

HW2 PSTAT131

Xilong Li

2022-04-10

```
library(tidymodels)
```

```
## -- Attaching packages ----- tidymodels 0.2.0 --
```

```
## v broom      0.7.12    v recipes      0.2.0
## v dials      0.1.1     v rsample      0.1.1
## v dplyr      1.0.8     v tibble      3.1.6
## v ggplot2    3.3.5     v tidyr       1.2.0
## v infer      1.0.0     v tune        0.2.0
## v modeldata  0.1.1     v workflows   0.2.6
## v parsnip    0.2.1     v workflowsets 0.2.1
## v purrr      0.3.4     v yardstick   0.0.9
```

```
## -- Conflicts ----- tidymodels_conflicts() --
```

```
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v readr      2.1.2    v forcats 0.5.1
## v stringr    1.4.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x stringr::fixed()    masks recipes::fixed()
## x dplyr::lag()        masks stats::lag()
## x readr::spec()       masks yardstick::spec()
```

```
library(ggplot2)
```

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

```
##   type longest_shell diameter height whole_weight shucked_weight viscera_weight
## 1    M          0.455   0.365  0.095     0.5140         0.2245         0.1010
## 2    M          0.350   0.265  0.090     0.2255         0.0995         0.0485
## 3    F          0.530   0.420  0.135     0.6770         0.2565         0.1415
## 4    M          0.440   0.365  0.125     0.5160         0.2155         0.1140
## 5    I          0.330   0.255  0.080     0.2050         0.0895         0.0395
## 6    I          0.425   0.300  0.095     0.3515         0.1410         0.0775
##   shell_weight rings
## 1         0.150   15
## 2         0.070    7
## 3         0.210    9
## 4         0.155   10
## 5         0.055    7
## 6         0.120    8
```

```
dim(abalone)
```

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

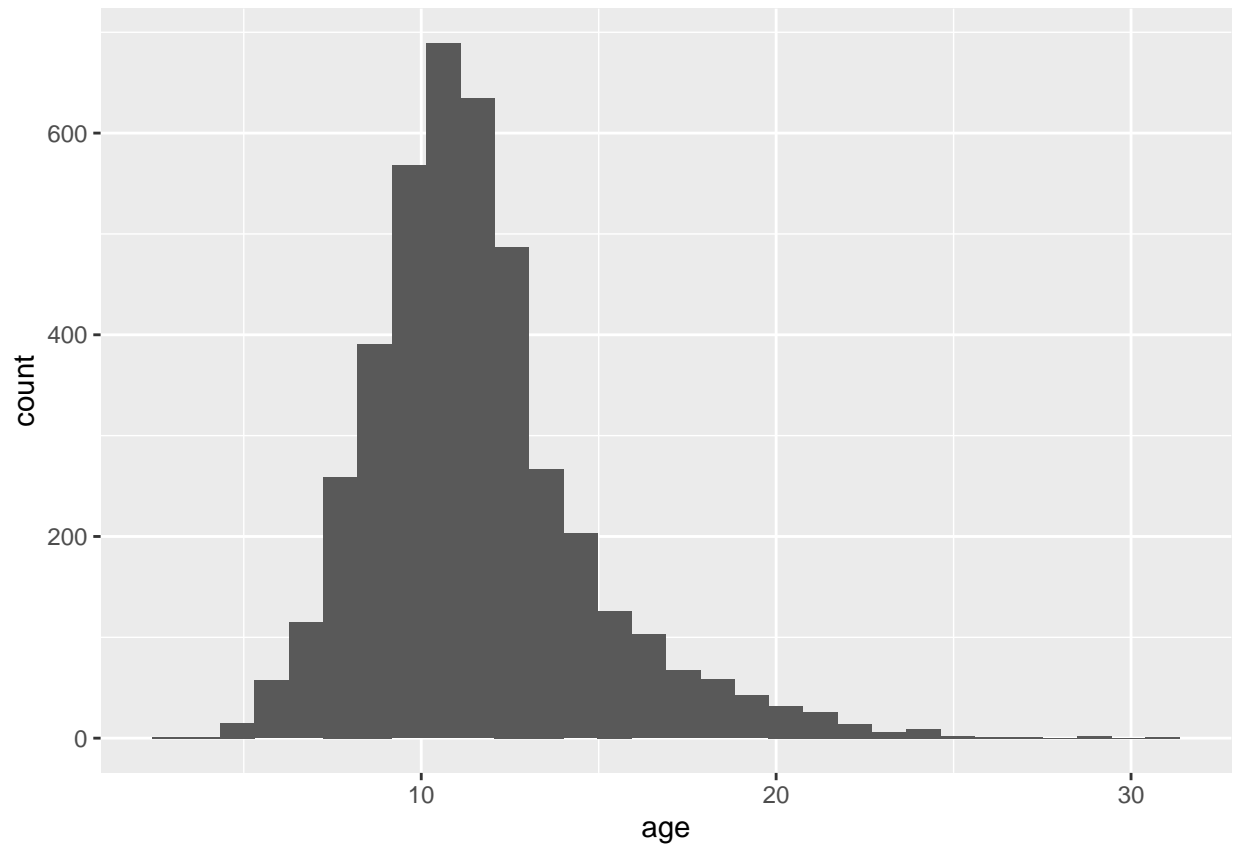
Question 1:

```
abalone<-
  abalone %>%
    mutate(age = rings + 1.5) %>%
    select(age, everything())
head(abalone)
```

```
##   age type longest_shell diameter height whole_weight shucked_weight
## 1 16.5    M          0.455   0.365  0.095     0.5140         0.2245
## 2  8.5    M          0.350   0.265  0.090     0.2255         0.0995
## 3 10.5    F          0.530   0.420  0.135     0.6770         0.2565
## 4 11.5    M          0.440   0.365  0.125     0.5160         0.2155
## 5  8.5    I          0.330   0.255  0.080     0.2050         0.0895
## 6  9.5    I          0.425   0.300  0.095     0.3515         0.1410
##   viscera_weight shell_weight rings
## 1         0.1010         0.150   15
## 2         0.0485         0.070    7
## 3         0.1415         0.210    9
## 4         0.1140         0.155   10
## 5         0.0395         0.055    7
## 6         0.0775         0.120    8
```

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

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



As the histogram shown above, the distribution of age is approximately normal, with a tail on the right. And most of the abalone has the age around 10 in this data set.

Question 2:

```
set.seed(2216)

abalone_split <- initial_split(abalone, prop = 0.80)
abalone_train <- training(abalone_split)
abalone_test  <- testing(abalone_split)
dim(abalone_train)
```

```
## [1] 3341  10
```

```
dim(abalone_test)
```

```
## [1] 836  10
```

Question 3:

```
head(abalone)
```

```
##   age type longest_shell diameter height whole_weight shucked_weight
## 1 16.5   M         0.455    0.365  0.095      0.5140      0.2245
## 2  8.5   M         0.350    0.265  0.090      0.2255      0.0995
## 3 10.5   F         0.530    0.420  0.135      0.6770      0.2565
## 4 11.5   M         0.440    0.365  0.125      0.5160      0.2155
## 5  8.5   I         0.330    0.255  0.080      0.2050      0.0895
## 6  9.5   I         0.425    0.300  0.095      0.3515      0.1410
##   viscera_weight shell_weight rings
## 1          0.1010         0.150    15
## 2          0.0485         0.070     7
## 3          0.1415         0.210     9
## 4          0.1140         0.155    10
## 5          0.0395         0.055     7
## 6          0.0775         0.120     8
```

```
abalone_recipe <-
  recipe(age ~
    type +
    longest_shell +
    diameter + height +
    whole_weight +
    shucked_weight +
    viscera_weight +
    shell_weight,
    data = abalone_train) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(~ starts_with("type"):shucked_weight) %>%
  step_interact(~ longest_shell:diameter) %>%
  step_interact(~ shucked_weight:shell_weight) %>%
  step_normalize()

abalone_recipe
```

```
## Recipe
##
## Inputs:
##
##   role #variables
##   outcome      1
##   predictor      8
##
## Operations:
##
## Dummy variables from all_nominal_predictors()
## Interactions with starts_with("type"):shucked_weight
## Interactions with longest_shell:diameter
## Interactions with shucked_weight:shell_weight
## Centering and scaling for <none>
```

```
# We should not include the predictor "rings",
# because the data of age is directly derived from rings;
# Thus, the training data would be 100% compatible with the testing data on the predictor of "rings"
```

Question 4:

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

#code is cited from lab2

Question 5:

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

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

```
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)                      4.45      0.673      6.61 4.47e-11  
## 2 longest_shell                    5.53      2.35      2.35 1.86e- 2  
## 3 diameter                        19.0      3.11      6.10 1.22e- 9  
## 4 height                          4.68      1.61      2.90 3.72e- 3  
## 5 whole_weight                     9.99      0.799     12.5 4.76e-35  
## 6 shucked_weight                  -18.9      1.12     -16.9 7.30e-62  
## 7 viscera_weight                  -8.99      1.42      -6.34 2.56e-10  
## 8 shell_weight                    12.6      1.54      8.17 4.22e-16  
## 9 type_I                          -2.04      0.238     -8.57 1.60e-17  
## 10 type_M                         -0.440     0.209     -2.10 3.55e- 2  
## 11 type_I_x_shucked_weight         4.46      0.725      6.15 8.84e-10  
## 12 type_M_x_shucked_weight         1.10      0.428      2.56 1.05e- 2  
## 13 longest_shell_x_diameter       -29.4      4.15      -7.10 1.48e-12  
## 14 shucked_weight_x_shell_weight  -1.34      1.66      -0.803 4.22e- 1
```

#code is cited from lab2

Question 6:

not yet finished

QuesTion 7:

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

```
## # A tibble: 6 x 1
##   .pred
##   <dbl>
## 1 12.6
## 2  9.27
## 3 13.2
## 4 12.6
## 5 13.0
## 6 11.6
```

```
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 12.6  11.5
## 2  9.27  9.5
## 3 13.2  12.5
## 4 12.6  12.5
## 5 13.0  13.5
## 6 11.6   9.5
```

```
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      2.13
## 2 rsq    standard      0.566
## 3 mae    standard      1.53
```

R-Squared means the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.(cited from <https://www.investopedia.com/terms/r/r-squared.asp>)

Therefore, since the value of R-squared is 0.5639424, this means that 56.39424% of the response variable can be explained by predictor variables.