Analysis of Human Breast Cancer Cells

Mady Welch

Exploratory data analysis

Download and import data:

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

Lets make a new data.frame that omits the first column:

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

We can put the data from the first column into a separate vector

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

Q1. How many observations are in this dataset?

```
nrow(wisc.data)
```

```
[1] 569
```

- There are 569 rows in wisc.data

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

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

```
  B   M
357 212
```

- 212 observations have a malignant diagnosis

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

```
colnames(wisc.data)
```

```
 [1] "radius_mean"             "texture_mean"
 [3] "perimeter_mean"          "area_mean"
 [5] "smoothness_mean"         "compactness_mean"
 [7] "concavity_mean"          "concave.points_mean"
 [9] "symmetry_mean"           "fractal_dimension_mean"
[11] "radius_se"               "texture_se"
[13] "perimeter_se"            "area_se"
[15] "smoothness_se"           "compactness_se"
[17] "concavity_se"            "concave.points_se"
[19] "symmetry_se"             "fractal_dimension_se"
[21] "radius_worst"            "texture_worst"
[23] "perimeter_worst"         "area_worst"
[25] "smoothness_worst"        "compactness_worst"
[27] "concavity_worst"         "concave.points_worst"
[29] "symmetry_worst"          "fractal_dimension_worst"
```

Now we can use `grep()` to find the column names that contain _mean

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

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

And now we can use `length()` to find how many matches there are

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

```
[1] 10
```

- 10 are suffixed with _mean

Principal Component Analysis

Check the column means and standard deviations to check if the data should be scaled

```
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
```

We need to scale with `scale = TRUE` argument in `prcomp()`

```
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)?

- Proportion of variance for PC1 = 44.27%

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

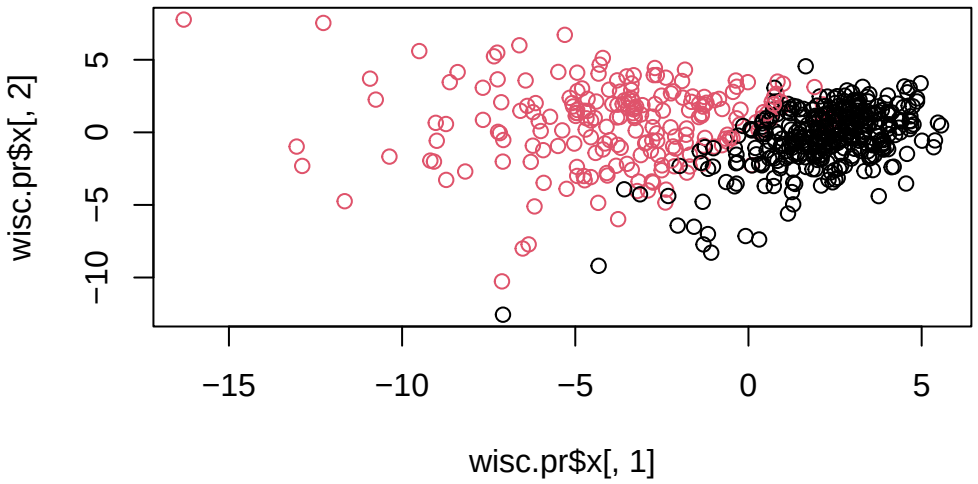
- 3 PCs are required to get at least 70% of the original variance.

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

- 7 PCs are required to get at least 90% of the original variance.

Lets make a PC plot (aka "score plot" or "PC1 vs PC2")

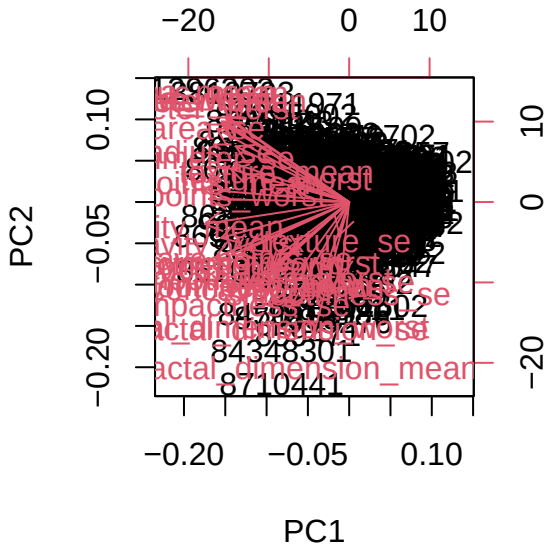
```
plot(wisc.pr$x[, 1], wisc.pr$x[, 2], col = diagnosis)
```



Interpreting PCA Results

Create a biplot:

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

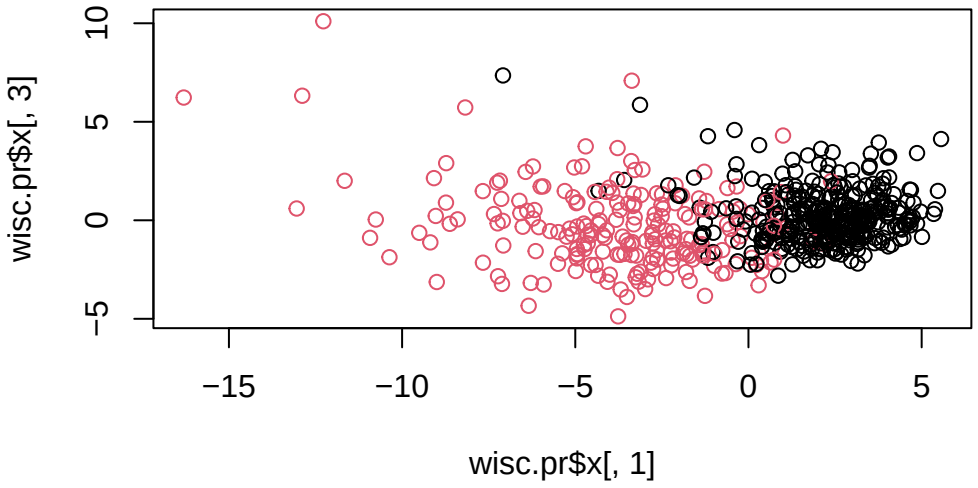


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

- This plot is very difficult to understand since it is all smushed together in a small plot and every point is labeled.

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)
```



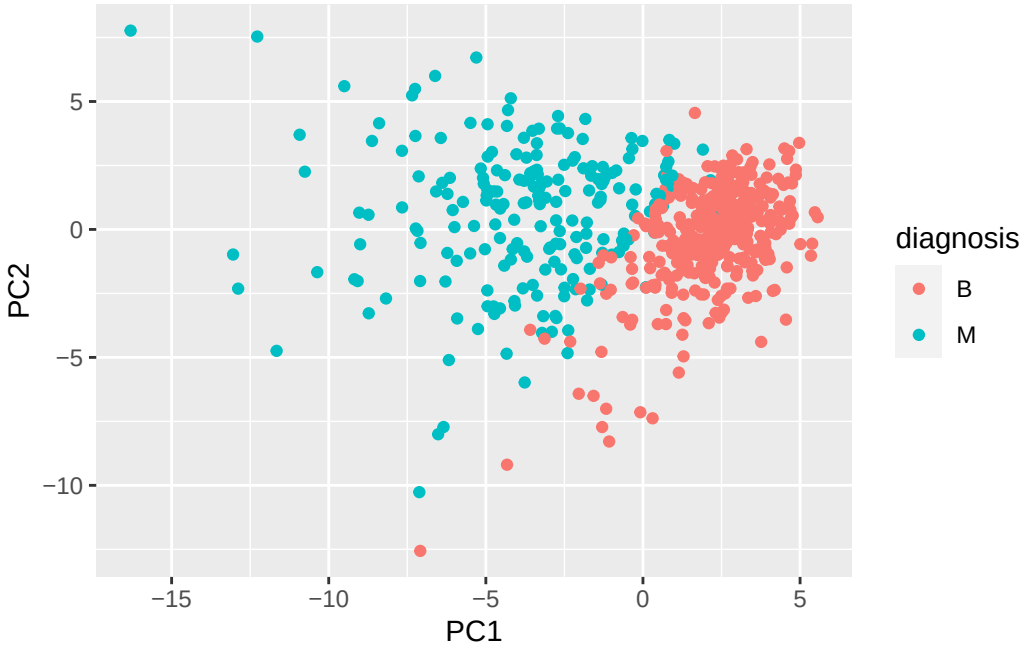
- The plots for PR1 vs PR2 and PR1 vs PR3 are very similar. They both have two groups that are close together around (0,0).

Now lets make a data.frame to use ggplot:

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

Now load ggplot2 package and make a scatterplot colored by diagnosis:

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



Variance explained

Calculate the variance of each component.

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

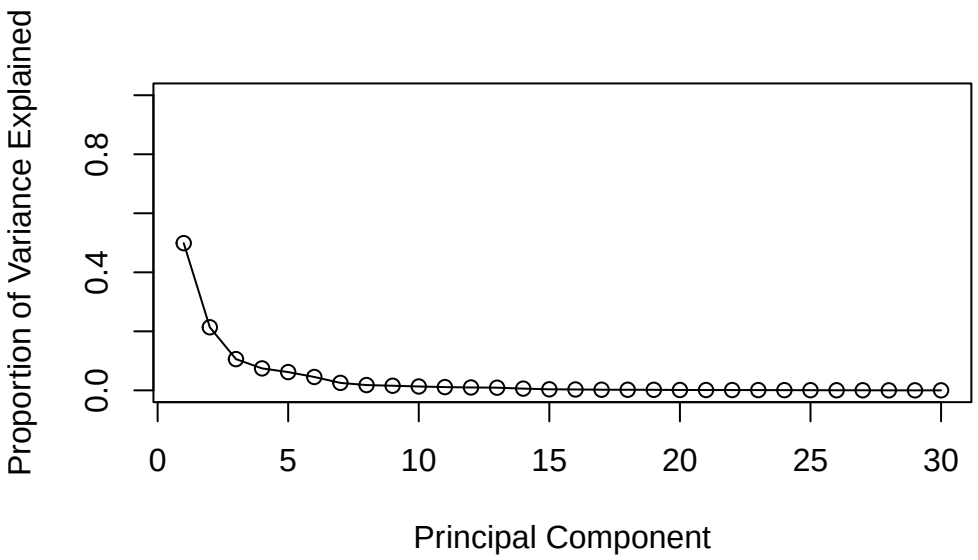
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

Calculate the variance explained by each principal component by dividing by the total variance explained of all principal components.

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

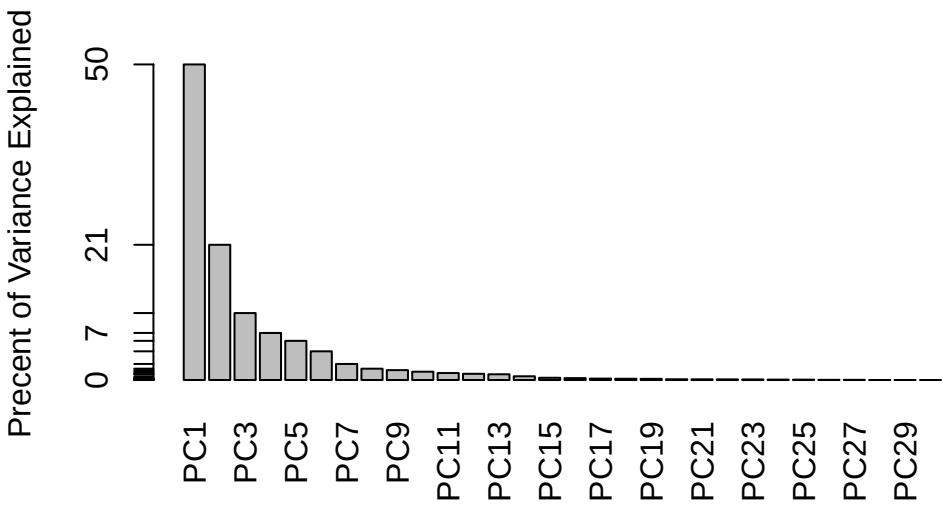
Plot the variance explained for each principal component.

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



Alternative scree plot of the same data:

```
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 )
```



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?

- It would be `concave.points_mean(pr.var[1])`

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

```
pve[1] + pve[2] + pve[3]
```

```
[1] 0.8183569
```

- 3 PCs are required to explain 80% of the variance.

Hierarchical Clustering

First scale the wisc.data data and assign to data.scaled Then calculate the distances between all pairs of observations in the new scaled dataset and assign to data.dist

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

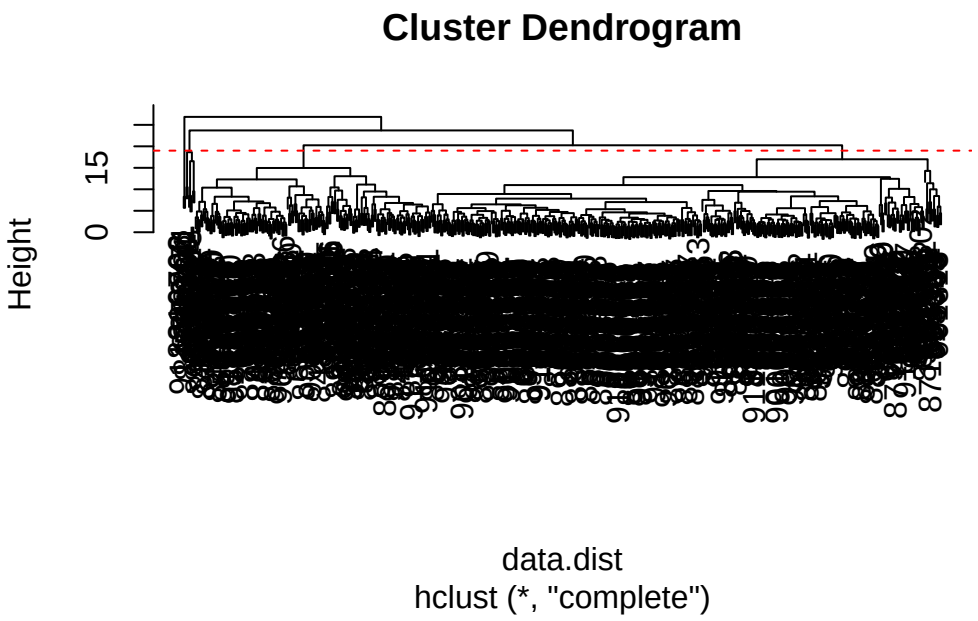
Create a hierarchical clustering model using complete linkage. Assign to wisc.hclust

```
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(h = 19, col = "red", lty = 2)
```



- Height = 19

Select 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, k = 4)
```

We can use the table() function to compare the cluster membership to the actual diagnoses.

```
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?

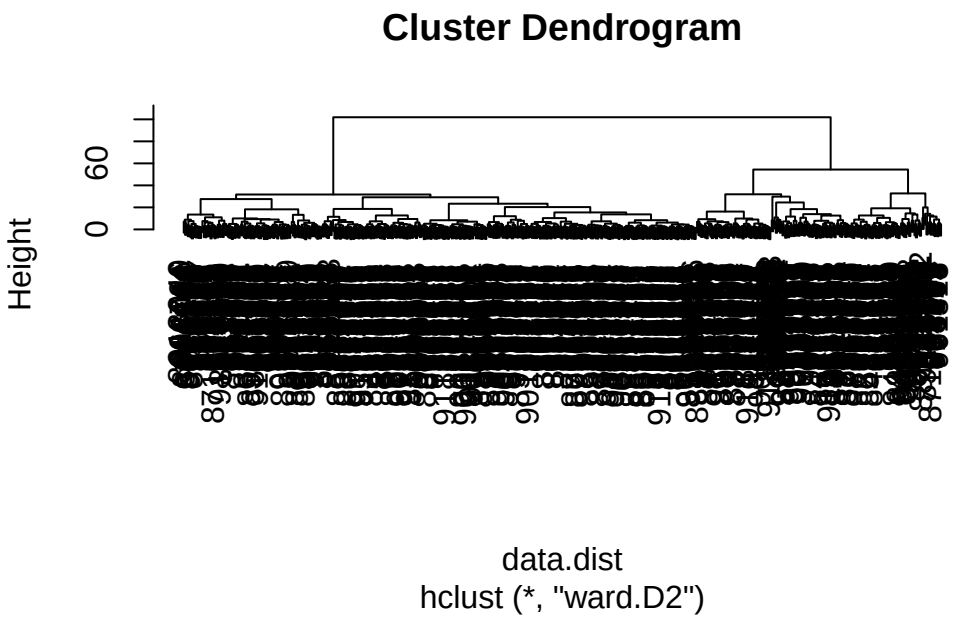
- You can find other matches when cutting into different numbers of clusters, but there is no way to know what number of clusters is best to use.

Using different methods

There are number of different "methods" we can use to combine points during the hierarchical clustering procedure. These include "single", "complete", "average", and "ward.D2"

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

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

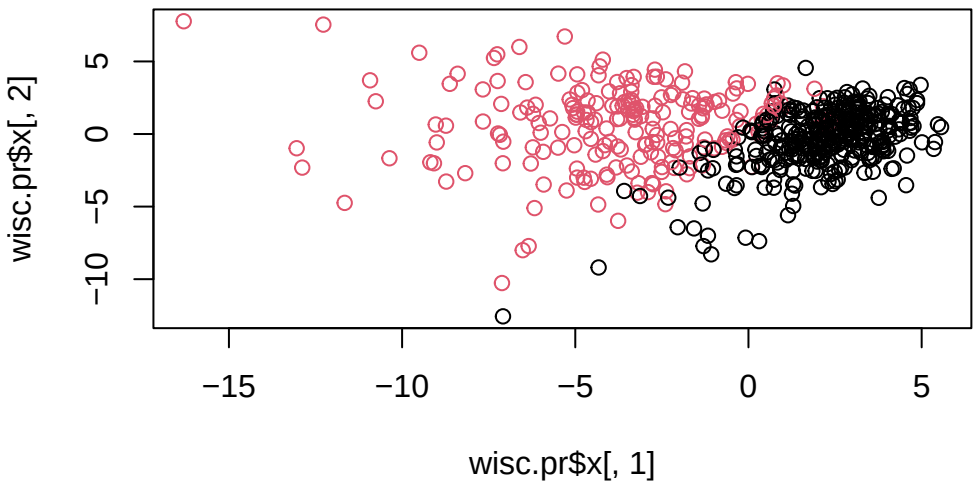


- The "ward.D2" method is my favorite because it gives a very clear cutoff point for clustering.

Combine PCA with Clustering

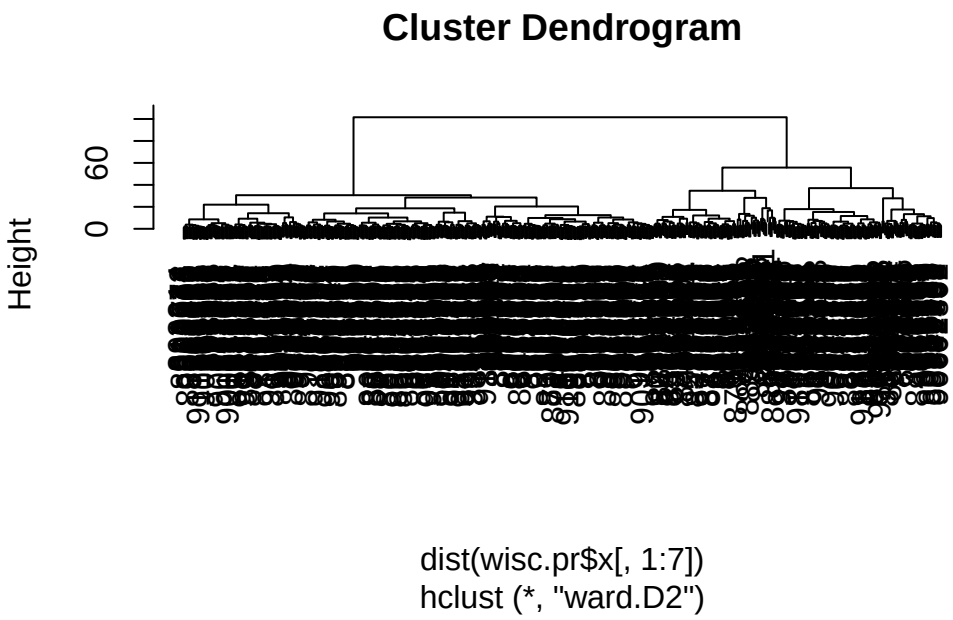
I want to cluster in "PC space"

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



The `hclust()` function wants a distance matrix as input…

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



Find my cluster membership vector with `cutree()`.

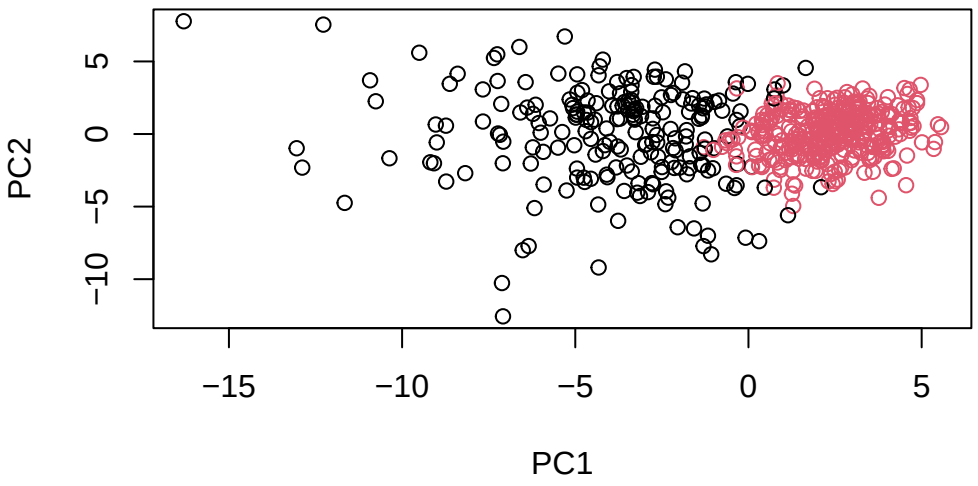
```
grps <- cutree(wisc.pr.hclust, k=2)

table(grps)
```

```
grps
  1   2
216 353
```

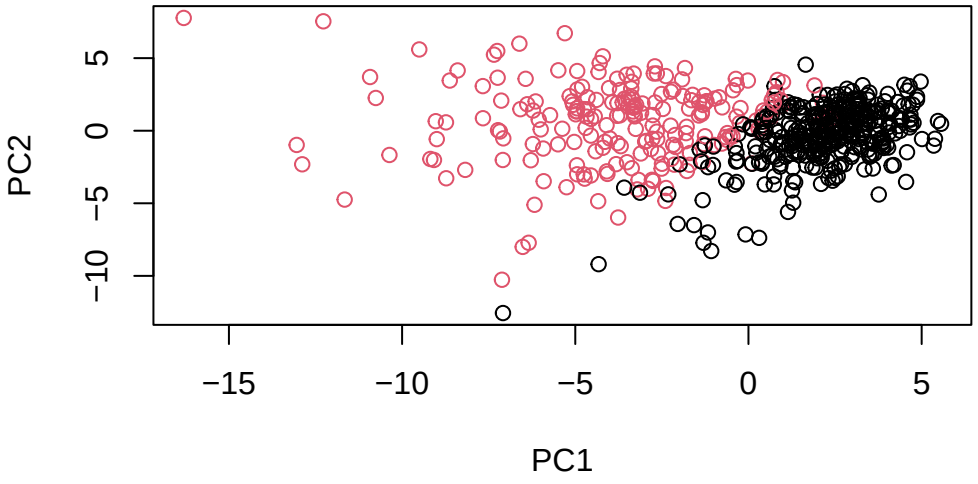
```
table(diagnosis, grps)
```

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

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




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



Use the distance along the first 7 PCs for clustering

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

Cut this hierarchical clustering model into 2 clusters and assign the results to wisc.pr.hclust.clusters

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

Using table(), compare the results from your new hierarchical clustering model with the actual diagnoses.

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

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

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

- The new model does a pretty decent job of separating the two diagnoses. However, a somewhat high proportion of malignant diagnoses are seen in cluster 2, meaning it would be difficult to tell whether or not a sample is malignant or benign based on the plot alone.

> 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.hclust.clusters, diagnosis)
```

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

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

```
  B   M
357 212
```

- The hierarchical clustering model did a pretty good job at separating the diagnoses, but there is a somewhat high proportion of malignant diagnoses in clusters 2, 3, and 4 that would make it difficult to tell if a sample is benign or malignant from looking at the data alone.

Sensitivity/Specificity

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

Sensitivity:

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

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

```
165/212
```

```
[1] 0.7783019
```

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

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

```
188/212
```

```
[1] 0.8867925
```

- `wisc.pr.hclust.clusters` had the highest sensitivity.

Specificity:

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

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

```
343/357
```

```
[1] 0.9607843
```

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

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

```
329/357
```

```
[1] 0.9215686
```

- `wisc.hclust.clusters` had the highest specificity.