homework 3

Sabrina Lem

4/12/2022

Question 1

```
set.seed(1027)

titanic_split <- initial_split(
  titanic, prop = 0.7, strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
```

Stratification may be important because the demographics for who survived and who did not may be quite different. Stratification allows for a more representative sample of the population. *Question 2*

```
survived <- factor(titanic_train$survived)
counts <- table(survived)
prop <-prop.table(counts)
prop
```

```
## survived
##       Yes        No
## 0.3836276 0.6163724
```

We can see that survived is a binary variable based on whether the passenger survived or did not survive. About 38.36 of the passengers survived according to this sample.

```
surv_class <- table(survived,titanic_train$pclass)
surv_class
```

```
##
## survived   1   2   3
##      Yes  90  64  85
##      No   55  65 264
```

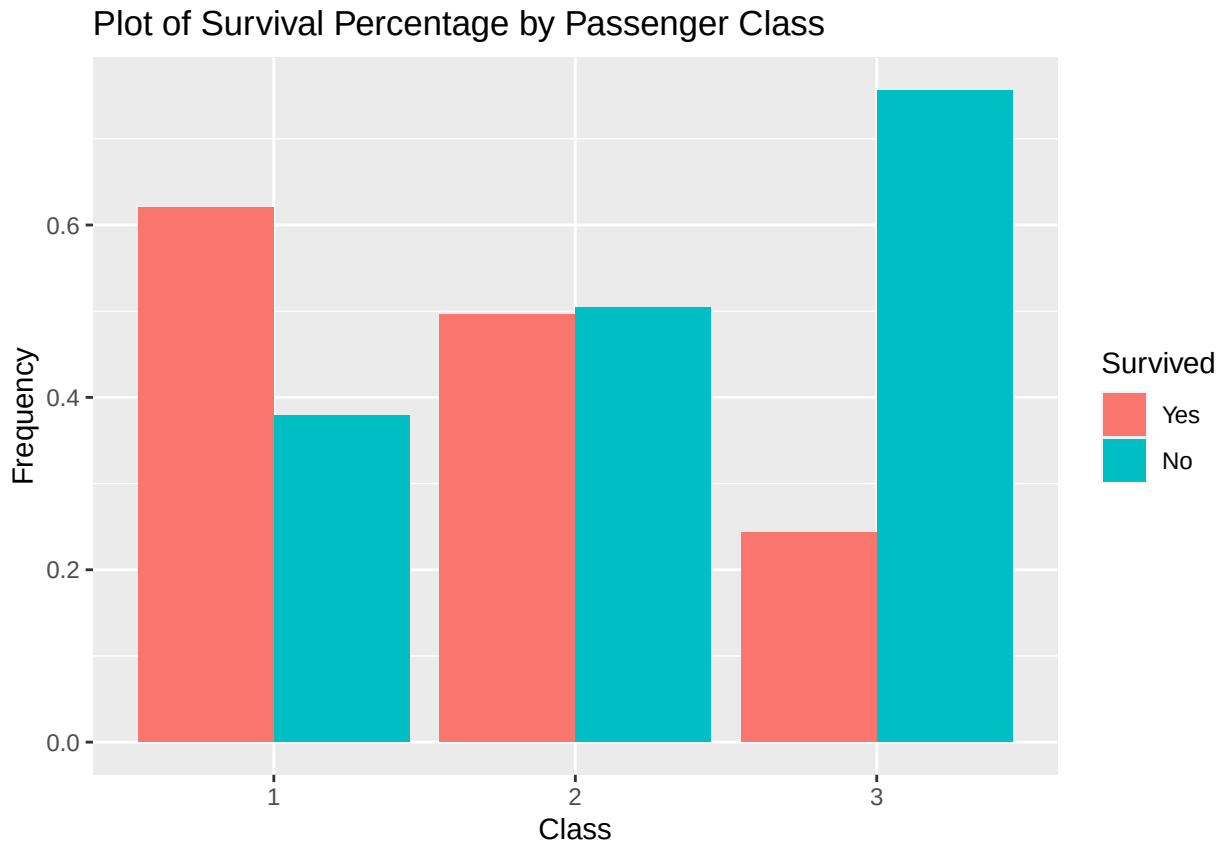
From the cross table with survived and class, it is apparent that more people survived in first class overall. First class passengers had more passengers survive than not survive, while second and third class had more people not survive than survive. First class also had the most people survive, 90 people of the sample.

```
surv_class_<- prop.table(surv_class,2)
surv_class_
```

```
##
## survived         1         2         3
##      Yes 0.6206897 0.4961240 0.2435530
##      No  0.3793103 0.5038760 0.7564470
```

The previous descriptive statistics can also be seen through proportions.

```
surv_class_<-as.data.frame(surv_class_)
names(surv_class_)<-c('Survived', 'Class', 'Frequency')
ggplot(data=surv_class_, aes(x=Class,y=Frequency, fill=Survived))+
  geom_col(position='dodge')+
  labs(title= "Plot of Survival Percentage by Passenger Class")
```



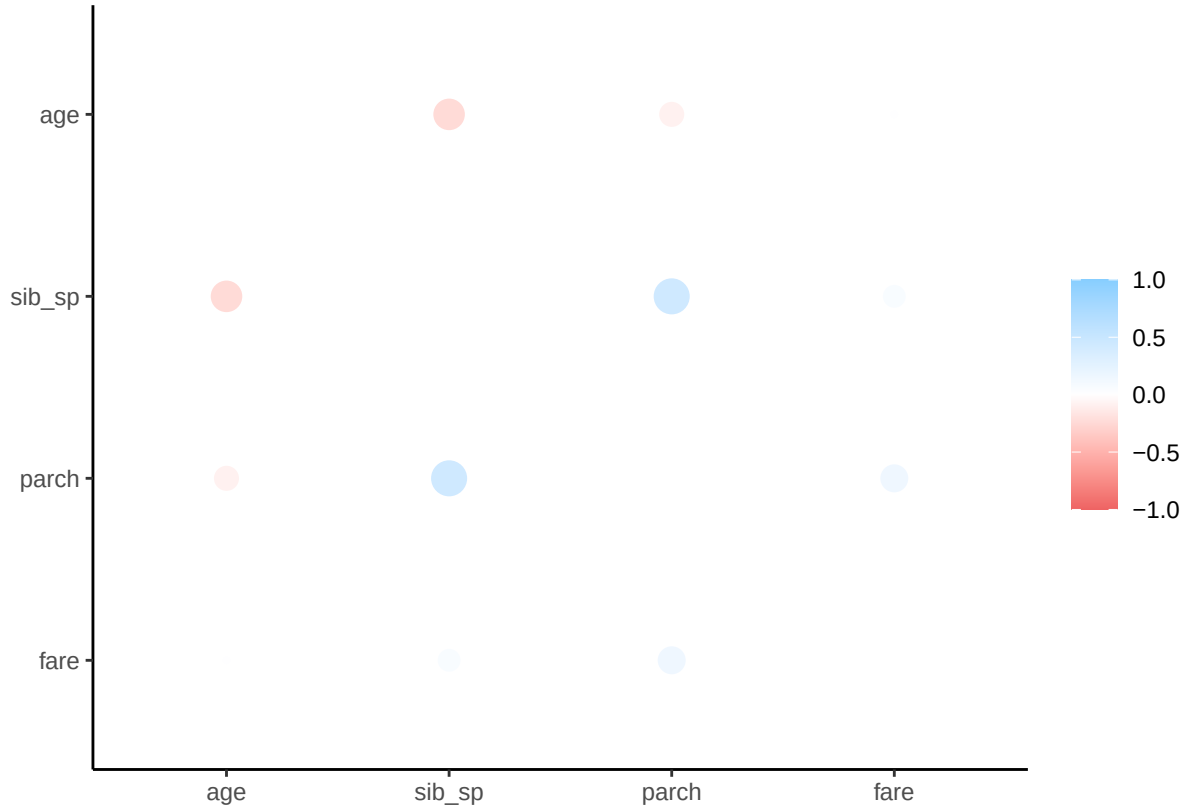
Question 3

```
titanic_train_num <- titanic_train %>%
  select(where(is.numeric))
#omit age due to missing values
titanic_train_num <-subset(titanic_train_num, select = -c(passenger_id))
cor_titanic_train <- titanic_train_num %>%
  correlate()
```

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

```
rplot(cor_titanic_train)
```

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



Visually we can see that the variables above the diagonal from the origin are negatively correlated, while the other half of the plot shows positive correlation.Age and sib_sp are negatively correlated, age and parch are negatively correlated,, parch and sib_sp are positively correlated, fare and sib_sp are positively correlated, and fare and parch are positively correlated.

Question 4

```
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(~ starts_with("sex"):fare) %>%
  step_interact(~ age:fare)
```

Question 5

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

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

log_fit <- fit(log_wkflw, titanic_train)
```

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, titanic_train)
```

Question 7

```
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

```
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

```
#Logistic Regression
log_reg_pred<-
  predict(log_fit, new_data = titanic_train, type = "prob")
log_reg_pred<-bind_cols(log_reg_pred, titanic_train %>%select(survived))


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

```
#IDA
lda_mod_pred<-
  predict(lda_fit, new_data = titanic_train, type = "prob")
lda_mod_pred<-bind_cols(lda_mod_pred, titanic_train %>%select(survived))


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

```
#QDA
qda_mod_pred<-
  predict(qda_fit, new_data = titanic_train, type = "prob")
qda_mod_pred<-bind_cols(qda_mod_pred, titanic_train %>%select(survived))


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

```
#Naive Bayesian
nb_mod_pred<-
  predict(nb_fit, new_data = titanic_train, type = "prob")
nb_mod_pred<-bind_cols(nb_mod_pred, titanic_train%>%select(survived))


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

```
accuracies <-c(log_reg_acc$.estimate, lda_mod_acc$.estimate,
               qda_mod_acc$.estimate, nb_mod_acc$.estimate)
models <- c("Logistic Regression", "LDA", "QDA", "Naive Bayes")
results <-tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.812 Logistic Regression
## 2      0.799 LDA
## 3      0.782 QDA
## 4      0.770 Naive Bayes
```

The Logistic model performed the best on the training set, with the highest accuracy of .8121.

Question 10

```
# PREDICTED VALUES AND ACCURACY
log_reg_pred_test<-predict(log_fit, new_data = titanic_test, type = "prob")
head(log_reg_pred_test)
```

```
## # A tibble: 6 x 2
##   .pred_Yes .pred_No
##       <dbl>    <dbl>
## 1    0.905    0.0955
## 2    0.118    0.882
## 3    0.299    0.701
## 4    0.0949   0.905
## 5    0.742    0.258
## 6    0.343    0.657
```

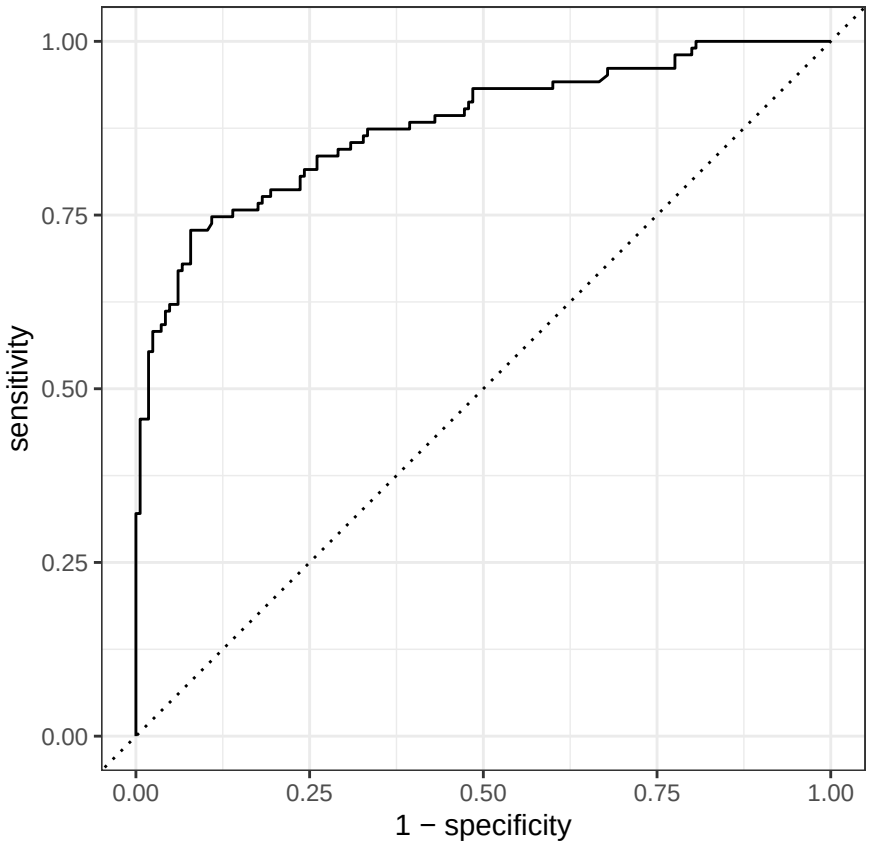
```
log_reg_pred_test_acc <- augment(log_fit, new_data = titanic_test) %>%
  accuracy(truth = factor(survived), estimate = .pred_class)
log_reg_pred_test_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.832
```

```
# CONFUSION MATRIX
augment(log_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes  70  12
##        No   33 153
```

```
# ROC AND AUC
augment(log_fit, new_data = titanic_test) %>%
  roc_curve(titanic_test$survived, .pred_Yes) %>%
  autoplot()
```




```
area <- augment(log_fit, new_data = titanic_test) %>%
  roc_auc(titanic_test$survived, .pred_Yes)
area
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.878
```

The accuracy is greater on the test data, so the model is not a good fit. The model most likely overfits the data.