LINEAR MIXED-EFFECTS MODELING WITH CROSSED RANDOM FACTORS

STATISTICAL MODELS

PSYCHOLOGY, UNIVERSITY OF GLASGOW

OVERVIEW

- generalizing to subjects and stimuli
  - dealing with "crossed" random factors
- random effects in complex designs
- non-convergence and model validation

GENERALIZING TO SUBJECTS AND STIMULI

LANGUAGE-AS-FIXED-EFFECT FALLACY

- Psycholinguistic experiments sample language materials as well as subjects
- Language stimuli should be treated as a random, not fixed factor
- Clark's suggestion: F', min-F'
- Modern solution: Linear-mixed effects with crossed random factors of subjects and stimuli

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*, 390-412.

Clark, H. H. (1973). The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior*, *12*, 335-359.

CROSSED RANDOM FACTORS

```
inner_join(subj, lists, "list_id")
```

	subj_id	list_id	stim_id	condition
1	1	1	A	treatment
2	1	1	B	treatment
3	1	1	C	control
4	1	1	D	control
5	2	2	A	control
6	2	2	B	control
7	2	2	C	treatment
8	2	2	D	treatment
9	3	1	A	treatment
10	3	1	B	treatment
11	3	1	C	control
12	3	1	D	control
13	4	2	A	control
14	4	2	B	control
15	4	2	C	treatment
16	4	2	D	treatment

GENERALIZING OVER ENCOUNTERS

The target of inference in much of psychology and related fields has been misidentified as a population of *subjects* or *stimuli*, when the actual target of inference is a population of events: **encounters**

- readers encountering particular types of words
- male participants judging attractiveness of female faces, or vice versa
- gamers encountering particular types of violent games
- audience members encountering particular types of dance movements
- insomniacs (versus controls) encountering emotional expressions
- birds hearing particular types of birdsongs

Barr, D. J. (2018). Generalizing Over Encounters. In the *Oxford Handbook of Psycholinguistics.*

SPECIFYING RANDOM EFFECTS

GENERAL PROCEDURE FOR FACTORIAL DESIGNS

for each random factor (subjects/stimuli):

1. identify within-unit factors
2. check highest-order combination of within-subject factors
    - *NO pseudoreplications:* no random slopes
    - *YES pseudoreplications:* all interactions/main effects get slopes

between-unit factors (or interactions involving them) never get random slopes

Barr, D. J. (2013). Random effects structure for testing interactions in linear mixed-effects models. *Frontiers in Psychology*, *4*, 328.

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, *68*, 255-278.

DETERMINING THE DESIGN FROM DATA

three way design, subjects only random factor

```
# A tibble: 128 x 5
   subj_id A     B     C          DV
     <int> <chr> <chr> <chr>   <dbl>
 1       5 A1    B1    C2      4.80
 2       5 A1    B2    C1      7.40
 3       1 A2    B1    C2      0.444
 4       6 A2    B1    C2      3.97
 5       4 A1    B1    C2      2.90
 6       4 A1    B2    C1     -0.0699
 7       3 A1    B2    C2      2.75
 8       1 A2    B2    C2      2.86
 9       3 A1    B2    C2      0.452
10       8 A1    B1    C1     -0.694
# … with 118 more rows
```

```
count(dat1, subj_id, A, B, C)

# A tibble: 32 x 5
   subj_id A     B     C         n
     <int> <chr> <chr> <chr> <int>
 1       1 A2    B1    C1        4
 2       1 A2    B1    C2        4
 3       1 A2    B2    C1        4
 4       1 A2    B2    C2        4
 5       2 A2    B1    C1        4
 6       2 A2    B1    C2        4
 7       2 A2    B2    C1        4
 8       2 A2    B2    C2        4
 9       3 A1    B1    C1        4
10       3 A1    B1    C2        4
# … with 22 more rows
```

A is between, BC within, 4 obs / cell

```
DV ~ A * B * C + (B * C | subj_id)
```

EXAMPLE WITH CROSSED RANDOM FACTORS

DV ~ A * B * C + (? | subj_id) + (? | item_id)

```
dat2
```

```
# A tibble: 128 x 6
   subj_id item_id A     B     C         DV
     <int>   <int> <chr> <chr> <chr>  <dbl>
 1       7      13 A1    B2    C2     1.41
 2       3       6 A1    B1    C2     3.62
 3       3       1 A2    B2    C2     2.79
 4       2       4 A1    B2    C1    -1.41
 5       4      10 A2    B1    C1     0.971
 6       7       5 A2    B2    C2     8.12
 7       7      16 A2    B1    C1     7.66
 8       6       3 A2    B1    C1     3.00
 9       7      10 A1    B1    C2     0.644
10       6       9 A2    B2    C2    10.7
# … with 118 more rows
```

BY-SUBJECT RANDOM EFFECTS

```
dat2 %>%
  count(subj_id, A, B, C)
```

```
# A tibble: 64 x 5
   subj_id A     B     C         n
     <int> <chr> <chr> <chr> <int>
 1       1 A1    B1    C1        2
 2       1 A1    B1    C2        2
 3       1 A1    B2    C1        2
 4       1 A1    B2    C2        2
 5       1 A2    B1    C1        2
 6       1 A2    B1    C2        2
 7       1 A2    B2    C1        2
 8       1 A2    B2    C2        2
 9       2 A1    B1    C1        2
10       2 A1    B1    C2        2
# … with 54 more rows
```

```
DV ~ A * B * C + (A * B * C | subj_id) + (? | item_id)
```

BY-STIMULUS RANDOM EFFECTS

```
dat2 %>%
  count(item_id, A, B, C)
```

```
# A tibble: 128 x 5
   item_id A     B     C         n
     <int> <chr> <chr> <chr> <int>
 1       1 A1    B1    C1        1
 2       1 A1    B1    C2        1
 3       1 A1    B2    C1        1
 4       1 A1    B2    C2        1
 5       1 A2    B1    C1        1
 6       1 A2    B1    C2        1
 7       1 A2    B2    C1        1
 8       1 A2    B2    C2        1
 9       2 A1    B1    C1        1
10       2 A1    B1    C2        1
# … with 118 more rows
```

```
DV ~ A * B * C + (A * B * C | subj_id) + (1 | item_id)
```

NON-CONVERGENCE AND MODEL VALIDATION

NON-CONVERGENCE

When you get a convergence warning you should in the first instance:

- double-check the model specification
- make sure all predictors are scaled and centred

then re-fit the model. If it still does not converge, seek to reduce the random effects structure, but **proceed with caution.**

You can also try different optimizers (see `?lme4::convergence`)

"*`model is singular`*" *is NOT a convergence warning*

REDUCING RANDOM EFFECTS STRUCTURE

Reducing random effects can help convergence, but the worst thing you can do is remove the slope for a theory-critical predictor.

1. Remove random correlations and re-fit
    - Use `(A * B || subject)`
2. Worst case scenario: effectwise testing, e.g.:
    - test A using (A | subject) + (A | stimulus)
    - test B using (B | subject) + (B | stimulus)
    - test AB using (A:B | subject) + (A:B | stimulus)

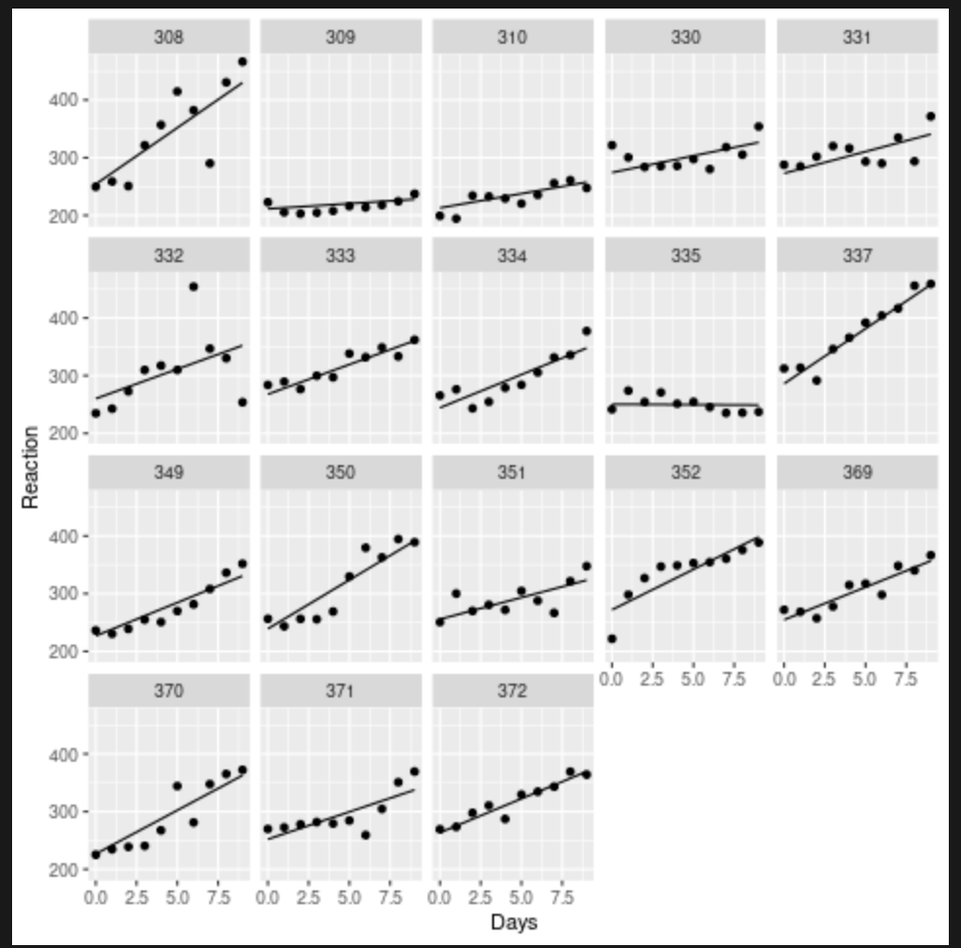
CHECKING ASSUMPTIONS

- linearity
- homogeneity of variance
- normality of residuals
    - outliers
    - multimodality
    - other weirdness (skew, etc)

LINEARITY

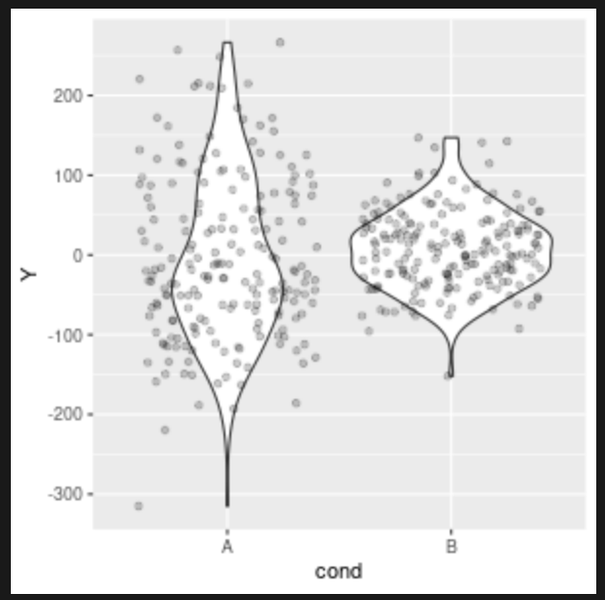
- fitted (line) v. observed (points)

```
mod <- lmer(Reaction ~ Days +
              (Days | Subject),
            sleepstudy, REML = FALSE)

## fitted values:
##   fitted(mod)
## residuals:
##   residuals(mod)
```



HOMOGENEITY OF VARIANCE

```
n_obs <- 200L
dat3 <- tibble(cond = rep(c("A", "B"),
                          each = n_obs)
               Y = c(rnorm(n_obs, 0, 10
                     rnorm(n_obs, 0, 50

ggplot(dat3, aes(cond, Y)) +
  geom_violin() +
  geom_jitter(alpha = .2)
```




```
n_pts <- 10L
n_obs <- 50
dat4 <- tibble(
  x = seq_len(n_pts),
  y = map(x, # purrr::map()
          ~ rnorm(n_obs, 5 * .x, 10 * .
  unnest() # tidyr::unnest()

ggplot(dat4, aes(x, y)) +
  geom_point(alpha = .2) +
  geom_smooth(method = "lm")
```

NORMALITY OF RESIDUALS

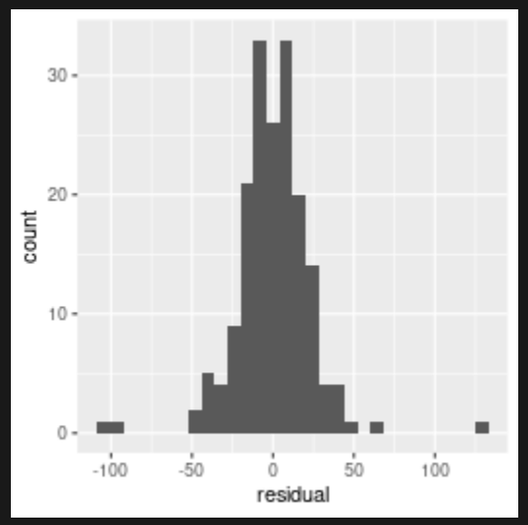
- don't visualize the raw DV to check for normality!

https://dalejbarr.shinyapps.io/raw_vs_resids/

VISUAL CHECKS

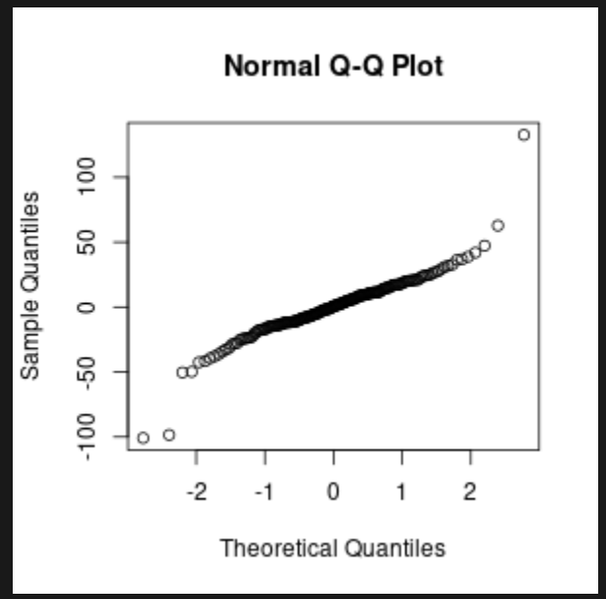
- histogram

```
## resids from the model fit to sleepst
my_resids <- residuals(mod)

## it is a vector, must put into a tibb
## for ggplot
rtbl <- tibble(residual = my_resids)

ggplot(rtbl, aes(residual)) +
  geom_histogram()
```



- Q-Q plot (quantile-quantile)

```
## sadly there is no qqplot for ggplot
## so we use base::qqnorm()
qqnorm(my_resids)
```



Vanhove, J. (2018). *Checking the assumptions of your statistical model without getting paranoid.* Preprint at https://psyarxiv.com/zvawb/.

Created by Dale Barr