

# Homework 2

Miranda Goodman {style='background-color: yellow;'}

## Table of contents

Question 1	2
Question 2	8
Question 3	12

<b>Appendix</b>	<b>17</b>
-----------------	-----------

[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)
```

Warning: package 'ggplot2' was built under R version 4.1.3

```
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)
```

Warning: package 'cowplot' was built under R version 4.1.3

---

## 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.

---

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

[1] 0

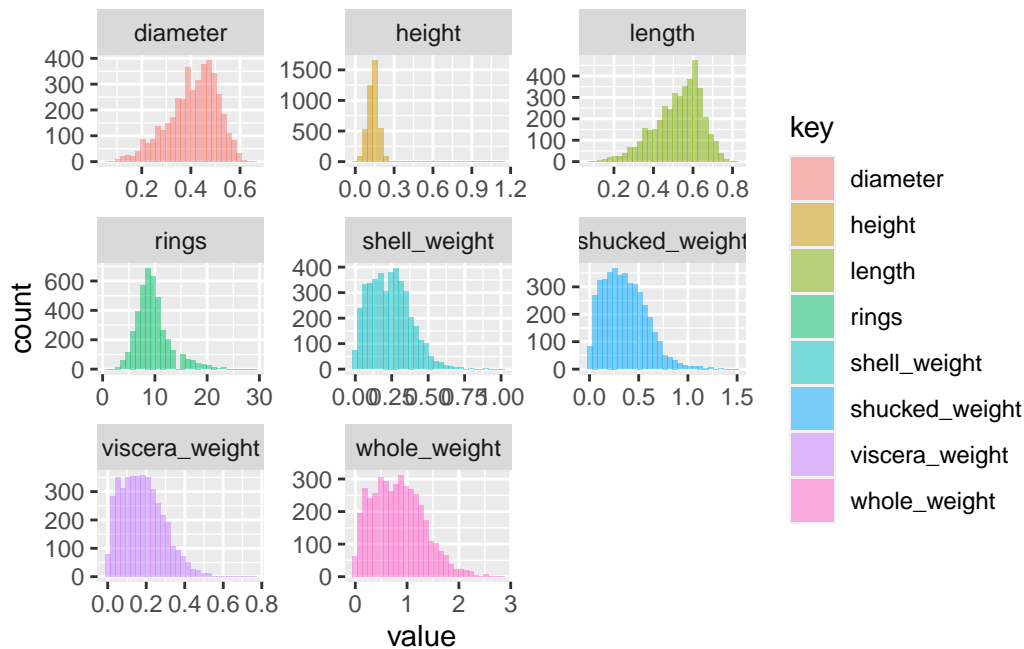
0 were dropped.

---

### 1.3 (5 points)

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

```
library(ggplot2)
abalone %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(x=value, fill=key)) +
  geom_histogram(alpha=0.5, position="identity", bins = 30) +
  facet_wrap(~ key, scales="free")
```



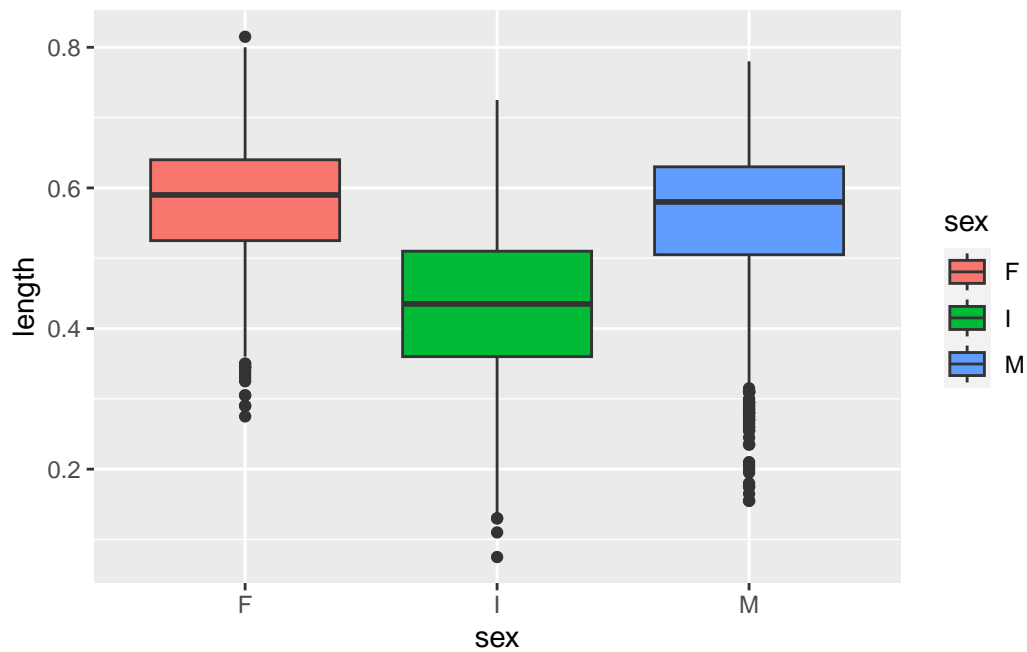
### 1.4 (5 points)

Create a boxplot 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?

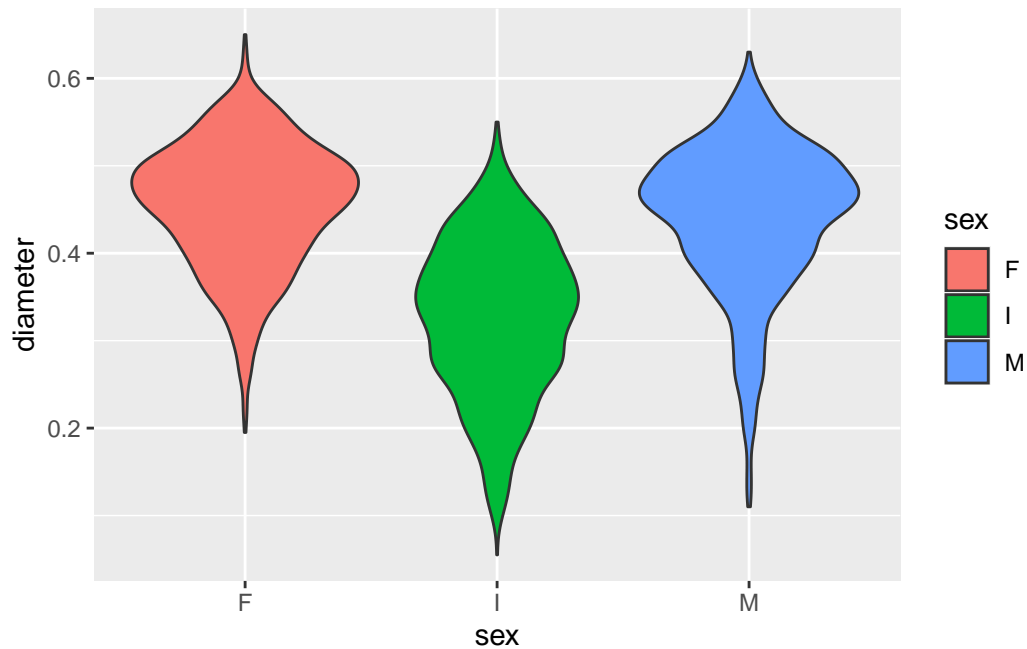
---

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

```
abalone %>%
  select(sex, length) %>%
  ggplot(aes(x = sex, y = length, fill = sex)) +
  geom_boxplot()
```



```
abalone %>%
  select(sex, diameter) %>%
  ggplot(aes(x = sex, y = diameter, fill = sex)) +
  geom_violin()
```

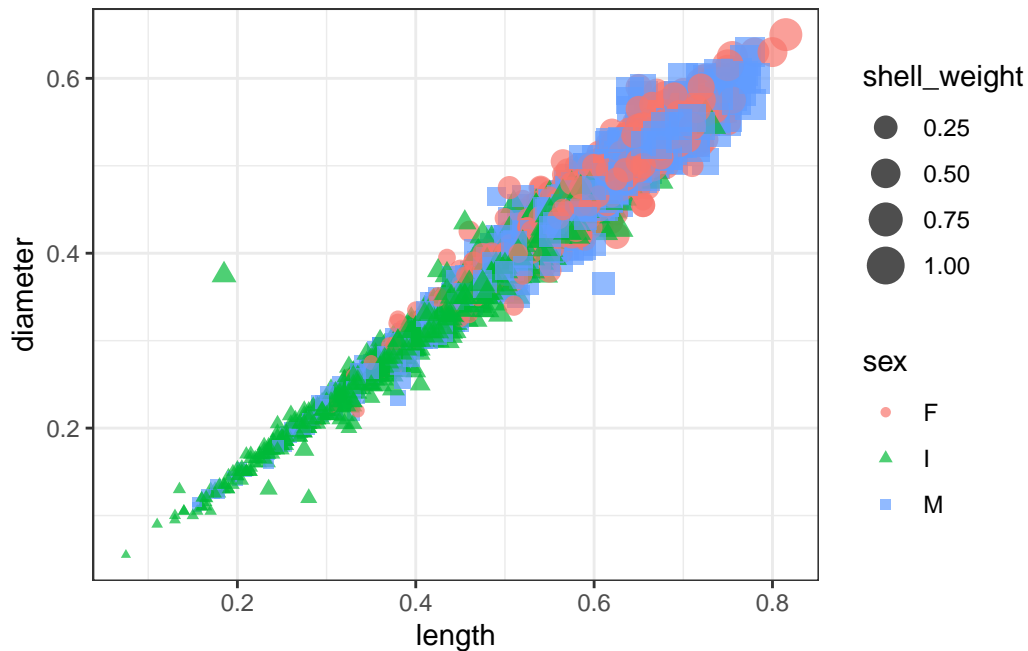


The infant abalone is shorter and narrower than the female and male abalones.

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?

```
abalone %>%
  ggplot(aes(x = length, y = diameter, color = sex, shape = sex, size = shell_weight)) +
    geom_point(alpha = 0.7) +
    theme_bw()
```



Yes, there is one imphant point that is an outlier compared to the rest of the data. It is at approximately  $x = 1.8$ .

1.6 (5 points)

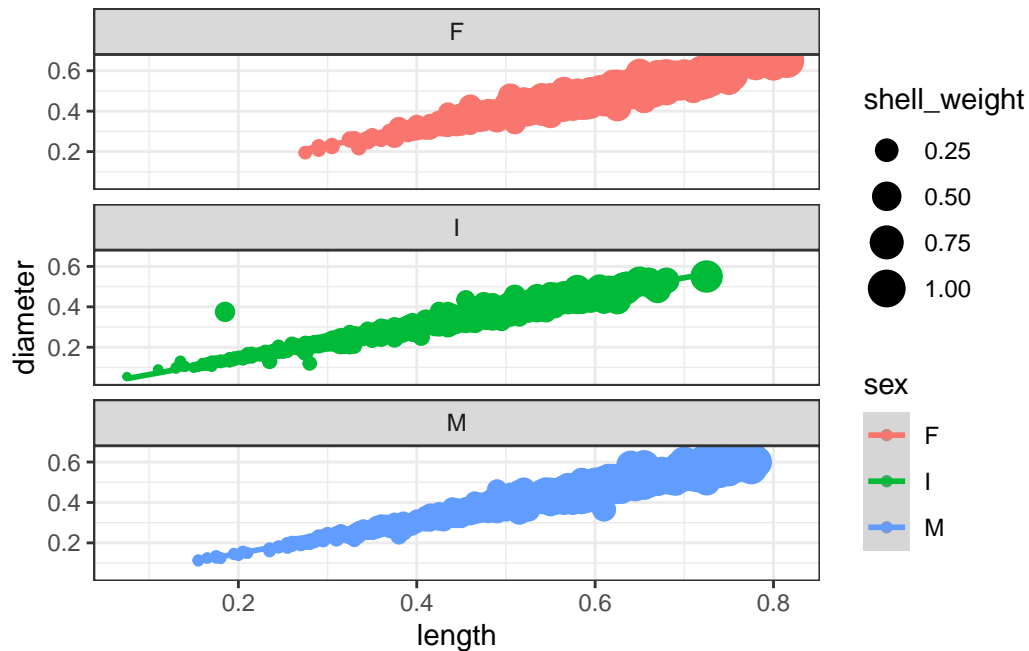
For each **sex**, create separate scatter plots of **length** and **diameter**. For each plot, also add a **linear** trendline to illustrate the relationship between the variables. Use the `facet_wrap()` function in R for this, and ensure that the plots are vertically stacked **not** horizontally. You should end up with a plot that looks like this: <sup>2</sup>

```
df %>%
  ggplot(aes(x = length, y = diameter, color = sex)) +
  geom_point(aes(size = shell_weight)) +
  geom_smooth(method = "lm") +
  facet_wrap(~ sex, ncol = 1) +
  theme_bw()
```

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

---

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



## Question 2

💡 40 points

More advanced analyses using `dplyr`, `purrr` and `ggplot2`

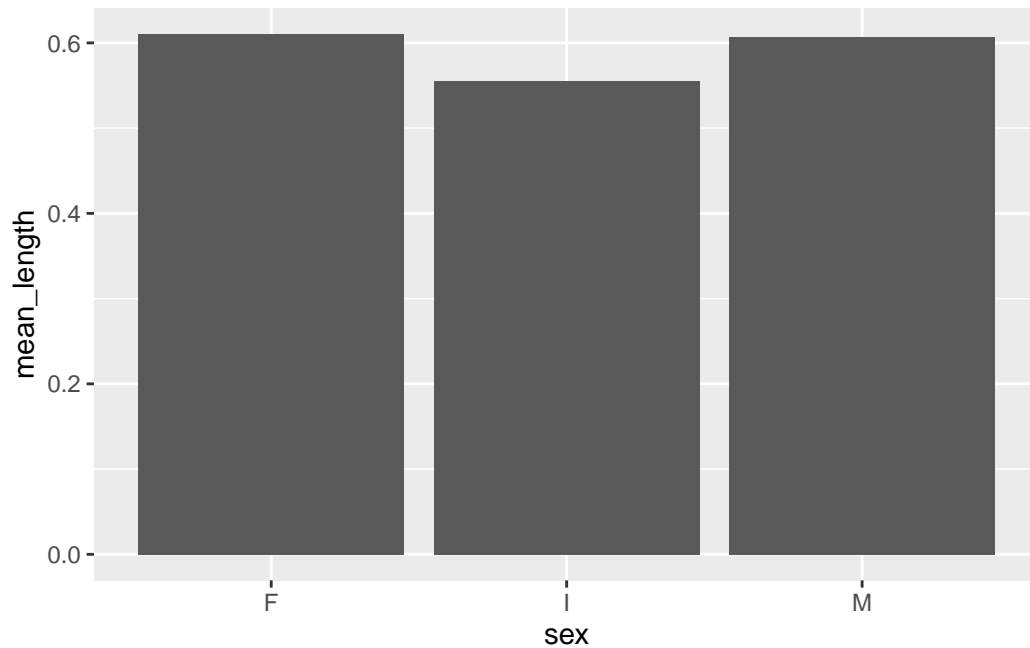
2.1 (10 points)

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(mean_length = mean(length)) %>%
  ggplot(aes(x = sex, y = mean_length)) +
```



```
geom_bar(stat = "identity")
```



2.2 (15 points)

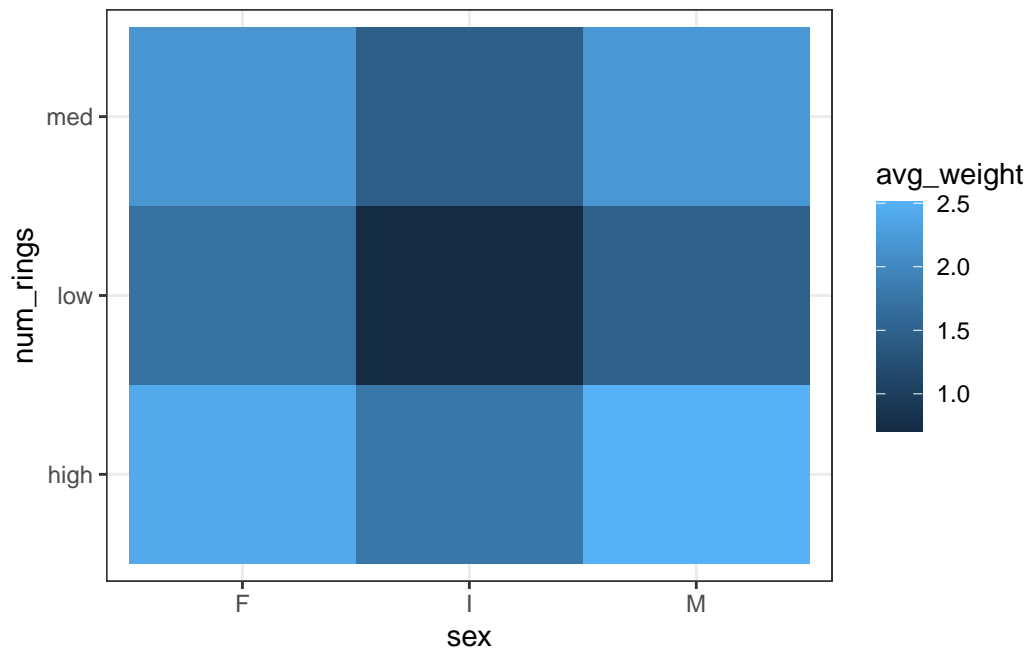
Implement the following in a **single command**:

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) %>%
  summarize(avg_weight = mean(whole_weight + shucked_weight + viscera_weight + shell_weight))
ggplot(aes(x = sex, y = num_rings, fill = avg_weight)) +
  geom_tile() +
  theme_bw()
```

`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>

```
df %>%
  select_if(is.numeric) %>%
  cor() %>%
```

<sup>3</sup>Table for 2.3

```
round(2)
```

	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

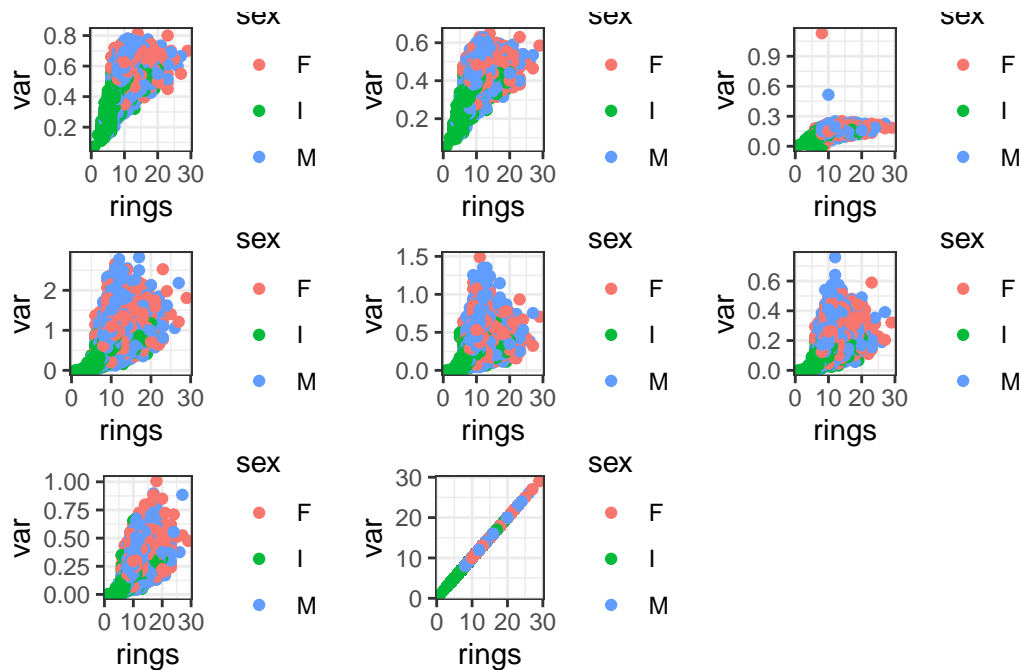
---

## 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.

```
data_prep <- df %>% select_if (is.numeric)
s_p <- map2(data_prep, names(data_prep), function(var, data) {
  ggplot(df, aes(x = rings, y = var, color = sex)) +
    geom_point() +
    theme_bw()
})

plot_grid(plotlist = s_p)
```



### 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.

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

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

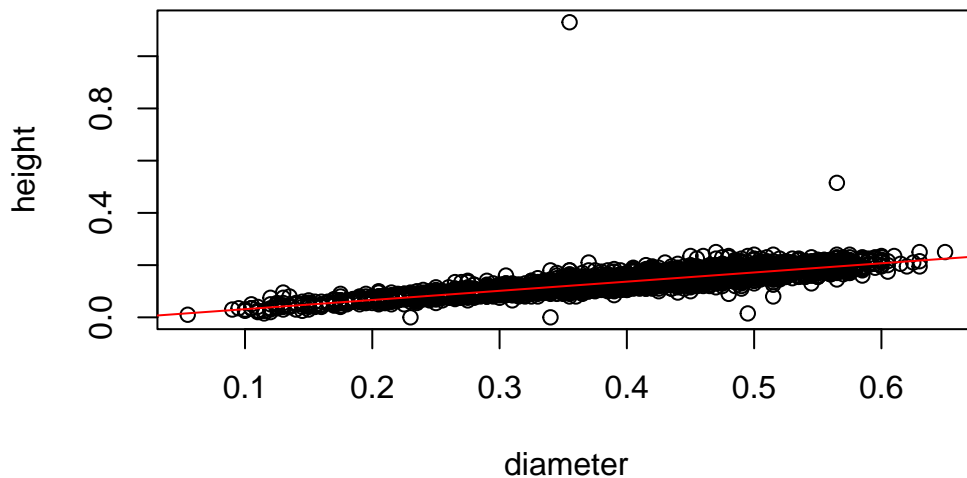
The intercept represents the expected outcome of height when diameter = 0. This is  $\beta_0$ , which in this case  $\beta_0 = -0.003803$ . The  $\beta_1$  or slope coefficient of .351376 means that for every 1 unit the diameter increases, on average the height will increase by .351376. The standard error means the variability of the coefficients. Both height and diameter are significant in the model. The p-value for height is 0.0119, which is less than 0.05 and the p-value for the diameter is less than 2e-16. This indicates that both variables are significant. The R-squared value indicates the model accounts for 69.5% of the variability in the data.

---

### 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(height ~ diameter, df)
abline(lm(height ~ diameter, df), col = "red")
```



The model is a good fit for the data as most of the points follow a linear pattern. In this case it is positively correlated and linear. However, there are two points that are obvious outliers from the rest of 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"`

I expect their height to be around 0.1-0.3

```
new_diameters <- c(
  0.15218946,
  0.48361548,
  0.58095513,
  0.07603687,
  0.50234599,
  0.83462092,
  0.95681938,
  0.92906875,
```

```

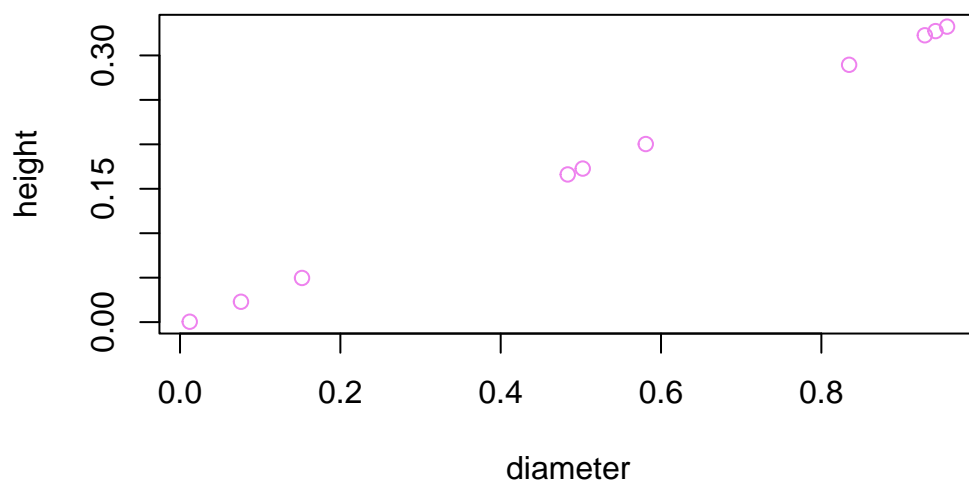
0.94245437,
0.01209518
)

```

```

ndf <- data.frame(diameter = new_diameters, height = predict(lrm, data.frame(diameter = ne
plot(ndf, col = "violet")

```

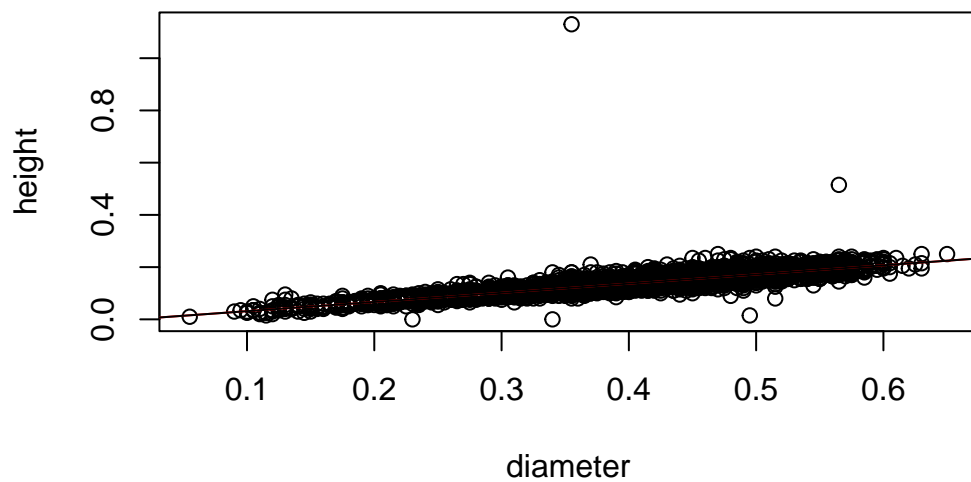


```

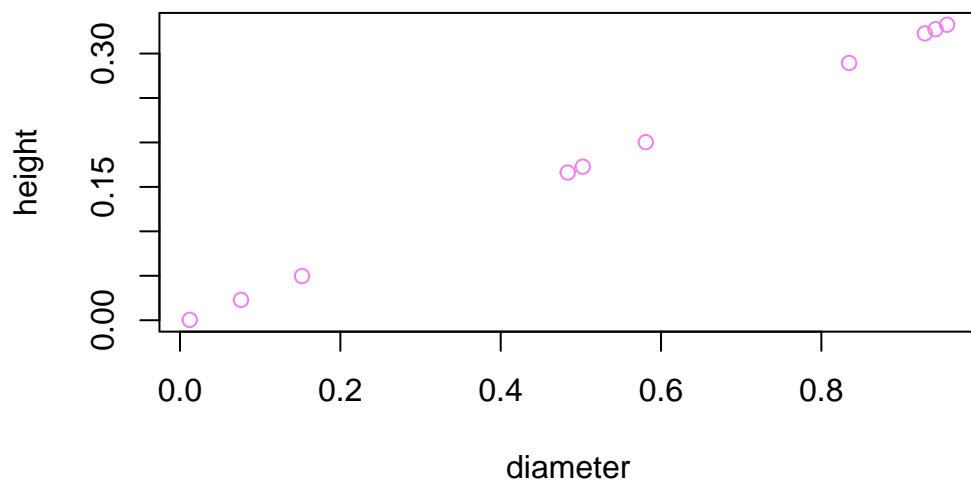
plot(height ~ diameter, df)
abline(lm(height ~ diameter, df), col = "red")
abline(lm(height ~ diameter, ndf, col = "violet"))

```

Warning: In `lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...)` :  
extra argument 'col' will be disregarded



```
plot(ndf, col = "violet")
```





---

## Appendix

### Session Information

Print your R session information using the following command

```
sessionInfo()
```

```
R version 4.1.2 (2021-11-01)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22621)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

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

other attached packages:
[1] cowplot_1.1.1 purrr_0.3.4  dplyr_1.0.8  ggplot2_3.4.1 tidyr_1.2.0
[6] readr_2.1.2

loaded via a namespace (and not attached):
 [1] pillar_1.8.1    compiler_4.1.2  tools_4.1.2    bit_4.0.4
 [5] digest_0.6.31   lattice_0.20-45 nlme_3.1-162    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_1.5-3
[17] cli_3.6.0        rstudioapi_0.14 curl_5.0.0      parallel_4.1.2
[21] yaml_2.3.7       xfun_0.37       fastmap_1.1.0   withr_2.5.0
[25] knitr_1.42       generics_0.1.2  vctrs_0.5.2     hms_1.1.1
[29] bit64_4.0.5      grid_4.1.2      tidyselect_1.1.1 glue_1.6.2
[33] R6_2.5.1         fansi_1.0.4     vroom_1.5.7     rmarkdown_2.20
```

```
[37] farver_2.1.1      tzdb_0.2.0        magrittr_2.0.3    splines_4.1.2
[41] scales_1.2.1      ellipsis_0.3.2    htmltools_0.5.4   colorspace_2.1-0
[45] renv_0.16.0-53    labeling_0.4.2     utf8_1.2.3        munsell_0.5.0
[49] crayon_1.5.0
```