Homework 2

Insert your name here

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:

```
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)
```

```
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.

```
library(readr)
url <- "http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"
```

```r
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)
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 `NA`s from the dataset and store the cleaned data in a tibble called `df`. How many rows were dropped?

```r
df <- abalone %>%
  na.omit()
rdropped <- nrow(abalone) - nrow(df)
message(rdropped, " row has been dropped")
```

```
0 row has been dropped
```

1.3 (5 points)

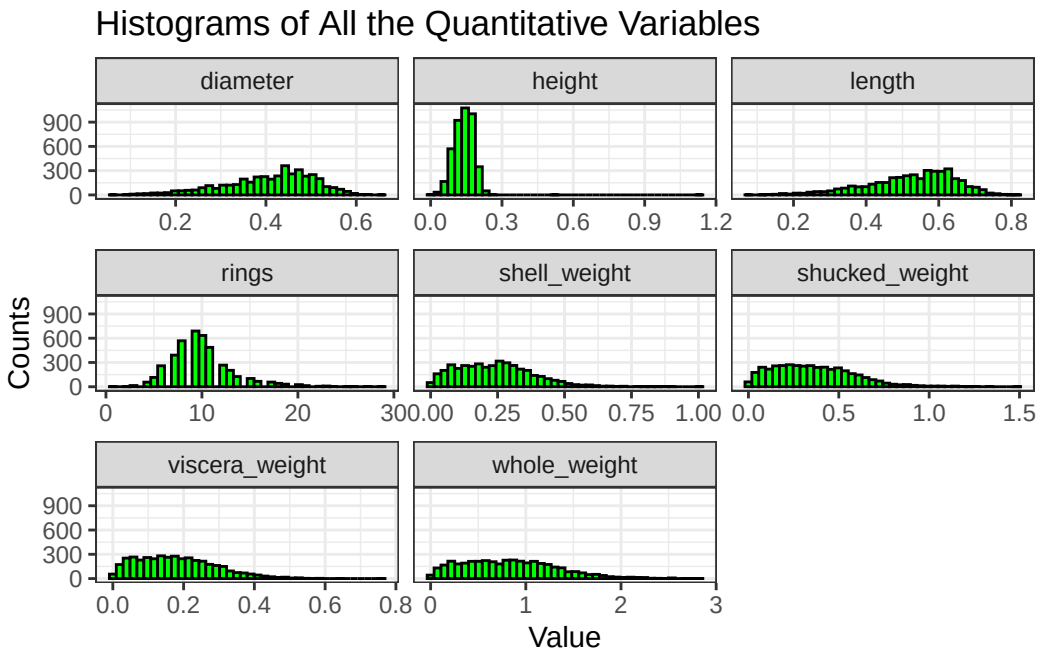
Plot histograms of all the quantitative variables in a **single plot** [1]

```
col_longer_names <- c(
  "length",
  "diameter",
  "height",
  "whole_weight",
  "shucked_weight",
  "viscera_weight",
  "shell_weight",
  "rings"
)

df_long <- pivot_longer(df, cols = col_longer_names)
```

```
Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(col_longer_names)

  # Now:
  data %>% select(all_of(col_longer_names))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
ggplot(df_long, aes(x = value)) +
  geom_histogram(bins = 40, fill = "green", color = "black") +
  facet_wrap(~ name, scales = "free_x") +
  theme_bw() +
  labs(x = "Value", y = "Counts", title = "Histograms of All the Quantitative Variables")
```

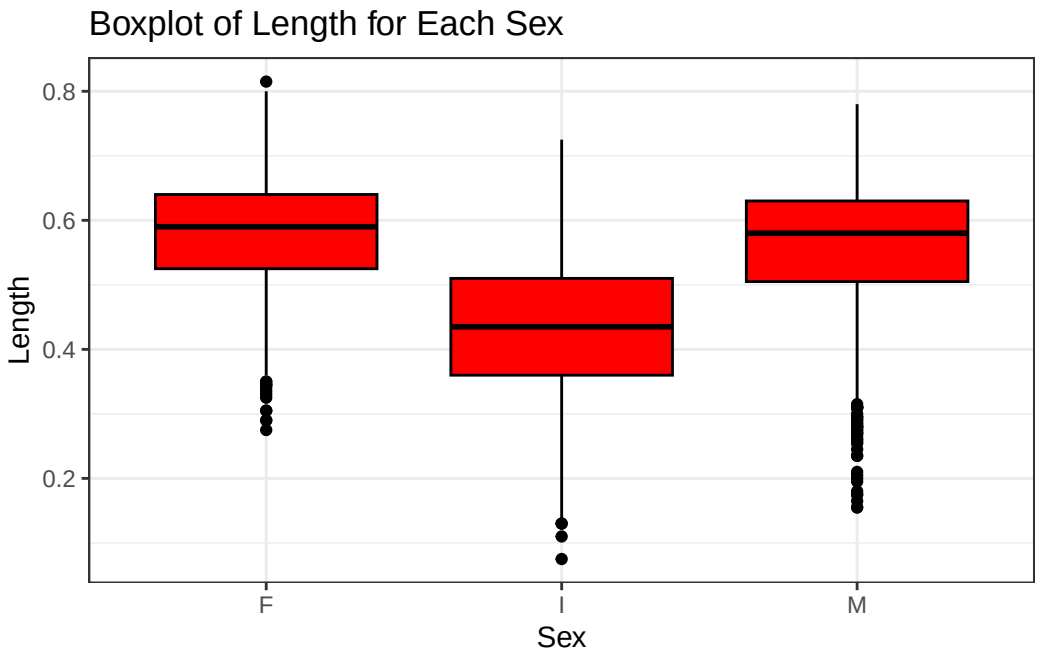
[1] You can use the `facet_wrap()` function for this. Have a look at its documentation using the help console in R



1.4 (5 points)

Create a boxplot of `length` for each `sex` and create a violin-plot of 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(fill = "red", color = "black") +
  theme_bw() +
  labs(title = "Boxplot of Length for Each Sex", x = "Sex", y = "Length")
```

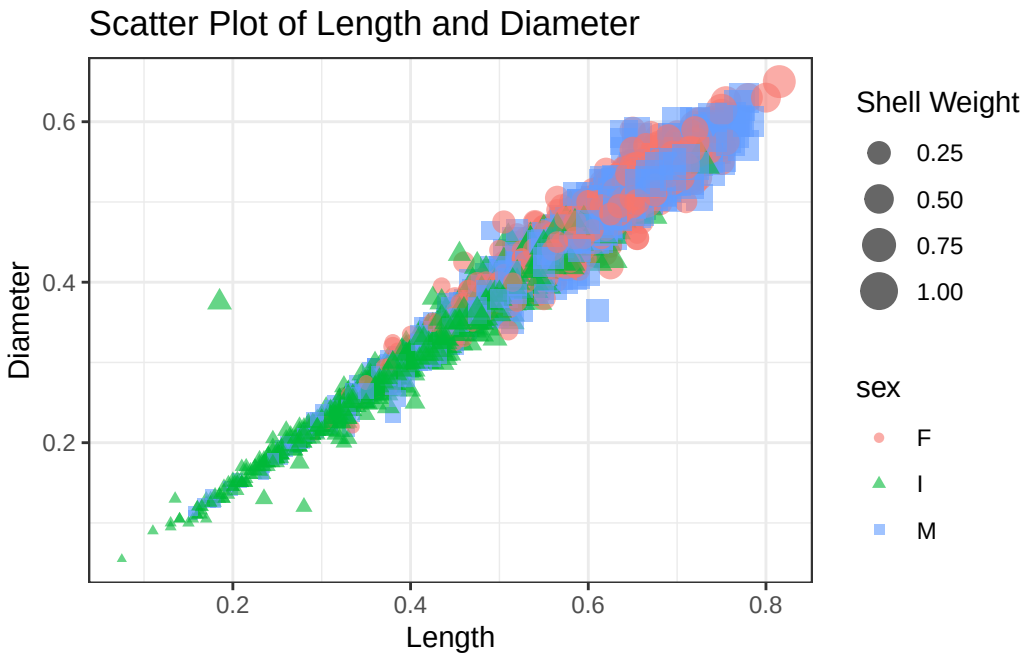



```
ggplot(df, aes(x = sex, y = diameter)) +
  geom_violin(fill = "lightblue", color = "black") +
  theme_bw() +
  labs(title = "Violin Plot of Diameter for Each Sex", x = "Sex", y = "Diameter")
```

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?

```
ggplot(df, aes(x = length, y = diameter, color = sex, shape = sex, size = shell_weight)) +
  geom_point(alpha = 0.6) +
  theme_bw() +
  labs(title = "Scatter Plot of Length and Diameter",
       x = "Length",
       y = "Diameter",
       size = "Shell Weight") +
  theme(legend.position = "right")
```



There is a green triangle(at about(0.2,0.4)) do not align with the trend.

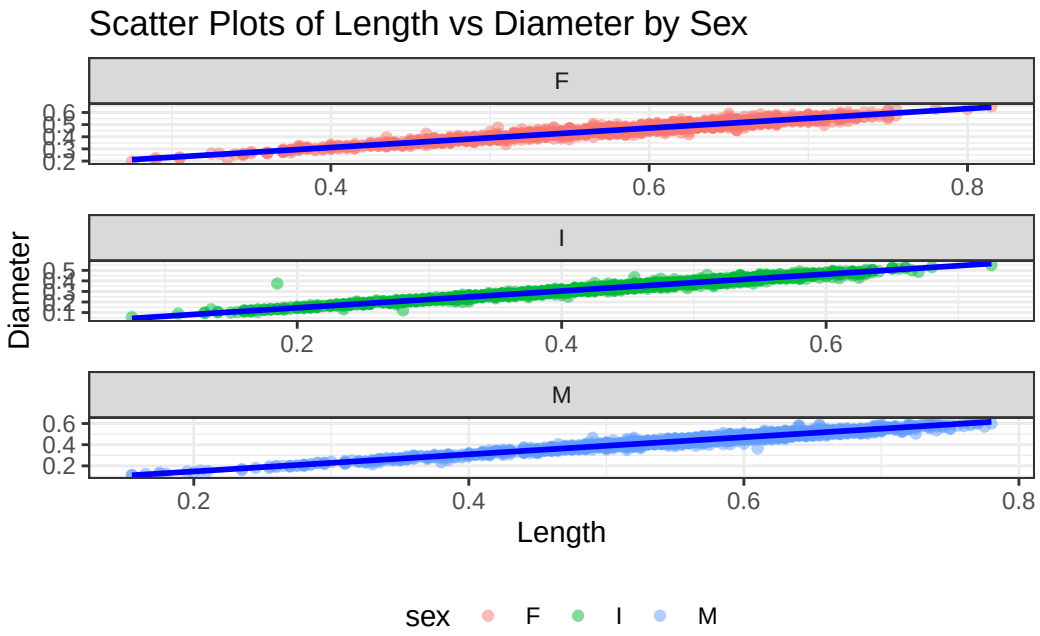
1.6 (5 points)

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(df, aes(x = length, y = diameter)) +
  geom_point(aes(color = sex), alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue") +
  facet_wrap(~ sex, scales = "free", ncol = 1) +
  theme_bw() +
  labs(title = "Scatter Plots of Length vs Diameter by Sex",
       x = "Length",
       y = "Diameter") +
  theme(legend.position = "bottom")
```

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

[2]Plot example for 1.6



Question 2

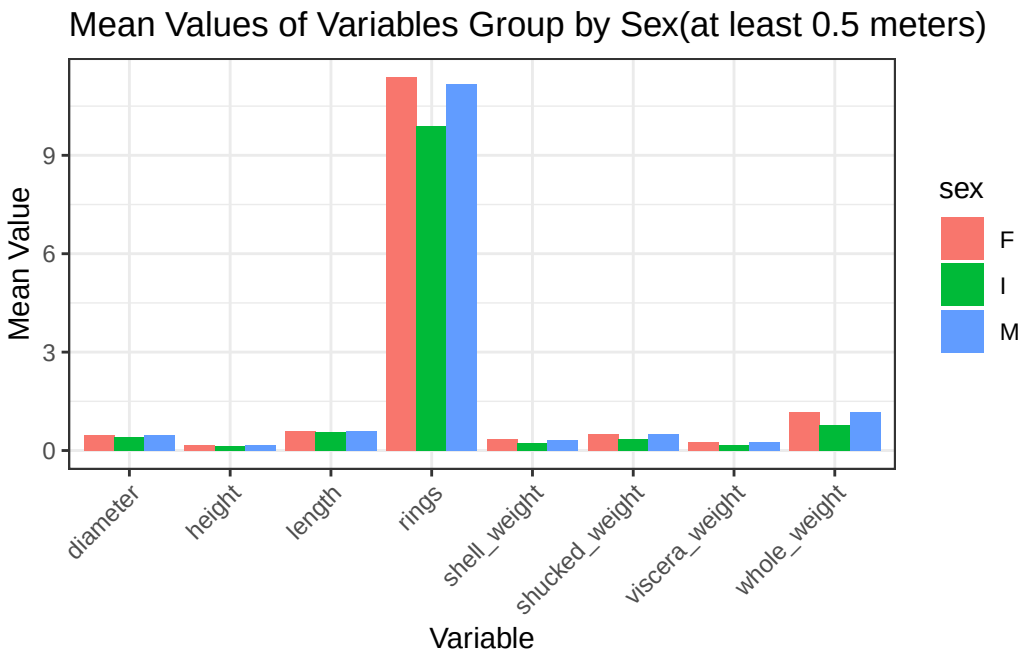
> 40 points
>
> More advanced analyses using `dplyr`, `purrrr` and `ggplot2`

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`.

```
abalone_filtered <- df %>%
  filter(length >= 0.5) %>%
  group_by(sex) %>%
  summarise(across(c(length, diameter, height, whole_weight, shucked_weight, viscera_weigh
  pivot_longer(-sex, names_to = "variable", values_to = "mean_value")
```

```r
ggplot(abalone_filtered, aes(x = variable, y = mean_value, fill = sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_bw() +
  labs(title = "Mean Values of Variables Group by Sex(at least 0.5 meters)",
       x = "Variable",
       y = "Mean Value") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



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:

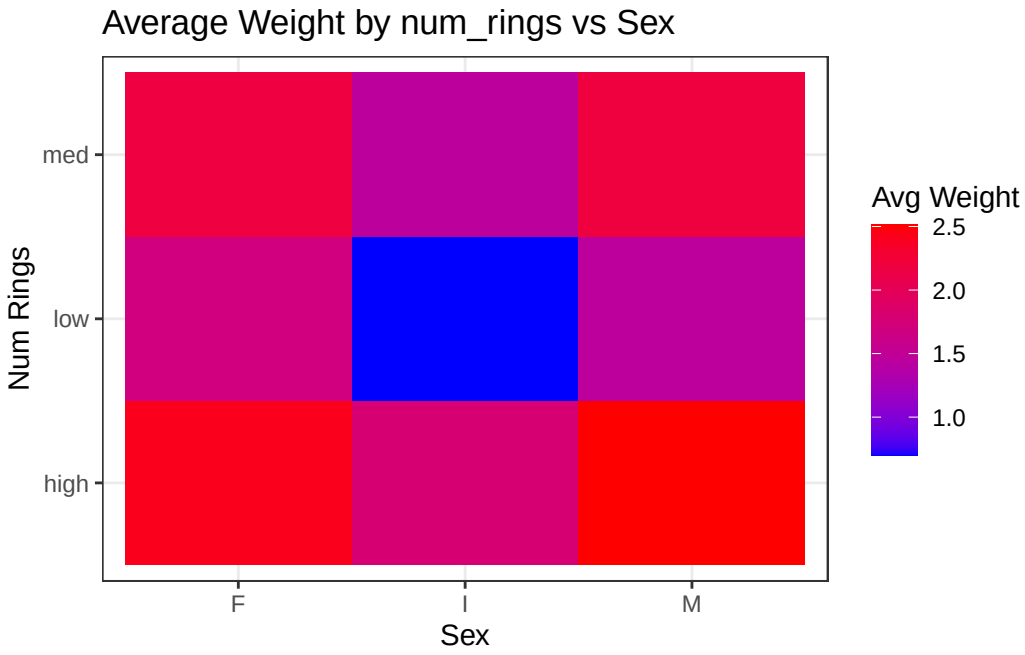
- `"low"` if `rings < 10`
- `"high"` if `rings > 20`, and
- `"med"` otherwise

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 = case_when(
    rings < 10 ~ "low",
    rings > 20 ~ "high",
    TRUE ~ "med"
  )) %>%
  group_by(num_rings, sex) %>%
  summarise(avg_weight = mean(whole_weight + shucked_weight + viscera_weight + shell_weigh
  ggplot(aes(x = sex, y = num_rings, fill = avg_weight)) +
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(title = "Average Weight by num_rings vs Sex", x = "Sex", y = "Num Rings", fill = "A
  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 [3]

```
cor_df <- df %>%
  select_if(is.numeric) %>%
  cor() %>%
  round(2) %>%
  as_tibble(cor_table, rownames = "Variable")

cor_df
```

```
# A tibble: 8 x 9
  Variable     length diameter height whole_weight shucked_weight viscera_weight
  <chr>         <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
1 length         1        0.99   0.83         0.93           0.9            0.9
2 diameter       0.99     1      0.83         0.93           0.89           0.9
3 height         0.83     0.83   1            0.82           0.77           0.8
4 whole_weight   0.93     0.93   0.82         1              0.97           0.97
5 shucked_wei~   0.9      0.89   0.77         0.97           1              0.93
6 viscera_wei~   0.9      0.9    0.8          0.97           0.93           1
7 shell_weight   0.9      0.91   0.82         0.96           0.88           0.91
8 rings          0.56     0.58   0.56         0.54           0.42           0.5
# i 2 more variables: shell_weight <dbl>, rings <dbl>
```

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.

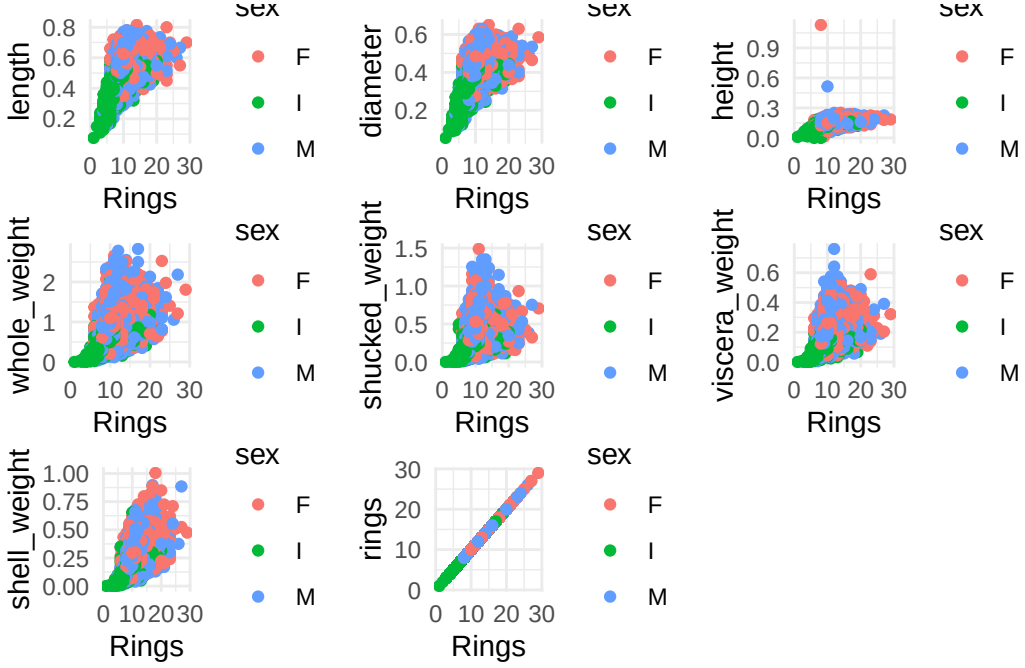
```
col_names <- select(df, where(is.numeric))

plots <- map2(names(col_names), list(df$rings), ~{
  ggplot(df, aes_string(x = .y, y = .x, color = "sex")) +
    geom_point() +
    labs(x = "Rings", y = .x) +
```

[3] Table for 2.3

```
    theme_minimal()
})
```

```
Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
i Please use tidy evaluation idioms with `aes()`.
i See also `vignette("ggplot2-in-packages")` for more information.
```

```
# Arrange the plots into a grid
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, df)

summary(model)
```

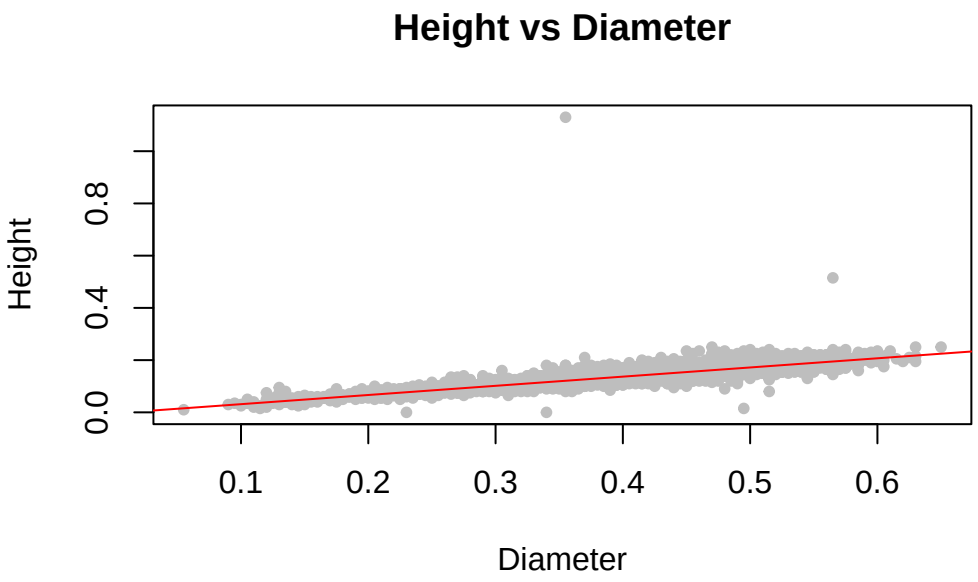
```
Call:
lm(formula = height ~ diameter, data = df)

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
```

Diameter is 0.35 means for each one-unit increase in diameter, the height of the abalone increases by approximately 0.35 units. p-value is much less than 0.05 indicates "diameter" is a significant predictor of "height" in the datasets.

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,
     xlab = "Diameter", ylab = "Height",
     main = "Height vs Diameter",
     col = "grey", pch=20)
abline(model, col="red")
```



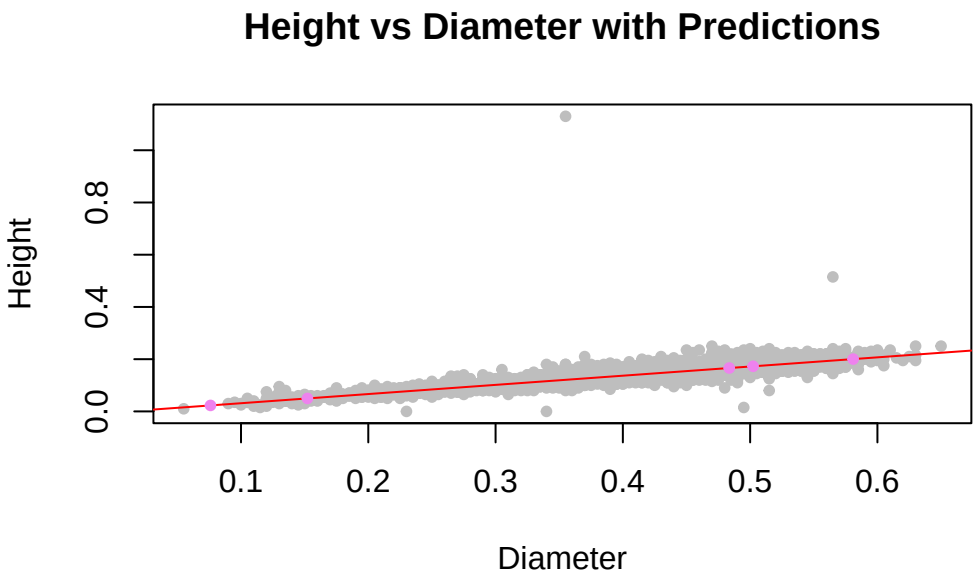
The linear model is an appropriate fit for this relationship, since the line follows the trend of the data.

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"`

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

```r
new_df <- data.frame(diameter = new_diameters)

pred_height <- predict(model, new_df)

plot(df$diameter, df$height,
     xlab = "Diameter", ylab = "Height",
     main = "Height vs Diameter with Predictions",
     col = "grey", pch=20)
abline(model, col="red")

points(new_diameters, pred_height, col="violet", pch=20)
```



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=Chinese (Simplified)_China.utf8
[2] LC_CTYPE=Chinese (Simplified)_China.utf8
[3] LC_MONETARY=Chinese (Simplified)_China.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_China.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_1.1.3   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    gtable_0.3.4      jsonlite_1.8.7    compiler_4.3.1
 [5] tidyselect_1.2.0  splines_4.3.1     scales_1.2.1      yaml_2.3.7
 [9] fastmap_1.1.1     lattice_0.21-8    R6_2.5.1          labeling_0.4.3
[13] generics_0.1.3    knitr_1.44        tibble_3.2.1      munsell_0.5.0
[17] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.1       utf8_1.2.3
```

```
[21] xfun_0.40          cli_3.6.1          mgcv_1.8-42        withr_2.5.1
[25] magrittr_2.0.3     digest_0.6.33      grid_4.3.1         rstudioapi_0.15.0
[29] hms_1.1.3          nlme_3.1-162       lifecycle_1.0.3    vctrs_0.6.3
[33] evaluate_0.21      glue_1.6.2         farver_2.1.1       fansi_1.0.4
[37] colorspace_2.1-0   rmarkdown_2.25     tools_4.3.1        pkgconfig_2.0.3
[41] htmltools_0.5.6
```