Predictive Modeling

DATA 606 - Statistics & Probability for Data Analytics

Jason Bryer, Ph.D. and Angela Lui, Ph.D.

November 15, 2023

One Minute Paper Results

What was the most important thing you learned during this class?

What important question remains unanswered for you?

Predictive Modeling

Example: Hours Studying Predicting Passing

```
study <- data.frame(
    Hours=c(0.50,0.75,1.00,1.25,1.50,1.75,1.75,2.00,2.25,2.50,2.75,3.00,
            3.25,3.50,4.00,4.25,4.50,4.75,5.00,5.50),
    Pass=c(0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,1,1,1,1,1)
)
study[sample(nrow(study), 5),]
```

```
##    Hours Pass
## 8   2.00    0
## 6   1.75    0
## 10  2.50    0
## 18  4.75    1
## 14  3.50    0
```

```
tab <- describeBy(study$Hours, group = study$Pass, mat = TRUE, skew = FALSE)
tab$group1 <- as.integer(as.character(tab$group1))
```

Prediction

Odds (or probability) of passing if studied **zero** hours?

$$log(\frac{p}{1-p}) = -4.078 + 1.505 \times 0$$

$$\frac{p}{1-p} = exp(-4.078) = 0.0169$$

$$p = \frac{0.0169}{1.169} = .016$$

Odds (or probability) of passing if studied **4** hours?

$$log(\frac{p}{1-p}) = -4.078 + 1.505 \times 4$$

$$\frac{p}{1-p} = exp(1.942) = 6.97$$

$$p = \frac{6.97}{7.97} = 0.875$$

Fitted Values

```r
study[1,]
```

```
##   Hours Pass
## 1   0.5    0
```

```r
logistic <- function(x, b0, b1) {
    return(1 / (1 + exp(-1 * (b0 + b1 * x)) ))
}
logistic(.5, b0=-4.078, b1=1.505)
```

```
## [1] 0.03470667
```

Model Performance

The use of statistical models to predict outcomes, typically on new data, is called predictive modeling. Logistic regression is a common statistical procedure used for prediction. We will utilize a **confusion matrix** to evaluate accuracy of the predictions.

		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR–}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR–) $= \frac{\text{FNR}}{\text{TNR}}$		

Predicting Heart Attacks

Source: https://www.kaggle.com/datasets/imnikhilanand/heart-attack-prediction?select=data.csv

```r
heart <- read.csv('../course_data/heart_attack_predictions.csv')
heart <- heart |>
    mutate_if(is.character, as.numeric) |>
    select(!c(slope, ca, thal))
str(heart)
```

```
## 'data.frame':    294 obs. of  11 variables:
##  $ age     : int  28 29 29 30 31 32 32 32 33 34 ...
##  $ sex     : int  1 1 1 0 0 0 1 1 1 0 ...
##  $ cp      : int  2 2 2 1 2 2 2 2 3 2 ...
##  $ trestbps: num  130 120 140 170 100 105 110 125 120 130 ...
##  $ chol    : num  132 243 NA 237 219 198 225 254 298 161 ...
##  $ fbs     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ restecg : num  2 0 0 1 1 0 0 0 0 0 ...
##  $ thalach : num  185 160 170 170 150 165 184 155 185 190 ...
##  $ exang   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ oldpeak : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ num     : int  0 0 0 0 0 0 0 0 0 0 ...
```

Note: `num` is the diagnosis of heart disease (angiographic disease status) (i.e. Value 0: < 50% diameter narrowing -- Value 1: > 50% diameter narrowing)

Missing Data

We will save this for another day...

```
complete.cases(heart) |> table()
```

```
## 
## FALSE  TRUE 
##    33   261
```

```
mice_out <- mice::mice(heart, m = 1)
```

```
## 
##  iter imp variable
##   1   1  trestbps  chol  fbs  restecg  thalach  exang
##   2   1  trestbps  chol  fbs  restecg  thalach  exang
##   3   1  trestbps  chol  fbs  restecg  thalach  exang
##   4   1  trestbps  chol  fbs  restecg  thalach  exang
##   5   1  trestbps  chol  fbs  restecg  thalach  exang
```

```
heart <- mice::complete(mice_out)
```

Data Setup

We will split the data into a training set (70% of observations) and validation set (30%).

```
train.rows <- sample(nrow(heart), nrow(heart) * .7)
heart_train <- heart[train.rows,]
heart_test <- heart[-train.rows,]
```

This is the proportions of survivors and defines what our "guessing" rate is. That is, if we guessed no one had a heart attack, we would be correct 62% of the time.

```
(heart_attack <- table(heart_train$num) %>% prop.table)
```

```
##
##        0        1
## 0.604878 0.395122
```

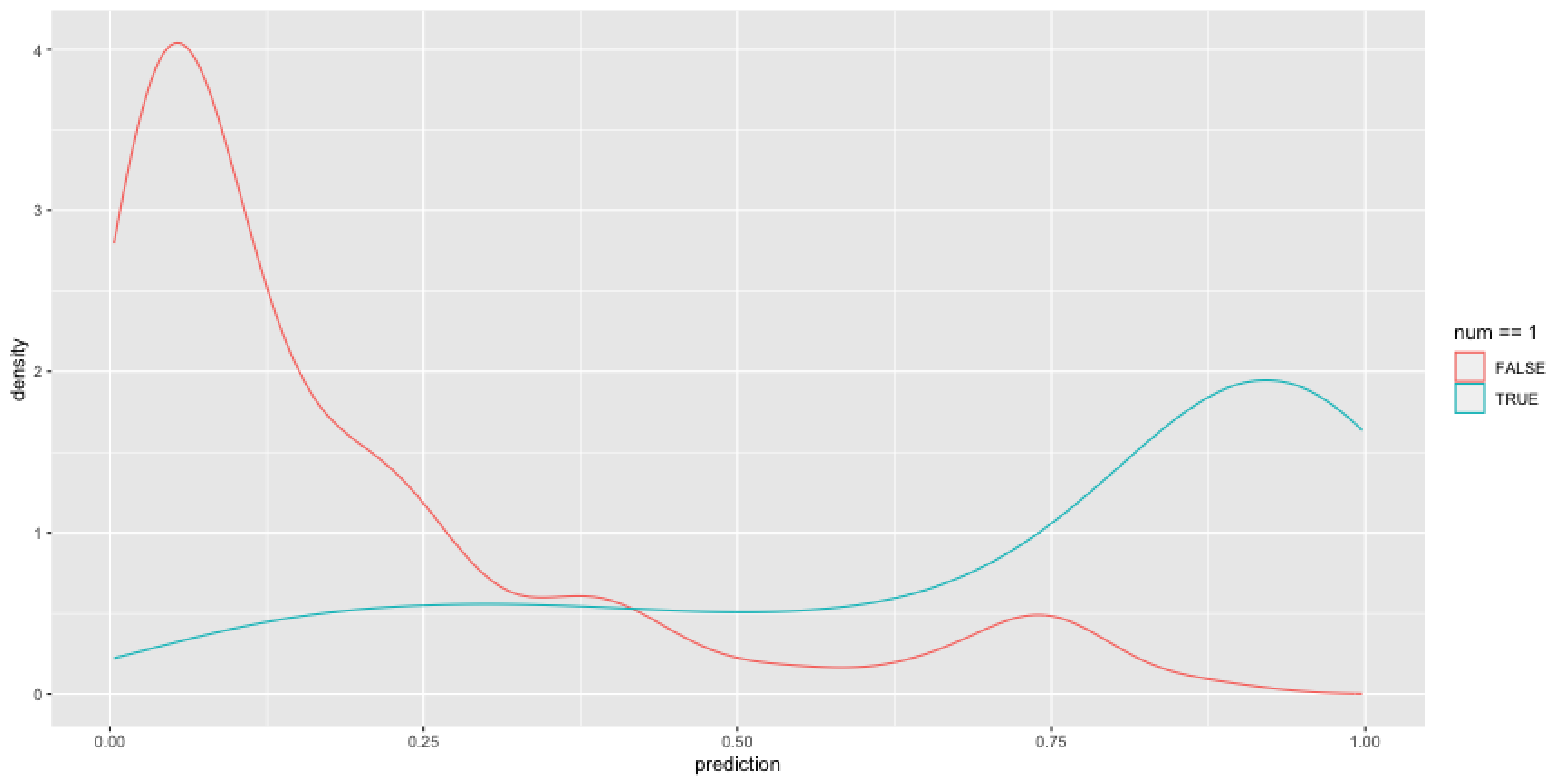
Model Training

```
lr.out <- glm(num ~ ., data=heart_train, family=binomial(link = 'logit'))
summary(lr.out)
```

```
##
## Call:
## glm(formula = num ~ ., family = binomial(link = "logit"), data = heart_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.668309   3.440478  -1.066 0.286324
## age         -0.002669   0.031400  -0.085 0.932256
## sex          0.710022   0.543585   1.306 0.191490
## cp           0.952200   0.259580   3.668 0.000244 ***
## trestbps    -0.005169   0.013774  -0.375 0.707467
## chol         0.006203   0.002930   2.117 0.034291 *
## fbs          1.287901   0.832381   1.547 0.121803
## restecg     -0.328599   0.529274  -0.621 0.534699
## thalach     -0.016133   0.011356  -1.421 0.155412
## exang        0.916571   0.535843   1.711 0.087169 .
## oldpeak      1.253931   0.294198   4.262 2.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 275.10  on 204  degrees of freedom
## Residual deviance: 148.27  on 194  degrees of freedom
## AIC: 170.27
##
```

Predicted Values

```
heart_train$prediction <- predict(lr.out, type = 'response', newdata = heart_train)
ggplot(heart_train, aes(x = prediction, color = num == 1)) + geom_density()
```



Results

```
heart_train$prediction_class <- heart_train$prediction > 0.5
tab <- table(heart_train$prediction_class,
             heart_train$num) %>% prop.table() %>% print()
```

```
##
##                  0          1
##   FALSE 0.54146341 0.10243902
##   TRUE  0.06341463 0.29268293
```

For the training set, the overall accuracy is 83.41%. Recall that 60.49% people did not have a heart attach. Therefore, the simplest model would be to predict that no one had a heart attack, which would mean we would be correct 60.49% of the time. Therefore, our prediction model is 22.93% better than guessing.

Checking with the validation dataset

```
(survived_test <- table(heart_test$num) %>% prop.table())
```

```
##
##         0         1
## 0.7191011 0.2808989
```

```
heart_test$prediction <- predict(lr.out, newdata = heart_test, type = 'response')
heart_test$prediciton_class <- heart_test$prediction > 0.5
tab_test <- table(heart_test$prediciton_class, heart_test$num) %>%
    prop.table() %>% print()
```

```
##
##                  0          1
##   FALSE 0.65168539 0.06741573
##   TRUE  0.06741573 0.21348315
```

The overall accuracy is 86.52%, or 14.6% better than guessing.

Receiver Operating Characteristic (ROC) Curve

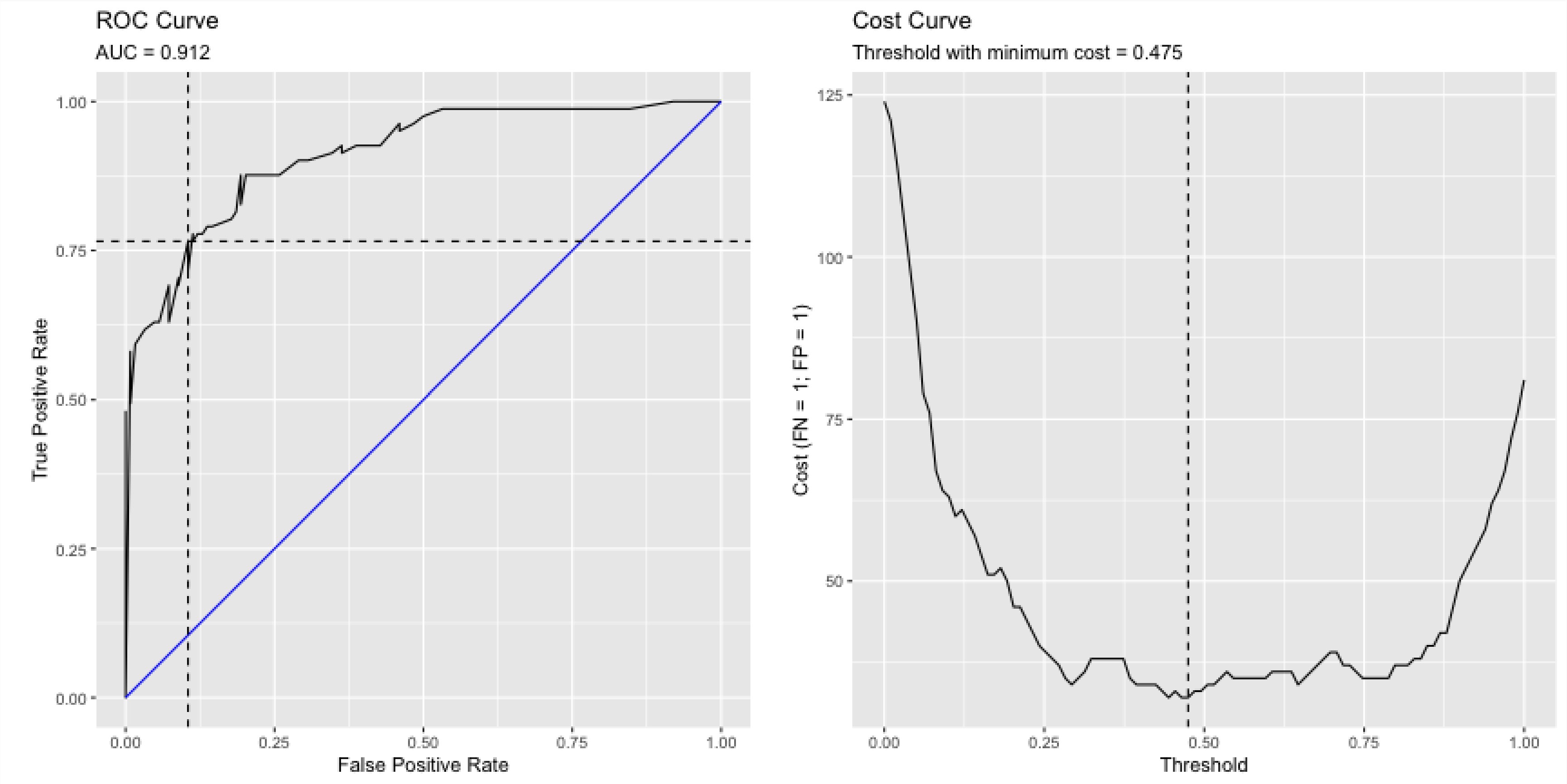
The ROC curve is created by plotting the true positive rate (TPR; AKA sensitivity) against the false positive rate (FPR; AKA probability of false alarm) at various threshold settings.

```r
roc <- calculate_roc(heart_train$prediction, heart_train$num == 1)
summary(roc)
```

```
## AUC = 0.912
## Cost of false-positive = 1
## Cost of false-negative = 1
## Threshold with minimum cost = 0.475
```

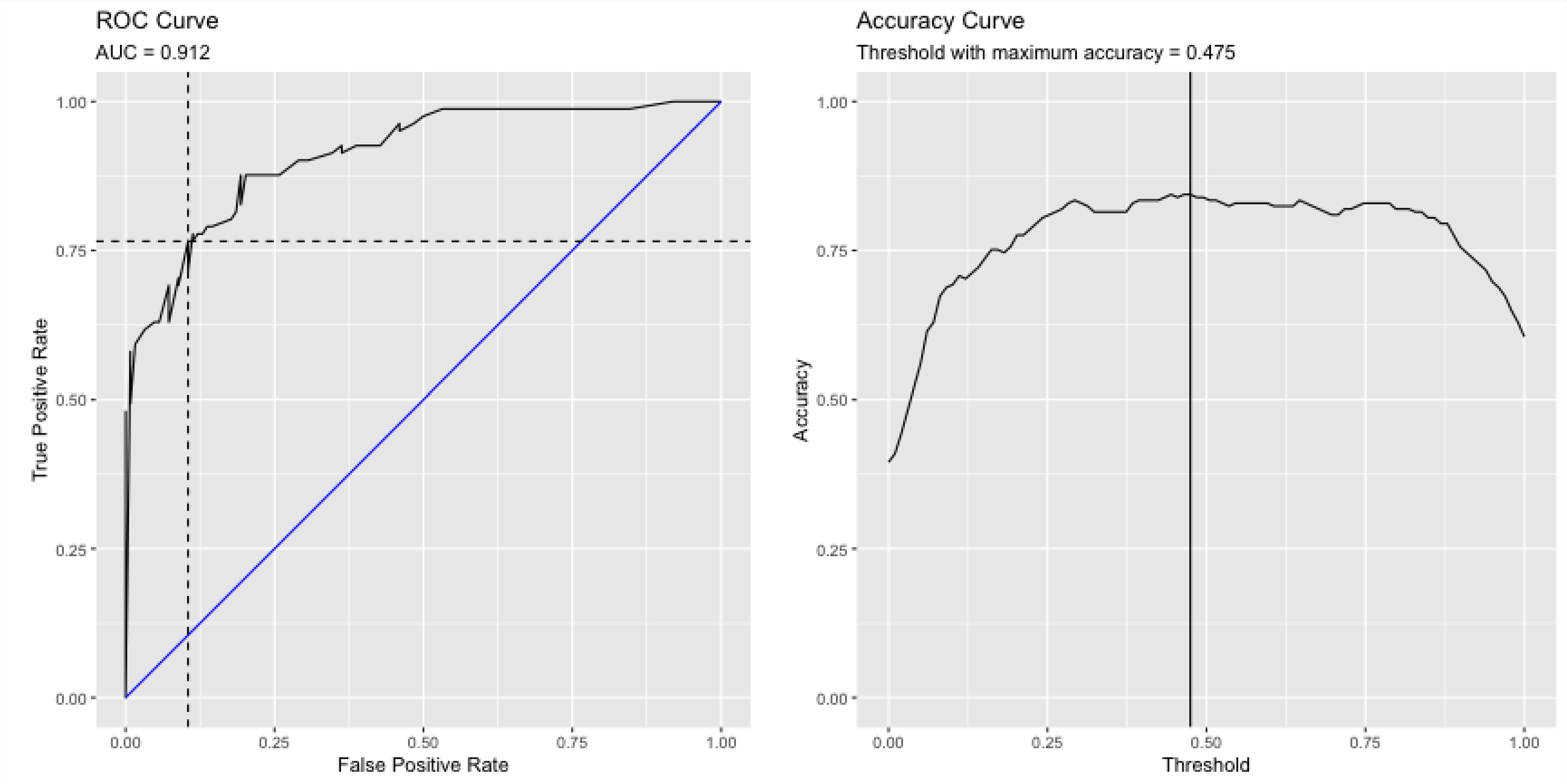
ROC Curve

```
plot(roc)
```



ROC Curve

```r
plot(roc, curve = 'accuracy')
```



Caution on Interpreting Accuracy

- Loh, Sooo, and Zing (2016) predicted sexual orientation based on Facebook Status.
- They reported model accuracies of approximately 90% using SVM, logistic regression and/or random forest methods.
- Gallup (2018) poll estimates that 4.5% of the Americal population identifies as LGBT.
- *My proposed model:* I predict all Americans are heterosexual.
- The accuracy of my model is 95.5%, or *5.5% better than Facebook's model!*
- Predicting "rare" events (i.e. when the proportion of one of the two outcomes large) is difficult and requires independent (predictor) variables that strongly associated with the dependent (outcome) variable.

Fitted Values Revisited

What happens when the ratio of true-to-false increases (i.e. want to predict "rare" events)?

Let's simulate a dataset where the ratio of true-to-false is 10-to-1. We can also define the distribution of the dependent variable. Here, there is moderate separation in the distributions.

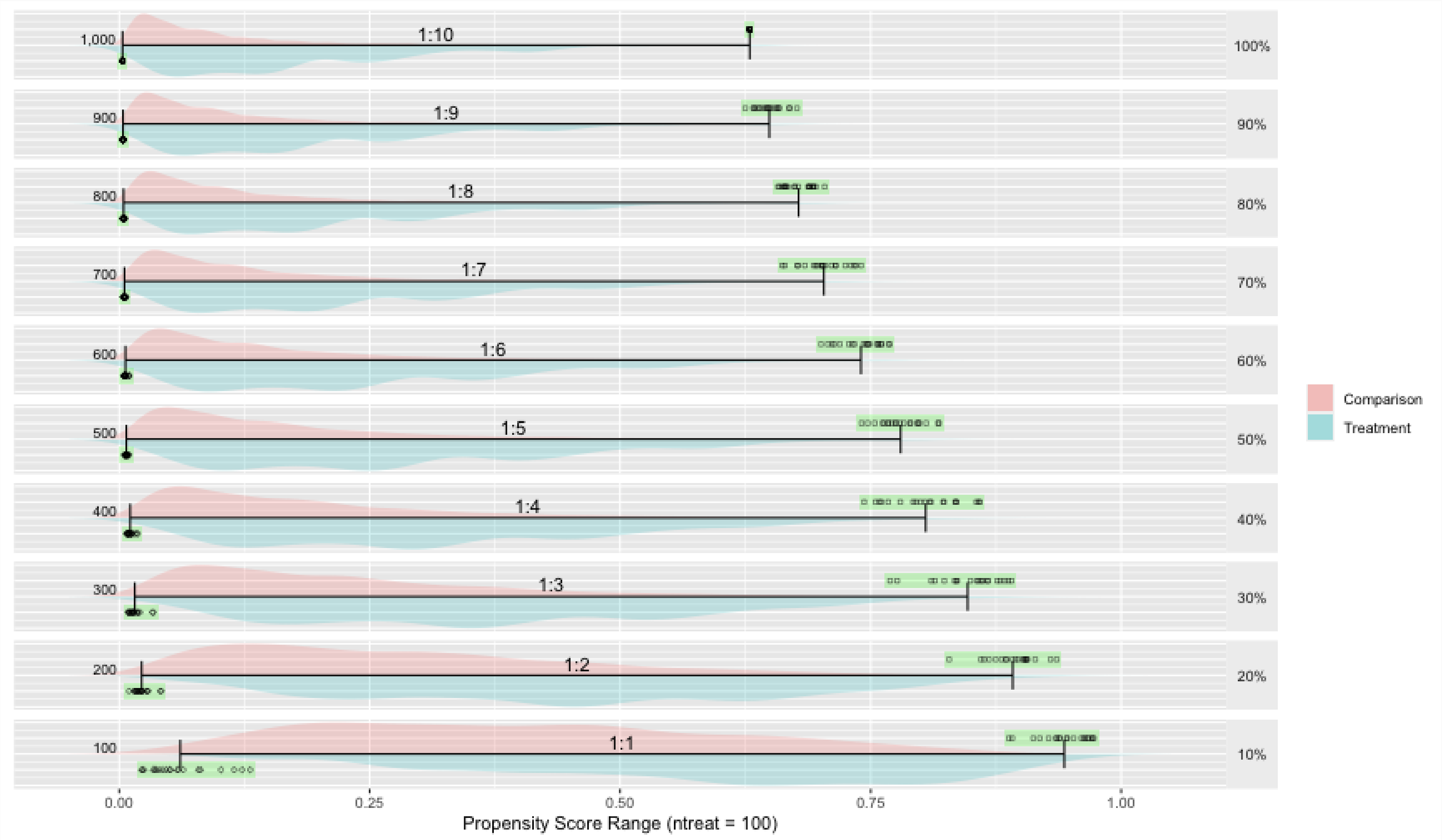
```
test.df2 <- getSimulatedData(
    treat.mean=.6, control.mean=.4)
```

The `multilevelPSA::psrange` function will sample with varying ratios from 1:10 to 1:1. It takes multiple samples and averages the ranges and distributions of the fitted values from logistic regression.

```
psranges2 <- psrange(test.df2, test.df2$treat, treat ~ .,
                     samples=seq(100,1000,by=100), nboot=20)
```

Fitted Values Revisited (cont.)

```
plot(psranges2)
```



Additional Resources

- The Path to Log Likelihood
- Visual Introduction to Maximum Likelihood Estimation
- VisualStats R Package
- Logistic Regression Details Pt 2: Maximum Likelihood
- StatQuest: Maximum Likelihood, clearly explained
- Probability concepts explained: Maximum likelihood estimation

One Minute Paper

Complete the one minute paper:

https://forms.gle/ngYXfC6jwY3TV6FXA

1. What was the most important thing you learned during this class?
2. What important question remains unanswered for you?