Principle of Marginality

MATH 4939

2025-03-24

Modelling Rules based on the Principle of Marginality

The developers of R believed so strongly in the importance of the principle of marginality (PoM) that they used a system for defining linear models that creates hurdles to discourage users from violating the PoM unless they actively do so. This is in keeping with the principle that you should violate it only if you know what you are doing (IYKWUAD).

The main rules are:

1. If a model contains a 'term' (an interaction or a main effect) then it should also contain all included lower-order terms.

   - If it includes `X*W*Z`, then it should also include `X*W`,`X*Z`,`W*Z`,`X`, `Y`, `Z`, and the constant term `1`.
   - If it includes `X*W` and `V*Z`, then it should also include `X`, `Y`, `V`, `Z`, and the constant term `1`.
   - The above hold whether any of the terms are numerical objects or factors.

2. If a model contains a polynomial term, e.g. `I(X^3)` or `I(X^2*W^2)`, then it must also contain all included lower powers, e.g. if it contains `I(X^2*W^2)`, then it must also contain `I(X^2*W^2)`, `I(X*W^2)`, `I(X^2*W)`, `I(X*W)`, . . . , and the constant term `1`.

3. If a model contains a trigonometric term to model a harmonic of a seasonal periodic effect: e.g. `cos(2*pi*3*year)` then it must also contain the complementary term, e.g. `sin(2*pi*3*year)`, and all slower harmonics: `cos(2*pi*2*year) + sin(2*pi*2*year)`, `cos(2*pi*year) + sin(2*pi*year)`, and the constant term, `1`.

This is why, if you specify a model as `Y ~ X*Z`, the model that is used is:

$$E(Y) = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ$$

You can prevent this specifying the interaction with a `:` instead of a `*`.

`lm(y ~ X:Z)` will fit the model:

$$E(Y) = \gamma_0 + \gamma_1 XZ$$

To go all the way and even drop the intercept, you need to use

```
lm(y ~ X:Z - 1)
```

which will fit the model:

$$E(Y) = \psi XZ$$

What difference does it make?

Using the income, education and job type data in the Prestige data set in the car package

```r
library(car)
```

```
    Loading required package: carData
```

```r
library(spida2)
library(p3d)
```

```
    Loading required package: rgl


    Attaching package: 'p3d'
```

```
The following objects are masked from 'package:spida2':

    cell, center, ConjComp, dell, disp, ell, ell.conj, ellbox, ellplus,
    ellpt, ellptc, ellpts, ellptsc, elltan, elltanc, na.include, uv
```

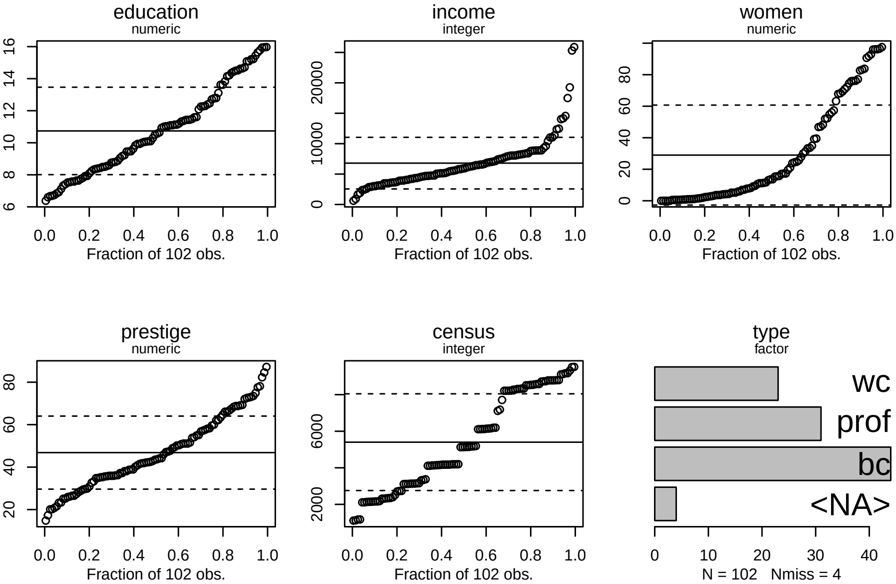
```
The following object is masked from 'package:car':

    Identify3d
```

```r
library(latticeExtra)
```

```
Loading required package: lattice
```

```r
xqplot(Prestige)
```


education
numeric
6
8
10
12
14
16
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 102 obs.
income
integer
0
10000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 102 obs.
women
numeric
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 102 obs.
prestige
numeric
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 102 obs.
census
integer
2000
6000
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of 102 obs.
type
factor
wc
prof
bc
<NA>
0
10
20
30
40
N = 102   Nmiss = 4

```r
dd <- droplevels(subset(Prestige, !is.na(type)))
# note that you can't use 'type != NA'
contrasts(dd$type)
```

```
     prof wc
bc      0  0
prof    1  0
wc      0  1
```

```r
fit1 <- lm(income ~ education*type, dd)
fit2 <- lm(income ~ education:type, dd)
fit3 <- lm(income ~ education:type - 1, dd)

fit1
```

```

Call:
lm(formula = income ~ education * type, data = dd)
```

```
Coefficients:
       (Intercept)           education            typeprof
           -1865.0               866.0             -3068.4
education:typeprof    education:typewc
             234.0              -569.2
```

```
fit2
```

```
Call:
lm(formula = income ~ education:type, data = dd)

Coefficients:
       (Intercept)    education:typebc  education:typeprof    education
           -2129.4               897.0               902.8
```

```
fit3
```

```
Call:
lm(formula = income ~ education:type - 1, data = dd)

Coefficients:
  education:typebc  education:typeprof    education:typewc
             647.0               753.0               457.3
```

```r
pred <- with(dd, pred.grid(type, education = seq(-1,20)))


# makes sure that 'type' has right levels

pred$fit_PoM <- predict(fit1, newdata = pred)
```

```
Warning: contrasts dropped from factor type
```

```r
pred$fit_NoMEs <- predict(fit2, newdata = pred)
```

```
Warning: contrasts dropped from factor type
```

```r
pred$fit_NoInt <- predict(fit3, newdata = pred)
```

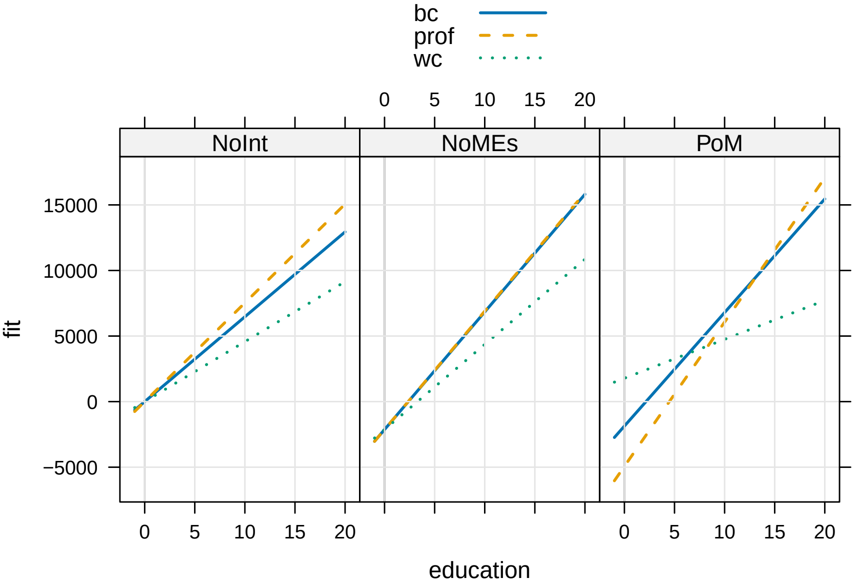
```
    Warning: contrasts dropped from factor type
```

```r
contrasts(pred$type)
```

```
          prof wc
    bc       0  0
    prof     1  0
    wc       0  1
```

```r
predl <- tolong(pred, sep = '_', timevar = 'model')

xyplot(fit ~ education | model, predl, type = 'l',
       par.settings = list(superpose.line = list(lwd = 2, lty = 1:3)),
       groups = type,
       layout = c(3,1),
       auto.key = list(space='top')
) +
  layer(panel.abline(v = 0)) +
  layer(panel.grid(h=-1,v=-1))
```


bc
prof
wc
NoInt
NoMEs
PoM
fit
15000
10000
5000
0
−5000
0
5
10
15
20
education

Key question:

- What would happen is were to measure education with different origin, e.g. with 0 at the start of high school?

```
dd$education2 <- dd$education - 8

fit1 <- lm(income ~ education2*type, dd)
fit2 <- lm(income ~ education2:type, dd)
fit3 <- lm(income ~ education2:type - 1, dd)

fit1
```

```
##
## Call:
## lm(formula = income ~ education2 * type, data = dd)
##
## Coefficients:
##         (Intercept)           education2             typeprof
##               5063.0                866.0              -1196.2
##               typewc  education2:typeprof    education2:typewc
##               -907.4                234.0               -569.2
```

```
fit2
```

```
## 
## Call:
## lm(formula = income ~ education2:type, data = dd)
## 
## Coefficients:
##          (Intercept)    education2:typebc  education2:typeprof  
##              4982.38               885.90               925.51  
##    education2:typewc  
##                45.57
```

```
fit3
```

```
## 
## Call:
```

```
## lm(formula = income ~ education2:type - 1, data = dd)
##
## Coefficients:
##   education2:typebc  education2:typeprof    education2:typewc
##                2116                 1705                 1559
```

```r
pred <- with(dd, pred.grid(type, education2 = seq(-1-8,20-8)))


# makes sure that 'type' has right levels

pred$fit_PoM <- predict(fit1, newdata = pred)
```

```
## Warning: contrasts dropped from factor type
```

```r
pred$fit_NoMEs <- predict(fit2, newdata = pred)
```

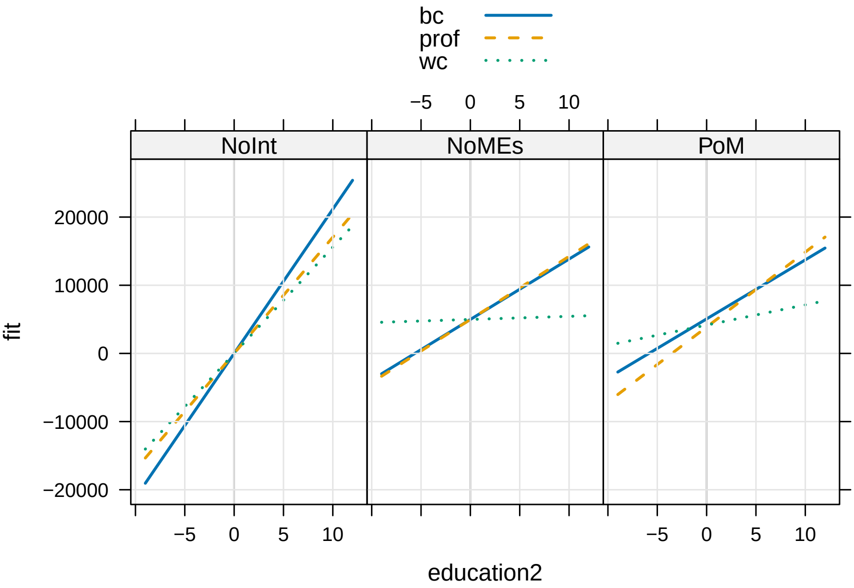
```
## Warning: contrasts dropped from factor type
```

```
pred$fit_NoInt <- predict(fit3, newdata = pred)
```

```
## Warning: contrasts dropped from factor type
```

```
predl <- tolong(pred, sep = '_', timevar = 'model')

xyplot(fit ~ education2 | model, predl, type = 'l',
       par.settings = list(superpose.line = list(lwd = 2, lty = 1:3)),
       groups = type,
       layout = c(3,1),
       auto.key = list(space='top')
) +
  layer(panel.abline(v = 0)) +
  layer(panel.grid(h=-1,v=-1))
```


bc
prof
wc
NoInt
NoMEs
PoM
fit
20000
10000
0
−10000
−20000
−5
0
5
10
education2