```r
#Packages
library(tidyr)
library(dplyr)
library(lme4)
library(car)
library(emmeans)
library(multcomp)
library(multcompView)
library(ggplot2)
library(grid)


# F2 Hatching rate ------------------------------------------------------------
hat_df <- read.csv("hat.csv")

hat_df <- hat_df %>%
  mutate(
    gmum    = factor(gmum, levels = c("Young", "Old")),
    mum     = factor(mum,  levels = c("Young", "Old")),
    mumdiet = factor(mumdiet),
    Diet    = factor(Diet)
  )

hat_summary <- hat_df %>%
  group_by(gmum, mum, mumdiet) %>%
  summarise(
    n_total = n(),
    n_survived = sum(hatch == 1, na.rm = TRUE),
    surv_percent = mean(hatch == 1, na.rm = TRUE) * 100
  ) %>%
  ungroup()

View(hat_summary)

#Model testing for hatching

hat_model <- glmer(
  hatch ~ gmum * mum * mumdiet + (1 | gID/mID),
  data = hat_df,
  family = binomial(link = "logit")
)

summary(hat_model)

Anova(hat_model, type = "III")

performance::check_overdispersion(hat_model)

#Model coefficients
ht_coef <- broom.mixed::tidy(
  hat_model,
  effects = "fixed",
  conf.int = TRUE,
  conf.method = "Wald"
) %>%
  dplyr::mutate(
    term = dplyr::recode(
      term,
      "(Intercept)" = "Intercept",
      "gmum1" = "Grandmaternal age (F0)",
      "mum1" = "Maternal age (F1)",
      "mumdiet1" = "Maternal diet (F1)"
    )
  ) %>%
  dplyr::select(
    term, estimate, std.error,
    conf.low, conf.high, statistic
  )
```

```
View(ht_coef)

# F2 life history traits ---------------------------------------------------
full_df_f2 <- read.csv("f2_lifehistories.csv")

full_df_f2 <- full_df_f2 %>%
  mutate(
    gmum    = factor(gmum, levels = c("Young", "Old")),
    mum     = factor(mum,  levels = c("Young", "Old")),
    mumdiet = factor(mumdiet),
    diet    = factor(diet)
  )

# Developmental time -------------------------------------------------------
#Model testing
hist(full_df_f2$ea)
options(contrasts = c("contr.sum", "contr.poly"))

ea_model1 <- lmer(log(ea) ~ gmum * mum * mumdiet * diet + (1 | gID/mID),
                  data = full_df_f2)
summary(ea_model1)
ea_model2 <- lm(log(ea) ~ gmum * mum * mumdiet * diet,
                  data = full_df_f2)
summary(ea_model2)
ea_model3 <- lm(log(ea) ~ gmum * mum + mumdiet + diet,
                  data = full_df_f2)
summary(ea_model3)

ea_model1 <- glmer(ea ~ gmum * mum + mumdiet + diet + (1 | gID/mID),
                  data = full_df_f2, family = poisson(link = "log"))
summary(ea_model1)

Anova(ea_model1, type = "III")
Anova(ea_model2, type = "III")
Anova(ea_model3, type = "III")

#No qualitative differences, model1 is used.

qqnorm(resid(ea_model1))
qqline(resid(ea_model1))
plot(ea_model1)

cld(emmeans(ea_model1, ~  gmum | mum),
    Letters = letters,
    adjust = "sidak")
plot(emmeans(ea_model1, ~ gmum | mum))

emm <- emmeans(ea_model1, ~ gmum * mum * mumdiet * diet)
pairs(emm, adjust = "sidak")

cld(emm,
    Letters = letters,
    adjust = "sidak")

#Model coefficients
ea_coef <- broom.mixed::tidy(
  ea_model1,
  effects = "fixed",
  conf.int = TRUE,
  conf.method = "Wald"
) %>%
  dplyr::mutate(
    term = dplyr::recode(
      term,
      "(Intercept)" = "Intercept",
      "gmum1" = "Grandmaternal age (F0)",
      "mum1" = "Maternal age (F1)",
      "mumdiet1" = "Maternal diet (F1)",
      "diet1" = "Offspring diet (F2)"
```

```
    )
  ) %>%
  dplyr::select(
    term, estimate, std.error,
    conf.low, conf.high, statistic
  )

View(ea_coef)

# Dev-SE ------------------------------------------------------------------

n_df <- full_df_f2 %>%
  group_by(gmum, f1_group, diet) %>%
  summarise(
    n = sum(!is.na(ea)),
    .groups = "drop"
  )

full_df_f2 <- full_df_f2 %>%
  mutate(
    f1_group = interaction(
      mum,
      mumdiet,
      sep = " | ",
      lex.order = TRUE
    ),
    f1_group = factor(
      f1_group,
      levels = c(
        "Young | Full",
        "Young | Restricted",
        "Old | Full",
        "Old | Restricted"
      )
    )
  )

dev_emm <- emmeans(
  ea_model1,
  ~ mum * mumdiet * diet | gmum
)

dev_emm_df <- as.data.frame(
  summary(
    dev_emm,
    type = "response",
    infer = c(TRUE, FALSE)
  )
)

# Identify estimate and confidence interval columns
estimate_col <- intersect(
  c("rate", "response"),
  names(dev_emm_df)
)[1]

lower_col <- grep(
  "LCL",
  names(dev_emm_df),
  value = TRUE
)[1]

upper_col <- grep(
  "UCL",
  names(dev_emm_df),
  value = TRUE
)[1]

# Check that the required columns were found
```

```
if (is.na(estimate_col) ||
    is.na(lower_col) ||
    is.na(upper_col)) {
  stop("Could not identify EMM estimate or confidence interval columns. Check
names(dev_emm_df).")
}

dev_emm_df <- dev_emm_df %>%
  mutate(
    estimate = .data[[estimate_col]],
    lower = .data[[lower_col]],
    upper = .data[[upper_col]]
  ) %>%
  mutate(
    f1_group = interaction(
      mum,
      mumdiet,
      sep = " | ",
      lex.order = TRUE
    ),
    f1_group = factor(
      f1_group,
      levels = c(
        "Young | Full",
        "Young | Restricted",
        "Old | Full",
        "Old | Restricted"
      )
    )
  )

pd <- position_dodge(width = 0.8)

dev_plot <- ggplot(
  full_df_f2,
  aes(
    x = f1_group,
    y = ea,
    fill = diet,
    color = diet
  )
) +
  geom_violin(
    position = pd,
    trim = FALSE,
    alpha = 0.50,
    width = 0.95,
    color = NA
  ) +
  geom_point(
    position = position_jitterdodge(
      jitter.width = 0.22,
      jitter.height = 0,
      dodge.width = 0.8
    ),
    size = 1.1,
    alpha = 1,
    stroke = 0.3
  ) +
  geom_errorbar(
    data = dev_emm_df,
    aes(
      x = f1_group,
      y = estimate,
      ymin = lower,
      ymax = upper,
      color = diet,
      group = diet
    ),
```

```
  inherit.aes = FALSE,
  position = pd,
  width = 0.3,
  linewidth = 0.5,
  color = "black"
) +
geom_point(
  data = dev_emm_df,
  aes(
    x = f1_group,
    y = estimate,
    color = diet,
    group = diet
  ),
  inherit.aes = FALSE,
  position = pd,
  shape = 18,
  size = 2.5,
  color = "black"
) +
geom_text(
  data = n_df,
  aes(
    x = f1_group,
    y = -Inf,
    label = paste0("", n),
    group = diet
  ),
  inherit.aes = FALSE,
  position = pd,
  vjust = -0.8,
  size = 4.2,
  color = "black"
) +
facet_grid(
  . ~ gmum,
  labeller = labeller(
    gmum = as_labeller(
      c(
        Young = "F[0] ~ age: ~ Young",
        Old = "F[0] ~ age: ~ Old"
      ),
      label_parsed
    )
  )
) +
scale_x_discrete(
  labels = c(
    "Young | Full" = "Y/F",
    "Young | Restricted" = "Y/R",
    "Old | Full" = "O/F",
    "Old | Restricted" = "O/R"
  )
) +
scale_fill_manual(
  values = c(
    "Full" = "#6A9CAF",
    "Restricted" = "#DD7927"
  )
) +
scale_color_manual(
  values = c(
    "Full" = "#6A9CAF",
    "Restricted" = "#DD7927"
  )
) +
labs(
  x = expression(F[1] ~ "age and diet"),
  y = "Developmental time (days)",
```

```
    fill = expression(F[2] ~ "diet"),
    color = expression(F[2] ~ "diet")
  ) +
  theme_classic2() +
  theme(
    text = element_text(
      family = "Helvetica",
      size = 26
    ),
    plot.margin = margin(
      0.5, 0.5, 0.5, 0.5,
      "cm"
    ),
    axis.title.x = element_text(
      margin = margin(t = 10)
    ),
    axis.title.y = element_text(
      margin = margin(r = 10)
    ),
    axis.text.x = element_text(
      size = 19
    ),
    panel.spacing.x = unit(0.8, "cm"),
    strip.background = element_blank(),
    strip.text = element_text(
      size = 21,
      face = "bold"
    ),
    legend.title = element_text(size = 19),
    legend.text = element_text(size = 17)
  )

print(dev_plot)

# Size ----------------------------------------------------------------------
#Model testing
hist(full_df_f2$size)

size_model1 <- lmer(size ~ gmum * mum * mumdiet * diet + (1 | gID/mID),
                    data = full_df_f2)
summary(size_model1)

size_model2 <- lm(size ~ gmum * mum * mumdiet * diet,
                  data = full_df_f2)
summary(size_model2)

Anova(size_model1, type = "III")
Anova(size_model2, type = "III")

#No qualitative differences, model1 is used.

qqnorm(resid(size_model1))
qqline(resid(size_model1))
plot(size_model1)

cld(emmeans(size_model1, ~  mum | mumdiet | diet),
    Letters = letters,
    adjust = "sidak")
plot(emmeans(size_model1, ~ diet | mumdiet | mum))

emm <- emmeans(size_model1, ~ mum | mumdiet | diet)
pairs(emm, adjust = "sidak")

#Model coefficients
size_coef <- broom.mixed::tidy(
  size_model1,
  effects = "fixed",
  conf.int = TRUE,
  conf.method = "Wald"
```

```
) %>%
  dplyr::mutate(
    term = dplyr::recode(
      term,
      "(Intercept)" = "Intercept",
      "gmum1" = "Grandmaternal age (F0)",
      "mum1" = "Maternal age (F1)",
      "mumdiet1" = "Maternal diet (F1)",
      "diet1" = "Offspring diet (F2)"
    )
  ) %>%
  dplyr::select(
    term, estimate, std.error,
    conf.low, conf.high, statistic
  )

View(size_coef)

# Size-SE ----------------------------------------------------------------

full_df_f2 <- full_df_f2 %>%
  mutate(
    f1_group = interaction(
      mum,
      mumdiet,
      sep = " | ",
      lex.order = TRUE
    ),
    f1_group = factor(
      f1_group,
      levels = c(
        "Young | Full",
        "Young | Restricted",
        "Old | Full",
        "Old | Restricted"
      )
    )
  )

n_df <- full_df_f2 %>%
  group_by(gmum, f1_group, diet) %>%
  summarise(
    n = sum(!is.na(size)),
    .groups = "drop"
  )

size_emm <- emmeans(
  size_model1,
  ~ mum * mumdiet * diet | gmum
)

size_emm_df <- as.data.frame(
  summary(
    size_emm,
    infer = c(TRUE, FALSE)
  )
)

estimate_col <- "emmean"

lower_col <- grep(
  "lower.CL",
  names(size_emm_df),
  value = TRUE
)[1]

upper_col <- grep(
  "upper.CL",
  names(size_emm_df),
```

```
  value = TRUE
)[1]

if (is.na(lower_col) || is.na(upper_col)) {
  stop("Could not identify confidence interval columns. Check names(size_emm_df).")
}

size_emm_df <- size_emm_df %>%
  mutate(
    estimate = .data[[estimate_col]],
    lower = .data[[lower_col]],
    upper = .data[[upper_col]]
  ) %>%
  mutate(
    f1_group = interaction(
      mum,
      mumdiet,
      sep = " | ",
      lex.order = TRUE
    ),
    f1_group = factor(
      f1_group,
      levels = c(
        "Young | Full",
        "Young | Restricted",
        "Old | Full",
        "Old | Restricted"
      )
    )
  )

pd <- position_dodge(width = 0.8)

size_plot <- ggplot(
  full_df_f2,
  aes(
    x = f1_group,
    y = size,
    fill = diet,
    color = diet
  )
) +
  geom_violin(
    position = pd,
    trim = FALSE,
    alpha = 0.50,
    width = 0.95,
    color = NA
  ) +
  geom_point(
    position = position_jitterdodge(
      jitter.width = 0.22,
      jitter.height = 0,
      dodge.width = 0.8
    ),
    size = 1.1,
    alpha = 1,
    stroke = 0.3
  ) +
  geom_errorbar(
    data = size_emm_df,
    aes(
      x = f1_group,
      y = estimate,
      ymin = lower,
      ymax = upper,
      group = diet
    ),
    inherit.aes = FALSE,
```

```
  position = pd,
  width = 0.3,
  linewidth = 0.5,
  color = "black"
) +
geom_point(
  data = size_emm_df,
  aes(
    x = f1_group,
    y = estimate,
    group = diet
  ),
  inherit.aes = FALSE,
  position = pd,
  shape = 18,
  size = 2.5,
  color = "black"
) +
geom_text(
  data = n_df,
  aes(
    x = f1_group,
    y = -Inf,
    label = paste0("", n),
    group = diet
  ),
  inherit.aes = FALSE,
  position = pd,
  vjust = -0.8,
  size = 4.2,
  color = "black"
) +
facet_grid(
  . ~ gmum,
  labeller = labeller(
    gmum = as_labeller(
      c(
        Young = "F[0] ~ age: ~ Young",
        Old = "F[0] ~ age: ~ Old"
      ),
      label_parsed
    )
  )
) +
scale_x_discrete(
  labels = c(
    "Young | Full" = "Y/F",
    "Young | Restricted" = "Y/R",
    "Old | Full" = "O/F",
    "Old | Restricted" = "O/R"
  )
) +
scale_fill_manual(
  values = c(
    "Full" = "#6A9CAF",
    "Restricted" = "#DD7927"
  )
) +

scale_color_manual(
  values = c(
    "Full" = "#6A9CAF",
    "Restricted" = "#DD7927"
  )
) +
labs(
  x = expression(F[1] ~ "age and diet"),
  y = "Dorsal plate length (\u03bcm)",
  fill = expression(F[2] ~ "diet"),
```

```
    color = expression(F[2] ~ "diet")
  ) +
  theme_classic2() +

  theme(
    text = element_text(
      family = "Helvetica",
      size = 26
    ),

    plot.margin = margin(
      0.5, 0.5, 0.5, 0.5,
      "cm"
    ),

    axis.title.x = element_text(
      margin = margin(t = 10)
    ),

    axis.title.y = element_text(
      margin = margin(r = 10)
    ),

    axis.text.x = element_text(
      size = 19
    ),

    panel.spacing.x = unit(0.8, "cm"),

    strip.background = element_blank(),

    strip.text = element_text(
      size = 21,
      face = "bold"
    ),

    legend.title = element_text(size = 19),
    legend.text = element_text(size = 17)
  )

print(size_plot)

# Survival at Day 15 ------------------------------------------------------

sur_sum <- full_df_f2 %>%
  group_by(gmum, mum, mumdiet, diet) %>%
  summarise(
    surv_prop = mean(sur, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(sur_perc = surv_prop * 100)

View(sur_sum)

#Model testing

ovi_model1 <- glmer(
  sur ~ gmum * mum * mumdiet * diet + (1 | gID/mID),
  data = full_df_f2,
  family = binomial)

ovi_model2 <- glm(
  sur ~ gmum * mum * mumdiet * diet,
  data = full_df_f2,
  family = binomial)

summary(ovi_model1)
summary(ovi_model2)
```

```r
#Model coefficients
ovir_coef <- broom.mixed::tidy(
  ovi_model1,
  effects = "fixed",
  conf.int = TRUE,
  conf.method = "Wald"
) %>%
  dplyr::mutate(
    term = dplyr::recode(
      term,
      "(Intercept)" = "Intercept",
      "gmum1" = "Grandmaternal age (F0)",
      "mum1" = "Maternal age (F1)",
      "mumdiet1" = "Maternal diet (F1)"
    )
  ) %>%
  dplyr::select(
    term, estimate, std.error,
    conf.low, conf.high, statistic
  )

View(ovir_coef)

Anova(ovi_model1, type = "III")
Anova(ovi_model2, type = "III")

#No qualitative differences, model1 is used.

performance::check_overdispersion(ovi_model1)

# Oviposition ------------------------------------------------------------
#Model testing
hist(full_df_f2_fec$fec)

fec_model1 <- glmer(fec ~ gmum * mum * mumdiet * diet + (1 | gID/mID),
                    data = full_df_f2_fec,
                    family = poisson(link = "log"))
summary(fec_model1)

fec_model2 <- glm(
  fec ~ gmum * mum * mumdiet * diet,
  data = full_df_f2_fec,
  family = poisson(link = "log")
)
summary(fec_model2)

Anova(fec_model1, type = "III")
Anova(fec_model2, type = "III")

#No qualitative differences, model1 is used.

performance::check_overdispersion(fec_model1)

cld(emmeans(fec_model1, ~ diet),
    Letters = letters,
    adjust = "sidak")
plot(emmeans(fec_model1, ~ diet))

emm <- emmeans(fec_model1, ~ diet)
pairs(emm, adjust = "sidak")

#Model coefficients
ovi_coef <- broom.mixed::tidy(
  fec_model1,
  effects = "fixed",
  conf.int = TRUE,
  conf.method = "Wald"
) %>%
  dplyr::mutate(
```

```
    term = dplyr::recode(
      term,
      "(Intercept)" = "Intercept",
      "gmum1" = "Grandmaternal age (F0)",
      "mum1" = "Maternal age (F1)",
      "mumdiet1" = "Maternal diet (F1)",
      "diet1" = "Offspring diet (F2)"
    )
  ) %>%
  dplyr::select(
    term, estimate, std.error,
    conf.low, conf.high, statistic
  )

View(ovi_coef)

# Ovi-SE ------------------------------------------------------------------

full_df_f2_fec <- full_df_f2 %>%
  filter(!is.na(fec)) %>%
  mutate(
    f1_group = interaction(
      mum,
      mumdiet,
      sep = " | ",
      lex.order = TRUE
    ),
    f1_group = factor(
      f1_group,
      levels = c(
        "Young | Full",
        "Young | Restricted",
        "Old | Full",
        "Old | Restricted"
      )
    )
  )

n_df <- full_df_f2_fec %>%
  group_by(gmum, f1_group, diet) %>%
  summarise(
    n = sum(!is.na(fec)),
    .groups = "drop"
  )

fec_emm <- emmeans(
  fec_model1,
  ~ mum * mumdiet * diet | gmum
)

fec_emm_df <- as.data.frame(
  confint(
    fec_emm,
    type = "response"
  )
)

estimate_col <- intersect(
  c("response", "rate", "emmean"),
  names(fec_emm_df)
)[1]

lower_col <- grep(
  "lower.CL|asymp.LCL",
  names(fec_emm_df),
  value = TRUE
)[1]

upper_col <- grep(
```

```
  "upper.CL|asymp.UCL",
  names(fec_emm_df),
  value = TRUE
)[1]

if (
  is.na(estimate_col) ||
  is.na(lower_col) ||
  is.na(upper_col)
) {
  print(names(fec_emm_df))
  stop("Could not identify EMM or confidence interval columns.")
}

fec_emm_df <- fec_emm_df %>%
  mutate(
    estimate = .data[[estimate_col]],
    lower = .data[[lower_col]],
    upper = .data[[upper_col]],

    f1_group = interaction(
      mum,
      mumdiet,
      sep = " | ",
      lex.order = TRUE
    ),

    f1_group = factor(
      f1_group,
      levels = c(
        "Young | Full",
        "Young | Restricted",
        "Old | Full",
        "Old | Restricted"
      )
    )
  )

pd <- position_dodge(width = 0.8)

fec_plot <- ggplot(
  full_df_f2_fec,
  aes(
    x = f1_group,
    y = fec,
    fill = diet,
    color = diet
  )
) +
  geom_violin(
    position = pd,
    trim = FALSE,
    alpha = 0.50,
    width = 0.95,
    color = NA
  ) +
  geom_point(
    position = position_jitterdodge(
      jitter.width = 0.22,
      jitter.height = 0,
      dodge.width = 0.8
    ),
    size = 1.1,
    alpha = 1,
    stroke = 0.3
  ) +
  geom_errorbar(
    data = fec_emm_df,
    aes(
```

```
    x = f1_group,
    y = estimate,
    ymin = lower,
    ymax = upper,
    group = diet
  ),
  inherit.aes = FALSE,
  position = pd,
  width = 0.3,
  linewidth = 0.5,
  color = "black"
) +
geom_point(
  data = fec_emm_df,
  aes(
    x = f1_group,
    y = estimate,
    group = diet
  ),
  inherit.aes = FALSE,
  position = pd,
  shape = 18,
  size = 2.5,
  color = "black"
) +
geom_text(
  data = n_df,
  aes(
    x = f1_group,
    y = -Inf,
    label = paste0("", n),
    group = diet
  ),
  inherit.aes = FALSE,
  position = pd,
  vjust = -0.8,
  size = 4.2,
  color = "black"
) +
facet_grid(
  . ~ gmum,
  labeller = labeller(
    gmum = as_labeller(
      c(
        Young = "F[0] ~ age: ~ Young",
        Old = "F[0] ~ age: ~ Old"
      ),
      label_parsed
    )
  )
) +
scale_x_discrete(
  labels = c(
    "Young | Full" = "Y/F",
    "Young | Restricted" = "Y/R",
    "Old | Full" = "O/F",
    "Old | Restricted" = "O/R"
  )
) +
scale_fill_manual(
  values = c(
    "Full" = "#6A9CAF",
    "Restricted" = "#DD7927"
  )
) +

scale_color_manual(
  values = c(
    "Full" = "#6A9CAF",
```

```
      "Restricted" = "#DD7927"
    )
  ) +
  labs(
    x = expression(F[1] ~ "age and diet"),
    y = "Oviposition (eggs per female)",
    fill = expression(F[2] ~ "diet"),
    color = expression(F[2] ~ "diet")
  ) +
  theme_classic2() +

  theme(
    text = element_text(
      family = "Helvetica",
      size = 26
    ),

    plot.margin = margin(
      0.5, 0.5, 0.5, 0.5,
      "cm"
    ),

    axis.title.x = element_text(
      margin = margin(t = 10)
    ),

    axis.title.y = element_text(
      margin = margin(r = 10)
    ),

    axis.text.x = element_text(
      size = 19
    ),

    panel.spacing.x = unit(0.8, "cm"),

    strip.background = element_blank(),

    strip.text = element_text(
      size = 21,
      face = "bold"
    ),

    legend.title = element_text(size = 19),
    legend.text = element_text(size = 17)
  )

print(fec_plot)
```