$\frac{1}{2}$

1 head-entity-ratio

issue 6 のイノシシ分析をここで行います。まずは 4900 から 5300 までを分析の対象とします。 ratio のブランチで頑張ってください。

4900-5100

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  -3.6334     0.2337 -15.547
npi           0.2477     0.1423   1.741   0.08141 .
aff           0.7828     0.1423   5.501   5.248e-08 ***
npi:aff       0.2672     0.1424   1.876   0.06035 .
```

5100-5300

```
Fixed effects:
            Estimate Std. Error t value
(Intercept) -2.88496    0.28026 -10.294
npi          0.06119    0.15523   0.394   0.6927
aff          0.76956    0.15522   4.958   8.856e-07 ***
npi:aff      0.31856    0.15537   2.050   0.04015 *
```

1.1 graph

```
#グラフの作成
#ディレクトリの確認から

getwd()
setwd("/home/kishiyama/home/thesis/npi-thesis/result/main")

#ライブラリーとデータの読み込み
library(ggplot2)
library(reshape)
data_all <- read.csv("./csv/output.csv", header =T)
#head(dataAll,100)

#digital 化
#そのままではグラフ化ができない
#グラフの全体（土台）の時間軸の幅をここで決める。
#（複合名詞全体は約 6000ms, 一文字は約 3000ms）
#comp_noun でまず 20~6000ms を 20ms 単位で bin のリストを作っておいて、
#それをカラム数にして dataAll と同じ行数で行列 bindata を作って後で dataAll の右側につっくけます。
```

```
# TODO: comp_noun は意味不明

# 最初の呈示    : 2000 ms

# n1          : 800 ms
# NPI         : 500 ms
# break       : 200 ms

# v1          : 600 ms

# 本来なら、ここからが注目する領域。(4100~4700)
# モーラからの推定。３モーラで６００ ms。
# ているの「て」は 200 ｍｓと仮定。
# すると、4100 + 200(反射) + 200 からスタート。つまり、4500 時点からスタート。
# そこから リージョンの 400 ms を検討するから、4500 -> 4900 となる。
# なお、5700 あたりが名詞句のヘッド。やっぱり東大生はスゴい。
# 5900 までの１秒
# ちゃんと正解に辿り着いている。
# しかも、NPI と NEG だと正解へのたどり着き具合が有意に低い。
# まずか先生の仮説をサポート。
# 全体の長さは 5500 として考えた方が良さそう。

# 文節は短い方がいい？
# aff         : 600 ms <- (+200)

# n1          : 1000 ms
# spillover   : 200 ms
# comp_noun <- seq(4100,4900,20)
# comp_noun <- seq(4100,5900,20)
# 全体が５秒だと仮定
# comp_noun <- seq(2000,7000,20)
# これで０から推定できる。

span_begin <- 4900
span_end <- 5300

comp_noun <- seq(span_begin,span_end,20)

#comp_noun <- seq(4100,6100,20)
#comp_noun <- seq(4100,4900,20)
#comp_noun <- seq(2420,3540,20)
binary_data <- matrix(span_begin, nrow = nrow(data_all), ncol = length(comp_noun))
colnames(binary_data) <- comp_noun

# グラフの開始時点を揃える基準となる音声ファイルの onset はここで決めます。
# 時間窓 20ms で見ていき、その区間を見続けていれば 1, 0 見ていなければ 0
# （MF01 では onset3 が critical region です）
```

```
for (i in 1:length(comp_noun)){
    binary_data[,i] <- ifelse((data_all$GazeStart < (span_begin + i*20) & data_all$GazeEnd > (span_b
        1,
        0)}

# Fixation1: 41~185ms
# Fixation2: 188~225ms

# ＜調整例＞
# 開始始時点が [0]
#   bindata[,i] <- ifelse((dataAll$GazeStart < (0 + i*20) & dataAll$GazeEnd > (0 +i*20)), 1, 0)
# 開始始時点が [onset2]
#   bindata[,i] <- ifelse((dataAll$GazeStart < (dataAll$onset2 + i*20) & dataAll$GazeEnd > (dataAll$

# 意味： この行の GazeStart~GazeEnd の範囲の中でその bin の数値（例：40ms）が入ってれば、1 と表示する。
# [onset2 + i*20] にすれば、onset2 が開始時点になって、それ以前の時間帯は見ないことになる。
# 例：i=0 であれば、一行目からは onset2 のその数値を GazeStart~GazeEnd に当てはめることになる

gr <- cbind(data_all, as.data.frame(binary_data))
#dataall(trial 情報) と bindata（bin 注視情報）と合併

# 数値であった AOI を文字に転換する
# AOI == 0 means looks to marginal area.
# TODO: BackGround は存在しうる。画面外に Fixation すると、そうなる。
# ここ。ここでそれぞれのターゲットを変える。
# ただ、それは今ではない。
# 前にすすめる
gr$Target <- ifelse(gr$AOI == 1, as.character(gr$AOI1), "BackGround")
gr$Target <- ifelse(gr$AOI == 2, as.character(gr$AOI2), gr$Target)
gr$Target <- ifelse(gr$AOI == 3, as.character(gr$AOI3), gr$Target)
gr$Target <- ifelse(gr$AOI == 4, as.character(gr$AOI4), gr$Target)

#必要なコラムだけを取り出す
#participant,item, cond, AOI,Target,bin1500~3520
gr<- gr[,c(1, 9, 10, 15, ncol(gr), 16:(ncol(gr)-1))]

# melt time binaries into one column.
# gr を立てにする。 t20 の時に AOI2 に入っているのか。
gr2 <- melt(gr,id=c("ParticipantName", "ItemNo", "Condition", "AOI", "Target"))

gr2$variable <- as.numeric(as.character(gr2$variable))
#melt() かけたら、binaries が variable, 入っているか (1/0) が value になる。
#ERROR は一度出たが、今回はでなかった。

#わかりやすいため並べ替えます
gr2 <- gr2[order(gr2$ParticipantName, gr2$ItemNo),]
#ERROR
```

```
#gr2$itemBがNullになってる。$ItmeNoBに変更。

#aggregate for each trial (participants x items)
gr3 <-aggregate(
    gr2$value,
    by = list(gr2$ParticipantName, gr2$ItemNo, gr2$Condition,  gr2$AOI, gr2$Target, gr2$variable),
    FUN = sum,
    na.rm = TRUE)
#Tobiiの記録の中で、一定の期間であるAOIが視野に入ったら一カウントされます。
#逆にその期間に視野に入ってないbinの部分には0と表記され、AOIが多く重複されています
#この式は一つのbinの中に重複のないように、AOIをaggregate()で消します。
#違う言い方だとすると、違う区間で111が入ったら、同じ行にまとめられます。

colnames(gr3) = c("subj","item","cond", "AOI", "variable", "bin","value")
gr3$AOI <- NULL
# 必要がないコラムを削除

# そうすると、もう一回cast()でデータを開いて、TargetがNAのところを0に入れ替える

# ただし、最初から最後まで一回も見られたのないAOIはどうしてもカウントされません
# それがカウントされないと、比率の母数が正しく求められません
# （例：「6回提示の中にTargetが何回見られた」の中、「6回」が0だとしても正しく求められません）
gr.temp <- cast(gr3)
# cast()を実行すると、gr3の"variable"と"value"にしたがって自動的に分解してもらえます。ちょっと時間かかりま
# (例：variableの"CompetitorCompound"などが横になって、"value"がその下に)

# 一回も出たことない Target  がNAになっているはずです。それを0に書き換えます。
# backgroud はそのまま
# 全てのターゲットは一回は出現するため、ここはスキップ
# gr.temp$BackGround <- ifelse(is.na(gr.temp$BackGround),0,gr.temp$BackGround)

# やっぱり名前にオペレーターは入れてはならない
# N1_V1_t :
# N1_V1_f :
# N2_V1_t :
# N2_V1_f :

gr.temp$A <- ifelse(is.na(gr.temp$A),0,gr.temp$C)
gr.temp$B <- ifelse(is.na(gr.temp$B),0,gr.temp$B)
gr.temp$C <- ifelse(is.na(gr.temp$C),0,gr.temp$C)
gr.temp$D <- ifelse(is.na(gr.temp$D),0,gr.temp$D)

# 他に見たい組み合わせも簡単にできるようにしておきます
# gr.temp$Combined <- gr.temp$TargetCompound + gr.temp$CompetitorCompound
# gr.temp$IrrelevantCompound <- gr.temp$IrrelevantCompoundA + gr.temp$IrrelevantCompoundB

#  (t1) N1_V1_t (t2) N1_V1_f (t3) N2_V1_t (t4) N2_V1_f
```

```r
# 画像の方
gr.temp$t1 <- gr.temp$A
gr.temp$t2 <- gr.temp$B
gr.temp$t3 <- gr.temp$C
gr.temp$t4 <- gr.temp$D

# まずは a と b でやってみる。

a4crrct <- gr.temp[gr.temp$cond=="a",]
a4wrong <- gr.temp[gr.temp$cond=="a",]
b4crrct <- gr.temp[gr.temp$cond=="b",]
b4wrong <- gr.temp[gr.temp$cond=="b",]

a4crrct["cond"] <- "a4crrct"
a4wrong["cond"] <- "a4wrong"
b4crrct["cond"] <- "b4crrct"
b4wrong["cond"] <- "b4wrong"

gr.temp <- rbind(a4crrct, a4wrong, b4crrct, b4wrong)
gr.temp$head <- ifelse((gr.temp$cond == "a4crrct" | gr.temp$cond == "b4crrct"), gr.temp$C, gr.temp$D

#aggregate for graph (Use t1~t4)
#グラフ作成ためのデータフレーム
gra <- aggregate(
    gr.temp$head,
    by=list(gr.temp$bin, gr.temp$cond),
    mean)
colnames(gra) <- c("bin", "cond", "mean")

# a: NPI+AFFG

# gra$cond <- ifelse((gra$cond == 1), "NPI_t_AFF_t", gra$cond)
# gra$cond <- ifelse((gra$cond == 2), "NPI_f_AFF_t", gra$cond)
# gra$cond <- ifelse((gra$cond == 3), "NPI_t_AFF_f", gra$cond)
# gra$cond <- ifelse((gra$cond == 4), "NPI_f_AFF_f", gra$cond)

#グラフ作成の本番
g.Combined.Region6000 <- ggplot(data=gra, aes(x=bin, y=mean, colour = cond))+
  geom_line(aes(group=cond, color=cond)) +
  geom_point(aes(group=cond, color=cond)) +
  scale_x_continuous("Time") +
  scale_y_continuous(limits=c(0,0.3), name="Proportion of looks to the head entity") +
  scale_color_discrete("Condition") +
  theme(axis.title.y = element_text(size = 16)) +
  theme(axis.title.x = element_text(size = 20)) +
  theme(legend.title = element_text(size = 16)) +
  theme(legend.text = element_text(size = 20))+
```

```
  ggtitle("Proportion of gaze to N1  doing V1")

#必要に応じて変えます
#グラフ保存
ppi <- 300
#画質を screen 上きれいに映すように pixel=300ppi に設定
png("../png/head_entity_from_4900_to_5300.png", width=12*ppi, height=6*ppi, res=ppi)
#png("../png/target_region_from_4300_4900.png", width=12*ppi, height=6*ppi, res=ppi)
#png() で png 拡張子のグラフで外部保存します。 ppi に応じて長さと幅の設定は  12*6*ppi

# エラー: Continuous value supplied to discrete scale
# 多分 cond  に数字を入れているから。
g.Combined.Region6000
dev.off()
#もう一度グラフのオブジェクトを読み込みます#dev.off() で閉じないとファイルが保存されません
```

1.2 LME

まず必要なのは舌のようなテーブル

participant	item	condition	ratio
P01	1	a	0.5
P01	2	a	0.5
P01	3	b	0.5
P01	4	b	0.5
:	:	:	:
P25	24	b	0.5

log ratio を計算します。 なぜ 「A と B を見ていた割合」(ここでは、「怒っているイノシシ」と「怒ってないイノシシ」) を比較できないのか、に関しては、 Arai(2016) の 2．眼球運動測定研究 が詳しい。

a. チーターしか怒っているイノシシを見ていなかった
b. チーターだけ怒っているイノシシを見ていなかった

刺激絵には「怒っているイノシシ」と「怒ってないイノシシ」の絵が含まれている。「チーターしか怒っている」というように依存関係の形成に破綻した時点で、被験者は「怒っている」を考慮した再分析を行うと仮定する。

その際、その再分析が語彙情報及び表示されている画像の情報を即座に反映するのであれば, (a) の条件では依存関係が破綻し再分析されるため (破綻条件) (a) の条件は (b) の条件 (非破綻条件) より、「怒っているイノシシ」に注視が集まるはずだと期待できる。

しかしこの場合、興味対象となる対象物が「怒っているイノシシ」と「怒ってないイノシシ」というように複数あるので、単純に特定の興味対象への注視時間を従属変数とすることができない。そのため分析では「特定の対象物を他の対象物に比べてどれくらい見ていたか」という割合を計算する。

そしてここでの興味の対象は, 試行の開始から終わりまで全ての眼球運動ではなく、認識された言語情報の処理に対する反応として起こる注視である。したがって予測的眼球運動のキューとなる言語情報のオンセット時間をマークし、各音声刺激と眼球運動データをリンクさせる必要がある.

この実験ではそのキューになるのは関係節内の極性の後であるので、関係切ないの極性 (怒って “いる/ない”) の時間をチェックして（実際には眼球運動が言語情報を反映するまでにかかる時間 (およそ 180 ~ 200 ms) を加える (Matin et al., 1993))、その時間から,「イノシシ」のオンセットまでの時間の幅を分析対象とする「時間枠」として設定する。この「時間枠」の中で「怒っているイノシシ」に向けられた注視と「怒ってないイノシシ」への注視の比率を計算し, 分析を行う。

本来ならば修飾句を含めることで、分析対象の時間枠が短くなりすぎ有効な注視が観測されにくくなるのを防ぐことも有用である。なお後述するが、時間枠の幅が狭ければ狭いほどデータは二項変数分布に近似する。

元々眼球運動測定器によって記録された眼球運動データは書くサンプリングデータが記録された「時間情報」と注視のあった画面上の位置を示す「座標軸情報」で構成されている。これを絵刺激上の対象物ごとに区切ったテンプレートと照合することで、「いつ、どの対象物を見ていたか」を示すカテゴリーデータが得られる。これにより、特定の対象物への注視は、その対象物に注視があったか否か、という二項変数として扱うことができる。この二項変数データに基く割合（どの程度ある対象物を見たか）や、各至高で得られた複数のサンプリングデータ内でどれくらいターゲットを見ていたか、という割合を求めることで 0 から 1 の間の数値に変換し、その値を連続変数として分散分析などの検定テストに適用している例が多く見られた。しかし、この方法には明らかな問題が有る。

Jeager(2008) が指摘するように、連続変数を分散分析などのパラメトリック手法（前提を設ける統計の手法）で分析するには、変数の母集団が正規分布に従ったり、上限や下限を持たないことが仮定される場合が多いが、割合を計算した場合は 0 から 1 の間の数値しか得られない。また、正規分布を前提とした線形回帰モデルなどの分析を適用した場合、残差は平均値としては独立してランダムに起きるはずであり、エラーの分布は正規分布に従わなければならないが、割合を計算した場合にはそのエラーの分散は平均値に依存し、割合は 0.5 をピークとして分散が最も高くなる。こうした「前提」と「データ」の不一致が第一種の過誤（擬陽性：本来は真である null 仮説を誤って棄却すること）を起こしやすくする。

さて、どうしたものでしょうか。

割合をそのまま従属変数として扱うことができない問題に対して広く知られている解決方法は、「オッズ」を計算し、対数変換することである。オッズとは、ある出来事が起きた回数（例えば、上の VWP において、一定時間内で「怒っているイノシシ」への注視が記録された回数）のそのイベントの起こらなかった回数 (「怒っているイノシシ」への注視が記録されなかった回数) に対する比率である。また、オッズは説明変数の乗法的な効果を説明するのに適している。これによって VWP では 一定時間内で別の場所を見ていた注視に比べてどのくらいターゲット対象物を見ていたかを表すことができる。

そして, そのオッズを対数変換し、ロジット (logit) とよばれる値に変換する。このロジットを, 正と負の境界を持たない, つまり回帰分析上都合のよい、加法的な効果で説明される従属変数として線形混合モデルに加える. このロジット値を従属変数として線形混合モデルを適用することは, 混合ロジスティック回帰（二項変数に対してロジットリンク関数を用いて行う一般化線形混合モデル）を用いるのと概念上は同義になるなお、割合からロジットへは以下のように変換できる。

$$frac12$$

‖ からを写経。

a 条件は依存関係に破綻するので、b 条件

```
# あと、新井先生の論文も呼んだほうが良い。このスクリプトの著者の話だし。
# ratio を求めます。

getwd()
setwd("/home/kishiyama/home/thesis/npi-thesis/result/main")

library(lme4)
library(reshape)

data <- read.csv("./csv/output.csv", header =T)
head(data)
#バランスを確認
table(data$ParticipantName, data$Condition)
table(data$Condition)

#実際の AOI の名前をマッピングします
data$target <- ifelse(data$AOI == 1, as.character(data$AOI1), "BackGround")
data$target <- ifelse(data$AOI == 2, as.character(data$AOI2), data$target)
data$target <- ifelse(data$AOI == 3, as.character(data$AOI3), data$target)
data$target <- ifelse(data$AOI == 4, as.character(data$AOI4), data$target)

data <-data[order(data$ParticipantName, data$ItemNo, data$GazeStart),]

#onset 時間点：どの onset から切り揃えるでしょうか
onset_var <- 4900
offset_var <- 5300
span <- offset_var - onset_var

data$slapse <- data$GazeStart - onset_var
data$elapse <- data$GazeEnd - onset_var

# 例： onset1 からでしたら 0,
# data$slapse <- data$GazeStart - 0
# data$elapse <- data$GazeEnd - 0

# onset2 からでしたら data$onset2
# data$slapse <- data$GazeStart - data$onset2
# data$elapse <- data$GazeEnd - data$onset2

# そうしたら、onset 以前の部分は負数になって、
# 後で時間区間の長さを調整する部分と合わせて 0 以下は 0 と入れ替えられて処理しないことになる

  #  slapse elapse
  #1   -824   -670
  #2   -603   -397
```

```
#3       3     173
#4     177     680
#5     926    1523
#6    1220    1603
```

```
# onset 内の時間区間の長さを調整
# 仮に onset の時間から引いたら、0 スタートで n エンドになる。
# n はオンセットからの時間区間となる。
data$slapse <- ifelse(data$slapse < 0, 0, data$slapse)
data$elapse <- ifelse(data$elapse >= span, span, data$elapse)
# 上の場合、開始点は 0（当たり前だのクラッカー）, 終了点は 800 となる。

# 例：開始点は 0, 終了点は 1000
# -> 0 以下は 0 で入れ替え、1000 以上は 1000 で入れ替えます
# 相殺してしまうと、概念上 slapse と elapse 以外の点が外されてしまします。（すごい技！）
# data$slapse <- ifelse(data$slapse < 0, 0, data$slapse)
# data$elapse <- ifelse(data$elapse >= 1000, 1000, data$elapse)
```

```
#  slapse elapse
#1      0    -199
#2      0     -10
#3      3       3
#4    177     177
#5    926    1000
#6   1220    1000
```

```
# 時間内の比率を見るから、dur だけを見ていい（開始点と終了点はもう無視していい）
# どれくらいの時間見ていたか、という点に興味が有る。 l
# わかりづらいけど、途中まで進まえると分かりやすいかもしれない。
data$dur <- data$elapse - data$slapse
# 万が一 0 以下（オーバーした区間）を除外
data$dur <- ifelse(data$dur < 0, 0, data$dur)

#必要なコラムだけを残ります
data <- data[,c("ParticipantName", "ItemNo", "target", "Condition", "slapse","elapse","dur")]
data <- data[order(data$ParticipantName,data$ItemNo,data$slapse),]

#CALUCULATING SUM (aggregation for each trial)
#同じ区間の中もし別に他の注視時間があれば全部合算する。
data <-aggregate(
    data$dur,
    by=list(data$ParticipantName, data$ItemNo, data$target, data$Condition),
    FUN=sum,
    na.rm=TRUE)
colnames(data) = c("subj","item","AOI","cond","sum")
```

```r
#sort
data <- data[order(data$subj, data$item),]
#"variable","value"に書き換えないと、cast()が実行できません
colnames(data) = c("subj","item","variable","cond","value")

# cast creates separate columns for each object fixated
# 各絵に分けたら、それぞれの絵にどれくらい見ているのか、
# 後で分けて計算しやすい。（いちいち取り出すではなく、コラム単位で計算できる）

# cast 関数が無い。
data2 <- cast(data)

# ここをいじるのかな？

# replace NULL
# ここはどのような列を作っているかにもよります。
data2$BackGround <- ifelse(is.na(data2$BackGround), 0, data2$BackGround)
data2$A<- ifelse(is.na(data2$A), 0, data2$A)
data2$B<- ifelse(is.na(data2$B), 0, data2$B)
data2$C<- ifelse(is.na(data2$C), 0, data2$C)
data2$D<- ifelse(is.na(data2$D), 0, data2$D)

# 先ほどの分析において、「チーターしか　怒っているイノシシ...」の "イ" の時点での有意差がみられました。
# この有意差は、正解ラベルへの fixation が多い、という方向のものでした。
# つまり、a条件 (+npi -neg) において、関係節（おこっている vs おこってない）の極性が明らかになった直後、
# （そして否定極性要素と否定の依存関係が破綻したあと)、
# 正解ラベルへの fixation が増したことになります。
# これは最初の研究課題、つまり
# "依存関係の形成に失敗した条件で、解析器は再分析を即座に開始するのか"
# を支持しうるものになります。
#
# ただし、この差には疑問が残ります。つまり、果たして構造の差を反映しているのでしょうか、
# というのが次の疑問です。
# a条件で fixation が上がったのは事実ですが、
# それは再分析した結果上がったのか、それとも a条件 で単に「イノシシ」と聴こえたから
# イノシシを見たのかは分かりません。
# （上がったことは事実なので、構造を再解釈せずとも別の可能性を模索したとか、
# とにかく何かを積極的に言うことができるので、世界の終わりというわけではありません。）

# 仮に a条件 (+npi -neg) において構造を踏まえて予測しておらず、
# 単に聞こえてきたものを見ているだけだというのなら、
# 「怒っていないイノシシ」と「怒っているイノシシ」への fixation に差はないはずです。
# 一方、「怒っていないイノシシ」と「怒っているイノシシ」への fixation において、
# 怒っているイノシシがより多く見られているはずです。
# ただ、これは単純に、「怒っていない」オブジェクトへの視線の動きが少ないだけかもしれません。
# npi と neg の要因を操作した結果だというためには、
# -npi -neg 条件では上の差は見られないハズです。
```

```
# <!-- -npi +neg とかを見ないのはなぜ？
# それだと、ターゲットが-->
#
# まずはあたりをつけてみませんか？
# グラフを作る手順はどのみち必要だったから。

# たぶん、data2$A とかの列にはその文節で見られている時間が記録されている。
# だから、data2$Target みたいなのを作って上げればいいはず。

# N1 V1 AFF N2 -> N2 V1 が正解
# TODO ここおかしい。たぶん、

# https://mathtrain.jp/explnlg
# ln for logarithmus naturalis

data4a <- data2[data2$cond == "a",]
data4b <- data2[data2$cond == "b",]
# 比べたい２つの条件を一つに。
data4ab <- rbind(data4a,data4b)
summary(data4ab$cond)
data4ab <- droplevels(data4ab)
# data4ab$C が a と b にとっての正解。
# ここ、dat$wts<- 1/(dat$AOI1+0.5)+1/(dat$sum - dat$AOI1+0.5) みたいにしたほうが良い。
data4ab$logit <- log(((data4ab$C) + 0.5)/((data4ab$C)+(data4ab$D)-(data4ab$C)+0.5))

# 下位条件を付けます
# wo-ni の奴やで。
data2 <- data4ab
data2$npi <- ifelse(data2$cond == "a", 1, 0)

#中心化
# data2$npi <- scale(data2$npi, scale=F)
# data2$aff <- scale(data2$aff, scale=F)
data2$npi <- scale(data2$npi, scale=T)

#転換しやすいため
#logit1:Competitor_TargetCompound
#logit2:TargetCompound
#logit3:CompetitorCompound
#logit4:TargetSimplex

# data2$logit<-data2$logit_c
tapply(data2$logit, list(data2$npi), mean)

#
# # #汎用性 LME モデル
# m10 <- lmer(logit ~ npi * correct + (1 + npi*correct |subj) + (1 + npi*correct |item), data = data
```

```
# m09 <- lmer(logit ~ npi * correct + (1 + correct + npi:correct |subj) + (1 + npi*correct |item), d
# m08 <- lmer(logit ~ npi * correct + (1 + correct + npi:correct |subj) + (1 + correct + npi:correct
# m07 <- lmer(logit ~ npi * correct + (1 + npi:correct |subj) + (1 + correct + npi:correct |item), d
# m06 <- lmer(logit ~ npi * correct + (1 + npi:correct |subj) + (1 + npi:correct |item), data = data

#
#m09 <- lmer(logit ~ npi * correct + (1 + correct + npi:correct |subj) + (1 + npi*correct |item), da
#m08 <- lmer(logit ~ npi * correct + (1 + correct + npi:correct |subj) + (1 + npi+ npi:correct |item
#
# m08 <- lmer(logit ~ correct * npi:correct + (1 + npi+correct |subj) + (1 + npi+correct |item), dat
#
# # m09 <- lmer(logit ~ npi * correct + (1 + npi + npi:correct |subj) + (1 + npi*correct |item), dat
#
# m08 <- lmer(logit ~ npi * correct + (1 + npi*correct |subj) + (1 + npi*correct |item), data = data
# m07 <- lmer(logit ~ npi * correct + (1 + npi*correct |subj) + (1 + npi*correct |item), data = data
# m06 <- lmer(logit ~ npi * correct + (1 + npi*correct |subj) + (1 + npi*correct |item), data = data
# m05 <- lmer(logit ~ npi * correct + (1 + npi*correct |subj) + (1 + npi*correct |item), data = data

m00 <- lmer(logit ~ npi + (1  |subj) + (1 |item), data = data2)
m10 <- lmer(logit ~ npi + (1 +npi |subj) + (1 + npi |item), data = data2)
m09 <- lmer(logit ~ npi + (1 |subj) + (1 + npi |item), data = data2)

# TODO: ここまでやる。

anova(m09,m00)

# anova(m09,m00)
# anova(m08,m00)

# m08wi  <-  lmer(logit ~ npi * correct + (1 + correct + npi:correct |subj) + (1 + npi+ npi:correct
# m08woi  <-  lmer(logit ~ npi + correct + (1 + correct + npi:correct |subj) + (1 + npi+ npi:correct
# m08won  <-  lmer(logit ~ correct + npi:correct + (1 + correct + npi:correct |subj) + (1 + npi+ npi
# m08woa  <-  lmer(logit ~ npi + npi:correct + (1 + correct + npi:correct |subj) + (1 + npi+ npi:cor

# # npi
# anova(m08wi, m08won)
#
# # correct
# anova(m08wi, m08woa)
#
# # npi:correct(interaction)
# anova(m08wi, m08woi)
#
# 有意差が消えた


m00  <-  lmer(logit ~ npi + (1|subj) + (1|item), data = data2)
```

```r
m00wom  <-  lmer(logit ~ 1 + (1|subj) + (1|item), data = data2)

# npi:correct(interaction)
anova(m00, m00wom)
```