

Homework 2

IAdvait Ashtikar

Table of contents

.....	2
Question 1	2
Question 2	9
Question 3	13
Appendix	17
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:

```
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)

```

```

`curl` package not installed, falling back to using `url()`
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.

```

```

abalone %>% head()

```

```

# A tibble: 6 x 9
  sex    length diameter height whole_weight shucked_weight viscera~1 shell~2 rings
<chr> <dbl>    <dbl> <dbl>      <dbl>      <dbl>      <dbl>    <dbl> <dbl>
1 M      0.455    0.365  0.095      0.514      0.224    0.101    0.15     15
2 M      0.35     0.265  0.09       0.226      0.0995   0.0485   0.07      7
3 F      0.53     0.42   0.135      0.677      0.256    0.142    0.21      9
4 M      0.44     0.365  0.125      0.516      0.216    0.114    0.155    10
5 I      0.33     0.255  0.08       0.205      0.0895   0.0395   0.055      7
6 I      0.425    0.3    0.095      0.352      0.141    0.0775   0.12      8
# ... with abbreviated variable names 1: viscera_weight, 2: shell_weight

```

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()
df %>% head()
```

```
# A tibble: 6 x 9
  sex    length diameter height whole_weight shucked_weight visce~1 shell~2 rings
<chr>  <dbl>    <dbl>  <dbl>    <dbl>        <dbl>    <dbl>    <dbl> <dbl>
1 M      0.455    0.365  0.095    0.514        0.224    0.101    0.15    15
2 M      0.35     0.265  0.09     0.226        0.0995   0.0485   0.07     7
3 F      0.53     0.42   0.135    0.677        0.256    0.142    0.21     9
4 M      0.44     0.365  0.125    0.516        0.216    0.114    0.155   10
5 I      0.33     0.255  0.08     0.205        0.0895   0.0395   0.055     7
6 I      0.425    0.3    0.095    0.352        0.141    0.0775   0.12     8
# ... with abbreviated variable names 1: viscera_weight, 2: shell_weight
```

```
# To calculate number of rows dropped
r_drop <- nrow(abalone) - nrow(df)
```

The number of rows dropped were 0.

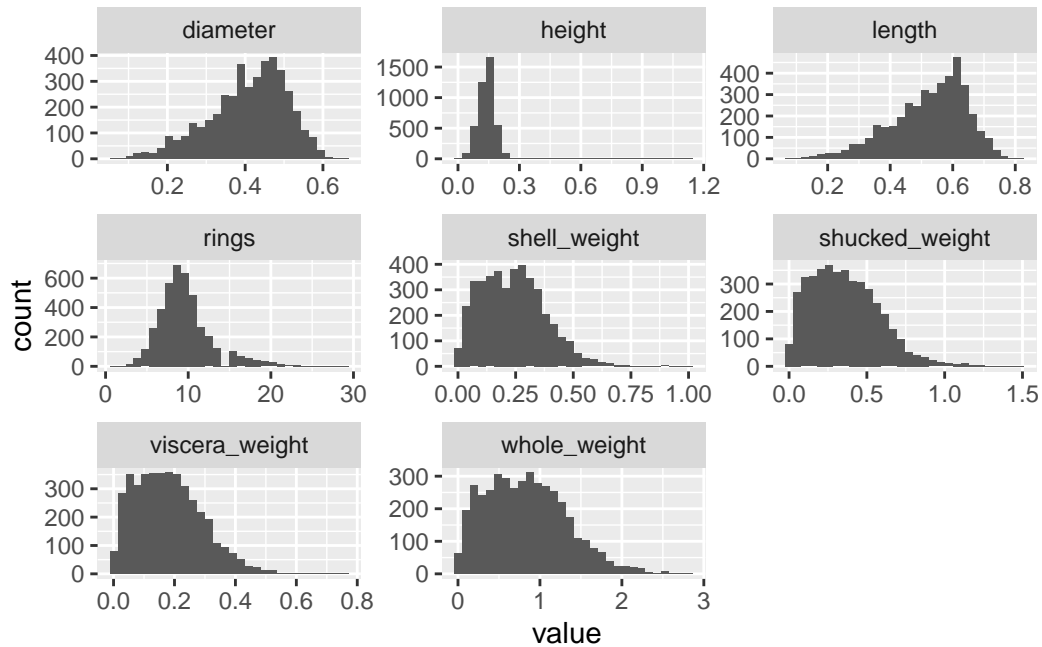
1.3 (5 points)

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()
```

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

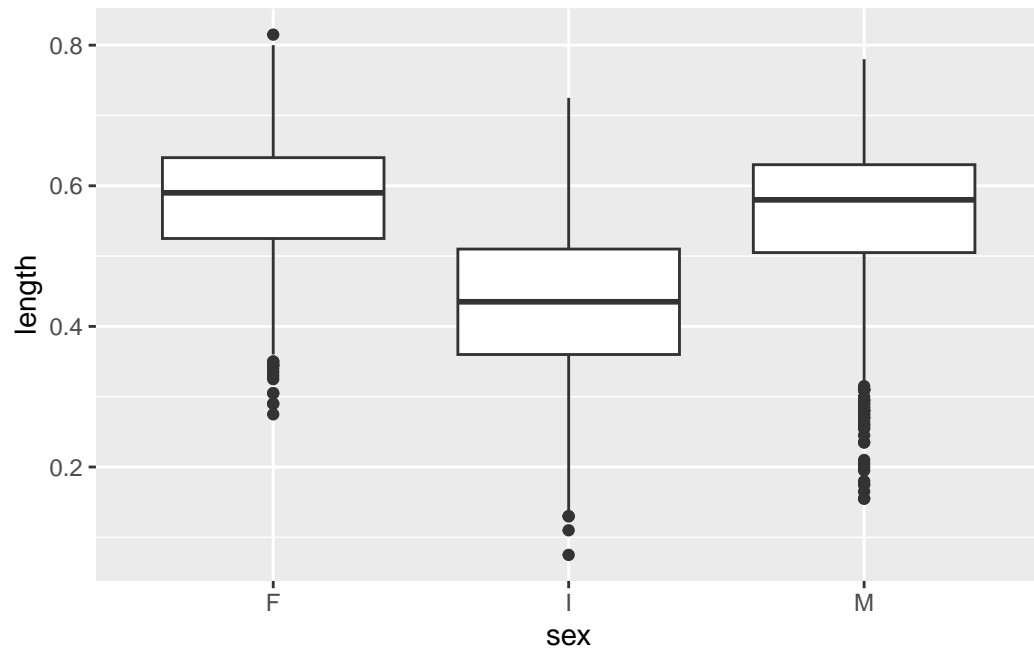
¹You can use the `facet_wrap()` function for this. Have a look at its documentation using the help console in R



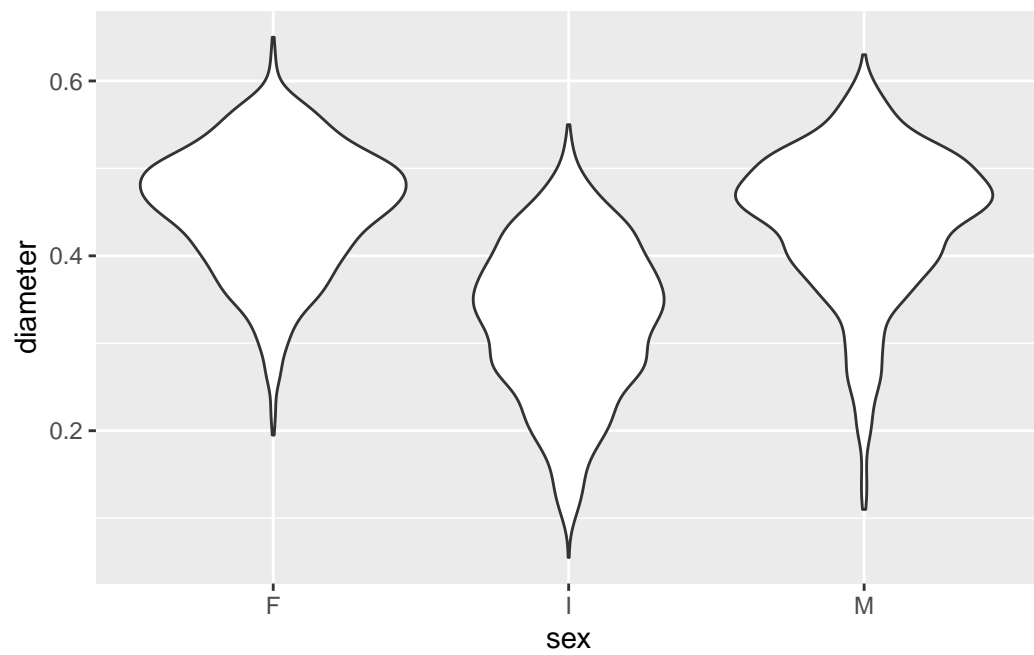
1.4 (5 points)

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

```
plt2 <- ggplot(df, aes(x = sex, y = length)) + geom_boxplot()
plt2
```



```
plt3 <- ggplot(df, aes(x = sex, y = diameter)) + geom_violin()  
plt3
```

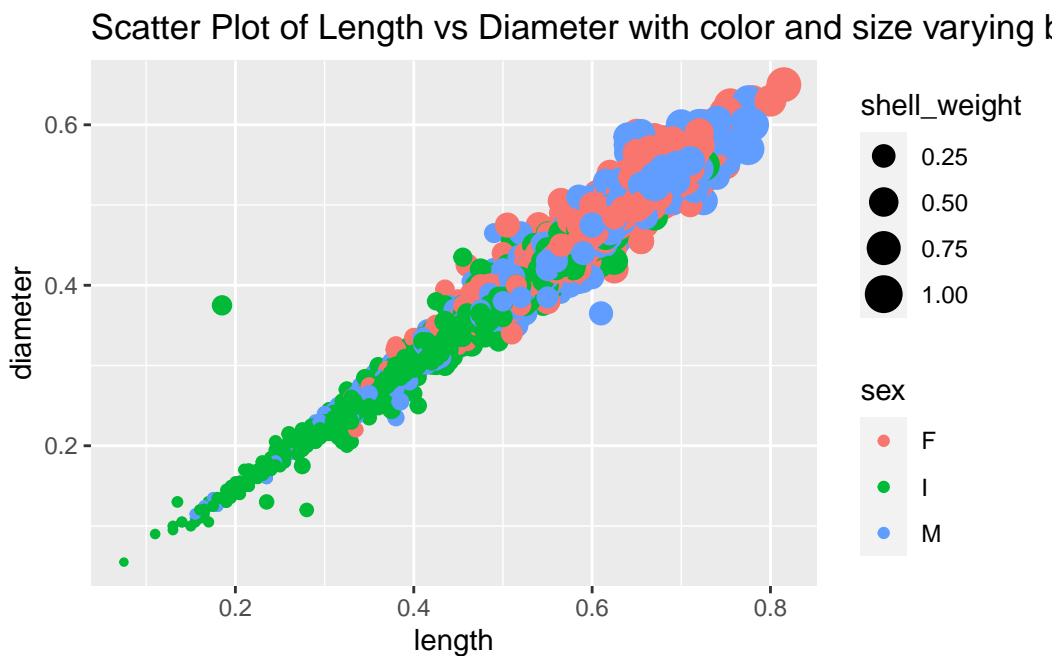


In the box plot, we can see that the median length of the abalones is similar for all three sex categories. In the violin plot, we can see that the median diameter is similar for all three sex categories and the distribution of diameter is slightly wider for the *I* category as compared to the *M* and *F* categories. Hence, we can see that there are some differences in the physical appearances of the abalones based on sex, but there aren't any substantial differences.

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?

```
plt4 <- ggplot(df, aes(x = length, y = diameter, color = sex, size = shell_weight)) + geom_point()
plt4
```



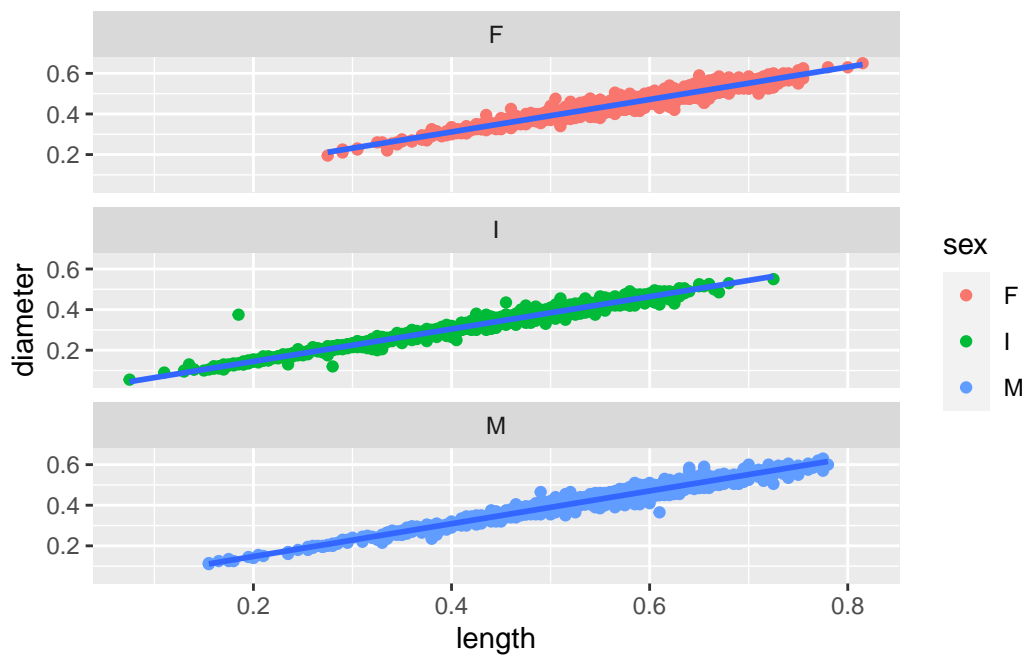
The plot does not show any notable anomalies in the data.

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

```
plt5 <- ggplot(df, aes(x = length, y = diameter)) +  
  geom_point(aes(color = sex)) +  
  geom_smooth(method = "lm", se = FALSE) +  
  facet_wrap(~sex, ncol = 1)  
plt5
```

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



²Plot example for 1.6

Question 2

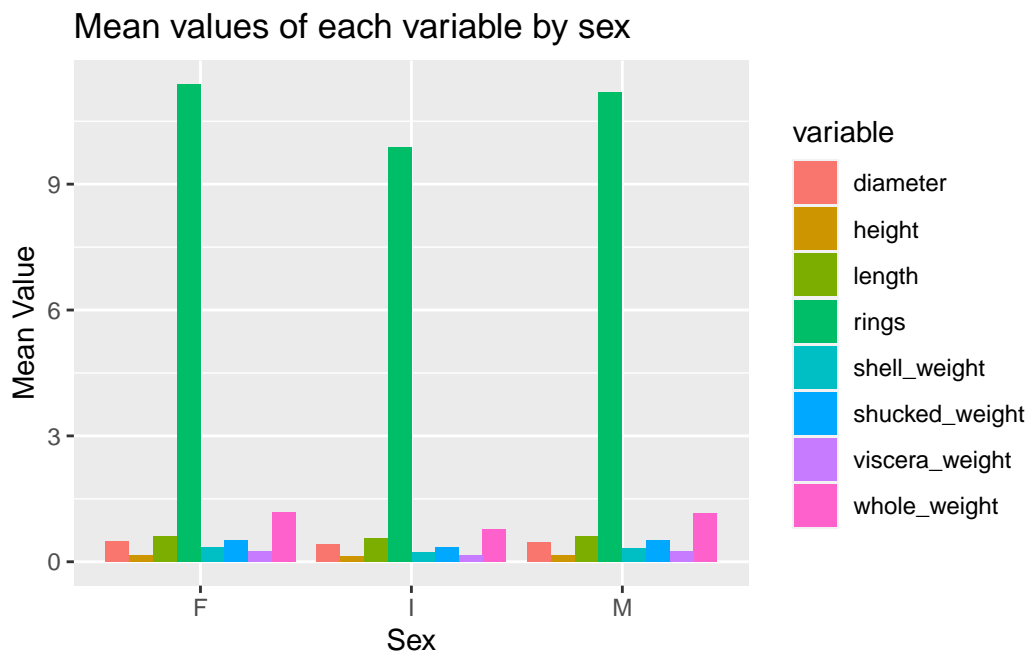
💡 40 points

More advanced analyses using `dplyr`, `purrr` 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`.

```
df %>% filter(length >= 0.5) %>%  
  group_by(sex) %>%  
  summarise_all(mean) %>%  
  gather(key = "variable", value = "mean_value", -sex) %>%  
  ggplot(aes(x = sex, y = mean_value, fill = variable)) +  
  geom_col(position = "dodge") +  
  labs(x = "Sex", y = "Mean Value") +  
  ggtitle("Mean values of each variable by sex")
```



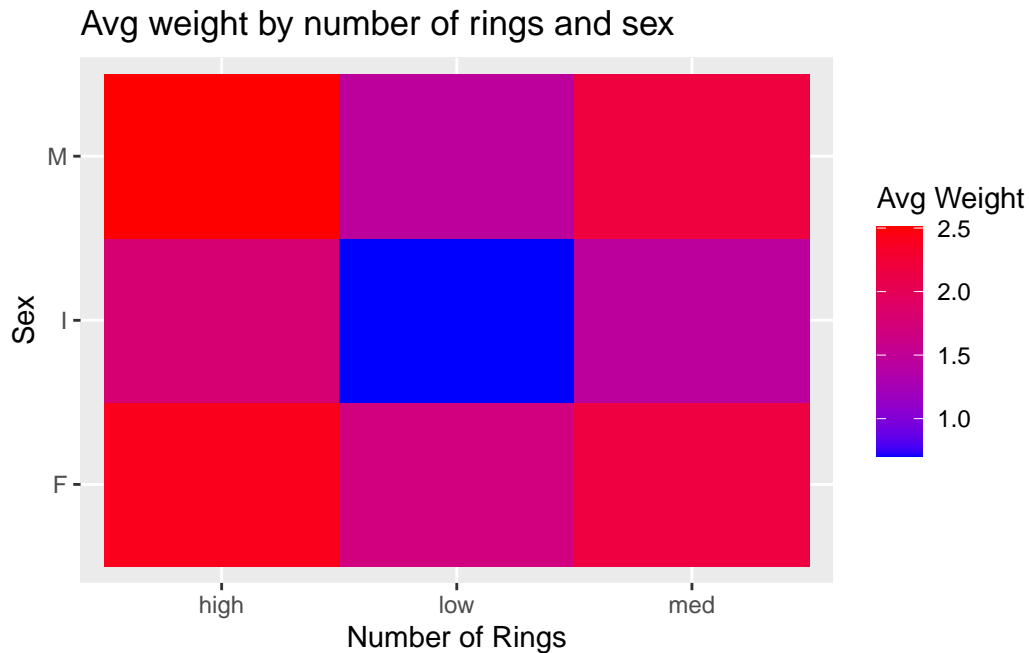
2.2 (15 points)

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.

```
df %>% mutate(num_rings = ifelse(
  rings < 10, "low", ifelse(
    rings > 20, "high", "med")
  )
) %>%
  group_by(num_rings, sex) %>%
  summarise(avg_weight = mean(whole_weight + shucked_weight + viscera_weight + shell_weight))
ggplot(aes(x = num_rings, y = sex, fill = avg_weight)) + geom_tile() + labs(x = "Number of rings", y = "Sex")
```

``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

```
library(broom)

df %>% select_if(is.numeric) %>%
  cor() %>%
  round(2) %>%
  as.data.frame()
```

	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

³Table 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		

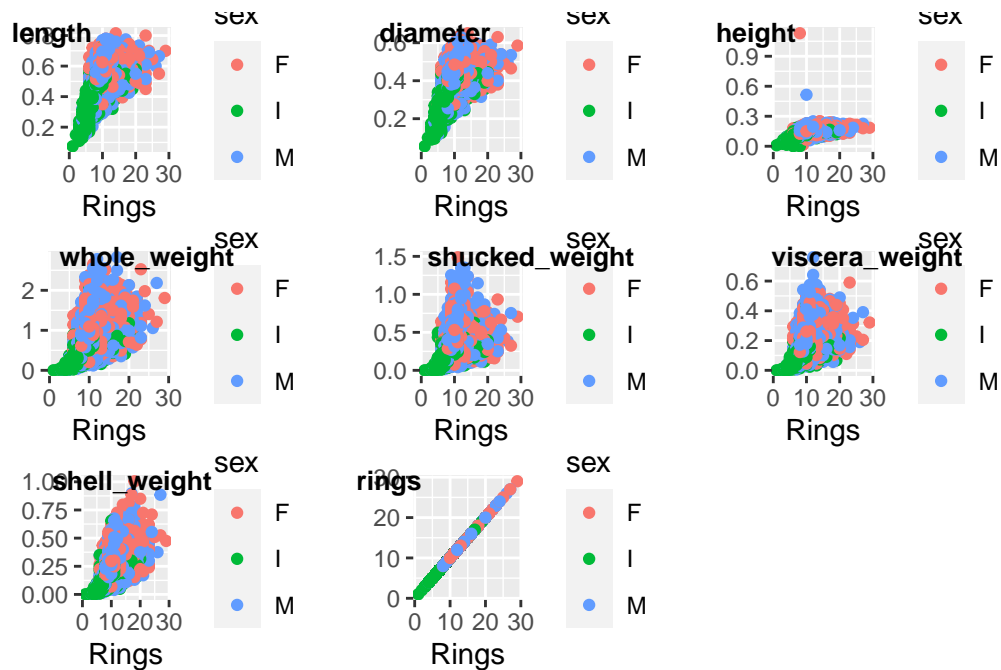
2.4 (10 points)

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.

```
df_quant <-
  df %>%
  select(!sex)

df2 <-
  df %>%
  select(rings)

plt5 <- map2(df_quant, df2, ~ ggplot(df) +
  geom_point(aes(x = rings, y = .x, color = sex)) +
  labs(x = "Rings", y = " "))
cowplot::plot_grid(plotlist = plt5, labels = colnames(df_quant), ncol = 3, label_size = 9.
```



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

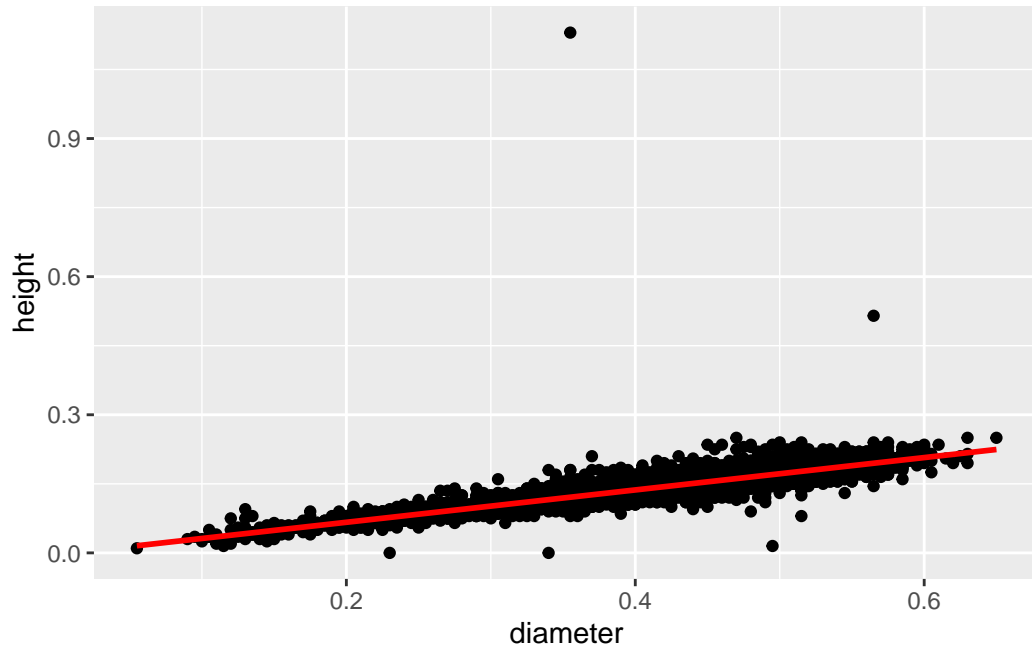
The intercept of the model is 0.4899 and the coefficient for the diameter variable is 7.8814. The p -value for the diameter variable is $2.2e-16$ which is extremely small. Based on the coefficients and significance values, we can infer that the diameter of an abalone has a positive and significant effect on its height. For every increase in 1 cm of diameter, the height of an abalone increases by approximately 7.88cm.

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.

```
plt6 <- ggplot(df, aes(x = diameter, y = height)) + geom_point() + geom_smooth(method = "lm")
plt6
```

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



The linear model is an appropriate fit for the relationship between height and diameter. The data points are around the regression line, and hence, indicates a strong linear relationship between the two variables.

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
```

```
)  
  
# To predict the new heights  
new_heights <- predict(model, newdata = data.frame(diameter = new_diameters))  
  
# Plotting new observations and their predicted heights along with the original scatterplot
```


Appendix

Session Information

Print your R session information using the following command

```
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] broom_1.0.3 cowplot_1.1.1 purrr_1.0.1 dplyr_1.1.0 ggplot2_3.4.1

[6] tidyr_1.3.0 readr_2.1.4

loaded via a namespace (and not attached):

[1] pillar_1.8.1	compiler_4.2.2	tools_4.2.2	bit_4.0.5
[5] digest_0.6.31	lattice_0.20-45	nlme_3.1-160	jsonlite_1.8.4
[9] evaluate_0.20	lifecycle_1.0.3	tibble_3.1.8	gtable_0.3.1
[13] mgcv_1.8-41	pkgconfig_2.0.3	rlang_1.0.6	Matrix_1.5-1
[17] cli_3.6.0	parallel_4.2.2	yaml_2.3.7	xfun_0.37
[21] fastmap_1.1.0	withr_2.5.0	knitr_1.42	generics_0.1.3
[25] vctrs_0.5.2	hms_1.1.2	bit64_4.0.5	grid_4.2.2
[29] tidyselect_1.2.0	glue_1.6.2	R6_2.5.1	fansi_1.0.4
[33] vroom_1.6.1	rmarkdown_2.20	farver_2.1.1	tzdb_0.3.0
[37] magrittr_2.0.3	backports_1.4.1	splines_4.2.2	scales_1.2.1
[41] ellipsis_0.3.2	htmltools_0.5.4	colorspace_2.1-0	renv_0.16.0-53

```
[45] labeling_0.4.2    utf8_1.2.3          munsell_0.5.0       crayon_1.5.2
```