

## eye\_morph\_selection\_analyses.R

```
#####Full analysis for: Wright et al. Adaptive divergence in Heliconius eyes#####
```

```
#Load packages
```

```
library(MASS)
```

```
library(lme4)
```

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

```
library(ggplot2)
```

```
library(stats)
```

```
library(effects)
```

```
## Loading required package: carData
```

```
## lattice theme set by effectsTheme()
```

```
## See ?effectsTheme for details.
```

```
library(splines)
```

```
library(xlsx)
```

```
library(car)
```

```
library(lattice)
```

```
library(grid)
```

```
library(gridExtra)
```

```
library(pbkrtest)
```

```
library(readxl)
```

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:gridExtra':
```

```
##
```

```
##      combine
```

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

```
##
```

```
##      recode
```

```
## The following object is masked from 'package:MASS':
```

```
##
```

```
##      select
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```

## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union

#import data
eye <- read.csv("~/Documents/LMU/Eye morphology/selection manuscript/eye_morphology_cp_mp_f1.csv")

#is the data properly structured?
str(eye)

## 'data.frame': 131 obs. of 14 variables:
## $ ID : chr "CAM_13031" "CAM_13086" "CAM_13087" "CAM_13118" .
## ..
## $ collection : chr "PAN_2007" "PAN_2007" "PAN_2007" "PAN_2007" ...
## $ sex : chr "female" "female" "female" "female" ...
## $ type : chr "CP" "CP" "CP" "CP" ...
## $ brood : chr "stock" "stock" "stock" "stock" ...
## $ wild_insectary : chr "insectary" "insectary" "insectary" "insectary" .
## ..
## $ raised : chr "Gamboa" "Gamboa" "Gamboa" "Gamboa" ...
## $ observer : chr "Shane" "Shane" "Shane" "Shane" ...
## $ inter_eye_width: num 1.54 1.58 1.55 1.54 1.26 ...
## $ tibia_length : num NA NA 5.9 6 4.85 ...
## $ L_area : num 8.32 7.28 9.12 8.22 6.51 ...
## $ L_whole_count : int 14322 13569 15142 14059 11818 15035 13497 13746 1
5346 13951 ...
## $ R_area : num 8.2 7.12 8.76 8.42 NA ...
## $ R_whole_count : int 13707 13745 14954 14456 NA 15107 NA NA NA NA ...

eye$type <- as.factor(eye$type)
eye$sex <- as.factor(eye$sex)
eye$brood <- as.factor(eye$brood)
eye$wild_insectary <- as.factor(eye$wild_insectary)
eye$raised <- as.factor(eye$raised)
eye$observer <- as.factor(eye$observer)
str(eye) #now correct

## 'data.frame': 131 obs. of 14 variables:
## $ ID : chr "CAM_13031" "CAM_13086" "CAM_13087" "CAM_13118" .
## ..
## $ collection : chr "PAN_2007" "PAN_2007" "PAN_2007" "PAN_2007" ...
## $ sex : Factor w/ 2 levels "female","male": 1 1 1 1 2 2 2 1 1
1 ...
## $ type : Factor w/ 4 levels "CP","CPxMP","MP",...: 1 1 1 1 3 1 3
1 1 1 ...
## $ brood : Factor w/ 14 levels "check","N156",...: 14 14 14 14 NA
14 NA 14 14 14 ...
## $ wild_insectary : Factor w/ 1 level "insectary": 1 1 1 1 NA 1 NA 1 1 1 .
## ..

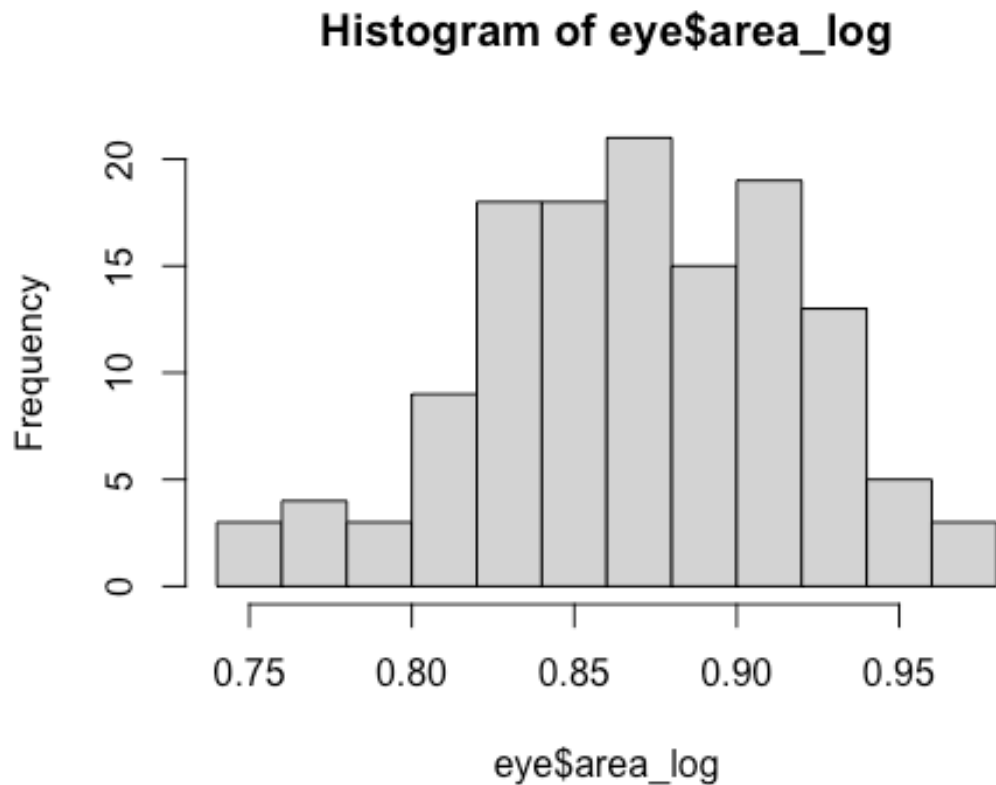
```

```
## $ raised      : Factor w/ 1 level "Gamboa": 1 1 1 1 1 1 1 1 1 1 ...
## $ observer    : Factor w/ 2 levels "Lisa","Shane": 2 2 2 2 1 2 1 1 1 1
...
## $ inter_eye_width: num  1.54 1.58 1.55 1.54 1.26 ...
## $ tibia_length   : num  NA NA 5.9 6 4.85 ...
## $ L_area         : num  8.32 7.28 9.12 8.22 6.51 ...
## $ L_whole_count  : int  14322 13569 15142 14059 11818 15035 13497 13746 1
5346 13951 ...
## $ R_area        : num  8.2 7.12 8.76 8.42 NA ...
## $ R_whole_count  : int  13707 13745 14954 14456 NA 15107 NA NA NA NA ...

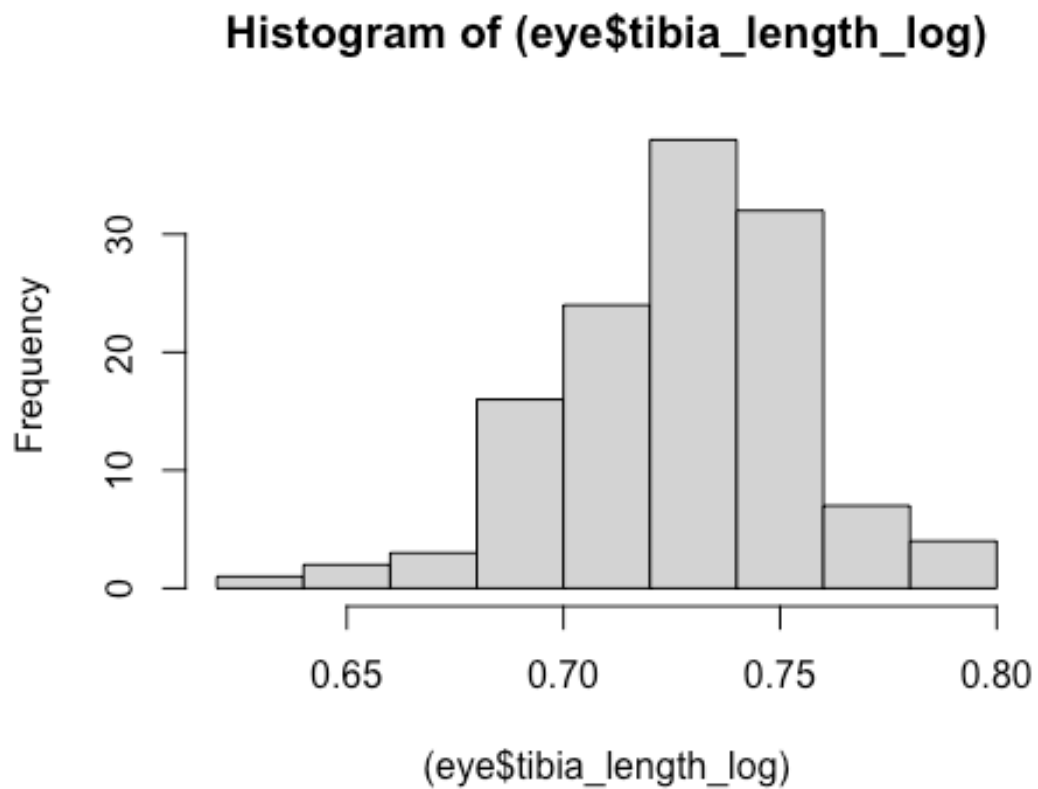
#####new composite variables of facet count & corneal area####
#when left eye was missing or damaged, right is used

eye$whole_count <- ifelse(is.na(eye$L_whole_count), eye$R_whole_count, eye$L_
whole_count)
eye$area <- ifelse(is.na(eye$L_area), eye$R_area, eye$L_area)

#log-transformed variables
eye$area_log <- (log10(eye$area))
hist(eye$area_log)
```

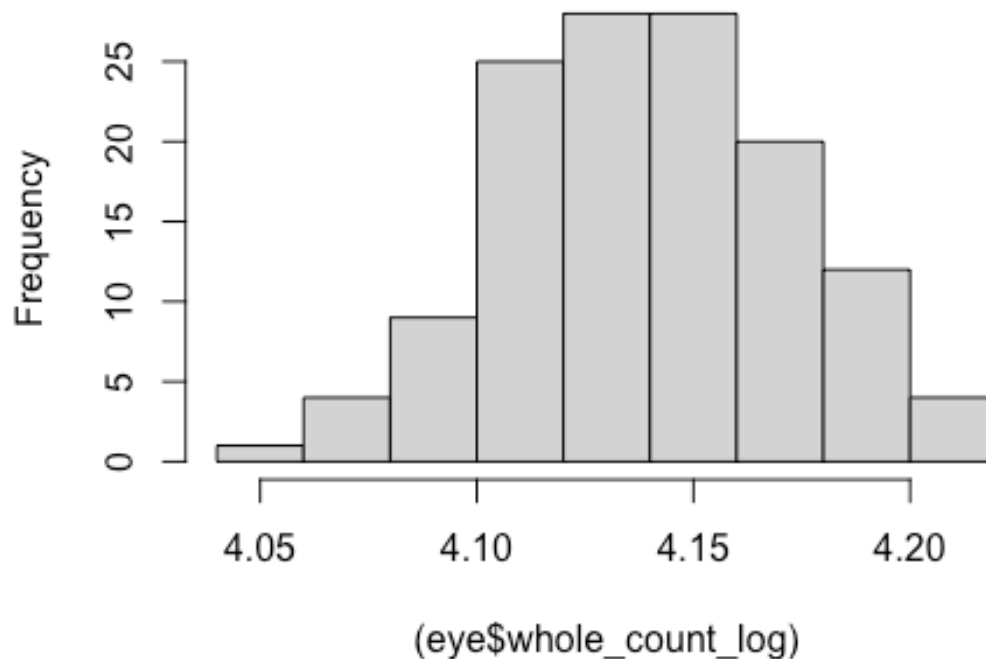


```
eye$tibia_length_log <- (log10(eye$tibia_length))  
hist((eye$tibia_length_log))
```



```
eye$whole_count_log <- (log10(eye$whole_count))  
hist((eye$whole_count_log))
```

# Histogram of (eye\$whole\_count\_log)



```
#some individuals are missing body size estimate (tibia length)
#remove from the dataset
eye <- eye[!(is.na(eye$tibia_length) | eye$tibia_length==""), ]
```

```
#how many per species & sex
t.first <- eye[match(unique(eye$ID), eye$ID),]
t.first %>%
  group_by(collection) %>%
  summarize(count = n())
```

```
## # A tibble: 3 × 2
##   collection count
##   <chr>      <int>
## 1 PAN_2007     12
## 2 PAN_2015     29
## 3 PAN_2017     86
```

```
#CP    female    19
#CP    male      15
#CPxMP female    15
#CPxMP male      15
#MP    female    15
#MP    male      18
```

```

#MPxCP female    15
#MPxCP male      15

#dataset without F1 hybrids
eye2 <- eye[-which(eye$type=="CPxMP"|
                  eye$type=="MPxCP"),]

#how many per species & sex
t.second <- eye2[match(unique(eye2$ID), eye2$ID),]
t.second %>%
  group_by(type,sex) %>%
  summarize(count = n())

## `summarise()` has grouped output by 'type'. You can override using the
## `.groups` argument.

## # A tibble: 4 × 3
## # Groups:   type [2]
##   type sex    count
##   <fct> <fct> <int>
## 1 CP    female    19
## 2 CP    male     15
## 3 MP    female    15
## 4 MP    male     18

#CP    female    19
#CP    male     15
#MP    female    15
#MP    male     18

####Correlations between left and right sides####
#using all processed samples (pure species + F1 hybrids)

cor.test(eye$L_whole_count,eye$R_whole_count, method="pearson")

##
## Pearson's product-moment correlation
##
## data: eye$L_whole_count and eye$R_whole_count
## t = 43.863, df = 116, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9586848 0.9798957
## sample estimates:
##      cor
## 0.971152

```

```

cor.test(eye$L_area,eye$R_area, method="pearson")

##
## Pearson's product-moment correlation
##
## data: eye$L_area and eye$R_area
## t = 63.881, df = 116, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.9800032 0.9903235
## sample estimates:
## cor
## 0.9860831

#import dataset with L & R hind tibia lengths
legs <- read.csv("~/Documents/LMU/Eye morphology/selection manuscript/legs.csv")
cor.test(legs$L_length,legs$R_length, method="pearson")

##
## Pearson's product-moment correlation
##
## data: legs$L_length and legs$R_length
## t = 25.475, df = 101, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.8984434 0.9523152
## sample estimates:
## cor
## 0.9302284

####Correlations between body size, facet count, corneal area####
#without hybrids, onle pure species
cor.test(eye2$tibia_length,eye2$whole_count, method="pearson")

##
## Pearson's product-moment correlation
##
## data: eye2$tibia_length and eye2$whole_count
## t = 6.7425, df = 65, p-value = 4.934e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.4744268 0.7640020
## sample estimates:
## cor
## 0.6415262

cor.test(eye2$area,eye2$whole_count, method="pearson")

##
## Pearson's product-moment correlation

```

```
##
## data: eye2$area and eye2$whole_count
## t = 14.361, df = 65, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.7991763 0.9195831
## sample estimates:
##      cor
## 0.8719898

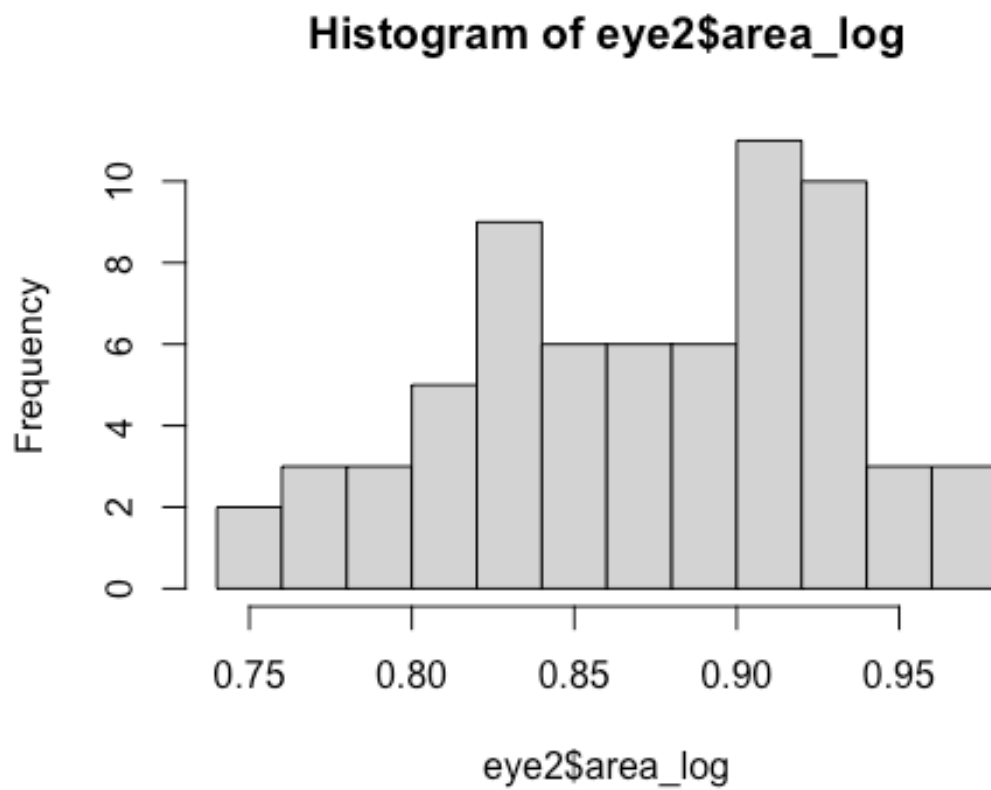
cor.test(eye2$tibia_length, eye2$area, method="pearson")

##
## Pearson's product-moment correlation
##
## data: eye2$tibia_length and eye2$area
## t = 13.167, df = 65, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.7704437 0.9071889
## sample estimates:
##      cor
## 0.8528228

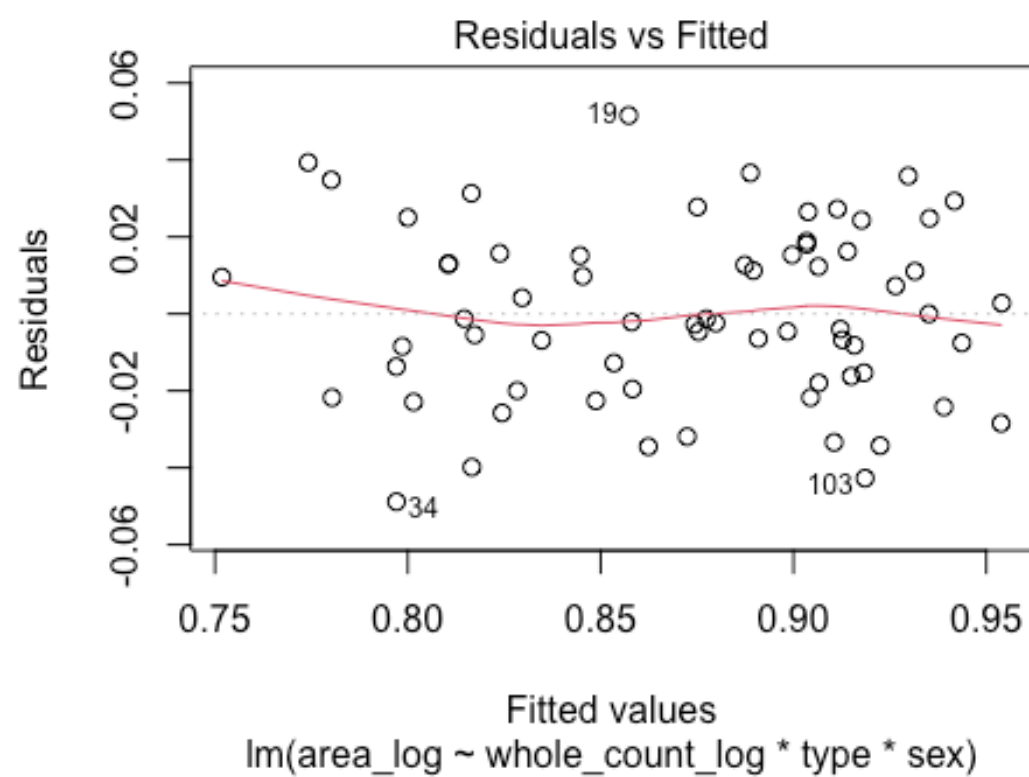
####WHAT IS THE RELATIONSHIP BETWEEN FACET COUNT AND CORNEAL AREA?####
#pure species only

hist(eye2$area_log)
```

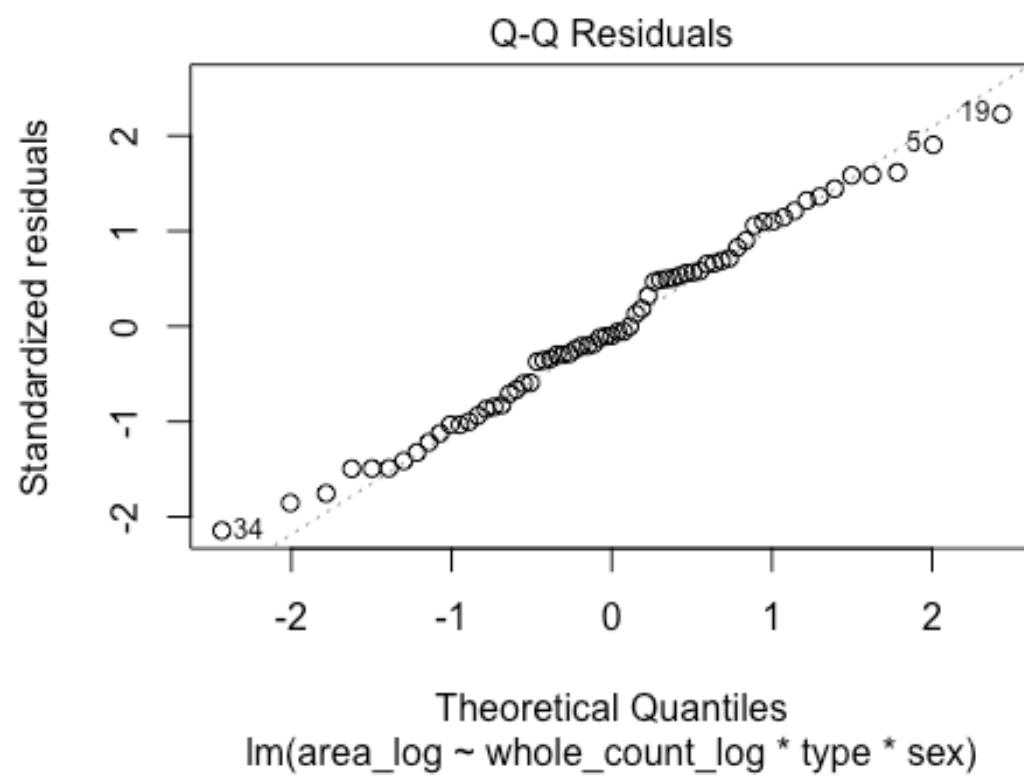




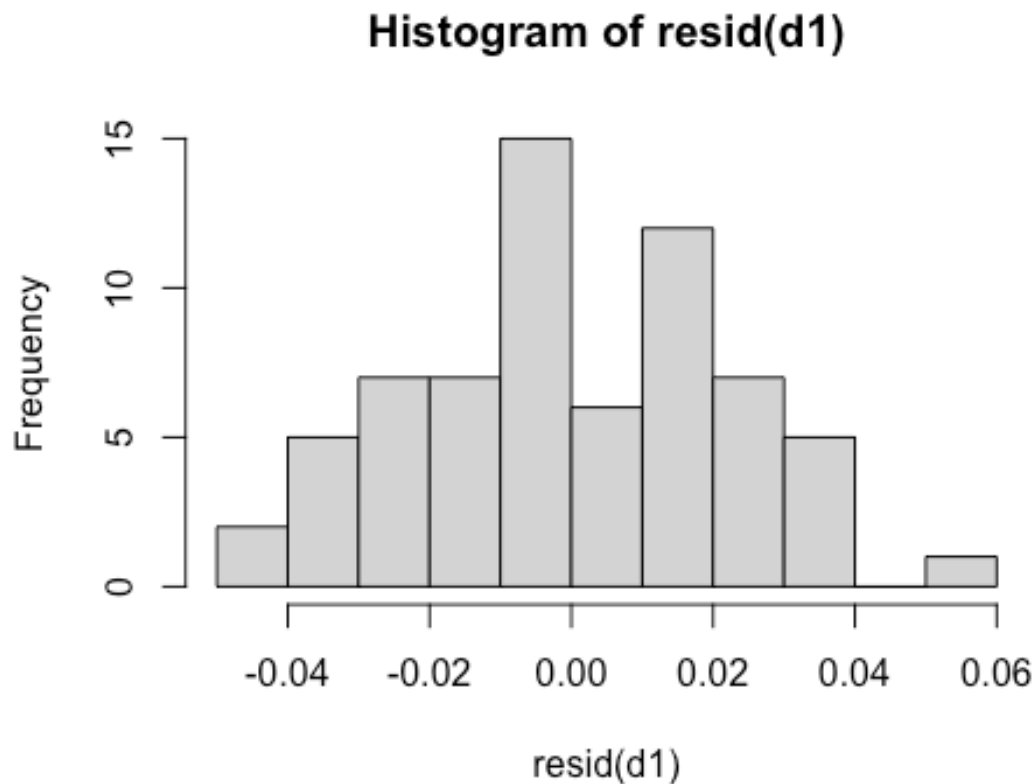
```
d1 <- lm(area_log~whole_count_log*type*sex, data=eye2)
plot(d1, which=1)
```



```
plot(d1, which=2)
```



```
hist(resid(d1))
```



```
drop1(d1, test="Chisq") #3-way interaction 0.098, remove

## Single term deletions
##
## Model:
## area_log ~ whole_count_log * type * sex
##               Df Sum of Sq    RSS    AIC Pr(>Chi)
## <none>                        0.033481 -493.30
## whole_count_log:type:sex  1 0.0013944 0.034875 -492.57 0.09824 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

d2 <- update(d1, ~.-whole_count_log:type:sex)
drop1(d2, test="Chisq") #all 2-way interactions ns

## Single term deletions
##
## Model:
## area_log ~ whole_count_log + type + sex + whole_count_log:type +
##             whole_count_log:sex + type:sex
##               Df Sum of Sq    RSS    AIC Pr(>Chi)
## <none>                        0.034875 -492.57
## whole_count_log:type  1 0.00089661 0.035772 -492.86 0.1922
```

```

## whole_count_log:sex    1 0.00023606 0.035111 -494.11    0.5014
## type:sex              1 0.00000165 0.034877 -494.56    0.9551

d3 <- update(d2, ~.-type:sex)
drop1(d3, test="Chisq")

## Single term deletions
##
## Model:
## area_log ~ whole_count_log + type + sex + whole_count_log:type +
##   whole_count_log:sex
##           Df Sum of Sq      RSS      AIC Pr(>Chi)
## <none>                                0.034877 -494.56
## whole_count_log:type  1 0.00125965 0.036136 -494.18    0.1231
## whole_count_log:sex   1 0.00041413 0.035291 -495.77    0.3738

d4 <- update(d3, ~.-whole_count_log:sex)
drop1(d4, test="Chisq")

## Single term deletions
##
## Model:
## area_log ~ whole_count_log + type + sex + whole_count_log:type
##           Df Sum of Sq      RSS      AIC Pr(>Chi)
## <none>                                0.035291 -495.77
## sex      1 0.00072766 0.036019 -496.40    0.2423
## whole_count_log:type  1 0.00141206 0.036703 -495.14    0.1050

d5 <- update(d4, ~.-whole_count_log:type)
drop1(d5, test="Chisq") #sex ns

## Single term deletions
##
## Model:
## area_log ~ whole_count_log + type + sex
##           Df Sum of Sq      RSS      AIC  Pr(>Chi)
## <none>                                0.036703 -495.14
## whole_count_log  1  0.060948 0.097651 -431.58 5.631e-16 ***
## type            1  0.007038 0.043741 -485.39 0.0006073 ***
## sex            1  0.000860 0.037563 -495.59 0.2128677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

d6 <- update(d5, ~.-sex)
drop1(d6, test="Chisq")

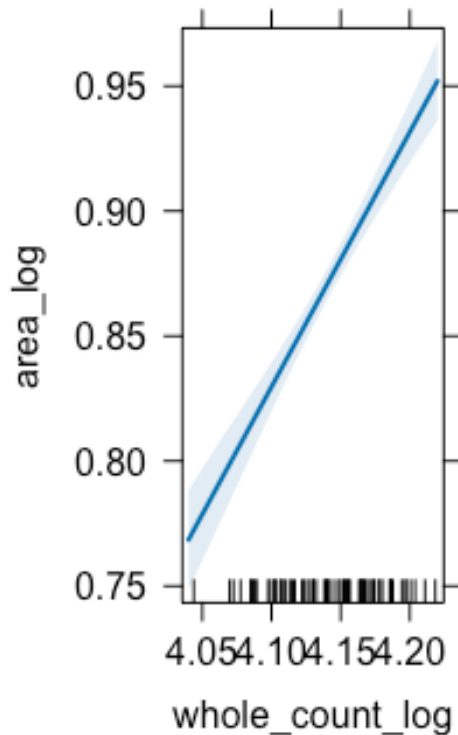
## Single term deletions
##
## Model:
## area_log ~ whole_count_log + type
##           Df Sum of Sq      RSS      AIC  Pr(>Chi)
## <none>                                0.037563 -495.59

```

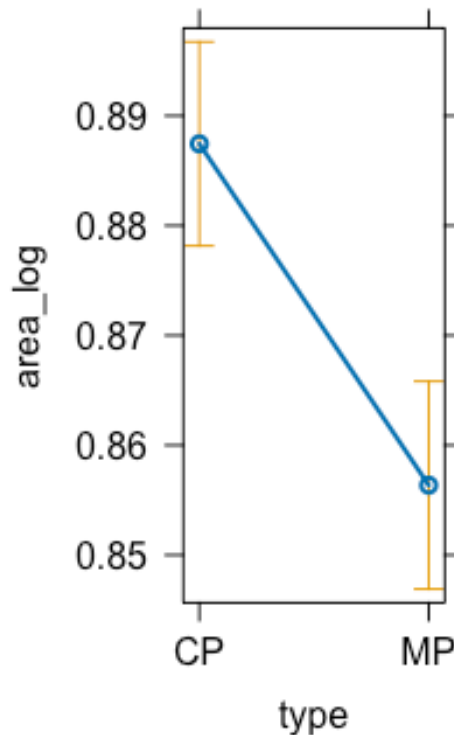
```
## whole_count_log  1  0.070850 0.108413 -426.58 < 2.2e-16 ***
## type             1  0.010692 0.048255 -480.81 4.194e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

plot(allEffects(d6))
```

**whole\_count\_log effect plot**



**type effect plot**



```
Anova(d6, test="Chisq")

## Anova Table (Type II tests)
##
## Response: area_log
##              Sum Sq Df F value    Pr(>F)
## whole_count_log 0.070850  1 120.715 2.269e-16 ***
## type            0.010692  1  18.216 6.650e-05 ***
## Residuals      0.037563 64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#facet count p<0.001
#type p<0.001

Anova(d4, test="Chisq")
```

```
## Anova Table (Type II tests)
##
## Response: area_log
##
##               Sum Sq Df  F value    Pr(>F)
## whole_count_log  0.060948  1 107.0742 3.918e-15 ***
## type             0.007038  1  12.3644 0.0008246 ***
## sex              0.000728  1   1.2784 0.2625574
## whole_count_log:type 0.001412  1   2.4807 0.1203373
## Residuals        0.035291 62
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

*#facet count:species n.s. p=0.12*

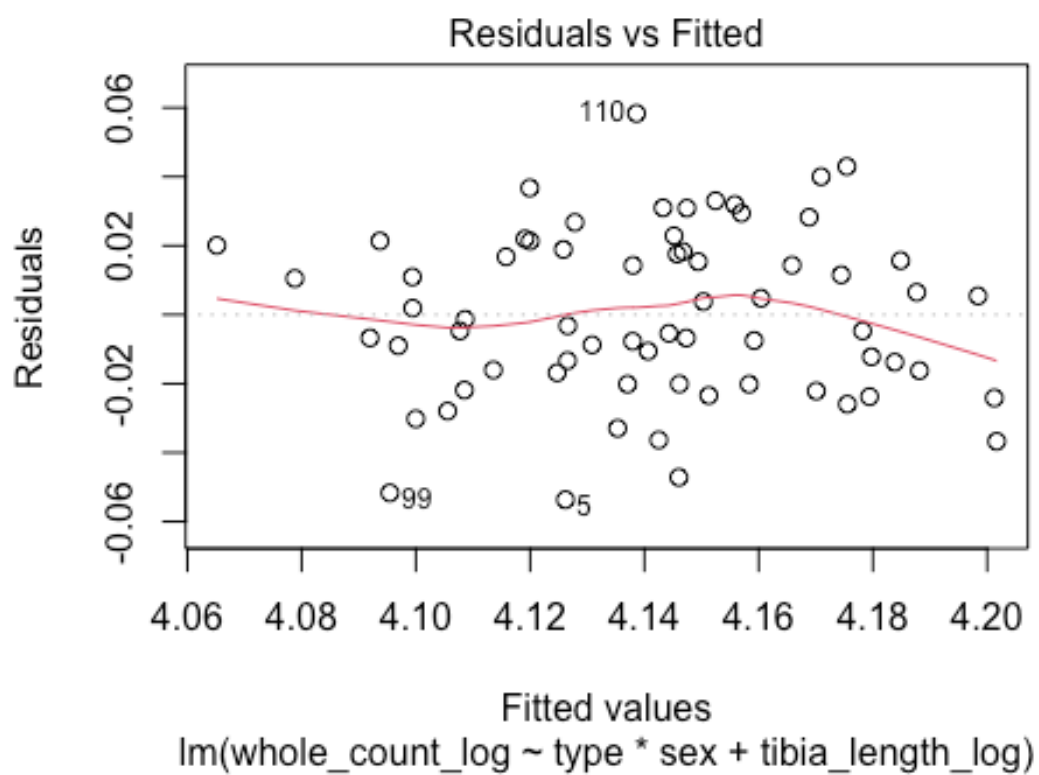
**####ANALYSES OF FACET COUNT - C & M only####**

*#full statistical model with only C & M*

```
hist(eye2$whole_count_log)
```

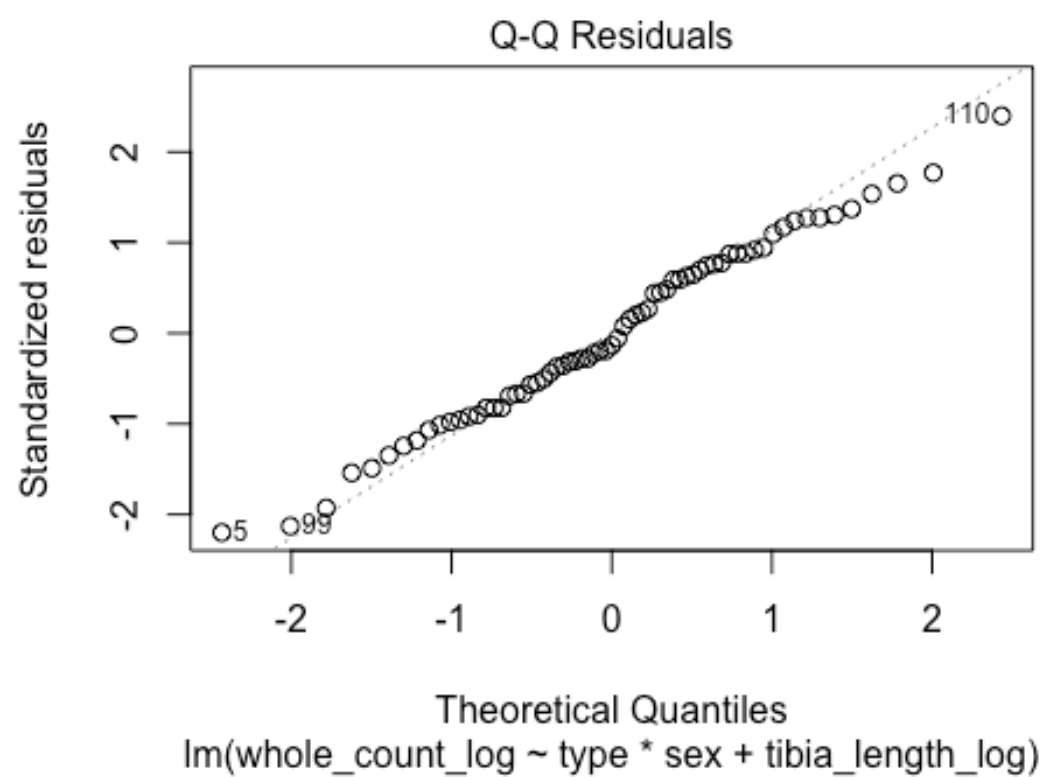


```
m1 <- lm(whole_count_log~type*sex+tibia_length_log, data=eye2)
plot(m1, which=1)
```

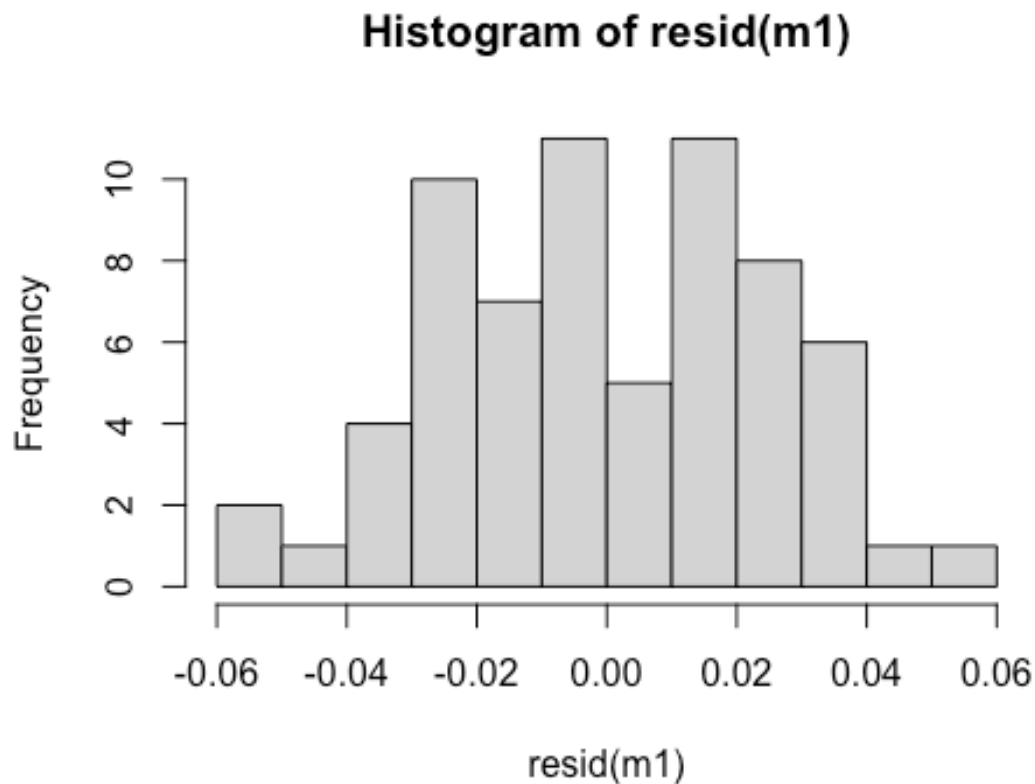


```
plot(m1, which=2)
```





```
hist(resid(m1))
```



```
drop1(m1, test="Chisq") #type:sex interaction ns

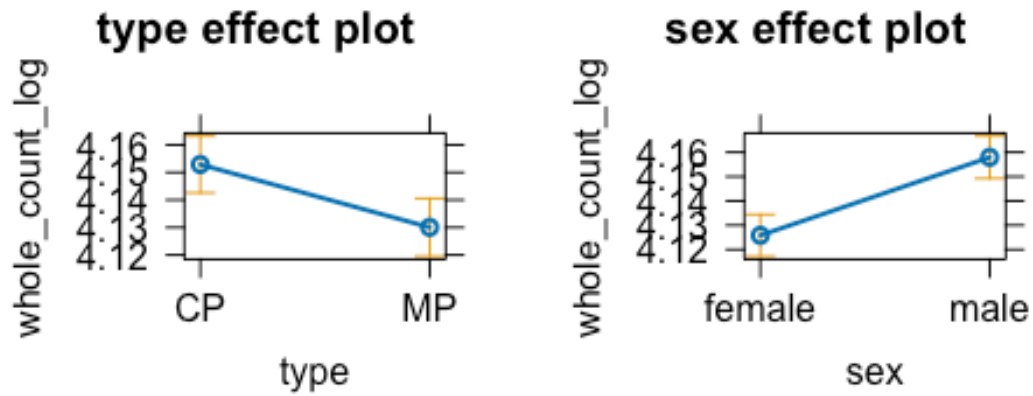
## Single term deletions
##
## Model:
## whole_count_log ~ type * sex + tibia_length_log
##           Df Sum of Sq    RSS   AIC  Pr(>Chi)
## <none>                0.039226 -488.69
## tibia_length_log    1 0.0127784 0.052005 -471.79 1.382e-05 ***
## type:sex            1 0.0000828 0.039309 -490.55    0.707
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

m2 <- update(m1, .~.-type:sex)
drop1(m2, test="Chisq") #all three significant

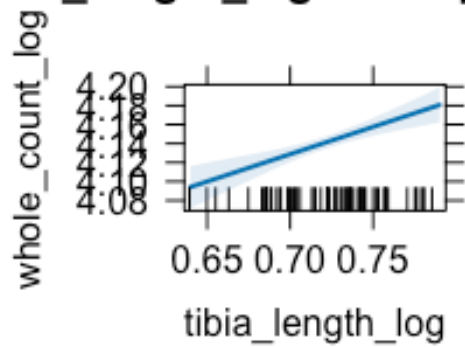
## Single term deletions
##
## Model:
## whole_count_log ~ type + sex + tibia_length_log
##           Df Sum of Sq    RSS   AIC  Pr(>Chi)
## <none>                0.039309 -490.55
## type            1 0.0045653 0.043874 -485.19 0.006663 **
## sex            1 0.0170625 0.056371 -468.39 8.892e-07 ***
```

```
## tibia_length_log 1 0.0127559 0.052065 -473.72 1.429e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

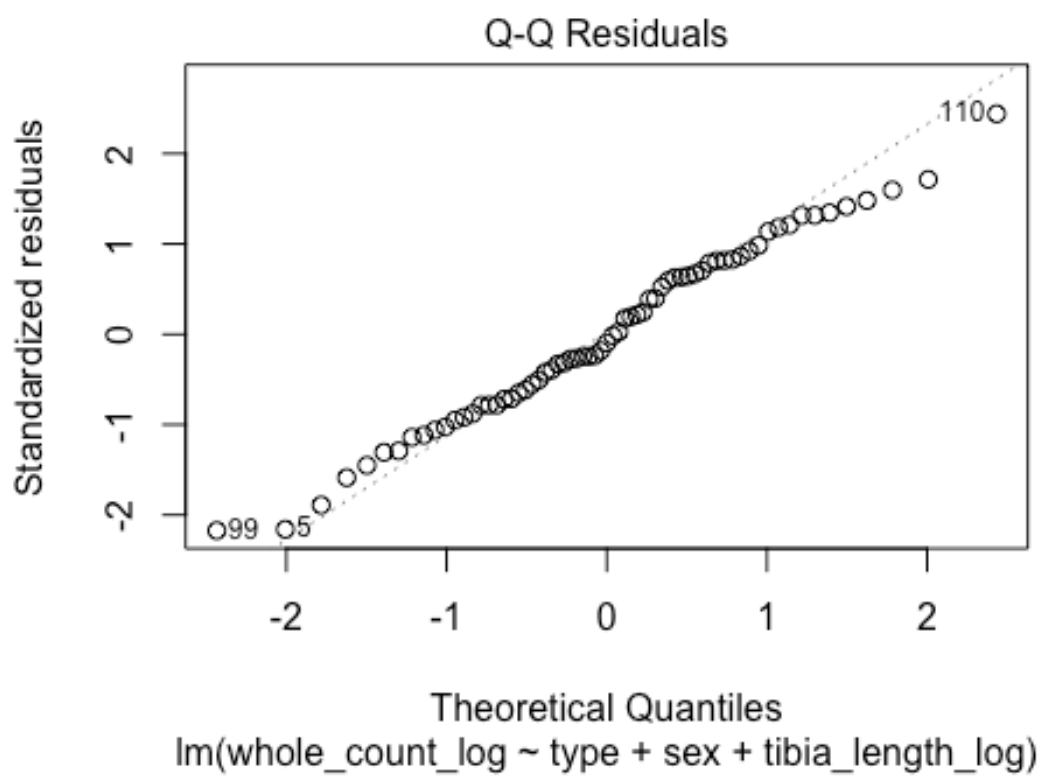
plot(allEffects(m2))
```



**tibia\_length\_log effect plot**

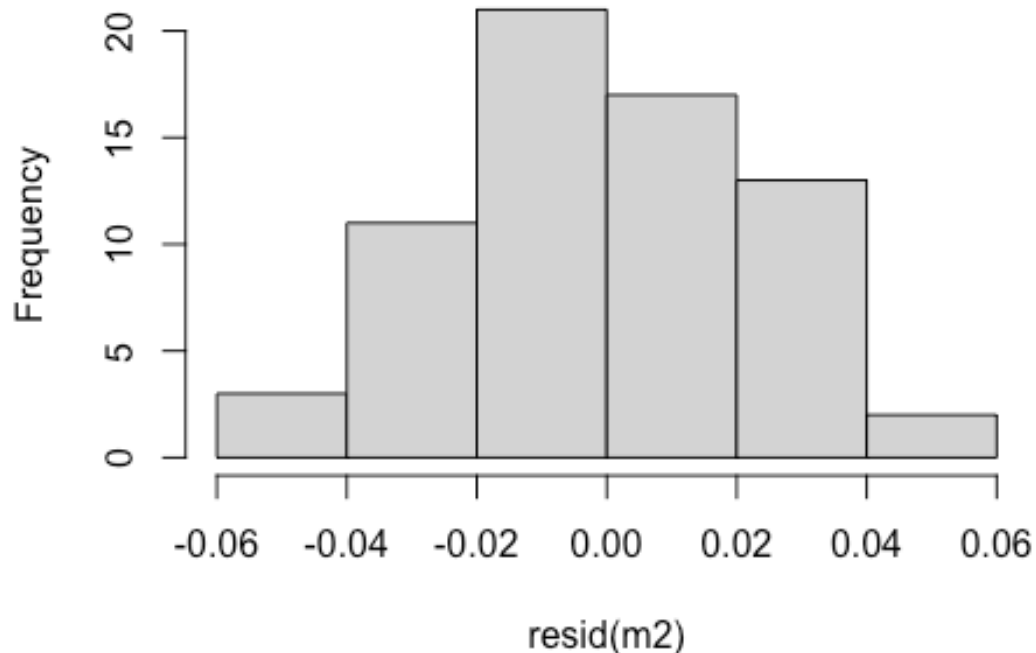


```
plot(m2, which = 2)
```



```
hist(resid(m2))
```

**Histogram of resid(m2)**



```
Anova(m2, test.statistic="Chisq")
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: whole_count_log
```

```
##
```

	Sum Sq	Df	F value	Pr(>F)	
--	--------	----	---------	--------	--

## type	0.004565	1	7.3168	0.008775	**
---------	----------	---	--------	----------	----

## sex	0.017063	1	27.3460	2.060e-06	***
--------	----------	---	---------	-----------	-----

## tibia_length_log	0.012756	1	20.4438	2.771e-05	***
---------------------	----------	---	---------	-----------	-----

## Residuals	0.039309	63			
--------------	----------	----	--	--	--

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#type p=0.0087
```

```
#sex p>0.0001
```

```
#tibia p>0.0001
```

```
Anova(m1, test.statistic="Chisq")
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: whole_count_log
```

```
##
```

	Sum Sq	Df	F value	Pr(>F)	
--	--------	----	---------	--------	--

## type	0.004565	1	7.2159	0.009262	**
---------	----------	---	--------	----------	----

```

## sex                0.017063  1 26.9687 2.435e-06 ***
## tibia_length_log 0.012778  1 20.1973 3.115e-05 ***
## type:sex          0.000083  1  0.1309  0.718760
## Residuals         0.039226 62
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#type:sex n.s. p=0.78

m2.1 <- update(m2, ~.-type)
PBmodcomp(m2, m2.1, nsim = 1000) #same result with parametric bootstrapping as
with Anova

## Bootstrap test; time: 2.51 sec; samples: 1000; extremes: 5;
## large : whole_count_log ~ type + sex + tibia_length_log
## whole_count_log ~ sex + tibia_length_log
##          stat df  p.value
## LRT      7.3617  1 0.006663 **
## PBtest 7.3617    0.005994 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#####plot of facet count based on MAM####
#using tibia length

#theme settings for plots
require(grid)
pub<- theme_update(
  panel.grid.major=element_line(colour=NA),
  panel.grid.minor=element_line(colour=NA),
  panel.background = element_rect(colour = NA, fill=NA, linewidth = 0.5),
  panel.border = element_rect(linetype = "solid", colour = "black", fill=NA),
  axis.line.x = element_line(color="black"),
  axis.line.y = element_line(color="black"),
  axis.title.x=element_text(size=15, face="bold", hjust=0.5, vjust=0.5, angle=0),
  axis.title.y=element_text(size=15, face="bold", hjust=0.5, vjust=1, angle=90),
  axis.text.x=element_text(colour="black", angle=0, size=15),
  axis.text.y=element_text(colour="black", angle=0, size=15),
  axis.ticks=element_line(colour="black", linewidth=0.5))

#get estimated marginal means of the final model (m2)
#####**species**#####

library(tidyverse)

## — Attaching core tidyverse packages ————— tidyverse 2.
0.0 —
## ✓ forcats 1.0.0      ✓ stringr 1.5.0
## ✓ lubridate 1.9.2    ✓ tibble 3.2.1

```

```

## ✓ purrr      1.0.1      ✓ tidyr      1.3.0
## ✓ readr      2.1.4

## — Conflicts ————— tidyverse_conflict
s() —
## ✗ dplyr::combine() masks gridExtra::combine()
## ✗ tidyr::expand() masks Matrix::expand()
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag() masks stats::lag()
## ✗ tidyr::pack() masks Matrix::pack()
## ✗ dplyr::recode() masks car::recode()
## ✗ dplyr::select() masks MASS::select()
## ✗ purrr::some() masks car::some()
## ✗ tidyr::unpack() masks Matrix::unpack()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(emmeans)
library(multcomp)

## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
##
## Attaching package: 'TH.data'
##
## The following object is masked from 'package:MASS':
##
##     geyser

library(multcompView)

model_means <- emmeans(object = m2, specs = ~ type, type = "response")

#p-value comparison table
pwpm(model_means, adjust="bonferroni", diffs = F)

##           CP           MP
## CP [4.153]  0.0088
## MP           [4.130]
##
## Row and column labels: type
## Upper triangle: P values
## Diagonal: [Estimates] (emmean) type = "response"

#effect size
eff_size(model_means, sigma = sigma(m2), edf = df.residual(m2))

## contrast estimate      SE df lower.CL upper.CL
## CP - MP           0.913 0.347 63      0.219      1.61

```

```
##
## Results are averaged over the levels of: sex
## sigma used for effect sizes: 0.02498
## Confidence level used: 0.95

#add letters to each mean to indicate difference
model_means_cld <- cld(object = model_means,
                        reversed=T,
                        adjust="bonferroni",
                        Letters = letters,
                        alpha = 0.05)

#set marginal means at a dataframe
facet <- as.data.frame(summary(model_means_cld))

#species

#color selection for consistency with other papers
library(RColorBrewer)
display.brewer.pal(12, 'Paired')
```



Paired (qualitative)

```
brewer.pal(12, 'Paired')
```



```

## [1] "#A6CEE3" "#1F78B4" "#B2DF8A" "#33A02C" "#FB9A99" "#E31A1C" "#FDBF6F"
## [8] "#FF7F00" "#CAB2D6" "#6A3D9A" "#FFFF99" "#B15928"

facet$type <- factor(facet$type, c("CP", "MP"))

species <- ggplot(facet, aes(type, emmean)) +
  geom_errorbar(data=facet, mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL),
    color="grey45",
    width=0, linewidth=1, position = position_dodge(0.4)) +
  geom_point(data=facet, aes(type, emmean, color=type), size=4,
    position = position_dodge(0.4)) +
  geom_text(aes(label = .group, y = emmean), position = position_dodge(0.4),
    vjust=0.3, hjust=-0.2, size=4.5) +
  labs(x="", y="e.m.m. log10 (facet count)") +
  scale_y_continuous(breaks=seq(4.11, 4.17, 0.02), limits=c(4.10, 4.174)) +
  scale_x_discrete(labels=c("", "", "")) +
  scale_color_manual(values = c("#1F78B4", "#FDBF6F")) +
  theme_update(axis.title.y=element_text(size=14, face="bold", hjust=0.5, vjust=
2.5, angle=90)) +
  theme(plot.margin = margin(0.4, 0.5, 0.3, 0.75, "cm")) +
  theme(legend.position="none", legend.key=element_blank(), legend.title=element_blank(),
    legend.text=element_text(size=10)) +
  theme(axis.text.x=element_blank()) +
  theme(text = element_text(family = "Times New Roman")) +
  theme(axis.title.y = element_text(vjust = 5)) +
  ggtitle("species effect") + theme(plot.title = element_text(hjust = 0.5, size=16, face="bold"))

#####**sex**#####
model_means2 <- emmeans(object = m2, specs = ~ sex, type = "response")

#p-value comparison table
pwpm(model_means2, adjust="bonferroni", diffs = F)

##          female      male
## female [4.126] <.0001
## male      [4.158]
##
## Row and column labels: sex
## Upper triangle: P values
## Diagonal: [Estimates] (emmean)   type = "response"

#effect size
eff_size(model_means2, sigma = sigma(m2), edf = df.residual(m2))

## contrast      estimate    SE df lower.CL upper.CL
## female - male    -1.29 0.271 63    -1.83   -0.744
##

```

```

## Results are averaged over the levels of: type
## sigma used for effect sizes: 0.02498
## Confidence level used: 0.95

#add letters to each mean to indicate difference
model_means_cld2 <- cld(object = model_means2,
                        reversed = T,
                        adjust="bonferroni",
                        Letters = letters,
                        alpha = 0.05)

#set marginal means at a dataframe
facet2 <- as.data.frame(summary(model_means_cld2))

#plot marginal means
facet2$sex <- factor(facet2$sex, c("male", "female"))

#sex
sex <- ggplot(facet2, aes(sex, emmean))+
  geom_errorbar(data=facet2, mapping=aes(x=sex, ymin=upper.CL, ymax=lower.CL),
               color="grey45",
               width=0, linewidth=1)+
  geom_point(data=facet2, aes(sex, emmean), size=4)+
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3, size=4.5) +
  labs(x="", y="")+
  scale_y_continuous(breaks=seq(4.11, 4.17, 0.02), limits=c(4.10, 4.174),
                    position="right")+
  scale_x_discrete(labels = c("", ""))+
  theme_update(axis.title.y=element_text(size=14, face="bold", hjust=0.5, vjust=
2.5, angle=90))+
  theme(legend.position=c(0.88, 0.89), legend.key=element_blank(), legend.title
=element_blank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(0.4, 1.05, 0.3, 0.85, "cm"))+
  theme(axis.text.x=element_blank())+
  theme(text = element_text(family = "Times New Roman"))+
  ggtitle("sex effect") + theme(plot.title = element_text(hjust = 0.5, size=1
6, face="bold"))

#####**species:sex**#####
model_means3 <- emmeans(object = m2, specs = ~ type:sex, type = "response")

#p-value comparison table
pwpm(model_means3, adjust="bonferroni", diffs = F)

##          CP female MP female CP male MP male
## CP female [4.137]  0.0527 <.0001 1.0000

```

```

## MP female          [4.114] <.0001 <.0001
## CP male            [4.169]  0.0527
## MP male            [4.146]
##
## Row and column labels: type:sex
## Upper triangle: P values  adjust = "bonferroni"
## Diagonal: [Estimates] (emmean)  type = "response"

#add letters to each mean to indicate difference
model_means_cld3 <- cld(object = model_means3,
                        adjust = "bonferroni",
                        reversed=T,
                        Letters = letters,
                        alpha = 0.05)

#set marginal means at a dataframe
facet3 <- as.data.frame(summary(model_means_cld3))

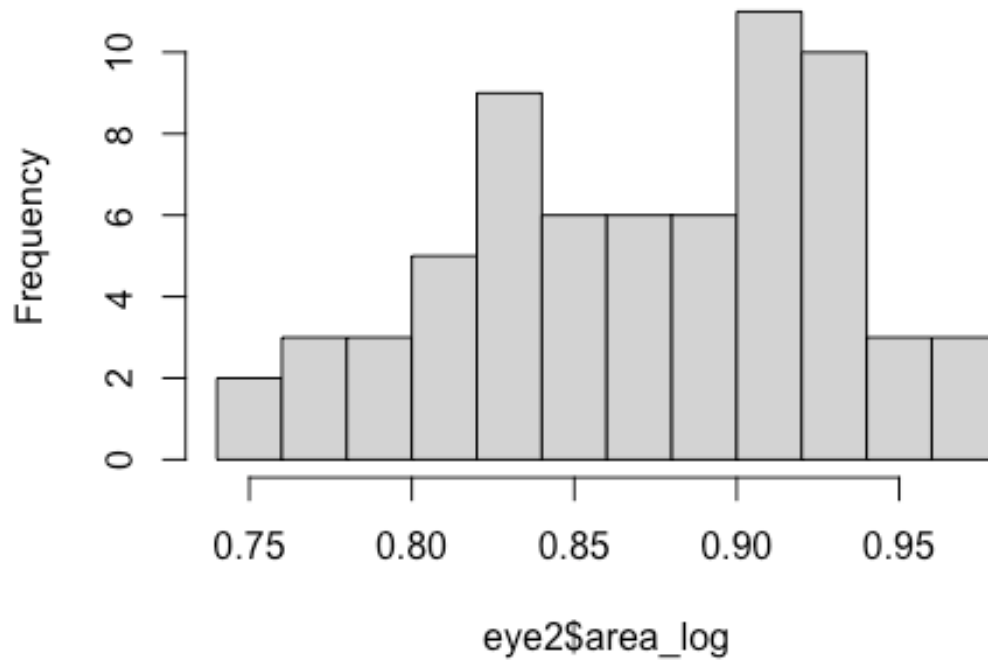
#plot marginal means
facet3$sex <- factor(facet3$sex, c("male", "female"))
facet3$type <- factor(facet3$type, c("CP", "MP"))

#species:sex plot
combo1<-ggplot(facet3, aes(type,emmean))+
  geom_errorbar(data=facet3,mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL,
                                         color=sex),
               width=0, linewidth=1, position = position_dodge(width=0.5))+
  geom_point(data=facet3, aes(type,emmean, color=sex), size=3,
             position = position_dodge2(width=0.5))+
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3,size=4.5, position = position_dodge2(width=
0.5)) +
  labs(x="",y="e.m.m. log10 (facet count)", color="")+
  scale_y_continuous(breaks=seq(4.09, 4.21, 0.03), limits=c(4.08, 4.21))+
  scale_x_discrete(labels=c("H. cydno", "H. melp"))+
  scale_color_manual(values=c("black","grey65"))+
  theme(legend.position="", legend.key=element_blank(),legend.title=element_b
lank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(1,0.5,0.5,0.5, "cm"))+
  theme(legend.text=element_text(size=11))+
  theme(legend.key=element_blank())+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))

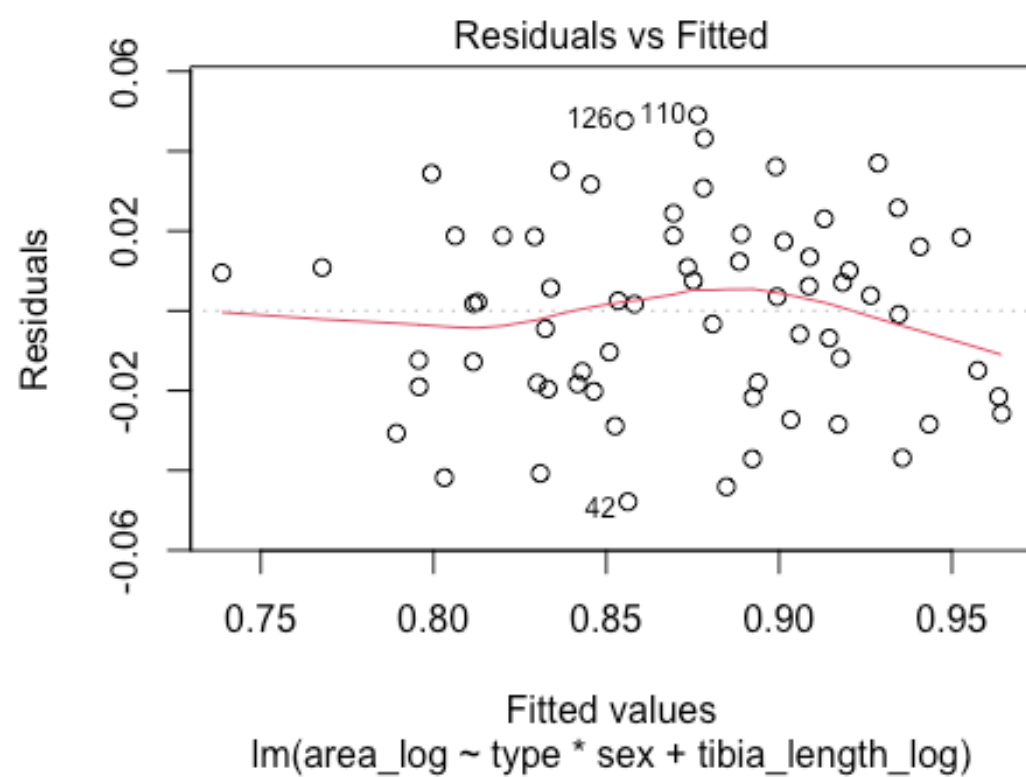
####ANALYSES OF CORNEAL AREA - C & M ONLY####
hist(eye2$area_log)

```

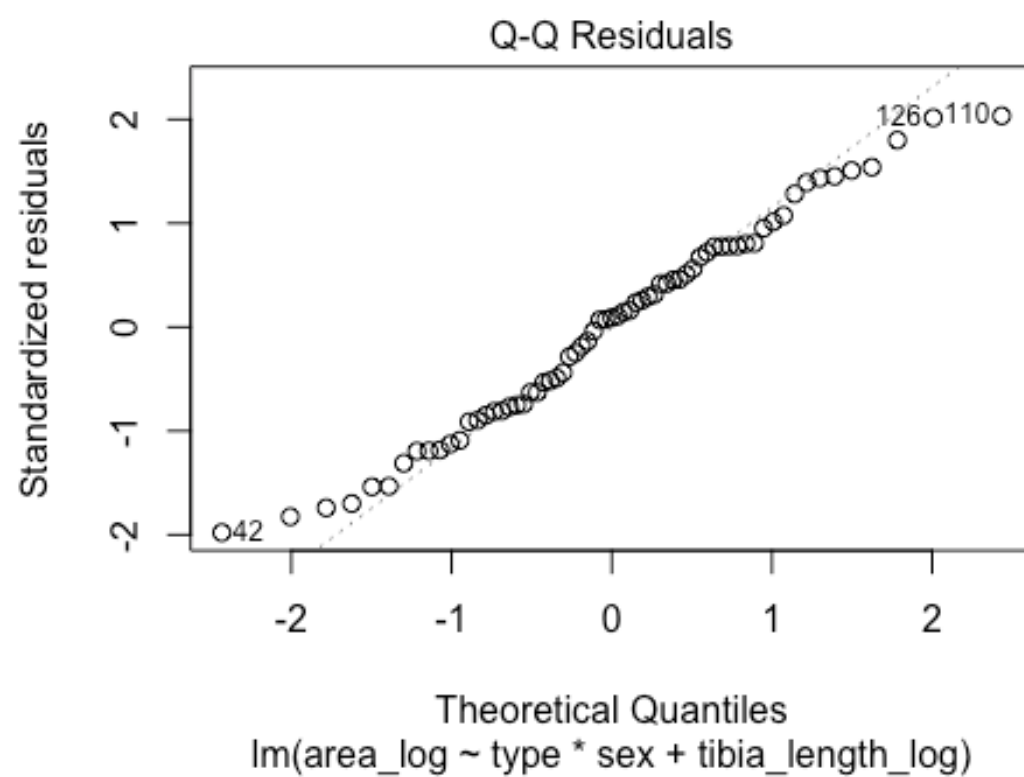
**Histogram of eye2\$area\_log**



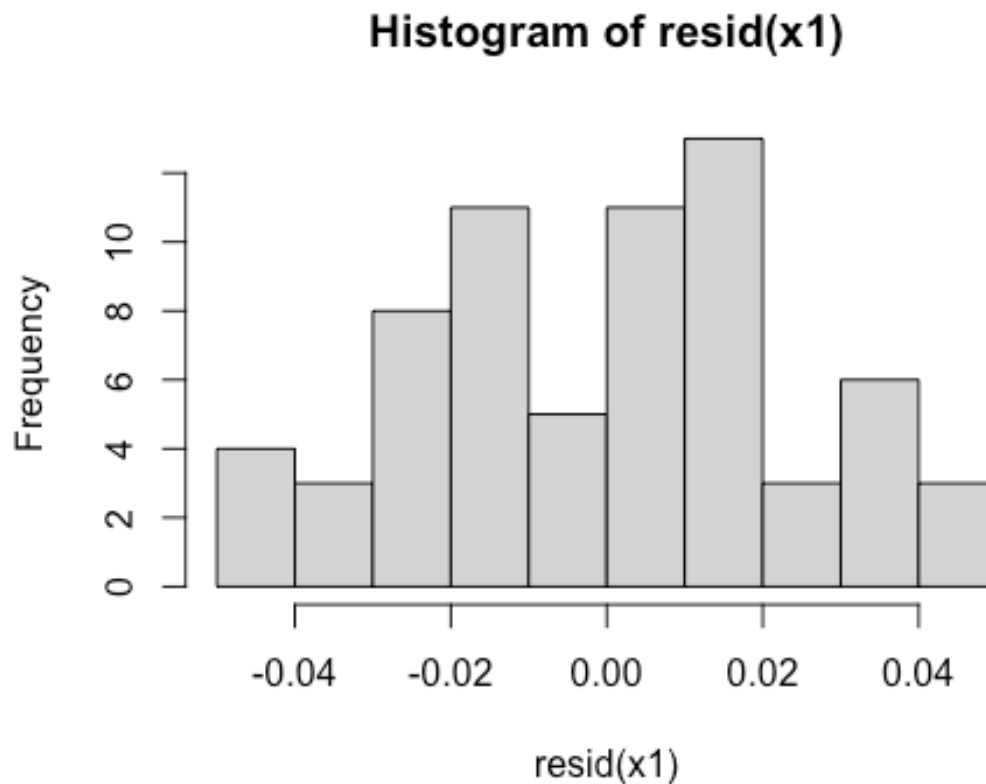
```
x1 <- lm(area_log~type*sex+tibia_length_log, data=eye2)
plot(x1,which=1)
```



```
plot(x1,which=2)
```



```
hist(resid(x1))
```



```
drop1(x1, test="Chisq") #interaction weakly trending (0.08)

## Single term deletions
##
## Model:
## area_log ~ type * sex + tibia_length_log
##           Df Sum of Sq    RSS   AIC  Pr(>Chi)
## <none>                 0.038397 -490.12
## tibia_length_log    1  0.057777 0.096174 -430.60 4.388e-15 ***
## type:sex            1  0.001702 0.040100 -489.21  0.08824 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

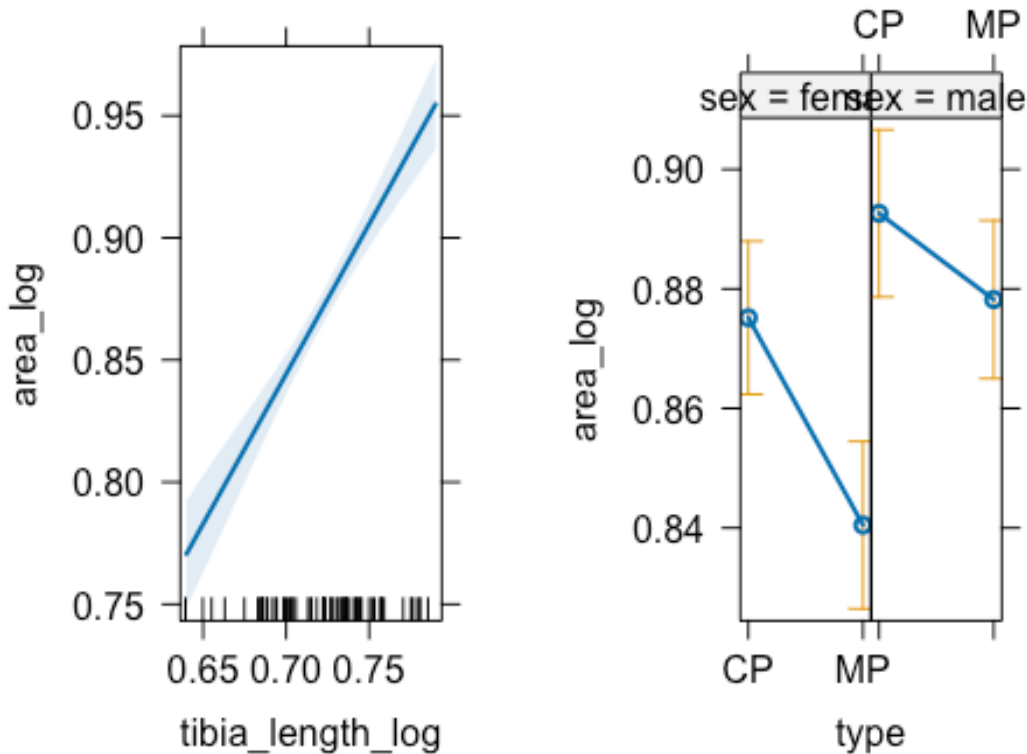
Anova(x1, test="Chisq") #interaction n.s. p=0.102, remove

## Anova Table (Type II tests)
##
## Response: area_log
##           Sum Sq Df F value    Pr(>F)
## type           0.005391  1  8.7050  0.004473 **
## sex             0.012487  1 20.1622 3.159e-05 ***
## tibia_length_log 0.057777  1 93.2916 5.608e-14 ***
## type:sex        0.001702  1  2.7485  0.102394
## Residuals      0.038397 62
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

plot(allEffects(x1))
```

tibia\_length\_log effect plot      type\*sex effect plot

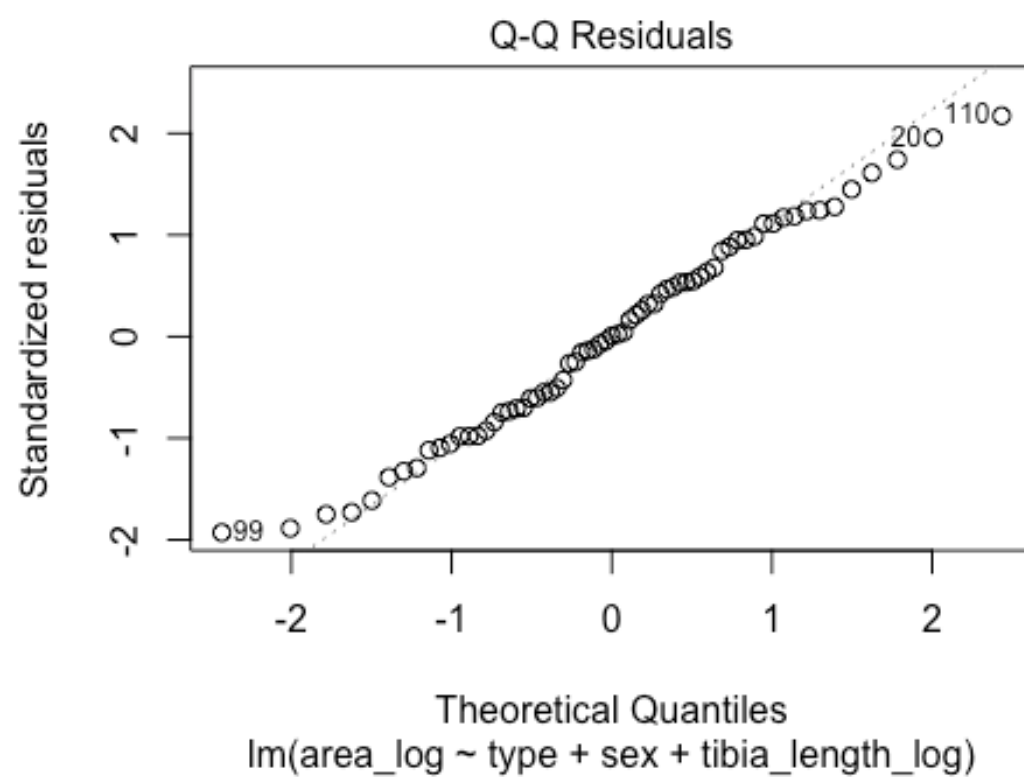


```
x2 <- update(x1, .~.-type:sex)
drop1(x2, test="Chisq")

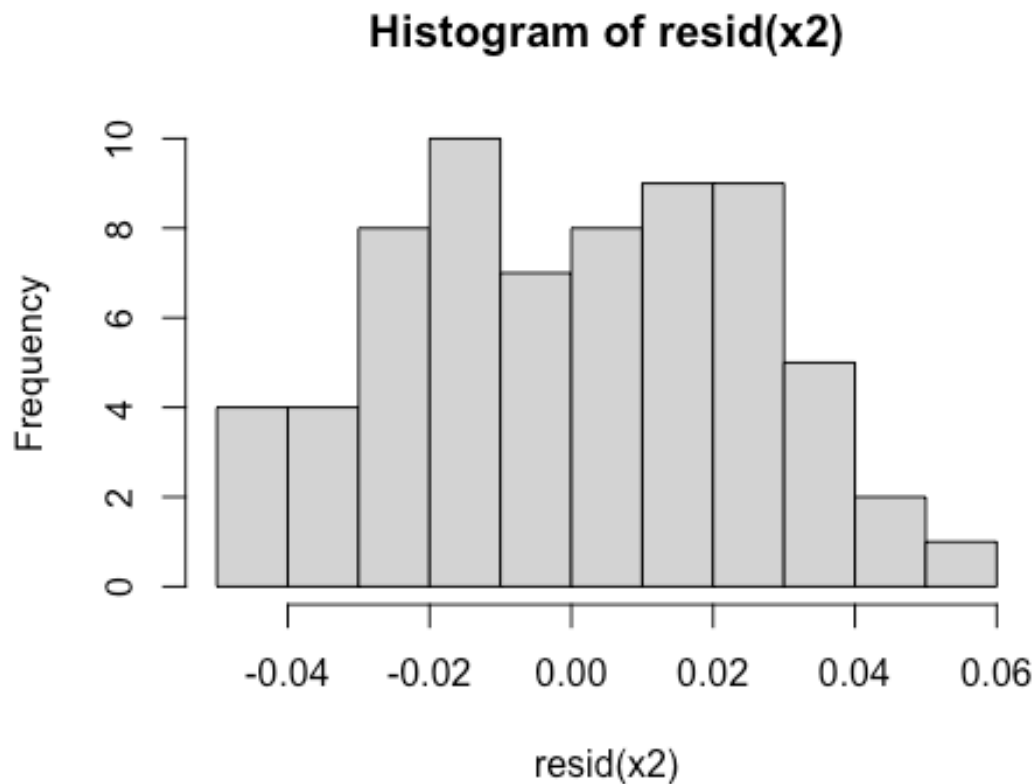
## Single term deletions
##
## Model:
## area_log ~ type + sex + tibia_length_log
##
##           Df Sum of Sq    RSS   AIC  Pr(>Chi)
## <none>                 0.040100 -489.21
## type           1  0.005391 0.045491 -482.76  0.003647 **
## sex            1  0.012487 0.052586 -473.05 2.028e-05 ***
## tibia_length_log 1  0.057551 0.097651 -431.58 1.144e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

plot(x2, which=2)
```





```
hist(resid(x2))
```



```
Anova(x2, test="Chisq")
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: area_log
```

```
##
```

	Sum Sq	Df	F value	Pr(>F)	
## type	0.005391	1	8.4699	0.004985	**
## sex	0.012487	1	19.6177	3.844e-05	***
## tibia_length_log	0.057551	1	90.4172	8.610e-14	***
## Residuals	0.040100	63			

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

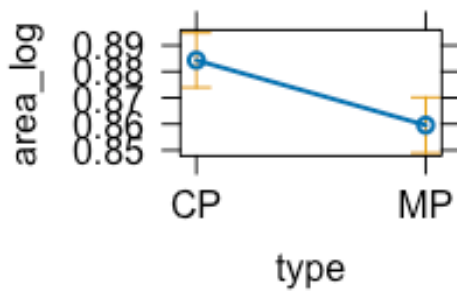
```
#type p=0.004
```

```
#sex p>0.001
```

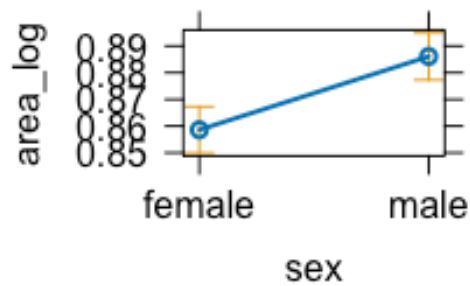
```
#tibia p>0.001
```

```
plot(allEffects(x2))
```

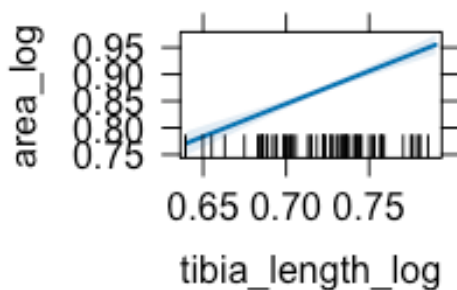
**type effect plot**



**sex effect plot**



**tibia\_length\_log effect plot**



#####plot of corneal area based on MAM####

#get estimated marginal means of the final model (x2)

#####\*\*species\*\*#####

```
model_means_x <- emmeans(object = x2, specs = ~ type, type = "response")
```

#p-value comparison table

```
pwpm(model_means_x, adjust="bonferroni", diffs = F)
```

```
##           CP           MP
```

```
## CP [0.885]  0.0050
```

```
## MP           [0.860]
```

```
##
```

```
## Row and column labels: type
```

```
## Upper triangle: P values
```

```
## Diagonal: [Estimates] (emmean)   type = "response"
```

#effect size

```
eff_size(model_means_x, sigma = sigma(x2), edf = df.residual(x2))
```

```

## contrast estimate      SE df lower.CL upper.CL
## CP - MP      0.983 0.349 63      0.286      1.68
##
## Results are averaged over the levels of: sex
## sigma used for effect sizes: 0.02523
## Confidence level used: 0.95

#add letters to each mean to indicate difference
model_means_cld_x <- cld(object = model_means_x,
                        reversed=T,
                        adjust="bonferroni",
                        Letters = letters,
                        alpha = 0.05)

#set marginal means at a dataframe
area <- as.data.frame(summary(model_means_cld_x))

#species

area$type <- factor(area$type, c("CP", "MP"))

species2 <- ggplot(area, aes(type,emmean))+
  geom_errorbar(data=area,mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL),
               color="grey45",
               width=0, linewidth=1, position = position_dodge(0.4))+
  geom_point(data=area, aes(type,emmean,color=type), size=4,
             position = position_dodge(0.4))+
  geom_text(aes(label = .group, y = emmean), position = position_dodge(0.4),
            vjust=0.3, hjust=-0.2, size=4.5) +
  labs(x="",y="e.m.m. log10 (corneal area)")+
  scale_y_continuous(breaks=seq(0.82, 0.92, 0.03), limits=c(0.81, 0.92))+
  scale_x_discrete(labels=c("H. cydno", "H. melp"))+
  scale_color_manual(values = c("#1F78B4", "#FDBF6F"))+
  theme_update(axis.title.y=element_text(size=14,face="bold",hjust=0.5,vjust=
2.5,angle=90))+
  theme(legend.position="none", legend.key=element_blank(),legend.title=element_blank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(0.3,0.5,0,0.75, "cm"))+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))+
  theme(axis.title.y = element_text(vjust = 5))+
  ggtitle("") + theme(plot.title = element_text(hjust = 0.5, size=16,face="bold"))

#####**sex**#####
model_means2_x <- emmeans(object = x2, specs = ~ sex, type = "response")

```

```

#p-value comparison table
pwpm(model_means2_x, adjust="bonferroni", diffs = F)

##           female      male
## female [0.858] <.0001
## male           [0.886]
##
## Row and column labels: sex
## Upper triangle: P values
## Diagonal: [Estimates] (emmean)   type = "response"

#effect size
eff_size(model_means2_x, sigma = sigma(x2), edf = df.residual(x2))

## contrast      estimate    SE df lower.CL upper.CL
## female - male    -1.09 0.264 63    -1.62   -0.561
##
## Results are averaged over the levels of: type
## sigma used for effect sizes: 0.02523
## Confidence level used: 0.95

#add letters to each mean to indicate difference
model_means_cld2_x <- cld(object = model_means2_x,
                          reversed = T,
                          adjust="bonferroni",
                          Letters = letters,
                          alpha = 0.05)

#set marginal means at a dataframe
area2 <- as.data.frame(summary(model_means_cld2_x))

#plot marginal means
area2$sex <- factor(area2$sex, c("male", "female"))

#sex
sex2<- ggplot(area2, aes(sex,emmean))+
  geom_errorbar(data=area2,mapping=aes(x=sex, ymin=upper.CL, ymax=lower.CL),
color="grey45",
              width=0, linewidth=1)+
  geom_point(data=area2, aes(sex,emmean), size=4)+
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3,size=4.5) +
  labs(x="",y="")+
  scale_y_continuous(breaks=seq(0.82, 0.92, 0.03), limits=c(0.81, 0.92),
                    position="right")+
  scale_x_discrete(labels = c("male", "female"))+
  theme_update(axis.title.y=element_text(size=14,face="bold",hjust=0.5,vjust=
2.5,angle=90))+
  theme(legend.position=c(0.88,0.89), legend.key=element_blank(),legend.title
=element_blank(),

```

```

    legend.text=element_text(size=10))+
  theme(plot.margin = margin(0.3,1.05,0,0.85, "cm"))+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))+
  ggtitle("") + theme(plot.title = element_text(hjust = 0.5, size=16,face="bold"))

```

#####species:sex#####

```
model_means3_x <- emmeans(object = x2, specs = ~ type:sex, type = "response")
```

*#p-value comparison table*

```
pwpm(model_means3_x, adjust="bonferroni", diffs = F)
```

```
##           CP female MP female CP male MP male
## CP female  [0.871]    0.0299  0.0002  1.0000
## MP female           [0.846]   <.0001  0.0002
## CP male                [0.898]  0.0299
## MP male                    [0.873]
##
## Row and column labels: type:sex
## Upper triangle: P values   adjust = "bonferroni"
## Diagonal: [Estimates] (emmean)   type = "response"
```

*#add letters to each mean to indicate difference*

```
model_means_cld3_x <- cld(object = model_means3_x,
                           adjust = "bonferroni",
                           reversed = T,
                           Letters = letters,
                           alpha = 0.05)
```

*#set marginal means at a dataframe*

```
area3 <- as.data.frame(summary(model_means_cld3_x))
```

*#plot marginal means*

```
area3$sex <- factor(area3$sex, c("male", "female"))
```

*#species:sex plot*

```
combo2<-ggplot(area3, aes(type,emmean))+
  geom_errorbar(data=area3,mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL,
                                       color=sex),
               width=0, linewidth=1, position = position_dodge(width=0.5))+
  geom_point(data=area3, aes(type,emmean, color=sex), size=3,
             position = position_dodge2(width=0.5))+
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3,size=4.5, position = position_dodge2(width=
0.5)) +
  labs(x="",y="e.m.m. log10 (corneal area)", color="")+
  scale_y_continuous(breaks=seq(0.83, 0.91, 0.02), limits=c(0.825, 0.915),
```

```

        position="left")+
scale_x_discrete(labels=c("H. cydno", "H. melp"))+
scale_color_manual(values=c("black","grey65"))+
theme(legend.position="", legend.key=element_blank(),legend.title=element_b
lank(),
      legend.text=element_text(size=10))+
theme(plot.margin = margin(1,0.5,0.5,0.5, "cm"))+
theme(legend.text=element_text(size=11))+
theme(legend.key=element_blank()+
      theme(axis.text.x=element_text(face = "bold"))+
      theme(text = element_text(family = "Times New Roman"))

```

#####MASTER PLOT OF ALL 4 EMMS - C & M ONLY#####

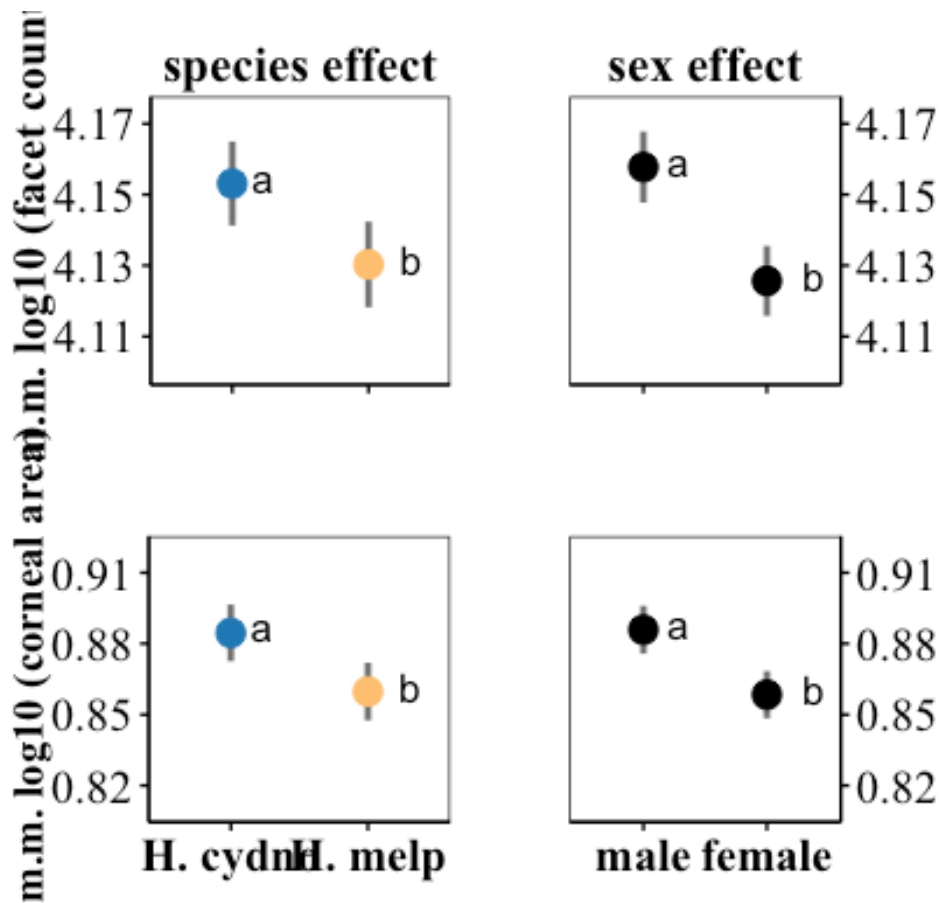
```
library(ggpubr)
```

#####\*\*species & sex separate\*\*#####

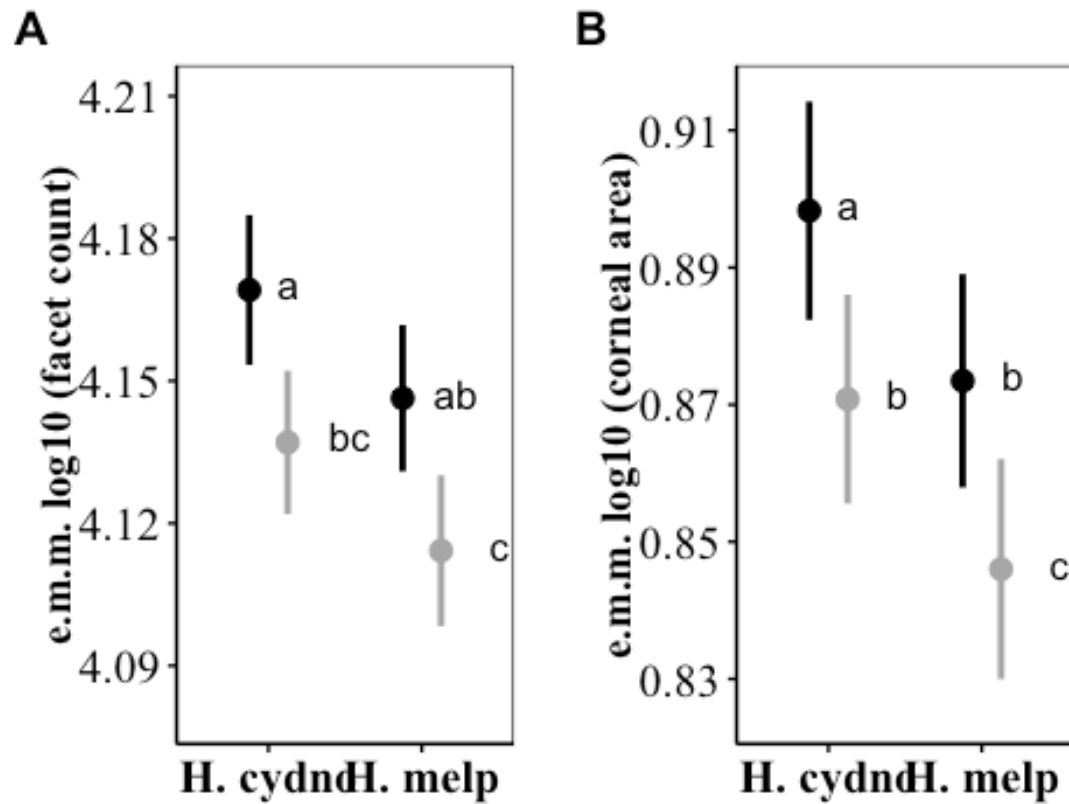
```

ggarrange(species, sex, species2, sex2,
          ncol = 2, nrow = 2, widths = c(1, 1.05),
          font.label = list(size = 15))

```



```
#####species:sex#####
ggarrange(combo1, combo2,
  labels = c("A", "B"),
  vjust=2,
  ncol = 2, nrow = 1, widths = c(1, 1),
  font.label = list(size = 15))
```



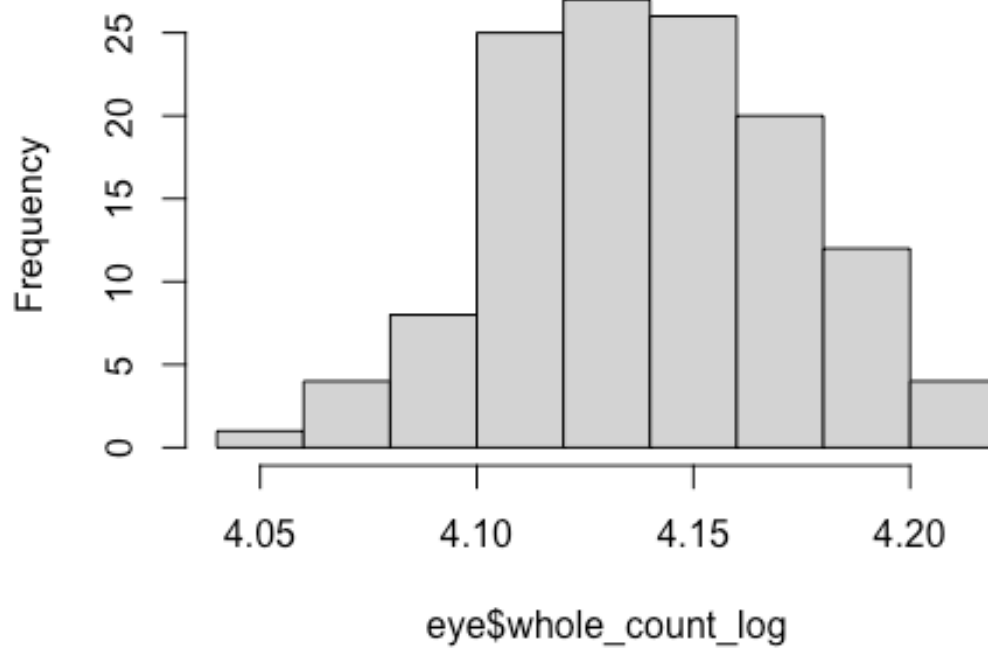
```
#####ANALYSES OF FACET COUNT - C, M & F1 HYBRIDS#####
```

```
#full statistical model
```

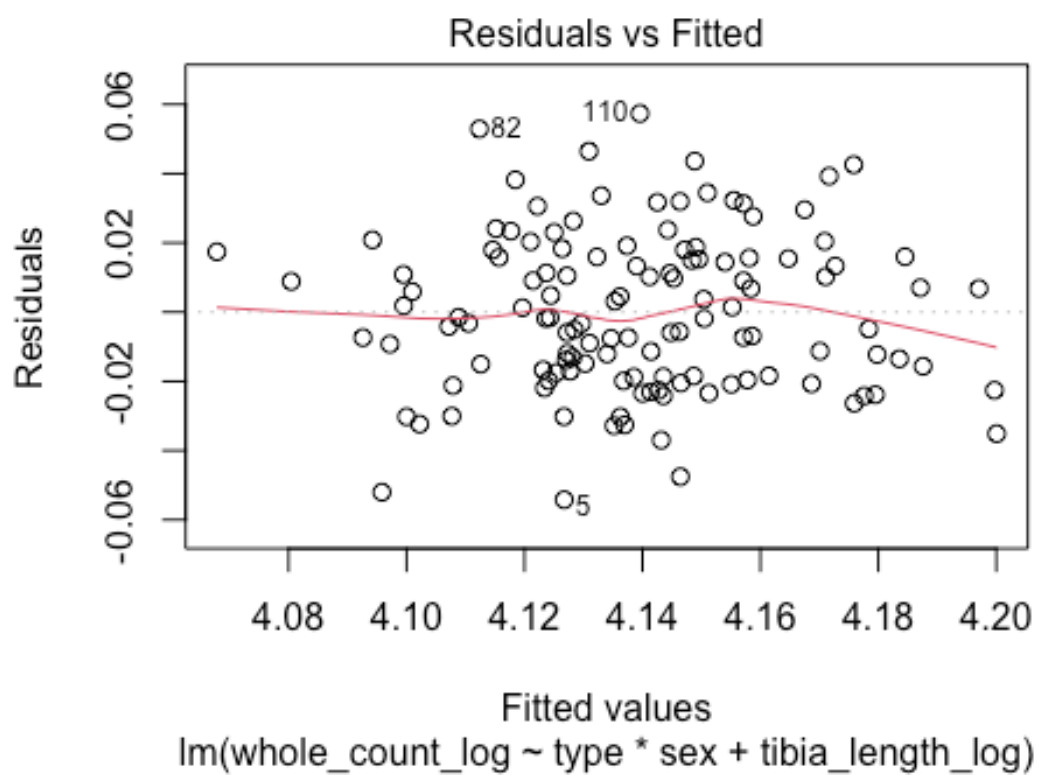
```
hist(eye$whole_count_log)
```



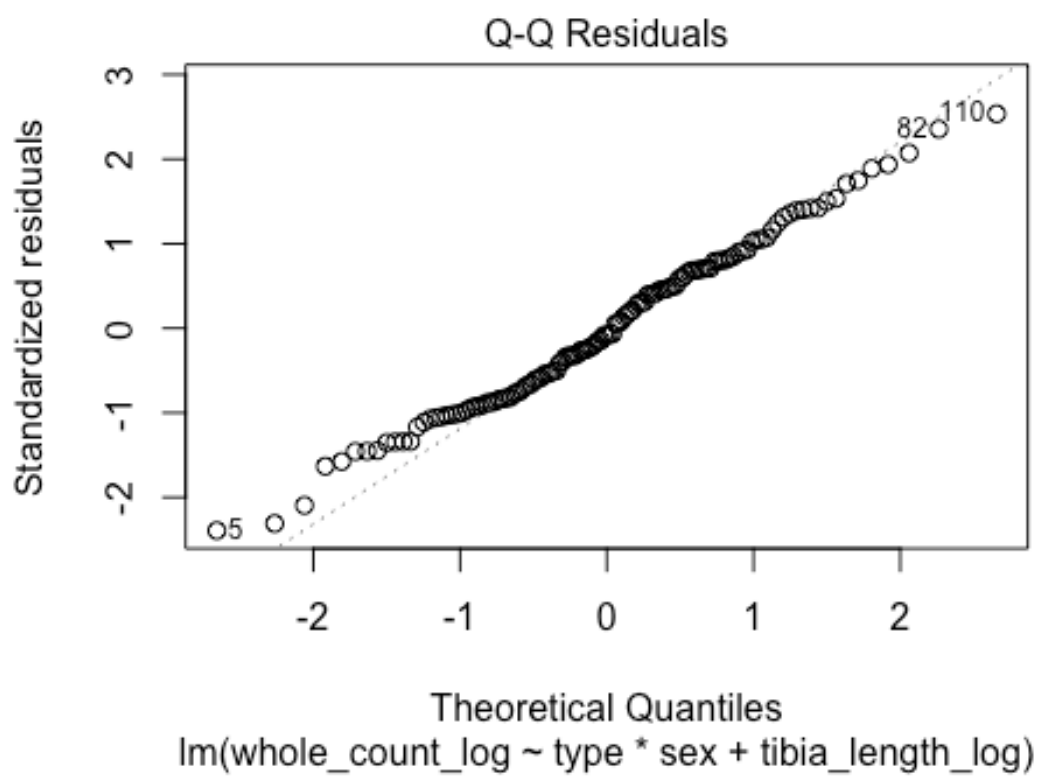
**Histogram of eye\$whole\_count\_log**



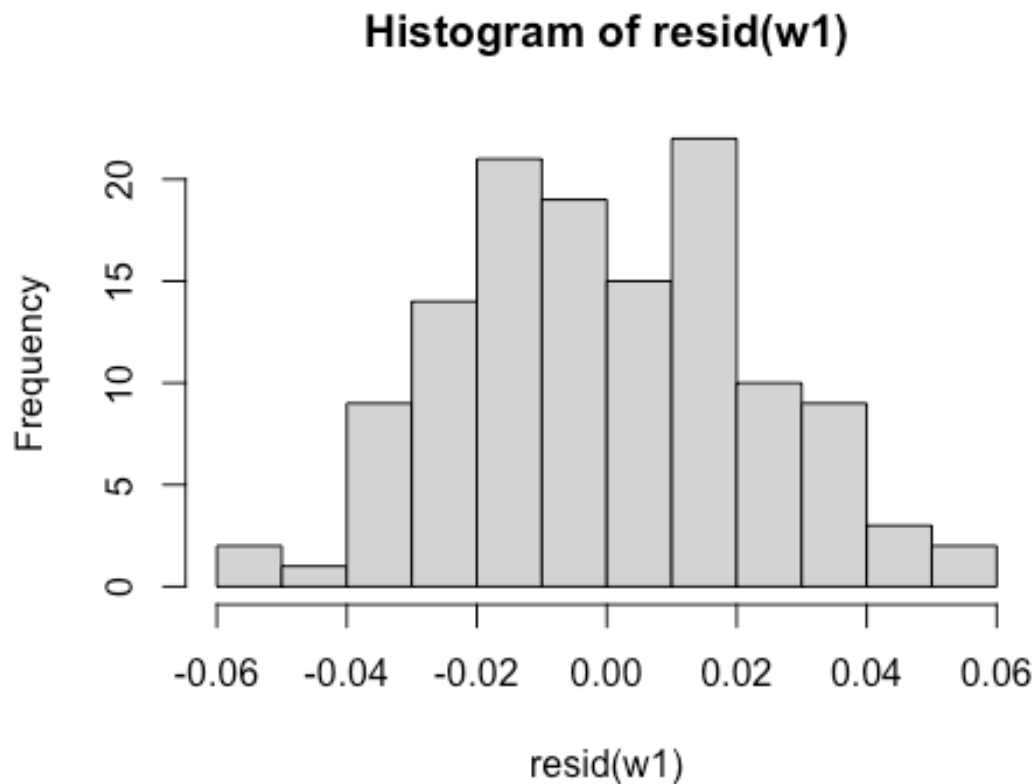
```
w1 <- lm(whole_count_log~type*sex+tibia_length_log, data=eye)
plot(w1, which=1)
```



```
plot(w1, which=2)
```



```
hist(resid(w1))
```



```
drop1(w1, test="Chisq")

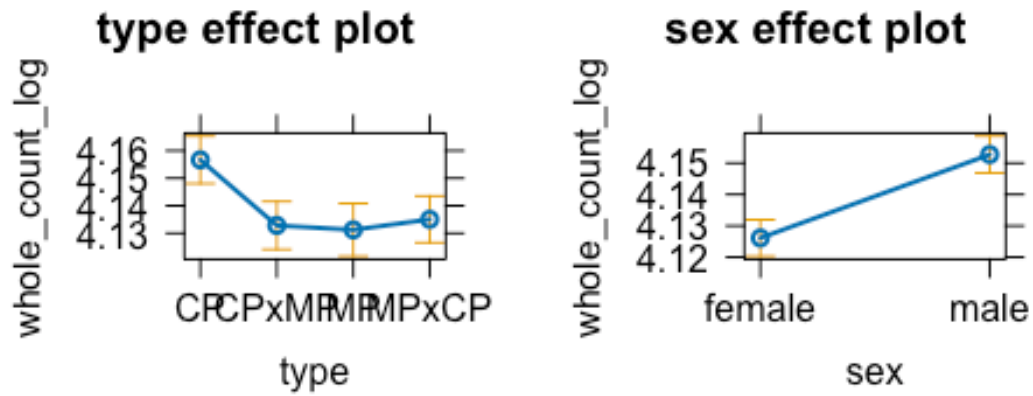
## Single term deletions
##
## Model:
## whole_count_log ~ type * sex + tibia_length_log
##               Df Sum of Sq    RSS   AIC Pr(>Chi)
## <none>                 0.064274 -945.78
## tibia_length_log   1 0.0177991 0.082073 -916.73 2.52e-08 ***
## type:sex           3 0.0022994 0.066573 -947.31  0.2155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

w2 <- update(w1, ~.-type:sex) #type:sex interaction ns
drop1(w2, test="Chisq") #all three significant

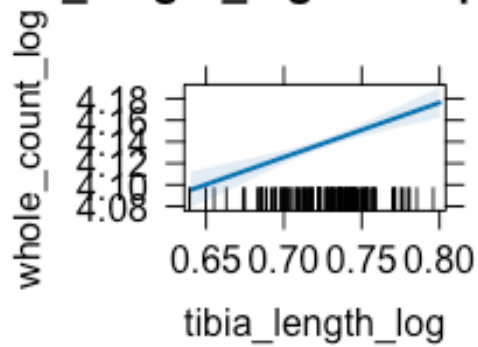
## Single term deletions
##
## Model:
## whole_count_log ~ type + sex + tibia_length_log
##               Df Sum of Sq    RSS   AIC  Pr(>Chi)
## <none>                 0.066573 -947.31
## type                   3  0.011743 0.078316 -932.68 0.0001255 ***
## sex                   1  0.022228 0.088801 -912.72 1.459e-09 ***
```

```
## tibia_length_log 1 0.016372 0.082945 -921.39 1.261e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

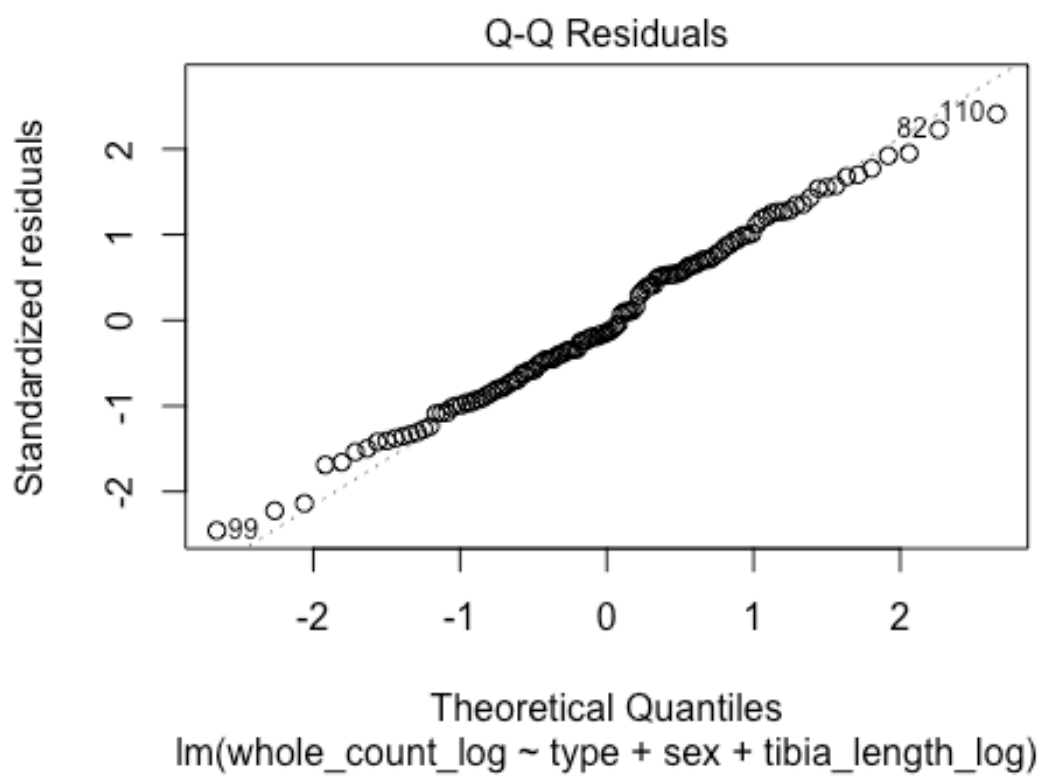
plot(allEffects(w2))
```



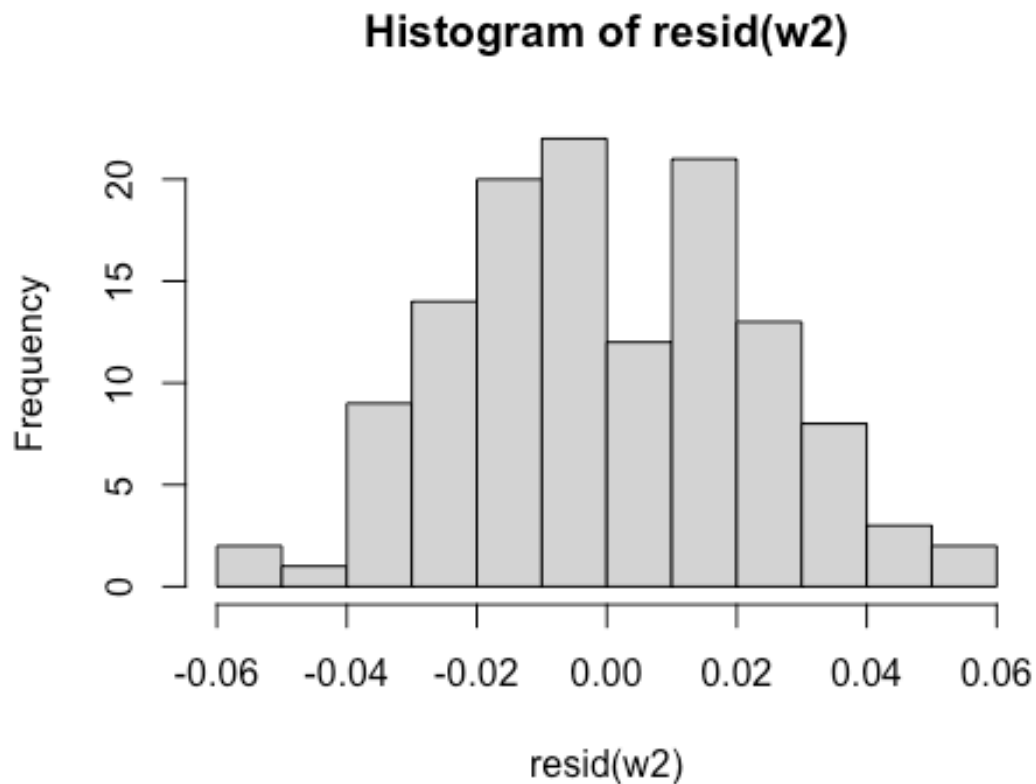
**tibia\_length\_log effect plot**



```
plot(w2, which = 2)
```



```
hist(resid(w2))
```



```
Anova(w2, test.statistic="Chisq")
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: whole_count_log
```

	Sum Sq	Df	F value	Pr(>F)
## type	0.011743	3	7.1147	0.0001924 ***
## sex	0.022228	1	40.4010	3.807e-09 ***
## tibia_length_log	0.016372	1	29.7570	2.639e-07 ***
## Residuals	0.066573	121		

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#type p>0.001
```

```
#sex p>0.001
```

```
#tibia p>0.001
```

```
Anova(w1, test.statistic="Chisq")
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: whole_count_log
```

	Sum Sq	Df	F value	Pr(>F)
## type	0.011743	3	7.1865	0.0001794 ***

```

## sex                0.022228    1 40.8088 3.465e-09 ***
## tibia_length_log 0.017799    1 32.6774 8.336e-08 ***
## type:sex          0.002299    3  1.4072 0.2441719
## Residuals         0.064274 118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#type:sex p=0.244

#posthoc comparisons
#by species type
library(multcomp)
posthoc <- glht(w2, linfct=mcp(type="Tukey"))
summary(posthoc,adjusted(type="bonferroni"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = whole_count_log ~ type + sex + tibia_length_log,
## data = eye)
##
## Linear Hypotheses:
##              Estimate Std. Error t value Pr(>|t|)
## CPxMP - CP == 0   -0.023837   0.005915  -4.030 0.000587 ***
## MP - CP == 0      -0.025446   0.007153  -3.558 0.003213 **
## MPxCP - CP == 0   -0.021629   0.006118  -3.535 0.003471 **
## MP - CPxMP == 0   -0.001608   0.006941  -0.232 1.000000
## MPxCP - CPxMP == 0  0.002208   0.006148   0.359 1.000000
## MPxCP - MP == 0    0.003817   0.006453   0.591 1.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)

#####plot of facet count based on MAM#####

#####**species**#####
model_means_w2 <- emmeans(object = w2, specs = ~ type, type = "response")

#p-value comparison table
pwpm(model_means_w2, adjust="bonferroni", diffs = F)

##              CP    CPxMP      MP    MPxCP
## CP    [4.157]  0.0006  0.0032  0.0035
## CPxMP      [4.133]  1.0000  1.0000
## MP              [4.131]  1.0000
## MPxCP              [4.135]

```



```
##
## Row and column labels: type
## Upper triangle: P values    adjust = "bonferroni"
## Diagonal: [Estimates] (emmean)    type = "response"

#effect size
eff_size(model_means_w2, sigma = sigma(w2), edf = df.residual(w2))

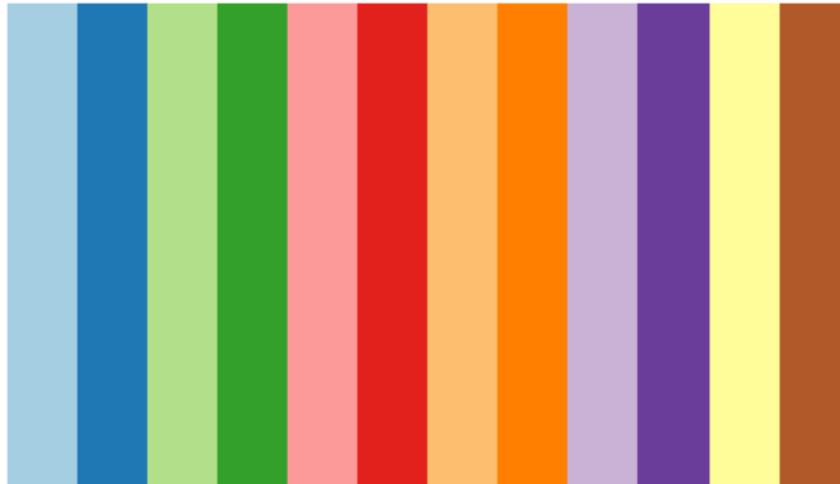
## contrast      estimate    SE  df lower.CL upper.CL
## CP - CPxMP      1.0163 0.260 121    0.501    1.532
## CP - MP         1.0848 0.313 121    0.466    1.704
## CP - MPxCP      0.9221 0.267 121    0.393    1.452
## CPxMP - MP      0.0686 0.296 121   -0.517    0.655
## CPxMP - MPxCP  -0.0942 0.262 121   -0.613    0.425
## MP - MPxCP     -0.1627 0.275 121   -0.708    0.382
##
## Results are averaged over the levels of: sex
## sigma used for effect sizes: 0.02346
## Confidence level used: 0.95

#add letters to each mean to indicate difference
model_means_cld_w2 <- cld(object = model_means_w2,
                          reversed=T,
                          adjust="bonferroni",
                          Letters = letters,
                          alpha = 0.05)

#set marginal means at a dataframe
facet_w2 <- as.data.frame(summary(model_means_cld_w2))

#species

#color selection for consistency with other papers
library(RColorBrewer)
display.brewer.pal(12, 'Paired')
```



Paired (qualitative)

```
brewer.pal(12, 'Paired')

## [1] "#A6CEE3" "#1F78B4" "#B2DF8A" "#33A02C" "#FB9A99" "#E31A1C" "#FDBF6F"
## [8] "#FF7F00" "#CAB2D6" "#6A3D9A" "#FFFF99" "#B15928"

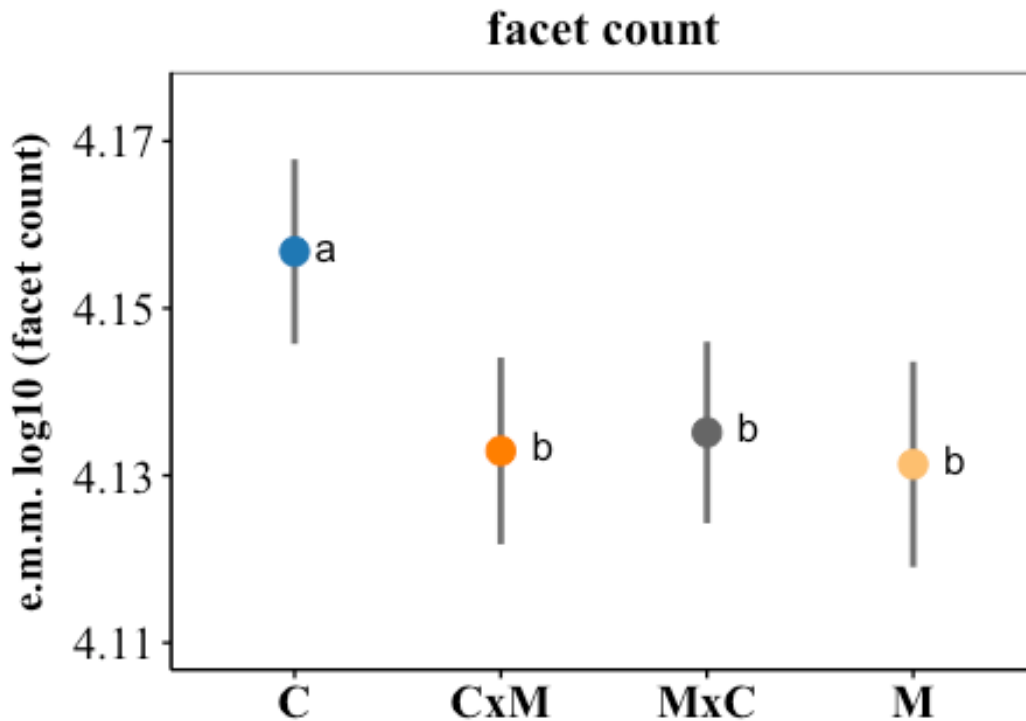
facet_w2$type <- factor(facet_w2$type, c("CP", "CPxMP", "MPxCP", "MP"))

ggplot(facet_w2, aes(type, emmean)) +
  geom_errorbar(data=facet_w2, mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL),
    color="grey45",
    width=0, linewidth=1, position = position_dodge(0.4)) +
  geom_point(data=facet_w2, aes(type, emmean, color=type), size=4,
    position = position_dodge(0.4)) +
  geom_text(aes(label = .group, y = emmean), position = position_dodge(0.4),
    vjust=0.3, hjust=-0.2, size=4.5) +
  labs(x="", y="e.m.m. log10 (facet count)") +
  scale_y_continuous(breaks=seq(4.11, 4.17, 0.02), limits=c(4.11, 4.175)) +
  scale_x_discrete(labels=c("C", "CxM", "MxC", "M")) +
  scale_color_manual(values = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F")) +
  theme_update(axis.title.y=element_text(size=14, face="bold", hjust=0.5, vjust=
2.5, angle=90)) +
  theme(plot.margin = margin(1, 0.5, 0.5, 0.75, "cm")) +
  #theme_update(axis.text.x=element_text(vjust=0.5, angle=45)) +
```

```

  theme(legend.position="none", legend.key=element_blank(),legend.title=element_blank(),
        legend.text=element_text(size=10))+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))+
  ggtitle("facet count") + theme(plot.title = element_text(hjust = 0.5, size=16,face="bold"))

```



```

#####sex#####
model_means_w2.1 <- emmeans(object = w2, specs = ~ sex, type = "response")

#p-value comparison table
pwpm(model_means_w2.1, adjust="bonferroni", diffs = F)

##           female      male
## female [4.126] <.0001
## male           [4.152]
##
## Row and column labels: sex
## Upper triangle: P values
## Diagonal: [Estimates] (emmean)   type = "response"

#effect size
eff_size(model_means_w2.1, sigma = sigma(w2), edf = df.residual(w2))

```

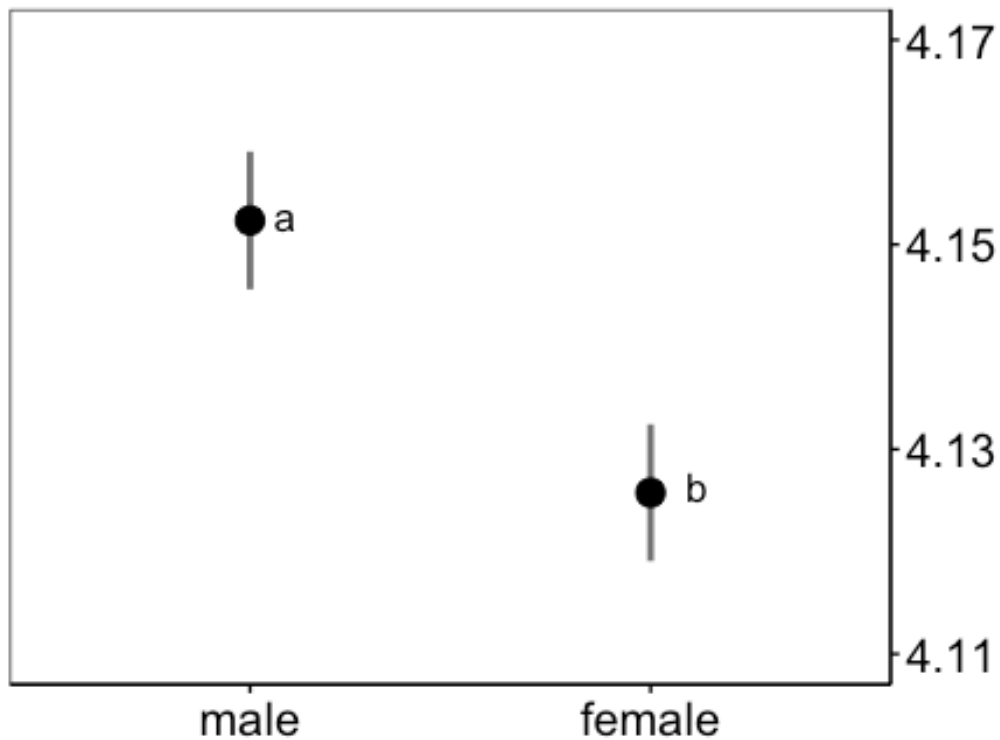
```
## contrast      estimate    SE  df lower.CL upper.CL
## female - male   -1.13 0.193 121   -1.52   -0.753
##
## Results are averaged over the levels of: type
## sigma used for effect sizes: 0.02346
## Confidence level used: 0.95

#add letters to each mean to indicate difference
model_means_cld_w2.1 <- cld(object = model_means_w2.1,
                             reversed = T,
                             adjust="bonferroni",
                             Letters = letters,
                             alpha = 0.05)

#set marginal means at a dataframe
facet_w2.1 <- as.data.frame(summary(model_means_cld_w2.1))

#plot marginal means
facet_w2.1$sex <- factor(facet_w2.1$sex, c("male", "female"))

#sex
ggplot(facet_w2.1, aes(sex, emmean)) +
  geom_errorbar(data=facet_w2.1, mapping=aes(x=sex, ymin=upper.CL, ymax=lower.CL), color="grey45",
               width=0, linewidth=1) +
  geom_point(data=facet_w2.1, aes(sex, emmean), size=4) +
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3, size=4.5) +
  labs(x="", y="") +
  scale_y_continuous(breaks=seq(4.11, 4.17, 0.02), limits=c(4.11, 4.17),
                     position="right") +
  scale_x_discrete(labels = c("male", "female")) +
  theme_update(axis.title.y=element_text(size=14, face="bold", hjust=0.5, vjust=
2.5, angle=90)) +
  theme(legend.position=c(0.88, 0.89), legend.key=element_blank(), legend.title
=element_blank(),
        legend.text=element_text(size=10)) +
  theme(plot.margin = margin(1, 0.25, 0.25, 0.7, "cm"))
```



#####\*\*species:sex\*\*#####

```
model_means_w2.2 <- emmeans(object = w2, specs = ~ type:sex, type = "response")
```

#p-value comparison table

```
pwpm(model_means_w2.2, adjust="bonferroni", diffs = F)
```

##	CP female	CPxMP female	MP female	MPxCP female	CP male	CPxMP male
## CP female	[4.143]	0.0027	0.0150	0.0162	<.0001	1.0000
## CPxMP female		[4.120]	1.0000	1.0000	<.0001	<.0001
## MP female			[4.118]	1.0000	<.0001	0.0000
## MPxCP female				[4.122]	<.0001	0.0000
## CP male					[4.170]	0.0000
## CPxMP male						[4.146]
## MP male						
## MPxCP male						

```

##           MP male MPxCP male
## CP female      1.0000      1.0000
## CPxMP female   0.0534      0.0045
## MP female      <.0001      0.0052
## MPxCP female   0.0805      <.0001
## CP male        0.0150      0.0162
## CPxMP male     1.0000      1.0000
## MP male        [4.145]      1.0000
## MPxCP male           [4.148]
##
## Row and column labels: type:sex
## Upper triangle: P values   adjust = "bonferroni"
## Diagonal: [Estimates] (emmean)   type = "response"

#add letters to each mean to indicate difference
model_means_cld_w2.2 <- cld(object = model_means_w2.2 ,
                             adjust = "bonferroni",
                             reversed=T,
                             Letters = letters,
                             alpha = 0.05)

#set marginal means at a dataframe
facet_w2.2 <- as.data.frame(summary(model_means_cld_w2.2 ))

#plot marginal means
facet_w2.2 $sex <- factor(facet_w2.2 $sex, c("male", "female"))
facet_w2.2 $type <- factor(facet_w2.2 $type, c("CP", "CPxMP", "MPxCP", "MP"))

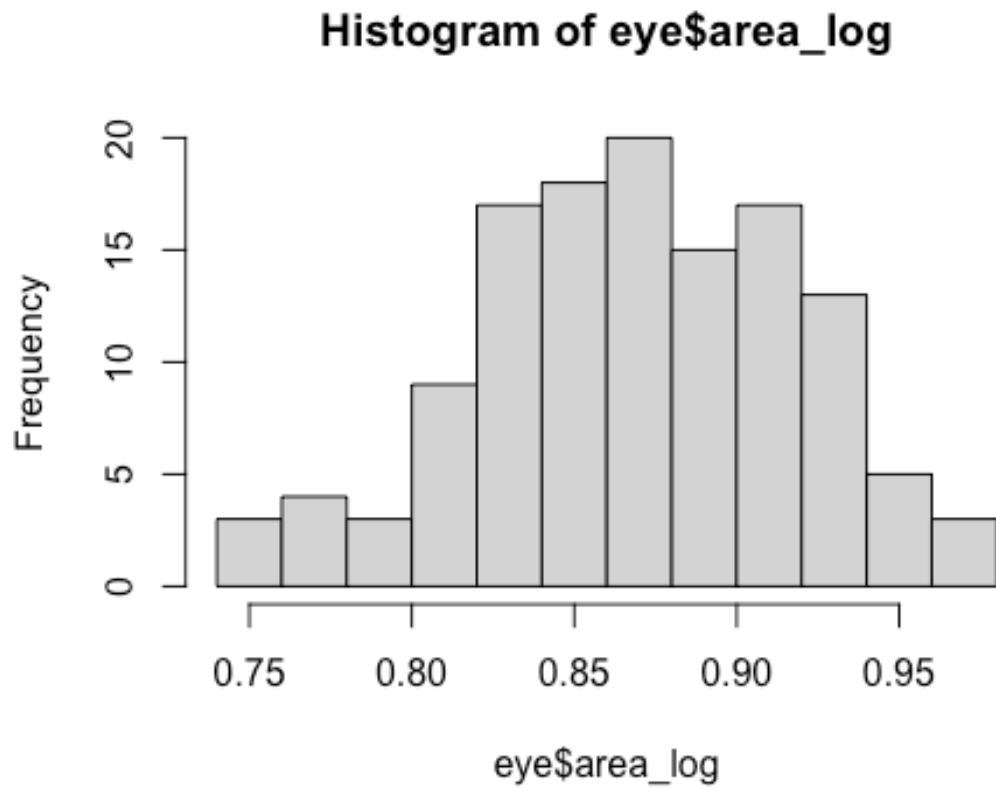
#species:sex plot
combo3 <- ggplot(facet_w2.2 , aes(type,emmean))+
  geom_errorbar(data=facet_w2.2 ,mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL, color=sex),
               width=0, linewidth=1, position = position_dodge(width=0.5))+
  geom_point(data=facet_w2.2 , aes(type,emmean, color=sex), size=3,
             position = position_dodge2(width=0.5))+
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3,size=4.5, position = position_dodge2(width=
0.5)) +
  labs(x="",y="e.m.m. log10 (facet count)", color="")+
  scale_y_continuous(breaks=seq(4.09, 4.19, 0.02), limits=c(4.09, 4.19),
                     position="left")+
  scale_x_discrete(labels=c("C", "CxM", "MxC", "M"))+
  scale_color_manual(values = c("black","grey"))+
  theme(legend.position="none", legend.key=element_blank(),legend.title=element_blank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(1,0.9,0.5,0.5, "cm"))+
  theme(legend.text=element_text(size=11))+
  theme(legend.key=element_blank())+

```

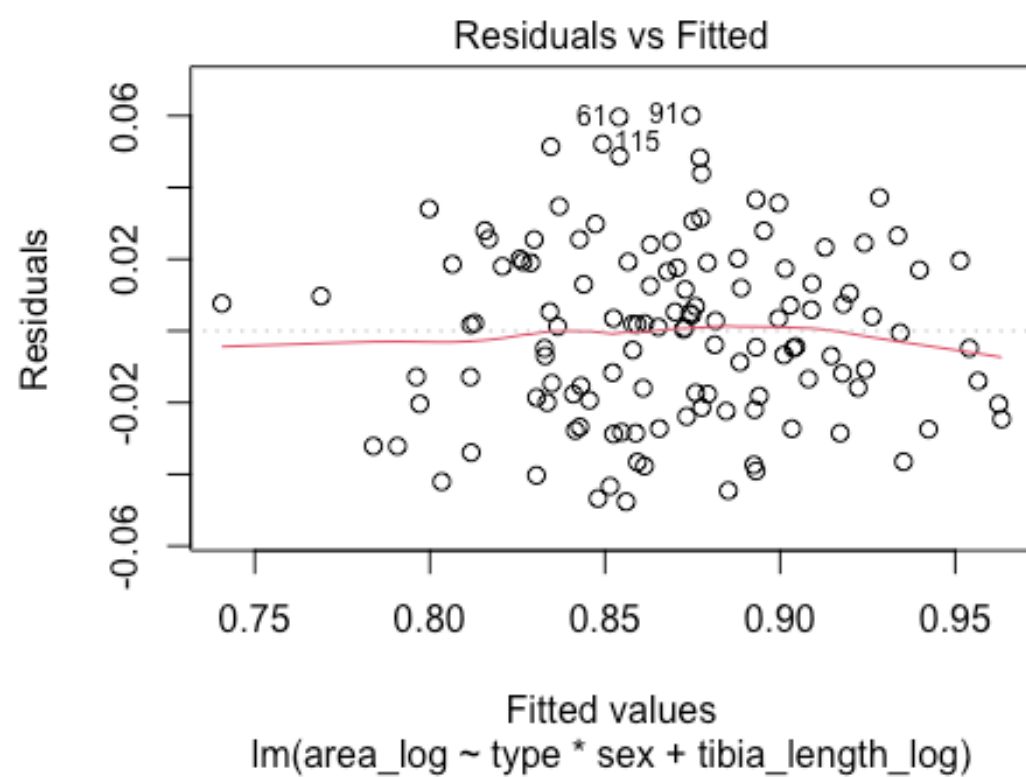
```
theme(axis.text.x=element_text(face = "bold"))+  
theme(text = element_text(family = "Times New Roman"))
```

####ANALYSES OF CORNEAL AREA - C, M, & F1 HYBRIDS####

```
hist(eye$area_log)
```

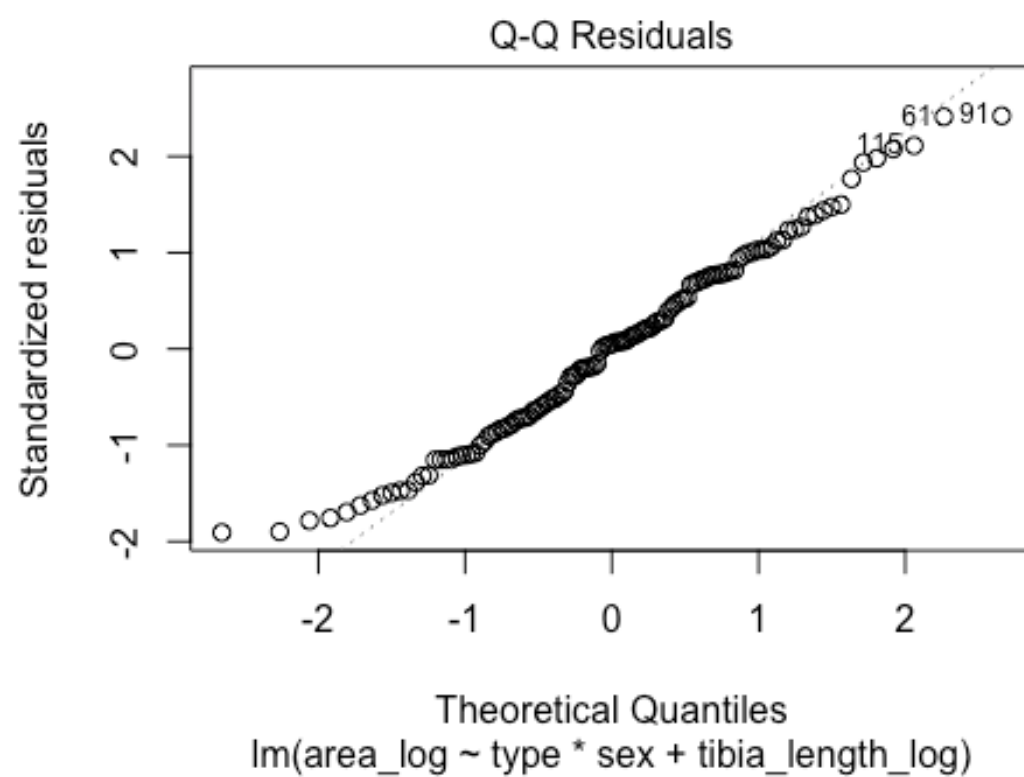


```
k1 <- lm(area_log~type*sex+tibia_length_log, data=eye)  
plot(k1,which=1)
```

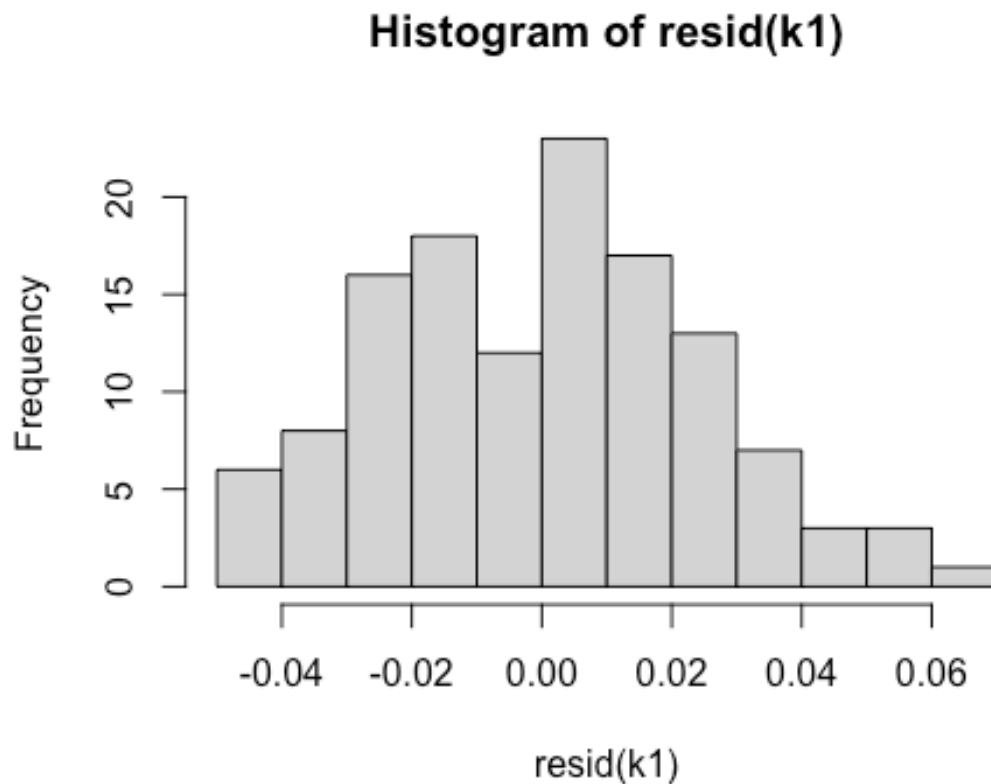


```
plot(k1,which=2)
```





```
hist(resid(k1))
```



```
drop1(k1, test="Chisq") #type:sex n.s.

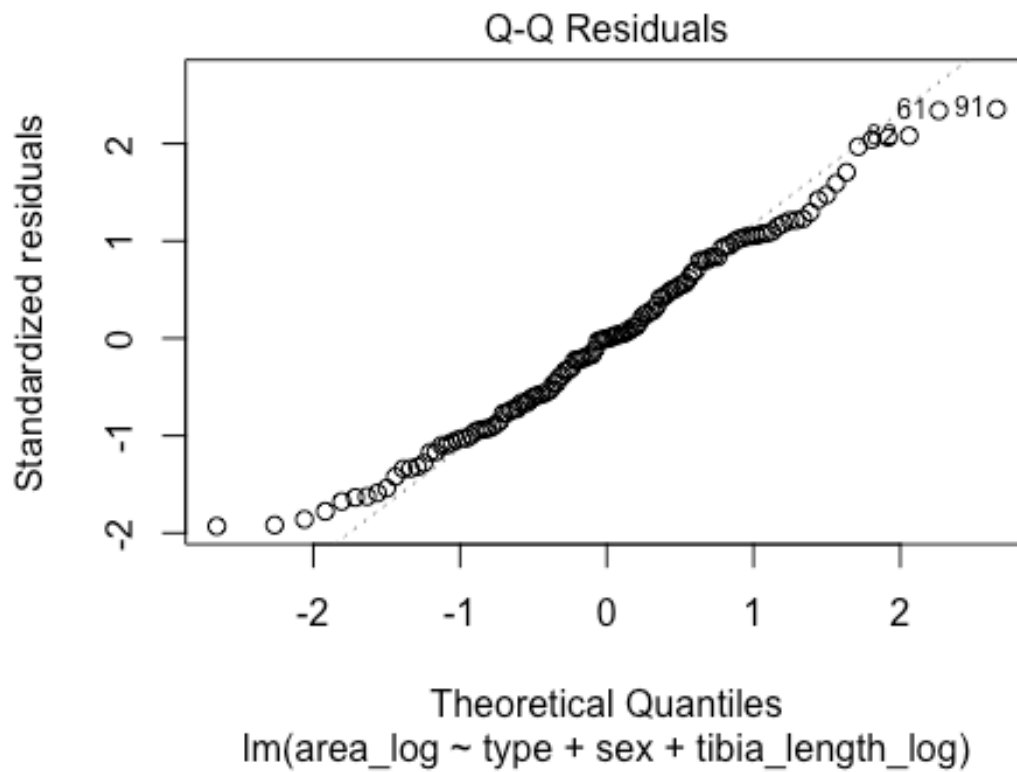
## Single term deletions
##
## Model:
## area_log ~ type * sex + tibia_length_log
##           Df Sum of Sq    RSS    AIC Pr(>Chi)
## <none>                0.077995 -921.20
## tibia_length_log    1  0.089577 0.167573 -826.08   <2e-16 ***
## type:sex            3  0.001785 0.079780 -924.33    0.4115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

k2 <- update(k1, .~.-type:sex)
drop1(k2, test="Chisq") #all effects significant

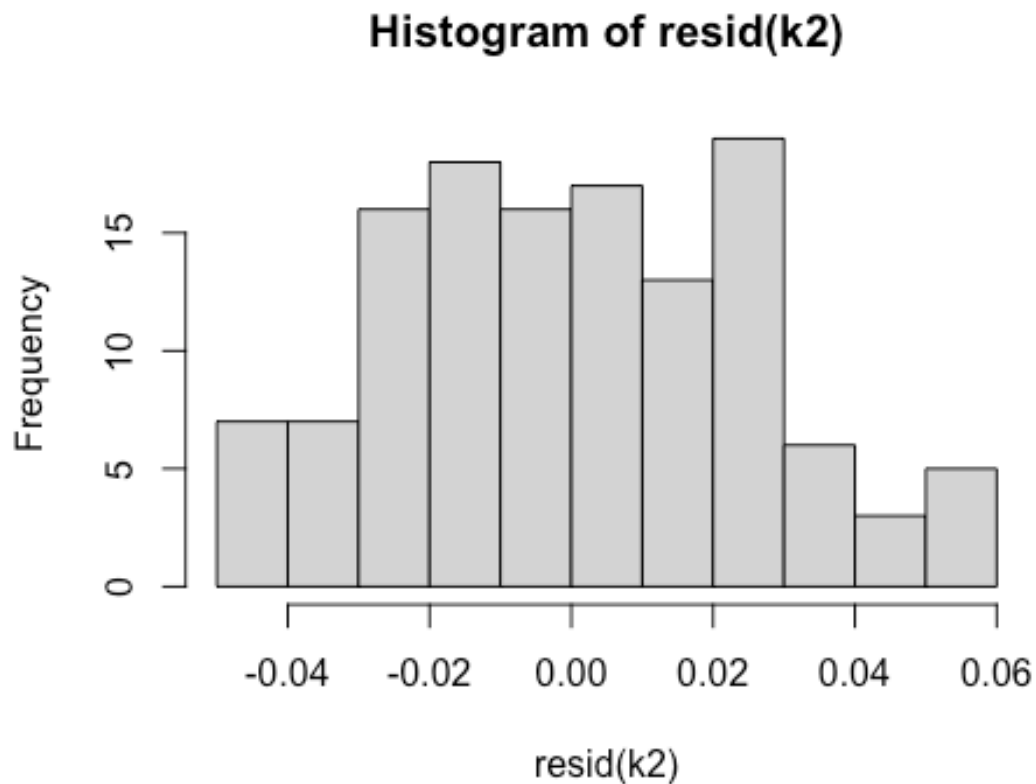
## Single term deletions
##
## Model:
## area_log ~ type + sex + tibia_length_log
##           Df Sum of Sq    RSS    AIC  Pr(>Chi)
## <none>                0.07978 -924.33
## type            3  0.020439 0.10022 -901.36 2.276e-06 ***
## sex             1  0.020961 0.10074 -896.70 5.238e-08 ***
```

```
## tibia_length_log 1 0.090602 0.17038 -829.96 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

plot(k2,which=2)
```



```
hist(resid(k2))
```



```
Anova(k2, test="Chisq")
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: area_log
```

```
##
```

	Sum Sq	Df	F value	Pr(>F)
## type	0.020439	3	10.333	4.180e-06 ***

## sex	0.020961	1	31.791	1.144e-07 ***
--------	----------	---	--------	---------------

## tibia_length_log	0.090602	1	137.413	< 2.2e-16 ***
---------------------	----------	---	---------	---------------

## Residuals	0.079780	121		
--------------	----------	-----	--	--

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

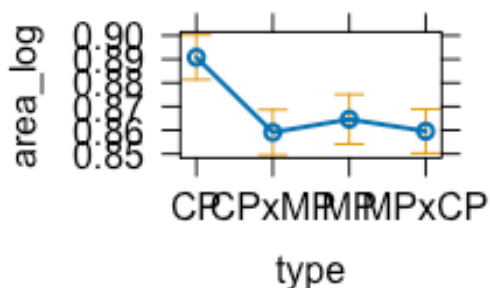
```
#type p>0.001
```

```
#sex p>0.001
```

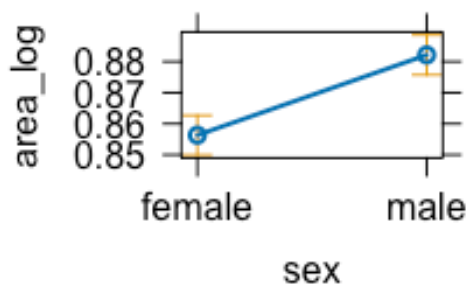
```
#tibia p>0.001
```

```
plot(allEffects(k2))
```

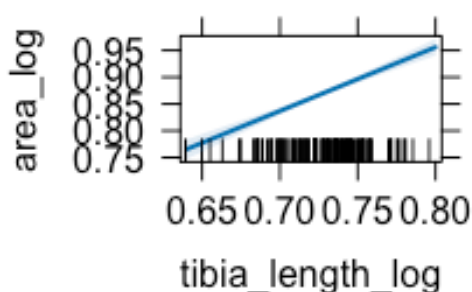
**type effect plot**



**sex effect plot**



**tibia\_length\_log effect plot**



```
#species comparison
posthoc_size2<-glht(k2, linfct=mcp(type="Tukey"))
summary(posthoc_size2, adjusted(type="bonferroni"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = area_log ~ type + sex + tibia_length_log, data = eye)
##
## Linear Hypotheses:
##              Estimate Std. Error t value Pr(>|t|)
## CPxMP - CP == 0    -0.031704   0.006475  -4.897 1.83e-05 ***
## MP - CP == 0       -0.026166   0.007830  -3.342  0.00665 **
## MPxCP - CP == 0    -0.031222   0.006698  -4.661 4.89e-05 ***
## MP - CPxMP == 0     0.005538   0.007599   0.729  1.00000
## MPxCP - CPxMP == 0  0.000482   0.006730   0.072  1.00000
## MPxCP - MP == 0    -0.005056   0.007064  -0.716  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```

```

Anova(k1, test="Chisq")

## Anova Table (Type II tests)
##
## Response: area_log
##
##           Sum Sq Df F value    Pr(>F)
## type           0.020439   3   10.3073 4.456e-06 ***
## sex             0.020961   1   31.7124 1.231e-07 ***
## tibia_length_log 0.089577   1  135.5228 < 2.2e-16 ***
## type:sex        0.001785   3    0.9002   0.4434
## Residuals      0.077995 118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#type:sex p=0.44

#####plot of corneal area based on MAM#####

#####**species**#####

model_means_k2 <- emmeans(object = k2, specs = ~ type, type = "response")

#p-value comparison table
pwpm(model_means_k2, adjust="bonferroni", diffs = F)

##           CP    CPxMP      MP    MPxCP
## CP    [0.891] <.0001  0.0066 <.0001
## CPxMP      [0.859]  1.0000  1.0000
## MP                [0.865]  1.0000
## MPxCP                [0.860]
##
## Row and column labels: type
## Upper triangle: P values adjust = "bonferroni"
## Diagonal: [Estimates] (emmean) type = "response"

#effect size
eff_size(model_means_k2, sigma = sigma(k2), edf = df.residual(k2))

## contrast      estimate    SE df lower.CL upper.CL
## CP - CPxMP      1.2347 0.264 121    0.711    1.758
## CP - MP          1.0190 0.312 121    0.402    1.636
## CP - MPxCP      1.2159 0.272 121    0.677    1.755
## CPxMP - MP      -0.2157 0.296 121   -0.802    0.371
## CPxMP - MPxCP   -0.0188 0.262 121   -0.538    0.500
## MP - MPxCP       0.1969 0.275 121   -0.348    0.742
##
## Results are averaged over the levels of: sex
## sigma used for effect sizes: 0.02568
## Confidence level used: 0.95

```

```

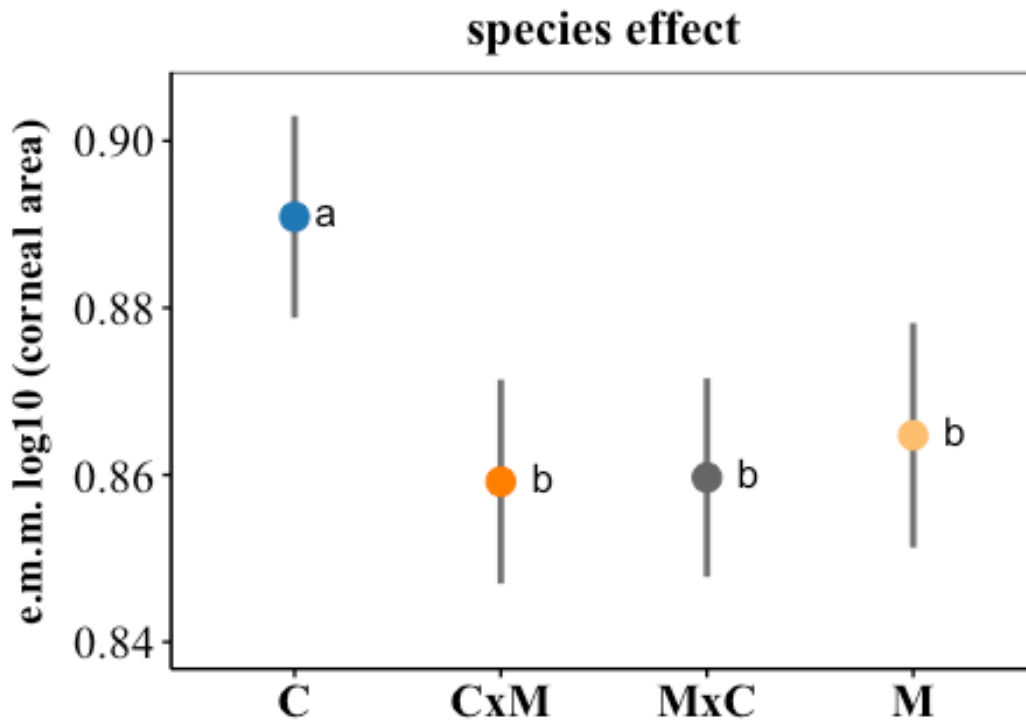
#add letters to each mean to indicate difference
model_means_cld_k2 <- cld(object = model_means_k2,
                           reversed=T,
                           adjust="bonferroni",
                           Letters = letters,
                           alpha = 0.05)

#set marginal means at a dataframe
area_k2 <- as.data.frame(summary(model_means_cld_k2))

area_k2$type <- factor(area_k2$type, c("CP", "CPxMP", "MPxCP", "MP"))

#species
ggplot(area_k2, aes(type,emmean))+
  geom_errorbar(data=area_k2,mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL),
               color="grey45",
               width=0, linewidth=1, position = position_dodge(0.4))+
  geom_point(data=area_k2, aes(type,emmean,color=type), size=4,
             position = position_dodge(0.4))+
  geom_text(aes(label = .group, y = emmean), position = position_dodge(0.4),
            vjust=0.3, hjust=-0.2, size=4.5) +
  labs(x="",y="e.m.m. log10 (corneal area)")+
  scale_y_continuous(breaks=seq(0.82, 0.92, 0.02), limits=c(0.84, 0.905))+
  scale_x_discrete(labels=c("C", "CxM", "MxC", "M"))+
  scale_color_manual(values = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"))+
  theme_update(axis.title.y=element_text(size=14,face="bold",hjust=0.5,vjust=
2.5,angle=90))+
  theme(legend.position="none", legend.key=element_blank(),legend.title=element_blank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(1,0.5,0.5,0.75, "cm"))+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))+
  ggtitle("species effect") + theme(plot.title = element_text(hjust = 0.5, size=16,face="bold"))

```



```
#####sex#####
model_means2_k2.1 <- emmeans(object = k2, specs = ~ sex, type = "response")

#p-value comparison table
pwpm(model_means2_k2.1, adjust="bonferroni", diffs = F)

##           female      male
## female [0.856] <.0001
## male      [0.882]
##
## Row and column labels: sex
## Upper triangle: P values
## Diagonal: [Estimates] (emmean)   type = "response"

#effect size
eff_size(model_means2_k2.1, sigma = sigma(k2), edf = df.residual(k2))

## contrast      estimate    SE  df lower.CL upper.CL
## female - male    -1.01 0.19 121    -1.38    -0.63
##
## Results are averaged over the levels of: type
## sigma used for effect sizes: 0.02568
## Confidence level used: 0.95
```



```

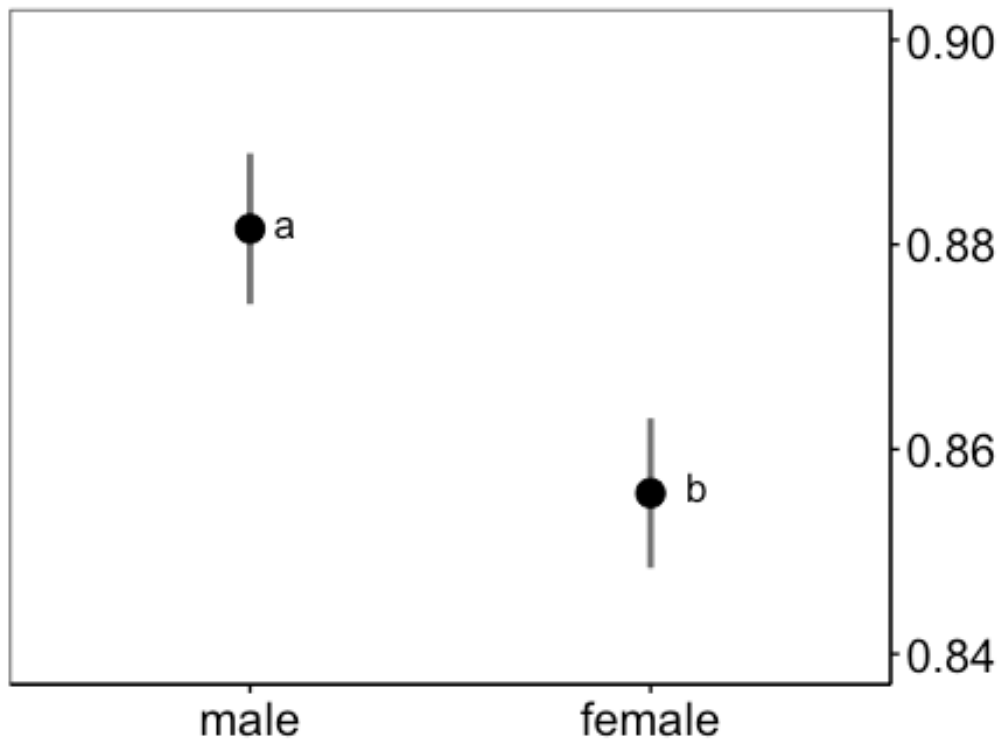
#add letters to each mean to indicate difference
model_means_cld2_k2.1 <- cld(object = model_means2_k2.1,
                             reversed = T,
                             adjust="bonferroni",
                             Letters = letters,
                             alpha = 0.05)

#set marginal means at a dataframe
area_k2.1 <- as.data.frame(summary(model_means_cld2_k2.1))

#plot marginal means
area_k2.1$sex <- factor(area_k2.1$sex, c("male", "female"))

#sex
ggplot(area_k2.1, aes(sex, emmean))+
  geom_errorbar(data=area_k2.1, mapping=aes(x=sex, ymin=upper.CL, ymax=lower.CL),
               color="grey45",
               width=0, linewidth=1)+
  geom_point(data=area_k2.1, aes(sex, emmean), size=4)+
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3, size=4.5) +
  labs(x="", y="")+
  scale_y_continuous(breaks=seq(0.84, 0.90, 0.02), limits=c(0.84, 0.90),
                     position="right")+
  scale_x_discrete(labels = c("male", "female"))+
  theme_update(axis.title.y=element_text(size=14, face="bold", hjust=0.5, vjust=
2.5, angle=90))+
  theme(legend.position=c(0.88, 0.89), legend.key=element_blank(), legend.title
=element_blank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(1, 0.25, 0.25, 0.7, "cm"))

```



#####*species:sex*#####

```
model_means3_k2.2 <- emmeans(object = k2, specs = ~ type:sex, type = "response")
```

*#p-value comparison table*

```
pwpm(model_means3_k2.2, adjust="bonferroni", diffs = F)
```

	CP female	CPxMP female	MP female	MPxCP female	CP male	CPxMP male
CP female	[0.878]	<.0001	0.0310	0.0002	<.0001	1.0000
CPxMP female		[0.846]	1.0000	1.0000	<.0001	<.0001
MP female			[0.852]	1.0000	<.0001	0.7813
MPxCP female				[0.847]	<.0001	0.6813
CP male					[0.904]	<.0001
CPxMP male						[0.872]
MP male						
MPxCP male						

```

##           MP male MPxCP male
## CP female      1.0000      1.0000
## CPxMP female   0.0113      0.0418
## MP female      <.0001      0.4942
## MPxCP female   0.0072      <.0001
## CP male        0.0310      0.0002
## CPxMP male     1.0000      1.0000
## MP male        [0.878]      1.0000
## MPxCP male           [0.873]
##
## Row and column labels: type:sex
## Upper triangle: P values   adjust = "bonferroni"
## Diagonal: [Estimates] (emmean)   type = "response"

#add letters to each mean to indicate difference
model_means_cld3_k2.2 <- cld(object = model_means3_k2.2,
                             adjust = "bonferroni",
                             reversed = T,
                             Letters = letters,
                             alpha = 0.05)

#set marginal means at a dataframe
area_k2.2 <- as.data.frame(summary(model_means_cld3_k2.2))

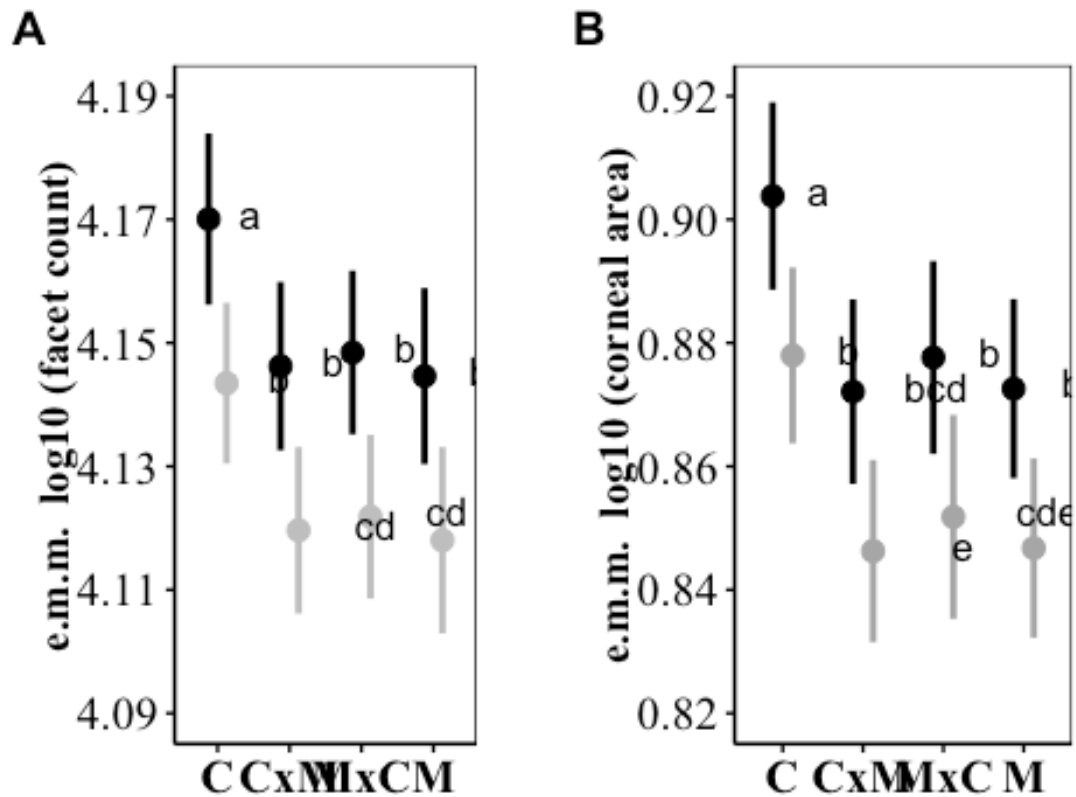
#plot marginal means
area_k2.2$sex <- factor(area_k2.2$sex, c("male", "female"))

#species:sex plot
combo4<-ggplot(area_k2.2, aes(type,emmean))+
  geom_errorbar(data=area_k2.2,mapping=aes(x=type, ymin=upper.CL, ymax=lower.
CL, color=sex),
               width=0, linewidth=1, position = position_dodge(width=0.5))+
  geom_point(data=area_k2.2, aes(type,emmean, color=sex), size=3,
             position = position_dodge2(width=0.5))+
  geom_text(aes(label = .group, y = emmean),
            vjust=0.3, hjust=-0.3,size=4.5, position = position_dodge2(width=
0.5)) +
  labs(x="",y="e.m.m. log10 (corneal area)", color="")+
  scale_y_continuous(breaks=seq(0.82, 0.92, 0.02), limits=c(0.82, 0.92))+
  scale_x_discrete(labels=c("C", "CxM", "MxC", "M"))+
  scale_color_manual(values=c("black","grey65"))+
  theme(legend.position="", legend.key=element_blank(),legend.title=element_b
lank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(1,0.5,0.5,0.5, "cm"))+
  theme(legend.text=element_text(size=11))+
  theme(legend.key=element_blank())+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))

```

```
#####*combo species:sex figure - all species#####
```

```
ggarrange(combo3, combo4,
  labels = c("A", "B"),
  vjust=2,
  ncol = 2, nrow = 1, widths = c(1, 1),
  font.label = list(size = 15))
```

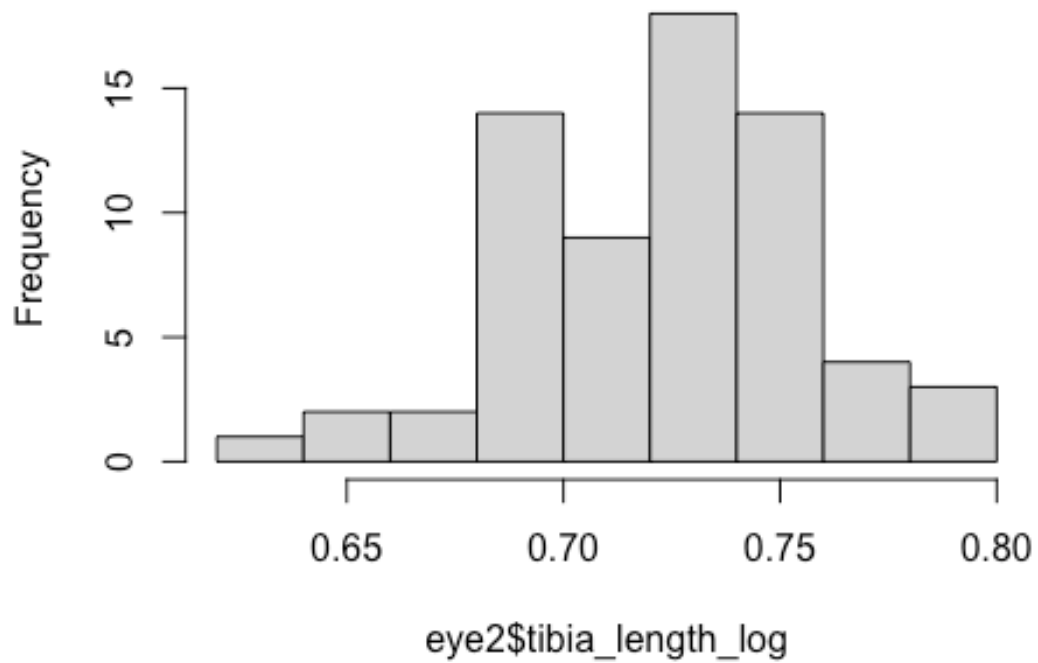


```
####DO SPECIES DIFFER IN BODY SIZE?####
```

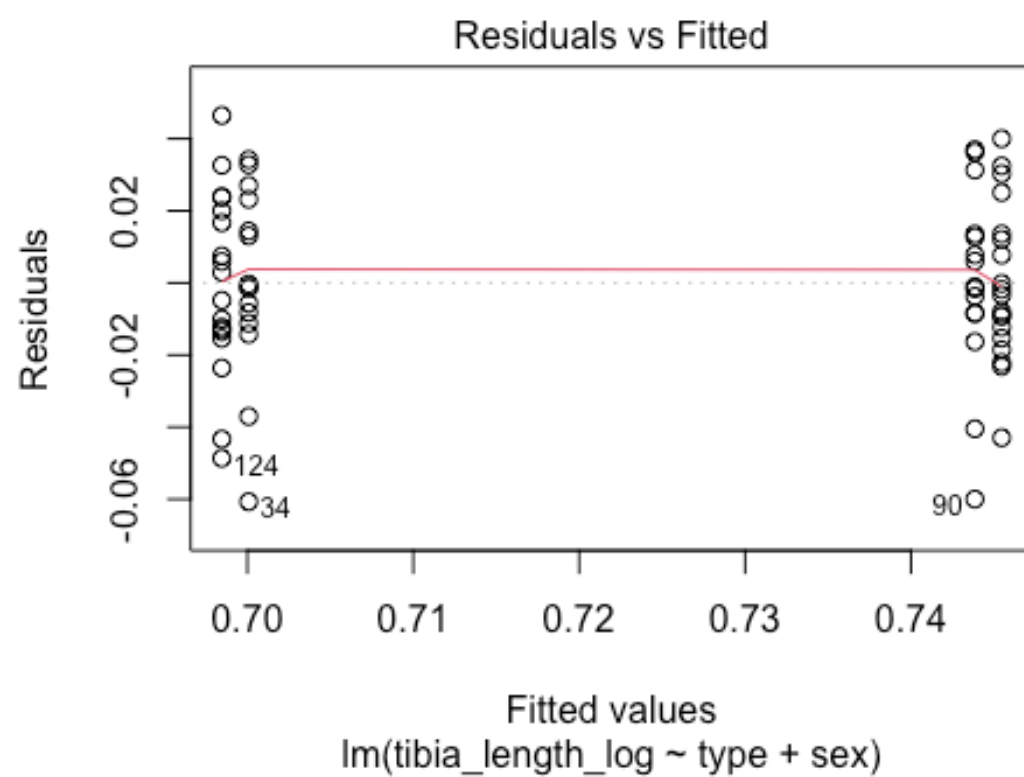
```
#pure species
```

```
hist(eye2$tibia_length_log)
```

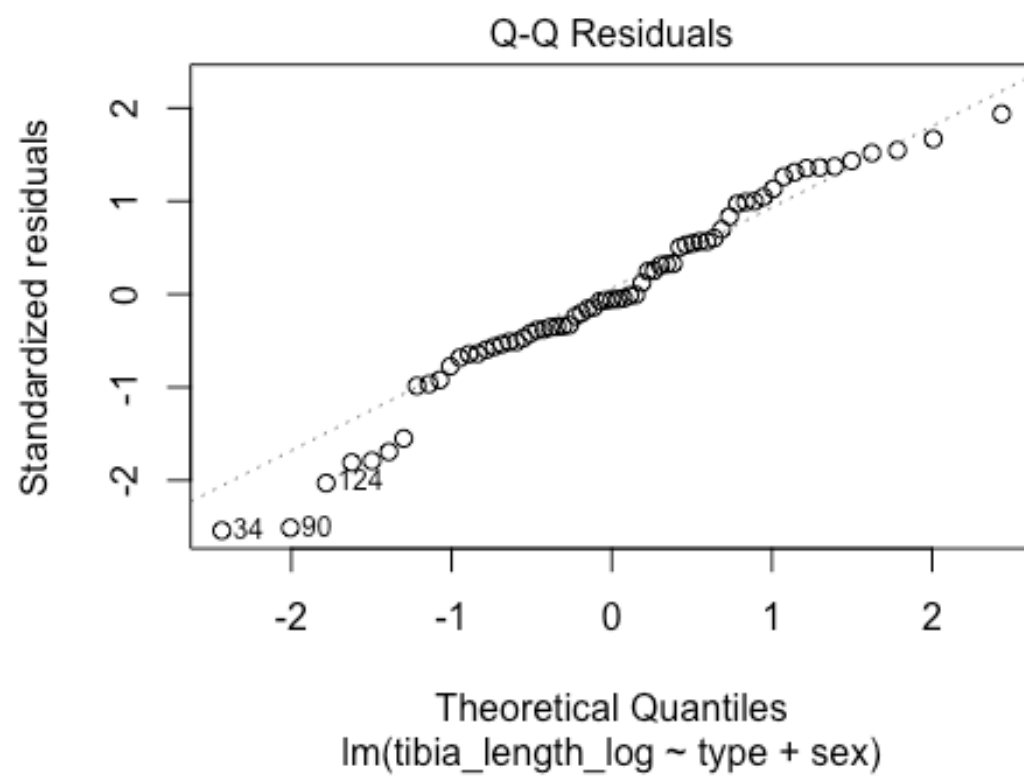
**Histogram of eye2\$tibia\_length\_log**



```
size <- lm(tibia_length_log~type+sex, data=eye2)
plot(size, which=1)
```

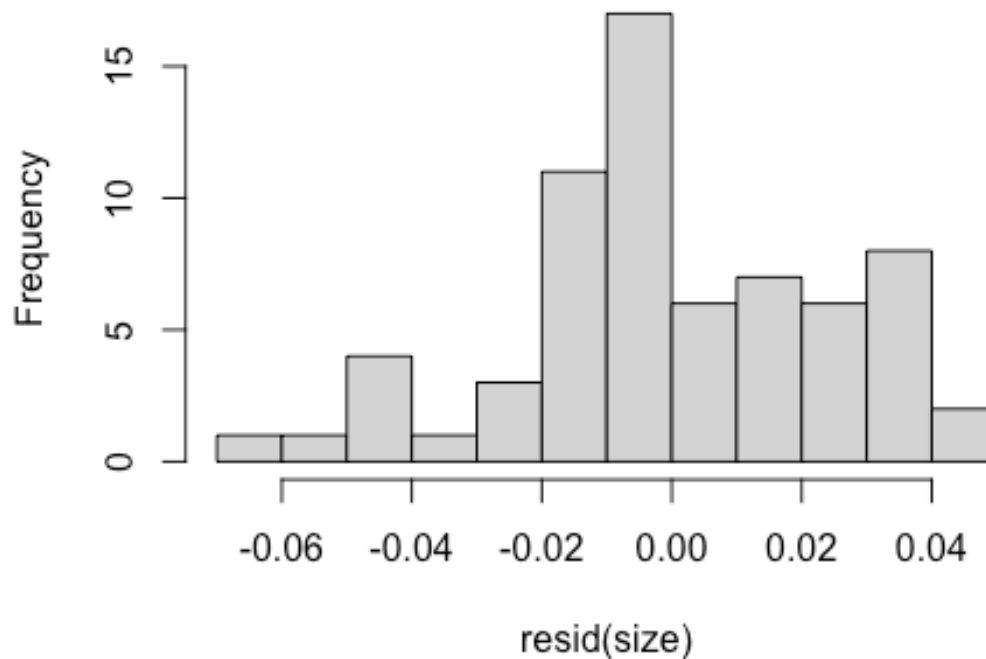


```
plot(size, which=2)
```



```
hist(resid(size))
```

Histogram of resid(size)



```
drop1(size, test="Chisq") #sex n.s.

## Single term deletions
##
## Model:
## tibia_length_log ~ type + sex
##      Df Sum of Sq    RSS    AIC  Pr(>Chi)
## <none>          0.038392 -494.13
## type   1  0.034117 0.072509 -453.53 6.707e-11 ***
## sex    1  0.000043 0.038435 -496.05  0.7833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

size2 <- update(size, ~.-sex)
drop1(size2, test="Chisq")

## Single term deletions
##
## Model:
## tibia_length_log ~ type
##      Df Sum of Sq    RSS    AIC  Pr(>Chi)
## <none>          0.038435 -496.05
## type   1  0.034749 0.073184 -454.90 5.075e-11 ***
```

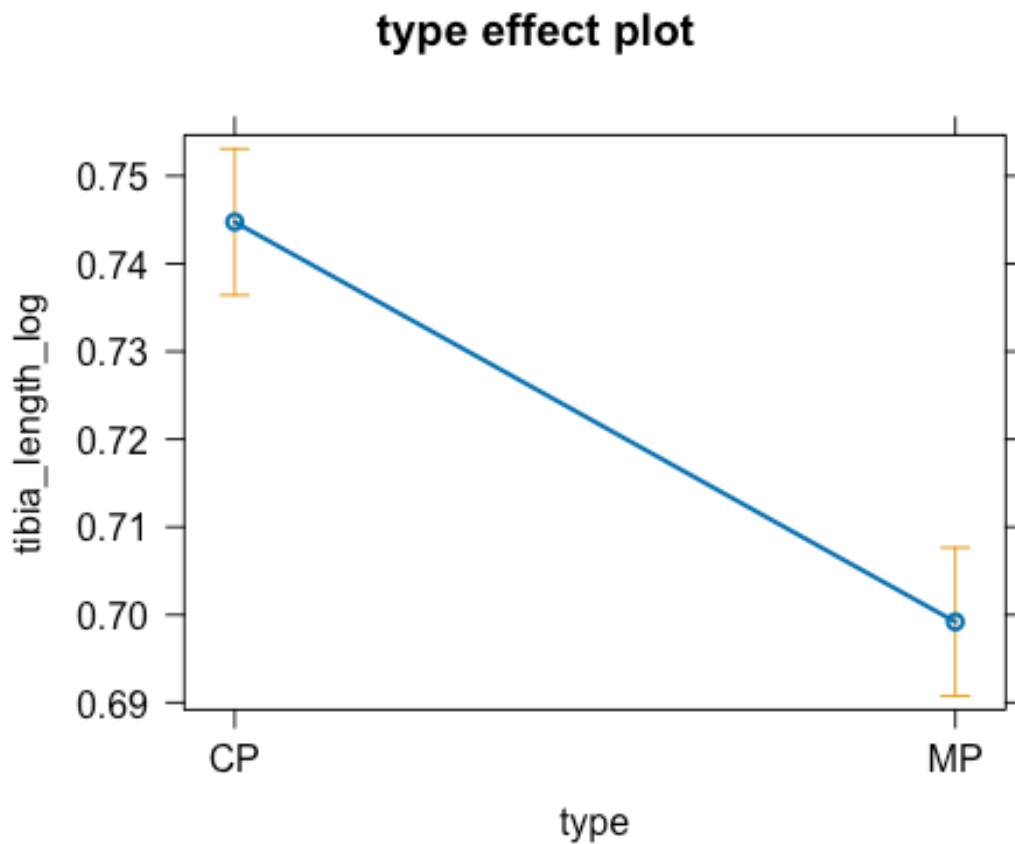


```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova(size2, test="Chisq")

## Anova Table (Type II tests)
##
## Response: tibia_length_log
##           Sum Sq Df F value    Pr(>F)
## type       0.034749  1  58.765 1.145e-10 ***
## Residuals  0.038435 65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#species p<0.001
plot(allEffects(size2))
```



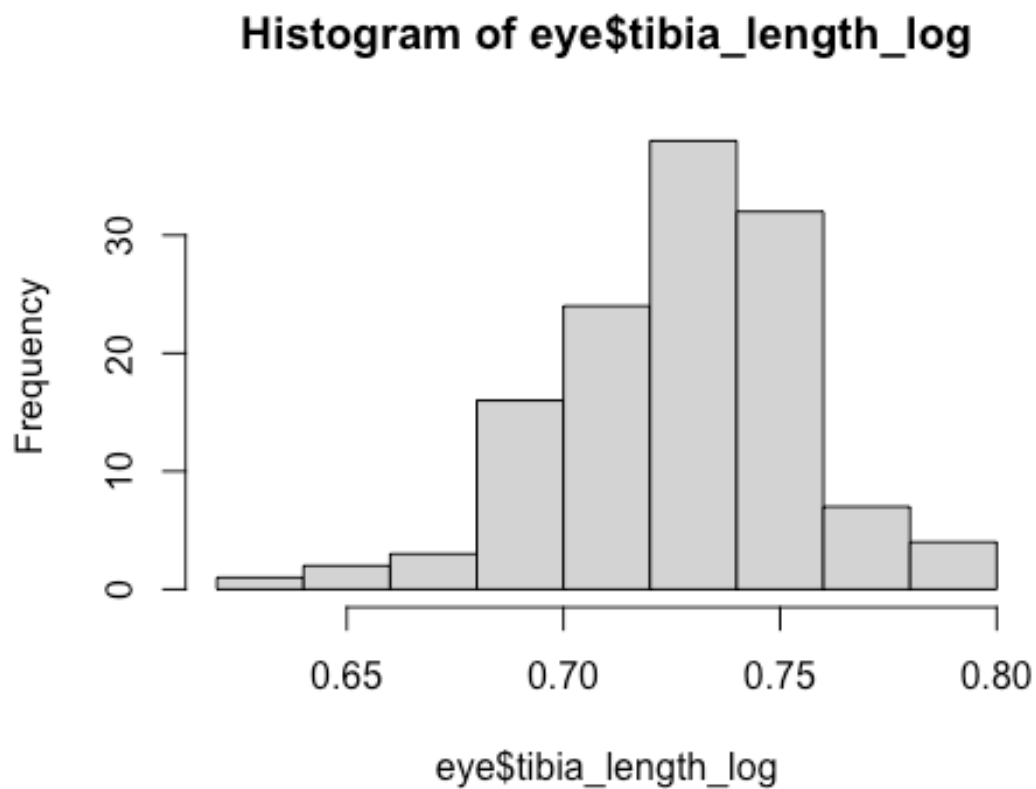
```
Anova(size, test="Chisq")

## Anova Table (Type II tests)
##
## Response: tibia_length_log
##           Sum Sq Df F value    Pr(>F)
## type       0.034117  1  56.8730 2.075e-10 ***
```

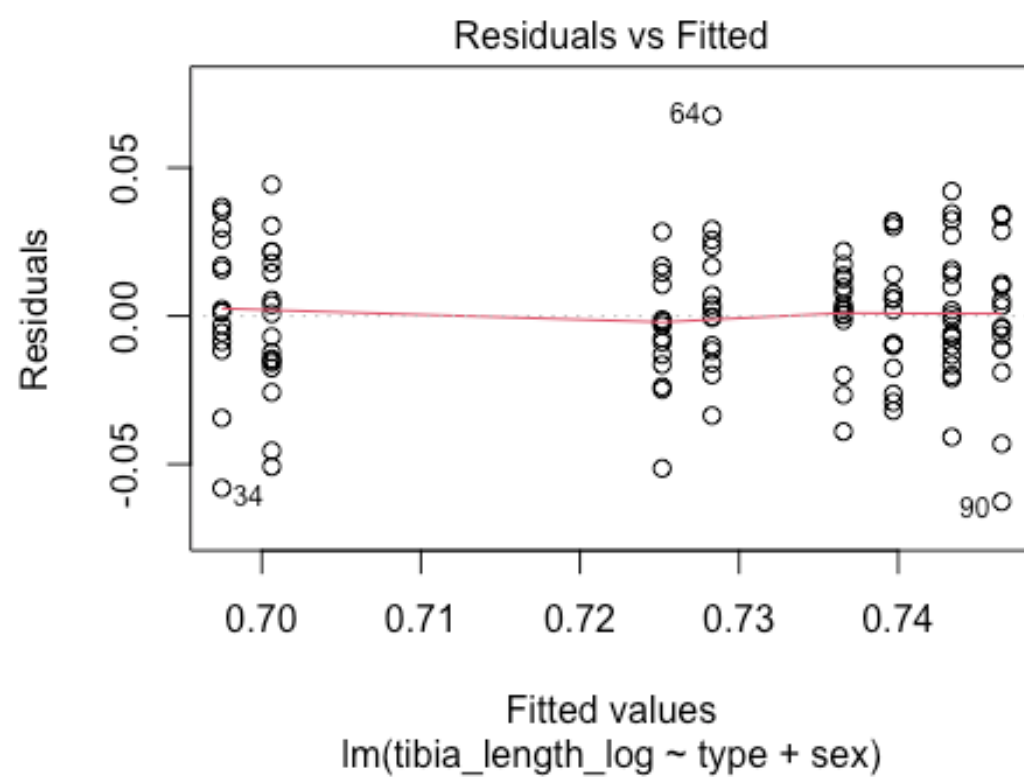
```
## sex      0.000043  1  0.0723    0.7889
## Residuals 0.038392 64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#sex p=0.788

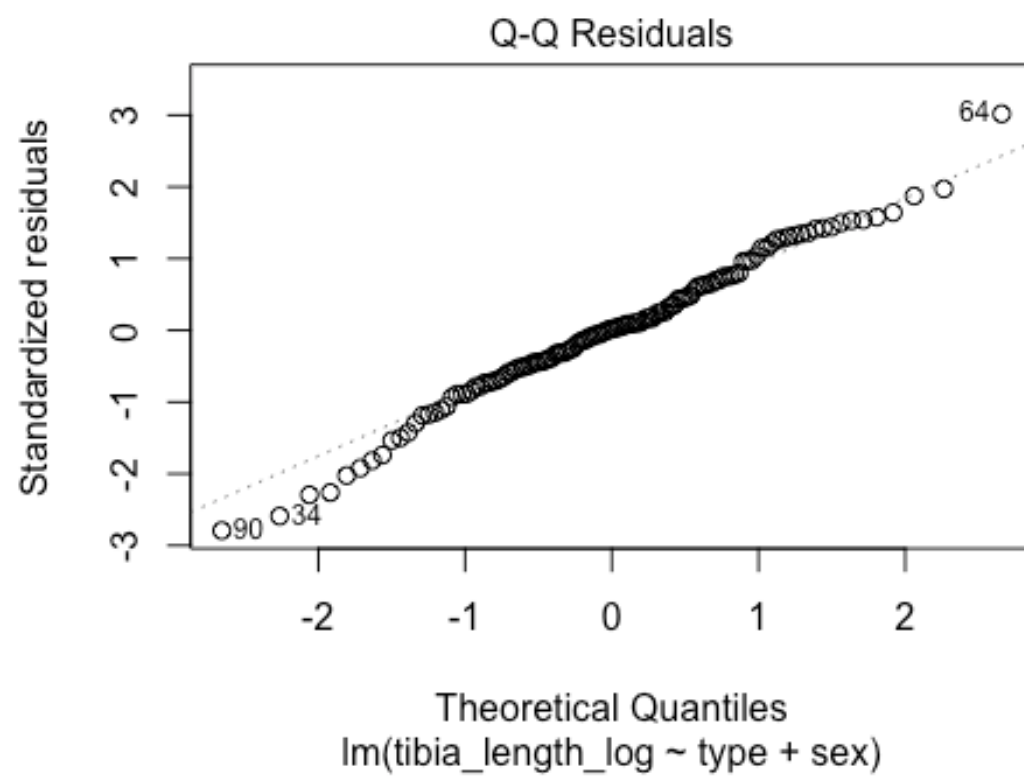
#with F1 hybrids
hist(eye$tibia_length_log)
```



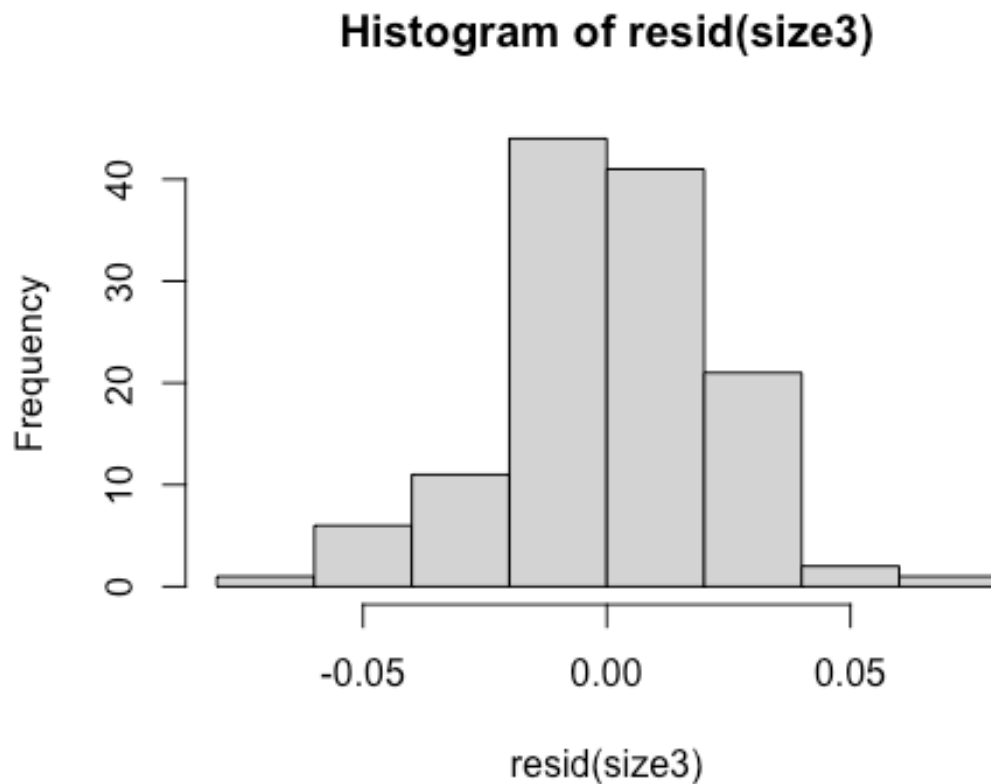
```
size3 <- lm(tibia_length_log~type+sex, data=eye)
plot(size3, which=1)
```



```
plot(size3, which=2)
```



```
hist(resid(size3))
```



```
drop1(size3, test="Chisq") #sex n.s.

## Single term deletions
##
## Model:
## tibia_length_log ~ type + sex
##      Df Sum of Sq    RSS    AIC  Pr(>Chi)
## <none>          0.063930 -954.46
## type    3  0.040227 0.104158 -898.47 2.208e-13 ***
## sex     1  0.000310 0.064241 -955.84    0.433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

size4 <- update(size3, ~.-sex)
drop1(size4, test="Chisq")

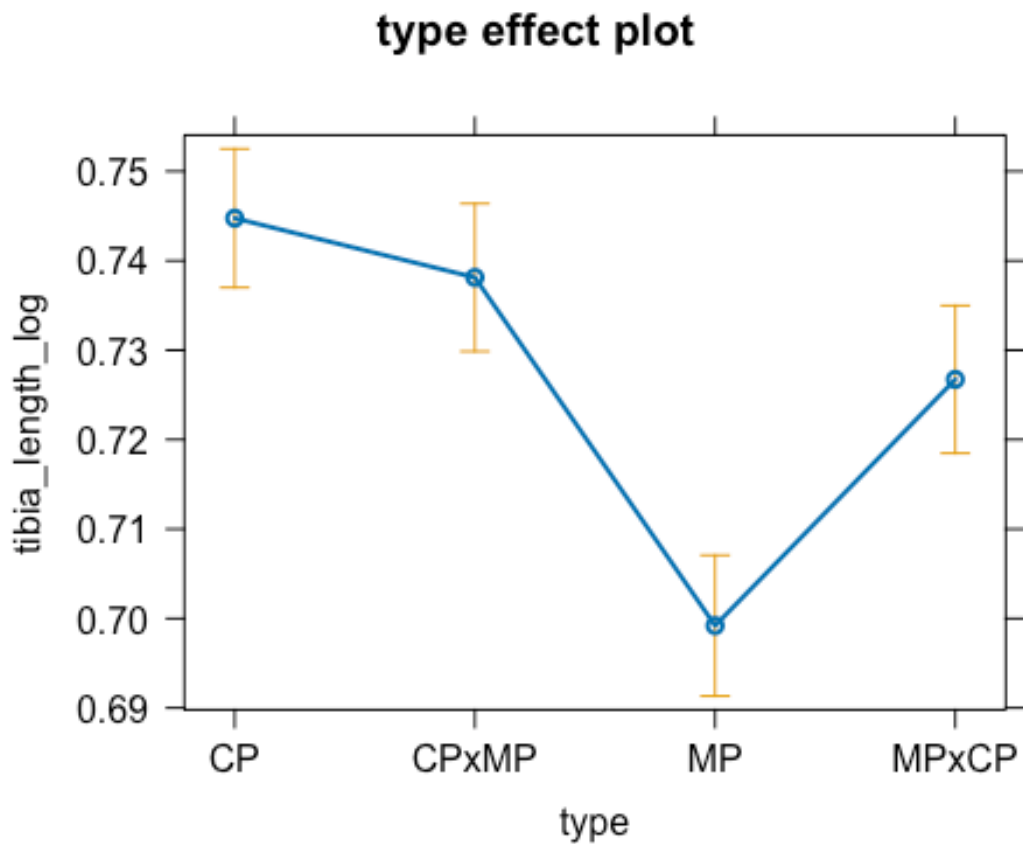
## Single term deletions
##
## Model:
## tibia_length_log ~ type
##      Df Sum of Sq    RSS    AIC  Pr(>Chi)
## <none>          0.064241 -955.84
## type    3  0.039932 0.104173 -900.45 2.961e-13 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova(size4, test="Chisq")

## Anova Table (Type II tests)
##
## Response: tibia_length_log
##          Sum Sq Df F value    Pr(>F)
## type      0.039932  3  25.486 6.842e-13 ***
## Residuals 0.064241 123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#type p>0.001
plot(allEffects(size4))
```



```
Anova(size3, test="Chisq")

## Anova Table (Type II tests)
##
## Response: tibia_length_log
##          Sum Sq Df F value    Pr(>F)
## type      0.040227  3 25.5890 6.541e-13 ***
```

```

## sex          0.000310    1  0.5919    0.4432
## Residuals 0.063930 122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#sex p=0.44

#species comparison
posthoc_size<-glht(size4, linfct=mcp(type="Tukey"))
summary(posthoc_size, adjusted(type="bonferroni"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = tibia_length_log ~ type, data = eye)
##
## Linear Hypotheses:
##              Estimate Std. Error t value Pr(>|t|)
## CPxMP - CP == 0    -0.006628   0.005725  -1.158   1.0000
## MP - CP == 0       -0.045552   0.005585  -8.157 2.03e-12 ***
## MPxCP - CP == 0    -0.018022   0.005725  -3.148   0.0124 *
## MP - CPxMP == 0    -0.038924   0.005765  -6.752 3.08e-09 ***
## MPxCP - CPxMP == 0  -0.011394   0.005901  -1.931   0.3348
## MPxCP - MP == 0     0.027531   0.005765   4.775 3.00e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)

#####Body size emm plot#####
model_means_size <- emmeans(object = size4, specs = ~ type, type = "response"
)

#p-value comparison table
pwpm(model_means_size, adjust="bonferroni", diffs = F)

##              CP    CPxMP      MP    MPxCP
## CP    [0.745]  1.0000  <.0001  0.0124
## CPxMP          [0.738]  <.0001  0.3348
## MP                        [0.699]  <.0001
## MPxCP                                [0.727]
##
## Row and column labels: type
## Upper triangle: P values  adjust = "bonferroni"
## Diagonal: [Estimates] (emmean)  type = "response"

#effect size
eff_size(model_means_size, sigma = sigma(size4), edf = df.residual(size4))

```

```
## contrast      estimate      SE   df lower.CL upper.CL
## CP - CPxMP      0.290 0.251 123  -0.2071  0.787
## CP - MP         1.993 0.275 123   1.4480  2.538
## CP - MPxCP      0.789 0.255 123   0.2829  1.294
## CPxMP - MP      1.703 0.275 123   1.1596  2.247
## CPxMP - MPxCP   0.499 0.260 123  -0.0164  1.013
## MP - MPxCP     -1.205 0.264 123  -1.7266 -0.683
##
## sigma used for effect sizes: 0.02285
## Confidence level used: 0.95

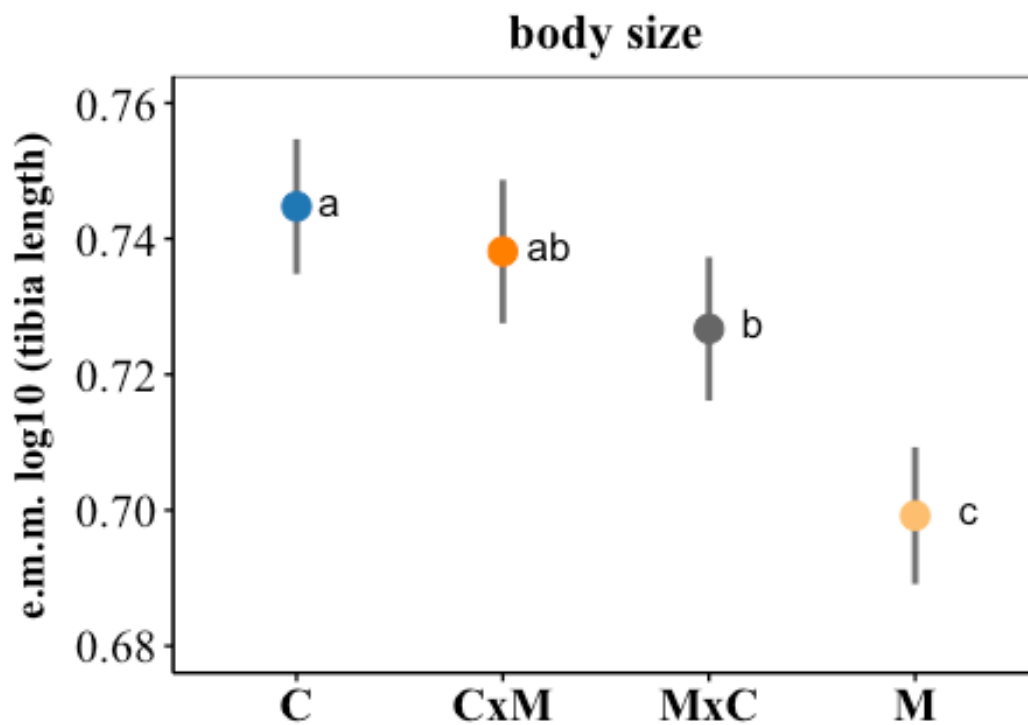
#add letters to each mean to indicate difference
model_means_cld_size <- cld(object = model_means_size,
                             reversed=T,
                             adjust="bonferroni",
                             Letters = letters,
                             alpha = 0.05)

#set marginal means at a dataframe
sizes <- as.data.frame(summary(model_means_cld_size))

#size
sizes$type <- factor(sizes$type, c("CP", "CPxMP", "MPxCP", "MP"))

ggplot(sizes, aes(type,emmean))+
  geom_errorbar(data=sizes,mapping=aes(x=type, ymin=upper.CL, ymax=lower.CL),
               color="grey45",
               width=0, linewidth=1, position = position_dodge(0.4))+
  geom_point(data=sizes, aes(type,emmean,color=type), size=4,
             position = position_dodge(0.4))+
  geom_text(aes(label = .group, y = emmean), position = position_dodge(0.4),
            vjust=0.3, hjust=-0.2, size=4.5) +
  labs(x="",y="e.m.m. log10 (tibia length)")+
  scale_y_continuous(breaks=seq(0.68, 0.76, 0.02), limits=c(0.68, 0.76))+
  scale_x_discrete(labels=c("C", "CxM", "MxC", "M"))+
  scale_color_manual(values = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"))+
  theme_update(axis.title.y=element_text(size=14,face="bold",hjust=0.5,vjust=
2.5,angle=90))+
  theme(legend.position="none", legend.key=element_blank(),legend.title=element_blank(),
        legend.text=element_text(size=10))+
  theme(plot.margin = margin(1,0.5,0.5,0.75, "cm"))+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))+
  ggtitle("body size") + theme(plot.title = element_text(hjust = 0.5, size=16,
face="bold"))
```





####FDR P-VALUE ADJUSTMENTS FROM ALL LMs####

```
pvalues <- c(0.008775,
             2.06E-06,
             2.77E-05,
             7.19E-01,
             0.004985,
             3.84E-05,
             8.61E-14,
             0.102394,
             0.0001924,
             3.81E-09,
             2.64E-07,
             0.2441719,
             4.18E-06,
             1.14E-07,
             2.20E-16,
             0.4434,
             1.15E-10,
             7.89E-01,
             6.54E-13,
             0.4432,
             2.27E-16,
```

```
0.1203373)
```

```
adjusted.p <- p.adjust(pvalues, method = "fdr", n = length(pvalues))
adjusted.p
```

```
## [1] 1.287000e-02 5.035556e-06 5.540000e-05 7.532381e-01 7.833571e-03
## [6] 7.040000e-05 6.314000e-13 1.407917e-01 3.256000e-04 1.397000e-08
## [11] 7.260000e-07 2.984323e-01 9.196000e-06 3.582857e-07 2.497000e-15
## [16] 4.877400e-01 5.060000e-10 7.890000e-01 3.597000e-12 4.877400e-01
## [21] 2.497000e-15 1.557306e-01
```

```
####COMPARISON TO PREVIOUS PAPER - SEYMOUR ET AL. (2015)####
```

```
#means per species & sex from this study
```

```
#pure species only
```

```
eye2 %>%
```

```
  group_by(type:sex) %>%
```

```
  summarise_at(vars(area), list(name = mean))
```

```
## # A tibble: 4 × 2
```

```
##   `type:sex` name
```

```
##   <fct>      <dbl>
```

```
## 1 CP:female  8.03
```

```
## 2 CP:male    8.32
```

```
## 3 MP:female  6.54
```

```
## 4 MP:male    7.10
```

```
#import data extracted from Seymoure et al. (2015)
```

```
method_comp <- read.csv("~/Documents/LMU/Eye morphology/selection manuscript/
method_comp.csv")
```

```
eye2$sex<- factor(eye2$sex, c("male", "female"))
```

```
method_comp$sex<- factor(method_comp$sex, c("male", "female"))
```

```
facet_comp <- ggplot(eye2, aes(type,whole_count))+ facet_grid(~sex)+
  geom_point(data=eye2, aes(type,whole_count), size=2.5,shape=1,color="grey40",
  ,
    position = position_jitter(0.085))+
  geom_point(data=method_comp, aes(type,whole_count), size=3, shape=17,
    position = position_jitter(0.085))+
  labs(x="",y="facet count")+
  scale_y_continuous(breaks=seq(1100, 1700, 100), limits=c(1100, 1700))+
  scale_x_discrete(labels=c("H. cydno", "H. melp"))+
  theme_update(axis.title.y=element_text(size=14,face="bold",hjust=0.5,vjust=
2.5,angle=90))+
  theme(legend.position="none", legend.key=element_blank(),legend.title=element_blank(),
    legend.text=element_text(size=10))+
  theme(plot.margin = margin(0.4,0.4,0.4,0.75, "cm"))+
  theme(axis.text.x=element_text(face = "bold"))+
```

```

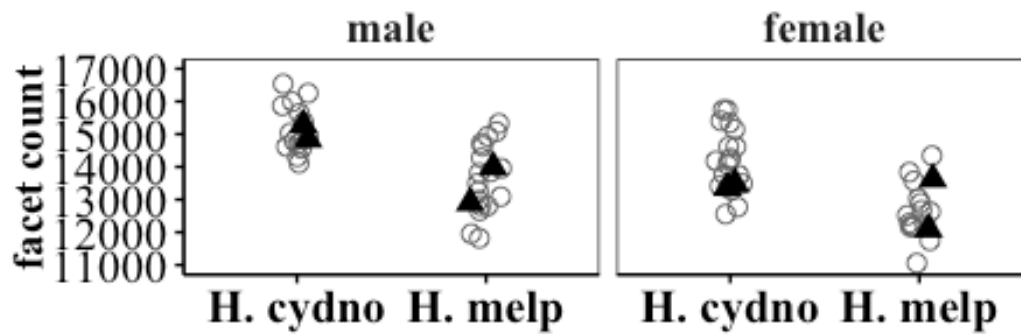
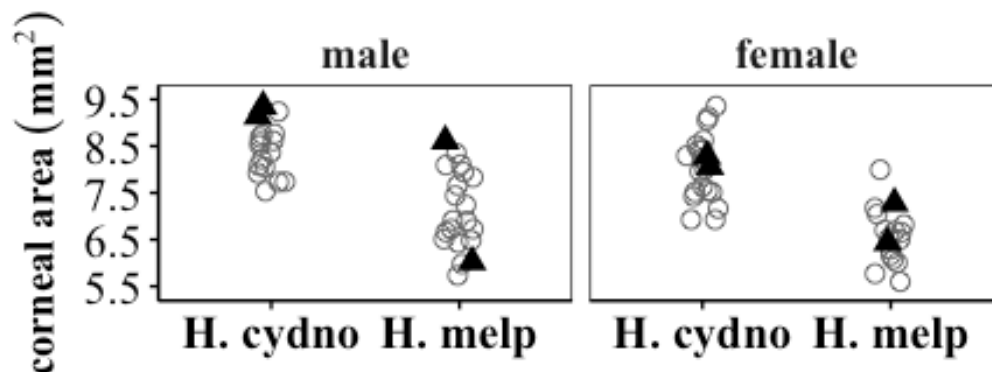
theme(strip.background = element_rect(fill = "transparent"))+
theme(strip.text = element_text(size = 14, face = "bold"))+
theme(text = element_text(family = "Times New Roman"))

area_comp <- ggplot(eye2, aes(type,area))+ facet_grid(~sex)+
  geom_point(data=eye2, aes(type,area), size=2.5,shape=1,color="grey40",
    position = position_jitter(0.085))+
  geom_point(data=method_comp, aes(type,area), size=3, shape=17,
    position = position_jitter(0.085))+
  labs(x="",y=bquote(bold('corneal area'~(mm^2))))+
  scale_y_continuous(breaks=seq(5.5, 9.5, 1.00), limits=c(5.4, 9.6))+
  scale_x_discrete(labels=c("H. cydno", "H. melp"))+
  theme_update(axis.title.y=element_text(size=14,face="bold",hjust=0.5,vjust=
2.5,angle=90))+
  theme(legend.position="none", legend.key=element_blank(),legend.title=element_blank(),
    legend.text=element_text(size=10))+
  theme(plot.margin = margin(0.4,0.4,0.4,1.08, "cm"))+
  theme(axis.text.x=element_text(face = "bold"))+
  theme(strip.background = element_rect(fill = "transparent"))+
  theme(strip.text = element_text(size = 14, face = "bold"))+
  theme(text = element_text(family = "Times New Roman"))

#combine plots
library(ggpubr)

ggarrange(facet_comp, area_comp,
  labels = c("A", "B"),
  vjust=1.23, hjust=-0.25,
  ncol = 1, nrow = 2, widths = c(1, 1))

```

**A****B**

```
#####SELECTION TESTS#####
```

```
library("Pstat")
```

```
#copy type (species) column and move to the first position
eye2.1 <- cbind(type_1 = eye2$type, eye2[, !(names(eye2) %in% "type")])
```

```
#####1) raw measurements####
```

```
raw_0.33 <- Pst(eye2.1, ci=1, csh=0.33, va=c("whole_count", "area"), boot=1000,
```

```
      Pw = c("CP", "MP"), pe = 0.95)
```

```
## Warning in dat.fra.prep(data): NAs introduced by coercion
```

```
## Warning in dat.fra.prep(data): NAs introduced by coercion
```

```
## Warning in dat.fra.prep(data): NAs introduced by coercion
```

```
## Warning in dat.fra.prep(data): NAs introduced by coercion
```

```
## Warning in dat.fra.prep(data): NAs introduced by coercion
```

```
## Warning in dat.fra.prep(data): NAs introduced by coercion
```

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33

raw_0.33$level <- c(0.33,0.33)

raw_0.67 <- Pst(eye2.1, ci=1, csh=0.67, va=c("whole_count", "area"), boot=100
0,
              Pw = c("CP","MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

raw_0.67$level <- c(0.67,0.67)

raw_1 <- Pst(eye2.1, ci=1, csh=1.0, va=c("whole_count", "area"), boot=1000,
            Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

```

```

raw_1$level <- c(1,1)

raw_1.33 <- Pst(eye2.1, ci=1, csh=1.33, va=c("whole_count", "area"), boot=100
0,
              Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33
raw_1.33$level <- c(1.33,1.33)

raw_1.5 <- Pst(eye2.1, ci=1, csh=1.5, va=c("whole_count", "area"), boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33

raw_1.5$level <- c(1.5,1.5)

raw_2 <- Pst(eye2.1, ci=1, csh=2, va=c("whole_count", "area"), boot=1000,
            Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```



[illegible]

```

## [1] "Populations sizes are:"
## CP MP
## 34 33

raw_3$level <- c(3,3)

raw_4 <- Pst(eye2.1, ci=1, csh=4, va=c("whole_count", "area"), boot=1000,
            Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

raw_4$level <- c(4,4)

raw_values <- rbind(raw_0.33, raw_0.67, raw_1, raw_1.33, raw_1.5, raw_2, raw_
3, raw_4)

#####2) Log-transformed measurements#####

```

```

log_0.33 <- Pst(eye2.1, ci=1, csh=0.33, va=c("whole_count_log", "area_log"),
boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33

log_0.33$level <- c(0.33,0.33)

log_0.67 <- Pst(eye2.1, ci=1, csh=0.67, va=c("whole_count_log", "area_log"),
boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33

log_0.67$level <- c(0.67,0.67)

log_1 <- Pst(eye2.1, ci=1, csh=1, va=c("whole_count_log", "area_log"), boot=1
000,
      Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

[illegible]

```

## [1] "Populations sizes are:"
## CP MP
## 34 33

log_1.33$level <- c(1.33,1.33)

log_1.5 <- Pst(eye2.1, ci=1, csh=1.5, va=c("whole_count_log", "area_log"), boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

log_1.5$level <- c(1.5,1.5)

log_2 <- Pst(eye2.1, ci=1, csh=2, va=c("whole_count_log", "area_log"), boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

```

[illegible]

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33

log_3$level <- c(3,3)

log_4 <- Pst(eye2.1, ci=1, csh=4, va=c("whole_count_log", "area_log"), boot=1
000,
          Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```



```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

log_4$level <- c(4,4)

log_values <- rbind(log_0.33, log_0.67, log_1, log_1.33, log_1.5, log_2, log_
3, log_4)

#####3) allometric-scaled#####

library(allomr)

#maybe different per species & sex? need to do separate?

#####CP males#####

#subset the data
eye2.1.cp.m <- eye2.1[which(eye2.1$type=="CP"),]
eye2.1.cp.m <- eye2.1.cp.m[which(eye2.1.cp.m$sex=="male"),]

#apply allometric scaling
count_al_cpm <- as.data.frame(allomr(eye2.1.cp.m$tibia_length, eye2.1.cp.m$wh
ole_count))

## Warning in (function (..., row.names = NULL, check.rows = FALSE, check.names =
## TRUE, : row names were found from a short variable and have been discarded

area_al_cpm <- as.data.frame(allomr(eye2.1.cp.m$tibia_length, eye2.1.cp.m$area))

## Warning in (function (..., row.names = NULL, check.rows = FALSE, check.names =
## TRUE, : row names were found from a short variable and have been discarded

#add scaled values back to dataset
eye2.1.cp.m2 <- cbind(eye2.1.cp.m, count_al_cpm$Yx)
colnames(eye2.1.cp.m2)[colnames(eye2.1.cp.m2) == "count_al_cpm$Yx"] <- "whole
_count_al"

```

```
eye2.1.cp.m2 <- cbind(eye2.1.cp.m2, area_al_cpm$Yx)
colnames(eye2.1.cp.m2)[colnames(eye2.1.cp.m2) == "area_al_cpm$Yx"] <- "area_al"
```

#### #####MP males####

##### *#subset the data*

```
eye2.1.mp.m <- eye2.1[which(eye2.1$type=="MP"),]
eye2.1.mp.m <- eye2.1.mp.m[which(eye2.1.mp.m$sex=="male"),]
```

##### *#apply allometric scaling*

```
count_al_mpm <- as.data.frame(allomr(eye2.1.mp.m$tibia_length, eye2.1.mp.m$whole_count))
```

```
## Warning in (function (... , row.names = NULL, check.rows = FALSE, check.names =
```

```
## TRUE, : row names were found from a short variable and have been discarded
```

```
area_al_mpm <- as.data.frame(allomr(eye2.1.mp.m$tibia_length, eye2.1.mp.m$area))
```

```
## Warning in (function (... , row.names = NULL, check.rows = FALSE, check.names =
```

```
## TRUE, : row names were found from a short variable and have been discarded
```

##### *#add scaled values back to dataset*

```
eye2.1.mp.m2 <- cbind(eye2.1.mp.m, count_al_mpm$Yx)
colnames(eye2.1.mp.m2)[colnames(eye2.1.mp.m2) == "count_al_mpm$Yx"] <- "whole_count_al"
```

```
eye2.1.mp.m2 <- cbind(eye2.1.mp.m2, area_al_mpm$Yx)
colnames(eye2.1.mp.m2)[colnames(eye2.1.mp.m2) == "area_al_mpm$Yx"] <- "area_al"
```

#### #####CP females####

##### *#subset the data*

```
eye2.1.cp.f <- eye2.1[which(eye2.1$type=="CP"),]
eye2.1.cp.f <- eye2.1.cp.f[which(eye2.1.cp.f$sex=="female"),]
```

##### *#apply allometric scaling*

```
count_al_cpf <- as.data.frame(allomr(eye2.1.cp.f$tibia_length, eye2.1.cp.f$whole_count))
```

```
## Warning in (function (... , row.names = NULL, check.rows = FALSE, check.names =
```

```
## TRUE, : row names were found from a short variable and have been discarded
```

```

area_al_cpf <- as.data.frame(allomr(eye2.1.cpf$tibia_length, eye2.1.cpf$area))

## Warning in (function (..., row.names = NULL, check.rows = FALSE, check.names =
## TRUE, : row names were found from a short variable and have been discarded

#add scaled values back to dataset
eye2.1.cpf.f2 <- cbind(eye2.1.cpf.f, count_al_cpf$Yx)
colnames(eye2.1.cpf.f2)[colnames(eye2.1.cpf.f2) == "count_al_cpf$Yx"] <- "whole_count_al"

eye2.1.cpf.f2 <- cbind(eye2.1.cpf.f2, area_al_cpf$Yx)
colnames(eye2.1.cpf.f2)[colnames(eye2.1.cpf.f2) == "area_al_cpf$Yx"] <- "area_al"

#####MP females####

#subset the data
eye2.1.mp.f <- eye2.1[which(eye2.1$type=="MP"),]
eye2.1.mp.f <- eye2.1.mp.f[which(eye2.1.mp.f$sex=="female"),]

#apply allometric scaling
count_al_mpf <- as.data.frame(allomr(eye2.1.mp.f$tibia_length, eye2.1.mp.f$whole_count))

## Warning in (function (..., row.names = NULL, check.rows = FALSE, check.names =
## TRUE, : row names were found from a short variable and have been discarded

area_al_mpf <- as.data.frame(allomr(eye2.1.mp.f$tibia_length, eye2.1.mp.f$area))

## Warning in (function (..., row.names = NULL, check.rows = FALSE, check.names =
## TRUE, : row names were found from a short variable and have been discarded

#add scaled values back to dataset
eye2.1.mp.f2 <- cbind(eye2.1.mp.f, count_al_mpf$Yx)
colnames(eye2.1.mp.f2)[colnames(eye2.1.mp.f2) == "count_al_mpf$Yx"] <- "whole_count_al"

eye2.1.mp.f2 <- cbind(eye2.1.mp.f2, area_al_mpf$Yx)
colnames(eye2.1.mp.f2)[colnames(eye2.1.mp.f2) == "area_al_mpf$Yx"] <- "area_al"

#combine datasets for analyses

```

```

eye2.2 <- (rbind(eye2.1.cp.m2, eye2.1.mp.m2, eye2.1.cp.f2, eye2.1.mp.f2))

#and now with allometric-scaled measurements
all_0.33 <- Pst(eye2.2, ci=1, csh=0.33, va=c("whole_count_al", "area_al"), boot=1000,
               Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33
all_0.33$level <- c(0.33,0.33)

all_0.67 <- Pst(eye2.2, ci=1, csh=0.67, va=c("whole_count_al", "area_al"), boot=1000,
               Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

[illegible]

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33

all_1$level <- c(1.00,1.00)

all_1.33 <- Pst(eye2.2, ci=1, csh=1.33, va=c("whole_count_al", "area_al"), bo
ot=1000,
              Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

all_1.33$level <- c(1.33,1.33)

all_1.5 <- Pst(eye2.2, ci=1, csh=1.5, va=c("whole_count_al", "area_al"), boot
=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

```

```

all_1.5$level <- c(1.50,1.50)

all_2 <- Pst(eye2.2, ci=1, csh=2, va=c("whole_count_al", "area_al"), boot=100
0,
          Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 34 33

all_2$level <- c(2.00,2.00)

all_3 <- Pst(eye2.2, ci=1, csh=3, va=c("whole_count_al", "area_al"), boot=100
0,
          Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```



[illegible]

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 34 33

all_4$level <- c(4.00,4.00)

all_values <- rbind(all_0.33, all_0.67, all_1, all_1.33, all_1.5, all_2, all_
3, all_4)

#####combine pst values into one data frame#####

pst_values <- rbind(raw_values, log_values, all_values)

####Fst values from Martin et al. 2013####

fst <- read.csv("~/Documents/LMU/Eye morphology/selection manuscript/fst_mart
in2013.csv")

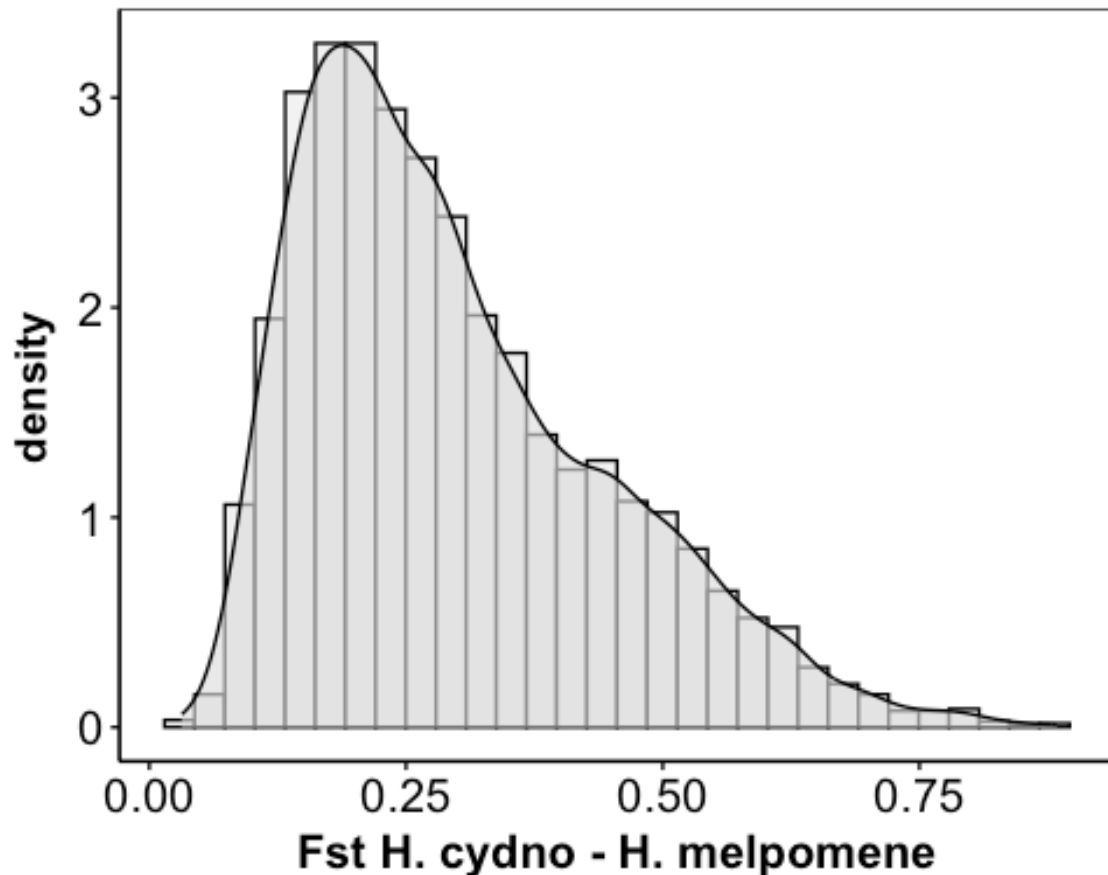
ggplot(fst,aes(chi_rosFst)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "grey95") +
  geom_density(fill="grey85", alpha=0.6)+
  labs(x= "Fst H. cydno - H. melpomene", y="density")

## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.
4.0.
## ⓘ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

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

```

```
## Warning: Removed 4308 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 4308 rows containing non-finite values (`stat_density()`).
.
```



```
#####Plotted with Pst values#####
```

```
#with only facet count
```

```
pst_values1 <- pst_values[which(pst_values$Quant_Varia=="whole_count_al"),]
```

```
quantile(fst$chi_rosFst, 0.95, na.rm=T)
```

```
##      95%
```

```
## 0.58481
```

```
pst_values1$Pst_Values
```

```
## [1] 0.8788755 0.9364345 0.9564986 0.9669353 0.9705724 0.9777657 0.9850665
```

```
## [8] 0.9887579
```

```
x.expression <- expression(bold(F[ST] ~ "H. cydno -" ~ "H. melpomene"))
```


```
count<- ggplot(fst,aes(chi_rosFst)) +
  geom_histogram(aes(y = ..density..),
```

```

    colour = "grey70", fill = "grey90") +
  geom_density(colour="grey70",fill="lightgrey", alpha=0.4)+
  geom_vline(aes(xintercept=0.8788755, color="0.33"), size=1.3, linetype="dashed",key_glyph = "rect")+
  geom_vline(aes(xintercept=0.9364345, color="0.67"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9564986, color="1.00"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9669353, color="1.33"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9705724, color="1.50"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9777657, color="2.00"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9850665, color="3.00"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9887579, color="4.00"), size=1.3, linetype="dashed")+
  scale_color_manual(
    name = expression(bold(c/h^2)),
    values = c("0.33" = "yellow", "0.67" = "orange", "1.00" = "darkorange", "1.33" = "red", "1.50" = "red2",
               "2.00" = "red3", "3.00" = "red4", "4.00" = "black"))+
  geom_vline(xintercept=0.584, linetype="dotted", color="black", size=0.75)+
  labs(x= x.expression, y="density")+
  scale_x_continuous(limits = c(0,1.0))+
  scale_y_continuous(limits = c(0,3.4), expand = c(0, 0.02))+
  theme(legend.key=element_blank(), legend.title=element_text(size=12),
        legend.title.align=0.5,legend.text=element_text(size=12))+
  theme(plot.margin = margin(0.5,0,0.5,0.5, "cm"))+
  ggtitle("facet count") + theme(plot.title = element_text(hjust = 0.5, size=15,face="bold"))+
  theme(text = element_text(family = "Times New Roman"))

```

```

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
##  Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

*#with only area*

```
pst_values2 <- pst_values[which(pst_values$Quant_Varia=="area_a1"),]
```

```
quantile(fst$chi_rosFst, 0.95, na.rm=T)
```

```
##      95%
```

```
## 0.58481
```

```
pst_values2$Pst_Values
```

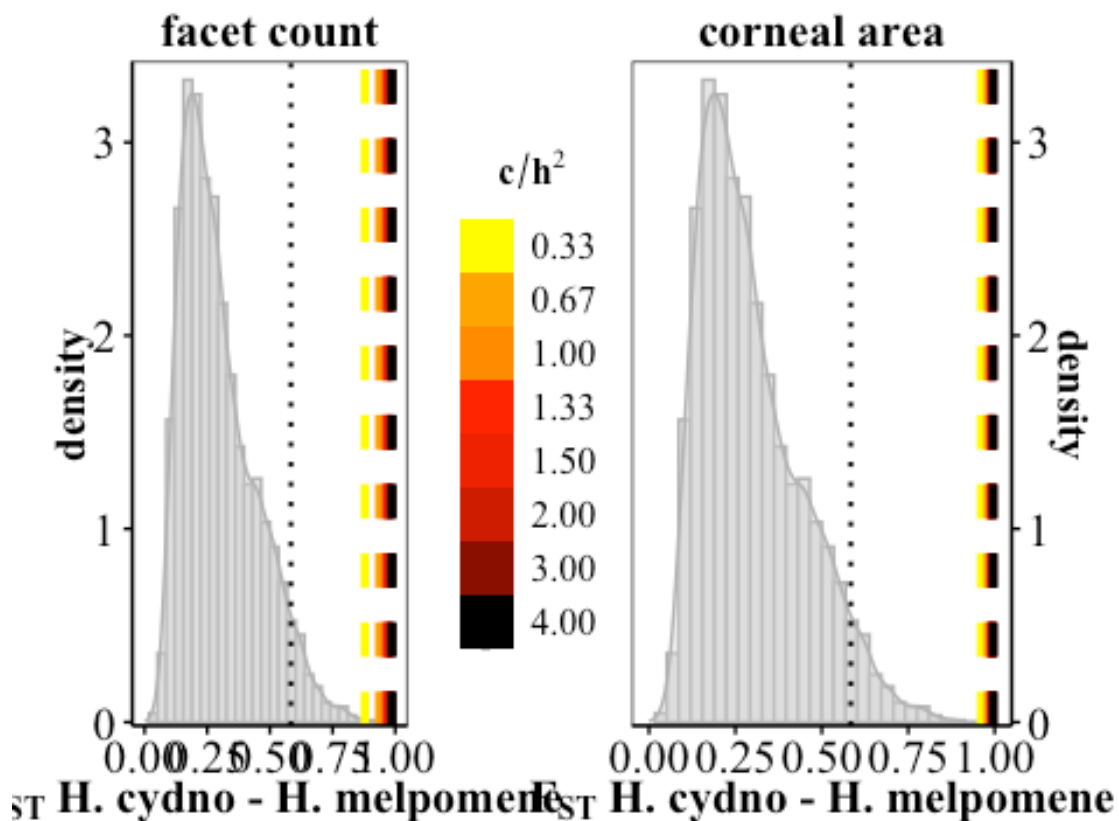
```
## [1] 0.9608007 0.9803010 0.9867153 0.9899785 0.9911042 0.9933132 0.9955322
## [8] 0.9966454
```

```
area <- ggplot(fst,aes(chi_rosFst)) +
  geom_histogram(aes(y = ..density..),
    colour = "grey70", fill = "grey90") +
  geom_density(colour="grey70",fill="lightgrey", alpha=0.4)+
  geom_vline(aes(xintercept=0.9608007, color="0.33"), size=1.3, linetype="dashed",key_glyph = "rect")+
  geom_vline(aes(xintercept=0.9803010, color="0.67"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9867153, color="1.00"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9899785, color="1.33"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9911042, color="1.50"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9933132, color="2.00"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9955322, color="3.00"), size=1.3, linetype="dashed")+
  geom_vline(aes(xintercept=0.9966454, color="4.00"), size=1.3, linetype="dashed")+
  scale_color_manual(
    name = expression(bold(c/h^2)),
    values = c("0.33" = "yellow", "0.67" = "orange", "1.00" = "darkorange", "1.33" = "red", "1.50" = "red2",
      "2.00" = "red3", "3.00" = "red4", "4.00" = "black"))+
  geom_vline(xintercept=0.584, linetype="dotted", color="black", size=0.75)+
  labs(x= x.expression, y="density")+
  scale_x_continuous(limits = c(0,1.0))+
  scale_y_continuous(limits = c(0,3.4), expand = c(0, 0.02),position = "right") +
  theme(legend.key=element_blank(), legend.title=element_text(size=12),
    legend.title.align=0.5,legend.text=element_text(size=12))+
  theme(legend.position='none')+
  theme(plot.margin = margin(0.5,0.5,0.5,0.1, "cm"))+
  ggtitle("corneal area") + theme(plot.title = element_text(hjust = 0.5, size =15,face="bold"))+
  theme(text = element_text(family = "Times New Roman"))
```

```
#combine plots
library(ggpubr)
```

```
ggarrange(count, area,
  #labels = c("A", "B"),
  #vjust=1.23, hjust=0.8,
  ncol = 2, nrow = 1, widths = c(1.2, 1))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 4308 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 4308 rows containing non-finite values (`stat_density()`).
.
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



```
#####calculate p-values for each Pst####

#use only cydno vs. melp fst values
fst2 <- as.data.frame(fst$chi_rosFst)

#remove NAs from the Fst values
fst2.1 <- complete.cases(fst2)
fst3 <- fst2[fst2.1, ]

#calculate the proportion of the Fst distribution above each Pst value
proportions <- lapply(pst_values$Pst_Values, function(x) sum(fst3 > x) / length(fst3))
```

```

#combine the x_values and proportions into a data frame
p.values <- data.frame(x = pst_values$Pst_Values, p_value = unlist(proportion
s))

pst_values <- cbind(pst_values, p.values)

#double-check that Pst values are aligned in new dataset
pst_values$Pst_Values - pst_values$x #all zeros, everything is correct

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
## [39] 0 0 0 0 0 0 0 0 0 0

#export to make a table
write.csv(pst_values, file='~/Documents/LMU/Eye morphology/selection manuscrip
t/pst_table.csv')

####SELECTION TESTS PER SEX SEPARATELY####

library("Pstat")

eye3m <- eye2.1[which(eye2.1$sex=="male"),]

#####1) raw measurements_male####
raw_0.33_m <- Pst(eye3m, ci=1, csh=0.33, va=c("whole_count", "area"), boot=10
00,
               Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

[illegible]



```

## [1] "Populations sizes are:"
## CP MP
## 15 18

raw_0.67_m$level <- c(0.67,0.67)

raw_1_m <- Pst(eye3m, ci=1, csh=1.0, va=c("whole_count", "area"), boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 15 18

raw_1_m$level <- c(1,1)

raw_1.33_m <- Pst(eye3m, ci=1, csh=1.33, va=c("whole_count", "area"), boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion

```

[illegible]

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

raw_1.5_m$level <- c(1.5,1.5)

raw_2_m <- Pst(eye3m, ci=1, csh=2, va=c("whole_count", "area"), boot=1000,
  Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

raw_2_m$level <- c(2,2)

raw_3_m <- Pst(eye3m, ci=1, csh=3, va=c("whole_count", "area"), boot=1000,
  Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

raw_3_m$level <- c(3,3)

```

```

raw_4_m <- Pst(eye3m, ci=1, csh=4, va=c("whole_count", "area"), boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

raw_4_m$level <- c(4,4)

raw_values_m <- rbind(raw_0.33_m, raw_0.67_m, raw_1_m,
                     raw_1.33_m, raw_1.5_m, raw_2_m, raw_3_m, raw_4_m)

#####2) Log-transformed measurements_male####
log_0.33_m <- Pst(eye3m, ci=1, csh=0.33, va=c("whole_count_log", "area_log"),
                 boot=1000,
                 Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion

```

[illegible]

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

log_0.67_m$level <- c(0.67,0.67)

log_1_m <- Pst(eye3m, ci=1, csh=1, va=c("whole_count_log", "area_log"), boot=
1000,
      Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 15 18

log_1_m$level <- c(1,1)

log_1.33_m <- Pst(eye3m, ci=1, csh=1.33, va=c("whole_count_log", "area_log"),
boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

```



```

log_1.33_m$level <- c(1.33,1.33)

log_1.5_m <- Pst(eye3m, ci=1, csh=1.5, va=c("whole_count_log", "area_log"), boot=1000,
                Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

log_1.5_m$level <- c(1.5,1.5)

log_2_m <- Pst(eye3m, ci=1, csh=2, va=c("whole_count_log", "area_log"), boot=1000,
                Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

[illegible]

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

log_3_m$level <- c(3,3)

log_4_m <- Pst(eye3m, ci=1, csh=4, va=c("whole_count_log", "area_log"), boot=
1000,
             Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

[illegible]

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18
all_0.33_m$level <- c(0.33,0.33)

all_0.67_m <- Pst(eye3m_al, ci=1, csh=0.67, va=c("whole_count_al", "area_al"),
, boot=1000,
      Pw = c("CP","MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18
all_0.67_m$level <- c(0.67,0.67)

all_1_m <- Pst(eye3m_al, ci=1, csh=1, va=c("whole_count_al", "area_al"), boot

```

```

=1000,
      Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

all_1_m$level <- c(1,1)

all_1.33_m <- Pst(eye3m_al, ci=1, csh=1.33, va=c("whole_count_al", "area_al"),
, boot=1000,
      Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

all_1.33_m$level <- c(1.33,1.33)

all_1.5_m <- Pst(eye3m_al, ci=1, csh=1.5, va=c("whole_count_al", "area_al"),
boot=1000,
      Pw = c("CP","MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

[illegible]



```

## [1] "Populations sizes are:"
## CP MP
## 15 18

all_2_m$level <- c(2,2)

all_3_m <- Pst(eye3m_al, ci=1, csh=3, va=c("whole_count_al", "area_al"), boot
=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 15 18

all_3_m$level <- c(3,3)

all_4_m <- Pst(eye3m_al, ci=1, csh=4, va=c("whole_count_al", "area_al"), boot
=1000,
              Pw = c("CP", "MP"), pe = 0.95)

```

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 15 18

all_4_m$level <- c(4,4)

all_values_m <- rbind(all_0.33_m, all_0.67_m, all_1_m,
                     all_1.33_m, all_1.5_m, all_2_m, all_3_m, all_4_m)

#####combine pst values into one male data frame with p-values####

pst_values_m <- rbind(raw_values_m, log_values_m, all_values_m)

#calculate p-values

#use only cydno vs. melp fst values, fst3

#calculate the proportion of the Fst distribution above each Pst value
proportions_m <- lapply(pst_values_m$Pst_Values, function(x) sum(fst3 > x) /

```

```
length(fst3))

# Combine the x_values and proportions into a data frame
p.values_m <- data.frame(x = pst_values_m$Pst_Values, p_value = unlist(proportions_m))

pst_values_m <- cbind(pst_values_m, p.values_m)

#double-check that Pst values are aligned in new dataset
pst_values_m$Pst_Values - pst_values_m$x #all zeros, everything is correct

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0 0 0 0 0

#export to make a table
write.csv(pst_values_m, file='~/Documents/LMU/Eye morphology/selection manuscript/pst_table_male.csv')

#####1) raw measurements_female#####

eye3f <- eye2.1[which(eye2.1$sex=="female"),]

raw_0.33_f <- Pst(eye3f, ci=1, csh=0.33, va=c("whole_count", "area"), boot=1000,
                  Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

raw_0.33_f$level <- c(0.33,0.33)

raw_0.67_f <- Pst(eye3f, ci=1, csh=0.67, va=c("whole_count", "area"), boot=100,
                  Pw = c("CP","MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

```

```

raw_0.67_f$level <- c(0.67,0.67)

raw_1.0_f <- Pst(eye3f, ci=1, csh=1.0, va=c("whole_count", "area"), boot=1000
,
          Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

raw_1.0_f$level <- c(1.0,1.0)

raw_1.33_f <- Pst(eye3f, ci=1, csh=1.33, va=c("whole_count", "area"), boot=1000,
          Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

[illegible]

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

raw_1.5_f$level <- c(1.5,1.5)

raw_2.0_f <- Pst(eye3f, ci=1, csh=2.0, va=c("whole_count", "area"), boot=1000
,
      Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15
raw_2.0_f$level <- c(2.0,2.0)

raw_3.0_f <- Pst(eye3f, ci=1, csh=3.0, va=c("whole_count", "area"), boot=1000
,
               Pw = c("CP","MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15
raw_3.0_f$level <- c(3.0,3.0)

```



```

raw_4.0_f <- Pst(eye3f, ci=1, csh=4.0, va=c("whole_count", "area"), boot=1000
,
          Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

raw_4.0_f$level <- c(4.0,4.0)

raw_values_f <- rbind(raw_0.33_f, raw_0.67_f, raw_1.0_f, raw_1.33_f, raw_1.5_
f,
          raw_2.0_f, raw_3.0_f, raw_4.0_f)

#####2) Log-transformed measurements_female####
log_0.33_f <- Pst(eye3f, ci=1, csh=0.33, va=c("whole_count_log", "area_log"),
boot=1000,
          Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion

```

[illegible]

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

log_0.67_f$level <- c(0.67,0.67)

log_1.0_f <- Pst(eye3f, ci=1, csh=1.0, va=c("whole_count_log", "area_log"), b
oot=1000,
               Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 19 15

log_1.0_f$level <- c(1.0,1.0)

log_1.33_f <- Pst(eye3f, ci=1, csh=1.33, va=c("whole_count_log", "area_log"),
boot=1000,
              Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

```

```

log_1.33_f$level <- c(1.33,1.33)

log_1.5_f <- Pst(eye3f, ci=1, csh=1.5, va=c("whole_count_log", "area_log"), boot=1000,
                Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

log_1.5_f$level <- c(1.5,1.5)

log_2.0_f <- Pst(eye3f, ci=1, csh=2, va=c("whole_count_log", "area_log"), boot=1000,
                Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

[illegible]

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

log_3.0_f$level <- c(3.0,3.0)

log_4.0_f <- Pst(eye3f, ci=1, csh=4, va=c("whole_count_log", "area_log"), boot=1000,
                Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 19 15

log_4.0_f$level <- c(4.0,4.0)

log_values_f <- rbind(log_0.33_f, log_0.67_f, log_1.0_f, log_1.33_f, log_1.5_
f,
                    log_2.0_f, log_3.0_f, log_4.0_f)

#####3) allometric-scaled_female#####

eye3f_al <- (rbind(eye2.1.cp.f2, eye2.1.mp.f2))

#with allomertic-scaled measurements
all_0.33_f <- Pst(eye3f_al, ci=1, csh=0.33, va=c("whole_count_al", "area_al")
, boot=1000,
                Pw = c("CP","MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```



```

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 19 15

all_0.33_f$level <- c(0.33,0.33)

all_0.67_f <- Pst(eye3f_al, ci=1, csh=0.67, va=c("whole_count_al", "area_al")
, boot=1000,
               Pw = c("CP","MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

all_0.67_f$level <- c(0.67,0.67)

```

```

all_1.0_f <- Pst(eye3f_al, ci=1, csh=1.0, va=c("whole_count_al", "area_al"),
boot=1000,
                Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

all_1.0_f$level <- c(1.0,1.0)

all_1.33_f <- Pst(eye3f_al, ci=1, csh=1.33, va=c("whole_count_al", "area_al")
, boot=1000,
                Pw = c("CP", "MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion

```

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

all_1.33_f$level <- c(1.33,1.33)

all_1.5_f <- Pst(eye3f_al, ci=1, csh=1.5, va=c("whole_count_al", "area_al"),
boot=1000,
          Pw = c("CP","MP"), pe = 0.95)

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

```

[illegible]

```

## [1] "Populations sizes are:"
## CP MP
## 19 15

all_2.0_f$level <- c(2.0,2.0)

all_3.0_f <- Pst(eye3f_al, ci=1, csh=3, va=c("whole_count_al", "area_al"), bot=1000,
                Pw = c("CP", "MP"), pe = 0.95)
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced

## [1] "Populations sizes are:"
## CP MP
## 19 15

all_3.0_f$level <- c(3.0,3.0)

all_4.0_f <- Pst(eye3f_al, ci=1, csh=4, va=c("whole_count_al", "area_al"), bot=1000,
                Pw = c("CP", "MP"), pe = 0.95)

```

```

## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in dat.fra.prep(data): NAs introduced by coercion
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## Warning in sqrt((nna.clm - 1)/nna.clm): NaNs produced
## [1] "Populations sizes are:"
## CP MP
## 19 15

all_4.0_f$level <- c(4.0,4.0)

all_values_f <- rbind(all_0.33_f, all_0.67_f, all_1.0_f, all_1.33_f, all_1.5_
f,
                    all_2.0_f, all_3.0_f, all_4.0_f)

#####combine pst values into one female data frame with p-values#####

pst_values_f <- rbind(raw_values_f, log_values_f, all_values_f)

#calculate p-values
#use only cydno vs. melp fst values, fst3

#calculate the proportion of the Fst distribution above each Pst value
proportions_f <- lapply(pst_values_f$Pst_Values, function(x) sum(fst3 > x) /
length(fst3))

```

```

# Combine the x_values and proportions into a data frame
p.values_f <- data.frame(x = pst_values_f$Pst_Values, p_value = unlist(proportions_f))

pst_values_f <- cbind(pst_values_f, p.values_f)

#double-check that Pst values are aligned in new dataset
pst_values_f$Pst_Values - pst_values_f$x #all zeros, everything is correct

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
## [39] 0 0 0 0 0 0 0 0 0 0

#export to make a table
write.csv(pst_values_f, file='~/Documents/LMU/Eye morphology/selection manuscript/pst_table_female.csv')

####ALLOMETRY####
library(smatr)

#####1) as an interactive term####

#smatr cannot handle 2 grouping terms, so make an interaction
eye2$inter <- interaction(eye2$type, eye2$sex)

eye2$inter<- factor(eye2$inter, c("CP.male", "CP.female",
                                "MP.male", "MP.female"))

#####testing differences in slope (beta)#####

#facet count
a1 <- sma(whole_count_log~tibia_length_log*inter, robust=F, data=eye2, multcomp = T, multcompmethod = "adjust")
multcompmatrix(a1, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               inter_2
## inter_1    CP.male MP.male MP.female CP.female
## CP.male    -
## MP.male      -
## MP.female      -
## CP.female      -

summary(a1)

```

```

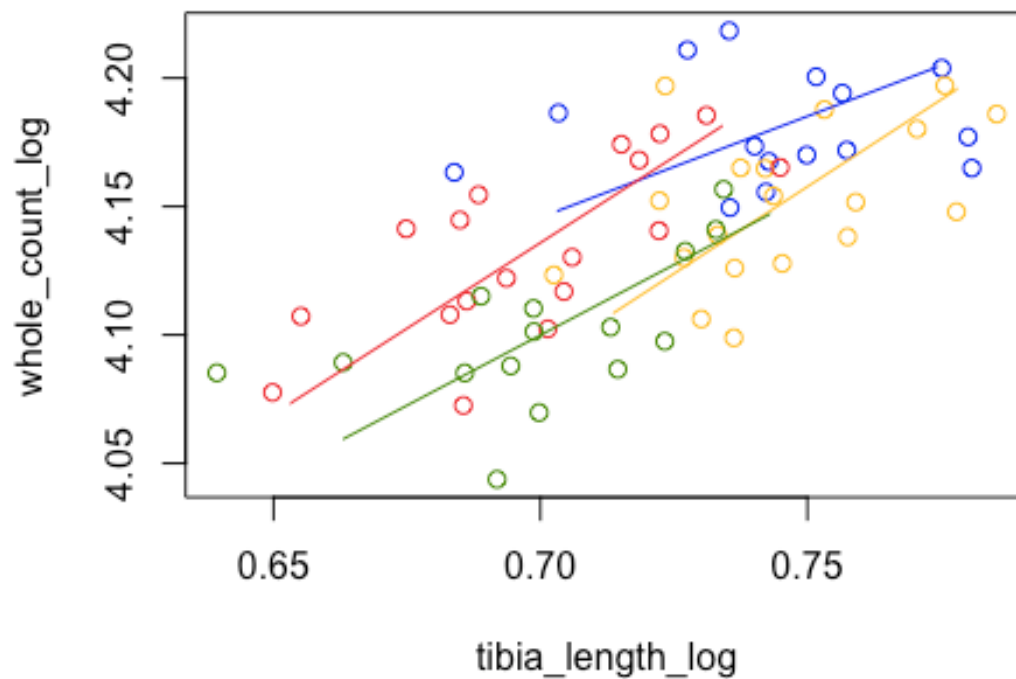
## Call: sma(formula = whole_count_log ~ tibia_length_log * inter, data = eye
2,
##      multcomp = T, multcompmethod = "adjust", robust = F)
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 2.981 with 3 degrees of freedom
## P-value : 0.39454
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in slope :
##      inter_1  inter_2    Pval  TestStat
## 1   CP.male  CP.female 0.56709 2.28963193
## 2   CP.male   MP.male 0.52587 2.45762825
## 3   CP.male MP.female 0.93682 0.80736426
## 4 CP.female   MP.male 1.00000 0.00278065
## 5 CP.female MP.female 0.98537 0.44349039
## 6   MP.male MP.female 0.98456 0.45284640
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##           elevation      slope
## estimate      3.594643 0.7873019
## lower limit    3.243936 0.4465518
## upper limit    3.945351 1.3880679
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: CP.female
##           elevation      slope
## estimate      3.142194 1.3540100
## lower limit    2.688987 0.8763565
## upper limit    3.595402 2.0920060
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##
## Group: MP.male
##           elevation      slope

```



```
## estimate      3.201911 1.3342359
## lower limit   2.858373 0.9302675
## upper limit   3.545450 1.9136273
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845
##
## Group: MP.female
##           elevation      slope
## estimate      3.333263 1.0952430
## lower limit    2.953909 0.6804684
## upper limit    3.712617 1.7628405
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798
```

```
plot(a1)
```



```
#####*plot facet count major axis regressions*#####
dev.off()
```

```

## null device
##          1

par(mgp=c(3.5,0.6,0))
par(mar = c(5, 6, 1.25, 1.25))
par(lwd=1)

plot(a1, col=c("#1F78B4", "#1F78B4", "#FDBF6F", "#FDBF6F"),
     pch=c(17,2,16,1),
     lwd=1,
     cex=0.9,
     lty=c(1,2,1,2),
     xlim=c(0.635,0.80), ylim=c(4.03,4.25),
     las=1,cex.axis = 1.3,
     xlab = "",
     ylab = "",
     tck=-0.015,
     family = "Times New Roman")

#title(xlab = "log10 (hind tibia length)", line = 2,cex.lab = 1.2,font.lab=2,
family = "Times New Roman")
title(ylab = "log10 (facet count)", line = 3,cex.lab = 1.2,font.lab=2,family
= "Times New Roman")

title("facet count", adj = 0.5, line = 0.6, family = "Times New Roman", cex.m
ain=1.35)

legend("topleft", c("C", "M"),
      col = c("#1F78B4", "#FDBF6F"), pch=c(17,16),
      bty="n")

#area
a2 <- sma(area_log~tibia_length_log*inter, data=eye2, multcomp = T, multcompm
ethod = "adjust")
multcompmatrix(a2, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          inter_2
## inter_1    CP.male MP.male CP.female MP.female
## CP.male    -      *      .
## MP.male    *      -
## CP.female  .      -
## MP.female          -

summary(a2)

## Call: sma(formula = area_log ~ tibia_length_log * inter, data = eye2,
##          multcomp = T, multcompmethod = "adjust")

```

```

##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 8.026 with 3 degrees of freedom
## P-value : 0.045487
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in slope :
##      inter_1  inter_2    Pval  TestStat
## 1  CP.male CP.female 0.079429 6.07655231
## 2  CP.male  MP.male 0.038657 7.39266161
## 3  CP.male MP.female 0.416808 2.94858246
## 4 CP.female  MP.male 0.999985 0.03885275
## 5 CP.female MP.female 0.973795 0.55816819
## 6  MP.male MP.female 0.910225 0.94558492
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##      elevation    slope
## estimate    0.22086518 0.9387334
## lower limit -0.08562769 0.6134198
## upper limit  0.52735804 1.4365697
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
## Group: CP.female
##      elevation    slope
## estimate    -0.44638368 1.811085
## lower limit -0.86747180 1.332310
## upper limit -0.02529556 2.461912
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736
## P-value : 5.157e-05
##
## Group: MP.male
##      elevation    slope
## estimate    -0.4652718 1.881835
## lower limit -0.8042493 1.458194
## upper limit -0.1262944 2.428555

```

```

##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06
##
## Group: MP.female
##           elevation    slope
## estimate    -0.2501853 1.518812
## lower limit -0.6572024 1.045336
## upper limit  0.1568317 2.206746
##
## H0 : variables uncorrelated.
## R-squared : 0.5928163
## P-value : 0.00078631

plot(a2)

#####**plot corneal area major axis regressions**#####

plot(a2, col=c("#1F78B4", "#1F78B4", "#FDBF6F", "#FDBF6F"),
      pch=c(17,2,16,1),
      lwd=1,
      cex=0.9,
      lty=c(1,2,1,2),
      xlim=c(0.635,0.80), ylim=c(0.73,1.00),
      las=1,cex.axis = 1.3,
      xlab = "",
      ylab = "",
      tck=-0.015,
      family = "Times New Roman")

title(xlab = "log10 (hind tibia length)", line = 2,cex.lab = 1.2,font.lab=2,family = "Times New Roman")

title(ylab = "log10 (corneal area)", line = 3,cex.lab = 1.2,font.lab=2,family = "Times New Roman")

title("corneal area", adj = 0.5, line = 0.6, family = "Times New Roman", cex.main=1.35)

legend("topleft", c("C", "M"),
      col = c("#1F78B4", "#FDBF6F"), pch=c(17,16),
      bty="n")

#####testing shift in elevation (grade shift - a)#####

#facet count

```

```

a3 <- sma(whole_count_log~tibia_length_log+inter, type="elevation",
          data=eye2, multcomp = T, multcompmethod = "adjust")
multcompmatrix(a3, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           inter_2
## inter_1    CP.male CP.female MP.female MP.male
## CP.male    -        *          *
## CP.female  *        -          **
## MP.female  *          -          **
## MP.male    **        **         -

summary(a3)

## Call: sma(formula = whole_count_log ~ tibia_length_log + inter, data = eye
2,
##      type = "elevation", multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 2.981 with 3 degrees of freedom
## P-value : 0.39454
## -----
##
## H0 : no difference in elevation.
## Wald statistic: 23.21 with 3 degrees of freedom
## P-value : 3.6441e-05
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in elevation :
##      inter_1  inter_2      Pval   TestStat
## 1  CP.male CP.female 0.0428279   7.2053170
## 2  CP.male  MP.male 0.9988230   0.1757316
## 3  CP.male MP.female 0.0210016   8.5103664
## 4 CP.female  MP.male 0.0019596  12.9098131
## 5 CP.female MP.female 0.9998065   0.0937200
## 6  MP.male MP.female 0.0012326  13.7796832
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male

```

```

##           elevation      slope
## estimate      3.298465 1.1853030
## lower limit    3.092715 0.9503365
## upper limit    3.504216 1.4802397
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: CP.female
##           elevation      slope
## estimate      3.267918 1.1853030
## lower limit    3.067266 0.9503365
## upper limit    3.468570 1.4802397
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##
## Group: MP.male
##           elevation      slope
## estimate      3.305897 1.1853030
## lower limit    3.117061 0.9503365
## upper limit    3.494733 1.4802397
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845
##
## Group: MP.female
##           elevation      slope
## estimate      3.270185 1.1853030
## lower limit    3.076896 0.9503365
## upper limit    3.463475 1.4802397
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798

plot(a3)

#area
a4 <- sma(area_log~tibia_length_log+inter, type="elevation",
          data=eye2, multcomp = T, multcompmethod = "adjust" )
multcompmatrix(a4, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           inter_2

```

```
## inter_1      CP.male CP.female MP.female MP.male
##   CP.male    -                ***
##   CP.female      -                *
##   MP.female ***                -    ***
##   MP.male        *                ***    -
```

`summary(a4)`

```
## Call: sma(formula = area_log ~ tibia_length_log + inter, data = eye2,
##   type = "elevation", multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 8.026 with 3 degrees of freedom
## P-value : 0.045487
## -----
##
## H0 : no difference in elevation.
## Wald statistic: 20.12 with 3 degrees of freedom
## P-value : 0.00016011
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in elevation :
##   inter_1  inter_2      Pval   TestStat
## 1  CP.male CP.female 0.31335412  3.51723843
## 2  CP.male  MP.male 0.99997969  0.04348447
## 3  CP.male MP.female 8.3887e-05 18.87163877
## 4 CP.female  MP.male 0.03208414  7.73359385
## 5 CP.female MP.female 0.84596688  1.22774439
## 6  MP.male MP.female 0.00035567 16.12526659
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##           elevation      slope
## estimate      -0.30393283 1.643953
## lower limit -0.51342742 1.383356
## upper limit -0.09443825 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
```

```

## Group: CP.female
##           elevation    slope
## estimate    -0.3218331 1.643953
## lower limit -0.5262659 1.383356
## upper limit -0.1174003 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736
## P-value : 5.157e-05
##
## Group: MP.male
##           elevation    slope
## estimate    -0.2991812 1.643953
## lower limit -0.4916616 1.383356
## upper limit -0.1067007 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06
##
## Group: MP.female
##           elevation    slope
## estimate    -0.3378333 1.643953
## lower limit -0.5348932 1.383356
## upper limit -0.1407734 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.5928163
## P-value : 0.00078631

plot(a4)

#####testing shift along common axis#####

#facet count
a5 <- sma(whole_count_log~tibia_length_log+inter, type="shift",
          data=eye2, multcomp = T, multcompmethod = "adjust")
multcompmatrix(a5, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           inter_2
## inter_1    CP.male MP.male MP.female CP.female
## CP.male    -      ***      ***
## MP.male    ***    -
## MP.female  ***      -      ***
## CP.female      ***      ***    -

```



```
summary(a5)
```

```
## Call: sma(formula = whole_count_log ~ tibia_length_log + inter, data = eye
2,
##      type = "shift", multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 2.981 with 3 degrees of freedom
## P-value : 0.39454
## -----
##
## H0 : no shift along common axis.
## Wald statistic: 62.68 with 3 degrees of freedom
## P-value : 1.5743e-13
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in shift :
##      inter_1  inter_2      Pval  TestStat
## 1  CP.male CP.female 0.28798033  3.680999
## 2  CP.male  MP.male 9.7151e-07 27.441499
## 3  CP.male MP.female 7.5420e-12 50.395166
## 4 CP.female  MP.male 0.00074106 14.737750
## 5 CP.female MP.female 8.8920e-07 27.612733
## 6  MP.male MP.female 0.58922893  2.202177
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##      elevation      slope
## estimate      3.298465 1.1853030
## lower limit  3.243936 0.9503365
## upper limit  3.945351 1.4802397
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: CP.female
##      elevation      slope
## estimate      3.267918 1.1853030
## lower limit  2.688987 0.9503365
## upper limit  3.595402 1.4802397
```

```
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##
## Group: MP.male
##           elevation      slope
## estimate    3.305897 1.1853030
## lower limit  2.858373 0.9503365
## upper limit  3.545450 1.4802397
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845
##
## Group: MP.female
##           elevation      slope
## estimate    3.270185 1.1853030
## lower limit  2.953909 0.9503365
## upper limit  3.712617 1.4802397
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798

plot(a5)

#area
a6 <- sma(area_log~tibia_length_log+inter, type="shift",
          data=eye2, multcomp = T, multcompmethod = "adjust")
multcompmatrix(a6, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           inter_2
## inter_1    CP.male MP.male MP.female CP.female
## CP.male    -      ***    ***
## MP.male    ***    -      ***
## MP.female  ***    -      ***
## CP.female      ***    ***    -

summary(a6)

## Call: sma(formula = area_log ~ tibia_length_log + inter, data = eye2,
##           type = "shift", multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
```

```

## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 8.026 with 3 degrees of freedom
## P-value : 0.045487
## -----
##
## H0 : no shift along common axis.
## Wald statistic: 62.94 with 3 degrees of freedom
## P-value : 1.3811e-13
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in shift :
##      inter_1  inter_2      Pval  TestStat
## 1  CP.male CP.female    0.98634  0.4318941
## 2  CP.male  MP.male 1.1129e-06 27.1788341
## 3  CP.male MP.female 1.7689e-10 44.2110534
## 4  CP.female  MP.male 9.4240e-06 23.0594591
## 5  CP.female MP.female 2.9485e-08 34.2231172
## 6  MP.male MP.female    0.88670  1.0548864
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##           elevation    slope
## estimate    -0.30393283 1.643953
## lower limit -0.08562769 1.383356
## upper limit  0.52735804 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
## Group: CP.female
##           elevation    slope
## estimate    -0.32183311 1.643953
## lower limit -0.86747180 1.383356
## upper limit -0.02529556 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736
## P-value : 5.157e-05
##
## Group: MP.male
##           elevation    slope
## estimate    -0.2991812 1.643953
## lower limit -0.8042493 1.383356

```

```

## upper limit -0.1262944 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06
##
## Group: MP.female
##           elevation      slope
## estimate    -0.3378333 1.643953
## lower limit -0.6572024 1.383356
## upper limit  0.1568317 1.947097
##
## H0 : variables uncorrelated.
## R-squared : 0.5928163
## P-value : 0.00078631

plot(a6)

#####2) as a single term, separated by sex####

#divide datasets by sex
eye2.m <- eye2[which(eye2$sex=="male"),]
eye2.f <- eye2[which(eye2$sex=="female"),]

eye.m <- eye[which(eye$sex=="male"),]
eye.f <- eye[which(eye$sex=="female"),]

#####testing differences in slope (beta)#####

#facet count - male
b1 <- sma(whole_count_log~tibia_length_log*type, data=eye2.m)
summary(b1)

## Call: sma(formula = whole_count_log ~ tibia_length_log * type, data = eye2
.m)
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 2.458 with 1 degrees of freedom
## P-value : 0.11696
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP

```

```

##           elevation      slope
## estimate      3.594643 0.7873019
## lower limit    3.243936 0.4465518
## upper limit    3.945351 1.3880679
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: MP
##           elevation      slope
## estimate      3.201911 1.3342359
## lower limit    2.858373 0.9302675
## upper limit    3.545450 1.9136273
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845

plot(b1)

#facet count - female
b1.1 <- sma(whole_count_log~tibia_length_log*type, data=eye2.f)
summary(b1.1)

## Call: sma(formula = whole_count_log ~ tibia_length_log * type, data = eye2
.f)
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 0.4435 with 1 degrees of freedom
## P-value : 0.50544
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation      slope
## estimate      3.142194 1.3540100
## lower limit    2.688987 0.8763565
## upper limit    3.595402 2.0920060
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##

```

```

## Group: MP
##           elevation      slope
## estimate    3.333263 1.0952430
## lower limit  2.953909 0.6804684
## upper limit  3.712617 1.7628405
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798

plot(b1.1)

par(mfrow = c(1, 2))
plot(b1,main = "males")
plot(b1.1,main = "females")
par(mfrow = c(1, 1))

#area - male
b2 <- sma(area_log~tibia_length_log*type, data=eye2.m)
summary(b2)

## Call: sma(formula = area_log ~ tibia_length_log * type, data = eye2.m)
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 7.393 with 1 degrees of freedom
## P-value : 0.0065491
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation      slope
## estimate    0.22086518 0.9387334
## lower limit -0.08562769 0.6134198
## upper limit  0.52735804 1.4365697
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
## Group: MP
##           elevation      slope
## estimate   -0.4652718 1.881835
## lower limit -0.8042493 1.458194
## upper limit -0.1262944 2.428555

```

```
##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06

plot(b2)

#area - female
b2.1 <- sma(area_log~tibia_length_log*type, data=eye2.f)
summary(b2.1)

## Call: sma(formula = area_log ~ tibia_length_log * type, data = eye2.f)
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 0.5582 with 1 degrees of freedom
## P-value : 0.455
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##          elevation    slope
## estimate    -0.44638368 1.811085
## lower limit -0.86747180 1.332310
## upper limit -0.02529556 2.461912
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736
## P-value : 5.157e-05
##
## Group: MP
##          elevation    slope
## estimate    -0.2501853 1.518812
## lower limit -0.6572024 1.045336
## upper limit  0.1568317 2.206746
##
## H0 : variables uncorrelated.
## R-squared : 0.5928163
## P-value : 0.00078631

plot(b2.1)

#####testing shift in elevation (grade shift - a)#####
```

```

#facet count - male
b3 <- sma(whole_count_log~tibia_length_log+type, type="elevation", data=eye2.
m)
summary(b3)

## Call: sma(formula = whole_count_log ~ tibia_length_log + type, data = eye2
.m,
##      type = "elevation")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 2.458 with 1 degrees of freedom
## P-value : 0.11696
## -----
##
## H0 : no difference in elevation.
## Wald statistic: 0.1757 with 1 degrees of freedom
## P-value : 0.67507
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##          elevation      slope
## estimate      3.326171 1.1480723
## lower limit    3.051381 0.8316018
## upper limit    3.600961 1.5644850
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: MP
##          elevation      slope
## estimate      3.331892 1.1480723
## lower limit    3.079316 0.8316018
## upper limit    3.584467 1.5644850
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845

plot(b3)

#facet count - female
b3.1 <- sma(whole_count_log~tibia_length_log+type, type="elevation", data=eye

```



```

2.f)
summary(b3.1)

## Call: sma(formula = whole_count_log ~ tibia_length_log + type, data = eye2
.f,
##      type = "elevation")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 0.4435 with 1 degrees of freedom
## P-value : 0.50544
## -----
##
## H0 : no difference in elevation.
## Wald statistic: 0.09372 with 1 degrees of freedom
## P-value : 0.7595
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation      slope
## estimate      3.238105 1.2253080
## lower limit    2.936997 0.8928909
## upper limit    3.539213 1.6915754
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##
## Group: MP
##           elevation      slope
## estimate      3.242166 1.2253080
## lower limit    2.952253 0.8928909
## upper limit    3.532079 1.6915754
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798

plot(b3.1)

plot(eye2.f$whole_count_log~eye2.f$tibia_length_log,
      col = eye2.f$type,
      pch = 16)

```

```

#area - male
b4 <- sma(area_log~tibia_length_log+type, type="elevation", data=eye2.m)
summary(b4)

## Call: sma(formula = area_log ~ tibia_length_log + type, data = eye2.m,
##      type = "elevation")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 7.393 with 1 degrees of freedom
## P-value : 0.0065491
## -----
##
## H0 : no difference in elevation.
## Wald statistic: 0.04348 with 1 degrees of freedom
## P-value : 0.83482
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation      slope
## estimate    -0.27058881 1.599145
## lower limit -0.55421065 1.209623
## upper limit  0.01303303 2.041624
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
## Group: MP
##           elevation      slope
## estimate    -0.26789644 1.599145
## lower limit -0.52876075 1.209623
## upper limit -0.00703214 2.041624
##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06

plot(b4)

#area - female
b4.1 <- sma(area_log~tibia_length_log+type, type="elevation", data=eye2.f)
summary(b4.1)

```

```

## Call: sma(formula = area_log ~ tibia_length_log + type, data = eye2.f,
##           type = "elevation")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 0.5582 with 1 degrees of freedom
## P-value : 0.455
## -----
##
## H0 : no difference in elevation.
## Wald statistic: 1.228 with 1 degrees of freedom
## P-value : 0.26785
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation    slope
## estimate    -0.3509322 1.683000
## lower limit -0.6531155 1.330451
## upper limit -0.0487490 2.134163
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736
## P-value : 5.157e-05
##
## Group: MP
##           elevation    slope
## estimate    -0.3651822 1.683000
## lower limit -0.6562231 1.330451
## upper limit -0.0741414 2.134163
##
## H0 : variables uncorrelated.
## R-squared : 0.5928163
## P-value : 0.00078631

plot(b4.1)

#####testing shift along common axis#####

#facet count - male
b5 <- sma(whole_count_log~tibia_length_log+type, type="shift", data=eye2.m)
summary(b5)

```

```

## Call: sma(formula = whole_count_log ~ tibia_length_log + type, data = eye2
.m,
##     type = "shift")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 2.458 with 1 degrees of freedom
## P-value : 0.11696
## -----
##
## H0 : no shift along common axis.
## Wald statistic: 27.44 with 1 degrees of freedom
## P-value : 1.6192e-07
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation      slope
## estimate      3.326171 1.1480723
## lower limit    3.243936 0.8316018
## upper limit    3.945351 1.5644850
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: MP
##           elevation      slope
## estimate      3.331892 1.1480723
## lower limit    2.858373 0.8316018
## upper limit    3.545450 1.5644850
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845

plot(b5)

#facet count - female
b5.1 <- sma(whole_count_log~tibia_length_log+type, type="shift", data=eye2.f)
summary(b5.1)

## Call: sma(formula = whole_count_log ~ tibia_length_log + type, data = eye2
.f,
##     type = "shift")

```

```

##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 0.4435 with 1 degrees of freedom
## P-value : 0.50544
## -----
##
## H0 : no shift along common axis.
## Wald statistic: 27.61 with 1 degrees of freedom
## P-value : 1.482e-07
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation      slope
## estimate      3.238105 1.2253080
## lower limit    2.688987 0.8928909
## upper limit    3.595402 1.6915754
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##
## Group: MP
##           elevation      slope
## estimate      3.242166 1.2253080
## lower limit    2.953909 0.8928909
## upper limit    3.712617 1.6915754
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798

plot(b5.1)

#area - male
b6 <- sma(area_log~tibia_length_log+type, type="shift", data=eye2.m)
summary(b6)

## Call: sma(formula = area_log ~ tibia_length_log + type, data = eye2.m,
##           type = "shift")
##
## Fit using Standardized Major Axis
##

```

```

## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 7.393 with 1 degrees of freedom
## P-value : 0.0065491
## -----
##
## H0 : no shift along common axis.
## Wald statistic: 27.18 with 1 degrees of freedom
## P-value : 1.8548e-07
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation    slope
## estimate   -0.27058881 1.599145
## lower limit -0.08562769 1.209623
## upper limit  0.52735804 2.041624
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
## Group: MP
##           elevation    slope
## estimate   -0.2678964 1.599145
## lower limit -0.8042493 1.209623
## upper limit -0.1262944 2.041624
##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06

plot(b6)

#area - female
b6.1 <- sma(area_log~tibia_length_log+type, type="shift", data=eye2.f)
summary(b6.1)

## Call: sma(formula = area_log ~ tibia_length_log + type, data = eye2.f,
##           type = "shift")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.

```

```

## Likelihood ratio statistic : 0.5582 with 1 degrees of freedom
## P-value : 0.455
## -----
##
## H0 : no shift along common axis.
## Wald statistic: 34.22 with 1 degrees of freedom
## P-value : 4.9142e-09
## -----
##
## Coefficients by group in variable "type"
##
## Group: CP
##           elevation    slope
## estimate    -0.35093225 1.683000
## lower limit -0.86747180 1.330451
## upper limit -0.02529556 2.134163
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736
## P-value : 5.157e-05
##
## Group: MP
##           elevation    slope
## estimate    -0.3651822 1.683000
## lower limit -0.6572024 1.330451
## upper limit  0.1568317 2.134163
##
## H0 : variables uncorrelated.
## R-squared : 0.5928163
## P-value : 0.00078631

```

```
plot(b6.1)
```

```
#####FDR P-VALUE CORRECTION #####
```

```

pvalues2 <- c(0.11696,
              0.0065491,
              0.50544,
              0.455,
              0.67507,
              0.7595,
              0.26785,
              1.6192e-07,
              1.482e-07,
              4.9142e-09)

```

```

adjusted.p2 <- p.adjust(pvalues2, method = "fdr", n = length(pvalues2))
adjusted.p2

```

```

## [1] 2.339200e-01 1.637275e-02 6.318000e-01 6.318000e-01 7.500778e-01
## [6] 7.595000e-01 4.464167e-01 5.397333e-07 5.397333e-07 4.914200e-08

#####3) now including F1 hybrids#####
eye$inter <- interaction(eye$type, eye$sex)

#facet count
c1 <- sma(whole_count_log~tibia_length_log*inter, data=eye, multcomp = T, multcompmethod = "adjust")
multcompmatrix(c1, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               inter_2
## inter_1      CP.female MPxCP.male MP.female CPxMP.male MP.male CP.male
## CP.female      -
## MPxCP.male      -
## MP.female      -
## CPxMP.male      -
## MP.male      -
## CP.male      -
## CPxMP.female
## MPxCP.female
##               inter_2
## inter_1      CPxMP.female MPxCP.female
## CP.female
## MPxCP.male
## MP.female
## CPxMP.male
## MP.male
## CP.male
## CPxMP.female -
## MPxCP.female -

summary(c1)

## Call: sma(formula = whole_count_log ~ tibia_length_log * inter, data = eye
,
##      multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 7.464 with 7 degrees of freedom
## P-value : 0.3822
## -----
##

```



```

## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in slope :
##      inter_1      inter_2      Pval      TestStat
## 1      CP.female CPxMP.female 1.00000 0.0552550141
## 2      CP.female      MP.female 1.00000 0.4434903863
## 3      CP.female MPxCP.female 0.94757 2.7064376571
## 4      CP.female      CP.male 0.97990 2.2896319310
## 5      CP.female CPxMP.male 1.00000 0.0006193157
## 6      CP.female      MP.male 1.00000 0.0027806498
## 7      CP.female MPxCP.male 0.99960 1.3585339811
## 8 CPxMP.female      MP.female 1.00000 0.6865808043
## 9 CPxMP.female MPxCP.female 0.91895 2.9509245933
## 10 CPxMP.female      CP.male 0.96172 2.5542830939
## 11 CPxMP.female CPxMP.male 1.00000 0.0665304562
## 12 CPxMP.female      MP.male 1.00000 0.0893557790
## 13 CPxMP.female MPxCP.male 0.99812 1.6362292468
## 14      MP.female MPxCP.female 0.99997 1.0257654093
## 15      MP.female      CP.male 1.00000 0.8073642614
## 16      MP.female CPxMP.male 1.00000 0.4144393854
## 17      MP.female      MP.male 1.00000 0.4528463996
## 18      MP.female MPxCP.male 1.00000 0.2917443153
## 19 MPxCP.female      CP.male 1.00000 0.0072575172
## 20 MPxCP.female CPxMP.male 0.95418 2.6390570492
## 21 MPxCP.female      MP.male 0.92155 2.9310307543
## 22 MPxCP.female MPxCP.male 1.00000 0.1987820145
## 23      CP.male CPxMP.male 0.98295 2.2302342983
## 24      CP.male      MP.male 0.96927 2.4576282480
## 25      CP.male MPxCP.male 1.00000 0.1232918867
## 26 CPxMP.male      MP.male 1.00000 0.0006597410
## 27 CPxMP.male MPxCP.male 0.99971 1.3122193965
## 28      MP.male MPxCP.male 0.99932 1.4444363378
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.female
##      elevation      slope
## estimate      3.142194 1.3540100
## lower limit  2.688987 0.8763565
## upper limit  3.595402 2.0920060
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##
## Group: CPxMP.female
##      elevation      slope
## estimate      3.042379 1.4666891
## lower limit  2.440755 0.8623369

```

```

## upper limit  3.644004 2.4945900
##
## H0 : variables uncorrelated.
## R-squared : 0.1375647
## P-value : 0.17351
##
## Group: MP.female
##           elevation      slope
## estimate      3.333263 1.0952430
## lower limit    2.953909 0.6804684
## upper limit    3.712617 1.7628405
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798
##
## Group: MPxCP.female
##           elevation      slope
## estimate      3.577457 0.7615128
## lower limit    3.263320 0.4413273
## upper limit    3.891595 1.3139949
##
## H0 : variables uncorrelated.
## R-squared : 0.08552689
## P-value : 0.29017
##
## Group: CP.male
##           elevation      slope
## estimate      3.594643 0.7873019
## lower limit    3.243936 0.4465518
## upper limit    3.945351 1.3880679
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: CPxMP.male
##           elevation      slope
## estimate      3.160335 1.3437926
## lower limit    2.708533 0.8652938
## upper limit    3.612138 2.0868965
##
## H0 : variables uncorrelated.
## R-squared : 0.4245295
## P-value : 0.0084988
##
## Group: MP.male
##           elevation      slope
## estimate      3.201911 1.3342359
## lower limit    2.858373 0.9302675

```

```

## upper limit  3.545450 1.9136273
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845
##
## Group: MPxCP.male
##           elevation      slope
## estimate    3.481762 0.9030686
## lower limit  3.102523 0.5239273
## upper limit  3.861001 1.5565765
##
## H0 : variables uncorrelated.
## R-squared : 0.08947516
## P-value : 0.27879

plot(c1)

eye.m <- eye[which(eye$sex=="male"),]

c1m <- sma(whole_count_log~tibia_length_log*inter, data=eye.m, multcomp = T,
multcompmethod = "adjust")
multcompmatrix(c1m, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           inter_2
## inter_1    CP.male CPxMP.male MPxCP.male MP.male
##   CP.male      -
##   CPxMP.male      -
##   MPxCP.male      -
##   MP.male      -

summary(c1m)

## Call: sma(formula = whole_count_log ~ tibia_length_log * inter, data = eye
.m,
##   multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 3.77 with 3 degrees of freedom
## P-value : 0.28744
## -----
##
## Results of multiple comparisons among groups.

```

```

##
## Test for pair-wise difference in slope :
##      inter_1    inter_2    Pval    TestStat
## 1    CP.male CPxMP.male 0.58208 2.230234298
## 2    CP.male  MP.male  0.52587 2.457628248
## 3    CP.male MPxCP.male 0.99957 0.123291887
## 4 CPxMP.male  MP.male  1.00000 0.000659741
## 5 CPxMP.male MPxCP.male 0.82484 1.312219397
## 6    MP.male MPxCP.male 0.79064 1.444436338
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##           elevation      slope
## estimate      3.594643 0.7873019
## lower limit    3.243936 0.4465518
## upper limit    3.945351 1.3880679
##
## H0 : variables uncorrelated.
## R-squared : 0.00413478
## P-value : 0.8199
##
## Group: CPxMP.male
##           elevation      slope
## estimate      3.160335 1.3437926
## lower limit    2.708533 0.8652938
## upper limit    3.612138 2.0868965
##
## H0 : variables uncorrelated.
## R-squared : 0.4245295
## P-value : 0.0084988
##
## Group: MP.male
##           elevation      slope
## estimate      3.201911 1.3342359
## lower limit    2.858373 0.9302675
## upper limit    3.545450 1.9136273
##
## H0 : variables uncorrelated.
## R-squared : 0.5165108
## P-value : 0.00077845
##
## Group: MPxCP.male
##           elevation      slope
## estimate      3.481762 0.9030686
## lower limit    3.102523 0.5239273
## upper limit    3.861001 1.5565765
##
## H0 : variables uncorrelated.

```

```

## R-squared : 0.08947516
## P-value : 0.27879

plot(c1m)

eye.f <- eye[which(eye$sex=="female"),]

c1f <- sma(whole_count_log~tibia_length_log*inter, data=eye.f, multcomp = T,
multcompmethod = "adjust")
multcompmatrix(c1f, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               inter_2
## inter_1      CP.female MPxCP.female CPxMP.female MP.female
## CP.female      -
## MPxCP.female      -
## CPxMP.female      -
## MP.female      -

summary(c1f)

## Call: sma(formula = whole_count_log ~ tibia_length_log * inter, data = eye
.f,
##      multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 3.692 with 3 degrees of freedom
## P-value : 0.29675
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in slope :
##      inter_1      inter_2      Pval      TestStat
## 1      CP.female CPxMP.female 0.99996 0.05525501
## 2      CP.female      MP.female 0.98537 0.44349039
## 3      CP.female MPxCP.female 0.46836 2.70643766
## 4 CPxMP.female      MP.female 0.95666 0.68658080
## 5 CPxMP.female MPxCP.female 0.41633 2.95092459
## 6      MP.female MPxCP.female 0.89316 1.02576541
##
## -----
## Coefficients by group in variable "inter"
##

```

```

## Group: CP.female
##           elevation      slope
## estimate      3.142194 1.3540100
## lower limit    2.688987 0.8763565
## upper limit    3.595402 2.0920060
##
## H0 : variables uncorrelated.
## R-squared : 0.2303859
## P-value : 0.037543
##
## Group: CPxMP.female
##           elevation      slope
## estimate      3.042379 1.4666891
## lower limit    2.440755 0.8623369
## upper limit    3.644004 2.4945900
##
## H0 : variables uncorrelated.
## R-squared : 0.1375647
## P-value : 0.17351
##
## Group: MP.female
##           elevation      slope
## estimate      3.333263 1.0952430
## lower limit    2.953909 0.6804684
## upper limit    3.712617 1.7628405
##
## H0 : variables uncorrelated.
## R-squared : 0.3199202
## P-value : 0.02798
##
## Group: MPxCP.female
##           elevation      slope
## estimate      3.577457 0.7615128
## lower limit    3.263320 0.4413273
## upper limit    3.891595 1.3139949
##
## H0 : variables uncorrelated.
## R-squared : 0.08552689
## P-value : 0.29017

plot(c1f)

#####**plot facet count major axis regressions: C, M, F1**#####
dev.off()

## null device
##           1

```

```

par(mgp=c(3.5,0.6,0))
par(mar = c(5, 6, 2, 1.25))
par(lwd=1)

eye$inter<- factor(eye$inter, c("CP.male", "CP.female",
                                "CPxMP.male", "CPxMP.female",
                                "MPxCP.male", "MPxCP.female",
                                "MP.male", "MP.female"))

#males
plot(c1, col=c("#1F78B4", "#1F78B4", "#FF7F00", "#FF7F00", "grey40", "grey40",
               "#FDBF6F", "#FDBF6F"),
     #pch=c(17,2, 15,0, 18,5, 19,1),
     pch="",
     lwd=3,
     cex=0.9,
     lty=c(1,0,1,0,1,0,1,0),
     xlab = "",
     ylab = "log10 (facet count)",
     cex.axis = 1.65, cex.lab = 1.7, font.lab = 2, las=1,
     xlim=c(0.635,0.80), ylim=c(4.03,4.25),
     family = "Times New Roman", tck=-0.015)

legend("topleft", c("C", "CxM", "MxC", "M"),
      col = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"), lty= 1, lwd=3,
      bty="n")

plot(c1m, col=c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"),
     #pch=c(17,15,18,19),
     pch=19,
     lwd=3,
     cex=0.25,
     lty=c(1,1,1,1),
     xlim=c(0.635,0.80), ylim=c(4.03,4.25),
     las=1, cex.axis = 1.3,
     xlab = "",
     ylab = "",
     tck=-0.015,
     family = "Times New Roman")

title(xlab = "log10 (hind tibia length)", line = 2, cex.lab = 1.2, font.lab=2, family = "Times New Roman")

title(ylab = "log10 (facet count)", line = 3, cex.lab = 1.2, font.lab=2, family = "Times New Roman")

title("males", adj = 0.5, line = 0.6, family = "Times New Roman", cex.main=1.35)

legend("topleft", c("C", "CxM", "MxC", "M"),
      col = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"), lty= 1, lwd=3,

```

```

    bty="n")

#females
plot(c1, col=c("#1F78B4", "#1F78B4", "#FF7F00", "#FF7F00", "grey40", "grey40",
"#FDBF6F", "#FDBF6F"),
    #pch=c(17,2, 15,0, 18,5, 19,1),
    pch="",
    lwd=3,
    cex=0.9,
    lty=c(0,1,0,1,0,1,0,1),
    xlab = "",
    ylab = "log10 (facet count)",
    cex.axis = 1.65, cex.lab = 1.7, font.lab = 2, las=1,
    xlim=c(0.635,0.80), ylim=c(4.03,4.25),
    family = "Times New Roman", tck=-0.015)

legend("topleft", c("C", "CxM", "MxC", "M"),
    col = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"), lty= 1, lwd=3,
    bty="n")

plot(c1f, col=c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"),
    #pch=c(17,15,18,19),
    pch=19,
    lwd=3,
    cex=0.25,
    lty=c(1,1,1,1),
    xlim=c(0.635,0.80), ylim=c(4.03,4.25),
    las=1, cex.axis = 1.3,
    xlab = "",
    ylab = "",
    tck=-0.015,
    family = "Times New Roman")

title(xlab = "log10 (hind tibia length)", line = 2, cex.lab = 1.2, font.lab=2, f
amily = "Times New Roman")

title(ylab = "log10 (facet count)", line = 3, cex.lab = 1.2, font.lab=2, family
= "Times New Roman")

title("females", adj = 0.5, line = 0.6, family = "Times New Roman", cex.main=
1.35)

legend("topleft", c("C", "CxM", "MxC", "M"),
    col = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"), lty= 1, lwd=3,
    bty="n")

#area
d1 <- sma(area_log~tibia_length_log*inter, data=eye, multcomp = T, multcompme

```



```

thod = "adjust")
multcompmatrix(d1, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               inter_2
## inter_1      CP.male CPxMP.male MP.female MP.male CP.female CPxMP.female
##   CP.male      -
##   CPxMP.male      -
##   MP.female      -
##   MP.male      -
##   CP.female      -
##   CPxMP.female      -
##   MPxCP.female
##   MPxCP.male
##               inter_2
## inter_1      MPxCP.female MPxCP.male
##   CP.male
##   CPxMP.male
##   MP.female
##   MP.male
##   CP.female
##   CPxMP.female
##   MPxCP.female -
##   MPxCP.male      -

summary(d1)

## Call: sma(formula = area_log ~ tibia_length_log * inter, data = eye,
##   multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 11.56 with 7 degrees of freedom
## P-value : 0.11589
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in slope :
##               inter_1      inter_2      Pval      TestStat
## 1      CP.male      CP.female 0.32038 6.076552e+00
## 2      CP.male      CPxMP.male 0.11628 8.108833e+00
## 3      CP.male      CPxMP.female 0.56824 4.735240e+00
## 4      CP.male      MPxCP.male 0.99969 1.319598e+00

```

```

## 5      CP.male MPxCP.female 0.94996 2.682725e+00
## 6      CP.male      MP.male 0.16804 7.392662e+00
## 7      CP.male      MP.female 0.91925 2.948582e+00
## 8      CP.female  CPxMP.male 1.00000 5.589360e-01
## 9      CP.female  CPxMP.female 1.00000 1.755995e-02
## 10     CP.female  MPxCP.male 0.99908 1.498170e+00
## 11     CP.female  MPxCP.female 1.00000 3.952514e-01
## 12     CP.female      MP.male 1.00000 3.885275e-02
## 13     CP.female      MP.female 1.00000 5.581682e-01
## 14     CPxMP.male  CPxMP.female 1.00000 2.315879e-01
## 15     CPxMP.male  MPxCP.male 0.90691 3.038767e+00
## 16     CPxMP.male  MPxCP.female 0.99930 1.451255e+00
## 17     CPxMP.male      MP.male 1.00000 3.961990e-01
## 18     CPxMP.male      MP.female 0.99557 1.831142e+00
## 19     CPxMP.female  MPxCP.male 0.99977 1.276150e+00
## 20     CPxMP.female  MPxCP.female 1.00000 4.061232e-01
## 21     CPxMP.female      MP.male 1.00000 5.021207e-05
## 22     CPxMP.female      MP.female 1.00000 5.225954e-01
## 23     MPxCP.male  MPxCP.female 1.00000 2.662393e-01
## 24     MPxCP.male      MP.male 0.98889 2.088041e+00
## 25     MPxCP.male      MP.female 1.00000 2.617518e-01
## 26     MPxCP.female      MP.male 1.00000 6.708751e-01
## 27     MPxCP.female      MP.female 1.00000 1.963364e-03
## 28      MP.male      MP.female 0.99999 9.455849e-01
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##          elevation      slope
## estimate    0.22086518 0.9387334
## lower limit -0.08562769 0.6134198
## upper limit  0.52735804 1.4365697
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
## Group: CP.female
##          elevation      slope
## estimate    -0.44638368 1.811085
## lower limit -0.86747180 1.332310
## upper limit -0.02529556 2.461912
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736
## P-value : 5.157e-05
##
## Group: CPxMP.male
##          elevation      slope

```

```

## estimate      -0.71608306 2.166657
## lower limit   -1.34718526 1.475379
## upper limit   -0.08498085 3.181827
##
## H0 : variables uncorrelated.
## R-squared : 0.5680501
## P-value : 0.0011742
##
## Group: CPxMP.female
##           elevation      slope
## estimate      -0.5253307 1.878349
## lower limit    -1.2020595 1.172645
## upper limit     0.1513980 3.008748
##
## H0 : variables uncorrelated.
## R-squared : 0.3346212
## P-value : 0.023879
##
## Group: MPxCP.male
##           elevation      slope
## estimate      -0.08807595 1.318846
## lower limit    -0.51496747 0.859966
## upper limit     0.33881557 2.022585
##
## H0 : variables uncorrelated.
## R-squared : 0.4588542
## P-value : 0.0055297
##
## Group: MPxCP.female
##           elevation      slope
## estimate      -0.2677770 1.5376629
## lower limit    -0.7647541 0.9951569
## upper limit     0.2292001 2.3759139
##
## H0 : variables uncorrelated.
## R-squared : 0.4385127
## P-value : 0.0071532
##
## Group: MP.male
##           elevation      slope
## estimate      -0.4652718 1.881835
## lower limit    -0.8042493 1.458194
## upper limit    -0.1262944 2.428555
##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06
##
## Group: MP.female
##           elevation      slope

```

```

## estimate      -0.2501853 1.518812
## lower limit   -0.6572024 1.045336
## upper limit    0.1568317 2.206746
##
## H0 : variables uncorrelated.
## R-squared : 0.5928163
## P-value : 0.00078631

plot(d1)

eye.m <- eye[which(eye$sex=="male"),]

d1m <- sma(area_log~tibia_length_log*inter, data=eye.m, multcomp = T, multcom
pmethod = "adjust")
multcompmatrix(d1m, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               inter_2
## inter_1      CP.male CPxMP.male MP.male MPxCP.male
## CP.male      -        *          *
## CPxMP.male *          -
## MP.male      *          -
## MPxCP.male          -

summary(d1m)

## Call: sma(formula = area_log ~ tibia_length_log * inter, data = eye.m,
##      multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 10.58 with 3 degrees of freedom
## P-value : 0.0142
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in slope :
##      inter_1  inter_2    Pval TestStat
## 1  CP.male CPxMP.male 0.026141 8.108833
## 2  CP.male MPxCP.male 0.822965 1.319598
## 3  CP.male  MP.male 0.038657 7.392662
## 4 CPxMP.male MPxCP.male 0.398758 3.038767
## 5 CPxMP.male  MP.male 0.989091 0.396199
## 6 MPxCP.male  MP.male 0.618722 2.088041

```

```

##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.male
##           elevation      slope
## estimate      0.22086518 0.9387334
## lower limit -0.08562769 0.6134198
## upper limit  0.52735804 1.4365697
##
## H0 : variables uncorrelated.
## R-squared : 0.4645721
## P-value : 0.0051359
##
## Group: CPxMP.male
##           elevation      slope
## estimate      -0.71608306 2.166657
## lower limit -1.34718526 1.475379
## upper limit -0.08498085 3.181827
##
## H0 : variables uncorrelated.
## R-squared : 0.5680501
## P-value : 0.0011742
##
## Group: MPxCP.male
##           elevation      slope
## estimate      -0.08807595 1.318846
## lower limit -0.51496747 0.859966
## upper limit  0.33881557 2.022585
##
## H0 : variables uncorrelated.
## R-squared : 0.4588542
## P-value : 0.0055297
##
## Group: MP.male
##           elevation      slope
## estimate      -0.4652718 1.881835
## lower limit -0.8042493 1.458194
## upper limit -0.1262944 2.428555
##
## H0 : variables uncorrelated.
## R-squared : 0.7633368
## P-value : 2.1756e-06

plot(d1m)

eye.f <- eye[which(eye$sex=="female"),]

d1f <- sma(area_log~tibia_length_log*inter, data=eye.f, multcomp = T, multcom

```

```

pmethod = "adjust")
multcompmatrix(d1f, sort=T)

## Multiple comparison matrix.
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               inter_2
## inter_1      CP.female MP.female CPxMP.female MPxCP.female
## CP.female    -
## MP.female          -
## CPxMP.female                -
## MPxCP.female                        -

summary(d1f)

## Call: sma(formula = area_log ~ tibia_length_log * inter, data = eye.f,
##           multcomp = T, multcompmethod = "adjust")
##
## Fit using Standardized Major Axis
##
## -----
## Results of comparing lines among groups.
##
## H0 : slopes are equal.
## Likelihood ratio statistic : 0.9642 with 3 degrees of freedom
## P-value : 0.80991
## -----
##
## Results of multiple comparisons among groups.
##
## Test for pair-wise difference in slope :
##      inter_1      inter_2      Pval      TestStat
## 1  CP.female CPxMP.female 1.00000 0.017559948
## 2  CP.female MPxCP.female 0.98916 0.395251399
## 3  CP.female  MP.female 0.97380 0.558168187
## 4 CPxMP.female MPxCP.female 0.98836 0.406123187
## 5 CPxMP.female  MP.female 0.97777 0.522595433
## 6 MPxCP.female  MP.female 1.00000 0.001963364
##
## -----
## Coefficients by group in variable "inter"
##
## Group: CP.female
##      elevation      slope
## estimate    -0.44638368 1.811085
## lower limit -0.86747180 1.332310
## upper limit -0.02529556 2.461912
##
## H0 : variables uncorrelated.
## R-squared : 0.6285736

```



*#males*

```
plot(d1m, col=c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"),
     #pch=c(17,15,18,19),
     pch=19,
     lwd=3,
     cex=0.25,
     lty=c(1,1,1,1),
     xlim=c(0.635,0.80), ylim=c(0.73,1.00),
     las=1,cex.axis = 1.3,
     xlab = "",
     ylab = "",
     tck=-0.015,
     family = "Times New Roman")

title(xlab = "log10 (hind tibia length)", line = 2,cex.lab = 1.2,font.lab=2,family = "Times New Roman")

title(ylab = "log10 (corneal area)", line = 3,cex.lab = 1.2,font.lab=2,family = "Times New Roman")

title("males", adj = 0.5, line = 0.6, family = "Times New Roman", cex.main=1.35)

legend("topleft", c("C","CxM", "MxC", "M"),
      col = c("#1F78B4","#FF7F00", "grey40", "#FDBF6F"), lty= 1, lwd=3,
      bty="n")
```

*#females*

```
plot(d1f, col=c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"),
     #pch=c(17,15,18,19),
     pch=19,
     lwd=3,
     cex=0.25,
     lty=c(1,1,1,1),
     xlim=c(0.635,0.80), ylim=c(0.73,1.00),
     las=1,cex.axis = 1.3,
     xlab = "",
     ylab = "",
     tck=-0.015,
     family = "Times New Roman")

title(xlab = "log10 (hind tibia length)", line = 2,cex.lab = 1.2,font.lab=2,family = "Times New Roman")

title(ylab = "log10 (corneal area)", line = 3,cex.lab = 1.2,font.lab=2,family = "Times New Roman")

title("females", adj = 0.5, line = 0.6, family = "Times New Roman", cex.main=1.35)
```



```
legend("topleft", c("C", "CxM", "MxC", "M"),  
      col = c("#1F78B4", "#FF7F00", "grey40", "#FDBF6F"), lty= 1, lwd=3,  
      bty="n")
```