Statistical Analyses for Faiad et. al (2023) PLoS One

Temperature affects predation of schistosome-competent snails by a novel invader, the marbled crayfish *Procambarus virginalis*

Code by: Maurice Goodman, Sara Faiad, & Giulio De Leo

2023

Data

Reading in and combining datasets, and filtering out the control data, we expect a total of 960 trials in the data:

- 600 predation trials
  - Round 1: 6 experiments * 6 (12 hr) trials * 5 temperature * 2 species = 360
  - Round 2: 2 experiments * 6 (12 hr) trials * 5 temperature * 2 species * 2 infection status = 240
- 360 control trials
  - Round 1: 2 experiments * 6 (12 hr) trials * 5 temperature * 2 species = 120
  - Round 2: 2 experiments * 6 (12 hr) trials * 5 temperature * 2 species * 2 infection status = 240

Removing data at time 0 (i.e., the start of the first trial in each experimental run), and excluding pilot trial data ("pilot_ex_un_v_in_1"), we have 960 observations (600 experimental, 360 control), as expected. We remove an additional 70 observations (10 control, 60 experimental):

- All 10 removed control trials, and 42 removed experimental, are marked "dodgy" for various reasons (usually too many snails or snails were missing)
- 16 experimental trials where crayfish were observed to be molting
- 2 experimental trials where consumption was negative, but had not been marked "dodgy" - the notes on these trials, however, indicate that they likely should have been marked "dodgy" but weren't.

This leaves 350 control and 540 experimental observations.

```
# Read in data
snail_trials <- read.csv("data/all_trials_snail_digital.csv")
cray_trials <- read.csv("data/all_trials_crayfish_digital.csv")

# Generate an id field to link the two sheets
snail_trials <- snail_trials |> unite(craytrialid, c(week, crayfish_id), remove = FALSE)
cray_trials <- cray_trials |> unite(craytrialid, c(week, crayfish_id), remove = FALSE)

# Combine datasets
# Remove data with more snails than density, and those that were molting
# There are also 2 values with consumption < 0...
all_data <- cray_trials |>
  dplyr::select(craytrialid, weight, berried) |>
  right_join(snail_trials, by = "craytrialid", multiple = "all") |>
  filter(time != 0) |>
  dplyr::select(-c(start_time, end_time, collector_name, notes)) |>
  mutate(
    consumption_all = snail_density - snails_remaining - dead_snails,
```

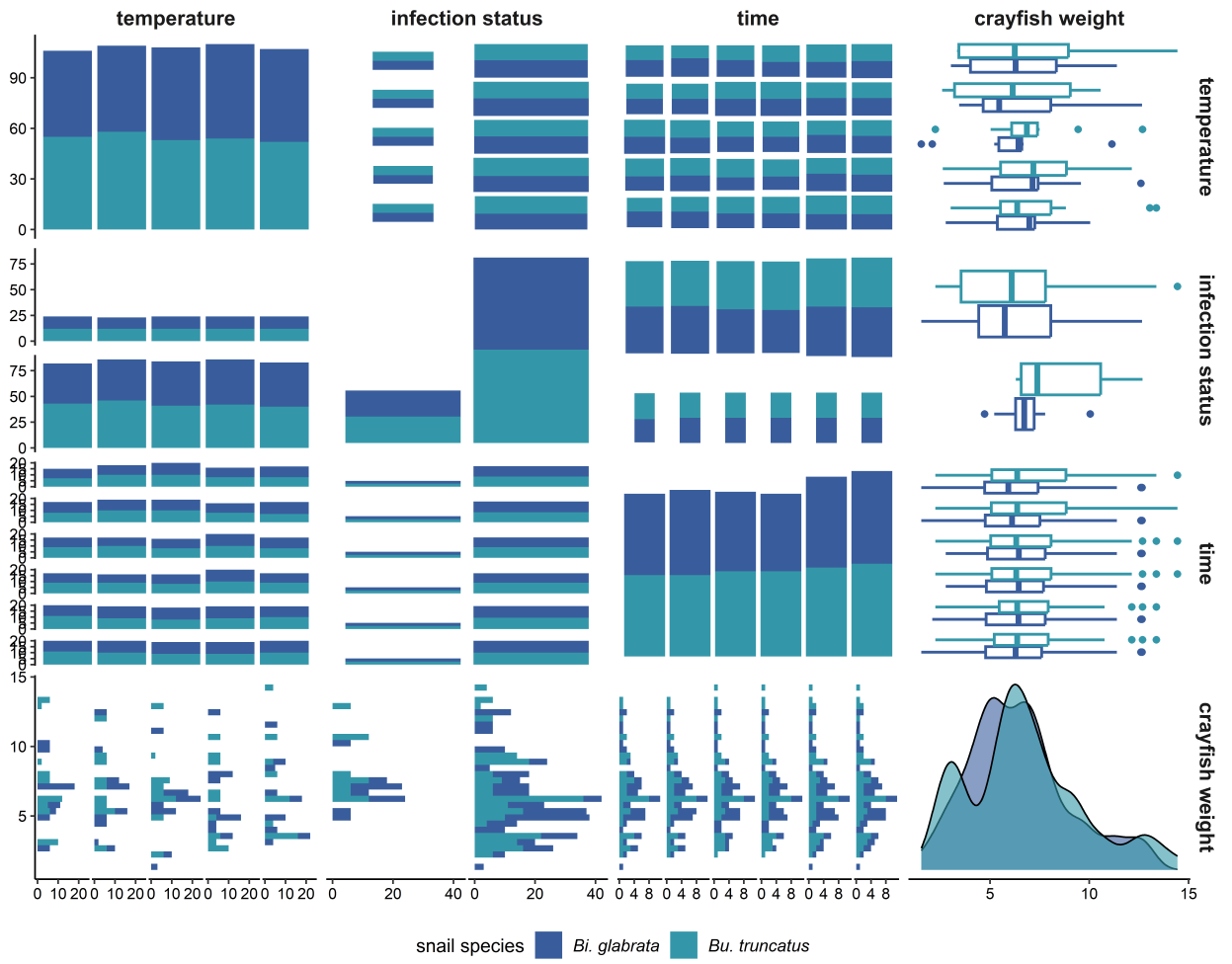
```
    p_consumed = consumption_all / (snail_density - dead_snails),
    condition = ifelse(grepl("control", condition), "control", "experimental"),
    snail_species = factor(ifelse(snail_species == "biomph", "Bi. glabrata", "Bu. truncatus")),
    infection_status = factor(infection_status),
    temp = factor(temp, ordered = TRUE),
    time = factor(time, ordered = TRUE)
  ) |>
  filter(week != "pilot_ex_un_v_in_1" & dodgey_remove != "Y" &
           molt != "Y" & consumption_all >= 0)

## Experimental data only
exp_data <- all_data |> filter(condition == "experimental")
```

Generalized pairs plot

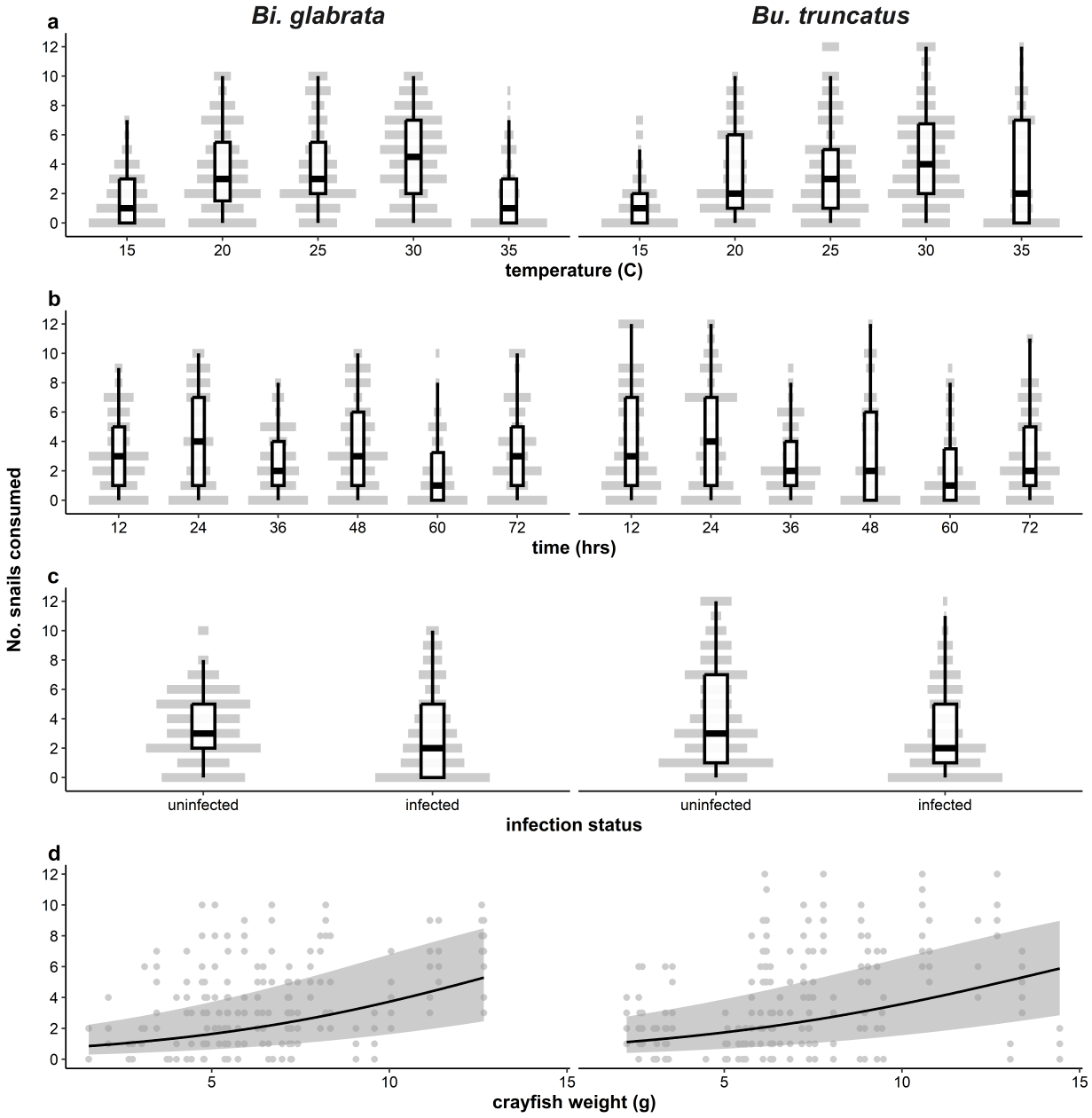
The data are more or less balanced across experimental treatments (temperature, species, time, and infection status), except that there are many fewer infected than uninfected snails since only uninfected snails were used in the first round of experiments:

```
exp_data |>
  dplyr::select(
    temperature = temp, `infection status` = infection_status, time,
    `crayfish weight` = weight
  ) |>
  ggpairs(
    aes(color = exp_data$snail_species),
    lower = list(combo = ggally_facethist2),
    diag = list(continuous = wrap("densityDiag", alpha = 0.6)),
    upper = list(combo = ggally_boxplot),
    legend = c(1, 1)
  ) +
  scale_fill_manual(values = wght_col[2:3]) +
  scale_color_manual(values = wght_col[2:3]) +
  theme(
    legend.position = "bottom",
    legend.text = element_text(face = "italic"),
    axis.text = element_text(size = 9)
  ) +
  labs(fill = "snail species")
```



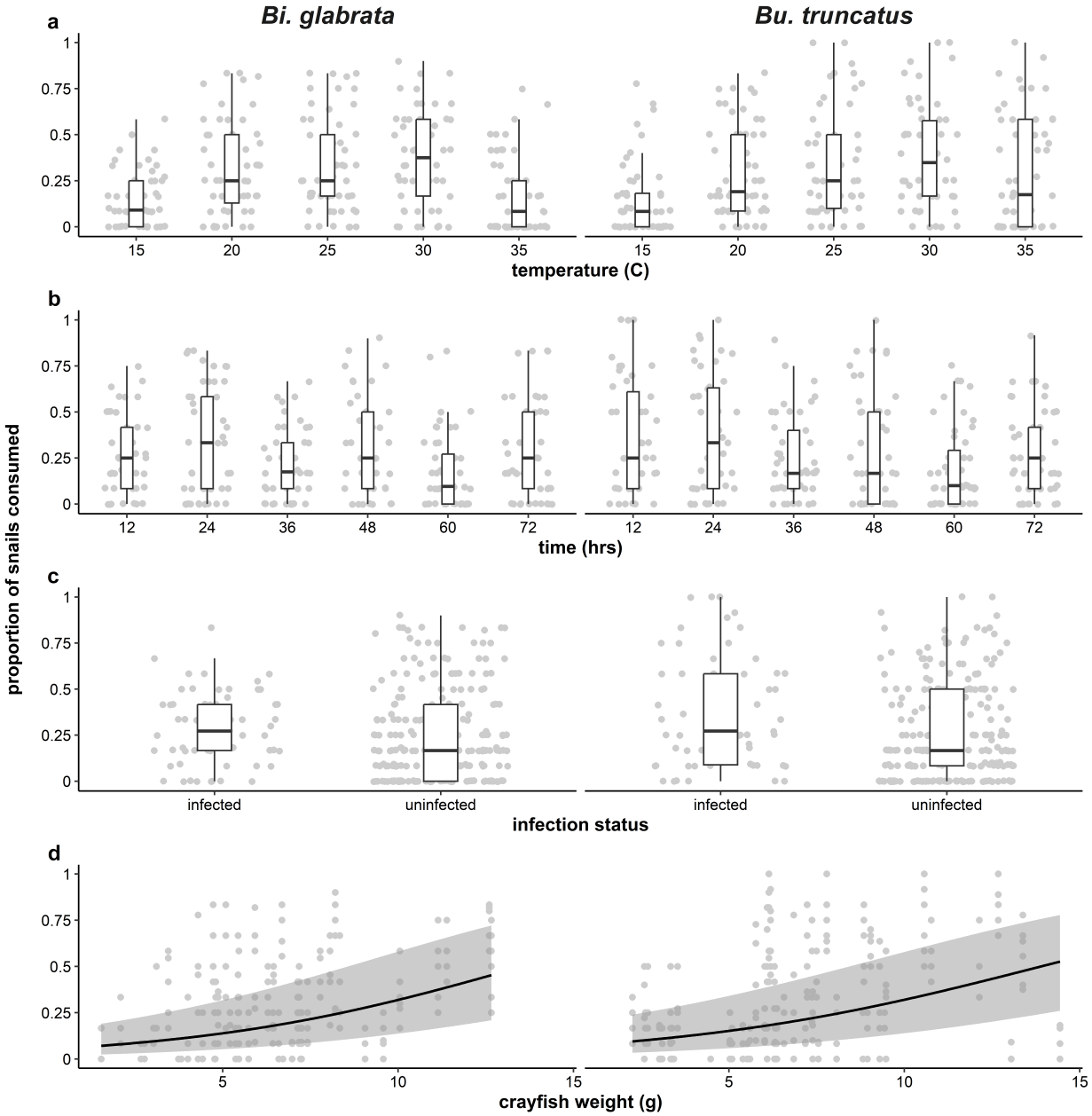
Raw consumption data

Plotting snail consumption against covariates, with centered histograms behind boxplots for discrete covariates, and points for the weight plot. The regression line for the number of snails consumed as a function of crayfish weight is estimated with a GLMM whose random effects structure mirrors the model described below.



Consumption as proportions

Here, plotting not the number consumed, but the proportion consumed (out of the initial snail density, minus the number of snails dead from other causes):



Generalized linear model

Model structure

In each trial, there are 12 snails available for consumption (minus those that died), meaning that the response is (0,12) bounded and the response distribution is binomial. The experiment has three main factors - snail species, infection status, and temperature - with repeated observations of the same crayfish across time steps (12, 24, 36, 48, 60, and 72 hours) and across experimental runs ("week"). We'll treat snail species, infected status, temperature, and time steps as fixed effects (with temperature and time as categorical, rather than continuous predictors), and include crayfish weight as well. All second and third order interactions among the experimental factors are included, as well as an interaction between temperature and weight.

The model is:

$$N_{c,i} \sim \text{Binomial}(p_i, N_{s,i} - N_{d,i})$$
$$\text{logit}(p_i) \sim \alpha + \beta X_i + \epsilon_c + \epsilon_w + \epsilon_{cw}$$

where:

- The response y_i is the number of snails eaten (i.e., *Consumption_all* variable), which is distributed according to a binomial distribution with n equal to the initial number of snails ($N_{s,i}$, always 12) minus the number of snails that died $N_{d,i}$.
- The linear predictors of the log-odds that any given snail is eaten $\text{logit}(p_i)$ are (i) an intercept α, (ii) some predictors X (detailed above), and (iii) random intercepts for each crayfish and experimental run / week (ϵ_c and ϵ_w) and an interaction which allows crayfish intercepts to vary by week (ϵ_{cw}).

Model fitting

Fitting a generalized linear mixed effects model (GLMM) using `glmmTMB`:

```
glmm <- glmmTMB(
  cbind(consumption_all, snail_density - dead_snails - consumption_all) ~
    (snail_species + infection_status + temp + time)^3  + weight * temp +
    (1 | crayfish_id * week),
  family = binomial(link = "logit"),
  data = exp_data
)
```

Main effects

Wald Chi-square tests for the main effects, showing significant effects of temperature, time, and weight. These are "Type II" tests, which according to the `car` documentation "are calculated according to the principle of marginality, testing each term after all others, except ignoring the term's higher-order relatives" - i.e., the temperature effect is a test of the overall effect of temperature in the model, including any interactions. None of the higher-order interaction terms are significant.

```
tidy_anova(glmm) # Defined in setup block at start of Rmd
```

	Chisq	Df	Pr(>Chisq)	
snail_species	0.023	1	0.881	
infection_status	0.422	1	0.516	
temp	29.056	4	<0.001	*
time	97.099	5	<0.001	*
weight	18.948	1	<0.001	*
snail_species:infection_status	0.000	1	0.995	
snail_species:temp	3.098	4	0.542	
snail_species:time	7.244	5	0.203	
infection_status:temp	2.059	4	0.725	
infection_status:time	3.831	5	0.574	
temp:time	16.228	20	0.702	
temp:weight	1.021	4	0.907	
snail_species:infection_status:temp	2.706	4	0.608	
snail_species:infection_status:time	10.049	5	0.074	
snail_species:temp:time	20.826	20	0.407	
infection_status:temp:time	19.352	20	0.499	

Predicted probability of consumption

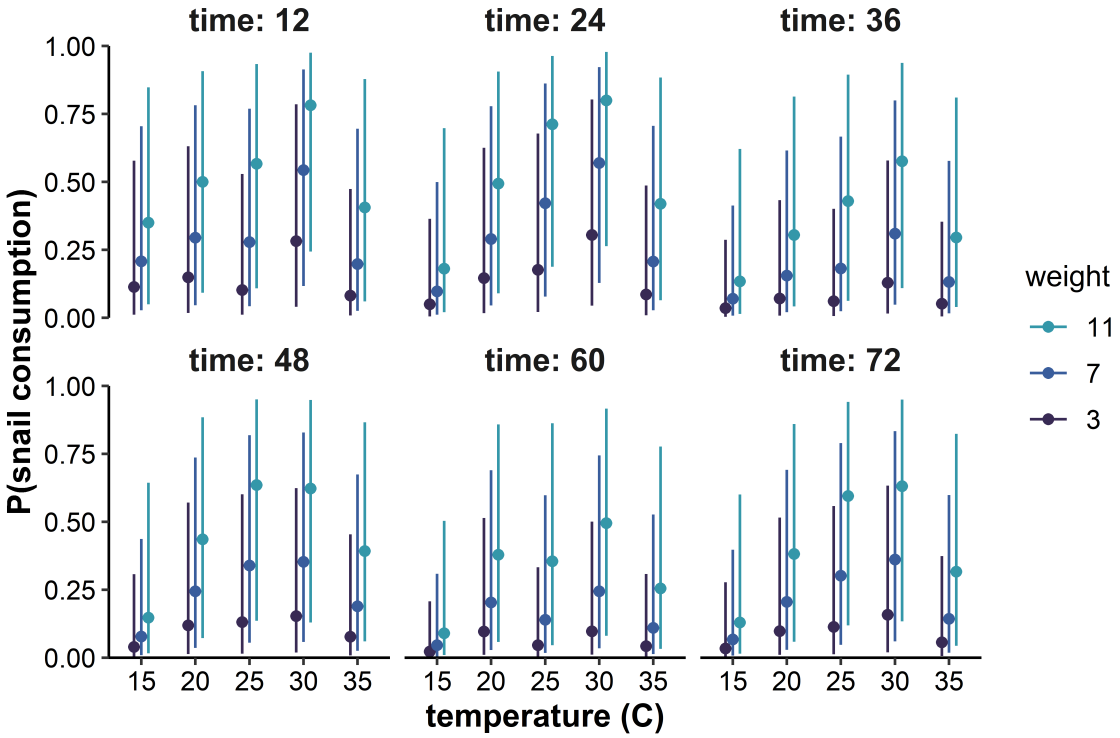
One way to plot the model estimates is by obtaining predictions on the response scale, i.e. in terms of the probability of consumption for each snail. Here we're plotting data for uninfected *Bu. truncatus*, but these should look similar for the others. Estimates are plotted for snails at weights of 3, 7, and 11 grams , which are approximately the 10% quantile, mean, and 90% quantile from the observed data, respectively.

```
# Combinations of variable levels to predict on
newdata <- expand.grid(
  snail_species = unique(exp_data$snail_species),
  infection_status = unique(exp_data$infection_status),
  temp = sort(unique(exp_data$temp)),
  time = sort(unique(exp_data$time)),
  crayfish_id = NA, week = NA,
  weight = c(3, 7, 11)
)

# Fitted means and standard errors on log-odds scale
fitted <- predict(glmm, newdata, allow.new.levels = TRUE, se.fit = TRUE)
newdata$fit_logit <- as.numeric(fitted$fit)
newdata$se_logit <- as.numeric(fitted$se.fit)

# Transform to probability scale, compute confidence intervals
newdata$fit_prob <- plogis(newdata$fit_logit)
newdata$lower <- plogis(qnorm(0.025, newdata$fit_logit, newdata$se_logit))
newdata$upper <- plogis(qnorm(0.975, newdata$fit_logit, newdata$se_logit))

# Plot
newdata |>
  mutate(weight = factor(weight)) |>
  filter(snail_species == "Bu. truncatus" & infection_status == "uninfected") |>
  ggplot(aes(temp, color = weight)) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0,
                position = position_dodge(width = 0.4)) +
  geom_point(aes(y = fit_prob), position = position_dodge(width = 0.4)) +
  facet_wrap(~time, labeller = label_both) +
  labs(x = "temperature (C)", y = "P(snail consumption)") +
  theme(strip.background = element_blank()) +
  guides(color = guide_legend(reverse = TRUE), fill = guide_legend(reverse = TRUE)) +
  scale_color_manual(values = wght_col) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1), breaks = seq(0, 1, 0.25))
```



Post-hoc tests

Temperature

Here are the estimated marginal means for probability of consumption at each temperature category. The estimated marginal means will differ somewhat from means computed directly from the data, because when averaging temperature effects over all other experimental factors to compute the estimated marginal means, we do so giving each factor equal weight. This produces estimates for the means within each temperature treatment that we would expect to see from a perfectly balanced experimental design:

```
kable(emmeans(glmm, data = exp_data, specs = "temp", type = "response"), digits = 3)
```

temp	prob	SE	df	asymp.LCL	asymp.UCL
15	0.102	0.032	Inf	0.055	0.184
20	0.205	0.054	Inf	0.119	0.330
25	0.267	0.063	Inf	0.162	0.406
30	0.373	0.073	Inf	0.244	0.524
35	0.133	0.039	Inf	0.073	0.230

Computing pairwise comparisons among temperature levels, applying a Tukey correction to the p-values:

```
# Estimated marginal means for each temp term
temp_emmeans <- emmeans(glmm, data = exp_data, specs = "temp")

## Compute pairwise comparisons
temp_emmeans ▷ pairs() ▷ tidy_emmeans()
```

contrast	estimate	SE	df	z.ratio	p.value	
temp15 - temp20	-0.816	0.431	Inf	-1.893	0.321	
temp15 - temp25	-1.160	0.443	Inf	-2.615	0.068	
temp15 - temp30	-1.653	0.442	Inf	-3.738	0.002	*
temp15 - temp35	-0.297	0.447	Inf	-0.663	0.964	
temp20 - temp25	-0.343	0.431	Inf	-0.796	0.932	
temp20 - temp30	-0.837	0.431	Inf	-1.944	0.294	
temp20 - temp35	0.519	0.440	Inf	1.180	0.763	
temp25 - temp30	-0.493	0.422	Inf	-1.169	0.769	
temp25 - temp35	0.863	0.443	Inf	1.947	0.293	
temp30 - temp35	1.356	0.435	Inf	3.118	0.016	*

Obtaining the letters corresponding to a compact letters display based on the Tukey pairwise comparisons:

```
# Letters and estimated marginal means on response scale
temp_cld <- as.data.frame(cld(temp_emmeans, Letters = letters, type = "response"))
```

To visualize the effect of weight on the temperature dependence of predation, we can fit a second model with `gamm`. This model includes temperature and weight only (not their interaction, as this was not significant):

```
## Fit model with only temperature smoother and random effects
temp_gamm <- gamm(
  cbind(consumption_all, snail_density - dead_snails - consumption_all) ~
    s(temp, k = 4) + weight,
  family = binomial(link = "logit"),
  random = list(crayfish_id = ~ 1 | crayfish_id * week),
  data = mutate(exp_data, temp = as.numeric(as.character(temp)))
)
```

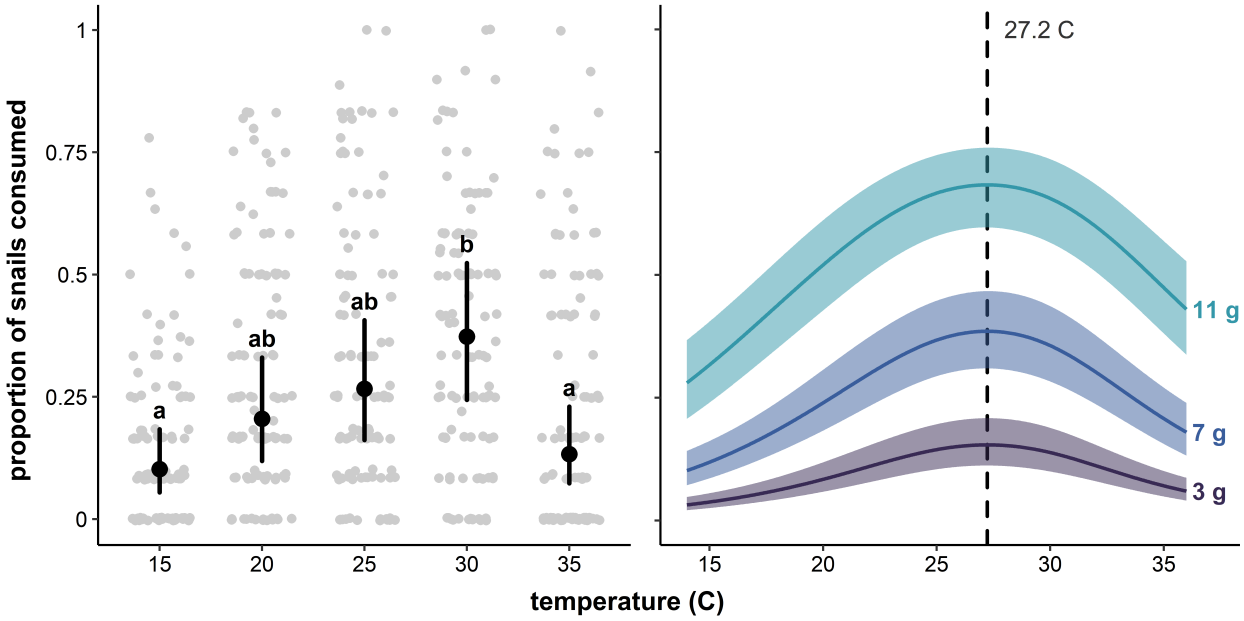
```
##
##  Maximum number of PQL iterations:  20
```

We can obtain predictions across a range of temperatures and at different weights:

```
## New data to predict on
newdata <- tibble(
  temp = rep(seq(14, 36, length.out = 500), 3),
  weight = rep(c(3, 7, 11), each = 500)
)

# Obtain fitted means and confidence intervals
newdata$fit <- predict(temp_gamm$gam, newdata)
newdata$se <- predict(temp_gamm$gam, newdata, se.fit = TRUE)$se.fit
newdata$fit_prob <- plogis(newdata$fit)
newdata$lower <- plogis(qnorm(0.025, newdata$fit, newdata$se))
newdata$upper <- plogis(qnorm(0.975, newdata$fit, newdata$se))
newdata$weight <- factor(newdata$weight)
```

Plotting the continuous curves beside discrete estimates, with point estimates, confidence intervals, and Tukey letters superimposed:



Time

Computing Tukey letters:

```
# Estimated marginal means for each temp term
time_emmeans <- emmeans(glmm, data = exp_data, specs = "time")

# Letters and estimated marginal means on response scale
time_cld <- as.data.frame(cld(time_emmeans, Letters = letters, type = "response"))
```

We can sample from the fitted means and standard errors to display the implied linear effect of time, as a way of essentially back-transforming the linear trend given by `contrast(time_emmeans, method = "poly")` to the scale of the observed data:

```
n <- 10000 # Number of posterior samples

## Sample from estimated means and covariance
time <- as.numeric(as.character(summary(time_emmeans)$time))
time_mu <- summary(time_emmeans)$emmean
time_vcov <- vcov(time_emmeans)
time_samples <- mgcv:::rmvn(n, time_mu, time_vcov)

# Fit regressions, extract coefficients
time_coef_sim <- matrix(NA_real_, nrow = n, ncol = 2)

for (i in 1:n) {
  time_lm <- lm(time_samples[i,] ~ time)
  time_coef_sim[i,] <- coef(time_lm)
}

# Coefficient means and covariance
time_mu_sim <- colMeans(time_coef_sim)
time_vcov_sim <- cov(time_coef_sim)
time_se_sim <- sqrt(diag(time_vcov_sim))
```

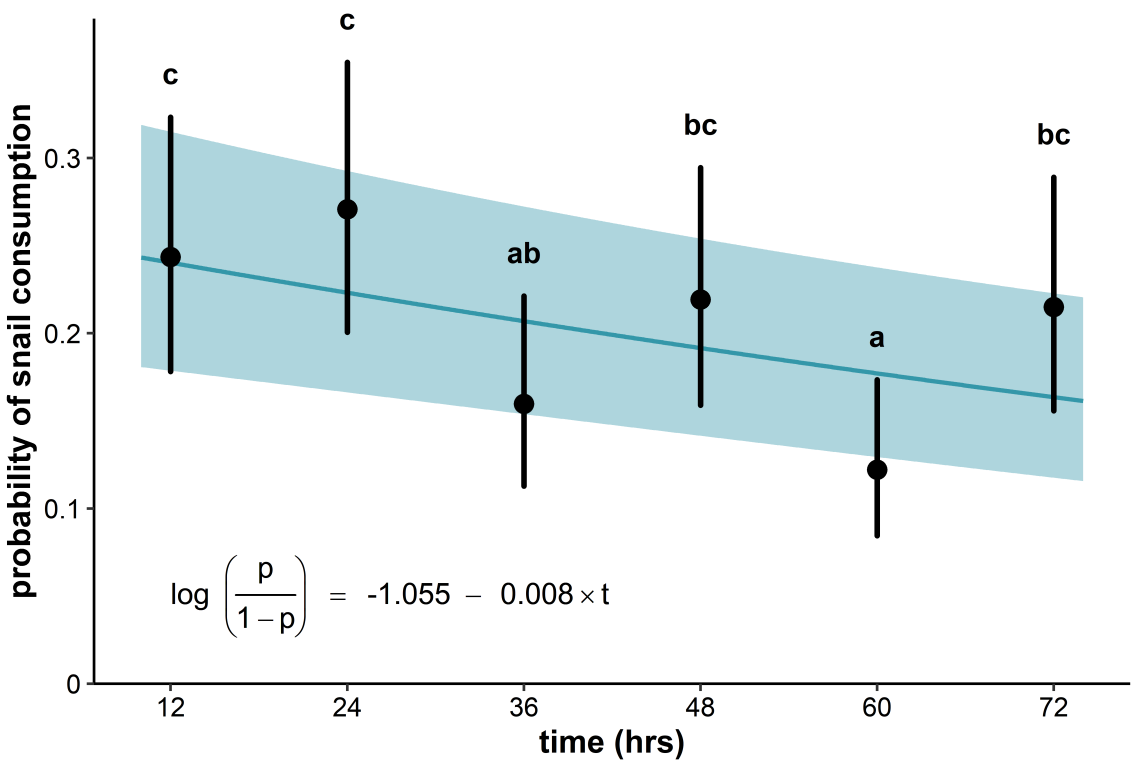
```
## Fitted means and standard errors for range of temperatures
time_seq <- seq(10, 74, by = 0.1)
t_mm <- cbind(rep(1, length(time_seq)), time_seq)
p_hat <- t_mm %*% time_mu_sim
p_se <- sqrt(diag(t_mm %*% time_vcov_sim %*% t(t_mm)))

## Fitted confidence interval
time_fit <- tibble(
  time = time_seq,
  mean = plogis(p_hat),
  lower = plogis(qnorm(0.025, p_hat, p_se)),
  upper = plogis(qnorm(0.975, p_hat, p_se))
)

## Equation for regression fit
eqn <- expr(
  log * ~~ bgroup("(", frac(p, 1 - p), ")" ) * phantom(0) ==
    ~~ !!round(time_mu_sim[1], 3) * ~~-~~ !!round(abs(time_mu_sim[2]), 3) %*% t
)
```

Superimposing the estimated means and confidence intervals on the linear tend component and confidence band:

```
time_cld |>
  mutate(time = as.numeric(as.character(time)), .group = trimws(.group)) |>
  ggplot(aes(time)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), data = time_fit,
              alpha = 0.4, fill = wght_col[3]) +
  geom_line(aes(y = mean), data = time_fit, color = wght_col[3], linewidth = 0.8) +
  geom_errorbar(aes(ymin = asymp.LCL, ymax = asymp.UCL), width = 0,
                linewidth = 1, lineend = "round") +
  geom_point(aes(y = prob), size = 3) +
  geom_text(aes(y = asymp.UCL, label = .group), nudge_y = 0.025, fontface = "bold") +
  annotate("text", label = eqn, x = 12, y = 0.05, hjust = 0) +
  scale_x_continuous(breaks = seq(12, 72, 12)) +
  scale_y_continuous(expand = c(0, 0), breaks = seq(0, 0.3, 0.1),
                     labels = c("0", "0.1", "0.2", "0.3")) +
  labs(x = "time (hrs)",  y = "probability of snail consumption") +
  expand_limits(y = 0) +
  coord_cartesian(clip = "off")
```



We can use these samples to test the linear effect of time:

```
round(c(
  slope = time_mu_sim[2], se = time_se_sim[2],
  quantile(time_coef_sim[,2], c(0.025, 0.975)),
  t = time_mu_sim[2]/time_se_sim[2],
  p.value = pt(time_mu_sim[2]/time_se_sim[2], df = df.residual(glmm)) * 2
), 4)
```

```
##   slope      se    2.5%   97.5%       t p.value
## -0.0080  0.0019 -0.0118 -0.0043 -4.1310  0.0000
```

Unavaiable snails

Here, we'll use the snail behavior data to examine whether the number of snails unavailable to crayfish was related to the treatments, i.e., whether treatments impacted snail avoidant behavior.

First, we create an unavailable snails variable, dropping NA values for and therefore excluding R1 data in which snail behavior was not evaluated:

```
r2_data <- all_data ▷
  drop_na(c(snails_out, snails_under_shelter)) ▷
  mutate(
    unavailable_snails = snails_out + snails_under_shelter,
    p_unavailable = unavailable_snails / (snail_density - dead_snails)
  )
```

The mean proportion of unavailable snails in each trial is 0.115 (SE = 0.007). By treatment group (experimental vs. control) and snail species, we have:

```
r2_data ▷
  group_by(snail_species, condition) ▷
  summarize(
    N = n(),
    `P(unavailable)` = mean(unavailable_snails / snail_density),
    `SE(P(unavailable))` = sd(unavailable_snails / snail_density) / sqrt(n())
  ) ▷
  kable(digits = 3)
```

snail_species	condition	N	P(unavailable)	SE(P(unavailable))
Bi. glabrata	control	118	0.018	0.006
Bi. glabrata	experimental	118	0.174	0.018
Bu. truncatus	control	116	0.106	0.011
Bu. truncatus	experimental	120	0.162	0.015

Model

Fitting another binomial model with the number of unavailable snails as a response, and dropping the crayfish random effect as crayfish are absent from control trials:

```
unavailable_snail_model <- glmmTMB(
  cbind(unavailable_snails, snail_density - dead_snails - unavailable_snails) ~
    snail_species * infection_status * condition + (1 | week),
  family = binomial(link = "logit"),
  data = r2_data
)
```

The null hypothesis tests for the main effects and interactions:

```
tidy_anova(unavailable_snail_model)
```

	Chisq	Df	Pr(>Chisq)	
snail_species	9.103	1	0.003	*
infection_status	3.526	1	0.06	
condition	90.575	1	<0.001	*
snail_species:infection_status	2.318	1	0.128	
snail_species:condition	66.375	1	<0.001	*
infection_status:condition	14.679	1	<0.001	*
snail_species:infection_status:condition	0.280	1	0.597	

Pairwise comparisons

Main effects

Examining estimated means across the main effects of condition:

```
kable(emmeans(unavailable_snail_model, specs = "condition", type = "response"), digits = 3)
```

condition	prob	SE	df	asymp.LCL	asymp.UCL
control	0.041	0.006	Inf	0.031	0.054
experimental	0.165	0.015	Inf	0.138	0.195

and snail species:

```
kable(emmeans(unavailable_snail_model, specs = "snail_species", type = "response"), digits = 3)
```

snail_species	prob	SE	df	asymp.LCL	asymp.UCL
Bi. glabrata	0.055	0.007	Inf	0.042	0.072
Bu. truncatus	0.126	0.012	Inf	0.104	0.152

Interaction

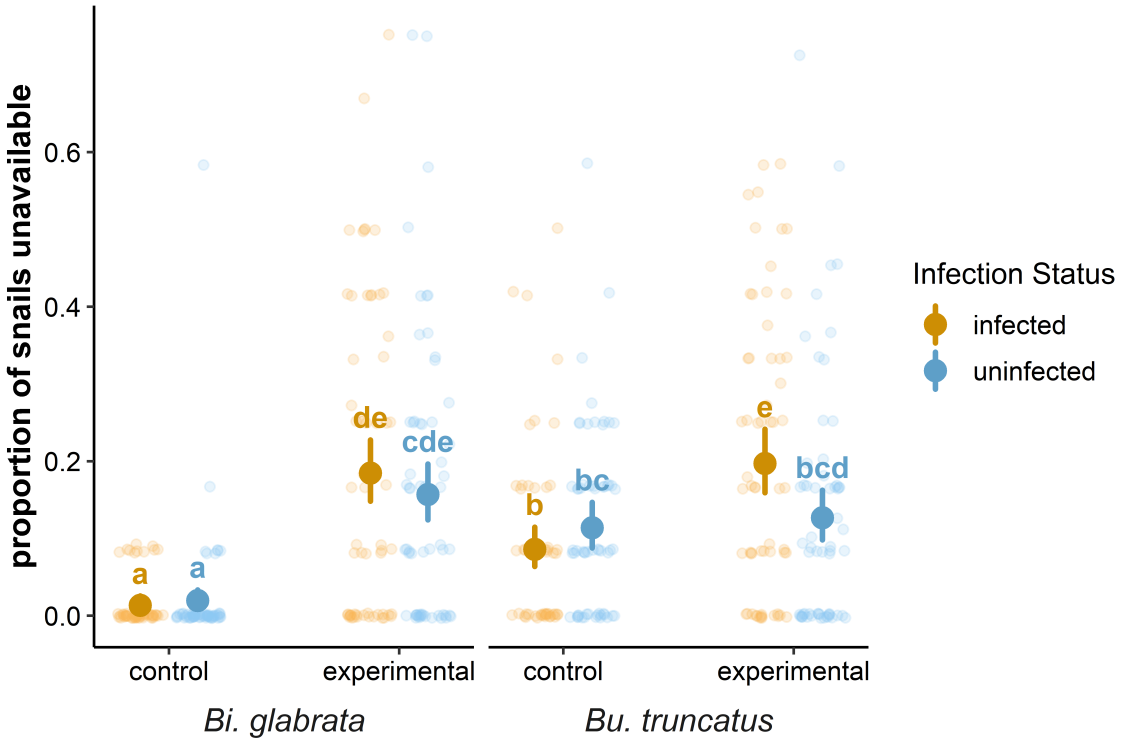
While the third order interaction between snail species, infection status, and condition is not significant, two of the lower order interactions are, so I'll compute pairwise interactions for these treatments together:

```
unsnail_emmeans <- emmeans(
  unavailable_snail_model,
  specs = c("snail_species", "infection_status", "condition")
)

unsnail_cld <- as.data.frame(cld(unsnail_emmeans, Letters = letters, type = "response"))
unsnail_cld$.group <- trimws(unsnail_cld$.group)
```

Plotting fitted means and letters:

```
unsnail_cld ▷
  mutate(condition = as.numeric(as.factor(condition))) ▷
  ggplot(aes(condition)) +
  geom_jitter(aes(x = as.numeric(as.factor(condition)) - 0.125, y = p_unavailable),
              data = r2_data ▷ filter(infection_status == "infected"),
              alpha = 0.2, width = 0.1, color = colorspace::lighten("#CD8E04", 0.4)) +
  geom_jitter(aes(x = as.numeric(as.factor(condition)) + 0.125, y = p_unavailable),
              data = r2_data ▷ filter(infection_status == "uninfected"),
              alpha = 0.2, width = 0.1, color = colorspace::lighten("#5E9FC8", 0.4)) +
  geom_pointrange(aes(y = prob, ymin = asymp.LCL, ymax = asymp.UCL, color = infection_status),
                  lineend = "round", linewidth = 1, position = position_dodge(0.5), size = 0.7) +
  geom_text(aes(y = asymp.UCL + 0.03, label = .group, color = infection_status), size = 4,
            position = position_dodge(0.5), fontface = "bold", show.legend = FALSE) +
  facet_wrap(~snail_species, nrow = 1, strip.position = "bottom") +
  scale_color_manual(name= "Infection Status", labels = c("infected", "uninfected"),
                     values = c("#CD8E04", "#5E9FC8")) +
  ylab("proportion of snails unavailable") +
  scale_x_continuous(breaks = 1:2, labels = c("control", "experimental")) +
  theme(
    axis.title.x = element_blank(),
    strip.text = element_text(face = "italic")
  )
```



Snail mortality

Summarizing the number of trials with snail mortality and the snail mortality rate (proportion of dead snails) in control and experimental tanks by snail species:

```
all_data ▷
  group_by(snail_species, condition) ▷
  summarize(
    N = n(),
    `trials with mortality` = sum(dead_snails > 0),
    `mortality rate` = mean(dead_snails / snail_density),
    `SE(mortality rate)` = sd(dead_snails / snail_density) / sqrt(n())
  ) ▷
  kable(digits = 3)
```

snail_species	condition	N	trials with mortality	mortality rate	SE(mortality rate)
Bi. glabrata	control	181	11	0.005	0.001
Bi. glabrata	experimental	268	42	0.017	0.003
Bu. truncatus	control	169	4	0.002	0.001
Bu. truncatus	experimental	272	83	0.046	0.005

Model

Fitting a similar model to the above, with the number of dead snails as the response and using all 890 observations (350 control and 540 experimental):

```
dead_snail_model <- glmmTMB(
  cbind(dead_snails, snail_density - dead_snails) ~
```

```
    snail_species * infection_status * condition + (1 | week),
  family = binomial(link = "logit"),
  data = all_data
)
```

Null hypothesis tests of the main effects and interactions, suggesting that mortality differed between species, between infected and uninfected snails, between experimental and control tanks, and that the effect of the treatment on snail mortality differed among species:

```
tidy_anova(dead_snail_model)
```

	Chisq	Df	Pr(>Chisq)	
snail_species	34.184	1	<0.001	*
infection_status	8.511	1	0.004	*
condition	41.179	1	<0.001	*
snail_species:infection_status	0.739	1	0.39	
snail_species:condition	9.299	1	0.002	*
infection_status:condition	0.735	1	0.391	
snail_species:infection_status:condition	0.056	1	0.814	

Pairwise comparisons

Infection status

There's only two categories here, so the p-value is the p-value of the main effect for infection status reported above, but here are the estimated mortality rates for infected and uninfected snails, averaging over species and condition due to the lack of interactions between infection status and these covariates. While the effect is significant, it's *very* small - a difference of 0.3%:

```
kable(emmeans(dead_snail_model, "infection_status", type = "response"), digits = 3)
```

infection_status	prob	SE	df	asymp.LCL	asymp.UCL
infected	0.006	0.002	Inf	0.003	0.012
uninfected	0.009	0.003	Inf	0.005	0.016

Condition by species

Examining the snail species by condition interaction, we see that mortality was significantly higher in treatment tanks than in control tanks for both species, but still fairly low overall, with *Bu. truncatus* and *Bi. glabrata* in the experimental having a mortality rate of about 3% and 1%, respectively.

```
dsnail_emmeans <- emmeans(dead_snail_model, specs = c("snail_species", "condition"))

dsnail_cld <- as.data.frame(cld(dsnail_emmeans, Letters = letters, type = "response"))
dsnail_cld$.group <- trimws(dsnail_cld$.group)

dsnail_cld |>
  ggplot(aes(condition)) +
  geom_pointrange(aes(y = prob, ymin = asymp.LCL, ymax = asymp.UCL), lineend = "round",
                  linewidth = 1, position = position_dodge(0.5), size = 0.7) +
  geom_text(aes(y = asymp.UCL + 0.005, label = .group), position = position_dodge(0.5),
            fontface = "bold", show.legend = FALSE) +
  facet_wrap(~snail_species, nrow = 1, strip.position = "bottom") +
  ylab("snail mortality") +
```

```
theme(
  axis.title.x = element_blank(),
  strip.text = element_text(face = "italic")
)
```