

Homework 2

Brian MacCurtin

Table of contents

Question 1	2
Question 2	8
Question 3	12

Appendix	16
-----------------	-----------

[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:

```
rm(list = ls())  
library(readr)
```

```
library(tidyr)
library(ggplot2)
library(dplyr)
library(purrr)
library(cowplot)
library(tidyverse)
```

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)
colnames(abalone) <- abalone_col_names
head(abalone)
```

	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395
5	I	0.425	0.300	0.095	0.3515	0.1410	0.0775
6	F	0.530	0.415	0.150	0.7775	0.2370	0.1415

	shell_weight	rings
1	0.070	7
2	0.210	9
3	0.155	10
4	0.055	7
5	0.120	8
6	0.330	20

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?

```
#Originally 4176 observations
df <- na.omit(abalone)
head(df)
```

	sex	length	diameter	height	whole_weight	shucked_weight	viscera_weight
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395
5	I	0.425	0.300	0.095	0.3515	0.1410	0.0775
6	F	0.530	0.415	0.150	0.7775	0.2370	0.1415

	shell_weight	rings
1	0.070	7
2	0.210	9
3	0.155	10
4	0.055	7
5	0.120	8
6	0.330	20

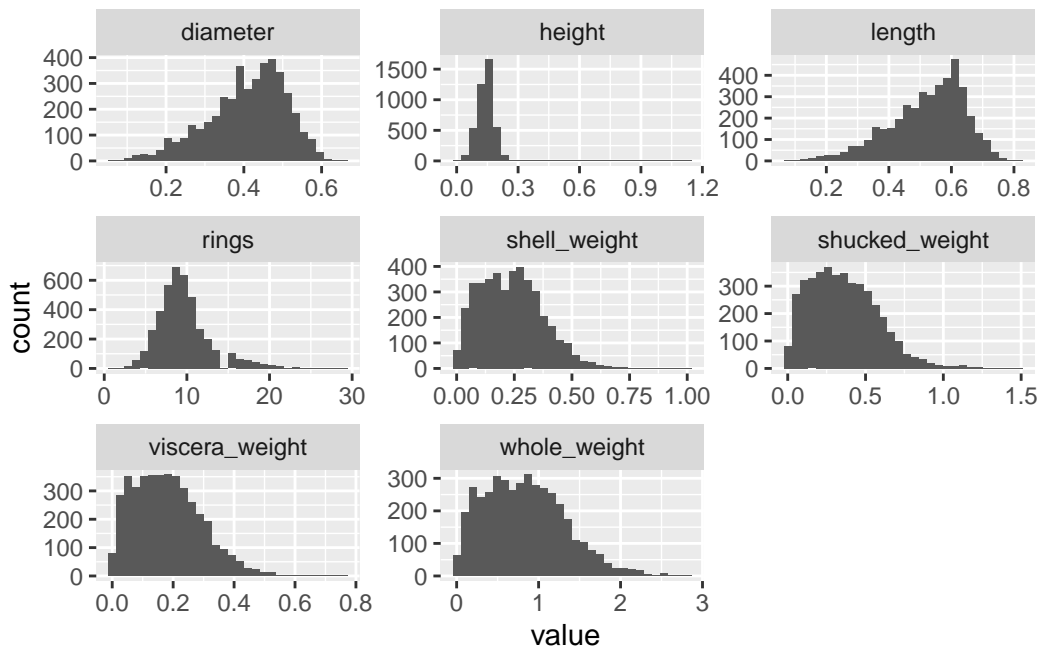
0 rows were dropped. Still have 4176 observations

1.3 (5 points)

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

```
df %>%  
  select(length, diameter, height, whole_weight, shucked_weight, viscera_weight, shell_weight)  
  gather() %>%  
  ggplot(aes(value)) +  
  facet_wrap(~ key, scales = "free") +  
  geom_histogram()
```

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

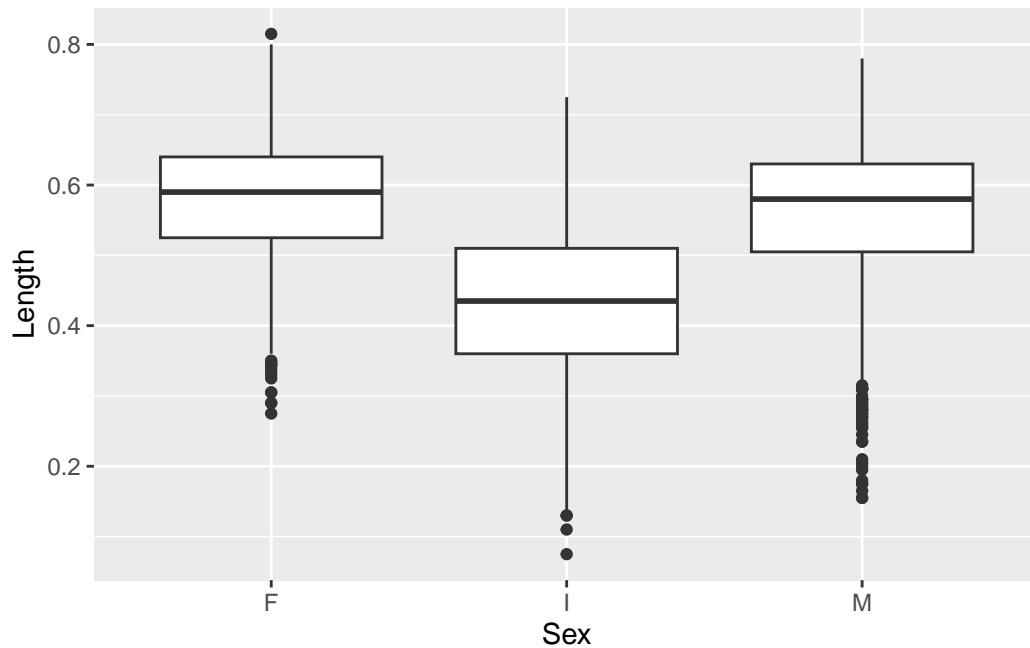


¹You can use the `facet_wrap()` function for this. Have a look at its documentation using the help console in R

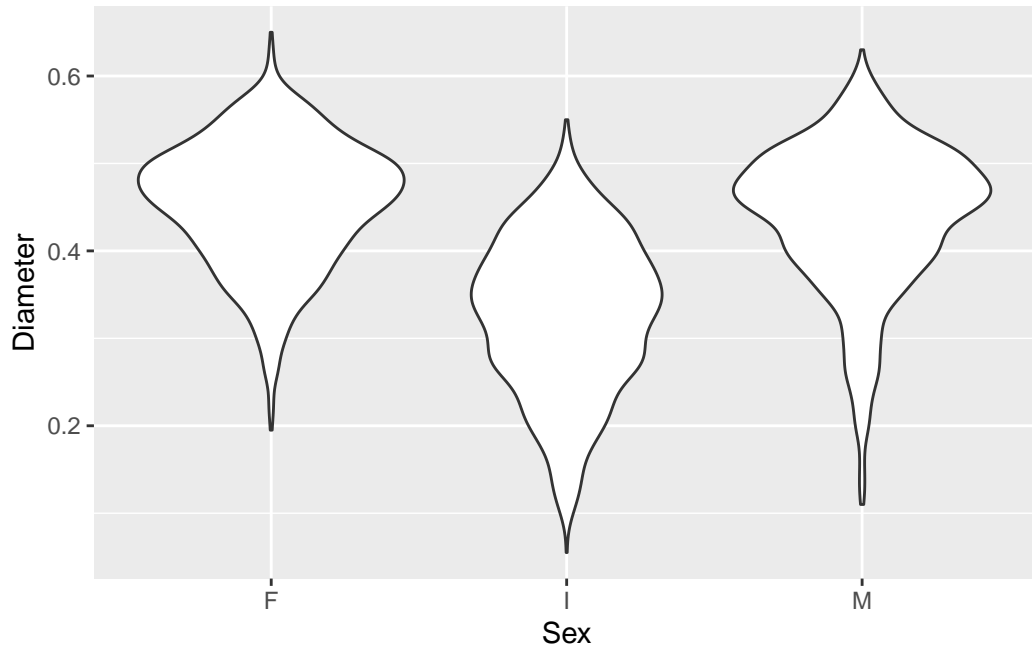
1.4 (5 points)

Create a 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?

```
ggplot(data = df, mapping = aes(x=sex, y=length)) +  
  geom_boxplot() +  
  labs(x = "Sex", y = "Length")
```



```
ggplot(data = df, mapping = aes(x=sex, y=diameter)) +  
  geom_violin() +  
  labs(x = "Sex", y = "Diameter")
```

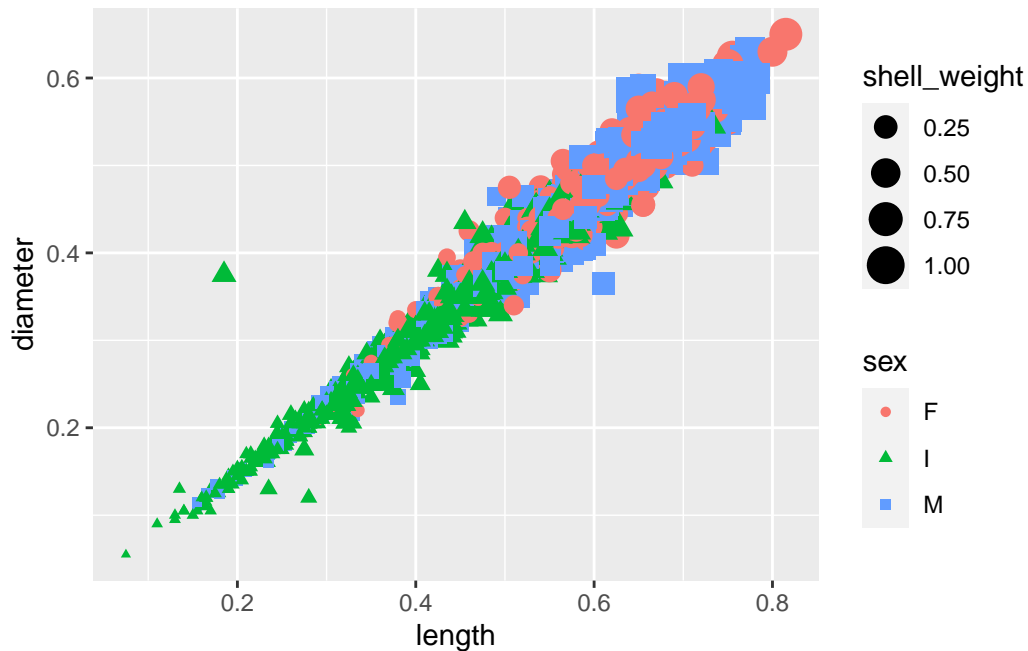


We can clearly see that the lengths and diameters of infants are less than non infants. However, between males and females, there is no obvious differences in the lengths and diameters of the abalone. They have similar means and similar spreads

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?

```
ggplot(data = df, mapping = aes(x=length, y=diameter, color = sex, shape= sex, size = shell_wight)) +  
  geom_point()
```



There is one infant who has about double the diameter of other abalones at the same length. Because of this, it also has a much bigger shell weight than other infant abalones at the same length. Otherwise, we can see that as the length of the abalone grows, the diameter and shell weight grow as well.

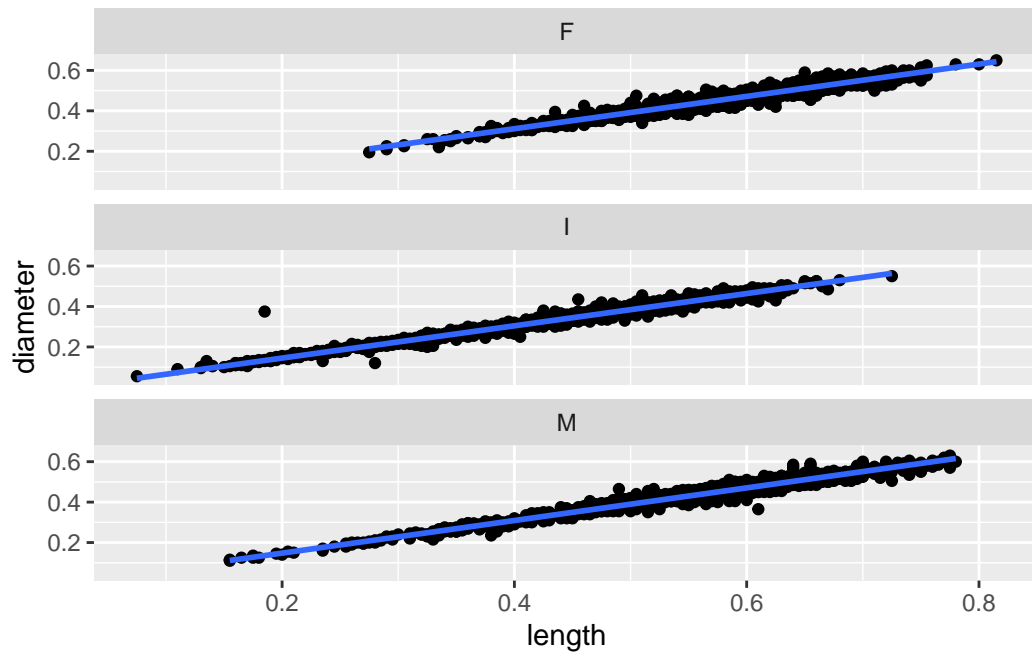
1.6 (5 points)

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

```
ggplot(data = df, mapping = aes(x=length, y=diameter)) +
  geom_point() +
  geom_smooth(method = lm) +
  facet_wrap(~sex, dir = "v")
```

``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`.

```
df4 <-
  df %>%
  filter(length >= .5) %>%
  group_by(sex) %>%
  summarize(meanlength = mean(length),
             meandiameter = mean(diameter),
```

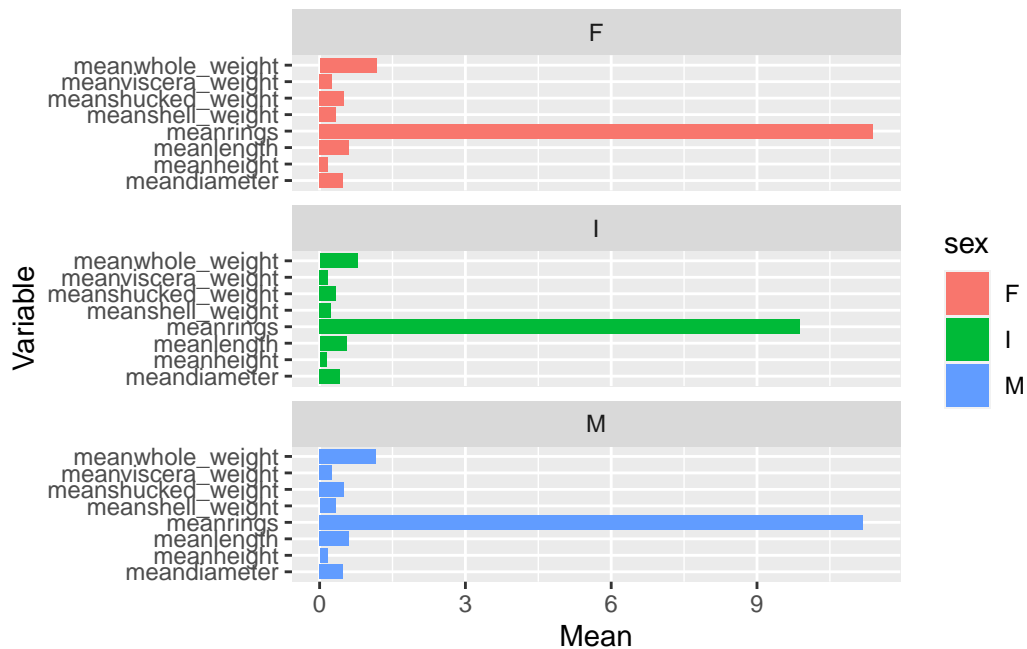


```

    meanheight = mean(height),
    meanwhole_weight = mean(whole_weight),
    meanshucked_weight = mean(shucked_weight),
    meanviscera_weight = mean(viscera_weight),
    meanshell_weight = mean(shell_weight),
    meanrings = mean(rings)) %>%
pivot_longer(!sex,
              values_to = "Mean",
              names_to = "Variable")

ggplot(data=df4, aes(x=Mean, y=Variable)) +
  geom_bar(stat="identity", aes(fill=sex)) +
  facet_wrap(vars(sex), dir="v")

```



2.2 (15 points)

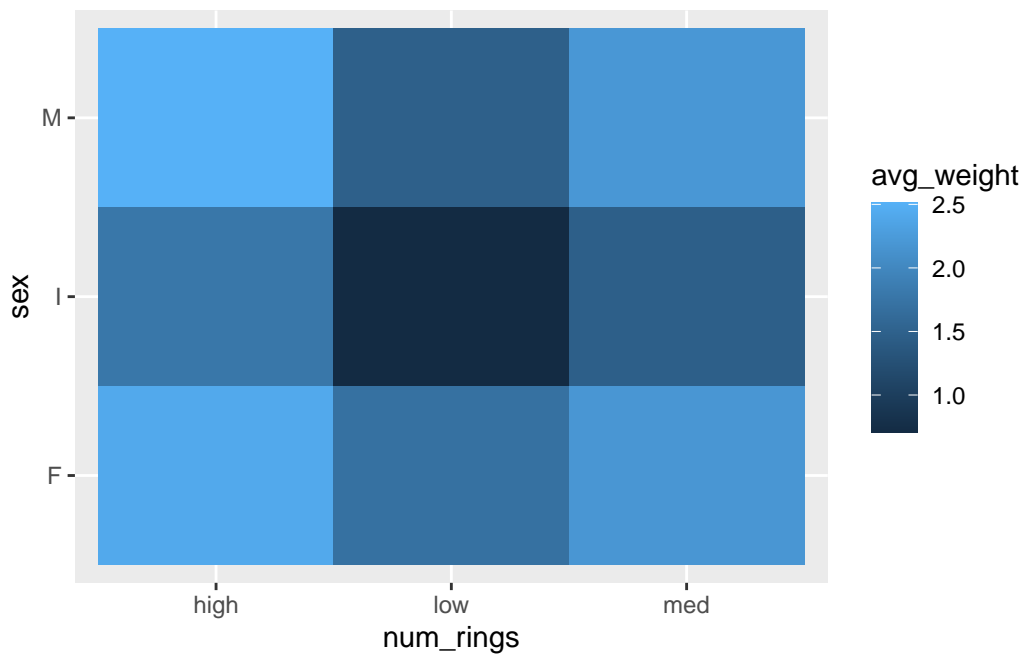
Implement the following in a **single command**:

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

`summarise()` has grouped output by 'sex'. 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

```
df3 <-
  df %>%
    select(length, diameter, height, whole_weight, shucked_weight, viscera_weight, shell_weight)

table <- round(cor(df3), 2)
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
shell_weight	0.90	0.91	0.82	0.96	0.88
rings	0.56	0.58	0.56	0.54	0.42
	viscera_weight	shell_weight	rings		
length	0.90		0.90	0.56	
diameter	0.90		0.91	0.58	
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	

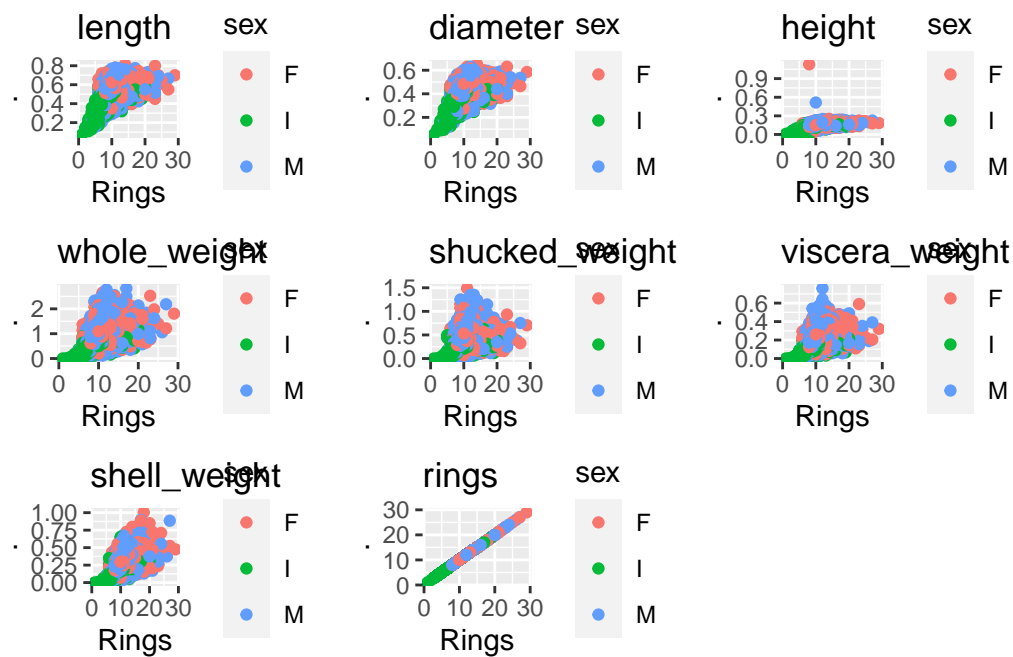
2.4 (10 points)

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.

³Table for 2.3

```
quant <-
  abalone %>%
  select(length, diameter, height, whole_weight, shucked_weight, viscera_weight, shell_weight) %>%
  map2(., names(.), ~ ggplot(abalone, aes(x = rings, y = ., color = sex)) +
    geom_point() +
    ggtitle(.y) +
    xlab("Rings"))

cowplot::plot_grid(plotlist = quant, 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	-0.01044	-0.00148	0.00852	1.00906

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.003784	0.001512	-2.502	0.0124 *
diameter	0.351346	0.003602	97.540	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0231 on 4174 degrees of freedom

Multiple R-squared: 0.6951, Adjusted R-squared: 0.695

F-statistic: 9514 on 1 and 4174 DF, p-value: < 2.2e-16

The y-intercept, β_0 - The mean height of a hypothetical abalone is -.0037 mm when the diameter is 0 mm

The slope, β_1 - For each additional mm in diameter, we expect the height of an abalone to increase by .3513 mm

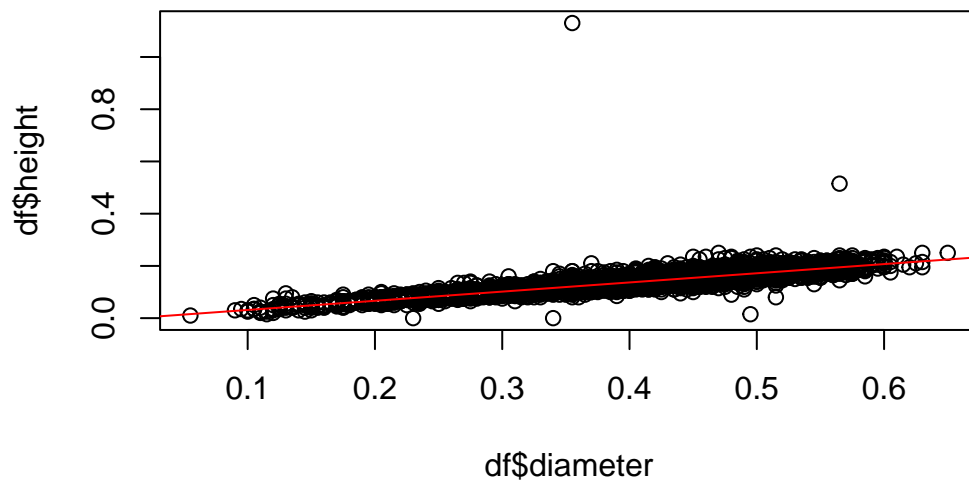
Since the p-value for the intercept is below the level of significance, this suggests that the intercept term is statistically different than zero

Since the p-value of diameter is below the level of significance, this suggests that diameter is a useful predictor when predicting height since the slope is statistically different than zero

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(df$diameter, df$height)
abline(model, col="red")
```



Yes a linear model seems appropriate. The regression line follows the trend of the data really well and there is no curvature to the points in the plot

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
)
```

```

predicted_height <- predict(model, data.frame(diameter = new_diameters))
predicted_height

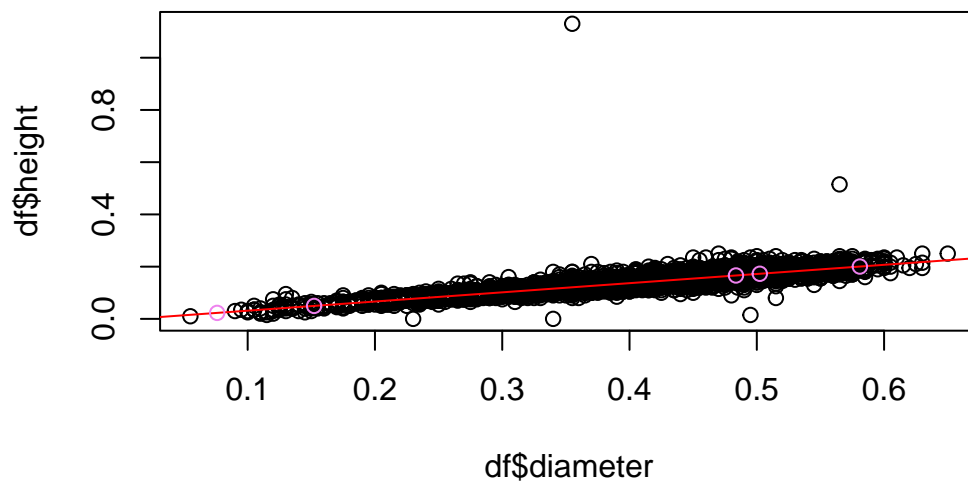
```

	1	2	3	4	5	6
	0.0496872736	0.1661323358	0.2003321901	0.0229313989	0.1727132174	0.2894565404
	7	8	9	10		
	0.3323904273	0.3226403666	0.3273433448	0.0004657697		

```

plot(df$diameter, df$height)
abline(model, col="red")
points(new_diameters, predicted_height, col = "violet")

```



Appendix

Session Information

Print your R session information using the following command

```
sessionInfo()
```

```
R version 4.2.1 (2022-06-23)
```

```
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] forcats_0.5.2  stringr_1.5.0  tibble_3.1.8   tidyverse_1.3.2
```

```
[5] cowplot_1.1.1  purrr_1.0.1    dplyr_1.0.10   ggplot2_3.4.0
```

```
[9] tidyr_1.2.1    readr_2.1.3
```

```
loaded via a namespace (and not attached):
```

```
[1] lattice_0.20-45  lubridate_1.9.0  assertthat_0.2.1
```

```
[4] digest_0.6.31    utf8_1.2.2      R6_2.5.1
```

```
[7] cellranger_1.1.0 backports_1.4.1  reprex_2.0.2
```

```
[10] evaluate_0.20    httr_1.4.4      pillar_1.8.1
```

```
[13] rlang_1.0.6      googlesheets4_1.0.1 readxl_1.4.1
```

```
[16] rstudioapi_0.14  Matrix_1.5-1    rmarkdown_2.20
```

```
[19] labeling_0.4.2   splines_4.2.1   googledrive_2.0.0
```

```
[22] munsell_0.5.0    broom_1.0.2     compiler_4.2.1
```

```
[25] modelr_0.1.10    xfun_0.36       pkgconfig_2.0.3
```

```
[28] mgcv_1.8-41      htmltools_0.5.4 tidysselect_1.2.0
```


[31]	fansi_1.0.3	crayon_1.5.2	tzdb_0.3.0
[34]	dbplyr_2.2.1	withr_2.5.0	grid_4.2.1
[37]	nlme_3.1-160	jsonlite_1.8.4	gtable_0.3.1
[40]	lifecycle_1.0.3	DBI_1.1.3	magrittr_2.0.3
[43]	scales_1.2.1	cli_3.6.0	stringi_1.7.12
[46]	farver_2.1.1	renv_0.16.0-53	fs_1.5.2
[49]	xml2_1.3.3	ellipsis_0.3.2	generics_0.1.3
[52]	vctrs_0.5.1	tools_4.2.1	glue_1.6.2
[55]	hms_1.1.2	fastmap_1.1.0	yaml_2.3.6
[58]	timechange_0.2.0	colorspace_2.0-3	gargle_1.2.1
[61]	rvest_1.0.3	knitr_1.41	haven_2.5.1