Homework 2

PSTAT 131/231

Contents

Linear Regression

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`.

```
abalone <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data", hea
colnames(abalone) = c("Sex", "Length", "Diameter", "Height", "Whole", "Shucked","Viscera", "Shell", "Ri
head(abalone)
```

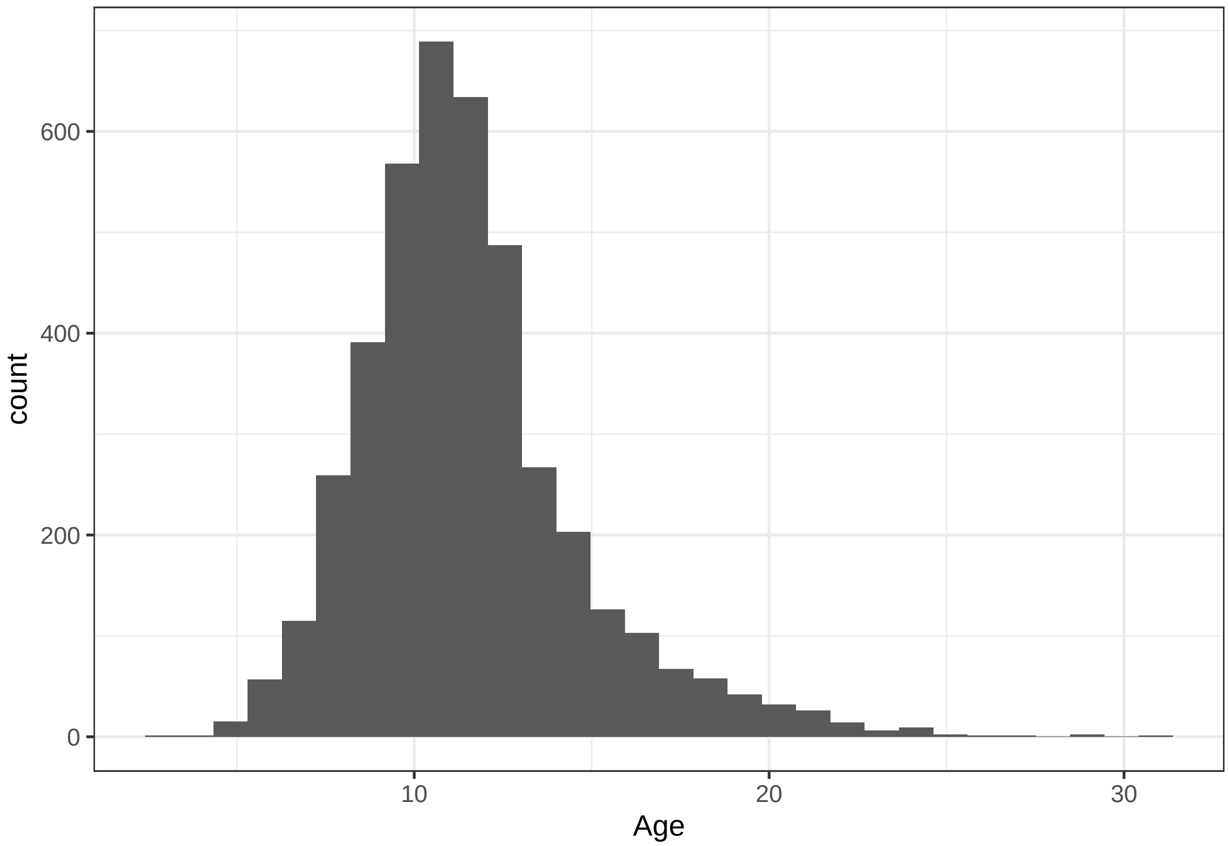
```
##   Sex Length Diameter Height  Whole Shucked Viscera Shell Rings
## 1   M  0.455    0.365  0.095 0.5140  0.2245  0.1010 0.150    15
## 2   M  0.350    0.265  0.090 0.2255  0.0995  0.0485 0.070     7
## 3   F  0.530    0.420  0.135 0.6770  0.2565  0.1415 0.210     9
## 4   M  0.440    0.365  0.125 0.5160  0.2155  0.1140 0.155    10
## 5   I  0.330    0.255  0.080 0.2050  0.0895  0.0395 0.055     7
## 6   I  0.425    0.300  0.095 0.3515  0.1410  0.0775 0.120     8
```

```
library(dplyr)
library(rlang)
library(ggplot2)
library(tidyverse)
library(tidymodels)
library(corrplot)
tidymodels_prefer()

data<- abalone %>%
  mutate(Age = Rings+1.5)

head(data)
```

```
##   Sex Length Diameter Height  Whole Shucked Viscera Shell Rings  Age
## 1   M  0.455    0.365  0.095 0.5140  0.2245  0.1010 0.150    15 16.5
## 2   M  0.350    0.265  0.090 0.2255  0.0995  0.0485 0.070     7  8.5
## 3   F  0.530    0.420  0.135 0.6770  0.2565  0.1415 0.210     9 10.5
```

```
## 4   M  0.440    0.365  0.125 0.5160  0.2155  0.1140 0.155    10 11.5
## 5   I  0.330    0.255  0.080 0.2050  0.0895  0.0395 0.055     7  8.5
## 6   I  0.425    0.300  0.095 0.3515  0.1410  0.0775 0.120     8  9.5
```

```
data %>%
  ggplot(aes(x = Age)) +
  geom_histogram(bins = 30) +
  theme_bw()
```



Age is normally distributed with slight skewed to the lower end, with a long tail to the right.Most of the abalone is less than 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(3435)
data_split <- initial_split(data, prop = 0.80,
                                strata = Age)
data_train <- training(data_split)
data_test <- testing(data_split)

head(data_train)
```

```
##    Sex Length Diameter Height  Whole Shucked Viscera Shell Rings Age
```

```
## 5    I  0.330    0.255  0.080 0.2050  0.0895  0.0395 0.055     7 8.5
## 17   I  0.355    0.280  0.085 0.2905  0.0950  0.0395 0.115     7 8.5
## 19   M  0.365    0.295  0.080 0.2555  0.0970  0.0430 0.100     7 8.5
## 36   M  0.465    0.355  0.105 0.4795  0.2270  0.1240 0.125     8 9.5
## 38   F  0.450    0.355  0.105 0.5225  0.2370  0.1165 0.145     8 9.5
## 43   I  0.240    0.175  0.045 0.0700  0.0315  0.0235 0.020     5 6.5
```

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

```
data_recipe <-  recipe(Age ~ Sex+Length+Diameter+Height+Whole+Shucked+Viscera+Shell, data = data_train)
  step_dummy(all_nominal_predictors())
data_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from all_nominal_predictors()
```

```
summary(data_recipe)
```

```
## # A tibble: 9 x 4
##   variable type    role      source
##   <chr>    <chr>   <chr>     <chr>
## 1 Sex      nominal predictor original
## 2 Length   numeric predictor original
## 3 Diameter numeric predictor original
## 4 Height   numeric predictor original
## 5 Whole    numeric predictor original
## 6 Shucked  numeric predictor original
## 7 Viscera  numeric predictor original
## 8 Shell    numeric predictor original
## 9 Age      numeric outcome   original
```

2. create interactions between

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

```
data_recipe <-  recipe(Age ~ Sex+Length+Diameter+Height+Whole+Shucked+Viscera+Shell, data = data_tr
  step_dummy(all_nominal_predictors())%>%
  step_interact(terms = ~ Sex:Shucked +Length:Diameter+Shucked:Shell)
data_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from all_nominal_predictors()
## Interactions with Sex:Shucked + Length:Diameter + Shucked:Shell
```

```
summary(data_recipe)
```

```
## # A tibble: 9 x 4
##   variable type    role      source
##   <chr>    <chr>   <chr>     <chr>
## 1 Sex      nominal predictor original
## 2 Length   numeric predictor original
## 3 Diameter numeric predictor original
## 4 Height   numeric predictor original
## 5 Whole    numeric predictor original
## 6 Shucked  numeric predictor original
## 7 Viscera  numeric predictor original
## 8 Shell    numeric predictor original
## 9 Age      numeric outcome   original
```

3. center all predictors, and

```
data_recipe <-  recipe(Age ~ Sex+Length+Diameter+Height+Whole+Shucked+Viscera+Shell, data = data_train)
  step_dummy(all_nominal_predictors())%>%
  step_interact(terms = ~ Sex:Shucked +Length:Diameter+Shucked:Shell)%>%
  step_center(all_predictors())
data_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
```

```
##
## Operations:
##
## Dummy variables from all_nominal_predictors()
## Interactions with Sex:Shucked + Length:Diameter + Shucked:Shell
## Centering for all_predictors()
```

```
summary(data_recipe)
```

```
## # A tibble: 9 x 4
##   variable type    role      source
##   <chr>    <chr>   <chr>     <chr>
## 1 Sex      nominal predictor original
## 2 Length   numeric predictor original
## 3 Diameter numeric predictor original
## 4 Height   numeric predictor original
## 5 Whole    numeric predictor original
## 6 Shucked  numeric predictor original
## 7 Viscera  numeric predictor original
## 8 Shell    numeric predictor original
## 9 Age      numeric outcome   original
```

4. scale all predictors.

```
data_recipe <-  recipe(Age ~ Sex+Length+Diameter+Height+Whole+Shucked+Viscera+Shell, data = data_train)
  step_dummy(all_nominal_predictors())%>%
  step_interact(terms = ~ Sex:Shucked +Length:Diameter+Shucked:Shell)%>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors())
data_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from all_nominal_predictors()
## Interactions with Sex:Shucked + Length:Diameter + Shucked:Shell
## Centering for all_predictors()
## Scaling for all_predictors()
```

```
summary(data_recipe)
```

```
## # A tibble: 9 x 4
##   variable type    role      source
##   <chr>    <chr>   <chr>     <chr>
```

```
## 1 Sex      nominal predictor original
## 2 Length   numeric predictor original
## 3 Diameter numeric predictor original
## 4 Height   numeric predictor original
## 5 Whole    numeric predictor original
## 6 Shucked  numeric predictor original
## 7 Viscera  numeric predictor original
## 8 Shell    numeric predictor original
## 9 Age      numeric outcome   original
```

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.

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

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(data_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, data_train)


data_predict = data.frame("F", 0.50, 0.10, 0.30, 4, 1,2, 1)
colnames(data_predict) = c("Sex", "Length", "Diameter", "Height", "Whole", "Shucked","Viscera", "Shell")
data_train_res1 <- predict(lm_fit, new_data = data_predict )
data_train_res1
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  13.4
```

Question 7

Now you want to assess your model's performance. To do this, use the `yardstick` package:

1. Create a metric set includes r square, 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 square value.

```
data_train_res <- predict(lm_fit, new_data = data_train %>% select(-Age))
data_train_res <- bind_cols(data_train_res, data_train%>% select(Age))
data_train_res %>%
  head()
```

```
## # A tibble: 6 x 2
##    .pred   Age
##    <dbl> <dbl>
## 1  8.19   8.5
## 2  9.70   8.5
## 3 10.2    8.5
## 4  9.91   9.5
## 5 10.3    9.5
## 6  6.81   6.5
```

```
rmse(data_train_res, truth = Age, estimate = .pred)
```

```
## # A tibble: 1 x 3
##    .metric .estimator .estimate
##    <chr>   <chr>          <dbl>
## 1 rmse    standard        2.20
```

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


data_metrics(data_train_res, truth = Age,
                estimate = .pred)
```

```
## # A tibble: 3 x 3
##    .metric .estimator .estimate
##    <chr>   <chr>          <dbl>
## 1 rmse    standard       2.20
## 2 rsq     standard       0.535
## 3 mae     standard       1.59
```