131 HW2

Danny

2022-10-016

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggthemes)
library(yardstick)
```

```
## For binary classification, the first factor level is assumed to be the event.
## Use the argument `event_level = "second"` to alter this as needed.
```

```
library(readr)
```

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:yardstick':
##
##     spec
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x readr::spec()   masks yardstick::spec()
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.1     v recipes      1.0.1
## v dials        1.0.0     v rsample      1.1.0
## v infer        1.0.3     v tune         1.0.0
## v modeldata    1.0.1     v workflows    1.1.0
## v parsnip      1.0.2     v workflowsets 1.0.0
## -- 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 readr::spec()     masks yardstick::spec()
## x recipes::step()   masks stats::step()
```

```
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
data <- read_csv("~/Downloads/homework-2/data/abalone.csv")
```

```
## Rows: 4177 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): type
## dbl (8): longest_shell, diameter, height, whole_weight, shucked_weight, visc...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
view(data)
```

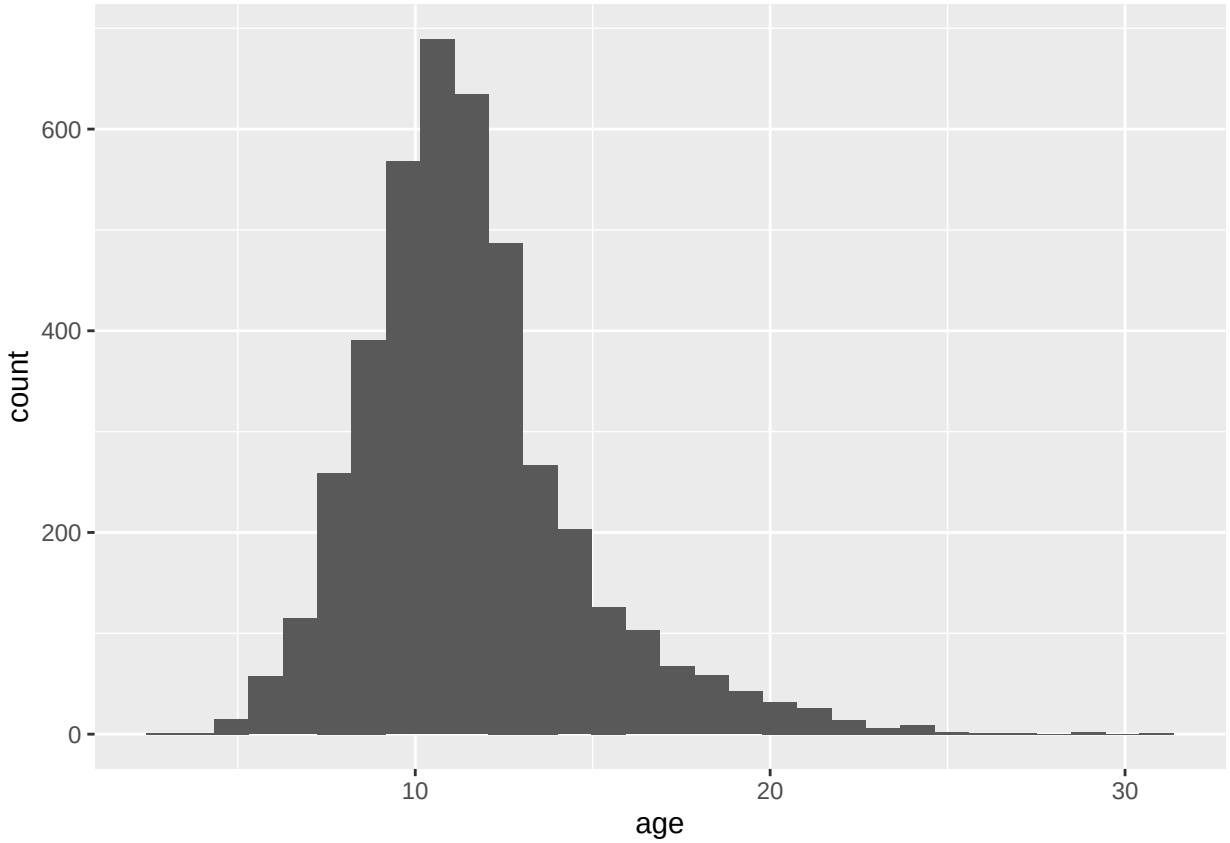
#Exercise 1:

```
abalone<- data %>%
  mutate(data, age =rings+1.5)
head(abalone)
```

```
## # A tibble: 6 x 10
##   type  longest_shell diameter height whole_weight shucked_weight viscera_weight
##   <chr>         <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
## 1 M             0.455    0.365  0.095        0.514         0.224          0.101
## 2 M             0.35     0.265  0.09         0.226         0.0995         0.0485
## 3 F             0.53     0.42   0.135        0.677         0.256          0.142
## 4 M             0.44     0.365  0.125        0.516         0.216          0.114
## 5 I             0.33     0.255  0.08         0.205         0.0895         0.0395
## 6 I             0.425    0.3    0.095        0.352         0.141          0.0775
## # ... with 3 more variables: shell_weight <dbl>, rings <dbl>, age <dbl>
```

```
ggplot(data = abalone, aes(age)) +
 geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



This is right skewed distribution, and age 10 has the largest amount. There are few outliers around age 30

#Exercise 2:

```
set.seed(1000)
abalone_split <- initial_split(abalone, prop = 0.80,strata = age)
abalone_train <- training(abalone_split)
abalone_test <- testing(abalone_split)
```

#Exercise 3:

```
train <- abalone_train %>%
  select(-rings)
abalone_recipe <- recipe(age ~ ., data = train) %>%
 step_dummy(all_nominal_predictors()) %>%
 step_interact(terms = ~ starts_with("type"):shucked_weight
               +longest_shell:diameter
               +shucked_weight:shell_weight)%>%
 step_center(all_nominal_predictors()) %>%
 step_scale(all_nominal_predictors())
```

We shouldn't use rings to predict age. Because age = rings + 1.5, they have exactly the same distribution and trend

#Exercise 4:

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

#Exercise 5:

```
lm_wflow <- workflow() %>%
  add_model(lm_model) %>%
  add_recipe(abalone_recipe)
```

#Exercise 6:

```
lm_fit <- fit(lm_wflow, train)
frame <- data.frame(type = "F",longest_shell = 0.5,diameter = 0.1, height = 0.3,
 whole_weight = 4, shucked_weight = 1, viscera_weight = 2, shell_weight = 1)
lm_fit %>%
 extract_fit_parsnip() %>%
 tidy()
```

```
## # A tibble: 14 x 5
##    term                          estimate std.error statistic  p.value
##    <chr>                            <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                      4.41      0.679     6.49  9.93e-11
##  2 longest_shell                    2.59      2.40      1.08  2.81e- 1
##  3 diameter                        22.9       3.20      7.17  8.98e-13
##  4 height                           4.79      1.64      2.92  3.52e- 3
##  5 whole_weight                     9.82      0.815    12.1   8.62e-33
##  6 shucked_weight                 -18.9       1.15    -16.5   5.29e-59
##  7 viscera_weight                  -8.43      1.45     -5.81  6.85e- 9
##  8 shell_weight                    12.6       1.57      8.04  1.21e-15
##  9 type_I                          -1.95      0.245    -7.98  2.01e-15
## 10 type_M                          -0.504     0.213    -2.36  1.82e- 2
## 11 type_I_x_shucked_weight          4.00      0.741     5.39  7.55e- 8
## 12 type_M_x_shucked_weight          1.05      0.436     2.41  1.61e- 2
## 13 longest_shell_x_diameter       -29.3       4.21     -6.96  4.01e-12
## 14 shucked_weight_x_shell_weight   -1.38      1.74     -0.793 4.28e- 1
```

```
train_res<-predict(lm_fit, new_data = frame)
train_res
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  22.7
```

```
head
```

```
## function (x, ...)
## UseMethod("head")
## <bytecode: 0x7fdf340ec9f8>
## <environment: namespace:utils>
```

#Exercise 7:

```
abalone_train_res <- predict(lm_fit, new_data = train %>% select(-age))
abalone_train_res %>%
  head()
```

```
## # A tibble: 6 x 1
##   .pred
##   <dbl>
## 1  8.10
## 2  9.33
```

```
## 3 10.5
## 4 10.1
## 5 11.0
## 6  6.35
```

```
abalone_train_res <- bind_cols(abalone_train_res, train %>% select(age))
abalone_train_res%>%
  head()
```

```
## # A tibble: 6 x 2
##   .pred   age
##   <dbl> <dbl>
## 1  8.10   8.5
## 2  9.33   9.5
## 3 10.5    8.5
## 4 10.1    9.5
## 5 11.0    9.5
## 6  6.35   6.5
```

```
abalone_metrics<-metric_set(rmse,rsq,mae)
abalone_metrics(abalone_train_res, truth=age,estimate=.pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.15
## 2 rsq     standard       0.562
## 3 mae     standard       1.55
```

the value of R^2 is 0.5618, which means 56.18% of the data fit the regression model.