How to get coefficients and their confidence intervals in mixed effects models?

Asked 10 years, 11 months ago Modified 2 years, 2 months ago Viewed 79k times Part of R Language Collective

40

In `lm` and `glm` models, I use functions `coef` and `confint` to achieve the goal:

```
m = lm(resp ~ 0 + var1 + var1:var2) # var1 categorical, var2 continuous
coef(m)
confint(m)
```

Now I added random effect to the model - used mixed effects models using `lmer` function from lme4 package. But then, functions `coef` and `confint` do not work any more for me!

```
> mix1 = lmer(resp ~ 0 + var1 + var1:var2 + (1|var3))
                                      # var1, var3 categorical, var2 continuous
> coef(mix1)
Error in coef(mix1) : unable to align random and fixed effects
> confint(mix1)
Error: $ operator not defined for this S4 class
```

I tried to google and use docs but with no result. Please point me in the right direction.

EDIT: I was also thinking whether this question fits more to https://stats.stackexchange.com/ but I consider it more technical than statistical, so I concluded it fits best here (SO)... what do you think?

r lme4 random-effects mixed-models

Share Improve this question Follow

edited Apr 13, 2017 at 12:44
Community Bot
1 1

asked Jun 17, 2012 at 15:36
Tomas
57.2k 48 234 371

To get you started until someone like @BenBolker shows up (an expert): `?lmer` lists methods `fixef` and `ranef` in addition to `coef`. Since your error says it's having trouble combining the two, the issue is likely that your model specification is somehow "unusual". – joran Jun 17, 2012 at 16:01

Thanks @joran. My model spec is maybe unusual in omitting the intercept - I want to do this, because otherwise the coefficients are nonsense. `var1` is categorical and I want "group specific intercepts" for each its category. If I allow the intercept (remove `0 +` from formula), `coef` runs but doesn't give what I expect. `fixef` works great, thanks! However the `confint` doesn't work at all. – Tomas Jun 17, 2012 at 16:09

I would extract the data you need directly from the S4 object -- see this post's answers: stackoverflow.com/questions/8526681/... – baha-kev Jun 17, 2012 at 16:26

Thanks @baha-kev, but are you sure the confidence intervals are in this object? I don't think so... – Tomas Jun 17, 2012 at 21:52

1 I am fixing the bug(let)? in `coef` in the r-forge versions of lme4 (lme4.0, the currently stable branch which corresponds to CRAN-lme4), and lme4, the development branch). `confint` is a bigger can of worms, as has been discussed, although the development branch of lme4 can calculate profile confidence intervals ... – Ben Bolker Jun 26, 2012 at 8:23

7 Answers

Sorted by: Highest score (default)

17

Not sure when it was added, but now confint() is implemented in lme4. For example the following example works:

```
library(lme4)
m = lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
confint(m)
```

Share Improve this answer Follow

answered Mar 10, 2017 at 20:16
jciloa
1,009 1 11 22

14

There are two new packages, lmerTest and lsmeans, that can calculate 95% confidence limits for `lmer` and `glmer` output. Maybe you can look into those? And coefplot2, I think can do it too (though as Ben points out below, in a not so sophisticated way, from the standard errors on the Wald statistics, as opposed to Kenward-Roger and/or Satterthwaite df approximations used in `lmerTest` and `lsmeans`)... Just a shame that there are still no inbuilt plotting facilities in package `lsmeans` (as there are in package `effects()`, which btw also returns 95% confidence limits on `lmer` and `glmer` objects but does so by refitting a model without any of the random factors, which is evidently not correct).

Share Improve this answer Follow

edited Feb 16, 2017 at 1:53
kdauria
6,250 4 33 53

answered Jun 26, 2013 at 20:36
Tom Wenseleers
7,463 7 61 103

2 `coefplot2` does it very naively, by computing 1.96 times the Wald standard errors -- it doesn't address the very significant issues of finite-size corrections to the CIs – Ben Bolker Jun 26, 2013 at 21:14

1 Check also this post stats.stackexchange.com/questions/117641/... for a more detailed answer – Tom Wenseleers Oct 1, 2015 at 11:29

`lmerTest` is now nicely described in JoSS jstatsoft.org/article/view/v082i13 – radek Dec 19, 2017 at 6:47

9 Note that many of these comments are now quite outdated. Using `emmeans` or `lmerTest` is the way to go, and there are plotting methods now. – Axeman May 15, 2018 at 9:00

9

I'm going to add a bit here. If `m` is a fitted `(g)lmer` model (most of these work for `lme` too):

- `fixef(m)` is the canonical way to extract coefficients from mixed models (this convention began with `nlme` and has carried over to `lme4`)
- you can get the full coefficient table with `coef(summary(m))`; if you have loaded `lmerTest` before fitting the model, or convert the model after fitting (and then loading `lmerTest`) via `coef(summary(as(m,"merModLmerTest")))`, then the coefficient table will include p-values. (The coefficient table is a matrix; you can extract the columns via e.g. `ctab[,"Estimate"]`, `ctab[,"Pr(>|t|)"]`, or convert the matrix to a data frame and use `$`-indexing.)
- As stated above you can get *likelihood profile* confidence intervals via `confint(m)`; these may be computationally intensive. If you use `confint(m, method="Wald")` you'll get the standard +/- 1.96SE confidence intervals. (`lme` uses `intervals(m)` instead of `confint()`.)

If you prefer to use `broom.mixed`:

- `tidy(m,effects="fixed")` gives you a table with estimates, standard errors, etc.
- `tidy(as(m,"merModLmerTest"), effects="fixed")` (or fitting with `lmerTest` in the first place) includes p-values
- adding `conf.int=TRUE` gives (Wald) CIs
- adding `conf.method="profile"` (along with `conf.int=TRUE`) gives likelihood profile CIs

You can also get confidence intervals by parametric bootstrap (`method="boot"`), which is considerably slower but more accurate in some circumstances.

Share Improve this answer Follow

edited Mar 19, 2021 at 23:23

answered Mar 19, 2021 at 23:03
Ben Bolker
207k 25 366 451

Hi Ben, thanks! I'm a bit confused, what does the standalone dot `.` mean? If it's just a model var name, why not use e.g. `m`? :-) – Tomas Mar 19, 2021 at 23:21

I could use `m`. Sometimes I use `.` as a placeholder. – Ben Bolker Mar 19, 2021 at 23:22

9

Assuming a normal approximation for the fixed effects (which confint would also have done), we can obtain 95% confidence intervals by

estimate + 1.96*standard error.

The following does not apply to the variance components/random effects.

```
library("lme4")
mylm <- lmer(Reaction ~ Days + (Days|Subject),  data =sleepstudy)

# standard error of coefficient
```

```
days_se <- sqrt(diag(vcov(mylm)))[2]

# estimated coefficient

days_coef <- fixef(mylm)[2]

upperCI <-  days_coef + 1.96*days_se
lowerCI <-  days_coef  - 1.96*days_se
```

Share Improve this answer Follow

answered Jun 17, 2012 at 18:52
julieth
430 4 9

1 Hi julieth, nice idea, however there is a difference between the real confidence intervals (computed by confint) and these .... Maybe the t-distribution would give the same result as confint (not sure about this though), but in this case I don't know the df which should be used. – Tomas Jun 17, 2012 at 21:57

In other words, this is the reason why I prefer to use functions like `confint` etc. to do all this for me... (especially if I'm not sure about the normal distribution of coefficients). – Tomas Jun 17, 2012 at 21:59

1 The t-distribution is asymptotically normal and the degrees of freedom for the error term in many multi-level designs is so high that the error distribution is normal at that point. Therefore, if you have a design with lots of degrees of freedom this is a perfectly reasonable confidence interval estimate. – John Jun 18, 2012 at 0:36

8

I suggest that you use good old lme (in package nlme). It has confint, and if you need confint of contrasts, there is a series of choices (estimable in gmodels, contrast in contrasts, glht in multcomp).

Why p-values and confint are absent in lmer: see http://finzi.psych.upenn.edu/R/Rhelp02a/archive/76742.html .

Share Improve this answer Follow

answered Jun 17, 2012 at 16:24
Dieter Menne
10.1k 43 67

Thanks Dieter, I will try the older package. The absence of p-value - and possibility to tell significance right away - also alarmed me! Doesn't make any sense to me, if I will be able to get confidence interval then I will simply look whether it contains zero - and have the significance anyway! Regards, – Tomas Jun 17, 2012 at 22:07

I forget to mention that confint(glht... from package multcomp give asymptotic confidence intervals for lmer. Douglas Bates caveats still apply, but his bold move to leave the p-value out of lmer/gaussian certainly has stirred the soup. – Dieter Menne Jun 18, 2012 at 6:24

Dieter, what do you mean with "confint(glht" ? There's no confint function in multcomp package... – Tomas Jun 18, 2012 at 11:00

Dieter, I tried the old package lme, nice, it has p-values. But my main concern is to get the confidence interval of fixed effect coefficients. How do I do that? confint returns some big matrix, glht seems too complicated.. – Tomas Jun 18, 2012 at 11:20

1 using `intervals(mix1)` will you give you asymptotic confidence intervals as in @julieth's answer below; `intervals(mix1)$fixed` extracts the fixed-effect intervals. These are based on the normal approximation, not the t distribution or anything more exotic ... – Ben Bolker Jun 26, 2012 at 8:27

1

To find the coefficient, you can simply use the summary function of lme4

```
m = lm(resp ~ 0 + var1 + var1:var2) # var1 categorical, var2 continuous
m_summary <- summary(m)
```

to have all coefficients :

```
m_summary$coefficient
```

If you want the confidence interval, multiply the standart error by 1.96:

```
CI <- m_summary$coefficient[,"Std. Error"]*1.96
print(CI)
```

Share Improve this answer Follow

answered May 23, 2017 at 13:32
denis
5,521 1 12 39

2 here the 1.96 factor is for 95% confidence interval of course – denis May 25, 2017 at 12:54

1

I'd suggest `tab_model()` function from `sjPlot` package as alternative. Clean and readable output ready for markdown. Reference here and examples here.

For those more visually inclined `plot_model()` from the same package might come handy too.

Alternative solution is via `parameters` package using `model_parameters()` function.

Share Improve this answer Follow

edited Nov 7, 2020 at 19:31

answered Nov 7, 2020 at 19:18

radek
7,200 8 58 81

2 or `broom.mixed::tidy()` – Ben Bolker Nov 7, 2020 at 20:05

@Ben Bolker's should be the top rated answer IMO :) – Extrapolator Mar 19, 2021 at 8:17