Class08_Mini-Project

Changcheng Li (PID: A69027828)

```
# 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("WisconsinCancer.csv", row.names=1)
head(wisc.df)
```

```
         diagnosis radius_mean texture_mean perimeter_mean area_mean
842302           M       17.99        10.38         122.80    1001.0
842517           M       20.57        17.77         132.90    1326.0
84300903         M       19.69        21.25         130.00    1203.0
84348301         M       11.42        20.38          77.58     386.1
84358402         M       20.29        14.34         135.10    1297.0
843786           M       12.45        15.70          82.57     477.1
         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
84300903         0.10960          0.15990         0.1974             0.12790
84348301         0.14250          0.28390         0.2414             0.10520
84358402         0.10030          0.13280         0.1980             0.10430
843786           0.12780          0.17000         0.1578             0.08089
         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
84300903        0.2069                0.05999    0.7456     0.7869        4.585
84348301        0.2597                0.09744    0.4956     1.1560        3.445
84358402        0.1809                0.05883    0.7572     0.7813        5.438
843786          0.2087                0.07613    0.3345     0.8902        2.217
         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
84300903   94.03      0.006150        0.04006      0.03832           0.02058
84348301   27.23      0.009110        0.07458      0.05661           0.01867
84358402   94.44      0.011490        0.02461      0.05688           0.01885
843786     27.19      0.007510        0.03345      0.03672           0.01137
         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
84300903     0.02250             0.004571        23.57         25.53
84348301     0.05963             0.009208        14.91         26.50
84358402     0.01756             0.005115        22.54         16.67
843786       0.02165             0.005082        15.47         23.75
         perimeter_worst area_worst smoothness_worst compactness_worst
842302            184.60     2019.0           0.1622            0.6656
842517            158.80     1956.0           0.1238            0.1866
84300903          152.50     1709.0           0.1444            0.4245
84348301           98.87      567.7           0.2098            0.8663
84358402          152.20     1575.0           0.1374            0.2050
843786            103.40      741.6           0.1791            0.5249
         concavity_worst concave.points_worst symmetry_worst
842302            0.7119               0.2654         0.4601
842517            0.2416               0.1860         0.2750
84300903          0.4504               0.2430         0.3613
84348301          0.6869               0.2575         0.6638
84358402          0.4000               0.1625         0.2364
843786            0.5355               0.1741         0.3985
         fractal_dimension_worst  X
842302                   0.11890 NA
842517                   0.08902 NA
84300903                 0.08758 NA
84348301                 0.17300 NA
84358402                 0.07678 NA
843786                   0.12440 NA
```

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

```
# Create diagnosis vector for later
diagnosis <- wisc.df[,1]
```

Q1. How many observations are in this dataset?

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

```
[1] 569
```

There are 569 observations.

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

```
table(diagnosis)
```

```
diagnosis
  B   M
357 212
```

There are 212 malignant diagnosis.

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

```
grep("_mean", colnames(wisc.data))
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

There are 10 variables suffixed with _mean.

```
# 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
                    X
                   NA
```

```
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
                    X
                   NA
```

```
# Perform PCA on wisc.data by completing the following code
wisc.pr <- prcomp(wisc.data[,-31], scale = TRUE)
```

```r
# Look at summary of results
v <- summary(wisc.pr)
pcvar <- v$importance[3,]
pcvar['PC1']
```

```
    PC1
0.44272
```

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

44.27%

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

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

```
PC3
  3
```

3 PCs are required (72.63%)

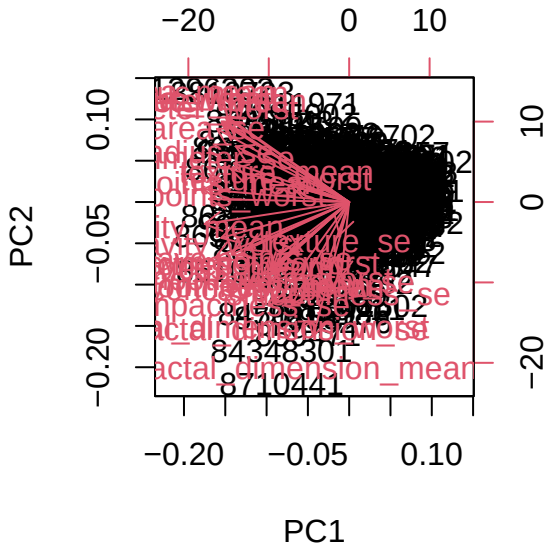
Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

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

```
PC7
  7
```

7 PCs are required (91.01)

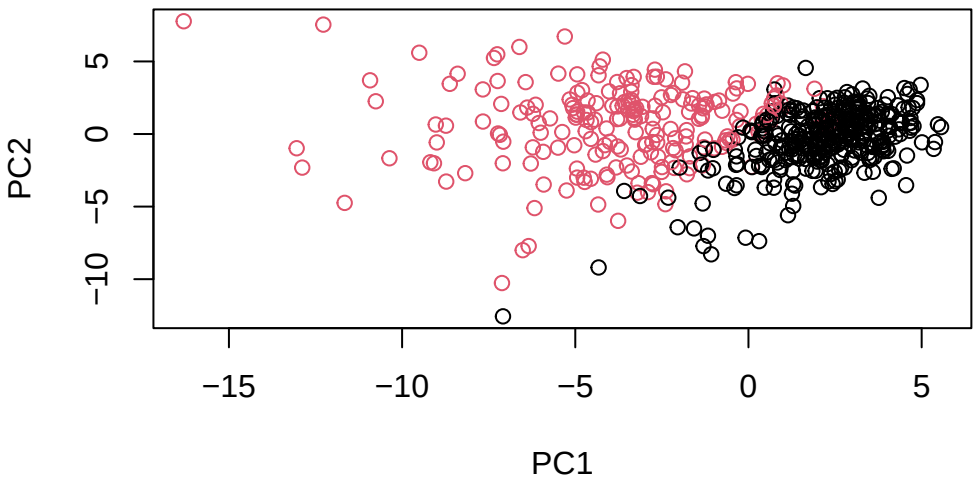
```r
biplot(wisc.pr)
```



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

Nothing stands out and it is difficult to understand. It is blurry because all factors related to diagnosis are shown together and it is difficult to distinguish their difference with regards to PC1 and PC2.

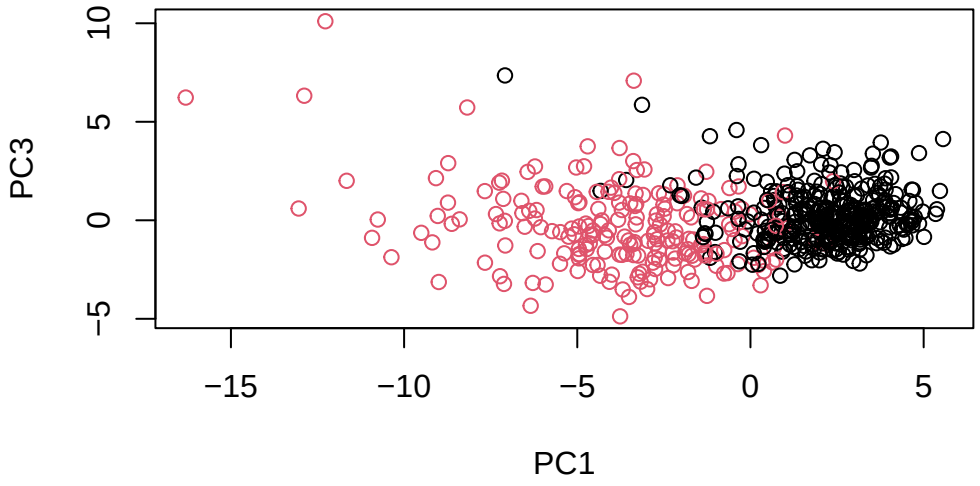
```
# Scatter plot observations by components 1 and 2
plot(wisc.pr$x[,1], wisc.pr$x[,2], col = as.factor(diagnosis),
     xlab = "PC1", ylab = "PC2")
```



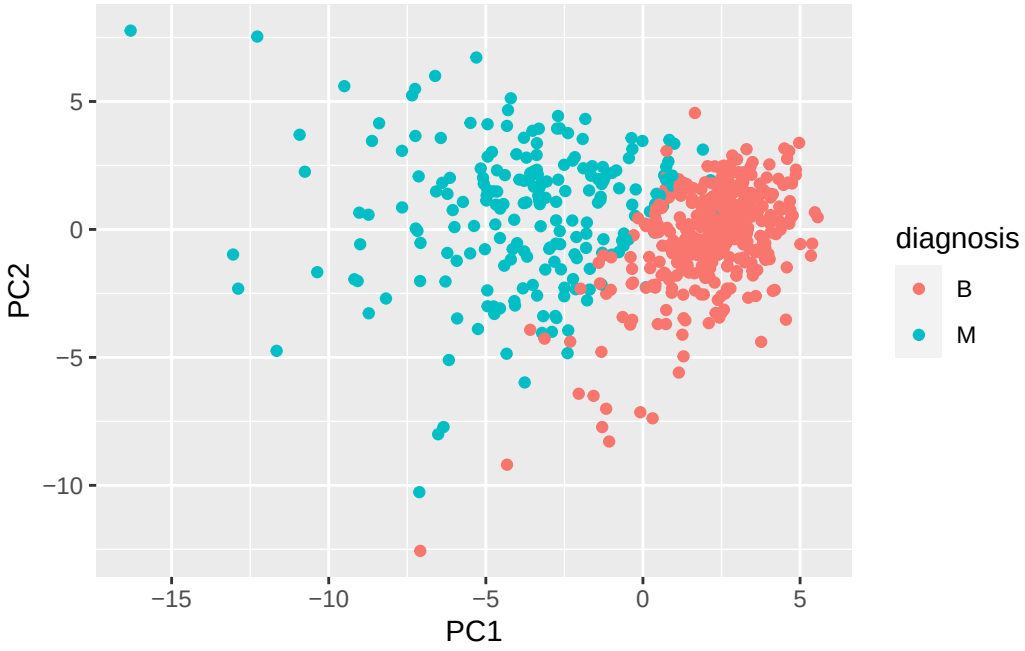
Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

Data with different diagnosis are predominantly distinguished by PC1 instead of PC2 and PC3.

```
# Repeat for components 1 and 3
plot(wisc.pr$x[,1], wisc.pr$x[,3], col = as.factor(diagnosis),
     xlab = "PC1", ylab = "PC3")
```



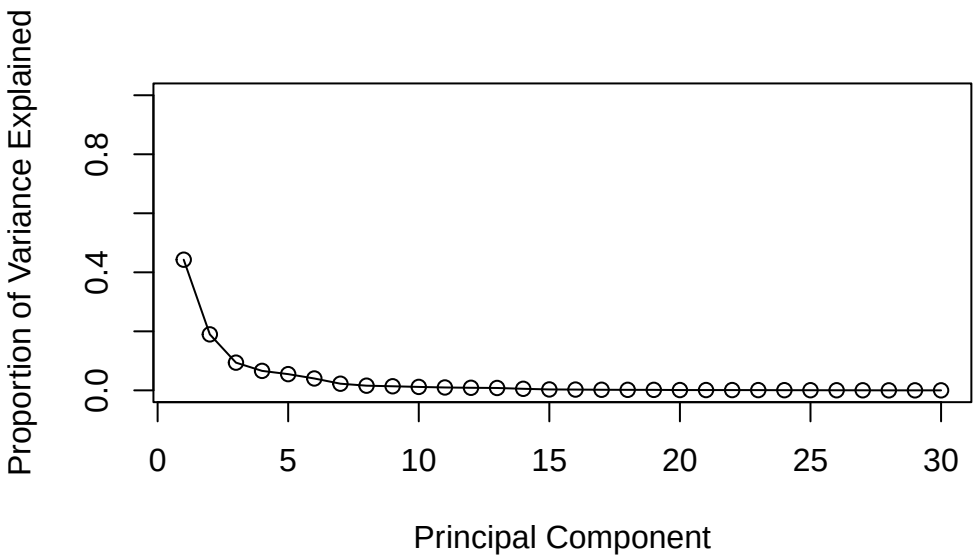

```
# Create a data.frame for ggplot
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis

# Load the ggplot2 package
library(ggplot2)

# Make a scatter plot colored by diagnosis
ggplot(df) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```



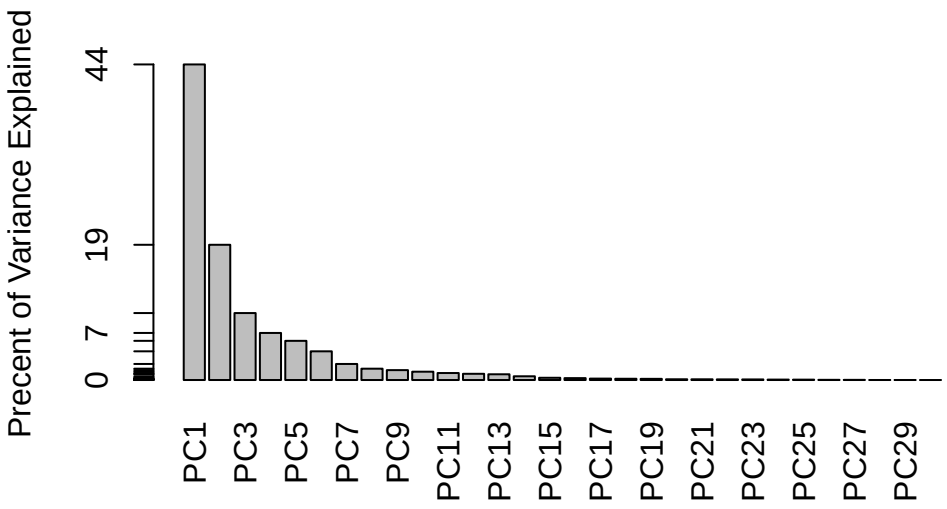

```
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

```
# 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")
```



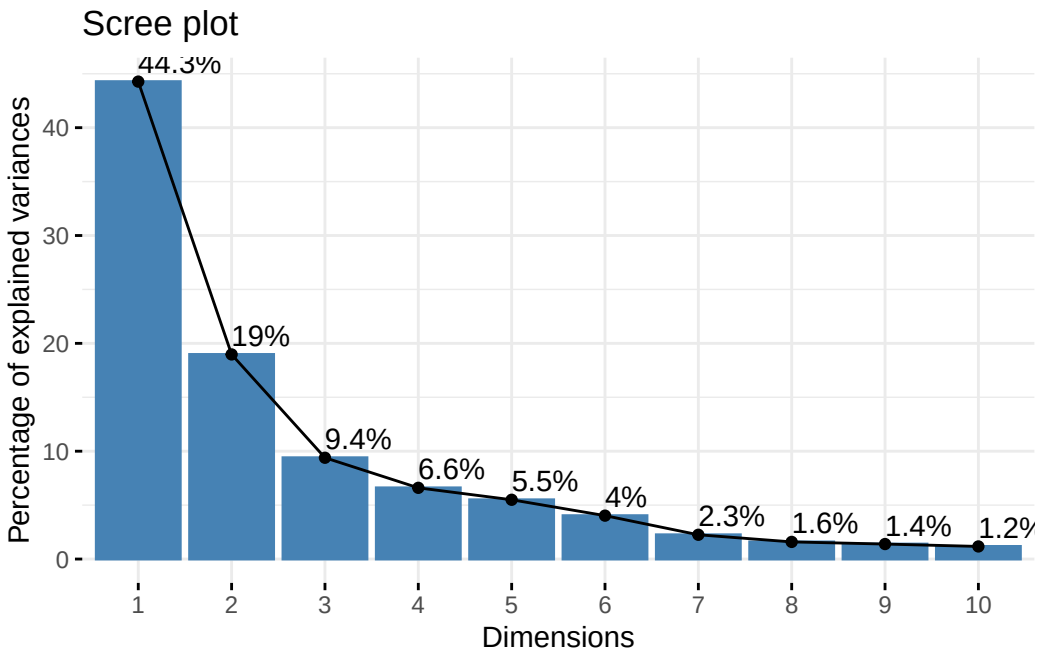

```
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```




```
## 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.

It is -0.2609

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

```
[1] -0.2608538
```

#3. Hierarchical clustering

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

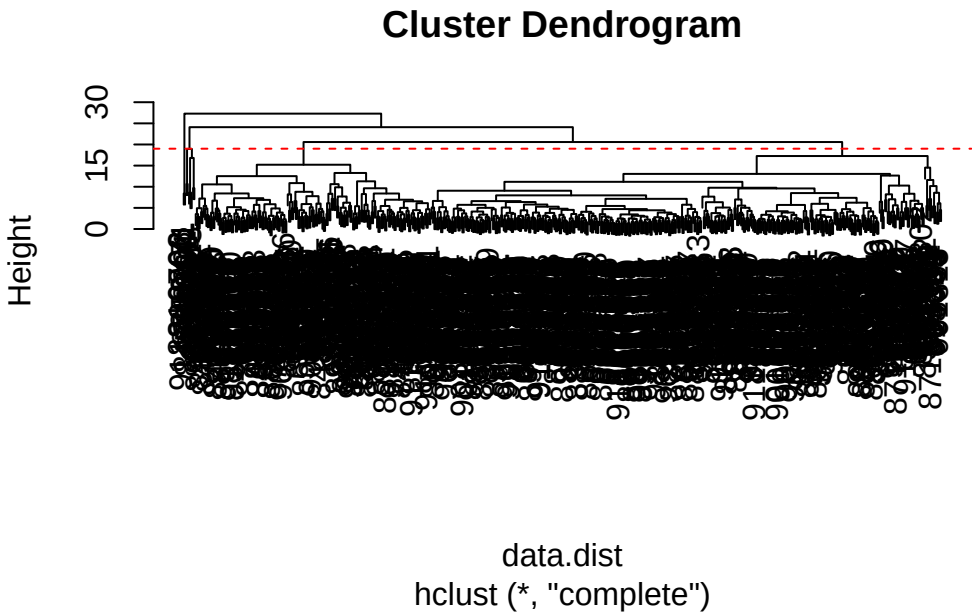
```
data.dist <- dist(data.scaled)
```

```
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?

The height is 19.

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




```
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?

Cutting results by 2 to 10 are shown below. I cannot find a better match other than 4 clusters: When cluster number is below 4 it does not separate malignant and benign at all. By

contrast, when cluster number is above 4, the further clustering does not increase the quality of malignant vs benign separation (The false positive rate does not decrease) so it is not better than 4 clusters.

```
df <- data.frame(c(2:10))
comp <- function(k){table(cutree(wisc.hclust, k), diagnosis)}
apply(df, 1, comp)
```

```
[[1]]
   diagnosis
      B   M
  1 357 210
  2   0   2

[[2]]
   diagnosis
      B   M
  1 355 205
  2   2   5
  3   0   2

[[3]]
   diagnosis
      B   M
  1  12 165
  2   2   5
  3 343  40
  4   0   2

[[4]]
   diagnosis
      B   M
  1  12 165
  2   0   5
  3 343  40
  4   2   0
  5   0   2

[[5]]
   diagnosis
      B   M
  1  12 165
```

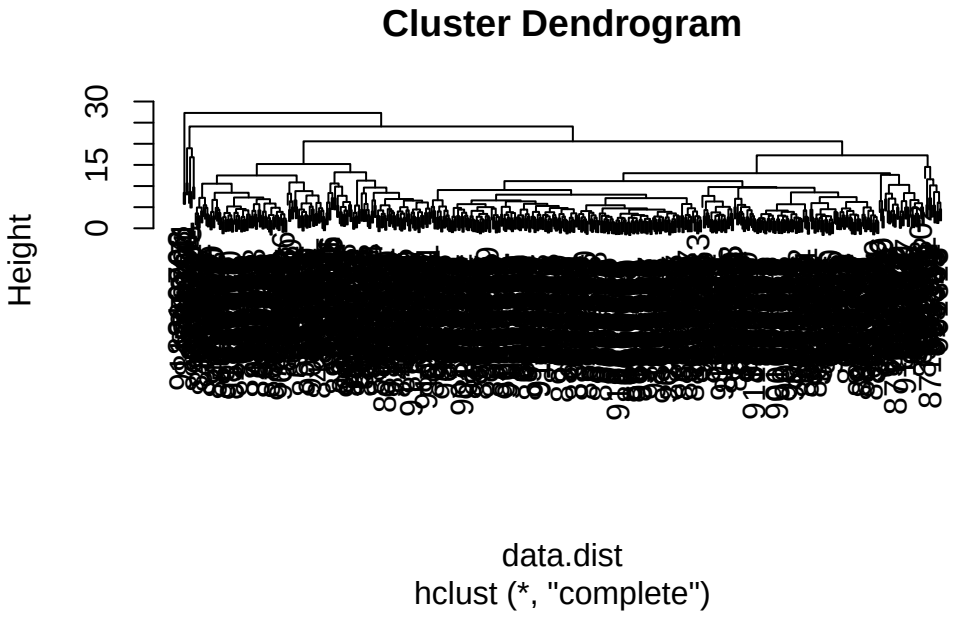
```
  2   0   5
  3 331  39
  4   2   0
  5  12   1
  6   0   2

[[6]]
   diagnosis
      B   M
  1  12 165
  2   0   3
  3 331  39
  4   2   0
  5  12   1
  6   0   2
  7   0   2

[[7]]
   diagnosis
      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

[[8]]
   diagnosis
      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

[[9]]
```

```
   diagnosis
      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
```

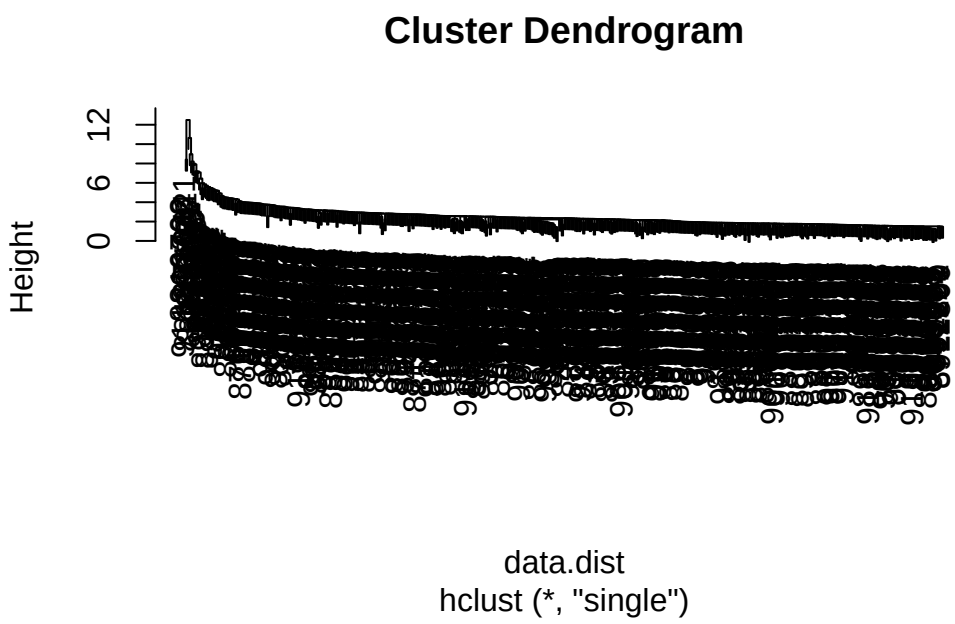
Q12. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

The method="ward.D2" is my favorite because it creates groups such that variance is minimized within clusters. Therefore, the two main branches of or dendrogram indicating two main clusters - malignant and benign.

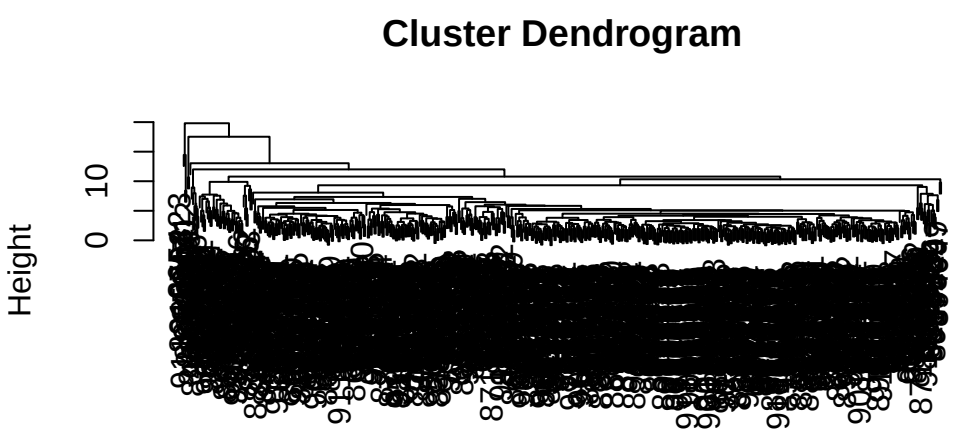
```
plot(hclust(data.dist, method = "complete"))
```



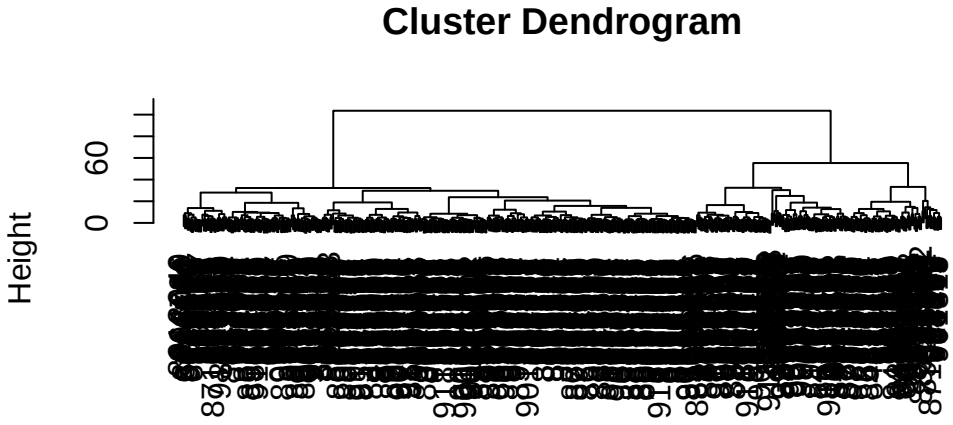
```
plot(hclust(data.dist, method = "single"))
```




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

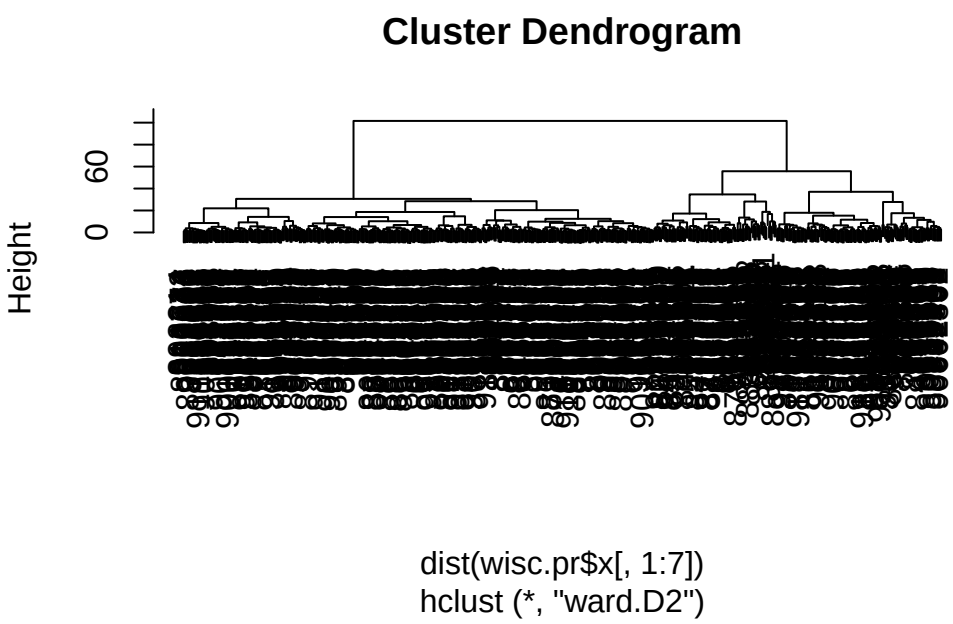



```
plot(hclust(data.dist, method = "ward.D2"))
```



#4. Combining methods

```
plot(hclust(dist(wisc.pr$x[,1:7]), method = "ward.D2"))
```



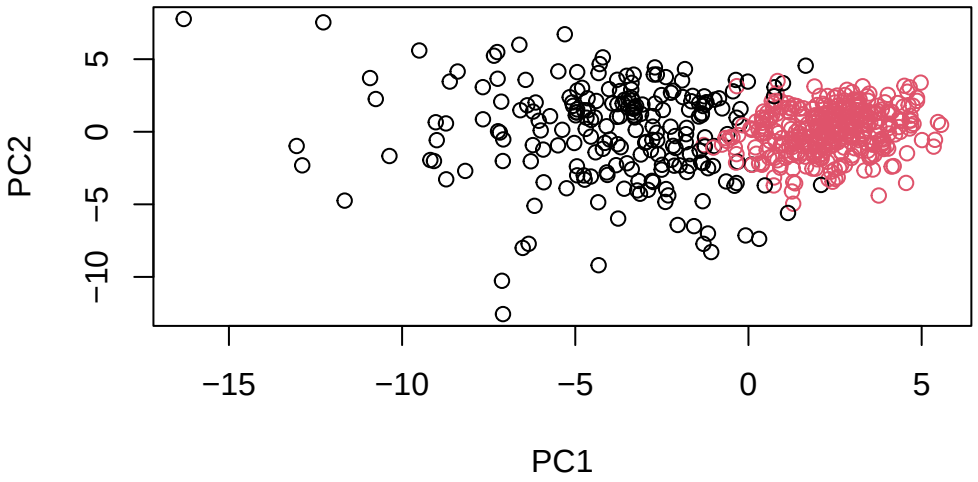

```
wisc.pr.hclust <- hclust(dist(wisc.pr$x[,1:7]), method = "ward.D2")
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
  1   2
216 353
```

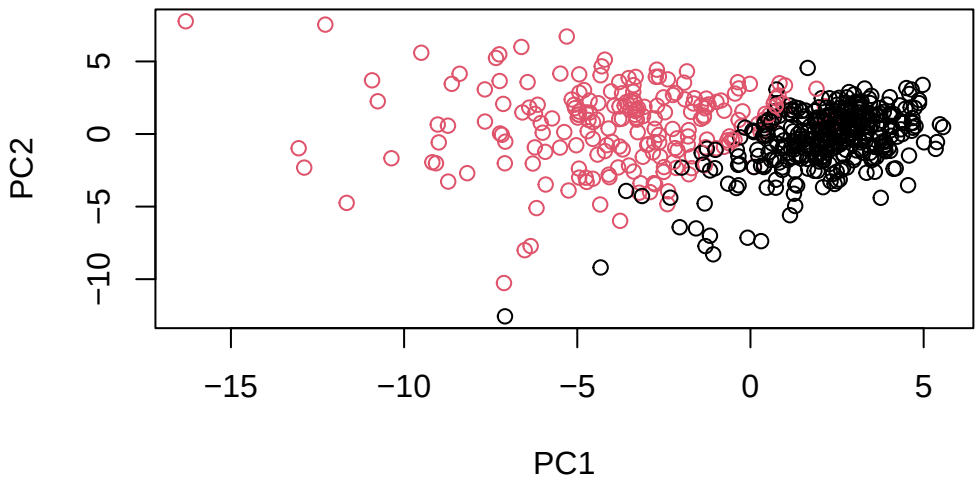
```
table(grps, diagnosis)
```

```
    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=as.factor(diagnosis))
```



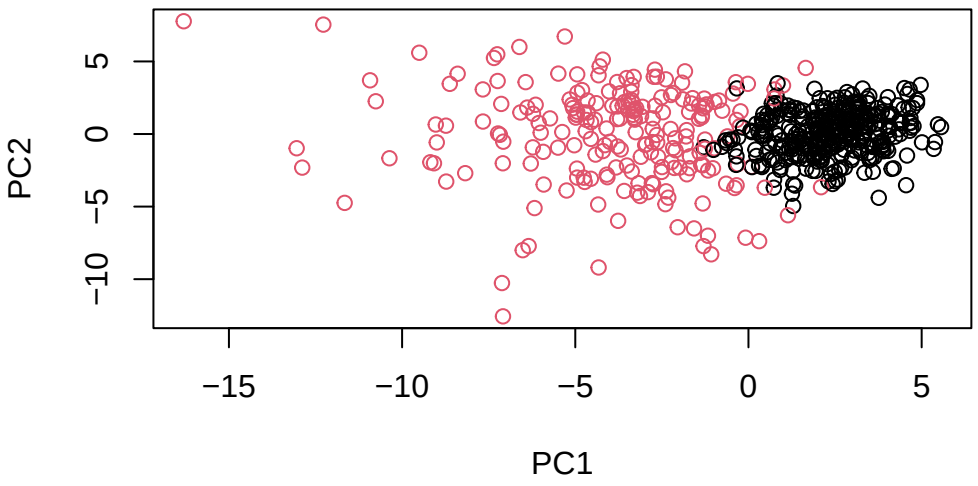

```
g <- as.factor(grps)
levels(g)
```

```
[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")
```

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

```
# Compare to actual diagnoses
table(wisc.pr.hclust.clusters, diagnosis)
```

```
                     diagnosis
wisc.pr.hclust.clusters   B   M
                      1  28 188
                      2 329  24
```

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

The result of separation by 4 clusters is shown below. It is not better than that of 2 clusters because it does not increase the specificity of separation (still 329B/24M in cluster 4 correspond-

ing to cluster 2 in the 2 clusters condition). But it indeed separate malignant group further into 3 sub-clusters and the first cluster is the malignant group with highest confidence.

```
table(cutree(wisc.pr.hclust, k=4), diagnosis)
```

```
   diagnosis
      B   M
  1   0  45
  2   2  77
  3  26  66
  4 329  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.

The performance of hierarchical clustering models created without PCA in separating the diagnosis is worse than that of using PCA because 1) In these models it requires at least 4 clusters to separate diagnosis, 2 clusters are not enough for separation at all (Sometimes even 4 clusters is not enough) and 2) The false positive rate of separation using these models is higher than that of with PCA.

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

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

```
table(cutree(hclust(data.dist, method = "single"), k = 4), diagnosis)
```

```
   diagnosis
      B   M
  1 356 209
  2   1   0
  3   0   2
  4   0   1
```

```
table(cutree(hclust(data.dist, method = "average"), k = 4), diagnosis)
```

```
  diagnosis
     B   M
1  355 209
2    2   0
3    0   1
4    0   2
```

```
table(cutree(hclust(data.dist, method = "ward.D2"), k = 4), diagnosis)
```

```
  diagnosis
     B   M
1    0 115
2    6  48
3  337  48
4   14   1
```

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

Specificity for the 2/4 clusters model with PCA: 329 / (329 + 24) = 93.2 % Specificity for the 4 clusters model without PCA(method = "complete"): 343 / (343 + 40 + 2) = 89.1 % Specificity for the 4 clusters model without PCA(method = "ward.D2"): (337 + 14) / (337 + 14 + 48 + 1) = 87.8 %

Sensitivity for the 2/4 clusters model with PCA: 188 / (188 + 24) = 88.7 % (45 + 77 + 66 / (45 + 77 + 66 + 24) = 88.7 %) Sensitivity for the 4 clusters model without PCA(method = "complete"): (165 + 5) / (165 + 5 + 40 + 2) = 80.2 % Sensitivity for the 4 clusters model without PCA(method = "ward.D2"): (115 + 48) / (115 + 48 + 48 + 1) = 76.9 %

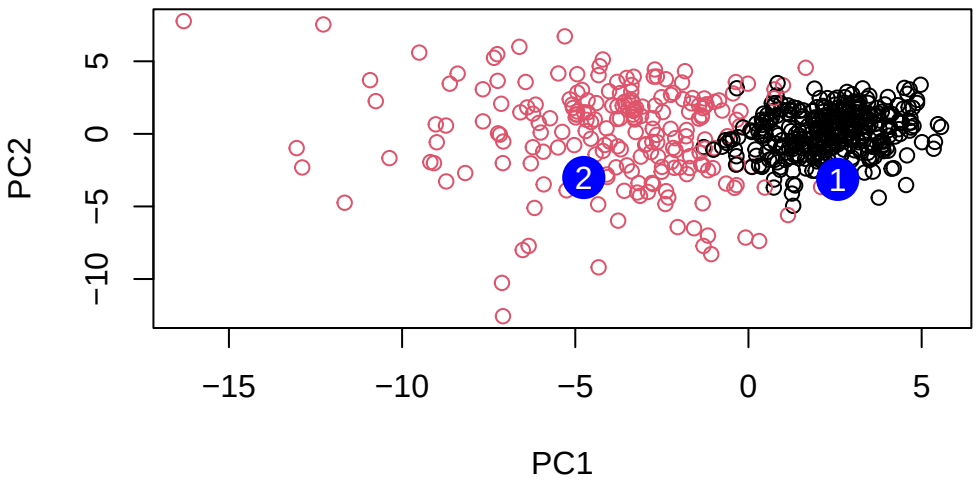
The 2/4 clusters model with PCA resulted best specificity and sensitivity.

#6. Prediction

```
url <- "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
```

```r
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?

Patient 2 should be prioritized as it falls in the malignant groups.