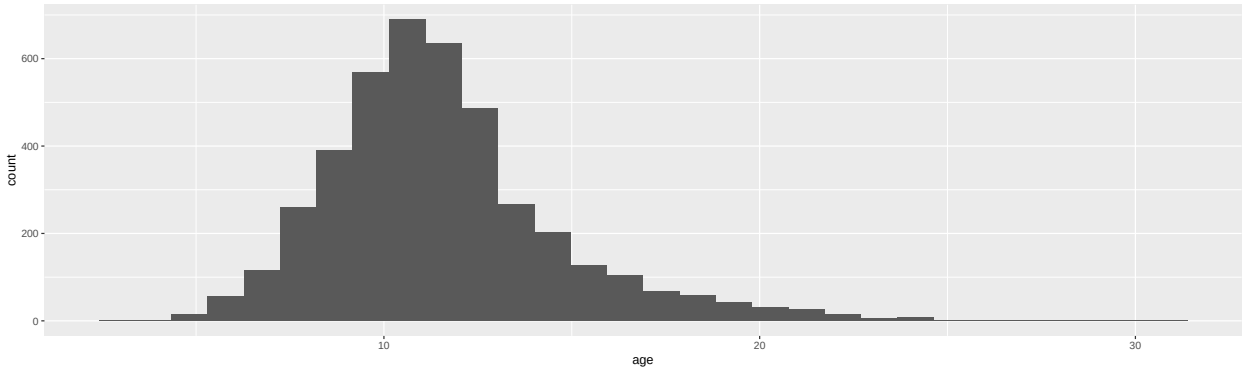
PSTAT131_HW2_DWH

Dongwoo (Kevin) Heo

2022-10-15

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`.

```
options(readr.show_col_types=FALSE)
abalone = read.csv("abalone.csv")
abalone$age <- abalone$rings + 1.5 ## Add `age` to the data set and predict abalone age
                                   ## which is calculated as the number of rings plus 1.5
ggplot(abalone, aes(age)) + geom_histogram()
```



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(1234) ## Set a seed at the beginning of the document to produce your results
data_split <- initial_split(abalone, prop = 4/5, strata = age)
data_train <- training(data_split) ## Split the abalone data into a training set
data_test <- testing(data_split)   ## Split the abalone data into a testing set
```

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 2. create interactions between - `type` and `shucked_weight`, - `longest_shell` and `diameter`, - `shucked_weight` and `shell_weight` 3. center all predictors, and 4. scale all predictors. You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

```
some_var <- names(data_train)[9]
aba_recipe <- recipe(formula = age~., data = data_train) %>%
  step_rm(some_var) %>%
  step_dummy(all_nominal_predictors()) %>%
  ## dummy code any categorical predictors
  step_interact(terms = ~ starts_with('type'):shucked_weight) %>%
  ## create interactions between `type` and `shucked_weight`
  step_interact(terms = ~ longest_shell:diameter) %>%
  ## create interactions between `longest_shell` and `diameter`
  step_interact(terms = ~ shucked_weight:shell_weight) %>%
  ## create interactions between `shucked_weight` and `shell_weight`
  step_center(all_numeric_predictors()) %>%
  step_scale(all_numeric_predictors()) %>%

  prep()
```

Question 4. Create and store a linear regression object using the "lm" engine.

```
lm_model <- linear_reg() %>% ## created and stored a linear regression object
  set_engine("lm")           ## Using "lm" engine
```

Question 5. Now:

```
workflow <- workflow() %>%  ## 1. set up an empty workflow,
  add_model(lm_model) %>%   ## 2. added the model I have created in Question 4
  add_recipe(aba_recipe)    ## 3. added the recipe that you created in Question 3.
```

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.

```
model <- fit(workflow,data_train)
newdata <- 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, rings = 0)
pre <- predict(model, new_data = newdata)
cat("Predicted age of the abalone will be ", unlist(pre[1]), ".", sep = "")
```

```
## Predicted age of the abalone will be 23.48823.
```

Question 7. Now you want to assess your model's performance. To do this, use the yardstick package:

1. Create a metric set that includes R^2, 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^2 value.

```
data_train_pred <- predict(model, new_data = data_train %>% select(-age))
data_train_pred <- bind_cols(data_train_pred, data_train %>% select(age))
metrics <- metric_set(rsq, rmse, mae)
metrics(data_train_pred, 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       1.55
```

I would say by evaluating the model, we get a value of 2.15, 0.558 of the r-square value, and 1.55 of mae value. And from rmse, we know that the average distance between the observation and the prediction is approximately 2.15 years. Yet, the model only has an r-square of 0.558, and only around 55.8% of the variance in age could be demonstrated by the predictor variables in the model. Therefore, it can be explained that the model's performance is not good enough.