Homework 2

Jacob Adams

Table of contents

Link to the Github repository

❗ Due: Feb 9, 2024 @ 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)
```

```
Warning: package 'cowplot' was built under R version 4.3.2
```

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",
  "diameter",
  "height",
  "whole_weight",
  "shucked_weight",
  "viscera_weight",
  "shell_weight",
  "rings"
)

abalone <- read_csv(url, 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?

0 rows were dropped all the data is present.

```r
df <- na.omit(abalone)
```

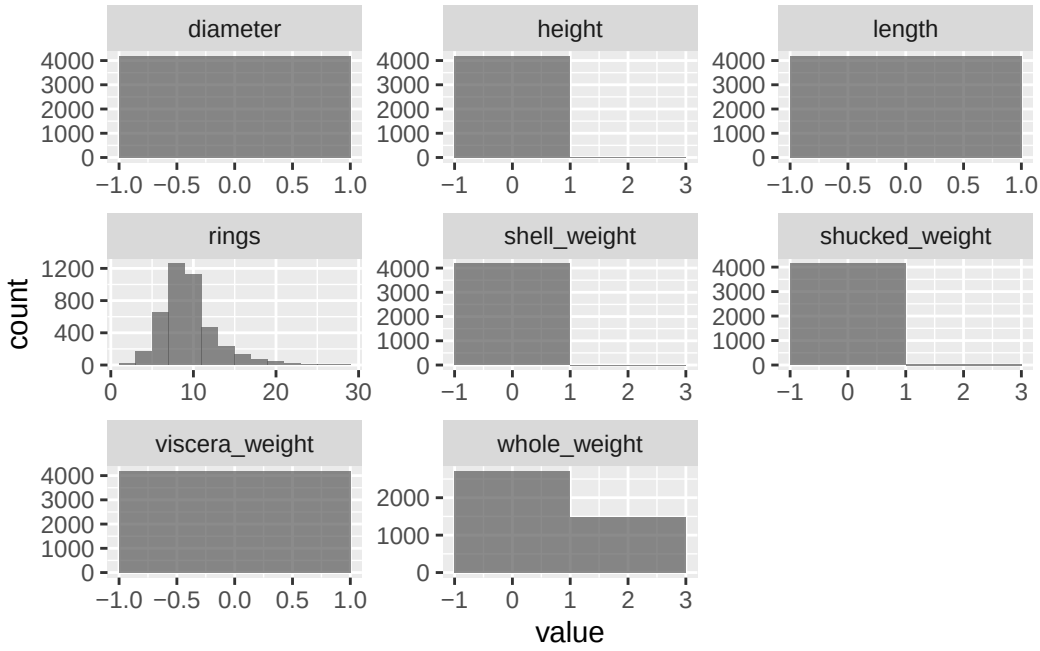
1.3 (5 points)

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

```
df_long <- pivot_longer(df, cols = c(diameter, length, height, whole_weight, shucked_weight,

df_long$sex <- NULL

ggplot(df_long,aes(x = value)) +
  geom_histogram(binwidth = 2, position = "identity", alpha = .7) +
  facet_wrap(~key, scales = "free")
```



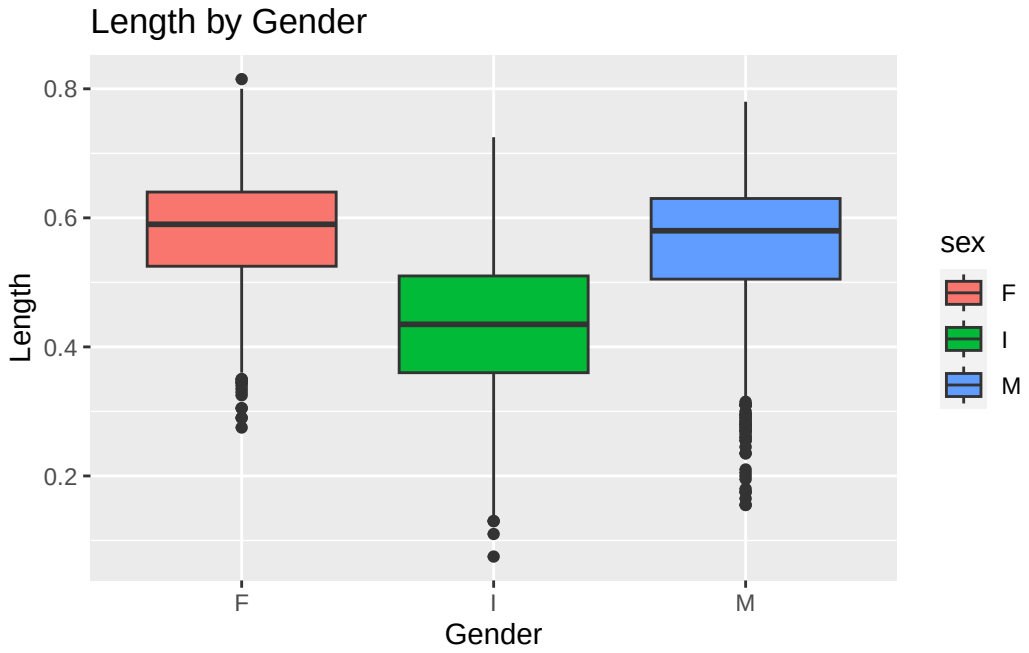
1.4 (5 points)

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?

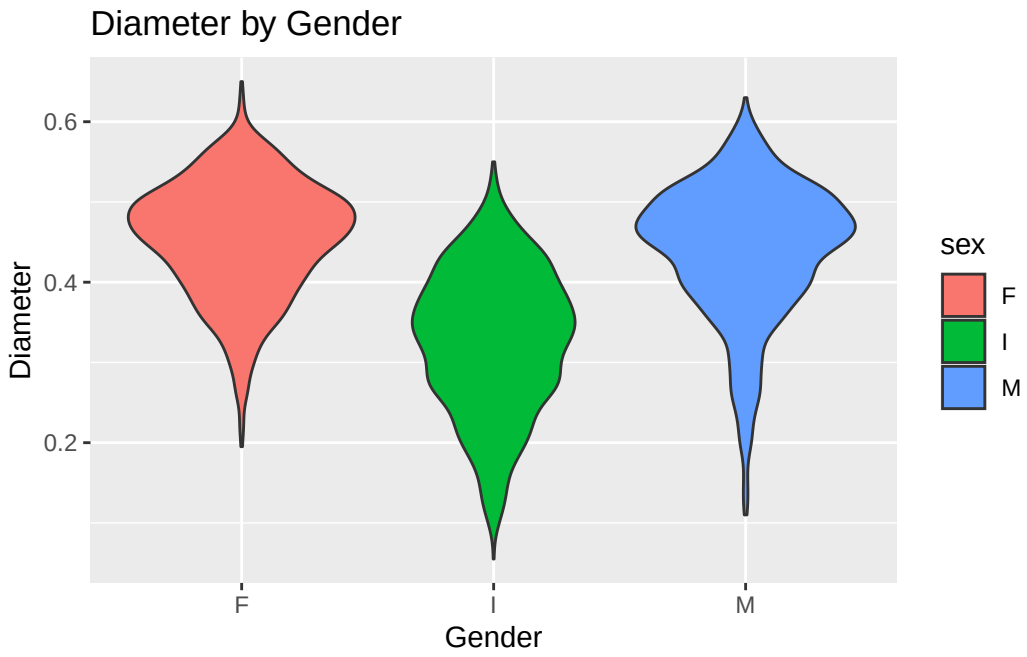
[1] You can use the `facet_wrap()` function for this. Have a look at its documentation using the help console in R

I did not notice any drastic differences in size for abalones by gender. The most interesting observation occurs in the number of male outliers that are smaller than most of their counterparts. Generally speaking, the females were slightly larger, and ,unsuprisingly, the infants were the smallest.

```
df %>%
  ggplot(aes(x = sex, y = length, fill = sex)) +
  geom_boxplot(position = "dodge") +
  labs(title = "Length by Gender", x = "Gender", y = "Length")
```




```
 #Violin Plot
df %>%
  ggplot(aes(x = sex, y = diameter, fill = sex)) +
  geom_violin(position = "dodge") +
  labs(title = "Diameter by Gender", x = "Gender", y = "Diameter")
```

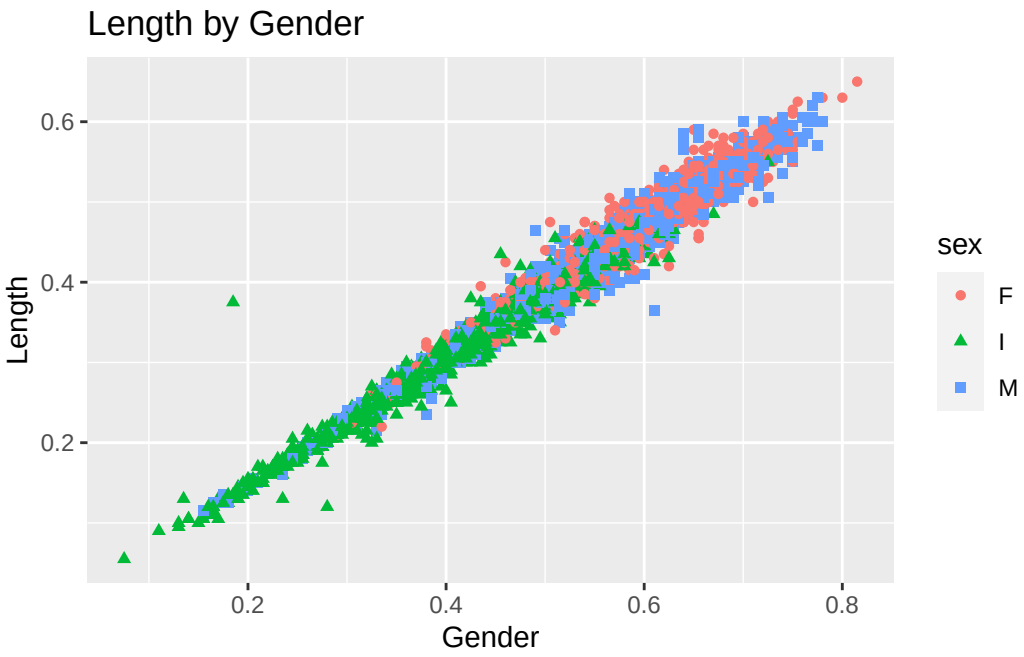


1.5 (5 points)

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?

Yes, some abalone in the gender I are far bigger than expected.

```
 # Insert your code here
df %>%
  ggplot(aes(x = length, y = diameter, color = sex, shape = sex)) +
  geom_point() +
  labs(title = "Length by Gender", x = "Gender", y = "Length")
```



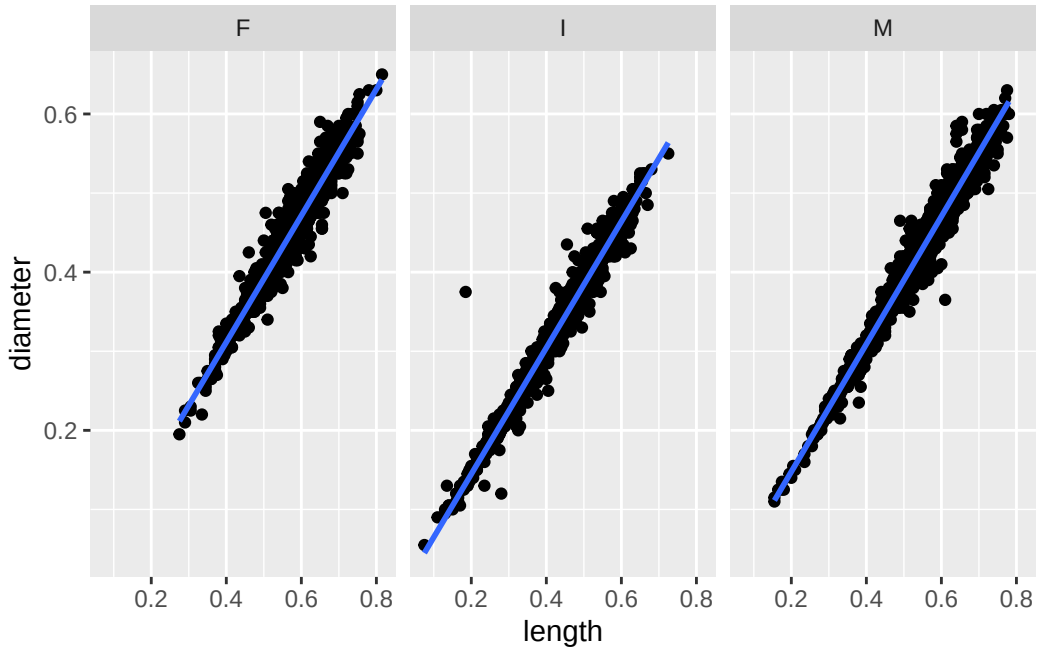
1.6 (5 points)

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]

```
# Insert your code here
df %>%
ggplot(aes(x = length, y = diameter)) +
geom_point() + geom_smooth(method = 'lm', se = FALSE) +
facet_wrap(~sex)
```

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

[2]Plot example for 1.6



Question 2

40 points

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

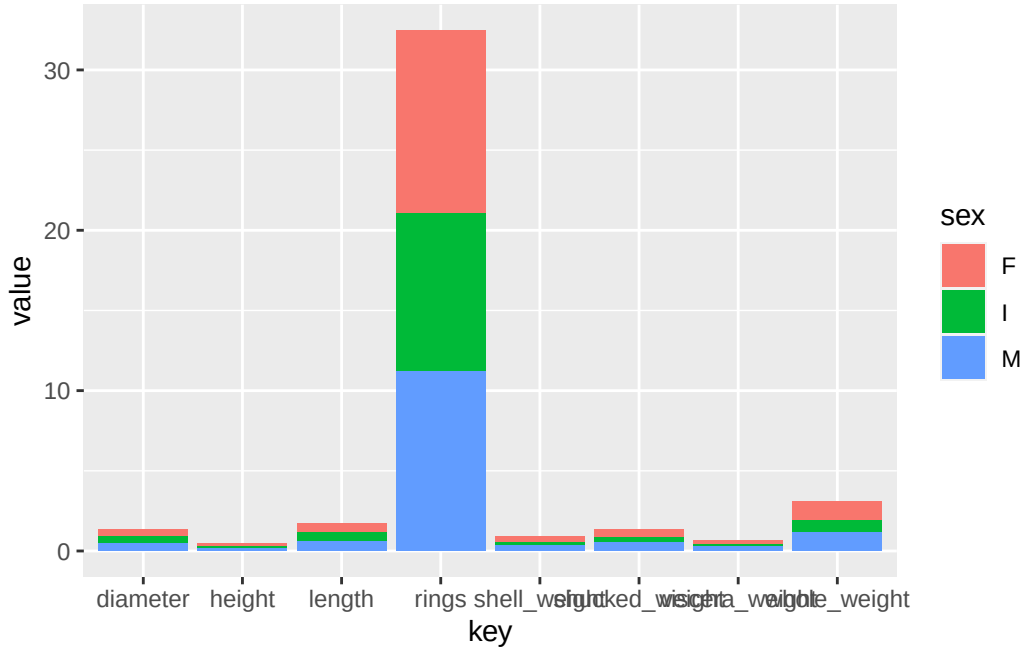
2.1 (10 points)

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`.

```
mean_df <- df %>%
  group_by(sex) %>%
  filter(length >= .5 ) %>%
  summarise(across(everything(), mean))
```

```r
df_long2 <- pivot_longer(mean_df, cols = c(diameter, length, height, whole_weight, shucked_we

df_long2 %>%
ggplot(aes(x = key, y = value, fill = sex)) + 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`.

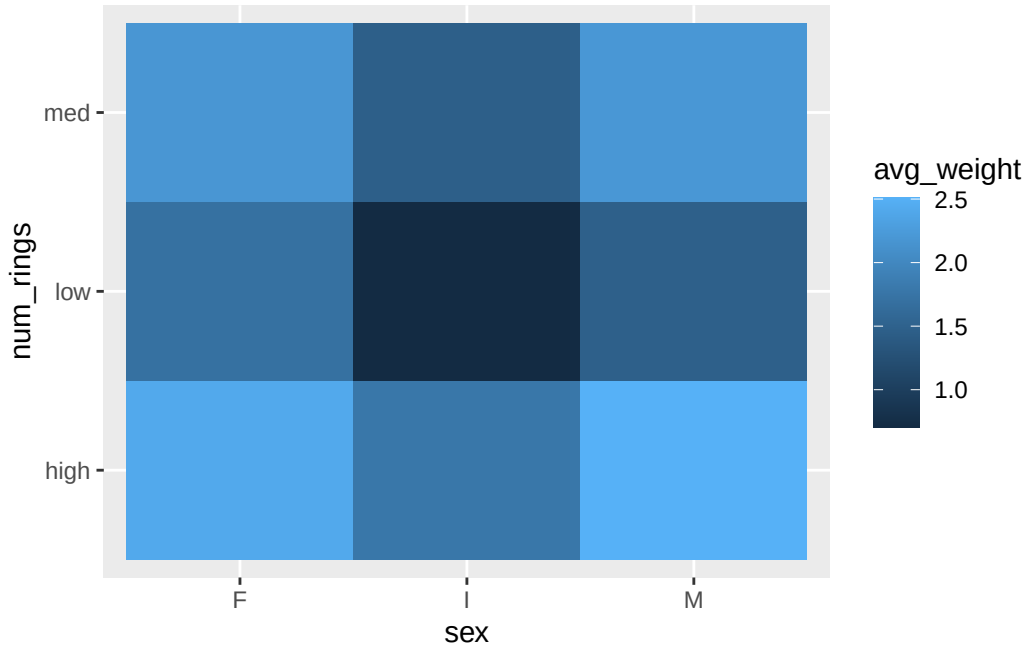
3. Use the `geom_tile()` function to create a tile plot of `num_rings` vs `sex` with the color indicating of each tile indicating the `avg_weight` value.

```r
 #1
df %>%
 mutate(

 num_rings = ifelse(rings < 10, "low", ifelse(rings >= 10 & rings <= 20, "med", "high"))

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

```
`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]

[3]Table for 2.3

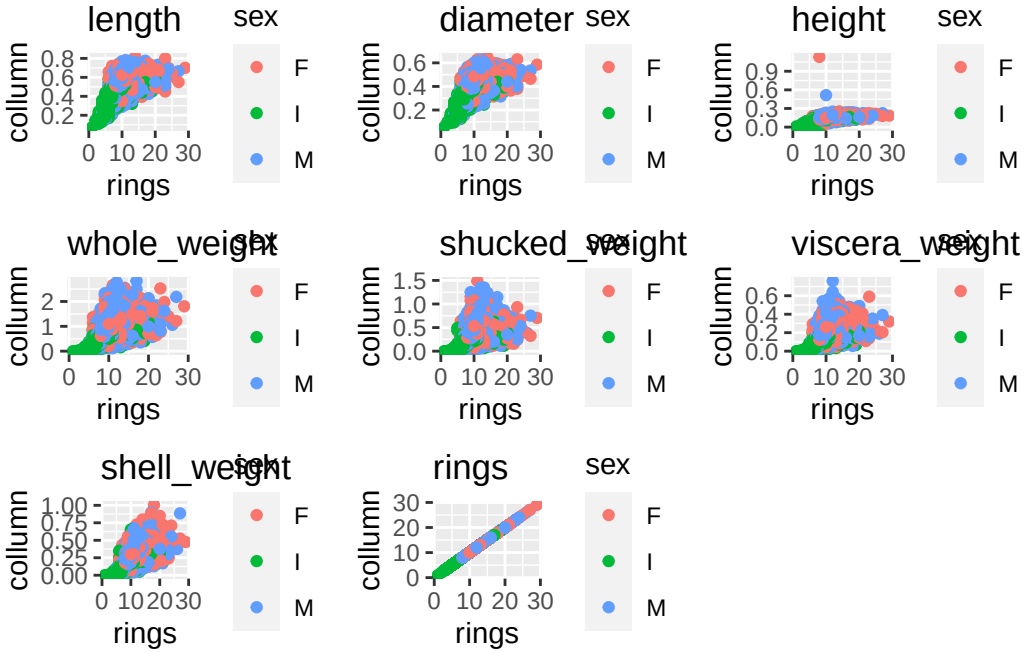
```
df2 <- df
df2$sex <- NULL
dfcor <- round(cor(df2, use = "everything"), 2)
```

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.

```
scatterer <- function(collumn,colname){
      ggplot(df) +
      geom_point(aes(x = rings, y = collumn, color=sex)) +
      ggtitle(as.character(colname))
    }

df %>%
  select(where(is.numeric)) %>%
  map2(colnames(.), scatterer) %>%
  cowplot::plot_grid(plotlist = .)
```



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.

Interpreting the model coefficients, we can see the p-value is very low for each coefficient, making diameter a trustworthy predictor of height.

```
regression_model <- lm(height~diameter,df)
summary(regression_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
```

```
coef(regression_model)
```

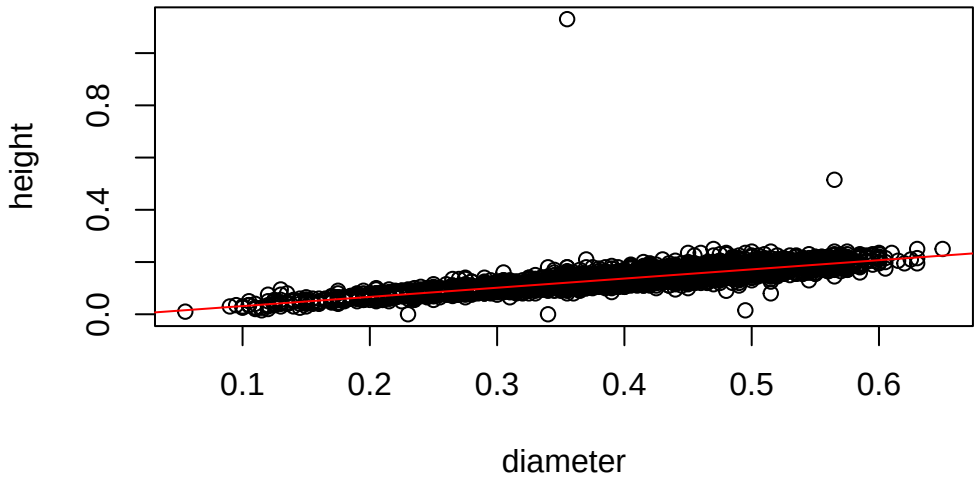
```
 (Intercept)     diameter
-0.003803398  0.351376278
```

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.

Yes, the linear model is appropriate for fitting the given data. Linear regression works best for variables that have a "linear" relationship. As we can clearly see, height and diameter have a visually appearing linear relationship. Also, the p-values of the co-coefficients for our model are very low. Thus, indicating their is a linear relationship.

```
plot(height~diameter, df)
abline(regression_model, col = "red")
```



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"`

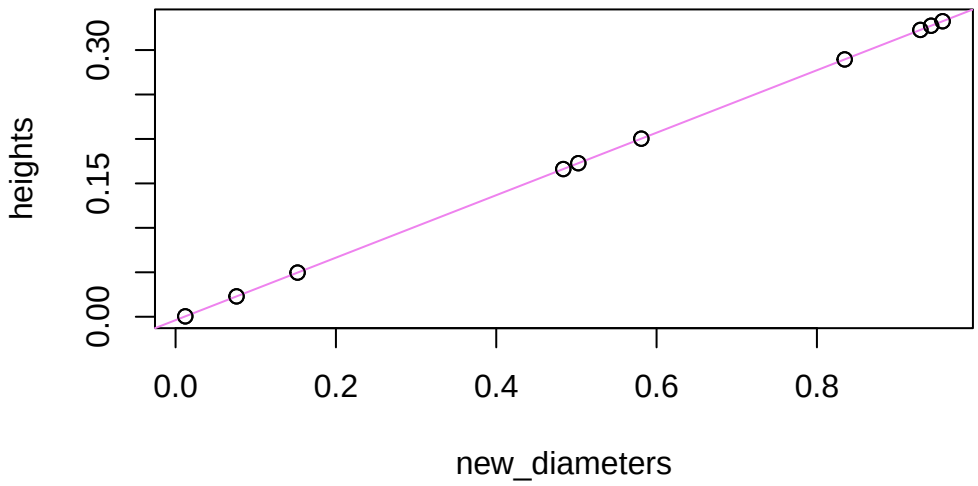
The expected values of height is height = -.003803398 + .351376278(diameter) for all new values of diameter. Judging by the previous model they should fit well.

In the new observations, they fit well along our model as expected.

```
new_diameters <- c(
  0.15218946,
  0.48361548,
  0.58095513,
  0.07603687,
  0.50234599,
  0.83462092,
  0.95681938,
  0.92906875,
  0.94245437,
  0.01209518
)


new_diameters_df <- data.frame(diameter = new_diameters)

heights <- predict(regression_model, new_diameters_df)
plot(new_diameters, heights)
abline(regression_model, col = "violet")
points(new_diameters,heights)
```


heights
0.00
0.15
0.30
0.0
0.2
0.4
0.6
0.8
new_diameters

Appendix

Session Information

Print your R session information using the following command

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 11 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

time zone: America/New_York
tzcode source: internal

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

other attached packages:
[1] cowplot_1.1.3 purrr_1.0.2   dplyr_1.1.2   ggplot2_3.4.3 tidyr_1.3.0
[6] readr_2.1.4

loaded via a namespace (and not attached):
 [1] Matrix_1.5-4.1    bit_4.0.5         gtable_0.3.4      jsonlite_1.8.7
 [5] crayon_1.5.2      compiler_4.3.1    tidyselect_1.2.0  parallel_4.3.1
 [9] splines_4.3.1     scales_1.2.1      yaml_2.3.7        fastmap_1.1.1
[13] lattice_0.21-8    R6_2.5.1          labeling_0.4.2    generics_0.1.3
[17] curl_5.0.2        knitr_1.43        tibble_3.2.1      munsell_0.5.0
[21] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.1       utf8_1.2.3
```

```
[25] xfun_0.40          bit64_4.0.5       cli_3.6.1          mgcv_1.8-42
[29] withr_2.5.0        magrittr_2.0.3    digest_0.6.33      grid_4.3.1
[33] vroom_1.6.3        rstudioapi_0.15.0 hms_1.1.3          nlme_3.1-162
[37] lifecycle_1.0.3    vctrs_0.6.3       evaluate_0.21      glue_1.6.2
[41] farver_2.1.1       fansi_1.0.4       colorspace_2.1-0   rmarkdown_2.24
[45] tools_4.3.1        pkgconfig_2.0.3   htmltools_0.5.6
```