Unsupervised Mini Project

Sanket Garg (A59026686)

Preparing the Data

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

Removing the First column since it is not required.

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

Storing Diagnosis as a separate variable.

```
diagnosis <- as.factor(wisc.df[,1])

diagnosis
```

```
  [1] M M M M M M M M M M M M M M M M M B B B M M M M M M M M M M M M M M
 [38] B M M M M M M M M B M B B B B B M M B M M B B B B M B M M B B B B M B M M
 [75] B M B M M B B B M M B M M M B B B M B B M M B B B M M B B B M B B M B B
[112] B B B B B B M M M B M M B B B M M B M B M M B M M B B M B B M B B B B M B
[149] B B B B B B B B M B B B B M M B M B B M M B B M M B B B B M B B M M M B M
[186] B M B B B M B B M M B M M M M B M M M B M B M B B M B M M M M B B M M B B
[223] B M B B B B B M M B B M B B M M B M B B B B M B B B B B M B M M M M M M M
[260] M M M M M M M B B B B B B M B M B B M B B M B M M B B B B B B B B B B B B
[297] B M B B M B M B B B B B B B B B B B B B B B M B B B M B M B B B B M M M B B
[334] B B M B M B M B B B M B B B B B B B M M M B B B B B B B B B B M M B M M
[371] M B M M B B B B B M B B B B B M B B B M B B M M B B B B B B M B B B B B B
[408] B M B B B B B M B B M B B B B B B B B B B B B M B M M B M B B B B B M B B
[445] M B M B B M B M B B B B B B B B M M B B B B B B M B B B B B B B B B M B
[482] B B B B B B M B M B B M B B B B B B M M B M B M B B B B B M B B M B M B M M
[519] B B B M B B B B B B B B B B B M B M M B B B B B B B B B B B B B B B B B B
```

```
[556] B B B B B B B M M M M M M B
Levels: B M
```

Q1. How many observations are in this dataset?

```
dim(wisc.data)
```

```
[1] 569  30
```

There are 569 observations in this dataset.

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

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

```
[1] 212
```

212 observations have a 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
```

The first 10 columns names in the data are suffixed with _mean.

2. Principal Component Analysis

Checking 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

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

0.4427 proportion 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?

Only the first 3 PCs are required to describe at least 70% of the original variance in the data.

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

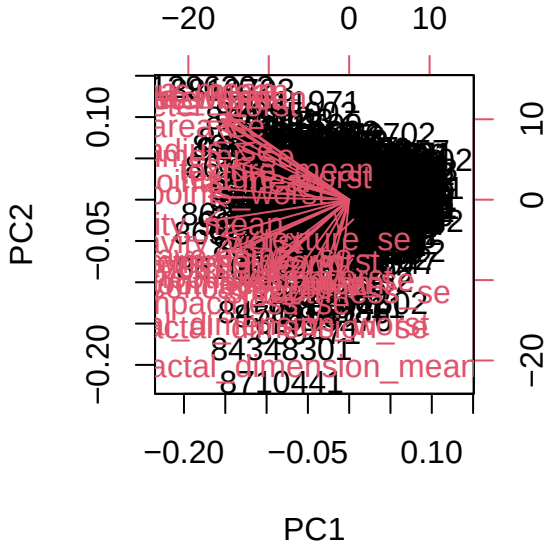
The first 7 PCs are required to describe at least 90% of the original variance in the data.

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

It is difficult to make sense of this plot because it is highly clustered and difficult to visualize individual data points.

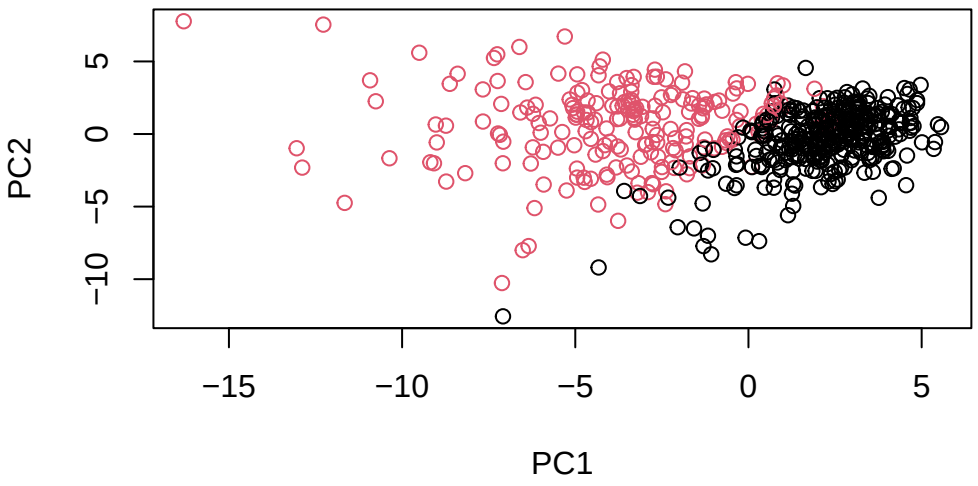
Making a biplot:

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



Scatter plot of PC1 vs PC2

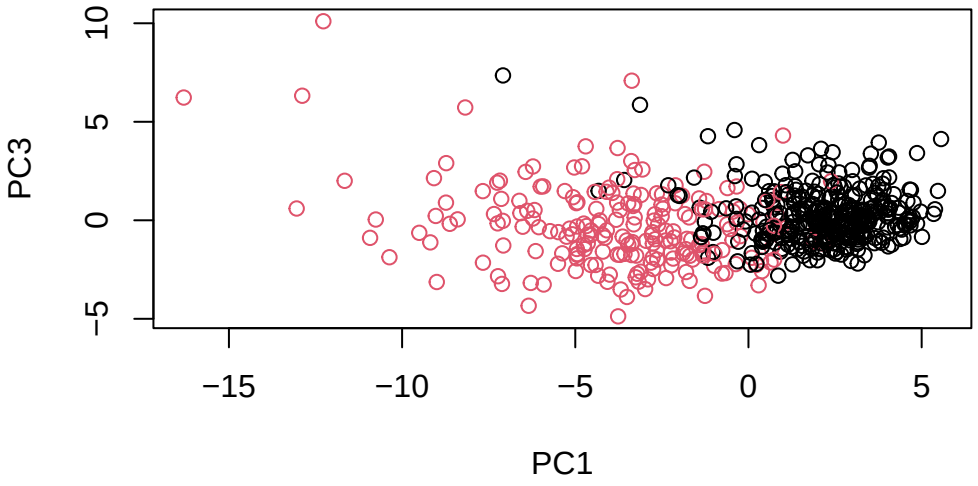
```
plot(wisc.pr$x[,1:2], col = diagnosis, xlab="PC1", ylab="PC2")
```



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

The Malignant ones are relatively well separated from the Benign ones. Scatter plot for PC1 vs PC3:

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

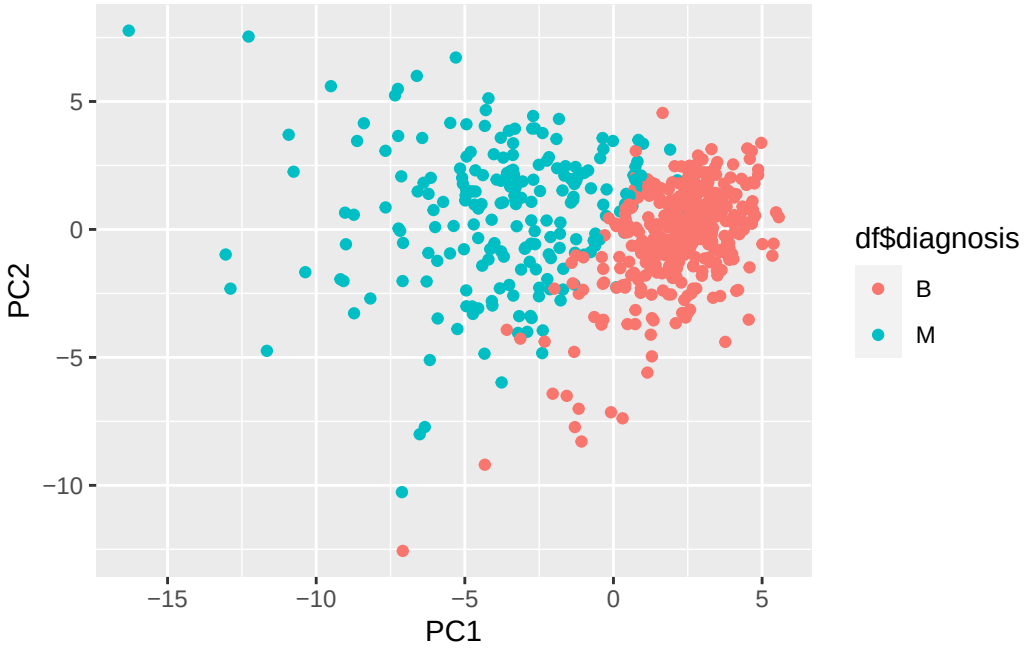


Using ggplot for fancier plots:

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

library(ggplot2)

ggplot(df) +
  aes(x = df$PC1, y=df$PC2, col=df$diagnosis) +
  geom_point() +
  labs(x = "PC1", y="PC2")
```



Calculate variance of each component:

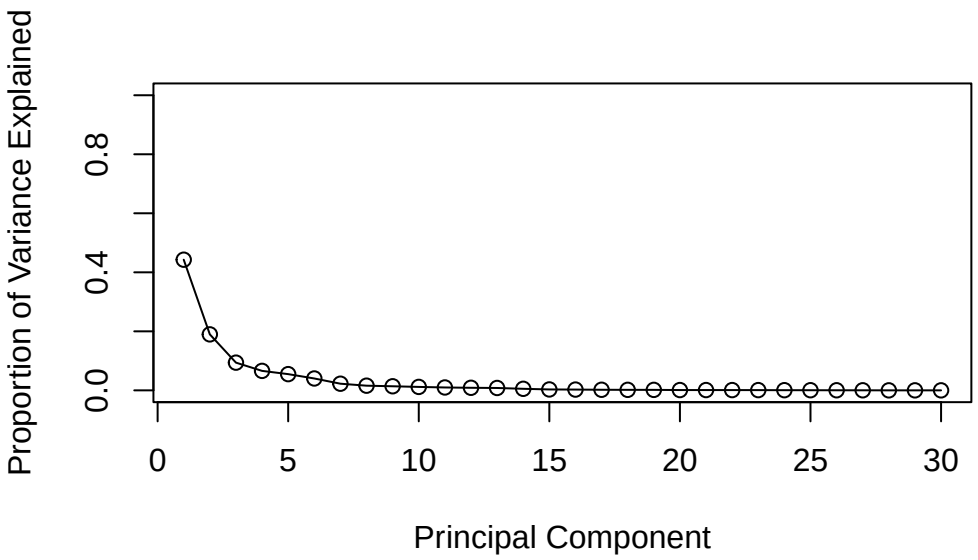
```
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

Variance explained by each component:

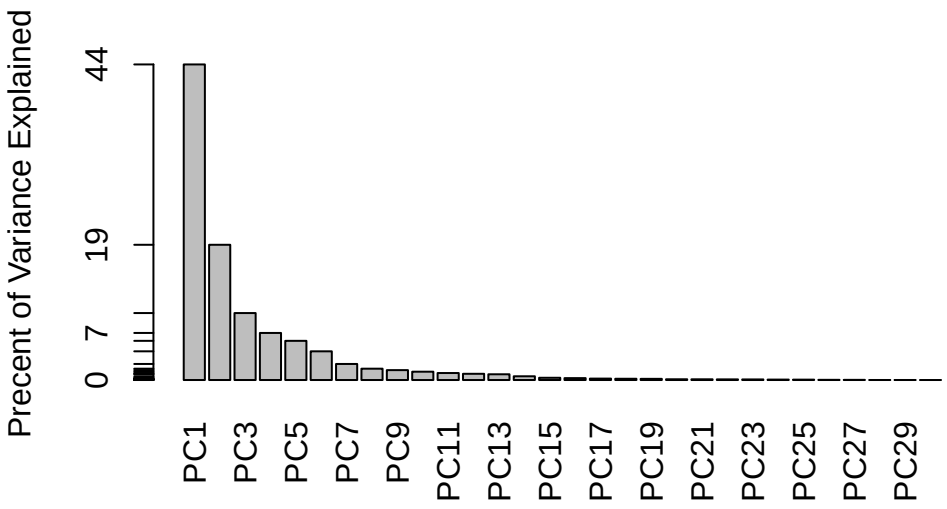
```
pve <- pr.var/sum(pr.var)

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



Scree plot in the form of bars:

```
# 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 )
```

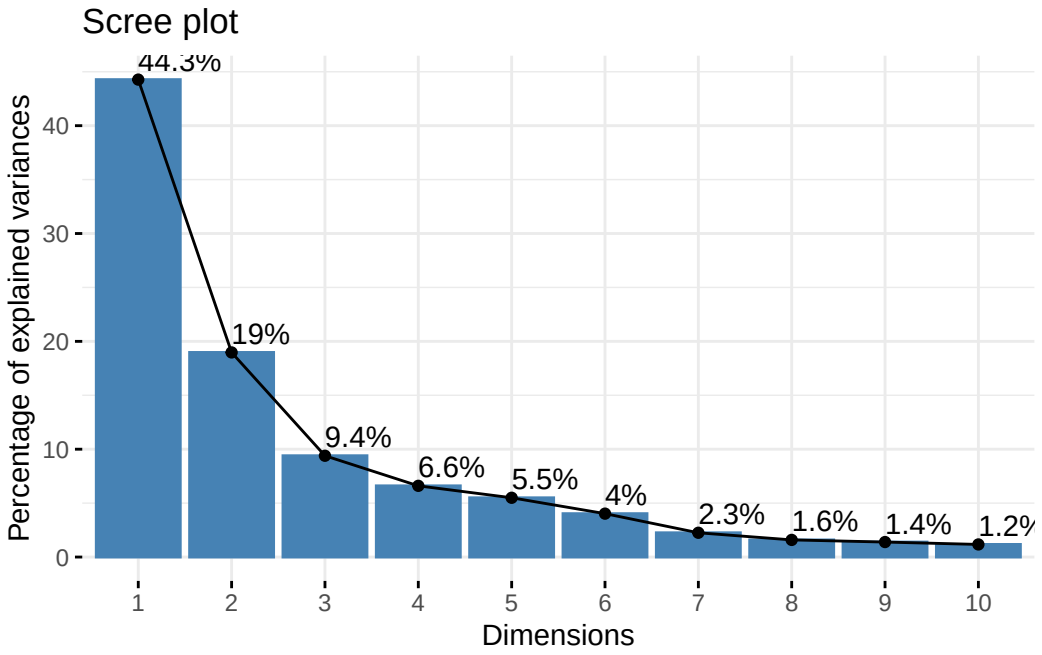


Using the factoextra package:

```
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["concave.points_mean",1]
```

```
[1] -0.2608538
```

#3. Hierarchical Clustering

Scale the data:

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

Calculate the Euclidean distance between all pairs of observations:

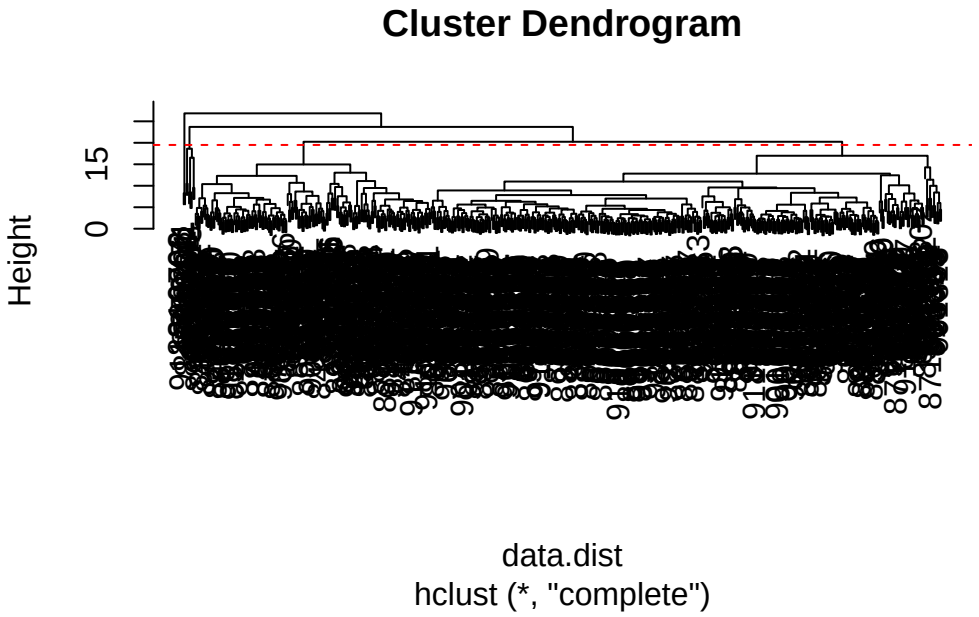
```
data.dist <- dist(data.scaled)
```

Perform the clustering:

```
wisc.hclust <- hclust(data.dist, "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=19.5, col="red", lty=2)
```



Use cutree() to cut the tree so that it has 4 clusters

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

Using table

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

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

Q11. Trying with a different number of clusters

```
wisc.hclust.clusters <- cutree(wisc.hclust, k=8)
table(wisc.hclust.clusters, diagnosis)
```

```
                    diagnosis
wisc.hclust.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
```

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

```
wisc.hclust_new <- hclust(data.dist, "ward.D2")
wisc.hclust_new.clusters <- cutree(wisc.hclust_new, k=4)
table(wisc.hclust_new.clusters, diagnosis)
```

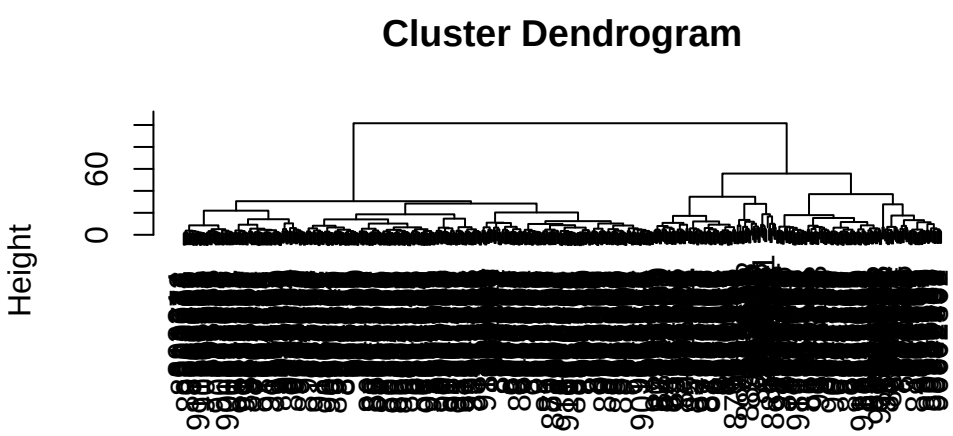
```
                        diagnosis
wisc.hclust_new.clusters   B   M
                       1   0 115
                       2   6  48
                       3 337  48
                       4  14   1
```

For grouping into 4 clusters, the "ward.D2" and the "complete" methods generate the best results for a total of 4 clusters.

#4. Combining the Results

Clustering on PCA results

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



Checking the clustering results:

```
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
```

Plotting based on the grps

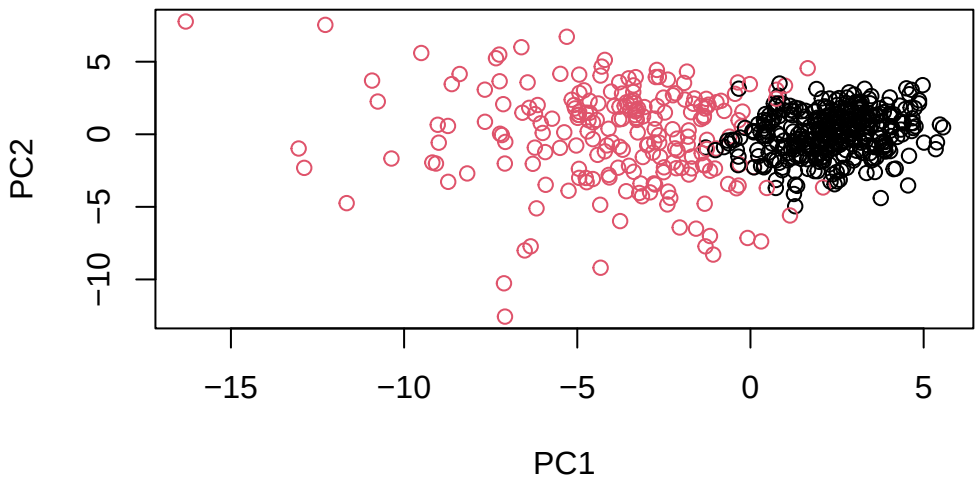
```
g <- as.factor(grps)
levels(g)
```

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

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

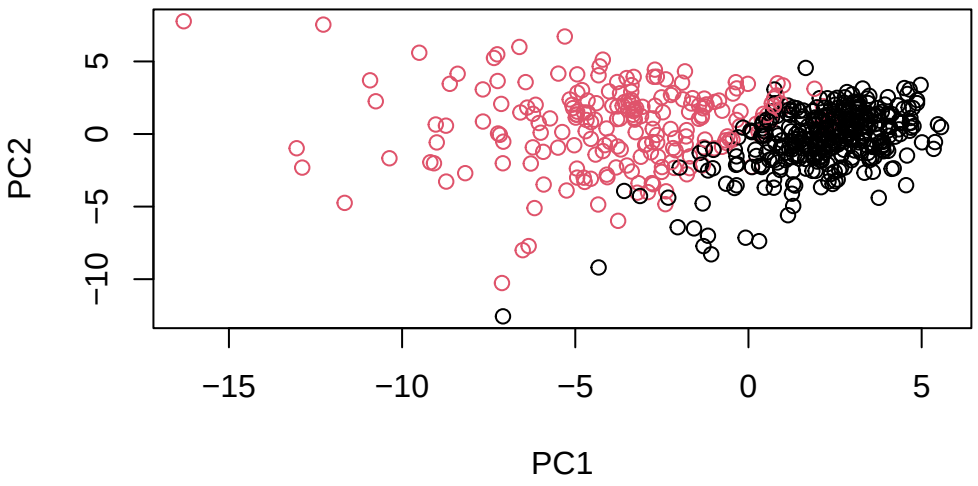
```
[1] "2" "1"
```

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



Plotting based on diagnosis variable

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



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

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

```
        diagnosis
new_grps   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.

```
a <- table(wisc.hclust_new.clusters, diagnosis)
a
```

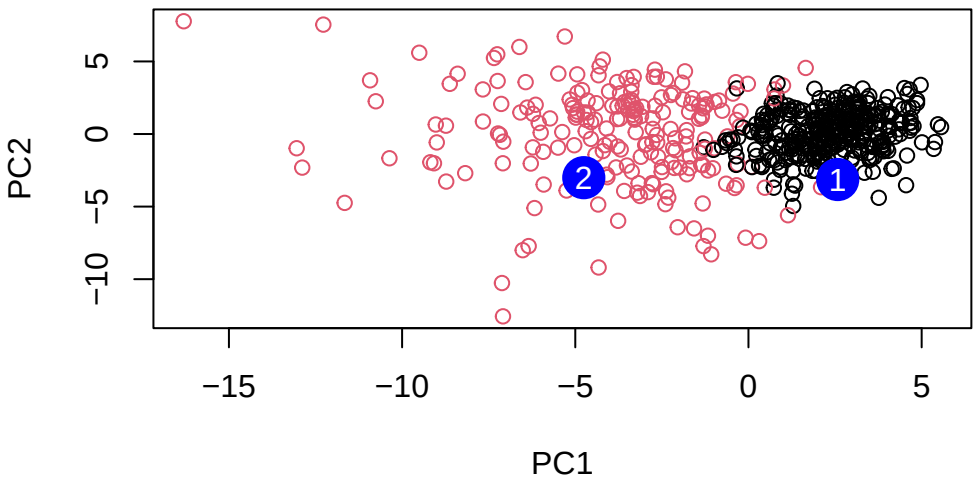
```
                         diagnosis
wisc.hclust_new.clusters   B   M
                       1   0 115
                       2   6  48
                       3 337  48
                       4  14   1
```

#6. 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
```

```
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?

Based on the results, we should prioritize the patient 2 for follow up based on the results.