class09: Machine Learning mini project

Jiachen Fan (A17662703)

Exploratory data analysis

```r
# Save your input data file into your Project directory
fna.data <- "WisconsinCancer.csv"

# Complete the following code to input the data and store as wisc.df
wisc.df <-  read.csv(fna.data, row.names=1)
```

```r
# We can use -1 here to remove the first column
wisc.data <- wisc.df[,-1]
```

```r
# Create diagnosis vector for later
diagnosis <- as.factor(wisc.df$diagnosis)
```

> Q1. How many observations are in this dataset?

```r
nrow(wisc.data)
```

```
[1] 569
```

There are 569 observations.

> Q2. How many of the observations have a malignant diagnosis?

```r
table(diagnosis)
```

```
diagnosis
  B   M 
357 212 
```

212 have a malignant diagnosis.

> Q3. How many variables/features in the data are suffixed with _mean?

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

```
[1] 10
```

10 variables in '_mean'.

2. Principal Component Analysis

```
# Check column means and standard deviations
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 
```

```
# Perform PCA on wisc.data by completing the following code
wisc.pr <- prcomp(wisc.data, scale = TRUE)
```

```
# Look at summary of results
v <- summary(wisc.pr)
pcvar <- v$importance[3,]
pcvar
```

```
    PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10 
0.44272 0.63243 0.72636 0.79239 0.84734 0.88759 0.91010 0.92598 0.93988 0.95157 
   PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20 
0.96137 0.97007 0.97812 0.98335 0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 
   PC21    PC22    PC23    PC24    PC25    PC26    PC27    PC28    PC29    PC30 
0.99657 0.99749 0.99830 0.99890 0.99942 0.99969 0.99992 0.99997 1.00000 1.00000 
```

> Q4. From your results, what proportion of the original variance is captured by the first principal components (PC1)?

44.27% of the original variance is captured by PC1.

> Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data?

```
which(pcvar >= 0.7)[1]
```

```
PC3
  3
```

Three principal components are required, and they are PC1, PC2, PC3.

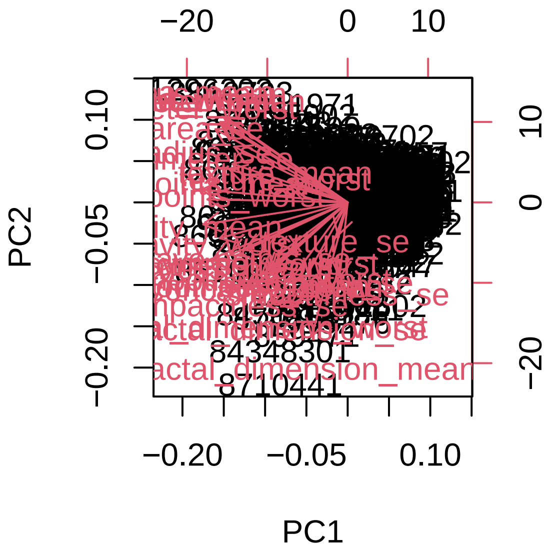
> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

```
which(pcvar >= 0.9)[1]
```

```
PC7
  7
```

Seven principal components.

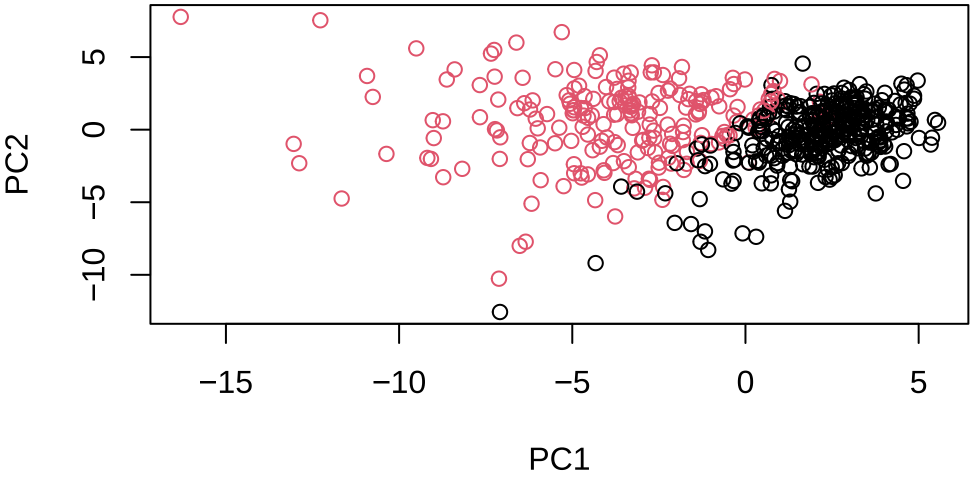
```
biplot(wisc.pr)
```



> Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

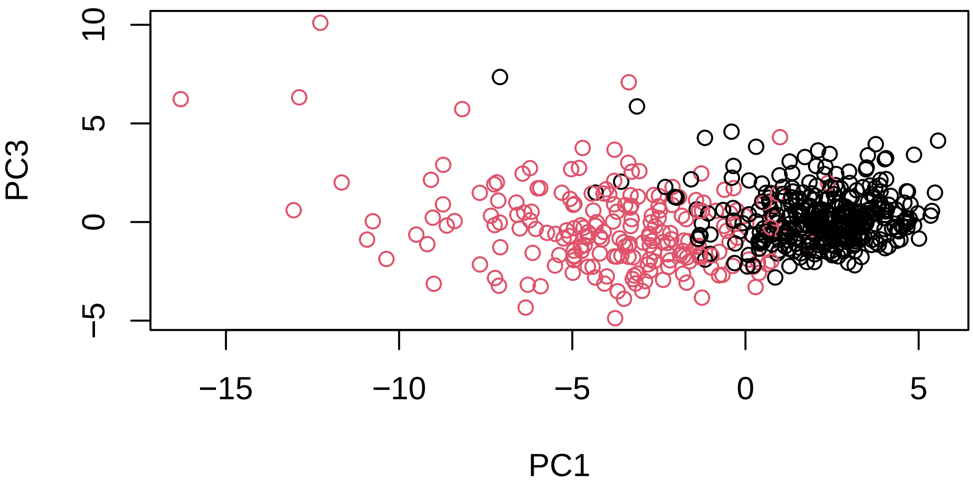
It is hard to see the contents because rownames are printed as well. It is difficult to understand.

```
# Scatter plot observations by components 1 and 2
plot( wisc.pr$x, col = diagnosis ,
     xlab = "PC1", ylab = "PC2")
```



Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

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

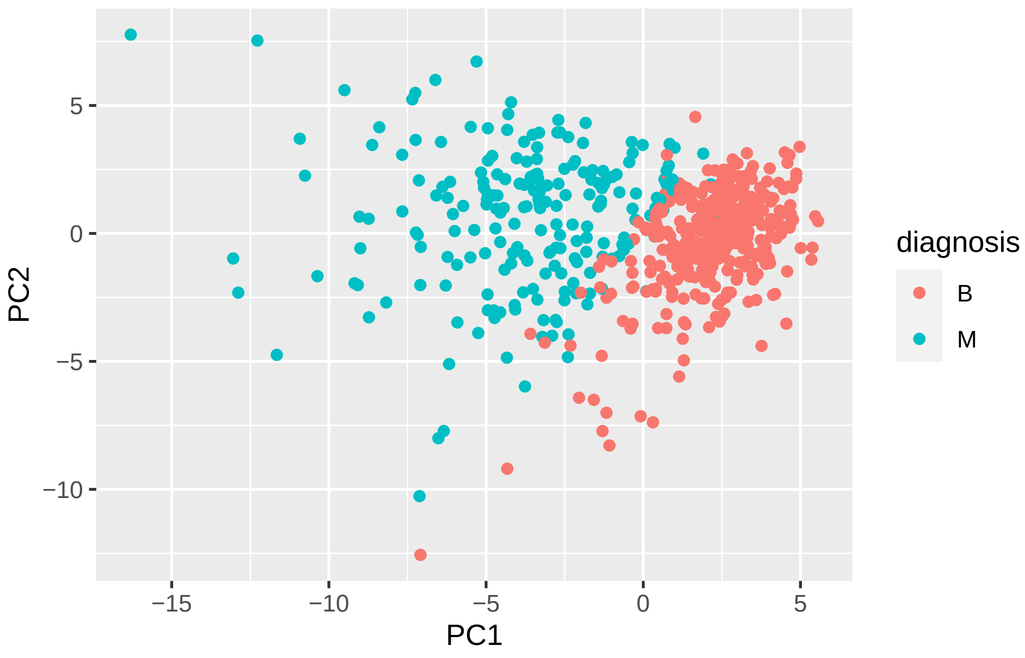


Try to use ‘ggplot2’

```
library(ggplot2)

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

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

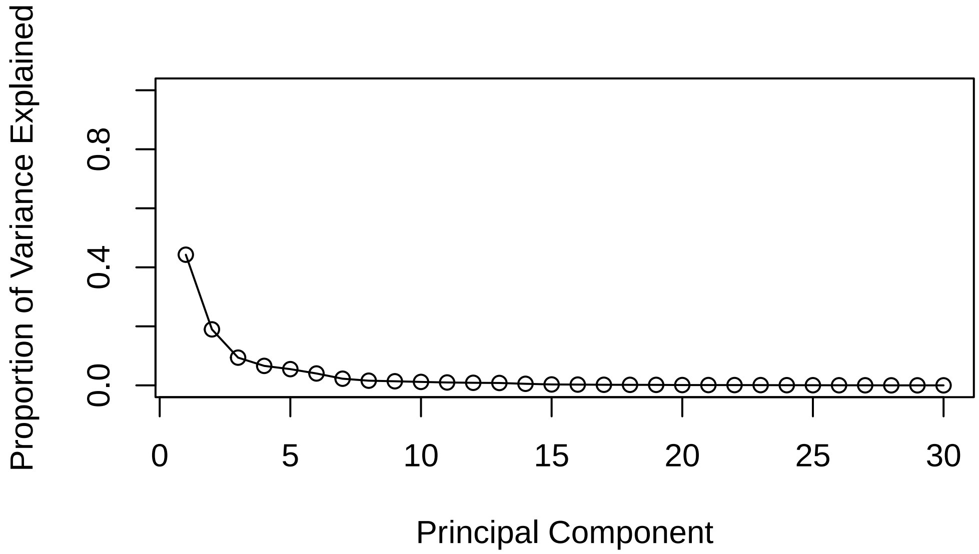


```r
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

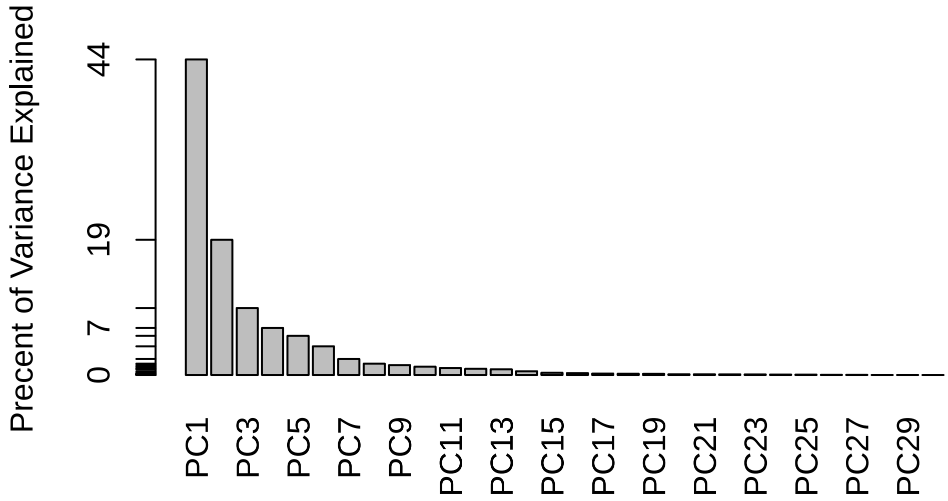
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```r
# Variance explained by each principal component: pve
pve <- pr.var / sum(pr.var)

# Plot variance explained for each principal component
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "o")
```



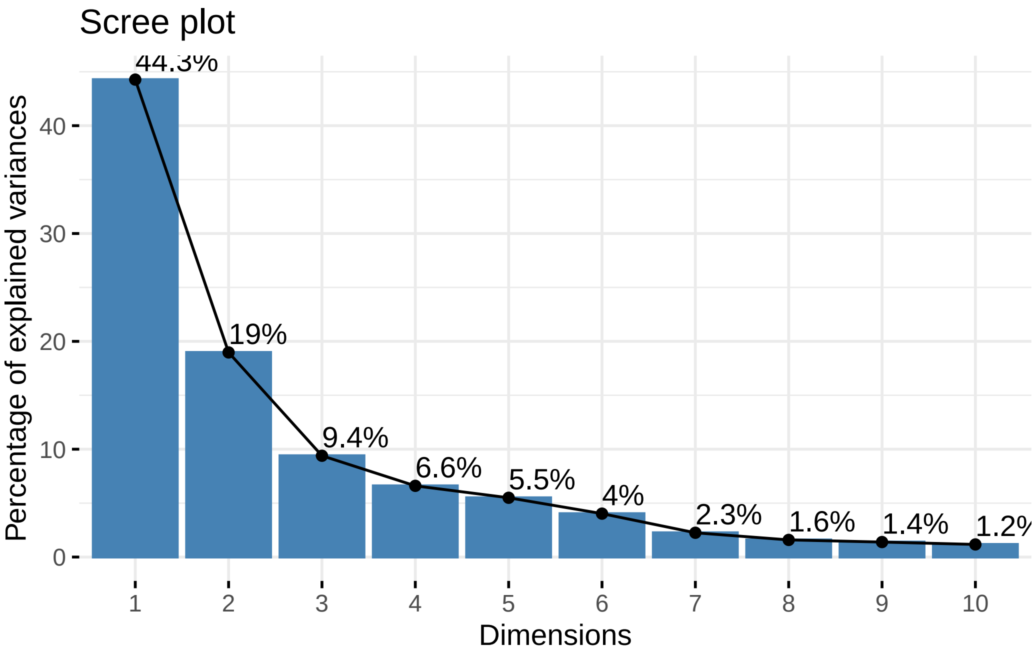
```
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```



```
## ggplot based graph
#install.packages("factoextra")
library(factoextra)
```

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

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



Q9. For the first principal component, what is the component of the loading vector (i.e. wisc.pr$rotation[,1]) for the feature concave.points_mean? This tells us how much this original feature contributes to the first PC.

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

```
concave.points_mean
         -0.2608538
```

This tells us 26.0853% of PC1 are contributed by this original feature.

3. Hierarchical clustering

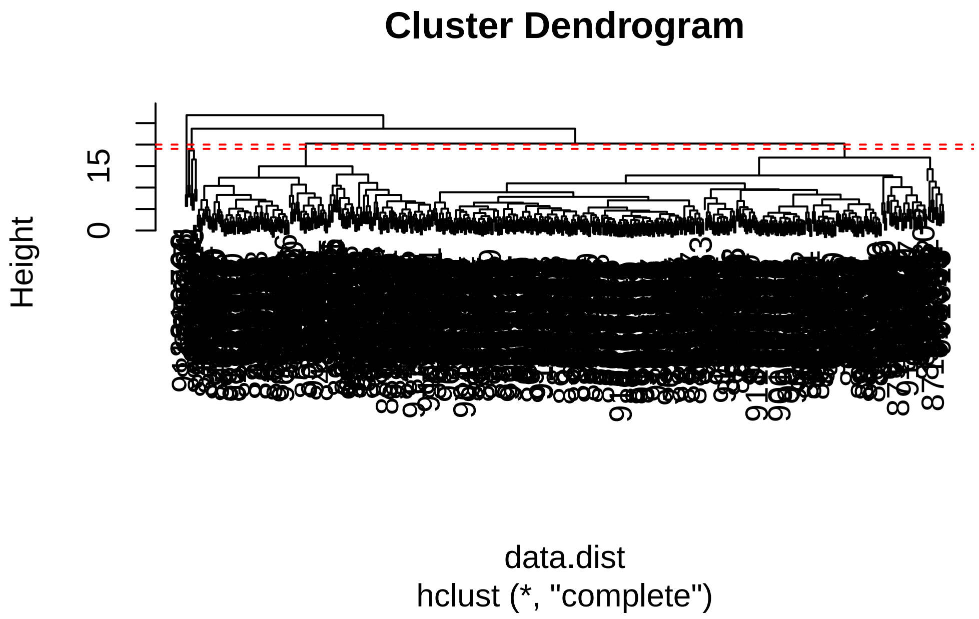
```
# Scale the wisc.data data using the "scale()" function
data.scaled <- scale(wisc.data)
```

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

```
wisc.hclust <- hclust(data.dist, method='complete')
```

Q10. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?

```
plot(wisc.hclust)
abline(h=c(19,20), col="red", lty=2)
```



The clustering model has 4 clusters at the heights of 19 and 20.

```
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. OPTIONAL: Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10? How do you judge the quality of your result in each case?

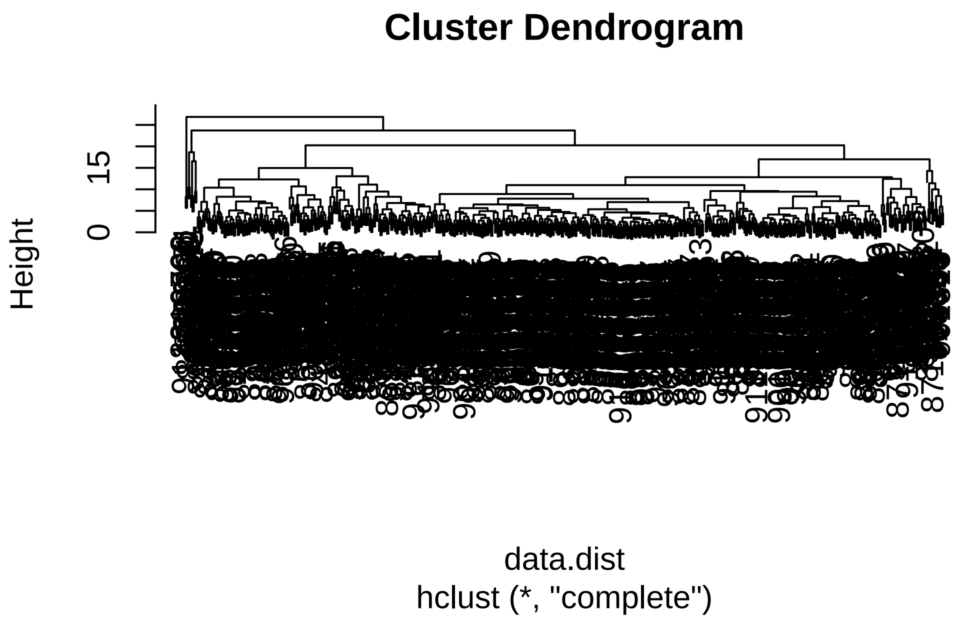
```
for (i in 2:10){
  clusters <- cutree(wisc.hclust, k=i)
  print(table(clusters,diagnosis))
}
```

```
        diagnosis
clusters   B   M
       1 357 210
       2   0   2
        diagnosis
clusters   B   M
       1 355 205
       2   2   5
       3   0   2
        diagnosis
clusters   B   M
       1  12 165
       2   2   5
       3 343  40
       4   0   2
        diagnosis
clusters   B   M
       1  12 165
       2   0   5
       3 343  40
       4   2   0
       5   0   2
        diagnosis
clusters   B   M
       1  12 165
       2   0   5
       3 331  39
       4   2   0
       5  12   1
       6   0   2
        diagnosis
clusters   B   M
       1  12 165
       2   0   3
       3 331  39
       4   2   0
       5  12   1
       6   0   2
       7   0   2
        diagnosis
clusters   B   M
       1  12  86
       2   0  79
       3   0   3
       4 331  39
       5   2   0
       6  12   1
       7   0   2
       8   0   2
        diagnosis
clusters   B   M
       1  12  86
       2   0  79
       3   0   3
       4 331  39
       5   2   0
       6  12   0
       7   0   2
       8   0   2
       9   0   1
        diagnosis
clusters   B   M
      1   12  86
      2    0  59
      3    0   3
      4  331  39
      5    0  20
      6    2   0
      7   12   0
      8    0   2
      9    0   2
      10   0   1
```

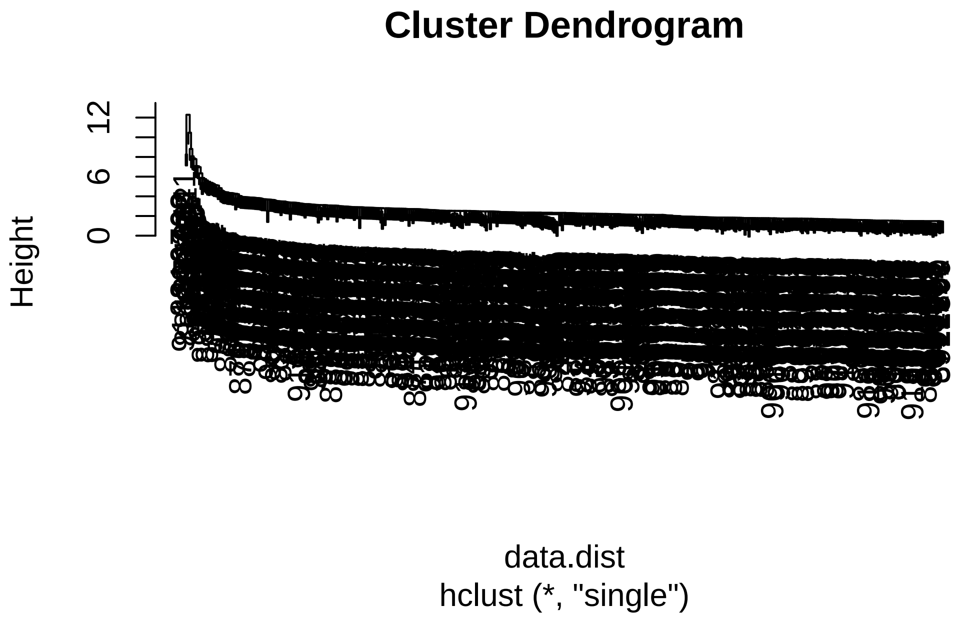
4 clusters is the best.

Q12. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

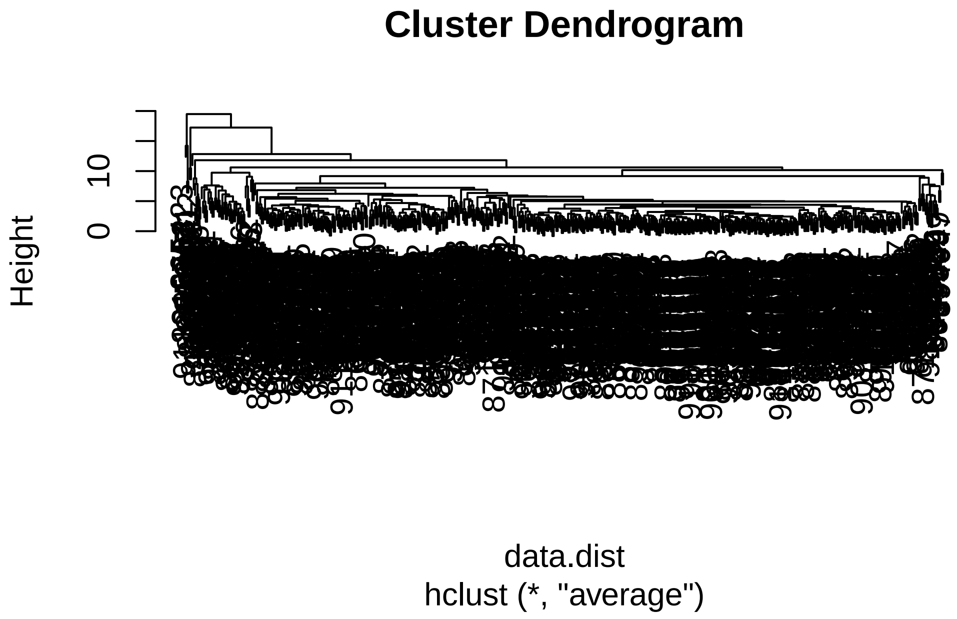
```
wisc.hclust.complete <- hclust(data.dist, method="complete")
plot(wisc.hclust.complete)
```



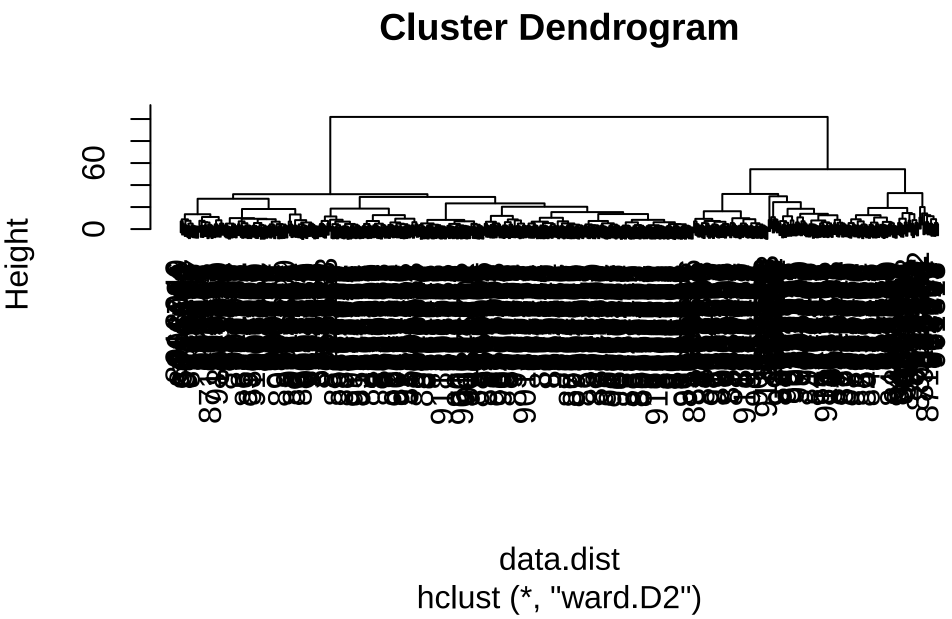
```
wisc.hclust.single <- hclust(data.dist, method="single")
plot(wisc.hclust.single)
```



```
wisc.hclust.average <- hclust(data.dist, method="average")
plot(wisc.hclust.average)
```



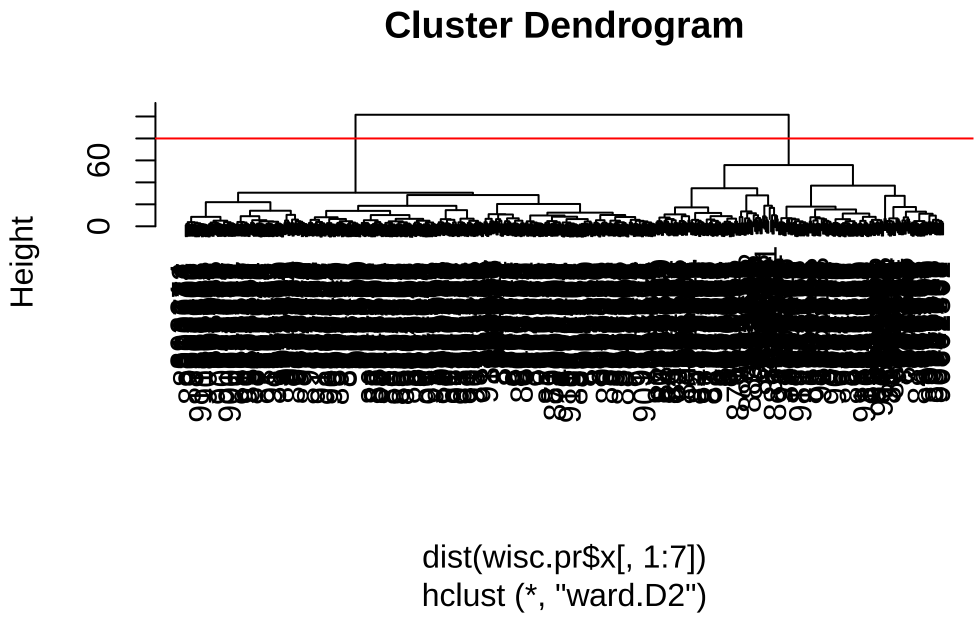
```
wisc.hclust.ward.D2 <- hclust(data.dist, method="ward.D2")
plot(wisc.hclust.ward.D2)
```



I think "ward.D2" method gives me best results. I can get a clear clustering result vs diagnoses match only with 2 clusters.

4. Combining methods

```
wisc.pr.hclust <- hclust(dist(wisc.pr$x[, 1:7]), method="ward.D2")
plot(wisc.pr.hclust)
abline(h=80,col='red')
```



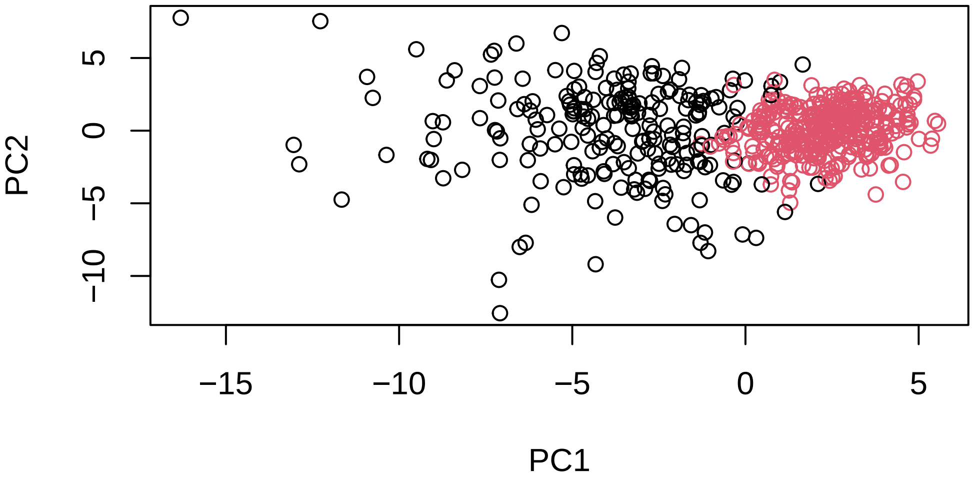
```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
  1   2 
216 353 
```

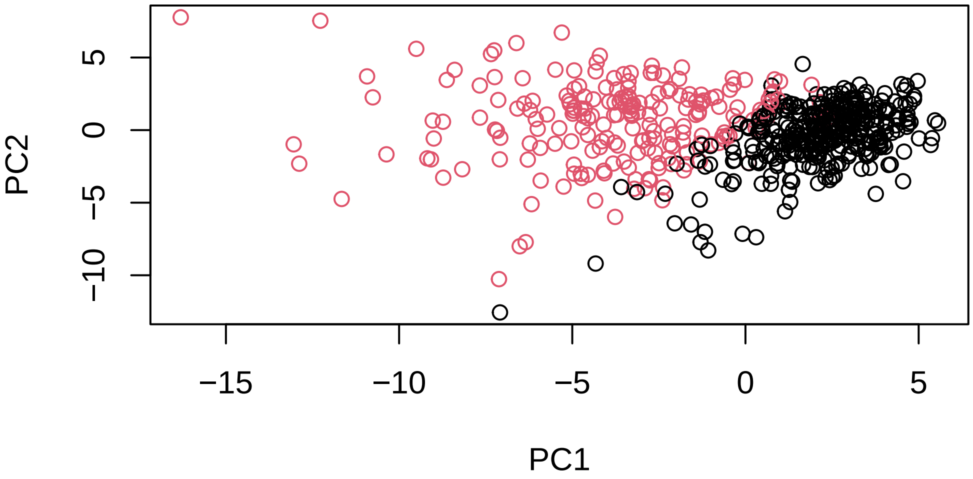
```
table(grps, diagnosis)
```

```
    diagnosis
grps   B   M
   1  28 188
   2 329  24
```

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



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



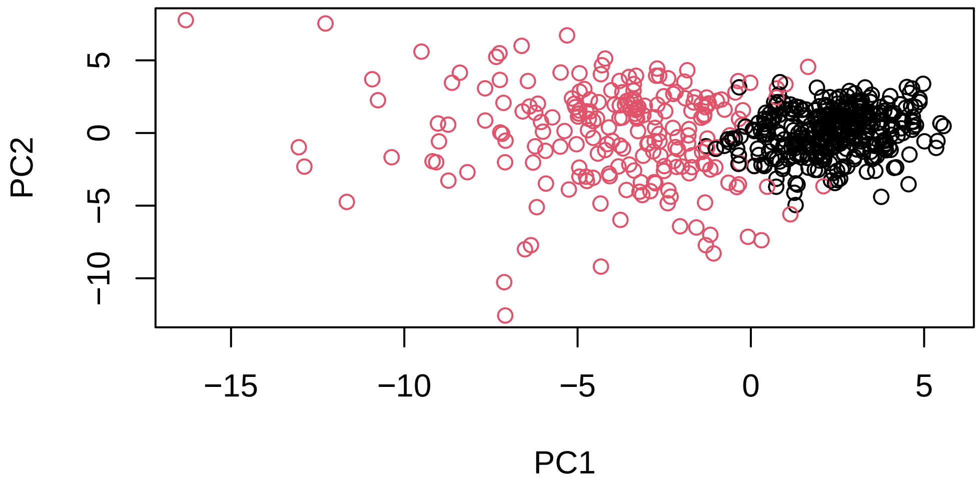
```
g <- as.factor(grps)
levels(g)
```

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

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

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

```
# Plot using our re-ordered factor
plot(wisc.pr$x[,1:2], col=g)
```



```
library(rgl)
plot3d(wisc.pr$x[,1:3], xlab="PC 1", ylab="PC 2", zlab="PC 3", cex=1.5, size=1, type="s",
```

```
## 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")
```

> Q13. How well does the newly created model with four clusters separate out the two diagnoses?

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

```
    diagnosis
grps   B   M
   1  28 188
   2 329  24
```

> Q14. How well do the hierarchical clustering models you created in previous sections (i.e. before PCA) do in terms of separating the diagnoses? Again, use the table() function to compare the output of each model (wisc.km$cluster and wisc.hclust.clusters) with the vector containing the actual diagnoses.

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

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

PCA model is better because we only need two clusters compared to 4 factors.

5. Sensitivity/Specificity

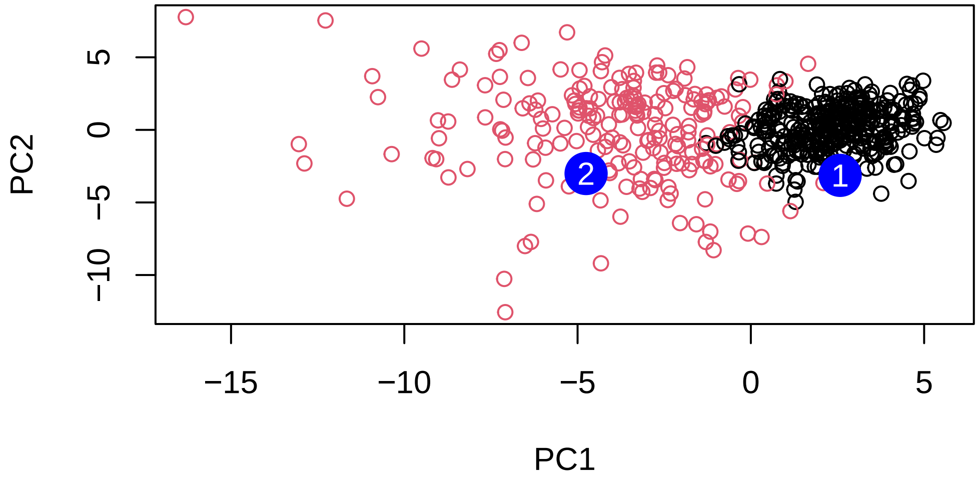
> Q15. OPTIONAL: Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

6. Prediction

```
url <- "new_samples.csv"
#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
```

```
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")
```



Q16. Which of these new patients should we prioritize for follow up based on your results?

The patients in the group 2 should be priortized for follow up.