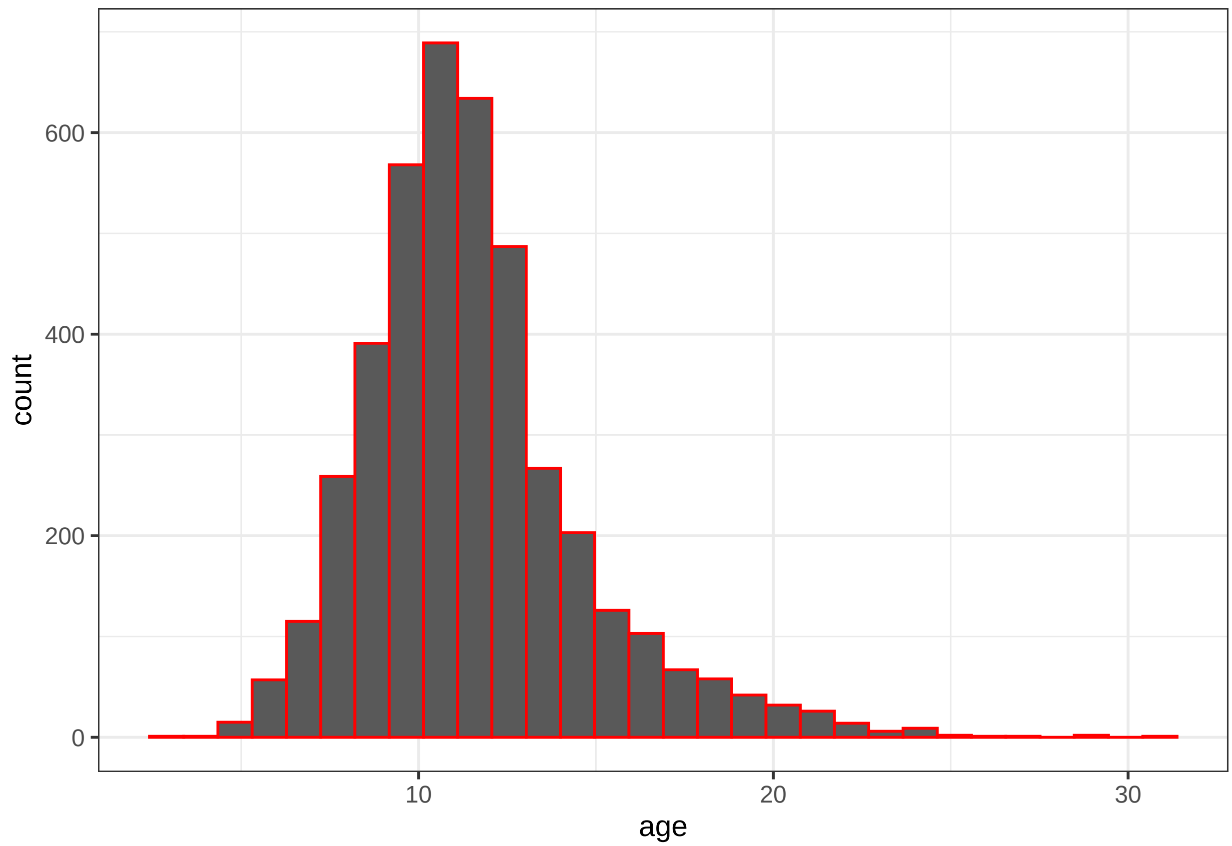
HW2

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`.

```
library(tidyverse)
library(tidymodels)
abalone<-read_csv('abalone.csv')
age<-abalone$rings+1.5
abalone<-abalone %>%
  mutate(age)
abalone %>%
  ggplot(aes(x = age)) + geom_histogram(color='red', bins = 30) +  theme_bw()
```



age is positively skewed, and a long tail to the right. Most of the age values in the data set are below 20.

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(2221)
abalone_split <- initial_split(abalone, prop = 0.80, strata = age)
abalone_train <- training(abalone_split)
abalone_test <- 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`.

Because the value of `age` is calculated by adding 1.5 to `rings`, there is a linear relationship between them, and the correlation coefficient is 1.

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.

```
abalone_train<-abalone_train[,-9]
abalone_recipe <- recipe(age ~ ., data = abalone_train) %>%
  step_dummy(all_nominal_predictors())

abalone_recipe_model<-abalone_recipe %>%
  step_interact(terms = ~ shucked_weight:starts_with("type")) %>%
  step_interact(terms = ~ longest_shell:diameter) %>%
  step_interact(terms = ~ shucked_weight:shell_weight)  %>%
  step_center(all_numeric(), -all_outcomes(), -has_role('id variable')) %>%
  step_scale(all_numeric(), -all_outcomes(), -has_role('id variable'))
abalone_train_1<-abalone_recipe_model %>%
  prep() %>%
  bake(new_data = abalone_train)
```

You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

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() %>%
  add_model(lm_model) %>%
  add_recipe(abalone_recipe_model)
```

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.

```
data<-data.frame(type="F",longest_shell=0.5,diameter = 0.10,
                 height = 0.30, whole_weight = 4,
                 shucked_weight = 1, viscera_weight = 2,
                 shell_weight = 1)
data<-tibble(data)
lm_fit <- fit(lm_wflow, abalone_train)
predict(lm_fit, new_data = data)
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  23.6
```

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
diamond_metrics <- metric_set(rmse, rsq, mae)

abalone_train_res <- predict(lm_fit, new_data = abalone_train %>% select(-age))
abalone_train_res <- bind_cols(abalone_train_res, abalone_train %>% select(age))

diamond_metrics(abalone_train_res, truth = age,
                estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.14
## 2 rsq     standard       0.570
## 3 mae     standard       1.53
```

In the multiple linear regression, 57% of age can be determined by explain these variables.