hw3

2022-10-18

Question 1:

```
titanic_split <- initial_split(titanic,
                               prop = 0.80,
                               strata = survived)
titanic_split
```

```
## <Training/Testing/Total>
## <712/179/891>
```

```
train <- training(titanic_split)
test <- testing(titanic_split)
```

The training data set has 712 observations and the testing data set has 179 observations. This makes sense as the total observations the titanic data set has is 891, which is 179 + 712.

There seems to be a lot of missing data for the variable cabin with 552 missing values; age has 136 missing values and embarked variable has 2 missing values.

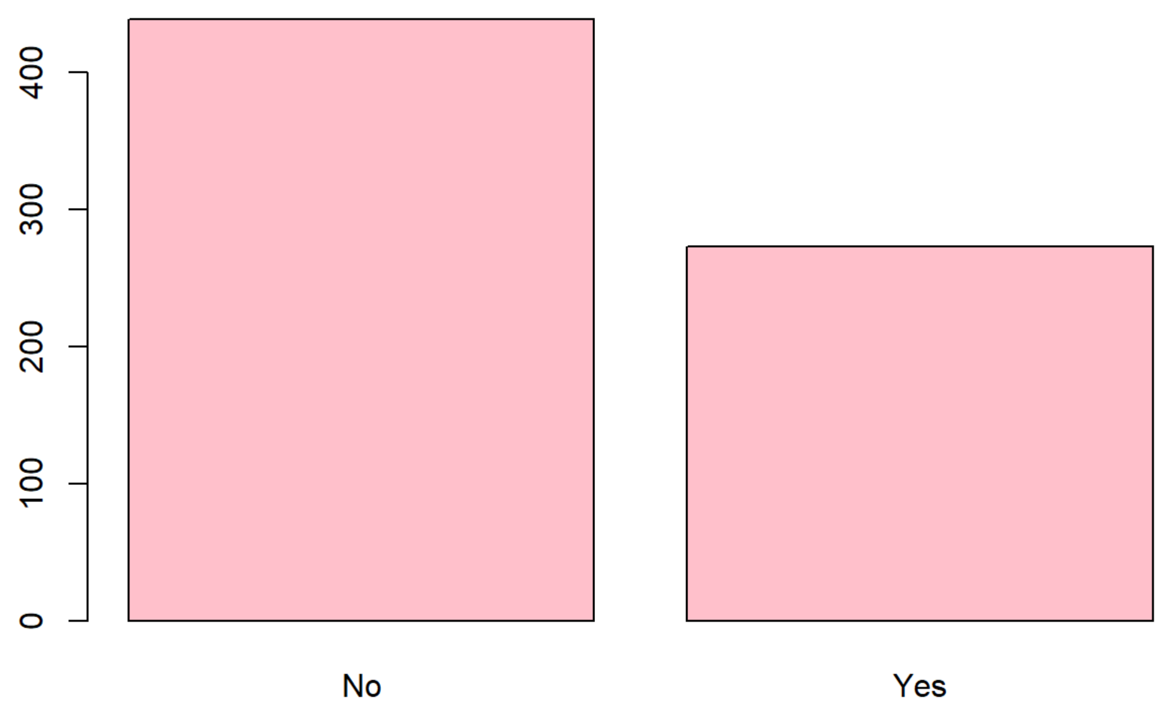
It is good to use stratified sampling for this data as we want to see the characteristics for each of the survivor and non-survivor groups and see why the survivor group is more likely to survive.

Question 2:

```
dist_surv <- table(train$survived)
dist_surv
```

```
##
##  No Yes
## 439 273
```

```
barplot((table(train$survived)), col = "pink")
```

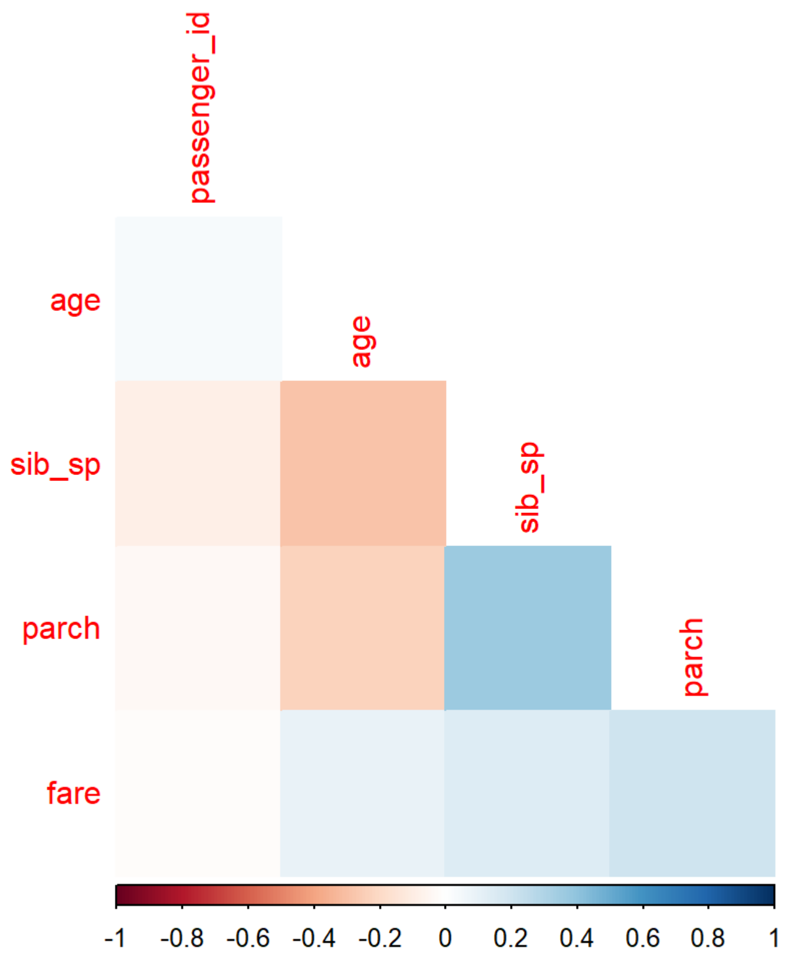


There seems to be more values for not surviving (439 observations) than surviving (273 observations).

Question 3:

```
#install.packages("corrr")
library(corrr)
library(ggplot2)

train %>%
  select_if(is.numeric) %>%
  cor(use = "complete.obs") %>%
  corrplot(type = "lower", diag = FALSE, method = "color")
```



I see that age and sib_sp are negatively correlated, and age and parch are also negatively correlated but not as high as age and sib_sp. Meanwhile sib_sp and parch are almost highly postively correlated. The other variables have little to no correlation.

Question 4:

```
titanic_recipe <-
  recipe(survived ~ pclass+sex+age+sib_sp+parch+fare, data = train) %>%
  step_impute_linear(age,
                     impute_with = imp_vars(sib_sp, parch, fare)) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~starts_with("sex"):fare + age:fare)
titanic_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          6
##
## Operations:
##
## Linear regression imputation for age
## Dummy variables from all_nominal_predictors()
## Interactions with starts_with("sex"):fare + age:fare
```

```
#summary(train)
```

Question 5

```
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)

log_fit <- fit(log_wkflow, train)

log_fit %>%
  tidy()
```

```
## # A tibble: 10 × 5
##    term            estimate std.error statistic  p.value
##    <chr>              <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)      3.86      0.613      6.30    2.99e-10
##  2 age             -0.0449    0.0116    -3.88    1.03e- 4
##  3 sib_sp          -0.463     0.123     -3.77    1.66e- 4
##  4 parch            0.000920  0.136      0.00675 9.95e- 1
##  5 fare             0.00785   0.0106     0.741   4.59e- 1
##  6 pclass_X2       -1.14      0.340     -3.34    8.35e- 4
##  7 pclass_X3       -2.24      0.349     -6.41    1.42e-10
##  8 sex_male        -2.51      0.299     -8.40    4.44e-17
##  9 sex_male_x_fare -0.0126    0.00833   -1.51    1.30e- 1
## 10 fare_x_age       0.000146  0.000187   0.781   4.35e- 1
```

Question 6

```
lda_mod <- discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")

lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(titanic_recipe)

lda_fit <- fit(lda_wkflow, train)

lda_fit
```

```
## ══ Workflow [trained] ══════════════════════════════════════════════════════════
## Preprocessor: Recipe
## Model: discrim_linear()
##
## ── Preprocessor ────────────────────────────────────────────────────────────────
## 3 Recipe Steps
##
## • step_impute_linear()
## • step_dummy()
## • step_interact()
##
## ── Model ───────────────────────────────────────────────────────────────────────
## Call:
## lda(..y ~ ., data = data)
##
## Prior probabilities of groups:
##       No      Yes
## 0.616573 0.383427
##
## Group means:
##          age    sib_sp     parch     fare pclass_X2 pclass_X3  sex_male
## No  30.74438 0.5375854 0.2984055 22.79547 0.1936219 0.6583144 0.8473804
## Yes 28.73447 0.4908425 0.5164835 47.90606 0.2600733 0.3443223 0.2967033
##     sex_male_x_fare fare_x_age
## No         19.46233   723.4217
## Yes        11.54489  1500.2063
##
## Coefficients of linear discriminants:
##                           LD1
## age             -2.574225e-02
## sib_sp          -2.507609e-01
## parch            2.062065e-02
## fare            -1.984204e-04
## pclass_X2       -7.658707e-01
## pclass_X3       -1.487011e+00
## sex_male        -2.124257e+00
## sex_male_x_fare -1.425555e-03
## fare_x_age       5.332163e-05
```

Question 7

```
qda_mod <- discrim_quad() %>%
  set_engine("MASS") %>%
  set_mode("classification")

qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(titanic_recipe)

qda_fit <- fit(qda_wkflow, train)

log_fit %>%
  tidy()
```

```
## # A tibble: 10 × 5
##    term             estimate std.error statistic  p.value
##    <chr>               <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)      3.86      0.613      6.30    2.99e-10
##  2 age             -0.0449    0.0116    -3.88    1.03e- 4
##  3 sib_sp          -0.463     0.123     -3.77    1.66e- 4
##  4 parch            0.000920  0.136      0.00675 9.95e- 1
##  5 fare             0.00785   0.0106     0.741   4.59e- 1
##  6 pclass_X2       -1.14      0.340     -3.34    8.35e- 4
##  7 pclass_X3       -2.24      0.349     -6.41    1.42e-10
##  8 sex_male        -2.51      0.299     -8.40    4.44e-17
##  9 sex_male_x_fare -0.0126    0.00833   -1.51    1.30e- 1
## 10 fare_x_age       0.000146  0.000187   0.781   4.35e- 1
```

Question 8

```
nb_mod <- naive_Bayes() %>%
  set_engine("klaR") %>%
  set_mode("classification") %>%
  set_args(usekernel = FALSE)

nb_wkflow <- workflow() %>%
  add_model(nb_mod) %>%
  add_recipe(titanic_recipe)

nb_fit <- fit(nb_wkflow, train)

nb_fit
```

```
## ══ Workflow [trained] ══════════════════════════════════════════════════════════
## Preprocessor: Recipe
## Model: naive_Bayes()
##
## ── Preprocessor ────────────────────────────────────────────────────────────────
## 3 Recipe Steps
##
## • step_impute_linear()
## • step_dummy()
## • step_interact()
##
## ── Model ───────────────────────────────────────────────────────────────────────
## $apriori
## grouping
##       No      Yes
## 0.616573 0.383427
##
## $tables
## $tables$age
##         [,1]     [,2]
## No  30.74438 13.43270
## Yes 28.73447 13.99961
##
## $tables$sib_sp
##          [,1]      [,2]
## No  0.5375854 1.2561537
## Yes 0.4908425 0.7332119
##
## $tables$parch
##          [,1]      [,2]
## No  0.2984055 0.7431712
## Yes 0.5164835 0.8138864
##
## $tables$fare
##         [,1]     [,2]
## No  22.79547 33.04831
## Yes 47.90606 63.47966
##
## $tables$pclass_X2
##          [,1]      [,2]
## No  0.1936219 0.3955868
## Yes 0.2600733 0.4394800
##
## $tables$pclass_X3
##          [,1]      [,2]
## No  0.6583144 0.4748159
## Yes 0.3443223 0.4760194
##
## $tables$sex_male
##          [,1]      [,2]
## No  0.8473804 0.3600307
## Yes 0.2967033 0.4576436
##
## $tables$sex_male_x_fare
##         [,1]     [,2]
## No  19.46233 33.02823
## Yes 11.54489 37.72080
##
## $tables$fare_x_age
##          [,1]     [,2]
## No   723.4217 1352.736
## Yes 1500.2063 2286.225
##
## ...
## and 1446 more lines.
```

Question 9

```
log_pred <- predict(log_fit, new_data = train,
                      type = "class") %>%
  bind_cols(train %>% select(survived))

log_acc <- log_pred %>%
  accuracy(truth = survived, estimate = .pred_class)

lda_pred <- predict(lda_fit, new_data = train,
                    type = "class") %>%
  bind_cols(train %>% select(survived))

lda_acc <- lda_pred %>%
  accuracy(truth = survived, estimate = .pred_class)

qda_pred <- predict(qda_fit, new_data = train,
                    type = "class") %>%
  bind_cols(train %>% select(survived))

qda_acc <- qda_pred %>%
  accuracy(truth = survived, estimate = .pred_class)

nb_pred <- predict(nb_fit, new_data = train,
                   type = "class") %>%
  bind_cols(train %>% select(survived))

nb_acc <- nb_pred %>%
  accuracy(truth = survived, estimate = .pred_class)

accuracies <- c(log_acc$.estimate, lda_acc$.estimate,
                qda_acc$.estimate, nb_acc$.estimate)
models <- c("Logistic Regression", "LDA", "QDA",
            "Naive Bayes")
results <- tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 4 × 2
##   accuracies models
##        <dbl> <chr>
## 1      0.817 Logistic Regression
## 2      0.802 LDA
## 3      0.792 QDA
## 4      0.782 Naive Bayes
```

log_acc = 0.81 lda_acc = 0.79 qda_acc = 0.80 nb_acc = 0.77

The log fit had the highest accuracy on the training data.

Question 10

```
predict(log_fit, new_data = test, type = "prob")
```

```
## # A tibble: 179 × 2
##    .pred_No .pred_Yes
##       <dbl>     <dbl>
##  1   0.0664    0.934
##  2   0.921     0.0788
##  3   0.921     0.0787
##  4   0.127     0.873
##  5   0.956     0.0444
##  6   0.256     0.744
##  7   0.951     0.0493
##  8   0.775     0.225
##  9   0.433     0.567
## 10   0.432     0.568
## # … with 169 more rows
```

```
augment(log_fit, new_data = test) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

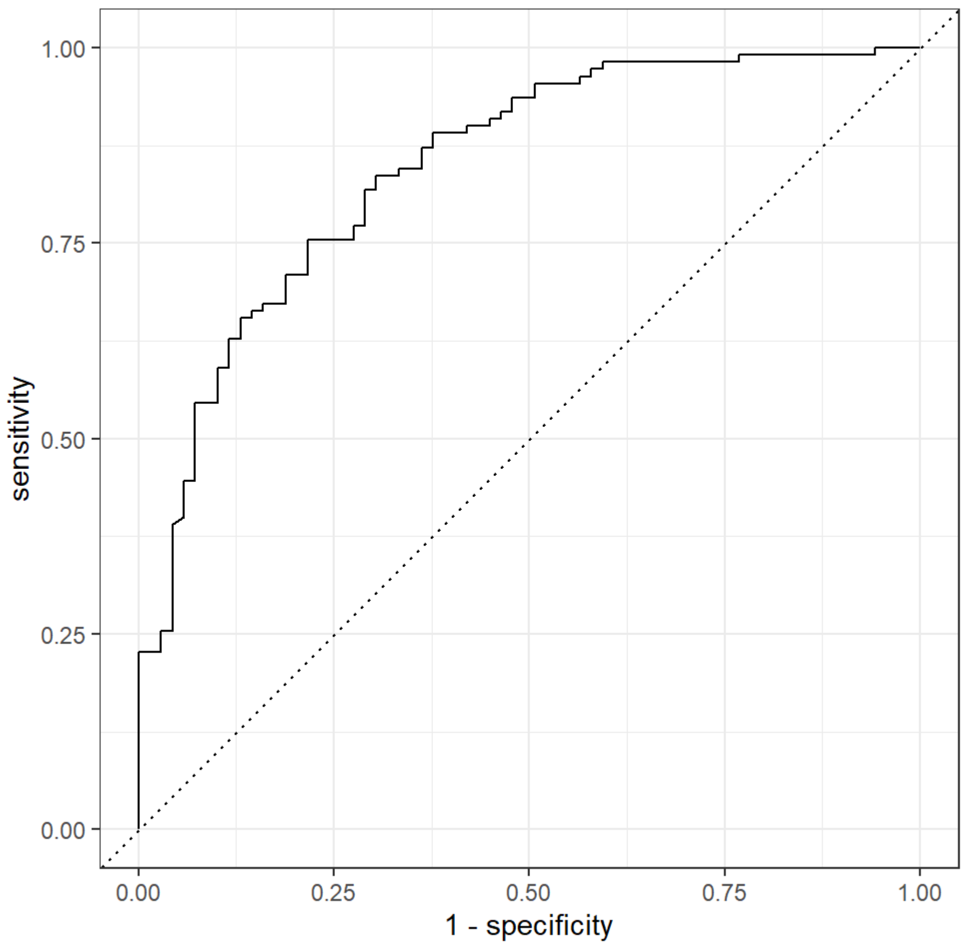
```
##           Truth
## Prediction No Yes
##        No  97  26
##        Yes 13  43
```

```
multi_metric <- metric_set(accuracy, sensitivity, specificity)

augment(log_fit, new_data = test) %>%
  multi_metric(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 3 × 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.782
## 2 sensitivity binary         0.882
## 3 specificity binary         0.623
```

```
augment(log_fit, new_data = test) %>%
  roc_curve(survived, .pred_No) %>%
  autoplot()
```



```
augment(log_fit, new_data = test) %>%
  roc_auc(survived, .pred_No)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.846
```

This model performed well (not super accurate but it is above 50%). The training predicted 81% accuracy, but the testing got 78% accuracy, which is still pretty good. They do differ a bit, but it's a not a big difference and the small difference may be due to the fact that it was different subjects that were tested.