

R Programlamaya Giriş — Listeler: Alıştırmalar

R Programlama ve İstatistik Akademisi

Dr. M. Fatih Tüzen

1. Liste Temelleri

Görev 1.1 — Basit liste oluşturma

Aşağıdaki bilgileri içeren isimli bir liste oluşturun:

- `ad = "Deniz"` (karakter)
- `yas = 24` (numeric)
- `notlar = c(78, 82, 91)` (numeric vektör)
- `mezun = TRUE` (logical)

Görev 1.2 — Yapıyı inceleme

Oluşturduğunuz listenin: - sınıfını (`class()`) - eleman sayısını (`length()`) - yapısını (`str()`) - eleman adlarını (`names()`) inceleyin.

2. Erişim Yöntemleri: `[]`, `[[]]`, `$`

Görev 2.1 — Alt liste vs doğrudan eleman

1. `[]` ile `ad` ögesini çekin ve dönen tipin ne olduğuna bakın.
2. `[[]]` ile `ad` ögesini çekin ve tipini kontrol edin.
3. `$` ile `yas` ögesine erişin.

Görev 2.2 — Hata yakalama

Var olmayan bir elemana erişmeye çalışın (ör. `kan_grubu`). `is.null()` kullanarak güvenli kontrol yapın ve uygun bir mesaj basın.

3. Güncelleme, Ekleme, Silme

Görev 3.1 — Güncelleme

`yas` değerini 25 yapın, yeni bir `universite = "Hacettepe"` ögesi ekleyin.

Görev 3.2 — Silme

`mezun` ögesini listeden kaldırın. (İpucu: `NULL` atayın.) Sonraki `str()` çıktısını yorumlayın.

4. İç İçe Listeler

Görev 4.1 — Sınıf listesi

Üç öğrenciden (ogr1, ogr2, ogr3) oluşan bir **sinif** listesi oluşturun. Her öğrenci kendi içinde en az **ad**, **yas**, **notlar** alanlarına sahip bir **liste** olsun.

- ogr2 için **adres** = `list(il = "İzmir", ilce = "Karşıyaka")` alanını da ekleyin.

- ogr3'ün **notlar** vektörünün ortalamasını hesaplayın.

Görev 4.2 — Derin erişim

`sinif$ogr2$adres$ilce` değerini çekin, ardından `[[]]` zinciriyle aynı sonuca ulaşın.

5. Liste-Fonksiyon Etkileşimi

Görev 5.1 — `lapply()` ve `sapply()`

Aşağıdaki listeyi oluşturun ve elemanların ortalamasını hem `lapply()` hem `sapply()` ile hesaplayın. Çıktı tiplerini karşılaştırın.

```
sayilar <- list(a = 1:5, b = 6:10, c = c(10, 20, 30, 40))
```

Görev 5.2 — NA'lı değerlerle çalışma

`set.seed(42)` ile aynı olacak şekilde aşağıdaki listeyi üretin ve her bir elemanın standart sapmasını `na.rm = TRUE` parametresiyle hesaplayın.

```
set.seed(42)
sayilar2 <- list(
  x = c(1, 2, 3, NA),
  y = c(10, 12, NA, 14),
  z = rnorm(5)
)
```

Görev 5.3 — Çoklu özet

`sayilar2` listesinin her elemanı için `mean` ve `sd` içeren **alt listeler** döndüren bir `lapply()` ifadesi yazın.

Görev 5.4 — Matrisleştirme

`sayilar2` için `range(., na.rm=TRUE)` fonksiyonunu `sapply()` ile uygulayın ve dönen yapıyı yorumlayın.

6. Model Çıktıları Liste Olarak: `lm()`

Görev 6.1 — `lm()` nesnesinin yapısı

Basit bir veri çerçevesi oluşturun, `lm(y ~ x)` kurun.

- `class(model)` ve `str(model)` ile inceleyin.
- `model$coefficients`, `model$residuals`, `model$fitted.values` alanlarını ayrı ayrı çekin.
- `summary(model)`'ün sınıfına bakın ve R^2 ile `coefficients` tablosunu çekin.

Görev 6.2 — Tahmin

Kurduğunuz modelle `x = c(6, 7)` için tahmin üretin.

7. Ek Uygulama: İsimleri Düzenlemek

Görev 7.1 — `names()` ile yeniden adlandırma

`sinif` içindeki tüm öğrencilerin `ad` alanını `isim` olarak yeniden adlandırın. (İpucu: döngü veya `lapply()` kullanabilirsiniz.)

Görev 7.2 — `unlist()` riskleri

Seçtiğiniz küçük bir alt listede `unlist()` uygulayın ve tip dönüşümlerini gözlemleyin. Yorumlayın: Hangi durumlarda `unlist()` tehlikeli olabilir?