class08_mini_project

Ryan Bench (PID:A69038034)

Table of contents

Background 1
Data Import 2
Exploratory Data Analysis 2
Principal Component Analysis 3
Variance explained 7
Communicating PCA results 9
Hierarchical Clustering 10
Results of hierarchical clustering 10
Combining PCA and clustering 11
Selecting number of clusters 12
Using different methods 12
Prediction 18

Background

The goal of this mini-project is for you to explore a complete analysis using the unsupervised learning techniques covered in class. You'll extend what you've learned by combining PCA as a preprocessing step to clustering using data that consist of measurements of cell nuclei of human breast masses. This expands on our RNA-Seq analysis from last day.

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

We will use the read.csv() function to import our data

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

Make sure I do not include sample ID or diagnosis columns in the data that we analyze below

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

```
[1] 569  30
```

Exploratory Data Analysis

Q1. How many observations are in this dataset?

There are 569 observations/samples/patients in the data set.

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

```
sum(wisc.df$diagnosis == "M")
```

```
[1] 212
```

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

```
  B   M
357 212
```

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

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

```
[1] 10
```

Principal Component Analysis

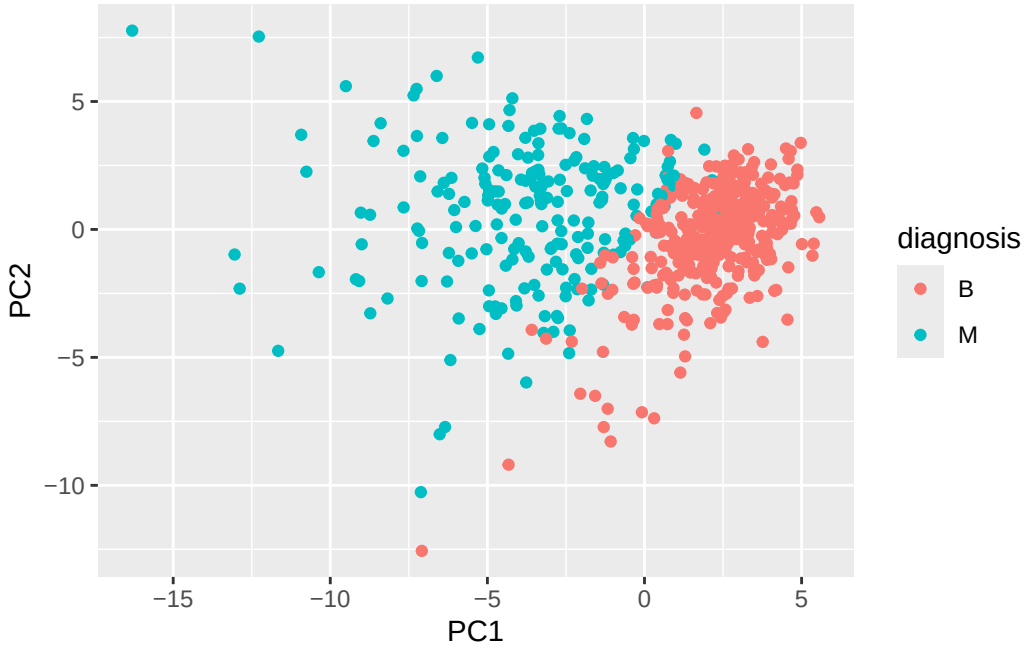
The main function in base R for PCA is called `prcomp()` Almost always want to scale the data by setting `scale=TRUE`

```
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
```

Let's make our main result figure - the "PC Plot" or "score plot", "orientaion plot"

```
library(ggplot2)
ggplot(wisc.pr$x, aes(x=PC1, y=PC2, col = diagnosis)) + geom_point()
```



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

The proportion of variance captured by the first principal components is 0.4427, or about 44%.

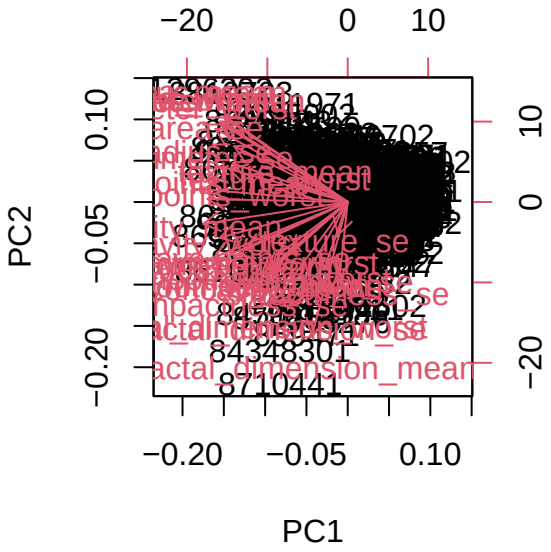
Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data?

At least 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?

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

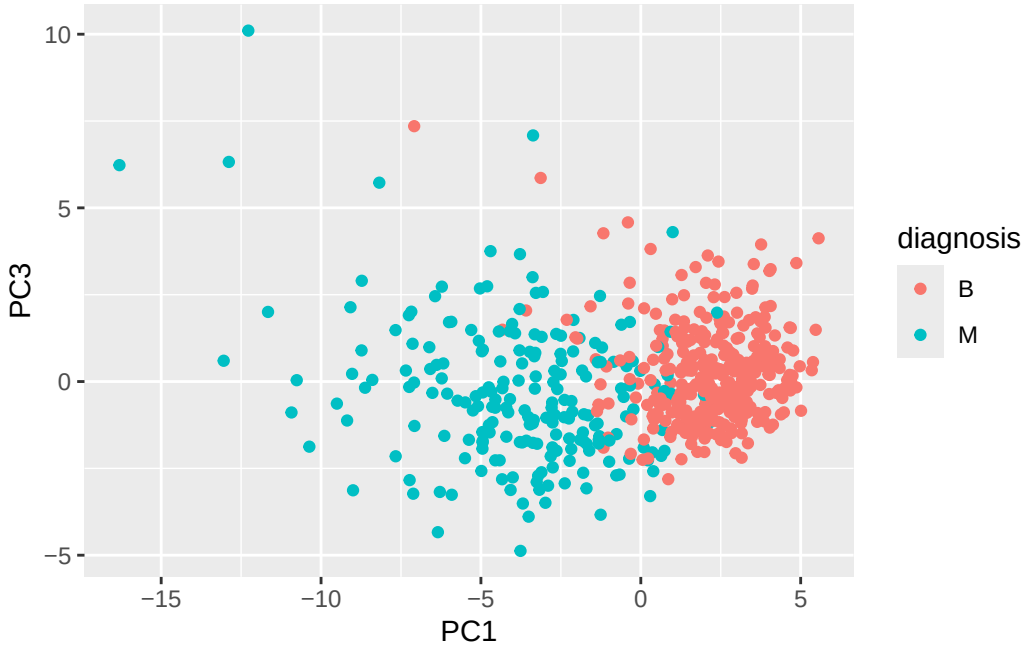
```
biplot(wisc.pr)
```



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

This plot is difficult to understand. It has a large cluster of points and labels that makes it impossible to differentiate most points.

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



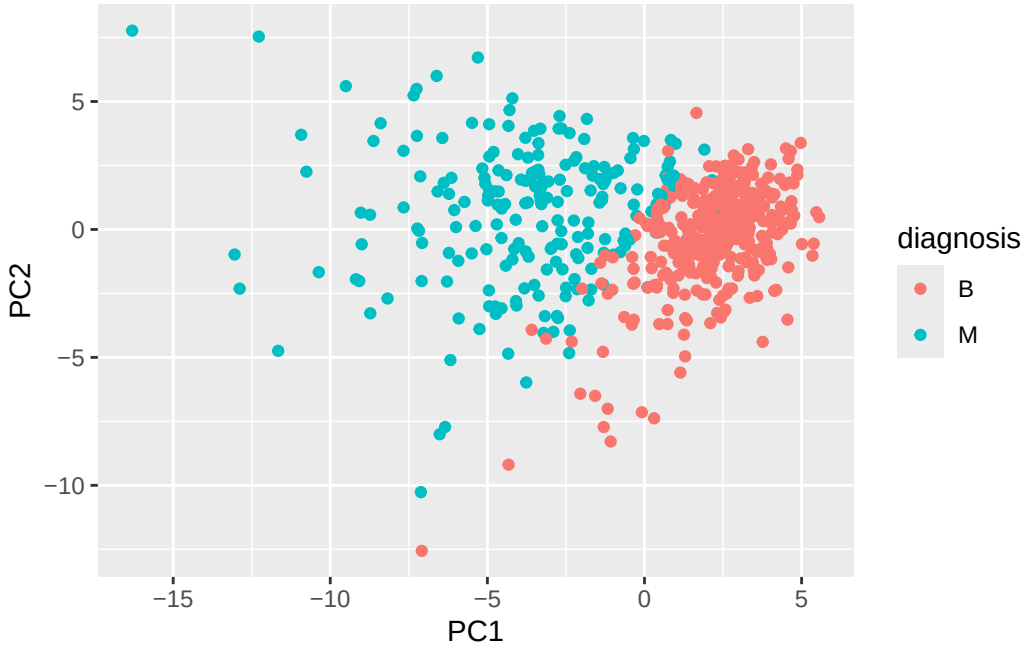
Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

I notice that this plot has a similar pattern to the first one. A benign diagnosis is farther to the right on this plot, and a malignant diagnosis is farther to the center or left in comparison. The data points also appear to be more dense as compared to the first plot.

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

library(ggplot2)

ggplot(df) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```



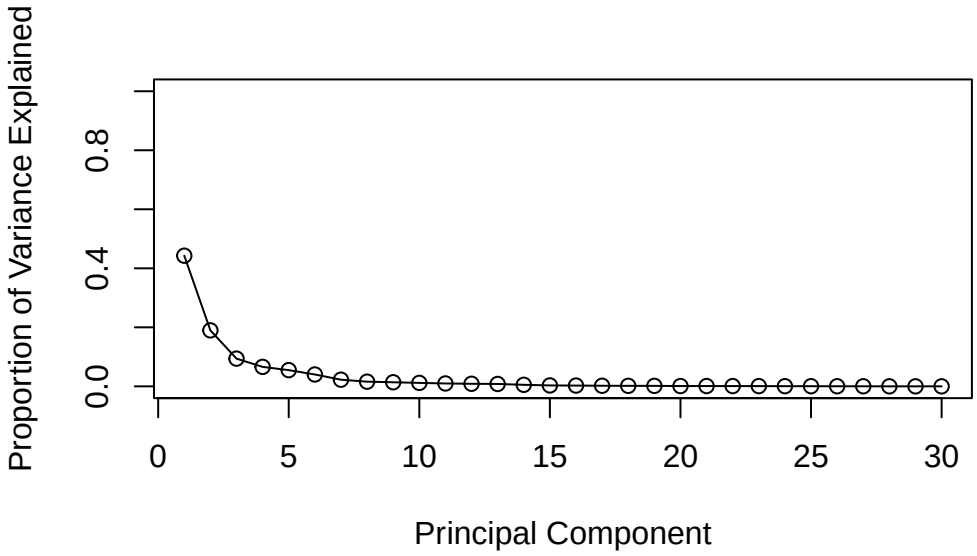
Variance explained

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

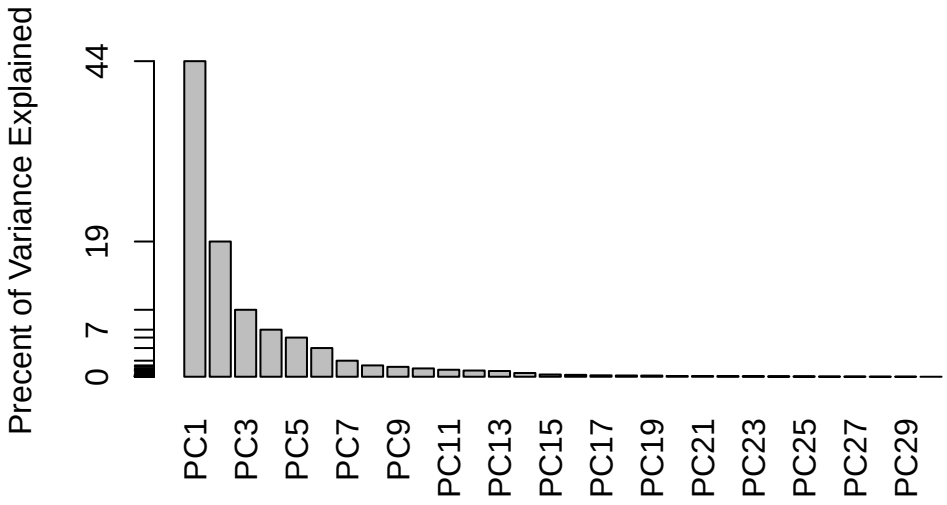
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

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

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




```
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 )
```



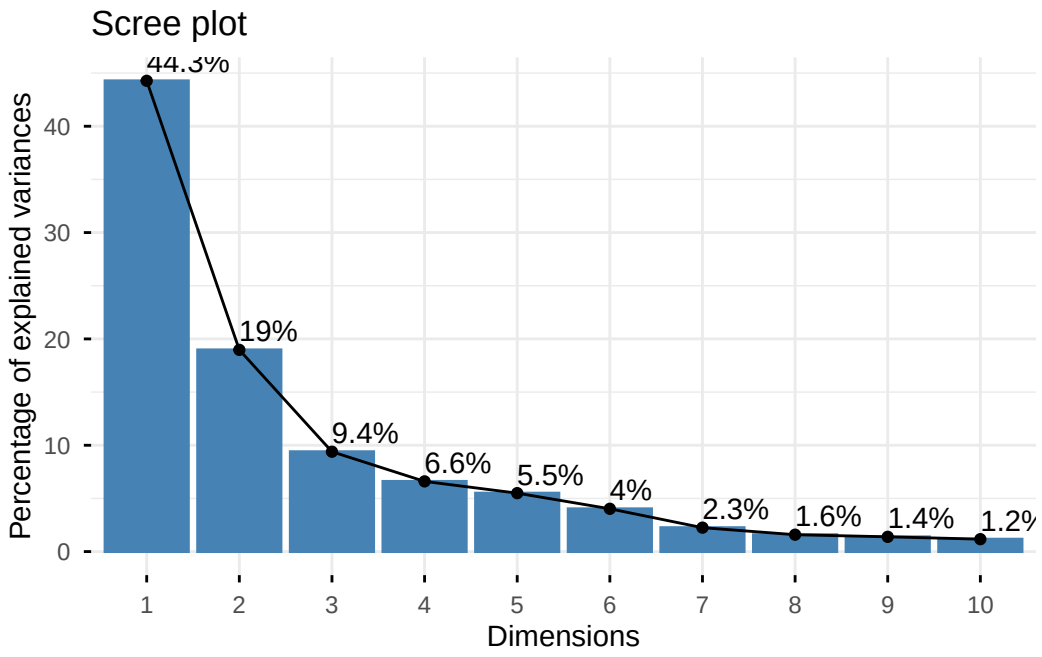
```
library(factoextra)
```

```
Warning: package 'factoextra' was built under R version 4.3.3
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(wisc.pr, addlabels = TRUE)
```

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```



Communicating PCA results

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

```
[1] -0.2608538
```

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.

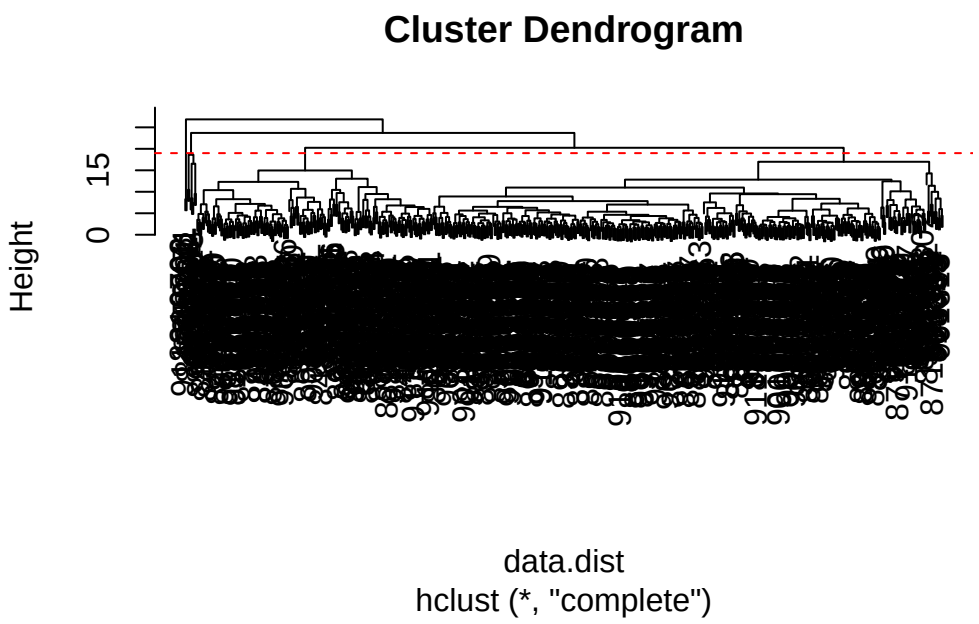
The value is -0.2608538, meaning there is a negative contribution to the first PC. Lower values will increase the score of PC1.

Hierarchical Clustering

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

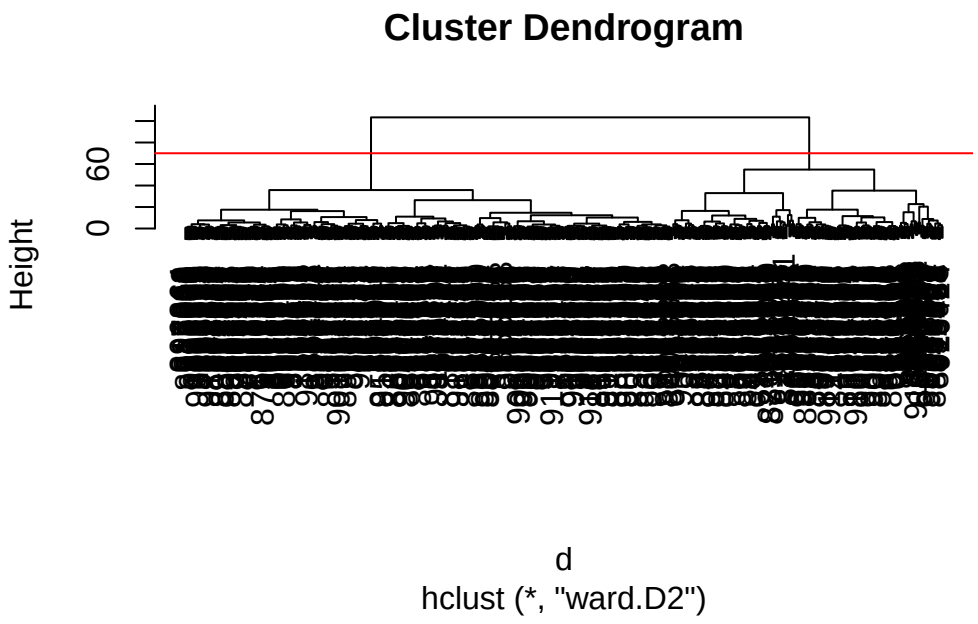
Results of hierarchical clustering

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



Combining PCA and clustering

```
d <- dist( wisc.pr$x[,1:3])
wisc.pr.hclust <- hclust(d, method = "ward.D2")
plot(wisc.pr.hclust)
abline(h=70, col = "red")
```



Get my cluster membership vector

```
grps <- cutree(wisc.pr.hclust, h=70)
table(grps)
```

```
grps
  1   2
203 366
```

```
table(diagnosis)
```

```
diagnosis
  B   M
357 212
```

Make a wee "cross-table"

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

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

Cluster 2 is indicative of benign, and cluster 1 is indicative of malignant TP: 179 FP: 24

Sensitivity: TP/(TP+FN)

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

At a height of 19, this is where the clustering model has 4 clusters.

Selecting number of clusters

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

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

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

> Q11. OPTIONAL: Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10? How do you judge the quality of your result in each case?

Using different methods

```
wisc.hclust.complete <- hclust(data.dist, method = "complete")
wisc.hclust.average  <- hclust(data.dist, method = "average")
wisc.hclust.single   <- hclust(data.dist, method = "single")
wisc.hclust.ward     <- hclust(data.dist, method = "ward.D2")
clusters.complete <- cutree(wisc.hclust.complete, k = 2)
clusters.average  <- cutree(wisc.hclust.average, k = 2)
clusters.single   <- cutree(wisc.hclust.single, k = 2)
clusters.ward     <- cutree(wisc.hclust.ward, k = 2)

table(clusters.complete, diagnosis)
```

```
                 diagnosis
clusters.complete   B   M
                1 357 210
                2   0   2
```

```
table(clusters.average, diagnosis)
```

```
                diagnosis
clusters.average   B   M
               1 357 209
               2   0   3
```

```
table(clusters.single, diagnosis)
```

```
               diagnosis
clusters.single   B   M
              1 357 210
              2   0   2
```

```
table(clusters.ward, diagnosis)
```

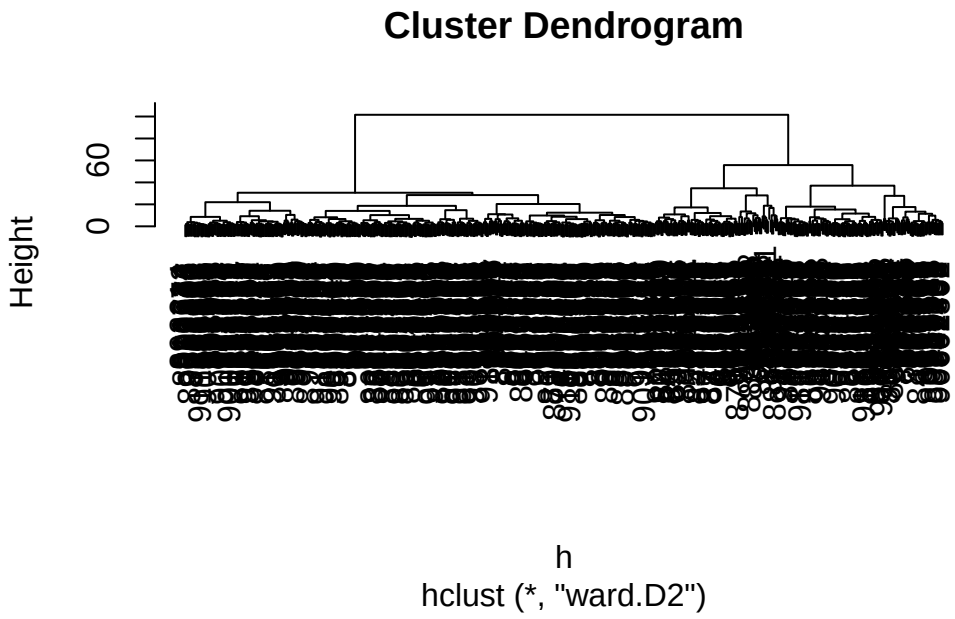
```
             diagnosis
clusters.ward   B   M
            1  20 164
            2 337  48
```

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

The ward.D2 method produces my favorite data set. I like this method the most because it separates out the benign and malignant cases the best. The other methods assign most samples to cluster 1, which fails to distinguish between diagnoses.

##Combining methods

```
h <- dist( wisc.pr$x[,1:7])
wisc.pr.hclust2 <- hclust(h, method = "ward.D2")
plot(wisc.pr.hclust2)
```



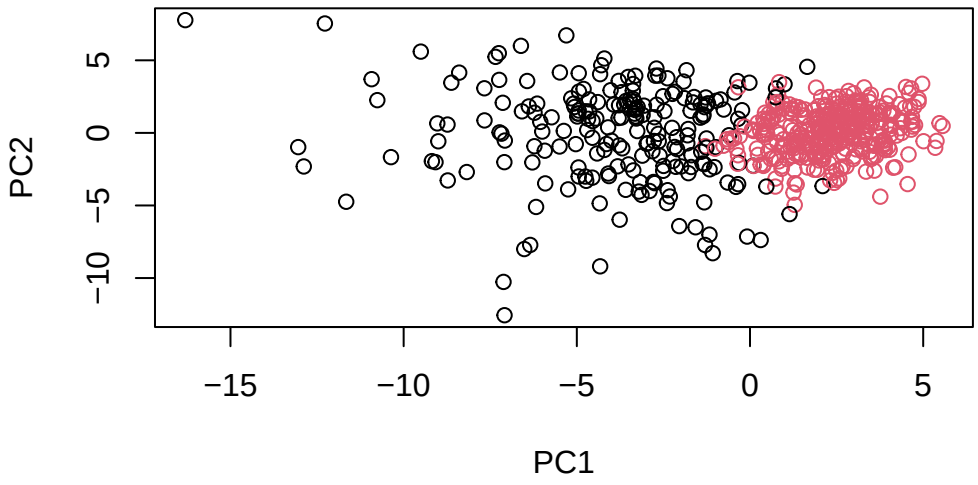

```
grps <- cutree(wisc.pr.hclust2, k=2)
table(grps)
```

```
grps
  1   2
216 353
```

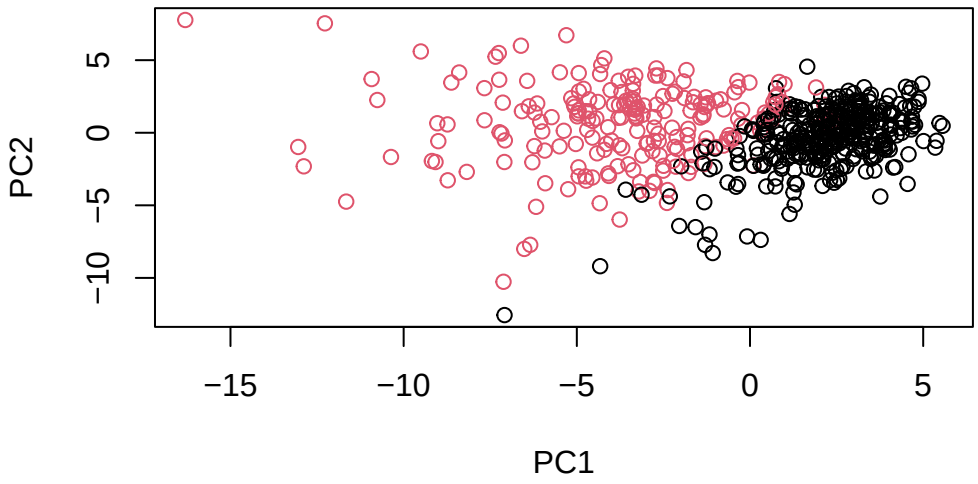
```
table(grps, diagnosis)
```

```
    diagnosis
grps   B   M
   1  28 188
   2 329  24
```

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




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



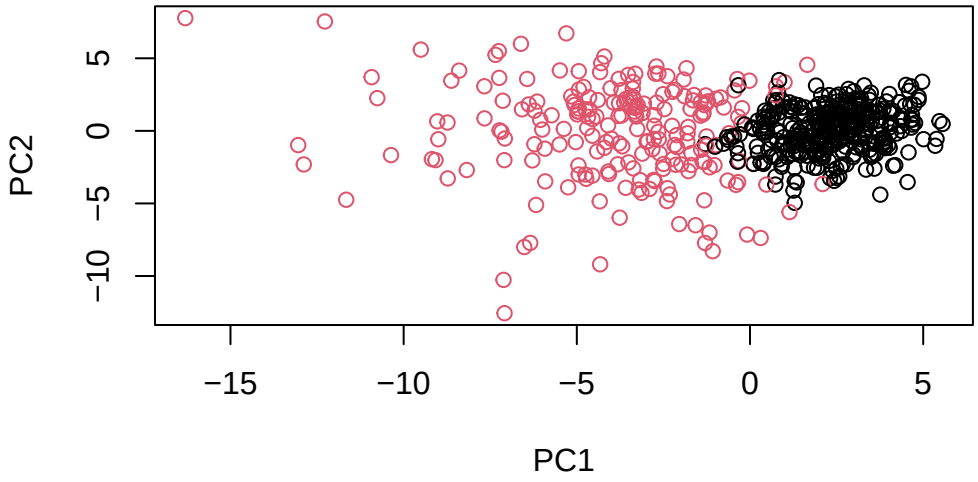
```
g <- as.factor(grps)
levels(g)
```

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

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

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

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




```
library(rgl)
```

```
Warning: package 'rgl' was built under R version 4.3.3
```

```
plot3d(wisc.pr$x[,1:3], xlab="PC 1", ylab="PC 2", zlab="PC 3", cex=1.5, size=1, type="s", col
```

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

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

It separates them out fairly well. Cluster 1 is associated with mostly malignant cases, and cluster 2 is mostly associated with benign cases. However there are a similar number of misclassifications in both.

```
wisc.km <- kmeans(data.scaled, centers = 2)
table(wisc.hclust.clusters, diagnosis)
```

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

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

```
   diagnosis
      B   M
  1  16 177
  2 341  35
```

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.

In terms of separating diagnosis, the hierarchical clustering methods can separate diagnoses reasonably well. There are clusters 2 and 4 though, that do not separate diagnoses very well. The k-means clustering separates the diagnosis more cleanly.

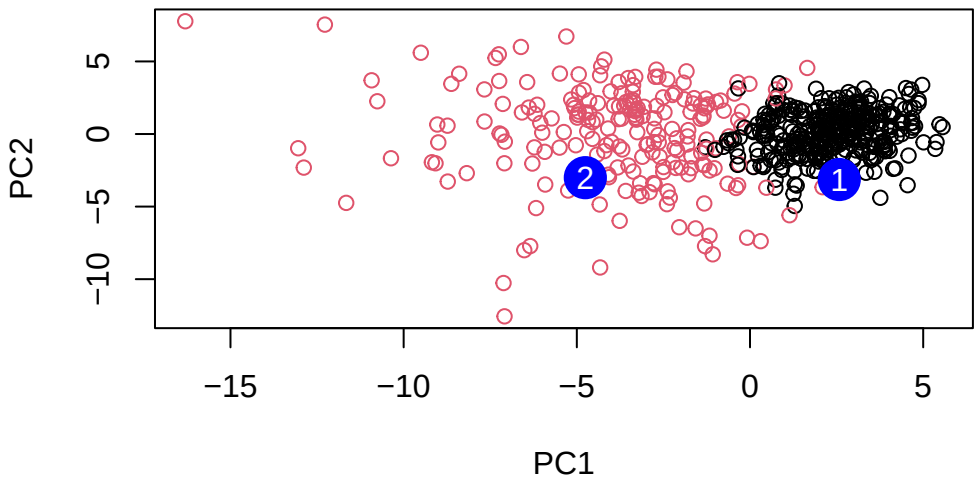
> Q15. OPTIONAL: Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

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 off our previous work, patient 2 should be prioritized for follow up.