PSTAT 231 HW2 muxi

muxi

2022-10-22

PSTAT 231 Homework 2

Question 1

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────────
────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ──────────────────────────────────────────────────
───── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## ── Attaching packages ─────────────────────────────────────────
───── tidymodels 1.0.0 ──
## ✓ broom        1.0.1     ✓ rsample      1.1.0
## ✓ dials        1.0.0     ✓ tune         1.0.1
## ✓ infer        1.0.3     ✓ workflows    1.1.0
## ✓ modeldata    1.0.1     ✓ workflowsets 1.0.0
## ✓ parsnip      1.0.2     ✓ yardstick    1.1.0
## ✓ recipes      1.0.2
## ── Conflicts ──────────────────────────────────────────────────
──── tidymodels_conflicts() ──
## ✕ scales::discard() masks purrr::discard()
## ✕ dplyr::filter()   masks stats::filter()
## ✕ recipes::fixed()  masks stringr::fixed()
## ✕ dplyr::lag()      masks stats::lag()
## ✕ yardstick::spec() masks readr::spec()
## ✕ recipes::step()   masks stats::step()
## • Use suppressPackageStartupMessages() to eliminate package startup messages
```

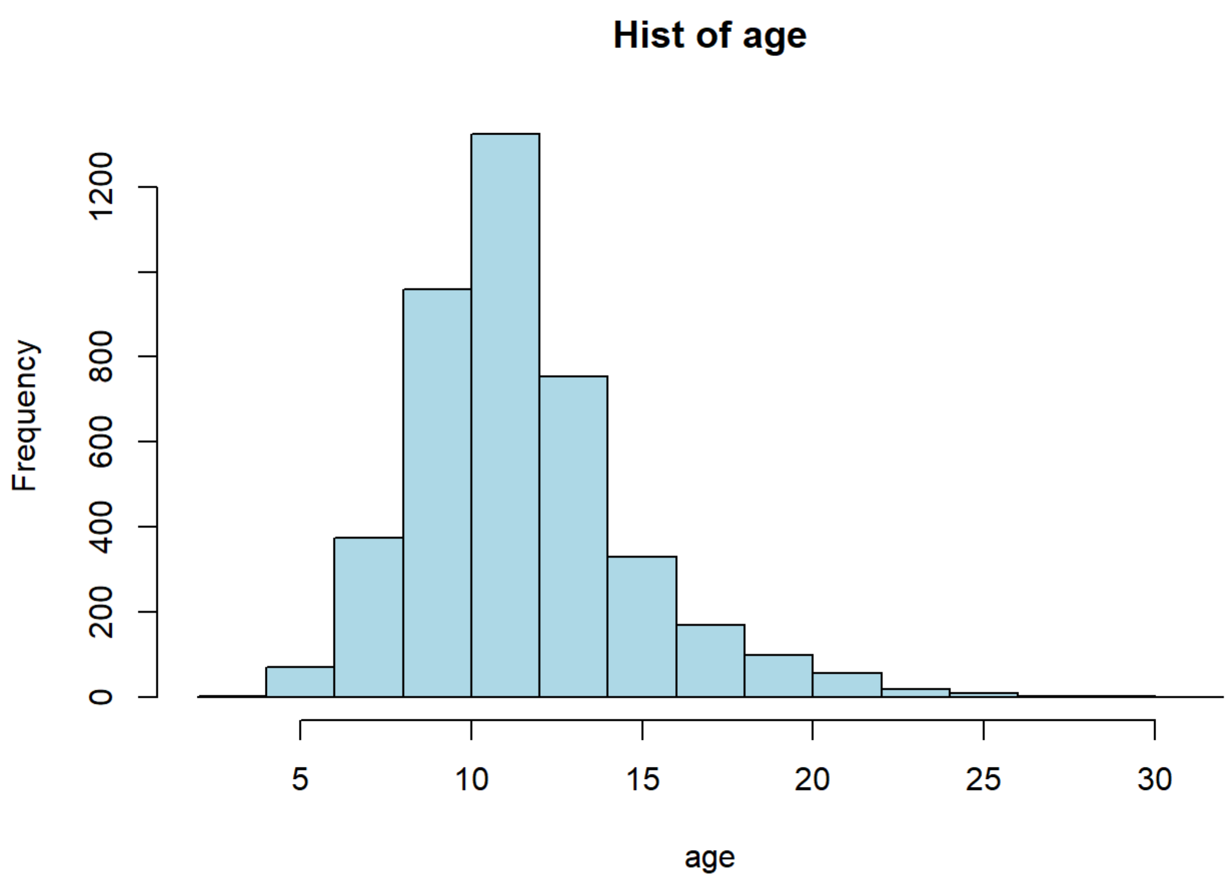
```
data=read.csv("abalone.csv")
head(data)
```

```
##   type longest_shell diameter height whole_weight shucked_weight viscera_weight
## 1    M         0.455    0.365  0.095       0.5140         0.2245         0.1010
## 2    M         0.350    0.265  0.090       0.2255         0.0995         0.0485
## 3    F         0.530    0.420  0.135       0.6770         0.2565         0.1415
## 4    M         0.440    0.365  0.125       0.5160         0.2155         0.1140
## 5    I         0.330    0.255  0.080       0.2050         0.0895         0.0395
## 6    I         0.425    0.300  0.095       0.3515         0.1410         0.0775
##   shell_weight rings
## 1        0.150    15
## 2        0.070     7
## 3        0.210     9
## 4        0.155    10
## 5        0.055     7
## 6        0.120     8
```

```
data=mutate(data,age=rings+1.5)
summary(data$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.50    9.50   10.50   11.43   12.50   30.50
```

```
hist(data$age,xlab="age",main="Hist of age",col="lightblue")
```



To begin with, I believe that age could be treated as quantitative predictor. Though rings are always integers, we could use the raw data as a estimate of the exact age.

From summary and hist graph, we could see that the age is right skewed and there is no obvious outlier.

Question 2

```
set.seed(1215)
data_split = initial_split(data, prop = 0.80)
data_train = training(data_split)
data_test = testing(data_split)
```

Question 3

As age and rings are strongly positive correlated(age = rings + 1.5), the residuals plot would be a level line through residuals=0. This will remove error term, lead to overfitting and make any other predictors meaningless.

```
#drop rings column
train=select(data_train,-c(rings))
test=select(data_test,-c(rings))
simple_data_recipe=recipe(age ~ ., data = train)
summary(simple_data_recipe)
```

```
## # A tibble: 9 × 4
##   variable       type    role      source
##   <chr>          <chr>   <chr>     <chr>
## 1 type           nominal predictor original
## 2 longest_shell  numeric predictor original
## 3 diameter       numeric predictor original
## 4 height         numeric predictor original
## 5 whole_weight   numeric predictor original
## 6 shucked_weight numeric predictor original
## 7 viscera_weight numeric predictor original
## 8 shell_weight   numeric predictor original
## 9 age            numeric outcome   original
```

```
data_recipe = recipe(age~ ., data = train)
recipe=data_recipe%>%
  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_nominal_predictors())%>%
  step_scale(all_nominal_predictors())
```

Question 4

```
lm_model = linear_reg() %>%
  set_engine("lm")
```

Question 5

```
lm_wflow = workflow() %>%
  add_model(lm_model) %>%
  add_recipe(recipe)
lm_fit = fit(lm_wflow, train)
summary(lm_fit)
```

```
##         Length Class      Mode
## pre     3      stage_pre  list
## fit     2      stage_fit  list
## post    1      stage_post list
## trained 1      -none-     logical
```

Question 6

```
pre=train[1,]
pre[2:8]=c(0.5,0.1,0.3,4,1,2,1)
predict(lm_fit, pre)
```

```
## # A tibble: 1 × 1
##   .pred
##   <dbl>
## 1  24.6
```

Question 7

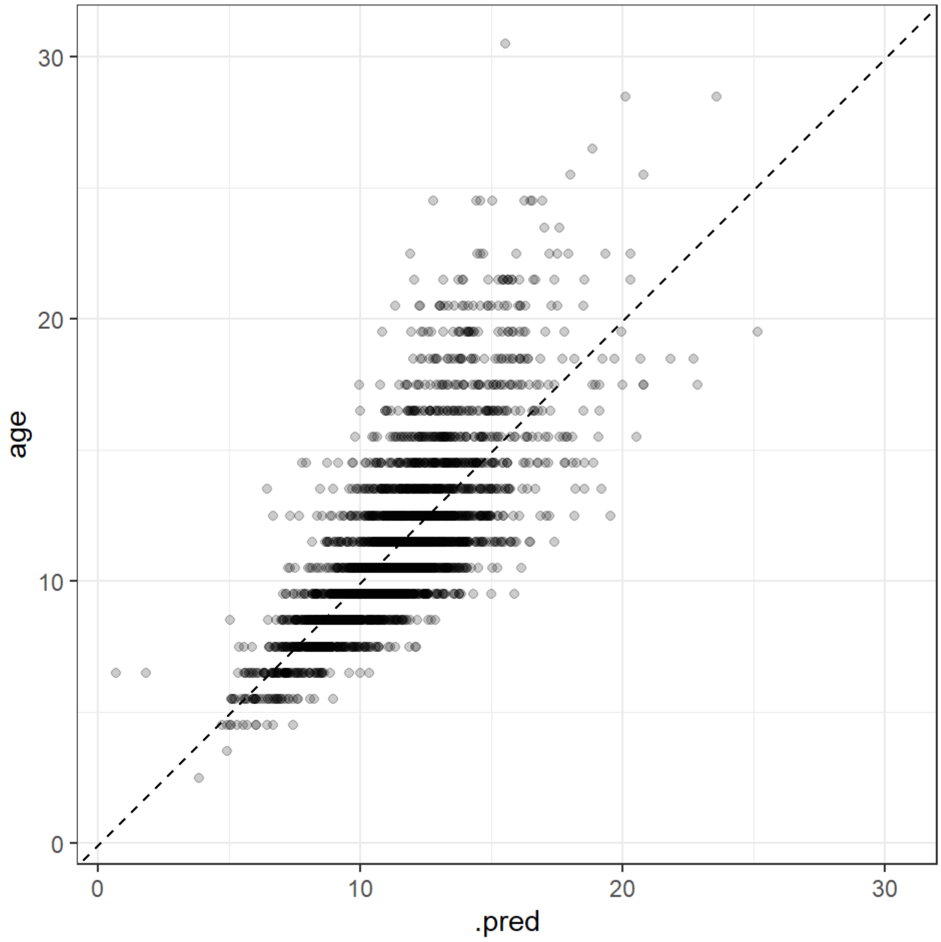
```
library(yardstick)
train_res = predict(lm_fit, new_data =train %>% select(-age))
#predicted values vs the actual observed ages
train_res = bind_cols(train_res, train %>% select(age))
train_res %>%
  head()
```

```
## # A tibble: 6 × 2
##   .pred   age
##   <dbl> <dbl>
## 1 13.9   18.5
## 2  8.61   8.5
## 3 11.2    9.5
## 4 13.2   14.5
## 5 11.0   13.5
## 6  9.17   8.5
```

```
#R2, RMSE, and MAE
metrics = metric_set(rmse, rsq, mae)
metrics(train_res, truth = age, estimate = .pred)
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.12
## 2 rsq     standard       0.564
## 3 mae     standard       1.54
```

```
train_res %>%
  ggplot(aes(x = .pred, y = age)) +
  geom_point(alpha = 0.2) +
  geom_abline(lty = 2) +
  theme_bw() +
  coord_obs_pred()
```



From R-square and plot, we could see that the model didn't do very well. If it predicted every observation accurately, the dots would form a straight line. Perhaps in the future, I will try other models and other interaction methods dealing with type and shucked_weight.

Question 8

Reproducible errors are $Var(\hat{f}(x_0))$, $[Bias(\hat{f}(x_0))]^2$.

Irreducible error is $Var(\epsilon)$.

Question 9

$\because Var(\hat{f}(x_0)) > 0, [Bias(\hat{f}(x_0))]^2 > 0$

$\therefore E[(y_0 - \hat{f}(x_0))^2] \geq Var(\epsilon)$

Question 10

$E[(y_0 - \hat{f}(x_0))^2] = E[y_0^2] - 2E[y_0]E[\hat{f}(x_0)] + E[\hat{f}(x_0)^2]$

$= Var(\epsilon) + E[y_0]^2 - 2E[y_0]E[\hat{f}(x_0)] + E[\hat{f}(x_0)]^2 - E[\hat{f}(x_0)]^2 + E[\hat{f}(x_0)^2]$

$= Var(\epsilon) + (E[\hat{f}(x_0)] - y_0)^2 + Var(\hat{f}(x_0))$

$= Var(\epsilon) + [Bias(\hat{f}(x_0))]^2 + Var(\hat{f}(x_0))$