

D3.js – samples (Coffee and Code)

- Scribble (line chart)
- [Line chart](#)
- [Pie chart](#)
- [Parallel coordinates chart](#)
- [Ridgeline chart](#)

"Я и визуализация"

Источник: [Лапки](#)

Диаграмма

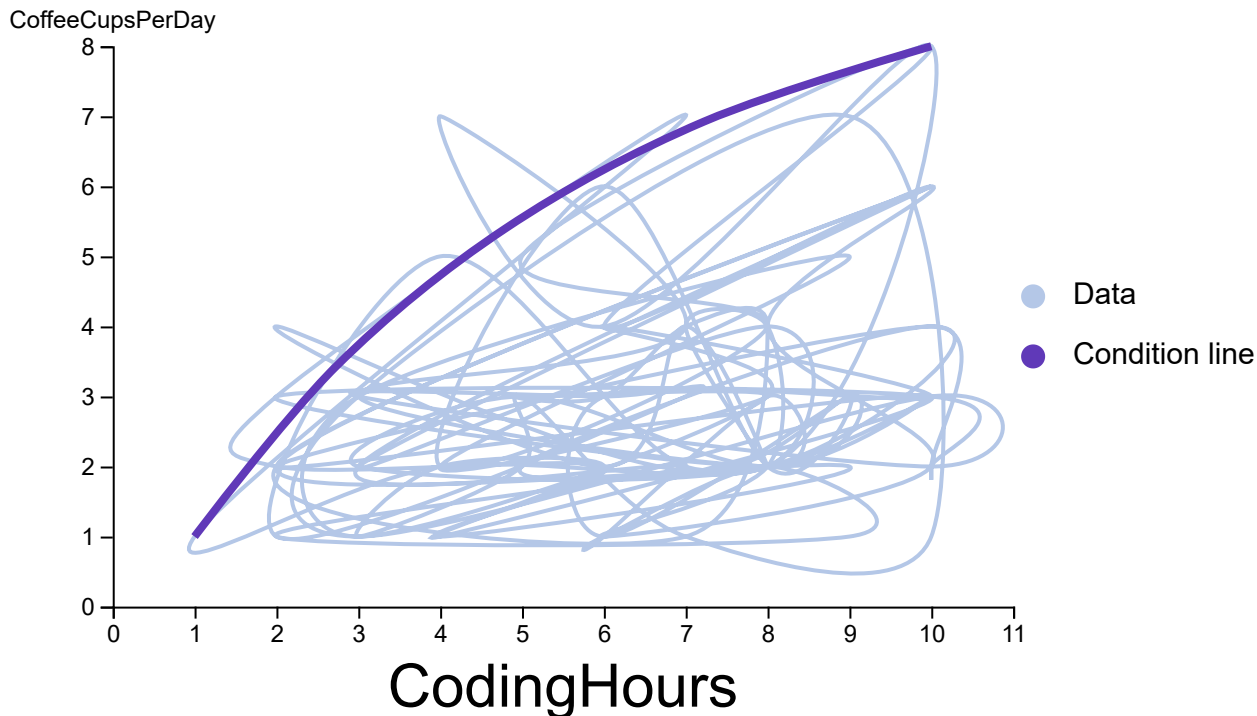


Диаграмма показывает мою первую попытку сделать линейный график. К моему удивлению, даже из этого, казалось бы, неудачного графика, можно сделать выводы: мы можем заметить, что существует такая функция (фиолетовый граик), относительно которой все наши данные (за редким исключением) оказываются с одной стороны, причём пустого пространства с этой стороны почти не остаётся. Таким образом, можно наблюдать, что зависимость часов кодинга от количества чашек кофе подчиняется некому конкретному условию. Например, из этого графика следует, что невозможно кодить мало, если ты выпил много кофе, но возможно наоборот: выпить всего одну чашку и работать весь день.

Код

D3.js – samples (Coffee and Code)

- [Scribble \(line chart\)](#)
- [Line chart](#)
- [Pie chart](#)
- [Parallel coordinates chart](#)
- [Ridgeline chart](#)

Линейный график показывает эволюцию одной или нескольких числовых переменных. Точки данных связаны прямыми отрезками или плавной линией. Он аналогичен точечной диаграмме, за исключением того, что точки измерения упорядочены (обычно по значению их оси x) и соединены. Это наиболее популярный график для демонстрации трендов и соотношения показателей (при использовании нескольких линий). Линейные графики очень полезны для получения «общей картины» за определенный промежуток времени и наблюдения за развитием в этот период времени.

Источник: [From Data To Viz](#)

Диаграмма

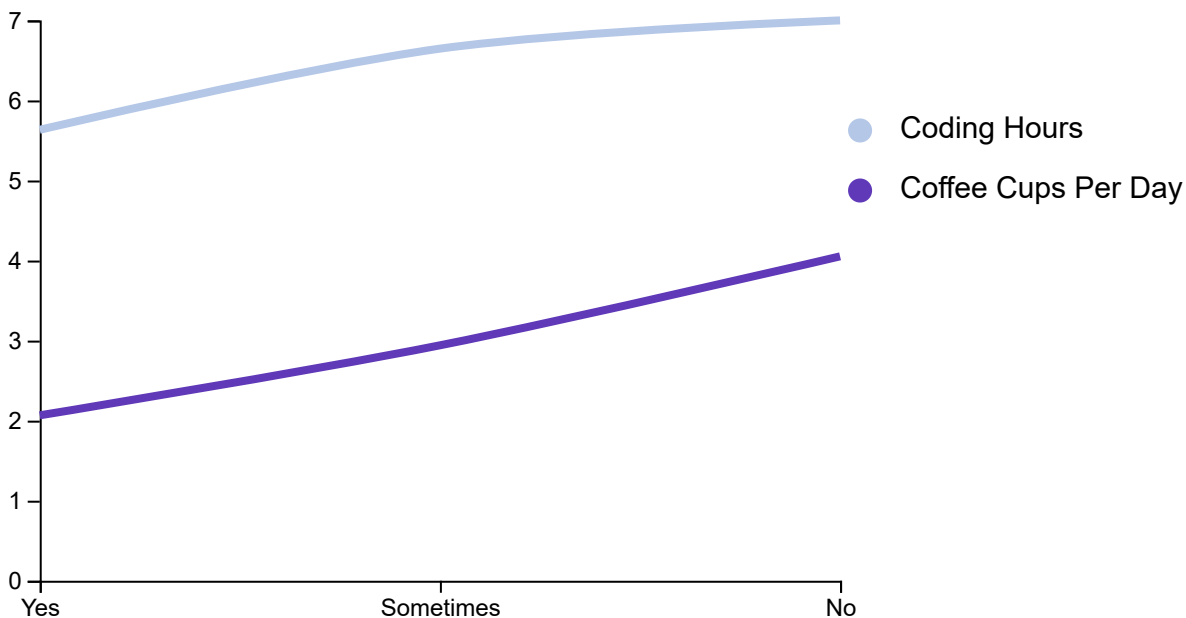


Диаграмма показывает, как зависит количество выпиваемого кофе в среднем за сутки, от того, программирует ли человек без него. Так же тут изображена зависимость количество часов кодирования. Легко заметить, что эти 2 параметра напрямую связаны.

Код

D3.js – samples (Coffee and Code)

- [Scribble \(line chart\)](#)
- [Line chart](#)
- Pie chart
- [Parallel coordinates chart](#)
- [Ridgeline chart](#)

Круговая диаграмма - это круг, разделенный на сектора, каждый из которых представляет собой часть целого. Он часто используется для отображения процента, где сумма секторов равна 100%.

Источник: [From Data To Viz](#)

Диаграмма

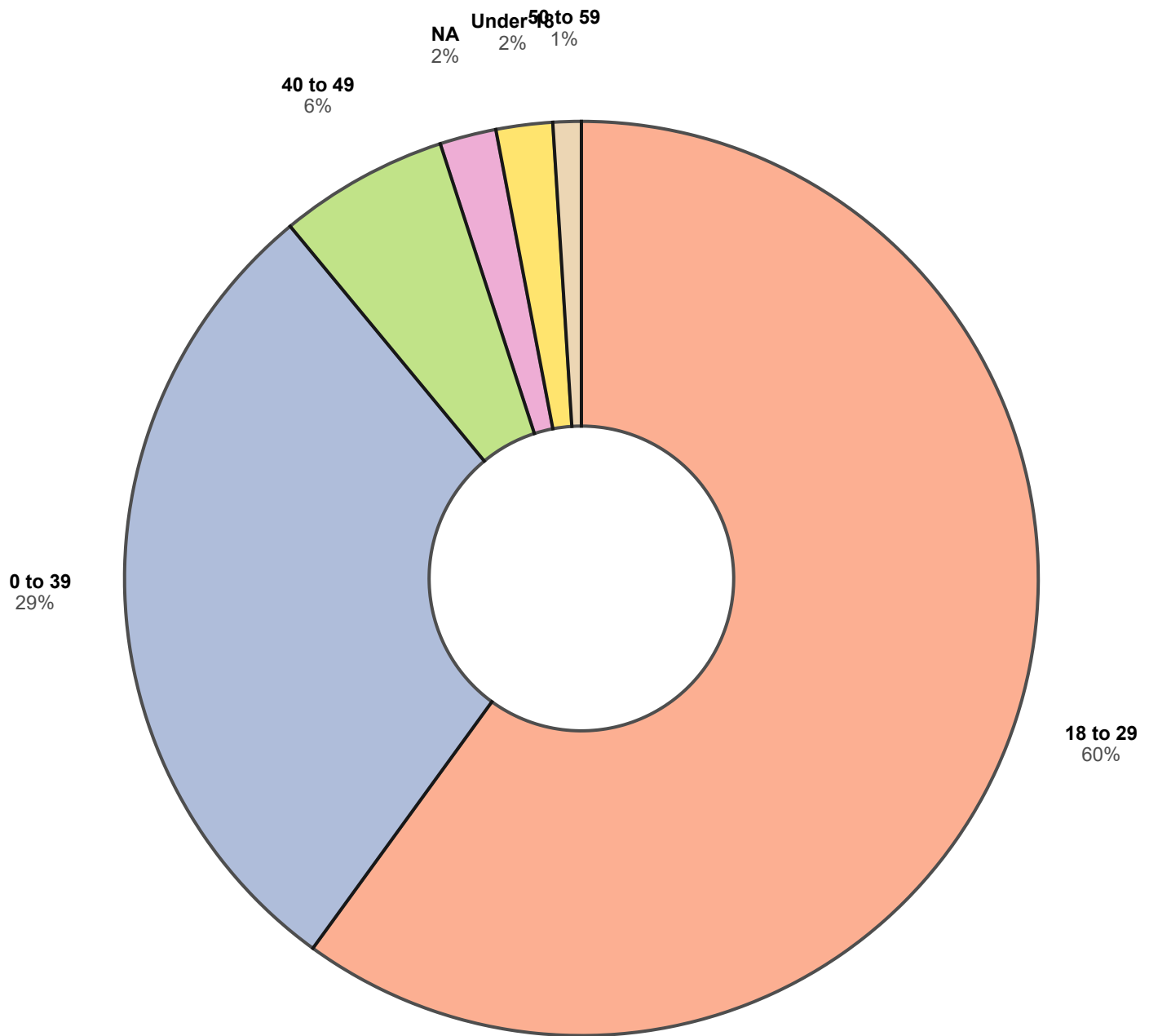


Диаграмма показывает разделение программистов из датасета по возрастам.

Код

D3.js – samples (Coffee and Code)

- [Scribble \(line chart\)](#)
- [Line chart](#)
- [Pie chart](#)
- Parallel coordinates chart
- [Ridgeline chart](#)

График параллельных координат позволяет сравнить особенность нескольких отдельных наблюдений по набору числовых переменных. Каждая вертикальная полоса представляет переменную и часто имеет свой собственный масштаб. Значения наносятся на график в виде серии линий, соединяющих оси в соответствии с набором переменных. Диаграммы с параллельными координатами идеально подходят для сравнения одновременно большого количества переменных и анализа их взаимосвязей.

Источник: [From Data To Viz](#)

Диаграмма

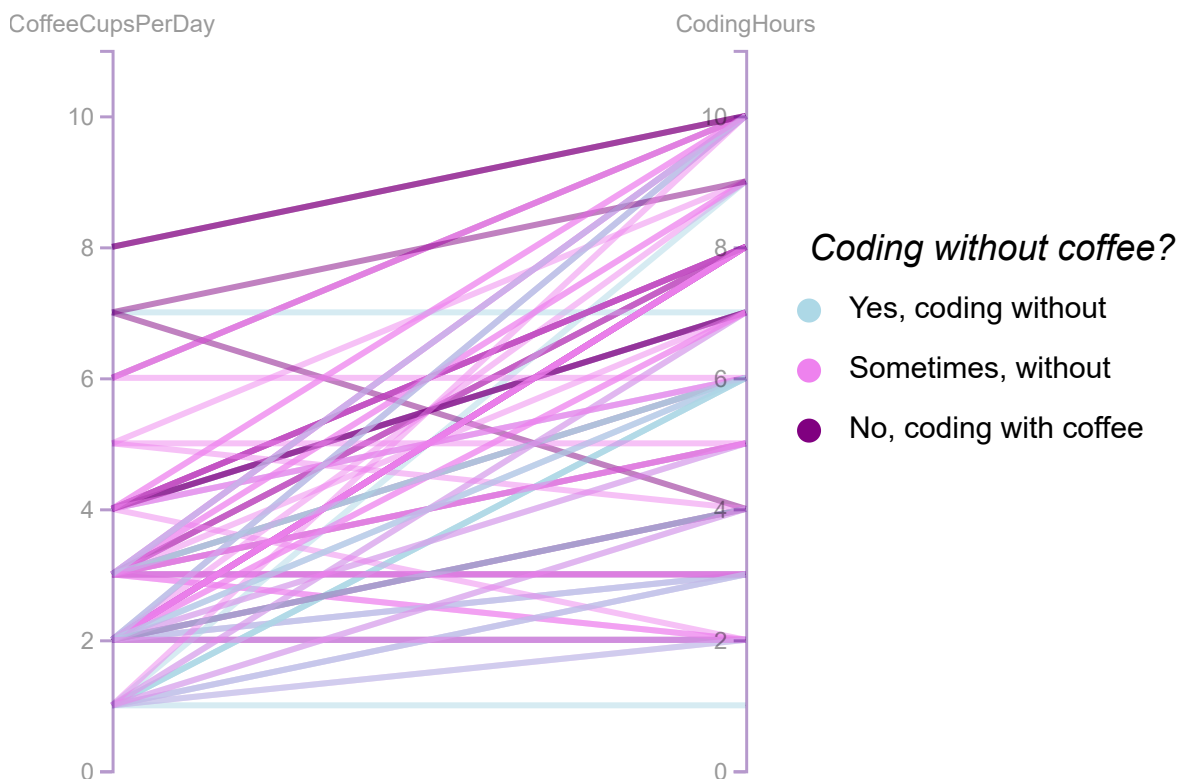


Диаграмма показывает связь количества выпитых чашек кофе с продолжительностью кодинга в различных группах людей (программисты сгруппированы по ответу на вопрос о кодировании без кофе)

Код

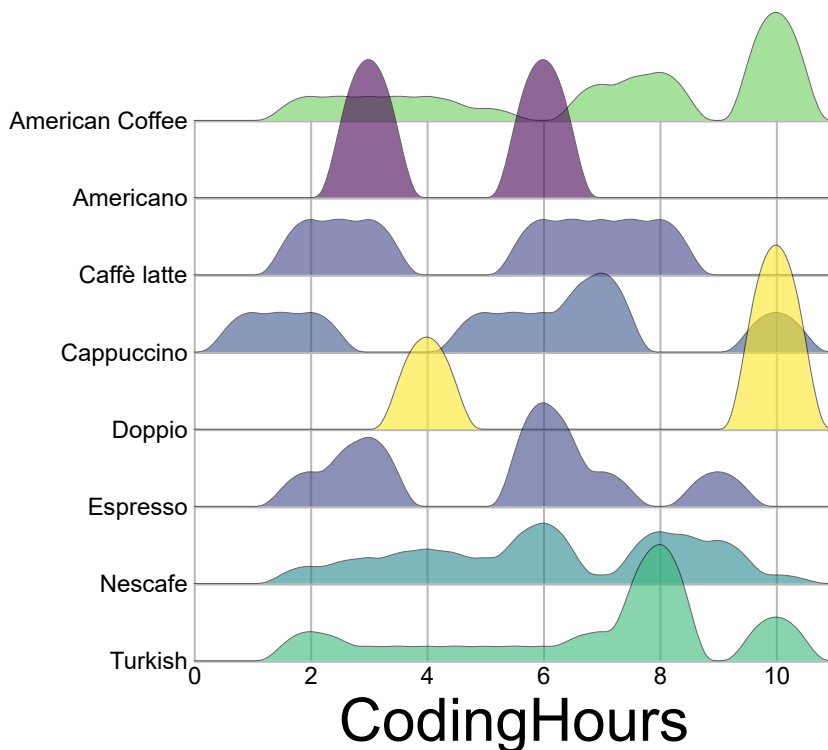
D3.js – samples (Coffee and Code)

- [Scribble \(line chart\)](#)
- [Line chart](#)
- [Pie chart](#)
- [Parallel coordinates chart](#)
- Ridgeline chart

Это график плотности распределения вероятности, иногда называемый `joyplot`, позволяет визуализировать распределение нескольких числовых переменных. Цвет зависит от среднего значения (чем больше среднее значение, тем светлее цвет).

Источник: [From Data To Viz](#)

Диаграмма



1) Диаграмма показывает плотность распределения вероятности программировать то или иное количество часов в сутки, в зависимости от того, какой кофе вы предпочитаете.

2) Цветовое оформление диаграммы показывает зависимость среднего времени, проведённого за программированием, относительно выбора кофе.

Код