дополнительную линию, расположенную в центре каждого столбца, которая может быть использована для отображения стандартного отклонения.

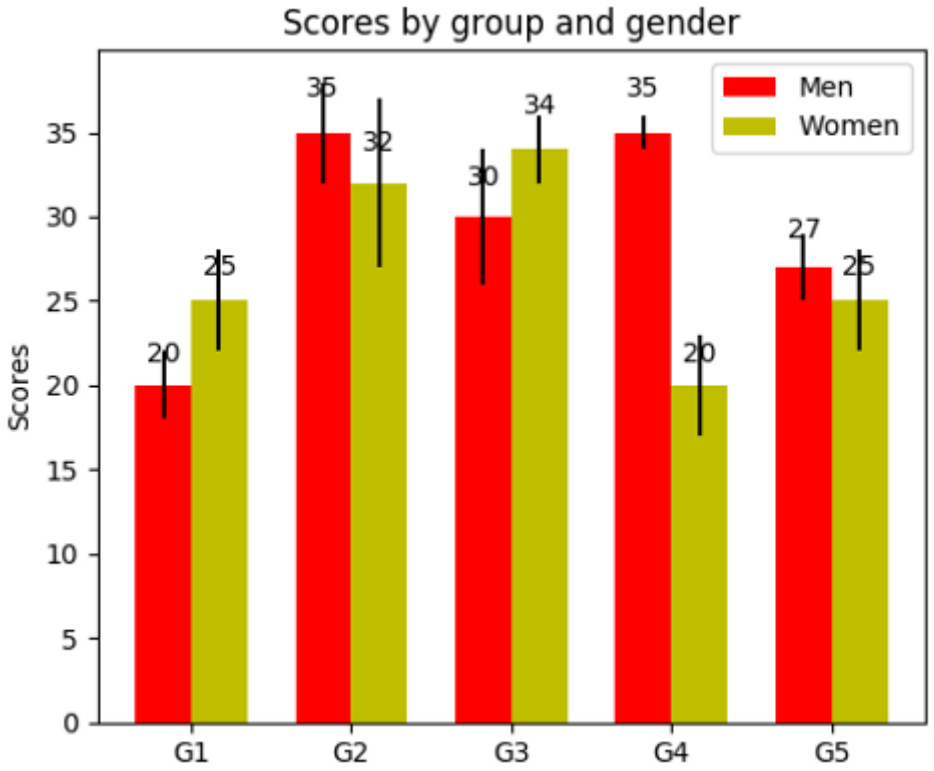
Сгруппированные столбчатые диаграммы позволяют сравнивать несколько переменных. Посмотрите на второй график снизу. Первой переменной, которую мы сравниваем, задается то, как оценки варьируются от группы к группе (группы G1, G2, … и так далее). Мы также сравниваем между собой распределение полов, что закодировано цветом. Теперь взгляните на код — вы заметите, что переменная y_data_list теперь фактически представляет собой список списков, где каждый вложенный список обозначает другую группу. Затем мы проходимся циклом по каждой группе, и для каждой группы рисуем столбец для каждого метки по оси x; все группы также дополнительно окрашиваются.

Составные столбчатые диаграммы отлично подходят для визуализации набора различных переменных. В приведенном ниже рисунке с разбивкой по строкам мы отслеживаем изменение нагрузки на сервер по дням недели. С помощью цветовых наборов мы можем легко видеть и понимать, какие серверы работают больше всего в каждый конкретный день и как в целом

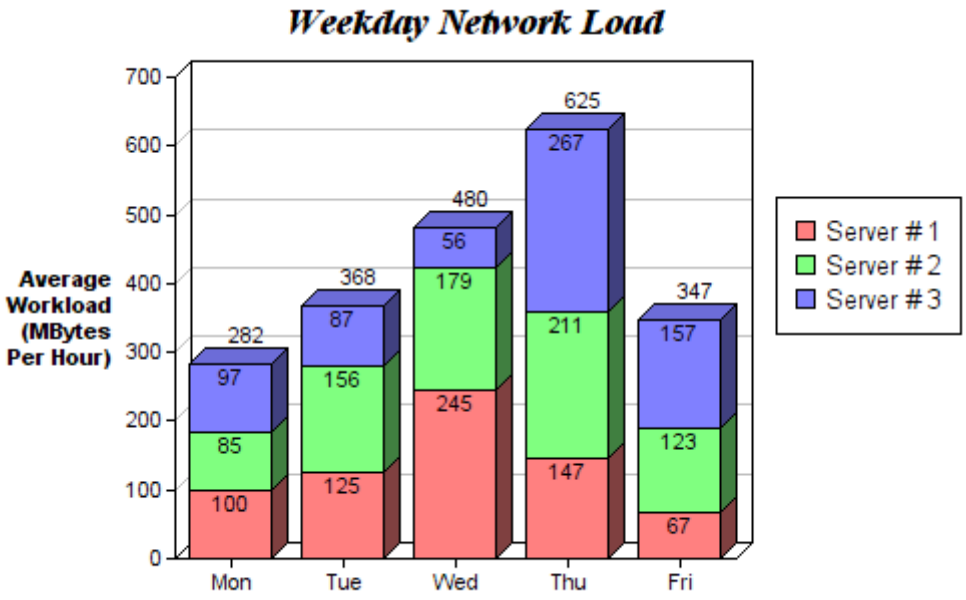
распределяется нагрузка по дням на все сервера. Код для этой диаграммы строится по тому же принципу, что и код для сгруппированных столбчатых диаграмм. Мы проходим циклом по каждой группе, с одним единственным исключением: на этот раз мы рисуем новые столбцы поверх старых, а не рядом с ними.



Обычная столбчатая диаграмма. Использование языков программирования



Сгруппированная столбчатая диаграмма. Распределение оценок по полу и возрасту



Составная столбчатая диаграмм. Нагрузка на сервер по дням недели