PSATA131-HW2

Yifan Xu

2022-10-17

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.0.0 --
```

```
## v broom        1.0.1     v rsample      1.1.0
## v dials        1.0.0     v tune         1.0.0
## v infer        1.0.3     v workflows    1.1.0
## v modeldata    1.0.1     v workflowsets 1.0.0
## v parsnip      1.0.2     v yardstick    1.1.0
## v recipes      1.0.1
```

```
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(ISLR)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

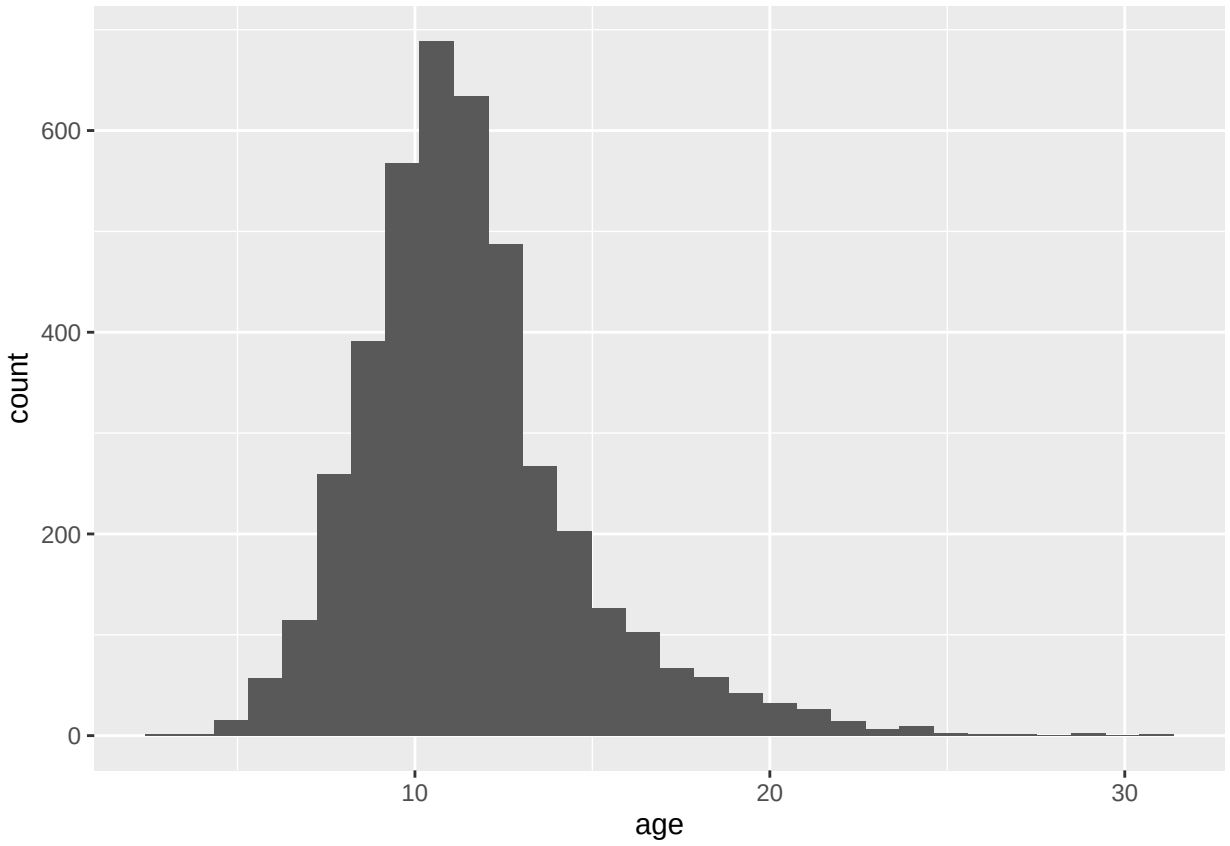
```
library(ggthemes)
library(yardstick)
tidymodels_prefer()
set.seed(100)
```

```
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
```

Q1

```
# Add age column to the abalone with "rings" + 1.5
abalone["age"] <- abalone["rings"]+1.5
# To assess the distribution of age, we can use histogram to check
abalone %>% ggplot(aes(age))+geom_histogram(bins=30)
```



According to the plot, we can conclude that the distribution of age relatively follows the normal distribution with mean at about 10-12, but it is slightly skewed to the right. The majority of data locates between 4 and 17, however, there exist some extreme outliers around 25 to 32.

Q2

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

Q3

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

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from all_nominal_predictors()
## Interactions with starts_with("type"):shucked_weight + longest_shell...
## Centering for all_predictors()
## Scaling for all_predictors()
```

We can't use rings to predict age, because the age column is just the linear transformation of the rings column, they have exactly the same trend and distribution with shift. Thherefore, rings cannot be used to predict age.

Q4

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

Q5

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

Q6

```
lm_fit <- fit(lm_wflow,abalone_train %>% select(-rings))
female_pred <- data.frame(type = "F", longest_shell = 0.50,
                          diameter = 0.10, height = 0.30,
                          whole_weight = 4, shucked_weight = 1,
                          viscera_weight = 2, shell_weight = 1)
predict(lm_fit, new_data = female_pred)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  20.5
```

```
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       0.0374   306.    0
##  2 longest_shell                    0.0318    0.289      0.110 9.12e- 1
##  3 diameter                         2.15      0.317      6.78  1.46e-11
##  4 height                           0.464     0.0984     4.72  2.49e- 6
##  5 whole_weight                     4.84      0.397     12.2   1.62e-33
##  6 shucked_weight                  -4.03      0.255    -15.8   2.35e-54
##  7 viscera_weight                  -1.06      0.158     -6.70  2.40e-11
##  8 shell_weight                     1.57      0.222      7.06  1.99e-12
##  9 type_I                          -0.915     0.114     -8.01  1.61e-15
## 10 type_M                          -0.171     0.104     -1.64  1.02e- 1
## 11 type_I_x_shucked_weight          0.499     0.0862     5.79  7.70e- 9
## 12 type_M_x_shucked_weight          0.202     0.110      1.84  6.53e- 2
## 13 longest_shell_x_diameter        -2.43      0.409     -5.94  3.08e- 9
## 14 shucked_weight_x_shell_weight   -0.232     0.207     -1.12  2.62e- 1
```

Q7

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

```
## # A tibble: 6 x 2
##   .pred   age
##   <dbl> <dbl>
## 1  9.58   8.5
## 2  8.06   8.5
## 3  9.19   9.5
## 4  9.71   8.5
## 5 10.0    9.5
## 6  5.96   5.5
```

```
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       2.16
## 2 rsq     standard       0.554
## 3 mae     standard       1.55
```

We get approximate 0.5543753 for R squared value which indicates that 55.437525% of the data fit the regression model.