Class 8: Breast Cancer Analysis Mini Project

Serena Quezada (PID: A18556865)

Table of contents

Background

The goal of this mini-project is for you to explore a complete analysis using the unsupervised learning techniques covered in class.

The data itself comes from the Wisconsin Breast Cancer Diagnostic Data Set first reported by K. P. Benne and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets".

Values in this data set describe characteristics of the cell nuclei present in digitized images of a fine needle aspiration (FNA) of a breast mass.

Data Import

Data was downloaded from the class website as a CSV file

```
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
```

1. Data Exploration

The first column `diagnosis` is the expert opinion on the sample (i.e. patient FNA)

```
wisc.data <- wisc.df[,-1]
dim(wisc.data)
```

```
[1] 569  30
```

Store the diagnosis as a vector for use later when we compare our results to those from experts in the field

```
diagnosis <- factor(wisc.df$diagnosis)
```

Q1. How many observations are in this dataset?

There are 569 observations/patients in the dataset

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

There are 212 observations that have a malignant diagnosis

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

```
  B   M
357 212
```

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

There are 10 variables/features in the data set that the "_mean" suffix

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

```
[1] 10
```

2. Principal Component Analysis (PCA)

The `prcomp()` function to do PCA has a `scale=FALSE` default. In general we nearly always want to set this to TRUE so our analysis is not dominated by columns/variables in our dataset that have high standard deviation and mean when compared to others just because the unts of measurement are on different units/scales.

```
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)?

It is 0.4427

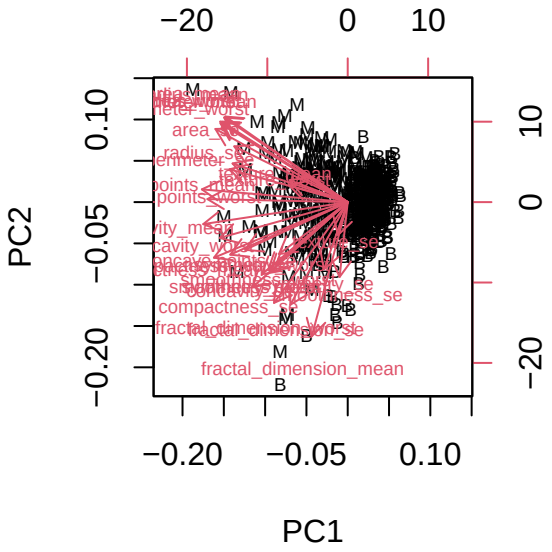
Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data?

The cumulative exceeds 0.70 after the third component (0.72636).

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

The cumulative proportion first passes the 0.90 threshold after the seventh component .91010.

```
biplot(wisc.pr,
       xlabs = as.character(diagnosis),
       cex = 0.6)
```

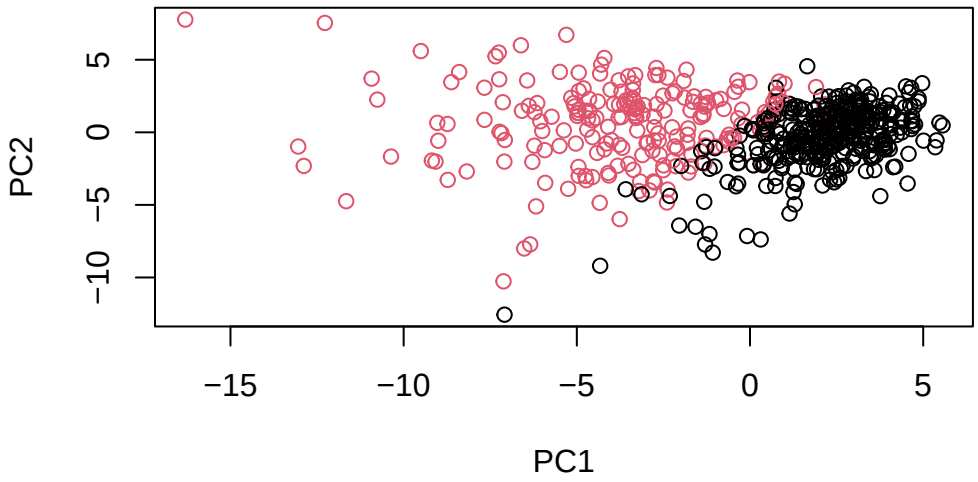


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

It is difficult to understand because it has lots of plots and numbers bunched up together and you aren't able to read anything or see the plots.

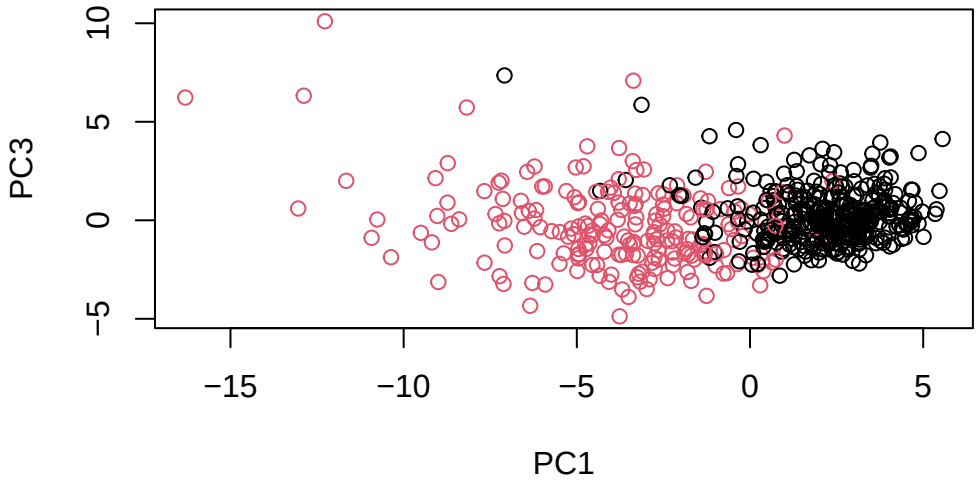
Generating a more standard scatter plot

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



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

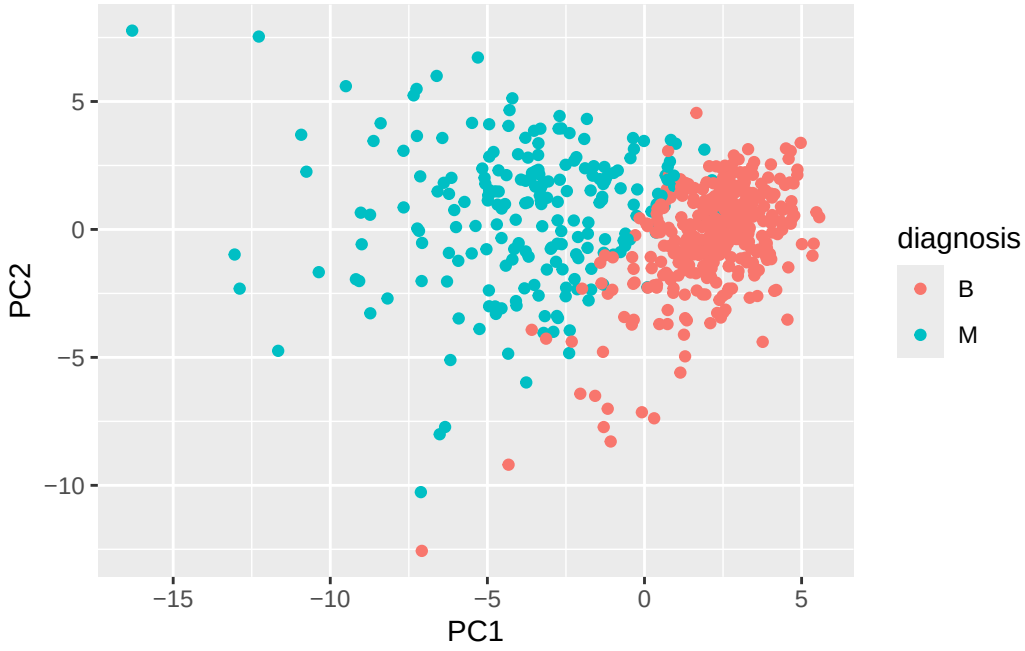
```
plot(wisc.pr$x[, c(1,3)],
     col = diagnosis,
     xlab = "PC1", ylab = "PC3")
```



The first plot is very different from the 2nd and 3rd plot where in the 1st plot everything was bunched up together and plots couldn't be seen. In plot 2 and 3 it explains more variance, and is capturing a separation between the red and black samples properly.

```
library(ggplot2)

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



PCA screeplot

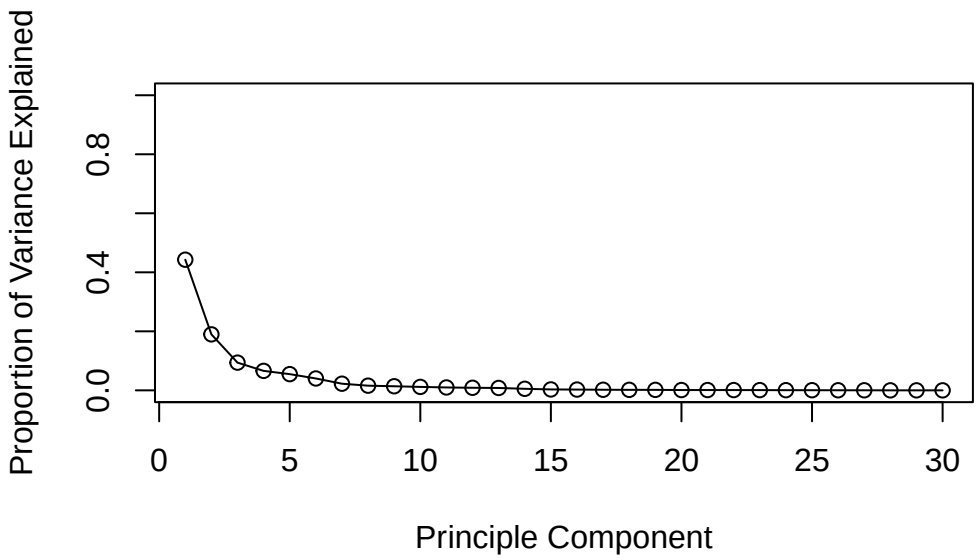
A plot of how much variance each PC captures. We can get this from `wisc.pr$dev` or from the output of `summary(wisc.pr)`

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

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

```
pc.var <- wisc.pr$sdev^2

pve <- pc.var / sum(pc.var)
  plot(pve, xlab = "Principle 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?

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

```
[1] -0.2608538
```

```
wisc.pr$rotation["concave.points_mean", "PC1"]
```

```
[1] -0.2608538
```

Q10. What is the minimum number of principal components required to explain 80% of the variance of the data?

The minimum number is 5 principal components to explain 80% of the variance data, because PC5 exceeds 80% threshold.

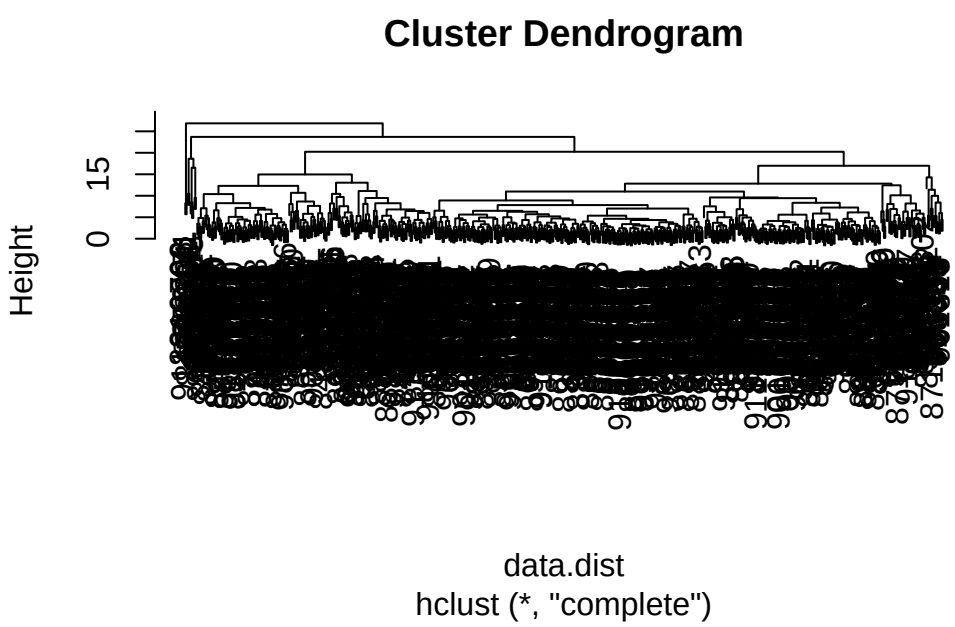
3. Hierarchical clustering

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

data.dist <- dist(data.scaled, method = "euclidean")

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

```
plot(wisc.hclust)
abline(h=150, col = "red", lty = 2)
```




```
h4 <- wisc.hclust$height[length(wisc.hclust$height) - 4 +1]
h4
```

```
[1] 18.63658
```

```
clusters_at_h <- cutree(wisc.hclust, h=18.63658)
length(unique(clusters_at_h))
```

```
[1] 5
```

5. Combining methods (PCA and CLustering)

Clustering the original data was not very productive. The PCA results looked promising. Here we combine these methods by clustering from our PCA results. In other words"clustering in PC space"...

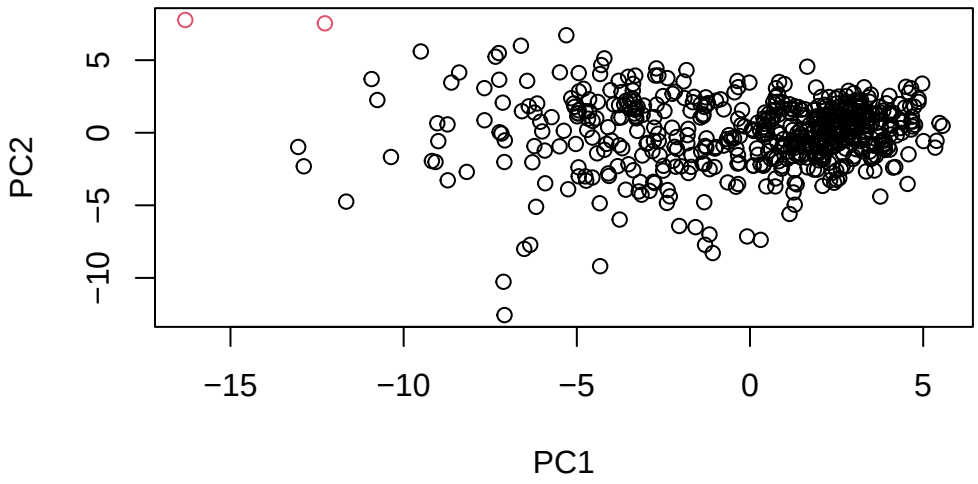
```
grps <- cutree(wisc.hclust, k = 2)
table(grps)
```

```
grps
  1   2
567   2
```

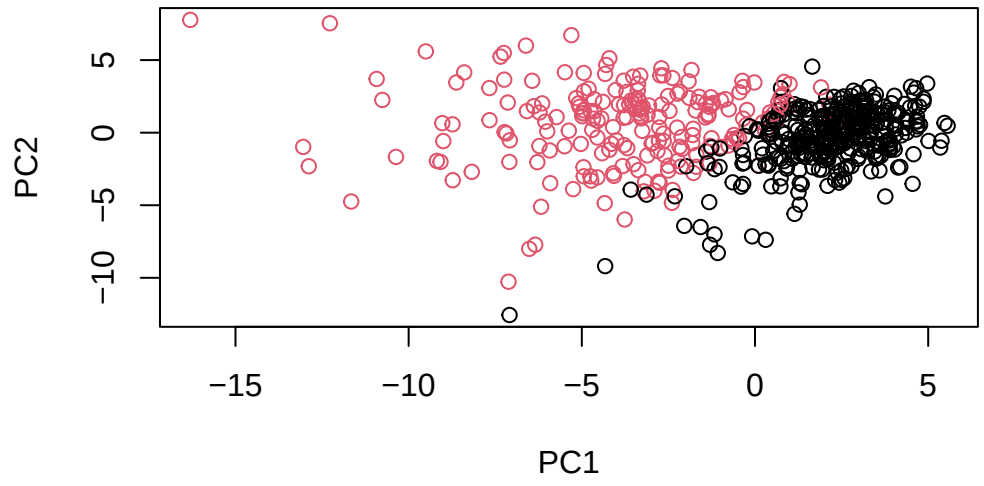
```
table(grps, diagnosis)
```

```
    diagnosis
grps   B   M
   1 357 210
   2   0   2
```

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



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




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

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

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

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

Q15. How well does the newly created model with four clusters separate out the two diagnoses?

```
dist7 <- dist(wisc.pr$x[, 1:7], method = "euclidean")

wisc.pr.hclust <- hclust(dist7, method = "ward.D2")

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

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

The model does a good job of separating the two diagnoses. The model captures most cancers while keeping false-positive alerts for healthy patients relatively low.

> Q16. How well do the k-means and hierarchical clustering models you created in previous sections 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.

```
wisc.df   <- read.csv("WisconsinCancer.csv", row.names = 1)
diagnosis <- factor(wisc.df$diagnosis)               # ensure it is a factor

wisc.data <- wisc.df[ , setdiff(names(wisc.df), "diagnosis")]

wisc.km <- kmeans(wisc.data, centers = 4)

dist.mat   <- dist(scale(wisc.data), method = "euclidean")
wisc.hclust <- hclust(dist.mat, method = "complete")
wisc.hclust.clusters <- cutree(wisc.hclust, k = 4)

table(wisc.km$cluster, diagnosis)
```

```
   diagnosis
      B   M
  1  94  87
  2   1 100
  3   0  19
  4 262   6
```

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

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

The k-means model places each patient into one of the 4 clusters and each cluster is made up of a single diagnosis, ex. cluster 1 contains only malignant, cluster 2 contains benign and so on. This model is excellent at avoiding false-positive alerts which is good when you want to avoid unnecessary follow-up for benign patients. The hierarchical clustering model merges the 2 groups that cause the smallest increase in within-cluster variance.

7. Prediction

```
#url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
npc
```

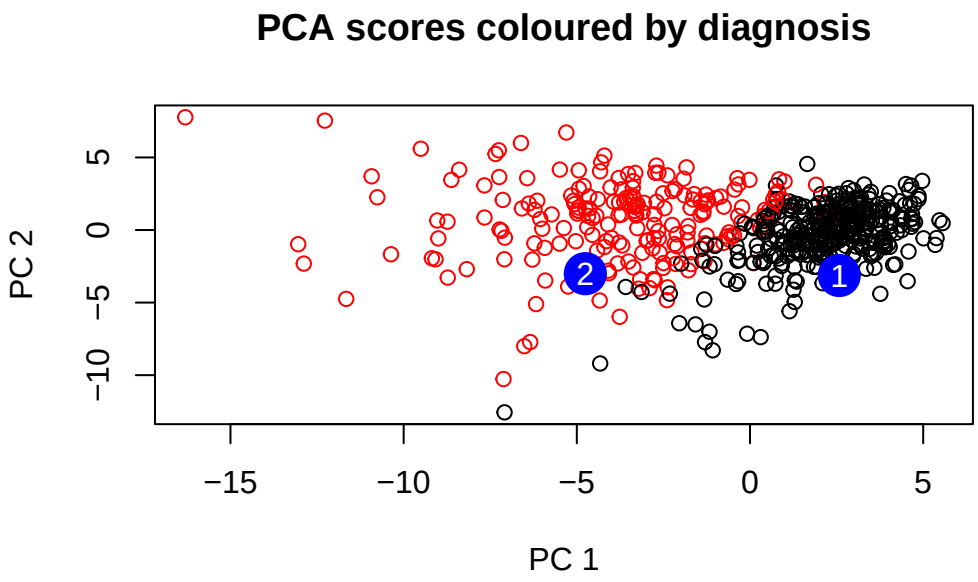
```
          PC1       PC2        PC3        PC4       PC5        PC6         PC7
[1,]  2.576616 -3.135913  1.3990492 -0.7631950  2.781648 -0.8150185 -0.3959098
[2,] -4.754928 -3.009033 -0.1660946 -0.6052952 -1.140698 -1.2189945  0.8193031
           PC8        PC9        PC10       PC11       PC12       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
```

```
col.vec <- ifelse(diagnosis == "M", "red", "black")

plot(wisc.pr$x[, 1:2],
     col = col.vec,
     xlab = "PC 1", ylab = "PC 2",
     main = "PCA scores coloured by diagnosis")

points(npc[, 1], npc[, 2], col = "blue", pch = 16, cex = 3)

text(npc[, 1], npc[, 2], labels = c(1, 2), col = "white")
```



Q18. Which of these new patients should we prioritize for follow up based on your results?

The patients that fall into the clusters that are dominated by malignant diagnoses should be the first ones to schedule for a follow up.