Homework 2

PSTAT 131/231

Nicole Magallanes

Contents

Linear Regression

```
library(tidymodels)
library(tidymodels)

abalone <- read.csv("/Users/nicolemagallanes/Desktop/hw2-nicolemagallanes/abalone.csv")
#view(abalone)
```

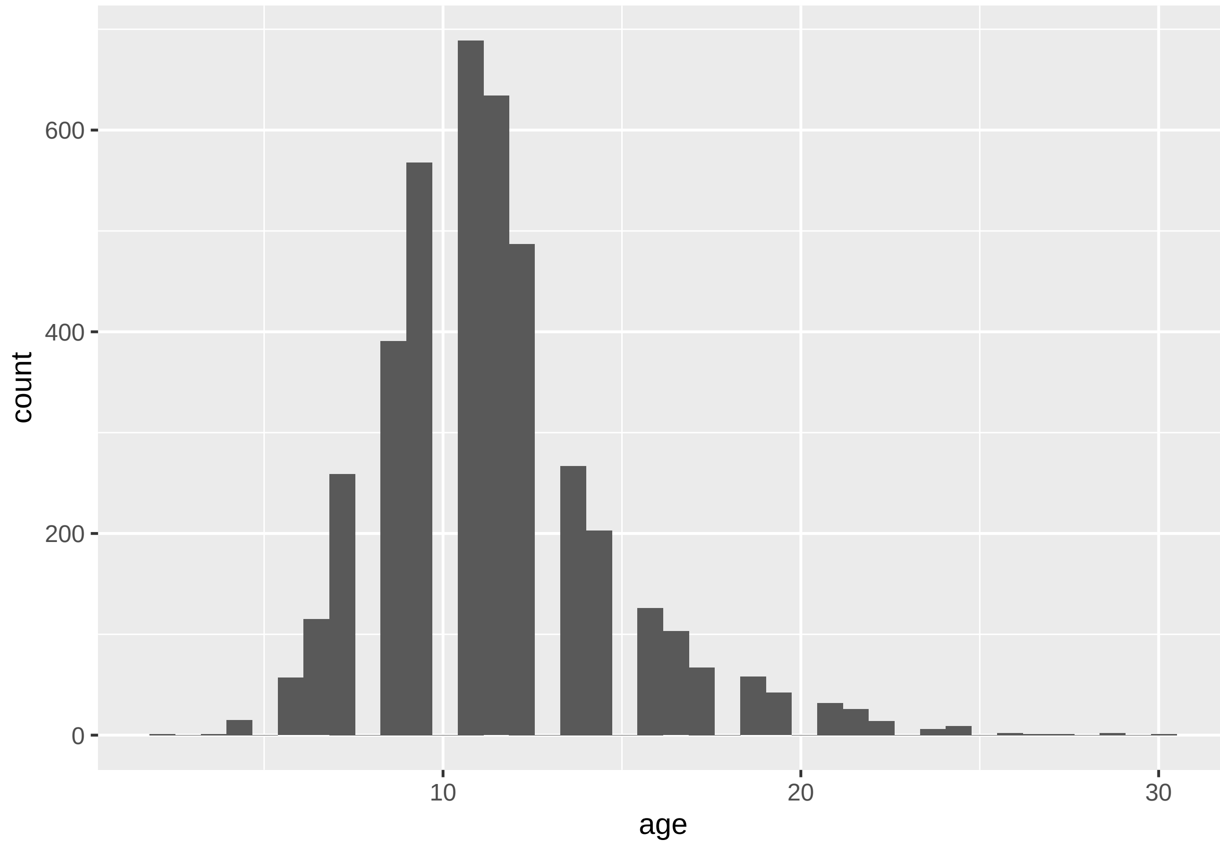
Question 1

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`.

```
age <- abalone$rings + 1.5
abalone$age <- age
# view(abalone)

abalone %>%
  ggplot(aes(x=age)) +
  geom_histogram(bins=40)
```



The distribution of Age seems to be somewhat positively skewed.We can see the data peaks at around age of 11-12 and as the distribution is right skewed we can see that there aren't a lot of abalone that are past the age of 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.

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

```
set.seed(4160)

abalone_split <- initial_split(abalone, prop = 0.80,
                               strata = age)
abalone_train <- training(abalone_split)
abalone_test <- testing(abalone_split)

#abalone_train
#abalone_test
```

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_recipe <- recipe(age~., data= abalone_train) %>%
  step_rm(rings) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~ starts_with("type"):shucked_weight +
                  longest_shell:diameter + shucked_weight:shell_weight) %>%
 step_scale(all_numeric_predictors()) %>%
  step_center(all_numeric_predictors())
abalone_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          9
##
## Operations:
##
## Variables removed rings
## Dummy variables from all_nominal_predictors()
## Interactions with starts_with("type"):shucked_weight + longest_shell...
## Scaling for all_numeric_predictors()
## Centering for all_numeric_predictors()
```

We do not include rings because rings to predict age because rings is already use to see the age, it is a variable that is included withing age.

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)
```

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.

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

newd <- data.frame(type = "F", longest_shell = 0.50, diameter = 0.10, height = 0.30, whole_weight = 4, s

#view(newd)

predict(lm_fit, new_data = newd)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  24.1
```

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)

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

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  9.52   8.5
## 2  8.09   8.5
## 3  9.29   9.5
## 4  9.72   8.5
## 5 10.5    8.5
## 6 10.1    9.5
```

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

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        2.15
```

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

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.15
## 2 rsq     standard       0.557
## 3 mae     standard       1.54
```

Our R-squared value tells us that 55.73% of our variance in the dependent variable can be explained by the independent variables. The higher the R-square, usually means the better our model fits the data. In this case our R-square value is really low, meaning our model only explains 55.73% of our fitted data.