Programming in R Final Coursework

Student Number: 2185380

10/01/2022

```
knitr::opts_chunk$set(echo = TRUE)

# Loading libraries
libraries <- c("tidyverse",
               "knitr",
               "vroom",
               "DHARMa",
               "fitdistrplus",
               "glmmTMB",
               "gamlr",
               "performance",
               "see",
               "insight")
lapply(libraries, library, character.only = TRUE)
```

Reading, Manipulating, and Visualizing Data

```
#Using the Vroom package due to its speed and compatability with .gz compressed files
mass_raw <- vroom("~/R/bioinformatics/tb1_exam/data/mass_pu21373.tsv.gz")

nao_raw <- vroom("~/R/bioinformatics/tb1_exam/data/NAO_pu21373.tsv.gz")
```

I first wanted to calculate the annual mean NAO and summer NAO and merge the data.

```
# Making a new dataframe containing annual sNAO
sNAO <- nao_raw %>%
  filter(month == "June" | month == "July" | month == "August") %>%
  group_by(year) %>%
  mutate(snao = mean(NAO)) %>%
  dplyr::select(snao, year) %>%
  unique()      # Removing excess rows

# Joining the datasets
data_combined <- left_join(mass_raw, sNAO, by="year")
```

Now that all of the data was in one dataframe, I wanted to tidy it up.

```r
data_combined <- data_combined %>%
  na.omit(weight) %>%       # Removing rows which have no measurements for weight
  filter(weight > 0) %>%    # Removing rows which have negative weight values
  group_by(year) %>%
  mutate(mass_mean = mean(weight),    # Calculating annual mean mass for visualization
         std_time = (year - 1980),    # Converting year to a numeric vector
         site = as.factor(site))      # Converting site to a factor

head(data_combined, n=5)
```

```
## # A tibble: 5 x 11
## # Groups:   year [1]
##      ID   age sex   n_babies site   year pop_size weight    snao mass_mean
##   <dbl> <dbl> <chr>    <dbl> <fct> <dbl>    <dbl>  <dbl>   <dbl>     <dbl>
## 1     9     0 f            0 1      1980      146   3.01 -0.0488      3.18
## 2    57     0 m            0 3      1980      146   1.43 -0.0488      3.18
## 3    84     0 m            0 2      1980      146   4.24 -0.0488      3.18
## 4   101     0 f            2 3      1980      146   4.72 -0.0488      3.18
## 5   130     0 f            2 2      1980      146   3.63 -0.0488      3.18
## # ... with 1 more variable: std_time <dbl>
```

I wanted to visualize the data to see if there were any potential relationships between weight and other variables.

```r
# Average weight measurements over time

time_plot <- ggplot(data = data_combined, aes(x=year, y=mass_mean)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Annual Mean Weight over Time",
    x = "Time (Year)",
    y = "Mean Weight (kg)") +
  geom_smooth(method = "loess") +
  theme_bw()

# Relationship between annual weight measurements and annual snao measurements

snao_plot <- ggplot(data = data_combined, aes(x = snao, y = mass_mean)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Annual Mean Weight vs Annual sNAO",
    x = "NAO Index",
    y = "Weight (kg)") +
  geom_smooth(method = "loess") +
  theme_bw()

# Relationship between annual weight measurements and population size

pop_plot <- ggplot(data = data_combined, aes(x=pop_size, y=mass_mean)) +
  geom_point() +
  geom_line() +
```

```
  labs(
    title = "Annual Mean Weight vs Population",
    x = "Population",
    y = "Weight (kg)") +
  geom_smooth(method = "loess") +
  theme_bw()

# Relationship between weight and sex

sex_plot <- ggplot(data=data_combined, aes(x=as.factor(sex), y=weight), color =sex) +
  geom_boxplot(
    fill = c("#CC6666", "#9999CC"),
    color = "black",
    outlier.colour="red",
    outlier.fill="red"
  ) +
  labs(
    title = "Weight vs Sex",
    x = "Sex",
    y = "Weight (kg)"
  ) +
  scale_x_discrete(labels = c("Female", "Male")) +
  theme_bw() +
  theme(legend.position = "none")

# Relationship between weight and sample site

site_plot <- ggplot(data = data_combined,
                    aes(x = as.factor(site), y = weight, fill=as.factor(site))) +
  geom_boxplot() +
  labs(
    title = "Weight vs Site",
    x = "Site",
    y = "Weight (kg)"
  ) +
  theme_bw() +
  scale_color_brewer("Set2") +
  theme(legend.position = "none")

# Relationship between annual mean weight and number of offspring

babies_plot <- ggplot(data = data_combined,
                      aes(x=as.factor(n_babies), y=weight, fill=as.factor(n_babies))) +
  geom_boxplot() +
  labs(
    title = "Annual Mean Weight vs Number of Offspring",
    x = "Number of Offspring",
    y = "Weight (kg)"
  ) +
  theme_bw() +
  scale_color_brewer("Set2") +
  theme(legend.position = "none")
```

```
# Relationship between weight and sex

age_plot <- ggplot(data = data_combined,
                   aes(x = as.factor(age), y=weight, fill=as.factor(age))) +
  geom_boxplot() +
  labs(
  title = "Weight vs Age",
  x = "Age (Years)",
  y = "Weight (kg)"
  ) +
  scale_colour_steps() +
  theme_bw() +
  theme(legend.position = "none")
```

I wanted to first see, on average, how the mass of Soay sheep was trending over time, as well as its relationship with sNAO (`Figure 1`).

```
# Set global margins
par(mar = c(4, 4, .1, .1))
```

```
time_plot
snao_plot
```

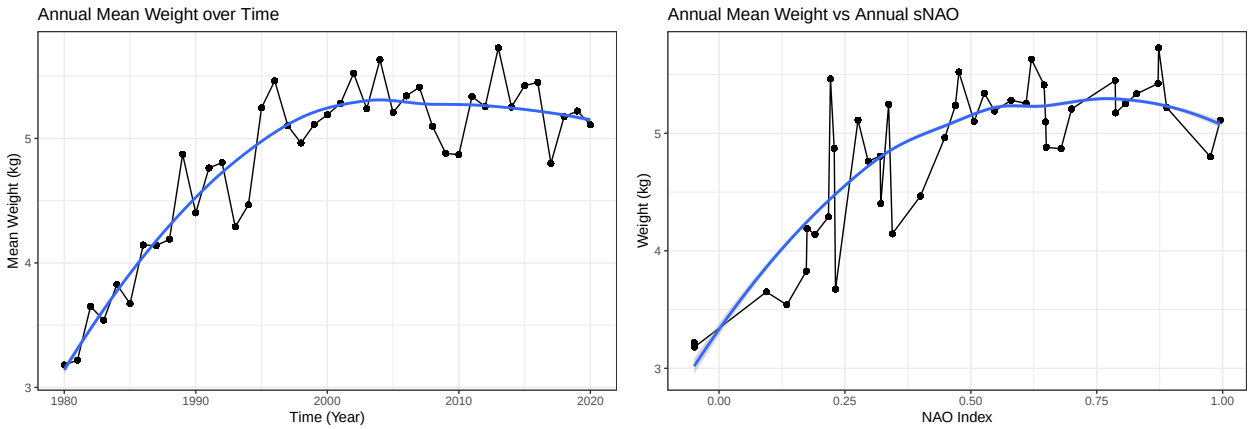


Figure 1: Figure 1.

Interesting relationships between other variables like sex, population number, and age were also seen (`Figure 2`).

```
sex_plot
pop_plot
age_plot
babies_plot
```

I wanted to check if there was any variability across the sampling sites (`Figure 3`).

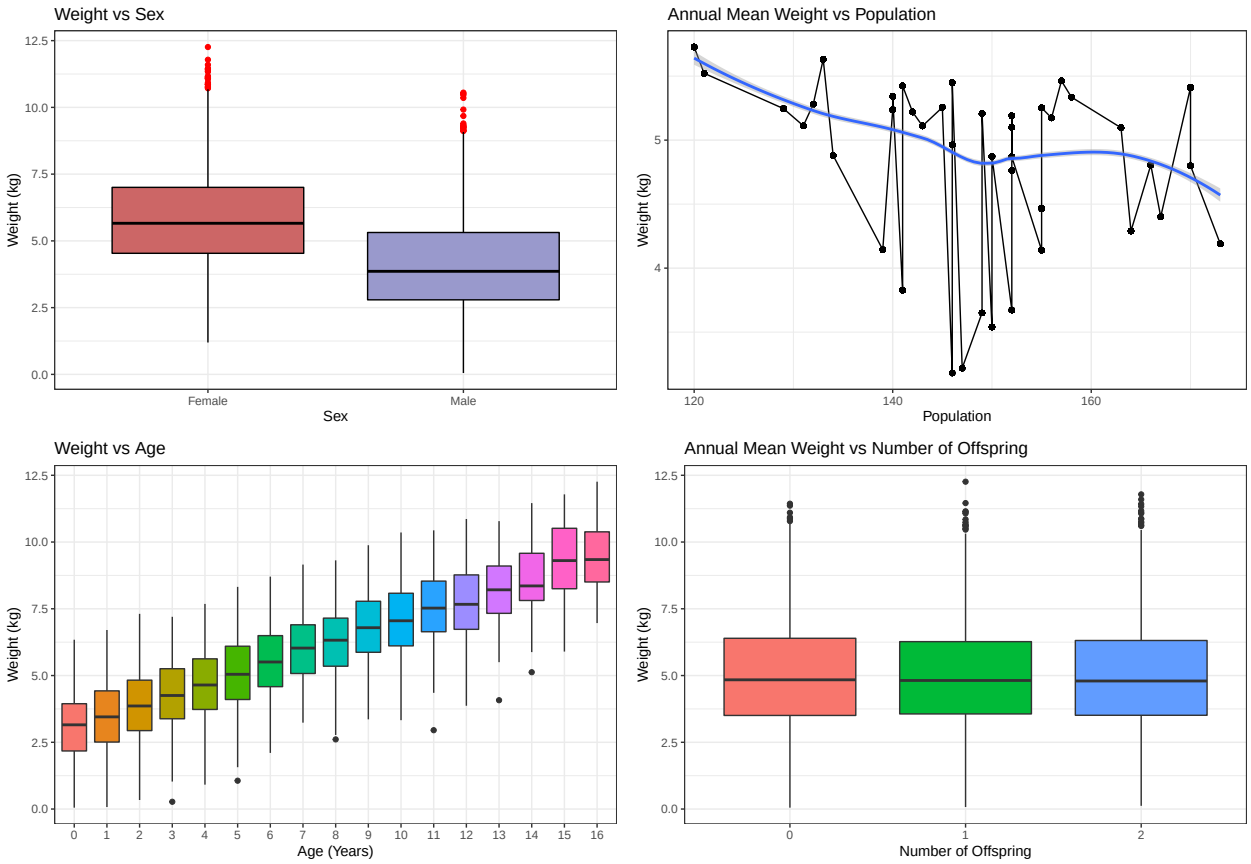


Figure 2: Figure 2.

```
site_plot
```

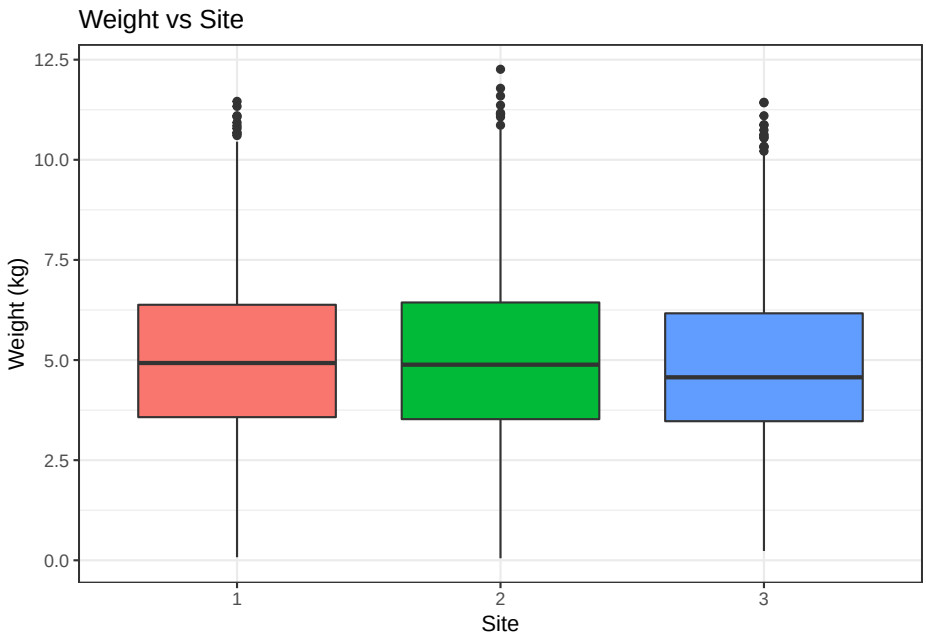


Figure 3: Figure 3.

Statistical Modelling

To determine the distribution of the `weight` measurements, I used the `fitdistrplus` package.

```
# Creating a fitdist object using the weight measurements
# Assuming Gaussian distribution as weight data is continuous
fit_gauss <- fitdist(data_combined$weight,
                     distr = "norm")

plot(fit_gauss)
```

```
# Outputting some goodness-of-fit statistics
gofstat(fit_gauss)
```

```
## Goodness-of-fit statistics
##                                1-mle-norm
## Kolmogorov-Smirnov statistic  0.03491963
## Cramer-von Mises statistic    2.09142073
## Anderson-Darling statistic   12.63121542
##
## Goodness-of-fit criteria
##                                 1-mle-norm
## Akaike's Information Criterion   24877.54
## Bayesian Information Criterion   24890.89
```

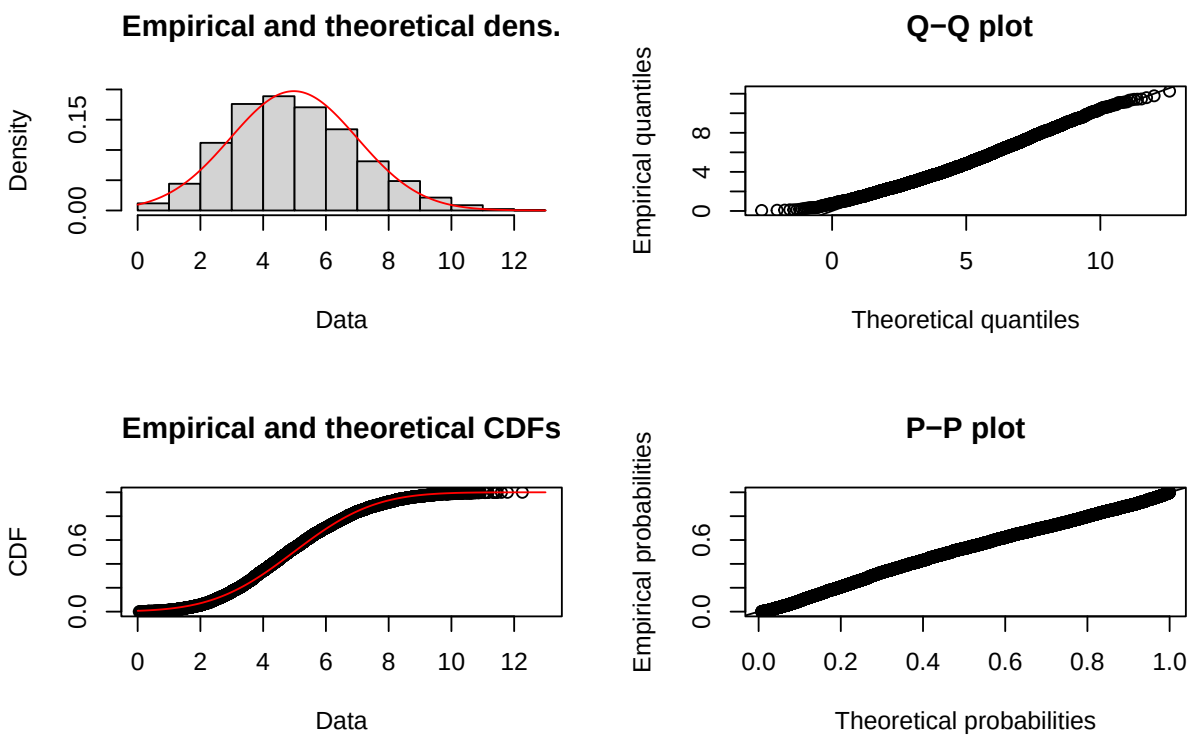


Figure 4: Figure 4.

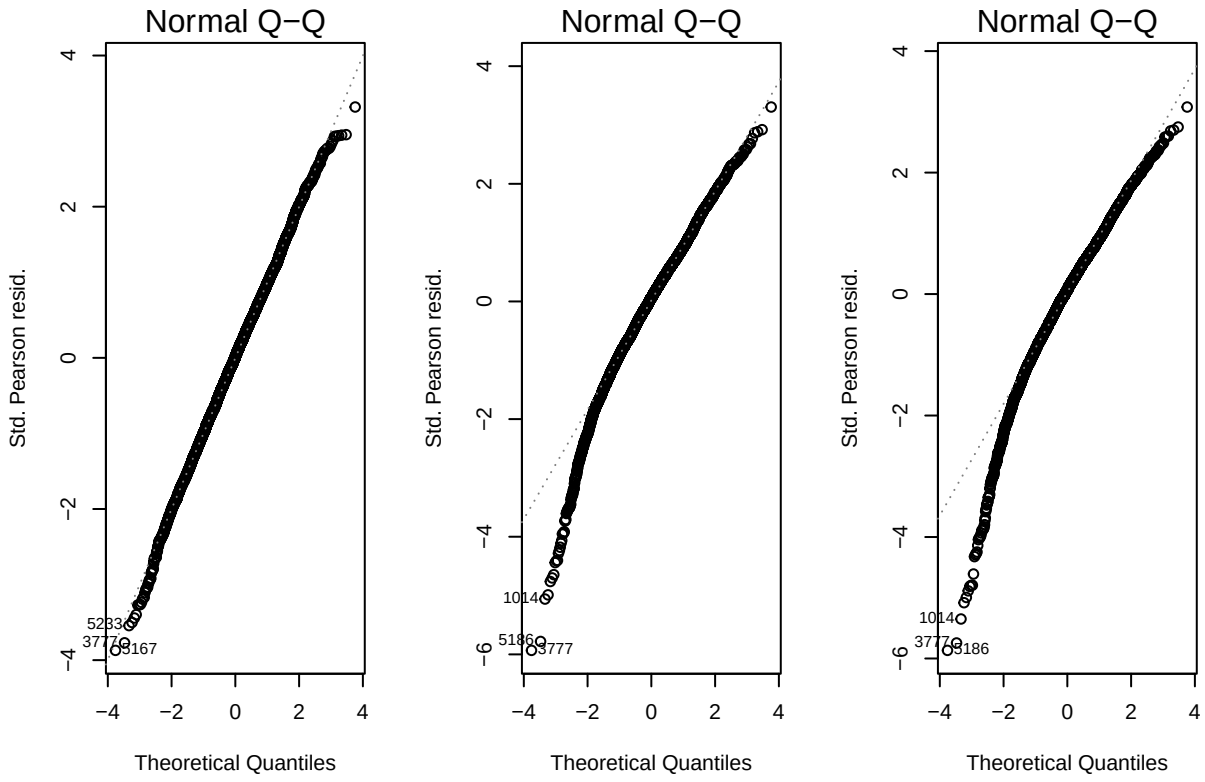
The data appears to follow the Gaussian distribution, although the Q-Q plot, density distributions, and stats indicate that the data is somewhat positively skewed.

I created some generalized linear models (GLMs) and compared different transformations and error distributions.

```
# Scaling variables with largely different scales
# Using a log(x + C) transformation for snao due to negative values
# Assuming the normal distribution
mod_gaus_1.1 <- glm(scale(weight) ~ log(snao + 1) + scale(std_time) + scale(age) + sex +
                      scale(pop_size) + site + n_babies,
                      data = data_combined,
                      family = "gaussian")

# Looking at different transformations
# Transforming variables to remove 0 and negative values
mod_gaus_1.2 <- glm(sqrt(weight) ~ log(snao + 1) + sqrt(std_time) + sqrt(age) + sex +
                      sqrt(pop_size) + site + n_babies,
                      data = data_combined,
                      family = gaussian(link="log"))

mod_gaus_1.3 <- update(mod_gaus_1.2, family = gaussian(link="inverse"))


# Comparing model fits using Akaike information criterion (AIC).
# The `gamlr` package  provides useful functions for calculating AIC
AIC_gaus_mods_v1 <- data.frame(model = c("mod_gaus_1.1", "mod_gaus_1.2", "mod_gaus_1.3"),
                          AIC = c(AIC(mod_gaus_1.1), AIC(mod_gaus_1.2), AIC(mod_gaus_1.3)))

AIC_gaus_mods_v1[order(AIC_gaus_mods_v1$AIC),]
```

```
##          model         AIC
## 2 mod_gaus_1.2   -8.901126
## 3 mod_gaus_1.3  317.592610
## 1 mod_gaus_1.1 7024.808250
```

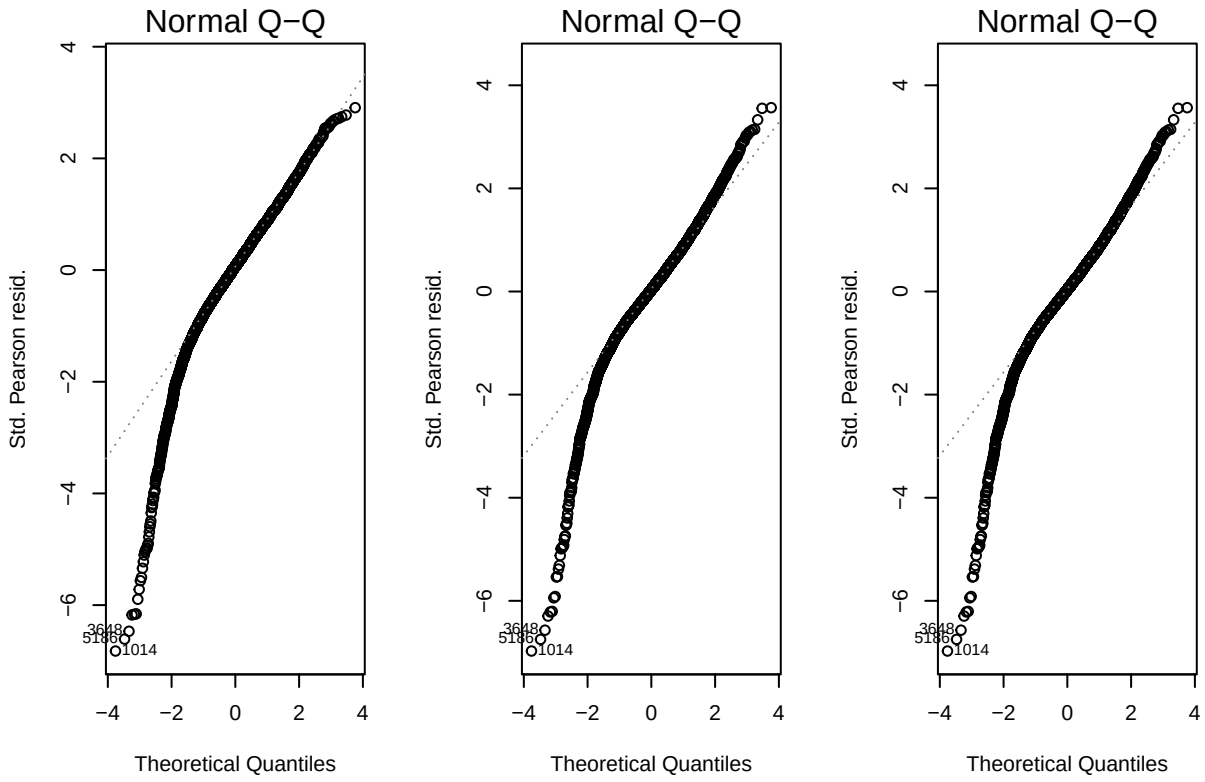
```
# Checking the diagnostic plots we can see that the log and inverse transformations don't
# improve the model fit or normality
par(mfrow = c(1, 3))

plot(mod_gaus_1.1, which=2)
plot(mod_gaus_1.2, which=2)
plot(mod_gaus_1.3, which=2)
```



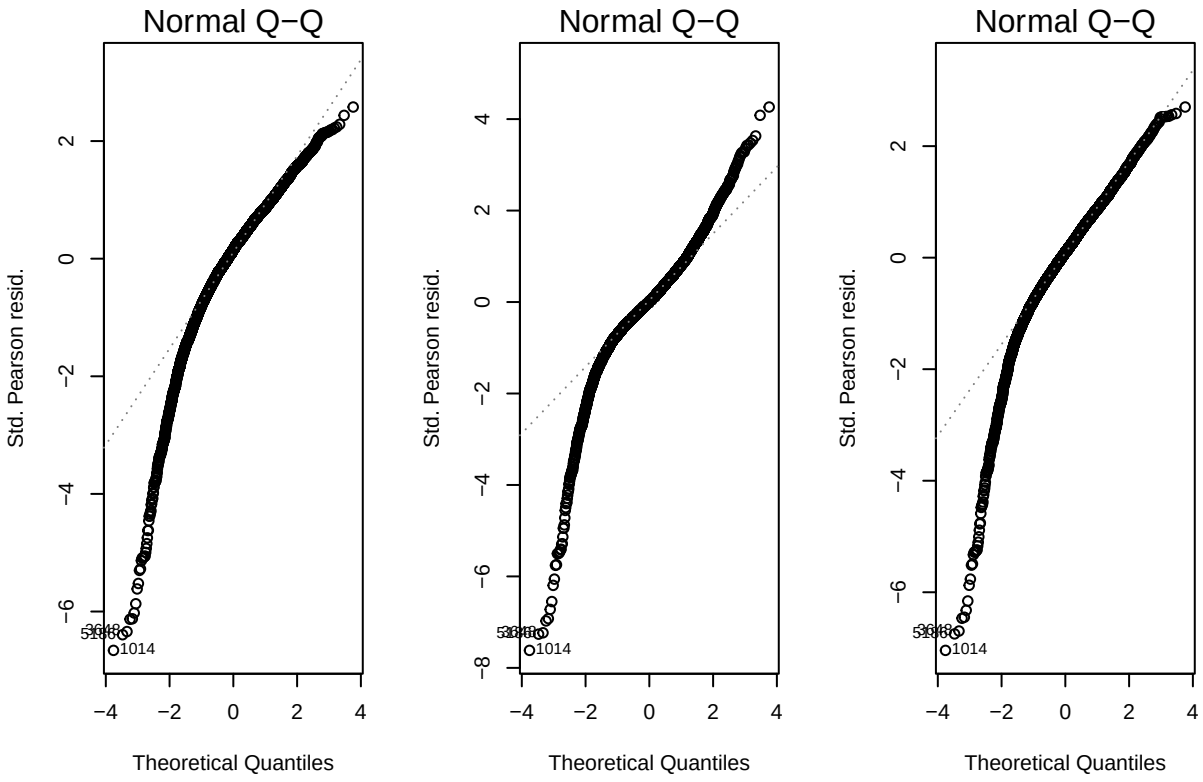

```
# Testing gamma distributions due to skewness
# Transforming to make variables non-negative or 0
mod_gam_1.1 <- glm(sqrt(weight) ~ log(snao + 1) + sqrt(std_time) + scale(age) + sex +
                     scale(pop_size) + site + n_babies,
                     data = data_combined,
                     family = Gamma(link = "log"))

mod_gam_1.2 <- update(mod_gam_1.1, family=Gamma(link="identity"))

mod_gam_1.3 <- update(mod_gam_1.1, family=Gamma(link="inverse"))

# Testing inverse Gaussian distribution due to skewness
# Transforming to make variables non-negative or 0
mod_ingam_1.1 <- glm(sqrt(weight) ~ log(snao + 1) + sqrt(std_time) + scale(age) + sex +
                       scale(pop_size) + site + n_babies,
                       data = data_combined,
                       family = inverse.gaussian())

mod_ingam_1.2 <- update(mod_ingam_1.1, family=inverse.gaussian(link="identity"))

mod_ingam_1.3 <- update(mod_ingam_1.1, family=inverse.gaussian(link="inverse"))
```

```
par(mfrow = c(1, 3))

# Comparing Gamma Distribution
plot(mod_gam_1.1, which=2)
```

```
plot(mod_gam_1.2, which=2)
plot(mod_gam_1.2, which=2)
```




```
# Comparing inverse Gaussian Distribution
plot(mod_ingam_1.1, which=2)
plot(mod_ingam_1.2, which=2)
plot(mod_ingam_1.3, which=2)
```



From these results, the best fit to the data was the Gaussian distribution with an identity link.

I looked at testing potential interactions and random effects.

```
# Looking at an interaction between sNAO and std_time
mod_gaus_1.4 <- glm(scale(weight) ~ log(snao + 1)*scale(std_time) + scale(age) + sex +
                      scale(pop_size) + site + n_babies,
                      data = data_combined,
                      family = "gaussian")

# To assess if ID should be nested within site, each ID should be unique to one site
nrow(data_combined %>%
  ungroup() %>%
  dplyr::select(ID, site) %>%
  unique())
```

```
## [1] 873
```

```
# Number of rows is almost exactly the same as the largest ID number, meaning that
# it should be nested within site.
# Using glmmTMB to model generalized linear mixed-effects model
mod_gaus_1.5 <- glmmTMB(as.vector(scale(weight)) ~ log(snao + 1)*scale(std_time) +
                          scale(age) + sex + scale(pop_size) + (1|site/ID),
                          data = data_combined,
                          family = "gaussian")
```

```
AIC_gaus_mods_v2 <- data.frame(model = c("mod_gaus_1.1", "mod_gaus_1.4", "mod_gaus_1.5"),
                          AIC = c(AIC(mod_gaus_1.1), AIC(mod_gaus_1.4), AIC(mod_gaus_1.5)))

AIC_gaus_mods_v2[order(AIC_gaus_mods_v2$AIC),]
```

```
##          model      AIC
## 3 mod_gaus_1.5 4240.829
## 2 mod_gaus_1.4 7022.718
## 1 mod_gaus_1.1 7024.808
```

mod_gaus_1.5 appears to be the best fit, however I noticed that the relationship between sNAO and weight did not follow the trend visualized in Figure 1. The results of the model suggest that sNAO declines as weight increases.

```
summary(mod_gaus_1.5)
```

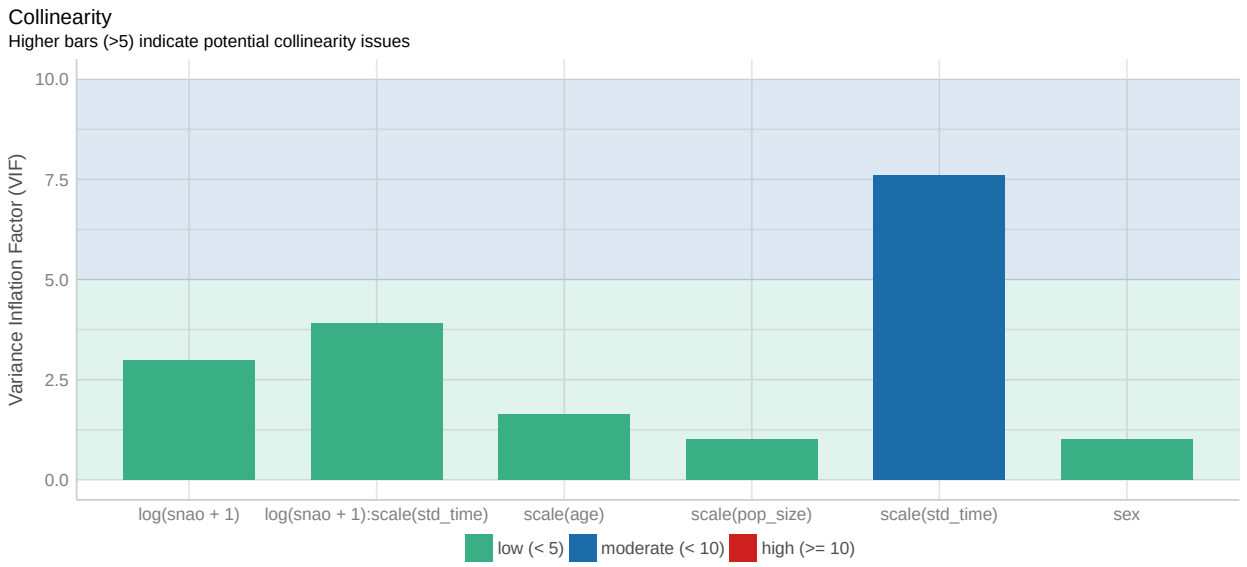
```
##  Family: gaussian  ( identity )
## Formula:          as.vector(scale(weight)) ~ log(snao + 1) * scale(std_time) +
##     scale(age) + sex + scale(pop_size) + (1 | site/ID)
## Data: data_combined
##
##      AIC      BIC   logLik deviance df.resid
##   4240.8   4307.6  -2110.4   4220.8     5847
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance  Std.Dev.
##  ID:site  (Intercept) 1.016e-01 0.31883
##  site     (Intercept) 3.434e-05 0.00586
##  Residual             8.985e-02 0.29974
## Number of obs: 5857, groups:  ID:site, 873; site, 3
##
## Dispersion estimate for gaussian family (sigma^2): 0.0898
##
## Conditional model:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    0.478705   0.037857   12.65   <2e-16 ***
## log(snao + 1)                 -0.096933   0.075181   -1.29    0.197
## scale(std_time)                0.027899   0.023965    1.16    0.244
## scale(age)                     0.772913   0.005854  132.02   <2e-16 ***
## sexm                          -0.877527   0.023793  -36.88   <2e-16 ***
## scale(pop_size)               -0.075442   0.004230  -17.83   <2e-16 ***
## log(snao + 1):scale(std_time)  0.004366   0.036962    0.12    0.906
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As both std_time and snao exhibit a similar relationship with weight (Figure 1), these two variables may be linearly related, resulting in collinearity which may explain the poor significance and reversed trend.

```r
# CHecking for collinearity using the 'performance' package to calculate VIF
collinearity_results <- check_collinearity(mod_gaus_1.5)

collinearity_results
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##                           Term  VIF Increased SE Tolerance
##                  log(snao + 1) 2.99         1.73      0.33
##                     scale(age) 1.62         1.27      0.62
##                            sex 1.00         1.00      1.00
##                 scale(pop_size) 1.02         1.01      0.98
##  log(snao + 1):scale(std_time) 3.91         1.98      0.26
##
## Moderate Correlation
##
##              Term  VIF Increased SE Tolerance
##  scale(std_time) 7.59         2.75      0.13
```

```r
plot(collinearity_results)
```



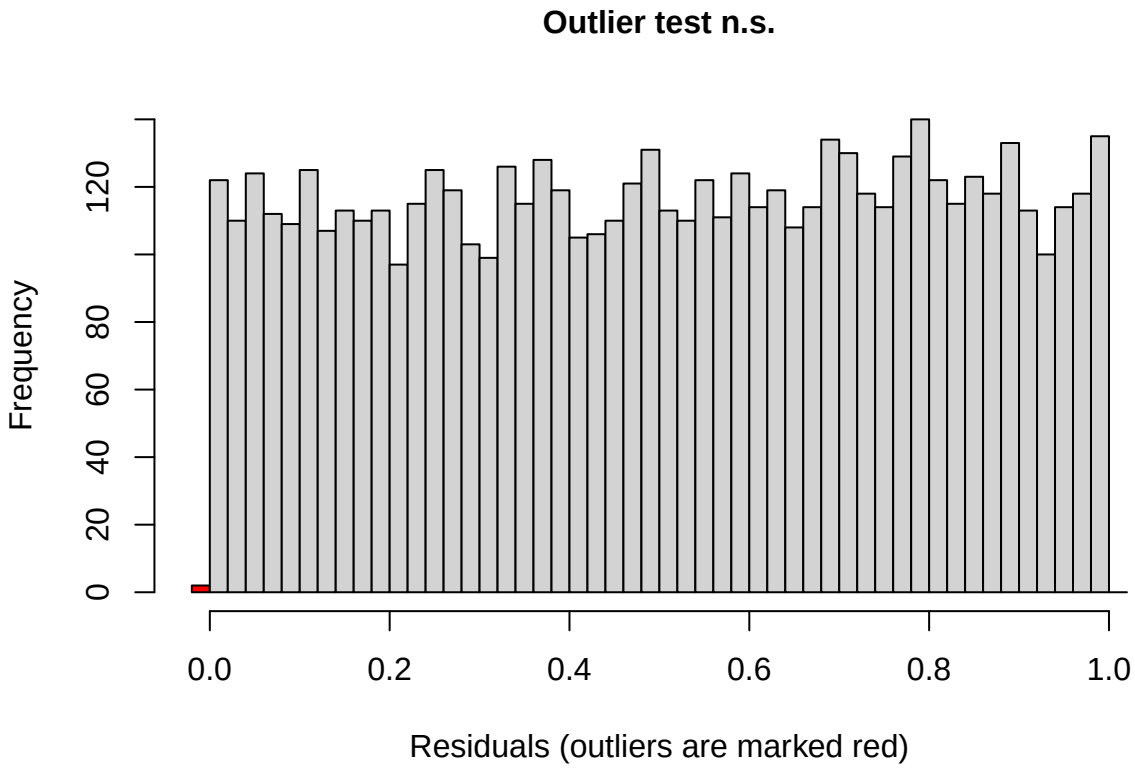
The results show that there is moderate correlation seen for `std_time`. To remedy this, I removed `std_time` from the model.

```r
# Removing std_time from the model
# Removing n_babes variable due to lack of significance or effect
mod_gaus_1.6 <- glmmTMB(as.vector(scale(weight)) ~ log(snao + 1) + scale(age) + sex +
                          scale(pop_size) + (1|site/ID),
                          data = data_combined,
                          family = "gaussian")

summary(mod_gaus_1.6)
```

```
##  Family: gaussian  ( identity )
## Formula:
## as.vector(scale(weight)) ~ log(snao + 1) + scale(age) + sex +
##     scale(pop_size) + (1 | site/ID)
## Data: data_combined
##
##      AIC      BIC   logLik deviance df.resid
##   4240.2   4293.6  -2112.1   4224.2     5849
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance  Std.Dev.
##  ID:site  (Intercept) 1.019e-01 0.319188
##  site     (Intercept) 2.722e-05 0.005218
##  Residual             8.988e-02 0.299797
## Number of obs: 5857, groups:  ID:site, 873; site, 3
##
## Dispersion estimate for gaussian family (sigma^2): 0.0899
##
## Conditional model:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.4406176  0.0283345   15.55   <2e-16 ***
## log(snao + 1)    0.0009812  0.0501892    0.02    0.984
## scale(age)       0.7774089  0.0052902  146.95   <2e-16 ***
## sexm            -0.8775676  0.0238130  -36.85   <2e-16 ***
## scale(pop_size) -0.0757503  0.0041955  -18.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

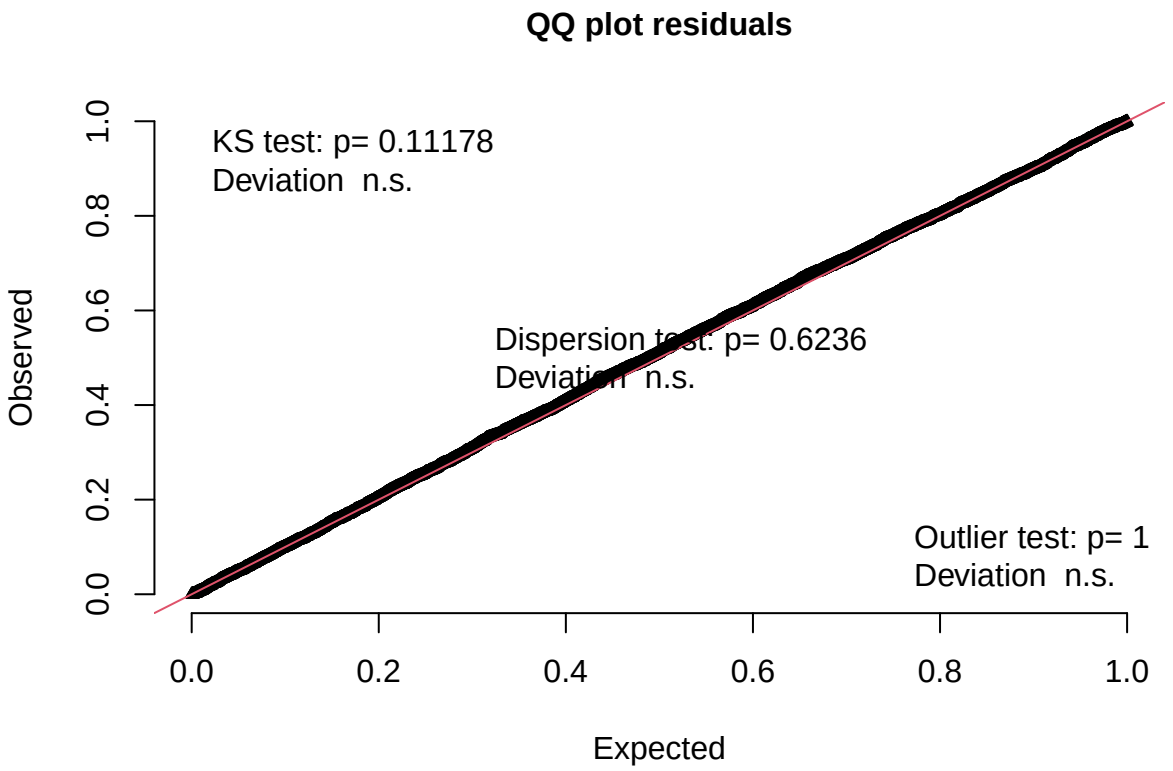
I assessed the fit of the new model and tested for deviation, heteroskedasticity, and outliers, using the `DHARMa` package.

```
set.seed(666)

# Increasing the number of simulations to 5000 to stabilize the simulated values and
# decrease the number of random outliers.
# Error message pops up as we're using a GLM instead of a GLMM as input, but this
# has little impact on the deviation statistics or outliers.
mod_gaus_1.6_sim <- simulateResiduals(mod_gaus_1.6, n = 5000)

# Testing for outliers
testOutliers(mod_gaus_1.6_sim, plot = TRUE)
```



Although there appear to be some outliers, the frequency is very low and insignificant. These may just be random and arise from too-low a number of simulations.

```
# Looking at the residuals to see if they follow the normal distribution and to see if
# there's any significant dispersion
plotQQunif(mod_gaus_1.6_sim)
```



There is no significant deviation detected, suggesting the fit to the Normal distribution is appropriate.

```
# Calculating a pseudo r2 to get a very rough idea of the goodness-of-fit
# Using Nakagawa's pseudo r2 from'performance' package as we are calculating r2 for a GLMM
r2_nakagawa(mod_gaus_1.6)
```

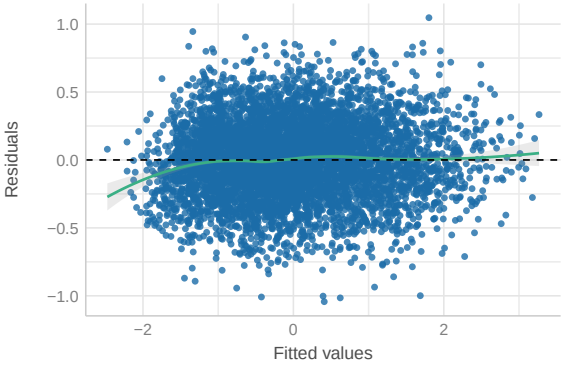
```
## # R2 for Mixed Models
##
##   Conditional R2: 0.909
##      Marginal R2: 0.806
```

0.8 and 0.9 are conditional and marginal values respectively.

```
check_model(mod_gaus_1.6)
```

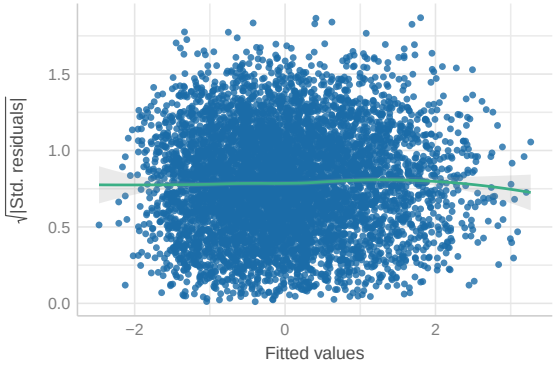
Linearity

Reference line should be flat and horizontal



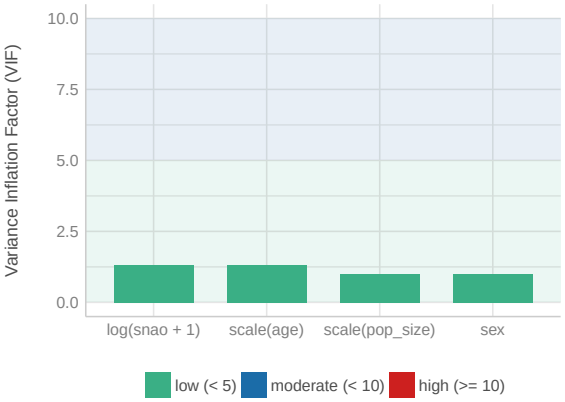
Homogeneity of Variance

Reference line should be flat and horizontal



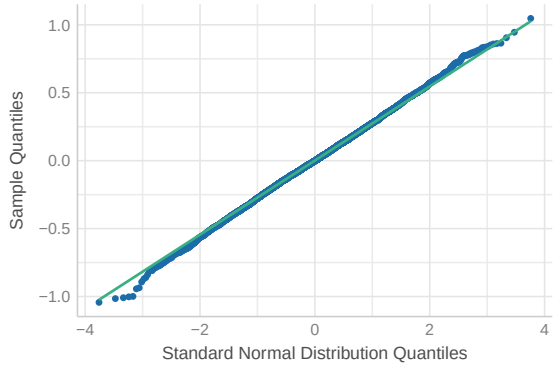
Collinearity

Higher bars (>5) indicate potential collinearity issues



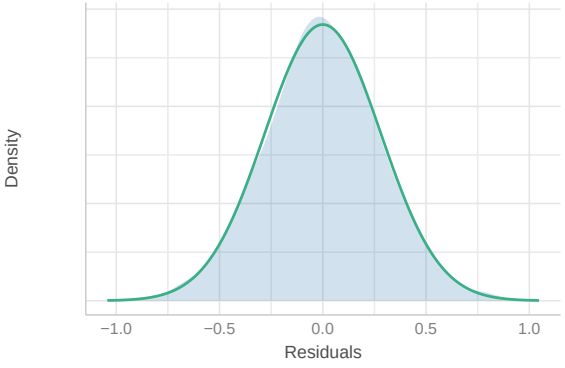
Normality of Residuals

Dots should fall along the line



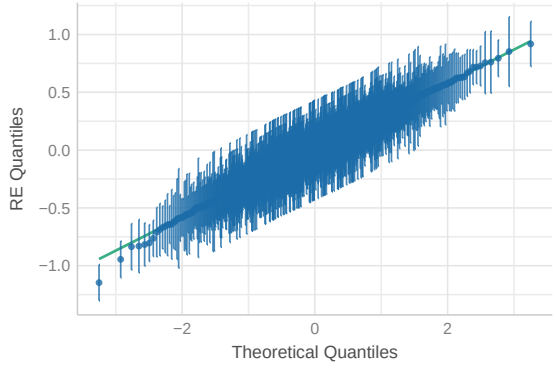
Normality of Residuals

Distribution should be close to the normal curve



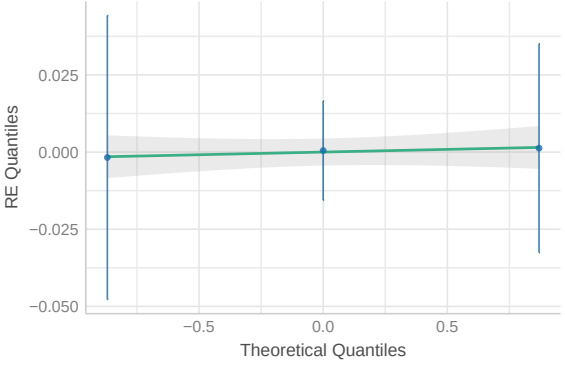
Normality of Random Effects (ID:site)

Dots should be plotted along the line



Normality of Random Effects (site)

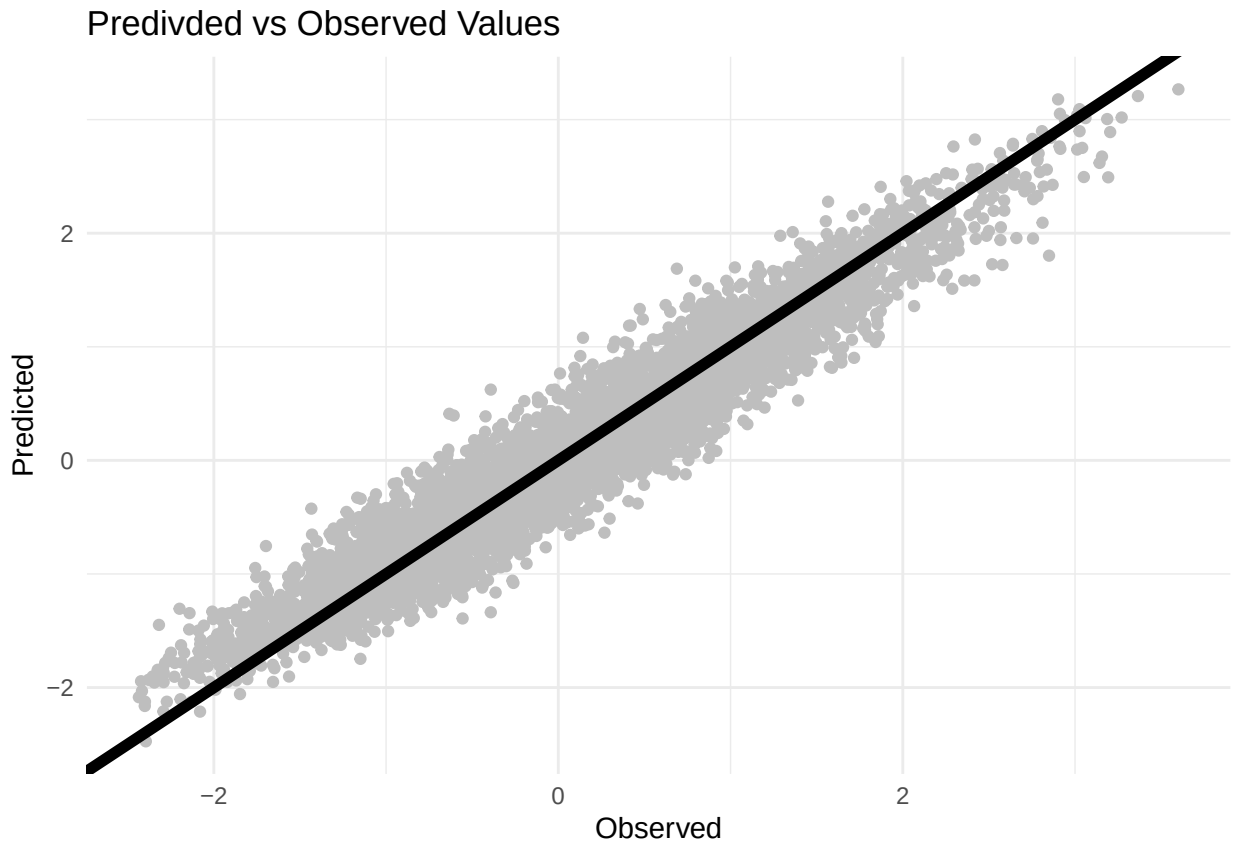
Dots should be plotted along the line



The data appears to be homoscedastic with no significant VIF between predictors.

For some final visual checks, I plotted the predicted values to the observed values.

```
# Generating predictions from the model
data_combined$predicted <- predict(mod_gaus_1.6,
                                   data = data_combined,
                                   predict.all=TRUE,
                                   type = "response")

#Plotting the predicted results against the observed data
mod_plot <- ggplot(data_combined, aes(x = scale(weight),
                              y = predicted)) +
  geom_point(col="grey") +
  geom_abline(slope = 1, size=2) +
  theme_minimal() +
  labs(
    x = "Observed",
    y = "Predicted",
    title = "Predivded vs Observed Values"
  )

mod_plot
```



Results

The following assumptions were made based on the summary below:

Predictor	Relationship with Soay Sheep mass	Evidence
Age	An increase in age is associated with an increase in sheep mass	p value ~ 0, Visual trend
Sex	Male sheep are on average associated with a decrease in mass compared to females	p value ~ 0, Visual trend
Population Size	Population increase is associated with a decrease in sheep mass	p value ~ 0, Visual trend

The variables with the strongest influence on Soay sheep mass are age and sex as evidenced by their relatively large coefficients of 0.774390 and -0.886717 respectively. Although there is a clear correlation between `std_time` and `snao` with `weight` individually, it appears that together with other predictors their effects become 'insignificant'.

```
summary(mod_gaus_1.6)
```

```
##  Family: gaussian  ( identity )
## Formula:
## as.vector(scale(weight)) ~ log(snao + 1) + scale(age) + sex +
##     scale(pop_size) + (1 | site/ID)
## Data: data_combined
##
##      AIC      BIC   logLik deviance df.resid
##   4240.2   4293.6  -2112.1   4224.2     5849
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance  Std.Dev.
##  ID:site  (Intercept) 1.019e-01 0.319188
##  site     (Intercept) 2.722e-05 0.005218
##  Residual             8.988e-02 0.299797
## Number of obs: 5857, groups:  ID:site, 873; site, 3
##
## Dispersion estimate for gaussian family (sigma^2): 0.0899
##
## Conditional model:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.4406176  0.0283345   15.55   <2e-16 ***
## log(snao + 1)    0.0009812  0.0501892    0.02    0.984
## scale(age)       0.7774089  0.0052902  146.95   <2e-16 ***
## sexm            -0.8775676  0.0238130  -36.85   <2e-16 ***
## scale(pop_size) -0.0757503  0.0041955  -18.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Data Sources

NAO and Soay sheep mass data was provided by Dr. Chris Clements on github.