AC_Va_h2_R_2023

Helen Payne

2024-06-18

```r
#load packages
library(lme4)
```

```
## Loading required package: Matrix
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
## The following object is masked from 'package:stats':
##
##     step
```

```r
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

##########**2023**##########

Read in the data:

```
AC_23 <- read_csv(here::here("data_sheets", "compiled_sheets", "AC_mastersheet_Fitness-mains_2023.csv"))
```

```
## Rows: 545 Columns: 50
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (5): Gen, Donor, Recipient, sample_ID_SEG, SegPos
## dbl  (35): Year, Transect, Sequence, Plant_ID, days_sow2flower, days_plant2f...
## lgl   (2): F_plant, F_Num_03
## date  (8): Sow_Date, Plant_Date, FFD, LFD, F_Num_01, F_Num_02, photo_date, p...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
AC_23_fit <- AC_23 %>%
  filter(Gen == "G1")

AC_22_23_full <- read_csv(here::here("data_sheets", "compiled_sheets", "AC_22_23_full.csv"))
```

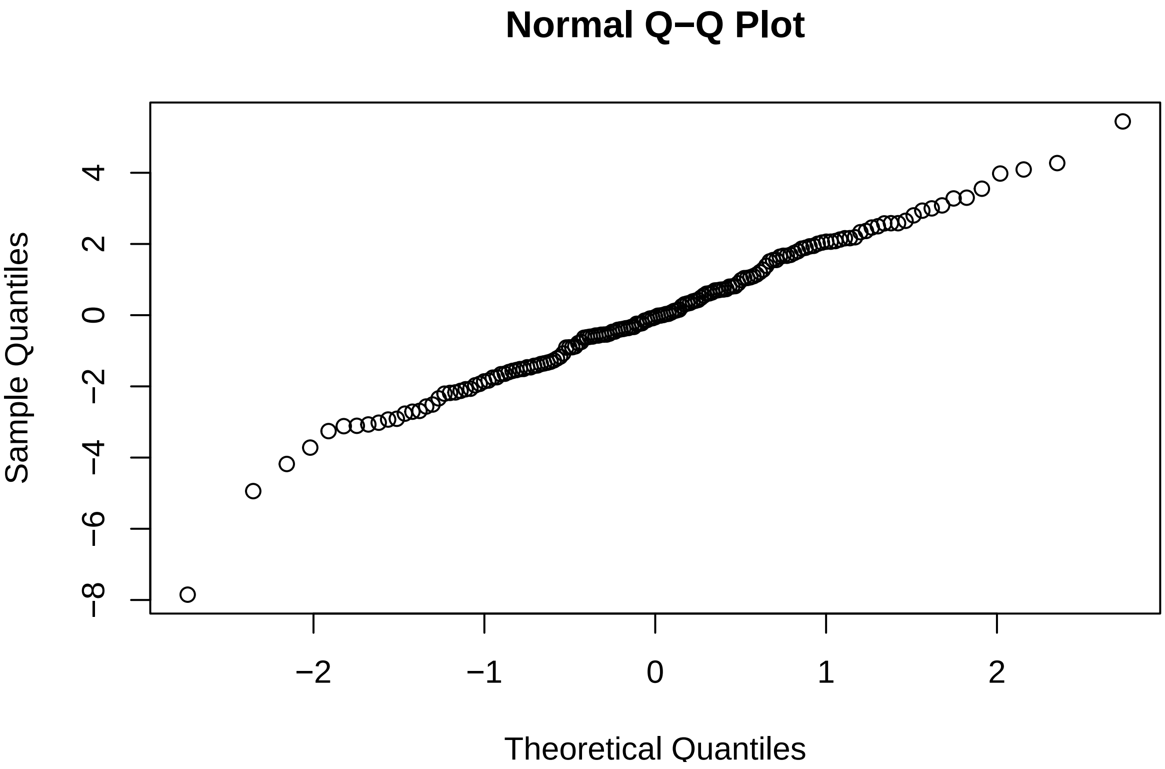
```
## Rows: 6733 Columns: 65
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (7): Gen, Donor, Recipient, sample_ID_SEG, SegPos, Block, Replicated
## dbl  (45): Year, Transect, Sequence, Plant_ID, days_sow2flower, days_plant2f...
## lgl   (4): F_plant, F_Num_03, Rep_FitP, any_FitP
## date  (9): Sow_Date, Plant_Date, FFD, LFD, F_Num_01, F_Num_02, photo_date, p...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Calculate the mean surv_to_flower for each group in AC_22_23_full and add it as a new column
AC_22_23_full <- AC_22_23_full %>%
  group_by(Year, Gen, Transect, Sequence, Donor, Recipient, SegPos) %>%
  mutate(prop_surv_to_flower = mean(surv_to_flower, na.rm = TRUE)) %>%
  ungroup()  # Remove grouping

AC_prop_sample <- AC_22_23_full %>%
  select(c(Year, Gen, Transect, Sequence, Donor, Recipient, SegPos, prop_surv_to_flower))  %>%
    distinct()

AC_23_fit <- AC_23_fit %>%
  left_join(AC_prop_sample %>% select(Year, Gen, Transect, Sequence, Donor, Recipient, SegPos, prop_surv
            by = c("Year", "Gen", "Transect", "Sequence", "Donor", "Recipient"))

# Create the new column
AC_23_fit <- AC_23_fit %>%
  mutate(est_fitness = prop_surv_to_flower * est_fecundity)
```

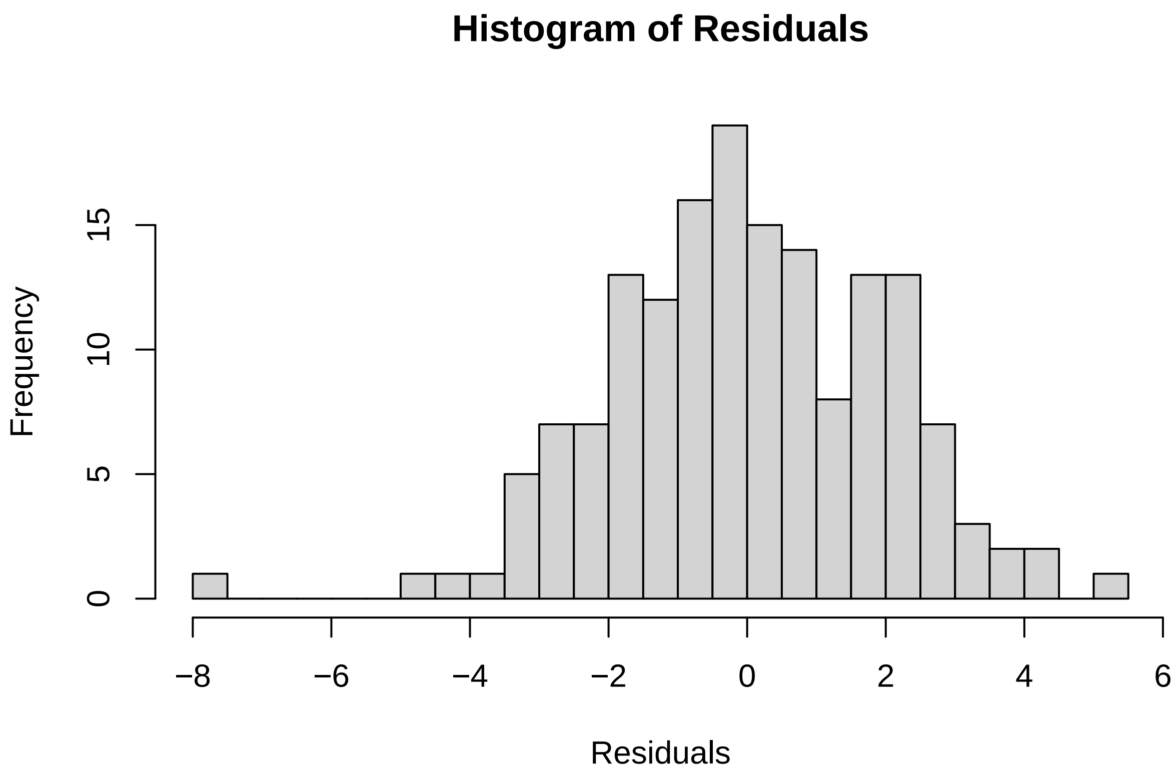
```
#log transform traits that need it
AC_23_fit$skel_dryweight_mg_SEG <- log(AC_23_fit$skel_dryweight_mg_SEG)
```

```r
AC_23_fit$msm_all <- log(AC_23_fit$msm_all)
AC_23_fit$SLA_SEG <- log(AC_23_fit$SLA_SEG)
AC_23_fit$est_fitness <- sqrt(AC_23_fit$est_fitness)


#mean center the traits of interest

traits <- c("corolla_diam_mm_SEG", "skel_dryweight_mg_SEG", "fl_duration", "est_fecundity", "msm_all",

# Mean center each trait
for (trait in traits) {
  trait_mean <- mean(AC_23_fit[[trait]], na.rm = TRUE)
  AC_23_fit[[paste0(trait, "_centered")]] <- AC_23_fit[[trait]] - trait_mean
}
```

```r
# Create the mixed model for corolla area

corolla_model <- lmer(corolla_diam_mm_SEG_centered ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit)

# Extract residuals from the model
residuals <- resid(corolla_model)

# Q-Q plot for normality
qqnorm(residuals) #looks good
```



```r
# Histogram for normality
hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal
```



```
# Create the mixed model for skeleton weight, with skeleton weight log transformed
#corolla_model_2 <- lmer(log(corolla_diam_mm_SEG_centered) ~ (1 |Transect)  + (1 |Recipient) + (1 | Don

# Extract residuals from the model
#residuals <- resid(corolla_model_2)

# Q-Q plot for normality
#qqnorm(residuals) #looks good


# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal


# Compare models using AIC, BIC, and log-likelihood
#corolla_model_comparison <- data.frame(
  #Model = c("corolla_model", "corolla_model_2"),
  #AIC = c(AIC(corolla_model), AIC(corolla_model_2)),
  #BIC = c(BIC(corolla_model), BIC(corolla_model_2)),
  #LogLikelihood = c(logLik(corolla_model), logLik(corolla_model_2))
#)

# Print model comparison
#print(corolla_model_comparison)

# Test the significance of the random effects
rand(corolla_model)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
```

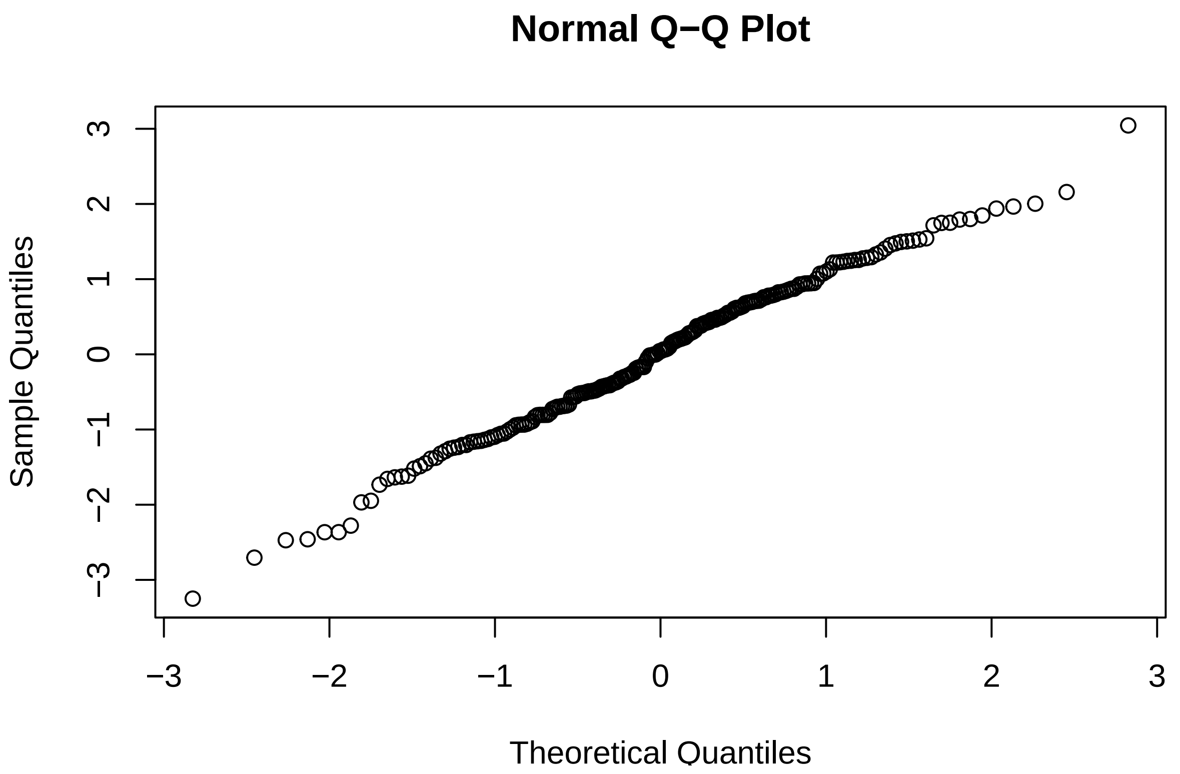
```
## Model:
## corolla_diam_mm_SEG_centered ~ (1 | Recipient) + (1 | Donor)
##                 npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>             4 -385.64 779.28
## (1 | Recipient)    3 -388.35 782.69 5.4124  1    0.01999 *
## (1 | Donor)        3 -385.76 777.51 0.2324  1    0.62978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#model 2 is the best
```

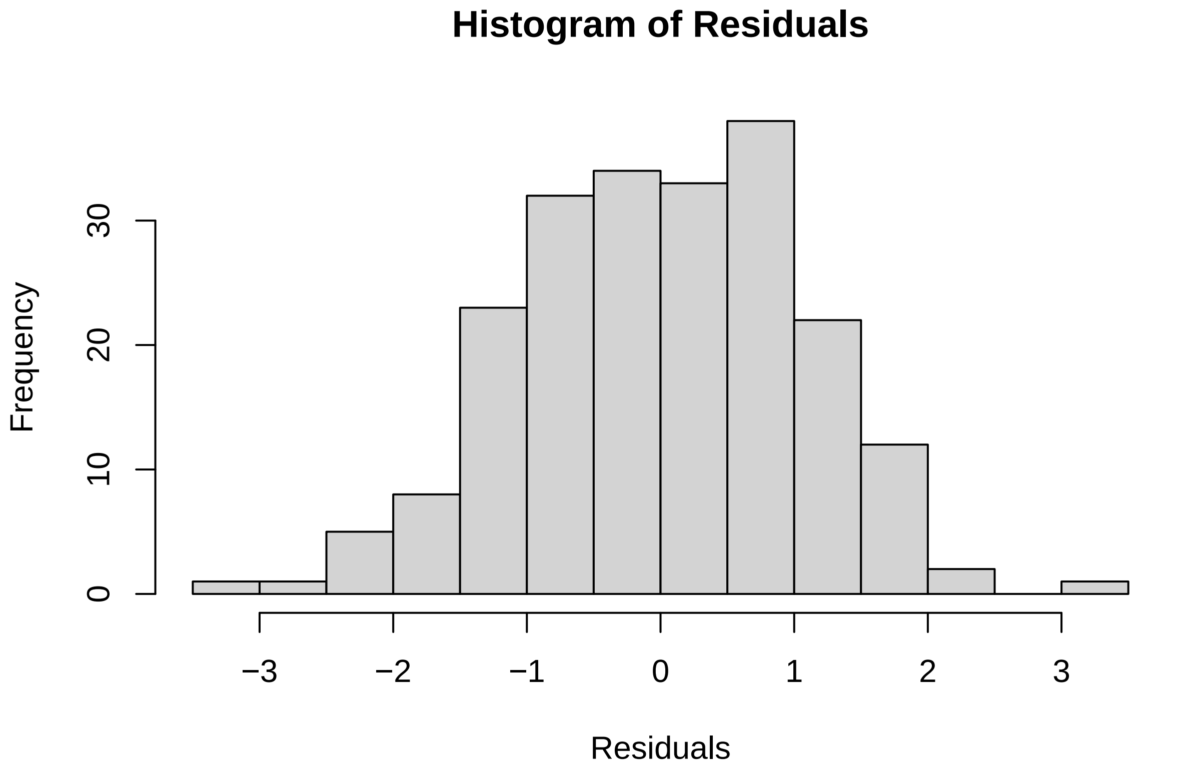
```
# Create the mixed model for skeleton weight, with skeleton weight log transformed
#skel_model <- lmer((skel_dryweight_mg_SEG_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit)

# Extract residuals from the model
#residuals <- resid(skel_model)

# Q-Q plot for normality
#qqnorm(residuals) #looks good



# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal

# Test the significance of the random effects
#rand(corolla_model)

# Create the mixed model for skeleton weight, with skeleton weight log transformed
skel_model_2 <- lmer((skel_dryweight_mg_SEG_centered) ~ (1 | Donor), data = AC_23_fit)

# Test the significance of the random effects
rand(skel_model_2)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## (skel_dryweight_mg_SEG_centered) ~ (1 | Donor)
##             npar  logLik    AIC        LRT Df Pr(>Chisq)
## <none>         3 -316.02 638.05
## (1 | Donor)    2 -316.02 636.05 0.00042227  1     0.9836
```

```
# Extract residuals from the model
residuals <- resid(skel_model_2)

# Q-Q plot for normality
qqnorm(residuals) #looks good
```



```
# Histogram for normality
hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal
```



```
# Compare models using AIC, BIC, and log-likelihood
#skel_model_comparison <- data.frame(
  #Model = c("skel_model", "skel_model_2"),
  #AIC = c(AIC(skel_model), AIC(skel_model_2)),
  #BIC = c(BIC(skel_model), BIC(skel_model_2)),
  #LogLikelihood = c(logLik(skel_model), logLik(skel_model_2))
#)

# Print model comparison
#print(skel_model_comparison)

#model 1 is the best
```

```
# Create the mixed model for flowering duration
#fl_duration_model <- lmer(fl_duration_centered ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit) #Sin

# Check if the model is singular
#fl_duration_model <- lmer(fl_duration_centered ~ (1 | Donor), data = AC_23_fit) #Singular
```

```
# Create the mixed model for estimated fecundity, sqrt transforming estimated fecundity
#est_fecundity_model <- lmer((est_fecundity_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit


# Extract residuals from the model
#residuals <- resid(est_fecundity_model)

# Q-Q plot for normality
#qqnorm(residuals) #good enough


# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish


# Create the mixed model for estimated fecundity, sqrt transforming estimated fecundity
#est_fecundity_model_2 <- lmer((est_fecundity_centered) ~ (1 | Donor), data = AC_23_fit) #still singula

# Test the significance of the random effects
#rand(est_fecundity_model)

# Extract residuals from the model
#residuals <- resid(est_fecundity_model_2)

# Q-Q plot for normality
#qqnorm(residuals) #good enough


# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish




# Compare models using AIC, BIC, and log-likelihood
#est_fecundity_model_comparison <- data.frame(
  #Model = c("est_fecundity_model", "est_fecundity_model_2"),
  #AIC = c(AIC(est_fecundity_model), AIC(est_fecundity_model_2)),
  #BIC = c(BIC(est_fecundity_model), BIC(est_fecundity_model_2)),
  #LogLikelihood = c(logLik(est_fecundity_model), logLik(est_fecundity_model_2))
#)

# Print model comparison
#print(est_fecundity_model_comparison)

#model 2 is the best
```

```
# Create the mixed model for mean seed mass, log transformed mean seed mass
#msm_model <- lmer((msm_all_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit) #Singular

# Extract residuals from the model
#residuals <- resid(msm_model)

# Q-Q plot for normality
#qqnorm(residuals) #good enough


# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish

# Create the mixed model for mean seed mass, log transformed mean seed mass
#msm_model_2 <- lmer((msm_all_centered) ~ (1 | Donor), data = AC_23_fit)

# Extract residuals from the model
#residuals <- resid(msm_model_2)

# Q-Q plot for normality
#qqnorm(residuals) #good enough

# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish


# Compare models using AIC, BIC, and log-likelihood
#msm_model_comparison <- data.frame(
  #Model = c("msm_model", "msm_model_2"),
  #AIC = c(AIC(msm_model), AIC(msm_model_2)),
  #BIC = c(BIC(msm_model), BIC(msm_model_2)),
  #LogLikelihood = c(logLik(msm_model), logLik(msm_model_2))
#)

# Print model comparison
#print(msm_model_comparison)

#model 1 is the best

# Test the significance of the random effects
#rand(msm_model_2)
```

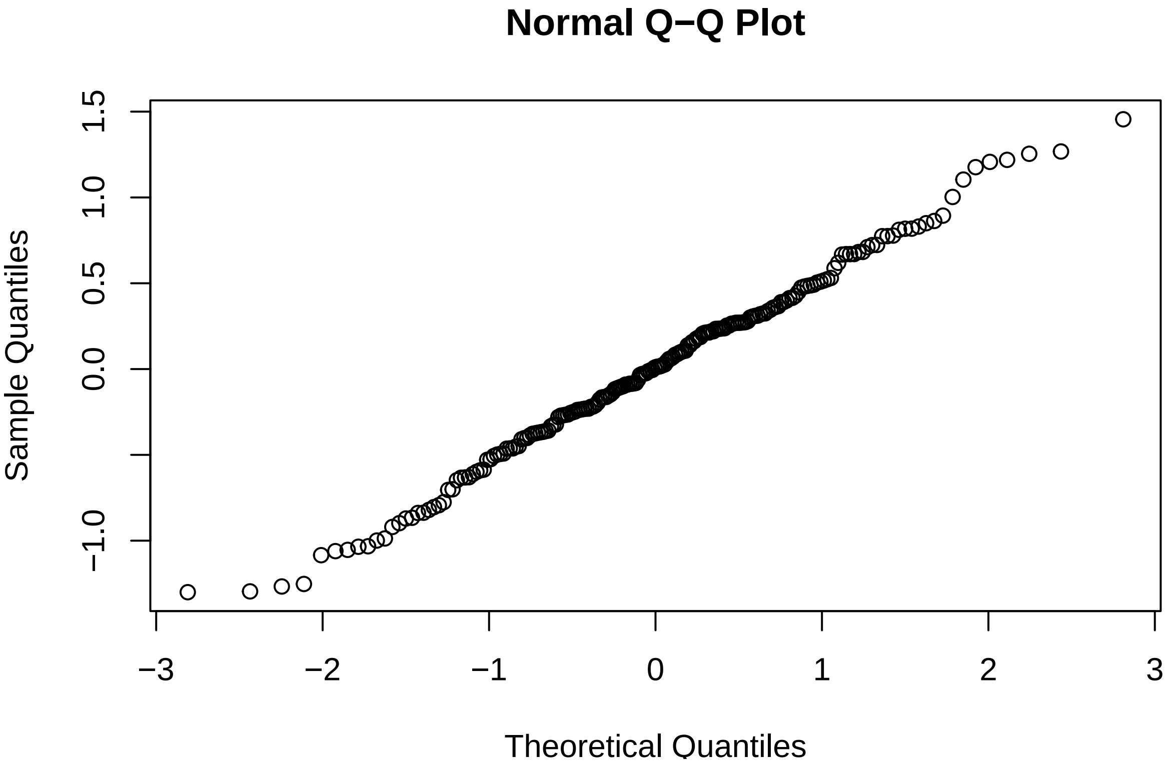
```
# Create the mixed model for SLA
#SLA_model <- lmer((SLA_SEG_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit) #singular

# Extract residuals from the model
```

```
#residuals <- resid(SLA_model)

# Q-Q plot for normality
#qqnorm(residuals) #okay..


# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #definitely a few ou


##adding in transect##
# Create the mixed model for SLA
#SLA_model_2 <- lmer((SLA_SEG_centered) ~ (1 | Donor), data = AC_23_fit)

# Extract residuals from the model
#residuals <- resid(SLA_model_2)

# Q-Q plot for normality
#qqnorm(residuals) #okay..


# Histogram for normality
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #definitely a few ou

# Compare models using AIC, BIC, and log-likelihood
#SLA_model_comparison <- data.frame(
  #Model = c("SLA_model", "SLA_model_2"),
  #AIC = c(AIC(SLA_model), AIC(SLA_model_2)),
  #BIC = c(BIC(SLA_model), BIC(SLA_model_2)),
  #LogLikelihood = c(logLik(SLA_model), logLik(SLA_model_2))
#)

# Print model comparison
#print(SLA_model_comparison)

#model 2 is the best
```

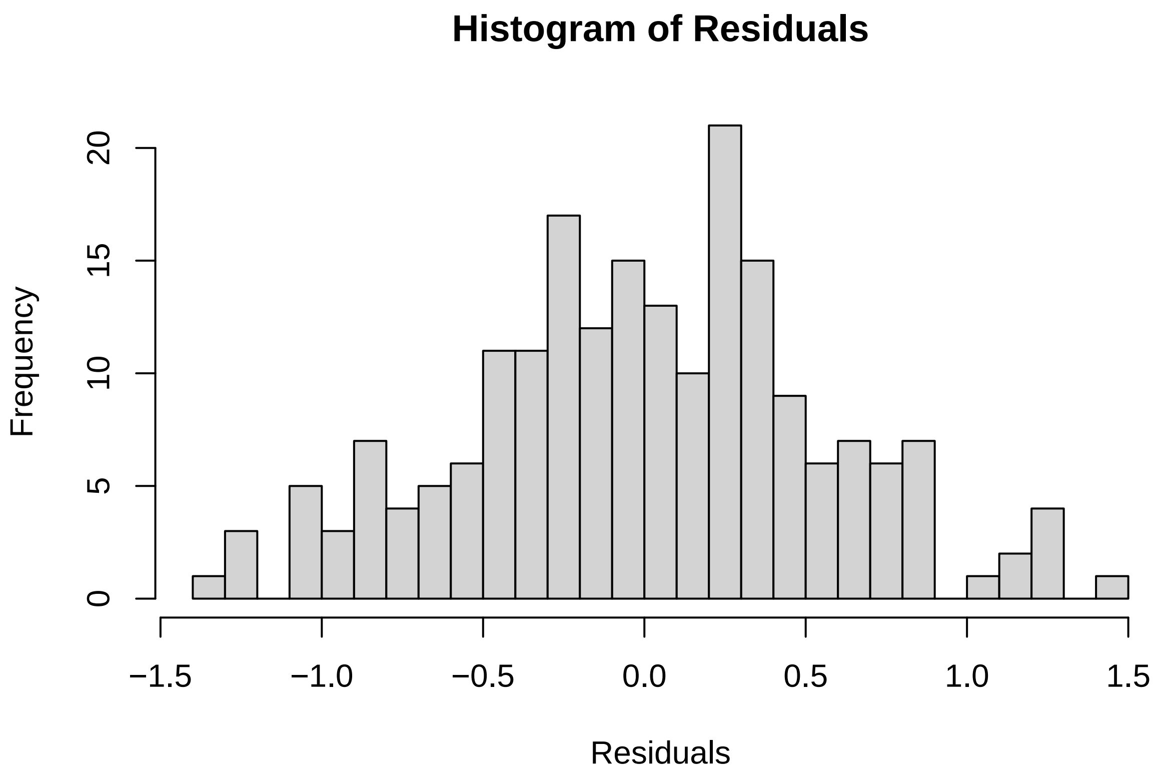
```
# Create the mixed model for mean seed mass, log transformed mean seed mass
#LMA_model <- lmer(LMA_SEG_centered ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit) #Singular

# Create the mixed model for mean seed mass, log transformed mean seed mass
#LMA_model <- lmer(LMA_SEG_centered ~ (1 | Donor), data = AC_23_fit) #Singular
```

```
# Create the mixed model for d13C, log transformed mean seed mass
#d13C_model <- lmer(d13C_SEG_centered ~ (1 | Recipient) + (1 | Donor), data = AC_23_fit) #Singular

# Extract residuals from the model
#residuals <- resid(d13C_model)

# Q-Q plot for normality
#qqnorm(residuals) #good enough


# Histogram for normality
```

```r
#hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish

# Create the mixed model for d13C, log transformed mean seed mass
d13C_model_2 <- lmer(d13C_SEG_centered ~ (1 |Transect) + (1 | Donor), data = AC_23_fit)

# Extract residuals from the model
residuals <- resid(d13C_model_2)

# Q-Q plot for normality
qqnorm(residuals) #good enough
```



```r
# Histogram for normality
hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish
```



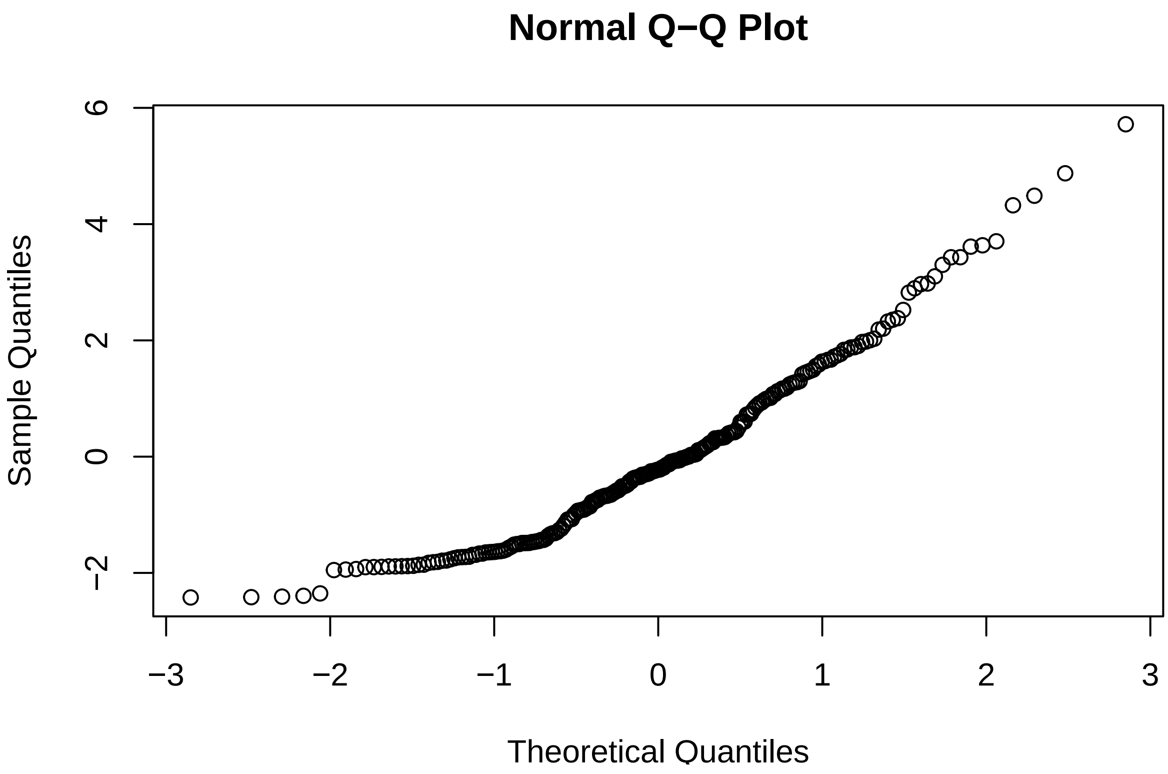
```
rand(d13C_model_2)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## d13C_SEG_centered ~ (1 | Transect) + (1 | Donor)
##                 npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>             4 -193.09 394.18
## (1 | Transect)     3 -217.30 440.61 48.429  1  3.424e-12 ***
## (1 | Donor)        3 -194.25 394.51  2.334  1     0.1265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

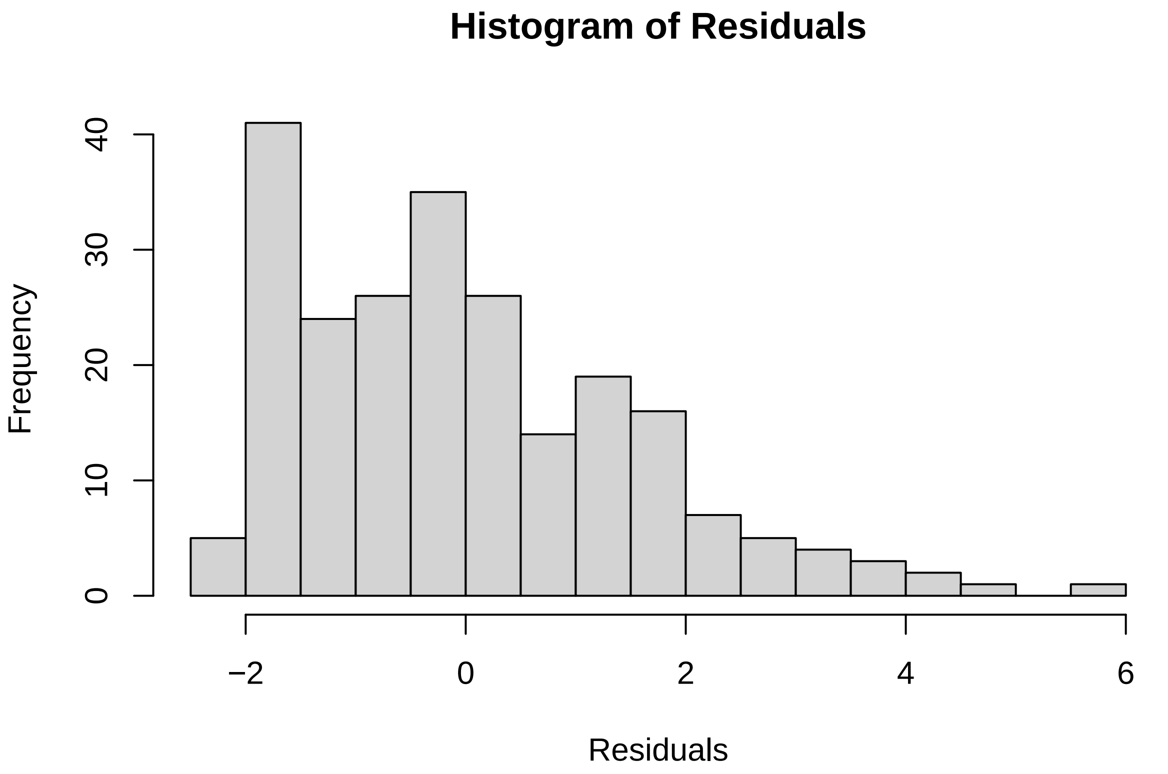
```
# Compare models using AIC, BIC, and log-likelihood
#d13C_model_comparison <- data.frame(
  #Model = c("d13C_model", "d13C_model_2"),
  #AIC = c(AIC(d13C_model), AIC(d13C_model_2)),
  #BIC = c(BIC(d13C_model), BIC(d13C_model_2)),
  #LogLikelihood = c(logLik(d13C_model), logLik(d13C_model_2))
#)

# Print model comparison
#print(d13C_model_comparison)

#model 2 is the best
```

```
# Create the mixed model for d13C, log transformed mean seed mass
est_fitness_model <- lmer(est_fitness_centered ~ (1|Transect) + (1 | Recipient) + (1 | Donor), data = A

# Extract residuals from the model
residuals <- resid(est_fitness_model)
```

```r
# Q-Q plot for normality
qqnorm(residuals) #good enough
```



```r
# Histogram for normality
hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish
```



```
# Test the significance of the random effects
rand(est_fitness_model) #all Random effects significantly improve the models fit!
```

```
## boundary (singular) fit: see help('isSingular')

## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## est_fitness_centered ~ (1 | Transect) + (1 | Recipient) + (1 | Donor)
##                 npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>             5 -441.61 893.21
## (1 | Transect)     4 -443.62 895.25 4.0364  1    0.04453 *
## (1 | Recipient)    4 -441.66 891.33 0.1180  1    0.73122
## (1 | Donor)        4 -441.61 891.22 0.0116  1    0.91441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Function to round values to a specified number of significant digits
round_df <- function(df, digits) {
  df[] <- lapply(df, function(x) if(is.numeric(x)) signif(x, digits) else x)
  return(df)
}

# Function to extract variance components and calculate required values
calculate_variances <- function(model, trait_name) {
  var_components <- as.data.frame(VarCorr(model))

  # Initialize variables
  V_mat <- NA
  V_sd_mat <- NA
  Va_mat <- NA
  V_pat <- NA
  V_sd_pat <- NA
  Va_pat <- NA
  res_var <- NA
  Vp <- NA
  Vp_sd <- NA
  h2 <- NA

  # Check if "Recipient" is included as a random effect
  if ("Recipient" %in% var_components$grp) {
    V_mat <- var_components$vcov[var_components$grp == "Recipient"]
    V_sd_mat <- sqrt(V_mat)
    Va_mat <- V_mat * 4
  }

  # Calculate other variance components
  if ("Donor" %in% var_components$grp) {
    V_pat <- var_components$vcov[var_components$grp == "Donor"]
    V_sd_pat <- sqrt(V_pat)
    Va_pat <- V_pat * 4
  }

  if ("Residual" %in% var_components$grp) {
    res_var <- var_components$vcov[var_components$grp == "Residual"]
  }

  # Calculate total phenotypic variance and narrow-sense heritability if components are available
  if (!is.na(Va_mat) & !is.na(Va_pat) & !is.na(res_var)) {
    Vp <- Va_mat + Va_pat + res_var
    Vp_sd <- sqrt(Vp)
    h2 <- Va_pat / Vp  # assumed calculation
  }

  # Extract the number of observations
  n_obs <- nobs(model)

  # Create the dataframe and add the traits column
  df <- data.frame(traits = trait_name, V_mat, V_sd_mat, Va_mat, V_pat, V_sd_pat, Va_pat, Vp, Vp_sd, h2

  # Round the dataframe values to four significant digits
  df <- round_df(df, 4)

  return(df)
}

# Calculate variances for each model and add trait names
corolla_variances <- calculate_variances(corolla_model, 'corolla_diameter')
skel_variances <- calculate_variances(skel_model_2, "skel_biomass_mg")
#est_fecundity_variances <- calculate_variances(est_fecundity_model, "estimated_fecundity")
#msm_variances <- calculate_variances(msm_model_2, "mean_seed_mass")
#SLA_variances <- calculate_variances(SLA_model, "SLA")
#LMA_variances <- calculate_variances(LMA_model, "LMA")
d13C_variances <- calculate_variances(d13C_model_2, "delta_C_13")
est_fitness_variances <- calculate_variances(est_fitness_model, "est_fitness")


# Combine the results into a single dataframe
variance_AC_2023_G1 <- rbind(
  corolla_variances,
  skel_variances,
  d13C_variances,
  est_fitness_variances
)

# Print the dataframe
print(variance_AC_2023_G1)
```

```
##             traits   V_mat V_sd_mat Va_mat     V_pat V_sd_pat   Va_pat     Vp
## 1 corolla_diameter 2.04800   1.4310 8.1920 0.3126000  0.55910 1.251000 14.530
## 2  skel_biomass_mg      NA       NA     NA 0.0009319  0.03053 0.003728     NA
## 3       delta_C_13      NA       NA     NA 0.0256700  0.16020 0.102700     NA
## 4      est_fitness 0.07652   0.2766 0.3061 0.0140600  0.11860 0.056240  2.944
##   Vp_sd     h2 n_obs
## 1 3.811 0.0861   161
## 2    NA     NA   212
## 3    NA     NA   202
## 4 1.716 0.0191   229
```

```
#Save the csv file if you want
write_csv(x = variance_AC_2023_G1, here::here("data_sheets", "compiled_sheets", "AC_Va_h2_R_2023.csv"))
```