Logistic Regression

EPSY 630 - Statistics II

Jason Bryer, Ph.D.

March 30, 2021

Agenda

Reminder: No class next week. Have a relaxing and safe Spring break!

1 Logistic Regression

2 Predictive Modeling

3 Questions

4 One minute papers

Logistic Regression

Relationship between dichotomous (x) and continuous

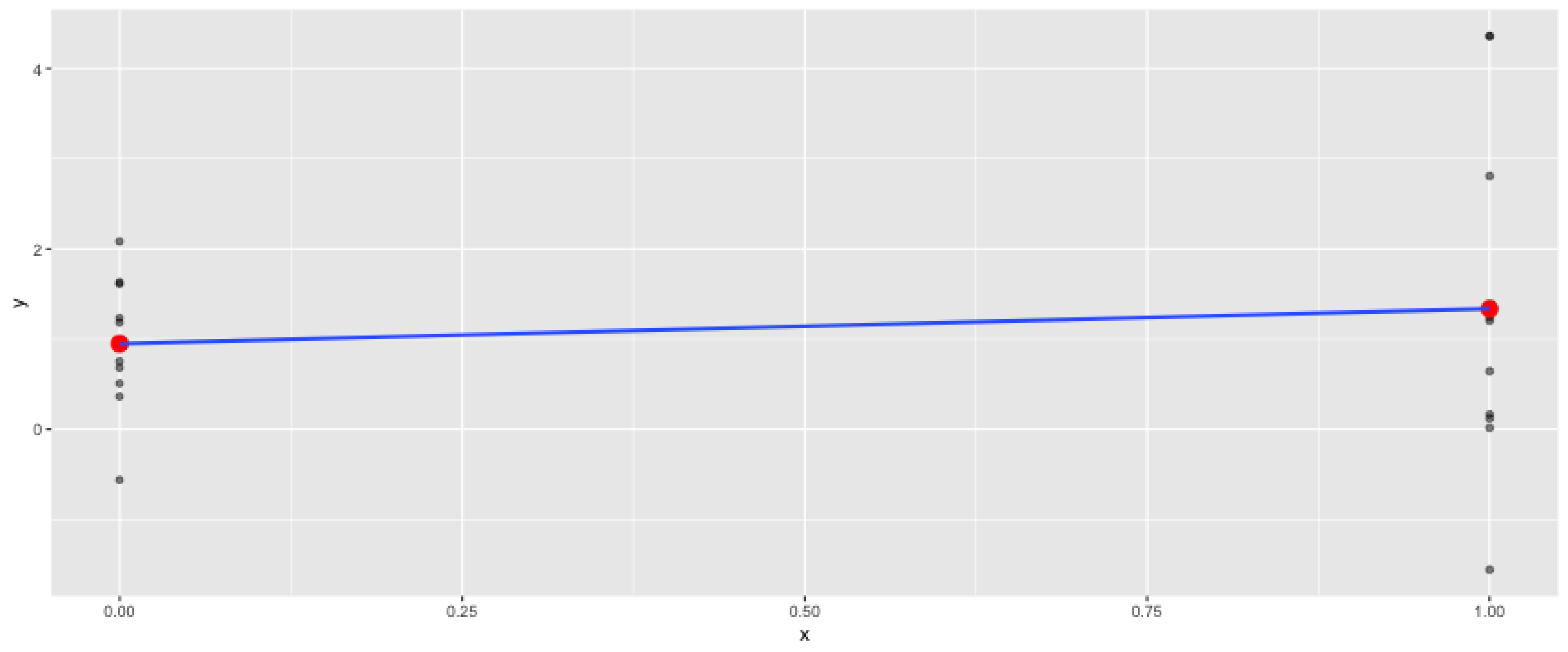
```
df <- data.frame(
    x = rep(c(0, 1), each = 10),
    y = c(rnorm(10, mean = 1, sd = 1),
          rnorm(10, mean = 2.5, sd = 1.5))
)
head(df)
```

```
##   x         y
## 1 0 1.1858679
## 2 0 0.7529346
## 3 0 1.2359476
## 4 0 2.0863606
## 5 0 1.6318793
## 6 0 0.5075317
```

```
tab <- describeBy(df$y, group = df$x, mat = TRUE, skew = FALSE)
tab$group1 <- as.integer(as.character(tab$group1))
```

Relationship between dichotomous (x) and continuous

```r
ggplot(df, aes(x = x, y = y)) +    geom_point(alpha = 0.5) +
    geom_point(data = tab, aes(x = group1, y = mean), color = 'red', size = 4) +
    geom_smooth(method = lm, se = FALSE, formula = y ~ x)
```



EPSY 630
Spring 2021

Regression so far...

At this point we have covered:

- Simple linear regression
  - Relationship between numerical response and a numerical or categorical predictor
- Multiple regression
  - Relationship between numerical response and multiple numerical and/or categorical predictors
- Maximum Likelihood Estimation

All of the approaches we have used so far have a quantitative variable with normally distributed errors (i.e. residuals).

What we haven't seen is what to do when the predictors are weird (nonlinear, complicated dependence structure, etc.) or when the response is weird (categorical, count data, etc.)

Odds

Odds are another way of quantifying the probability of an event, commonly used in gambling (and logistic regression).

For some event E,

$$\text{odds}(E) = \frac{P(E)}{P(E^c)} = \frac{P(E)}{1 - P(E)}$$

Similarly, if we are told the odds of E are x to y then

$$\text{odds}(E) = \frac{x}{y} = \frac{x/(x+y)}{y/(x+y)}$$

which implies

$$P(E) = x/(x+y), \quad P(E^c) = y/(x+y)$$

Generalized Linear Models

Generalized linear models (GLM) are a generalization of OLS that allows for the response variables (i.e. dependent variables) to have an error distribution that is ***not*** distributed normally. All generalized linear models have the following three characteristics:

1. A probability distribution describing the outcome variable .

2. A linear model: $\eta = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$.

3. A link function that relates the linear model to the parameter of the outcome distribution: $g(p) = \eta$ or $p = g^{-1}(\eta)$.

We can estimate GLMs using maximum likelihood estimation (MLE). What will change is the log-likelihood function.

Logistic Regression

Logistic regression is a GLM used to model a binary categorical variable using numerical and categorical predictors.

We assume a binomial distribution produced the outcome variable and we therefore want to model p the probability of success for a given set of predictors.

To finish specifying the Logistic model we just need to establish a reasonable link function that connects η to p. There are a variety of options but the most commonly used is the logit function.

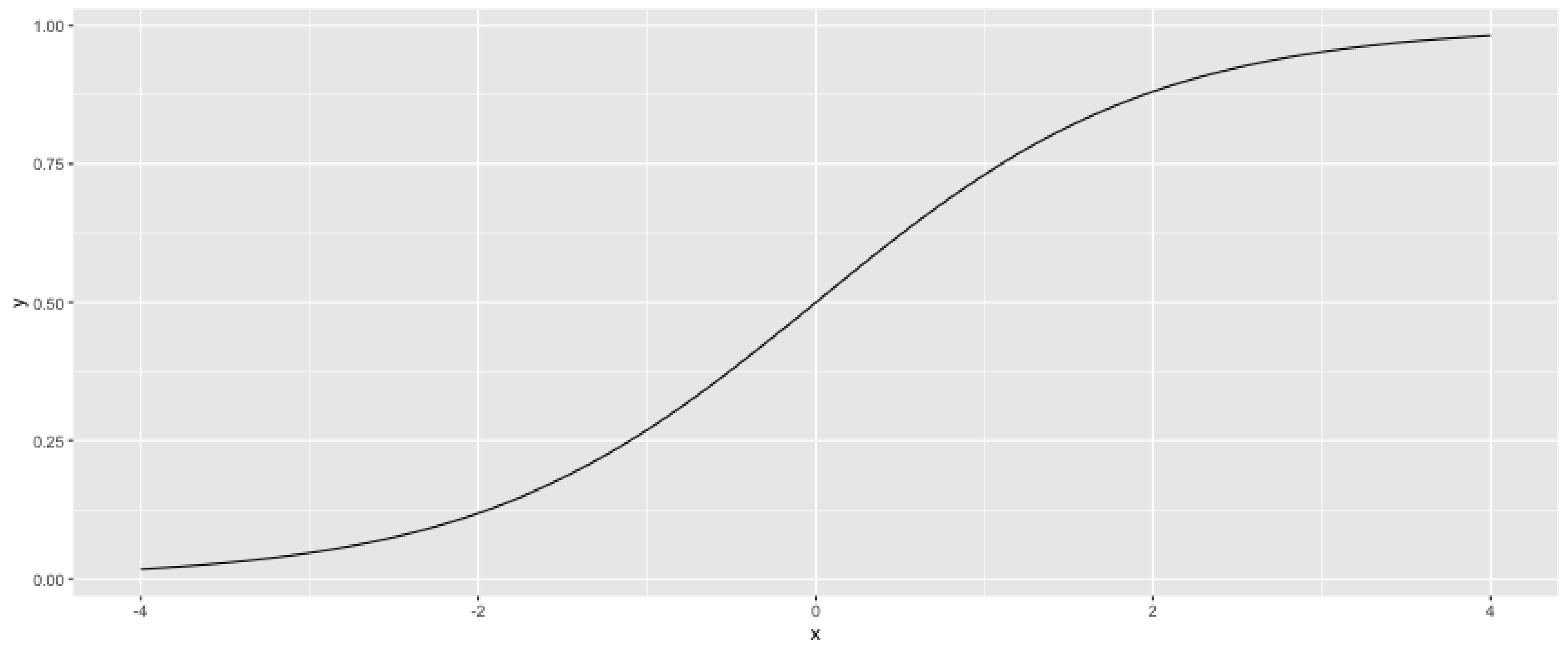
Logit function

$$logit(p) = \log\left(\frac{p}{1-p}\right), \text{ for } 0 \le p \le 1$$

The Logistic Function

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

```
logistic <- function(t) { return(1 / (1 + exp(-t))) }
ggplot() + stat_function(fun = logistic, n = 101) +  xlim(-4, 4) + xlab('x')
```



EPSY 630
Spring 2021

t as a Linear Function

$$t = \beta_0 + \beta_1 x$$

The logistic function can now be rewritten as

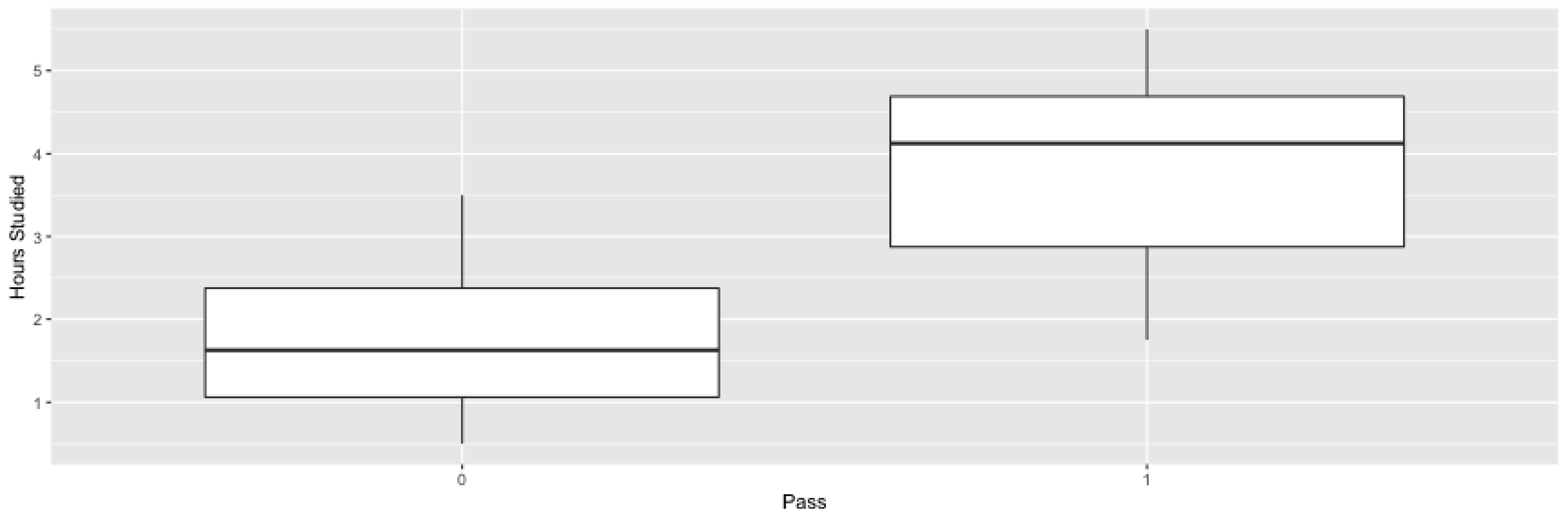
$$F\left(x\right) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Similar to OLS, we wish to minimize the errors. However, instead of minimizing the least squared residuals, we will use a maximum likelihood function.

Example: Hours Studying Predicting Passing

```r
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)
)
```

```r
ggplot(study, aes(x=factor(Pass), y=Hours)) + geom_boxplot() + xlab('Pass') + ylab('Hours Stud
```



Loglikelihood Function

We need to define logit function and the log-likelihood function that will be used by the optim function. Instead of using the normal distribution as above (using the dnorm function), we are using a binomial distribution and the logit to link the linear combination of predictors.

```r
logit <- function(x, beta0, beta1) {
    return( 1 / (1 + exp(-beta0 - beta1 * x)) )
}
loglikelihood.binomial <- function(parameters, predictor, outcome) {
    a <- parameters[1] # Intercept
    b <- parameters[2] # beta coefficient
    p <- logit(predictor, a, b)
    ll <- sum( outcome * log(p) + (1 - outcome) * log(1 - p))
    return(ll)
}
```

Estimating parameters using the optim function

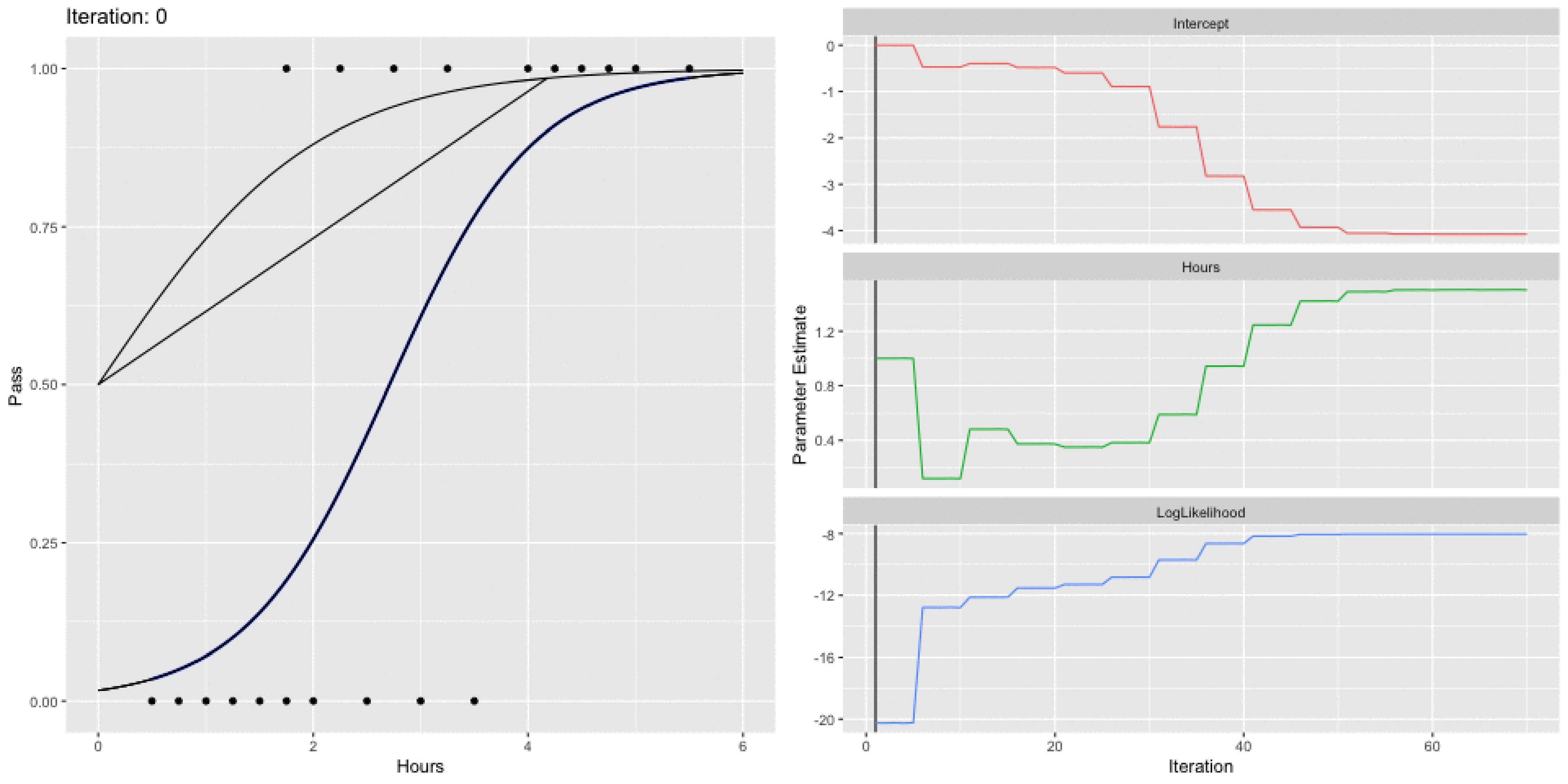
```r
optim.binomial <- optim_save(
    c(0, 1), # Initial values
    loglikelihood.binomial,
    method = "L-BFGS-B",
    control = list(fnscale = -1),
    predictor = study$Hours,
    outcome = study$Pass
)

optim.binomial$par
```

```
## [1] -4.077575  1.504624
```

EPSY 630
Spring 2021

How did the optimizer get this result?



The glm function

```
( lr.out <- glm(Pass ~ Hours, data = study, family = binomial(link = 'logit')) )
```

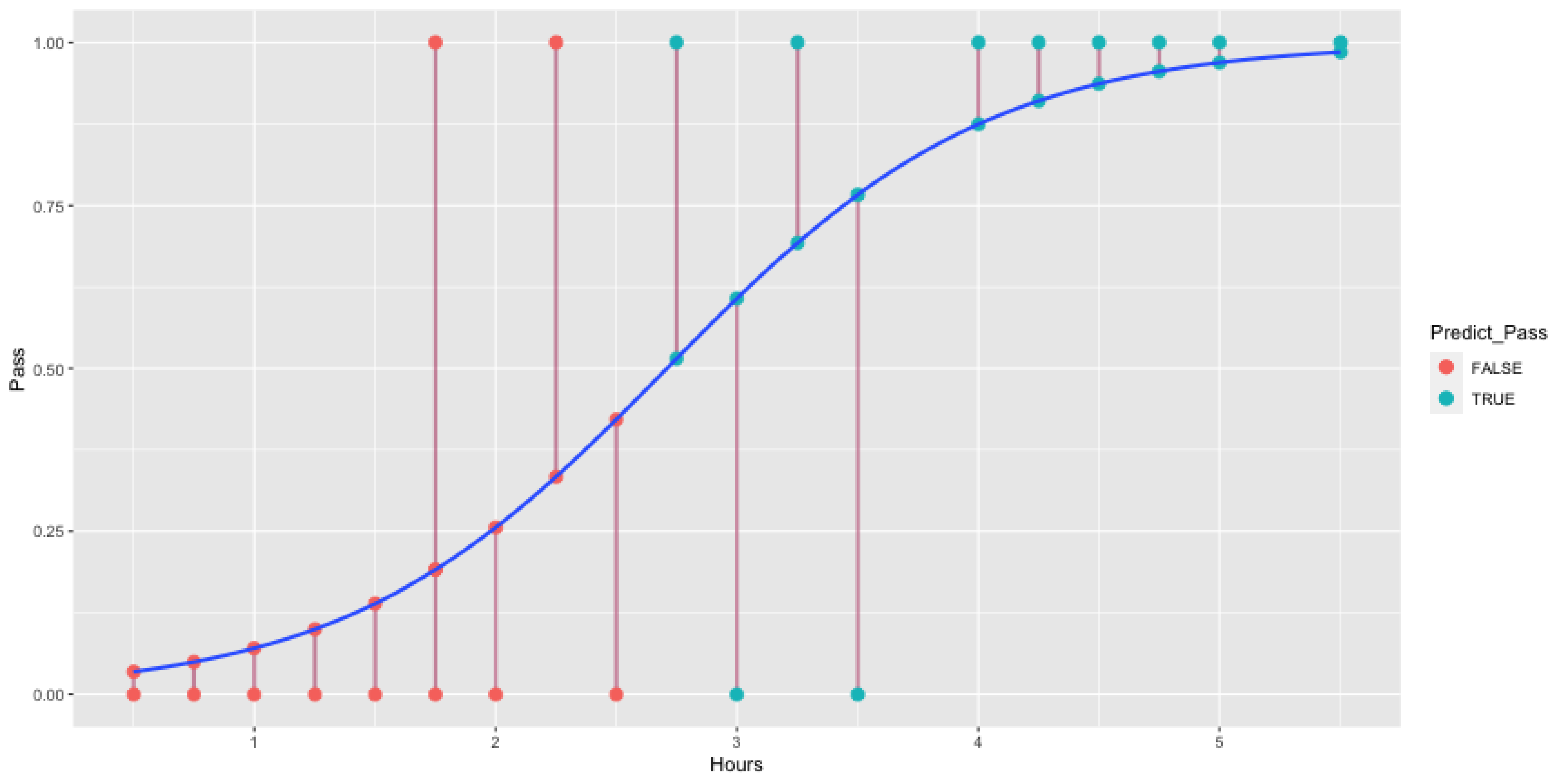
```
##
## Call:  glm(formula = Pass ~ Hours, family = binomial(link = "logit"),
##     data = study)
##
## Coefficients:
## (Intercept)        Hours
##      -4.078        1.505
##
## Degrees of Freedom: 19 Total (i.e. Null);  18 Residual
## Null Deviance:        27.73
## Residual Deviance: 16.06     AIC: 20.06
```

How does this compare to the `optim` function?

```
optim.binomial$par
```

```
## [1] -4.077575  1.504624
```

Plotting the Results



Predictive Modeling

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?

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

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

Fitted Values

```
study[1,]
```

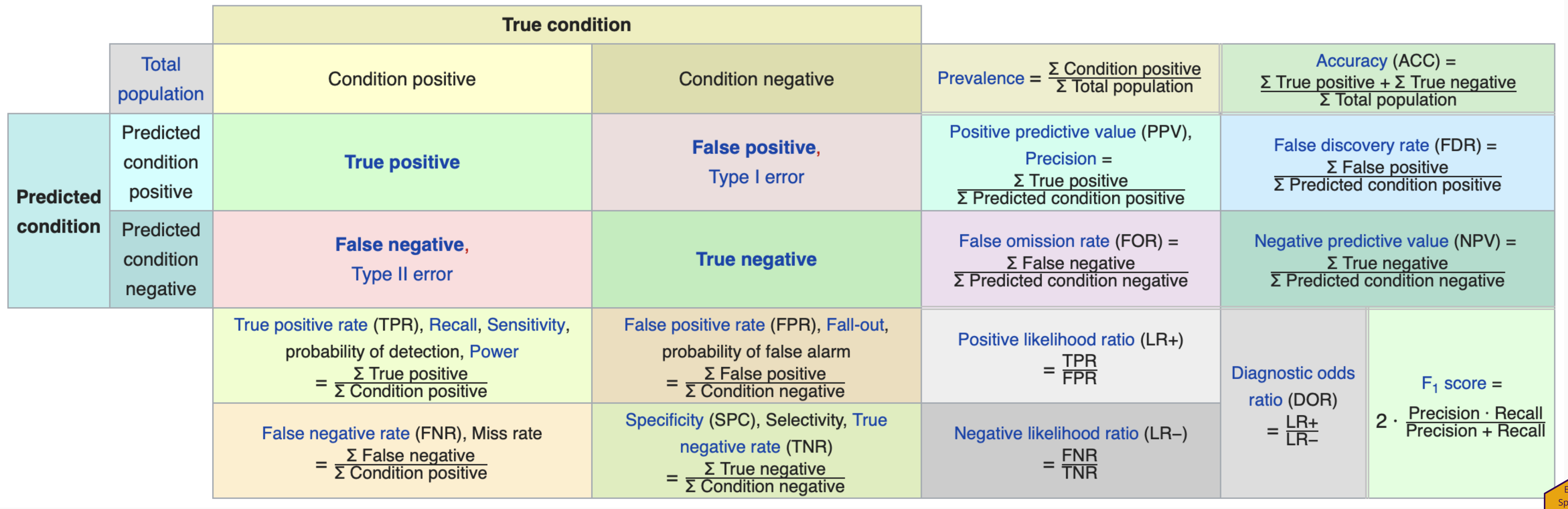
```
##   Hours Pass    Predict Predict_Pass          p
## 1   0.5    0 0.03471034        FALSE 0.03471462
```

```
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 survivors of the Titanic

```
str(titanic_train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Data Setup

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

```
train.rows <- sample(nrow(titanic), nrow(titanic) * .7)
titanic_train <- titanic[train.rows,]
titanic_test <- titanic[-train.rows,]
```

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

```
(survived <- table(titanic_train$survived) %>% prop.table)
```

```
## 
##        No       Yes 
## 0.6124454 0.3875546
```

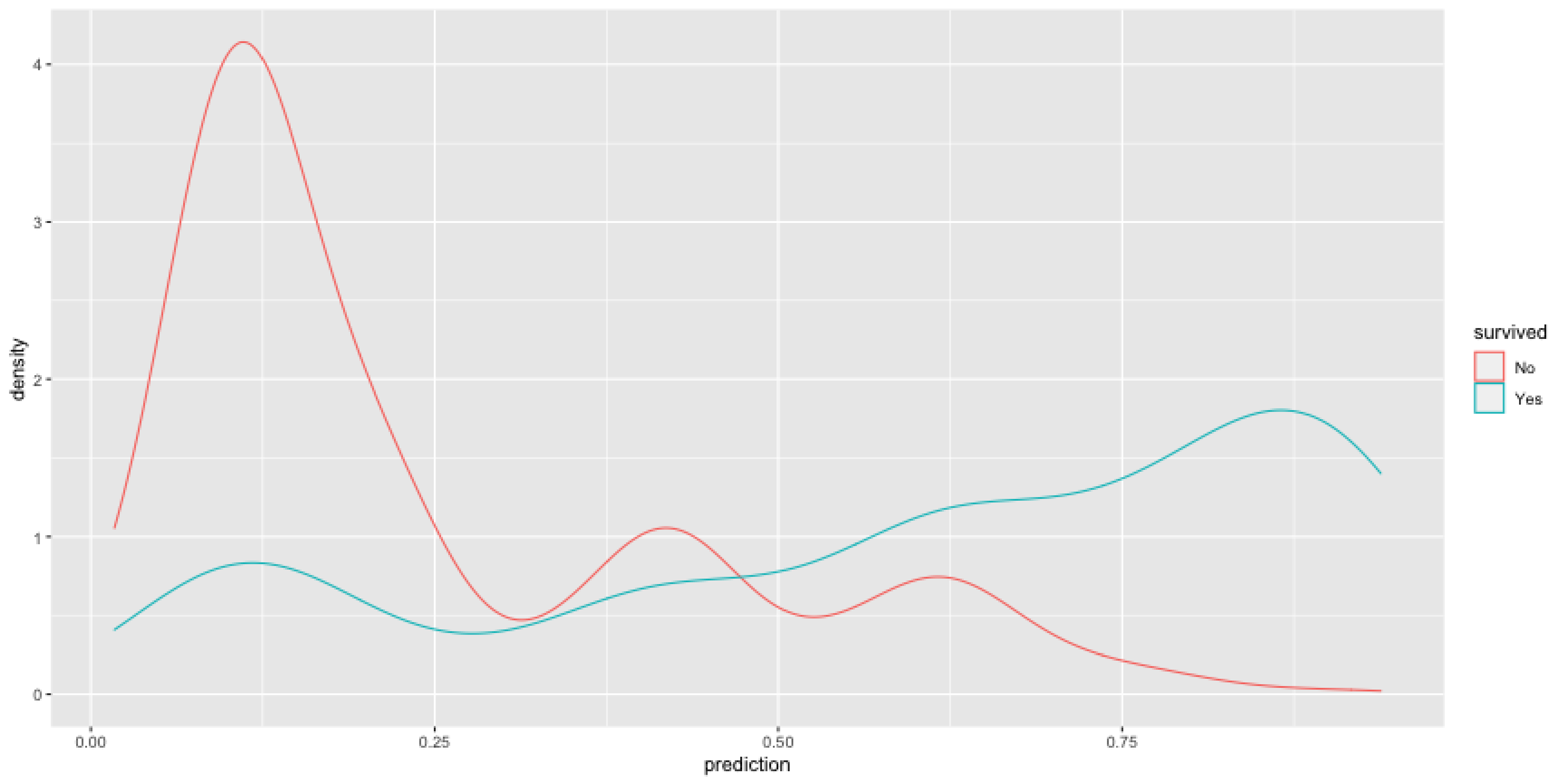
Model Training

```
lr.out <- glm(survived ~ pclass + sex + sibsp + parch, data=titanic_train, family=binomial(link = 'logit'))
summary(lr.out)
```

```
## 
## Call:
## glm(formula = survived ~ pclass + sex + sibsp + parch, family = binomial(link = "logit"), 
##     data = titanic_train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.2174  -0.6918  -0.4780   0.6890   2.3446  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  1.37944    0.16112   8.562  < 2e-16 ***
## pclass.L    -1.26861    0.14736  -8.609  < 2e-16 ***
## pclass.Q     0.07621    0.16802   0.454  0.65012    
## sexmale     -2.62525    0.18446 -14.232  < 2e-16 ***
## sibsp       -0.28536    0.10716  -2.663  0.00775 ** 
## parch        0.17332    0.11362   1.526  0.12713    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

Predicted Values

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



Results

```r
titanic_train$prediction_class <- titanic_train$prediction > 0.5
tab <- table(titanic_train$prediction_class,
             titanic_train$survived) %>% prop.table() %>% print()
```

```
##
##                  No        Yes
##    FALSE 0.52292576 0.12117904
##    TRUE  0.08951965 0.26637555
```

For the training set, the overall accuracy is 78.93%. Recall that 38.76% of passengers survived. Therefore, the simplest model would be to predict that everyone died, which would mean we would be correct 61.24% of the time. Therefore, our prediction model is 17.69% better than guessing.

Checking with the validation dataset

```
(survived_test <- table(titanic_test$survived) %>% prop.table())
```

```
##
##        No       Yes
## 0.6310433 0.3689567
```

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

```
##
##                  No        Yes
##   FALSE 0.55470738 0.13231552
##   TRUE  0.07633588 0.23664122
```

The overall accuracy is 79.13%, or 16% better than guessing.

Receiver Operating Characteristic (ROC) Curve

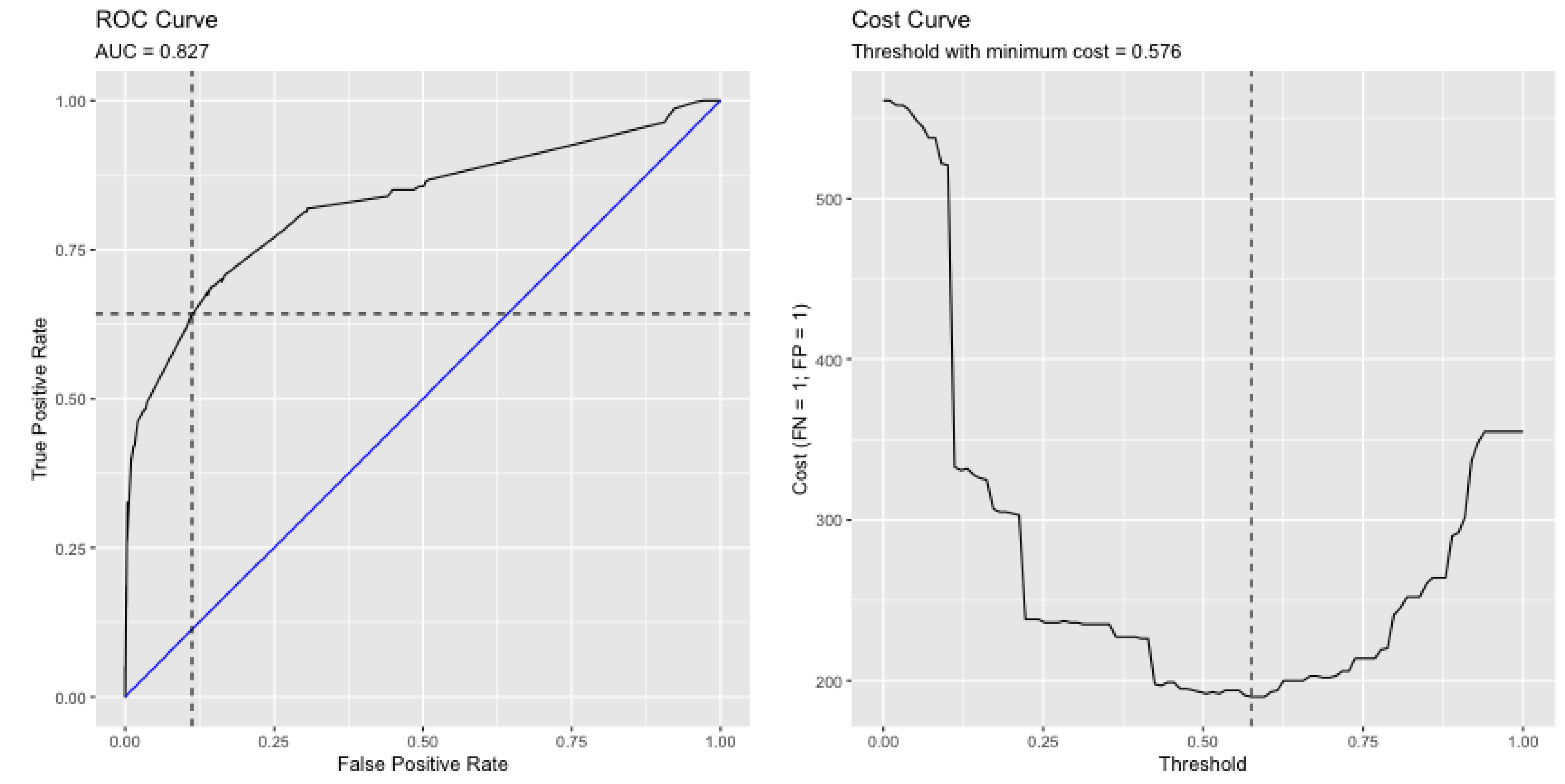
The ROC curve is created by plotting the true positive rate (TPR; AKA sensitivity) against the false positive rate (FPR; AKA probability of false alarm) at various threshold settings.

```r
roc <- calculate_roc(titanic_train$prediction, titanic_train$survived == 'Yes')
summary(roc)
```

```
## AUC = 0.827
## Cost of false-positive = 1
## Cost of false-negative = 1
## Threshold with minimum cost = 0.576
```

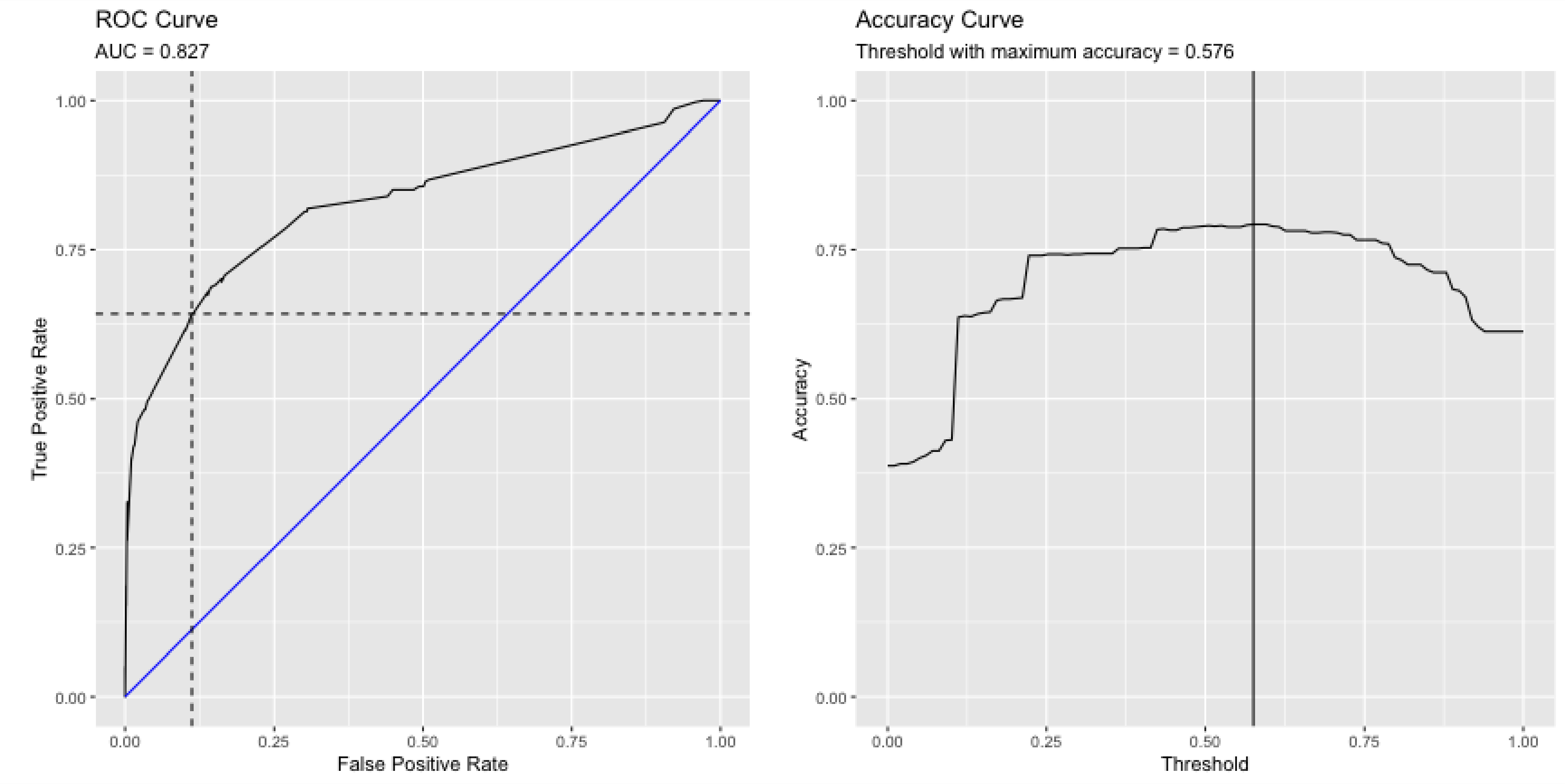
ROC Curve

```
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

- Logistic Regression Details Pt 2: Maximum Likelihood
- StatQuest: Maximum Likelihood, clearly explained
- Probability concepts explained: Maximum likelihood estimation

Questions

One Minute Paper

Complete the one minute paper:

https://forms.gle/yB3ds6MYE89Z1pURA

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