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`!

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.

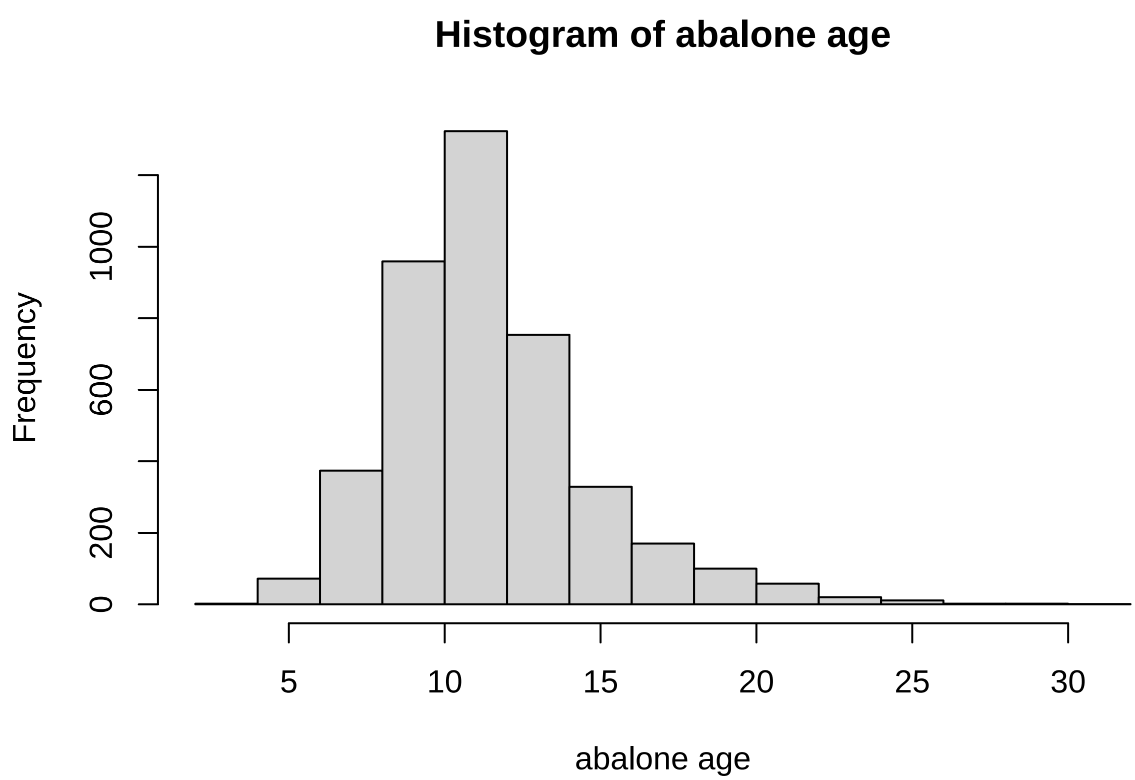
Assess and describe the distribution of `age`.

```
library(tidyverse)
library(tidymodels)

# read in the data
abalone <- read_csv("C:/Users/cupca/OneDrive/Documents/UCSB/Fall2022/PSTAT131/homework-2/data/abalone.cs

# add age as a variable
abalone_w_age <- abalone %>% mutate(age = rings + 1.5)
abalone_w_age
```

```
## # A tibble: 4,177 x 10
##    type  longest_sh~1 diame~2 height whole~3 shuck~4 visce~5 shell~6 rings   age
##    <chr>        <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl>
##  1 M            0.455   0.365  0.095   0.514  0.224   0.101     0.15    15  16.5
##  2 M            0.35    0.265  0.09    0.226  0.0995  0.0485    0.07     7   8.5
##  3 F            0.53    0.42   0.135   0.677  0.256   0.142     0.21     9  10.5
```

```
##   4 M             0.44    0.365  0.125   0.516  0.216   0.114    0.155    10  11.5
##   5 I             0.33    0.255  0.08    0.205  0.0895  0.0395   0.055     7   8.5
##   6 I             0.425   0.3    0.095   0.352  0.141   0.0775   0.12      8   9.5
##   7 F             0.53    0.415  0.15    0.778  0.237   0.142    0.33     20  21.5
##   8 F             0.545   0.425  0.125   0.768  0.294   0.150    0.26     16  17.5
##   9 M             0.475   0.37   0.125   0.509  0.216   0.112    0.165     9  10.5
##  10 F             0.55    0.44   0.15    0.894  0.314   0.151    0.32     19  20.5
## # ... with 4,167 more rows, and abbreviated variable names 1: longest_shell,
## #   2: diameter, 3: whole_weight, 4: shucked_weight, 5: viscera_weight,
## #   6: shell_weight
```

```
# simple histogram of abalone age
hist(abalone_w_age$age,
     xlab = "abalone age",
     main = "Histogram of abalone age")
```



Answer: After making a histogram of the abalone age variable, it seems that age is approximately normally distributed, showing a bell-shaped curve.

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.

```r
set.seed(3435)

# preparing data for splitting
abalone_w_age <- abalone_w_age %>%
  select(-rings) %>% # remove rings variable
  mutate(type=as.factor(type)) # convert type variable to factor type

# split into training/testing sets
abalone_split <- initial_split(abalone_w_age, 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.

```r
# recipe (do not include rings)
abalone_recipe <- recipe(age~., data = abalone_train) %>%
  step_dummy(all_nominal_predictors()) %>% # dummy code categorical predictors
  step_interact(terms = ~ starts_with("type"):shucked_weight) %>% # create interactions
  step_interact(terms = ~ longest_shell:diameter) %>%
  step_interact(terms = ~ shucked_weight:shell_weight) %>%
  step_center(all_predictors()) %>% # center predictors
  step_scale(all_predictors()) # scale predictors
```

Answer: We should not include rings to predict age because age is calculated based on rings (rings + 1.5). If we used rings, it would be like using a scaled version of the outcome as a predictor to help predict the outcome, which doesn't make much sense. More importantly, the rings variable and the age variable are collinear, and that would cause problems when we're trying to figure out which predictor variables have what effect on the outcome variable.

Question 4

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

```r
# create model, store linear regression object
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() %>% # empty workflow
  add_model(lm_model) %>%  # add model
  add_recipe(abalone_recipe) # add recipe

lm_fit <- fit(lm_wflow, abalone_train) # see how our fitted model did

lm_fit %>%
  # This returns the parsnip object:
  extract_fit_parsnip() %>%
  # Now tidy the linear model object:
  tidy()
```

```
## # A tibble: 14 x 5
##    term                          estimate std.error statistic  p.value
##    <chr>                            <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                   11.4        0.0375  305.     0
##  2 longest_shell                  0.591      0.286     2.07   3.86e- 2
##  3 diameter                       2.06       0.313     6.61   4.59e-11
##  4 height                         0.236      0.0696    3.39   7.10e- 4
##  5 whole_weight                   4.29       0.387    11.1    4.66e-28
##  6 shucked_weight                -4.06       0.250   -16.2    5.35e-57
##  7 viscera_weight                -0.792      0.158    -5.00   6.12e- 7
##  8 shell_weight                   1.74       0.212     8.20   3.32e-16
##  9 type_I                        -0.942      0.117    -8.07   9.36e-16
## 10 type_M                        -0.239      0.104    -2.29   2.21e- 2
## 11 type_I_x_shucked_weight        0.525      0.0876    5.99   2.26e- 9
## 12 type_M_x_shucked_weight        0.293      0.109     2.68   7.41e- 3
## 13 longest_shell_x_diameter      -2.75       0.396    -6.95   4.32e-12
## 14 shucked_weight_x_shell_weight -0.00330    0.205    -0.0161 9.87e- 1
```

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
# new observation to predict
new_abalone <- data.frame(type=as.factor("F"),
                          longest_shell=0.50,
                          diameter=0.10,
                          height=0.30,
                          whole_weight=4,
                          shucked_weight=1,
```

```
                           viscera_weight=2,
                           shell_weight=1)
new_abalone <- as_tibble(new_abalone) # convert to tibble

abalone_train_res <- predict(lm_fit, new_data = new_abalone) # use model to predict
abalone_train_res
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  23.7
```

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.

```
# metric set
abalone_metrics <- metric_set(rmse, rsq, mae)

# tibble of predicted values
abalone_train_res <- predict(lm_fit, new_data = abalone_train %>% select(-age))
abalone_train_res %>%
  head()
```

```
## # A tibble: 6 x 1
##   .pred
##   <dbl>
## 1  8.03
## 2  9.68
## 3 10.4
## 4 10.1
## 5 10.9
## 6  6.26
```

```
# add actual observed ages
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  8.03   8.5
## 2  9.68   8.5
## 3 10.4    8.5
```

```
## 4 10.1    9.5
## 5 10.9    9.5
## 6  6.26   6.5
```

```r
# apply metric set to tibble
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       0.551
## 3 mae     standard       1.55
```

Answer: R^2 measures the proportion of variability in the outcome Y (age of the abalone), that can be explained through the predictor X (longest_shell, diameter, height, whole_weight, shucked_weight, viscera_weight, and shell_weight). So according to the metric set, it seems that there is about 0.55 variability in abalone age that can be accounted for by our predictors.