Classification

Part 2

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/03/22

Slides Updated: 2023-03-21

Agenda

1. Introducing **logit**
2. Running logit
3. Evaluating logit

Logit Regression

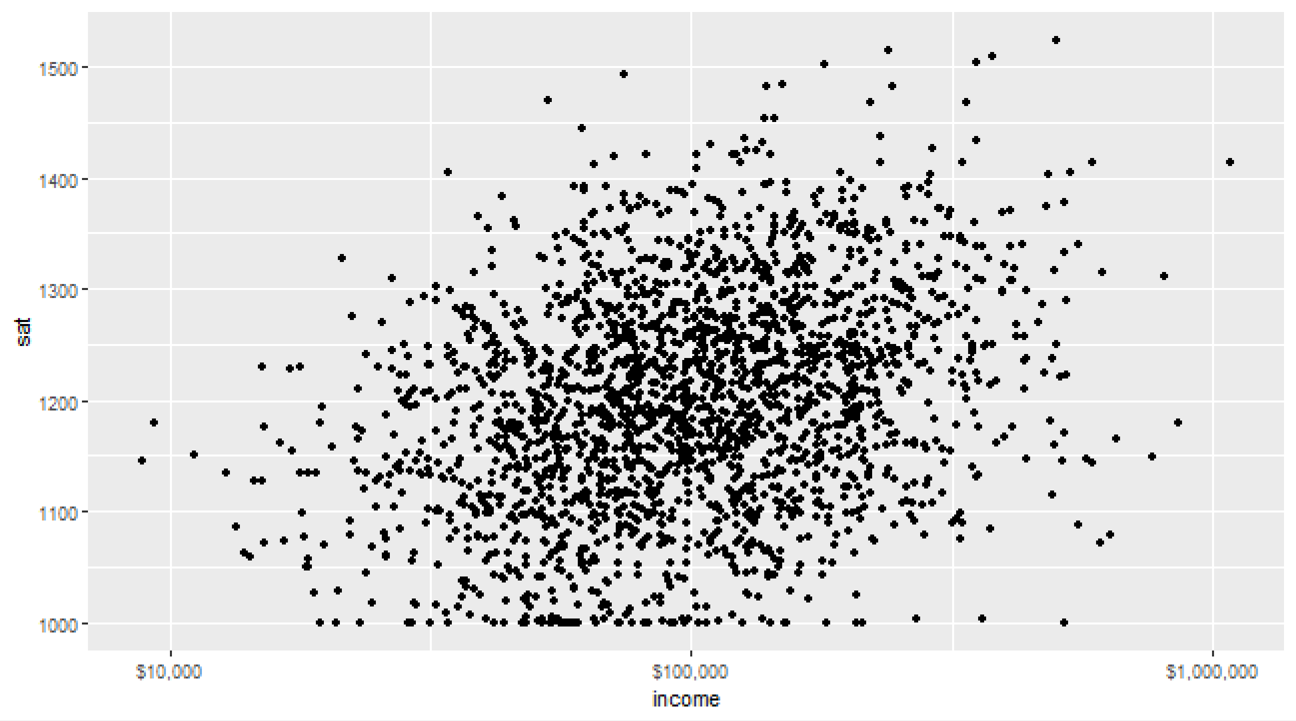
- A different **type** of **regression**
    - What do we mean by **type**?
- Let's take a step back

```
require(tidyverse)
require(scales)
ad <- readRDS('../data/admit_data.rds')
```

Regression Types

- "Linear" regression...why is it "linear"?

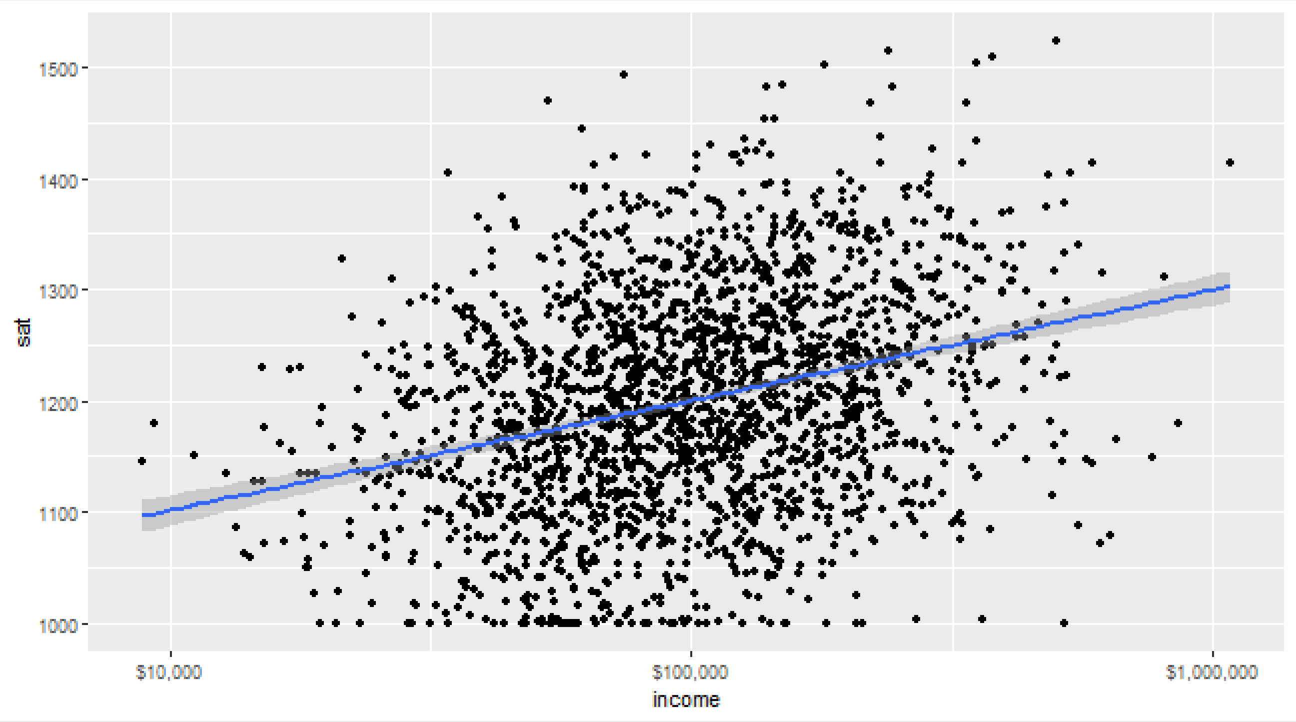
```
(p <- ad %>%
  ggplot(aes(x = income,y = sat)) +
  geom_point() + scale_x_log10(labels = dollar))
```



Regression Types

- "Linear" regression...why is it "linear"?
- Because you can summarize it with a line!

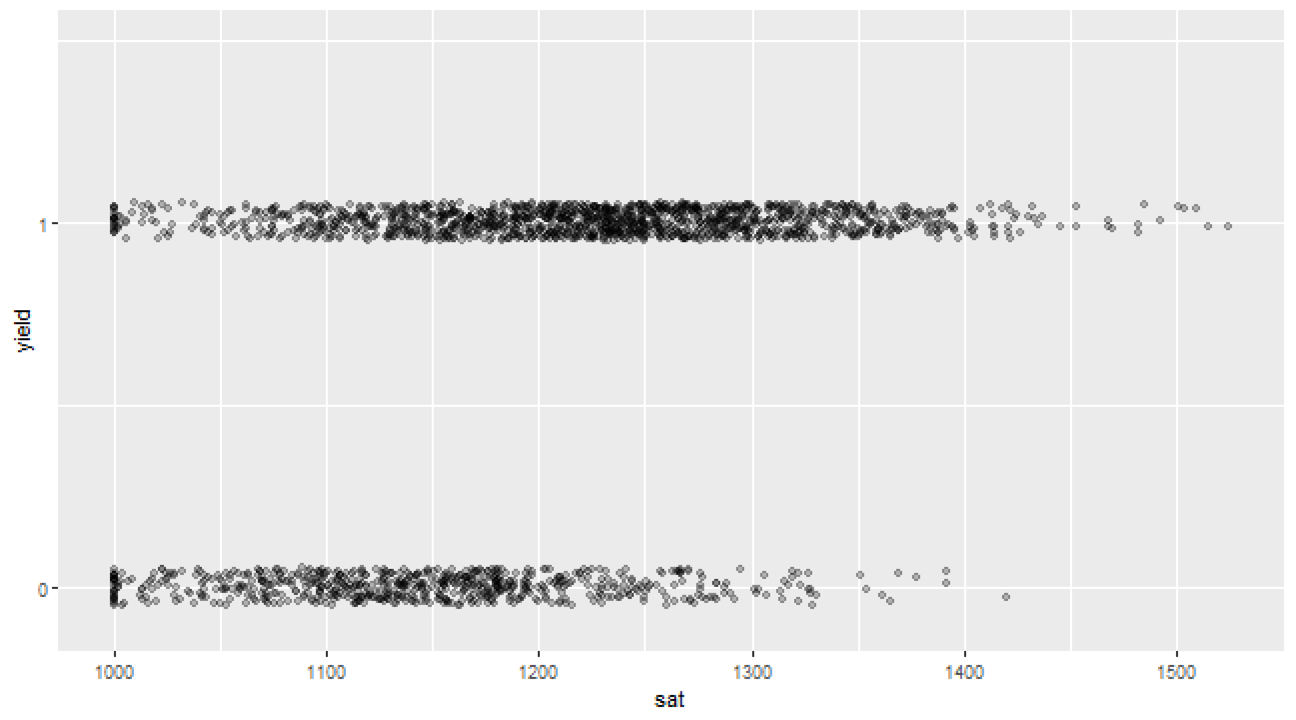
```
p + geom_smooth(method = 'lm')
```



Regression Types

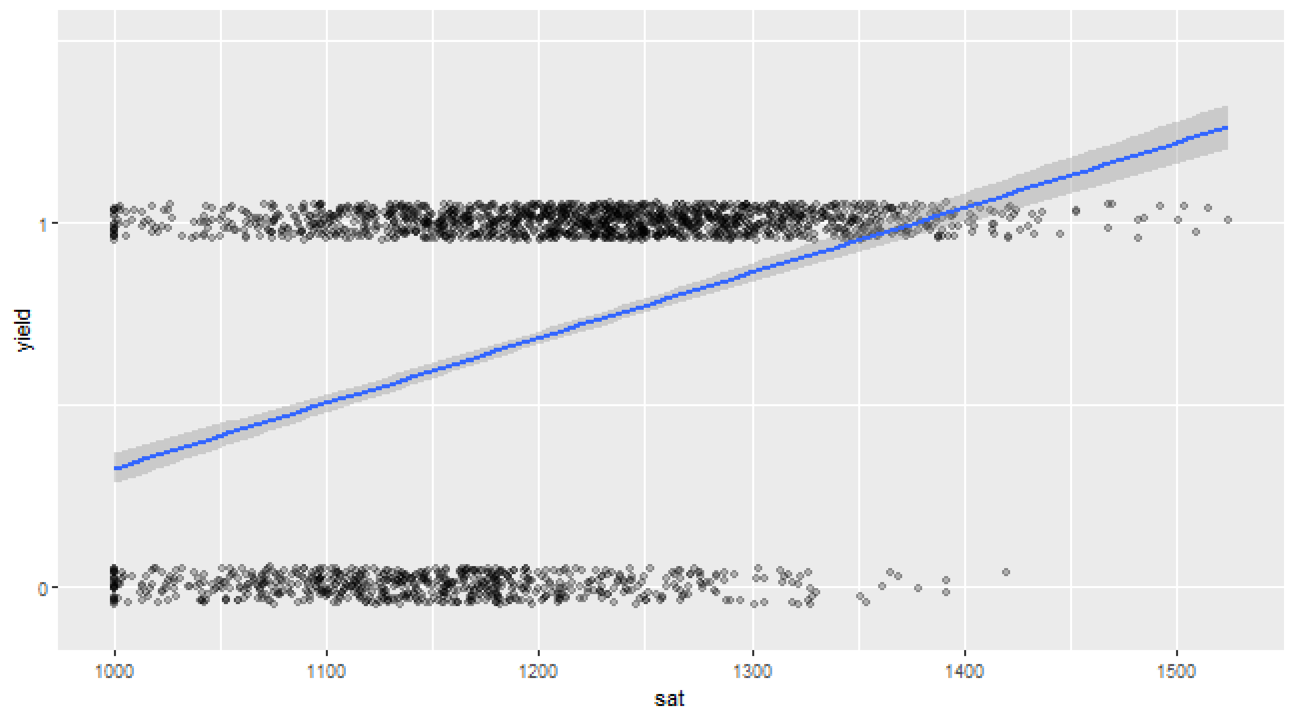
- But what if the outcome is binary?

```
(p <- ad %>% ggplot(aes(x = sat,y = yield)) +
   scale_y_continuous(breaks = c(0,1),limits = c(-.1,1.5)) +
  geom_jitter(width = .01,height = .05,alpha = .25))
```



Regression Types

- But what if the outcome is binary?
- Lines seem too clumsy
  - If `1` = attend, how can you go higher?



Logit

- Theory: binary outcomes are **proxies** for some **latent** measure
  - Binary outcome `yield`: either attend or not attend
  - Latent outcome `willingness`: continuous measure
- The higher your `willingness`, the more likely you are to attend
- Logit regression: model the `willingness`
  - What is `willingness` actually?
  - Probability of attending: $Pr(attend)$
- Part of a broader class of models called "generalized linear model" (GLM)

$$Pr(y = 1|x) = G(\alpha + \beta X)$$

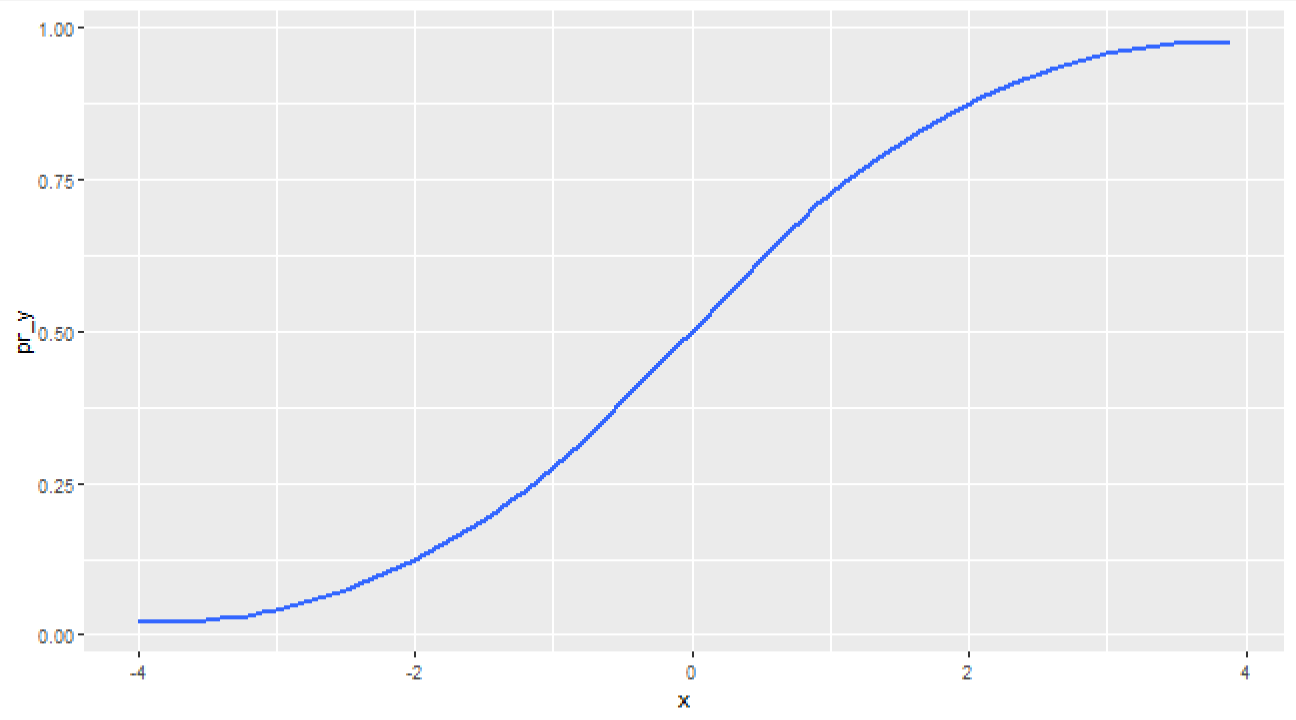
GLMs

- $Pr(y = 1|x) = G(\alpha + \beta X)$
- Does this look familiar?
- Linear regression: $Y = \alpha + \beta X$
    - Outcome: $Y \rightarrow Pr(y = 1|x)$
    - Mapping: $\alpha + \beta X \rightarrow G(\alpha + \beta X)$
- G is the "link function"
    - Transforms values of $\alpha + \beta X$ into **probabilities**
- Logistic function: specific type of link function

$$G(x) = \frac{1}{1 + exp(-x)}$$

Logistic Function

```
x <- runif(100,-4,4)
pr_y <- 1/(1 + exp(-x))
as_tibble(pr_y,x) %>%
  ggplot(aes(x = x,y = pr_y)) +
  geom_smooth()
```

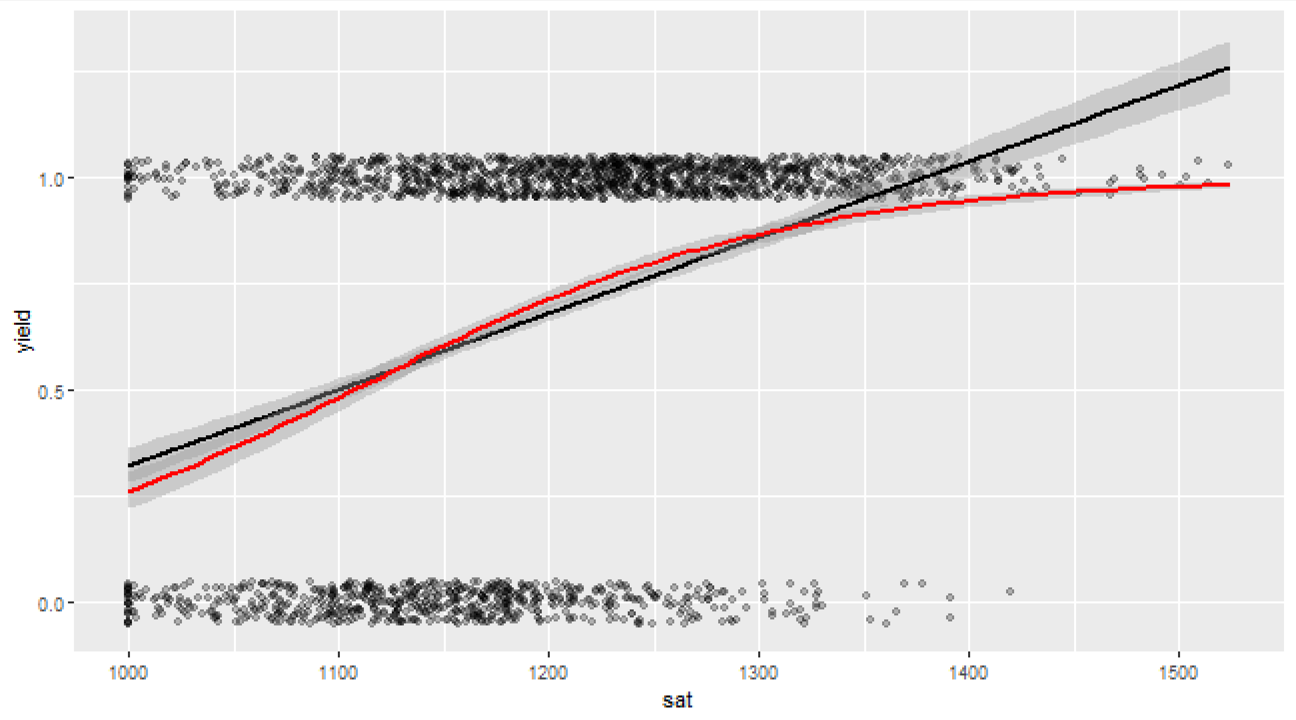


Logistic Function

- But what about real data like $\alpha + \beta X$?
- $G(X) = \frac{exp(\alpha+\beta X)}{1+exp(\alpha+\beta X)}$
- We estimate this with `glm(formula,data,family)`
    - Note similarity to `lm(formula,data)`
- `family = binomial(link = "logit")`

Logistic Regression (logit)

```
ad %>% ggplot(aes(x = sat,y = yield)) +
  geom_jitter(width = .01,height = .05,alpha = .25) +
  geom_smooth(method = 'lm',color = 'black') +
  geom_smooth(method = 'glm',color = 'red',
              method.args = list(family = binomial(link = 'logit')))
```



Logistic Regression (logit)

```
# Train model
mLogit <- glm(formula = yield ~ sat,data = ad,family = binomial(link
= 'logit'))

# Predict model
ad <- ad %>%
  mutate(prob_attend = predict(mLogit,type = 'response')) %>%
  mutate(pred_attend = ifelse(prob_attend > .5,1,0))

# Evaluate model
eval <- ad %>%
  group_by(yield) %>%
  mutate(total_attend = n()) %>%
  group_by(yield,pred_attend,total_attend) %>%
  summarise(nStudents=n(),.groups = 'drop') %>%
  mutate(prop = nStudents / total_attend) %>%
  ungroup() %>%
  mutate(accuracy = percent(sum((yield == pred_attend)*nStudents) /
sum(nStudents)))
```

Logistic Regression (logit)

```
eval
```

```
## # A tibble: 4 × 6
##   yield pred_attend total_attend nStudents  prop accuracy
##   <int>       <dbl>        <int>     <int> <dbl> <chr>
## 1     0           0          684       220 0.322 70%
## 2     0           1          684       464 0.678 70%
## 3     1           0         1466       173 0.118 70%
## 4     1           1         1466      1293 0.882 70%
```

Logistic Regression (logit)

- Can also calculate ROC Curve and AUC

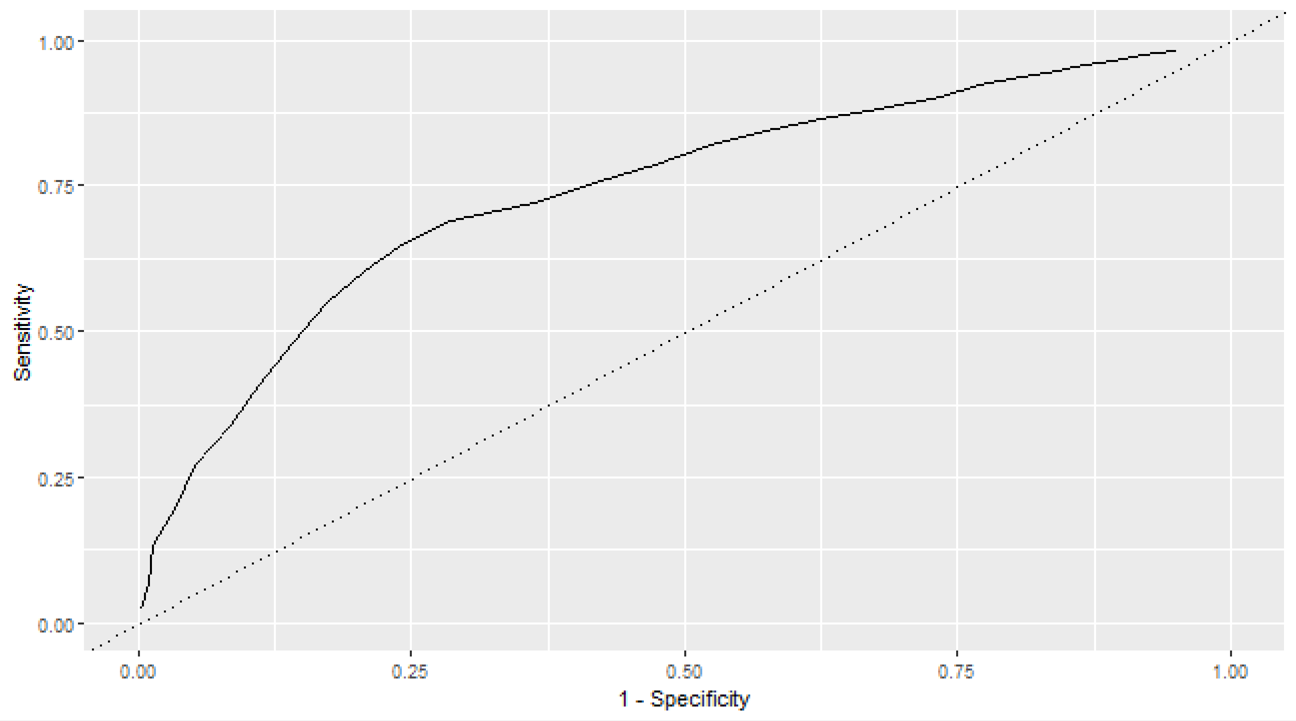
```
toplot <- NULL
for(thresh in seq(0,1,by = .025)) {
  toplot <- ad %>%
    mutate(pred_attend = ifelse(predict(mLogit,type = 'response') >
thresh,1,0)) %>%
    group_by(yield) %>%
    mutate(total_attend = n()) %>%
    group_by(yield,pred_attend,total_attend) %>%
    summarise(nStudents=n(),.groups = 'drop') %>%
    mutate(prop = nStudents / total_attend) %>%
    ungroup() %>%
    mutate(threshold = thresh) %>%
    bind_rows(toplot)
}
```

Logistic Regression (logit)

```
p <- toplot %>%
  mutate(metric = ifelse(yield == 1 & pred_attend == 1,'Sensitivity',
                         ifelse(yield == 0 & pred_attend ==
0,'Specificity',NA))) %>%
  drop_na(metric) %>%
  select(prop,metric,threshold) %>%
  spread(metric,prop) %>%
  ggplot(aes(x = 1-Specificity,y = Sensitivity)) +
  geom_line() +
  xlim(c(0,1)) + ylim(c(0,1)) +
  geom_abline(slope = 1,intercept = 0,linetype = 'dotted')
```

Logistic Regression (logit)

```
p
```



Logistic Regression (logit)

```r
require(tidymodels)
roc_auc(data = ad %>%
  mutate(prob_attend = predict(mLogit,type = 'response'),
         truth = factor(yield,levels = c('1','0'))) %>%
  select(truth,prob_attend),truth,prob_attend)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.742
```

Comparing Models

- Two big questions in prediction:
  1. Do I have the correct predictors X?
  2. Do I have the best model?
- Two types of outcomes (thus far)
  1. Continuous Y: use **RMSE**
  2. Binary Y: use **AUC**
- Let's determine the best model from the following:
  - X: (1) `sat + legacy` vs. (2) `sat + legacy + income`
  - Model: (1) conditional means vs. (2) `lm` vs. (3) `glm`

Comparing Models

- Conditional means - simple X

```r
results <- NULL

# Train & Predict
toEval <- ad %>%
  mutate(satDec = ntile(sat,n = 10)) %>%
  group_by(satDec,legacy) %>%
  mutate(prob_attend = mean(yield),
         truth = factor(yield,levels = c('1','0'))) %>%
    ungroup() %>%
    select(truth,prob_attend)

# Evaluate
results <- roc_auc(data = toEval,truth,prob_attend) %>%
  mutate(model = 'CM',
         predictors = 'Simple') %>%
  bind_rows(results)
```

Comparing Models

- Conditional means - complex X

```
# Train & Predict
toEval <- ad %>%
  mutate(satDec = ntile(sat,n = 10),
         incDec = ntile(income,n = 10)) %>%
  group_by(satDec,incDec,legacy) %>%
  mutate(prob_attend = mean(yield),
         truth = factor(yield,levels = c('1','0'))) %>%
    ungroup() %>%
    select(truth,prob_attend)

# Evaluate
results <- roc_auc(data = toEval,truth,prob_attend) %>%
  mutate(model = 'CM',
         predictors = 'Complex') %>%
  bind_rows(results)
```

Comparing Models

- Linear regression (`lm`) - simple X

```
# Train
m <- lm(yield ~ sat + legacy,ad)

# Predict
toEval <- ad %>%
  mutate(prob_attend = predict(m),
         truth = factor(yield,levels = c('1','0'))) %>%
    ungroup() %>%
    select(truth,prob_attend)

# Evaluate
results <- roc_auc(data = toEval,truth,prob_attend) %>%
  mutate(model = 'LM',
         predictors = 'Simple') %>%
  bind_rows(results)
```

Comparing Models

- Linear regression (`lm`) - complex X

```
# Train
m <- lm(yield ~ sat + legacy + income,ad)

# Predict
toEval <- ad %>%
  mutate(prob_attend = predict(m),
         truth = factor(yield,levels = c('1','0'))) %>%
    ungroup() %>%
    select(truth,prob_attend)

# Evaluate
results <- roc_auc(data = toEval,truth,prob_attend) %>%
  mutate(model = 'LM',
         predictors = 'Complex') %>%
  bind_rows(results)
```

Comparing Models

- Logit regression (`glm`) - simple X

```
# Train
m <- glm(yield ~ sat + legacy,ad,family = binomial(link = 'logit'))

# Predict
toEval <- ad %>%
  mutate(prob_attend = predict(m,type = 'response'),
         truth = factor(yield,levels = c('1','0'))) %>%
    ungroup() %>%
    select(truth,prob_attend)

# Evaluate
results <- roc_auc(data = toEval,truth,prob_attend) %>%
  mutate(model = 'GLM',
         predictors = 'Simple') %>%
  bind_rows(results)
```

Comparing Models

- Logit regression (`glm`) - complex X

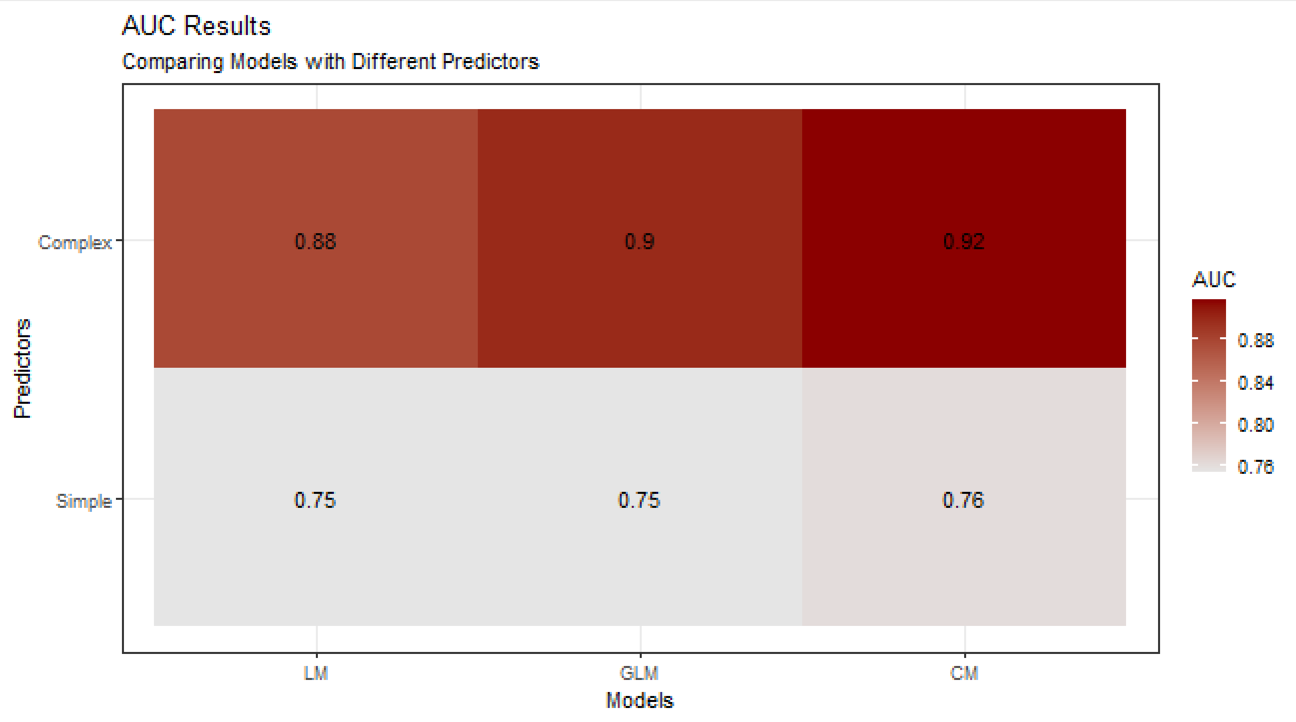
```
# Train
m <- glm(yield ~ sat + legacy + income,ad,family = binomial(link =
'logit'))

# Predict
toEval <- ad %>%
  mutate(prob_attend = predict(m,type = 'response'),
         truth = factor(yield,levels = c('1','0'))) %>%
    ungroup() %>%
    select(truth,prob_attend)

# Evaluate
results <- roc_auc(data = toEval,truth,prob_attend) %>%
  mutate(model = 'GLM',
         predictors = 'Complex') %>%
  bind_rows(results)
```

Comparing Models

```
p <- results %>%
  ggplot(aes(x = reorder(model,.estimate),
             y = reorder(predictors,.estimate),
             fill = .estimate,label = round(.estimate,2))) +
  geom_tile() +
  scale_fill_continuous(low = 'grey90',high = 'darkred') +
  geom_text() +
  labs(title = 'AUC Results',
       subtitle = 'Comparing Models with Different Predictors',
       x = 'Models',y = 'Predictors',
       fill = 'AUC') +
  theme_bw()
```

Comparing Models

```
p
```



Conclusion

- Conditional means outperform regression models?
  - Yes: conditional means allow for cell-specific predictions
  - No: conditional means are more susceptible to **overfitting**
- How would you re-evaluate these models-X-predictors to account for overfitting?