

## GLMM Workshop

JACET 英語語彙・英語辞書・リーディング研究会合同研究会

@早稲田大学

3/9/2019

### Today's Menu

- はじめに
    - 経緯
    - 注意事項
  - WS の準備
    - RStudio Server の準備
    - R および RStudio の準備
  - 一般（化）線形（混合）モデルの概略
  - データ分析 + 一般化線形モデル + 一般化線形混合モデル
  - おわりに
- 

### はじめに

- 経緯

今回の WS の分析の部分は下記のテクニカルレポートの内容を元にお話させていただきます（論文，データ，R コードすべてオンライン公開済）。

田村祐（2016）「外国語教育研究における二値データの分析ーロジスティック回帰を例にー」『外国語教育メディア学会中部支部外国語教育基礎研究部会 2015 年度報告論集』29-82. [リンク](#)

### 注意事項②

- 本 WS は 2 時間の設定となっていますが，実際の GLMM の習得には，その倍以上の復習が必要となります

- 今回使用するコードはすべて web 上で公開していますので、復習の際にご利用ください
- GitHub: [https://github.com/tam07pb915/JACET-SIG\\_GLMM-Workshop](https://github.com/tam07pb915/JACET-SIG_GLMM-Workshop)
- 個別の分析に関するご相談にはお応えできない可能性が高いですので、参考文献をご参照ください

## 簡単に自己紹介

- 2014 年(D1 年時)に俗に言う「緑本」(久保, 2012)を読んでから統計モデリングを勉強するように
- Nagoya.R #12 という勉強会で、「一般化線形混合モデル入門の入門」という R を使った線形混合効果モデルのやり方について発表
- その後から自分の研究でも LME や GLMM を使うように
- Tamura, Y. (2015). Reinvestigating consciousness-raising grammar task and noticing. *JABAET Journal*, 19, 19–47. -> mixed-effect logistic regression
- Tamura, Y., Fukuta, J., Nishimura, Y., Harada, Y., Hara, K., & Kato, D. (2019). Japanese EFL learners' sentence processing of conceptual plurality: An analysis focusing on reciprocal verbs. *Applied Psycholinguistics*, 41, 59–91. -> Gamma GLMM

## 一般（化）線形（混合）モデルの概略

- 一般線形モデル
- いわゆる普通の回帰分析（単回帰，重回帰）
- 正規分布が仮定される
- t 検定や分散分析も線形モデルの一種
- 一般化線形モデル
- 正規分布以外の確率分布族を用いるもの
  - 回数データ（単位時間あたりの生起回数）->ポアソン分布，負の二項分布
  - 二値データ（正答・誤答）->二項分布
  - 非負の連続量（反応時間等）->ガンマ分布，逆正規分布

## 確率分布を考える

---

## 混合効果（mixed effect）とはなにか

- 固定効果＋変量効果のこと

- 分散分析でいう F1（被験者分析）と F2（項目分析）を同時にやってしまうようなもの
- 固定効果->説明変数（独立変数）のこと。いわゆる研究者が見たいもの
- 変量効果->応答変数（従属変数）に影響を与えてしまうばらつき
- 追試を行うときに変わるモノ->変量効果
  - 例: 単語の頻度効果を調べる追試
    - 固定効果->単語の頻度（高・低またはあるコーパスでの頻度）
    - 変量効果->実験参加者、単語（もとの研究と同じ刺激語であっても）

## 応用言語学の分野でも新しくない

- R と、混合効果モデルが簡単に実装可能な *lme4* パッケージの爆発的普及で一般化
- 混合効果モデルのイントロ論文として私がいつもおすすめるもの
- Cunnings, I. (2012). An overview of mixed-effects statistical models for second language researchers. *Second Language Research*, 28, 369–382.  
[doi:10.1177/0267658312443651](https://doi.org/10.1177/0267658312443651)
- Linck, J. A. & Cunnings, I. (2015), The utility and application of mixed-effects models in second language research. *Language Learning*, 65, 185-207.  
[doi:10.1111/lang.12117](https://doi.org/10.1111/lang.12117)

## 今回の WS で扱う分析

### ロジスティック回帰分析

- 0/1（成功・失敗）のデータを扱うときに使う
- 二項分布を用いる
- $n$ 回（ただし  $n > 0$ ）の試行のうちの成功回数  $k$  が発生する確率  $q$  を求める
- 各試行は独立な試行（ベルヌーイ試行）

$$p(k|N, q) = {}_N C_k * q^k (1 - q)^{N-k}$$

- コインを 10 回投げて 4 回表が出る確率は？（ただし  $q=0.5$  とする）  
 $p(4|10, 0.5) = {}_{10} C_4 0.5^4 (1 - 0.5)^{10-4} = 0.205$

### ロジスティック関数からロジット関数へ

- 先ほどのロジスティック関数を変形すると次のようになる

$$\log\left(\frac{q_i}{1-q_i}\right) = z_i$$

- この式の左辺はロジット関数と呼ばれる
- ロジスティック関数とロジット関数は逆関数の関係にある
- ロジスティクス関数と線形予測子の関係をつなぐリンク関数がロジットリンク関数

## ロジスティック回帰のまとめ

- 下記のような方程式を解くこと

$$\log\left(\frac{q_i}{1-q_i}\right) = \beta_0 + \beta_1 X_1 + \dots$$

- 個人ごとの成功確率である  $q_i$  は得られたデータからわかることであり、 $X_1$  は独立変数の値である
- $\beta_0$  と  $\beta_1 \dots$  を最尤推定(maximum Likelihood estimation)で求める
- オッズ ( $\frac{q_i}{1-q_i}$ ) の対数( $\log$ )が線形予測子と等しいので次のようにも表現できる

$$\begin{aligned} \frac{q_i}{1-q_i} &= \exp(\beta_0 + \beta_1 X_1 + \dots) \\ &= \exp(\beta_0) * \exp(\beta_1 X_1) \dots \end{aligned}$$

## データ分析の準備

- RStudio Server をお使いの方は、今回の WS に必要なパッケージはすでにインストール済みです (尾関先生ありがとうございます)
- ご自身の R または RStudio をご利用の方は、下記のコード (以降はコメントアウトですので打たなくて OK です) ]を実行してパッケージのインストールをお願いします

```
install.packages("lme4") #分析
install.packages("sjPlot") #可視化
install.packages("broom")
install.packages("emmeans") #下位検定と可視化
install.packages("dplyr") #ハンドリング
install.packages("tidyr") #ハンドリング
install.packages("psych") #記述統計
install.packages("MASS") #分析
install.packages("car") #分析
```

## 背景 (GLM)

- とある文法項目に対するフィードバックの効果検証 (※仮想データです)

- 明示的 (explicit) フィードバック群
- 暗示的 (implicit) フィードバック群
- 事前と事後の伸びを比較
- 学習者ごとにその項目を使った回数が異なる->割合にしてしまいがちだがそうしない
- 参加者の TOEIC スコアが異なる->low, mid, high みたいにしてしまいがちだがそうしない
- だいたいの場合分け方が恣意的
- 群分けすることでせっかくのデータが失われる
- 非線形の関係があるなら群分けしても...と思うがそれなら GAM (General Additive Model) という手も (cf. [Murakami, 2016](#))

---

## 一般化線形モデル(glm 関数)

- 仮想データの読み込み

```
sample1 <- read.csv("http://bit.ly/kisoken2015Tamura_Sample1")
head(sample1, 5) #データの確認
```

```
##   ID   group TOEIC timing try success
## 1   1 implicit  579  test1   5        2
## 2   2 implicit  735  test1  13        4
## 3   3 implicit  477  test1  12        2
## 4   4 implicit  653  test1   8        4
## 5   5 implicit  416  test1   8        1
```

- 変数の確認
- ID: 参加者 ID
- group: implicit / explicit
- TOEIC: 参加者の TOEIC スコア
- timing: test1(事前)/test2(事後)
- try: 全使用回数
- success: 正用回数

## 成功確率の記述統計を出してみる②

```
sample1%>% #%>%はパイプ演算子で、左側でやったことを右側に渡すという意味
  dplyr::group_by(group)%>% #グループごとに分ける
  dplyr::group_by(timing, add=T)%>% #事前と事後で分ける
```

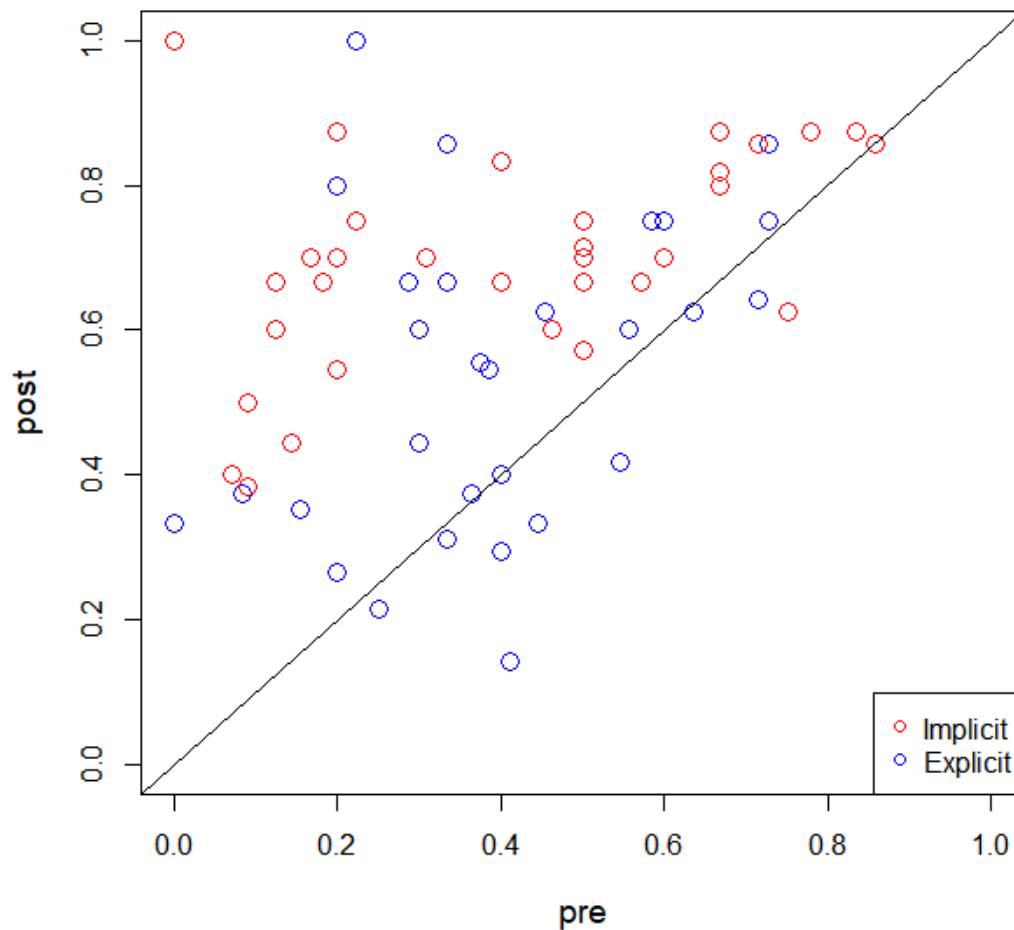
```
dplyr::summarize_at(.,vars(acc),funcs(mean,sd,min,max)) #acc の行について
平均, 標準偏差, 最大・最長地を計算

## # A tibble: 4 x 6
## # Groups:   group [?]
##   group    timing mean    sd   min   max
##   <fct>    <fct> <dbl> <dbl> <dbl> <dbl>
## 1 explicit test1  0.388 0.197 0     0.727
## 2 explicit test2  0.542 0.217 0.143 1
## 3 implicit test1  0.406 0.256 0     0.857
## 4 implicit test2  0.700 0.148 0.385 1
```

## 成功確率の散布図を描いてみる②

コードが長いのでスライド上では省略しています。お手元の資料または GitHub 上の R コードを御覧ください。

```
plot(as.matrix(subset(acc_exp,timing=="test1")[,2]),
     as.matrix(subset(acc_exp,timing=="test2")[,2]),
     xlab="",ylab="",main="",
     ylim=c(0,1),xlim=c(0,1),cex.lab=1.2,
     col="blue",axes=F,cex=1.5)
par(new=T) #図を重ねるために
plot(as.matrix(subset(acc_imp,timing=="test1")[,2]),
     as.matrix(subset(acc_imp,timing=="test2")[,2]),
     xlab="pre",ylab="post",
     main="",ylim=c(0,1),xlim=c(0,1),cex.lab=1.2,
     col="red",cex=1.5)
legend("bottomright",legend=c("Implicit","Explicit"),
      col=c("red","blue"),pch=c(1,1)) #凡例を右下に
abline(0,1) #真ん中に斜線を引っ張る
```



---

## データ分析前の準備

### 1.コーディングの変更

R のデフォルトはダミーコーディング (0, 1)ですが, これを変更します

参考: <http://talklab.psy.gla.ac.uk/tvw/catpred/>

```
c<-contr.treatment(2)
my.coding <-matrix(rep(1/2,2,ncol=1))
my.simple <-c-my.coding
print(my.simple)#my.simple が-0.5, 0.5 の行列になっているのを確認
```

```
##      2
## 1 -0.5
## 2  0.5

contrasts(sample1$group)<-my.simple #group 変数のコーディングを変更
contrasts(sample1$timing)<-my.simple #timing 変数のコーディングを変更
str(sample1) #データセットの構造の確認

## 'data.frame':   128 obs. of  7 variables:
## $ ID      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ group   : Factor w/ 2 levels "explicit","implicit": 2 2 2 2 2 2 2 2 2 2 ...
## ..- attr(*, "contrasts")= num [1:2, 1] -0.5 0.5
## .. ..- attr(*, "dimnames")=List of 2
## .. .. $ : chr  "explicit" "implicit"
## .. .. $ : chr  "2"
## $ TOEIC   : int  579 735 477 653 416 317 603 892 696 559 ...
## $ timing  : Factor w/ 2 levels "test1","test2": 1 1 1 1 1 1 1 1 1 1 ...
## ..- attr(*, "contrasts")= num [1:2, 1] -0.5 0.5
## .. ..- attr(*, "dimnames")=List of 2
## .. .. $ : chr  "test1" "test2"
## .. .. $ : chr  "2"
## $ try     : int  5 13 12 8 8 10 8 12 3 5 ...
## $ success: int  2 4 2 4 1 2 4 8 2 2 ...
## $ acc     : num  0.4 0.308 0.167 0.5 0.125 ...
```

## 本分析

- ロジスティック回帰は `cbind(成功回数, 試行回数-成功回数)` が応答変数
- 人の生死など, 1 人につき 0 か 1 かのみが与えられる場合は応答変数列は 0 または 1 が含まれるベクトルでよい
- モデル式
- 基本の形は 応答変数 ~ 説明変数 + ...
- $X+Y+X:Y$  は  $X$  の主効果,  $Y$  の主効果,  $X:Y$  の交互作用の意味
- $X+Y+X:Y$  は  $X*Y$  と同義である
- 様々なモデルを比較するため, `model` というリスト形式の変数を作ってそこに結果を格納する

```
model<-list()
model[[1]]<-glm(cbind(success,try-success)~timing+group+timing:group,
               family=binomial(link="logit"),data=sample1)
```

## 分析結果の確認

```
summary(model[[1]])
```



```
##
## Call:
## glm(formula = cbind(success, try - success) ~ timing + group +
##      timing:group, family = binomial(link = "logit"), data = sample1)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4334  -0.9310   0.1469   1.0627   2.6398
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.07161    0.05963  -1.201  0.229776
## timing2       0.82290    0.11926   6.900  5.2e-12 ***
## group2        0.37897    0.11926   3.178  0.001484 **
## timing2:group2 0.86899    0.23852   3.643  0.000269 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 294.91  on 127  degrees of freedom
## Residual deviance: 229.43  on 124  degrees of freedom
## AIC: 544.74
##
## Number of Fisher Scoring iterations: 4
```

- Residual deviance (残差逸脱度)が自由度より大きければ要注意

```
summary(model[[1]])

##
## Call:
## glm(formula = cbind(success, try - success) ~ timing + group +
##      timing:group, family = binomial(link = "logit"), data = sample1)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4334  -0.9310   0.1469   1.0627   2.6398
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.07161    0.05963  -1.201  0.229776
## timing2       0.82290    0.11926   6.900  5.2e-12 ***
## group2        0.37897    0.11926   3.178  0.001484 **
## timing2:group2 0.86899    0.23852   3.643  0.000269 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 294.91 on 127 degrees of freedom
## Residual deviance: 229.43 on 124 degrees of freedom
## AIC: 544.74
##
## Number of Fisher Scoring iterations: 4

model[[1]]$df.residual

## [1] 124
```

### 交互作用があるかどうかをどう判定するか

- 交互作用項が含まれていないモデルをつくり、尤度比検定や AIC(Akaike Information Criterion)の比較を行う

```
#モデルの更新にはupdate()関数が便利。変える部分だけ+や-で調整する
model[[2]]<-update(model[[1]],.~-timing:group) #交互作用を抜く
anova(model[[1]],model[[2]],test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: cbind(success, try - success) ~ timing + group + timing:group
## Model 2: cbind(success, try - success) ~ timing + group
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1 124 229.43
## 2 125 242.85 -1 -13.413 0.0002499 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

AIC(model[[1]])-AIC(model[[2]]) #小さい方がよいので負の値なら[[1]]がベター

## [1] -11.41298
```

- 交互作用項が含まれる方が良いモデル

### 熟達度の影響を考慮する

- 交互作用があるのはわかったが、熟達度 (TOEIC スコア) の影響があるかもしれない

```
model[[3]]<-update(model[[1]],.~.+c.TOEIC)
#summary(model[[3]])
```

- model[[2]]と model[[3]]を比べると model[[3]]のほうが残差逸脱度も低くて良さそう

```
anova(model[[2]],model[[3]],test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(success, try - success) ~ timing + group
## Model 2: cbind(success, try - success) ~ timing + group + c.TOEIC + ti
ming:group
##   Resid. Df Resid. Dev Df Deviance   Pr(>Chi)
## 1      125      242.85
## 2      123      120.79  2    122.05 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 交互作用項のさらなる追加

*#binomial* のデフォルトリンク関数は *logit* なので省略可能

```
model[[4]]<-glm(cbind(success,try-success)~
                timing*group*c.TOEIC,
                data=sample1,family = binomial)
```

## モデル選択という考え方

- 回帰分析の変数投入方法にはいろいろな方法があるが、MASS パッケージの *step* 関数を使うと、AIC に基づいてモデル選択の結果を返してくれる
- 一番複雑な *model[[4]]* を *step* 関数に渡すと、AIC が最小となるモデルが得られる

```
step(model[[4]])

## Start:  AIC=431.89
## cbind(success, try - success) ~ timing * group * c.TOEIC
##
##               Df Deviance    AIC
## - timing:group:c.TOEIC  1   108.66 429.97
## <none>                  108.58 431.89
##
## Step:  AIC=429.97
## cbind(success, try - success) ~ timing + group + c.TOEIC + timing:grou
p +
##   timing:c.TOEIC + group:c.TOEIC
##
##               Df Deviance    AIC
## - group:c.TOEIC  1   109.71 429.02
## <none>          108.66 429.97
## - timing:group  1   118.45 437.76
## - timing:c.TOEIC 1   119.78 439.09
##
## Step:  AIC=429.02
## cbind(success, try - success) ~ timing + group + c.TOEIC + timing:grou
p +
```

```

##      timing:c.TOEIC
##
##              Df Deviance    AIC
## <none>              109.71 429.02
## - timing:group      1   120.33 437.64
## - timing:c.TOEIC    1   120.79 438.10
##
## Call:  glm(formula = cbind(success, try - success) ~ timing + group +
##      c.TOEIC + timing:group + timing:c.TOEIC, family = binomial,
##      data = sample1)
##
## Coefficients:
##      (Intercept)          timing2          group2          c.TOEIC
##      -0.061733         0.887305         0.536503         0.004732
## timing2:group2 timing2:c.TOEIC
##      0.826030         -0.003197
##
## Degrees of Freedom: 127 Total (i.e. Null);  122 Residual
## Null Deviance:      294.9
## Residual Deviance: 109.7      AIC: 429

step(model[[4]])->model[[5]] #選ばれたモデルをmodel[[5]]に

## Start:  AIC=431.89
## cbind(success, try - success) ~ timing * group * c.TOEIC
##
##              Df Deviance    AIC
## - timing:group:c.TOEIC  1   108.66 429.97
## <none>              108.58 431.89
##
## Step:  AIC=429.97
## cbind(success, try - success) ~ timing + group + c.TOEIC + timing:group +
##      timing:c.TOEIC + group:c.TOEIC
##
##              Df Deviance    AIC
## - group:c.TOEIC      1   109.71 429.02
## <none>              108.66 429.97
## - timing:group      1   118.45 437.76
## - timing:c.TOEIC    1   119.78 439.09
##
## Step:  AIC=429.02
## cbind(success, try - success) ~ timing + group + c.TOEIC + timing:group +
##      timing:c.TOEIC
##

```

```
##              Df Deviance    AIC
## <none>              109.71 429.02
## - timing:group      1   120.33 437.64
## - timing:c.TOEIC    1   120.79 438.10
```

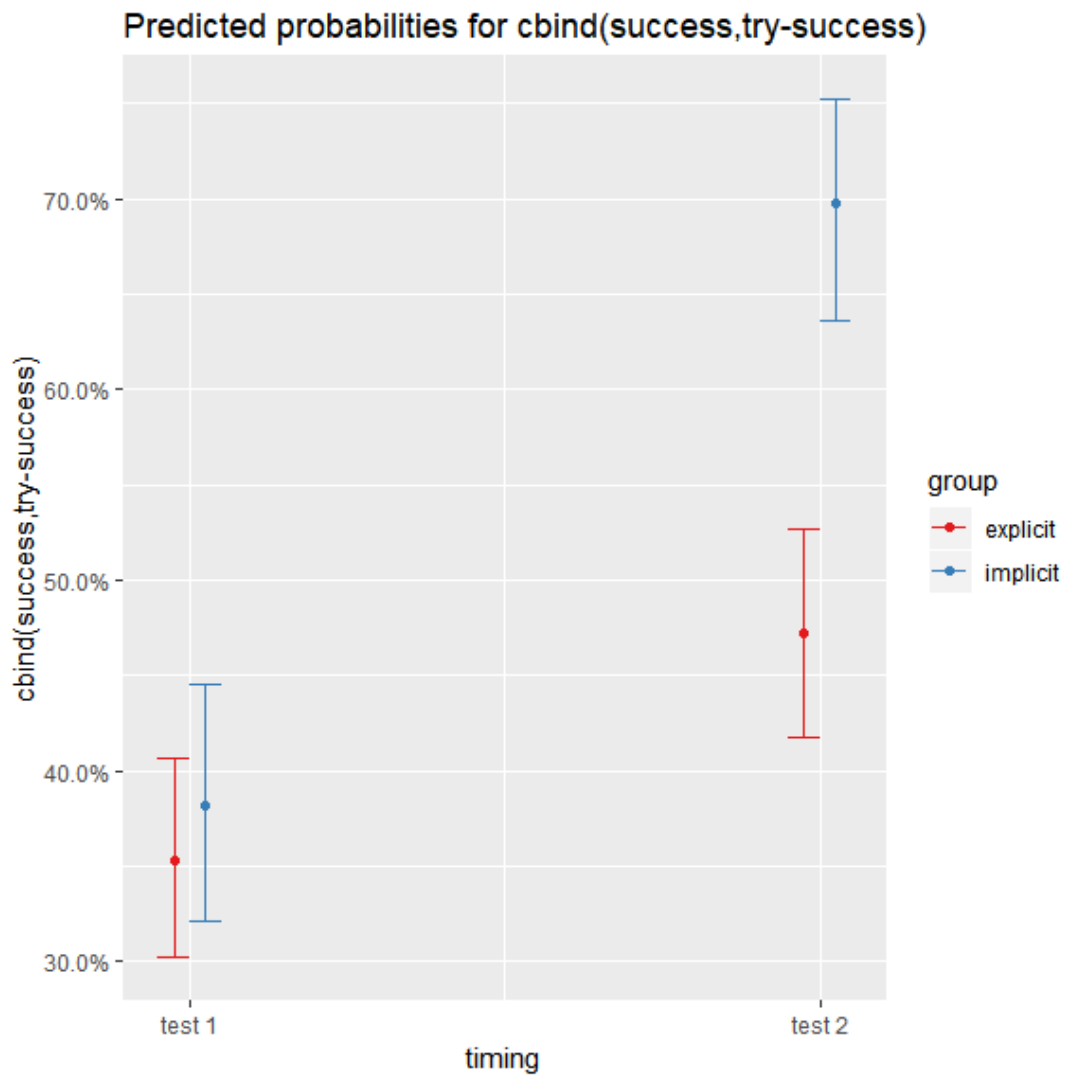
---

```
summary(model[[5]])
```

```
##
## Call:
## glm(formula = cbind(success, try - success) ~ timing + group +
##      c.TOEIC + timing:group + timing:c.TOEIC, family = binomial,
##      data = sample1)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4168  -0.4807   0.1656   0.6841   2.6773
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.0617332  0.0633830  -0.974  0.330071
## timing2       0.8873053  0.1267660   7.000 2.57e-12 ***
## group2        0.5365029  0.1273182   4.214 2.51e-05 ***
## c.TOEIC        0.0047316  0.0004831   9.793 < 2e-16 ***
## timing2:group2  0.8260304  0.2546364   3.244 0.001179 **
## timing2:c.TOEIC -0.0031966  0.0009663  -3.308 0.000939 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 294.91  on 127  degrees of freedom
## Residual deviance: 109.71  on 122  degrees of freedom
## AIC: 429.02
##
## Number of Fisher Scoring iterations: 4
```

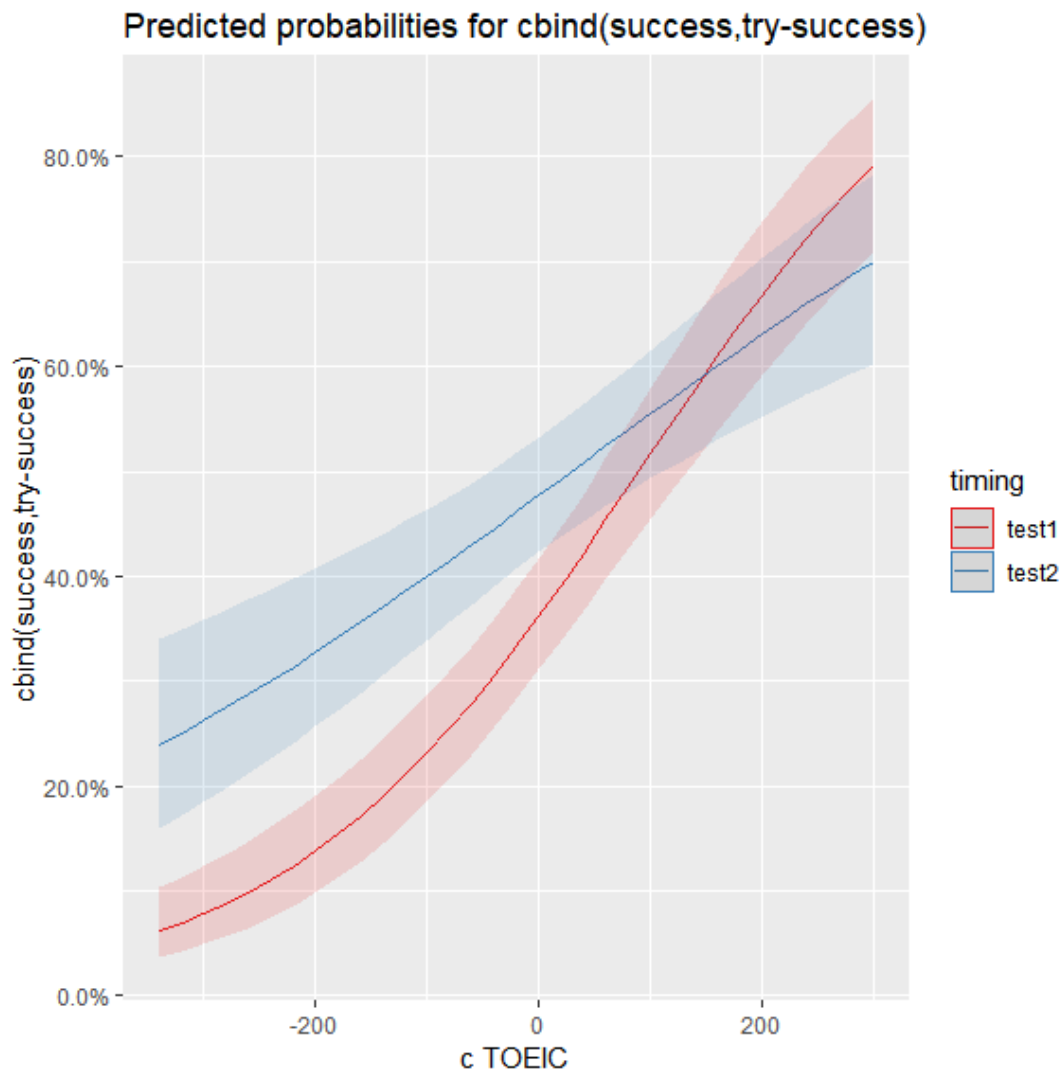
### 図示（フィードバックのタイプの影響）

```
plot_model(model[[5]], type="pred", terms = c("timing", "group"))
```



図示（フィードバックと熟達度の関係）

```
plot_model(model[[5]],type="pred",terms = c("c.TOEIC","timing"))
```



```
#plot_model(model[[4]],type="pred",terms = c("c.TOEIC","timing","group"))
```

### 効果量（オッズ比）を求める①

- ある事象の起こる確率が  $p$  で表されるとき、オッズは次のようになる

$$Odds = \frac{p}{1-p}$$

\* オッズ比はオッズの比で表されるので、確率  $p$  と確率  $q$  のオッズ比は次のようになる

$$\frac{\frac{p}{1-p}}{\frac{q}{1-q}}$$

### 効果量（オッズ比）を求める②

- ロジット関数を思い出すと，これはオッズの対数であり，商の対数は対数の差と等しいので次のように表せる

$$\log\left(\frac{q_i}{1-q_i}\right) = \log(q_i) - \log(1-q_i)$$

\*つまり，2つの確率のロジットの差がオッズ比の対数である

$$\log\left(\frac{\frac{p}{1-p}}{\frac{q}{1-q}}\right) = \log\left(\frac{p}{1-p}\right) - \log\left(\frac{q}{1-q}\right)$$

### 効果量（オッズ比）を求める③

- このオッズ比の仕組みが今理解できなくても大丈夫です
- 大事なことは次のこと

### 偏回帰係数を指数変換するとオッズ比

- Rではexp()という関数があるので，オッズ比は次のように求められる

*##\$coefficients は結果の中の推定値に直接アクセスするという意味*  
`model[[5]]$coefficients %>% exp`

```
##      (Intercept)      timing2      group2      c.TOEIC
##      0.9401337      2.4285765      1.7100163      1.0047428
## timing2:group2 timing2:c.TOEIC
##      2.2842333      0.9968085
```

### グループごとのフィードバックの効果のオッズ比

- 明示的フィードバックグループ

```
glm(cbind(success, try-success) ~ timing*group.e + c.TOEIC,
    data = sample1, family = binomial) %>%
  tidy(., conf.int=T, exponentiate=T)
```

*#tidy 関数でオッズ比と信頼区間をわかりやすく表示*

```
## # A tibble: 5 x 7
##   term                estimate std.error statistic p.value conf.low conf.
high
##   <chr>                <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
<dbl>
```



```
## 1 (Intercept)      0.728  0.0805      -3.95 7.84e- 5    0.621
0.852
## 2 timing2          1.52   0.160      2.62 8.75e- 3    1.11
2.09
## 3 group.e2         1.77   0.127      4.47 7.93e- 6    1.38
2.27
## 4 c.TOEIC          1.00   0.000478    9.84 7.69e-23    1.00
1.01
## 5 timing2:group.e2 2.62   0.252      3.84 1.26e- 4    1.61
4.31
```

## グループごとのフィードバックの効果のオッズ比

- 暗示的フィードバックグループ

```
glm(cbind(success, try-success)~ timing*group.i+c.TOEIC
, data = sample1, family = binomial)%>%
  tidy(., conf.int=T, exponentiate=T)
#tidy 関数でオッズ比と信頼区間をわかりやすく表示

## # A tibble: 5 x 7
##   term                estimate std.error statistic  p.value conf.low conf.
high
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
<dbl>
## 1 (Intercept)      1.29  0.0975      2.57 1.01e- 2    1.06
1.56
## 2 timing2          4.00  0.195      7.12 1.09e-12    2.74
5.88
## 3 group.i2         1.77  0.127      4.47 7.93e- 6    1.38
2.27
## 4 c.TOEIC          1.00  0.000478    9.84 7.69e-23    1.00
1.01
## 5 timing2:group.i2 2.62  0.252      3.84 1.26e- 4    1.61
4.31
```

## 他にも便利な関数の紹介①

- GLM などのモデルを HTML 形式のきれいなテーブルにしてくれる sjPlot パッケージの tab\_model 関数

```
tab_model(model[[5]]) #引数でいろいろ調整できます
```

<i>Predictors</i>	<b>cbind(success,try-success)</b>		
	<i>Odds Ratios</i>	<i>CI</i>	<i>p</i>
(Intercept)	0.94	0.83 – 1.06	0.330
test 2	2.43	1.90 – 3.12	<b>&lt;0.001</b>
implicit	1.71	1.33 – 2.20	<b>&lt;0.001</b>
c TOEIC	1.00	1.00 – 1.01	<b>&lt;0.001</b>
timing2:group2	2.28	1.39 – 3.77	<b>0.001</b>
timing2:c.TOEIC	1.00	0.99 – 1.00	<b>0.001</b>
Observations	128		
Cox & Snell's $R^2$ / Nagelkerke's $R^2$	0.765 / 0.850		

## 他にも便利な関数の紹介②

- epiDisplay というパッケージの `logistic.display` 関数でもテーブル形式でオッズ比がみれます

```
install.packages("epiDisplay")
```

```
library(epiDisplay)
```

```
## Warning: package 'epiDisplay' was built under R version 3.5.2
```

```
## Loading required package: foreign
```

```
## Loading required package: survival
```

```
## Loading required package: nnet
```

```
##
```

```
## Attaching package: 'epiDisplay'
```

```
## The following objects are masked from 'package:psych':
```

```
##
```

```
##      alpha, cs, lookup
```

```
logistic.display(model[[5]])
```

```
##
```

```
##              OR lower95ci upper95ci      Pr(>|Z|)
```

```
## timing2      2.4285765 1.8942982 3.1135455 2.567835e-12
```

```
## group2       1.7100163 1.3323759 2.1946926 2.510277e-05
```

```
## c.TOEIC      1.0047428 1.0037918 1.0056947 1.203979e-22
```

```
## timing2:group2 2.2842333 1.3867352 3.7625942 1.178803e-03
```

```
## timing2:c.TOEIC 0.9968085 0.9949225 0.9986982 9.393128e-04
```

## 注意点

- 連続変数を入れるときは、従属変数との線形的な関係があることが前提
- 例えば、低熟達度では高く、中熟達度で下がり、高熟達度でまた高くなるような非線形の関係がある場合はうまくモデリングできない
- 非線形の関係があるかどうかは図示または一般化加法モデル(**General Additive Model**)によって確認が必要
- **GAM** をやっている論文はあまり見かけないが、Akira Murakami さんの下記の論文は例がわかりやすい上に R のコードも付録としてついているのでオススメ

Murakami, A. (2016), Modeling systematicity and individuality in nonlinear second language development: The case of English grammatical morphemes. *Language Learning*, 66, 834-871. doi:10.1111/lang.12166

## GLMM の分析に使うデータの概要

- 下記の発表で用いられたデータ

田村祐 (2016) 「文法性判断課題における反応時間と主観的測度は正答率を予測するか-文法項目の違いに焦点をあてて-」 外国語教育メディア学会中部支部外国語教育基礎研究部会第三回年次例会. 名古屋学院大学. [\[投影資料\]](#)

- 応答変数 (**response variable**): 文法性判断課題の正答・誤答データ
- 説明変数
- 反応時間
- 主観的測度 (規則が説明できる or 直感である)
- 名詞の種類 (普通名詞 12 項目・物質名詞 12 項目)
  - 普通名詞の例: apple, dog, pen, knife, child
  - 物質名詞の例: gold, wine, toast, rice

---

## 概要

- 実験
- それぞれの項目ごとに、文法文と非文法文があり、2 つのリストに参加者をランダムに振り分けてカウンターバランス
- 刺激文 24 にフィラー文 40 文を加えた合計 64 文の文法性判断課題
- PC 上で行われ、注視点(1000ms)->ブランク(500ms)->刺激文の提示
- キー押下によって文法性判断

- 判断後に直前の回答の判断のリソースを「自分の知っている規則である」「直感である」の2つから選択
- 参加者
- 日本語を第一言語とする大学生・大学院生 24 名
- TOEIC の平均スコアは 704 ( $SD = 95.39, n = 22$ )

## 一般化線形混合モデル(glmer 関数)

- データの読み込み

```
sample2 <- read.csv("http://bit.ly/kisoken2015Tamura_Sample2", header=T)
head(sample2)
```

```
##   ID itemID judgment    rt sub.measure form Type
## 1  1     41         1 13420          0    A  CNP
## 2  1     43         1  5952          1    A  CNP
## 3  1     45         1  8717          1    A  CNP
## 4  1     47         1 10398          1    A  CNP
## 5  1     49         1  9511          1    A  CNP
## 6  1     51         0 25420          1    A  CNP
```

- 変数の確認
- ID: 参加者 ID
- itemID: 項目の ID
- judgment: 文法性判断の正答・誤答 (0/1)
- rt: 反応時間
- sub.measure: 主観的測度 (0=直感, 1=規則 0)
- form: 参加者のフォーム
- Type: 名詞の種類

## 成功確率の記述統計

- 先ほどのデータと違って 1 人のデータは 1 行に 0/1 が複数行となっているのでちょっと工夫が必要
- 普通名詞のほうが正答率が高そう
- 直感よりも規則のほうが高そう

# 名詞ごとに、規則か直感かで正答率の差をみる

```
sample2 %>%
  dplyr::group_by(ID) %>%
  dplyr::group_by(Type, add=T) %>%
```

```
dplyr::group_by(sub.measure, add=T)%>%
dplyr::summarise(mean=mean(judgment, na.rm=T)) -> stat_acc

stat_acc<-as.data.frame(stat_acc)
stat_acc$sub.measure<-as.factor(stat_acc$sub.measure)
describeBy(stat_acc$mean, group = list(stat_acc$sub.measure, stat_acc$Type))

##
## Descriptive statistics by group
## : 0
## : CNP
## vars n mean sd median trimmed mad min max range skew kurtosis
## se
## X1 1 22 0.67 0.31 0.67 0.71 0.4 0 1 1 -0.58 -0.53 0.07
## -----
## : 1
## : CNP
## vars n mean sd median trimmed mad min max range skew kurtosis
## X1 1 24 0.82 0.21 0.89 0.85 0.16 0.25 1 0.75 -1.23 0.42
## se
## X1 0.04
## -----
## : 0
## : MNP
## vars n mean sd median trimmed mad min max range skew kurtosis
## se
## X1 1 24 0.45 0.25 0.46 0.46 0.3 0 0.83 0.83 -0.35 -1.02
## 0.05
## -----
## : 1
## : MNP
## vars n mean sd median trimmed mad min max range skew kurtosis
## se
## X1 1 24 0.57 0.24 0.6 0.58 0.16 0 1 1 -0.34 -0.08
## 0.05
```

---

## 主観的測度の割合の記述統計

```
sample2%>%
dplyr::group_by(ID)%>%
dplyr::group_by(Type, add = T)%>%
dplyr::summarise(mean=mean(sub.measure, na.rm=T)) -> stat_sub
```

```
stat_sub<-as.data.frame(stat_sub)
describeBy(stat_sub$mean,group=stat_sub$Type)

##
## Descriptive statistics by group
## group: CNP
##   vars  n mean   sd median trimmed  mad  min max range  skew kurtosis
## X1    1 24 0.76 0.19  0.79    0.78 0.19 0.08  1  0.92 -1.75    4.23
##      se
## X1 0.04
## -----
## group: MNP
##   vars  n mean   sd median trimmed  mad  min  max range  skew kurtosi
## X1    1 24 0.58 0.22  0.62    0.6 0.19 0.08 0.83  0.75 -0.79   -0.5
##      se
## X1 0.04
```

## • 普通名詞のほうが主観と答える割合が高そう

### 反応時間の記述統計

```
#For RT
sample2%>%
  dplyr::group_by(ID)%>%
  dplyr::group_by(sub.measure,add=T)%>%
  dplyr::group_by(Type,add=T)%>%
  dplyr::summarise(mean=mean(rt,na.rm=T)) ->stat_rt

stat_rt$sub.measure<-as.factor(stat_rt$sub.measure)
describeBy(stat_rt$mean,group = list(stat_rt$sub.measure,stat_rt$Type))

##
## Descriptive statistics by group
## : 0
## : CNP
##   vars  n    mean      sd median trimmed   mad   min    max
## X1    1 22 13804.19 6030.2  13464 13588.97 7059.04 2776.33 27309.4
##      range skew kurtosis    se
## X1 24533.07 0.29   -0.65 1285.64
## -----
## : 1
## : CNP
##   vars  n    mean      sd median trimmed   mad   min    max
## X1    1 24 9733.1 3585.92 9339.23 9689.01 4055.99 3959.78 15915.89
##      range skew kurtosis    se
```

```
## X1 11956.11 0.15      -1.27 731.97
## -----
## : 0
## : MNP
##   vars  n      mean      sd  median trimmed   mad   min    max
## X1    1 24 15443.42 6485.25 14680.83 15230.07 5446.95 5029.33 27184.5
##       range skew kurtosis    se
## X1 22155.17 0.4      -0.92 1323.8
## -----
## : 1
## : MNP
##   vars  n      mean      sd  median trimmed   mad  min    max  ra
nge
## X1    1 24 11157.53 5454.14 9218.39 10429.59 4800.74 4326 24987.7 2066
1.7
##   skew kurtosis    se
## X1  1.1      0.49 1113.32
```

- 直感って言うときのほうが RT が長そう

## 箱ひげ図を描く

- 青い点は平均値でバーは 96%信頼区間

*#beeswarm がなければインストールが必要*  
`install.packages("beeswarm")`

## 図を描くためにデータフレームを作る

```
sample2%>%
  dplyr::filter(Type=="CNP")%>%
  dplyr::group_by(ID)%>%
  dplyr::summarise(mean=mean(judgment,na.rm=T))->stat_cnp #ランダムに欠損
がある場合, na.rm=T を指定しないと欠損のある学習者のデータが落ちてしまう
```

```
stat_cnp<-as.data.frame(stat_cnp)
```

*#For common nouns*

```
sample2%>%
  dplyr::filter(Type=="MNP")%>%
  dplyr::group_by(ID)%>%
  dplyr::summarise(mean=mean(judgment,na.rm=T))->stat_mnp #ランダムに欠損
がある場合, na.rm=T を指定しないと欠損のある学習者のデータが落ちてしまう
```

```
stat_mnp<-as.data.frame(stat_mnp)
```

```
library(beeswarm)

## Warning: package 'beeswarm' was built under R version 3.5.2

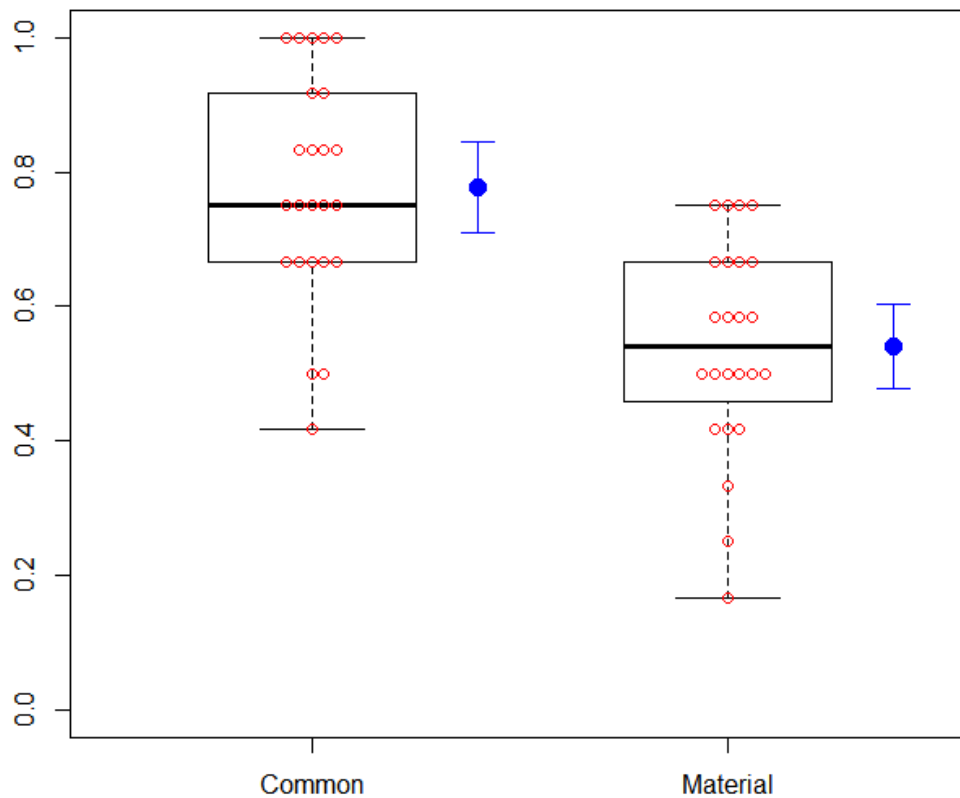
acc.all<-cbind(stat_cnp$mean,stat_mnp$mean) #二つのデータフレームをくっつける
acc.all<-as.data.frame(acc.all)
names(acc.all)<-c("Common","Material") #列名をつける
head(acc.all) #確認

##      Common  Material
## 1 0.9166667 0.6666667
## 2 0.6666667 0.5000000
## 3 0.8333333 0.7500000
## 4 1.0000000 0.6666667
## 5 0.7500000 0.5000000
## 6 0.5000000 0.6666667

#平均値とその95%信頼区間を描くための準備
desc.cnp<-describe(stat_cnp$mean)
desc.mnp<-describe(stat_mnp$mean)
upper.cnp<-desc.cnp$mean+(1.96*desc.cnp$se) #信頼区間の上限
lower.cnp<-desc.cnp$mean-(1.96*desc.cnp$se) #信頼区間の下限
upper.mnp<-desc.mnp$mean+(1.96*desc.mnp$se)
lower.mnp<-desc.mnp$mean-(1.96*desc.mnp$se)

boxplot(acc.all,boxwex=0.5,ylim=c(0,1))#まずは箱ひげ図
beeswarm(acc.all,add=T,col="red") #蜂群図を重ねる
points(1.4,desc.cnp$mean,col="blue",pch=19,cex=1.5) #平均値の点をうつ
points(2.4,desc.mnp$mean,col="blue",pch=19,cex=1.5)
arrows(1.4,desc.cnp$mean,1.4,upper.cnp,angle = 90,length = 0.1,col="blue")
#平均値から上下に線を引く
arrows(1.4,desc.cnp$mean,1.4,lower.cnp,angle = 90,length = 0.1,col="blue")
arrows(2.4,desc.mnp$mean,2.4,upper.mnp,angle = 90,length = 0.1,col="blue")
arrows(2.4,desc.mnp$mean,2.4,lower.mnp,angle = 90,length = 0.1,col="blue")
```





## コーディングの変更と連続変数の中心化

# 因子型に変更しないとコーディングを変更できない

```
sample2$sub.measure<-as.factor(sample2$sub.measure)
```

```
sample2$Type<-as.factor(sample2$Type)
```

```
contrasts(sample2$sub.measure)<-my.simple
```

```
contrasts(sample2$Type)<-my.simple
```

# RT のデータは値が非常に大きく中心化だけではモデルが収縮しづらいので標準化する

```
sample2$rt%>%
```

```
  scale%>%
```

```
  as.numeric->sample2$zrt
```

## 結果の確認

```
summary(fit[[1]])
```

```

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: judgment ~ Type * sub.measure * zrt + (1 + Type + sub.measure
+
## zrt | ID) + (1 + Type + sub.measure + zrt | itemID)
## Data: sample2
## Control:
## glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##          AIC          BIC    logLik deviance df.resid
##          702          824     -323     646      548
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.5992 -0.6844  0.3395  0.5925  1.6014
##
## Random effects:
## Groups Name             Variance Std.Dev. Corr
## itemID (Intercept)    0.32887  0.5735
##              Type2      0.79867  0.8937    0.75
##              sub.measure2 0.70060  0.8370    0.65  0.99
##              zrt        0.04734  0.2176   -0.61  0.07  0.21
## ID      (Intercept)    0.26973  0.5194
##              Type2      0.46535  0.6822   -0.82
##              sub.measure2 0.43806  0.6619   -0.47 -0.12
##              zrt        0.30472  0.5520   -0.81  0.32  0.90
## Number of obs: 576, groups: itemID, 48; ID, 24
##
## Fixed effects:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.7076    0.2071   3.416 0.000634 ***
## Type2            -1.3154    0.3764  -3.495 0.000475 ***
## sub.measure2      0.6973    0.3247   2.147 0.031765 *
## zrt              -0.1034    0.1820  -0.568 0.569974
## Type2:sub.measure2 -0.1679    0.5827  -0.288 0.773207
## Type2:zrt         0.1582    0.2769   0.571 0.567777
## sub.measure2:zrt  -0.1115    0.2676  -0.417 0.676826
## Type2:sub.measure2:zrt -0.6521    0.5413  -1.205 0.228304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) Type2  sb.ms2 zrt      Ty2:.2 Typ2:z sb.m2:
## Type2        -0.125
## sub.measur2  -0.180  0.234
## zrt          -0.349  0.126  0.412

```

```
## Typ2:sb.ms2  0.253 -0.104 -0.234 -0.061
## Type2:zrt    0.085 -0.175 -0.114 -0.166  0.296
## sb.ms2:zrt   0.227 -0.079 -0.143  0.106  0.054  0.097
## Typ2:sb.m2: -0.064  0.232  0.081  0.066 -0.139  0.093 -0.189
```

## 多重共線性の確認

- Jaeger が作成した `glmer` の出力から VIF が計算可能なスクリプトを使用する
- 2.5 以上のものがあれば要チェック

```
source("https://raw.githubusercontent.com/aufrank/R-hacks/master/mer-util
s.R")
vif.mer(fit[[1]])
```

```
##              Type2              sub.measure2              zrt
##              1.150589              1.373793              1.298081
##      Type2:sub.measure2      Type2:zrt      sub.measure2:zrt
##              1.190906              1.212838              1.117969
## Type2:sub.measure2:zrt
##              1.161966
```

- 今回は問題なさそう

```
tab_model(fit[[1]], show.stat = T, transform = NULL)
```

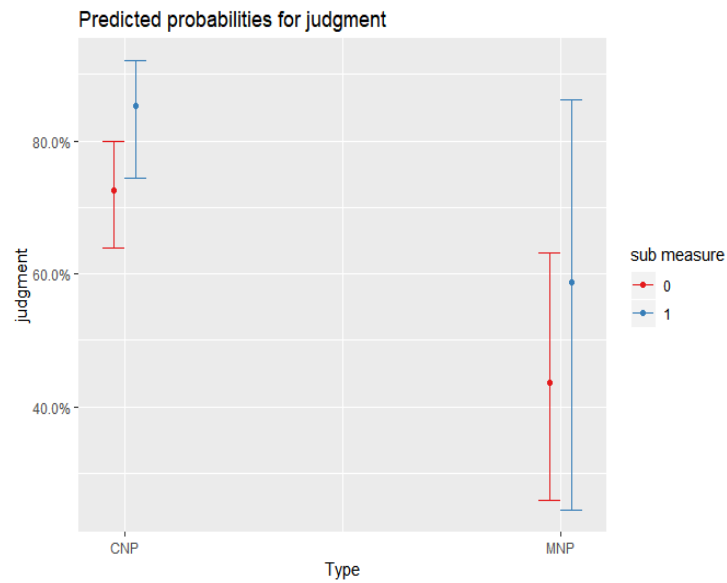
*#transform=NULL を消すと、一番左が Log Odds ではなく Odds Ratio になる*

Predictors	judgment			
	Log-Odds	CI	Statistic	p
(Intercept)	0.71	0.30 – 1.11	3.42	<b>0.001</b>
MNP	-1.32	-2.05 – -0.58	-3.49	<b>&lt;0.001</b>
sub measure 1	0.70	0.06 – 1.33	2.15	<b>0.032</b>
zrt	-0.10	-0.46 – 0.25	-0.57	0.570
Type2:sub.measure2	-0.17	-1.31 – 0.97	-0.29	0.773
Type2:zrt	0.16	-0.38 – 0.70	0.57	0.568
sub.measure2:zrt	-0.11	-0.64 – 0.41	-0.42	0.677
Type2:sub.measure2:zrt	-0.65	-1.71 – 0.41	-1.20	0.228

## 図示

- 交互作用はどれも有意ではなかったが、一応図示してみる
- 名詞の種類と主観的測度

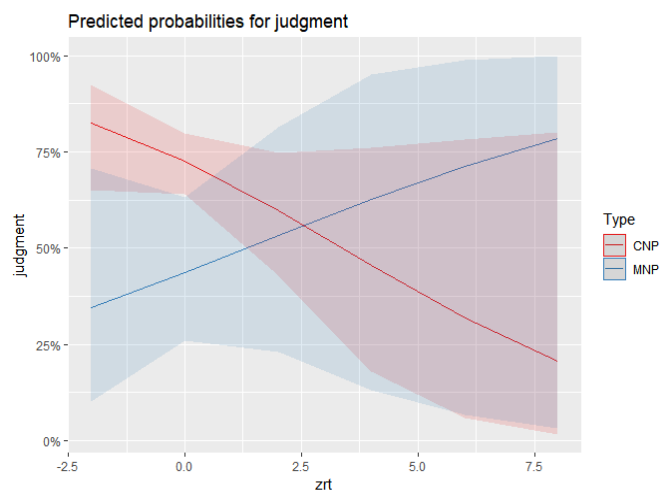
```
plot_model(fit[[1]],type="pred",terms=c("Type","sub.measure"))
```



## 図示②

- 名詞の種類と反応時間

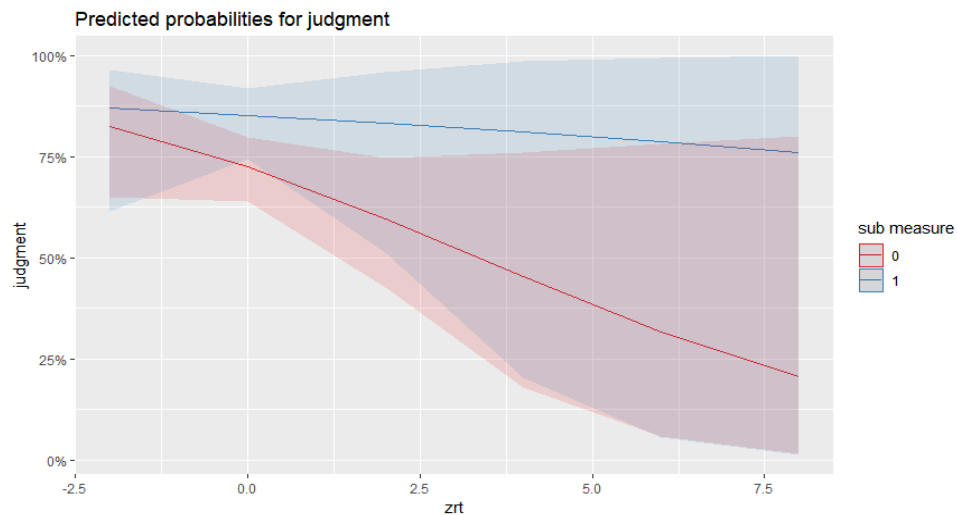
```
plot_model(fit[[1]],type="pred",terms=c("zrt","Type"))
```



## 図示③

- 主観的測度と反応時間

```
plot_model(fit[[1]],type="pred",terms=c("zrt","sub.measure"))
```



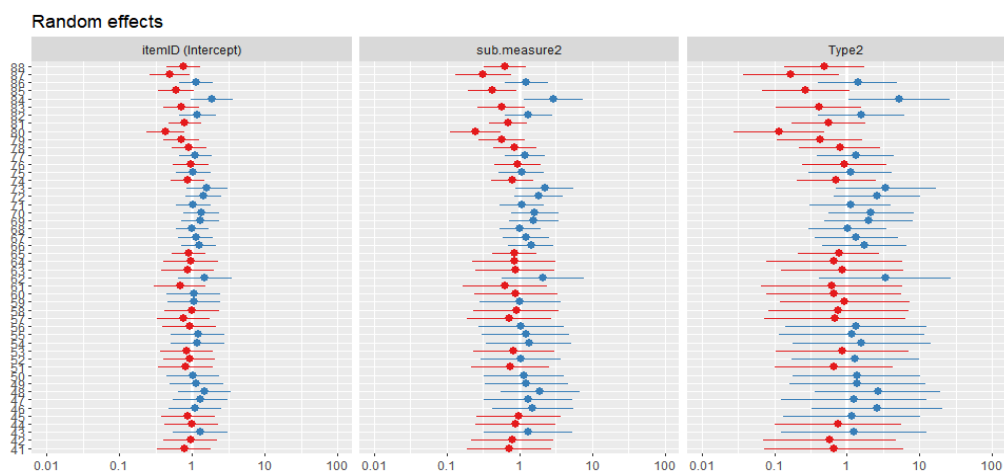
## 変量効果を図示する①

- 項目ごとの変量効果

#スライドにおさめるために1 つずつにしていますが, 普段なら[[1]]は不要

```
plot_model(fit[[2]],type="re")[[1]]
```

#plot\_model(fit[[1]],type="re",sort.est = T,grid = F) とすると並び替えができる



## 最終的な結果

```
summary(fit[[2]])
```

```
tab_model(fit[[2]])
```

<i>Predictors</i>	<b>judgment</b>		
	<i>Odds Ratios</i>	<i>CI</i>	<i>p</i>
(Intercept)	2.15	1.50 – 3.10	<b>&lt;0.001</b>
MNP	0.33	0.16 – 0.66	<b>0.002</b>
sub measure 1	1.90	1.11 – 3.26	<b>0.019</b>
<b>Random Effects</b>			
$\sigma^2$	3.29		
T00 itemID	0.23		
T00 ID	0.25		
T11 itemID.Type2	1.41		
T11 itemID.sub.measure2	0.54		
T11 ID.Type2	0.40		
T11 ID.sub.measure2	0.17		
P01 itemID.Type2	0.76		
P01 itemID.sub.measure2	0.92		
P01 ID.Type2	-0.50		
P01 ID.sub.measure2	-0.14		
ICC itemID	0.06		
ICC ID	0.07		
Observations	576		
Marginal R <sup>2</sup> / Conditional R <sup>2</sup>	0.095 / 0.330		

---

## 結果の解釈

- 主観的測度のオッズ比は 1.9 なので、規則を説明できるという場合のほうが、直感であるという場合よりも文法性判断に正答する確率が 1.9 倍高い
  - 名詞タイプのオッズ比は 0.33 なので、物質名詞が関わる文法性判断課題の正答率は普通名詞のそれよりも 0.33 倍
  - 普通名詞を 1 としたときにその 3 割ほどしか正答確率がない
  - 反応時間が遅くても速くても正答率に影響があるという傾向が認められなかった
  - 主観的測度との交互作用なし→文法性判断が直感でも規則でも RT は正答率に影響しない
  - 名詞タイプとの交互作用なし→普通名詞のときでも物質名詞のときでも RT は正答率に影響しない
- 

## おわりに

- 一部コードが違いますが、より概念的な説明や参考文献などは、最初に紹介したテクニカルレポートに載っていますのでそちらをご参照ください
  - かなりパッケージに依存した形で処理を行っていますので、パッケージのアップデート等によって動かなくなる可能性もあります
  - R のバージョンを新しくする
  - パッケージがアップデートされたらマニュアルを読んだりネットで調べて変更点を把握する
  - ようなことも必要になることがあります
  - 論文にするときの報告の仕方等は、Linck and Cunnings (2015)に詳しいです
  - とにかく近年は統計解析技術の進歩が非常に早いです
  - これさえできればというものでもないです
  - 今回紹介した手法が 5 年後もそのまま使えるかはわかりません
- 

## 参考文献リスト

- Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68, 255–278. doi:10.1016/j.jml.2012.11.001]
- Cunnings, I. (2012). An overview of mixed-effects statistical models for second language researchers. *Second Language Research*, 28, 369–382.

[doi:10.1177/0267658312443651](https://doi.org/10.1177/0267658312443651)]

- Murakami, A. (2016), Modeling systematicity and individuality in nonlinear second language development: The case of English grammatical morphemes. *Language Learning*, 66, 834-871.  
[[doi:10.1111/lang.12166](https://doi.org/10.1111/lang.12166)] (<https://doi.org/10.1111/lang.12166>)
- Linck, J. A. & Cunnings, I. (2015), The utility and application of mixed-effects models in second language research. *Language Learning*, 65, 185-207.  
[doi:10.1111/lang.12117](https://doi.org/10.1111/lang.12117)
- 久保拓弥 (2012). 『データ解析のための統計モデリング入門』 東京: 岩波書店.
- 清水裕士 (2014). 『個人と集団のマルチレベル分析』 京都: ナカニシヤ出版

## おまけ

- 手っ取り早く交互作用のあるモデルを検証したいときは, `emmeans` 関数もあります
- 特に水準が 3 つ以上ある場合には便利

```
emmeans(model[[5]], pairwise~group|timing)$contrasts #事前と事後における群間比較
```

```
## timing = test1:
## contrast      estimate      SE  df z.ratio p.value
## explicit - implicit -0.1234877 0.1783095 Inf  -0.693  0.4886
##
## timing = test2:
## contrast      estimate      SE  df z.ratio p.value
## explicit - implicit -0.9495181 0.1817841 Inf  -5.223  <.0001
##
## Results are given on the log odds ratio (not the response) scale.
```

参考 URL: [https://cran.r-](https://cran.r-project.org/web/packages/emmeans/vignettes/interactions.html)

[project.org/web/packages/emmeans/vignettes/interactions.html](https://cran.r-project.org/web/packages/emmeans/vignettes/interactions.html)

```
emmeans(model[[5]], pairwise~timing|group)$contrasts #両群における事前と事後の比較
```

```
## group = explicit:
## contrast      estimate      SE  df z.ratio p.value
## test1 - test2 -0.4742901 0.1620026 Inf  -2.928  0.0034
##
## group = implicit:
## contrast      estimate      SE  df z.ratio p.value
## test1 - test2 -1.3003205 0.1957403 Inf  -6.643  <.0001
##
## Results are given on the log odds ratio (not the response) scale.
```



---

## おまけ 2

- 今回はもともと `long` 型で 1 行が 1 試行となっていたが、このようなデータに整形するのに困ったら、`tidyr` パッケージがオススメです
- 変数が多い研究デザイン系の縦横変換
- `tidyr` — シンプルなデータ変形ツール