# Introduction to mixed-effects regression

## Lecture 1 of advanced regression for linguists

Martijn Wieling and Jacolien van Rij

Seminar für Sprachwissenschaft
University of Tübingen

LOT Summer School 2013, Groningen, June 24

# Course setup

- Five lectures from 9 AM - 11 AM:
  - Today: Introduction to mixed-effects regression with reaction time data
  - Tuesday: Mixed-effects regression and eye-tracking data
  - Wednesday: Introduction to generalized additive modeling with dialect data
  - Thursday: Generalized additive modeling with pupil data
  - Friday: Generalized additive modeling with EEG data
- User-centered, so each lecture:
  - Part I: introductory lecture (ca. 60 minutes)
  - Short break
  - Part II: hands-on lab session (ca. 45 minutes)
    - You won't finish all exercises from the lab session during the lecture. To get the most out of the course, try to finish them by yourself before the next lecture.
- Questions: ask immediately when something is unclear!
  - Caveat: I am not a statistician, so I won't have all the answers...

# Today's lecture

- Introduction
- Recap: multiple regression
- Mixed-effects regression analysis: explanation
- Methodological issues
- Case-study: Lexical decision latencies (Baayen, 2008: 7.5.1)
- Conclusion

# Introduction

- Consider the following situation (taken from Clark, 1973):
  - Mr. A and Mrs. B study reading latencies of verbs and nouns
  - Each randomly selects 20 words and tests 50 participants
  - Mr. A finds (using a sign test) verbs to have faster responses
  - Mrs. B finds nouns to have faster responses

- How is this possible?

# Introduction

- Consider the following situation (taken from Clark, 1973):
  - Mr. A and Mrs. B study reading latencies of verbs and nouns
  - Each randomly selects 20 words and tests 50 participants
  - Mr. A finds (using a sign test) verbs to have faster responses
  - Mrs. B finds nouns to have faster responses
- How is this possible?

# The language-as-fixed-effect fallacy

- The problem is that Mr. A and Mrs. B disregard the variability in the words (which is huge)
  - Mr. A included a difficult noun, but Mrs. B included a difficult verb
  - Their set of words does not constitute the complete population of nouns and verbs, therefore their results are limited to their words
- This is known as the language-as-fixed-effect fallacy (LAFEF)
  - Fixed-effect factors have repeatable and a small number of levels
  - Word is a random-effect factor (a non-repeatable random sample from a larger population)

# Why linguists are not always good statisticians

- LAFEF occurs frequently in linguistic research until the 1970's
  - Many reported significant results are wrong (the method is anti-conservative)!
- Clark (1973) combined a by-subject ($F_1$) analysis and by-item ($F_2$) analysis in a measure called *min F'*
  - Results are significant and generalizable across subjects and items when *min F'* is significant
  - Unfortunately many researchers (>50%!) incorrectly interpreted this study and may report wrong results (Raaijmakers et al., 1999)
  - E.g., they only use $F_1$ and $F_2$ and not *min F'* or they use $F_2$ while unneccesary (e.g., counterbalanced design)

# Our problems solved...

- Apparently, analyzing this type of data is difficult...
- Fortunately, using mixed-effects regression models solves these problems!
  - The method is easier than using the approach of Clark (1973)
  - Results can be generalized across subjects and items
  - Mixed-effects models are robust to missing data (Baayen, 2008, p. 266)
  - We can easily test if it is necessary to treat item as a random effect
- But first some words about regression...

# Our problems solved...

- Apparently, analyzing this type of data is difficult...
- Fortunately, using mixed-effects regression models solves these problems!
  - The method is easier than using the approach of Clark (1973)
  - Results can be generalized across subjects and items
  - Mixed-effects models are robust to missing data (Baayen, 2008, p. 266)
  - We can easily test if it is necessary to treat item as a random effect
- But first some words about regression...

# Regression vs. ANOVA

- Most people either use ANOVA or regression
  - ANOVA: categorical predictor variables
  - Regression: continuous predictor variables
- Both can be used for the same thing!
  - ANCOVA: continuous and categorical predictors
  - Regression: categorical (dummy coding) and continuous predictors
- Why I use regression as opposed to ANOVA
  - No temptation to dichotomize continuous predictors
  - Intuitive interpretation (your mileage may vary)
  - Mixed-effects analysis is relatively easy to do and does not require a balanced design (which is generally necessary for repeated-measures ANOVA)
- This course will focus on regression

# Recap: multiple regression

- Multiple regression: predict one numerical variable on the basis of other independent variables (numerical or categorical)
  - (*Logistic* regression is used to predict a categorical dependent)
- We can write a regression formula as $y = I + ax_1 + bx_2 + ...$
- E.g., predict the reaction time of a participant on the basis of word frequency, word length and speaker age:
  $RT = 200 - 5WF + 3WL + 10SA$

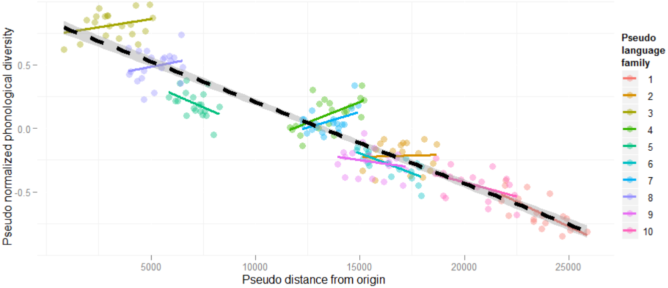
# Mixed-effects regression modeling: introduction

- Mixed-effects regression modeling distinguishes fixed-effects and random-effects factors
- Fixed-effects factors:
  - Repeatable levels
  - Small number of levels (e.g., Gender, Word Category)
  - Same treatment as in multiple regression (treatment coding)
- Random-effects factors:
  - Levels are a non-repeatable random sample from a larger population
  - Often large number of levels (e.g., Subject, Item)

# What are random-effects factors?

- Random-effect factors are factors which are likely to introduce systematic variation
  - Some participants have a slow response (RT), while others are fast
    = Random Intercept for Subject
  - Some words are easy to recognize, others hard
    = Random Intercept for Item
  - The effect of word frequency on RT might be higher for one participant than another: non-native speakers might benefit more from frequent words than native speakers
    = Random Slope for Item Frequency per Subject
  - The effect of speaker age on RT might be different for one word than another: modern words might be recognized easier by younger speakers
    = Random Slope for Subject Age per Item
- Note that it is essential to test for random slopes!

# Random slopes are necessary!




| | | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|---|
| Linear regression | DistOrigin | -6.418e-05 | 1.808e-06 | -35.49 | <2e-16 *** |
| + Random intercepts | DistOrigin | -2.224e-05 | 6.863e-06 | -3.240 | <0.001 *** |
| + Random slopes | DistOrigin | -1.478e-05 | 1.519e-05 | -0.973 | n.s. |

This example is explained at http://hlplab.wordpress.com

# Specific models for every observation

- Mixed-effects regression analysis allow us to use random intercepts and slopes (i.e. adjustments to the population intercept and slopes) to make the regression formula as precise as possible for every individual observation in our random effects
  - Parsimony: a single parameter (standard deviation) models this variation for every random slope or intercept (a normal distribution with mean 0 is assumed)
  - The adjustments to population slopes and intercepts are Best Linear Unbiased Predictors (BLUPs)
  - Likelihood-ratio tests assess whether the inclusion of random intercepts and slopes is warranted
- Note that multiple observations for each level of a random effect are necessary for mixed-effects analysis to be useful (e.g., participants respond to multiple items)

# Specific models for every observation

- $RT = 200 - 5WF + 3WL + 10SA$ (general model)
  - The intercepts and slopes may vary (according to the estimated standard variation for each parameter) and this influences the word- and subject-specific values

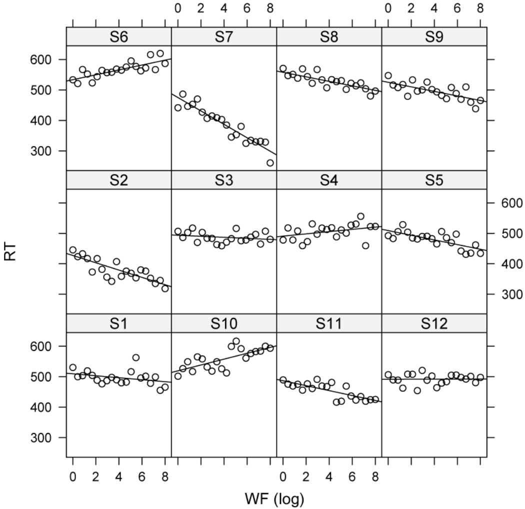
- $RT = 400 - 5WF + 3WL - 2SA$ (word: scythe)
- $RT = 300 - 5WF + 3WL + 15SA$ (word: twitter)
- $RT = 300 - 7WF + 3WL + 10SA$ (subject: non-native)
- $RT = 150 - 5WF + 3WL + 10SA$ (subject: fast)

- And it is easy to use!

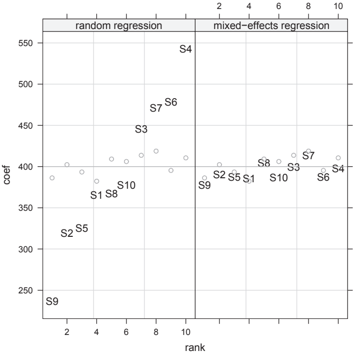
```
> lmer( RT ~ WF + WL + SA + (1+SA|Wrd) + (1+WF|Subj) )
```

- `lmer` figures out by itself if the random-effects are nested (schools-pupils), or crossed (participants-items)

# Specific models for every subject



# BLUPs of `lmer` do not suffer from shrinkage




- The BLUPS (i.e. adjustment to the model estimates per item/speaker) are close to the real adjustments, as `lmer` takes into account regression towards the mean (fast subjects will be slower next time, and slow subjects will be faster) thereby avoiding overfitting and improving prediction

# Methodological issues

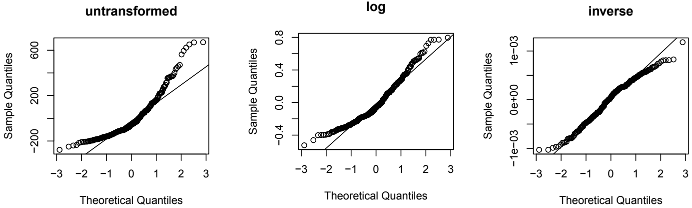
- Parsimony
- Assumptions about the residuals
  - Normally distributed and homoskedastic
  - No trial-by-trial dependencies
- Assumptions about the predictors
  - We assume linearity, but we will investigate non-linearities when discussing generalized additive modeling on Wednesday
- Model criticism

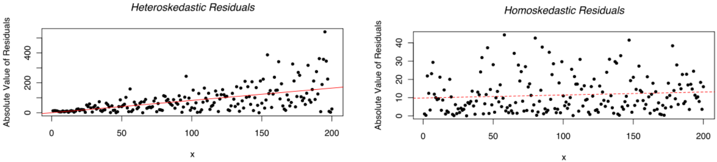
# Parsimony

- All models are wrong
- Some models are better than others
- The correct model can never be known with certainty
- The simpler the model, the better it is

# Residuals: normally distributed and homoskedastic

- The errors should follow a normal distribution with mean zero and the same standard deviation for any cell in your design, and for any covariate
    - If not then transform the dependent variable: log(Y), or -1000/Y
    - And use mixed-effects regression



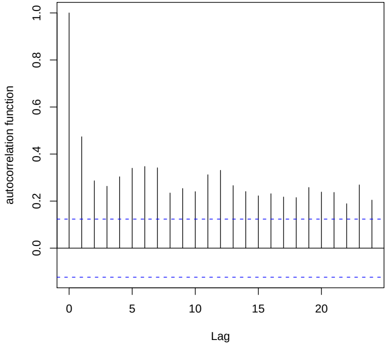
# Residuals: no trial-by-trial dependencies

- Residuals should be independent
  - With trial-by-trial dependencies, this second assumption is violated, which may result in models that underperform
- Possible remedies:
  - Include trial as a predictor in your model
  - Include the value of the dependent variable at the previous trial as a predictor in your model

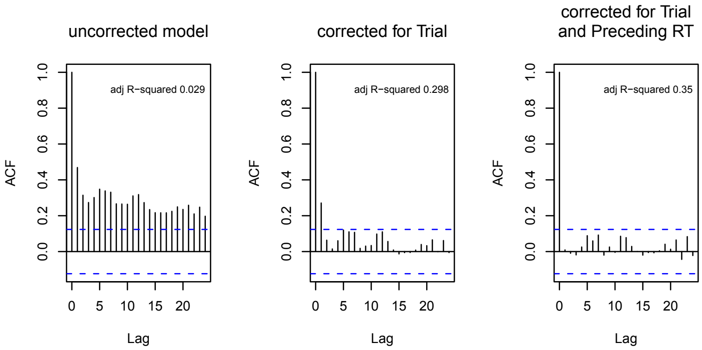
# Trial-by-trial dependencies in a word naming task

- Word naming (reading aloud) of Dutch verbs
- Trial-by-trial dependencies for subject 19

```
> acf(dat$RTinv, main=" ", ylab="autocorrelation function")
```



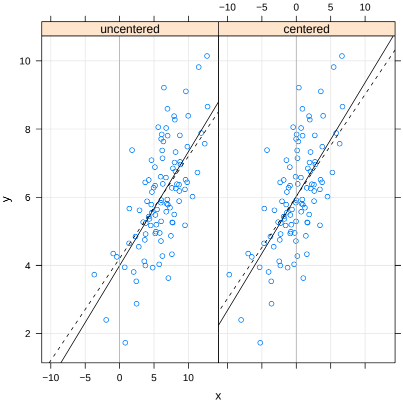
# Modeling trial-by-trial dependencies



# Model criticism

- Check the distribution of residuals: if not normally distributed then transform dependent variable (as illustrated before)
- Check outlier characteristics and refit the model when large outliers are excluded to verify that your effects are not ‘carried’ by these outliers
- Important: no *a priori* exclusion of outliers without a clear reason
  - A good reason is not that the value is over 2.5 SD above the mean
  - A good reason (e.g.,) is that the response is faster than possible

# Center your variables (i.e. subtract the mean)




- Otherwise random slopes and intercepts may show a spurious correlation
- Also helps the interpretation of factorial predictors in model (marking differences at means of other variables, rather than at values equal to 0)

# Case study: long-distance priming

- De Vaan, Schreuder & Baayen (The Mental Lexicon, 2007)
- Design
  - long-distance priming (39 intervening items)
  - base condition (`baseheid`): base preceded neologism (fluffy - fluffiness)
  - derived condition (`heid`): identity priming (fluffiness - fluffiness)
- Prediction
  - Subjects in the derived condition (`heid`) would be faster than those in the base condition (`baseheid`)

# A first model: counterintuitive results!

(note: $t > 2 \Rightarrow p < 0.05$, for $N \gg 100$)

```
> library(lme4)
> library(languageR)
> dat = read.table('datprevrt.txt', header=T) # adapted primingHeid data set
> dat.lmer1 = lmer(RT ~ Condition + (1|Word) + (1|Subject), data=dat)
> print(dat.lmer1, corr=FALSE)

   AIC    BIC logLik deviance REMLdev
 -92.4 -68.79   51.2   -113.5  -102.4
Random effects:
 Groups   Name        Variance  Std.Dev.
 Word     (Intercept) 0.0034112 0.058405
 Subject  (Intercept) 0.0408434 0.202098
 Residual             0.0440842 0.209962
Number of obs: 832, groups: Word, 40; Subject, 26
Fixed effects:
              Estimate Std. Error t value
(Intercept)    6.60296    0.04215  156.66
Conditionheid  0.03127    0.01467    2.13 # slower in heid than baseheid...
```

# Evaluation

- Counterintuitive inhibition
- But various potential factors are not accounted for in the model
  - Longitudinal effects: trial rank, RT to preceding trial
  - RT to prime as predictor
  - Response to the prime (correct/incorrect): a yes response to a target associated with a previously rejected prime may take longer
  - The presence of atypical outliers

# An effect of trial?

```
> dat.lmerA = lmer(RT ~ Trial + Condition + (1|Subject) + (1|Word),
                  data=dat)
> summary(dat.lmerA)@coefs

                  Estimate   Std. Error    t value
(Intercept)    6.6333837122 4.666295e-02 142.155253
Trial         -0.0001461422 9.619663e-05  -1.519203
Conditionheid  0.0309771010 1.465343e-02   2.113983
```

# An effect of previous trial RT?

```
> dat$PrevRT = log(dat$PrevRT) # RT is already log-transformed
> dat.lmerA = lmer(RT ~ PrevRT + Condition + (1|Subject) + (1|Word),
                   data=dat)
> summary(dat.lmerA)@coefs
```

```
               Estimate Std. Error  t value
(Intercept)   5.80464482 0.22298097 26.032019
PrevRT        0.12124596 0.03337103  3.633270
Conditionheid 0.02785278 0.01463263  1.903471
```

# An effect of RT to prime?

```
> dat.lmerA = lmer(RT ~ RTtoPrime + PrevRT + Condition + (1|Subject)
                      + (1|Word), data=dat)
> summary(dat.lmerA)@coefs

                 Estimate Std. Error    t value
(Intercept)    4.74877346 0.29531776 16.0802165
RTtoPrime      0.16378549 0.03185814  5.1410883
PrevRT         0.11900751 0.03301011  3.6051835
Conditionheid -0.00611743 0.01599205 -0.3825295
```

# An effect of the decision for the prime?

```
> dat.lmerA = lmer(RT ~ RTtoPrime + ResponseToPrime + PrevRT + Condition
                       + (1|Subject) + (1|Word), data=dat)
> summary(dat.lmerA)@coefs

                            Estimate Std. Error   t value
(Intercept)               4.76343629 0.29225924 16.298668
RTtoPrime                 0.16495149 0.03146911  5.241696
ResponseToPrimeincorrect  0.10041997 0.02258933  4.445460
PrevRT                    0.11420383 0.03268044  3.494562
Conditionheid            -0.01777176 0.01605670 -1.106813
```

# Interaction for prime-related predictors?

```
> dat.lmerA = lmer(RT ~ RTtoPrime * ResponseToPrime + PrevRT + Condition
                   + (1|Subject) + (1|Word), data=dat)
> summary(dat.lmerA)@coefs

                                      Estimate Std. Error   t value
(Intercept)                         4.32440861 0.31519809 13.719654
RTtoPrime                           0.22763285 0.03593658  6.334293
ResponseToPrimeincorrect            1.45478703 0.40524815  3.589867
PrevRT                              0.11833846 0.03250820  3.640265
Conditionheid                      -0.02656561 0.01617865 -1.642017
RTtoPrime:ResponseToPrimeincorrect -0.20249849 0.06055956 -3.343791
```

- Interpretation: the RT to the prime is only predictive for the RT of the target word when the prime was judged to be a correct word

# An effect of base frequency?

(Note the lower variance of the random intercept for word: previous value was 0.0034)

```
> dat.lmerA = lmer(RT ~ RTtoPrime * ResponseToPrime + PrevRT + BaseFrequency
                          + Condition + (1|Subject) + (1|Word), data=dat)
> summary(dat.lmerA)@coefs

Random effects:
 Groups   Name        Variance  Std.Dev.
 Word     (Intercept) 0.0011514 0.033932
 Subject  (Intercept) 0.0239910 0.154890
 Residual             0.0422398 0.205523
Number of obs: 832, groups: Word, 40; Subject, 26

Fixed effects:
                                      Estimate  Std. Error   t value
(Intercept)                        4.440969006 0.319604869 13.895186
RTtoPrime                          0.218242365 0.036152502  6.036715
ResponseToPrimeincorrect           1.397052342 0.405163681  3.448118
PrevRT                             0.115425086 0.032455493  3.556411
BaseFrequency                     -0.009242775 0.004370665 -2.114730
Conditionheid                     -0.024656390 0.016178566 -1.524016
RTtoPrime:ResponseToPrimeincorrect -0.193986804 0.060549680 -3.203763
```

# Testing random slopes: no main frequency effect!

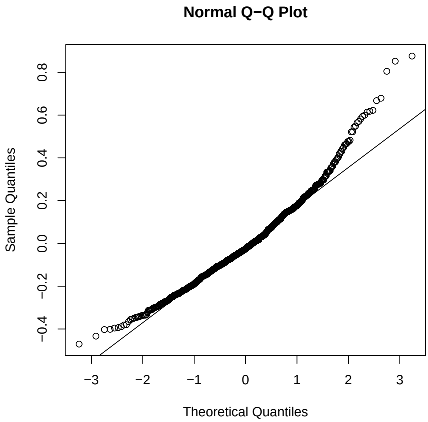
```
> dat.lmerA2 = lmer(RT ~ RTtoPrime * ResponseToPrime + PrevRT
                     + BaseFrequency + Condition + (1|Subject)
                     + (0+BaseFrequency|Subject) + (1|Word), data=dat)
> anova(dat.lmerA,dat.lmerA2) # compares simpler model to more complex model

           Df     AIC     BIC logLik  Chisq Chi Df Pr(>Chisq)
dat.lmerA  10 -165.99 -118.75 92.994
dat.lmerA2 11 -169.37 -117.41 95.684 5.3806      1    0.02036 *

> summary(dat.lmerA2)@coefs

                                     Estimate  Std. Error   t value
(Intercept)                       4.482297852 0.317364029 14.123522
RTtoPrime                         0.218117437 0.035948557  6.067488
ResponseToPrimeincorrect          1.416751948 0.402057111  3.523758
PrevRT                            0.108490899 0.032351291  3.353526
BaseFrequency                    -0.007946724 0.005351489 -1.484956
Conditionheid                    -0.024535184 0.016033756 -1.530221
RTtoPrime:ResponseToPrimeincorrect -0.196673139 0.060079998 -3.273521
```

# Testing for correlation parameters in random effects

```
> dat.lmerA3 = lmer(RT ~ RTtoPrime * ResponseToPrime + PrevRT
                         + BaseFrequency + Condition
                         + (1+BaseFrequency|Subject) + (1|Word), data=dat)
> print(dat.lmerA3, corr=F)

...
Random effects:
 Groups   Name          Variance  Std.Dev. Corr
 Word     (Intercept)   0.0011857 0.034434
 Subject  (Intercept)   0.0166779 0.129143
          BaseFrequency 0.0001861 0.013642 0.406
 Residual               0.0414191 0.203517
Number of obs: 832, groups: Word, 40; Subject, 26
...

> anova(dat.lmerA2,dat.lmerA3)

           Df     AIC     BIC logLik Chisq Chi Df Pr(>Chisq)
dat.lmerA2 11 -169.37 -117.41 95.684
dat.lmerA3 12 -168.31 -111.62 96.155 0.941      1      0.332
```

# Model criticism

```
> qqnorm(resid(dat.lmerA2))
> qqline(resid(dat.lmerA2))
```



# The trimmed model

```
> dat2 = dat[ abs(scale(resid(dat.lmerA2))) < 2.5 , ]
> dat.lmerB2 = lmer(RT ~ RTtoPrime * ResponseToPrime + PrevRT
                        + BaseFrequency + Condition + (1|Subject)
                        + (0+BaseFrequency|Subject) + (1|Word), data=dat2)
> summary(dat.lmerB2)@coefs
```

```
                                      Estimate  Std. Error   t value
(Intercept)                        4.447353314 0.285976261 15.551477
RTtoPrime                          0.235109620 0.031999967  7.347183
ResponseToPrimeincorrect           1.560005900 0.355513219  4.388039
PrevRT                             0.095539172 0.029256885  3.265528
BaseFrequency                     -0.008150909 0.004591343 -1.775278
Conditionheid                     -0.038137598 0.014354398 -2.656858
RTtoPrime:ResponseToPrimeincorrect -0.216159053 0.053158270 -4.066330
```

# The trimmed model

- Just 2% of the data removed

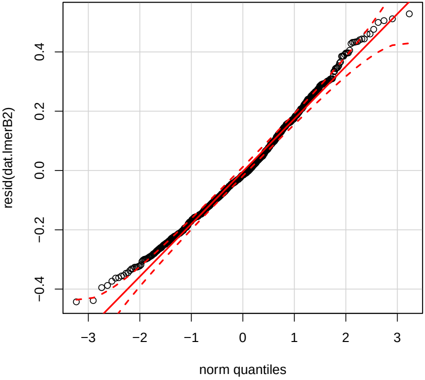
```
> noutliers = sum(abs(scale(resid(dat.lmerA2)))>=2.5)
> noutliers

[1] 17

> noutliers/nrow(dat)

[1] 0.02043269
```

- Improved fit (explained variance):

```
> cor(dat$RT,fitted(dat.lmerA2))^2
[1] 0.52106

> cor(dat2$RT,fitted(dat.lmerB2))^2
[1] 0.5717716
```

# Checking the residuals of trimmed model

```
> library(car)
> qqp(resid(dat.lmerB2))
```



# MCMC sampling to determine significance

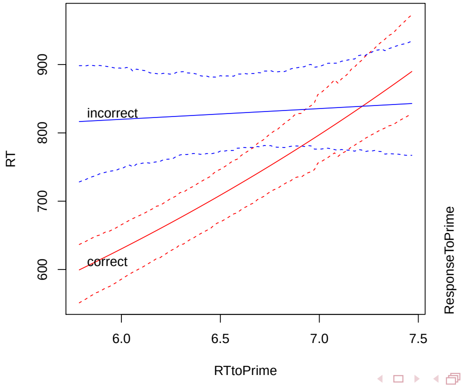
## Note: does not work with correlated random effects

```
> library(languageR)
> pvals.fnc(dat.lmerB2, withMCMC=T)

                                   Estimate MCMCmean ...  pMCMC Pr(>|t|)
(Intercept)                          4.4474   4.2570     0.0001   0.0000
RTtoPrime                            0.2351   0.2500     0.0001   0.0000
ResponseToPrimeincorrect             1.5600   1.6285     0.0001   0.0000
PrevRT                               0.0955   0.1089     0.0010   0.0011
BaseFrequency                       -0.0082  -0.0074     0.1316   0.0762
Conditionheid                       -0.0381  -0.0408     0.0052   0.0080
RTtoPrime:ResponseToPrimeincorrect  -0.2162  -0.2265     0.0001   0.0001
```

# MCMC-based confidence intervals

## Note: does not work with correlated random effects

```
> plotLMER.fnc(dat.lmerB2, pred="RTtoPrime", intr=list("ResponseToPrime",
    levels(dat2$ResponseToPrime), "beg", list(col=c("red", "blue"), lty=rep
    (1,2))), fun=exp, mcmcMat=lmerB2.mcmc$mcmc, cexsize=1.0, verbose=FALSE)
```



# Conclusion

- Mixed-effects regression is more flexible than using ANOVAs
- Testing for inclusion of random intercepts and slopes is essential when you have multiple responses per subject or item
- Mixed-effects regression is easy with `lmer` in `R`
- After the break: lab-session to illustrate the commands used here
- Tomorrow: more about mixed-effects regression using eye-tracking data

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

NWO Netherlands Organisation for Scientific Research

# Thank you for your attention!