Homework 2

Gaurang Kakade

Table of contents

Link to the Github repository

❗ Due: Tue, Feb 14, 2023 @ 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:

```r
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
```

```r
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",
```

```r
    "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.
```

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 <- abalone %>%
    drop_na()

  no_of_rows = nrow(abalone) - nrow(df)
  no_of_rows
```

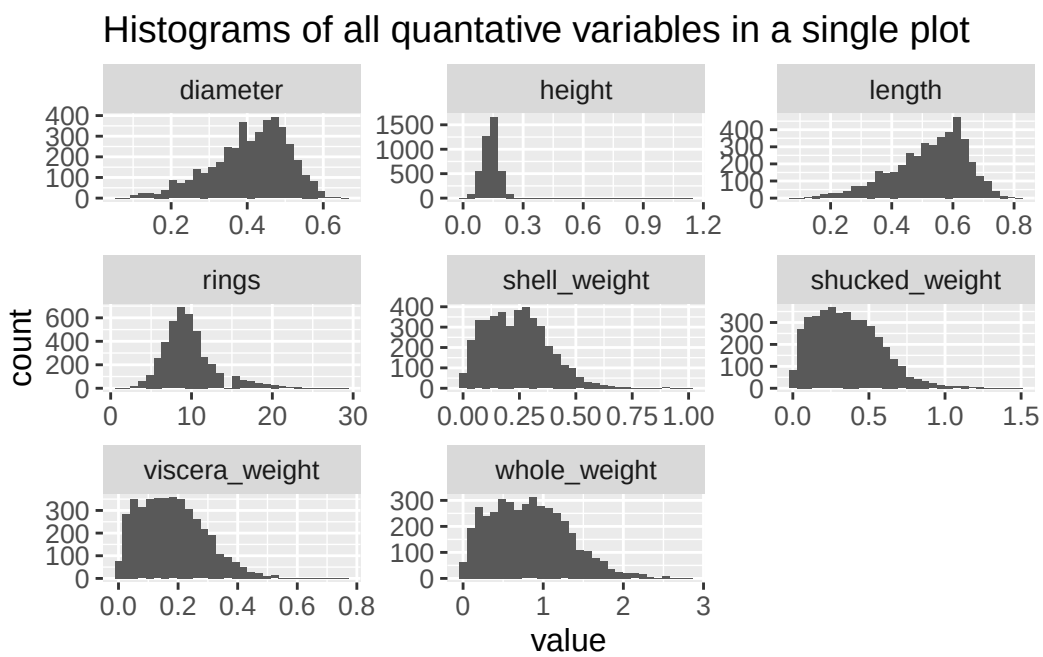
```
[1] 0
```

1.3 (5 points)

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

```
df %>%
  select(!sex) %>%
  gather() %>%
ggplot(
  aes(value)) +
  facet_wrap(~key, scales = 'free') +
  geom_histogram() +
  ggtitle("Histograms of all quantative variables in a single plot") +
  theme(
    text = element_text(size = 12)
  )
```

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

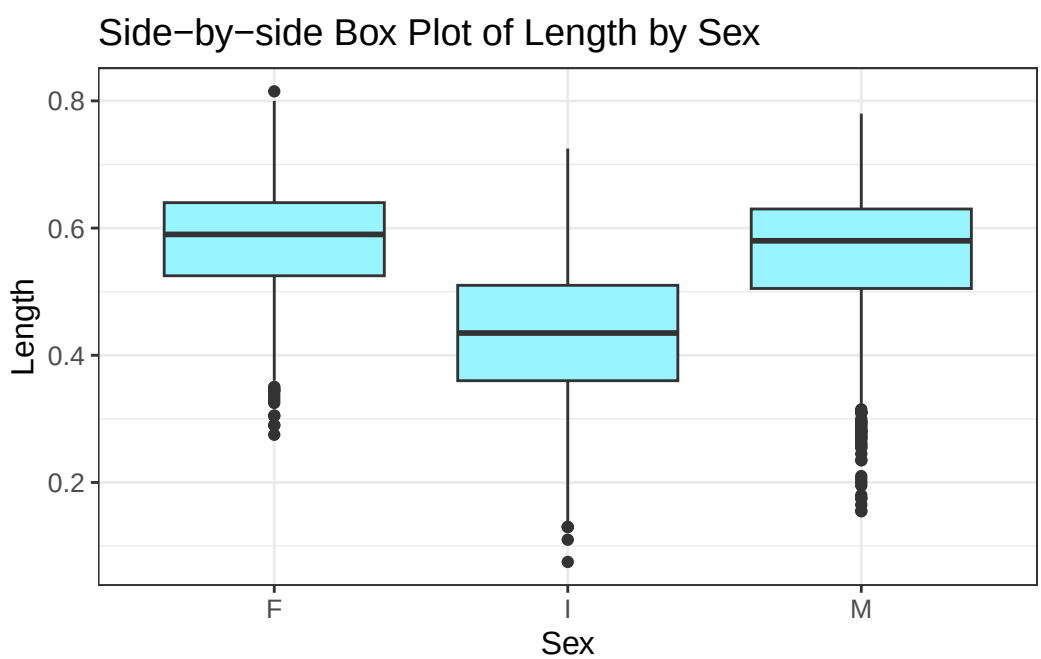


[1] You can use the `facet_wrap()` function for this. Have a look at its documentation using the help console in R

1.4 (5 points)

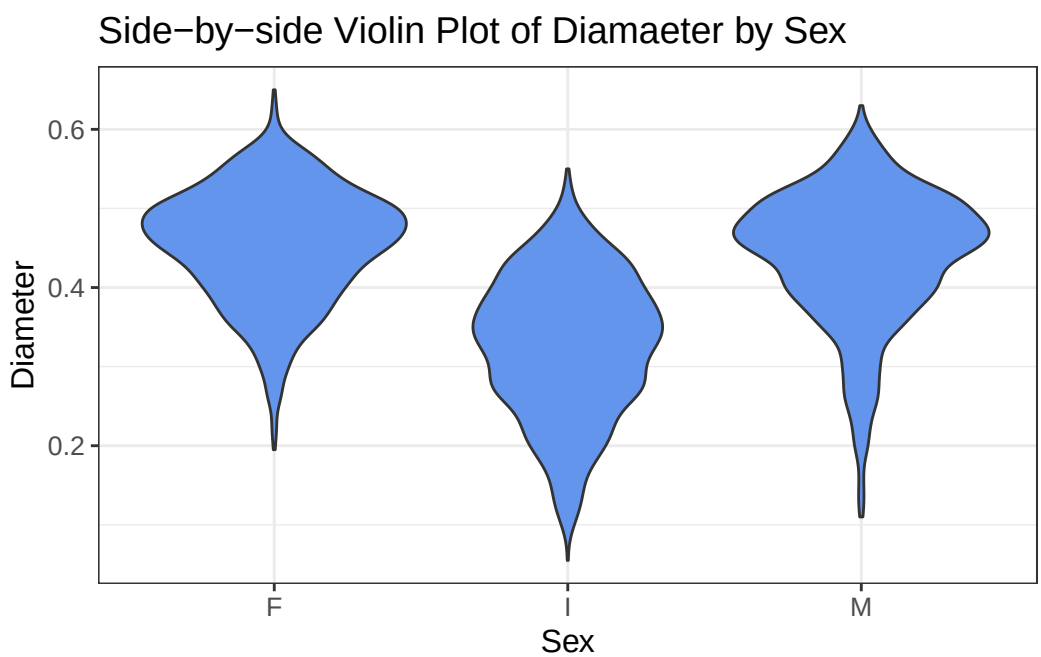
Create a boxplot of `length` for each `sex` and create a violin-plot of of `diameter` for each `sex`. Are there any notable differences in the physical appearences of abalones based on your analysis here?

```
library(ggplot2)
ggplot(
  data = df,
  mapping = aes(x = sex, y = length)
) +
  geom_boxplot(fill = "cadetblue1") + # Add color
  theme_bw() +
  ylab("Length") +
  xlab("Sex") +
  theme(
    text = element_text(size = 12) # Change the base font size
  ) +
  ggtitle("Side-by-side Box Plot of Length by Sex") # Add title
```




```
library(ggplot2)
ggplot(
  data = df,
```

```
  mapping = aes(x = sex, y = diameter)
) +
  geom_violin(fill = "cornflowerblue") +
  theme_bw() +
  ylab("Diameter") +
  xlab("Sex") +
  theme(
    text = element_text(size = 12)
  ) +
  ggtitle("Side-by-side Violin Plot of Diamaeter by Sex")
```



We can observe that how the length and diameter variables are distributed among each sex using the boxplot and violin plot. The boxplot of length reveals that male abalones have slightly bigger median lengths than female and infant abalones. Whereas the violin plot shows that the median diameters of male abalones,female abalones and infant abalones tend to be somewhat similar without any major differences. Whereas, the distribution of female abalones seem to be wider than male and infant abalones. From these plots, one can see notable differences in the physical appearances of abalones. Hence, these differences in the distribution of physical appearance could indicate that the abalones are physically distinct based on their sex.

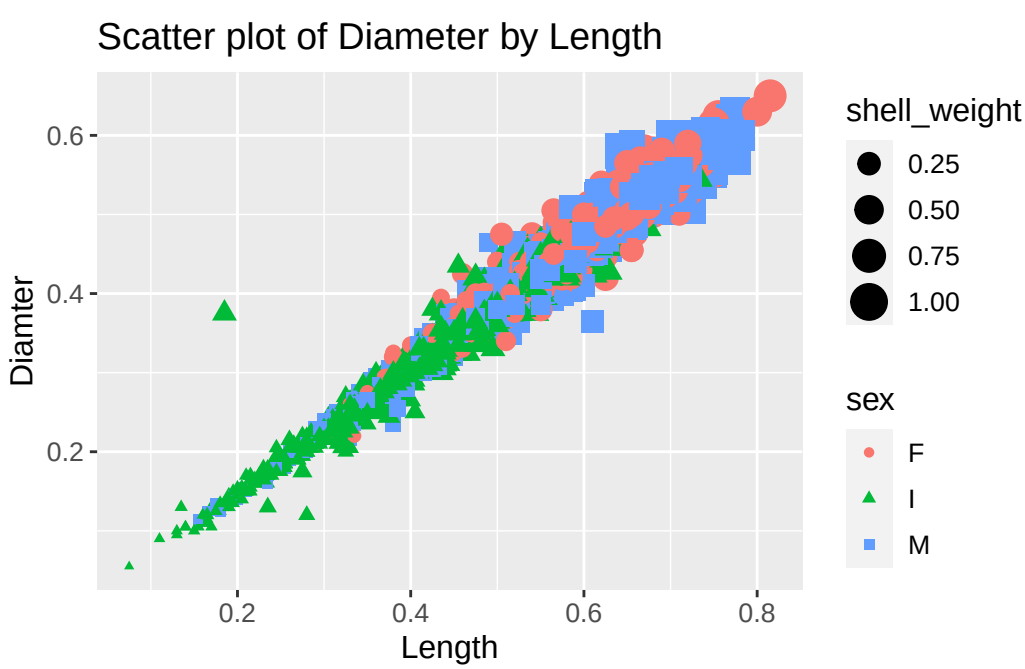
1.5 (5 points)

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?

```
library(ggplot2)
G <- ggplot(
  data = df,
  mapping = aes(x=length, y=diameter, shape=sex, color=sex, size=shell_weight)
) +
  geom_point()
  scale_size_continuous(range = c(1,10))
```

```
<ScaleContinuous>
 Range:
 Limits:    0 --    1
```

```
G + ylab("Diamter") +
    xlab("Length") +
    theme(
      text = element_text(size = 12)
) +
    ggtitle("Scatter plot of Diameter by Length")
```



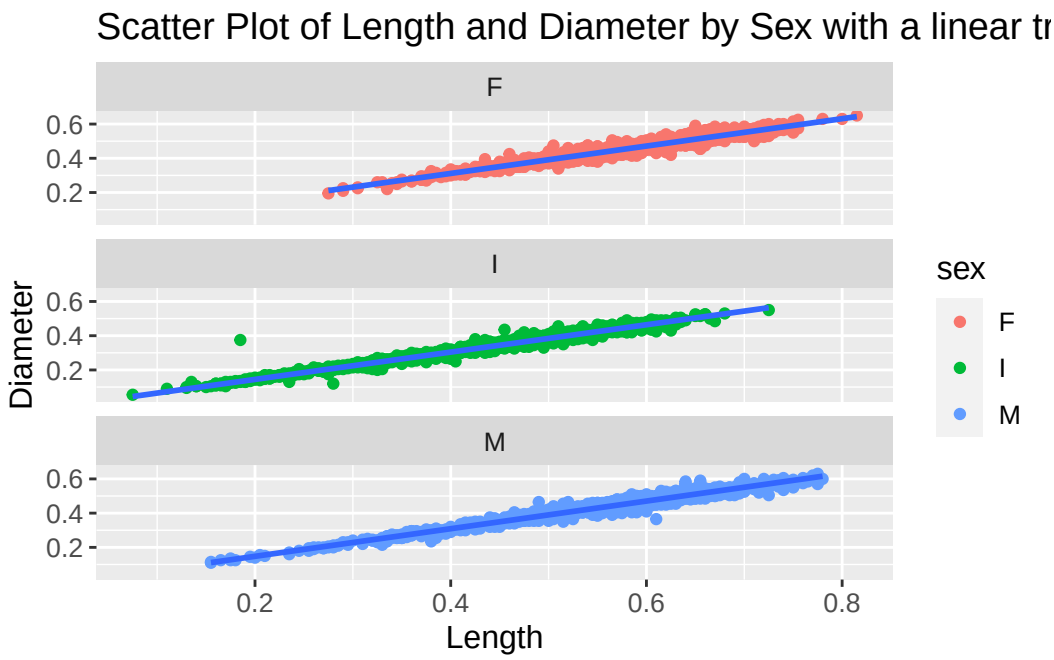
1.6 (5 points)

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]

```
library(ggplot2)
ggplot(
  data = df,
  aes(x = length, y = diameter)) +
  geom_point(aes(color = sex)) +
  geom_smooth(aes(group = sex), method = "lm") +
  facet_wrap(~ sex, ncol = 1) +
  ylab("Diameter") +
  xlab("Length") +
  theme(
    text = element_text(size = 12)) +
  ggtitle("Scatter Plot of Length and Diameter by Sex with a linear trend")
```

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

[2] Plot example for 1.6



—

Question 2

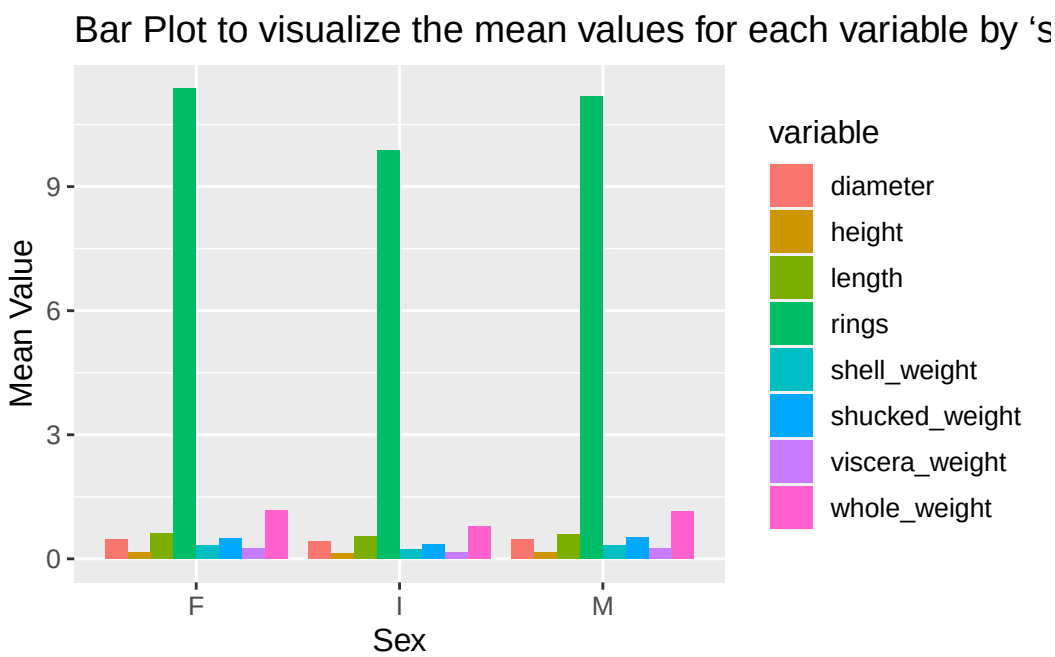
> 40 points
>
> More advanced analyses using `dplyr`, `purrrr` and `ggplot2`

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`.

```
# Filtering that data to include a length of at least 0.5 meters.
 df %>%
  filter(length >= 0.5) %>%
  group_by(sex) %>%
  summarise_all(mean) %>%
  gather(key = "variable", value = "mean_value", -sex) %>%
```

```r
ggplot(
  aes(x = sex, y = mean_value, fill = variable)) +
geom_col(position = 'dodge') +
ylab("Mean Value") +
xlab("Sex") +
theme(
  text = element_text(size = 12)
) +
  ggtitle("Bar Plot to visualize the mean values for each variable by `sex`.")
```



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:

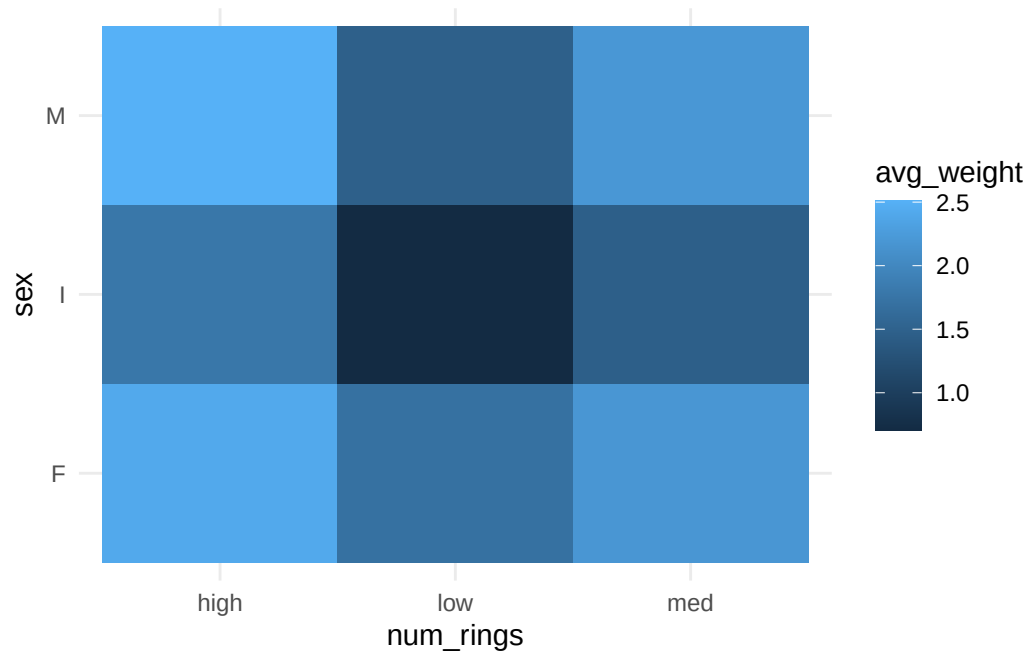
- `"low"` if `rings < 10`
- `"high"` if `rings > 20`, and
- `"med"` otherwise

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.

```
library(dplyr)
ggplot(data = df %>%
         mutate(
           num_rings = ifelse(rings < 10, "low",
                              ifelse(rings > 20, "high", "med"))) %>%
         group_by(num_rings, sex) %>%
         summarize(avg_weight = mean(whole_weight + shucked_weight + viscera_weight + shel
       aes(x = num_rings, y = sex, fill = avg_weight)) +
  geom_tile() +
  xlab("num_rings") +
  ylab("sex") +
  theme(
    text = element_text(size = 12)) +
  theme_minimal()
```

```
`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 %>%
  select_if(is.numeric) %>%
  cor() %>% # The cor() function in R calculates the pairwise correlations
            # between all the numeric variables in a data frame or matrix.
            # It returns a symmetric matrix of the correlation coefficients
            # between each pair of variables.
  round(2) %>% # rounding to decimal points
  as.data.frame() # is used to convert the result of the cor() function from
```

	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

[3] Table for 2.3

```
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
```

```
                    # a matrix to a dataframe
```

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.

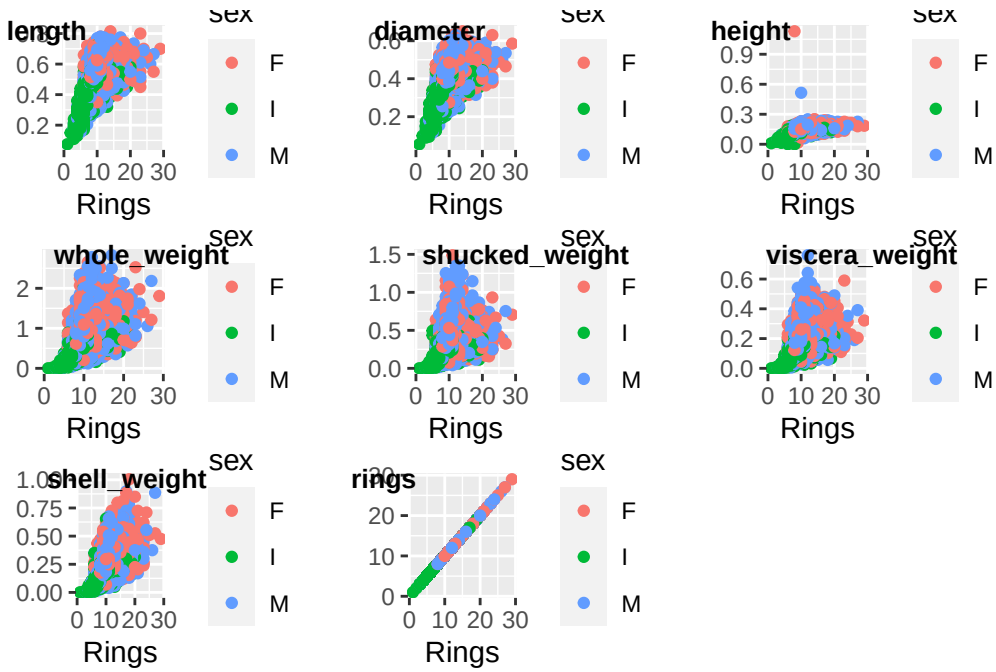
```
library(purrr)
library(ggplot2)

df_quantative <- df %>%
                   select(!sex)

df_1 <-
  df %>%
  select(rings)

plt_1 <- map2(df_quantative, df_1, ~ggplot(df) +
             geom_point(aes(x = rings, y = .x, col = sex)) +
             ylab(" ") +
             xlab("Rings"))

cowplot::plot_grid(plotlist = plt_1, labels = colnames(df_quantative), ncol = 3, label_siz
```



—

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, 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
```

We can observe that the model's intercept is -0.003803 and the coefficient for the variable `diameter` is 0.351376. The p-value for the diameter variable is 2.2e-16 which significantly small. Hence, on the basis of the significance values and the coefficients, we can note that the diameter of abalone has a positive and a significant effect on the variable `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
df %>%
  ggplot(
    mapping = aes(x = diameter, y = height)
  ) +
  geom_point(color = "aquamarine3") +
  geom_smooth(method = "lm", color = "darkblue") +
  ylab("Height") +
  xlab("Diameter") +
  theme(
    text = element_text(size = 12)
  ) +
  ggtitle("Scatter plot of `height` vs `diameter` with a regression line ")
```

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

The `linear model` seems to be an appropriate fit for the relationship betweem `height` and `diameter`. The data points can be seen to be scattered around the regression line and hence indicating a strong linear relationship between the two variables `height` and `diameter`.

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"`

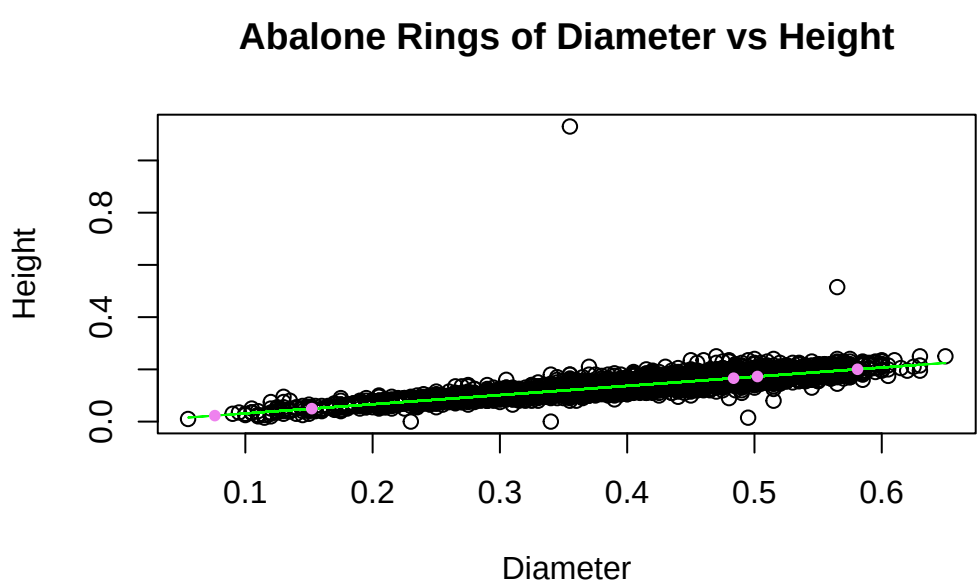
```
new_diameters <- c(
  0.15218946,
  0.48361548,
  0.58095513,
  0.07603687,
  0.50234599,
  0.83462092,
  0.95681938,
```

```
  0.92906875,
  0.94245437,
  0.01209518
)

New_data <- data.frame(diameter = new_diameters)
New_heights <- predict(model,New_data)
New_heights
```

```
           1            2            3            4            5            6
0.0496723682 0.1661276096 0.2003304536 0.0229141546 0.1727090665 0.2894625947
           7            8            9           10
0.3324002348 0.3226493217 0.3273527111 0.0004465615
```

```
# Plotting the new observations alomng with the predictions in the plot
plot(
  df$diameter, y = df$height, ylab = "Height", xlab = "Diameter", pch = 21, main = 'Abalon
lines(df$diameter, fitted(model), col = 'green')
points(new_diameters %>%
         unlist(),
       New_heights,
       col = 'violet',
       pch = 20)
```



The new heights are 0.0496723682, 0.1661276096, 0.2003304536, 0.0229141546, 0.1727090665, 0.289462594, 0.3324002348, 0.3226493217, 0.3273527111, and 0.0004465615 respectively.

Appendix

Session Information

Print your R session information using the following command

```r
sessionInfo()
```

```
R version 4.2.2 (2022-10-31)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur ... 10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.2/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

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

other attached packages:
[1] cowplot_1.1.1 purrr_1.0.1   dplyr_1.0.10  ggplot2_3.4.1 tidyr_1.2.1
[6] readr_2.1.3

loaded via a namespace (and not attached):
 [1] tidyselect_1.2.0 xfun_0.36        splines_4.2.2    lattice_0.20-45
 [5] colorspace_2.0-3 vctrs_0.5.1      generics_0.1.3   htmltools_0.5.4
 [9] yaml_2.3.6       mgcv_1.8-41      utf8_1.2.2       rlang_1.0.6
[13] pillar_1.8.1     glue_1.6.2       withr_2.5.0      DBI_1.1.3
[17] bit64_4.0.5      lifecycle_1.0.3  stringr_1.5.0    munsell_0.5.0
[21] gtable_0.3.1     evaluate_0.20    labeling_0.4.2   knitr_1.41
[25] tzdb_0.3.0       fastmap_1.1.0    parallel_4.2.2   curl_5.0.0
[29] fansi_1.0.3      renv_0.16.0-53   scales_1.2.1     vroom_1.6.0
[33] jsonlite_1.8.4   farver_2.1.1     bit_4.0.5        hms_1.1.2
[37] digest_0.6.31    stringi_1.7.12   grid_4.2.2       cli_3.6.0
[41] tools_4.2.2      magrittr_2.0.3   tibble_3.1.8     crayon_1.5.2
```

```
[45] pkgconfig_2.0.3  ellipsis_0.3.2   Matrix_1.5-1     assertthat_0.2.1
[49] rmarkdown_2.20   rstudioapi_0.14  R6_2.5.1         nlme_3.1-160
[53] compiler_4.2.2
```