AC_Va_h2_R_2022

Helen Payne

2024-06-18

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

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

```
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
```

```
library(lmerTest)
```

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

```
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
```

##########**2022**##########

Read in the data:

```
AC_22_fit <- read_csv(here::here("data_sheets", "compiled_sheets", "AC_mastersheet_Fitness-mains_2022.cs
```

```
## Rows: 319 Columns: 56
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): Donor, Recipient, Gen, Replicated
## dbl  (43): Year, Sequence, Cohort, Block, Transect, Plant_ID, F_plant_ID, fl...
## num   (1): F_multi
## lgl   (3): F_plant, Rep_FitP, any_FitP
## date  (5): Germ_Date, Sow_Date, Plant_Date, FFD, LFD
##
## 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_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_22_fit <- AC_22_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_22_fit <- AC_22_fit %>%
  mutate(est_fitness = prop_surv_to_flower * est_fecundity)
```

```
#log scale traits that are highly skewed
AC_22_fit$skel_dryweight_mg_SEG <- log(AC_22_fit$skel_dryweight_mg_SEG)
AC_22_fit$est_fecundity <- sqrt(AC_22_fit$est_fecundity + 1)
AC_22_fit$SLA_SEG <- log(AC_22_fit$SLA_SEG)
```

```r
AC_22_fit$est_fitness <- sqrt(AC_22_fit$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_22_fit[[trait]], na.rm = TRUE)
  AC_22_fit[[paste0(trait, "_centered")]] <- AC_22_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_22_fit)
```

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

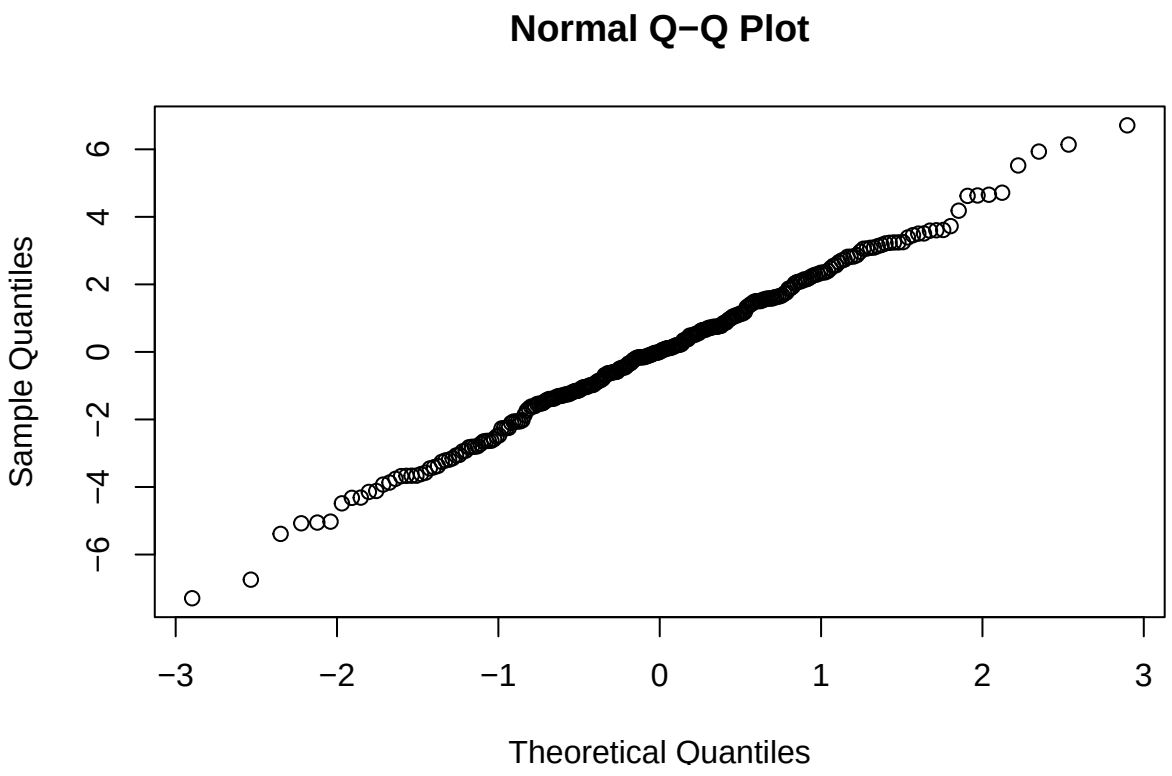
```r
# Check if the model is singular
isSingular(corolla_model) #TRUE, likely due to over-fitting, will not include
```

```
## [1] TRUE
```

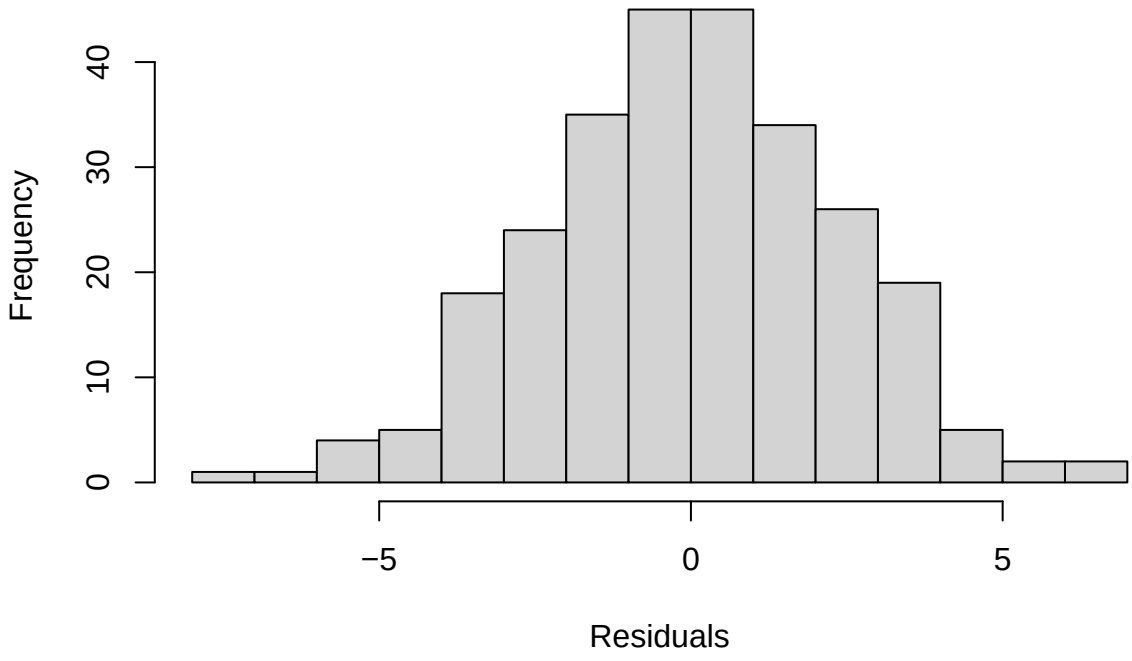
```r
corolla_model_2 <- lmer(corolla_diam_mm_SEG_centered ~ (1 | Transect) + (1 | Donor), data = AC_22_fit)

rand(corolla_model_2)#including Donor and Transect as random effects significantly improves the models
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## corolla_diam_mm_SEG_centered ~ (1 | Transect) + (1 | Donor)
##                 npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>             4 -634.65 1277.3
## (1 | Transect)     3 -637.93 1281.9 6.5473  1   0.010504 *
## (1 | Donor)        3 -639.51 1285.0 9.7201  1   0.001823 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

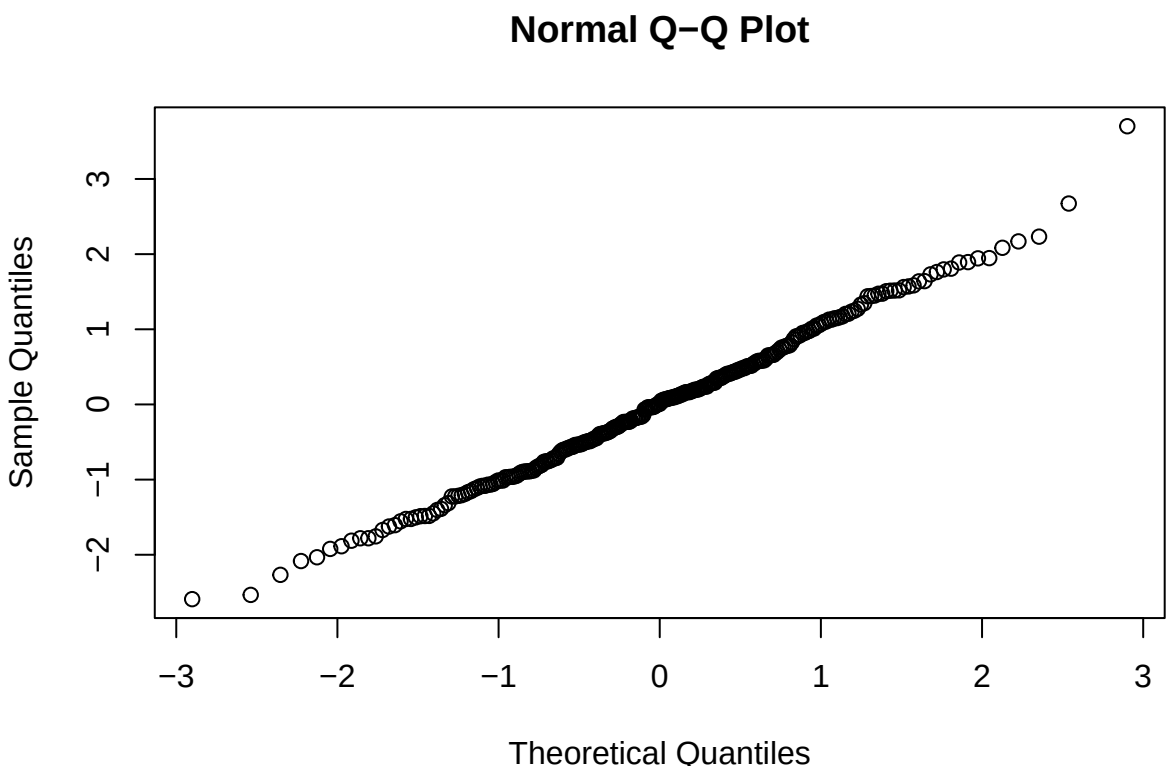



```r
# Histogram for normality
hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal
```

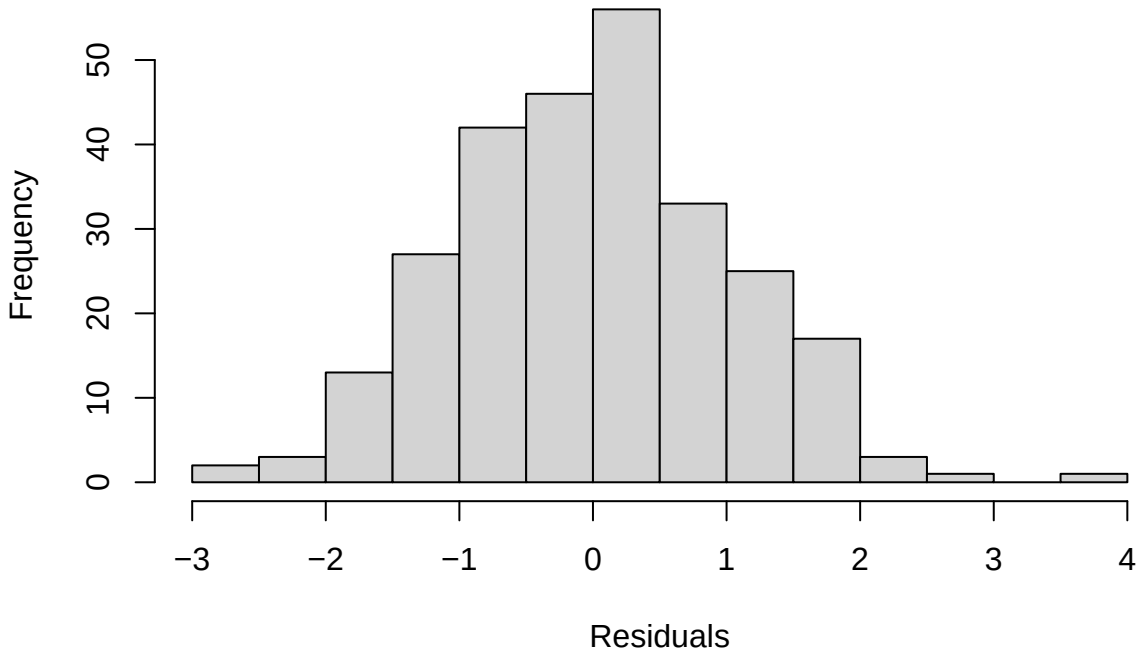



```r
#use Model 2 (Donor)
```

```r
# 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_22_fit)

skel_model <- lmer((skel_dryweight_mg_SEG_centered) ~ (1 | Donor), data = AC_22_fit)

# Extract residuals from the model
residuals <- resid(skel_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
```




```
# Test the significance of the random effects
#rand(skel_model) #none of these random effects significantly improve the model's fit


# Create the mixed model for skeleton weight, with skeleton weight log transformed
#skel_model_2 <- lmer(log(skel_dryweight_mg_SEG_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_22

# 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 2 is the best
```

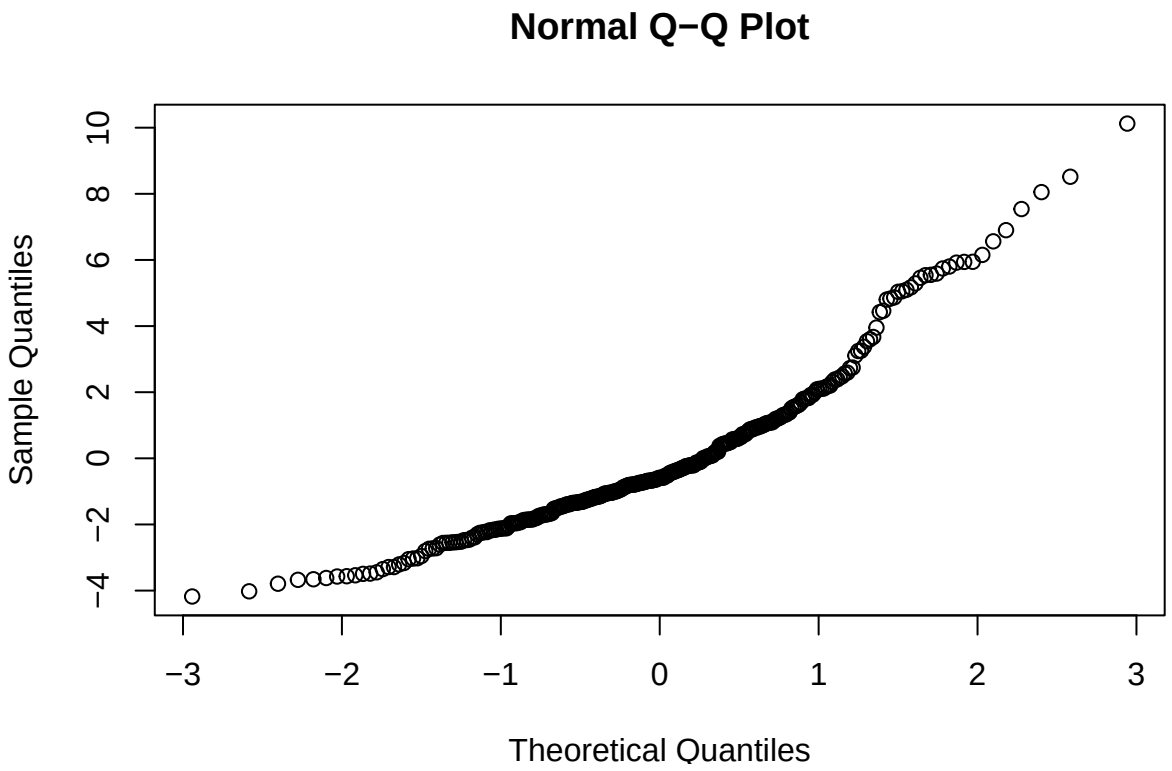
```r
# Create the mixed model for flowering duration
fl_duration_model <- lmer(fl_duration_centered ~ (1 | Recipient) + (1 | Donor), data = AC_22_fit)
```

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

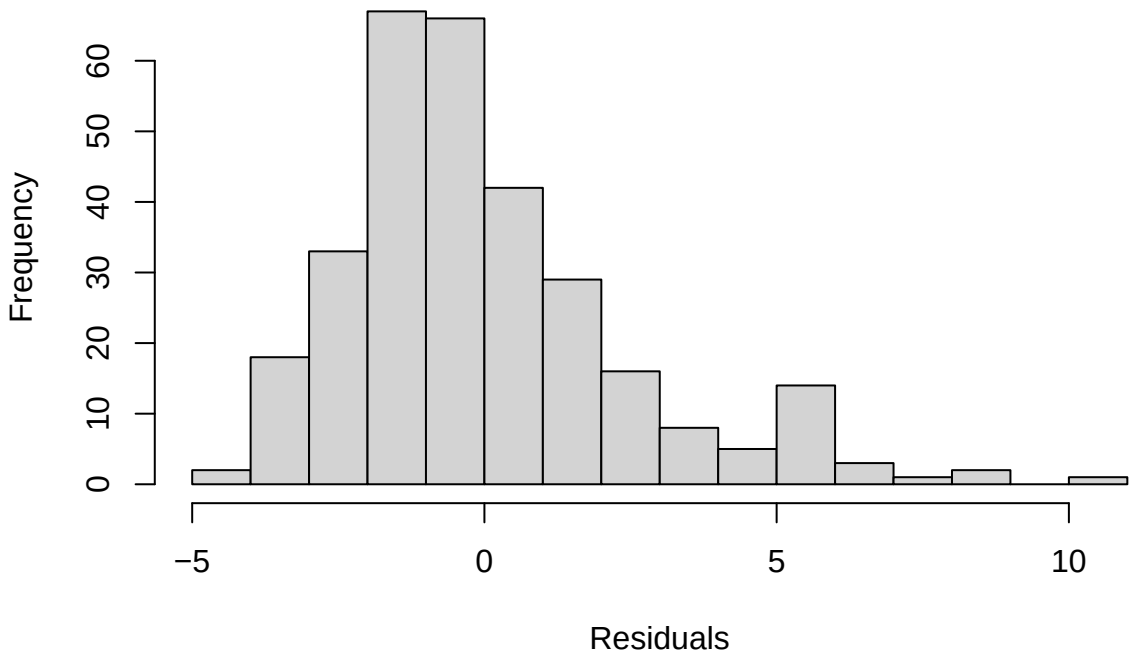
```r
# Create the mixed model for flowering duration
fl_duration_model <- lmer(fl_duration_centered ~ (1 | Donor), data = AC_22_fit)
```

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

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

est_fecundity_model <- lmer((est_fecundity_centered) ~ (1|Transect) + (1 | Donor), data = AC_22_fit)

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

# 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
```



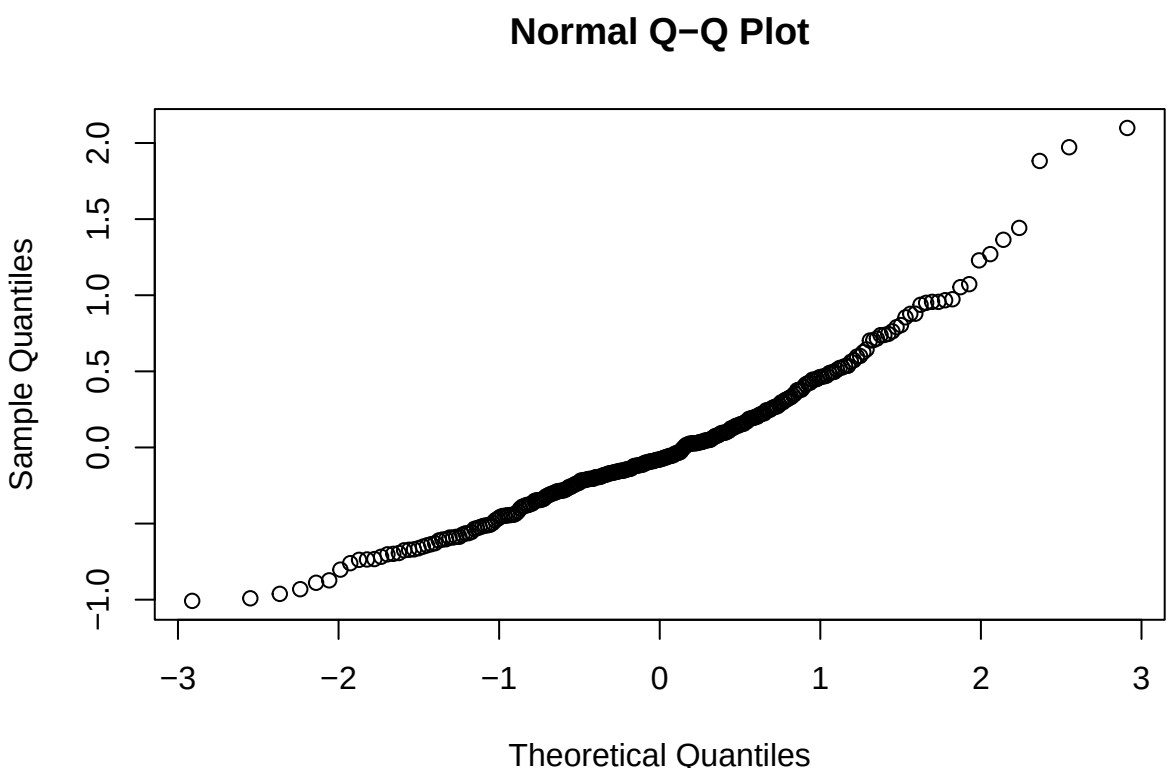

```
# Test the significance of the random effects
rand(est_fecundity_model) #Transect random effects significantly improve the model's fit
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## (est_fecundity_centered) ~ (1 | Transect) + (1 | Donor)
##                 npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>             4 -728.42 1464.8
## (1 | Transect)     3 -730.30 1466.6 3.7585  1    0.05254 .
## (1 | Donor)        3 -729.42 1464.8 2.0073  1    0.15654
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

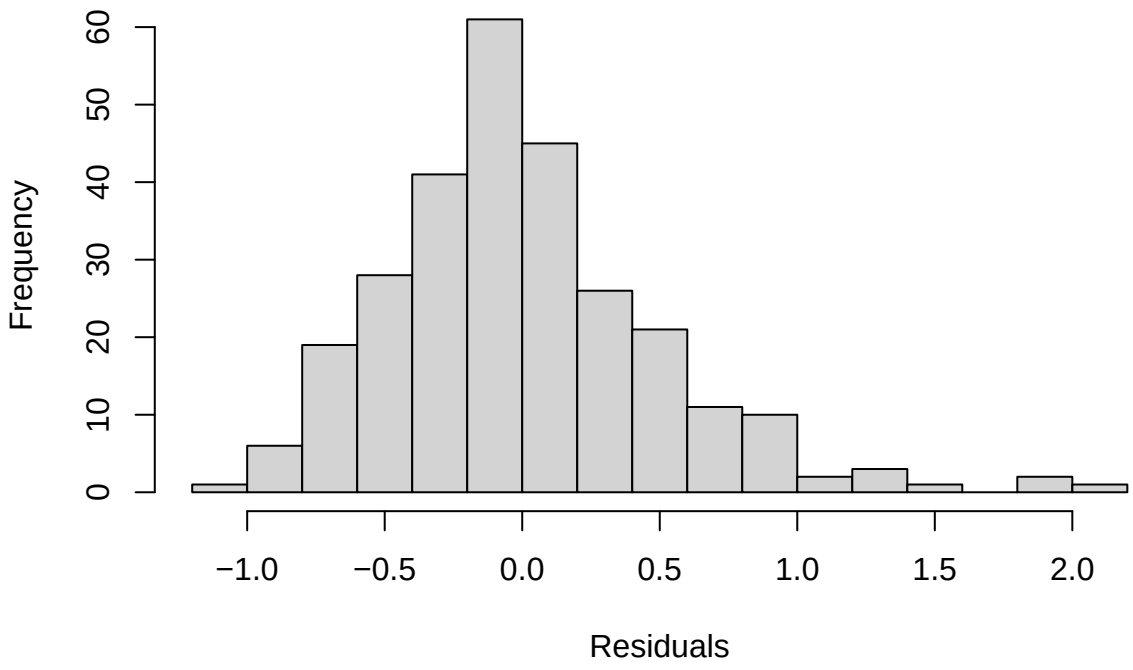
```
# Create the mixed model for estimated fecundity, sqrt transforming estimated fecundity
#est_fecundity_model_2 <- lmer(sqrt(est_fecundity_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_

# 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
#anova(est_fecundity_model, est_fecundity_model_2)
```

```r
# 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
```

```r
# Create the mixed model for mean seed mass, log transformed mean seed mass
msm_model <- lmer((msm_all_centered) ~ (1| Transect) + (1 | Recipient) + (1 | Donor), data = AC_22_fit)

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

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



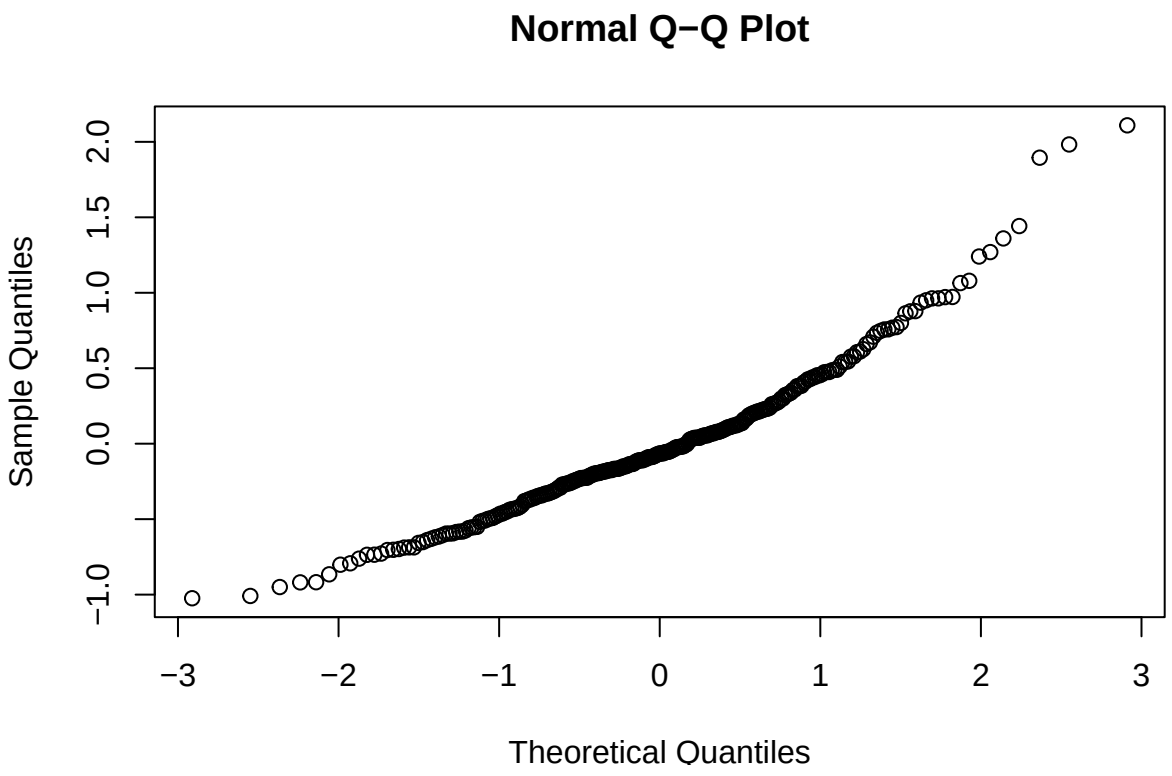

```r
# Histogram for normality
hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals") #normal-ish
```



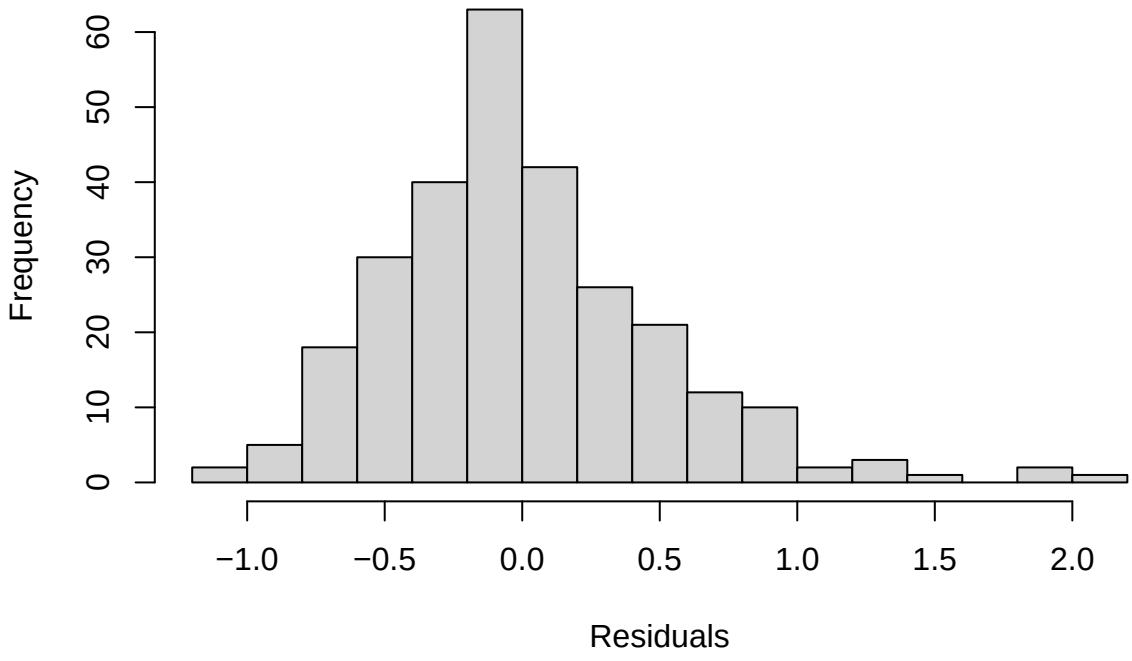

```
# Test the significance of the random effects
rand(msm_model) #none of these random effects significantly improve the model's fit
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## (msm_all_centered) ~ (1 | Transect) + (1 | Recipient) + (1 | Donor)
##                  npar  logLik    AIC     LRT Df Pr(>Chisq)
## <none>              5 -243.96 497.91
## (1 | Transect)      4 -244.05 496.11 0.19422  1     0.6594
## (1 | Recipient)     4 -244.95 497.90 1.98898  1     0.1584
## (1 | Donor)         4 -244.46 496.92 1.01157  1     0.3145
```

```
# Create the mixed model for mean seed mass, log transformed mean seed mass
msm_model_2 <- lmer((msm_all_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_22_fit)

# Extract residuals from the model
residuals <- resid(msm_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
```




```r
#compare models
anova(msm_model, msm_model_2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: AC_22_fit
## Models:
## msm_model_2: (msm_all_centered) ~ (1 | Recipient) + (1 | Donor)
## msm_model: (msm_all_centered) ~ (1 | Transect) + (1 | Recipient) + (1 | Donor)
##             npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## msm_model_2    4 491.65 506.16 -241.82   483.65
## msm_model      5 493.56 511.70 -241.78   483.56  0.09  1     0.7642
```

```
# 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)
```

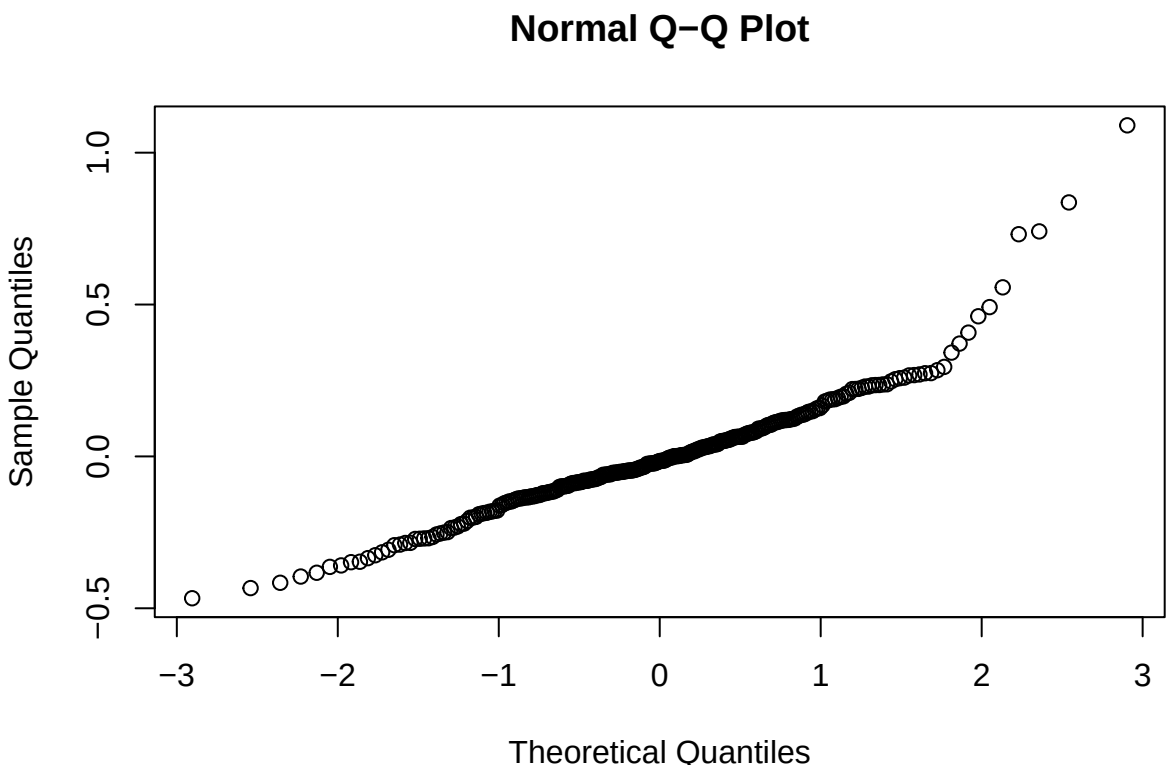
```
##         Model      AIC      BIC LogLikelihood
## 1   msm_model 497.9109 516.0490     -243.9554
## 2 msm_model_2 496.1051 510.6156     -244.0526
```

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

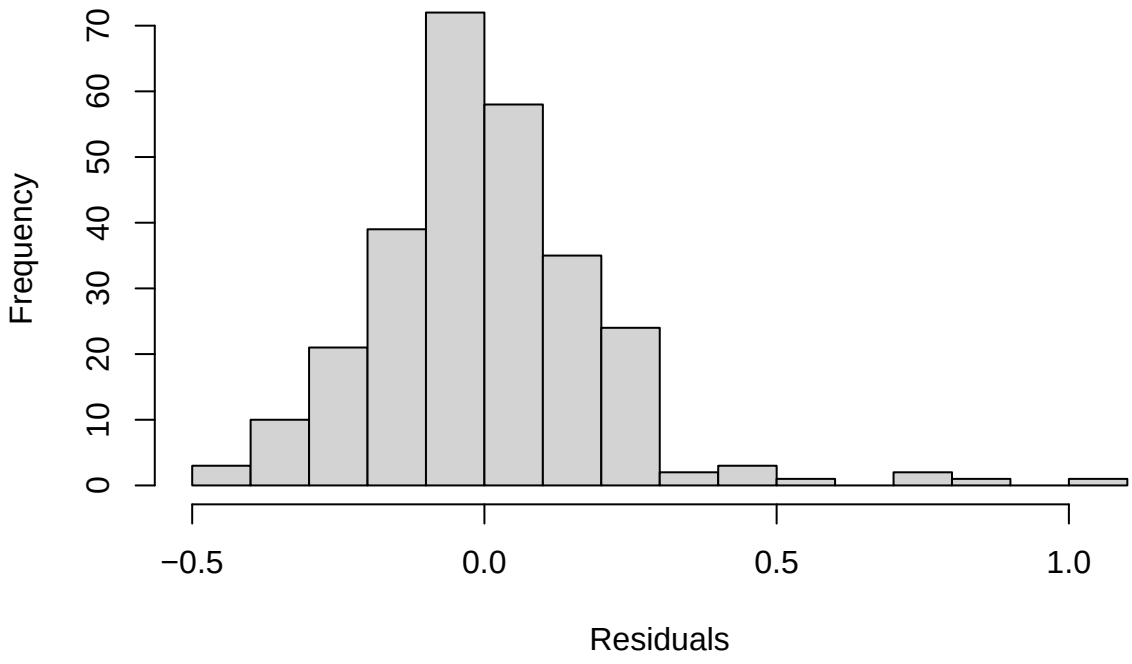
```
# Create the mixed model for SLA
SLA_model <- lmer((SLA_SEG_centered) ~ (1|Transect) + (1 | Recipient) + (1 | Donor), data = AC_22_fit)
```

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

```
# 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 out
```




```
# Test the significance of the random effects
rand(SLA_model) #Transect and Recipient significantly improve the models fit
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## (SLA_SEG_centered) ~ (1 | Transect) + (1 | Recipient) + (1 | Donor)
##                 npar  logLik     AIC    LRT Df Pr(>Chisq)
## <none>             5  24.024 -38.048
## (1 | Transect)     4 -18.475  44.950 84.999  1     <2e-16 ***
## (1 | Recipient)    4  23.887 -39.774  0.274  1     0.6006
## (1 | Donor)        4  24.024 -40.048  0.000  1     1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##adding in transect##
# Create the mixed model for SLA
#SLA_model_2 <- lmer(log(SLA_SEG_centered) ~ (1 | Recipient) + (1 | Donor), data = AC_22_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 1 is the best
```

```
# Create the mixed model for mean seed mass, log transformed mean seed mass
LMA_model <- lmer(LMA_SEG_centered ~ (1|Transect) + (1 | Recipient) + (1 | Donor), data = AC_22_fit)
```

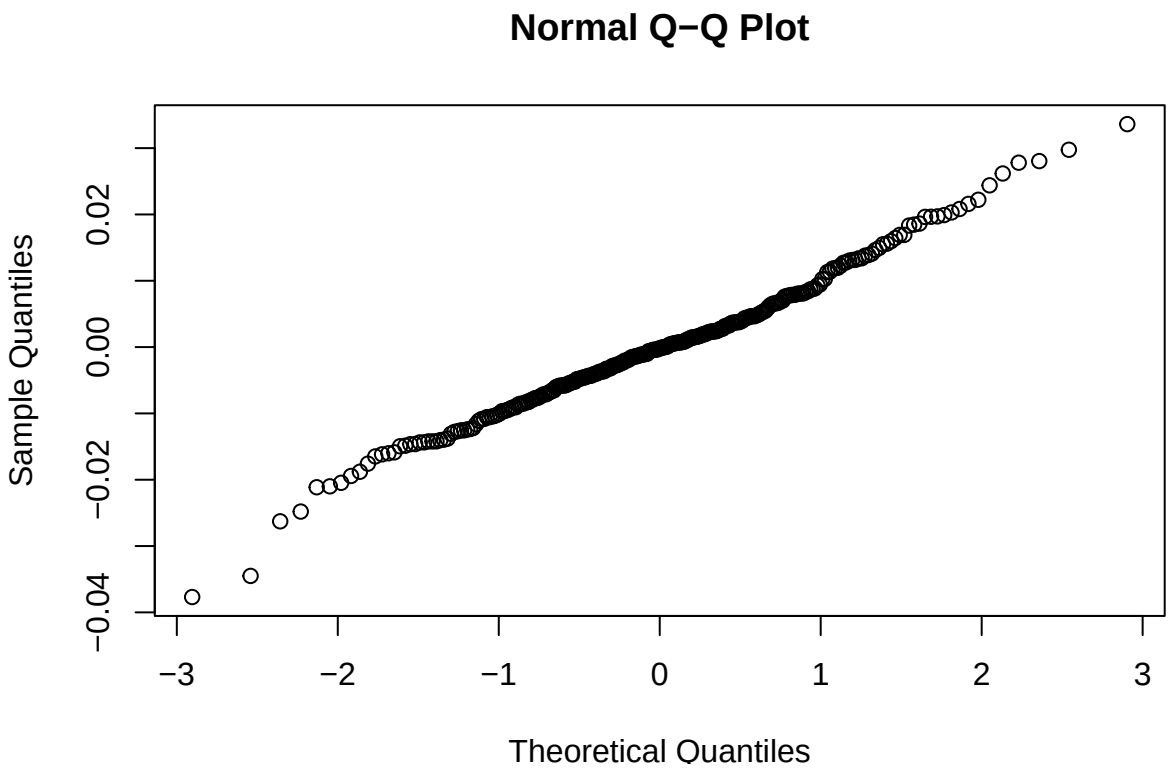
```
## boundary (singular) fit: see help('isSingular')
```

```
LMA_model
```

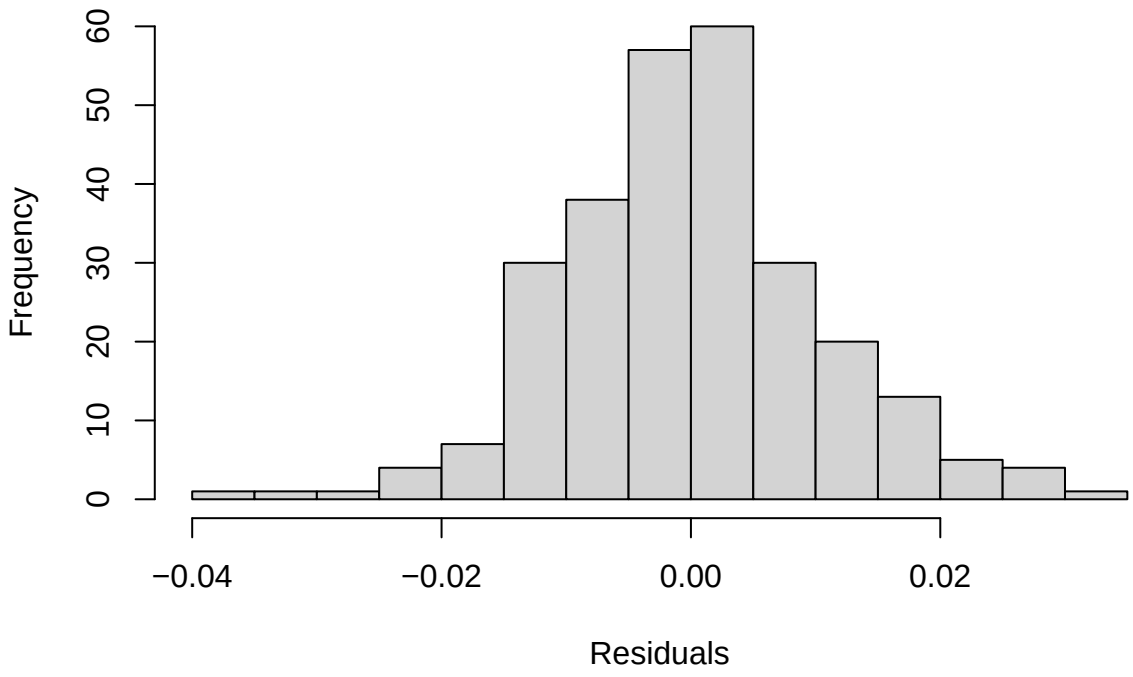
```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: LMA_SEG_centered ~ (1 | Transect) + (1 | Recipient) + (1 | Donor)
##    Data: AC_22_fit
## REML criterion at convergence: -1628.403
## Random effects:
##  Groups    Name        Std.Dev.
##  Recipient (Intercept) 2.522e-03
##  Donor     (Intercept) 1.018e-06
##  Transect  (Intercept) 8.844e-03
```

```
##  Residual             1.110e-02
## Number of obs: 272, groups:  Recipient, 107; Donor, 40; Transect, 9
## Fixed Effects:
## (Intercept)
##    -0.00139
## optimizer (nloptwrap) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings
```

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

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




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



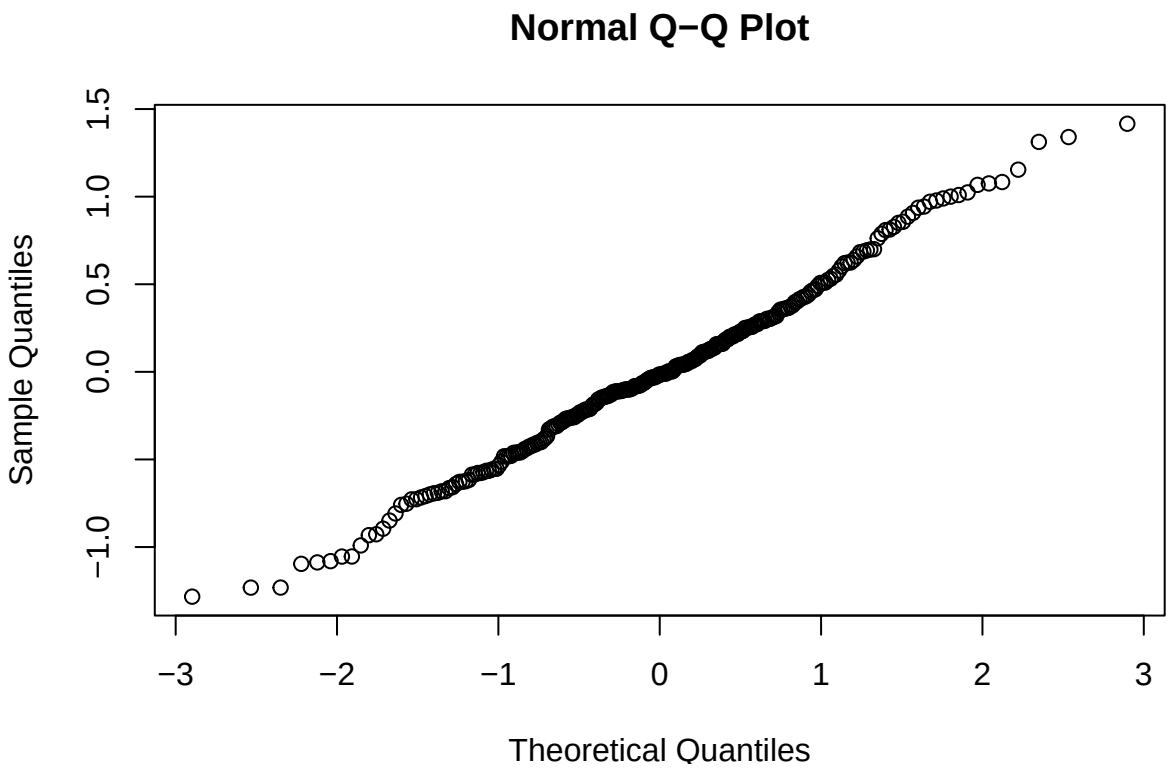

```
# Test the significance of the random effects
rand(LMA_model) #Transect significantly improves the models fit
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## LMA_SEG_centered ~ (1 | Transect) + (1 | Recipient) + (1 | Donor)
##                  npar logLik     AIC    LRT Df Pr(>Chisq)
## <none>              5 814.20 -1618.4
## (1 | Transect)      4 771.38 -1534.8 85.651  1     <2e-16 ***
## (1 | Recipient)     4 813.96 -1619.9  0.476  1     0.4902
## (1 | Donor)         4 814.20 -1620.4  0.000  1     1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

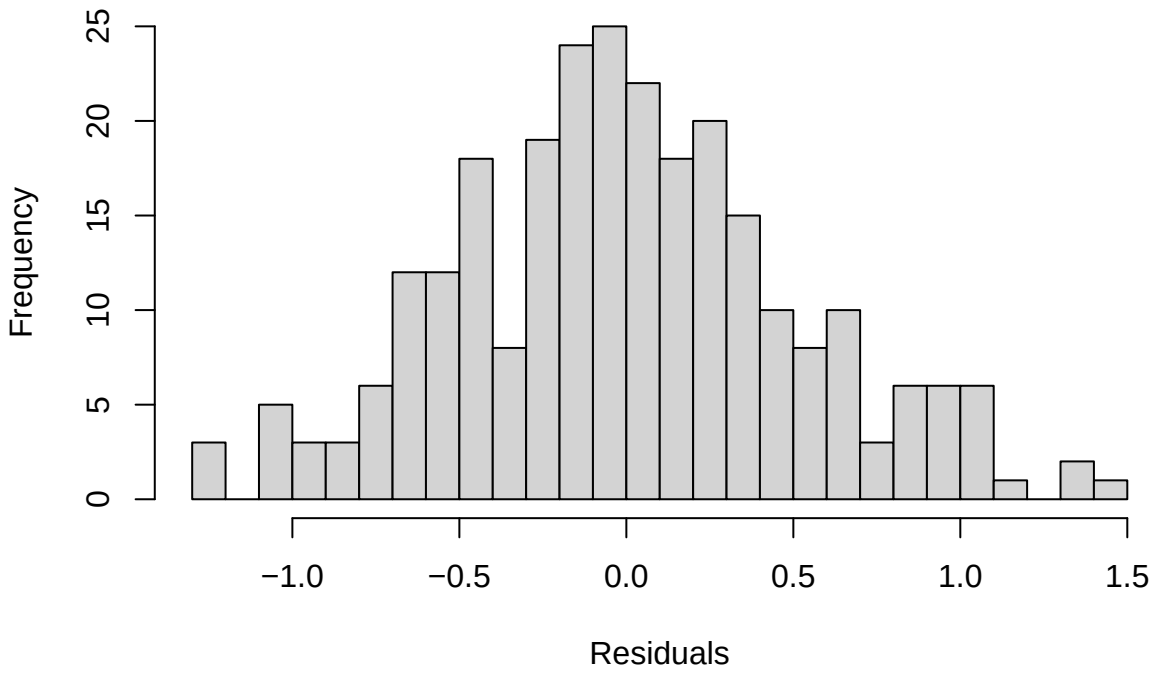
```
# Create the mixed model for d13C, log transformed mean seed mass
#LMA_model_2 <- lmer(LMA_SEG_centered ~ (1 | Recipient) + (1 | Donor), data = AC_22_fit)

# Extract residuals from the model
#residuals <- resid(LMA_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
#LMA_model_comparison <- data.frame(
```

```
 # Model = c("LMA_model", "LMA_model_2"),
 # AIC = c(AIC(LMA_model), AIC(LMA_model_2)),
 # BIC = c(BIC(LMA_model), BIC(LMA_model_2)),
 # LogLikelihood = c(logLik(LMA_model), logLik(LMA_model_2))
#)

# Print model comparison
#print(LMA_model_comparison)

#model 1 is the best
```

```
# Create the mixed model for d13C, log transformed mean seed mass
d13C_model <- lmer(d13C_SEG_centered ~ (1|Transect) + (1 | Recipient) + (1 | Donor), data = AC_22_fit)

# Extract residuals from the model
residuals <- resid(d13C_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
```




```
# Test the significance of the random effects
rand(d13C_model) #all Random effects significantly improve the models fit!
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## d13C_SEG_centered ~ (1 | Transect) + (1 | Recipient) + (1 | Donor)
##                  npar  logLik    AIC     LRT Df Pr(>Chisq)
## <none>              5 -283.52 577.05
## (1 | Transect)      4 -297.90 603.81 28.7566  1  8.207e-08 ***
## (1 | Recipient)     4 -285.46 578.93  3.8801  1  0.0488630 *
## (1 | Donor)         4 -289.25 586.50 11.4476  1  0.0007159 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

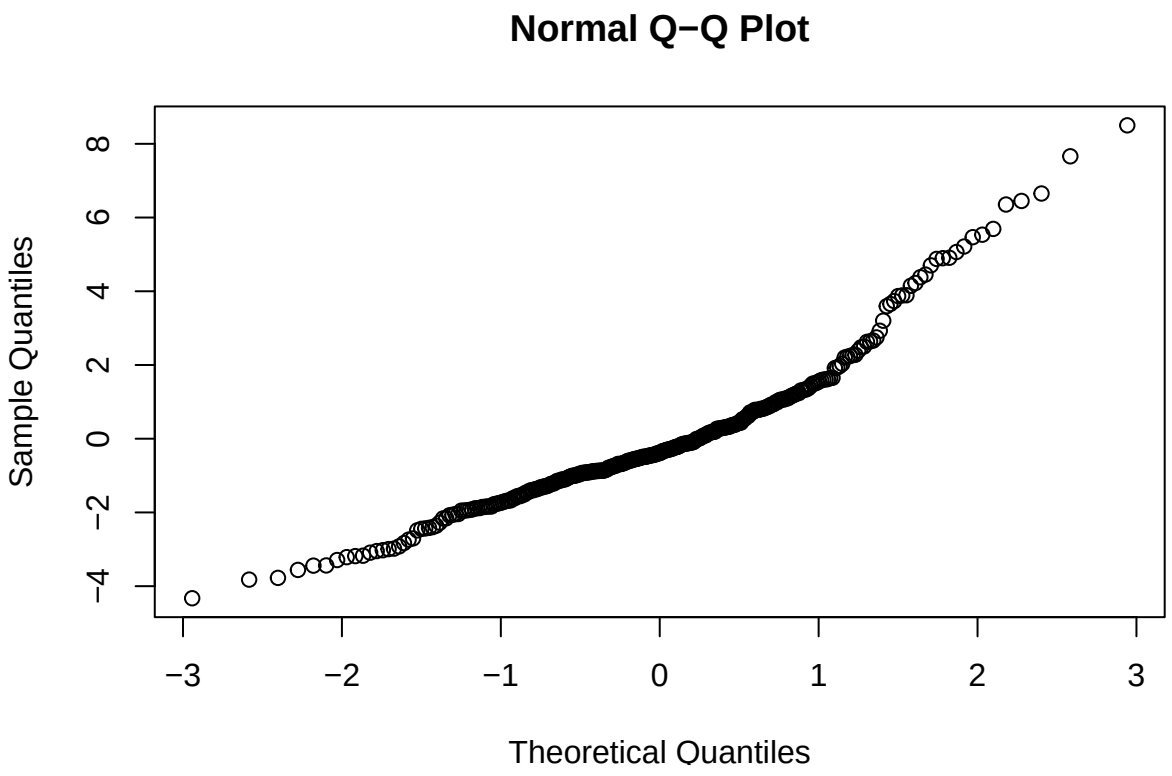
```
# Create the mixed model for d13C, log transformed mean seed mass
#d13C_model_2 <- lmer(d13C_SEG_centered ~ (1 | Recipient) + (1 | Donor), data = AC_22_fit)

# Extract residuals from the model
#residuals <- resid(d13C_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
#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 1 is the best
```

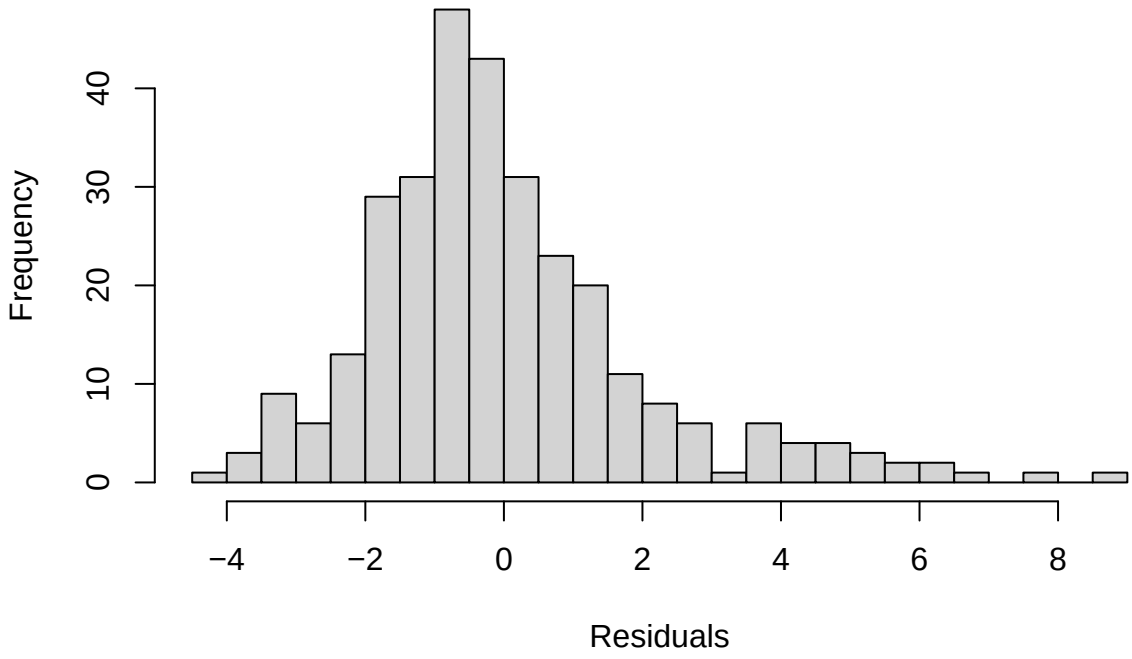
```
# Create the mixed model for d13C, log transformed mean seed mass
est_fitness_model <- lmer(est_fitness_centered ~ (1|Transect) + (1 | Recipient) + (1 | Donor), data = AC

est_fitness_model
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: est_fitness_centered ~ (1 | Transect) + (1 | Recipient) + (1 |
##     Donor)
##    Data: AC_22_fit
## REML criterion at convergence: 1429.342
## Random effects:
##  Groups    Name        Std.Dev.
##  Recipient (Intercept) 0.9849
##  Donor     (Intercept) 0.5889
##  Transect  (Intercept) 0.5203
##  Residual              2.2224
## Number of obs: 307, groups:  Recipient, 107; Donor, 40; Transect, 9
## Fixed Effects:
## (Intercept)
##    -0.02047
```

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

# 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
```




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

```
## 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 -714.67 1439.3
## (1 | Transect)     4 -717.41 1442.8 5.4871  1    0.01916 *
## (1 | Recipient)    4 -717.40 1442.8 5.4506  1    0.01956 *
## (1 | Donor)        4 -715.26 1438.5 1.1752  1    0.27833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 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
skel_variances <- calculate_variances(skel_model, "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, "delta_C_13")
est_fitness_variances <- calculate_variances(est_fitness_model, "est_fitness")

# Combine the results into a single dataframe
variance_AC_2022 <- rbind(
  skel_variances,
  d13C_variances,
  est_fecundity_variances,
  msm_variances,
  SLA_variances,
  est_fitness_variances
)

# Print the dataframe
print(variance_AC_2022)
```

```
##                traits    V_mat V_sd_mat   Va_mat   V_pat V_sd_pat  Va_pat
## 1     skel_biomass_mg       NA       NA       NA 0.10680   0.3268 0.42710
## 2          delta_C_13 0.068490  0.26170 0.273900 0.13300   0.3647 0.53190
## 3 estimated_fecundity       NA       NA       NA 0.31480   0.5610 1.25900
## 4      mean_seed_mass 0.037310  0.19320 0.149200 0.01856   0.1362 0.07425
## 5                 SLA 0.001721  0.04148 0.006884 0.00000   0.0000 0.00000
## 6         est_fitness 0.970000  0.98490 3.880000 0.34680   0.5889 1.38700
##          Vp  Vp_sd     h2 n_obs
## 1        NA     NA     NA   269
## 2  1.14500 1.0700 0.4645   266
## 3        NA     NA     NA   307
## 4  0.51090 0.7148 0.1453   278
## 5  0.04925 0.2219 0.0000   272
## 6 10.21000 3.1950 0.1359   307
```

```r
#Save the csv file if you want
write_csv(x = variance_AC_2022, here::here("data_sheets", "compiled_sheets", "AC_Va_h2_R_2022.csv"))
```