PSTAT131_HW3

Luke Todd

4/18/2022

```r
titanic$survived <- factor(titanic$survived, levels = c("Yes", "No"))
levels(titanic$survived)
```

Changing survived and pclass to factors

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

```r
titanic$pclass <- factor(titanic$pclass)
class(titanic$pclass)
```

```
## [1] "factor"
```

```r
# titanic <- read_csv(file = "data/titanic.csv") %>%
#   mutate(survived = factor(survived,
#                            levels = c("Yes", "No")),
#          pclass = factor(pclass))
```

Question 1

```r
titanic_split <- initial_split(titanic, prop = 0.8,
                               strata = survived)

titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)

dim(titanic_train)
```

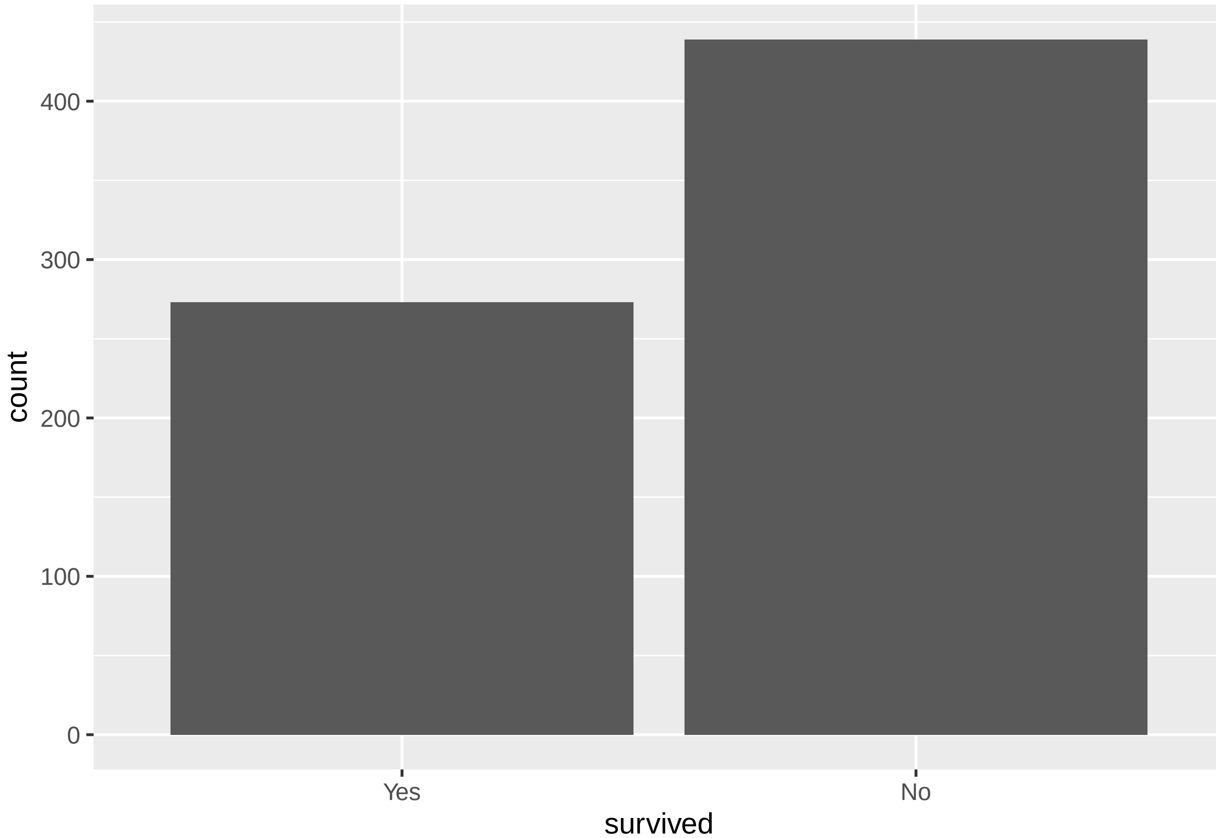
```
## [1] 712  12
```

I think that using stratified sampling is good for this data because we won't get a skewed random sample. It better represents the entire population.

Question 2

```r
# plot(titanic_train$survived)

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



The survived variable's distribution shows that it was more common for someone on the Titanic to die, than it was for them to survive.

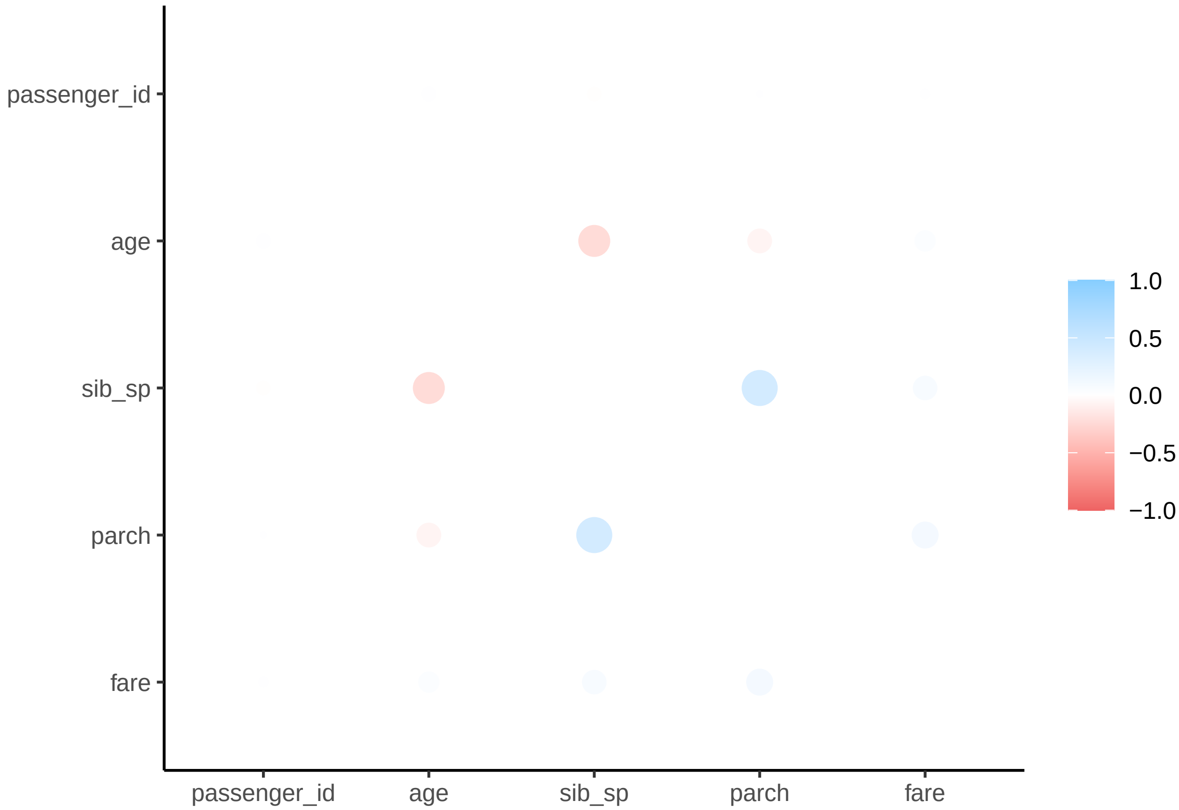
Question 3

```r
cor_titanic <- titanic_train %>%
  select(-survived, -pclass, -name, -sex, -ticket, -cabin, -embarked) %>%
  correlate()
```

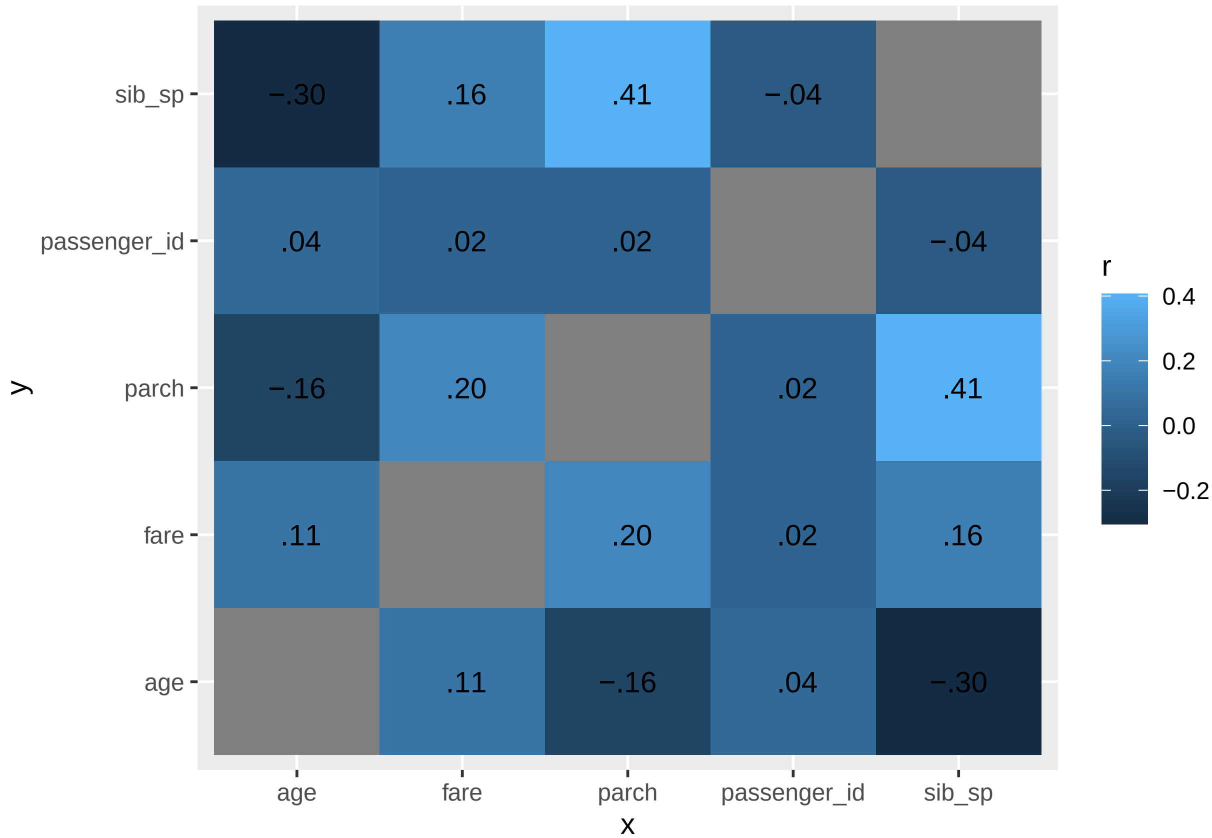
```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```r
rplot(cor_titanic)
```

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




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



The highest levels of correlation are between sib_sp and age, with a negative correlation of 0.31, as well as sib_sp and parch, with a positive correlation of 0.43. These correlation levels are not very high.

Question 4

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

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)
```

```
## Warning: Interaction specification failed for: ~sex:fare + age:fare. No
## interactions will be created.
```

```r
log_fit %>% tidy()
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -0.600     0.158     -3.80 1.46e- 4
## 2 pclass_X2      0.711     0.230      3.10 1.97e- 3
## 3 pclass_X3      1.79      0.198      9.02 1.85e-19
```

Question 6

```r
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)
```

```
## Warning: Interaction specification failed for: ~sex:fare + age:fare. No
## interactions will be created.
```

Question 7

```r
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)
```

```
## Warning: Interaction specification failed for: ~sex:fare + age:fare. No
## interactions will be created.
```

Question 8

```r
nb_mod <- naive_Bayes() %>%
  set_mode("classification") %>%
  set_engine("klaR") %>%
```

```r
  set_args(usekernel = FALSE)

nb_wkflow <- workflow() %>%
  add_model(nb_mod) %>%
  add_recipe(titanic_recipe)

nb_fit <- fit(nb_wkflow, titanic_train)
```

```
## Warning: Interaction specification failed for: ~sex:fare + age:fare. No
## interactions will be created.
```

Question 9

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

```
## # A tibble: 712 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1     0.234    0.766
##  2     0.234    0.766
##  3     0.234    0.766
##  4     0.646    0.354
##  5     0.234    0.766
##  6     0.234    0.766
##  7     0.234    0.766
##  8     0.234    0.766
##  9     0.234    0.766
## 10     0.472    0.528
## # ... with 702 more rows
```

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

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.688
```

```r
predict(lda_fit, new_data = titanic_train, type = "prob")
```

```
## # A tibble: 712 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1     0.219    0.781
##  2     0.219    0.781
##  3     0.219    0.781
##  4     0.673    0.327
##  5     0.219    0.781
```

```
## 6      0.219    0.781
## 7      0.219    0.781
## 8      0.219    0.781
## 9      0.219    0.781
## 10     0.471    0.529
## # ... with 702 more rows
```

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

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.688
```

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

```
## # A tibble: 712 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1     0.169    0.831
##  2     0.169    0.831
##  3     0.169    0.831
##  4     0.816    0.184
##  5     0.169    0.831
##  6     0.169    0.831
##  7     0.169    0.831
##  8     0.169    0.831
##  9     0.169    0.831
## 10     0.510    0.490
## # ... with 702 more rows
```

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

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.677
```

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

```
## # A tibble: 712 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1     0.191    0.809
##  2     0.191    0.809
##  3     0.191    0.809
##  4     0.536    0.464
```

```
## 5      0.191    0.809
## 6      0.191    0.809
## 7      0.191    0.809
## 8      0.191    0.809
## 9      0.191    0.809
## 10     0.748    0.252
## # ... with 702 more rows
```

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

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.677
```

Log fit and LDA fit had the highest accuracy on the training data.

Question 10

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

```
## # A tibble: 179 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1     0.234    0.766
##  2     0.234    0.766
##  3     0.646    0.354
##  4     0.234    0.766
##  5     0.472    0.528
##  6     0.472    0.528
##  7     0.234    0.766
##  8     0.234    0.766
##  9     0.234    0.766
## 10     0.234    0.766
## # ... with 169 more rows
```

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

```
##           Truth
## Prediction Yes No
##        Yes  23 18
##        No   46 92
```

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

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.642
```