Homework 2

Milo Schmitt{style='background-color: yellow;'}

Table of contents

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.

```r
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) # Insert your code here
```

```
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.
```

1.2 (5 points)

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

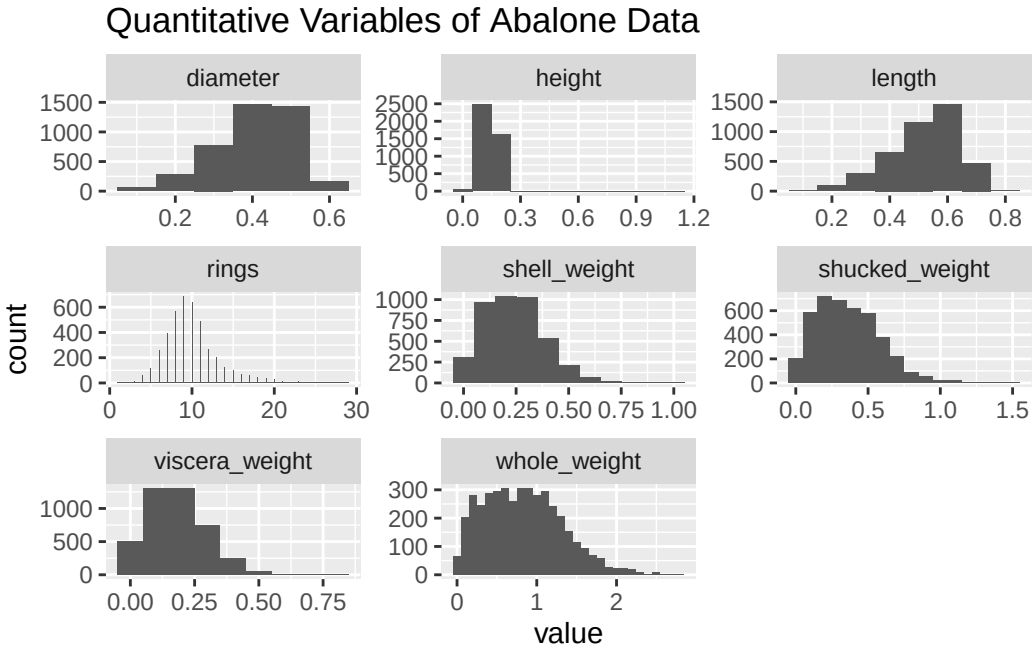
```r
df <- na.omit(abalone) # Insert your code here
```

none of the rows were dropped.

1.3 (5 points)

Plot histograms of all the quantitative variables in a **single plot** [1]

```
# Insert your code here

 quant_vars <- select_if(df, is.numeric)

 quant_vars <- tidyr::pivot_longer(quant_vars, everything(), names_to = "variable", value

 ggplot(quant_vars, aes(x = value)) +
          geom_histogram(binwidth = 0.1) +
          facet_wrap(~ variable, scales = "free") +
          labs(title = "Quantitative Variables of Abalone Data")
```

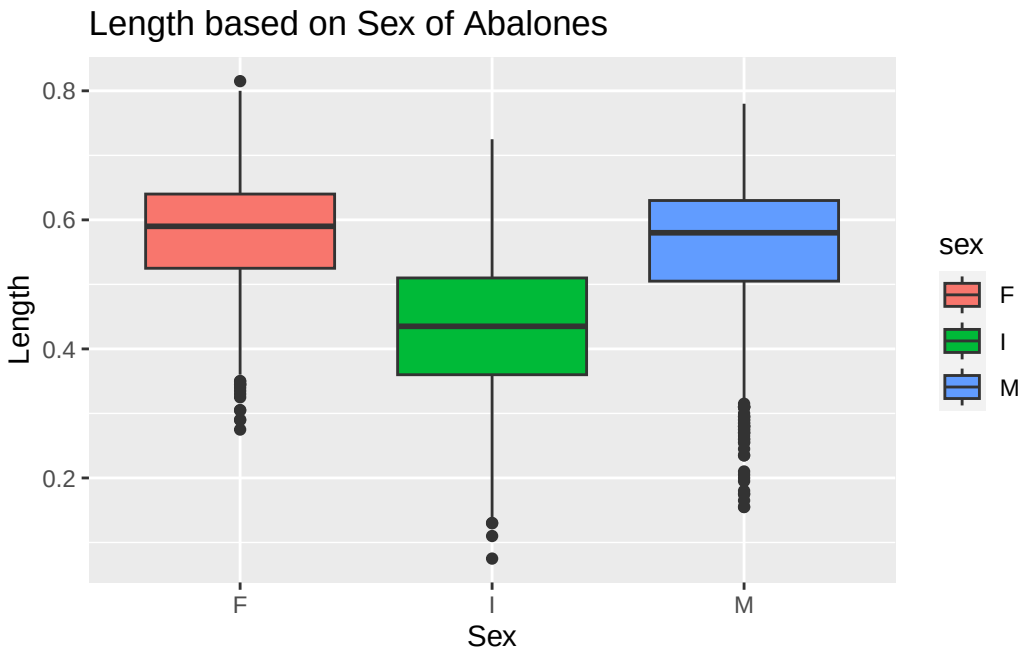


1.4 (5 points)

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

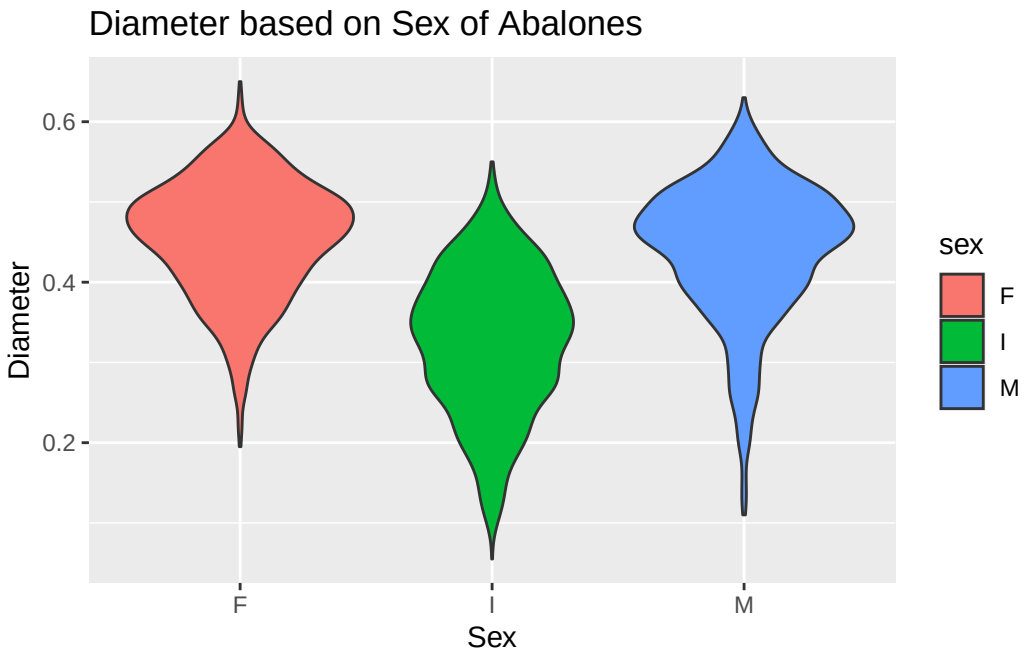
Create a boxplot of `length` for each `sex` and create a violin-plot of of `diameter` for each `sex`. Are there any notable differences in the physical appearences of abalones based on your analysis here?

```
# Insert your code for boxplot here
 ggplot(df, mapping = aes(x = sex, y = length, fill = sex)) +
 geom_boxplot() +
 labs(title = "Length based on Sex of Abalones",
      x = "Sex",
      y = "Length")
```



It seems that Male and Female Abalones have similar lengths, while the 'I' sex is noticeably shorter in most cases.

```
# Insert your code for violinplot here
 ggplot(df, mapping = aes(x = sex, y = diameter, fill = sex)) +
 geom_violin() +
 labs(title = "Diameter based on Sex of Abalones",
      x = "Sex",
      y = "Diameter")
```

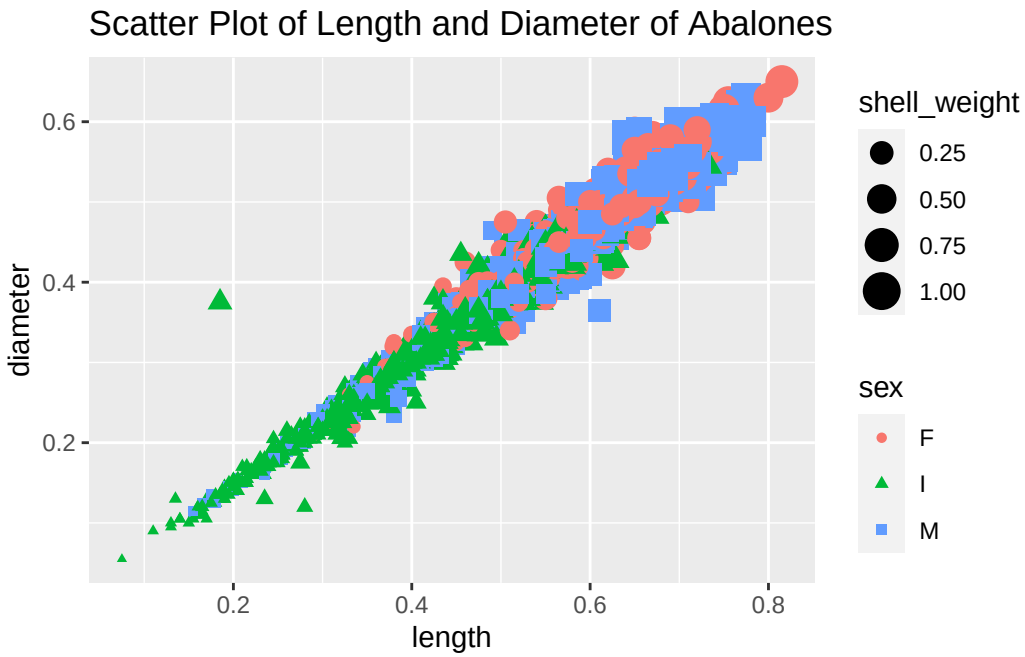


Sex has a similar impact on the diameter of an abalone as it does on the length. Males and females are similar, while 'I' is smaller.

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

ggplot(df, mapping = aes(x = length, y = diameter, shape = sex, color = sex, size = shell_
  geom_point() +
  labs(title = "Scatter Plot of Length and Diameter of Abalones")
```



It seems that Abalone length and diameter are positively correlated. Higher length indicates a strong chance of having a higher diameter. There is one anomaly, however, which has a length of ~0.2 and a diameter of almost 0.4. One could hypothesize that this abalone has some sort of condition or mutation which sets it apart from the others.

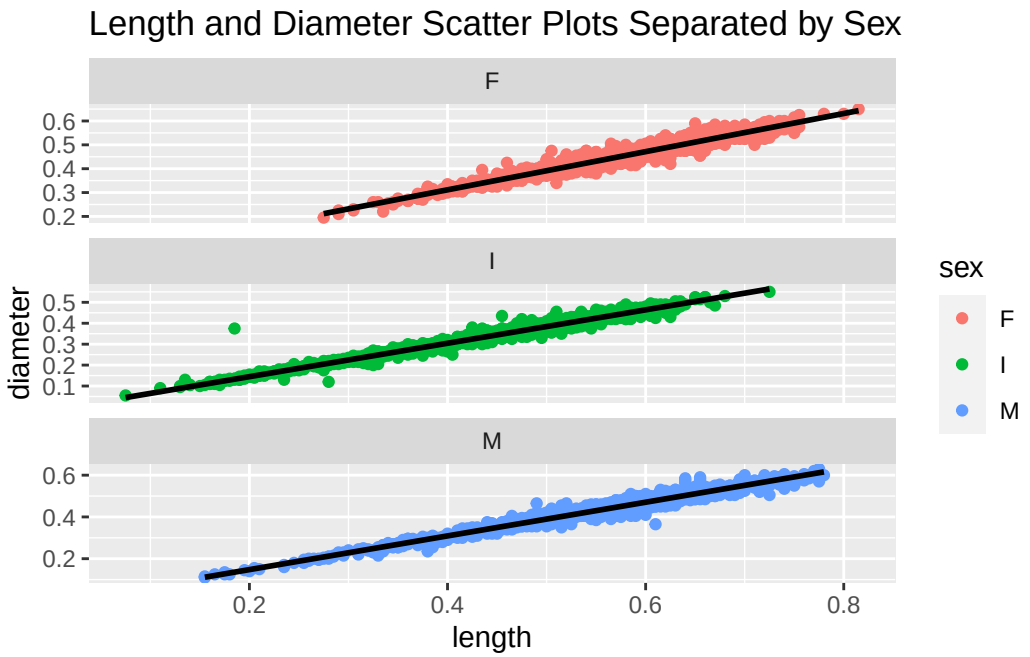
1.6 (5 points)

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]

```
# Insert your code here
  ggplot(df, mapping = aes(x = length, y = diameter, color = sex)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  facet_wrap(~ sex, scales = "free_y", ncol = 1) +
  labs(title = "Length and Diameter Scatter Plots Separated by Sex")
```

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

[2]Plot example for 1.6



Question 2

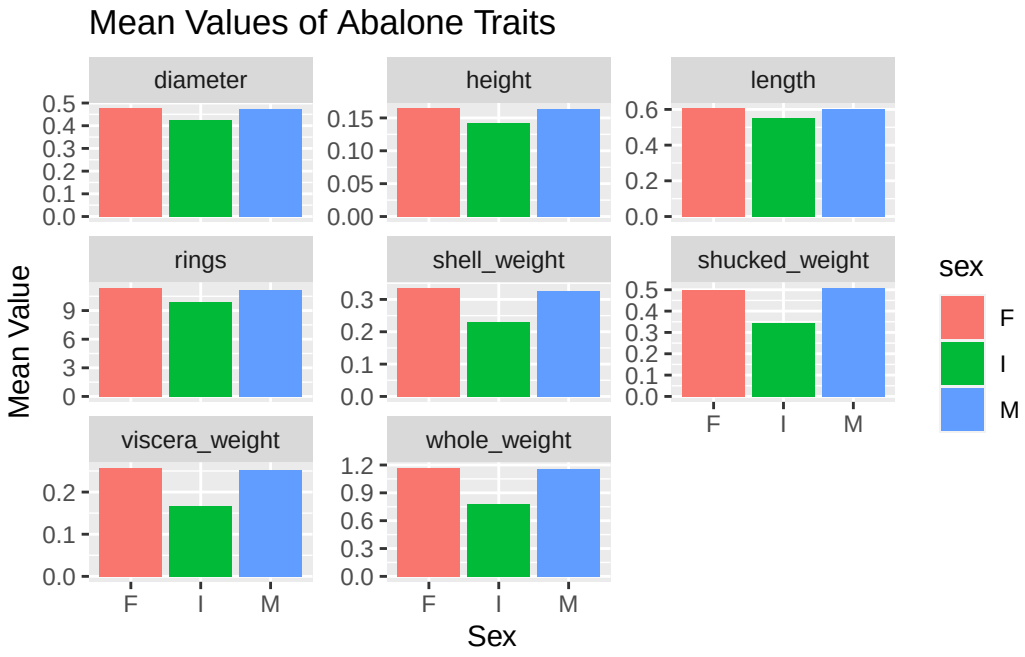
> 40 points
>
> More advanced analyses using `dplyr`, `purrrr` and `ggplot2`

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 <- df %>%
  filter(length >= 0.5)

means <- filtered_df %>%
  group_by(sex) %>%
```

```r
  summarise_all(mean)

means2 <- tidyr::pivot_longer(means, -sex, names_to = "variable", values_to = "mean_value"

ggplot(means2, mapping = aes(x = sex, y = mean_value, fill = sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ variable, scales = "free_y") +
  labs(title = "Mean Values of Abalone Traits",
       x = "Sex",
       y = "Mean Value")
```



2.2 (15 points)

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

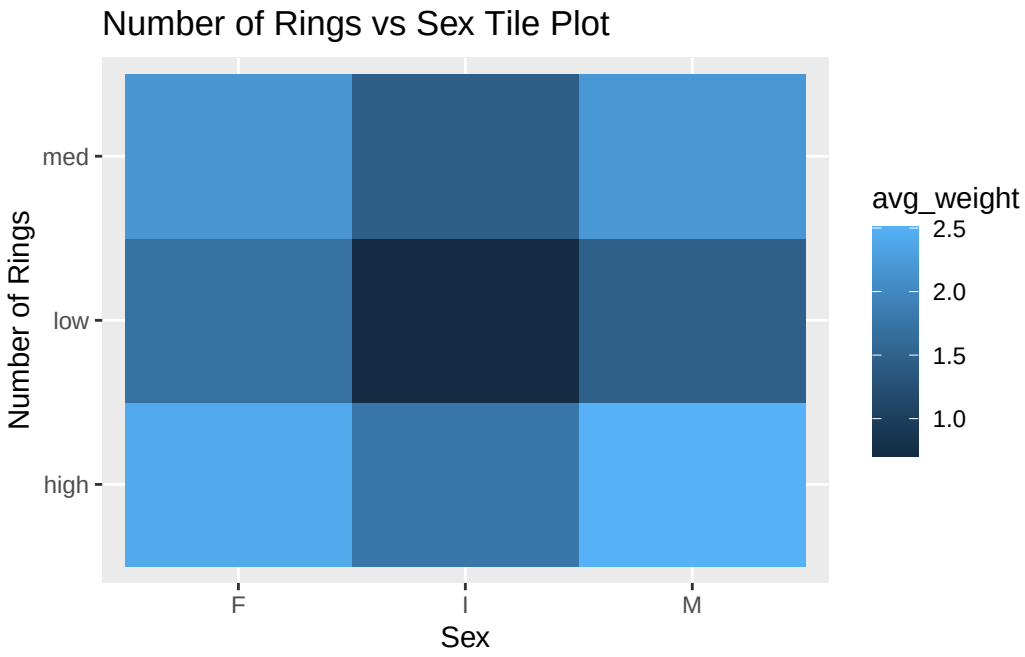
1. Temporarily create a new variable called `num_rings` which takes a value of:

- `"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 %>%
  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_weigh
  ggplot(df, mapping = aes(x = sex, y = num_rings, fill = avg_weight)) +
  geom_tile() +
  labs(title = "Number of Rings vs Sex Tile Plot",
       x = "Sex",
       y = "Number of Rings")
```

```
`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]

```
df_numeric <- select_if(df, is.numeric)

corr_table <- round(cor(df_numeric), 2)

corr_table
```

	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

[3] Table for 2.3

```
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
```

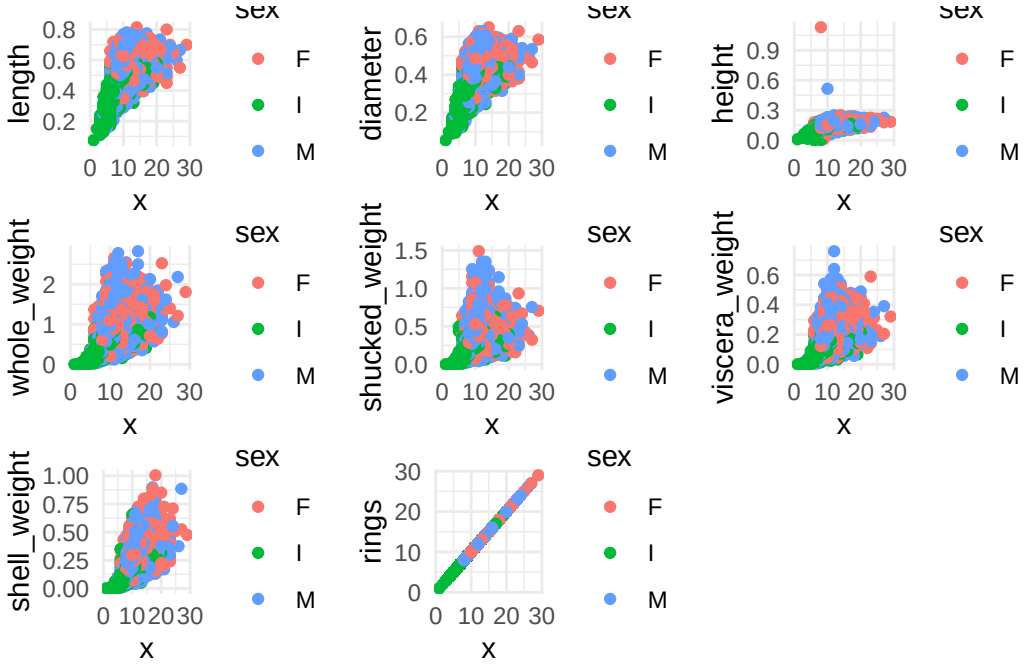
2.4 (10 points)

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.

```
abalone_plots <- map2(names(df_numeric), list(df$rings), ~{
  ggplot(df, aes_string(x = .y, y = .x, color = "sex")) +
    geom_point() +
    theme_minimal()
})
```

```
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.
```

```
plot_grid(plotlist = abalone_plots, ncol = 3)
```



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 = df)

summary(model)
```

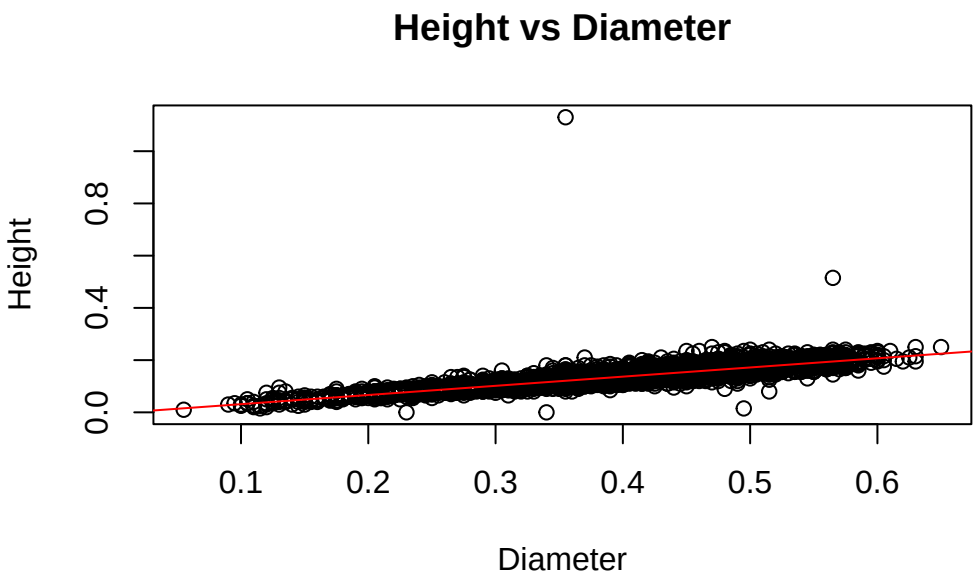
```
Call:
lm(formula = height ~ diameter, data = df)

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 coefficient for diameter is ~0.35, indicating a 0.35 unit height increase for every one unit diameter increase. The extremely small p-value indicates that diameter has a significant correlation with height.

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, xlab = "Diameter", ylab = "Height", main = "Height vs Diamete
abline(model, col = "red")
```



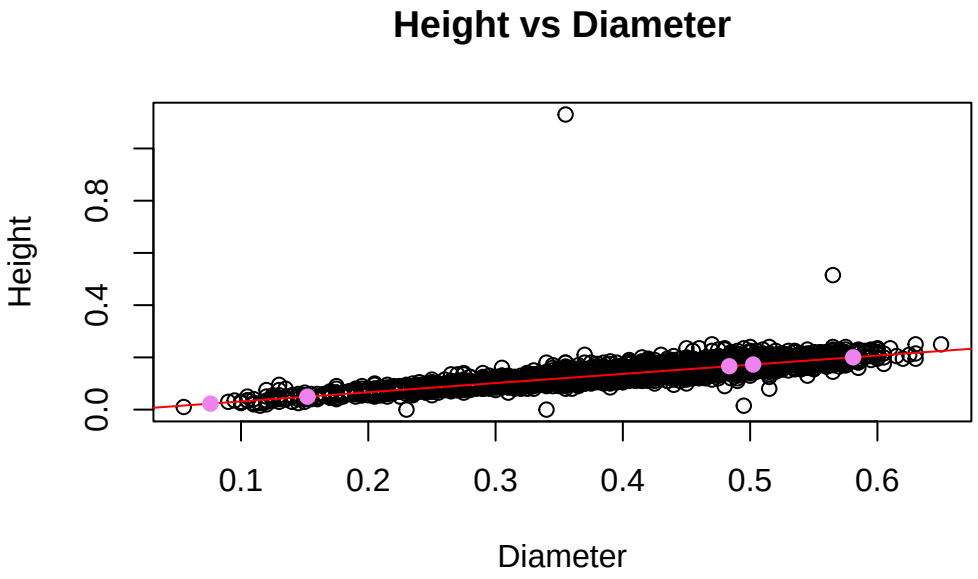
The scatterplot follows a very linear trend, so the line is a good fit for the data.

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
)
```

```r
predict_height <- predict(model, newdata = data.frame(diameter = new_diameters))

plot(df$diameter, df$height, xlab = "Diameter", ylab = "Height", main = "Height vs Diamete
abline(model, col = "red")
points(new_diameters, predict_height, col = "violet", pch = 19)
```



—

Appendix

Session Information

Print your R session information using the following command

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Monterey 12.6

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_1.1.1 purrr_1.0.2   dplyr_1.1.4   ggplot2_3.4.4 tidyr_1.3.0
[6] readr_2.1.4

loaded via a namespace (and not attached):
 [1] Matrix_1.6-4      bit_4.0.5         gtable_0.3.4      jsonlite_1.8.7
 [5] crayon_1.5.2      compiler_4.3.1    tidyselect_1.2.0  parallel_4.3.1
 [9] splines_4.3.1     scales_1.3.0      yaml_2.3.7        fastmap_1.1.1
[13] lattice_0.22-5    R6_2.5.1          labeling_0.4.3    generics_0.1.3
[17] curl_5.1.0        knitr_1.45        tibble_3.2.1      munsell_0.5.0
[21] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.2       utf8_1.2.4
[25] xfun_0.41         bit64_4.0.5       cli_3.6.1         mgcv_1.9-0
[29] withr_2.5.2       magrittr_2.0.3    digest_0.6.33     grid_4.3.1
```

```
[33] vroom_1.6.3       rstudioapi_0.15.0 hms_1.1.3         nlme_3.1-164
[37] lifecycle_1.0.4   vctrs_0.6.4       evaluate_0.23     glue_1.6.2
[41] farver_2.1.1      fansi_1.0.5       colorspace_2.1-0  rmarkdown_2.25
[45] tools_4.3.1       pkgconfig_2.0.3   htmltools_0.5.7
```