Many Models with lme4::lmer() and broom.mixed::tidy()

Brian High

2022-10-24

Objectives

- Run many models with `lme4::lmer()` and extract estimates with `broom.mixed::tidy()`.
- Use `furrr::future_map()` for parallel processing.
- Compare results of:
  - `lme4::confint.merMod(method = "Wald")`
  - `broom.mixed::tidy(conf.int = TRUE, conf.method = "Wald")`

Setup

We will install LaTeX support for rendering PDFs, if not already present.

```
# Install LaTeX environment (for rendering PDFs from RMarkdown, etc.)
pdflatex_ver <- try(system("pdflatex -v", intern = T, wait = T), silent = T)
pdflatex_ver <- grep("^pdfTeX", pdflatex_ver, value = T)
if (!(exists("pdflatex_ver") & length(pdflatex_ver) > 0)) {
  if (!requireNamespace("tinytex", quietly = TRUE)) install.packages("tinytex")
  if (!dir.exists(tinytex::tinytex_root(error = F))) tinytex::install_tinytex()
}
```

We will use the *broom.mixed* package to support `lmer()` from the *lme4* package.

```
# Install pacman if not installed.
if (!requireNamespace("pacman", quietly = TRUE)) {
  install.packages('pacman', repos = 'https://cloud.r-project.org', quiet = TRUE)
}

# Load packages, installing as needed.
suppressPackageStartupMessages(
  pacman::p_load(
    nycflights13,
    tibble,
    tidyr,
    dplyr,
    lme4,
    broom.mixed,
    purrr,
    furrr,
    tictoc,
    kableExtra
  )
)
```

We will set the number of parallel “workers” to half the number of available CPU cores.

```
# Show number of available CPU cores.
availableCores()
```

```
## system
##      2
```

```
# Set number of multicore "workers" to 1/2 the number of cores.
plan(multisession, workers = availableCores()/2)
```

Prepare formulas

We use a list so that we can have named groups of formulas.

```
# Create a list of formulas for use with lmer().
formulas <- list(
  crude = c(
    "arr_delay ~ distance + (1|carrier)",
    "arr_delay ~ air_time + (1|carrier)",
    "arr_delay ~ month + (1|carrier)"
  ),
  min = c(
    "arr_delay ~ distance + air_time + (1|carrier)",
    "arr_delay ~ distance + month + (1|carrier)",
    "arr_delay ~ air_time + month + (1|carrier)"
  ),
  max = c("arr_delay ~ distance + air_time + month + (1|carrier)")
)

# Convert the list of formulas into a data.frame (tibble).
formula_df <- formulas %>%
  enframe(name = "fgroup", value = "formula") %>% unnest(formula)
```

Prepare data

Our dataset will be `flights` from the *nycflights13* package.

```
# Select only the model variables to minimize parallelization overhead.
data_df <-
  flights %>% select(arr_delay, distance, air_time, month, carrier)
```

Run the models

We could use `nest()` for the following steps in a single pipeline, using `tidy()` for confidence intervals, but the terms might be misaligned.

Instead, using a stepwise approach, we retain the term names, use `confint.merMod()` to get confidence intervals, join on the terms and the formula number, and show the results to compare the two methods of getting confidence intervals.

```r
# Start timer.
tic()

# Fit the models with lmer() using future_map() for multicore processing.
model_fit_list <- formula_df$formula %>% future_map(lmer, data = data_df)

# Extract the estimates with broom.mixed::tidy().
est <-
  model_fit_list %>%
  map(broom.mixed::tidy, conf.int = TRUE,
      conf.method = "Wald", conf.level = 0.95) %>%
  bind_rows(.id = "ID") %>%
  select(-group)

# Calculate confidence intervals with confint.merMod().
CI <-
  model_fit_list %>%
  map(confint.merMod, method = "Wald", level = 0.95) %>%
  map(as_tibble, rownames = "term") %>%
  bind_rows(.id = "ID")

# Merge estimates and confidence intervals by formula number (ID) and term.
results_df <- formula_df %>%
  rownames_to_column(var = "ID") %>%
  inner_join(est, by = "ID") %>%
  inner_join(CI, by = c("ID", "term")) %>%
  select(-ID, -effect, -std.error, -statistic) %>%
  arrange(fgroup, formula, term)

# Stop timer.
toc()
```

```
## 22.107 sec elapsed
```

Display the results.

```r
# Note: Use kable_styling(full_width = TRUE) for HTML output.
results_df %>% knitr::kable(digits = 4) %>% kable_styling(font_size = 8)
```

fgroup	formula	term	estimate	conf.low	conf.high	2.5 %	97.5 %
crude	arr_delay ~ air_time + (1|carrier)	(Intercept)	5.5096	0.6767	10.3424	0.6767	10.3424
crude	arr_delay ~ air_time + (1|carrier)	air_time	0.0085	0.0065	0.0105	0.0065	0.0105
crude	arr_delay ~ distance + (1|carrier)	(Intercept)	8.9182	4.8971	12.9394	4.8971	12.9394
crude	arr_delay ~ distance + (1|carrier)	distance	-0.0014	-0.0016	-0.0011	-0.0016	-0.0011
crude	arr_delay ~ month + (1|carrier)	(Intercept)	8.5815	4.1367	13.0263	4.1367	13.0263
crude	arr_delay ~ month + (1|carrier)	month	-0.2242	-0.2686	-0.1797	-0.2686	-0.1797
max	arr_delay ~ distance + air_time + month + (1|carrier)	(Intercept)	-2.7542	-6.7541	1.2457	-6.7541	1.2457
max	arr_delay ~ distance + air_time + month + (1|carrier)	air_time	0.6717	0.6599	0.6835	0.6599	0.6835
max	arr_delay ~ distance + air_time + month + (1|carrier)	distance	-0.0862	-0.0877	-0.0847	-0.0877	-0.0847
max	arr_delay ~ distance + air_time + month + (1|carrier)	month	-0.0443	-0.0881	-0.0006	-0.0881	-0.0006
min	arr_delay ~ air_time + month + (1|carrier)	(Intercept)	6.9904	2.1389	11.8419	2.1389	11.8419
min	arr_delay ~ air_time + month + (1|carrier)	air_time	0.0086	0.0066	0.0106	0.0066	0.0106
min	arr_delay ~ air_time + month + (1|carrier)	month	-0.2262	-0.2706	-0.1817	-0.2706	-0.1817
min	arr_delay ~ distance + air_time + (1|carrier)	(Intercept)	-3.0548	-7.0448	0.9351	-7.0448	0.9351
min	arr_delay ~ distance + air_time + (1|carrier)	air_time	0.6725	0.6608	0.6843	0.6608	0.6843
min	arr_delay ~ distance + air_time + (1|carrier)	distance	-0.0863	-0.0878	-0.0848	-0.0878	-0.0848
min	arr_delay ~ distance + month + (1|carrier)	(Intercept)	10.3305	6.2897	14.3714	6.2897	14.3714
min	arr_delay ~ distance + month + (1|carrier)	distance	-0.0013	-0.0016	-0.0011	-0.0016	-0.0011
min	arr_delay ~ distance + month + (1|carrier)	month	-0.2190	-0.2635	-0.1745	-0.2635	-0.1745

Compare confidence intervals

Determine whether or not the two methods of calculating the confidence intervals produce identical results.

```
identical(results_df$conf.low, results_df$`2.5 %`)
```

```
## [1] TRUE
```

```
identical(results_df$conf.high, results_df$`97.5 %`)
```

```
## [1] TRUE
```

Use a single pipeline

Since the results showed the two methods for producing confidence intervals are equivalent, we will now use a single pipeline, without running `confint.merMod()`, so we can compare performance.

```
# Start timer.
tic()

# Fit the models with lmer() using future_map() and extract the estimates and
# confidence intervals with broom.mixed::tidy() using map().
results_df <- formula_df %>%
  mutate(model = future_map(formula, lmer, data = data_df)) %>%
  mutate(est = map(model, broom.mixed::tidy, conf.int = TRUE,
                   conf.method = "Wald", conf.level = 0.95)) %>%
  select(-model) %>% unnest(cols = everything()) %>%
  filter(is.na(group)) %>%
  select(-group, -effect, -std.error, -statistic) %>%
  arrange(fgroup, formula, term)

# Stop timer.
toc()
```

```
## 18.294 sec elapsed
```

fgroup	formula	term	estimate	conf.low	conf.high
crude	arr_delay ~ air_time + (1|carrier)	(Intercept)	5.5096	0.6767	10.3424
crude	arr_delay ~ air_time + (1|carrier)	air_time	0.0085	0.0065	0.0105
crude	arr_delay ~ distance + (1|carrier)	(Intercept)	8.9182	4.8971	12.9394
crude	arr_delay ~ distance + (1|carrier)	distance	-0.0014	-0.0016	-0.0011
crude	arr_delay ~ month + (1|carrier)	(Intercept)	8.5815	4.1367	13.0263
crude	arr_delay ~ month + (1|carrier)	month	-0.2242	-0.2686	-0.1797
max	arr_delay ~ distance + air_time + month + (1|carrier)	(Intercept)	-2.7542	-6.7541	1.2457
max	arr_delay ~ distance + air_time + month + (1|carrier)	air_time	0.6717	0.6599	0.6835
max	arr_delay ~ distance + air_time + month + (1|carrier)	distance	-0.0862	-0.0877	-0.0847
max	arr_delay ~ distance + air_time + month + (1|carrier)	month	-0.0443	-0.0881	-0.0006
min	arr_delay ~ air_time + month + (1|carrier)	(Intercept)	6.9904	2.1389	11.8419
min	arr_delay ~ air_time + month + (1|carrier)	air_time	0.0086	0.0066	0.0106
min	arr_delay ~ air_time + month + (1|carrier)	month	-0.2262	-0.2706	-0.1817
min	arr_delay ~ distance + air_time + (1|carrier)	(Intercept)	-3.0548	-7.0448	0.9351
min	arr_delay ~ distance + air_time + (1|carrier)	air_time	0.6725	0.6608	0.6843
min	arr_delay ~ distance + air_time + (1|carrier)	distance	-0.0863	-0.0878	-0.0848
min	arr_delay ~ distance + month + (1|carrier)	(Intercept)	10.3305	6.2897	14.3714
min	arr_delay ~ distance + month + (1|carrier)	distance	-0.0013	-0.0016	-0.0011
min	arr_delay ~ distance + month + (1|carrier)	month	-0.2190	-0.2635	-0.1745

Display the results.

```r
# Note: Use kable_styling(full_width = TRUE) for HTML output.
results_df %>% knitr::kable(digits = 4) %>% kable_styling(font_size = 10)
```