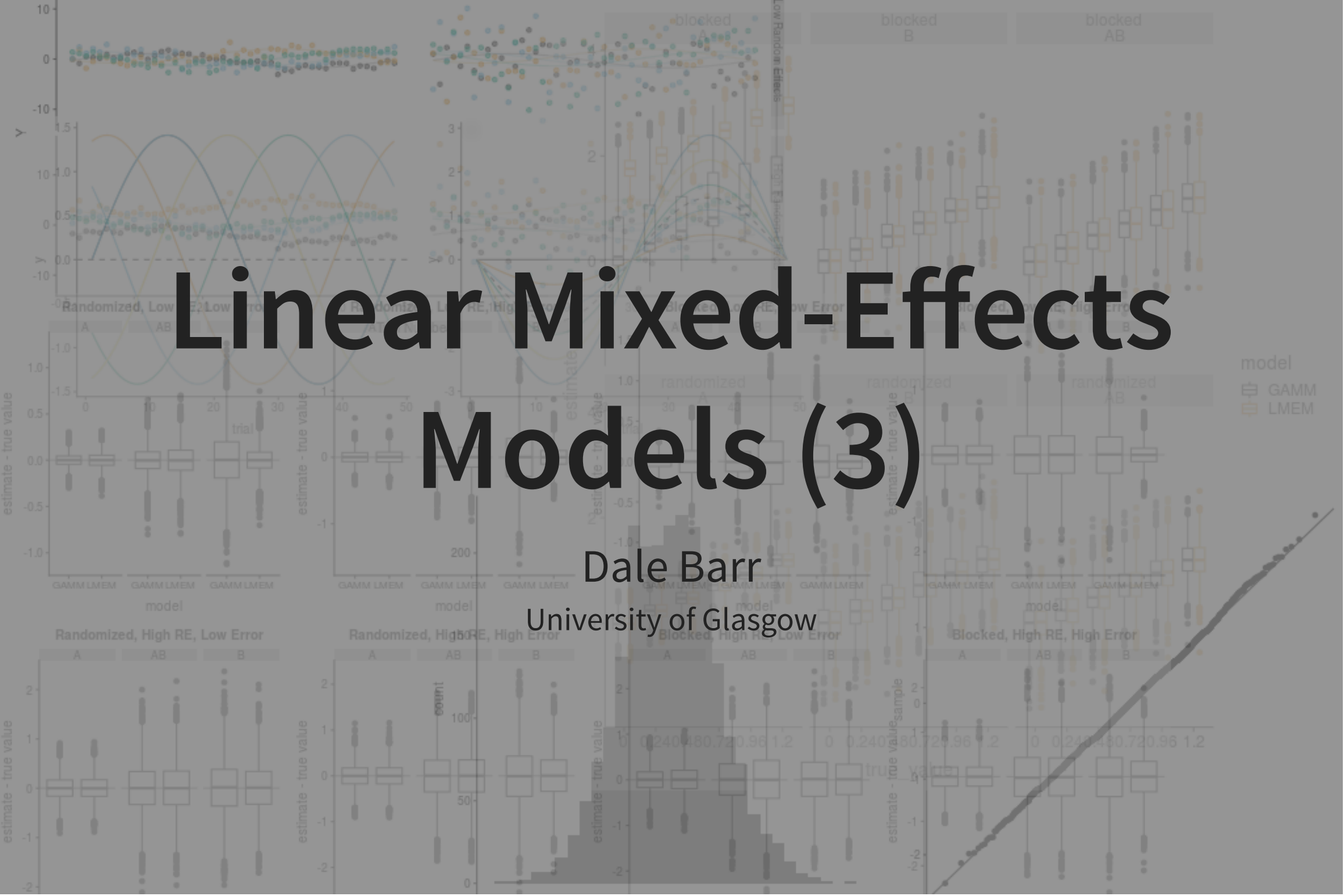
Linear Mixed-Effects Models (3)

Dale Barr

University of Glasgow



overview

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

Language-as-fixed-effect fallacy

- Psycholinguistic experiments sample language materials as well as subjects
- Language stimuli should 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

```
# A tibble: 4 × 2
  subj_id list_id
    <int>   <int>
1       1       1
2       2       2
3       3       2
4       4       1
```

```
# A tibble: 8 × 3
  list_id stim_id condition
    <int> <chr>   <chr>
1       1 A       treatment
2       1 B       treatment
3       1 C       control
4       1 D       control
5       2 A       control
6       2 B       control
7       2 C       treatment
8       2 D       treatment
```

```
# A tibble: 16 × 4
   subj_id list_id stim_id condition
     <int>   <int> <chr>   <chr>
 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       2 A       control
10       3       2 B       control
11       3       2 C       treatment
12       3       2 D       treatment
13       4       1 A       treatment
14       4       1 B       treatment
15       4       1 C       control
16       4       1 D       control
```

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 *Oxford Handbook of Psycholinguistics.*

specifying random effects

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 × 5
   subj_id A     B     C          DV
     <int> <chr> <chr> <chr>   <dbl>
 1       6 A2    B1    C2      1.33
 2       7 A1    B1    C2      5.90
 3       3 A2    B1    C1      1.15
 4       8 A2    B2    C2      2.91
 5       2 A1    B2    C1      1.66
 6       1 A1    B1    C1      0.0620
 7       6 A2    B2    C1      3.21
 8       8 A2    B1    C1     -1.04
 9       2 A1    B1    C2     -0.0528
10       2 A1    B2    C1      1.39
# ℹ 118 more rows
```

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

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

A is between, BC within, 4 obs / cell

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

crossed random factors

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

```
# A tibble: 128 × 6
   subj_id item_id A     B     C          DV
     <int>   <int> <chr> <chr> <chr>   <dbl>
 1       8      15 A1    B1    C1      1.08
 2       3       3 A2    B1    C1     -1.81
 3       5      12 A1    B1    C2      3.55
 4       4       1 A2    B2    C2      3.97
 5       3      15 A1    B2    C1     -0.827
 6       7      12 A1    B2    C2      6.20
 7       1      12 A2    B2    C2      1.55
 8       6       2 A2    B1    C2      3.23
 9       7       6 A1    B1    C1      3.26
10       7       7 A1    B1    C2      7.35
# ℹ 118 more rows
```

by-subject random effects

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

```
# A tibble: 64 × 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
# ℹ 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 × 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
# ℹ 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.**

Also, try different optimizers (`?lme4::convergence`)

`"singular fit"` 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)`

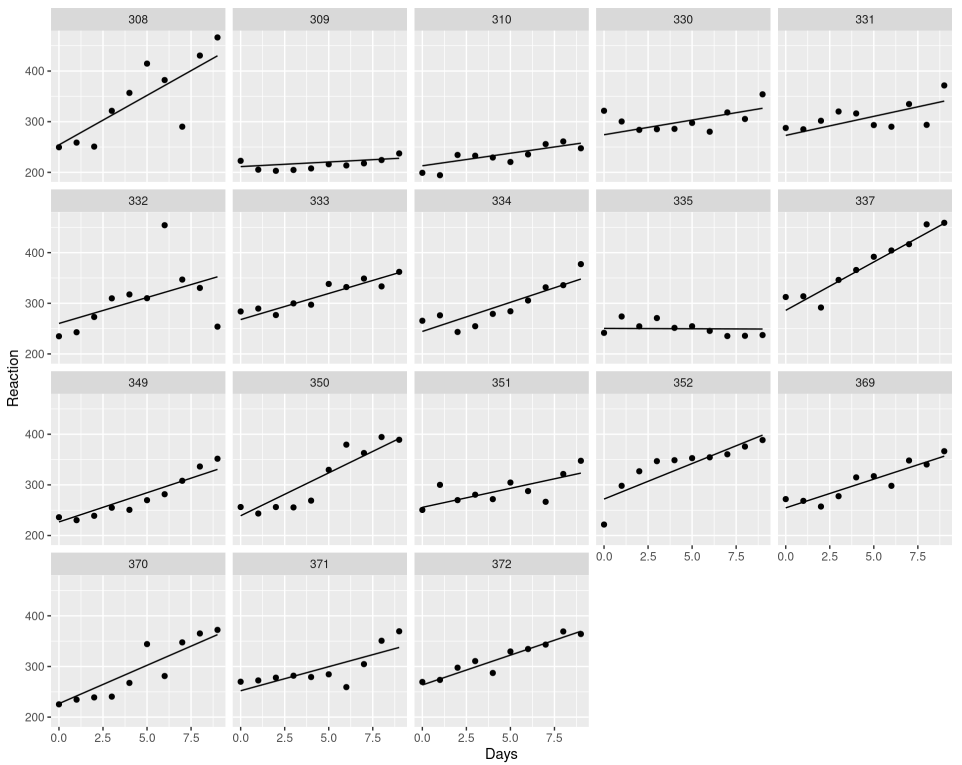
checking assumptions

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

linearity

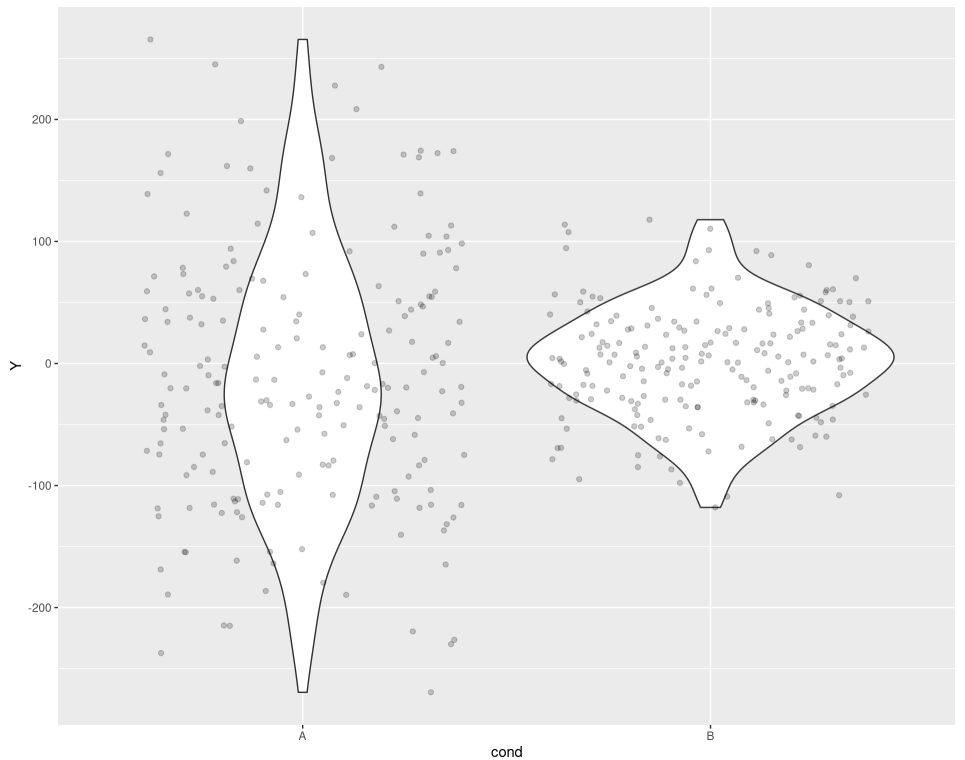
- fitted (line) v. observed (points)

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

## fitted values:  fitted(mod)
## residuals:     residuals(mod)
ss2 <- sleepstudy |>
  mutate(fits = fitted(mod))

ggplot(ss2, aes(Days, Reaction)) +
  geom_line(aes(y = fits,
                group = Subject)) +
  geom_point() +
  facet_wrap(~Subject)
```



homogeneity of variance

```
n_obs <- 200L
dat3 <- tibble(
  cond = rep(c("A", "B"),
             each = n_obs),
  Y = c(rnorm(n_obs, 0, 100),
        rnorm(n_obs, 0, 50)))

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



normality of residuals

Visualizing Raw Data vs Residuals

Mean RT for Subject A:

500 600 900

500 540 580 620 660 700 740 780 820 860 900

Mean RT for Subject B:

500 650 900

500 540 580 620 660 700 740 780 820 860 900

Mean RT for Subject C:

500 700 900

500 540 580 620 660 700 740 780 820 860 900

Mean RT for Subject D:

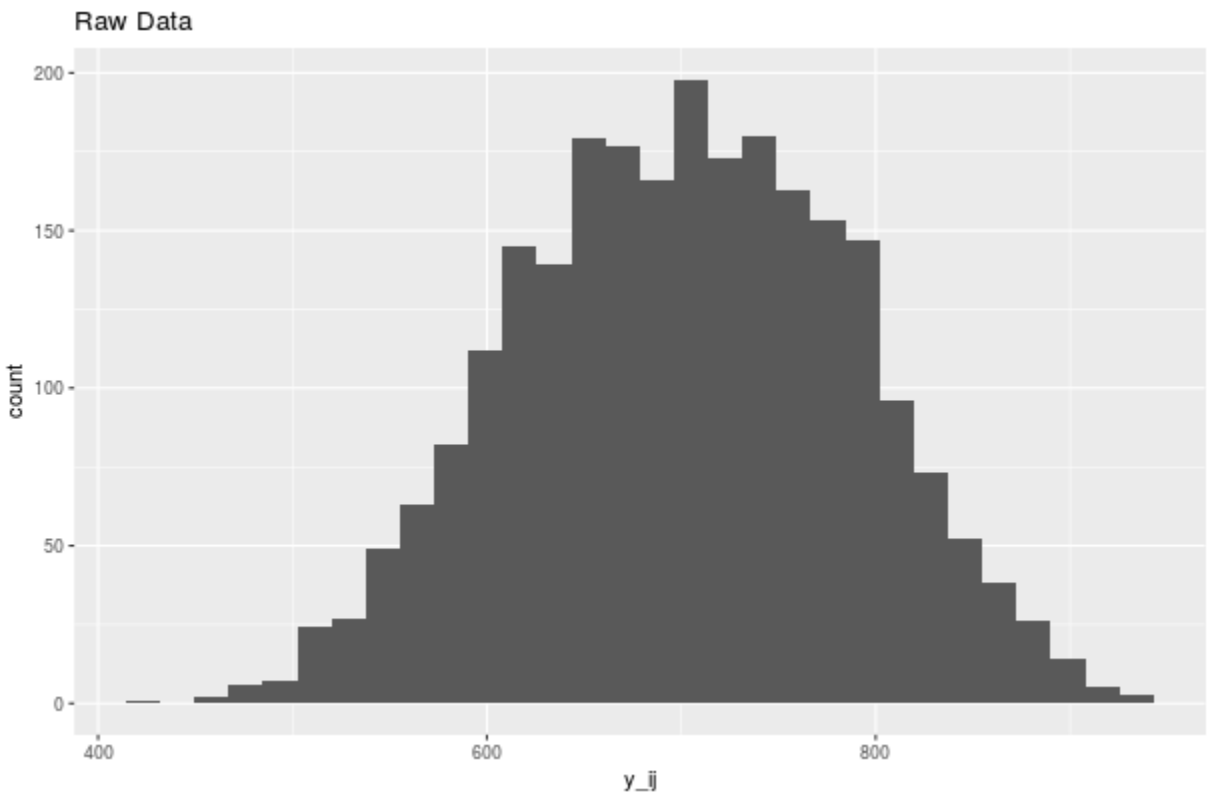
500 750 900

500 540 580 620 660 700 740 780 820 860 900

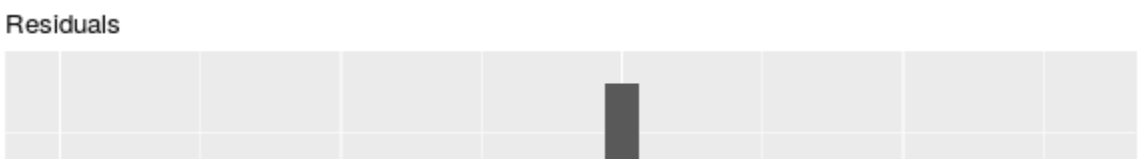
Mean RT for Subject E:

500 800 900

500 540 580 620 660 700 740 780 820 860 900

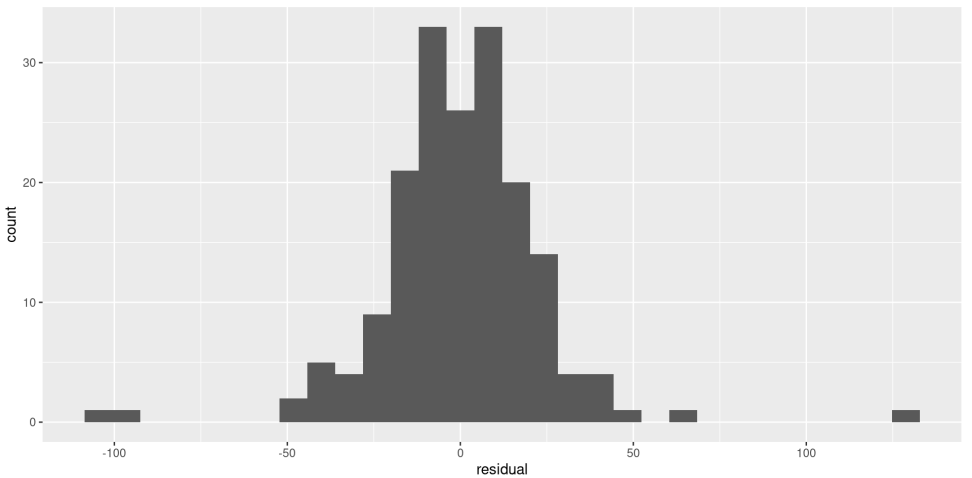


Shapiro-Wilk normality test data: (raw data) W = 0.99658, p-value = 1.962e-05



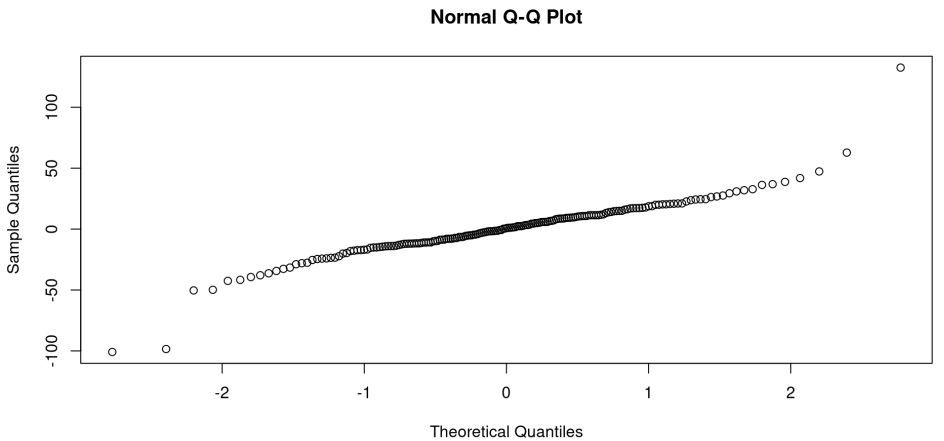
visual checks: histogram

```r
my_resids <- residuals(mod)

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

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



visual checks: quantile-quantile (qq)

```
## 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/