canoe.moore

Christopher Moore's Blog

Linear mixed-effects regression p-values in R: A likelihood ratio test function

By Christopher Moore on September 7, 2010 9:18 PM | 7 Comments

Following **Douglas Bates' advice** for those required to produce *p*-values for fixed-effects terms in a mixed-effects model, I wrote a function to perform a likelihood ratio test for each term in an `lmer()` object. Bates has championed the notion that calculating the *p*-values for fixed effects is not trivial. That's because with unbalanced, multilevel data, the **denominator degrees of freedom used to penalize certainty are unknown** (i.e., we're uncertain about how uncertain we should be). As the author of **lme4**, the foremost mixed-effects modeling package in R, he has practiced what he preaches by declining to **approximate denominator degrees of freedom as SAS does**. Bates contends that alternative inferential approaches make *p*-values unnecessary. I agree with that position, focusing instead on information criteria and effect sizes. However, as a program evaluator, I recognize that some stakeholders find *p*-values useful for understanding findings.

A likelihood ratio test can be used to test $H_0 : \beta = 0$ if the sample size is large. According to **Fitzmaurice, Laird, and Ware**, twice the difference between the maximized log-likelihoods of two nested models, $G^2 = 2(\hat{l}_{\text{full}} - \hat{l}_{\text{reduced}})$, represents the degree to which the reduced model is inadequate. When the sample size is large, $G^2 \sim \chi^2$ with degrees of freedom equal to the difference in parameters between the full and reduced models, $m_{\text{full}} - m_{\text{reduced}}$. The *p*-value is $\Pr(\chi^2 > G^2 | H_0)$.

What if the sample size is not large? A *p*-value based on χ^2 will be too liberal (i.e., the type I error rate will exceed the nominal *p*-value). More conservatively, we might say that $G^2 \sim F$ with $m_{\text{full}} - m_{\text{reduced}}$ numerator and $N_{\text{effective}} - m_{\text{full}} - 1$ denominator degrees of freedom. According to **Snijders and Bosker**, the effective sample size lies somewhere between M total micro-observations (i.e., at level one) and N clusters randomly sampled in earlier stages (i.e., at higher levels). Formally, the effective sample size is $N_{\text{effective}} = \frac{Nn}{1+(n-1)\rho}$, where n observations are nested within each cluster and intraclass correlation is $\rho = \frac{\tau^2}{\tau^2+\sigma^2}$. Even if $M = Nn$ is large, $N_{\text{effective}}$ (and statistical power) could be quite small if N is small and ρ is large: $\frac{5*100}{1+(100-1)0.8} = 6.2$. Unbalanced designs, modeling three or more levels, and cross-level interactions add to our uncertainty about the denominator degrees of freedom.

The function I wrote chews up the `lmer()` model call and concatenates the frame and model matrix slots, after which it iteratively fits (via maximum likelihood instead of restricted ML) models reduced by each fixed effect and compares them to the full model, yielding a vector of *p*-values based on $\chi^2(1)$. As the example shows, the function can handle shortcut formulas whereby lower order terms are implied by an interaction term. The function doesn't currently handle weights, `glmer()` objects, or on-the-fly transformations of the dependent variable [e.g., `log(dep.var) ~ ...`]. The accuracy of resulting *p*-values depends on large sample properties, as discussed above, so I don't recommend using the function with small samples. I'm working on another function that will calculate *p*-values based on the effective sample size estimated from intraclass correlation. I will post that function in a future entry. I'm sure the following function could be improved, but I wanted to go ahead share it with other applied researchers whose audience likes *p*-values. Please **let me know** if you see ways to make it better.

```
p.values.lmer <- function(x) {
  summary.model <- summary(x)
  data.lmer <- data.frame(model.matrix(x))
  names(data.lmer) <- names(fixef(x))
  names(data.lmer) <- gsub(pattern=":", x=names(data.lmer), replacement=".", fixed=T)
  names(data.lmer) <- ifelse(names(data.lmer)=="(Intercept)", "Intercept", names(data.lmer))
  string.call <- strsplit(x=as.character(x@call), split=" + (", fixed=T)
  var.dep <- unlist(strsplit(x=unlist(string.call)[2], " ~ ", fixed=T))[1]
  vars.fixef <- names(data.lmer)
  formula.ranef <- paste("+ (", string.call[[2]][-1], sep="")
  formula.ranef <- paste(formula.ranef, collapse=" ")
  formula.full <- as.formula(paste(var.dep, "~ -1 +", paste(vars.fixef, collapse=" + "),
                  formula.ranef))
  data.ranef <- data.frame(x@frame[,
              which(names(x@frame) %in% names(ranef(x)))])
  names(data.ranef) <- names(ranef(x))
  data.lmer <- data.frame(x@frame[, 1], data.lmer, data.ranef)
  names(data.lmer)[1] <- var.dep
  out.full <- lmer(formula.full, data=data.lmer, REML=F)
  p.value.LRT <- vector(length=length(vars.fixef))
  for(i in 1:length(vars.fixef)) {
    formula.reduced <- as.formula(paste(var.dep, "~ -1 +", paste(vars.fixef[-i],
                       collapse=" + "), formula.ranef))
    out.reduced <- lmer(formula.reduced, data=data.lmer, REML=F)
    print(paste("Reduced by:", vars.fixef[i]))
    print(out.LRT <- data.frame(anova(out.full, out.reduced)))
```

Categories

About me (1)
Personal (23)
Praxes (45)

Monthly Archives

March 2014 (1)
July 2013 (1)
April 2013 (1)
March 2013 (1)
November 2012 (1)
October 2012 (1)
September 2012 (1)
May 2012 (1)
October 2011 (1)
September 2011 (1)
August 2011 (1)
July 2011 (2)
June 2011 (1)
January 2011 (1)
October 2010 (2)
September 2010 (3)
August 2010 (3)
July 2010 (1)
June 2010 (1)
April 2010 (1)
March 2010 (3)
February 2010 (3)
January 2010 (1)
November 2009 (1)
October 2009 (1)
September 2009 (4)
August 2009 (1)
July 2009 (2)
June 2009 (1)
May 2009 (3)
March 2009 (1)
February 2009 (3)
January 2009 (2)
November 2008 (1)
October 2008 (3)
September 2008 (3)
August 2008 (2)
July 2008 (3)
April 2008 (5)

Recent Entries

Multilevel modeling: From lme4 to Mplus
Measures of Effective Teaching weighting schemes: An improved plot
Applying generalizability theory with R: A package
Which schools are closing achievement gaps in Minnesota?
Evaluation 2012 presentation

```
    p.value.LRT[i] <- round(out.LRT[2, 7], 3)
  }
  summary.model@coefs <- cbind(summary.model@coefs, p.value.LRT)
  summary.model@methTitle <- c("\n", summary.model@methTitle,
                           "\n(p-values from comparing nested models fit by maximum likelihood)")
  print(summary.model)
}

library(lme4)
library(SASmixed)
lmer.out <- lmer(strength ~ Program * Time + (Time|Subj), data=Weights)
p.values.lmer(lmer.out)
```

Yields:

```
Linear mixed model fit by REML
(p-values from comparing nested models fit by maximum likelihood)
Formula: strength ~ Program * Time + (Time | Subj)
   Data: Weights
  AIC  BIC logLik deviance REMLdev
 1343 1383 -661.7     1313    1323
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subj     (Intercept) 9.038486 3.00641
          Time        0.031086 0.17631  -0.118
 Residual             0.632957 0.79559
Number of obs: 399, groups: Subj, 57

Fixed effects:
                Estimate Std. Error   t value p.value.LRT
(Intercept)     79.99018    0.68578 116.64000       0.000
ProgramRI        0.07009    1.02867   0.07000       0.944
ProgramWI        1.11526    0.95822   1.16000       0.235
Time            -0.02411    0.04286  -0.56000       0.564
ProgramRI:Time   0.12902    0.06429   2.01000       0.043
ProgramWI:Time   0.18397    0.05989   3.07000       0.002

Correlation of Fixed Effects:
            (Intr) PrgrRI PrgrWI Time   PrRI:T
ProgramRI   -0.667
ProgramWI   -0.716  0.477
Time        -0.174  0.116  0.125
ProgrmRI:Tm  0.116 -0.174 -0.083 -0.667
ProgrmWI:Tm  0.125 -0.083 -0.174 -0.716  0.477
```

Categories: Praxes

7 Comments

Christopher Desjardins | September 23, 2010 9:42 PM | Reply

This is very useful. However, I would recommend that readers also consider using a fit index, such as the AIC or BIC, or in the case of comparing nested models using the anova() function. I know that lots of folks rely on p-values for decision making, I am quite skeptical of them as you know especially when compared against some arbitrary 0.05 cutpoint. In fact, I would argue using alpha = 0.05 to base decisions is bad practice indeed. Instead, I recommend examining and exploring alternative hypotheses via model fitting as I feel they more accurately capture the inherent uncertainty underlying statistics. Or at a minimum just presenting the p-values and letting the readers decide. However, I do understand as an applied researcher and someone who works for the government, that interested parties are less interested in the underlying academics and more about whether such and such effect is 'significant' or not.
Cheers,
Chris

Steve Brady | October 20, 2010 2:49 PM | Reply

This works well for singular random effects.

Can the function be modified to handle nested random effects? At the moment, when I try it with nested random effects, I receive the following error:

Error in names(data.ranef) 'names' attribute [6] must be the same length as the vector [2]

Ling Cui | May 1, 2012 5:16 PM | Reply

Thank you very much for sharing this; very handy and useful!

elmar tobi | August 28, 2012 9:10 AM | Reply

Thank you very very much. I have been using a anova and 2*(1-pnorm(abs(t-value)) where possible. P-values may indeed not be perfect but in genetics its the norm...
A very happy lmer user!

Jaap Denissen | August 30, 2013 6:10 AM | Reply

Applying generalizability theory with R

Estimating congeneric reliability with R

Converting Rasch units (RITs) to thetas

Urban canoeing: Dining at The Sample Room

Urban canoeing: North/Northeast Minneapolis

Links

50 Rivers Project

Adventure Learning at University of Idaho (AL@UI)

Christopher David Desjardin's Blog

Citizen-Statistician

CodeCogs Online LaTeX Equation Editor

Dad vs Wild

Fiscal Issues & Geeky Stuff

Jeremy Y. Wang: A Grad Student's Thoughts on Mind, Brain, Education and Science

KBEM Jazz 88, operated by Minneapolis Public Schools

KFAI Radio Without Boundaries

My Kitchen and Food Adventures

Southern Poverty Law Center

stacyboemiller: A Little Bit of Chaos. A Lot of Creativity.

Surly Bikes Blog

Statistical Modeling, Causal Inference, and Social Science

Talkin' 'Bout the Next Generation

Wilder Research

Search

Search

This is great, thanks!

I spent a great deal of time looking for a way to obtain p-values for multilevel models. I used to work with pvals.fnc from the languageR package, but this no longer works with newer versions of R.

The function written by Christopher works like a charm and is consistent with state-of-the-art methodological advice. I needed to tweak the function a bit, however, pertaining to three lines of code:

p.value.LRT[i] summary.model$coefficients summary.model$methTitle

Christopher Moore | November 2, 2013 9:27 AM | Reply

Thanks to readers for informing me that the p.values.lmer() function no longer works with the new version of lme4. Please see earlier comments and Laurie Samuels' suggestions below for adapting the function work with the new version. I have no plans to update the function myself because I am too busy writing my dissertation.

> Hi Christopher-
> Thank you so much for posting your code for getting p-values from lme4 (http://blog.lib.umn.edu/moor0554/canoemoore/2010/09/lmer_p-values_lrt.html) --- I have found it really useful. I recently updated my lme4 to the latest version, and there are apparently some changes that make the new version incompatible with code used for the old version... (http://lme4.r-forge.r-project.org/misc/lme4_conversion.html). So I wanted to let you know the two things I changed in your code to make it work with the new version:
>
> 1. The p.value.LRT[i] line now needs to be "p.value.LRT[i] 2. On the next two lines, the @ notation doesn't work with the latest version of lme4, so @coefs and @methTitle now need to be $coefficients and $methTitle, respectively.
>
> I think this is what Jaap Denissen's comment was saying, but I didn't understand the comment until after I had worked through everything. I would have just posted all this as a comment myself so that other people could see it, but I kept failing the captcha...
>
> Thank you again!
> Laurie Samuels
> Vanderbilt University

Laurie Samuels replied to comment from Christopher Moore | November 14, 2013 3:53 PM | Reply

It looks like the autoformatting on the blog is having trouble with the R code in my original email. In the p.value.LRT line, the only change is that the 7 needs to be changed to an 8.

Leave a comment

Name

Email Address

URL

☐ Remember personal info?

Comments (You may use HTML tags for style)

Captcha:

Type the characters you see in the picture above.

Preview Submit