Specifying Contrasts for linear model

At the moment my motivation for making this available is so that someone can tell me if I'm doing what I want to do.

Get `tidyverse` for data wrangling and `lme4` for modelling and load some data that I prepared earlier.

```
library(tidyverse)
library(lme4)

load("rti_narrative_data.Rda")
```

This loads a data frame called `Df` that looks like this:

```
##      Subno     T1234    Condition      score          cond_by_task
##  2      :  4   T1:161   comp:500   Min.   : 0.000   T1.comp:125
##  3      :  4   T2:161   int :144   1st Qu.: 4.000   T2.comp:125
##  5      :  4   T3:161              Median : 6.000   T3.comp:125
##  7      :  4   T4:161              Mean   : 5.571   T4.comp:125
##  8      :  4                       3rd Qu.: 8.000   T1.int : 36
##  9      :  4                       Max.   :10.000   T2.int : 36
##  (Other):620                       NA's   :5        (Other): 72
```

`T1234` is a factor representing tests at four different time points.

`Condition` comprises two groups, `comp` and `int`

I am constructing comparisons as follows.

Create a "flat" single factor representing all cells in the design.

```
Df <- Df %>%
  mutate(cond_by_task = interaction(T1234,Condition)) %>%
  filter(complete.cases(.))
summary(Df$cond_by_task)
```

```
## T1.comp T2.comp T3.comp T4.comp  T1.int  T2.int  T3.int  T4.int
##     125     125     125     121      36      36      36      35
```

Then define some contrasts:

Main effect of group

```
cond <- cbind(c(-1,-1,-1,-1,1,1,1,1))
colnames(cond) <- c(' group')
```

Main effects of time with separate contrasts giving slope, averaged across groups, between T1 and T2, T2 and T3, and T3 and T4.

```
time <- cbind(rep(c(-1,1,0,0),2),
              rep(c(0,-1,1,0),2),
              rep(c(0,0,-1,1),2))
colnames(time) <- c(' T12',' T23',' T34')
```

Interaction effects. This asks whether there the slopes for the two groups differ, looking separately at slopes between T1 and T2, T2 and T3, and T3 and T4.

```
inter <- cbind(c(1,-1,0,0,-1,1,0,0),
               c(0,1,-1,0,0,-1,1,0),
               c(0,0,1,-1,0,0,-1,1))
colnames(inter) <- c(' T12:group',' T23:group',' T34:group')
```

Then put these all together and assign them to the `cond_by_task` factor.

```
conts <- cbind(cond,time,inter)
contrasts(Df$cond_by_task, how.many = ncol(conts)) <- conts
contrasts(Df$cond_by_task)
```

```
##          group  T12  T23  T34  T12:group  T23:group  T34:group
## T1.comp     -1   -1    0    0          1          0          0
## T2.comp     -1    1   -1    0         -1          1          0
## T3.comp     -1    0    1   -1          0         -1          1
## T4.comp     -1    0    0    1          0          0         -1
## T1.int       1   -1    0    0         -1          0          0
## T2.int       1    1   -1    0          1         -1          0
## T3.int       1    0    1   -1          0          1         -1
## T4.int       1    0    0    1          0          0          1
```

You don't need an intercept because lmer gives you it for free (and won't allow you to use your own contrast but specify no intercept in the model).

Then run the model and get the coefficients.

```
m <- lmer(score ~ cond_by_task + (1|Subno),
          data = Df)

summary(m)$coef
```

```
##                          Estimate Std. Error   t value
## (Intercept)             5.0411258  0.1375121 36.659499
## cond_by_task group     -0.9775574  0.1375121 -7.108881
## cond_by_task T12        2.8194592  0.1346592 20.937743
## cond_by_task T23        3.3848072  0.1559507 21.704337
## cond_by_task T34        1.7137108  0.1362447 12.578185
## cond_by_task T12:group  0.1607760  0.1346592  1.193947
## cond_by_task T23:group  0.6034408  0.1559507  3.869433
## cond_by_task T34:group  0.3936612  0.1362447  2.889370
```

So, the question is: Is this set of contrasts giving me what I've claimed it's giving me? If not, what am I doing wrong?

One way of answering this question - of checking that the contrasts that I've specified really do test the statistical model that I think they are testing, is to separate out the contrasts and get model fits for each, then look to see if the fits for these models make sense. I'm going to do this without the random by-subject effect, because this makes things clearer (because subjects are only in one condition, so if you exclude a condition factor but include by-subject effects, then this effectively reintroduces the condition effect).

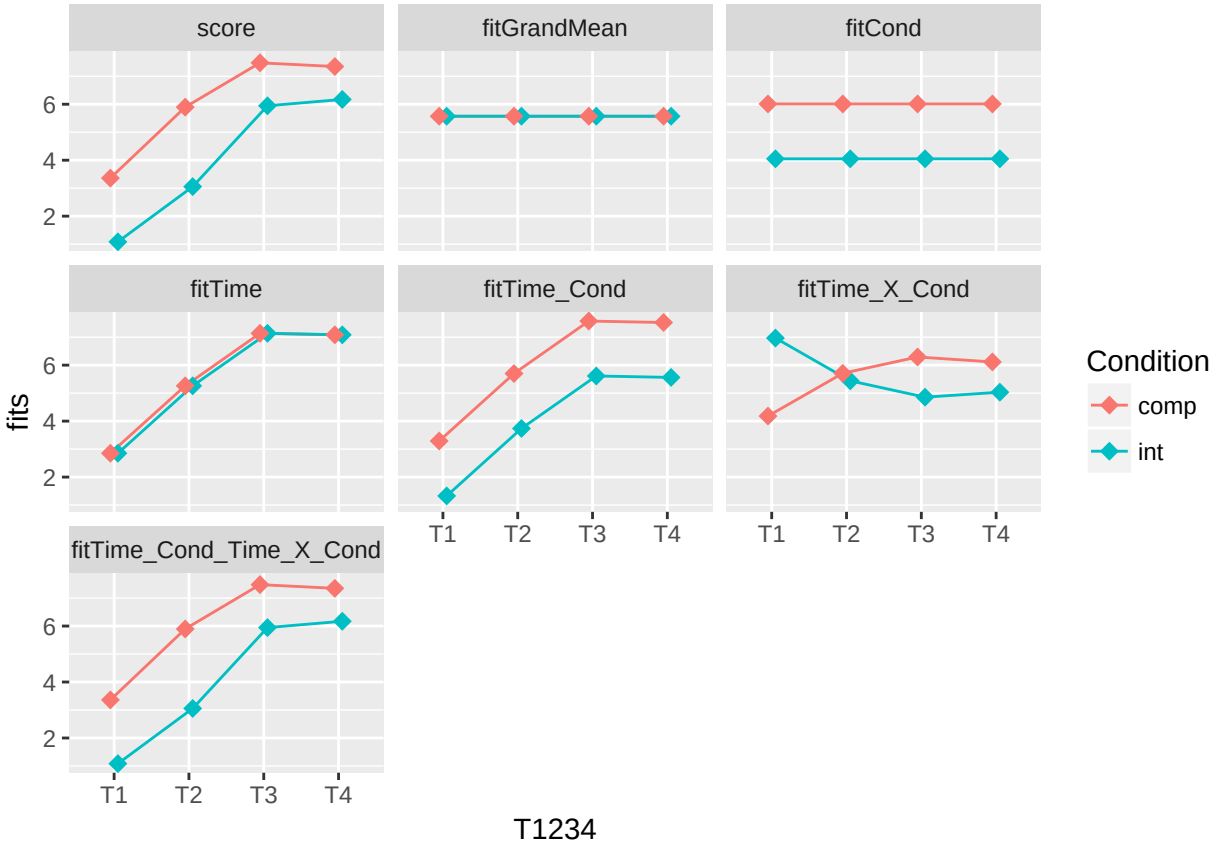
So, fit some models and use them to predict scores, and save these into the data frame: `Df$fit = fitted(mymodel)`.

```
# grand mean
m <-lm(score ~ 1, data = Df)
Df$fitGrandMean = fitted(m)

# just main effect of condition (group)
conts <- cond
contrasts(Df$cond_by_task, how.many = ncol(conts)) = conts
```

```
m = lm(score ~ cond_by_task, data = Df)
Df$fitCond = fitted(m)

# just main effect of time
conts <- time
contrasts(Df$cond_by_task, how.many = ncol(conts)) = conts
m = lm(score ~ cond_by_task, data = Df)
Df$fitTime = fitted(m)

# main effects of cond and time
conts = cbind(time,cond)
contrasts(Df$cond_by_task, how.many = ncol(conts)) = conts
m = lm(score ~ cond_by_task, data = Df)
Df$fitTime_Cond = fitted(m)

# just interaction
conts = inter
contrasts(Df$cond_by_task, how.many = ncol(conts)) = conts
m = lm(score ~ cond_by_task, data = Df)
Df$fitTime_X_Cond = fitted(m)

# main effects and interaction
conts = cbind(time,cond,inter)
contrasts(Df$cond_by_task, how.many = ncol(conts)) = conts
m = lm(score ~ cond_by_task, data = Df)
Df$fitTime_Cond_Time_X_Cond = fitted(m)
```

And then have a look at them.

```
pd = position_dodge(.2)
Df %>% gather(model, fits, -Subno, -T1234, -Condition, -cond_by_task) %>%
  mutate(model = factor(model,levels = unique(model))) %>%
  ggplot(aes(y = fits, x = T1234, colour = Condition)) +
  geom_line(aes(group = Condition),
            stat = 'summary', fun.y = "mean", na.rm = T,
            position = pd) +
  geom_point(stat = 'summary', fun.y = "mean", na.rm = T,
             size = 3, shape = 18,
             position = pd) +
  facet_wrap(~model)
```



Score is the raw score. So, this suggests that the contrasts are doing what they are meant to be doing.