Class 08 Mini Project

Kiana Bohanon (PID: A18702316)

Table of contents

Background

In today's class we will apply the methods and techniques clustering and PCA to help make sense of a real world breast cancer **FNA (fine needle aspirations)** biopsy data set.

Data Import

We start by importing our data. It is a CSV file, so we will use the `read.csv()` function. First, download the file containing the data, and place it in the project for this class on your computer.

```r
#read.csv("WisconsinCancer.csv")
```

```r
# Save your input data file into your Project directory
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
```

Omit the first column of `diagnosis` because I don't want to use this for my machine learning models. We will use it later on to compare our results to the expert diagnosis:

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

# Create diagnosis vector for later
diagnosis <- wisc.df$diagnosis
```

Exploratory Data Analysis

Q1. How many observations are in this dataset?

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

```
[1] 569
```

569 observations are in this dataset.

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

```
# Use the table() function:
#table(diagnosis)
table(diagnosis)["M"]
```

```
  M
212
```

212 observations have a malignant diagnosis.

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

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

```
[1] 10
```

There are 10 variables/features in the data suffixed with `_mean`.

Principal Component Analysis

The main function here is `prcomp()` and we want to make sure we set the optional argument `scale=TRUE`: We will need to check if the data needs to be scaled before performing PCA by looking at the mean and standard deviation.

```
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 component (PC1)?

44.27%.

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

Three PCs are required to describe at least 70% of the original variance.

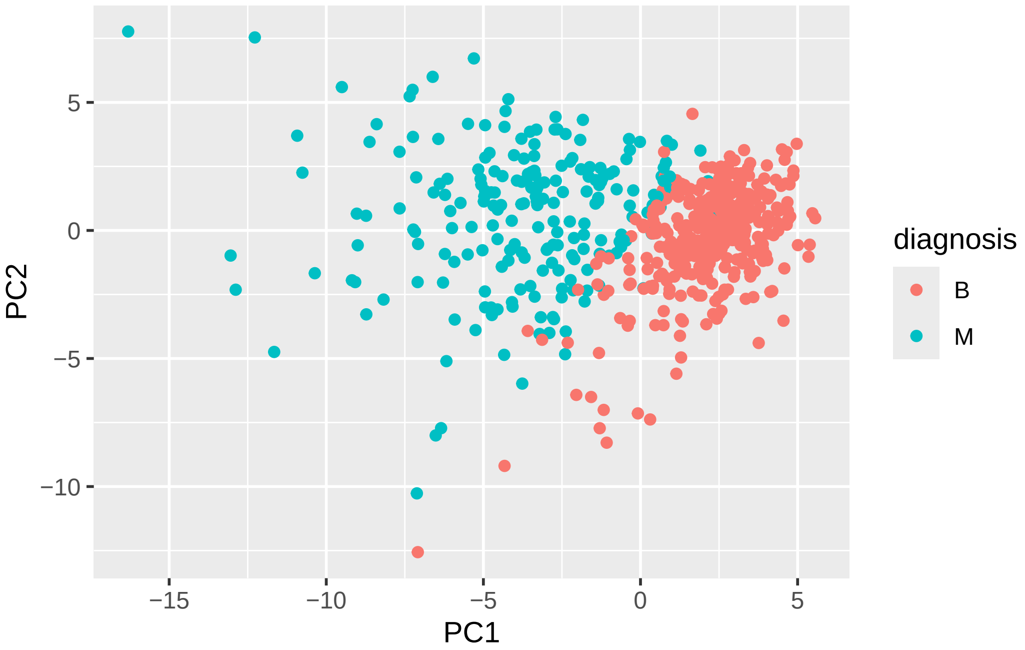
Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

Seven PCs are required to describe at least 90% of the original variance.

Our main PCA “score plot” or “PC plot” of results:

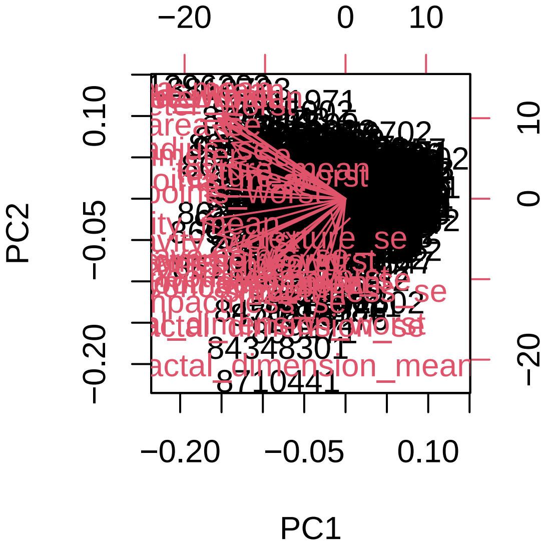
```
library(ggplot2)
```

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



Biplot:

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

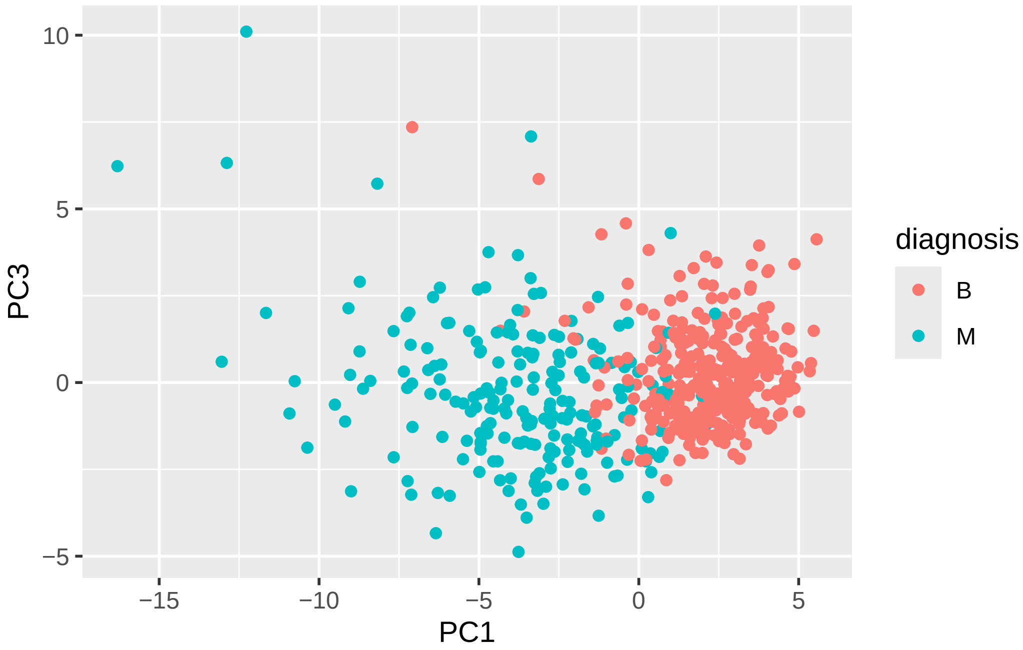


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

This plot is very confusing and not easy to understand. A lot of the data is on top of each other and you can't really tell anything or come to a conclusion easily.

Similar ggplot for PC1 and PC3:

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



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

These plots are a lot easier to interpret and you can easily tell where the trends are. The data is organized much nicer as well.

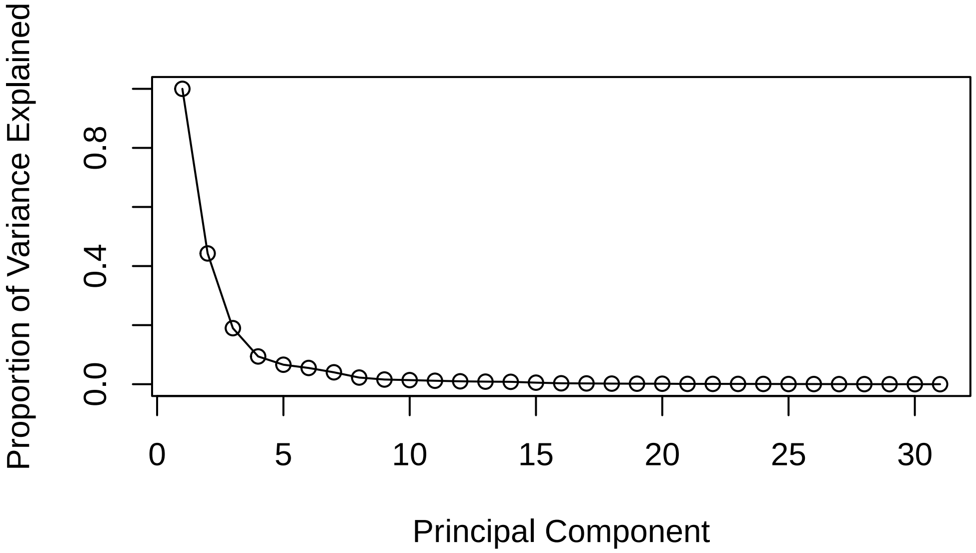
Variance Explained

```
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 <- wisc.pr$sdev^2 / sum(wisc.pr$sdev^2)

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



Communicating PCA Results

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

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

```
[1] -0.2608538
```

```
sort(abs(wisc.pr$rotation[, 1]), decreasing = TRUE)[1:5]
```

```
 concave.points_mean       concavity_mean concave.points_worst 
           0.2608538            0.2584005            0.2508860 
    compactness_mean      perimeter_worst 
           0.2392854            0.2366397 
```

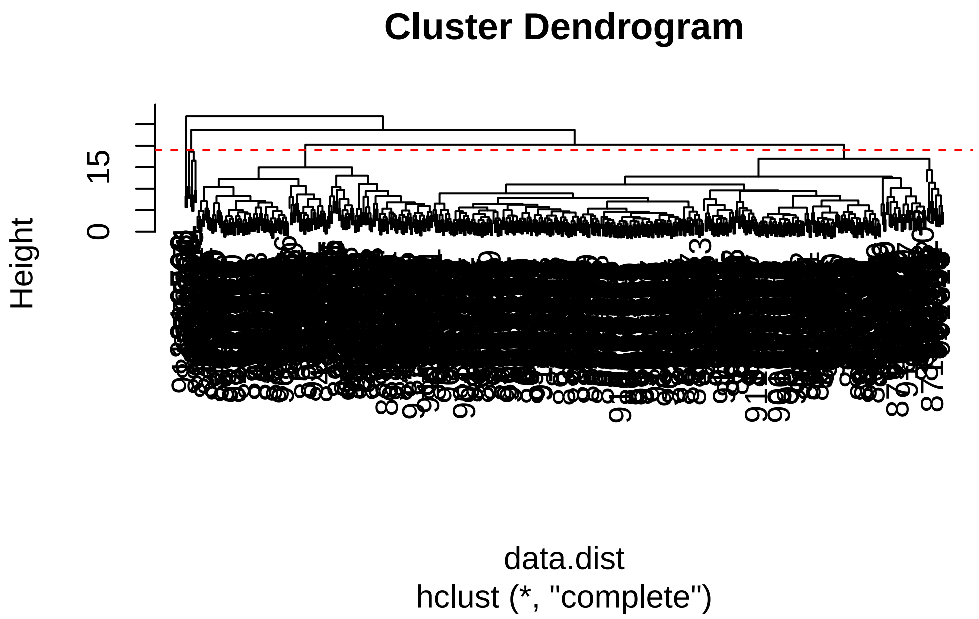
No, there are not any features with larger contributions than this one.

Hierarchical Clustering

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

Creating the cluster deprogram:

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



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

The height where the clustering model has 4 clusters is at 19.

Selecting number of clusters

```
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
```

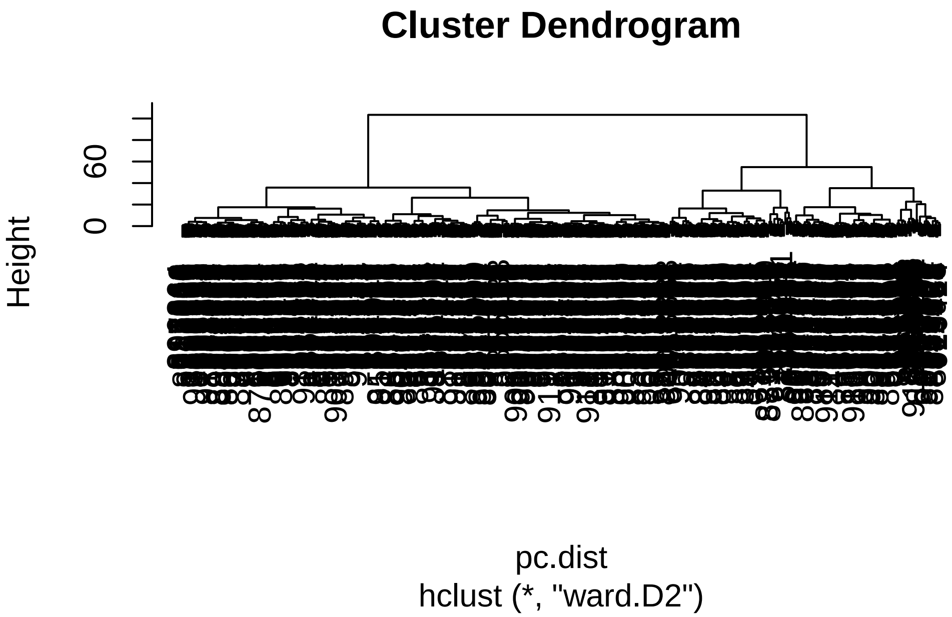
Q12. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

The ward.D2 method gives the best results because it minimizes within-cluster variance, producing compact and well-separated clusters. Compared to single, complete, and average linkage, ward.D2 avoids chaining and creates clusters that better reflect the underlying structure of the data.

Combining Methods

Here we will take our PCA results and use those as input for clustering. In other words our `wisc.pr$x` scores that we plotted above (the main output from PCA - how the data lie on our new principal component axis/variables) and use a subset of the PCs as input for `hclust()`.

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



Cut the dendrogram/tree into two main groups/clusters:

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

```
grps
  1   2 
203 366 
```

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

I want to know how clustering into `grps` with values of 1 or 2 correspond to the expert `diagnosis`.

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

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

My clustering **group 1** are mostly “M” diagnosis (179) and my clustering **group 2** are mostly “B” diagnosis.

24 False positives 179 True positives 333 True negatives 33 False negatives

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.

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

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

These models do fairly well in terms of separating the diagnosis, as the numbers are fairly similar.

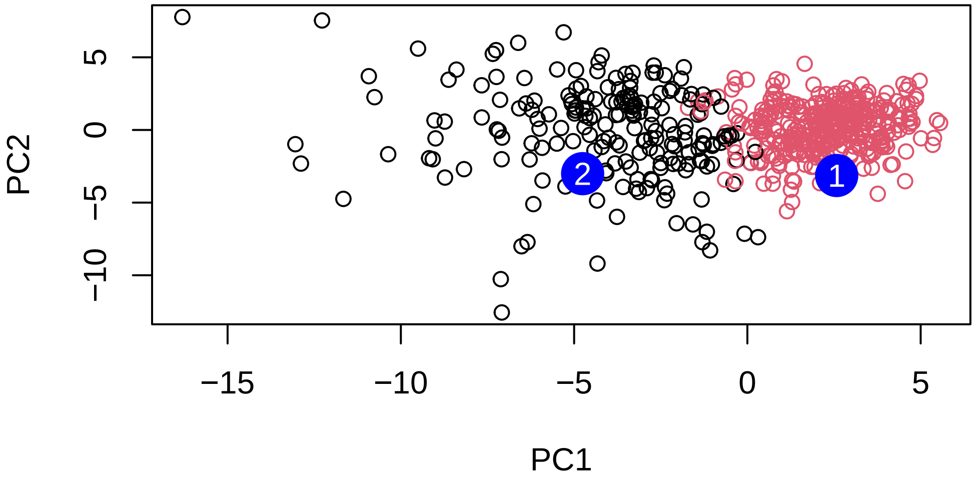
Prediction

```
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
```

```r
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?

We should prioritize Patient 2 because their results are more similar to the malignant plots than Patient 1, so it is more likely that they may have malignant cancer cells.