Different ways to write interaction terms in lm?

I have a question about which is the best way to specify an interaction in a regression model. Consider the following data:

```
d <- structure(list(r = structure(c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
     1L, 1L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L), .Label = c("r1","r2"),
     class = "factor"), s = structure(c(1L, 1L, 1L, 1L, 1L,
     2L, 2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 2L, 2L),
    .Label = c("s1","s2"), class = "factor"), rs = structure(c(1L, 1L,
     1L,1L, 1L,2L, 2L, 2L, 2L, 3L, 3L, 3L, 3L, 3L, 4L, 4L, 4L, 4L, 4L),
    .Label = c("r1s1","r1s2", "r2s1", "r2s2"), class = "factor"),
     y = c(19.3788027518437, 23.832287726332, 26.2533235300492,
     15.962906892112, 24.2873740664331, 28.5181676764727, 25.2757801195961,
     25.3601044326474, 25.3066440027202, 24.3298865128677, 32.5684219007394,
     31.0048406654209, 31.671238316086, 34.1933764518288, 36.8784821769123,
     41.6691435168277, 40.4669714825801, 39.2664137501106, 39.4884849591932,
     49.247505535468)), .Names = c("r","s", "rs", "y"),
     row.names = c(NA, -20L), class = "data.frame")
```

Two equivalent ways to specify the model with interactions are:

```
lm0 <- lm(y ~ r*s, data=d)
lm1 <- lm(y ~ r + s + r:s, data=d)
```

My question is if I could specify the interaction considering a new variable (rs) with the same levels of interaction:

```
lm2 <- lm(y ~ r + s + rs, data=d)
```

What advantages/disadvantages have this approach? And why the results from these two approaches are different?

```
summary(lm1)

lm(formula = y ~ r + s + r:s, data = d, x = TRUE)
            coef.est coef.se
(Intercept) 21.94     1.46
rr2         11.32     2.07
ss2          3.82     2.07
rr2:ss2      4.95     2.92
---
n = 20, k = 4
residual sd = 3.27, R-Squared = 0.87


summary(lm2)

lm(formula = y ~ r + s + rs, data = d, x = TRUE)
            coef.est coef.se
(Intercept) 21.94     1.46
rr2         11.32     2.07
ss2          8.76     2.07   # ss2 coef is different from lm1
rsr1s2      -4.95     2.92
---
n = 20, k = 4
residual sd = 3.27, R-Squared = 0.87
```

r | regression | interaction

edited Dec 2 '11 at 20:30 — chl ♦ 34.8k 6 107 230

asked Dec 2 '11 at 20:23 — Manuel Ramón 810 1 8 13

You mean that `rs` is defined as `interaction(r, s)` ? – chl ♦ Dec 2 '11 at 20:30

Perhaps you could show us the code that created rsr1s2? – jbowman Dec 2 '11 at 21:20

The rs factor was defined manualy (simply paste factors r and s). See the data set. – Manuel Ramón Dec 2 '11 at 21:29

1 I guess is related the way in the variables are relationated see `attr(terms(lm1),"factors")` and `attr(terms(lm2),"factors")` – Galled Dec 2 '11 at 22:13

2 Answers

The results are different because the way lm sets up the model with the interaction is different from how it is set up when you set it up yourself. If you look at the residual sd, it's the same,

which indicates (not definitively) that the underlying models are the same, just expressed (to the lm internals) differently.

If you define your interaction as `paste(d$s, d$r)` instead of `paste(d$r, d$s)` your parameter estimates will change again, in interesting ways.

Note how in your model summary for lm1 the coefficient estimate for ss2 is 4.94 lower than in the summary for lm2, with the coefficient for rr2:ss2 being 4.95 (if you print to 3 decimal places, the difference goes away). This is another indication that an internal rearrangement of terms has occurred.

I can't think of any advantage to doing it yourself, but there may be one with more complex models where you don't want a full interaction term but instead only some of the terms in the "cross" between two or more factors.

answered Dec 2 '11 at 22:40

jbowman
12.8k 1 26 53

The only advantage I see to define the interactions as in lm2 is that it is easy to perform multiple comparisons for the interaction term. What I do not quite understand is why different results are obtained if, in principle, it seems that the 2 approaches are the same. – Manuel Ramón Dec 3 '11 at 20:27

4 The approaches are the same, but the exact parameterizations of the model being estimated are different, so the results appear different. Consider a model with two binary regressors x_1, x_2 and an interaction. You have four categories, but you can write the model several different ways, e.g., let 1 be a constant term, with variables $(1, x_1, x_2, x_1 * x_2)$ or $(x_1, x_2, x_1 * x_2, (1 - x_1) * (1 - x_2)$, or others. The variables are just linear combinations of each other. The coefficient estimates will be different, but the model is really the same. – jbowman Dec 3 '11 at 20:43

Therefore, although different, both approaches are correct, aren't it? – Manuel Ramón Dec 3 '11 at 20:56

Right. Mathematically the matrices of independent variables in the various formulations are just linear transforms of each other, so the parameter estimates of one model can be calculated from the parameter estimates of another if one knows how the two models were actually set up. – jbowman Dec 4 '11 at 22:13

You might understand this behavior better if you look at the model matrices.

```
model.matrix(lm1 <- lm(y ~ r*s, data=d))
model.matrix(lm2 <- lm(y ~ r + s + rs, data=d))
```

When you look at these matrices, you can compare the constellations of `s2=1` with the other variables (i.e. when `s2=1`, which values do the other variables take?). You will see that these constellations differ slightly, which just means that the base category is different. Everything else is essientially the same. In particular, note that in your `lm1`, the coefficient on `ss2` equals the coefficients `ss2+rsr1s2` of `lm2`, i.e. 3.82=8.76-4.95, short of rounding errors.

For instance, executing the following code gives you exactly the same output as using the automatic setting of R:

```
d$rs <- relevel(d$rs, "r1s1")
summary(lm1 <- lm(y~ factor(r) + factor(s) + factor(rs), data=d))
```

This also provides a quick answer to your question: the really only reason to change the way factors are set up is to provide expositional clarity. Consider the following example: Suppose you regress wage on a dummy for high school completion interacted with a factor indicating if you belong to a minority.

That is: $wage = \alpha + \beta\, edu + \gamma\, edu * minority + \epsilon$

If said minority factor takes value 1 if you do belong to a minority, the coefficient β can be interpreted as a wage difference for non-minority individuals who have completed high school. If this is your coefficient of interest, then you should code it as such. Otherwise, suppose the minority factor takes the value of 1 if you do not belong to a minority. Then, in order to see how much more non-minority individuals earn when they complete high school, you would have to "manually" compute $\beta + \gamma$. Note though that all information is contained in the estimates though, and substantial results do not change by setting up the factors differently!

edited Apr 12 '14 at 19:45

answered Apr 12 '14 at 19:32

coffeinjunky
516 4 12