Code ▾

Homework 2

PSTAT 131/231

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 (http://archive.ics.uci.edu/ml/datasets/Abalone)). The full data set consists of $4,177$ observations of abalone in Tasmania. (Fun fact: Tasmania (https://en.wikipedia.org/wiki/Tasmania) supplies about 25% of the yearly world abalone harvest.)

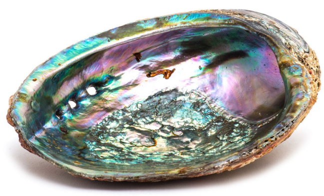


Fig 1. Inside of an abalone shell.

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

Hide

```
library(tidyverse)
library(tidymodels)
library(ggplot2)
```

Read data

Hide

Attach column with actual observe age

```
abalone<-read.csv('~/Desktop/PSTAT 131 /HW/homework-2/d
ata/abalone.csv')
dim(abalone)
```

```
## [1] 4177    9
```

Hide

```
colnames(abalone)
```

```
## [1] "type"           "longest_shell"  "diameter"
"height"
## [5] "whole_weight"   "shucked_weight" "viscera_weigh
t" "shell_weight"
## [9] "rings"
```

Question 1

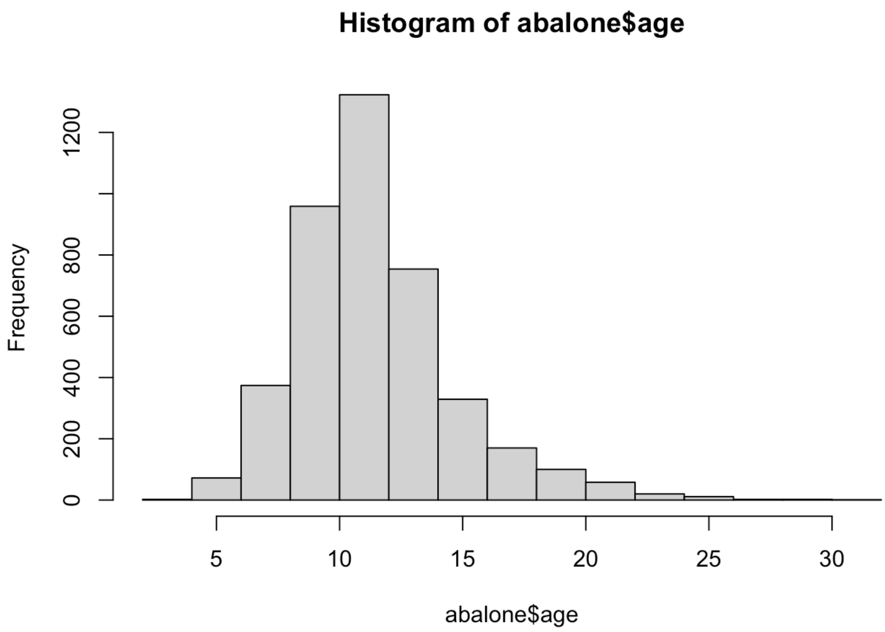
Your goal is to predict abalone age, which is calculated as the number of rings plus 1.5. Notice there currently is no `age` variable in the data set. Add `age` to the data set.

Assess and describe the distribution of `age`.

Hide

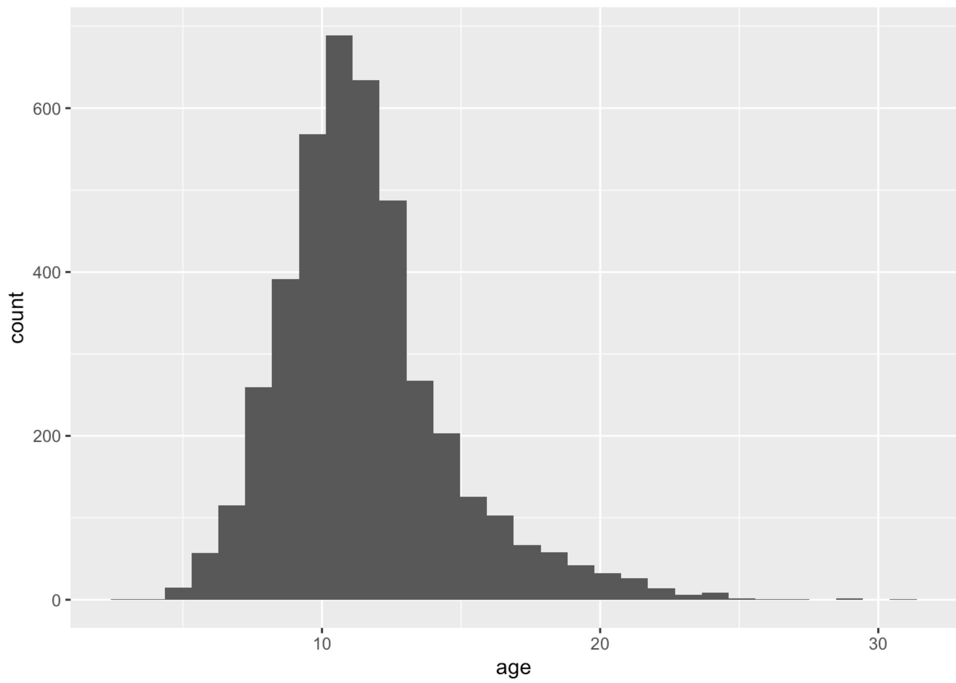
```
abalone$age<-abalone$rings +1.5

hist(abalone$age)
```



Hide

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



I think the distribution of age from the abalone dataset base on histogram is quite normal but a bit right skewed .

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.

Hide

```
# Train 80% abolone set

set.seed(9898)
abalone_split <-initial_split(abalone,prop=0.8,strata =
age)

# 80 % to train 20% to test
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` .

We should not include rings to predict age because rings + 1.5 = age .

Hide

```
#training data = abalone_split

#abalone[,-9] # remove column index for rings

simple_abalone_recipe <- recipe(age~.,data = abalone[,-
9])
```

Steps for your recipe:

1. dummy code any categorical predictors

Hide

```
abalone_recipe <- recipe(age~.,data = abalone[,-9]) %>%
step_dummy(all_nominal_predictors())
```

2. create interactions between

```
-   `type` and `shucked_weight`,
-   `longest_shell` and `diameter`,
-   `shucked_weight` and `shell_weight`
```

Hide

```
?step_interact
#-   `type` and `shucked_weight`
abalone_recipe<- abalone_recipe  %>%  step_interact(ter
ms = ~ shucked_weight:starts_with("type")) %>% step_int
eract(terms = ~ diameter:starts_with("longest_shell"))
%>% step_interact(terms = ~ shell_weight:starts_with("s
hucked_weight"))
```

3. center all predictors, and

Hide

```
#(y-y_bar)/sd
abalone_recipe <- abalone_recipe %>% step_center(.)
```

4. scale all predictors.

Hide

```
abalone_recipe <- abalone_recipe %>% step_scale()
```

You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

Question 4

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

#specify linear model

Hide

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

#Make life easier when trying out a series of models or several different reciepes

Hide

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

Fit lm to the train set

Hide

```
lm_fit <- fit(lm_wflow,abalone_train)
```

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.

Hide

```
lm_wflow <- workflow() %>%
  add_model(lm_model) %>%
  add_recipe(abalone_recipe)
#Make life easier when trying out a series of models or several different reciepes
```

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.

Hide

```
# abalone_train$

####
Prediction_1 <- predict(lm_fit,new_data=data.frame(long
est_shell=0.5,diameter = 0.10,height = 0.30, whole_weig
ht = 4,shucked_weight = 1,viscera_weight = 2,shell_weig
ht = 1, type="F"))
Prediction_1
```

```
## # A tibble: 1 × 1
##   .pred
##   <dbl>
## 1  22.9
```

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).

Hide

```
library(yardstick)

 # generates predicted values for age for each observat
ion in the training set:

abalone_train_res<-predict(lm_fit, new_data =abalone_tr
ain %>% select(-age))

abalone_train_res %>%  head()
```

```
## # A tibble: 6 × 1
##    .pred
##    <dbl>
## 1   9.45
## 2   8.09
## 3   9.31
## 4   9.73
## 5  10.3
## 6   9.97
```

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).

Attach column with actual observe age

Hide

```
abalone_train_res<-bind_cols(abalone_train_res,abalone_train %>%  select(age))

abalone_train_res %>% head()
```

```
## # A tibble: 6 × 2
##    .pred   age
##    <dbl> <dbl>
## 1   9.45   8.5
## 2   8.09   8.5
## 3   9.31   9.5
## 4   9.73   8.5
## 5  10.3    8.5
## 6   9.97   9.5
```

1. Create a metric set that includes R^2, RMSE (root mean squared error), and MAE (mean absolute error).

Hide

```
library(dplyr)
rmse(abalone_train_res, truth = age, estimate = .pred)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        2.13
```

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

Hide

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

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.13
## 2 rsq     standard       0.561
## 3 mae     standard       1.53
```