PSTAT131/HW2

Cynthia Cao

2022-10-08

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- 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()
## * Learn how to get started at https://www.tidymodels.org/start/
```

```
abalone <- read_csv("/Users/cynnnthiaaa/Desktop/abalone.csv")
```

```
## Rows: 4177 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): type
## dbl (8): longest_shell, diameter, height, whole_weight, shucked_weight, visc...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

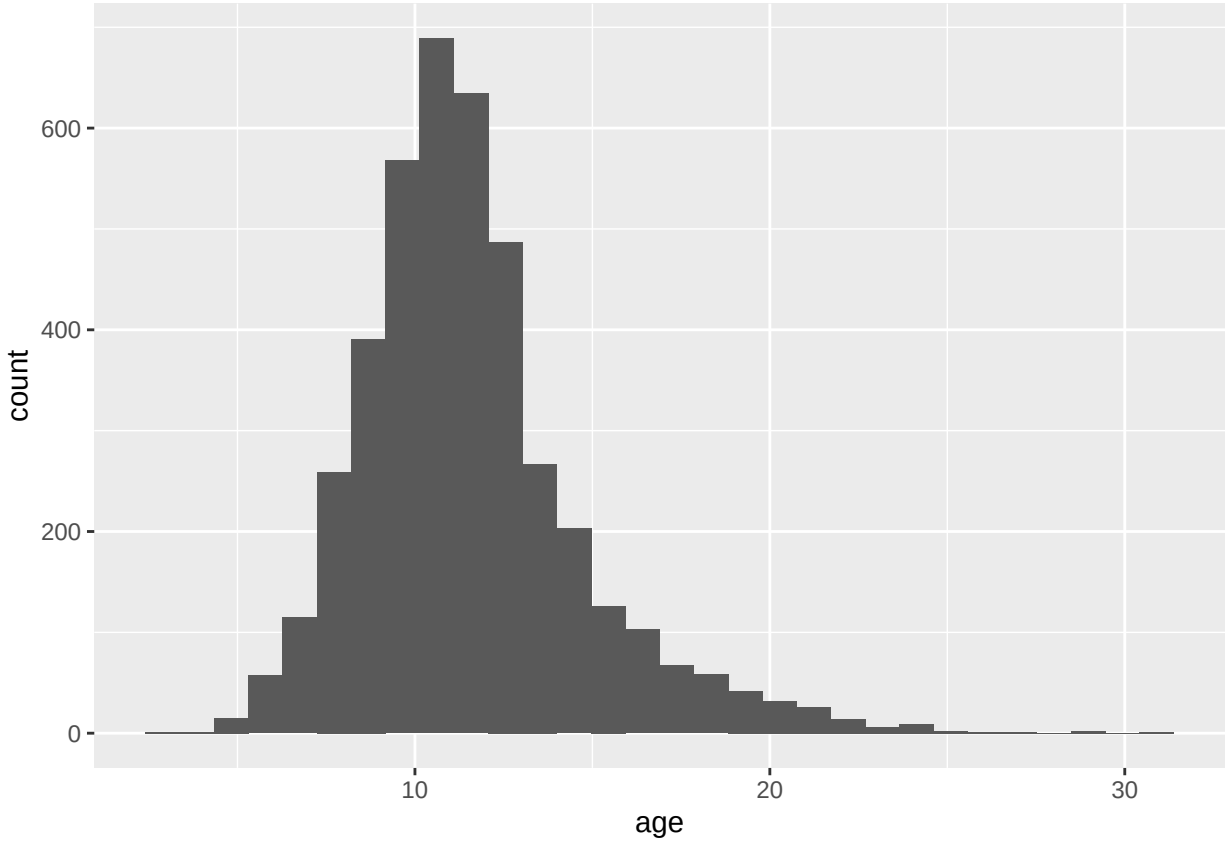
Q1:

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

```
## # A tibble: 4,177 x 10
##    type  longest_sh~1 diame~2 height whole~3 shuck~4 visce~5 shell~6 rings   age
##    <chr>        <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl>
##  1 M            0.455   0.365  0.095   0.514  0.224   0.101    0.15     15  16.5
##  2 M            0.35    0.265  0.09    0.226  0.0995  0.0485   0.07      7   8.5
##  3 F            0.53    0.42   0.135   0.677  0.256   0.142    0.21      9  10.5
##  4 M            0.44    0.365  0.125   0.516  0.216   0.114    0.155    10  11.5
##  5 I            0.33    0.255  0.08    0.205  0.0895  0.0395   0.055     7   8.5
##  6 I            0.425   0.3    0.095   0.352  0.141   0.0775   0.12      8   9.5
##  7 F            0.53    0.415  0.15    0.778  0.237   0.142    0.33     20  21.5
##  8 F            0.545   0.425  0.125   0.768  0.294   0.150    0.26     16  17.5
##  9 M            0.475   0.37   0.125   0.509  0.216   0.112    0.165     9  10.5
## 10 F            0.55    0.44   0.15    0.894  0.314   0.151    0.32     19  20.5
## # ... with 4,167 more rows, and abbreviated variable names 1: longest_shell,
## #   2: diameter, 3: whole_weight, 4: shucked_weight, 5: viscera_weight,
## #   6: shell_weight
```

```
ggplot(abalone, aes(x=age)) + geom_histogram()
```

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



From the graph, we can see that the distribution of age is roughly normal, most observations are between age of 7- 15.

Q2:

```
set.seed(3456)
abalone_split <- initial_split(abalone, prop = 0.75,strata = age)
train  <- training(abalone_split)
```

```
test   <- testing(abalone_split)
train
```

```
## # A tibble: 3,131 x 10
##    type  longest_sh~1 diame~2 height whole~3 shuck~4 visce~5 shell~6 rings   age
##    <chr>        <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl>
##  1 M            0.35    0.265  0.09    0.226  0.0995  0.0485   0.07      7   8.5
##  2 I            0.33    0.255  0.08    0.205  0.0895  0.0395   0.055     7   8.5
##  3 I            0.425   0.3    0.095   0.352  0.141   0.0775   0.12      8   9.5
##  4 I            0.355   0.28   0.085   0.290  0.095   0.0395   0.115     7   8.5
##  5 M            0.365   0.295  0.08    0.256  0.097   0.043    0.1       7   8.5
##  6 M            0.465   0.355  0.105   0.480  0.227   0.124    0.125     8   9.5
##  7 F            0.45    0.355  0.105   0.522  0.237   0.116    0.145     8   9.5
##  8 I            0.24    0.175  0.045   0.07   0.0315  0.0235   0.02      5   6.5
##  9 I            0.21    0.15   0.05    0.042  0.0175  0.0125   0.015     4   5.5
## 10 I            0.39    0.295  0.095   0.203  0.0875  0.045    0.075     7   8.5
## # ... with 3,121 more rows, and abbreviated variable names 1: longest_shell,
## #   2: diameter, 3: whole_weight, 4: shucked_weight, 5: viscera_weight,
## #   6: shell_weight
```

```
test
```

```
## # A tibble: 1,046 x 10
##    type  longest_sh~1 diame~2 height whole~3 shuck~4 visce~5 shell~6 rings   age
##    <chr>        <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl>
##  1 M            0.44    0.365  0.125   0.516   0.216  0.114    0.155    10  11.5
##  2 F            0.55    0.44   0.15    0.894   0.314  0.151    0.32     19  20.5
##  3 F            0.525   0.38   0.14    0.606   0.194  0.148    0.21     14  15.5
##  4 F            0.47    0.355  0.1     0.476   0.168  0.0805   0.185    10  11.5
##  5 M            0.5     0.4    0.13    0.664   0.258  0.133    0.24     12  13.5
##  6 M            0.45    0.32   0.1     0.381   0.170  0.075    0.115     9  10.5
##  7 F            0.615   0.48   0.165   1.16    0.513  0.301    0.305    10  11.5
##  8 F            0.56    0.44   0.14    0.928   0.382  0.188    0.3      11  12.5
##  9 M            0.59    0.445  0.14    0.931   0.356  0.234    0.28     12  13.5
## 10 M            0.605   0.475  0.18    0.936   0.394  0.219    0.295    15  16.5
## # ... with 1,036 more rows, and abbreviated variable names 1: longest_shell,
## #   2: diameter, 3: whole_weight, 4: shucked_weight, 5: viscera_weight,
## #   6: shell_weight
```

Q3:

```
new_abalone <- select(train,-rings)
abalone_recipe <- recipe(age~.,data = new_abalone) %>%
  step_dummy(type)

abalone_recipe <- step_interact(abalone_recipe, terms = ~ shucked_weight : starts_with('type'))
abalone_recipe <- step_interact(abalone_recipe, terms = ~ longest_shell : diameter)
abalone_recipe <- step_interact(abalone_recipe, terms = ~ shucked_weight : shell_weight)

abalone_recipe <- step_center(abalone_recipe, longest_shell, diameter, height, whole_weight, shucked_wei
abalone_recipe <- step_scale(abalone_recipe, longest_shell, diameter, height, whole_weight, shucked_weig

abalone_recipe
```

```
## Recipe
##
```

```
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from type
## Interactions with shucked_weight:starts_with("type")
## Interactions with longest_shell:diameter
## Interactions with shucked_weight:shell_weight
## Centering for longest_shell, diameter, height, whole_weight, ...
## Scaling for longest_shell, diameter, height, whole_weight, ...
```

Since we already know the condition that age=rings+1.5, the relationship between two variables is fixed, then its unnecessary to include it in the prediction.

Q4:

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

Q5:

```
workflow1 <- workflow() %>%
  add_model(lm_model) %>%
  add_recipe(abalone_recipe)
```

Q6:

```
fit1 <- fit(workflow1, new_abalone)
type <- c('F')
longest_shell <- c(0.50)
diameter <- c(0.10)
height <- c(0.30)
whole_weight <- c(4)
shucked_weight <- c(1)
viscera_weight <- c(2)
shell_weight <- c(1)
data1 <- data.frame(type, longest_shell,diameter,height,whole_weight,
                          shucked_weight, viscera_weight, shell_weight)
predict(fit1, new_data = data1)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  22.4
```

Q7:

```
library(yardstick)
metric <- metric_set(rsq, rmse, mae)
result1 <- predict(fit1, new_data=new_abalone %>% select(-age))
result2 <- bind_cols(result1, new_abalone %>% select(age))
result2
```

```
## # A tibble: 3,131 x 2
##    .pred   age
##    <dbl> <dbl>
##  1  9.35   8.5
##  2  8.09   8.5
##  3  9.37   9.5
##  4  9.78   8.5
##  5 10.3    8.5
##  6  9.98   9.5
##  7 10.9    9.5
##  8  6.31   6.5
##  9  5.97   5.5
## 10  8.51   8.5
## # ... with 3,121 more rows
```

```
metric(result2, truth = age, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.558
## 2 rmse    standard       2.15
## 3 mae     standard       1.54
```

R Square: 0.5580969 (about 55.81% of variability of Y can be explained by X) Root Mean Square Error: 2.1463035
Mean Absolute Error: 1.5375481