Modèles à effets mixtes

(sous R)

Dominique Muller

(dominique.muller@univ-grenoble-alpes.fr)

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

Un exemple (Judd et al., 2012)

- Une chercheuse présente 15 visages de blancs et 15 visages de noirs
- Pour chaque visage, un jugement sur une échelle de -20 à +20
- Souvent, dans cette situation, les chercheurs vont :
  - moyenner les réponses sur les 15 visages blancs et les 15 visages noirs
  - comparer ces deux moyennes en intra-sujet

→ Risque important d'erreurs de Type 1 avec cette analyse

Pourquoi utiliser des modèles mixtes ?

- Pour contrôler les erreurs de Type I (et donc favoriser la réplicabilité)
- Pour permettre une grande variété d'analyse (plus de flexibilité)
- Pour garder un maximum d'informations

Quelques références

- Baayen, R. H. (2008). *Analyzing linguistic data. A Practical introduction to statistics using R.* Cambridge University Press.
- Baayen, R.H., Davidson, D.J., & Bates, D.M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language* 59, 390-412.
- Bates, D.M. (in progress, 2010). lme4: Mixed-effects modeling with R
- Judd, C. M., Westfall, J., & Kenny, D. A. (2012). Treating stimuli as a random factor in social psychology: A new and comprehensive solution to a pervasive but largely ignored problem. *Journal of Personality and Social Psychology*, *103*, 54-69.
- Judd, C. M., Westfall, J., & Kenny, D. A. (2016). Experiments with more than one random factor: Designs, analytic models, and statistical power. *Annual Review of Psychology, 68,* 601-625.
- Wonnacott, T.H. & Wonnacott, R.J. (1990). *Introductory statistics for business and economics* (4th Edition).

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Un exemple simplifié

- Une chercheuse présente 1 visage de blanc et 1 visage de noir
- Pour chaque visage, un jugement sur une échelle de -20 à +20

```
  pp  Cond     y
1  1  noir -2.30
2  1 blanc  9.27
3  2  noir  0.71
4  2 blanc -5.56
5  3  noir -3.94
6  3 blanc  0.42
...
```

2 problèmes se posent :

- Indépendance des résidus
- Problème de généralisabilité

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - ✓ Concepts de base
  - ✓ Déclaration du modèle
  - ✓ Effets fixes
  - ✓ Effets aléatoires

Solution classique au problème stat

- Pour éliminer le problème de non-indépendance des résidus, le tableau de données est "reconditionné" (pour cela utilisation de la fonction "cast" (package "reshape")

=> `DFcourt <- cast(DF,pp~Cond,value="y")`

```
  pp  Cond     y
1  1  noir -2.30
2  1 blanc  9.27
3  2  noir  0.71
4  2 blanc -5.56
5  3  noir -3.94
6  3 blanc  0.42
...
```

```
  pp blanc   noir
1  1  9.27  -2.30
2  2 -5.56   0.71
3  3  0.42  -3.94
...
```

Solution classique au problème stat

- Ensuite, nous créons un score de différence W_1

  => `DFcourt$W1 <- DFcourt$blanc - DFcourt$noir`

- Enfin, nous pouvons réaliser la régression correspondant au modèle suivant : $W_{1i} = \alpha_0 + \varepsilon_i$

```
  pp blanc   noir    W1
1  1  9.27  -2.30 11.57
2  2 -5.56   0.71 -6.27
3  3  0.42  -3.94  4.36
4  4  6.67 -10.04 16.71
5  5 -8.98  -4.36 -4.62
6  6  4.20   1.63  2.57
...
```

Cette régression (une ANOVA intra à deux modalités) ne pose plus de problème d'indépendance des résidus

Solution classique au problème stat

$$W_{1i} = \alpha_0 + \varepsilon_i$$

- Pour réaliser cette régression dans R, nous utilisons :

  `fit.lm <- lm(W1~1,DFcourt)` => 1 indique qu'il faut estimer l'intercept

  `summary(fit.lm)`

- Ce qui nous donne :

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.3467     1.1935    0.29    0.774

Residual standard error: 6.537 on 29 degrees of freedom
```

Solution classique au problème stat

- Nous avons donc réussi à régler le problème stat
- Il reste néanmoins le problème de méthode lié à la faible possibilité de généraliser
- L'étude avec 15 visages par groupe permet de traiter ce problème, mais en moyennant les 15 items, nous perdons une information critique pour cette généralisation

=> recours à une autre méthode stat pour gérer la non-indépendance et la généralisabilité

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Une autre solution au problème stat

- Au lieu de "reconditionner" le tableau de données et de travailler sur un score de différence, nous allons modéliser les sources de non-indépendance

```
  pp  Cond     y
1  1  noir -2.30
2  1 blanc  9.27
3  2  noir  0.71
4  2 blanc -5.56
5  3  noir -3.94
6  3 blanc  0.42
...
```

Ici nous allons donc modéliser (et estimer) la variabilité liée aux participants

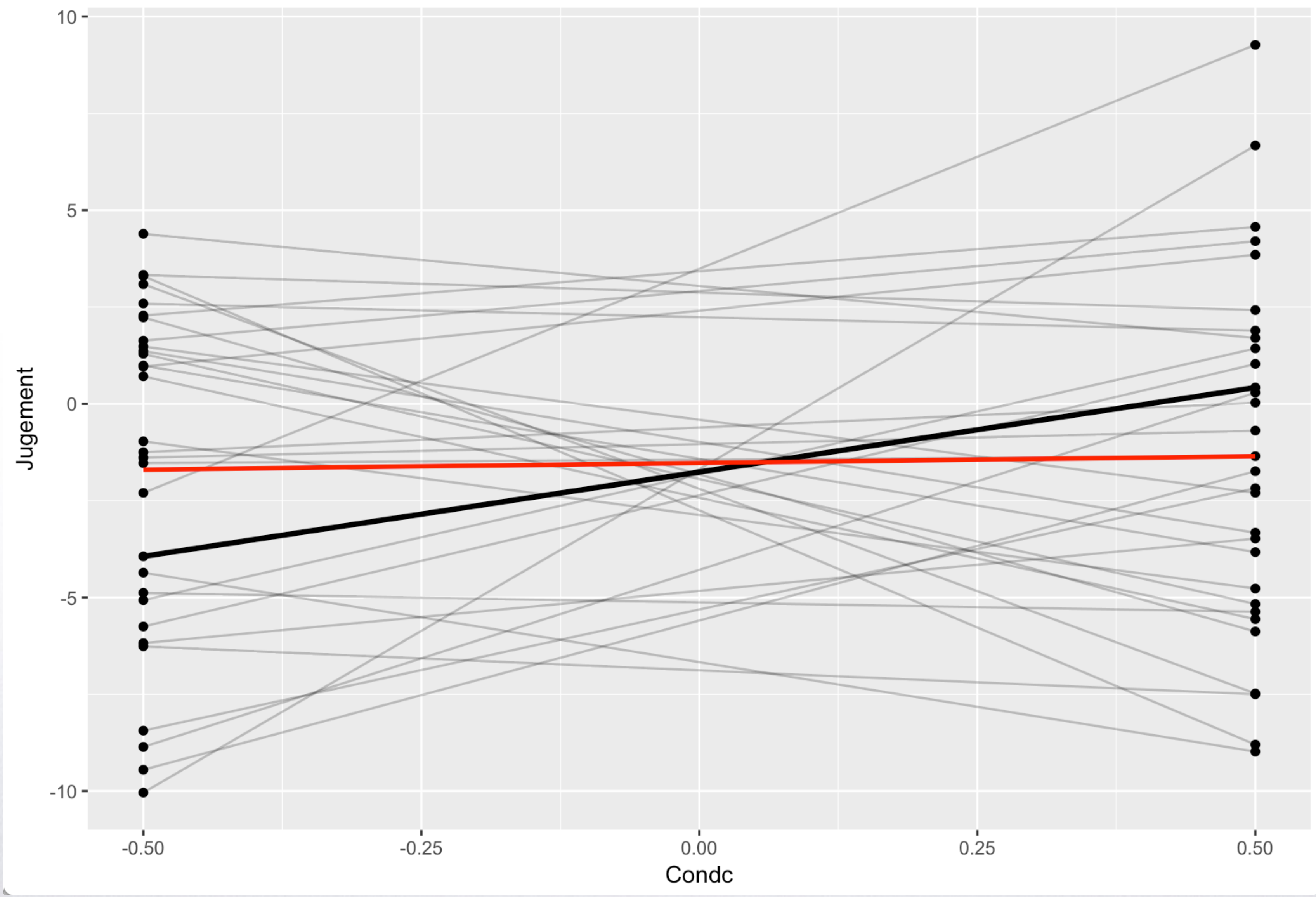
Une autre solution au problème stat

- Sans ce problème de non-indépendance, nous aurions pu utiliser le modèle : $Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \varepsilon_{ij}$

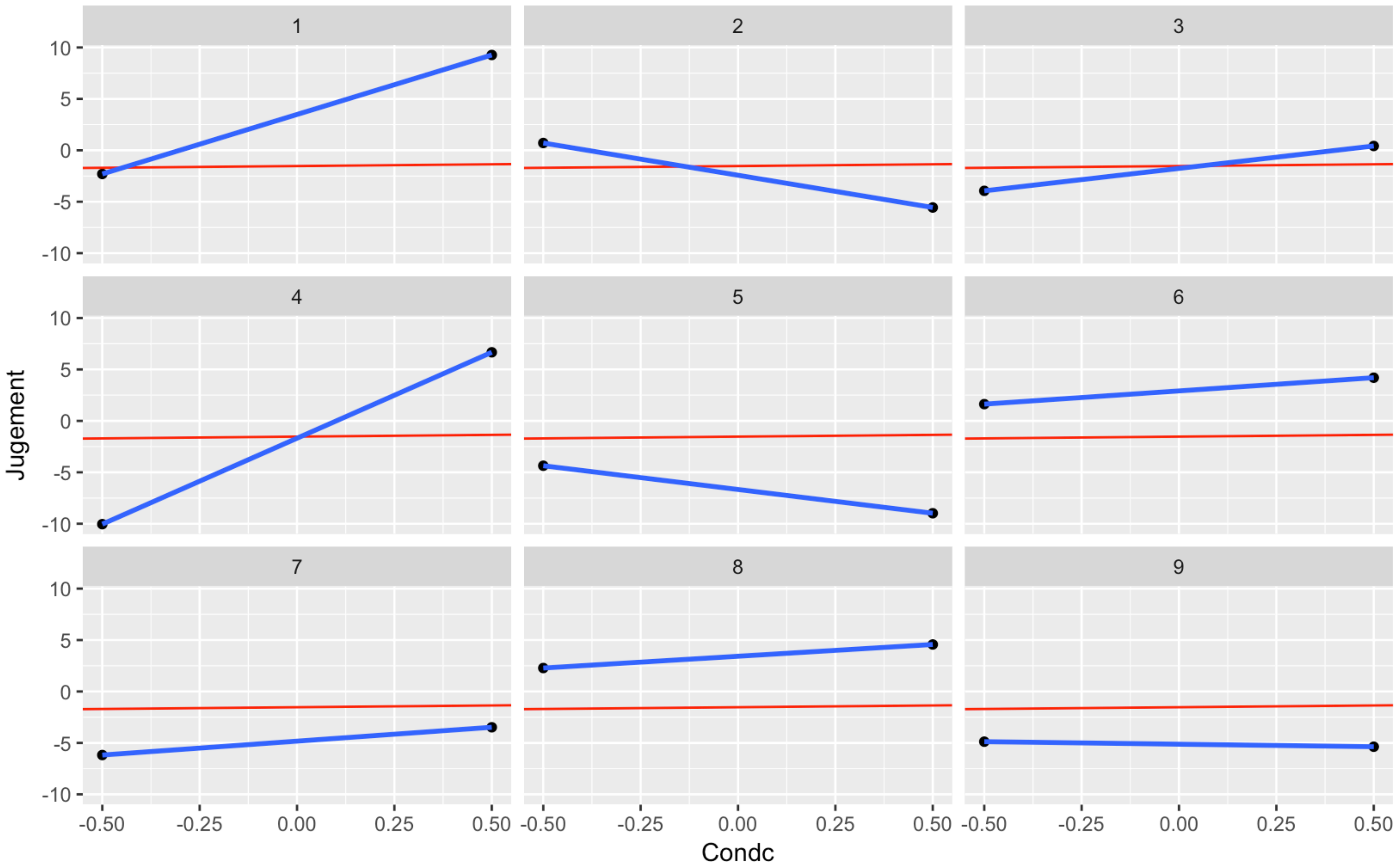
```
  pp  Cond     y Condc  stim
1  1  noir -2.30  -0.5 stim1
2  1 blanc  9.27   0.5 stim2
3  2  noir  0.71  -0.5 stim1
4  2 blanc -5.56   0.5 stim2
5  3  noir -3.94  -0.5 stim1
6  3 blanc  0.42   0.5 stim2
...
```

- *i* = Participants
- *j* = Stimuli
- *Y* : la VD
- *C* = Condc, un contraste pour la variable condition
- ε_{ij} = l'erreur résiduelle pour le participant *i* et le stim. *j*

- Modéliser les sources de non-indépendance implique de décomposer les sources de variances contenues dans ε_{ij}. Autrement dit, décomposer tout ce qui n'est pas l'intercept et la pente.

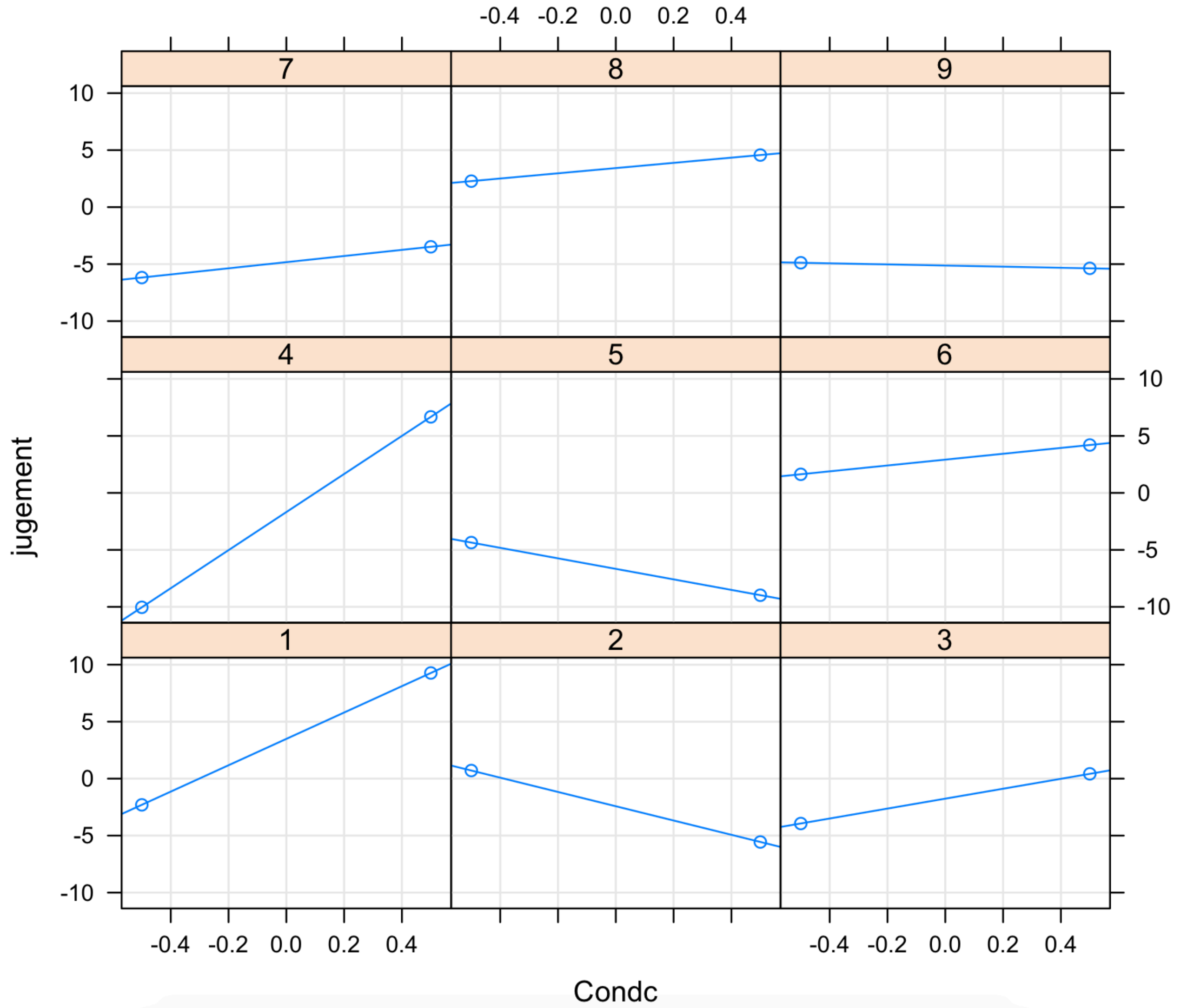


10
5
0
-5
-10
Jugement
-0.50
-0.25
0.00
0.25
0.50
Condc



1
2
3
4
5
6
7
8
9
10
5
0
-5
-10
-0.50
-0.25
0.00
0.25
0.50
Jugement
Condc

```
xyplot(y~Condc|pp, data=DF, ylab="jugement", type=c("p","r","g"),layout = c(3,3,4))
```



Fonction du package « lattice »
"type" indique que l'on veut les points et les droites de regression

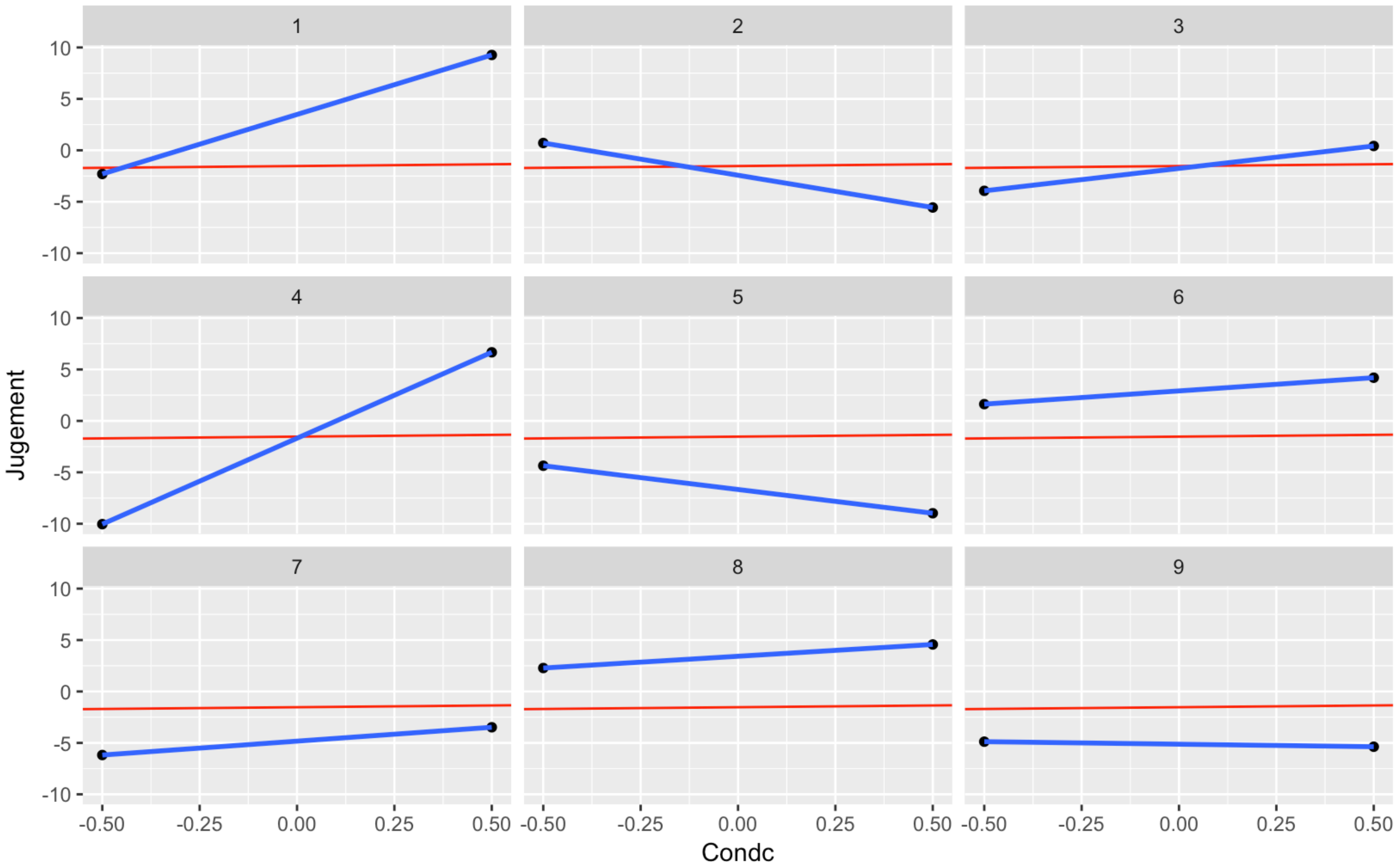
Attention, ce ne sont pas les pentes qui seraient utilisées en modèles mixtes

Modéliser les sources de variances

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \varepsilon_{ij}$$

Avec un tel plan expérimental, ε_{ij} peut renvoyer à 3 sources de variation :

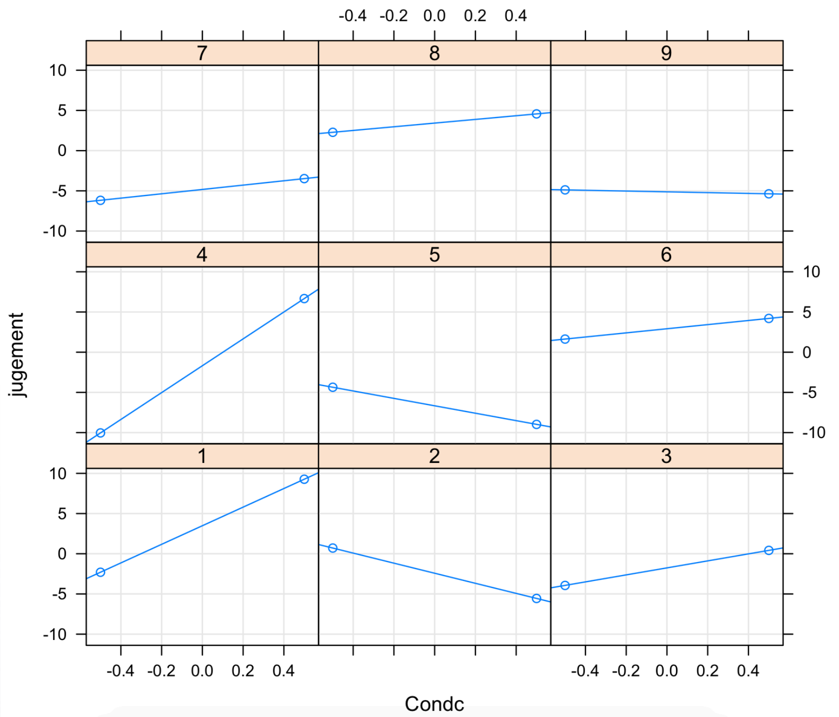
- α_0 peut varier d'un participant à un autre : Certains participants ont des jugements globalement plus positifs ou plus négatifs que la moyenne
- α_1 peut varier d'un participant à un autre : Certains participants sont plus ou moins sensibles aux conditions
- ce qu'il reste en plus de ces variations


1
2
3
4
5
6
7
8
9
10
5
0
-5
-10
Jugement
-0.50
-0.25
0.00
0.25
0.50
Condc

ATTENTION

	pp	Cond	y	Condc	stim
1	1	noir	-2.30	-0.5	stim1
2	1	blanc	9.27	0.5	stim2
3	2	noir	0.71	-0.5	stim1
4	2	blanc	-5.56	0.5	stim2
5	3	noir	-3.94	-0.5	stim1
6	3	blanc	0.42	0.5	stim2

...



Dans la vraie vie, nous n'utiliserions pas un modèle mixte pour ce type de situation, car nous n'avons qu'une observation par condition

Pour chaque participant

$$Y_{ij} = \beta_{0i} + \beta_{1i} C_{ij} + \varepsilon_{ij}$$

- Pour chaque participant (pour chaque *i*), nous pourrions estimer un intercept (β_{0i}) et un effet de la condition (blanc vs. noir; β_{1i})

- Nous avons donc ici un premier niveau, on parle d'effet de Niveau 1

Variabilité de l'intercept

$$\beta_{0i} = \alpha_0 + \mu_{0i}$$

- Nous pourrions maintenant estimer la variabilité de cet intercept (β_{0i}) d'un participant à un autre
- α_0 nous indique la valeur moyenne pour cet intercept
- μ_{0i} correspond à l'erreur de prédiction (ou l'ajustement nécessaire) pour chaque participant (BLUPS)

Variabilité de la pente

$$\beta_{1i} = \alpha_1 + \mu_{1i}$$

- Nous pourrions également estimer la variabilité de la pente (β_{1i}) d'un participant à un autre
- α_1 nous indique la valeur moyenne pour cette pente
- μ_{1i} correspond à l'erreur de prédiction (ou l'ajustement nécessaire) pour chaque participant (un autre BLUPS)

Un modèle général

Niveau 1

$$Y_{ij} = \beta_{0i} + \beta_{1i} C_{ij} + \varepsilon_{ij}$$

Niveau 2

$$\beta_{0i} = \alpha_0 + \mu_{0i} \qquad \beta_{1i} = \alpha_1 + \mu_{1i}$$

$$Y_{ij} = (\alpha_0 + \mu_{0i}) + (\alpha_1 + \mu_{1i}) C_{ij} + \varepsilon_{ij}$$

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \boxed{\mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}}$$

Modèles mixtes

$$Y_{ij} = \underbrace{\alpha_0 + \alpha_1 C_{ij}}_{\text{Effets fixes}} + \underbrace{\mu_{0i} + \mu_{1i} C_{ij}}_{\text{Effets aléatoires}} + \varepsilon_{ij}$$

- Effet fixe : effet d'une variable dont les niveaux peuvent être reproduits et forment un ensemble fini
- Effet aléatoire : effet d'une variable dont les niveaux sont échantillonnés aléatoirement à partir d'une population plus importante

➡ Les modèles mixtes possèdent à la fois des effets fixes et des effets aléatoires (souvent plusieurs)

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Estimation du modèle

$$Y_{ij} = \boxed{\alpha_0 + \alpha_1 C_{ij}} + \boxed{\mu_{0i} + \mu_{1i} C_{ij}} + \varepsilon_{ij}$$

- Utilisation du package le plus populaire pour les modèles mixtes : lme4
- Avec le DF en format long, nous utilisons la fonction lmer :

```
fit.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF,
            control=lmerControl(check.nobs.vs.nRE="ignore"))
```

- Pour avoir les résultats :

```
summary(fit.lmer)
```

Résultats (lme4)

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fit.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + Condc + (1 + Condc | pp)
   Data: DF
REML criterion at convergence: 344.3
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 pp       (Intercept)  4.028   2.007
          Condc       22.291   4.721    0.13
 Residual             10.222   3.197
Number of obs: 60, groups:  pp, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept)  -1.5280     0.5519  -2.768
Condc         0.3467     1.1935   0.290
```

A noter : lorsque nous déclarons une variable, le "1" devient inutile.

Ainsi, "`1 + Condc`" est équivalent à "`Condc`"

Méthodes d'estimation

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

- La méthode d'estimation habituelle, la méthode des moindres carrés (ou Least Square Estimation, LSE), ne permet pas de modéliser plusieurs sources d'erreurs

- Les modèles mixtes reposent donc sur la méthode du maximum de vraisemblance (Maximum Likelihood Estimation, MLE) ou par défaut le Restricted Maximum Likelihood (REML)

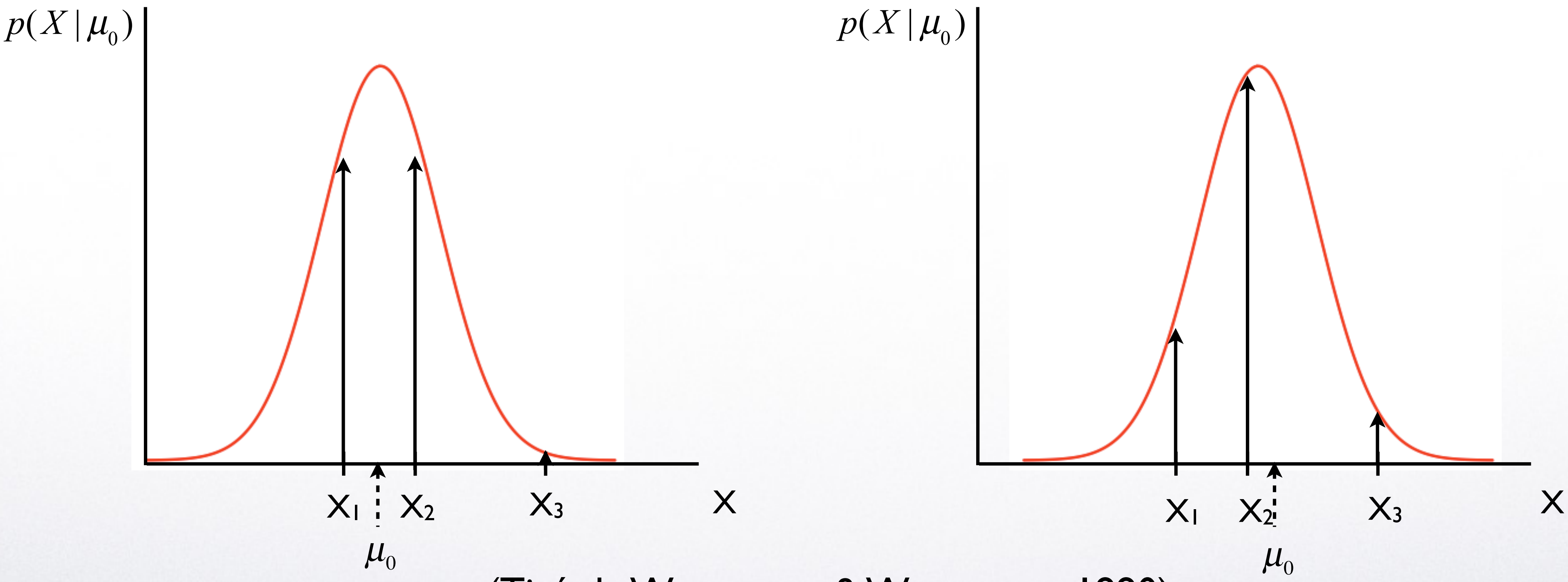
Pour utiliser MLE => `lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF,REML=FALSE)`

Méthodes d'estimation

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

- Avec le LSE : nous cherchons les paramètres permettant la description la plus exacte possible des données observées
- Avec le MLE : nous cherchons à deviner les paramètres de la population ayant le plus de chance d'avoir produit les données observées

Exemple de MLE



(Tiré de Wonnacott & Wonnacott, 1990)

La MLE procède par itérations pour trouver la valeur la plus probable pour le paramètre à estimer (ici la moyenne)

Résultats (lme4)

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + Condc + (1 + Condc | pp)
   Data: DF
REML criterion at convergence: 344.3
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 pp       (Intercept)  4.028   2.007
          Condc       22.291   4.721    0.13
 Residual             10.222   3.197
Number of obs: 60, groups:  pp, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept)  -1.5280     0.5519  -2.768
Condc         0.3467     1.1935   0.290
```

Indices d'ajustement du modèle

Effets aléatoires

Effets fixes

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Effets fixes : interprétation

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  -1.5280     0.5519  -2.768
Condc         0.3467     1.1935   0.290
```

Les paramètres estimés pour les effets fixes s'interprètent exactement comme d'habitude :

- a_0 = - 1.52, intercept c'est-à-dire la prédiction pour une valeur 0 de Condc.

  Condc étant un contraste, il s'agit de la prédiction pour un niveau moyen

- a_1 = 0.35, la pente, c'est-à-dire de combien change la prédiction lorsque nous augmentons d'une unité sur Condc.

  Comme nous avons une unité de différence entre les deux conditions (-0.5 pour noir et 0.5 pour blanc), il s'agit de la différence de moyennes : l'évaluation pour le visage blanc est supérieur de 0.35

Effets fixes : tests

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  -1.5280     0.5519  -2.768
Condc         0.3467     1.1935   0.290
```

- La valeur du *t* pour l'effet de la condition est identique à celle que nous avons trouvée avec l'ANOVA à un facteur intra
- lmer ne nous donne pas de ddl et de valeur *p* (voir Baayen et al., 2008)
- Nous disposons de plusieurs solutions, plusieurs estimations possibles pour estimer la significativité
  - Solution rapide : un t supérieur à 2 est significatif
  - F test avec approximations de Kenward-Roger ou de Satterthwaite
  - ...

Effets fixes : App. Satterthwaite

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

- Nous devons activer le package « lmerTest »
- Recréer notre modèle : `fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)`
- Utiliser la fonction `summary(fitA.lmer)`

```
Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  -1.5280     0.5519 29.0000  -2.768  0.00971 **
Condc         0.3467     1.1935 29.0000   0.290  0.77353
```

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Une variable aléatoire, un effet fixe

- Solution classique : Anova intra
- Modèle mixte : modéliser les erreurs
  - Concepts de base
  - Déclaration du modèle
  - Effets fixes
  - Effets aléatoires

Effets aléatoires

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 pp       (Intercept)  4.028   2.007
          Condc       22.291   4.721    0.13
 Residual             10.222   3.197
```

Le tableau des effets aléatoires nous donne les variances (et SD) des différents effets aléatoires (ne donne pas les moyennes car elles sont de 0)

- La variance estimée de l'effet aléatoire pour l'intercept ($\sigma^2_{\mu_{0i}}$) est de 4.028
- La variance estimée de l'effet aléatoire pour la pente ($\sigma^2_{\mu_{1i}}$) est de 22.291
- La variance estimée pour l'erreur résiduelle ($\sigma^2_{\varepsilon_{ij}}$) est de 10.222

Variances et ajustements

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 pp       (Intercept)  4.028   2.007
          Condc       22.291   4.721    0.13
 Residual             10.222   3.197
```

- Le tableau des effets aléatoires liste les variances et non les ajustements (μ_{0i} et μ_{1i}) réellement appliqués aux observations
- Nous pouvons obtenir ces valeurs d'ajustements avec la fonction ranef (pour random effects)
- On appelle ces valeurs d'ajustements les Best Linear Unbiased Predictors (BLUPS)

Ajustements et BLUPS

`ranef(fitA.lmer)` :

```
$pp
   (Intercept)       Condc
1   2.38539614  6.16677321
2  -0.50379726 -3.49808211
3  -0.03469231  2.06991259
4   0.20445334  8.49223984
5  -2.33746133 -2.92429700
6   1.98516719  1.45210693
...
```

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

$$\hat{Y}_{ij} = -1.53 + 0.35 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij}$$

$$\hat{Y}_{ij} = (-1.53 + \mu_{0i}) + (0.35 + \mu_{1i}) C_{ij}$$

Pour pp = 1 : $\hat{Y}_{1j} = (-1.53 + 2.39) + (0.35 + 6.17) C_{1j}$

$$\hat{Y}_{1j} = 0.86 + 6.51 C_{1j}$$

=> dans R, on retrouvera ces valeurs avec la fonction `coef(fitA.lmer)`

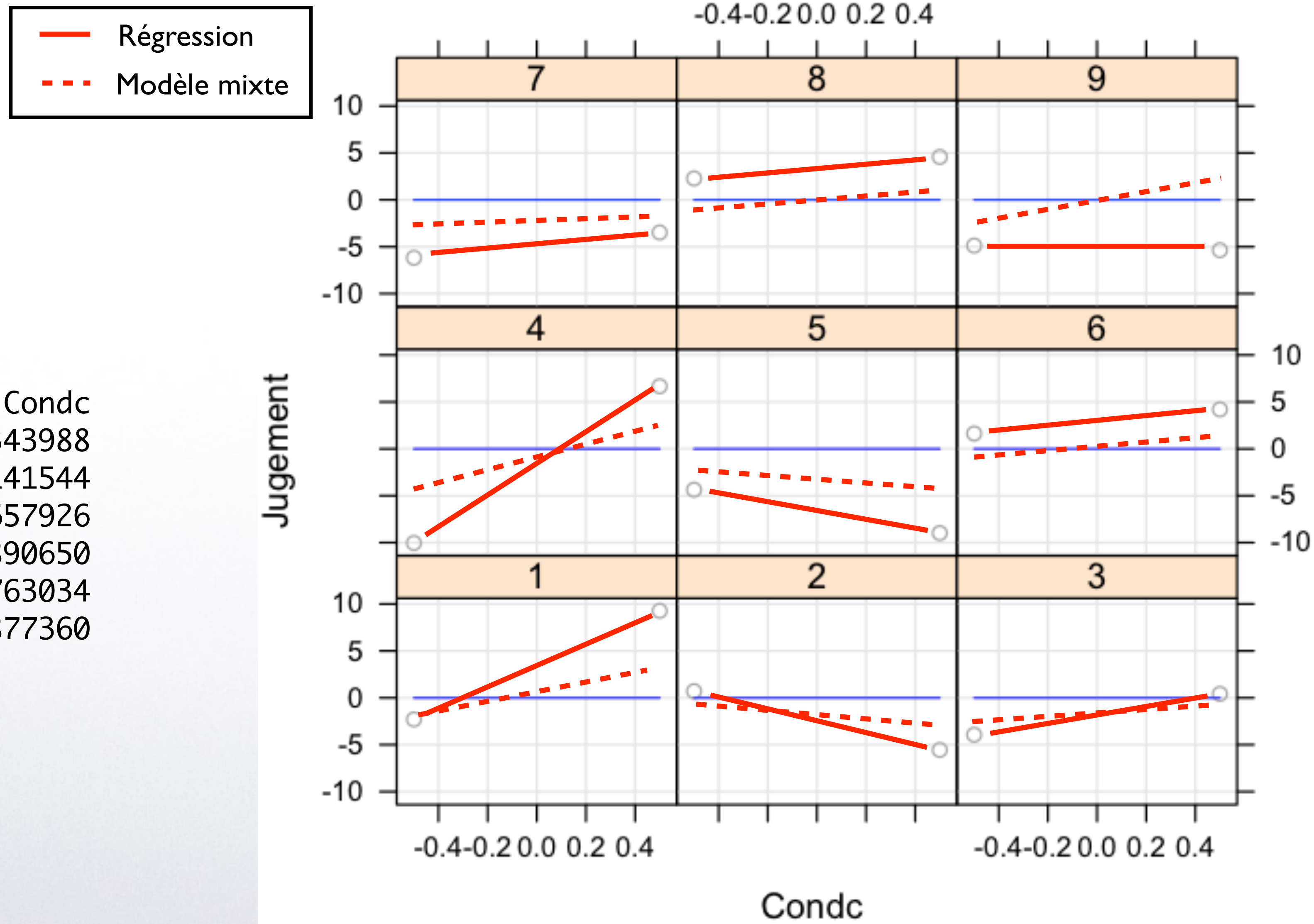
- Pour le Participant 1, l'ajustement pour l'intercept (μ_{0i}) = 2.39
  On applique donc un ajustement de 2.39 pour cette observation
- Pour ce même participant, l'ajustement pour la pente (μ_{1i}) = 6.17
  On applique donc un ajustement de 6.17 pour cette observation

```
xylowess.fnc(y ~ Condc | pp, data = DF, ylab = "Jugement",layout = c(3,3,4))
```

avec le package “langageR”

```
Fixed effects:
            Estimate
(Intercept)  -1.5280
Condc         0.3467
```

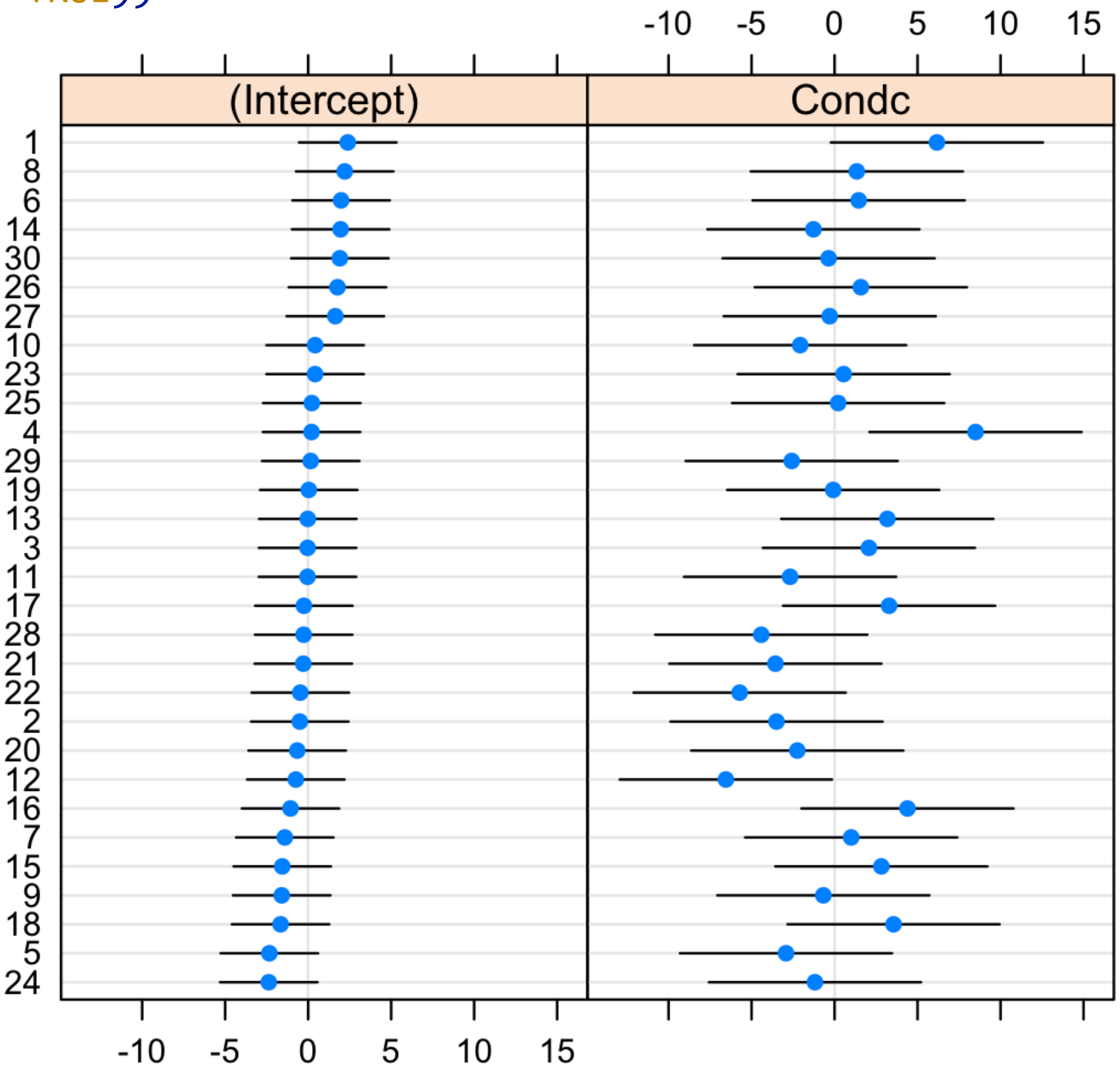
```
coef(fitA.lmer)
 $pp
   (Intercept)      Condc
 1   0.8573961  6.51343988
 2  -2.0317973 -3.15141544
 3  -1.5626923  2.41657926
 4  -1.3235467  8.83890650
 5  -3.8654613 -2.57763034
 6   0.4571672  1.79877360
 ...
```



```r
dotplot(ranef(fitA.lmer, condVar = TRUE))
```

Intervalles de confiance à 95% pour les BLUPS

```
$pp
   (Intercept)       Condc
1   2.38539614  6.16677321
2  -0.50379726 -3.49808211
3  -0.03469231  2.06991259
4   0.20445334  8.49223984
5  -2.33746133 -2.92429700
6   1.98516719  1.45210693
...
```



Si la fonction dotplot ne fonctionne pas, activer le package lattice

Corrélation des effets aléatoires

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
```

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 pp       (Intercept)  4.028   2.007
          Condc       22.291   4.721    0.13
 Residual             10.222   3.197
```

Corrélation entre l'intercept aléatoire et la pente aléatoire

- Dans la déclaration du modèle, le fait d'avoir l'intercept et la pente dans le même terme aléatoire `(1 + Condc | pp)` indique implicitement qu'on autorise qu'ils soient corrélés.
- Pour ne pas autoriser (et estimer) cette corrélation, il nous faut 2 termes :

=> `lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp), data=DF)`

Corrélation des effets aléatoires

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
fitC.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp), data=DF)
```

- Nous avons deux modèles dont un qui comporte une contrainte supplémentaire. Nous avons donc un modèle Augmenté et un modèle Contraint que nous pouvons comparer
- Parenthèse "logiciel" : cette comparaison des modèles donnerait un résultat identique :

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
fitC.lmer <- lmer(y ~ 1 + Condc + (1 + Condc || pp), data=DF)
```

=> autrement dit, le symbole "||" fixe toutes les corrélations à 0

Corrélation des effets aléatoires

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
fitC.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp), data=DF)
```

- Nous avons deux modèles dont un qui comporte une contrainte supplémentaire. Nous avons donc un modèle Augmenté et un modèle Contraint que nous pouvons comparer
- Pour cela, nous allons utiliser la fonction anova : `anova(fitA.lmer,fitC.lmer)`

```
Models:
fitC.lmer: y ~ Condc + (1 | pp) + (0 + Condc | pp)
fitA.lmer: y ~ 1 + Condc + (1 + Condc | pp)
          Df    AIC    BIC  logLik  Chisq Chi Df Pr(>Chisq)
fitC.lmer  5 357.27 367.74 -173.63
fitA.lmer  6 359.14 371.71 -173.57 0.1243      1     0.7244
```

Le modèle augmenté ne fait pas significativement mieux donc corrélation inutile

Nombre de paramètres estimés

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
fitC.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp), data=DF)
```

```
          Df    AIC    BIC  logLik  Chisq Chi Df Pr(>Chisq)
fitC.lmer  5 357.27 367.74 -173.63
fitA.lmer  6 359.14 371.71 -173.57 0.1243      1     0.7244
```

- Pour le modèle augmenté, nous estimons 6 paramètres : intercept, pente, intercept aléatoire, pente aléatoire, la corrélation entre intercept et pente aléatoire, et erreur
- Pour le modèle contraint, nous estimons 5 paramètres : intercept, pente, intercept aléatoire, pente aléatoire et erreur

⇨ 6-5 = 1 donc cela explique le ddl de cette comparaison

Remarque sur les comparaisons de modèles

- Pour tester des effets aléatoires, on utilise la fonction "anova" qui compare les "log likelihood" (LogLik)
- Pour tester des effets fixes, on utilise les approximations vues précédemment (KR, Satterthwaite, ...)

Effets aléatoires : tests

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp), data=DF)
```

Pour ce nouveau model (sans la corrélation), nous avons :

```
Random effects:
 Groups   Name        Variance Std.Dev.
 pp       (Intercept)  4.248   2.061
 pp.1     Condc       23.170   4.814
 Residual              9.782   3.128
```

Si nous voulons tester les effets aléatoires, nous pouvons utiliser le même type de comparaisons de modèles que précédemment :

- Pour tester l'intercept : `fitC.lmer <- lmer(y ~ 1 + Condc + (0 + Condc | pp), data=DF)`
- Pour tester la pente : `fitC.lmer <- lmer(y ~ 1 + Condc + (1 | pp), data=DF)`

COURS 2

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

RAPPEL

Un exemple (Judd et al., 2012)

- Une chercheuse présente 15 visages de blancs et 15 visages de noirs
- Pour chaque visage, un jugement sur une échelle de -20 à +20
- Souvent, dans cette situation, les chercheurs vont :
  - moyenner les réponses sur les 15 visages blancs et les 15 visages noirs
  - comparer ces deux moyennes en intra-sujet

→ Risque important d'erreurs de Type 1 avec cette analyse

Un exemple (Judd et al., 2012)

- Une chercheuse présente 15 visages de blancs et 15 visages de noirs
- Pour chaque visage, un jugement sur une échelle de -20 à +20

→ Nous disposons maintenant d'un outil permettant de traiter ce type de plan sans moyenner les items

Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

Structure des données

```
xtabs(~pp+stim,DF)
```

On peut voir que les participants sont totalement croisées avec les conditions

pp \ stim	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Structure des données

```
xtabs(~stim+Cond,DF)
```

Par contre, les stimuli sont nichés (nested) à l'intérieur des conditions. Chaque stimuli n'est que dans une condition (blanc vs noir)

Pas un problème pour R tant que les stimuli portent des noms distinctifs (et pas par exemple, 1, 2, 3 à l'intérieur des conditions)

```
        Cond
stim     blanc noir
  stim1      0   30
  stim10     0   30
  stim11     0   30
  stim12     0   30
  stim13     0   30
  stim14     0   30
  stim15     0   30
  stim16    30    0
  stim17    30    0
  stim18    30    0
  stim19    30    0
  stim2      0   30
  stim20    30    0
  stim21    30    0
  stim22    30    0
  stim23    30    0
  stim24    30    0
  stim25    30    0
  stim26    30    0
  stim27    30    0
  stim28    30    0
  stim29    30    0
  stim3      0   30
  stim30    30    0
  stim4      0   30
  stim5      0   30
  stim6      0   30
  stim7      0   30
  stim8      0   30
  stim9      0   30
```

RAPPEL

… un facteur fixe et un facteur aléatoire

$$Y_{ij} = \underbrace{\alpha_0 + \alpha_1 C_{ij}}_{\text{Effets fixes}} + \underbrace{\mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}}_{\text{Effets aléatoires (participants)}}$$

Ici nous n'avions que les participants en variable aléatoire

Un exemple (Judd et al., 2012)

$$Y_{ij} = \underbrace{\alpha_0 + \alpha_1 C_{ij}}_{\text{Effets fixes}} + \underbrace{\mu_{0j}}_{\text{Effet aléatoire stimuli}} + \underbrace{\mu_{0i} + \mu_{1i} C_{ij}}_{\text{Effets aléatoires participants}} + \varepsilon_{ij}$$

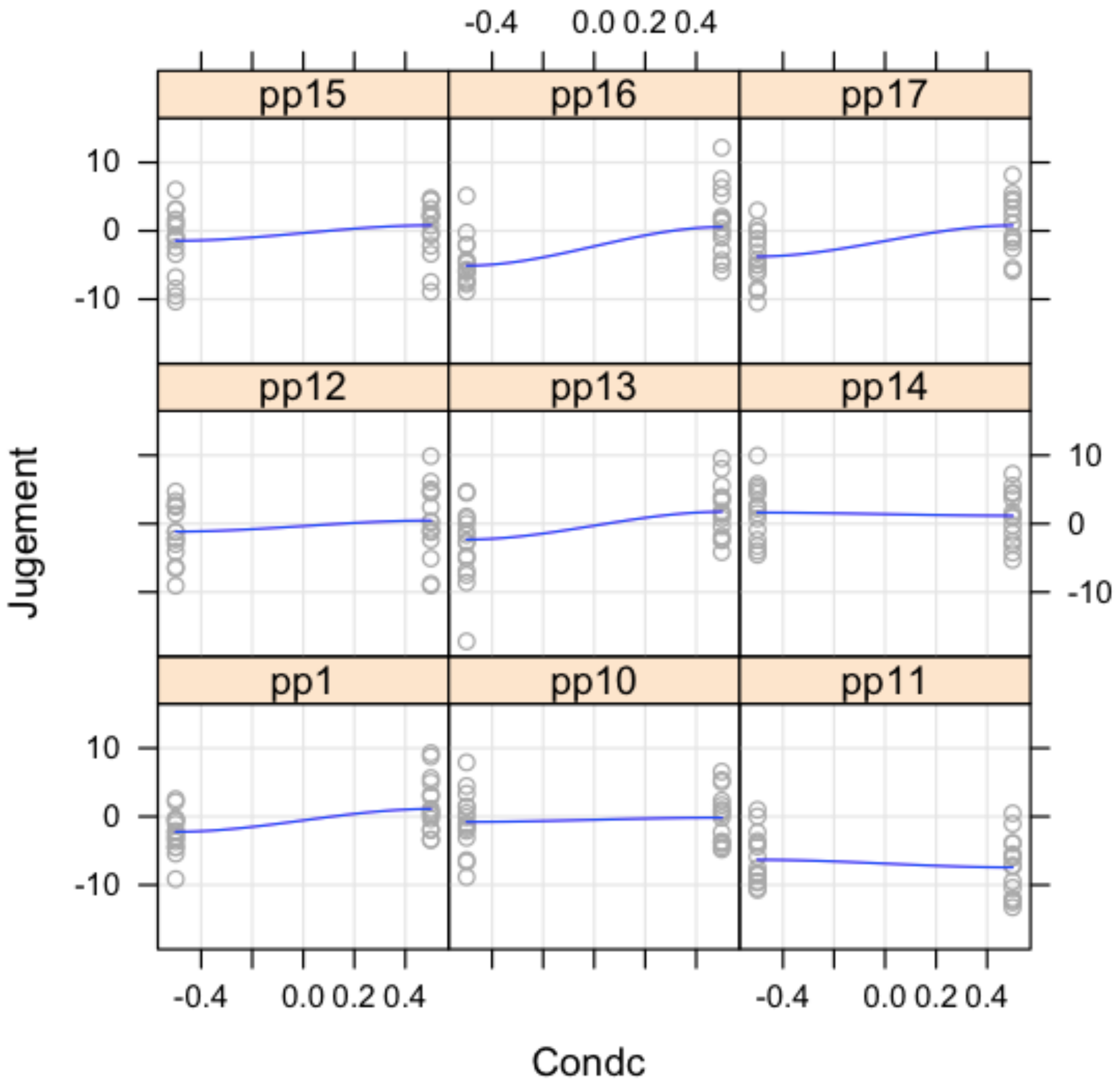
- En plus des paramètres de l'exemple précédent, nous avons maintenant un second effet aléatoire : les stimuli
- Nous ajoutons donc dans notre modèle un intercept aléatoire pour les stimuli (μ_{0j})

⇨ Pourquoi pas de pente aléatoire pour les stimuli ?

```
xylowess.fnc(y ~ Condc | pp, data = DF, ylab = "Jugement",layout = c(3,3,4))
```

Fonction du package
« languageR »

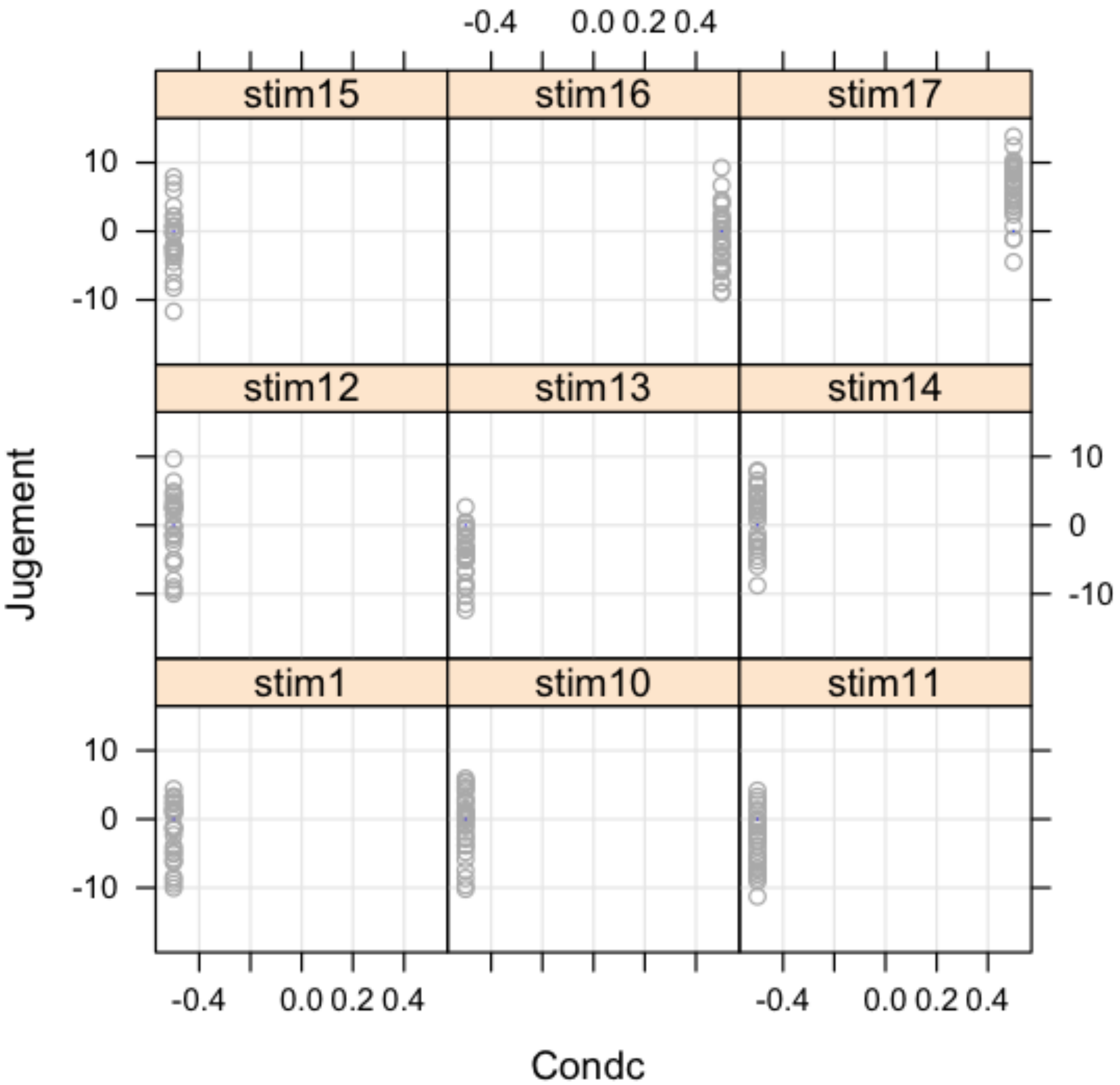
Estimer un intercept et une pente par participant a un sens



```
xylowess.fnc(y ~ Condc | stim, data = DF, ylab = "Jugement",layout = c(3,3,4))
```

Estimer un intercept par stimulus a un sens

Estimer une pente par stimulus n'aurait pas de sens



Déclaration du modèle

- Un facteur fixe et une variable aléatoire

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
```

- Un facteur fixe et deux variables aléatoires

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc  + (1 | stim) + (1 + Condc | pp), data=DF)
```

Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

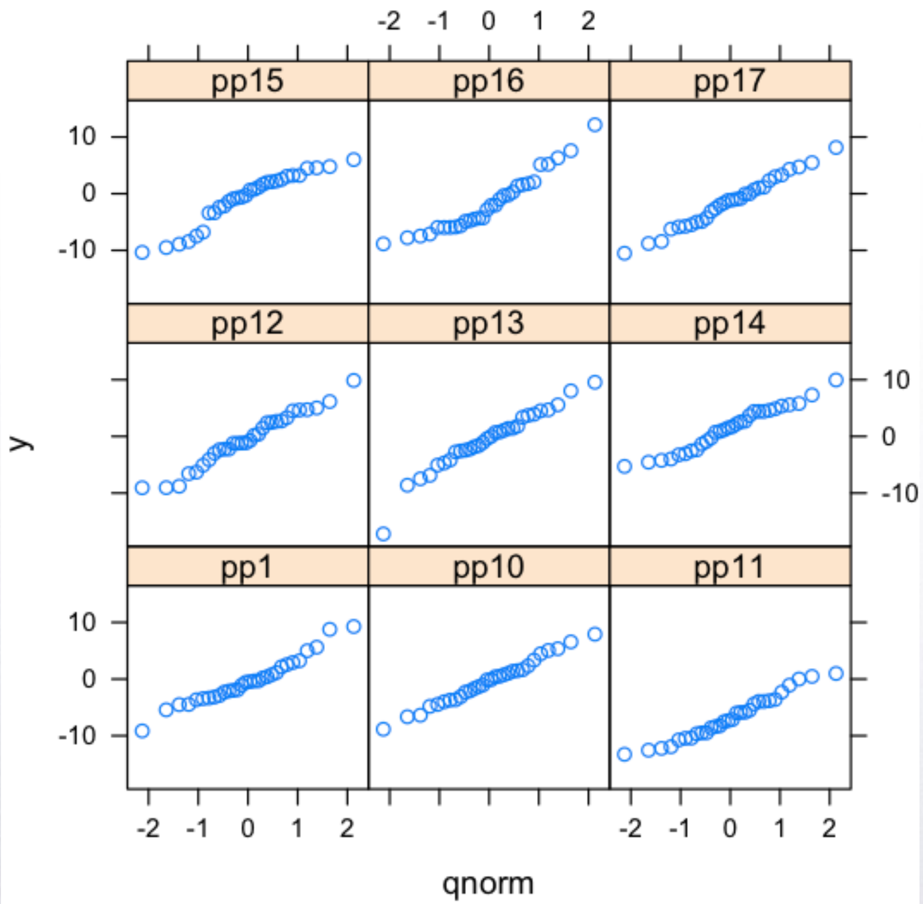
Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

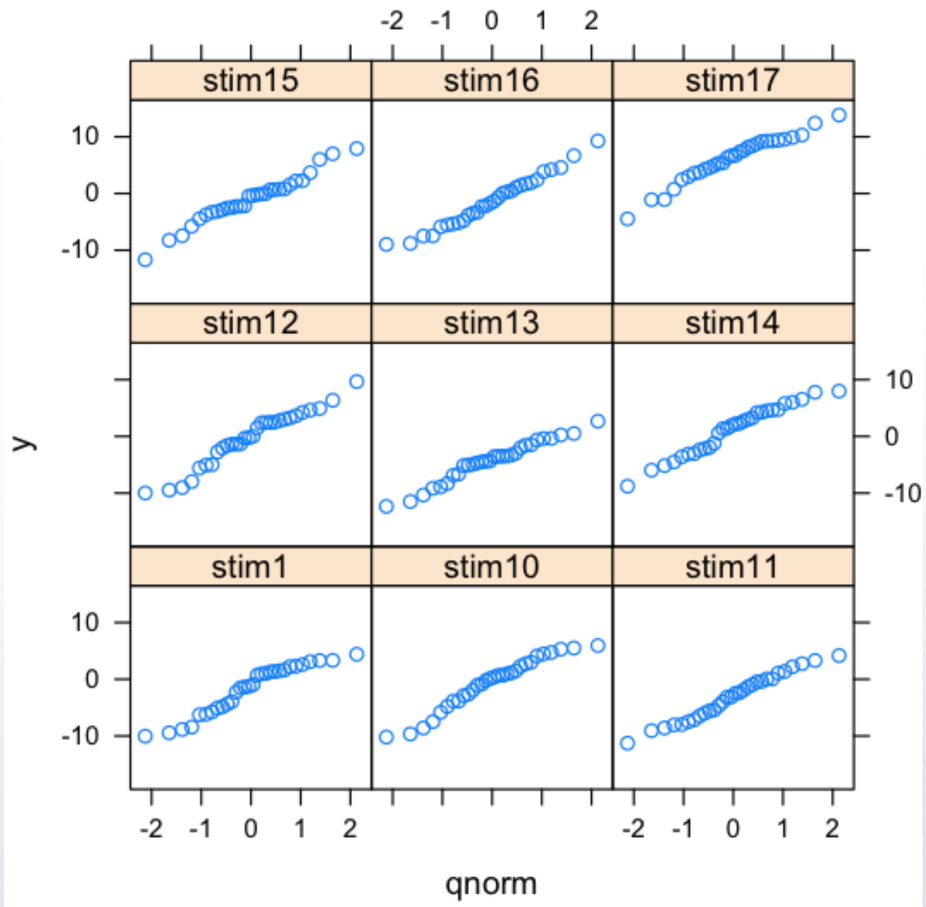
Etude des distributions

```
qqmath(~y|pp,data=DF,layout = c(3,3,4))
```

Fonction du package « lattice »




```
qqmath(~y|stim,data=DF,layout = c(3,3,4))
```

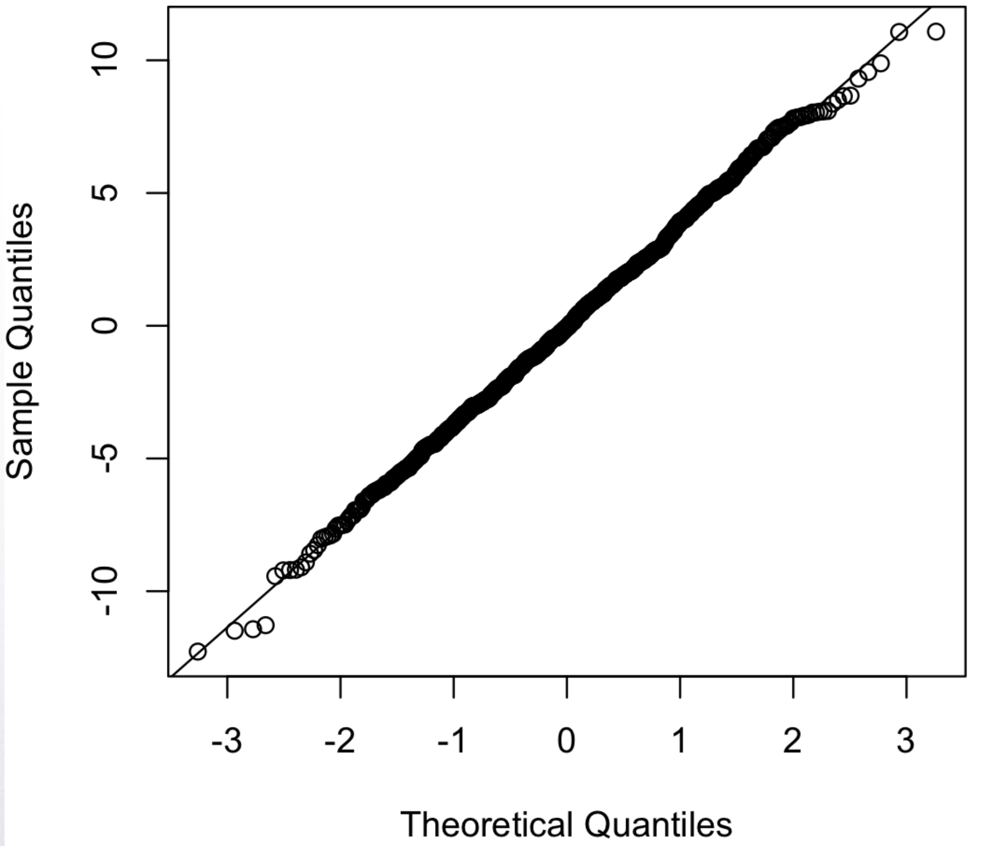


Etude des distributions : residus

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
```

```
qqnorm(residuals(fitA.lmer))
qqline(residuals(fitA.lmer))
```

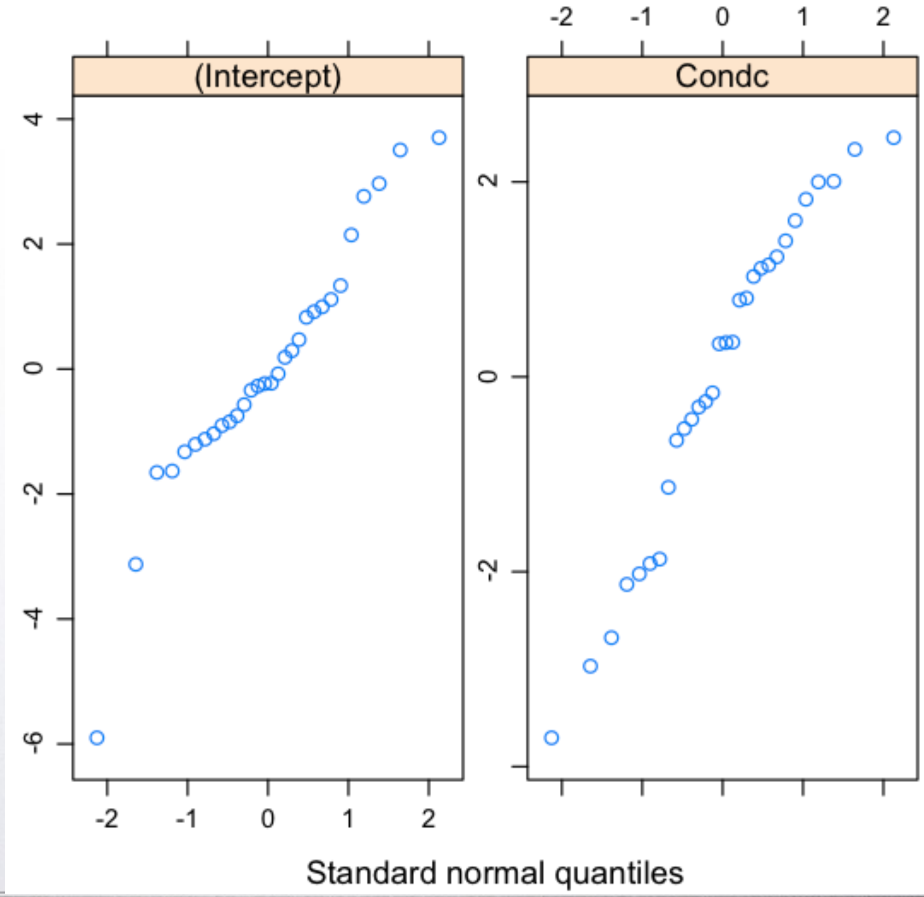


Etude des distributions : BLUPs

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
```

```
qqmath(ranef(fitA.lmer))
```

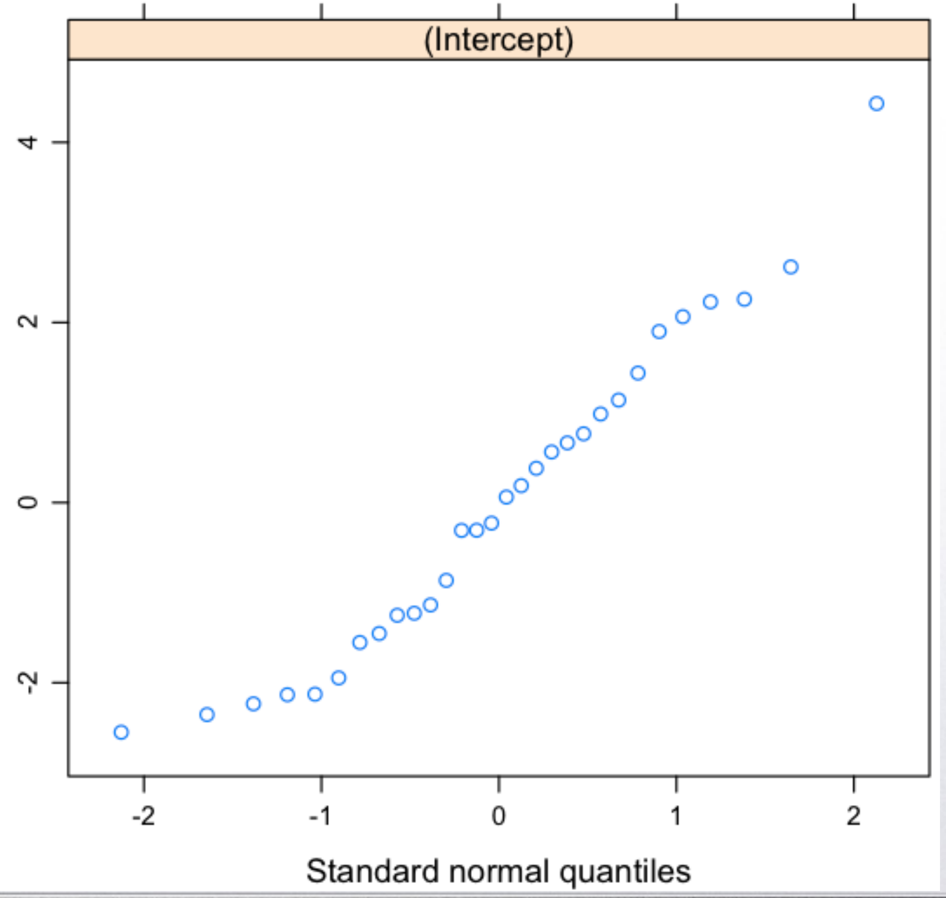


Etude des distributions : BLUPs

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
```

```
qqmath(ranef(fitA.lmer))
```



Etude des distributions : résidus

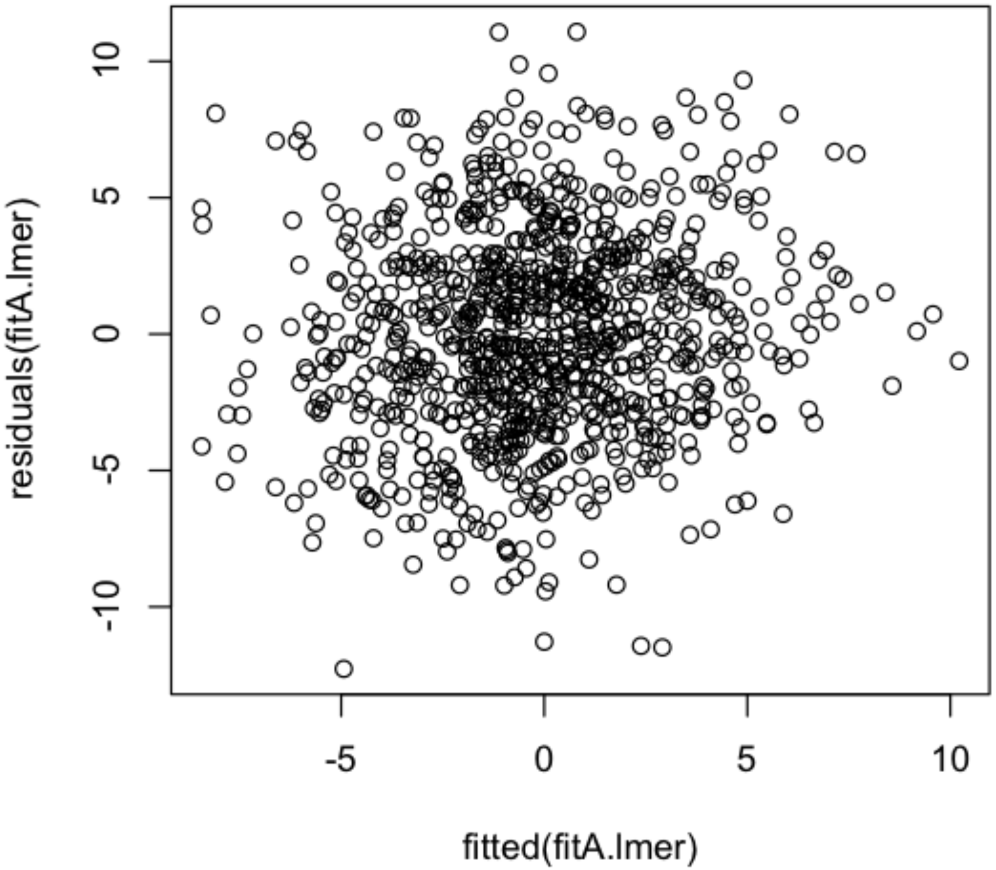
$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
```

```
plot(fitted(fitA.lmer),residuals(fitA.lmer))
```

valeurs prédites

résidus



Ou plus simplement :

```
plot(fitA.lmer)
```

Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

Résultats

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
summary(fitA.lmer)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim)
   Data: DF
REML criterion at convergence: 5179.045
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 pp       (Intercept)  4.294   2.072
          Condc        4.182   2.045    0.27
 stim     (Intercept)  3.670   1.916
 Residual             15.557   3.944
Number of obs: 900, groups: pp, 30; stim, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept)  -0.1804     0.5317  -0.339
Condc         2.5211     0.8354   3.018
```

Effets fixes : tests

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  -0.1804     0.5317  -0.339
Condc         2.5211     0.8353   3.018
```

=> 2.52 à l'avantage des blancs

- La valeur de *t* supérieure à 2 indique que l'effet de la condition est significatif
- Pour accéder aux valeurs p, nous pouvons utiliser à nouveau le package lmerTest `summary(fitA.lmer)`

```
Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  -0.1804     0.5317 50.4600  -0.339   0.7358
Condc         2.5211     0.8354 38.5100   3.018   0.0045 **
```

Importance du biais

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
Fixed effects:
            Estimate Std. Error t value
(Intercept)  -0.1804     0.5317  -0.339
Condc         2.5211     0.8353   3.018
```

- Test sans prendre en compte la variance liée aux stimuli

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp), data=DF)
Fixed effects:
            Estimate Std. Error t value
(Intercept)  -0.1804     0.4005  -0.450
Condc         2.5211     0.4566   5.521
```

(à noter que cet effet est identique en ANOVA)

Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

Deux variables aléatoires, un effet fixe

- Structure et choix des variables aléatoires
- Etude des distributions
- Résultats effets fixes
- Résultats effets aléatoires

Effets aléatoires : tests de la corrélation

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 + Condc | pp) + (1 | stim), data=DF)
fitC.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp) + (1 | stim), data=DF)
```

- Pour comparer ces modèles, utilisons la fonction anova : `anova(fitA.lmer,fitC.lmer)`

```
          Df    AIC    BIC  logLik  Chisq Chi Df Pr(>Chisq)
fitC.lmer  6 5194.4 5223.2 -2591.2
fitA.lmer  7 5195.1 5228.7 -2590.5 1.3021      1     0.2538
```

- Le modèle avec corrélation ne fait donc pas significativement mieux

Effets aléatoires : tests

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp) + (1 | stim), data=DF)
```

- Pour tester les effets aléatoires, nous utilisons les 3 modèles contraints :
  - `fitC1.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp), data=DF)`
  - `fitC2.lmer <- lmer(y ~ 1 + Condc + (0 + Condc | pp) + (1 | stim), data=DF)`
  - `fitC3.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (1 | stim), data=DF)`

- Les comparaisons du modèle A avec ces modèles C sont toutes significatives
- Nous pouvons donc garder ce modèle

Effets aléatoires : tests

$$Y_{ij} = \alpha_0 + \alpha_1 C_{ij} + \mu_{0j} + \mu_{0i} + \mu_{1i} C_{ij} + \varepsilon_{ij}$$

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp) + (1 | stim), data=DF)
```

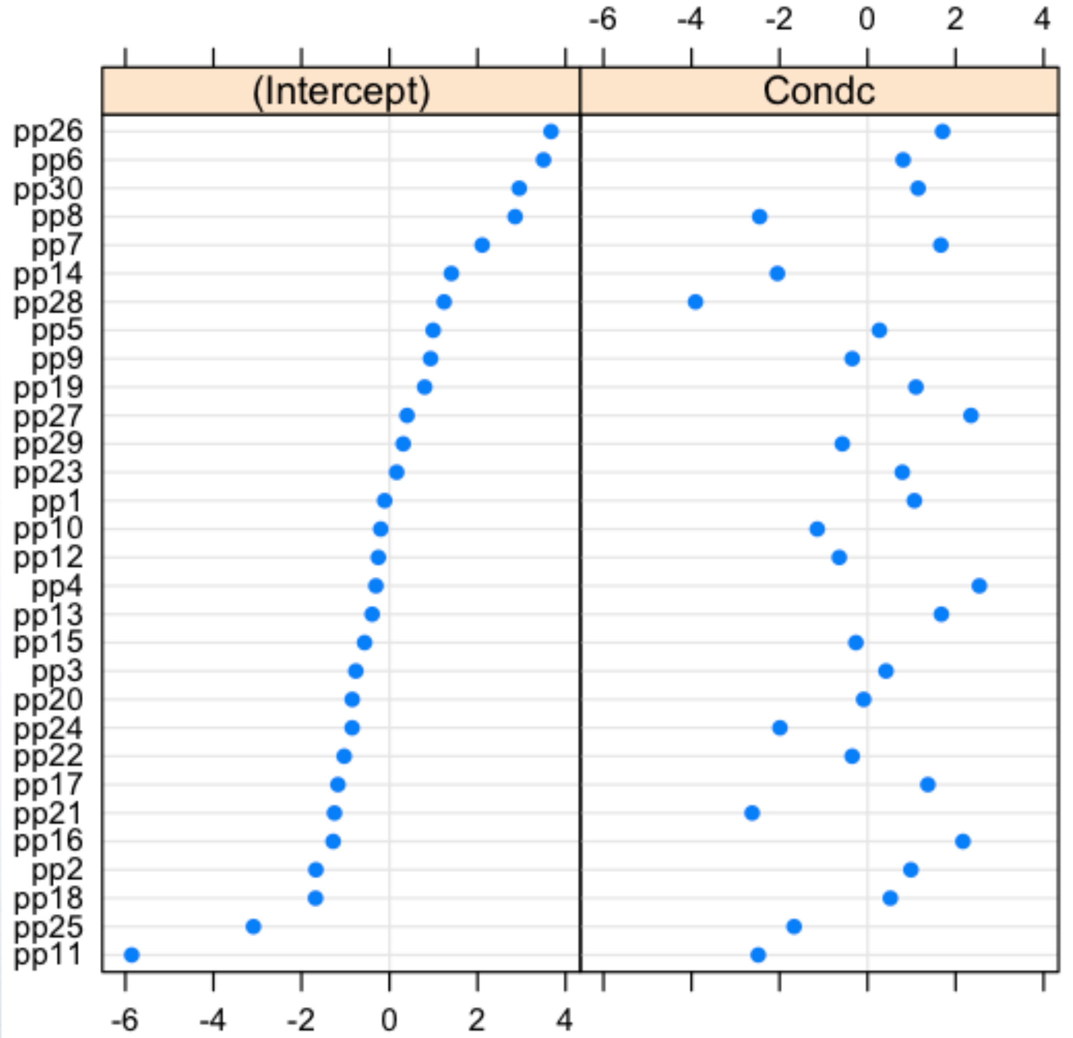
```
Random effects:
 Groups   Name        Variance Std.Dev.
 pp       (Intercept)  4.2940  2.0722
 pp       Condc        4.1823  2.0451
 stim     (Intercept)  3.6706  1.9159
 Residual             15.5572  3.9443
Number of obs: 900, groups: pp, 30; stim, 30
```

Les trois variances testées sont donc significatives :

- les participants varient en terme de niveau global de jugement
- les participants varient en terme de sensibilité à la condition
- le niveau de jugement varie en fonction des stimuli

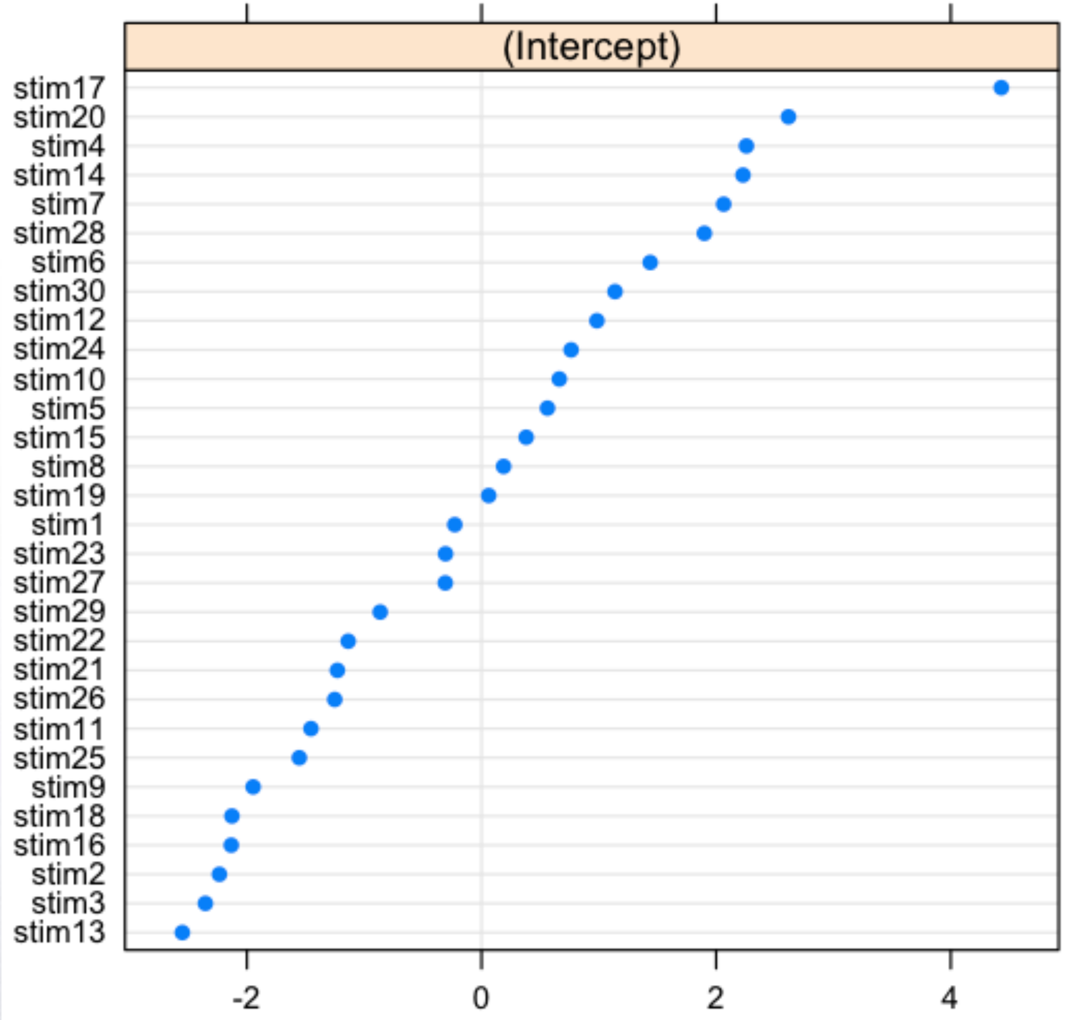
Effets aléatoires participants : BLUPS

```
dotplot(ranef(fitA.lmer))
```



Effets aléatoires stimuli : BLUPS

```
dotplot(ranef(fitA.lmer))
```



Effets fixes : tests

```
fitA.lmer <- lmer(y ~ 1 + Condc + (1 | pp) + (0 + Condc | pp) + (1 | stim), data=DF)
```

- Test des effets fixes avec ce modèle allégé `summary(fitA.lmer)`

```
 Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  -0.1804     0.5317 50.4600  -0.339   0.7358
Condc         2.5211     0.8354 38.5100   3.018   0.0045 **
```

Rédaction (exemple)

We used R (R Core Team, 2012) and lme4 (Bates, Maechler & Bolker, 2012) to perform a linear mixed effects analysis of the relationship between pitch and politeness. As fixed effects, we entered politeness, gender, and their interaction term into the model. As random effects, we had intercepts for subjects and items, as well as by-subject and by-item random slopes for the effect of politeness. Visual inspection of residual plots did not reveal any obvious deviations from homoscedasticity or normality. P-values were obtained by Satterthwaite approximation with the lmerTest package (Kuznetsova, Brockhoff, & Christensen, 2015). The analysis revealed…

- L'idéal étant même d'indiquer la version de R et la version de lmer

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

Plan général

- Introduction
- Une variable aléatoire, un effet fixe
- Deux variables aléatoires, un effet fixe
- Deux variables aléatoires, deux effets fixes

Plan avec interaction (Correll et al.)

Plan avec interaction (Correll et al.)

- Correll et al. présentent 25 blancs et 25 noirs
- Chaque personne stimuli apparait une fois avec une arme et une fois sans arme
  - les participants voient donc les 4 cases du plan (couleur x arme)
  - les personnes stimuli sont soit blanches, soit noires mais apparaissent armées et non armées
- Pour chaque essai, les participants doivent indiquer le plus vite possible s'il faut tirer ou non

Deux variables aléatoires, deux effets fixes

- Choix du modèle
- Etude des effets aléatoires
- Etude des effets fixes

Deux variables aléatoires, deux effets fixes

- Choix du modèle
- Etude des effets aléatoires
- Etude des effets fixes

Effets fixes

- VI_1 : la personne stimuli est noire (-0.5) ou blanche (0.5)

```
DF2$racec <- -0.5*(DF2$race=="black") + 0.5*(DF2$race=="white")
```

- VI_2 : la personne stimuli porte une arme (-0.5) ou non (0.5)

```
DF2$objectc <- -0.5*(DF2$object=="gun") + 0.5*(DF2$object=="nogun")
```

- Les auteurs prédisent une interaction :

  lorsque la cible est armée les participants seront plus rapides pour répondre « tirer » si la personne cible est noire plutôt que blanche

  lorsque la cible n'est pas armée les participants seront plus rapides pour répondre « ne pas tirer » quand la cible est blanche plutôt que noire

Effets aléatoires

$$Y_{ij} = \alpha_0 + \alpha_1 Ra_{ij} + \alpha_2 Ob_{ij} + \alpha_1 Ra_{ij} * Ob_{ij} + \mu_{0i} + \mu_{1i} Ra_{ij} + \mu_{2i} Ob_{ij} + \mu_{3i} Ra_{ij} * Ob_{ij} + \mu_{0j} + \mu_{1j} Ob_{ij} + \varepsilon_{ij}$$

```
m0 <- lmer(logTime ~ racec*objectc + (racec*objectc|pp) + (objectc|person), data=DF2)
```

- Chaque participant voient les 4 cases du plan. Donc 4 effets aléatoires possibles : un intercept, une pente pour l'effet couleur, une pente pour l'effet arme et une pente pour l'interaction => `(racec*objectc|pp)`
- Chaque personne stimuli apparait une fois avec une arme et une fois sans arme, mais chaque stimuli est blanc OU noire. Donc 2 effets aléatoires possibles : un intercept et une pente pour l'effet arme => `(objectc|person)`

Deux variables aléatoires, deux effets fixes

- Choix du modèle
- Etude des effets aléatoires
- Etude des effets fixes

Deux variables aléatoires, deux effets fixes

- Choix du modèle
- Etude des effets aléatoires
- Etude des effets fixes

Sélection des effets aléatoires

Face à un modèle aussi complexe, nous pouvons utiliser la procédure de Bates, Kliegl, Vasishth et Baayen (2015) :

- On fait tourner le modèle total
- On fait tourner un modèle total sans corrélation. On repère les petites variances et on les teste
- On remet les corrélations et on les teste
- A chaque étape, on s'assure que le modèle n'est pas "over-specified"

Sélection des effets aléatoires

Nous allons avoir besoin du package "RePsychLing" qui n'est pas disponible sur CRAN. Pour ça, nous devons donc utiliser les commandes suivantes :

```
install.packages("devtools")
library(devtools)
devtools::install_github("dmbates/RePsychLing")
library(RePsychLing)
```

Sélection des effets aléatoires

$$Y_{ij} = \alpha_0 + \alpha_1 Ra_{ij} + \alpha_2 Ob_{ij} + \alpha_3 Ra_{ij} * Ob_{ij} + \mu_{0i} + \mu_{1i} Ra_{ij} + \mu_{2i} Ob_{ij} + \mu_{3i} Ra_{ij} * Ob_{ij} + \mu_{0j} + \mu_{1j} Ob_{ij} + \varepsilon_{ij}$$

```
m0 <- lmer(logTime ~ racec*objectc + (racec*objectc|pp) + (objectc|person), data=DF2)
summary(rePCA(m0))
```

=> PCA = Principal Component Analysis

```
$person
Importance of components:
                         [,1]   [,2]
Standard deviation     0.9126 0.4077
Proportion of Variance 0.8337 0.1663
Cumulative Proportion  0.8337 1.0000

$pp
Importance of components:
                         [,1]   [,2]      [,3] [,4]
Standard deviation     0.4054 0.2727 1.056e-05    0
Proportion of Variance 0.6885 0.3115 0.000e+00    0
Cumulative Proportion  0.6885 1.0000 1.000e+00    1
```

=> Signe d'une matrice "singulière" et d'un modèle "over-specified"

Sélection des effets aléatoires

Face à un modèle aussi complexe, nous pouvons utiliser la procédure de Bates, Kliegl, Vasishth et Baayen (2015) :

- On fait tourner le modèle total
- On fait tourner un modèle total sans corrélation. On repère les petites variances et on les teste
- On remet les corrélations et on les teste
- A chaque étape, on s'assure que le modèle n'est pas "over-specified"

Sélection des effets aléatoires

Face à un modèle aussi complexe, nous pouvons utiliser la procédure de Bates, Kliegl, Vasishth et Baayen (2015) :

- On fait tourner le modèle total
- On fait tourner un modèle total sans corrélation. On repère les petites variances et on les teste
- On remet les corrélations et on les teste
- A chaque étape, on s'assure que le modèle n'est pas "over-specified"

Sélection des effets aléatoires

```
m1 <- lmer(logTime ~ racec*objectc + (racec*objectc||pp) + (objectc||person), data=DF2)
```

`VarCorr(m1)` => extrait seulement les variances de "summary"

```
Groups   Name          Std.Dev.
person   objectc       0.106197
person.1 (Intercept)   0.051638
pp       racec:objectc 0.000000
pp.1     objectc       0.031366
pp.2     racec         0.000000
pp.3     (Intercept)   0.046969
Residual               0.118193
```

=> Notons l'importance théorique de cette variance non-significative

=> Test d'un m2 sans ces deux termes aléatoires

```
m2 <- lmer(logTime ~ racec*objectc + (objectc||pp) + (objectc||person), data=DF2)
anova(m1,m2)
```

=> Donne un test non-significatif donc nous pouvons nous passer de ces deux termes aléatoires

Sélection des effets aléatoires

```
m2 <- lmer(logTime ~ racec*objectc + (objectc||pp) + (objectc||person), data=DF2)

summary(rePCA(m2))

$person
Importance of components:
                         [,1]   [,2]
Standard deviation     0.8985 0.4369
Proportion of Variance 0.8088 0.1912
Cumulative Proportion  0.8088 1.0000

$pp
Importance of components:
                         [,1]   [,2]
Standard deviation     0.3974 0.2654
Proportion of Variance 0.6916 0.3084
Cumulative Proportion  0.6916 1.0000
```

=> Par ailleurs, les autres tests donne lieu a des effets significatifs, donc nous gardons les autres termes aléatoires

Sélection des effets aléatoires

Face à un modèle aussi complexe, nous pouvons utiliser la procédure de Bates, Kliegl, Vasishth et Baayen (2015) :

- On fait tourner le modèle total
- On fait tourner un modèle total sans corrélation. On repère les petites variances et on les teste
- On remet les corrélations et on les teste
- A chaque étape, on s'assure que le modèle n'est pas "over-specified"

Sélection des effets aléatoires

Face à un modèle aussi complexe, nous pouvons utiliser la procédure de Bates, Kliegl, Vasishth et Baayen (2015) :

- On fait tourner le modèle total
- On fait tourner un modèle total sans corrélation. On repère les petites variances et on les teste
- On remet les corrélations et on les teste
- A chaque étape, on s'assure que le modèle n'est pas "over-specified"

Sélection des effets aléatoires

```
m6 <- lmer(logTime ~ racec*objectc + (objectc|pp) + (objectc|person), data=DF2)
summary(m6)

Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 person   (Intercept) 0.0026644 0.05162
          objectc     0.0112833 0.10622  -0.32
 pp       (Intercept) 0.0022044 0.04695
          objectc     0.0009837 0.03136  -0.22
 Residual             0.0139697 0.11819
Number of obs: 3502, groups:  person, 50; pp, 36
```

=> Attention également aux corrélations trop fortes (-1 ou 1)

`m7 <-lmer(logTime ~ racec*objectc + (objectc||pp) + (objectc|person), data=DF2)`

`anova(m6,m7)` => Test non-significatif donc corrélation inutile

`m8 <-lmer(logTime ~ racec*objectc + (objectc||pp) + (objectc||person), data=DF2)`

`anova(m7,m8)` => Test significatif donc corrélation utile

=> Nous gardons donc le Modèle 7

Sélection des effets aléatoires

```
m7 <- lmer(logTime ~ racec*objectc + (objectc||pp) + (objectc|person), data=DF2)

summary(m7)
```

```
Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 person   (Intercept) 0.0026650 0.05162
          objectc     0.0112778 0.10620  -0.32
 pp       objectc     0.0009846 0.03138
 pp.1     (Intercept) 0.0022062 0.04697
 Residual             0.0139696 0.11819
```

```
summary(rePCA(m7))
```

```
$person
Importance of components:
                         [,1]   [,2]
Standard deviation     0.9121 0.4076
Proportion of Variance 0.8336 0.1664
Cumulative Proportion  0.8336 1.0000
```

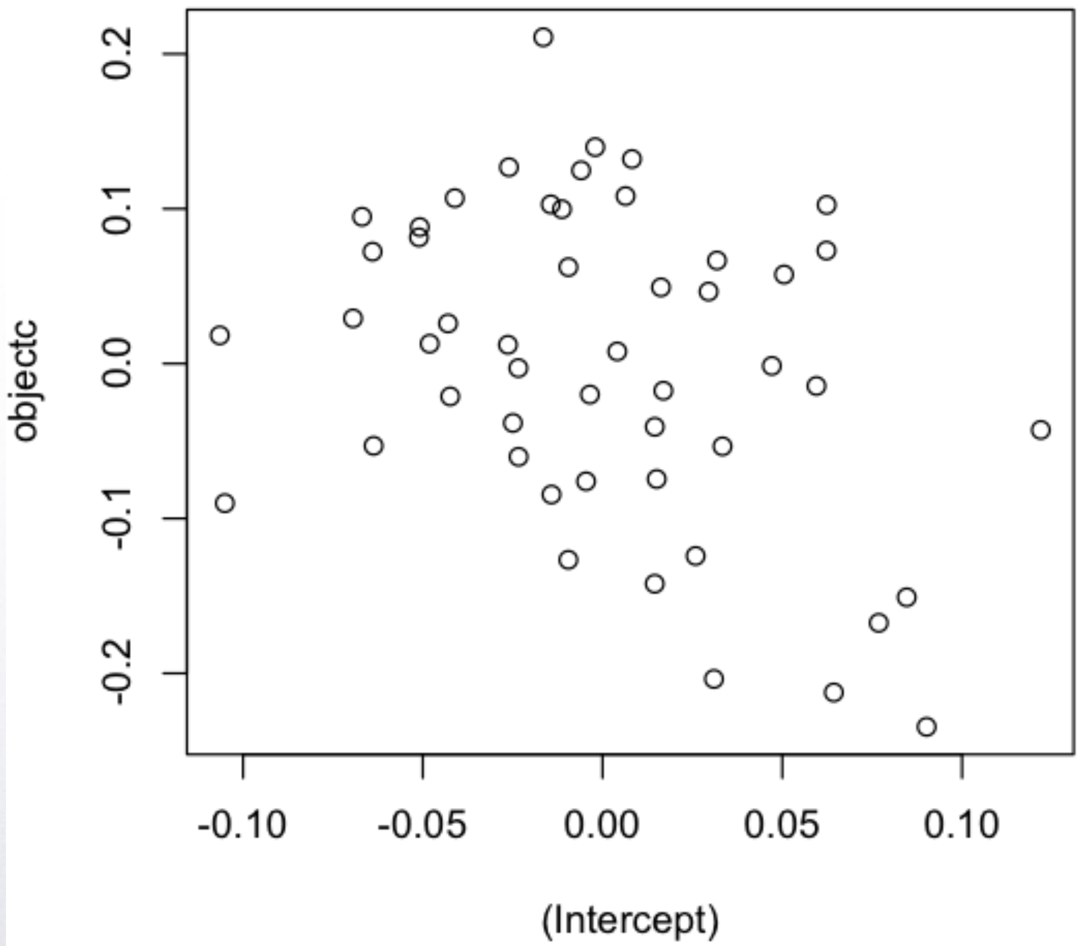
```
$pp
Importance of components:
                         [,1]   [,2]
Standard deviation     0.3974 0.2655
Proportion of Variance 0.6914 0.3086
Cumulative Proportion  0.6914 1.0000
```

Etude de la corrélation pour "person"

```
m7 <- lmer(logTime ~ racec*objectc + (objectc||pp) + (objectc|person), data=DF2)
```

```
ranefPerson<-ranef(m7)$person
plot(ranefPerson,xlab="(Intercept)")
```

- La corrélation (*r* = -0.32) entre l'intercept et la pente pour les personnes stimuli (person) est significative ($p < .03$)
- Elle indique que plus les réponses sont lentes pour une personne cible, moins l'effet arme est grand

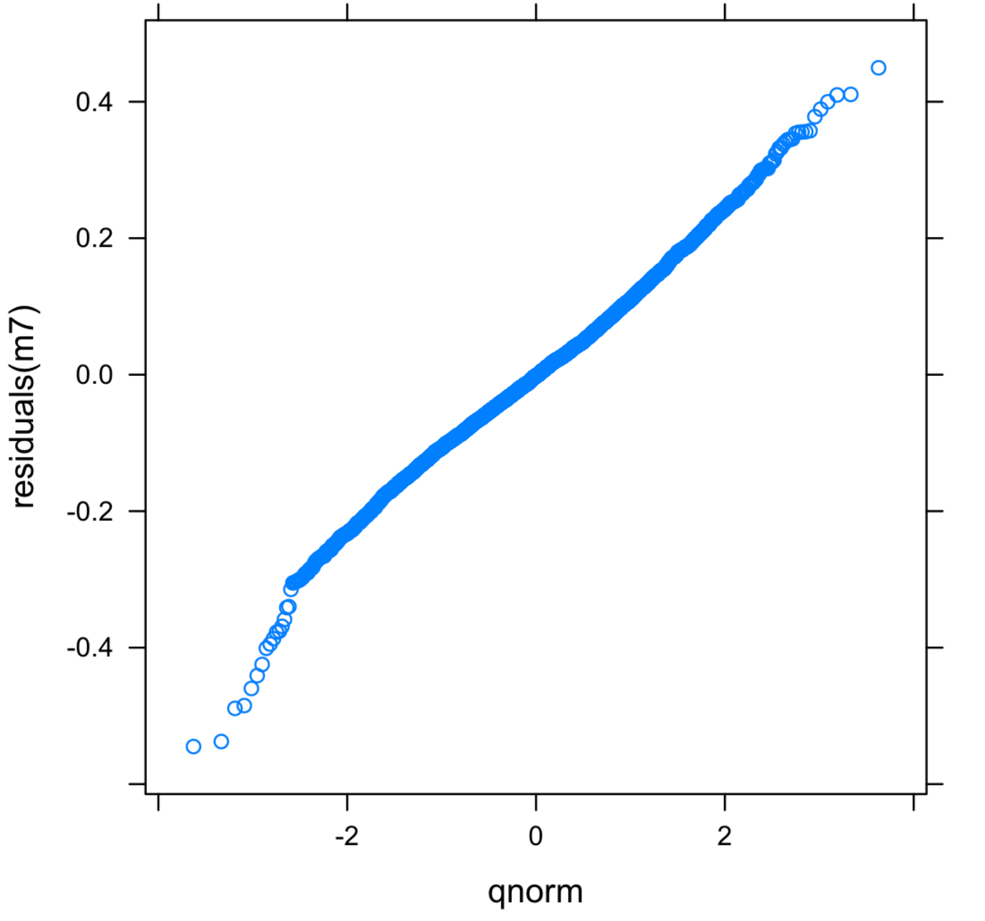


Etude des distributions

```
m7 <- lmer(logTime ~ racec*objectc + (1+objectc||pp) + (1+objectc|person), data=DF2)
```

```
qqmath(residuals(m7))
```

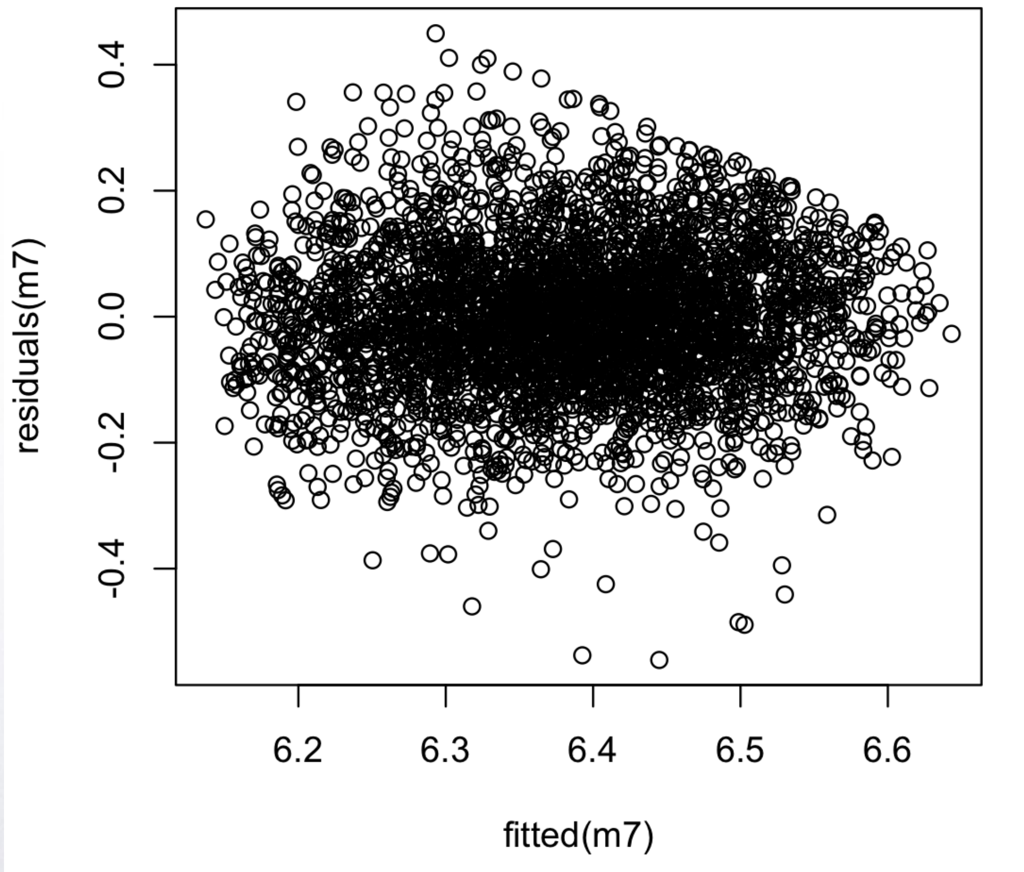
Notons que la distribution n'est pas tout à fait normale



Etude des distributions

```
m7 <- lmer(logTime ~ racec*objectc + (1+objectc||pp) + (1+objectc|person), data=DF2)
```

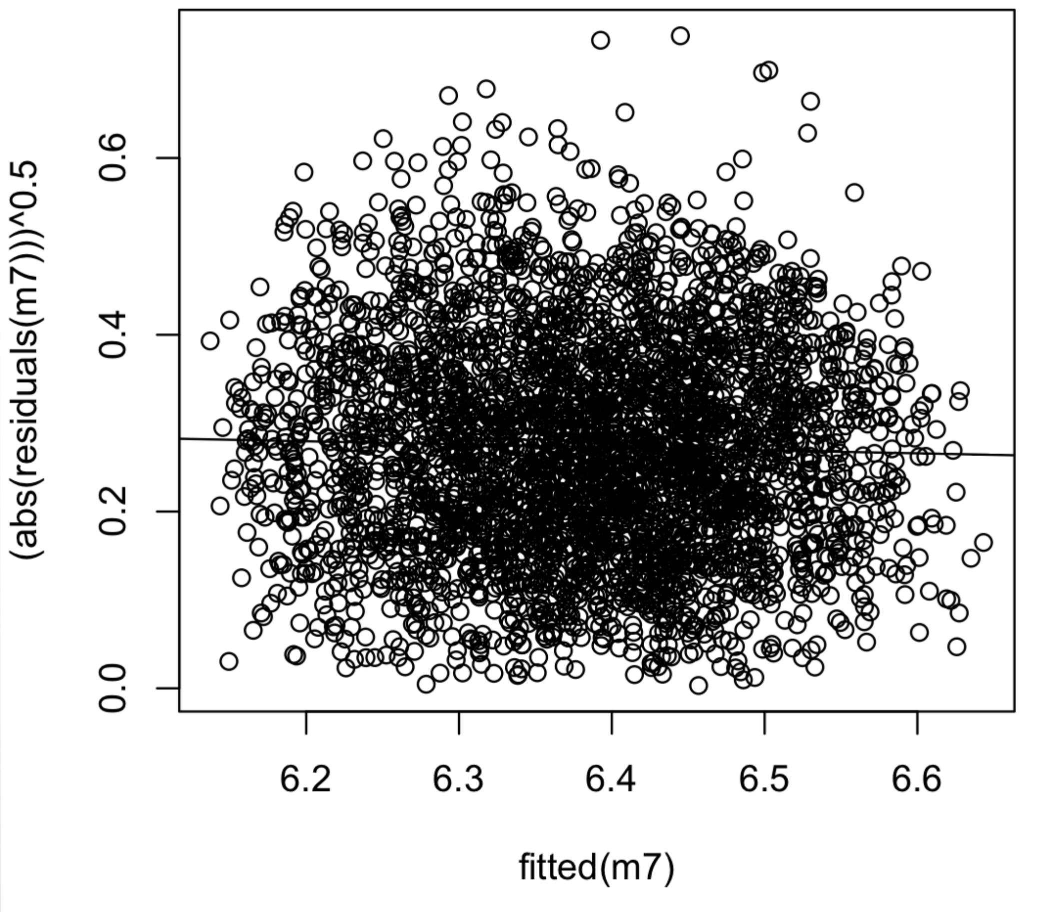
```
plot(fitted(m7),residuals(m7))
```



Etude des distributions

```
m7 <- lmer(logTime ~ racec*objectc + (1+objectc||pp) + (1+objectc|person), data=DF2)
```

```
plot(fitted(m7),(abs(residuals(m7)))^.5)
abline(lm((abs(residuals(m7)))^.5~fitted(m7)))
```



Deux variables aléatoires, deux effets fixes

- Choix du modèle
- Etude des effets aléatoires
- Etude des effets fixes

Deux variables aléatoires, deux effets fixes

- Choix du modèle
- Etude des effets aléatoires
- Etude des effets fixes

Résultats

```
m7 <- lmer(logTime ~ racec*objectc + (1+objectc||pp) + (objectc|person), data=DF2)
```

```
Fixed effects:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.38692    0.01089   586.5  < 2e-16 ***
racec         -0.00192    0.01514    -0.1    0.900
objectc        0.10591    0.01640     6.5 2.55e-08 ***
racec:objectc -0.06036    0.03109    -1.9    0.058 .
```

$$\widehat{Y}_{ij} = 6.387 + -0.002 Ra_{ij} + 0.106 Ob_{ij} - 0.060 Ra_{ij} * Ob_{ij}$$ (effets fixes uniquement)

$$\widehat{Y}_{ij} = (6.387 + -0.002 Ra_{ij}) + (0.106 - 0.060 Ra_{ij}) Ob_{ij}$$

- a_2 = 0.106 : lorsqu'on augmente d'une unité sur objectc les RT augmentent de 0.106. Autrement dit, les participants sont plus lents lorsque l'objet n'est pas une arme.
- a_3 = -0.060 : l'effet précédent est tendanciellement moins vrai lorsque l'on augmente d'une unité sur la variable Couleur («Race» en anglais) donc moins vrai pour les personnages blancs

Résultats

```
lmer(logTime ~ racec*objectc + (1+objectc||pp) + (objectc|person), data=DF2)
```

```
Fixed effects:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.38692    0.01089   586.5  < 2e-16 ***
racec         -0.00192    0.01514    -0.1    0.900
objectc        0.10591    0.01640     6.5 2.55e-08 ***
racec:objectc -0.06036    0.03109    -1.9    0.058 .
```

$$\hat{Y}_{ij} = 6.387 + -0.002 Ra_{ij} + 0.106 Ob_{ij} - 0.060 Ra_{ij} * Ob_{ij}$$ (effets fixes uniquement)

$$\hat{Y}_{ij} = (6.387 + 0.106 Ob_{ij}) + (-0.002 - 0.060 Ob_{ij}) Ra_{ij}$$

- a_1 = -0.002 : lorsqu'on augmente d'une unité sur racec les RT diminuent de 0.002.
- a_3 = -0.060 : lorsque l'on passe de la condition Gun à la condition No Gun l'effet perd 0.06. Comme l'effet était quasiment 0 pour object = 0, comme attendu, nous avons une inversion de l'effet "objet"

Résultats

```
lmer(logTime ~ racec*objectc + (1+objectc||pp) + (objectc|person), data=DF2)
```

$$\hat{Y}_{ij} = 6.387 + -0.002 Ra_{ij} + 0.106 Ob_{ij} - 0.060 Ra_{ij} * Ob_{ij}$$

Black/No gun $\quad \hat{Y}_{ij} = 6.387 + -0.002(-0.5) + 0.106(0.5) - 0.060(-0.5 * 0.5) = 6.46$

Black/Gun $\quad \hat{Y}_{ij} = 6.387 + -0.002(-0.5) + 0.106(-0.5) - 0.060(-0.5 * -0.5) = 6.32$

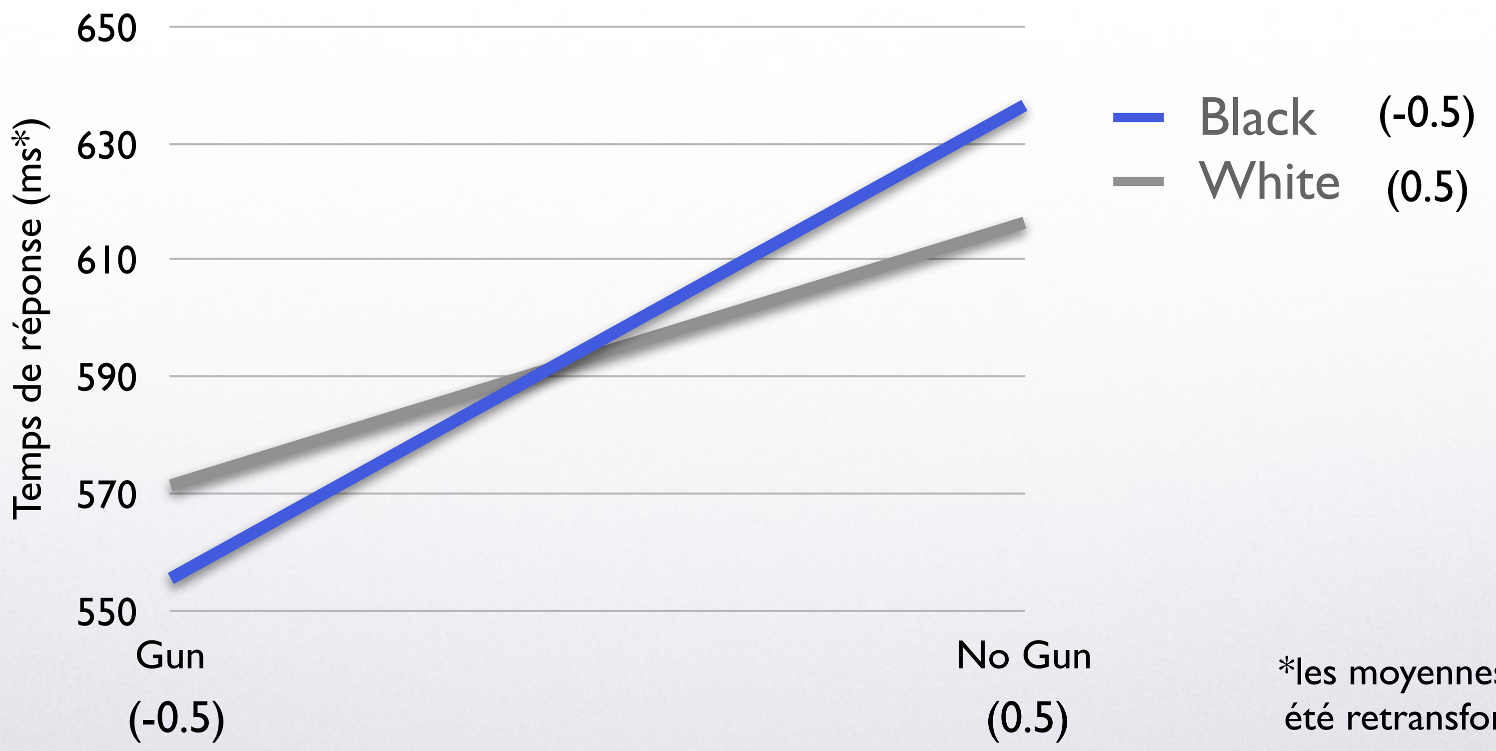
White/Gun $\quad \hat{Y}_{ij} = 6.387 + -0.002(0.5) + 0.106(0.5) - 0.060(0.5 * 0.5) = 6.42$

White/No gun $\quad \hat{Y}_{ij} = 6.387 + -0.002(0.5) + 0.106(-0.5) - 0.060(0.5 * -0.5) = 6.35$

Résultats

```
lmer(logTime ~ racec*objectc + (1+objectc||pp) + (objectc|person), data=DF2)
```

$$Y_{ij} = 6.387 + -0.002Ra_{ij} + 0.106Ob_{ij} - 0.060Ra_{ij} * Ob_{ij}$$



*les moyennes en log ont été retransformés en ms

Exercice 1

- Les sujets voient 8 mots
- 4 mots sont présentés avec un SOA court et 4 mots sont présentés avec un SOA long
- VD : Temps de réaction

=> Votre mission : tester l'effet du SOA

(les données sont déjà sur votre ordinateur, activez simplement le package languageR, le data frame s'appelle "quasif")

Exercice 2

- Les participants réalisent une tâche d'indiçage : ils doivent détecter une cible le plus rapidement possible
- VI_1 : la cible est précédée d'un indice valide (il apparait du même coté que la cible) ou invalide (coté opposé)
- VI_2 : la cible est une arme ou un objet neutre
- VD : Temps de réaction

=> Votre mission : tester si l'effet d'indiçage est plus marqué lorsque l'indice est une arme

(les données sont dans le fichier ArmesetAttention.txt sur le serveur du cours)