Class 08 Mini Project

Austin Teel (A17293709)

Table of contents

Background

The goal of this project is to apply what R techniques we have learned in this course such as machine learning, in order to analyze the data that we are getting from the *Wisconsin Breast Cancer Diagnostic Data Set*

The data set offers features of imaged biopsy samples. These features consist of `radius`,`texture`,`perimeter`, `area`, `smoothness`, `compactness`, `concativity`, and `symmetry`.

Exploratory data analysis

The data is in CSV format:

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

```r
head(wisc.df, 4)
```

```
         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
         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
         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
         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
         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
         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
         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
         fractal_dimension_worst
842302                   0.11890
842517                   0.08902
84300903                 0.08758
84348301                 0.17300
```

Now that we have the data loaded we will create a new data frame in order to exclude the pathologist provided diagnosis because we will not need this in our data.

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

We must now seperate the diagnosis column and we can do this by creating a diagnosis vector.

```r
diagnosis <- wisc.df[,1]
```

> Q1. How many observations are there in this dataset?

We can answer this question by analyzing the data with a `nrow()` function.

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

```
[1] 569
```

Through this function we can find that there is 569 rows which means this is how many observations there was.

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

To answer this question we can use a `sum()` function on our vector that holds the diagnosis and see how many of those vector values are Malignant(M)

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

```
[1] 212
```

We have 212 malignant diagnosis.

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

This can be found with the `grep()`.

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

```
[1] 10
```

This shows us that there are 10 variables in the data suffixed with `_mean`.

Performing PCA

The main functinon in base R is called `prcomp()` we will use the optional argument `scale=TRUE` here as the data columns/features/dimensions are on very different scales in the original data set.

```
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)
```

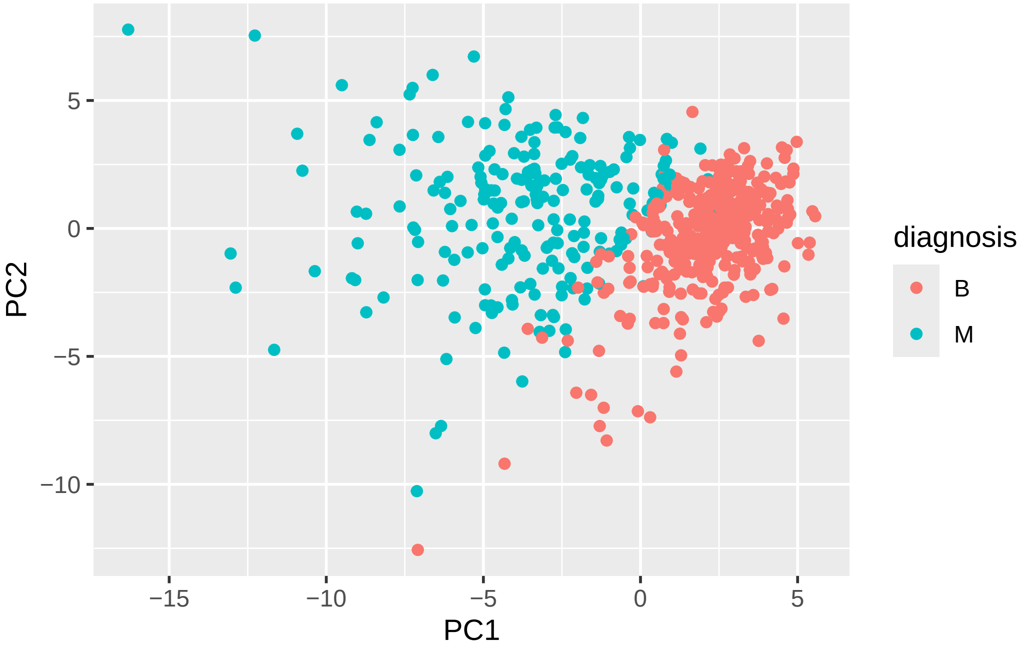
```
            radius_mean            texture_mean          perimeter_mean 
           3.524049e+00            4.301036e+00            2.429898e+01 
              area_mean         smoothness_mean        compactness_mean 
           3.519141e+02            1.406413e-02            5.281276e-02 
         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 
```

```r
wisc.pr <- prcomp( wisc.data, scale=T)
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
```

```r
library(ggplot2)

ggplot(wisc.pr$x)+
  aes(PC1,PC2, col=diagnosis)+
  geom_point()
```



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

The function above `summary(wisc.pr)` tells us that the first PC has 0.4427 proportion fo variance.

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

```
propvar <- summary(wisc.pr)$importance["Cumulative Proportion", ]

which(propvar > 0.70)[1]
```

```
PC3
  3
```

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

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

```
which(propvar > 0.90)[1]
```

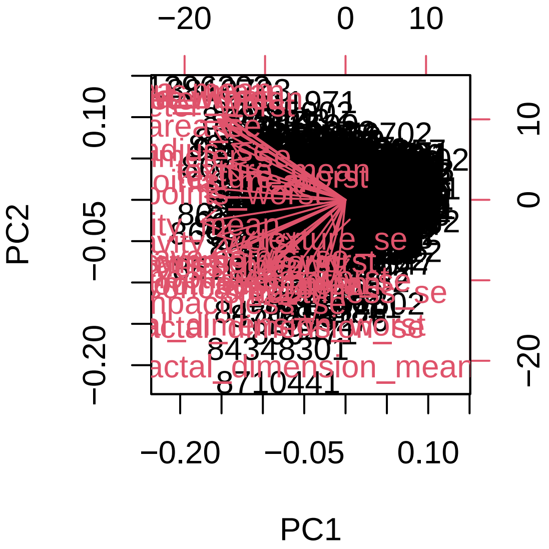
```
PC7 
  7 
```

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

Interpreting PCA results

> Q.7 What stands out to you about this plot? Is it easy or difficult to understand? Why?

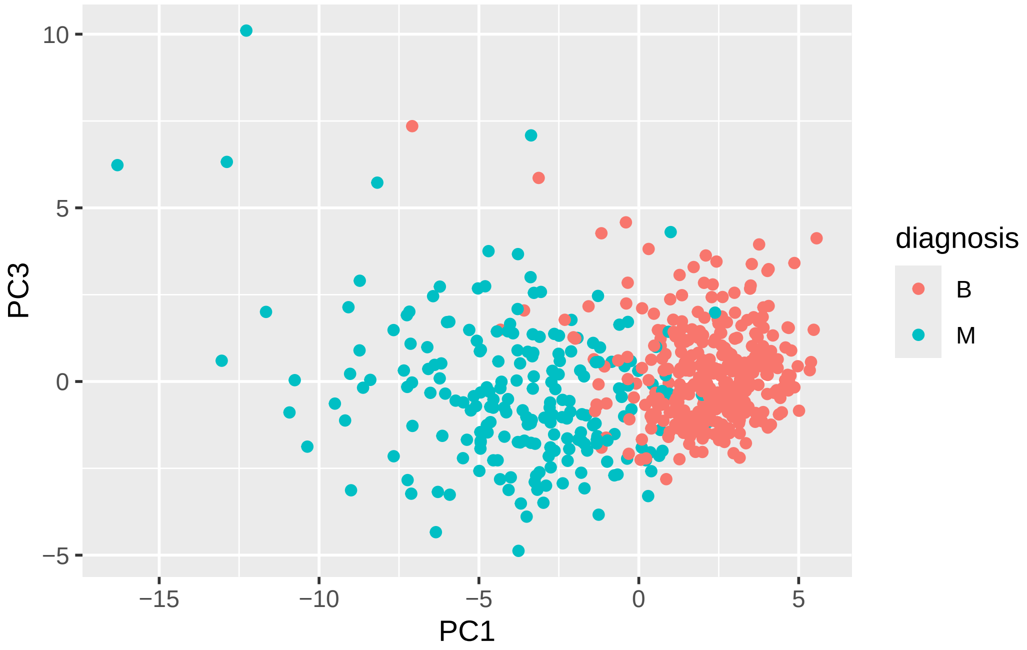
```
biplot(wisc.pr)
```



This plot is very difficult to understand because the cluster in unable to be seen clearly by the viewer.

> Q.8 Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

```
ggplot(wisc.pr$x)+
  aes(PC1,PC3, col=diagnosis)+
  geom_point()
```



In this plot I notice that there is still a clear separation of B and M when comparing PC1 to PC3.

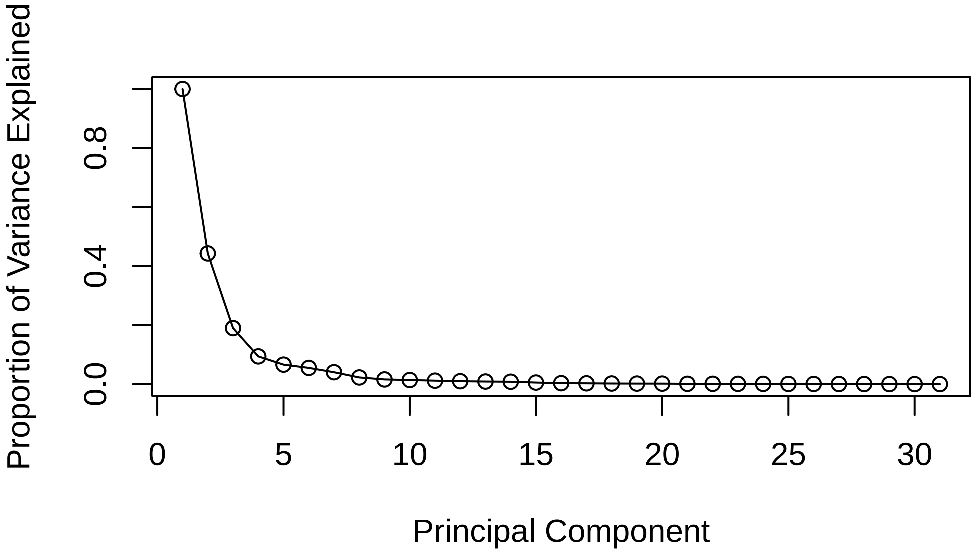
Variance Explained

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

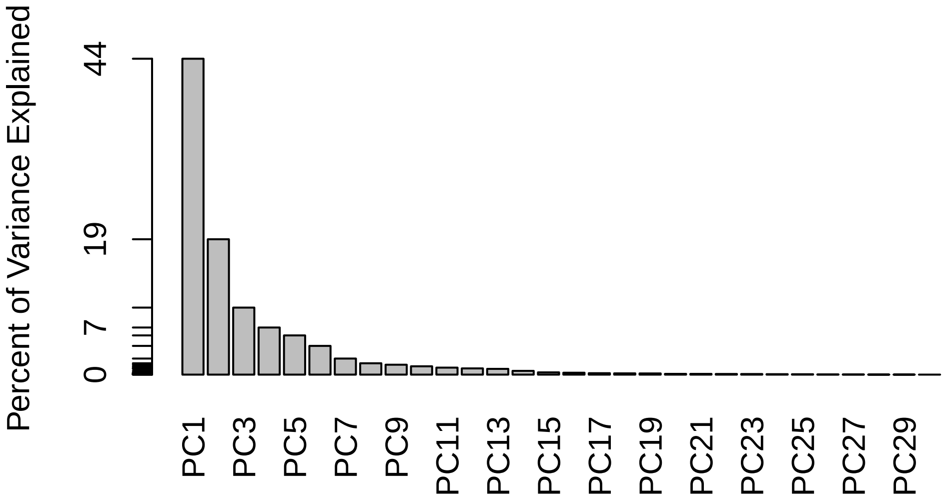
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

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

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



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



> Q.9 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. Are there any features with larger contributions than this one?

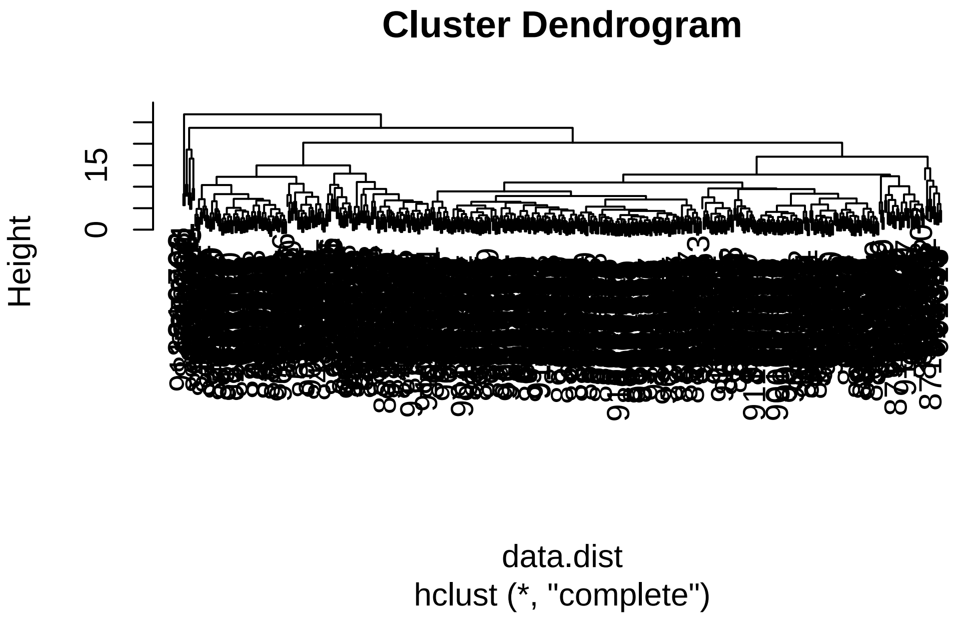
The component of the loading vector is

Hierarchical Clustering

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

```
[1] 10.309426  6.771675 10.463467  8.663413  8.402233  9.843286
```

```
wisc.hclust <- hclust(data.dist)
plot(wisc.hclust)
```

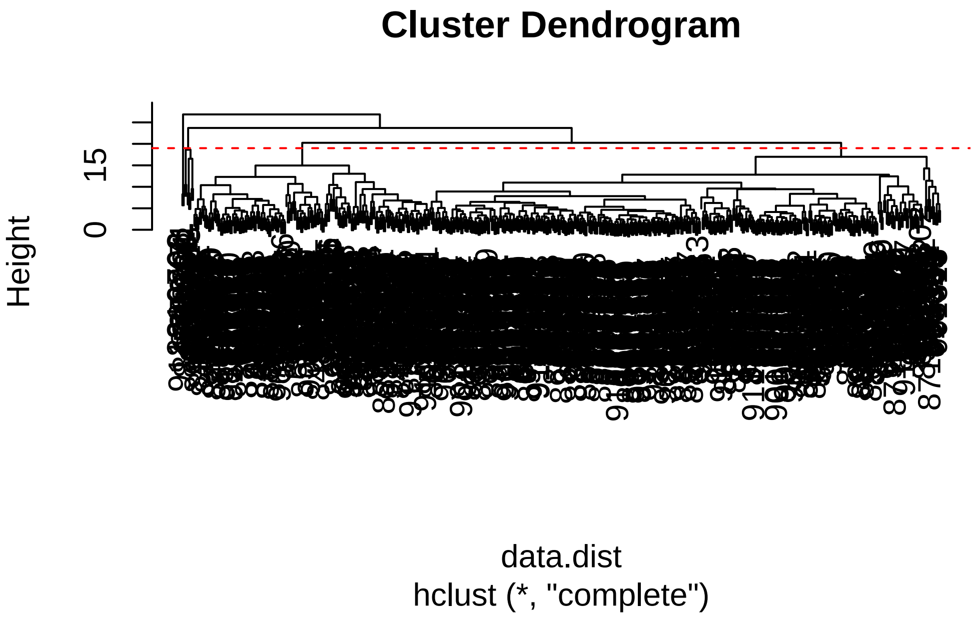


Results of hierarchical clustering

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

19

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



```
wisc.hclust.clusters <- cutree(wisc.hclust, h=19)
```

Using Different Methods

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

The method that gives me my favorite results for the same data. dist was the tree because with the cutree function we are abel to seperate the groups.

Combining methods

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

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

```
grps
  1   2 
203 366 
```

I can now run `table()` with both my clustering `grps` and the expert `diagnosis`.

> Q13. How well does the newly created hclust model with two clusters separate out the two “M” and “B” diagnoses?

The new model seperated the two diagnosis very well and gave us the specific amounts in each group.

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

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

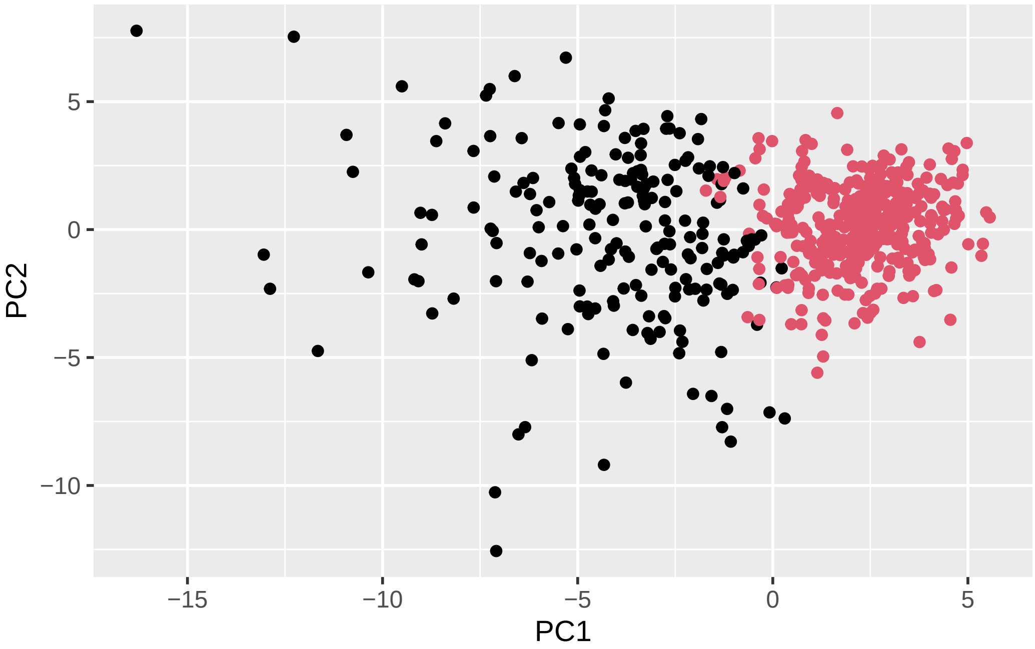
> Q14. How well do the hierarchical clustering models you created in the previous sections (i.e. without first doing PCA) do in terms of separating the diagnoses? Again, use the table() function to compare the output of each model (wisc.hclust.clusters and wisc.pr.hclust.clusters) with the vector containing the actual diagnoses.

The hierarchical clustering models that we created in the previous section did not do well in separating the true diagnosis of the patient but the following does.

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

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

```
ggplot(wisc.pr$x) +
  aes(PC1, PC2) +
  geom_point(col=grps)
```



Sensitivity

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

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

Our cluster “1” has 179 “M” diagnosis Our cluster “2” has 333 “B” diagnosis

179 TP 24 FP 333 TN 33 FN

sensitivity: TP/ (TP+FN)

```
179/(179+33)
```

```
[1] 0.8443396
```

Specifity: TN/(TN+FP)

```
333/(333+24)
```

```
[1] 0.9327731
```

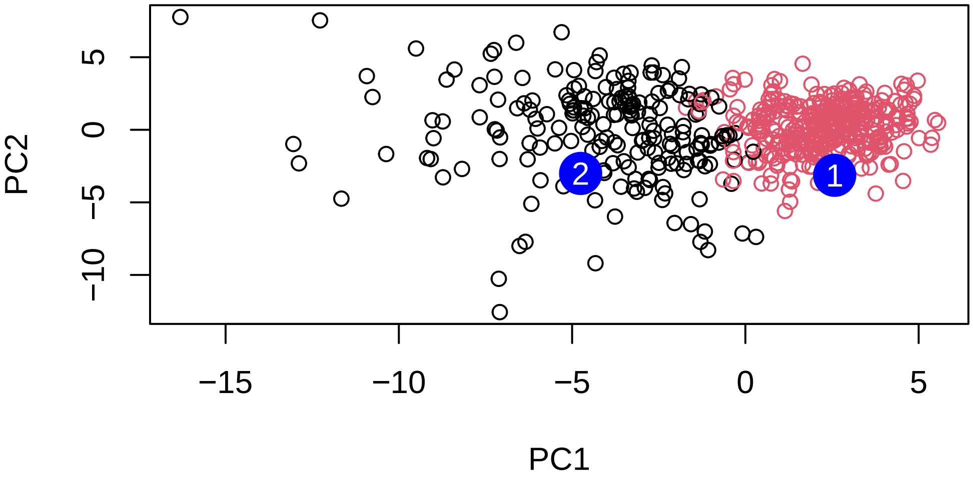
Prediction

We can use our PCA model

```
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=grps)
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 these results we should prioritize the people in group two due to the fact that they are the malignant patients.