# PSTAT 131 HW2

Jenny Do (5669841)

April 09, 2022

## Linear Regression

For this lab, we will be working with a data set from the UCI (University of California, Irvine) Machine Learning repository (see website here). The full data set consists of $4,177$ observations of abalone in Tasmania. (Fun fact: Tasmania supplies about 25% of the yearly world abalone harvest.)

The age of an abalone is typically determined by cutting the shell open and counting the number of rings with a microscope. The purpose of this data set is to determine whether abalone age (**number of rings + 1.5**) can be accurately predicted using other, easier-to-obtain information about the abalone.

The full abalone data set is located in the `\data` subdirectory. Read it into *R* using `read_csv()`. Take a moment to read through the codebook (`abalone_codebook.txt`) and familiarize yourself with the variable definitions.

Make sure you load the `tidyverse` and `tidymodels`!

First, we want to read in the data as instructed and load in the necessary packages

```
# Load in dataset using the read.csv() function
abalone <- read.csv(file = 'abalone.csv')
# Look at the first couple of observations to get a feel for the dataset
head(abalone)
```

```
##   type longest_shell diameter height whole_weight shucked_weight viscera_weight
## 1    M         0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2    M         0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3    F         0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4    M         0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5    I         0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6    I         0.425    0.300  0.095       0.3515         0.1410         0.0775
##   shell_weight rings
## 1        0.150    15
## 2        0.070     7
## 3        0.210     9
## 4        0.155    10
## 5        0.055     7
## 6        0.120     8
```

```
# Load in the needed packages
library(ggplot2)
library(tidyverse)
library(tidymodels)
library(corrplot)
tidymodels_prefer()
```

**Question 1**

**Your goal is to predict abalone age, which is calculated as the number of rings plus 1.5. Notice there currently is no age variable in the data set. Add age to the data set.**

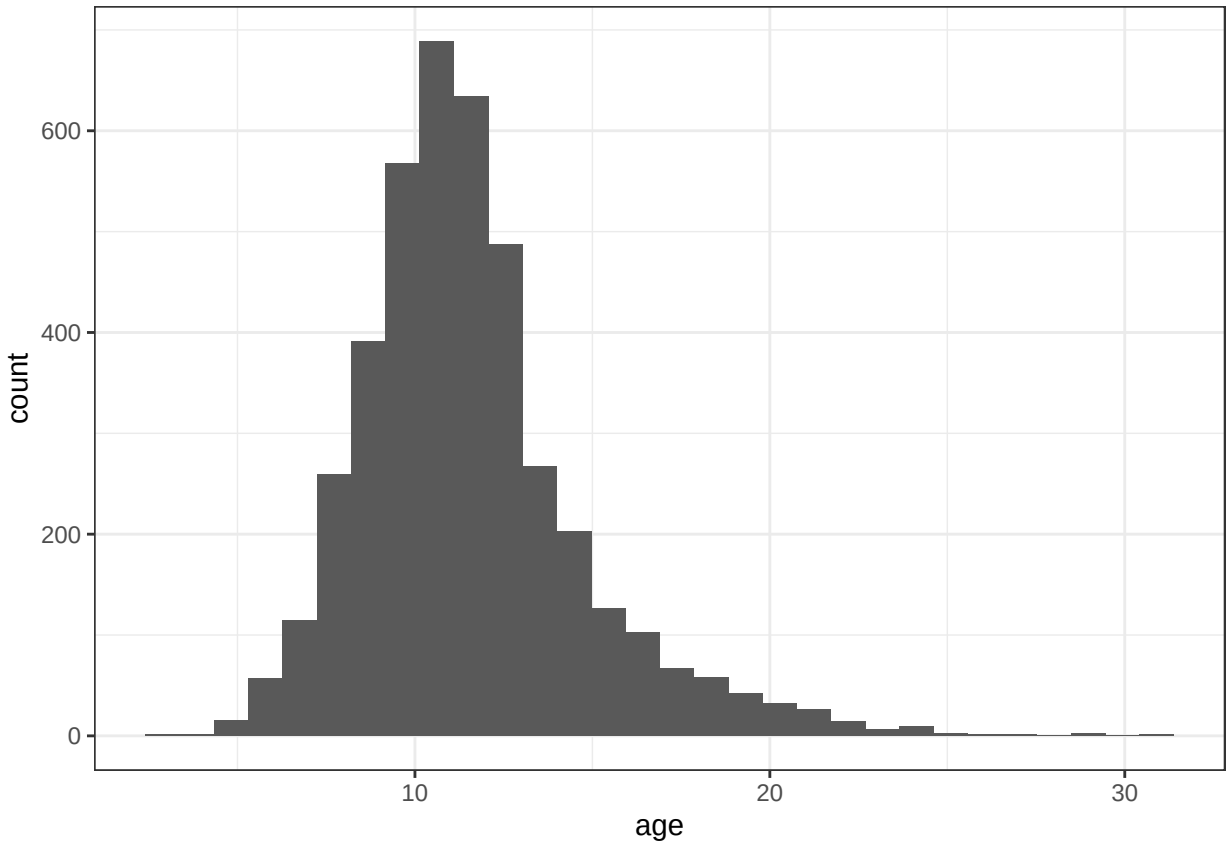
**Assess and describe the distribution of age.**

```
# use the $ operator to create a 'age variable to the dataset'
# note that age is calculated as number of rings plus 1.5
age <- (abalone$rings)+1.5
abalone$age <- age
# use function head() to see that the new variable has been successfully added
head(abalone)
```

```
##   type longest_shell diameter height whole_weight shucked_weight viscera_weight
## 1    M         0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2    M         0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3    F         0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4    M         0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5    I         0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6    I         0.425    0.300  0.095       0.3515         0.1410         0.0775
##   shell_weight rings  age
## 1        0.150    15 16.5
## 2        0.070     7  8.5
## 3        0.210     9 10.5
## 4        0.155    10 11.5
## 5        0.055     7  8.5
## 6        0.120     8  9.5
```

We will be using a histogram to assess the distribution of age.

```
# Use a histogram to assess the distribution of age
abalone %>%
  ggplot(aes(x = age)) +
  geom_histogram(bins = 30 ) +
  theme_bw()
```




**Answer(Question 1):** After plotting the histogram with the variable age (outcome variable), we can see that the histogram is unimodal. In addition, the data is skewed to the right (positively skewed) as much of much of the mass of its distribution is at the lower end, with a long tail to the right. Most of the abalone in the data is less than 20 years old.

### Question 2

**Split the abalone data into a training set and a testing set. Use stratified sampling. You should decide on appropriate percentages for splitting the data.**

*Remember that you'll need to set a seed at the beginning of the document to reproduce your results.*

```
# Need to set seed as the splitting process is random
# choose favorite number 27
set.seed(27)
# Split the data
abalone_split <- initial_split(abalone, prop = 0.80,
                               strata = age)
# training data
abalone_train <- training(abalone_split)
# test data
abalone_test <- testing(abalone_split)
```

**Answer(Question 2):** Note for splitting the data we will be using a 80/20 percentage split (80% for training, 20% for test). We will also be using variable 'age' for stratified sampling as that is what we are trying to predict.

### Question 3

Using the **training** data, create a recipe predicting the outcome variable, `age`, with all other predictor variables. Note that you should not include `rings` to predict `age`. Explain why you shouldn't use `rings` to predict `age`.

Steps for your recipe:

1. dummy code any categorical predictors
2. create interactions between
    - `type` and `shucked_weight`,
    - `longest_shell` and `diameter`,
    - `shucked_weight` and `shell_weight`
3. center all predictors, and
4. scale all predictors.

You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

**Answer(Question 3):** We shouldn't be using 'rings' to predict age because this would result in a flaw in the model. It would lead us to think that we can predict age 100% accurately using rings (as we used rings to create age, age = ring + 1.5).

```
# create a recipe, similar to the lm() function
# - simple recipe to see how creating a recipe works
simple_abalone_recipe <-
  recipe(age ~ type + longest_shell + diameter + height + whole_weight +
      shucked_weight + viscera_weight + shell_weight   , data = abalone_train)
simple_abalone_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
```

```
# Recipe with all needed adjustments - this is the main recipe we are going
# to use:
# Note we are excluding variable 'rings'
## 1. Create the dummy-code all categorical predictors with step functions
abalone_recipe <- recipe(age ~ type + longest_shell + diameter + height +
      whole_weight + shucked_weight + viscera_weight
      + shell_weight, data = abalone_train) %>%
  step_dummy(all_nominal_predictors()) %>%
## 2.  create interactions between -
#Use the step_interact function to create interactions
# interaction between 'type' and 'shucked_weight'
  step_interact(terms = ~ type:shucked_weight) %>%
# interaction between 'longest_shell' and 'diameter'
  step_interact(terms = ~ longest_shell:diameter)%>%
```

```
# interaction between `shucked_weight` and `shell_weight`
  step_interact(terms = ~ shucked_weight:shell_weight)%>%
## 3.  center all predictors - use step_center function
  step_center(all_predictors())%>%
## 4.  scale all predictors - use step_scale function
  step_scale(all_predictors())
abalone_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from all_nominal_predictors()
## Interactions with type:shucked_weight
## Interactions with longest_shell:diameter
## Interactions with shucked_weight:shell_weight
## Centering for all_predictors()
## Scaling for all_predictors()
```

(Reference: https://recipes.tidymodels.org/reference/step_interact.html, https://recipes.tidymodels.org/reference/step_center.html, https://recipes.tidymodels.org/reference/step_scale.html)

### Question 4

Create and store a linear regression object using the `"lm"` engine.

```
# specify the model engine we want to fit
# want linear regression
lm_model_abalone <- linear_reg() %>%
  set_engine("lm")
```

### Question 5

Now:

1. set up an empty workflow,
2. add the model you created in Question 4, and
3. add the recipe that you created in Question 3.

```
# set up empty workflow
lm_wflow_abalone <- workflow() %>%
  # add model from question 4
  add_model(lm_model_abalone) %>%
  # add recipe from question 3
  add_recipe(abalone_recipe)
```

### Question 6

Use your `fit()` object to predict the age of a hypothetical female abalone with longest_shell = 0.50, diameter = 0.10, height = 0.30, whole_weight = 4, shucked_weight = 1, viscera_weight = 2, shell_weight = 1.

```
# fit the linear model to the training set
lm_fit_abalone <- fit(lm_wflow_abalone, abalone_train)
# Viewing the model results
lm_fit_abalone %>%
  # This returns the parsnip object:
  extract_fit_parsnip() %>%
  # Now tidy the linear model object:
  tidy()
```

```
## # A tibble: 12 x 5
##    term                          estimate std.error statistic  p.value
##    <chr>                            <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                     11.5        0.0381   300.    0
##  2 longest_shell                    0.686      0.289      2.37  1.77e- 2
##  3 diameter                         2.55       0.317      8.04  1.24e-15
##  4 height                           0.261      0.0714     3.66  2.60e- 4
##  5 whole_weight                     4.87       0.402     12.1   4.80e-33
##  6 shucked_weight                  -3.85       0.244    -15.8   2.61e-54
##  7 viscera_weight                  -0.977      0.162     -6.05  1.63e- 9
##  8 shell_weight                     1.72       0.219      7.85  5.54e-15
##  9 type_I                          -0.366      0.0546    -6.70  2.40e-11
## 10 type_M                           0.00719    0.0450     0.160 8.73e- 1
## 11 longest_shell_x_diameter        -3.36       0.390     -8.62  1.01e-17
## 12 shucked_weight_x_shell_weight -0.287       0.203     -1.41  1.58e- 1
```

```
# Create a new data frame with all information needed to
# predict the age of a hypothetical female abalone
hypo_female_abalone <- data.frame(type = "F", longest_shell = 0.50, diameter = 0.10, height = 0.30, whol
```

```
# use the predict function with the new data to find the predicted age
predict(lm_fit_abalone, new_data = hypo_female_abalone)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  22.3
```

**Answer(Question 6):** Using the `fit()` object, we predict that a hypothetical female abalone with longest_shell = 0.50, diameter = 0.10, height = 0.30, whole_weight = 4, shucked_weight = 1, viscera_weight = 2, shell_weight = 1 is around 22 years old (22.3282).

### Question 7

Now you want to assess your model's performance. To do this, use the `yardstick` package:

1. Create a metric set that includes $R^2$, RMSE (root mean squared error), and MAE (mean absolute error).

```
# Create a metric set
# rsq denotes R^2
# rmse denotes root mean squared error
# mae denotes mean absolute error
abalone_metrics <- metric_set(rsq,rmse, mae)
```

2. Use `predict()` and `bind_cols()` to create a tibble of your model's predicted values from the **training data** along with the actual observed ages (these are needed to assess your model's performance).

```
# use predict() to generate predicted values for age for each observation in the training set
# use head() to view first six rows of results
abalone_train_res <- predict(lm_fit_abalone, new_data = abalone_train %>% select(-age))
abalone_train_res %>%
  head()
```

```
## # A tibble: 6 x 1
##   .pred
##   <dbl>
## 1  9.30
## 2  8.24
## 3  9.49
## 4 10.0
## 5 10.3
## 6 10.1
```

```
# use bind_cols to attach a column with the actual observed age observations
# again use head() to view first six rows of results
abalone_train_res <- bind_cols(abalone_train_res, abalone_train %>% select(age))
abalone_train_res %>%
  head()
```

```
## # A tibble: 6 x 2
##   .pred   age
##   <dbl> <dbl>
## 1  9.30   8.5
## 2  8.24   8.5
## 3  9.49   9.5
## 4 10.0    8.5
## 5 10.3    8.5
## 6 10.1    9.5
```

3. Finally, apply your metric set to the tibble, report the results, and interpret the $R^2$ value.

```
# create and view a "metric set" of R^2, RMSE and MSE
# note that R^2 favors flexible methds, which may overfit the data
abalone_metrics(abalone_train_res, truth = age,
                estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
```

```
## 1 rsq     standard       0.541
## 2 rmse    standard       2.20
## 3 mae     standard       1.58
```

**Answer(Question 7):** After applying the metric set to the tibble, we find that the root mean square error (RMSE) is 2.1996 and the mean absolute error (MAE) is 1.5827. In addition, the $R^2$ value is 0.5410. $R^2$ by definition is the proportion of the variability in Y that can be explained using X. An $R^2$ of 1.0 indicates that the data perfectly fits the linear model while an $R^2$ of 0 indicates a total lack of fit. Therefore, an $R^2$ of 0.5410 indicates that about 46% (0.4590) of the variability in the outcome data cannot be explained by the model while about 54% (0.5410) of the variability in the outcome data can be explained by the model.

(Reference for the entirety of HW2: Lab 2 and Section)