Week 10 - Homework

STAT 420, Summer 2018, Connor Segneri - segneri3

Contents

Exercise 1 (Simulating Wald and Likelihood Ratio Tests) 1
Exercise 2 (Surviving the Titanic) 5
Exercise 3 (Breast Cancer Detection) 8

Exercise 1 (Simulating Wald and Likelihood Ratio Tests)

In this exercise we will investigate the distributions of hypothesis tests for logistic regression. For this exercise, we will use the following predictors.

```r
sample_size = 150
set.seed(420)
x1 = rnorm(n = sample_size)
x2 = rnorm(n = sample_size)
x3 = rnorm(n = sample_size)
```

Recall that

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

Consider the true model

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1$$

where

- $\beta_0 = 0.4$
- $\beta_1 = -0.35$

(a) To investigate the distributions, simulate from this model 2500 times. To do so, calculate

$$P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

for an observation, and then make a random draw from a Bernoulli distribution with that success probability. (Note that a Bernoulli distribution is a Binomial distribution with parameter $n = 1$. There is no direction function in `R` for a Bernoulli distribution.)

Each time, fit the model:

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

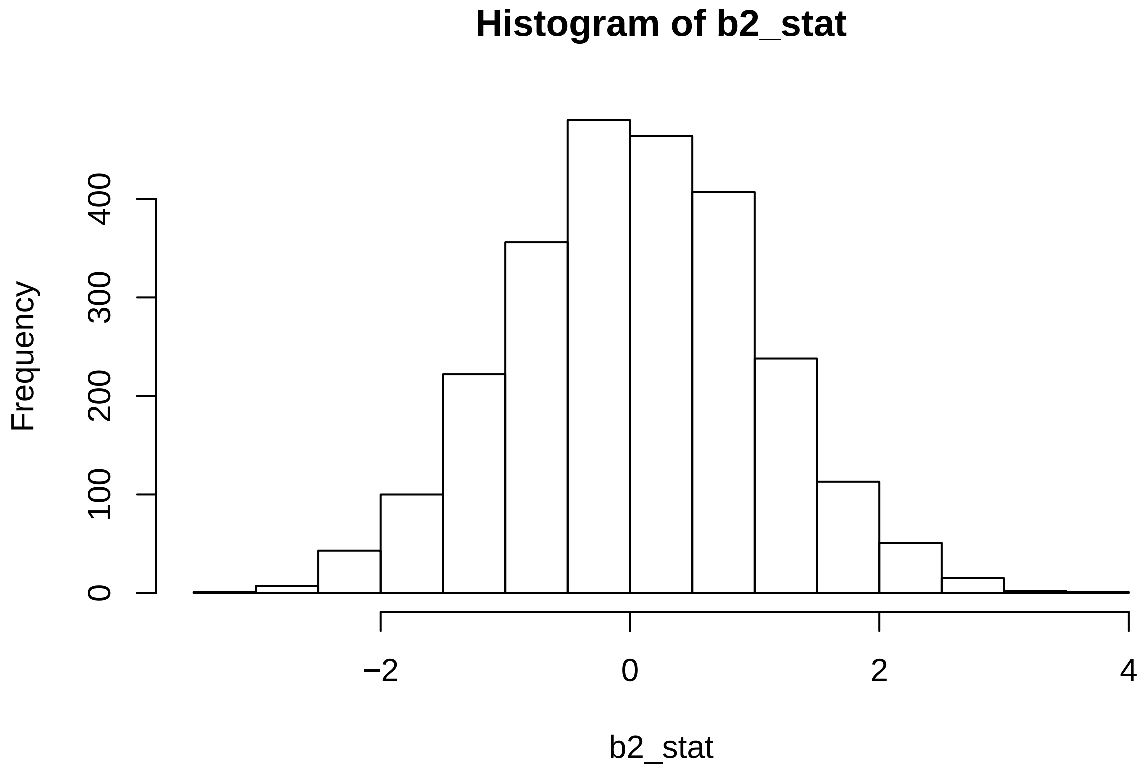
Store the test statistics for two tests:

- The Wald test for $H_0 : \beta_2 = 0$, which we say follows a standard normal distribution for "large" samples
- The likelihood ratio test for $H_0 : \beta_2 = \beta_3 = 0$, which we say follows a χ^2 distribution (with some degrees of freedom) for "large" samples

```
b0 = 0.4
b1 = -0.35

b2_stat = rep(0,2500)
b2_b3_stat = rep(0,2500)
for (i in 1:2500) {
  eta = b0 + b1*x1
  p = 1 / (1 + exp(-eta))
  y = rbinom(n = sample_size, size = 1, prob = p)
  data = data.frame(y, x1, x2, x3)

  fit_all = glm(y ~ ., data = data, family = binomial)
  fit_b1 = glm(y ~ x1, data = data, family = binomial)

  b2_stat[i] = summary(fit_all)$coefficients[3,3]
  b2_b3_stat[i] = anova(fit_b1, fit_all)$Deviance[2]
}
```

(b) Plot a histogram of the empirical values for the Wald test statistic. Overlay the density of the true distribution assuming a large sample.

```
hist(b2_stat)
```

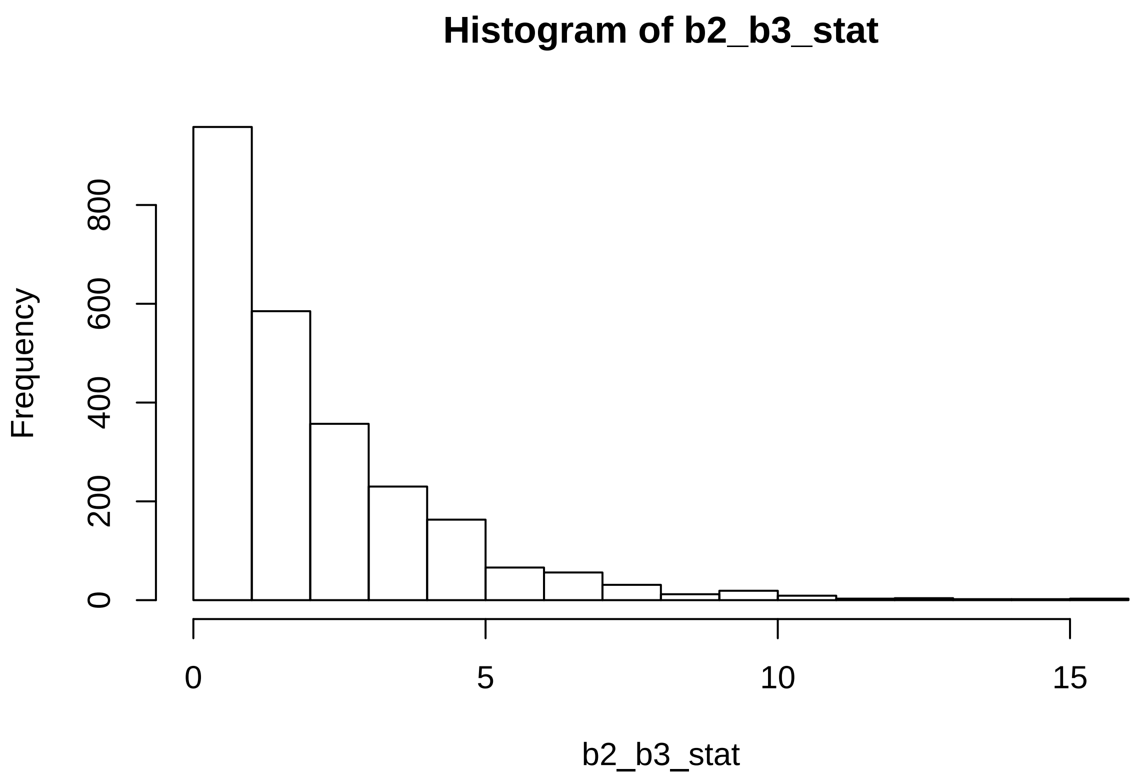


(c) Use the empirical results for the Wald test statistic to estimate the probability of observing a test statistic larger than 1. Also report this probability using the true distribution of the test statistic assuming a large sample.

The probability of observing a Wald test statistic larger than 1 for testing $H_0 : \beta_2 = 0$ is `0.168`.

(d) Plot a histogram of the empirical values for the likelihood ratio test statistic. Overlay the density of the true distribution assuming a large sample.

```r
hist(b2_b3_stat)
```



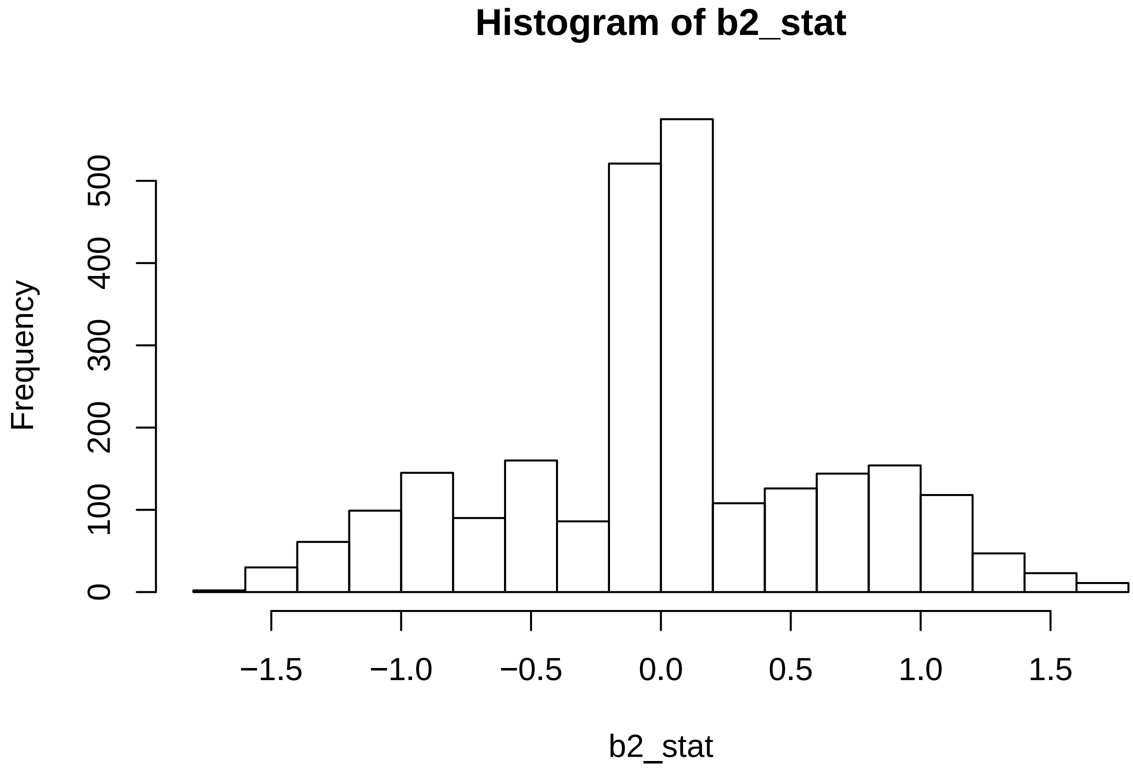
(e) Use the empirical results for the likelihood ratio test statistic to estimate the probability of observing a test statistic larger than 5. Also report this probability using the true distribution of the test statistic assuming a large sample.

The probability of observing a likelihood ratio test statistic larger than 5 for testing $H_0 : \beta_2 = \beta_3 = 0$ is `0.0828`.

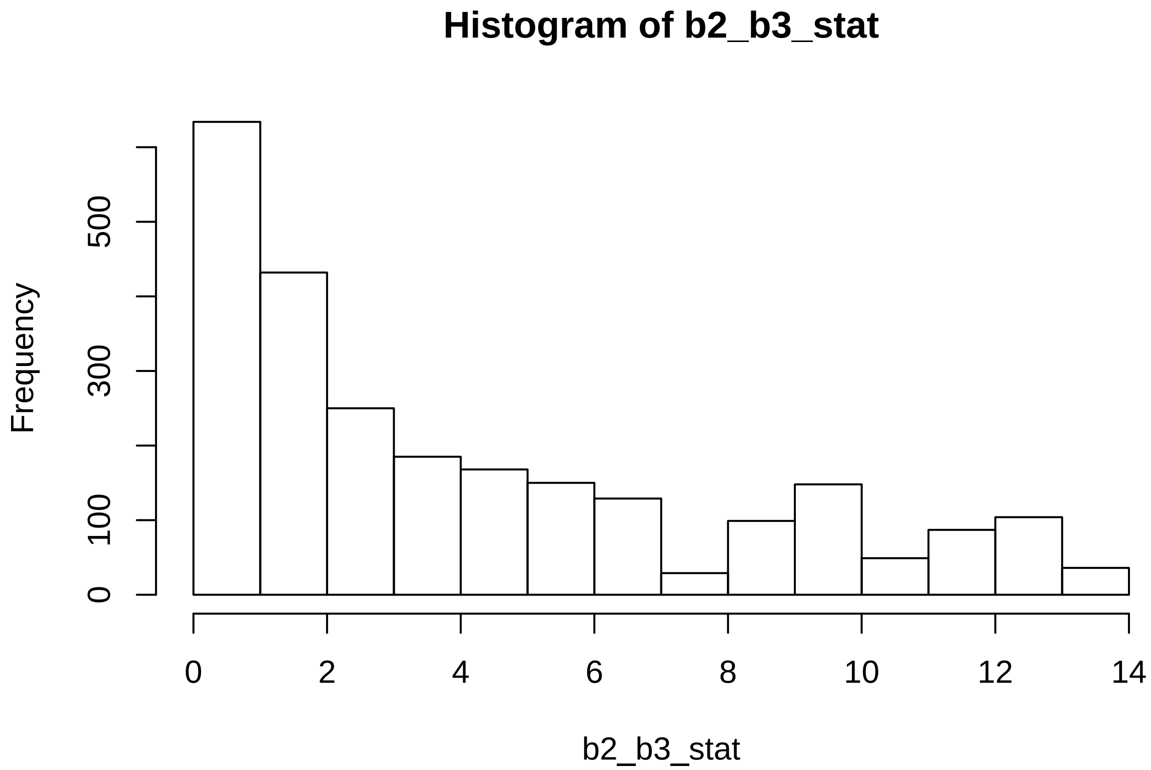
(f) Repeat **(a)**-**(e)** but with simulation using a smaller sample size of 10. Based on these results, is this sample size large enough to use the standard normal and χ^2 distributions in this situation? Explain.

```r
sample_size = 10
set.seed(420)
x1 = rnorm(n = sample_size)
x2 = rnorm(n = sample_size)
x3 = rnorm(n = sample_size)
```

```r
b2_stat = rep(0,2500)
b2_b3_stat = rep(0,2500)
for (i in 1:2500) {
  eta = b0 + b1*x1
  p = 1 / (1 + exp(-eta))
  y = rbinom(n = sample_size, size = 1, prob = p)
  data = data.frame(y, x1, x2, x3)

  fit_all = glm(y ~ ., data = data, family = binomial)
  fit_b1 = glm(y ~ x1, data = data, family = binomial)

  b2_stat[i] = summary(fit_all)$coefficients[3,3]
  b2_b3_stat[i] = anova(fit_b1, fit_all)$Deviance[2]
}

hist(b2_stat)
```



```r
hist(b2_b3_stat)
```



Based on the two histogram plots above, it seems as though a sample size of 10 is not large enough to use the standard normal and χ^2 distributions. The two histograms above do not look like they follow these distributions, the bars are too scattered in both cases.

Exercise 2 (Surviving the Titanic)

For this exercise use the ptitanic data from the rpart.plot package. (The rpart.plot package depends on the rpart package.) Use ?rpart.plot::ptitanic to learn about this data set. We will use logistic regression to help predict which passengers aboard the Titanic will survive based on various attributes.

```
# install.packages("rpart")
# install.packages("rpart.plot")
library(rpart)
library(rpart.plot)
data("ptitanic")
```

For simplicity, we will remove any observations with missing data. Additionally, we will create a test and train data set.

```
ptitanic = na.omit(ptitanic)
set.seed(42)
trn_idx = sample(nrow(ptitanic), 300)
```

```r
ptitanic_trn = ptitanic[trn_idx, ]
ptitanic_tst = ptitanic[-trn_idx, ]
```

(a) Consider the model

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_3 x_4$$

where

$$p(\mathbf{x}) = P[Y = 1 \mid \mathbf{X} = \mathbf{x}]$$

is the probability that a certain passenger survives given their attributes and

- x_1 is a dummy variable that takes the value 1 if a passenger was 2nd class.
- x_2 is a dummy variable that takes the value 1 if a passenger was 3rd class.
- x_3 is a dummy variable that takes the value 1 if a passenger was male.
- x_4 is the age in years of a passenger.

Fit this model to the training data and report its deviance.

```r
model_a = glm(survived ~ pclass + sex + age + sex*age, data = ptitanic_trn, family = binomial)
```

The deviance of this model is `259.4409`.

(b) Use the model fit in **(a)** and an appropriate statistical test to determine if class played a significant role in surviving on the Titanic. Use $\alpha = 0.01$. Report:

- The null hypothesis of the test
- The test statistic of the test
- The p-value of the test
- A statistical decision
- A practical conclusion

The null hypothesis of the test is $H_0 : \beta_1 = \beta_2 = 0$.

```r
model_no_class = glm(survived ~ sex + age + sex*age, data = ptitanic_trn, family = binomial)
test = anova(model_no_class, model_a, test = "LRT")
test
```

```
## Analysis of Deviance Table
##
## Model 1: survived ~ sex + age + sex * age
## Model 2: survived ~ pclass + sex + age + sex * age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       296        299
## 2       294        259  2     39.7  2.4e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test statistic (deviance) of the test is 39.6786.

The p-value of the test is \ensuremath{2.4205\times 10^{-9}}.

A statistical decision at $\alpha = 0.01$ is to reject the null hypothesis.

Based on the test, class seems to have played a significant role in predicting whether a person survived or not on the Titanic.

(c) Use the model fit in **(a)** and an appropriate statistical test to determine if an interaction between age and sex played a significant role in surviving on the Titanic. Use $\alpha = 0.01$. Report:

- The null hypothesis of the test
- The test statistic of the test
- The p-value of the test
- A statistical decision
- A practical conclusion

```r
summary(model_a)$coefficients[6,]
```

```
##   Estimate Std. Error    z value   Pr(>|z|)
##  -0.064711   0.024360  -2.656480   0.007896
```

The null hypothesis of the test is $H_0 : \beta_5 = 0$.

The test statistic (z-value) of the test is -2.6565.

The p-value of the test is 0.0079.

A statistical decision at $\alpha = 0.01$ is to reject the null hypothesis.

The interaction between age and sex played a significant role in surviving on the Titanic.

(d) Use the model fit in **(a)** as a classifier that seeks to minimize the misclassification rate. Classify each of the passengers in the test data set. Report the misclassification rate, the sensitivity, and the specificity of this classifier. (Use survived as the positive class.)

```r
pred_te = predict(model_a, newdata = ptitanic_tst)
class_te = ifelse(pred_te > 0, "survived", "died")

# misclassification rate
mean(class_te != ptitanic_tst$survived)
```

```
## [1] 0.2212
```

```r
# sensitivity
make_conf_mat = function(predicted, actual) {
  table(predicted = predicted, actual = actual)
}
get_sens = function(conf_mat) {
  conf_mat[2, 2] / sum(conf_mat[, 2])
}
conf_mat = make_conf_mat(predicted = class_te, actual = ptitanic_tst$survived)
get_sens(conf_mat)
```

```
## [1] 0.6444
```

```r
# specificity
get_spec =  function(conf_mat) {
  conf_mat[1, 1] / sum(conf_mat[, 1])
}
get_spec(conf_mat)
```

```
## [1] 0.877
```

Exercise 3 (Breast Cancer Detection)

For this exercise we will use data found in `wisc-train.csv` and `wisc-test.csv`, which contain train and test data, respectively. `wisc.csv` is provided but not used. This is a modification of the Breast Cancer Wisconsin (Diagnostic) data set from the UCI Machine Learning Repository. Only the first 10 feature variables have been provided. (And these are all you should use.)

- UCI Page
- Data Detail

You should consider coercing the response to be a factor variable if it is not stored as one after importing the data.

(a) The response variable `class` has two levels: `M` if a tumor is malignant, and `B` if a tumor is benign. Fit three models to the training data.

- An additive model that uses `radius`, `smoothness`, and `texture` as predictors
- An additive model that uses all available predictors
- A model chosen via backwards selection using AIC. Use a model that considers all available predictors as well as their two-way interactions for the start of the search.

For each, obtain a 5-fold cross-validated misclassification rate using the model as a classifier that seeks to minimize the misclassification rate. Based on this, which model is best? Relative to the best, are the other two under fitting or over fitting? Report the test misclassification rate for the model you picked as the best.

```r
library(tidyverse)
wisc_tr = read_csv('wisc-train.csv')
wisc_tr$class = as.factor(wisc_tr$class)
wisc_te = read_csv('wisc-test.csv')
wisc_te$class = as.factor(wisc_te$class)

m_add_3 = glm(class ~ radius + smoothness + texture, data = wisc_tr, family = binomial)
m_add_all = glm(class ~ ., data = wisc_tr, family = binomial)
m_2way_all = glm(class ~ .+.^2, data = wisc_tr, family = binomial)
m_back_aic = step(m_2way_all, direction = "backward", trace = 0)

library(boot)
set.seed(1)
cv.glm(wisc_tr, m_add_3, K = 5)$delta[1]
```

```
## [1] 0.07707
```

```
set.seed(1)
cv.glm(wisc_tr, m_add_all, K = 5)$delta[1]
```

```
## [1] 0.1171
```

```
set.seed(1)
cv.glm(wisc_tr, m_back_aic, K = 5)$delta[1]
```

```
## [1] 0.12
```

The model with the lowest 5-fold cross-validated misclassification rate is the additive model that only uses `radius`, `smoothness`, and `texture` as predictors. The misclassification rate for this model is `0.07707`. The rest of the models are all over fitting.

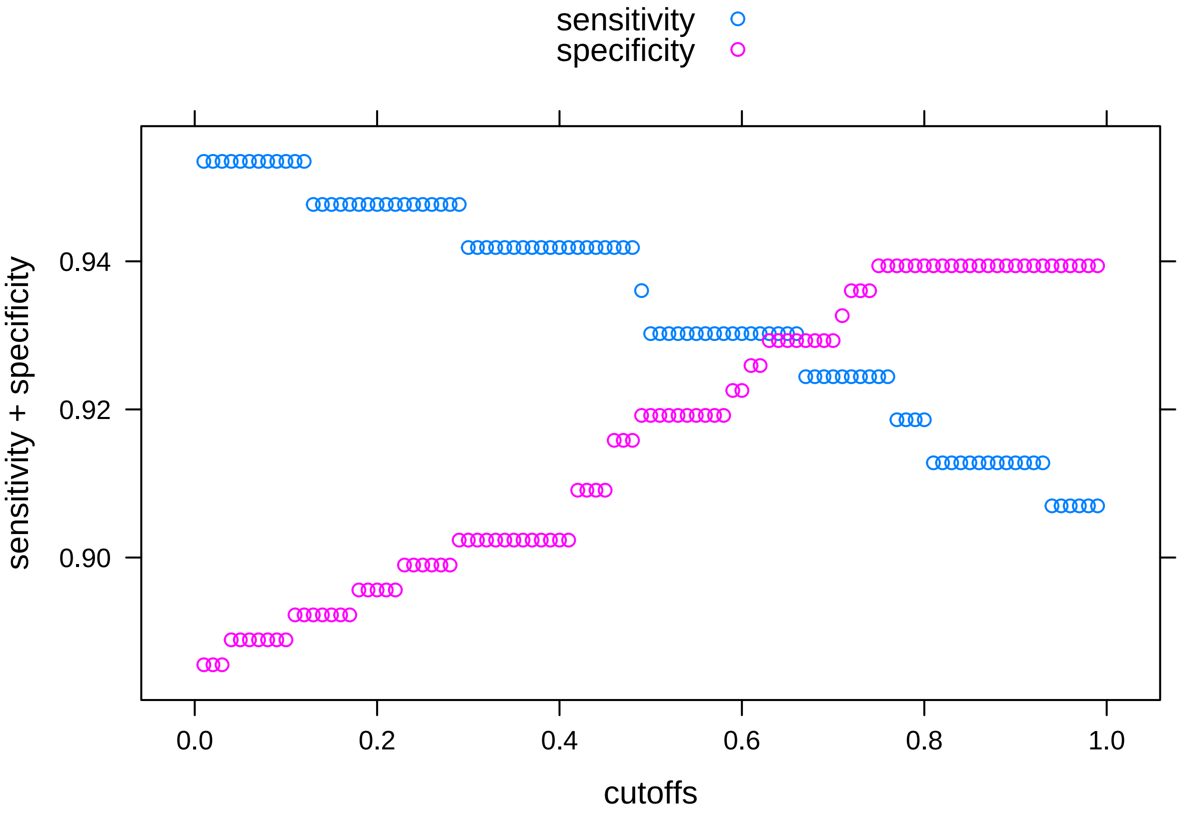
(b) In this situation, simply minimizing misclassifications might be a bad goal since false positives and false negatives carry very different consequences. Consider the `M` class as the "positive" label. Consider each of the probabilities stored in `cutoffs` in the creation of a classifier using the **additive** model fit in **(a)**.

```
cutoffs = seq(0.01, 0.99, by = 0.01)
```

That is, consider each of the values stored in `cutoffs` as c. Obtain the sensitivity and specificity in the test set for each of these classifiers. Using a single graphic, plot both sensitivity and specificity as a function of the cutoff used to create the classifier. Based on this plot, which cutoff would you use? (0 and 1 have not been considered for coding simplicity. If you like, you can instead consider these two values.)

$$\hat{C}(\mathbf{x}) = \begin{cases} 1 & \hat{p}(\mathbf{x}) > c \\ 0 & \hat{p}(\mathbf{x}) \leq c \end{cases}$$

```
pred_te = predict(m_add_3, newdata = wisc_te)

sens = rep(0, length(cutoffs))
spec = rep(0, length(cutoffs))
for (i in 1:length(cutoffs)) {
  cutoff = cutoffs[i]
  class_te = ifelse(pred_te > cutoff, "M", "B")
  conf_mat = make_conf_mat(predicted = class_te, actual = wisc_te$class)

  sens[i] = get_sens(conf_mat)
  spec[i] = get_spec(conf_mat)
}

library(lattice)
data = data.frame(cutoffs, sensitivity = sens, specificity = spec)
xyplot(sensitivity + specificity ~ cutoffs, data, auto.key=TRUE)
```



Sensitivity is essentially the true positive rate, so when sensitivity is high, the number of false negatives is low. Specificity is essentially the true negative rate, so when specificity is high, the number of false positives is low. The response variable, `class`, has two levels. `M` if the tumor is malignant, and `B` if the tumor is benign. In this context, I believe that a false negative, or identifying a tumor as benign when it is actually malignant, would carry a heavier consequence than a false positives. So looking at the plot above, I'd prefer cutoffs below about `0.65`, because as sensitivity rises, the number of false negatives decrease.