Homework 2

PSTAT 131/231

Contents

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`!

```r
library(tidyverse)
library(tidymodels)
```

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.

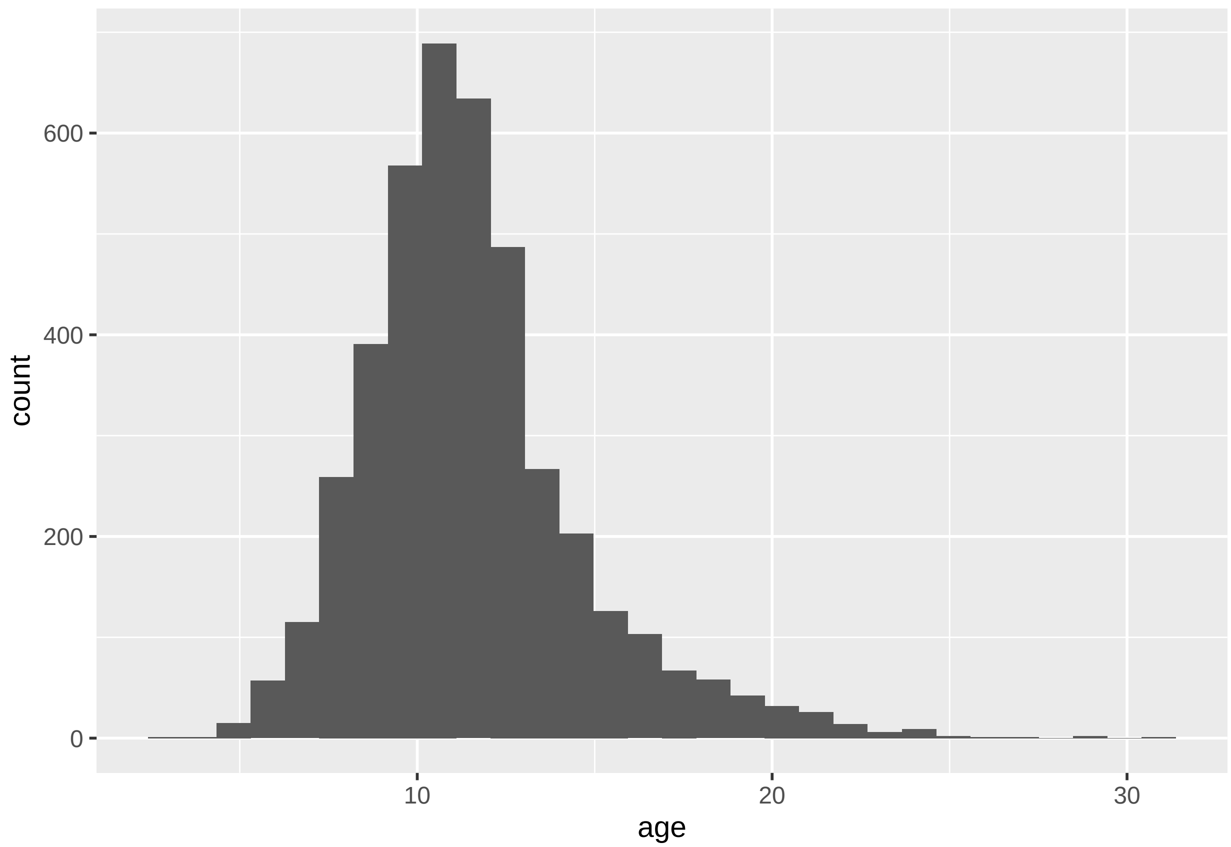
Assess and describe the distribution of `age`.

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

age<-abalone$rings+1.5

abalone<-abalone %>% mutate(age)

abalone %>% ggplot(aes(x = age)) +
  geom_histogram(bins = 30)
```



The histgram of `age` is positively skewed and the data forms a bell curve skewed to the right. The distribution is centered around 10, and most of abalone have an age around 10 or 12, with a few having an age above 20.

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.

```
set.seed(12345)
abalone_split <- initial_split(abalone, prop = 0.8, strata = age)

abalone_train <- training(abalone_split)

abalone_test <- testing(abalone_split)
```

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

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_train<-abalone_train[,-9]

abalone_recipe_1 <- recipe(age ~ ., data = abalone_train) %>%
  step_dummy(all_nominal_predictors())

abalone_recipe<-abalone_recipe_1 %>%
  step_interact(terms = ~ shucked_weight:starts_with("type")) %>%
  step_interact(terms = ~ longest_shell:diameter) %>%
  step_interact(terms = ~ shucked_weight:shell_weight)  %>%
  step_center(all_numeric(), -all_outcomes(), -has_role('id variable')) %>%
  step_scale(all_numeric(), -all_outcomes(), -has_role('id variable'))

abalone_train_1<-abalone_recipe %>%
  prep() %>%
  bake(new_data = abalone_train)
```

We shouldn't use rings to predict age. Because it's in the formula for the age variable. We need find the appropriate step functions by investigate the tidymodels documentation.

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)

lm_wflow
```

```
## == Workflow ====================================================================
## Preprocessor: Recipe
## Model: linear_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 6 Recipe Steps
##
## * step_dummy()
## * step_interact()
## * step_interact()
## * step_interact()
## * step_center()
## * step_scale()
##
## -- Model -----------------------------------------------------------------------
## Linear Regression Model Specification (regression)
##
## Computational engine: lm
```

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.

```
data<-data.frame(type="F",longest_shell=0.5,diameter = 0.10, height = 0.30, whole_weight = 4,
                 shucked_weight = 1, viscera_weight = 2, shell_weight = 1)

data<-tibble(data)

lm_fit <- fit(lm_wflow, abalone_train)

predict(lm_fit, new_data = data)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  23.3
```

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.

```
library(yardstick)

#1
```

```
abalone_metrics <- metric_set(rmse, rsq, mae)

#2
abalone_train_res <- predict(lm_fit, new_data = abalone_train %>% select(-age))
abalone_train_res <- bind_cols(abalone_train_res, abalone_train %>% select(age))

#3
abalone_metrics(abalone_train_res, truth = age, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.16
## 2 rsq     standard       0.549
## 3 mae     standard       1.55
```

The rsq value is 0.549. Which mean in `age` with `type` + `longest_shell` + `diameter` + `height` + `whole_weight` + `shucked_weight` + `viscera_weight` + `shell_weight`. 54.9% of `age` can be determined by explain these variables.