Homework 3

PSTAT 131/231

Classification

For this assignment, we will be working with part of a Kaggle data set that was the subject of a machine learning competition and is often used for practicing ML models. The goal is classification; specifically, to predict which passengers would survive the Titanic shipwreck.

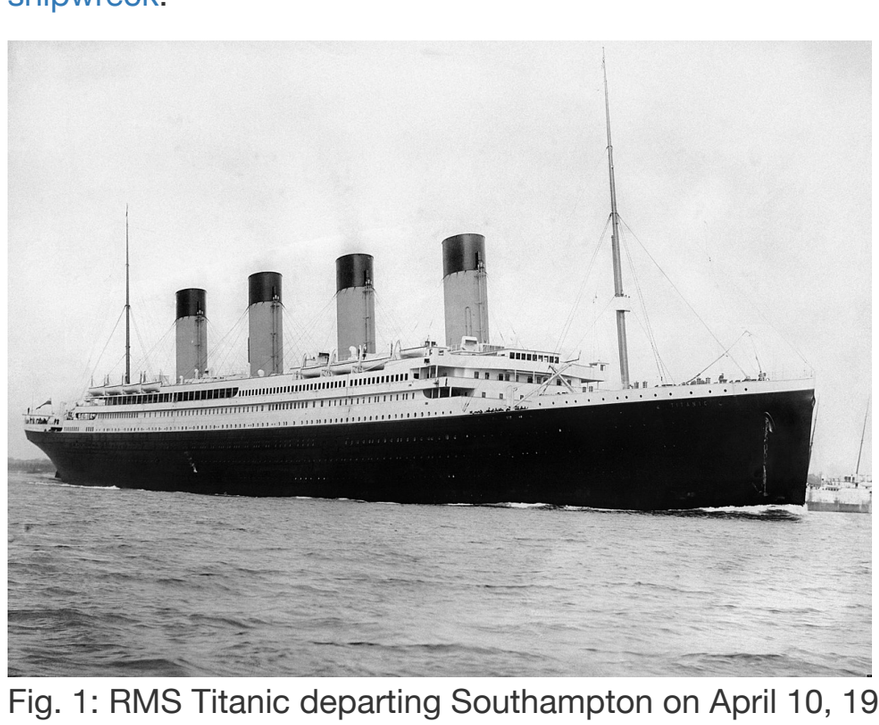


Fig. 1: RMS Titanic departing Southampton on April 10, 1912.

Load the data from `data/titanic.csv` into *R* and familiarize yourself with the variables it contains using the codebook (`data/titanic_codebook.txt`).

Notice that `survived` and `pclass` should be changed to factors. When changing `survived` to a factor, you may want to reorder the factor so that *"Yes"* is the first level.

Make sure you load the `tidyverse` and `tidymodels`!

Remember that you'll need to set a seed at the beginning of the document to reproduce your results.

```
library(ggplot2)
library(tidyverse)
library(tidymodels)
library(corrplot)
library(ggthemes)
library(corrr)
library(discrim)
#install.packages("pROC")
library(pROC)
library(klaR)
tidymodels_prefer()
Titanic_data=read.csv("titanic.csv")
Titanic_data$survived <- factor(Titanic_data$survived)
Titanic_data$pclass <- factor(Titanic_data$pclass)
Titanic_data %>%
  head()
```

```
##   passenger_id survived pclass
## 1            1       No      3
## 2            2      Yes      1
## 3            3      Yes      3
## 4            4      Yes      1
## 5            5       No      3
## 6            6       No      3
##                                                  name    sex age sib_sp parch
## 1                             Braund, Mr. Owen Harris   male  22      1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1     0
## 3                              Heikkinen, Miss. Laina female  26      0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1     0
## 5                            Allen, Mr. William Henry   male  35      0     0
## 6                                    Moran, Mr. James   male  NA      0     0
##             ticket    fare cabin embarked
## 1        A/5 21171  7.2500  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250  <NA>        S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

Question 1

Split the data, stratifying on the outcome variable, `survived.` You should choose the proportions to split the data into. Verify that the training and testing data sets have the appropriate number of observations. Take a look at the training data and note any potential issues, such as missing data.

```
set.seed(3435)

titanic_split <- initial_split(Titanic_data, prop = 0.70,
                                strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)

titanic_train %>%
  head()
```

```
##    passenger_id survived pclass                               name    sex age
## 1             1       No      3            Braund, Mr. Owen Harris   male  22
## 5             5       No      3           Allen, Mr. William Henry   male  35
## 7             7       No      1            McCarthy, Mr. Timothy J   male  54
## 8             8       No      3     Palsson, Master. Gosta Leonard   male   2
## 14           14       No      3        Andersson, Mr. Anders Johan   male  39
## 15           15       No      3 Vestrom, Miss. Hulda Amanda Adolfina female  14
##    sib_sp parch    ticket    fare cabin embarked
## 1       1     0 A/5 21171  7.2500  <NA>        S
## 5       0     0    373450  8.0500  <NA>        S
## 7       0     0     17463 51.8625   E46        S
## 8       3     1    349909 21.0750  <NA>        S
## 14      1     5    347082 31.2750  <NA>        S
## 15      0     0    350406  7.8542  <NA>        S
```

The primary issue I see is that we have a lot of missing data with the variable "cabin". We also see a bit of missing data with the variable "age".

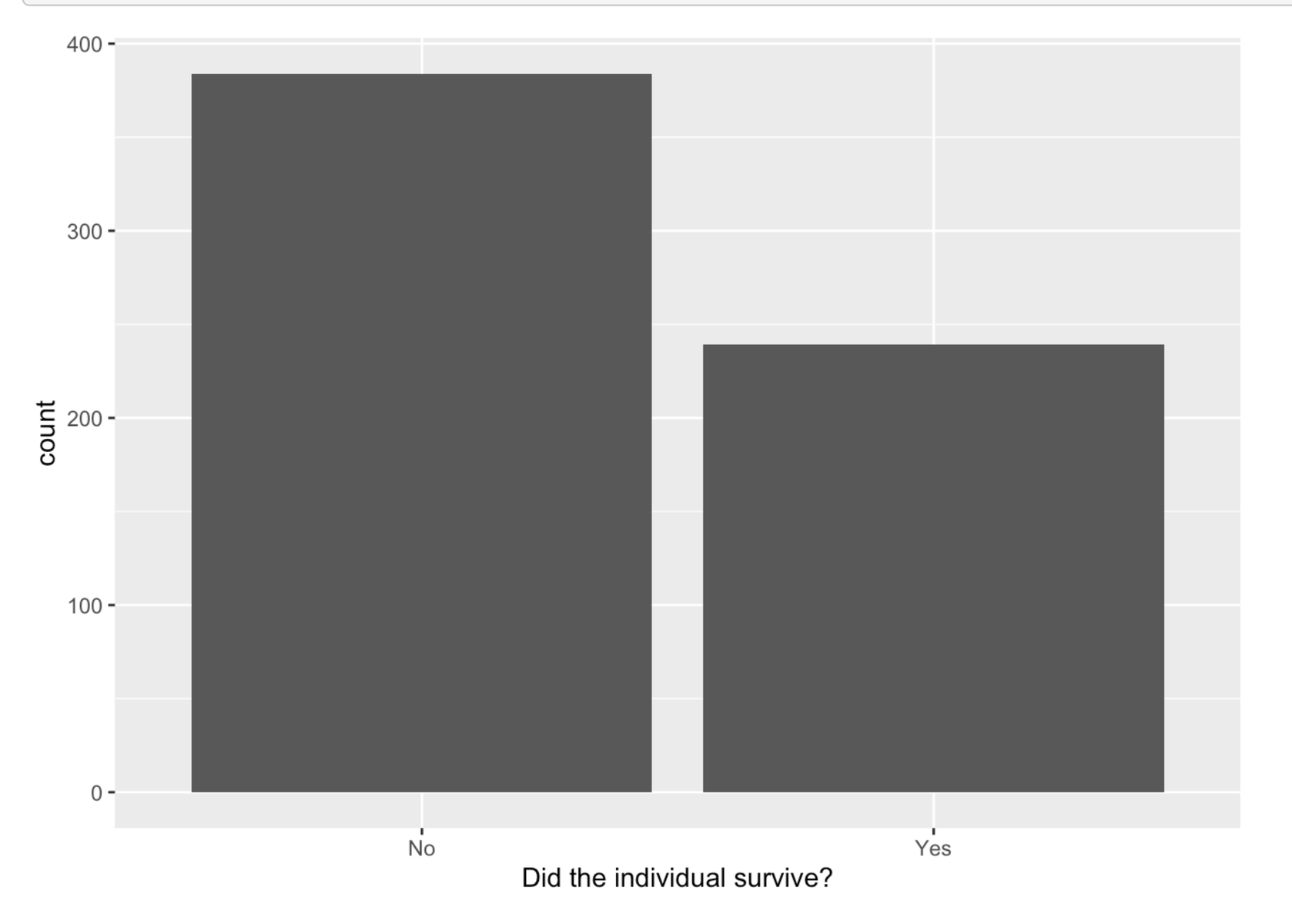
Why is it a good idea to use stratified sampling for this data?

Stratified sampling allows us to properly represent each subgroup within our sample. By using stratified sampling, we can accurately represent the features of those who survived and those who died appropriately.

Question 2

Using the **training** data set, explore/describe the distribution of the outcome variable `survived`.

```
 titanic_train %>%
  ggplot(aes(x = survived)) +
  geom_bar() + xlab("Did the individual survive?")
```

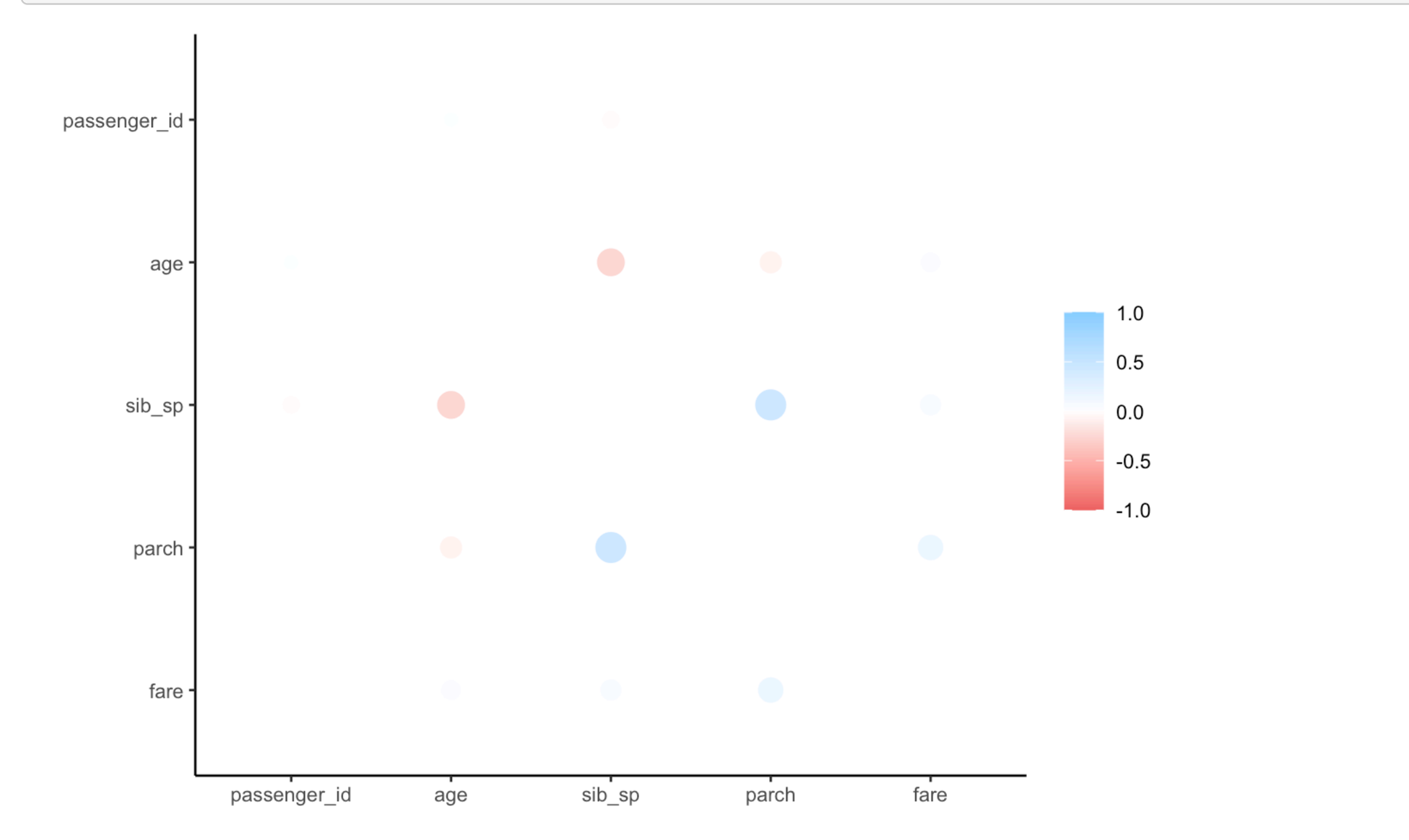


The chart above indicates the distribution of our outcome variable "survived". Here, we see that our training data includes significantly more people who did not survive than who did. This makes sense as most people on the Titanic did not end up surviving.

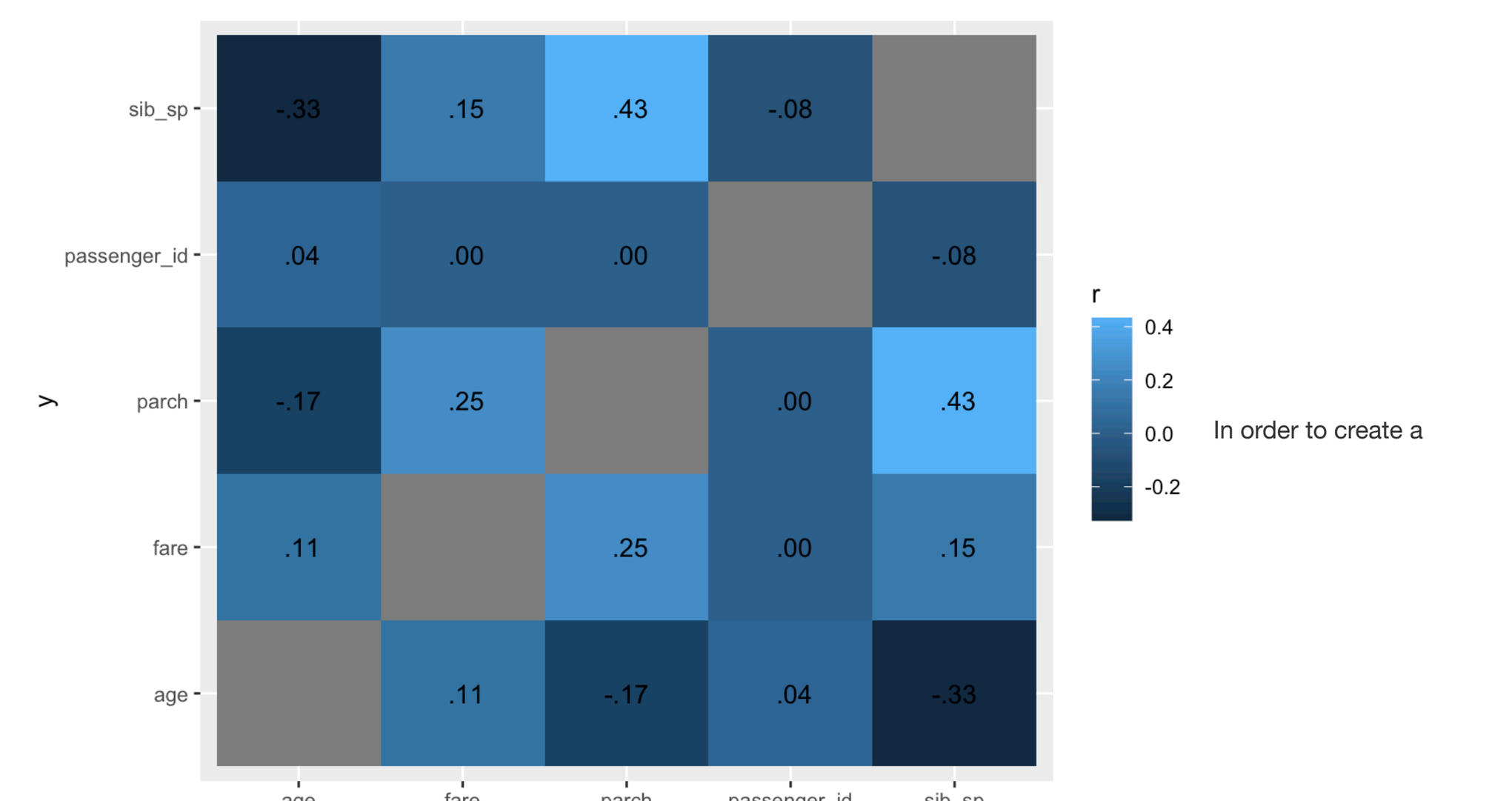
Question 3

Using the **training** data set, create a correlation matrix of all continuous variables. Create a visualization of the matrix, and describe any patterns you see. Are any predictors correlated with each other? Which ones, and in which direction?

```
cor_Titanic <- titanic_train %>%
  select(-survived) %>%
  select(-pclass) %>%
  select(-name) %>%
  select(-sex) %>%
  select(-ticket) %>%
  select(-cabin) %>%
  select(-embarked) %>%
  correlate()
rplot(cor_Titanic)
```




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



In order to create a correlation matrix, we had to drop a number of predictors from our dataset. However, the correlation matrix above indicates that we do see a moderate positive correlation (.43) between the variables of sib_sp and parch. We also see a light positive correlation (.25) with variables fare and parch and a light negative correlation (-.33) with variables age and sib_sp.

Question 4

Using the **training** data, create a recipe predicting the outcome variable `survived`. Include the following predictors: ticket class, sex, age, number of siblings or spouses aboard, number of parents or children aboard, and passenger fare.

Recall that there were missing values for `age`. To deal with this, add an imputation step using `step_impute_linear()`. Next, use `step_dummy()` to **dummy** encode categorical predictors. Finally, include interactions between:

- Sex and passenger fare, and
- Age and passenger fare.

You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

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

Question 5

Specify a **logistic regression** model for classification using the `"glm"` engine. Then create a workflow. Add your model and the appropriate recipe. Finally, use `fit()` to apply your workflow to the **training** data.

```
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)
```

Hint: Make sure to store the results of `fit()`. You'll need them later on.

Repeat Question 5, but this time specify a linear discriminant analysis model for classification using the `"MASS"` engine.

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)
```

Repeat Question 5, but this time specify a quadratic discriminant analysis model for classification using the `"MASS"` engine.

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

Repeat Question 5, but this time specify a naive Bayes model for classification using the `"klaR"` engine. Set the `usekernel` argument to `FALSE`.

```
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

Now you've fit four different models to your training data.

Use `predict()` and `bind_cols()` to generate predictions using each of these 4 models and your **training** data. Then use the *accuracy* metric to assess the performance of each of the four models.

Which model achieved the highest accuracy on the training data?

```
a <- predict(log_fit, new_data = titanic_train, type = "prob")
b <- predict(lda_fit, new_data = titanic_train, type = "prob")
c <- predict(qda_fit, new_data = titanic_train, type = "prob")
d <- predict(nb_fit, new_data = titanic_train, type = "prob")
new_column <- bind_cols(a,b,c,d)
new_column
```

```
## # A tibble: 623 × 8
##    .pred_No...1 .pred_Yes...2 .pred_No...3 .pred_Yes...4 .pred_No...5
##           <dbl>         <dbl>        <dbl>         <dbl>        <dbl>
##  1        0.894        0.106         0.926        0.0741     9.56e- 1
##  2        0.922        0.0776        0.946        0.0536     9.71e- 1
##  3        0.674        0.326         0.726        0.274      6.94e- 1
##  4        0.885        0.115         0.916        0.0844     1.00e+ 0
##  5        0.977        0.0233        0.985        0.0155     1.00e+ 0
##  6        0.223        0.777         0.191        0.809      2.61e- 1
##  7        0.924        0.0756        0.942        0.0580     1.00e+ 0
##  8        0.530        0.470         0.458        0.542      3.80e- 1
##  9        0.459        0.541         0.382        0.618      9.96e- 1
## 10        0.776        0.224         0.762        0.238      1.23e-11
## # … with 613 more rows, and 3 more variables: .pred_Yes...6 <dbl>,
## #   .pred_No...7 <dbl>, .pred_Yes...8 <dbl>
```

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

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

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

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

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

```
## # A tibble: 4 × 2
##   accuracies models
##        <dbl> <chr>
## 1      0.812 Logistic Regression
## 2      0.795 LDA
## 3      0.793 QDA
## 4      0.783 Naive Bayes
```

The first model we fit (the logistic regression) has the highest accuracy.

Question 10

Fit the model with the highest training accuracy to the **testing** data. Report the accuracy of the model on the **testing** data.

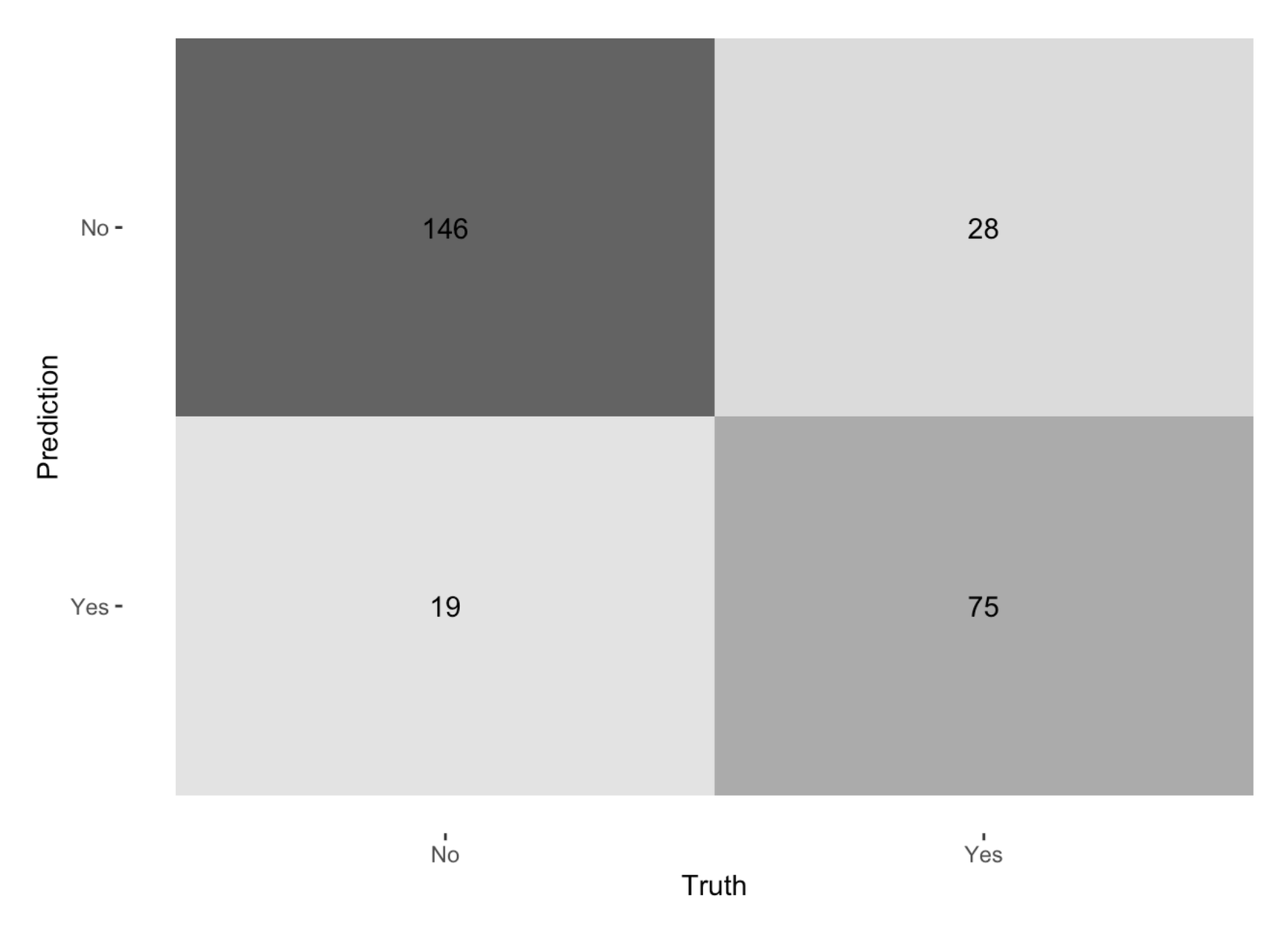
Again using the **testing** data, create a confusion matrix and visualize it. Plot an ROC curve and calculate the area under it (AUC).

How did the model perform? Compare its training and testing accuracies. If the values differ, why do you think this is so?

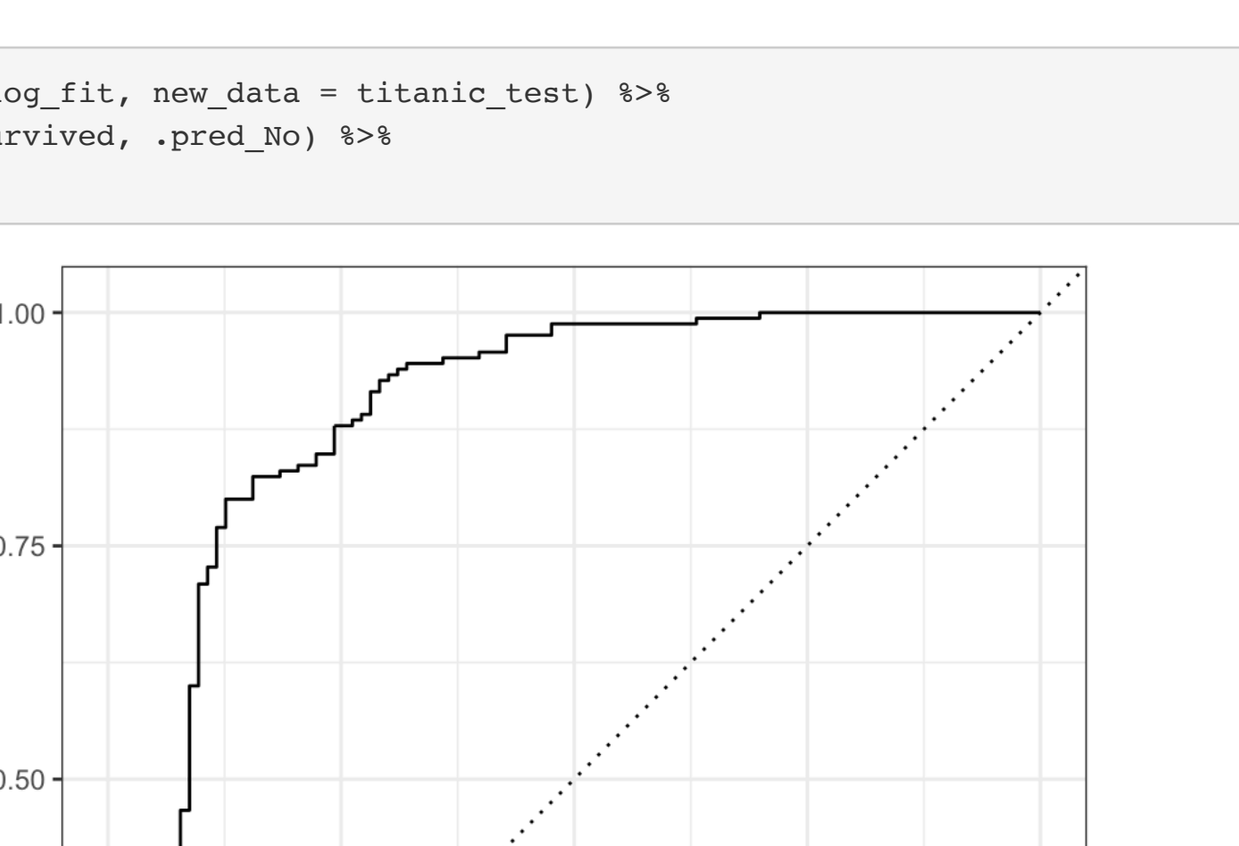
```
new_log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

new_log_wkflow <- workflow() %>%
  add_model(new_log_reg) %>%
  add_recipe(titanic_recipe)

new_log_fit <- fit(new_log_wkflow, titanic_test)

new_log_reg_acc <- augment(new_log_fit, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)

new_log_reg_acc
```

```
## # A tibble: 1 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.825
```

```
augment(new_log_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class) %>%
  autoplot(type = "heatmap")
```




```
 augment(new_log_fit, new_data = titanic_test) %>%
  roc_curve(survived, .pred_No) %>%
  autoplot()
```




```
auc(titanic_test$survived, titanic_test$fare)
```

```
## Area under the curve: 0.7081
```

Overall, the logistic regression model that was fit to the testing data had an 82.4% accuracy rate. This is very similar to the previous model as the logistic regression model that was fit to the training data had an 81.4% accuracy rate. I believe it is also fair to say that the model did not perform well as it will incorrectly predict the survival status of approximately 20% of all observations.

Required for 231 Students

In a binary classification problem, let p represent the probability of class label 1, which implies that $1-p$ represents the probability of class label 0. The *logistic function* (also called the "inverse logit") is the cumulative distribution function of the logistic distribution, which maps a real number *z* to the open interval $(0,1)$.

Question 11

Given that:

$$p(z) = \frac{e^z}{1+e^z}$$

Prove that the inverse of a logistic function is indeed the *logit* function:

$$z(p) = ln\left(\frac{p}{1-p}\right)$$

Question 12

Assume that $z = \beta_0 + \beta_1 x_1$ and $p = logistic(z)$. How do the odds of the outcome change if you increase x_1 by two? Demonstrate this.

Assume now that β_1 is negative. What value does p approach as x_1 approaches ∞? What value does p approach as x_1 approaches $-\infty$?