HW2

Yiting Zhang

2022-04-10

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`!

```
library(tidyverse)
library(dplyr)
library(tidymodels)
library(ISLR)
library(yardstick)
tidymodels_prefer()
```

```
abalone <- read.csv("abalone.csv")
```

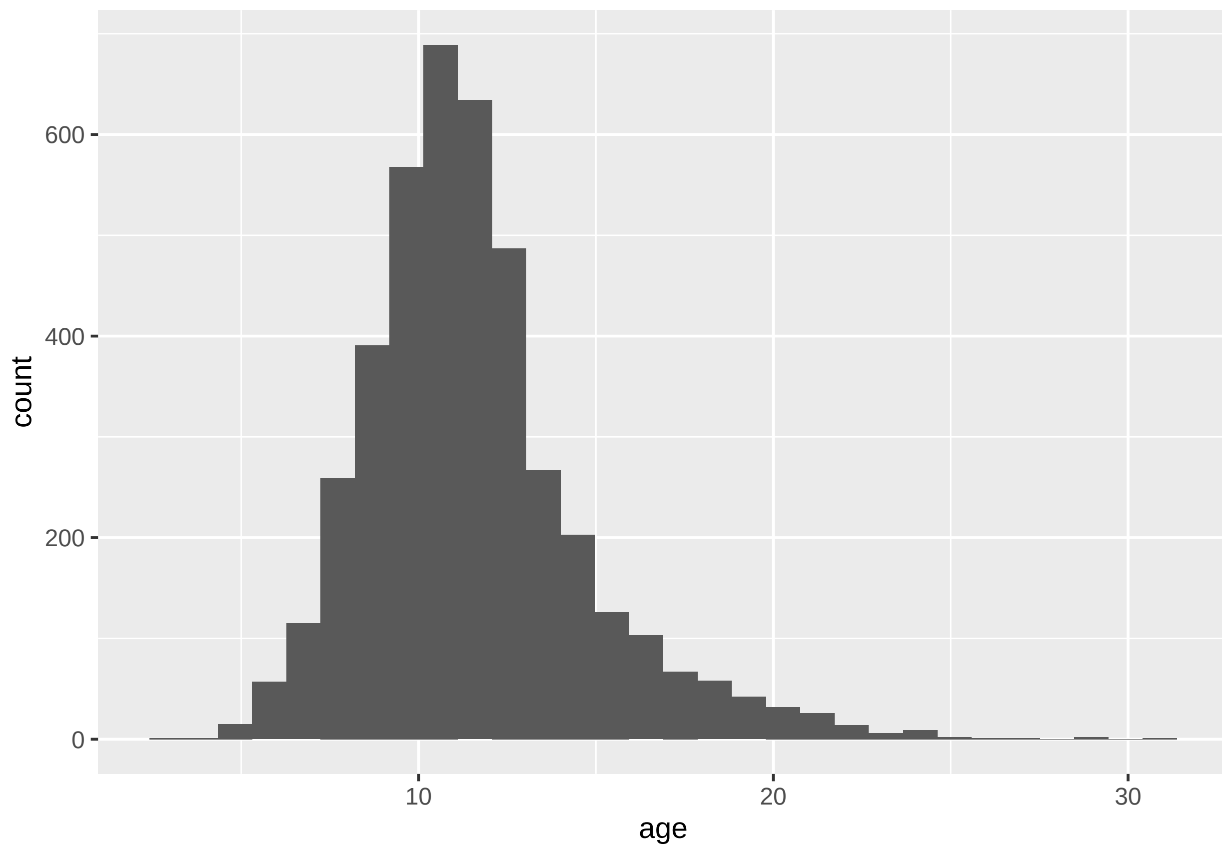
Question 1

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`.

```
abalone <- abalone%>%
  mutate(age = abalone$rings + 1.5)

ggplot(data=abalone, aes(age)) + geom_histogram()
```



By plotting a histogram, it seems like the abalone age data follows a normal distribution which skewed to the right a little bit with center around 10.

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.

```
set.seed(100)
abalone_split <- initial_split(abalone, prop = 0.80, strata = age)
abalone_train <- training(abalone_split)
abalone_test <- testing(abalone_split)
```

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.

```
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()) %>%
  step_interact(terms = ~ type:shucked_weight) %>%
  step_interact(terms = ~ longest_shell:diameter) %>%
  step_interact(terms = ~ shucked_weight:shell_weight) %>%
  step_center(all_predictors()) %>%
  step_normalize() %>%
  step_scale()
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()
## Centering and scaling for <none>
## Scaling for <none>
```

Rings cannot be used to predict age because it is used in the formula for the age variable. If it is used, then the prediction will be 100% correct.

Question 4

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

```
lm_model <- 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.

```
lm_wflow <- workflow() %>%
  add_model(lm_model) %>%
  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.

```
lm_fit <- lm_wflow %>% fit(abalone_train)

Q6_data <- data.frame(type="F",longest_shell = 0.50, diameter = 0.10,
                      height = 0.30, whole_weight = 4, shucked_weight = 1,
                      viscera_weight = 2, shell_weight = 1,stringsAsFactors = TRUE)


prediction <-predict(lm_fit, new_data = Q6_data)

prediction
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  19.8
```

19.84488 years old by prediction.

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

```
abalone_train_res <- bind_cols(predict(lm_fit, abalone_train), abalone_train$age)

abalone_metrics <- metric_set(rsq, rmse, mae)
abalone_metrics(abalone_train_res, truth = ...2,
                estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.550
## 2 rmse    standard       2.17
## 3 mae     standard       1.56
```

The R^2 value is 0.5498393, which means that the linear regression model we made explains 54.98393% of the variation in age variable.