

DogaBilimleriCevaplar

Ismail Bekar

10/26/2018

Alıştırma 2.1

1. `seq(1, 27, 0.5)` vektörünün 9 ve 12.sıradaki elementleri nelerdir?

```
data <- seq(1, 27, 0.5)
```

```
data[9]
```

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

```
data[12]
```

```
## [1] 6.5
```

2. 4 ve 34 arasındaki tüm çift sayıları içeren bir vektör yaratın ve bunu `a` objesine atayın. İpucu: `seq()` fonksiyonunu kullanın.

```
a <- seq(from = 4, to = 34, by=2)
```

3. `a` vektöründe 17'den büyük olan tüm elementleri seçerek çıkarın.

```
a[a > 17]
```

```
## [1] 18 20 22 24 26 28 30 32 34
```

Alıştırma 2.2

1. `TRUE` ve 50'yi içeren bir vektör oluşturun.

```
data1 <- c(TRUE, 50)
```

2. "A" ve 1 içeren bir vektör oluşturun.

```
data2 <- c("A", 1)
```

3. `TRUE` ve "C" içeren bir vektör oluşturun.

```
data3 <- c(TRUE, "C")
```

Alıştırma 2.3

1. İki tane 5'e 2lik matris oluşturun. Bir tanesi 1-20 arasındaki tüm çift sayıları içermeli, diğeri tüm tek sayıları. Matrisleri satırlara göre doldurun.

```
matriks1 <- matrix(seq(from = 1, to = 20, by = 2), nrow = 5, ncol = 2)
```

```
matriks2 <- matrix(seq(from = 2, to = 20, by = 2), nrow = 5, ncol = 2)
```

2. Bu iki matrisi satır ile birleştirin (`rbind`) ve yeni bir objeye atayın.

```
matriks3 <- rbind(matriks1, matriks2)
```

```
matriks3
```

```
##      [,1] [,2]
## [1,]    1  11
## [2,]    3  13
## [3,]    5  15
## [4,]    7  17
## [5,]    9  19
## [6,]    2  12
## [7,]    4  14
## [8,]    6  16
## [9,]    8  18
## [10,]   10  20
```

3. Bu iki matriksi sütun ile birleştirin (cbind) ve yeni bir objeye atayın.

```
matriks4 <- cbind(matriks1, matriks2)
matriks4
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1  11    2   12
## [2,]    3  13    4   14
## [3,]    5  15    6   16
## [4,]    7  17    8   18
## [5,]    9  19   10   20
```

Alıştırma 2.4

```
january <- matrix(1:31, nrow = 5, ncol = 7, byrow = T)
```

```
## Warning in matrix(1:31, nrow = 5, ncol = 7, byrow = T): data length [31] is
## not a sub-multiple or multiple of the number of rows [5]
```

```
colnames(january) <- c(
  "monday", "tuesday", "wednesday",
  "thursday", "friday", "saturday", "sunday")
rownames(january) <- c("week 1", "week 2", "week 3", "week 4", "week 5")
```

1. Yukarıdaki kodu çalıştırın.
2. January ayından Saturday ve Sunday sütunlarını silin.

```
january[,c(-6,-7)]
```

```
##      monday tuesday wednesday thursday friday
## week 1      1       2         3         4       5
## week 2      8       9        10        11      12
## week 3     15      16        17        18      19
## week 4     22      23        24        25      26
## week 5     29      30        31         1       2
```

3. 9, 10, ve 11. günlerde hasta oldunuz (geçmiş olsun). Bu günleri “hasta” olarak değiştirin. Matriksteki diğer girdiler bundan nasıl etkilendi?
4. Kalan günleri “turp gibi” olarak değiştirin

```
# Birinci çözüm
january[c(1,3:5),] <- "turp gibi"
january[2, c(1,5:7)] <- "turp gibi"
january[2, c(2:4)] <- "hasta"
january
```

```
##      monday      tuesday      wednesday      thursday      friday
## week 1 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 2 "turp gibi" "hasta"      "hasta"      "hasta"      "turp gibi"
## week 3 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 4 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 5 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
##      saturday      sunday
## week 1 "turp gibi" "turp gibi"
## week 2 "turp gibi" "turp gibi"
## week 3 "turp gibi" "turp gibi"
## week 4 "turp gibi" "turp gibi"
## week 5 "turp gibi" "turp gibi"
```

```
# İkinci çözüm
january[!january %in% c(9,10,11)] <- "turp gibi"
january[january %in% c(9,10,11)] <- "hasta"
january
```

```
##      monday      tuesday      wednesday      thursday      friday
## week 1 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 2 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 3 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 4 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 5 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
##      saturday      sunday
## week 1 "turp gibi" "turp gibi"
## week 2 "turp gibi" "turp gibi"
## week 3 "turp gibi" "turp gibi"
## week 4 "turp gibi" "turp gibi"
## week 5 "turp gibi" "turp gibi"
```

5. `length()` fonksiyonunu matris üzerinde kullanınca ne sonuç alıyorsunuz? Aynı matris üzerinde `nrow()` ve `ncol()` fonksiyonlarını kullanın. İlk sonuçla nasıl ilişkililer?

```
length(january)
```

```
## [1] 35
```

```
nrow(january)
```

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

```
ncol(january)
```

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

Alıştırma 3.1

Bu alıştırma için 2.kısımda oluşturduğunuzda `january` matrisini kullanın.

1. Tüm wednesdaylerin tarihlerini yazdırın.

```
january <- matrix(1:31, nrow = 5, ncol = 7, byrow = T)
colnames(january) <- c(
  "monday", "tuesday", "wednesday",
  "thursday", "friday", "saturday", "sunday")
rownames(january) <- c("week 1", "week 2", "week 3", "week 4", "week 5")
```

```
for (i in 11:15) {
  print(january[i])
}
```

2. Beşinci haftada yer alan February günlerini NA ile değiştirin.

```
for (i in 4:7) {
  january[5,i] <- NA
}
january
```

```
##          monday      tuesday    wednesday   thursday    friday
## week 1 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 2 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 3 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 4 "turp gibi" "turp gibi" "turp gibi" "turp gibi" "turp gibi"
## week 5 "turp gibi" "turp gibi" "turp gibi" NA          NA
##          saturday    sunday
## week 1 "turp gibi" "turp gibi"
## week 2 "turp gibi" "turp gibi"
## week 3 "turp gibi" "turp gibi"
## week 4 "turp gibi" "turp gibi"
## week 5 NA          NA
```

Alıştırma 3.2

1. Sizin sıranızda oturan insanların ismiyle bir vektör oluşturup, onlara bir loop ile küçük sürprizler yapın (güzel bir şeyler yazın).

```
insanlar <- c("Bilgecan", "Nurbahar")

for (i in 1:length(insanlar)) {
  g.insanlar <- paste("Çok güzel insan", insanlar[i], sep = " ")
  print(g.insanlar)
}

## [1] "Çok güzel insan Bilgecan"
## [1] "Çok güzel insan Nurbahar"
```

Alıştırma 3.3

```
x <- rnorm(n = 50, mean = 0, sd = 1)
m <- matrix(x, ncol = 5, nrow = 10)
```

- m matrisindeki her sütunun ortalamasını hesaplayan bir loop yazın ve sonucu a objesine atayarak saklayın.

```
a <- c()

for (i in 1:ncol(m)) {
  a[i] <- mean(m[,i])
  print(a)
}
```

```
## [1] 0.3435442
## [1] 0.3435442 0.4475100
## [1] 0.3435442 0.4475100 0.2517439
## [1] 0.3435442 0.4475100 0.2517439 0.2519245
## [1] 0.34354424 0.44750999 0.25174392 0.25192449 0.02304629
```

Alıştırma 3.4

- `x` isimli 1 ve 100 arasındaki tüm sayıları içeren bir vektör oluşturun.
- Vektör için `x[i] * 2` işlemini 100 kez (vektörün uzunluğu) gerçekleştirin. Fakat eğer `x[i]` 32'den büyük ve 50'den küçükse `x[i] * 3`'ü hesaplayın. Çıktıyı bir vektöre atayın.

```
x <- seq(1:100)
```

```
for (i in 1:length(x)) {
  if (x[i] > 32 & x[i] < 50) {
    a[i] <- x[i] * 3
  }
  if (x[i] < 33 | x[i] > 49) {
    a[i] <- x[i] * 2
  }
}
a
```

```
## [1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34
## [18] 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 99 102
## [35] 105 108 111 114 117 120 123 126 129 132 135 138 141 144 147 100 102
## [52] 104 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136
## [69] 138 140 142 144 146 148 150 152 154 156 158 160 162 164 166 168 170
## [86] 172 174 176 178 180 182 184 186 188 190 192 194 196 198 200
```

Alıştırma 3.5

- Kene istilasına uğramış beş farklı coğrafi alan var. Bilim insanları her bölgeden 10 geyik için örneklem alarak sayım yapıyor. Bunun gibi basit bir senaryoyu her bölgedeki beklenen ortalama kene sayısını bildiğimizi ve sayımı yapılan kenelerin poisson dağılımına uyduğunu varsayarak simüle edebiliriz.

```
tick_mean <- c(4, 32, 17, 10, 12)
tick_count <- matrix(nrow = 10, ncol = 5)
for (i in 1:5) {
  tick_count[, i] <- rpois(10, tick_mean[i])
}
```

- Yukarıdaki kodu çalıştırın.
- Yeni bir veriseti oluşturun ve bu verisetinde 10 veya 10'dan daha az kenesi olan hayvanlara 0, 10'dan daha fazla kenesi olanlara 1 atayın. Bunu loop kullanarak yapın. Daha sonra loop kullanmadan yapın.

```
# Dört farklı çözüm yöntemi için dört farklı veri
```

```
tick_bi <- tick_count
tick_bi2 <- tick_count
tick_bi3 <- tick_count
tick_bi4 <- tick_count
```

```
# Birinci çözüm
```

```

for (i in 1:length(tick_bi)) {
  if (tick_bi[i] > 10) {
    tick_bi[i] <- 1
  } else {
    tick_bi[i] <- 0
  }
}

# İkinci çözüm
for (i in 1:length(tick_bi2)) {
  if (tick_bi2[i] <= 10) {
    tick_bi2[i] <- 0
  }
  if (tick_bi2[i] > 10) {
    tick_bi2[i] <- 1
  }
}

# Üçüncü çözüm
for (i in 1:nrow(tick_bi3)) {
  for (h in 1:ncol(tick_bi3)) {
    if (tick_bi3[i,h] <= 10) {
      tick_bi3[i,h] <- 0
    }
    if (tick_bi3[i,h] > 10) {
      tick_bi3[i,h] <- 1
    }
  }
}

# Dördüncü çözüm
tick_bi4[tick_bi4 < 10] <- 0
tick_bi4[tick_bi4 >= 10] <- 1

```

Alıştırma 5.1

trees ve *mtcars* R'da yüklü olan diğer iki veri setidir.

1. Bu iki veri setinin kaç satıra sahip olduğuna bakın.

```
nrow(trees)
```

```
## [1] 31
```

```
nrow(mtcars)
```

```
## [1] 32
```

2. Kaç sütuna sahipler? Sütunların isimleri neler?

```
ncol(trees)
```

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

```
ncol(mtcars)
```

```
## [1] 11
```

3. `str()` fonksiyonunu kullanarak *iris* veri setinin kaç türe sahip olduğunu bulun.

```
str(iris)

## 'data.frame':    150 obs. of  5 variables:
## $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

4. İki veri setinde hangi class'lar mevcut?

```
str(trees)

## 'data.frame':    31 obs. of  3 variables:
## $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
## $ Height: num  70 65 63 72 81 83 66 75 80 75 ...
## $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```

```
str(mtcars)

## 'data.frame':    32 obs. of  11 variables:
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num  160 160 108 258 360 ...
## $ hp : num  110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt : num  2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num  16.5 17 18.6 19.4 17 ...
## $ vs : num  0 0 1 1 0 1 0 1 1 1 ...
## $ am : num  1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
str(iris)

## 'data.frame':    150 obs. of  5 variables:
## $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

Alıştırma 5.2

1. *iris* veri setinde `Sepal.Width` sütunundaki 9. elementi bulun?

```
iris[9, 2]
```

```
## [1] 2.9
```

```
iris[9,"Sepal.Width"]
```

```
## [1] 2.9
```

2. *iris* veri setindeki 17.satırı seçin.

```
iris[17,]
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 17           5.4           3.9           1.3           0.4  setosa
```

3. iris veri setinin 1, 4 ve 7. satırları ile bir obje oluşturun.

```
a <- iris[c(1,4,7),]
```

4. iris veri setindeki tüm tek sayılı satırları seçmek için seq() fonksiyonunu kullanın.

```
iris[seq(from= 1, to= 150, by= 2),]
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1           3.5           1.4           0.2  setosa
## 3           4.7           3.2           1.3           0.2  setosa
## 5           5.0           3.6           1.4           0.2  setosa
## 7           4.6           3.4           1.4           0.3  setosa
## 9           4.4           2.9           1.4           0.2  setosa
## 11          5.4           3.7           1.5           0.2  setosa
## 13          4.8           3.0           1.4           0.1  setosa
## 15          5.8           4.0           1.2           0.2  setosa
## 17          5.4           3.9           1.3           0.4  setosa
## 19          5.7           3.8           1.7           0.3  setosa
## 21          5.4           3.4           1.7           0.2  setosa
## 23          4.6           3.6           1.0           0.2  setosa
## 25          4.8           3.4           1.9           0.2  setosa
## 27          5.0           3.4           1.6           0.4  setosa
## 29          5.2           3.4           1.4           0.2  setosa
## 31          4.8           3.1           1.6           0.2  setosa
## 33          5.2           4.1           1.5           0.1  setosa
## 35          4.9           3.1           1.5           0.2  setosa
## 37          5.5           3.5           1.3           0.2  setosa
## 39          4.4           3.0           1.3           0.2  setosa
## 41          5.0           3.5           1.3           0.3  setosa
## 43          4.4           3.2           1.3           0.2  setosa
## 45          5.1           3.8           1.9           0.4  setosa
## 47          5.1           3.8           1.6           0.2  setosa
## 49          5.3           3.7           1.5           0.2  setosa
## 51          7.0           3.2           4.7           1.4 versicolor
## 53          6.9           3.1           4.9           1.5 versicolor
## 55          6.5           2.8           4.6           1.5 versicolor
## 57          6.3           3.3           4.7           1.6 versicolor
## 59          6.6           2.9           4.6           1.3 versicolor
## 61          5.0           2.0           3.5           1.0 versicolor
## 63          6.0           2.2           4.0           1.0 versicolor
## 65          5.6           2.9           3.6           1.3 versicolor
## 67          5.6           3.0           4.5           1.5 versicolor
## 69          6.2           2.2           4.5           1.5 versicolor
## 71          5.9           3.2           4.8           1.8 versicolor
## 73          6.3           2.5           4.9           1.5 versicolor
## 75          6.4           2.9           4.3           1.3 versicolor
## 77          6.8           2.8           4.8           1.4 versicolor
## 79          6.0           2.9           4.5           1.5 versicolor
## 81          5.5           2.4           3.8           1.1 versicolor
## 83          5.8           2.7           3.9           1.2 versicolor
```


## 85	5.4	3.0	4.5	1.5	versicolor
## 87	6.7	3.1	4.7	1.5	versicolor
## 89	5.6	3.0	4.1	1.3	versicolor
## 91	5.5	2.6	4.4	1.2	versicolor
## 93	5.8	2.6	4.0	1.2	versicolor
## 95	5.6	2.7	4.2	1.3	versicolor
## 97	5.7	2.9	4.2	1.3	versicolor
## 99	5.1	2.5	3.0	1.1	versicolor
## 101	6.3	3.3	6.0	2.5	virginica
## 103	7.1	3.0	5.9	2.1	virginica
## 105	6.5	3.0	5.8	2.2	virginica
## 107	4.9	2.5	4.5	1.7	virginica
## 109	6.7	2.5	5.8	1.8	virginica
## 111	6.5	3.2	5.1	2.0	virginica
## 113	6.8	3.0	5.5	2.1	virginica
## 115	5.8	2.8	5.1	2.4	virginica
## 117	6.5	3.0	5.5	1.8	virginica
## 119	7.7	2.6	6.9	2.3	virginica
## 121	6.9	3.2	5.7	2.3	virginica
## 123	7.7	2.8	6.7	2.0	virginica
## 125	6.7	3.3	5.7	2.1	virginica
## 127	6.2	2.8	4.8	1.8	virginica
## 129	6.4	2.8	5.6	2.1	virginica
## 131	7.4	2.8	6.1	1.9	virginica
## 133	6.4	2.8	5.6	2.2	virginica
## 135	6.1	2.6	5.6	1.4	virginica
## 137	6.3	3.4	5.6	2.4	virginica
## 139	6.0	3.0	4.8	1.8	virginica
## 141	6.7	3.1	5.6	2.4	virginica
## 143	5.8	2.7	5.1	1.9	virginica
## 145	6.7	3.3	5.7	2.5	virginica
## 147	6.3	2.5	5.0	1.9	virginica
## 149	6.2	3.4	5.4	2.3	virginica

5. iris veri setindeki Sepal.Width sütununu silin ve başka bir objeye atayın.

```
aa <- iris[, "Sepal.Width"]
iris[, "Sepal.Width"] <- NULL
```

6. iris veri setinde petal uzunluğu 3'den büyük olanları seçin.

```
iris[iris$Petal.Length > 3,]
```

##	Sepal.Length	Petal.Length	Petal.Width	Species
## 51	7.0	4.7	1.4	versicolor
## 52	6.4	4.5	1.5	versicolor
## 53	6.9	4.9	1.5	versicolor
## 54	5.5	4.0	1.3	versicolor
## 55	6.5	4.6	1.5	versicolor
## 56	5.7	4.5	1.3	versicolor
## 57	6.3	4.7	1.6	versicolor
## 58	4.9	3.3	1.0	versicolor
## 59	6.6	4.6	1.3	versicolor
## 60	5.2	3.9	1.4	versicolor
## 61	5.0	3.5	1.0	versicolor
## 62	5.9	4.2	1.5	versicolor

## 63	6.0	4.0	1.0 versicolor
## 64	6.1	4.7	1.4 versicolor
## 65	5.6	3.6	1.3 versicolor
## 66	6.7	4.4	1.4 versicolor
## 67	5.6	4.5	1.5 versicolor
## 68	5.8	4.1	1.0 versicolor
## 69	6.2	4.5	1.5 versicolor
## 70	5.6	3.9	1.1 versicolor
## 71	5.9	4.8	1.8 versicolor
## 72	6.1	4.0	1.3 versicolor
## 73	6.3	4.9	1.5 versicolor
## 74	6.1	4.7	1.2 versicolor
## 75	6.4	4.3	1.3 versicolor
## 76	6.6	4.4	1.4 versicolor
## 77	6.8	4.8	1.4 versicolor
## 78	6.7	5.0	1.7 versicolor
## 79	6.0	4.5	1.5 versicolor
## 80	5.7	3.5	1.0 versicolor
## 81	5.5	3.8	1.1 versicolor
## 82	5.5	3.7	1.0 versicolor
## 83	5.8	3.9	1.2 versicolor
## 84	6.0	5.1	1.6 versicolor
## 85	5.4	4.5	1.5 versicolor
## 86	6.0	4.5	1.6 versicolor
## 87	6.7	4.7	1.5 versicolor
## 88	6.3	4.4	1.3 versicolor
## 89	5.6	4.1	1.3 versicolor
## 90	5.5	4.0	1.3 versicolor
## 91	5.5	4.4	1.2 versicolor
## 92	6.1	4.6	1.4 versicolor
## 93	5.8	4.0	1.2 versicolor
## 94	5.0	3.3	1.0 versicolor
## 95	5.6	4.2	1.3 versicolor
## 96	5.7	4.2	1.2 versicolor
## 97	5.7	4.2	1.3 versicolor
## 98	6.2	4.3	1.3 versicolor
## 100	5.7	4.1	1.3 versicolor
## 101	6.3	6.0	2.5 virginica
## 102	5.8	5.1	1.9 virginica
## 103	7.1	5.9	2.1 virginica
## 104	6.3	5.6	1.8 virginica
## 105	6.5	5.8	2.2 virginica
## 106	7.6	6.6	2.1 virginica
## 107	4.9	4.5	1.7 virginica
## 108	7.3	6.3	1.8 virginica
## 109	6.7	5.8	1.8 virginica
## 110	7.2	6.1	2.5 virginica
## 111	6.5	5.1	2.0 virginica
## 112	6.4	5.3	1.9 virginica
## 113	6.8	5.5	2.1 virginica
## 114	5.7	5.0	2.0 virginica
## 115	5.8	5.1	2.4 virginica
## 116	6.4	5.3	2.3 virginica
## 117	6.5	5.5	1.8 virginica

```
## 118      7.7      6.7      2.2 virginica
## 119      7.7      6.9      2.3 virginica
## 120      6.0      5.0      1.5 virginica
## 121      6.9      5.7      2.3 virginica
## 122      5.6      4.9      2.0 virginica
## 123      7.7      6.7      2.0 virginica
## 124      6.3      4.9      1.8 virginica
## 125      6.7      5.7      2.1 virginica
## 126      7.2      6.0      1.8 virginica
## 127      6.2      4.8      1.8 virginica
## 128      6.1      4.9      1.8 virginica
## 129      6.4      5.6      2.1 virginica
## 130      7.2      5.8      1.6 virginica
## 131      7.4      6.1      1.9 virginica
## 132      7.9      6.4      2.0 virginica
## 133      6.4      5.6      2.2 virginica
## 134      6.3      5.1      1.5 virginica
## 135      6.1      5.6      1.4 virginica
## 136      7.7      6.1      2.3 virginica
## 137      6.3      5.6      2.4 virginica
## 138      6.4      5.5      1.8 virginica
## 139      6.0      4.8      1.8 virginica
## 140      6.9      5.4      2.1 virginica
## 141      6.7      5.6      2.4 virginica
## 142      6.9      5.1      2.3 virginica
## 143      5.8      5.1      1.9 virginica
## 144      6.8      5.9      2.3 virginica
## 145      6.7      5.7      2.5 virginica
## 146      6.7      5.2      2.3 virginica
## 147      6.3      5.0      1.9 virginica
## 148      6.5      5.2      2.0 virginica
## 149      6.2      5.4      2.3 virginica
## 150      5.9      5.1      1.8 virginica
```

7. Bir veri setinde `length()` fonksiyonunu kullandığınızda ne sonuç alıyorsunuz? Matriksten farkı ne?

```
length(iris)
```

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

Alıştırma 5.3

1. `c(1,2,3)` ve `c(4,5,6,7)` ile iki sütunlu bir veri seti oluşturun. Sonuç ne?

```
data.frame(x= c(1,2,3), y= c(4,5,6,7))
```

2. `c("a","b","c")` ve `c(1,2,3)` ile `data.frame` kullanarak bir veri seti oluşturun. Bir diğer veri setini önce `cbind`, sonra `data.frame` kullanarak oluşturun. Aradaki fark ne?

```
a <- data.frame(x= c("a","b","c"))
b <- data.frame(y= c(1,2,3))
aa <- data.frame(x= c("a","b","c"), y= c(1,2,3))
ab <- cbind(a,b)
str(aa)
```

```
## 'data.frame':   3 obs. of  2 variables:
```

```
str(ab)
```

1. See e.g. *United States v. Blue Hill*, 200 F.3d 1008 (1st Cir. 2000).

- ```
setwd("~/GitHub/DogaBilimleriR/Sunumlar")
bluehill <- read.csv("BlueHill.csv")
```

- ```
mean(bluehill$MNTM)
```

3. Dördüncü sütunu seçin ve bir objeye atayın. `Class`'ı ve `modu` nedir? İpucu: `class` ve `mode` fonksiyonlarını kullanın.

```
## [1] "integer"
```

```
## [1] "numeral"
```

- ```
b <- bluehill[1,]
class(b)
```

```
mode(b)
```

5. Üçüncü ve dördüncü alıştırmaları düşündüğünüzde veri setlerindeki seçim ile matrislerdeki seçim arasında nasıl farklar var?