

# R\_VeriAnalizi\_HAFTA\_3

October 21, 2020

## 1 R\_VeriAnalizi\_HAFTA\_2

```
[4]: #vektör oluşturmak için c() fonksiyonunu kullanıyoruz
```

```
[5]: vektor=c(10,20,30,40,50)
```

```
[7]: vektor
```

1. 10 2. 20 3. 30 4. 40 5. 50

```
[8]: #Bir vektör başka bir vektör içerisinde veri olarak kullanılabilir
```

```
[10]: vektor2=c(-1,-2,6,-9,7)
```

```
[11]: vektor3=c(8,54,62,vektor2)
```

```
[12]: vektor3
```

1. 8 2. 54 3. 62 4. -1 5. -2 6. 6 7. -9 8. 7

```
[13]: #Bir vektör sayısal,karakter veya mantıksal değerler içerebilir.  
#Ancak farklı türde bir vektörde yer alamaz.Eğer farklı türde  
#veri girilmeye çalışılırsa R bu verilerin tümünü karakter türü  
#olarak alır ve vektör içerisine katar
```

```
[15]: vektor4=c(61,3.14,"Sakarya",TRUE)
```

```
[16]: vektor4
```

1. '61' 2. '3.14' 3. 'Sakarya' 4. 'TRUE'

## 2 Vektör Elemanlarına Erişme İşlemi

```
[19]: #Vektör elemanlarına indis (indeks) değer ile erişilir.İndeks vektör  
#içerisindeki verinin hangi pozisyonunda olduğunu belirtir.  
#İndeks değeri []parantez içerisine yazılır. R'da indeks değerleri 1 den  
#başlar
```

```
[20]: vektor3
```

```
1. 8 2. 54 3. 62 4. -1 5. -2 6. 6 7. -9 8. 7
```

```
[26]: vektor3[8] #indeks değeri 8 olan veri elemanına erişim sağlar
```

```
7
```

```
[27]: #vektör içerisindeki birden çok elemana aynı anda erişmek için  
#aşağıdaki kod yazılmalıdır.
```

```
[28]: vektor3[2:4]
```

```
1. 54 2. 62 3. -1
```

### 3 Vektörler ile Aritmetik işlemler gerçekleştirme

```
[30]: x=c(1,3,5,7,9)  
      y=c(2,4,6,8,10)
```

```
[31]: toplam=x+y
```

```
[32]: toplam
```

```
1. 3 2. 7 3. 11 4. 15 5. 19
```

```
[33]: carpma = x*y
```

```
[34]: carpma
```

```
1. 2 2. 12 3. 30 4. 56 5. 90
```

```
[35]: bolme=x/y
```

```
[36]: bolme
```

```
1. 0.5 2. 0.75 3. 0.8333333333333333 4. 0.875 5. 0.9
```

```
[37]: cikartma=y-x
```

```
[38]: cikartma
```

```
1. 1 2. 1 3. 1 4. 1 5. 1
```

### 4 Mantıksal Vektör Oluşturma İşlemi

```
[40]: #Mantıksal vektörler, TRUE,FALSE,ve NA değerlerini alabilir.  
#Aynı zamanda T ve F kısaltmaları da kullanılabilir.
```

```
[41]: vektor5=c(TRUE,FALSE,TRUE,FALSE,NA,NA)
```

```
[42]: vektor5
```

1. TRUE 2. FALSE 3. TRUE 4. FALSE 5. <NA> 6. <NA>

```
[43]: #Vektörler içerisinde bir şarta bağlı işlemlerin sonucu mantıksal  
#çıktı olarak alınabilir.
```

```
vektor6=c(12,15,1,17,21,23,33,35)
```

```
[45]: vektor6 > 21
```

1. FALSE 2. FALSE 3. FALSE 4. FALSE 5. FALSE 6. TRUE 7. TRUE 8. TRUE

## 5 Karakter Tipinde Vektör Oluşturma İşlemi

```
[46]: #Vektörlerde karakter değerler "" içerisinde yazılır.
```

```
[48]: karakterVektor=c("Emrah","Semiz","Sakarya","istanbul")
```

```
[49]: karakterVektor
```

1. 'Emrah' 2. 'Semiz' 3. 'Sakarya' 4. 'istanbul'

```
[50]: #Vektörler başına ya da sonuna yeni eleman eklemek için aşağıdaki  
#işlem yapılır.
```

```
[54]: karakterVektor=c(karakterVektor,"fatih","sultan","mehmet")
```

```
[55]: karakterVektor
```

1. 'Emrah' 2. 'Semiz' 3. 'Sakarya' 4. 'istanbul' 5. 'fatih' 6. 'sultan' 7. 'mehmet'

```
[56]: #Vektör elemanlarının uzunluğunu öğrenmek için length() fonksiyonunu  
#kullanılır.
```

```
[59]: length(karakterVektor)
```

7

```
[60]: #Karakter vektörler içerisinde yer alan her bir elemanın karakter  
#uzunluklarını öğrenmek için nchar() fonksiyonu kullanılır.
```

```
[61]: nchar(karakterVektor)
```

1. 5 2. 5 3. 7 4. 8 5. 5 6. 6 7. 6

## 6 Vektörlerin Bileştirilmesi İşlemi

```
[62]: #İki vektörün elemanları tek bir vektör olarak birleştirmek için  
#c() fonksiyonu kullanılır.
```

```
[63]: x
```

1. 1 2. 3 3. 5 4. 7 5. 9

```
[64]: y
```

1. 2 2. 4 3. 6 4. 8 5. 10

```
[65]: z = c(x,y)
```

```
[66]: x
```

1. 1 2. 3 3. 5 4. 7 5. 9

```
[67]: z
```

1. 1 2. 3 3. 5 4. 7 5. 9 6. 2 7. 4 8. 6 9. 8 10. 10

## 7 Vektör Üyelerini İsimlendirme İşlemi

```
[68]: #Vektör elemanlarına isimlendirme yapılarak, mevcut elemanlara bu  
#adlandırma üzerinden erişilebilir. Bu işlem için names() fonksiyonu  
#kullanılır.
```

```
[69]: isimler = c("Emrah","Semiz","Ordu","Ünye","Sivas")
```

```
[70]: names(isimler)=c("ID10","ID20","ID30","ID40","ID50")
```

```
[71]: isimler
```

ID10 'Emrah' ID20 'Semiz' ID30 'Ordu' ID40 'Ünye' ID50 'Sivas'

## 8 Vektör Elemanlarının Güncellenmesi

```
[72]: #Güncelleme işleminde, değiştirilecek vektör elemanının indeks değeri  
#kullanılarak gerekli değişiklik yapılır.
```

```
[73]: isimler
```

ID10 'Emrah' ID20 'Semiz' ID30 'Ordu' ID40 'Ünye' ID50 'Sivas'

isimler[5]="Sivas/Zara"

```
[74]: isimler
```

ID10      'Emrah' ID20      'Semiz' ID30      'Ordu' ID40      'Ünye' ID50      'Sivas'

```
[75]: isimler[5] = "Zara/Sivas"
```

```
[76]: isimler
```

ID10      'Emrah' ID20      'Semiz' ID30      'Ordu' ID40      'Ünye' ID50      'Zara/Sivas'

```
[78]: #Vektör Elemanlarını Silme İşlemi  
#bu işlem için [-indeks numarası] biçimi kullanılır. negatif deger olduğu  
#için bir değişkene atamadıkca sadece bellekten silinir. Vektörden silinmez
```

```
[79]: isimler[-4]
```

ID10      'Emrah' ID20      'Semiz' ID30      'Ordu' ID50      'Zara/Sivas'

```
[80]: isimler
```

ID10      'Emrah' ID20      'Semiz' ID30      'Ordu' ID40      'Ünye' ID50      'Zara/Sivas'

```
[81]: isimler = isimler[-4]
```

```
[82]: isimler
```

ID10      'Emrah' ID20      'Semiz' ID30      'Ordu' ID50      'Zara/Sivas'

```
[83]: #Belli bir indeks aralığındaki vektör elemanlarını silmek için  
#vektor_adi[-c(indeks numarası:indeksnumarası)] biçimi kullanılır
```

```
[84]: yeniIsimler = c("Emrah", "Şeyma", "Semiz", "Ordu", "Sivas")
```

```
[87]: silinenIsimler = yeniIsimler[-c(2:4)]
```

```
[88]: yeniIsimler
```

1. 'Emrah' 2. 'Şeyma' 3. 'Semiz' 4. 'Ordu' 5. 'Sivas'

```
[89]: silinenIsimler
```

1. 'Emrah' 2. 'Sivas'

```
[90]: #Tüm verileri silmek için NULL parametresi kullanılır.
```

```
[91]: yeniIsimler=NULL
```

```
[92]: yeniIsimler
```

NULL

## 9 Vektör Elemanlarının Sıralanması

```
[94]: #Bu işlem için order() ve sort() fonksiyonların kullanılır.
```

```
[96]: adlar=c("Zehra","Ayşe","Mehmet Ziya","Alperen","Özge")
```

```
[97]: adlar
```

1. 'Zehra' 2. 'Ayşe' 3. 'Mehmet Ziya' 4. 'Alperen' 5. 'Özge'

```
[98]: sort(adlar) #A-Z ye sıralama yapar
```

1. 'Alperen' 2. 'Ayşe' 3. 'Mehmet Ziya' 4. 'Özge' 5. 'Zehra'

```
[100]: #Z-A ya (büyükten küçüğe doğru) sıralama işlemi için decreasing=FALSE  
#fonksiyonu kullanılır. Default olarak A-Z sıralama yapar
```

```
[102]: sort(adlar,decreasing = TRUE)
```

1. 'Zehra' 2. 'Özge' 3. 'Mehmet Ziya' 4. 'Ayşe' 5. 'Alperen'

```
[103]: order(adlar,decreasing = TRUE) #indis numarasını verir ve indir numarasını  
#sıralar
```

1. 1 2. 5 3. 3 4. 2 5. 4

## 10 Faktör Kavramı

```
[107]: #Faktörler sayısal ya da karakteristik değerler içeren değişkenler  
#olarak adlandırılır.Faktörlerde düzeyssel yapı vardır.  
#Faktör oluşturmak için factor() fonksiyonu kullanılır.  
#Faktör oluşturmak için ilk olarak vektör oluşturulur sonra  
#vektöre faktöre dönüştürülür
```

```
[108]: faktor=factor(c("orta","yüksek","orta","düşük","orta","yüksek","düşük"))
```

```
[109]: faktor
```

1. orta 2. yüksek 3. orta 4. düşük 5. orta 6. yüksek 7. düşük

Levels: 1. 'düşük' 2. 'orta' 3. 'yüksek'

```
[110]: #Faktöre Elemanlarına Erişim ve Güncelleme Yapma İşlemi
```

```
[111]: #Değişiklik yapma işlemi
```

```
[113]: faktor[3]="Çok Yüksek" #level içerisinde hangi değer var ise o değer  
#verilebilir. Level içinde yer alanlar dışında verilemez
```

```
Warning message in `[<-.factor`(`*tmp*`, 3, value = "Çok Yüksek"):  
"invalid factor level, NA generated"
```

```
[114]: faktor[3]="düşük"
```

```
[115]: faktor
```

1. orta 2. yüksek 3. düşük 4. düşük 5. orta 6. yüksek 7. düşük

Levels: 1. 'düşük' 2. 'orta' 3. 'yüksek'

```
[116]: #Silme İşlemi
```

```
[120]: yeniFaktor = faktor[-4]
```

```
[121]: yeniFaktor
```

1. orta 2. yüksek 3. düşük 4. orta 5. yüksek 6. düşük

Levels: 1. 'düşük' 2. 'orta' 3. 'yüksek'

```
[ ]:
```