MIXED EFFECT MODELS WITH LME4

Mixed effect models with lme4

In this instruction you will build intuition for **mixed effect models** using the lme4 package. You will:

1. Fit a mixed model on synthetic data and interpret fixed and random effects.
2. Modify the synthetic data so that it no longer depends on fuel_type, then check whether the effect is (incorrectly) reported as significant:
   - with a model *without* random effects,
   - with a model *with* random effects.

1. Setup: packages and synthetic data plan

Load lme4:

```
install.packages("lme4")
library(lme4)
install.packages("lmerTest")
library(lmerTest) # provides p-values in summaries
```

We simulate a pilot training / fuel consumption scenario:

- We fix the plane travel distance (so distance is constant; it is included only for completeness).
- The response is continuous: fuel_usage (e.g. liters).
- fuel_type is a factor with 3 levels (e.g. "A", "B", "C").
- pilot is a grouping variable for the random effect: each pilot has their own baseline fuel usage (random intercept).

2. Part I — Fuel usage depends on pilot and fuel type

2.1 Simulate data

```
set.seed(7)

n_pilots <- 35
fuel_types <- c("A", "B", "C")

# For each pilot, we record fuel usage for only a subset of fuel types.
# This makes the dataset more realistic (and helps show why random effects mat
pilot_ids <- paste0("P", seq_len(n_pilots))

rows <- lapply(pilot_ids, function(p) {
  k <- sample(c(1, 2), size = 1, prob = c(0.45, 0.55))  # each pilot has 1 or
  fts <- sample(fuel_types, size = k, replace = FALSE)
  data.frame(
    pilot = p,
    fuel_type = factor(fts, levels = fuel_types),
    distance = 1000,    # fixed distance (constant column)
    stringsAsFactors = FALSE
  )
})

fuel_df <- do.call(rbind, rows)

fuel_df$fuel_type <- factor(fuel_df$fuel_type, levels = fuel_types)

# True parameters (unknown to the learner)
beta0 <- 250                  # baseline fuel usage at fuel_type "A"
beta_fuel <- c(A = 0, B = -10, C = 18)  # differences vs "A"
sd_pilot <- 22                # random intercept variability across pilots
sd_eps <- 14                  # within-observation noise

pilot_intercept <- rnorm(n_pilots, mean = 0, sd = sd_pilot)
names(pilot_intercept) <- pilot_ids

fuel_df$fuel_usage <- beta0 +
  beta_fuel[as.character(fuel_df$fuel_type)] +
```

```
  pilot_intercept[fuel_df$pilot] +
  rnorm(nrow(fuel_df), mean = 0, sd = sd_eps)

fuel_df <- fuel_df[order(fuel_df$pilot), ]
head(fuel_df)
str(fuel_df)
```

2.2 Explore the dataset

```
summary(fuel_df$fuel_usage)
table(fuel_df$fuel_type)
table(fuel_df$pilot)

boxplot(fuel_usage ~ fuel_type, data = fuel_df,
        xlab = "fuel_type", ylab = "fuel_usage",
        main = "Fuel usage by fuel type (synthetic)")
```

2.3 Fit a mixed model with a random intercept

Model:

`fuel_usage ~ fuel_type + (1 | pilot)`

```
m1 <- lmer(fuel_usage ~ fuel_type + (1 | pilot), data = fuel_df)
summary(m1)
```

Interpretation:

- **Fixed effects** (`fuel_typeB`, `fuel_typeC`) describe the *average* difference in fuel usage between fuel types (with the baseline being fuel type "A").
- **Random effect** `(1 | pilot)` allows each pilot to have their own baseline offset.
  - `VarCorr(m1)` reports the estimated variance components.

Inspect random-effect variability:

```
VarCorr(m1)
```

Pilot-specific deviations from the population baseline:

```
head(ranef(m1))
```

Fixed effects:

```
fixef(m1)   # beta coefficients
```

Optional diagnostic checks:

```
isSingular(m1)  # should usually be FALSE for a well-behaved random effect
```

2.4 Predict and understand "marginal vs conditional"

In `lme4`, `predict()` by default uses the fitted random effects (i.e. it is *conditional on the pilot*).

```
fuel_df$pred_conditional <- predict(m1)
```

If you want predictions based only on fixed effects (i.e. *marginal over pilots*), use `re.form = NA`:

```
fuel_df$pred_marginal <- predict(m1, re.form = NA)
```

Compare a few rows:

```
head(fuel_df[, c("pilot", "fuel_type", "fuel_usage", "pred_conditional", "pred
```

3. Part II — Remove fuel_type dependence and test significance

Now we modify the response so that `fuel_type` no longer affects `fuel_usage`. We keep the same pilot structure and measurement layout, so pilot-to-pilot variation still exists.

3.1 Create data with no fuel_type effect

Create new data, removing the dependence on `fuel_type`.

Fit a model *without* random effects (`lm`). Then fit a model *with* random effects (`lmer`). Compare p-values reported by `summary` and discuss.

4. Wrap-up

This instruction demonstrated two core ideas:

- A mixed model like `fuel_usage ~ fuel_type + (1 | pilot)` separates:
  - **between-pilot variation** (random intercept),
  - **population-level fuel type differences** (fixed effects).
- If the data are clustered by pilot, ignoring that structure (using `lm()` only) can lead to misleading uncertainty and apparent significance.