Наиболее важными для нас являются диаграммы размаха, точечные графики, ядерные оценки плотности и гистограммы. Остальные способы визуализации приведены для расширения кругозора.

Одномерные массивы данных

- Диаграмма размаха (box plot), также известная как ящик с усами.
  Этот график упрощает данные, оставляя только несколько параметров: медиану (грубо говоря такое число, что половина данных меньше его, а вторая половина больше), верхнюю и нижние квартили (т.е. такие числа, что четверть данных больше верхней квартили и четверть меньше нижней), существенный минимум и максимум (наименьшее и наибольшее значения, не считая выбросов), а также выбросы, которые существенно меньше или больше остальных чисел в ряду.

  Обычно мы применяем этот способ, если мы хотим визуально сравнить несколько категорий данных.

  В R этот график реализован с помощью обычной функции boxplot, но более удобным для нас будет использование функции qplot пакета ggplot2.

  Базовая форма вызова функции qplot(gear, mpg, data=mtcars, geom="boxplot", fill=factor(gear))*, где data — массив из которого мы берем данные, gear — переменная отвечающая за категорию данных, mpg — за сами данные, а fill раскрашивает диаграммы в соответствии со значением переменной gear. Здесь к gear мы применили функцию factor, которая из числовой переменной сделала категориальную.

  Картинки, приведенные в файле, можно также увидеть в project в папке pictures

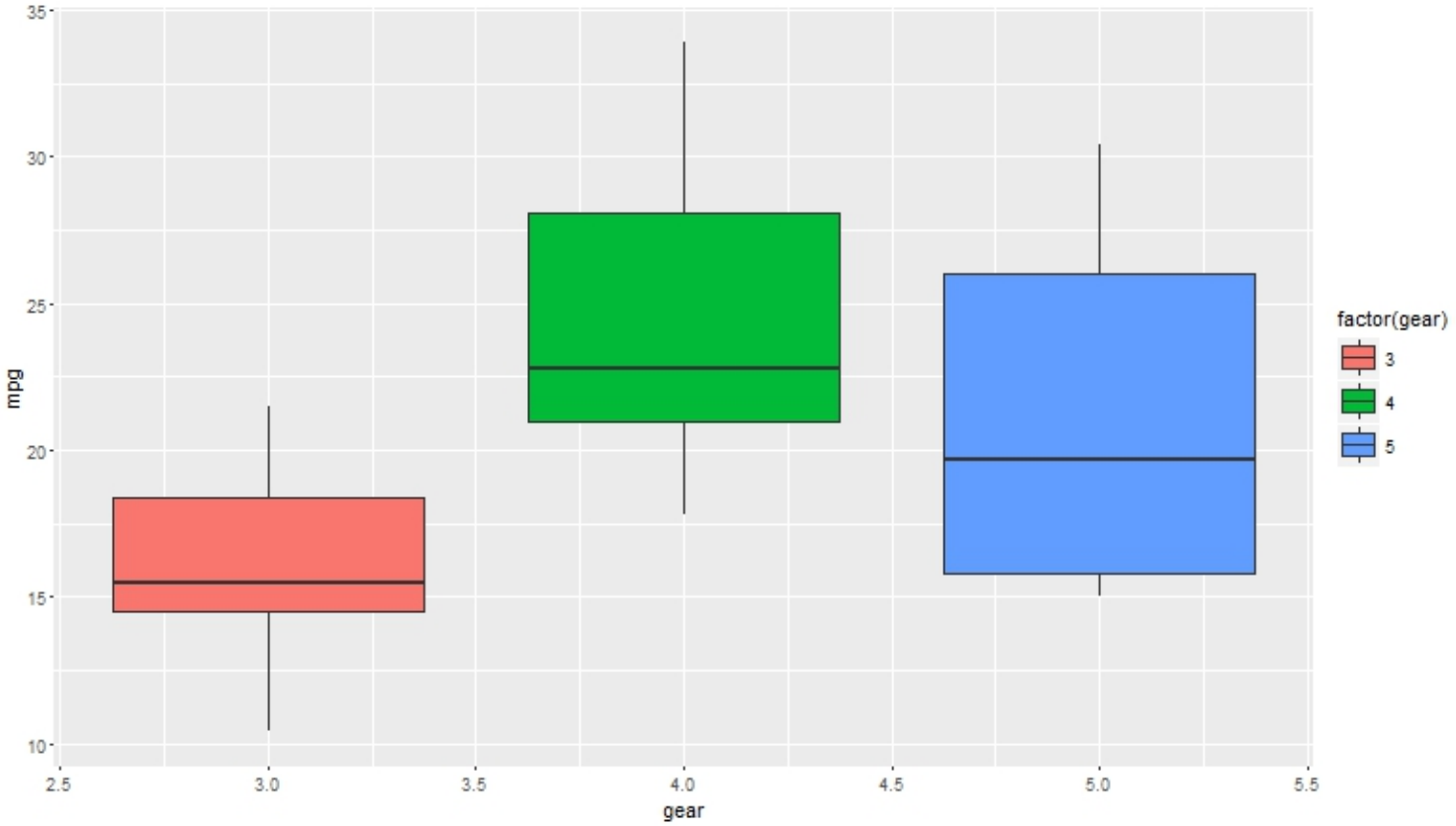

Рис. 1: Box plot

  У функции qplot есть еще много других параметров, в частности, xlab, ylab отвечают за подписи по осям, main за название.

- Столбцовая диаграмма (Bar plot)
  Для категориальных данных, принимающих мало различных значений, используют столбцовую диаграмму, которая строит несколько столбцов с высотами, соответствующими количеству наблюдений.

  В R этом можно сделать, например, так qplot(gear, data=mtcars, geom="bar", fill=I("red"))*

  Также как и в прошлый раз, мы можем использовать для параметра fill не постоянное значение "red", а вектор значений, скажем, qplot(gear, data=mtcars, geom="bar", fill=factor(cyl))*

- Гистограмма (histogram). Гистограмма разбивает оси на интервалы, заменяет значения переменных на номер интервала попадания и строит по полученным данным столбцовую диаграмму.

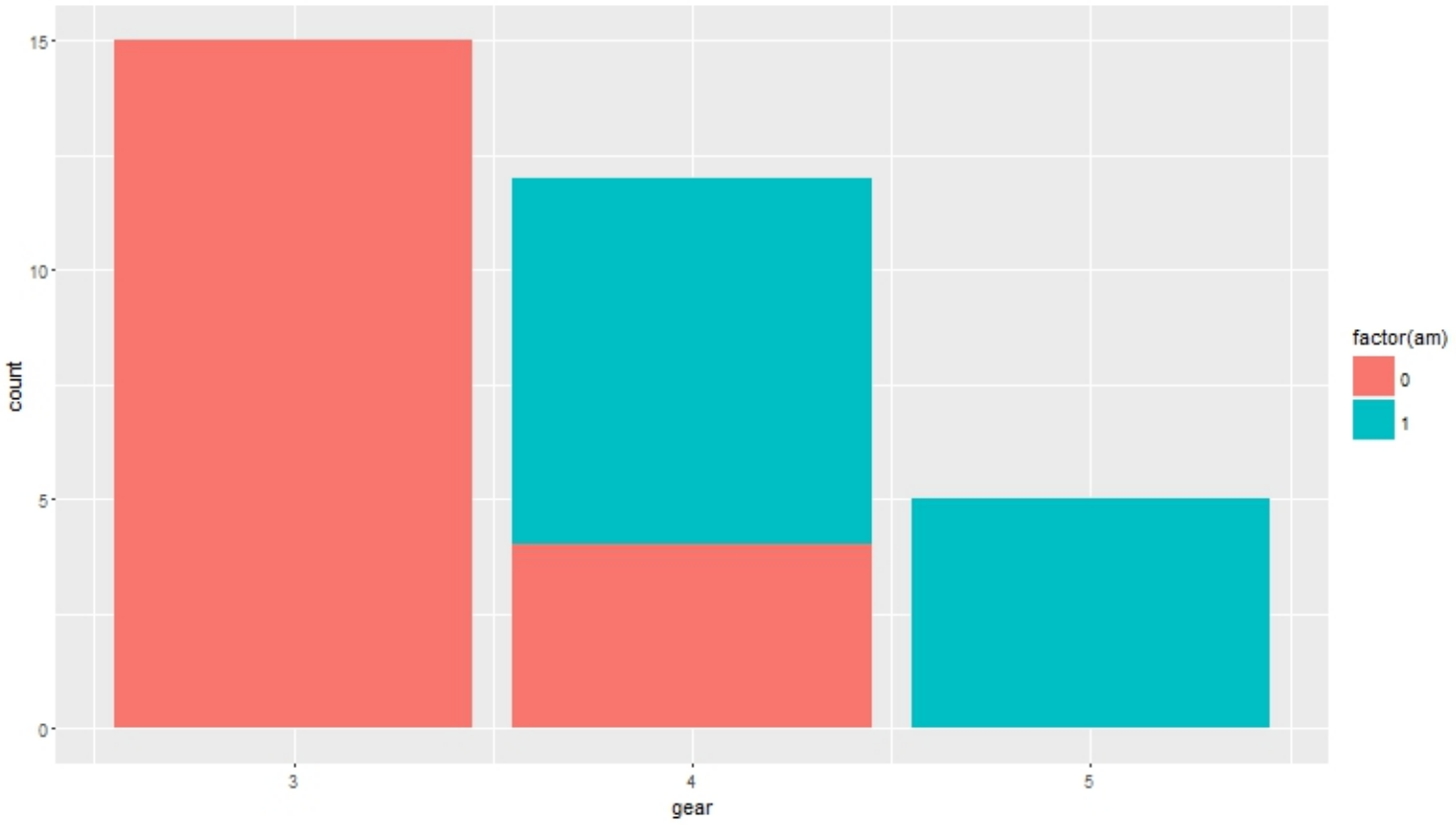

Рис. 2: Bar plot

Гистограмма оценивает плотность распределения соответствующей случайной величины. Высокие столбцы соответствуют частой встречаемости значения, низкие — редкой.

В R ее можно построить с помощью базовой функции hist или же через qplot(mpg, data=mtcars, geom="histogram", fill=I("red"))* Отметим также вызов с большим числом параметров — здесь мы выбрали ширину интервалов и цвет границы: qplot(mpg, data=mtcars,geom="histogram",fill=I("red"),binwidth = 2,col=I("blue"))*

- Ядерная оценка плотности (density plot). Гистограмма дает дискретизированное приближение функции плотности, между тем можно получить множество гладких оценок для этой функции. Популярным методом являются так называемые ядерные оценки.

  В R такой график можно построить с помощью qplot(mpg, data=mtcars,geom="density",fill=factor(cyl), alpha=0.5),* где параметр alpha задает прозрачность.

  Совместить на одном графике и то и то можно так:*

- Скрипичные графики (violin plot).

  Если у нас много графиков, то сравнивать их плотности очень тяжело — картинка получается перегруженной. В этом случае обычно используют box plot. Но существует также некоторый компромисс, называемый violin plot

  Violin plot строит вдоль вертикальной оси симметричную фигуру, чей контур соответствует плотности. Это позволяет и увидеть соотношение между несколькими категория данных, и увидеть больше, чем в boxplot.

  В R это реализуется с помощью qplot(gear, mpg, data=mtcars,geom=c("violin"),fill=factor(gear))*

  Чтобы понять как это график соотносится с диаграммой размаха, построим их вместе qplot(gear, mpg, data=mtcars, geom=c("violin","boxplot"), fill=factor(gear))*

Двумерные массивы данных

- Диаграмма рассеивания (scatterplot)

  Scatterplot — это просто двумерный точечный график. В R его можно создать просто функцией plot(x,y), но более изящно выглядит qplot(disp, mpg, data=mtcars, geom="point",col=factor(gear))*

  В этом случае geom можно и не указывать

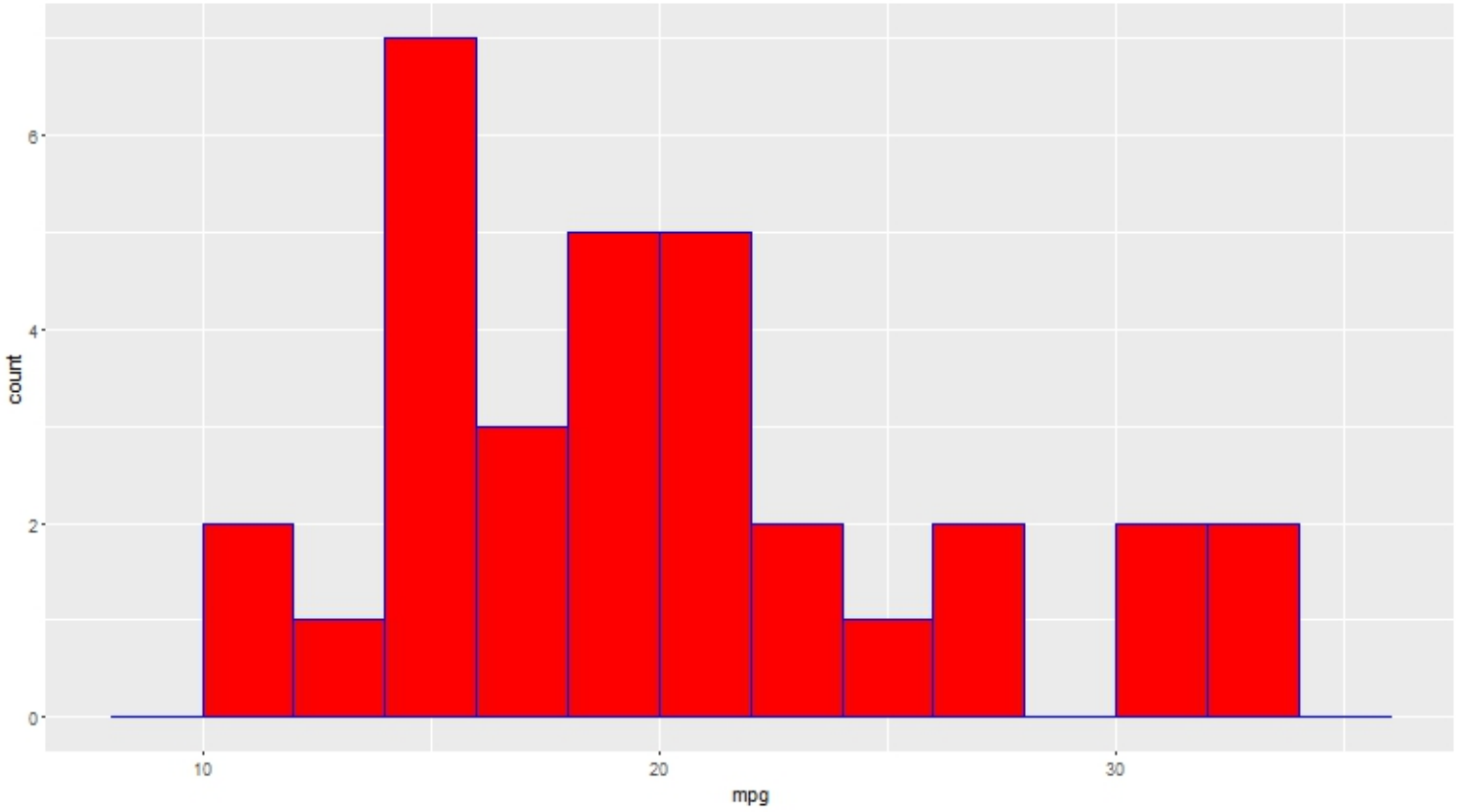

Рис. 3: Histogram

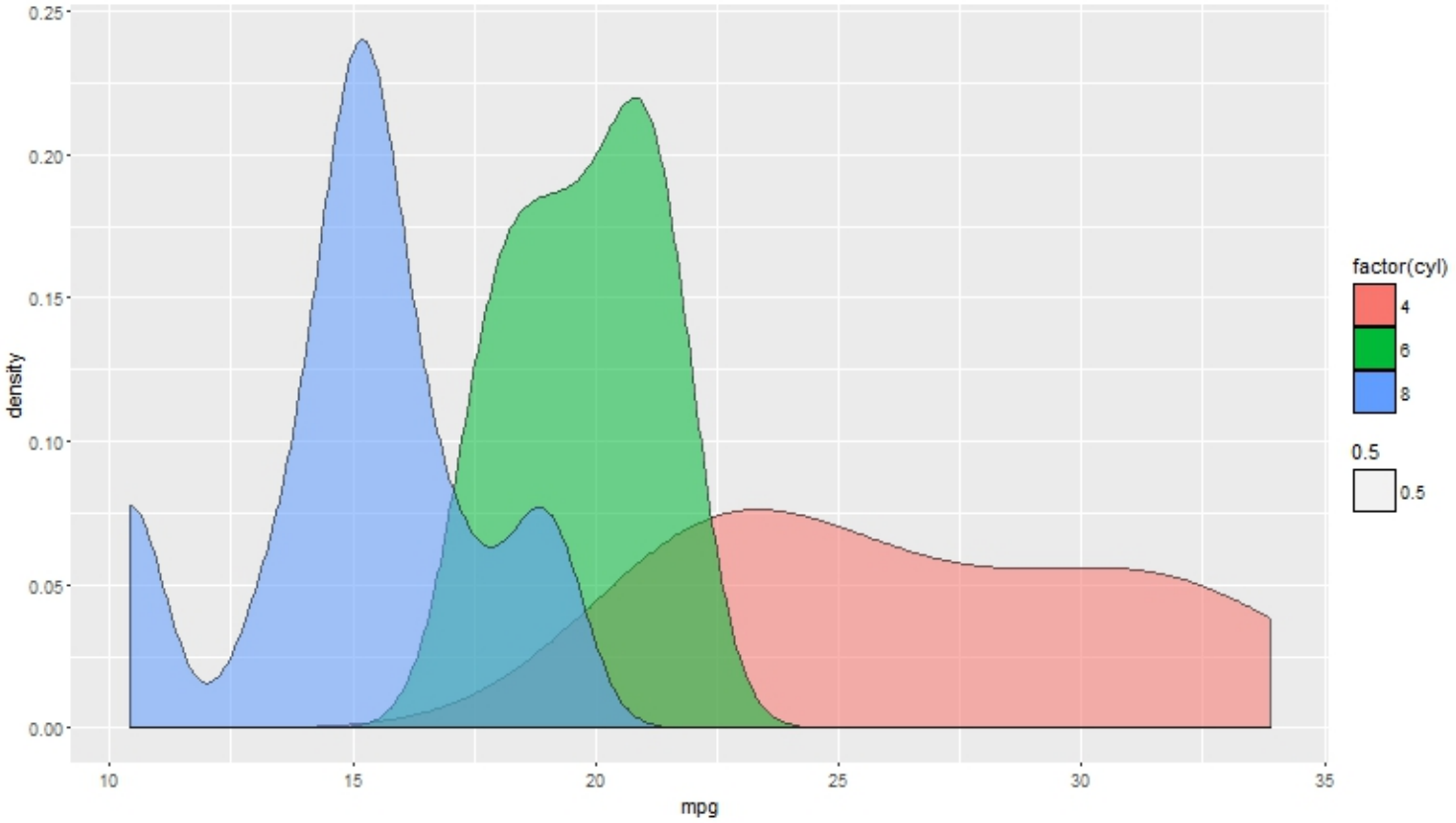

Рис. 4: Density

- Диаграмма размаха с выделением

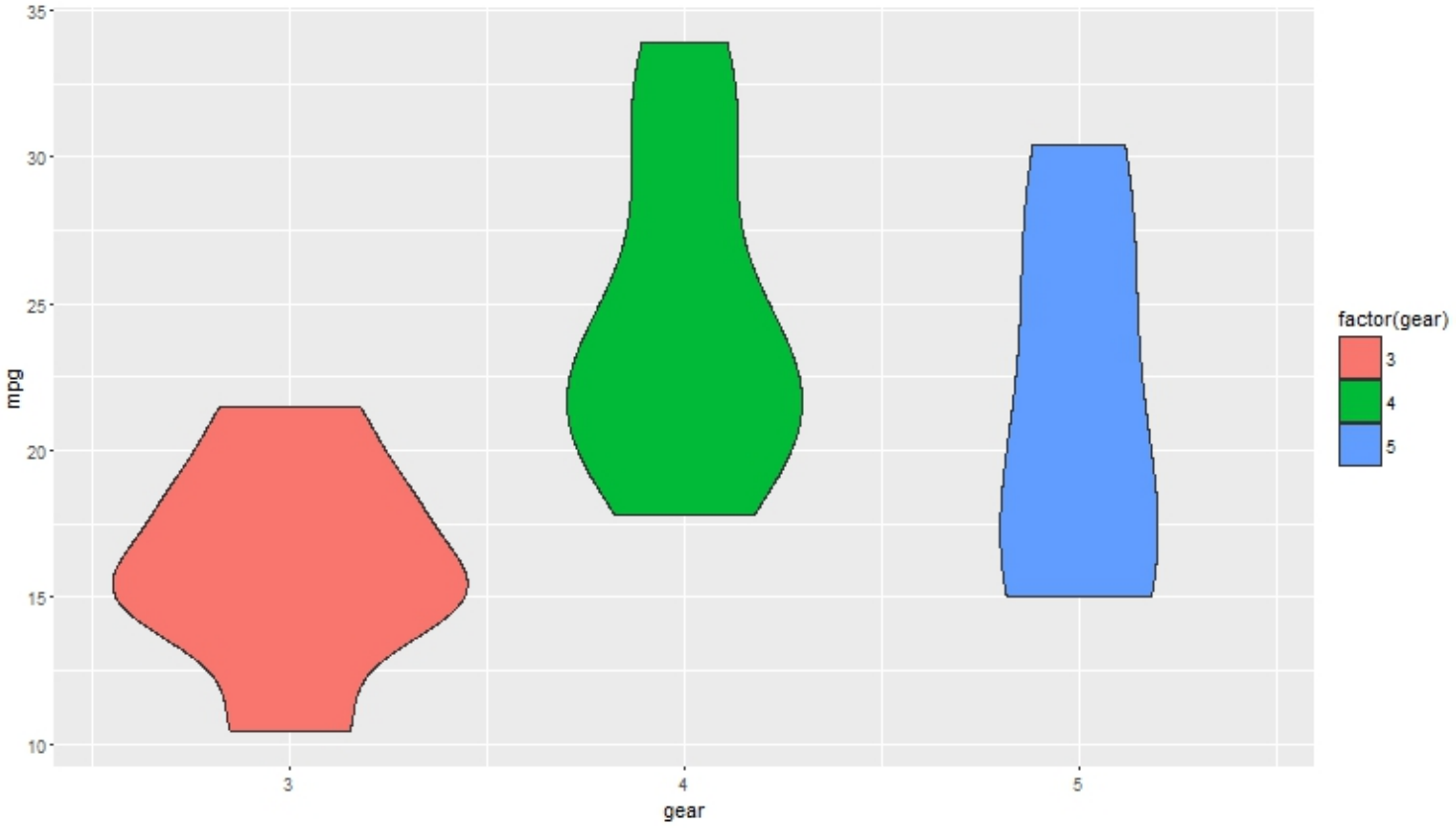

Рис. 5: Violin

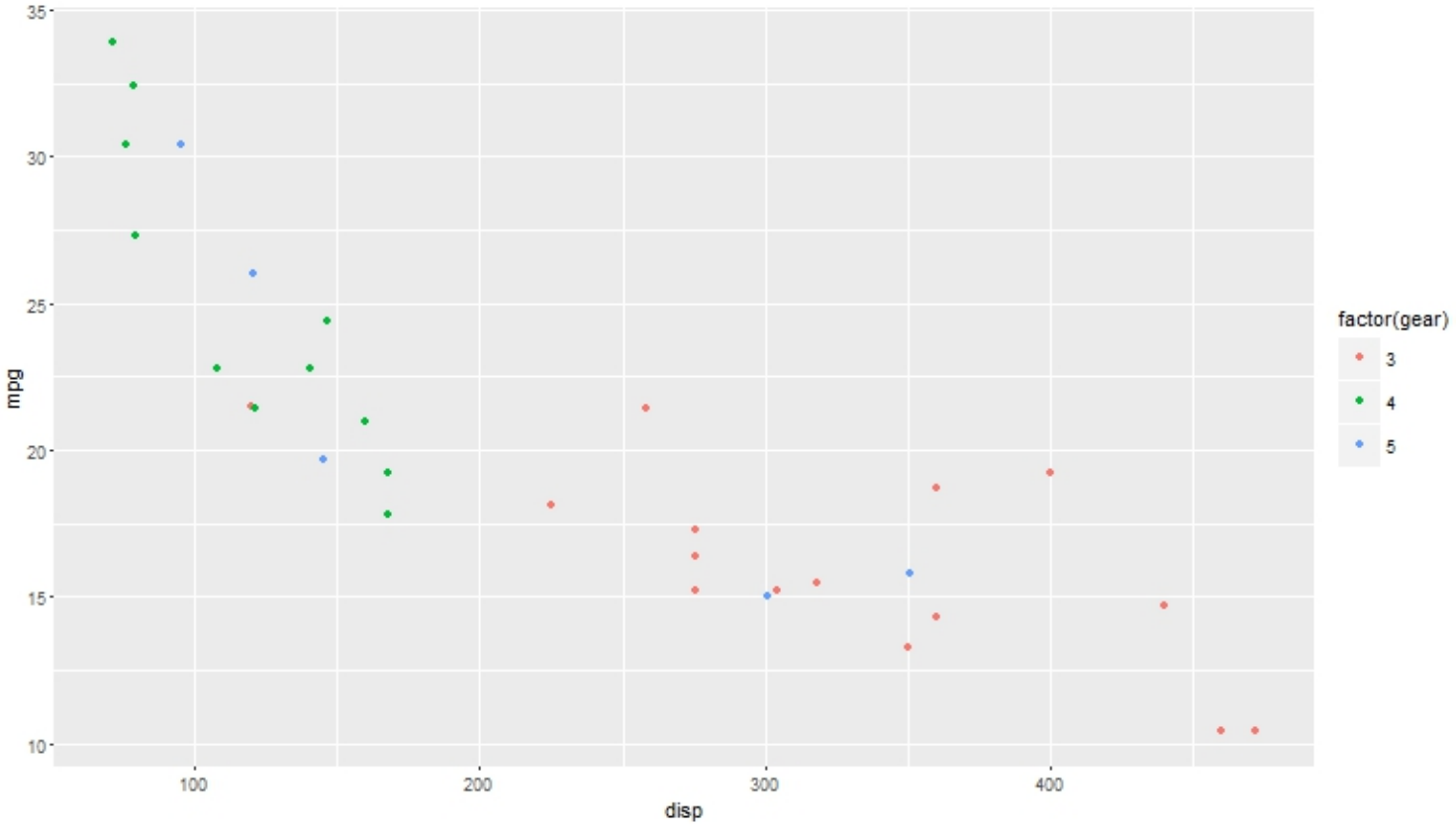

Рис. 6: Scatterplot

Отобразить вторую переменную можно и на одномерных графиках с помощью выделения цветом. Мы уже видели такого рода графики в случае дискретных вторых переменных, но это возможно и в непрерывном случае

Скажем, qplot(disp, mpg, data=mtcars, geom="violin",fill=factor(gear))* расположит наши графики плотности для mpg в соответствии со значением disp. Аналогично можно сделать и для boxplot

Трехмерные массивы данных

- Трехмерный график
  Функция plot3d из пакета rgl построит трехмерный график
  plot3d(mtcars$mpg,mtcars$disp,mtcars$hp)*
- Диаграмма рассеивания с дополнительным выделением

  Можно использовать для такой визуализации scatterplot, добавляя третью размерность через цвет или размер
  qplot(disp, mpg, data=mtcars,color=factor(hp))
  qplot(disp, mpg, data=mtcars,size=factor(hp)).

  Таким образом, можно отобразить до 5 переменных
  qplot(disp, mpg, data=mtcars,size=factor(hp),col=factor(gear),shape=factor(am))

Многомерные массивы данных

- Матрица диаграмм рассеивания (scatterplot matrix)
  Матрица диаграм рассеивания — это набор scatterplots, построенный по плоскостям, соответствующим каждой паре переменных.

  В R она строится с помощью pairs(∼ mpg+disp+drat+wt,data=mtcars,main="Simple Scatterplot Matrix",col=mtcars$cyl)*

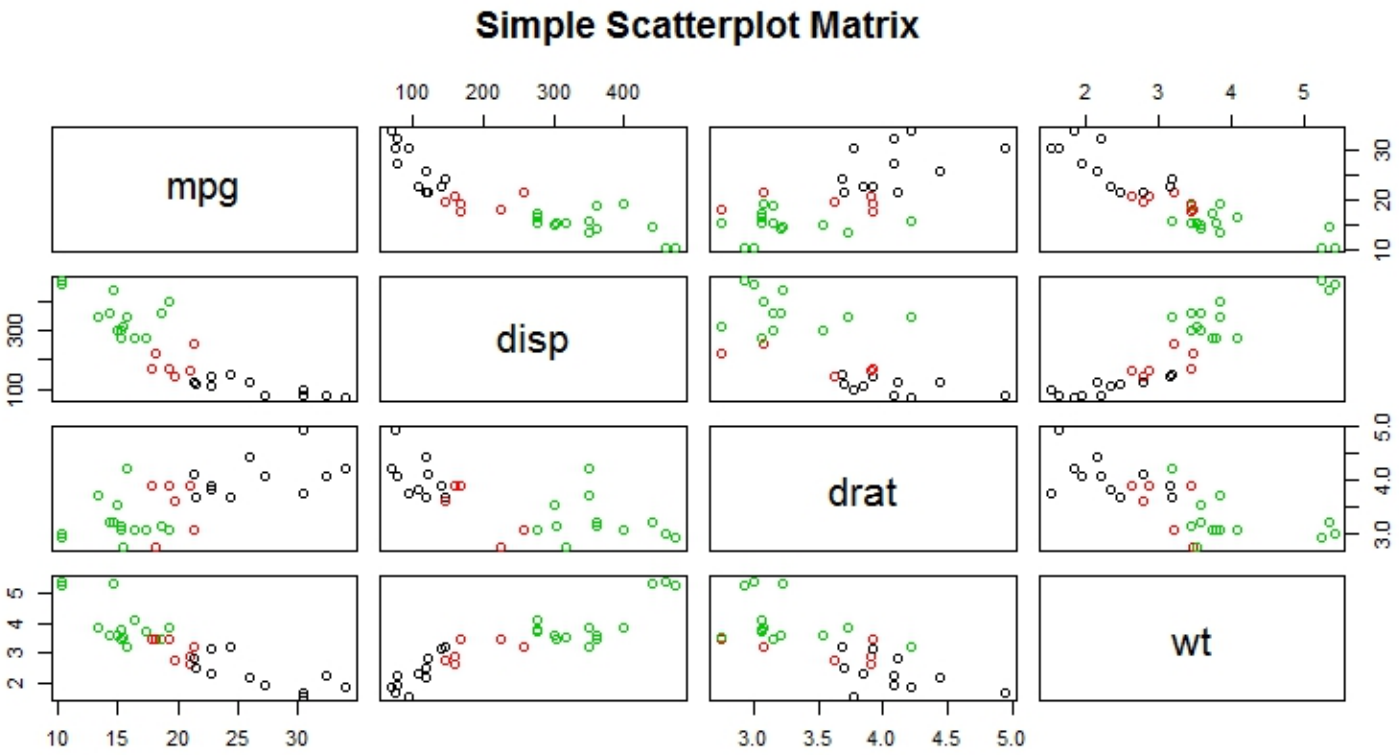

Рис. 7: Scatterplot matrix

- Кривые Эндрюса (Andrews plot)
  Кривая Эндрюса строит по точке $x_1, ..., x_d$ отрезок ряда Фурье

$$f(t) = \frac{x_1}{\sqrt{2}} + x_2 \cos t + x_3 \sin t + x_4 \cos(2t) + ...$$

и заменяет точку на такого рода кривую. Если размерность не слишком велика, то существенно разные коэффициенты при одной гармонике будут заметны.

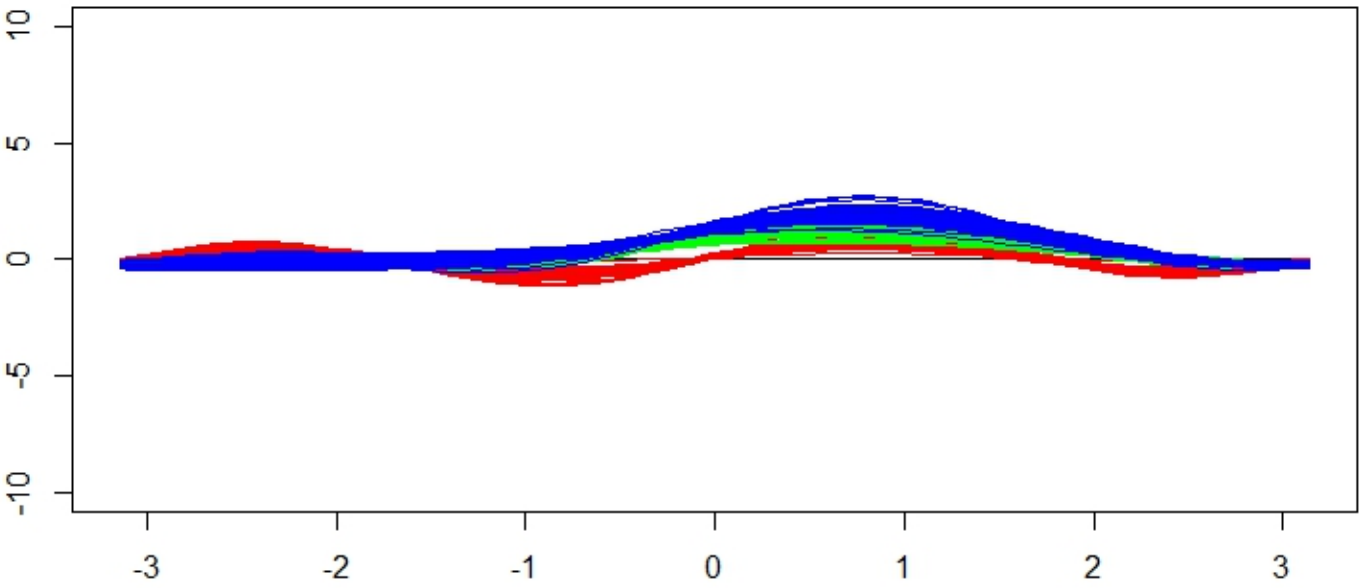

Рис. 8: Andrews Plot

Реализована она, например, функцией andrews пакета andrews с синтаксисом andrews(iris,clr=5), где clr — номер столбца, отвечающего за цвет кривой.

Глядя на картинку мы видим, что красный цвет заметно отделяется от двух остальных, соответствующий первой переменной (setosa).

Не так просто понять, какое соответствие переменной и цвета, поэтому используем brewer.pal(3, "Set1")[iris$Species]* пакета ColorRBrewer. Тем самым мы создали вектор цветов, соотвествующую переменной, отвечающей за вид. Сравнивая переменные Species и полученного вектора, видим, что E41A1C (красный) соответствует setosa, 4DAF4A (зеленый) — virginica, 377EB8 (синий) — versicolor.

Для большого числа данных такого рода визуализация слишком громоздка, но для нескольких десятков работает.

- Параллельные координаты

  Параллельные координаты предлагаются заменить данные $x_1, ..., x_d$ на ломаную с вершинами $(1, x_1)$,...,(d, x_d). Если размерность не слишком велика или данные хорошо сгруппированы, то мы сможем визуально отличить качественное отличие нескольких групп таких ломаных

  В R реализацию предлагает, например, функция parcoord пакета MASS.

  Пример ее использования:
  parcoord(iris[,c(3,4,2,1)],col=iris[,5])

  При этом цвет задает вектором col, а сами данные — массивом iris.

- Радарная диаграмма (spider plot, radar chart, star plot, web plot)
  В этом виде диаграмм мы строим аналог параллельных координат, но замкнутый по кругу. Иначе говоря, оси, соответствующие нашим переменным, расположены через равные углы по кругу, каждой точке соответствует многоугольник.

  В R есть стандартная функция stars, реализующая эту процедуру в виде stars(mtcars[,1:11]).

  В этом случае мы получим матрицу радарных диаграмм

  Иногда удобнее построить их на одном графике, для этого можно использовать chartJSRadar из пакета library(radarchart). Примером может служить chartJSRadar(skills). Стоит обратить внимание, что эта функция почему-то оперирует с транспонированным относительно обычного положения массивом — записи по столбцам, переменные по строкам. Транспонировать data.frame можно с помощью as.data.frame(t())

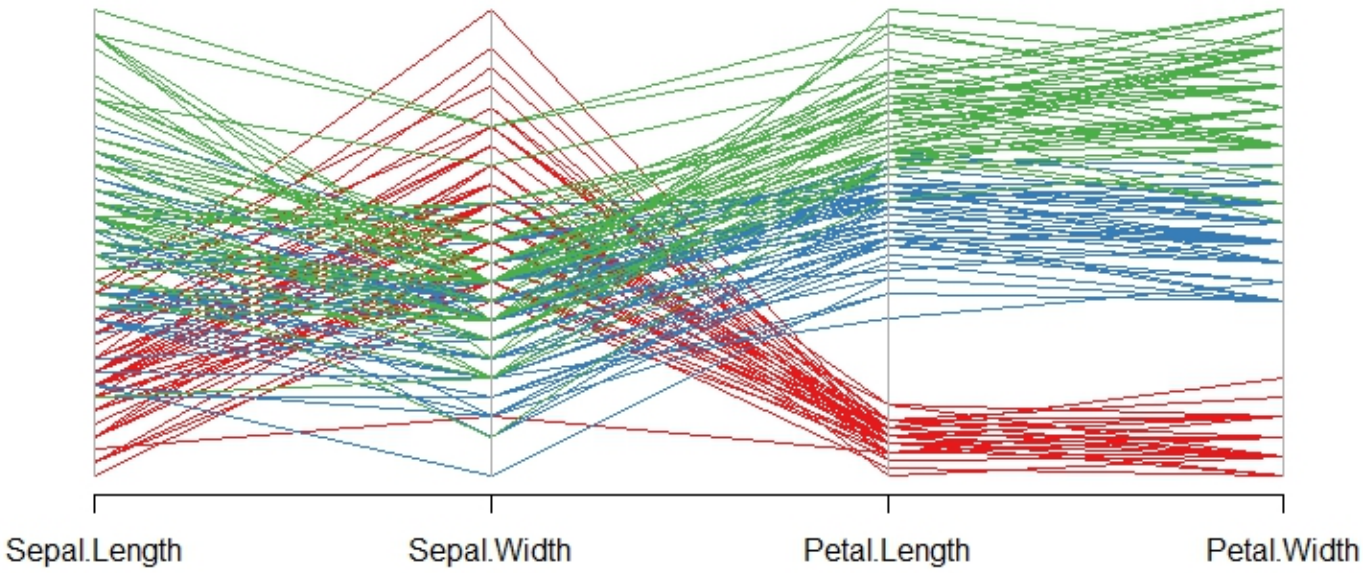

Рис. 9: Parallel Curves

- Лица Чернова
  Лица Чернова, предложенные замечательным математиком Черновым, могут показаться забавной шуткой, но тем не менее достаточно эффективны.

  Идея заключается в том, чтобы для каждой точки нарисовать лицо, различные параметры которого соответствуют значениям координат нашей точки.

  Реализация этого алгоритма есть в библиотеке TeachingDemos в функции faces

  Применение достаточно просто:
  faces(mtcars[,1:10])

  В том же пакете есть аналогичная функция faces2 с другой системой кодирования лиц

  Система кодирования лица по переменным указана в описании соответствующих функций.

Туры

Тур представляет собой динамическую визуализацию данных, устроенную следующим образом: 1) Мы фиксируем некоторую размерность d, выбираем подпространство размерности d и визуализируем его каким-либо образом.
2) Начинаем по некоторому принципу менять выбранное подпространство, каждый раз визуализируя его.

Наиболее естественный пример — мы можем изображать многомерные данные в некоторой проекции и если вращать плоскость проекции, то мы совершим путешествие по нашим многомерным данным, которое и называется туром.

В зависимости от метода выбора направления смены пространств используют:
grand tour, соответствующий случайной смене пространств,
guided tour, соответствующий смене пространств в направлении увеличения некоторого параметра.
little tour поочередно проводит нас по всем d-мерным пространствам, натянутым на какие-либо d координат, поворотом двигаясь из одного в другое.

В R для такого вида визуализации есть удобный пакет tourr, к которому может также понадобиться пакет ash.

В нем мы можем построить тур с помощью команд grand_tour(d=), guided_tour(index_f = , d=,) или little_tour(d=). Здесь index_f задает параметр оптимизации, который может принимать значения holes (holes index, измеряющий степень разреженности данных), cmass (central mass index, измеряющий степень сконцентрированности данных), lda_pp(col) (LDA projection pursuit index, измеряющий линейную отделимость точек разных типов (типы задаются вектором col)), pda_pp(col) (PDA projection pursuit index, другой подобный индекс).

Построим тур t мы можем теперь визуализировать его каким-либо методом, в зависимости от размерности d=1. У всех указанных методов визуализации есть параметры aps = 1, fps = 30, max_frames = Inf, rescale = TRUE (указаны их

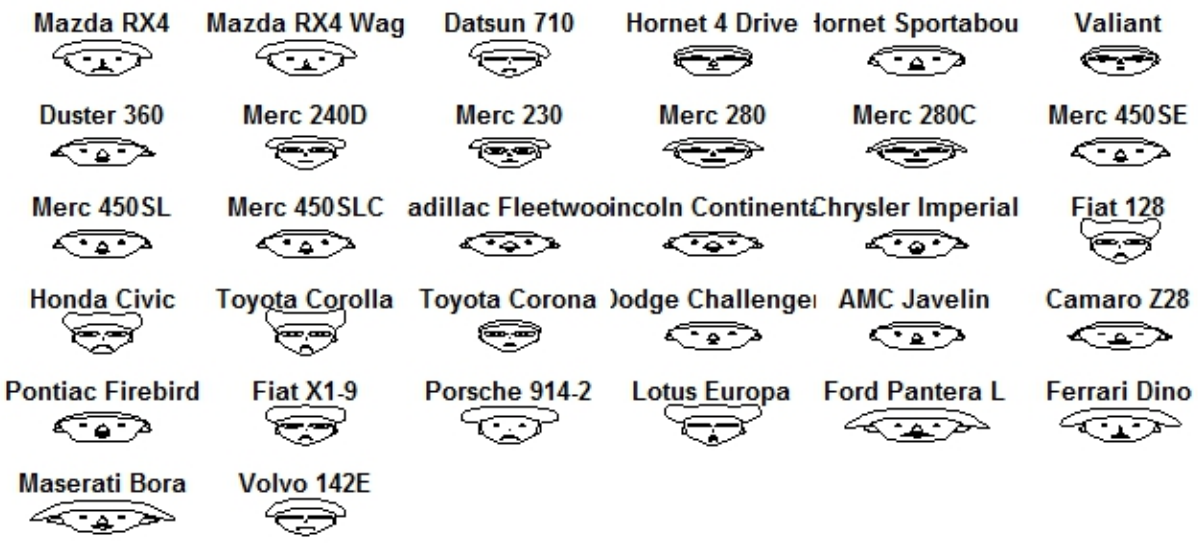

Рис. 10: Chernoff Faces

значения по умолчанию), отвечающие за шаг изменения угла, количество кадров в секунду, максимальное число показанных кадров и нормировку данных. Кроме того полезна переменная col, позволяющая окрасить данные в соответствии с вектором col.

- Одномерная визуализация осуществляется с помощью функции animate_dist(data = , method = , tour_path = t). Параметром method может служить "hist"или "density"в зависимости от способа визуализации (гистограмма или оценка плотности).
- Двумерная визуализация осуществляется с помощью animate_xy(data=,tour_path=t)
- Трехмерная визуализация осуществляется с помощью animate_depth(data=,tour_path=t)
- Многомерная визуализация осуществляется с помощью
  animate_pcp(data=,tour_path=t) (параллельные координаты),
  animate_faces(data=,tour_path=t) (лица Чернова)
  animate_stars(data=,tour_path=t) (звезды)
  animate_scatmat (матрицы диаграмм рассеивания)

Итоговый формат вызова функции выглядит так:
animate_xy(data=mtcars,tour_path=grand_tour(d=2)),*
animate_faces(data=mtcars,tour_path=guided_tour(d=4,index_f=holes))*