Графическая визуализация в ggplot2

Зачем?

Графический интерфейс базовых графических пакетов R достаточно проста и невзрачна. В пакете ggplot2 находится ряд графических решений, позволяющих делать более зрелищную визуализацию. ggvis позволяет делать интерактивные графики. К сожалению, базовый R не поддерживает такие объекты, поэтому для этого нужна среда визуализации. Одну из таких сред предоставляет RStudio. Вашему вниманию предлагается обзор основных функций ggplot2.

Графики qplot

Аналог базовой функции в ggplot2 носит название qplot и имеет параметры qplot(x, y, data=, color=, shape=, size=, alpha=, geom=, facets= xlim=, ylim= xlab=, ylab=, main=, sub=).
Эта функция позволяет строить как обычные scatterplot, так и гистограммы и boxplot.
Аргументы x,y, как обычно, это переменные, отложенные по осям. Второй аргумент может быть опущен, если нам нужны линейные данные (например, для гистограммы).
data обозначется data frame, на основе которого строится график. Указав в его качестве какой-либо data frame, мы можем обращаться в остальных переменных к его столбцам без указания. Например qplot(Petal.Width,Sepal.Width, data = iris) построит график iris$Petal.Width от iris$Sepal.Width.
Как и прежде color отвечает за цвет линий. В случае заполненных графиков, например, гистограмм, вместо него можно указать fill = , что приведет к заполнению соответствующим цветом. Как и прежде, мы можем использовать в качестве color массивы, так, например, qplot(Petal.Width,Sepal.Width, data = iris, color = Species) построит график с разноветными видам ириса.
Shape задает форму точек на графике, size — их размер. Скажем, qplot(Petal.Width,Sepal.Width, data = iris, color = Species, shape = (Petal.Length>1.6), size = (Sepal.Length>5)) выделит различными цветами разные виды, а треугольной формой цветки с длинными лепестками, а большим размером цветы с крупной чашечкой.
alpha задает прозрачность накладывающихся участков графика.
Параметр geom задает формат графика — "point"(обычный точечный график), "line"(линейный),

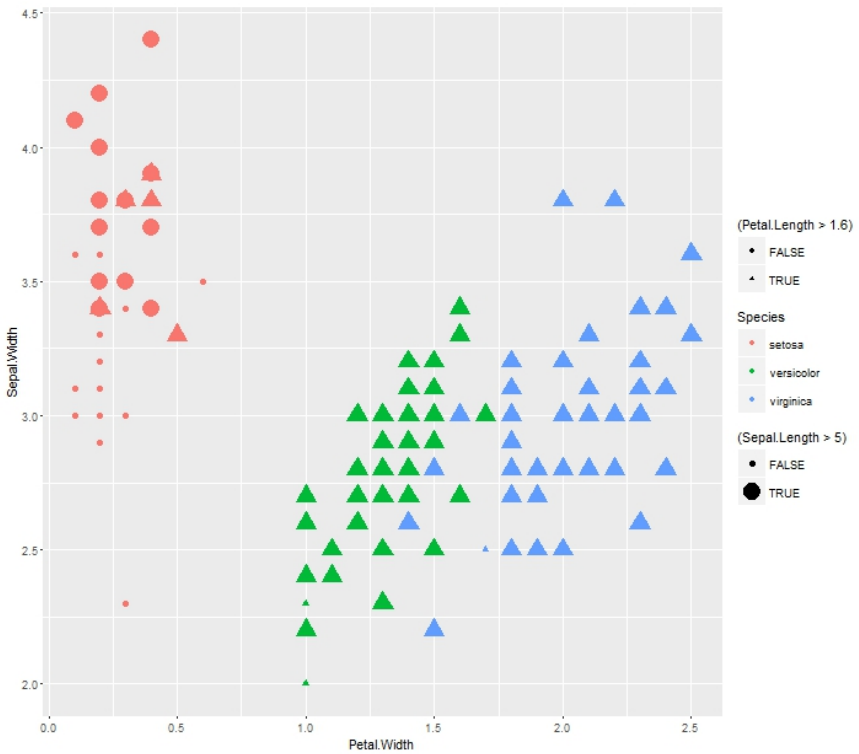

Рис. 1: График qplot для iris с различными формами и цветами точек в зависимости от их показателей

"smooth"(сглаженный), "boxplot"(box график), "dotplot "histogram"(гистограммы с разными формами визуализации), "density"(ядерная оценка плотности на основе выборки), "jitter"(график, в который

добавлен небольшой случайный шум, размывающий облако точек, к примеру, в дискретном случае), "polygon"(заполненный многоугольник).
facets задает дополнительные переменные, итогом qplot будет массив графиков со всеми возможными значениями вспомогательных переменных.
xlab, ylab задают названия осей, xlim, ylim — границы изменения осей, main, sub — титул графика.
Пример 1. Несколько примеров:
qplot(mpg,data=mtcars, geom = "density fill=factor(gear),alpha = 0.5) — плотность параметра mtg (миль на галлон топлива) для разных значений gear.
qplot(mpg,data=mtcars, geom = "histogram fill=gear, color = "black alpha = 0.5,bins = 40) — гистограмма с разбиением на 40 фрагментов, на которой цветом отображено количество данных с каждым значением gear (передача).
qplot(hp, mpg, data=mtcars, color=am, facets=gear∼cyl, size=I(3)) —график mtg (миль на галлон топлива) от hp (мощности в лошадиных силах) при каждой паре значений количестве передачи и числа цилиндров, где цвет отражает am — ручная или автоматическая коробка передач.
qplot(gear, mpg, data=mtcars,geom = c("jitter "point")) — график mpg от gear, где к переменным добавлен шум, что позволяет меньше зависеть от дискретизации.

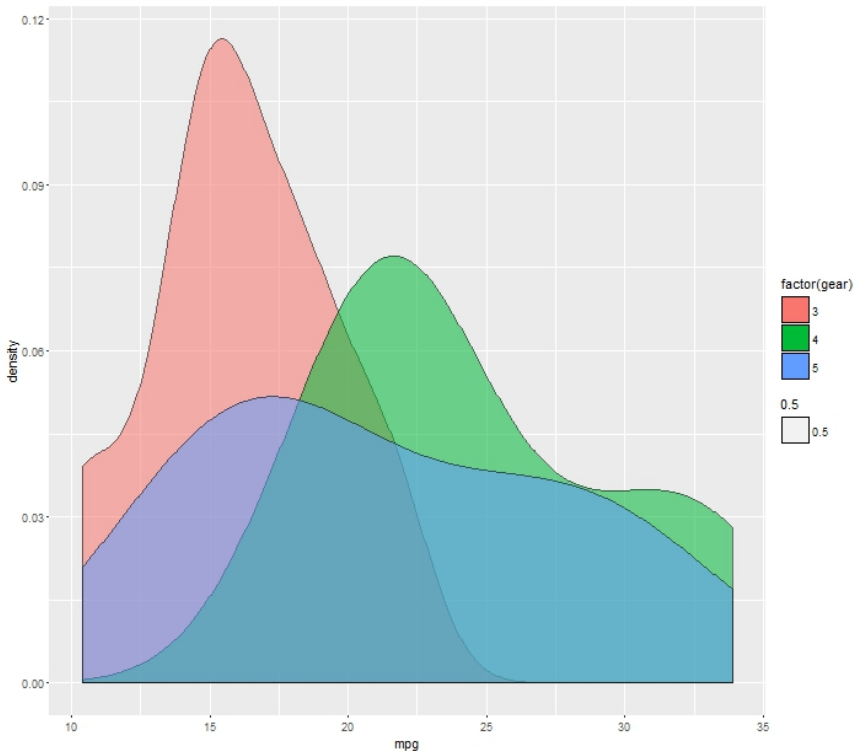

Рис. 2: Плотности mtcars для разных значений переменной gear

Графики ggplot

Другой функцией для задания графика является ggplot. В ggplot задается аргумент data (данные, с которыми мы будем работать) и заданное для построения отображение (его можно создать с помощью функции aes()). Функция aes(x,y) задает формат графика — данные по осям, а также цвет, форму и другие параметры. Кроме того, после ggplot используется функция geom_*(), задающая геометрию графика. Например, geom_point() задаст точечный график.
Пример 2. ggplot(mtcars, aes(mpg,hp,color = gear))+geom_point() создает точечный график hp от mpg с расцветкой, зависящей от gear.
ggplot(mtcars, aes(gear,cyl,fill = gear))+geom_bar(stat = "identity") создает прямоугольный график cyl от gear.
ggplot(mtcars, aes(gear)) + geom_bar(stat="count") создает ряд значений для величины gear.
ggplot(mtcars, aes(mpg,hp,color = gear))+geom_point(size = 3) + geom_line(color = "black linetype

= "dashed")+expand_limits(y=0) создает график hp от mpg, geom_point задает параметры точек, geom_line — параметры линий, expand_limits — границы по y (по умолчанию они были бы в пределах изменения параметра). ggplot(mtcars, aes(mpg,hp,color = gear,group=gear))+geom_point(size = 3) + geom_line(color = "black linetype = "dashed") задает линейные графики для каждого значения переменной gear.

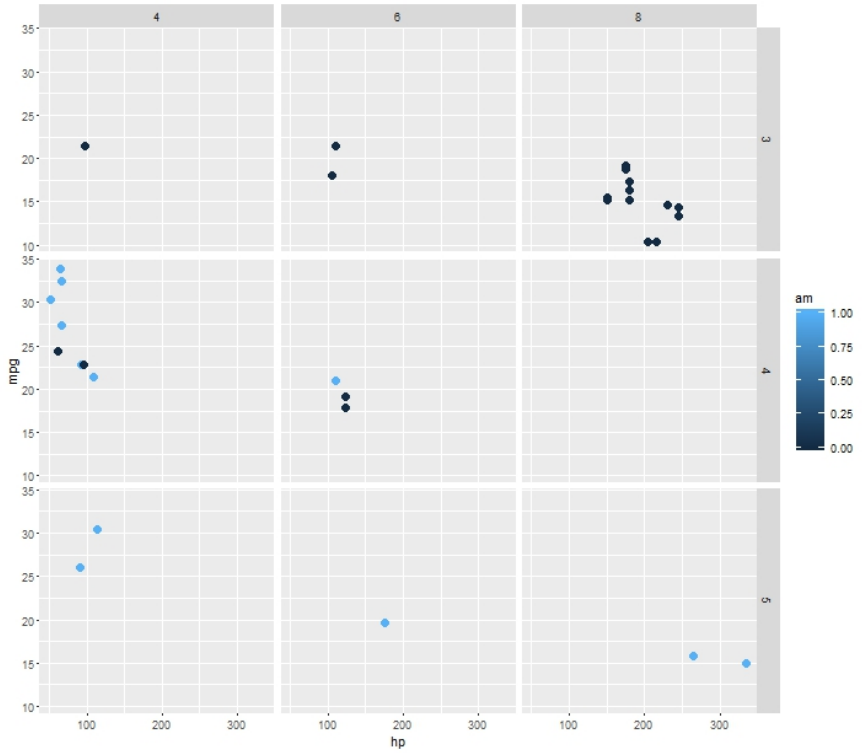

Рис. 3: Графики qplot для зависимости mpg от hp с facets

Интерактивные графики ggvis

Для этого нам понадобятся библиотеки ggvis, shiny, dplyr и ряд вспомогательных библиотек.
Интерфейс достаточно прост. Рассмотрим, например, построение ядерной оценки плотности для переменной wt базы mtcars:
mtcars %>% ggvis(∼wt) %>% layer_densities(adjust = input_slider(0.1, 3, value = 1))
mtcars описывает выбранную базу, ∼wt — переменную графика, layer_densities — геометрию графика (построение плотности), а input_slider позволяет не задавать напрямую шаг ядерного сглаживания, а меня его с помощью слайдера на экране. Меняя положение слайдера мы можем сделать оценку удовлетворительной. Кроме input_slider имеются и другие способы задания параметров: input_checkbox(), input_checkboxgroup(), input_numeric(), input_radiobuttons(), input_select(), input_text().

Пример 3. mtcars %>% ggvis(∼wt) %>% layer_densities(adjust = input_slider(0.1, 2, value = 1, step = 0.1, label = "Bandwidth adjustment"), kernel = input_select(c("Gaussian" = "gaussian", "Epanechnikov" = "epanechnikov", "Rectangular" = "rectangular", "Triangular" = "triangular", "Biweight" = "biweight", "Cosine" = "cosine", "Optcosine" = "optcosine"),label = "Kernel"))

Можно использовать в input_slider параметр map, преобразующий введенный пользователем параметр по заданной формуле, например, mtcars %>% ggvis(∼wt) %>% layer_densities(adjust = input_slider(1, 20, value = 10,map = function(x) x/10))
Если вы хотите привязать один slider к нескольким графикам, то можете сперва положить input_slider в некоторую переменную, а затем использовать ее в нескольких графиках.

Полезной функцией является также add_tooltip. Скажем,
mtcars %>% ggvis(∼wt, ∼mpg) %>% layer_points() %>% add_tooltip(function(df) c(df$wt, " ", df$mpg))
позволит при наведении на точку увидеть ее координаты.

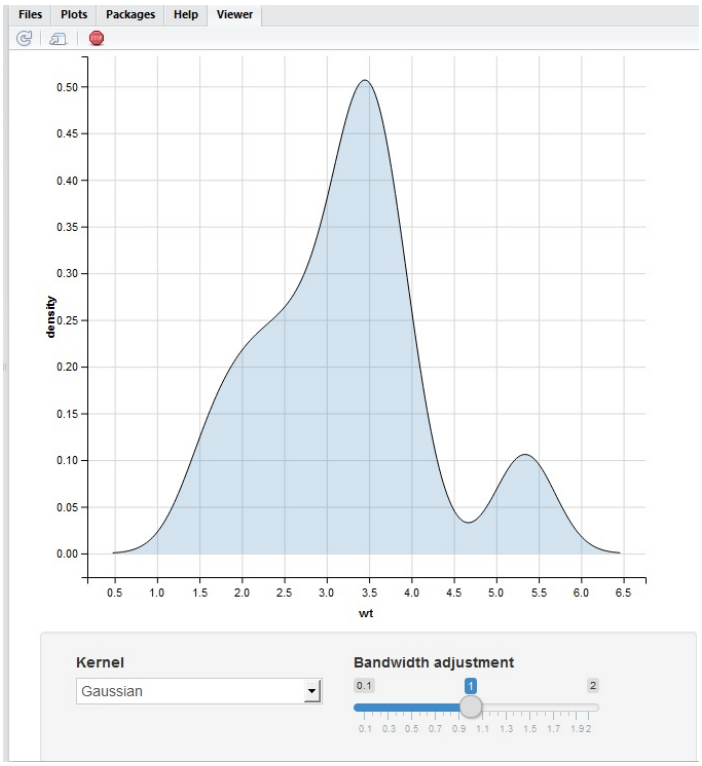

Рис. 4: Интерактивный график плотности с возможностью изменения типа ядра и шага