Homework 2

Madeleine Willson

Table of contents

❗ 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.

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)
```

```
Rows: 4177 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): sex
dbl (8): length, diameter, height, whole_weight, shucked_weight, viscera_wei...

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.
```

```
  #View(abalone)
```

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 <- abalone %>% drop_na()
  nrow(abalone) - nrow(df)
```

```
[1] 0
```

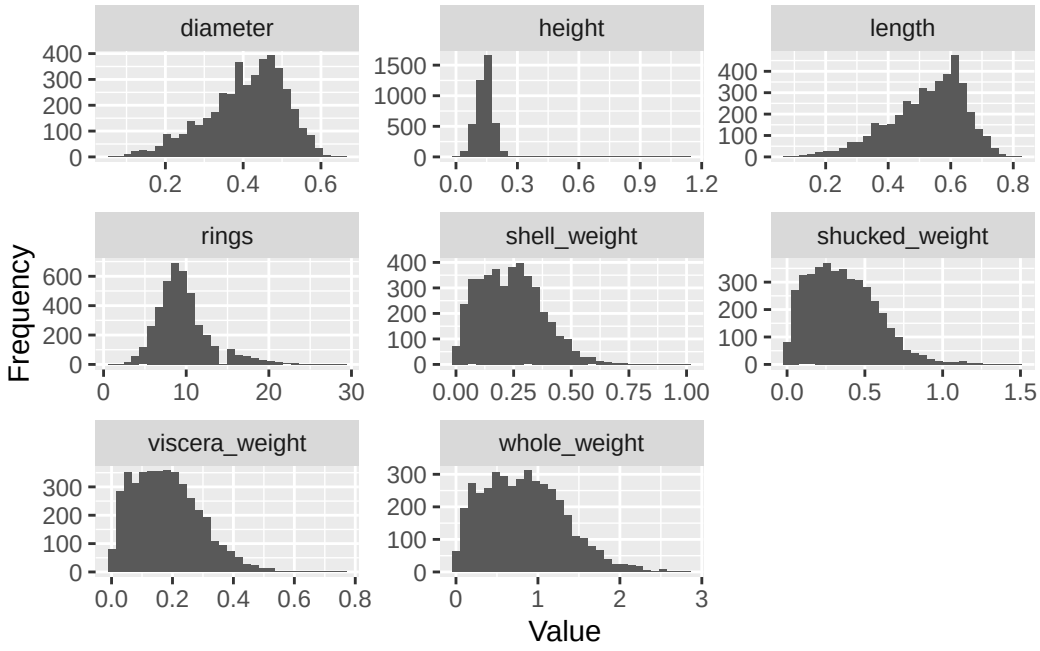
None were dropped.

1.3 (5 points)

Plot histograms of all the quantitative variables in a **single plot** [^1] [^1]: You can use the facet_wrap() function for this. Have a look at its documentation using the help console in R

```r
# Insert your code here
df_long <- pivot_longer(df, -sex, names_to = "variable", values_to = "value")

ggplot(df_long, aes(x = value)) +
          geom_histogram() +
          facet_wrap(~variable, scales = "free") +
          labs(x= "Value", y = "Frequency")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

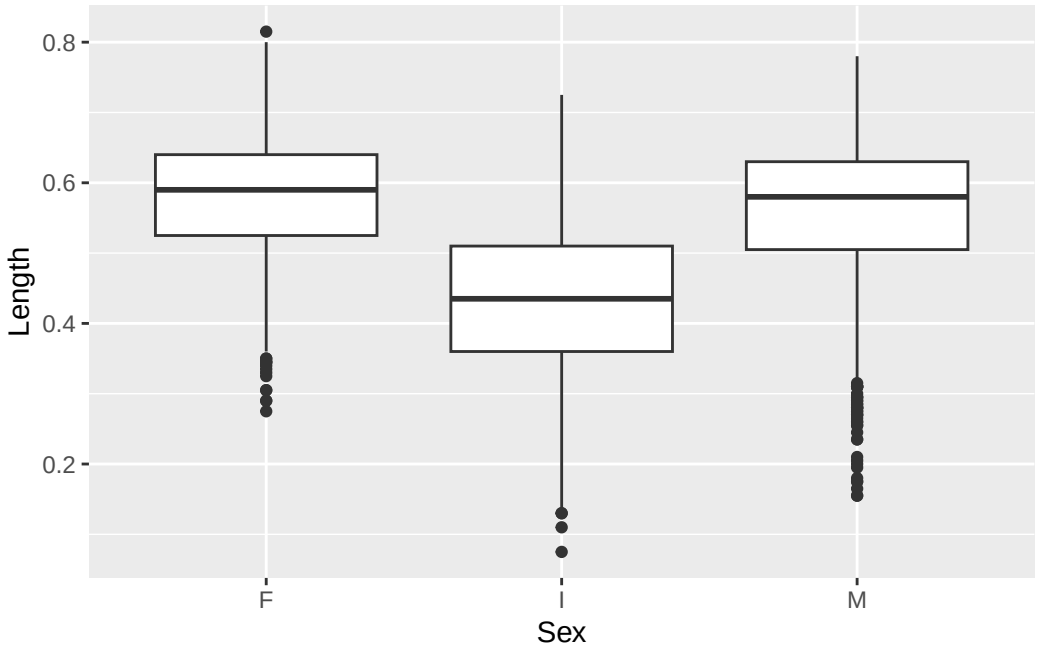


1.4 (5 points)

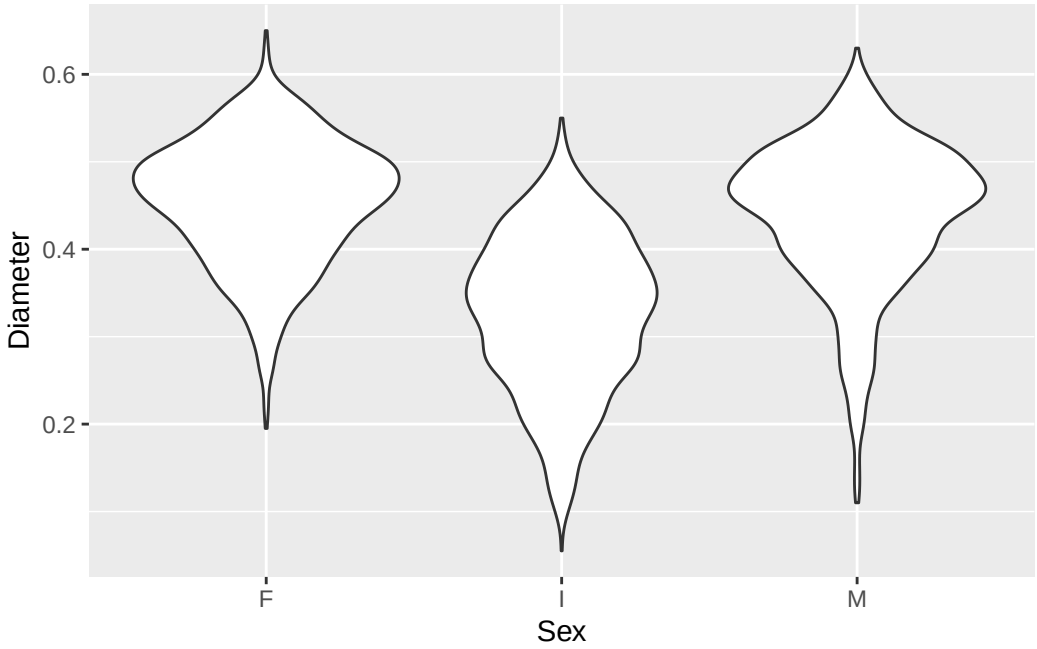
Create a boxplot of `length` for each `sex` and create a violin-plot of of `diameter` for each `sex`. Are there any notable differences in the physical appearences of abalones based on your analysis here?

```r
# Insert your code for boxplot here
boxplot_length <- ggplot(df, aes(x = sex, y = length)) +
```

```
  geom_boxplot() +
  labs(x = "Sex", y = "Length")
boxplot_length
```




```
 # Insert your code for violinplot here
violinplot_diameter <- ggplot(df, aes(x = sex, y = diameter)) +
  geom_violin() +
  labs(x = "Sex", y = "Diameter")
violinplot_diameter
```

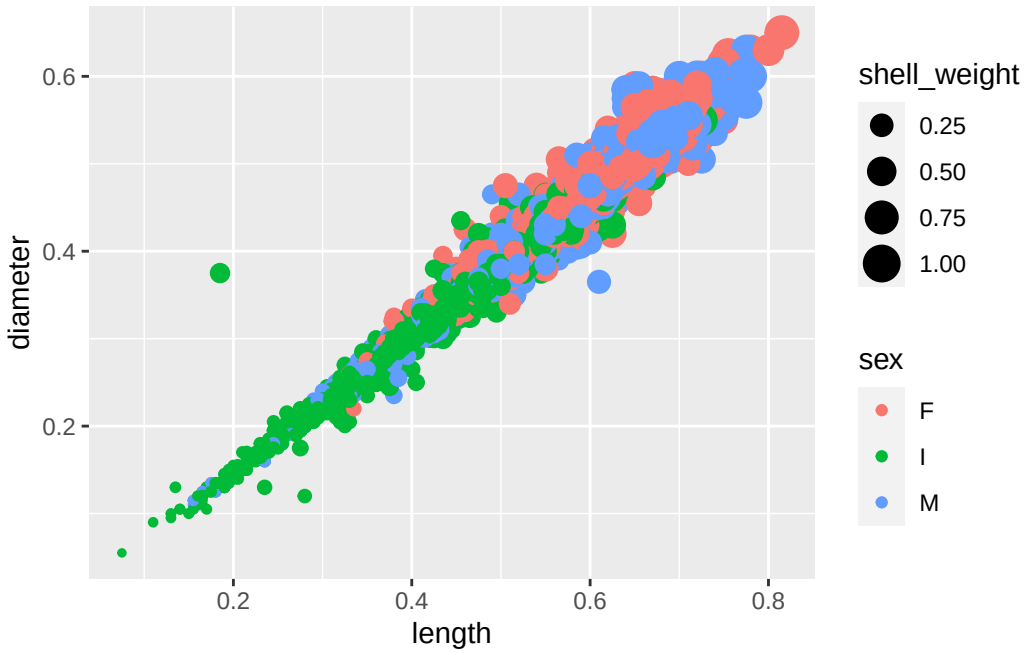


Female abalones have a longer length and a bigger diameter.

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?

```
 # Insert your code here
plot <- ggplot(df, aes(x = length, y = diameter, color = sex, size = shell_weight)) + geom
plot
```



The main anomaly is a I value at (.19,.039) with a smaller shell weight. The diameter is much bigger than standard for the length. There is another cluster of I points that are apart from the group, but they have smaller diameters than standard for their given length.

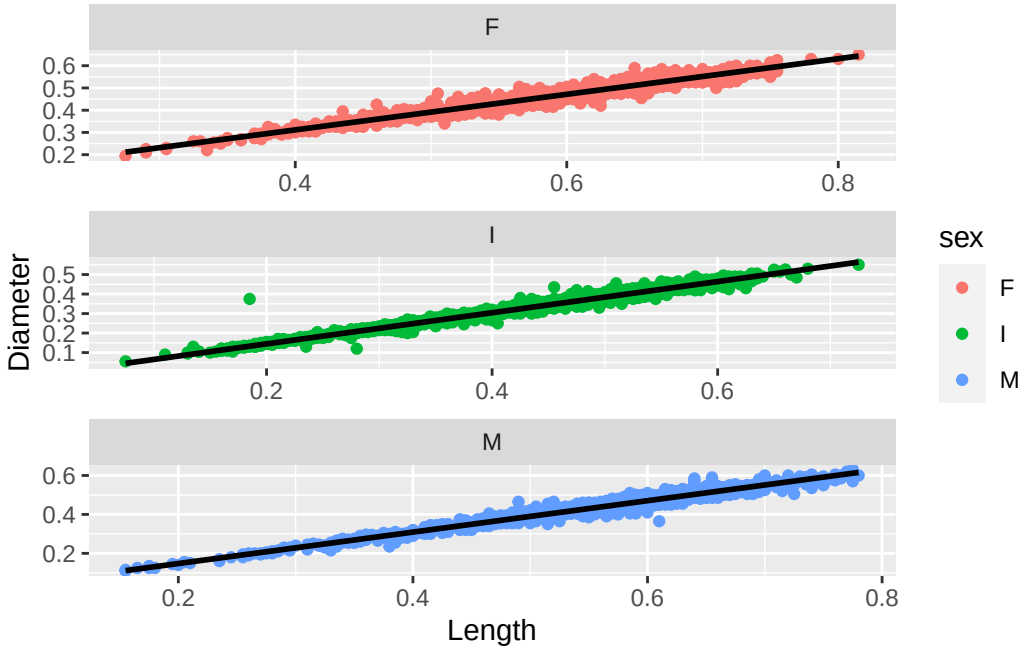
1.6 (5 points)

For each `sex`, create separate scatter plots of `length` and `diameter`. For each plot, also add a **linear** trend line 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: [1]

```
scatter_plots <- ggplot(df, aes(x = length, y = diameter, color = sex)) +
          geom_point() +
          geom_smooth(method = "lm", se = FALSE, color = "black") +
          facet_wrap(~ sex, scales = "free", ncol = 1) +
          labs(x= "Length", y = "Diameter")
scatter_plots
```

[1] Plot example for 1.6

```
`geom_smooth()` using formula = 'y ~ x'
```



Question 2

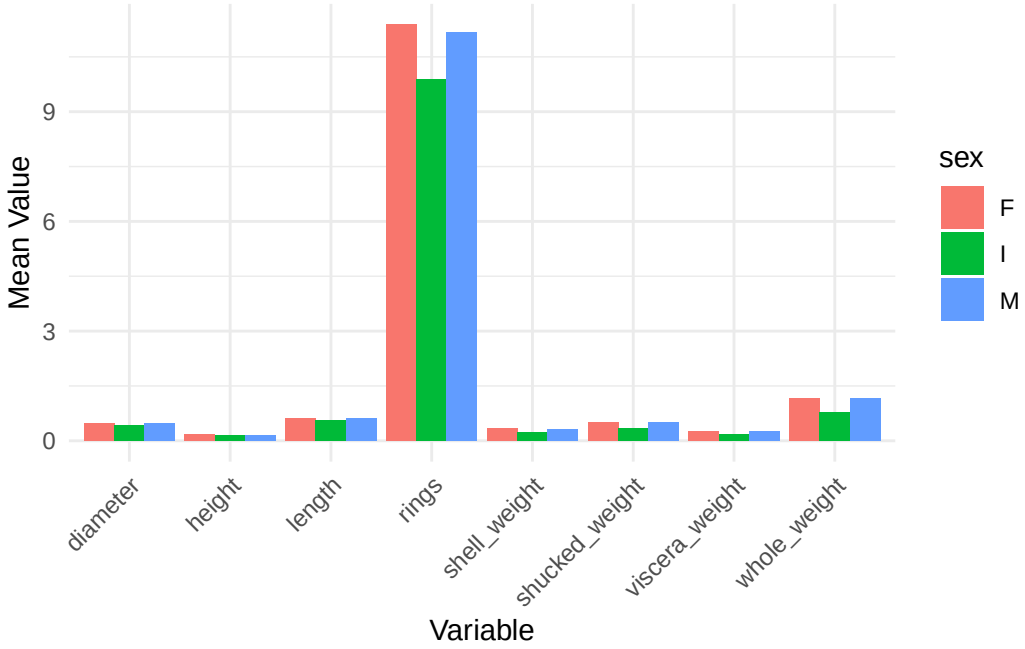
> 40 points
>
> More advanced analyses using `dplyr`, `purrrr` and `ggplot2`

2.1 (10 points)

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

```
filtered <- df %>% filter(length >= .5)
```

```r
# Group the  by sex and calculate the mean of each group's variables
mean_values <- filtered %>%
  group_by(sex) %>%
  summarize_all(mean)

mean_values_long <- pivot_longer(mean_values, -sex, names_to = "variable", values_to = "me

# Create a bar plot to visualize the mean values for each variable by sex
bar_plot <- ggplot(mean_values_long, aes(x = variable, y = mean_value, fill = sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Variable", y = "Mean Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for bet

bar_plot
```



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

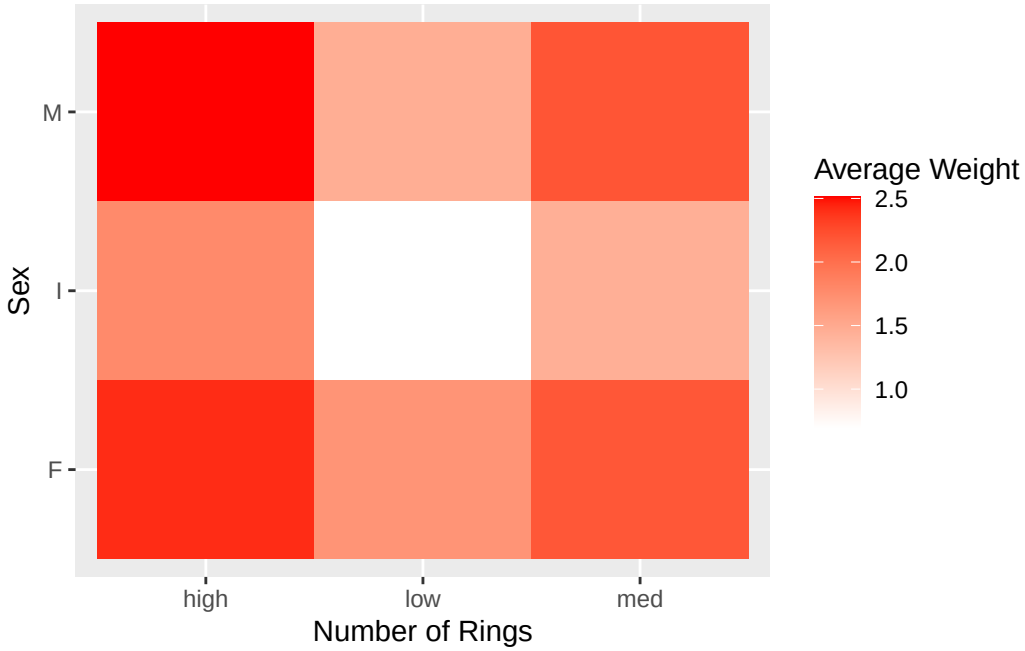
2. Group `df` by this new variable and `sex` and compute `avg_weight` as the average of the `whole_weight + shucked_weight + viscera_weight + shell_weight` for each combination of `num_rings` and `sex`.

3. 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.

```
df_plot <- df %>%
  mutate(num_rings = case_when(
    rings < 10 ~ "low",
    rings > 20 ~ "high",
    TRUE ~ "med"
  )) %>%
  group_by(num_rings, sex) %>%
  summarize(avg_weight = mean(whole_weight + shucked_weight + viscera_weight + shell_weigh
  ggplot(aes(x = num_rings, y = sex, fill = avg_weight)) +
  geom_tile() +
  labs(x = "Number of Rings", y = "Sex", fill = "Average Weight") +
  scale_fill_gradient(low = "white", high = "red")
```

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

```
df_plot
```



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 [2]

```
# Compute pairwise correlations rounded to 2 decimal points
correlation_table <- df %>%
  select_if(is.numeric) %>%
  cor()

round(correlation_table, 3)
```

```
                length diameter height whole_weight shucked_weight
length           1.000    0.987  0.828        0.925          0.898
diameter         0.987    1.000  0.834        0.925          0.893
height           0.828    0.834  1.000        0.819          0.775
whole_weight     0.925    0.925  0.819        1.000          0.969
shucked_weight   0.898    0.893  0.775        0.969          1.000
```

[2] Table for 2.3

```
viscera_weight  0.903    0.900  0.798        0.966          0.932
shell_weight    0.898    0.905  0.817        0.955          0.883
rings           0.557    0.575  0.557        0.540          0.421
               viscera_weight shell_weight rings
length                  0.903        0.898 0.557
diameter                0.900        0.905 0.575
height                  0.798        0.817 0.557
whole_weight            0.966        0.955 0.540
shucked_weight          0.932        0.883 0.421
viscera_weight          1.000        0.908 0.504
shell_weight            0.908        1.000 0.628
rings                   0.504        0.628 1.000
```

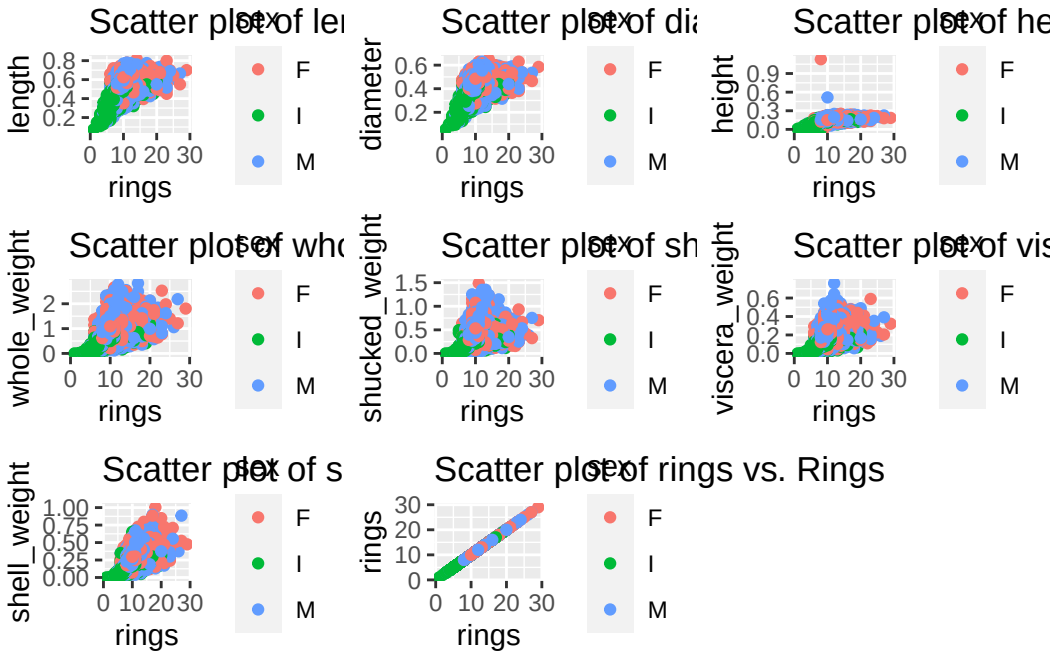
2.4 (10 points)

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 <- df %>%
  select_if(is.numeric)

# Create scatter plots for each quantitative variable
scatter_plots <- map2(quantitative_vars, names(quantitative_vars), function(var, name) {
  ggplot(df, aes_string(x = "rings", y = name, color = "sex")) +
    geom_point() +
    labs(title = paste("Scatter plot of", name, "vs. Rings"))
})
```

```
Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
i Please use tidy evaluation idioms with `aes()`.
i See also `vignette("ggplot2-in-packages")` for more information.
```

```
# Arrange grid
grid <- plot_grid(plotlist = scatter_plots, ncol = 3)
grid
```



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.

```
lm_model <- lm(height ~ diameter, data = df)
lm_model
```

```
Call:
lm(formula = height ~ diameter, data = df)
```

```
Coefficients:
(Intercept)     diameter
  -0.003803     0.351376
```

```r
summary(lm_model)
```

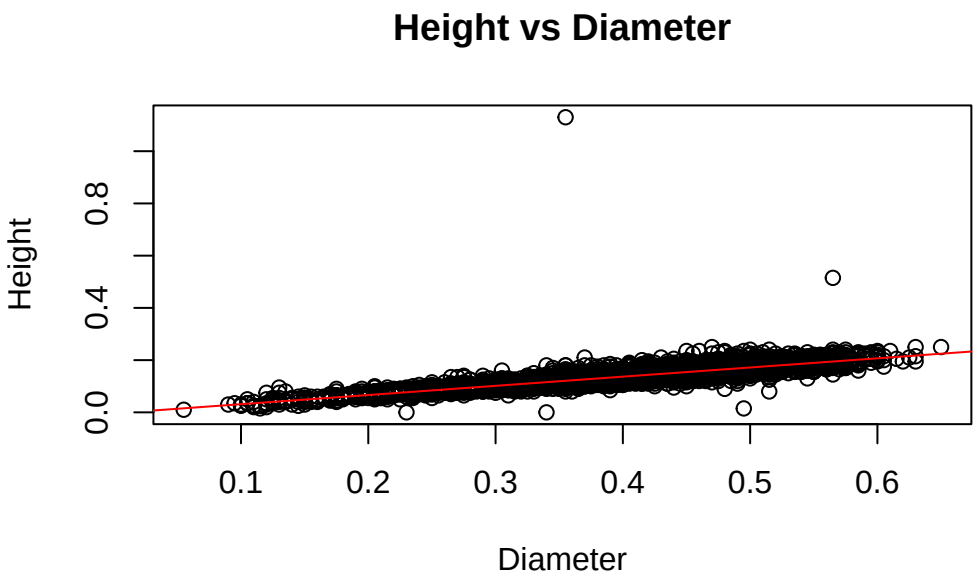
```
Call:
lm(formula = height ~ diameter, data = df)

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
```

The model coefficients are the intercept which is -.003803 and the slope which is .351376. This means that the model says, if diameter could be zero, the associated height would be -.003803. The p-value for the intercept was .0119 which means it is significant since it is below .05. Also, for every unit of increase in diameter, the height increases by .351376. The p-value for the slope was <2e-16 meaning that it was even more significant since it is well below .05.

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.

```r
plot(df$diameter, df$height, main = "Height vs Diameter", xlab = "Diameter", ylab = "Heigh
abline(lm(height ~ diameter, data = df), col = "red")
```

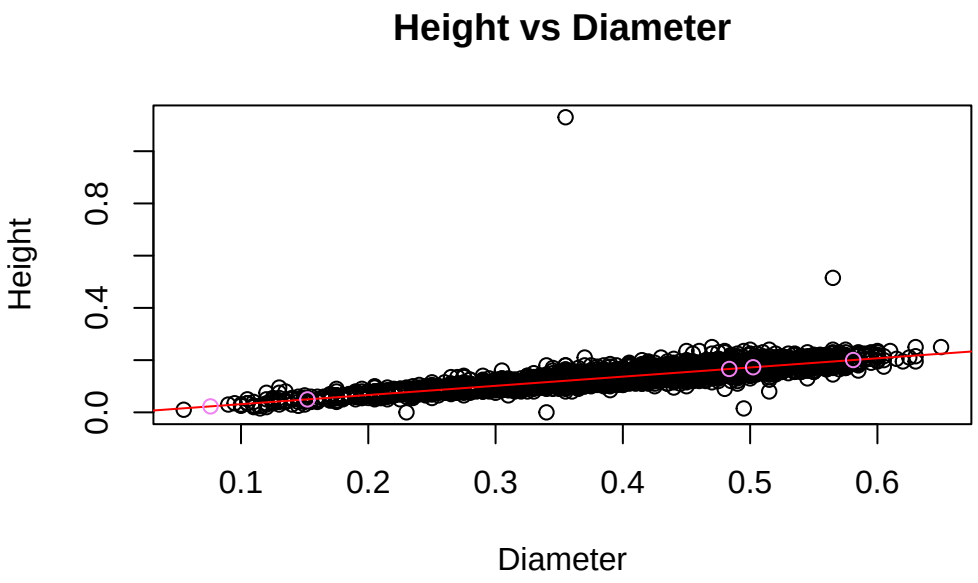


3.3 (10 points)

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"`

```r
new_diameters <- c(
  0.15218946,
  0.48361548,
  0.58095513,
  0.07603687,
  0.50234599,
  0.83462092,
  0.95681938,
  0.92906875,
  0.94245437,
  0.01209518
)


new_predictions <- predict(lm_model, newdata = data.frame(diameter = new_diameters))
```

```r
plot(df$diameter, df$height, main = "Height vs Diameter", xlab = "Diameter", ylab = "Heigh
abline(lm(height ~ diameter, data = df), col = "red")

new_predictions <- predict(lm_model, newdata = data.frame(diameter = new_diameters))
points(new_diameters, new_predictions, col = "violet")
```



—

Appendix

Session Information

Print your R session information using the following command

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Big Sur 11.7.4

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] cowplot_1.1.3 purrr_1.0.2   dplyr_1.1.4   ggplot2_3.4.4 tidyr_1.3.0
[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    tidyselect_1.2.0  parallel_4.3.2
 [9] splines_4.3.2     scales_1.3.0      yaml_2.3.8        fastmap_1.1.1
[13] lattice_0.21-9    R6_2.5.1          labeling_0.4.3    generics_0.1.3
[17] curl_5.2.0        knitr_1.45        tibble_3.2.1      munsell_0.5.0
[21] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.3       utf8_1.2.4
[25] xfun_0.41         bit64_4.0.5       cli_3.6.2         mgcv_1.9-0
[29] withr_3.0.0       magrittr_2.0.3    digest_0.6.34     grid_4.3.2
```

```
[33] vroom_1.6.5       rstudioapi_0.15.0 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
```