Homework 3

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.0.5

## -- Attaching packages -------------------------------------- tidymodels 0.2.0 --

## v broom        0.8.0     v rsample      0.1.1
## v dials        0.1.1     v tune         0.2.0
## v infer        1.0.0     v workflows    0.2.6
## v modeldata    0.1.1     v workflowsets 0.2.1
## v parsnip      0.2.1     v yardstick    0.0.9
## v recipes      0.2.0

## Warning: package 'dials' was built under R version 4.0.5

## Warning: package 'parsnip' was built under R version 4.0.5

## Warning: package 'recipes' was built under R version 4.0.5

## Warning: package 'tune' was built under R version 4.0.5
```

```
## Warning: package 'workflows' was built under R version 4.0.5

## Warning: package 'workflowsets' was built under R version 4.0.5

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggthemes)
library(tidymodels)
library(ISLR)
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.0.5

##
## Attaching package: 'ISLR2'

## The following objects are masked from 'package:ISLR':
##
##     Auto, Credit
```

```
library(discrim)
```

```
## Warning: package 'discrim' was built under R version 4.0.5

##
## Attaching package: 'discrim'

## The following object is masked from 'package:dials':
##
##     smoothness
```

```
library(poissonreg)
```

```
## Warning: package 'poissonreg' was built under R version 4.0.5
```

```
library(corrr)
library(klaR)
```

```
## Warning: package 'klaR' was built under R version 4.0.5

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.0.5

##
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':
##
##     Boston

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(yardstick)
titanic <- read.csv('titanic.csv')
```

```
titanic$pclass <- as.factor(titanic$pclass)
titanic$survived <- as.factor(titanic$survived)
titanic$survived <- relevel(titanic$survived, ref = 'Yes')
levels(titanic$survived)
```

```
## [1] "Yes" "No"
```

Question 1:

```
set.seed(608)
titanicSplit <- initial_split(titanic, prop = 0.80,
                              strata = survived)
titanic_train <- training(titanicSplit)
titanic_test <- testing(titanicSplit)
```

Training set has 712 obvs and testing set has 179

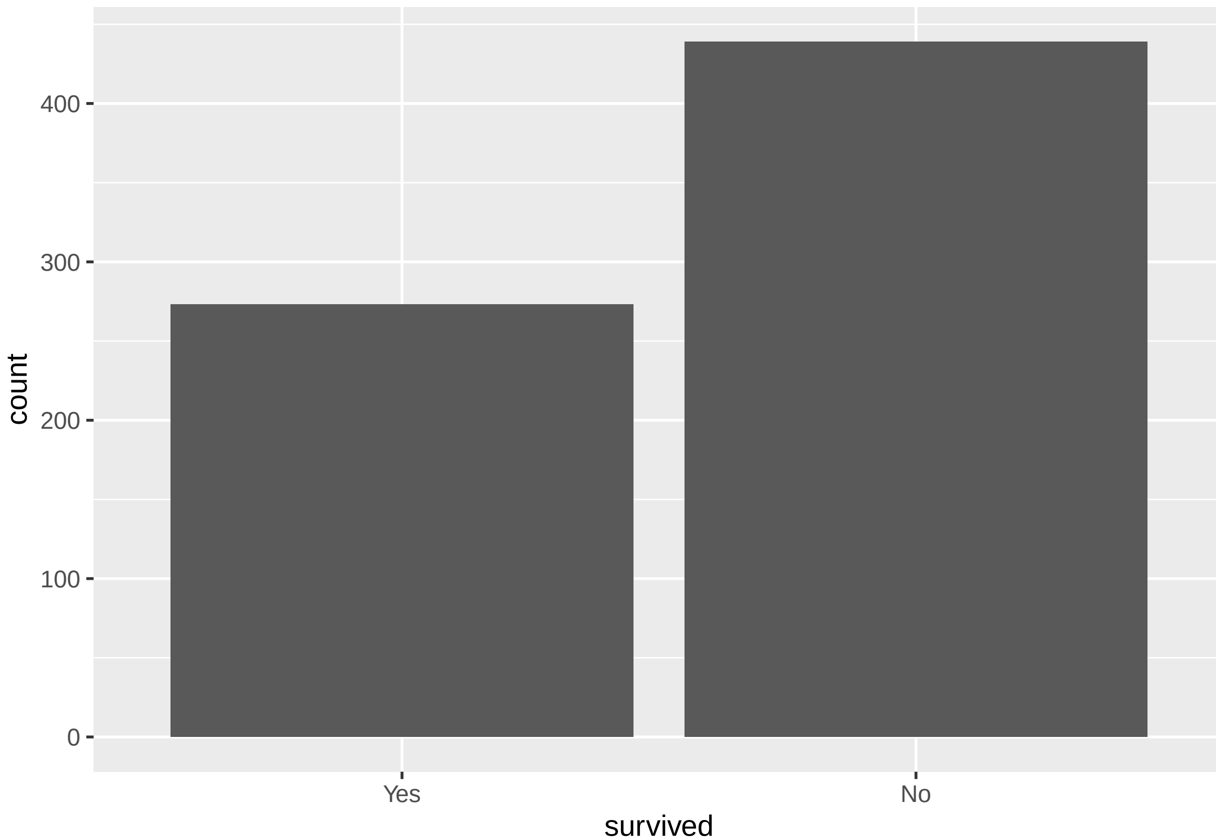
```
head(titanic_train)
```

```
##    passenger_id survived pclass                           name  sex age sib_sp
## 1             1       No      3        Braund, Mr. Owen Harris male  22      1
## 5             5       No      3       Allen, Mr. William Henry male  35      0
## 6             6       No      3               Moran, Mr. James male  NA      0
## 7             7       No      1        McCarthy, Mr. Timothy J male  54      0
## 13           13       No      3 Saundercock, Mr. William Henry male  20      0
## 14           14       No      3    Andersson, Mr. Anders Johan male  39      1
##    parch    ticket    fare cabin embarked
## 1      0 A/5 21171  7.2500  <NA>        S
## 5      0    373450  8.0500  <NA>        S
## 6      0    330877  8.4583  <NA>        Q
## 7      0     17463 51.8625   E46        S
## 13     0 A/5. 2151  8.0500  <NA>        S
## 14     5    347082 31.2750  <NA>        S
```

There is missing data in Cabin and in age for some passengers It is a good idea to use stratified sampling for this data so it is not skewed towards one survival outcome.

Question 2:

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



Training data contains more obs from passengers who did not survive than those who did

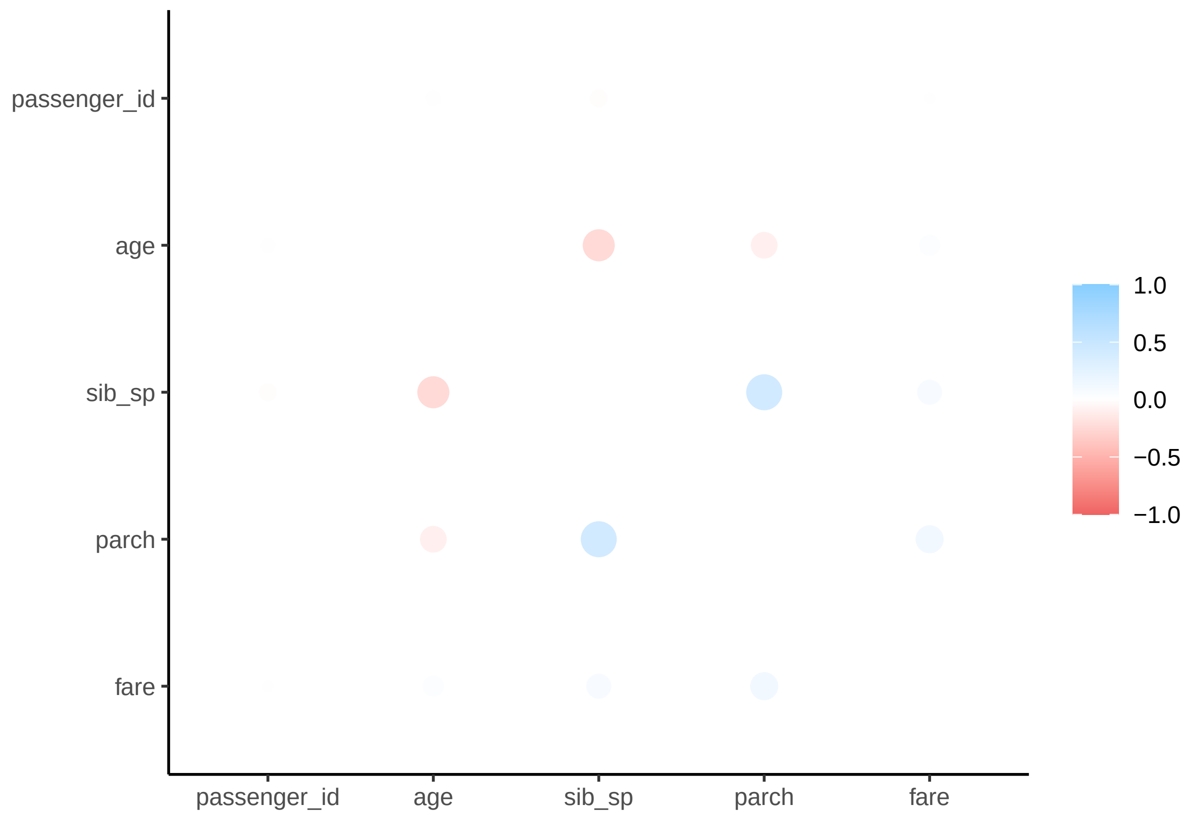
Question 3:

```
cor_titanic <- titanic %>%
  select_if(is.numeric) %>%
  correlate()
```

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

```
rplot(cor_titanic)
```

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



Negatively Correlated: (Age, Parch), (age, sib_sp) Positively Correlated: (sib_sp, parch), (parch, fare)

Question 4:

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

Question 5:

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

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

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

Question 6:

```
linDisc_mod <- discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")

linDisc_wkflow <- workflow() %>%
```

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

linDisc_fit <- fit(linDisc_wkflow, titanic_train)
```

Question 7:

```
quadDisc_mod <- discrim_quad() %>%
  set_mode("classification") %>%
  set_engine("MASS")

quadDisc_wkflow <- workflow() %>%
  add_model(quadDisc_mod) %>%
  add_recipe(titanic_recipe)

quadDisc_fit <- fit(quadDisc_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:

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

```
## # A tibble: 712 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1    0.100     0.900
##  2    0.0731    0.927
##  3    0.104     0.896
##  4    0.317     0.683
##  5    0.167     0.833
##  6    0.0311    0.969
##  7    0.0629    0.937
##  8    0.540     0.460
##  9    0.149     0.851
## 10    0.104     0.896
## # ... with 702 more rows
```

```
predict(linDisc_fit, new_data = titanic_train, type = "prob")
```

```
## # A tibble: 712 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1    0.0637    0.936
##  2    0.0468    0.953
##  3    0.0651    0.935
##  4    0.257     0.743
##  5    0.102     0.898
##  6    0.0199    0.980
##  7    0.0483    0.952
##  8    0.644     0.356
##  9    0.280     0.720
## 10    0.0649    0.935
## # ... with 702 more rows
```

```
predict(quadDisc_fit, new_data = titanic_train, type = "prob")
```

```
## # A tibble: 712 x 2
##      .pred_Yes .pred_No
##          <dbl>    <dbl>
##  1 0.00340     9.97e- 1
##  2 0.00230     9.98e- 1
##  3 0.00359     9.96e- 1
##  4 0.0447      9.55e- 1
##  5 0.00644     9.94e- 1
##  6 0.196       8.04e- 1
##  7 0.000000215 1.00e+ 0
##  8 0.000693    9.99e- 1
##  9 1.00        7.80e-16
## 10 0.00357     9.96e- 1
## # ... with 702 more rows
```

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

```
## # A tibble: 712 x 2
##     .pred_Yes .pred_No
##         <dbl>    <dbl>
##  1 0.00884    9.91e- 1
##  2 0.00801    9.92e- 1
##  3 0.00848    9.92e- 1
##  4 0.489      5.11e- 1
##  5 0.00969    9.90e- 1
##  6 0.0546     9.45e- 1
##  7 0.00000259 1.00e+ 0
##  8 0.00658    9.93e- 1
##  9 1          8.34e-27
## 10 0.00859    9.91e- 1
## # ... with 702 more rows
```

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

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.796
```

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


quadDisc_acc <- augment(quadDisc_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)


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

linDisc_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.796
```

```r
log_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.808
```

```r
quadDisc_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.768
```

```r
nb_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.756
```

log model has the highest accuracy at 0.8075843

Question 10:

```r
predict(log_fit, new_data = titanic_test, type = "prob")
```

```
## # A tibble: 179 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1     0.106    0.894
##  2     0.787    0.213
##  3     0.469    0.531
##  4     0.239    0.761
##  5     0.104    0.896
##  6     0.633    0.367
##  7     0.159    0.841
##  8     0.536    0.464
##  9     0.698    0.302
## 10     0.104    0.896
## # ... with 169 more rows
```

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


mod_acc
```

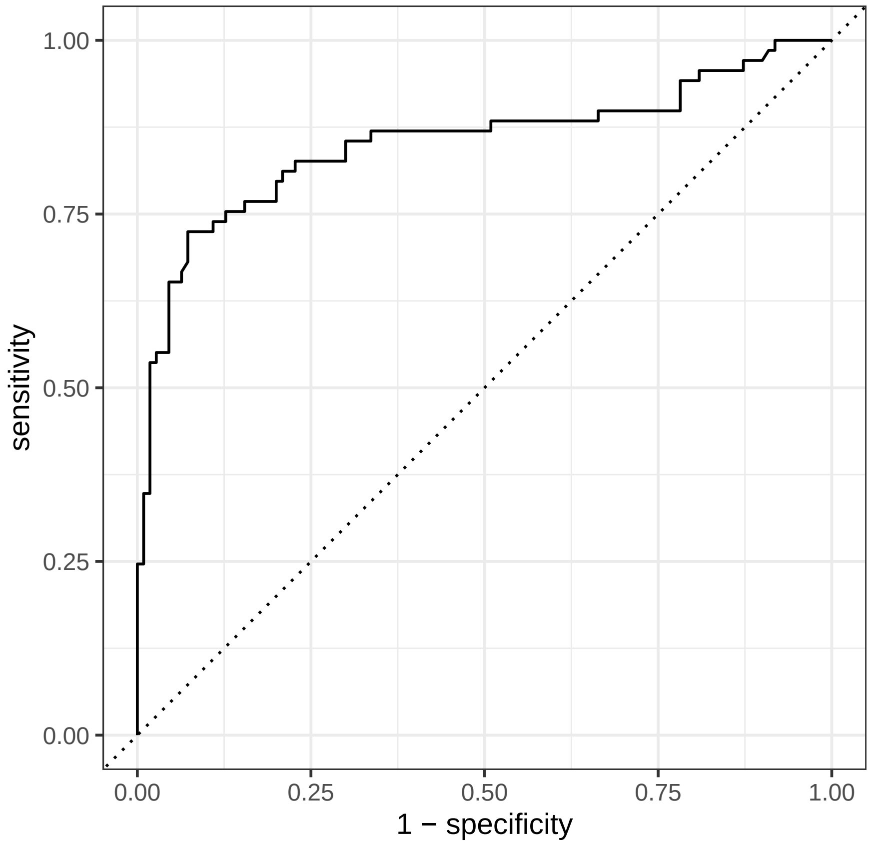
```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.827
```

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

```
##           Truth
## Prediction Yes No
##        Yes  52 14
##        No   17 96
```

.8268156 accuracy on the testing data

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




```r
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         0.852
```

the area under the curve is 0.8524374 The model performed pretty well. Around 85% accuracy. The testing accuracy is about 0.02 higher than the training which I assume is a result of underfitting the model to the training data.