

Autocorrelation_stats

Debora

2025-06-16

Load and Preprocess Datasets

We load two datasets: one with a preparation technique (repeated frond selection) and one without.

```
# This dataset contains replicates for which a preparation technique was performed (repeated first born
original_dataset_2 <- read.csv("https://raw.githubusercontent.com/Cuddington-Lab/thermal-experiments/main/original_dataset_2.csv")
original_dataset_2$prep <- rep("yes",times=length(original_dataset_2$Experiment_Number))

# This dataset contains replicates of experiments performed without a preparation technique
original_dataset_1 <- read.csv("https://raw.githubusercontent.com/Cuddington-Lab/thermal-experiments/main/original_dataset_1.csv")
original_dataset_1$prep <- rep("no",times=length(original_dataset_1$Experiment_Number))
```

Combine Datasets

We combine both datasets and filter out rows based on specific conditions for standard deviation (Obs_sd) and autocorrelation (Obs_ac) to clean the data.

```
# Blending both datasets, as there are no significant differences between preparation methods (probably
datin <- rbind(original_dataset_1,original_dataset_2)

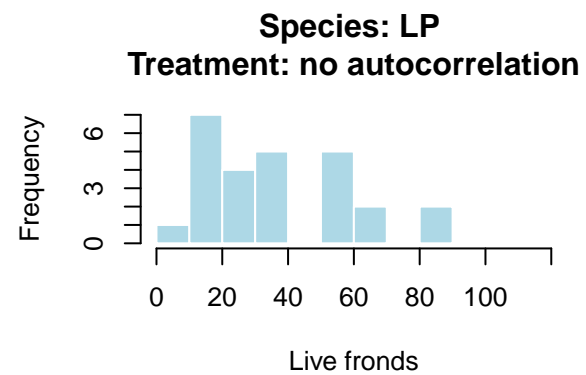
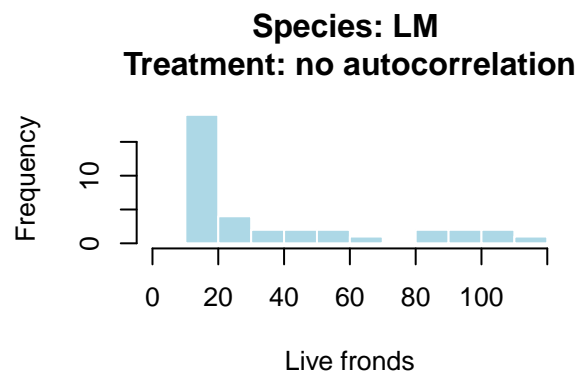
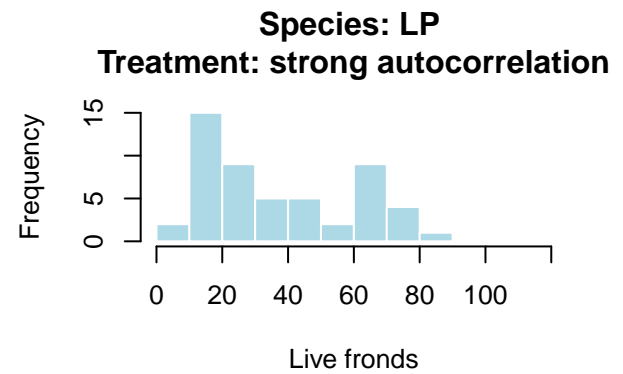
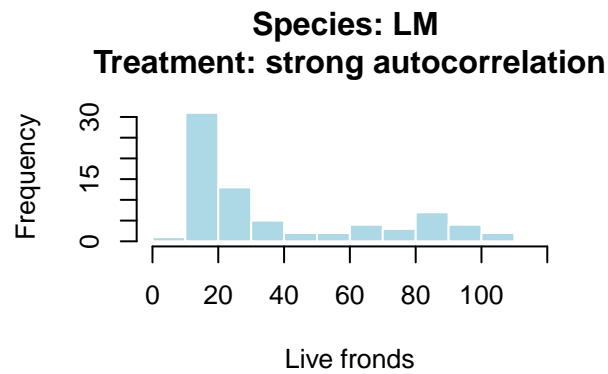
datin <- datin[!(datin$Treatment == 0 & (datin$Obs_sd <= 2.1 | datin$Obs_sd >= 2.9))
              &!(datin$Treatment == 0.95 & (datin$Obs_sd <= 2.1 | datin$Obs_sd >= 2.9)),]

datin <- datin[!(datin$Treatment == 0 & (datin$Obs_ac <= -0.2 | datin$Obs_ac >= 0.2))
              &!(datin$Treatment == 0.95 & (datin$Obs_ac <= 0.92 | datin$Obs_ac >= 0.98)),]

datin <- subset(datin, !Errors == "y"|is.na(Errors))
datin <- subset(datin, !Treatment == "constant")
```

View dataset and response variable

	Species	Exp_run	Treatment	total_living_fronds
395	LM	2733	cold-hot	90
396	LP	2733	cold-hot	74
397	LM	2733	hot-cold	91
398	LP	2733	hot-cold	63
399	LM	2733	no autocorrelation	101
400	LP	2733	no autocorrelation	88



View number of replicates

LM

```
##
##           no autocorrelation cold-hot hot-cold
## 10-15           16           16           16
## 27              12           12           12
## 37              9            9            9
```

LP

```
##
##           no autocorrelation cold-hot hot-cold
## 10-15           5            5            5
## 27              12           12           12
## 37              9            9            9
```

Model fitting and validation

```
model_results <- list()
compare_results <- list()

species_list <- list("LM", "LP")

for (sp in species_list) {
  species_data <- subset(dataset, Species == sp)
```

```

# Assigning new IDs to experiment runs to ensure IDs are unique (as different
# average temperatures had same IDs)
species_data$Exp_run <- rep(seq(1, length(species_data$Exp_run) / 3), each = 3)

library(lme4)

# Fit the models
simple <- glm(total_living_fronds ~ Label*Mean_temperature, data=species_data, family=poisson)
exp_number <- glmer(total_living_fronds ~ Label*Mean_temperature + (1|Exp_run), data=species_data, family=poisson)

# Compare simple and mixed model
anova(exp_number, simple, test = "Chisq")
L0 <- logLik(simple)
L1 <- logLik(exp_number)
L.ratio <- as.vector(- 2 * (L0 - L1))
p_value <- 0.5 * (1 - pchisq(L.ratio, 1))
options(scipen = 999)
print(paste("Comparing simple x mixed model:"))
cat("p_value:", format(p_value, digits = 10), "\n")

# Store models
model_results[[sp]] <- list(simple = simple, exp_number = exp_number)

# Compare models by AIC
Cand.modsF <- list("no random effects" = model_results[[sp]]$simple,
                  "experiment number" = model_results[[sp]]$exp_number)

# Get AIC for each model and store it
AIC_values <- sapply(Cand.modsF, function(model) AIC(model))

# Create a summary table
compare_results[[sp]] <- knitr::kable(data.frame(Model = names(AIC_values), AIC = AIC_values), "simple")

cat("Model assessment:", sp, "\n")
print(compare_results[[sp]])

# Find the best model based on AIC
best_model_name <- names(AIC_values)[which.min(AIC_values)]
model_results[[sp]] <- Cand.modsF[[best_model_name]]

best_model <- model_results[[sp]]

model_dispersion <- sum(residuals(best_model, type = "pearson")^2) / df.residual(best_model)
print(paste("Model dispersion ratio for species", sp, ":", model_dispersion))

cat("Wald test type 3 for significance of predictor:", sp, "\n")
library(car)
phi <- sum(residuals(best_model, type="pearson")^2) / df.residual(best_model)
anova_table <- Anova(best_model, type = "III")

# Adjusting chi-square values (quasipoisson)
anova_table$`Chisq` <- anova_table$`Chisq` / phi

```

```
# Recalculating p-values using chi-square distribution
anova_table$`Pr(>Chisq)` <- pchisq(anova_table$`Chisq`, anova_table$Df, lower.tail = FALSE)

print(anova_table)
}
```

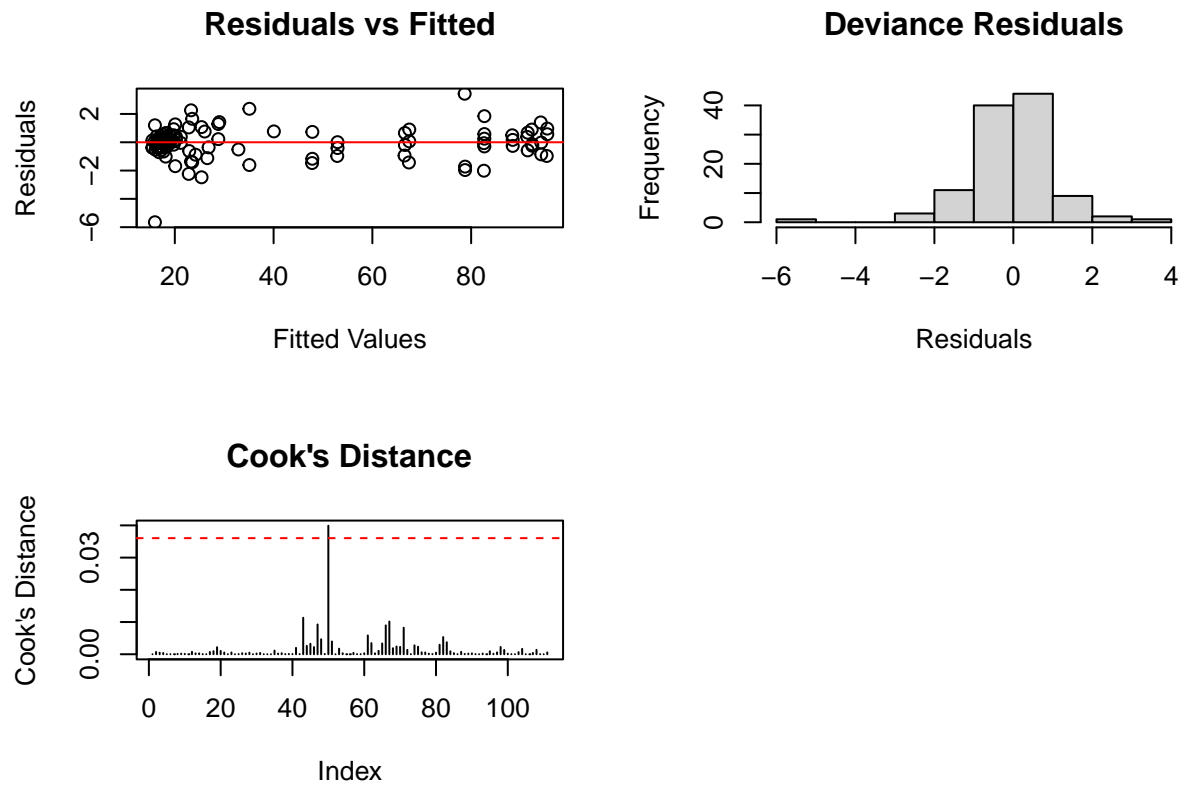
```
## [1] "Comparing simple x mixed model:"
## p_value: 0
## Model assessment: LM
##
##
##           Model                AIC
## -----
## no random effects  no random effects  927.19
## experiment number  experiment number  806.41
## [1] "Model dispersion ratio for species LM : 1.08782676497932"
## Wald test type 3 for significance of predictor: LM
## Analysis of Deviance Table (Type III Wald chisquare tests)
##
## Response: total_living_fronds
##           Chisq Df      Pr(>Chisq)
## (Intercept)    1213.0290  1 <0.0000000000000002 ***
## Label           0.5370  1      0.4637
## Mean_temperature    221.4834  2 <0.0000000000000002 ***
## Label:Mean_temperature    2.9828  2      0.2251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Comparing simple x mixed model:"
## p_value: 0
## Model assessment: LP
##
##
##           Model                AIC
## -----
## no random effects  no random effects  803.69
## experiment number  experiment number  690.76
## [1] "Model dispersion ratio for species LP : 2.39287799238354"
## Wald test type 3 for significance of predictor: LP
## Analysis of Deviance Table (Type III Wald chisquare tests)
##
## Response: total_living_fronds
##           Chisq Df      Pr(>Chisq)
## (Intercept)    133.4712  1 < 0.00000000000000022 ***
## Label           0.2989  1      0.5846
## Mean_temperature    35.9570  2      0.00000001556 ***
## Label:Mean_temperature    0.4591  2      0.7949
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model validation

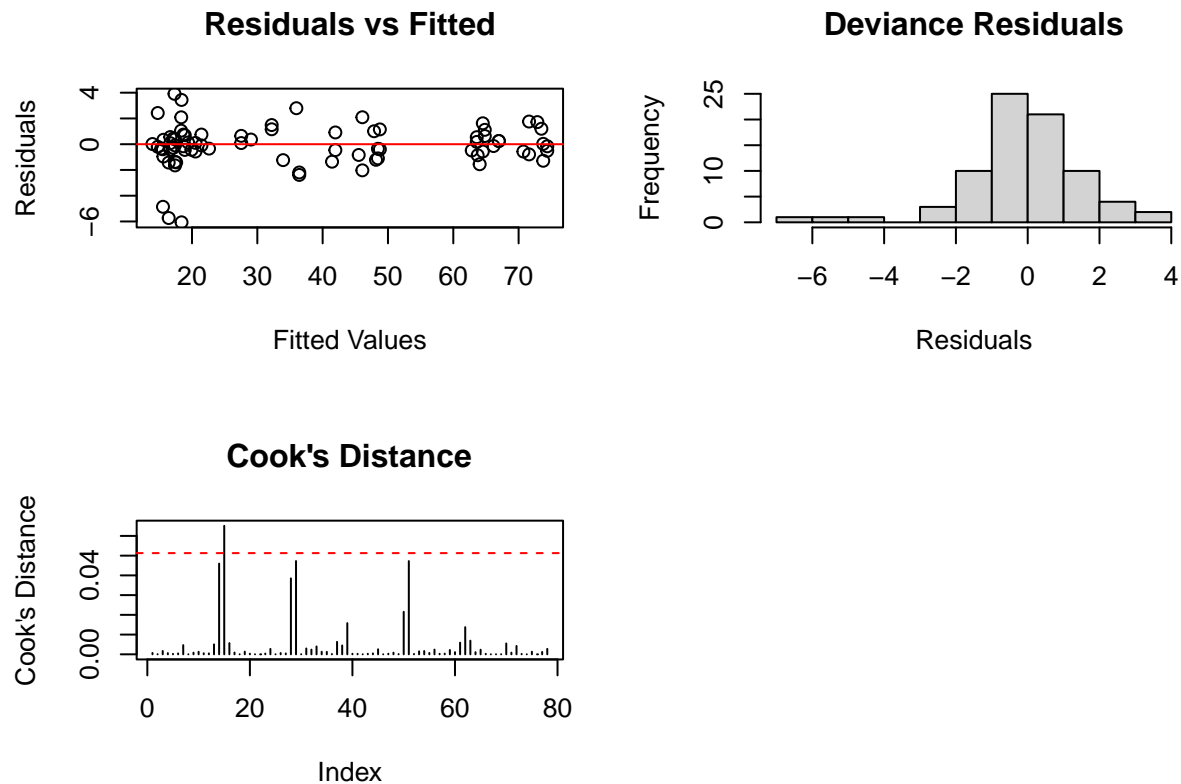
All assumptions were met.

```
## [1] "Model dispersion ratio for species LM : 1.08782676497932"
## LM
```

```
## Deviance of the best model: 128.6759
## Deviance of the null model: 2304.494
## Pseudo R^2 based on deviance: 0.9441631
```



```
## [1] "Model dispersion ratio for species LP : 2.39287799238354"
## LP
## Deviance of the best model: 205.3573
## Deviance of the null model: 1188.315
## Pseudo R^2 based on deviance: 0.8271862
```



Adjusting model results to account for overdispersion

```
for (sp in species_list) {
  full_mod1 <- model_results[[sp]]
  quasi_table <- function(model,ctab=coef(summary(model))) {
    phi <- sum(residuals(model, type="pearson")^2)/df.residual(model)
    qctab <- within(as.data.frame(ctab),
      { `Std. Error` <- `Std. Error`*sqrt(phi)
        `z value` <- Estimate/`Std. Error`
        `Pr(>|z|)` <- 2*pnorm(abs(`z value`), lower.tail=FALSE)
      })
    return(qctab)
  }

  print(paste("Adjusted model summary - species :", sp, ":"))
  printCoefmat(quasi_table(full_mod1),digits=2)
  #As specified, the estimates are identical;
  #the standard errors and p-values have been appropriately
  #inflated, the z-values have been appropriately deflated.
  #https://stackoverflow.com/questions/68915173/how-do-i-fit-a-quasi-poisson-model-with-lme4-or-glmmTMB
  # Checks for Overdispersion: It calculates a number (phi) that tells you how much the data spreads out
  # Fixes Standard Errors: It takes the standard errors from the model and makes them bigger based on phi
  # Recalculates Other Numbers: Since standard errors changed, it also updates the z-values and p-values.
  # Returns the New Table: It gives you a corrected version of the table with the adjusted values.
}
```

```
# Basically, your Poisson model didn't fit well because the data was too spread out, so this function h
}
```

```
## [1] "Adjusted model summary - species : LM :"
```

	Estimate	Std. Error	z	value
## (Intercept)	2.822	0.081	34.8	
## Labelstrong autocorrelation	0.056	0.077	0.7	
## Mean_temperature27	1.515	0.106	14.3	
## Mean_temperature37	0.464	0.125	3.7	
## Labelstrong autocorrelation:Mean_temperature27	-0.055	0.087	-0.6	
## Labelstrong autocorrelation:Mean_temperature37	-0.188	0.113	-1.7	

```
##
## Pr(>|z|)
## (Intercept) <0.0000000000000002 ***
## Labelstrong autocorrelation 0.5
## Mean_temperature27 <0.0000000000000002 ***
## Mean_temperature37 0.0002 ***
## Labelstrong autocorrelation:Mean_temperature27 0.5
## Labelstrong autocorrelation:Mean_temperature37 0.1 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Adjusted model summary - species : LP :"
```

	Estimate	Std. Error	z	value
## (Intercept)	2.76	0.24	11.6	
## Labelstrong autocorrelation	0.11	0.21	0.5	
## Mean_temperature27	1.27	0.27	4.7	
## Mean_temperature37	0.27	0.29	0.9	
## Labelstrong autocorrelation:Mean_temperature27	-0.10	0.22	-0.5	
## Labelstrong autocorrelation:Mean_temperature37	-0.17	0.25	-0.7	

```
##
## Pr(>|z|)
## (Intercept) < 0.0000000000000002 ***
## Labelstrong autocorrelation 0.6
## Mean_temperature27 0.000002 ***
## Mean_temperature37 0.3
## Labelstrong autocorrelation:Mean_temperature27 0.6
## Labelstrong autocorrelation:Mean_temperature37 0.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```