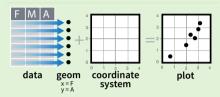
Veri Görselleştirme ggplot2

Hatırlatıcı Not

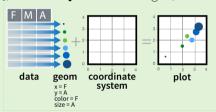


Temel Öğeler

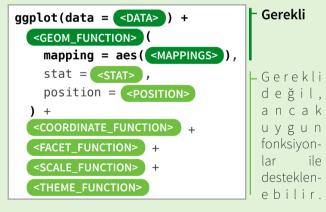
ggplot2'nin temeli "grammar of graphics"e, yani her grafiği aynı bileşenlerden oluşturma fikrine dayanmaktadır. Bunlar bir **veri** seti, **koordinat sistemi** ve **geom**lar (veri noktalarının görsel işaretler ile belirtilmesi) olabilir.



Değerleri göstermek için, verinin içindeki değiskenler geom'un görsel estetiğe dayalı özellikleri (size [boyut], color[renk], x- ve y-konumları gibi) ile eşleştirilir.



Grafik oluşturmak için aşağıdaki şablon kullanılabilir.



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

katmanlarla istenilen cizim oluşacaktır. Her katman için sadece bir geom fonksiyonu eklenmelidir.

estetik eşleştirmeler

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

Belirtilen veri, geom ve eşleştirmeler ile tam bir çizim yaratılır. Uygun fonksiyonlarla desteklenebilir.

last plot()

Sonuncu çizimi çağırır.

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

Sonuncu çizimi 5'x5' boyutunda "plot.png" ismiyle dizine kaydeder. Dosyanın türü dosya uzantısından belirtilebilir.

Temel Cizim Öğeleri

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

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

a + geom blank()

(Sınırları genişletmek için kullanışlıdır)



b + geom_curve(aes(yend = lat + 1, xend=long+1,curvature=z)) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size



geom_path(lineend="butt", linejoin="round', linemitre=1) x, y, alpha, color, group, linetype, size



+ 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

Doğru Parcaları

sık kullanılan estetikler: 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(vend=lat+1, xend=long+1))

b + geom spoke(aes(angle = 1:1155, radius = 1))

Tek Değisken

Devamli

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



c + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size



+ geom_density(kernel = "gaussian") x, y, alpha, color, fill, group, linetype, size, weight



+ geom_dotplot() x, y, alpha, color, fill



c + geom_freqpoly()

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



+ 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

Ayrık

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



geom bar()

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

İki Değisken

Devamlı X. Devamlı Y

Geomlar - Veri noktalarını belirtmek için geom fonksiyonunu, veri değişkenlerini belirtmek için geom'un estetik özelliklerini kullanınız.

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



e + geom_label(aes(label = cty), nudge_x = 1, nudge y = 1, check overlap = TRUE) x. v. label, alpha, angle, color, family, fontface. hjust, linéheight, size, vjust



e + geom_point()

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

x, v, alpha, color, fill, shape, size



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

Ayrık X, Devamlı Y

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



geom_col()

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



geom_boxplot()

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



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



x, y, alpha, color, fill, group geom violin(scale = "area")



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



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



g + geom_count()

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

Devamlı İki Değiskenli Dağılım

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



h + geom bin2d(binwidth = c(0.25, 500))x, y, alpha, color, fill, linetype, size, weight

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



h + geom density2d()



h + geom hex()

x, y, alpha, colour, fill, size

Devamlı Fonksivon

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

Hata Görselleştirmesi

df < -data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)i <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))</pre>



+ geom_crossbar(fatten = 2)

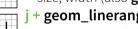
x, y, ymax, ymin, alpha, color, fill, group, linetype, size



+ geom errorbar()

linetype, shape, size

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



+ geom linerange() x, ymin, ymax, alpha, color, group, linetype, size



+ geom_pointrange() x, y, ymin, ymax, alpha, color, fill, group,

Haritalar

data <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests))) map <- map_data("state")</pre> 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

Üç Değişken

seals\$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))</pre> l <- ggplot(seals, aes(long, lat))</pre>



+ geom_contour(aes(z = z))

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



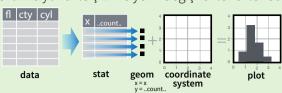
geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE) x, y, alpha, fill



+ geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size, width

Stats - Katman oluşturmanın bir başka yolu

stat fonksiyonu ile çizime yeni değişkenler eklenebilir.



Bir stat, geom fonksiyonunun varsayılan stat ifadesini değiştirerek **geom_bar(stat="count")** veya yeni bir katman oluşturmak için varsayılan geom'u çağıran stat fonksiyonunu kullanarak stat_count(geom="bar") görselleştirilebilir. Stat değişkenlerini estetik eşleştirmelere bağlamak için ..name.. söz dizimini kullanın.



geom stat fonksiyonu geom eşleştirmeleri + stat_density2d(aes(fill = ..level..), geom = "polygon") stat'ın oluşturduğu değişkenler

c + stat bin(binwidth = 1, origin = 10)

1B dağılım

x, y | ...count..., ...density..., ...ndensity...

c + stat_count(width = 1) x, y, | ...count.., ..prop.. c + stat density(adjust = 1, kernel = "gaussian")

x, y, | ...count.., ..density.., ..scaled..

2B dağılım **e + stat bin 2d(**bins = 30, drop = T**)**

x, y, fill | ..count.., ..density..

e + stat_bin_hex(bins=30) x, y, fill | ..count.., ..density..

e + stat_density_2d(contour = TRUE, n = 100) x, y, color, size | ..level..

e + stat_ellipse(level = 0.95, segments = 51, type = "t")

l + stat_contour(aes(z = z)**)** x, y, z, order | ..level..

 $l + stat_summary_hex(aes(z = z), bins = 30, fun = max)$ x, y, z, fill | ..value..

 $l + stat_summary_2d(aes(z = z), bins = 30, fun = mean)$ x, y, z, fill | ..value.. Üç değişken

f + **stat boxplot(**coef = 1.5)

Karşılaştırmalar x, y | ..lower.., ..middle.., ..upper.., ..width.., ..ymin.., ..ymax..

f + stat_ydensity(kernel = "gaussian", scale = "area") x, y | ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..

Fonksiyonlar e + stat_ecdf(n = 40) **x, y** | ..x.., ..y..

 $e + stat_quantile(quantiles = c(0.1, 0.9),$

formula = $y \sim log(x)$, method = "rq") x, y | ..quantile..

 $e + stat_smooth(method = "lm", formula = y \sim x,$ se=T, level=0.95) x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax..

ggplot() + stat_function(aes(x = -3:3), n = 99,

fun = dnorm, args = list(sd=0.5)) x | ..x.., ..y..

e + stat_identity(na.rm = TRUE)

ggplot() + stat_qq(aes(sample=1:100), dist = qt, dparam=list(df=5)) sample, x, y | ...sample..., ..theoretical..

e + stat_sum() x, y, size | ..n.., ..prop..

e + stat_summary(fun.data = "mean_cl_boot")

h + stat_summary_bin(fun.y = "mean", geom = "bar")

e + stat_unique()

Genel Amaclı

Ölcek (Scales)

Ölçek (scales) veri değerlerini estetiğin görsel değerleriyle eşleştirir. Bunu değiştirmek için yeni bir ölçek ekleyin.



limits = c("d", "e", "p", "r"), breaks =c("d", "e", "p", "r"), name = "fuel", labels = c("D", "E", "P", "R"))

lejant ya da ksen etiketleri

lejant ya da eksen kesmele

Genel amaçlı ölçekler

Çoğu estetik ile kullanılabilir

scale_*_continuous() - devamlı değerleri görsel olarak

scale_*_discrete() - ayrık değerleri görsel olarak scale_*_identity() - veri değerlerini görsel olarak

scale_*_manual(values = c()) - ayrık değerleri manual olarak seçilmiş görsel olarak

scale_*_date(date_labels = "%m/%d"), date breaks = "2 weeks") - veri değerleri tarih olarak.

scale_*_datetime() - verinin x değerleri tarih/zaman scale_x_date() gibi aynı argümanları kullan. olarak. Etiketleme formatı için ?strptime bak.

X ve Y konum ölcekleri

x ya da y estetikleri ile kullanın (x burada gösteriliyor)

scale_x_log10() - x'i log10 ölçeğinde çizer scale_x_reverse() - x ekseninin yönünü çevirir scale_x_sqrt() - x'i karekök ölçeğinde çizer

Color ve fill ölcekleri (Avrık) n <- d + geom bar(aes(fill = fl))



+ scale_fill_brewer(palette = "Blues") Palet seçenekleri için: RColorBrewer::display.brewer.all()

n + scale_fill_grey(start = 0.2, end = 0.8, na.value = "red")

Color ve fill ölçekleri (Devamlı)

o <- c + geom dotplot(aes(fill = ..x..))



o + scale_fill_distiller(palette = "Blues")

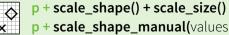
o + scale_fill_gradient(low="red", high="yellow")

o + scale_fill_gradient2(low="red", high="blue", mid = "white", midpoint = 25)

o + scale_fill_gradientn(colours=topo.colors(6)) Also: rainbow(), heat.colors(), terrain.colors(), cm.colors(), RColorBrewer::brewer.pal()

Shape ve size ölçekleri

p <- e + geom_point(aes(shape = fl, size = cyl))



p + scale shape manual(values = c(3:7))

- . (/ 0 1 \$* \$\$ \$' \$, \$- \$. \$(\$/ \$0 \$1 '* '\$ " ', '- '. $\square \circ \triangle + \times \Diamond \nabla \boxtimes \# \oplus \oplus \boxtimes \boxtimes \boxtimes \blacksquare \bullet \blacktriangle \bullet \bullet \circ \square \Diamond \triangle \nabla$



 $p + scale_radius(range = c(1,6))$ p + scale_size_area(max_size = 6) alanını hazırlar.

Koordinat sistemleri

r <- d + geom bar()



r + coord cartesian(xlim = c(0, 5))xlim. vlim

Standart kartezyen koordinat sistemi

r + coord fixed(ratio = 1/2)

ratio, xlim, ylim Kartezyen koordinatları ile x ve y eksenleri arasındaki sabit görüntü oranı

r + coord flip()

xlim, ylim

Döndürülmüş kartezyen koordinatları



r + coord polar(theta = "x", direction=1) theta, start, direction Kutup koordinatları



r + coord_trans(ytrans = "sqrt") xtrans, ytrans, limx, limy Dönüştürülmüş kartezyen koordinatları. xtrans ve ytrans bir pencere fonksiyonu

 π + coord quickmap()

 π + coord_map(projection = "ortho", orientation=c(41, -74, 0))

projection, orientation, xlim, vlim mapproj paketindeki harita projeksiyonları, (mercator (default), azequalarea, lagrange, etc.)

ile ayarlanabilir.

Bölme

Bölme bir ya da daha fazla ayrık değişkenin üzerine kurulu bir cizimin değerlerini altcizimlere ayırır.

t <- ggplot(mpg, aes(cty, hwy)) + geom point()



t + facet_grid(. ~ fl) sütunları fl temelinde böler

t + facet grid(year ~ .) satırları year temelinde böler

t + facet grid(year ~ fl) hem satırları hem de sütunları böler

> t + facet wrap(~ fl) bölünen parçaları dikdörtgen biçiminde verir

Set **scales** to let axis limits vary across facets

t + facet grid(drv ~ fl, scales = "free")

x and y axis limits adjust to individual facets

• "free_x" - x axis limits adjust

• "free_y" - y axis limits adjust

Bölme etiketlerini hizalamak icin **labeller** kullanın

t + facet_grid(. ~ fl, labeller = label_both) fl: c fl: d fl: e fl: p

t + facet_grid(fl ~ ., labeller = label_bquote(alpha ^ .(fl))) α^{c} α^{d} α^{e} α^{p} α^{r}

t + facet_grid(. ~ fl, labeller = label_parsed) c d

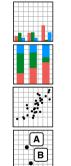
mek için ölcek

onks. kullanıı

Konum Ayarlamaları

Konum ayarlamaları geomların alanı nasıl kaplayacağını ve bu alanın nasıl düzenleneceğini belirler.

s <- ggplot(mpg, aes(fl, fill = drv))



s + geom_bar(position = "dodge") Öğeleri yan yana sıralar

s + geom_bar(position = "fill") Öğeleri birbiri üzerine yığar, yüksekliği tek standart haline getirir

e + geom_point(position = "jitter") Aşırı grafik çizimini önlemek için her öğenin

X ve Y eksenlerine rassal gürültü ekler e + geom_label(position = "nudge")

Etiketleri çizilen noktalardan uzağa yerleştirir s + geom_bar(position = "stack") Öğeleri birbiri üzerine yığar

Herbir konum manuel width(genişlik) ve height(yükseklik) argümanlarıyla, bir fonksiyon olarak, yeniden bicimlendirilebilir.

s + geom_bar(position = position_dodge(width = 1))

Etiketler

t + labs(x = "x ekseni adı", y = "y ekseni adı",

title = "Çizime başlık ekler", subtitle = "Çizime altbaşlık ekler", caption = "Çizime altyazı ekler",

<AES> = "Yeni ←<AES> lejant başlığı")

t + annotate(geom = "text", x = 8, y = 9, label = "A")

Leiant

geom'un yeri geom estetiği için manuel değerler

n + theme(legend.position = "bottom")

Lejantı yerleştirin: "bottom", "top", "left", or "right" n + guides(fill = "none")

Her estetik için lejantı değiştirme: colorbar, legend, or none (no lejant)

n + scale_fill_discrete(name = "Title", labels = c("A", "B", "C", "D", "E"))

Lejant başlığını and ölçek fonksiyonuyla etiketleri ayarlayın.

Temalar



Beyaz arka plan ızgara çizgileriyle



r + theme_classic() r + theme_light() r + theme linedraw()

r + theme_minimal() Minimal tema

theme_void() Boş tema

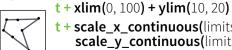
Yakınlastırma

Kırpmadan (tercih edilen)



t + coord cartesian($x \lim = c(0, 100), y \lim = c(10, 20)$

Kırparak (görünmeyen veri noktalarını keser)



t + scale x continuous(limits = c(0, 100)) +scale_y_continuous(limits = c(0, 100))