HW 3

PSTAT 131/231 Arthur Starodynov

Contents

```
titanic.data$pclassfac <- as.factor(titanic.data$pclass)
titanic.data$survived <- as.factor(titanic.data$survived)
```

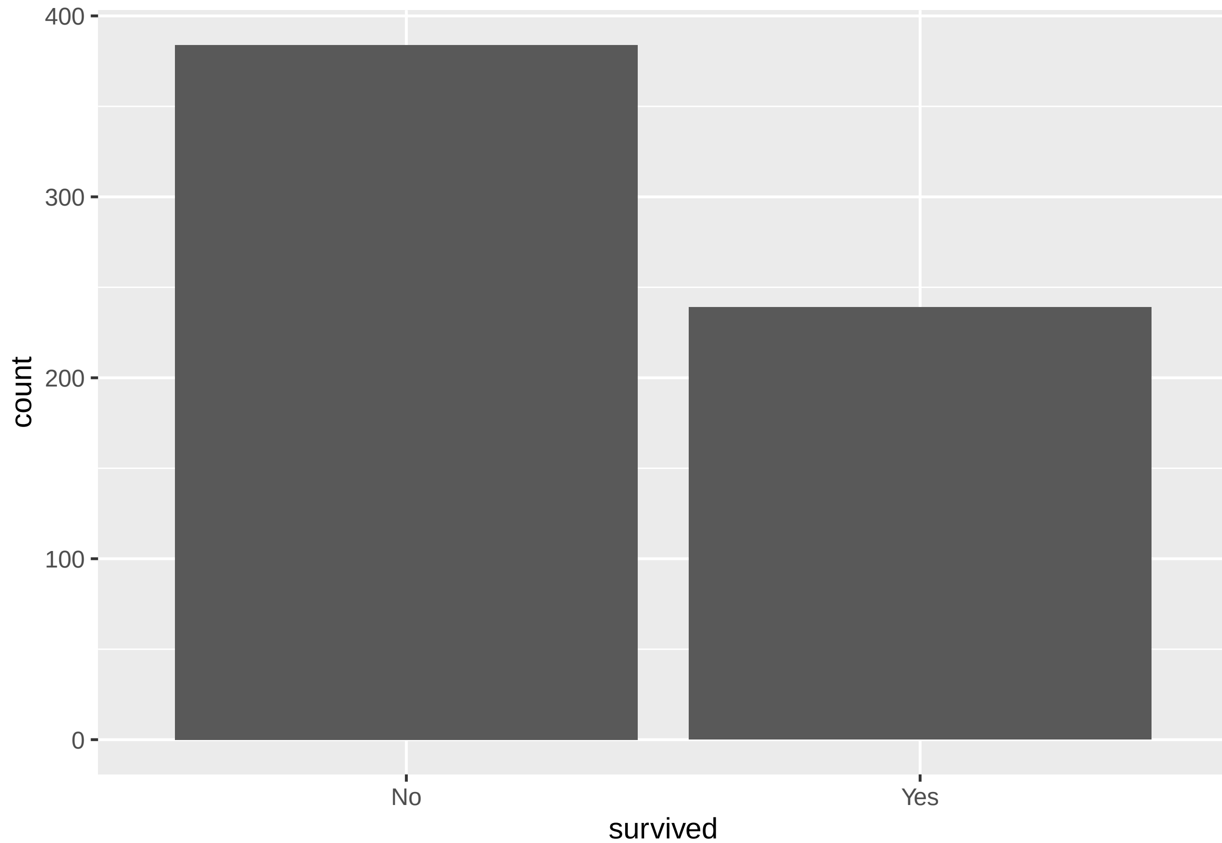
1)

```
p <- 0.7
strats <- titanic.data$survived

rr <- split(1:length(strats), strats)
idx <- sort(as.numeric(unlist(sapply(rr, function(x) sample(x, length(x) * p)))))

train <- titanic.data[idx, ]
test <- titanic.data[-idx, ]
```

We want to use stratified sample sets so that all parties and variables can get represented and classes within the training and test sets.

2)

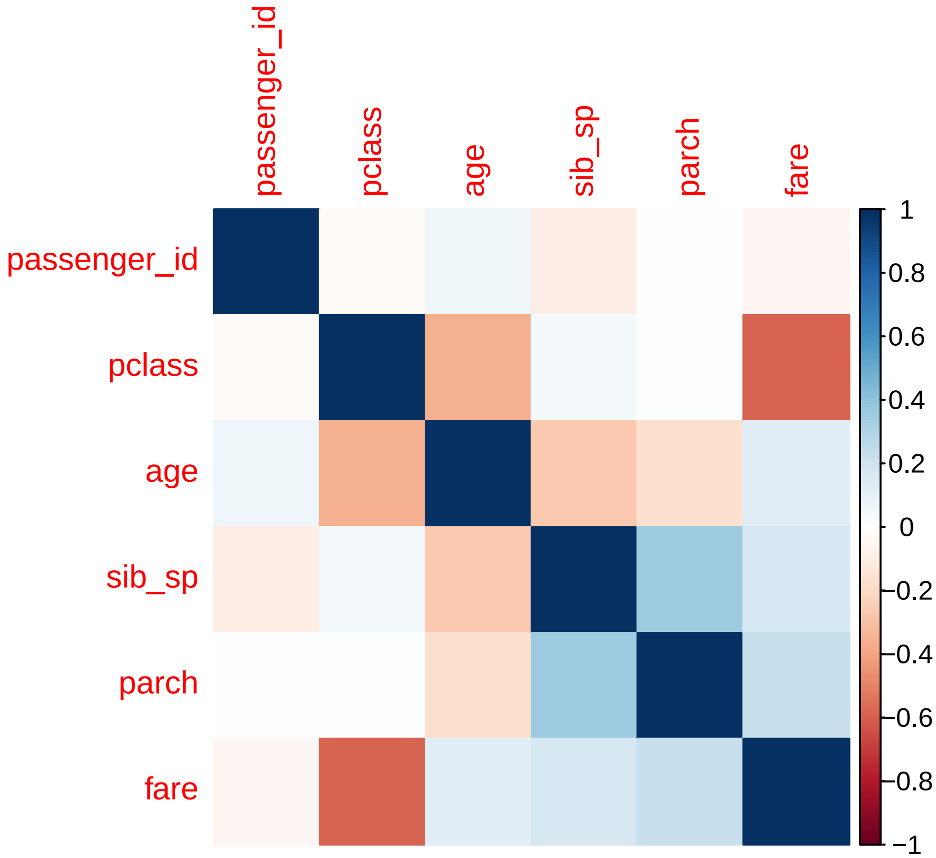
```
ggplot(train, aes(x=survived)) +
  geom_bar(aes(fill=pclass), position="dodge")
```



It is seen that more people did not survive vs those that did. And if you specify which class survived more or less it is clear that 1st class passegengers survived while third class did not.

3)

```
library(corrplot)
train2 = train[ , !(names(train) %in% c("cabin", "embarked"))] %>% copy()
train2 = train2 %>% drop_na()
corrplot(cor(train2[ , sapply(train2, is.numeric)]),
         method="color")
```



There is negative correlation between fare and pclass which represents that those who paid more will get a higher class level. In addition there is negative correlation between sib_sp and age representing that if you have siblings and parents more than likely you are of a younger age because you have parents. But there is positive correlation between parch and sib_sp as more siblings and parents means more parents and meaning more spouses.

4)

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

5)

```
tit_model = logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")
tit_workflow = workflow() %>%
  add_model(tit_model) %>%
  add_recipe(recipe_tit)
tit_fit = tit_workflow %>%
  fit(train)
```

6)

```
library(discrim)
dis_model = discrim_linear() %>%
  set_engine("MASS") %>%
  set_mode("classification")
dis_workflow = workflow() %>%
  add_model(dis_model) %>%
  add_recipe(recipe_tit)
dis_fit = dis_workflow %>%
  fit(train)
```

7)

```
quad_model = discrim_quad() %>%
  set_engine("MASS") %>%
  set_mode("classification")
quad_workflow = workflow() %>%
  add_model(quad_model) %>%
  add_recipe(recipe_tit)
quad_fit = quad_workflow %>%
  fit(train)
```

8)

```
library(klaR)
bayes_model = naive_Bayes() %>%
  set_engine("klaR", usekernel=FALSE) %>%
  set_mode("classification")
bayes_workflow = workflow() %>%
  add_model(bayes_model) %>%
  add_recipe(recipe_tit)
bayes_fit = bayes_workflow %>%
  fit(train)
```

9)

```
predict_train_model= bind_cols(predict(tit_fit, train),
                               predict(dis_fit, train),
                               predict(quad_fit, train),
                               predict(bayes_fit, train),
                               train$survived)
colnames(predict_train_model) = c("TIT Predict", "DIS Predict", "QUAD Predict",
                                  "Bayes Predict", "True")
print(accuracy(predict_train_model,
               truth='True', estimate="TIT Predict")$.estimate)
```

```
## [1] 0.8378812
```

```
print(accuracy(predict_train_model,
               truth='True', estimate="DIS Predict")$.estimate)
```

```
## [1] 0.8073836
```

```
print(accuracy(predict_train_model,
               truth='True', estimate="QUAD Predict")$.estimate)
```

```
## [1] 0.8202247
```

```
print(accuracy(predict_train_model,
               truth='True', estimate="Bayes Predict")$.estimate)
```

```
## [1] 0.8025682
```

The model that received the highest accuracy was the logistic regression model being around 84% accurate.

10)

```
New_predict_test = bind_cols(predict(tit_fit, test),
                             test$survived)
colnames(New_predict_test) = c("TIT Predict", "True")
print(accuracy(New_predict_test,
               truth="True", estimate="TIT Predict")$.estimate)
```

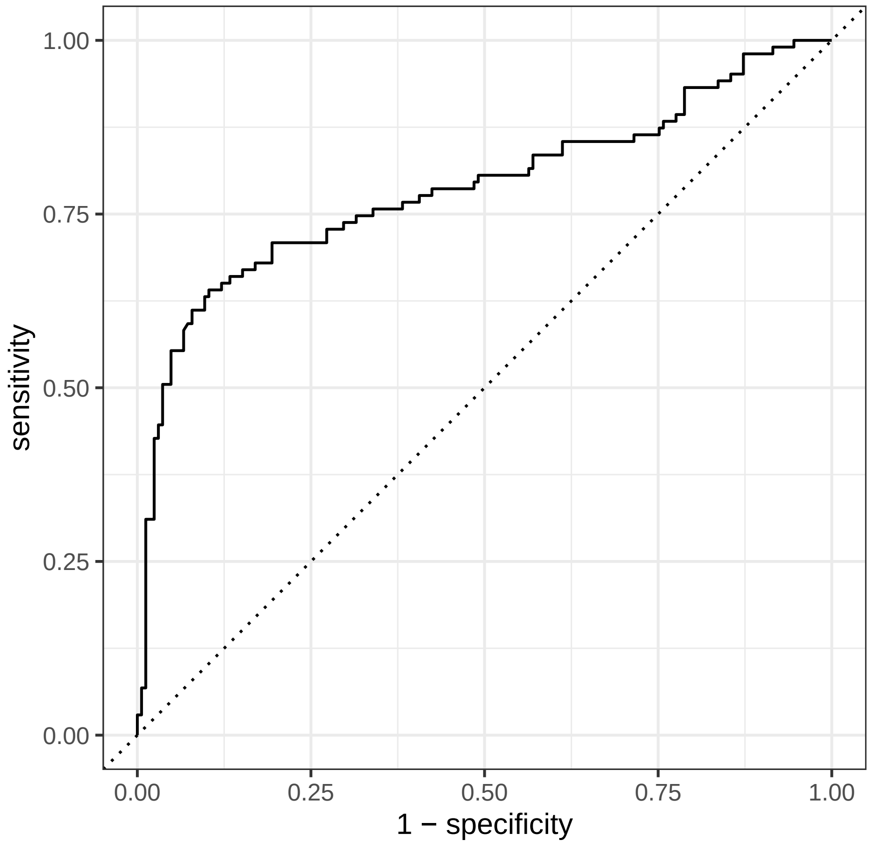
```
## [1] 0.7798507
```

78% accuracy

```
conf_mat(New_predict_test, truth="True", estimate="TIT Predict")
```

```
##           Truth
## Prediction  No Yes
##        No  141  35
##        Yes  24  68
```

```
roc_curve = tit_fit %>%
  predict(new_data=test, type="prob") %>%
  bind_cols(test) %>%
  roc_curve(survived, .pred_Yes, event_level="second")
autoplot(roc_curve)
```




```
auc_curve = tit_fit %>%
  predict(new_data=test, type="prob") %>%
  bind_cols(test) %>%
  roc_auc(survived, .pred_Yes, event_level="second")
print(auc_curve$.estimate)
```

```
## [1] 0.7868491
```

Model preformed pretty well with the relative accuracies being 84 and 78% accurate. The values differ a bit based on the stratification of the model and which observations went into where within the two test samples.