

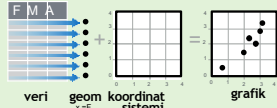
# ggplot2 ile Veri Görselleştirme

## Referans Kağıdı

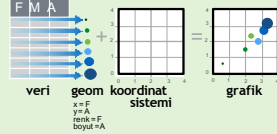


### Temel Öğeler

ggplot2 grafiklerin dilbilgisine dayanmaktadır, buradaki ana fikir her grafik, aynı şekilde oluşabilecek birkaç bileşenden oluşur. Bunlar: bir veri kümesi, bir grup geom — veri noktalarını geometrik olarak temsil eden işaretler ve bir koordinat sistemidir.



Değerleri göstermek için verideki değişkenler, boyut, renk ve x ve y koordinatları gibi geom' un parametrelerine göre atanır.



qplot() veya ggplot() ile grafik oluşturun.

parametreler veri geom

qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")  
Veri verildiğinde, geom ve parametreler ile tam bir grafik oluşturulabilir. Bazı yararlı varsayılan değerleri destekler.

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

Tamamlayıcılar ekleyerek çizdirmek istediğiniz grafiğe temel oluşturur. Varsayılan olmadan, qplot() 'tan daha fazla kontrol yolu sağlar.

veri

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

+ ile daha fazla  
eleman eklenir

Tamamlayıcı = geom +  
varsayılan

istatistik + tamamlayıcıya  
özgü atamalar

Ek tamamlayıcılar

geom\_\*() veya stat\_\*() fonksiyonları ile grafiğe yeni bir tamamlayıcı eklenir. Bu ikisi bir dizi estetik gösterimi, istatistik yöntem ve konum düzenlemesini geom ile sağlarlar.

last\_plot()

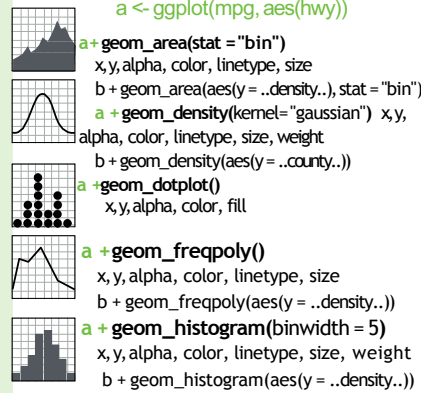
Son grafiği gösterir.

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

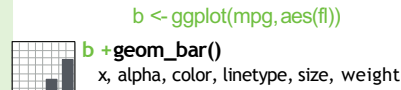
Son grafiği 5'x 5' olacak boyutta "plot.png" ismiyle çalışma dizinine çalışılan dosya olarak kaydeder. Dosya türünü dosya uzantısıyla eşleştirir.

**Geoms** -Veri noktalarını temsil etmek için bir geom, değişkenleri temsil etmek için geom' un parametrelerini kullanın. Her fonksiyon bir tamamlayıcı ekler.

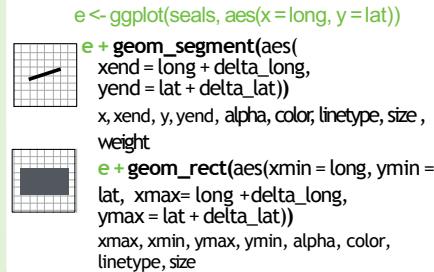
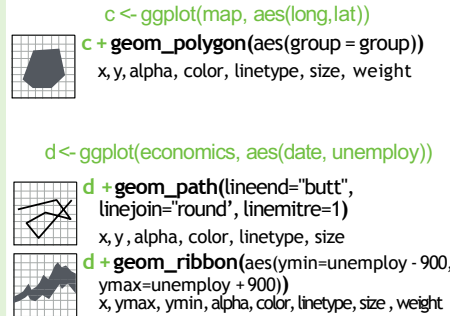
### Tek Değişkenli Sürekli



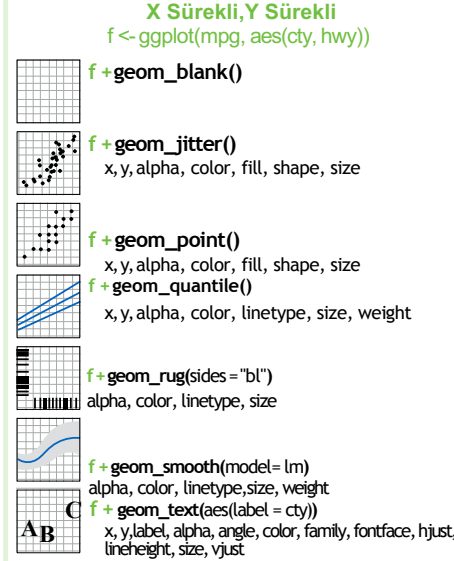
### Kesikli



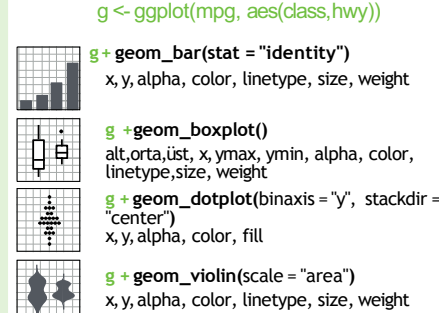
### Grafik İlkeleri



### İki Değişkenli



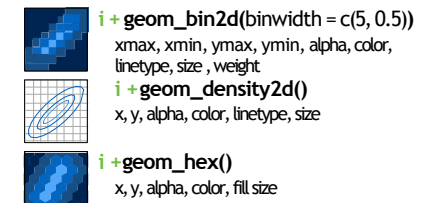
### X Kesikli, Y Sürekli



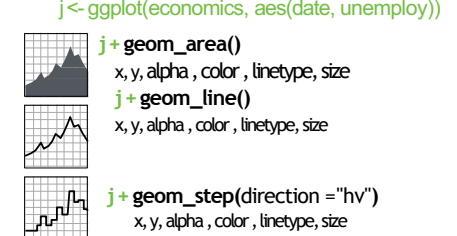
### X Kesikli, Y Kesikli



### İki Değişkenli Sürekli Dağılım



### Sürekli Fonksiyon



### Hataları Görselleştirme

```
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))
```



### Haritalar

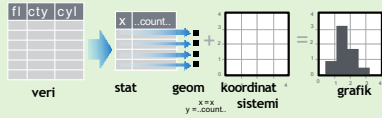
```
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))
l + geom_map(aes(map_id = state), map = map) +
  expand_limits(x = map$long, y = map$lat)
map_id, alpha, color, linetype, size, weight
```

### Üç Değişkenli



## İstatistik (Stat) - Tamamlayıcı oluşturma için alternatif bir yolu

Bazı grafikler asıl verinin bir çeşit görselliğe dönüştürür. Bunun için stat kullanılır. Örneğin, `a + geom_bar(stat="bin")`



Stat ve geom fonksiyonları beraber kullanılarak, örneğin; geom fonksiyonunun stat parametresini veya istatistik fonksiyonunun geom parametresini değiştirerek tamamlayıcı oluşturmak mümkündür. Örneğin, `stat_bin(geom="bar")` fonksiyonu ile `geom_bar(stat="bin")` ile aynı grafiği çıkarır. Her istatistiği (stat),



`a + stat_bin(binwidth = 1, origin = 0)`  
x, y | ..count.., ..ncount.., ..density..  
`a + stat_bin2d(binwidth = 1, binaxis = "x")`  
x, y | ..count.., ..ncount..  
`a + stat_density2d(adjust = 1, kernel = "gaussian")`  
x, y | ..count.., ..density.., ..scaled..

Tek Boyutlu dağılımlar

`f + stat_bin2d(bins = 30, drop = TRUE)`  
x, y, fill | ..count.., ..density..  
`f + stat_binhex(bins = 30)`  
x, y, fill | ..count.., ..density..  
`f + stat_density2d(contour = TRUE, n = 100)`  
x, y, color, size | ..level..

İki Boyutlu dağılımlar

`m + stat_contour(aes(z = z))`  
x, y, z, order | ..level..  
`m + stat_spoke(aes(radius = z, angle = z))`  
Angle, radius, x, yend, y, yend | ..x.., ..xend.., ..y.., ..yend..  
`m + stat_summary_hex(aes(z = z), bins = 30, fun = mean)`  
x, y, z, fill | ..value..  
`m + stat_summary2d(aes(z = z), bins = 30, fun = mean)`  
x, y, z, dolgu | ..value..

3 Değişken

`g + stat_boxplot(coef = 1.5)`  
x, y | ..lower.., ..middle.., ..upper.., ..outliers..  
`g + stat_ydensity(adjust = 1, kernel = "gaussian", scale = "area")`  
x, y | ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..

Karşılaştırmalar

`f + stat_ecdf(n = 40)`  
x, y | ..x.., ..y..  
`f + stat_quantile(quantiles = c(0.25, 0.5, 0.75), formula = y ~ log(x), method = "rq")`  
x, y | ..quantiles.., ..x.., ..y..  
`f + stat_smooth(method = "auto", formula = y ~ x, se = TRUE, n = 80, fullrange = FALSE, level = 0.95)`  
x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax..

Fonksiyonlar

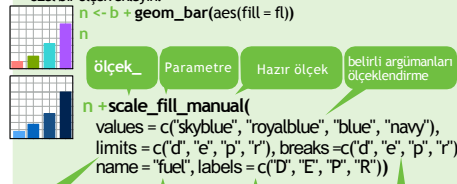
`ggplot() + stat_function(aes(x = -3:3), fun = dnorm, n = 101, args = list(sd=0.5))`  
x | ..y..

Genel olarak

`f + stat_identity()`  
`ggplot() + stat_qq(aes(sample=1:100), distribution = qt, dparams = list(df=5))`  
sample, x, y | ..x.., ..y..  
`f + stat_sum()`  
x, y, fill | ..fill..  
`f + stat_summary(fun.data = "mean_cl_boot")`  
`f + stat_unique()`

## Ölçekler

Ölçekler (scale), verideki değerlerin bir parametreyi oluşturan görsel değerlerine nasıl atanacağını kontrol eder. Atamayı değiştirmek için özel bir ölçek ekleyin.



`n <- b + geom_bar(aes(fill = fl))`  
`n + scale_fill_manual(values = c("skyblue", "royalblue", "blue", "navy"), limits = c("d", "e", "p", "r"), breaks = c("d", "e", "p", "r"), name = "fuel", labels = c("D", "E", "P", "R"))`

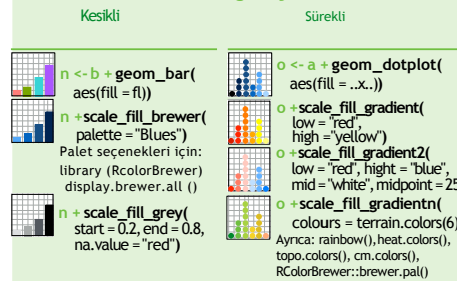
Değer aralıkları Lejanttaki etiketler Lejanttaki boşluklar  
Genel amaçlı ölçekler Herhangi bir parametre kullanın: Alpha, color, linetype, size

`scale_*_continuous()` - sürekli değerleri görsel değerlerle eşleyin  
`scale_*_discrete()` - ayrık değerleri görsel değerlerle eşleme  
`scale_*_identity()` - veri değerlerini görsel değerler olarak kullan  
`scale_*_manual(values = c())` - ayrık değerleri eşlemek için manuel olarak seçilen görsel değerler

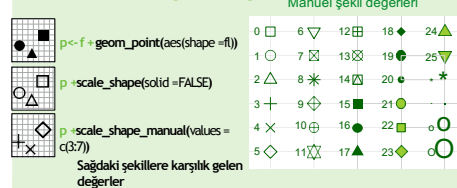
X ve Y konum ölçekleri

X veya Y parametresiyle kullanın (burada x örneklendirildi)  
`scale_x_date(labels = date_format("%m/%d"), breaks = date_breaks("2 weeks"))` - x değerlerini tarih olarak ele alın. Yazım formatlarına %strptime ile bakın.  
`scale_x_datetime()` - X değerlerini tarih olarak ele alın. Aynı argümanları `scale_x_date()` olarak kullanın.  
`scale_x_log10()` - X' i log10 ölçeğinde çizim.  
`scale_x_reverse()` - X eksenini ters çevirin.  
`scale_x_sqrt()` - X' i karekök ölçeğinde çizim.

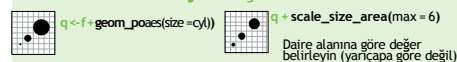
Renk ve dolgu ölçekleri



Şekil ölçekleri



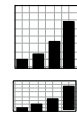
Boyut ölçekleri



## Koordinat Sistemleri

`r <- b + geom_bar()`

`r + coord_cartesian(xlim = c(0, 5))`



xlim, ylim

Varsayılan kartezyen koordinat sistemi

`r + coord_fixed(ratio = 1/2)`



oran, xlim, ylim

x ve y birimleri arasında sabit en boy oranı ile kartezyen koordinatlar

`r + coord_flip()`



xlim, ylim

Yan çevrilmiş kartezyen koordinatlar

`r + coord_polar(theta = "x", direction=1)`



Theta, start, direction

Kutupsal koordinatlar

`r + coord_trans(ytrans = "sqrt")`



xtrans, ytrans, xlim, ylim

Dönüştürülmüş kartezyen koordinatlar. Ektralan ve sınırları bir pencere fonksiyonuna atayın.

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

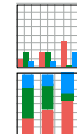


Projection, orientation, xlim, ylim, functions  
Mapproj paketinden harita fonksiyonları (mercator (varsayılan), azequalearea, lagrange, vb.)

## Konum Ayarlamaları

Konum ayarlamaları geom nesnelerinin nasıl düzenleneceğini belirler aksi takdirde aynı alanı kaplar.

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



`s + geom_bar(position = "dodge")`

Öğeleri yan yana düzenleme

`s + geom_bar(position = "fill")`

Öğeleri üst üste kümeleyin, yüksekliği normalleştirin

`s + geom_bar(position = "stack")`

Öğeleri üst üste kümeleyin

`f + geom_point(position = "jitter")`

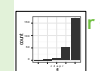


Aşırı çizimi önlemek için her bir x ve y konumu ögesine rastgele gürtütlü ekleyin.

Her konum ayarı, manuel olarak genişlik ve yükseklik ayarlarının bir fonksiyonu olarak yeniden yazılabilir.

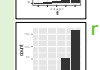
`s + geom_bar(position = position_dodge(width = 1))`

## Temalar



`r + theme_bw()`

Izgara çizgileri ile beyaz arka plan



`r + theme_grey()`

Gri arka plan (varsayılan tema)



`r + theme_classic()`

Izgara çizgileri olmadan beyaz arka plan

ggthemes - Ek ggplot2 temaları içeren paket

## Bölümleme

Grafiği bir veya daha fazla kesikli değişkenin değerlerine göre alt grafiklere böler.

`t <- ggplot(mpg, aes(cty, hwy)) + geom_point()`



`t + facet_grid(. ~ fl)`

fl değişkenine göre sütunlara ayırır.

`t + facet_grid(year ~ .)`

year değişkenine göre satırlara ayırır.

`t + facet_grid(year ~ fl)`

year ve fl değişkenlerine göre hem satırlara hem sütunlara ayırır.

`t + facet_wrap(~ fl)`

bölümleri dikkörtgen olarak yerleştirme

Genel koordinat sınırları için scales kullanın.

`t + facet_grid(y ~ x, scales = "free")`

x ve y eksen sınırlarını ayrı ayrı ayarlayın.

• "free\_x" - x eksen sınırnı ayarlayın.

• "free\_y" - y eksen sınırnı ayarlayın.

Labeller parametresi ile bölüntülere isim verebilirsiniz.

`t + facet_grid(. ~ fl, labeller = label_both)`

fl: c	fl: d	fl: e	fl: p	fl: r
c	d	e	p	r

`t + facet_grid(. ~ fl, labeller = label_bquote(alpha ^ .(x)))`

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

Yukarıdakilerin hepsi

## Etiketler

`t + ggtitle("New Plot Title")`

Grafiğin üzerine bir ana başlık ekleyin.

`t + xlab("New X label")`

X eksenindeki etiketi değiştirin.

`t + ylab("New Y label")`

Y eksenindeki etiketi değiştirin.

`t + labs(title = "New title", x = "New x", y = "New y")`

Yukarıdakilerin hepsi

Lejant etiketlerini değiştirmek için ölçek fonksiyonu kullanın.

## Lejantlar

`t + theme(legend.position = "bottom")`

Lejant, 'alta', 'üste', 'sola' veya 'sağa' yerleştirin

`t + guides(color = "none")`

renk çubuğu, gösterge, veya hiçbir (gösterge yok) için açıklama yapın.

`t + scale_fill_discrete(name = "Title", labels = c("A", "B", "C"))`

Lejant başlığı ve etiketleri ölçek fonksiyonuyla ayarlayın.

## Yakınlaştırma

Kırpma Olmadan (tercih edilir)

`t + coord_cartesian(xlim = c(0, 100), ylim = c(10, 20))`



Kırparak

(Görünmeyen veri noktalarını kaldırın.)

`t + xlim(0, 100) + ylim(10, 20)`

`t + scale_x_continuous(limits = c(0, 100)) + scale_y_continuous(limits = c(0, 100))`



Learn more at docs.ggplot2.org • ggplot2 0.9.3.1 • Updated: 3/15