# Statistical Models for Dependent Data: Handout

*Henrik Singmann*

*November 2017*

### Overview: Statistical Models in R

1. Identify probability distribution of data (more correct: of residuals/conditional distribution)
2. Make sure variables are of correct type via `str()`
3. Set appropriate contrasts (orthogonal contrasts if model includes interaction): `afex::set_sum_contrasts()`
4. Describe statistical model using `formula`
5. Fit model: pass `formula` and `data.frame` to corresponding modeling function (e.g., `lm()`, `glm()`)
6. Check model fit (e.g., inspect residuals)
7. Test terms (i.e., main effects and interactions): Pass fitted model to `car::Anova()`
8. Follow-up tests:
   - Estimated marginal means: Pass fitted model to `lsmeans::lsmeans()`/`emmeans::emmeans()`
   - Specify specific contrasts on estimated marginal means (e.g., `contrast()`, `pairs()`)

- `afex` combines fitting (5.) and testing (7.):
  - ANOVAs: `afex::aov_car()`, `afex::aov_ez()`, or `afex::aov_4()`
  - (Generalized) linear mixed-effects models: `afex::mixed()`

### `R` Formula Interface for Statistical Models: `~`

- `R` `formula` interface allows symbolic specification of statistical models, e.g. linear models: `lm(ACT ~ SATQ, sat.act)`
- Dependent variable(s) left of `~` (can be multivariate or missing), independent variables right of `~`:

| Formula | Interpretation |
|---|---|
| `~ x` or `~1+x` | Intercept and main effect of `x` |
| `~ x-1` or `~0 + x` | Only main effect of `x` and no intercept (questionable) |
| `~ x+y` | Main effects of `x` and `y` |
| `~ x:y` | Interaction between `x` and `y` (and no main effect) |
| `~ x*y` or `~ x+y+x:y` | Main effects and interaction between `x` and `y` |

- **Formulas behave differently for coninuous and categorical covariates!!**
  - Always use `str(data)` before fitting: `int` & `num` is continuous, `Factor` or `character` is categorical.
  - Categorical/nominal variables have to be `factor`s. Create via `factor()`.
- Categorical variables are transformed into numerical variables using contrast functions (via `model.matrix()`; see Cohen et al., 2002)
  - **If models include interactions, orthogonal contrasts (e.g., `contr.sum`) in which the intercept corresponds to the (unweighted) grand mean should be used**: `afex::set_sum_contrasts()`
  - Dummy/treatment contrasts (`R` default) lead to simple effects for lower order effects.
  - For linear models: Coding only affects interpretation of parameters/tests not overall model fit.
- For models with only numerical covariates, suppressing intercept works as expected.
- For models with categorical covariates, suppressing intercept or other lower-order effects often leads to very surprising results (and should generally be avoided).

### Tests of Model Terms/Effects with `car::Anova()`

- `car::Anova(model, type = 3)` general solution for testing effects.
- Type II and III tests equivalent for balanced designs (i.e., equal group sizes) and highest-order effect.
- Type III tests require orthogonal contrasts (e.g.,`contr.sum`); recommended:
  - For experimental designs in which imbalance is completely random and not structural,
  - Complete cross-over interactions (i.e., main effects in presence of interaction) possible.
- Type II are more appropriate if imbalance is structural (i.e., observational data; maybe here).

### Follow-up Tests with `lsmeans`/`emmeans`

- `lsmeans(model, ~factor)`/`emmeans(model, ~factor)` produces estimates marginal means (or least-square means for linear regression) for model terms (e.g., `lsmeans(m6, ~education*gender)`).
- Additional functions allow specifying contrasts/follow-up tests on the means, e.g.:
  - `pairs()` tests all pairwise comparisons among means.
  - `contrast()` allows to define arbitrary contrasts on marginal means.
  - For more examples see the Vignettes: https://cran.r-project.org/package=emmeans

### ANOVAs with afex

- `afex` ANOVA functions require column with participant ID:
  - `afex::aov_car()` allows specification of ANOVA using `aov`-like formula. Specification of participant id in `Error()` term. For example:
    `aov_car(dv ~ between_factor + Error(id/within_factor), data)`
  - `afex::aov_4()` allows specification of ANOVA using `lme4`-like formula. Specification of participant id in random term. For example:
    `aov_4(dv ~ between_factor + (within_factor|id), data)`
  - `afex::aov_ez()` allows specification of ANOVA using characters. For example:
    `aov_ez("id", "dv", data, between = "between_factor", within = "within_factor")`

### Repeated-Measures and IID Assumption

- Ordinary linear regression, between-subjects ANOVA, and basically all standard statistical models share one assumption: Data points are *independent and identically distributed* (*iid*).
  - Independence assumption refers to residuals: After taking structure of model (i.e., parameters) into account, probability of a data point having a specific value is independent of all other data points.
  - Identical distribution: All observations sampled from same distribution.
- For repeated-measures independence assumption often violated, which can have dramatic consequences on significance tests from model (e.g., increased or decreased Type I errors).
- Three ways to deal with repeated-measures:
  1. *Complete pooling*: Ignore dependency in data (often not appropriate, results likely biased, not trustworthy)
  2. *No pooling*: Separate data based on factor producing dependency and calculate separate statistical model for each subset (combining results can be non-trivial)
  3. *Partial pooling*: Analyse data jointly while taking dependency into account (gold standard, e.g., mixed models)