Using `R` formulae to test for main effects in the presence of higher-order interactions

Roger Levy

May 12, 2014

Abstract

Traditional analysis of variance (ANOVA) software allows researchers to test for the significance of main effects in the presence of interactions without exposure to the details of how the software encodes main effects and interactions to make these tests possible. Now that increasing numbers of researchers are using more general regression software, including mixed-effects models, to supplant the traditional uses of ANOVA software, conducting such tests generally requires greater knowledge of how to parameterize one's statistical models appropriately. Here I present information on how to conduct such tests using `R`, including relevant background information and worked examples.

1 Introduction

Suppose that you have predictors `X` and `Y` and response variable `Response`, and you are interested in performing a statistical data analysis of the effects of `X`, `Y`, and their interaction. One thing you may be interested in is assessing the significance of the main effect of `X`. In general, whenever some predictor `X` is present in a model both as a main effect and as a part of a higher order interaction, you should be very careful about what you mean by the "main effect of `X`". **The key thing to remember is that under the hood, the main effect of `X` *always* really means the effect of `X` when `Y=0`.** If `Y` is a continuous predictor, for example, centering it changes what the main effect of `X` means.

That being said, there are some circumstances in which it is sensible to talk about a "main effect of `X`" when `X` and `Y` interact. If `Y` is a factor, for example, one might want to say that there is a main effect of `X` if the average effect of `X` across all values of `Y` consistently goes in one direction or the other. In these circumstances, it's useful to know how to test for the main effect of `X` using `R` formulae. In certain special cases you can do it using the `aov()` function, but sometimes you may want to do it in other cases—for example, analyzing an experiment with crossed subject and item random effects using `lme4`. Thus it's useful to know the general method for writing formuale for fitting nested models whose difference is the presence or absence of a main effect of `X` when `X` interacts with `Y`.

If you are just interested in how to test for main effects in the presence of interactions, you can skip straight to Section 3. Section 2 gives background detail on `R` model formulae and contrast coding that may give you a deeper understanding of why this works. Section 4 gives some examples.

2 A bit of background on numeric predictors, categorical predictors, and interactions in `R` formulae

It helps to start with some review of how numeric and categorical predictors (factors) are treated in regression modeling in `R`, and also how interactions work. This section closely follows material from Chambers and Hastie (1991), the authoritative reference. Reviewing Section 11 of Venables et al. (2013) is also very useful.

We start with the interpretation of interactions. The basic way of specifying an interaction between `X` and `Y` is `X:Y`. This has the following interpretation (following Chambers and Hastie 1991, p. 25):

- **If `X` and `Y` are both factors** of m and n levels respectively, then `X:Y` introduces an $m \times n$ matrix of parameters into the model, one per unique combination of values of `X` and `Y`.
- **If `X` is a n-level factor and `Y` is numeric**, then `X:Y` introduces a linear effect of `Y` with a different slope for each level of `X` (hence n slopes).
- **If `X` and `Y` are both numeric**, then `X:Y` introduces a linear effect of the product of `X` and `Y` (i.e., equivalent to as `I(X:Y)`, and introducing just one parameter).

Note that `X:X` is always reduced down to `X` (even for numeric predictors).

Second comes the treatment of the `R` formula operator `*`. This is pretty simple:

`X*Y` is always expanded out to `1 + X + Y + X:Y`

Finally comes the question of how the inclusion of factors in one's statistical model affects how the model is parameterized. The simplest way of representing the contribution of an n-level factor `X` into one's statistical model is with dummy coding—introducing n dummy variables with the following mapping:

Level of X	X_1	X_2	$\dots$	X_n
1	1	0	$\dots$	0
2	0	1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
n	0	0	$\dots$	1

and one regression weight β_i for each dummy variable, so that the regression equation `Response ~ X` would then be interpreted as:

$$\eta = \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{1}$$

(where η is the linear predictor). However, there is often redundancy between components of model formulae. The simple formula `Response ~ X`, for example, is actually interpreted with an implicit intercept, equivalent to `Response ~ 1 + X`. But adding an extra intercept parameter into Equation (1) would not change the statistical model. To eliminate this redundancy, `R` uses CONTRAST CODING, which allows an n-level factor to be represented with $n-1$ dummy variables. A "true" contrast is one in which for each dummy variable, the sum of the coefficients across factor levels is zero. Such contrasts include Helmert contrasts (which are also orthogonal) and sum or deviation contrasts (which are not), but not the `R`-default treatment contrasts. `R` is fairly clever about identifying when dummy coding cannot be used (see Chambers and Hastie, 1991, pp. 38–39, for details). The relevant bottom line for us here is that when true (sum-to-zero) contrasts are used for factors, the associations of parameters correspond intuitively to what we think of as "main effects" and "interactions": for an interaction between `X` and `Y`, the main effect of `X` is the average (across all levels of `Y`, not weighted by amount of data in each level) effect of changing between levels of `X`, and likewise for the main effect of `Y`; and the interaction between `X` and `Y` is the deviation of actual behavior from that predicted by the main effects.

This puts us in a situation where we're ready to test for main effects in the presence of interactions. Based on the discussion immediately above, what we want to mean by "there is a significant main effect of a factor `X`" in the presence of an interaction with `Y` is that we can conclude with some degree of statistical certainty that the overall differences between levels of `X` (averaged across all values of `Y`) are not all zero. Intuitively, we want to implement that test by including the interaction terms `X:Y` in our model, but not the main-effect term `X`. However, we cannot do this directly in `R` using factors, because as we just saw, the parceling out of dummy variables and corresponding parameters happens with respect to the entire model formula. So for factors (unlike for numeric variables), `1 + X + X:Y` is the same model as `X*Y`, because `X:Y` already introduces a separate parameter for each combination of levels of `X` and `Y` – the different formulae simply lead to different parameterizations. We can, however, isolate the contributions of the main effect of `X` by using true contrast coding directly to create numerical variables to represent the different levels of `Y`. Once `Y` is coded with numeric variables, we can subtract out the main effect of `X` from the full model with the interaction, because `1 + Y + X:Y` is not equivalent to `X*Y` when `Y` is numeric. The next section explains how to do this in practice.

3 Testing for main effects in the presence of interactions

Assume that you are interested in testing for a main effect of `X` in the presence of an interaction with another predictor `Y`. As described in the introduction, the main effect of `X` is really just the effect of changing `X` around while holding `Y` constant at zero. The strategy is (i) to fit the full model (with main effects and interactions), (ii) then fit a reduced model with a main effect of `Y` and an interaction, but with no main effect of `X`, and then (iii) to do a nested model comparison between the full and reduced models to assess the statistical significance of the main effect of X. Step (ii) is the non-trivial part, so I explain below how to do it.

1. **If Y is numeric**, then you can use

   ```
   Response ~ X*Y - X
   ```

   or

   ```
   Response ~ Y + X:Y
   ```

   to write the null-hypothesis model for which there's no main effect of X but there is an interaction. Make sure that what you mean by the "main effect of X" is the effect of changing X around while Y stays constant at zero, because that is what you will get!

2. **If Y is a factor**, then you need to convert Y into an equivalent numeric variable that will allow you to isolate the main effect of X, because when Y is a factor, X*Y - X and Y + X:Y will be the same model (perhaps with a different parameterization). By the main effect of X when Y is a factor, you presumably mean the "across-the-board" effect (averaging over values of Y) of changing X. You can get this by using *any* true contrast scheme for Y (i.e., any scheme in which the coefficients add up do zero), because the main effect of X is the effect of changing X when Y=0, and true contrast schemes put "grand average" behavior at Y=0. Thus both contr.sum() and contr.helmert() work. Here's a relatively easy way to convert a K-level factor Y into a set of numeric variables consistent with a sum-coding representation:

   ```
   Y.numeric <- sapply(Y,function(i) contr.sum(K)[i,])
   ```

   If Y had N observations, then now Y.numeric will be a $K \times N$ matrix where each row is one of the contrast variables. Then you can do

   ```
   Y1 <- Y.numeric[1,]
   Y2 <- Y.numeric[2,]
   ```

   and so forth up to

   ```
   YK <- Y.numeric[K,]
   ```

   and then your null-hypothesis model becomes

   ```
   Response ~ Y1 + X:Y1 + Y2 + X:Y2 + ... + YK + X:YK
   ```

   When Y is a 2-level factor this is especially easy and you just can do

   ```
   Response ~ Y.numeric + X:Y.numeric
   ```

Once you have fit your reduced model, then fit the full model Response ~ X * Y and do a nested model comparison between the two.

4 Examples

4.1 Simple linear regression and equivalence to aov()

Testing for a main effect of X when X is a two-level factor and the model also includes a main effect of a three-level factor Y and the interaction between X and Y:

```
> set.seed(1)
> dat <- expand.grid(X=factor(c("x1","x2")),
+                    Y=factor(c("y1","y2","y3")),
+                    repetitions=1:5)
> beta.XY <- matrix(c(1,4,5,2,3,3),2,3)
> # Create response with very small noise level so that patterns are clear
> dat$Response <- with(dat,beta.XY[cbind(X,Y)] + rnorm(nrow(dat),sd=0.1))
> with(dat,                             # empirical condition means
+      tapply(Response,list(X,Y),mean)) # closely match true effects
```

```
         y1       y2       y3
x1 1.013616 5.032565 3.028430
x2 3.984901 2.011127 2.978836
```

```
> Y.numeric <- sapply(dat$Y,function(i) contr.sum(3)[i,])
> dat$Y1 <- Y.numeric[1,]
> dat$Y2 <- Y.numeric[2,]
> m0 <- lm(Response ~ Y1 + X:Y1 + Y2 + X:Y2,dat)
> m1 <- lm(Response ~ X*Y,dat)
> anova(m0,m1,test="F") # Test has 1 d.f. in the numerator; no sig. main effect of X
```

```
Analysis of Variance Table

Model 1: Response ~ Y1 + X:Y1 + Y2 + X:Y2
Model 2: Response ~ X * Y
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     25 0.24372
2     24 0.23543  1 0.0082913 0.8452 0.3671
```

```
> summary(aov(Response ~ X*Y,dat)) # Linear-regression and ANOVA results match
```

```
            Df Sum Sq Mean Sq  F value Pr(>F)
X            1   0.01   0.008    0.845  0.367
Y            2   5.23   2.614  266.512 <2e-16 ***
X:Y          2  44.89  22.446 2288.164 <2e-16 ***
Residuals   24   0.24   0.010
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

4.2 Mixed-effects regression

The same case in a mixed-effects models analysis with crossed subjects and items, where both `X` and `Y` are within-subjects and within-items. Note that the null-hypothesis model *includes* random slopes for the main effect of X and Y, as well as for their interaction (including a separate random slope for all combinations of X and Y achieves this).

```
> library(mvtnorm)
> library(lme4.0)
> set.seed(1)
> M <- 12
> dat <- expand.grid(X=factor(c("x1","x2")),Y=factor(c("y1","y2","y3")),
+                    subj=factor(paste("S",1:M)),item=factor(paste("I",1:M)))
> dat$XY <- with(dat,factor(paste(X,Y)))
> beta.XY <- matrix(c(1,4,5,2,3,3),2,3)
> b.S <- rmvnorm(M,rep(0,6),diag(6)/100)
> b.I <- rmvnorm(M,rep(0,6),diag(6)/100)
> dat$Response <- with(dat,beta.XY[cbind(X,Y)] +
+                      b.S[cbind(subj,XY)] +
+                      b.I[cbind(item,XY)] +
+                      rnorm(nrow(dat),sd=0.1))
> Y.numeric <- sapply(dat$Y,function(i) contr.sum(3)[i,])
> dat$Y1 <- Y.numeric[1,]
> dat$Y2 <- Y.numeric[2,]
> m0 <- lmer(Response ~ Y1 + X:Y1 + Y2 + X:Y2 + (XY|subj) + (XY|item),dat)
> m1 <- lmer(Response ~ X*Y + (XY|subj) + (XY|item),dat)
```

```
> anova(m0,m1)
```

```
Data: dat
Models:
m0: Response ~ Y1 + X:Y1 + Y2 + X:Y2 + (XY | subj) + (XY | item)
m1: Response ~ X * Y + (XY | subj) + (XY | item)
   Df     AIC     BIC logLik  Chisq Chi Df Pr(>Chisq)
m0 48 -1040.4 -811.85 568.20
m1 49 -1038.5 -805.16 568.24 0.0666      1     0.7963
```

If I were to report this mixed-effects regression analysis in a journal article I might say:

> We tested for a main effect of X by converting Y to a sum-coding numeric representation and conducting a likelihood-ratio test between mixed-effects models differing only in the presence or absence of a fixed main effect of X. Both models included in their fixed effects an intercept, a main effect of Y, and an interaction between X and Y. Both models also had maximal random effects structure, namely random intercepts plus random slopes for X, Y, and their interaction for both subjects and items. The likelihood-ratio test showed no evidence for a main effect of X ($p = 0.796$, 1 d.f.).

References

Chambers, J. M. and Hastie, T. J. (1991). Statistical models. In Chambers, J. M. and Hastie, T. J., editors, *Statistical Models in S*, chapter 2, pages 13–44. Chapman and Hall.

Venables, W. N., Smith, D. M., and the R Core Team (2013). *An Introduction to R*, 3.0.1 edition.