Homework 5

Luke Todd

2023-06-05

Loading Packages

```
# should haves (from last week)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(here)
```

```
## here() starts at /Users/luketodd/Desktop/Rprojects/ES 193DS/ES 193DS Homework 5
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(ggeffects)
library(performance)
library(naniar) # or equivalent
library(flextable) # or equivalent
```

```
##
## Attaching package: 'flextable'
##
## The following object is masked from 'package:purrr':
##
##     compose
```

```
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
```

```
library(broom)
# would be nice to have
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(AICcmodavg)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

Loading Data

```
sar <- read.csv(here("Data/hf109-01-sarracenia.csv")) %>%
  clean_names() %>%
  select(totmass, species, feedlevel, sla, chlorophyll, amass, num_lvs, num_phylls)
```

7a

7b

7c

```
gg_miss_var(sar)
```

```
# creating a dataset without the missing values
sar_nona <- sar %>%
  drop_na(sla, chlorophyll, amass, num_lvs, num_phylls)
```

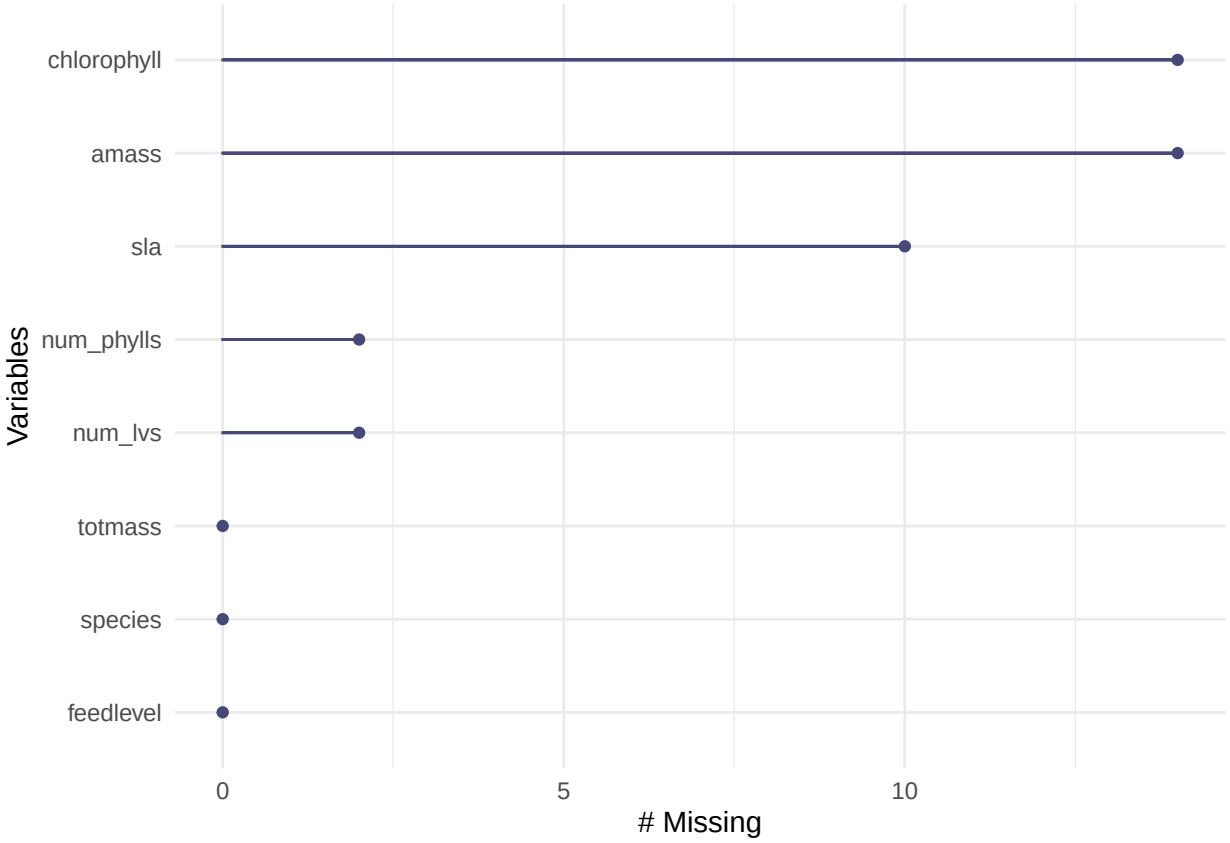


Figure 1: This figure displays the number of missing values for each variable in the sar data frame. Based on this figure, we can see that chlorophyll (Chlorophyll content), amass (Photosynthetic Rate), and sla (Specific Leaf Area) are missing the most values with 14, 14, and 10 missing values, respectively. Num_phylls and num_lvs are missing 2 values, and the rest are missing none.

7d

```
# calc Pearson's r for numerical values
sar_cor <- sar_nona %>%
  select(feedlevel:num_phylls) %>%
  cor(method = "pearson")

# plot correlation values
corrplot(sar_cor,
         method = "ellipse",
         addCoef.col = "black")
```

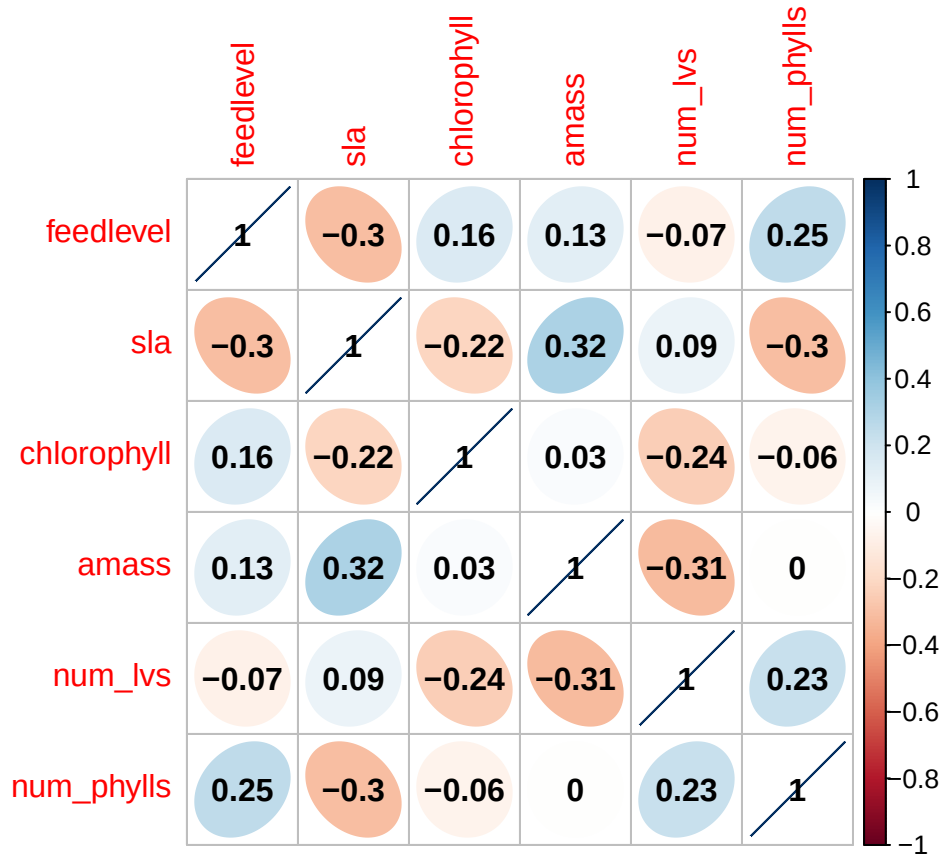
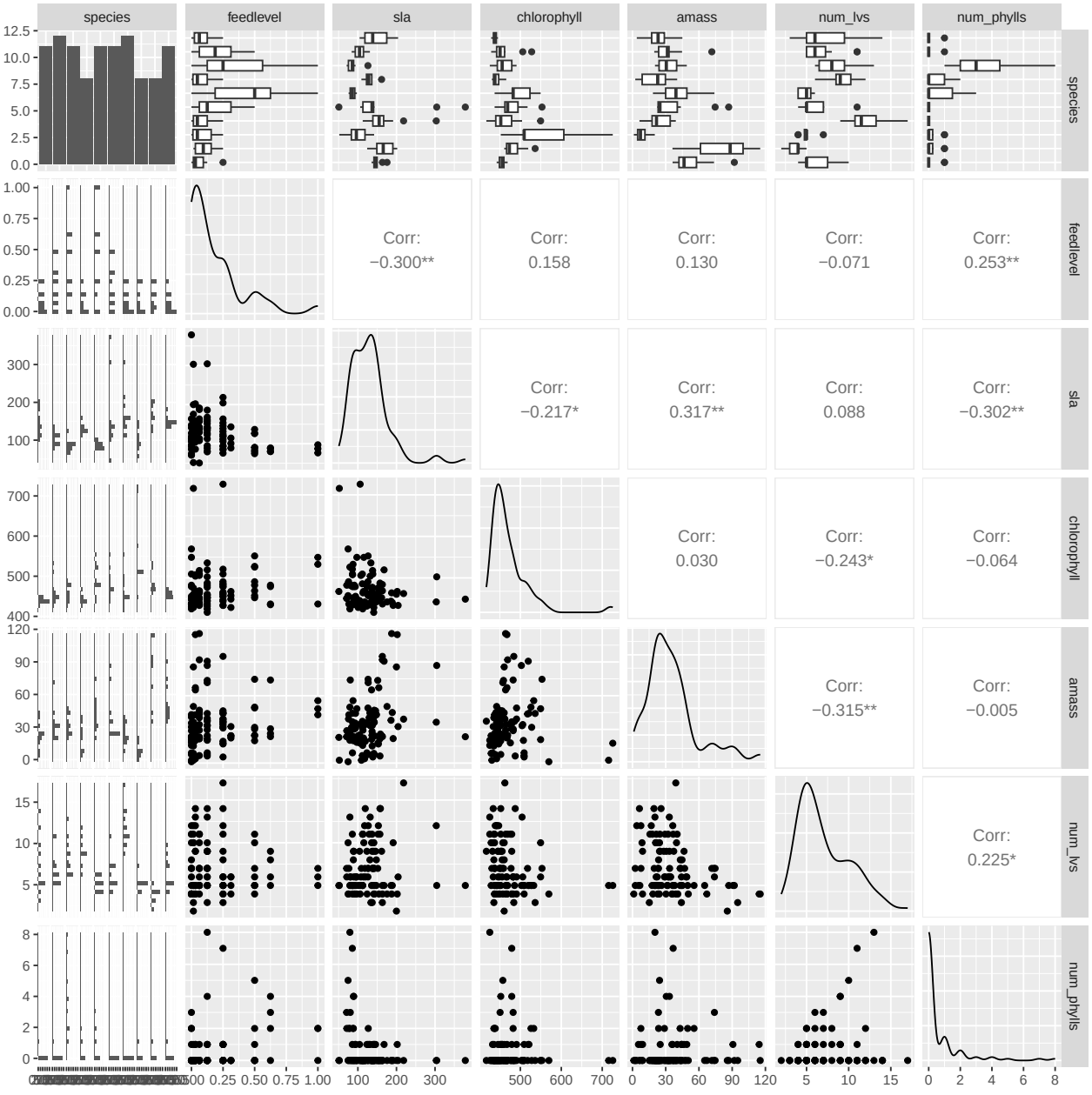


Figure 2: This figure displays the Person's correlation values between different variables. High absolute values means that there is a greater correlation between the variables. For example, sla and amass have the greatest positive correlation between each other with a value of 0.32. On the otherhand, num_lvs and amass have the greatest negative correlation between each other with a value of -0.31.

7e

```
sar_nona %>%
  select(species:num_phylls) %>%
  ggpairs()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



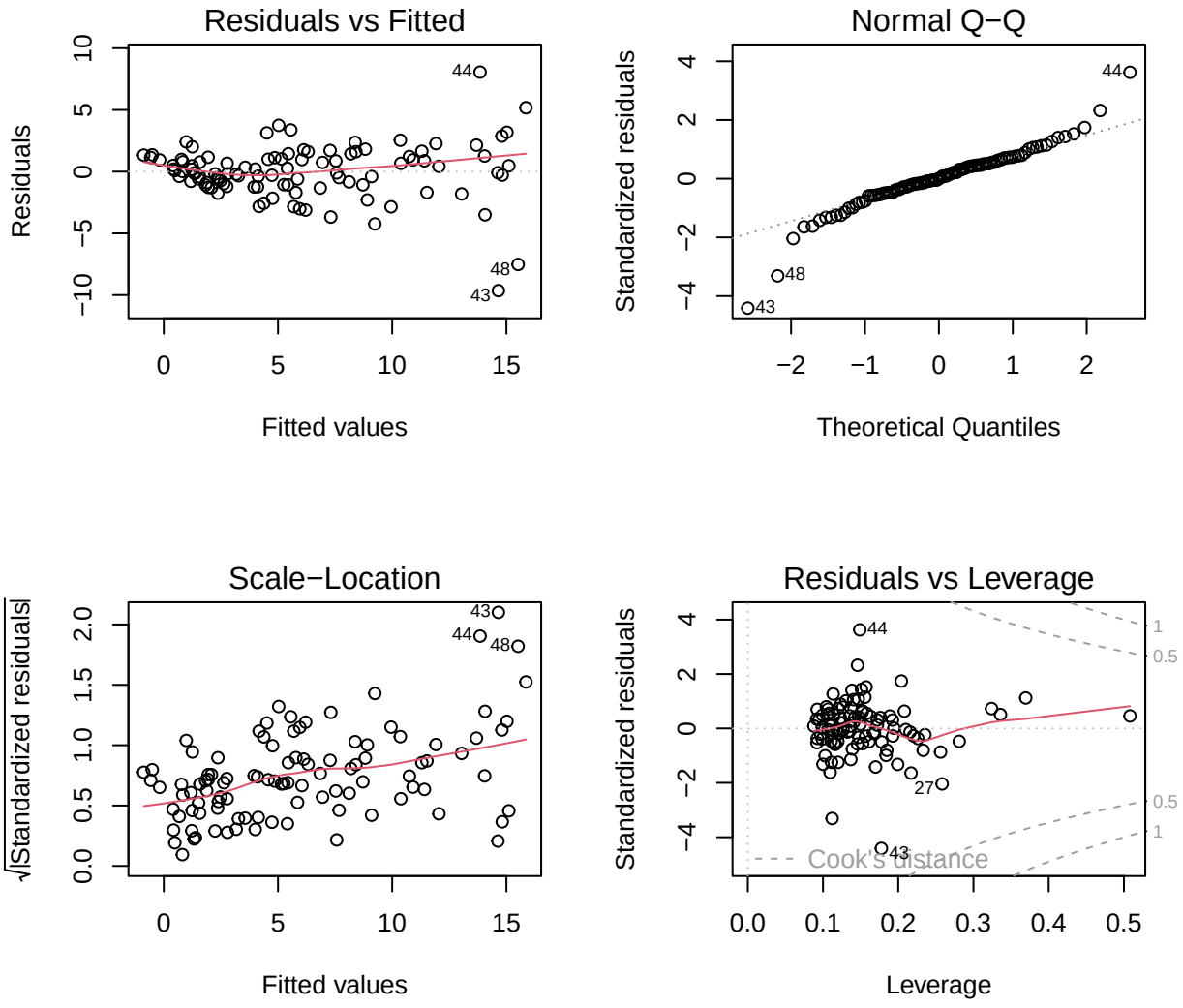
7f

As we are trying to predict totmass, the y-value is set to totmass. For the null model, we set the x-value to 1, as this selects for just the intercept. For the full model, we select every variable that we are interested in.

```
null <- lm(totmass ~ 1, data = sar_nona)
full <- lm(totmass ~ species + feedlevel +
             sla + chlorophyll + amass +
             num_lvs + num_phylls,
           data = sar_nona)
```

7g/h

```
# visual diagnostics for full model
par(mfrow = c(2, 2))
plot(full)
```




```
# statistical diagnostic checks
check_normality(full)
```

```
## Warning: Non-normality of residuals detected (p < .001).
```

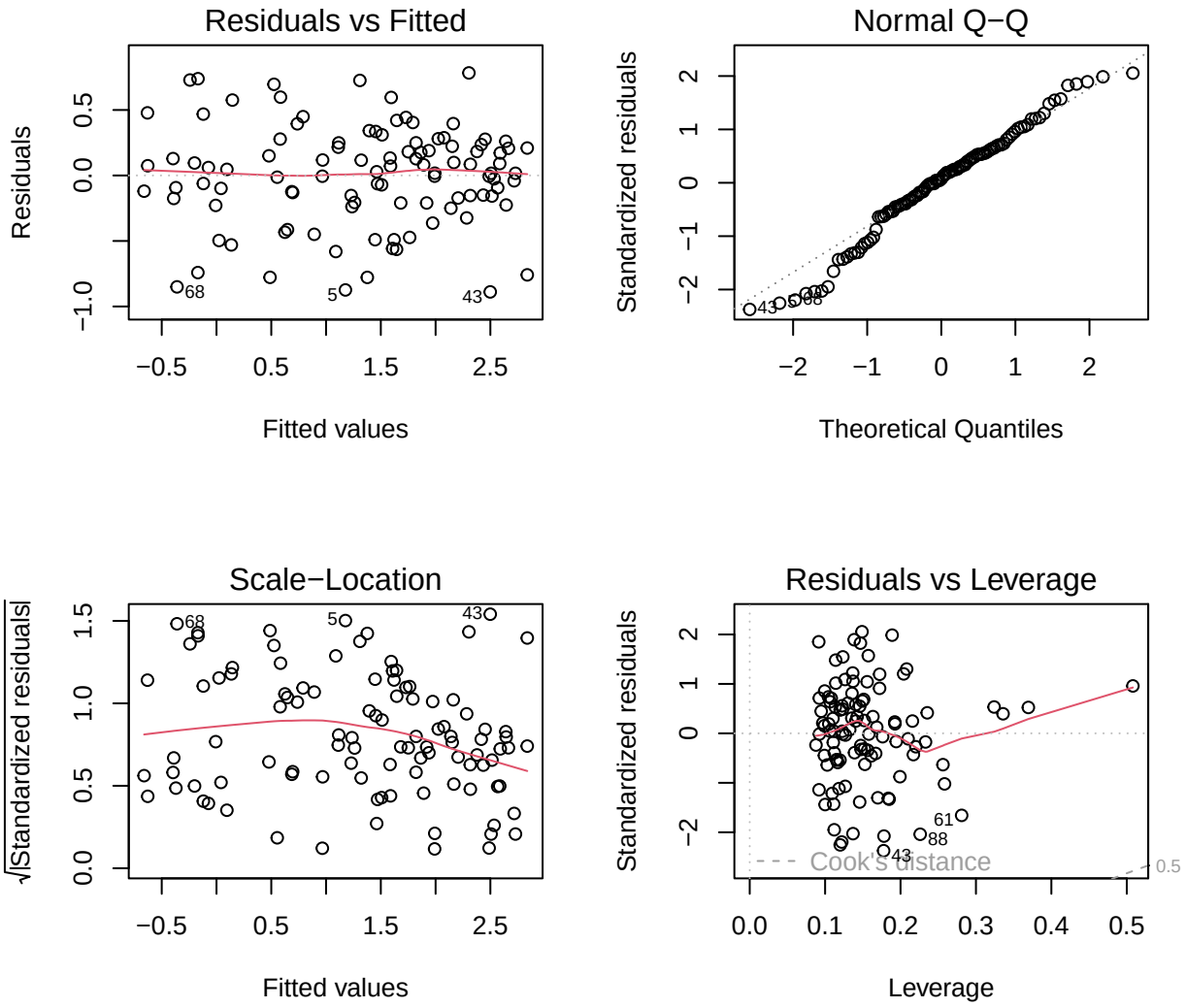
```
check_heteroscedasticity(full)
```

```
## Warning: Heteroscedasticity (non-constant error variance) detected (p < .001).
```

Checking normality and homoscedasticity assumptions both gave p-values less than 0.001, indicating that our current full model is non-normal and heteroscedastic. Because of this, we will log-transform our model and retest the diagnostics.

```
# creating log transformed models
null_log <- lm(log(totmass) ~ 1, data = sar_nona)
full_log <- lm(log(totmass) ~ species + feedlevel +
                 sla + chlorophyll + amass + num_lvs +
                 num_phylls,
               data = sar_nona)
```

```
# visual diagnostic checks
par(mfrow = c(2, 2))
plot(full_log)
```



```r
# statistical diagnostic checks
check_normality(full_log)
```

```
## OK: residuals appear as normally distributed (p = 0.107).
```

```r
check_heteroscedasticity(full_log)
```

```
## OK: Error variance appears to be homoscedastic (p = 0.071).
```

The statistical diagnostic checks for normality and homoscedasticity passed, so we will continue using the log-transformed model.

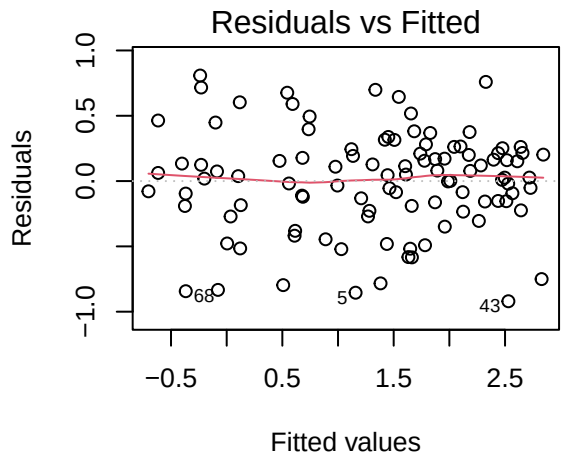
7i

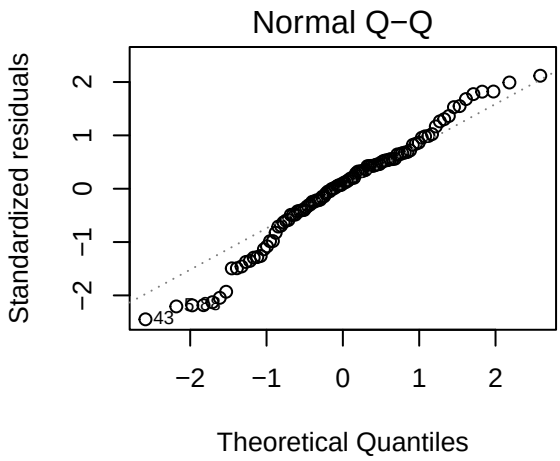
```r
# using ANOVA tables to create new models, eliminating one non-significant variable at a time
anova(full_log)
```

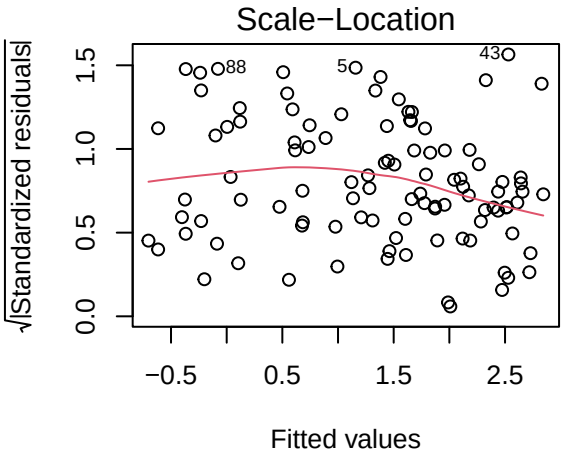
```
## Analysis of Variance Table
##
## Response: log(totmass)
##             Df Sum Sq Mean Sq F value    Pr(>F)
## species      9 91.175 10.1305 59.3944 < 2.2e-16 ***
## feedlevel    1  0.110  0.1099  0.6442 0.4243716
## sla          1  1.300  1.3003  7.6236 0.0070259 **
## chlorophyll  1  2.564  2.5636 15.0301 0.0002049 ***
## amass        1  0.033  0.0330  0.1937 0.6609205
## num_lvs      1  2.921  2.9205 17.1227 8.068e-05 ***
## num_phylls   1  0.100  0.0999  0.5859 0.4460675
## Residuals   87 14.839  0.1706
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

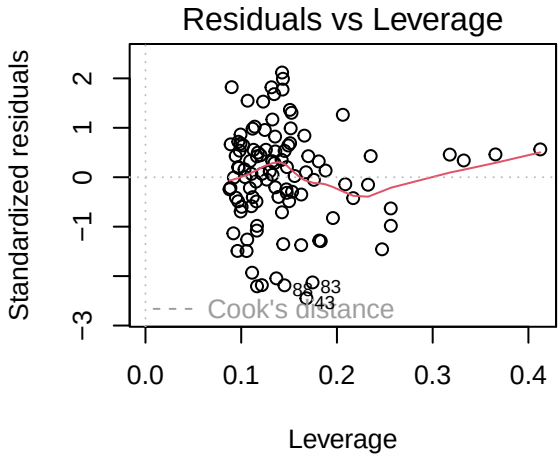
```r
# based on this ANOVA, I will create a model without "amass" since it had the highest p-value
model2 <- lm(log(totmass) ~ species + feedlevel + sla + chlorophyll + num_lvs + num_phylls,
             data = sar_nona)

# checking model2 visual diagnostics
par(mfrow = c(2, 2))
plot(model2)
```










```
# checking model statistical diagnostics
check_normality(model2)
```

```
## OK: residuals appear as normally distributed (p = 0.075).
```

```
check_heteroscedasticity(model2)
```

```
## OK: Error variance appears to be homoscedastic (p = 0.054).
```

Based on the visual and statistical diagnostic test, model2 passes all tests. Next, I will eliminate another variable and see if it still passes these tests.

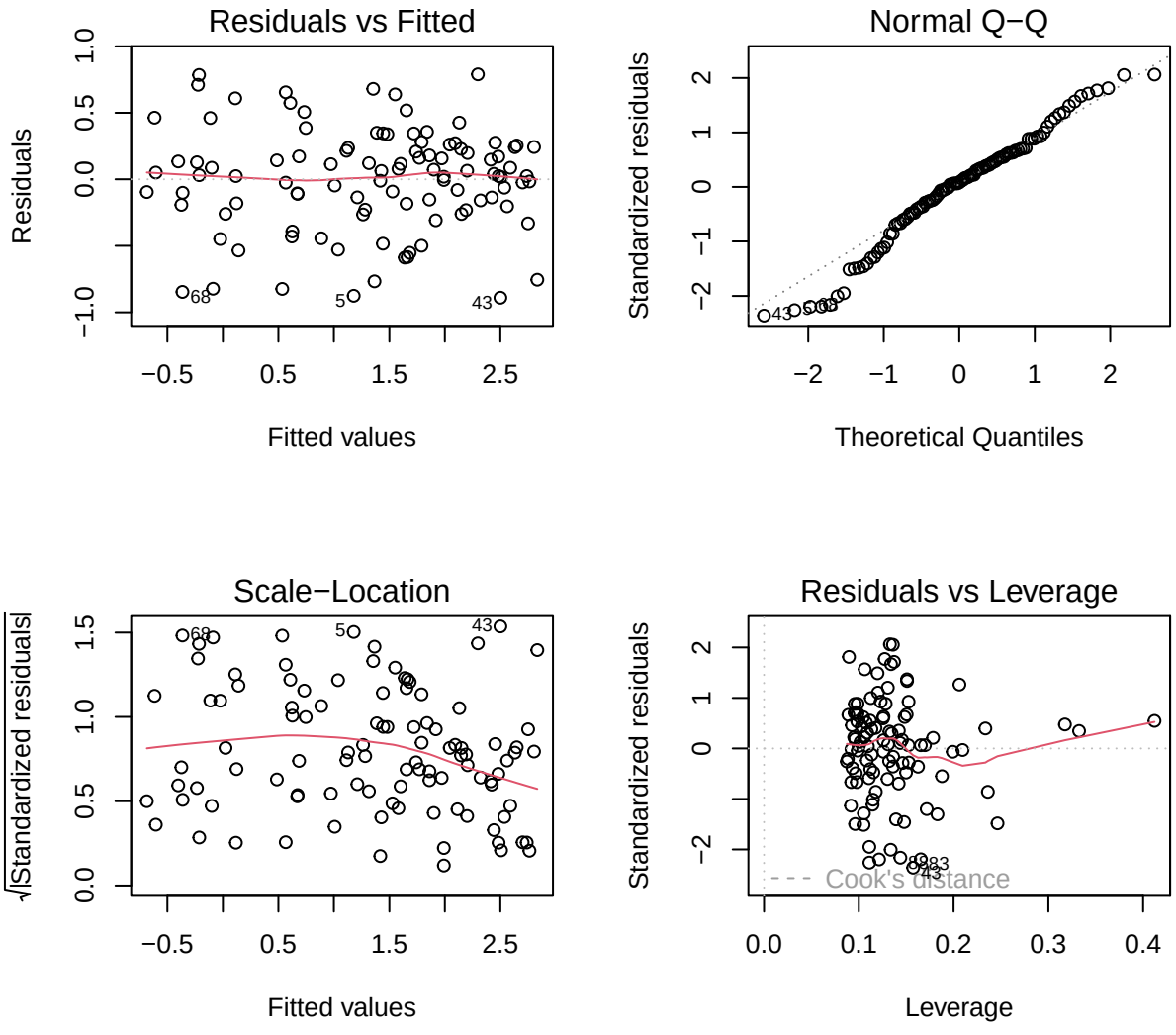
```
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: log(totmass)
##             Df Sum Sq Mean Sq F value    Pr(>F)
## species      9 91.175 10.1305 59.6575 < 2.2e-16 ***
```

```
## feedlevel    1  0.110  0.1099  0.6471 0.4233265
## sla          1  1.300  1.3003  7.6574 0.0068914 **
## chlorophyll  1  2.564  2.5636 15.0966 0.0001975 ***
## num_lvs      1  2.866  2.8665 16.8803 8.903e-05 ***
## num_phylls   1  0.083  0.0827  0.4868 0.4872174
## Residuals   88 14.943  0.1698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# based on the above ANOVA, I will create a model without "num_phylls" since it had the highest p-value
model3 <- lm(log(totmass) ~ species + feedlevel + sla + chlorophyll + num_lvs,
             data = sar_nona)

# checking model2 visual diagnostics
par(mfrow = c(2, 2))
plot(model3)
```




```
# checking model statistical diagnostics
check_normality(model3)
```

```
## OK: residuals appear as normally distributed (p = 0.062).
```

```
check_heteroscedasticity(model3)
```

```
## OK: Error variance appears to be homoscedastic (p = 0.067).
```

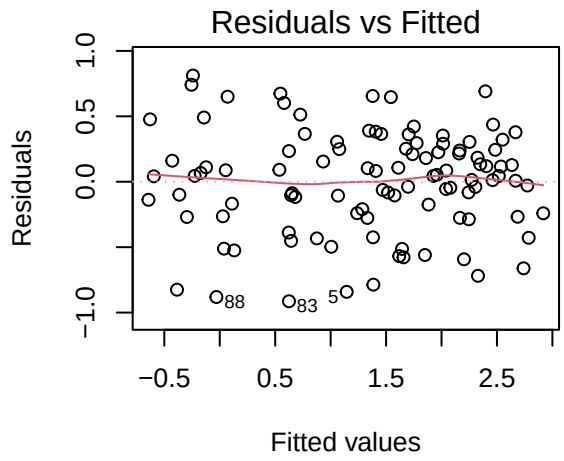
Based on the visual and statistical diagnostic test, model3 passes all tests. Next, I will eliminate another variable and see if it still passes these tests.

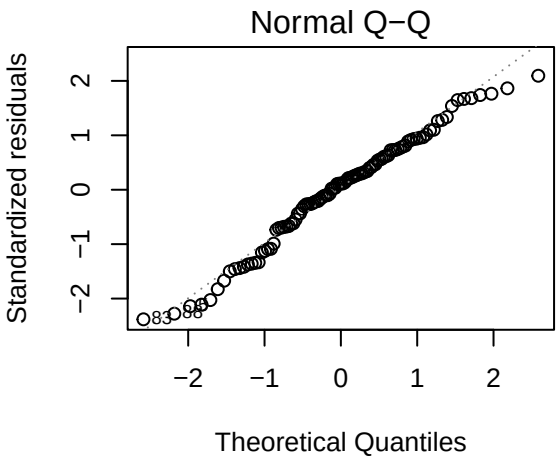
```
anova(model3)
```

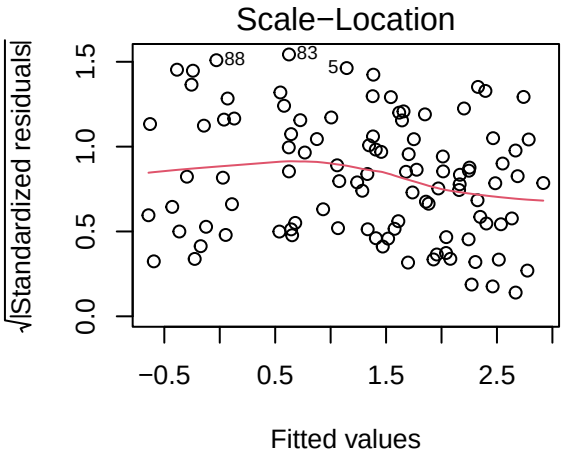
```
## Analysis of Variance Table
##
## Response: log(totmass)
##             Df Sum Sq Mean Sq F value    Pr(>F)
## species      9 91.175 10.1305 60.0035 < 2.2e-16 ***
## feedlevel    1  0.110  0.1099  0.6508 0.4219656
## sla          1  1.300  1.3003  7.7018 0.0067230 **
## chlorophyll  1  2.564  2.5636 15.1842 0.0001885 ***
## num_lvs      1  2.866  2.8665 16.9782 8.457e-05 ***
## Residuals   89 15.026  0.1688
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

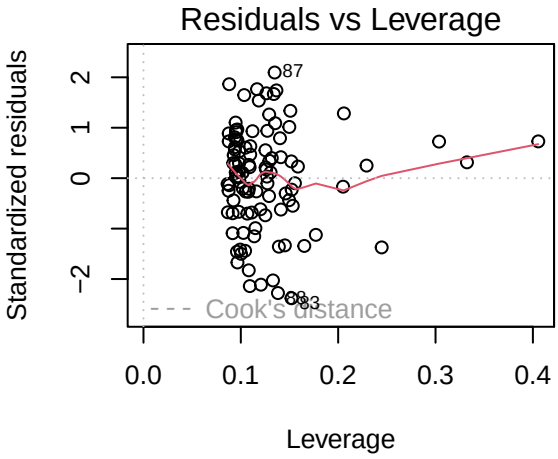
```
# based on the above ANOVA, I will create a model without "feedlevel" since it had the highest p-value
model4 <- lm(log(totmass) ~ species + sla + chlorophyll + num_lvs,
             data = sar_nona)

# checking model2 visual diagnostics
par(mfrow = c(2, 2))
plot(model4)
```










```
# checking model statistical diagnostics
check_normality(model4)
```

```
## OK: residuals appear as normally distributed (p = 0.169).
```

```
check_heteroscedasticity(model4)
```

```
## Warning: Heteroscedasticity (non-constant error variance) detected (p = 0.036).
```

The statistical tests show that model4 appears to have heteroscedasticity, or non-constant error variance, with a p-value of 0.036.

The type of model selection that I used above is called backward model selection. It begins at a full model and you slowly eliminate the least significant variable, until you are left with a model that passes all diagnostics, but has the least amount of predictor variables. Most of the time, it is better to have the least amount of predictor variables as possible since it increases the interpretability of your model. Based on the above method, "model3" appears to be our best model.

7j

```
# full model variance inflation factor check
car::vif(full_log)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## species     42.351675  9        1.231351
## feedlevel    1.621993  1        1.273575
## sla          1.999989  1        1.414210
## chlorophyll  1.949828  1        1.396362
## amass        2.872084  1        1.694722
## num_lvs      2.813855  1        1.677455
## num_phylls   2.995510  1        1.730754
```

Based on the results, we can see that every variable has a GVIF value greater than 1, indicating that there is some multicollinearity. Ideally, you want to be as close as possible to 1.

7k

```
# comparing models using AIC
MuMIn::AICc(full, full_log, null, model2, model3, model4)
```

```
##          df     AICc
## full     17 497.3964
## full_log 17 133.9424
## null      2 630.4028
## model2   16 131.7899
## model3   15 129.5498
## model4   14 130.8121
```

The above chart shows that model3 has the lowest AICc value. Therefore, we will choose model3 as the best model.

8a

The best model was model3, which used the variables species, feedlevel, sla, chlorophyll, and num_lvs to predict log(totmass). I chose this model by using backward model selection, starting from the full model, and slowly eliminating unnecessary, insignificant variables. Model3 was the smallest model that passed the model diagnostics, and it also had the lowest AICc.

Model	Formula	AICc	Diagnostic P/NP
full	totmass ~ species + feedlevel + sla + chlorphyll + amass + num_lvs + num_phylls	497.3964	NP
full_log	log(totmass) ~ species + feedlevel + sla + chlorphyll + amass + num_lvs + num_phylls	133.9424	P
null	totmass ~ 1	630.4028	NA

Model	Formula	AICc	Diagnostic P/NP
model2	log(totmass) ~ species + feedlevel + sla + chlorophyll + num_lvs + num_phylls	131.7899	P
model3	log(totmass) ~ species + feedlevel + sla + chlorophyll + num_lvs	129.5498	P
model4	log(totmass) ~ species + sla + chlorophyll + num_lvs	130.8121	NP