3_hafta_görselleştirme

Hakan Mehmetcik

2024-10-14

3. Hafta: Veri Görselleştirme ve ggplot Kullanımı

Veri görselleştirme, istatiksel veri analizinin en önemli yönlerinden biridir.

- İyi bir görselleştirme, analizlerinizi daha temiz, anlaşılır ve kavrayıcı hale getirir.
- İyi bir görselleştirme, veriyi kendinizin de anladığınıza dair bir doğrulama sağlar!

Eski New York Times stajyeri ve FlowingData.com'un yaratıcısı Nathan Yau, veri grafiklerinin oluşturulmasını yemek pişirmeye benzetiyor: Herkes grafik komutlarını yazmayı öğrenip bilgisayarda grafikler oluşturabilir. Benzer şekilde, herkes bir mikrodalga fırında yemek ısıtmayı öğrenebilir. Yüksek kaliteli bir görselleştirme ile sıradan bir görselleştirme arasındaki fark, büyük şeflerle acemileri ayıran unsurların aynısıdır: araçlarının ustalığı, malzemeleri hakkında bilgisi, içgörü ve yaratıcılık (Yau 2013).

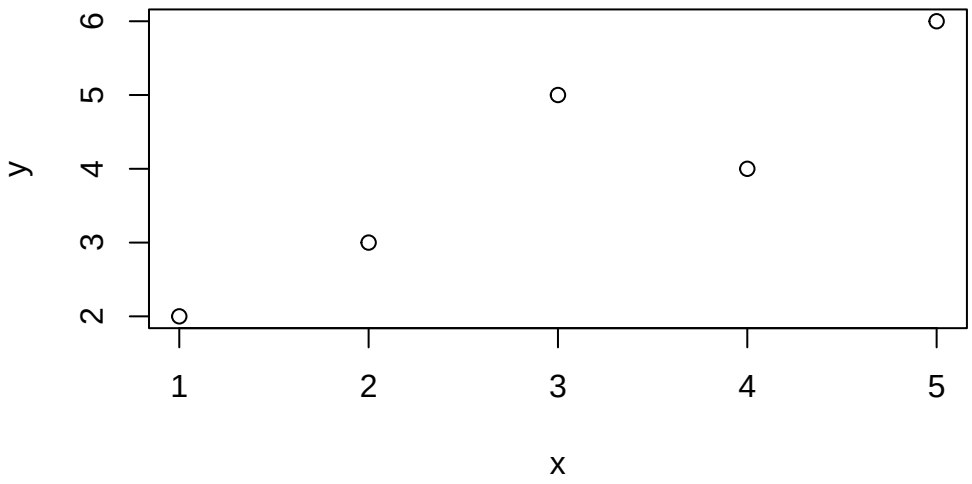
Bu bölümde, veri grafiklerinin temelini anlamak için bir çerçeve sunuyoruz. Bu amaçla, öncelikle, R ile veri görselleştirmeye dair temel kavramları ele alacağız. Ardından, **ggplot2** paketini kullanarak daha gelişmiş görselleştirme tekniklerine geçeceğiz.

1. R ile Temel Veri Görselleştirme

R, verilerinizi görselleştirmenin birçok yolunu sunar. En temel görselleştirme yöntemleri arasında, `plot()` fonksiyonu ile grafik oluşturma yer alır. Aşağıda, `plot()` fonksiyonu ile bir scatter plot (dağılım grafiği) oluşturma örneği verilmiştir:

```
# Örnek veri seti
x <- c(1, 2, 3, 4, 5)
y <- c(2, 3, 5, 4, 6)

# Scatter plot oluşturma
plot(x, y)
```



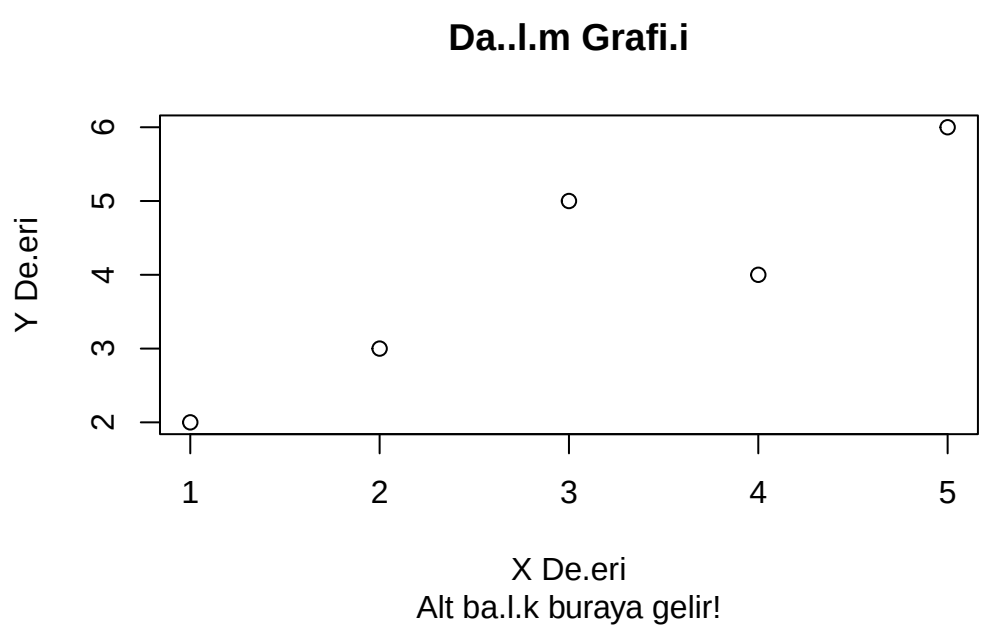
Bu kod, `x` ve `y` vektörlerini kullanarak bir dağılım grafiği oluşturur. `plot()` fonksiyonu, R'nın varsayılan olarak sağladığı genel (temel) bir fonksiyondur; tıpkı `print()` ve `summary()` gibi. Varsayılan çizim yönteminin yardım belgelerine (yani `?plot` ya da `help("plot")` yazarak) baktığınızda, grafiğinizi özelleştirmek için belirtebileceğiniz çok uzun bir argüman listesi göreceksiniz.

Başlık ve Eksenleri Özelleştirme

Daha uygun eksen etiketleri belirtmek, bir başlık eklemek veya alt başlık eklemek doğru bir başlangıç olacaktır. Bunu sağlamak için belirtmeniz gereken argümanlar:

- **main**: Başlığı içeren bir karakter dizisi.
- **sub**: Alt başlığı içeren bir karakter dizisi.
- **xlab**: X ekseni etiketini içeren bir karakter dizisi.
- **ylab**: Y ekseni etiketini içeren bir karakter dizisi.

```
plot(x,y,
     main="Dağılım Grafiği",
     sub = "Alt başlık buraya gelir!",
     xlab="X Değeri",
     ylab="Y Değeri")
```

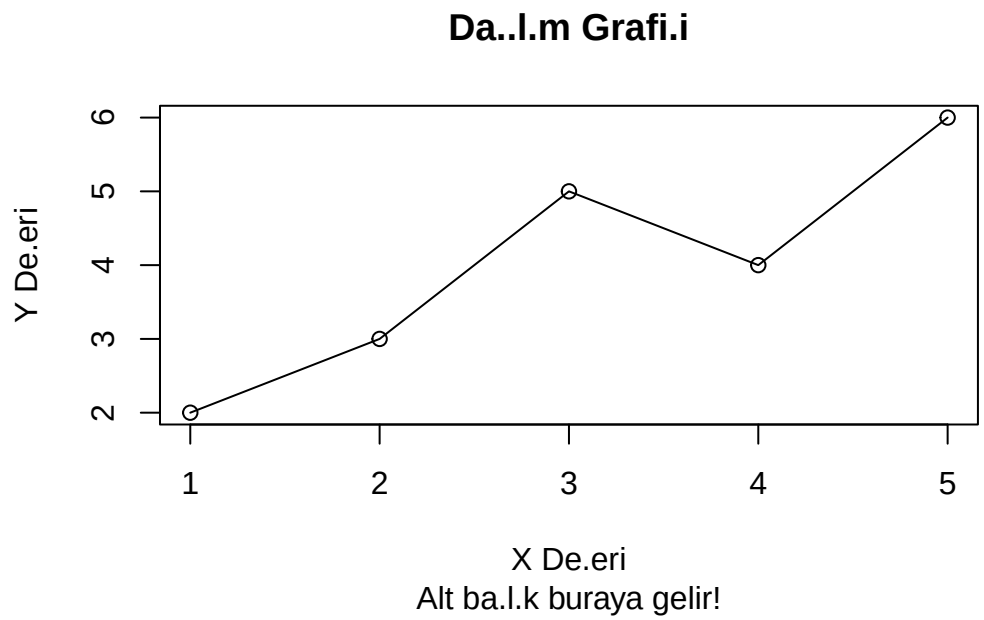


Grafik Türünü Değiştirme

`type` argümanı, grafiğin görsel stilini belirtir. Bunun için olası değerler:

- `type = "p"`: Sadece noktaları çizin.
- `type = "l"`: Noktalar üzerinden bir çizgi çizin.
- `type = "o"`: Noktaların üstüne çizgi çizin.
- `type = "b"`: Hem noktaları hem de çizgileri çizin, ancak üst üste gelmesin.
- `type = "h"`: "Histogram benzeri" dikey çubuklar çizin.
- `type = "s"`: Yatay sonra dikey giden bir merdiven çizin.
- `type = "S"`: Dikey sonra yatay giden bir merdiven çizin.
- `type = "c"`: "b" versiyonundan yalnızca bağlantı çizgilerini çizin.
- `type = "n"`: Hiçbir şey çizmeyin. (Bunun bazı durumlarda yararlı olduğu söyleniyor.)

```
plot(x,y,
     main="Dağılım Grafiği",
     sub = "Alt başlık buraya gelir!",
     xlab="X Değeri",
     ylab="Y Değeri",
     type = "o")
```

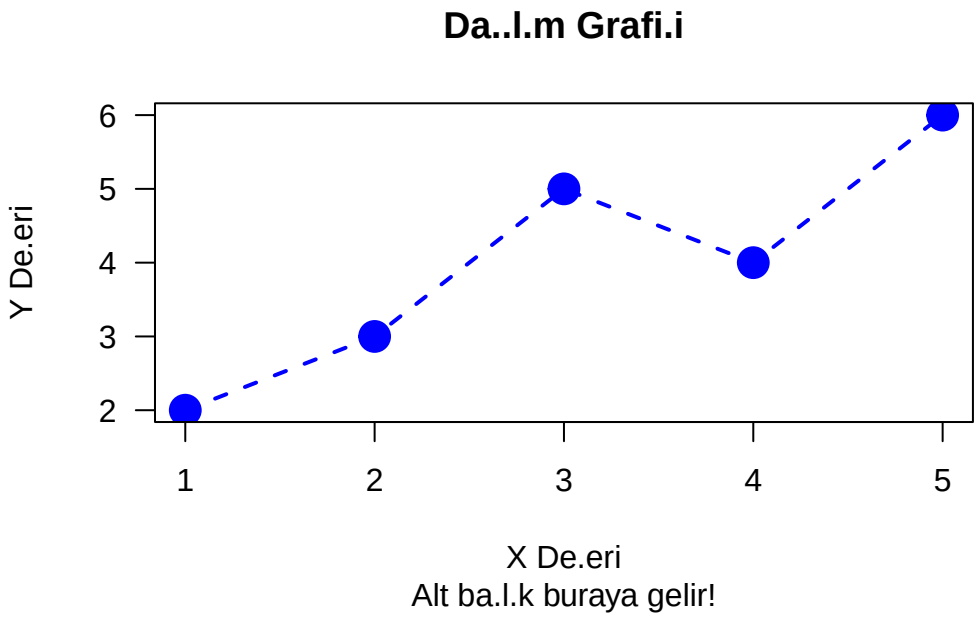


Diğer Değerleri Değiştirme

- Grafik karakterleri, `pch` ve `lty` değerleri ile değiştirilebilir.
- Grafiğin rengi `col` ile değiştirilebilir.
- Grafik boyutu `cex` ile değiştirilebilir.

```
  plot(x,y,
     main="Dağılım Grafiği",
     sub = "Alt başlık buraya gelir!",
     xlab="X Değeri",
     ylab="Y Değeri",
     type = "o",
     col="blue",
```

```
     pch=19,
     cex = 2,
     lty = 2,
     lwd = 2,
     las = 1)
```



2. ggplot2 ile Gelişmiş Veri Görselleştirme

ggplot2 paketi, R ile veri görselleştirmenin en popüler ve güçlü yollarından biridir. ggplot2, "Grammar of Graphics" (Grafiklerin Grameri) ilkesine dayanmaktadır ve çok çeşitli grafik türlerini kolayca oluşturmanıza olanak tanır.

ggplot2 Kurulumu

ggplot2 paketini kullanmaya başlamak için öncelikle paketi kurmalısınız. Aşağıdaki kod ile ggplot2'yi yükleyebilirsiniz:

```
# install.packages("ggplot2")
library(ggplot2)
```

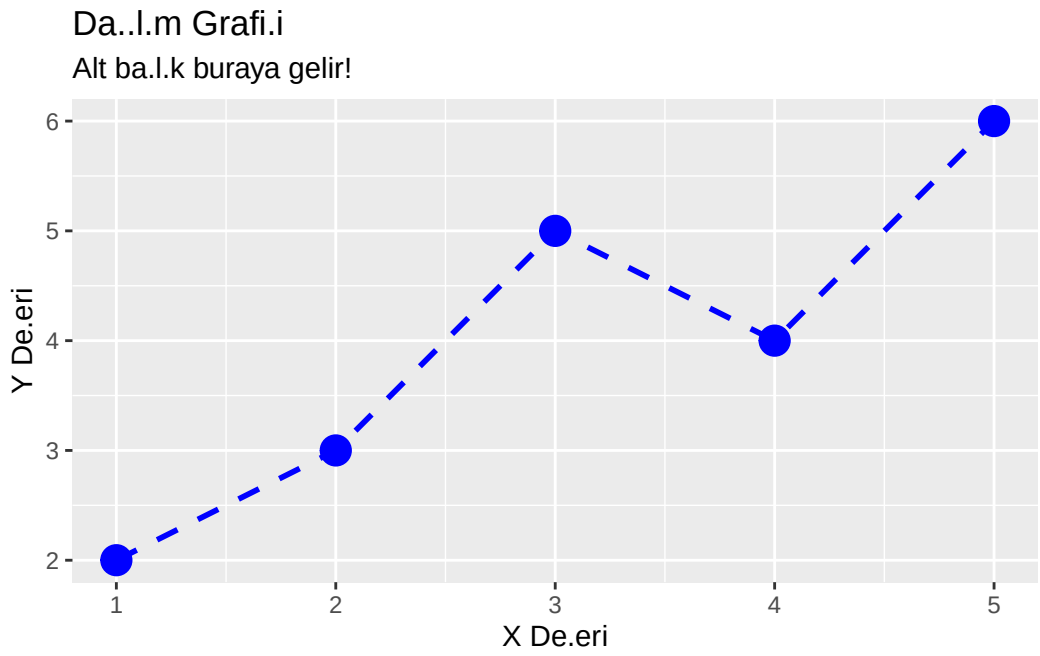
Basit bir ggplot Kullanımı

ggplot2 ile bir dağılım grafiği oluşturmak için aşağıdaki gibi bir kod kullanabilirsiniz:

```
# Örnek veriler
x <- c(1, 2, 3, 4, 5)
y <- c(2, 3, 5, 4, 6)

# ggplot ile dağılım grafiği oluşturma
ggplot(data = data.frame(x, y), aes(x = x, y = y)) +
  geom_line(color = "blue", linetype = "dashed", size = 1) +  # Çizgi
  geom_point(color = "blue", size = 5) +  # Noktalar
  labs(title = "Dağılım Grafiği",
       subtitle = "Alt başlık buraya gelir!",
       x = "X Değeri",
       y = "Y Değeri")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



Bu kod, `data` veri çerçevesindeki `x` ve `y` değerlerini kullanarak bir dağılım grafiği oluşturur. `geom_point()` fonksiyonu, noktaları eklemek için kullanılır.

3. Ggplot'un Temel Unsurları

ggplot çerçevesinde, veri grafiklerini **beş temel unsur** açısından anlamak mümkündür: Estetikler, görsel ipuçları, koordinat sistemleri, ölçek, ve bağlam.

1. Estetikler (Aesthetics)

Estetikler, bir grafik üzerindeki nesnelerin görsel özelliklerini tanımlayan parametrelerdir. Bu özellikler, verilerin nasıl temsil edileceğini belirler ve verinin izleyiciye nasıl iletileceğini etkiler. Estetikler, bir değişkenin hangi görsel unsur ile temsil edileceğini tanımlar ve bu sayede grafik üzerindeki unsurların farklı özellikler kazanmasını sağlar.

Tablo 1: Estetikler

Estetik	Veri Türü	Açıklama
`x`	Sayısal / Kategorik	X ekseninde gösterilecek değişken.
`y`	Sayısal	Y ekseninde gösterilecek değişken.
`fill`	Kategorik / Sayısal	Çubuk, alan veya noktaların iç rengini belirler.
`color`	Kategorik / Sayısal	Noktaların veya çizgilerin dış rengini belirler.
`size`	Sayısal	Noktaların veya çizgilerin boyutunu ayarlar.
`shape`	Kategorik	Noktaların şeklini belirler (örneğin, daire, üçgen).
`linetype`	Kategorik	Çizgi türünü belirler (örneğin, kesik, düz).
`alpha`	Sayısal	Noktaların veya alanların saydamlık düzeyini ayarlar.
`label`	Kategorik	Noktaların üzerine yazılacak metni belirler.
`stroke`	Sayısal	Kenar kalınlığını ayarlar, genellikle `shape` ile birlikte kullanılır.
`order`	Sayısal / Kategorik	Çizim sırasını belirler, özellikle yığılmış grafikte kullanılır.
`xend` / `yend`	Sayısal	Çizgi veya ok uç noktalarını belirler.

2. Görsel İpuçları (Visual Cues)

Görsel ipuçları, grafiklerde belirli bilgileri iletmek için kullanılan grafiksel unsurlardır. Bu unsurlar, izleyicinin dikkatini çekmek ve verilerin anlamını kolaylaştırmak için kullanılır. Görsel ipuçları şunları içerir:

Tablo 2: Görsel ipuçları ve anlamları.

Görsel İpucu	Değişken Türü	Soru
Konum	sayısal	Diğer şeylerle ilişkili olarak nerede?
Uzunluk	sayısal	Ne kadar büyük (bir boyutta)?
Açı	sayısal	Ne kadar geniş? Bir şeyle paralel mi?
Yön	sayısal	Hangi eğimde? Zaman serisinde, yükseliyor mu yoksa alçalıyormu?
Şekil	kategorik	Hangi gruba ait?
Alan	sayısal	Ne kadar büyük (iki boyutta)?
Hacim	sayısal	Ne kadar büyük (üç boyutta)?
Gölge	her ikisi	Ne ölçüde? Ne kadar şiddetli?
Renk	her ikisi	Ne ölçüde? Ne kadar şiddetli?

i Note

Görsel ipuçları, izleyicinin dikkatini çekmek istediğiniz unsurlara odaklanmasını sağlamak için kullanılan grafiksel öğelerdir.
Grafiksel algı üzerine yapılan araştırmalara göre (1980'lerin ortalarına kadar uzanıyor) insanlar konumdaki farklılıkları (örneğin, bir çubuğun diğerinden ne kadar daha uzun olduğunu) doğru bir şekilde algılamakta oldukça iyidir; ancak açılardaki farklılıkları algılamakta o kadar iyi değillerdir. Bu, birçok insanın pasta grafikleri yerine çubuk grafiklerini tercih etmesinin bir nedenidir. Renklerdeki farklılıkları algılamadaki göreceli zayıf yeteneğimiz, birçok veri bilimcisinin ısı haritalarına karşı duyduğu düşük ilginin başlıca faktörüdür.
Renk, en göz alıcı, ancak en yanlış algılanan ve kötüye kullanılan görsel ipuçlarından biridir. Renk seçiminde, herhangi bir veri bilimcisinin anlaması gereken birkaç temel fikir vardır. Öncelikle, renk insanlara görsel olarak çekici gelse de, genellikle umduğumuz kadar bilgilendirici değildir. İkinci olarak, nüfusun yaklaşık %8'i—çoğu erkek—bir tür renk körlüğüne sahiptir. Renk körlüğü sorunlarını önlemek için, veri grafiklerinde kırmızı ile yeşili karşılaştırmaktan kaçının. Ek olarak, grafikleri Noel temalı gibi görünmekten kurtarabilirsiniz!
Not: RColorBrewer paketi, bu paletleri doğrudan R'de kullanma işlevselliği sağlar.

3. Koordinat Sistemleri (Coordinate Systems)

Koordinat sistemleri, verilerin grafik üzerinde nasıl düzenleneceğini belirler. ggplot2, verilerinizi görselleştirmek için farklı koordinat sistemleri kullanmanıza olanak tanır. En yaygın koordinat sistemleri şunlardır:

- **Kartezyen Koordinat Sistemi**: X ve Y eksenlerine dayanan en yaygın sistemdir. Sayısal verilerin karşılaştırılması için kullanılır.
- **Kutuplar**: Noktaların, açılar ve uzaklıklar kullanılarak tanımlandığı bir sistemdir. Özellikle dairesel verilere yönelik grafikleri oluşturmak için kullanılır.
- **Geografik Koordinatlar**: Coğrafi verilerin (örneğin, harita üzerindeki noktalar) temsil edildiği bir sistemdir. Bu sistemde, noktalar dünya üzerindeki belirli konumları ifade eder. Özellikle haritalar üzerinde veri görselleştirme amacı ile kullanılır.

4. Ölçek (Scale)

Ölçek, verilerin görsel ipuçlarına nasıl dönüştürüleceğini tanımlar. ggplot2'de, her değişken için bir ölçek belirlemek mümkündür. Ölçekler, aşağıdakileri içerir:

- **Sayısal Ölçekler**: Sayısal değerlerin nasıl gösterileceğini belirler (örneğin, lineer veya logaritmik ölçek).
- **Kategorik Ölçekler**: Kategorik verilerin sıralamasını ve görünümünü belirler.
- **Zaman Ölçekleri**: Zaman serisi verilerinin nasıl gösterileceğini belirler. Zaman ölçekleri genellikle tarih ve saat biçiminde düzenlenir.

Ölçekler, izleyicilere verilerin anlamını iletmek için kritik öneme sahiptir.

5. Bağlam (Context)

Bağlam, bir grafikte görülen bilgilerin ne anlama geldiğini açıklamak için sağlanan ek bilgilerdir. Bağlam unsurları şunları içerir:

- **Başlıklar**: Grafiğin genel amacını ve içeriğini özetleyen metinler.
- **Eksen Etiketleri**: X ve Y eksenlerinin neyi temsil ettiğini açıklayan etiketler.
- **Açıklamalar**: Grafikteki belirli noktalar veya bölgeler hakkında ek bilgiler sağlayan notlar.
- **Referans Noktaları**: Ek bağlam sağlayan diğer grafik veya veri unsurları.

Bağlam, izleyicilerin veriyi daha iyi anlamalarına yardımcı olur ve grafiklerin yorumlanmasını kolaylaştırır.

Estetiklerin Diğer Unsurlardan Farklılıkları

- **Görsel İpuçları**: Estetikler, görsel ipuçlarının temellerini oluşturur. Görsel ipuçları, izleyicinin dikkatini çekmek için grafiksel unsurların konumunu, boyutunu, şeklini, rengini ve diğer görsel özelliklerini kullanır. Estetikler, bu unsurların neye göre belirleneceğini tanımlar.
- **Koordinat Sistemleri**: Koordinat sistemleri, verilerin grafik üzerinde nasıl düzenleneceğini belirler. Estetikler ise, bu düzenleme içinde hangi verilerin hangi görsel öğelerle temsil edileceğini belirler.
- **Ölçek**: Ölçekler, verilerin görsel ipuçlarına dönüşümünü tanımlar. Estetikler, bu dönüşüm için hangi özelliklerin kullanılacağını belirler.
- **Bağlam**: Bağlam, grafik içindeki bilgilerin anlamını sağlamak için ek bilgiler sunar. Estetikler ise, bağlamdan bağımsız olarak, verilerin nasıl temsil edileceğini tanımlar.

ggplot Grafiğinde Estetiklerin Lokasyonu

Estetiklerin lokasyonu (aesthetics location) terimi, ggplot2'de estetiklerin (görsel özelliklerin) nasıl tanımlandığını ve hangi düzeyde kullanıldığını ifade eder. Estetikler, belirli bir grafik öğesinin görünümünü etkileyen özelliklerdir ve bu özelliklerin grafik üzerinde hangi nesnelerle ilişkilendirileceği önemli bir konudur. ggplot2, estetikleri iki ana düzeyde tanımlamanıza olanak tanır:

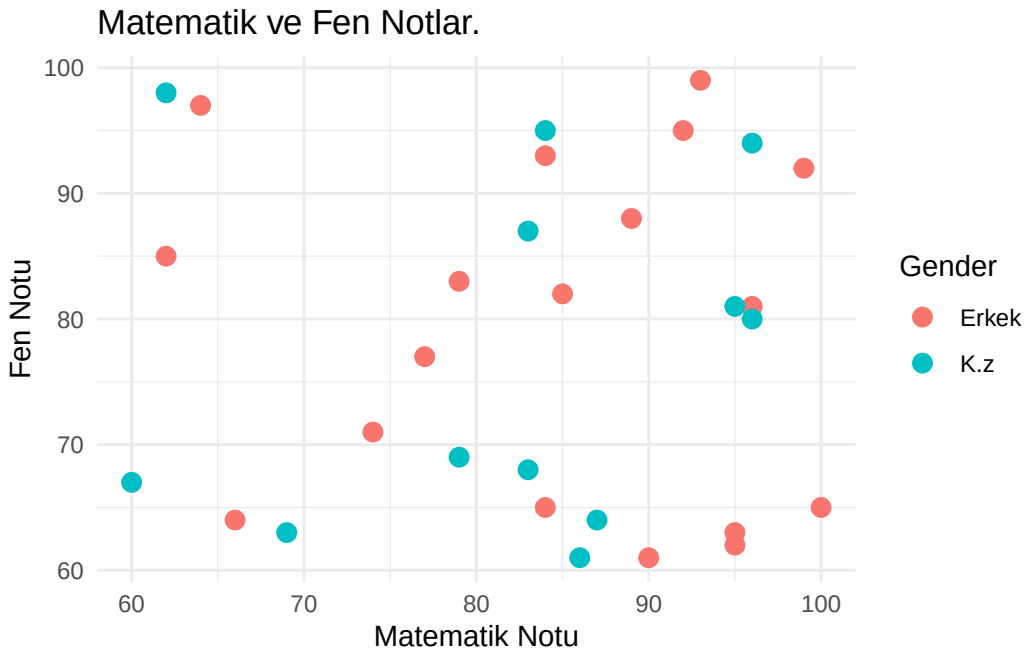
1. **Grafik Düzeyinde (Global Aesthetics)**:
    - Global estetikler, `ggplot()` fonksiyonu içinde tanımlanır ve tüm grafik boyunca geçerli olur.
    - Bu, grafik genelinde belirli bir estetiği ayarlamak istediğinizde kullanışlıdır. Örneğin, tüm noktaların belirli bir renkte veya boyutta görünmesini istiyorsanız, bunu grafik düzeyinde tanımlayabilirsiniz.

```r
# Gerekli kütüphaneleri yükleyin
library(dplyr)

# Örnek veri setini oluşturun
set.seed(42)  # Sonuçların tekrarlanabilir olması için
students <- data.frame(
  Name = paste("Öğrenci", 1:30),
  Math = sample(60:100, 30, replace = TRUE),  # Rastgele matematik notları
  Science = sample(60:100, 30, replace = TRUE),  # Rastgele fen notları
  Gender = sample(c("Kız", "Erkek"), 30, replace = TRUE)  # Rastgele cinsiyetler
```

```r
)

# Global estetik ile grafik oluşturma
ggplot(data = students, aes(x = Math, y = Science, color = Gender)) +
  geom_point(size = 3) +  # Tüm noktaların boyutu
  labs(title = "Matematik ve Fen Notları", x = "Matematik Notu", y = "Fen Notu") +
  theme_minimal()
```

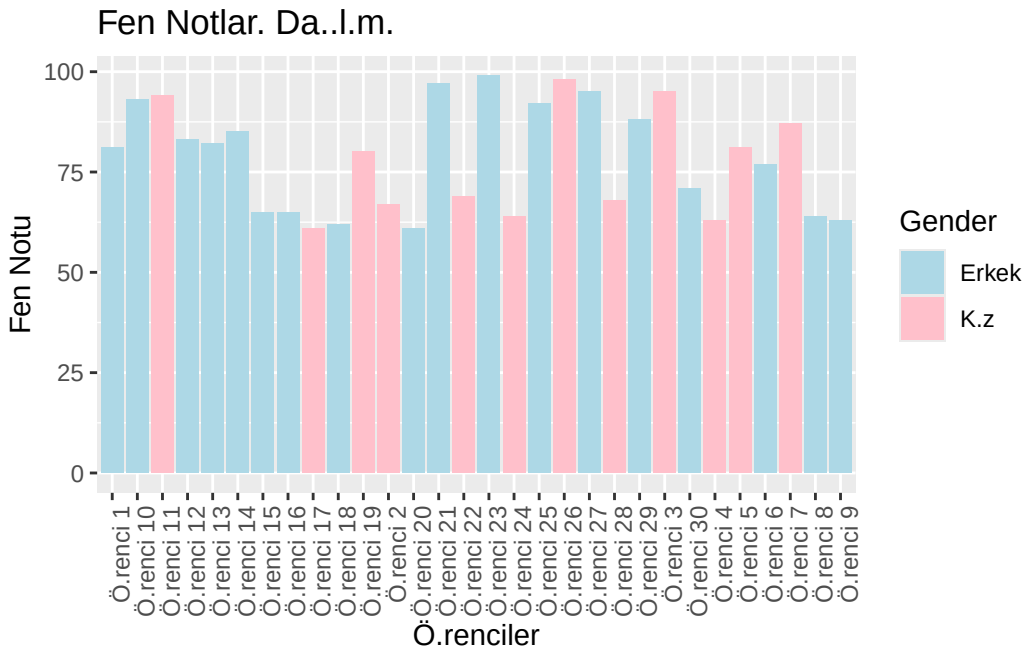


Global Estetik Kullanımı: İlk grafik, tüm öğrencilerin matematik ve fen notlarını gösterirken, noktaların rengi cinsiyete göre belirlenmiştir. Bu, her bir cinsiyet için farklı bir renk atayarak izleyicinin veri gruplarını kolayca ayırt etmesini sağlar.

2. **Öğe Düzeyinde (Local Aesthetics)**:

- Yerel estetikler, belirli bir geometrik öğe (örneğin, çubuklar, noktalar, çizgiler) için ayrı olarak tanımlanabilir.
- Bu, belirli bir grafik öğesine özgü estetik değişiklikler yapmak istediğinizde kullanışlıdır. Örneğin, sadece bir çubuk grafiğindeki çubukların rengini değiştirmek istiyorsanız, bunu yerel estetiklerle ayarlayabilirsiniz.

```r
# Yerel estetik ile çubuk grafiği oluşturma
ggplot(data = students, aes(x = Name, y = Science, fill = Gender)) +
  geom_bar(stat = "identity") +  # Her öğrencinin fen notunu çubuk olarak göster
  labs(title = "Fen Notları Dağılımı", x = "Öğrenciler", y = "Fen Notu") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  # Eksen etiketlerini döndür
  scale_fill_manual(values = c("Kız" = "pink", "Erkek" = "lightblue"))  # Cinsiyete göre özel
```



Yerel Estetik Kullanımı: İkinci grafik ise her öğrencinin fen notlarını gösteren bir çubuk grafiğidir. Bu grafikte, cinsiyete göre çubukların dolgu rengi ayarlanmıştır. Böylece, her öğrencinin cinsiyeti görsel olarak temsil edilirken, çubukların boyutları da fen notlarını temsil eder.

Küçük Çoklu Grafikleri ve Katmanlar

Veri grafiklerini oluşturmanın temel zorluklarından biri, çok değişkenli bilgileri iki boyutlu bir görüntüye sıkıştırmaktır. Üç boyutlu görüntüler bazen yararlı olsa da, genellikle daha fazla kafa karıştırıcıdır. Bunun yerine, iki boyutlu bir veri grafiğine daha fazla değişken eklemenin üç yaygın yolu şunlardır:

- **Küçük Çoklu Grafikler**: Ayrıca fasetler olarak da bilinir, tek bir veri grafiği, aynı temel grafiğin birkaç küçük çoklu grafiğinden oluşabilir; her bir küçük alt görüntüde bir (ayrık) değişken değişmektedir.

- **Katmanlar**: Mevcut bir veri grafiğinin üzerine yeni bir katman çizmek bazen uygundur. Bu yeni katman bağlam veya karşılaştırma sağlayabilir, ancak insanların güvenilir bir şekilde ayrıştırabileceği katman sayısında bir sınırlama vardır.
- **Animasyon**: Zaman ek değişken ise, o zaman bir animasyon bazen o değişkenin değişimlerini etkili bir şekilde iletebilir. Elbette, bu basılı sayfada işe yaramaz ve kullanıcının tüm verileri aynı anda görmesini imkansız kılar.

Biraz pratik yaparak, yukarıda ana hatlarıyla belirtilen sınıflandırmaya göre veri grafiklerini incelemeyi öğrenebilirsiniz. Örneğin, temel bir dağılım grafiğiniz, iki değişken arasındaki ilişkiyi göstermek için kartezyen düzlemde konum kullanır ve lineer ölçeklere sahiptir.

4. ggplot2 ile Basit Bir Dağılım Grafiği Oluşturma

Bu bölümde, basit bir veri seti oluşturarak ggplot2 ile nasıl daha karmaşık grafikler oluşturabileceğinizi göstereceğiz.

Örnek 1

Bu örnek için basit bir veri seti oluşturalım. Bu veri seti, beş öğrencinin matematik ve fen derslerindeki notlarını içerecek:

```
# Örnek veri setini oluşturun
set.seed(42)  # Sonuçların tekrarlanabilir olması için
students <- data.frame(
  Name = paste("Öğrenci", 1:50),
  Math = sample(60:100, 50, replace = TRUE),  # Rastgele matematik notları
  Science = sample(60:100, 50, replace = TRUE),  # Rastgele fen notları
  Gender = sample(c("Kız", "Erkek"), 50, replace = TRUE),  # Rastgele cinsiyetler
  Age = sample(15:20, 50, replace = TRUE),  # Rastgele yaşlar
  Course = sample(c("Matematik", "Fen", "Kimya"), 50, replace = TRUE)  # Rastgele kurslar
)

# Veri setini görüntüleyin
print(students)
```

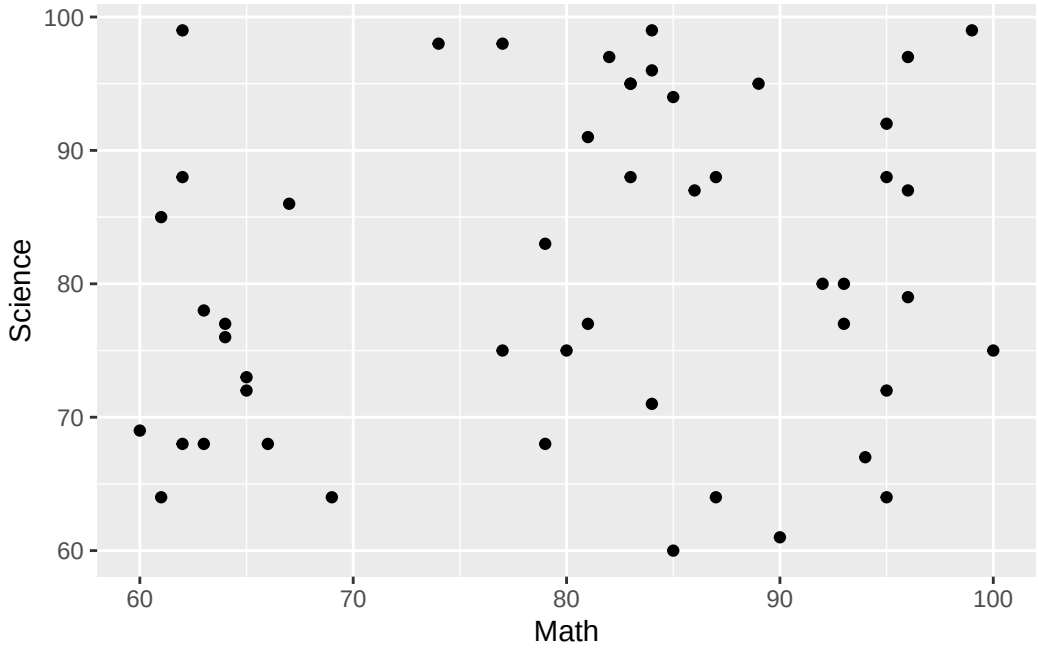
```
     Name Math Science Gender Age    Course
```

1 Öğrenci 1 96 97 Kız 16 Kimya 2 Öğrenci 2 60 69 Erkek 16 Matematik 3 Öğrenci 3 84 99 Erkek 15 Matematik 4 Öğrenci 4 69 64 Erkek 18 Matematik 5 Öğrenci 5 95 92 Kız 17 Matematik 6 Öğrenci 6 77 98 Kız 19 Matematik 7 Öğrenci 7 83 95 Kız 20 Fen 8 Öğrenci 8 66 68 Kız 20

Kimya 9 Öğrenci 9 95 88 Kız 16 Kimya 10 Öğrenci 10 84 71 Kız 18 Kimya 11 Öğrenci 11 96 79 Kız 16 Matematik 12 Öğrenci 12 79 68 Erkek 17 Matematik 13 Öğrenci 13 85 94 Kız 16 Matematik 14 Öğrenci 14 62 88 Erkek 15 Fen 15 Öğrenci 15 100 75 Erkek 18 Kimya 16 Öğrenci 16 84 96 Kız 20 Kimya 17 Öğrenci 17 86 87 Erkek 19 Matematik 18 Öğrenci 18 95 64 Erkek 16 Kimya 19 Öğrenci 19 96 87 Kız 15 Fen 20 Öğrenci 20 90 61 Kız 15 Kimya 21 Öğrenci 21 64 77 Erkek 19 Fen 22 Öğrenci 22 79 83 Kız 17 Kimya 23 Öğrenci 23 93 77 Erkek 20 Kimya 24 Öğrenci 24 87 64 Kız 20 Matematik 25 Öğrenci 25 99 99 Erkek 18 Matematik 26 Öğrenci 26 62 99 Erkek 16 Matematik 27 Öğrenci 27 92 80 Kız 15 Kimya 28 Öğrenci 28 83 95 Erkek 19 Fen 29 Öğrenci 29 89 95 Kız 17 Matematik 30 Öğrenci 30 74 98 Erkek 20 Kimya 31 Öğrenci 31 81 77 Erkek 16 Kimya 32 Öğrenci 32 67 86 Erkek 19 Matematik 33 Öğrenci 33 95 72 Erkek 17 Matematik 34 Öğrenci 34 63 78 Erkek 18 Matematik 35 Öğrenci 35 81 91 Erkek 16 Fen 36 Öğrenci 36 77 75 Kız 15 Matematik 37 Öğrenci 37 87 88 Erkek 17 Matematik 38 Öğrenci 38 64 76 Erkek 17 Fen 39 Öğrenci 39 63 68 Erkek 20 Kimya 40 Öğrenci 40 93 80 Kız 18 Matematik 41 Öğrenci 41 94 67 Erkek 18 Fen 42 Öğrenci 42 83 88 Erkek 18 Fen 43 Öğrenci 43 82 97 Kız 17 Fen 44 Öğrenci 44 85 60 Erkek 15 Fen 45 Öğrenci 45 65 72 Erkek 15 Kimya 46 Öğrenci 46 65 73 Erkek 15 Fen 47 Öğrenci 47 61 64 Erkek 20 Kimya 48 Öğrenci 48 62 68 Erkek 19 Fen 49 Öğrenci 49 80 75 Kız 20 Matematik 50 Öğrenci 50 61 85 Kız 18 Matematik

Şimdi, bu veri setini kullanarak basit bir dağılım grafiği oluşturalım. İlk olarak, sadece pozisyon estetiğini kullanacağız:

```
# Dağılım grafiği oluşturma
ggplot(data = students, aes(x = Math, y = Science)) +
  geom_point()
```



```
  labs(title = "Öğrencilerin Matematik ve Fen Notları",
       x = "Matematik Notu",
       y = "Fen Notu") +
  theme_minimal()
```

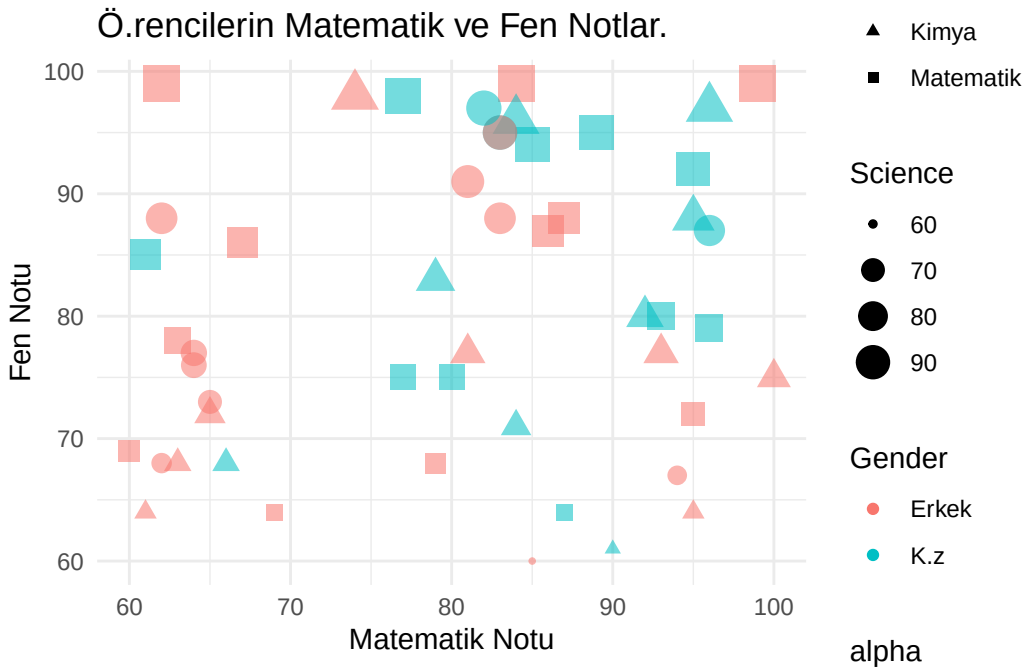
NULL

- **`ggplot(data = students, aes(x = Math, y = Science))`**: Burada `students` veri setini kullanıyoruz ve `Math` değişkenini x eksenine, `Science` değişkenini y eksenine atıyoruz.
- **`geom_point()`**: Noktaları grafiğe eklemek için kullanılır.
- **`labs()`**: Başlık ve eksen etiketlerini ayarlamak için kullanılır.

Grafiğe ek Estetik (asthetics) ekleme

Ek estetik (aesthetics) eklemek, ggplot2'de bir grafik oluştururken görsel unsurları zenginleştirmek ve verilerinizi daha etkili bir şekilde sunmak için kullanılır. Ek estetikler, grafiklerde daha fazla bilgi vermek ve verilerin daha iyi anlaşılmasını sağlamak için grafik öğelerine uygulanabilir. Burada, her öğrencinin ismini noktalara ekleyeceğiz:

```
# Ek estetikler ile Dağılım grafiğini zenginleştirme
ggplot(data = students, aes(x = Math, y = Science)) +
  geom_point(aes(color = Gender, size = Science, shape = Course, alpha = 0.7)) +  # Ek esteti
  labs(title = "Öğrencilerin Matematik ve Fen Notları",
       x = "Matematik Notu",
       y = "Fen Notu") +
  theme_minimal()
```



- `color = Gender`: Noktaların rengini cinsiyete göre ayarlıyoruz.
- `size = Science`: Noktaların boyutunu fen notuna göre ayarlıyoruz. Daha yüksek fen notları, daha büyük noktalar anlamına gelir.
- `shape = Course`: Alınan kursa göre noktaların şeklini ayarlıyoruz. Örneğin, Matematik, Fen ve Kimya kursları farklı şekillerde gösterilecektir.
- `alpha = 0.7`: Noktaların saydamlık düzeyini ayarlıyoruz. Bu, yoğunluk yüksek olan bölgelerdeki noktaların daha görünür olmasını sağlar.

Ek Estetiklerin Kullanım Mantığı:

- **Veri Görselleştirmeyi Zenginleştirme**: Ek estetikler, grafiklerinizi zenginleştirir ve izleyicilere daha fazla bilgi sunar. Örneğin, bir scatter plot'ta noktaların büyüklüğünü veya rengini değiştirerek verinin belirli yönlerini vurgulayabilirsiniz.
- **Veri Gruplarını Belirginleştirme**: Kategorik değişkenleri kullanarak, verileri gruplara ayırabilir ve bu gruplar arasındaki farkları net bir şekilde göstererek anlamayı kolaylaştırabilirsiniz. Örneğin, farklı renkler kullanarak bir cinsiyet grubunu veya bir yaş grubunu belirginleştirebilirsiniz.
- **Karmaşık İlişkileri Gösterme**: Ek estetikler, karmaşık ilişkilerin daha iyi anlaşılmasına yardımcı olur. Örneğin, hem renk hem de boyut estetiğini kullanarak bir değişkenin diğerine olan etkisini daha net bir şekilde gösterebilirsiniz.

Limitler ve Dikkat Edilmesi Gerekenler

1. **Aşırı Karmaşıklaşma**: Grafiklerde çok fazla estetik eklemek, grafiği karmaşık hale getirebilir ve izleyicinin veri ile ilgili ana mesajı anlamasını zorlaştırabilir. Basit ve net grafikler genellikle daha etkilidir.
2. **Görsellik ve Anlam Kaybı**: Farklı estetiklerin bir arada kullanılması, görselliği artırırken, aynı zamanda verinin anlamını da kaybettirebilir. Estetiklerin doğru bir şekilde ayarlanması, grafiklerin net ve anlaşılır olmasını sağlamalıdır.
3. **Veri Tipine Uygunluk**: Her estetik, belirli bir veri türü ile uyumludur. Sayısal veriler için uygun estetikler kullanılmalıdır (örneğin, `size`, `alpha` gibi). Kategorik veriler içinse `fill`, `color`, `shape` gibi estetikler kullanılmalıdır. Uygun estetiklerin seçilmesi, verilerin doğru bir şekilde görselleştirilmesini sağlar.
4. **Okunabilirlik**: Ek estetikler, grafiklerin okunabilirliğini etkileyebilir. Özellikle renk ve boyut gibi özelliklerin aşırı kullanımı, izleyicinin grafikteki detayları algılamasını zorlaştırabilir. Renk paletleri ve yazı tipleri gibi unsurların dikkatli bir şekilde seçilmesi gereklidir.

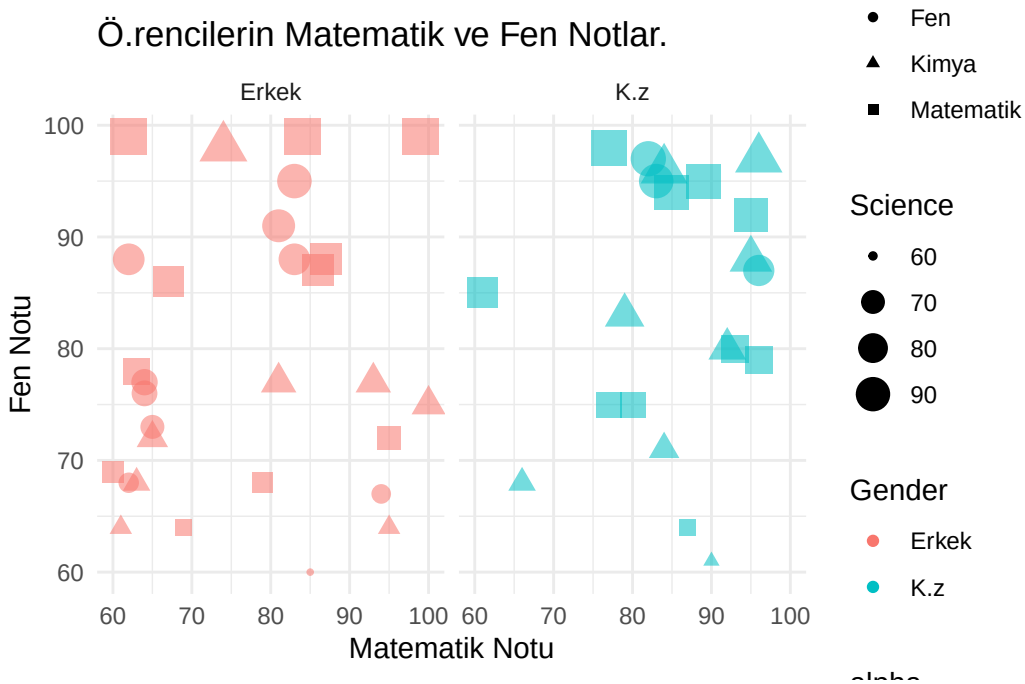
Grafiğe Faset (Facet) ekleme

Fasetler (facets), bir grafikte farklı kategorilere göre alt grafikler oluşturmanıza olanak tanır. Bu, belirli bir değişkenin etkisini ve kategoriler arasındaki farklılıkları daha iyi görselleştirmek için oldukça kullanışlıdır. ggplot2'de `facet_wrap()` ve `facet_grid()` fonksiyonları kullanılarak fasetler eklenebilir.

Kategorik Değişkenle Faset Ekleme

Aşağıda, grafiklerimize faset ekleyerek öğrencilerin notlarını cinsiyete göre ayıracağız. Bu, aynı grafikte kız ve erkek öğrencilerin notlarını ayrı ayrı görselleştirmemizi sağlar.

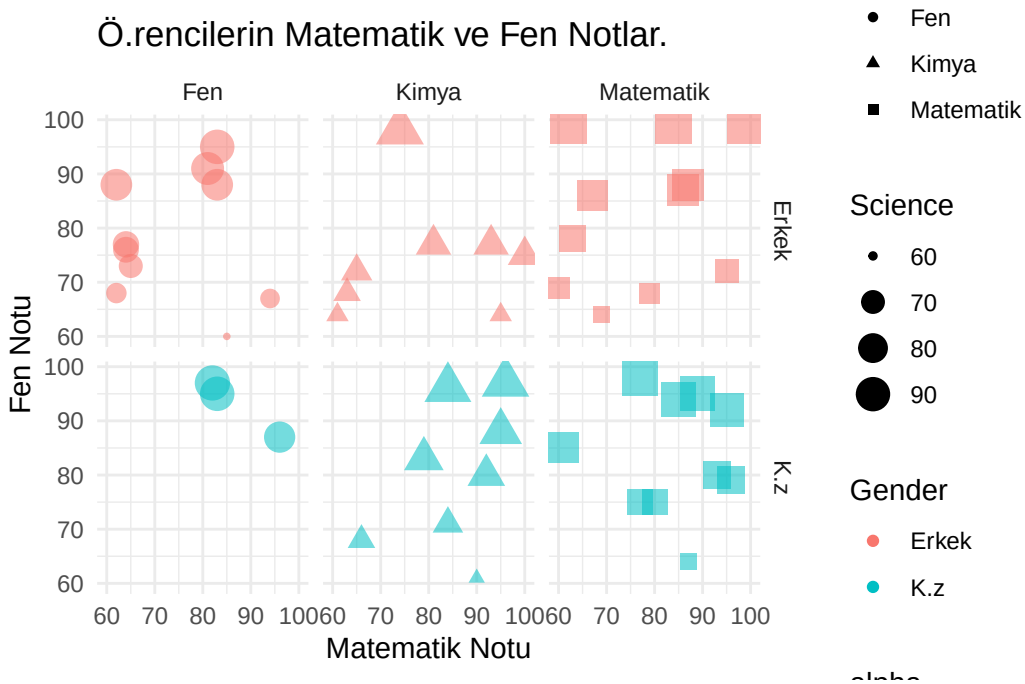
```
# Dağılım grafiği oluşturma ve faset ekleme
ggplot(data = students, aes(x = Math, y = Science)) +
  geom_point(aes(color = Gender, size = Science, shape = Course, alpha = 0.7)) +  # Ek estet
  labs(title = "Öğrencilerin Matematik ve Fen Notları",
       x = "Matematik Notu",
       y = "Fen Notu") +
  theme_minimal() +
  facet_wrap(~ Gender)  # Cinsiyete göre fasetleme
```



- **`facet_wrap(~ Gender)`**: Bu komut, grafikte cinsiyet değişkenine göre ayrı alt grafikler oluşturur. Her cinsiyet için ayrı bir dağılım grafiği çizilir.

Eğer daha karmaşık bir yapı isterseniz, aynı anda birden fazla değişkene göre fasetleme yapabilirsiniz. Örneğin, hem cinsiyete hem de kursa göre fasetlemek:

```
# Cinsiyete ve kursa göre fasetleme
ggplot(data = students, aes(x = Math, y = Science)) +
  geom_point(aes(color = Gender, size = Science, shape = Course, alpha = 0.7)) +  # Ek estet
  labs(title = "Öğrencilerin Matematik ve Fen Notları",
       x = "Matematik Notu",
       y = "Fen Notu") +
  theme_minimal() +
  facet_grid(Gender ~ Course)  # Cinsiyet ve kursa göre 2 boyutlu fasetleme
```



Sayısal Değişkene Göre Fasetleme

ggplot2'de bir sayısal değişkene göre fasetleme yapmak, veri görselleştirme sırasında belirli bir aralıkta bulunan verileri alt grafikler halinde düzenlemek için kullanışlıdır. Örneğin, öğrencilerin yaşlarına göre dağılım grafikleri oluşturabiliriz. Aşağıda, sayısal değişkene göre nasıl fasetleme yapılacağını gösteren bir örnek bulunmaktadır.

Örnek Veri Seti

Daha önce oluşturduğumuz `students` veri setini kullanacağız. Bu veri seti, öğrencilerin matematik ve fen notlarını, cinsiyetlerini, yaşlarını ve katıldıkları kursları içermektedir.

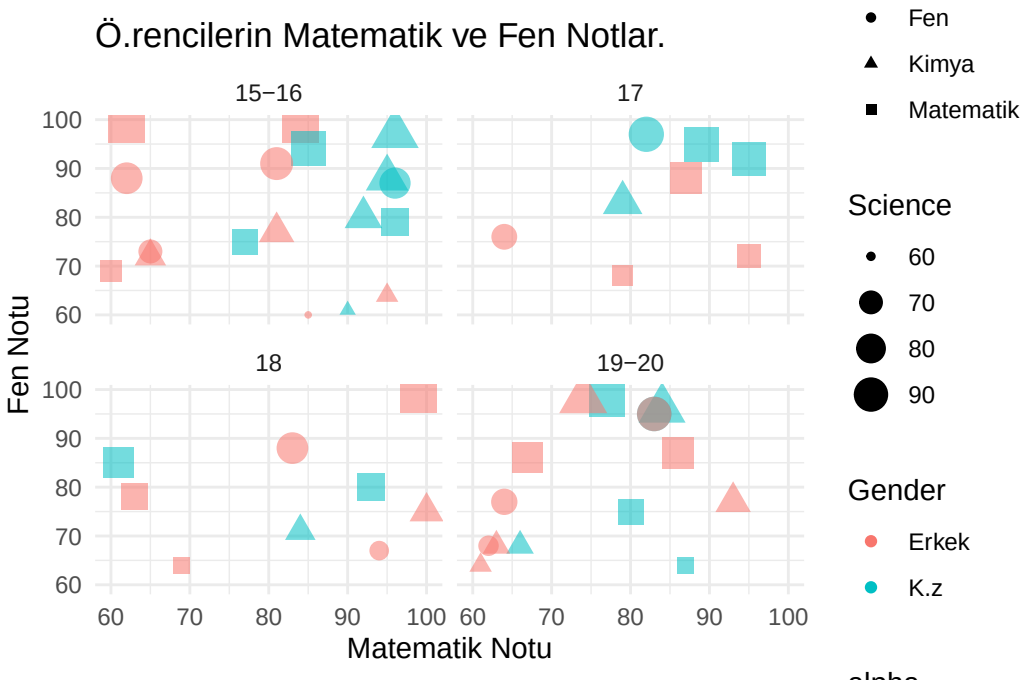
Yaş Aralıklarına Göre Fasetleme

Öncelikle, yaşları belirli gruplara ayıralım. Daha sonra, her yaş grubuna göre dağılım grafiği oluşturacağız.

```
# Gerekli kütüphaneleri yükleyin
library(ggplot2)
library(dplyr)
```

```r
# Yaş gruplarını oluşturma
students <- students %>%
  mutate(AgeGroup = case_when(
    Age < 17 ~ "15-16",
    Age < 18 ~ "17",
    Age < 19 ~ "18",
    TRUE ~ "19-20"
  ))

# Dağılım grafiği oluşturma ve yaş gruplarına göre fasetleme
ggplot(data = students, aes(x = Math, y = Science)) +
  geom_point(aes(color = Gender, size = Science, shape = Course, alpha = 0.7)) +  # Ek estet
  labs(title = "Öğrencilerin Matematik ve Fen Notları",
       x = "Matematik Notu",
       y = "Fen Notu") +
  theme_minimal() +
  facet_wrap(~ AgeGroup)  # Yaş grubuna göre fasetleme
```



- **Yaş Grupları**: `mutate()` fonksiyonu ile her öğrencinin yaşı, belirli gruplara ayrılmıştır. `case_when()` kullanılarak yaş aralıkları oluşturulmuştur.
- **`facet_wrap(~ AgeGroup)`**: Bu komut, grafiklerin yaş gruplarına göre ayrı alt grafikler oluşturmasını sağlar. Her yaş grubu için bir dağılım grafiği çizilir.

Örnek 2

Bu örnekte, ekonomik verimlilikle ilgili soruları yanıtlamak için ilgili ölçümleri içeren bir veri seti olan CIACountries veri setini kullanacağız. Aşağıda, CIACountries veri setini kullanarak ggplot2 ile grafikler oluşturacağız. Her bir grafik örneğinde, estetikler, görsel ipuçları, koordinat sistemleri, ölçek ve bağlam kullanımı ile ilgili açıklamalara yer vereceğiz.

Örnek Veri Seti

Öncelikle CIACountries veri setinin yapılandırılması gerekmektedir. Bu veri setinde aşağıdaki değişkenler bulunmaktadır:

- `pop`: Nüfus
- `area`: Alan
- `gdp`: Gayri safi yurt içi hasıla
- `educ`: Eğitime harcanan GSYİH yüzdesi
- `roadways`: Birim alandaki yol uzunluğu
- `net_users`: İnternet kullanım oranı
- `oil_prod`: Günlük petrol üretimi (varil)

Örnek Veri Setini Yükleme

Aşağıdaki örnekte, CIACountries veri setinin oluşturulmuş bir versiyonunu kullanacağız:

```
# Gerekli kütüphaneleri yükleyin
library(mosaicData)

# CIACountries veri setini yükleyin
data("CIACountries")

# Veri setinin ilk birkaç satırını görüntüleyin
head(CIACountries)
```

country	pop	area	oil_prod	gdp	educ	roadways	net_users
Afghanistan	32564342	652230	0	1900	NA	0.0646244	>5%
Albania	3029278	28748	20510	11900	3.3	0.6261305	>35%
Algeria	39542166	2381741	1420000	14500	4.3	0.0477193	>15%

country	pop	area	oil_prod	gdp	educ	roadways	net_users
American Samoa	54343	199	0	13000	NA	1.2110553	NA
Andorra	85580	468	NA	37200	NA	0.6837607	>60%
Angola	19625353	1246700	1742000	7300	3.5	0.0412521	>15%

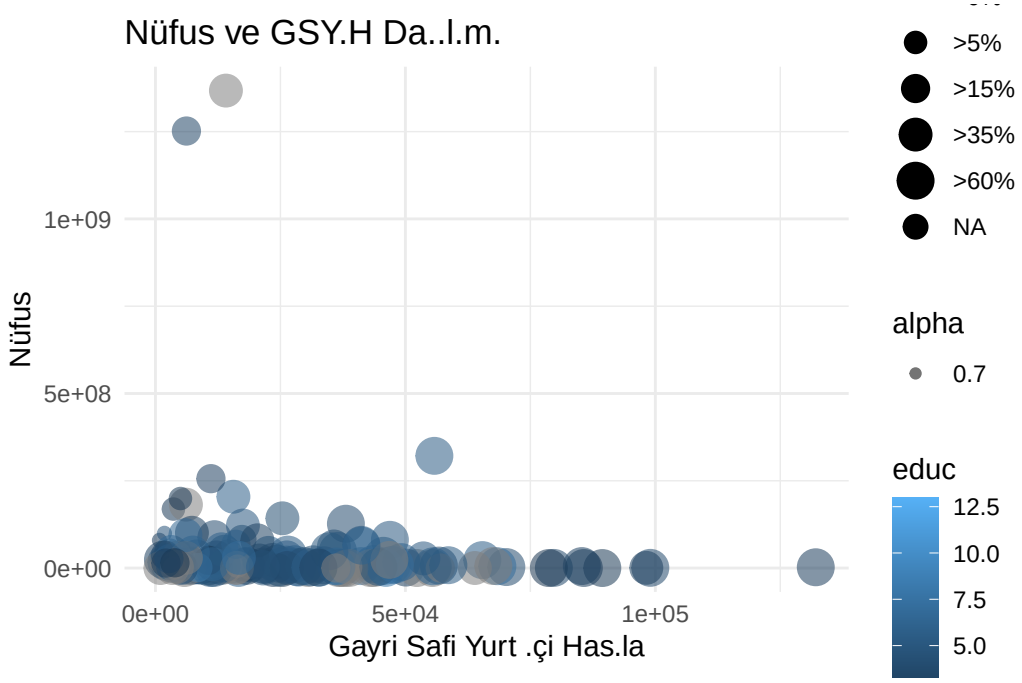
1. Estetikler ve Görsel İpuçları Kullanımı

Aşağıdaki grafikte, `gdp` ve `pop` değişkenlerini kullanarak bir dağılım grafiği oluşturalım. Bu grafikte `color` estetiği ile `educ` değişkenine göre noktaların rengini ayarlayacağız.

```
# Dağılım grafiği oluşturma
ggplot(data = CIACountries, aes(x = gdp, y = pop)) +
  geom_point(aes(color = educ, size = net_users, alpha = 0.7)) +  # Ek estetikler
  labs(title = "Nüfus ve GSYİH Dağılımı",
       x = "Gayri Safi Yurt İçi Hasıla",
       y = "Nüfus") +
  theme_minimal()
```

```
Warning: Using size for a discrete variable is not advised.
```

```
Warning: Removed 21 rows containing missing values or values outside the scale range
(`geom_point()`).
```

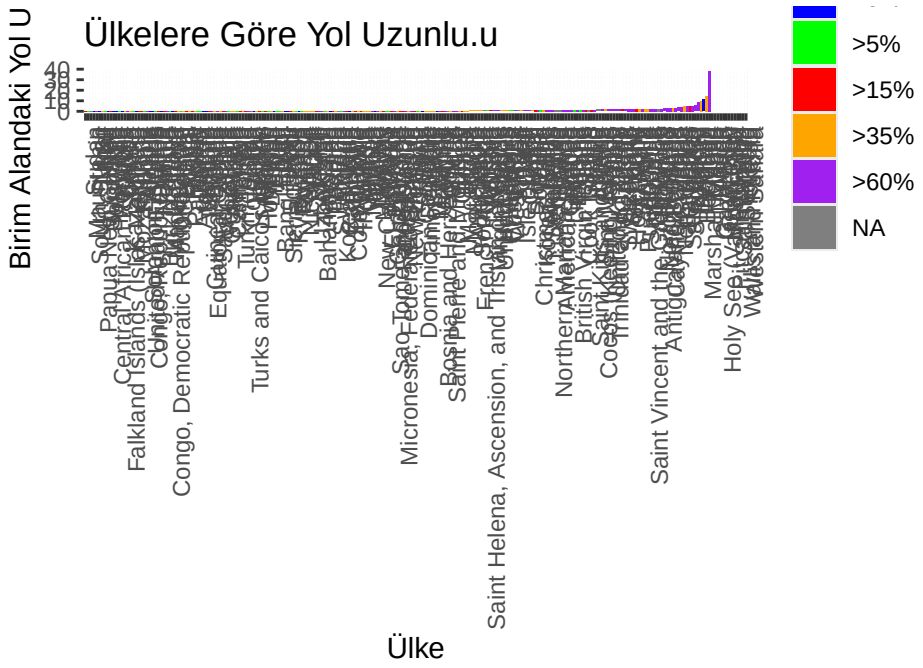


2. Koordinat Sistemleri ve Ölçek Kullanımı

Aşağıdaki grafikte, `roadways` değişkenini kullanarak bir çubuk grafiği oluşturalım. Bu grafikte `gdp` ve `pop` değişkenleri ile birlikte `fill` estetiği ekleyelim.

```
# Çubuk grafiği oluşturma
ggplot(data = CIACountries, aes(x = reorder(country, roadways), y = roadways)) +
  geom_col(aes(fill = factor(net_users))) +  # Dolgu rengi internet kullanıcılarına göre
  labs(title = "Ülkelere Göre Yol Uzunluğu",
       x = "Ülke",
       y = "Birim Alandaki Yol Uzunluğu") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  # X eksenindeki metni döndür
  scale_fill_manual(values = c("blue", "green", "red", "orange", "purple"),
                    name = "İnternet Kullanımı")  # Kategorik renkler
```

```
Warning: Removed 13 rows containing missing values or values outside the scale range
(`geom_col()`).
```

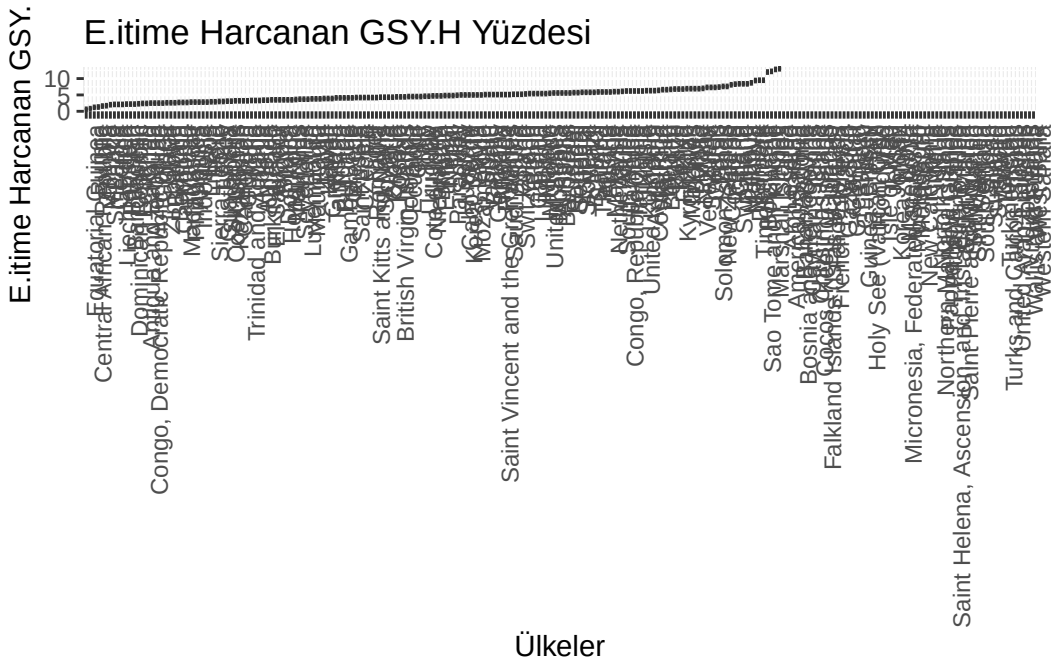


3. Bağlam Kullanımı

Grafiklerimize bağlam eklemek, verilerin anlamını artırır. Aşağıdaki grafikte, her bir grafikte başlık, eksen etiketleri ve açıklamalar kullanarak bağlam sağlayalım.

```
# Kutu grafiği oluşturma
ggplot(data = CIACountries, aes(x = reorder(country, educ), y = educ)) +
  geom_boxplot(fill = "lightblue") +  # Kutu grafiği
  labs(title = "Eğitime Harcanan GSYİH Yüzdesi",
       x = "Ülkeler",
       y = "Eğitime Harcanan GSYİH Yüzdesi") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # X eksenindeki metni döndür
```

```
Warning: Removed 63 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



5. R'de Kanonik Veri Grafikleri

Her veri bilimcisinin bilmesi ve yorumlaması gereken birkaç veri grafiği vardır.Aşağıdaki tablo her istatiksel veri analizi ile uğraşan kişinin bilmesi gereken temel grafik türlerini, bunların kullanım alanlarını, hangi veri türlerinin kullanıldığını, ggplot estetiklerini ve olası diğer unsurları özetlemektedir:

Tablo 3: Temel Grafik Türleri

Grafik Türü	Tanım	Kullanım	Veri Türü	ggplot Estetikleri	Olası Diğer Unsurlar
Tek Değişkenli Grafikler					
Histogram	Sayısal verilerin dağılımını gösteren çubuk grafiği.	Sürekli değişkenlerin dağılımını analiz etmek için.	Sürekli sayısal	`x`, `fill`	Bin genişliği, eksen etiketleri

Grafik Türü	Tanım	Kullanım	Veri Türü	ggplot Estetikleri	Olası Diğer Unsurlar
Kutu Grafiği	Veri setinin özet istatistiklerini gösteren grafik.	Verinin dağılımını ve aykırı değerleri incelemek için.	Sürekli sayısal, kategorik	`x`, `y`, `fill`	Medyan, çeyrekler, aykırı değerler
Pasta Grafiği	Kategorik verilerin toplam içindeki oranlarını gösteren grafik.	Kategorilerin birbirine oranlarını göstermek için.	Kategorik	`fill`	Efsane, etiketler
Yoğunluk Grafiği	Sayısal verinin sürekli dağılımını gösteren grafik.	Verinin dağılımını görselleştirmek için.	Sürekli sayısal	`x`, `fill`	Band genişliği, eksen etiketleri
Çubuk Grafiği	Kategorik verilerin sayısını veya değerlerini gösteren grafik.	Kategorik verilerin karşılaştırılmasında.	Kategorik, sayısal	`x`, `y`, `fill`	Eksende döndürme, etiketler
Çok Değişkenli Grafikler					
Dağılım Grafiği	İki sayısal değişken arasındaki ilişkiyi gösteren grafik.	Değişkenler arasındaki ilişkiyi keşfetmek için.	Sürekli sayısal	`x`, `y`, `color`, `size`, `shape`	Eğilim çizgisi, eksen etiketleri
Zaman Serisi Grafiği	Zamanla değişen verilerin görselleştirildiği grafik.	Zaman içindeki eğilimleri analiz etmek için.	Sürekli sayısal, tarih	`x`, `y`, `color`	Eksen etiketleri, eğilim çizgisi

Grafik Türü	Tanım	Kullanım	Veri Türü	ggplot Estetikleri	Olası Diğer Unsurlar
Mozaik Grafiği	İki veya daha fazla kategorik değişkenin ilişkisini gösteren grafik.	Kategorik değişkenlerin bağıntısını analiz etmek için.	Kategorik	`fill`, `x`, `y`	Efsane, etiketler
Isı Haritası	İki değişken arasındaki ilişkiyi renklerle gösteren grafik.	Veri yoğunluğunu ve ilişkilerini görselleştirmek için.	Sürekli sayısal, kategorik	`x`, `y`, `fill`	Renk paleti, eksen etiketleri

Açıklama

- **Tek Değişkenli Grafikler**: Sadece bir değişkenin analiz edildiği grafiklerdir. Örneğin, histogram ve kutu grafiği tek değişkenli grafiklerdir.
- **Çok Değişkenli Grafikler**: İki veya daha fazla değişkenin birlikte analiz edildiği grafiklerdir. Dağılım grafiği, zaman serisi grafiği, mozaik grafiği ve ısı haritası çok değişkenli grafiklerdir.

Tek Değişkenli Görselleştirmeler

Sayısal Değişkenler

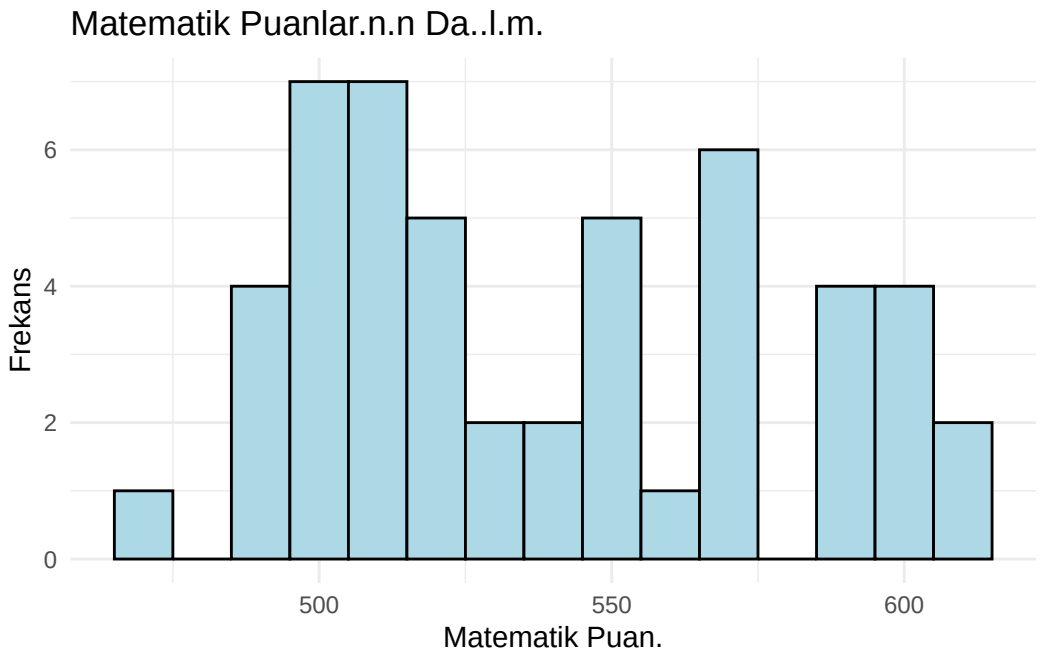
Tek bir değişkenin dağılımını anlamak genellikle faydalıdır. Eğer değişken sayısal ise, dağılımı genellikle bir histogram veya yoğunluk grafiği ile grafiksel olarak özetlenir. Her ikisi de aynı bilgiyi iletmekle birlikte, histogram önceden tanımlanmış aralıklar (bins) kullanarak kesikli bir dağılım oluştururken, yoğunluk grafiği bir kernel düzleştirici kullanarak sürekli bir eğri oluşturur.

Histogram

Aşağıdaki kod, bir histogram oluşturmak için ggplot2 kullanır. Bu histogram, her eyaletin matematik SAT puanlarının dağılımını göstermektedir:

```r
# Gerekli kütüphaneleri yükleyin
library(ggplot2)

# Histogram oluşturma
ggplot(SAT_2010, aes(x = math)) +
  geom_histogram(binwidth = 10, fill = "lightblue", color = "black") +
  labs(title = "Matematik Puanlarının Dağılımı",
       x = "Matematik Puanı",
       y = "Frekans") +
  theme_minimal()
```



Yorum:

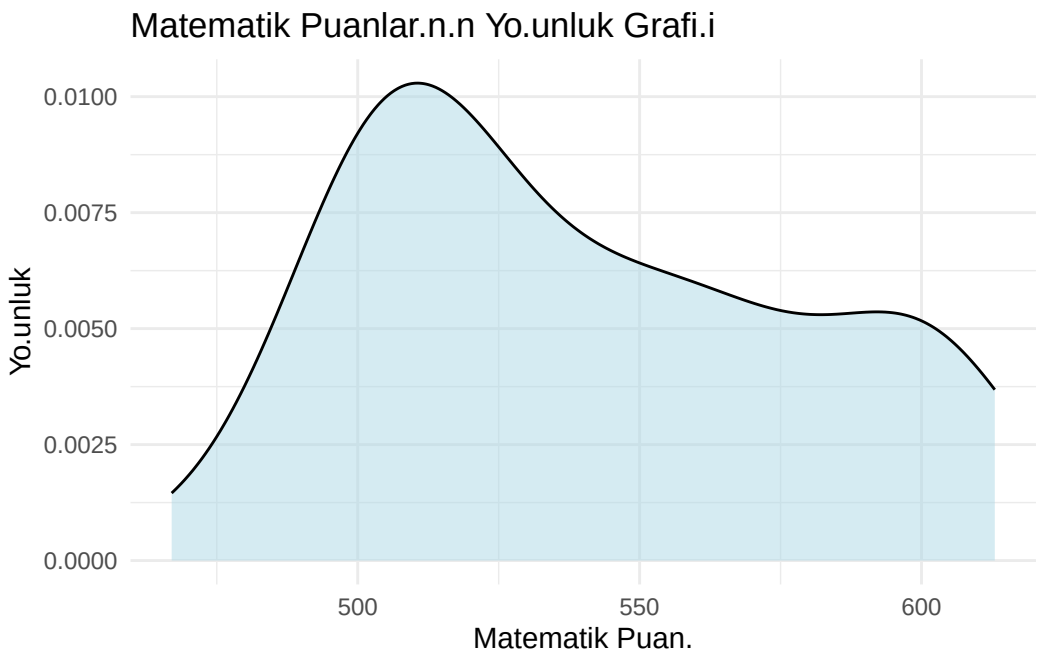
- Histogram, matematik puanlarının dağılımını göstermektedir.
- **Veri Dağılımı**: Histogramda görülen çubukların yüksekliği, belirli puan aralıklarında (bin) kaç öğrencinin olduğunu gösterir. Örneğin, 600-610 puan aralığında birçok öğrenci varsa, bu aralığın çubuğu daha yüksek olacaktır.
- **Aykırı Değerler**: Eğer histogramın uçlarında (yüksek veya düşük puanlarda) çok az sayıdaki öğrencinin puanı varsa, bu aykırı değerler hakkında bilgi verebilir.
- **Bin Genişliği**: `binwidth` parametresi, histogramın her bir çubuğunun genişliğini ayarlar. Daha küçük bir bin genişliği, daha detaylı bir dağılım gösterirken; daha

büyük bir bin genişliği, genel eğilimleri gösterir. En uygun bin genişliği, verinin dağılımına bağlı olarak deneme-yanılma yoluyla belirlenmelidir. Genellikle, Sturges veya Freedman-Diaconis kuralı gibi yöntemler kullanılabilir. Yukarıda verilen bağlamda, her bir aralık, 10 puanlık bir SAT puanı aralığını kapsar. Histogramın görünümünün, seçilen aralık genişliklerine bağlı olarak önemli ölçüde değişebileceğini unutmamak önemlidir; dolayısıyla evrensel olarak en iyi seçim yoktur. Veriniz için en uygun aralık genişliğini belirlemek, veri setinin özelliklerini göz önünde bulundurarak bireysel bir değerlendirme gerektirir.

Yoğunluk Grafiği

Aşağıdaki kod, yoğunluk grafiğini oluşturmak için kullanılır. Bu grafik, her eyaletin ortalama matematik SAT puanlarının dağılımını gösterir:

```
# Yoğunluk grafiği kodu
ggplot(SAT_2010, aes(x = math)) +
  geom_density(fill = "lightblue", alpha = 0.5) +
  labs(title = "Matematik Puanlarının Yoğunluk Grafiği",
       x = "Matematik Puanı",
       y = "Yoğunluk") +
  theme_minimal()
```



Yorum:

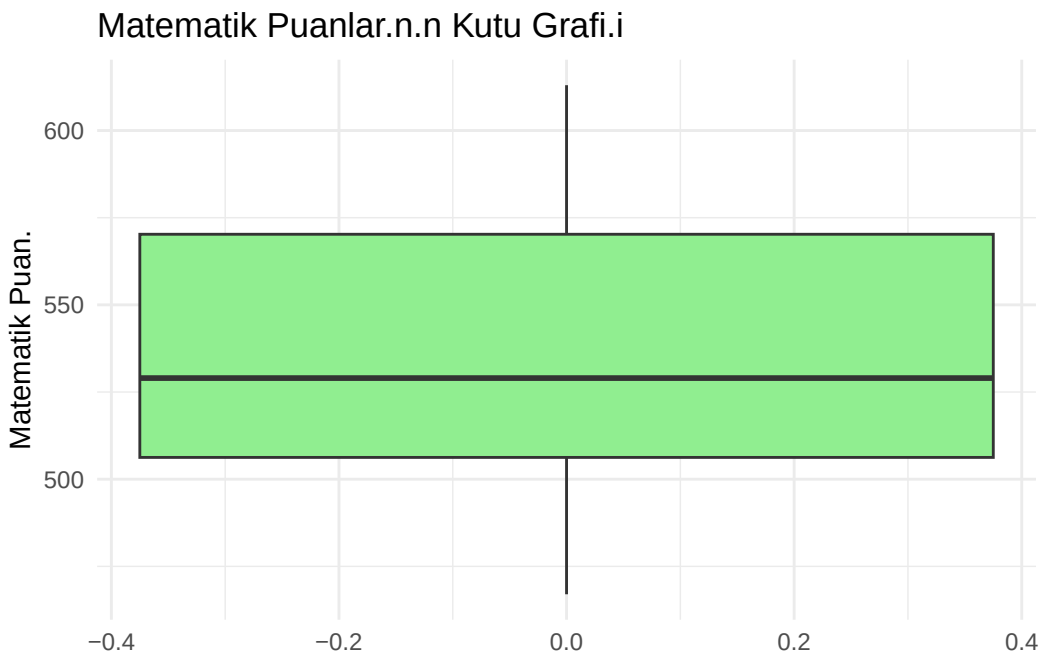
- Yoğunluk grafiği, matematik puanlarının dağılımını pürüzsüz bir eğri ile gösterir.
- **Eğilim**: Yoğunluk grafiği, verinin genel dağılımını ve yoğunluk noktalarını anlamak için idealdir. Grafik, hangi puan aralıklarının daha yoğun olduğunu gösterir.
- **Yayılma**: Eğrinin genişliği, verinin ne kadar yayıldığını gösterir. Dar bir eğri, verinin belirli bir alanda toplandığını, geniş bir eğri ise verinin daha yaygın olduğunu gösterir.
- **Yoğunluk grafiğinde `adjust`** argümanı, kernel düzleştiricide kullanılan bant genişliğini ayarlamak için kullanılır. Bu ayar, düzleştirme sürecini ince ayar yapmaya ve yoğunluk tahminindeki detay ve pürüzsüzlük seviyesini etkilemeye olanak tanır.

Kutu Grafiği

Kutu grafiği (box plot), bir veri setinin dağılımını görselleştirmek için kullanılan etkili bir araçtır. Genellikle bir değişkenin (örneğin, matematik puanları) merkezi eğilimini ve yayılmasını göstermektedir.

- **Bileşenler**:
  - **Kutu**: Verinin ilk çeyreği (Q1) ve üçüncü çeyreği (Q3) arasında yer alan dikdörtgen, veri dağılımının orta %50'sini temsil eder. Kutunun içindeki yatay çizgi, verinin medyanını gösterir.
  - **Kollar (Whiskers)**: Kutu grafiğinin üst ve alt kısmında yer alan kollar, Q1 ve Q3 dışındaki maksimum ve minimum değerleri gösterir. Bu kollar genellikle 1.5 IQR (Interquartile Range - Çeyrekler Arası Aralık) kadar uzanır.
  - **Aykırı Değerler**: Kutu grafiğinin dışındaki noktalar, aykırı değerleri temsil eder. Aykırı değerler, genel dağılımın dışında kalan veri noktalarıdır ve dikkatle incelenmesi gereken verilerdir.

```
# Kutu grafiği oluşturma
ggplot(SAT_2010, aes(y = math)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Matematik Puanlarının Kutu Grafiği",
       y = "Matematik Puanı") +
  theme_minimal()
```



Yorum:

- Kutu grafiği, matematik puanlarının merkezi eğilimlerini ve yayılmasını gösterir.
- **Medyan**: Kutu grafiğinin ortasında yer alan çizgi, medyanı temsil eder ve verinin merkezi değerini gösterir.
- **Çeyrekler**: Kutu, 1. çeyrek (Q1) ve 3. çeyrek (Q3) değerlerini gösterir. Q1, verinin %25'inin altında olduğu noktayı, Q3 ise %75'inin altında olduğu noktayı temsil eder.
- **Aykırı Değerler**: Kutu grafiği, veri setinde bulunan aykırı değerleri belirlemede yardımcı olur. Aykırı değerler, kutunun dışındaki noktalar olarak gösterilir. Aykırı değerlerin varlığı, verinin normal dağılım gösterip göstermediği hakkında ipucu verebilir.

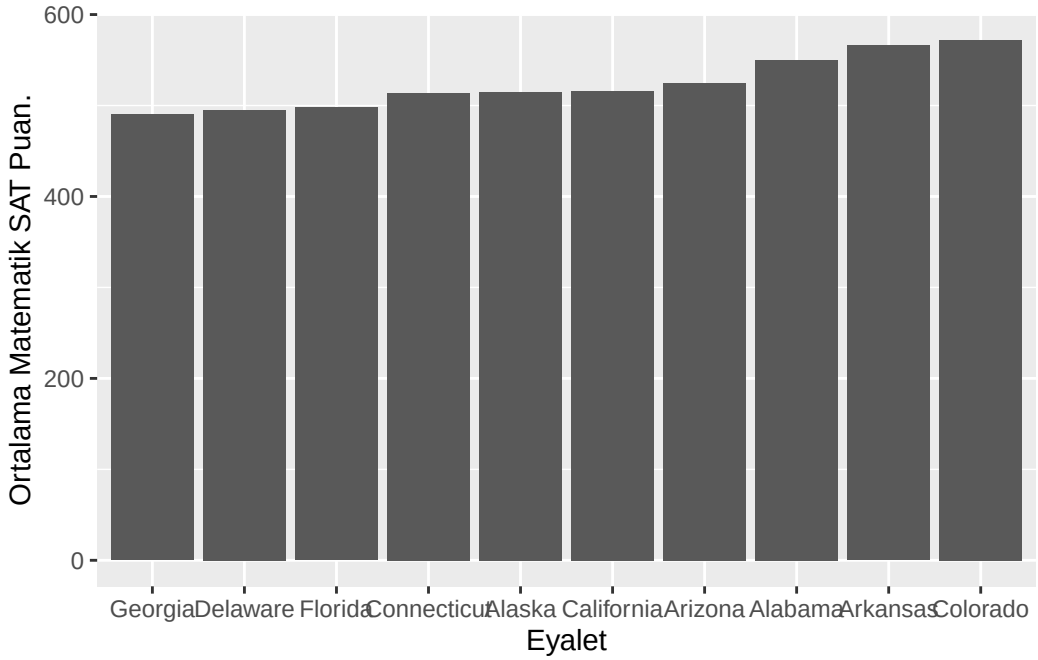
Kategorik Değişkenler

Eğer değişkeniniz kategorikse, değerlerin sürekli bir yoğunluğa sahip olduğu düşüncesi mantıklı değildir. Bunun yerine, bir kategorik değişkenin dağılımını göstermek için çubuk grafiği kullanabiliriz.

Çubuk Grafiği

Aşağıdaki kod, bir çubuk grafiği oluşturarak belirli eyaletlerin ortalama matematik SAT puanlarının dağılımını göstermektedir:

```r
# Çubuk grafiği oluşturma
ggplot(
  data = head(SAT_2010, 10),  # Sadece ilk 10 eyaleti göster
  aes(x = reorder(state, math), y = math)  # Eyalet isimlerini matematik puanlarına göre sıra
) +
  geom_col() +  # Çubukları çizin
  labs(x = "Eyalet", y = "Ortalama Matematik SAT Puanı")
```



Yorum:

- Çubuk grafiği, her eyaletin ortalama matematik puanını gösterir.
- **Karşılaştırma**: Farklı eyaletlerin ortalama puanları arasındaki karşılaştırmaları net bir şekilde sunar. Örneğin, belirli bir eyaletin puanı diğerlerinden belirgin şekilde daha yüksek veya düşükse, bu grafik bunu vurgular.
- **Yükseklik**: Çubukların yüksekliği, ortalama puanı temsil eder. Bu, izleyicilere hangi eyaletlerin daha iyi veya daha kötü performans gösterdiği hakkında bilgi verir.

> **Note**
>
> Daha önce belirtildiği gibi, kategorik bir değişkenin dağılımını göstermek için pasta grafiklerinin kullanımını önermiyoruz. Çoğu durumda, frekans tablosu daha bilgilendiricidir.

Çok Değişkenli Görselleştirmeler

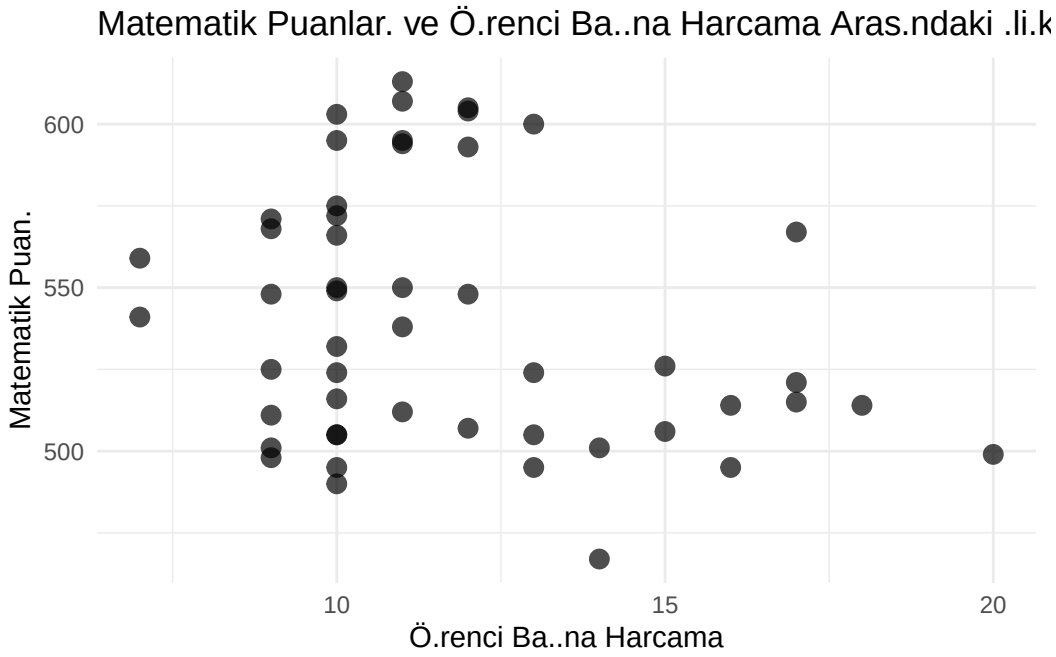
İki Sayısal Değişken

Çok değişkenli görselleştirmeler, birden fazla değişken arasındaki ilişkileri aynı anda gösterme konusunda oldukça etkilidir. Bu tür grafiklerden biri, iki nicel (sayısal) değişkenin gözlemlerini görselleştirmek için mükemmel bir yöntem olan **dağılım grafiği**dir. ggplot2 paketinde, dağılım grafiği `geom_point()` komutu ile oluşturulur. Dağılım grafiğinin ana amacı, birçok durum arasında iki değişken arasındaki ilişkiyi göstermektir. Genellikle, x ekseni bir değişkeni, y ekseni ise diğer bir değişkenin değerini temsil eden kartezyen koordinat sistemi kullanır.

Basit Dağılım Grafiği

İki sayısal değişken arasındaki ilişkiyi göstermek için matematik puanları ile öğrenci başına harcamalar arasında bir dağılım grafiği oluşturalım.

```
# Dağılım grafiği oluşturma
ggplot(SAT_2010, aes(x = expenditure, y = math)) +
  geom_point(size = 3, alpha = 0.7) +  # Noktaların boyutu ve saydamlık
  labs(title = "Matematik Puanları ve Öğrenci Başına Harcama Arasındaki İlişki",
       x = "Öğrenci Başına Harcama",
       y = "Matematik Puanı") +
  theme_minimal()
```



Yorum:

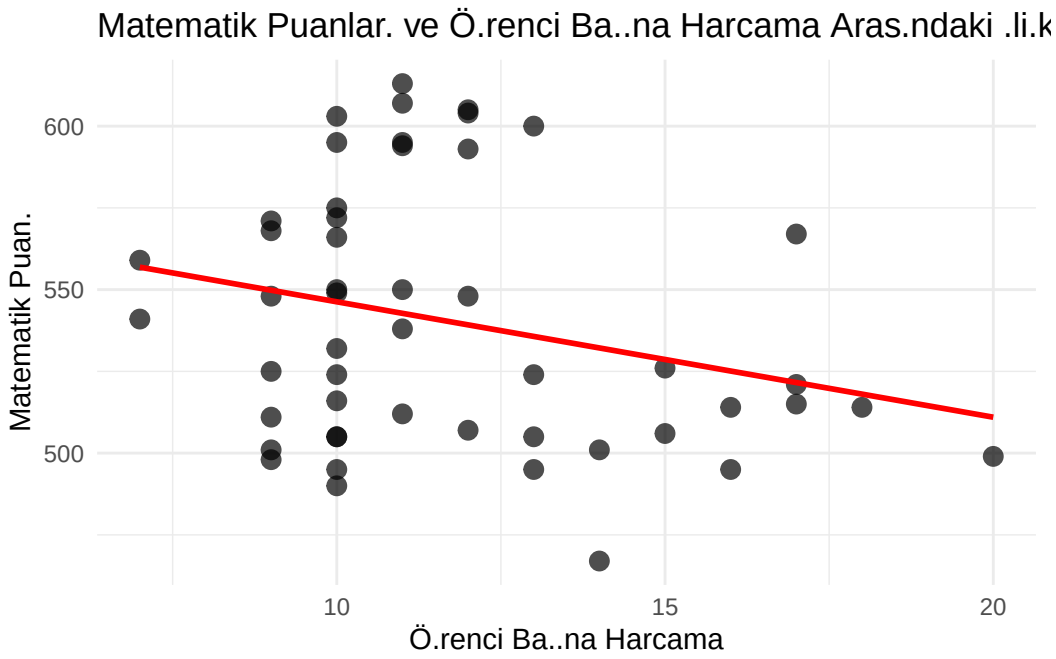
- **İlişki**: Bu dağılım grafiği, öğrenci başına harcama ile matematik puanları arasındaki ilişkiyi gösterir. Harcama arttıkça matematik puanlarının nasıl değiştiğini analiz edebilirsiniz.

Trend Çizgisi Ekleyin

Şimdi, grafik üzerine bir trend çizgisi ekleyelim ve eksen etiketlerini daha spesifik hale getirelim. Basit doğrusal regresyon çizgisini (method = "lm") noktaların üzerinden geçirecek şekilde `geom_smooth()` fonksiyonunu kullanacağız:

```
# Dağılım grafiği oluşturma ve trend çizgisi ekleme
ggplot(SAT_2010, aes(x = expenditure, y = math)) +
  geom_point(size = 3, alpha = 0.7) +  # Noktaların boyutu ve saydamlık
  geom_smooth(method = "lm", se = FALSE, color = "red") +  # Trend çizgisi ekleme
  labs(title = "Matematik Puanları ve Öğrenci Başına Harcama Arasındaki İlişki",
       x = "Öğrenci Başına Harcama",
       y = "Matematik Puanı") +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Yukarıdaki şekilde, ortalama matematik SAT puanı ile öğrenci başına harcama (binlerce ABD doları cinsinden) arasındaki ilişkiyi gösteriyoruz. Üçüncü (kategorik) bir değişken, fasetleme ve/veya katman ekleyerek grafik üzerinde gösterilebilir. Aşağıda, `SAT_2010` veri setini kullanarak, öğrenci başına harcama ile matematik SAT puanı arasındaki ilişkiyi gösteren bir dağılım grafiği oluşturacağız. Bu grafikte, öğrencilerin SAT'ye girme yüzdesine göre eyaletleri "yüksek", "orta" ve "düşük" gruplara ayıran bir `SAT_rate` değişkeni ekleyeceğiz. Sonrasında, bu grubu görselleştirmek için grafik üzerinde fasetleme kullanacağız.

Adımlar:

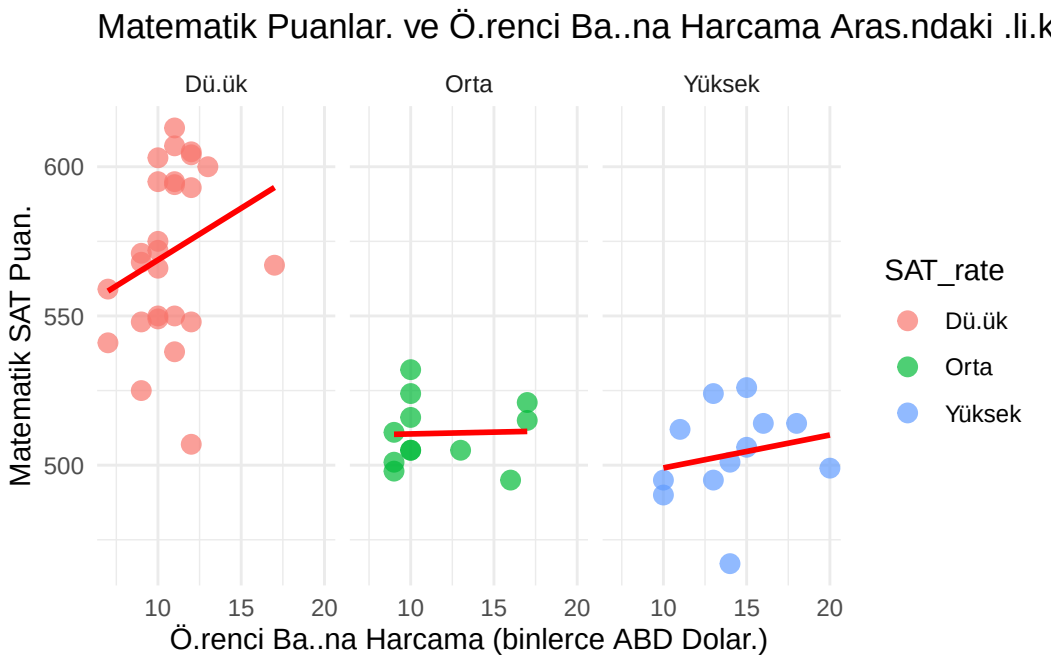
1. `mutate()` fonksiyonu ile `SAT_rate` değişkenini oluşturacağız.
2. Dağılım grafiğini `SAT_rate` değişkenine göre fasetleyeceğiz.

```
# SAT_rate değişkenini ekleme
SAT_2010_rate <- SAT_2010 %>%
  mutate(SAT_rate = case_when(
    sat_pct >= 70 ~ "Yüksek",
    sat_pct >= 40 & sat_pct < 70 ~ "Orta",
    TRUE ~ "Düşük"
  ))

# Dağılım grafiği oluşturma ve fasetleme
```

```r
ggplot(SAT_2010_rate, aes(x = expenditure, y = math)) +
  geom_point(aes(color = SAT_rate), size = 3, alpha = 0.7) +  # Noktaların boyutu ve saydaml
  geom_smooth(method = "lm", se = FALSE, color = "red") +  # Trend çizgisi ekleme
  labs(title = "Matematik Puanları ve Öğrenci Başına Harcama Arasındaki İlişki",
       x = "Öğrenci Başına Harcama (binlerce ABD Doları)",
       y = "Matematik SAT Puanı") +
  facet_wrap(~ SAT_rate) +  # SAT_rate değişkenine göre fasetleme
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Yorum:

- **İlişki**: Her bir fasette, öğrenci başına harcama ile matematik puanları arasındaki ilişkiyi ayrı ayrı görebiliriz.
- **Trend Çizgisi**: Kırmızı trend çizgisi, her grubun genel eğilimini temsil eder. Örneğin, yüksek SAT yüzdesine sahip grupta harcama arttıkça matematik puanlarının da arttığını görebiliriz.
- **Gruplar Arası Farklılıklar**: Farklı SAT gruplarının (yüksek, orta, düşük) arasındaki ilişki farklılıklarını gözlemlemek, eğitimin etkilerini anlamada önemli bir rol oynar.

Zaman Serisi

NHANES (National Health and Nutrition Examination Survey) veri tablosu, bireylerin tıbbi, davranışsal ve morfometrik ölçümlerini sağlar. Aşağıdaki grafikte, boy ve yaş arasındaki ilişkiyi gösteren bir dağılım grafiği bulunmaktadır. Her nokta bir kişiyi temsil eder ve o noktanın konumu, o kişinin iki değişkenin değerini belirtir. Dağılım grafikleri, iki değişken arasındaki basit ilişkileri görselleştirmek için kullanışlıdır. Örneğin, aşağıdaki grafikte doğumdan ergenliğe kadar olan boy büyüme desenini görebilirsiniz.

Verilerin daha iyi bir düzenlenmesi için (bunu daha sonra ele alacağız), çizgilerin mekânsal ilişkisinin (erişkin erkeklerin, erişkin kadınlardan daha uzun olma eğilimi) efsane etiketlerinin sıralaması ile eşleşmesini sağlamak için biraz daha düzenleme yapmak yararlıdır. Burada, faktör seviyelerini sıfırlamak için `fct_relevel()` fonksiyonunu kullanıyoruz (forcats paketinden).

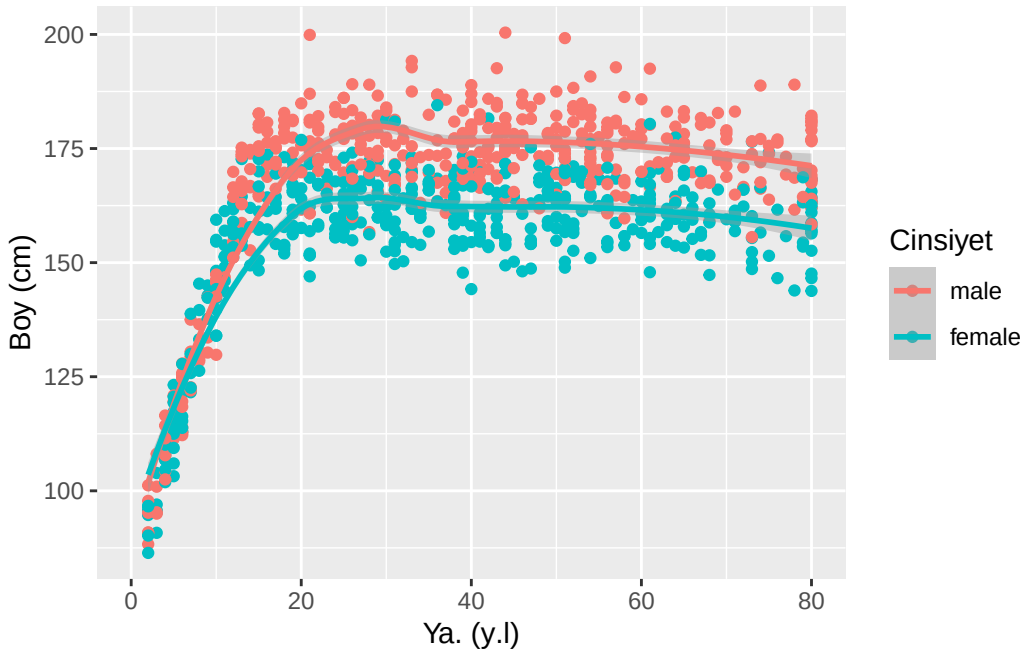
```
# install.packages("NHANES")
library(NHANES)

ggplot(
  data = slice_sample(NHANES, n = 1000),
  aes(x = Age, y = Height, color = fct_relevel(Gender, "male"))
) +
  geom_point() +
  geom_smooth() +
  xlab("Yaş (yıl)") +
  ylab("Boy (cm)") +
  labs(color = "Cinsiyet")
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
Warning: Removed 46 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 46 rows containing missing values or values outside the scale range
(`geom_point()`).
```



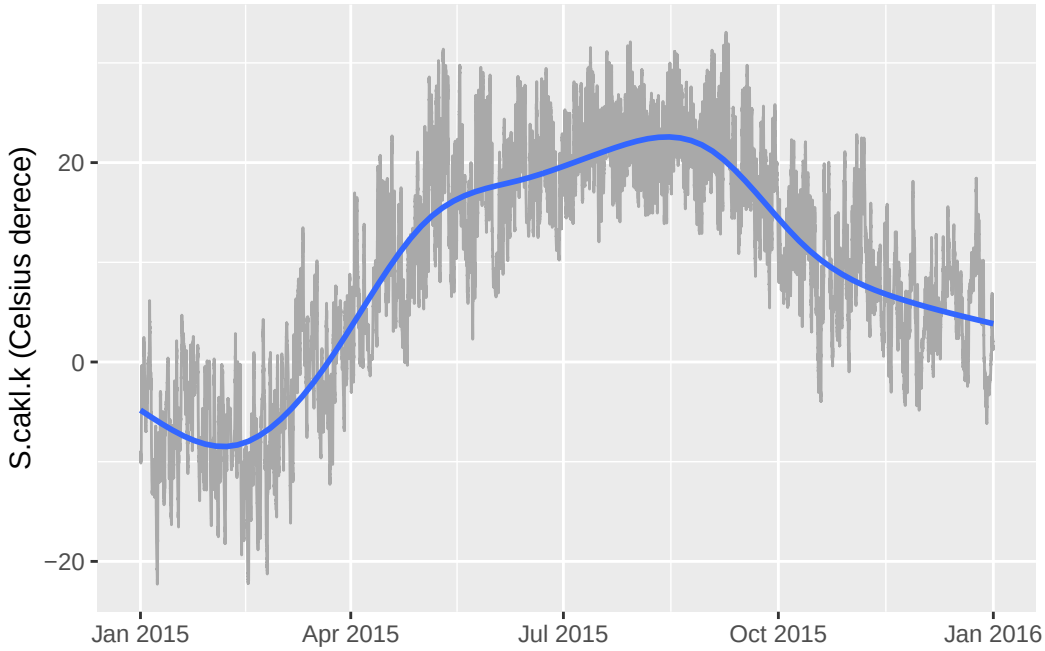
Bazı dağılım grafiklerinin özel anlamları vardır. Aşağıdaki şekilde gösterilen bir zaman serisi, yatay eksende zaman ve noktaların bağlı çizgilerle birleştirildiği bir dağılım grafiğidir; bu durum zaman içindeki sürekliliği gösterir. Aşağıdaki grafikte, Batı Massachusetts'teki bir hava istasyonundaki sıcaklık yıl boyunca gösterilmektedir. Mevsimlere dayanan tanıdık dalgalanmalar belirgindir. Bu tür grafikleri yorumlarken özellikle dikkatli olunmalıdır; zaman gerçekten iyi bir açıklayıcı değişken mi?

```
# install.packages("macleish")
library(macleish)
```

```
Loading required package: etl
```

```
ggplot(data = whately_2015, aes(x = when, y = temperature)) +
  geom_line(color = "darkgray") +
  geom_smooth() +
  xlab(NULL) +
  ylab("Sıcaklık (Celsius derece)")
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



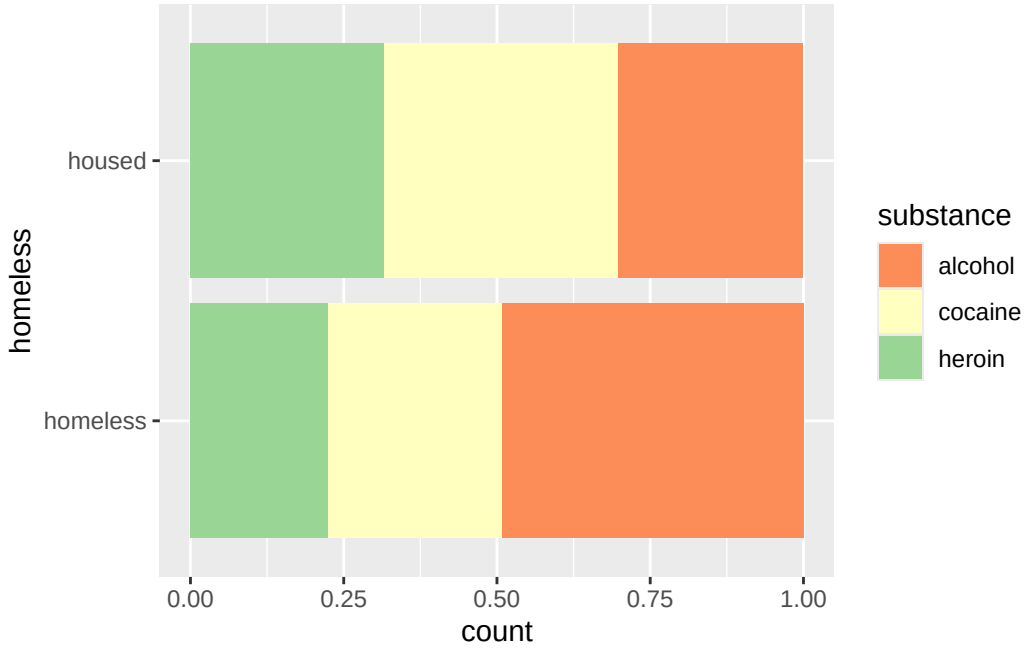
İki Kategorik Değişken

Birden fazla kategorik değişken kullanarak bilgilendirici bir grafiksel görüntü elde etmek için **yığılmış çubuk grafiği** kullanılabilir.

Yığılmış Çubuk Grafiği

Aşağıdaki kod, yığılmış çubuk grafiği oluşturarak "homeless" (sokakta yaşayan) ve "substance" (madde) değişkenlerini görselleştirmektedir:

```
# Yığılmış çubuk grafiği oluşturma
ggplot(data = mosaicData::HELPrct, aes(x = homeless)) +
  geom_bar(aes(fill = substance), position = "fill") +
  scale_fill_brewer(palette = "Spectral") +
  coord_flip()  # Çubukları yatay göstermek için
```



İki Kategorik Değişkenin İlişkisi

Hem açıklayıcı hem de yanıt değişkenleri kategorik (veya gruplandırılmış) olduğunda, noktalar ve çizgiler o kadar iyi çalışmaz. Örneğin, bir kişinin yaşına ve vücut kitle indeksine (BMI) dayanarak diyabet olasılığı nedir? Aşağıdaki şekilde gösterilen mozaik grafik (veya eikosogram), her hücredeki gözlem sayısının kutunun alanına orantılı olduğunu göstermektedir. Böylece, diyabetin yaşlı insanlar ve obez olanlar arasında daha yaygın olduğunu görebiliriz; çünkü mavi alanların büyüklüğü bağımsızlık modeline göre beklenenden daha fazladır, pembe alanlar ise daha azdır. Bu durum, Venn diagramlarından aşina olduğumuz olasılık kavramlarının daha doğru bir tasvirini sunar.

Mozaik Grafiği

Aşağıdaki kod, NHANES veri setinden yaş ve BMI değişkenlerine göre diyabeti gösteren bir mozaik grafiği oluşturmaktadır:

```
# install.packages("ggmosaic")
library(ggmosaic)


# Veriyi hazırlama
```

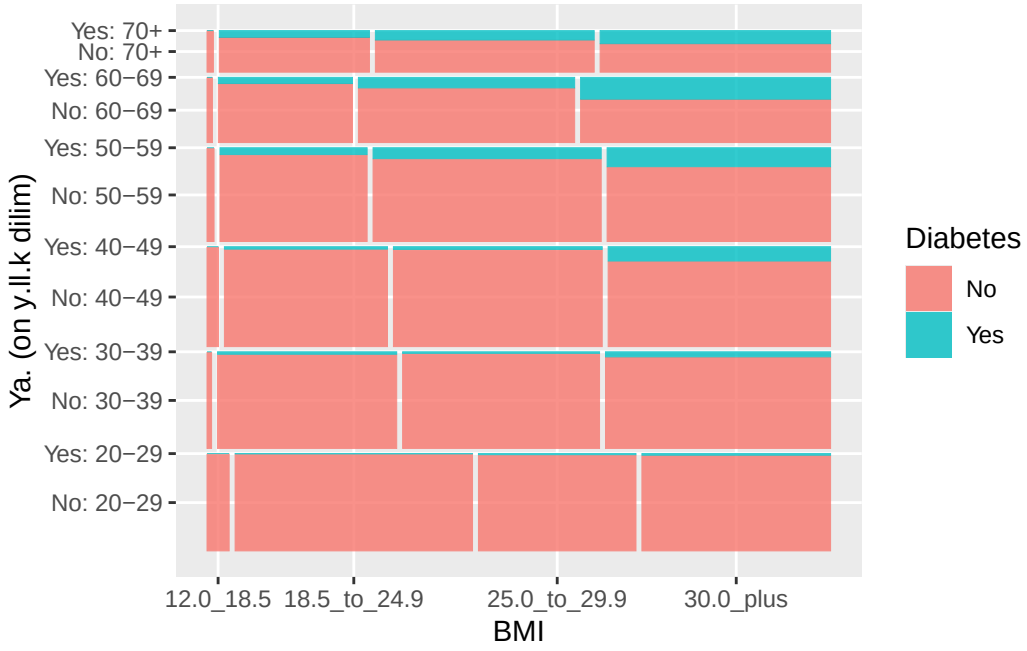
```r
mosaic_to_plot <- NHANES %>%
  filter(Age > 19) %>%
  mutate(AgeDecade = droplevels(AgeDecade)) %>%
  select(AgeDecade, Diabetes, BMI_WHO) %>%
  na.omit()

# Mozaik grafiği oluşturma
ggplot(mosaic_to_plot) +
  geom_mosaic(
    aes(x = product(BMI_WHO, AgeDecade), fill = Diabetes)
  ) +
  ylab("BMI") +
  xlab("Yaş (on yıllık dilim)") +
  coord_flip()
```

```
Warning: The `scale_name` argument of `continuous_scale()` is deprecated as of ggplot2
3.5.0.

Warning: The `trans` argument of `continuous_scale()` is deprecated as of ggplot2 3.5.0.
i Please use the `transform` argument instead.

Warning: `unite_()` was deprecated in tidyr 1.2.0.
i Please use `unite()` instead.
i The deprecated feature was likely used in the ggmosaic package.
  Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
```



Not: `geom_mosaic()` fonksiyonu ggplot2'nin bir parçası değildir; bunun yerine ggmosaic paketinde bulunmaktadır.

Bir Sayısal ve Bir Kategorik Değişken

Bir sayısal yanıt değişkenini bir kategorik açıklayıcı değişkenle karşılaştırmak için yaygın bir seçenek, **kutu ve çubuk grafiği (box-and-whisker plot)** oluşturmaktır. Bu grafik, beş sayı özetinin (minimum [0. persentil], Q1 [25. persentil], medyan [50. persentil], Q3 [75. persentil] ve maksimum [100. persentil]) grafiksel bir gösterimi olarak düşünülmesi en kolay olanıdır.

Kutu Grafiği

Aşağıdaki kod, her ay için sıcaklık verilerini kullanarak bir kutu grafiği oluşturmaktadır:

```
# Gerekli kütüphaneleri yükleyin
library(ggplot2)
library(dplyr)
library(lubridate)

# Veri setini hazırlayın
whately_2015 %>%
  mutate(month = as.factor(lubridate::month(when, label = TRUE))) %>%
```

```
  group_by(month) %>%
  skim(temperature) %>%
  select(-na)
```

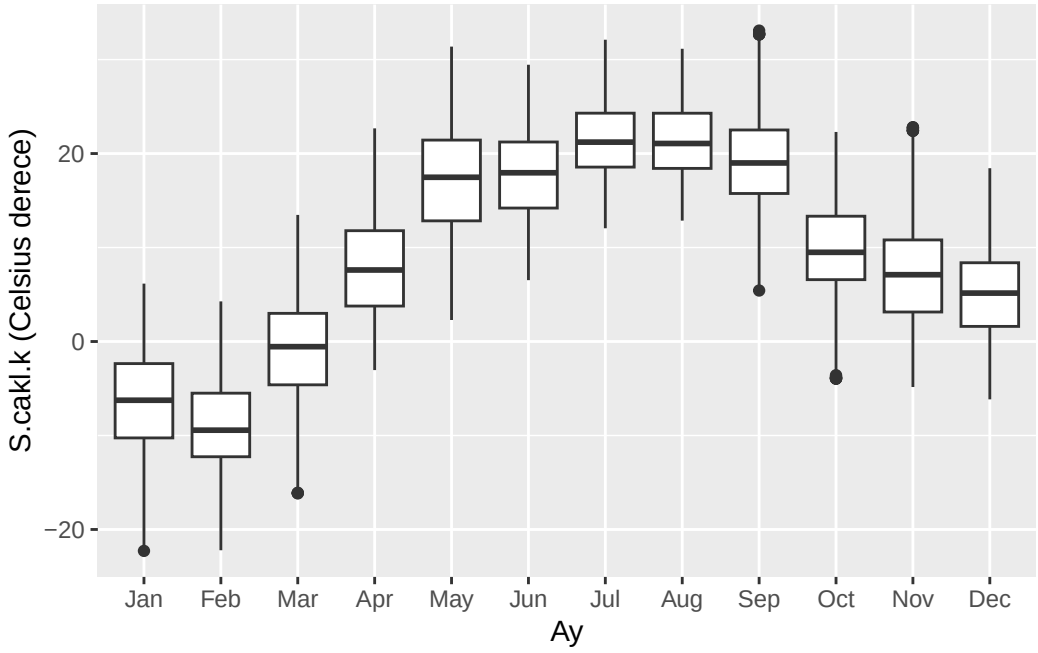
Variable type: numeric

var	month	n	mean	sd	p0	p25	p50	p75	p100
temperature	Jan	4464	-6.37	5.14	-22.28	-10.26	-6.25	-2.35	6.16
temperature	Feb	4032	-9.26	5.11	-22.21	-12.26	-9.43	-5.50	4.27
temperature	Mar	4464	-0.87	5.06	-16.16	-4.61	-0.55	2.99	13.47
temperature	Apr	4320	8.04	5.51	-3.04	3.77	7.61	11.79	22.68
temperature	May	4464	17.36	5.94	2.29	12.84	17.48	21.43	31.38
temperature	Jun	4320	17.75	5.11	6.53	14.20	17.95	21.23	29.45
temperature	Jul	4464	21.56	3.90	12.05	18.56	21.22	24.30	32.11
temperature	Aug	4464	21.45	3.79	12.86	18.42	21.07	24.29	31.15
temperature	Sep	4320	19.28	5.07	5.43	15.75	19.00	22.51	33.08
temperature	Oct	4464	9.79	5.00	-3.97	6.58	9.49	13.33	22.30
temperature	Nov	4320	7.28	5.65	-4.84	3.14	7.11	10.81	22.81
temperature	Dec	4464	4.95	4.59	-6.16	1.61	5.15	8.38	18.44

Yukarıdaki kod, `whately_2015` veri setinden her ay için sıcaklıkları gruplandırır ve sıcaklık verisinin özetini sağlar.

Şimdi, kutu grafiğini oluşturalım:

```
# Kutu grafiği oluşturma
ggplot(
  data = whately_2015,
  aes(
    x = lubridate::month(when, label = TRUE),
    y = temperature
  )
) +
  geom_boxplot() +
  xlab("Ay") +
  ylab("Sıcaklık (Celsius derece)")
```



Kapsamlı Grafiklerin Tablosu

Aşağıda, çeşitli sayısal ve kategorik değişkenler için önerilen grafik türlerini ve ggplot2 ile kullanılacak fonksiyonları daha kapsamlı bir şekilde özetleyen bir tablo bulunmaktadır. Tablo, grafik türlerini, fonksiyonları ve ek bilgileri içermektedir.

Tablo 4: Temel Grafik Türleri Özet

Yanıt (y)	Açıklayıcı (x)	Grafik Türü	`geom_*()` Fonksiyonu	Ek Bilgiler
Kategorik	Kategorik	Çubuk Grafiği	`geom_bar()`	Frekans veya toplam değer gösterir.
Kategorik	Kategorik	Yığılmış Çubuk Grafiği	`geom_bar(position = "fill")`	Oranları gösterir.
Sayısal	Kategorik	Kutu Grafiği	`geom_boxplot()`	Medyan, çeyrekler ve aykırı değerleri gösterir.
Sayısal	Sayısal	Dağılım Grafiği	`geom_point()`	Değişkenler arasındaki ilişkiyi gösterir.

Yanıt (y)	Açıklayıcı (x)	Grafik Türü	`geom_*()` Fonksiyonu	Ek Bilgiler
Sayısal	Sayısal	Histogram	`geom_histogram()`	Dağılımı gösterir; `binwidth` ile ayarlanabilir.
Sayısal	Sayısal	Yoğunluk Grafiği	`geom_density()`	Dağılımın yoğunluğunu gösterir; `adjust` ile ayarlanabilir.
Sayısal	Sayısal	Zaman Serisi Grafiği	`geom_line()`	Zamanla değişimi gösterir.
Sayısal	Kategorik	Çizgi Grafiği	`geom_line()`	Kategorik değişkenler için sıralı grafikler.
Sayısal	Sayısal	Bubbles Grafiği	`geom_point(aes(size = <size_var>))`	Ekstra boyut bilgisi eklemek için kullanılır.

Ek Bilgiler ve Açıklamalar:

- **Çubuk Grafiği (`geom_bar()`)**: Kategorik verilerin frekansını veya toplam değerini gösterir. Verilerin toplamını görsel olarak ifade ederken etiketler eklemek faydalı olabilir.
- **Yığılmış Çubuk Grafiği (`geom_bar(position = "fill")`)**: Kategorilerin oranlarını gösterir. Yüksekliği toplamı ifade ederken, her rengin oranını da gösterir.
- **Kutu Grafiği (`geom_boxplot()`)**: Verinin dağılımını ve aykırı değerleri görselleştirir. Medyan, 1. ve 3. çeyrek gibi özet istatistikleri sunar. Aykırı değerler kutunun dışında gösterilir.
- **Dağılım Grafiği (`geom_point()`)**: İki sayısal değişken arasındaki ilişkiyi gösterir. Her bir nokta, bir veri noktasını temsil eder ve renk veya şekil estetikleri ile kategorik değişkenler de gösterilebilir.
- **Histogram (`geom_histogram()`)**: Sayısal verilerin dağılımını göstermek için kullanılır. `binwidth` parametresi, histogramın çubuk genişliğini belirler. En uygun bin genişliği verinin doğasına göre ayarlanmalıdır.
- **Yoğunluk Grafiği (`geom_density()`)**: Verinin yoğunluğunu göstermek için kullanılır. Smoothing (düzleştirme) işlemi ile histogramdan daha pürüzsüz bir görünüm sağlar. `adjust` parametresi ile yoğunluk düzleştirmesinin ne kadar pürüzsüz olacağını ayarlamak mümkündür.

- **Zaman Serisi Grafiği (`geom_line()`)**: Zaman içinde değişimi göstermek için kullanılır. Genellikle çizgi şeklinde gösterilir ve veri noktaları arasında bağlantı sağlar.
- **Bubbles Grafiği (`geom_point(aes(size = <size_var>))`)**: Çok değişkenli analizlerde, boyut bilgisi eklenerek verilerin daha etkili görselleştirilmesine olanak tanır.