

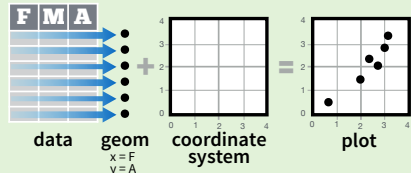
Data Visualization with ggplot2

Cheat Sheet



Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system**.



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **qplot()** or **ggplot()**

aesthetic mappings **data** **geom**

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

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

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

Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

data

```
ggplot(mpg, aes(hwy, cty)) +  
  geom_point(aes(color = cyl)) +  
  geom_smooth(method = "lm") +  
  coord_cartesian() +  
  scale_color_gradient() +  
  theme_bw()
```

add layers,
elements with +

layer = geom +
default stat +
layer specific
mappings

additional
elements

Add a new layer to a plot with a **geom_*()** or **stat_*()** function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

last_plot()

Returns the last plot

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

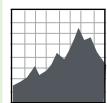
Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

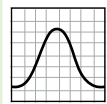
Continuous

a <- ggplot(mpg, aes(hwy))



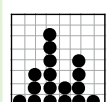
a + geom_area(stat = "bin")

x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")



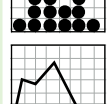
a + geom_density(kernel = "gaussian")

x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..count..))



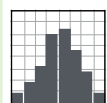
a + geom_dotplot()

x, y, alpha, color, fill



a + geom_freqpoly()

x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

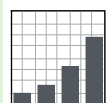


a + geom_histogram(binwidth = 5)

x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

b <- ggplot(mpg, aes(fl))

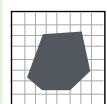


b + geom_bar()

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

Graphical Primitives

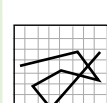
c <- ggplot(map, aes(long, lat))



c + geom_polygon(aes(group = group))

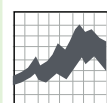
x, y, alpha, color, fill, linetype, size

d <- ggplot(economics, aes(date, unemploy))



d + geom_path(lineend="butt",
linejoin="round", linemitre=1)

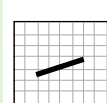
x, y, alpha, color, linetype, size



d + geom_ribbon(aes(ymin=unemploy - 900,
ymax=unemploy + 900))

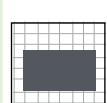
x, ymax, ymin, alpha, color, fill, linetype, size

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



e + geom_segment(aes(
xend = long + delta_long,
yend = lat + delta_lat))

x, xend, y, yend, alpha, color, linetype, size



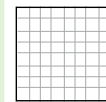
e + geom_rect(aes(xmin = long, ymin = lat,
xmax = long + delta_long,
ymax = lat + delta_lat))

xmax, xmin, ymax, ymin, alpha, color, fill,
linetype, size

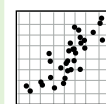
Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

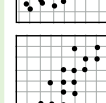


f + geom_blank()



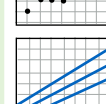
f + geom_jitter()

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



f + geom_point()

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



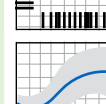
f + geom_quantile()

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



f + geom_rug(sides = "bl")

alpha, color, linetype, size



f + geom_smooth(model = lm)

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

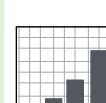


f + geom_text(aes(label = cty))

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

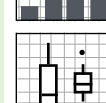
Discrete X, Continuous Y

g <- ggplot(mpg, aes(class, hwy))



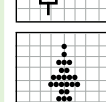
g + geom_bar(stat = "identity")

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



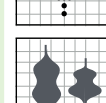
g + geom_boxplot()

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



g + geom_dotplot(binaxis = "y",
stackdir = "center")

x, y, alpha, color, fill

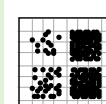


g + geom_violin(scale = "area")

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

Discrete X, Discrete Y

h <- ggplot(diamonds, aes(cut, color))



h + geom_jitter()

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

Three Variables

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

m <- ggplot(seals, aes(long, lat))



m + geom_contour(aes(z = z))

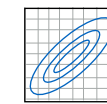
x, y, z, alpha, colour, linetype, size, weight

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))

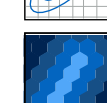


i + geom_bin2d(binwidth = c(5, 0.5))
xmax, xmin, ymax, ymin, alpha, color, fill,
linetype, size, weight



i + geom_density2d()

x, y, alpha, colour, linetype, size

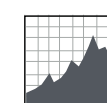


i + geom_hex()

x, y, alpha, colour, fill size

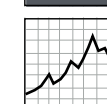
Continuous Function

j <- ggplot(economics, aes(date, unemploy))



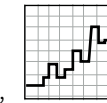
j + geom_area()

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



j + geom_line()

x, y, alpha, color, linetype, size



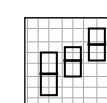
j + geom_step(direction = "hv")

x, y, alpha, color, linetype, size

Visualizing error

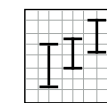
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)

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



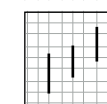
k + geom_crossbar(fatten = 2)

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



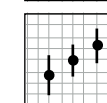
k + geom_errorbar()

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



k + geom_linerange()

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



k + geom_pointrange()

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

Maps

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

map <- map_data("state")

l <- ggplot(data, aes(fill = murder))



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

map_id, alpha, color, fill, linetype, size