1. Hafta: Giriş

Hakan Mehmetcik

1. R ve R'ın Temel Yapısı

R Dilinin Genel Özellikleri ve Kullanımı

R, veri analizi, istatistiksel hesaplamalar ve grafiksel görselleştirme için tasarlanmış güçlü ve açık kaynaklı bir programlama dilidir. R'ın temel avantajlarından biri, akademik ve profesyonel dünyada geniş bir kullanım alanına sahip olmasıdır. R, özellikle büyük veri setleri üzerinde hızlı ve etkili analizler yapmayı sağlar.

R ile neler yapabilirsiniz?

- Sayısal veriler üzerinde işlem yapabilir, istatistiksel analizler gerçekleştirebilirsiniz.
- Verilerinizi görselleştirmek için grafikler ve çizelgeler oluşturabilirsiniz.
- Gelişmiş istatistiksel modeller kurabilir, veriler üzerinde tahminlerde bulunabilirsiniz.

R'ın Veri Analizi ve İstatistikteki Yeri

R, özellikle veri bilimciler, istatistikçiler ve araştırmacılar tarafından veri analizi için yaygın olarak kullanılır. Akademik dünyada, bilimsel araştırmalarda ve iş dünyasında veri analizi için standart bir araçtır. Bunun nedeni R'ın geniş bir paket ekosistemine sahip olmasıdır; yani, özel amaçlar için kullanabileceğiniz binlerce hazır fonksiyon ve paket bulunur.

R'ın Avantajları:

- Esneklik: R, çok farklı veri türleri ve analizlerle çalışmanıza olanak sağlar.
- Geniş Paket Desteği: R, geniş bir topluluğa sahiptir ve hemen hemen her türlü veri analizi için uygun bir paket bulabilirsiniz (örneğin: `ggplot2`, `dplyr`, `tidyverse`). - Veri Görselleştirme: Grafiksel görselleştirme için R, güçlü araçlar sunar. Karmaşık veri setlerini kolayca anlaşılır grafiklere dönüştürebilirsiniz.

R'ı GitHub Codespaces Üzerinde Kullanmaya Başlama

Bu derste, GitHub Codespaces kullanarak R ile çalışacağız. Codespaces, bulut tabanlı bir geliştirme ortamıdır. Yani, bilgisayarınıza herhangi bir yazılım yüklemeden GitHub üzerinde R projeleri geliştirebilirsiniz. GitHub Codespaces, R dili için hazır bir çalışma ortamı sağlar ve GitHub repository'nizle entegre bir şekilde çalışmanıza olanak tanır.

R ile GitHub Codespaces'te Çalışmaya Başlama Adımları:

1. Repository'ye Erişin: GitHub'da ders için oluşturulan repository'ye gidin (örneğin: IST2083 GitHub Repository).
2. Codespace'i Açın: Repository sayfasında "Code" butonuna tıklayın ve "Open with Codespaces" seçeneğini seçin.
3. Ortamı Başlatın: Codespace, sizin için bir geliştirme ortamı oluşturacak. Bu ortamda R kodları yazabilir ve çalıştırabilirsiniz.
4. Dosyalarla Çalışma: R dilinde yazılmış `.qmd` (Quarto) dosyalarını açarak ders notlarını inceleyebilir ve R konsolu üzerinde örnek kodları çalıştırabilirsiniz.

GitHub Codespaces'in Avantajları:

- Kurulum Gerektirmez: Hiçbir yazılım indirip kurmanıza gerek kalmaz, her şey tarayıcı üzerinde çalışır.
- Bulut Tabanlı: Tüm verileriniz ve projeleriniz bulutta saklanır, böylece her yerden erişim sağlayabilirsiniz.
- Anında Kullanıma Hazır: R dili ve ilgili paketler önceden yüklenmiştir; hemen kullanmaya başlayabilirsiniz.

Sonuç Olarak: R, veri analizi ve istatistik için en güçlü araçlardan biridir. GitHub Codespaces ise bu dili kullanmayı kolaylaştıran bir platform sunar. Bu derste, GitHub Codespaces kullanarak R ile nasıl çalışılacağını ve temel veri analizi kavramlarını öğrenmeye başlayacaksınız.

---

2. R Konsolu ve Temel İşlemler

R, bir programlama dili olmasının yanı sıra, aynı zamanda bir hesap makinesi gibi basit matematiksel işlemler yapabileceğiniz bir ortam sağlar. R konsolunu kullanarak basit komutları hemen çalıştırabilirsiniz. Aşağıda R konsolunda nasıl çalışacağınızı ve temel matematiksel işlemleri nasıl gerçekleştireceğinizi öğreneceksiniz.

R Konsolunu Kullanarak Basit Komutlar Çalıştırma

GitHub Codespaces'te R konsolunu açarak komutlar yazabilirsiniz. R konsolu, yazdığınız her komutu anında çalıştırır ve size sonucu gösterir. Örneğin, R konsoluna aşağıdaki gibi bir toplama işlemi yazabilirsiniz:

```
3+5
```

[1] 8

Bu komutu çalıştırdığınızda, R size işlemin sonucunu gösterecektir:

```
[1] 8
```

Temel Matematiksel İşlemler

R, toplama, çıkarma, çarpma ve bölme gibi temel matematiksel işlemleri doğrudan konsol üzerinde yapabilir.

> **Note**
>
> Bu tür işlemler aynı zamanda temel matematiksel işlemleri gerçekleştirmek için R'da sunulan çeşitli **aritmetik operatörler** olarak adlandırılırlar.

Aşağıda bazı temel matematiksel işlemlerin örnekleri verilmiştir:

```
# Toplama
toplam <- 5 + 3
print(toplam)  # 8
```

[1] 8

```
# Çıkarma
cikarma <- 5 - 3
print(cikarma)  # 2
```

[1] 2

```
# Çarpma
carpma <- 5 * 3
print(carpma)  # 15
```

[1] 15

```
# Bölme
bolme <- 5 / 3
print(bolme)  # 1.666667
```

[1] 1.666667

```
# Üs Alma
us_alma <- 2^3
print(us_alma)  # 8
```

[1] 8

```
# Modülüs
modulus <- 5 %% 3
print(modulus)  # 2
```

[1] 2

```
# Tam Bölme
tam_bolme <- 5 %/% 3
print(tam_bolme)  # 1
```

[1] 1

Karşılaştırma Operatörleri

Karşılaştırma operatörleri, iki değeri karşılaştırarak TRUE ya da FALSE değerleri döndürür. R dilinde yaygın olarak kullanılan karşılaştırma operatörleri şunlardır:

```
# Eşittir
print(5 == 3)  # FALSE
```

[1] FALSE

```
print(5 == 5)  # TRUE
```

[1] TRUE

```
# Eşit Değildir
print(5 != 3)  # TRUE
```

[1] TRUE

```
print(5 != 5)  # FALSE
```

[1] FALSE

```
# Büyüktür
print(5 > 3)  # TRUE
```

[1] TRUE

```
print(3 > 5)  # FALSE
```

[1] FALSE

```
# Küçüktür
print(5 < 3)  # FALSE
```

[1] FALSE

```
print(3 < 5)  # TRUE
```

[1] TRUE

```
# Büyük Eşittir
print(5 >= 3)  # TRUE
```

[1] TRUE

```
print(5 >= 5)  # TRUE
```

[1] TRUE

```
print(3 >= 5)  # FALSE
```

[1] FALSE

```r
# Küçük Eşittir
print(5 <= 3)  # FALSE
```

[1] FALSE

```r
print(5 <= 5)  # TRUE
```

[1] TRUE

```r
print(3 <= 5)  # TRUE
```

[1] TRUE

Mantıksal Operatörler

Mantıksal operatörler, R dilinde mantıksal ifadeleri değerlendirmek ve birleştirmek için kullanılır. Bu operatörler, genellikle karşılaştırma operatörleri ile birlikte kullanılarak daha karmaşık koşullar oluşturur. İşte R dilinde yaygın olarak kullanılan mantıksal operatörler: **1. Ve (& ve &&)**

İki mantıksal ifadenin her ikisi de doğruysa TRUE döner. & operatörü vektörler üzerinde eleman bazında çalışır. && operatörü yalnızca ilk elemanları karşılaştırır.

```r
TRUE & TRUE   # TRUE
```

[1] TRUE

```r
TRUE & FALSE  # FALSE
```

[1] FALSE

```r
TRUE && TRUE  # TRUE
```

[1] TRUE

```r
TRUE && FALSE # FALSE
```

[1] FALSE

2. Veya (| ve ||)

İki mantıksal ifadeden en az biri doğruysa TRUE döner. | operatörü vektörler üzerinde eleman bazında çalışır. || operatörü yalnızca ilk elemanları karşılaştırır.

```
TRUE | TRUE   # TRUE
```

[1] TRUE

```
TRUE | FALSE  # TRUE
```

[1] TRUE

```
FALSE | FALSE # FALSE
```

[1] FALSE

```
TRUE || TRUE  # TRUE
```

[1] TRUE

```
TRUE || FALSE # TRUE
```

[1] TRUE

```
FALSE || FALSE # FALSE
```

[1] FALSE

3. Değil (!)

Bir mantıksal ifadenin tersini döner.

```
!TRUE  # FALSE
```

[1] FALSE

```
!FALSE # TRUE
```

[1] TRUE

Diğer Önemli Operatörler

R dilinde çeşitli operatörler, veri manipülasyonu ve analizinde sıkça kullanılır. İşte R dilinde yaygın olarak kullanılan bazı önemli operatörler:

Kolon Operatörü (:)

Kolon operatörü, belirli bir aralıktaki sayıları oluşturmak için kullanılır.

```
# 1'den 10'a kadar olan sayıları oluşturma
sequence <- 1:10
print(sequence)  # 1 2 3 4 5 6 7 8 9 10
```

[1] 1 2 3 4 5 6 7 8 9 10

Atama Operatörü (<- ve =)

Atama operatörleri, bir değeri bir değişkene atamak için kullanılır.

```
# Atama operatörleri
x <- 10
y = 20
print(x)  # 10
```

[1] 10

```
print(y)  # 20
```

[1] 20

Üye Olma Operatörü (%in%)

Üye olma operatörü, bir elemanın bir vektörde olup olmadığını kontrol eder.

```
# Üye olma operatörü
vec <- c(1, 2, 3, 4, 5)
print(3 %in% vec)  # TRUE
```

[1] TRUE

```r
print(6 %in% vec)  # FALSE
```

[1] FALSE

Dizi Operatörü (`seq()`)

Dizi operatörü, belirli bir aralıkta ve adım büyüklüğünde sayılar oluşturmak için kullanılır.

```r
# Dizi operatörü
sequence <- seq(1, 10, by = 2)
print(sequence)  # 1 3 5 7 9
```

[1] 1 3 5 7 9

Tekrar Operatörü (`rep()`)

Tekrar operatörü, belirli bir değeri veya vektörü tekrar etmek için kullanılır.

```r
# Tekrar operatörü
repeated <- rep(1:3, times = 3)
print(repeated)  # 1 2 3 1 2 3 1 2 3
```

[1] 1 2 3 1 2 3 1 2 3

---

3. R'da Mesajlarını Anlama ve Çözme ve Yardım Alma

R ile çalışırken bazen hatalarla karşılaşabilirsiniz. Hatalar, kodunuzda bir sorun olduğunu ve R'ın bunu çalıştıramadığını belirtir. Ancak endişelenmenize gerek yok, bu hatalar çoğunlukla küçük yazım veya kullanım hatalarından kaynaklanır.

**Örnek Bir Hata:

Aşağıdaki komutta `*` yerine `x` yazıldığı için bir hata oluşacaktır:

```r
3 x 5
```

Çıktı:

```
Error: unexpected symbol in "3 x"
```

Bu hata, R'ın `x` sembolünü tanımadığını ve bu yüzden işlemi gerçekleştiremediğini belirtir. Bu hatayı düzeltmek için `x` sembolünü `*` ile değiştirebilirsiniz:

```
3 * 5
```

Bu sefer doğru sonuç olan `15` çıktısını elde edeceksiniz.

Genel Hata Çözme İpuçları:

1. Sembolleri Doğru Kullanın: R'da toplama için `+`, çarpma için `*`, bölme için `/` gibi semboller kullanılır. Yanlış bir sembol kullanırsanız hata alırsınız.
2. Kodunuzda Yazım Hataları Olup Olmadığını Kontrol Edin: R büyük/küçük harfe duyarlıdır, yani `A` ve `a` R için farklı şeyler ifade eder. Yazım hatalarına dikkat edin.
3. Hata Mesajını Dikkatlice Okuyun: R'ın verdiği hata mesajları, sorunun ne olduğunu anlamanıza yardımcı olabilir. Örneğin, yukarıdaki hata mesajı, "unexpected symbol" ifadesiyle bir sembol hatası olduğunu belirtmiştir.
