Homework 4

Benise Limon

#call the packages I want to use   
library(tidyverse)  
library(here)  
library(lterdatasampler)  
library(performance)  
library(broom)  
library(ggeffects)  
library(car)  
library(naniar)  
library(flextable)

How does fish length predict fish weight for trout perch (across all sample years)?  
This data set, like most observational environmental or ecological data sets, is noisy (i.e. a lot of spread between points). It is also somewhat messy. You are expected to read the metadata, protocols, and all relevant information on the EDI portal before working with the data.

# Problem 1

1. Write your null and alternative hypotheses in mathematical and biological terms.

Biological:

H0: Fish length does not predict fish weight for trout perch across all sample years.

HA: Fish length does predict fish weight for trout perch across all sample years.

Mathematical:

H0: β1 = 0

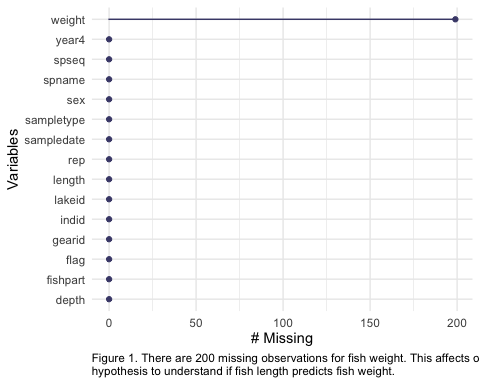
HA: β1 ≠ 0

2. Create a visualization of the missing data for the filtered data set containing the  
observations you will use.

#make an object that reads the CSV file containing the data   
fish <- read.csv(here("data/ntl6\_v12.csv"))

#filter the data to only include trout   
fish\_clean <- fish %>%   
 filter(spname == "TROUTPERCH")

#visualize missing data   
gg\_miss\_var(fish\_clean) +  
 #add a caption of how missing data affects hypothesis   
 labs (caption = str\_wrap("Figure 1. There are 200 missing observations for fish weight. This affects our hypothesis to understand if fish length predicts fish weight.")) +  
 #space out where I want the caption  
 theme(  
 plot.caption = element\_text(hjust = 0)  
 )



a. Write an accompanying caption explaining how/if the missing data is relevant to  
your hypotheses.

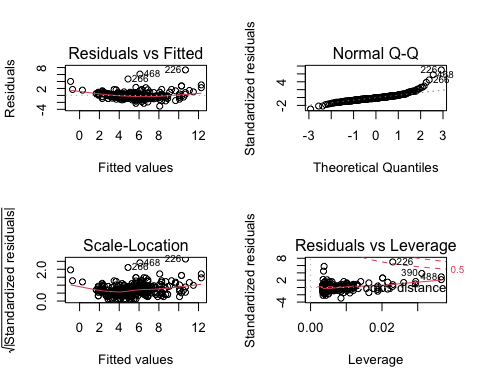
There are 200 observations from the “weight” column are missing. Since it is missing, we will have to delete the null values from the data set, meaning there are less observations to work with. The linear regression is intended to see if fish length can predict fish weight, missing some values of fish weight gives us less data to base our analysis on.

3. Run your test.

#create a model object containing the linear model for if fish length predicts weight using the filtered data frame  
modelobject <- lm(weight ~ length, data = fish\_clean)

4. Visually check the assumptions of your test. Display your diagnostic plots in a grid.

# makes the viewer pane show a 2x2 grid of plots  
# format: par(mfrow = c(number of rows, number of columns))  
par(mfrow = c(2, 2))  
plot(modelobject)



# turns off the 2x2 grid - pop this under the code chunk where you set the 2x2 grid  
dev.off()

null device   
 1

5. For each diagnostic plot, describe in 1-2 sentences what it is showing you, and what you  
decide after looking at the plot.

Residuals vs Fitted: Tells you about homoscedasticity of variance. In the relationship between errors or residuals there is a cluster in the middle around the red line. The red line is pretty straight meaning they are constant throughout the range of x and y values.

Scale-location: Tells you about homoscedasticity of variance. It uses the square root of standardized residuals. The plot shows the data is more clustered in the middle but around the red line. The line is pretty straight across which means that there is good homoscedasticity.

Normal QQ: Shows us if the errors are normally distributed. The data is pretty clustered around the straight normal line meaning that the data is pretty normally distributed.

Residuals vs Leverage: Shows us if the outliers are influencing the model estimates. It measures the influence of a single observation on the model. There are a good amount of data points labeled meaning there are some outliers present in the data. This is also called the Cook’s model.

6. Display the results from summary() using your model object.

# store the model summary as an object  
model\_summary <- summary(modelobject)  
  
model\_summary

Call:  
lm(formula = weight ~ length, data = fish\_clean)  
  
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  
 (199 observations deleted due to missingness)  
Multiple R-squared: 0.8164, Adjusted R-squared: 0.8158   
F-statistic: 1281 on 1 and 288 DF, p-value: < 2.2e-16

7. Create a table that summarizes the ANOVA table. Make sure this table has informative  
column and row names and p-values displayed without scientific notation.

# store the ANOVA table as an object  
# anova(): special function to get analysis of variance tables for a model  
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

# don't name this chunk! some intricacies with Quarto: do not name chunks with tables in them  
  
model\_squares\_table <- tidy(model\_squares) %>%   
 # round the sum of squares and mean squares columns to have 5 digits (could be less)  
 mutate(across(sumsq:meansq, ~ round(.x, digits = 2))) %>%   
 # round the F-statistic to have 1 digit  
 mutate(statistic = round(statistic, digits = 1)) %>%   
 # replace the very very very small p value with < 0.001  
 mutate(p.value = case\_when(  
 p.value < 0.001 ~ "< 0.001"  
 )) %>%   
 # rename the length cell to be meaningful  
 mutate(term = case\_when(  
 term == "length" ~ "Length (mm)",  
 TRUE ~ term  
 )) %>%   
 # make the data frame a flextable object  
 flextable() %>%   
 # change the header labels to be meaningful  
 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 (mm) | 1 | 1,432.29 | 1,432.29 | 1,280.8 | < 0.001 |
| Residuals | 288 | 322.05 | 1.12 |  |  |

8. In 1-2 sentences, describe how the ANOVA table relates to the information you get from  
the summary() object.

The ANOVA table relates to the information from the summary() object because the it is created using the anova function which extracts summary information from the model object, including the degrees of freedom, sum of squares, mean squares, and the p-value. The ANOVA table uses the summary of the model object to understand if there is a significant relationship between the predictor variable and what you are trying to predict.

9. In 2-3 sentences, summarize your results in prose with in-text references to test results. Include all relevant information.

The results from this linear regression were that fish length can be used to predict fish weight. The null hypothesis was rejected because the p-value was <0.001. The high R^2 value explains 81.58% of the variance all fish weight observations can be predicted by the model. The low mean squared value of 1.12 supports the alternative hypothesis because on average the model only differs from the actual values by 1.12g. The scatter plot visually shows the observations are clustered along the regression line, indicating the regression line is a good fit. Analysis of variance, F(1,288) = 1280 p< 0.001, a = 0.05.

10. Create a visualization with model predictions and confidence intervals on top of the  
underlying data. Finalize your plot.

# extract model predictions using ggpredict  
predictions <- ggpredict(modelobject, terms = "length")  
  
predictions

# Predicted values of weight  
  
length | Predicted | 95% CI  
-----------------------------------  
 50 | -1.71 | [-2.12, -1.30]  
 60 | 0.29 | [-0.02, 0.59]  
 65 | 1.29 | [ 1.03, 1.54]  
 75 | 3.29 | [ 3.12, 3.45]  
 85 | 5.28 | [ 5.16, 5.41]  
 95 | 7.28 | [ 7.12, 7.44]  
 105 | 9.28 | [ 9.04, 9.53]  
 120 | 12.28 | [11.88, 12.68]

plot\_predictions <- ggplot(data = fish\_clean,   
 aes(x = length, y = weight)) +  
 # first plot the underlying data from fish\_clean  
 geom\_point() +  
 # then plot the predictions  
 geom\_line(data = predictions,   
 aes(x = x, y = predicted),   
 color = "green", linewidth = 1) +  
 # then plot the 95% confidence interval from ggpredict  
 geom\_ribbon(data = predictions,   
 aes(x = x, y = predicted, ymin = conf.low, ymax = conf.high),   
 alpha = 0.2) +  
 # theme and meaningful labels  
 theme\_bw() +  
 # labeling the x and y axis   
 labs(x = "Trout Perch length (mm)",  
 y = "Trout Perch weight (g)",   
 caption = "Figure 2. An increase in fish length (mm) does predict an increase in fish weight (g)") +  
 #adjusting caption  
 theme(  
 plot.caption = element\_text(hjust = 0))  
  
plot\_predictions

