

English Nominal Recursion Study, Part 3:

Trying to test the interaction of Adult-Child by item (Item is fixed)

```
> rm(list=ls()); options(scipen=999) # To avoid scientific notation
> # Install packages if necessary. Only need to do this once.
> # install.packages("lme4")
> # install.packages("car")
> # Load packages -- do this every time
> library(lme4) # For lmer function
Loading required package: Matrix
> library(car) # For F-tests with p-values, Wald chi-squared tests
>
> # Read data into a data frame
> rdata =
read.table("http://www.utstat.toronto.edu/~brunner/workshops/mixed/Recursion.data.txt",
header=T)
> dim(rdata)
[1] 2100 18
> # Eliminate Items e1 and e6
> rdata = subset(rdata, Item != 'e1'); rdata = subset(rdata, Item != 'e6')
> dim(rdata)
[1] 1932 18
> head(rdata); attach(rdata)
```

	Participant	AgeinMonths	Group	Agegroup	Condition	Item	StrictTarget	AllTargets
1	1	48.5	Child	4yos	A	a1a	1	1
2	1	48.5	Child	4yos	A	a1b	0	0
3	1	48.5	Child	4yos	A	a2	0	0
4	1	48.5	Child	4yos	A	a3	0	0
5	1	48.5	Child	4yos	A	a4	0	0
6	1	48.5	Child	4yos	A	a5	NA	NA

```
> head(rdata); attach(rdata)
```

	Years	Months	Days	KABC_Hand	CELF_SS	CELF_WS	CELF_EV	CELF_Sent	PPVT	KBIT
1	4	0	14	4	16	13	22	21	92	NA
2	4	0	14	4	16	13	22	21	92	NA
3	4	0	14	4	16	13	22	21	92	NA
4	4	0	14	4	16	13	22	21	92	NA
5	4	0	14	4	16	13	22	21	92	NA
6	4	0	14	4	16	13	22	21	92	NA

```
> table(Item)
```

Item	a1a	a1b	a2	a3	a4	a5	a6	b1	b2	b3	b4	b5	b6	c1	c2	c3	c4	c5	c6	e1
	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	0
	e2	e3	e4	e5	e6															
	84	84	84	84	0															

```
> Item = factor(Item); table(Item); length(table(Item))
```

Item	a1a	a1b	a2	a3	a4	a5	a6	b1	b2	b3	b4	b5	b6	c1	c2	c3	c4	c5	c6	e2
	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84
	e3	e4	e5																	
	84	84	84																	

```
[1] 23
> # Item has 23 levels now, okay.
> contrasts(Group) = contr.sum; contrasts(Condition) = contr.sum
> contrasts(Item) = contr.sum
```

```

>
> table4 = table(AllTargets,Item,Group); table4
, , Group = Adult

      Item
AllTargets ala alb a2 a3 a4 a5 a6 b1 b2 b3 b4 b5 b6 c1 c2 c3 c4 c5 c6 e2 e3 e4 e5
0      1    1  3  2  7  6  5  3  6  3  5  3  2 12  7  8 10  7  3  9 10  7 10
1     12   11 10 11  6  7  8 10  7 10  8 10 11  1  6  5  3  6 10  4  2  6  3

, , Group = Child

      Item
AllTargets ala alb a2 a3 a4 a5 a6 b1 b2 b3 b4 b5 b6 c1 c2 c3 c4 c5 c6 e2 e3 e4 e5
0     27   46 62 61 67 46 53 37 63 47 60 37 55 67 59 66 64 55 65 63 56 66 69
1     44   25  9  9  4 23 16 34  8 24 11 34 16  4 12  5  7 16  6  8 14  5  2

> model4 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial, nAGQ=1)
Warning messages:
1: In (function (fn, par, lower = rep.int(-Inf, n), upper = rep.int(Inf, :
failure to converge in 10000 evaluations
2: In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
Model failed to converge with max|grad| = 0.0254562 (tol = 0.001, component 1)
> # Didn't converge. Test anyway for comparison
> Anova(model4, type="III") # Each effect controlled for all others
Analysis of Deviance Table (Type III Wald chisquare tests)

Response: AllTargets
            Chisq Df                Pr(>Chisq)
(Intercept) 15.196  1                0.00009692635 ***
Group        30.149  1                0.000000004001 ***
Item        142.811 22 < 0.0000000000000000022 ***
Group:Item   40.040 22                0.0107 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> # p-value for Group * Item = 0.0107
>
> model4 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial, nAGQ=0)
> Anova(model4, type="III") # Each effect controlled for all others
Analysis of Deviance Table (Type III Wald chisquare tests)

Response: AllTargets
            Chisq Df                Pr(>Chisq)
(Intercept) 13.695  1                0.0002151 ***
Group        27.544  1                0.0000001536 ***
Item        138.767 22 < 0.0000000000000000022 ***
Group:Item   38.309 22                0.0169121 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> # P-value for interaction = 0.0169121
>
> naive4 = glm(AllTargets ~ Group*Item, family=binomial) # Ignoring repeated
measures
> Anova(naive4, type="III") # Warning about fitted values near zero
Analysis of Deviance Table (Type III tests)

Response: AllTargets
            LR Chisq Df                Pr(>Chisq)
Group        33492  1 <0.000000000000000002 ***
Item         175 22 <0.000000000000000002 ***
Group:Item    39 22                0.0148 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred

```

```

> # P-value for interaction = 0.0148
>
> # Several values of nAGQ failed, and also these.
> # model4 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial, nAGQ=1,
> #               contrasts = list(Group="contr.treatment", Item="contr.treatment"))
> # model4 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial, nAGQ=1,
> #               contrasts = list(Group="contr.poly", Item="contr.poly"))
> # model4 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial, nAGQ=1,
> #               contrasts = list(Group="contr.helmert", Item="contr.helmert"))
>
> # There is some indication of an interaction between Group and Item, but I can't
> # get the model to actually fit. Rather than pursuing it, limit the analysis to just
> # kids.
>

#####
# So here's a collection of potential ways to deal with convergence problems.
# They did not accomplish anything for me with this data set, but it's nice to know
# how to do them.
#####

# The verbose option lets you see the progress of maximum likelihood.
glmer(AllTargets ~ Item + (1 | Participant), family=binomial, nAGQ=1, verbose=T)

# Choose a set of starting values for the search.
naive = glm(AllTargets ~ Item, family=binomial)
logregbetahats = naive$coef
summary(
  glmer(AllTargets ~ Item + (1 | Participant), family=binomial, nAGQ=1, verbose=T,
  start = list(fixef=logregbetahats, theta=1))
) # End of summary
# Failed to converge after 10,000 iterations. This search converged when I let
# glmer choose the starting values.

# My brilliant idea of using a nAGQ=0 run as a source of starting values happens to
# be the default.

# Try a different optimizer. The default is "bobyqa"
model44 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial,
  nAGQ=1, control = glmerControl(optimizer = "Nelder_Mead"))

# Specify maximum number of iterations as well as Nelder_Mead
model44 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial,
  nAGQ=1, control = glmerControl(optimizer = "Nelder_Mead",
  optCtrl = list(maxfun=100)) # Worked!

# Try 100,000 iterations instead of 10,000.
model44 = glmer(AllTargets ~ Group*Item + (1 | Participant), family=binomial,
  nAGQ=1, control = glmerControl(optCtrl = list(maxfun=100000))

# What worked for me (sometimes) was trying an equivalent model.

```