Homework 2

Robert Chappell

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
```

```
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)
```

```
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 <- na.omit(abalone)
  droppedrows = nrow(abalone) - nrow(df)
  droppedrows
```

```
[1] 0
```

1.3 (5 points)

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

---

[1] You can use the `facet_wrap()` function for this. Have a look at its documentation using the help console in R
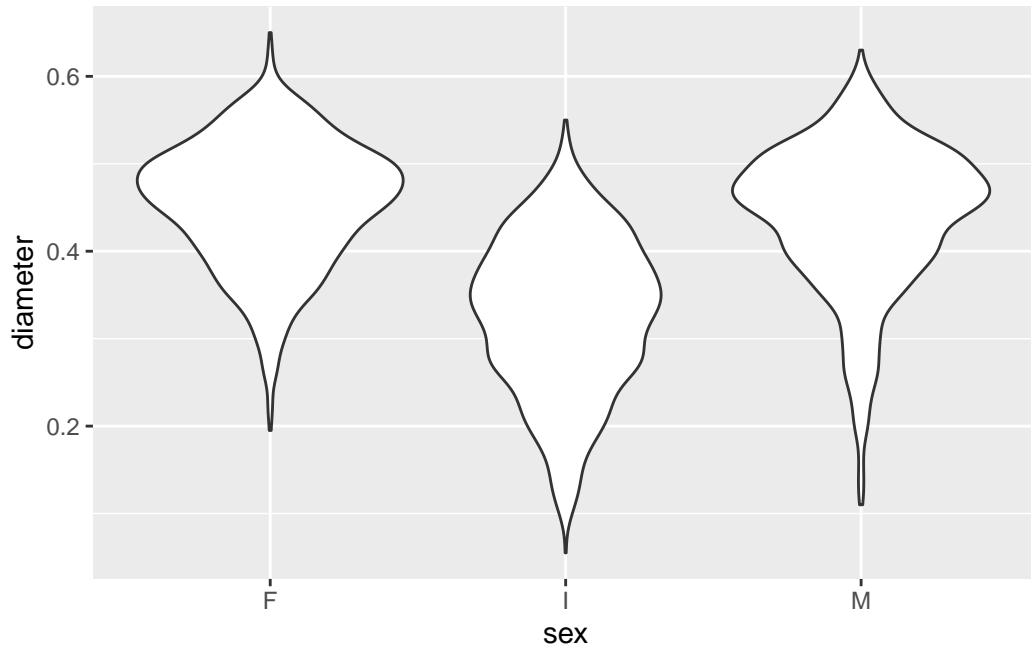
3

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?

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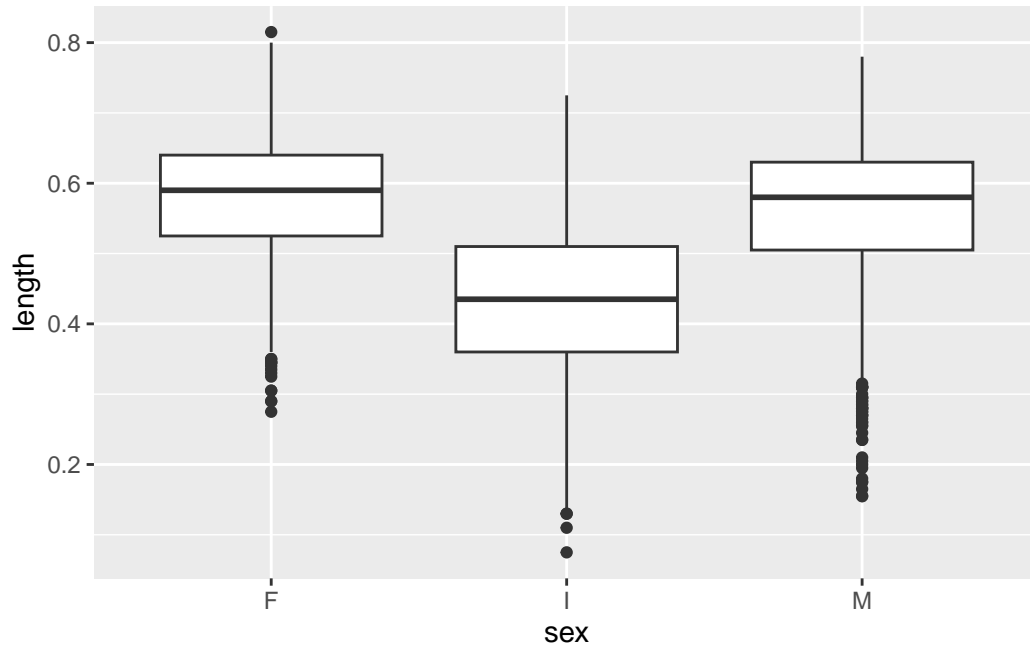


```
ggplot(df, aes(x = sex, y = diameter)) +
  geom_violin()
```

4

Based on these graphs, I do not see a noticeable difference in Male and Female abalones, for snails with sex I, there is a noticeable drop in length. Looking at the violin plots, once again the F and M sexes have very similar diameters, with I having a lower diameter than both

---

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(df, aes(x = length, y = diameter, color = sex, shape = sex, size = shell_weight)) +
  geom_point()
```
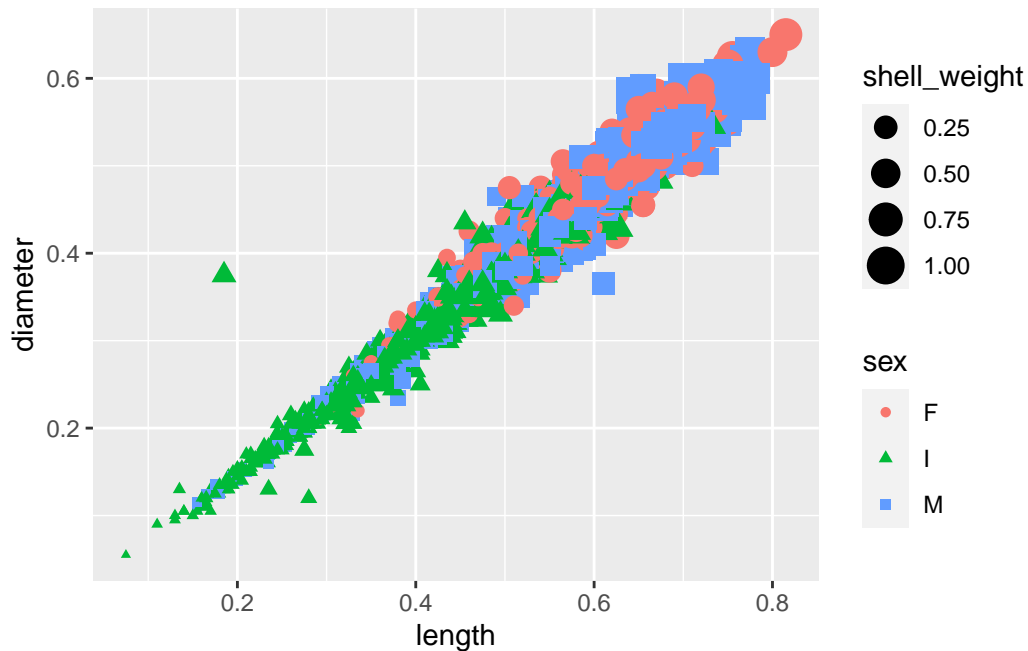
From the scatterplot above, you can see a couple points that are anomalies, being the I snail with close to 0.4 diameter and 0.2 length, there are other points that are slightly off, but not much.


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(abalone, aes(x = length, y = diameter)) +
  geom_point() +
  geom_smooth(method = "lm") +
  facet_wrap(~ sex, ncol = 1)
```

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

---

[2] Plot example for 1.6
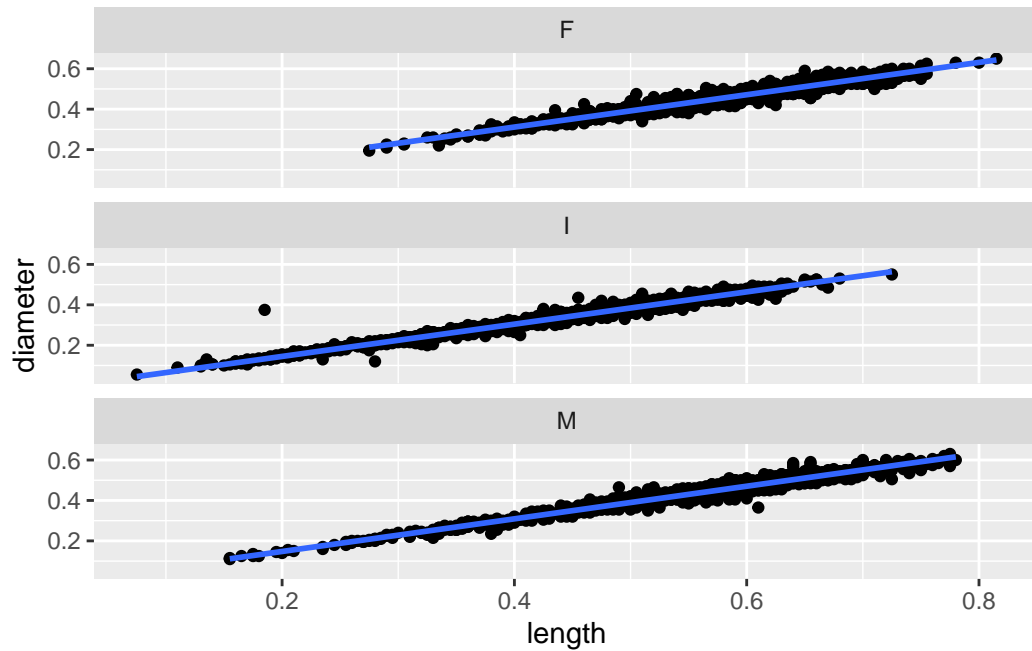
Question 2

> 💡 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`.

```
df %>%
  filter(length >= 0.5) %>%
  group_by(sex)%>%
  summarize(
    mean_length = mean(length),
```

```
    mean_diameter = mean(diameter),
    mean_height = mean(height),
    mean_whole_weight = mean(whole_weight),
    mean_shucked_weight = mean(shucked_weight),
    mean_viscera_weight = mean(viscera_weight),
    mean_shell_weight = mean(shell_weight),
    mean_rings = mean(rings)
) %>%
pivot_longer(-sex, names_to = "variable", values_to = "value")%>%
ggplot(aes(x = sex, y = value, fill = variable)) +
geom_bar(stat = "identity") +
labs(x = "Sex",
     y = "Mean value")
```
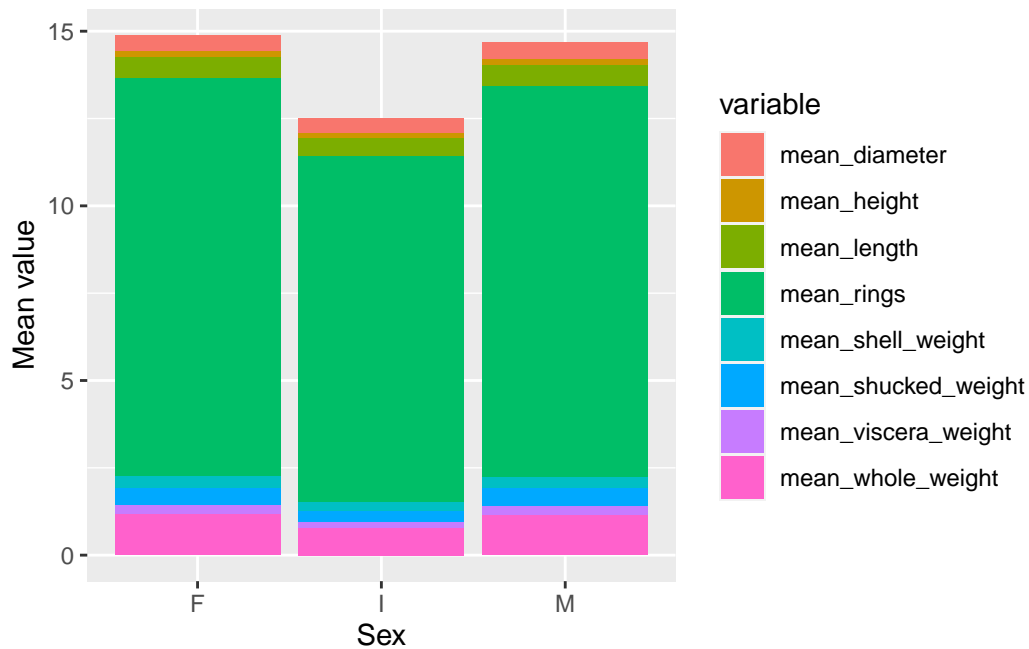

2.2 (15 points)

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 com-
   bination 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(newvar = ifelse(rings < 10, 'low', ifelse(rings > 20, 'high', 'med')))%>%
  group_by(newvar, sex) %>%
  summarize(avg_weight = mean(whole_weight + shucked_weight + viscera_weight + shell_weigh
  ggplot(aes(x = sex, y = newvar, fill = avg_weight)) +
  geom_tile()
```

`summarise()` has grouped output by 'newvar'. You can override using the
`.groups` argument.
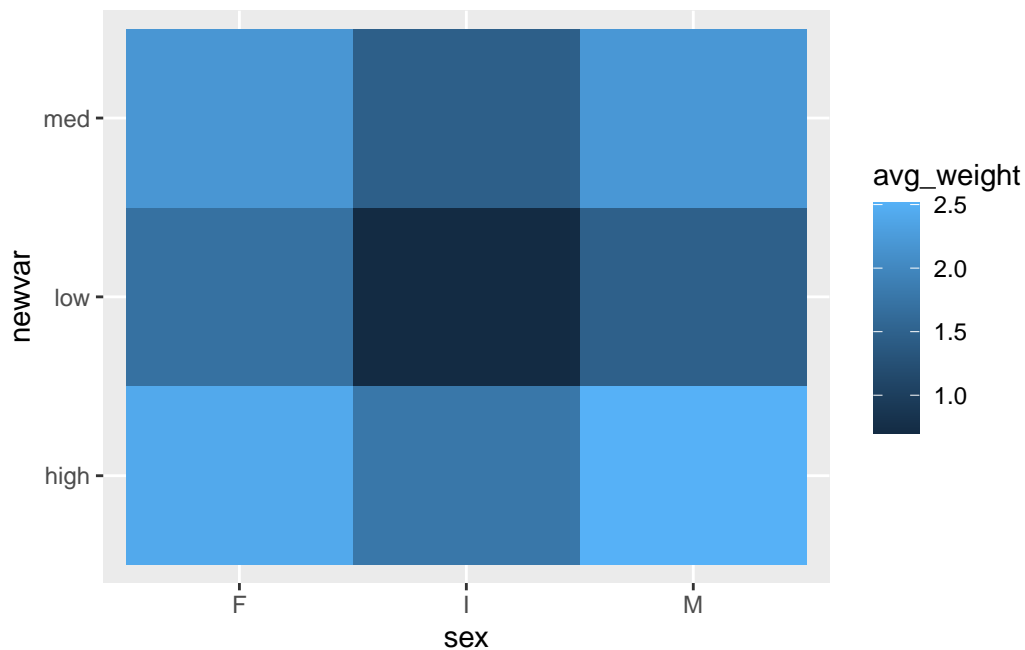
2.3 (5 points)

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]

```r
df %>%
  keep(is.numeric) %>%
  cor()%>%
  round(digits = 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.
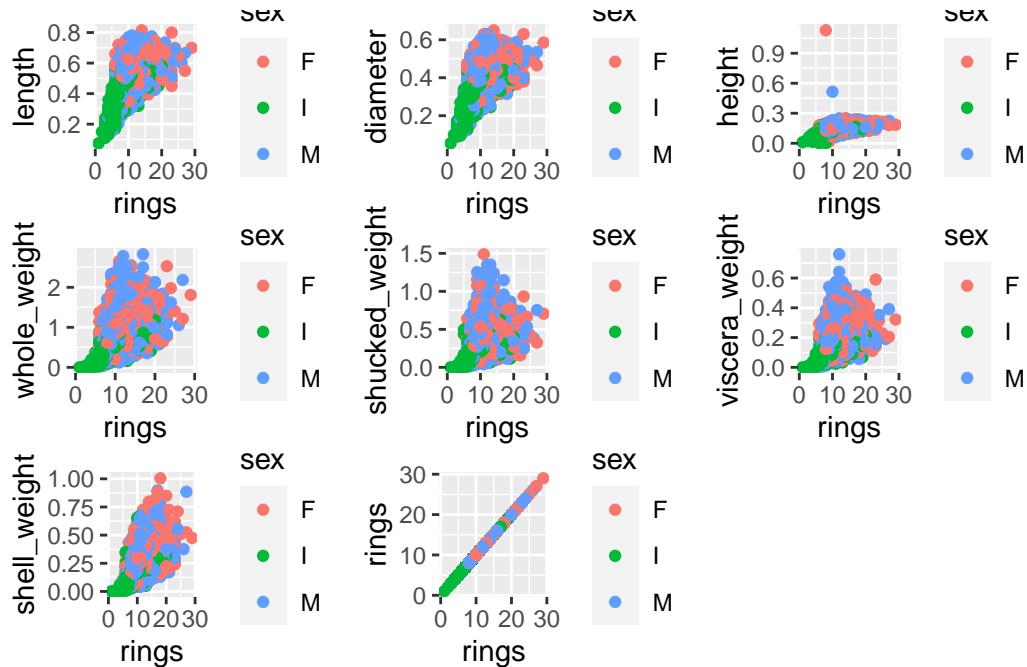
```r
plots <- map2(names(df)[2:9], c("Length", "Diameter", "Height", "Whole Weight", "Shucked W
  ggplot(df, aes(x = rings, y = !!sym(var), color = sex)) +
    geom_point()
```

---

[3]Table for 2.3

```
})
#I am not sure how to expand the view of the plots
cowplot::plot_grid(plotlist = 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 intercept means that for when diameter $= 0$, height would be -0.0038, the slope means that for every one unit increase in diameter, the height would increase by 0.351376. The p-value is < 2.2e-16, which is highly statistically significant, meaning that the slope is not 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)
regline = lm(height ~ diameter, data = df)
abline(regline, col = 'red')
```
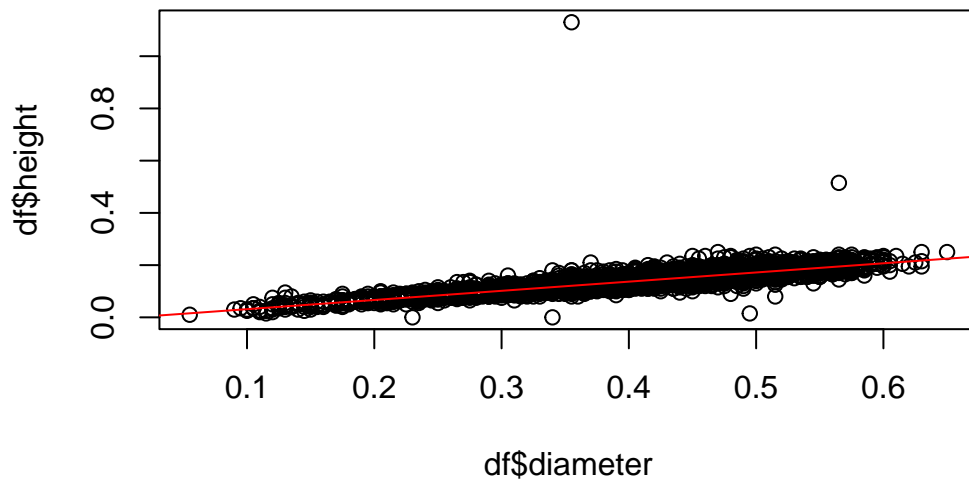
Yes, the line appears to fit the data, excluding outlines that should be removed.


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_height <- predict(regline, newdata = data.frame(diameter = new_diameters))
new_height
```

```
           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
```

```
plot(df$diameter, df$height)
regline = lm(height ~ diameter, data = df)
abline(regline, col = 'red')
points(new_diameters, new_height, col = "violet")
```
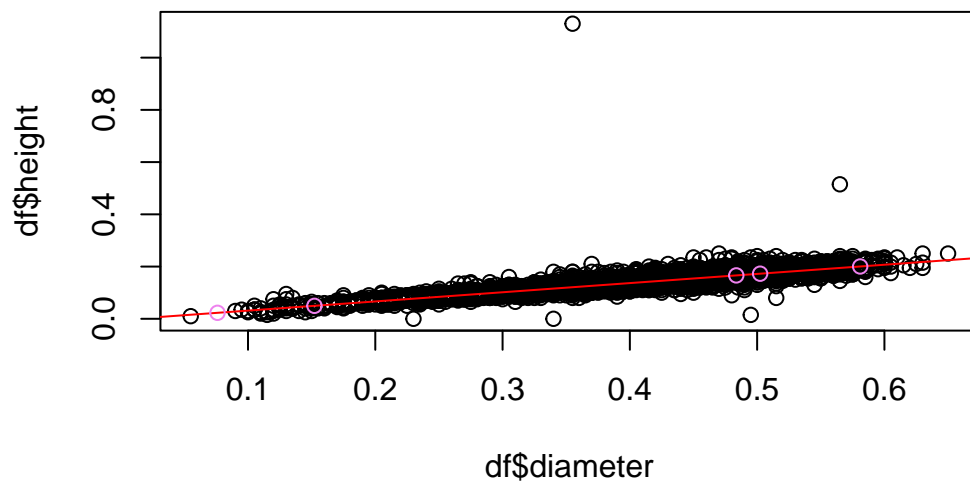
Appendix

> **i** Session Information
>
> Print your `R` session information using the following command
>
> ```r
> sessionInfo()
> ```
>
> ```
> R version 4.2.2 (2022-10-31 ucrt)
> 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.utf8
> [2] LC_CTYPE=English_United States.utf8
> [3] LC_MONETARY=English_United States.utf8
> [4] LC_NUMERIC=C
> [5] LC_TIME=English_United States.utf8
>
> 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_3.4.0 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
```