R Scripting

Lab for unit 4 - Structured programming

Marcus Wurzer

04. Oktober 2022

Please solve the following problems!

1. The `CO2` dataset (included in R) contains information about the uptake of carbon dioxide by different types of plants exposed to different treatments.

   a. Suppose we wish to subtract the mean value of the `uptake` variable in the `CO2` data frame, where the mean is calculated separately for each `Type`/`Treatment` combination using `tapply()`. Save these means in an object called `means`.
   b. Next, we need an index vector that specifies which mean has to be subtracted from which observation. Use the *same* (identical) call to `tapply()` you used for the computation of the group means above, but without specifying a function to be applied. Save the resulting vector as `idx`.
   c. Compute the group mean-adjusted CO2 uptake for each plant.
   d. We used two `tapply()` calls to reach our goal, but this can be simplified using the `ave()` function that combines the two operations instead (i.e., `ave()` replaces `tapply()`). Change the code accordingly and check the result.

```r
# a.
means <- tapply(CO2$uptake, CO2[c("Type", "Treatment")], mean)
means
```

```
##              Treatment
## Type          nonchilled  chilled
##   Quebec        35.33333 31.75238
##   Mississippi   25.95238 15.81429
```

```r
# b.
idx <- tapply(CO2$uptake, CO2[c("Type", "Treatment")])
idx
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [39] 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4
## [77] 4 4 4 4 4 4 4 4
```

```r
# c.
adj.uptake <- CO2$uptake - means[idx]
adj.uptake
```

```
##  [1] -19.33333333  -4.93333333  -0.53333333   1.86666667  -0.03333333
##  [6]   3.86666667   4.36666667 -21.73333333  -8.03333333   1.76666667
## [11]   6.46666667   5.26666667   6.06666667   8.96666667 -19.13333333
## [16]  -2.93333333   4.96666667   6.76666667   7.56666667   8.56666667
## [21]  10.16666667 -17.55238095  -7.65238095  -1.45238095   2.84761905
## [26]   0.74761905   3.64761905   6.94761905 -22.45238095  -4.45238095
## [31]   3.24761905   7.04761905   6.84761905   5.74761905  10.64761905
## [36] -16.65238095 -10.75238095   6.34761905   2.24761905   7.14761905
## [41]   7.84761905   9.64761905 -15.35238095  -6.75238095   0.24761905
## [46]   4.04761905   4.94761905   6.44761905   9.54761905 -13.95238095
## [51]  -3.95238095   4.64761905   5.84761905   6.44761905   5.14761905
## [56]   5.54761905 -14.65238095  -6.55238095  -0.15238095   1.94761905
## [61]   2.54761905   2.14761905   1.84761905  -5.31428571  -0.91428571
## [66]   2.28571429   3.08571429   3.68571429   6.38571429   6.08571429
## [71]  -8.11428571  -4.41428571  -3.51428571  -2.81428571  -3.31428571
## [76]  -2.11428571  -1.41428571  -5.21428571   2.18571429   2.08571429
## [81]   2.08571429   2.08571429   3.08571429   4.08571429
```

```r
# d.
adj.uptake <- CO2$uptake - ave(CO2$uptake, CO2$Type, CO2$Treatment, FUN = mean)
adj.uptake
```

```
##  [1] -19.33333333  -4.93333333  -0.53333333   1.86666667  -0.03333333
##  [6]   3.86666667   4.36666667 -21.73333333  -8.03333333   1.76666667
## [11]   6.46666667   5.26666667   6.06666667   8.96666667 -19.13333333
## [16]  -2.93333333   4.96666667   6.76666667   7.56666667   8.56666667
## [21]  10.16666667 -17.55238095  -7.65238095  -1.45238095   2.84761905
## [26]   0.74761905   3.64761905   6.94761905 -22.45238095  -4.45238095
## [31]   3.24761905   7.04761905   6.84761905   5.74761905  10.64761905
## [36] -16.65238095 -10.75238095   6.34761905   2.24761905   7.14761905
## [41]   7.84761905   9.64761905 -15.35238095  -6.75238095   0.24761905
## [46]   4.04761905   4.94761905   6.44761905   9.54761905 -13.95238095
## [51]  -3.95238095   4.64761905   5.84761905   6.44761905   5.14761905
## [56]   5.54761905 -14.65238095  -6.55238095  -0.15238095   1.94761905
## [61]   2.54761905   2.14761905   1.84761905  -5.31428571  -0.91428571
## [66]   2.28571429   3.08571429   3.68571429   6.38571429   6.08571429
## [71]  -8.11428571  -4.41428571  -3.51428571  -2.81428571  -3.31428571
## [76]  -2.11428571  -1.41428571  -5.21428571   2.18571429   2.08571429
## [81]   2.08571429   2.08571429   3.08571429   4.08571429
```