hw3

Yiting Zhang

2022-04-20

```
titanic <- read.csv(file='titanic.csv')
titanic$survived <- factor(titanic$survived, levels=c ('Yes','No'))
titanic<-titanic%>%mutate(titanic$survived)%>%mutate(pclass=factor(titanic$pclass))
#levels(titanic$survived)
#head(titanic)
```

Question 1

```
# Stratified Sampling
set.seed(3435)
titanic_split <- initial_split(titanic, strata = survived, prop = 0.8)
titanic_split
```

```
## <Analysis/Assess/Total>
## <712/179/891>
```

```
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)

data<-dim(titanic)[1]
train<-dim(titanic_train)[1]
test<-dim(titanic_test)[1]
train/data
```

```
## [1] 0.7991021
```

```
test/data
```

```
## [1] 0.2008979
```

We can verify that the training and testing data sets have the appropriate number of observations by calculating the ratio above.

In both the testing and training data, stratified sampling will retain the real ratio of Survived,avoiding sampling error in which one dataset has more observations where Survived is Yes than the other dataset.

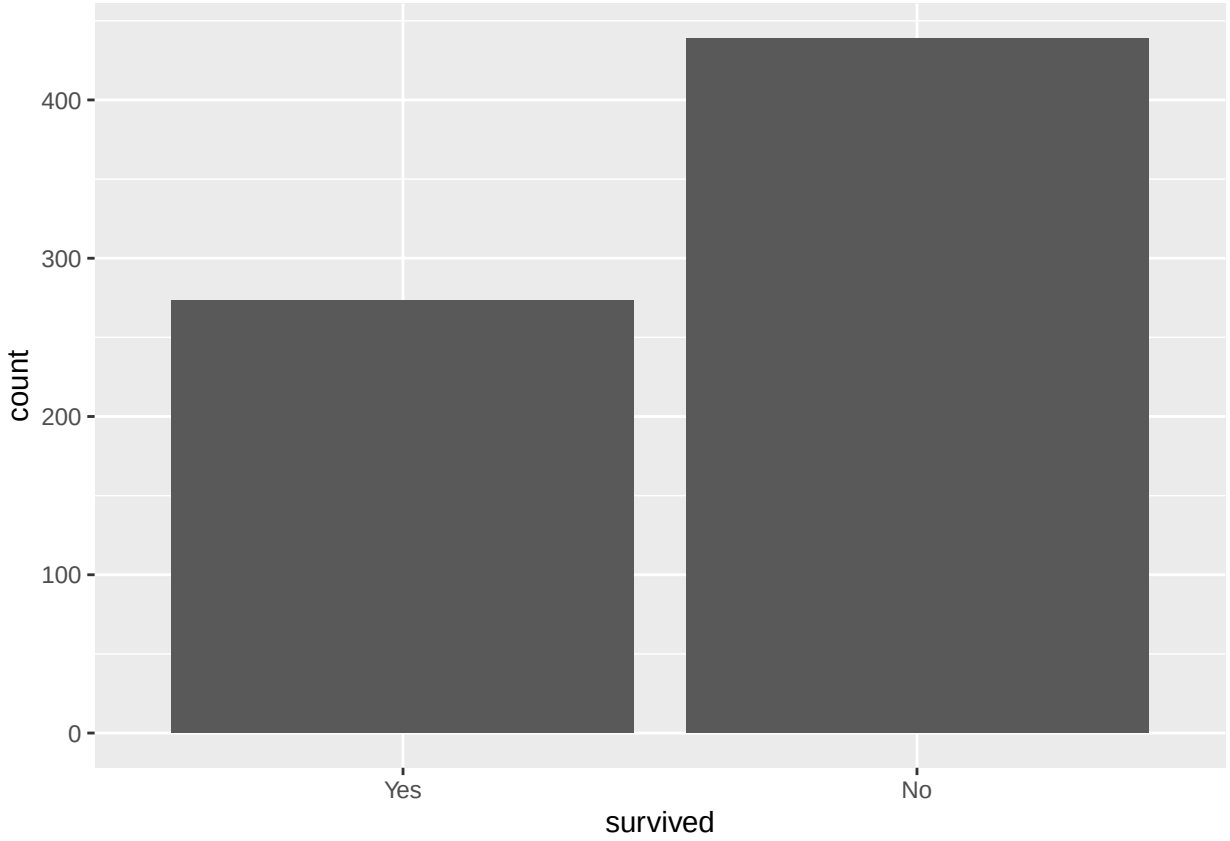
```
sum (is.na(titanic_train))
```

```
## [1] 701
```

There are 701 missing data in our titanic training dataset. This will influence our model. But overall, the stratified sampling method is a good idea. It enables us to collect a representative sample of the full observation under investigation.

Question 2

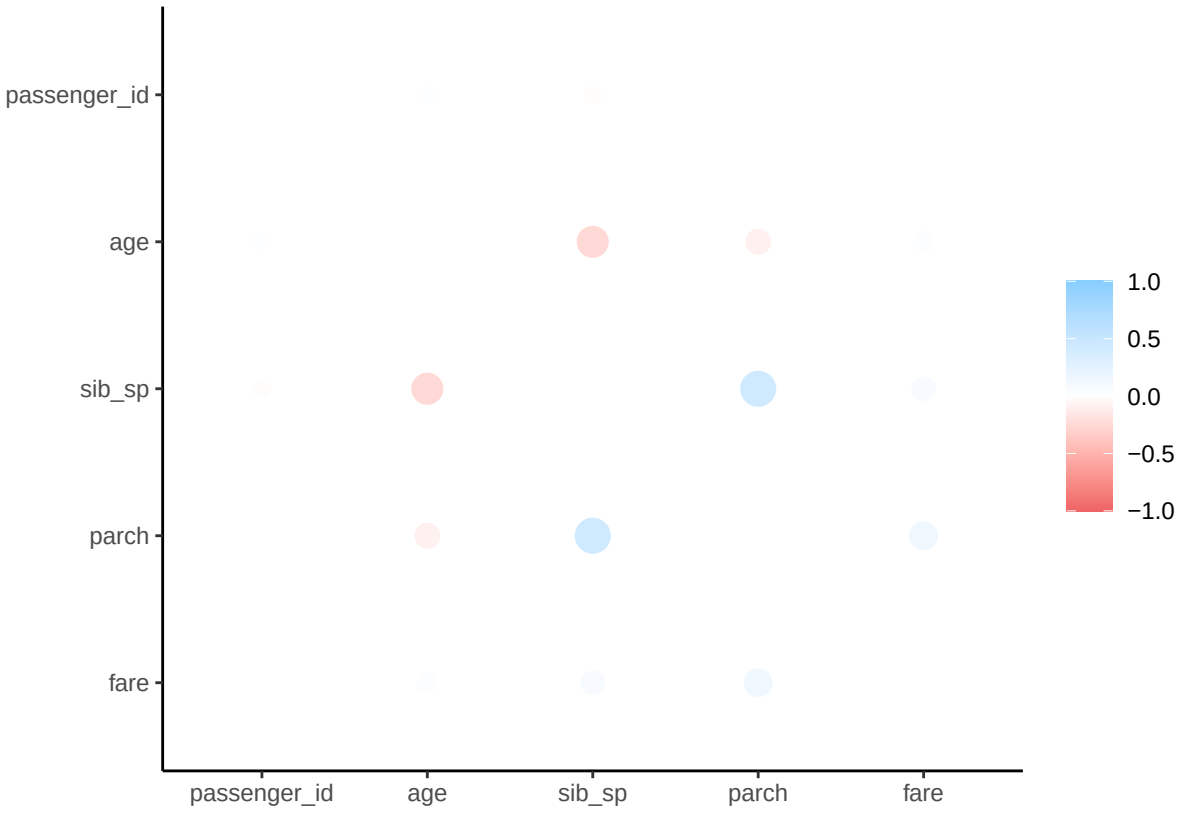
```
# Explore Distribution
titanic_train %>% ggplot(aes(x=survived)) + geom_bar()
```



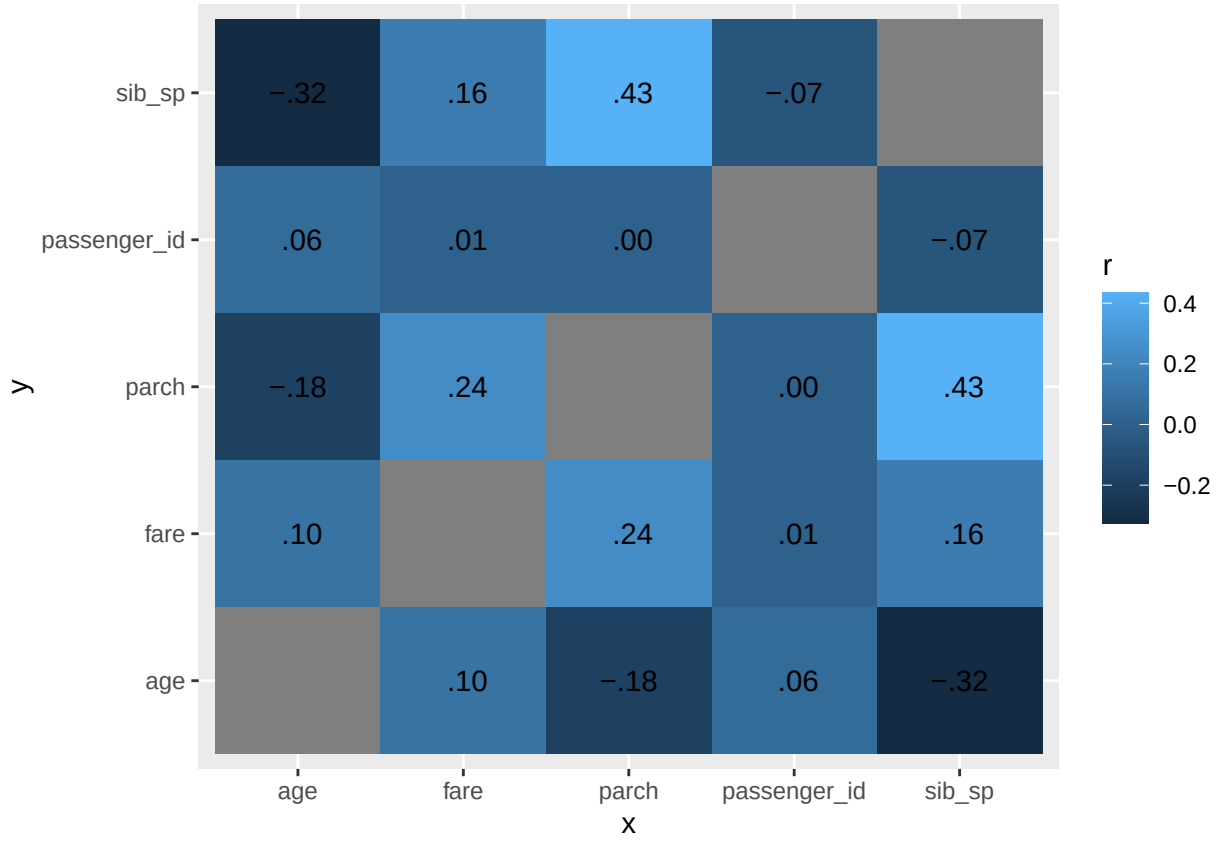
We can observe from the plot that about over 450 the passengers in the training dataset did not survive while only approximately 250 passengers survived.

Question 3

```
# Visualization
cor_titanic <- titanic_train %>%
  select(where(is.numeric)) %>%
  correlate()
rplot(cor_titanic)
```




```
cor_titanic %>%
  stretch() %>%
  ggplot(aes(x, y, fill = r)) +
  geom_tile() +
  geom_text(aes(label = as.character(fashion(r))))
```



Number of parents/children on board and number of siblings/spouses on board are positively correlated. Number of parents/children on board and fare are slightly positively correlated. Age and number of siblings/spouses on board are negatively correlated. Number of parents/children and age are negatively correlated.Passenger ID almost have no correlation with any other predictors.

Question 4

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

Question 5

```
# Specify an Engine
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")
# Workflow
log_wkflow <- workflow() %>%
```

```
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)

log_fit <- fit(log_wkflow, titanic_train)

# Model Results
log_fit %>%
  tidy()
```

```
## # A tibble: 10 x 5
##    term             estimate std.error statistic  p.value
##    <chr>               <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)     -4.34      0.651       -6.66  2.72e-11
##  2 age              0.0606    0.0128       4.75  1.99e- 6
##  3 sib_sp           0.436     0.129        3.37  7.57e- 4
##  4 parch            0.280     0.151        1.85  6.40e- 2
##  5 fare            -0.00639   0.0109      -0.587 5.57e- 1
##  6 pclass_X2        1.16      0.343        3.39  6.92e- 4
##  7 pclass_X3        2.33      0.361        6.45  1.15e-10
##  8 sex_male         2.37      0.297        8.00  1.29e-15
##  9 sex_male_x_fare  0.0136    0.00859      1.59  1.13e- 1
## 10 age_x_fare      -0.000281  0.000190    -1.48  1.39e- 1
```

Question 6

```
# LDA
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, titanic_train)
```

Question 7

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

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

qda_fit <- fit(qda_wkflow, titanic_train)
```

Question 8

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

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

nb_fit <- fit(nb_wkflow, titanic_train)
```

Question 9

```
log_predict<-predict(log_fit, new_data = titanic_train, type = "prob")

log_reg_acc <- augment(log_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
log_reg_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.816
```

```
lda_predict<-predict(lda_fit, new_data = titanic_train, type = "prob")

lda_acc <- augment(lda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
lda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.806
```

```
qda_predict<-predict(qda_fit, new_data = titanic_train, type = "prob")

qda_acc <- augment(qda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
qda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.787
```

```
nb_predict<-predict(nb_fit, new_data = titanic_train, type = "prob")

nb_acc <- augment(nb_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
nb_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.778
```

```
pred_df <- bind_cols(log_predict, lda_predict, qda_predict, nb_predict, titanic_train$survived)
names <- c("Logistic Regression", "Linear Discriminant Analysis", "Quadratic Discriminant Analysis", "Na
colnames(pred_df)%>%names
```

```
## NULL
```

```
pred_df
```

```
## # A tibble: 712 x 9
##    .pred_Yes...1 .pred_No...2 .pred_Yes...3 .pred_No...4 .pred_Yes...5
##            <dbl>        <dbl>         <dbl>        <dbl>         <dbl>
##  1        0.106         0.894       0.0636         0.936  0.00581
##  2        0.0786        0.921       0.0453         0.955  0.00442
##  3        0.290         0.710       0.237          0.763  0.0608
##  4        0.0990        0.901       0.0679         0.932  0.0000344
##  5        0.0116        0.988       0.00692        0.993  0.0000000163
##  6        0.776         0.224       0.839          0.161  0.599
##  7        0.0631        0.937       0.0473         0.953  0.000000119
##  8        0.492         0.508       0.593          0.407  0.265
##  9        0.222         0.778       0.156          0.844  0.00958
## 10        0.530         0.470       0.645          0.355  0.000674
## # ... with 702 more rows, and 4 more variables: .pred_No...6 <dbl>,
## #   .pred_Yes...7 <dbl>, .pred_No...8 <dbl>, ...9 <fct>
```

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

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.816 Logistic Regression
## 2      0.806 LDA
## 3      0.787 QDA
## 4      0.778 Naive Bayes
```

Therefore, by comparing the four models, we find that logistic regression model got the highest accuracy on the training data.

Question 10

```
# Fitting to Testing Data
predict(log_fit, new_data = titanic_test, type = "prob")
```

```
## # A tibble: 179 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1    0.110    0.890
##  2    0.493    0.507
##  3    0.806    0.194
##  4    0.170    0.830
##  5    0.169    0.831
##  6    0.0440   0.956
##  7    0.162    0.838
##  8    0.563    0.437
##  9    0.909    0.0913
## 10    0.722    0.278
## # ... with 169 more rows
```

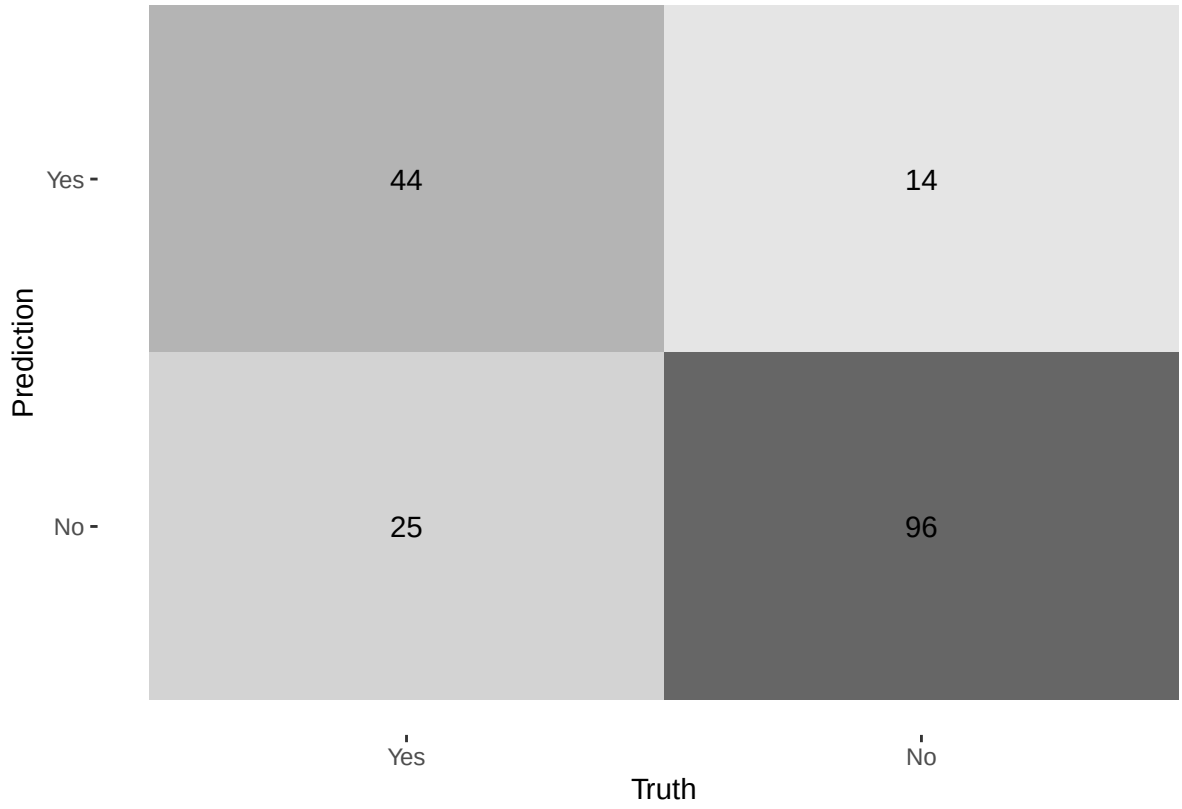
```
# Check testing accuracy
multi_metric <- metric_set(accuracy, sensitivity, specificity)

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

```
## # A tibble: 3 x 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.816
## 2 sensitivity binary         0.692
## 3 specificity binary         0.893
```

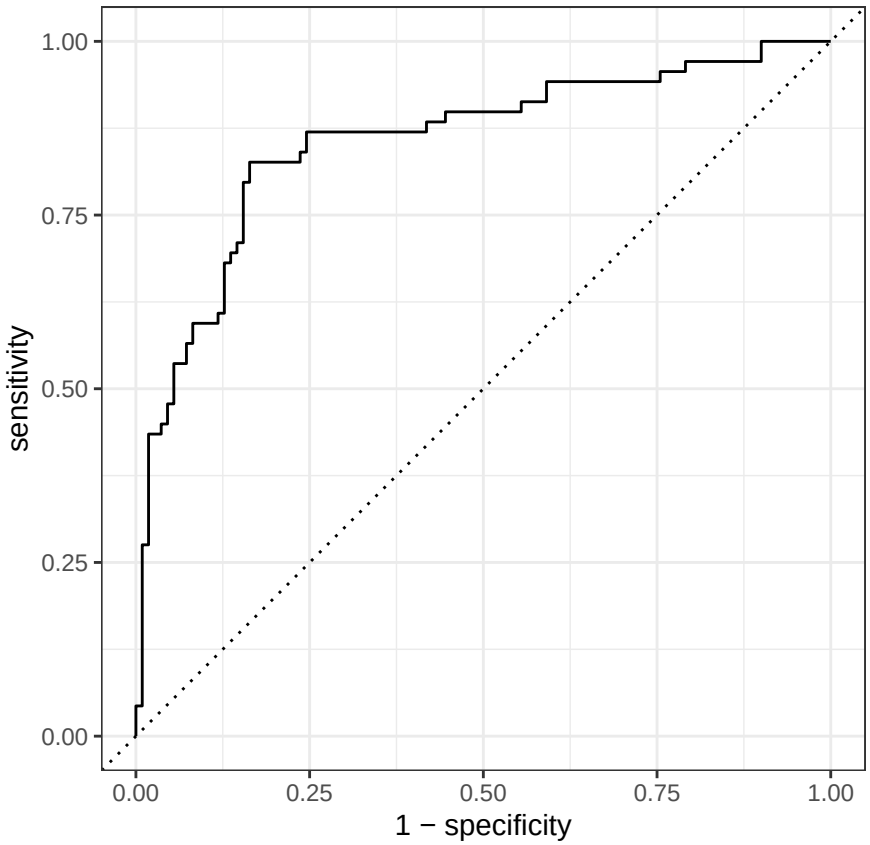
```
augment(log_fit, new_data = titanic_test) %>%
  multi_metric(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 3 x 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.782
## 2 sensitivity binary         0.638
## 3 specificity binary         0.873
```

```
# View the confusion matrix on the testing data
augment(log_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class) %>%
  autoplot(type = "heatmap")
```




```
# ROC curve on the testing data
augment(log_fit, new_data = titanic_test) %>%
  roc_curve(survived, .pred_Yes) %>%
  autoplot()
```




```
# Area under the ROC curve (AUC)
augment(log_fit, new_data = titanic_test) %>%
  roc_auc(survived, .pred_Yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.856
```

The training accuracy is 0.816 and the testing accuracy is 0.782. The high accuracy indicates that the model perform well in this case. And the higher accuracy in training dataset is reasonable because it the model is fitted better on the traning dataset. And the AUC score looks good. Thus the model is usable.