Hand crafting model matrix

Gan Yao

2022-12-15

This analysis attempts to manually create model matrix as suggested in TR658 by Shaw et al. However, as it turns out, this is not necessary at least for this `echin2` data and our Lilium data, as we can use R formula to generate the exact same model matrix.

Reproduction of `echin2` analysis in TR658

```
library(aster)
library(tidyverse)
```

```
data(echin2)
```

```
vars <- c("lds1", "lds2", "lds3", "ld01", "ld02",
  "ld03", "roct2003", "ld04", "roct2004", "ld05",
  "roct2005")
pred <- c(0, 1, 2, 3, 4, 5, 6, 6, 8, 8, 10)
fam <- c(1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 3)
nind <- length(unique(echin2$id))
nnode <- length(levels(echin2$varb))
```

```
x <- echin2$resp
dim(x) <- c(nind, nnode)
r <- 0 * x + 1
```

Hand-crafted model matrix

```
modmat.super <- NULL
names.super <- NULL
for (i in levels(echin2$varb)) {
 modmat.super <- cbind(modmat.super, as.numeric(echin2$varb ==i))
 names.super <- c(names.super, i)
 }
```

```
in.greenhouse <- is.element(echin2$varb, grep("lds", levels(echin2$varb), value = TRUE))
print(unique(echin2$varb[in.greenhouse]), max.levels = 0)
```

```
## [1] lds1 lds2 lds3
```

```
print(unique(echin2$varb[!in.greenhouse]), max.levels = 0)
```

```
## [1] ld01     ld02     ld03     roct2003 ld04     roct2004 ld05     roct2005
```

```
for (i in levels(echin2$flat)) if (i > "1") {
 modmat.super <- cbind(modmat.super, as.numeric(in.greenhouse &
 echin2$flat == i))
```

```
 names.super <- c(names.super, paste("flat", i,
 sep = ""))
 }
```

```
for (i in levels(echin2$row)) if (i > "10") {
 modmat.super <- cbind(modmat.super, as.numeric((!in.greenhouse) &
 echin2$row == i))
 names.super <- c(names.super, paste("row", i,
 sep = ""))
 }
```

```
for (i in levels(echin2$yearcross)) if (i >= "2000") {
 modmat.super <- cbind(modmat.super, as.numeric(echin2$yearcross ==i))
 names.super <- c(names.super, paste("yc", i, sep = ""))
 }
```

```
modmat.super <- cbind(modmat.super, as.numeric(!in.greenhouse) *  echin2$posi)
names.super <- c(names.super, "posi")
```

```
for (i in levels(echin2$crosstype)) if (i > "W") {
 modmat.super <- cbind(modmat.super, as.numeric(echin2$crosstype == i & echin2$varb == "roct2005"))
 names.super <- c(names.super, paste("cross", i, sep = ""))
 }
```

```
for (i in levels(echin2$crosstype)) if (i > "W") {
 modmat.super <- cbind(modmat.super, as.numeric(echin2$crosstype == i & echin2$varb == "lds3"))
 names.super <- c(names.super, paste("crossgreen", i, sep = ""))
 }
```

```
nodename <- unique(as.character(echin2$var))
modmat.super <- array(as.vector(modmat.super), c(dim(x), length(names.super)))
dimnames(modmat.super) <- list(NULL, nodename, names.super)
```

Model fitted with handcrafted model matrix

```
out.super <- aster(x, r, pred, fam, modmat.super)
summary(out.super)
```

```
##
## Call:
## NULL
##
##              Estimate Std. Error z value Pr(>|z|)
## ld01          0.77412    0.27150   2.851 0.004355 **
## ld02          0.95432    0.24678   3.867 0.000110 ***
## ld03          2.32493    0.37898   6.135 8.53e-10 ***
## ld04          2.22592    0.39099   5.693 1.25e-08 ***
## ld05          3.65585    0.27983  13.064  < 2e-16 ***
## lds1         -0.57400    0.40929  -1.402 0.160787
## lds2          0.89143    0.49525   1.800 0.071867 .
## lds3          2.04511    0.44247   4.622 3.80e-06 ***
## roct2003     -2.26195    0.20461 -11.055  < 2e-16 ***
## roct2004     -1.15350    0.11817  -9.761  < 2e-16 ***
## roct2005     -0.29824    0.08701  -3.428 0.000609 ***
## flat2        -0.25165    0.13647  -1.844 0.065188 .
```

```
## flat3         0.25957    0.18745   1.385 0.166122
## row11         0.14635    0.03641   4.019 5.85e-05 ***
## row12         0.12414    0.03545   3.502 0.000462 ***
## row13         0.12050    0.03470   3.472 0.000516 ***
## yc2000       -0.02756    0.02940  -0.937 0.348565
## posi         -0.32282    0.07003  -4.609 4.04e-06 ***
## crossWi      -0.58086    0.15305  -3.795 0.000148 ***
## crossWr      -0.08069    0.14552  -0.554 0.579272
## crossgreenWi -0.52844    0.35997  -1.468 0.142100
## crossgreenWr -0.29471    0.49249  -0.598 0.549563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model matrix by formula

By creating necessary variables and applying R formula, we can get the exact same model.

```
resp <- echin2$resp
varb <- echin2$varb
inGreenHouse <- as.numeric(in.greenhouse)
outGreenHouse <- as.numeric(!in.greenhouse)
fitField <- as.numeric(echin2$varb == 'roct2005')
fitChamber <- as.numeric(echin2$varb == 'lds3')
yearcross <- echin2$yearcross
flat2 <- as.numeric(echin2$flat == 2)
flat3 <- as.numeric(echin2$flat == 3)
row11 <- as.numeric(echin2$row == 11)
row12 <- as.numeric(echin2$row == 12)
row13 <- as.numeric(echin2$row == 13)
crossWi <- as.numeric(echin2$crosstype == 'Wi')
crossWr <- as.numeric(echin2$crosstype == 'Wr')
posi <- echin2$posi
```

```
out.super.test_2 <- aster(resp ~ -1 + varb + yearcross +inGreenHouse:flat2 + inGreenHouse:flat3
                        + outGreenHouse:row11 + outGreenHouse:row12 + outGreenHouse:row13 + outGreenHous
                        + fitField:crossWi + fitField:crossWr + fitChamber:crossWi + fitChamber:crossWr
                        pred, fam, varb, echin2$id, echin2$root)
```

```
sum(is.na(echin2))
```

```
## [1] 0
```

```
summary(out.super.test_2)
```

```
##
## Call:
## aster.formula(formula = resp ~ -1 + varb + yearcross + inGreenHouse:flat2 +
##     inGreenHouse:flat3 + outGreenHouse:row11 + outGreenHouse:row12 +
##     outGreenHouse:row13 + outGreenHouse:posi + fitField:crossWi +
##     fitField:crossWr + fitChamber:crossWi + fitChamber:crossWr,
##     pred = pred, fam = fam, varvar = varb, idvar = echin2$id,
##     root = echin2$root)
##
##                     Estimate Std. Error z value Pr(>|z|)
## varbld01             0.77412    0.27150   2.851 0.004355 **
```

```
## varbld02             0.95432    0.24678   3.867 0.000110 ***
## varbld03             2.32493    0.37898   6.135 8.53e-10 ***
## varbld04             2.22592    0.39099   5.693 1.25e-08 ***
## varbld05             3.65585    0.27983  13.064  < 2e-16 ***
## varblds1            -0.57400    0.40929  -1.402 0.160787
## varblds2             0.89143    0.49525   1.800 0.071867 .
## varblds3             2.04511    0.44247   4.622 3.80e-06 ***
## varbroct2003        -2.26195    0.20461 -11.055  < 2e-16 ***
## varbroct2004        -1.15350    0.11817  -9.761  < 2e-16 ***
## varbroct2005        -0.29824    0.08701  -3.428 0.000609 ***
## yearcross2000       -0.02756    0.02940  -0.937 0.348565
## inGreenHouse:flat2  -0.25165    0.13647  -1.844 0.065188 .
## inGreenHouse:flat3   0.25957    0.18745   1.385 0.166122
## outGreenHouse:row11  0.14635    0.03641   4.019 5.85e-05 ***
## outGreenHouse:row12  0.12414    0.03545   3.502 0.000462 ***
## outGreenHouse:row13  0.12050    0.03470   3.472 0.000516 ***
## outGreenHouse:posi  -0.32282    0.07003  -4.609 4.04e-06 ***
## fitField:crossWi    -0.58086    0.15305  -3.795 0.000148 ***
## fitField:crossWr    -0.08069    0.14552  -0.554 0.579272
## crossWi:fitChamber  -0.52844    0.35997  -1.468 0.142100
## crossWr:fitChamber  -0.29471    0.49249  -0.598 0.549563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`out.super` and `out.super.test_2` are the same.