class08_mini_project

Steven Gan

2022-10-23

Table of contents

Exploratory data analysis

Preparing the data

```
fna.data <- "WisconsinCancer.csv"
wisc.df <- read.csv(fna.data, row.names=1)

head(wisc.df, 2)
```

```
       diagnosis radius_mean texture_mean perimeter_mean area_mean
842302         M       17.99        10.38          122.8      1001
842517         M       20.57        17.77          132.9      1326
       smoothness_mean compactness_mean concavity_mean concave.points_mean
842302         0.11840          0.27760         0.3001             0.14710
842517         0.08474          0.07864         0.0869             0.07017
       symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
842302        0.2419                0.07871    1.0950     0.9053        8.589
842517        0.1812                0.05667    0.5435     0.7339        3.398
       area_se smoothness_se compactness_se concavity_se concave.points_se
842302  153.40      0.006399        0.04904      0.05373           0.01587
842517   74.08      0.005225        0.01308      0.01860           0.01340
       symmetry_se fractal_dimension_se radius_worst texture_worst
842302     0.03003             0.006193        25.38         17.33
842517     0.01389             0.003532        24.99         23.41
       perimeter_worst area_worst smoothness_worst compactness_worst
842302           184.6       2019           0.1622            0.6656
842517           158.8       1956           0.1238            0.1866
       concavity_worst concave.points_worst symmetry_worst
842302          0.7119               0.2654         0.4601
842517          0.2416               0.1860         0.2750
       fractal_dimension_worst
842302                 0.11890
842517                 0.08902
```

```
wisc.data <- wisc.df[,-1]

diagnosis <- factor(wisc.df$diagnosis)
```

Exploratory data analysis

Q1: 569 observations in this dataset.

```
nrow(wisc.df)
```

```
[1] 569
```

Q2: 212 observations have malignant diagnosis.

```
sum(diagnosis == "M")
```

```
[1] 212
```

```
# table(wisc.df$diagnosis)
```

Q3: 10 features in the data have "_mean" suffix.

```
length(grep("_mean", colnames(wisc.df)))
```

```
[1] 10
```

```
# sum(grepl("_mean", colnames(wisc.df)))
```

Principle component analysis (PCA)

Performing PCA

```
colMeans(wisc.data)
```

```
           radius_mean            texture_mean          perimeter_mean 
          1.412729e+01            1.928965e+01            9.196903e+01 
             area_mean         smoothness_mean        compactness_mean 
          6.548891e+02            9.636028e-02            1.043410e-01 
        concavity_mean     concave.points_mean           symmetry_mean 
          8.879932e-02            4.891915e-02            1.811619e-01 
```

```
 fractal_dimension_mean               radius_se              texture_se
           6.279761e-02            4.051721e-01            1.216853e+00
           perimeter_se                 area_se           smoothness_se
           2.866059e+00            4.033708e+01            7.040979e-03
         compactness_se            concavity_se       concave.points_se
           2.547814e-02            3.189372e-02            1.179614e-02
            symmetry_se    fractal_dimension_se            radius_worst
           2.054230e-02            3.794904e-03            1.626919e+01
          texture_worst         perimeter_worst              area_worst
           2.567722e+01            1.072612e+02            8.805831e+02
       smoothness_worst       compactness_worst         concavity_worst
           1.323686e-01            2.542650e-01            2.721885e-01
   concave.points_worst          symmetry_worst fractal_dimension_worst
           1.146062e-01            2.900756e-01            8.394582e-02
```

```
apply(wisc.data, 2, sd)
```

```
            radius_mean            texture_mean          perimeter_mean
           3.524049e+00            4.301036e+00            2.429898e+01
              area_mean         smoothness_mean        compactness_mean
           3.519141e+02            1.406413e-02            5.281276e-02
         concavity_mean     concave.points_mean           symmetry_mean
           7.971981e-02            3.880284e-02            2.741428e-02
 fractal_dimension_mean               radius_se              texture_se
           7.060363e-03            2.773127e-01            5.516484e-01
           perimeter_se                 area_se           smoothness_se
           2.021855e+00            4.549101e+01            3.002518e-03
         compactness_se            concavity_se       concave.points_se
           1.790818e-02            3.018606e-02            6.170285e-03
            symmetry_se    fractal_dimension_se            radius_worst
           8.266372e-03            2.646071e-03            4.833242e+00
          texture_worst         perimeter_worst              area_worst
           6.146258e+00            3.360254e+01            5.693570e+02
       smoothness_worst       compactness_worst         concavity_worst
           2.283243e-02            1.573365e-01            2.086243e-01
   concave.points_worst          symmetry_worst fractal_dimension_worst
           6.573234e-02            6.186747e-02            1.806127e-02
```

```
wisc.pr <- prcomp(wisc.data, scale = TRUE)
summary(wisc.pr)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
                           PC8    PC9    PC10   PC11    PC12    PC13    PC14
Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 0.00523
Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812 0.98335
                          PC15    PC16    PC17    PC18    PC19    PC20   PC21
Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
                          PC22    PC23   PC24    PC25    PC26    PC27    PC28
Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987
Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005
Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997
                          PC29    PC30
Standard deviation     0.02736 0.01153
Proportion of Variance 0.00002 0.00000
Cumulative Proportion  1.00000 1.00000
```

Q4: 44.3% of the original variance is captured by the PC1.

```r
wisc.pr$sdev[1] ^ 2 / sum(wisc.pr$sdev ^ 2)
```

```
[1] 0.4427203
```

Q5: 3 PCs are required for at least 70% of variance.

```r
var <- wisc.pr$sdev ^ 2 / sum(wisc.pr$sdev ^ 2)

b = 0
for (i in 1:length(var)) {
  b <- b + var[i]
  if(b > 0.7) {
    print(i)
    break
  }
}
```

```
[1] 3
```

Q6: 7 PCs are required for at least 90% of variance.

```
b = 0
for (i in 1:length(var)) {
  b <- b + var[i]
  if(b > 0.9) {
    print(i)
    break
  }
}
```

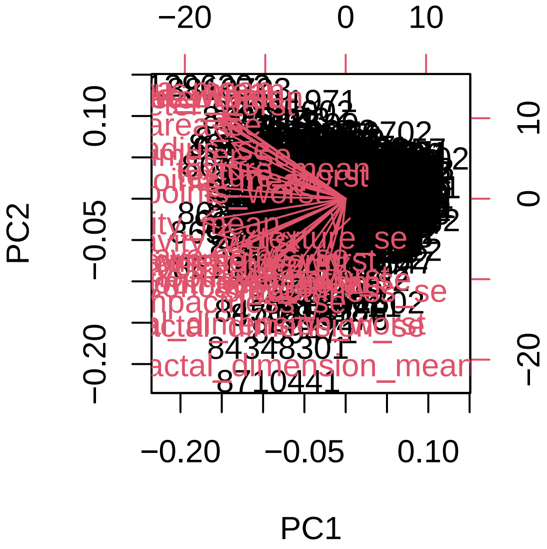
```
[1] 7
```

```
rm(b)
```

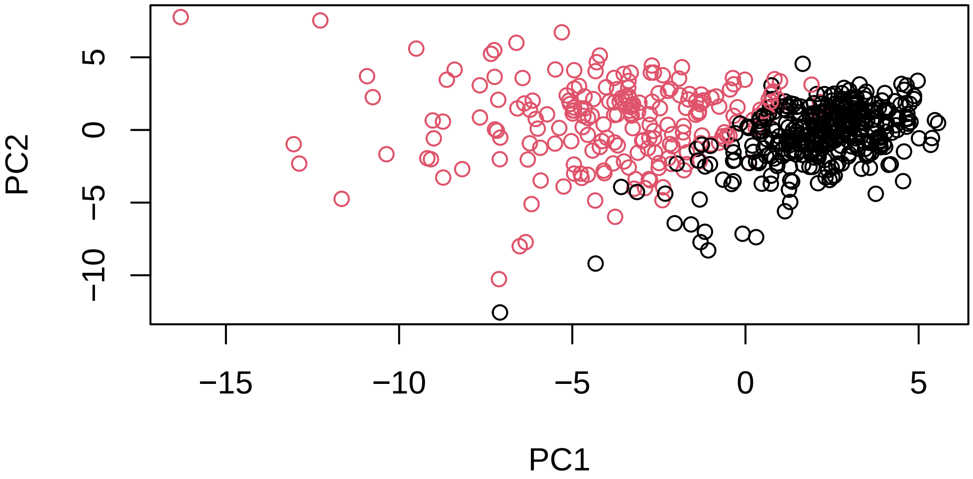
Interpreting PCA results

Q7: This is a mess. No distinguishable information and difficult to comprehend.

```
biplot(wisc.pr)
```

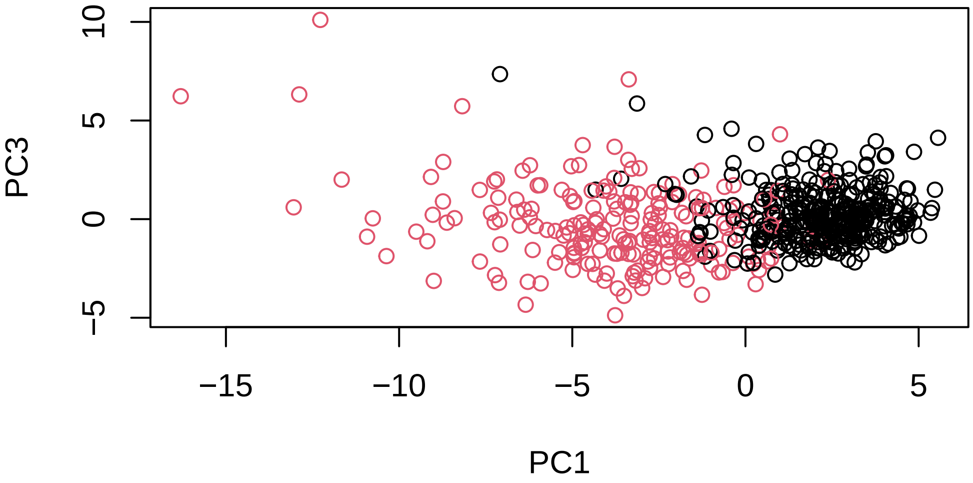


```r
plot(wisc.pr$x[,1], wisc.pr$x[,2], col = diagnosis,
     xlab = "PC1", ylab = "PC2")
```

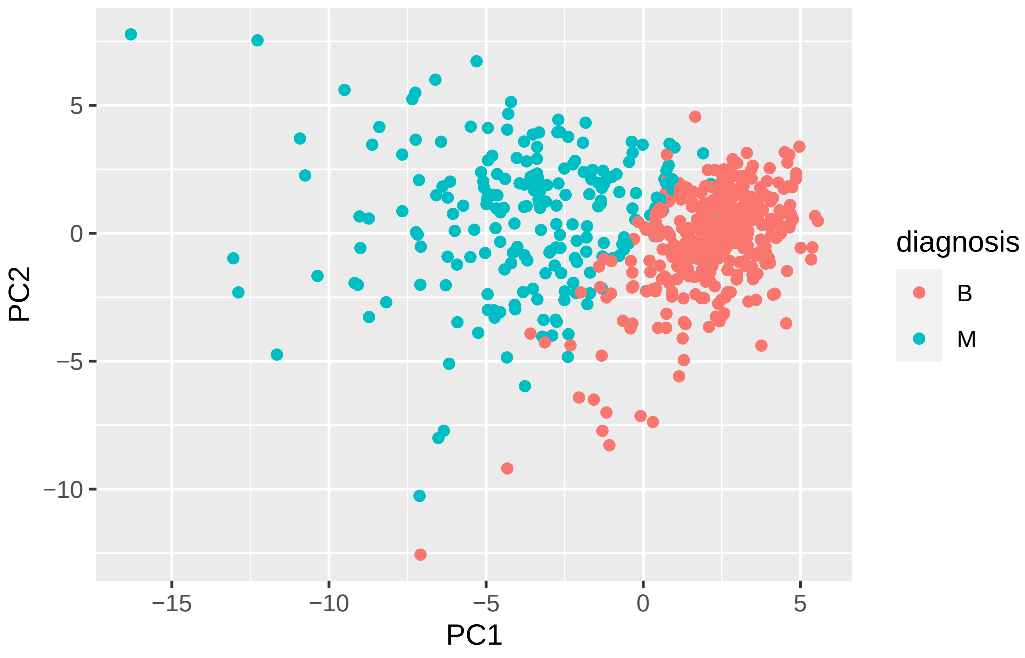


Q8: PC1 and PC2 did a better job in separating samples with Malignant or Benign cancers than PC1 and PC3, where more Malignant and Benign dots mixed with each others.

```r
plot(wisc.pr$x[,1], wisc.pr$x[,3], col = diagnosis ,
     xlab = "PC1", ylab = "PC3")
```



```r
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis

library(ggplot2)

ggplot(df) +
  aes(PC1, PC2, col = diagnosis) +
  geom_point()
```

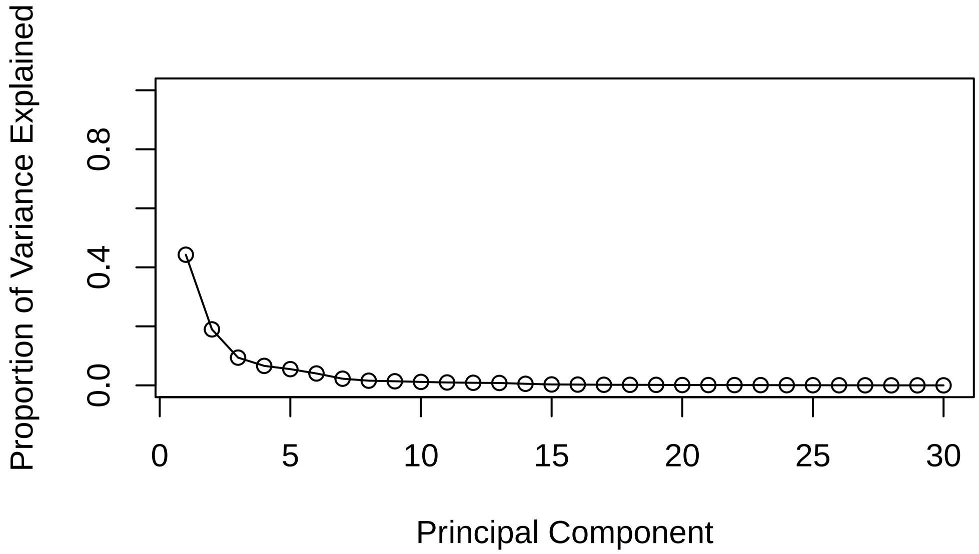


Variance explained

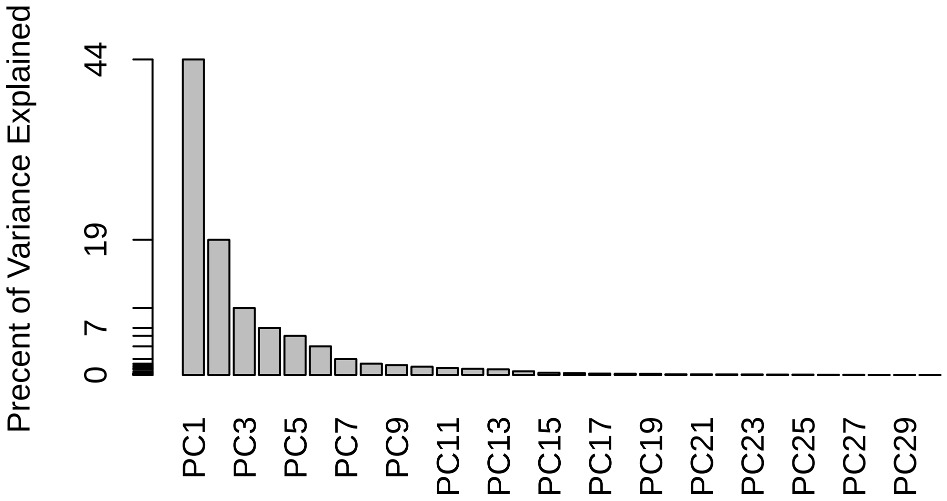
```
pr.var <- wisc.pr$sdev ^ 2
head(pr.var)
```

```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
pve <- pr.var/ sum(pr.var)

plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "o")
```



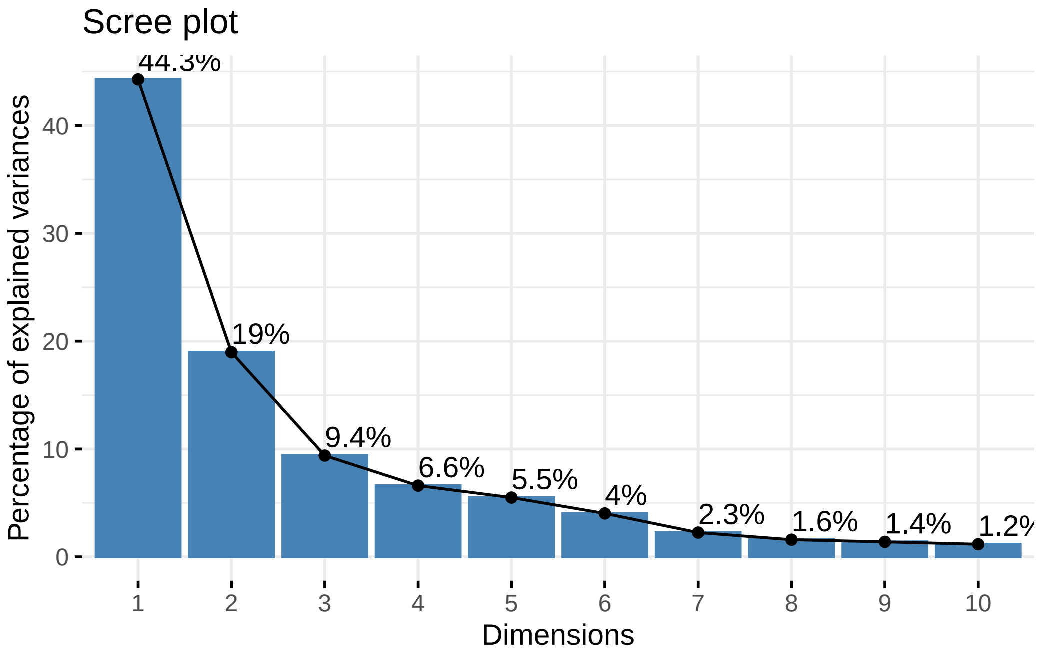
```
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg = paste0("PC",1:length(pve)), las = 2, axes = FALSE)
axis(2, at = pve, labels = round(pve, 2) * 100 )
```



```r
# install.packages("factoextra")
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
fviz_eig(wisc.pr, addlabels = TRUE)
```



Communicating PCA results

Q9: The "concave.points_mean" component of the loading vector of PC1.

```
wisc.pr$rotation["concave.points_mean",1]
```

```
[1] -0.2608538
```

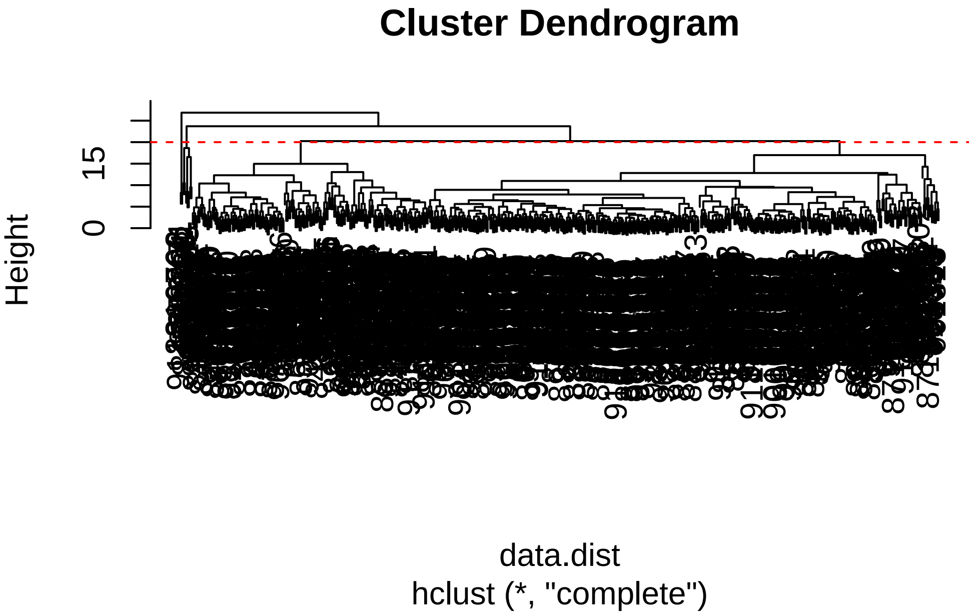
Hierarchical clustering

```
data.scaled <- scale(wisc.data)
data.dist <- dist(data.scaled)

wisc.hclust <- hclust(data.dist, "complete")
```

Results of hierarchical clustering

Q10: 4 clusters appears at about height is 20.

```
plot(wisc.hclust)
abline(wisc.hclust, h = 20, col = "red", lty = 2)
```



Selecting number of clusters

```
wisc.hclust.clusters <- cutree(wisc.hclust, k = 4)
table(wisc.hclust.clusters, diagnosis)
```

```
                    diagnosis
wisc.hclust.clusters   B   M
                   1  12 165
                   2   2   5
                   3 343  40
                   4   0   2
```

Q11: Analysis Results: From the evaluation table and the graph below, it can be easily shown that cutree at 4 is the best.

Method: Take the two clusters with highest patients numbers as either the predicted M or B samples, based on the actual diagnosis results of these two clusters, denoted as “1” (clustered

M) and “2” (clustered B). And all the rest clusters are considered as indecisive results, denoted as “3” (ind).

Three values are used to judge the results in each case: False-negative rate (fnR): P{M|2} = P{M2} / P{2} = n{M2} / n{2}
False-positive rate (fpR): P{B|1} = P{B1} / P{1} = n{B1} / n{1}
Indecisive rate (indR): P{3} = n{3} / n{Total}

```
library(dplyr)
```

```

Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(reshape2)

# clusDiag funtion identify the specific clusters correspond to diagnosis
clusDiag <- function(clusDiagTable, diag) {
  arra <- clusDiagTable %>%
    filter(diagnosis == diag) %>%
    arrange(desc(Freq)) %>%
    select(1) %>%
    slice(1)
  return(as.integer(arra))
}

# evaluation of fpR, fnR, and indR
evalErr_hclust <- function(hclust, k){
  hclust.clusters <- cutree(hclust, k = k)
  compare <- as.data.frame(table(hclust.clusters, diagnosis))

  clusM <- clusDiag(compare, "M")
  clusB <- clusDiag(compare, "B")

  if (clusM == clusB) {
    print(paste0("Unable to identify coresponding M/B clusters at cutree ", k))
    print("Major cluster are mixed with M/B samples, as below:")
    print(table(hclust.clusters, diagnosis)[clusB,])
    return(c(NA, NA, NA))
  }

  nM2 = filter(compare, diagnosis == "M" & hclust.clusters == clusB)$Freq
  nB1 = filter(compare, diagnosis == "B" & hclust.clusters == clusM)$Freq

  n2 = sum(filter(compare, hclust.clusters == clusB)$Freq)
  n1 = sum(filter(compare, hclust.clusters == clusM)$Freq)
  n3 = sum(filter(compare, hclust.clusters != clusM
                & hclust.clusters != clusB)$Freq)

  nTot = sum(compare$Freq)

  fnR <- nM2 / n2 * 100
  fpR <- nB1 / n1 * 100
  indR <- n3 / nTot * 100

  values <- c(fnR = fnR, fpR = fpR, indR = indR)
  return(values)
}

# evaluation with sequential cutree value k
evalErr_hclust_k <- function(hclust, start_k, end_k) {
  evalDf = NULL
  for (i in start_k:end_k) {
    values <- evalErr_hclust(hclust, i)
    evalDf <- rbind(evalDf, values)
    rownames(evalDf)[i - (start_k - 1)] = i
  }
  return(evalDf)
}
```

Analysis:

```
library(ggplot2)

wisc.hc.evalDf <- evalErr_hclust_k(wisc.hclust, 2, 10)
```

```
[1] "Unable to identify coresponding M/B clusters at cutree 2"
```

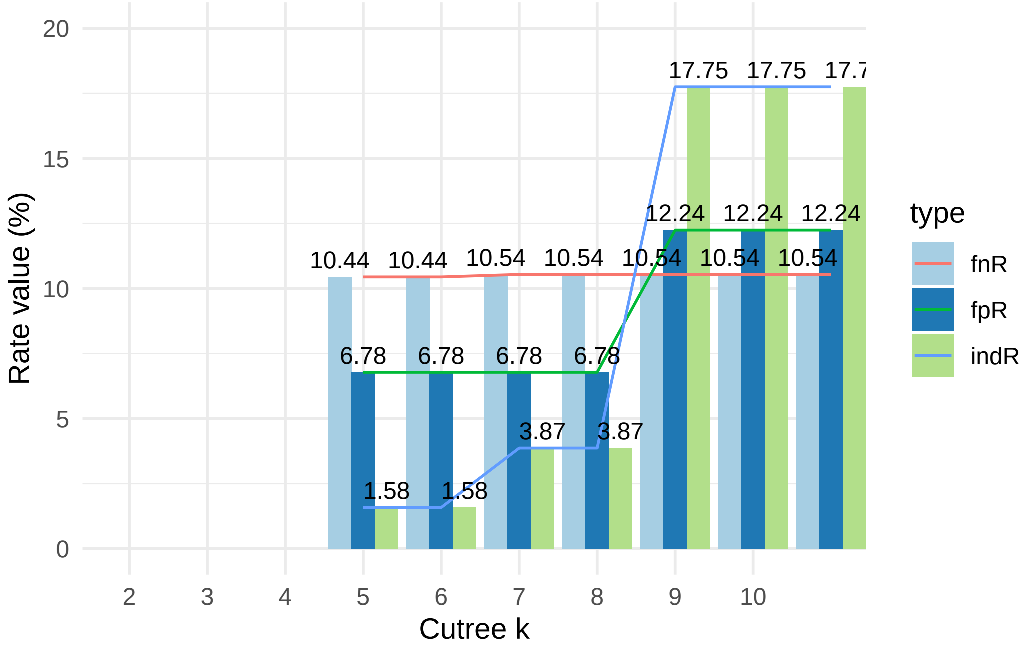
```
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
357 210
[1] "Unable to identify coresponding M/B clusters at cutree 3"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 205
```

```
print(wisc.hc.evalDf)
```

```
        fnR       fpR      indR
2        NA        NA        NA
3        NA        NA        NA
4  10.44386  6.779661  1.581722
5  10.44386  6.779661  1.581722
6  10.54054  6.779661  3.866432
7  10.54054  6.779661  3.866432
8  10.54054 12.244898 17.750439
9  10.54054 12.244898 17.750439
10 10.54054 12.244898 17.750439
```

```
#plotting
bar_hclust_eval <- function(hc.evalDf, range_y = 100) {
  hc.evalDf <- melt(hc.evalDf)
colnames(hc.evalDf) <- c("k", "type", "value")

ggplot(hc.evalDf, aes(fill = type, y = value, x = k)) +
  geom_bar(position = position_dodge(), stat = "identity") +
  geom_line(stat="identity", aes(color = type, y = value, x = k)) +
  labs(x = "Cutree k", y = "Rate value (%)") +
  scale_x_discrete(
    limits = factor(hc.evalDf$k[1]:hc.evalDf$k[length(hc.evalDf$k)])) +
  scale_y_continuous(limits = c(0, range_y)) +
  geom_text(aes(label = round(value, 2)),
            vjust = -0.5, position = position_dodge(0.9), size = 3) +
  scale_fill_brewer(palette = "Paired") +
  theme_minimal()
}

bar_hclust_eval(wisc.hc.evalDf, range_y = 20)
```

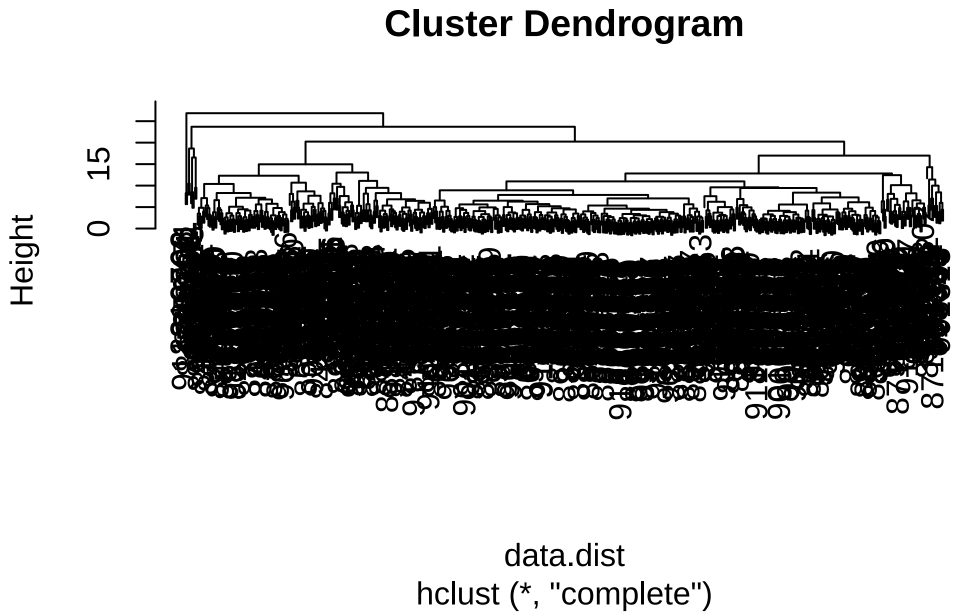
```
Warning: Removed 6 rows containing missing values (geom_bar).

Warning: Removed 6 row(s) containing missing values (geom_path).

Warning: Removed 6 rows containing missing values (geom_text).
```



Q12: As shown below, method “single” and “average” failed in separating M and B samples, under any Cutree between 2 to 10, with most of their samples mixed in the main cluster, while all other clusters containing only a few samples.

Both “ward.D2” and “complete” can separate B and M samples. The former separate samples Cutree of 2, while the later at 3. In comparing of error rate, however, “complete” method has an overall lower error rate, and false-negative rate, while has a relatively higher indecisive rate, due to 4 clusters it distinguished.

```
# Analysis on different clustering methods
hclust_evalErrAna <- function
(data.dist, start_k, end_k,
 method = c("complete", "single", "average", "ward.D2")) {
  for (i in 1:length(method)) {
    hclust.i <- hclust(data.dist, method[i])
    plot(hclust.i)

    hc.evalDf.i <- evalErr_hclust_k(hclust.i, start_k, end_k)

    if (sum(!is.na(hc.evalDf.i)) == 0) {
      print(paste0("No acceptable clusters for M/B separation under ",
                   method[i], "-based clustering analysis"))
      next
    }
    print(bar_hclust_eval(hc.evalDf.i, range_y = 60))
  }
}

hclust_evalErrAna(data.dist, 2, 10)
```



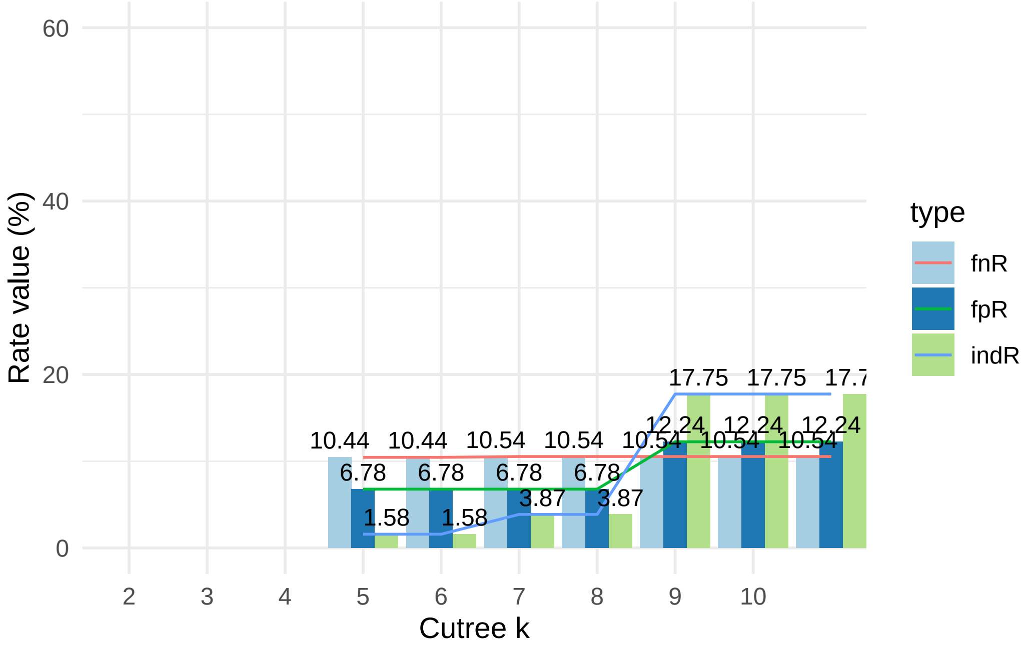
```
[1] "Unable to identify coresponding M/B clusters at cutree 2"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
357 210
[1] "Unable to identify coresponding M/B clusters at cutree 3"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
```

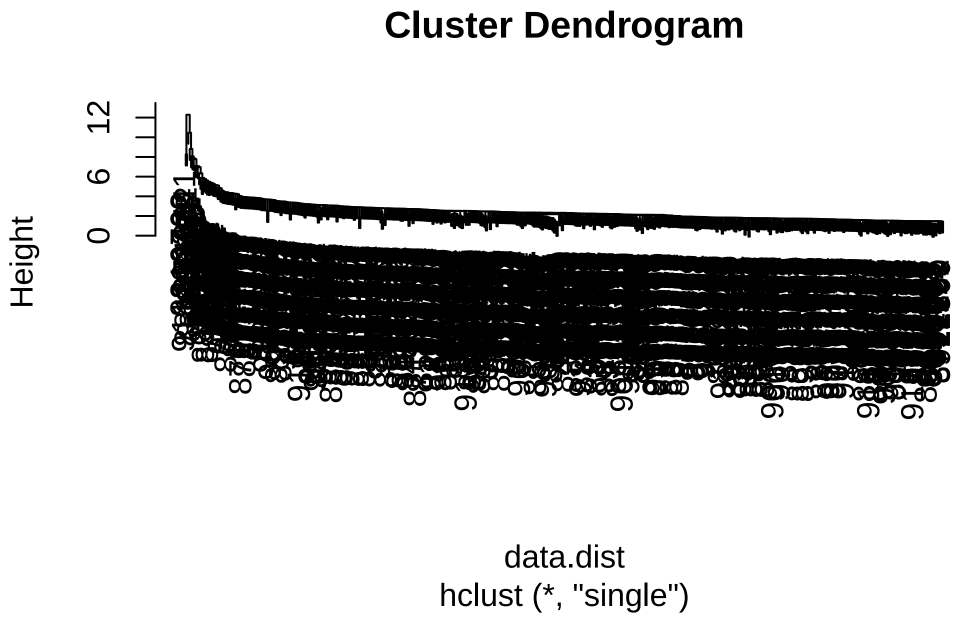
```
355 205
```

```
Warning: Removed 6 rows containing missing values (geom_bar).
```

```
Warning: Removed 6 row(s) containing missing values (geom_path).
```

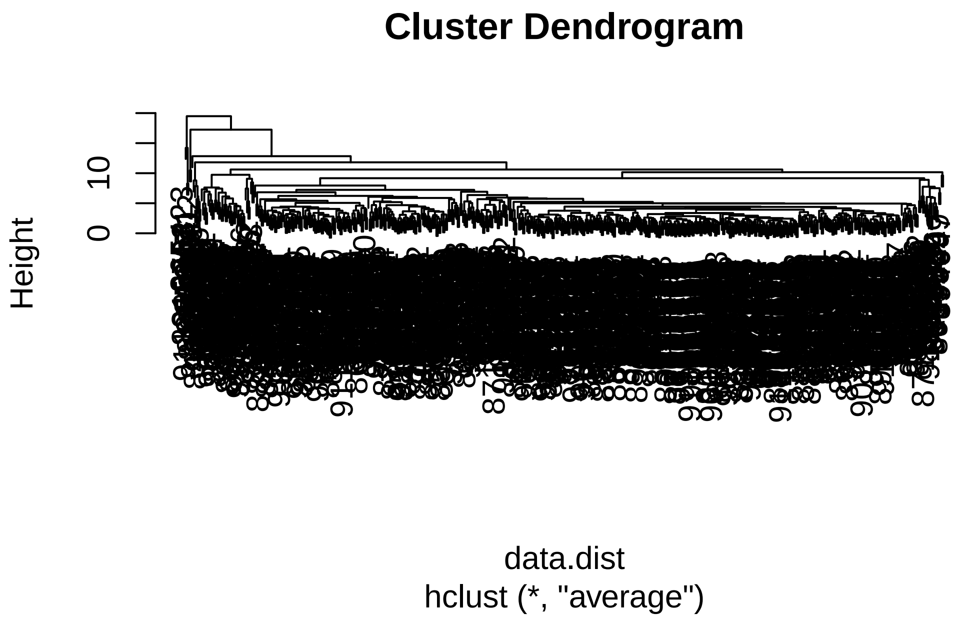
```
Warning: Removed 6 rows containing missing values (geom_text).
```





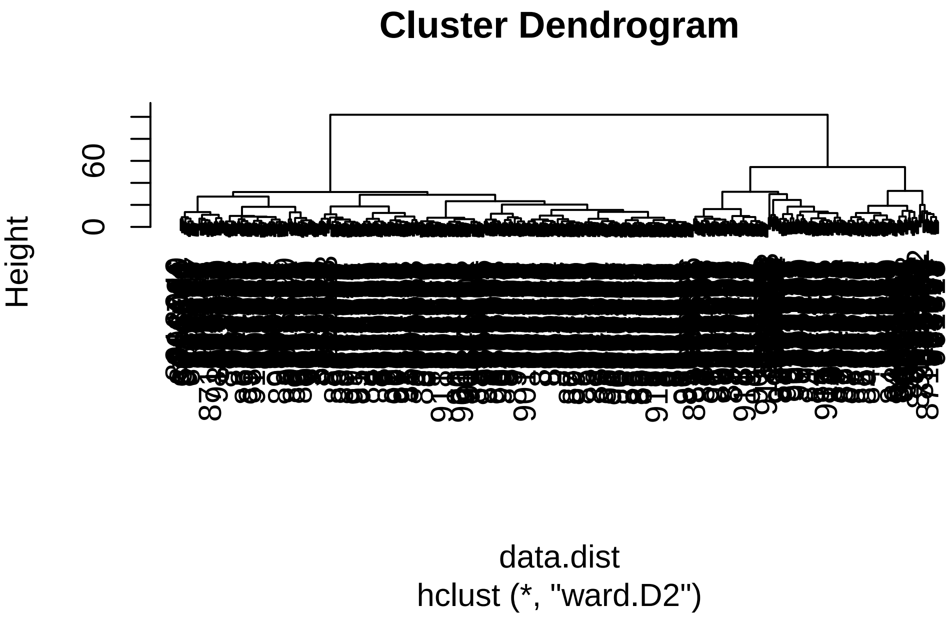
```
[1] "Unable to identify coresponding M/B clusters at cutree 2"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
357 210
[1] "Unable to identify coresponding M/B clusters at cutree 3"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
356 210
[1] "Unable to identify coresponding M/B clusters at cutree 4"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
356 209
[1] "Unable to identify coresponding M/B clusters at cutree 5"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
356 209
[1] "Unable to identify coresponding M/B clusters at cutree 6"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
356 208
[1] "Unable to identify coresponding M/B clusters at cutree 7"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
```

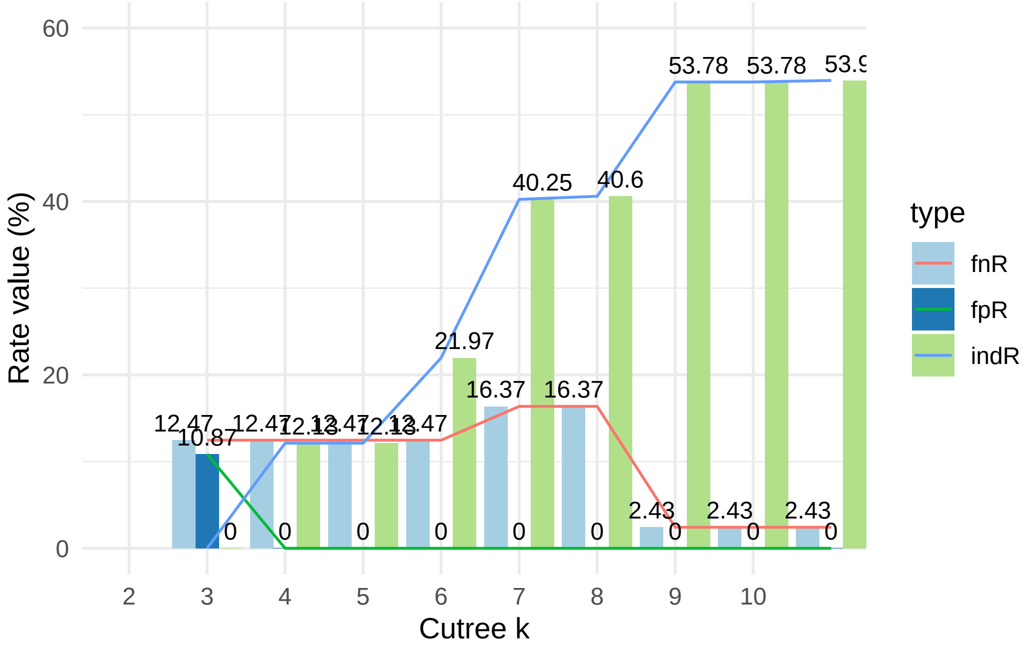
```
356 207
[1] "Unable to identify coresponding M/B clusters at cutree 8"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 207
[1] "Unable to identify coresponding M/B clusters at cutree 9"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 206
[1] "Unable to identify coresponding M/B clusters at cutree 10"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 205
[1] "No acceptable clusters for M/B separation under single-based clustering analysis"
```



```
[1] "Unable to identify coresponding M/B clusters at cutree 2"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
357 209
[1] "Unable to identify coresponding M/B clusters at cutree 3"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
```

```
355 209
[1] "Unable to identify coresponding M/B clusters at cutree 4"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 209
[1] "Unable to identify coresponding M/B clusters at cutree 5"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 208
[1] "Unable to identify coresponding M/B clusters at cutree 6"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 202
[1] "Unable to identify coresponding M/B clusters at cutree 7"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 162
[1] "Unable to identify coresponding M/B clusters at cutree 8"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
355 162
[1] "Unable to identify coresponding M/B clusters at cutree 9"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
353 162
[1] "Unable to identify coresponding M/B clusters at cutree 10"
[1] "Major cluster are mixed with M/B samples, as below:"
  B   M
353 162
[1] "No acceptable clusters for M/B separation under average-based clustering analysis"
```

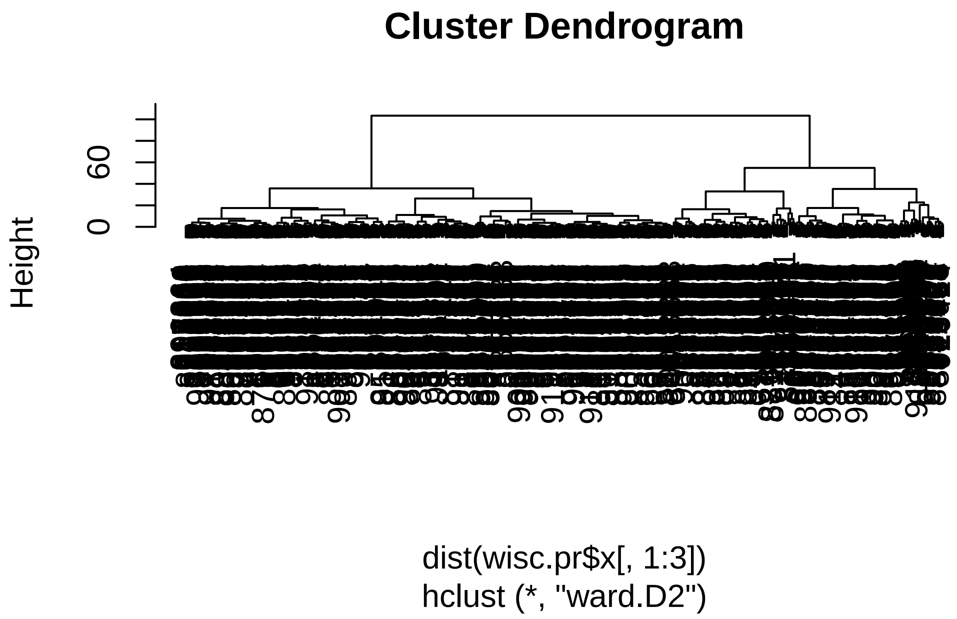

Cluster Dendrogram
Height
0
60
data.dist
hclust (*, "ward.D2")


Rate value (%)
0
20
40
60
Cutree k
2
3
4
5
6
7
8
9
10
12.47
10.87
12.47
12.13
12.47
12.13
12.47
21.97
16.37
16.37
40.25
40.6
53.78
53.78
53.9
2.43
2.43
2.43
0
0
0
0
0
0
0
0
0
type
fnR
fpR
indR

Combining methods

Clustering on PCA results

```
wisc.pr.hclust <- hclust(dist(wisc.pr$x[, 1:3]), "ward.D2")
plot(wisc.pr.hclust)
```



```
grps <- cutree(wisc.pr.hclust, k = 2)
table(grps)
```

```
grps
  1   2 
203 366 
```

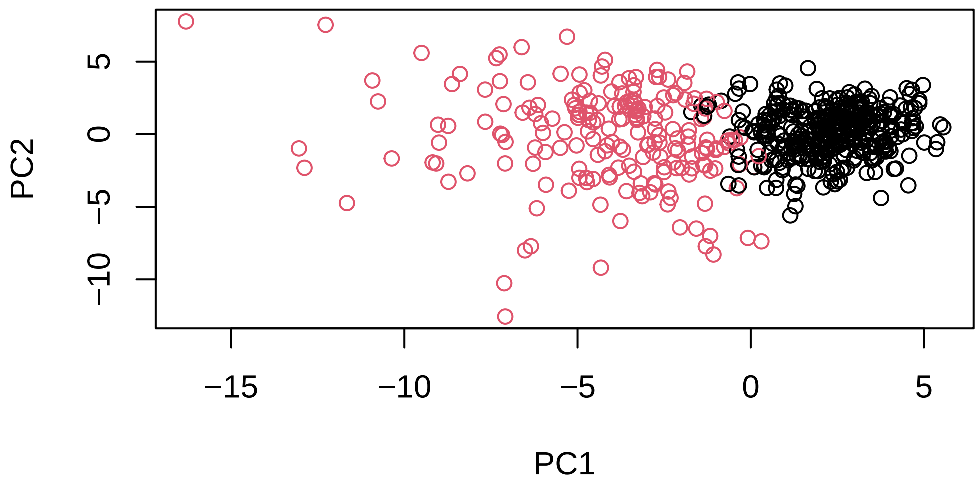
```
table(diagnosis, grps)
```

```
         grps
diagnosis   1   2
        B  24 333
        M 179  33
```

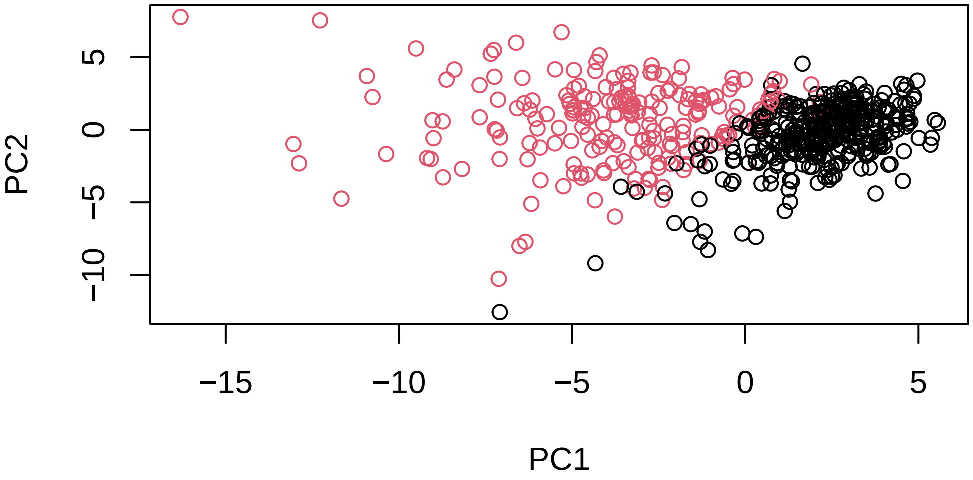
```r
g <- as.factor(grps)
levels(g)
```

```
[1] "1" "2"
```

```r
g <- relevel(g, 2)

plot(wisc.pr$x[,1:2], col = g)
```



```r
plot(wisc.pr$x[,1:2], col = diagnosis)
```



```
#library(rgl)

#plot3d(wisc.pr$x[,1:3], xlab = "PC1", ylab = "PC2", zlab = "PC3",
#        cex = 1.5, size = 1, type = "s", col = grps)
#rglwidget(width = 400, height = 400)
```

```
## Use the distance along the first 7 PCs for clustering i.e. wisc.pr$x[, 1:7]
wisc.pr.hclust <- hclust(dist(wisc.pr$x[, 1:7]), method = "ward.D2")
wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k = 2)
```

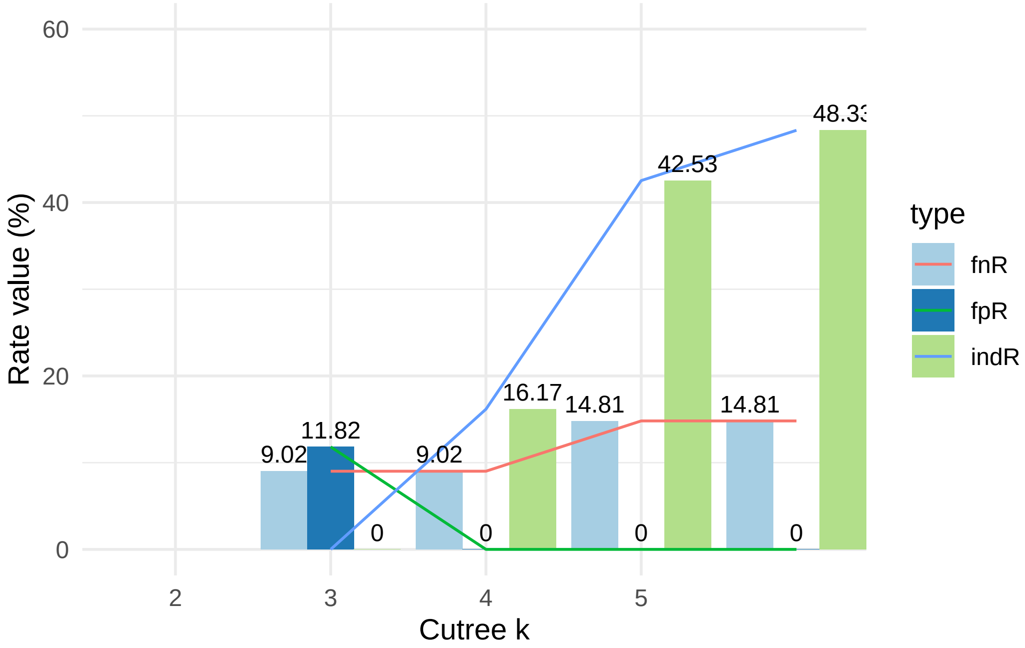
Q13: It works only a little bit better than hclust with 3 PCs, with only 52 incorrect assigned data in comparing to 57. As shown in the Error rate graph, similar results can be obtained with cutree is 2. However, I would argue that this does not necessary guarantee that higher PCs for htclust should result in lower error rate, as the error rate in 30 PCs analysis is surprisingly higher than 3 and 7 PCs analysis.

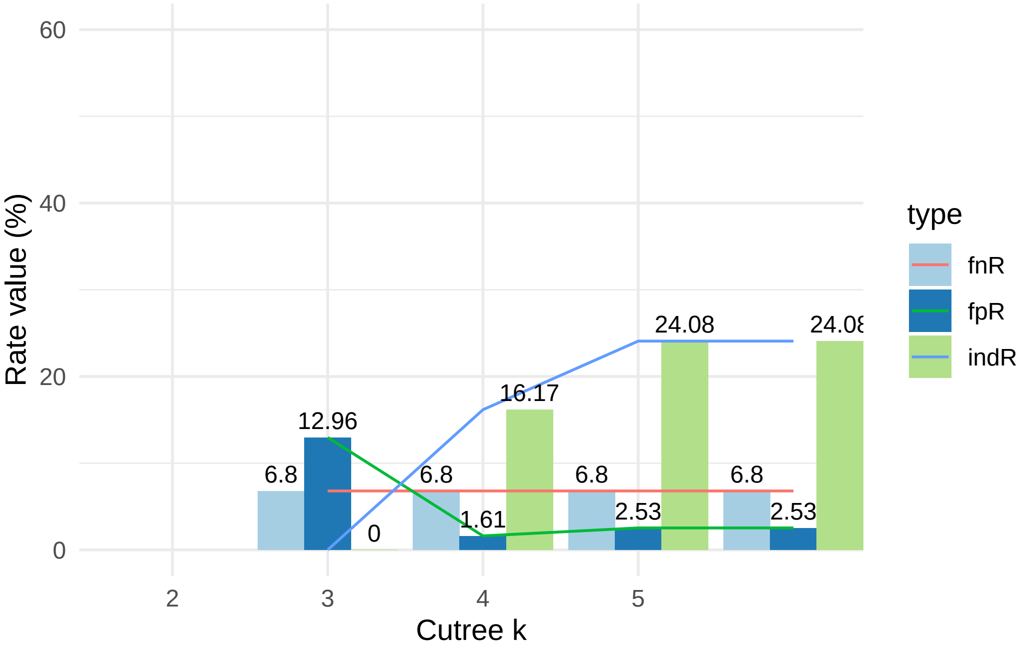
```
table(wisc.pr.hclust.clusters, diagnosis)
```

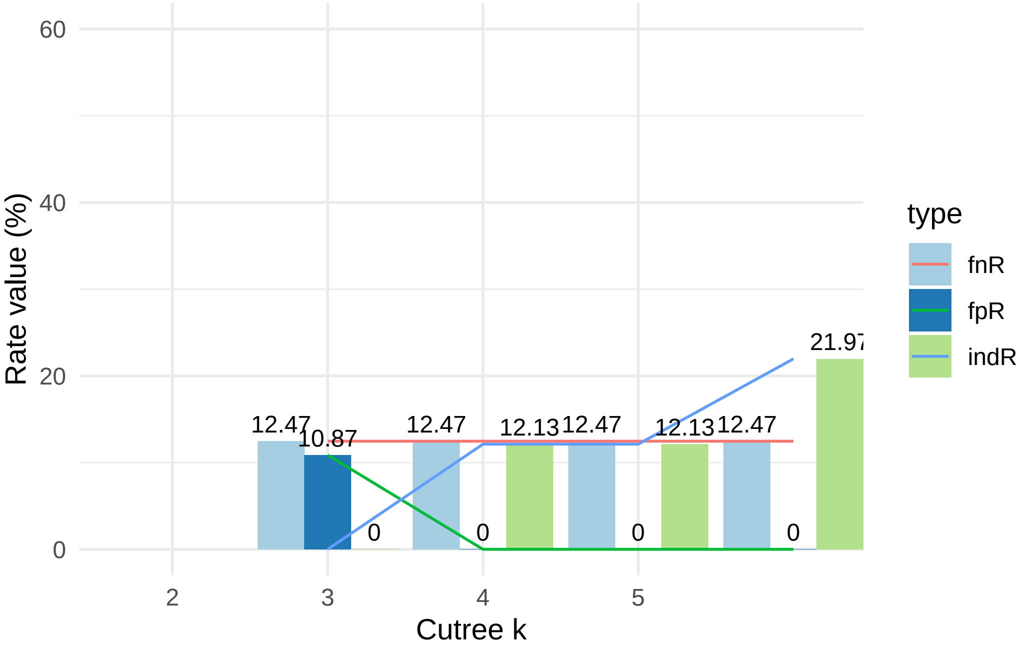
```
                       diagnosis
wisc.pr.hclust.clusters   B   M
                      1  28 188
```

```
            2 329   24
```

```
#Analysis on different PCs
for (i in c(3, 7, 30)) {
  wisc.pr.hclust.i <- hclust(dist(wisc.pr$x[, 1:i]), method = "ward.D2")
  wisc.pr.hc.evalDf.i <- evalErr_hclust_k(wisc.pr.hclust.i, 2, 5)

  if (sum(!is.na(wisc.pr.hc.evalDf.i)) == 0) {
    print(paste0("No acceptable clusters for M/B separation under ",
                 method[i], "-based clustering analysis"))
    next
  }

  print(bar_hclust_eval(wisc.pr.hc.evalDf.i, range_y = 60))
}
```







Q14: No significant difference to previous hclust data.

```
table(wisc.hclust.clusters, diagnosis)
```

```
                   diagnosis
wisc.hclust.clusters   B   M
                   1  12 165
                   2   2   5
                   3 343  40
                   4   0   2
```

Sensitivity/Specificity

Q15: Based on the graph below, the best sensitivity, 88.68%, achieved at 7 PCs analysis using method “ward.D2” with Cutree at k = 2. The best specificity, however, if stick to the data, would be near 100% when cutree goes to about 10. Yet this is merely a play with the figures, as there will be only a few samples in the clusters when cutree reaches 10. Faily speaking, the highest specificity should be about 93.2%, achieved at 7 PCs clustering using “ward.D2” methods.

Method: (Modify previous evaluation function) Sensitivity (senR): TP / TP + FN (M1 / M1 + M2) Specificity (speR): TN / TN + FN (B2 / B2 + M2)

Generalized complete Functions:

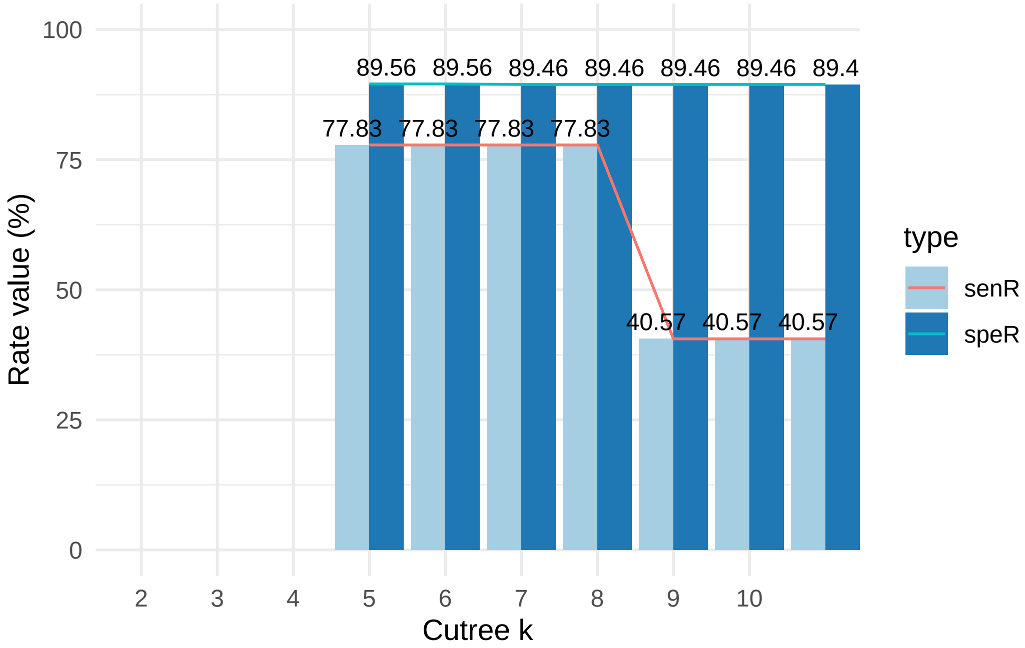
```
library(dplyr)
library(reshape2)

# clusDiag funtion identify the specific clusters correspond to diagnosis
clusDiag <- function(clusDiagTable, diag) {
  arra <- clusDiagTable %>%
    filter(diagnosis == diag) %>%
    arrange(desc(Freq)) %>%
    select(1) %>%
    slice(1)
  return(as.integer(arra))
}

# evaluation of Sensitivity and Specificity or Error rate
eval_hclust <- function (hclust, k, evalMethod, logs = F) {

  if (evalMethod != "SS" & evalMethod != "Err")
    {return(print("Error in evalMethod"))}
```

```
hclust.clusters <- cutree(hclust, k = k)
compare <- as.data.frame(table(hclust.clusters, diagnosis))

clusM <- clusDiag(compare, "M")
clusB <- clusDiag(compare, "B")


if (clusM == clusB) {
  if (logs == T) {
    print(paste0("Unable to identify M/B clusters at cutree ", k))
    print("Major cluster are mixed with M/B samples, as below:")
    print(table(hclust.clusters, diagnosis)[clusB,])
  }
  if (evalMethod == "SS") {return(c(NA, NA))}
  if (evalMethod == "Err") {return(c(NA, NA, NA))}
}

if (evalMethod == "SS") {
  nM1 = filter(compare, diagnosis == "M" & hclust.clusters == clusM)$Freq
  nB2 = filter(compare, diagnosis == "B" & hclust.clusters == clusB)$Freq

  nM = sum(filter(compare, diagnosis == "M")$Freq)
  n2 = sum(filter(compare, hclust.clusters == clusB)$Freq)

  senR <- nM1 / nM * 100
  speR <- nB2 / n2 * 100

  values <- c(senR = senR, speR = speR)
  return(values)
}

if (evalMethod == "Err") {
  nM2 = filter(compare, diagnosis == "M" & hclust.clusters == clusB)$Freq
  nB1 = filter(compare, diagnosis == "B" & hclust.clusters == clusM)$Freq

  n2 = sum(filter(compare, hclust.clusters == clusB)$Freq)
  n1 = sum(filter(compare, hclust.clusters == clusM)$Freq)
  n3 = sum(filter(compare, hclust.clusters != clusM
                & hclust.clusters != clusB)$Freq)

  nTot = sum(compare$Freq)
```

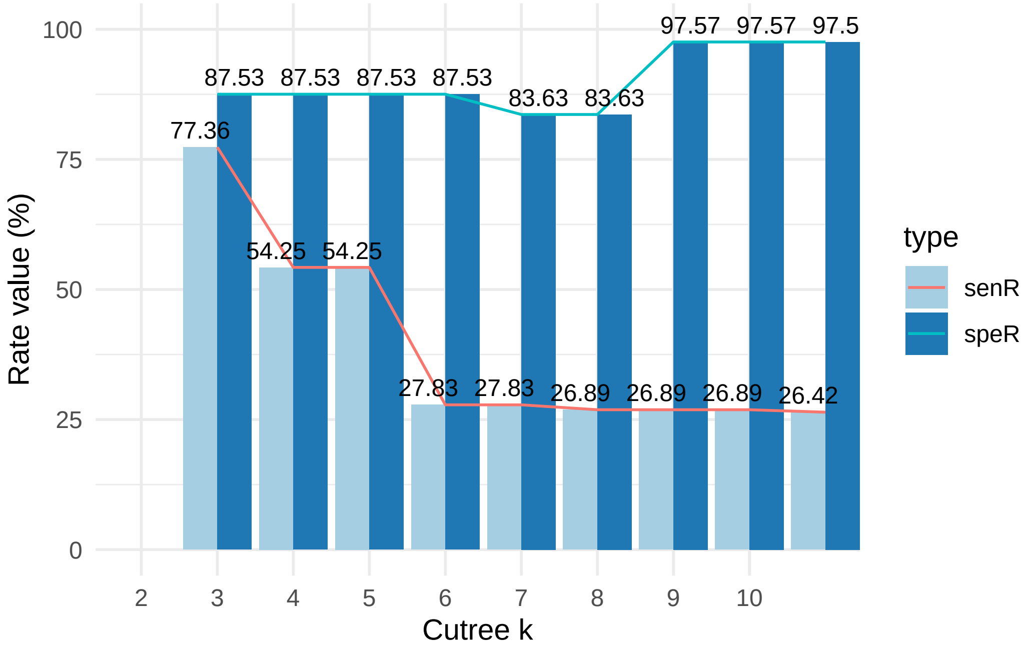
```
    fnR <- nM2 / n2 * 100
    fpR <- nB1 / n1 * 100
    indR <- n3 / nTot * 100

    values <- c(fnR = fnR, fpR = fpR, indR = indR)
    return(values)
  }
}

# evaluation with sequential cutree value k
eval_hclust_k <- function(hclust, start_k, end_k, evalMethod, logs = F) {
  evalDf = NULL
  for (i in start_k:end_k) {
    values <- eval_hclust(hclust, i, evalMethod, logs)
    evalDf <- rbind(evalDf, values)
    rownames(evalDf)[i - (start_k - 1)] = i
  }
  return(evalDf)
}

#plotting
bar_hclust_eval <- function(hc.evalDf, range_y = 100) {
  hc.evalDf <- melt(hc.evalDf)
colnames(hc.evalDf) <- c("k", "type", "value")

ggplot(hc.evalDf, aes(fill = type, y = value, x = k)) +
  geom_bar(position = position_dodge(), stat = "identity") +
  geom_line(stat="identity", aes(color = type, y = value, x = k)) +
  labs(x = "Cutree k", y = "Rate value (%)") +
  scale_x_discrete(
    limits = factor(hc.evalDf$k[1]:hc.evalDf$k[length(hc.evalDf$k)])) +
  scale_y_continuous(limits = c(0, range_y)) +
  geom_text(aes(label = round(value, 2)),
            vjust = -0.5, position = position_dodge(0.9), size = 3) +
  scale_fill_brewer(palette = "Paired") +
  theme_minimal()
}

# Analysis on different clustering methods
hclust_evalAna <- function (
    data.dist, start_k = 2, end_k = 10,
    method = c("complete", "single", "average", "ward.D2"),
    dendroPlot = F, evalMethod, logs = F, range_y = 100) {

  for (i in 1:length(method)) {
    hclust.i <- hclust(data.dist, method[i])
    if (dendroPlot == T) {plot(hclust.i)}

    hc.evalDf.i <- eval_hclust_k(hclust.i, start_k, end_k, evalMethod, logs)

    if (sum(!is.na(hc.evalDf.i)) == 0) {
      print(paste0("No acceptable clusters for M/B separation under ",
                   method[i], "-based clustering analysis"))
      next
    }
    print(paste0("Clustering Method: ", method[i]))
    print(bar_hclust_eval(hc.evalDf.i, range_y))
  }
}

#Analysis on different PCs
hclustPCs_evalAna <- function (
    pr, start_k = 2, end_k = 10, PCs = c(3, 7, 30), method = c("ward.D2"),
    dendroPlot = F, evalMethod, logs = F, range_y = 100) {

  for (i in PCs) {
    for (j in 1:length(method)) {
      pr.hclust.i <- hclust(dist(pr$x[, 1:i]), method[j])
      if (dendroPlot == T) {plot(pr.hclust.i)}
      pr.hc.evalDf.i <- eval_hclust_k(pr.hclust.i, start_k, end_k,
                                      evalMethod, logs)

      if (sum(!is.na(pr.hc.evalDf.i)) == 0) {
        print(paste0("No acceptable clusters for M/B separation under Method: ",
                     method[j], " for ", i, " PCs"))
        next
      }
      print(paste0("PCs: ", i, " Clustering method: ", method[j]))
      print(bar_hclust_eval(pr.hc.evalDf.i, range_y))
    }
  }
}
```

```
#Test 1: Q11
#wisc.hclust %>%
#  eval_hclust_k(2, 10, "Err") %>%
#  bar_hclust_eval(range_y = 20)

#Test 2: Q12
#data.dist %>%
#  hclust_evalAna(2, 10, dendroPlot = T, evalMethod = "Err", range_y = 60)

#Test 3: Q13
#wisc.pr %>%
#  hclustPCs_evalAna(2, 5, evalMethod = "Err", range_y = 60)

data.dist %>%
  hclust_evalAna(2, 10, evalMethod = "SS")
```

```
[1] "Clustering Method: complete"

Warning: Removed 4 rows containing missing values (geom_bar).

Warning: Removed 4 row(s) containing missing values (geom_path).

Warning: Removed 4 rows containing missing values (geom_text).
```

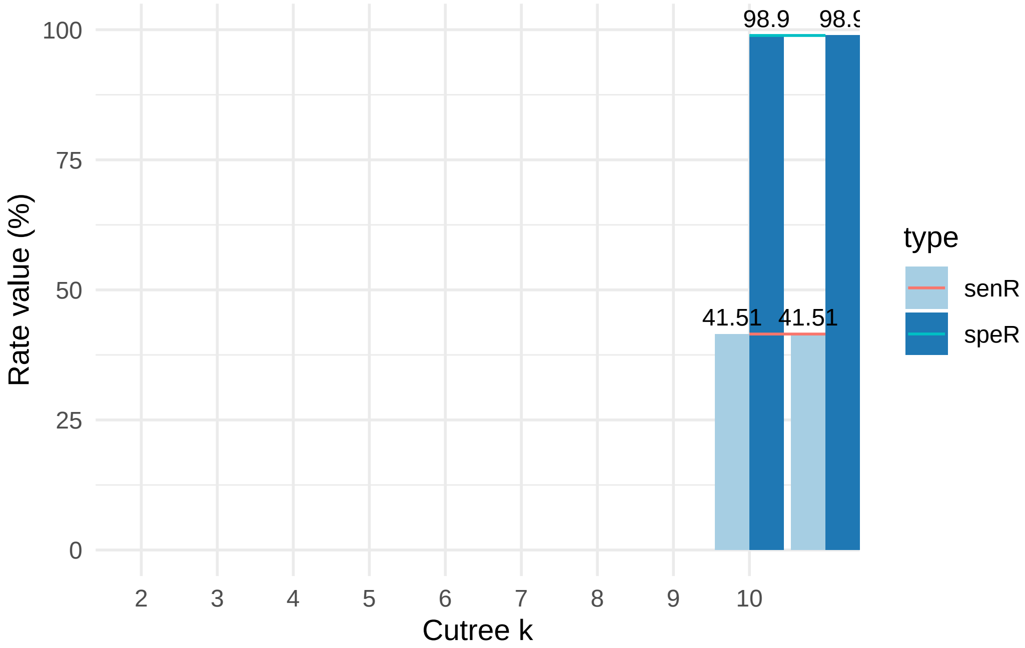


```
[1] "No acceptable clusters for M/B separation under single-based clustering analysis"
[1] "No acceptable clusters for M/B separation under average-based clustering analysis"
[1] "Clustering Method: ward.D2"
```

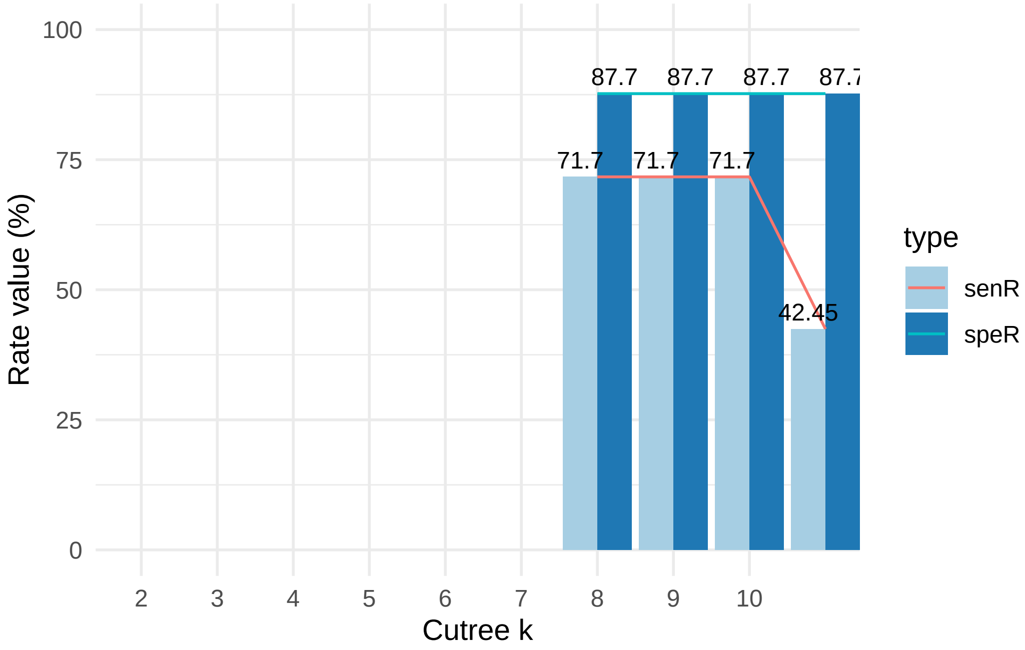


```
wisc.pr %>%
  hclustPCs_evalAna(method = c("complete", "single", "average", "ward.D2"), evalMethod = "
```

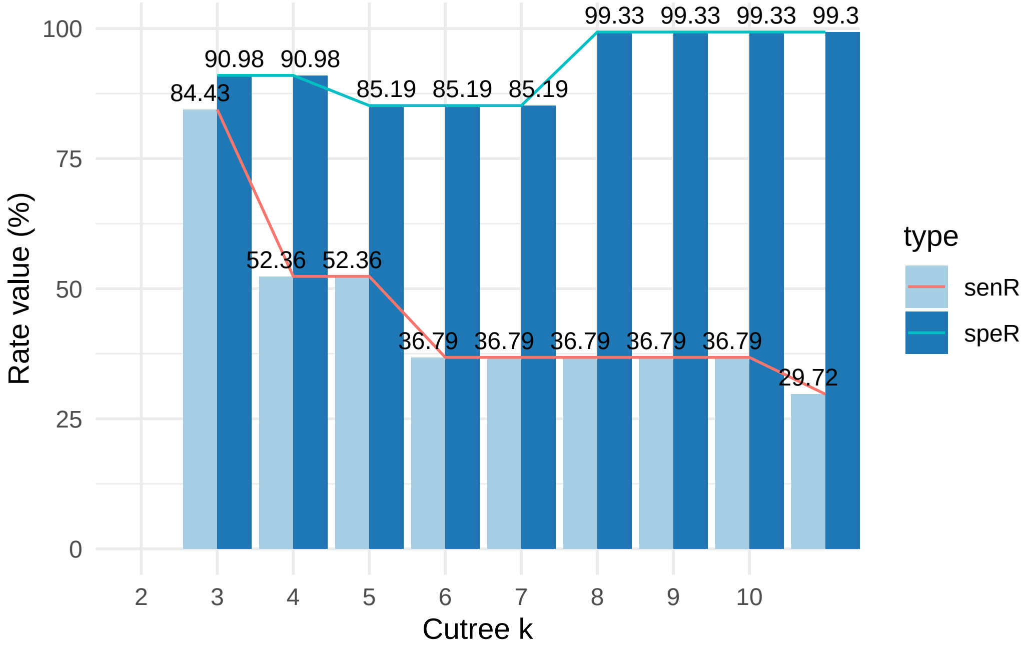
```
[1] "PCs: 3 Clustering method: complete"

Warning: Removed 14 rows containing missing values (geom_bar).

Warning: Removed 14 row(s) containing missing values (geom_path).

Warning: Removed 14 rows containing missing values (geom_text).
```



```
[1] "No acceptable clusters for M/B separation under Method: single for 3 PCs"
[1] "PCs: 3 Clustering method: average"

Warning: Removed 10 rows containing missing values (geom_bar).

Warning: Removed 10 row(s) containing missing values (geom_path).

Warning: Removed 10 rows containing missing values (geom_text).
```

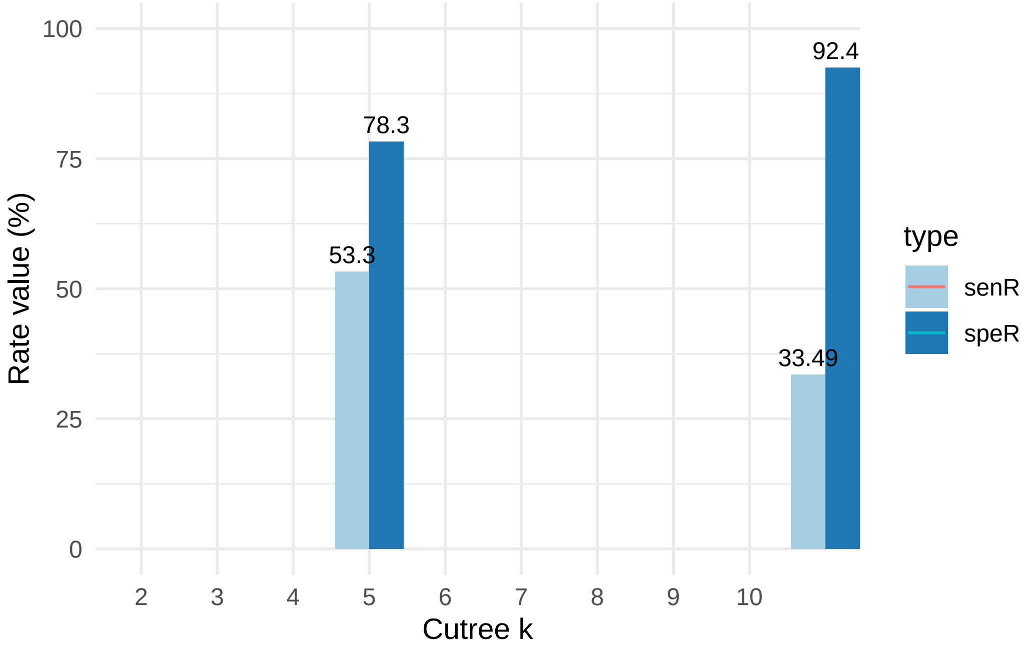


```
[1] "PCs: 3 Clustering method: ward.D2"
```

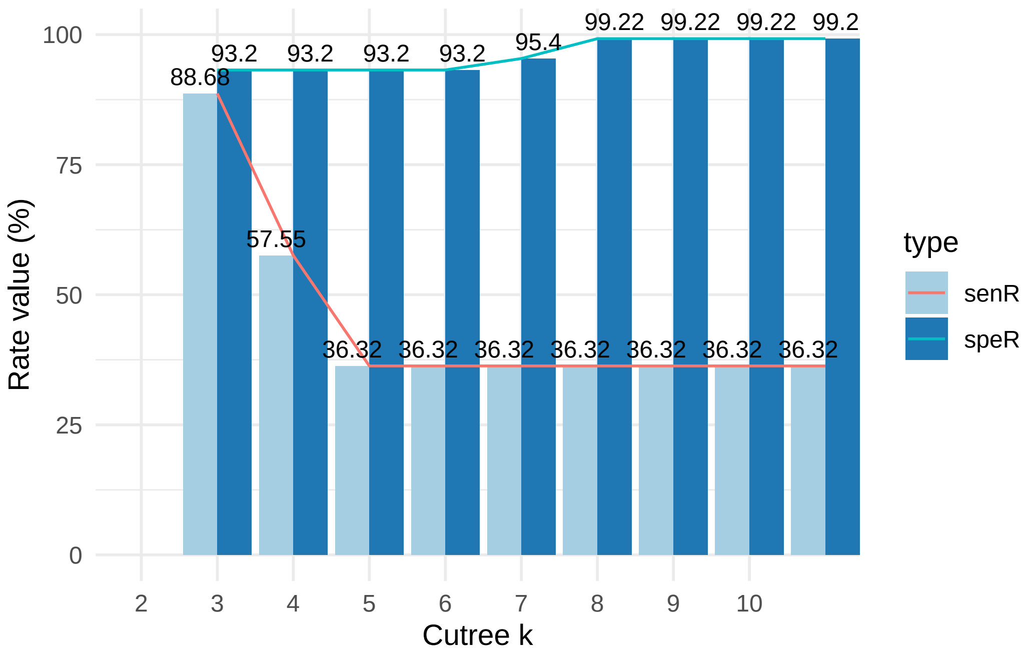


```
[1] "PCs: 7 Clustering method: complete"
```

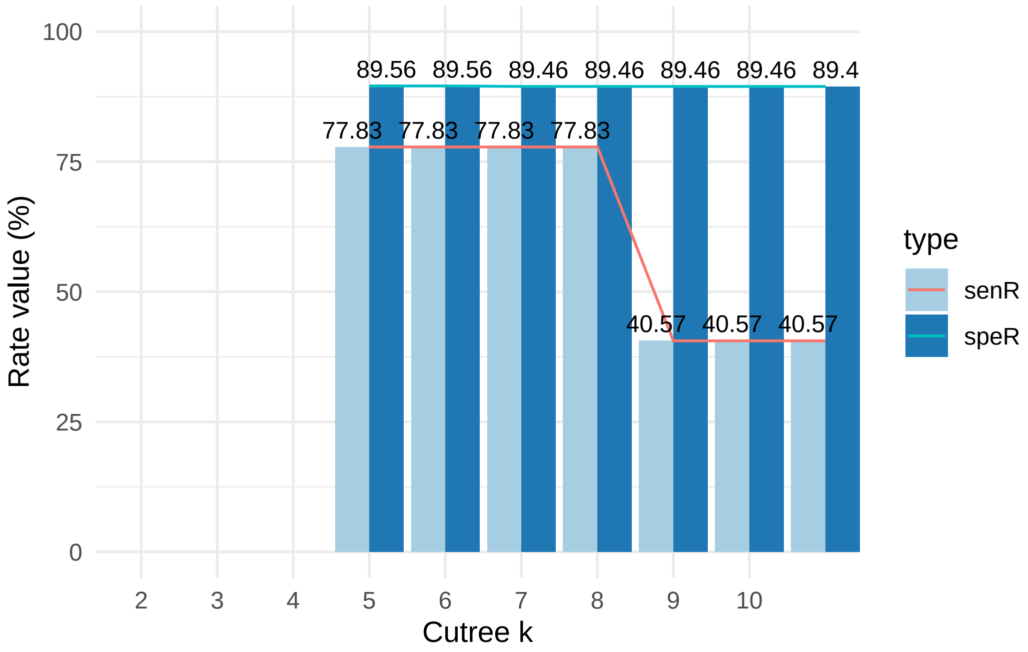
```
Warning: Removed 14 rows containing missing values (geom_bar).

Warning: Removed 4 row(s) containing missing values (geom_path).

Warning: Removed 14 rows containing missing values (geom_text).
```



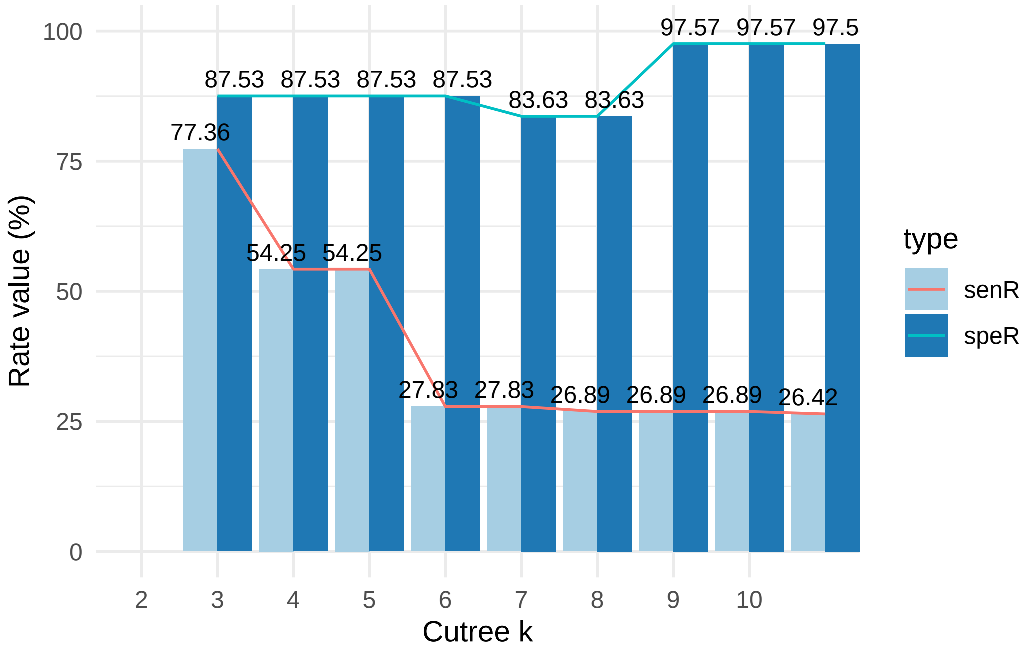
```
[1] "No acceptable clusters for M/B separation under Method: single for 7 PCs"
[1] "No acceptable clusters for M/B separation under Method: average for 7 PCs"
[1] "PCs: 7 Clustering method: ward.D2"
```



```
[1] "PCs: 30 Clustering method: complete"

Warning: Removed 4 rows containing missing values (geom_bar).

Warning: Removed 4 row(s) containing missing values (geom_path).

Warning: Removed 4 rows containing missing values (geom_text).
```



```
[1] "No acceptable clusters for M/B separation under Method: single for 30 PCs"
[1] "No acceptable clusters for M/B separation under Method: average for 30 PCs"
[1] "PCs: 30 Clustering method: ward.D2"
```



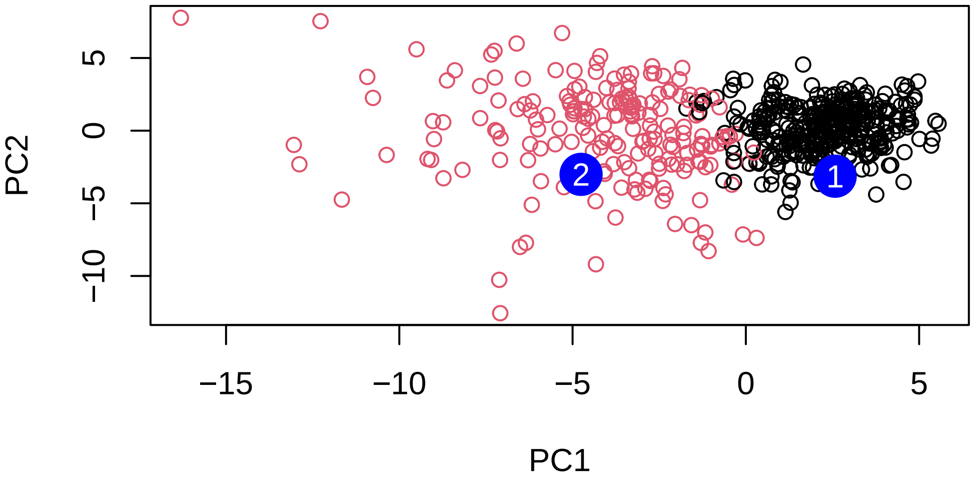
Prediction

```
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
npc
```

```
           PC1       PC2        PC3        PC4       PC5        PC6        PC7
[1,]  2.576616 -3.135913  1.3990492 -0.7631950  2.781648 -0.8150185 -0.3959098
[2,] -4.754928 -3.009033 -0.1660946 -0.6052952 -1.140698 -1.2189945  0.8193031
            PC8       PC9       PC10      PC11      PC12      PC13     PC14
[1,] -0.2307350 0.1029569 -0.9272861 0.3411457  0.375921 0.1610764 1.187882
[2,] -0.3307423 0.5281896 -0.4855301 0.7173233 -1.185917 0.5893856 0.303029
          PC15       PC16        PC17        PC18        PC19       PC20
[1,] 0.3216974 -0.1743616 -0.07875393 -0.11207028 -0.08802955 -0.2495216
[2,] 0.1299153  0.1448061 -0.40509706  0.06565549  0.25591230 -0.4289500
           PC21       PC22       PC23       PC24        PC25         PC26
[1,]  0.1228233 0.09358453 0.08347651  0.1223396  0.02124121  0.078884581
[2,] -0.1224776 0.01732146 0.06316631 -0.2338618 -0.20755948 -0.009833238
             PC27        PC28         PC29         PC30
[1,]  0.220199544 -0.02946023 -0.015620933  0.005269029
[2,] -0.001134152  0.09638361  0.002795349 -0.019015820
```

Q16: Based on the graph, patient 1 should be prioritized for follow up.

```
plot(wisc.pr$x[,1:2], col = g)
points(npc[,1], npc[,2], col = "blue", pch = 16, cex = 3)
text(npc[,1], npc[,2], c(1,2), col = "white")
```


5
0
−5
−10
PC2
−15
−10
−5
0
5
PC1
1
2