Class 08 Mini-Project

Sindy Chavez

Unsupervised learning analysis of breast cancer cells

First I need to get the data. This comes from the University Of Wisconsin and lives online here;

It has been downloaded onto my computer in the same file as the class 08 mini project

```
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
842302                   0.11890
842517                   0.08902
84300903                 0.08758
84348301                 0.17300
84358402                 0.07678
843786                   0.12440
```

```
dim(wisc.df)
```

```
[1] 569  31
```

How could we get rid of this expert ‘diagnosis’ column.

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

> Q1. How many individuals (i.e. samples/rows)?

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

```
[1] 569
```

> Q2. How many "cancer" M samples are there?

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

```

  B   M 
357 212 
```

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

The 'grep()' function would be useful.

```r
colnames(wisc.data)
```

```
 [1] "radius_mean"             "texture_mean"           
 [3] "perimeter_mean"          "area_mean"              
 [5] "smoothness_mean"         "compactness_mean"       
 [7] "concavity_mean"          "concave.points_mean"    
 [9] "symmetry_mean"           "fractal_dimension_mean" 
[11] "radius_se"               "texture_se"             
[13] "perimeter_se"            "area_se"                
[15] "smoothness_se"           "compactness_se"         
[17] "concavity_se"            "concave.points_se"      
[19] "symmetry_se"             "fractal_dimension_se"   
[21] "radius_worst"            "texture_worst"          
[23] "perimeter_worst"         "area_worst"             
[25] "smoothness_worst"        "compactness_worst"      
[27] "concavity_worst"         "concave.points_worst"   
[29] "symmetry_worst"          "fractal_dimension_worst"
```

Ask for '_mean' pattern withinin the columns (colnames(wisc.data))

```r
match.positions <- grep("_mean", colnames(wisc.data))
length(match.positions)
```

```
[1] 10
```

Principal Component Analysis (PCA)

Let's try PCA on this data to see what major features might be hidden in this high dimensional data that are hard to see any other way.

The function 'prcomp()' is useful, but you need to change the default 'scale()' from FALSE to TRUE so that R can change the scales and make it easier to compare the data as apples to apples

Do we need to "scale" this data before PCA? We look at the mean and sd of the variables (i.e. columns)

If the units of the data are already the same, you shouldn't have to scale

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

```
            radius_mean            texture_mean          perimeter_mean 
                   3.52                    4.30                   24.30 
              area_mean         smoothness_mean        compactness_mean 
                 351.91                    0.01                    0.05 
         concavity_mean     concave.points_mean           symmetry_mean 
                   0.08                    0.04                    0.03 
 fractal_dimension_mean               radius_se              texture_se 
                   0.01                    0.28                    0.55 
           perimeter_se                 area_se           smoothness_se 
                   2.02                   45.49                    0.00 
         compactness_se            concavity_se       concave.points_se 
                   0.02                    0.03                    0.01 
            symmetry_se    fractal_dimension_se            radius_worst 
                   0.01                    0.00                    4.83 
          texture_worst         perimeter_worst              area_worst 
                   6.15                   33.60                  569.36 
       smoothness_worst       compactness_worst         concavity_worst 
                   0.02                    0.16                    0.21 
   concave.points_worst          symmetry_worst fractal_dimension_worst 
                   0.07                    0.06                    0.02 
```

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

0.4427

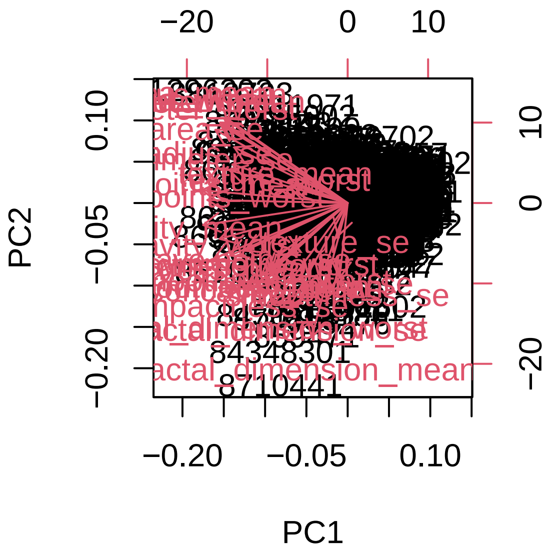
> Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data?

3

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

7

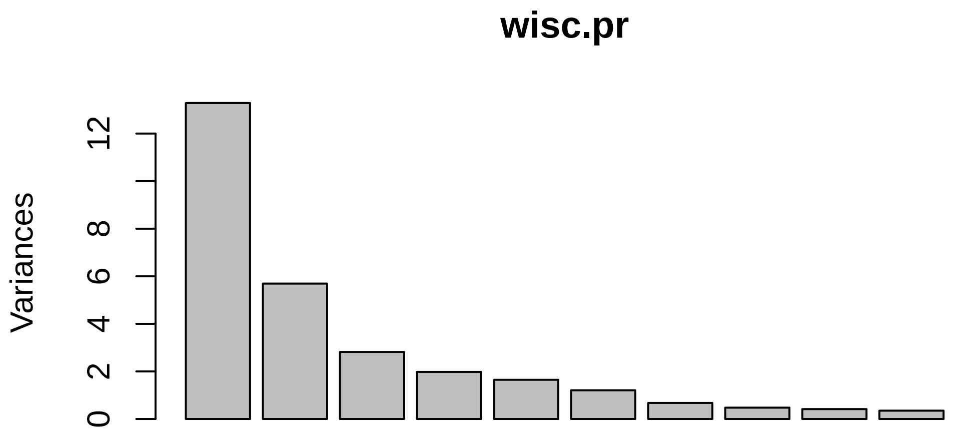
```
biplot(wisc.pr)
```



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

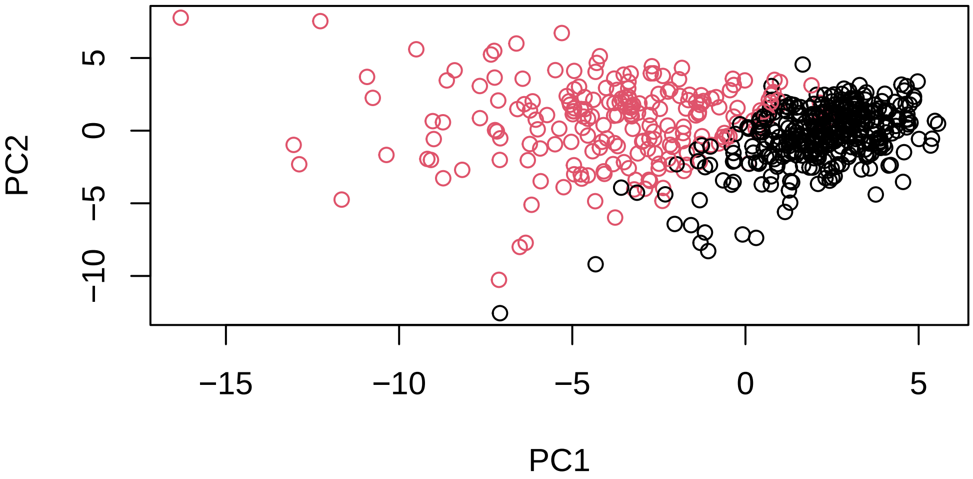
It has all the dimensions, but is too messy to read because there are too many labels.

```
plot(wisc.pr)
```



One of our main results from methods like PCA is a so called "score plots" a.k.a. "PC plots", "ordination plots", "PC1 vs PC2", etc. Let's make one ourselves...

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



Each dot in the plot is a person You can almost draw a line between benign and malignant tumors

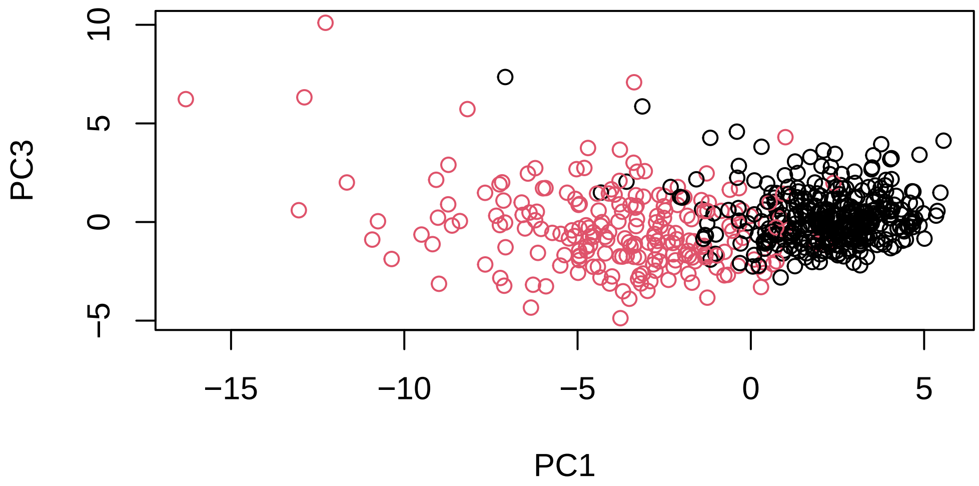
PCA is a method for compressing a lot of data into something that captures the essence of the original data

Takes a data set with lots of dimensions and flattens it to 2 or 3 dimensions(I think it said dimensions? - try looking at Lecture Notes)

The new rotated axes describe the Principal components - from drawing new axes along the data (if the data has a general correlation)

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

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



The two plots look very similar. Both the PC1vPC2 and PC1vPC3 have somewhat clear clusters, but there is more overlap between the clusters in the PC1vPC3 plot.

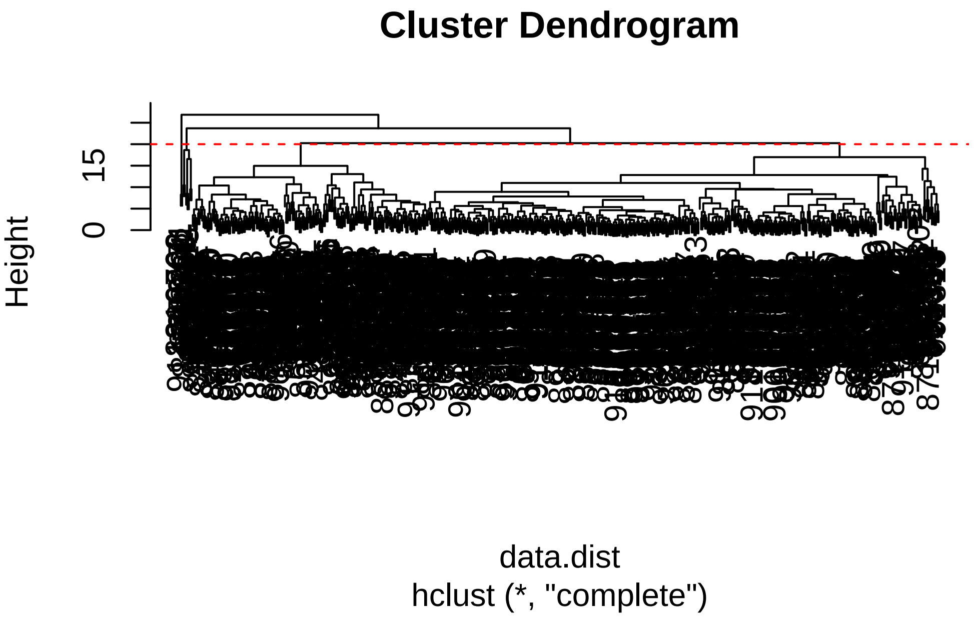
> Q9. For the first principal component, what is the component of the loading vector (i.e. wisc.pr$rotation[,1]) for the feature concave.points_mean? This tells us how much this original feature contributes to the first PC.

I think this is the point where most of the variance has been captured. I think it's also called the 'elbow' or 'scree'.

Hierarchical clustering

Kinda sucks but here is goes

```r
data.scaled <- scale(wisc.data)
data.dist <- dist(data.scaled)
wisc.hclust <- hclust(data.dist)
plot(wisc.hclust)
abline(h=20, col="red", lty=2)
```



Not much structure in the original data, trying to cut with the ’cutree()” function only leads to a mess.

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

Height 20

> Using different methods

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

I tried the ‘ward.D2’ method later and it created a cleaner dendrogram where it was easier to see how the data might cluster.

Combining methods

Clustering on PCA results

PCA is often used as a first step in further analysis. Here we will combine PCA and clustering.

We have our PCA results ‘wisc.pr$x’

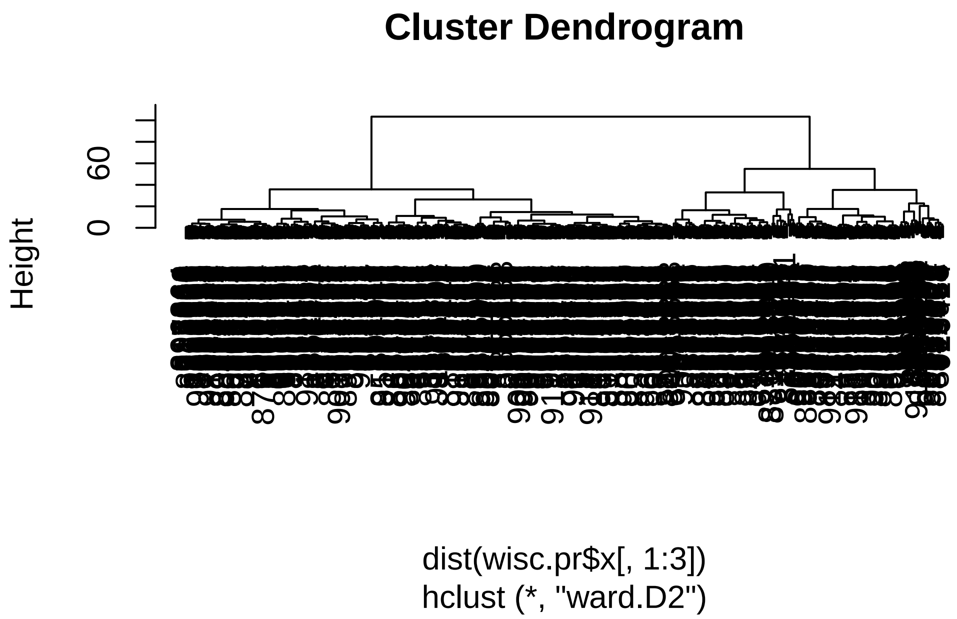
wisc.pr$x[,1:3]

```
summary(wisc.pr$x[,1:3])
```

```
      PC1                PC2                 PC3
 Min.   :-16.305   Min.   :-12.5621   Min.   :-4.8761
 1st Qu.: -2.507   1st Qu.: -1.2272   1st Qu.:-1.0162
 Median :  1.173   Median :  0.1958   Median :-0.1556
 Mean   :  0.000   Mean   :  0.0000   Mean   : 0.0000
 3rd Qu.:  2.755   3rd Qu.:  1.4985   3rd Qu.: 0.8561
 Max.   :  5.561   Max.   :  7.7690   Max.   :10.1035
```

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

plot(wisc.pr.hclust)
```

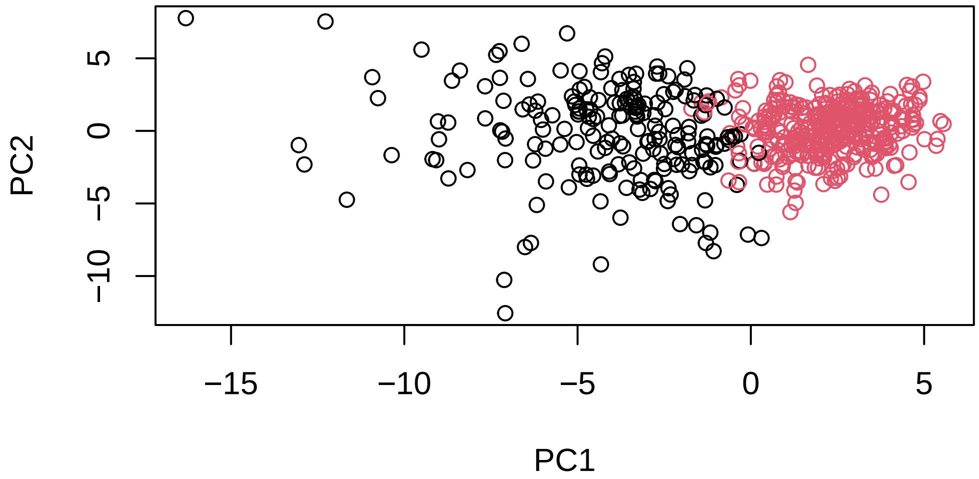


```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

grps

```
  1   2
203 366
```

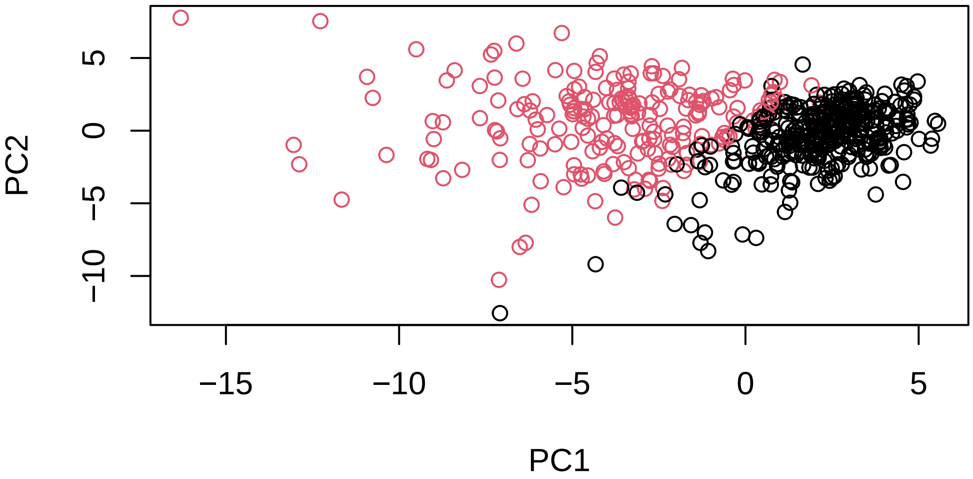
```
plot(wisc.pr$x[,1:2], col=grps)
```



The colors in the plot above are reversed from the original colors we had for malignant and benign. Let's flip the colors

Color diagnosis instead of grps

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



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

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

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

The model does a pretty good job of making two groups where most of group 1 aligns with a *malignant* diagnosis and most of group 2 aligns with a *benign* diagnosis.

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

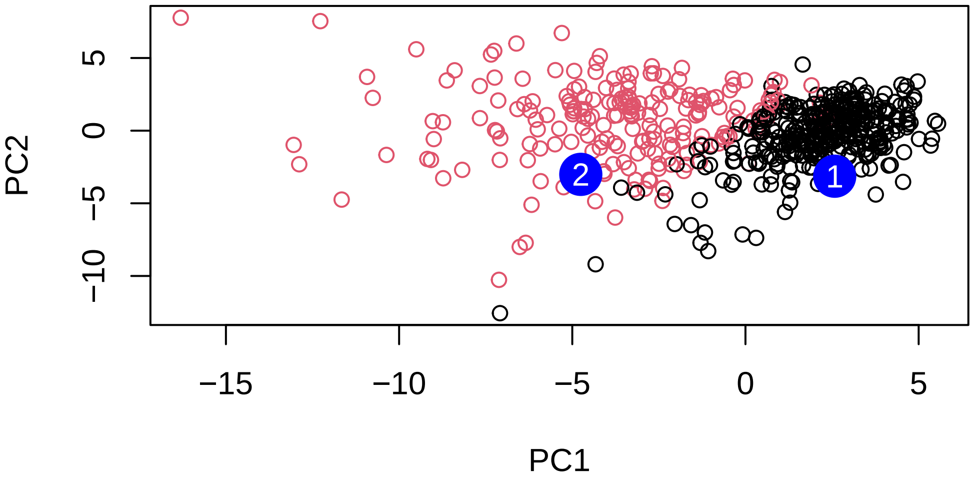
```
                       diagnosis
wisc.pr.hclust.clusters   B   M
                      1  24 179
                      2 333  33
```

Prediction

```r
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
```

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