Analyzing Data with Free Software

Henrik Singmann

Albert-Ludwigs-Universität Freiburg

UNI
FREIBURG

- A brief Introduction to R
- A (very) brief recap of the statistical model.
- An example experiment
- Analyzing the data in R

What is R

- R is a statistical programming language as compared to a statistical analyses package
- No click-and-play but a prompt/shell: `>`
- R is a **functional** programming language
- In R there are 3 sorts of things:
  - `data`
  - functions
  - `<-`
- Usual workflow:

```
> data.N <- function.X(data = data.A, …)
```

Workflow in R

- R follows its own workflow that is based on functions and data (objects).

- A function is a „machine“ that performs certain operations on ist arguments and returns one object (called return **value**).

- the assignment operator `<-` links data and functions:
  `> data.N <- function.X(data = data.A, other.argument = x)`

R Basics

```
rm(list = ls())  # remove everything

a <- rnorm(20)  # assign a with 20 values

A  # gives Error, a and A are different

rnorm(10) # not assignment, just prints

?rnorm # what does the function do?
```

?rnorm

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

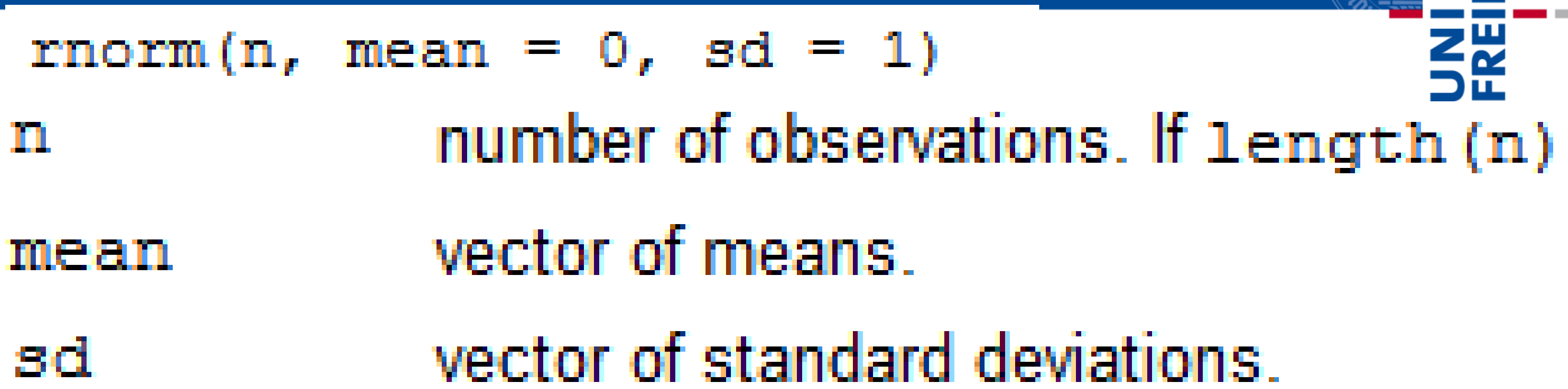
Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.

?rnorm



- `n`, `mean`, and `sd` are the arguments of `rnorm()`
- `mean` and `sd` have default values (`0` and `1`)
- `rnorm` needs to be called with at least a value for `n`
- Mapping of arguments is either via position or via name.

using arguments:

```
rnorm(n, mean = 0, sd = 1)
```

```
rnorm(5)
# identical calls (but other random data):
rnorm(n = 5)
rnorm(5, 0, 1)
rnorm(5, sd = 1, mean = 0)
rnorm(sd = 1, mean = 0, n = 5)
```

Important: The use of <-

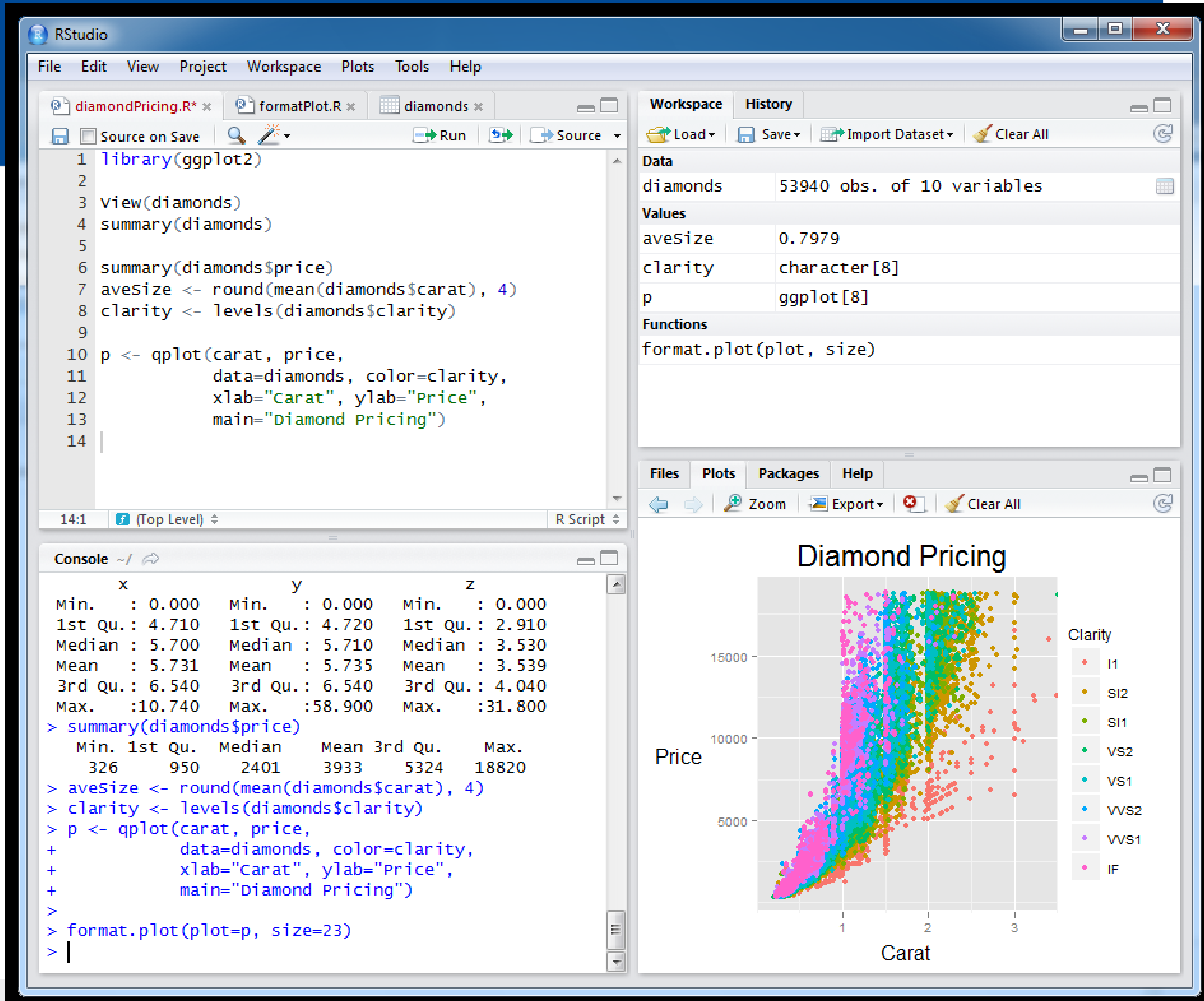
- When performing any (most) operations you can decide whether you want to print the result or save the result.

- If you use the assignment operator (<-) the result is saved to that object and not printed.
  (Note if you assign something to an existing object, that object is overwritten)

- If you don't use the assignment operator the result is just printed and not saved (i.e., it is lost for further use).

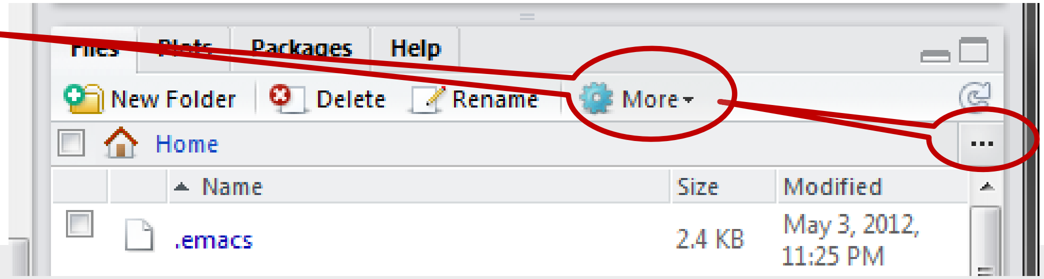
Using R

- When working with R you can have an arbitrary number of objects in your workspace.
- View your workspace with `ls()` or `ls.str()`
- Remove objects with `rm()` and all objects with `rm(list = ls())`
- All file operatins are relative to the current working directory. Find out what it is with `getwd()`, set it with `setwd()` or use the GUI of RStudio
- If you want to output an object to recreate it use `dput()`.
- If you see a + instead of the prompt symbol > hit ESC.


RStudio
File Edit View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save Run Source
library(ggplot2)
View(diamonds)
summary(diamonds)
summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)
p <- qplot(carat, price,
data=diamonds, color=clarity,
xlab="Carat", ylab="Price",
main="Diamond Pricing")
14:1 (Top Level) R Script
Workspace History
Load Save Import Dataset Clear All
Data
diamonds 53940 obs. of 10 variables
Values
aveSize 0.7979
clarity character[8]
p ggplot[8]
Functions
format.plot(plot, size)
Files Plots Packages Help
Zoom Export Clear All
Console ~/
x y z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean : 5.731 Mean : 5.735 Mean : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max. :10.740 Max. :58.900 Max. :31.800
> summary(diamonds$price)
Min. 1st Qu. Median Mean 3rd Qu. Max.
326 950 2401 3933 5324 18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+ data=diamonds, color=clarity,
+ xlab="Carat", ylab="Price",
+ main="Diamond Pricing")
>
> format.plot(plot=p, size=23)
>
Diamond Pricing
Price
15000
10000
5000
Carat
1
2
3
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

Using RStudio

- It is good practice to type in the script window and save the script as an .R file.
- Execute the current line or selection from the Script with CTRL+ENTER (Strg + Enter).
- The workspace is conveniently displayed and can be saved via the menu.
- The working dircetory can also be set via the menu conveniently



Data Types in R: Vectors

- Unidimensional data (i.e., a vector) is created by the `c()` function and accessed with `[ ]`:

```
> a <- c(2, 4, 65, 9)
 > a[3:4]
[1] 65  9

> b <- c("hans", "uli", "peter")
> b[1]
[1] "hans"
```

Data Types in R: data.frame

- Most important data type
- Two-dimensional: rows represent observations and columns variables
-
```
> e <- data.frame(a = LETTERS[1:5], b =
sample(letters, 5), c = rnorm(5))
> e
  a b          c
1 A r -0.6638516
2 B b -1.9774291
3 C m  1.2685798
4 D c  1.6817004
5 E s -0.1857508
```

Data Types in R: data.frame

- `data.frames` can be accessed with `[ , ]`:

```
> e[1:2,c("b", "c")]
  b          c
1 r -0.6638516
2 b -1.9774291
```

- `data.frames` can be accessed with $ (which always returns a column/vector):

```
> e$c
[1] -0.6638516 -1.9774291  1.2685798
1.6817004 -0.1857508
```

Data Type in R: Lists

- Lists can contain all other types of data (even lists):

```
> f <- list(el1 = e, el2 = rnorm(5))
> f
$el1
  a b          c
1 A r -0.6638516
2 B b -1.9774291
3 C m  1.2685798
4 D c  1.6817004
5 E s -0.1857508

$el2
[1]  2.3827040 -0.4565383 -0.5001515
1.4185623 -0.7036125
```

Data Type in R: Accessing Lists

- `[ ]` can select multiple elements and returns a list:
  `f[1:2]`

- `[[ ]] or $` selects only one element and returns the element:

```
> f[["el2"]]
> f$el2
> f[[2]]
[1]  2.3827040 -0.4565383 -0.5001515
1.4185623 -0.7036125
```

Resources for R

- A thorough introduction to R is beyond the scope of this talk.

- Introductory books:
  - Maindonald, J., & Braun, W. J. (2010). *Data Analysis and Graphics Using R: An Example-Based Approach* (3. Aufl.). Cambridge: Cambridge University Press.
  - Teetor, P. (2011). *R Cookbook* (1st ed.). Sebastopol CA: O'Reilly.
  - Matloff, N. (2011). *The Art of R Programming: A Tour of Statistical Software Design* (1. Aufl.). San Francisco: No Starch Press. (**For Programmers**)
  - Kabacoff, R. (2011). *R in action : data analysis and graphics with R*. Greenwich, Conn: Manning.
    (**From the author of the famous website: Quick-R**)

A very brief introduction to statistics

Multiple Regression / GLM

- General Linear Model: Basic linear statistical model
- One interval scaled response variable *y*
- *m* predictors:
  - numerical: age, scores, …
  - categorical: condition, treatment, gender, …
    (categorical variables with *n* levels are represented in *n*-1 predictors, using effects coding)
- $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + \varepsilon$
- Observations are independent

Extensions to the GLM

- The GLM can be extended for within-subject categorical predictors: repeated measures ANOVA

- Repeated measures ANOVA allows to generailze across units of obersvations (i.e., participants), but assumes sphericity across measurements.

- A mixed model or multilevel model overcomes this limitations and allows for generalizations across participants and items:
  - Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, *59*(4), 390–412. doi:10.1016/j.jml.2007.12.005
  - Judd, C. M., Westfall, J., & Kenny, D. A. (2012). Treating stimuli as a random factor in social psychology: A new and comprehensive solution to a pervasive but largely ignored problem. *Journal of Personality and Social Psychology*, *103*(1), 54–69. doi:10.1037/a0028347

Statistic Books with R

- Baayen, R. H. (2008). *Analyzing linguistic data: a practical introduction to statistics using R*. Cambridge, UK; New York: Cambridge University Press. (**cheap and deals with mixed models, I liked it a lot**)

- Baguley, T. (2012). *Serious stats : a guide to advanced statistics for the behavioral sciences*. Houndmills, Basingstoke, Hampshire; New York: Palgrave Macmillan. (**very readable, impressively big, and up-to-date**)

- Fox, J., & Weisberg, S. (2011). *An R Companion to Applied Regression*. Thousand Oaks, Calif.: SAGE Publications. (**very good for the standard GLM and GLS**)

- Gelman, A. B., & Hill, J. (2009). *Data analysis using regression and multilevel/hierarchical models*. Cambridge; New York: Cambridge University Press. (**the reference, mathematical**)

Analyzing Data

Outlook

- We will analyze data in a format as produced by PsychoPy/PsyTML.
- Data for each participant is in a single file and each item occupies one row
- This dataset consists of 10 participants from a reasoning experiment in which participants had to rate how much they liked the conclusions (i.e., last sentence) of presented syllogisms.

Example Syllogism

No friendly animals are elgs.

Some elgs are sharks.

Some sharks are not friendly.

- Sentences were presented sequentially.
- Participants had to indicate how much they liked the last sentence (i.e., conclusion) on a scale
  - from 1 ("Don't like it at all")
  - to 5 ("Like it very much")

Item types

- Each participant worked on 24 syllogisms.

- We manipulated the validity of the Syllogisms:
  - 12 Syllogisms were logically valid (i.e., conclusion follows necessary from the premises)
  - 12 Syllogisms were logically invalid

- We manipulated the believability of the Syllogisms:
  - 8 conclusions were believable (e.g., "Some sharks are not friendly.")
  - 8 conclusions were unbelievable (e.g., "Some millionaires are not rich")
  - 8 conclusione were abstract (e.g., "Some rups are not milk shakes.")

Design

- In addition to the two within-subjects factors validity and believability we had one between-subjects manipulation (`condition`):
  We manipulated whether the syllogisms were really valid (`logic`) or only appeared to be so (`fluency`).

- Design: Validity (2 levels, within-subjects) × Believability (3 levels, within-subjects) × Condition (2 levels, between-subjects)

- Hypotheses: People like valid syllogisms more than invalid ones, but in both conditions (cf. Morsanyi & Handley, 2012, JEP: LMC)

Analysis Strategy

- Run a so-called split-plot ANOVA or mixed model analysis.
- In SPSS data would need to be transformed, only 1 row per participant with aggregated means per cell. (This is the so-called wide or broad format)
- In R we can usually leave the data in the long format: One obervation per row.

Steps in the code

- Read in the data
- Preprocess the data
- Run the analysis
- Plot the data
- Run Post-Hoc tests / contrasts.

afex

- `afex` is an R package for the analysis of factorial experiments allowing to compute ANOVA and ANCOVA for data in the long format.

- The functions are:
  - `ez.glm` – ANOVA and ANCOVA, similar to SPSS glm
  - `aov.car` – ANOVA and ANCOVA usind a formula interface
  - `univ` – returns univariate instead of multivariate tests (formerly `univariate`)
  - `nice.anova` – produces a nice ANOVA table
  - `mixed` – allows for the analysis using linear mixed models (i.e., multiple rdnom effects, or multilevel models)

- See: http://www.psychologie.uni-freiburg.de/Members/singmann/R/afex

afex

- Note: When running an ANOVA with `afex`, `afex` aggregates the data if necessary before running the analysis!

- You can even choose which aggregation function to use. The default is `mean()`.

- A package similar to afex is `ez` (but it is not written by me), and Type 3 sums of squares are not the default in ez.

- To use afex, load it:

```
require(afex)  #  or
library(afex)
```

nice.anova

```
                         Effect          df  MSE    F   p
1                          cond        1, 8 0.25 1.35 .28
2                      validity        1, 8 0.16 3.09 .12
3                 cond:validity        1, 8 0.16 3.09 .12
4                 believability  1.36, 10.9 1.73 1.67 .23
5            cond:believability  1.36, 10.9 1.73 0.99 .37
6        validity:believability 1.52, 12.17 0.54 2.92 .10
7 cond:validity:believability 1.52, 12.17 0.54 1.53 .25

Warning message:
In aov.car(formula = as.formula(formula), data = data,
   fun.aggregate = fun.aggregate,  :
  More than one observation per cell, aggregating the
   data using mean (i.e, fun.aggregate = mean)!
```

Formulas in R

- Instead of using character vectors to specify the factors in the design, `afex` allows using a formula.

- Formulas are a very basic concept in R and useful for all types of statistical models, as they closely correspon dto the mathematical definition of the model.

- Formulas usually have a right-hand side (What do I want to predict?), the formula operator ~, and a left hand side (What are the predictors?), e.g.:
  `y ~ x1 + x2 + x3`

- `+` denotes a main effect, `:` denots an interaction, and `*` denotes main effect and interactions:
  `y ~ (x1 + x2) * x3`

aov.car

- `aov.car(resp ~ cond + Error(id/validity * believability), rf)`

- The dv needs to be specified on the left hand side, the between subjects factors on the right hand side and the within-subject factors and the id-variable inside the Error term.

- `aov.car` uses an interface similar to the base R function `aov`, but can handle unbalanced design (which `aov` can't handle).

Other formula functions

- `lm()` is the main R function for (multiple) regression
- `glm()` is the function for generalized linear models (e.g., logistic or poisson regression)
- `t.test()` or `cor.test()` can also be called using a formula.

Linear Mixed Models

- Linear Mixed Models are a modern form of regression-type like models and can be used if there are multiple random effects or hierarchical or multilevel structures in the data.

- Linear Mixed Models in R are best calculated using package `lme4` (the predecessor package `nlme` can also be used in certain situations, but is not discussed here).

- `afex` contains the convenience function `mixed` to obtain p-values for mixed models and fits them with lme4.

mixed()

- Whereas in the classical analysis participants are treated as a random effect, one could also treat the syllogisms as a random effect.
- This can be done using mixed models + the data does not need to be aggregated (we use all data directly):

```
mixed(resp ~ cond * validity *
believability + (1+ (validity *
believability)|id) + (1|nr), rf)
```

- Note: Running `mixed()` takes some time.

mixed()

	Effect	df1	df2	Fstat	p.value
1	(Intercept)	1	5.3554	1968.8677	0.0000
2	cond	1	8.0000	1.0759	0.3299
3	validity	1	5.3554	1.6072	0.2572
4	believability	2	6.0809	2.8359	0.1349
5	cond:validity	1	8.0000	1.6072	0.2405
6	cond:believability	2	7.0000	0.7688	0.4991
7	validity:believability	2	5.4296	1.7091	0.2656
8	cond:validity:believability	2	7.0000	1.1472	0.3707

Post-Hoc Contrasts

- Post-hoc contrasts for repeated-measures ANOVA models can be done using package `phia`, functions `testInteractions` or `testFactors`.

- For all other tests, use package `multcomp`, function `glht`:
  Frank Bretz, Torsten Hothorn and Peter Westfall (2010), *Multiple Comparisons Using R*, CRC Press, Boca Raton.

Summary

- With R you can do complicated analysis with little rows.
- For the "usual" data, the usual analysis strategies are applicable: GLM
- If your data has some replications for each within-subjects cell, consider using a mixed models approach (a random effect should have at least 6 factor levels)