Homework 2

Matt Viana

Table of contents

[Link to the Github repository](https://github.com/STAT380/hw2.git)

|  |
| --- |
| Due: Feb 9, 2024 @ 11:59pm |
| Please read the instructions carefully before submitting your assignment.   1. This assignment requires you to only upload a PDF file on Canvas 2. Don’t collapse any code cells before submitting. 3. Remember to make sure all your code output is rendered properly before uploading your submission.   ⚠️ Please add your name to the author information in the frontmatter before submitting your assignment ⚠️ |

For this assignment, we will be using the [Abalone dataset](http://archive.ics.uci.edu/ml/datasets/Abalone) from the UCI Machine Learning Repository. The dataset consists of physical measurements of abalone (a type of marine snail) and includes information on the age, sex, and size of the abalone.

We will be using the following libraries:

library(readr)  
library(tidyr)  
library(ggplot2)  
library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

library(purrr)  
library(cowplot)

## Question 1

|  |
| --- |
| 30 points |
| EDA using readr, tidyr and ggplot2 |

###### 1.1 (5 points)

Load the “Abalone” dataset as a tibble called abalone using the URL provided below. The abalone\_col\_names variable contains a vector of the column names for this dataset (to be consistent with the R naming pattern). Make sure you read the dataset with the provided column names.

library(readr)  
url <- "http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"  
  
abalone\_col\_names <- c(  
 "sex",   
 "length",   
 "diameter",   
 "height",   
 "whole\_weight",   
 "shucked\_weight",   
 "viscera\_weight",   
 "shell\_weight",   
 "rings"  
)  
  
abalone <- read\_csv(url, col\_names = abalone\_col\_names, show\_col\_types = FALSE)

###### 1.2 (5 points)

Remove missing values and NAs from the dataset and store the cleaned data in a tibble called df. How many rows were dropped?

df <- na.omit(abalone)

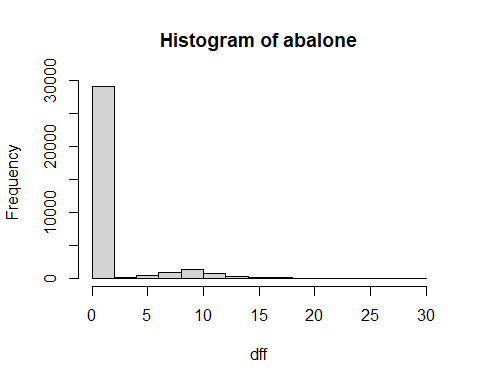
##### 1.3 (5 points)

Plot histograms of all the quantitative variables in a **single plot** [[1]](#footnote-31)

dff <- as.double(unlist(df))

Warning: NAs introduced by coercion

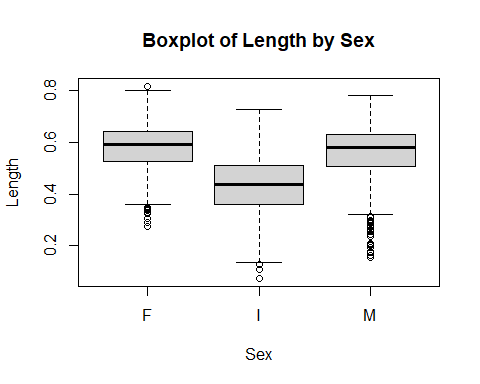
hist(dff, main = "Histogram of abalone")



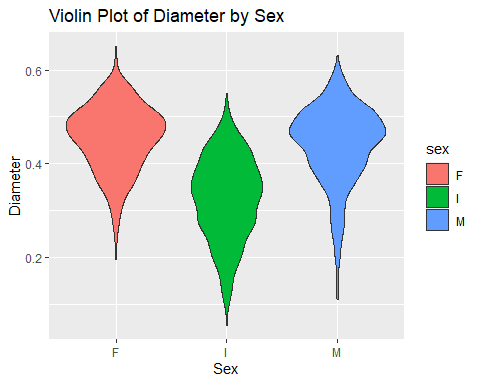
##### 1.4 (5 points)

Create a boxplot of length for each sex and create a violin-plot of of diameter for each sex. Are there any notable differences in the physical appearences of abalones based on your analysis here?

boxplot(length ~ sex, data = abalone,   
 main = "Boxplot of Length by Sex",  
 xlab = "Sex", ylab = "Length")



ggplot(abalone, aes(x = sex, y = diameter, fill = sex)) +  
 geom\_violin() +  
 labs(title = "Violin Plot of Diameter by Sex", x = "Sex", y = "Diameter")

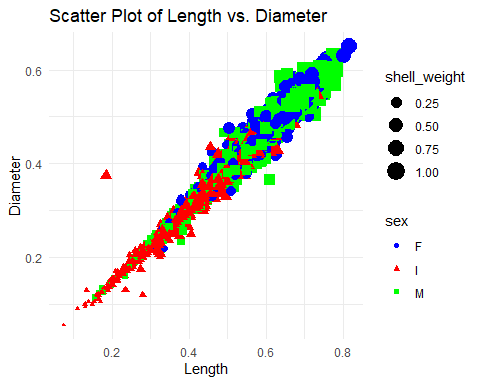


The Violin plot is more organic. It doesn’t have steps like the boxplot does. You can look at each area instead of the whole..

###### 1.5 (5 points)

Create a scatter plot of length and diameter, and modify the shape and color of the points based on the sex variable. Change the size of each point based on the shell\_wight value for each observation. Are there any notable anomalies in the dataset?

ggplot(abalone, aes(x = length, y = diameter, color = sex, shape = sex, size = shell\_weight)) +  
 geom\_point() +  
 labs(title = "Scatter Plot of Length vs. Diameter",  
 x = "Length", y = "Diameter") +  
 scale\_color\_manual(values = c("blue", "red", "green")) +  
 scale\_shape\_manual(values = c(19, 17, 15)) +  
 theme\_minimal()

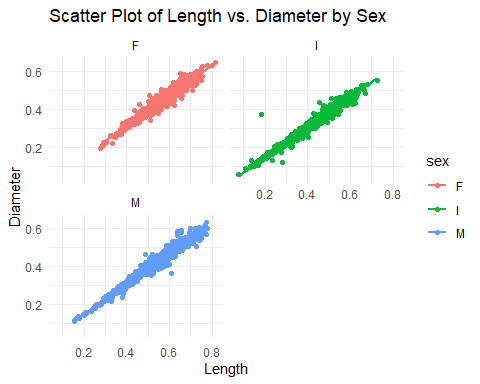


###### 1.6 (5 points)

For each sex, create separate scatter plots of length and diameter. For each plot, also add a **linear** trendline to illustrate the relationship between the variables. Use the facet\_wrap() function in R for this, and ensure that the plots are vertically stacked **not** horizontally. You should end up with a plot that looks like this: [[2]](#footnote-46)

ggplot(abalone, aes(x = length, y = diameter, color = sex)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE) +  
 labs(title = "Scatter Plot of Length vs. Diameter by Sex",  
 x = "Length", y = "Diameter") +  
 facet\_wrap(~ sex, nrow = 2) +  
 theme\_minimal()

`geom\_smooth()` using formula = 'y ~ x'



—

## Question 2

|  |
| --- |
| 40 points |
| More advanced analyses using dplyr, purrrr and ggplot2 |

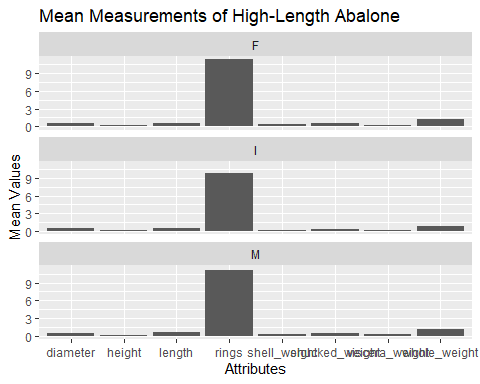
###### 2.1 (10 points)

Filter the data to only include abalone with a length of at least meters. Group the data by sex and calculate the mean of each variable for each group. Create a bar plot to visualize the mean values for each variable by sex.

high\_length\_abalone <- abalone %>%  
 group\_by(sex) %>%  
 filter(length >= 0.5) %>%  
 summarise(across(  
 c(length, diameter, height, whole\_weight, shucked\_weight, viscera\_weight, shell\_weight, rings),  
 mean,  
 na.rm = TRUE  
 ))

Warning: There was 1 warning in `summarise()`.  
ℹ In argument: `across(...)`.  
ℹ In group 1: `sex = "F"`.  
Caused by warning:  
! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.  
Supply arguments directly to `.fns` through an anonymous function instead.  
  
 # Previously  
 across(a:b, mean, na.rm = TRUE)  
  
 # Now  
 across(a:b, \(x) mean(x, na.rm = TRUE))

high\_length\_longer <- high\_length\_abalone %>%  
 pivot\_longer(  
 cols = !sex,  
 names\_to = "Attributes",  
 values\_to = "Values"  
 )  
  
ggplot(data = high\_length\_longer, aes(x = Attributes, y = Values)) +  
 geom\_col() +   
 facet\_wrap(~sex, ncol = 1) +   
 labs(  
 title = "Mean Measurements of High-Length Abalone",  
 x = "Attributes",  
 y = "Mean Values"  
 )



###### 2.2 (15 points)

Implement the following in a **single command**:

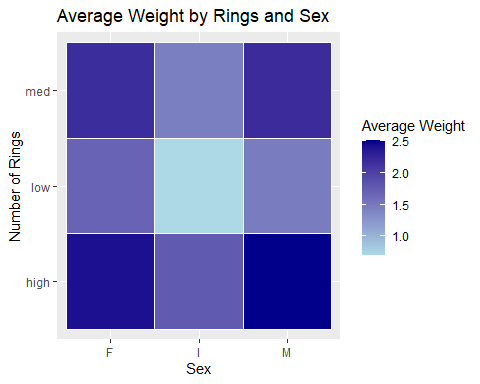
1. Temporarily create a new variable called num\_rings which takes a value of:

* "low" if rings < 10
* "high" if rings > 20, and
* "med" otherwise

1. Group df by this new variable and sex and compute avg\_weight as the average of the whole\_weight + shucked\_weight + viscera\_weight + shell\_weight for each combination of num\_rings and sex.
2. Use the geom\_tile() function to create a tile plot of num\_rings vs sex with the color indicating of each tile indicating the avg\_weight value.

abalone %>%  
 mutate(num\_rings = case\_when(  
 rings < 10 ~ "low",  
 rings > 20 ~ "high",  
 TRUE ~ "med"  
 )) %>%  
 group\_by(num\_rings, sex) %>%  
 summarise(avg\_weight = mean(whole\_weight + shucked\_weight + viscera\_weight + shell\_weight, na.rm = TRUE)) %>%  
 ggplot(aes(x = sex, y = num\_rings, fill = avg\_weight)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient(low = "lightblue", high = "darkblue") +  
 labs(  
 title = "Average Weight by Rings and Sex",  
 x = "Sex",  
 y = "Number of Rings",  
 fill = "Average Weight"  
 )

`summarise()` has grouped output by 'num\_rings'. You can override using the  
`.groups` argument.



###### 2.3 (5 points)

Make a table of the pairwise correlations between all the numeric variables rounded to 2 decimal points. Your final answer should look like this [[3]](#footnote-63)

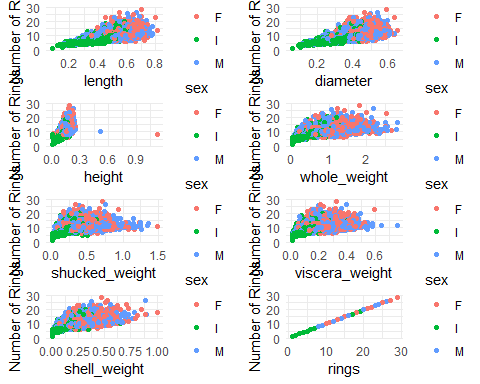
numeric\_vars <- select\_if(df, is.numeric)  
  
correlation\_matrix <- round(cor(numeric\_vars), 2)  
  
print(correlation\_matrix)

length diameter height whole\_weight shucked\_weight  
length 1.00 0.99 0.83 0.93 0.90  
diameter 0.99 1.00 0.83 0.93 0.89  
height 0.83 0.83 1.00 0.82 0.77  
whole\_weight 0.93 0.93 0.82 1.00 0.97  
shucked\_weight 0.90 0.89 0.77 0.97 1.00  
viscera\_weight 0.90 0.90 0.80 0.97 0.93  
shell\_weight 0.90 0.91 0.82 0.96 0.88  
rings 0.56 0.57 0.56 0.54 0.42  
 viscera\_weight shell\_weight rings  
length 0.90 0.90 0.56  
diameter 0.90 0.91 0.57  
height 0.80 0.82 0.56  
whole\_weight 0.97 0.96 0.54  
shucked\_weight 0.93 0.88 0.42  
viscera\_weight 1.00 0.91 0.50  
shell\_weight 0.91 1.00 0.63  
rings 0.50 0.63 1.00

###### 2.4 (10 points)

Use the map2() function from the purrr package to create a scatter plot for each *quantitative* variable against the number of rings variable. Color the points based on the sex of each abalone. You can use the cowplot::plot\_grid() function to finally make the following grid of plots.

quantitative\_vars <- select(abalone, -sex) %>%  
 keep(is.numeric)  
  
scatter\_plots <- map2(quantitative\_vars, names(quantitative\_vars), ~  
 ggplot(abalone, aes(x = .data[[.y]], y = rings, color = sex)) +  
 geom\_point() +  
 labs(x = .y, y = "Number of Rings") +  
 theme\_minimal())  
  
plot\_grid(plotlist = scatter\_plots, ncol = 2)



—

## Question 3

|  |
| --- |
| 30 points |
| Linear regression using lm |

###### 3.1 (10 points)

Perform a simple linear regression with diameter as the covariate and height as the response. Interpret the model coefficients and their significance values.

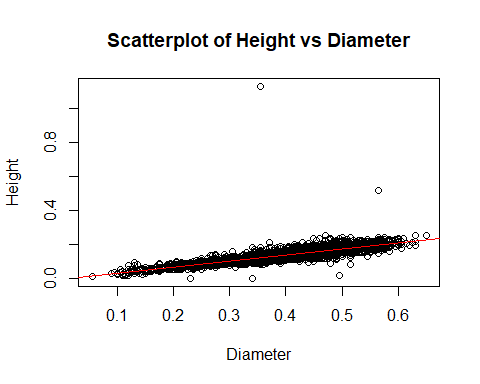
model <- lm(height ~ diameter, data = abalone)  
  
summary(model)

Call:  
lm(formula = height ~ diameter, data = abalone)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.15513 -0.01053 -0.00147 0.00852 1.00906   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -0.003803 0.001512 -2.515 0.0119 \*   
diameter 0.351376 0.003602 97.544 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.0231 on 4175 degrees of freedom  
Multiple R-squared: 0.695, Adjusted R-squared: 0.695   
F-statistic: 9515 on 1 and 4175 DF, p-value: < 2.2e-16

###### 3.2 (10 points)

Make a scatterplot of height vs diameter and plot the regression line in color="red". You can use the base plot() function in R for this. Is the linear model an appropriate fit for this relationship? Explain.

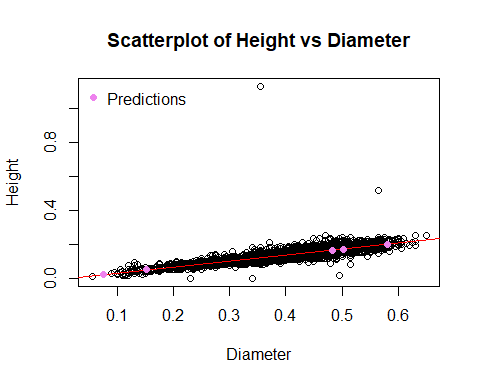
plot(abalone$diameter, df$height, xlab = "Diameter", ylab = "Height", main = "Scatterplot of Height vs Diameter")  
  
model <- lm(height ~ diameter, data = abalone)  
  
abline(model, col = "red")



###### 3.3 (10 points)

Suppose we have collected observations for “new” abalones with new\_diameter values given below. What is the expected value of their height based on your model above? Plot these new observations along with your predictions in your plot from earlier using color="violet"

new\_diameters <- c(  
 0.15218946,  
 0.48361548,  
 0.58095513,  
 0.07603687,  
 0.50234599,  
 0.83462092,  
 0.95681938,  
 0.92906875,  
 0.94245437,  
 0.01209518  
 )  
   
new\_data <- data.frame(diameter = new\_diameters)  
  
new\_data$predicted\_height <- predict(model, newdata = new\_data)  
  
plot(abalone$diameter, abalone$height, xlab = "Diameter", ylab = "Height", main = "Scatterplot of Height vs Diameter", col = "black")  
abline(model, col = "red")  
  
points(new\_data$diameter, new\_data$predicted\_height, col = "violet", pch = 19)  
legend("topleft", legend = "Predictions", pch = 19, col = "violet", bty = "n")



—

# Appendix

|  |
| --- |
| Session Information |
| Print your R session information using the following command  sessionInfo()  R version 4.3.2 (2023-10-31 ucrt) Platform: x86\_64-w64-mingw32/x64 (64-bit) Running under: Windows 11 x64 (build 22621)  Matrix products: default   locale: [1] LC\_COLLATE=English\_United States.utf8  [2] LC\_CTYPE=English\_United States.utf8  [3] LC\_MONETARY=English\_United States.utf8 [4] LC\_NUMERIC=C  [5] LC\_TIME=English\_United States.utf8   time zone: America/New\_York tzcode source: internal  attached base packages: [1] stats graphics grDevices datasets utils methods base   other attached packages: [1] cowplot\_1.1.3 purrr\_1.0.2 dplyr\_1.1.4 ggplot2\_3.4.4 tidyr\_1.3.1  [6] readr\_2.1.5   loaded via a namespace (and not attached):  [1] Matrix\_1.6-1.1 bit\_4.0.5 gtable\_0.3.4 jsonlite\_1.8.8   [5] crayon\_1.5.2 compiler\_4.3.2 renv\_1.0.3 tidyselect\_1.2.0  [9] parallel\_4.3.2 splines\_4.3.2 scales\_1.3.0 yaml\_2.3.8  [13] fastmap\_1.1.1 lattice\_0.21-9 R6\_2.5.1 labeling\_0.4.3  [17] generics\_0.1.3 curl\_5.2.0 knitr\_1.45 tibble\_3.2.1  [21] munsell\_0.5.0 pillar\_1.9.0 tzdb\_0.4.0 rlang\_1.1.3  [25] utf8\_1.2.4 xfun\_0.42 bit64\_4.0.5 cli\_3.6.2  [29] mgcv\_1.9-0 withr\_3.0.0 magrittr\_2.0.3 digest\_0.6.34  [33] grid\_4.3.2 vroom\_1.6.5 hms\_1.1.3 nlme\_3.1-163  [37] lifecycle\_1.0.4 vctrs\_0.6.5 evaluate\_0.23 glue\_1.7.0  [41] farver\_2.1.1 fansi\_1.0.6 colorspace\_2.1-0 rmarkdown\_2.25  [45] tools\_4.3.2 pkgconfig\_2.0.3 htmltools\_0.5.7 |

::: {.content-visible when-format=“html”} | length | diameter | height | whole\_weight | shucked\_weight | viscera\_weight | shell\_weight | rings | | |:——-|——-:|——-:|——-:|——-:|——-:|——-:|——-:|——-:| | length | 1.00 | 0.99 | 0.83 | 0.93 | 0.90 | 0.90 | 0.90 | 0.56 | | diameter | 0.99 | 1.00 | 0.83 | 0.93 | 0.89 | 0.90 | 0.91 | 0.57 | | height | 0.83 | 0.83 | 1.00 | 0.82 | 0.77 | 0.80 | 0.82 | 0.56 | | whole\_weight | 0.93 | 0.93 | 0.82 | 1.00 | 0.97 | 0.97 | 0.96 | 0.54 | | shucked\_weight | 0.90 | 0.89 | 0.77 | 0.97 | 1.00 | 0.93 | 0.88 | 0.42 | | viscera\_weight | 0.90 | 0.90 | 0.80 | 0.97 | 0.93 | 1.00 | 0.91 | 0.50 | | shell\_weight | 0.90 | 0.91 | 0.82 | 0.96 | 0.88 | 0.91 | 1.00 | 0.63 | | rings | 0.56 | 0.57 | 0.56 | 0.54 | 0.42 | 0.50 | 0.63 | 1.00 |

1. You can use the facet\_wrap() function for this. Have a look at its documentation using the help console in R [↑](#footnote-ref-31)
2. Plot example for 1.6 [↑](#footnote-ref-46)
3. Table for 2.3 [↑](#footnote-ref-63)