R ile Programlama

Veriler, işlemler ve grafikler

Dr. Halil Yurdugül
yurdugul@hacettepe.edu.tr

Komut Satırı ve Temel İşlemler

- R betik tabanlı (script based) bir istatistiksel işlem yazılımıdır.
- Her ne kadar RStudio, EMACS, RCommander, Rattle, R AnalyticFlow, Tinn-R gibi kullanıcı dostu arabirimler geliştirilmiş olsa da temelde R yazılımı bir komut istemcisi (prompt) ile başlar.
- > Bu durumda kullanıcıdan komut girmesi beklenir. Bu komutlar değişkensiz işlemlere dayanacağı gibi değişkenleri içeren işlemleri de kapsamaktadır.
- > Bu işlemler aritmetik işlemler olacağı gibi mantıksal işlemler de olabilir.
- R yazılımında aynı zamanda komutlar bütünü olan programlar oluşturulabilir. Bu bağlamda R yapısal programlama ilkeleriyle programlamaya da izin verir.

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Temel İşlemler ve Operatörler

Operatör	Açıklama	Komut Gösterimi	Komut Çıktısı
+, -, *, /	Dört işlem	> 3+4-2*5/10	> 6
( )	Parantez	> (3+4)-(2*5/10)	> 6
^	Üs alma	> 3 ^ 4	> 81
%%	Bölmede kalan	> 7 %% 2	> 1
%/%	Bölmede tamsayı	> 7 %/% 2	> 3

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Matematiksel İşlem Fonksiyonları

Fonksiyon Adı	Açıklama
abs(*x*)	Mutlak değer
sqrt(*x*)	Karekök
ceiling(*x*)	Yukarı yuvarlama (3.475) → 4
floor(*x*)	Aşağı yuvarlama (3.475) → 3
trunc(*x*)	Tamsayı kısmını alma (5.99) → 5
round(*x*, **digits=***n*)	Yuvarlama (3.475, digits=2) → 3.48
cos(*x*), **sin**(*x*), **tan**(*x*)	also acos(*x*), cosh(*x*), acosh(*x*), etc.
log(*x*), **log10**(*x*)	Doğal ya da tabana göre logaritma
exp(*x*)	Üstel ifade e^*x*

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Mantıksal İşlemler ve Operatörler

Operatör	Açıklama	Komut Gösterimi	Komut Çıktısı
<, <=	Küçük, küçük eşit	> x = 1 < 2	> TRUE
>, >=	Büyük, büyük eşit	> x = 1 >= 2	> FALSE
==, !=	Eşit, eşit değil	> x = 1 != 2	> TRUE
x | y	Veya	> x = 1>2 | 3>4	> TRUE
x & y	Ve	> x = 1>2 & 3>4	> FALSE
!x	Değil	> x=1>2; !x	> TRUE

R'da Değişkensiz İşlemler

R ortamında komut satırından bu yazılımı gelişmiş bir hesap makinesi olarak kullanmak olanaklıdır:

```
> 1/2-4+4^2                                  → 12,5

> (7 %% 5) * 12 - sqrt(18*2) / log(exp(1))   → 18

> 2 + 4 / 2 + 4                              → 8

> (2 + 4) /( 2 + 4)                          → 1
```

Bu tür işlemler belleğin veri segmentini değil yığıt segmentini kullanırlar. Yapısal bir programlama için değişkenleri kullanmak gerekir.

R'da değişkenler

Değişkenler verilerin bellek adreslerini tutan sembollerdir. Örneğin bellekte bir hücrenin adresi x olsun. Bu hücreye 12 değerini atamak için 3 seçenek vardır. Bunlar sırasıyla atama operatörü (=), nesne işaretçisi (<-) ve assign("*değişken adı*", *değer*) fonksiyonudur.

Aşağıdaki her üç komutta aynı işlevi görür.

```
> x = 12
> x <- 12
> assign("x", 12)
```

Yapısal programlama ile uğraşanlar atama operatörünü, nesne yönelimli programlama ile uğraşanlar nesne işaretçisini daha çok tercih etmektedirler.

R’da veri türleri ve veri yapıları

Birazcık programlama bilgisi:
Bilgi işlem düzeyinde değişkenler bellek adreslerini tutan semboller iken, ilgili adres kaç bellek hücresiyle ilişkilendirileceği ise veri türlerinin konusudur. Örneğin C ya da türevi dillerde;

```
int a;
a=2;
```

ifadesinde 1. satırda iki hücrelik (16 byte) alan ayrılır ve adresi a ile temsil edilirken 2. satırda ise ilgili adrese 2 sayısı (000...010) değeri atanır.

Veri türleri daha çok bellek yönetimi ile ilgili bir kavram iken veri yapıları ise dinamik bellek yönetimi ile ilgili bir kavramdır.

Alt elemanları atomik veri türlerinden int, char, float oluşan çoklu değişkenlere ilişkin veri yapıları (data structure) söz konusudur.

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

R’da veri türleri

Veri Türü	Açıklama	Komut Satırı	Komut Çıktısı
Numeric	Doğal sayılar	> x = 3.4 > class(x)	> “numeric”
Integer	Tamsayılar	> x = 34L > class(x)	> “integer”
Character	Harf ya da metin	> x = “Merhaba” > class(x)	> “character”
Logical	TRUE-FALSE	> x = TRUE > class(x)	> “logical”
Complex	Sanal sayılar	> x = 1 + 4i > class(x)	> “complex”

Veri yapıları

Birden fazla ilişkili tekil veri bir araya gelerek veri yapılarını oluşturur. Bu tekil veriler aynı veri türlerinden (homojen) olabileceği gibi farklı veri türlerinden de (heterojen) oluşabilmektedir.

Boyut	Homojen	Heterojen
Tek Boyutlu	Vector	
İki Boyutlu	Matrix	DataFrame
Çok Boyutlu	Array	List

Bu sınıflamanın haricinde bir de Factor adlı veri yapısı da söz konusudur.

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Vektörler

Vectors

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Vektörler

- Vektörler aynı veri türüne sahip tekil verilerin bir araya gelmesiyle oluşur. Tek boyutludur.
- Bir dizi yaratmak için yaygın olarak c() fonksiyonundan yararlanılır.
- > **x** <- **c**(“Ece”,”Efe”,”Can”,”Oya”,”Gül”,”Ege”)
- > **x** <- **c**(TRUE, FALSE, TRUE, TRUE, FALSE)
- > **x** <- **c**(1, 2, 5, 7, 9, 4, 3, 2, 5)
- > str(**x**) → num [1:9] 1 2 5 7 9 4 3 2 5
- > typeof(**x**) → “double”
- > **y** <- 1/**x**
- > **y** <- round(1/**x**, digits=2)
- > **z** <- ((**x**-mean(**x**))^2)/(length(**x**)-1)
- > **w** <- **x** > 5

Vektör elemanlarına erişim

- Yas <- **c**(9, 8, 6, 7)
- Yas[1] → 9
- Yas[4] → 7
- Yas[c(1,2,4)] → 9, 8, 7
- Yas[1:3] → 9, 8, 6
- Yas <- **c**(Ece=9, Efe=8, Can=6, Oya=7)
  names(Yas) → "Ece" "Efe" "Can" "Oya"
  Yas[“Ece”] → 9
- Yas[“Oya”] → 7
- typeof(Yas) → "double"
  class(Yas) → "numeric"
  attributes(Yas) → $names "Ece" "Efe" "Can" "Oya"

Vektörlere ilişkin fonksiyonlar

- Vektörlerde ardışık değerler için seq() fonksiyonundan, tekrarlı değerler için rep(), sıralı veriler için de sort() fonksiyonundan sıkça yararlanılır.
- x <- 2:6 → 2, 3, 4, 5, 6
- x <- seq(2,6) → 2, 3, 4, 5, 6
- x <- seq(2, 6, by=0.5) → 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0
- x <- seq(6, 2, by=-0.5) → 6.0, 5.5, 5.0, 4.5, 4.0, 3.5, 3.0, 2.5, 2.0
- **seq**(*from*, *to*, by/length)
- seq(1, 10, by=3) → 1 4 7 10
- seq(1, 10, length=3) → 1.0 5.5 10.0
- **rep**(değer, tekrar)
- rep(1:4, 1) → 1, 2, 3, 4
- rep(1:4, 2) → 1, 2, 3, 4, 1, 2, 3, 4
- rep(1:4, each=2) → 1, 1, 2, 2, 3, 3, 4, 4

Vektörler ile ilgili fonksiyonlar

- > **x** <- **c**(2, 3, 2, 4, 3, 5, 6, 7, 6, 8)
  > length(x) → 10
  > unique(x) → 2 3 4 5 6 7 8
  > duplicated(x) → F F T F T F F F T F
  > rev(x) → 8 6 7 6 5 3 4 2 3 2
  > sort(x) → 2 2 3 3 4 5 6 6 7 8
  > append(x,5) → 2 3 2 4 3 5 6 7 6 8 5
  > sum(x) → 46
  > min(x) → 2
  > max(x) → 8
  > sd(x) → 2.1187
  > var(x) → 4.488889
- > median(x) → 4.5

Vektörler ile ilgili fonksiyonlar

- > x <- **c**(2, 3, 2, 4, 3, 5, 6, 7, 6, 8)
- > y <- **c**(3, 2, 2, 6, 4, 5, 6, 8, 8, 9)

> cor(x, y) → 0.9329528
> cov(x, y) → 5.022222
> setdiff(x, y) → 7
> setdiff(y, x) → 9
> rank(x) → 1.5, 3.5, 1.5, 5.0, 3.5, 6.0, 7.5, 9.0, 7.5, 10.0
> all(x>5) → FALSE (2, 3, 2, 3, 4, 5)
> any(x>5) → TRUE (6, 7, 6, 8)
> all(x==y) → FALSE
> range(x) → 2, 8
> diff(range(x)) → 6

Vektörler ile ilgili notlar

- Bir vektörden eleman eksiltmek/silmek için indis değeri önünde “-” kullanmak gerekir.

```
x <- c(2, 3, 2, 4, 3, 5, 6, 7, 6, 8)
y <- x[-5]          → 2, 3, 2, 4, 5, 6, 7, 6, 8
y <- x[-c(3:5)]     → 2, 3, 5, 6, 7, 6, 8
y <- x[-c(1,3,5)]   → 3, 4, 5, 6, 7, 6, 8
```

- c() ile oluşturulan vektörler “numeric”, seq(1:5) ile oluşturulan vektörler ise “integer” veri türüne sahiptir.
- R’da vektör indisleri 1’den başlar. C programcıları buna dikkat etmeli.
- Vektörün 0. elemanı metadata için kullanılır. Örneğin x[0] → “numeric” ifadesiyle karşılaşırlar.

Vektör cebri

- Bir vektör üzerindeki skaler ya da birden fazla vektör üzerinden yapılan matematiksel işlemler vektör cebri olarak adlandırılır.

```
> x <- c(1, 3, 5, 7, 9)
> y <- c(2, 4, 6, 8, 0)

> z <- x + y              → 3  7 11 15  9
> z <- x - y              → -1 -1 -1 -1  9
> z <- x * y              → 2 12 30 56  0
> z <- x / y              → 0.500 0.750 0.833 0.875 Inf
> z <- (1 / x) + sqrt(y)  → 2.414 2.333 2.649 2.971 0.111
```

Matrisler

Matrix

Matrisler (Matrix)

- Matrisler, iki boyutlu vektörler olarak düşünülebilir. Matrisler de vektörler gibi aynı veri türüne sahip değerlerin bir araya gelmesiyle oluşur.
- Bir matris oluşturmak için yaygın olarak matrix() fonksiyonundan yararlanılır.
-
- > **x** <- **matrix**(değerler, satır_sayi, kolon_sayi)
- > **x** <- **matrix**(1:15, 5, 3)
-

```
     [,1] [,2] [,3]
[1,]    1    6   11
[2,]    2    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

Bu komut çıktısında bir gariplik var.
Matrise sıralı erişim kolonlardan başlamış.
Bunun önüne geçmek için byrow=TRUE parametresine ihtiyaç var.

Matrisler (Matrix)

- Matrisler, iki boyutlu vektörler olarak düşünülebilir. Matrisler de vektörler gibi aynı veri türüne sahip değerlerin bir araya gelmesiyle oluşur.
- Bir matris oluşturmak için yaygın olarak matrix() fonksiyonundan yararlanılır.
-
- > **x** <- **matrix**(değerler, satır_sayi, kolon_sayi)
- > **x** <- **matrix**(1:15, 5, 3, byrow=TRUE)
-

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
[5,]   13   14   15
```

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Matrisler (Matrix)

- Matrisler, iki boyutlu vektörler olarak düşünülebilir. Matrisler de vektörler gibi aynı veri türüne sahip değerlerin bir araya gelmesiyle oluşur.
- Bir matris oluşturmak için yaygın olarak matrix() fonksiyonundan yararlanılır.
-
- > **x** <- **matrix**(c(1,3,5,7,9,2,4,6,8,0), nrow=2, ncol=5)
- [,1] [,2] [,3] [,4] [,5]
- [1,] 1 5 9 4 8
- [2,] 3 7 2 6 0

  Matrisin hücrelerine erişim ve değiştirme...
- > x[1,4] → 4
- > x[2,5]= 10
- Satır ve kolonlara erişim
- > x[2,] ya da x[,4] ya da x[c(1,2),] ya da x[, c(2,4)]

Matrisler (Matrix)

- Matrisler, iki boyutlu vektörler olarak düşünülebilir. Matrisler de vektörler gibi aynı veri türüne sahip değerlerin bir araya gelmesiyle oluşur.
- Bir matris oluşturmak için yaygın olarak matrix() fonksiyonundan yararlanılır.
- *Vektörü matrise dönüştürme*
- > **x** <- **c**(1,3,5,7,9,2,4,6,8,0)
- dim(**x**) → NULL ???
- dim(**x**) <- c(2,5)
- > **x**

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9    4    8
[2,]    3    7    2    6    0
```

- *Matrisi vektöre dönüştürme*
- **x**<-as.vector(x)

Matris cebri (Matrix algebra)

- *Matris transpozu*
- > x <- **matrix**(**c**(1,3,5,7,9,2,4,6,8,0), 5, 2)
- ```
       [,1] [,2]
  [1,]    1    2
  [2,]    3    4
  [3,]    5    6
  [4,]    7    8
  [5,]    9    0
  > y <- t(x)
       [,1] [,2] [,3] [,4] [,5]
  [1,]    1    3    5    7    9
  [2,]    2    4    6    8    0
  ```
  ```

Matris cebri (Matrix algebra)

- *Matris birleştirme*
- İki matrisi **cbind**() ve/veya **rbind**() fonksiyonu ile birleştirebiliriz.
- > x <- **matrix**(**c**(1,3,5,7,9,2,4,6,8,0), 5, 2)
- > y <- **matrix**(**c**(5,5), 1, 2)
- > w<- **matrix**(**c**(5,5,5,5,5), 5, 1)
-

	[,1]	[,2]
[1,]	1	2
[2,]	3	4
[3,]	5	6
[4,]	7	8
[5,]	9	0

	[,1]	[,2]
[1,]	1	2

	[,1]
[1,]	1
[2,]	2
[3,]	3
[4,]	4
[5,]	5

- > z <- **rbind**(x,y)
- > z <- **cbind**(x,w)

Matris cebri (Matrix algebra)

- *Matris toplama ve çıkarma*
- İki matrisinde satır ve kolon sayılarının aynı (eşit) olması gerekir.
- > x <- **matrix**(**c**(1,2,3,4,5,6), 3, 2)
- > y <- **matrix**(**c**(6,5,4,3,2,1), 3, 2)
- > **z<- x + y**
- > **z<- x - y**

- x <- **matrix**(**c**(1,2,3,4,5,6), 3, 2) ile matrix(1:6,3,2) aynı ifadedir.
- y <- **matrix**(**c**(6,5,4,3,2,1), 3, 2) ile matrix(6:1,3,2) aynı ifadedir.

Matris cebri (Matrix algebra)

- *Matris çarpımı*
- İki matrisin çarpımında, her iki matrisin boyutlarının matris çarpım kuralına uygun (soldaki matrisin kolon sayısı ile sağdaki matrisin satır sayısının eşit olma ilkesi) olması gerekir.
- > x <- **matrix**(**c**(1,2,3,4,5,6), 3, 2)
- > y <- **matrix**(**c**(6,5,4,3,2,1), 2, 3)
- > **z<- x * y**
  - **R hata uyarısı döndürdü…**
  - **Bir de aşağıdaki komutu deneyiniz...**
- > **z<- x %*% y**
- Sonuç matrisinin (z) satır ve kolon sayılarına dikkat ettiniz mi? Z(3x3) → x matrisinin satır sayısı ile y matrisinin kolon sayılarından oluşmaktadır.
- Vektör-matris çarpımı → **x %*% Y %*% x** (x vektör, Y matris)

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Matris fonksiyonları

- *Matris cebrini içeren fonksiyonlar*
- Bir önceki Z matrisini kullanarak aşağıdaki fonksiyonları uygulayınız:
- > diag(**z**)
- > diag(**5**)
- > rowMeans(**z**)
- > colMeans(**z**)
- > rowSums(**z**)
- > colSums(**z**)
- > solve(**z**) → Matris tersi
- > eigen(**z**) → Matris özdeğerleri
- > svd(**z**) → Single value decomposition
- > col(**z**) → Matrisin Cholesky bileşenini verir. $Z=A.A^{-1}$

Diziler

Arrays

Diziler (Array)

- vektörler tek boyutlu, matrisler iki boyutlu tablolar olarak düşünülürse diziler ise ikiden daha fazla boyuta sahip nümerik veri yapılarıdır.
- C kökenli programlamada geçen array() kavramı/fonksiyonu ile karışabilir, bu nedenle kafa karışıklığını önlemek gerekir. Aslında diziler, vektörler ve matrislerin genelleştirilmiş biçimi olarak da görülebilir.
- diziler için **array**(veri, boyut_lar) fonksiyonundan yararlanılır.
-
- Aşağıdaki komutları deneyerek gözlemleyiniz.
- > array(1:18, c(18))
- > array(1:18, c(2, 9))
- > array(1:18, c(2, 3, 3))
- son komuttaki 3. boyuta dikkat ettiniz mi?
- > array(0, c(2, 3, 3, 2)) bunu ya da array(0, c(2, 3, 3, 2, 2)) denemeyin bile :)

Listeler

List

Liste (List)

- Vektörlerin tüm elemanları aynı veri türüne (numeric, character, logical) sahiptir. Listelerde ise elemanların hepsinin aynı türde olması gerekmez. Özellikle C programlamada vektörler array() iken listeler ise struct() veri yapısına karşılık gelir.
-
```
> z <- list(5, "Ece", TRUE)
> z
[[1]]
     [1] 5
[[2]]
     [1] "Ece"
[[3]]
     [1] TRUE
```

Liste (List)

- Vektörlerin tüm elemanları aynı veri türüne (mümeric, character, logical) sahiptir. Listelerde ise elemanların hepsinin aynı türde olması gerekmez. Özellikle C programlamada vektörler array() iken listeler ise struct() veri yapısına karşılık gelir. Vektörler oluşturmak için c() fonksiyonunfan, listelerde ise list() fonksiyonundan yararlanılır.

- liste <- **list**(ad="Ece", num=974, yas=9, cins=”K”, ders=**c**(mat=5, fen=4, muzik=5))
- > liste[[1]] → [1] "Ece"
  > liste[["ad"]] → [1] "Ece"
  > liste$ad → [1] "Ece"
- > liste[["ders"]]
- > liste[["ders"]]["mat"]

Liste (List)

- Listeler özünde jenerik bir yapıdır ve eleman olarak yalnızca tekil değişkenler değil aynı zamanda diğer veri yapılarını da (vektör, matris vb) eleman olarak alabilir.
- > x = **c**(2,3,4)
- > y = **c**(”ece”,”efe”,”ege”, “oya”, “can”, “cem”)
- > z = **c**(TRUE, FALSE, TRUE, FALSE)
- > w = **matrix**(1:6, 3, 2)
- > liste = **list**(x, y, z, w, 3.14)
- > liste

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Listelerde indeks

- Listeler özünde jenerik bir yapıdır ve eleman olarak yalnızca tekil değişkenler değil aynı zamanda diğer veri yapılarını da (vektör, matris vb) eleman olarak alabilir.
- > liste[2] → "ece" "efe" "ege" "oya" "can" "cem"
- > liste[[2]] → "ece" "efe" "ege" "oya" "can" "cem"
- > liste[[2]][3] → "ege"
- > liste[[2]][3]=”ali”
- > liste[[2]][3] → "ali"

Görüldüğü gibi listeler nesnelerden oluşan tekboyutlu bir dizidir. Dolayısıyla her bir elemanı atomik değer, vektör, matris vb olabilir. Alt el

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Listelerde indeks

- Listelerde sayısal indeks yerine *kullanıcı tanımlı indeks*lerde kullanılabilir.
- Bunun için daha önce vektörlerde bahsi geçen adlandırmalara ihtiyaç vardır.
- > liste = list(ad=c(“ece”, “ege”, “efe”), puan=c(5,3,1), cinsiyet=c(“K”,”E”,”E”))
- > liste

```
$ad
    [1] "ece" "ege" "efe"
$puan
    [1] 5 3 1
$cinsiyet
    [1] "K" "E" "E"
```

- > liste$ad
- > liste$puan[3] = 2

Listelerde indeks

- Listelerde sayısal indeks yerine *kullanıcı tanımlı indeks*lerde kullanılabilir.
- Bu durumda liste üyelerine erişim için $ operatörünü kullanak gerekir.
- Ancak listeyi R'ın çalışma alanında ilişkilendirir isek alt üyelere erişim için $ operatörüne de ihtiyaç kalmaz.
-
- > liste = list(ad=c("ece", "ege", "efe"), puan=c(5,3,1), cinsiyet=c("K","E","E"))
- > **attach**(liste)
  > puan → [1] 5 3 1
  > ad → [1] "ece" "ege" "efe"
  > cinsiyet[2] → [1] "E"
  > **detach**(liste)
  > puan → Hata: 'puan' nesnesi bulunamadı

Veri Çerçevesi

Data Frames

Veri Çerçeveleri (Data frames)

- Alt elemanları aynı veri türünde (özellikle nümerik) olan veri yapıları sırasıyla vektör (tek boyutlu), matris (iki boyutlu) ve dizi (çok boyutlu) olarak adlandırılmıştır.
- Aynı veri türüne sahip olmayan veri yapıları ise tek boyutlu olma durumunda liste, iki boyutlu olma durumunda ise veri çerçevesi olarak ifade edilir.
- Veri çerçeveleri R'da en yaygın kullanılan veri yapılarıdır.
- Veri çerçevelerini bir tablo gibi düşünebilirsiniz...
-

Ad	Cinsiyet	Yaş	Boy	Göz Rengi
Ece	K	9	1,34	Ela
Efe	E	8	1.28	Yeşil
Ege	E	7	1.15	Mavi

Veri Çerçeveleri (Data frames)

```
> x <- c("Ece","Efe","Ege")
> y <- c("K","E","E")
> z <- c(9,8,7)
> w <- c(1.34, 1.28, 1.15)
> t <- c("Ela", "Yeşil", "Mavi")
```

```
> veri <- data.frame(x,y,z,w,t)
> names(veri) <- c("Ad", "Cinsiyet", "Yaş", "Boy", "GözRenk")
> veri
  Ad Cinsiyet Yaş  Boy GözRenk
1 Ece        K   9 1.34     Ela
2 Efe        E   8 1.28   Yeşil
3 Ege        E   7 1.15    Mavi
```

Ad	Cinsiyet	Yaş	Boy	Göz Rengi
Ece	K	9	1,34	Ela
Efe	E	8	1.28	Yeşil
Ege	E	7	1.15	Mavi

Veri Çerçeveleri (Data frames)

```
  Ad Cinsiyet Yaş  Boy GözRenk
1 Ece        K  9 1.34     Ela
2 Efe        E  8 1.28   Yeşil
3 Ege        E  7 1.15    Mavi
```

```
> veri[1,4]
[1] 1.34
> veri[2,]
  Ad Cinsiyet Yaş  Boy GözRenk
2 Efe        E  8 1.28   Yeşil
> veri[, 5]
[1] Ela   Yeşil Mavi
Levels: Ela Mavi Yeşil
> veri$Ad
[1] Ece Efe Ege
Levels: Ece Efe Ege
> attach(veri)
> Yaş
 [1] 9 8 7
```

Veri Çerçeveleri (Data frames)

```
  Ad Cinsiyet Yaş  Boy GözRenk
1 Ece        K  9 1.34     Ela
2 Efe        E  8 1.28   Yeşil
3 Ege        E  7 1.15    Mavi
```

```
> veri[1,4]
```
→ [1] 1.34

```
> veri[2,]
```
→
```
  Ad Cinsiyet Yaş  Boy GözRenk
2 Efe        E  8 1.28   Yeşil
```

```
> veri[, 5]
```
→ [1] Ela Yeşil Mavi
Levels: Ela Mavi Yeşil

```
> veri$Ad
```
→ [1] Ece Efe Ege
Levels: Ece Efe Ege

```
> attach(veri)
> Yaş
```
→ [1] 9 8 7

Faktörler

Factors

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Faktörler (Factors)

Bazı vektör değerleri tekrar eden değerler olabilir. Bu değerleri özellikle bir kategori altında toplamak için **factor**() fonksiyonundan yararlanılır.

```
> X <- c("Erkek","Erkek","Erkek","Kadın","Kadın","Erkek","Kadın","Erkek")
> Y <- factor(X)
> Y              → "Erkek","Erkek","Erkek","Kadın","Kadın","Erkek","Kadın","Erkek"
                 → Levels: Erkek Kadın
> table(Y)       →   Y
                 → Erkek Kadın
                 →     5       3
> Y <- table(Y)

> barplot(Y)
```

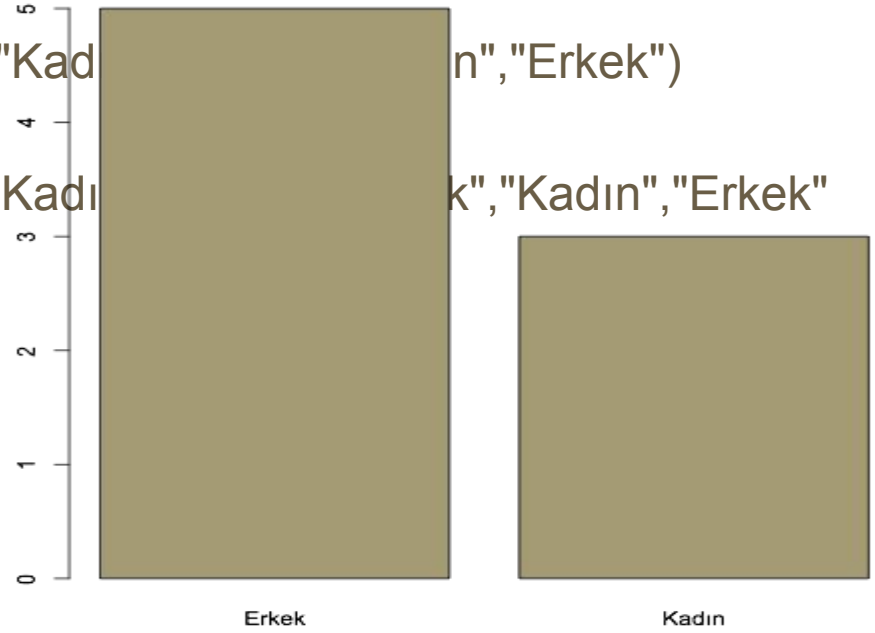
Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Faktörler (Factors)

Bazı vektör değerleri tekrar eden değerler olabilir. Bu değerleri özellikle bir kategori altında toplamak için factor() fonksiyonundan yararlanılır.

```
> X <- c("Erkek","Erkek","Erkek","Kadın","Kad                    n","Erkek")
> Y <- factor(X)
> Y              → "Erkek","Erkek","Erkek","Kadı                 k","Kadın","Erkek"
                 → Levels: Erkek Kadın
> table(Y)       →  Y
                 → Erkek Kadın
                 →    5       3
> Y <- table(Y)

> barplot(Y)
```



Faktörler (Factors)

Bazı vektör değerleri tekrar eden değerler olabilir. Bu değerleri özellikle bir kategori altında toplamak için **factor**() fonksiyonundan yararlanılır.

```
> X <- c("Erkek","Erkek","Erkek","Kadın","Kadın","Erkek","Kadın","Erkek")
> Y <- factor(X)
> Y              → "Erkek","Erkek","Erkek","Kadın","Kadın","Erkek","Kadın","Erkek"
                 → Levels: Erkek Kadın
> table(Y)       →   Y
                 → Erkek Kadın
                 →     5       3
> Y <- prop.table(Y)

> barplot(Y)
```

Faktörlerin Farklı Kullanımları

SPSS'te ya da diğer istatistiksel yazılımlarda kategorilerin nümerik olarak kodlandığı bilinen bir durumdur.

```
> Cinsiyet = factor(c(1, 0, 1, 0, 0, 0, 1, 1, 0), levels = c(0, 1), labels = c("K", "E"))
> Y <- factor(Cinsiyet)
> Y                → K E K E E E K K E
                   →  Levels: E K
> table(Y)         →   Y
                   →  E K
                   →  5  3
> barplot(Y)

>
```

Tablolar

Tables

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Tablolar (Tables)

Her ne kadar bir veri yapısı olarak adlandırılmasa da tablolar özellikle matris ve veri çerçeveleri gibi iki boyutlu veriler için R içinde kullanımı yaygın bir fonksiyondur. Tablolar özellikle R kapsamında frekanslar ve özet istatistikler için önemli bir veri biçimidir. Bir bakıma tablolar, matris ya da veri çerçevelerinin tablolaştırılmış halidir.

```
> x <- matrix(c(2,3,4,3,3,4,5,6,5,4,6,3),3,3)
> y <- table(x)      → 2  3  4  5  6     ← değer grupları
                     → 1  3  2  2  1     ← frekanslar
> z <- as. table(x)      →        A   B   C
                         →   A    2   3   5
                         →   B    3   3   6
                         →   C    4   4   6
```

Tablolar (Tables)

Her ne kadar bir veri yapısı olarak adlandırılmasa da tablolar özellikle matris ve veri çerçeveleri gibi iki boyutlu veriler için R içinde kullanımı yaygın bir fonksiyondur. Tablolar özellikle R kapsamında frekanslar ve özet istatistikler için önemli bir veri biçimidir. Bir bakıma tablolar, matris ya da veri çerçevelerinin tablolaştırılmış halidir.

```
> x <- matrix(c(2,3,4,3,3,4,5,6,5,4,6,3),3,3)
> y <- table(x)      → 2  3  4  5  6      ← değer grupları
                     → 1  3  2  2  1      ← frekanslar
> mean(y)
> sum(y)
> var(y)
> sd(y)
> quantile(y)
```

Dosya İşlem

I/O Process

Dosya İşlemleri (File Processing)

R ortamındaki verileri disk yüzeylerine kaydedilmesi ve/veya disk yüzeylerindeki verileri R ortamına aktarılması veri dosyaları işlemi olarak nitelendirilir.

Disk yüzeyine yazmak ve okumak için save()/load() ve write()/read.table() fonksiyonlarından yararlanılır. Burada

save(yazılacak veri, file=dosya_adi)

biçiminde kullanılır. Bu fonksiyon verileri belirtilen dosyaya ikili formatta yazdırır ve genellikle uzantısı da *.RData şeklindedir. Bu şekilde saklanan dosyalar

load(dosya_adi) şeklinde R ortamına aktarılır.

Bunlara ek olarak; eğer dosya adını windows/macintosh işletim sistemlerinin dosya buldurucusundan etkileşimli olarak buldurmak için

load(file= file.choose())

fonksiyonundan da yararlanılabilir.

Dosya İşlemleri (File Processing)

```
> veri <- c(ece=8, efe=4, ege=6, can=2, cem=7)
> save(veri, file="deneme.Rdata")
```

Adı verilen dosyayı disk yüzeyinde bulup herhangi bir text editörü ile açarsanız verilerin binary formatta saklandığını görebilirsiniz.

```
> load("/Users/halilyurdugul/deneme.Rdata")
```

ya da

```
> zz <- load("/Users/halilyurdugul/deneme.Rdata")
> zz                → [1] "veri"
> veri
ece efe ege can cem
  8   4   6   2   7
>
```

Dosya İşlemleri (File Processing)

R ortamındaki verileri disk yüzeylerine kaydedilmesi ve/veya disk yüzeylerindeki verileri R ortamına aktarılması veri dosyaları işlemi olarak nitelendirilir.

Verileri disk yüzeyine txt formatında yazmak ve okumak için write()/read.table() fonksiyonlarından yararlanılır. Burada eğer veri vektör yapısında ise write(), eğer matrix ya da veri çerçevesi yapısında ise write.table() kullanılır.

write(yazılacak veri, file=dosya_adi, ayraç)

biçiminde kullanılır. Ayraç belirtilmezse tab karakteri kullanılır. Bu şekilde kaydedilmiş dosyaları açmak için ise

read.table(dosya_adi, ayraç) şeklinde R ortamına aktarılır.

Dosya İşlemleri (File Processing)

```
> veri <- c(ece=8, efe=4, ege=6, can=2, cem=7)
> write(veri, file="deneme.txt")
```

Burada ayraç kullanılmadığı için dosyada veriler tab karakteriyle ayrılmıştır. Veri metin formatı txt olduğu için bir text editöründen saklanan veriler görülebilir.

```
> read.table(file="/Users/halilyurdugul/deneme.txt")
```

```
 V1 V2 V3 V4 V5
  8  4  6  2  7
```

Dosya İşlemleri (File Processing)

```
> x <- matrix(c(2,3,4,3,3,4,5,6,5,4,6,3),3,3)
> write.table(x, file="denemex.txt")

> read.table(file="denemex.txt", header=TRUE)
  V1 V2 V3
1  2  3  5
2  3  3  6
3  4  4  5
```

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Diğer formattaki dosyalar

Text dosyaları için: Veri <- **read.table**(file="dosya_adi", header = TRUE)

CSV dosyaları için: Veri <- **read.table**(file="dosya_adi", header = TRUE, sep = ",")

CSV dosyaları için: Veri <- **read.csv**(file="dosya_adi", header = TRUE, sep = ",")

Excel dosyaları için:
library(XLConnect)
Veri <- **readWorksheetFromFile**("dosya_adi", sheet = 1)

Excel dosyaları için:
library(readxl)
Veri <- **read_excel**("dosya_adi")

XML dosyaları için:
library(xml)
Veri <- **xmlTreeParse**("XML veri dosyası>")

SPSS dosyaları için:
library(foreign)
Veri <- **read.spss**("dosya_adi.sav")

Systat dosyaları için:
library(foreign)
Veri <- **read.systat**("dosya_adi.sav")

Veri Tabanıyla Çalışmak

MySQL veri tabanı için:

```
library(“RMySQL”)
baglan <- dbConnect(MySQL(),
                    user=””,
                    password=””,
                    host=””,
                    dbname=””)
oku <- dbReadTable(baglan, tablo_adi)
sorgu <- dbGetQuery(baglan, “SELECT * FFROM tablo”)
veri <- fetch(sorgu)

close(baglan)
```

Veri Tabanıyla Çalışmak

Oracle veri tabanı için:

```
library(“"ROracle”)
drv<-dbDriver("Oracle")
baglan <- dbConnect(drv,
                    user=””,
                    password=””,
                    host=””,
                    dbname=””)
sorgu <- dbGetQuery(baglan, “SELECT * FFROM tablo”)
veri <- fetch(sorgu)

close(baglan)
```

Veri Tabanıyla Çalışmak

ODBC (Open Database Connectivity) veri tabanı arabirimi için:

```
library(“RODBC”)
drv<-dbDriver("Oracle")
baglan <- dbConnect(DSN adı,              ← ODBC Data Source Name
                    uid=””,
                    pwd=””)
sorgu <- sqlQuery(baglan, “SELECT * FFROM tablo”)
veri <- fetch(sorgu)

close(baglan)
```

Betimsel İstatistik ve Grafikler

Descriptive Statistics & Graphics

Dr. Halil Yurdugül yurdugul@hacettepe.edu.tr

Betimsel İstatistik

Betimsel istatistik; verilerin organize edilmesi, çeşitli katsayılar, tablo ve/veya grafiklerle özetlenmesi anlamına gelir.

```
> veri <- read.csv("/DosyaKonumu/Akciger.txt", header=TRUE, sep="\t")
> names(veri)                    → "hacim" "yas" "agirlik" "sigara" "cins" "dogum"
> dim(veri)                      → 725   6
> class(veri)                    → “data.frame”
> summary(veri)                  →
> pie(veri$cins)                 → Hata verecektir, bunu deneyin: pie(table(veri$cins))
> hist(veri$yas)
> hist(veri$yas, col="red")
> hist(veri$yas, break=12, col="red")
> hist(veri$yas, breaks=5, col="red", main="Yaş Frekans Dağılımı")
> hist(veri$yas, breaks=5, col="red", ylab="Sıklık", xlab="Yaş", main="Yaş Frekans")
```

Betimsel İstatistik

Betimsel istatistik; verilerin organize edilmesi, çeşitli katsayılar, tablo ve/veya grafiklerle özetlenmesi anlamına gelir.

```
> veri <- read.csv("/DosyaKonumu/Akciger.txt", header=TRUE, sep="\t")
> names(veri)                    → "hacim" "yas" "agirlik" "sigara" "cins" "dogum"
> dim(veri)                      → 725   6
> class(veri)                    → “data.frame”
> summary(veri)                  →
> pie(veri$cins)                 → Hata verecektir, bunu deneyin: pie(table(veri$cins))
> hist(veri$yas)
> plot(veri$yas)
> plot(density(veri$yas))
> plot(veri$yas, veri$agirlik)
> plot(veri)
```

Betimsel İstatistik

Betimsel istatistik; verilerin organize edilmesi, çeşitli katsayılar, tablo ve/veya grafiklerle özetlenmesi anlamına gelir.

```
> veri <- read.csv("/DosyaKonumu/Akciger.txt", header=TRUE, sep="\t")
> names(veri)                    → "hacim" "yas" "agirlik" "sigara" "cins" "dogum"
> dim(veri)                      → 725   6
> class(veri)                    → “data.frame”
> summary(veri)                  →
> pie(veri$cins)                 → Hata verecektir, bunu deneyin: pie(table(veri$cins))
> hist(veri$yas)
> plot(veri$yas)
> plot(density(veri$yas))
> plot(veri$yas, veri$agirlik)
> plot(veri)
```

65-89 slaytlar sunum dosyasından çıkartıldı.

R'da Dosya ve Dizin

Dosya ve Dizin İşlemleri

Dosya-Dizin İşlemleri

getwd() → Çalışılan dizini göster.
setwd(choose.dir()) → Dizin belirle.
setwd("/konum/dizin") → Changes the wd
dir() → Dizindeki dosyaları listele
dir.create("C:/test") → Kök dizinde ‘test’ dizini oluştur.
setwd("C:/test") → “c:/test” dizinini çalışan dizin yap.

İnternetten bir dosya indirilmesi.

```
download.file("http://xyz.com/veri.xls", “C:/veri.xls", method="auto",
              quiet=FALSE, mode = "wb",
              cacheOK = TRUE)
```

R'da Programlama

Yapısal Programlama
Karar ve Döngü Yapıları,
Kullanıcı Tanımlı Fonksiyonlar
Hata Kontrolleri ve Paket Yönetimi

Bu bölüm sunum dosyasından çıkartıldı.