Homework 2

Vardan Martirosyan

2022-10-13

First, we read in the data.

```
dataset <- read.csv("/Users/vardan/Desktop/pstat131/Homework/Homework2/data/abalone.csv")
```

Then, we load in the tidyverse and tidymodels libraries as desired. Additionally, we also load the 'ggplot2' library, which can help with some of the questions asked of us.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.1     v rsample      1.1.0
## v dials        1.0.0     v tune         1.0.0
## v infer        1.0.3     v workflows    1.1.0
## v modeldata    1.0.1     v workflowsets 1.0.0
## v parsnip      1.0.1     v yardstick    1.1.0
## v recipes      1.0.1
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```
library(ggplot2)
```

Question 1

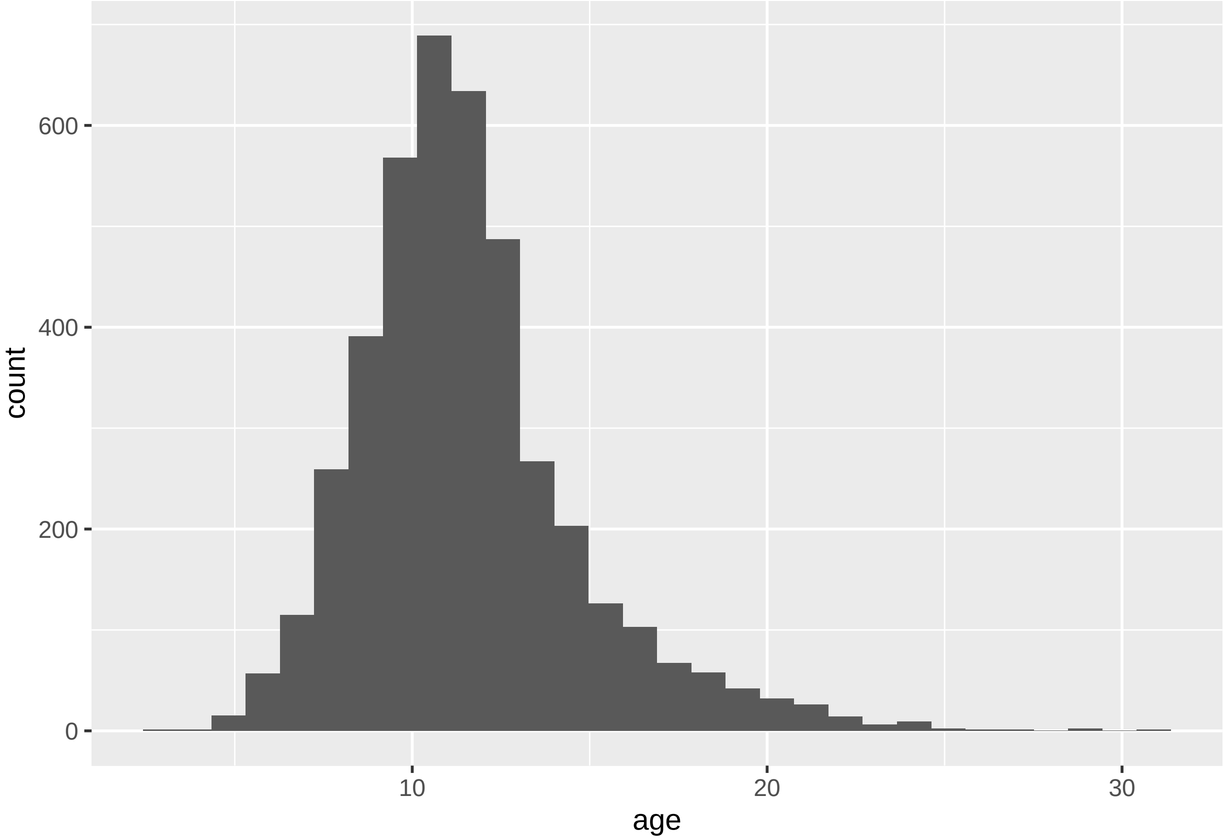
Our goal is to add ‘age’ to the dataset. We do this as follows:

```
new_dataset <- dataset %>%
  mutate(
    age = rings + 1.5
  )
```

We have added the age column to the dataset, as desired. We are then asked to assess and describe the distribution of the ‘age’ variable. To do this, we can create a histogram plot of the ‘age’ variable and examine it. We do this as follows:

```
ggplot(new_dataset, aes(x = age)) + geom_histogram()
```

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



Looking at this plot, we see that ‘age’ is distributed approximately normally, slightly skewed to the right, with a mean at x = 11. Additionally, we can see that it has some outliers around x = 0 and x = 20-30, but most of the values are concentrated between x = 5 and x = 22.

Question 2

We are then asked to split the data into a training and a testing set using stratified sampling, and that we should decide on appropriate percentages for splitting the data. From Lab 2, we recall that we chose to do the split 80-20: that is, 80 percent of the data will be put into the training set, and 20 percent will be put into the testing set. This seems like an appropriate percentage to me, so it is one that we will use. We then set the seed, and split the data, as follows:

```
set.seed(69)


new_dataset_split <- initial_split(new_dataset, prop = 0.80, strata = age)

new_dataset_train <- training(new_dataset_split)
new_dataset_test <- testing(new_dataset_split)
```

Question 3

We are asked to create a recipe predicting the outcome variable, age, with all other predictor variables. We are asked to state why we shouldn't use rings to predict age. We do not want to use the variables ‘rings’ to predict ‘age’ because the variables ‘age’ and ‘rings’ are collinear by construction. We recall that ‘age’ is literally the variable ‘rings’, with 1.5 added. This means that the two variables have a linear relationship, which would cause problems if we try to use the variable ‘rings’ as a predictor. In particular, it may lead to our model being overfit, and/or the variance being inflated past it's true value. Thus, this is why we should not use the ‘rings’ variable to predict ‘age’. We then code the recipe to predict the outcome variable ‘age’ as desired:

```
#First, we create the recipe, removing the predictor 'rings'.

#Then, we want to code the dummy variables for any categorical predictors.
#We note that 'type', which indicates the sex, is the only categorical predictor.

#Now, we want to create interactions between three different variables.

#Finally, we normalize and center all predictors, as is asked of us.

#All of these steps are below as follows:

new_dataset_train <- new_dataset_train %>% select(-rings)

age_recipe <-
  recipe(age ~ ., data = new_dataset_train) %>%
  step_dummy('type') %>%
  step_interact( ~ starts_with("type"):shucked_weight) %>%
  step_interact( ~ longest_shell:diameter) %>%
  step_interact( ~ shucked_weight:shell_weight) %>%
  step_normalize(all_predictors()) %>%
  step_center(all_predictors())
```

Question 4

We are asked to create and store a linear regression object using the “lm” engine.

```
lm_model <- linear_reg() %>%
  set_engine("lm")
```

Question 5

We are now asked to set up an empty workflow, add the model we created in Q4, and the recipe we created in Q3. We do this as follows:

```
lm_wflow <- workflow() %>%
  add_model(lm_model) %>%
  add_recipe(age_recipe)
```

Question 6

We are now asked to use our fit() object to predict the age of a hypothetical female abalone with several given values for the predictors.

```
#First, let us fit the linear model according to our training set.
lm_fit <- fit(lm_wflow, new_dataset_train)

#Then, let us view the results of this.
results <- lm_fit %>%
  # This will return the parsnip object.
  extract_fit_parsnip() %>%
  # Now we tidy the linear model object.
  tidy()

results
```

```
## # A tibble: 14 x 5
##    term                          estimate std.error statistic  p.value
##    <chr>                            <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                    11.4       0.0376   304.    0
##  2 longest_shell                   0.492     0.292      1.69  9.20e- 2
##  3 diameter                        2.31      0.322      7.17  9.28e-13
##  4 height                          0.252     0.0705     3.57  3.57e- 4
##  5 whole_weight                    4.96      0.402     12.4   2.64e-34
##  6 shucked_weight                 -4.27      0.257    -16.6   1.25e-59
##  7 viscera_weight                 -0.855     0.158     -5.42  6.53e- 8
##  8 shell_weight                    1.59      0.217      7.32  3.13e-13
##  9 type_I                         -0.962     0.117     -8.25  2.22e-16
## 10 type_M                         -0.228     0.104     -2.18  2.90e- 2
## 11 type_I_x_shucked_weight         0.548     0.0878     6.24  4.90e-10
## 12 type_M_x_shucked_weight         0.255     0.111      2.31  2.12e- 2
## 13 longest_shell_x_diameter       -3.10      0.400     -7.75  1.18e-14
## 14 shucked_weight_x_shell_weight  -0.0414    0.202     -0.205 8.38e- 1
```

Then, we use the predict() function, along with the lm_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.

```r
input <- data.frame(type = 'F', longest_shell = 0.5, diameter = 0.10,
                    height = 0.30, whole_weight = 4.0,
                    shucked_weight = 1.0, viscera_weight = 2.0,
                    shell_weight =  1.0)

predicted_age <- predict(lm_fit, new_data = input)


predicted_age
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  25.2
```

Thus, the predicted age of this female abalone is 25.18972.

Question 7

We then want to assess our model's performance using the 'yardstick' package.

1. We are first asked to create a metric set that includes R^2, RMSE, and MAE. We do this as follows:

```r
dataset_metrics <- metric_set(rmse, rsq, mae)
```

2. We are then asked to se predict() and bind_cols() to create a tibble of your model's predicted values from the training data along with the actual observed ages.

```r
new_dataset_train_res <- predict(lm_fit, new_data = new_dataset_train %>% select(-age))
new_dataset_train_res <- bind_cols(new_dataset_train_res, new_dataset_train %>% select(age))
```

3. We are finally asked to apply our metric set to the tibble, report the results, and interpret the R2 value.

```r
dataset_metrics(new_dataset_train_res, truth = age, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.17
## 2 rsq     standard       0.557
## 3 mae     standard       1.55
```

From this, we can see that the Root Mean Squared Error is equal to 2.1672959, the R^2 value is equal to 0.5574355, and the mean absolute error is equal to 1.5544993. We interpret the R^2 value as follows: Approximately 55.74 percent of the variability in the 'age' response variable is explained by the linear regression model.