



Kosullu Onermeler



Dr. Kübra Atalay Kabasakal
Bahar 2023

Koşullu Önermeler

R'da fonksiyon yazarken sıklıkla kullanılan koşullu önermeler, mantıksal (Boolean) operatörler

- (`TRUE` ve `FALSE`)
- `if()`
- `else()`
- `else if()`
- `ifelse()`
- `while()`
- `switch()` fonksiyonlarından oluşur.

if() durum cümlesi

Hemen hemen bütün programlama dillerinde ortak olarak kullanılan `if()` durum cümlesi, koşullu bir ifadeyi bir ölçüt ile karşılaştırarak test eder ve koşullu ifadenin gerçekleşmesi (TRUE) durumunda elde edilecek çıktıları belirlemede kullanılır. `if()` durum cümlesinin genel kullanımı aşağıdaki gibidir.

```
if(koşul){  
    print(durum cümlesi)  
}
```

- `{}` parantez kullanimini dikkat etmek gerekir.
- `if()` ifade `koşul` `()` parantez içine yazılır.
- `if(koşul){}` koşul yazıldıktan sonra mutlaka `{}` eklenmeli

if() durum cümlesi

Ölçme açısından bakılacak olursa **koşul** bir **ölçütü**, **durum** cümlesi ise **değerlendirmeyi** gösterilebilir.

Örneğin, yapılan bir sınavda geçme notu 60 olarak belirlendiğinde, 75 alan bir öğrencinin durumu aşağıdaki **if()** durum cümlesiyle belirlenebilmektedir.

```
x <- 75
if(x>=65){
  print("Başarılı")
}
```

```
## [1] "Başarılı"
```

if() durum cümlesi

Ancak kontrol durumu çoğunlukla tek önermeye bağlı değildir.

bu kod çıktı vermeyecektir

```
x <- 60
if(x>=65){
  print("Başarılı")
}
```

else() durum cümlesi

`else()` durum cümlesi `if()` durum cümlesindeki koşullar sağlanmadığı durumda ortaya çıkacak alternatifi belirlemek için kullanılır.

```
if(koşul1){  
  print(durum cümlesi)  
}else{  
  print(durum cümlesi)  
}
```

```
x <- 75 # Başarılı Durum  
if(x>=65){  
  print("Başarılı")  
}else{  
  print("Başarısız")  
}
```

```
## [1] "Başarılı"
```

else if () durum cümlesi

Koşul her zaman iki kategori ile tanımlanamayabilir. Bu durumda kullanımı `else if()` ile destekleyebiliriz.

```
if(koşul1){  
  print(durum cümlesi)  
}else if(koşul2)  
  print(durum cümlesi)  
}else{  
  print(durum cümlesi)  
}
```

```
x <- 75 # Başarılı Durum  
if(x>=90){  
  print("AA")  
}else if(x>=80){  
  print("BA")  
}else if(x>=70){  
  print("BB")  
}else if(x>=65){  
  print("CB")  
}else if(x>=60){  
  print("CC")  
}else if(x>=50){  
  print("DD")  
}else if(x>=30){  
  print("FD")  
}else{  
  print("FF")  
}
```

```
## [1] "BB"
```

ORNEKLER

a sayinin carpmaya gore tersi $1/a$ 'dir. Ancak bu durum 0 icin tanimsizdir. `if()` durum cümlesi kullanarak bu durumu kodlayınız.

`x <- 5` için test ediniz.

```
## [1] "5'in carpmaya gore tersi 1/5"
```

`x <- 0` için test ediniz.

```
## [1] "1/0 tanımsızdır."
```


ORNEKLER

a sayinin carpmaya gore tersi $1/a$ 'dir. ancak bu durum 0 icin tanimsizdir. `if()` durum cümlesi kullanarak bu durumu `x <- c(1,2,3,0,-4)` için deneyiniz.

```
## [1] "1'in carpmaya gore tersi 1/1"  
## [2] "2'in carpmaya gore tersi 1/2"  
## [3] "3'in carpmaya gore tersi 1/3"  
## [4] "1/0 tanımsızdır."  
## [5] "-4'in carpmaya gore tersi 1/-4"
```

ORNEKLER

-2 ile 2 arasında sayılar üretip, bunu x değişkenine atayalım.

```
x <- rnorm(1)  
x
```

```
## [1] 2.137767
```

Random olarak üretilen sayının 1'den büyük olması durumunda çıktı "1'den büyük" -1 ile 1 arasında olması durumunda "-1 ile +1 arasında" -1'den küçük olması durumunda ise "-1'den küçük" çıktısı versin.

```
## [1] 0.5058193
```

```
## [1] "sayı -1 ile +1 arasında"
```

if() & all()

Her ne kadar `if()` önermesi bir elemanlı vektorlerde çıktı verirken `if()` önermesi içinde kullanılabilen `all` fonksiyonu ile vektorün tüm elemanları için koşul test edilebilir.

```
x <- c(1,2,-3,4)
if(all(x>0)){
  print("tum sayilar 0'dan buyuktur")
} else{
  print("tum sayilar 0'dan buyuk degildir")
}
```

```
## [1] "tum sayilar 0'dan buyuk degildir"
```

if() & any()

Bir vektörde içinde yer alan her hangi bir elemana dair test ise `if()` fonksiyonu içinde `any()` fonksiyonu ile sağlanabilir.

```
x <- c(1,2,-3,4)
if(any(x<0)){

  print("nesne en az bir negatif sayi icerir")

} else{

  print("nesne negatif sayi icermez")

}
```

```
## [1] "nesne en az bir negatif sayi icerir"
```

if() çoklu işlem

```
x <- 2
if(x == 2) {
  goster3 <- "Dogru"
  goster3b <- c(1,2,3)
  goster3c <- sample(1:1000,4)
} else {
  goster3 <- "Yanlis"
  goster3b <- c(3,2,1)
  goster3c <- 10000 + sample(1:1000,4)
}
goster3
```

```
## [1] "Dogru"
```

```
goster3b
```

```
## [1] 1 2 3
```

```
goster3c
```

```
## [1] 192 955 761 847
```

ifelse()

`ifelse()` durum cümlesi, `if()` durum cümlelerinde vektörlerin kullanımından kaynaklı sıkıntılara çözüm sunar. Bu bakımdan `ifelse()`, `if()` durum cümlelerinin vektörler için kullanılabilir halidir.

`ifelse()` durum cümlesinin genel kullanımı aşağıdaki gibidir.

`ifelse(koşul, Doğru İfade, Yanlış İfade)`

```
x <- 20  
ifelse(x>=65, "Başarılı", "Başarısız")
```

```
## [1] "Başarısız"
```

ifelse()

Eksik verinin 99 ile gösterildiği bir vektörde eksik veri yerine NA atama

```
(x <- c(1,2,3,4,99,5))
```

```
## [1] 1 2 3 4 99 5
```

```
ifelse(x==99, NA, x)
```

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

ifelse()

Elimizdeki bir nesnede yer alan sayıların tek ya da çift olduğunu yazdırma

```
set.seed(41)
sayilar <- sample(50:90,27)
sayilar
```

```
## [1] 89 84 54 81 57 78 55 71 80 62 87 67 70 83 82 61 66 53 50 69
## [21] 79 64 85 51 73 74 88
```

Çıktıyı sağlamaya çalışın

```
## [1] "Tek Sayi" "Cift Sayi" "Cift Sayi" "Tek Sayi" "Tek Sayi"
## [6] "Cift Sayi" "Tek Sayi" "Tek Sayi" "Cift Sayi" "Cift Sayi"
## [11] "Tek Sayi" "Tek Sayi" "Cift Sayi" "Tek Sayi" "Cift Sayi"
## [16] "Tek Sayi" "Cift Sayi" "Tek Sayi" "Cift Sayi" "Tek Sayi"
## [21] "Tek Sayi" "Cift Sayi" "Tek Sayi" "Tek Sayi" "Tek Sayi"
## [26] "Cift Sayi" "Cift Sayi"
```


ifelse()

Elimizdeki bir nesnede yer alan sayıların 0, pozitif ya negatif olduğu belirleme

```
set.seed(987)
sayilar <- sample(-10:10,27,replace=TRUE)
sayilar
```

```
## [1] 4 3 4 2 1 7 -10 5 6 -8 7 -3 9 7 -9
## [16] 10 4 -1 -8 8 -3 0 4 5 8 1 3
```

Çıktıyı sağlamaya çalışın

```
## [1] "Pozitif" "Pozitif" "Pozitif" "Pozitif" "Pozitif" "Pozitif" "Pozitif"
## [7] "Negatif" "Pozitif" "Pozitif" "Negatif" "Pozitif" "Negatif"
## [13] "Pozitif" "Pozitif" "Negatif" "Pozitif" "Pozitif" "Negatif"
## [19] "Negatif" "Pozitif" "Negatif" "Sıfır" "Pozitif" "Pozitif"
## [25] "Pozitif" "Pozitif" "Pozitif"
```

ORNEKLER

finalden 50 ve üzeri alan ve en az 11 derse devam eden öğrencilerin geçme notları finalin %60 ve vizenin %40 alınarak hesaplanır, 11'den az derse devam eden öğrencilerin geçme notu final notunun %60' olarak alınır

```
vize <- c(60,70,80,90,55)
final <- c(45,65,70,50,80)
devam <- c(14,10,13,12,11)
```