Aim 3 - Body Morphology

Kyle Emerson

2/2/2023

## R Markdown

library(lme4)

## Warning: package 'lme4' was built under R version 4.1.3

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.1.3

library(car)

## Warning: package 'car' was built under R version 4.1.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.1.3

library(readr)

## Warning: package 'readr' was built under R version 4.1.3

library(moments)

## Warning: package 'moments' was built under R version 4.1.3

library(psych)

## Warning: package 'psych' was built under R version 4.1.3

##   
## Attaching package: 'psych'

## The following object is masked from 'package:car':  
##   
## logit

library(pastecs)  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.1.3

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

library(ggbiplot)

## Loading required package: plyr

## Warning: package 'plyr' was built under R version 4.1.3

## Loading required package: scales

## Warning: package 'scales' was built under R version 4.1.3

##   
## Attaching package: 'scales'

## The following objects are masked from 'package:psych':  
##   
## alpha, rescale

## The following object is masked from 'package:readr':  
##   
## col\_factor

## Loading required package: grid

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.7 v dplyr 1.0.9  
## v tidyr 1.2.0 v stringr 1.5.0  
## v purrr 0.3.4 v forcats 0.5.1

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'dplyr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x ggplot2::%+%() masks psych::%+%()  
## x scales::alpha() masks ggplot2::alpha(), psych::alpha()  
## x dplyr::arrange() masks plyr::arrange()  
## x scales::col\_factor() masks readr::col\_factor()  
## x purrr::compact() masks plyr::compact()  
## x dplyr::count() masks plyr::count()  
## x purrr::discard() masks scales::discard()  
## x tidyr::expand() masks Matrix::expand()  
## x tidyr::extract() masks pastecs::extract()  
## x dplyr::failwith() masks plyr::failwith()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::first() masks pastecs::first()  
## x dplyr::id() masks plyr::id()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::last() masks pastecs::last()  
## x dplyr::mutate() masks plyr::mutate()  
## x tidyr::pack() masks Matrix::pack()  
## x dplyr::recode() masks car::recode()  
## x dplyr::rename() masks plyr::rename()  
## x purrr::some() masks car::some()  
## x dplyr::summarise() masks plyr::summarise()  
## x dplyr::summarize() masks plyr::summarize()  
## x tidyr::unpack() masks Matrix::unpack()

library(modelbased)

## Warning: package 'modelbased' was built under R version 4.1.3

library(dplyr)  
library(emmeans)  
  
file.choose()

## [1] "C:\\R\\Aim 3 - Stress and the Microbiota\\Body-Morphology.docx"

df <-read.csv("C:\\Users\\kjeme\\OneDrive\\Desktop\\Woodley Lab\\Aim 3 - CORT & the Microbiota\\Aim 3 (Restarted)\\KE Data FIles\\Aim 3 - Body Morphology (Shaikh)\\Body Morphology Data Sheet.csv")  
df$Water\_Trtmt = factor(df$Water\_Trtmt)  
df$Stressor = factor(df$Stressor)  
df$Replicate = factor(df$Replicate)  
df$Trtmt\_Combo = factor(df$Trtmt\_Combo)  
  
  
shapiro.test(df$Body\_Length)

##   
## Shapiro-Wilk normality test  
##   
## data: df$Body\_Length  
## W = 0.98655, p-value = 0.8665

shapiro.test(df$Body\_Depth)

##   
## Shapiro-Wilk normality test  
##   
## data: df$Body\_Depth  
## W = 0.97854, p-value = 0.5473

shapiro.test(df$Tail\_Muscle\_Height)

##   
## Shapiro-Wilk normality test  
##   
## data: df$Tail\_Muscle\_Height  
## W = 0.96069, p-value = 0.1219

shapiro.test(df$Tail\_Length)

##   
## Shapiro-Wilk normality test  
##   
## data: df$Tail\_Length  
## W = 0.98597, p-value = 0.8463

shapiro.test(df$Tail\_Fin\_Height)

##   
## Shapiro-Wilk normality test  
##   
## data: df$Tail\_Fin\_Height  
## W = 0.98096, p-value = 0.6456

shapiro.test(df$Body\_Width)

##   
## Shapiro-Wilk normality test  
##   
## data: df$Body\_Width  
## W = 0.97602, p-value = 0.4536

shapiro.test(df$Tail\_Fin\_Width)

##   
## Shapiro-Wilk normality test  
##   
## data: df$Tail\_Fin\_Width  
## W = 0.95117, p-value = 0.05204

Close but passes

#Levenes Test  
  
leveneTest(df$Body\_Length, df$Trtmt\_Combo, center = mean, na.rm = TRUE)

## Levene's Test for Homogeneity of Variance (center = mean: TRUE)  
## Df F value Pr(>F)  
## group 5 1.266 0.2975  
## 40

leveneTest(df$Body\_Depth, df$Trtmt\_Combo, center = mean, na.rm = TRUE)

## Levene's Test for Homogeneity of Variance (center = mean: TRUE)  
## Df F value Pr(>F)  
## group 5 1.2493 0.3046  
## 40

leveneTest(df$Tail\_Muscle\_Height, df$Trtmt\_Combo, center = mean, na.rm = TRUE)

## Levene's Test for Homogeneity of Variance (center = mean: TRUE)  
## Df F value Pr(>F)  
## group 5 0.8067 0.5518  
## 40

leveneTest(df$Tail\_Length, df$Trtmt\_Combo, center = mean, na.rm = TRUE)

## Levene's Test for Homogeneity of Variance (center = mean: TRUE)  
## Df F value Pr(>F)  
## group 5 0.3863 0.8552  
## 40

leveneTest(df$Tail\_Fin\_Height, df$Trtmt\_Combo, center = mean, na.rm = TRUE)

## Levene's Test for Homogeneity of Variance (center = mean: TRUE)  
## Df F value Pr(>F)  
## group 5 0.7384 0.5992  
## 40

leveneTest(df$Body\_Width, df$Trtmt\_Combo, center = mean, na.rm = TRUE)

## Levene's Test for Homogeneity of Variance (center = mean: TRUE)  
## Df F value Pr(>F)  
## group 5 0.7379 0.5996  
## 40

leveneTest(df$Tail\_Fin\_Width, df$Trtmt\_Combo, center = mean, na.rm = TRUE)

## Levene's Test for Homogeneity of Variance (center = mean: TRUE)  
## Df F value Pr(>F)  
## group 5 0.3313 0.891  
## 40

All Pass Levene’s Test.

#MANCOVA to test homogeneity of slopes  
  
man1 <- manova(cbind(Body\_Length,Body\_Depth, Tail\_Muscle\_Height, Tail\_Length, Tail\_Fin\_Height, Body\_Width, Tail\_Fin\_Width)~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
summary(man1)

## Df Pillai approx F num Df den Df Pr(>F)   
## Water\_Trtmt 1 0.58088 5.544 7 28 0.0004427 \*\*\*  
## Stressor 2 0.98824 4.047 14 58 7.679e-05 \*\*\*  
## Body\_Mass 1 0.95269 80.547 7 28 < 2.2e-16 \*\*\*  
## Water\_Trtmt:Stressor 2 0.45690 1.227 14 58 0.2819542   
## Water\_Trtmt:Body\_Mass 1 0.19194 0.950 7 28 0.4852397   
## Stressor:Body\_Mass 2 0.43597 1.155 14 58 0.3336011   
## Water\_Trtmt:Stressor:Body\_Mass 2 0.20070 0.462 14 58 0.9439509   
## Residuals 34   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary.aov(man1)

## Response Body\_Length :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Water\_Trtmt 1 7.989 7.989 8.3682 0.006617 \*\*   
## Stressor 2 5.927 2.964 3.1042 0.057775 .   
## Body\_Mass 1 85.532 85.532 89.5896 4.676e-11 \*\*\*  
## Water\_Trtmt:Stressor 2 1.875 0.937 0.9819 0.384980   
## Water\_Trtmt:Body\_Mass 1 3.264 3.264 3.4191 0.073157 .   
## Stressor:Body\_Mass 2 7.362 3.681 3.8557 0.030962 \*   
## Water\_Trtmt:Stressor:Body\_Mass 2 1.367 0.683 0.7157 0.496055   
## Residuals 34 32.460 0.955   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Response Body\_Depth :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Water\_Trtmt 1 4.933 4.933 9.0637 0.0048894 \*\*   
## Stressor 2 11.517 5.758 10.5803 0.0002676 \*\*\*  
## Body\_Mass 1 102.694 102.694 188.6849 1.955e-15 \*\*\*  
## Water\_Trtmt:Stressor 2 3.421 1.710 3.1426 0.0559340 .   
## Water\_Trtmt:Body\_Mass 1 0.692 0.692 1.2722 0.2672508   
## Stressor:Body\_Mass 2 1.307 0.654 1.2010 0.3133353   
## Water\_Trtmt:Stressor:Body\_Mass 2 0.575 0.288 0.5286 0.5942087   
## Residuals 34 18.505 0.544   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Response Tail\_Muscle\_Height :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Water\_Trtmt 1 4.436 4.436 8.8977 0.005252 \*\*   
## Stressor 2 6.344 3.172 6.3620 0.004497 \*\*   
## Body\_Mass 1 91.463 91.463 183.4563 2.935e-15 \*\*\*  
## Water\_Trtmt:Stressor 2 0.066 0.033 0.0663 0.936015   
## Water\_Trtmt:Body\_Mass 1 0.190 0.190 0.3813 0.541002   
## Stressor:Body\_Mass 2 0.136 0.068 0.1368 0.872610   
## Water\_Trtmt:Stressor:Body\_Mass 2 1.059 0.530 1.0622 0.356897   
## Residuals 34 16.951 0.499   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Response Tail\_Length :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Water\_Trtmt 1 71.77 71.77 16.9389 0.0002321 \*\*\*  
## Stressor 2 110.46 55.23 13.0352 6.279e-05 \*\*\*  
## Body\_Mass 1 959.09 959.09 226.3692 < 2.2e-16 \*\*\*  
## Water\_Trtmt:Stressor 2 4.97 2.49 0.5866 0.5617432   
## Water\_Trtmt:Body\_Mass 1 1.69 1.69 0.3989 0.5319026   
## Stressor:Body\_Mass 2 3.41 1.71 0.4028 0.6715726   
## Water\_Trtmt:Stressor:Body\_Mass 2 3.47 1.74 0.4096 0.6671451   
## Residuals 34 144.05 4.24   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Response Tail\_Fin\_Height :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Water\_Trtmt 1 9.040 9.040 7.0550 0.01195 \*   
## Stressor 2 68.717 34.359 26.8137 1.024e-07 \*\*\*  
## Body\_Mass 1 80.636 80.636 62.9286 3.062e-09 \*\*\*  
## Water\_Trtmt:Stressor 2 2.926 1.463 1.1419 0.33116   
## Water\_Trtmt:Body\_Mass 1 0.246 0.246 0.1919 0.66412   
## Stressor:Body\_Mass 2 3.181 1.591 1.2413 0.30178   
## Water\_Trtmt:Stressor:Body\_Mass 2 0.031 0.016 0.0122 0.98792   
## Residuals 34 43.567 1.281   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Response Body\_Width :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Water\_Trtmt 1 9.606 9.606 20.8652 6.196e-05 \*\*\*  
## Stressor 2 8.863 4.432 9.6260 0.0004869 \*\*\*  
## Body\_Mass 1 120.244 120.244 261.1852 < 2.2e-16 \*\*\*  
## Water\_Trtmt:Stressor 2 3.501 1.751 3.8023 0.0323393 \*   
## Water\_Trtmt:Body\_Mass 1 0.793 0.793 1.7229 0.1981083   
## Stressor:Body\_Mass 2 0.723 0.361 0.7848 0.4643131   
## Water\_Trtmt:Stressor:Body\_Mass 2 0.750 0.375 0.8150 0.4511107   
## Residuals 34 15.653 0.460   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Response Tail\_Fin\_Width :  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Water\_Trtmt 1 3.588 3.588 18.8400 0.0001208 \*\*\*  
## Stressor 2 0.190 0.095 0.4998 0.6110314   
## Body\_Mass 1 39.376 39.376 206.7629 5.155e-16 \*\*\*  
## Water\_Trtmt:Stressor 2 0.201 0.101 0.5282 0.5944170   
## Water\_Trtmt:Body\_Mass 1 0.760 0.760 3.9924 0.0537546 .   
## Stressor:Body\_Mass 2 0.265 0.132 0.6947 0.5061694   
## Water\_Trtmt:Stressor:Body\_Mass 2 0.064 0.032 0.1688 0.8453733   
## Residuals 34 6.475 0.190   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## 98 observations deleted due to missingness

MANCOVA to test homogeneity of slopes. We will look at how each specific body measurement/dimension responded to both treatments, while maintaining body mass as a covariate. All pass, which signifies the slopes of the lines for these traits were parallel amongst treatments. Continuing on.

We now must collect our Estimated Marginal Means.

model\_BodyLength <- lm(Body\_Length ~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
means\_complex\_BodyLength <- estimate\_means(model\_BodyLength)

## We selected `at = c("Water\_Trtmt", "Stressor")`.

## NOTE: Results may be misleading due to involvement in interactions

means\_complex\_BodyLength$Mean

## [1] 23.26811 23.76371 23.52869 23.23572 23.65609 23.87981

These are our EMMs for Body Length Across Treatments. This is now a new data frame in the global environment.

Order: AA, NA, AB, NB, AC, NC

model\_BodyDepth <- lm(Body\_Depth ~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
means\_complex\_BodyDepth <- estimate\_means(model\_BodyDepth)

## We selected `at = c("Water\_Trtmt", "Stressor")`.

## NOTE: Results may be misleading due to involvement in interactions

means\_complex\_BodyDepth$Mean

## [1] 14.23012 14.68386 14.38029 13.76895 14.42708 14.58465

model\_TailMuscleHeight <- lm(Tail\_Muscle\_Height ~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
means\_complex\_TailMuscleHeight <- estimate\_means(model\_TailMuscleHeight)

## We selected `at = c("Water\_Trtmt", "Stressor")`.

## NOTE: Results may be misleading due to involvement in interactions

means\_complex\_TailMuscleHeight$Mean

## [1] 10.41262 10.52105 10.11888 10.25704 10.54756 10.72924

model\_TailLength <- lm(Tail\_Length ~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
means\_complex\_TailLength <- estimate\_means(model\_TailLength)

## We selected `at = c("Water\_Trtmt", "Stressor")`.

## NOTE: Results may be misleading due to involvement in interactions

means\_complex\_TailLength$Mean

## [1] 49.82343 49.42929 49.84423 50.59415 48.96342 49.51957

model\_TailFinHeight <- lm(Tail\_Fin\_Height ~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
means\_complex\_TailFinHeight <- estimate\_means(model\_TailFinHeight)

## We selected `at = c("Water\_Trtmt", "Stressor")`.

## NOTE: Results may be misleading due to involvement in interactions

means\_complex\_TailFinHeight$Mean

## [1] 17.73924 17.54926 16.60320 17.88404 15.91858 15.78763

model\_BodyWidth <- lm(Body\_Width ~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
means\_complex\_BodyWidth <- estimate\_means(model\_BodyWidth)

## We selected `at = c("Water\_Trtmt", "Stressor")`.

## NOTE: Results may be misleading due to involvement in interactions

means\_complex\_BodyWidth$Mean

## [1] 16.43980 16.81783 16.52385 16.09063 16.51585 17.21426

model\_TailFinWidth <- lm(Tail\_Fin\_Width ~Water\_Trtmt\*Stressor\*Body\_Mass, data = df)  
  
means\_complex\_TailFinWidth <- estimate\_means(model\_TailFinWidth)

## We selected `at = c("Water\_Trtmt", "Stressor")`.

## NOTE: Results may be misleading due to involvement in interactions

means\_complex\_TailFinWidth$Mean

## [1] 7.317604 7.408261 7.479050 7.709226 8.023434 8.166225

We now have all of our EMMs. Now, we will get our residuals.

residuals <- man1$residuals  
man1$residuals

## Body\_Length Body\_Depth Tail\_Muscle\_Height Tail\_Length Tail\_Fin\_Height  
## 5 -0.556078500 0.17391817 0.83688678 -1.14598984 0.84276679  
## 6 1.943354733 0.59268719 -0.87839337 -2.90222194 -1.57637469  
## 11 -0.915817576 -0.26633240 -0.53702621 -0.55212329 -1.13802216  
## 12 -1.944560516 -0.66910714 -0.46792403 -1.24628687 -0.65526093  
## 17 -0.129608242 -0.47344751 0.18127571 0.88718296 1.15459973  
## 18 0.445122411 -0.34102172 0.64210630 3.50922284 0.94712880  
## 23 -0.838407331 0.41161261 0.45649522 -0.39760436 -0.20103825  
## 24 1.995995021 0.57169081 -0.23342040 1.84782049 0.62620070  
## 29 0.001874873 0.88557557 -0.46746047 1.43290135 1.96584198  
## 30 -0.611311913 0.17900733 0.17063384 3.10806983 0.68976990  
## 35 0.523770638 1.83699756 0.45312205 -1.15768994 0.77674337  
## 36 0.389171475 0.05986974 0.84416782 1.56654286 1.46304878  
## 41 1.657243881 -1.73504044 -0.79427046 1.28113490 -0.76078580  
## 42 -0.595268545 -0.72981708 0.17449238 -1.27922804 -1.49597983  
## 47 -1.165453558 -0.56220681 0.24305547 -2.07673943 -1.82429802  
## 48 -0.200026851 0.06561412 -0.62374063 -2.87499152 -0.81434038  
## 53 0.627371352 0.41861276 0.82257774 0.47749504 1.30866680  
## 54 1.206346187 0.47237364 0.89897051 -2.62318385 1.48552725  
## 59 -0.824586877 0.16264931 -0.11679174 -1.34271761 -1.45119763  
## 60 -0.380655322 -0.30560794 0.04833203 0.24235174 0.20125488  
## 65 -0.061007109 -0.72423155 -0.86712691 0.36569607 0.35839260  
## 66 -0.673001070 0.37769479 -0.29457299 0.66355370 0.59548206  
## 71 0.414918407 -0.42441514 -0.76455706 0.31290781 -1.06624435  
## 72 -0.309385569 0.02292414 0.27316843 1.90389711 -1.43188161  
## 77 0.600792084 0.87566172 0.35989071 2.35073336 0.44854968  
## 78 0.273130443 0.48286419 -0.43950874 1.00726331 1.13080653  
## 83 -0.083609833 1.06355843 -0.18285728 -3.88982575 0.91528732  
## 84 -0.849395628 0.11918717 0.54929780 -0.06555444 -1.01979721  
## 89 -1.031042048 -1.54688132 -0.79375428 -2.30449961 -1.28114297  
## 90 0.144210065 -0.76455160 -0.25027651 0.19614889 -0.61986087  
## 95 0.354943718 -0.20753438 1.56518149 2.24624208 0.37401619  
## 96 0.590971198 -0.02230422 -0.80797320 0.45949216 0.05214132  
## 101 0.813992792 -0.12025887 1.04519070 2.02966009 1.70270874  
## 102 0.406625060 0.56891223 -0.12076913 3.12633059 0.23661144  
## 113 -0.379587922 -0.40739551 -0.14680490 -2.43734751 -0.47012904  
## 114 1.191853030 0.05545335 -0.88747743 -0.60800404 0.19698800  
## 119 -1.005488068 -0.03478486 0.21904101 1.20545906 -0.67999648  
## 120 -1.027394893 -0.06192634 -0.10918024 -3.31609818 -0.98618266  
## 125 -0.399612627 0.96442151 -0.14361761 0.73825859 0.63249847  
## 126 0.447437891 -0.53745322 -0.66512360 -1.32899555 -0.01587830  
## 131 0.635407990 -0.33415269 0.73096577 0.17724154 -0.21220049  
## 132 0.219068813 -0.10676736 0.36324557 1.16178382 0.53541898  
## 137 -0.700031184 -0.02031453 0.91175146 0.90947888 -0.42292674  
## 138 0.917938915 -0.41701400 -0.38215917 -0.75428404 0.21075106  
## 143 -0.242646651 0.07143688 -0.65444491 0.14869908 -1.05688393  
## 144 -0.877563147 0.37984342 -0.16061750 -1.05218233 0.32922096  
## Body\_Width Tail\_Fin\_Width  
## 5 0.71389607 0.66290860  
## 6 0.44261984 -0.27951174  
## 11 -0.57897684 0.04772773  
## 12 -0.22961376 0.09771909  
## 17 0.34730032 -0.63336186  
## 18 -0.41801841 -0.26307763  
## 23 -0.36461416 0.40065191  
## 24 0.08740696 -0.03305609  
## 29 0.21489035 -0.23690870  
## 30 -0.00944740 0.50920369  
## 35 -0.02185087 -0.66140123  
## 36 0.03794557 -0.19720229  
## 41 -0.03753320 0.54480985  
## 42 -0.76115804 0.15939858  
## 47 -0.17038487 0.15090450  
## 48 0.74753845 -0.26880441  
## 53 0.30018683 0.06802935  
## 54 1.45627930 0.34051882  
## 59 -0.22489059 -0.24401475  
## 60 -0.54017661 -0.86679440  
## 65 -0.89118521 0.16861400  
## 66 -0.91536156 0.06786685  
## 71 1.51770511 0.29658150  
## 72 -0.70255726 0.16919864  
## 77 0.86728483 0.78736871  
## 78 -0.23081895 -0.22308154  
## 83 0.08141299 -0.08760935  
## 84 0.73626109 0.52394261  
## 89 -1.02099510 -1.00387215  
## 90 -0.61418696 0.29528230  
## 95 -0.02487498 -0.07222136  
## 96 0.20591708 -0.21980922  
## 101 0.73927394 0.16995289  
## 102 0.29613414 0.08574430  
## 113 -0.31824682 -0.05179422  
## 114 -0.01821914 0.06386859  
## 119 -0.04696875 0.10241946  
## 120 -0.65197337 -0.37019101  
## 125 0.56913481 -0.27185360  
## 126 -0.37083589 0.20743214  
## 131 -0.23145745 0.17054075  
## 132 -0.19711642 0.60805582  
## 137 -0.14172429 -0.24897635  
## 138 -0.72234585 0.07413227  
## 143 0.90063171 -0.22522386  
## 144 0.19371338 -0.31410717

Here are our residuals for rach body dimension measurement, generated by the MANCOVA. Below is a code that can be used if this is the first time you are doing this analysis:

# write.table(residuals, file = “dfwithresidualsbodymorph.csv”,sep = “,”)

If you are doing this for the first time, this will give you a new data frame CSV file containing your individuals and their residuals. Once you have your residuals correctly assigned to each tadpole, we are going to mass adjust those variables. To do so, I am going to add a new column with EMM to each column. Then, I am going to add a third column represented our mass adjusted body morph next to the EMM column I will add the Residuals for each body dimension to the EMM based on treatment group and have the resuing number be our MAV in the orginial CSV file. Now, we can conduct our PCA.

#write.table(residuals, file = "dfwithresidualsbodymorph.csv",sep = ",")  
  
df.pca = df[,19:25]

Creates Data Frame for ONLY PCA values of interest.

KMO(df.pca)

## Kaiser-Meyer-Olkin factor adequacy  
## Call: KMO(r = df.pca)  
## Overall MSA = 0.52  
## MSA for each item =   
## MAV\_Body\_Length MAV\_Body\_Depth MAV\_Tail\_Muscle\_Height   
## 0.57 0.57 0.55   
## MAV\_Tail\_Length MAV\_Tail\_Fin\_Height MAV\_Body\_Width   
## 0.44 0.43 0.61   
## MAV\_Tail\_Fin\_Width   
## 0.44

Overall, we pass our KMO test.

cortest.bartlett(df.pca)

## R was not square, finding R from data

## $chisq  
## [1] 181.8613  
##   
## $p.value  
## [1] 1.28842e-27  
##   
## $df  
## [1] 21

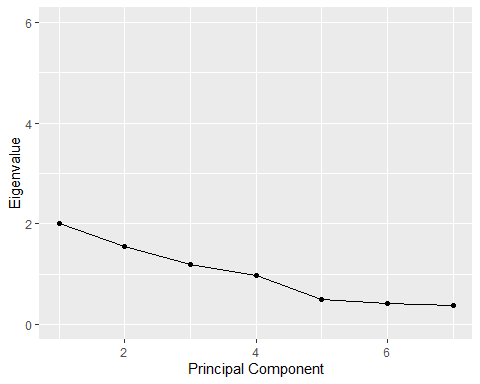
Passed Bartletts.

pca.p <- principal(df.pca, nfactors = 3, rotate = "varimax")

Output: Valuees = Eigenvalues, Scores = Factor Scores.

qplot(c(1:7), pca.p$values) +  
 geom\_line() +  
 xlab("Principal Component") +  
 ylab("Eigenvalue") +  
 ylim(0,6)

## Warning: `qplot()` was deprecated in ggplot2 3.4.0.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

 Looks like we have 3 Principal Components for this data set.

pca.p$values

## [1] 2.0132562 1.5417609 1.1920082 0.9786549 0.4850978 0.4201108 0.3691111

Eigenvalues.

pca.p$scores

## RC1 RC2 RC3  
## [1,] NA NA NA  
## [2,] NA NA NA  
## [3,] NA NA NA  
## [4,] NA NA NA  
## [5,] 1.183439841 0.32717432 0.14773805  
## [6,] 1.839666527 -2.02329709 -0.84072160  
## [7,] NA NA NA  
## [8,] NA NA NA  
## [9,] NA NA NA  
## [10,] NA NA NA  
## [11,] -0.466460461 -0.77547729 -0.34795650  
## [12,] -0.774100722 -0.87690680 -0.28471882  
## [13,] NA NA NA  
## [14,] NA NA NA  
## [15,] NA NA NA  
## [16,] NA NA NA  
## [17,] 0.436784594 0.44972524 -1.59131761  
## [18,] 0.092459048 1.77273388 -1.01146826  
## [19,] NA NA NA  
## [20,] NA NA NA  
## [21,] NA NA NA  
## [22,] NA NA NA  
## [23,] 0.383289653 0.05750959 -0.14165894  
## [24,] 1.502697678 0.64199739 -1.07587907  
## [25,] NA NA NA  
## [26,] NA NA NA  
## [27,] NA NA NA  
## [28,] NA NA NA  
## [29,] -0.192328942 1.07164742 -1.59980352  
## [30,] -1.146009412 2.01758623 0.20470683  
## [31,] NA NA NA  
## [32,] NA NA NA  
## [33,] NA NA NA  
## [34,] NA NA NA  
## [35,] 0.800254491 0.13895069 -1.85749430  
## [36,] -0.548421059 1.91605185 -0.77162895  
## [37,] NA NA NA  
## [38,] NA NA NA  
## [39,] NA NA NA  
## [40,] NA NA NA  
## [41,] -1.864763926 1.04097286 1.08751910  
## [42,] -2.041931085 0.09905760 0.60983067  
## [43,] NA NA NA  
## [44,] NA NA NA  
## [45,] NA NA NA  
## [46,] NA NA NA  
## [47,] -1.685361659 -0.36742403 0.74939787  
## [48,] -0.361746801 -1.13588857 -0.44874568  
## [49,] NA NA NA  
## [50,] NA NA NA  
## [51,] NA NA NA  
## [52,] NA NA NA  
## [53,] 1.529785543 0.80412530 0.87966266  
## [54,] 2.748390370 -0.14181022 1.27684526  
## [55,] NA NA NA  
## [56,] NA NA NA  
## [57,] NA NA NA  
## [58,] NA NA NA  
## [59,] 0.336477662 -1.17717746 1.15939554  
## [60,] -0.005441917 -0.18861883 -0.04037520  
## [61,] NA NA NA  
## [62,] NA NA NA  
## [63,] NA NA NA  
## [64,] NA NA NA  
## [65,] -0.622810981 -0.08362022 1.05608642  
## [66,] 0.023808794 0.10984819 0.64761936  
## [67,] NA NA NA  
## [68,] NA NA NA  
## [69,] NA NA NA  
## [70,] NA NA NA  
## [71,] 1.217163727 -0.67757344 2.04183295  
## [72,] -0.080479165 0.40140318 1.83833423  
## [73,] NA NA NA  
## [74,] NA NA NA  
## [75,] NA NA NA  
## [76,] NA NA NA  
## [77,] 1.018215247 1.30100610 0.09291284  
## [78,] -0.033275602 0.39958013 -1.73814339  
## [79,] NA NA NA  
## [80,] NA NA NA  
## [81,] NA NA NA  
## [82,] NA NA NA  
## [83,] 0.847719770 -1.27365582 -1.67588198  
## [84,] 0.061264346 0.14255391 0.48208790  
## [85,] NA NA NA  
## [86,] NA NA NA  
## [87,] NA NA NA  
## [88,] NA NA NA  
## [89,] -2.318089715 -1.47710223 -1.53674056  
## [90,] -1.218469678 0.21584840 -0.08439961  
## [91,] NA NA NA  
## [92,] NA NA NA  
## [93,] NA NA NA  
## [94,] NA NA NA  
## [95,] -0.127181639 1.87063025 -0.44829117  
## [96,] -0.091156138 -0.20547407 -1.22923994  
## [97,] NA NA NA  
## [98,] NA NA NA  
## [99,] NA NA NA  
## [100,] NA NA NA  
## [101,] 0.681475419 1.47521491 -0.16573140  
## [102,] 0.355093779 0.67825309 -0.32696705  
## [103,] NA NA NA  
## [104,] NA NA NA  
## [105,] NA NA NA  
## [106,] NA NA NA  
## [107,] NA NA NA  
## [108,] NA NA NA  
## [109,] NA NA NA  
## [110,] NA NA NA  
## [111,] NA NA NA  
## [112,] NA NA NA  
## [113,] -0.603088529 -1.24889246 -0.18524069  
## [114,] 0.197540703 -0.81263131 -0.51846888  
## [115,] NA NA NA  
## [116,] NA NA NA  
## [117,] NA NA NA  
## [118,] NA NA NA  
## [119,] -0.605266829 0.01784438 0.19255955  
## [120,] -0.739857395 -1.85148639 -0.58362287  
## [121,] NA NA NA  
## [122,] NA NA NA  
## [123,] NA NA NA  
## [124,] NA NA NA  
## [125,] 0.900742784 -0.36638363 -0.05985277  
## [126,] -0.459021955 -0.88756369 0.92801943  
## [127,] NA NA NA  
## [128,] NA NA NA  
## [129,] NA NA NA  
## [130,] NA NA NA  
## [131,] -0.031053168 0.27610382 1.32701996  
## [132,] -0.103895086 0.68363296 1.46854058  
## [133,] NA NA NA  
## [134,] NA NA NA  
## [135,] NA NA NA  
## [136,] NA NA NA  
## [137,] -0.220970264 0.23253670 0.83712389  
## [138,] -0.414766641 -0.47051032 0.67985936  
## [139,] NA NA NA  
## [140,] NA NA NA  
## [141,] NA NA NA  
## [142,] NA NA NA  
## [143,] 0.390276708 -1.14088159 0.78850313  
## [144,] 0.209402085 -0.95961294 0.06875319

Our factor scores to be analyzed for our PCA. If conducting for the first time, you can bind these factor scores to the original master data frame by: # Factor <-pca.p$scores # df <- cbind(df,Factor)

Factor <-pca.p$scores  
df <- cbind(df,Factor)  
write.table(df, file = "EM3dfwithfactorscores,3PCs.csv", sep = ",")

Now, scores are added correctly to our master data frame.

pca.p$loadings

##   
## Loadings:  
## RC1 RC2 RC3   
## MAV\_Body\_Length 0.566 0.178   
## MAV\_Body\_Depth 0.829 -0.196  
## MAV\_Tail\_Muscle\_Height 0.344 0.534 0.244  
## MAV\_Tail\_Length 0.882   
## MAV\_Tail\_Fin\_Height 0.125 0.582 -0.688  
## MAV\_Body\_Width 0.820 0.260  
## MAV\_Tail\_Fin\_Width 0.152 0.209 0.882  
##   
## RC1 RC2 RC3  
## SS loadings 1.846 1.480 1.422  
## Proportion Var 0.264 0.211 0.203  
## Cumulative Var 0.264 0.475 0.678

PC1 Loads: Body Depth, Body Width PC2 Loads: Tail Length PC3 Loads: Tail Fin Width, Tail Fin Height (negative) Unloaded: Body Length, Tail Muscle Height

Now, we can statisictially analyze whether our treatments impacted Body Morphology.

pca1\_glmm <- glmer(RC1~Water\_Trtmt\*Stressor\*Gosner\_Stage + (1|Replicate), data = df, na.action = na.omit, family = "gaussian")

## Warning in glmer(RC1 ~ Water\_Trtmt \* Stressor \* Gosner\_Stage + (1 |  
## Replicate), : calling glmer() with family=gaussian (identity link) as a shortcut  
## to lmer() is deprecated; please call lmer() directly

Anova(pca1\_glmm)

## Analysis of Deviance Table (Type II Wald chisquare tests)  
##   
## Response: RC1  
## Chisq Df Pr(>Chisq)   
## Water\_Trtmt 0.6441 1 0.42222   
## Stressor 4.6646 2 0.09707 .  
## Gosner\_Stage 0.2160 1 0.64214   
## Water\_Trtmt:Stressor 3.7456 2 0.15370   
## Water\_Trtmt:Gosner\_Stage 0.0083 1 0.92753   
## Stressor:Gosner\_Stage 1.0350 2 0.59600   
## Water\_Trtmt:Stressor:Gosner\_Stage 2.1620 2 0.33925   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

emmeans(pca1\_glmm, list(pairwise ~ Stressor), adjust = "tukey")

## NOTE: Results may be misleading due to involvement in interactions

## $`emmeans of Stressor`  
## Stressor emmean SE df lower.CL upper.CL  
## A 0.150 0.304 15.6 -0.496 0.796  
## B -0.488 0.328 15.8 -1.185 0.208  
## C 0.290 0.315 17.3 -0.374 0.954  
##   
## Results are averaged over the levels of: Water\_Trtmt   
## Degrees-of-freedom method: kenward-roger   
## Confidence level used: 0.95   
##   
## $`pairwise differences of Stressor`  
## 1 estimate SE df t.ratio p.value  
## A - B 0.638 0.447 15.7 1.427 0.3516  
## A - C -0.140 0.438 16.4 -0.320 0.9452  
## B - C -0.779 0.455 16.5 -1.711 0.2307  
##   
## Results are averaged over the levels of: Water\_Trtmt   
## Degrees-of-freedom method: kenward-roger   
## P value adjustment: tukey method for comparing a family of 3 estimates

N.S.

pca2\_glmm <- glmer(RC2~Water\_Trtmt\*Stressor\*Gosner\_Stage + (1|Replicate), data = df, na.action = na.omit, family = "gaussian")

## Warning in glmer(RC2 ~ Water\_Trtmt \* Stressor \* Gosner\_Stage + (1 |  
## Replicate), : calling glmer() with family=gaussian (identity link) as a shortcut  
## to lmer() is deprecated; please call lmer() directly

Anova(pca2\_glmm)

## Analysis of Deviance Table (Type II Wald chisquare tests)  
##   
## Response: RC2  
## Chisq Df Pr(>Chisq)  
## Water\_Trtmt 0.8515 1 0.3561  
## Stressor 0.9046 2 0.6362  
## Gosner\_Stage 0.4350 1 0.5096  
## Water\_Trtmt:Stressor 1.1737 2 0.5561  
## Water\_Trtmt:Gosner\_Stage 0.1224 1 0.7265  
## Stressor:Gosner\_Stage 2.4107 2 0.2996  
## Water\_Trtmt:Stressor:Gosner\_Stage 0.0300 2 0.9851

emmeans(pca2\_glmm, list(pairwise ~ Stressor), adjust = "tukey")

## NOTE: Results may be misleading due to involvement in interactions

## $`emmeans of Stressor`  
## Stressor emmean SE df lower.CL upper.CL  
## A -0.0205 0.322 14.9 -0.708 0.667  
## B 0.1935 0.349 15.5 -0.548 0.935  
## C -0.2196 0.340 17.1 -0.937 0.498  
##   
## Results are averaged over the levels of: Water\_Trtmt   
## Degrees-of-freedom method: kenward-roger   
## Confidence level used: 0.95   
##   
## $`pairwise differences of Stressor`  
## 1 estimate SE df t.ratio p.value  
## A - B -0.214 0.475 15.2 -0.450 0.8950  
## A - C 0.199 0.469 16.0 0.425 0.9058  
## B - C 0.413 0.487 16.2 0.848 0.6796  
##   
## Results are averaged over the levels of: Water\_Trtmt   
## Degrees-of-freedom method: kenward-roger   
## P value adjustment: tukey method for comparing a family of 3 estimates

NS

pca3\_glmm <- glmer(RC3~Water\_Trtmt\*Stressor\*Gosner\_Stage + (1|Replicate), data = df, na.action = na.omit, family = "gaussian")

## Warning in glmer(RC3 ~ Water\_Trtmt \* Stressor \* Gosner\_Stage + (1 |  
## Replicate), : calling glmer() with family=gaussian (identity link) as a shortcut  
## to lmer() is deprecated; please call lmer() directly

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

Anova(pca3\_glmm)

## Analysis of Deviance Table (Type II Wald chisquare tests)  
##   
## Response: RC3  
## Chisq Df Pr(>Chisq)   
## Water\_Trtmt 0.1415 1 0.7068   
## Stressor 45.8966 2 1.081e-10 \*\*\*  
## Gosner\_Stage 2.2789 1 0.1311   
## Water\_Trtmt:Stressor 0.0363 2 0.9820   
## Water\_Trtmt:Gosner\_Stage 0.1346 1 0.7137   
## Stressor:Gosner\_Stage 4.3415 2 0.1141   
## Water\_Trtmt:Stressor:Gosner\_Stage 4.3206 2 0.1153   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Strong effect of stressor on PC3 - Tail Fin Width

emmeans(pca3\_glmm, list(pairwise ~ Stressor), adjust = "tukey")

## NOTE: Results may be misleading due to involvement in interactions

## $`emmeans of Stressor`  
## Stressor emmean SE df lower.CL upper.CL  
## A -0.712 0.177 13.9 -1.092 -0.332  
## B -0.221 0.194 15.2 -0.634 0.192  
## C 0.834 0.193 16.1 0.426 1.243  
##   
## Results are averaged over the levels of: Water\_Trtmt   
## Degrees-of-freedom method: kenward-roger   
## Confidence level used: 0.95   
##   
## $`pairwise differences of Stressor`  
## 1 estimate SE df t.ratio p.value  
## A - B -0.491 0.263 14.6 -1.869 0.1827  
## A - C -1.546 0.262 15.1 -5.910 0.0001  
## B - C -1.055 0.273 15.7 -3.861 0.0039  
##   
## Results are averaged over the levels of: Water\_Trtmt   
## Degrees-of-freedom method: kenward-roger   
## P value adjustment: tukey method for comparing a family of 3 estimates

Significant differences between our CORT group and our other two!

#pca4\_glmm <- glmer(RC4~Water\_Trtmt\*Stressor\*Gosner\_Stage + (1|Replicate), data = df, na.action = na.omit, family = "gaussian")  
  
#Anova(pca4\_glmm)

#Stress <- c("Vehicle Control", "Predator Cues", "CORT")  
#For labeling X axis  
  
# ggplot(df, aes(x= factor(Stressor, level=c('B', 'A','C')), y = RC3, fill = Water\_Trtmt)) +  
# geom\_boxplot(width = 0.5, outlier.colour = "transparent", alpha =0.35) +  
# stat\_boxplot(geom = "errorbar", width = .5) +  
# theme\_classic() +  
# scale\_fill\_manual(values = c("seagreen3", "lightcyan"),  
# name = "Microbial Environment",  
# labels = c("Natural", "Autoclaved")) +  
# labs(x = "Stressor", y = "PC-3") +  
# theme(aspect.ratio = 1) +  
# coord\_cartesian(ylim = c(-2, 3)) +  
# scale\_x\_discrete(labels = Stress)+  
# theme(axis.text = element\_text(face = "bold", size = 16)) +  
# theme(axis.title = element\_text(face = "bold", size = 20)) +  
# theme(legend.position = c(.83, .92)) +  
# theme(legend.title = element\_text(size = 14)) +  
# theme(legend.text = element\_text(size = 14))

# ggplot(df, aes(x= factor(Stressor, level=c('B', 'A','C')), y = RC1, fill = Water\_Trtmt)) +  
# geom\_boxplot(width = 0.5, outlier.colour = "transparent", alpha =0.35) +  
# stat\_boxplot(geom = "errorbar", width = .5) +  
# theme\_classic() +  
# scale\_fill\_manual(values = c("seagreen3", "lightcyan"),  
# name = "Microbial Environment",  
# labels = c("Natural", "Autoclaved")) +  
# labs(x = "Stressor", y = "PC-1") +  
# theme(aspect.ratio = 1) +  
# coord\_cartesian(ylim = c(-2, 3)) +  
# scale\_x\_discrete(labels = Stress)+  
# theme(axis.text = element\_text(face = "bold", size = 16)) +  
# theme(axis.title = element\_text(face = "bold", size = 20)) +  
# theme(legend.position = c(.83, .92)) +  
# theme(legend.title = element\_text(size = 14)) +  
# theme(legend.text = element\_text(size = 14)) +  
# theme(legend.position="none")

# ggplot(df, aes(x= factor(Stressor, level=c('B', 'A','C')), y = RC2, fill = Water\_Trtmt)) +  
# geom\_boxplot(width = 0.5, outlier.colour = "transparent", alpha =0.35) +  
# stat\_boxplot(geom = "errorbar", width = .5) +  
# theme\_classic() +  
# scale\_fill\_manual(values = c("seagreen3", "lightcyan"),  
# name = "Microbial Environment",  
# labels = c("Natural", "Autoclaved")) +  
# labs(x = "Stressor", y = "PC-2") +  
# theme(aspect.ratio = 1) +  
# coord\_cartesian(ylim = c(-2, 3)) +  
# scale\_x\_discrete(labels = Stress)+  
# theme(axis.text = element\_text(face = "bold", size = 16)) +  
# theme(axis.title = element\_text(face = "bold", size = 20)) +  
# theme(legend.position = c(.83, .92)) +  
# theme(legend.title = element\_text(size = 14)) +  
# theme(legend.text = element\_text(size = 14)) +  
# theme(legend.position="none")