

R Programlama Diline Giriş

Dinçer GÖKSÜLÜK, Phd.

13 Mayıs 2022

Contents

Eğitime Başlarken...	1
Neden R? R yazılımını farklı kılan nedir?	1
Giriş	2
Yardım	3
Yorumlar, boşluklar, özel karakterler	3
R Nesneleri	5
Nesne yapıları/sınıfları (Structure, class)	5
Vektörler	5
Matrisler	8
Veri Tabloları	10
Listeler	12
Nesnelerde sınıflar arası dönüşümler	12

Eğitime Başlarken...

R programlama dili ile veri analizi ve yapay zeka uygulamaları eğitimine başlamadan önce aşağıdaki gereksinimlerin karşılandığından emin olunuz...

- **R 4.2.0** sürümü kurulmuş olmalıdır.
- **RStudio Desktop 2022.02.2+485** sürümü kurulmuş olmalıdır.
- Windows veya Mac OS işletim sistemlerinden birisi önerilmektedir.
- Kurs için gerekli olan R kütüphaneleri kurulmuş olmalıdır.

R programı, açık kaynak kodlu, ücretsiz ve güçlü analiz becerilerine sahip olması gibi birtakım özelliklerinden dolayı araştırmacılar tarafından büyük ilgi görmüş, yalnızca istatistiksel hesaplamalar yapan bir program olmaktan çıkıp, hemen hemen her alanda kullanımı artan, çok güçlü bir programlama dili haline gelmiştir.

Neden R? R yazılımını farklı kılan nedir?

R ücretsizdir. Açık kaynak kodlu bir proje olarak, R'yi ücret ödemediğinizde kullanabilirsiniz. Yıllık ücret, lisans ve kullanıcı kısıtlaması gibi durumları düşünmek zorunda kalmazsınız. Fakat daha da önemlisi R açık kaynak kodlu bir yazılımdır. R içerisinde bulunan kodlara ve fonksiyonlara erişebilir, inceleyebilir ve istediğiniz gibi düzenleyebilirsiniz.

R bir programlama dilidir. R içerisinde analizler fonksiyon ve komutlar yardımıyla yapılır ve diğer programlama dillerine göre öğrenmesi kolay bir dildir. Ayrıca R programı, C ve Java gibi birçok programlama dili ile uyumlu çalışabilmektedir.

R'nin temel tasarım amaçlarından biri grafiklerdir. R, içerisinde grafikler oluşturan etkin araçlara sahiptir. Ayrıca isterseniz kendi grafiklerinizi kendiniz oluşturabilirsiniz.

R, esnek bir istatistiksel analiz aracıdır. Tüm standart veri analiz araçları R'nin içerisinde mevcuttur. Farklı formatlardaki verilerinizi R'ye aktarabilir, veri üzerinde istediğiniz değişiklikleri yapabilir ve uygun istatistiksel yöntemleri (regresyon, ANOVA, GLM, vs.) kullanabilirsiniz.

R, güçlü ve dinamik bir topluluğa sahiptir. R ile ilgili herhangi bir sorunla karşılaştığınızda uluslararası düzeyde R'ye katkıda bulunan binlerce kullanıcıya danışabilirsiniz. Şu anda, R kullanıcıları sayesinde, içerisinde çok farklı analizlerin yapılabildiği CRAN ağına 13462, BIOCONDUCTOR ağına ise 1649 paket mevcuttur ve bu sayı her geçen gün artmaktadır. İnternet üzerinden R ile ilgili bilgi ve dokümanlara kolaylıkla erişebilirsiniz.

R, sınırsız olanaklara sahiptir. R ile, daha önceden belirlenmiş kalıpları seçmek ve kullanmakla kısıtlanmazsınız. R'ye katkıda bulunan kişiler tarafından oluşturulmuş hazır kodları kullanabilir ya da kendi kodlarınızı kendiniz oluşturabilirsiniz. Bu sayede, hem kendinize hem de dünya genelinde R programını kullanan milyonlarca kullanıcıya katkıda bulunabilirsiniz.

Giriş

R yazılımında kodlar > imleci yanına girilerek ve ENTER tuşuna basılarak çalıştırılır. Bununla birlikte R Studio yazılımında yeni bir *script* oluşturularak, kodlar scripte girilerek ve Run simgesi tıklanarak (veya Ctrl + ENTER (Windows / Linux), Cmd + ENTER (Mac OS)) da çalıştırılabilir.

R'de en basit olarak hesap makinesindeki benzer aritmetik işlemler yapılabilir. En temel terimler toplama için +, çıkarma için -, çarpma için *, bölme için / simgeleridir. Bununla birlikte üs almak için ^ simgesi, kök almak için ise `sqrt(...)` fonksiyonu kullanılmaktadır:

```
17 + 32
```

```
## [1] 49
```

```
42 * 3 + 5/0.2
```

```
## [1] 151
```

```
11^2 + sqrt(49)
```

```
## [1] 128
```

Benzer olarak `exp(...)`, `log(...)`, `sin(...)`, `cos(...)`, `abs(...)`, vb. fonksiyonlar yardımı ile temel matematiksel işlemler yapılabilmektedir:

```
exp(3.2) # Üssel fonksiyon
```

```
## [1] 24.53253
```

```
log(100) # Doğal logaritma ln(x)
```

```
## [1] 4.60517
```

```
log(100, base=10) # Logaritma. taban değeri 'base' ile veriliyor.
```

```
## [1] 2
```

```
abs(-23) # Mutlak değer
```

```
## [1] 23
```

```
sin(pi / 2) # Trigonometri. Sinüs.
```

```
## [1] 1
```

round(...) fonksiyonu ile değerlerin ondalık yuvarlamaları yapılabilir:

```
round(log(100), digits = 2)
```

```
## [1] 4.61
```

R'da sıklıkla kullanılan bazı matematik fonksiyonları aşağıda verilmiştir:

Fonksiyon	İşlev
<code>sqrt(...)</code>	Karekök alma. <code>sqrt(2)</code> ile $2^{(0.5)}$ aynı amaç için kullanılır. Küpkök veya özel bir kök değerini almak için $2^{(3/5)}$, $9^{(1/3)}$ gibi işlemler yapılabilir.
<code>log(...)</code>	Logaritma alma.
<code>exp(...)</code>	Üssel fonksiyon.
<code>abs(...)</code>	Mutlak değer.
<code>sin(...), cos(...), ...</code>	Trigonometrik fonksiyonlar
<code>round(...), ceiling(...), floor(...)</code>	Değer yuvarlama işlemleri.
<code>sign(...)</code>	İşaret fonksiyonu. (Pozitif / Negatif)
...	...

Yardım

R diğer dillere göre daha kolay bir programlama dilidir. Bununla birlikte R ile çalışırken fonksiyonların kullanımı ile ilgili yardım almak için, ilgili fonksiyonun başına `?` eklenir:

```
?mean
```

R yazılımının **Help** menüsünde ilgili paket içerisinde yer alan fonksiyon(lar) için açıklayıcı bilgilere ve örneklerle ulaşılabilir. R resmi internet sayfası üzerinde de (<https://cran.r-project.org/manuals.html>) başta **An Introduction to R** olmak üzere çeşitli konularda yardımcı kaynaklar bulunmaktadır.

Eğitici kaynaklar dışında R kullanıcılarının iletişimini arttırmak, bilgi birikimlerini paylaşmak ve R ile çalışırken karşılaşılan problemlere destek aramak/sunmak amaçlı çeşitli forumlar ve mail listeleri oluşturulmuştur:

R mailing list: <https://www.r-project.org/mail.html>

Turkish Community of R: <http://www.meetup.com/tcr-users/>

R ile ilgili kitaplar <https://www.r-project.org/doc/bib/R-books.html> adresinde bulunabilir, güncel bilgilere ulaşmak için de The R Journal (<https://journal.r-project.org>) dergisi takip edilebilir.

Bazı diğer kaynaklar aşağıda verilmiştir:

<https://www.computerworld.com/article/2497143/business-intelligence/business-intelligence-beginner-s-guide-to-r-introduction.html>

<http://swirlstats.com/>

Yorumlar, boşluklar, özel karakterler

Yorumlar: R kodları içerisinde yorumlar `#` işareti ile eklenir.

```
rmnorm(n = 100, mean = 10, sd = 2) # Ortalaması 10, Standart sapması 2 olan normal dağılımdan
# 100 genişliğinde bir örneklem üretilmiştir.
```

Dikkat: # işaretinin yanlışı yerde kullanılması kodların hata vermesine neden olabilir. #’dan sonra aynı satırda yazılan bütün kodlar yorum olarak algılanmaktadır.

```
rmnorm(10, 0, # 1)
1)
```

```
## [1] -0.33202264 0.10811568 1.07110310 -0.11781693 2.09091833 -0.09788073
## [7] -0.61185019 0.08527435 0.77475994 0.31348392
```

Boşluklar (Spacing): Çoğu R komutlarında kullanılan boşlukların miktarı sonuçlar üzerinde bir etki yapmamaktadır.

```
rmnorm(n = 3, sd = 5, mean = 2)
```

```
## [1] -4.066855 -2.434333 1.276177
```

Ancak, metin girişlerinde kullanılan boşluklar dikkate alınır.

```
text1 <- "Lorem ipsum dolor sit amet ..."
text2 <- "Lorem ipsum dolor sit amet ..."

list(Text1 = text1, Text2 = text2)
```

```
## $Text1
## [1] "Lorem ipsum dolor sit amet ..."
##
## $Text2
## [1] "Lorem ipsum dolor sit amet ..."

```

Özel karakterler: Her programlama dilinde olduğu gibi R yazılımında da bazı karakterler özel fonksiyonlar için ayrılmıştır.

Karakter	İşlev
\$	Liste, matris, veri tablosu türünden nesnelerde belirli elemanların seçilmesi için kullanılır.
%	Matrislerde temel işlemler için kullanılır.
[...], [[...]]	Vektör, matris, dizi ve listelerde boyut indisleri verilen bir elemanın seçilmesini sağlar (indeksleme).
{...}	... içerisinde yazılan kodların bir grup olarak ele alınmasını sağlar. Özellikle if ... else ... mantıksal sınamalarında kullanılır.
(...)	Fonksiyonlarda kullanılan parametrelerin tanımlanmasında kullanılır.
!	Mantıksal (logical) sınamalarda sonucu tersine çevirir. TRUE ve FALSE arası geçişi sağlar. !FALSE sonucunu TRUE yapar.
&, &&	Mantıksal sınamalarda ve (and) görevi görür.
,	Mantıksal sınamalarda veya (or) görevi görür.
:	Dizi oluşturma işlemlerinde kullanılır. 1:10, 4:9, 100:60 gibi.
;	Aynı satırda birden fazla kod bloğu yazılacak ise satır sonu belirlemek için kullanılır. x <- 5; y <- "Female" kodu tek satırda yazılmasına karşın iki satırda yazılmış kod olarak R tarafından işleme alınır. ; kullanılmaz ise bu kod hata verecektir.

R Nesneleri

R, nesne yönelimli bir programlama dilidir ve fonksiyonlar kullanılarak elde edilen sonuçlar nesnelerde saklanılmaktadır. Nesneleri tanımlamak için `=` ya da `<-` simgesi kullanılmaktadır. Daha sonra bu nesneler kullanılarak farklı işlemler yapılabilir, yeni nesneler türetilir. Nesneler büyük/küçük harfe duyarlıdır ve “ç,ğ” gibi Türkçe karakterler barındıran nesne isimleri kullanılmamalıdır. Ayrıca, nesnelere isim verilirken özel karakterler kullanılmamalı, boşluk bırakılmamalı ve nesne isimleri sayı ile başlamamalıdır. `a` ve `b` isimlerinde iki adet nesnenin oluşturulması ve bu iki nesne toplanarak yeni bir `c` nesnesinin oluşturulması aşağıda gösterilmiştir:

```
a = 4
b = -7
c = a + b
c
```

```
## [1] -3
```

```
x = a * c
x
```

```
## [1] -12
```

`x` yerine `X` çalıştırıldığında R yazılımı bu nesneyi tanımadığı için hata tanımlayacaktır. Bilgisayarda saklı mevcut nesneler `objects(...)` veya `ls(...)` fonksiyonları ile listelenebilir:

```
objects()
```

```
## [1] "a"      "b"      "c"      "text1" "text2" "x"
```

Nesnelerin koleksiyonu workspace ismini alır ve nesneler `save(...)` fonksiyonu ile kaydedilebilir, `load(...)` fonksiyonu ile çağırılabilir:

```
# Dosya uzantısı olarak Rd, Rda, Rdata uzantılarında birisi kullanılabilir.
save(a, b, c, file = "saved/savedObjects.RData")
load("saved/savedObjects.RData")
```

Nesne yapıları/sınıfları (Structure, class)

- **Skaler:** Tek değerden oluşan vektör (3, 5, “Female”, TRUE, ...)
- **Vektörler:** Benzer türden değerler dizisi. Satır vektörü, sütun vektörü.
- **Matrisler:** Birden fazla satır/sütun vektörlerinin oluşturduğu 2 boyutlu (n x p) yapı. Matrisler/Vektörler içerisinde yer alan bütün elemanlar **aynı türden olmalıdır**. Numeric vektör, integer vektör, logical vektör gibi.
- **Çok Boyutlu Dizi (Array):** Satır/Sütun vektörlerinin 3 ve üzeri boyutta kullanıldığı yapılardır. 2 Boyutlu diziler *matrisler* olarak adlandırılır. **Çok boyutlu diziler** bu kurs kapsamında incelenmeyecektir.
- **Veri Tablosu (Data Frame):** Farklı türlerden satır/sütun vektörlerinin birleşimi ile oluşan 2 boyutlu nesnelerdir. Çoğu çalışmada verilerin yapısı bu formata uymaktadır.
- **Listeler (List):** Elemanları farklı yapılardan oluşan nesnelerdir. Listenin birinci elemanı bir vektör, ikinci elemanı matris ve bir diğer elemanı ise veri çerçevesi olabilir. Bu özelliği ile listeler farklı türden elemanların bir arada bulunabildiği bir yapıdır.
- **Fonksiyonlar (Functions):** R yazılımı tarafından çalıştırılabilen kodların oluşturduğu yapılardır.

Vektörler

R’de nesneler ile çalışmak birçok kolaylık sağlar. Örneğin, bir nesne içerisinde birden fazla değer saklanabilir ve ilgilenilen bir fonksiyon ile nesnenin tüm elemanlarına işlem yaptırılabilir. Aşağıda 1’den 7’ye kadar olan sayılar vektör sınıfında bir nesnede saklanmış, daha sonra bu nesneye iki farklı işlem uygulanmıştır.

```
y = c(1, 2, 3, 4, 5, 6, 7)
y
```

```
## [1] 1 2 3 4 5 6 7
```

```
10 * y - 3
```

```
## [1] 7 17 27 37 47 57 67
```

```
y^2
```

```
## [1] 1 4 9 16 25 36 49
```

seq(...) ve rep(...) fonksiyonları kullanılarak ardışık ve tekrarlı diziler üretilebilir:

```
a = 1:5
```

```
a
```

```
## [1] 1 2 3 4 5
```

```
b = seq(-7,12,3)
```

```
b
```

```
## [1] -7 -4 -1 2 5 8 11
```

```
c = rep(7,10)
```

```
c
```

```
## [1] 7 7 7 7 7 7 7 7 7 7
```

```
d = c(a, b, c)
```

```
d
```

```
## [1] 1 2 3 4 5 -7 -4 -1 2 5 8 11 7 7 7 7 7 7 7 7 7
```

length(...) fonksiyonu kullanılarak ilgili vektörün eleman sayısı elde edilebilir:

```
length(d)
```

```
## [1] 22
```

Vektör üzerinden aritmetik ortalama, standart sapma, medyan gibi birçok tanımlayıcı istatistik elde edilebilir. Örneğin, 8 örnekten elde edilen **mir-181d** miRNA ekspresyon düzeyleri için tanımlayıcı istatistikler aşağıdaki gibi hesaplanabilir:

```
mir181d = c(2.3, 3.6, 4.7, 5.8, 2.3, 10.1, 6.5, 2.2)
```

```
mean(mir181d)
```

```
## [1] 4.6875
```

```
median(mir181d)
```

```
## [1] 4.15
```

```
sd(mir181d)
```

```
## [1] 2.740927
```

```
sum(mir181d)
```

```
## [1] 37.5
```

```
min(mir181d)
```

```
## [1] 2.2
```

```
max(mir181d)
```

```
## [1] 10.1
```

```
summary(mir181d)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.200  2.300   4.150   4.688   5.975   10.100
```

Vektör elemanları [] köşeli parantezi yardımıyla çağırılabilir:

```
mir181d[2]
```

```
## [1] 3.6
```

```
mir181d[4:7]
```

```
## [1]  5.8  2.3 10.1  6.5
```

```
mir181d[-3]
```

```
## [1]  2.3  3.6  5.8  2.3 10.1  6.5  2.2
```

Vektörler nümerik sınıfta olacağı gibi, karakter ve mantıksal sınıflarda da tanımlanabilir:

```
grup = c("karaciger", "karaciger", "karaciger", "karaciger", "bobrek", "bobrek", "bobrek", "bobrek")
grup
```

```
## [1] "karaciger" "karaciger" "karaciger" "karaciger" "bobrek"    "bobrek"
```

```
## [7] "bobrek"    "bobrek"
```

```
table(grup)
```

```
## grup
##      bobrek karaciger
##          4          4
```

```
deneme = mir181d < 5
deneme
```

```
## [1] TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE
```

```
deneme2 = (grup == "bobrek")
deneme2
```

```
## [1] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE
```

```
class(mir181d)
```

```
## [1] "numeric"
```

```
class(grup)
```

```
## [1] "character"
```

```
class(deneme)
```

```
## [1] "logical"
```

| ve & simgeleri ile mantıksal operatörler birleştirilebilir. | simgesi “veya” mantıksal operatörlerinde, & ise “ve” mantıksal operatörlerinde kullanılabilir:

```
deneme | deneme2
```

```
## [1] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE
```

```
deneme & deneme2
```

```
## [1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE
```

Mantıksal operatörleri kullanarak farklı vektörler üzerinde işlem yapmak mümkündür:

```
mir181d[deneme]
```

```
## [1] 2.3 3.6 4.7 2.3 2.2
```

```
mir181d[grup == "karaciger"]
```

```
## [1] 2.3 3.6 4.7 5.8
```

Mantıksal sınamaların sonucu TRUE veya FALSE olarak dönmektedir.

Fonksiyon	İşlev
==	Eşittir.
!=	Eşit değildir.
<	Küçüktür.
<=	Küçük veya eşittir.
>	Büyüktür.
>=	Büyük veya eşittir.
&	ve (and)
	veya (or)

Matrisler

Vektörler ile yalnızca tek bir değişken için çalışılabilmektedir. Aynı sınıftaki (nümerik, karakter, mantıksal) birden fazla vektör, matris sınıfları içerisinde tabular formatta saklanabilir. Örneğin karaciğer yağlanması rahatsızlığı olan 5 hastanın yaşları (yıl), ağırlıkları (kg), AST karaciğer değişkeni düzeyleri (ıu/l) ve mir-99a miRNA ekspresyon düzeyleri, her biri vektör olarak tanımlanıp, daha sonra `cbind(...)` veya `rbind(...)` fonksiyonları kullanılarak matris sınıfında saklanabilir:

```
yas = c(37, 61, 39, 41, 51)
kilo = c(57, 90, 60, 97, 90)
ast = c(54, 49, 97, 62, 97)
mir99a = c(0.10, 0.02, 0.01, 6.29, 0.19)
data1 = cbind(yas, kilo, ast, mir99a)
data1
```

```
##      yas kilo ast mir99a
## [1,]  37   57  54  0.10
## [2,]  61   90  49  0.02
## [3,]  39   60  97  0.01
## [4,]  41   97  62  6.29
## [5,]  51   90  97  0.19
```

```
data2 = rbind(yas, kilo, ast, mir99a)
data2
```

```
##      [,1] [,2] [,3] [,4] [,5]
## yas   37.0 61.00 39.00 41.00 51.00
## kilo  57.0 90.00 60.00 97.00 90.00
## ast   54.0 49.00 97.00 62.00 97.00
## mir99a 0.1 0.02 0.01 6.29 0.19
```

Matrislerde satır ve sütun isimleri sırasıyla `rownames(...)` ve `colnames(...)` fonksiyonları ile tanımlanabilir:


```
rownames(data1) = c("ornek1", "ornek2", "ornek3", "ornek4", "ornek5")
data1
```

```
##      yas kilo ast mir99a
## ornek1 37  57 54  0.10
## ornek2 61  90 49  0.02
## ornek3 39  60 97  0.01
## ornek4 41  97 62  6.29
## ornek5 51  90 97  0.19
```

```
colnames(data2) = rownames(data1)
data2
```

```
##      ornek1 ornek2 ornek3 ornek4 ornek5
## yas      37.0  61.00 39.00 41.00 51.00
## kilo     57.0  90.00 60.00 97.00 90.00
## ast      54.0  49.00 97.00 62.00 97.00
## mir99a    0.1   0.02  0.01  6.29  0.19
```

Matrisin devriği `t(...)` fonksiyonu ile alınabilir. Matrislerin boyutu ise `dim(...)` fonksiyonu yardımıyla öğrenilebilir. Örneğin `data2` nesnesi 4x5'lik (4 satır, 5 sütun) bir matristir.

```
t(data2)
```

```
##      yas kilo ast mir99a
## ornek1 37  57 54  0.10
## ornek2 61  90 49  0.02
## ornek3 39  60 97  0.01
## ornek4 41  97 62  6.29
## ornek5 51  90 97  0.19
```

```
dim(data2)
```

```
## [1] 4 5
```

```
dim(t(data2))
```

```
## [1] 5 4
```

Matrislerin elemanlarına ilişkin işlemlerde `[]` köşeli parantez kullanılabilir. Fakat bu sefer, hem satır, hem sütunlar köşeli parantez içerisinde belirtilmelidir. `[,]` operatörü için virgülden önceki bölüm satırları, virgülden sonrası sütunları ifade etmektedir:

```
data1[1,1]
```

```
## [1] 37
```

```
data1[3,4]
```

```
## [1] 0.01
```

```
data1[2,]
```

```
##      yas      kilo      ast mir99a
## 61.00  90.00  49.00   0.02
```

```
data1[,4]
```

```
## ornek1 ornek2 ornek3 ornek4 ornek5
##  0.10  0.02  0.01  6.29  0.19
```

```
data1[-1,]
```

```
##      yas kilo ast mir99a
## ornek2 61  90 49  0.02
## ornek3 39  60 97  0.01
## ornek4 41  97 62  6.29
## ornek5 51  90 97  0.19
```

```
data1[, -2:-3]
```

```
##      yas mir99a
## ornek1 37  0.10
## ornek2 61  0.02
## ornek3 39  0.01
## ornek4 41  6.29
## ornek5 51  0.19
```

Matrisler sıfırdan `matrix(...)` fonksiyonu yardımıyla da oluşturulabilir:

```
?matrix
data = matrix(c(37, 61, 39, 41, 51, 57, 90, 60, 97, 90, 54,
  49, 97, 62, 97, 0.10, 0.02, 0.01, 6.29, 0.19),
  nrow = 4, ncol = 5, byrow = "TRUE",
  dimnames = list(c("yas", "kilo", "ast", "mir99a"),
    c("ornek1", "ornek2", "ornek3", "ornek4", "ornek5")))
data
```

```
##      ornek1 ornek2 ornek3 ornek4 ornek5
## yas      37.0  61.00  39.00  41.00  51.00
## kilo      57.0  90.00  60.00  97.00  90.00
## ast       54.0  49.00  97.00  62.00  97.00
## mir99a     0.1   0.02   0.01   6.29   0.19
```

```
class(data)
```

```
## [1] "matrix" "array"
```

Veri Tabloları

Matris sınıflarında aynı sınıftan birden fazla vektörler saklanabilir. Veri tablolarında ise farklı türdeki sınıfa ait birden fazla vektörü saklamak mümkündür. R yazılımında verilerin istatistiksel analizleri genellikle veri tabloları sınıfındaki nesneler ile yapılmaktadır. Sebebi, veri tablolarının nitel (isimsel, sıralı) ve nicel (kesikli, sürekli) değişken yapılarında verileri saklayabilmesidir. Bu amaçla `data.frame(...)` fonksiyonundan faydalanılır.

```
yas = c(37, 61, 39, 41, 51)
kilo = c(57, 90, 60, 97, 90)
ast = c(54, 49, 97, 62, 97)
mir99a = c(0.10, 0.02, 0.01, 6.29, 0.19)
cinsiyet = c(rep("kadin", 3), rep("erkek", 2))
cinsiyet
```

```
## [1] "kadin" "kadin" "kadin" "erkek" "erkek"
```

```
fibrozis = c(rep("dusuk", 2), rep("yuksek", 3))
fibrozis
```

```
## [1] "dusuk" "dusuk" "yuksek" "yuksek" "yuksek"
```

```
DATA = data.frame(yas, cinsiyet, kilo, ast, fibrozis, mir99a)
DATA
```

```
##   yas cinsiyet kilo ast fibrozis mir99a
## 1  37   kadın  57  54   düşük  0.10
## 2  61   kadın  90  49   düşük  0.02
## 3  39   kadın  60  97   yuksek 0.01
## 4  41   erkek  97  62   yuksek 6.29
## 5  51   erkek  90  97   yuksek 0.19
```

```
class(DATA)
```

```
## [1] "data.frame"
```

Veri tablosu sınıfında saklanan DATA nesneli veriler, **faktör** (istatistik terminolojisinde nitel, qualitative) ve **nümerik** (istatistik terminolojisinde nicel, quantitative) değişken yapılarında kaydedilmiştir.

```
str(DATA)
```

```
## 'data.frame':   5 obs. of  6 variables:
## $ yas      : num  37 61 39 41 51
## $ cinsiyet: chr  "kadın" "kadın" "kadın" "erkek" ...
## $ kilo     : num  57 90 60 97 90
## $ ast      : num  54 49 97 62 97
## $ fibrozis: chr  "dusuk" "dusuk" "yuksek" "yuksek" ...
## $ mir99a   : num  0.1 0.02 0.01 6.29 0.19
```

```
summary(DATA)
```

```
##      yas      cinsiyet      kilo      ast
## Min.   :37.0   Length:5      Min.   :57.0   Min.   :49.0
## 1st Qu.:39.0   Class :character  1st Qu.:60.0   1st Qu.:54.0
## Median :41.0   Mode  :character  Median :90.0   Median :62.0
## Mean   :45.8                                     Mean   :78.8   Mean   :71.8
## 3rd Qu.:51.0                                     3rd Qu.:90.0   3rd Qu.:97.0
## Max.   :61.0                                     Max.   :97.0   Max.   :97.0
##      fibrozis      mir99a
## Length:5      Min.   :0.010
## Class :character  1st Qu.:0.020
## Mode  :character  Median :0.100
##                                     Mean   :1.322
##                                     3rd Qu.:0.190
##                                     Max.   :6.290
```

Veri tablolarında [] köşeli parantezler ile veri elemanları çağrılabilir. Lakin, veri tablolarında sütunlar genellikle değişkenleri temsil ettiği için, sütunlar daha büyük öneme sahiptir. Değişken isimleri ve \$ simgeleri ile değişkenlere ilişkin veri çağrılabilir:

```
DATA[,4]
```

```
## [1] 54 49 97 62 97
```

```
DATA[, "ast"]
```

```
## [1] 54 49 97 62 97
```

```
DATA$ast
```

```
## [1] 54 49 97 62 97
```

Listeler

Listeler, farklı türden R nesnelerini (skaler, vektör, matris, veri tabloları, fonksiyonlar, vb.) içerisinde barındırabilen yapılar olarak tanımlanabilir. Listeler içerisinde başka listeleri de saklayabilme özelliğine sahiptir.

R programlama dilinde yazılan fonksiyonların büyük bir bölümünde sonuçlar listeler içerisinde kullanıcıya döndürülmektedir. Listeler `list(...)` fonksiyonu kullanılarak oluşturulabilir.

```
# x, y, z, ... elemanları bir liste içerisinde saklanmaktadır.
liste <- list(x, y, z, ...)
```

- **Örnek:** İki hastaya ait çeşitli bilgileri (Ad, Soyad, Cinsiyet, Yaş, Toplam Kolesterol, vb.) bir liste içerisinde saklayalım.

```
hasta1 <- list("John", "Doe", "Erkek", 32, 435)
hasta2 <- list("Doe", "Kadın", 35, "Jane", 295)
```

```
# Listenin belirli bir elemanının seçilmesi
hasta1[1]
```

```
## [[1]]
## [1] "John"
```

```
hasta1[[1]]
```

```
## [1] "John"
```

```
hasta2[[4]]
```

```
## [1] "Jane"
```

hasta1 ve **hasta2** listelerinde yer alan hasta bilgileri karışık sıralama ile verilmiştir. Liste içerisinde yer alan liste elemanları karışıklığı önleyebilmek adına isimlendirilerek kaydedilebilir.

```
hasta1 <- list(Ad = "John", Soyad = "Doe", Cinsiyet = "Erkek", Yas = 32, TOTCHOL = 435)
hasta2 <- list(Soyad = "Doe", Cinsiyet = "Kadın", Yas = 35, Ad = "Jane", TOTCHOL = 295)
```

```
# Listenin belirli bir elemanının seçilmesi
hasta1[1]
```

```
## $Ad
## [1] "John"
```

```
hasta1["Ad"]
```

```
## $Ad
## [1] "John"
```

```
hasta1[["Ad"]]
```

```
## [1] "John"
```

```
hasta1$Ad
```

```
## [1] "John"
```

Nesnelerde sınıflar arası dönüşümler

`as.<nesne_sınıfı veya yapısı>(...)` fonksiyon grupları ile nesneler ve/veya sınıflar arası dönüştürme işlemleri yapılabilir.

- Sınıflar arası dönüşümler: `as.numeric()`, `as.character()`, `as.logical()`, `as.integer()`, `as.Date()`
- Nesne yapıları arası dönüşümler: `as.vector()`, `as.matrix()`, `as.data.frame()`, `as.list()`

`is.<nesne_sınıfı veya yapısı>(...)` fonksiyon grupları ile bir R nesnesinin ve içeriğinin ilgili nesne yapısına ve/veya veri sınıfına uyup uymadığı kontrol edilebilir.

- Veri sınıfının kontrolü: `is.numeric()`, `is.logical()`, ...
- Nesne yapısının kontrolü: `is.vector()`, `is.data.frame()`, ...

```
vec <- c(30, 32, 26, 48)
```

```
# vec nesnesinin sınıfı ("numeric vector")
class(vec)
```

```
## [1] "numeric"
```

```
vec2 <- as.matrix(vec) # Sütun matrisi
class(vec2) # Sütun matrisi
```

```
## [1] "matrix" "array"
```

```
is.matrix(vec2)
```

```
## [1] TRUE
```

```
vec2 <- as.data.frame(vec) # Veri tablosu
is.data.frame(vec2)
```

```
## [1] TRUE
```

```
vec2 <- as.list(vec) # Liste
is.list(vec2)
```

```
## [1] TRUE
```

```
is.matrix(vec2)
```

```
## [1] FALSE
```