Fashion Image Analysis

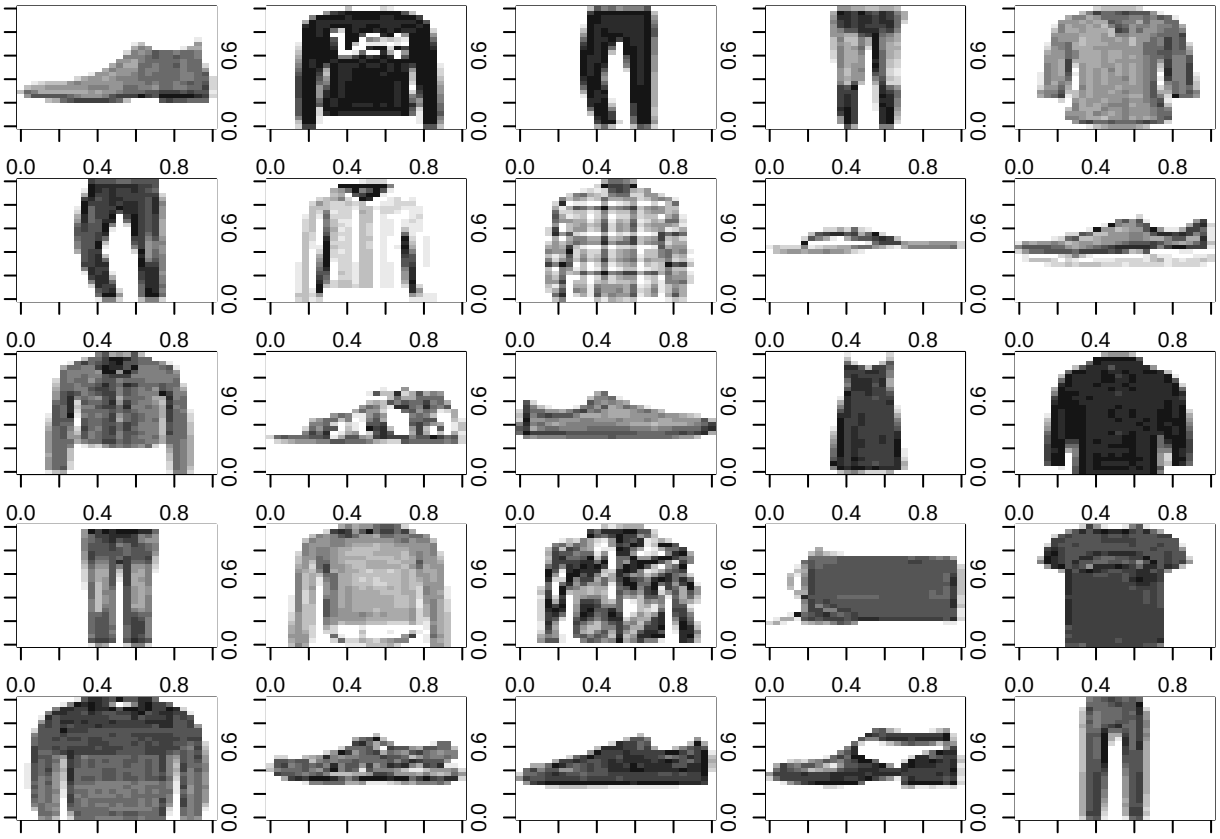
2024-03-28

The MNIST database is a famous benchmarking data set of handwritten digits. However, let's play with a similarly structured set of data on clothing items!

A plot of the first 25 images in the data oriented properly!

```
# Loading the data
load_mnist <- function() {
  load_image_file <- function(filename) {
    ret = list()
    f = file(filename,'rb')
    readBin(f,'integer',n=1,size=4,endian='big')
    ret$n = readBin(f,'integer',n=1,size=4,endian='big')
    nrow = readBin(f,'integer',n=1,size=4,endian='big')
    ncol = readBin(f,'integer',n=1,size=4,endian='big')
    x = readBin(f,'integer',n=ret$n*nrow*ncol,size=1,signed=F)
    ret$x = matrix(x, ncol=nrow*ncol, byrow=T)
    close(f)
    ret
  }
  load_label_file <- function(filename) {
    f = file(filename,'rb')
    readBin(f,'integer',n=1,size=4,endian='big')
    n = readBin(f,'integer',n=1,size=4,endian='big')
    y = readBin(f,'integer',n=n,size=1,signed=F)
    close(f)
    y
  }
  test <<- load_image_file('t10k-images-idx3-ubyte')

  test$y <<- load_label_file('t10k-labels-idx1-ubyte')
}


show_digit <- function(arr784, col=gray(12:1/12), ...) {
  image(matrix(arr784, nrow=28)[,28:1], col=col, ...)
}
```

```
load_mnist()

# Plot the first 25 images
par(mfrow = c(5, 5), mar = c(1, 1, 1, 1))  # Adjust margins to 1 on all sides

for (i in 1:25) {
  show_digit(test$x[i, ])
}
```



Lets run principal components (without scaling) on the images. I'm saving this object in an Rdata file so that it doesn't run every time.

```
pca_result <- prcomp(test$x, scale. = FALSE)

# save(pca_result, file = "pca_results.RData")
```

What is the maximum number of components that are permittable?

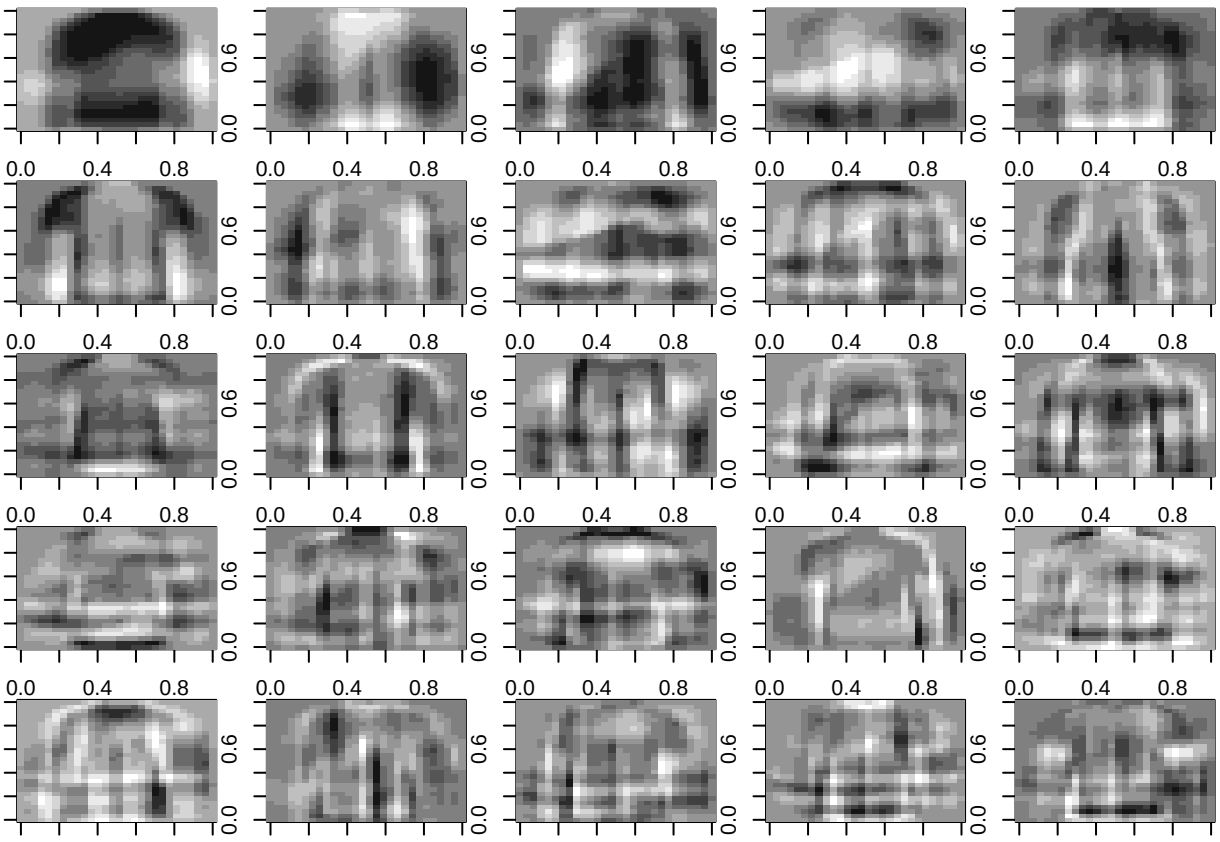
```
# Calculate the maximum number of components permissible
max_components <- min(nrow(test$x), ncol(test$x))
max_components
```

```
## [1] 784
```

Plotting the first 25 resulting eigenvectors as images

```
# Load the PCA results from the RData file
load("pca_results.RData")

# Extract the eigenvectors corresponding to the first 25 principal components
eigenvectors <- pca_result$rotation[, 1:25]

# Plot the first 25 eigenvectors as images
par(mfrow = c(5, 5),  mar = c(1, 1, 1, 1))  # Arrange plots in a 5x5 grid
```

```
for (i in 1:25) {
  show_digit(eigenvectors[, i])
}
```



What percentage of the original variation in the pixels is explained by the first 25 PCs?

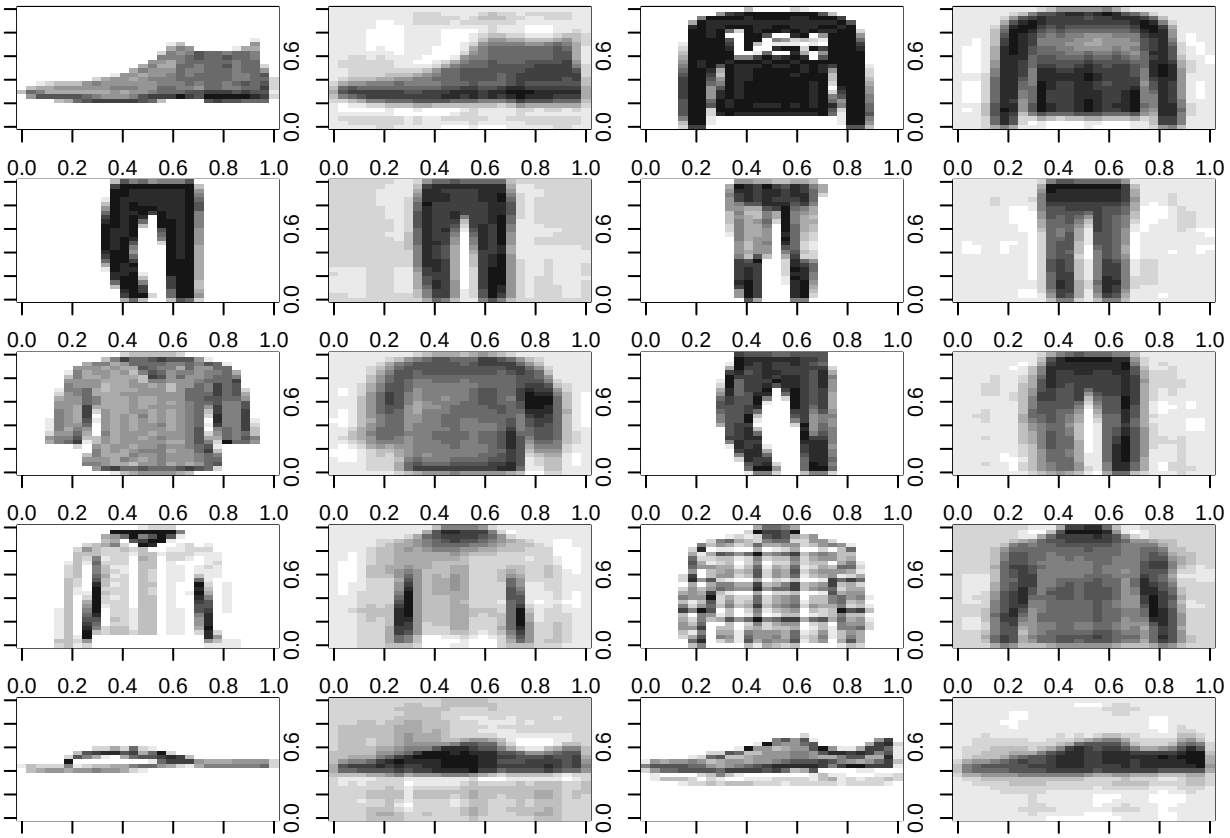
```
variance_explained <- pca_result$sdev^2
total_variance <- sum(variance_explained)
variance_explained_first_25 <- sum(variance_explained[1:25])
percentage_explained <- (variance_explained_first_25 / total_variance) * 100
cat('The percentage of the original variation in the pixels is explained by the first 25 PCs:', percenta
```

```
## The percentage of the original variation in the pixels is explained by the first 25 PCs: 80.45961
```

Lets reconstruct approximations of the original observations using 25 PCs and plot side-by-sides for the first 10 digits of the reconstructions and originals in a 5x4 matrix of images.

```
# Load the MNIST data
load_mnist()

# Project the original data onto the first 25 principal components
projected_data <- test$x %*% pca_result$rotation[, 1:25]

# Reconstruct the original observations using the projected data and eigenvectors
reconstructed_data <- projected_data %*% t(pca_result$rotation[, 1:25])
```

```r
# Plot side-by-side comparisons for the first 10 digits of the reconstructions and originals
par(mfrow = c(5, 4), mar = c(1, 1, 1, 1))  # Arrange plots in a 5x4 grid with reduced margins
for (i in 1:10) {
  # Original image
  show_digit(test$x[i, ])

  # Reconstructed image
  show_digit(reconstructed_data[i,])
}
```



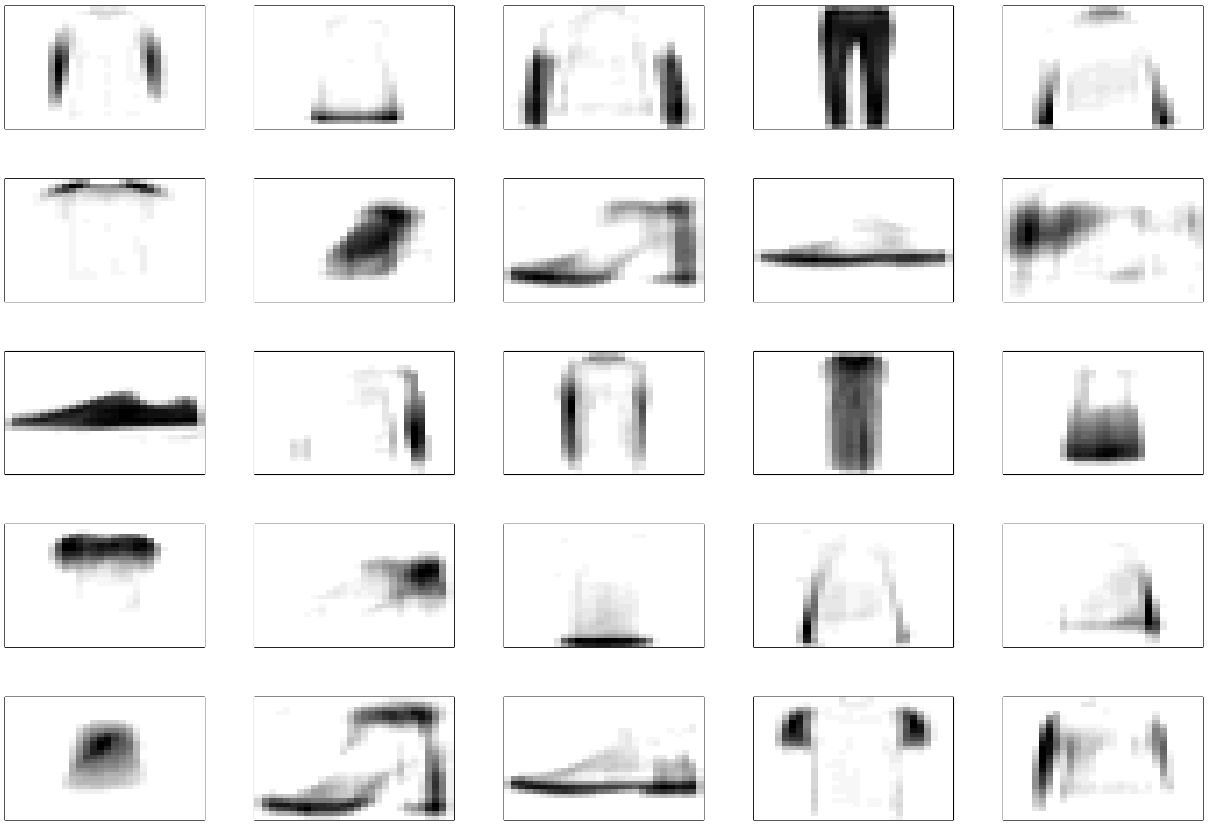
Lets run NMF and plot the 25 basis vectors as images

```r
nmf <- function(x, q, eps=0.001, maxit=2000, w=NULL, h=NULL){
  n <- nrow(x)
  p <- ncol(x)
  if(any(x<0)){x <- as.matrix(x)+abs(min(x))}
  else{x <- as.matrix(x)}
  if(is.null(w)){
    w <- matrix(runif(n*q, min(x), max(x)), n, q)
  }
  if(is.null(h)){
    h <- matrix(runif(p*q, min(x), max(x)), q, p)
  }
  ed <- sum((x-w%*%h)^2)
  conv <- FALSE
```

```
  ctr <- 1
  while(!conv){
    ctr <- ctr+1
    h <- h * (t(w) %*% x) / (t(w) %*% w %*% h)
    w <- w * (x %*% t(h)) / (w %*% h %*% t(h))
    wh <- w%*%h
    ed[ctr] <- sum((x-wh)^2)
    if((ed[ctr-1]-ed[ctr] < eps)|(ctr==maxit)){
      conv <- TRUE
    }
  }
  list(ed=ed, w=w, h=h, x=x)
}
```

```
nmf_result <- nmf(fnmfres$x, q=25, eps=0.001, maxit=600, w=fnmfres$w, h=fnmfres$h)
# save(nmf_result, file = "nmf_result.RData")
```

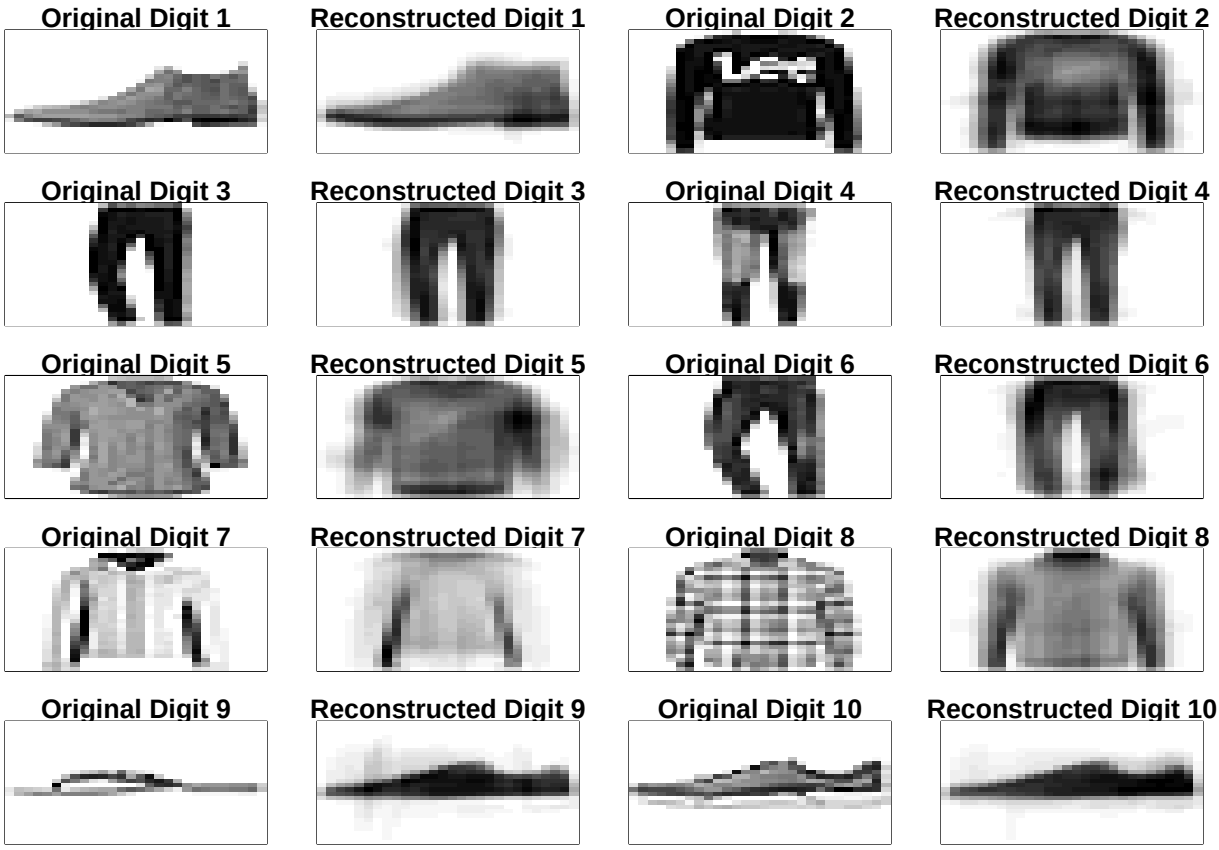
```
load('fnmfres.RData')
load('nmf_result.RData')

par(mar=c(1, 1, 1, 1))

# Plot the basis vectors as images
par(mfrow=c(5, 5)) # Set up the plotting layout
for (i in 1:25) {
  image(matrix(nmf_result$h[i, ], nrow = 28), col = gray((32:0)/32), xaxt = "n", yaxt = "n")
}
```



The brighter images obtained from NMF is attributed to its ability to capture localized patterns or features, as it explicitly enforces non-negativity constraints on both the data and the components. The absence of a grey background in NMF images could be attributed to its ability to focus on capturing meaningful patterns without incorporating irrelevant or low-variance components. PCA tends to capture global patterns in the data, which may result in blurred representations.

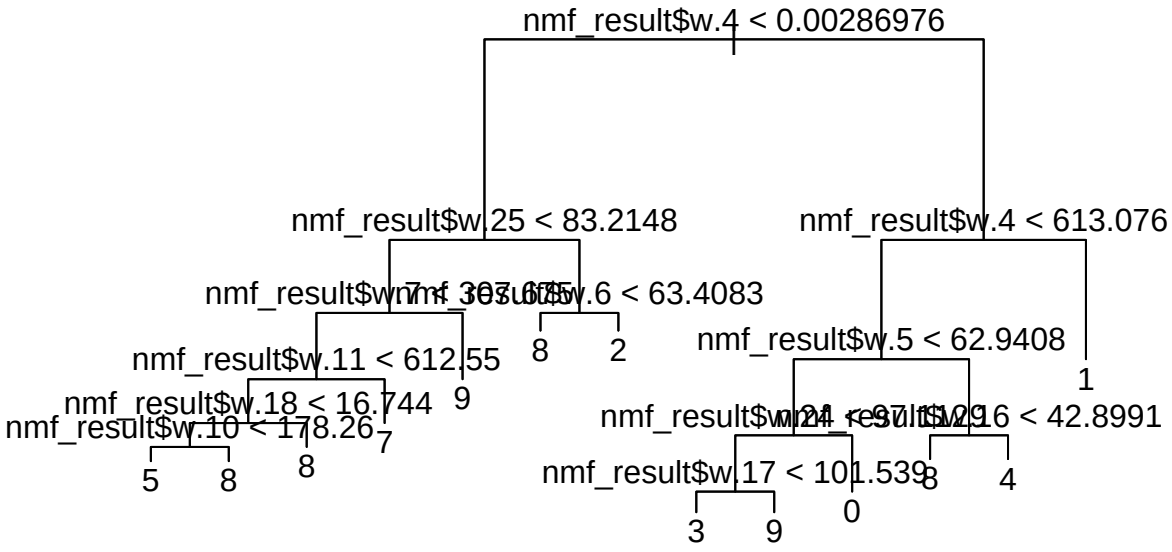
Now I'm reconstructing approximations of the original observations using 25 NMF bases and plotting side-by-sides for the first 10 digits of the reconstructions and originals in a 5x4 matrix of images

```
reconstructed <- nmf_result$w %*% nmf_result$h

# Plot side-by-sides for the first 10 digits of the reconstructions and originals in a 5x4 matrix of im
par(mfrow=c(5, 4), mar=c(1, 1, 1, 1))

# Iterate over the first 10 digits
for (i in 1:10) {
  # Original digit
  original_digit <- matrix(fnmfres$x[i, ], nrow = 28)
  # Reconstructed digit
  reconstructed_digit <- matrix(reconstructed[i, ], nrow = 28)

  # Plot original digit
  image(original_digit, col = gray((32:0)/32), xaxt = "n", yaxt = "n", main = paste("Original Digit", i
  # Plot reconstructed digit
  image(reconstructed_digit, col = gray((32:0)/32), xaxt = "n", yaxt = "n", main = paste("Reconstructed
}
```



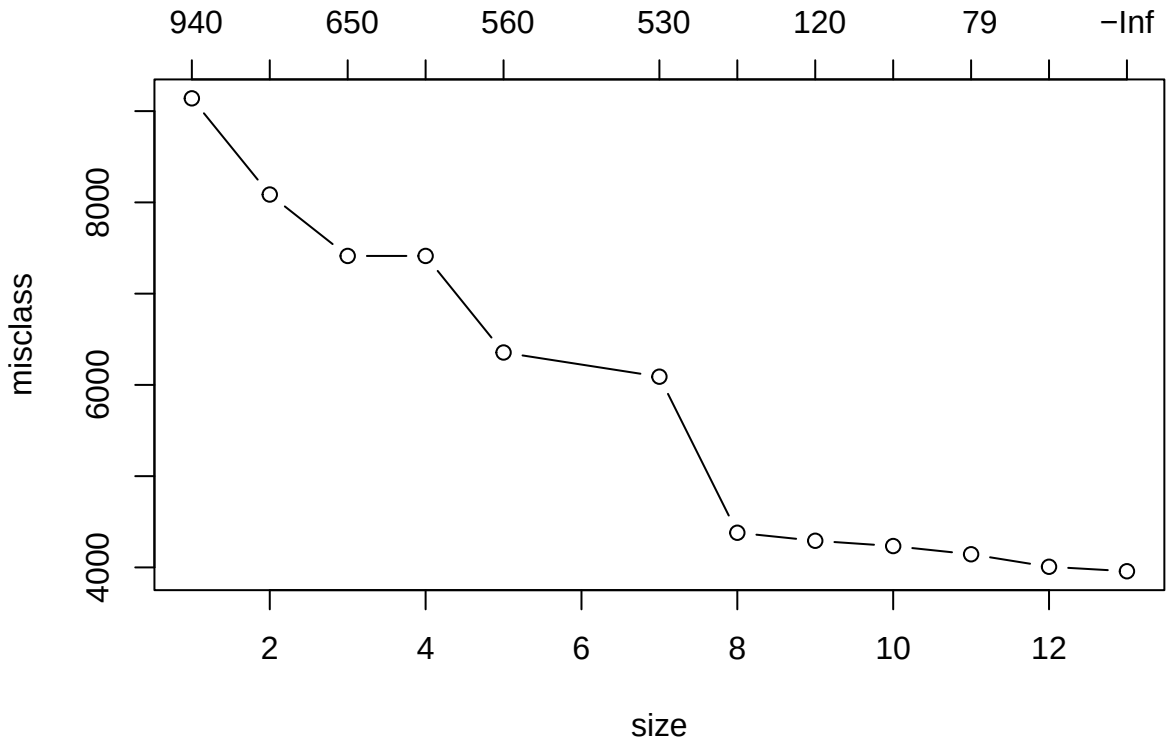
Fitting a classification tree with labels as the response variable and the NMF 'scores' as the predictors

```
library(tree)

tree_model <- tree(factor(test$y)~nmf_result$w)
plot(tree_model)
text(tree_model)
```




```
# Cross-validation using prune.misclass
cv_tree <- cv.tree(tree_model, FUN = prune.misclass)
```

```
plot(cv_tree, type='b')
```



Number of terminal nodes suggested to be removed: None

```
# Cross-validated misclassification rate of the best tree
misclassifications <- cv_tree$dev[which.min(cv_tree$dev)]

# Print the results
print(paste("Cross-validated misclassification rate:", misclassifications/length(test$y)))
```

```
## [1] "Cross-validated misclassification rate: 0.3958"
```