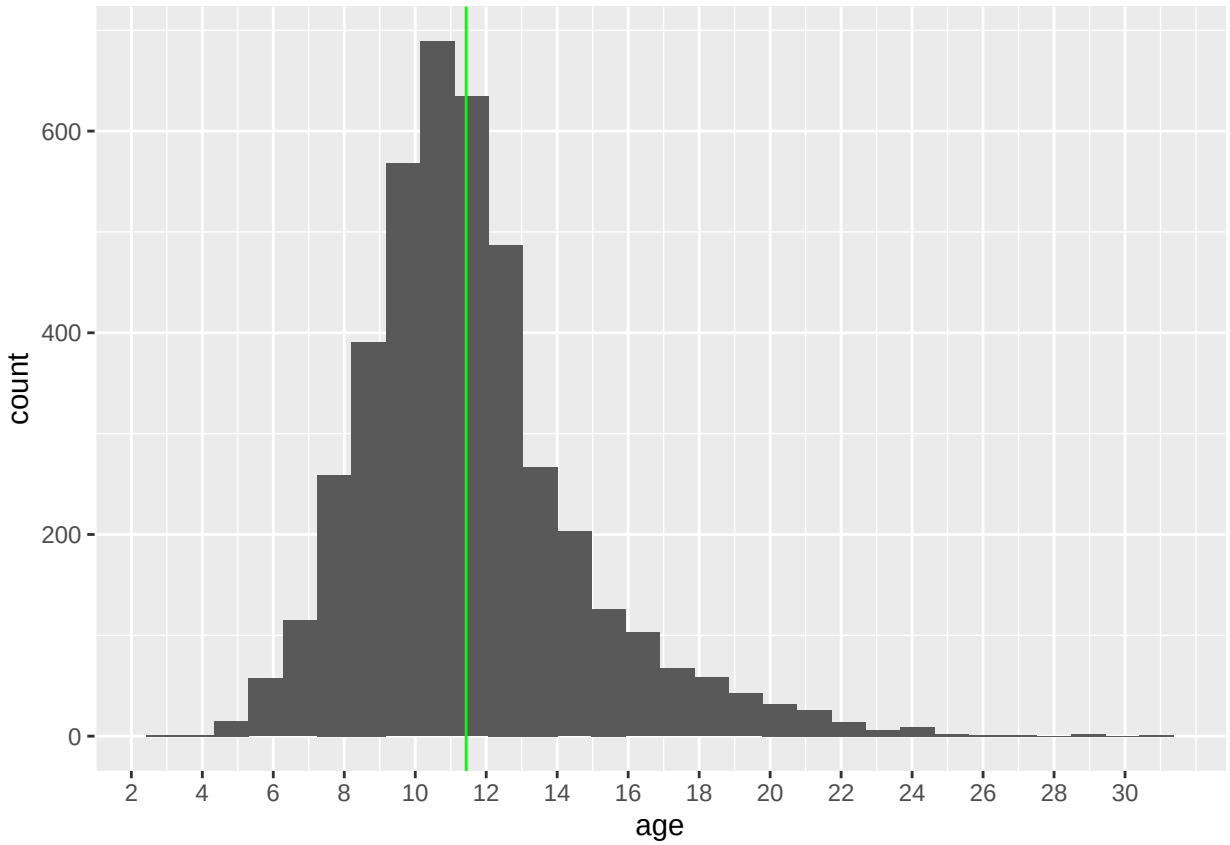
PSTAT131HW#2

2022-04-06

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## -- Attaching packages --------------------------------------- tidymodels 0.2.0 --

## v broom        0.7.12     v rsample      0.1.1
## v dials        0.1.0      v tune         0.2.0
## v infer        1.0.0      v workflows    0.2.6
## v modeldata    0.1.1      v workflowsets 0.2.1
## v parsnip      0.2.1      v yardstick    0.0.9
## v recipes      0.2.0

## -- 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()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

1)

```
abalone.data$age <- abalone.data$rings +1.5
ggplot(abalone.data, aes(x=age)) +
  geom_histogram(bins=30) +
  scale_x_continuous(breaks=seq(0, 30, 2)) +
  geom_vline(aes(xintercept=mean(age)), col='green')
```



This is clearly a right skewed distrbution with finding out the the mean is around 12 with a standard deviation around 3

2)

```
p <- 0.7
strats <- abalone.data$type

rr <- split(1:length(strats), strats)
idx <- sort(as.numeric(unlist(sapply(rr, function(x) sample(x, length(x) * p)))))

train <- abalone.data[idx, ]
test <- abalone.data[-idx, ]
```

3)

```
train <- train %>% na.omit()
recipe_normal <-
  recipe(age ~ type + longest_shell + diameter + height + whole_weight + shucked_weight + viscera_weigh
  step_dummy(type, one_hot = F) %>%
  step_interact(terms = ~starts_with('type'):shucked_weight)
  step_interact(recipe_normal, terms = ~longest_shell:diameter)
```

```
## Recipe
##
```

```
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from type
## Interactions with starts_with("type"):shucked_weight
## Interactions with longest_shell:diameter
```

```
  step_interact(recipe_normal, terms = ~shucked_weight:shell_weight)
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from type
## Interactions with starts_with("type"):shucked_weight
## Interactions with shucked_weight:shell_weight
```

```
  step_center(recipe_normal) %>%
  step_scale(recipe_normal)
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Dummy variables from type
## Interactions with starts_with("type"):shucked_weight
## Centering for <none>
## Scaling for recipe_normal
```

4)

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

5)

```
abalone_workflow <- workflow() %>%
                        add_model(lm_model) %>%
                        add_recipe(recipe_normal)
```

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.

```
abalone_fit <- abalone_workflow %>%
                    fit(train)
predict.sample <- list(longest_shell = .5,
                       diameter = .10,
                       height = .3,
                       whole_weight = 4,
                       shucked_weight =1,
                       viscera_weight = 2,
                       shell_weight =1,
                       type = 'F')
predict.sample <- as.data.frame(predict.sample)
predict(abalone_fit, predict.sample) %>% unlist() %>% unname()
```

```
## [1] 17.116
```

7)

```
library(yardstick)
multimetric = metric_set(rsq, rmse, mae)
boundtestdata = bind_cols(predict(abalone_fit, train),
                          train$age)
```

```
## New names:
## * `` -> ...2
```

```
colnames(boundtestdata) = c("Predicted Age", "True Age")
multimetric(data=boundtestdata,
            truth="True Age",
            estimate="Predicted Age")
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.549
## 2 rmse    standard       2.16
## 3 mae     standard       1.54
```

Looking at the values provided we can see that we get an R^2 value of 0.54 which represents that our model performs very poorly in having the response predicted by the predictor variables.

‘‘‘