# ggplot2 ile Veri Görselleştirme : : REFERANS KAĞIDI



### Temeller

ggplot2 grafiklerin dil bilgisi (grammar of graphics) prensiplerini temel alarak oluşturulmuştur. Bu prensiplere göre her grafik aynı parçalardan oluşturulabilir: bir veri seti, koordinat sistemi, ve "geom"lar - veri noktalarını temsil eden görsel işaretler.



Değerleri göstermek için, verideki değişkenleri ilgili geom'un parametrelerine (aes, estetikleri) atayın. Bu parametrelerin bazıları büyüklük (size), rénk (color), x ve y koordinatlarıdır.



Asağıdaki taslağı kullanarak bir grafik oluşturabilirsiniz.



**ggplot(**data = mpg, **aes(**x = cty, y = hwy**))** Katman ekleyerek bitirilmesi gereken bir grafiğin tabanını oluşturur. Her katman için bir geom fonksiyonu ekleyin.

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

Bir veri seti, geom ve parametre seti için tam teşekküllü bir grafik oluşturur. Çok sayıda faydalı varsayılan değer içerir.

last\_plot() Son grafiği getirir

ggsave("plot.png", width = 5, height = 5) Son grafiği 5' x 5' boyutunda "plot.png" isimli bir dosya olarak çalışma klasörüne kaydeder. Dosya türünü, uzantısına (ör. png) göre

### Geom

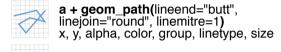
Veri noktalarının gösterimleri için bir geom fonksiyonu kullanın. Geom'un aes parametrelerini de değişken gösterimleri için kullanabilirsiniz. Her fonksiyon, grafiğe bir katman ekler.

#### **BASIT GRAFIKLER**

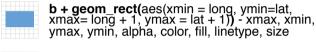
a <- ggplot(economics, aes(date, unemploy))  $b \leftarrow ggplot(seals, aes(x = long, y = lat))$ 

**a + geom\_blank()** (Grafiğin sınırlarını genişletmek için faydalı.)

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



**a + geom\_polygon(**aes(group = group)**)** x, y, alpha, color, fill, group, linetype, size





**a + geom\_ribbon(**aes(ymin=unemploy - 900, ymax=unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

### CIZGI PARCALARI

ortak parametreler: 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 = 1:1155, radius = 1))

### TEK DEĞİŞKENLİ sürekli

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,



**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

#### kesikli d <- ggplot(mpg, aes(fl))</pre>



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

### **IKI DEĞİŞKENLİ**

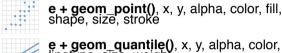
sürekli x . sürekli v e <- ggplot(mpg, aes(cty, hwy))



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\_quantile()**, x, y, alpha, color, group, linetype, size, weight



e + geom\_rug(sides = "bl"), x, y, alpha, color, **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

#### kesikli x , sürekli v 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

### kesikli x , kesikli v

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



g + geom\_count(), x, y, alpha, color, fill, shape, size, strokě

### sürekli iki değişkenli dağılım

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 + aeom hex()x, y, alpha, colour, fill, size

### sürekli fonksivon

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



i + geom\_area()
x, y, alpha, color, fill, linetype, size



i + geom\_step(direction = "hv") x, y, alpha, color, group, linetype, size

### hata değerlerini görsellestirmek

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))</pre>



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,



j + geom\_pointrange() x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

### haritalar

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

### ÜC DEĞİSKENLİ

seals\$z <- with(seals, sqrt(delta\_long^2 + delta\_lat^2)) I <- ggplot(seals, aes(long, lat))



I + geom contour(aes(z = z))x, y, z, alpha, colour, group, linetype,



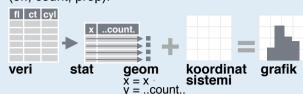
I + geom\_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE) x, y, alpha, fill



I + geom\_tile(aes(fill = z)), x, y, alpha, color, fill, linetype, size, width



Katman oluşturmanın alternatif yolu. Bir stat grafik cizdirmek için yeni değiskenler kullanır. (ör., count, prop).



Bir istatistiği (ör. histogram), bir geom fonksiyonunun stat parametrešini değiştirerek geom\_bar(stat="count") veya bir stat fonksiyonu kullanarak stat count(geom="bar") görselleştirebilirsiniz. Stat fonksiyonu geom fonksiyonuna denk bir katman oluşturur. Stat değişkenlerini aes parametrelerine atamak icin ..isim.. stilini kullanın.



c + stat\_bin(binwidth = 1, origin = 10) x, y l ...count..., ..ncount..., ..density..., ..ndensity... c + stat\_count(width = 1) x, y, | ...count..., ...prop... **c + stat\_density(**adjust = 1, kernel = "gaussian"**) x, y,** l ...count.., ..density.., ..scaled..

e + stat\_bin\_2d(bins = 30, drop = T) x, y, fill T ...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")

I + stat\_contour(aes(z = z)) x, y, z, order | ..level..

**I + stat summary hex(**aes(z = z); bins = 30; fun = max) **x**, **y**, **z**, **fill** I ..value..

 $I + stat\_summary\_2d(aes(z = z), bins = 30, fun = mean)$ x, v, z, fill | ..value..

f + stat\_boxplot(coef = 1.5) x, y l ..lower.., ..middle.., ..upper.., ..width.., ..ymin.., ..ymax.. f + stat\_ydensity(kernel = "gaussian", scale = "area") x, ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..widt

**e + stat\_ecdf(n = 40) x, y | ..x.., ..y..** 

e + stat\_quantile(quantiles = c(0.1, 0.9), formula = y ~ log(x), method = "rq") x, y I ...quantile...

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

**ggplot() + stat\_function(**aes(x = -3:3), n = 99, fun = dnorm, args = list(sd=0.5)) **x** linux...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()

# Olcek (Scale)

Ölçekler (**scale**) veri değerlerinin görsel değerlerle ilişkilerini düzenler. Bir atamayı değiştirmek için yeni bir scale fonksiyonu kullanın.



#### GENEL AMAÇLI ÖLÇEKLER

Çoğu aes parametresine uygundur.

scale\_\*\_continuous() - sürekli değerleri görsele atama

scale\_\*\_discrete() - kesikli değerleri görsele atama

scale\_\*\_identity() - veri değerlerini görsel değer yapma

scale\_\*\_manual(values = c()) - kesikli değerleri elle seçili görsel değerlere atama

scale\_\*\_date(date\_labels = "%m/%d"), date\_breaks = "2 weeks") - veri değerlerini tarih olarak kullanma.

scale\_\*\_datetime() - veri x değerlerini tarih-zaman olarak atama. scale\_x\_date() ile aynı parametreleri kullanır. Etiket formatları için konsola ?strptime yazın.

### X & Y KONUM ÖLÇEKLERİ

x ve y aes parametreleriyle kullanın (x örn. aşağıda)

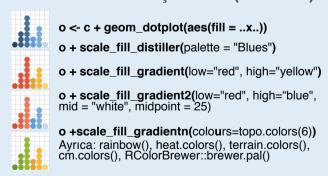
scale\_x\_log10() - x değerleri log10 ölçeğinde scale\_x\_reverse() - x koordinatını ters yüz etme scale\_x\_sqrt() - x değerleri kare kök ölçeğinde

### COLOR VE FILL ÖLCEKLERİ (KESİKLİ)



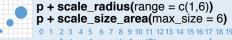


### COLOR VE FILL ÖLÇEKLERİ (SÜREKLİ)



### SHAPE VE SIZE ÖLCEKLERİ

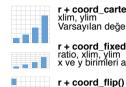
p <- e + geom point(aes(shape = fl, size = cvl)) p + scale\_shape() + scale\_size() p + scale\_shape\_manual(values = c(3:7))



 $\Box \circ \triangle + \times \Diamond \neg \boxtimes * \oplus \circ \boxtimes \boxtimes \boxtimes \Box \circ \triangle \Diamond \circ \circ \circ \Box \Diamond \triangle \neg$ 

### Koordinat Sistemleri

r <- d + geom\_bar()



+ coord\_cartesian(xlim = c(0, 5)) xlim, ylim Varsavılan değerdeki koordinat sistemi

r + coord\_fixed(ratio = 1/2)

ratio, xiim, yiim x ve y birimleri arasındaki oranı sabitleyen koordinat sistemi

xlım, ylım Yan dönmüş kartezyen koordinatlar r + coord\_polar(theta = "x", direction=1 ) theta, start, direction Kutupsal koordinatlar (daire grafiği)

r + coord trans(vtrans = "sart") trans, tyrans, limy, limy Dönüştürülmüş koordinatlar. xtrans ve ytrans parametrelerini özel fonksiyonlara atayabilirsiniz.



π + coord quickmap()

 $\pi$  + coord\_map(projection = "ortho", orientation=c(41, -74, 0))projection, orienztation, xlim, ylim mapproj paketindeki harita gösterimleri(mercator (varsayılan) azegualarea, lagrange, vs.)

### Konum Düzenlemeleri

qeomların üst üste çakısmasını önlemek için yapılan konum (position) düzenlemeleri.



s + geom\_bar(position = "dodge") Değerleri yan yana koyma.

s + geom\_bar(position = "fill") Değerleri tepe tepeye koyup normalize etme.



e + geom\_point(position = "jitter") X ve Y değerleriyle oynayarak noktaların üst üste çakışmasını engelleme

• A

e + geom\_label(position = "nudge") Etiketleri noktalarla üst üste gelmesini engelleme.



s + geom\_bar(position = "stack") Değerleri diğerlerinin tepesine koyma

Her pozisyon parametresi bir fonksiyon olarak çağırılabilir ve elle uzunluk ve genişlik parametreleri ayarlanabilir.

s + geom\_bar(position = position\_dodge(width = 1))

### Temalar (Theme)



r + theme\_bw() Beyaz arkaplan, ızgara çizgileri

karanlık tema



r + theme\_classic() r + theme light()

r + theme linedraw() r + theme minimal() Minimal tema

r + theme void() Boş tema

## Cephe(Facet)

Cepheler (facet) bir grafiği bir veya daha fazla kesikli değişkenin değerlerine göre alt grafiklere avırır.



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

t + facet\_grid(. ~ fl) fl değişkenine göre sütunlara ayırma t + facet\_grid(year ~ .) year değişkenine göre satırlara ayırma

t + facet\_grid(year ~ fl)
year ve fl'ye göre hem satır hem sütuna ayırma

t + facet\_wrap(~ fl)

cepheleri dikdörtgen düzeninde yerleştirme

Cephelerin genel koordinat limitleri için scales kullanın.

t + facet\_grid(drv ~ fl, scales = "free") x ve y koordinat limitlerini ayrı ayrı ayarlayabilirsiniz "free\_x" - x koordinatı limit ayarı "free\_y" - y koordinatı limit ayarı

labeller parametresi ile cephe isimlerini verebilirsiniz.

t + facet grid(, ~ fl. labeller = label both) fl: c fl: d fl: e fl: p fl: r t + facet\_grid(fl ~ ., labeller = label\_bquote(alpha ^ .(fl)))  $lpha^c$   $lpha^d$   $lpha^e$   $lpha^p$   $lpha^r$ 

t + facet\_grid(. ~ fl, labeller = label\_parsed) c de pr

# Etiketler (label)

t + labs( x = "Yeni x doğrusu Etiketi", y = "Yeni y doğrusu etiketi". title ="Grafiğe başlık ekleme"

subtitle = "Ğrafiğe alt başlık ekleme" caption = "Grafiğe alt yazı ekle")

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

### .ejant (Legend n + theme(legend.position = "bottom")

Place legend at "bottom", "top", "left", or "right" n + guides(fill = "none")

Set legend type for each aesthetic: colorbar, legend, or none (no legend)

n + scale\_fill\_discrete(name = "Title", labels = c("A", "B", "C", "D", "E")) Set legend title and labels with a scale function.

# Zooming (Yakınlaştırma)



Without clipping (preferred)

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

With clipping (removes unseen data points)



t + scale\_x\_continuous(limits = c(0, 100)) + scale\_y\_continuous(limits = c(0, 100))

