Class09 Miniproject

Tom Quach (A15549142)

10/26/2021

Importing our CSV file and reading it by using the read.csv() code

```
fna.data <- "WisconsinCancer.csv"
fna.data
```

```
## [1] "WisconsinCancer.csv"
```

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

We will now create a data.frame that omits the first column of the csv data because it is essentially the answer given by professional pathologist

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

Store the diagnosis column of the original dataset as a factor() which will be useful for plotting

```
diagnosis <- factor(wisc.df[, 1])
```

```
diagnosis
```

```
##   [1] 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
##  [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 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
## [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 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 B M M B B B B B B M 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 B M B M M 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
```

There are a total of 569 observations in this dataset

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

```
table(diagnosis)
```

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

There are 212 malignant diagnosis in this observations

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

```
(grep("mean", colnames(wisc.df)))
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11
```

```
length(grep("mean", colnames(wisc.df)))
```

```
## [1] 10
```

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

2.PCA

Checking the mean and standard deviation of the column section in our wisc.data

```
round(colMeans(wisc.data), 2)
```

```
##             radius_mean            texture_mean          perimeter_mean
##                   14.13                   19.29                   91.97
##               area_mean         smoothness_mean        compactness_mean
##                  654.89                    0.10                    0.10
##          concavity_mean     concave.points_mean           symmetry_mean
##                    0.09                    0.05                    0.18
##  fractal_dimension_mean               radius_se              texture_se
##                    0.06                    0.41                    1.22
##            perimeter_se                 area_se           smoothness_se
##                    2.87                   40.34                    0.01
##          compactness_se            concavity_se       concave.points_se
##                    0.03                    0.03                    0.01
##             symmetry_se    fractal_dimension_se             radius_worst
##                    0.02                    0.00                   16.27
##           texture_worst         perimeter_worst              area_worst
##                   25.68                  107.26                  880.58
##        smoothness_worst       compactness_worst         concavity_worst
##                    0.13                    0.25                    0.27
##    concave.points_worst          symmetry_worst fractal_dimension_worst
##                    0.11                    0.29                    0.08
```

```
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)?

PC1 captures 44.27% of the original variance

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

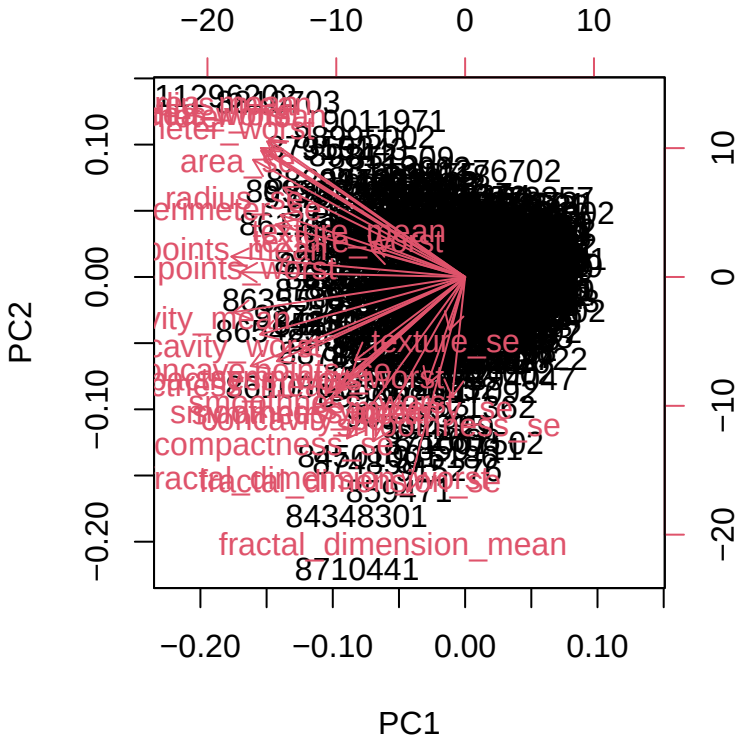
It takes at least 3 PCs to describe at least 70% of the orignial variacne

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

It takes at least PC7 to describe at least 90% of the original variance in the data

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

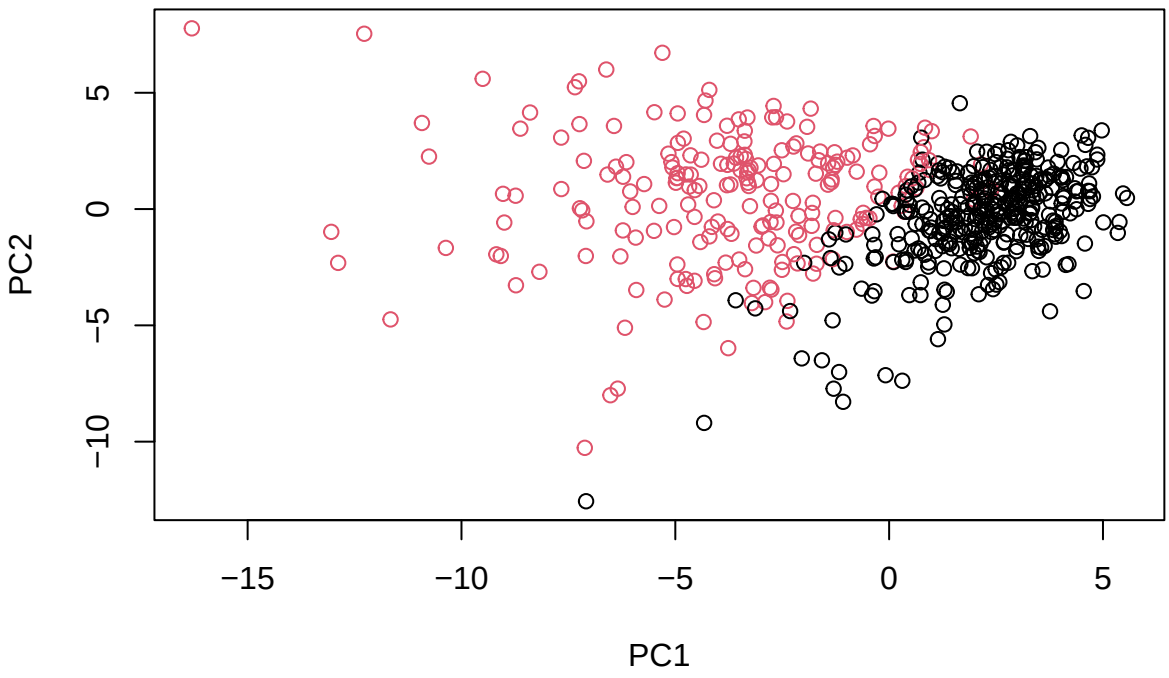
```
biplot(wisc.pr)
```



Nothing really stands out to me since it is just a mess of black cluster points and bunch of read lines going towards the left

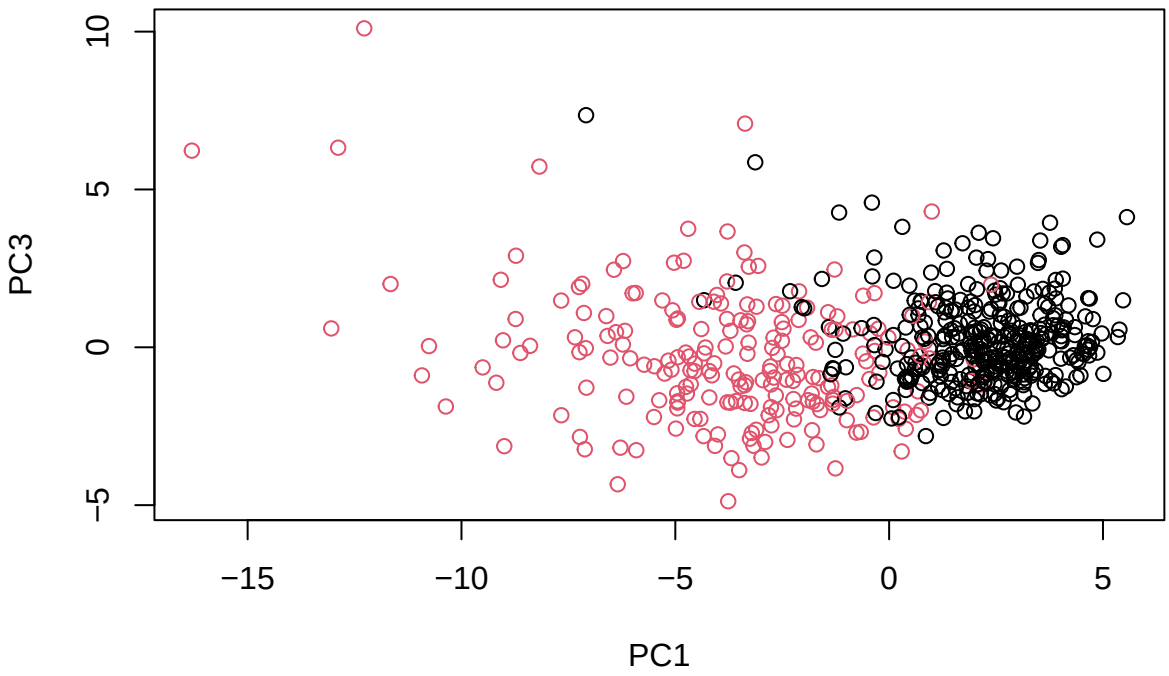
Generating a easier plot to read

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



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")
```



Between the two plots above, I notice that there are similar clusterings between PC1 vs PC2 and PC1 vs PC3. However, PC1 vs PC2 has a cleaner border between the two clustering compared to PC1 vs PC3 because PC2 measures more variance compared to PC3.

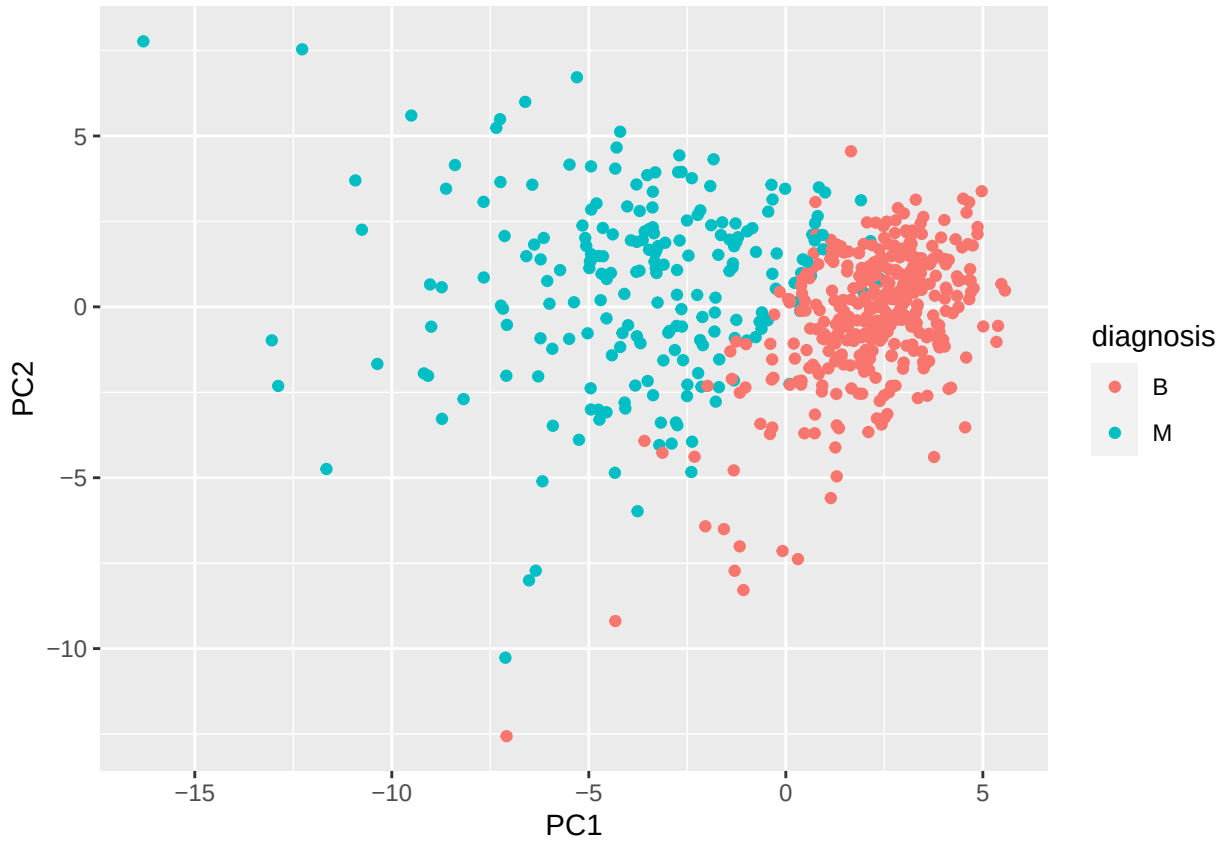
ggplot

Creating a data frame for us to use in ggplot2

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

Calling ggplot2 and making a scatter plot

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



Calculating the 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
```

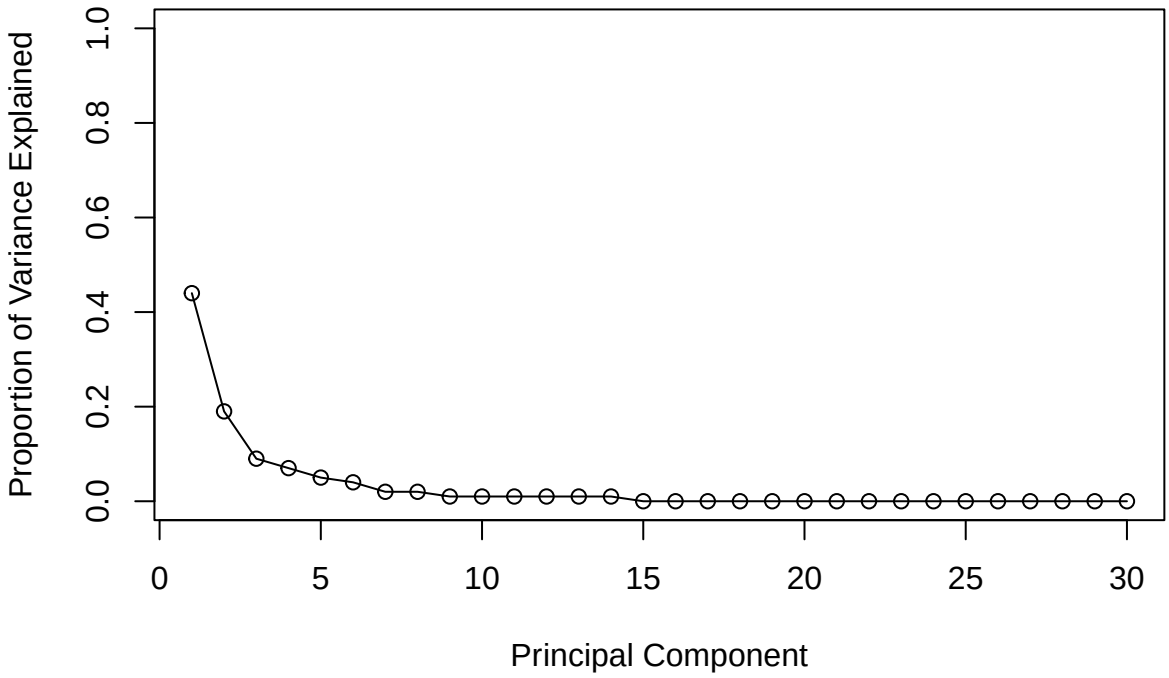
Variance explained by each principal component

```
pve <- round(pr.var / sum(pr.var) , 2)
pve
```

```
##  [1] 0.44 0.19 0.09 0.07 0.05 0.04 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.00
## [16] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

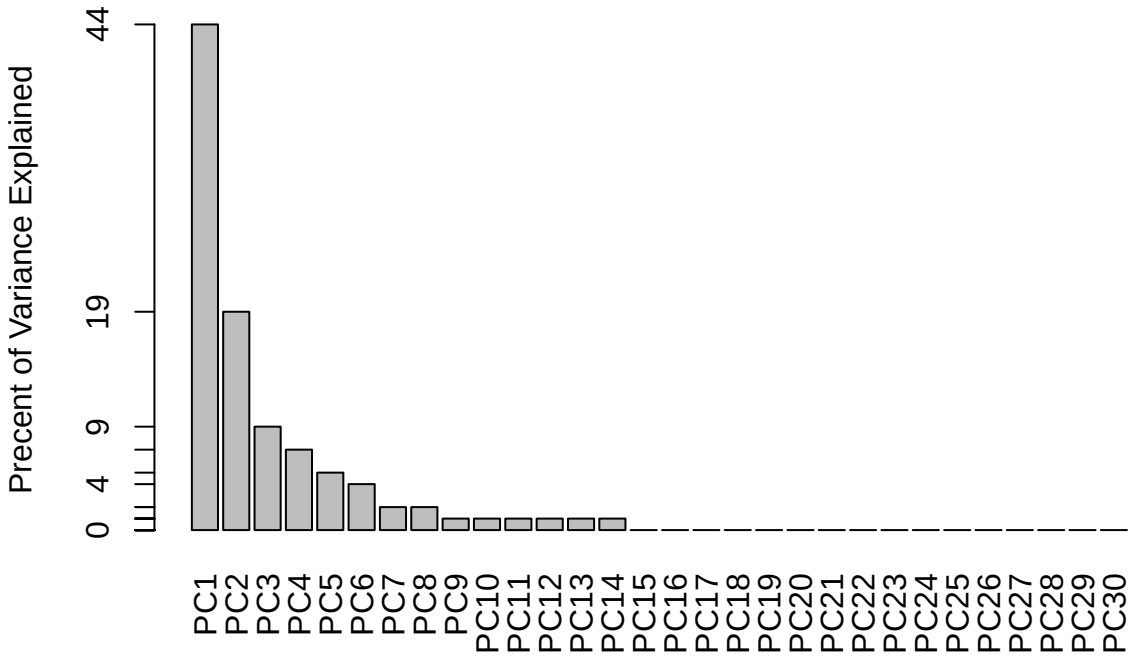
Plot variance explained for each principal component

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



Alternative graph

```
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 )
```



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
```

-0.2608538

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

```
var <- summary(wisc.pr)
sum(var$importance[3,] < 0.8)
```

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

We need at least 5 principal components to explain 80% of the variance of the data.

Hierarchical clustering

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

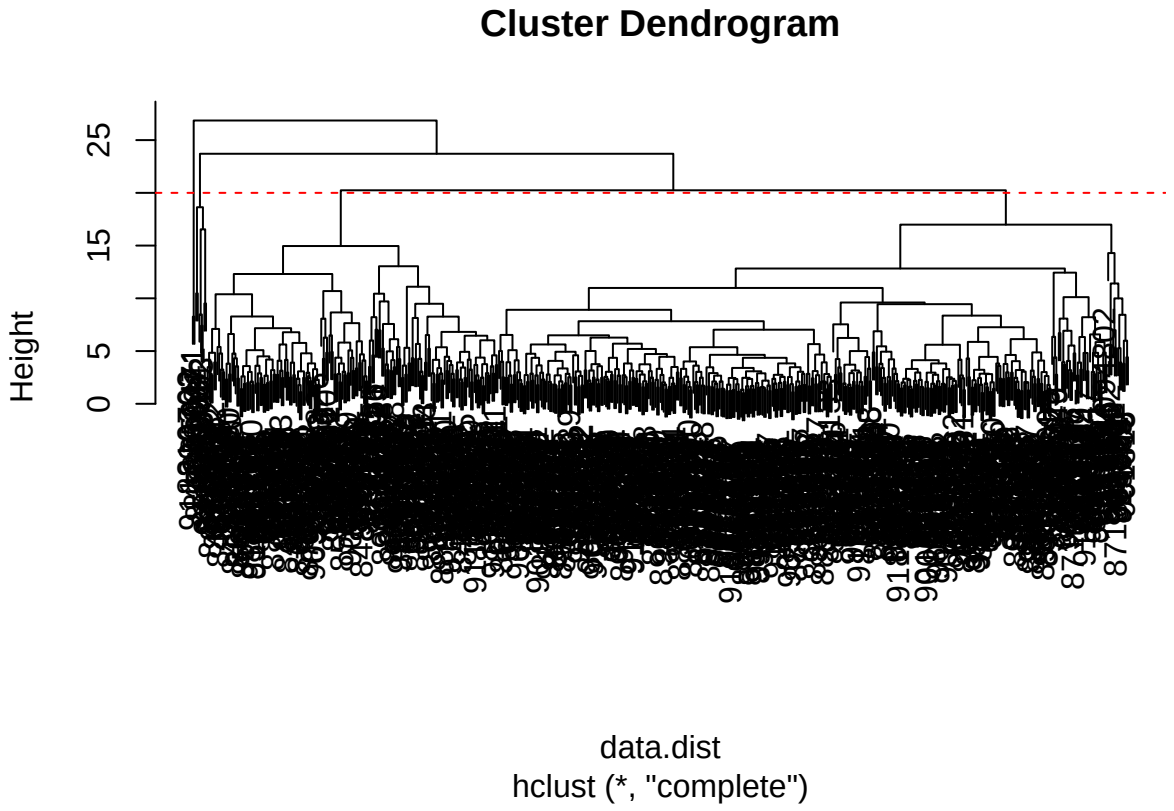
```
data.dist <- dist(data.scaled)
```

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

```
##
## Call:
## hclust(d = data.dist, method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 569
```

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

```
plot(wisc.hclust)
abline(h = 20, col="red", lty=2)
```



There will be 4 clusters at height 20

Using cutree() to get 4 clusters

```
wisc.hclust.clusters <- cutree(wisc.hclust, h = 20)
```

```
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?

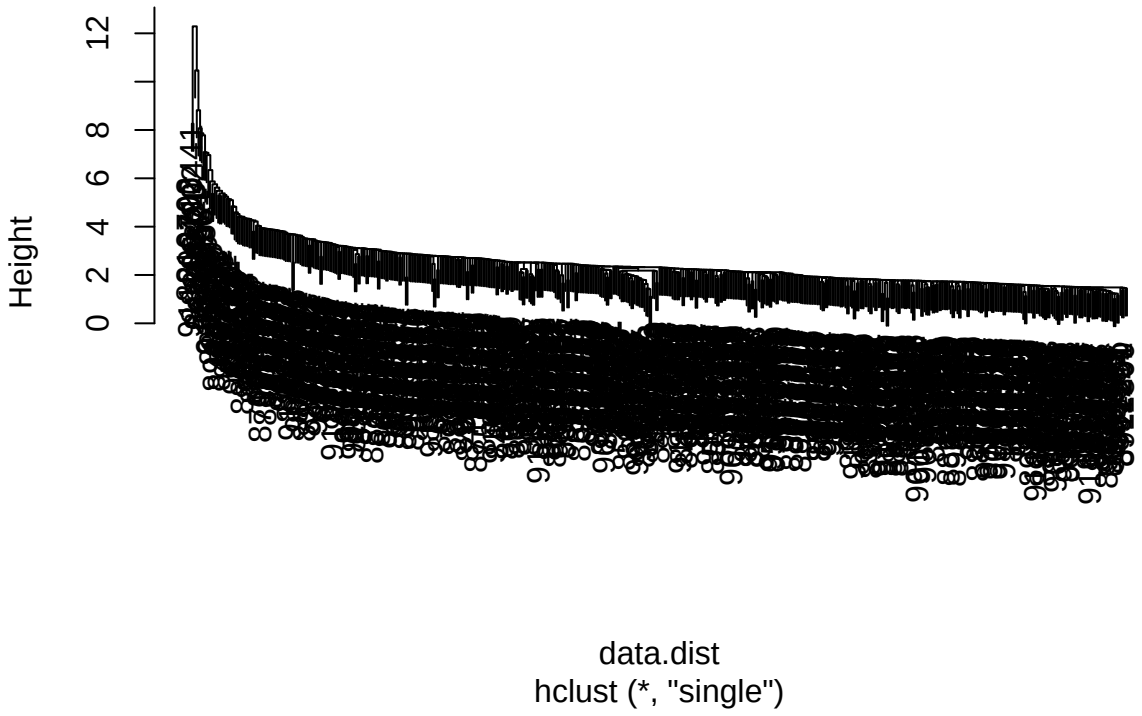
```
wisc.hclust.clusters <- cutree(wisc.hclust, h = 19)
table(wisc.hclust.clusters, diagnosis)
```

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

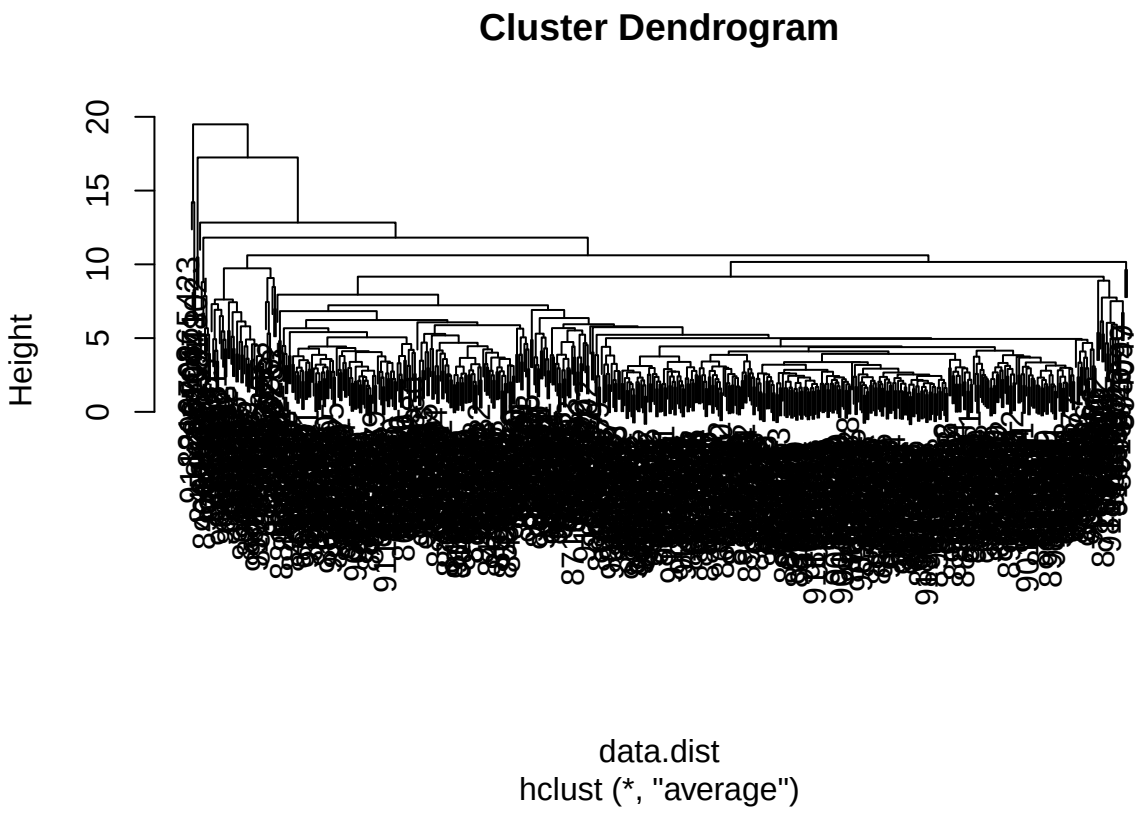
After trying the clusters between 2 and 10, I found that having 4 clusters give us the most efficient readings in the difference between benign and malignant results

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

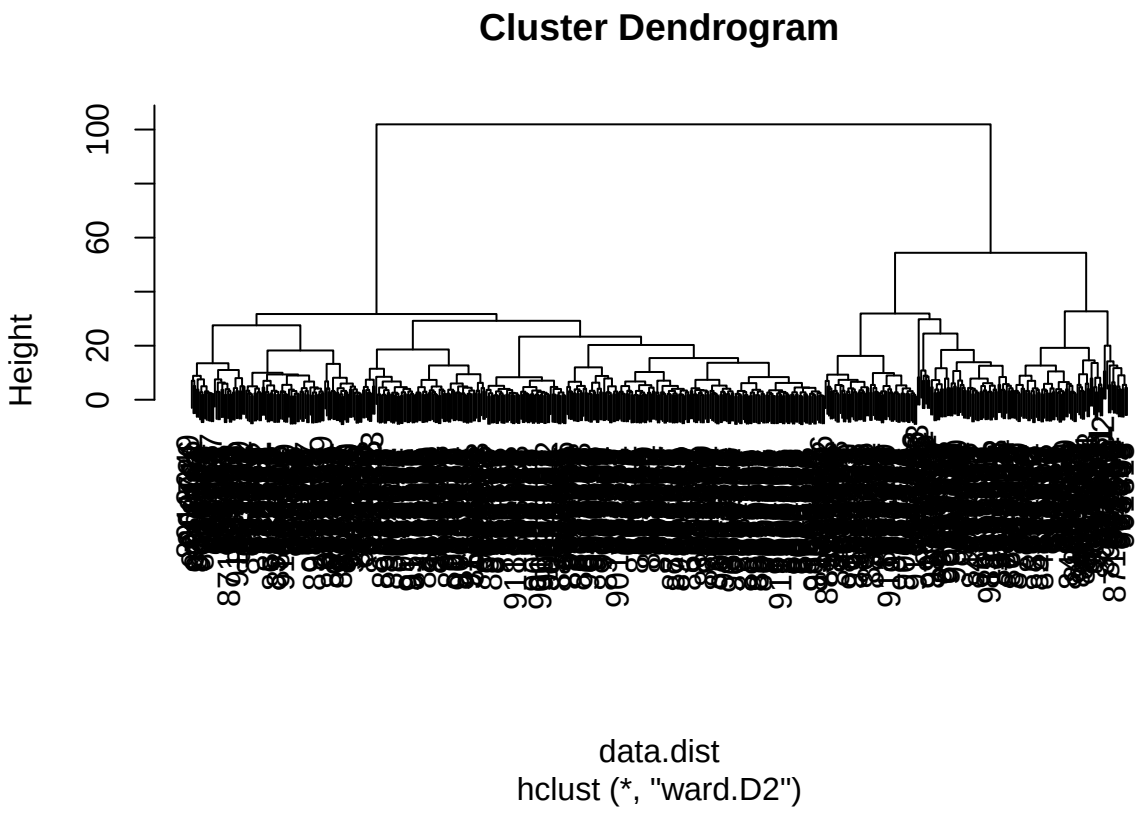
```
wisc.hclust <- hclust(data.dist, method = "single")
plot(wisc.hclust)
```




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




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



I also personally really enjoy the ward.D2 method as I can see the clusters and separation more clearly

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

Combing methods

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

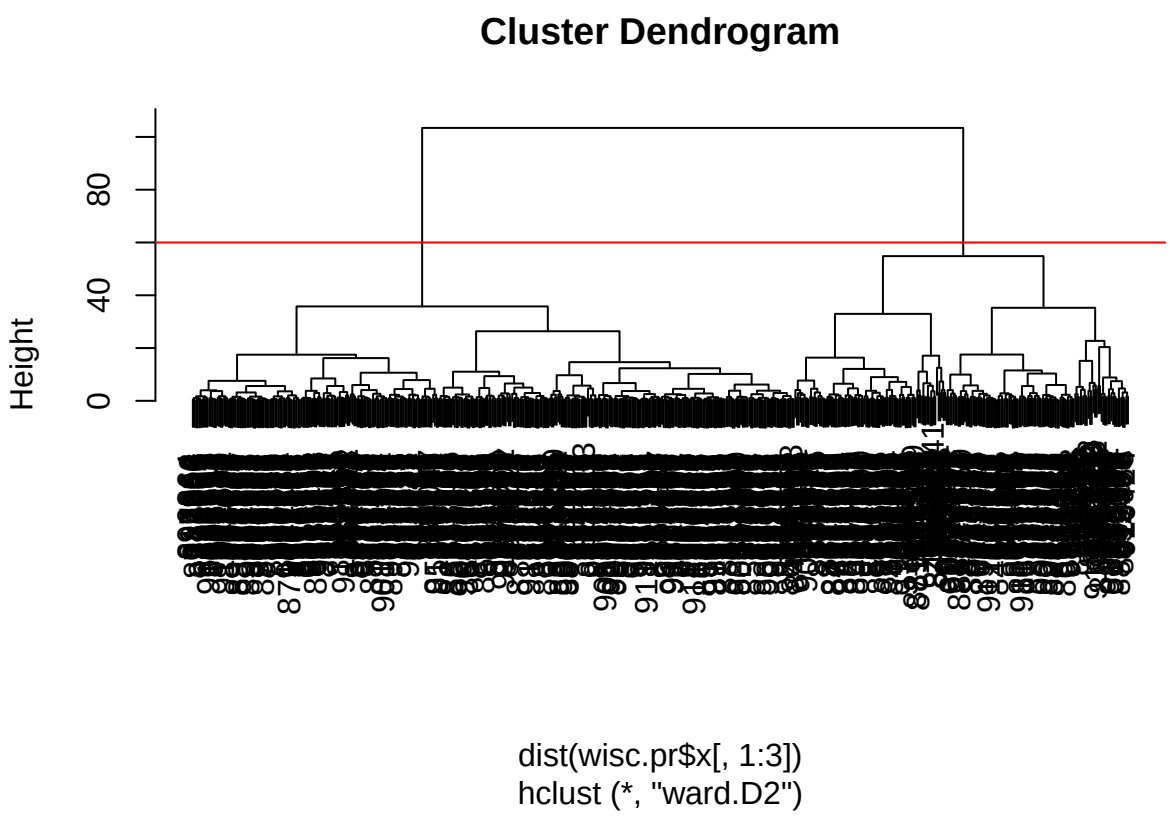
```
## grps
##   1   2
## 184 385
```

```
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
```

```
wisc.pr.hclust <- hclust(dist(wisc.pr$x[,1:3]), method = "ward.D2")
plot(wisc.pr.hclust)
abline(h=60, col = "red")
```




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

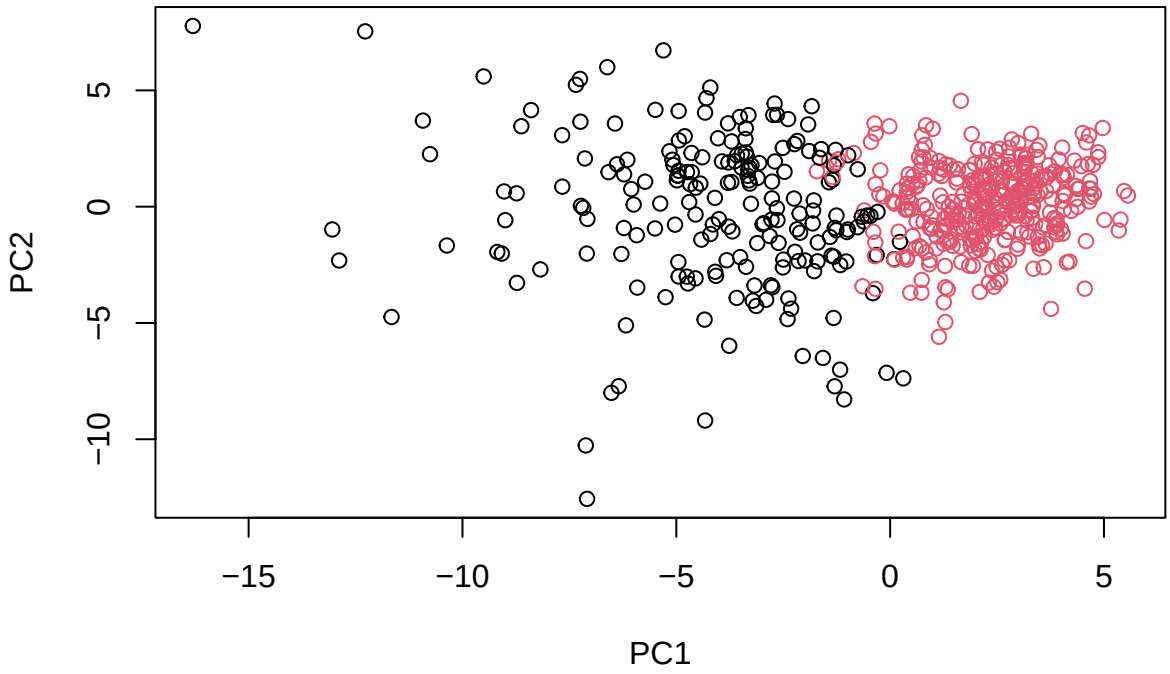
```
## grps
##   1   2
## 203 366
```

Cross table compare of diagnosis and my cluster groups

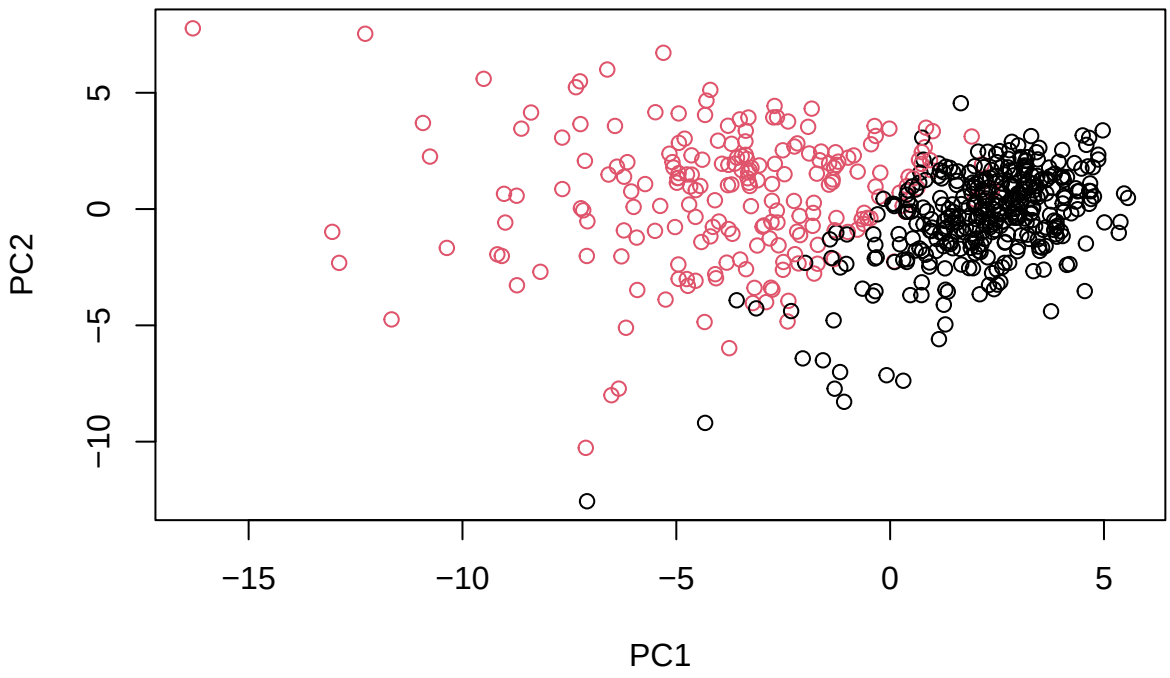
```
table(diagnosis, grps)
```

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

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




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



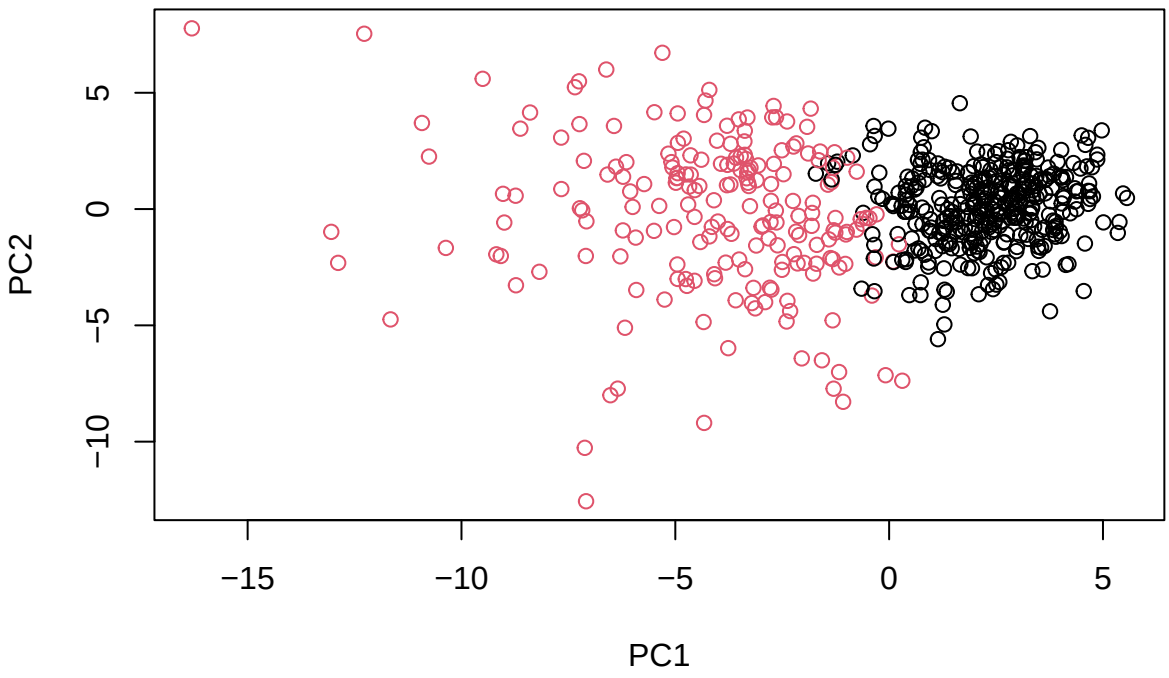

```
g <- as.factor(grps)
levels(g)
```

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

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

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

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



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

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

```
wisc.pc.hclust.clusters <- cutree(wisc.pc.hclust, k=2)
```

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

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

The newly created model with four clusters seem to be able to separate out the two dianoses pretty well. We can see the difference between the two efficiently

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  14 175
##   2 343  37
```

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

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

The previous k-means and hierarchical clustering seems to be able to show us better variance compared to this current one. However, the current one we have is doing an efficient job as well.

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

```
(333 + 179)/nrow(wisc.data)
```

```
## [1] 0.8998243
```

```
333/(333+24)
```

```
## [1] 0.9327731
```

Sensitivity

```
table(diagnosis)
```

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

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

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

```
wisc.pc.hclust.clusters.sensitivity <- 188/212
wisc.pc.hclust.clusters.sensitivity
```

```
## [1] 0.8867925
```

```
wisc.km.sensitivity <- 175/212
wisc.km.sensitivity
```

```
## [1] 0.8254717
```

```
wisc.hclust.clusters.sensitivity <- 165/212
wisc.hclust.clusters.sensitivity
```

```
## [1] 0.7783019
```

The wisc.km.sensitivty model gave me the best model for sensitivity

specificity

```
table(diagnosis)
```

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

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

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

```
wisc.pc.hclust.clusters.specificity <- 329/357
wisc.pc.hclust.clusters.specificity
```

```
## [1] 0.9215686
```

```
wisc.km.specificity <- 343/357
wisc.km.specificity
```

```
## [1] 0.9607843
```

```
wisc.hclust.clusters.specificity <- 343/357
wisc.hclust.clusters.specificity
```

```
## [1] 0.9607843
```

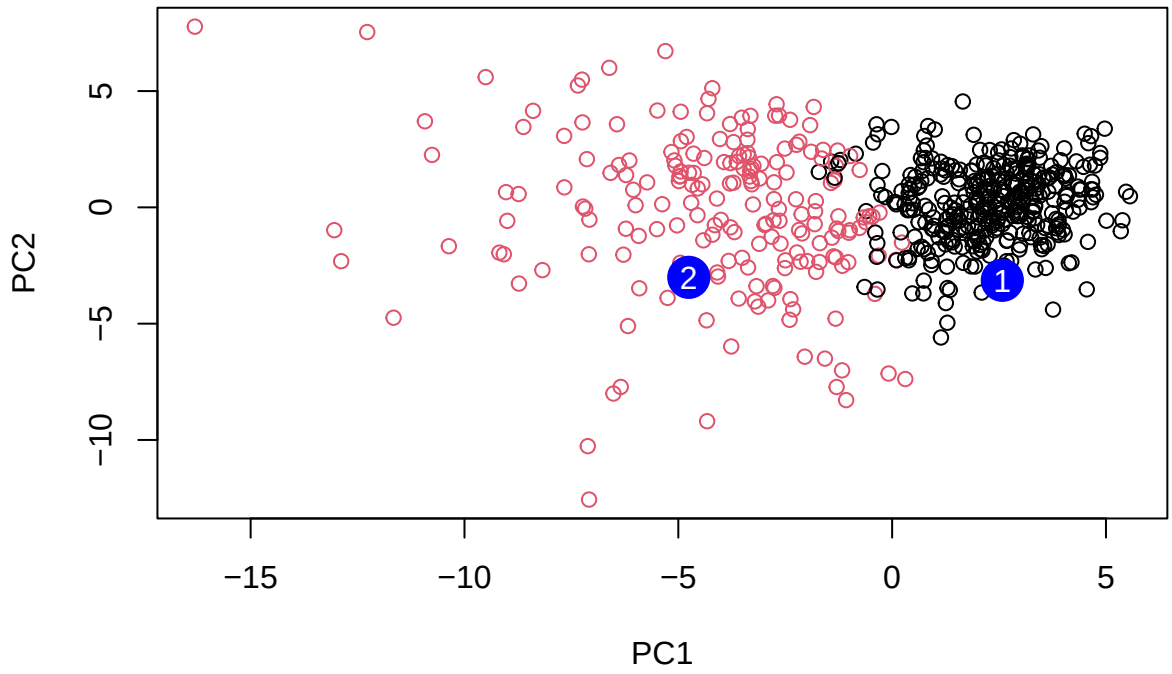
Both the wisc.km and wisc.hclust.clusters value gave me the best specificity value

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
```

```
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")
```



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

We should focus more on the patients in group 2 because they are more malignat