

# Color Analysis Code

Sara Shapiro

2025-01-03

## Script for lobster color data analysis

load the following packages: `library(ggplot2)` `library(ggfortify)` `library(tidyverse)` `library(ggpubr)` `require(patchwork)`  
`library(gginnards)` `library(reshape)` `library(Ternary)` `library(ggrain)` `library(ggsignif)` `library(dplyr)` `library(ggtern)`

Note: In the following examples, I only included dummy data for one temperature.

## Set the working directory and import the .csv file

```
setwd("C:/Users/sks/Documents/Tepolt Lab/2024_PL_lab_images/github")  
data <- read.csv("dummy_data.csv")
```

## Set up data frames for each ROI

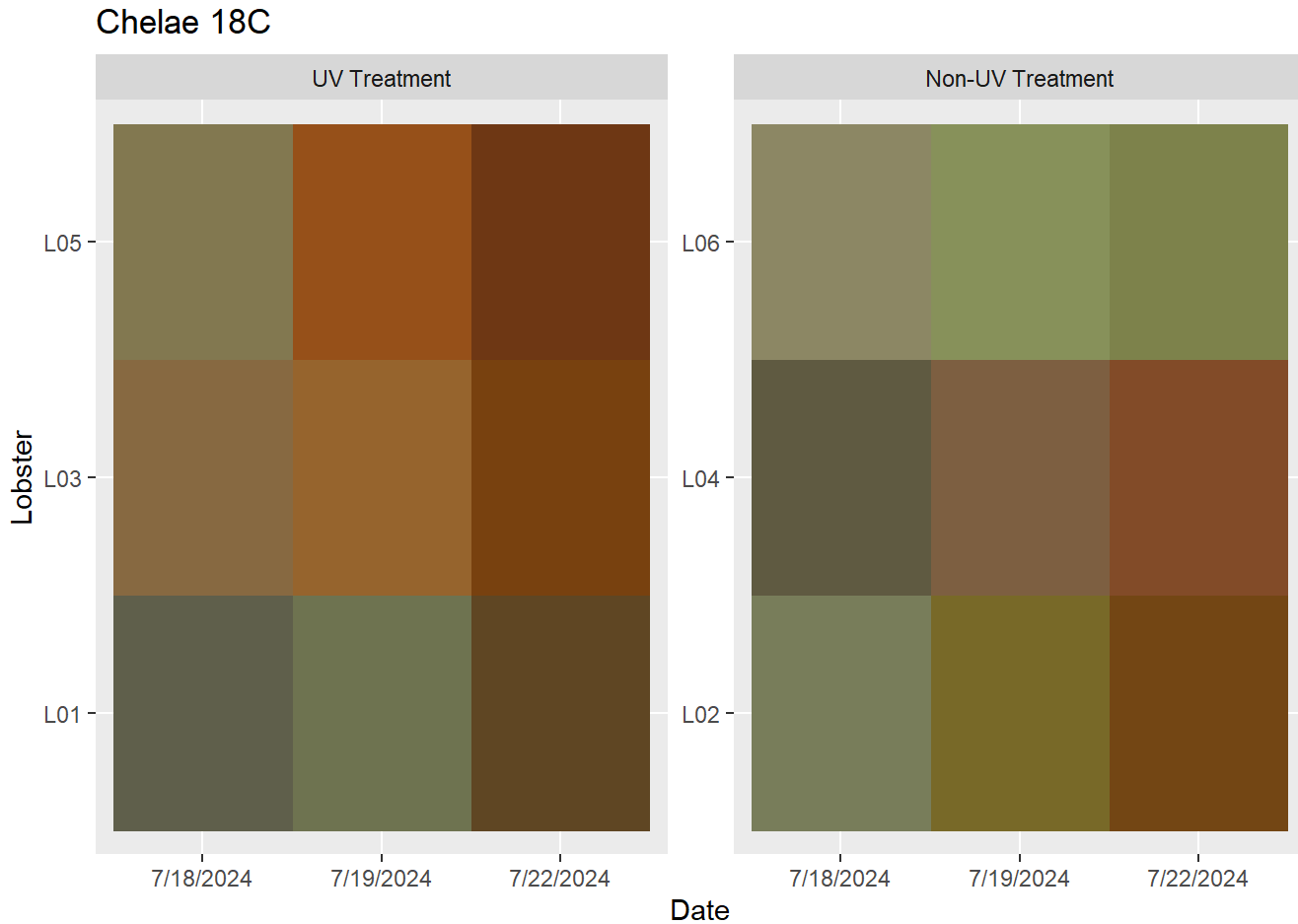
```
CH_R <- c(data$CH_R)  
CH_G <- c(data$CH_G)  
CH_B <- c(data$CH_B)  
lob <- c(data$Lobster)  
date <- c(data$Date)  
uv <- c(data$UV_Treatment)  
temp <- c(data$Temp)  
df_CH <- data.frame(lob,temp,date,uv,CH_R,CH_G,CH_B)  
  
CTM_R <- c(data$CTM_R)  
CTM_G <- c(data$CTM_G)  
CTM_B <- c(data$CTM_B)  
df_CTM <- data.frame(lob,temp,date,uv,CTM_R,CTM_G,CTM_B)  
  
CTE_R <- c(data$CTE_R)  
CTE_G <- c(data$CTE_G)  
CTE_B <- c(data$CTE_B)  
df_CTE <- data.frame(lob,temp,date,uv,CTE_R,CTE_G,CTE_B)  
  
AS2_R <- c(data$AS.2_R)  
AS2_G <- c(data$AS.2_G)  
AS2_B <- c(data$AS.2_B)  
df_AS2 <- data.frame(lob,temp,date,uv,AS2_R,AS2_G,AS2_B)  
  
AS5_R <- c(data$AS.5_R)  
AS5_G <- c(data$AS.5_G)  
AS5_B <- c(data$AS.5_B)  
df_AS5 <- data.frame(lob,temp,date,uv,AS5_R,AS5_G,AS5_B)
```

# Color Grid Example

```
uv.labs <- c("UV Treatment", "Non-UV Treatment") # formatting to re-label facet titles
names(uv.labs) <- c("UV", "X") # connecting the old labels with the new labels
```

```
ch18_cg <- ggplot(df_CH, aes(x = date, y = lob)) +
  geom_raster(aes(fill = rgb(CH_R, CH_G, CH_B, maxColorValue = 255))) +
  scale_fill_identity() +
  facet_wrap(. ~ uv, scales = "free_y", labeller = labeller(uv = uv.labs)) +
  labs(title = "Chelae 18C", x = "Date", y = "Lobster")
```

ch18\_cg



# Ternary Plot using ggtern Example

Note: You can ignore the Warning message that states "Ignoring unknown aesthetics: z", it doesn't impact the graph result

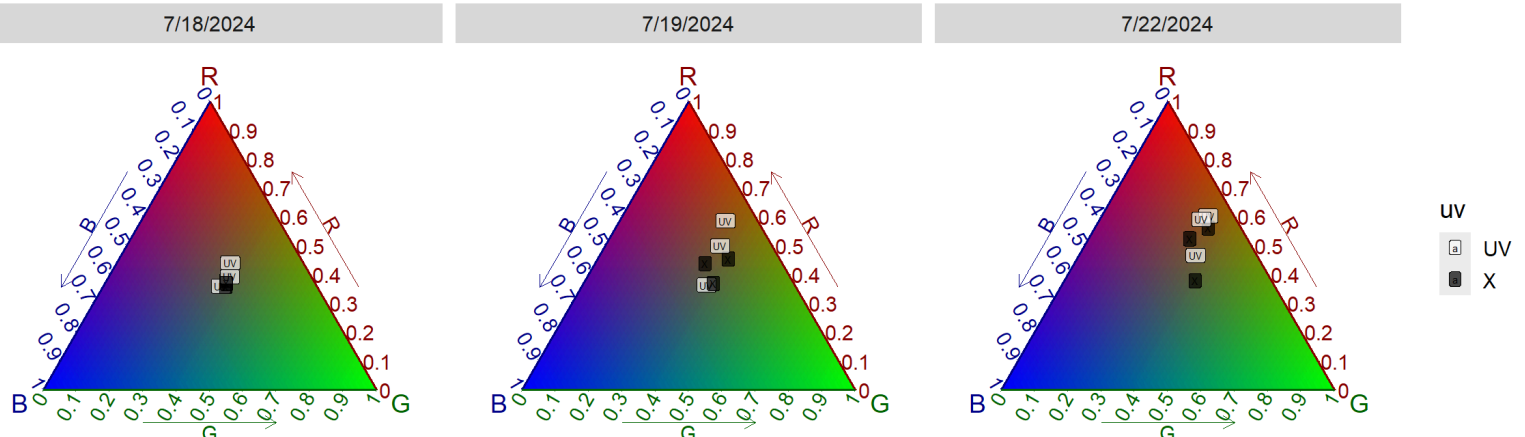
```
# Create the base color spectrum that will color the background of the Ternary plot
all.colors <- expand.grid(R = 0:100, G = 0:100, B = 0:100)
all.colors <- all.colors[rowSums(all.colors) == 100, ]
all.colors <- all.colors / 100
all.colors$hex <- with(all.colors, rgb(R, G, B))

# Create Ternary Plot
ch_tern_d1 <- ggtern(all.colors, aes(x = B, y = R, z = G)) + # this plots the background
  geom_point(aes(color = hex), size = 3, shape = 17) + # this needs to reference all.colors in order for the
  # background to properly fill
  geom_label(data = df_CH, inherit.aes = FALSE, # this is where you plot your data points
    mapping = aes(x = CH_B, y = CH_R, z = CH_G, label = uv, fill = uv),
    size = 2, alpha = 0.7) +
  scale_fill_manual(values=c("white", "black")) +
  facet_grid(. ~ date) + # you could delete the facet if you want to create one graph for each date (you would
  # just need to create a new data set for each date and change the geom_label accordingly)
  scale_color_identity() +
  scale_L_continuous(breaks = seq(0, 1, 0.1), labels = seq(0, 1, 0.1)) +
  scale_T_continuous(breaks = seq(0, 1, 0.1), labels = seq(0, 1, 0.1)) +
  scale_R_continuous(breaks = seq(0, 1, 0.1), labels = seq(0, 1, 0.1)) +
  theme_rgbw(14) +
  ggtitle("Chelae 18C Day 1")
```

```
## Warning in geom_label(data = df_CH, inherit.aes = FALSE, mapping = aes(x =
## CH_B, : Ignoring unknown aesthetics: z
```

```
ch_tern_d1
```

## Chelae 18C Day 1



# Ternary Plot Example using Ternary Package

Note: You can add as many Ternary plots to one full graphing area at a time, just adjust the graphing space mfrow parameters to accommodate all plots. I haven't found a way to assign a figure data name to the plot, so you have to manually save it in the Plots viewer (I find zooming in helps with the dimensions).

```

ch18_df <- df_CH %>% # creates a new column with the red, green, and blue values together
  rowwise %>%
  mutate(ch = list(c(CH_R,CH_G,CH_B))) %>%
  ungroup

# Create data sets for each day, separated by UV treatment
ch_d1 <- filter(ch18_df, date == "7/18/2024" & uv == "X")
ch_d1_UV <- filter(ch18_df, date == "7/18/2024" & uv == "UV")
ch_d2 <- filter(ch18_df, date == "7/19/2024" & uv == "X")
ch_d2_UV <- filter(ch18_df, date == "7/19/2024" & uv == "UV")
ch_d5 <- filter(ch18_df, date == "7/22/2024" & uv == "X")
ch_d5_UV <- filter(ch18_df, date == "7/22/2024" & uv == "UV")

# Create graphing space for Ternary Plots
par(mfrow = c(1,3), mar = rep(0.3, 4)) # for mfrow = c(,), the first number is rows and second is columns;
this sets up the space to plot the ternary plots

# Make Day 1 Initial Ternary Plot
TernaryPlot(alab = "Redder \u2192", blab = "Greener \u2192", clab = "\u2190 Blue",
  lab.col = c("red", "darkgreen", "blue"),
  main = "CH - Day 1", # Title
  point = "up", lab.cex = 1, grid.minor.lines = 0,
  grid.lty = "solid", col = rgb(0.9, 0.9, 0.9), grid.col = "white",
  axis.col = rgb(0.6, 0.6, 0.6), ticks.col = rgb(0.6, 0.6, 0.6),
  axis.rotate = FALSE,
  padding = 0.08,
  axis.labels = seq(0, 10, by = 1),
  xlim=NULL,
  isometric=TRUE,
)

# Color the background
cols <- TernaryPointValues(rgb)
ColourTernary(cols, spectrum = NULL)

# Add data points
ch_d1_18_points <- ch_d1$ch
AddToTernary(points, ch_d1_18_points, pch = 1, cex = 1.5, col = c("black") , #can change point type and col
or
  bg = vapply(ch_d1_18_points,
    function (x) rgb(x[1], x[2], x[3], 128,
      maxColorValue = 255),
    character(1))
)

ch_d1_18_UV_points <- ch_d1_UV$ch
AddToTernary(points, ch_d1_18_UV_points, pch = 1, cex = 1.5, col = c("white") ,
  bg = vapply(ch_d1_18_UV_points,
    function (x) rgb(x[1], x[2], x[3], 128,
      maxColorValue = 255),
    character(1))
)

# Make Day 2 Initial Ternary Plot

```

```

TernaryPlot(alab = "Redder \u2192", blab = "Greener \u2192", clab = "\u2190 Blue",
  lab.col = c("red", "darkgreen", "blue"),
  main = "CH - Day 2", # Title
  point = "up", lab.cex = 1, grid.minor.lines = 0,
  grid.lty = "solid", col = rgb(0.9, 0.9, 0.9), grid.col = "white",
  axis.col = rgb(0.6, 0.6, 0.6), ticks.col = rgb(0.6, 0.6, 0.6),
  axis.rotate = FALSE,
  padding = 0.08,
  axis.labels = seq(0, 10, by = 1),
  xlim=NULL,
  isometric=TRUE,
)

```

*# Color the background*

```

cols <- TernaryPointValues(rgb)
ColourTernary(cols, spectrum = NULL)

```

*# Add data points*

```

ch_d2_18_points <- ch_d2$ch
AddToTernary(points, ch_d2_18_points, pch = 1, cex = 1.5, col = c("black") ,
  bg = vapply(ch_d2_18_points,
    function (x) rgb(x[1], x[2], x[3], 128,
      maxColorValue = 255),
    character(1))
)

```

```

ch_d2_18_UV_points <- ch_d2_UV$ch
AddToTernary(points, ch_d2_18_UV_points, pch = 1, cex = 1.5, col = c("white") ,
  bg = vapply(ch_d2_18_UV_points,
    function (x) rgb(x[1], x[2], x[3], 128,
      maxColorValue = 255),
    character(1))
)

```

*# Make Day 2 Initial Ternary Plot*

```

TernaryPlot(alab = "Redder \u2192", blab = "Greener \u2192", clab = "\u2190 Blue",
  lab.col = c("red", "darkgreen", "blue"),
  main = "CH - Day 5", # Title
  point = "up", lab.cex = 1, grid.minor.lines = 0,
  grid.lty = "solid", col = rgb(0.9, 0.9, 0.9), grid.col = "white",
  axis.col = rgb(0.6, 0.6, 0.6), ticks.col = rgb(0.6, 0.6, 0.6),
  axis.rotate = FALSE,
  padding = 0.08,
  axis.labels = seq(0, 10, by = 1),
  xlim=NULL,
  isometric=TRUE,
)

```

*# Color the background*

```

cols <- TernaryPointValues(rgb)
ColourTernary(cols, spectrum = NULL)

```

*# Add data points*

```

ch_d5_18_points <- ch_d5$ch

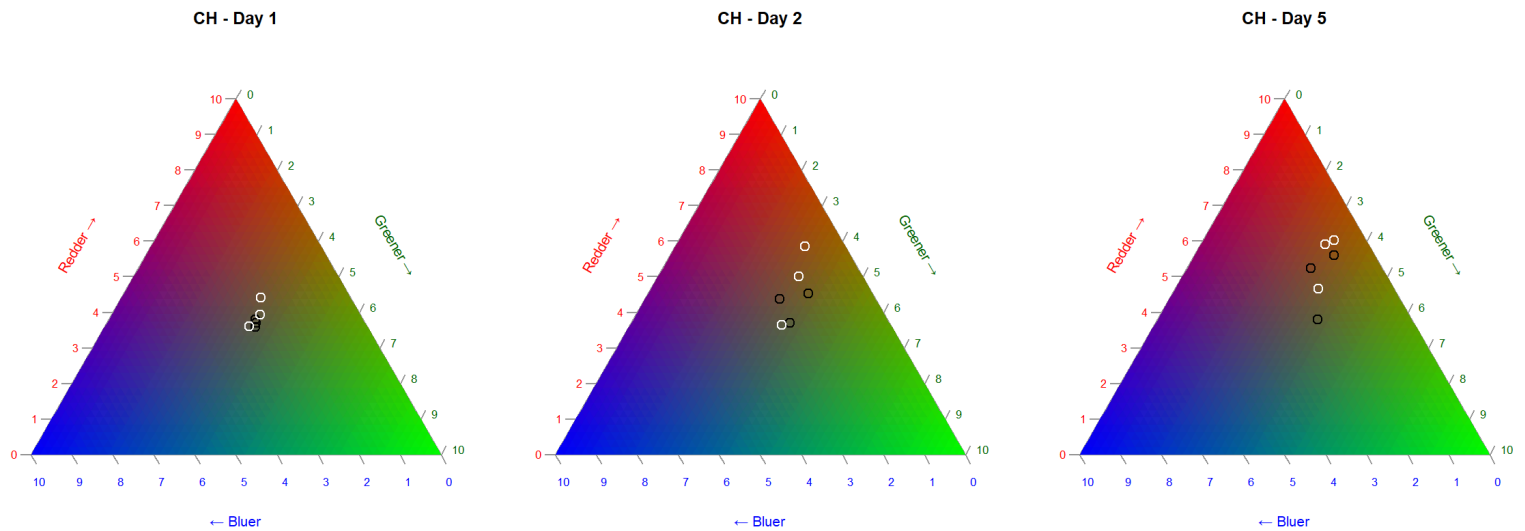
```

```

AddToTernary(points, ch_d5_18_points, pch = 1, cex = 1.5, col = c("black") ,
             bg = vapply(ch_d5_18_points,
                         function (x) rgb(x[1], x[2], x[3], 128,
                                           maxColorValue = 255),
                         character(1))
)

ch_d5_18_UV_points <- ch_d5_UV$ch
AddToTernary(points, ch_d5_18_UV_points, pch = 1, cex = 1.5, col = c("white") ,
             bg = vapply(ch_d5_18_UV_points,
                         function (x) rgb(x[1], x[2], x[3], 128,
                                           maxColorValue = 255),
                         character(1))
)

```



## Boxplots Example

Graphs the Red, Green, and Blue values for each region across time to see if there's any significant change in any of the colors between treatments over time. Interpretation of results: if colors decrease, then overall pigment is getting darker; if colors increase, then overall pigment is getting lighter; increase and/or no change in one color while other colors decrease would result in pigment change towards the increased and/or no change color (ex. Red inc/same, Green & Blue dec -> overall pigment would be redder).

Note: Significance in these plots is comparing treatments over time, not within each treatment over time. Roughly speaking, it's comparing the slopes of the geom\_smooth function for each treatment.

```

# Create data frames for each color
df_R <- data.frame(lob,date,temp,uv,CH_R,CTM_R,CTE_R,AS2_R,AS5_R)
df_R$uv <- factor(df_R$uv, levels = c("X","UV"))
df_G <- data.frame(lob,date,temp,uv,CH_G,CTM_G,CTE_G,AS2_G,AS5_G)
df_G$uv <- factor(df_G$uv, levels = c("X","UV"))
df_B <- data.frame(lob,date,temp,uv,CH_B,CTM_B,CTE_B,AS2_B,AS5_B)
df_B$uv <- factor(df_B$uv, levels = c("X","UV"))

kw_chR <- kruskal.test(CH_R ~ uv, data = df_R) # get Kruskal-Wallis value comparing color and UV treatment
pval_kw_chR <- kw_chR$p.value # create p value for Kruskal-Wallis test for plot

kw_chG <- kruskal.test(CH_G ~ uv, data = df_G)
pval_kw_chG <- kw_chG$p.value

kw_chB <- kruskal.test(CH_B ~ uv, data = df_B)
pval_kw_chB <- kw_chB$p.value

bp_CH_R <- ggplot(df_R,aes(x=factor(date), y=CH_R, color=uv, fill=uv)) +
  geom_boxplot(aes(fill = uv), alpha = 0.50) +
  scale_fill_manual(values = c("gray38","white")) +
  labs(y = "CH_R", x = "Date") + ggtitle("18C Chelae Red") +
  scale_color_manual(values = c("black", "black")) +
  theme(legend.text=element_text(size=12), legend.key.size=unit(2, 'cm')) +
  theme(axis.text.x = element_text(angle=60, hjust=1)) +
  theme(text = element_text(size = 15), plot.title = element_text(color="firebrick"))+
  ylim(0,255) +
  geom_smooth(aes(group=uv, linetype = uv), linewidth = 1, method = "glm", se = FALSE)+
  annotate("text", x = Inf, y = Inf,
    label = paste("p kw =", format.pval(pval_kw_chR, digits = 3)),
    hjust = 1, vjust = 1, size = 5)

bp_CH_G <- ggplot(df_G,aes(x=factor(date), y=CH_G, color=uv, fill=uv)) +
  geom_boxplot(aes(fill = uv), alpha = 0.50) +
  scale_fill_manual(values = c("gray38","white")) +
  labs(y = "CH_R", x = "Date") + ggtitle("18C Chelae Green") +
  scale_color_manual(values = c("black", "black")) +
  theme(legend.text=element_text(size=12), legend.key.size=unit(2, 'cm')) +
  theme(axis.text.x = element_text(angle=60, hjust=1)) +
  theme(text = element_text(size = 15), plot.title= element_text(color="chartreuse4"))+
  ylim(0,255)+
  geom_smooth(aes(group=uv, linetype = uv), linewidth = 1, method = "glm", se = FALSE)+
  annotate("text", x = Inf, y = Inf,
    label = paste("p kw =", format.pval(pval_kw_chG, digits = 3)),
    hjust = 1, vjust = 1, size = 5)

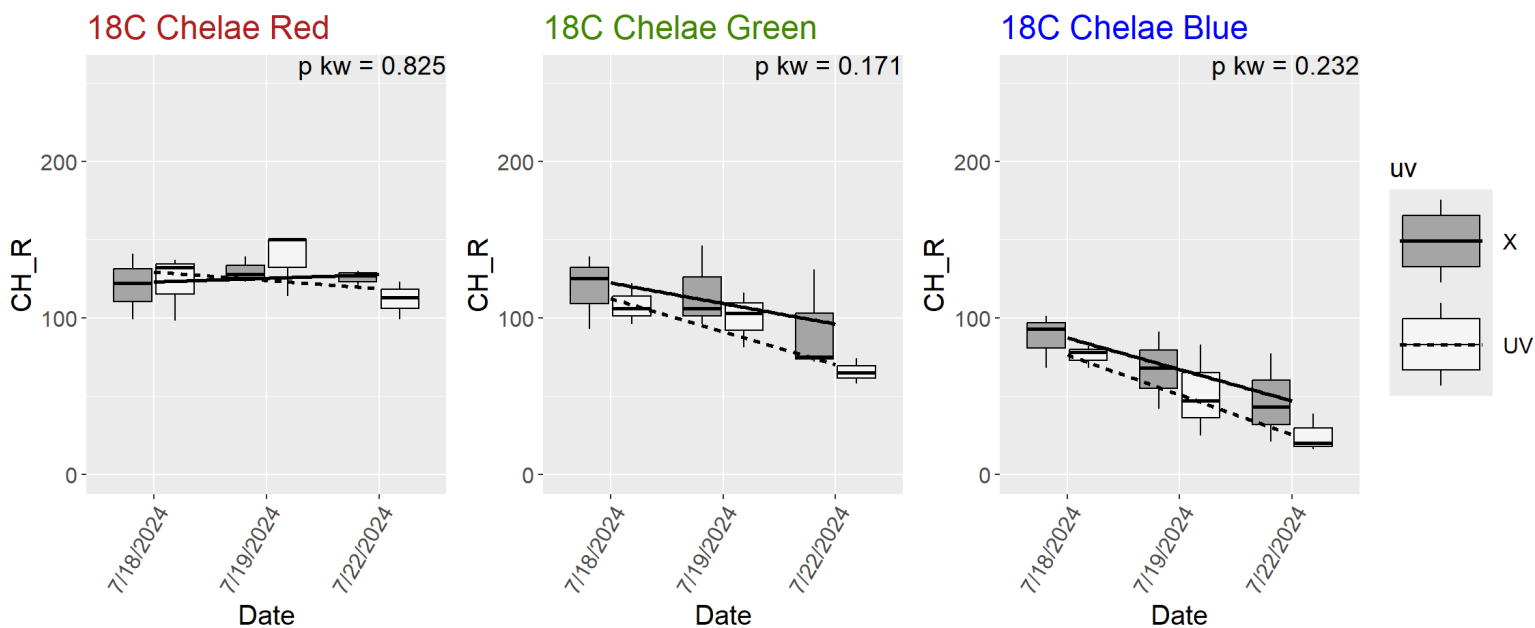
bp_CH_B <- ggplot(df_B,aes(x=factor(date), y=CH_B, color=uv, fill=uv)) +
  geom_boxplot(aes(fill = uv), alpha = 0.50) +
  scale_fill_manual(values = c("gray38","white")) +
  labs(y = "CH_R", x = "Date") + ggtitle("18C Chelae Blue") +
  scale_color_manual(values = c("black", "black")) +
  theme(legend.text=element_text(size=12), legend.key.size=unit(2, 'cm')) +
  theme(axis.text.x = element_text(angle=60, hjust=1)) +
  theme(text = element_text(size = 15), plot.title= element_text(color="blue"))+
  ylim(0,255)+

```

```
geom_smooth(aes(group=uv, linetype = uv), linewidth = 1, method = "glm", se = FALSE)+
annotate("text", x = Inf, y = Inf,
        label = paste("p kw =", format.pval(pval_kw_chB, digits = 3)),
        hjust = 1, vjust = 1, size = 5)
```

```
CH_bp <- ((bp_CH_R + bp_CH_G + bp_CH_B) + plot_layout(guides = "collect"))
CH_bp
```

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



## Line Plots Example

These line plots follow individuals over time and the significance is the same as the boxplots.



```

# Create the line plots (using the same Kruskal-Wallis P-value from above)
lp_CH_R <- ggplot(df_CH,aes(date,CH_R, fill=uv, color=uv, group=lob)) +
  geom_point(aes(fill=uv)) + geom_line(aes(color=uv), alpha=0.4) +
  stat_smooth(aes(group=uv), linewidth = 1.5, method = "glm", se = FALSE) +
  stat_summary(fun = mean, geom = "line", aes(group = uv, color = uv)) +
  theme_classic() +
  theme(axis.text.x = element_text(angle=60, hjust=1)) +
  scale_fill_manual(values=c("mediumorchid3", "deepskyblue3")) +
  scale_color_manual(values=c("mediumorchid3", "deepskyblue3")) +
  ggtitle("Red Chelae 18C Values") +
  theme(plot.title = element_text(color="firebrick")) +
  ylim(0,255)+
  annotate("text", x = Inf, y = Inf,
    label = paste("p kw =", format.pval(pval_kw_chR, digits = 3)),
    hjust = 1, vjust = 1, size = 5)

lp_CH_G <- ggplot(df_CH,aes(date,CH_G, fill=uv, color=uv, group=lob)) +
  geom_point(aes(fill=uv)) + geom_line(aes(color=uv), alpha=0.4) +
  geom_smooth(aes(group=uv), linewidth = 1.5, method = "glm", se = FALSE) +
  stat_summary(fun = mean, geom = "line", aes(group = uv, color = uv)) +
  theme_classic() +
  theme(axis.text.x = element_text(angle=60, hjust=1)) +
  scale_fill_manual(values=c("mediumorchid3", "deepskyblue3")) +
  scale_color_manual(values=c("mediumorchid3", "deepskyblue3")) +
  ggtitle("Green Chelae 18C Values") +
  theme(plot.title = element_text(color="chartreuse4")) +
  ylim(0,255)+
  annotate("text", x = Inf, y = Inf,
    label = paste("p kw =", format.pval(pval_kw_chG, digits = 3)),
    hjust = 1, vjust = 1, size = 5)

lp_CH_B <- ggplot(df_CH,aes(date,CH_B, fill=uv, color=uv, group=lob)) +
  geom_point(aes(fill=uv)) + geom_line(aes(color=uv), alpha=0.4) +
  geom_smooth(aes(group=uv), linewidth = 1.5, method = "glm", se = FALSE) +
  stat_summary(fun = mean, geom = "line", aes(group = uv, color = uv)) +
  theme_classic() +
  theme(axis.text.x = element_text(angle=60, hjust=1)) +
  scale_fill_manual(values=c("mediumorchid3", "deepskyblue3")) +
  scale_color_manual(values=c("mediumorchid3", "deepskyblue3")) +
  ggtitle("Blue Chelae 18C Values") +
  theme(plot.title = element_text(color="blue")) +
  ylim(0,255)+
  annotate("text", x = Inf, y = Inf,
    label = paste("p kw =", format.pval(pval_kw_chB, digits = 3)),
    hjust = 1, vjust = 1, size = 5)

lp_CH <- ((lp_CH_R + lp_CH_G + lp_CH_B) + plot_layout(guides = "collect"))
lp_CH

```

```

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

```

