Homework 2

Leo Soccio

Table of contents

. 2
Question 1 2
Question 2 8
Question 3 11

Appendix **15**

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
```

```r
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.

```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, col.names=abalone_col_names, na.strings=c("","NA"))
abalone <- tibble (abalone)
```

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?

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

```
[1] 0
```

No rows were dropped.

1.3 (5 points)

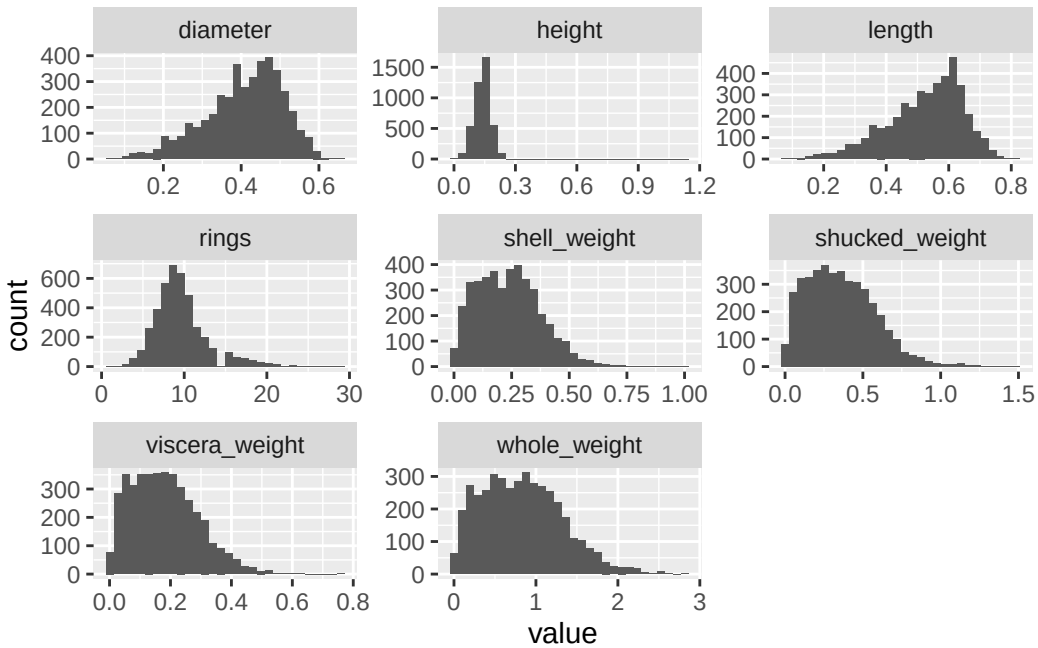
Plot histograms of all the quantitative variables in a **single plot** [1]

```
quant <- df %>% select(-sex)
pivotq <- quant %>%
  pivot_longer(cols=c(length,diameter,height,whole_weight,shucked_weight,
                      viscera_weight,shell_weight,rings)
               ,names_to="variable",values_to="value")
ggplot(pivotq, aes(x=value))+geom_histogram()+facet_wrap(vars(variable),
```

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

```
                                                                scales="free")
```

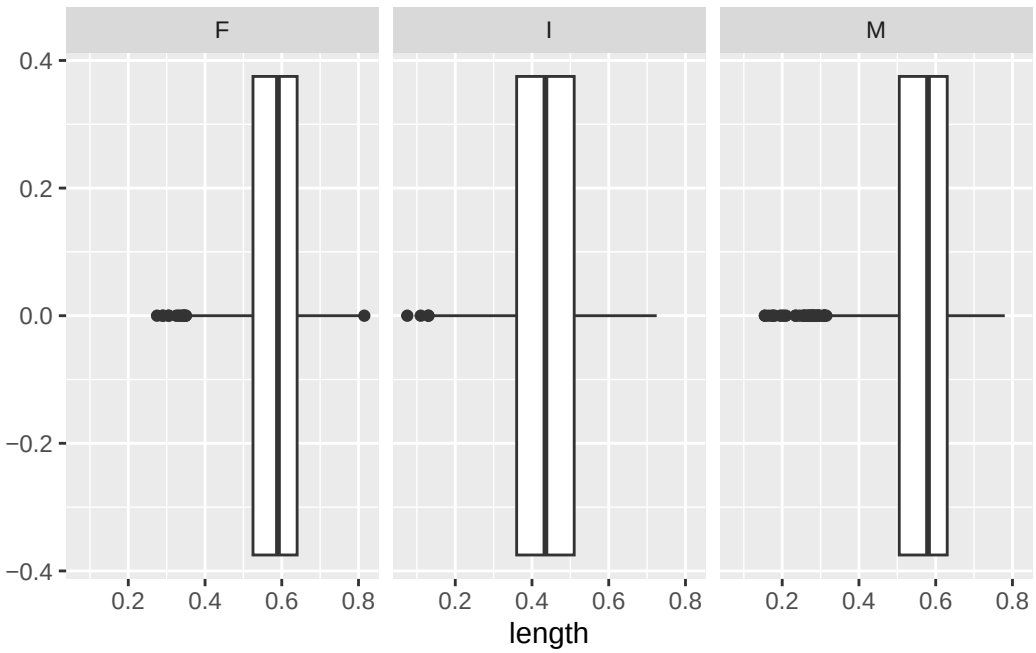
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



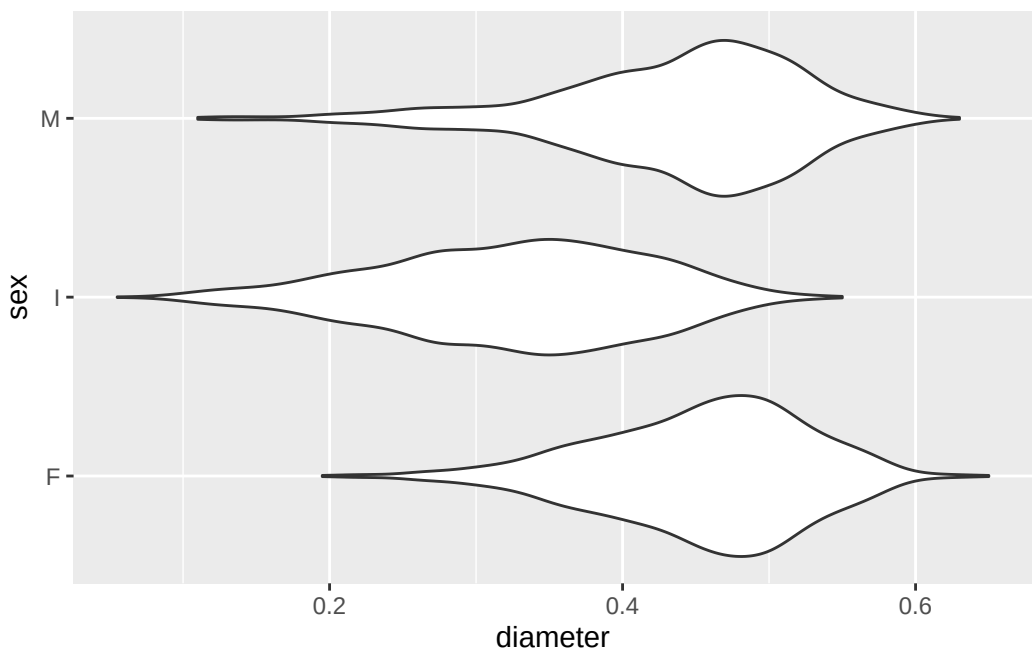
1.4 (5 points)

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?

```
# boxplot
ggplot(df,aes(x=length))+geom_boxplot()+facet_wrap(vars(sex))
```




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

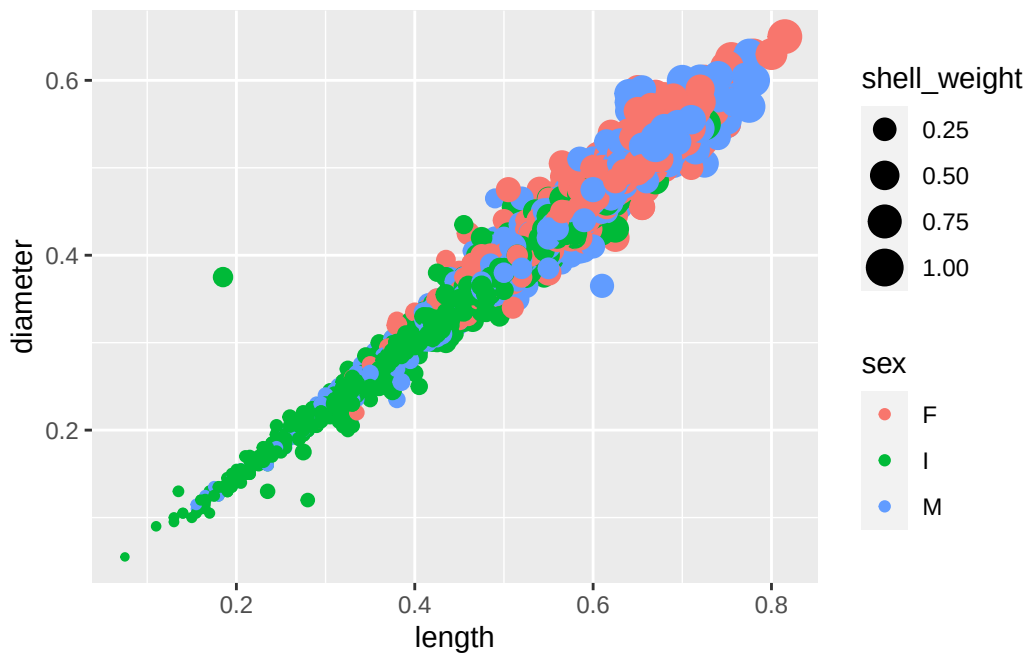


The "I" sex is noticeably smaller in both length and diameter when compared to both males and females. After looking up this dataset, this is a very logical result, as "I" refers to infant abalone, so it makes sense that they would be smaller than adult abalone. The males and females seem to be roughly the same size with large amounts of overlap in their respective plots for both length and 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))+geom_point(aes(color=sex,size=shell_weight))
```



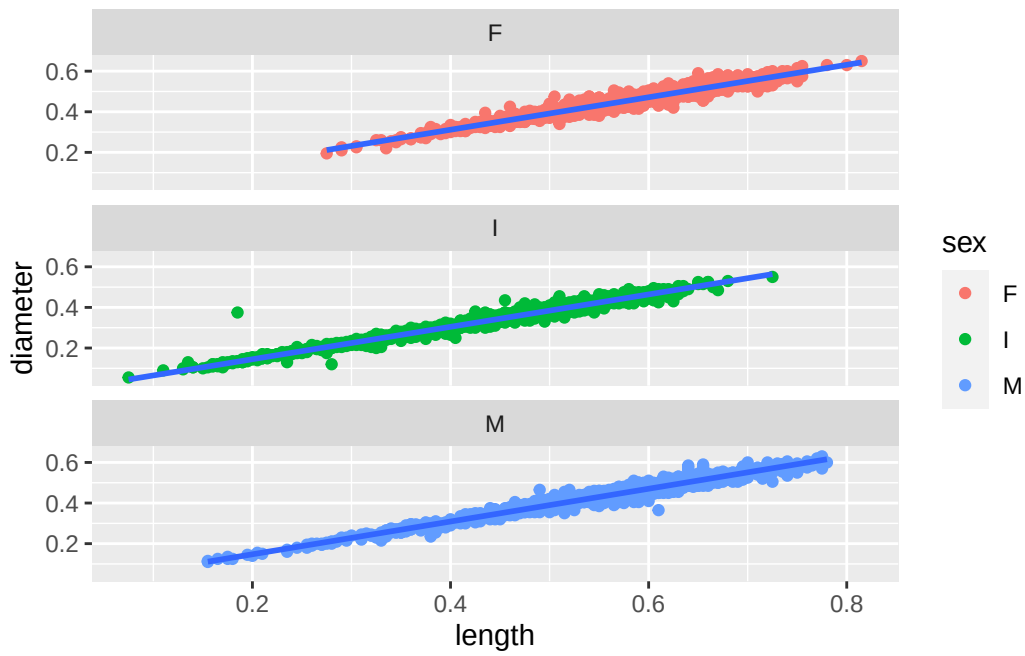
Nearly everything makes sense in this plot. Again, the infant abalone are smaller than the males and females, and larger abalone have greater weight. There seems to be a severe outlier where a single infant abalone is just shy of 0.2 mm in length and 0.4 mm in diameter.

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))+
  geom_smooth(method="lm",se=FALSE)+facet_wrap(vars(sex),nrow=3)
```

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



—

[2]Plot example for 1.6

Question 2

40 points

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

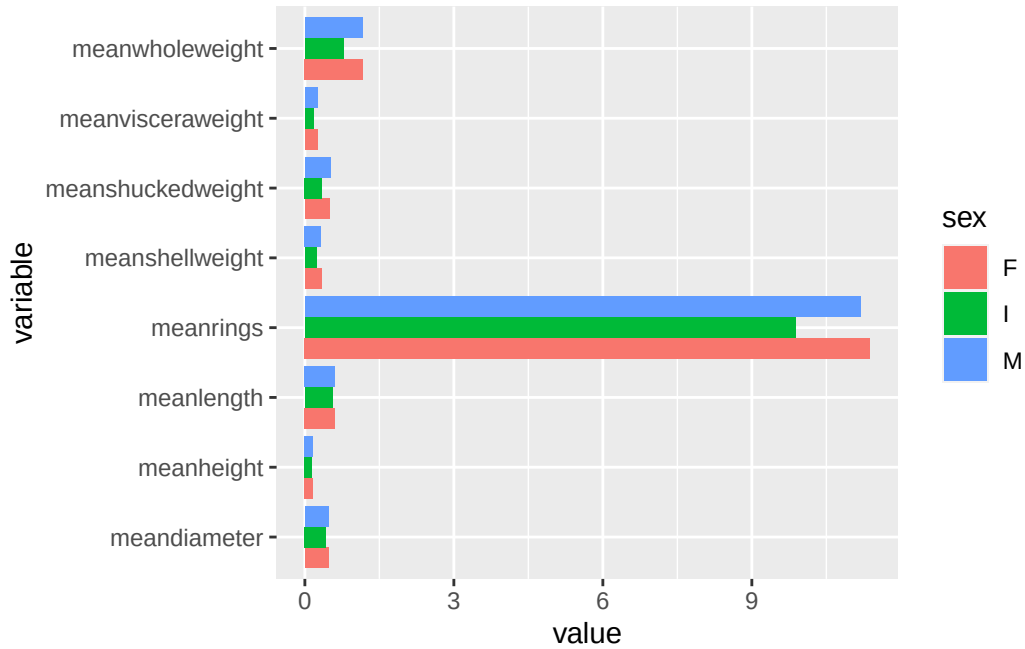
2.1 (10 points)

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`.

```
df21<-df %>% filter(length>=0.5) %>%
  group_by(sex) %>%
  summarize(meanlength=mean(length), meandiameter=mean(diameter),
            meanheight=mean(height), meanwholeweight=mean(whole_weight),
            meanshuckedweight=mean(shucked_weight),
            meanvisceraweight=mean(viscera_weight),meanshellweight=mean(shell_weight),
            meanrings=mean(rings))
df21
```

```
# A tibble: 3 x 9
  sex   meanlength meandiameter meanhe~1 meanw~2 means~3 meanv~4 means~5 meanr~6
  <chr>      <dbl>        <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 F          0.608        0.478    0.165   1.17    0.501   0.258   0.336   11.4
2 I          0.551        0.426    0.142   0.780   0.343   0.167   0.231    9.88
3 M          0.604        0.474    0.163   1.16    0.509   0.252   0.327   11.2
# ... with abbreviated variable names 1: meanheight, 2: meanwholeweight,
#   3: meanshuckedweight, 4: meanvisceraweight, 5: meanshellweight,
#   6: meanrings
```

```
pivot21<-df21 %>% pivot_longer(cols=c(
  meanlength,meandiameter,meanheight,meanwholeweight,
  meanshuckedweight,meanvisceraweight,meanshellweight,meanrings
  ),names_to="variable",values_to="value")
ggplot(pivot21, aes(x=value,y=variable,fill=sex))+geom_col(position = position_dodge())
```



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:

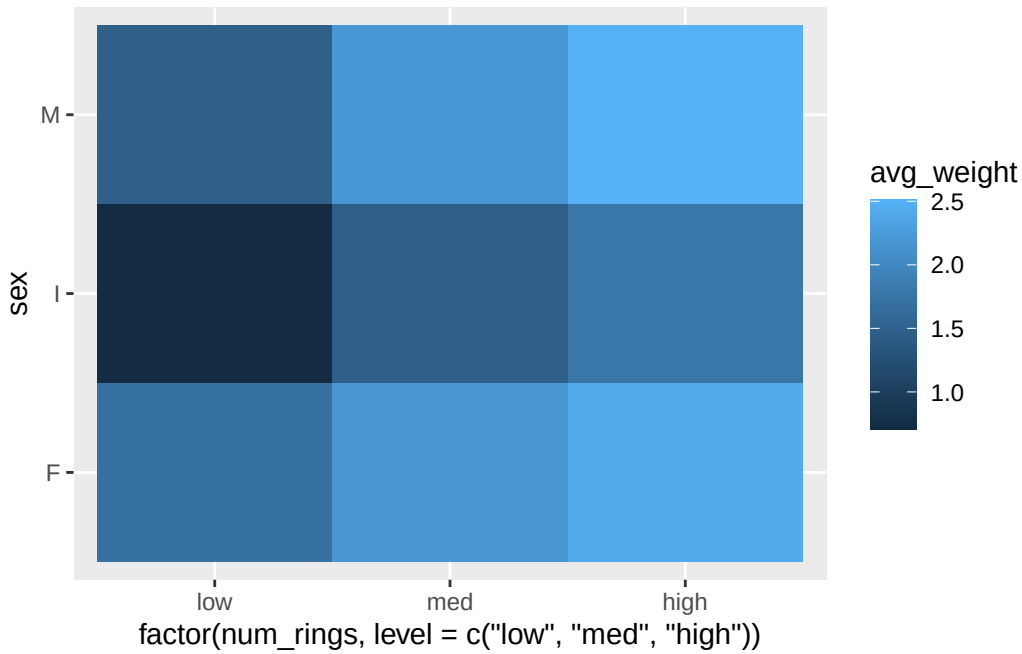
- `"low"` if `rings < 10`
- `"high"` if `rings > 20`, and
- `"med"` otherwise

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(num_rings,sex)%>%
  summarize(avg_weight=mean(whole_weight+shucked_weight+viscera_weight+shell_weight))%>%
  ggplot()+geom_tile(aes(x=factor(num_rings,level=c("low","med","high")),
                         y=sex,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]

```
df %>% select(-sex)%>%
  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

[3] Table for 2.3

```
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.

```
# plotlist <- map2(pivotq$variable, df$rings, lm()) ??? I could not figure out this proble
```

—

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
```

The intercept (beta-0) value is -0.003784. This means that a hypothetical abalone with a diameter of 0 mm would have a predicted height of -0.003784 mm, despite this not being possible.

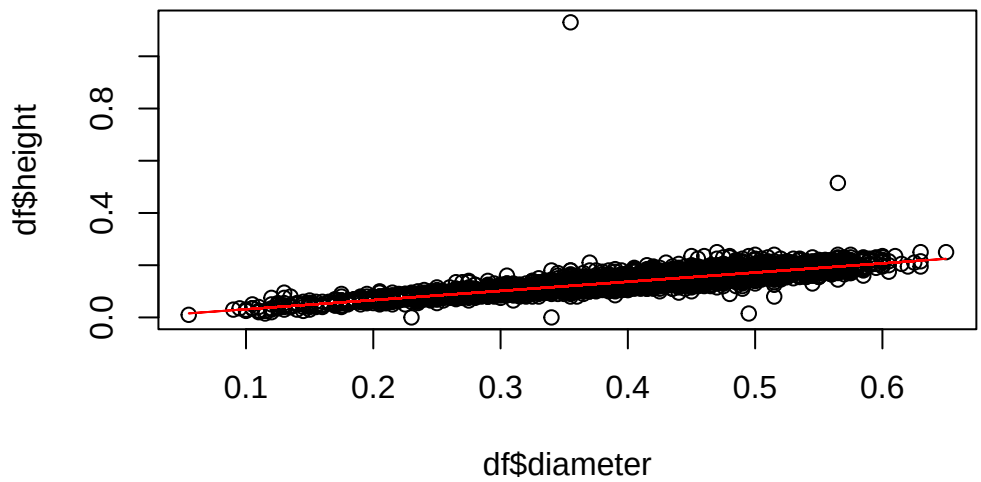
The slope (beta-1) value is 0.351346. This means that for every increase of 1 mm in diameter, the predicted height will increase by 0.351346 mm.

- The p-value for beta-0 is 0.0124. This means that, assuming there is no relation between height and diameter, there would be a 1.24% chance to find results like these or more extreme results from a random sample of abalone. Since this is below the significance level of 0.05, we have evidence that the intercept is significant to the model.
- The p-value for beta-1 is practically zero. This means that, for a model with only the intercept, there would be a practically 0% chance to find results like these or more extreme results from a random sample of abalone. Since this is below the significance level of 0.05, we have very strong evidence that the slope is significant to the model.

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.

```r
plot(df$height~df$diameter)
lines(df$diameter,fitted(lm(df$height~df$diameter)), col="red")
```



- The linear model appears to be a pretty good fit for this data. Most of the points on the scatterplot are pretty close to the regression line, so the line acts as a solid tool for modeling this data.

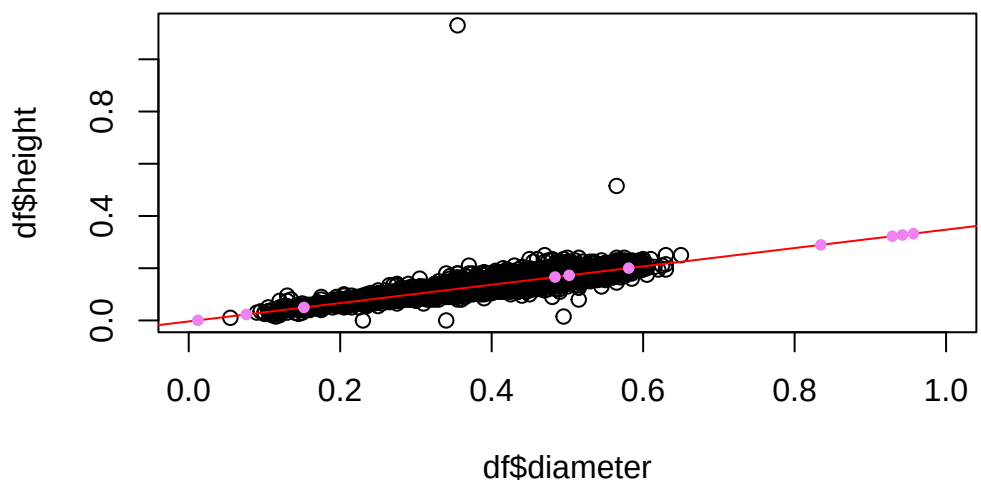
3.3 (10 points)

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"`

```r
new_diameters <- c(
  0.15218946,
  0.48361548,
  0.58095513,
  0.07603687,
  0.50234599,
  0.83462092,
```

```
  0.95681938,
  0.92906875,
  0.94245437,
  0.01209518
)

diameters_frame <- data.frame(new_diameters)
diameters_frame <- diameters_frame %>% rename(diameter=new_diameters)
height <- predict (model, diameters_frame)
plot(df$diameter,df$height,xlim=c(0,1))
abline(model,col="red")
points(x=new_diameters,y=height,pch=20, col="violet")
```




```
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
```

—

Appendix

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 22000)

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] pillar_1.8.1     compiler_4.2.2   tools_4.2.2      digest_0.6.31
 [5] lattice_0.20-45  nlme_3.1-160     gtable_0.3.1     jsonlite_1.8.4
 [9] evaluate_0.20    lifecycle_1.0.3  tibble_3.1.8     mgcv_1.8-41
[13] pkgconfig_2.0.3  rlang_1.0.6      Matrix_1.5-1     cli_3.6.0
[17] DBI_1.1.3        rstudioapi_0.14  yaml_2.3.6       xfun_0.36
[21] fastmap_1.1.0    withr_2.5.0      stringr_1.5.0    knitr_1.41
[25] generics_0.1.3   vctrs_0.5.1      hms_1.1.2        grid_4.2.2
[29] tidyselect_1.2.0 glue_1.6.2       R6_2.5.1         fansi_1.0.3
[33] rmarkdown_2.20   farver_2.1.1     tzdb_0.3.0       magrittr_2.0.3
```

```
[37] splines_4.2.2    scales_1.2.1     ellipsis_0.3.2   htmltools_0.5.4
[41] assertthat_0.2.1 colorspace_2.0-3 renv_0.16.0-53   labeling_0.4.2
[45] utf8_1.2.2       stringi_1.7.12   munsell_0.5.0
```