

# Homework 4

Victoria Yang

5/25/23

Link to github repository:

[https://github.com/v-yc/ENVS-193DS\\_homework-04\\_Yang-Victoria](https://github.com/v-yc/ENVS-193DS_homework-04_Yang-Victoria)

## 1. Setup

Load in required packages.

```
library(tidyverse)
library(here)
library(performance)
library(broom)
library(flextable)
library(ggeffects)
library(car)
library(naniar)
```

Read in the data and subset data of interest (length and weight of trout perch species).

```
# read in data
fish <- read_csv(here("data", "knb-lter-ntl.6.34", "ntl6_v12.csv")) %>%

# only include troutperch species
filter(spname == "TROUTPERCH") %>%

# select the columns of interest
select(length, weight)
```

## 2. Initial data visualization

In mathematical terms:

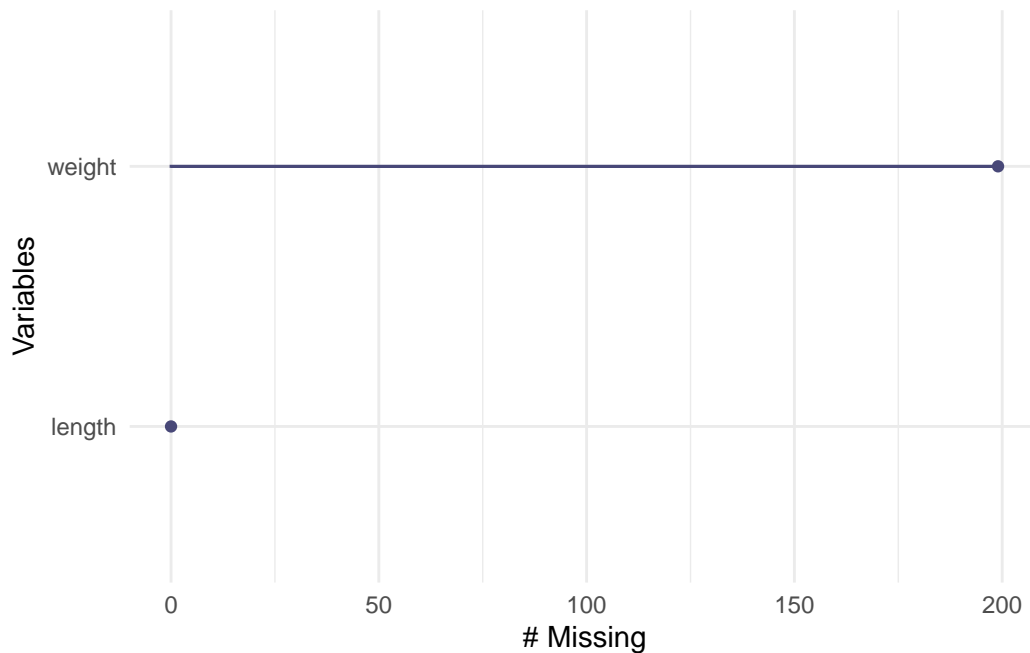
- The null hypothesis is that the slope of the linear model (where fish length is the predictor variable and fish weight is the response variable) is 0.
- The alternative hypothesis is that the slope of the linear model (where fish length is the predictor variable and fish weight is the response variable) is not 0.

In biological terms,

- The null hypothesis is that fish length is not a good predictor of fish weight for trout perch.
- The alternative hypothesis is that fish length is a good predictor of fish weight for trout perch.

Visualize the missing data:

```
gg_miss_var(fish)
```



There are about 200 missing data points for the weight variable, which is significant because there are only 489 observations in this dataset. This would mean that about 41% of the data is missing.

However, we do not know why these data points are missing because this is from an online data set. Additionally, there are still 290 observations left if we remove the missing data, which is a large sample size. As a result, I will remove the NA values by subsetting the data for this analysis because there will be enough data points left to make a good model.

```
fish_subset <- fish %>%  
  
  # drops rows with NA values in the columns specified  
  drop_na(length, weight)
```

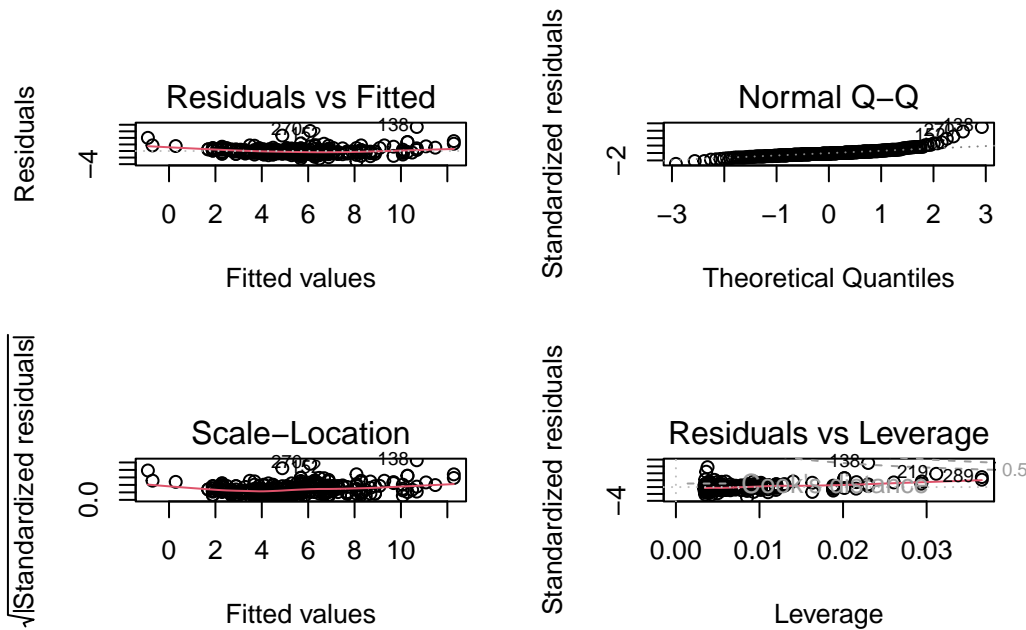
### 3. Create a linear model and check assumptions

Creating a linear model where fish length is the predictor variable and fish weight is the response variable:

```
modelobject <- lm(weight ~ length, data = fish_subset)
```

Check assumptions by plotting residual diagnostic plots.

```
# display diagnostic plots in a 2x2 grid  
par(mfrow = c(2, 2))  
  
# plot diagnostic plots to check linear model assumptions  
plot(modelobject)
```



Interpretation of diagnostic plots:

- Residuals vs fitted plot: The red line is mostly straight and the residuals are mostly evenly distributed about the dotted line. Consequently, I would say the residuals look mostly homoskedastic.
- Normal Q-Q plot: The majority of the points fall on the dotted line so I would say the residuals look mostly normally distributed.
- Scale-location plot: The data points are evenly distributed about the red line and the red line looks somewhat straight. Consequently, I would say that the residuals are homoskedastic.
- Residuals vs Leverage: There are 3 points labeled and one of them is outside the dotted line, which suggests that it might be an outlier that influences the model. Since only one data point is outside the dotted lines and I do not know where it came from, I think it would be safer to keep it in the model.

## Create a summary table

First, create a model summary to get the information of interest. The `summary()` function gives us the coefficients (slope and y-intercept) of the linear model, the standard error of the coefficients, multiple R<sup>2</sup> (quantifies how well the independent variable predicts the dependent variable), and the p-value for the model fit.

```
# store the model summary as an object
model_summary <- summary(modelobject)
model_summary
```

Call:

```
lm(formula = weight ~ length, data = fish_subset)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.0828	-0.4862	-0.1830	0.4128	7.3191

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-11.702476	0.481564	-24.30	<2e-16 ***
length	0.199852	0.005584	35.79	<2e-16 ***

---

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

Residual standard error: 1.057 on 288 degrees of freedom

Multiple R-squared: 0.8164, Adjusted R-squared: 0.8158

F-statistic: 1281 on 1 and 288 DF, p-value: < 2.2e-16

Then, create an ANOVA table and a summary statistics table to report the ANOVA results.

```
# store the ANOVA table as an object
model_squares <- anova(modelobject)
model_squares
```

Analysis of Variance Table

Response: weight

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
length	1	1432.29	1432.29	1280.8	< 2.2e-16 ***
Residuals	288	322.05	1.12		

---

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

```

# make the summary statistics table
model_squares_table <- tidy(model_squares) %>%

# round the sum of squares and mean squares to 2 digits
mutate(across(sumsq:meansq, ~ round(.x, digits = 1))) %>%

# round the F-statistic to 1 digit
mutate(statistic = round(statistic, digits = 1)) %>%

# replace the small p values with < 0.001
mutate(p.value = case_when(p.value < 0.001 ~ "< 0.001")) %>%

# make the data frame a flextable object
flextable() %>%

# change header labels
set_header_labels(df = "Degrees of Freedom",
                  sumsq = "Sum of squares",
                  meansq = "Mean squares",
                  statistic = "F-statistic",
                  p.value = "p-value")

model_squares_table

```

term	Degrees of Freedom	Sum of squares	Mean squares	F-statistic	p-value
length	1	1,432.3	1,432.3	1,280.8	< 0.001
Residuals	288	322.1	1.1		

The ANOVA has information on the degrees of freedom, f-statistic, and p-value of the model like the `summary()` object. However, it also includes information on the sum of squares and mean squares values, which is useful for understanding where the F-statistic, p-value and R<sup>2</sup> values from the `summary()` object come from.

Results:

(F<sub>1,288</sub> = 1280.8, p < 0.001, R<sup>2</sup> = 0.8)