hwk 3 - 131

cristian razo

4/19/2022

Question 1

Question 1

```
dt=read.csv("Desktop/homework-3/data/titanic.csv")
set.seed(42)
split <- initial_split(dt, prop = 0.80, strata = survived)
train <- training(split)
test <- testing(split)
```

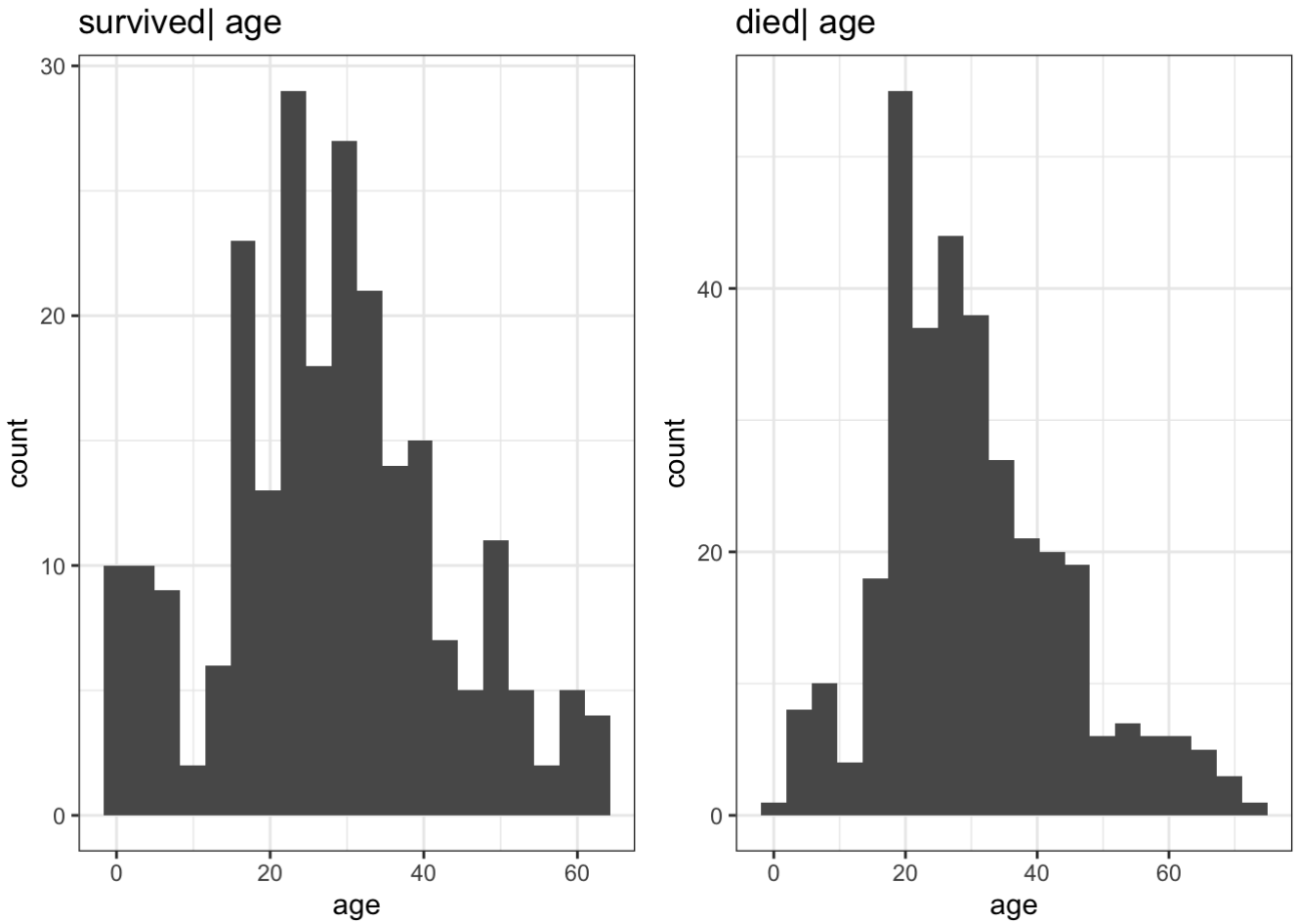
I loaded the titanic data set and split it into two sets . The two sets will be our traning and testing data set .

Question 2 Exploratory analysis

```
surv=train[train[2]=='Yes',]

died=train[train[2]=='No',]

v1=surv %>%
  ggplot(aes(x=age))+
  geom_histogram(bins = 20)+
  ggtitle('survived| age')+
  theme_bw()

v2=died %>%
  ggplot(aes(x = age)) +
  geom_histogram(bins = 20) +
  ggtitle('died| age')+
  theme_bw()

grid.arrange(v1, v2, nrow = 1)
```

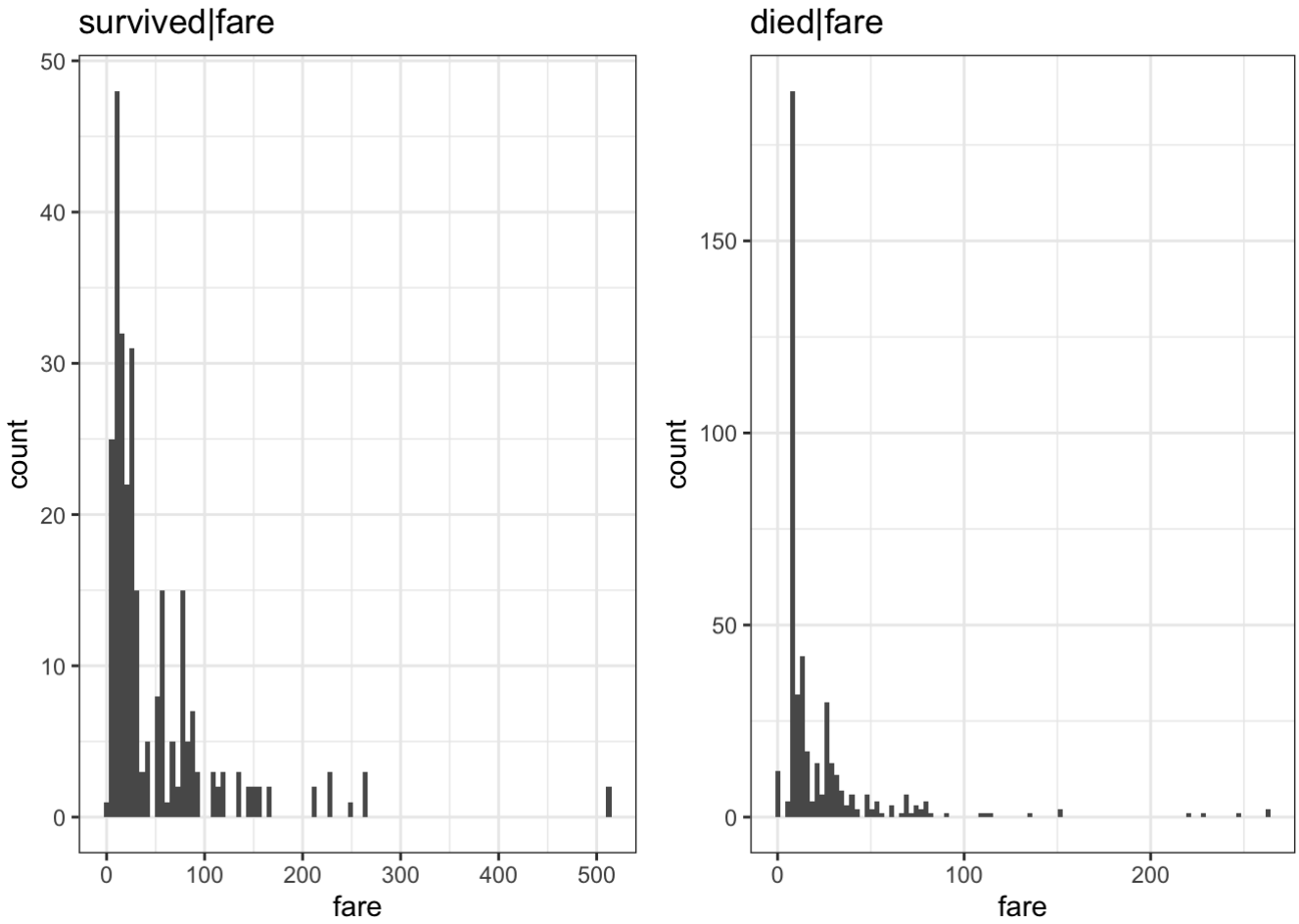
```
## Warning: Removed 37 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 103 rows containing non-finite values (stat_bin).
```



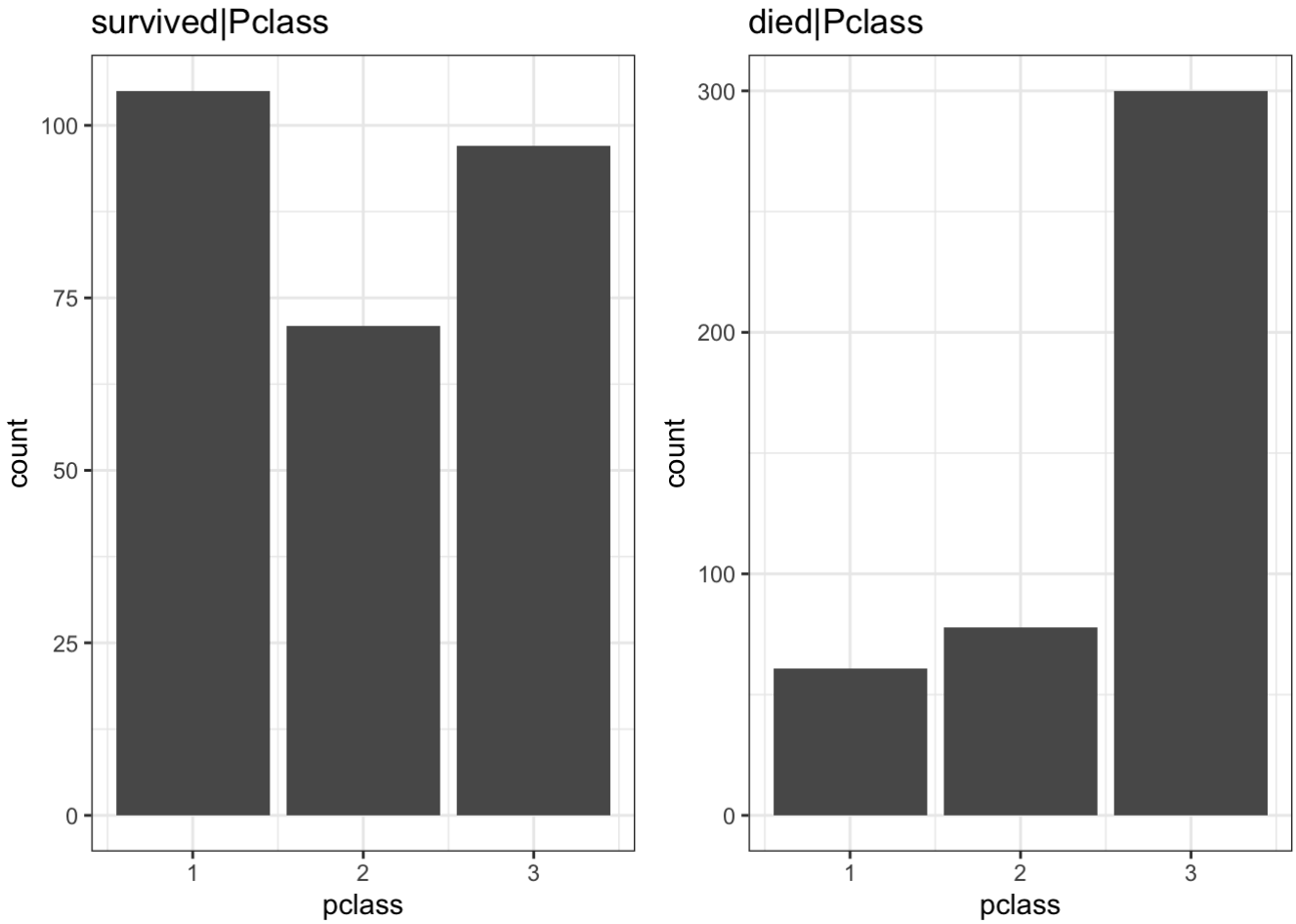
We can see from the visual that the survived distribution is centered and scaled around the younger people .This gives us insight that most of the people that survived were young adults and children .Vice versa occured when we look at the distribution for the dead rows . The visualization informs us that the distribution is centered and scaled around older people. The dead distribution also has more range for age then it did for the survived data set. The dead distribution approximates a normal distribution while the survived distribution is more approximate to a positive skewed t distribution.

```
v3=surv %>%
  ggplot(aes(x=fare))+
  geom_histogram(bins = 100)+
  ggtitle('survived|fare')+
  theme_bw()

v4=died %>%
  ggplot(aes(x = fare)) +
  geom_histogram(bins = 100) +
  ggtitle('died|fare')+
  theme_bw()

grid.arrange(v3, v4, nrow = 1)
```



It seems that there is much variance between fare values for both data sets .A relation I can see is that people with low fare tickets were most likely to die and people with high fare tickets are most likely to survive.This is very informative and is area we should be interested in .

```
v5=surv %>%
  ggplot(aes( x=pclass)) +
  geom_bar() +
  ggtitle('survived|Pclass')+
  theme_bw()

v6=died %>%
  ggplot(aes( x=pclass)) +
  geom_bar() +
  ggtitle('died|Pclass')+
  theme_bw()

grid.arrange(v5, v6, nrow = 1)
```



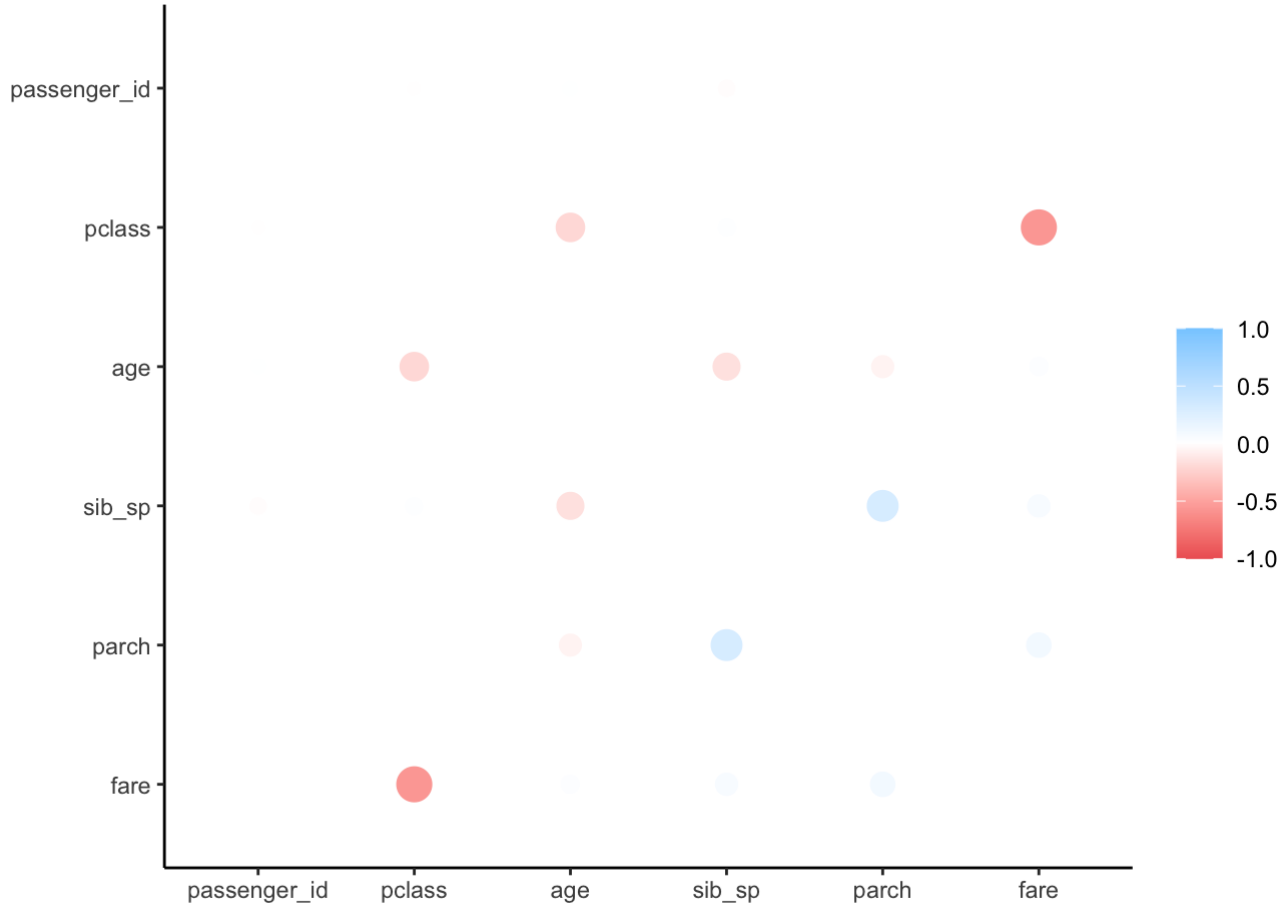
The survived bar plot is more well rounded while the died bar plot suggest that most death came from the 3 pclass.This will suggest that Pclass will play as an important predictor for our classification model.

Question 3

```
train_nu=select_if(train,is.numeric)
rplot(correlate(train_nu))
```

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
## Don't know how to automatically pick scale for object of type noquote. Defaulting to
continuous.
```



The matrix visualization helps us see that we found 5 significant relationship between our features in the titanic dataset. There are 4 negative significant correlations(age~pclass , age~sib_sp , age~parch , Pclass~fare) and 1 positive correlations (Sib_sp~parch) .Age is an important predictor because three other predictors are dependent on age.

Question 4

```
sub_train=train[c(2,3,5,6,7,8,10)]
view(sub_train)
colSums(is.na(sub_train))
```

```
## survived   pclass      sex      age   sib_sp    parch     fare
##        0        0        0      140        0        0        0
```

```
titanic_recipe=recipe(survived ~ . ,data=sub_train) %>%
  step_impute_linear(age,impute_with = imp_vars(all_predictors()))%>%
  prep(sub_train)%>%
  step_dummy(all_nominal_predictors())
```

Created a recipe that will predict the outcome survived. It will include 6 predictors (ticket class, sex,age,number of siblings , number of parents or children aboard , and passanger fare). Step impute linear deals with our NA values in our data set . I also did step dummy that will factor and make all nominal variables into indicator variables.

Question 5

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

log_wrkflw=workflow() %>%
  add_model(log_reg) %>%
  add_recipe(titanic_recipe)

log_fit=fit(log_wrkflw,sub_train)
```

A classification logistic model with the engine set as GLM

Question 6

```
lda_mod = discrim_linear() %>%
                      set_mode('classification')%>%
                                  set_engine("MASS")
lda_wrkflw=workflow() %>%
                      add_model(lda_mod) %>%
                                    add_recipe(titanic_recipe)
lda_fit=fit(lda_wrkflw,sub_train)
```

A classification LDA model with the engine set as MASS

Question 7

```
qda_model=discrim_quad()%>%
                          set_engine("MASS") %>%
                                      set_mode("classification")
qda_wrkflw=workflow() %>%
                          add_model(qda_model) %>%
                                                add_recipe(titanic_recipe)
qda_fit=fit(qda_wrkflw,sub_train)
```

A classification QDA with the engine set as MASS

Question 8

```
nb=naive_Bayes()%>%
                set_engine("klaR") %>%
                          set_mode("classification")%>%
                                    set_args(usekernel=FALSE)


nb_wrkflw=workflow() %>%
  add_model(nb) %>%
  add_recipe(titanic_recipe)


nb_fit=fit(nb_wrkflw,sub_train)
```

A classification Naive bayes model with the engine set as klaR

Question 9

```
log_rec_acc= augment(log_fit,new_data = sub_train)%>%
accuracy(truth=factor(survived),estimate=.pred_class)


lda_rec_acc= augment(lda_fit,new_data = sub_train)%>%
  accuracy(truth=factor(survived), estimate=.pred_class)


qda_rec_acc= augment(qda_fit,new_data = sub_train)%>%
  accuracy(truth=factor(survived), estimate=.pred_class)


nb_rec_acc= augment(nb_fit,new_data = sub_train)%>%
  accuracy(truth=factor(survived), estimate=.pred_class)

accuracies=c(log_rec_acc$.estimate,lda_rec_acc$.estimate,qda_rec_acc$.estimate,nb_rec_ac
c$.estimate)

models =c("Logistic Regression",'LDA','QDA','Naive Bayes')

results=tibble(accuracies=accuracies,model=models)

results %>%
  arrange(-accuracies)
```

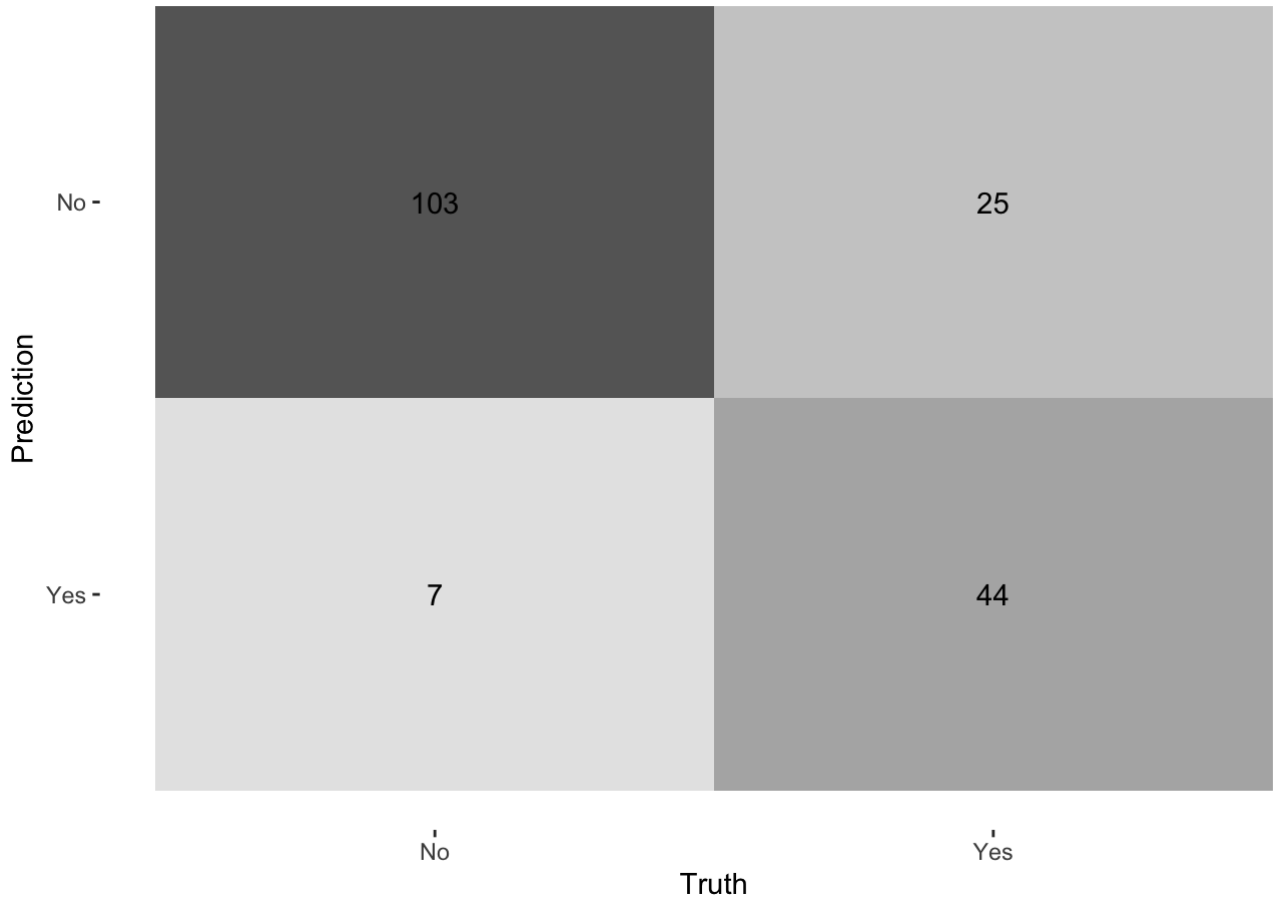
```
## # A tibble: 4 × 2
##   accuracies model
##        <dbl> <chr>
## 1      0.808 Logistic Regression
## 2      0.801 QDA
## 3      0.799 LDA
## 4      0.781 Naive Bayes
```

We can see that the logistic regression model is the model with the best/highest accuracy score .

10 Fitting onto testing set

```
sub_test=test[c(2,3,5,6,7,8,10)]

multi_met=metric_set(accuracy,sensitivity,specificity)

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

```
## Warning in vec2table(truth = truth, estimate = estimate, dnn = dnn, ...):
## `truth` was converted to a factor
```



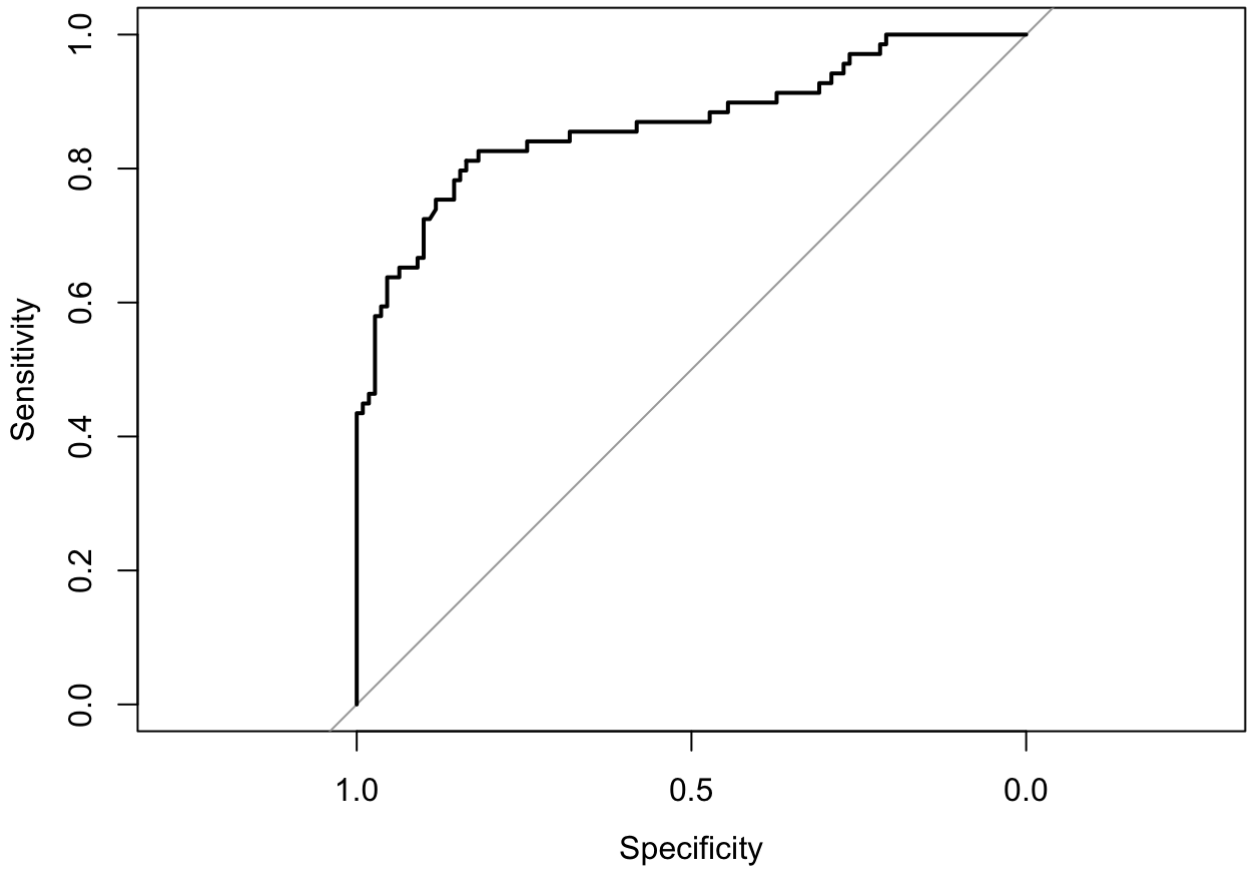

```
log_rec_acc2= augment(log_fit,new_data = sub_test)%>%
                                          multi_met(truth=factor(survived), estimate=.pr
ed_class)
log_rec_acc2
```

```
## # A tibble: 3 × 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.821
## 2 sensitivity binary         0.936
## 3 specificity binary         0.638
```

```
newdt=augment(log_fit,new_data = sub_test)
roc(factor(newdt$survived),newdt$.pred_No,plot=TRUE)
```

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

```
## Setting direction: controls > cases
```




```
##
## Call:
## roc.default(response = factor(newdt$survived), predictor = newdt$.pred_No,     plot =
TRUE)
##
## Data: newdt$.pred_No in 110 controls (factor(newdt$survived) No) > 69 cases (factor(n
ewdt$survived) Yes).
## Area under the curve: 0.8677
```

Since our curve almost reaches the left corner it informs us that our model is performing well because that is what we want in our classification model . The accuracy model is .821 and the AOC is .8667 . The model performed better on the testing set than it did with the training set . The model achieved a score .808 with the training set and .821 when fitted onto the testing set. The accuracy differ from each other because the predictor variables were able to fit better on the test set even though the model is trained on the training data set. They do not diff by alot which is means our model is able to fit both data sets well.