class09

Jibin (PID: A53300326)

2021/10/27

Exploratory data analysis

Use the read.csv() function to read the CSV (comma-separated values) file containing the data (available from our class website: WisconsinCancer.csv)

```
fna.data <-read.csv("https://bioboot.github.io/bimm143_S20/class-material/WisconsinCancer.csv")
```

Complete the following code to input the data and store as wisc.df

```
wisc.df <- data.frame (fna.data, row.names=1)
```

We can use -1 here to remove the first column

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

Create diagnosis vector for later

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

```
##   [1] M M M M M M M M M M M M M M M M M M M B B B M M M M M M M M M M M M M M M
##  [38] B M M M M M M M M B M B B B B B M M B M M B B B B M B M M B B B B M B M M
##  [75] B M B M M B B B M M B M M M B B B M B B M M B B B M M B B B B M B B M B B
## [112] B B B B B B M M M B M M B B B M M B M B M M B M M B B M B B M B B B B M B
## [149] B B B B B B B B M B B B B M M B M B B M M B B M M B B B B M B B M M M B M
## [186] B M B B B M B B M M B M M M M B M M M B M B M B B M B M M M M B B M M B B
## [223] B M B B B B B M M B B M B B M M B M B B B B M B B B B B M B M M M M M M M
## [260] M M M M M M M B B B B B B M B M B B M B B M B M M B B B B B B B B B B B B
## [297] B M B B M B M B B B B B B B B B B B B B B M B B B M B M B B B B M M M B B
## [334] B B M B M B M B B B M B B B B B B B M M M B B B B B B B B B B B M M B M M
## [371] M B M M B B B B B M B B B B B M B B B M B B M M B B B B B B M B B B B B B
## [408] B M B B B B B M B B M B B B B B B B B B B B B M B M M B M B B B B B M B B
## [445] M B M B B M B M B B B B B B B B M M B B B B B B M B B B B B B B B B B M B
## [482] B B B B B B M B M B B M B B B B B M M B M B M B B B B B M B B M B M B M M
## [519] B B B M B B B B B B B B B B M B M M B B B B B B B B B B B B B B B B B B B
## [556] B B B B B B B M M M M M M B
## Levels: B M
```

Q1. How many observations are in this dataset?

```
dim (wisc.data)
```

```
## [1] 569  30
```

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

```
table(diagnosis)
```

```
## diagnosis
##   B   M
## 357 212
```

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

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

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Principal Component Analysis

Performing PCA

Check column means and standard deviations

```
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
```

Perform PCA on wisc.data by completing the following code

```
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)?

44.27%

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

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

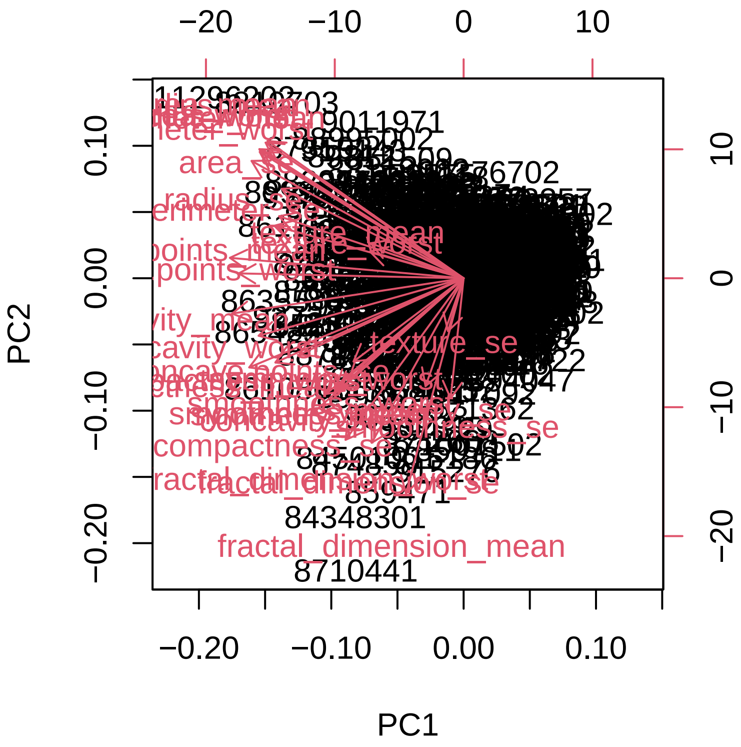
> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

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

Interpreting PCA results

Create a biplot of the wisc.pr using the biplot() function.

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

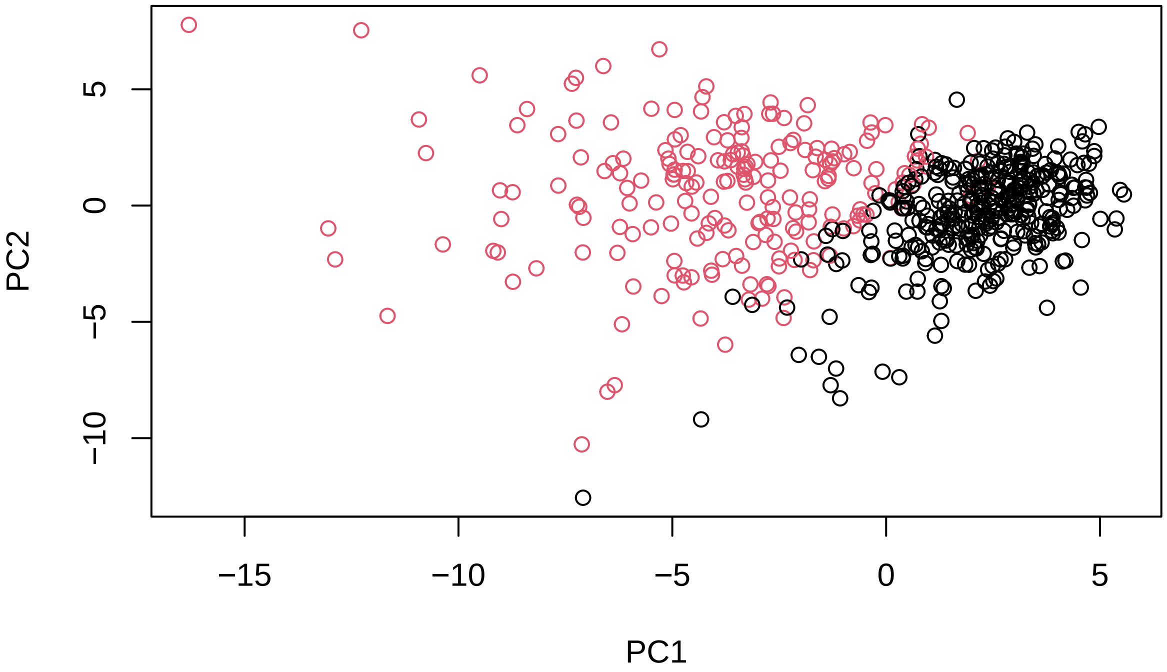


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

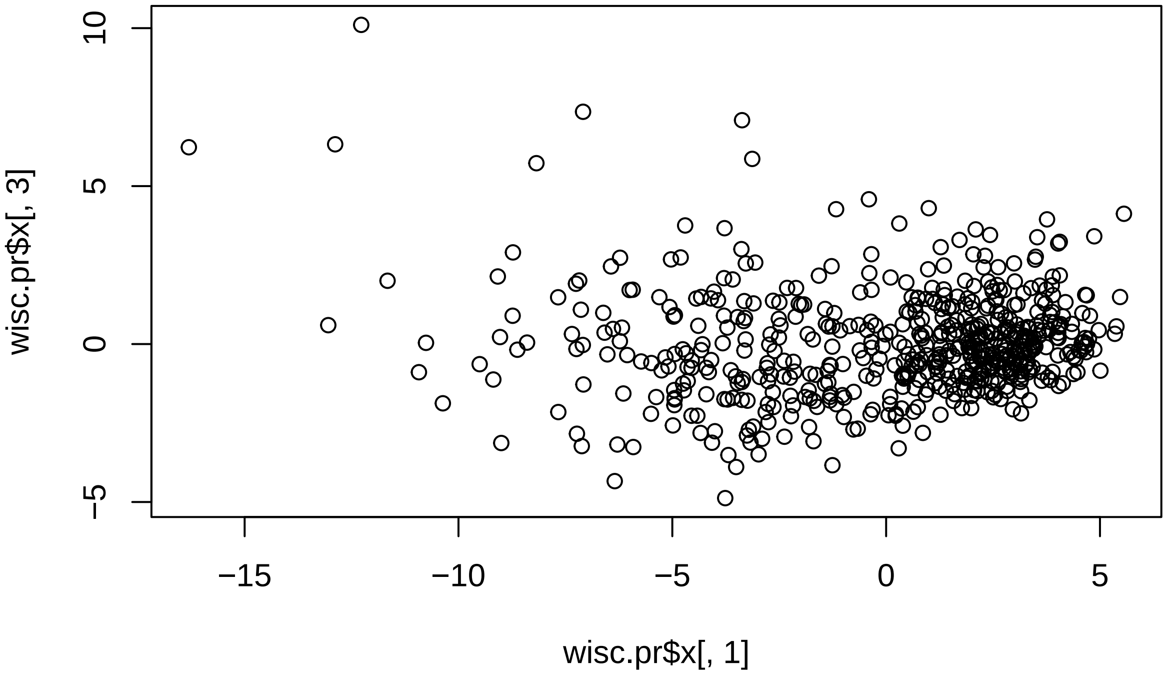
This is a hot mess of a plot and it's diffcult to interpret.

Scatter plot observations by components 1 and 2

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



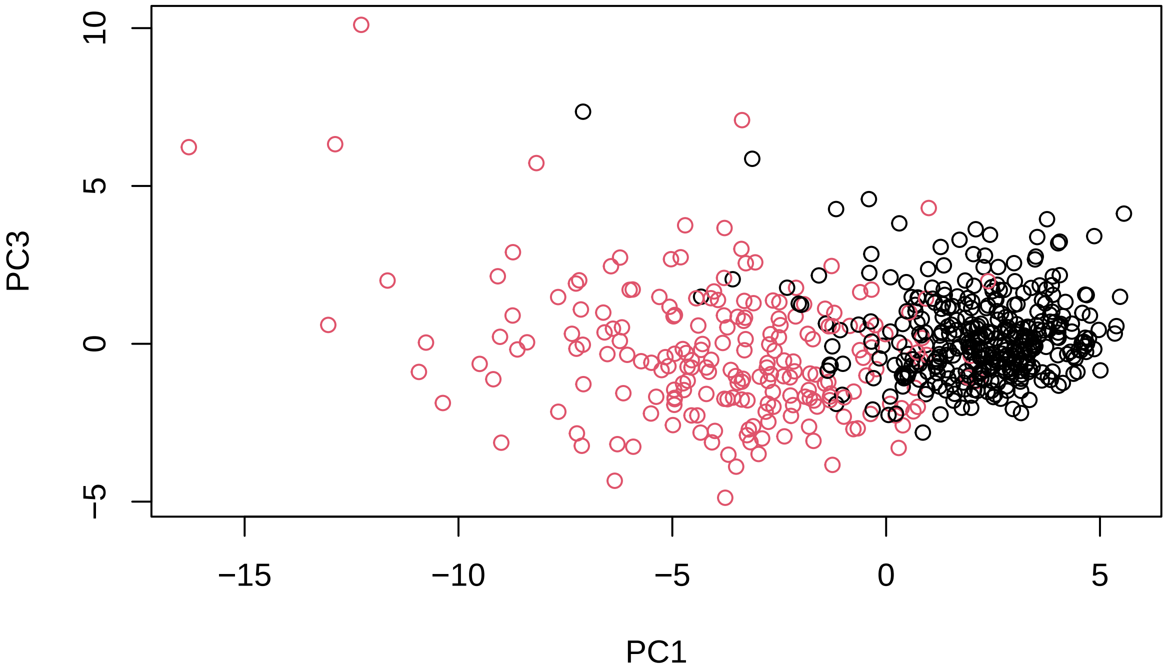
```
plot(wisc.pr$x [,1], wisc.pr$x [,3])
```



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

Repeat for components 1 and 3

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



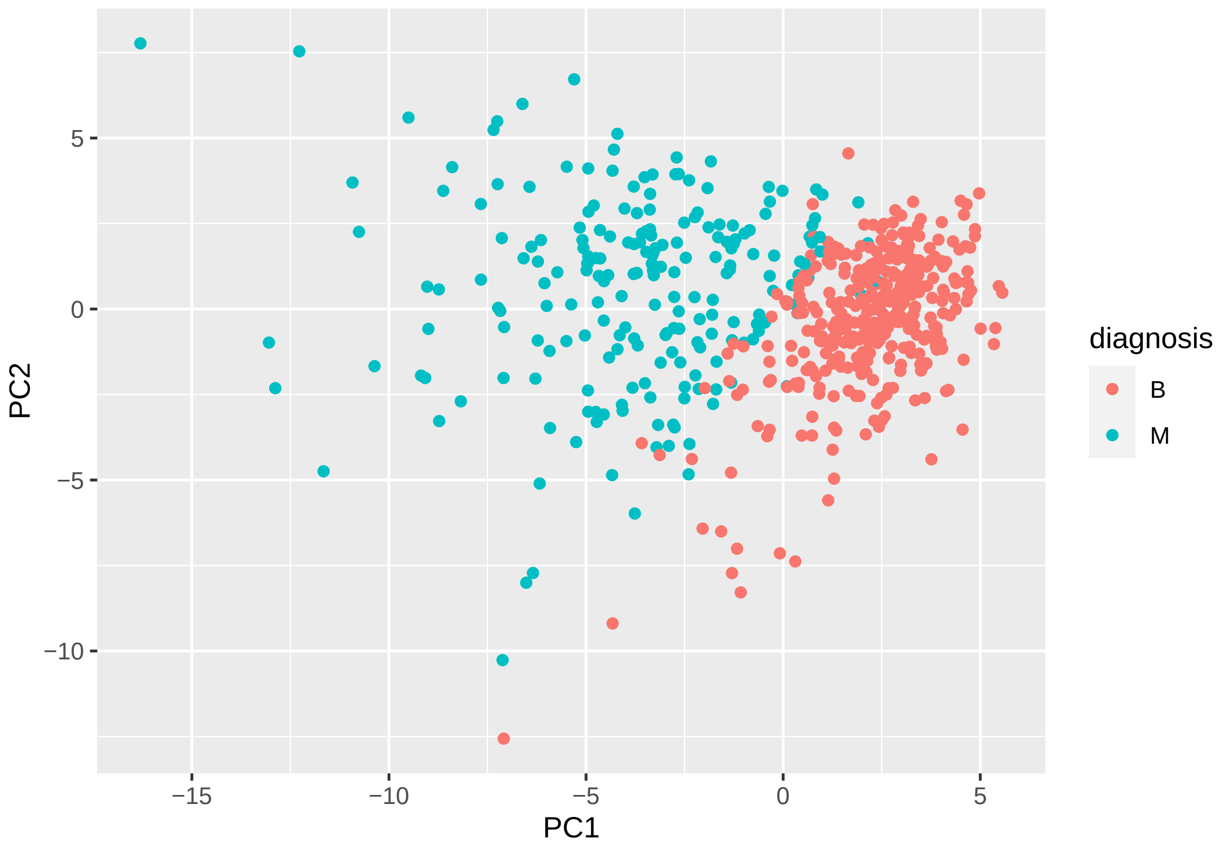
The first plot (PC2 vs PC1) has a cleaner cut separating the two subgroups than this polt, since PC 2 explains more variance in the original data than PC 3.

Create a data.frame for ggplot

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

Make a scatter plot colored by diagnosis

```
library("ggplot2")
ggplot(df) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```



Variance explained

Calculate variance of each component

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

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

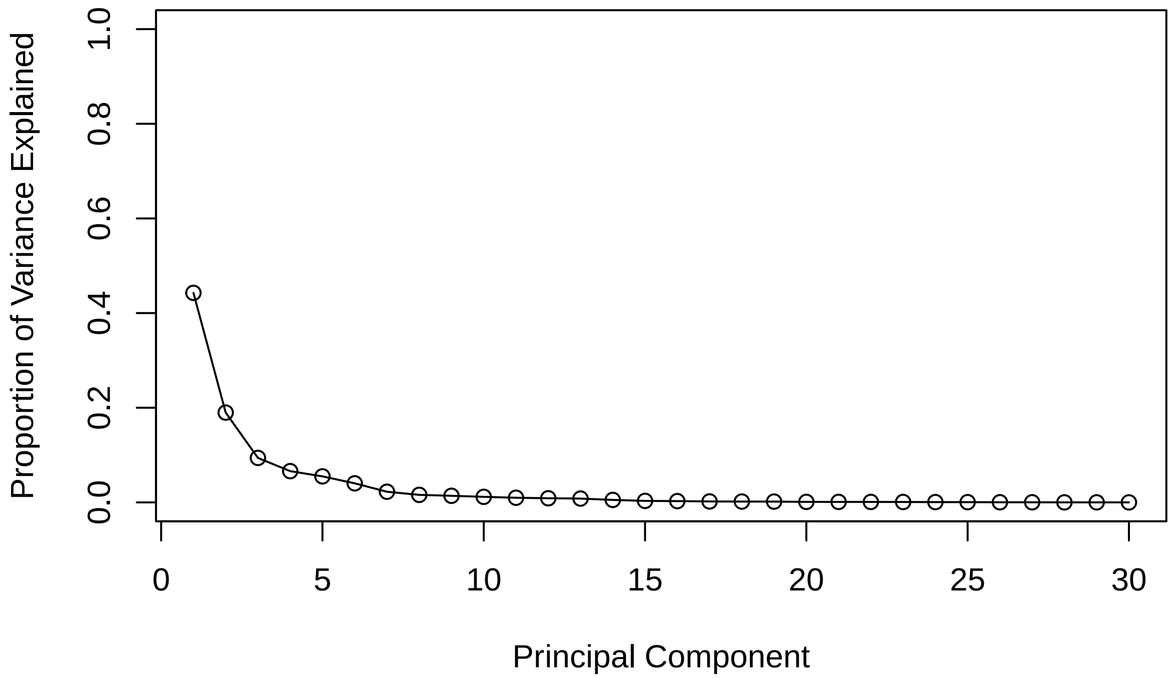
Variance explained by each principal component: pve

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

```
##  [1] 4.427203e-01 1.897118e-01 9.393163e-02 6.602135e-02 5.495768e-02
##  [6] 4.024522e-02 2.250734e-02 1.588724e-02 1.389649e-02 1.168978e-02
## [11] 9.797190e-03 8.705379e-03 8.045250e-03 5.233657e-03 3.137832e-03
## [16] 2.662093e-03 1.979968e-03 1.753959e-03 1.649253e-03 1.038647e-03
## [21] 9.990965e-04 9.146468e-04 8.113613e-04 6.018336e-04 5.160424e-04
## [26] 2.725880e-04 2.300155e-04 5.297793e-05 2.496010e-05 4.434827e-06
```

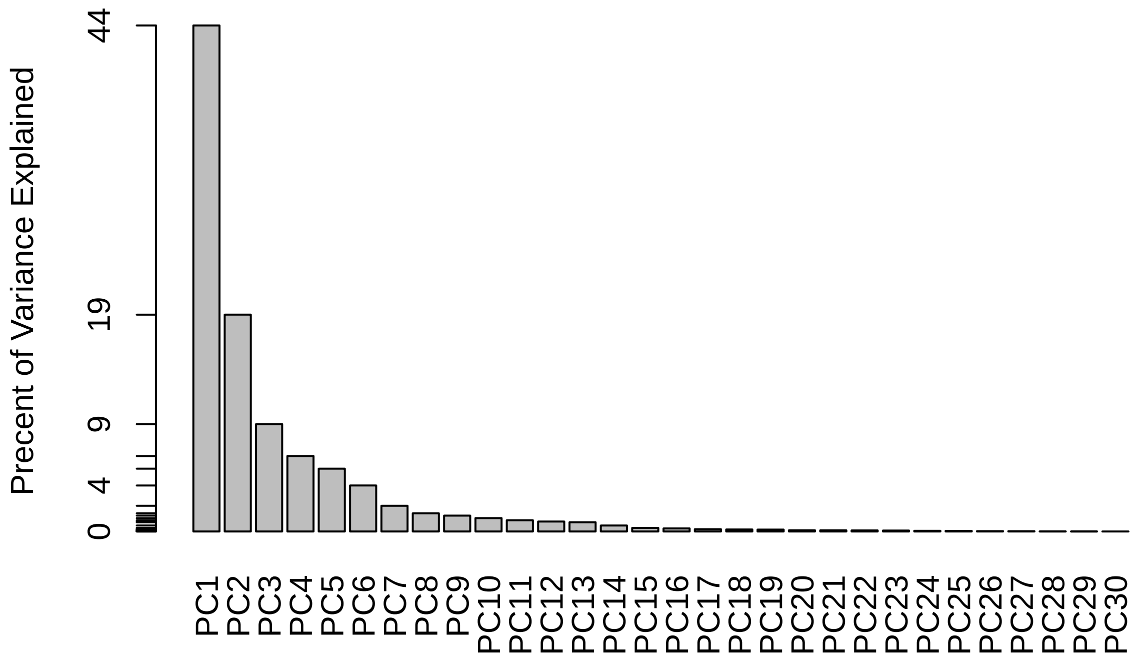
Plot variance explained for each principal component

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



Alternative scree plot of the same data, note data driven y-axis

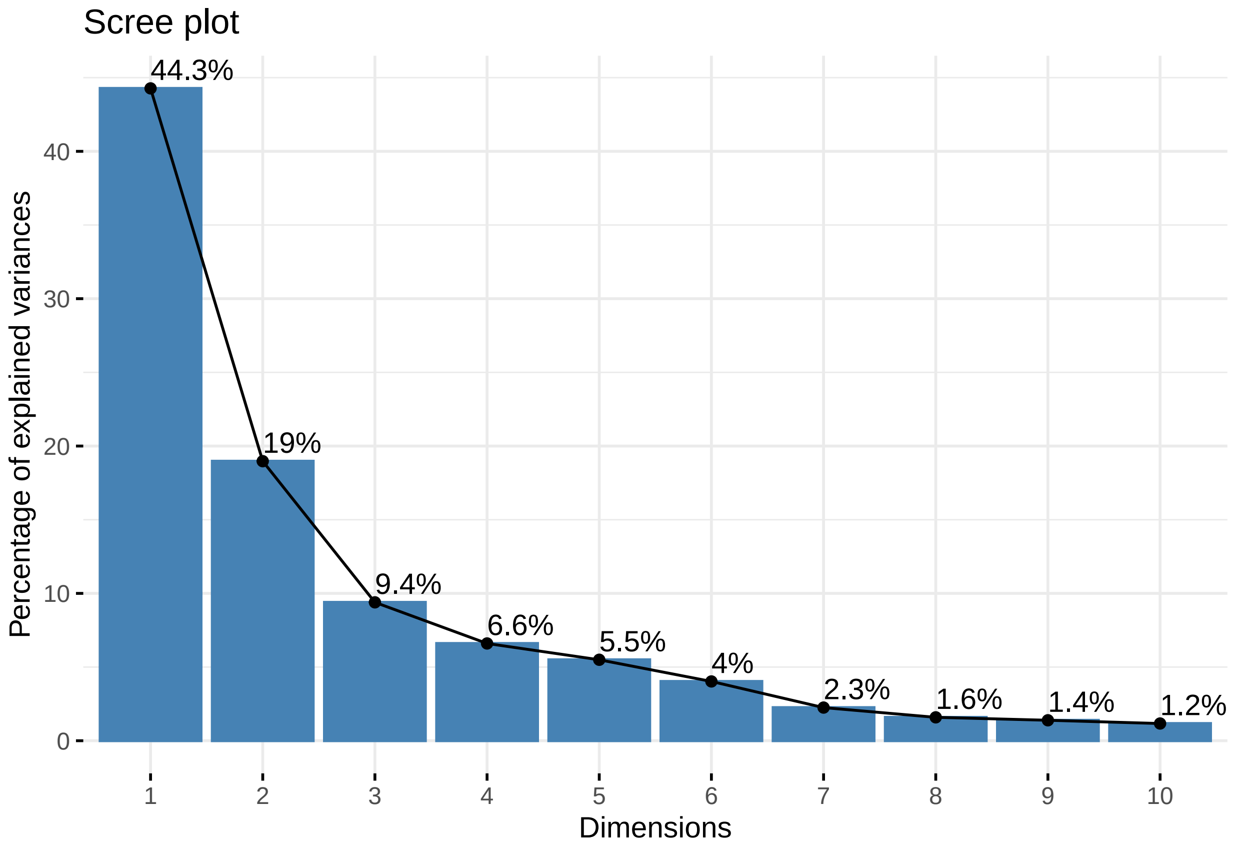
```
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```



ggplot based graph

#install.packages("factoextra")

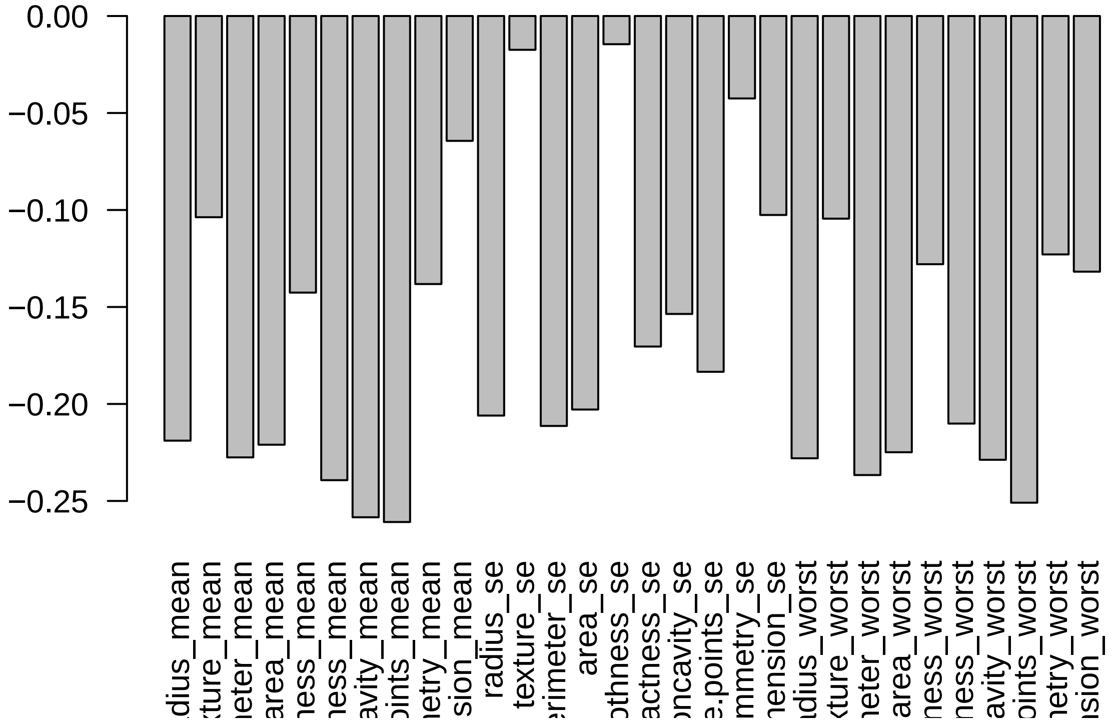
```
library(factoextra)
fviz_eig(wisc.pr, addlabels = TRUE)
```



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?

```
barplot(wisc.pr$rotation[,1], las=2, mar=c(10, 3, 3, 20))
```



### concave.points_mean

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

```
var <- summary(wisc.pr)
var$importance
```

```
##                              PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation     3.644394 2.385656 1.678675 1.407352 1.284029 1.098798
## Proportion of Variance 0.442720 0.189710 0.093930 0.066020 0.054960 0.040250
## Cumulative Proportion  0.442720 0.632430 0.726360 0.792390 0.847340 0.887590
##                              PC7       PC8       PC9      PC10      PC11
## Standard deviation     0.8217178 0.6903746 0.6456739 0.5921938 0.5421399
## Proportion of Variance 0.0225100 0.0158900 0.0139000 0.0116900 0.0098000
## Cumulative Proportion  0.9101000 0.9259800 0.9398800 0.9515700 0.9613700
##                             PC12      PC13      PC14      PC15      PC16
## Standard deviation     0.5110395 0.4912815 0.3962445 0.3068142 0.2826001
## Proportion of Variance 0.0087100 0.0080500 0.0052300 0.0031400 0.0026600
## Cumulative Proportion  0.9700700 0.9781200 0.9833500 0.9864900 0.9891500
##                             PC17      PC18      PC19      PC20      PC21
## Standard deviation     0.2437192 0.2293878 0.2224356 0.1765203 0.1731268
## Proportion of Variance 0.0019800 0.0017500 0.0016500 0.0010400 0.0010000
## Cumulative Proportion  0.9911300 0.9928800 0.9945300 0.9955700 0.9965700
##                             PC22      PC23      PC24      PC25      PC26
## Standard deviation     0.1656484 0.1560155 0.1343689 0.1244238 0.0904303
## Proportion of Variance 0.0009100 0.0008100 0.0006000 0.0005200 0.0002700
## Cumulative Proportion  0.9974900 0.9983000 0.9989000 0.9994200 0.9996900
##                              PC27      PC28       PC29       PC30
## Standard deviation     0.08306903 0.0398665 0.02736427 0.01153451
## Proportion of Variance 0.00023000 0.0000500 0.00002000 0.00000000
## Cumulative Proportion  0.99992000 0.9999700 1.00000000 1.00000000
```

```
sum(var$importance[3,] <0.8)
```

```
## [1] 4
```

So, four at least.

Hierarchical clustering

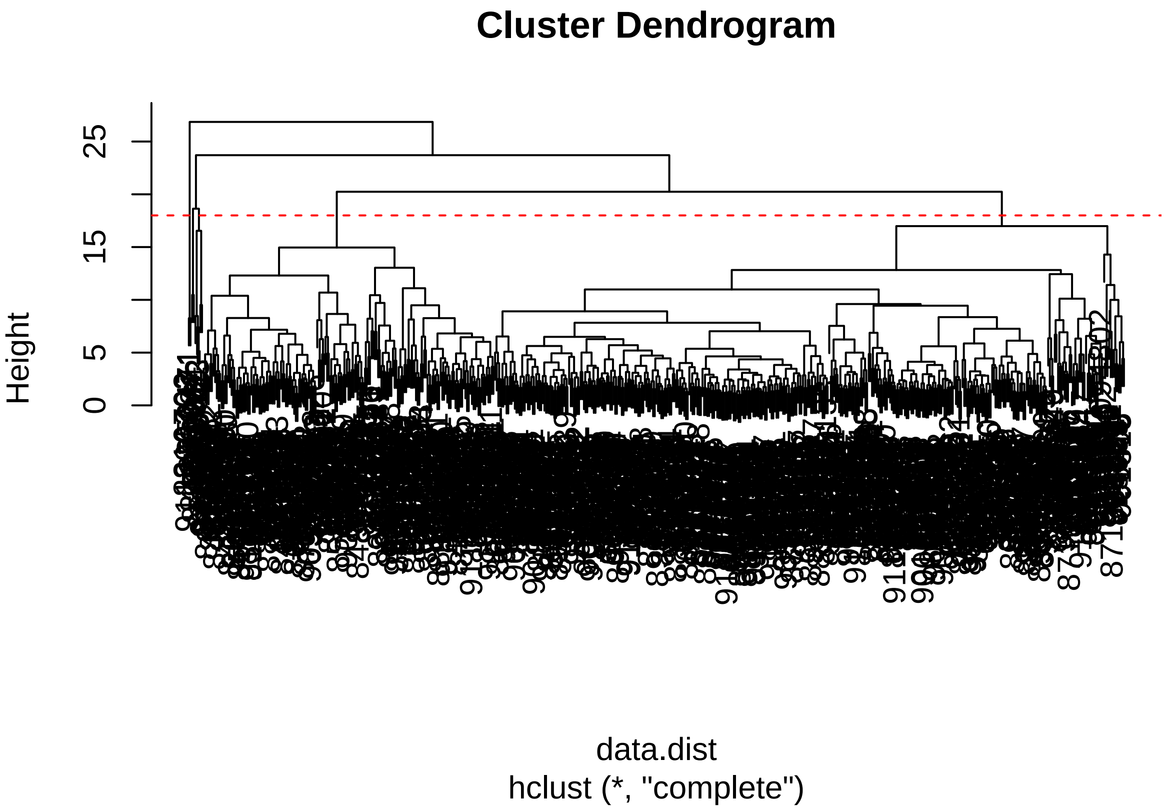
Scale the wisc.data data using the "scale()" function???

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

Results of hierarchical clustering

> Q11. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?

```
plot(wisc.hclust)
abline(wisc.hclust, h=18, col="red", lty=2)
```



Selecting number of clusters

Use cutree() to cut the tree so that it has 4 clusters. Assign the output to the variable wisc.hclust.clusters.

```
wisc.hclust.clusters <- cutree(wisc.hclust, 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. Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10?

Perhaps no.

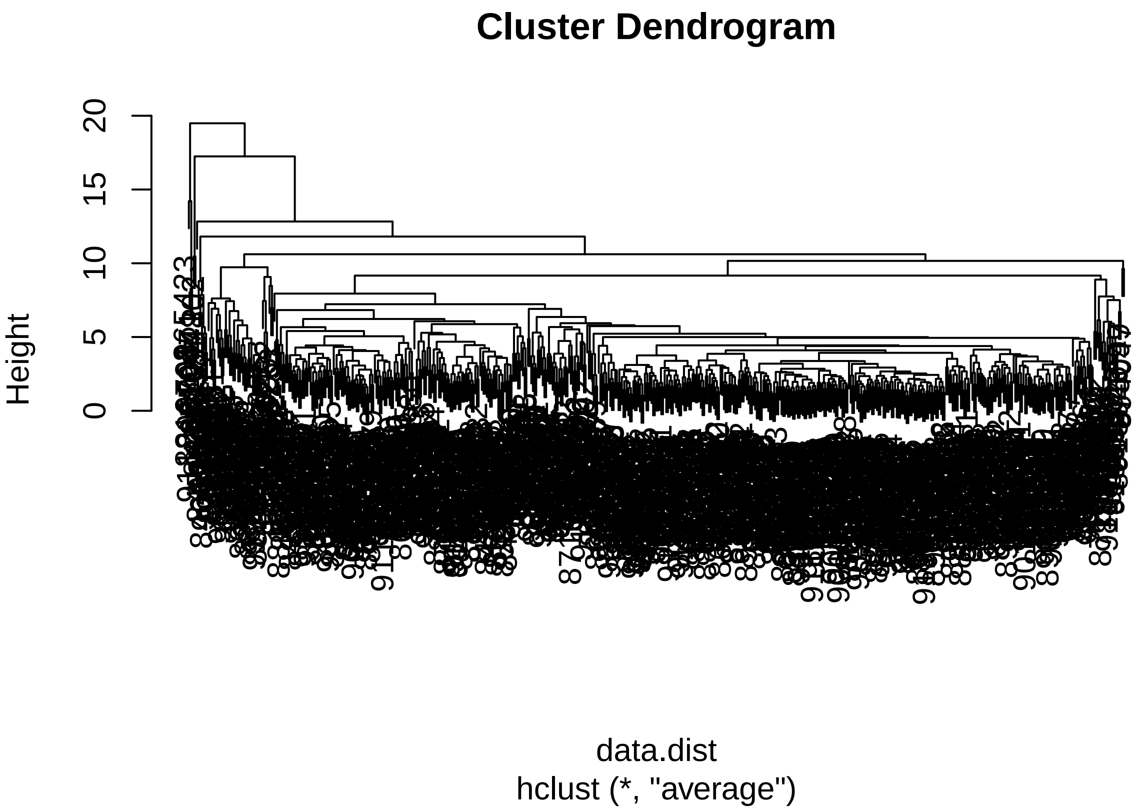
```
wisc.hclust.clusters <- cutree(wisc.hclust, 5)
table(wisc.hclust.clusters, diagnosis)
```

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

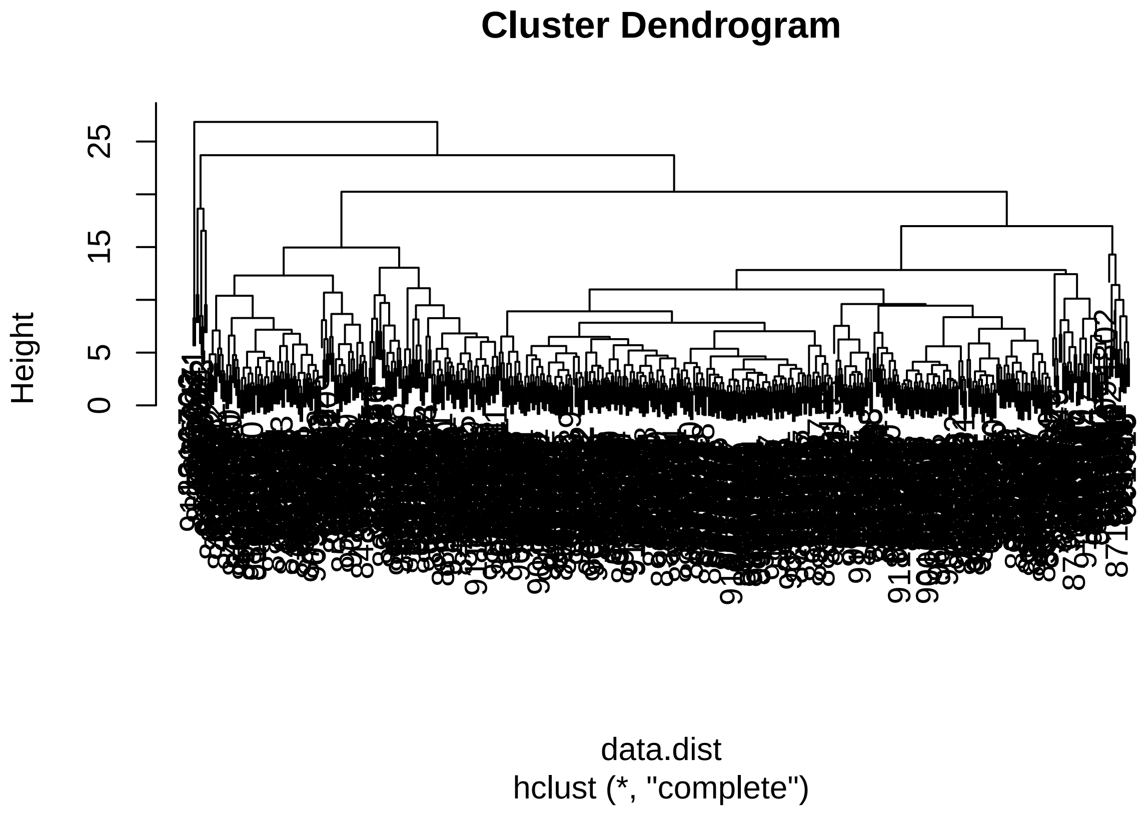
Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

The method "ward.D2" looked much better, as he two major clusters are more clearly separated in this method

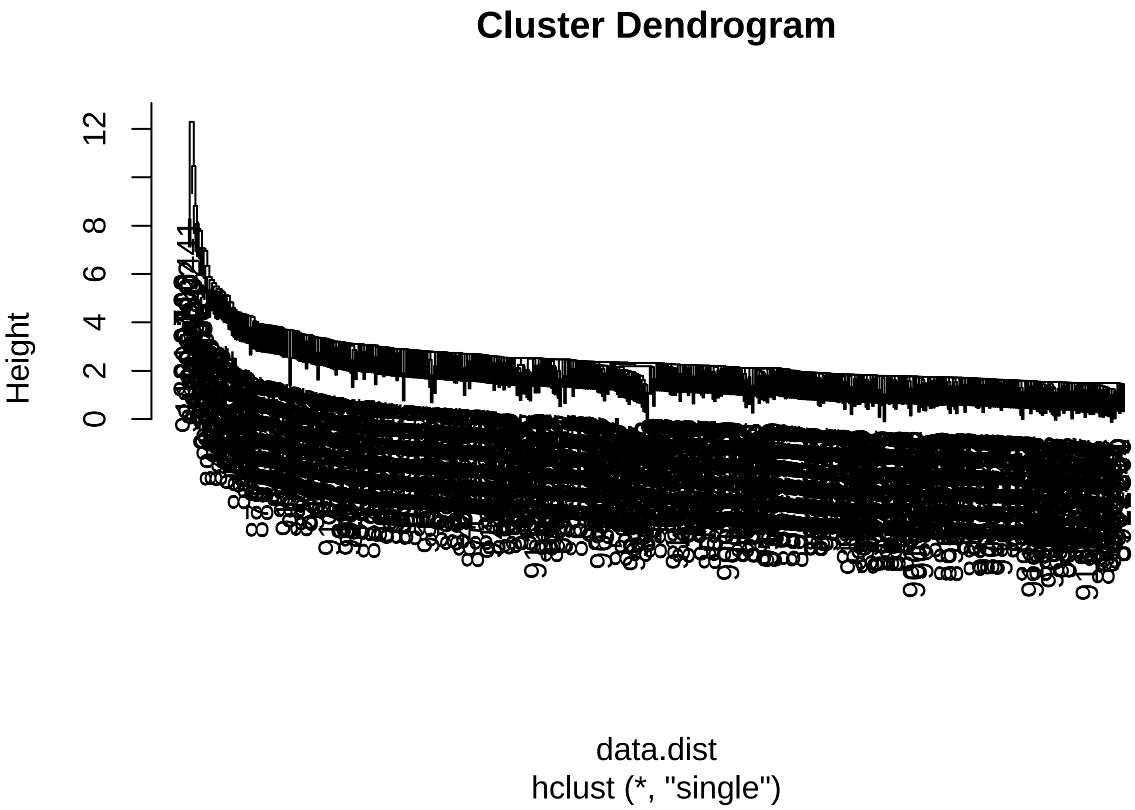
```
plot(hclust(data.dist, method = "average"))
```



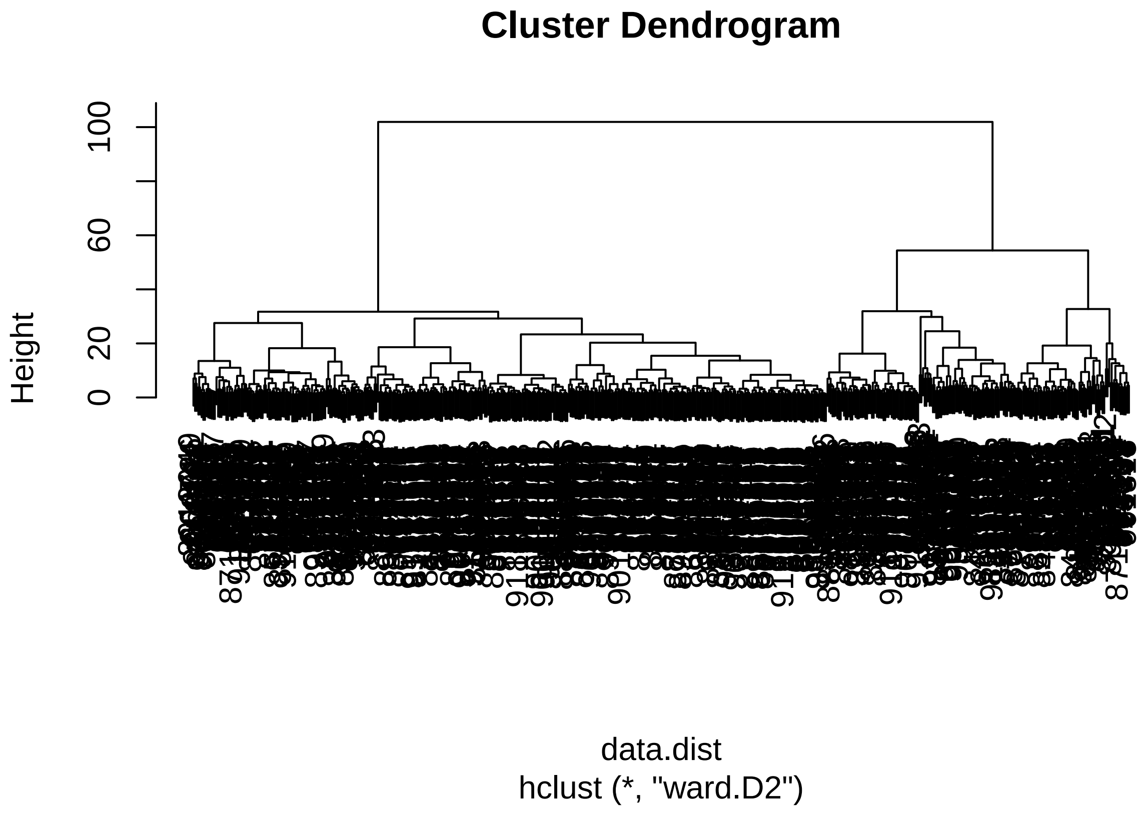
```
plot(hclust(data.dist, method = "complete"))
```



```
plot(hclust(data.dist, method = "single"))
```



```
plot(hclust(data.dist, method = "ward.D2"))
```



OPTIONAL: K-means clustering

K-means clustering and comparing results

```
wisc.km <- kmeans(wisc.data, centers= 2, nstart= 20)

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

```
##    diagnosis
##       B   M
##   1 356  82
##   2   1 130
```

```
table(wisc.km$cluster, wisc.hclust.clusters)
```

```
##    wisc.hclust.clusters
##       1   2   3   4   5
##   1  68   3 365   2   0
##   2 109   2  18   0   2
```

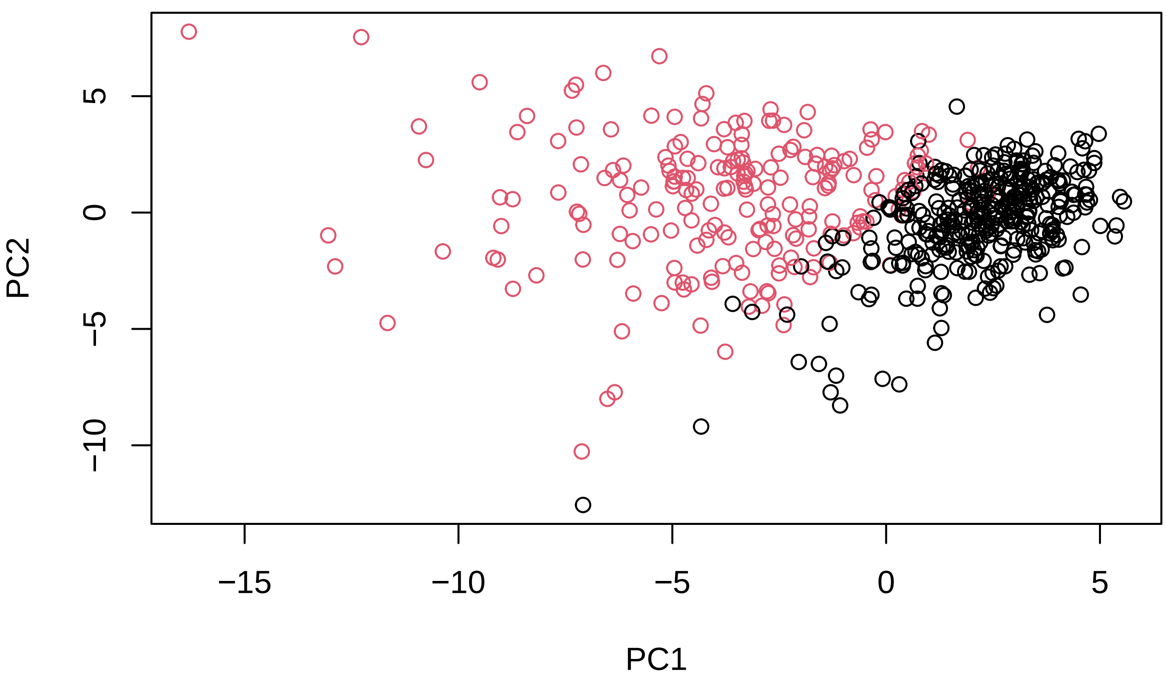
Q14. How well does k-means separate the two diagnoses? How does it compare to your hclust results?

It works well, but less accurate that hclust.

Combining methods

Here we aim to combine our PCA results with clustering. Essentially, we are going to cluster in “PC”, that is cluster on the results `wisc.pr`.

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

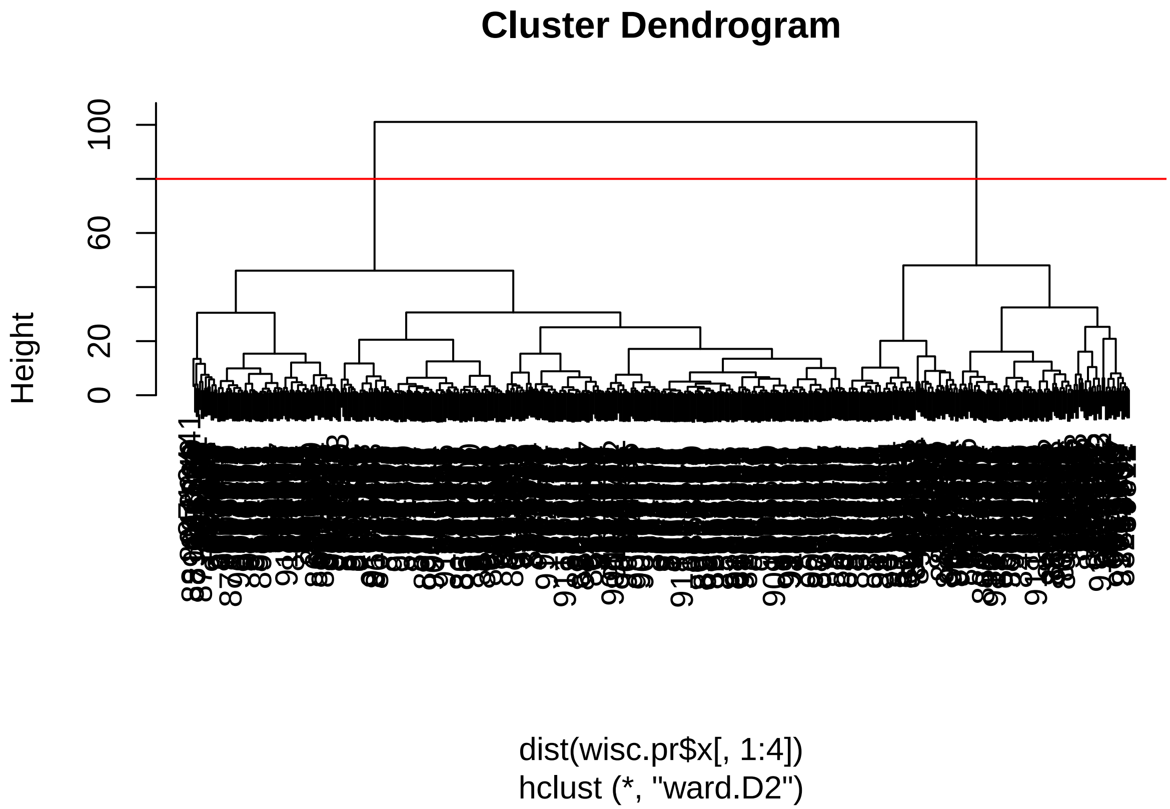


Clustering on PCA results

I will use 4 PCs and `hclust()` and `dist()` as an input.

```
wisc.pr.hclust <- hclust(dist(wisc.pr$x[,1:4]), method = "ward.D2")
```

```
plot(wisc.pr.hclust)
abline(h=80, col="red")
```



Let's find our cluster membership vector by cutting this tree into k=2 groups.

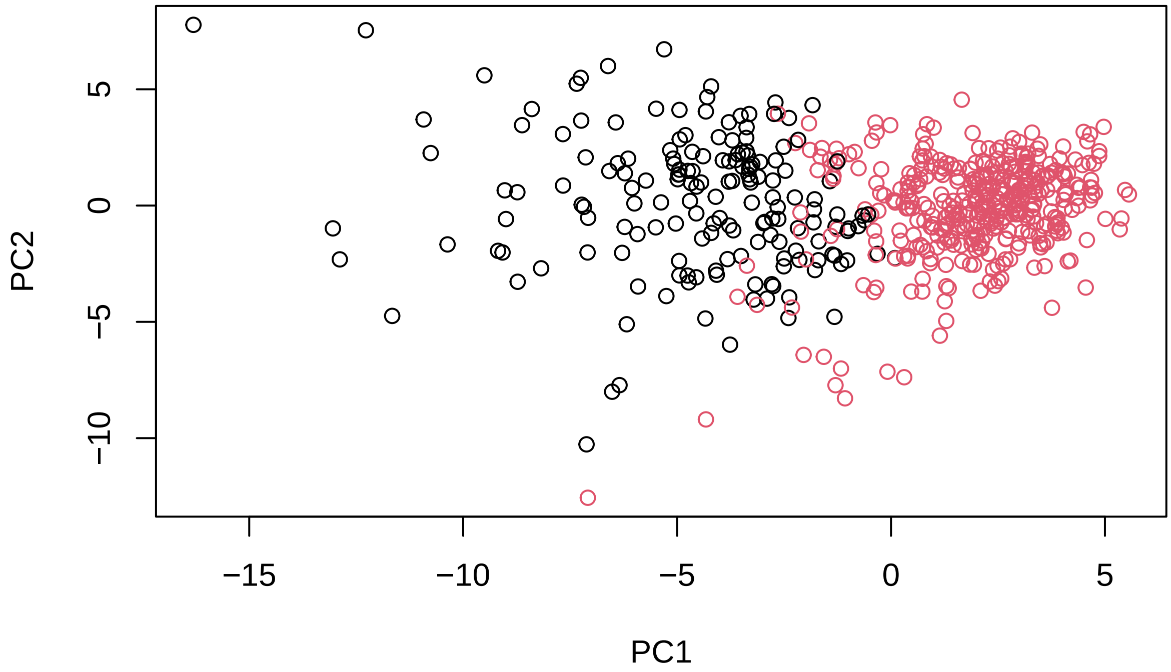
```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
## grps
##   1   2
## 171 398
```

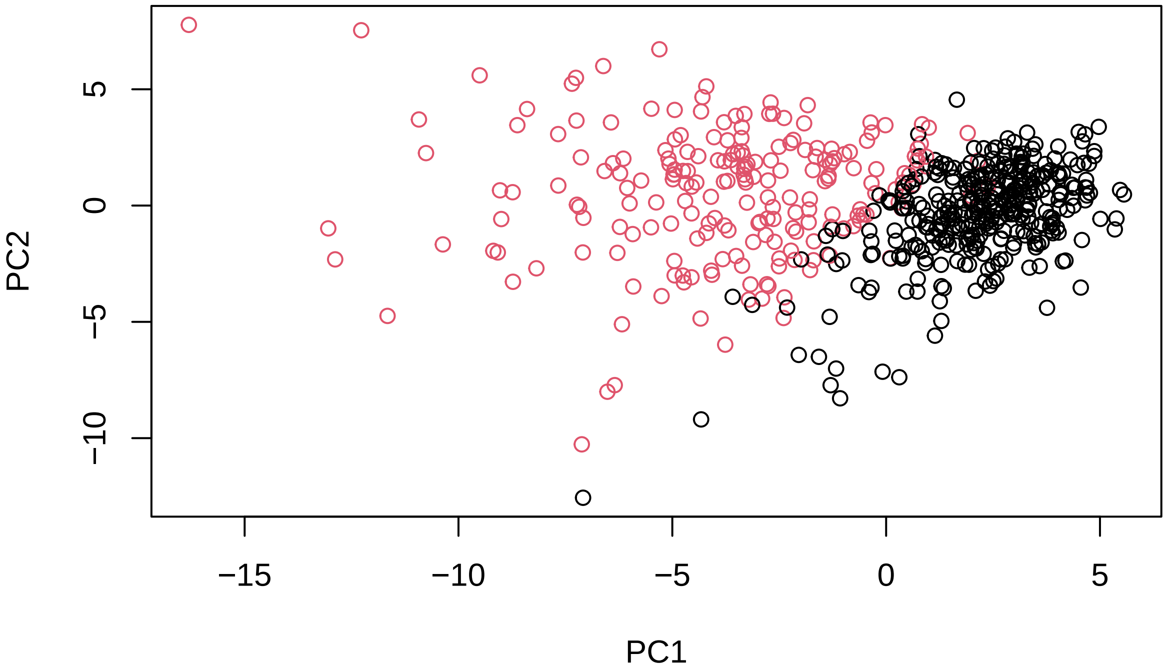
```
table(grps, diagnosis)
```

```
##     diagnosis
## grps   B   M
##    1   6 165
##    2 351  47
```

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



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



To match things up we can turn our groups into a factor and reorder the levels so cluster 2 comes first and thus gets the first color (black) and cluster 1 gets the second color (red).

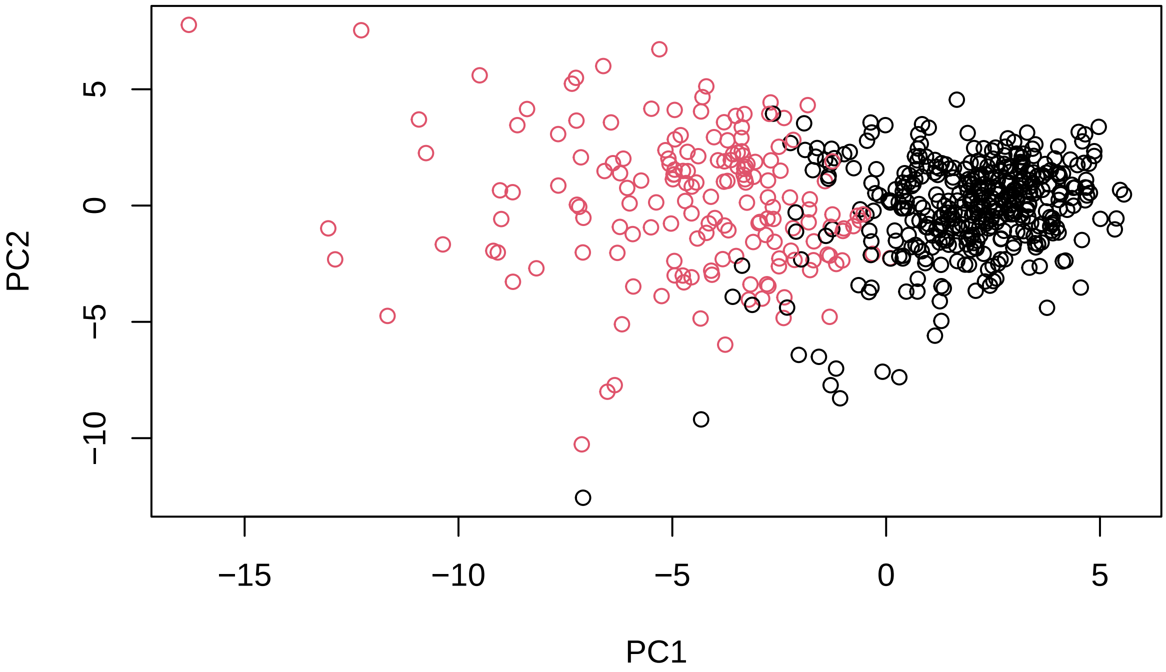
```
g <- as.factor(grps)
levels(g)
```

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

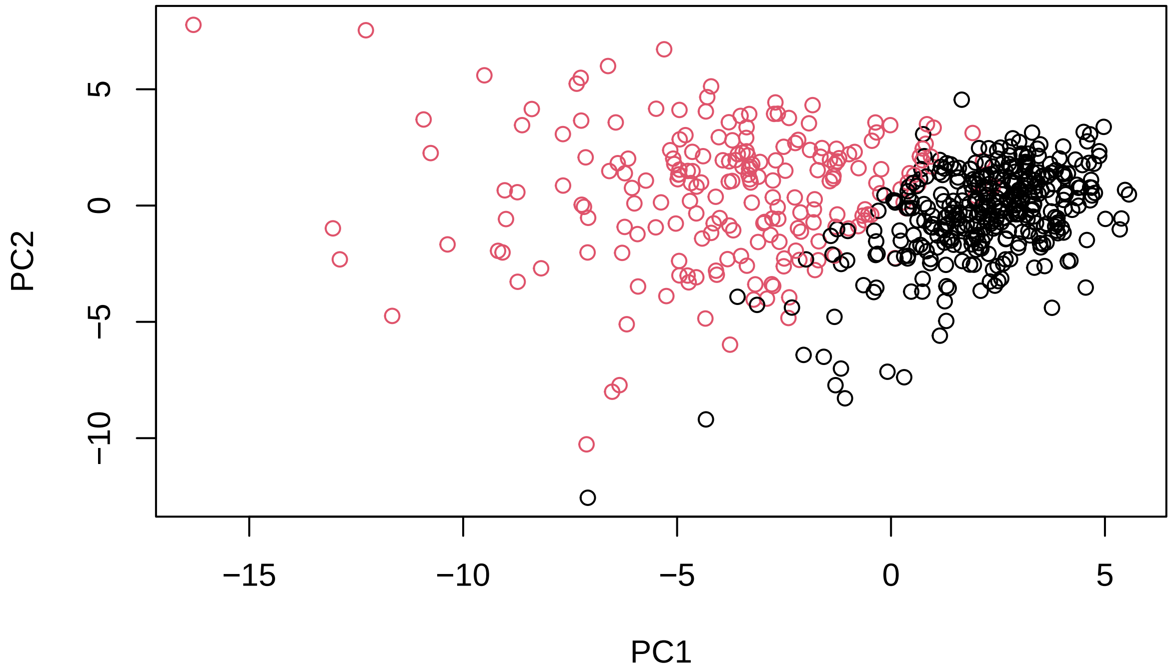
```
g <- relevel(g,2)
levels(g)
```

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

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



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



```
diagnosis
```

grps B M 1 6 165 2 351 47

TP: 165, FP:6 TN: 351, FN: 47

Accuracy, essentially how many did we get correct?

Sensitivity: TP/(TP+FN)

Specificity: TN/(TN+FN)

```
# **Accuracy**: (TP+TN)/total cases
# **Sensitivity**: TP/(TP+FN)
# **Specificity**: TN/(TN+FN)
Accuracy <- (165+351)/nrow(wisc.data)*100
Sensitivity <- (165)/(165+47)*100
Specificity <- (351)/(351+47)*100
evulation <- rbind(Accuracy, Sensitivity, Specificity)
evulation
```

```
##                 [,1]
## Accuracy    90.68541
## Sensitivity 77.83019
## Specificity 88.19095
```

Use the distance along the first 7 PCs for clustering i.e. wisc.pr$x[, 1:7]

```
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
```

```
# **Accuracy**: (TP+TN)/total cases
# **Sensitivity**: TP/(TP+FN)
# **Specificity**: TN/(TN+FN)
Accuracy <- (188+329)/nrow(wisc.data)*100
Sensitivity <- (188)/(188+24)*100
Specificity <- (329)/(329+24)*100
evulation <- rbind(Accuracy, Sensitivity, Specificity)
evulation
```

```
##                 [,1]
## Accuracy    90.86116
## Sensitivity 88.67925
## Specificity 93.20113
```

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

It looked good, with higher sensitivity and specificity.

> Q16. How well do the k-means and 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.

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

```
##    diagnosis
##       B   M
##   1 356  82
##   2   1 130
```

```
# **Accuracy**: (TP+TN)/total cases
# **Sensitivity**: TP/(TP+FN)
# **Specificity**: TN/(TN+FN)
Accuracy <- (130+356)/nrow(wisc.data)*100
Sensitivity <- (130)/(130+82)*100
Specificity <- (356)/(356+82)*100
evulation <- rbind(Accuracy, Sensitivity, Specificity)
evulation
```

```
##                  [,1]
## Accuracy    85.41301
## Sensitivity 61.32075
## Specificity 81.27854
```

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

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

```
# **Accuracy**: (TP+TN)/total cases
# **Sensitivity**: TP/(TP+FN)
# **Specificity**: TN/(TN+FN)
Accuracy <- (165+343)/nrow(wisc.data)*100
Sensitivity <- (165)/(165+40)*100
Specificity <- (343)/(343+40)*100
evulation <- rbind(Accuracy, Sensitivity, Specificity)
evulation
```

```
##                  [,1]
## Accuracy    89.27944
## Sensitivity 80.48780
## Specificity 89.55614
```

> Q17. Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

Using the distance along the first 7 PCs for clustering give the best specificity and sensitivity.

Prediction

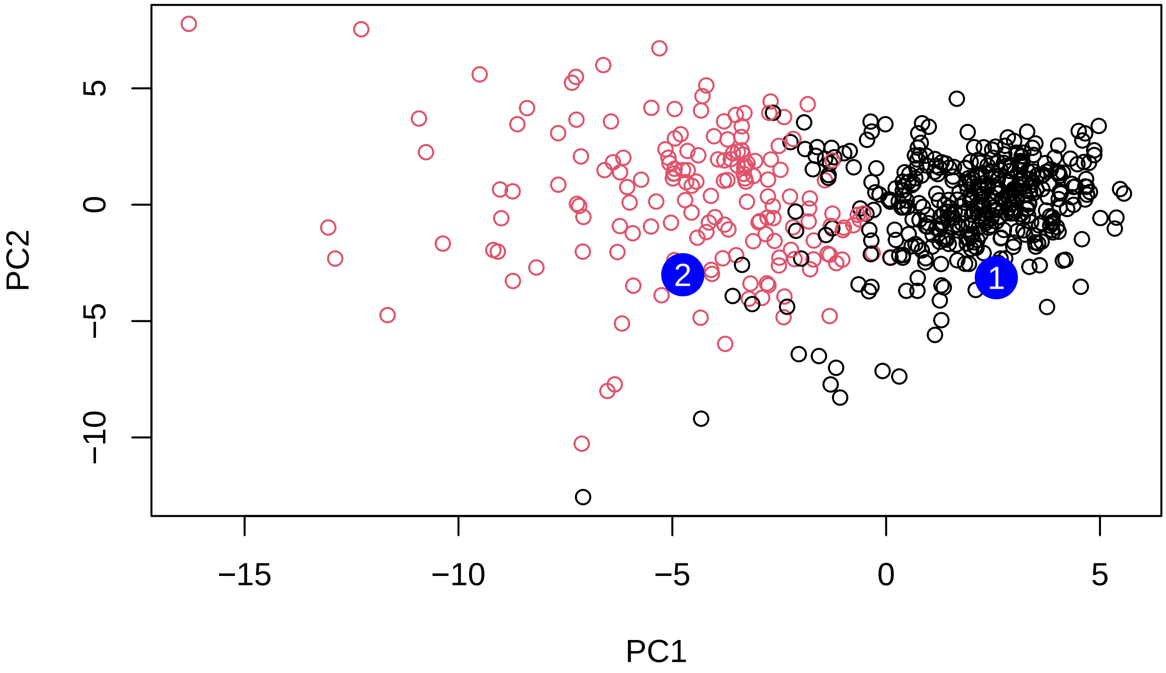
We will use the `predict()` function that will take our PCA model from before and new cancer cell data and project

```
#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
```

Now add these new samples to our PCA plot

```
plot(wisc.pr$x[,1:2], col=g)
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 pentients showed as red dots in group 2 should be prioritized for follow up, as it overlaps with the patients diagnosed as malignant in the previous analysis.