MLR\_Homework

Arjun Mannem

2023-12-01

### 1. First, add a new code chunk containing the code below which loads the tidyverse, ggthemes, and the flextable R packages.

# Install packages if needed (note: this may take some time to run)  
for(pack in c('tidyverse', 'ggthemes', 'flextable')) {  
if(!(pack %in% installed.packages())) {  
 install.packages(pack)  
}  
}  
  
# Load necessary packages  
library(tidyverse)

## Warning: package 'ggplot2' was built under R version 4.3.2

## Warning: package 'dplyr' was built under R version 4.3.2

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggthemes)  
library(flextable)

## Warning: package 'flextable' was built under R version 4.3.2

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

### 2. In another code chunk, load the data on hawks and set the default ggplot2 and flextable themes using the code below.

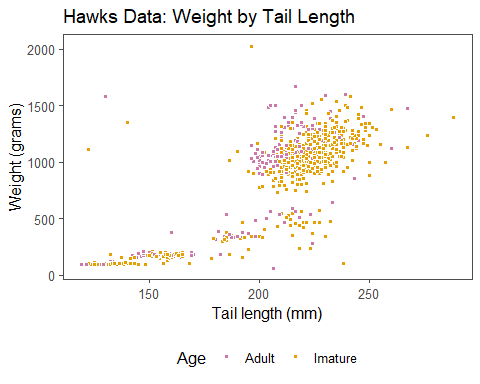
# Set ggplot theme for visualizations  
theme\_set(ggthemes::theme\_few())  
  
# Create a counter for tables  
tbl\_counter <- 1  
  
# Function for making flextable  
make\_flex <- function(myData, ndigits = 2, caption = NULL) {  
   
 if("p.value" %in% colnames(myData)) {  
 myData$p.value <- format.pval(myData$p.value, digits = ndigits)  
 }  
   
 myFlex <- myData %>%   
 flextable::flextable() %>%   
 flextable::autofit() %>%   
 flextable::colformat\_double(digits = ndigits) %>%   
 flextable::fit\_to\_width(7.5)  
   
 if(!is.null(caption)) {  
 myFlex <- myFlex %>%   
 flextable::set\_caption(paste0("Table ", tbl\_counter, ": ", caption))  
 }  
   
 assign("tbl\_counter",   
 value = tbl\_counter + 1, envir = .GlobalEnv)  
   
 return(myFlex)   
}  
  
# Set options for flextables  
set\_flextable\_defaults(na\_str = "NA")  
  
# Vectors of color-blind friendly colors  
speciesColors <- c("#009E73", "#D55E00", "#56B4E9")  
ageColors <- c("#CC79A7", "#E69F00", "#F0E442")  
  
# Importing data on hawks  
hawks <- read\_csv("https://raw.githubusercontent.com/dilernia/STA323/main/Data/hawks.csv") %>%   
 dplyr::select(species, age, wing\_mm:tail\_mm)

## Rows: 908 Columns: 19  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): capture\_time, band\_number, species, age, sex, crop  
## dbl (12): month, day, year, wing\_mm, weight\_g, culmen\_mm, hallux\_mm, tail\_m...  
## time (1): release\_time  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

### 3. Reproduce the scatter plot below showing the weight of the hawks (weight\_g) by the tail lengths (tail\_mm), and color the points based on the age category of each hawk using the code below.

# Creating scatter plot coloring points by sex  
hawks %>%   
 ggplot(aes(x = tail\_mm, y = weight\_g)) +   
 geom\_point(aes(x = tail\_mm, y = weight\_g, fill = age), pch = 21, color = "white") +   
 labs(title = "Hawks Data: Weight by Tail Length",  
 x = "Tail length (mm)",  
 y = "Weight (grams)",  
 fill = "Age") +  
 scale\_fill\_manual(values = ageColors) +  
 theme(legend.position = "bottom")

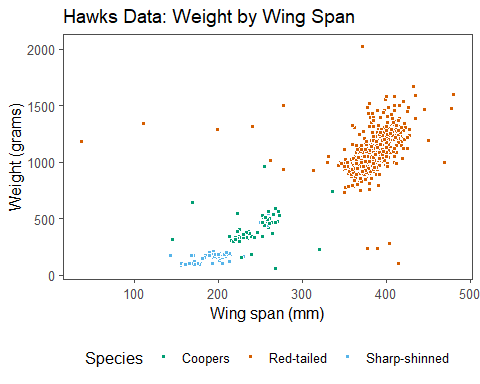
## Warning: Removed 10 rows containing missing values (`geom\_point()`).



### 4. Modifying the code from the previous part, create a scatter plot showing the weight of the hawks (weight\_g) by the wing span (wing\_mm), and color the points based on the species of each hawk. Make sure to update all plot labels accordingly.

# Creating scatter plot coloring points by sex  
hawks %>%   
 ggplot(aes(x = wing\_mm, y = weight\_g)) +   
 geom\_point(aes(x = wing\_mm, y = weight\_g, fill = species), pch = 21, color = "white") +   
 labs(title = "Hawks Data: Weight by Wing Span",  
 x = "Wing span (mm)",  
 y = "Weight (grams)",  
 fill = "Species") +  
 scale\_fill\_manual(values = speciesColors) +  
 theme(legend.position = "bottom")

## Warning: Removed 11 rows containing missing values (`geom\_point()`).



### 5. Using the lm() function in R, fit a multiple linear regression model with species (species) and wing span (wing\_mm) as the predictor variables and weight (weight\_g) as the response variable. Print the model coefficients and table of model summary metrics using the make\_flex() function, reproducing the output below.

# Fitting three-predictor multiple linear regression model  
mlrModel <- lm(weight\_g ~ species + wing\_mm,  
 data = hawks)  
  
# Printing model output  
mlrModel %>%   
 broom::tidy() %>%   
 make\_flex(caption = "Coefficient estimates for model",  
 ndigits = 2)

Table 1: Coefficient estimates for model

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | -139.95 | 45.72 | -3.06 | 0.0023 |
| speciesRed-tailed | 355.22 | 30.65 | 11.59 | < 2e-16 |
| speciesSharp-shinned | -135.72 | 22.70 | -5.98 | 3.2e-09 |
| wing\_mm | 2.29 | 0.17 | 13.31 | < 2e-16 |

mlrModel %>%   
 broom::glance() %>%   
 make\_flex(caption = "Model summary metrics",ndigits=2)

Table 2: Model summary metrics

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.90 | 0.90 | 149.47 | 2,561.61 | <2e-16 | 3.00 | -5,762.13 | 11,534.26 | 11,558.26 | 19,949,999.65 | 893 | 897 |

### 6. Provide a statement of the estimated regression equation using species (species) and wing span (wing\_mm) as the predictor variables and weight (weight\_g) as the response variable.

The estimated MLR model is

ŷ = -139.95 + 355.22 \* x1 - 135.72 \* x2 + 2.29 \* x3,

where:

ŷ represents the estimated weight of a species in grams

βHat0 is considered as Coopers hawks

x1 represents the Red-tailed hawks

x2 represents the Sharp-shinned hawks

x3 represents the wing span in mm

### 7. Provide the value of and interpret the estimated slope for wing span in context.

Since βHat3 = 2.29, for every 1 mm increase in wing span, we expect the weight of species to increase by 2.29 grams on average, holding the species constant.

### 8. Provide the value of and interpret the estimated slope for the indicator variable for red-tailed hawks in context.

Since βHat1 = 355.22, the expected weight in grams of a Red-tailed hawk species is 355.22 grams greater than that of Coopers hawk species on average, holding the wing span in mm constant.

### 9. Provide and interpret the value of r2 in this context.

r^2 = 0.90

*Interpretation*: About 90% of the variability in the estimated weight of a species in grams is explained by the species and wing span in mm.

### 10. Obtain a predicted value and 90% prediction interval for a red-tailed hawk with a wing span of 400mm using the predict() function.

# Obtaining predicted value  
predict(mlrModel, newdata = tibble(species = "Red-tailed",wing\_mm = 400),interval = "prediction",level=0.90)

## fit lwr upr  
## 1 1132.058 885.6923 1378.424

*Predicted Value*: 1132.058 grams

*90% prediction interval* (in grams): (885.6923, 1378.424)

### 11. Obtain a predicted value and 95% prediction interval for a Cooper’s hawk with a wing span of 250mm using the predict() function.

# Obtaining predicted value  
predict(mlrModel, newdata = tibble(species = "Coopers",wing\_mm = 250),interval = "prediction",level=0.95)

## fit lwr upr  
## 1 433.0427 137.5703 728.5152

*Predicted Value*: 433.0427 grams

*95% prediction interval* (in grams): (137.5703, 728.5152)