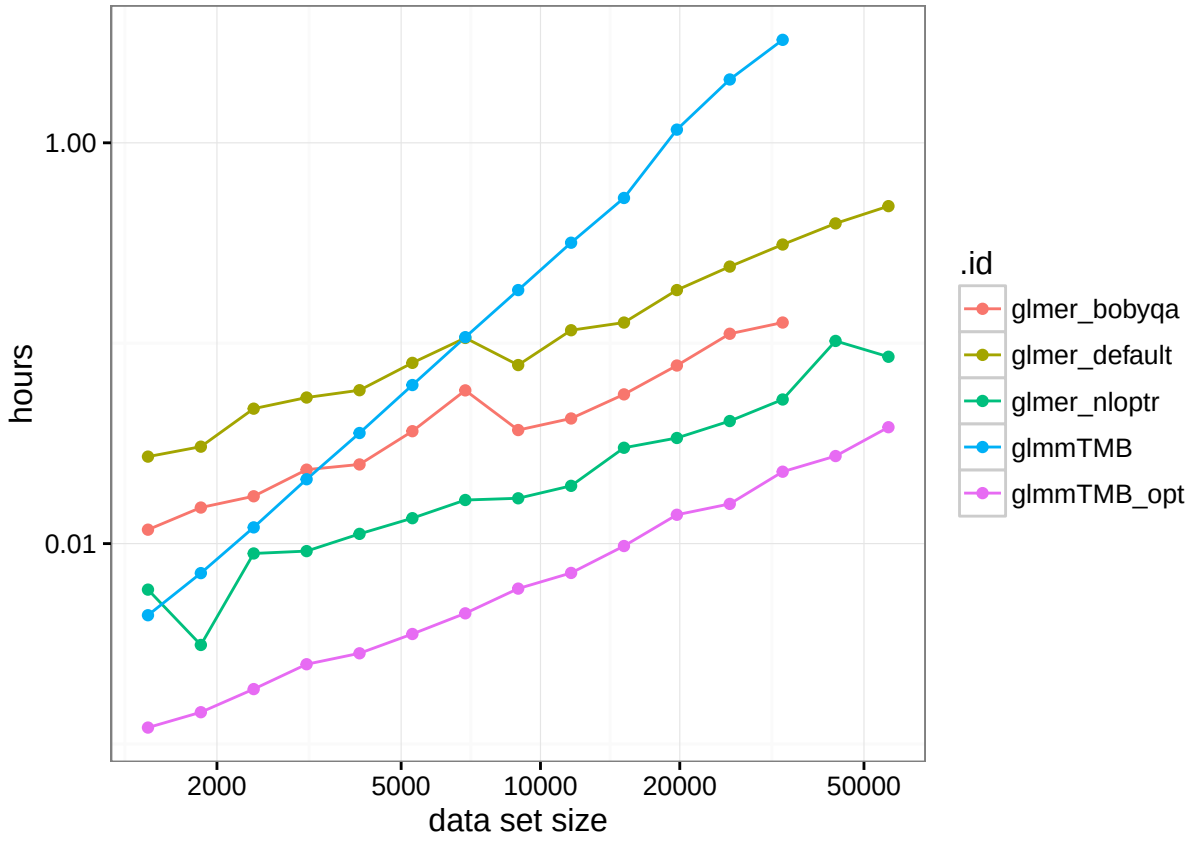
Ben Bolker

18:50 24 November 2015

```
load("mergedData2.rda")
library("plyr")
library("reshape2")
library("ggplot2"); theme_set(theme_bw())
library("gridExtra")
library("nlme")
options(digits=3)
```

Times



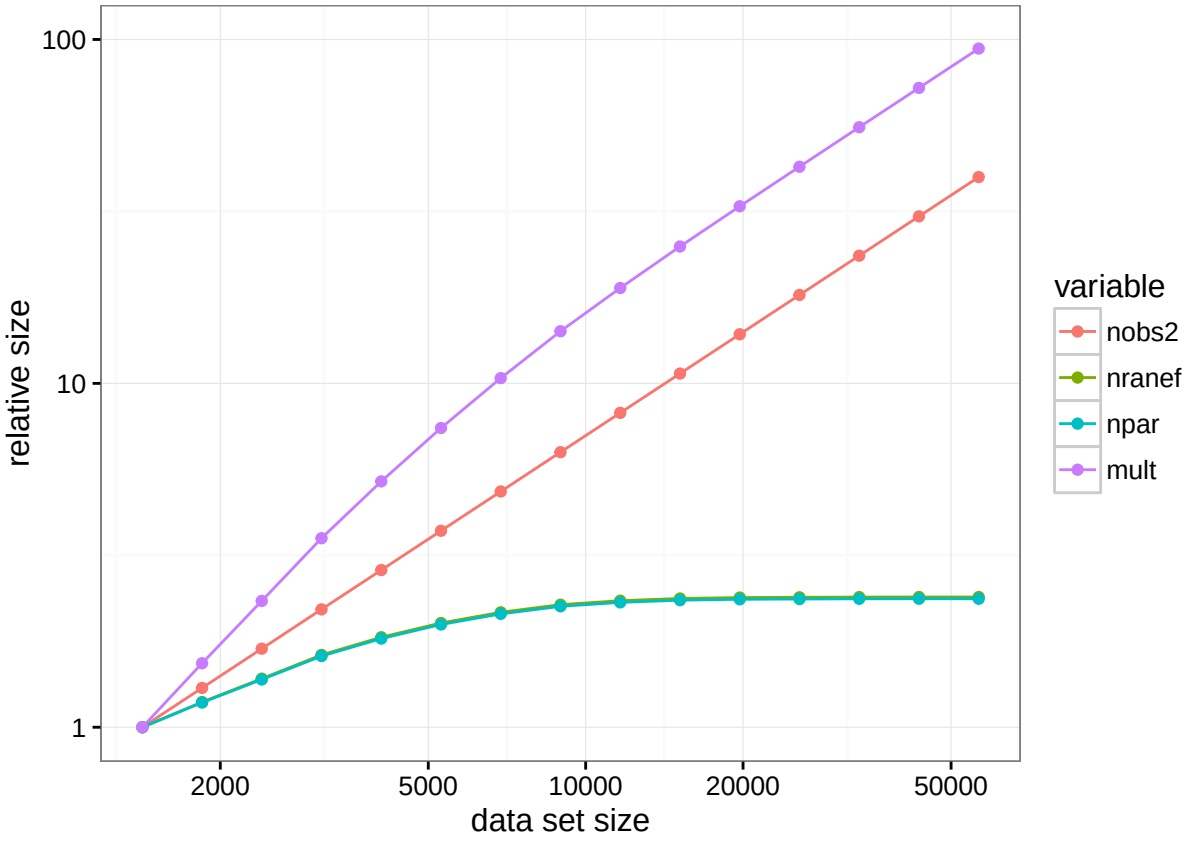
`glmmTMB` *without* optimization scales quadratically all the rest scale as ~ nobs^0.75. `glmmTMB` optimized is best (140 seconds), `glmer` is best with nloptr (~5 mins for full data set), second-best with bobyqa (~12 minutes? – not actually run), worst with default (~30 minutes?).

Scaling coefficients:

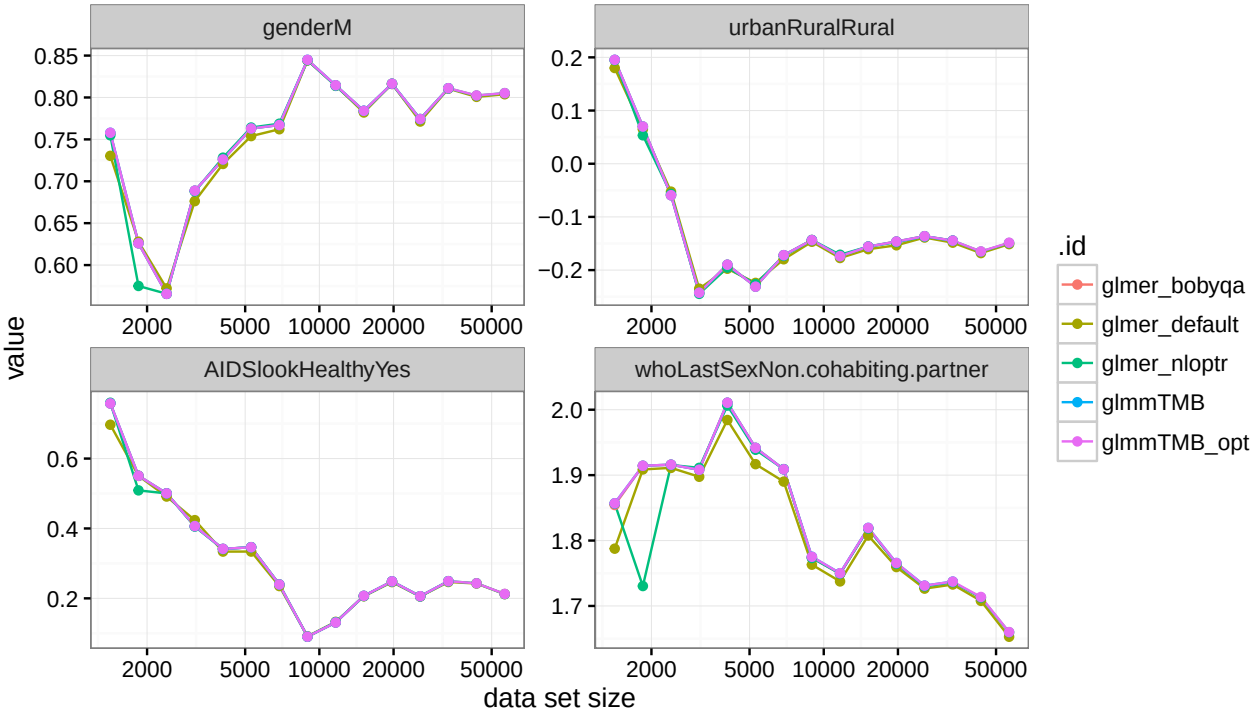
```
coef(lmList(log(elapsed)~log(nobs)|.id,data=times))[,2]
```

```
## [1] 0.709 0.753 0.809 2.119 0.936
```

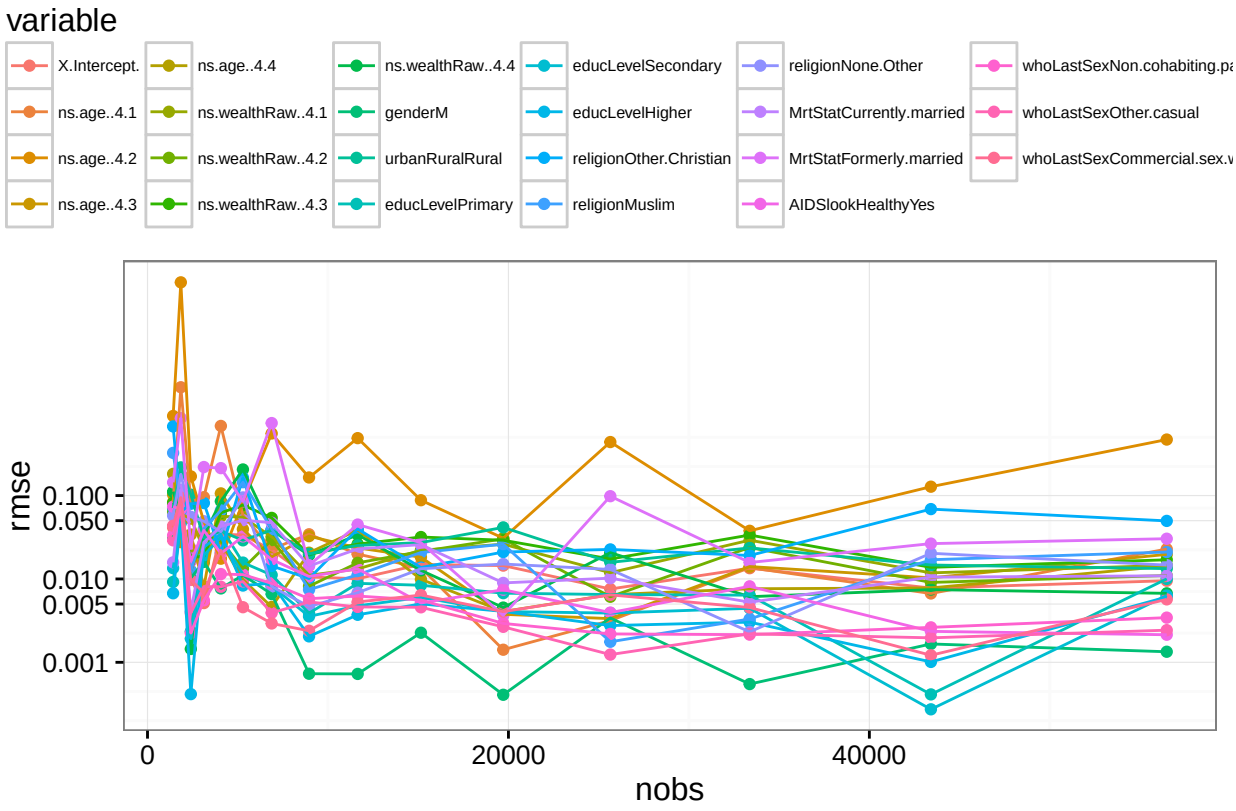
Number of groups



Variation in results: subset of parameters

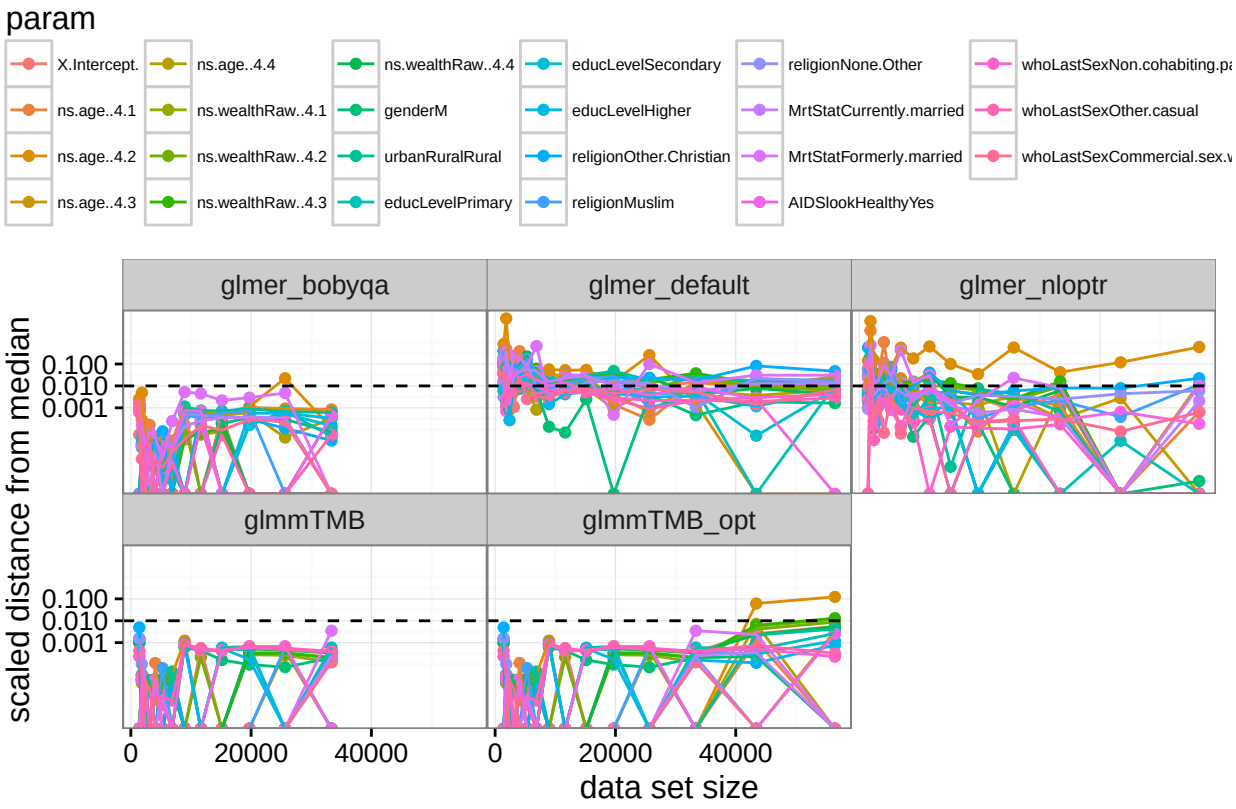


Overall variation in parameters



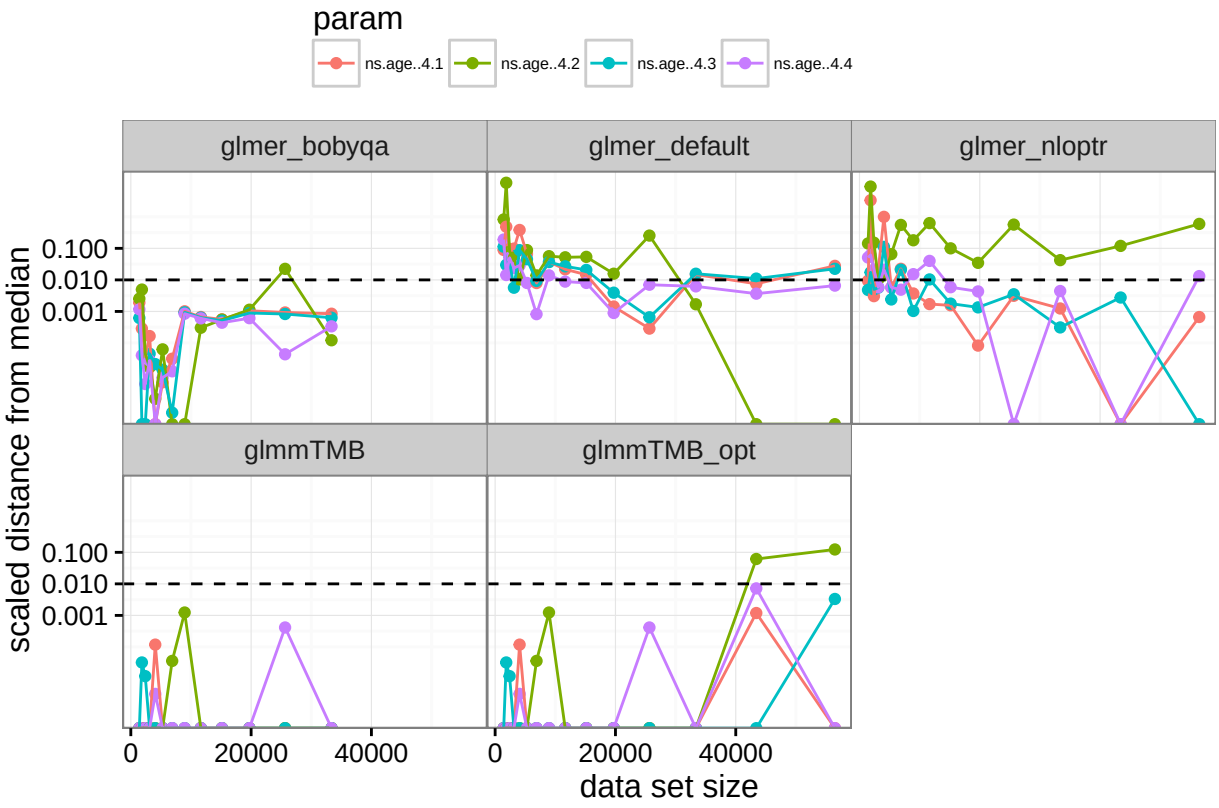
Deviation from median est by param/method

```
## Warning: Removed 92 rows containing missing values (geom_point).
```




```
ggplot_pd2 %+% subset(parmsdiff2_m,
                      grepl("^ns\\.age",param))
```

```
## Warning: Removed 16 rows containing missing values (geom_point).
```

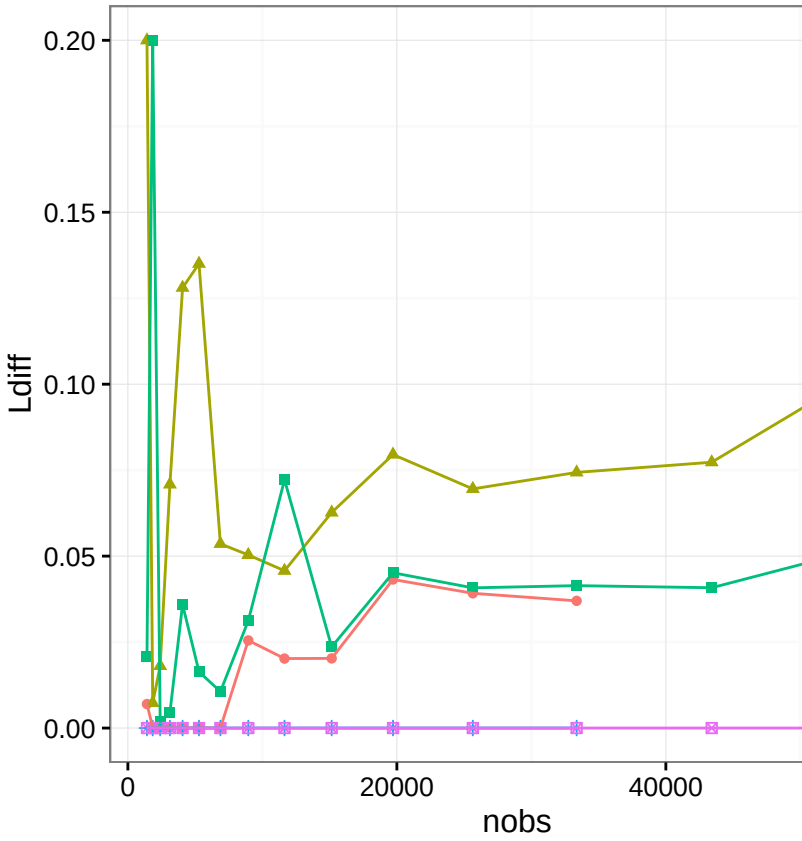


Although the difference seems large (10%), the actual difference is fairly small in absolute terms, and relative to the confidence intervals:

```
##      nobs    variable glmer_bobyqa glmer_default glmer_nloptr glmmTMB
## 325 56479 ns.age..4.2           NA       -0.0172      -0.0274      NA
##     glmmTMB_opt
## 325      -0.015
```

```
##              .id  2.5 % 97.5 %        .rownames
## 3    glmmTMB_opt -0.199  0.169 cond.ns(age, 4)2
## 29  glmer_nloptr -0.210  0.156      ns(age, 4)2
## 55 glmer_default -0.200  0.166      ns(age, 4)2
```

AICs



And in any case, the AICs are smallest for glmmTMB . . .

Conclusion: we should go ahead with glmmTMB. Only problem: I need to write `anova()` and `drop1()` methods!