Running A Mixed Model in R with *lmer*

A step-to-step guide of the essential

Teresa Del Bianco 02-10-20

What is *lmer*?

- Lmer is a function included in the package lme4 that fits linear mixed effects model
- As a function, lmer takes arguments:
  - The response variable Y
  - The independent variables (X) = fixed effects
  - The random effects

} Formula

  - The data

The Working Example: Reaction Times In A Sleep Deprivation Study

Background: On day 0 the subjects had their normal amount of sleep. Starting that night they were restricted to 3 hours of sleep per night for 10 days. The observations represent the average reaction time on a series of tests given each day to each subject (see Belenky et al., 2003)

Variables:

- Reaction: Average reaction time (ms).
- Days: Number of days of sleep deprivation
- Subject: Subject number on which the observation was made (18)

Key functions: head()

The Formula

```
m1 <- lmer(Reaction #this is our dependent variable
           ~ Days + #this is the fixed term or predictor: the effect we are mostly
                    # interested in!
             ( 1 + Days | Subject), #This expression means:
                                    # let each Subject have a different baseline RT
                                    # (intercept) and an individual level of variation
                                    # of RT between Days (slope) --> let each Subject
                                    # have a varying intercept and slope
           data=sleepstudy)
```

Key functions: library(), lmer()

The Model Output

```
## Linear mixed model fit by REML. t-tests use
##   Satterthwaite's method [lmerModLmerTest]
## Formula: Reaction ~ Days + (1 + Days || Subject)
##    Data: sleepstudy
##
## REML criterion at convergence: 1743.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9536 -0.4634  0.0231  0.4634  5.1793
##
## Random effects:
##  Groups    Name        Variance Std.Dev. Corr
##  Subject   (Intercept) 612.10   24.741
##            Days         35.07    5.922   0.07
##  Residual              654.94   25.592
## Number of obs: 180, groups:  Subject, 18
##
## Fixed effects:
##             Estimate Std. Error      df t value
## (Intercept)  251.405      6.825  17.000  36.838
## Days          10.467      1.546  17.000   6.771
##             Pr(>|t|)
## (Intercept)  < 2e-16 ***
## Days         3.26e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.138
```

Key functions: summary()

The Random Effects

Groups	Name	Variance	Std.Dev.	Corr
Subject	(Intercept)	612	24.7	
	Days	35	5.9	0.07
Residual		655	25.6	

Key functions: VarCorr(), print()

The Confidence Intervals of the Fixed Effects

```
Random effects:
 Groups   Name        Variance
 Subject  (Intercept) 612.10
          Days         35.07
 Residual             654.94
```

```
##
## Fixed effects:
##             Estimate Std. Error      d
## (Intercept)  251.405      6.825  17.0(
## Days          10.467      1.546  17.0(
##             Pr(>|t|)
## (Intercept)  < 2e-16 ***
## Days        3.26e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
##                                  2.5 %      97.5 %
## sd_(Intercept)|Subject       14.3814012  37.7159906
## cor_Days.(Intercept)|Subject -0.4815008   0.6849863
## sd_Days|Subject               3.8011642   8.7533658
## sigma                        22.8982669  28.8579964
## (Intercept)                 237.6806955 265.1295145
## Days                          7.3586533  13.5759188
```

Key functions: set.seed(), confint()

The Deviance Test

```
Analysis of Deviance Table (Type III Wald chisquare tests)

Response: Reaction
             Chisq Df Pr(>Chisq)
(Intercept) 1357.045  1  < 2.2e-16 ***
Days          45.853  1  1.275e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Key functions: library(), Anova()

Reporting

- Parameter estimates (example: Intercept = …, Slope = …), Standard Errors, Confidence Intervals (and method used to calculate them). Example: SE = …, 95% CI = … - ...
- Anova: Test statistics, degrees of freedom, significance tests.

  Example: Chi/F/T(df) = …, p </= …

- Variance, Standard deviations and correlations of random effects with confidence intervals

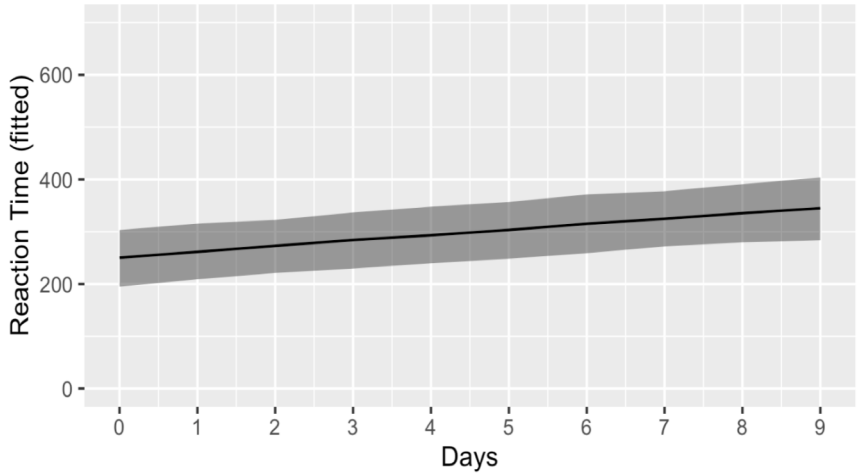
Plotting a Mixed Model

- My method: I generate prediction intervals to plot
- It’s not a standardised way, it’s just my way!

Key functions: expand.grid(), library(), predictInterval()

Plotting a Mixed Model

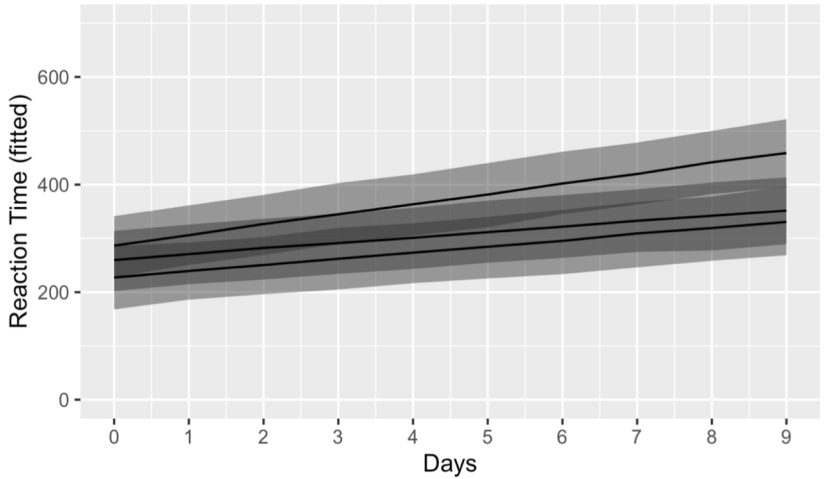
- Fixed Effects: plotting the fitted values of an “invented” subject with 95% confidence intervals



Key functions: library(), ggplot()

Plotting a Mixed Model

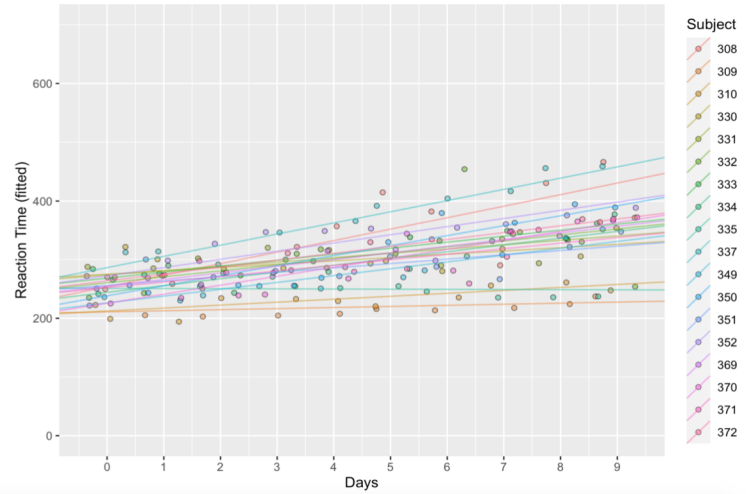
- Representing the Random Effect: plotting the fitted values of 3 random subjects, with 95% confidence intervals



Key functions: library(), ggplot()

Plotting a Mixed Model

- Representing the Random Effect: plotting all the individual coefficients plus the raw data



Key functions: library(), ggplot()

Exercise

What if I want to plot the predicted values of all subjects with 95% confidence intervals?

Key functions: library(), expand.grid(), predictInterval(), ggplot()

Now it's your turn! Fitting LMM on the Stroop Dataset

A dataset containing reaction-times, accuracy, and other attributes (10 total variables) of 5400 experimental trials (Stroop, 1935).

Task: Build a model that investigates how reaction times change based on target_type, assuming varying intercept and slope for subjects

Use the variables:

- Subject: Case identifier, in numerals
- Target_type: Type of stimulus for a given trial. 1 means congruent stimulus, 2 means incongruent stimulus
- Rt: Reaction time, in milliseconds

Hint: use *lmer* from library *lmerTest* to fit the formula *rt ~ Target_Type + (…)* ← *write the random effect in brackets!*