afex – Analysis of Factorial EXperiments

Henrik Singmann

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

BaselR Meeting March 2013

afex - overview

- R package for the convenient analysis of factorial experiments

- Main functionality:
  - works with data in the long format (i.e., one observation per row)
  - ANOVA specification: `aov.car()`, `ez.glm()`, and `nice.anova()`
  - Obtain p-values for generalized and linear mixed models (GLMMs and LMMs): `mixed()`

  - Compare two vectors using different statistical tests: `compare.2.vectors()`

- `afex` tries to imitate commercial statistical packages by using effect / contrast coding (i.e., sum to zero coding) and uses type 3 sums of squares.

ANOVA

- ANOVA is a general linear model (GLM) with only categorical predictors.
- One interval scaled response variable *y*
- *m* predictors (*β)*:
  - condition, treatment, gender, …
    (categorical variables with *n* levels are represented in *n*-1 predictors, usually using effects coding)
- $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m + \varepsilon$,
  where $\varepsilon \sim N(0, \sigma^2)$
- Observations are independent

R and ANOVA

UNI FREIBURG

The standard analysis of variance (ANOVA) is a somewhat neglected statistical procedure in (base) R:

- "Although the methods encoded in procedures available in SAS and SPSS can seem somewhat oldfashioned, they do have some added value relative to analysis by mixed model methodology, and they have a strong tradition in several applied areas." (Dalgaard, 2007, p. 2, R News)

ANOVA in Base R: `aov()`

UNI FREIBURG

- Only for balanced designs (from `?aov`):
  `aov` is designed for balanced designs, and the results can be hard to interpret without balance: […]. If there are two or more error strata, the methods used are statistically inefficient without balance, and it may be better to use `lme` in package `nlme`.

- Basically only supports "type 2" sums of squares

- Can be very cumbersome for within-subject factors (e.g., http://stats.stackexchange.com/q/6865/442)

Default coding in R

- When predictors are categorical (as is the case for ANOVA), they need to be transformed in (k – 1) numerical predictor variables using a coding scheme.

- The default coding in R is treatment coding (= the intercept corresponds to the mean of the first group):

```
> options("contrasts")
$contrasts
        unordered           ordered
"contr.treatment"      "contr.poly"
```

  - Treatment coding has the unfortunate downside that main effects are simple effects when interactions are present (i.e., effects of one variable when the other is 0).

- The usual coding for ANOVA is effects coding or sum-to-zero coding (main effects are interpretable in light of interactions):

```
> options("contrasts")
$contrasts
[1] "contr.sum"  "contr.poly"
```

Alternatives to `aov()`

- `car::Anova()` from John Fox
  - can handle any number of between- and within-subjects factors
  - allows for so called "type 2" and "type 3" sums of squares.
  - relatively uncomfortable for within-subject factors, as data needs to be in wide format (i.e., one participant per row)

- `ez` (by Mike Lawrence) provides a wrapper for `car::Anova()`, `ezANOVA()`

- `afex` is another `car` wrapper:
  - `aov.car()` provides an `aov()` like formula interface
  - `ez.glm()` specification of factors using character vectors
  - `afex` automatically sets default contrasts to `contr.sum` (i.e., sum-to-zero or deviation coding)

Example data

- Reasoning experiment with 60 participants:
  - Participants had to rate 24 syllogisms (Klauer & Singmann, in press, JEP:LMC, Experiment 3)
- 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 (cf. Morsanyi & Handley, 2012, JEP: LMC)

the data (in d)

```
> str(d)
'data.frame':   1440 obs. of  6 variables:
 $ id           : Factor w/ 60 levels "1","2","3","4",..: 1 1 1 ...
 $ cond         : Factor w/ 2 levels "fixed","random": 1 1 ...
 $ validity     : Factor w/ 2 levels "invalid","valid": 2 2 1 1 ...
 $ believability: Factor w/ 3 levels "abstract","believable",..: ...
 $ resp         : int  4 5 3 4 4 3 4 2 3 5 ...

> xtabs( ~ believability + validity + id, data = d)
, , id = 1
              validity
believability  invalid valid
  abstract           4     4
  believable         4     4
  unbelievable       4     4
[...]
```

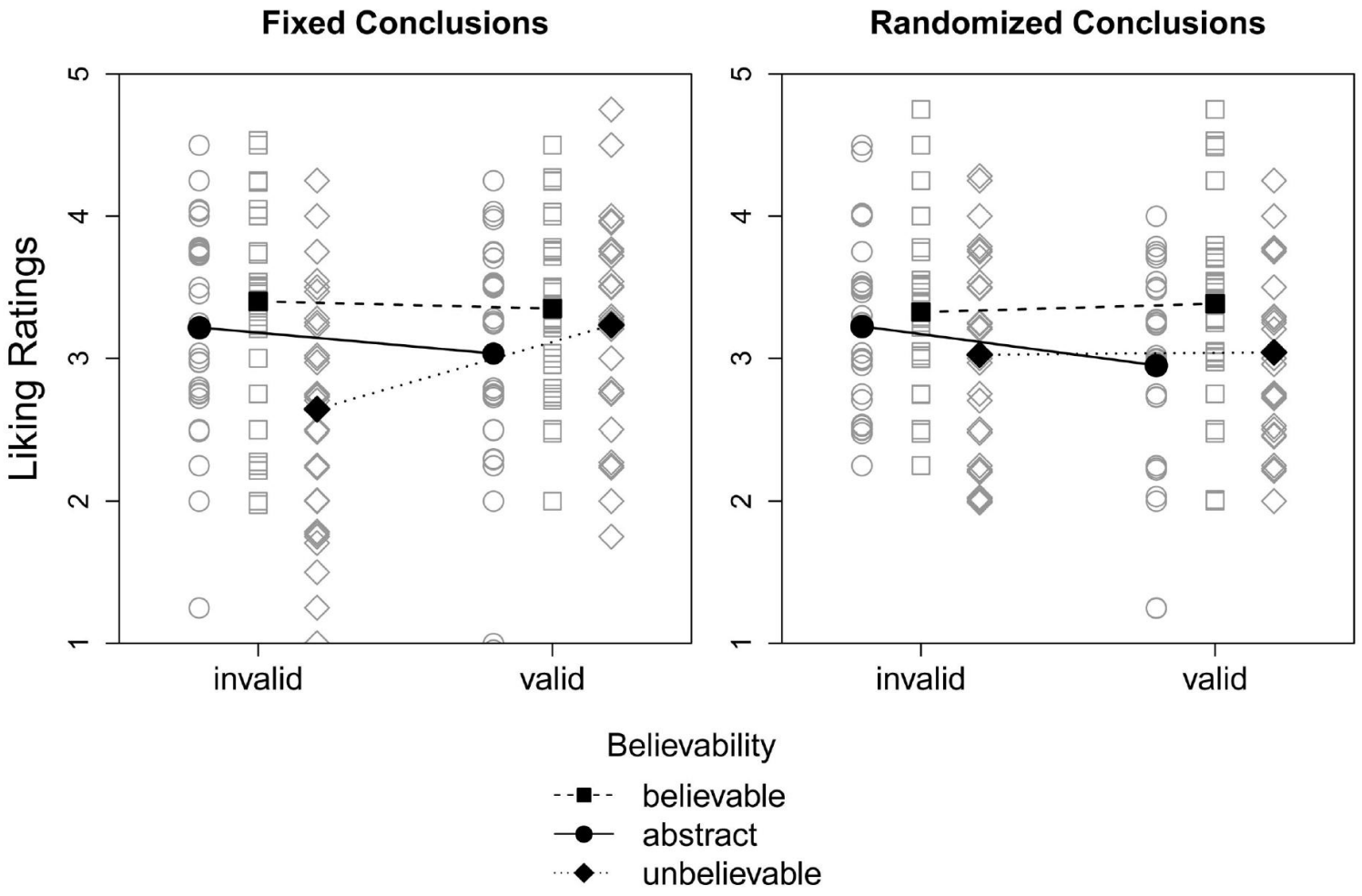


Figure 3. Mean (filled symbols) and individual (nonfilled symbols) liking ratings in Experiment 3 for the group with fixed contents (left panel) and the group with randomized contents (right panel) as a function of validity/pseudo-validity and conclusion believability. A small amount of vertical jitter was added to individual liking ratings to avoid perfect overlap of two ratings.

Graph produced with my `raw.means.plot2()` function (`plotrix` package, Lemon, 2006, R NEWS).

ANOVA in afex

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

Some differences to `aov()`:

- Error term is mandatory (to specify the id variable).
- within-subject factors only need to be present in the Error term (but can be present outside of it, where they will be ignored).
- within-subject factors don't need to be enclosed in brackets and are always fully crossed

```
ez.glm("id", "resp", d, between = "cond",
       within = c("believability", "validity"))
```

- Calls `aov.car()` with the respective formula

ANOVA in afex

```
ez.glm("id", "resp", d, between = "cond",
       within = c("believability", "validity"))
```

	Effect	df	MSE	F	ges	p
1	cond	1, 58	0.94	0.01	<.001	.90
2	believability	1.84, 106.78	0.59	8.36 ***	.05	<.001
3	cond:believability	1.84, 106.78	0.59	0.29	.002	.73
4	validity	1, 58	0.38	0.17	<.001	.68
5	cond:validity	1, 58	0.38	2.07	.005	.16
6	believability:validity	1.85, 107.52	0.28	8.29 ***	.02	<.001
7	cond:believability:validity	1.85, 107.52	0.28	3.58 *	.01	.03

```
Warning message:
In aov.car(resp ~ cond + Error(id/(believability * validity)), d) :
  More than one observation per cell, aggregating the data using
mean (i.e, fun.aggregate = mean)!
```

ANOVA in afex

Default output contains the "recommended effect size for repeated-measures design" (Bakeman, 2005, Behavior Research Methods), η^2_G

```
                      Effect          df  MSE         F   ges     p
1                       cond        1, 58 0.94      0.01 <.001   .90
2              believability 1.84, 106.78 0.59 8.36 ***   .05 <.001
3         cond:believability 1.84, 106.78 0.59      0.29  .002   .73
4                   validity        1, 58 0.38      0.17 <.001   .68
5              cond:validity        1, 58 0.38      2.07  .005   .16
6       believability:validity 1.85, 107.52 0.28 8.29 ***   .02 <.001
7 cond:believability:validity 1.85, 107.52 0.28   3.58 *   .01   .03

Warning message:
In aov.car(resp ~ cond + Error(id/(believability * validity)), d) :
  More than one observation per cell, aggregating the data using
mean (i.e, fun.aggregate = mean)!
```

ANOVA in afex

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

	Effect	df	MSE	F	ges	p
1	cond	1, 58	0.94	0.01	<.001	.90
2	believability	1.84, 106.78	0.59	8.36 ***	.05	<.001
3	cond:believability	1.84, 106.78	0.59	0.29	.002	.73
4	validity	1, 58	0.38	0.17	<.001	.68
5	cond:validity	1, 58	0.38	2.07	.005	.16
6	believability:validity	1.85, 107.52	0.28	8.29 ***	.02	<.001
7	cond:believability:validity	1.85, 107.52	0.28	3.58 *	.01	.03

```
Warning message:
In aov.car(resp ~ cond + Error(id/(believability * validity)), d) :
  More than one observation per cell, aggregating the data using
mean (i.e, fun.aggregate = mean)!
```

`aov.car()` automatically aggregates data for the within-subject factors (with warning). Warning can be suppressed by explicitly specifying the aggregation function.

ANOVA with afex

- `aov.car()` and `ez.glm()` per default return a print-ready ANOVA table, produced by `nice.anova()` (i.e., `return = "nice"`)
  - Specify df-correction: Greenhouse-Geisser (default), Huynh-Feldt, none
  - Specify effect size: η^2_G (default) or η^2_P

- `return` argument can be changed to return:
  - `"Anova"`: the S3 object returned from `car::Anova()`
  - `"univariate"`: A list of univariate ANOVA tables
  - `"lm"`: The `lm` object passed to `car::Anova()`
  - `"data"`: The transformed (to wide) data.
  - `"full"`: All of the above (except the univariate part), which can be conveniently passed to e.g., the `phia` (de Rosario-Martinez, 2012) package for (multivariate) post-hoc tests (contrasts)

Beyond ANOVA: mixed models

UNI FREIBURG

- Repeated-measures ANOVA has limitations (e.g., Keselman, et al., 2001, BJS&MP):
  - Sphericity assumption: df correction known to be problematic
  - Only one observation per cell of the design allowed
  - No simultaneous analysis of multiple random effects (e.g., participant and item effects)

- Linear Mixed Models (LMMs) overcome many of these limitations and can be used if there are multiple and crossed random effects or hierarchical or multilevel structures in the data.

- `afex` contains the convenience function `mixed()` to obtain *p* values for mixed models and fits them with `lme4::lmer` (the package of choice for mixed models in R).

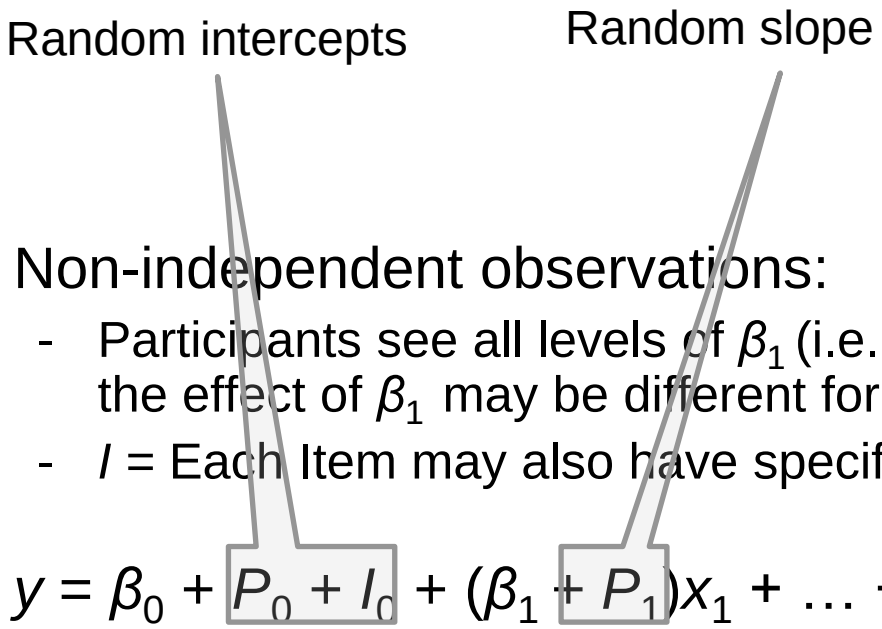
Linear Mixed Models (LMMs)

- One interval scaled response variable *y*
- *m* predictors (*β)*
- *Linear Model* (Observations are independent)*:*
  - $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m + \varepsilon$,
    where $\varepsilon \sim N(0, \sigma^2)$

- Non-independent observations:
  - Participants see all levels of β_1 (i.e., within-subjects factor), and the effect of β_1 may be different for each participant *P*
  - *I* = Each Item may also have specific effects

- $y = \beta_0 + P_0 + I_0 + (\beta_1 + P_1)x_1 + \ldots + \beta_m x_m + \varepsilon$,
  where $\varepsilon \sim N(0, \sigma^2)$,
  $(P_0, P_1) \sim N(0, [\ldots])$,
  $I_0, \sim N(0, \omega^2)$

Linear Mixed Models (LMMs)



- Non-independent observations:
  - Participants see all levels of β_1 (i.e., within-subjects factor), and the effect of β_1 may be different for each participant *P*
  - *I* = Each Item may also have specific effects

- $y = \beta_0 + P_0 + I_0 + (\beta_1 + P_1)x_1 + \ldots + \beta_m x_m + \varepsilon$,
  where $\varepsilon \sim N(0, \sigma^2)$,
  $(P_0, P_1) \sim N(0, [\ldots])$,
  $I_0, \sim N(0, \omega^2)$,

`lme4` and *p* values

- Obtaining *p* values for `lme4` models is not trivial, as
  a. the sampling distribution of the NULL hypothesis is problematic and
  b. the correct number of denominator degrees of freedoms is unknown.

- `mixed()` implements the "best" two options to overcome this problem (according to the lme4 faq)
  - for LMMs: the Kenward-Rogers approximation for df (`method = "KR"`) [also offered in `car::Anova(…, test = "F")`]
  - for GLMMs and LMMs: Parametric bootstrap (`method = "PB"`)
  - both options are achieved through package `pbkrtest` (Halekoh & Hojsgaard, 2012).

mixed()

UNI FREIBURG

- `mixed()` is a wrapper of `lme4::lmer()` with the additional arguments:
  - `type`: type of "sums of squares" (i.e., how should effects be calculated), default is 3
  - `method`: Kenward-Rogers (`"KR"`, the default, needs a lot of RAM) or parametric bootstrap (`"PB"`, can be parallelized using the `parallel` package)
  - `args.test`: further arguments passed to `pbkrtest`.

```
m1 <- mixed(resp ~ cond * validity * believability
  + (1 + (believability * validity)|id) +
  (1 + validity|content), d)
```

mixed() – return value

- returns an S3 object of class "mixed" with print, summary, and anova methods (which are all the same)

```
> str(m1, 1)
List of 6
 $ anova.table      :'data.frame':      8 obs. of  11
  variables:
 $ full.model       :Formal class 'mer' [package "lme4"] with
  34 slots
 $ restricted.models:List of 8
 $ tests            :List of 8
 $ type             : num 3
 $ method           : chr "KR"
 - attr(*, "class")= chr "mixed"
```

mixed()

```
> m1
                        Effect      stat ndf     ddf F.scaling p.value
1                  (Intercept) 1872.7004   1 37.8616    1.0000  0.0000
2                         cond    0.0142   1 56.4587    1.0000  0.9056
3                     validity    0.0122   1 19.9984    1.0000  0.9133
4                believability    3.2928   2 25.5996    0.9974  0.0534
5                cond:validity    1.7338   1 83.4363    1.0000  0.1915
6           cond:believability    0.2854   2 51.3245    0.9826  0.7529
7       validity:believability    2.8739   2 17.4208    0.9902  0.0834
8 cond:validity:believability    0.3880   2 84.5394    0.9890  0.6796
```

- All effects disappear, which is in line with our hypothesis that the finding is simply an item effect (as believability is completely nested within the item, it disappears as well)

Generallized Linear Mixed Models (GLMMs)

- One interval scaled response variable *y*
- *m* predictors (β), repeated measures on β_1, and *P* and *I* effects
- $y = \beta_0 + P_0 + I_0 + (\beta_1 + P_1)x_1 + \ldots + \beta_m x_m + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$, $(P_0, P_1) \sim N(0, [\ldots])$, $I_0, \sim N(0, \omega^2)$.
- The dependent variable *dv* directly corresponds to the predicted variable *y*.

- For e.g., binomial (i.e., 0,1) data this is not the case and we need a function that links *y* to *dv*, which would be the logit function.
  (In addition to the link function we also need to specify the distribution of ε)

GLMMs

- Suppose the dependent variable in our data was not interval scaled, but binary (i.e., if <= 3, 0, else 1).

- In this case we would need to extend our linear mixed model to a non-gaussian linking function from the binomial family (the default binomial linking function is logit).

- For this case we would use `method = "PB"`:

```
m2 <- mixed(resp2 ~ cond * validity *
believability + (1 + (believability *
validity)|id) + (1 + validity|content), d,
family = binomial, method = "PB", args.test =
list(nsim = 30))
```

GLMM – results

```
> m2
                                 Effect   stat p.value
1                           (Intercept) 1.3459  0.3226
2                                  cond 0.2106  0.6774
3                              validity 0.0758  0.7419
4                         believability 8.6075  0.0323
5                         cond:validity 1.4799  0.2258
6                    cond:believability 2.7979  0.1613
7               validity:believability 8.0541  0.0645
8 cond:validity:believability 2.9233  0.2581
```

mixed()

- `mixed()` can be used to obtain *p* values for LMMs and GLMMs by fitting different versions of the model (using `lmer`) and then comparing those with a larger model (via `pbkrtest`).

- Type 3 tests: The full model is compared with a model in which only the effect is excluded.

- Type 2 tests: For each effect a model in which all higher order effects are excluded is tested against one in which all higher and this effects are excluded.

- Note, effects are excluded by directly altering the model matrix (and not by excluding it via R formula).

compare.2.vectors()

- compares two vectors using different tests:

```
> compare.2.vectors(1:10, c(7:20, 200))

$parametric
   test test.statistic test.value test.df         p
1     t              t  -1.325921 23.0000 0.1978842
2 Welch              t  -1.632903 14.1646 0.1245135

$nonparametric
             test test.statistic test.value test.df           p
1 stats::Wilcoxon              W   8.000000      NA 0.0002228503
2     permutation              Z  -1.305464      NA 0.0979700000
3  coin::Wilcoxon              Z  -3.719353      NA 0.0000200000
4          median              Z   3.545621      NA 0.0005600000
```

- uses per default 100,000 Monte Carlo samples to estimate an approximation of the exact conditional distribution (for the last three tests) using coin (Hothorn, Hornik, van de Wiel, & Zeileis, 2008, JSS)

Summary

- afex provides convenience functions for specifying the statistical model for factorial experimental designs:
  - ANOVA: `aov.car()` and `ez.glm()`
  - `mixed()` for LMMs and GLMMs (i.e., models with potentially crossed random effects)

- The objects returned by those functions can be passed to other functions for further inspection
  - to `phia` (de Rosario-Martinez, 2012) for ANOVAs or LMMs
  - to `multcomp` (Hothorn, Bretz & Westfall, 2008) for mixed models

- Two vectors (unpaired or paired) can be compared with `compare.2.vectors` using *t*-, (Welch-), Wilcoxon-, and permutation-test

UNI FREIBURG

Thank you for your attention.

Outlook on future developments

UNI FREIBURG

- Provide easier analysis of LMMs with only one random effect (i.e., an `ez.lmm` function), to analyze classical ANOVA designs with mixed models.
- Provide method to conveniently obtain multiple comparisons via `multcomp` from objects returned by `mixed`.
- Implement tests based on wald-Tests in addition to LRT based tests (similar to car::Anova).

Why are Type 3 tests standard?

UNI FREIBURG

- Type 2 tests assume no higher order effects for any effect, and tests of lower order effects are meaningless if higher-order effects are present.

- Type 3 tests do not have this requirements, they calculate tests of lower-order effects in presence of higher-order effects.

- Many statisticians prefer Type 2 tests as
  - they are more powerful (Lansgrund, 2003),
  - do not violate marginality (Venables, 2000),
  - and most notably if interactions are present, main effects are per se not interpretable.

Why are Type 3 tests standard?

UNI FREIBURG

- Others recommend Type 3 tests (in applied settings):
  - beta errors for interactions are possible (as they are harder to detect as main effects, e.g., Maxwell & Delaney, 2004), in these cases lower-order effects would be not-interpretable even in the absence of higher-order effects.

- Tests of main effects are mostly problematic if the significant higher-order interactions are disordinal (which one can usually evaluate for experimental designs).

- Type 3 are the standard in most commercial software packages and therefore the usually expected type of tests.