ENVS 193 - HW 4

Riley Zamora

2023-05-23

Load data using here package

```
library(here)
```

```
## here() starts at C:/Users/riley/Documents/ENVS 193/ENVS-193DS_homework-04_Zamora-Riley
```

```
data_path <- here('ntl6_v12.csv') # returns the path to specified file
data <- read.csv(data_path) # load in data specified by here function
head(data)
```

```
##   lakeid year4 sampledate gearid       spname sampletype depth rep indid length
## 1     AL  1981 1981-08-04 BSEINE BLACKCRAPPIE               -1   4 321M1     42
## 2     AL  1981 1981-08-04 BSEINE BLACKCRAPPIE               -1   4 321M2     41
## 3     AL  1981 1981-08-04 BSEINE BLACKCRAPPIE               -1   4 321M3     46
## 4     AL  1981 1981-08-04 BSEINE BLACKCRAPPIE               -1   4 321M4     26
## 5     AL  1981 1981-08-04 BSEINE BLACKCRAPPIE               -1   4 321M5     21
## 6     AL  1981 1981-08-04 BSEINE BLACKCRAPPIE               -1   4 321R1    240
##   weight sex fishpart        spseq flag
## 1     NA               3211010-100
## 2     NA               3211010-100
## 3     NA               3211010-100
## 4     NA            K  3211010-100
## 5     NA            K  3211010-100
## 6    210            S  3211010-100
```

Problem 1

Question at hand:
How does fish length predict fish weight for trout perch (across all sample years)?

Our model (SLR):
Let Y represent the weight for all sampled trout perch.
Let x_1 represent the length for all sampled trout perch.
Let β_0 be the intercept coefficient.
Let β_1 be the slope coefficient.
Let ϵ be the error term.

$$Y = \beta_0 + \beta_1 x_1 + \epsilon$$

Wrangle data

```
library(tidyverse)
data_wrangle <- data %>%
  select(spname, length, weight, year4) %>%
  filter(spname == 'TROUTPERCH')
```

Part 1.

Null hypothesis: There is no linear relationship between fish length and fish weight for trout perch across all sample years.

$$H_0 : \beta_1 = 0$$

Alternate hypothesis: There is a linear relationship between fish length and fish weight for trout perch across all sample years.

$$H_a : \beta_1 \neq 0$$

Part 2.

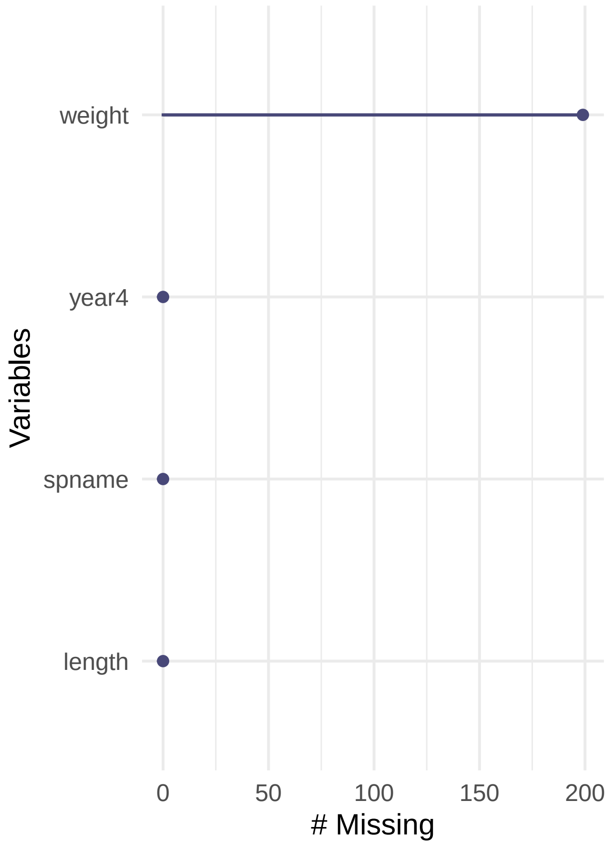
```
library(naniar) # for missing data vis
library(gridExtra) # to help arrange my plots
```

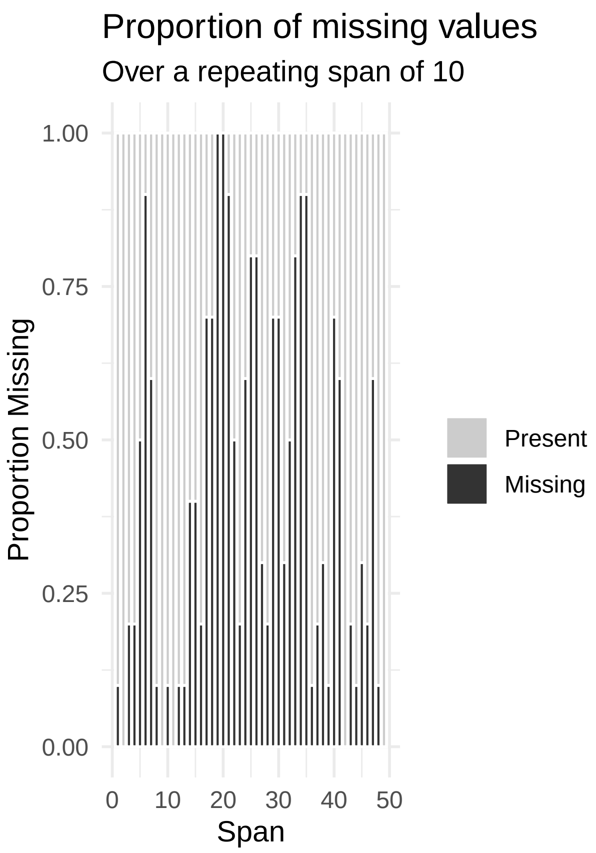
```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
missing_var <- gg_miss_var(data_wrangle)
weight_span <- gg_miss_span(data_wrangle, weight, span_every = 10)

grid.arrange(missing_var, weight_span, layout_matrix = rbind(c(1,2)))
```





Sub-part A:

In these missing value visualizations, we can see exactly what variables contain missing values and the span of how much missing data is in the variables that contain missing data. The left hand plot shows that `weight` is the only variable in my filtered data set. The right hand gives a better idea of the proportion of missing data for a span of 10 data points. This can help us understand if we have bias in the data and also give us an idea of whether some assumptions may be violated. We can also use these visualizations to help us decide how we can handle our missing values.

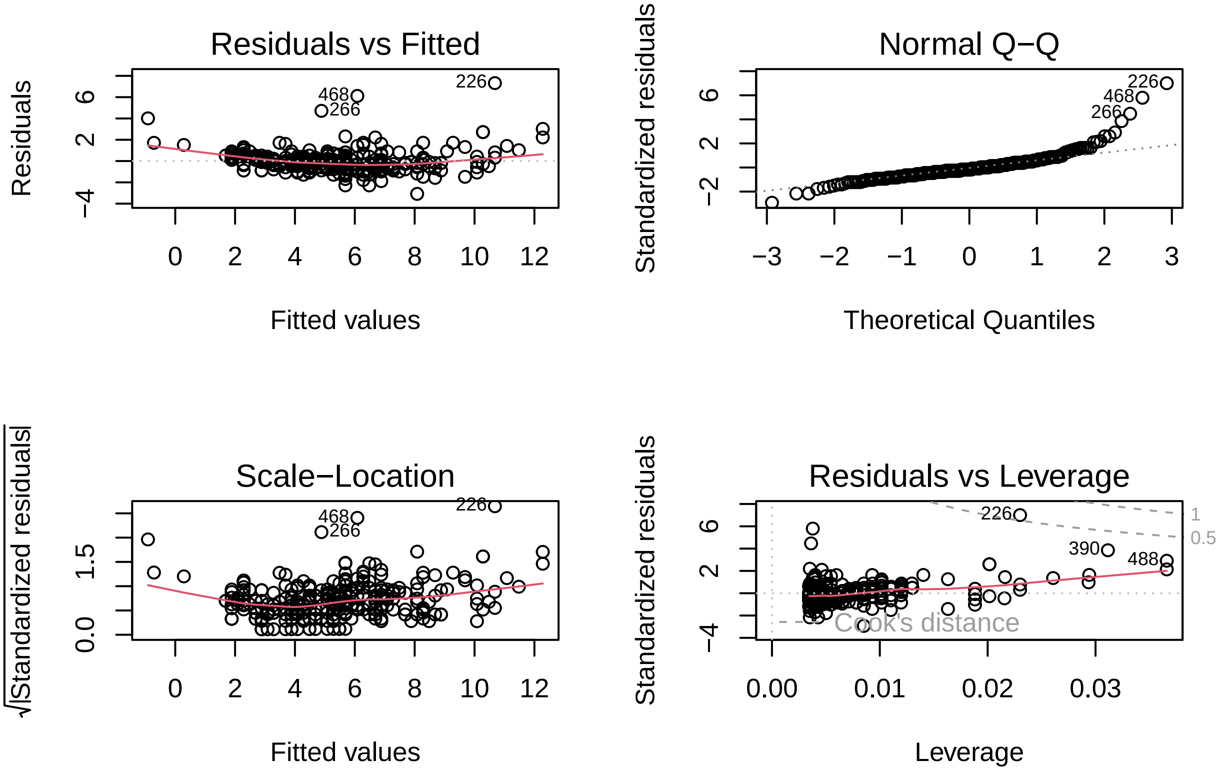
Part 3.

```
fit <- lm(weight~length, data = data_wrangle) # fit a linear model
fit
```

```
##
## Call:
## lm(formula = weight ~ length, data = data_wrangle)
##
## Coefficients:
## (Intercept)       length
##    -11.7025       0.1999
```

Part 4.

```
par(mfrow = c(2,2)) # aligns our resulting plots in a grid
plot(fit) # shows the assumption plots
```



Part 5.

Residuals Vs Fitted: This plot is showing us the *predicted* values vs the residuals. In other words, we are seeing what values our model gives and the residuals which are the difference between the actual values and predicted value. As a rule of thumb, if the smoothed curve is relatively flat then we pass the linearity assumption. I am deciding that we have some outliers but in general we pass linearity assumption.

Normal Q-Q: This plot shows us standardized residuals vs theoretical quantiles of $N(0,1)$. In other words it tells us if Y is normally distributed. If the points falls closely to the line, then we can say the data is normal and in our case it does.

Scale-Location: This plot shows $\sqrt{standardized\ residuals}$ vs Fitted values which rells us if we have constat variance. Since the variability around the red line is even on either side and the red line is relatively flat for the amount of data we have, we pass the homoscedasticity assumption.

Residuals Vs Leverage: This plot shows us our standardized residuals vs the leverage points which helps us see if there are influential points that alter our data. We have some outliers and some and some leverage points that seem to slightly affect our data.

Part 6.

```
summary(fit) # the summary function will show all the estimates and coefficients
```

```
## 
## Call:
## lm(formula = weight ~ length, data = data_wrangle)
## 
## 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
```

Part 7.

```
library(flextable)
```

```
## Warning: package 'flextable' was built under R version 4.2.3
```

```
## 
## Attaching package: 'flextable'
```

```
## The following object is masked from 'package:purrr':
## 
##     compose
```

```
library(broom)

fit_anova <- anova(fit)

# Code from workshop
model_squares_table <- tidy(fit_anova) %>%
  # 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" ~ "Perch Trout 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
```

```
## Warning: fonts used in `flextable` are ignored because the `pdflatex` engine
## is used and not `xelatex` or `lualatex`. You can avoid this warning by using
## the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a compatible
## engine by defining `latex_engine: xelatex` in the YAML header of the R Markdown
## document.
```

term	Degrees of Freedom	Sum of squares	Mean squares	F-statistic	p-value
Perch Trout length (mm)	1	1,432.29	1,432.29	1,280.8	< 0.001
Residuals	288	322.05	1.12		

Part 8.

The summary and anova functions both give us an f-statistic of ≈ 1281 on 288 residual degrees of freedom. The p-value we get across both summary and anova is the same. The summary function gives us more information about t-statistics and the β estimates. The anova is aimed solely at the variance so we are only given the f-stat.

Part 9.

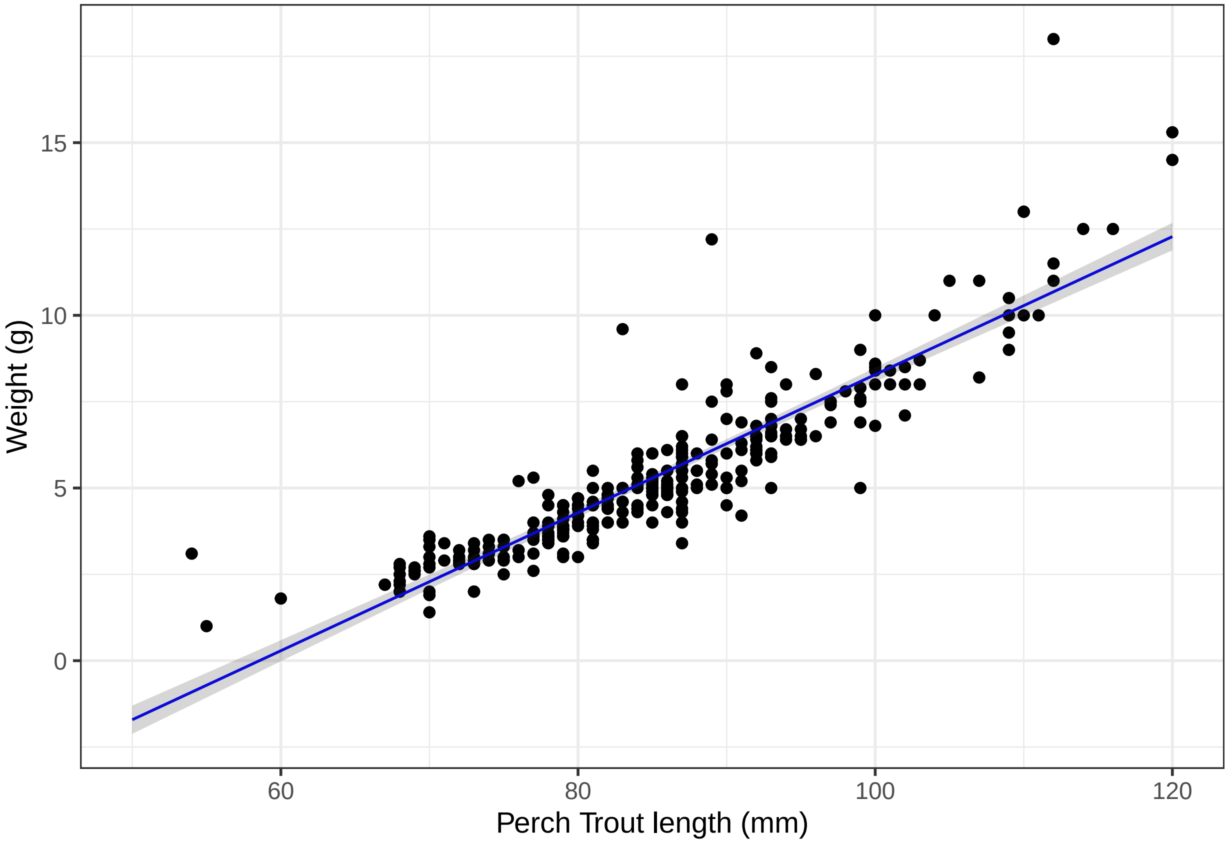
Answering our question of **How does fish length predict fish weight for trout perch (across all sample years)?** we conclude that length is a *statistically significant* predictor of weight for perch trout across all sample years. Looking specifically at our p-value, 2×10^{-16}, which is far below a threshold of 0.001. Looking at the coefficient of determination, multiple R^2, which has a value of 0.8164 we determine that this model is well fit and we should have about 81.64% trust in the findings.

Part 10.

```
library(ggeffects)
predictions <- ggpredict(fit, terms = "length")

plot_predictions <- ggplot(data = data_wrangle,
                           aes(x = length, y = weight)) +
  # first plot the underlying data
  geom_point() +
  # then plot the predictions
  geom_line(data = predictions,
            aes(x = x, y = predicted),
            color = "blue") +
  # 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() +
  labs(x = "Perch Trout length (mm)",
       y = "Weight (g)")

plot_predictions
```



Sub-part A:

This plot shows us the predictions of weight based on the length of perch trout. We can see that there is some uniformity within the predictions and length of perch trout. We have most of the predictions falling closely to the line but outside of the prediction interval.