Author: *Vishnu Varthan*

==========

Question 1 : EDA using readr, tidyr and ggplot2

1.1 ---> 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 Remove missing values and NAs from the dataset and store the cleaned data in a tibble called df. How many rows were dropped?

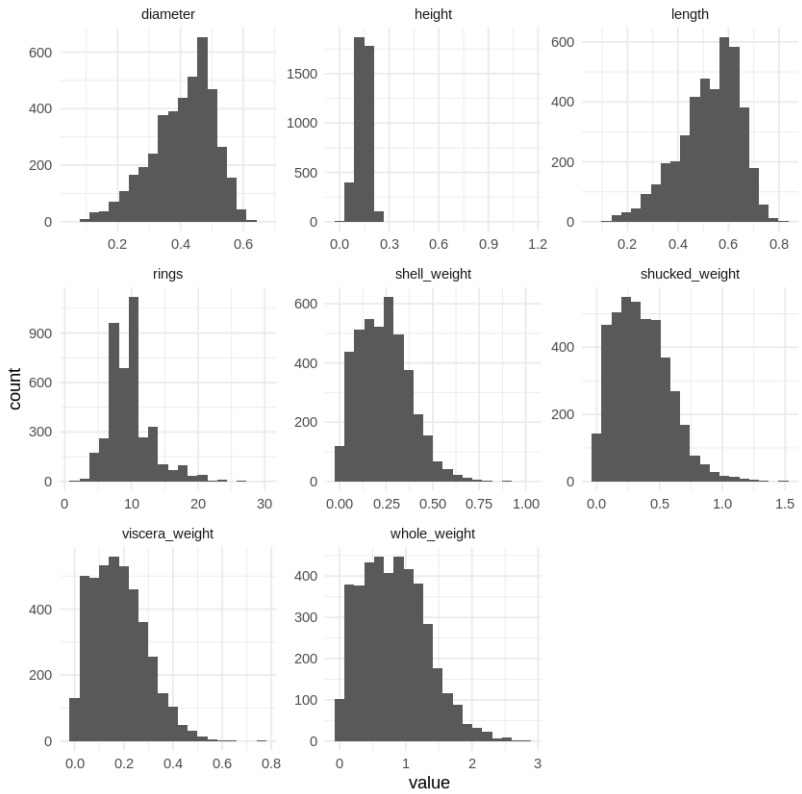
```
df <- na.omit(abalone)
dropped_rows <- nrow(abalone) - nrow(df)

#Number of rows dropped
print(paste("Number of rows dropped:", dropped_rows))
```

```
[1] "Number of rows dropped: 0"
```

==========

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

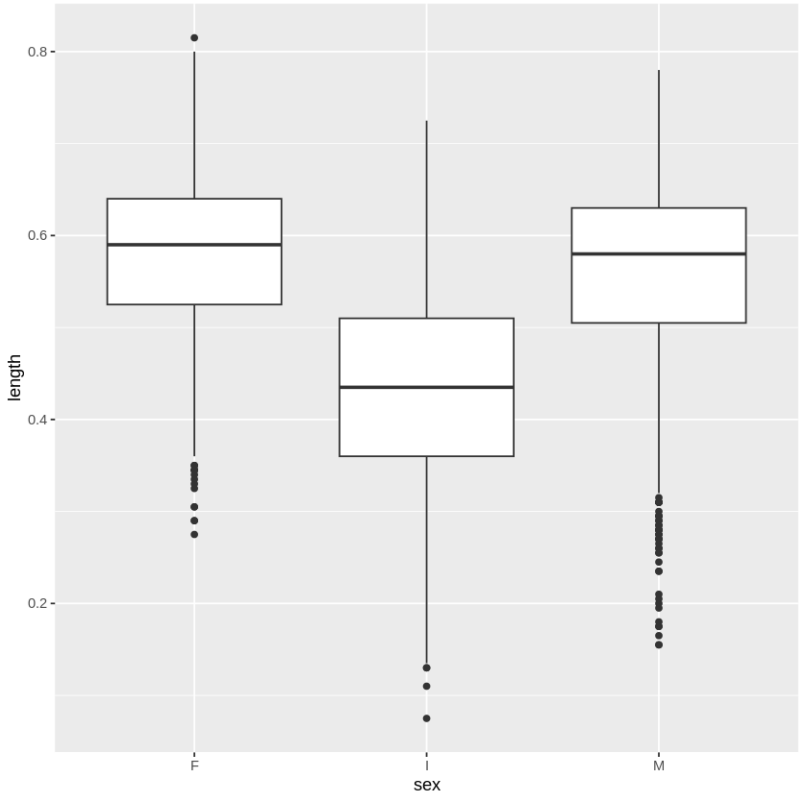
```
library(tidyr)
library(ggplot2)

histogram <- df %>%
  gather(variable, value, -sex) %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 20) +
  facet_wrap(~variable, scales = "free") +
  theme_minimal()

print(histogram)
```

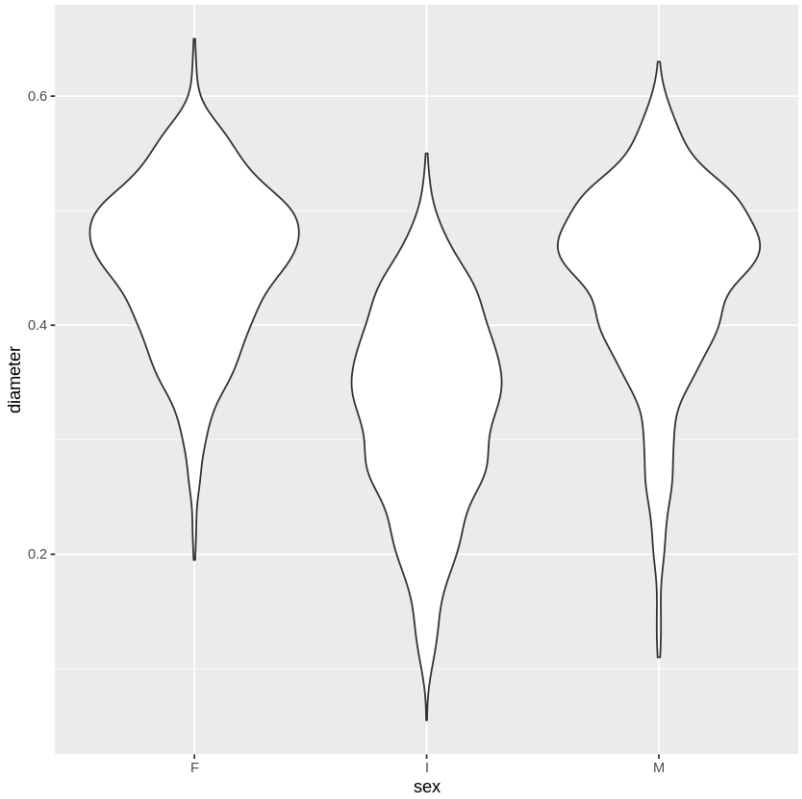


==========

1.4 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?

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

violinplot_dia <- ggplot(df, aes(x = sex, y = diameter)) +
  geom_violin()

print(boxplot_len)
print(violinplot_dia)
```





Based on the analysis of abalone data, we can observe the following differences in the body of the abalone:

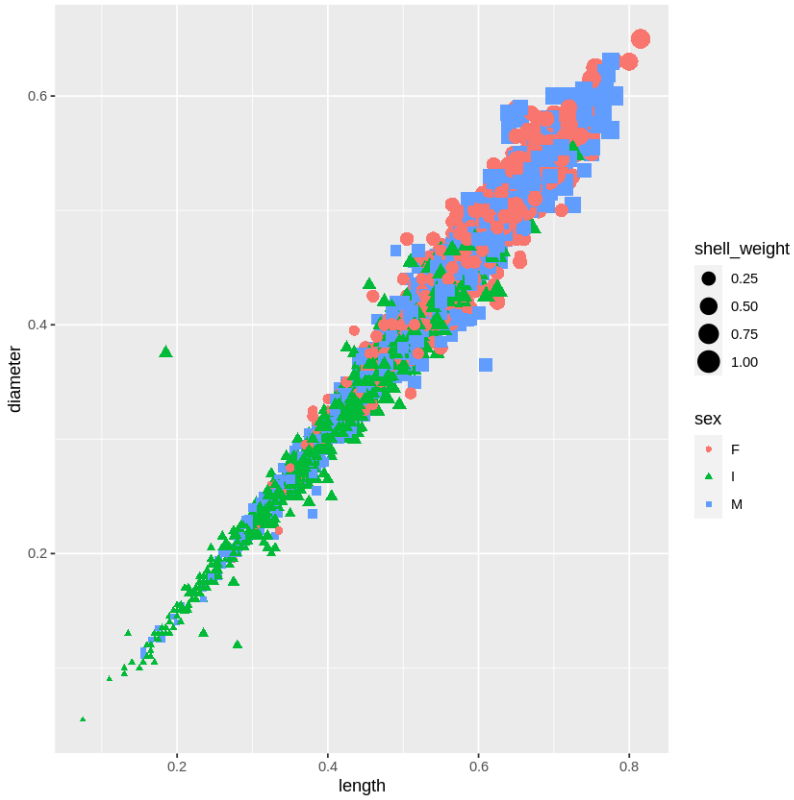
- The length of the abalone varies according to gender. Female abalone tend to be shorter than male abalone; infant abalone is the shortest.

- The diameter of the abalone also varies depending on gender. Male abalone is larger in diameter than female and infant abalone.

==========

1.5 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?

```
scatter_plot <- ggplot(df, aes(x = length, y = diameter, color = sex, shape = sex, size = she
  geom_point() +
  scale_size_continuous(range = c(1, 6))

print(scatter_plot)

```



Based on the scatter plot, we can see that there are some discrepancies in the data set. These outliers are represented by points far from the main points.

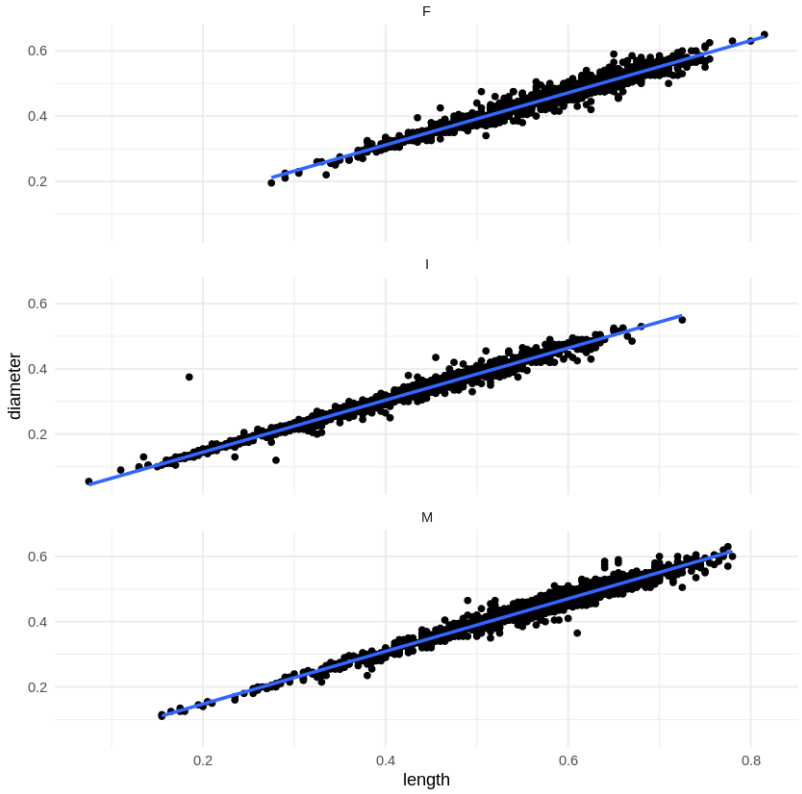
==========

1.6 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:

```
scatter_with_trend <- ggplot(df, aes(x = length, y = diameter)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~sex, ncol = 1) +
  theme_minimal()

print(scatter_with_trend)

```

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

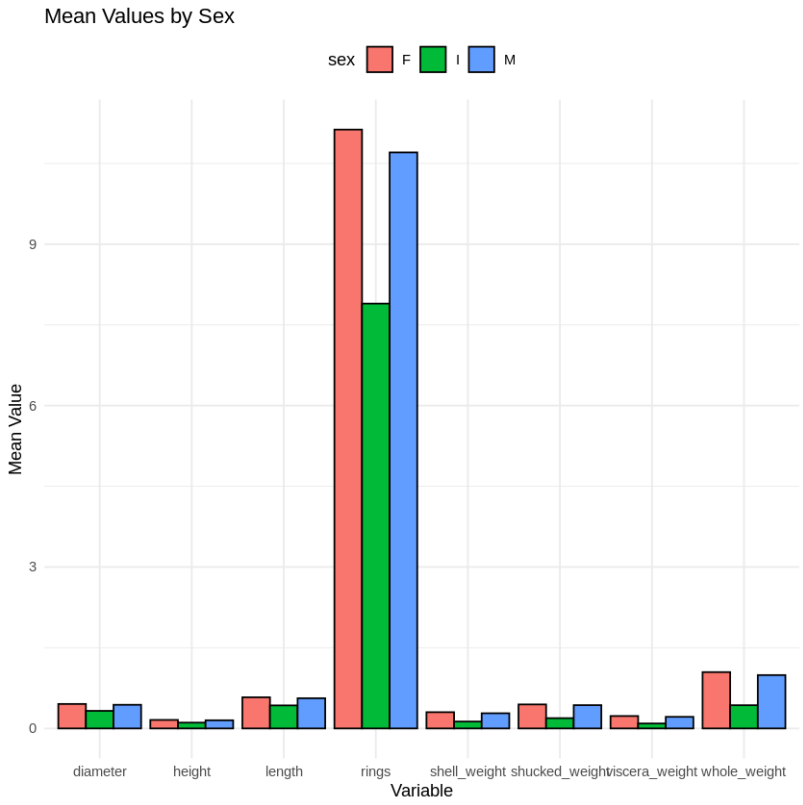


Question 2: More advanced analyses using dplyr, purrrr and ggplot2

2.1 Filter the data to only include abalone with a length of at least 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.

```r
library(dplyr)
library(ggplot2)

# Filter data for abalone with length >= 0.1 meters
filtered_df <- df %>%
  filter(length >= 0.1)

# Group data by sex and calculate mean of each variable
mean_values <- filtered_df %>%
  group_by(sex) %>%
  summarise(across(everything(), mean))

# Reshape data for plotting
mean_values_long <- mean_values %>%
  pivot_longer(cols = -sex, names_to = "variable", values_to = "mean_value")

# Create bar plot
bar_plot <- ggplot(mean_values_long, aes(x = variable, y = mean_value, fill = sex)) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  labs(title = "Mean Values by Sex", x = "Variable", y = "Mean Value") +
  theme_minimal() +
  theme(legend.position = "top")

print(bar_plot)


```

```
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
```



==========

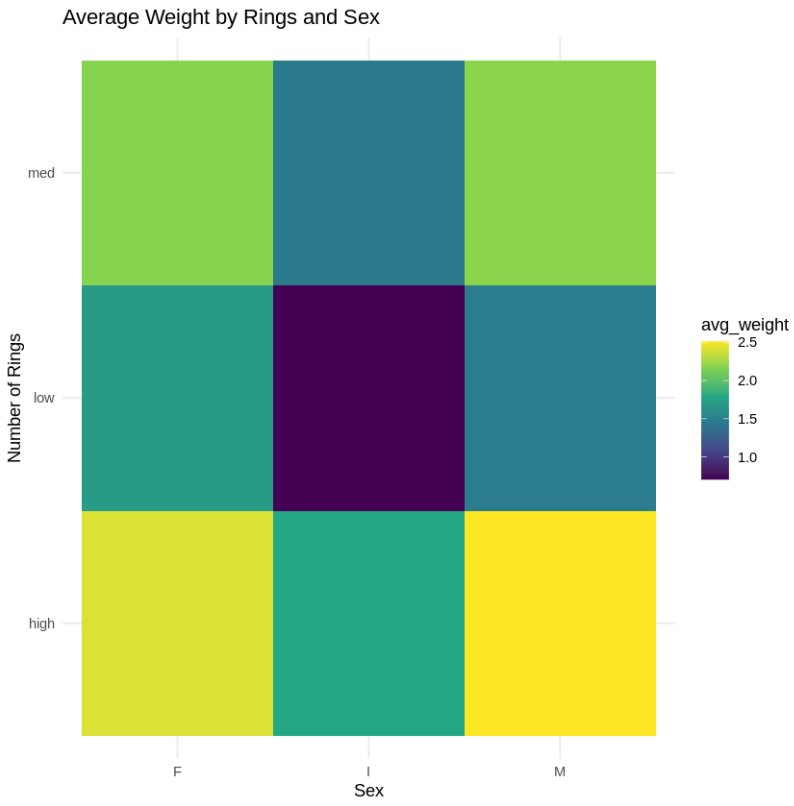
2.2 Implement the following in a single command:

Temporarily create a new variable called num_rings which takes a value of: "low" if rings < 10 "high" if rings > 20, and "med" otherwise 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.

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

```
# Create num_rings variable
df <- df %>%
  mutate(num_rings = case_when(
    rings < 10 ~ "low",
    rings > 20 ~ "high",
    TRUE ~ "med"
  ))

# Group by num_rings and sex, compute avg_weight
avg_weight <- df %>%
  group_by(num_rings, sex) %>%
  summarise(avg_weight = mean(whole_weight + shucked_weight + viscera_weight + shell_weight))

# Create tile plot
tile_plot <- ggplot(avg_weight, aes(x = sex, y = num_rings, fill = avg_weight)) +
  geom_tile() +
  labs(title = "Average Weight by Rings and Sex", x = "Sex", y = "Number of Rings") +
  scale_fill_viridis_c() +
  theme_minimal()

print(tile_plot)

```

`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.

```
# Calculate pairwise correlations
cor_table <- df %>%
  select_if(is.numeric) %>%
  cor() %>%
  round(2)

# Print the correlation table
print(cor_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     0.57   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
```

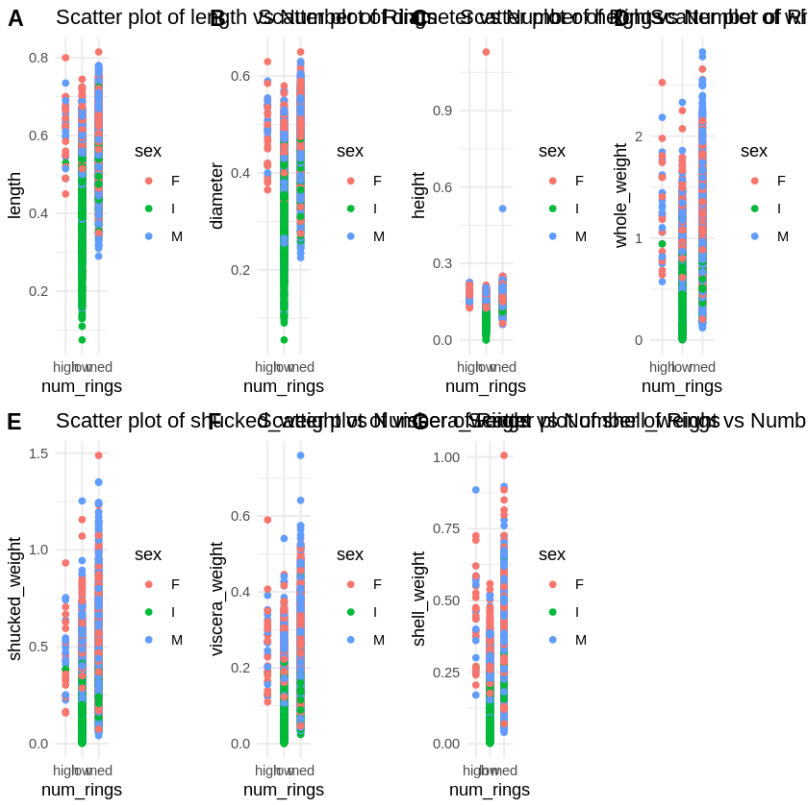
==========

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.

```
install.packages("cowplot")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library(purrr)
library(cowplot)

# Define function to create scatter plot
create_scatter_plot <- function(variable) {
  ggplot(df, aes_string(x = "num_rings", y = variable, color = "sex")) +
    geom_point() +
    labs(title = paste("Scatter plot of", variable, "vs Number of Rings")) +
    theme_minimal()
}

# List of quantitative variables
quantitative_vars <- c("length", "diameter", "height", "whole_weight", "shucked_weight", "vis

# Create scatter plots using map function
scatter_plots <- map(quantitative_vars, function(var) create_scatter_plot(var))

# Combine scatter plots into a grid
grid <- plot_grid(plotlist = scatter_plots, nrow = 2, labels = "AUTO")

print(grid)

```



Ques 3 Linear regression using lm

3.1: Simple Linear Regression

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

# Interpretation of coefficients
# Intercept (β0): The expected height when diameter is 0.
# Slope (β1): The change in height for a one-unit increase in diameter.
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
```

3.2: Scatterplot with Regression Line

```
plot(df$diameter, df$height, xlab = "Diameter", ylab = "Height", main = "Scatterplot of Heig
abline(model, col = "red")
```