# class08

Maddie Maslyar (PID: 69042845)

## 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.

## Data import

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

Make sure we do not include sample ID or diagnosis in further analysis

```
# Save diagnosis outside of data frame
diagnosis <- as.factor(wisc.df$diagnosis)
# Remove diagnosis from data frame in new data frame
wisc.data <- wisc.df[,-1]
head(wisc.data)
```

```
         radius_mean texture_mean perimeter_mean area_mean smoothness_mean
842302         17.99        10.38         122.80    1001.0         0.11840
842517         20.57        17.77         132.90    1326.0         0.08474
84300903       19.69        21.25         130.00    1203.0         0.10960
84348301       11.42        20.38          77.58     386.1         0.14250
84358402       20.29        14.34         135.10    1297.0         0.10030
843786         12.45        15.70          82.57     477.1         0.12780
         compactness_mean concavity_mean concave.points_mean symmetry_mean
842302            0.27760         0.3001             0.14710        0.2419
842517            0.07864         0.0869             0.07017        0.1812
```

```
84300903          0.15990         0.1974             0.12790        0.2069
84348301          0.28390         0.2414             0.10520        0.2597
84358402          0.13280         0.1980             0.10430        0.1809
843786            0.17000         0.1578             0.08089        0.2087
         fractal_dimension_mean radius_se texture_se perimeter_se area_se
842302                  0.07871    1.0950     0.9053        8.589  153.40
842517                  0.05667    0.5435     0.7339        3.398   74.08
84300903                0.05999    0.7456     0.7869        4.585   94.03
84348301                0.09744    0.4956     1.1560        3.445   27.23
84358402                0.05883    0.7572     0.7813        5.438   94.44
843786                  0.07613    0.3345     0.8902        2.217   27.19
         smoothness_se compactness_se concavity_se concave.points_se
842302        0.006399        0.04904      0.05373           0.01587
842517        0.005225        0.01308      0.01860           0.01340
84300903      0.006150        0.04006      0.03832           0.02058
84348301      0.009110        0.07458      0.05661           0.01867
84358402      0.011490        0.02461      0.05688           0.01885
843786        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
```

Q1. How many observations are in this dataset?

```
ncol(wisc.data)
```

```
[1] 30
```

A1. here are 30 observations.

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 
```

A2. There are 212 observations with a malignant diagnosis.

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

```
#grep searches for pattern matches
length(grep("_mean", colnames(wisc.data)))
```

```
[1] 10
```

A3. There are 10 variables with the suffix "_mean".

## Principal Component Analysis

The main function in base R for PCA is called `prcomp()`

Scaling data ensures that each feature contributes equally to the analysis, preventing variables with larger scales from dominating the principal components

Make sure to set `prcomp(x, scale = TRUE)`.

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

Q5. How many principal components (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?

```r
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
```

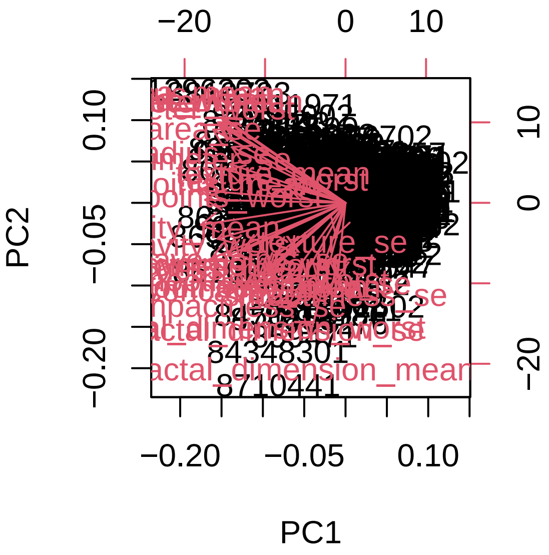
A4. 44.27% of the original variance is captured in PC1.

A5. Three PCs can describe greater than 70% of the original variance in the data.

A6. Seven PCs can describe greater than 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?

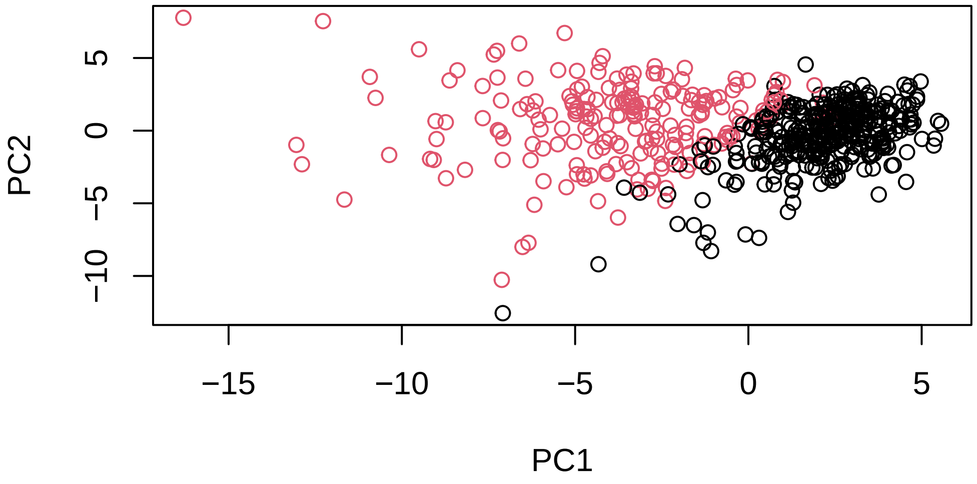
```r
biplot(wisc.pr)
```



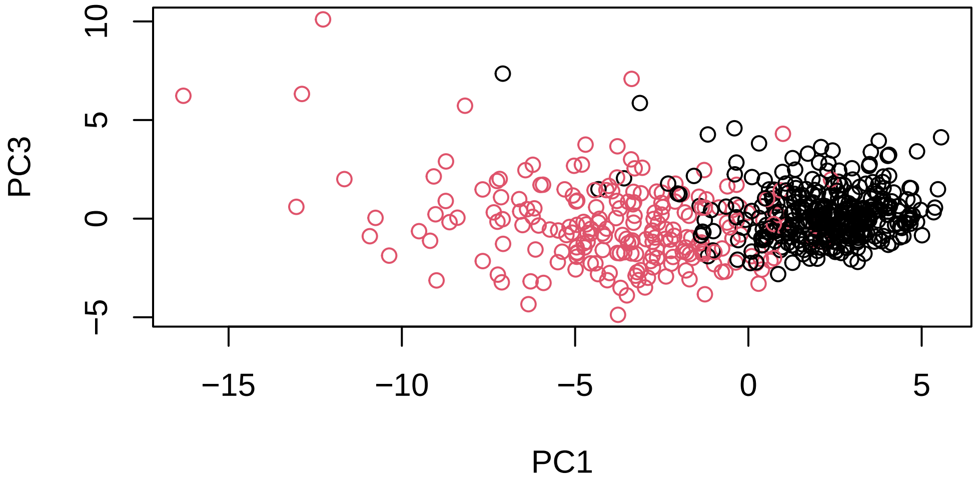
A7. I am unable to interpret the plot due to the format and size of the data points.

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

```r
plot(wisc.pr$x, col = diagnosis ,
     xlab = "PC1", ylab = "PC2")
```



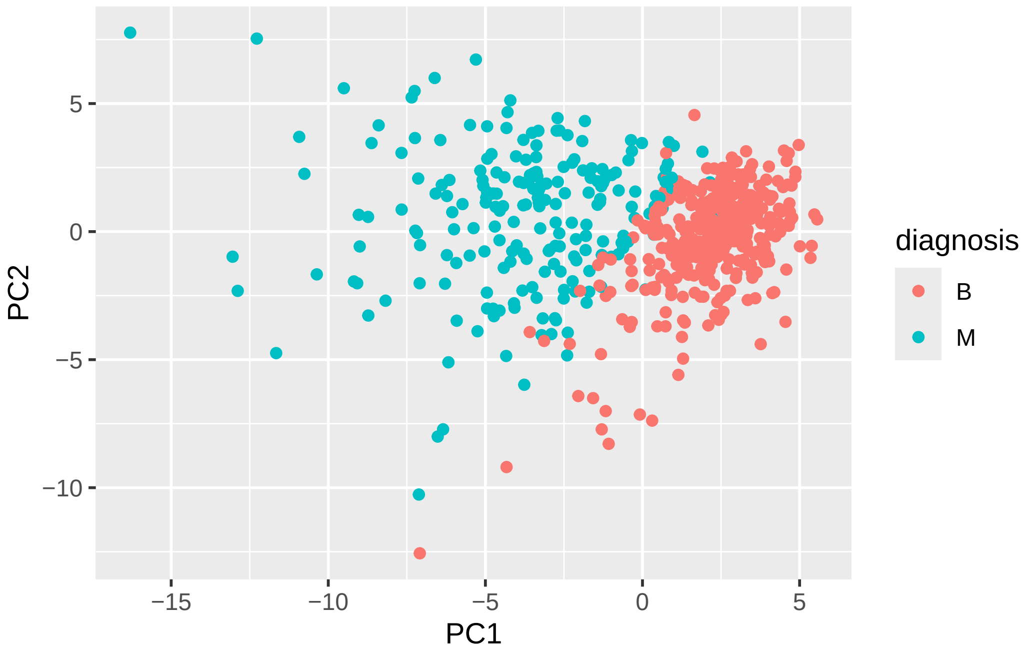
```r
plot(wisc.pr$x[, -2 ], col = diagnosis,
     xlab = "PC1", ylab = "PC3")
```



A8. The plots look similar. In both, the malignant and benign tumors cluster into two groups.

Make our main result figure - the “PC plot” or “score plot”

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



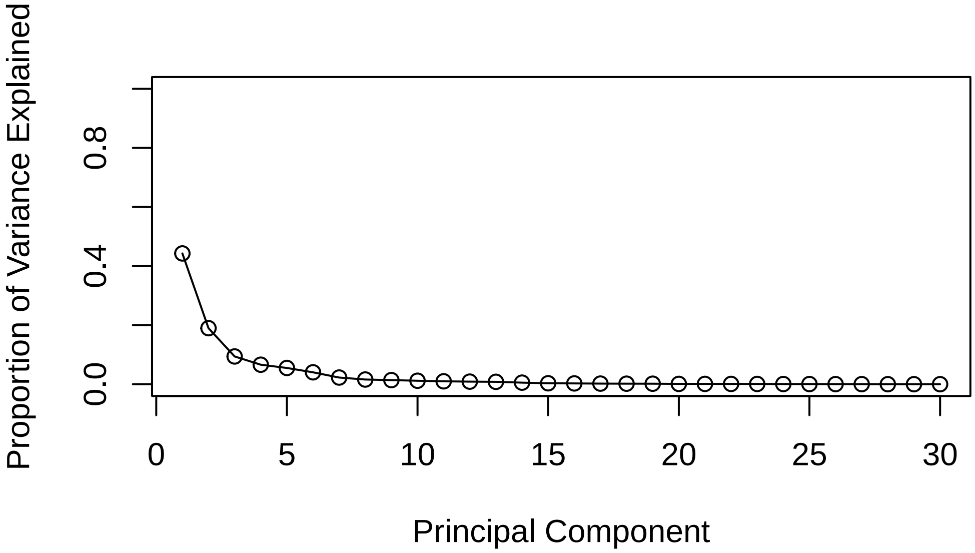
The Malignant and benign tumors separate into two clusters.

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

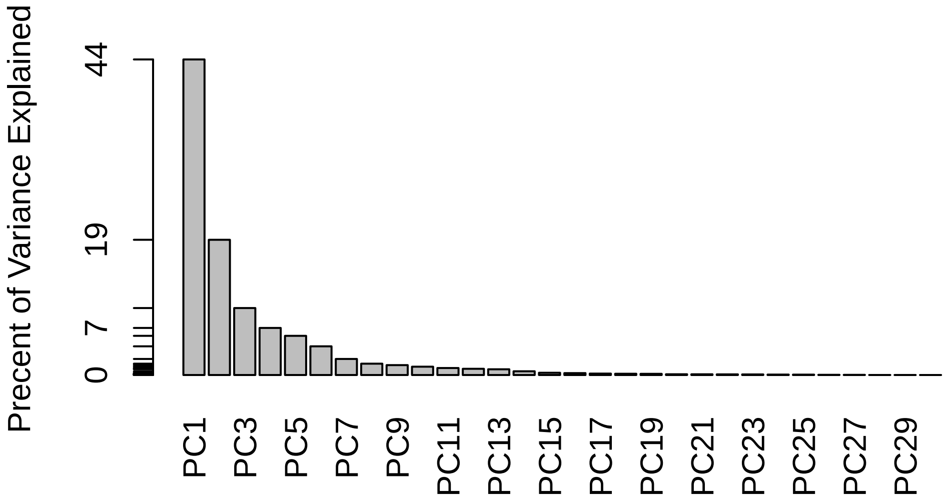
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
# Variance explained by each principal component: pve
pve <- pr.var / sum(pr.var)

# 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 )
```

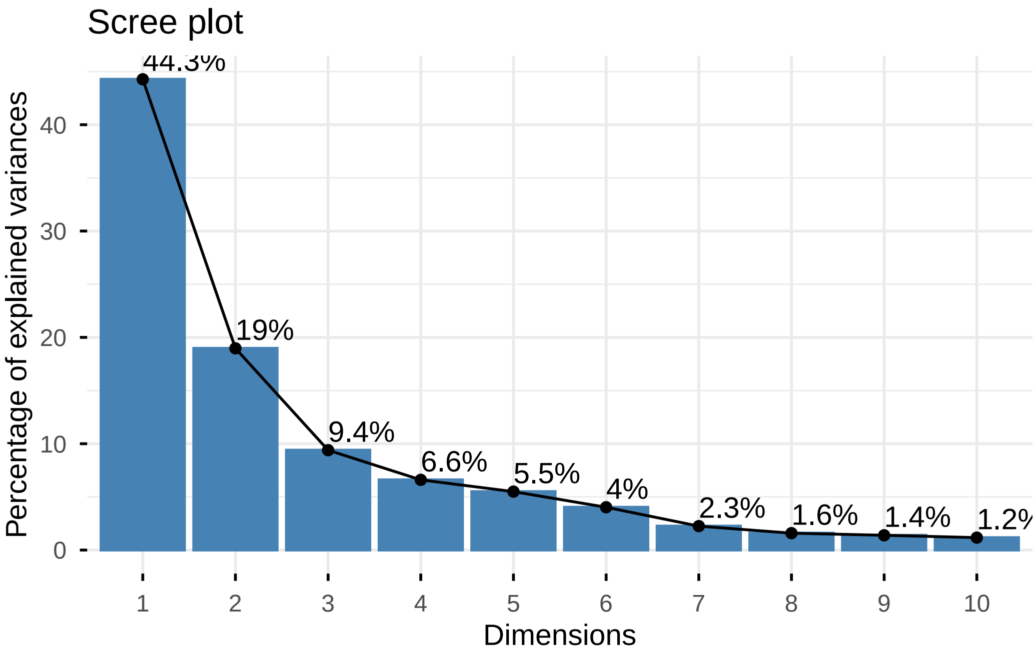


```
## ggplot based graph
#install.packages("factoextra")
library(factoextra)
```

```
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`.
```



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.

```
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
```

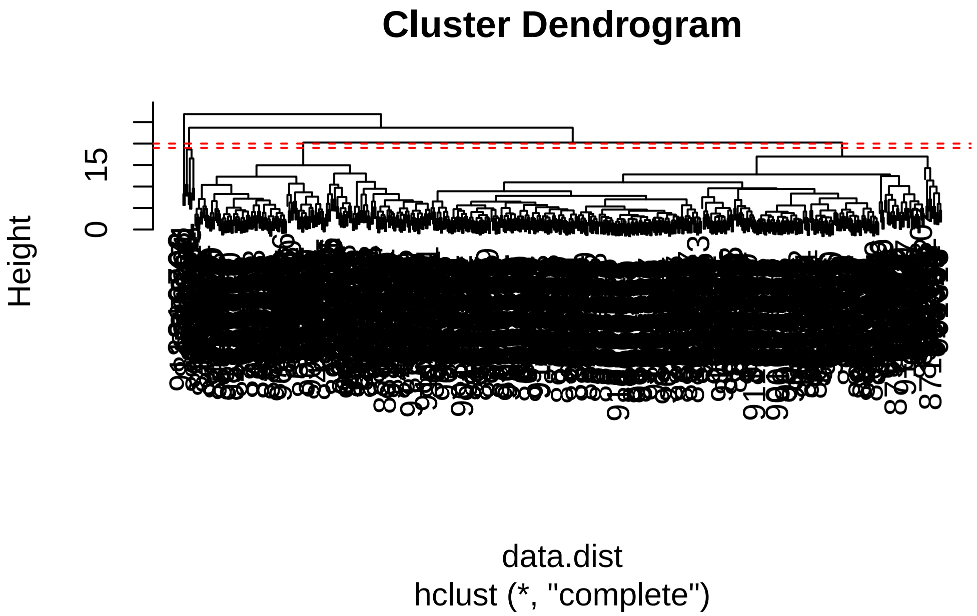
A9. The component of the loading vector is -0.26085376.

### 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, method = "complete")
```

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

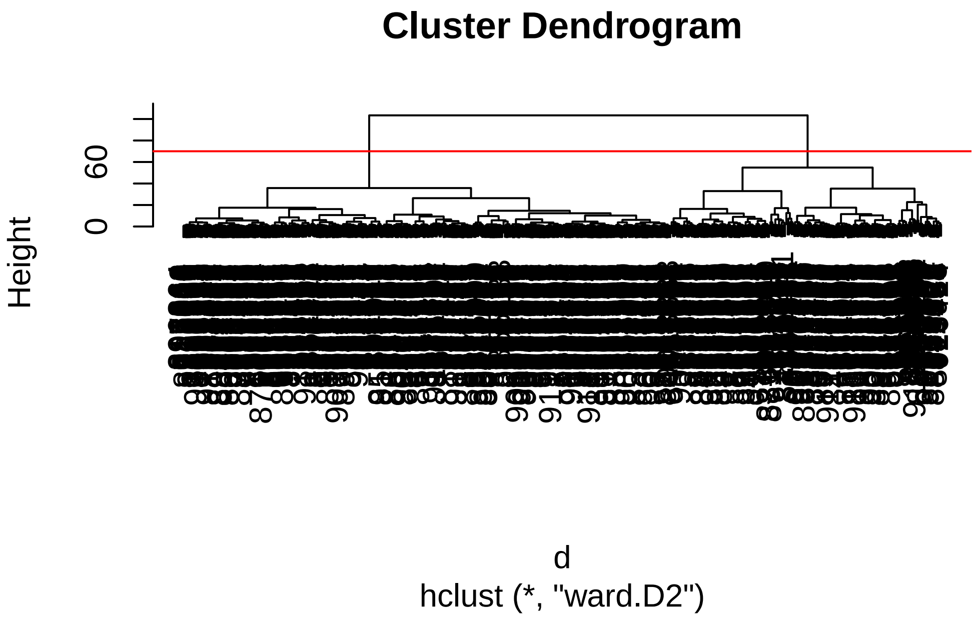
```
plot(wisc.hclust)
abline(h = 19:20, col="red", lty=2)
```



A10: The clustering model has four clusters between height 19-20.

## 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 cross-table

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

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

True positive: 179 False positive: 24 True negative: 333 False negative: 33

Sensitivity: TP/(TP+FN)

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

```
                       diagnosis
wisc.pr.hclust.clusters   B   M
                      1  24 179
                      2 333  33
```

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

```
(333+179)/(24+179+333+33)
```

```
[1] 0.8998243
```

A13. The new model identifies the two diagnoses accurately 89.98% of the time.

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.

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

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

```
(343+165)/(12 + 165 + 2 + 5 + 343 + 40 + 2)
```

```
[1] 0.8927944
```

A14. The hierarchical clustering models successfully predict 89.28% of diagnoses. This is relatively similar to the new model.