

Detecting faked IATs via task-switch costs

Anke Cajar

(data from André Krügel)

Background on data

This is data from an experiment showing that we can reliably detect whether outcomes from the Implicit Association Test (IAT) are faked by analysing task-switch costs in the combined blocks of the IAT.* Every participant performed two IATs: the control group performed the same normative IAT twice and the faking group was instructed to fake the second IAT by deliberately slowing down response times in the compatible block. It has been shown that switches from a target-concept word to an attribute-dimension word between consecutive trials produces stronger switch costs (i.e., response-time differences between task-repetition and task-switch trials) in the incompatible block than in the compatible block. The present data show that even after successful faking of the IAT, these switch costs are preserved (although the true compatible block became the faked incompatible block). Thus, switch costs can be used to detect IAT faking reliably and with high accuracy.

* Background on the IAT and faking strategies: The IAT measures people's implicit associations between a target concept and an attribute dimension. People have to assign consecutively presented words as fast as possible to one of four categories—two categories belong to the target concept (e.g., family vs career words) and two categories belong to the attribute dimension (e.g., male vs female words). When strongly associated concepts share the same response key in the categorization task (e.g., career and male words or family and female words), response times are usually shorter than when less associated concepts share the same response key (e.g., career and female words or family and male words). The IAT block with shorter response times is called the compatible block, whereas the block with longer response times is called the incompatible block. IAT results can be deliberately faked, most easily and reliably by slowing down response times in the actual compatible block (making it look like the incompatible block and thus reversing associations). To date, an effective and accurate method for detecting faked IATs is still missing.

Variables needed

2 x 2 x 2 x 2 factorial mixed design

Data table on a trial-by-trial basis:

- ID: Participant ID
- Item: ID for presented word
- Time (within-subject):
 - Baseline: first IAT (normative IAT)
 - Retest: second IAT (normative or faked, depending on Group)
- Group (between-subject):
 - No_Faking: control group, where people took the same normative IAT twice
 - Faking: experimental group, where people were instructed to fake the retest IAT by slowing down RTs in the compatible block

- Block (within-subject):
 - Compatible: IAT combined block with shorter RTs
 - Incompatible: IAT combined block with longer RTs
- TaskSwitch (within-subject):
 - Yes: Switch from target concept to attribute dimension (or the other way around) from one trial to the next
 - No: No switch from target concept to attribute dimension (or the other way around) from one trial to the next
- RT: trial response time

Load packages

```
rm(list=ls())

library(tidyverse)
library(MASS)
library(lme4)
library(car)
```

Read data and change factor levels

```
iat <- read.table("IAT_data.dat", header=TRUE, stringsAsFactors=TRUE)

iat <- iat %>% mutate(Group = relevel(Group, ref="No_Faking"))
str(iat)
```

```
## 'data.frame':    12160 obs. of  7 variables:
## $ ID           : int  1 1 1 1 1 1 1 1 1 1 ...
## $ Item          : int  13 11 5 5 12 12 20 9 9 17 ...
## $ Time          : Factor w/ 2 levels "Baseline","Retest": 1 1 1 1 1 1 1 1 1 1 ...
## $ Group         : Factor w/ 2 levels "No_Faking","Faking": 1 1 1 1 1 1 1 1 1 1 ...
## $ Block         : Factor w/ 2 levels "Compatible","Incompatible": 1 1 1 1 1 1 1 1 1 1 ...
## $ TaskSwitch    : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 1 2 1 2 ...
## $ RT            : int  795 551 656 477 668 412 634 656 464 638 ...
```

Remove RT outliers

(see IAT scoring procedure from Greenwald et al., 2003)

```
iat <- iat %>%
  filter(RT >= 400 & RT <= 10000)

summary(iat$RT)
```

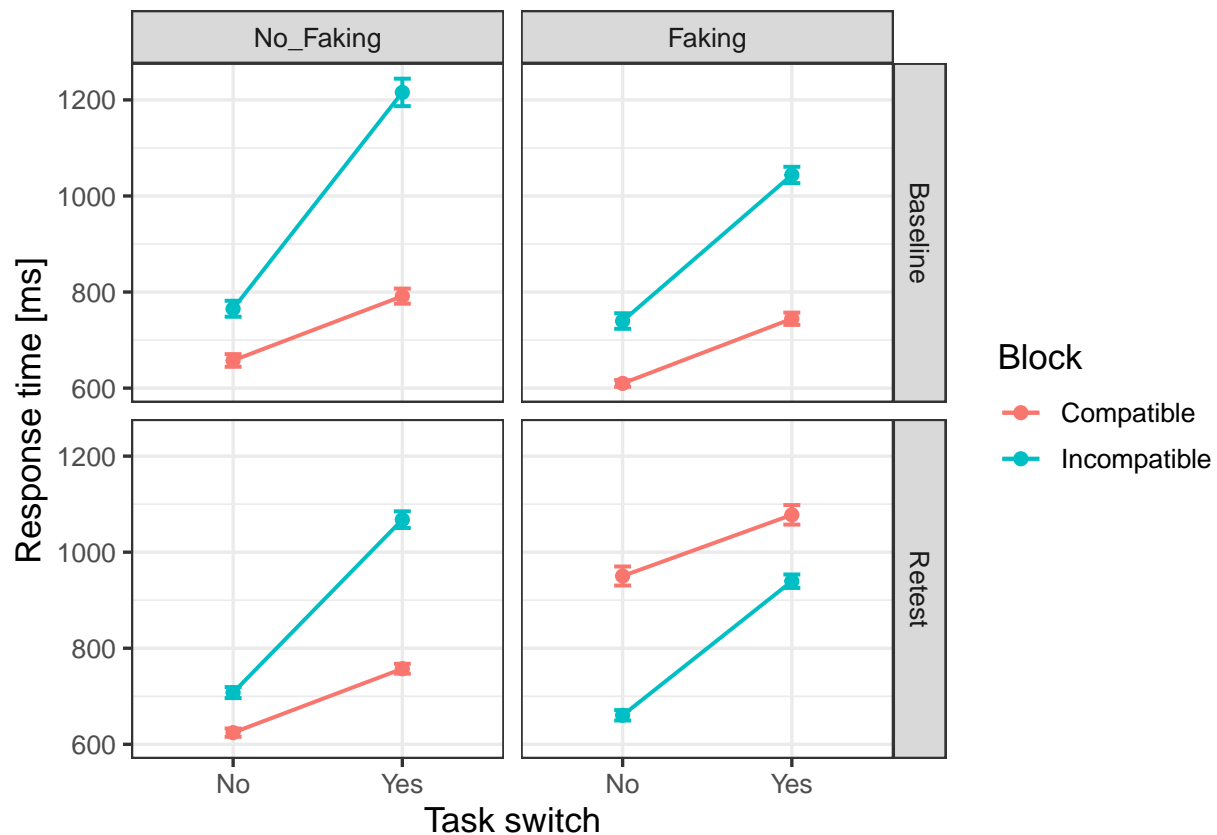
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  400.0   575.0   711.0   845.2   958.0  9003.0
```

Plot switch costs

```
switch_costs <- iat %>%
  group_by(Time, Group, Block, TaskSwitch) %>%
  summarise(M = mean(RT), SD=sd(RT), N=n(), SE=SD/sqrt(N))
```

`summarise()` has grouped output by 'Time', 'Group', 'Block'. You can override
using the `.groups` argument.

```
fig_RT <- ggplot(switch_costs, aes(x=TaskSwitch, y=M, color=Block)) +
  geom_point(size=2) +
  geom_line(size=0.7, aes(group=Block)) +
  geom_errorbar(aes(ymin=M-SE, ymax=M+SE), width=.1, size=0.7) +
  facet_grid(Time~Group) +
  labs(x="Task switch", y="Response time [ms]") +
  theme_bw(base_size=13)
print(fig_RT)
```



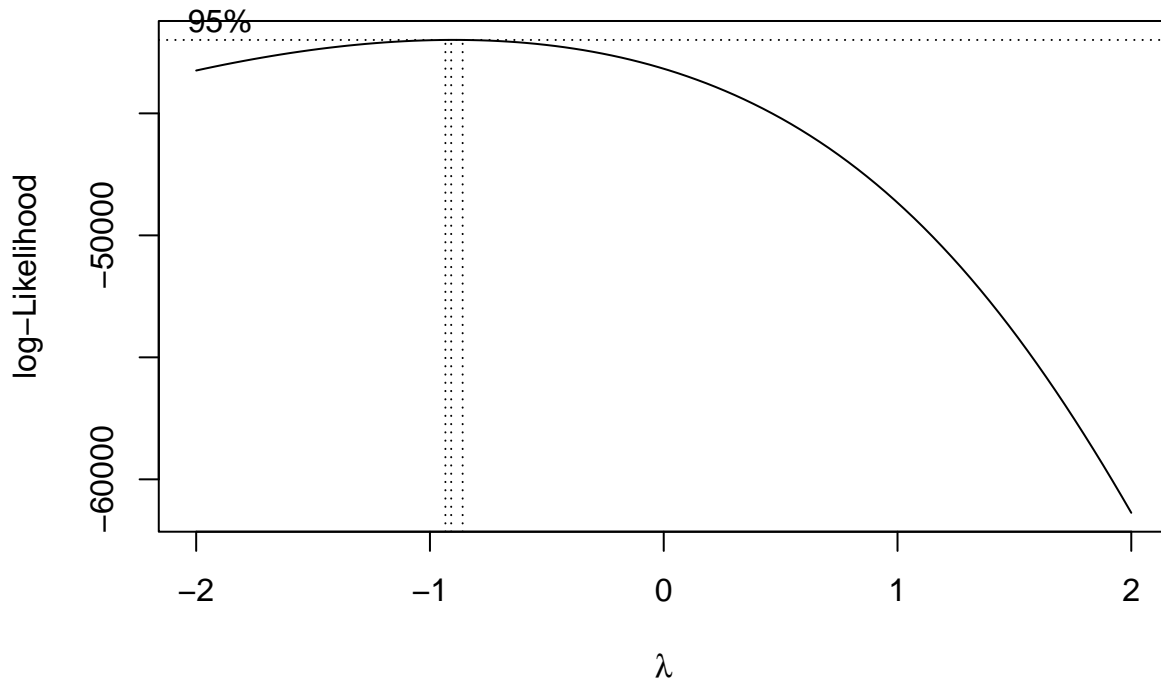
LMMs for switch costs

Define contrasts

```
contrasts(iat$Group) <- contr.treatment(2)
contrasts(iat$Time) <- contr.treatment(2)
contrasts(iat$Block) <- contr.sum(2)/-2
contrasts(iat$TaskSwitch) <- contr.sum(2)/-2
```

Transform RTs

```
lambdaList <- boxcox(lm(RT ~ Group*Time*Block*TaskSwitch, data=iat))
```

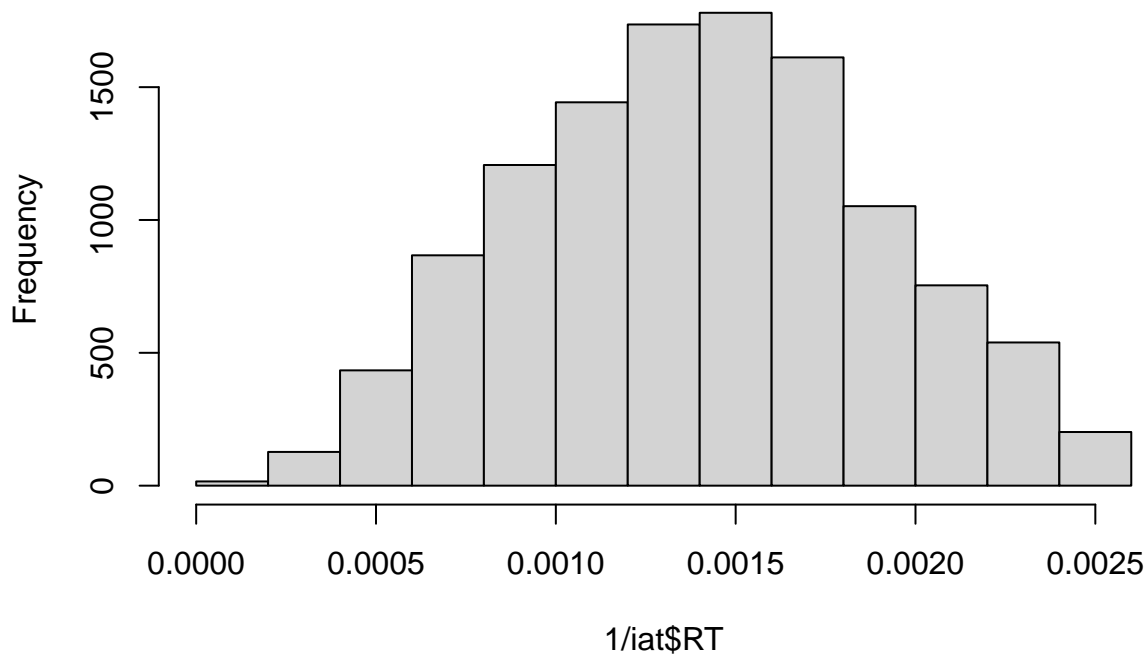


```
(lambda <- lambdaList$x[which.max(lambdaList$y)])
```

```
## [1] -0.9090909
```

```
hist(1/iat$RT)
```

Histogram of 1/iat\$RT



Run LMMs

Models that don't converge (I tried model selection by starting with the maximal model and working my way down to the minimal model):

```
# model1 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time*Block*TaskSwitch|ID) + (1 + Time*Block*
# model2 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time*Block*TaskSwitch||ID) + (1 + Time*Block*
# model3 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time+Block+TaskSwitch|ID) + (1 + Time+Block+
# model4 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time+Block+TaskSwitch||ID) + (1 + Time+Block+
# model5 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time*Block*TaskSwitch|ID) + (1|Item), data=i
# model6 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time*Block*TaskSwitch||ID) + (1|Item), data=
# model7 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time+Block+TaskSwitch|ID) + (1|Item), data=i
# model8 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time+Block+TaskSwitch||ID) + (1|Item), data=
# model9 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time+TaskSwitch||ID) + (1|Item), data=iat, R
# model10 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time||ID) + (1|Item), data=iat, REML=FALSE)
# model11 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1 + Time||ID) + (1 + Time||Item), data=iat, REML=FALSE)
```

Minimal model works:

```
model12 <- lmer(1/RT ~ Group*Time*Block*TaskSwitch + (1|ID) + (1|Item), data=iat, REML=FALSE)
```

Surprisingly, this one also works (same as model 7, but with different order of factors):

```
model13 <- lmer(1/RT ~ Group*Time*TaskSwitch*Block + (1 + Time+TaskSwitch+Block|ID) + (1|Item), data=iat)
summary(model13)
```

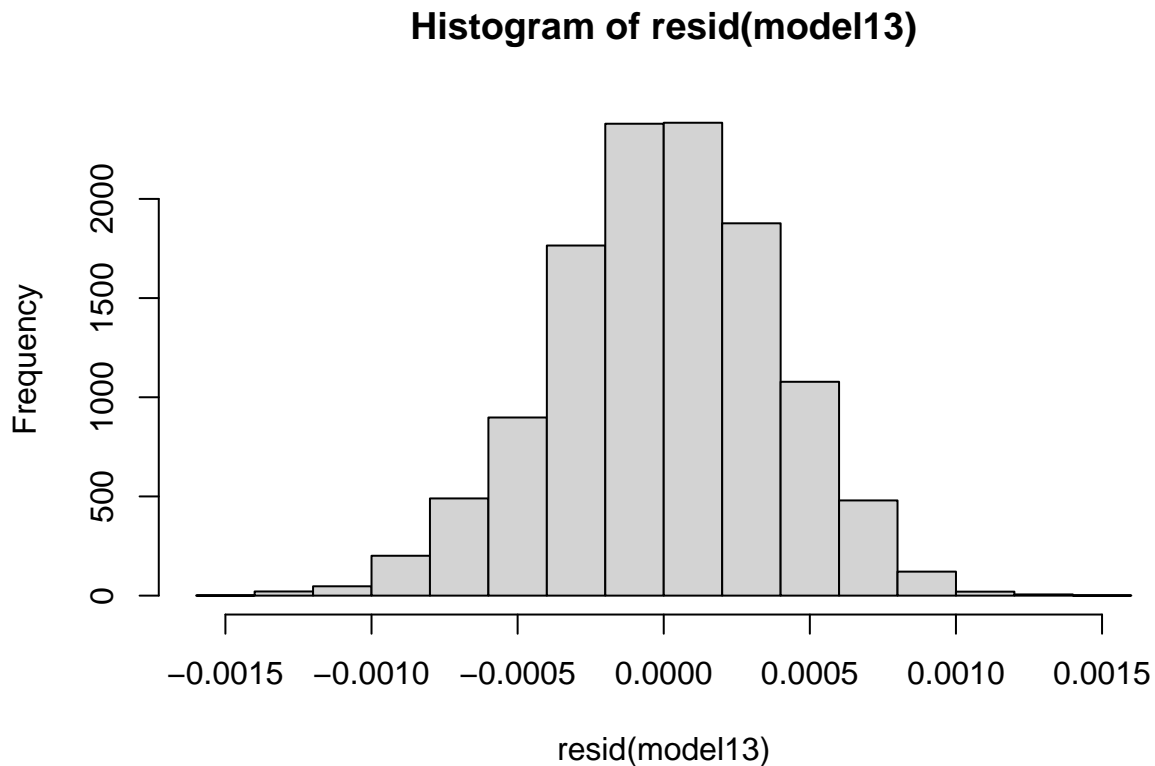
```
## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula: 1/RT ~ Group * Time * TaskSwitch * Block + (1 + Time + TaskSwitch +
##      Block | ID) + (1 | Item)
##      Data: iat
##
##      AIC      BIC    logLik deviance df.resid
## -151477.8 -151271.4  75766.9 -151533.8    11741
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.9956 -0.6391  0.0143  0.6818  3.7502
##
## Random effects:
##      Groups      Name      Variance Std.Dev.  Corr
##      ID      (Intercept) 3.098e-08 1.760e-04
##      Time2      1.867e-08 1.366e-04 -0.37
##      TaskSwitch1 4.498e-09 6.707e-05 -0.05  0.16
##      Block1      1.518e-08 1.232e-04  0.12 -0.77  0.02
##      Item      (Intercept) 4.395e-10 2.096e-05
##      Residual      1.453e-07 3.811e-04
## Number of obs: 11769, groups:  ID, 38; Item, 20
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)    1.421e-03  4.126e-05  34.442
## Group2          5.815e-05  5.797e-05   1.003
## Time2           4.393e-05  3.290e-05   1.335
## TaskSwitch1     -3.984e-04  2.089e-05 -19.070
```

```
## Block1 -2.800e-04 3.159e-05 -8.862
## Group2:Time2 -1.897e-04 4.652e-05 -4.078
## Group2:TaskSwitch1 2.127e-05 2.952e-05 0.721
## Time2:TaskSwitch1 -4.223e-06 2.001e-05 -0.211
## Group2:Block1 -3.972e-06 4.466e-05 -0.089
## Time2:Block1 2.278e-05 1.999e-05 1.140
## TaskSwitch1:Block1 -2.602e-04 2.824e-05 -9.214
## Group2:Time2:TaskSwitch1 5.965e-05 2.824e-05 2.112
## Group2:Time2:Block1 5.006e-04 2.821e-05 17.743
## Group2:TaskSwitch1:Block1 4.629e-05 3.987e-05 1.161
## Time2:TaskSwitch1:Block1 3.955e-05 3.999e-05 0.989
## Group2:Time2:TaskSwitch1:Block1 -1.229e-04 5.645e-05 -2.178

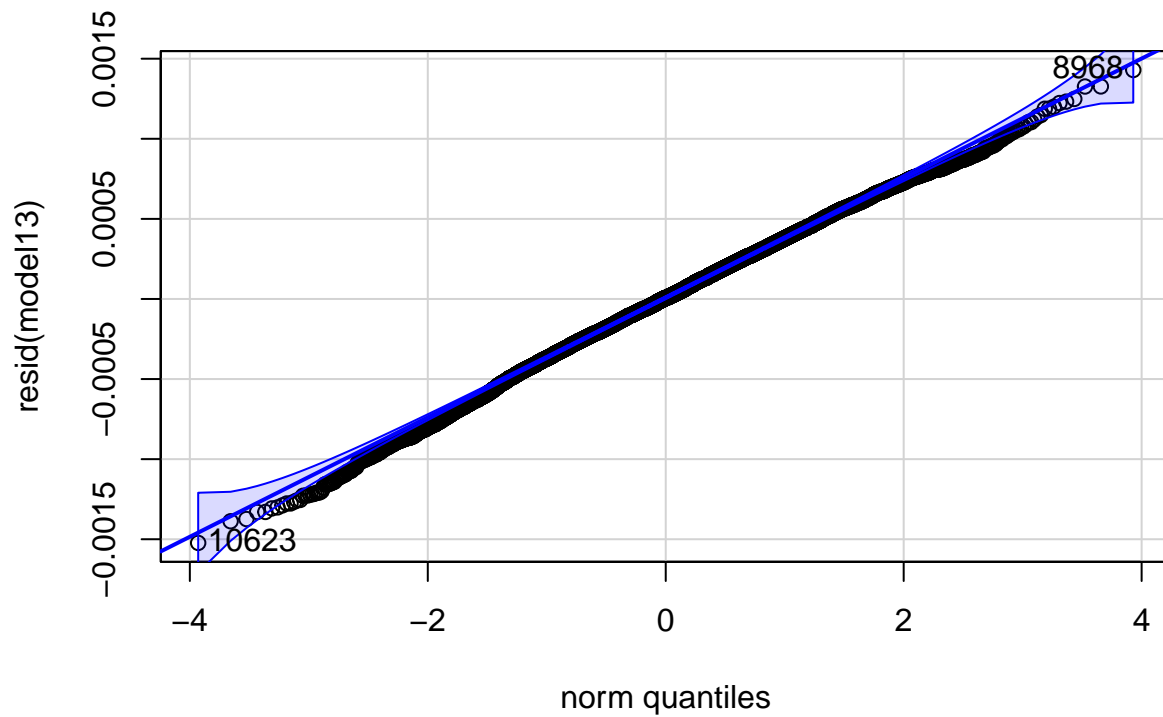
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE) or
## vcov(x) if you need it
```

Regression diagnostics for model 13

```
hist(resid(model13))
```



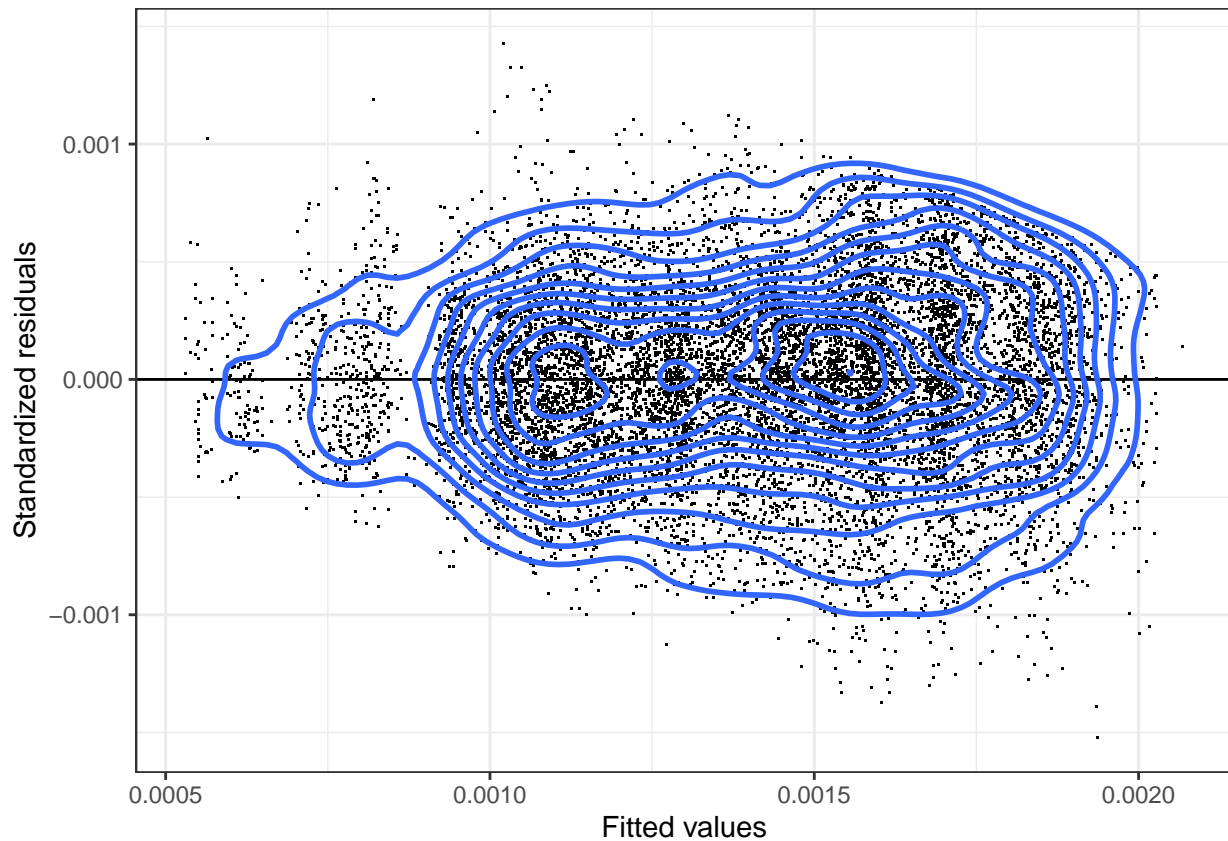
```
qqPlot(resid(model13))
```



```
## [1] 10623 8968
```

```
qplot(x=fitted(model13), y=resid(model13), geom="point", shape=I("."),
      xlab="Fitted values", ylab="Standardized residuals") +
  geom_hline(yintercept=0) + theme_bw() + geom_hex() + geom_density2d(size=1)
```

```
## Warning: Computation failed in `stat_binhex()`:
```



Analysis/modeling issues:

- Is the contrast coding appropriate?
- What is currently the best method for model selection (concerning questions like: from maximal to minimal model, doing rePCA, ...)? Is the stuff from the RePsychLing package and vignettes still the way to do it?
- How do I make sure that the random effects structure of my selected model is really supported by the data (again, rePCA)?
- How do I decide which optimizer to use for model fitting? I heard bobyqa is supposed to be good...
- It seems that the order in which I put the factors in the formula changes the outcome. How do I decide on the order (most important factor first, than second etc.)?
- Model results (model 13) don't really fit the mean response time/switch costs plot, as part of the effects you see in the figure goes into the random effects (By chance, there were more older subjects > 45 years in the no-faking group, who had considerably longer mean response times in the baseline IAT). How do I report this in a paper? Should I plot the fixed effects from the model instead of mean response times (which I'd rather not)?