HW 5: Pledged Individual Midterm

Due: Tuesday March 6, 2pm

Instructions

- You must do this assignment by yourself. Do not discuss it with your classmates.
- Open notes. You may use the Internet to search, but you may not post questions.
- State the honor code.
- Produce your assignment as a RMarkdown (or knitr) document rendered to pdf (knit to pdf).
- Also submit your Rmd or Rnw file (it will not be graded but we want it for reference purposes).
- Show all the code (use `echo=TRUE` as option in R chunks) as well as the results.
- 100 total points. Easier problems are worth more points than difficult ones.
- See Syllabus for other policies.
- Remember that on Thursday March 1 there is no class.

Problem 1 [30 points]

Let:

```
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

Define the matrix A in R and perform the following operations:

- 1. Compute the row mean of A using the `apply` function. [5]
- 2. Raise each elememt of the matrix A to the power of 2. Hint: use the `^` operator. [3]
- 3. Take the square of A. This is matrix exponentiation defined as $A^2 = A \times A$. Use matrix multiplication operator for this problem. [3]
- 4. Take the square of A using eigen-decomposition (`eigen()` function in R). Remember that if a square matrix A has an eigen-decomposition, $A = V \times D \times V^{-1}$, then $A^n = V \times D^n \times V^{-1}$, where columns of V contains the eigenvectors of A and D is a diagonal matrix with eigenvalues of A on the diagonal. You should get the same results as part 3. [4]
- 5. Compute the inner product of the third row of A and the third column of A. [3]
- 6. Compute the outer product of the column vector `c` with the fourth row of A where

$$c = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

  . There resulting matrix should be 4×4. [4]

- 7. Extract the first and fourth columns of A. [3]
- 8. Discrete difference operator is an useful tool to efficiently compute row-wise/column-wise differences of a matrix and it is widely used in speeding up certain machine learning algorithms. In this problem, we want to take row-wise difference of matrix A. This means we want to take elementwise

difference between the first and second rows of A, elementwise difference between the second and third rows of A, etc. Since $A \in \mathbb{R}^{4\times4}$, the resulting matrix containing row-wise differences will have dimension of 3×4. Instead of using a for-loop to iterate over the rows of A and take differences between consecutive rows, we use the discrete difference operator to speed up the operation. For this problem, use `sapply()` to construct this matrix difference operator B. Then use $B \times A$ to compute the row-wise difference. Be sure to print out the results. Define first difference operator $B \in \mathbb{R}^{3\times4}$ to be [5]

$$B = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Problem 2 [25 points]

The MNIST database is a large database of handwritten digits which has been widely used to train and compare performances of different machine learning systems. For this problem, we want to visualize and see what an average handwritten digits (1-9) would look like. Make sure your resulting plot looks similar to the example plot provided. Some instructions are as follows:

- The processed dataset `zip.rds` can be found on Canvas under Files/data. Each row of this dataset corresponds to one example of a handwritten digits. The first column of the dataset is an integer from 0-9, indicating the digit for each row. The `2:257` elements of each row is a flattened 16×16 image (i.e. a 16×16 matrix) of a handwritten digit, where each element indicates the pixel intensity at each location in the image.

- Use base R graphics, specifically the `image()` function.

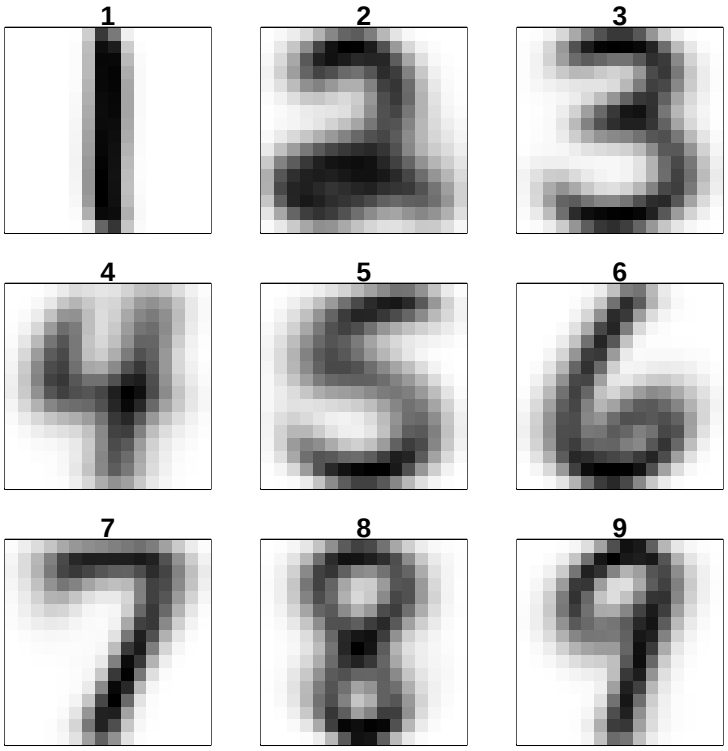
- The resulting plot should be a 3×3 grid, as shown in the example plot.

- To set the customized color palette to match the example plot, the following code can be useful:

```
colors<-c('white','black')
cus_col<-colorRampPalette(colors=colors)
```

- For plotting the average handwritten digit `1`, you are supposed to extract all rows corresponding to `1` and exclude the first column because only `2:257` columns are actual pixel values. Then you compute the average values for each pixel over all examples of digit `1`. Repeat this for all digits 1-9. Data structure such as `array()` can be helpful.

- The image from the previous step can be upside down. Reorder the columns of the resulting array accordingly.



Problem 3 [25 points]

Classification is an important task in many machine learning applications. Before actually building any classifiers, it is always a good idea to perform exploratory data analysis and visualize the data in low-dimensional space. Principal component analysis (PCA) is a widely used tool for visualizing the most important features of data in 1,2,3-dimensional space. For this problem, no prior knowledge about PCA is needed. Follow the instructions below and replicate the example plot:

- Use the same `zip.rds` dataset for this problem.
- For this problem, we are only interested in differentiating digit `3` from digit `8`. Therefore, subset the dataset and keep only rows corresponding to digit `3` and digit `8`. Your new data frame should have 1200 rows.
- Recall from the previous problem, each row has 257 elements where the first element denotes the digit labels and the `2:257` elements are the actual pixel values for that particular image. The `2:257` elements are called features in the pattern recognition literature. We use $X \in \mathbb{R}^{1200\times 256}$ to denote this data matrix.

- Use the `princomp()` function from the `stats` package to perform PCA on data matrix X and save the results as `pca`. Please read the documentation of `princomp()`.

- To project the data onto the first two principal component directions, we need to extract the first two columns from the 256×256 `princomp$loadings` matrix. Projecting data matrix X onto the first principal component direction just means multiplying (i.e. `%*%`) the data matrix X with the first column of `princomp$loadings` and projecting data matrix X onto the second principal component direction just means multiplying (i.e. `%*%`) the data matrix X with the second column of `princomp$loadings`. Each of the two projections will produce a 1200×1 vector, which shows how each of the digits will "look like" in this particular subspace defined by that particular principal component direction.

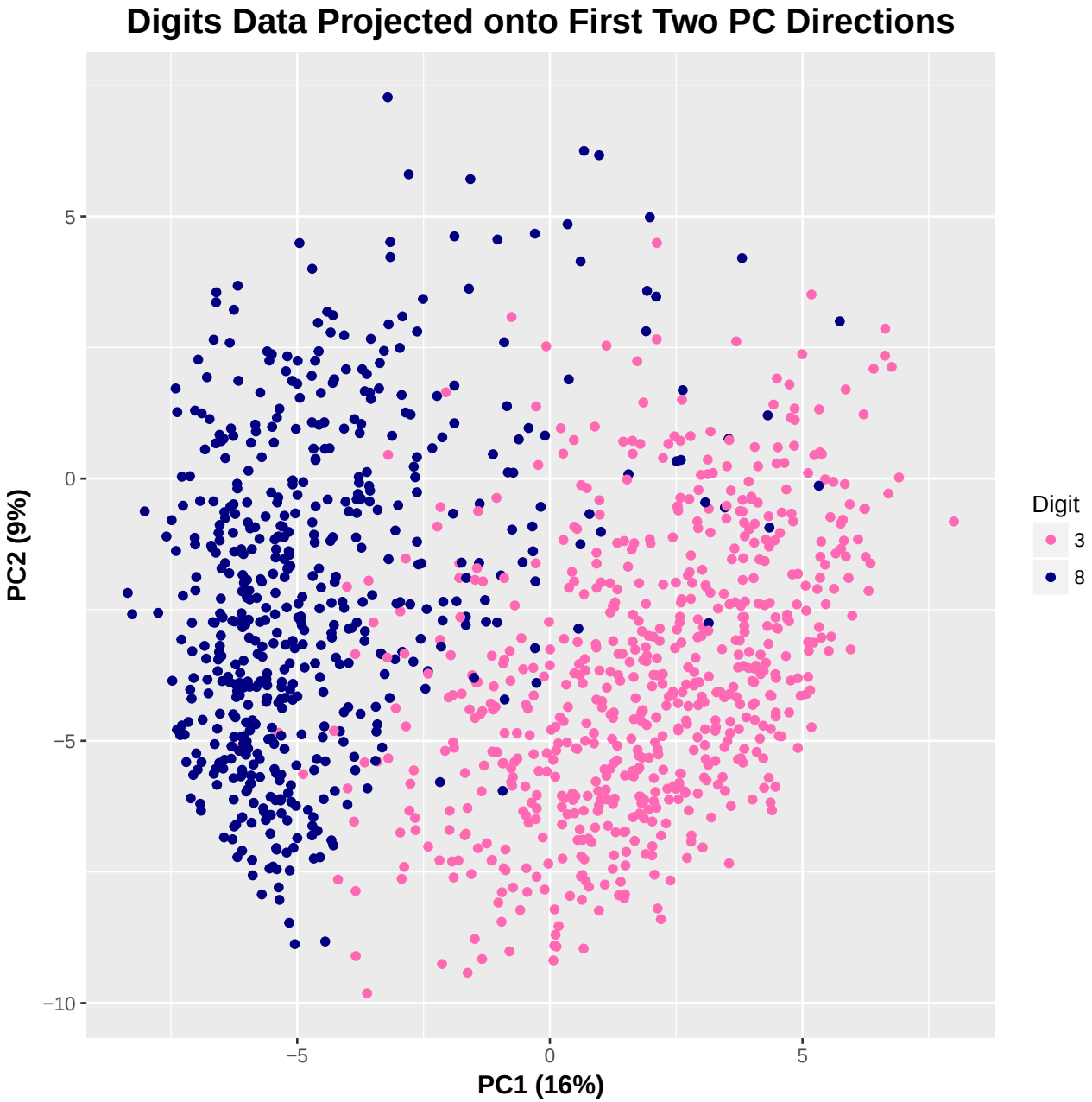
- Use `ggplot2` to plot data projected onto first principal component direction against data projected onto second principal component direction, with colors indicating the digit label for each point.

- Make sure you have a title and labels for both axes. Make sure the font size is readable. [5]
- For the x-axis and y-axis, be sure to include proportion of variance explained by that direction in the axis label (as shown in the example plot). Proportion of variance explained by each principal component direction is a widely used way to assess the quality of each principal component direction. You should compute the proportion of variance explained by the first two directions that we extracted from the previous step using the following code. [5]

```
prop.of.variance <- round(pca$sdev^2/sum(pca$sdev^2),2)
```

- Remeber to label your legend title. [5]

- You can choose any two colors you like.
- Overall plot quality. [10]



Problem 4 [20 points. Note: each item is worth 5 points for STAT 405 and 4 points for STAT 605. Last item is for STAT 605 only.]

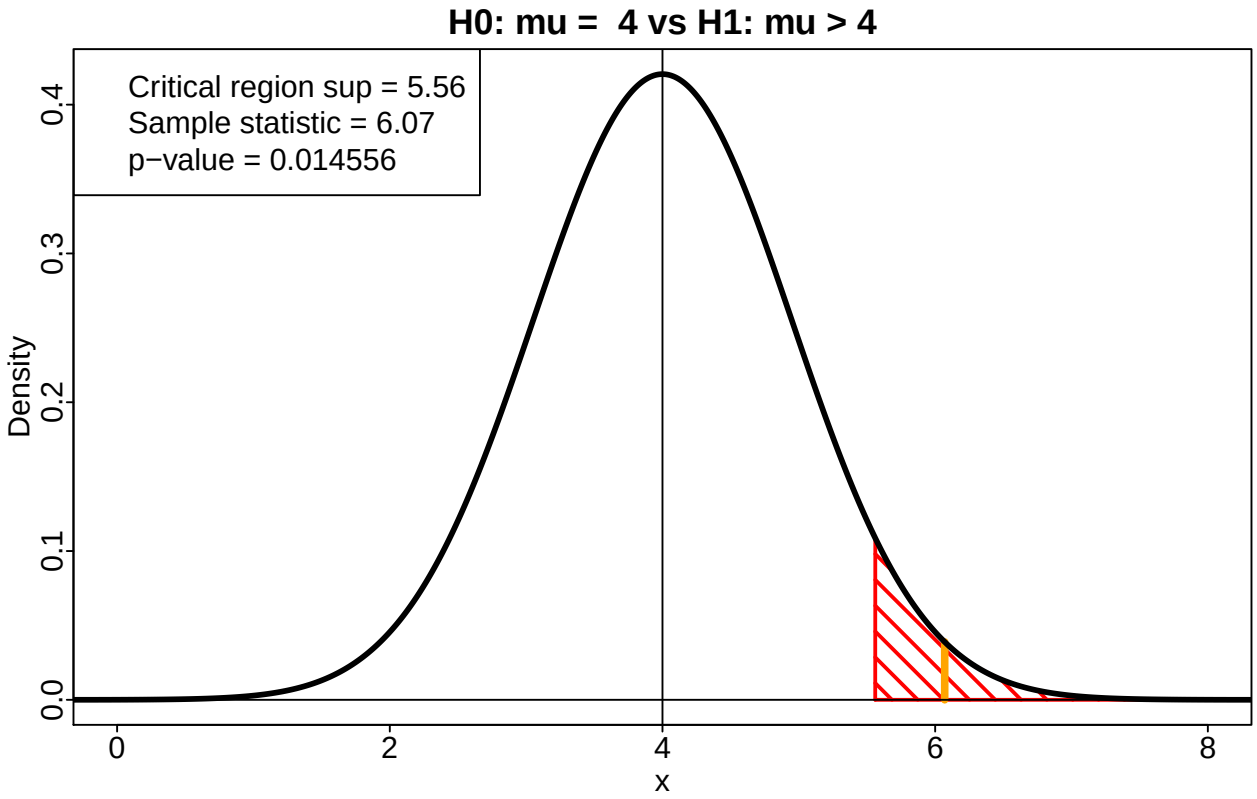
- To be a successful statistician, a very important concept you need to master is hypothesis testing. There are numerous resources on the Internet to consult if you need to refresh concepts.
- You will assume that you are sampling from a Normal distribution with known variance, so you will employ a z-test (not t-test).
- The parameter and result values to use are:

```
mu0 <- 4              ## Null hypothesis mean value
stdev <- 3            ## Known population standard deviation
signif.level <- 0.05  ## Test significance level
sample.mean <- 6.07   ## Mean of the random sample
n <- 10               ## Sample size
mu1 <- 6.2            ## Alternative hypotesis mean value to use
                      ## for error type 2 and power
```

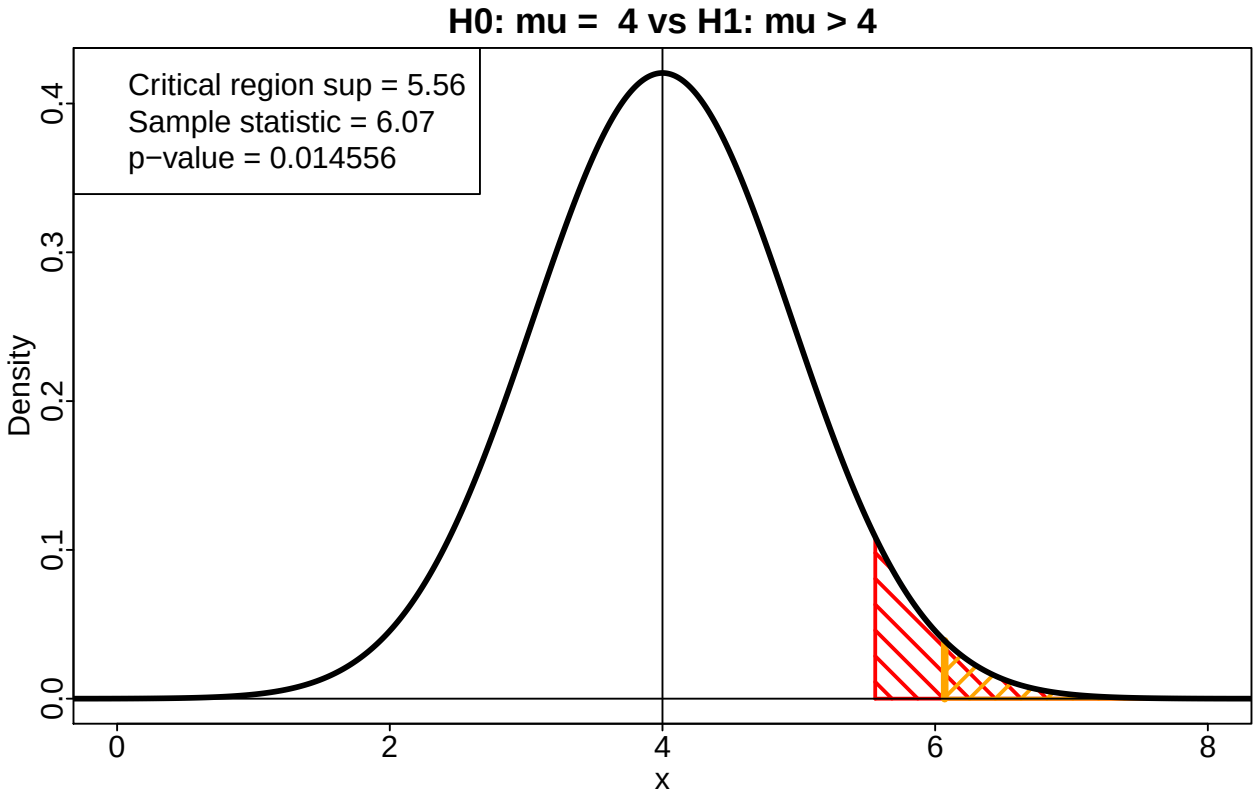
- What you will have to do is to replicate, using base R graphics, the following plots, that illustrate the key components of a test of hypothesis.
- **STAT 405 students will have to produce only one-sided test plots.**
- **STAT 605 students will produce both one-sided and two-sided test plots.**
- It is highly recommended (but not required) that you produce a function with parameters that can be tweaked, such as:

```
hyp.testing <- function(mu0, stdev, signif.level,
                        sample.mean, n,
                        show_crit, show_pvalue,
                        show_alt, mu1, show_beta, show_power,
                        two_sided) {
  ## Your code
}
```

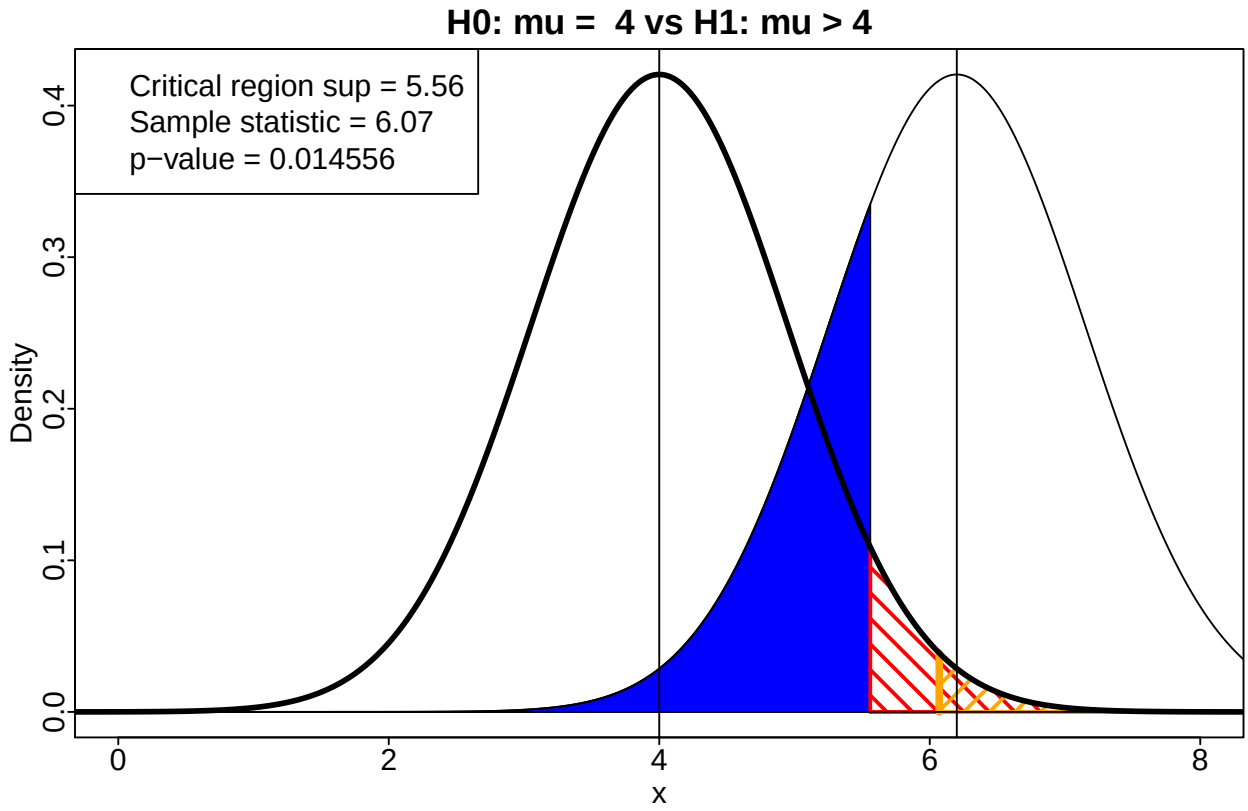
- 1. Density plot showing the critical region (red stripes. Hint: use `polygon`. It is also known as error of type 1 or α), and where the `sample.mean` (orange vertical line) is located:

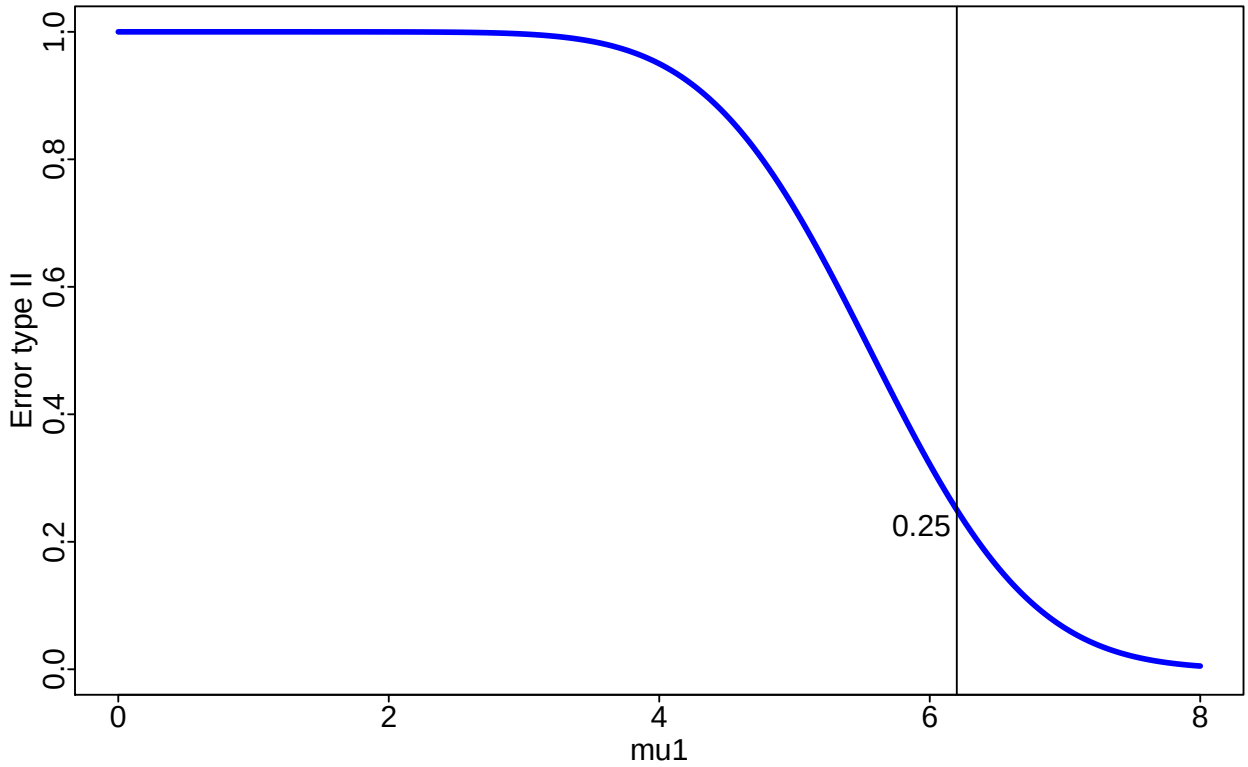


- 2. Superimpose the p-value probability region (orange stripes):

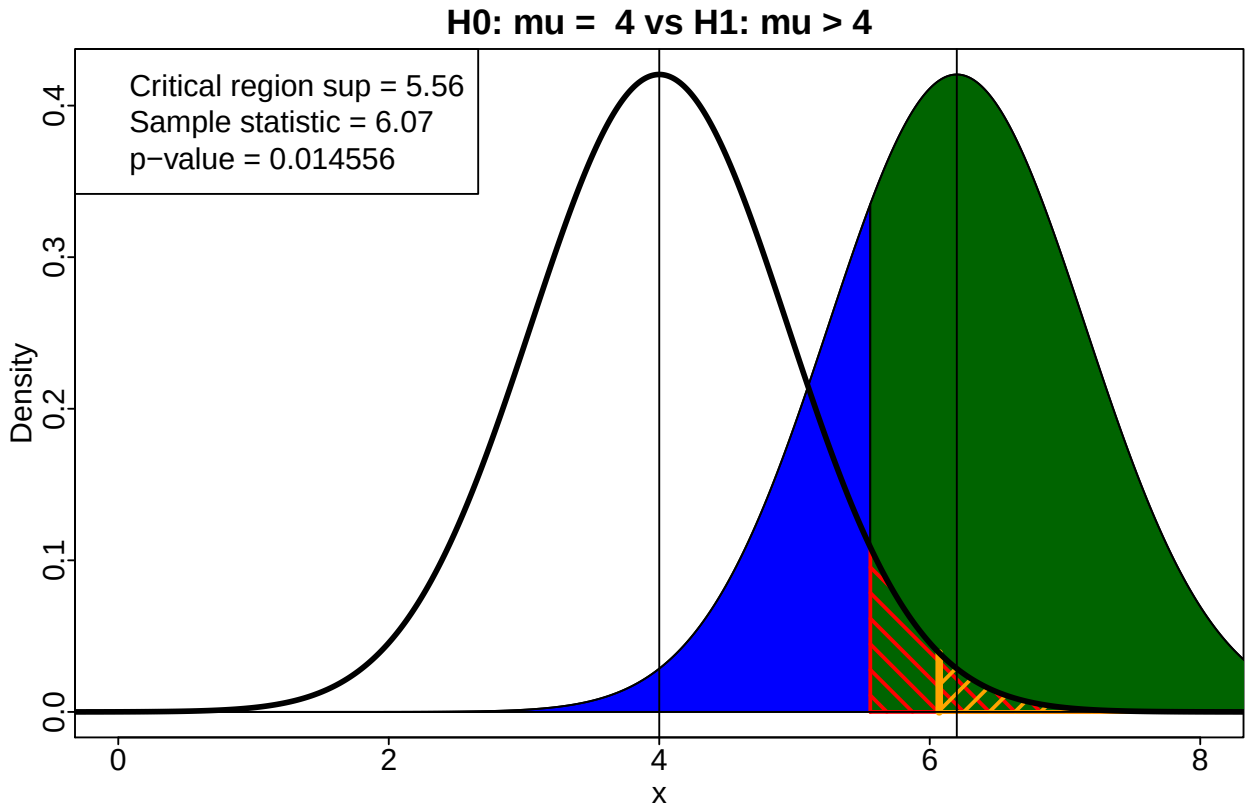


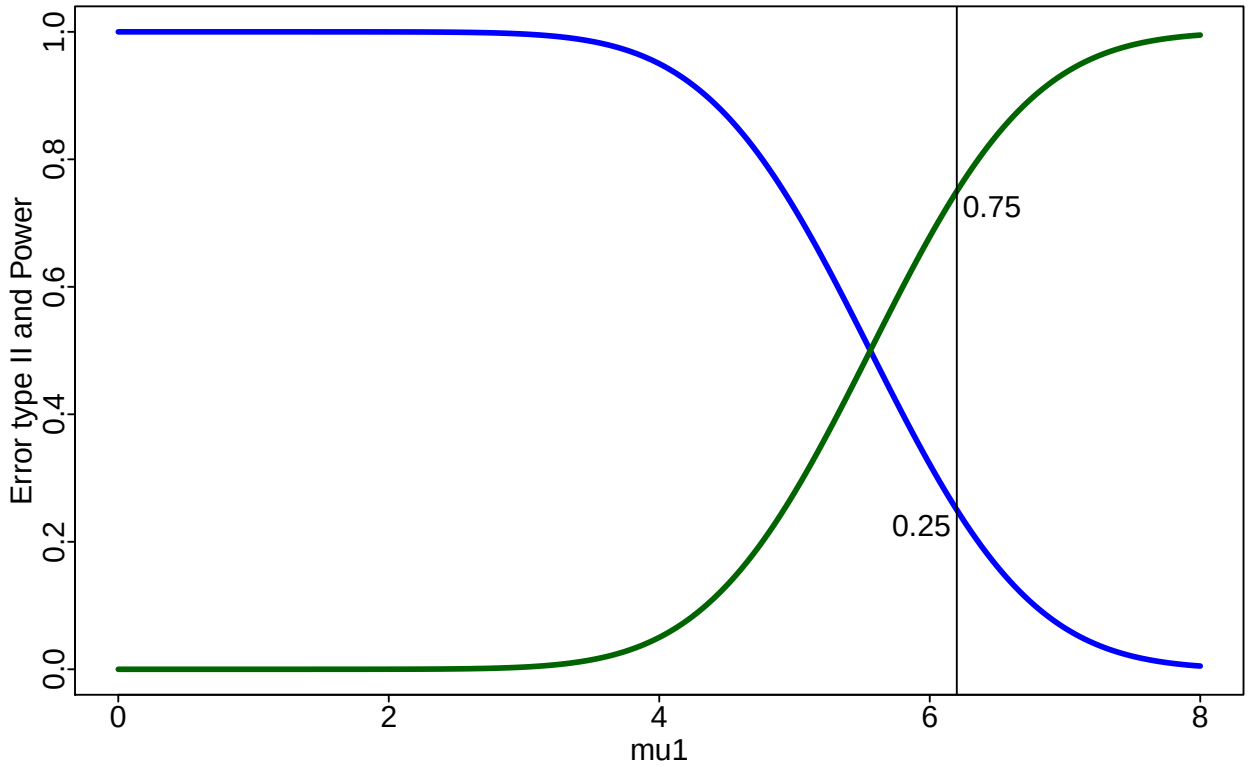
- 3. Add the error of type II (or β. Blue region and blue line):



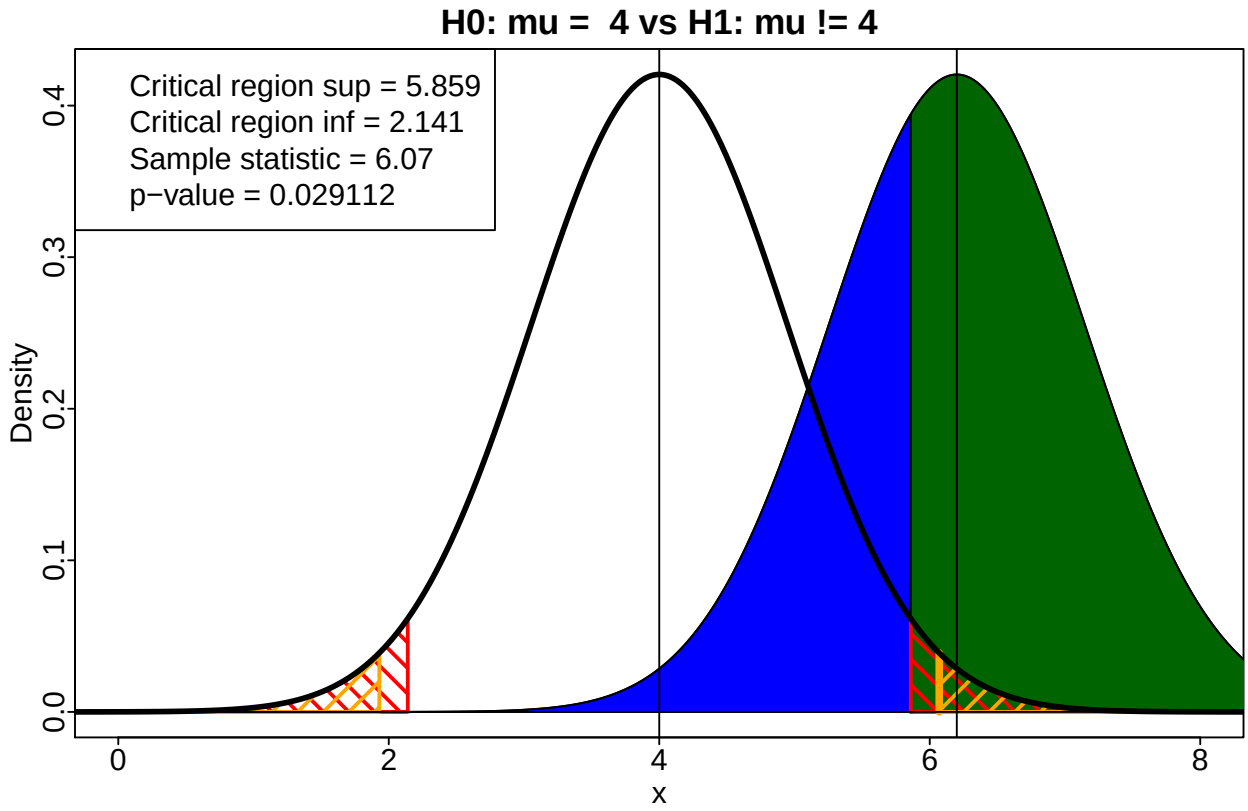


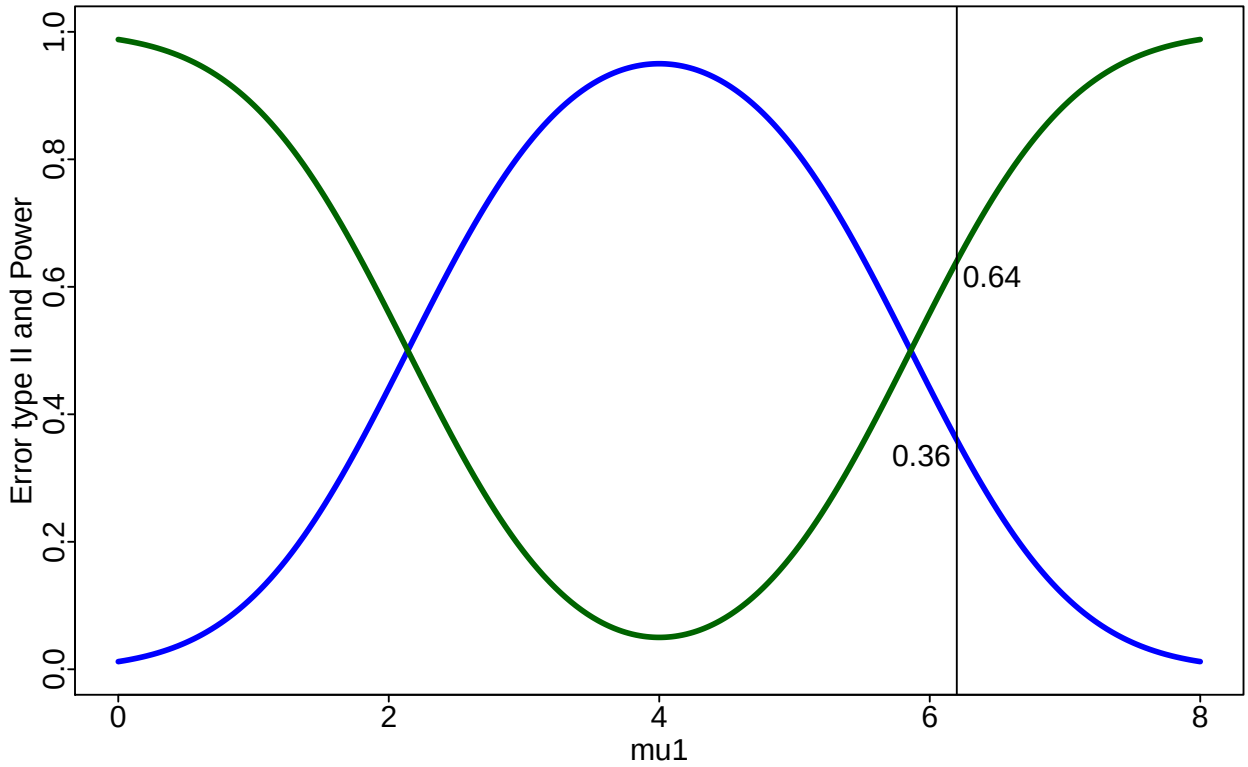
- 4. Add the power ($1 - \beta$. Dark green region and dark green line):





- 5. (STAT 605 only) Create the two-sided version:


H0: mu = 4 vs H1: mu != 4
Critical region sup = 5.859
Critical region inf = 2.141
Sample statistic = 6.07
p−value = 0.029112
Density
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
x


Error type II and Power
0.0
0.2
0.4
0.6
0.8
1.0
0.64
0.36
0
2
4
6
8
mu1