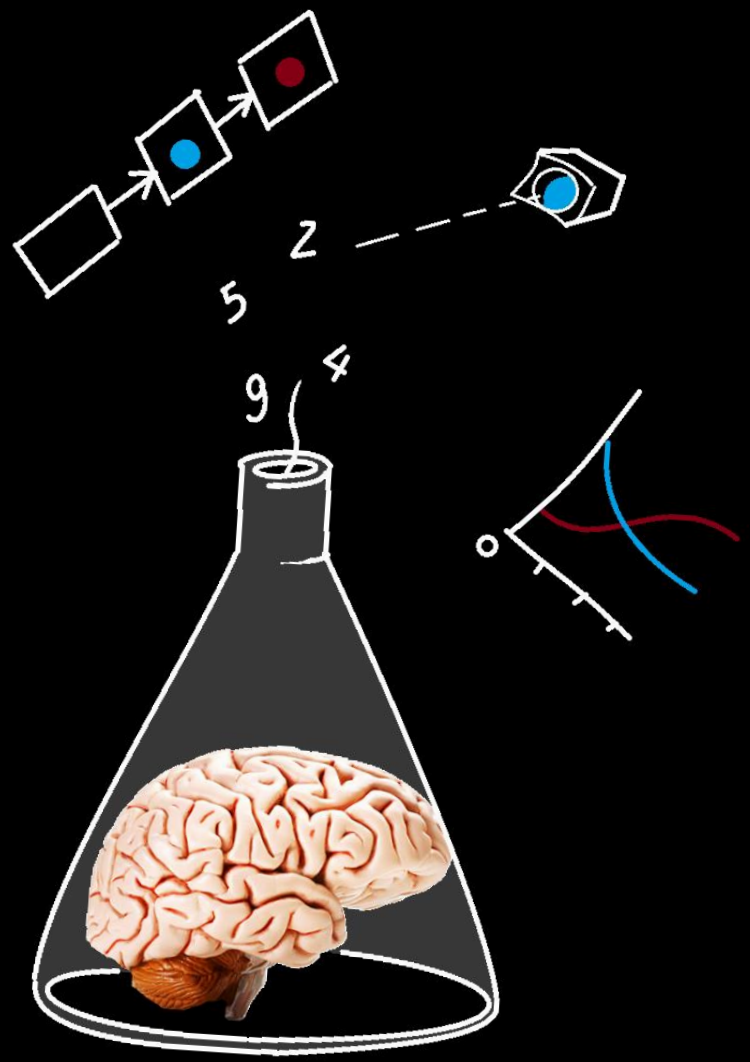
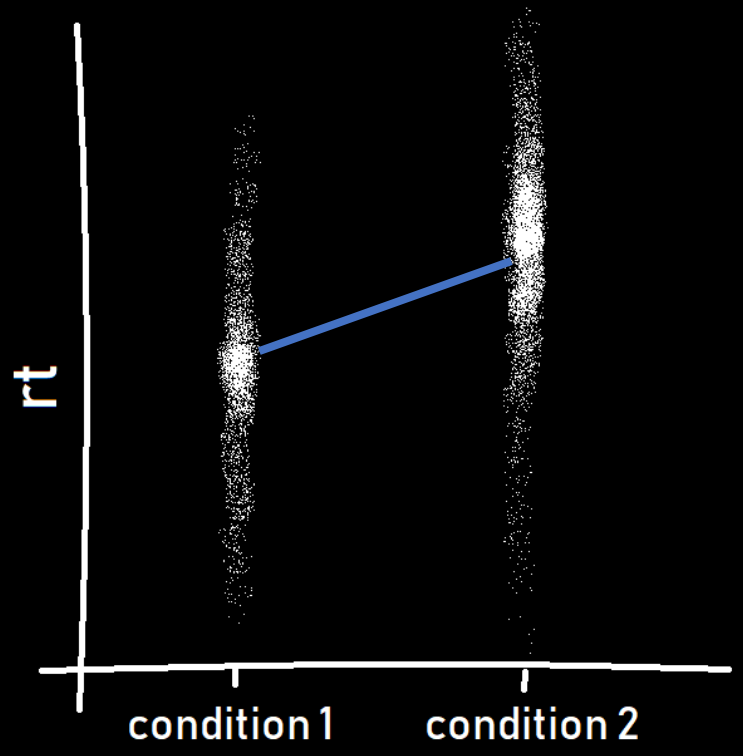
28-09

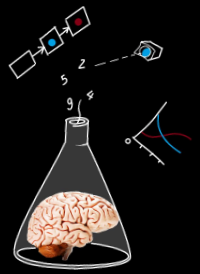
Linear mixed-effect models in *R*



The logic of LMMs

Explaining more variance than regular ANOVA's





The logic of LMMs

Fixed effects vs. random effects

Experimental variable (conditions)

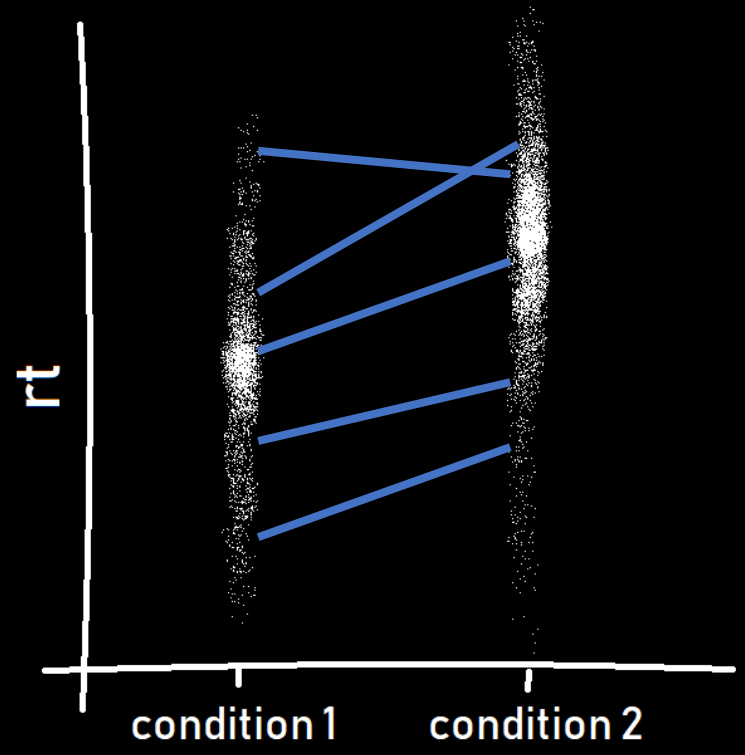
Those things about which we have hypotheses; (a specific direction of effect)

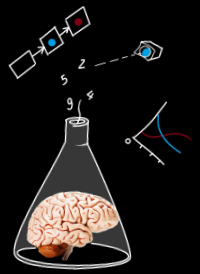
Covariates for which we expect a particular pattern

subjects
items (stimuli)

Those things that we expect may be variable, but for which we do not expect a particular pattern

Both in terms of *intercept* and *slope*, i.e. overall performance and effect strength





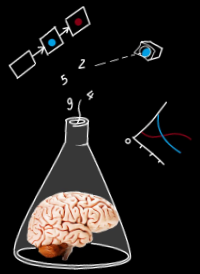
An example

Cats, Dogs & Capybaras

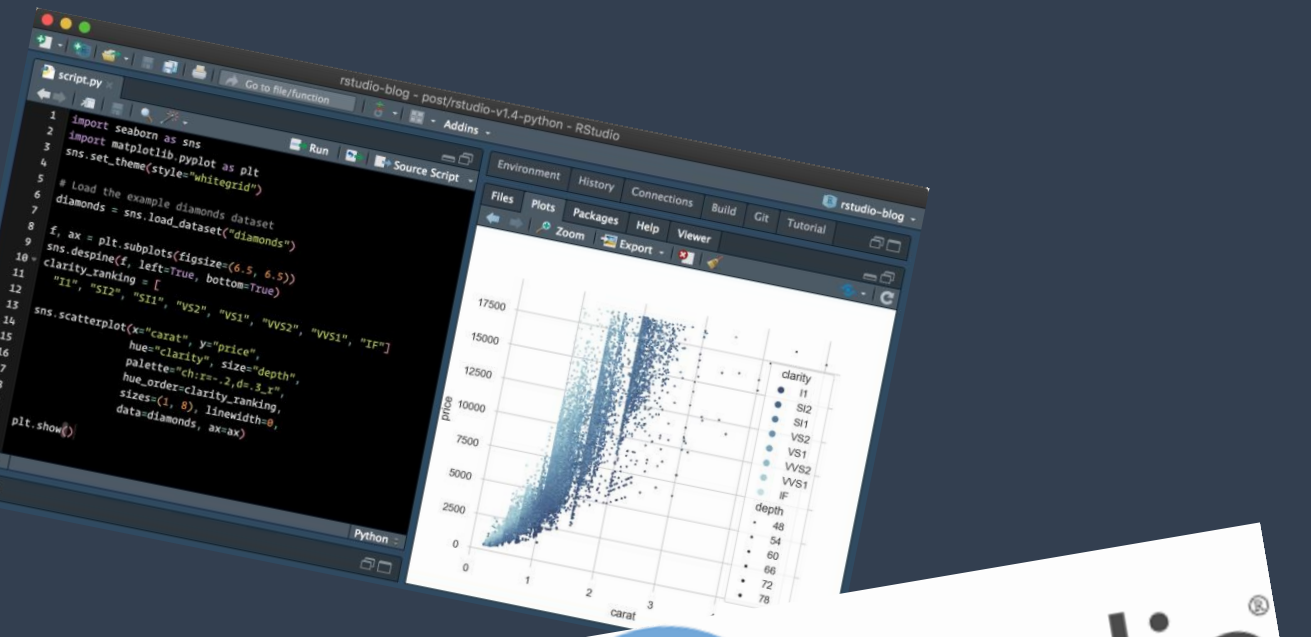
Fixed effect: picture-sound congruency (*congruent* vs. *incongruent*)

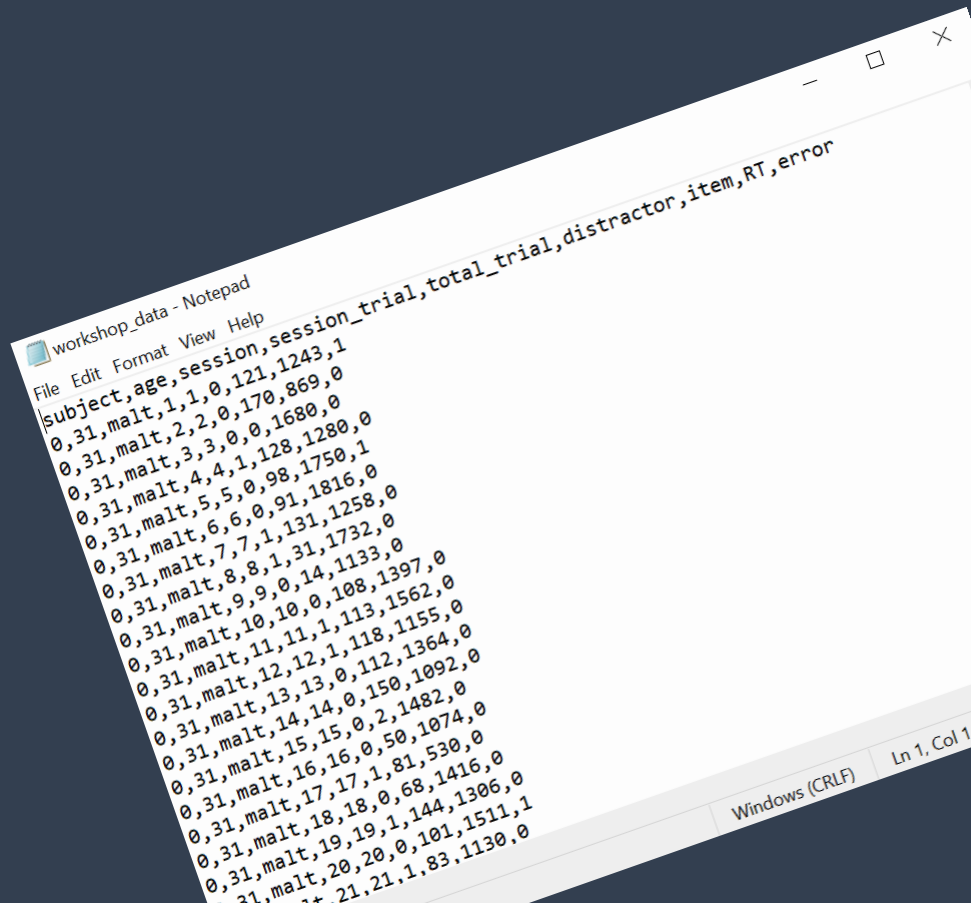
Random effect: subjects, animal type, animal picture, sound
(both in terms of intercepts as well as slopes)

→ The data will be analyzed on the basis of single trials, rather than averages per condition and participant

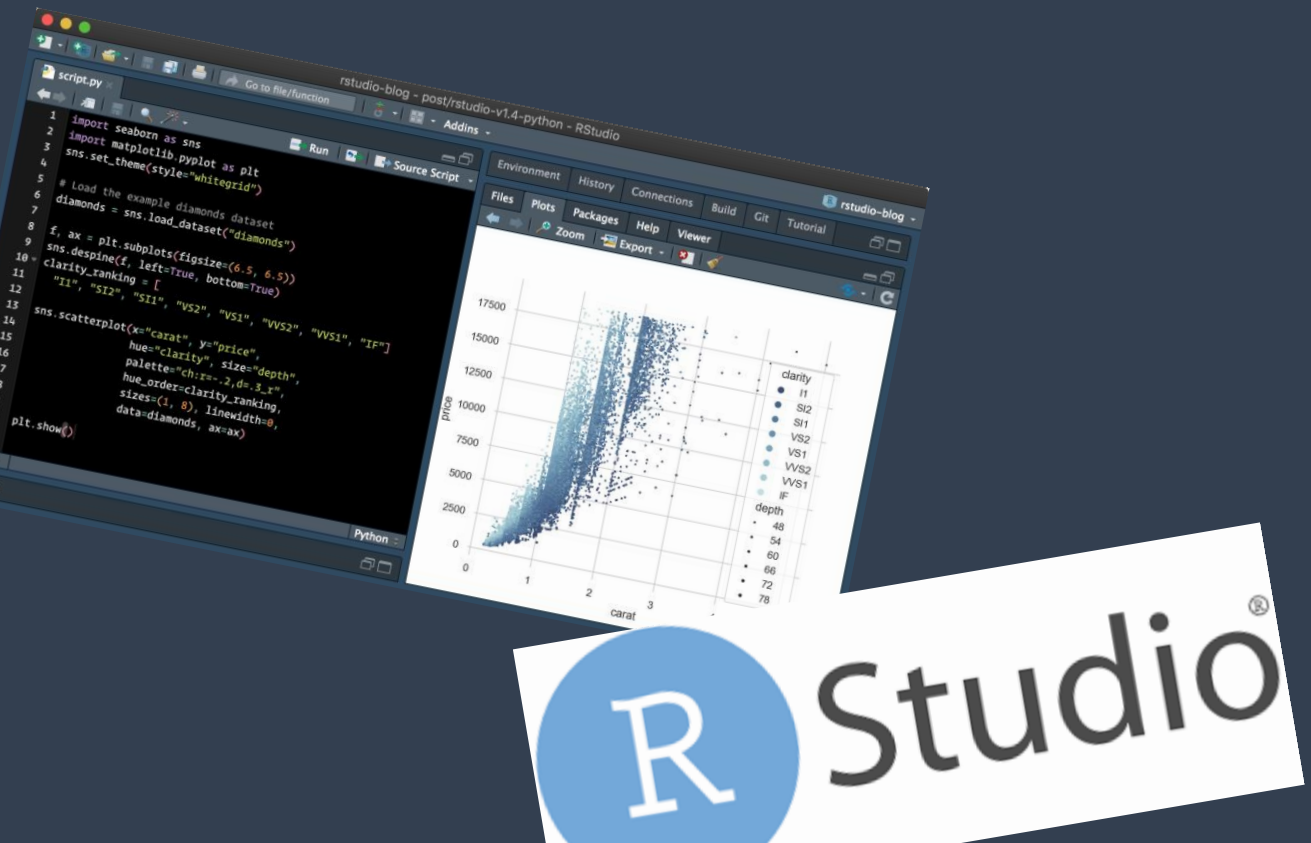


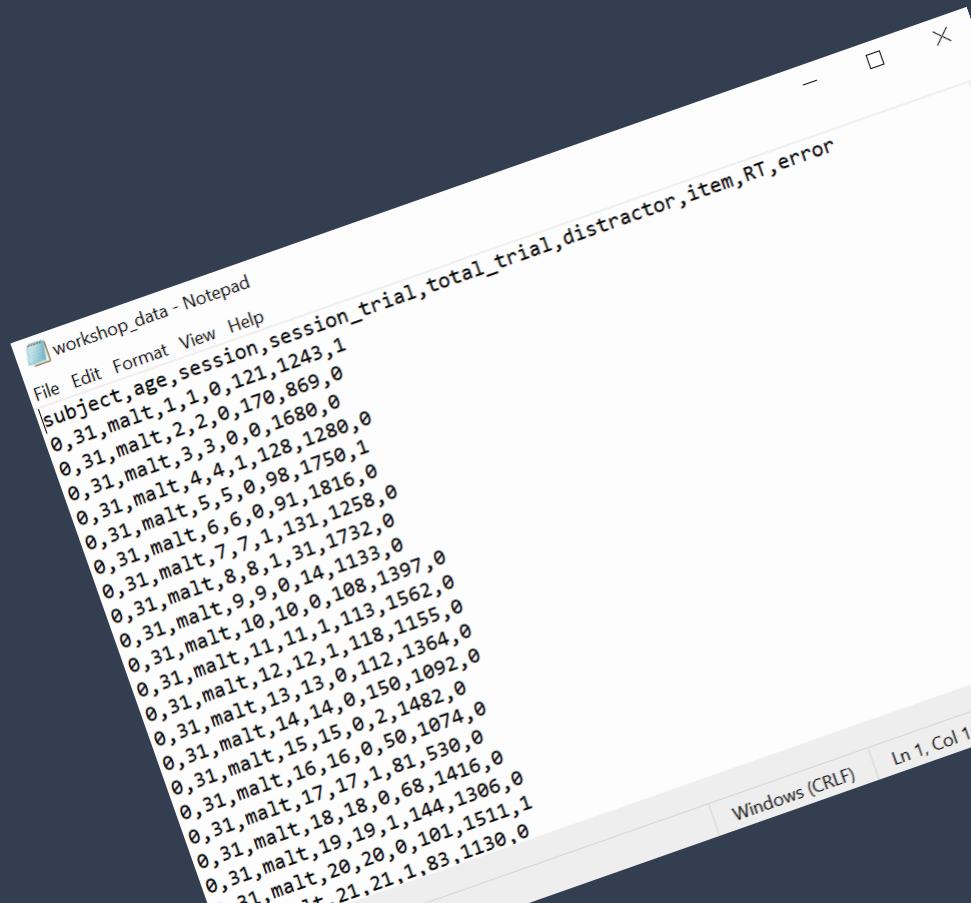
Linear mixed models: The tutorial





Place ‘workshop_data.txt’ in a folder of your preference
Do the same for LMM_script.R





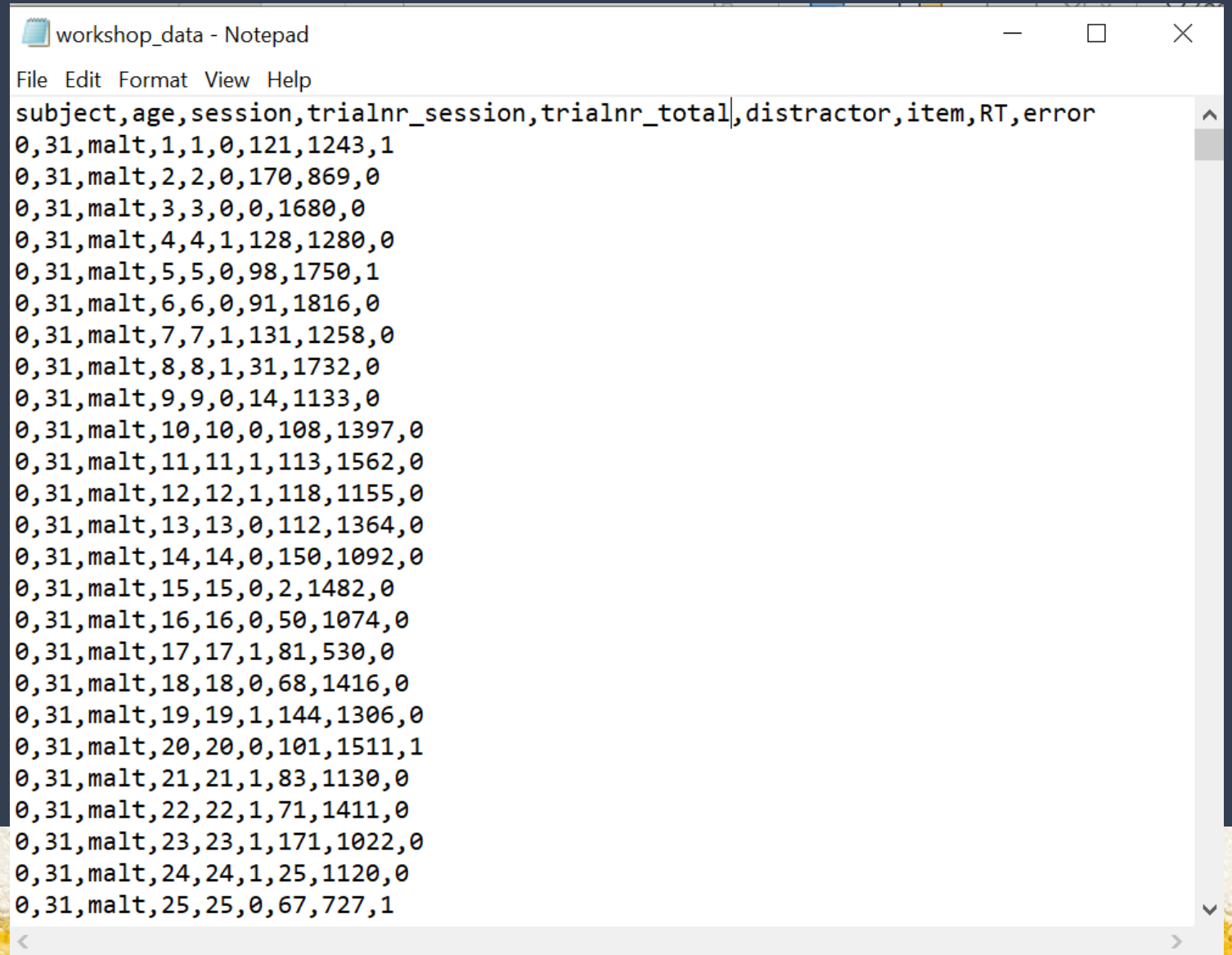
A hypothetical study of the influence of alcohol on cognition

- We test people's ability to focus; how strongly are they distracted in a speeded reaction task?
- 40 participants are tested in 3 sessions each: one after drinking a litre of the finest beer, one after drinking a litre of malt, and one after drinking a litre of lemonade
- Per session they do 400 trials: 200 without being distracted, 200 with distractor
- We don't want too many stimulus repetitions, so we have 200 different stimuli, each of which is being tested once per condition per participant.

→ A 2 x 3 design, with distraction *(present vs. absent)* and alcohol consumption *(alcohol, placebo, control)* as factors

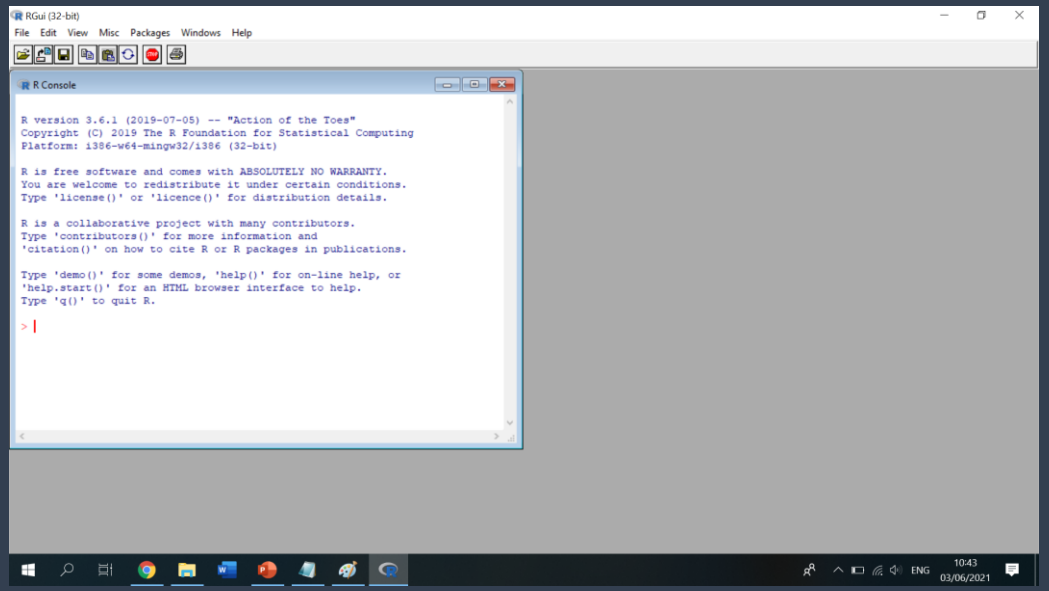
*…*and a few other things that might be contributing variance

Let's take a look at our data



R accepts various file formats
This txt file is interpreted as a csv file.
First row indicates column names (separated by comma)
All subsequent rows are individual trials.

And now: the script!



To install LMM functionality: type *install.packages("lme4")* and press Enter

RGui (32-bit)

File Edit Packages Windows Help

R Console

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for
'help.start()' for an HTML browser interfa
Type 'q()' to quit R.

>
```

C:\Users\Josh\Desktop\Josh work\Courses\Gastcolleges\LMM workshop\LMM_script.R - R Editor

```
library(lme4) # This is the package that we need for using LMMs


# First, specify the path to the datafile:
setwd('C://Users//Josh//Desktop//Josh work//Courses//Gastcolleges//LMM workshop')
data <- read.csv('workshop_data.txt')


# For the analysis of RTs, we want to exclude incorrectly answered trials:
data <- data[data$error == 0,]

# And we want to remove outliers:
outlier1 <- mean(data$RT) + 2.5 * sd(data$RT)
outlier2 <- mean(data$RT) - 2.5 * sd(data$RT)



                                                    as such. In the matrix, our IV
                                                    distraction and distraction condition
                                                    d "yes", respectively:
labels=c("no","yes"))

# We can choose reference levels for our factors of interest.
# We can then interpret all effects as differences from the reference

data$distractor <- relevel(data$distractor, ref="no")
```

R syntax:
"A <- operation"

ENG 10:58 03/06/2021

RGui (32-bit)

File Edit Packages Windows Help

R Console

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for
'help.start()' for an HTML browser interfa
Type 'q()' to quit R.

>
```

C:\Users\Josh\Desktop\Josh work\Courses\Gastcolleges\LMM workshop\LMM_script.R - R Editor

```
library(lme4) # This is the package that we need for using LMMs


# First, specify the path to the datafile:
setwd('C://Users//Josh//Desktop//Josh work//Courses//Gastcolleges//LMM workshop')
data <- read.csv('workshop_data.txt')


# For the analysis of RTs, we want to exclude incorrectly answered trials:
data <- data[data$error == 0,]

# And we want to remove outliers:
outlier1 <- mean(data$RT) + 2.5 * sd(data$RT)
outlier2 <- mean(data$RT) - 2.5 * sd(data$RT)
```

as such. In the matrix, our IV
on and distraction condition
respectively:

```
# We can choose reference levels for our factors of interest.
# We can then interpret all effects as differences from the reference
```

R syntax:

data <- load in data

Henceforth "data$subject" refers to first column of data

ENG 10:58 03/06/2021

RGui (32-bit)

File Edit Packages Windows Help

R Console

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' f
'help.start()'
Type 'q()' to q

>
```

C:\Users\Josh\Desktop\Josh work\Courses\Gastcolleges\LMM workshop\LMM_script.R - R Editor

```
library(lme4) # This is the package that we need for using LMMs


# First, specify the path to the datafile:
setwd('C://Users//Josh//Desktop//Josh work//Courses//Gastcolleges//LMM workshop')
data <- read.csv('workshop_data.txt')


# For the analysis of RTs, we want to exclude incorrectly answered trials:
data <- data[data$error == 0,]

# And we want to remove outliers:
outlier1 <- mean(data$RT) + 2.5 * sd(data$RT)
outlier2 <- mean(data$RT) - 2.5 * sd(data$RT)
```

"data <- data[data$column == something,]"

data is filtered down to data for which the value in column is something

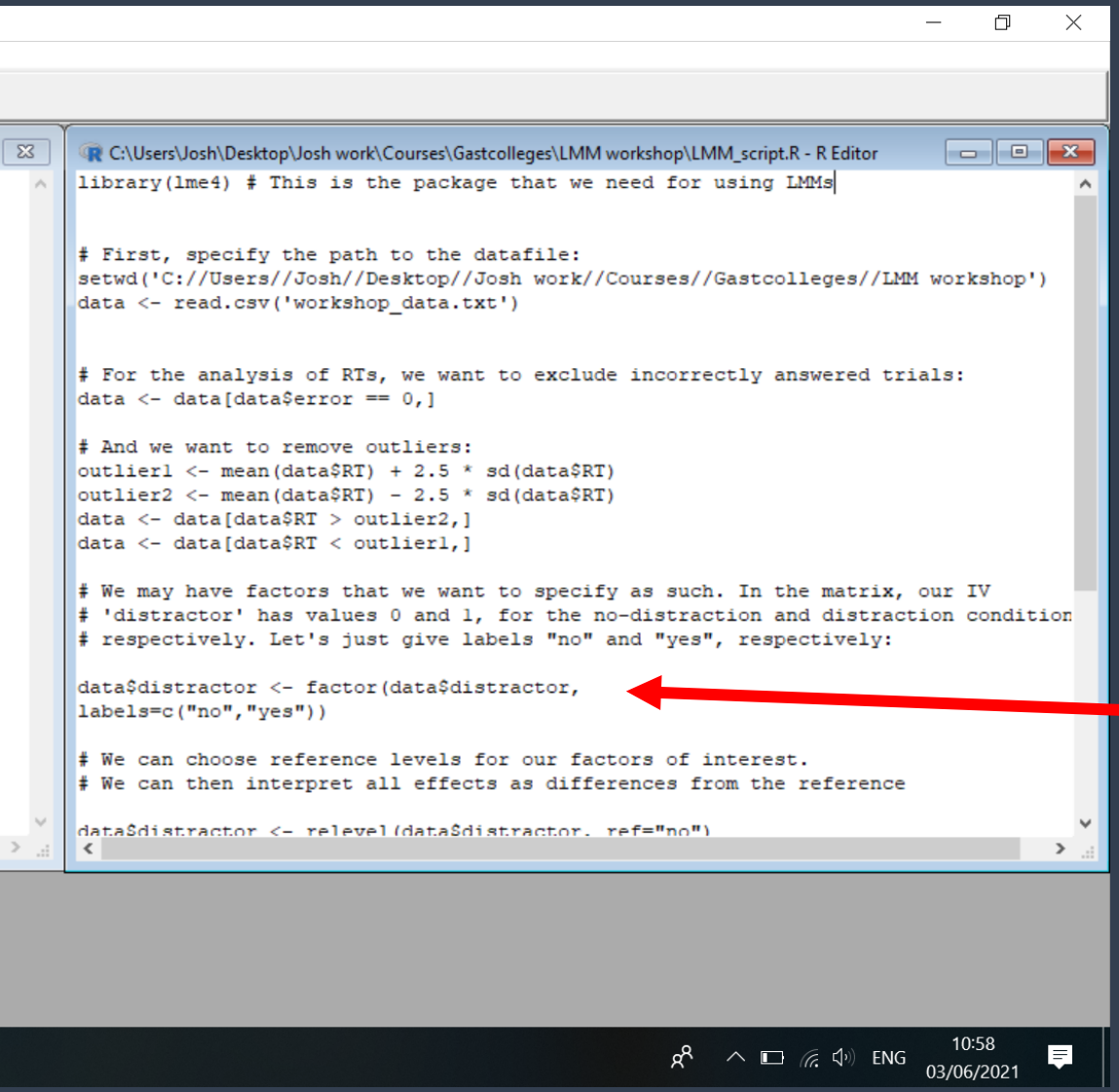
```
. In the matrix, our IV
ction and distraction condition
', respectively:

labels=c("no", "yes"))

# We can choose reference levels for our factors of interest.
# We can then interpret all effects as differences from the reference

data$distractor <- relevel(data$distractor, ref="no")
```

ENG 10:58 03/06/2021



R doesn't always know whether your IV is a continuous variable or a factor (if condition names are numbers rather than names)

specify that "distractor" is a factor

C:\Users\Josh\Desktop\Josh work\Courses\Gastcolleges\LMM workshop\LMM_script.R - R Editor

```
data$distractor <- factor(data$distractor,
labels=c("no","yes"))

# We can choose reference levels for our factors of interest.
# We can then interpret all effects as differences from the reference

data$distractor <- relevel(data$distractor, ref="no")
data$session <- relevel(data$session, ref="control")

# Here is the code for running a very simple LMM with distractor as fixed effect,
# and by-subject and by-item intercepts as random effect

model1 <- lmer(
    RT ~ distractor + (1|subject)+(1|item),
    data=data,)


# To read out the results, request a summary of the model
summary(model1)
```

The analysis is going to show the extent to which DV's in one condition differ from another. You can choose your reference conditions

C:\Users\Josh\Desktop\Josh work\Courses\Gastcolleges\LMM workshop\LMM_script.R - R Editor

```
data$distractor <- factor(data$distractor,
labels=c("no","yes"))

# We can choose reference levels for our factors of interest.
# We can then interpret all effects as differences from the reference

data$distractor <- relevel(data$distractor, ref="no")
data$session <- relevel(data$session, ref="control")

# Here is the code for running a very simple LMM with distractor as fixed effect,
# and by-subject and by-item intercepts as random effect

model1 <- lmer(
    RT ~ distractor + (1|subject)+(1|item),
    data=data,)


# To read out the results, request a summary of the model
summary(model1)
```

And finally, the actual analysis…

Printing the outcome (in the left window)

```
model <- lmer(DV ~ IV+(random effect1)+(random effect2),data=data)
```

Let’s run our script:

Select all (cntrl + a
or ‘apple’ + a)

Then cntrl + r
or ‘apple’ + r

R Console

```
Linear mixed model fit by REML ['lmerMod']
Formula: RT ~ distractor + (1 | subject) + (1 | item)
   Data: data

REML criterion at convergence: 559103.1

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.04904 -0.68993 -0.00143  0.68488  3.06371

Random effects:
 Groups   Name        Variance Std.Dev.
 item     (Intercept)    123.7  11.12
 subject  (Intercept)   8362.0  91.44
 Residual              135272.7 367.79
Number of obs: 38144, groups:  item, 200; subject, 40

Fixed effects:
             Estimate Std. Error t value
(Intercept)  1166.054     14.721   79.21
distractoryes   63.701      3.768   16.91

Correlation of Fixed Effects:
            (Intr)
distractrys -0.127
```

b-value = 63.70
SE = 3.77
t = 16.91

t = |1.96| ≈ *p* = .05

Intercept can be left alone (it's the ref. average, sign. different from zero)

R Console

```
Linear mixed model fit by REML ['lmerMod']
Formula: RT ~ distractor + (1 | subject) + (1 | item)
   Data: data

REML criterion at convergence: 559103.1

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.04904 -0.68993 -0.00143  0.68488  3.06371

Random effects:
 Groups   Name        Variance Std.Dev.
 item     (Intercept)    123.7  11.12
 subject  (Intercept)   8362.0  91.44
 Residual              135272.7 367.79
Number of obs: 38144, groups:  item, 200; subject, 40

Fixed effects:
             Estimate Std. Error t value
(Intercept)  1166.054     14.721   79.21
distractoryes   63.701      3.768   16.91

Correlation of Fixed Effects:
            (Intr)
distractrys -0.127
```

Random effects

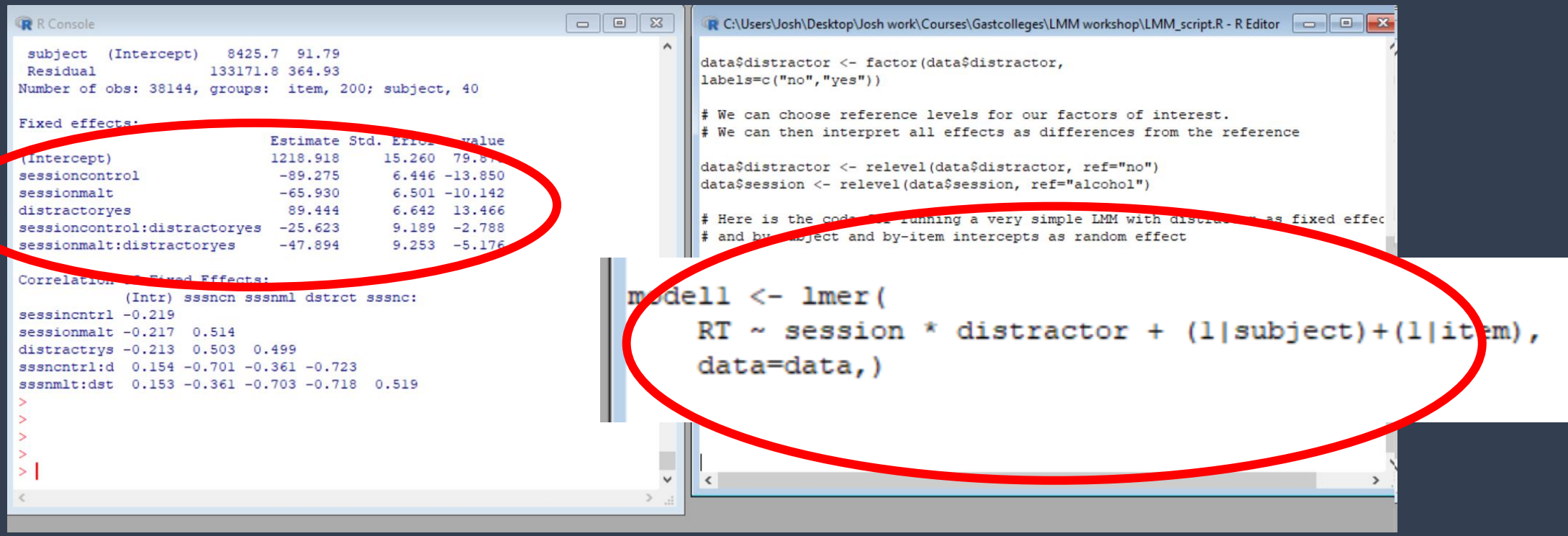
A small bit of variance was explained by differences across stimuli (items);
much more variance was explained by inter-subject differences.
But the residual – i.e., variance not accounted for by these random effects, was even larger.

Now you try:

- Test for a main effect of *session*
- Test for an effect of *session* on the error rate

  because error is a binomial variable, after "data=data," we add
  family="binomial"
  and instead of lmer(), we use glmer()
  Don't forget to 'inactivate' the line of code that excludes incorrectly answered trials! (place a '#' in front of it).
- Test for the *interaction between session and distractor* on RTs. In the function, write session*distractor

About interactions…



The * operator tests interaction plus main effects, while the : operator tests only the interaction

Be careful with main effects! That main effect of distractor is a main effect of distractor *in the reference condition of session*

But now we want the perfect model!

(1+distraction | subject)

On the right of the vertical line: variable name

On the left: 1 = random intercept

(subjects may on average vary from one another)

‘distraction’ = random slope

(the effect of distraction may vary across subjects)

But now we want the perfect model!

What do you think – do we want random slopes for subjects in the model?
What about random slopes for items?

But now we want the perfect model!

There is no consensus about what is a perfect LMM

My advice: let the structure of your model be theory-driven

(Should we include day of the week? Nationality? Age?)

It's basically always reasonable to assume that effects may vary across subjects; so by default I'd include random slopes

We can verify that a model with random slopes explains more variance than a model without random slopes

Likelihood-ratio test

model1 <- lmer *(with random slope)*
model2 <- lmer *(without random slope)*

anova(model1 , model2)

```
model1 <- lmer(
    RT ~ session * distractor  + (1+distractor|subject)
    data=data,)

model2 <- lmer(
    RT ~ session * distractor + (1|subject)+(1|item),
    data=data,)

anova(model1,model2)
```

```
> anova(model1,model2)
refitting model(s) with ML (instead of REML)
Data: data
Models:
model2: RT ~ session * distractor + (1 | subject) + (1 | item)
model1: RT ~ session * distractor + (1 + distractor | subject) + (1 +
model1:     distractor | item)
       npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
model2    7 370113 370170 -185050   370099
model1   11 370121 370210 -185049   370099 0.3737  4     0.9846
```

The universe impacts our behavior in a virtually infinite number of ways
Include the parts of the universe that you feel are really relevant

How do you report your analyses?

- “We ran a linear mixed-effect model (LMM) with A and B as fixed effects and by-C and by-D random intercepts as well as random slopes.”
- “The maximal random structure that successfully converged was one that included C and D random intercepts and by-C random slopes for the distractor factor”

How do you report your analyses?

- “Values of $t > |1.96|$ were deemed significant”

How do you report your analyses?

- Report *b*, SE and *t*. Mind positivity/negativity of *b* and *t*, as this indicates direction of effect

- Example text will be shared on Canvas.