

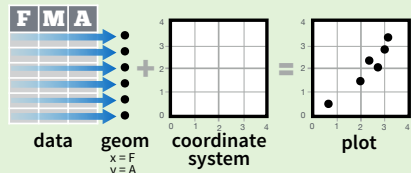
Визуализация данных при помощи ggplot2

Шпаргалка

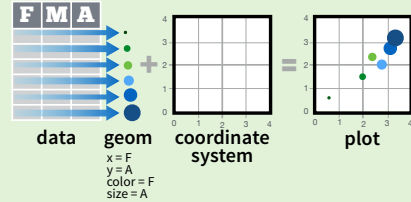


ОСНОВЫ

ggplot2 основан на "графической грамматике" — идее, согласно которой любой график можно построить из одних и тех же составляющих: массива данных (data), системы координат и геометрических объектов (geom) — визуальных маркеров, отражающих частные значения наблюдений (точки, линии, столбцы и т.д.).



Значения наносятся на график при помощи визуальных свойств геометрического объекта (aesthetics), таких как размер, цвет, координаты x и y.



Чтобы построить график заполните этот шаблон:

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION> (  
    mapping = aes(<MAPPINGS>),  
    stat = <STAT>,  
    position = <POSITION>  
  ) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

Обязательно

Не обязательно, подставляет значения по умолчанию

ggplot(data = mpg, aes(x = cty, y = hwy))

Выводит пустой график, на который слой за слоем добавляются геометрические объекты.

визуальная схема

данные

геометрия

qplot(x = cty, y = hwy, data = mpg, geom = "point")

Выводит полноценный график, с заданными вами данными, геометрией и разметкой. Множество полезных значений подставляются по умолчанию.

last_plot() Возвращает последний график.

ggsave("plot.png", width = 5, height = 5)

Сохраняет последний график в рабочий каталог как файл размером 5x5 дюймов с именем "plot.png". Сопоставляет формат файла с расширением.

Geoms - Используйте функцию geom чтобы нанести частные значения. Чтобы отобразить переменные используйте визуальные свойства.

Графические основы

```
a <- ggplot(economics, aes(date, unemploy))
```

```
b <- ggplot(seals, aes(x = long, y = lat))
```

a + geom_blank()

(Полезно для расширения границ)

b + geom_curve(aes(yend = lat + 1,

xend=long+1,curvature=z)) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend="butt",

linejoin="round", linemitre=1) - x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(group = group))

x, y, alpha, color, fill, group, linetype, size

b + geom_rect(aes(xmin = long, ymin=lat,

xmax=long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin=unemploy - 900,

ymax=unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

Отрезки

Визуальные свойства: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept=0, slope=1))

b + geom_hline(aes(yintercept = lat))

b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend=lat+1, xend=long+1))

b + geom_spoke(aes(angle = 1:115, radius = 1))

Одна переменная

Непрерывная

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
```

c + geom_area(stat = "bin")

x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")

x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()

x, y, alpha, color, fill

c + geom_freqpoly()

x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5)

x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy))

x, y, alpha, color, fill, linetype, size, weight

Дискретная

```
d <- ggplot(mpg, aes(fl))
```

d + geom_bar()

x, alpha, color, fill, linetype, size, weight

Две переменные

Непрерывная X, непрерывная Y

```
e <- ggplot(mpg, aes(cty, hwy))
```

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)

x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter(height = 2, width = 2)

x, y, alpha, color, fill, shape, size

e + geom_point()

x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()

x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl")

x, y, alpha, color, linetype, size

e + geom_smooth(method = lm)

x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)

x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Дискретная X, непрерывная Y

```
f <- ggplot(mpg, aes(class, hwy))
```

f + geom_col()

x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()

x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y",

stackdir = "center") - x, y, alpha, color, fill, group

f + geom_violin(scale = "area")

x, y, alpha, color, fill, group, linetype, size, weight

Дискретная X, дискретная Y

```
g <- ggplot(diamonds, aes(cut, color))
```

g + geom_count()

x, y, alpha, color, fill, shape, size, stroke

Три переменные

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
```

```
l <- ggplot(seals, aes(long, lat))
```

l + geom_contour(aes(z = z))

x, y, z, alpha, colour, group, linetype, size, weight

Двумерное нормальное распределение

```
h <- ggplot(diamonds, aes(carat, price))
```

h + geom_bin2d(binwidth = c(0.25, 500))

x, y, alpha, color, fill, linetype, size, weight

h + geom_density2d()

x, y, alpha, colour, group, linetype, size

h + geom_hex()

x, y, alpha, colour, fill, size

Непрерывное отображение

```
i <- ggplot(economics, aes(date, unemploy))
```

i + geom_area()

x, y, alpha, color, fill, linetype, size

i + geom_line()

x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")

x, y, alpha, color, group, linetype, size

Визуализация ошибок

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
```

```
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
```

j + geom_crossbar(fatten = 2)

x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar()

x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)

j + geom_linerange()

x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange()

x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

Карты

```
data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
```

```
map <- map_data("state")
```

```
k <- ggplot(data, aes(fill = murder))
```

k + geom_map(aes(map_id = state), map = map) + expand_limits(x = map\$long, y = map\$lat)

map_id, alpha, color, fill, linetype, size

