Week 10 - Homework

STAT 420, Summer 2019, Sushanta Panda

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

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.

```
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:

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

```
beta_0 = 0.4
beta_1 = -0.35

wald_test = rep(0,2500)
liklihood_test = rep(0,2500)
for(i in (1:2500)){
  eta = beta_0 + beta_1 * x1 #simulate from this model
  p = ( 1/(1+exp(-eta)) )
  y = rbinom(n = sample_size, size = 1, p)
  model_full = glm(y ~ x1 + x2 + x3, family = binomial)
  model_x1 = glm(y ~ x1, family = binomial)
  wald_test[i] = summary(model_full)$coefficient[3,3]
  liklihood_test[i] = anova(model_x1,model_full,test="LRT")[2,4]
}
```

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

  ```
  head(wald_test,10)
  ```

  ```
  ##  [1] -0.54079  0.33254  0.51541  0.12578  1.00581  0.17232 -2.00006  0.548
  99
  ##  [9] -0.01783 -0.81418
  ```

- 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

  ```
  head(liklihood_test,10)
  ```

  ```
  ##  [1] 0.34330 1.49612 0.47319 0.18610 1.31068 1.49238 4.47852 1.06730 0.017
  53
  ## [10] 0.66959
  ```

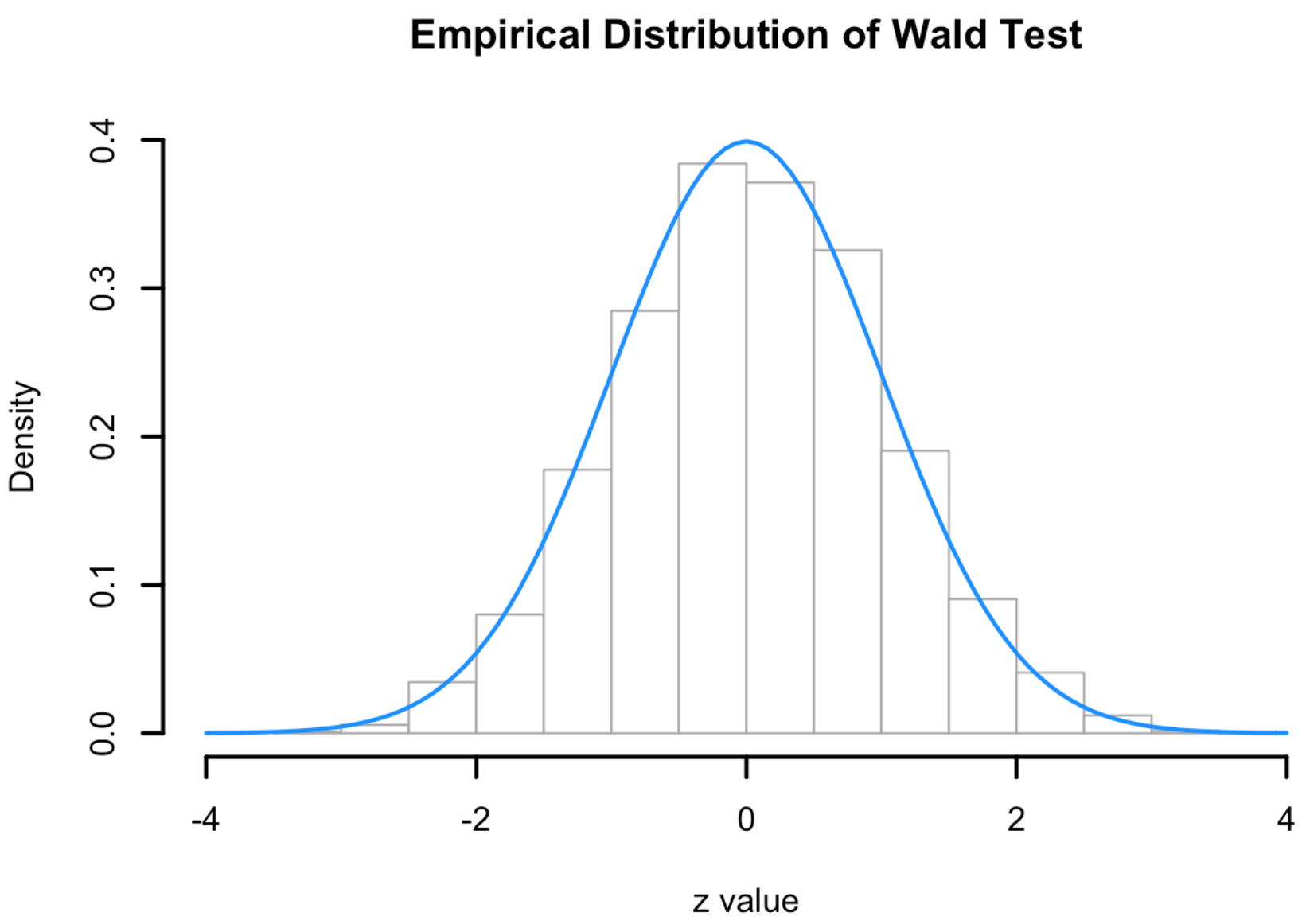
(b) Plot a histogram of the empirical values for the Wald test statistic. Overlay the density of the true distribution assuming a large sample.

```
sample_data = seq(-4,4,length = 1000)
sample_data_norm = dnorm(sample_data)

x = seq(-4,4,length = 1000)

hist(wald_test,
     probability = TRUE,
     xlim = c(-4,4) ,
     ylim = c(0,0.4) ,
     xlab = "z value" ,
     col="white",
     border="darkgrey",
     lwd = 2,
     main = "Empirical Distribution of Wald Test"
     )

curve(dnorm(x),
      lwd=2,
      col = "dodgerblue",
      add = TRUE)
```



(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.

```
mean (wald_test > 1)
```

```
## [1] 0.168
```

The empirical results for the Wald test statistic to estimate the probability of observing a test statistic larger than 1 is **0.168**

```
pnorm(1, mean = 0, sd = 1, lower.tail = FALSE)
```

```
## [1] 0.1587
```

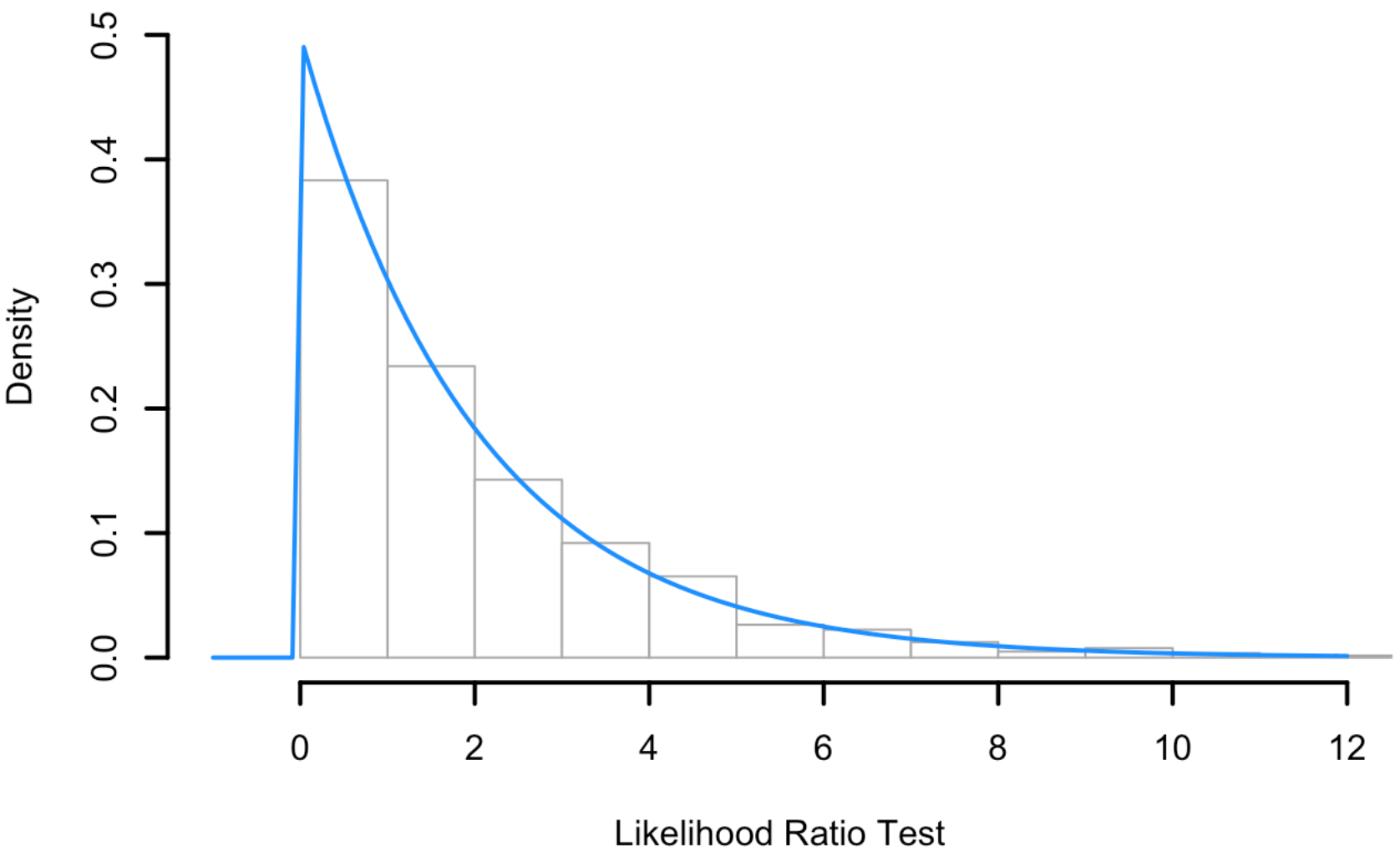
The probability using the true distribution of the **wald test** statistic assuming a large sample is : **0.1587**

(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.

```
x = seq(-4, 4 , length = 1000)

hist(liklihood_test,
     xlim = c(-1,12) ,
     xlab = "Likelihood Ratio Test" ,
     ylim = c(0,0.5) ,
     probability = TRUE,
     col="white",
     border="darkgrey",
     lwd = 2,
     main = "Empirical Distribution of Likelihood Ratio Test"
     )

curve(dchisq(x, df = 2),
      lwd=2,
      col = "dodgerblue",
      add = TRUE)
```



(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.

```
mean (liklihood_test > 5)
```

```
## [1] 0.0828
```

The empirical results for the **likelihood ratio test statistic** to estimate the probability of observing a test statistic larger than 5 is **0.0821**

```
pchisq(5, df = 2, ncp = 0, lower.tail = FALSE, log.p = FALSE)
```

```
## [1] 0.08208
```

The probability using the true distribution of the **likelihood ratio test statistic** statistic assuming a large sample is : **0.0821**

(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.

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

Simulate, for sample size = 10

```
beta_0 = 0.4
beta_1 = -0.35

wald_test = rep(0,2500)
liklihood_test = rep(0,2500)
for(i in (1:2500)){
  eta = beta_0 + beta_1 * x1 #simulate from this model
  p = ( 1/(1+exp(-eta)) )
  y = rbinom(n = sample_size, size = 1, p)
  model_full = glm(y ~ x1 + x2 + x3, family = binomial)
  model_x1 = glm(y ~ x1, family = binomial)
  wald_test[i] = summary(model_full)$coefficient[3,3]
  liklihood_test[i] = anova(model_x1,model_full,test="LRT")[2,4]
}
```

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 "small" samples (Size = 10)

```
head(wald_test,10)
```

```
##  [1] -0.9632828  1.0282954  0.0672655  0.0004000  0.0003410  0.4702837
##  [7] -0.2076200  1.4628381 -1.3552942  0.0005961
```

- 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 "small" samples (Size = 10)

```
head(liklihood_test,10)
```

```
##  [1]  2.3746  6.5867  0.7800 10.1387  4.3157  0.2895  0.4918  3.0071  5.53
63
## [10] 13.8126
```

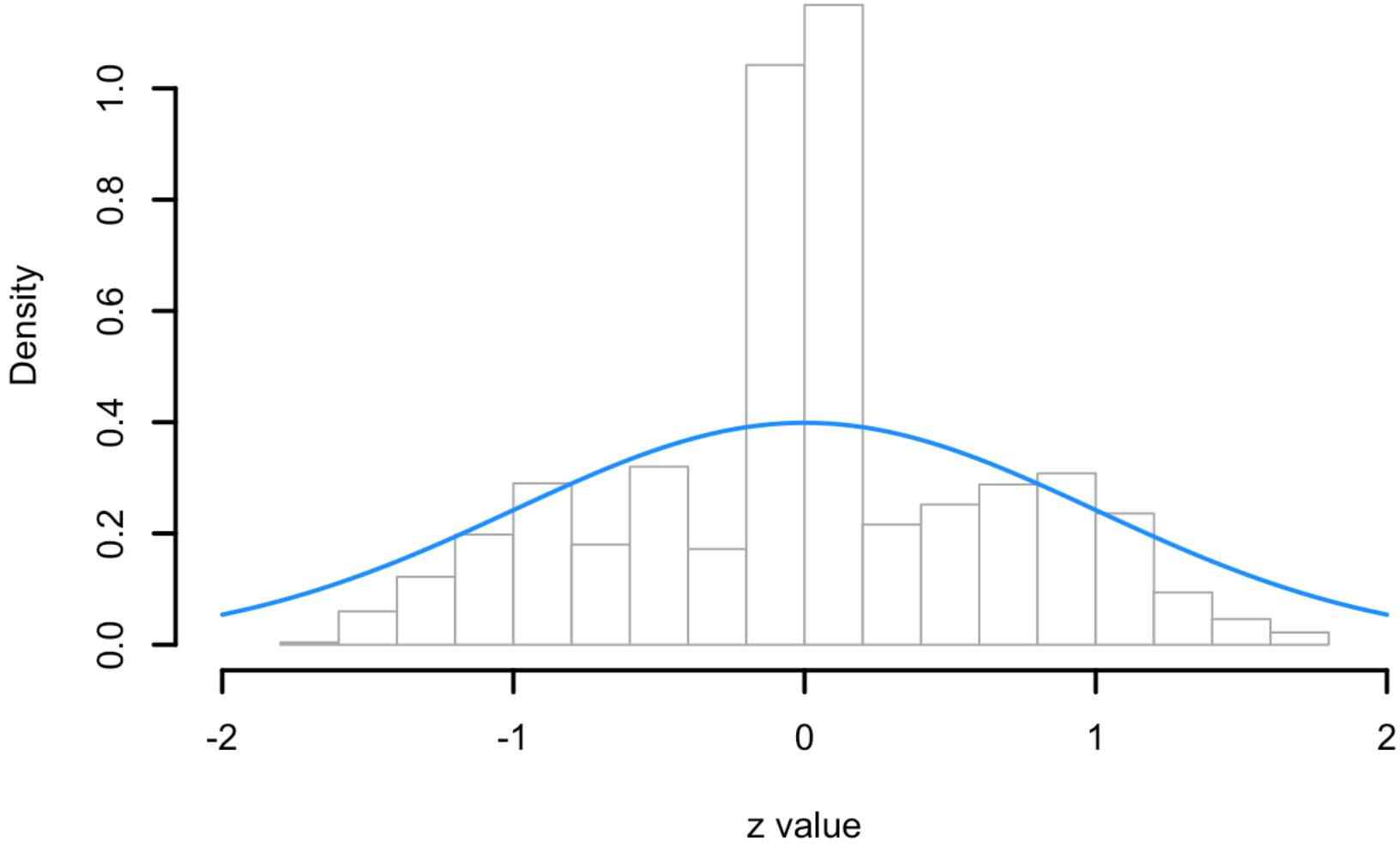
Plot a histogram of the empirical values for the Wald test statistic. Overlay the density of the true distribution assuming a small sample (Size = 10)

```
sample_data = seq(-4,4,length = 1000)
sample_data_norm = dnorm(sample_data)

x = seq(-4,4,length = 1000)

hist(wald_test,
     probability = TRUE,
     xlim = c(-2,2) ,
     #ylim = c(0,0.4) ,
     xlab = "z value" ,
     col="white",
     border="darkgrey",
     lwd = 2,
     main = "Empirical Distribution of Wald Test"
     )

curve(dnorm(x),
      lwd=2,
      col = "dodgerblue",
      add = TRUE)
```



Comments

It seems with lower the sample size, the wald test statistics distribution **doesn't follow a normal distribution**. This is because of the fact that, lower the number of sample size is not enough to produce the correct empirical distribution of the wald test distribution.

Empirical results for the Wald test statistic to estimate the probability of observing a test statistic larger than 1

```
mean (wald_test > 1)
```

```
## [1] 0.0796
```

The empirical results for the Wald test statistic to estimate the probability of observing a test statistic larger than 1 is **0.0796**

Report the probability using the true distribution of the test statistic assuming a small sample (Size = 10)

```
pnorm(1, mean = 0, sd = 1, lower.tail = FALSE)
```

```
## [1] 0.1587
```

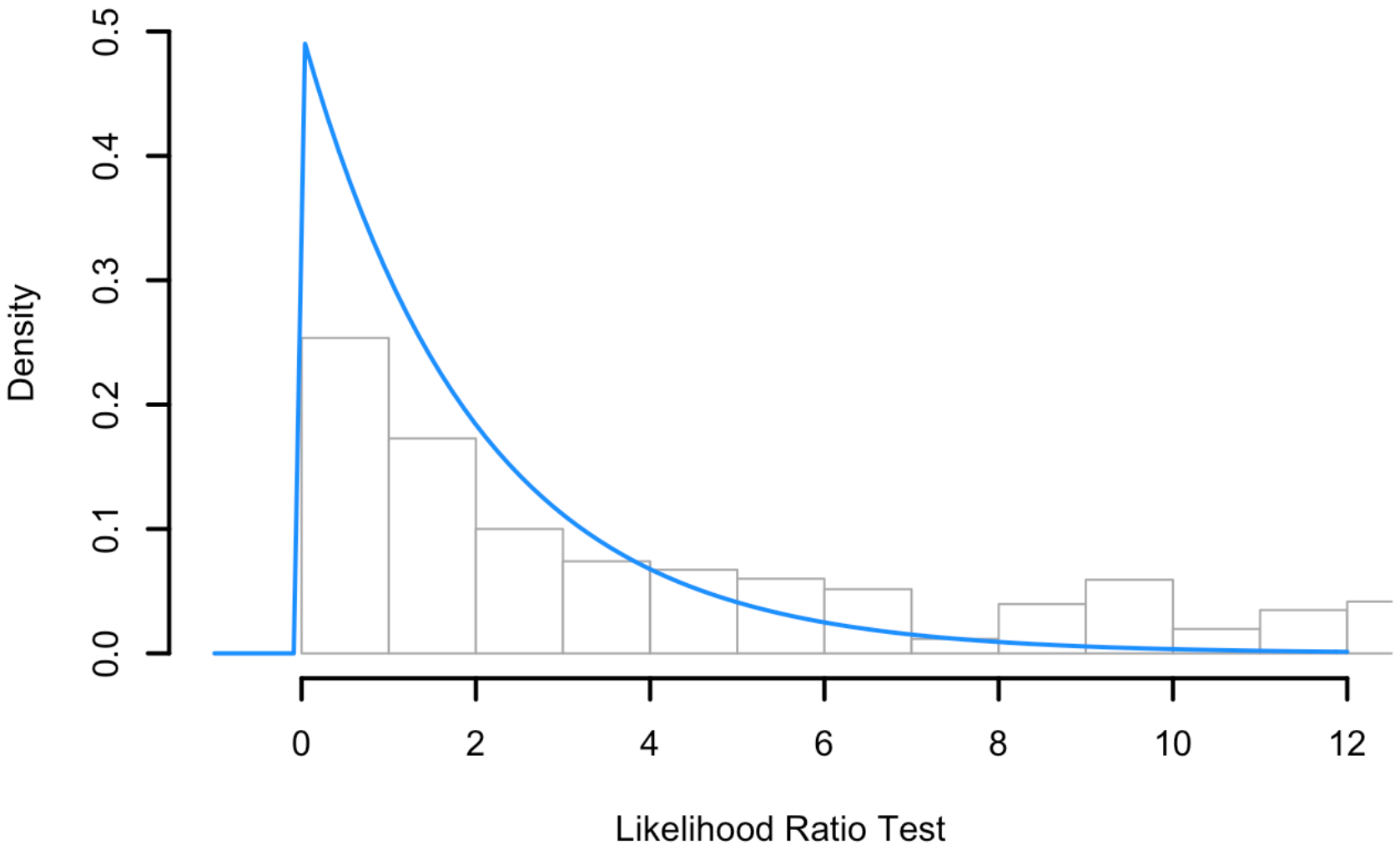
The probability using the true distribution of the **wald test** statistic assuming a large sample is : **0.1587**

Plot a histogram of the empirical values for the likelihood ratio test statistic. Overlay the density of the true distribution assuming a small sample (Size = 10)

```
x = seq(-4, 4 , length = 1000)

hist(liklihood_test,
     xlim = c(-1,12) ,
     xlab = "Likelihood Ratio Test" ,
     ylim = c(0,0.5) ,
     probability = TRUE,
     col="white",
     border="darkgrey",
     lwd = 2,
     main = "Empirical Distribution of Likelihood Ratio Test"
     )

curve(dchisq(x, df = 2),
      lwd=2,
      col = "dodgerblue",
      add = TRUE)
```



Comments

In the same way, the Likelihood Ratio test statistics distribution **doesn't follow a χ2 distribution**. This is because of the fact that, lower the number of sample size is not enough to produce the correct empirical distribution of the Likelihood test distribution.

Empirical results for the likelihood ratio test statistic to estimate the probability of observing a test statistic larger than 5

```
mean (liklihood_test > 5)
```

```
## [1] 0.3324
```

The empirical results for the **likelihood ratio test statistic** to estimate the probability of observing a test statistic larger than 5 is **0.0821**

Report this probability using the true distribution of the test statistic assuming a small sample (Size = 10)

```
pchisq(5, df = 2, ncp = 0, lower.tail = FALSE, log.p = FALSE)
```

```
## [1] 0.08208
```

The probability using the true distribution of the **likelihood ratio test statistic** statistic assuming a large sample is : **0.0821**

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 dataset. We will use logistic regression to help predict which passengers aboard the Titanic (https://en.wikipedia.org/wiki/RMS_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 dataset.

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

```
model = glm(survived ~ pclass + sex + age + sex:age, data = ptitanic_trn, family =
binomial)
summary(model)
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sex:age, family = binomial,
##     data = ptitanic_trn)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.276  -0.668  -0.431   0.643   2.451
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.3415     0.6749    3.47  0.00052 ***
## pclass2nd    -0.9500     0.4416   -2.15  0.03148 *
## pclass3rd    -2.1269     0.4235   -5.02  5.1e-07 ***
## sexmale      -1.0175     0.6494   -1.57  0.11715
## age           0.0027     0.0166    0.16  0.87064
## sexmale:age  -0.0505     0.0203   -2.49  0.01291 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 408.81  on 299  degrees of freedom
## Residual deviance: 281.50  on 294  degrees of freedom
## AIC: 293.5
##
## Number of Fisher Scoring iterations: 4
```

The deviance is: **281.4987**

(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

$$H_0 : \beta_1 = \beta_2 = 0$$

- The test statistic of the test

  Since there are 2 parameters to test the significance and are nested, we will do the **likelihood ratio test statistic** by creating the null and full model as below

  Full and null model

```
full_model = glm(survived ~ pclass + sex + age + sex:age, data = ptitanic_trn
, family = binomial)
full_model
```

```
##
## Call:  glm(formula = survived ~ pclass + sex + age + sex:age, family = bin
omial,
##     data = ptitanic_trn)
##
## Coefficients:
## (Intercept)    pclass2nd    pclass3rd      sexmale          age  sexmale:a
ge
##      2.3415      -0.9500      -2.1269      -1.0175       0.0027      -0.05
05
##
## Degrees of Freedom: 299 Total (i.e. Null);  294 Residual
## Null Deviance:       409
## Residual Deviance: 281   AIC: 293
```

```
null_model = glm(survived ~ sex + age + sex:age, data = ptitanic_trn, family
= binomial)
null_model
```

```
##
## Call:  glm(formula = survived ~ sex + age + sex:age, family = binomial,
##     data = ptitanic_trn)
##
## Coefficients:
## (Intercept)      sexmale          age  sexmale:age
##      0.3832      -1.0821       0.0263      -0.0462
##
## Degrees of Freedom: 299 Total (i.e. Null);  296 Residual
## Null Deviance:       409
## Residual Deviance: 312   AIC: 320
```

Likelihood Ratio Test Statistics

```
anova(null_model, full_model, test = "LRT")
```

```
## 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        312
## 2       294        282  2     30.7  2.2e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The p-value of the test

```
anova(null_model, full_model, test = "LRT")[,'Pr(>Chi)'][2]
```

```
## [1] 2.185e-07
```

The p-value of the **Likelihood Ratio Test Statistics** is : **2.184610^{-7}**

- A statistical decision

```
anova(null_model, full_model, test = "LRT")[,'Pr(>Chi)'][2] > 0.01
```

```
## [1] FALSE
```

Since the P value of the Likelihood Ratio Test Statistics is very very small 2.184610^{-7} **< 0.01 (α)**, we **reject the null hypothesis**, we reject the hypothesis that $\beta_1 = \beta_2 = 0$

- A practical conclusion

Since we **reject the null hypothesis** from the Likelihood Ratio Test Statistics, hence class played a **significant role** for surviving people on the Titanic, hence we will keep the class in the model which helps to predict the survival of the people

(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

$$H_0 : \beta_5 = 0$$

- The test statistic of the test

Since, it's a single predictor which significance needs to evaluate, we will test the **Wald test** to test the significance of the interaction between age and sex

```
full_model = glm(survived ~ pclass + sex + age + sex:age, data = ptitanic_trn
, family = binomial)
summary(full_model)
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sex:age, family = binomial,
##     data = ptitanic_trn)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.276  -0.668  -0.431   0.643   2.451
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.3415     0.6749    3.47  0.00052 ***
## pclass2nd    -0.9500     0.4416   -2.15  0.03148 *
## pclass3rd    -2.1269     0.4235   -5.02  5.1e-07 ***
## sexmale      -1.0175     0.6494   -1.57  0.11715
## age           0.0027     0.0166    0.16  0.87064
## sexmale:age  -0.0505     0.0203   -2.49  0.01291 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 408.81  on 299  degrees of freedom
## Residual deviance: 281.50  on 294  degrees of freedom
## AIC: 293.5
##
## Number of Fisher Scoring iterations: 4
```

- The p-value of the test

  ```
  summary(full_model)$coefficient['sexmale:age','Pr(>|z|)']
  ```

  ```
  ## [1] 0.01291
  ```

  The p-value of the **Wald Test** is : **0.0129**

- A statistical decision

  Since the p-value from the **wald test** is **0.0129** > 0.01 (α), we **failed to reject the significance of interaction between age & sex**, means we keep $\beta_5 = 0$

- A practical conclusion

  Since we **failed to reject** $\beta_5 = 0$, the interaction of age and sex is not significance, hence the interaction between Age and Sex is ** not significant role** for surviving people on the Titanic, should excluded from the model

(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 dataset. Report the misclassification rate, the sensitivity, and the specificity of this classifier. (Use survived as the positive class.)

Fitting the Model from point (a)

```
model = glm(survived ~ pclass + sex + age + sex:age, data = ptitanic_trn, family =
binomial)
```

Clasify each passenger in the PTitanic Test Dataset

```
test_predict = ifelse(predict(model,ptitanic_tst) > 0 , "survived", "died")
head(test_predict,10)
```

```
##          5          6          7          9         10         11         12
## "survived"     "died" "survived" "survived"     "died"     "died" "survived"
##         13         14         17
## "survived" "survived" "survived"
```

Mis-classification rate from PTitanic Test Dataset

```
#testing mis-classification rate
mean(test_predict != ptitanic_tst$survived)
```

```
## [1] 0.2118
```

The mis-classification rate from the ptitanic test datset is : **0.2118**

Function for Confusion matrix, Sensitivity and Specificity

```
#Confusion matrix creation
make_conf_mat = function(predicted, actual){
  table(predicted = predicted, actual = actual)
}

#Sensitivity Function
get_sens = function(conf_mat){
  conf_mat[2, 2] / sum(conf_mat[, 2])
}

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

Sensitivity from PTitanic Test Dataset

```
conf_mat_predict = make_conf_mat(predicted = test_predict, actual = ptitanic_tst$s
urvived)
get_sens(conf_mat_predict)
```

```
## [1] 0.7333
```

The sensitivity of the Test dataset is : **0.7333**

Specificity from PTitanic Test Dataset

```
conf_mat_predict = make_conf_mat(predicted = test_predict, actual = ptitanic_tst$s
urvived)
get_spec(conf_mat_predict)
```

```
## [1] 0.8251
```

The specificity of the Test dataset is : **0.8251**

Exercise 3 (Breast Cancer Detection)

For this exercise we will use data found in `wisc-train.csv` (wisc-train.csv) and `wisc-test.csv` (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) dataset from the UCI Machine Learning Repository. Only the first 10 feature variables have been provided. (And these are all you should use.)

- UCI Page (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))
- Data Detail (https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names)

```
wisc_train = read.csv("wisc-train.csv")
wisc_test = read.csv("wisc-test.csv")
```

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

```
model1 = glm(class ~ radius + smoothness + texture, data = wisc_train, family
= binomial)
model1
```

```
##
## Call:  glm(formula = class ~ radius + smoothness + texture, family = binom
ial,
##     data = wisc_train)
##
## Coefficients:
## (Intercept)       radius   smoothness      texture
##     -41.714        1.419      132.100        0.464
##
## Degrees of Freedom: 99 Total (i.e. Null);  96 Residual
## Null Deviance:       135
## Residual Deviance: 32.8  AIC: 40.8
```

- An additive model that uses all available predictors

```
model2 = glm(class ~ ., data = wisc_train, family = binomial)
model2
```

```
##
## Call:  glm(formula = class ~ ., family = binomial, data = wisc_train)
##
## Coefficients:
## (Intercept)       radius      texture    perimeter         area   smoothne
ss
##   -39.41377      5.84646      0.64761     -0.86056      0.00766    -89.612
04
## compactness    concavity      concave     symmetry      fractal
##   -14.47072    -60.04595    317.42464    -12.12505    346.32988
##
## Degrees of Freedom: 99 Total (i.e. Null);  89 Residual
## Null Deviance:       135
## Residual Deviance: 22    AIC: 44
```

- 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.

```
model = glm(class ~ . + . ^ 2, data = wisc_train, family = binomial)
model3 = step(model, direction = "backward", trace = 0)
model3
```

```
##
## Call:  glm(formula = class ~ radius + texture + perimeter + concavity +
##     concave + symmetry + fractal + radius:concavity + concavity:symmetry +
##     concave:symmetry, family = binomial, data = wisc_train)
##
## Coefficients:
##        (Intercept)              radius             texture           perim
eter
##             -2944.6               665.8                22.2              -
97.6
##          concavity             concave            symmetry              fra
ctal
##           -44214.2             62902.9              -106.7              294
23.2
##   radius:concavity  concavity:symmetry    concave:symmetry
##             1207.0            149158.8           -312420.2
##
## Degrees of Freedom: 99 Total (i.e. Null);  89 Residual
## Null Deviance:       135
## Residual Deviance: 1.27e-07  AIC: 22
```

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 underfitting or over fitting? Report the test misclassification rate for the model you picked as the best.

```r
#Cross-Validation
library(boot)
set.seed(42)
cv_model1 = cv.glm(wisc_train,model1, K = 5)$delta[1]
cv_model1
```

```
## [1] 0.06627
```

```r
set.seed(42)
cv_model2 = cv.glm(wisc_train,model2, K = 5)$delta[1]
cv_model2
```

```
## [1] 0.08257
```

```r
set.seed(42)
cv_model3 = cv.glm(wisc_train,model3, K = 5)$delta[1]
cv_model3
```

```
## [1] 0.1212
```

```r
library(knitr)
cv_model = data.frame(
  Model = c("Model 1", "Model 2", "Model 3"),
  Fold_5_CV_Score = c(cv_model1,cv_model2,cv_model3)
)
kable(cv_model, format = "pandoc",padding = 2)
```

Model	Fold_5_CV_Score
Model 1	0.0663
Model 2	0.0826
Model 3	0.1212

From the above, it seems that the best model is **`model 1`** , that is the additive model that uses `radius` , `smoothness` , and `texture` as predictors, because the 5 fold cross validation mis-classification error rate is **0.0663** (**which is the lowest among the 3 models**)

Relative to the best model (**`model 1`**), it seems both the model (**`model 2`** and **`model 3`**) are overfitting. The reason behind is, both the models (**`model 2`** and **`model 3`**) have higher number of parameters (**`model 2`** number of parameters = **11**, **`model 3`** number of parameters = **11**) as compared to the the best model (`Model 1` number of parameter = **4**), also have high 5 fold cross validation (**`model 2`** 5 fold cv score = 0.0826, **`model 3`** number of parameters = 0.1212) as compared to the best model **`model 1`** whose 5 fold cv score = **0.0663**.

Best Model (Model 1) Misclassification Rate

```
mean(ifelse(predict(model1,wisc_test) > 0 , "M", "B") != wisc_test$class)
```

```
## [1] 0.08955
```

The best model (i.e. **`Model 1`**) Test Misclassification rate is **0.0896**

(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.)

Function for Confusion Matrix, Sensitivity, Specificity

```
#Confustion matrix creation
make_conf_mat = function(predicted, actual){
  table(predicted = predicted, actual = actual)
}

#Sensitivity Function
get_sens = function(conf_mat){
  conf_mat[2, 2] / sum(conf_mat[, 2])
}

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

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

Additive Model

```
model_add = glm(class ~ ., data = wisc_train, family = binomial)
```

```
sens_99 = seq(1,length(cutoffs))
spec_99 = seq(1,length(cutoffs))
for(i in (1:length(cutoffs))){
  test_predict = ifelse(predict(model_add, wisc_test, type = "response") > cutoffs
[i] , "M", "B")
  conf_mat_predict = make_conf_mat(predicted = test_predict, actual = wisc_test$cl
ass)
  sens_99[i] = get_sens(conf_mat_predict)
  spec_99[i] = get_spec(conf_mat_predict)
}
```

```
head(sens_99, 10)
```

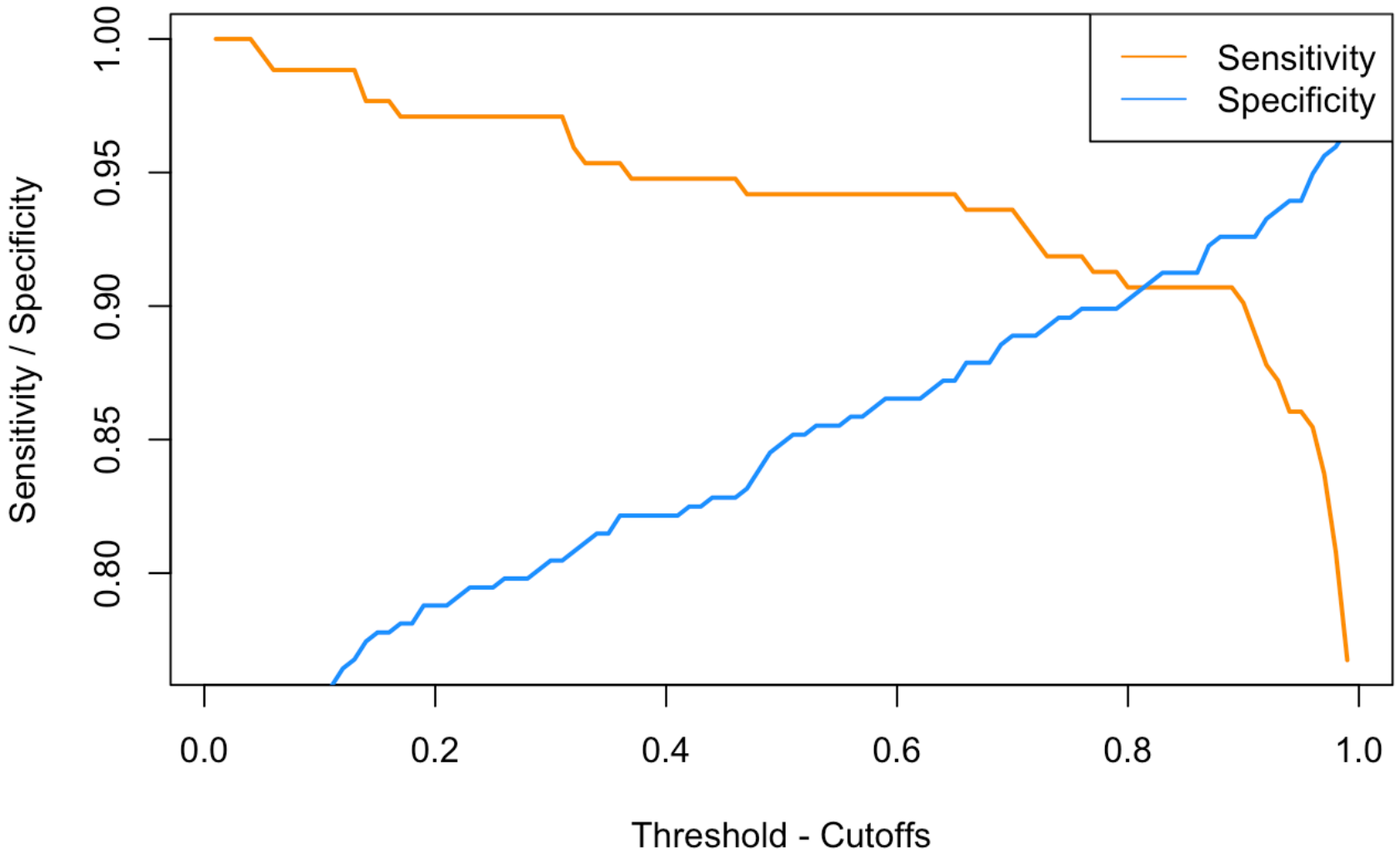
```
##  [1] 1.0000 1.0000 1.0000 1.0000 0.9942 0.9884 0.9884 0.9884 0.9884 0.9884
```

```
head(spec_99, 10)
```

```
##  [1] 0.5758 0.6296 0.6700 0.6869 0.7104 0.7273 0.7306 0.7340 0.7374 0.7475
```

Plot all Specificity and all Sensitivity for all Thresholds Value

```
plot(sens_99~cutoffs,type = "l",col = "darkorange", lwd = 2, lty = 1, xlab = "Thre
shold - Cutoffs", ylab = "Sensitivity / Specificity")
lines(spec_99~cutoffs,type = "l",col = "dodgerblue", lwd = 2, lty = 1)
legend("topright", legend = c("Sensitivity", "Specificity"), col=c("darkorange","d
odgerblue"),lty = c(1,1))
```




```
which.min(abs(sens_99 - spec_99))
```

```
## [1] 81
```

```
cutoffs[which.min(abs(sens_99 - spec_99))]
```

```
## [1] 0.81
```

Based on the above plot, it seems that the best threshold to clasify the Model is **0.81**, where the Sensitivity crossed the Specificitiy. This is point where the Sensitivity is minimum and specificity is maximum at its best