pstat231-hw2

2022-10-13

Q1:

```
library(tidymodels)
```

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

```
## v broom        1.0.1     v recipes      1.0.1
## v dials        1.0.0     v rsample      1.1.0
## v dplyr        1.0.9     v tibble       3.1.7
## v ggplot2      3.3.6     v tidyr        1.2.0
## v infer        1.0.3     v tune         1.0.0
## v modeldata    1.0.1     v workflows    1.1.0
## v parsnip      1.0.2     v workflowsets 1.0.0
## v purrr        0.3.4     v yardstick    1.1.0
```

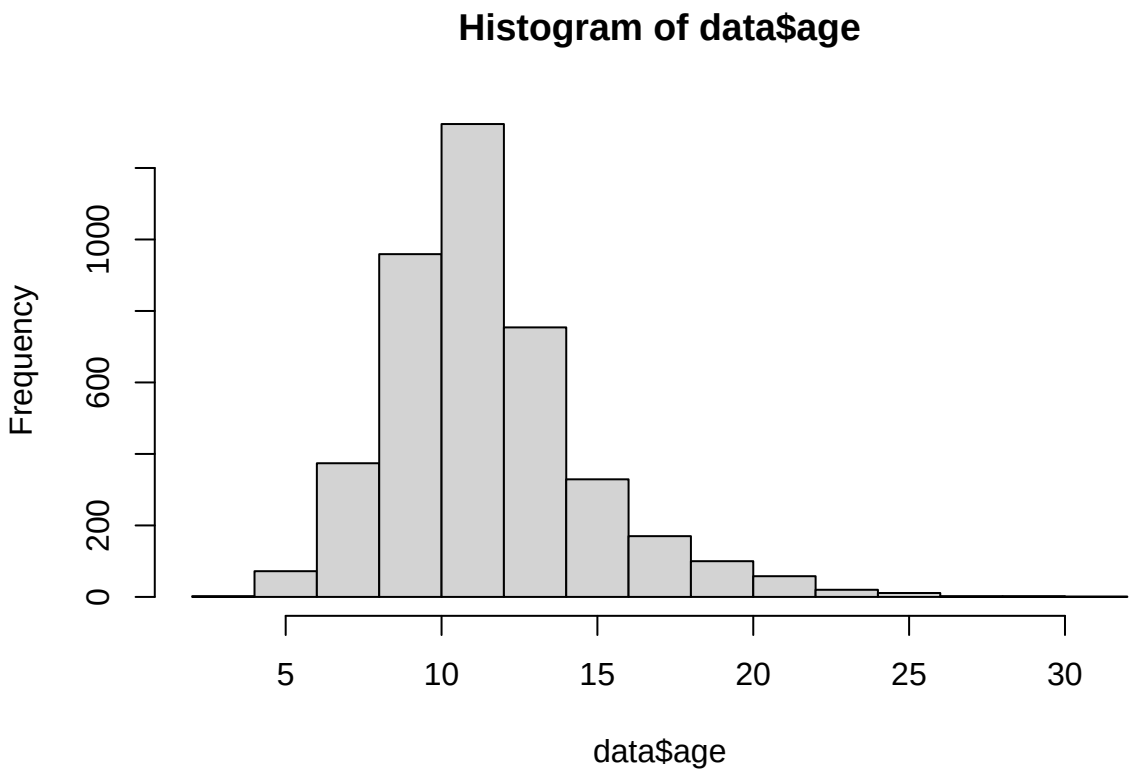
```
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Search for functions across packages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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## 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()
```

```
data <- read.csv("data/abalone.csv")
data['age'] = data['rings'] + 1.5
hist(data$age)
```



The distribution is right skewed with most samples having ages between 7.5 and 15.

Q2:

```
set.seed(42)

data_split <- initial_split(data, prop = 0.80,
                                  strata = age)
data_train <- training(data_split) %>% select(-rings)
data_test <- testing(data_split) %>% select(-rings)
```

Q3:

```
recipe <-
  recipe(age ~ ., data = data_train) %>%
    step_dummy(all_nominal_predictors()) %>%
    step_interact(terms = ~ starts_with(('type_')) + shucked_weight) %>%
    step_interact(terms = ~ longest_shell + diameter) %>%
    step_interact(terms = ~ shucked_weight + shell_weight) %>%
    step_center(all_numeric_predictors()) %>%
    step_scale(all_numeric_predictors())
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 for all_numeric_predictors()
## Scaling for all_numeric_predictors()
```

The whole purpose of this model is to calculate abalones' age without having to count rings. Having the ring count gives the age, so the model would be pointless if we still needed this value for predicting age.

Q4:

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

Q5:

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

Q6:

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

sample <- data.frame('F', .50, .10, .30, 4, 1, 2, 1)
names(sample) <- names(data_train %>% select(-age))

predict(lm_fit, new_data = sample)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  15.5
```

Q7:

```
library(yardstick)

lm_metrics <- metric_set(rsq, rmse, mae)
pred <- predict(lm_fit, data_train)

results <- bind_cols(pred, data_train %>% select(age))

lm_metrics(results, truth = age, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.534
## 2 rmse    standard       2.19
## 3 mae     standard       1.58
```

The model observes 53% of the variability in the target variable.

Q8: Bias and Variance represent reproducible errors. The zero-mean random noise represents irreducible error.

Q9: Given a model with low bias and low variance, the model will have a good generalization for all test

samples. Even with a “perfect” fit, the model must account for random noise and hence, the irreducible error(random noise) will always be present.

Q10 Link to handwritten proof: https://drive.google.com/file/d/1kgP5bkUw42K_D5GH507fN9N2_175wAbf/view?usp=sharing