PSTAT131, Homework 2

Sofia Spasibenko

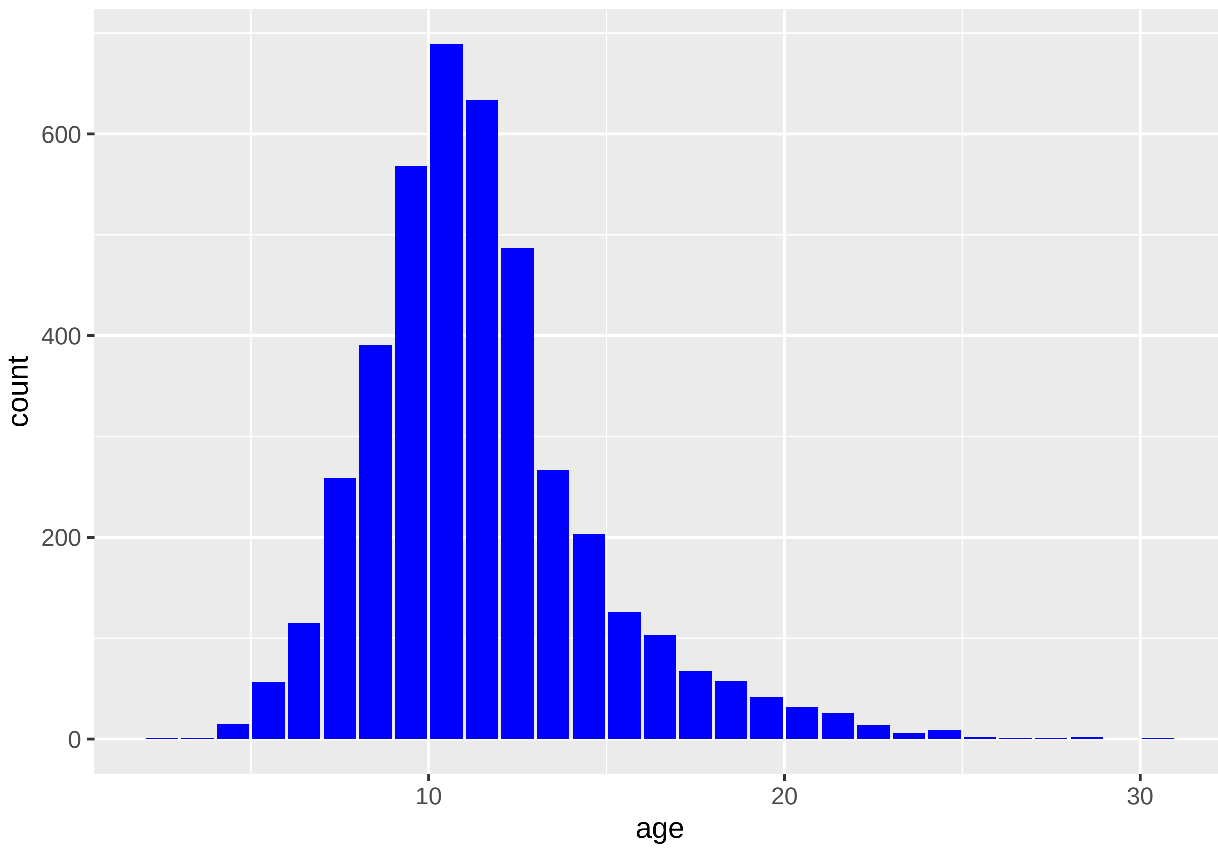
Linear Regression

```
abalone = read_csv("/Users/Sofia/Desktop/PSTAT131/homework-2/data/abalone.csv")
```

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.

```
abalone <- mutate(abalone, age = rings + 1.5)
ggplot(abalone, aes(x=age))+ geom_bar(stat = 'count', fill = 'blue')
```



Assess and describe the distribution of age.

The distribution of age seems to be a right-skewed Normal curve centered around 10.

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(128)

abalone_split <- initial_split(abalone, prop = 0.70,
                               strata = age)
abalone_tr <- training(abalone_split)
abalone_te <- testing(abalone_split)
```

Remember that you'll need to set a seed at the beginning of the document to reproduce your results.

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`.

We shouldn't use rings to predict age since we are trying to predict their age by using their outside appearance.

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.

```
abalone_recipe <- recipe(age ~ type + longest_shell + diameter + height +
                           whole_weight + shucked_weight + viscera_weight +
                           shell_weight, data = abalone_tr) %>%
  step_dummy(type) %>%  #dummy code for type in abalone
  step_interact(terms = ~ shucked_weight:starts_with("type_") ) %>%  #interactions
  step_interact(terms = ~ longest_shell:diameter) %>%
  step_interact(terms = ~ shucked_weight:shell_weight) %>%
  step_center(all_numeric_predictors()) %>%  #center all predictors
  step_scale(all_numeric_predictors()) %>% #scale all predictors
  prep()

bake(abalone_recipe, new_data = NULL)
```

```
## # A tibble: 2,922 x 14
##    longest_~1 diame~2 height whole~3 shuck~4 visce~5 shell~6   age type_I type_M
##         <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>  <dbl>  <dbl>
##  1     -1.61   -1.53  -1.51   -1.26   -1.21   -1.28   -1.31    8.5  1.46  -0.763
##  2     -0.819  -1.08  -1.13   -0.967  -0.975  -0.936  -0.845   9.5  1.46  -0.763
##  3     -1.40   -1.28  -1.38   -1.09   -1.18   -1.28   -0.881   8.5  1.46  -0.763
##  4     -0.487  -0.528 -0.871  -0.706  -0.589  -0.512  -0.809   9.5 -0.685  1.31
##  5     -0.611  -0.528 -0.871  -0.618  -0.544  -0.580  -0.667   9.5 -0.685 -0.763
##  6     -2.36   -2.34  -2.41   -1.54   -1.47   -1.43   -1.56    6.5  1.46  -0.763
```

```
## 7     -2.65   -2.59  -2.15   -1.60   -1.49   -1.50   -1.62    6.5  1.46  -0.763
## 8     -2.61   -2.59  -2.28   -1.60   -1.53   -1.53   -1.59    5.5  1.46  -0.763
## 9     -0.528  -0.326 -0.488  -0.745  -0.811  -0.640  -0.631   8.5 -0.685 -0.763
## 10    -1.65   -1.63  -1.77   -1.35   -1.27   -1.41   -1.38    7.5  1.46  -0.763
## # ... with 2,912 more rows, 4 more variables: shucked_weight_x_type_I <dbl>,
## #   shucked_weight_x_type_M <dbl>, longest_shell_x_diameter <dbl>,
## #   shucked_weight_x_shell_weight <dbl>, and abbreviated variable names
## #   1: longest_shell, 2: diameter, 3: whole_weight, 4: shucked_weight,
## #   5: viscera_weight, 6: shell_weight
```

```
# abalone_mod <- abalone_recipe
```

Question 4

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

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

Question 5

Now:

1. set up an empty workflow,
2. add the model you created in Question 4, and
3. add the recipe that you created in Question 3.

```
lm_wflow <- workflow() %>%     #empty workflow
  add_model(lm_model) %>%      #model from Question 4
  add_recipe(abalone_recipe)   #recipe from 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.

```
lm_fit <- fit(lm_wflow, abalone_tr)
new <- 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 = new)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  23.3
```

Our model predicts 23.29291 as the age value.

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.

```r
multi_metric <- metric_set(rsq, rmse, mae)  #1. make the metric set
abalone_tr_res <- predict(lm_fit,
                          new_data = abalone_tr %>% select(-(rings:age))) #2. making tibble
abalone_tr_res <- bind_cols(abalone_tr_res, abalone_tr %>% select(age))
abalone_tr_res %>%
  head()
```

```
## # A tibble: 6 x 2
##   .pred   age
##   <dbl> <dbl>
## 1  8.05   8.5
## 2  9.28   9.5
## 3  9.63   8.5
## 4 10.0    9.5
## 5 10.8    9.5
## 6  6.09   6.5
```

```r
multi_metric(abalone_tr_res, truth = age,  #3. applying metric set
             estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.560
## 2 rmse    standard       2.13
## 3 mae     standard       1.53
```

Our model performance returns a 0.5603852 R^2 value, a 2.1294926 root mean squared error value, and a 1.5328272 mean absolute error value. The R^2 value tells us that about 56% of variability in age can be explained by the predictors, which isn't particularly high, confirming that abalone ages are hard to predict.