

Output from choccake.R

```
> # choccake.R
>
> library(lsmmeans)
> library(lme4)
> library(car)
Warning message:
package 'car' was built under R version 3.0.2
> #library(nlme)
> #library(Matrix)
> #library(pbkrtest)
>
>
> # get the data
> cake<-read.table(file="choccake.dat",header=F,
+                 colClasses=c("factor","factor","factor","numeric"))
> names(cake)<-c("recipe","replicat","baketemp","angle")
> # the name "break" has problems and cannot be used in R
>
> # create a replicat factor (called allreplicat below)
> # that has a unique level for each replicat within each recipe.
> cake <- within(cake, batch <- factor(replicat:recipe))
> head(cake)
  recipe replicat baketemp angle batch
1      1         1         1    42   1:1
2      1         1         2    46   1:1
3      1         1         3    47   1:1
4      1         1         4    39   1:1
5      1         1         5    53   1:1
6      1         1         6    42   1:1
> is.numeric(cake$recipe)
[1] FALSE
> is.numeric(cake$angle)
[1] TRUE
> is.factor(cake$batch)
[1] TRUE
>
> # the aov function gives the traditoinal type III anova F tests. It actually
> # fits two separate models, one at each level of experimental unit
> m0 <-aov(angle ~ recipe*baketemp + Error(batch/baketemp), data=cake)
> summary(m0)
```

Error: batch

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
recipe	2	135	67.54	0.249	0.781
Residuals	42	11403	271.49		

Error: batch:baketemp

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
baketemp	5	2100	420.1	20.520	<2e-16 ***
recipe:baketemp	10	206	20.6	1.006	0.439
Residuals	210	4299	20.5		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

>

```

> # the lmer will fit the model as a single mixed effect model and the anova
> # function will give the standard F tests.
> m1 <- lmer(angle ~ recipe*baketemp + (1|batch), data=cake)
> anova(m1)
Analysis of Variance Table

            Df  Sum Sq Mean Sq F value
recipe      2   10.19    5.09  0.2488
baketemp    5 2100.30  420.06 20.5199
recipe:baketemp 10  205.98   20.60  1.0062
>
> # Now for contrasts. Construct the contrasts of interest
> c1<-c(1,1,-2) # recipe contrast 1: normal vs extra sugar
> c2<-c(1,-1,0) # recipe contrast 2: choc temp
> c3<-c(-5,-3,-1,1,3,5) # linear in baketemp
> c41<-c(5,-1,-4,-4,-1,5) # quadratic in baketemp
> c42<-c(-5,7,4,-4,-7,5) # cubic in baketemp
> c43<-c(1,-3,2,2,-3,1) #quartic in baketemp
> c44<-c(-1,5,-10,10,-5,1) #quintic in baketemp
> c5 <- c(1,0,0,-1,rep(0,14)) # mu11-mu12
> c6 <- c(1,-1,rep(0,16)) # mu11-m21
>
>
> lsmeans(m1, specs=list(lsm1~recipe, lsm2~baketemp, lsm3~recipe:baketemp),
+         contr=list(lsm1=list(normal_vs_extra.sugar=c1,
+                               chocolate_temp=c2),
+                   lsm2=list(linear_baketemp=c3),
+                   lsm3=list(mu11_mu12=c5,
+                               mu11_mu21=c6)
+         ))
$`recipe lsmeans`
  recipe  lsmean      SE df lower.CL upper.CL
    1  33.12222 1.736841 42  29.61714 36.62731
    2  31.64444 1.736841 42  28.13936 35.14953
    3  31.60000 1.736841 42  28.09491 35.10509

$`recipe lsm1`
              estimate      SE df t.ratio p.value
normal_vs_extra.sugar 1.566667 4.254374 42  0.36825 0.71454
chocolate_temp        1.477778 2.456264 42  0.60164 0.55065
p values are not adjusted

$`baketemp lsmeans`
  baketemp  lsmean      SE      df lower.CL upper.CL
    1  27.97778 1.176703 77.43643 25.63487 30.32068
    2  29.95556 1.176703 77.43643 27.61265 32.29846
    3  31.42222 1.176703 77.43643 29.07932 33.76513
    4  32.17778 1.176703 77.43643 29.83487 34.52068
    5  35.84444 1.176703 77.43643 33.50154 38.18735
    6  35.35556 1.176703 77.43643 33.01265 37.69846

$`baketemp lsm2`
              estimate      SE df t.ratio p.value
linear_baketemp 55.31111 5.64301 210  9.8017  0
p values are not adjusted

$`recipe:baketemp lsmeans`
  recipe baketemp  lsmean      SE      df lower.CL upper.CL

```

1	1	29.13333	2.038109	77.43643	25.07531	33.19136
2	1	26.86667	2.038109	77.43643	22.80864	30.92469
3	1	27.93333	2.038109	77.43643	23.87531	31.99136
1	2	31.53333	2.038109	77.43643	27.47531	35.59136
2	2	29.40000	2.038109	77.43643	25.34197	33.45803
3	2	28.93333	2.038109	77.43643	24.87531	32.99136
1	3	30.80000	2.038109	77.43643	26.74197	34.85803
2	3	31.73333	2.038109	77.43643	27.67531	35.79136
3	3	31.73333	2.038109	77.43643	27.67531	35.79136
1	4	33.53333	2.038109	77.43643	29.47531	37.59136
2	4	32.13333	2.038109	77.43643	28.07531	36.19136
3	4	30.86667	2.038109	77.43643	26.80864	34.92469
1	5	38.66667	2.038109	77.43643	34.60864	42.72469
2	5	34.46667	2.038109	77.43643	30.40864	38.52469
3	5	34.40000	2.038109	77.43643	30.34197	38.45803
1	6	35.06667	2.038109	77.43643	31.00864	39.12469
2	6	35.26667	2.038109	77.43643	31.20864	39.32469
3	6	35.73333	2.038109	77.43643	31.67531	39.79136

```
$`recipe:baketemp lsm3`
```

	estimate	SE	df	t.ratio	p.value
mu12	-2.400000	1.652104	210.00000	-1.45269	0.14780
mu21	2.266667	2.882321	77.43643	0.78640	0.43403

p values are not adjusted

Warning messages:

```
1: In lsmeans(m1, specs = list(lsm1 ~ recipe, lsm2 ~ baketemp, lsm3 ~ :
  lsmeans of recipe may be misleading due to interaction with other predictor(s)
2: In lsmeans(m1, specs = list(lsm1 ~ recipe, lsm2 ~ baketemp, lsm3 ~ :
  lsmeans of baketemp may be misleading due to interaction with other
predictor(s)
```

```
>
>
> K1<-rbind("nonlinear baketemp I"=c41,"nonlinear baketemp II"=c42,
+          "nonlinear baketemp III"=c43,"nonlinear baketemp IV"=c44)
> # It seems like the following line (commented out) should work to get the
> # test of nonlinearity in baking temperature, but it does not
> #summary(glht(m1, linfct=mcp(baketemp=K1)), test=Ftest())
>
> # Instead, I am going to use the linearHypothesis function. To make
> # this easier, I will refit the model with a cell means parameterization
> # for the treatments. Then create the contrast matrix that will test
> # nonlinearity in baking temperature (averaged over recipe) if applied to the
> # treatment means. That matrix is K1a below.
> m1a <- lmer(angle ~ -1+recipe:baketemp + (1|batch), data=cake)
> m1a
Linear mixed model fit by REML ['lmerMod']
Formula: angle ~ -1 + recipe:baketemp + (1 | batch)
Data: cake
REML criterion at convergence: 1633.246
Random effects:
 Groups   Name          Std.Dev.
 batch    (Intercept)  6.468
 Residual                    4.524
Number of obs: 270, groups: batch, 45
Fixed Effects:
recipe1:baketemp1  recipe2:baketemp1  recipe3:baketemp1  recipe1:baketemp2
```

```

                29.13                26.87                27.93                31.53
recipe2:baketemp2 recipe3:baketemp2 recipe1:baketemp3 recipe2:baketemp3
                29.40                28.93                30.80                31.73
recipe3:baketemp3 recipe1:baketemp4 recipe2:baketemp4 recipe3:baketemp4
                31.73                33.53                32.13                30.87
recipe1:baketemp5 recipe2:baketemp5 recipe3:baketemp5 recipe1:baketemp6
                38.67                34.47                34.40                35.07
recipe2:baketemp6 recipe3:baketemp6
                35.27                35.73

```

```

>
> (K1a <- K1%x%matrix(1,nrow=1,ncol=3))
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,]    5    5    5   -1   -1   -1   -4   -4   -4   -4   -4   -4   -1   -1
[2,]   -5   -5   -5    7    7    7    4    4    4   -4   -4   -4   -7   -7
[3,]    1    1    1   -3   -3   -3    2    2    2    2    2    2    2   -3   -3
[4,]   -1   -1   -1    5    5    5  -10  -10  -10   10   10   10   -5   -5
      [,15] [,16] [,17] [,18]
[1,]    -1     5     5     5
[2,]    -7     5     5     5
[3,]    -3     1     1     1
[4,]    -5     1     1     1
> linearHypothesis(m1a,K1a,test="F") # gives nonlinearity in baking temp test
Linear hypothesis test

```

Hypothesis:

```

5 recipe1:baketemp1 + 5 recipe2:baketemp1 + 5 recipe3:baketemp1 -
recipe1:baketemp2 - recipe2:baketemp2 - recipe3:baketemp2 - 4 recipe1:baketemp3 -
4 recipe2:baketemp3 - 4 recipe3:baketemp3 - 4 recipe1:baketemp4 - 4
recipe2:baketemp4 - 4 recipe3:baketemp4 - recipe1:baketemp5 - recipe2:baketemp5 -
recipe3:baketemp5 + 5 recipe1:baketemp6 + 5 recipe2:baketemp6 + 5
recipe3:baketemp6 = 0
- 5 recipe1:baketemp1 - 5 recipe2:baketemp1 - 5 recipe3:baketemp1 + 7
recipe1:baketemp2 + 7 recipe2:baketemp2 + 7 recipe3:baketemp2 + 4
recipe1:baketemp3 + 4 recipe2:baketemp3 + 4 recipe3:baketemp3 - 4
recipe1:baketemp4 - 4 recipe2:baketemp4 - 4 recipe3:baketemp4 - 7
recipe1:baketemp5 - 7 recipe2:baketemp5 - 7 recipe3:baketemp5 + 5
recipe1:baketemp6 + 5 recipe2:baketemp6 + 5 recipe3:baketemp6 = 0
recipe1:baketemp1 + recipe2:baketemp1 + recipe3:baketemp1 - 3 recipe1:baketemp2
- 3 recipe2:baketemp2 - 3 recipe3:baketemp2 + 2 recipe1:baketemp3 + 2
recipe2:baketemp3 + 2 recipe3:baketemp3 + 2 recipe1:baketemp4 + 2
recipe2:baketemp4 + 2 recipe3:baketemp4 - 3 recipe1:baketemp5 - 3
recipe2:baketemp5 - 3 recipe3:baketemp5 + recipe1:baketemp6 + recipe2:baketemp6
+ recipe3:baketemp6 = 0
- recipe1:baketemp1 - recipe2:baketemp1 - recipe3:baketemp1 + 5
recipe1:baketemp2 + 5 recipe2:baketemp2 + 5 recipe3:baketemp2 - 10
recipe1:baketemp3 - 10 recipe2:baketemp3 - 10 recipe3:baketemp3 + 10
recipe1:baketemp4 + 10 recipe2:baketemp4 + 10 recipe3:baketemp4 - 5
recipe1:baketemp5 - 5 recipe2:baketemp5 - 5 recipe3:baketemp5 +
recipe1:baketemp6 + recipe2:baketemp6 + recipe3:baketemp6 = 0

```

Model 1: restricted model

Model 2: angle ~ -1 + recipe:baketemp + (1 | batch)

```

      Res.Df Df      F Pr(>F)
1         214
2         210  4  1.6315 0.1675
>

```

```
> # Finally, get the profile plot
> lsmip(m1, recipe~baketemp)
```

