Problem Set 6

Classification Part 1

Alex Tomack

Due Date: 2023-03-24

Getting Set Up

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...`. Accept defaults and save this file as `[LAST NAME]_ps6.Rmd` to your `code` folder.

Copy and paste the contents of this file into your `[LAST NAME]_ps6.Rmd` file. Then change the `author: [YOUR NAME]` (line 4) to your name.

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in.

This problem set is worth 10 total points, plus three extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must both have the correct code **and include a comment describing what each line does**. In addition, some questions ask you to provide a written response in addition to the code. Furthermore, some of the code chunks are totally empty, requiring you to try writing the code from scratch. Make sure to comment each line, explaining what it is doing!

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates…you name it. However, the final submission must be complete by you. There are no group assignments. To submit, compiled the completed problem set and upload the PDF file to Brightspace by midnight on 2023/03/24.

Good luck!

Question 0

Require `tidyverse` and `tidymodels` (for calculating AUC), and load the `admit_data.rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/7_Classification/data/admit_data.rds?raw=true') data to an object called `ad` . (Tip: use the `read_rds()` function with the link to the raw data.)

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.0     ✔ purrr   1.0.0
## ✔ tibble  3.2.0     ✔ dplyr   1.1.0
## ✔ tidyr   1.2.1     ✔ stringr 1.5.0
## ✔ readr   2.1.3     ✔ forcats 0.5.2
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
ad<-read_rds("https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/7_Classification/data/
admit_data.rds?raw=true")
```

Question 1 [2 points + 1 EC]

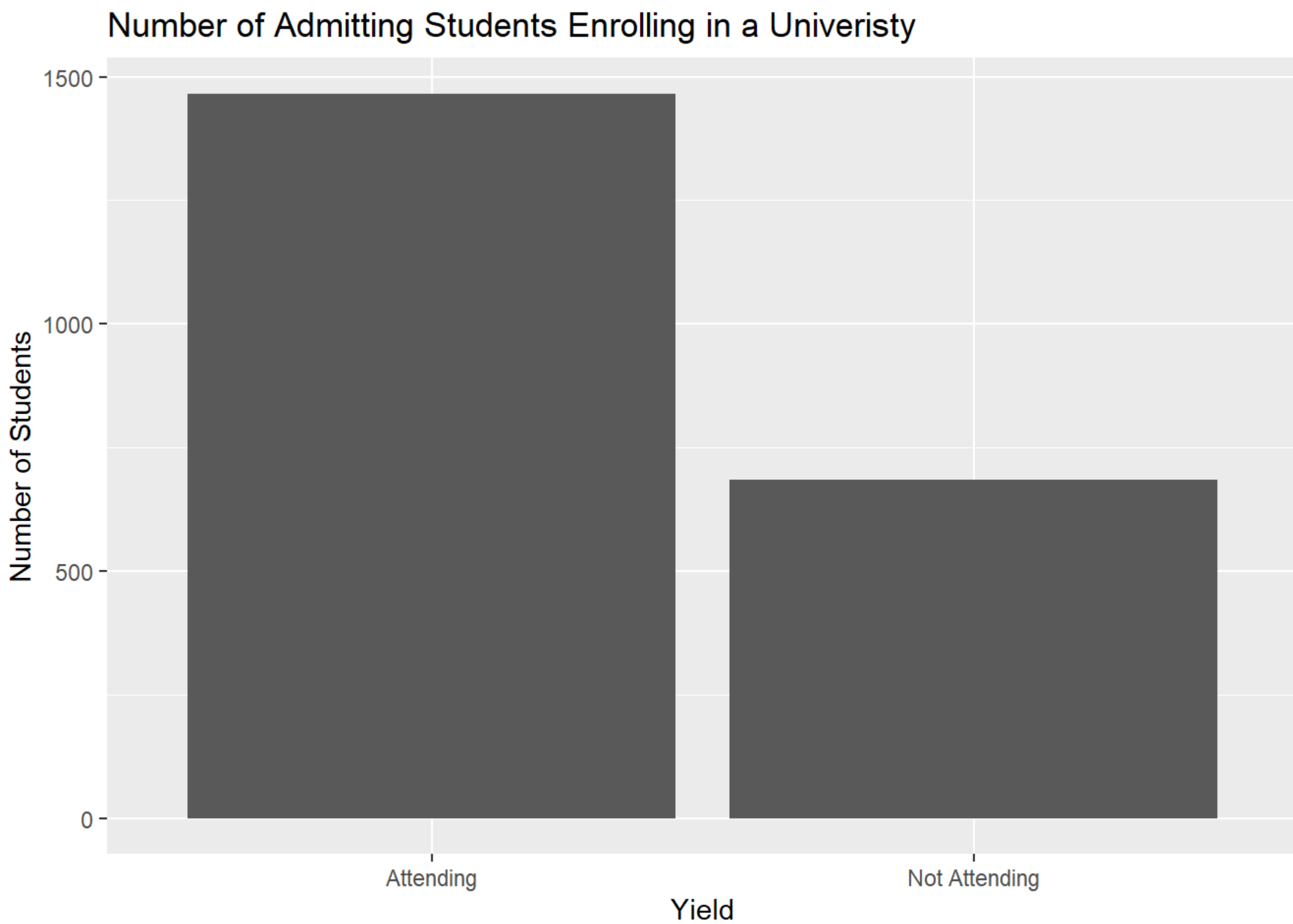
Plot the univariate visualizations for `yield` , `income` , and `sat` . Justify your choices for how you are visualizing these variables. Then plot the conditional variation between `yield` and `income` , and `yield` and `sat` . Again, justify your choices and then interpret the results. Do these variables matter for `yield` ?

EXTRA CREDIT (+1 point): Explain the pattern you observe in the univariate visualization of the SAT scores. What might explain this?

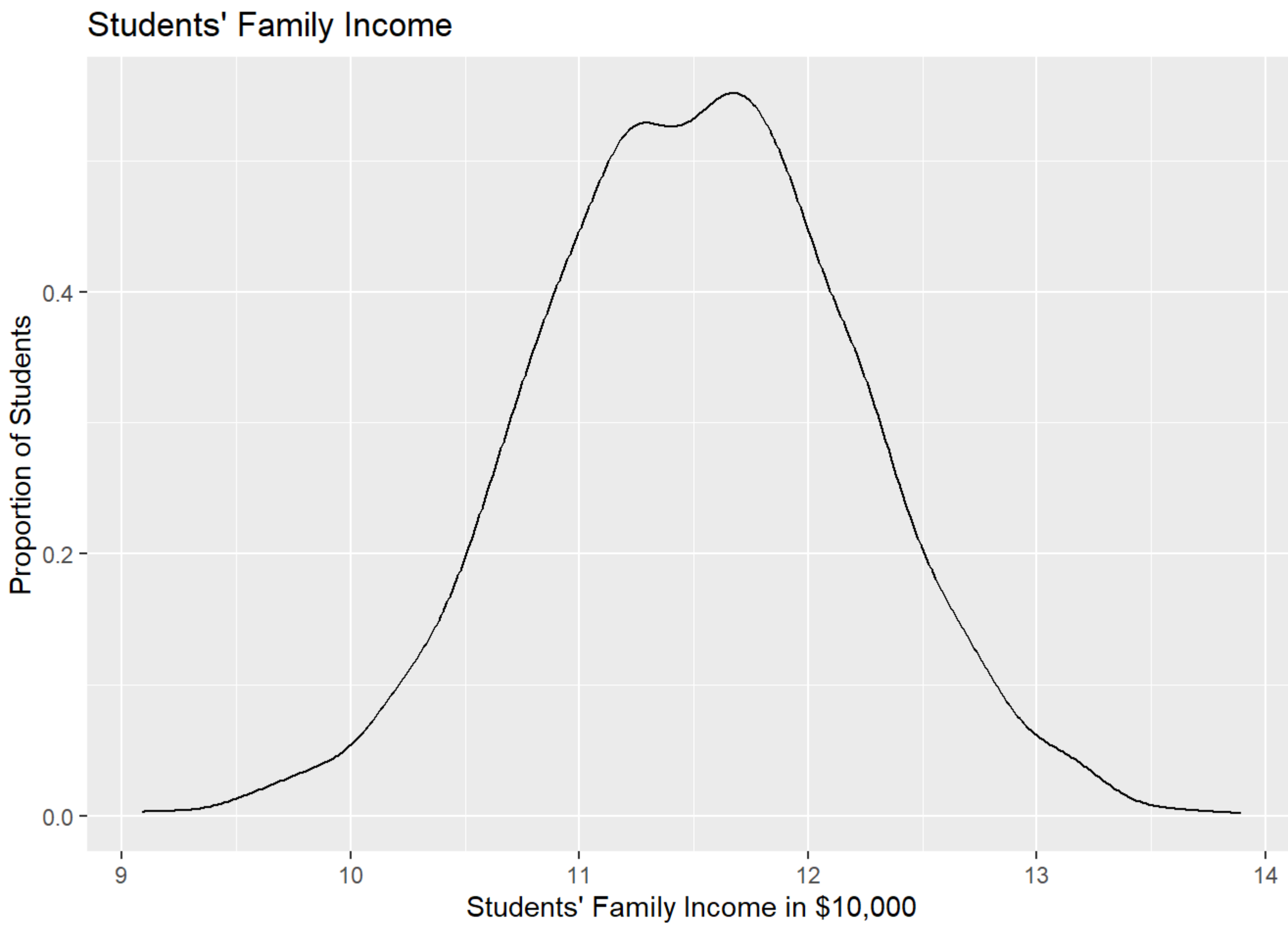
```
glimpse(ad) # yield is cat, income is cont, sat is cont
```

```
## Rows: 2,150
## Columns: 14
## $ ID          <chr> "0001", "0002", "0003", "0004", "0005", "0006", "0007", "0…
## $ income      <dbl> 289720.59, 176763.29, 81204.02, 93320.52, 144991.22, 72720…
## $ sat         <dbl> 1107.403, 1387.607, 1000.000, 1134.883, 1202.686, 1053.033…
## $ gpa         <dbl> 3.597153, 4.000000, 3.072323, 3.682776, 3.970005, 3.474787…
## $ visit       <dbl> 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0…
## $ legacy      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1…
## $ registered  <dbl> 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1…
## $ sent_scores <dbl> 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ distance    <dbl> 10.23279, 89.75984, 152.29961, 317.50274, 240.30712, 63.07…
## $ tuition     <dbl> 45000, 45000, 45000, 45000, 45000, 45000, 45000, 45000, 45…
## $ need_aid    <dbl> 0.000, 0.000, 8488.293, 3338.779, 0.000, 12093.802, 15156.…
## $ merit_aid   <dbl> 0.00, 35190.18, 0.00, 0.00, 30567.16, 0.00, 31633.66, 3219…
## $ net_price   <dbl> 45000.000, 9809.815, 36511.707, 41661.221, 14432.840, 3290…
## $ yield       <int> 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0…
```

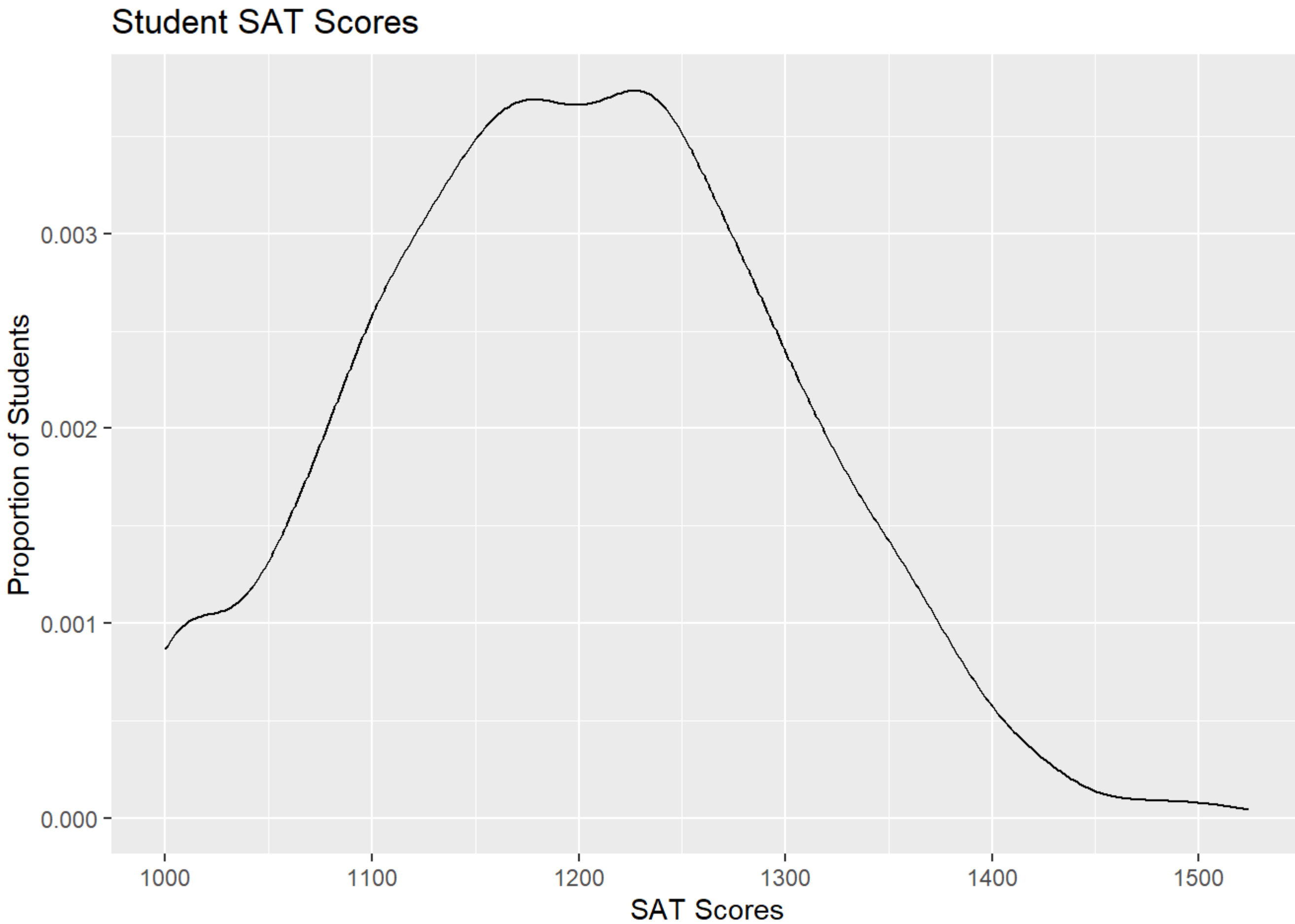
```
# univariate analysis
ad%>%
  mutate(yield_clean=ifelse(yield==0, "Not Attending", "Attending"))%>%
  ggplot(aes(x=yield_clean))+
  geom_bar()+
  labs(x="Yield",
       y="Number of Students",
       title="Number of Admitting Students Enrolling in a Univeristy")
```




```
ad%>%
  ggplot(aes(x=log(income)))+
  geom_density()+
  labs(x="Students' Family Income in $10,000",
       y="Proportion of Students",
       title="Students' Family Income")
```




```
ad%>%
  ggplot(aes(x=sat))+
  geom_density()+
  labs(x="SAT Scores",
       y="Proportion of Students",
       title="Student SAT Scores")
```




```
# Conditional Variation
```

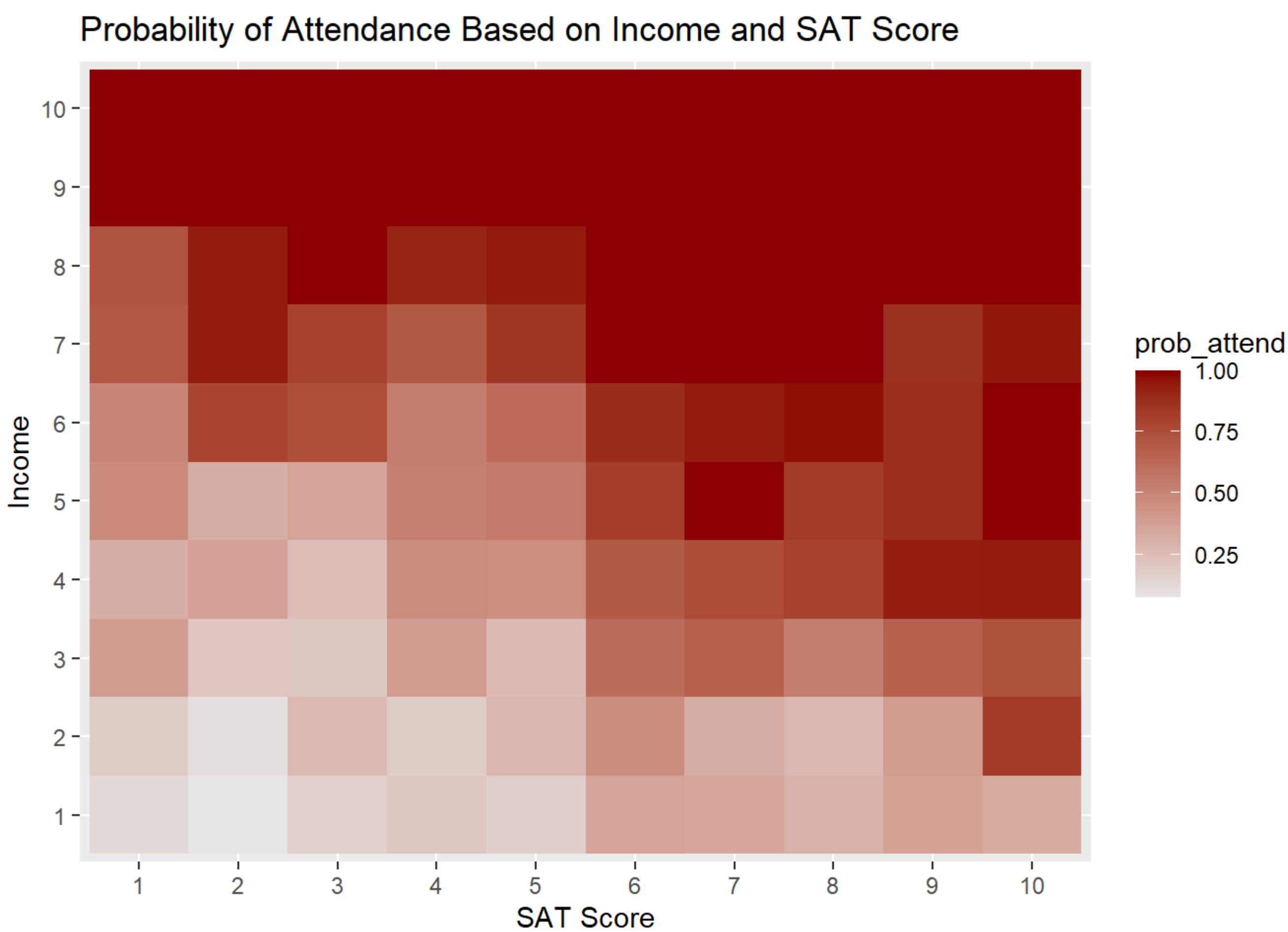
> Yield is a binary variable, so I chose a categorical univariate visualization. I did a quick mutate on it so that our graph would look a bit better, swapping out the numerical scaling on the x axis for two simple “attending” and “not attending” labels representative of the 1 and 0 respectively. Income is continuous so I used a density plot– had to log the variable. SAT is also a continuous variable, so I used another density plot.

Question 2 [2 points]

Look at these same conditional relationships between `yield` and `income` and `sat` , except divide the continuous measures of `income` and `sat` into deciles using the `ntile()` function, and create a single heatmap for all three variables, where the deciles of `income` and `sat` are on the axes, and the tiles are shaded by the average attendance in each cell. Which students are most likely to attend? Which are least likely to attend? Can you determine whether income or SAT scores matter more for attendance based on this plot?

```
ad %>%
  mutate(sat_dec=ntile(sat, n=10), # calculate deciles for income using the ntile() function
         income_dec=ntile(income, n=10)) %>% # calculate deciles for SAT scores using the ntile
() function
  group_by(sat_dec, income_dec) %>% # group_by these two decile variables for income and SAT sco
res
  summarise(prob_attend=mean(yield)) %>% # calculate the probability of attending
  ggplot(aes(x = factor(sat_dec),y = factor(income_dec) ,fill =prob_attend)) + # Put the deciles
on the axes as factors, and fill by the probability of attending
  geom_tile() +
  scale_fill_gradient(low="grey90", high="darkred") + # OPTIONAL: tweak the aesthetics
  labs(title = 'Probability of Attendance Based on Income and SAT Score', # Make sure to add hel
pful labels!
       x = 'SAT Score',
       y = 'Income')
```

```
## `summarise()` has grouped output by 'sat_dec'. You can override using the
## `.groups` argument.
```



- Students with higher income and SAT are most likely to attend, with a student's level of income being a better predictor of their attendance than their SAT score.

Question 3 [2 points]

Now start with the simplest way of predicting attendance: the conditional mean. As above, calculate deciles for `income` and `sat` called `incomeDec` and `satDec` using the `ntile()` function. Then calculate the average attendance in each cell using `group_by()` and `mutate()`, and finally predict attendance as 1 if the average is greater than 0.5, and 0 otherwise, using an `ifelse()` function. Evaluate the performance in terms of **accuracy**, **sensitivity**, and **specificity**, making sure to clearly define each metric.

```
ad <- ad %>%
 mutate(incomeDec=ntile(income, n=10), # calculate deciles for income using the ntile() function
         satDec=ntile(sat, n=10)) %>% # calculate deciles for SAT scores using the ntile() funct
ion
  group_by(satDec, incomeDec) %>% # group_by these two decile variables for income and SAT score
s
  mutate(prob_attend=mean(yield)) %>% # calculate the probability of attending
  mutate(pred_attend=ifelse(prob_attend>.5,1,0)) %>% # calculate the predicted attendance for ea
ch student using 0.5 as the threshold
  ungroup()

ad %>%
  group_by(yield) %>% # Calculate total attendees and non-attendees
  mutate(total_attend=n()) %>% # Calculate total attendees and non-attendees
  group_by(yield, pred_attend, total_attend) %>% # Calculate number of students falling into all
four groups (pred attend & attend, pred attend & not attend, pred not & attend, pred not & not a
ttend)
  summarise(nStudents=n(), .groups="drop") %>%
  mutate(proportion=nStudents/total_attend) %>% # calculate the proportion of students in each g
roup
  ungroup() %>% # ALWAYS UNGROUP
  mutate(spec=ifelse(yield==0 & pred_attend==0, proportion, NA),
         sens=ifelse(yield==1 & pred_attend==1, proportion, NA),
         accuracy=(535+1255)/2150,
         guess=mean(yield)) # OPTIONAL: calculate overall accuracy within chunk (could just do i
t manually too)
```

```
## # A tibble: 4 × 9
##   yield pred_attend total_attend nStudents proport…¹   spec   sens accur…² guess
##   <int>       <dbl>        <int>     <int>     <dbl>  <dbl>  <dbl>   <dbl> <dbl>
## 1     0           0          684       535     0.782  0.782 NA       0.833   0.5
## 2     0           1          684       149     0.218 NA     NA       0.833   0.5
## 3     1           0         1466       211     0.144 NA     NA       0.833   0.5
## 4     1           1         1466      1255     0.856 NA      0.856   0.833   0.5
## # … with abbreviated variable names ¹proportion, ²accuracy
```

```
  #filter(spec, sens) # OPTIONAL: filter to only look at sensitivity and specificity
```

> - We have a pretty decent sensitivity and specificity– accurately predicting 85% of students who attend and a 78% of those who dont. The proportions of students which we correctly predicted are significantly greater than the proportions we should expect by guessing all or no students attend (each 50%). Our model accurately predicts 83% of student choices.

Question 4 [2 points]

Now predict whether students will attend using a linear regression model (using the `lm()` function) that predicts `yield` as a function of `income` and `sat` (**not** using deciles, just the continuous versions). Calculate **accuracy**, **sensitivity**, and **specificity** from this model where the threshold is again 0.5, and compare to the results from Question 3. Does this model do better?

```
m1 <- lm(formula=yield~income+sat, data=ad) # Estimate linear regression model

ad %>%
  mutate(pred_attend=ifelse(predict(m1)>.5,1,0)) %>% # Calculate probability of attending based on the predicted regression result
  group_by(yield) %>% # Calculate total attendees and non-attendees
  mutate(total_attend=n()) %>% # Calculate total attendees and non-attendees
  group_by(yield,pred_attend, total_attend) %>% # Calculate number of students falling into all four groups (pred attend & attend, pred attend & not attend, pred not & attend, pred not & not attend)
  summarise(nStudents=n(),.groups="drop") %>%
  mutate(proportion=nStudents/total_attend) %>% # calculate the proportion of students in each group
  ungroup() %>% # ALWAYS UNGROUP
  mutate(sensitivity=ifelse(yield==1&pred_attend==1, proportion,NA),
         specificity=ifelse(yield==0&pred_attend==0, proportion,NA),
         accuracy=(366+1345)/2150)%>% # OPTIONAL: calculate overall accuracy within chunk (could just do it manually too)
  select(sensitivity, specificity, accuracy) # OPTIONAL: filter to only look at sensitivity and specificity
```

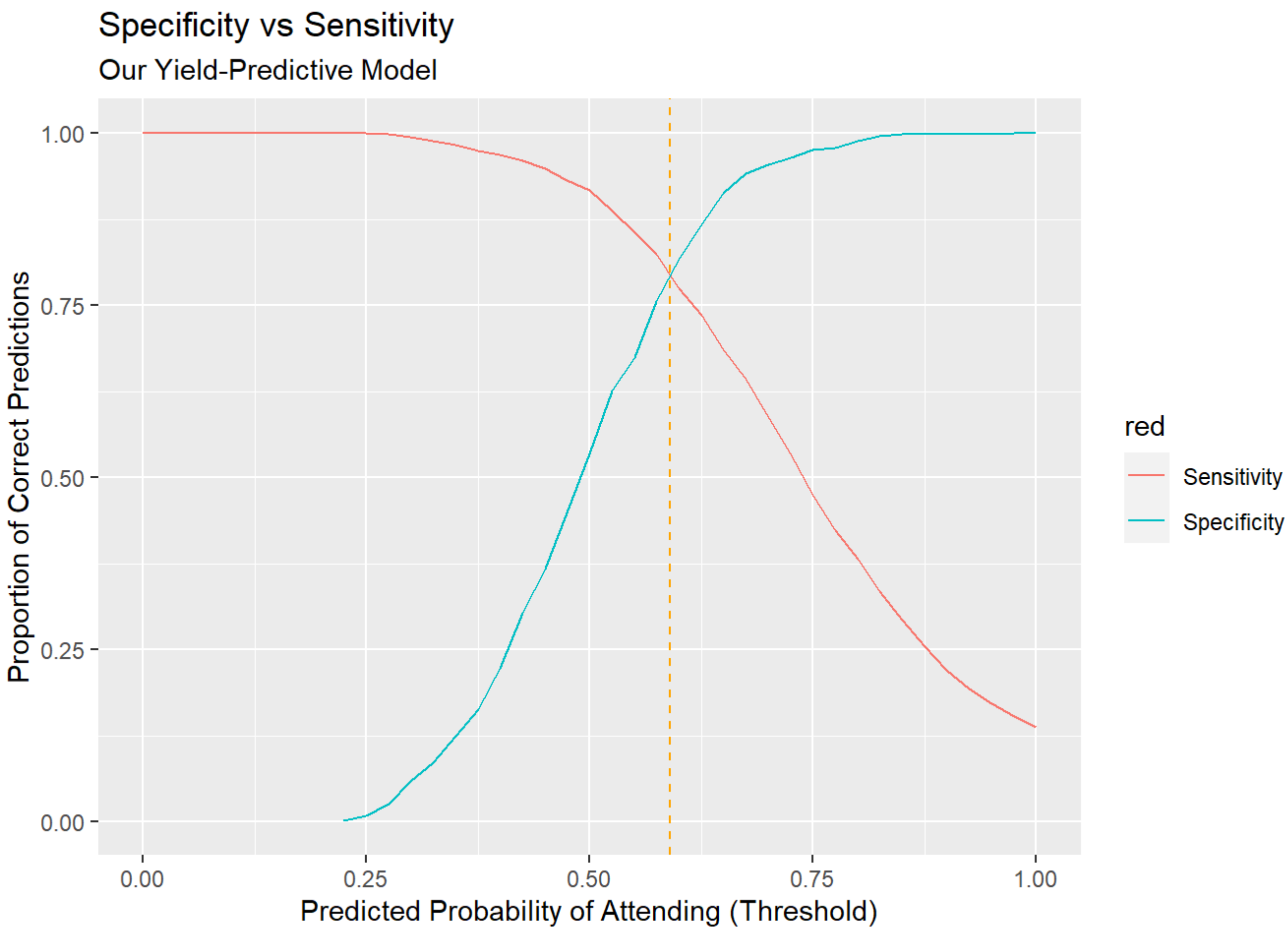
```
## # A tibble: 4 × 3
##   sensitivity specificity accuracy
##         <dbl>       <dbl>    <dbl>
## 1      NA           0.535    0.796
## 2      NA          NA        0.796
## 3      NA          NA        0.796
## 4       0.917      NA        0.796
```

- The model that uses linear regression performs slightly worse than the model from before. We lose ~4 percentage points of accuracy, gaining 6 percentage points in sensitivity but losing a substantial portion of specificity.

Question 5 [2 points]

Now recalculate **sensitivity**, **specificity**, and **accuracy** using different thresholds, ranging from 0 to 1, incrementing by 0.025 (use the `seq(from,to,by)` function). Plot the relationship between these thresholds and both the sensitivity and the specificity. What is the optimal threshold to balance the trade-off between **sensitivity** and **specificity**? Then plot ROC Curve and calculate the AUC.

```
threshRes <- NULL
for(thresh in seq(0,1, by=.025)) { # Loop over thresholds incrementing froo 0 to 1 by 0.025
  tmp <- ad %>%
  mutate(pred_attend=ifelse(predict(m1)>thresh,1,0)) %>% # Calculate probability of attending ba
sed on the predicted regression result
  group_by(yield) %>% # Calculate total attendees and non-attendees
  mutate(total_attend=n()) %>% # Calculate total attendees and non-attendees
  group_by(yield, pred_attend, total_attend) %>% # Calculate number of students falling into all
four groups (pred attend & attend, pred attend & not attend, pred not & attend, pred not & not a
ttend)
  summarise(nStudents=n(),.groups="drop") %>%
  mutate(proportion=nStudents/total_attend) %>% # calculate the proportion of students in each g
roup
  ungroup() %>% # ALWAYS UNGROUP
  mutate(threshold=thresh) # Save the threshold value

  threshRes <- threshRes %>%
    bind_rows(tmp)
}

# Plot relationship between threshold and sens/spec
threshRes %>%
  mutate(metric = ifelse(yield==0&pred_attend==0, "Specificity",
                         ifelse(yield==1&pred_attend==1, "Sensitivity",NA))) %>% # Use two neste
d ifelse() to capture sensitivity and specificity, make NA otherwise
  drop_na(metric) %>% # Drop the rows that are neither sensitivity nor specificity
  ggplot(aes(x=threshold, y=proportion, color=metric)) + # Plot the threshold values on the x-ax
is, the proportions on the y-axis, and color by metric
  geom_line() +
  labs(title = 'Specificity vs Sensitivity', # Always include clear labels!
       subtitle = 'Our Yield-Predictive Model',
       x = 'Predicted Probability of Attending (Threshold)',
       y = 'Proportion of Correct Predictions',
       color = 'red')+
  geom_vline(xintercept=.59, color="orange", linetype="dashed")
```




```
# Plot ROC CurvethreshRes %>%
  mutate(metric = ifelse(yield==0&pred_attend==0, "Specificity",
                         ifelse(yield==1&pred_attend==1, "Sensitivity", NA))) %>% # Use two nest
ed ifelse() to capture sensitivity and specificity, make NA otherwise
  drop_na(metric) %>% # Drop the rows that are neither sensitivity nor specificity
  select(proportion, metric, threshold) %>% # Select only the proportion, the metric, and the th
reshold values
  spread(key=metric, value=proportion) %>% # Create two new columns of proportions, one for Spec
ificity and the other for Sensitivity
  ggplot(aes(x=1-Specificity, y=Sensitivity)) + # Plot Sensitivity on the y-axis and 1-Specifici
ty on the x-axis
  geom_line() +
  xlim(0,1) + # Force axes to be between 0 and 1
  ylim(0,1) + # Force axes to be between 0 and 1
  geom_abline(slope=1, intercept=0, linetype="dashed") + # Add a dotted diagonal line for refere
nce
  labs(title = 'ROC Curve', # Always include clear labels!
       subtitle = 'Tradeoff Between Sensitivity and Specificity',
       x = '1-Sensitivity',
       y = 'Specificity')
```

```
## Error in ifelse(yield == 0 & pred_attend == 0, "Specificity", ifelse(yield == : object 'yiel
d' not found
```

```
# Calculate AUC
require(tidymodels) # Require tidymodels if you haven't already
```

```
## Loading required package: tidymodels
```

```
## Warning: package 'tidymodels' was built under R version 4.2.3
```

```
## ── Attaching packages ────────────────────────────────────── tidymodels 1.0.0 ──
```

```
## ✓ broom        1.0.2     ✓ rsample      1.1.1
## ✓ dials        1.1.0     ✓ tune         1.0.1
## ✓ infer        1.0.4     ✓ workflows    1.1.3
## ✓ modeldata    1.1.0     ✓ workflowsets 1.0.0
## ✓ parsnip      1.0.4     ✓ yardstick    1.1.0
## ✓ recipes      1.0.5
```

```
## Warning: package 'dials' was built under R version 4.2.3
```

```
## Warning: package 'infer' was built under R version 4.2.3
```

```
## Warning: package 'modeldata' was built under R version 4.2.3
```

```
## Warning: package 'parsnip' was built under R version 4.2.3
```

```
## Warning: package 'recipes' was built under R version 4.2.3
```

```
## Warning: package 'rsample' was built under R version 4.2.3
```

```
## Warning: package 'tune' was built under R version 4.2.3
```

```
## Warning: package 'workflows' was built under R version 4.2.3
```

```
## Warning: package 'workflowsets' was built under R version 4.2.3
```

```
## Warning: package 'yardstick' was built under R version 4.2.3
```

```
## ── Conflicts ──────────────────────────────────────── tidymodels_conflicts() ──
## ✖ scales::discard() masks purrr::discard()
## ✖ dplyr::filter()   masks stats::filter()
## ✖ recipes::fixed()  masks stringr::fixed()
## ✖ dplyr::lag()      masks stats::lag()
## ✖ yardstick::spec() masks readr::spec()
## ✖ recipes::step()   masks stats::step()
## • Search for functions across packages at https://www.tidymodels.org/find/
```

```
forAUC <- ad %>%
  mutate(pred_yield=predict(m1), type="response") %>% # Calculate the probability of attending f
rom the model predictions (just reuse the main model calculated in Q4)
  mutate(yield2=factor(yield, levels=c("1", "0"))) # Convert the outcome to a factor with levels
of c('1','0')!

roc_auc(forAUC, yield2, pred_yield) # Calculate the AUC
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.875
```

> - The optimal threshold is 0.59– this allows us to maximize both sensitivity and specificity while eliminating the gap between them. OUr model is pretty good despite using a linear regression on a binary variable. We're accurately predicting 87% of positive and negative cases, which is solid B lettergrade in performance.

Question 6 [2 EXTRA CREDIT points]

Re-do questions 4 and 5 using a logistic regression. Does this perform better than a linear regression model?

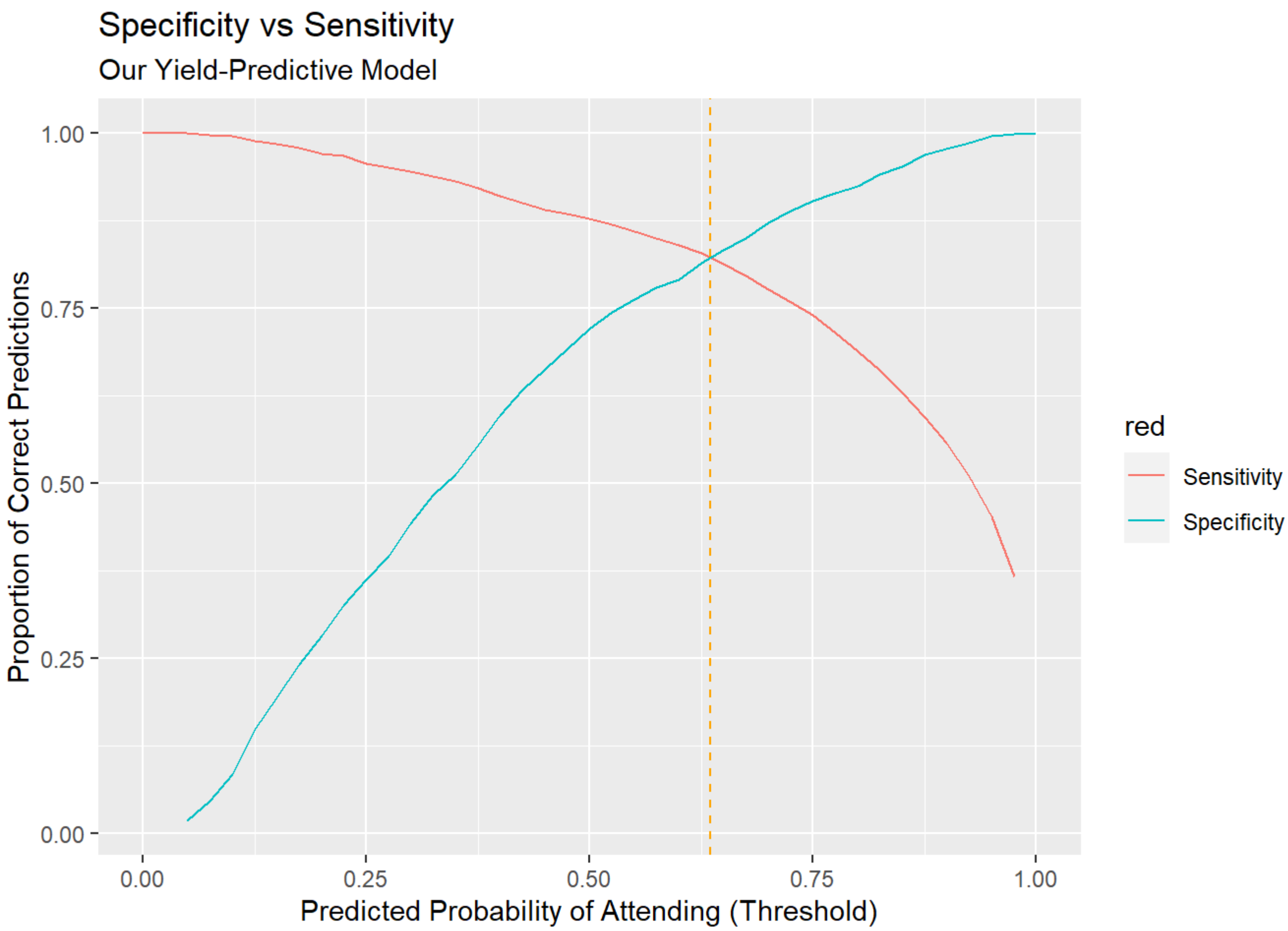
```
# INSERT CODE HERE. (If you completed 4 and 5, you can just copy the code and modify the linear
regression model and the predict() functions)

g1 <- glm(formula=yield~income+sat, data=ad, family=binomial(link="logit")) # Estimate linear re
gression model
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
ad %>%
  mutate(pred_attend=ifelse(predict(g1, type="response")>.5,1,0)) %>% # Calculate probability of
attending based on the predicted regression result
  group_by(yield) %>% # Calculate total attendees and non-attendees
  mutate(total_attend=n()) %>% # Calculate total attendees and non-attendees
  group_by(yield,pred_attend, total_attend) %>% # Calculate number of students falling into all
four groups (pred attend & attend, pred attend & not attend, pred not & attend, pred not & not a
ttend)
  summarise(nStudents=n(),.groups="drop") %>%
  mutate(proportion=nStudents/total_attend) %>% # calculate the proportion of students in each g
roup
  ungroup() %>% # ALWAYS UNGROUP
  mutate(sensitivity=ifelse(yield==1&pred_attend==1, proportion,NA),
         specificity=ifelse(yield==0&pred_attend==0, proportion,NA))
```

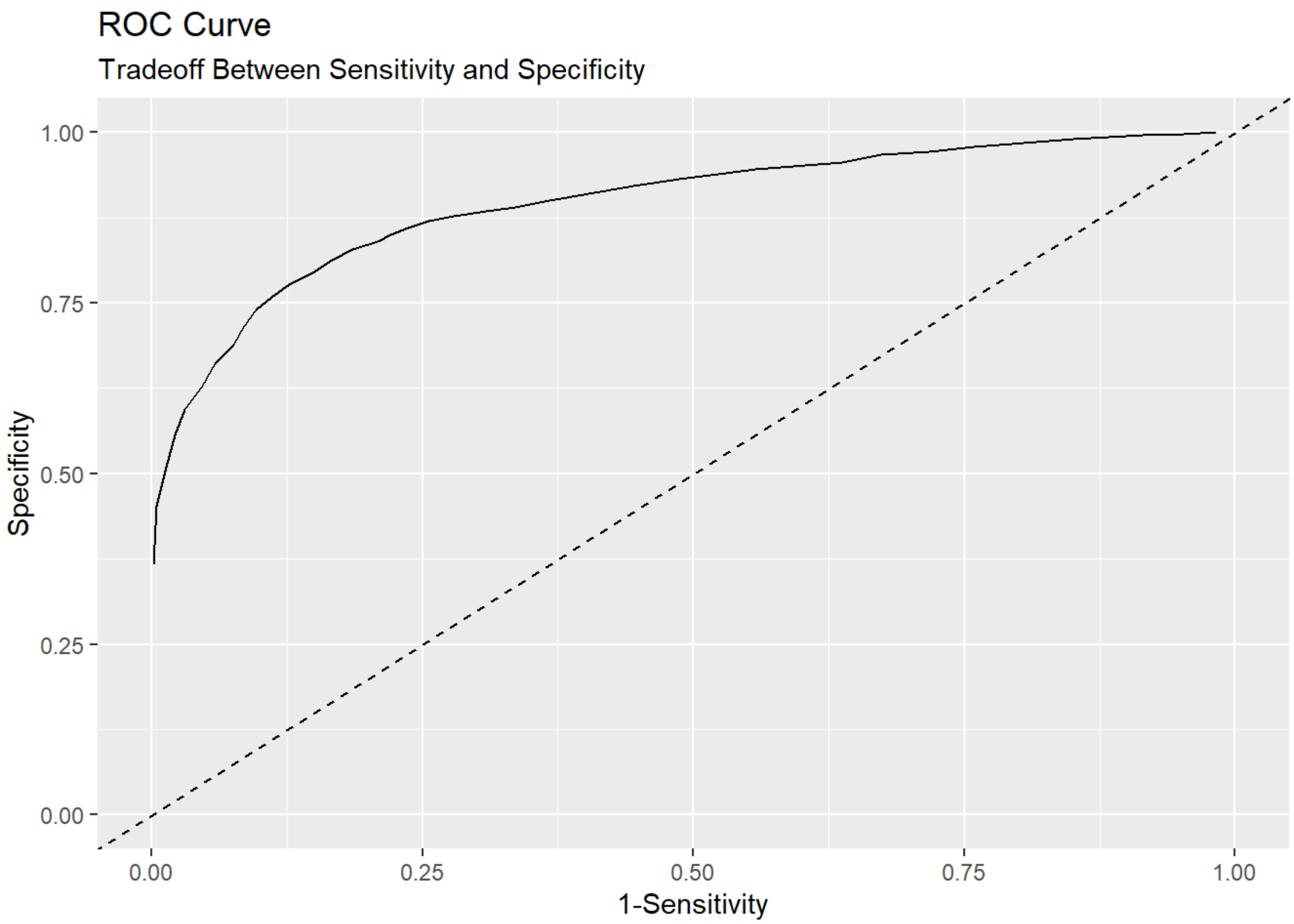
```
## # A tibble: 4 × 7
##   yield pred_attend total_attend nStudents proportion sensitivity specificity
##   <int>       <dbl>        <int>     <int>      <dbl>       <dbl>       <dbl>
## 1     0           0          684       493      0.721      NA           0.721
## 2     0           1          684       191      0.279      NA          NA
## 3     1           0         1466       179      0.122      NA          NA
## 4     1           1         1466      1287      0.878       0.878      NA
```

```
threshRes <- NULL
for(thresh in seq(0,1, by=.025)) { # Loop over thresholds incrementing froo 0 to 1 by 0.025
  tmp <- ad %>%
  mutate(pred_attend=ifelse(predict(g1, type="response")>thresh,1,0)) %>% # Calculate probabilit
y of attending based on the predicted regression result
  group_by(yield) %>% # Calculate total attendees and non-attendees
  mutate(total_attend=n()) %>% # Calculate total attendees and non-attendees
  group_by(yield, pred_attend, total_attend) %>% # Calculate number of students falling into all
four groups (pred attend & attend, pred attend & not attend, pred not & attend, pred not & not a
ttend)
  summarise(nStudents=n(),.groups="drop") %>%
  mutate(proportion=nStudents/total_attend) %>% # calculate the proportion of students in each g
roup
  ungroup() %>% # ALWAYS UNGROUP
  mutate(threshold=thresh) # Save the threshold value

  threshRes <- threshRes %>%
    bind_rows(tmp)
}

# Plot relationship between threshold and sens/spec
threshRes %>%
  mutate(metric = ifelse(yield==0&pred_attend==0, "Specificity",
                         ifelse(yield==1&pred_attend==1, "Sensitivity",NA))) %>% # Use two neste
d ifelse() to capture sensitivity and specificity, make NA otherwise
  drop_na(metric) %>% # Drop the rows that are neither sensitivity nor specificity
  ggplot(aes(x=threshold, y=proportion, color=metric)) + # Plot the threshold values on the x-ax
is, the proportions on the y-axis, and color by metric
  geom_line() +
  labs(title = 'Specificity vs Sensitivity', # Always include clear labels!
       subtitle = 'Our Yield-Predictive Model',
       x = 'Predicted Probability of Attending (Threshold)',
       y = 'Proportion of Correct Predictions',
       color = 'red')+
  geom_vline(xintercept=.635, color="orange", linetype="dashed")
```




```
# Plot ROC Curve
threshRes %>%
  mutate(metric = ifelse(yield==0&pred_attend==0, "Specificity",
                         ifelse(yield==1&pred_attend==1, "Sensitivity", NA))) %>% # Use two nest
ed ifelse() to capture sensitivity and specificity, make NA otherwise
  drop_na(metric) %>% # Drop the rows that are neither sensitivity nor specificity
  select(proportion, metric, threshold) %>% # Select only the proportion, the metric, and the th
reshold values
  spread(key=metric, value=proportion) %>% # Create two new columns of proportions, one for Spec
ificity and the other for Sensitivity
  ggplot(aes(x=1-Specificity, y=Sensitivity)) + # Plot Sensitivity on the y-axis and 1-Specifici
ty on the x-axis
  geom_line() +
  xlim(0,1) + # Force axes to be between 0 and 1
  ylim(0,1) + # Force axes to be between 0 and 1
  geom_abline(slope=1, intercept=0, linetype="dashed") + # Add a dotted diagonal line for refere
nce
  labs(title = 'ROC Curve', # Always include clear labels!
       subtitle = 'Tradeoff Between Sensitivity and Specificity',
       x = '1-Sensitivity',
       y = 'Specificity')
```

```
## Warning: Removed 3 rows containing missing values (`geom_line()`).
```




```
# Calculate AUC
require(tidymodels) # Require tidymodels if you haven't already
forAUC <- ad %>%
  mutate(pred_yield=predict(g1, type="response")) %>% # Calculate the probability of attending f
rom the model predictions (just reuse the main model calculated in Q4)
  mutate(yield2=factor(yield, levels=c("1", "0"))) # Convert the outcome to a factor with levels
of c('1','0')!

roc_auc(forAUC, yield2, pred_yield) # Calculate the AUC
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.897
```

- Our predictions, using a logistic model, are slightly improved from our linear model– allowing us to accurately predict close to 90% of positive and negative cases.