Hw2

Peter Chu

2022-10-31

Question 1

```
data <- read_csv('titanic.csv')
```

```
## Rows: 891 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): survived, name, sex, ticket, cabin, embarked
## dbl (6): passenger_id, pclass, age, sib_sp, parch, fare
##
## 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.
```

```
data$survived <- factor(data$survived, ordered = TRUE)
data$pclass <- factor(data$pclass)

set.seed(100)

data_split <- initial_split(data, strata = survived, prop = 0.7)
data_train <- training(data_split)
data_test <- testing(data_split)

data_split
```

```
## <Training/Testing/Total>
## <623/268/891>
```

```
dim(data_train)
```

```
## [1] 623  12
```

```
dim(data_test)
```

```
## [1] 268  12
```

```
data_train
```

```
## # A tibble: 623 x 12
##    passenger_id survived pclass name       sex     age sib_sp parch ticket  fare
##           <dbl> <ord>    <fct>  <chr>      <chr> <dbl>  <dbl> <dbl> <chr>  <dbl>
##  1            1 No       3      Braund, M~ male     22      1     0 A/5 2~  7.25
##  2            5 No       3      Allen, Mr~ male     35      0     0 373450  8.05
##  3            6 No       3      Moran, Mr~ male     NA      0     0 330877  8.46
##  4            7 No       1      McCarthy,~ male     54      0     0 17463  51.9
##  5           13 No       3      Saunderco~ male     20      0     0 A/5. ~  8.05
##  6           14 No       3      Andersson~ male     39      1     5 347082 31.3
##  7           17 No       3      Rice, Mas~ male      2      4     1 382652 29.1
##  8           21 No       2      Fynney, M~ male     35      0     0 239865 26
##  9           25 No       3      Palsson, ~ fema~     8      3     1 349909 21.1
## 10           27 No       3      Emir, Mr.~ male     NA      0     0 2631    7.22
## # ... with 613 more rows, and 2 more variables: cabin <chr>, embarked <chr>
```

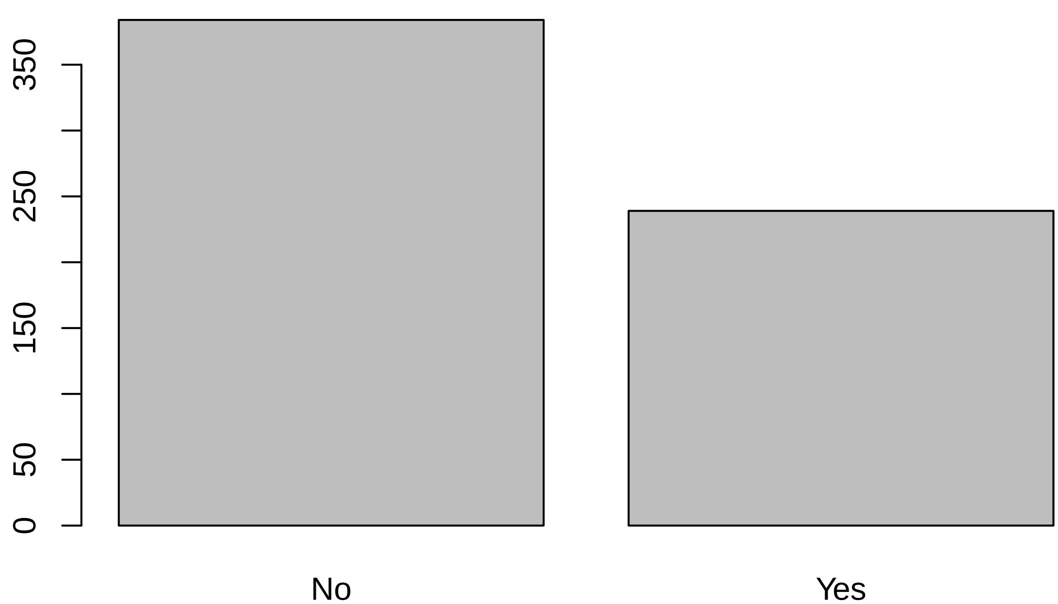
```
#number of cols and rows match
```

The training and testing data sets have the appropriate number of observations. The issues with the training data is that there are a lot of missing values. Furthermore, many of the observations have missing data in areas where others have them, but then have missing data in other areas.

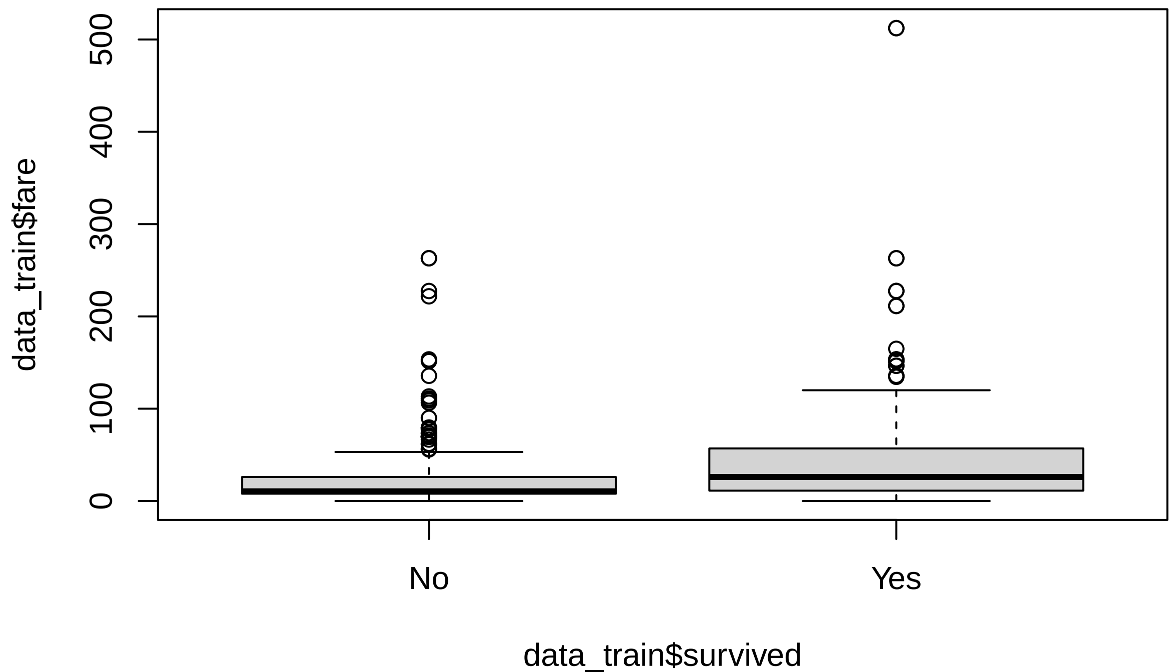
Stratified sampling is a good idea for this data as it allows us to capture the huge number of observations with a single sample that best represents the entire population.

Question 2

```
plot(data_train$survived)
```



```r
plot(data_train$fare~data_train$survived)
```



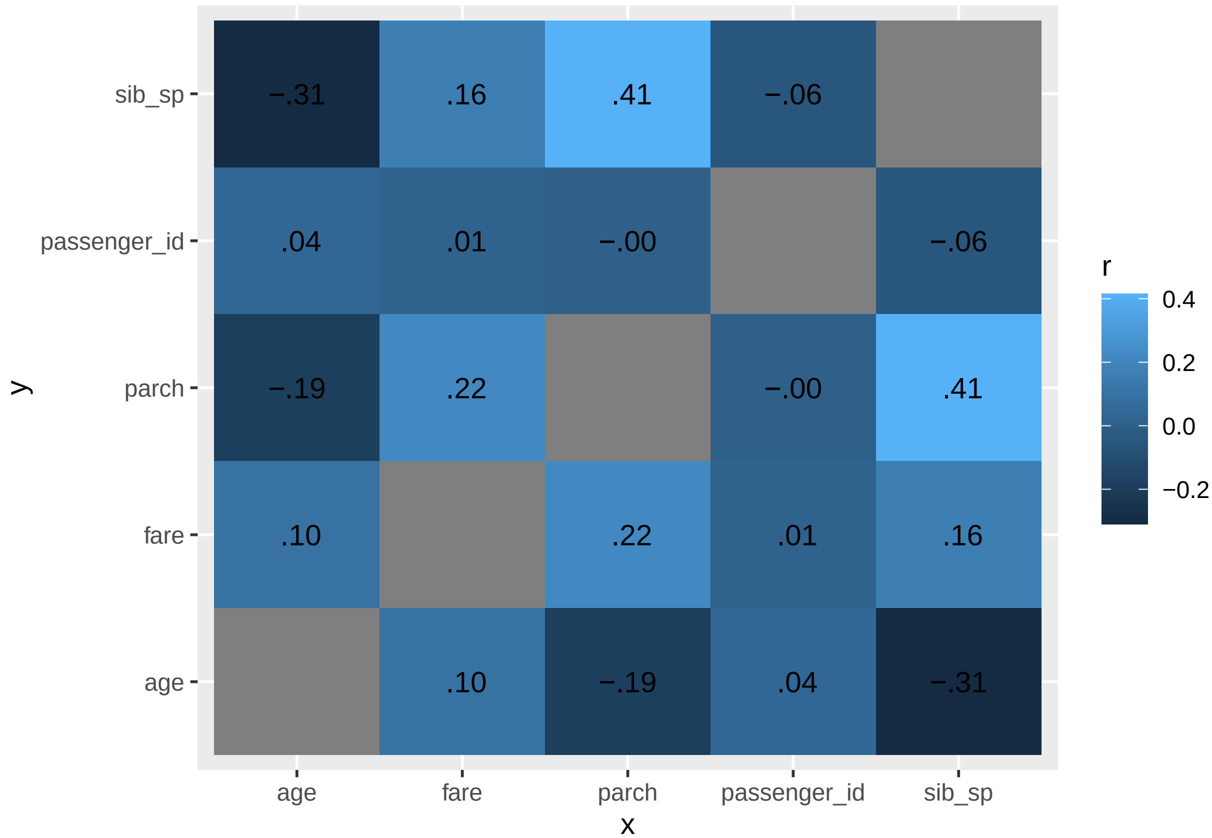
On average more people did not survive. The boxplot also shows that on average, those that had a higher fare ended up surviving. This could lead a lot of conclusions, but I don't think we can claim any of them as certain.

Question 3

```r
cor_data <- data %>%
  select(-survived) %>%
  correlate()
```

```
## Non-numeric variables removed from input: `pclass`, `name`, `sex`, `ticket`, `cabin`, and `embarked`
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
```

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



A lot of the variables are weakly correlated, but some are decently correlated. For example, pclass and age have a correlation of 0.37 is the negative direction. Similaryly, sib_sp and parch have a correlation of 0.41 in the positive direction. pclass and fare have the highest correlation at -0.55.

Question 4

```
data_train_recipe <- recipe(survived~pclass + sex + age + sib_sp + parch + fare, data = data_train) %>%
  step_impute_linear('age') %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~ starts_with('sex'):fare) %>%
  step_interact(terms = ~ starts_with('age'):fare)

data_train_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
## Interactions with starts_with("age"):fare
```

Question 5

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

log_wflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(data_train_recipe)

log_fit <- fit(log_wflow, data_train)

log_fit %>%
  tidy()
```

```
## # A tibble: 10 x 5
##    term              estimate std.error statistic  p.value
##    <chr>                <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)      3.92       0.672        5.82  5.73e- 9
##  2 age             -0.0511     0.0127      -4.03  5.56e- 5
##  3 sib_sp          -0.493      0.129       -3.81  1.38e- 4
##  4 parch           -0.0741     0.155       -0.478 6.33e- 1
##  5 fare             0.0116     0.0115       1.00  3.15e- 1
##  6 pclass_X2       -1.07       0.376       -2.84  4.52e- 3
##  7 pclass_X3       -2.27       0.388       -5.84  5.27e- 9
##  8 sex_male        -2.23       0.301       -7.40  1.32e-13
##  9 sex_male_x_fare -0.0121     0.00836     -1.45  1.47e- 1
## 10 age_x_fare       0.0000599  0.000211     0.284 7.76e- 1
```

Question 6

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

lda_wflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(data_train_recipe)

lda_fit <- fit(lda_wflow, data_train)

lda_fit
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: discrim_linear()
##
## -- Preprocessor ----------------------------------------------------------------
## 4 Recipe Steps
```

```
##
## * step_impute_linear()
## * step_dummy()
## * step_interact()
## * step_interact()
##
## -- Model -----------------------------------------------------------------------
## Call:
## lda(..y ~ ., data = data)
##
## Prior probabilities of groups:
##        No       Yes
## 0.6163724 0.3836276
##
## Group means:
##          age    sib_sp     parch     fare pclass_X2 pclass_X3  sex_male
## No  29.99633 0.5781250 0.3229167 22.10414 0.1640625 0.6875000 0.8385417
## Yes 28.15097 0.5020921 0.4476987 47.05976 0.2468619 0.3640167 0.3263598
##     sex_male_x_fare age_x_fare
## No         18.53645   702.8996
## Yes        12.58586  1475.9813
##
## Coefficients of linear discriminants:
##                           LD1
## age             -3.127696e-02
## sib_sp          -2.771216e-01
## parch           -3.348311e-02
## fare             1.859579e-03
## pclass_X2       -7.367099e-01
## pclass_X3       -1.578941e+00
## sex_male        -1.978542e+00
## sex_male_x_fare -8.444224e-04
## age_x_fare       1.071036e-05
```

Question 7

```r
qda_model <- discrim_quad() %>%
  set_mode('classification') %>%
  set_engine('MASS')

qda_wflow <- workflow() %>%
  add_model(qda_model) %>%
  add_recipe(data_train_recipe)

qda_fit <- fit(qda_wflow, data_train)

qda_fit
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: discrim_quad()
##
## -- Preprocessor ----------------------------------------------------------------
```

```
## 4 Recipe Steps
##
## * step_impute_linear()
## * step_dummy()
## * step_interact()
## * step_interact()
##
## -- Model -----------------------------------------------------------------------
## Call:
## qda(..y ~ ., data = data)
##
## Prior probabilities of groups:
##        No       Yes
## 0.6163724 0.3836276
##
## Group means:
##          age    sib_sp     parch     fare pclass_X2 pclass_X3  sex_male
## No  29.99633 0.5781250 0.3229167 22.10414 0.1640625 0.6875000 0.8385417
## Yes 28.15097 0.5020921 0.4476987 47.05976 0.2468619 0.3640167 0.3263598
##     sex_male_x_fare age_x_fare
## No         18.53645   702.8996
## Yes        12.58586  1475.9813
```

Question 8

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

nb_wflow <- workflow() %>%
  add_model(nb_mod) %>%
  add_recipe(data_train_recipe)

nb_fit <- fit(nb_wflow, data_train)
```

Question 9

```
options(pillar.sigfig = 1)
pred1 <- predict(log_fit, new_data = data_train, type = 'prob')
pred2 <- predict(lda_fit, new_data = data_train, type = 'prob')
pred3 <- predict(qda_fit, new_data = data_train, type = 'prob')
pred4 <- predict(nb_fit, new_data = data_train, type = 'prob')
full_data_pred <- bind_cols(pred1, pred2, pred3, pred4, data_train %>% select(survived))
```

```
## New names:
## * `.pred_No` -> `.pred_No...1`
## * `.pred_Yes` -> `.pred_Yes...2`
## * `.pred_No` -> `.pred_No...3`
## * `.pred_Yes` -> `.pred_Yes...4`
## * `.pred_No` -> `.pred_No...5`
## * `.pred_Yes` -> `.pred_Yes...6`
## * `.pred_No` -> `.pred_No...7`
## * `.pred_Yes` -> `.pred_Yes...8`
```

```r
full_data_pred
```

```
## # A tibble: 623 x 9
##    .pred_No...1 .pred_Yes...2 .pred_No...3 .pred_Yes...4 .pred_No...5
##           <dbl>         <dbl>        <dbl>         <dbl>        <dbl>
##  1          0.9          0.1           0.9          0.07           1.
##  2          0.9          0.09          0.9          0.05           1.
##  3          0.9          0.1           0.9          0.07           1.
##  4          0.7          0.3           0.8          0.2            1.
##  5          0.8          0.2           0.9          0.1            1.
##  6          1.           0.03          1.           0.02           1.
##  7          0.9          0.06          1.           0.05           1.
##  8          0.8          0.2           0.8          0.2            1.
##  9          0.5          0.5           0.4          0.6            1.
## 10          0.9          0.1           0.9          0.07           1.
## # ... with 613 more rows, and 4 more variables: .pred_Yes...6 <dbl>,
## #   .pred_No...7 <dbl>, .pred_Yes...8 <dbl>, survived <ord>
```

```r
log_acc <- augment(log_fit, new_data = data_train) %>%
  accuracy(truth = as.factor(data_train$survived), estimate = .pred_class)

lda_acc <- augment(lda_fit, new_data = data_train) %>%
  accuracy(truth = as.factor(data_train$survived), estimate = .pred_class)

qda_acc <- augment(qda_fit, new_data = data_train) %>%
  accuracy(truth = as.factor(data_train$survived), estimate = .pred_class)

nb_acc <- augment(nb_fit, new_data = data_train) %>%
  accuracy(truth = as.factor(data_train$survived), estimate = .pred_class)

accuracies <- c(log_acc$.estimate, lda_acc$.estimate, qda_acc$.estimate, nb_acc$.estimate)

models <- c("Logisitc Regression", "LDA", "Naive Bayes", "QDA")

results <- tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1        0.8 Logisitc Regression
## 2        0.8 LDA
## 3        0.8 Naive Bayes
## 4        0.8 QDA
```

The logistic regression had the highest accuracy on the training data

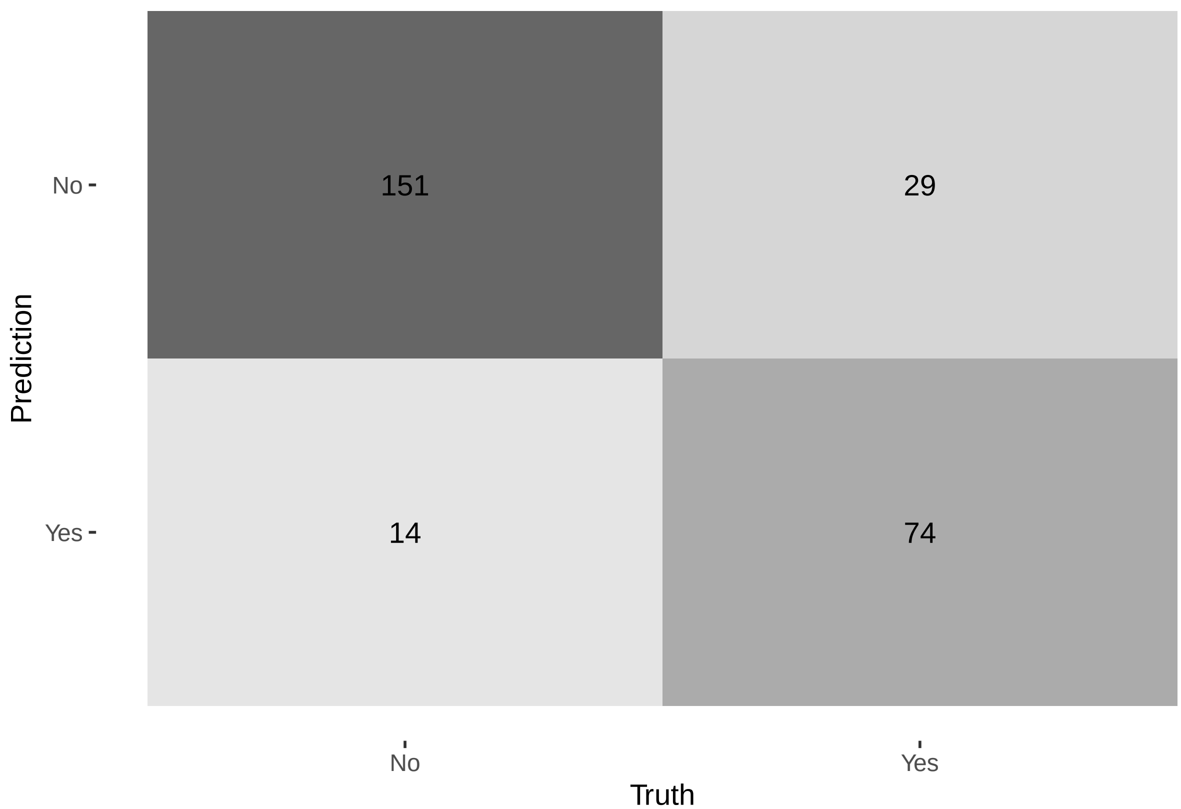
Question 10

```r
prediction <- predict(log_fit, new_data = data_test, type = 'prob')
```

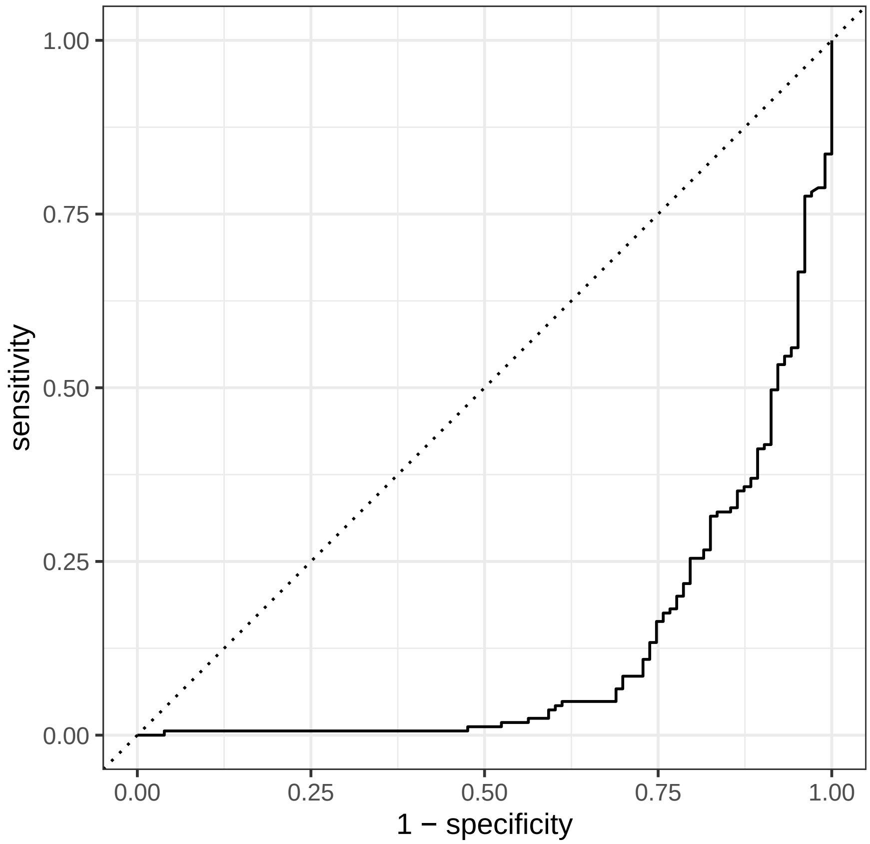
```
accuracy_mod <- augment(log_fit, new_data = data_test) %>%
  accuracy(truth = as.factor(survived), estimate = .pred_class)

accuracy_mod$.estimate
```

```
## [1] 0.8395522
```

```
augment(log_fit, new_data = data_test) %>%
  conf_mat(truth = survived, estimate = .pred_class) %>%
  autoplot(type = 'heatmap')
```




```
augment(log_fit, new_data = data_test) %>%
  roc_curve(survived, .pred_Yes) %>%
  autoplot()
```




```
roc(data_test$survived,predictor =(factor(prediction$.pred_Yes, ordered = TRUE)))
```

```
## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

##
## Call:
## roc.default(response = data_test$survived, predictor = (factor(prediction$.pred_Yes,     ordered = T
##
## Data: (factor(prediction$.pred_Yes, ordered = TRUE)) in 165 controls (data_test$survived No) < 103 ca
## Area under the curve: 0.8796
```

The model performed well. It even performed better on the test data than the training data. This may be caused by our random sampling, but overall the accuracy is similar and higher than the other forms of regression we used. The AUC is 0.879.

Question 11

We have $p(z) = ln(\frac{e^z}{1-e^z}) \rightarrow p(1+e^z) = e^z \rightarrow p * 1 + p * e^z = e^z \rightarrow p = e^z - pe^z \rightarrow e^z(1-p) = p \rightarrow e^z = \frac{p}{1-p} \rightarrow z(p) = log_e(\frac{p}{1-p}) \rightarrow z(p) = ln(\frac{p}{1-p})$

Question 12

Increasing x_1 by 2 units would change the odds of the outcome by $e^{2\beta_1}$ We have $\frac{Pr(Y=1|x)}{1-Pr(Y=1|x)} = e^{\beta_0+\beta_1 x}$ So increasing x by 2 would lead to $\frac{Pr(Y=1|x)}{1-Pr(Y=1|x)} = e^{\beta_0+\beta_1(x+2)} = e^{\beta_0} * e^{\beta_1 x} * e^{2\beta_1} = e^{\beta_0+\beta_1 x} * e^{2\beta_1}$ which shows that an increase in x by 2 would lead to a factor of $e^{2\beta_1}$

If we assume that β_1 is now negative, then as $x_1 \to \infty$ we have $-\beta_1 * \infty = -\infty$ so $p \to -\infty$. If $x_1 \to -\infty$ then we have $-\beta_1 * -\infty = \infty$ so $p \to \infty$