

# Homework 2

Matt Viana

## Table of contents

Question 1 .....	2
Question 2 .....	6
Question 3 .....	11
Appendix .....	15

[Link to the Github repository](#)

⚠ 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](#) 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)
```

---

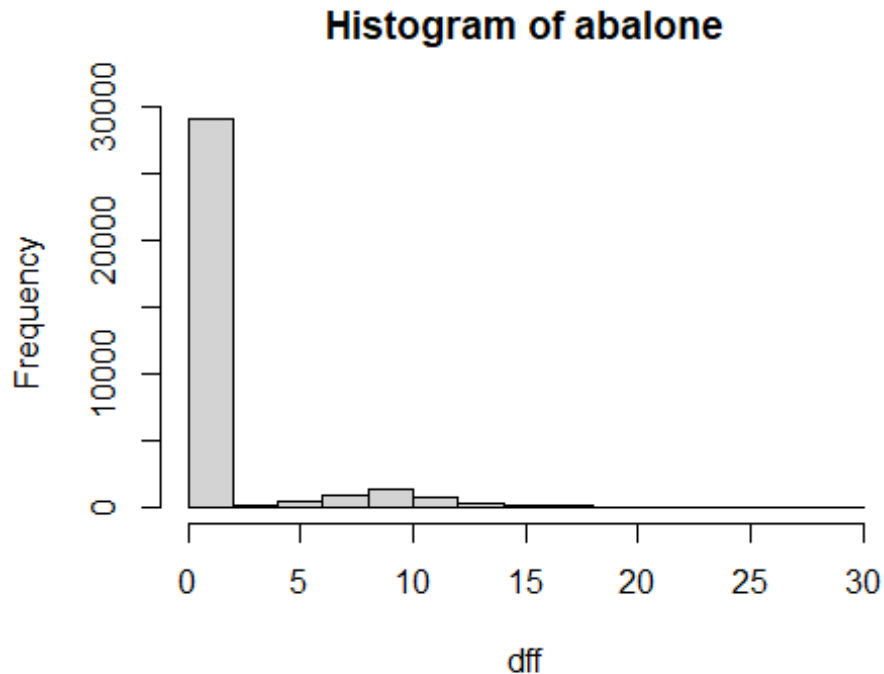
### 1.3 (5 points)

Plot histograms of all the quantitative variables in a **single plot** <sup>1</sup>

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

Warning: NAs introduced by coercion

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



---

### 1.4 (5 points)

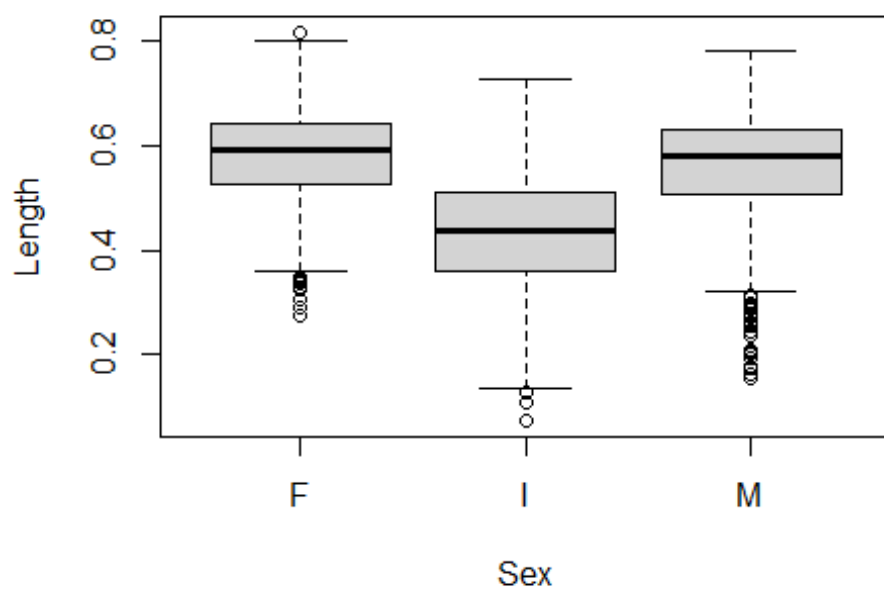
Create a boxplot of length for each sex and create a violin-plot of of diameter for each sex. Are there any notable differences in the physical appearances of abalones based on your analysis here?

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

---

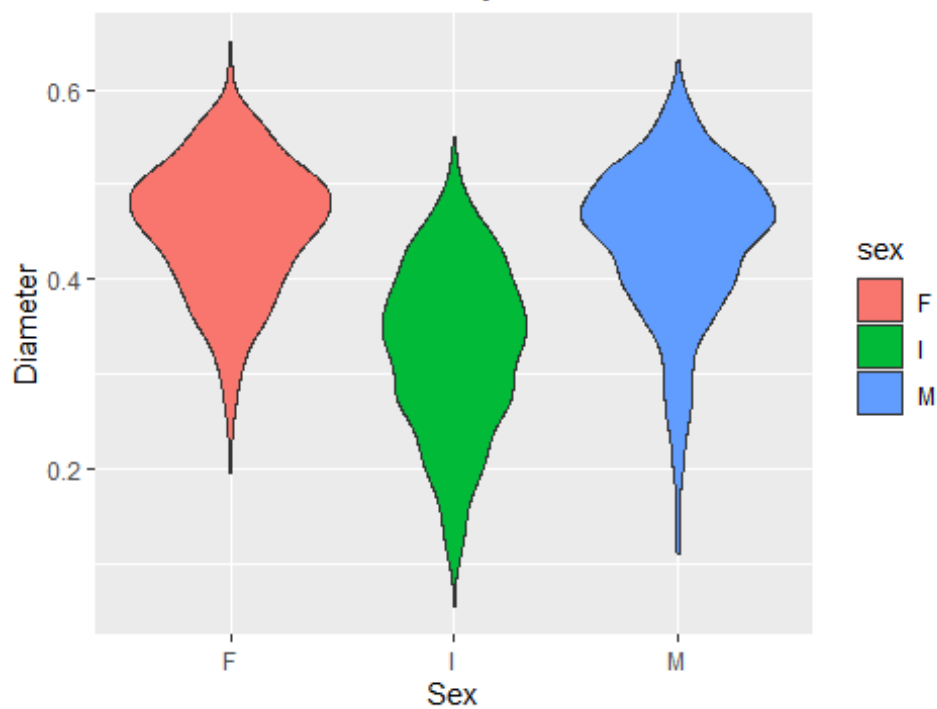
<sup>1</sup> You can use the `facet_wrap()` function for this. Have a look at its documentation using the help console in R

**Boxplot of Length by Sex**



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

**Violin Plot of Diameter by Sex**



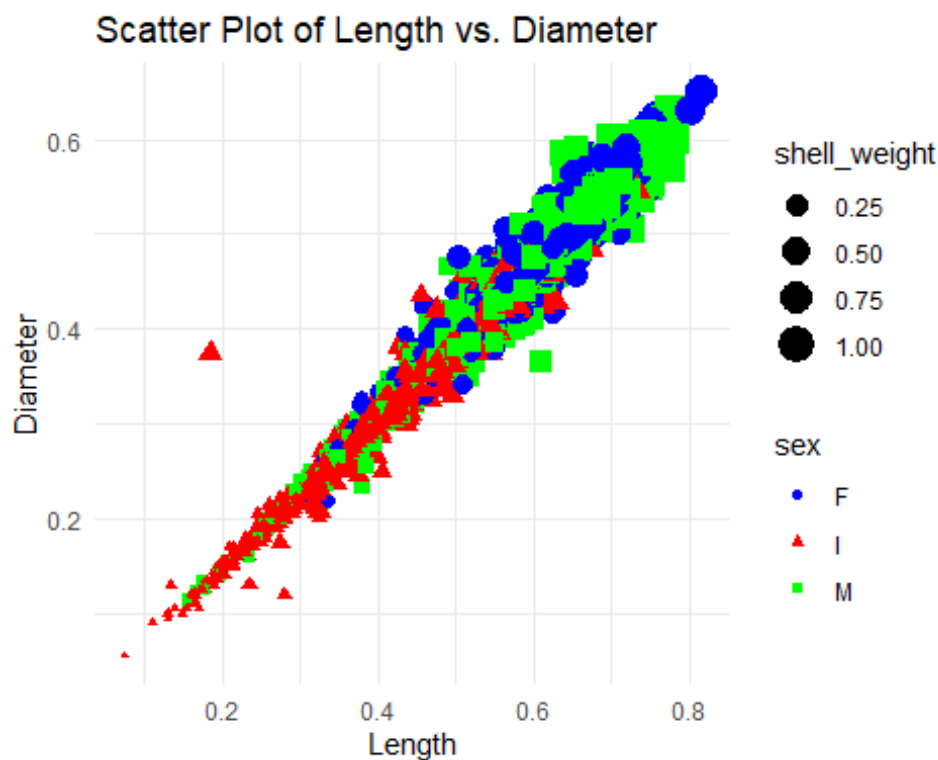
The Violin plot is more organic. It doesn't have steps like the boxplot does. You can look at each area instead of the whole..

---

### 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\_weight 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()
```



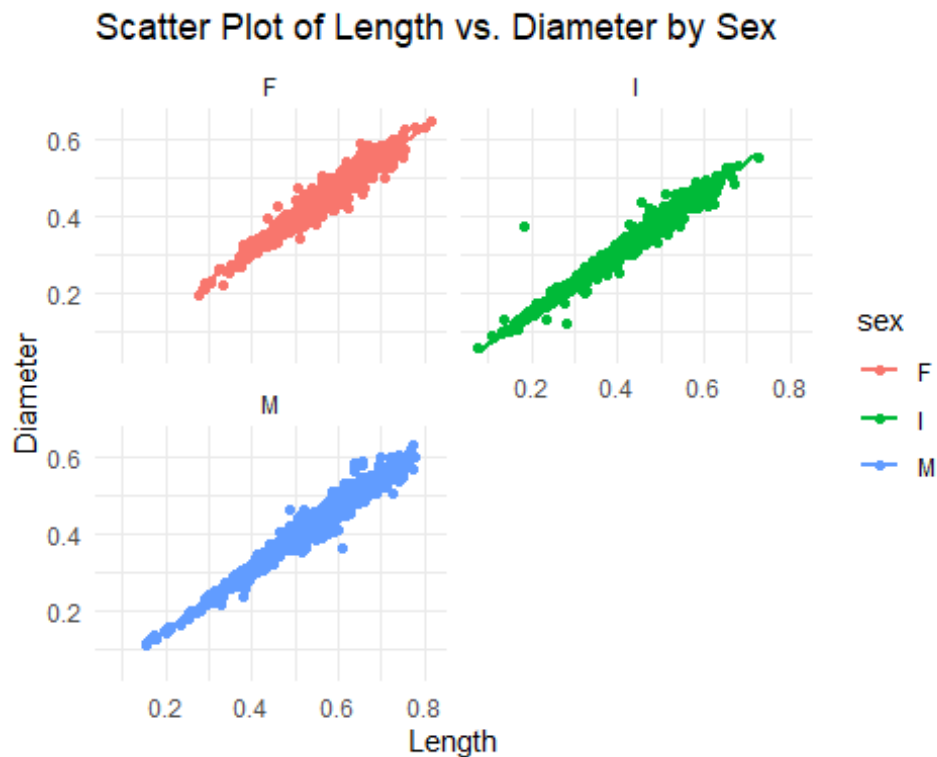
### 1.6 (5 points)

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: <sup>2</sup>

```
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`, `purrr` and `ggplot2`

<sup>2</sup> Plot example for 1.6

## 2.1 (10 points)

Filter the data to only include abalone with a length of at least 0.5 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()`.

! In argument: `across(...)`.

! In group 1: `sex = "F"`.

Caused by warning:

! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.

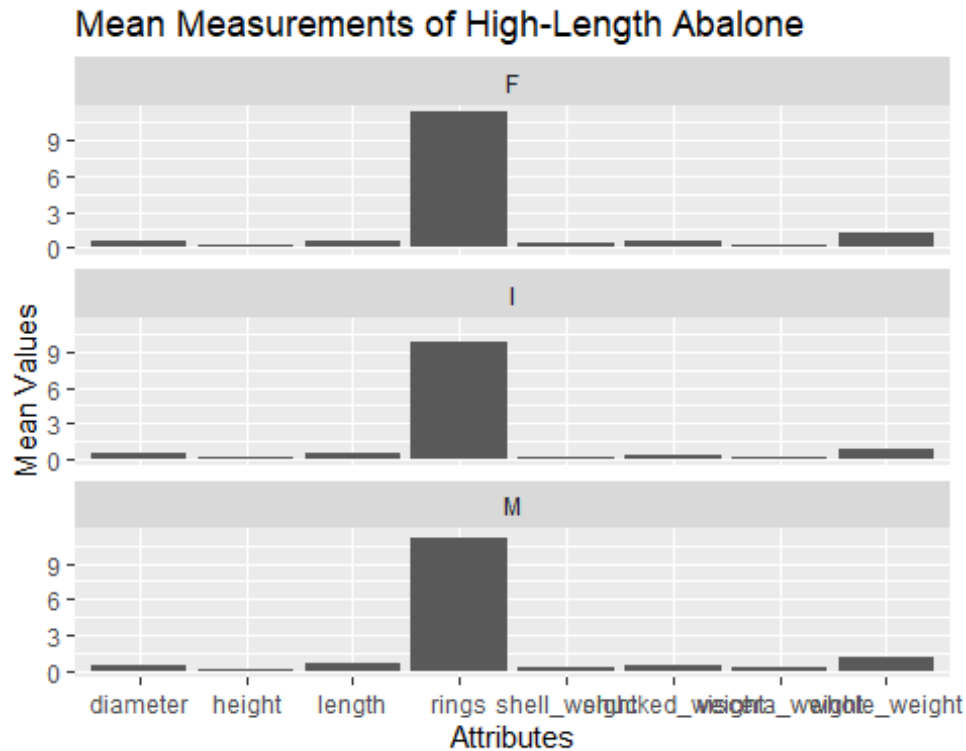
Supply arguments directly to `fns` through an anonymous function instead.

```
# 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**:

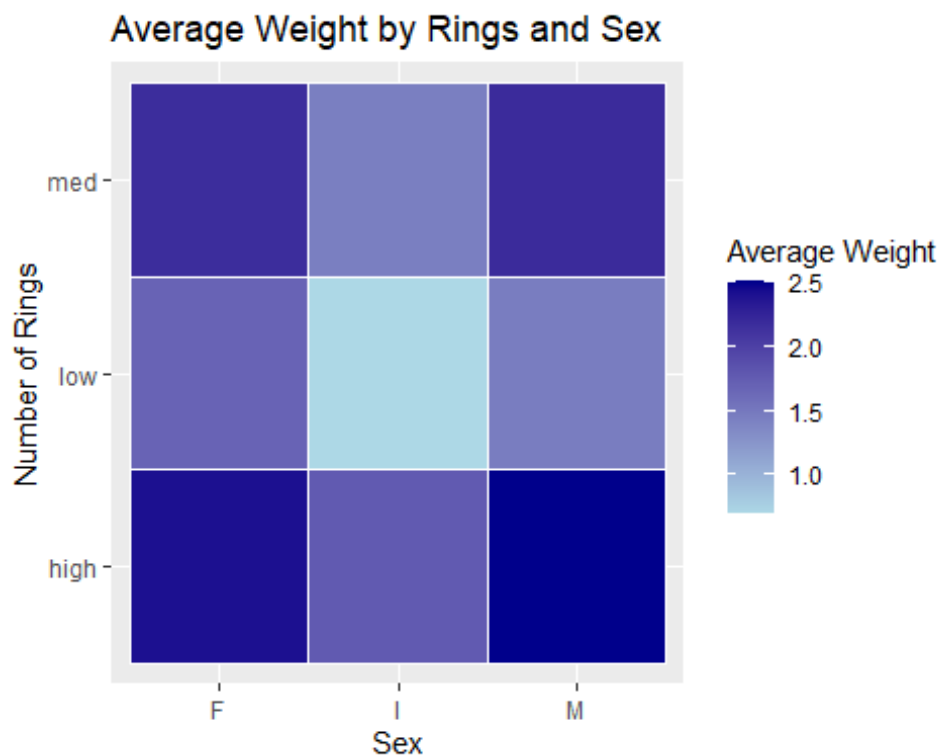
- Temporarily create a new variable called `num_rings` which takes a value of:
  - "low" if `rings < 10`
  - "high" if `rings > 20`, and
  - "med" otherwise
- Group `df` by this new variable and `sex` and compute `avg_weight` as the average of the `whole_weight` + `shucked_weight` + `viscera_weight` + `shell_weight` for each combination of `num_rings` and `sex`.
- 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 <sup>3</sup>

```
numeric_vars <- select_if(df, is.numeric)
correlation_matrix <- round(cor(numeric_vars), 2)
```

<sup>3</sup> Table for 2.3

```
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

---

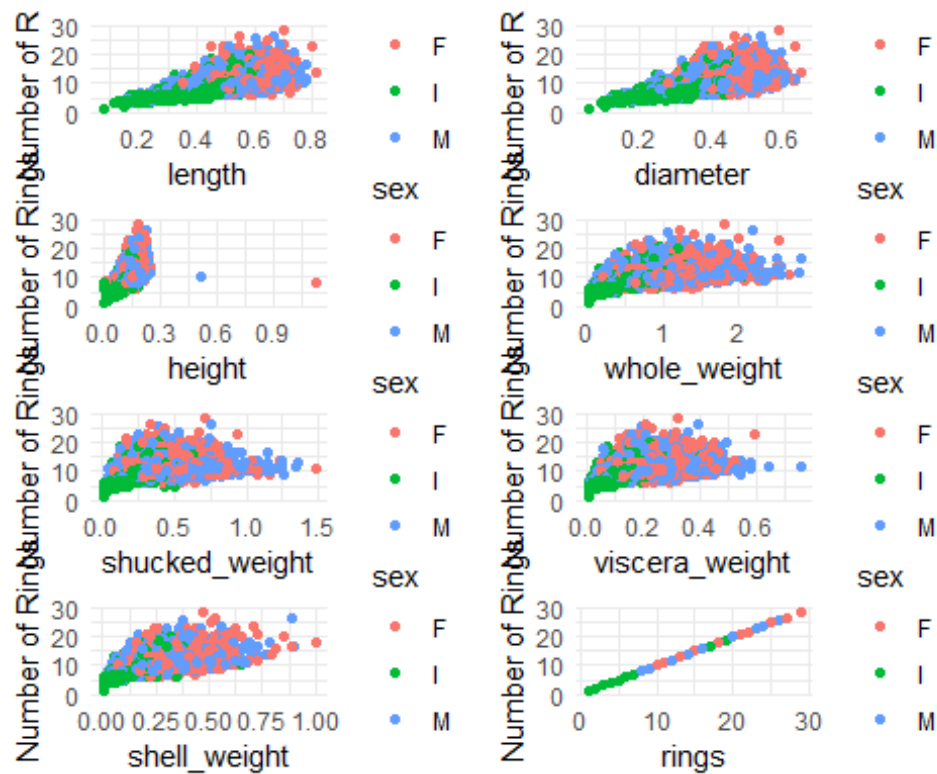
## 2.4 (10 points)

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
```

---

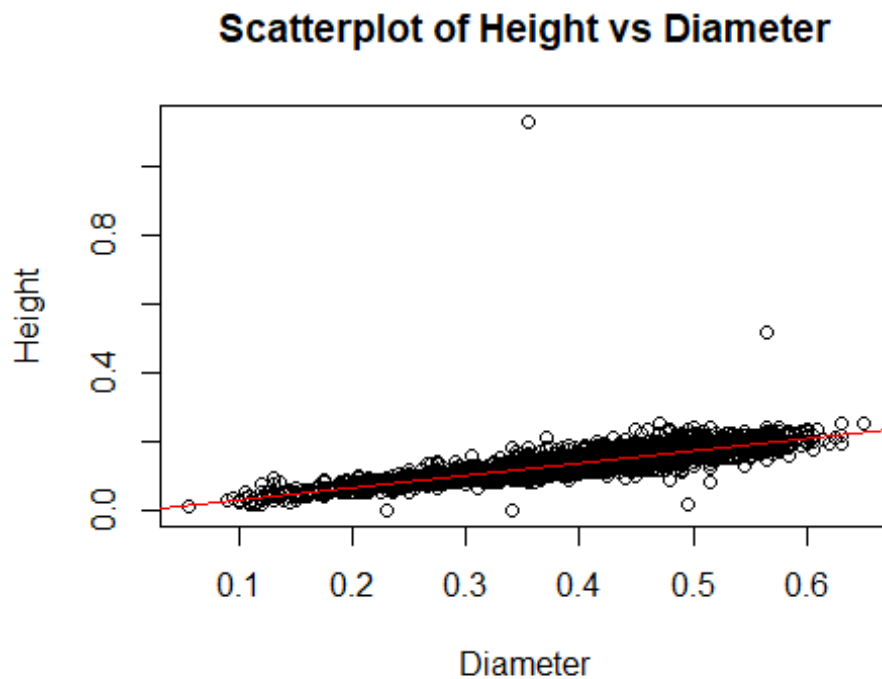
### 3.2 (10 points)

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
)

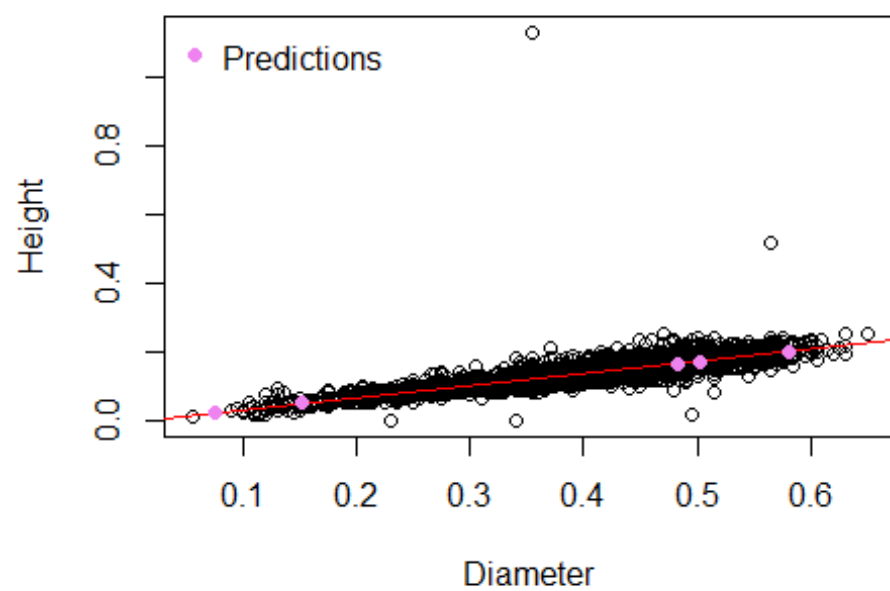
new_data <- data.frame(diameter = new_diameters)

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")
```

**Scatterplot of Height vs Diameter**



## 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 |