Practical 2.9 & 2.10 Solution: Supervised learning: MNIST digit classification

CHEN

2024-04-23

Dimensionality reduction & Feature selection

```
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.2
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
mnist_raw <- read.csv("mnist_train.csv", header = FALSE)
```

```
pixels_gathered <- mnist_raw %>%
  head(1000) %>%
  rename(label = V1) %>%
  mutate(instance = row_number()) %>%
  gather(pixel, value, -label, -instance) %>%
  tidyr::extract(pixel, "pixel", "(\\d+)", convert = TRUE) %>%
  mutate(pixel = pixel - 2,
         x = pixel %% 28,
         y = 28 - pixel %/% 28)
```

```
features=data.frame(label=mnist_raw$V1[1:1000])
# set labels (of 1000 examples) as the first column in features dataframe
for (i in 1:56)
  features=cbind(features,fi=c(1:1000)*0)
  # create 56 features (28 for rows and 28 for column) for the 1000 examples
for (i in 1:28) {
  # loop over 28 rows and 28 columns
  for (j in 1:1000) {
    # compute row & column means for each digit example using pixels gathered
    features[j,i+1]= mean(pixels_gathered$value[pixels_gathered$instance==j&pixels_gathered$y==i]);
    # first 28 features: row means (each row has fixed y)
    features[j,i+29] = mean(pixels_gathered$value[pixels_gathered$instance==j&pixels_gathered$x==i-1]);
    # next 28 features: column means (each row has fixed x)
  }
}
```

Compute means for each label and feature.

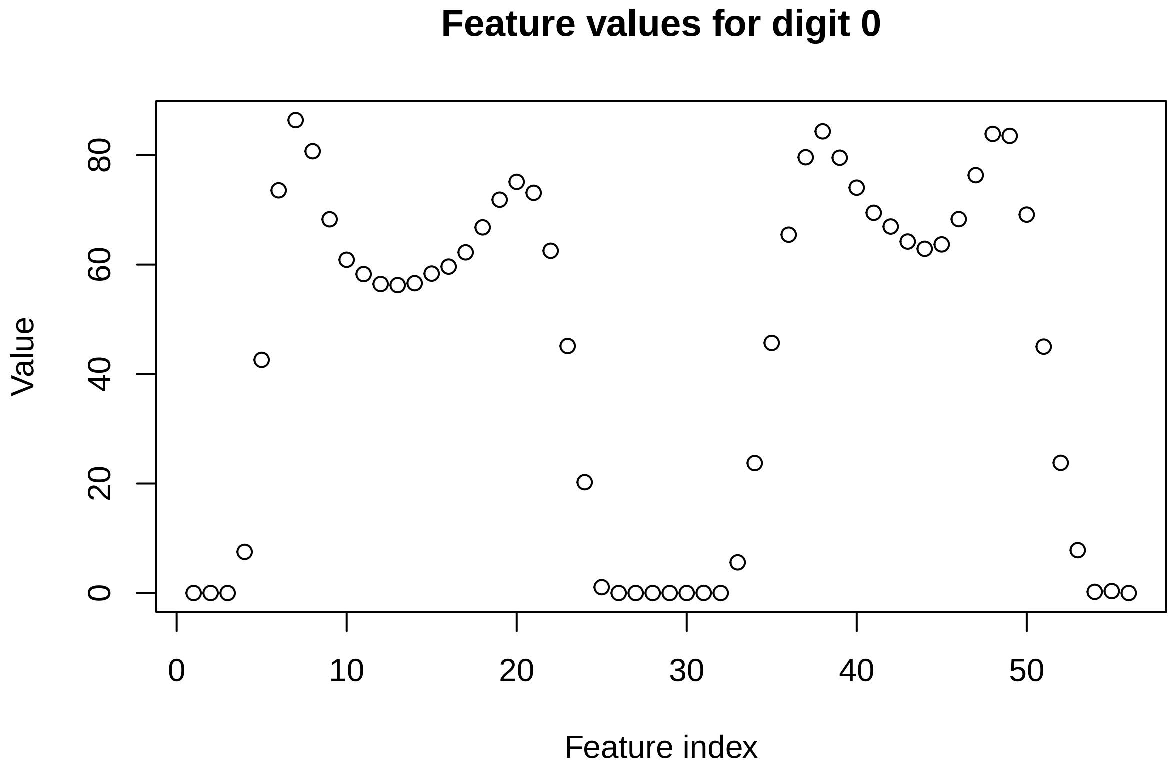
```
fstats <- matrix(1:560, nrow=10, ncol=56)
for (i in 1:10)
  for ( j in 1:56) {
  fstats[i,j] <- mean(features[features$label==i-1, j+1])
  }
```

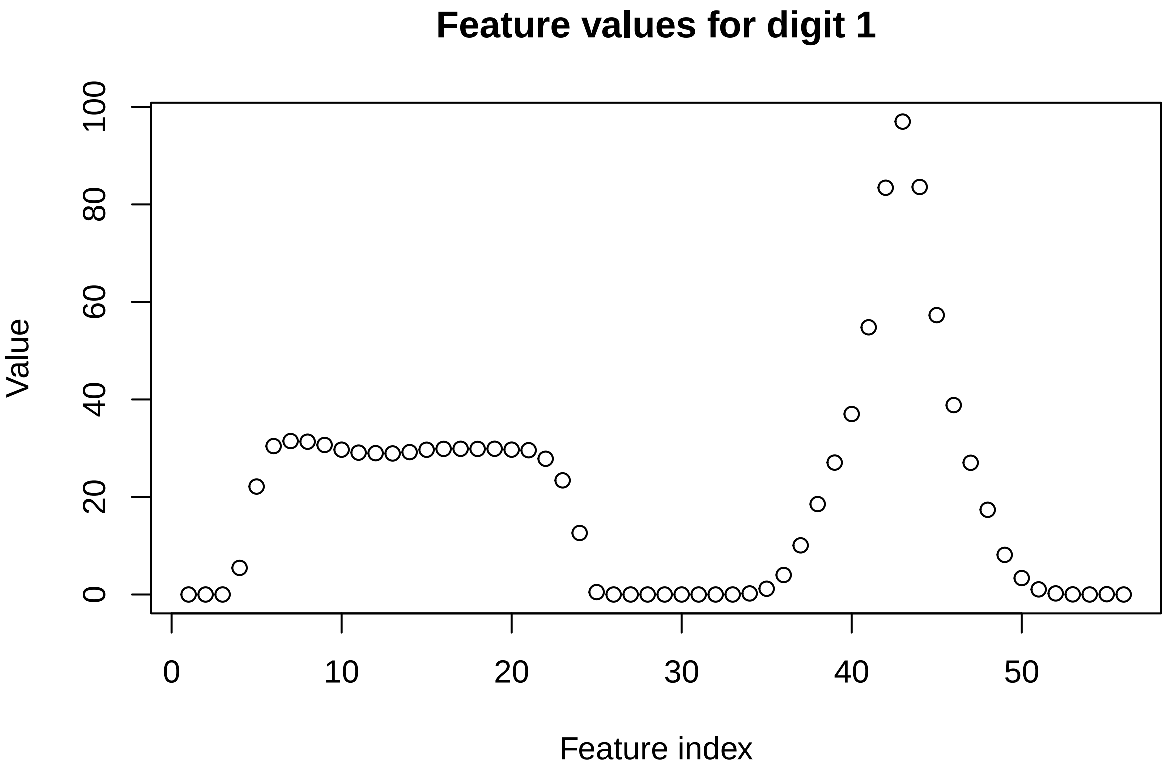
Plot features of each label.

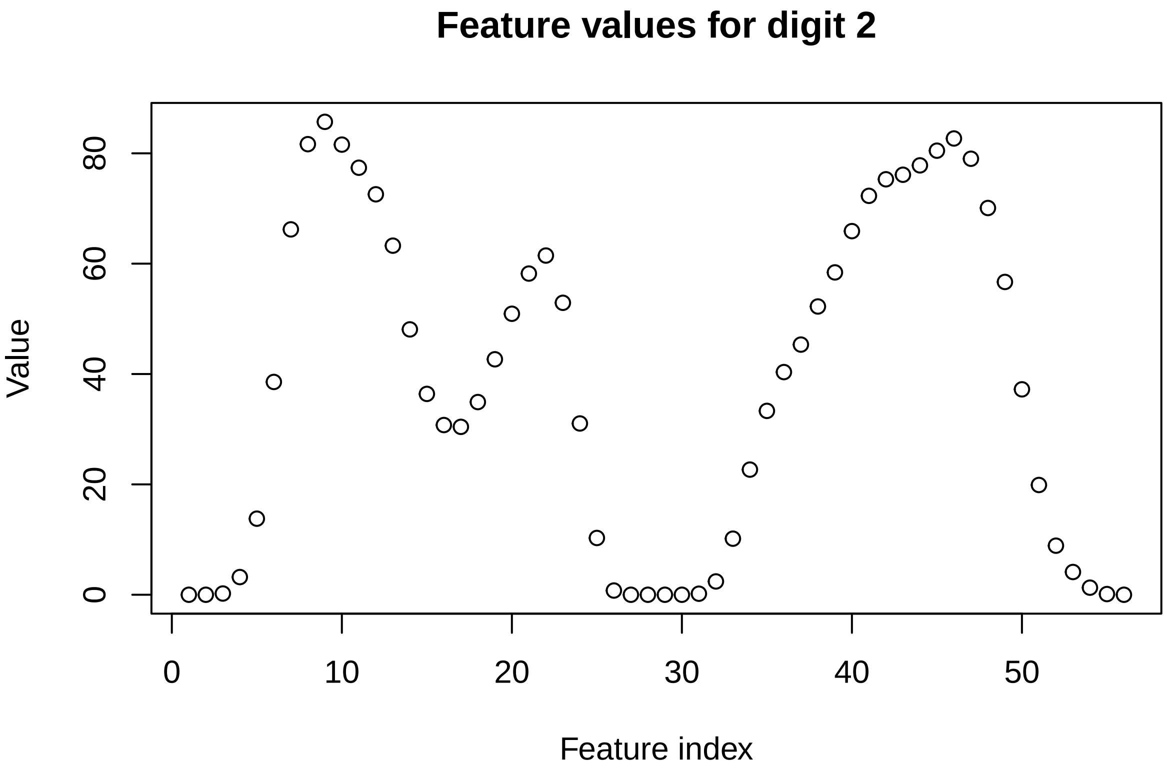
```
par(nrow=c(5,2))
```

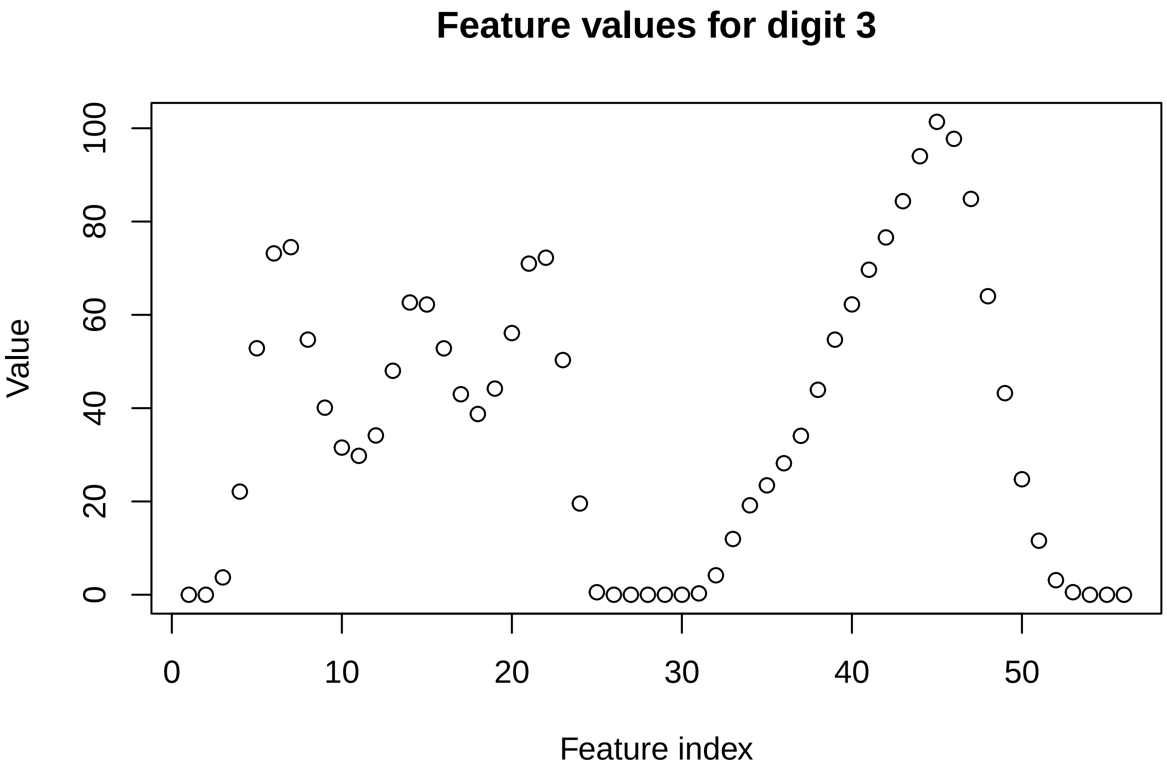
```
## Warning in par(nrow = c(5, 2)): "nrow" is not a graphical parameter
```

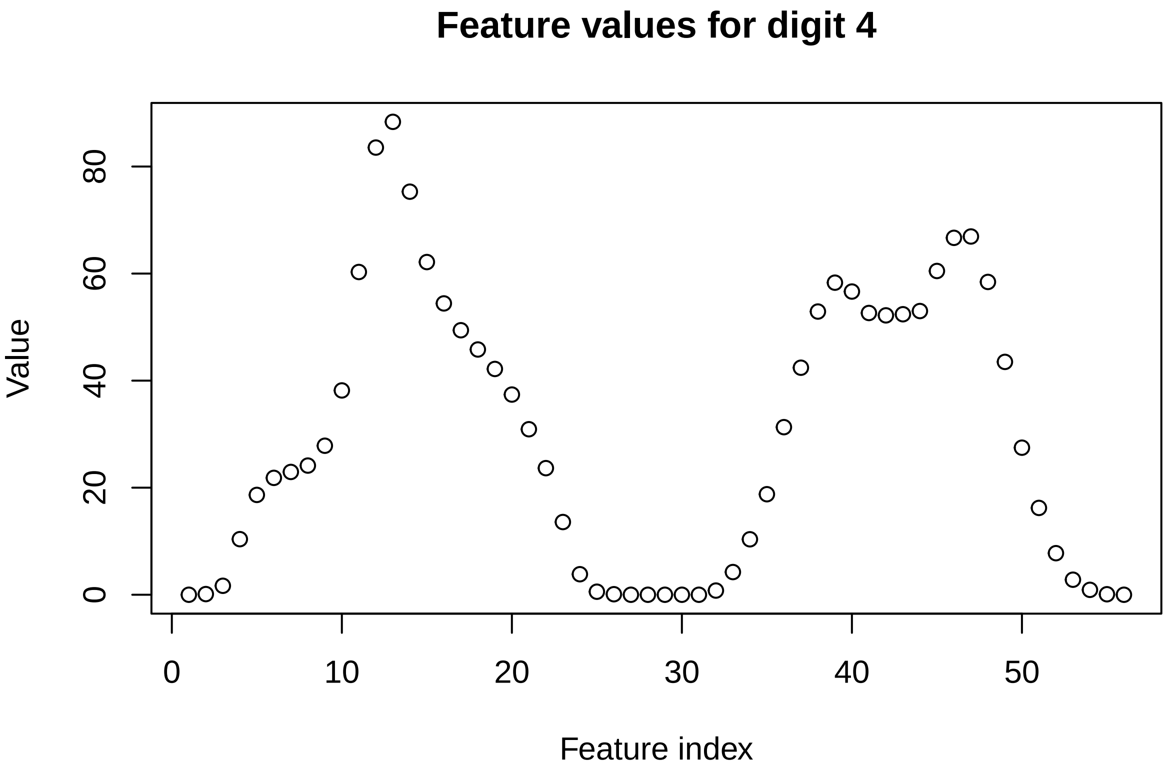
```
for (i in 1:10) {
  plot(fstats[i,], ylab = "Value", xlab = "Feature index")
  title(paste("Feature values for digit", toString(i-1)))
}
```

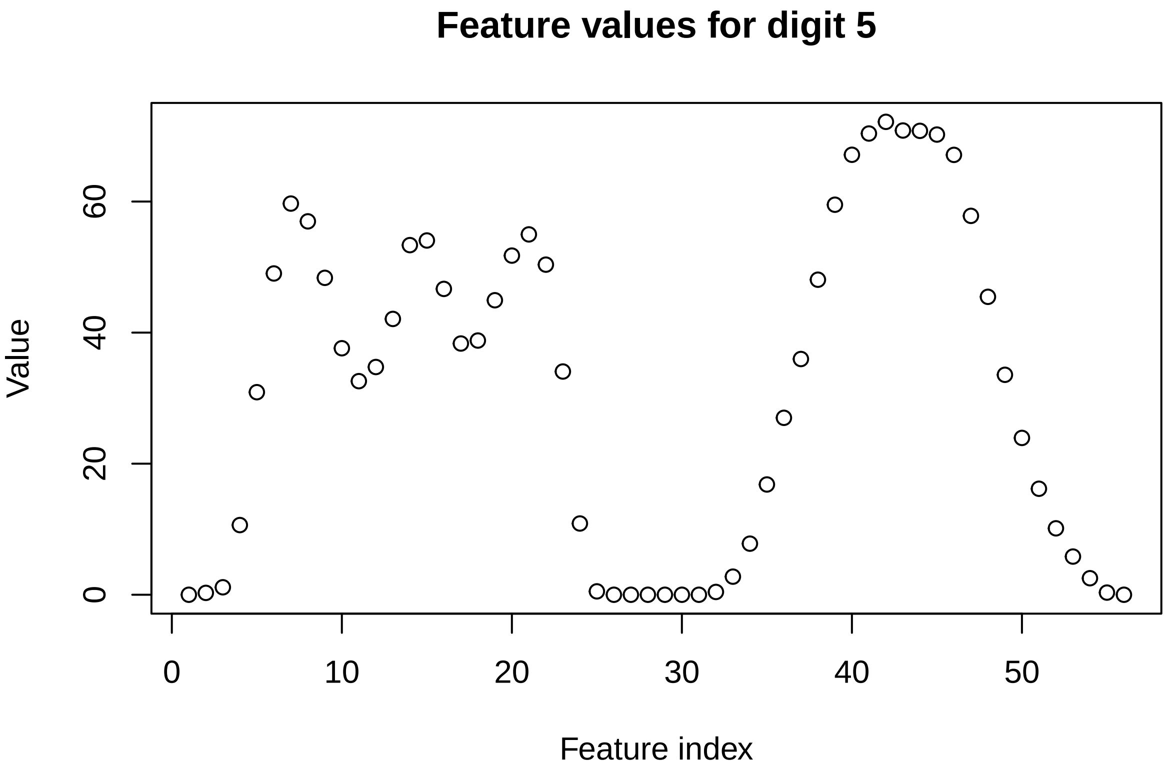


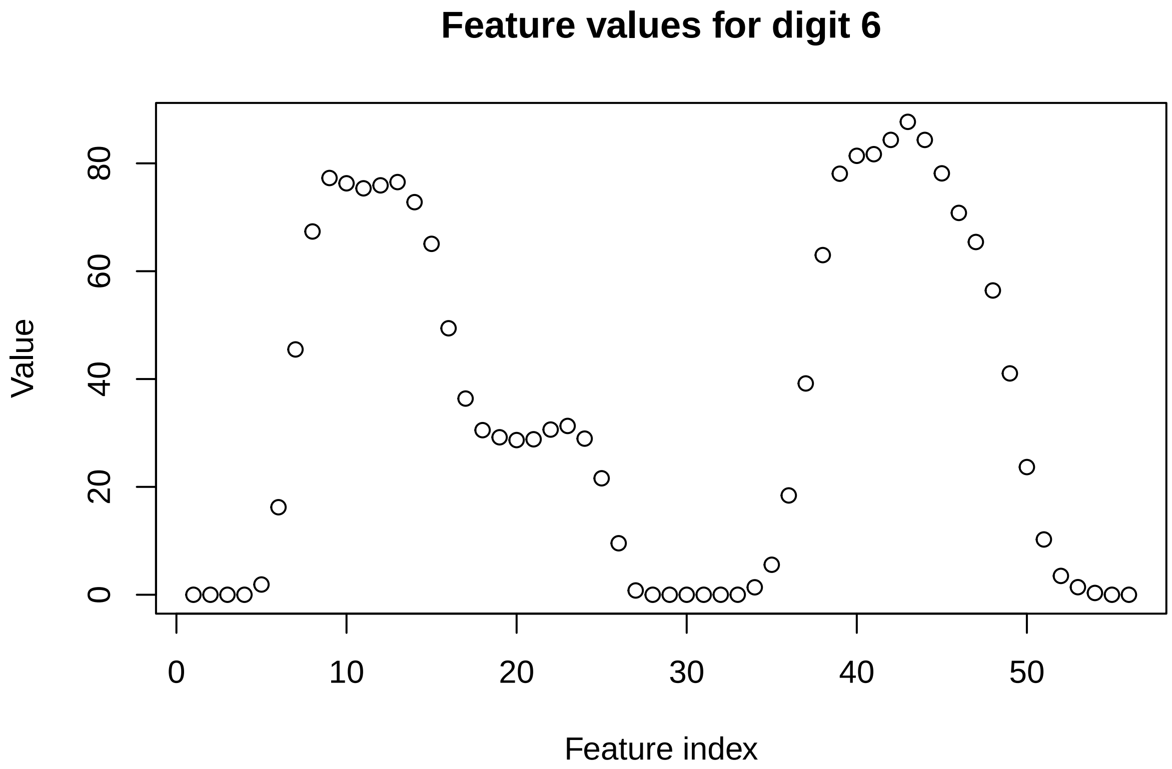

Feature values for digit 1
Value
0
20
40
60
80
100
0
10
20
30
40
50
Feature index

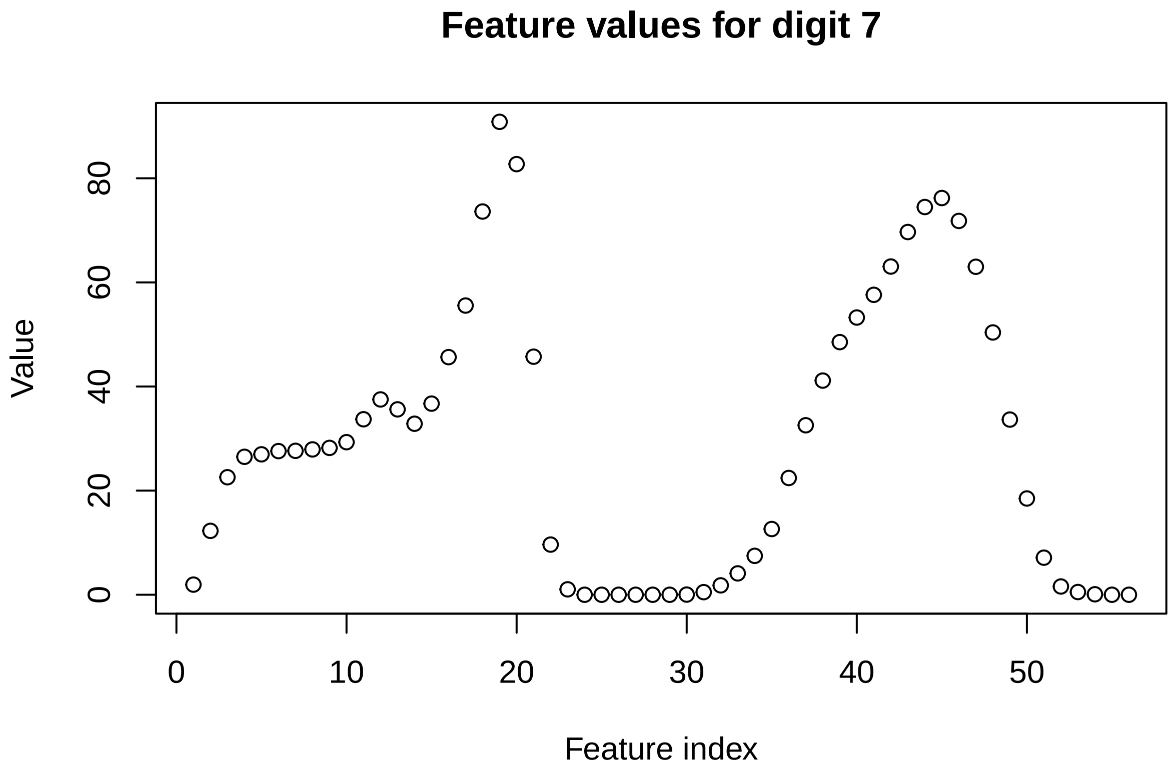

Feature values for digit 2
Value
0
20
40
60
80
0
10
20
30
40
50
Feature index

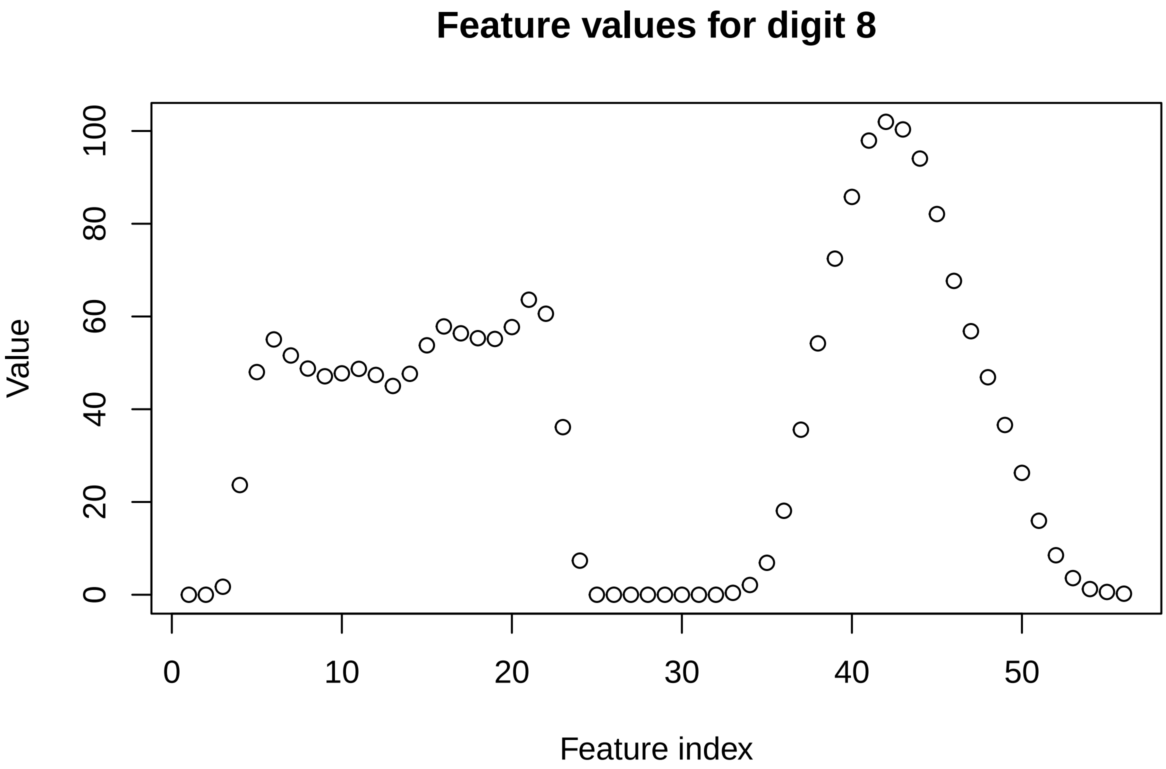

Feature values for digit 3
Value
0
20
40
60
80
100
0
10
20
30
40
50
Feature index

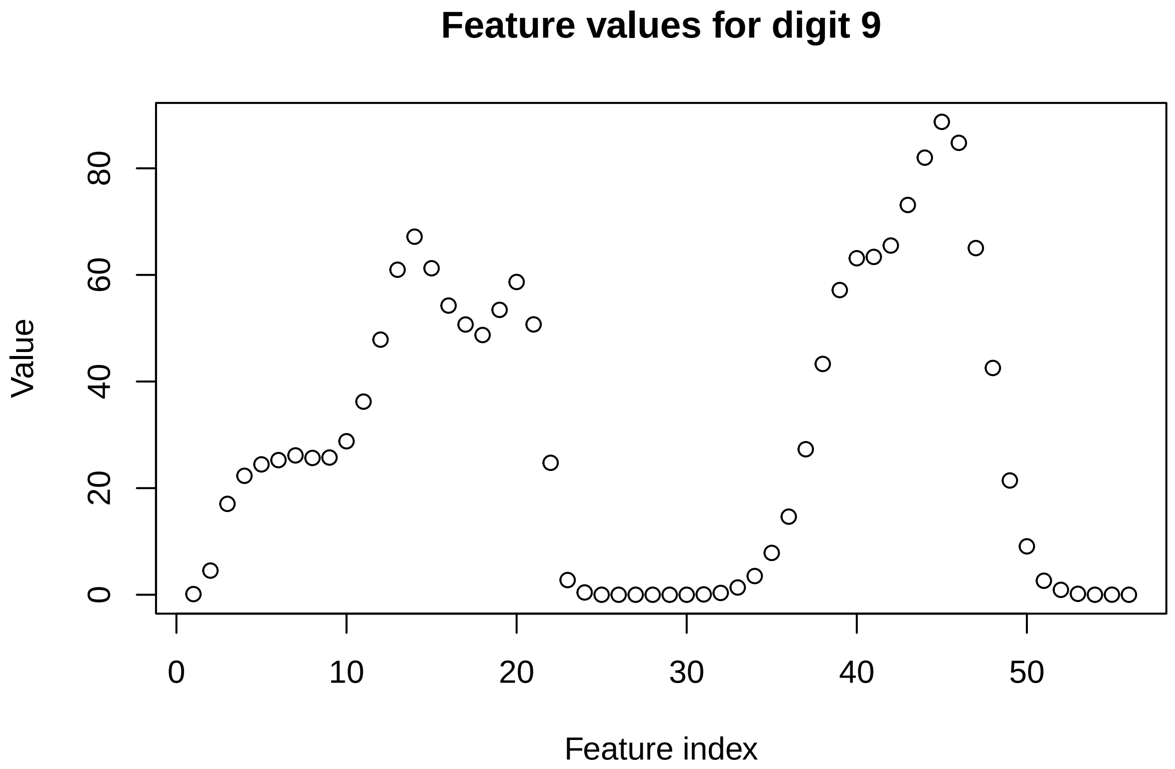

Feature values for digit 4
Value
0
20
40
60
80
0
10
20
30
40
50
Feature index


Feature values for digit 5
Value
0
20
40
60
0
10
20
30
40
50
Feature index


Feature values for digit 6
Value
0
20
40
60
80
0
10
20
30
40
50
Feature index


Feature values for digit 7
Value
0
20
40
60
80
0
10
20
30
40
50
Feature index


Feature values for digit 8
Value
0
20
40
60
80
100
0
10
20
30
40
50
Feature index


Feature values for digit 9
Value
0
20
40
60
80
0
10
20
30
40
50
Feature index