HW2

Ben Hertzberg

2022-10-05

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(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.2.0 --

## v broom        0.8.0     v recipes      0.2.0
## v dials        0.1.1     v rsample      0.1.1
## v dplyr        1.0.8     v tibble       3.1.6
## v ggplot2      3.3.5     v tidyr        1.2.0
## v infer        1.0.0     v tune         0.2.0
## v modeldata    0.1.1     v workflows    0.2.6
## v parsnip      0.2.1     v workflowsets 0.2.1
## v purrr        0.3.4     v yardstick    0.0.9

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v readr   2.1.2     v forcats 0.5.1
## v stringr 1.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
```

```
## x dplyr::filter()      masks stats::filter()
## x stringr::fixed()     masks recipes::fixed()
## x dplyr::lag()         masks stats::lag()
## x readr::spec()        masks yardstick::spec()
```

```
library(ggplot2)
```

```
abalone <- read.csv('./data/abalone.csv')
```

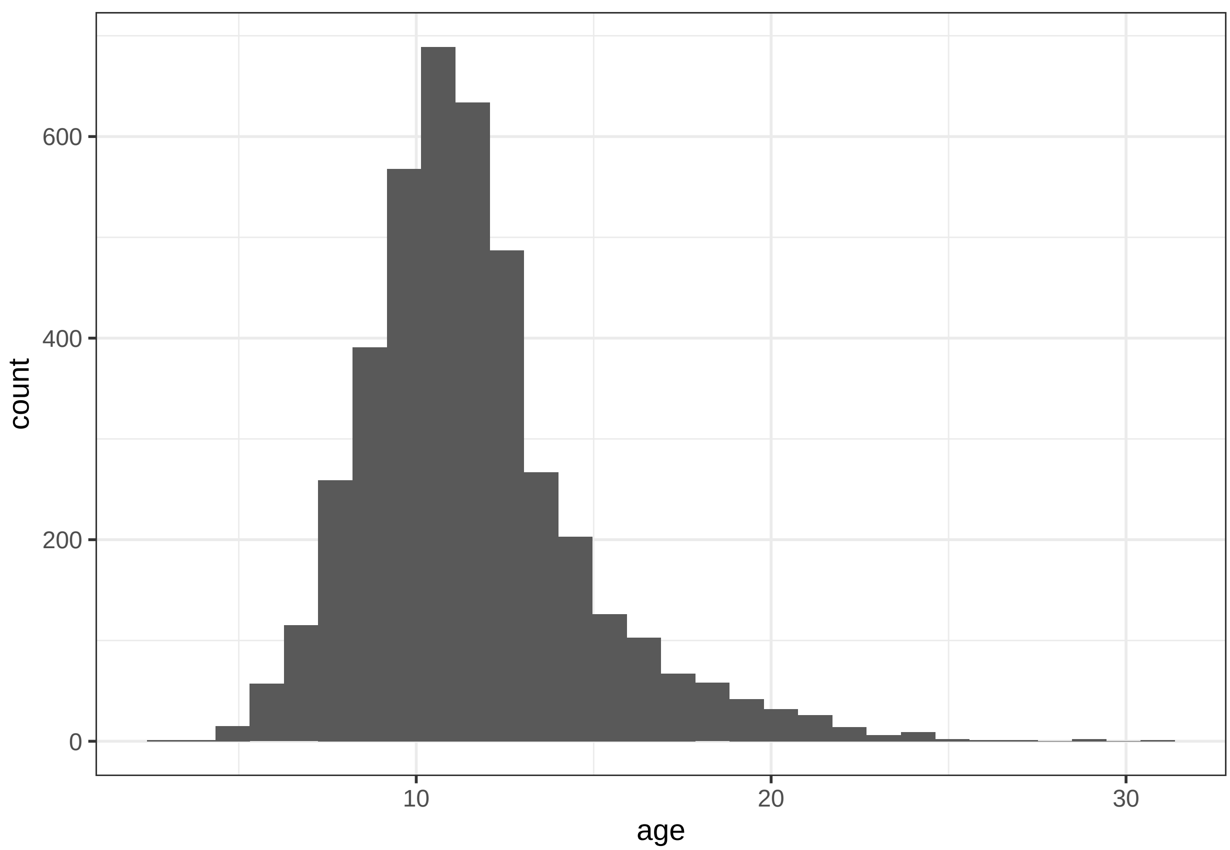
Question 1

The age data is unimodally centered around 11 and skewed right. It ranges from 2.5 to 30.50 years. The median is 10.50 years and the mean is 11.43 years.

```
abalone$age <- abalone$rings+1.5

abalone %>%
  ggplot(aes(x = age)) +
  geom_histogram() +
  theme_bw()
```

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



```r
summary(abalone$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.50    9.50   10.50   11.43   12.50   30.50
```

Question 2

```r
set.seed(619)

ab_split <- initial_split(abalone, prop = 0.80,
                                strata = age)
ab_train <- training(ab_split)
ab_test <- testing(ab_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.

```r
abRecipe <- recipe(age ~ type + longest_shell + diameter + height + whole_weight + shucked_weight + vis
  step_dummy(all_nominal_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%


  step_interact(terms = ~ starts_with("type"):shucked_weight) %>%
  step_interact(terms = ~ longest_shell:diameter) %>%
  step_interact(terms = ~ shucked_weight:shell_weight)
```

Question 4

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

```r
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.

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

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.

```r
lm_fit <- fit(lm_wflow, ab_train)
```

```r
vals <- data.frame(type = c('F'), longest_shell = c(0.5), diameter=c(0.1), height =c(0.3), whole_weight=

predict(lm_fit, vals)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  24.4
```

The predicted age for the values listed above is 24.36 years.

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.

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

ab_tr_res <- predict(lm_fit, new_data = ab_train %>% select(-age))

ab_tr_res <- bind_cols(ab_tr_res, ab_train %>% select(age))

ab_metrics(ab_tr_res, truth = age,
                estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.550
## 2 rmse    standard       2.15
## 3 mae     standard       1.54
```

Rsq = 0.550, RMSE = 2.147, MAE = 1.541 The Rsq = 0.550 value informs us that about 55% of the variability in abalone age can be explained by the predictor variables used in the model above.

Required for 231 Students

In lecture, we presented the general bias-variance tradeoff, which takes the form:

$$E[(y_0 - \hat{f}(x_0))^2] = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

where the underlying model $Y = f(X) + \epsilon$ satisfies the following:

- ϵ is a zero-mean random noise term and X is non-random (all randomness in Y comes from ϵ);
- (x_0, y_0) represents a test observation, independent of the training set, drawn from the same model;
- $\hat{f}(.)$ is the estimate of f obtained from the training set.

Question 8 Which term(s) in the bias-variance tradeoff above represent the reproducible error? Which term(s) represent the irreducible error?

Question 9 Using the bias-variance tradeoff above, demonstrate that the expected test error is always at least as large as the irreducible error.

Question 10 Prove the bias-variance tradeoff.

Hints:

- use the definition of $Bias(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0)$;
- reorganize terms in the expected test error by adding and subtracting $E[\hat{f}(x_0)]$