mini project

Rachel Diao

2/14/2022

```
#Import file first
fna.data <- 'WisconsinCancer.csv'
wisc.df <- read.csv(fna.data, row.names=1)

#head(wisc.df)
```

Omit the first column (diagnosis) because it's unnecessary for analysis for now. However, we will store the vector for use as a factor later.

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

#Store as a vector
diagnosis <- wisc.df$diagnosis
```

Question 1

There are 569 observations in this dataset.

```
#Number of observations = number of rows in the dataframe
nrow(wisc.df)
```

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

Question 2

There are 212 malignant diagnoses.

```
#Find out how many patients had a malignant diagnosis using grep for 'M' within character vector diagno
#grep() returns the positions in the character vector; find the length of this list to get number
length(grep('M', diagnosis))
```

```
## [1] 212
```

Question 3

10 variables in the data are suffixed with '_mean'.

```
length(grep('_mean', colnames(wisc.df)))
```

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

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
```

```
#Run PCA on data!
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
```

Question 4

44.27% of the original variance is captured by first principal component (PC1).

Question 5

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

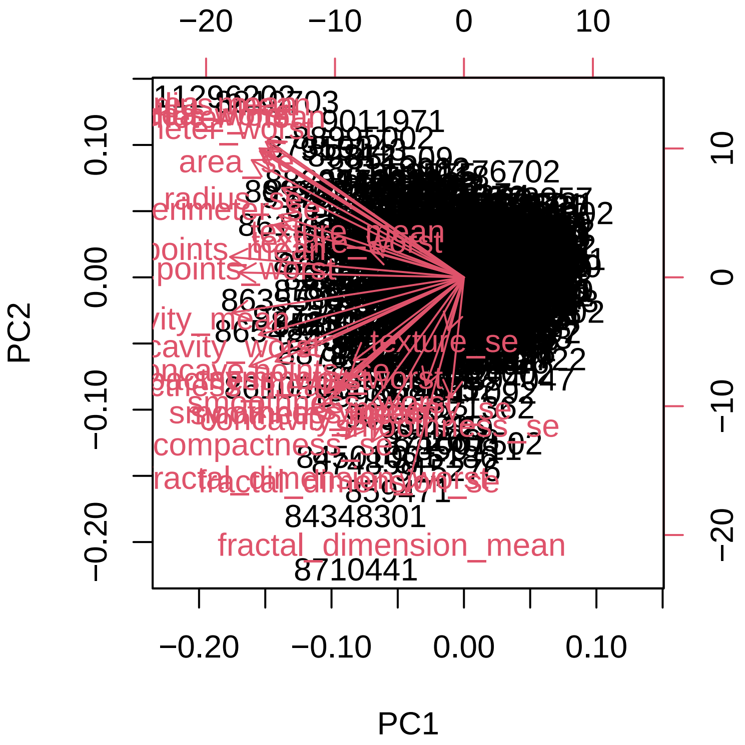
Question 6

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

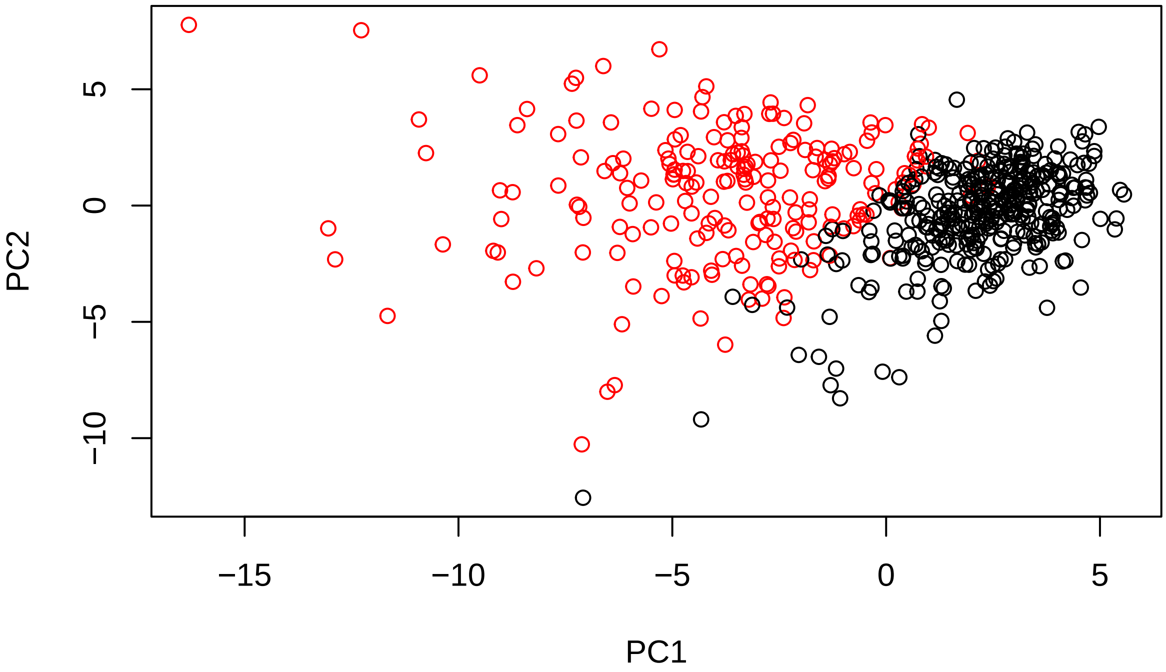
Question 7

This biplot is much too cluttered because here are too many variables to account for per observation and the dataset is also much too large for this visualization to be effective.

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

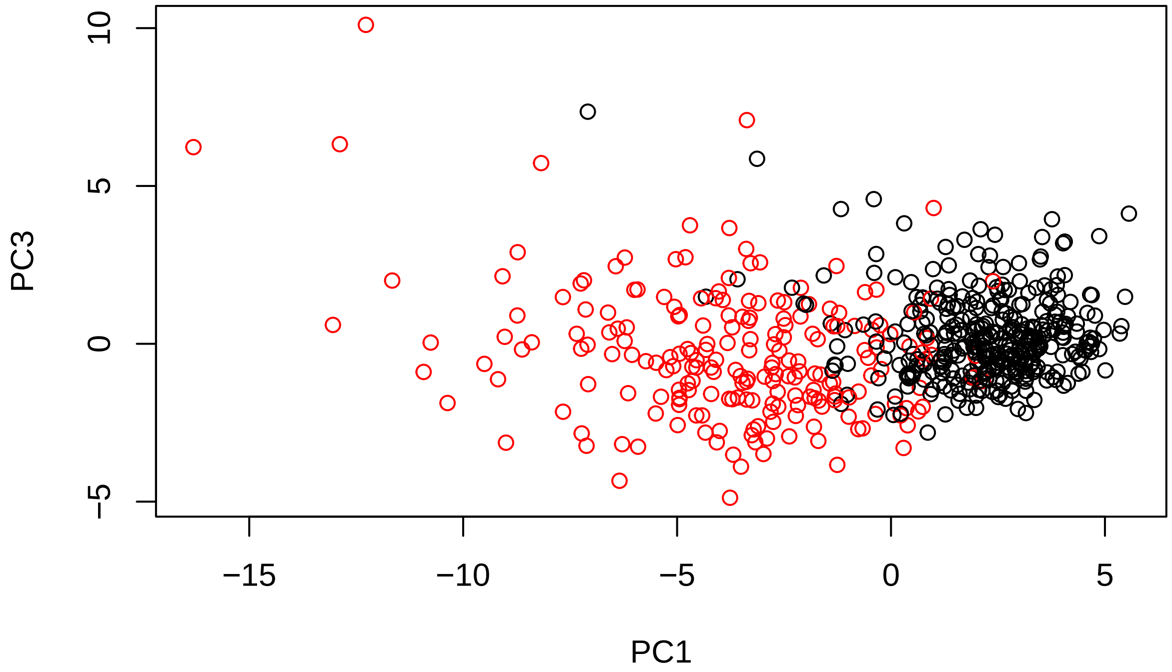


```
# Scatter plot observations by components 1 and 2
#Want each point colored based on diagnosis vector using ifelse()
#ifelse(conditional, if yes, if not)
plot(wisc.pr$x[,1], wisc.pr$x[,2], col = ifelse(diagnosis == 'M','red','black'),
     xlab = "PC1", ylab = "PC2")
```



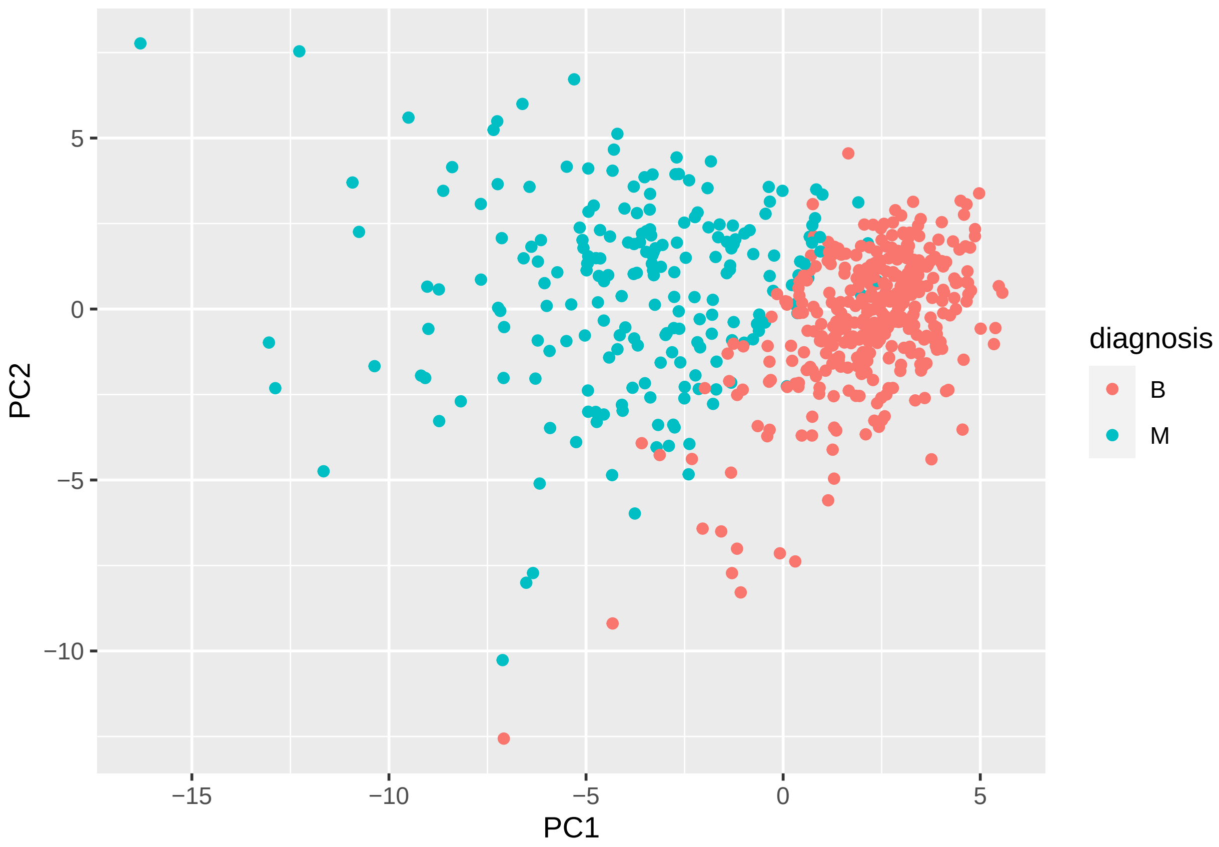
## Question 8 The plot of PCs 1 and 2 seems to generate 2 slightly more separate clusters than the plot of PCs 1 and 3. This may be because PC2 explains a greater proportion of variance than PC3.

```
# Repeat for components 1 and 3
plot(wisc.pr$x[,1], wisc.pr$x[,3], col = ifelse(diagnosis == 'M','red','black'),
     xlab = "PC1", ylab = "PC3")
```



Now plot in ggplot2!

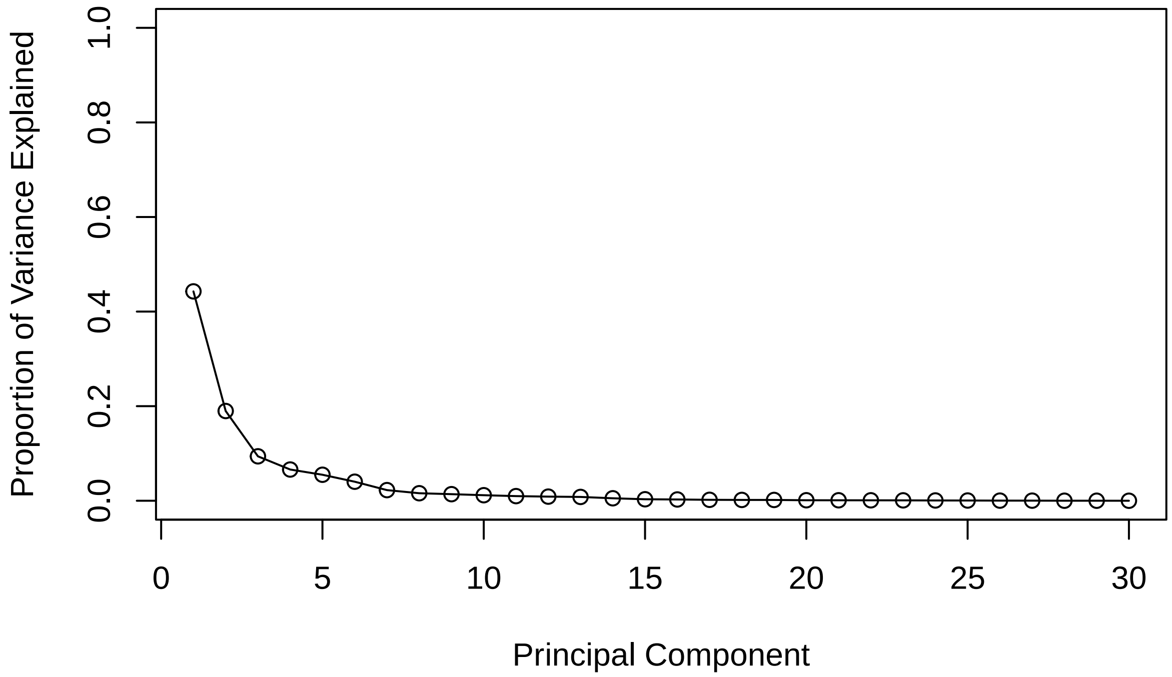
```
# Create a data.frame for ggplot
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis

# Load the ggplot2 package
library(ggplot2)

# Make a scatter plot colored by diagnosis
ggplot(df) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```



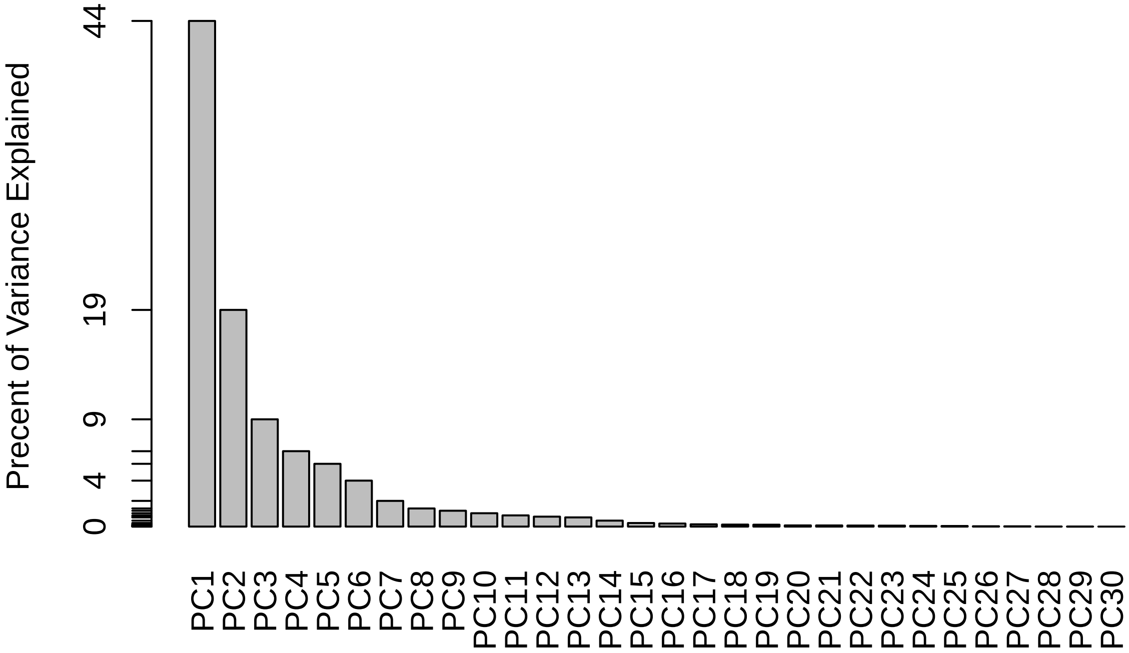
```
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

```
# Variance explained by each principal component: pve
pve <-  wisc.pr$sdev^2/sum(wisc.pr$sdev^2)

# 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 )
```



Question 9

Loading scores show the influence of each variable on each principle component. The loading for concave.points_mean for the first principal component is -0.26085376.

```
wisc.pr$rotation[,1]
```

```
##             radius_mean            texture_mean          perimeter_mean
##             -0.21890244             -0.10372458             -0.22753729
##               area_mean         smoothness_mean        compactness_mean
##             -0.22099499             -0.14258969             -0.23928535
##          concavity_mean     concave.points_mean           symmetry_mean
##             -0.25840048             -0.26085376             -0.13816696
##  fractal_dimension_mean               radius_se              texture_se
##             -0.06436335             -0.20597878             -0.01742803
##            perimeter_se                 area_se           smoothness_se
##             -0.21132592             -0.20286964             -0.01453145
##          compactness_se            concavity_se       concave.points_se
##             -0.17039345             -0.15358979             -0.18341740
##             symmetry_se    fractal_dimension_se            radius_worst
##             -0.04249842             -0.10256832             -0.22799663
##           texture_worst         perimeter_worst              area_worst
##             -0.10446933             -0.23663968             -0.22487053
##        smoothness_worst       compactness_worst         concavity_worst
##             -0.12795256             -0.21009588             -0.22876753
##    concave.points_worst          symmetry_worst fractal_dimension_worst
##             -0.25088597             -0.12290456             -0.13178394
```

Question 10

A minimum of 5 PCs are required to explain 80% of the variance in the data.

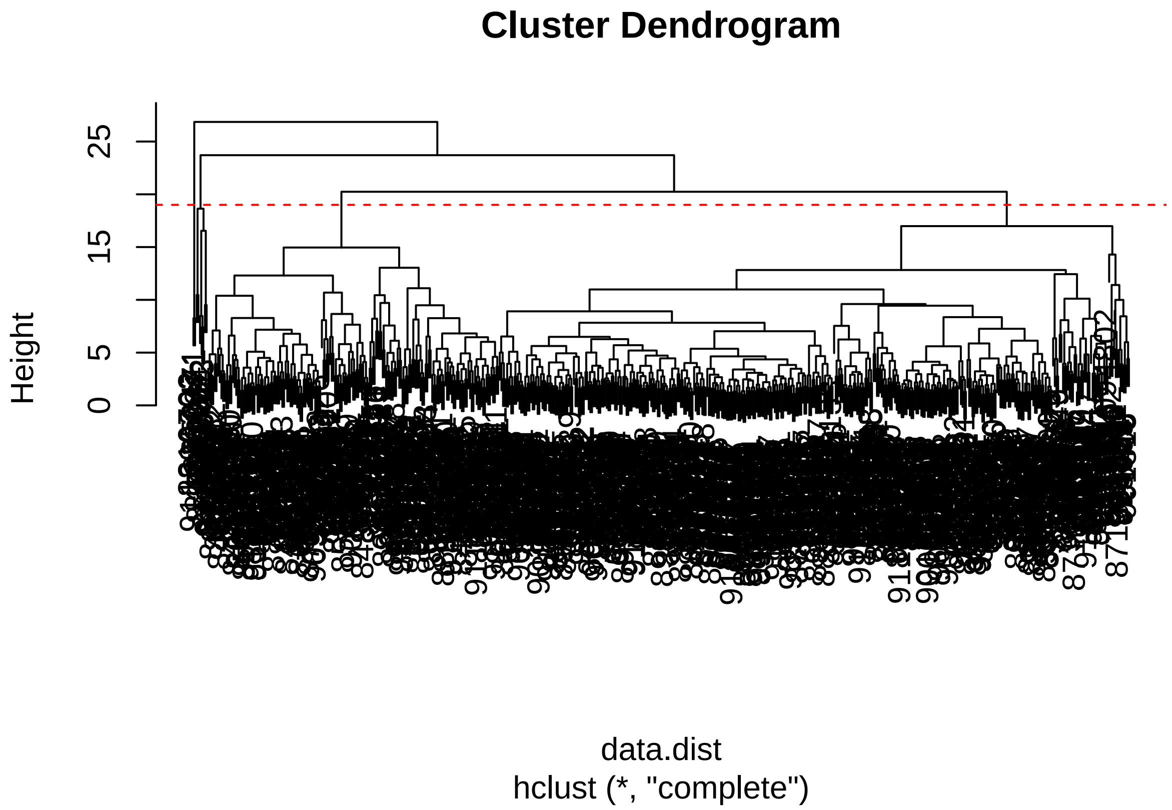
Hierarchical Clustering

```
# Scale the wisc.data data using the "scale()" function
data.scaled <- scale(wisc.data)

#Calculate Euclidean distances
data.dist <- dist(data.scaled)

#Perform clustering using complete linkage
wisc.hclust <- hclust(data.dist, method='complete')
```

Question 11

The height at which the clustering model has 4 clusters is h=19.

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



Cut the tree so there are only 4 clusters (change argument k)

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

#Compare cluster membership to diagnoses
table(wisc.hclust.clusters, diagnosis)
```

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

Question 12

The cluster vs. diagnoses match is only slightly better when the tree is cut into higher numbers of clusters, with certain clusters being able to completely match to a benign diagnosis (there isn’t much of a change in the matching to malignant diagnoses).

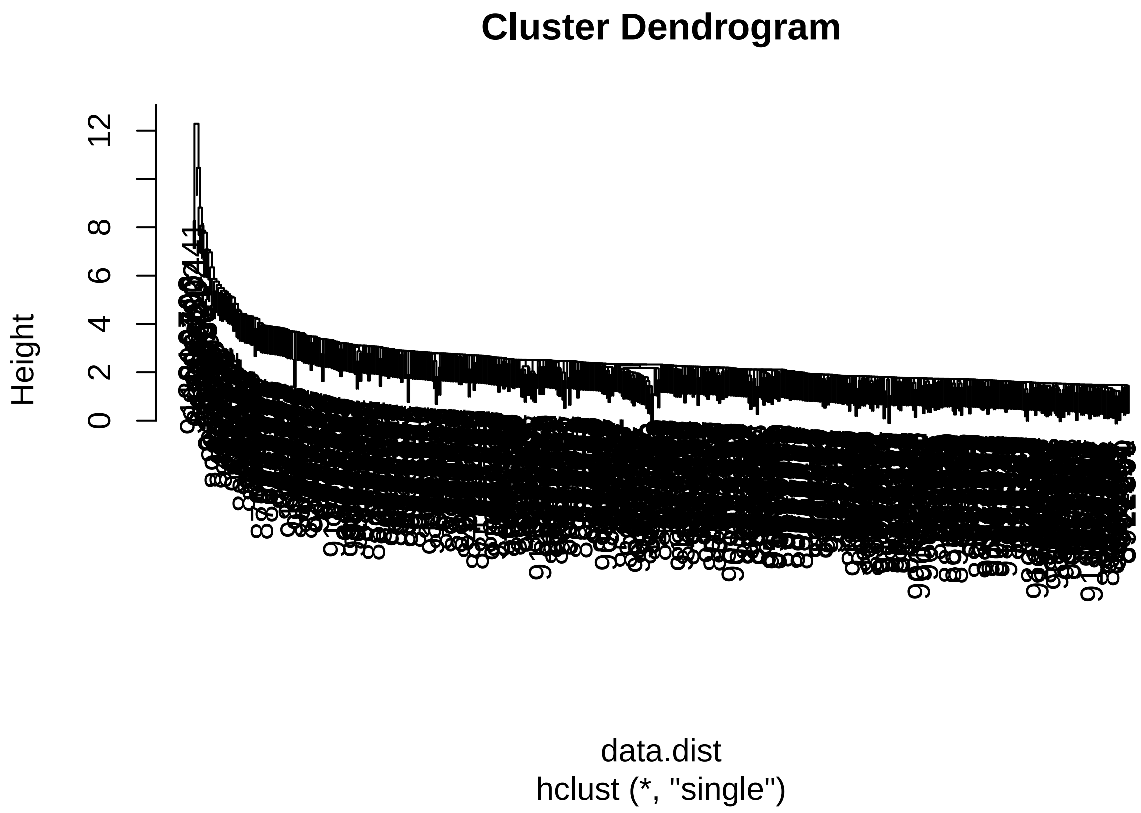
```
wisc.hclust.clusters3 <- cutree(wisc.hclust, k=2)
table(wisc.hclust.clusters3, diagnosis)
```

```
##                      diagnosis
## wisc.hclust.clusters3   B   M
##                        1 357 210
##                        2   0   2
```

Question 13

Ward's method gives my favorite results for this dataset because for the number of clusters that I chose (k=4), it provided greater separation between clusters than any of the other methods. Each cluster more clearly mapped to a particular diagnosis more clearly than with the other methods.

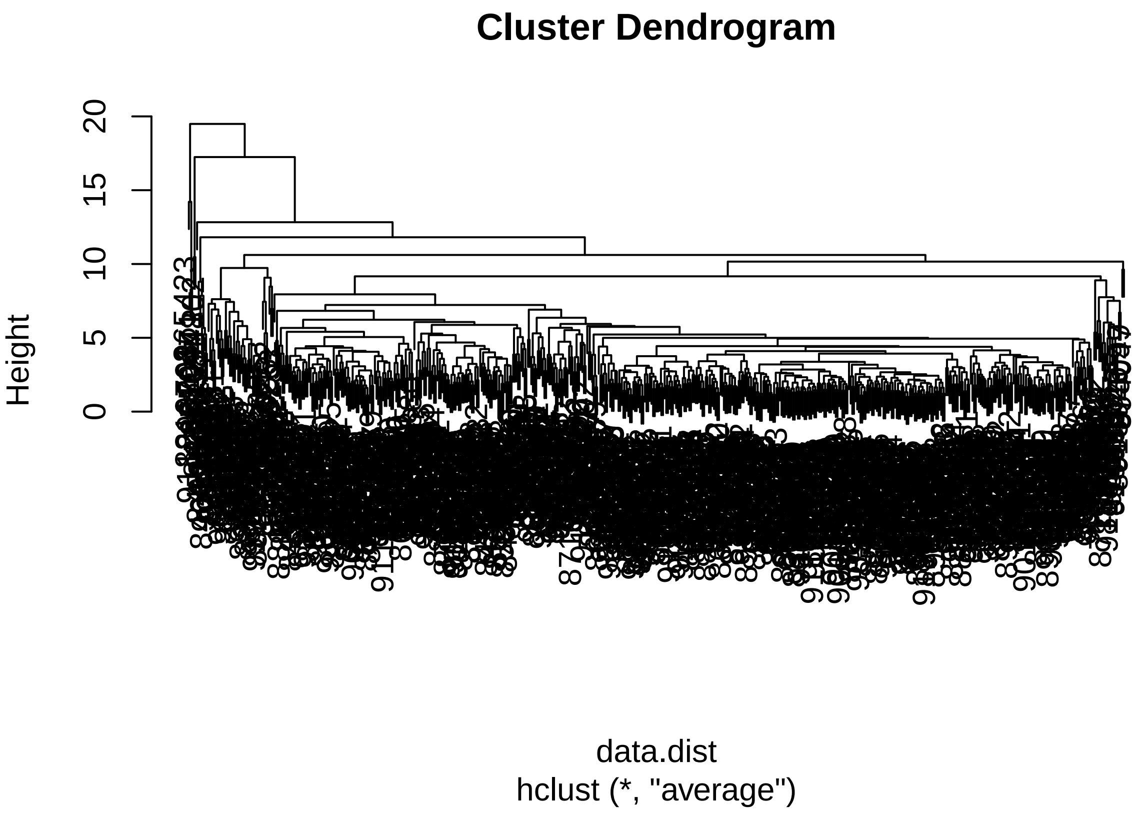
```
#Perform clustering using single linkage
wisc.hclust_single <- hclust(data.dist, method='single')
plot(wisc.hclust_single)
```



```
wisc.hclust.clusters_s <- cutree(wisc.hclust_single, k=4)
table(wisc.hclust.clusters_s, diagnosis)
```

```
##                         diagnosis
## wisc.hclust.clusters_s   B   M
##                       1 356 209
##                       2   1   0
##                       3   0   2
##                       4   0   1
```

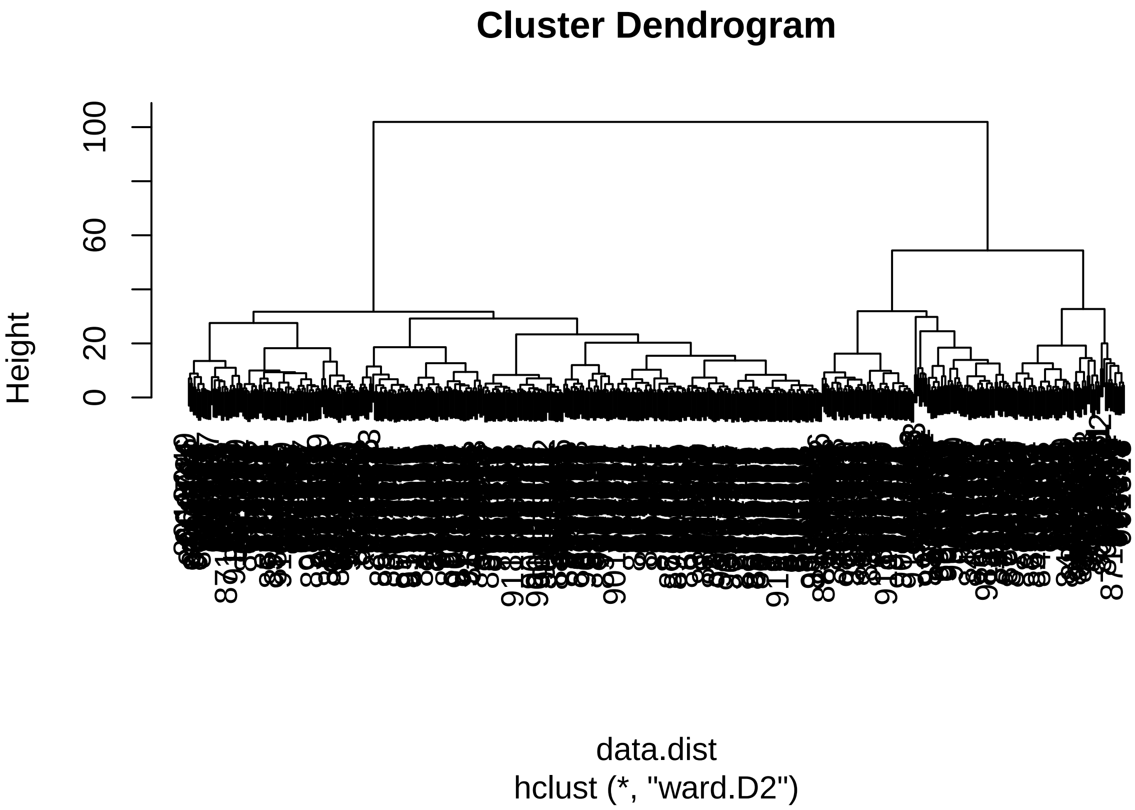
```
#Perform clustering using average linkage
wisc.hclust_av <- hclust(data.dist, method='average')
plot(wisc.hclust_av)
```



```
wisc.hclust.clusters_av <- cutree(wisc.hclust_av, k=4)
table(wisc.hclust.clusters_av, diagnosis)
```

```
##                        diagnosis
## wisc.hclust.clusters_av   B   M
##                       1 355 209
##                       2   2   0
##                       3   0   1
##                       4   0   2
```

```
#Perform clustering using Ward's method
wisc.hclust_ward <- hclust(data.dist, method='ward.D2')
plot(wisc.hclust_ward)
```



```
wisc.hclust.clusters_w <- cutree(wisc.hclust_ward, k=4)
table(wisc.hclust.clusters_w, diagnosis)
```

```
##                         diagnosis
## wisc.hclust.clusters_w   B   M
##                        1   0 115
##                        2   6  48
##                        3 337  48
##                        4  14   1
```

K-Means Clustering

Question 14

K-means clustering separates out diagnosis clusters more robustly than does the hierarchical clustering methods we used above for the same # of clusters k (k=2). In the k-means clustering, cluster 2 is very clearly the malignant diagnosis cluster and cluster 1 much more clearly skews towards diagnoses.

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

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

```
#Compare to hierarchical clustering
table(wisc.hclust.clusters3, diagnosis)
```

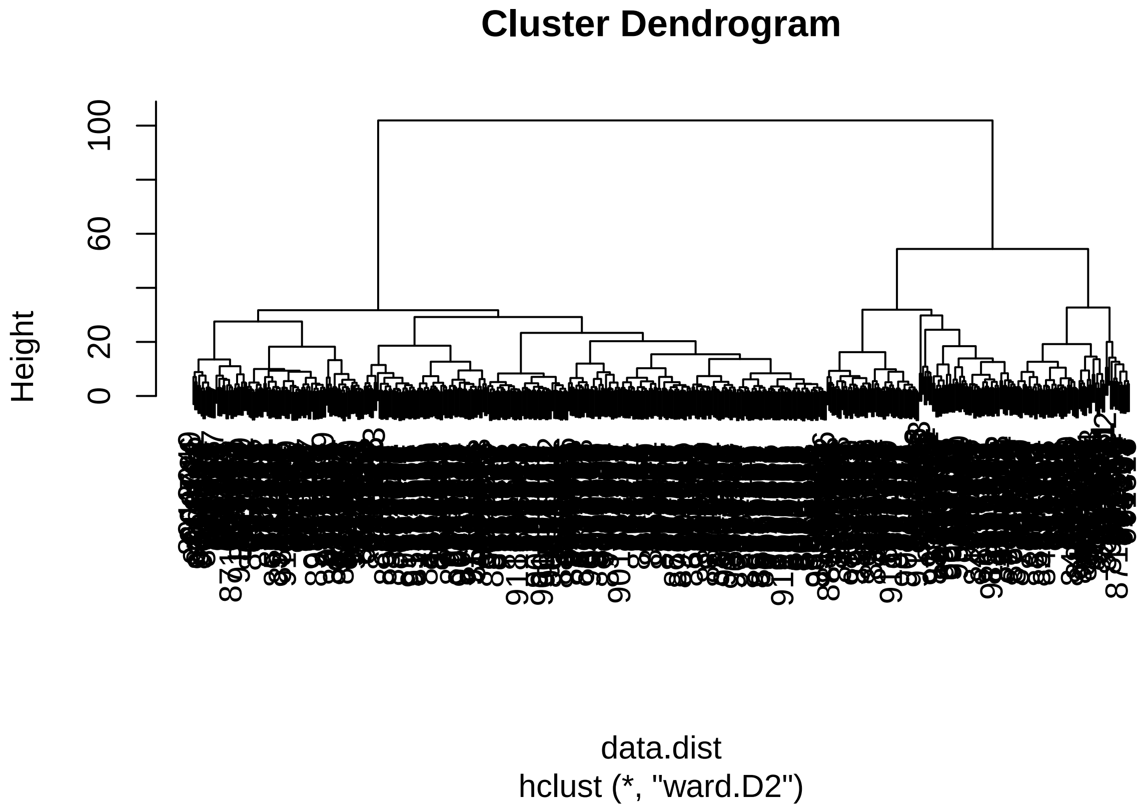
```
##                      diagnosis
## wisc.hclust.clusters3   B   M
##                     1 357 210
##                     2   0   2
```

Combining Methods

```
wisc.pr.hclust <- hclust(data.dist, method='ward.D2')
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

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

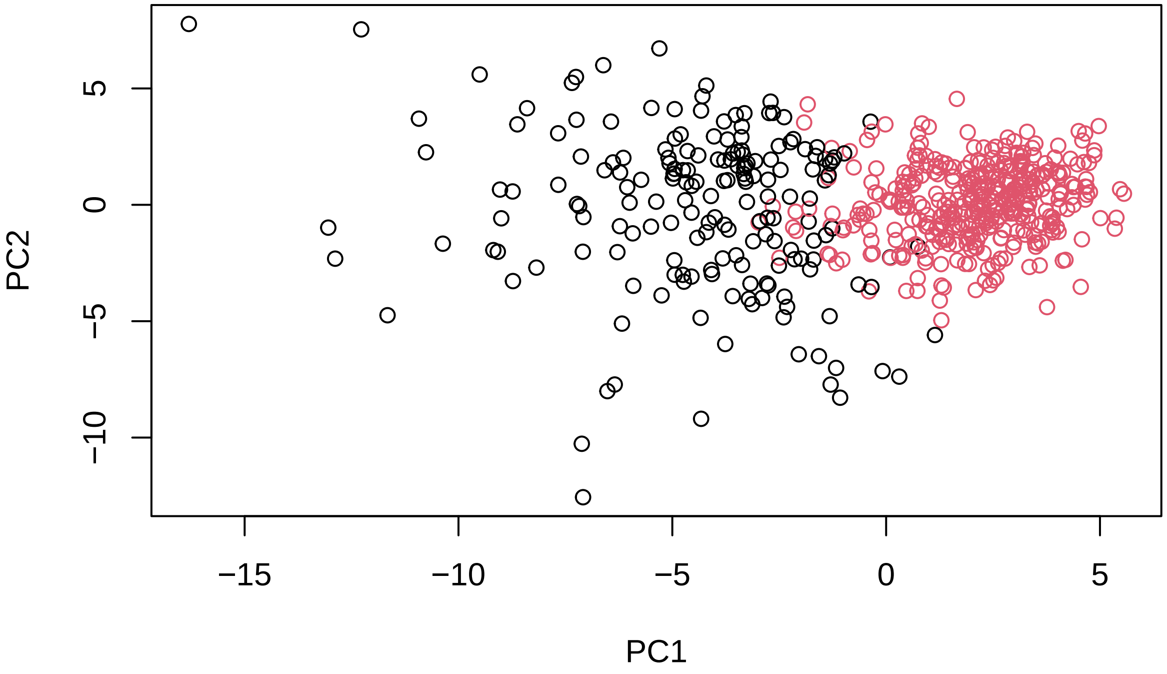
```
plot(wisc.pr.hclust)
```



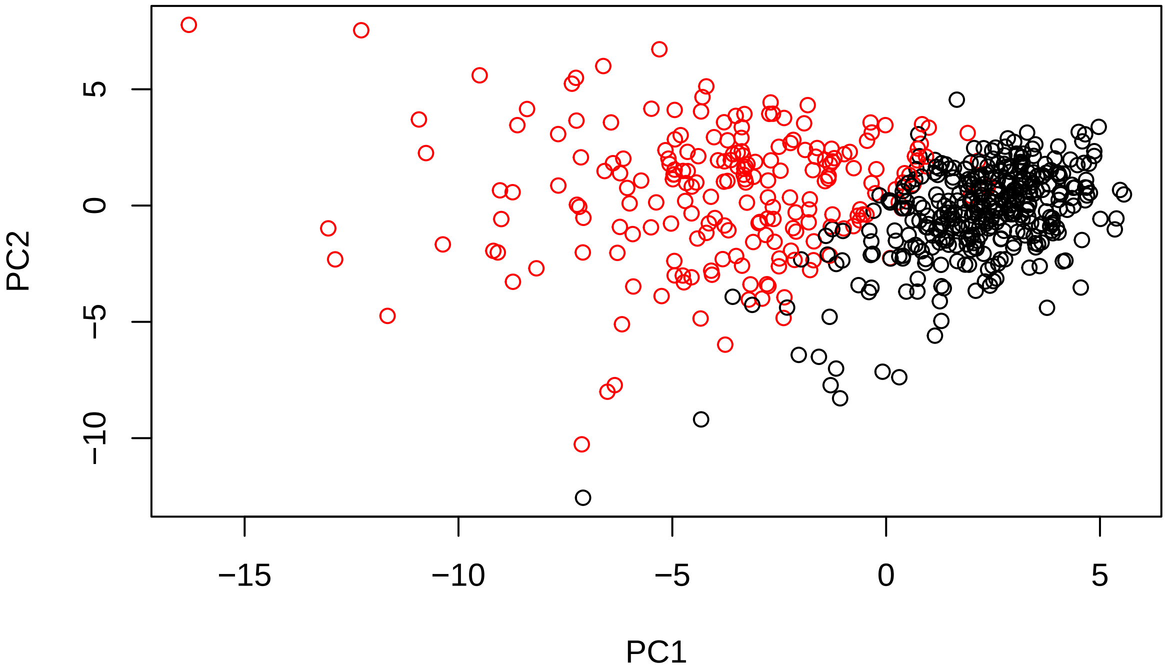
```
table(grps, diagnosis)
```

```
##      diagnosis
## grps    B   M
##     1  20 164
##     2 337  48
```

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



```
#Now the colors are based on diagnosis rather than the cluster they belong to
plot(wisc.pr$x[,1:2], col=ifelse(diagnosis == 'M','red','black'))
```



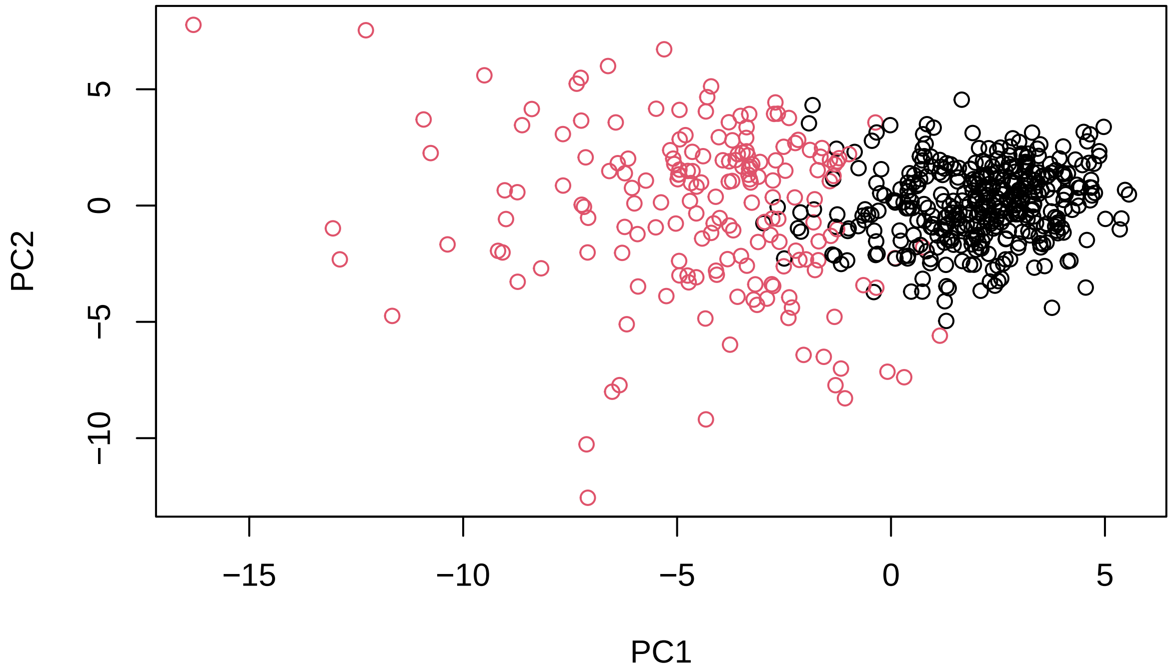
```r
g <- as.factor(grps)
levels(g)
```

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

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

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

```r
# Plot using our re-ordered factor
plot(wisc.pr$x[,1:2], col=g)
```



Question 15

with 4 clusters, the model using just the first 7 PCs does not separate out the clusters very well.

```
#Use the distance along the first 7 PCs for clustering
dist_first7 <- dist(wisc.pr$x[,1:7])
wisc.pr.hclust <- hclust(dist_first7, method="ward.D2")

#Separate into 2 clusters
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
```

```
#When separated into 4 clusters:
wisc.pr.hclust.clusters4 <- cutree(wisc.pr.hclust, k=4)
table(wisc.pr.hclust.clusters4, diagnosis)
```

```
##                         diagnosis
## wisc.pr.hclust.clusters4   B   M
##                        1   0  45
##                         2   2  77
##                         3  26  66
##                         4 329  24
```

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

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