

# CviMVP\_Nursery\_Hatchery\_Analysis

2024-02-22

## R Markdown

```
#load necessary packages  
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v dplyr      1.1.4      v readr      2.1.5  
## v forcats    1.0.0      v stringr   1.5.1  
## v ggplot2    3.5.1      v tibble    3.2.1  
## v lubridate  1.9.3      v tidyr     1.3.1  
## v purrr      1.0.2  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()     masks stats::lag()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)  
library(gridExtra)
```

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

```
library(multcompView)  
library(car) #upload package with Anova function
```

```
## Loading required package: carData  
##  
## Attaching package: 'car'  
##  
## The following object is masked from 'package:dplyr':  
##  
##      recode  
##  
## The following object is masked from 'package:purrr':  
##  
##      some
```

```
library(cowplot)
```

```
##  
## Attaching package: 'cowplot'  
##  
## The following object is masked from 'package:lubridate':  
##  
##     stamp
```

```
#Set working directory
```

```
setwd(dirname(rstudioapi::getActiveDocumentContext())$path))
```

```
#download data in csv format
```

```
#download spawn treatment data
```

```
spawn <- read.csv("../data/performance_H2F/CviMVP_Larvae_Growth-spawn_trt.csv")
```

```
#download group data
```

```
group <- read.csv("../data/performance_H2F/CviMVP_Larvae_Growth-group.csv")
```

```
#download larvae image length data
```

```
larvae_img <- read.csv("../data/performance_H2F/CviMVP_Larvae_Growth-larvae_img.csv")
```

```
#make mega sheet with all length data
```

```
#join image data to group data in new dataframe
```

```
len_df <- group %>%  
  left_join(larvae_img, by = "Group_Day_Key")
```

```
#remove some columns
```

```
len_analysis_df <- subset(len_df, select = -c(Ind_ID, Group_Day_Label.y, Image_Key, Img_Notes, SpawnTrt_L))
```

```
#subset data for oysters aged 15-78 days (later stage offspring)
```

```
latestage_df <- subset(len_analysis_df, Day_analysis == "78" | Day_analysis == "15-21")
```

```
#a sample marked WV is erroneously in this data set- remove
```

```
latestage_df <- subset(latestage_df, Image_Label != "WV_06_200L_6")
```

```
#rename cols
```

```
colnames(latestage_df) <- c("group", "group_day_key", "group_day_label", "day_trt", "day_an", "day", "date")
```

```
#make column with measurements in millimeters instead of microns
```

```
latestage_df$shell_length_mm <- (latestage_df$shell_length_um)/1000  
head(latestage_df)
```

```
##      group group_day_key group_day_label day_trt day_an day      date  
## 378 MVP-LA      7E8E995C      LA_19_Eyed 19_Eyed 15-21 19 2023-05-28  
## 379 MVP-LA      7E8E995C      LA_19_Eyed 19_Eyed 15-21 19 2023-05-28  
## 380 MVP-LA      7E8E995C      LA_19_Eyed 19_Eyed 15-21 19 2023-05-28  
## 381 MVP-LA      7E8E995C      LA_19_Eyed 19_Eyed 15-21 19 2023-05-28  
## 382 MVP-LA      7E8E995C      LA_19_Eyed 19_Eyed 15-21 19 2023-05-28
```

```
## 383 MVP-LA      7E8E995C      LA_19_Eyed 19_Eyed 15-21 19 2023-05-28
##      image_label shell_length_um shell_length_mm
## 378 TX_21_small_21      167.430      0.167430
## 379 LA_19_eyed_1      316.357      0.316357
## 380 LA_19_eyed_2      319.598      0.319598
## 381 LA_19_eyed_3      332.948      0.332948
## 382 LA_19_eyed_4      324.306      0.324306
## 383 LA_19_eyed_5      332.164      0.332164
```

```
#prep data for ANOVA
```

```
#use grep function to make a vector of all indices in latestage_len with the phrase "small" in them, ignore case
small <- grep("small", latestage_df$group_day_label, ignore.case = TRUE)
```

```
small_img <- grep("small", latestage_df$image_label, ignore.case = TRUE)
```

```
#make new dataframe WITHOUT any indices saved in small, mean remaining data are everything except small
no_small_df <- latestage_df[-small, ]
no_small_df <- latestage_df[-small_img, ]
```

```
#rename rows
```

```
colnames(no_small_df) <- c("group", "group_day_key", "group_day_label", "day_trt", "day_an", "day", "day_label")
```

```
#exclude "MVP" from group names and include wild or selection line label
```

```
no_small_df["group"][no_small_df["group"] == "MVP-TX"] <- "W1-TX"
no_small_df["group"][no_small_df["group"] == "MVP-LA"] <- "W2-LA"
no_small_df["group"][no_small_df["group"] == "MVP-FL"] <- "W3-FL"
no_small_df["group"][no_small_df["group"] == "MVP-LOLA"] <- "S1-LOLA"
no_small_df["group"][no_small_df["group"] == "MVP-DEBY"] <- "S2-DEBY"
no_small_df["group"][no_small_df["group"] == "MVP-JR"] <- "W4-VA"
no_small_df["group"][no_small_df["group"] == "MVP-NH"] <- "W5-NH"
no_small_df["group"][no_small_df["group"] == "MVP-ME"] <- "W6-ME"
no_small_df["group"][no_small_df["group"] == "MVP-LARMIX"] <- "H1-LARMIX"
```

```
no_small_df <- no_small_df %>%
```

```
  mutate(group = group %>%
```

```
    fct_relevel("W1-TX", "W2-LA", "W3-FL", "S1-LOLA", "S2-DEBY", "W4-VA", "W5-NH", "W6-ME", "H1-LARMIX"))
```

```
#save latestage_df as a csv file for future analyses
```

```
write.csv(no_small_df, "../data/performance_H2F/latestage_length.csv")
```

```
#calculate mean oyster length per group per day using aggregate
```

```
#make standard error equation
```

```
se <- function(x){
  sd(x)/sqrt(length(x))
}
```

```
#find mean lengths of each group on each day
```

```
means_mm <- aggregate(shell_length_mm ~ group + day_an, FUN = mean, data = no_small_df)
colnames(means_mm) <- c("group", "days", "mean_length_mm")
```

```
#find standard error of length of each group on each day
```

```
se_mm <- aggregate(shell_length_mm ~ group + day_an, FUN = se, data = no_small_df)
colnames(se_mm) <- c("group", "days", "se_length_mm")
```

```
#Check if late stage oyster data (day 19 onward, small oysters excluded) match assumptions of ANOVA
```

```
len.lm <- lm(shell_length_mm ~ day_an*group, data = no_small_df) #fit to lm
len.lm.resid <- len.lm$residuals #save residuals
shapiro.test(len.lm.resid) #use Shapiro Wilk test to assess normality. p < 2.2e-16, data is not normal
```

```
##
## Shapiro-Wilk normality test
##
## data: len.lm.resid
## W = 0.9128, p-value < 2.2e-16
```

```
#plot(len.lm)
```

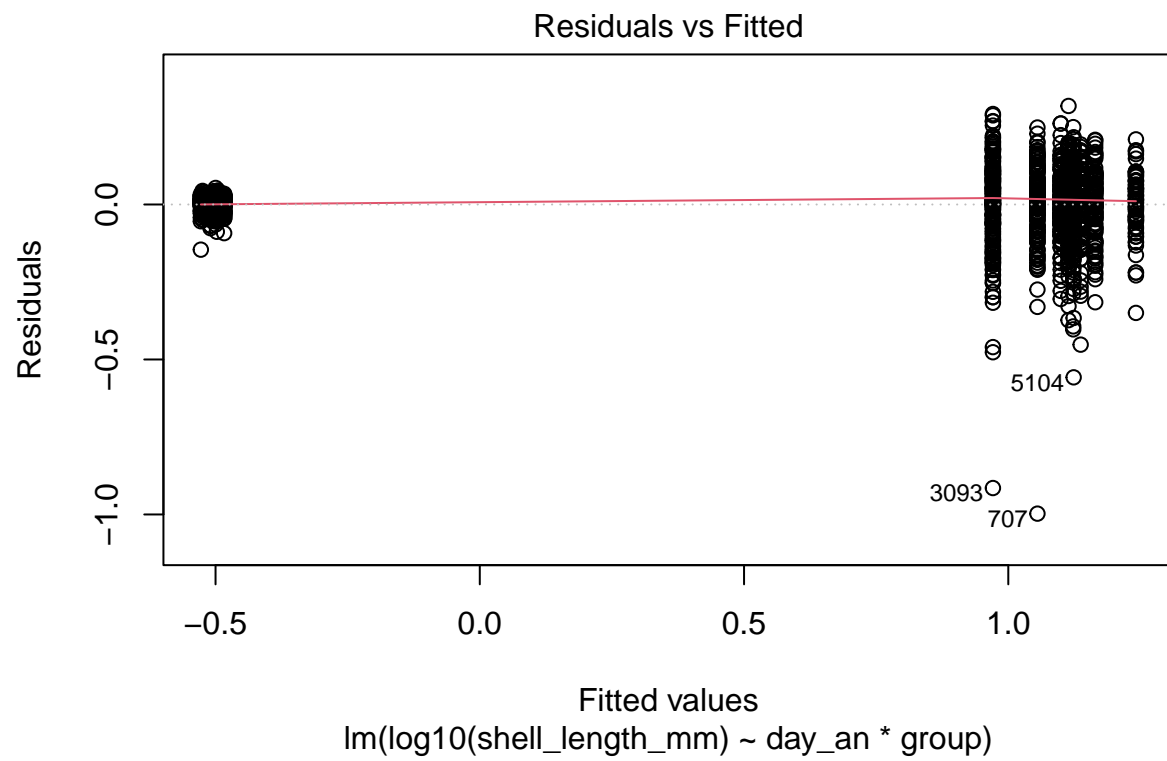
```
#normality is off, definite funnel in homoscedasticity plots, but KEL said to expect that what with the
```

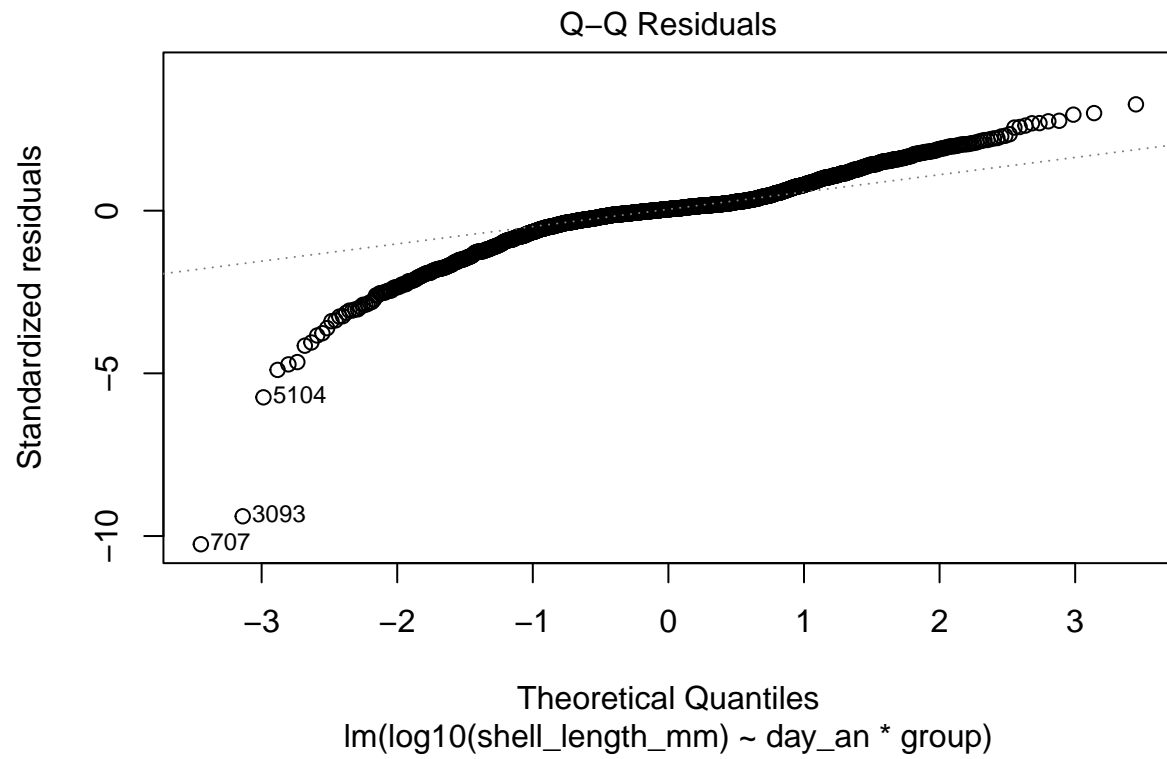
```
#log10 transformation lm
```

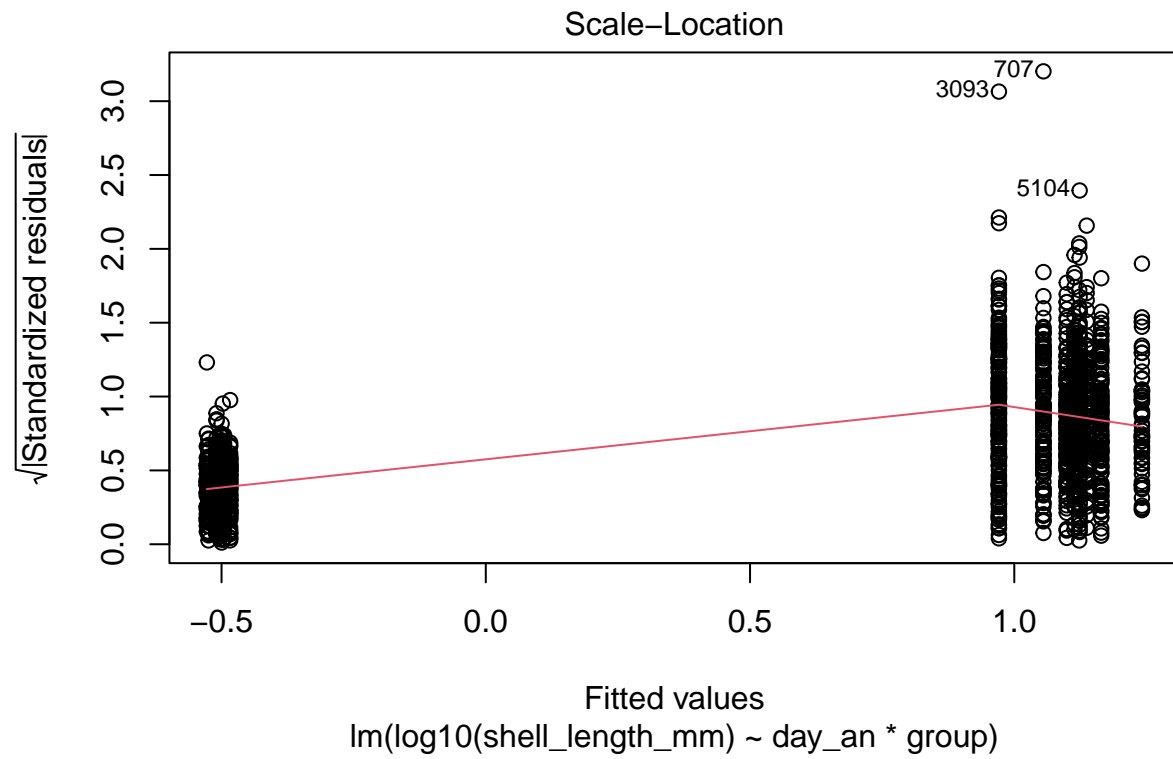
```
loglen.lm <- lm(log10(shell_length_mm) ~ day_an*group, data = no_small_df)
loglen.lm.resid <- loglen.lm$residuals
shapiro.test(loglen.lm.resid)#p < 2.2e-16, no change in normality
```

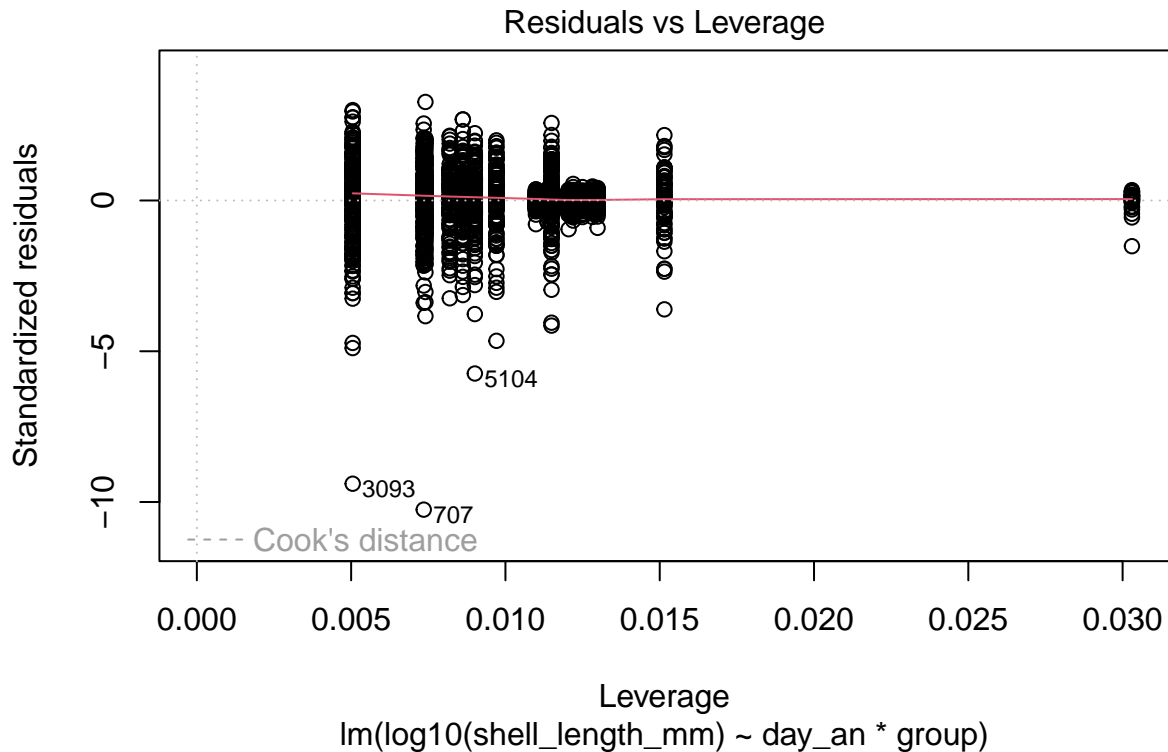
```
##
## Shapiro-Wilk normality test
##
## data: loglen.lm.resid
## W = 0.88572, p-value < 2.2e-16
```

```
plot(loglen.lm)
```









*# I'm planning to use the log10 transformed data for the analysis, transformations helped get rid of so*

*#run ANOVA on log transformed length data*

```
no_small_df$loglen <- log10(no_small_df$shell_length_mm)
```

```
len.anova3 <- Anova(lm(loglen ~ day_an*group, data = no_small_df), type = 3)
```

```
## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted
## Warning in printHypothesis(L, rhs, names(b)): one or more coefficients in the hypothesis include
## arithmetic operators in their names;
## the printed representation of the hypothesis will be omitted
```

```
len.anova3
```

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

```
##
```

```
## Response: loglen
```

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	9.194	1	964.4116	<2e-16 ***
day_an	68.017	1	7134.8554	<2e-16 ***
group	0.106	8	1.3858	0.1978
day_an:group	1.814	8	23.7832	<2e-16 ***
Residuals	16.731	1755		



```
## ---
```

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

*#significant interactive effect of group and age ( $p < 2e-16$ ), but no significant effect of group on its*

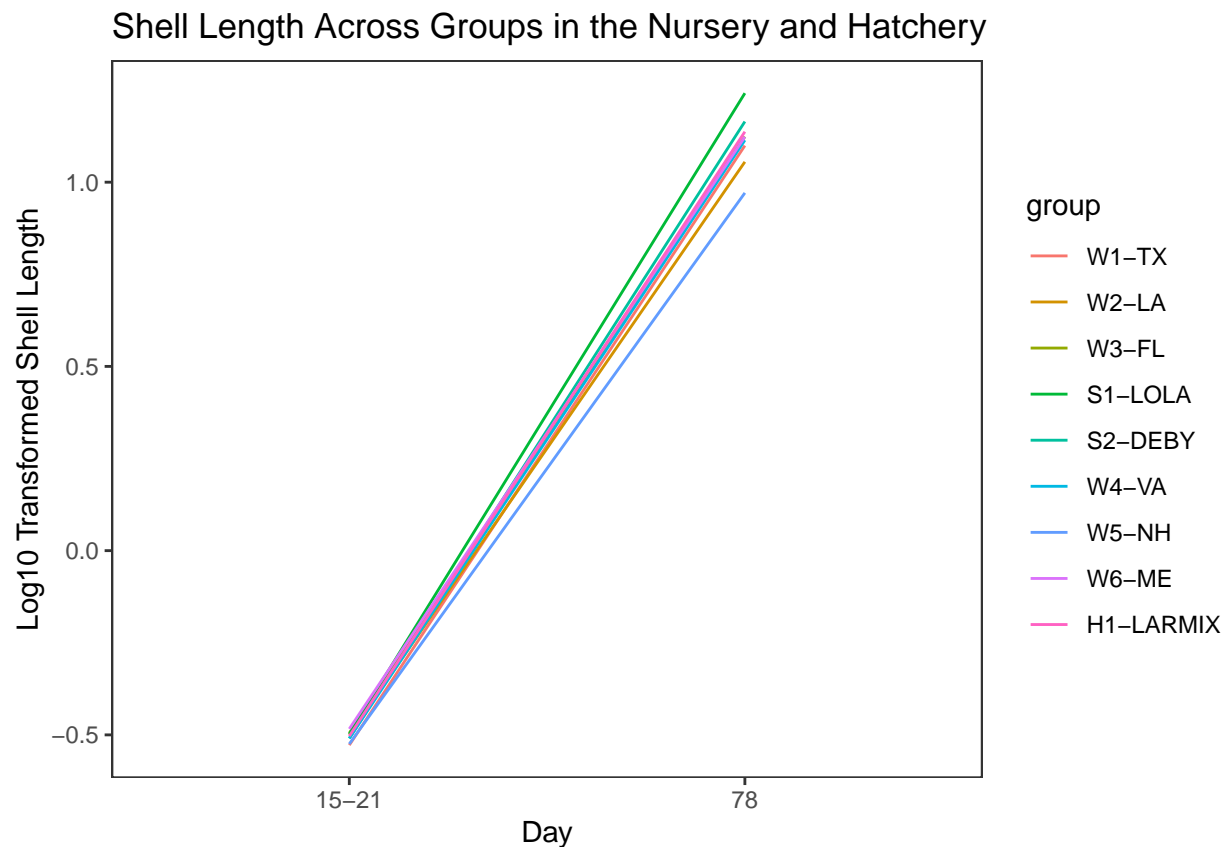
```
all_pval <- len.anova3$`Pr(>F)`
```

```
all_pval <- na.omit(all_pval)
```

```
#Interaction plot for length as a function of day*
```

```
ggplot(no_small_df, aes(day_an, loglen, group = group))+  
  #geom_point(data = means_mm, aes(days, log10(mean_length_mm), color = group))+  
  geom_smooth(method = "lm", se = F, linewidth = .5, aes(color = group))+  
  theme_bw()+  
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+  
  xlab("Day")+  
  ylab("Log10 Transformed Shell Length")+  
  ggtitle("Shell Length Across Groups in the Nursery and Hatchery")
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
#convert group and day from chr to factor  
no_small_df$group <- as.factor(no_small_df$group)  
no_small_df$day_an <- as.factor(no_small_df$day_an)  
str(no_small_df)
```

```
## 'data.frame':    1773 obs. of  11 variables:
## $ group          : Factor w/ 9 levels "W1-TX","W2-LA",...: 2 2 2 2 2 2 2 2 2 ...
## $ group_day_key  : chr  "7E8E995C" "7E8E995C" "7E8E995C" "7E8E995C" ...
## $ group_day_label: chr  "LA_19_Eyed" "LA_19_Eyed" "LA_19_Eyed" "LA_19_Eyed" ...
## $ day_trt        : chr  "19_Eyed" "19_Eyed" "19_Eyed" "19_Eyed" ...
## $ day_an         : Factor w/ 2 levels "15-21","78": 1 1 1 1 1 1 1 1 1 ...
## $ day            : int   19 19 19 19 19 19 19 19 19 ...
## $ date           : chr   "2023-05-28" "2023-05-28" "2023-05-28" "2023-05-28" ...
## $ image_label    : chr   "LA_19_eyed_1" "LA_19_eyed_2" "LA_19_eyed_3" "LA_19_eyed_4" ...
## $ shell_length_um: num   316 320 333 324 332 ...
## $ shell_length_mm: num    0.316 0.32 0.333 0.324 0.332 ...
## $ loglen         : num   -0.5 -0.495 -0.478 -0.489 -0.479 ...
```

```
#subsetting based on oyster age (days) for oysters on day 78 (end of hatchery stage, pre field deployment)
predep <- subset(no_small_df, day_an == "78")
```

```
##ANOVA for end of hatchery (day 78) shell lengths only
```

```
#Linear model and ANOVA for predep shell lengths
predep$group <- as.factor(predep$group)

predep.lm <- lm(shell_length_mm ~ group, data = predep) #fit to lm

predep.lm.resid <- predep.lm$residuals #save residuals

shapiro.test(predep.lm.resid) #use Shapiro Wilk test to assess normality
```

```
##
## Shapiro-Wilk normality test
##
## data:  predep.lm.resid
## W = 0.9974, p-value = 0.08176
```

```
# p = 0.08176, fail to reject H0, data are normal
```

```
#plots to assess if data meet assumptions of 2-way ANOVA
#plot(predep.lm)
```

```
#this data looks good, normal, homoscedastic, no outliers
```

```
#I should use Anova function to have control over the sum of squares type, but Tukey Kramer test won't
```

```
#Tukey-Kramer labels only allow one - per comparison label, so I will replace the - with _
predep$group <- gsub("-", "_", predep$group)
```

```
predep.anova3 <- Anova(lm(shell_length_mm ~ group, data = predep))
predep.anova3
```

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

```
##
```

```
## Response: shell_length_mm
```

```
##          Sum Sq   Df F value    Pr(>F)
```

```
## group      4509.8      8  51.029 < 2.2e-16 ***
## Residuals 11765.3 1065
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
predep.anova <- aov(lm(shell_length_mm ~ group, data = predep))
summary(predep.anova)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## group          8   4510   563.7    51.03 <2e-16 ***
## Residuals    1065  11765    11.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#extract p-value
predep_pval <- summary(predep.anova)[[1]][["Pr(>F)"]][1]
print(predep_pval)
```

```
## [1] 4.998853e-70
```

```
#post-hoc comparison- Tukey-Kramer test
predep.tuk <- TukeyHSD(x=predep.anova,conf.level = 0.95)
```

```
#extract significant letters from TK test and store
require(multcompView)
```

```
predep_cld <- multcompLetters4(predep.anova, predep.tuk)
```

```
predep_dt <- group_by(predep, group) %>%
  summarise(m=mean(shell_length_mm)) %>%
  arrange(desc(m))
```

```
predep_cld <- as.data.frame.list(predep_cld$group)
predep_dt$predep_cld <- predep_cld$Letters
```

```
#make shell length numeric in predep
predep$shell_length_mm <- as.numeric(predep$shell_length_mm)
```

```
#P Adjust
```

```
#adjust p-value to correct for multiple tests. Make vector of p-values from each ANOVA
p_values <- c(all_pval, predep_pval)
```

```
p_values
```

```
## [1] 4.034677e-169 0.000000e+00 1.977514e-01 6.510494e-35 4.998853e-70
```

```
p_adj <- p.adjust(p_values, method = "BH")
```

```
p_adj
```

```
## [1] 1.008669e-168 0.000000e+00 1.977514e-01 8.138117e-35 8.331421e-70
```

```
#plot. fatten = NULL will remove the median lines in each box, allowing me to display the mean instead.
predep_box <- ggplot(data = predep, aes(group, shell_length_mm))+
  geom_boxplot(size = .2, fatten = NULL)+
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2, linewidth = .4, color = "red")+
  stat_summary(fun = mean, geom = "errorbar", aes(ymax = ..y.., ymin = ..y..), width = 0.75, size = .4,
  ylab("Day 78 Shell Length (mm)")+
  xlab("Broodstock Group")+
  labs(subtitle = "p < 0.001")+
  ylim(c(0,32))+
  geom_text(data = predep_dt, size = 3, aes(label = predep_cld, y = 29, x = group), vjust = -0.5)+
  ggtitle("Pre-Field-Deployment (Day 78) Shell Lengths")+
  theme_bw()+
  theme(plot.subtitle = element_text(size = 9), axis.title.y = element_text(size = 11), axis.title.x =
  scale_x_discrete(name = "Broodstock Group", limits = c("H1_LARMIX", "W1_TX", "W2_LA", "W3_FL", "S1_LOLA
```

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

```
predep_box
```

```
## Warning: The dot-dot notation ('..y..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(y)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
## Removed 1 row containing missing values or values outside the scale range
## ('geom_segment()').
```

# Pre-Field-Deployment (Day 78) Shell Lengths

p < 0.001

