# Math 70 Homework 6

Alex Craig

## Problem 1

### Instructions

(Problem 5.5 pg. 309) Let $X$ and $Y$ be two independent normally distributed random variables with means $\mu_X$ and $\mu_Y$ and variances $\sigma_X^2$ and $\sigma_Y^2$.

**(a)** Prove that a necessary condition for $X \prec Y$ is that $\mu_X < \mu_Y$.

**(b)** Provide an example when $\mu_X < \mu_Y$ but $X \not\prec Y$.

**(c)** Prove that if $\sigma_X = \sigma_Y$, then $\mu_X < \mu_Y$ is a necessary and sufficient condition for $X \prec Y$.

### Solution

**(a)** To prove that a necessary condition for $X \prec Y$ is that $\mu_X \leq \mu_Y$, we'll first consider the definition of uniformly smaller. A random variable $X$ is said to be uniformly smaller than $Y$, denoted by $X \prec Y$, if for any $x$:

$$Pr(X < x) > Pr(Y < x)$$

Let $\Phi(x)$ denote the standard normal cdf. Since $X$ and $Y$ are normally distributed, we can express the above as:

$$\Phi(\frac{x - \mu_X}{\sigma_X}) > \Phi(\frac{x - \mu_Y}{\sigma_Y})$$

Let us now conduct a proof by contradiction. Assume $\mu_X \geq \mu_Y$, and let $x = \mu_X$. Then, we have:

$$\Phi(\frac{\mu_X - \mu_X}{\sigma_X}) > \Phi(\frac{\mu_X - \mu_Y}{\sigma_Y}) \Rightarrow \Phi(0) > \Phi(\frac{\mu_X - \mu_Y}{\sigma_Y})$$

Because $\Phi(x)$ is an increasing function, this implies that:

$$0 > \frac{\mu_X - \mu_Y}{\sigma_Y}$$

We know that $\sigma_Y$ is positive, so therefore:

$$0 > \mu_X - \mu_Y \Rightarrow \mu_X < \mu_Y$$

We have a contradiction with our original assumption that $\mu_C \geq \mu_Y$ Therefore, given that $X \prec Y$, we know that $\mu_X \not\geq \mu_Y$, and therefore $\mu_X < \mu_Y$. Therefore, $\mu_X < \mu_Y$ is a necessary condition for $X \prec Y$.

**(b)** In this example, let:

$$X \sim \mathcal{N}(\mu_X = 0, \sigma_X^2 = 4)$$

$$Y \sim \mathcal{N}(\mu_Y = 1, \sigma_Y^2 = 1)$$

In this example it is true that $\mu_X < \mu_Y$, but let's check if $X \prec Y$:

$$Pr(X < x) = \Phi(\frac{x - 0}{\sqrt{4}}) = \Phi(\frac{x}{2})$$

$$Pr(Y < x) = \Phi(\frac{x - 1}{\sqrt{1}}) = \Phi(x - 1)$$

Let $x = 6$. Then, we have:

$$Pr(X < 6) = \Phi(3)$$

$$Pr(Y < 6) = \Phi(5)$$

Because $\Phi(x)$ is an increasing function, we can see that:

$$\Phi(3) < \Phi(5) \Rightarrow Pr(X < 6) < Pr(Y < 6) \Rightarrow X \not\prec Y$$

And therefore, $X \not\prec Y$.

**(c)** From part (a), we know that $X \prec Y$ is equivalent to:

$$\Phi(\frac{x - \mu_X}{\sigma_X}) > \Phi(\frac{x - \mu_Y}{\sigma_Y})$$

If we let $\sigma_X = \sigma_Y = \sigma$, then we have:

$$\Phi(\frac{x - \mu_X}{\sigma}) > \Phi(\frac{x - \mu_Y}{\sigma})$$

Because $\Phi(x)$ is an increasing function, this implies that:

$$\frac{x - \mu_X}{\sigma} > \frac{x - \mu_Y}{\sigma} \Rightarrow x - \mu_X > x - \mu_Y \Rightarrow \mu_X < \mu_Y$$

Therefore, given that $\sigma_X = \sigma_Y = \sigma$, we know that $\frac{x-\mu_X}{\sigma} > \frac{x-\mu_Y}{\sigma}$ holds true if and only if $\mu_X < \mu_Y$. Therefore, $\mu_X < \mu_Y$ is a necessary and sufficient condition for $X \prec Y$ when $\sigma_X = \sigma_Y$.

# Problem 2

## Instructions

File `bp.csv` contains blood pressure (BP) for normal patients (controls, high=0) and hypertension patients (high=1).

**(a)** Display two cdfs to demonstrate that BP among normal patients is uniformly smaller than among hypertension patients.

**(b)** Display the data-driven ROC curve for the identification of normal patients and the superimposed binormal counterpart.

**(c)** Compute and display AUCs using three methods: (1) empirical, as the sum or rectangles, (2) empirical, using vectorized computation, and (3) theoretical, using the formula.

**(d)** The cost associated with overlooking a hypertension patient is $10K and the cost of the false identification of hypertension is $1K. Display the data-driven total cost and the superimposed continuous counterpart as a function of the threshold along with the respective optimal thresholds.

**(e)** Display the optimal threshold on the binormal ROC curve and the respective BP scale using `axis(side=3)`.

## Solution

### Explanation

**(a)** In this part, we are plotting two CDFs to visualize the blood pressure differences between normal patients and hypertension patients. We first split the data into two groups, normal patients and hypertension patients. Then, we calculate the empirical CDF for each group using the `ecdf()` function. Plotting these functions allows us to visually compare the distributions of BPs for the two groups. The CDF for non-hypertension patients is always higher than the CDF for hypertension patients, indicating that BP is uniformly smaller among non-hypertension patients.

**(b)** We are creating a ROC curve to visualize the performance of the binary classifier that distinguishes between normal and hypertension patients. The curve is constructed by plotting the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various threshold settings. The empirical ROC curve is derived from the actual data, while the binormal ROC curve is based on the binormal distribution, which is fitted to the actual data. Both curves help us understand the trade-off between sensitivity and specificity.

**(c)** We compute the AUC in three different ways:

1. Empirical AUC as the sum of rectangles: For each possible threshold, we calculate the rectangle's area with width as the difference in false positive rates and height as the sensitivity. We then sum all these areas.
2. Empirical AUC using vectorized computation: Instead of looping through each possible threshold, we create a long vector where each value of `Y$BP` (normal patients) is compared with each value of `X$BP` (hypertension patients). We then take the mean of this long vector as the AUC.
3. Theoretical AUC using the formula: Assuming the BPs of normal and hypertension patients each follow a normal distribution, we calculate the AUC using the cumulative distribution function of the normal distribution.

Here are the printed results:

```
AUC calculated using sum of rectangles: 0.8968544
AUC calculated using vectorized computation: 0.8970754
AUC calculated using theoretical formula: 0.8936977
```

**(d)** In this part, we calculate the total cost associated with a certain threshold value. We assume that the cost of overlooking a hypertension patient is $10K, and the cost of falsely identifying a normal patient as hypertension is $1K. The total cost is computed as the weighted sum of the cost of false positives and the cost of false negatives. By calculating the total cost for different thresholds, we can identify the optimal threshold, which minimizes the total cost. The empirical total cost is calculated directly from the data, while the binormal total cost is calculated from the fitted binormal distribution.

**(e)** In this section, we plot the ROC curves as we did in part (b). We also overlay the optimal empirical and binomial thresholds on the ROC curve, as well as the AUC found in part(c-3). This allows us to visualize the optimal threshold and the AUC on the ROC curve.

### Code

```r
### Loading Data ###

data <- read.csv("./homeworks/hw6/data/bp.csv", header = T)

### Part A ###

# Define patients with hypertension
X <- data[data$high == 1, ]
# Define patients without hypertension
Y <- data[data$high == 0, ]
```

```r
# Define the empirical cdf's
cdf_x <- ecdf(X$BP)
cdf_y <- ecdf(Y$BP)

# Define plot limits
xlim <- range(data$BP)
ylim <- c(0, 1)

# Define the x labels
x_labels <- c(
    80,
    100,
    120,
    140,
    160,
    180,
    200,
    220
)

## Plotting

# Open a png device
png("./homeworks/hw6/plots/q2a.png", width = 1600, height = 1200)

# Set up margins
par(mar = c(8, 8, 8, 8))

# Plot the empiral cdf's
plot(
    cdf_x,
    verticals = TRUE,
    do.points = FALSE,
    xlim = xlim,
    ylim = ylim,
    main = "Empirical CDF for Blood Pressure of Hypertension
    & Non-Hypertension Patients",
    cex.main = 3,
    xlab = "Blood Pressure",
    ylab = "Empirical CDF (Probability)",
    cex.lab = 2,
    cex.axis = 2,
    col = "red",
    lwd = 3,
    xaxt = "n"
)
plot(
    cdf_y,
    verticals = TRUE,
    do.points = FALSE,
    add = TRUE,
    col = "blue",
    lwd = 3
)

# Add rug plot for both hypertension & non-hypertension patients
rug(X$BP, col = "red", lwd = 2, ticksize = 0.075)
rug(Y$BP, col = "blue", lwd = 2, ticksize = 0.025)

# Add x-axis labels
```

```
axis(
    side = 1,
    at = x_labels,
    labels = x_labels,
    cex.axis = 2
)

# Add a legend
legend(
    "topleft",
    legend = c(
        "Hypertension Patients",
        "Non-Hypertension Patients"
    ),
    col = c("red", "blue"),
    lwd = 3, cex = 1.5
)

# Close the png device
dev.off()

### Part B ###

# Sort the data
data <- data[order(data$BP), ]

# Define number of patients
n <- nrow(data)

# Define vectors for sensitivity & false positive
sensitivity <- rep(0, n)
false_positive <- rep(0, n)

# Loop through each patient
for (i in 1:n) {
    # Calculate empirical sensitivity & false positive
    sensitivity[i] <- mean(Y$BP < data$BP[i])
    false_positive[i] <- mean(X$BP < data$BP[i])
}

# Define thresholds
thresholds <- seq(xlim[1], xlim[2], length.out = 1000)

# Define binomial cdf's
cdf_x_binom <- pnorm(thresholds, mean = mean(X$BP), sd = sd(X$BP))
cdf_y_binom <- pnorm(thresholds, mean = mean(Y$BP), sd = sd(Y$BP))

## Plotting

# Open a png device
png("./homeworks/hw6/plots/q2b.png", width = 1600, height = 1200)

# Set up margins
par(mar = c(8, 8, 8, 8))

# Plot the binomial ROC curve (cdf_y vs cdf_x)
plot(
    cdf_x_binom,
    cdf_y_binom,
    type = "s",
    xlim = c(0, 1),
    ylim = c(0, 1),
    main = "ROC Curve for Identification of Non-Hypertension Patients",
    cex.main = 3,
    xlab = "False Positive (1 - Specificity)",
    ylab = "Sensitivity",
    cex.lab = 2,
    cex.axis = 2,
    col = "blue",
    lwd = 3
)

# Plot the empirical ROC curve (false positive vs sensitivity)
lines(
    false_positive,
    sensitivity,
    type = "s",
    lwd = 3,
    col = "red"
)

# Add a dashed line for the diagonal
abline(
    a = 0,
    b = 1,
    lty = 2,
    lwd = 3
)

# Add a legend
legend(
    "bottomright",
    legend = c(
        "Empirical ROC Curve",
        "Binomial ROC Curve"
    ),
    col = c("blue", "red"),
    lwd = 3, cex = 1.5
)

# Close the png device
dev.off()

### Part C ###

# Part 1. (sum of rectangles)

# Define AUC
AUC <- 0

# Loop through each patient
for (i in 1:n) {
    if (i > 1) {
        width <- false_positive[i] - false_positive[i - 1]
        height <- sensitivity[i]
        AUC <- AUC + width * height
    }
}

# Print the AUC
```

```r
cat("AUC calculated using sum of rectangles:", AUC, "\n")

# Part 2. (vectorized computation)

# Define AUC
AUC <- 0

# Define the length of the vectors
Y_len <- length(Y$BP)
X_len <- length(X$BP)

# Define the long vectors
Y_long <- rep(Y$BP, times = X_len)
X_long <- rep(X$BP, times = Y_len)

# Calculate the AUC
AUC <- mean(Y_long < X_long)

# Print the AUC
cat("AUC calculated using vectorized computation:", AUC, "\n")

# Part 3. (theoretical)

# Define distribution parameters
mu_x_hat <- mean(X$BP)
mu_y_hat <- mean(Y$BP)
sigma_x_hat <- sd(X$BP)
sigma_y_hat <- sd(Y$BP)

# Define the AUC
AUC <- pnorm((mu_x_hat - mu_y_hat) / sqrt(sigma_x_hat^2 + sigma_y_hat^2))

# Print the AUC
cat("AUC calculated using theoretical formula:", AUC, "\n")

### Part D ###

# Define the cost of false positive & false negative
false_positive_weight <- 10
false_negative_weight <- 1

## Empirical

# Define an empty vector for the total cost
total_cost_emp <- rep(0, n)

# Loop through each patient
for (i in 1:n) {
    # Calculate the total cost
    total_cost_emp[i] <- false_positive_weight * false_positive[i] +
        false_negative_weight * (1 - sensitivity[i])
}

# Find all indices of the minimum total cost
min_indexes_emp <- which(total_cost_emp == min(total_cost_emp))

# Define the empirical optimal threshold
optimal_threshold_emp <- mean(data$BP[min_indexes_emp])

# Define the empirical minimum total cost
```

```
min_total_cost_emp <- mean(total_cost_emp[min_indexes_emp])

## Binomial

# Define an empty vector for binomial total cost
total_cost_binom <- rep(0, n)

# Calculate the binomial total cost
total_cost_binom <- false_positive_weight * cdf_x_binom +
    false_negative_weight * (1 - cdf_y_binom)

# Find all indices of the minimum total cost
min_indexes_binom <- which(total_cost_binom == min(total_cost_binom))

# Define the binomial optimal threshold
optimal_threshold_binom <- mean(thresholds[min_indexes_binom])

# Define the binomial minimum total cost
min_total_cost_binom <- mean(total_cost_binom[min_indexes_binom])

## Plotting

# Open a png device
png("./homeworks/hw6/plots/q2d.png", width = 1600, height = 1200)

# Set up margins
par(mar = c(8, 8, 12, 8))

# Define y-axis limits
ylim <- c(0, max(total_cost_emp))

# Plot the binomial total cost vs threshold
plot(
    thresholds,
    total_cost_binom,
    type = "l",
    xlim = xlim,
    ylim = ylim,
    main = "Total Cost vs Threshold",
    cex.main = 3,
    xlab = "Threshold (Blood Pressure)",
    ylab = "Total Cost (Thousands of Dollars)",
    cex.lab = 2,
    cex.axis = 2,
    col = "blue",
    lwd = 3,
    xaxt = "n"
)

# Plot the empirical total cost vs threshold
lines(
    data$BP,
    total_cost_emp,
    type = "s",
    col = "red",
    lwd = 3
)

# Add a blue dot for the binomial optimal threshold
points(
    optimal_threshold_binom,
    min_total_cost_binom,
    col = "blue",
    pch = 19,
    cex = 2.5
)

# Add a red dot for the empirical optimal threshold
points(
    optimal_threshold_emp,
    min_total_cost_emp,
    col = "red",
    pch = 19,
    cex = 2.5
)

# Add x-axis labels
axis(
    side = 1,
    at = x_labels,
    labels = x_labels,
    cex.axis = 2
)

# Add a legend total cost
legend(
    "bottomright",
    legend = c(
        paste("Empirical Optimal Threshold: ",
            round(optimal_threshold_emp, 0), " Blood Pressure",
            sep = ""
        ),
        paste(
            "Empirical Minimum Total Cost: $",
            round(min_total_cost_emp, 2), "K",
            sep = ""
        ),
        paste("Binomial Optimal Threshold: ",
            round(optimal_threshold_binom, 0), " Blood Pressure",
            sep = ""
        ),
        paste(
            "Binomial Minimum Total Cost: $",
            round(min_total_cost_binom, 2), "K",
            sep = ""
        )
    ),
    col = c("red", "red", "blue", "blue"),
    pch = c(19, NA, 19, NA),
    lwd = 0, cex = 1.75,
    pt.cex = c(2.5, NA, 2.5, NA)
)


# Close the png device
dev.off()

### Part E ###

## Plotting
```

```r
# Open a png device
png("./homeworks/hw6/plots/q2e.png", width = 1600, height = 1200)

# Set up margins
par(mar = c(8, 8, 12, 8))

# Plot the binomial ROC curve (cdf_y vs cdf_x)
plot(
    cdf_x_binom,
    cdf_y_binom,
    type = "s",
    xlim = c(0, 1),
    ylim = c(0, 1),
    main = "ROC Curve for Identification of Non-Hypertension Patients",
    cex.main = 3,
    xlab = "False Positive (1 - Specificity)",
    ylab = "Sensitivity",
    cex.lab = 2,
    cex.axis = 2,
    col = "blue",
    lwd = 3
)

# Plot the empirical ROC curve (false positive vs sensitivity)
lines(
    false_positive,
    sensitivity,
    type = "s",
    lwd = 3,
    col = "red"
)

# Add a blue dot for the binomial optimal threshold
points(
    cdf_x_binom[round(min_indexes_binom, 0)],
    cdf_y_binom[round(min_indexes_binom, 0)],
    col = "blue",
    pch = 19,
    cex = 2.5
)

# Add a red dot for the empirical optimal threshold
points(
    false_positive[round(min_indexes_emp, 0)],
    sensitivity[round(min_indexes_emp, 0)],
    col = "red",
    pch = 19,
    cex = 2.5
)

# Add a dashed line for the diagonal
abline(
    a = 0,
    b = 1,
    lty = 2,
    lwd = 3
)

# Define threshold label positions
```

```
threshold_labels_pos <- c(0.01, .2, .4, .6, .8, 0.99)

# Define threshold labels
threshold_labels <- round(
    qnorm(threshold_labels_pos, mean = mu_x_hat, sd = sigma_x_hat),
    0
)

# Add threshold labels
axis(
    side = 3,
    at = threshold_labels_pos,
    labels = threshold_labels,
    cex.axis = 2
)

# Add side 3 label
mtext(
    "Threshold (Blood Pressure)",
    side = 3,
    line = 3,
    cex = 2
)

# Add text in middle of plot to display AUC %
text(
    x = 0.35,
    y = 0.6,
    labels = paste("AUC = ", round(AUC, 3) * 100, "%", sep = ""),
    cex = 3
)


# Add a legend
legend(
    "bottomright",
    legend = c(
        "Empirical ROC Curve",
        "Binomial ROC Curve",
        paste("Empirical Optimal Threshold: ",
            round(optimal_threshold_emp, 0), " Blood Pressure",
            sep = ""
        ),
        paste("Binomial Optimal Threshold: ",
            round(optimal_threshold_binom, 0), " Blood Pressure",
            sep = ""
        )
    ),
    col = c("blue", "red", "red", "blue"),
    pch = c(NA, NA, 19, 19),
    lwd = c(3, 3, 0, 0), cex = 1.5
)

# Close the png device
dev.off()
```

**Plots**

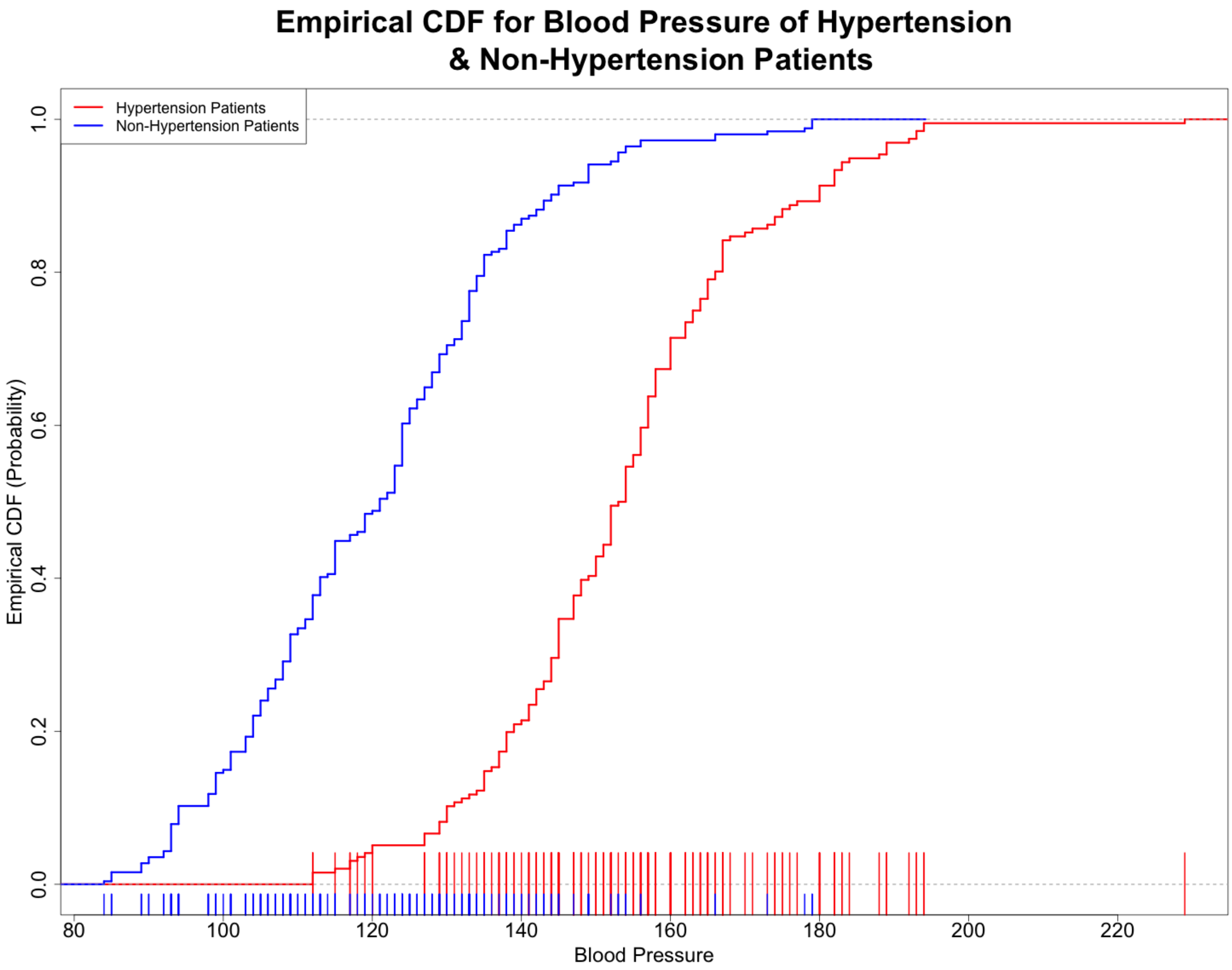

Figure 1: (a) Empirical CDFs

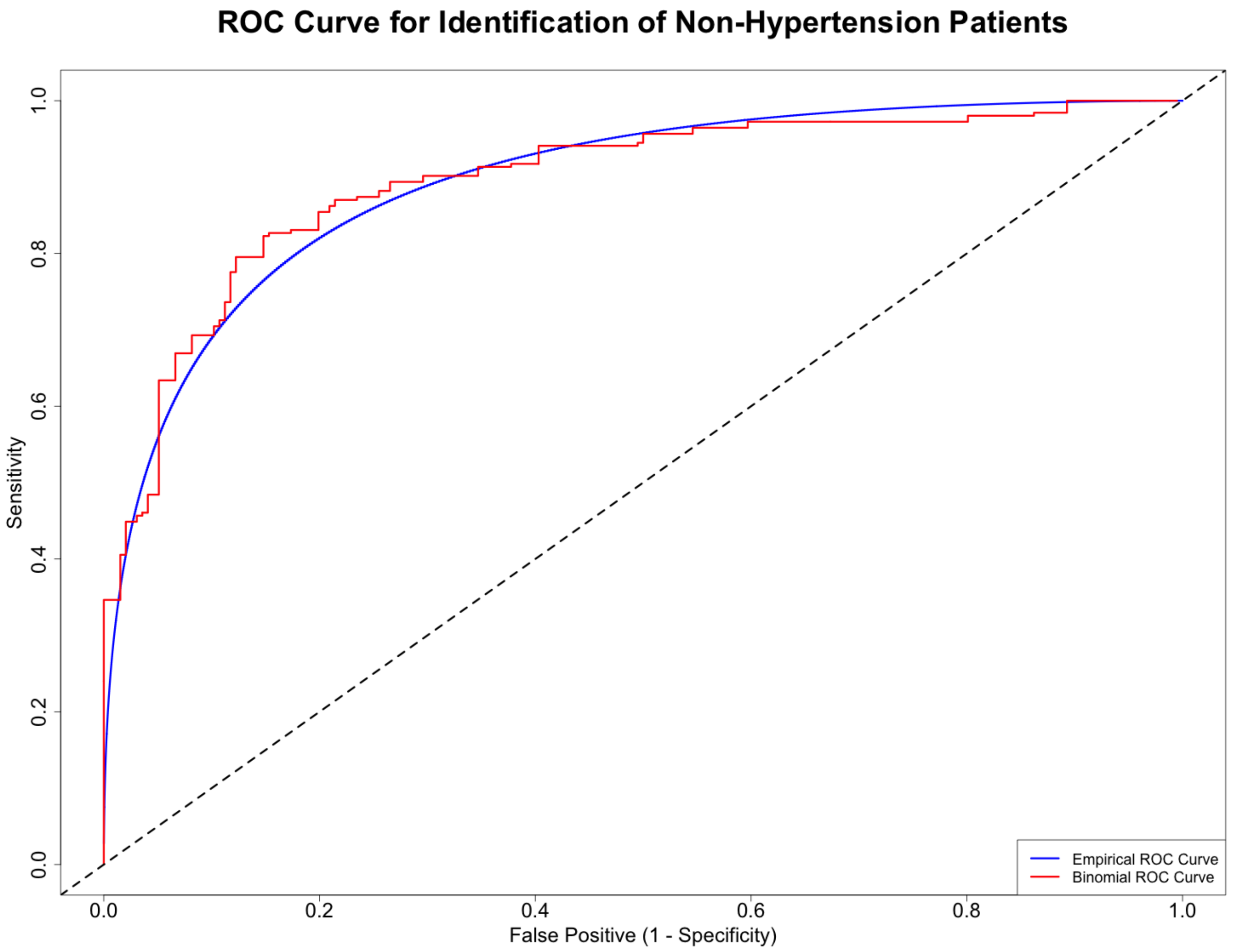

Figure 2: (b) ROC Curve

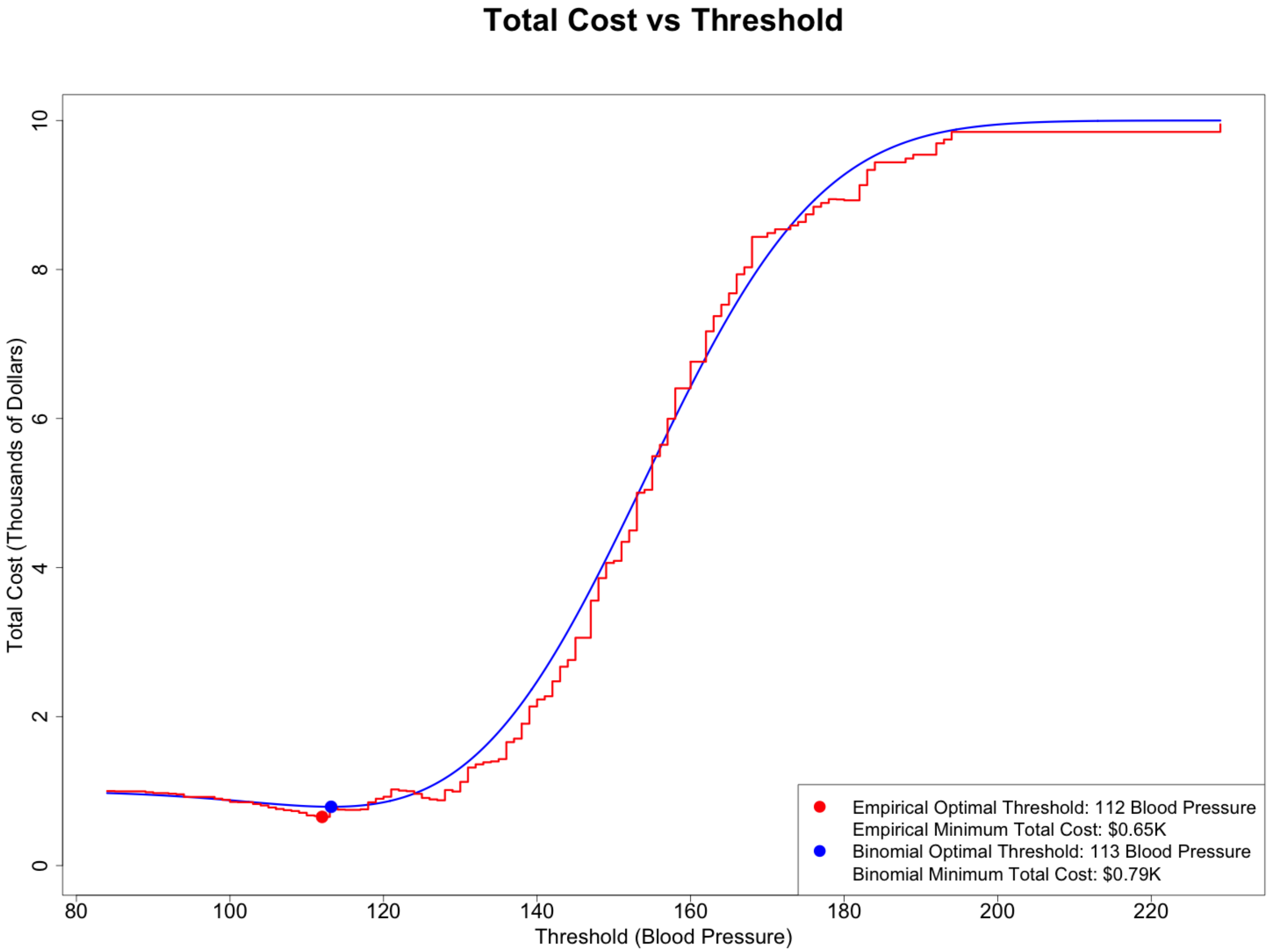

Figure 3: (d) Total Cost

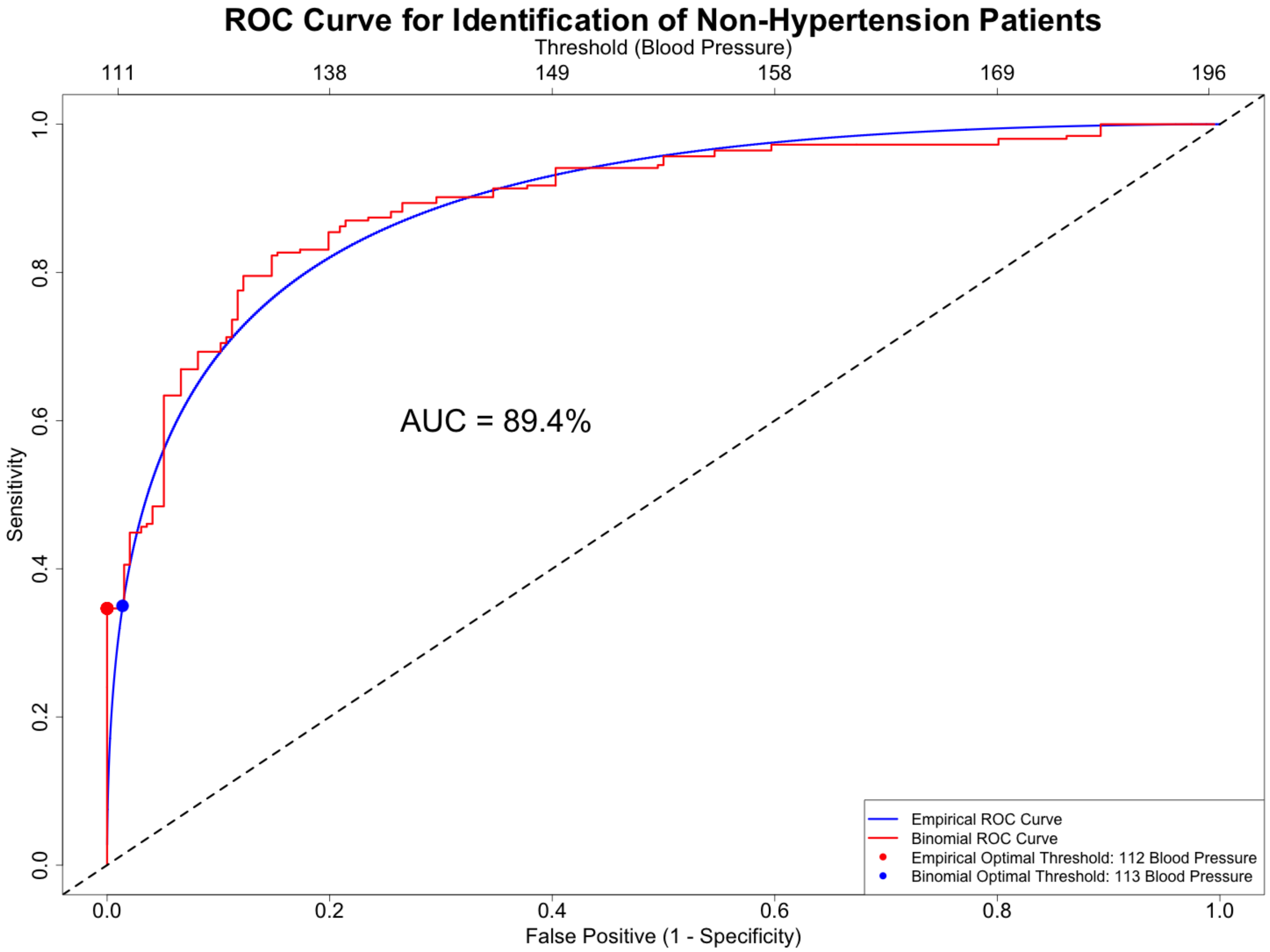

Figure 4: (e) Optimal Threshold & ROC Curve