

```
---
title: "Bio 313 Spring 24 Behavior Data Analysis"
author: "Wallace"
date: "2024-01-24"
output:
  pdf_document: default
  html_document: default
---

#Ignore this first chunk!
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

#MY NAME:

#MY RESEARCH QUESTION:

#MY SAMPLE:

#MY INDEPENDENT VARIABLE (X AXIS):

#MY INDEPENDENT VARIABLE TYPE (categorical or continuous, if categorical are
there more than two categories?):

#MY DEPENDENT VARIABLE (Y AXIS):

#MY DEPENDENT VARIABLE TYPE (categorical or continuous, if categorical are there
more than two categories?):

#MY PREDICTION:

#MY P VALUE (based on step 5 below):

##### Let's get started!

#Step 0: In files (on the right, by plots), click upload and upload the data
mastersheet.
#Click on the file name and select import
#In "name" write "df"

#Step 1: load relevant packages. These packages allow you do do specific
functions, such as generate a graph
```{r}
library(readr)
library(dplyr)
library(ggplot2)
...

#Step 2: import and view the class-wide data file with the data you scored from
your Behavior Day 2 videos
```{r}

View(df)
...

```

```
#Step 3: select only the specific *rows* you need for your question (aka which
videos)
```{r}

mydf<-df %>% filter([COLUMN NAME] == "[CHARACTER]")

...

#Step 4: select only the specific *columns* you need for your question (aka
which metrics)
```{r}

mydf<-mydf %>% select([COLUMN A], [COLUMN B])

...

#Step 5: view your specific data to make sure it looks good
```{r}

View(mydf)

...

#Step 6: Plot your dependent variable for all fish to make sure it looks in the
right range (this will make the most unnecessarily wide boxplot you've ever
seen)
#BONUS: here's a great link to different color names: https://sape.inf.usi.ch/quick-reference/ggplot2/colour
```{r}

ggplot(mydf, aes(x=1, y=[DEPENDENT VARIABLE])) +
geom_boxplot(fill="cornflowerblue",) + geom_jitter(width = 0.01)+
theme_classic()

...

#Step 7a: If your independent variable is continuous, follow these steps below
```{r}

mydf<-na.omit(mydf)
mod <- lm(mydf$[DEPENDENT VARIABLE] ~ mydf$[INDEPENDENT VARIABLE])
p_value <- summary(mod)$coefficients[8]
print(paste("[DEPENDENT VARIABLE] as a function of [INDEPENDENT VARIABLE], p
value = ", round(p_value, 4)))

...

#Step 7b: If your independent variable is categorical with two categories,
follow these steps below
```{r}

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

```
mydf$[INDEPENDENT VARIABLE]<-as.factor(mydf$[INDEPENDENT VARIABLE])
test<-t.test([DEPENDENT VARIABLE] ~ [INDEPENDENT VARIABLE], data = mydf)
pval<-test$p.value
print(paste("[DEPENDENT VARIABLE] as a function of [INDEPENDENT VARIABLE], p
value= ", pval))
```

...

#Step 7c: If your independent variable is categorial with more than two categories, follow these steps below

```
```{r}

mydf<-na.omit(mydf)
mydf$[INDEPENDENT VARIABLE]<-as.factor(mydf$[INDEPENDENT VARIABLE])
test<-aov([DEPENDENT VARIABLE] ~ [INDEPENDENT VARIABLE], data = mydf)
summary(test)
pval<-summary(test)[[1]][["Pr(>F)"]][[1]]
print(paste("[DEPENDENT VARIABLE] as a function of [INDEPENDENT VARIABLE], p
value= ", pval))
```

...

#Step 8a: Plot your result (from 7a)

```
```{r}
ggplot(mydf, aes([INDEPENDENT VARIABLE], [DEPENDENT VARIABLE])) +
  geom_point(size = 4, colour = "blue") + geom_smooth(method = "lm",
  se = FALSE, colour = "gray60") + theme(text = element_text(size = 10)) +
  labs(x = "[INDEPENDENT VARIABLE]", y = "[DEPENDENT VARIABLE]") +
  theme_classic()
```

...

#Step 8b: Plot your result (from 7b)

```
```{r}
ggplot(mydf, aes(x=[INDEPENDENT VARIABLE], y=[DEPENDENT VARIABLE], fill=
[INDEPENDENT VARIABLE])) + geom_boxplot() + geom_point(size = 3)+
  theme_classic() + scale_fill_manual(values=c("red", "yellow"))
```

...

#Step 8c: Plot your result (from 7c)

```
```{r}
ggplot(mydf, aes(x=[INDEPENDENT VARIABLE], y=[DEPENDENT VARIABLE], fill=
[INDEPENDENT VARIABLE])) + geom_boxplot() + geom_point(size = 3)+
  theme_classic() + scale_fill_manual(values=c("red", "orange", "yellow"))
```

...

!(Note: if you want to export your graph, copy the line into the console, and then the resulting graph will show up in the plots panel on the right!)

#Step 9: Repeat this again from step 3, now with the gill flaring information! I recommend you call the new dataset mydf2 or mydfgill or something like that so you don't accidentally overwrite your old information. You can insert new chunks here, or you can make just add code into one big chunk here.

```
#  
#  
#  
#  
#
```

#Step 10: When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

## Note on R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.