

# Mixed Effect Models in R

**Giorgio Arcara**

Neurophysiology Lab  
IRCCS San Camillo Hospital, Venice, Italy

June 18, 2024



# Models

“All models are wrong, but some are useful”  
*George Box*

# What are mixed effect models

Mixed effect models (or Mixed Models) are a powerful statistical tool that allows to deal with complex *dependencies* in the data, in a *modeling* framework.

Mixed effect models are an extension of linear models. As such, knowledge on linear model is required to properly use mixed models.

# Why “Mixed ”Models

They are called mixed models because they are able to handle both *fixed* and *random* effects.

# Fixed and Random Effects

There are many ways to explain the difference between fixed and random effects:

- **fixed effects** reflects variables you are manipulating or that you are interested into (e.g., Priming condition, Stimulus type).
- **random effects** are related to observations sampled “randomly ” from a larger population (e.g. Participants or Items). You are interested in taking into account their effects, but you are not really interested in them.
- ...

See also [here](#)

# Pros of using Mixed Effect Models

## Pros

With Mixed Effect models you can answer to more sophisticated scientific questions, and to handle more properly the data structure.

In many cases, Mixed Effect models have higher statistical *Power* than other analysis (e.g., ANOVA)

# Cons of Mixed Effect Models

## Cons

The main problem of using mixed models is related to their flexibility.

Very often there are many ways to *fit* a mixed models and it is difficult to understand what is the “best” way to perform the analysis (the same is not true for ANOVA, in which typically, given a design, there is only one way to perform the appropriate ANOVA).

Hence, with mixed models it is easier to perform a suboptimal (or even wrong) analysis.

# When to use Mixed Effect Models

Mixed effect models are perfect for these cases:

- You want to take into account both Subjects and Item variability.
- You are interested in both covariates and categorical variables, and also in their interactions.
- You have dichotomous outcome variable (Generalized Linear Mixed Effect Models).



# When the use Mixed Effect Models can be problematic

Mixed effect models may be difficult to handle in these cases:

- You have a complex factorial design.
- You have no idea on what you are doing.

# What to expect from this course

If you are not familiar with either R or Linear regression, it is very hard to properly understand mixed models. In particular understanding regressions is the knowledge that allow to interpret the **fixed effects** of mixed models.

You don't need to understand *everything* right now, but you need to grasp some concepts and know where to retrieve additional information.

# Interim note

Actually also t-tests and ANOVAs are linear models and can be (more clearly) understood under this perspective.

See references [here](#) and [here](#)

# What to expect from this course

This is a course on Mixed Effect Models in R

Working with R requires other additional knowledge (in **data wrangling**, basic input/output, plotting, etc.)

# What to expect from this course

This course I will refer to some working<sup>1</sup> .R scripts that you could use as reference to develop your mixed effects models, or as starting point to elaborate some topics.

You can download the course materials from here  
<https://github.com/giorgioarcara/mixed-effect-models>

---

<sup>1</sup>hopefully

# Some crucial points to keep in mind

Before starting, keep in mind these important points.

- ① Understanding Mixed Models requires understanding (well) linear regressions and other more basic aspects of statistics.
- ② Fitting a mixed model is almost never a one-shot process. It requires checking assumptions before, and then checking that the fit was good, adjusting if necessary.

# Filling the gaps (1)

There are many free resources online that will make you independent with R and mixed Models

- **Analyzing Linguistic Data Download**. An excellent Book by H. Baayen. It contains all the information you need. The Mixed Model part is a bit outdated.
- **Tutorials on Mixed Models** by Bodo Winter [Link].
- **GLMM wiki**. A comprehensive webpage maintained by Ben Bolker and other world experts on Mixed Models.
- **Google for your problem!**. If the answer is on Cross-Validated you are on the right track.
- **help**. Don't forget to use the `help` function in R.

## Filling the gaps (2)

- **Another Tutorial** [[Link](#)].
- **An interesting reading list**[<http://jakewestfall.org/blog/index.php/2015/06/20/reading-list-introduction-to-linear-mixed-models-for-cognitive-scientists/>]



## Filling the gaps (3)

There are two other things you can do to improve your ability in deal with mixed models for your analysis:

- ask to colleagues with more experience.
- get ideas from published papers with similar design.

# Learning mixed models

In general, consider learning mixed models as a neverending process, in which you can always improve and learn new things.

## Few words on Linear Regressions

With linear regression you *model* a response variable as a function of one predictor, using the formula of a straight line.

$$Y = \beta_0 + \beta_1 X_1 + \epsilon \quad (1)$$

the R syntax is a simplified version of the mathematical formula.

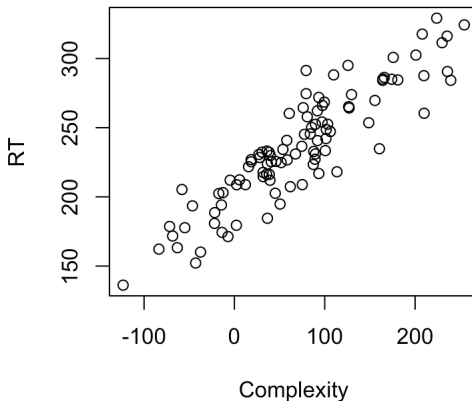
```
lm(y ~ x1)
```

```
lm(RT ~ Complexity, data=dat1) # a more realistic example
```

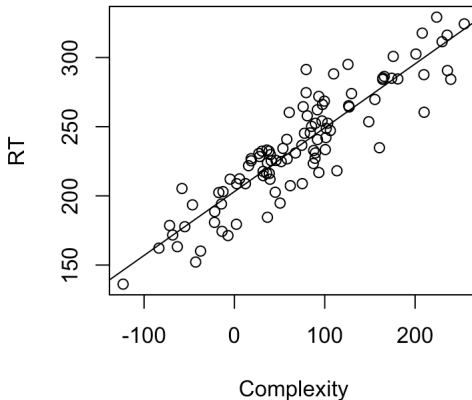
# Some regression terminology

- **Dependent variable:** the variable to be predicted from other variables.
- **Predictors:** the variables that are supposed to predict the dependent variable.
- **Betas/Coefficients:** together with the Intercept, the core of the result of the regression.
- **Fitted values:** the results of the regression (they lie in the regression line).
- **Residuals:** the discrepancy between fitted and observed values, they are related to the error of the model.

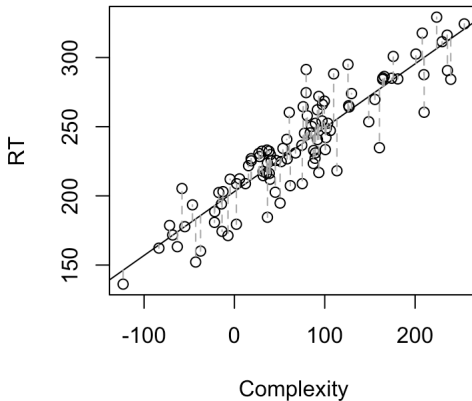
# Predictor and Dependent Variable



# Regression Model Fit



# Residuals



# One note about “Prediction ”in regression

Beware that, strictly speaking, regression does not tell us nothing about prediction in the sense of “forecasting ”or, even worse, about causality.

The term “prediction ”is used because predictors is a common term in regression.

The more appropriate term would be **association**



# Fitting Regressions in R

```
set.seed(220) # change the seed for different results

## SIMPLE LINEAR REGRESSION ##
Complexity = rnorm(100, mean = 80, sd = 10)
Error = rnorm(100, mean=0, sd = 8)
RT = 200 + 0.5*Complexity + Error
dat1 = data.frame(RT=RT, Complexity=Complexity)

mod1 = lm(RT~Complexity, dat1)
summary(mod1)
```

# Linear Regression Example (mod1)

Call:

```
lm(formula = RT ~ Complexity, data = dat1)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-16.9704	-4.0812	0.7874	4.4138	20.6332

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	209.83311	5.69333	36.856	< 2e-16 ***
Complexity	0.37735	0.07154	5.275	7.96e-07 ***

---

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

Residual standard error: 7.173 on 98 degrees of freedom

Multiple R-squared: 0.2211, Adjusted R-squared: 0.2132

F-statistic: 27.82 on 1 and 98 DF, p-value: 7.959e-07

# Interpreting results of regressions

Results show many things

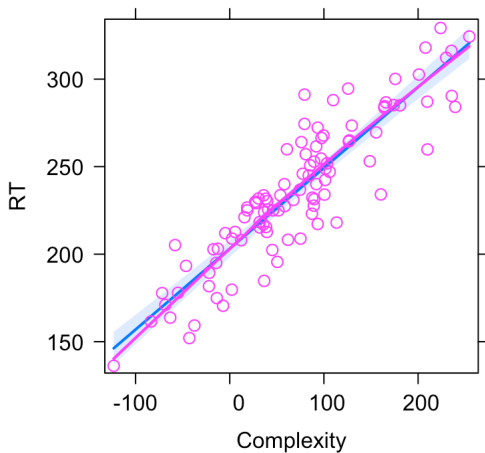
- The Estimate column reports both the Intercept and the coefficient (*beta*) for our predictor
- The p-values obtained from `summary(mod1)` tells if the coefficient for the predictor is significantly different from zero. item
- the  $R^2$  indicate the proportion of explained variance, which is an index of quality of fit for regressions.

The estimated regression equation is

$$RT = 209.833 + 0.377 * Complexity$$

# Simple Linear Regression

**Complexity effect plot**



Partial Effect of Complexity on RT

# Multiple Linear Regressions

With **multiple** linear regression you can *model* a response variable as a function of more than one predictor

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \epsilon \quad (2)$$

# Multiple Linear Regression (mod2)

Call:

```
lm(formula = RT ~ Complexity + Imageability, data = dat2)
```

Residuals:

Min	1Q	Median	3Q	Max
-20.478	-3.884	1.279	4.540	25.016

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	205.69685	7.03643	29.233	< 2e-16 ***
Complexity	0.35765	0.07629	4.688	9.01e-06 ***
Imageability	-0.92280	0.14699	-6.278	9.66e-09 ***

---

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

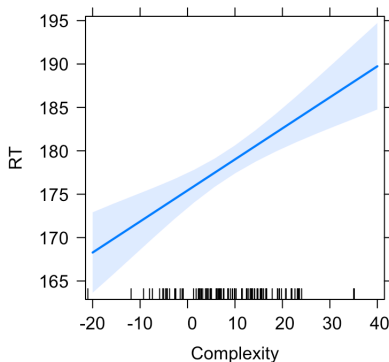
Residual standard error: 7.865 on 97 degrees of freedom

Multiple R-squared: 0.392, Adjusted R-squared: 0.3795

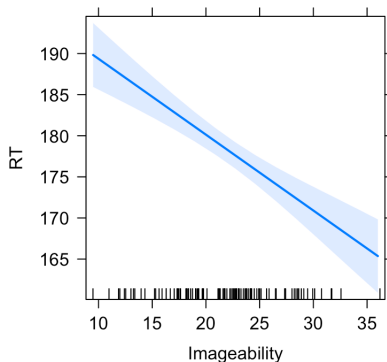
F-statistic: 31.27 on 2 and 97 DF, p-value: 3.309e-11

# Multiple Linear Regression

**Complexity effect plot**



**Imageability effect plot**



Partial Effects of Complexity and Imageability on RT

# Partial effects in Multiple Linear Regression

When plotting a partial effect, all other effects are kept to a given value (typically, the median).



# Effects in linear regressions

A peculiar aspect of regression is that the effect of each predictor is calculated *taking into account the effect of other predictors*.

(strictly speaking, each effect is estimated such to be *orthogonal*, that is uncorrelated, to all the others)

This is a very important aspect of regression also related to some problems (see Collinearity in these slides)

## Other important aspects of regressions

Regression can handle not only continuous variables, but also categorical variables (in R, *factors*). In this case the factor must be transformed using *dummy coding* ( [see here](#) or [here](#) ).

# Factors in Linear Regression (mod3)

Call:

```
lm(formula = RT ~ PrimeType + Imageability, data = dat3)
```

Residuals:

Min	1Q	Median	3Q	Max
-17.5095	-5.1034	-0.6084	5.1287	17.1625

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	201.3567	3.5101	57.365	< 2e-16 ***
PrimeTypePositive	6.0705	1.4776	4.108	8.33e-05 ***
Imageability	-2.0946	0.1734	-12.080	< 2e-16 ***

---

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

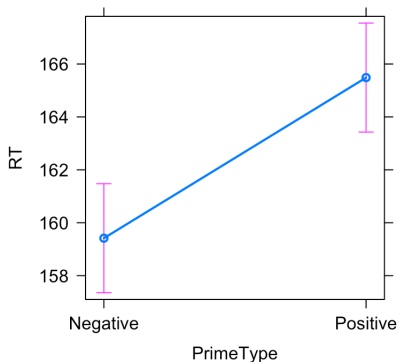
Residual standard error: 7.295 on 97 degrees of freedom

Multiple R-squared: 0.6087, Adjusted R-squared: 0.6006

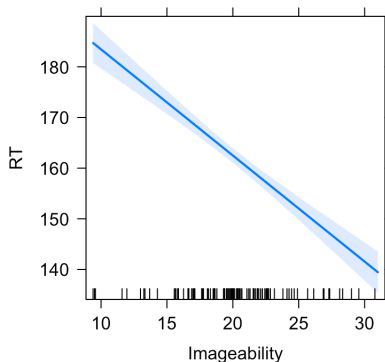
F-statistic: 75.43 on 2 and 97 DF, p-value: < 2.2e-16

# Factors in Linear Regression

**PrimeType effect plot**



**Imageability effect plot**



Partial Effects of PrimeType and Imageability on RT

# Interpreting results of regressions

Note that to properly check the significant factor you should use another function.

```
require(car)
```

```
Anova(mod3, type="III")
```

Anova Table (Type III tests)

Response: RT

	Sum Sq	Df	F value	Pr(>F)	
(Intercept)	1970.0	1	37.019	2.327e-08	***
PrimeType	898.2	1	16.879	8.328e-05	***
Imageability	7765.1	1	145.919	< 2.2e-16	***
Residuals	5161.9	97			

# Interpreting results of regressions

Broadly speaking, the Anova function actually tests whether the term contributes significantly to the model.

# Interaction between Factors (mod4)

Call:

```
lm(formula = RT ~ PrimeType * TaskDiff, data = dat4)
```

Residuals:

Min	1Q	Median	3Q	Max
-20.0864	-4.9877	-0.8893	4.4921	21.4090

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	202.152	1.574	128.419	< 2e-16 ***
PrimeTypePositive	9.949	2.226	4.469	2.15e-05 ***
TaskDiffEasy	4.685	2.226	2.105	0.0379 *
PrimeTypePositive:TaskDiffEasy	-6.790	3.148	-2.157	0.0335 *

---

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

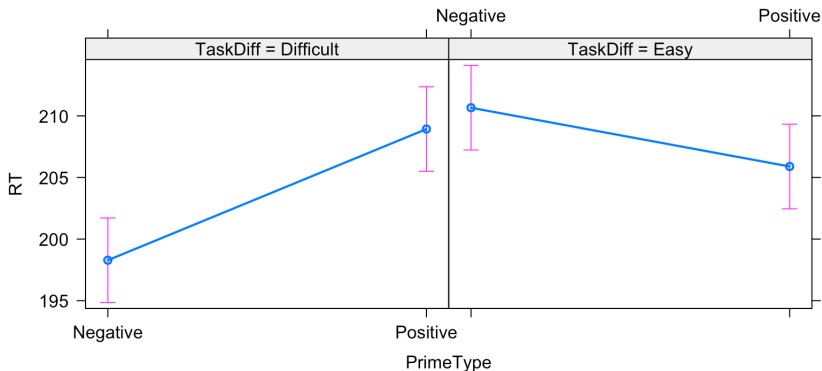
Residual standard error: 7.871 on 96 degrees of freedom

Multiple R-squared: 0.1909, Adjusted R-squared: 0.1656

F-statistic: 7.552 on 3 and 96 DF, p-value: 0.0001374

# Interaction between Factors

**PrimeType\*TaskDiff effect plot**



Partial Effects of interaction between PrimeType and Imageability on RT



# Interaction between Factors

## Beware!

To interpret significant results in the case of interactions of factors, you should not use the significance of coefficients.

The equivalent of testing the significance of the overall interaction is made again using the `Anova` function on the model, of the `car` package.

For post-hocs you can use the `emmeans` package, using `emmeans` to compare means and `emtrends` to compare slopes.

# Post-hocs of mod4

```
> pairs(emmeans(mod4, ~PrimeType*TaskDiff))
```

contrast	estimate	SE	df	t.ratio	p.value
Negative,Difficult - Positive,Difficult	-10.65	2.45	96	-4.355	0.0002
Negative,Difficult - Negative,Easy	-12.38	2.45	96	-5.061	<.0001
Negative,Difficult - Positive,Easy	-7.61	2.45	96	-3.111	0.0129
Positive,Difficult - Negative,Easy	-1.73	2.45	96	-0.707	0.8942
Positive,Difficult - Positive,Easy	3.04	2.45	96	1.243	0.6010
Negative,Easy - Positive,Easy	4.77	2.45	96	1.950	0.2144

P value adjustment: tukey method for comparing a family of 4 estimates

# Interaction between a Factor and a continuous variable (mod5)

Call:

```
lm(formula = RT ~ PrimeType * Imageability, data = dat5)
```

Residuals:

Min	1Q	Median	3Q	Max
-145.591	-33.166	3.205	32.017	121.188

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	218.992	35.925	6.096	2.27e-08 ***
PrimeTypePositive	-43.745	48.347	-0.905	0.36783
Imageability	-5.492	1.731	-3.172	0.00203 **
PrimeTypePositive:Imageability	5.441	2.383	2.283	0.02461 *

---

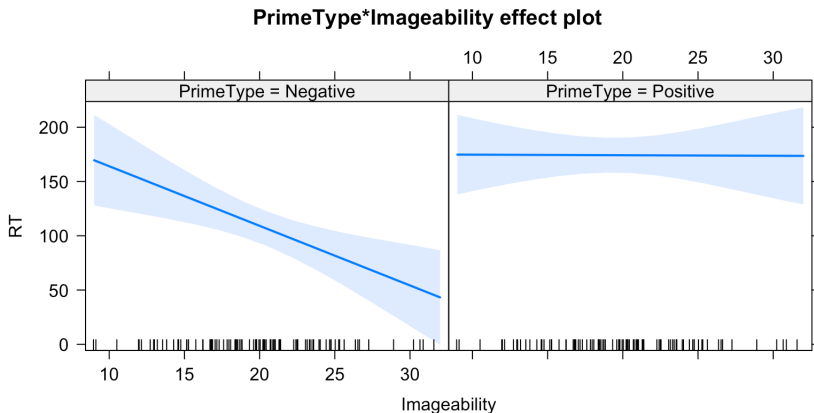
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.44 on 96 degrees of freedom

Multiple R-squared: 0.3175, Adjusted R-squared: 0.2961

F-statistic: 14.88 on 3 and 96 DF, p-value: 4.951e-08

# Interaction between a Factor and Continuous variable



Partial Effects of interaction of PrimeType and Imageability on RT

# Post hocs of mod5

Beware of interpretation of post-hocs in the case of interaction between a factor and a continuous variable.

```
> pairs(emmeans(mod5, ~PrimeType)) #misleading
```

NOTE: Results may be misleading due to involvement in interactions

contrast	estimate	SE	df	t.ratio	p.value
Negative - Positive	-63.4	11.4	96	-5.580	<.0001

```
> pairs(emmeans(mod5, ~PrimeType | Imageability))
```

Imageability = 19.7:

contrast	estimate	SE	df	t.ratio	p.value
Negative - Positive	-63.4	11.4	96	-5.580	<.0001

# Post hocs of mod5

It could be very informative to test the difference in the marginal means at several points of the covariate.

```
> Imag_values= quantile(dat5$Imageability, probs = c(0.05, 0.5, 0.90))
> pairs(emmeans(mod5, ~PrimeType | Imageability, at = list(Imageability = Imag_values)))
```

Imageability = 12.1:

contrast	estimate	SE	df	t.ratio	p.value
Negative - Positive	-22.3	21.4	96	-1.043	0.2995

Imageability = 19.8:

contrast	estimate	SE	df	t.ratio	p.value
Negative - Positive	-63.9	11.4	96	-5.628	<.0001

Imageability = 25.3:

contrast	estimate	SE	df	t.ratio	p.value
Negative - Positive	-94.2	17.6	96	-5.359	<.0001

## Note about terms in lm

It is easy to guess that the higher the number of variables of interest (fixed effects), the higher the number of coefficients to be estimated. If you take into account all the possible interactions, they grow exponentially.

Usually, in linear model. The choice of what coefficients and interactions is part of **Model Selection** (see the section [here](#))

# Results of Regressions

Checking regression results **not just about p-values.**



# Assumptions in Regressions

A regression analysis is meaningful only if some *assumptions* are met.

One crucial assumption (see also [these slides](#)) is the **independence** of observations. This means that each observation should be unrelated to the others.

## more on dependency

Suppose you have an experiment in which you want to investigate the effect of frequency and Length of stimuli on Reaction times in two different experimental conditions (and each Participant is administered with both conditions).

can you use linear regressions?

## more on dependency

Suppose you have an experiment in which you want to investigate the effect of frequency and Length of stimuli on Reaction times in two different experimental conditions (and each Participant is administered with both conditions).

can you use linear regressions?

NO

## from Linear regression to mixed effect models

In the example of the preceding slide there is a known dependency which is not properly taken into account in the model.

This leads to several problems: you cannot trust p-values, and you cannot trust the estimated parameters.

# from Linear regression to mixed effect models

To solve the problem you should specify some-way the dependencies. And you can do that with **Mixed Effect Models**

# Linear Mixed Models

The following is one of the simplest formula for a mixed model. In this simple case it is basically the same of a regression with the addition of a term.

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \mathbf{S}_1 + \epsilon \quad (3)$$

see also [here](#) for further formulas

# An example of Mixed effect model

In the next slides I will refer to an example, (with simulated data). You can download the data from [here](#). together with an accompanying R scripts.

The data refer to an hypothetical experiment:

**Priming effects on Metaphors and Literal sentences**

# An example of Mixed effect model (example1)

This experiment<sup>2</sup> was conducted on 20 participants. The experiment includes **two factors**:

- Stimulus type (two levels: Metaphor vs Literal)
- Priming type (two levels: Positive vs Negative)

We are also interested on the potential role of **two continuous variables** associated with the stimuli that could have some role

- Complexity
- Imageability

Each stimulus type (Metaphor and Literal) has **100 stimuli**, administered twice (Positive and Negative prime)

---

<sup>2</sup>This is the example1 in the accompanying code



# An example of Mixed effect model

Let's start from an hypothetical regression syntax including only predictors (i.e. *fixed effects*)

```
mod.lm = lm(RT ~ StimType*PrimeType + Imageability + Complexity,  
data=dat)  
# WRONG: dependencies are not considered.
```

# An example of Mixed effect model

Let's start from an hypothetical regression syntax including only predictors (i.e. *fixed effects*)

```
mod.lm = lm(RT ~ StimType*PrimeType + Imageability + Complexity,  
data=dat)  
# WRONG: dependencies are not considered.
```

A mixed model

```
mod.lmer = lmer(RT ~ StimType*PrimeType + Imageability + Complexity + (
```

# Mixed Effects Models

The fixed part of a Mixed model is basically the same as for a linear regression.

In addition, you specify the random effects, to specify dependencies in the data.

# Results from LM

Call:

```
lm(formula = RT ~ StimType * PrimeType + Imageability + Complexity, data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-605.35	-123.86	-1.14	121.07	565.27

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	493.9050	20.0185	24.672	< 2e-16	***
StimTypeMetaphor	-21.5665	5.8038	-3.716	0.000204	***
PrimeTypePositive	-62.5149	5.7500	-10.872	< 2e-16	***
Imageability	-143.4412	2.1403	-67.020	< 2e-16	***
Complexity	0.7444	0.2335	3.188	0.001437	**
StimTypeMetaphor:PrimeTypePositive	105.4571	8.1318	12.968	< 2e-16	***

---

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

Residual standard error: 181.8 on 7994 degrees of freedom

Multiple R-squared: 0.3857, Adjusted R-squared: 0.3853

F-statistic: 1004 on 5 and 7994 DF, p-value: < 2.2e-16

# Results from LMM

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: RT ~ StimType * PrimeType + Imageability + Complexity + (1 | Subj_ID) + (1 | Item_ID)
Data: dat
```

```
REML criterion at convergence: 93168.7
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-2.75299	-0.76198	-0.04284	0.73742	3.01997

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
Item_ID	(Intercept)	9569	97.82
Subj_ID	(Intercept)	18649	136.56
Residual		5949	77.13

```
Number of obs: 8000, groups: Item_ID, 200; Subj_ID, 20
```

```
Fixed effects:
```

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	493.9050	74.4849	199.2845	6.631	3.06e-10 ***
StimTypeMetaphor	-21.5665	14.3047	201.8251	-1.508	0.133
PrimeTypePositive	-62.5149	2.4390	7778.9995	-25.631	< 2e-16 ***
Imageability	-143.4412	7.3385	196.0014	-19.546	< 2e-16 ***
Complexity	0.7444	0.8006	196.0015	0.930	0.354
StimTypeMetaphor:PrimeTypePositive	105.4571	3.4493	7778.9995	30.574	< 2e-16 ***

```
---
```

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

```
(Correlation of Fixed Effects not shown)
```

# A first glance

At a first glance, there are some differences between `lm` and `lmer`, both in the output, and in the p-values.

This is related to the fact that some dependencies are taken into account.

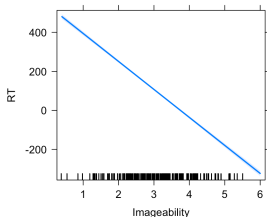
## Interim note

**IMPORTANT:** In all the examples about mixed models in the present slides I will also report p-values for coefficients of mixed models. `lmer` function does not return automatically p-values, because p-values are something tricky in the context of mixed effect models (see [here](#))

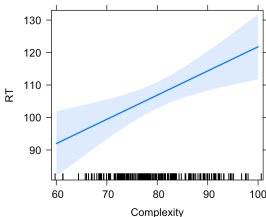
In all the example you can see p-values because I have always attached the `lmerTest` package, which provides p-values (and change the output of `lmer` function). See the [paper](#) on `lmerTest` for details.

# Linear model on the data

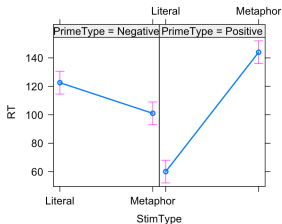
Imageability effect plot



Complexity effect plot



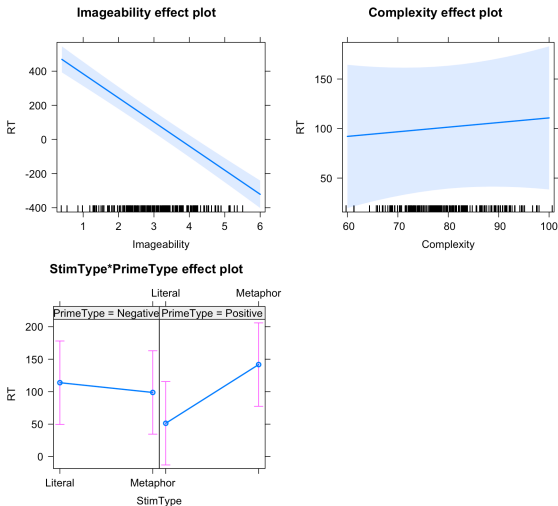
StimType\*PrimeType effect plot



partial effects of mod.lm



# Linear Mixed model on the data



partial effects of mod.lmer

# Linear model vs Linear Mixed Model

In the partial effects you can see how the error bands and bars are wider in the case of mixed model.

This is because in the linear model, each observation is treated basically as a separate Subject, and the model is fit like it is on thousands of separate (and independent) subjects (rather than on hundreds of Items administered to dozens of subjects).

As such, there is an overestimation of the confidence in the estimated parameters.

# Some terminology

Two crucial terms in Mixed Effect Models

- **Random Intercept**
- **Random Slope**

# Random Intercept

The **Random Intercept** represents an adjustment made to all observations that come from a specific level of a grouping variable. (e.g. the adjustment from all observation that come from the same participant)

In the mixed model estimation, it is associated to a standard deviation in the model.

The R syntax for specifying a random intercept is the following.

```
(1 | Subj_ID)
```

# Random Intercept in Psycholinguistic experiments

Typically in Psycholinguistic experiments there are two main random effects to be taken into account: participants, and stimuli (Clark, 1973).

This is easily taken into account in mixed effect models, specifying two crossed (i.e., independent) random effects:

$(1 \mid \text{Subj\_ID}) + (1 \mid \text{Item\_ID})$ .

# Random Intercept in Psycholinguistic experiments

In coarse words, adding Random intercepts in the models with the syntax:

$$(1|\text{Subj\_ID}) + (1|\text{Item\_ID})$$

Indicates that the model should taken into account that there could be some dependencies in all observations that come from the same Subject and the same Item. In particular, some subjects could be in general faster then others or slower than others, or that responses to some Items could be faster the those to others.

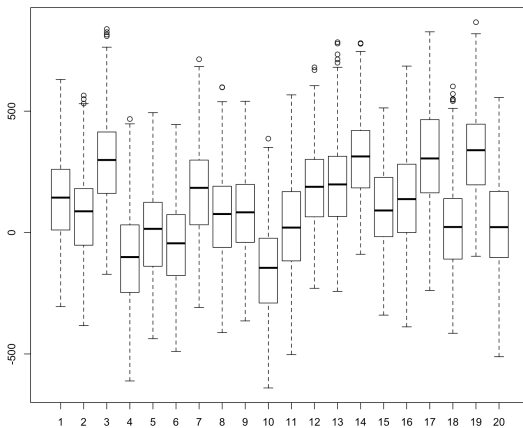
This should be taken into account in assessing significance and in determining results.

# A further note on Random Intercept

There is another way to interpret Random Intercepts and is about generalizability of results.

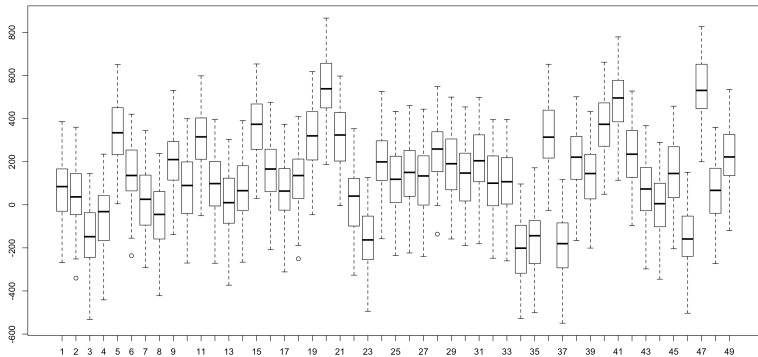
- $(1 \mid \text{Subj\_ID})$  : will the results generalize to new Subjects?
- $(1 \mid \text{Item\_ID})$ : will the results generalize to new Items?

# Variability in Subjects





# Variability in Items



First 50 Items among 200 are shown

## Checking random effects

The adjustments of random effects can be inspected by using this command:

```
> ranef(mod.lmer1)
$Item_ID
      (Intercept)
1      18.748191
2     -167.967693
3     -172.824119
4      -3.032910
5      70.462635
```

The result indicates the adjustments to be made. For example, responses to Item 3 are particularly fast.

# Interim Note

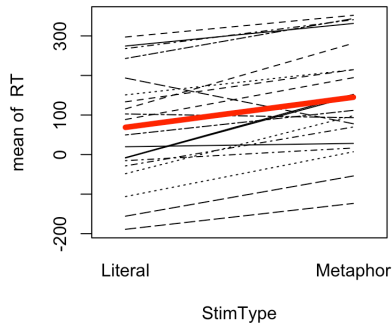
When you use a simple ANOVA, actually the within-Subject dependency is normally taken into account. To avoid problems, it is common to average all Items. This solution avoids dependency, but also prevents the possibility to generalize to other items.

In the past other analysis (like  $F_1$  and  $F_2$ ,  $F^1$  or  $MinF^1$ , were used, see Baayen, 2008).

# Random Slope

The **Random Slope** is an adjustment in the model that try to take into account the fact that a fixed effect may be different across a grouping variable. For example, it may be the case that an experimental manipulation (e.g., Metaphor vs Literal) has a different effect in all subjects, and that some subjects may show a different effect.

# Random Slope



Black lines, are by subject slopes.

Red line is the group slope (averaging all subjects)

# Random Slope

In mixed models random slopes are described (and modeled) by a standard deviation and correlation.

The standard R syntax for specifying a random slope is the following.

```
(1 + StimType | Subj_ID)
```

Note that the syntax above, specify both a random slope, and a random intercept.

# Random Slope

The **Random Slope** specification is normally associated to more than one parameter. One is the standard deviation of the slope, the other is the correlation between this slope with the Intercept.

# Model with random slope

## A model with a Random Slope

```
lmer(RT ~ StimType * PrimeType + Imageability + Complexity +  
      (1 + StimType | Subj_ID) + (1 | Item_ID), data = dat)
```

This model include the possibility that the slope associated to StimType (i.e., the effect of StimType) may be different across Subjects. We can include the information in the model as additional parameters.



# Model with random slope

Linear mixed model fit by REML. t-tests use Satterthwaite's method [`'lmerModLmerTest'`]

Formula: `RT ~ StimType * PrimeType + Imageability + Complexity + (1 +`

`StimType | Subj_ID) + (1 | Item_ID)`

Data: `dat`

REML criterion at (see for example [here](https://github.com/lme4/lme4/issues/512)): 917

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.60713	-0.74448	0.00371	0.72035	2.85084

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
Item_ID	(Intercept)	9594	97.95	
Subj_ID	(Intercept)	20365	142.71	
	StimTypeMetaphor	4292	65.51	-0.30
Residual		4900	70.00	

Number of obs: 8000, groups: Item\_ID, 200; Subj\_ID, 20

Fixed effects:

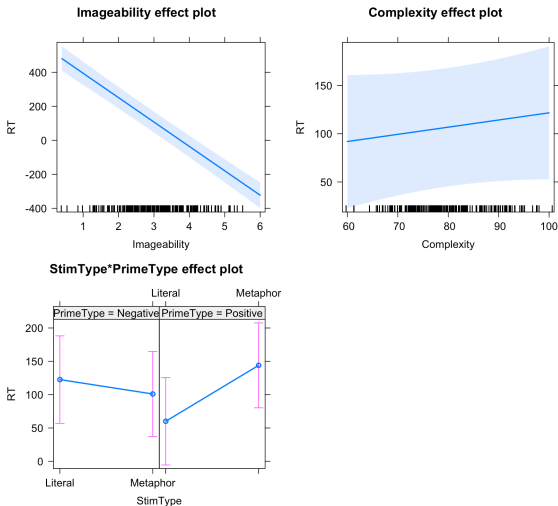
	Estimate	Std. Error	df	t value	Pr(> t )	
(Intercept)	493.9050	75.0524	194.2459	6.581	4.25e-10	***
StimTypeMetaphor	-21.5665	20.4617	65.2559	-1.054	0.296	
PrimeTypePositive	-62.5149	2.2136	7759.9899	-28.241	< 2e-16	***
Imageability	-143.4412	7.3380	196.0445	-19.548	< 2e-16	***
Complexity	0.7444	0.8005	196.0442	0.930	0.354	
StimTypeMetaphor:PrimeTypePositive	105.4571	3.1305	7759.9899	33.687	< 2e-16	***

# Model with random slope

The random slope is associated with a standard deviation and a correlation. This last parameter is the correlation between the slope of the Subject and the Intercept.

This is taken into account because the slope and Intercept adjustments may be correlated, and this could be included in the model (to further consider dependencies in the data).

# Linear mixed model on the data (Random Slope)



partial effects of mod.lmer2

# About Random Slope

Did you notice something about the way I modeled the Random Slope?

# About Random Slope

Did you notice something about the way I modeled the Random Slope?

What about the `PrimeType`?

# About Random Slope

I could fit a more complex model

# About Random Slope

I could fit a more complex model

```
lmer(RT ~ StimType * PrimeType + Imageability + Complexity +  
      (1 + StimType*PrimeType | Subj_ID) + (1 | Item_ID), data = dat)
```

# About Random Slope

I can also model the possibility that the effect of imageability changes across participants!



# About Random Slope

I can also model the possibility that the effect of imageability changes across participants!

```
lmer(RT ~ StimType * PrimeType + Imageability + Complexity +  
      (1 + StimType*PrimeType | Subj_ID) + ( 1 + Imageability | Subject_ID)  
      (1 | Item_ID) , data = dat)
```

# About Random Slope

And what about an even more complex random structure?

# About Random Slope

And what about an even more complex random structure?

```
lmer(RT ~ StimType * PrimeType + Imageability + Complexity +  
      (1 + StimType*PrimeType*Imageability | Subj_ID) +  
      (1 | Item_ID) , data = dat)
```

# About Random Slope

And what about an even more complex random structure?

```
lmer(RT ~ StimType * PrimeType + Imageability + Complexity +  
      (1 + StimType*PrimeType*Imageability | Subj_ID) +  
      (1 | Item_ID) , data = dat)
```

Although (in theory) it is possible to fit this kind of models, they typically don't work, leading to a problem in *convergence*

## by-Item or by-subject random slope

From Bayeen (2008) book, p. 316

*“in general, predictors tied to subjects (age, sex, handedness, education level) may require by-item random slopes, and predictors related to items (frequency, length, number of neighbours, etc.), may require by-subject random slopes.”*

# Convergence issues

Fitting a mixed model is a process that can show issues. The more common problems are issue in **convergence**

Broadly speaking, sometimes the algorithms beyond mixed models (or `lmer`) are not able to find a solution.

The large majority of the time, problems in convergence are related to random structure that are too much complicated.

# Convergence issues

Sometimes the model converge (so no Error), but a Warning about convergence appears.

In such cases, something did not go smooth in the fit, and you may want to check your model better. However this does not mean that the fit is wrong. Convergence issues is an advanced topic, that will be treated in following slides. My main suggestion is to follow the excellent explanation here [here](#)) by Michael Clark.

# Convergence issues

A practical approach to convergence issues is the following:

- ① don't ignore the convergence issue.
- ② make all the checks for a good modeling
  - use a reasonable complex model for the number of observations.
  - centered predictor.
  - adequate random structure.
- ③ use `allFit` function to see if changing the optimizer lead to changes in fit. If results are similar you could reasonably trust your results.

see [these slides](#) for more information about convergence and optimizers.



## Problems in random structure

There are some indicators of issues in the random structure:

- Variance component of random effects are close to zero.
- Correlation parameters are close to  $\pm 1$

## Other random structures

To solve issues with random structure, one of the most common strategy is to *center* predictors. This is particular common for number of trial, when used as covariate and then random slope.

```
dat$ntrial_c = scale(dat$ntrial)  
ntrial_c + (1 + ntrial_c | Subj_ID)
```

## Other random structures

An additional strategy is to remove the correlation between Intercept and slope. The syntax is the following.

```
(1 | Subj_ID) + (0 + StimType | Subj_ID)  
# uncorrelated random-intercepts and random slopes
```

## Other possibilities

There are other possibilities of random structure. See for example [here](#) or [here](#)

# How to choose the Random structure?

So how to choose the random structure?

# How to choose the Random structure?

So how to choose the random structure?

- 1 Keep it Maximal”(Barr et al., 2013)
- 2 “Parsimonious Mixed Models ”(Bates et al., 2015)

These are the two different main perspectives in choosing the random structure with pros and cons.

# Maximal Random structure

The maximal structure is univocal, and according to the simulations by Barr et al. (2013), it is also the more robust choice. The downside is that as the design becomes more and more complicated, the random structure will become *extremely* complicated, and it is very likely that convergence issues will occur.

The maximal random structure is basically the structure that includes, for every intercept, all the possible interactions of fixed effects. In an example with StimType, and PrimeType as fixed effects, and Sub\_ID and Item\_ID as random variables the model would be:

```
RT ~ StimType*PrimeType +  
(1+StimType*PrimeType | Subj_ID) + (1+StimType*PrimeType | Item_ID)
```

## Parsimonious Mixed Models

A recommendation on parsimonious random structure can be interpreted as a general comment to opt for simpler and theoretically motivated random structures.

In a couple of articles Bates and colleagues describe a specific approach of fitting mixed models, by checking the redundancy in the variance-covariance component that is created in mixed models, to empirically verify for a too complex random structure.

The two articles are [this](#) and [this](#), and make use of the `rePCA` function in `lme4`. See also [these slides](#) in advanced topic section.



## by-Item or by-subject random slope

Before developing a structured approach, an alternative is to use reasonable random slope models.

. From Bayeen (2008) book, p. 316

*“in general, predictors tied to subjects (age, sex, handedness, education level) may require by-item random slopes, and predictors related to items (frequency, length, number of neighbours, etc.), may require by-subject random slopes.”*

## Effect of Random structure on fixed effects

Does the random structure affects the fixed effects estimates?

YES

However (this is an empirical consideration) typically in the random structure affect mostly the standard error of the estimates and, in turn, the significance fixed effects.

# Parsimonious Random structure

The parsimonious random structure is a reasonable random structure guided by the data, and that try to avoid overparametrizing the models.

It is a bit more difficult to determines a univocal parsimonious structure but typically it includes all relevant random intercepts, and also (if necessary) a random slope for some crucial variables guided from theory (for example, the main experimental variable) or from checks (see next slides).

```
# a potential parsimonious mixed model  
RT ~ StimType*PrimeType +  
(1 + StimType | Subj_ID) + (1 | Item_ID)
```

# Interim Summary

So far we have discussed the main following points:

- Linear Mixed models are an extension of Linear regression
- Linear Mixed models have also this "random part" which may be a bit difficult to define.

## Next steps

Have we discussed all topics related to linear regressions, relevant for Mixed models?

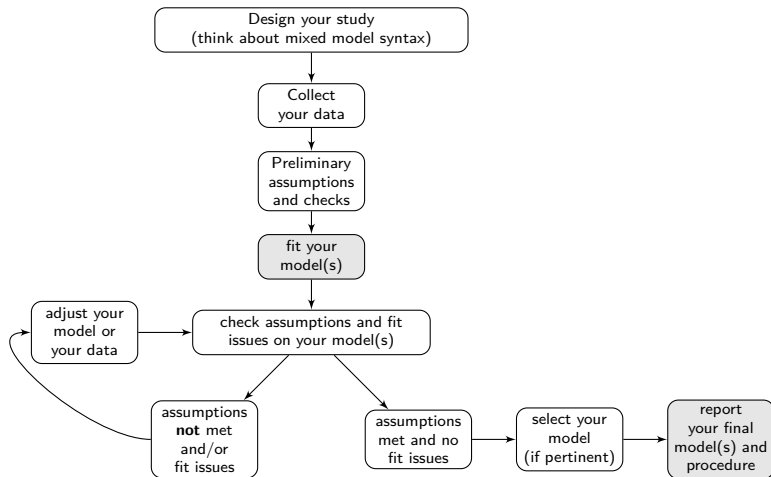
# Next steps

Have we discussed all topics related to linear regressions, relevant for Mixed models?

Absolutely not. There are at least two major points remaining:

- checking the assumptions (and solving problem when assumptions are not met)
- model selection

# A potential pipeline for mixed model analysis steps



# Checks assumptions and fit issues

To my knowledge (updated at , there are no official recommendation on the sequece of checks for fit issues. Guided by a practical considerations I would perform checks in this order:

- 1 check if random structure has issues (zero variance or  $\pm 1$  correlation of random slopes)
- 2 check convergence issues (if necessary with `allFit` function)
- 3 check model assumption (e.g. with `performance` package)

I would consider a model only if the three checks above are good (and this is reflected in my suggested pipeline)



## Limits of a linear pipeline mixed models

Sketching a linear pipeline for mixed models is hard.

In many situations one need to go back and forward, until finding a convincing model.

Here I sketched a potential pipeline that is more a suggestion, but other pipelines can be reasonable (or better).

# A potential pipeline for mixed model analysis steps

Check also [the github repo associated with these slides](#) for a practical example of a pipeline.

# Assumptions in Regressions

The assumptions to be checked for mixed effect models, are mostly the same to be checked for regression.

Often, checking assumptions implies performing some **diagnostics** on the fit.

Diagnostics may be numbers or (more often) plots.

# Why is important to check assumptions?

If not many things about the models could be not reliable:

- estimated parameters
- p-values
- confidence intervals
- ...

# Assumptions in Regressions

- 1 Linear relationship between predictors and dependent variable.
- 2 Homoscedasticity of the residuals.
- 3 Normality of the residuals.
- 4 No problems in multicollinearity of predictors.
- 5 Independence of residuals.
- 6 Independence of observations.

see also [here](#)

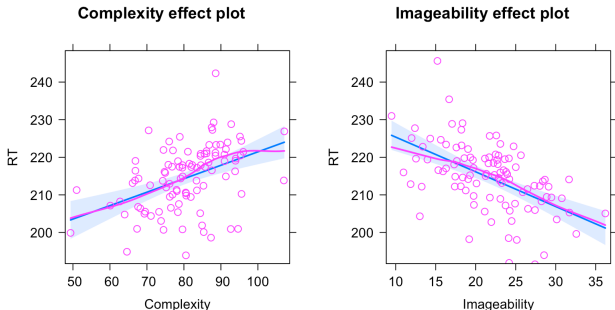
note that this list may be different across different sources.

# Assumptions in Regressions (1)

1 Linear relationship between predictors and dependent variable.

→ plot partial residuals

```
plot(allEffects(mod2, partial.residuals=T))
```



Partial residuals of the two effects

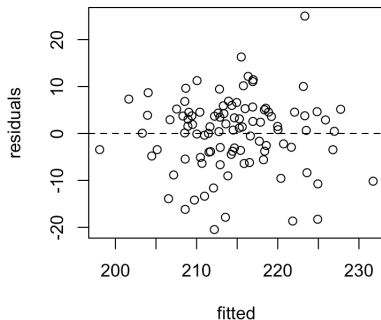
# Solving issues (1)

If partial residuals suggest non linearity is present, you may add a transformation to the predictor to account for the non-linearity (see Baayen, 2008)

## Assumptions in Regressions (2)

- 2 Homoscedasticity of the residuals → plot residuals against fitted.

```
plot(fitted(mod2), resid(mod2))
```





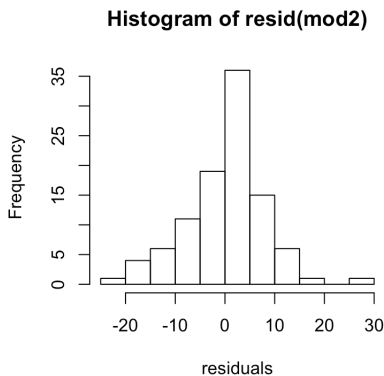
## Solving issues (2)

Try to check if you are missing a relevant predictor or a random effect that can increase quality of fit (see Model Selection section)

## Assumptions in Regressions (3)

3 Normality of the residuals → plot residuals against fitted.

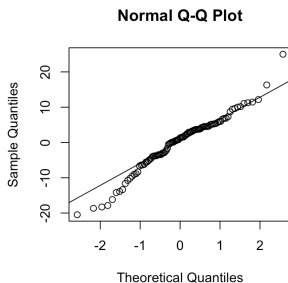
```
hist(resid(mod2))
```



Normality of residuals can be checked also looking at quantiles

# Assumptions in Regressions (3)

Another common way to inspect normality of the residuals is to use a qqplot.



## Solving issues (3)

Try to check if you are missing a predictor or a random effect that can increase quality of fit.

Another common strategy is to refit the model excluding the outliers, to see if they influenced the results.

It may be the case that your residuals suggest the need of a different family to be specified. Indeed in mixed models, you can specify also different expected residual distribution. See [here](#)

Although suboptimal, a common strategy to avoid non-normal distribution is to apply a transformation to dependent variable *before* fitting the model. For example log transform the data for skewed reaction times.

# Assumptions in Regressions (4)

## 4 No problems with multicollinearity.

Multicollinearity is an index of the overall correlation between predictors that may affect parameter estimation. Remember, in regression each term is estimated to be uncorrelated from the others. What if they are highly correlated? See [here](#)

## Assumptions in Regressions (4)

4 No problems with multicollinearity.

→ check if Variance Inflation Factor  $< 10$

```
vif(mod3) # note: I need at least two  
# predictors to check collinearity
```

```
PrimeType Imageability  
1.025689      1.025689
```

Note that in some cases GVIF is computed. See [here](#) or [here](#) for details or `help(vif)`

## Solving issues (4)

You have several choices:

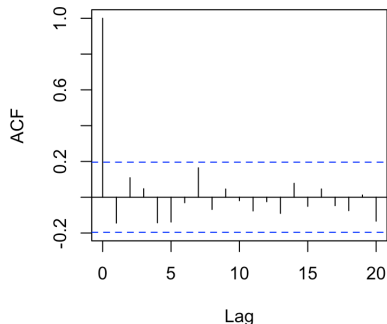
- remove one of the collinear variables.
- run a Principal Component Analysis (PCA) to solve the issue (see Baayen, 2008, p. 198).
- you may residualize on variable on the other. (see [here](#))
- you may use a composite score (e.g., averaging or summing scores of two variables, if meaningful).

For further considerations see also this blog post [here](#)

# Assumptions in Regressions (5)

- 5 Independence of residuals  $\rightarrow$  use some measures or plot to check autocorrelation

**Series resid(mod2)**





## Solving issues (5)

Autocorrelation are typically related to trial order.

- you can add a covariate with Trial Number and a random slope by participant  $n\text{Trial} + (1 + n\text{Trial} \mid \text{Subj\_ID})$
- you can add, as covariate, the value for the preceding trial. (see Baayen & Milin, 2010)
- you may want to use extension able to deal with autocorrelations as Generalized Additive Mixed Models (GAM) see this wonderful tutorial by Jacolien Van Rij.

# Assumptions in Regressions (6)

## 6 Independence of observations

→ this is theoretically defined (e.s. observations come from the same participant)

## Solving issues (6)

Use a mixed-effect model with appropriate random structure.

# Checking Assumptions

The R package performance allow a series of checks (related to the assumptions to be checked in models)

Check the package content [here](#)

# Model Selection

When using mixed effect models (or in general when fitting models) there are many opportunities to perform the analysis.

This is different for example, from ANOVA, when given an experimental design, there is typically just one possible analysis.

We have seen a hint of this when seeing that several random structures can be used to analyze the same data. However, the flexibility is not only for random part, but also for the fixed part.

# Model selection strategies

There are three main *model selection* strategies.

- 1 Fit just one model (theoretically driven), inspect the results.
- 2 compare few theoretical driven models.
- 3 stepwise selection of the best fitting model

The choice depends on the hypotheses or experimental aims<sup>3</sup>

---

<sup>3</sup>there are also other possible methods, for example see “Parsimonious Mixed models” paper and associated R vignettes. [link](#)

# Model selection (1)

## Fit just one model

This is the more conservative and robust choice. Given some hypotheses, you fit a model and check the significance of the relevant terms, using Anova p-values to test for significance.

# Model selection (2)

## Compare few theoretical driven models

With this choice you can compare few theoretically driven models to test which one fit the data better.



## Model selection (2) - LRT

To do this you, in the case of *nested*<sup>4</sup> models, it is normally used the Likelihood ratio Test (LRT). To perform LRT:

```
anova(mod1, mod2)
```

If the test is significant, the more complex model is selected, otherwise, the simpler model is selected.

---

<sup>4</sup>a smaller model is *nested* to a larger one if it contains all the terms and variables of the larger models

## Interim clarification: nested vs non-nested

Consider the three models below.

```
mod.lmer8a = lmer(RT ~ StimType*PrimeType +  
                  (1| Subj_ID) + (1 | Item_ID),  
                  data=dat)
```

```
mod.lmer8b = lmer(RT ~ StimType*PrimeType + Imageability  
                  (1| Subj_ID) + (1 | Item_ID),  
                  data=dat)
```

```
mod.lmer8d = lmer(RT ~ StimType*PrimeType + Concreteness + Complexity +  
                  (1| Subj_ID) + (1 | Item_ID),  
                  data=dat)
```

mod.lmer8a is nested on mod.lmer8b. On the other side the model mod.lmer8d has no nesting relationship with the others.

## Model selection (2) - LRT

The LRT can be used also to select the more appropriate random structure, with the same principle of before (if not significant, select the simplest random structure).

## Model selection (2) - LRT

Some notes:

- 1 when doing an LRT, a warning is returned. Data are fitted with ML. By default, mixed model fitting occurs using REML method<sup>5</sup>. However LRT cannot be used if models are fit with REML. For this reason, when using anova to compare models, the models are refit with the ML.
- 2 it makes no sense to compare with LRT two models in which both Random and fixed parts are different.

---

<sup>5</sup>for difference between ML and REML see [here](#) or [here](#)

## Model selection (2) - AIC, BIC

For non-nested models you can use some measure as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC).

For these measures, the lower the measure the better the fit. Given the same data, they do not need nested models.

## Model selection (2) - AIC, BIC

A common question is: what should I choose between AIC and BIC?

This is a somehow "ill-posed" question: there is not a better or worse, as they are different methods of model comparison, but answering to different questions. (see [here](#))

BIC penalizes more the number of free parameters.

## Model selection (3) - stepwise selection

With the stepwise selection, a series of model are fit on the same data, and predictors or random variables, are included (or excluded) in a progressive fashion, up to reach the best fit the model.

These are not recommended (Harrel, 2015), as many models are fit, typically without taking into account properly an inflated risk of Type 1 error.

## Model selection (3) - stepwise selection

Model selection can be made by hand, starting from a NULL model and adding fixed (or random) variables, and then checking whether adding variables lead to better models (using LRT or AIC/BIC).

An alternative is to start from a full model and delete (one at the time) the variables that has the worse performance (for example the highest p-value).



## Model selection (3) - stepwise selection

You can do model selection automatically using the `step` function, loaded with the `lmerTest` package.

In this cases you can select the direction of selection (forward or backward).

# Where to go from here?

To understand mixed models, the best way is to use them on your data.

You can start from the examples in the github repository associated with these slides (see [here](#))

# Advanced Topics

There are many developments of mixed effect models. Here some of the more common examples:

- Generalized Linear Mixed Effect Models (GLMM).
- Non-linearities.
- Generalized Additive Mixed Models (GAMM).
- More on AIC
- Fit issues: singular fit and convergence warnings
- Miscellaneous topics

# GLMM

An assumption of Linear Mixed Models (LMM) is that the expected conditional distribution of dependent variable is gaussian.

GLMM allow to relax this assumption and use mixed models with dependent variables with different distributions (e.g. Binomial, Poisson, etc.) The more common model used for GLMM is with binomial distribution. In these models the answer (at single trial level) can be either Correct (typically codified with 1 or wrong, codified with 0).

# GLMM

In GLMM a common issue is that they are even more prone to have convergence issues.

For this reason it is better to avoid complicated design with GLMM.

Using GLMM in R typically requires minimal changes, that is to specify the family of the distribution of the data.

# GLMM

the term **family** refers to the expected conditional distribution.

the term **link function** indicates how predictors and response are linked.

see [here](#) for further details.

# GLMM

the family and link can be selected according to three main strategies

- ① Knowledge of response distribution
- ② Theoretical considerations
- ③ empirical fit to the data.

see [here](#) for details.

# GLMM and beyond

Sometimes to overcome limitation of GLMM (as implemented in `lme4` package) it is necessary to use other packages to fit other models (e.g. for zero inflated or negative binomial distributions you may use `glmmTMB` or `GLMMadaptive`).

Keep in mind that these package (albeit wonderful) in my experience can be less integrated with other existing packages or can cause more compatibility issues (e.g. as for 2024 May, I'm not able to use `glmmTMB` package neither online nor in my laptop).



# GLMM

For GLMM most of the things discussed for LMM are valid.

An important difference is on the inspection of the residuals. To check appropriateness of residuals in the case of GLMER distribution check the Dharma package ([see here](#))

# Diagnostic of GLMM

Diagnostic of GLMM are a bit more difficult to tackle, due to the fact that expected residual variability changes depending on expected values of dependent variable.

In general the main checks in GLMM is to ascertain that there are no strange patterns in residuals. This is not straightforward as for linear model with gaussian distributions.

In general, **residuals should be uniformly distributed across all predictors value or fitted values**

# DHARMa

DHARMa package offers a series of checks to identify *potential* misfits in the model, regardless of expected distribution of residuals.

It provides with a series of plots, significance tests to uncover patterns in the residuals that can help in deciding how to move.

DHARMa has excellent support documentation so check the reference website ([here](#))

# A note on checking assumptions

One should never forget the following principles:

- good residuals do not necessarily imply a good model.
- slightly bad residuals do not necessarily imply a bad model.
- residual checks is important, but not the core of the analysis.
- a good model is not necessarily the *real* model or a *replicable* model

# A short pipeline to work with *DHARMa*

- 1 simulate residuals from your fitted model
- 2 check with the various plots and tests, how *observed* residuals are as compared to *simulated* residuals

## A short pipeline to work with *DHARMa*

- 1 generate simulated residuals.
- 2 plot residuals to search for systematic (and unwanted) patterns.
- 3 run some tests for some major issues (Overdispersion, Zeroinflation, but may depend on the data).
- 4 plot residuals against each predictor to further search patterns.

# A short pipeline to work with *DHARMa* (with more details).

```
simulationOutput <- simulateResiduals(fittedModel = mod1c, plot = F)

# check overdispersion
testOverdispersion(simulationOutput)
testZeroinflation(simulationOutput)

# check residuals against predictors and random variables
plotResiduals(simulationOutput, dat_bin$TOM)
plotResiduals(simulationOutput, dat_bin$Interpretability)
...
```

## The rationale of *DHARMa* in short (1)

To overcome the fact that it is difficult to model mathematically the residuals, *DHARMa* uses a simulation approach.

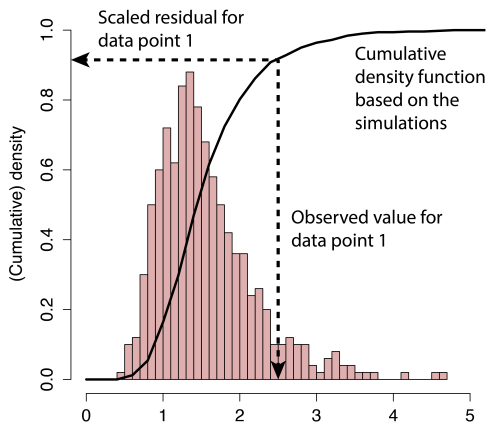
- 1 It simulates new data from fitted model parameters many times
- 2 For each observation, calculate the *empirical cumulative density function* for the simulated observations, which indicates the possible values (and their probability) at the predictor combination of the observed value, assuming the fitted model is correct.
- 3 define residuals for observed data from this simulations (see next slide)

See [this link](#) for a more detailed explanation.



## The rationale of *DHARMa* in short (2)

Residuals have values from 0 to 1 and represents the empirical density function at the value of the observed data.



## Interpreting DHARMA residuals

Residuals have values from 0 to 1 and represents the empirical density function at the value of the observed data.

In other words the value respond to the question: *what is the probability of observing this value in the simulation? (which assume correct fit)*

- a value of 0 indicates that all of simulated data are higher than observed data.
- a value of 0.5 indicates that half simulated data are higher than observed data.
- a value of 1 indicates that no simulated data are higher than observed data.

## Interpreting DHARMA residuals (*a practical example*)

Suppose that in your observed data you have (among many others) these values: 10, 1, 3.

After you fit a model, thanks to DHARMA algorithms you can simulate some further data (we call them *simdata*).

Let's see this hypothetical scenario.

- 10 is higher than 100% *simdata*  $\rightarrow$  residual is 1.0.
- 1 is lower than 100% *simdata*  $\rightarrow$  residual is 0.
- 3 is lower than 30% of *simdata*  $\rightarrow$  is 0.3.

If the model is ok we expect that residuals are evenly distributed: some high, some low, some intermediate values, with no clear systematic pattern.

## Interpreting DHARMA residuals

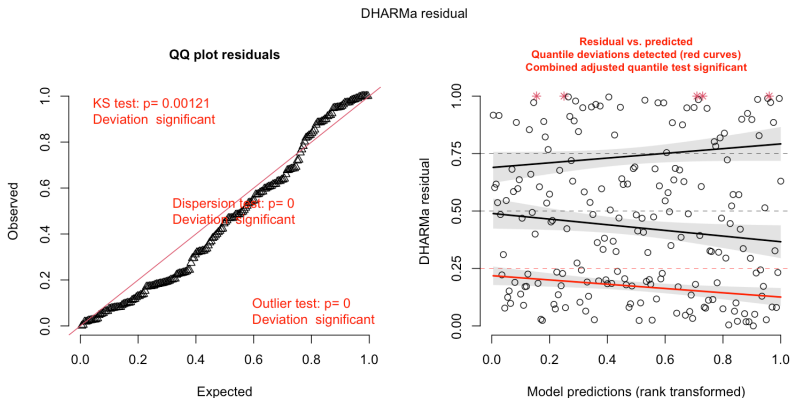
Once this non-parametric residuals are determined they can be plotted in several ways.

If things are working properly we should not see any pattern and they should be distributed evenly, across predictors and dependent variable value.

Any misfit or issue could lead to uneven distribution of residuals.

# Main DHARMa plot

The main DHARMa plot (`plot`) contains already lots of information. it is the combination of `plotQQunif` and `plotResiduals`



## Main issues you may encounter: Overdispersion

The main issue you can spot with DHARMA is probably  
**Overdispersion**

Significant overdispersion indicates that the variability of your data is higher than expected.

Overdispersed lead mostly to inflated type I error (you may find too easily significant effects).

## Main issues you may encounter: Zero Inflation

Zero inflation is a special case of overdispersion and indicates that there are too many zero in your data (as compared to expectations).

It can lead to wrong parameter estimates and inflated type I error

## What if you spot issues with your GLMM model? (1)

There is no a perfect recipe, but here some considerations

- overdispersion is often consequence of missing predictor or random part.
- different problems can lead to similar effects on diagnostic plots, so there is no a one-to-one mapping: problem  $\leftrightarrow$  symptom.



## What if you spot issues with your GLMM model? (2)

Here some hints:

### **overdispersion detected**

- → Try to include other predictors or improve the random part.
- not working? → use models that account for overdispersion.  
(e.g. negative binomial distribution).
- *or* try some data transformation or a different link function  
(check empirically if it improves model fit).

# What if you spot issues with your GLMM model? (3)

## Zero-inflation detected

- → Try to include other predictors or improve the random part.
- not working? use models that account for zero inflation (zi models)
- *or* try some data transformation or a different link function (check empirically if it improves model fit).

## What if you spot issues with your GLMM model? (4)

Note that these are just generic suggestions, but other issues can lead to overdispersion (e.g. a missing non-linear predictor), so they should be taken as indications and not as rules.

# Ordinal Regressions

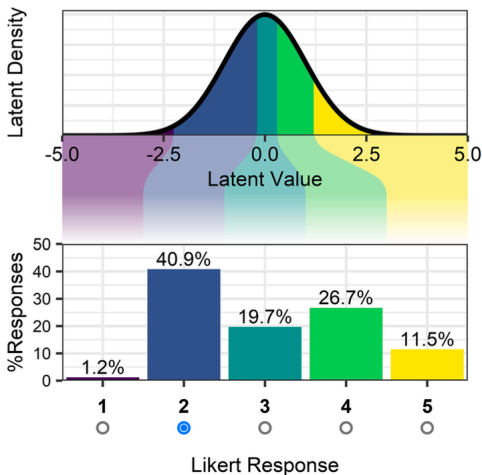
Whenever the dependent variable is not on an interval scale, but rather on an *ordinal* scale, it would be more appropriate to use **ordinal regressions**

For example a likert scale is more properly described as an ordinal variable. A score of 4 (fourth response) is not twice a score of 2 (second response).

# Latent variable

Technically in ordinal regressions is assumed that there is a latent variable (continuous) that is associated with the observed ordinal responses. Part of the aim of ordinal regression is to estimate the *thresholds* of this latent variable.

# Thresholds in ordinal regressions



from Taylor et al., (2022)

## Mixed model Ordinal regression in practice

a popular package for ordinal regression is `ordinal` The syntax is quite similar to that of `lme4`

```
mod_ord = clmm(rating~StimType*Group+ Interpretability  
+TOM+ Age+(1|Subj_ID) + (1|Item_ID),  
data=dat_ord, link="logit")
```

# Ordinal regressions

Cumulative Link Mixed Model fitted with the Laplace approximation

```
formula: rating ~ StimType * Group + Interpretability +  
TOM + Age + (1 | Subj_ID) + (1 | Item_ID)  
data:    dat_ord
```

```
link   threshold nobs logLik   AIC      niter      max.grad cond.H  
probit flexible  300  -430.10 888.21 1449(3008) 2.42e-02 8.0e+05
```

Random effects:

```
Groups Name          Variance Std.Dev.  
Subj_ID (Intercept) 0.03715  0.1927  
Item_ID (Intercept) 0.01648  0.1284
```

Number of groups: Subj\_ID 30, Item\_ID 10

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
StimTypeImplicature	1.365587	0.248562	5.494	3.93e-08 ***
GroupControls	-0.358672	0.237630	-1.509	0.1312
GroupSchizophrenics	-0.462293	0.236987	-1.951	0.0511 .
Interpretability	-0.093176	0.018909	-4.928	8.32e-07 ***
TOM	0.052371	0.039696	1.319	0.1871
Age	0.003760	0.008408	0.447	0.6547
StimTypeImplicature:GroupControls	-0.225960	0.304349	-0.742	0.4578
StimTypeImplicature:GroupSchizophrenics	0.108320	0.302575	0.358	0.7203

---

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



# Ordinal regressions

Ordinal regressions add in the output the thresholds' coefficients.

Threshold coefficients:

	Estimate	Std. Error	z value
1 2	-1.1761	0.5917	-1.988
2 3	-0.4043	0.5883	-0.687
3 4	0.4392	0.5890	0.746
4 5	1.0685	0.5915	1.807

# Ordinal regressions - Assumptions

Some assumptions of regressions are shared with ordinal regressions (e.g. no multicollinearity).

An important additional assumption is the *Proportional odds assumption*. It states that the estimated effect of coefficients hold for all values of the dependent variable.

## Ordinal regressions - Proportional odds assumption (1)

Note, that a positive coefficient in ordinal regression indicates that the higher the value, the higher the probability of being in an higher level of the dependend variable.

The *proportional odds assumption* states that a one-unit increase in an IV has the same effect on the probabily of a response being in a higher level regardless of the level.

## Ordinal regressions - Proportional odds assumption (2)

Many packages do not offer formal tests for violation of this assumption.

Also, existing tests appear to be anticonservative, and better solutions are preferred (see a discussion [here](#))

## Ordinal regressions - other diagnostics

Unfortunately there are no other diagnostics available for ordinal regressions fit with `ordinal` package (e.g. DHARMa does not work, cause `clmm` model does not have a residual function associated).

# Non-linearities

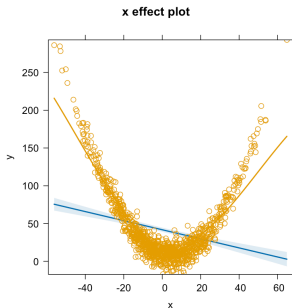
So far we discussed about linear relationships and we also have seen that one assumption is about linear relationships between predictors and dependent variable (see [these slides](#)).

However it is possible to also include non-linear relationships. See book by Baayen (2008), freely available online [Download](#).

# Detecting Non-linearities

The best way to detect the need for non-linear effect is *partial residuals plots*, that you can easily obtain with the `effect` package. using the code

```
plot(allEffects(mod, partial.residuals=TRUE))
```



# Detecting Non-linearities

Note that in a case like the one of the previous slides, there is an issue in the fulfillment of assumptions. The relationship between predictor and dependent variable is not adequately explained as a linear effect.

In this case (no matter the other metrics, e.g. AIC), it would be *fundamental* to include a non-linear term, otherwise the conclusion drawn from the model would be incorrect.

Of course this is an exaggerated example, and typically with real data effects are much more nuanced.



## Fitting non linearities (top down)

There are many ways to fit non-linearities in a top down fashion.

In this case you specify the type of transformation to be applied to the predictor. You can then check whether adding the non-linear terms improve the fit, both in term of model comparisons (e.g. AIC, BIC, etc.) and in terms of better diagnostics and or better partial residuals.

a typical approach is to fit polynomials with increasing order, and stop whenever they are not useful in improving model fit/diagnostics.

## Fitting non linearities with `I()` function

You can fit non linearity specifying higher order polynomials with the `I()` function

```
... + predictor + I(predictor, 2) + I(predictor, 3) +  
...
```

## Fitting non linearities with `poly()` function

You can fit non linearity specifying higher order polynomials with the `poly()` function

```
... + poly(predictor, 3) + ...
```

## Fitting non linearities with `rcs()` function

You can fit non linearity specifying higher order polynomials with the `rcs()` function, in the `rms` package. `rsc` stands for *restricted cubic splines* which represent a way to fit non linearities that tend to reduce curvatures at the edges.

```
... + rcs(predictor) + ...
```

# GAMM

Generalized Additive Mixed Models (GAMM) represent a further development. In GAMM allows to flexibly model non-linearities in a bottom-up fashion. Basically the model itself try to understand if a non-linear function is better to explain the data.

GAMM are particularly powerful for handling interactions between continuous variables, which are quite an issue in linear models.

GAMM allows also flexible ways to model random effects, in particular, introducing the *random smooth* terms.

# Tutorials and explanations on GAMM

there are many excellent introductory tutorials on GAMM.

- this [preprint](#) By Baayen and Linke explain in plain terms the math beyond GAMM (see Figure 4, to discover how GAMM magic works).
- This [tutorial](#) by Jacolien Van Rij, provide a nice introduction to the topic.
- [here](#) an excellent tutorial on gamm visualization

# GAMM

Non linearity can be fit also in other ways. But this requires to specifies the *a-priori* non linear transformation and test if it is appropriate.

GAMM instead allow to bottom-up check if non-linear relationships (even complex) yield better fit, estimating the best degree of complexity.

# GAMM - Pros

## Pros

- non-linear transformations are estimated in a bottom up-fashion
- they allow to account for non-linear interactions between continuous variables
- they allow to consider several aspects (e.g. autocorrelation of residuals)



# GAMM - Cons

## Cons

- it is tempting to fit overly complex data.
- they are prone to overfit. (if not used properly).
- interpretation of complex effect is made by visual inspection.

# A note on GAMM modeling

In GAMM significance of terms can not be the best way to check for important predictor. It is better to use modeling approach, using model selection (see next section).

## Model Selection - AIC (1)

AIC and AIC comparison some drawbacks. In particular:

- it should be corrected for small samples
- there is no threshold for selecting a better model

this limitations are partly solved using the AICc, delta and AIC weights.

## Model Selection - AIC (2)

AICc, delta and AIC weights can be found in the MuMin package.

you can find a clear explanation [here](#) or [here](#)

A nice and detailed result is this article by Burnham and colleagues (2011). AIC model selection and multimodel inference in behavioral ecology: some background, observations, and comparisons

# Singular Fit

Sometimes the model results tells that it is singular. You can find a good explanation on the meaning in `?isSingular` from `lme4` package. See also [here](#)

Shortly a **singular** model means that probably you are overfitting the data, and you have too many parameters for your observations.

# Convergence and Optimizers

Convergence problems and their solutions, depend on the use of *Optimizers* in mixed model fitting.

Optimizers are an advanced topic. You can find a plain term explanation [here](#). But see also [here](#) and [here](#)

Shortly, mixed model fitting cannot be done with simple linear algebra steps (like ordinary lm). For this reason, it is necessary to use optimizers. Optimizers use algorithms to iteratively search for a solution and stop when the solution is found.

# Convergence and Optimizers

When there are issues in convergence, a good strategy is to test whether changing the optimizers changes the results, using the `allFit` function on the model.

In such case there is no precise indication on how similar the results should be across optimizers to be considered a good fit (it should be done qualitatively).

See for example [here](#) a thread in which Ben Bolkers gives this suggestion.

## Recommendations (before the experiment)

- 1 Use as many participants, and as many item as possible.
- 2 Avoid too much complicated design (especially if you have few participants and/or few items)
- 3 Make sure to have variability in the data.



# Recommendations for analyses

- 1 Try to reduce to a reasonable set of predictors (for better interpretability)
- 2 center predictors (with `scale`) function, to improve fit and avoid spurious random slope correlations.

# Recommendations (after the experiment)

- 1 Spend some time to check the data, and that everything is good.
- 2 use plots to check the data.
- 3 start the analysis with specific hypotheses.

# Correlation of fixed effects

It is printed when using `summary` on a mixed model.

It should not be confused with collinearity. It tells the expected correlation between the effects on a new sample. See [here](#) for details.

# Parsimonious mixed models and rePCA function

- start with a maximal model.
- use rePCA on the model to extract a PCA on variance-covariance component.
- scree plot for each random term to investigate if one component is enough to capture enough random-structure variance.

## *References*

The two main references are [this](#) and [this](#).

## Some considerations on Modeling procedures and checks

People are often concerned too much on whether they fit the correct model, with the correct procedure, or on diagnostics.

Remember, that all models are wrong (*cit.*), and that you should also care about *replication* of your effects, with different samples, and different conditions.

A “perfect” model could be still useless if it’s never replicated.