FISH 481 Ichthyology Respirometry Lab Module

Jon Masingale

12/4/2020

Intro

Last week, both Ichthyology lab sections collected respirometry data on rainbow trout (Oncorchynchus mykiss). The data were then compiled in preparation for analysis. In this document, you will learn how to analyze the data you have collected using a statistical software called R Programming. Specifically, we will be focusing on ventilation with temperature treatments and ventilation in relation to dissolved oxygen (Objectives 1 and 2 respectively, from last week's lab hand-out).

R Programming is effective not only for statistically and visually analyzing data, but for creating reproducible script to be used for both current and future data sets. Reproduction comes from the ability to save scripts for future use, and also allows for manipulation of raw data without changing the original file.

For this exercise, you should have the latest version of R Programming (https://cran.r-project.org/bin/windows/base/) and R studio (https://rstudio.com/products/rstudio/download/) installed on your computer. You will need R Studio open for the script we will be using.

Loading packages

For analyzing these data, we will primarily be visualizing, so let's start by loading a package called "ggplot2", which is commonly used in R to create figures. This package typically produces figures which are more attractive than those produced in base R.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

Load the data

Now we will read in the data file provided using the following code chunk. Make sure to change the directory to the location where you downloaded the data file.

```
respiration <- read.csv("C:/Users/Jawn/OneDrive/School/2020-2021/FISH-504/Final_Project/Respiration_2018
                        na.strings = c("NA", ""))
```

Checking data structure

If we use the "head" function, we can read each column header along with the first six rows of each entry. For the purpose of this exercise, we are interested in the columns "Obj1.Minutes", "Treatment", "Obj1.Ventil", and "Temp", "FL", and "Mass".

Depending on the values entered in the different variable columns of the raw data file, sometimes our variables may not be categorized correctly in order for us to be able to analyze the data properly. Using the following code shows us the structure of the data file, indicating the category of each variable.

```
head(respiration)
```

```
##        Date     Day Clock Obj1.Minutes Treatment Obj1.Ventil Temperature Temp
## 1 2/27/2018 Tuesday  0:00            0   Control       86.67        5.41    5
## 2 2/27/2018 Tuesday  2:00            2   Control       73.33        5.41    5
## 3 2/27/2018 Tuesday  4:00            4   Control       69.33        5.41    5
## 4 2/27/2018 Tuesday  6:00            6   Control       69.33        5.41    5
## 5 2/27/2018 Tuesday  8:00            8   Control       72.00        5.41    5
## 6 2/27/2018 Tuesday 10:00           10   Control       70.66        5.41    5
##          Group Obj2.Time Obj2.Minutes Obj2.Ventil BA.Ventil.Diff Day.Obj3
## 1 Collin/Aspen      0:00          0.0          88             NA  Tuesday
## 2 Collin/Aspen      1:30          1.5          76             NA  Tuesday
## 3 Collin/Aspen      3:00          3.0          72             NA  Tuesday
## 4 Collin/Aspen      4:30          4.5          72             NA  Tuesday
## 5 Collin/Aspen      6:00          6.0          68          -2.66  Tuesday
## 6 Collin/Aspen      <NA>           NA          NA             NA Thursday
##   Obj.3.Temp Obj3.Treat  FL  Mass Hematocrit Glucose Hemacrit.Diff Glucose.Diff
## 1          5    Control  NA    NA         38     162            NA          -25
## 2         18    Control 124 22.00         28     149            NA           42
## 3         15    Control 118 15.90         35     144             5           24
## 4         22    Control  NA    NA         41     126           -19          -28
## 5         10    Control  NA    NA         NA      NA            NA           NA
## 6          5    Control 119 19.57         38      98             0            0
```

```
str(respiration)
```

```
## 'data.frame':    86 obs. of  22 variables:
##  $ Date          : chr  "2/27/2018" "2/27/2018" "2/27/2018" "2/27/2018" ...
##  $ Day           : chr  "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
##  $ Clock         : chr  "0:00" "2:00" "4:00" "6:00" ...
##  $ Obj1.Minutes  : int  0 2 4 6 8 10 0 2 4 6 ...
##  $ Treatment     : chr  "Control" "Control" "Control" "Control" ...
##  $ Obj1.Ventil   : num  86.7 73.3 69.3 69.3 72 ...
##  $ Temperature   : num  5.41 5.41 5.41 5.41 5.41 ...
##  $ Temp          : int  5 5 5 5 5 5 18 18 18 18 ...
##  $ Group         : chr  "Collin/Aspen" "Collin/Aspen" "Collin/Aspen" "Collin/Aspen" ...
##  $ Obj2.Time     : chr  "0:00" "1:30" "3:00" "4:30" ...
##  $ Obj2.Minutes  : num  0 1.5 3 4.5 6 NA 0 2 4 6 ...
##  $ Obj2.Ventil   : int  88 76 72 72 68 NA 132 156 144 156 ...
##  $ BA.Ventil.Diff: num  NA NA NA NA -2.66 NA NA NA NA NA ...
##  $ Day.Obj3      : chr  "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
##  $ Obj.3.Temp    : int  5 18 15 22 10 5 10 15 22 5 ...
##  $ Obj3.Treat    : chr  "Control" "Control" "Control" "Control" ...
##  $ FL            : int  NA 124 118 NA NA 119 128 125 130 NA ...
##  $ Mass          : num  NA 22 15.9 NA NA ...
##  $ Hematocrit    : int  38 28 35 41 NA 38 39 42 50 38 ...
##  $ Glucose       : int  162 149 144 126 NA 98 105 95 96 162 ...
##  $ Hemacrit.Diff : int  NA NA 5 -19 NA 0 0 13 -10 NA ...
##  $ Glucose.Diff  : int  -25 42 24 -28 NA 0 -52 45 0 -25 ...
```

Changing structure of a variable

All of these variables are classified as either "numeric" or "integer". This is good for all except "Temp"; we need this variable to be a factor, as this was a factorial treatment in the experiment. You can use the following code chunk to do this.

The code is instructing R to recognize the variable "Temp" within the dataset "respiration" by connecting them with a dollar sign. The arrow instructs R to essentially modify the variable with the subsequent script. the "as.factor" means "change this variable to a factor."

```
respiration$Temp <- as.factor(respiration$Temp)
```

Creating a linear model

To analyze some basic statistics on this dataset, we can easily create linear models using the "lm" function, which allows us to quickly view summary statistics of the response variable against a given explanatory variable. The first part of the code lines is creating an object (respiration temp <- ; respiration.time <-). The first line creates a model oxygen consumption rate (Obj1.Ventil) as the response variable and temperature treatment (Temp) as the explanatory variable, while the second model uses time (Obj1.Minutes) as the explanatory variable.

```
respiration.temp <- lm(Obj1.Ventil ~ Temp, data = respiration)
respiration.time <- lm(Obj1.Ventil ~ Obj1.Minutes, data = respiration)
```

We can then view summary statistics of the models with the "summary" function.

```
respiration.temp <- lm(Obj1.Ventil ~ Temp + Treatment, data = respiration)
respiration.time <- lm(Obj1.Ventil ~ Obj1.Minutes + Treatment, data = respiration)
```

For some quick statistical analysis, we can perform an ANOVA test using the "anova" function. While we won't go any further into stats here, quickly analyze the p-values of these ANOVA tests. What does this tell you about the effect of temperature on oxygen consumption rate? How about the effect of time?

```
anova(respiration.temp)
```

```
## Analysis of Variance Table
##
## Response: Obj1.Ventil
##           Df Sum Sq Mean Sq  F value Pr(>F)
## Temp       5  67073 13414.6 118.8114 <2e-16 ***
## Treatment  1    119   118.6   1.0504 0.3086
## Residuals 79   8920   112.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(respiration.time)
```

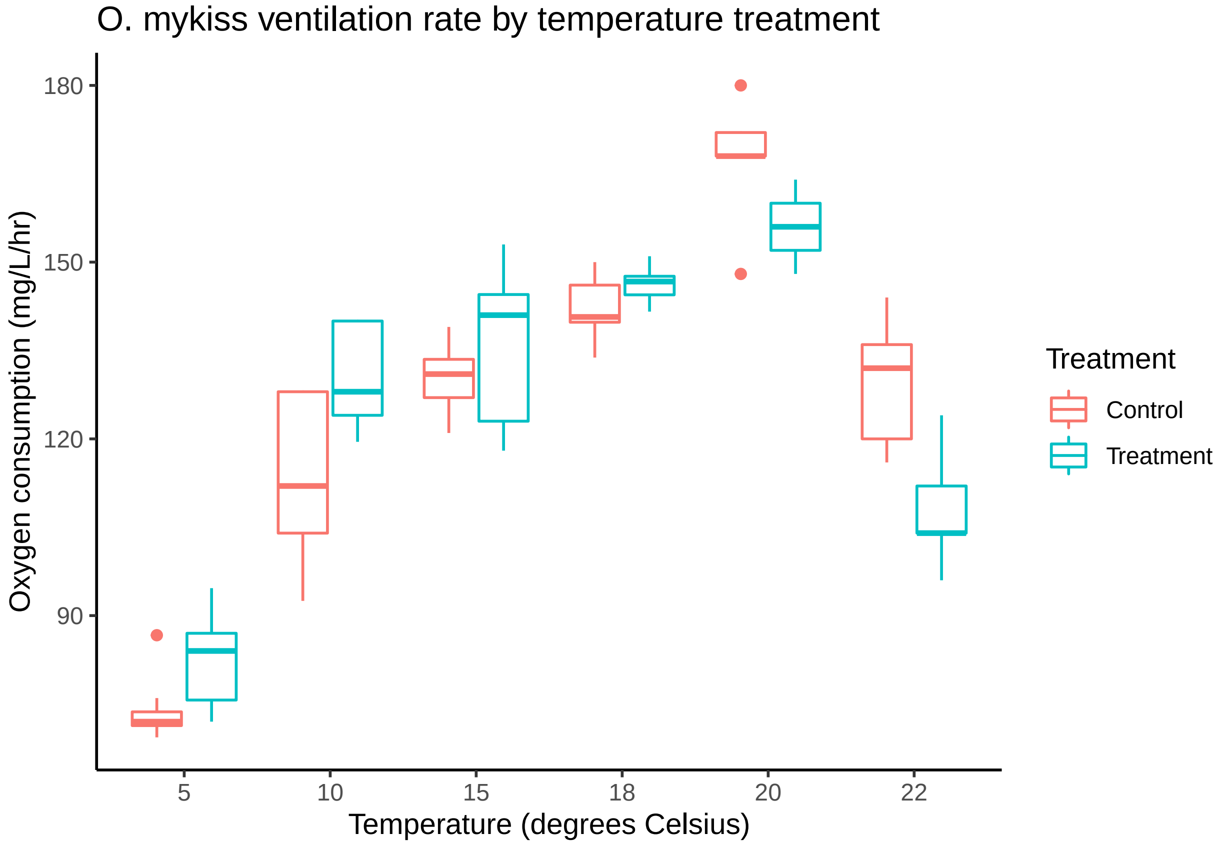
```
## Analysis of Variance Table
##
## Response: Obj1.Ventil
##              Df Sum Sq Mean Sq F value Pr(>F)
## Obj1.Minutes  1    111  110.71  0.1211 0.7287
## Treatment     1    119  118.59  0.1297 0.7196
## Residuals    83  75882  914.24
```

Objective 1 Results

One quick way to easily visualize data is to create a box plot. Here we can take a look at the visual differences of ventilation rate means across temperatures, displaying the difference of respiration of fish in the treatment group (depleted dissolved oxygen) and the control group. Use the following code chunk.

We can see that as temperature increases, ventilation rate also increases, but then drops off again at a point. Can you think of a physiological reason why this might happen?

```r
ggplot(respiration, aes(x=Temp, y=Obj1.Ventil, color=Treatment)) +
  geom_boxplot() +
  labs(title = "O. mykiss ventilation rate by temperature treatment",
       x = "Temperature (degrees Celsius)",
       y = "Oxygen consumption (mg/L/hr)") +
  theme_classic()
```



When using the "ggplot2" package, any plot can be started by using the function "ggplot". Within the parentheses, you will first specify the data set being used. then use the function "aes" for specifying the aesthetics of the plot. Choose the names of the columns in your data set for the x and y axis. In this plot, we can specify a color to indicate treatment versus control means by using the "color" function and calling the variable "Treatment". The plot also includes a title and axis labels. Changing from the default theme to "theme_classic" simply gets rid of grid lines and uses a plain white background, creating a cleaner looking plot.

Note: when using ggplot, each line of code following a "+" is adding another layer, so it is important to be aware of the order that plot elements are created.

Analyzing ventilation post-handling

Does ventilation rate drop over time, post-release?

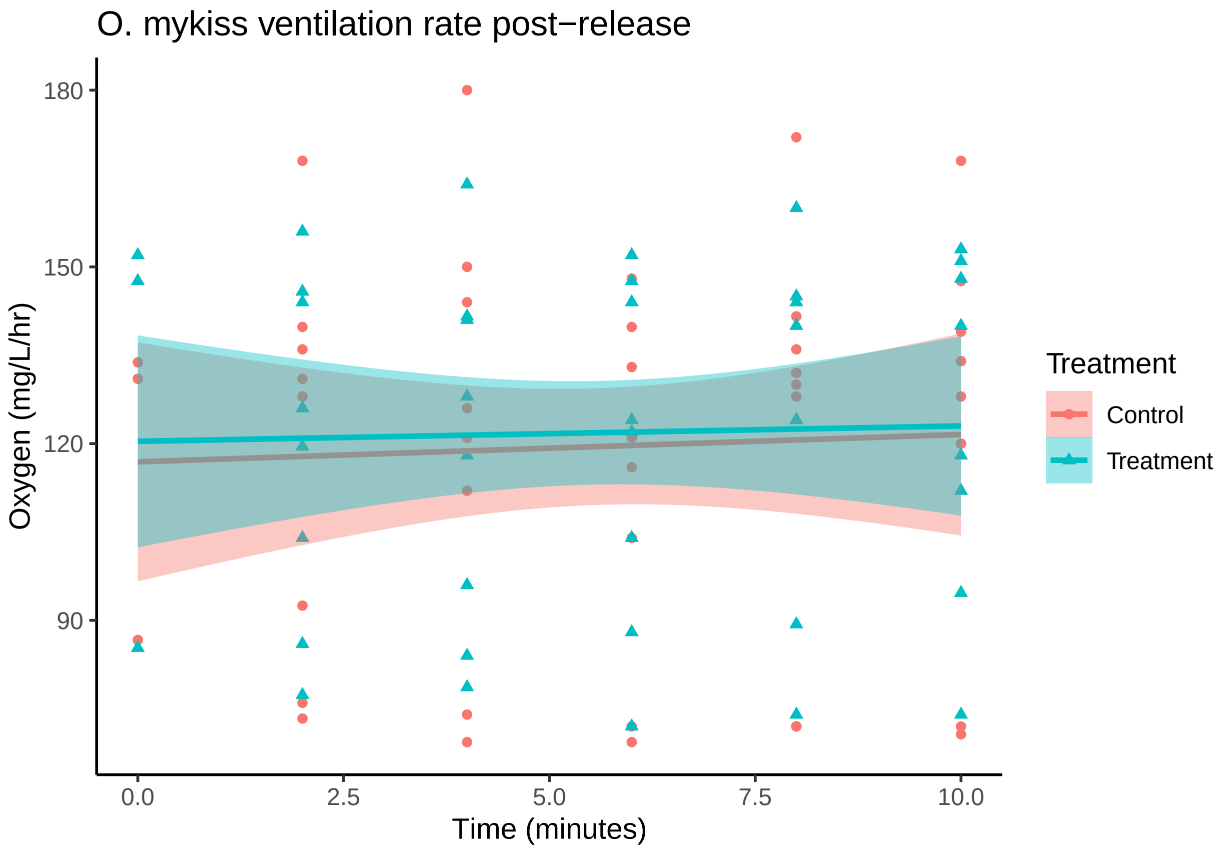
H(o): Handling stress will cause higher initial ventilation, but ventilation will decrease over time.

Use the following code chunk to visualize this. We use ventilation as the response variable and time as the explanatory variable. Again, we use the "color" function to differentiate control and treatment. The function "geom_point" creates a scatter plot and "geom_smooth" adds a regression line to the plot.

Strictly from visual analysis of this plot, would we reject the null hypothesis?

```
ggplot(respiration, aes(x=Obj1.Minutes, y=Obj1.Ventil, color=Treatment, shape=Treatment)) +
  geom_point() +
  geom_smooth(method='lm', aes(fill=Treatment)) +
  labs(title = "O. mykiss ventilation rate post-release",
       x = "Time (minutes)",
       y = "Oxygen (mg/L/hr)") +
  theme_classic()
```

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



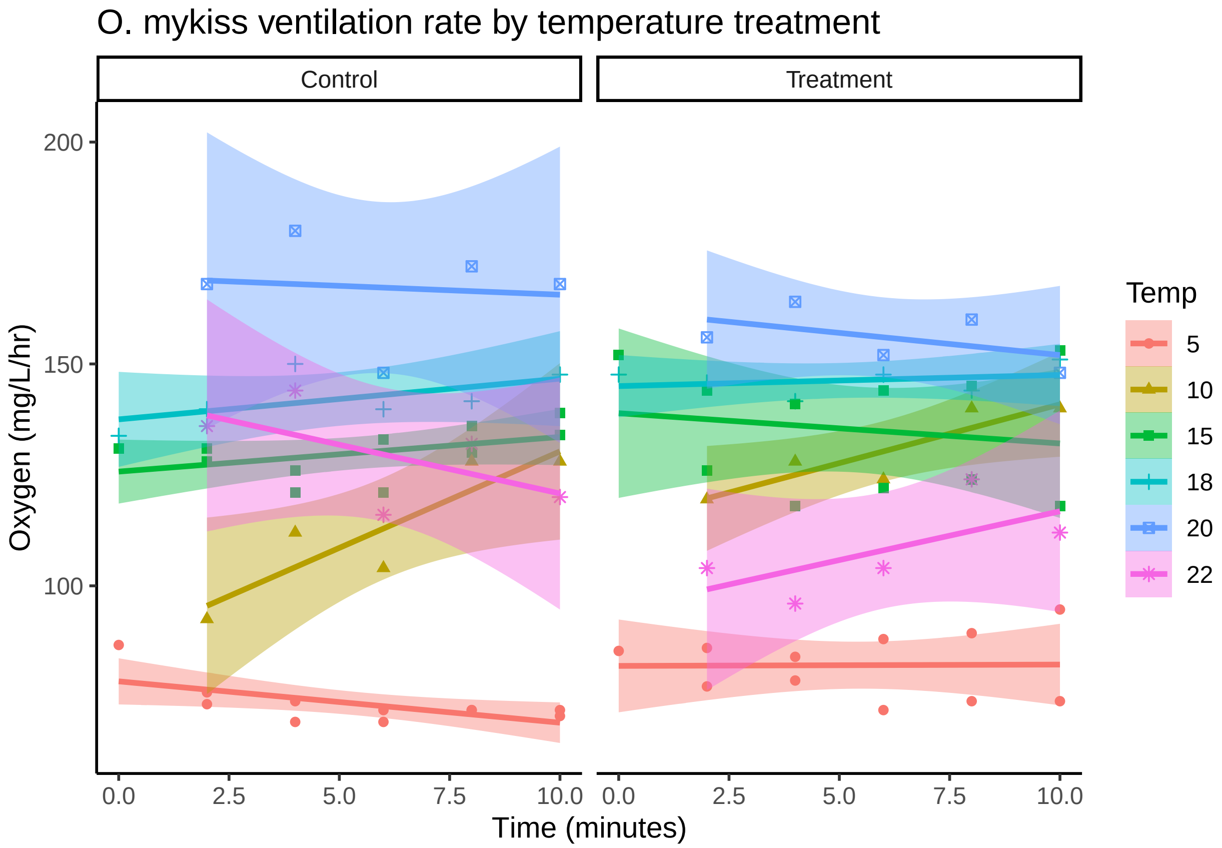
Analyzing temperature effects

Does temperature affect ventilation rate? If so, does this effect change through time?

This plot displays ventilation rate over time, comparing oxygen saturation treatment and control for each temperature treatment. Notice the “facet_wrap” function, with “Treatment” in the parentheses. This separates the treatment and control groups to visually analyze them side by side while still showing the trajectories of individual temperature treatments over time.

```
ggplot(respiration, aes(x=Obj1.Minutes, y=Obj1.Ventil, color=Temp, shape=Temp)) +
  geom_point() +
  geom_smooth(method=’lm’, aes(fill=Temp)) +
  labs(title = "O. mykiss ventilation rate by temperature treatment",
       x = "Time (minutes)",
       y = "Oxygen (mg/L/hr)") +
  facet_wrap( ~ Treatment, ncol = 2, scales = "fixed") +
  theme_classic()
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Now that we have applied some data analysis and visualization to data from the first two objectives of last week’s lab activity, try to adapt this code to produce figures and analyses for Objective 3, analyzing the effect of dissolved oxygen treatment on hematocrit and glucose levels. You can also use these code chunks as a template if you have data of your own to analyze.