Class 08 Mini Project

Natalie Ogg A91030809

NOTE: *I somehow messed it up a bit in the beginning so I use* `diagnosis1` *instead of* `diagnosis` *sometimes, but it's the same thing.*

Today we are applying machine learning to breast cancer biopsy data from fine needle aspiration (FNA).

First I put the .csv file into the class08 file on my computer. Then I call it up and rename it:

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

```
         diagnosis radius_mean texture_mean perimeter_mean area_mean
842302           M       17.99        10.38         122.80    1001.0
842517           M       20.57        17.77         132.90    1326.0
84300903         M       19.69        21.25         130.00    1203.0
84348301         M       11.42        20.38          77.58     386.1
84358402         M       20.29        14.34         135.10    1297.0
843786           M       12.45        15.70          82.57     477.1
         smoothness_mean compactness_mean concavity_mean concave.points_mean
842302           0.11840          0.27760         0.3001             0.14710
842517           0.08474          0.07864         0.0869             0.07017
84300903         0.10960          0.15990         0.1974             0.12790
84348301         0.14250          0.28390         0.2414             0.10520
84358402         0.10030          0.13280         0.1980             0.10430
843786           0.12780          0.17000         0.1578             0.08089
         symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
842302          0.2419                0.07871    1.0950     0.9053        8.589
842517          0.1812                0.05667    0.5435     0.7339        3.398
84300903        0.2069                0.05999    0.7456     0.7869        4.585
```

```
84348301        0.2597                0.09744    0.4956     1.1560         3.445
84358402        0.1809                0.05883    0.7572     0.7813         5.438
843786          0.2087                0.07613    0.3345     0.8902         2.217
         area_se smoothness_se compactness_se concavity_se concave.points_se
842302    153.40      0.006399        0.04904      0.05373           0.01587
842517     74.08      0.005225        0.01308      0.01860           0.01340
84300903   94.03      0.006150        0.04006      0.03832           0.02058
84348301   27.23      0.009110        0.07458      0.05661           0.01867
84358402   94.44      0.011490        0.02461      0.05688           0.01885
843786     27.19      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
```

Now we want to omit the first column, which is the diagnosis.

```
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
```

We are saving the diagnosis column for later, as a factor.

```
diagnosis1 <- as.factor(wisc.df$diagnosis)
```

Exploring the data!

Q1. Number of observations:

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

```
[1] 569
```

Q2. How many malignant?

Use `table` to measure number of each character in the set:

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

```
  B   M 
357 212 
```

Other method, ask for sum where values equal M:

```r
sum(wisc.df$diagnosis == "M")
```

```
[1] 212
```

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

`grep` returns the positions of matching variable names:

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

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

Assign that vector to `mean_vars` for mean variable, then use length.

```r
mean_vars <- grep("_mean$", colnames(wisc.data))
length(mean_vars)
```

```
[1] 10
```

PCA

We need to scale our input data before PCA because the columns are measured in very different units with different means and variances. We set `scale=TRUE` argument to `prcomp()`.

`scale()` sets means to 0 and standard deviations to 1.

```
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)?

0.4427 (from table above)

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

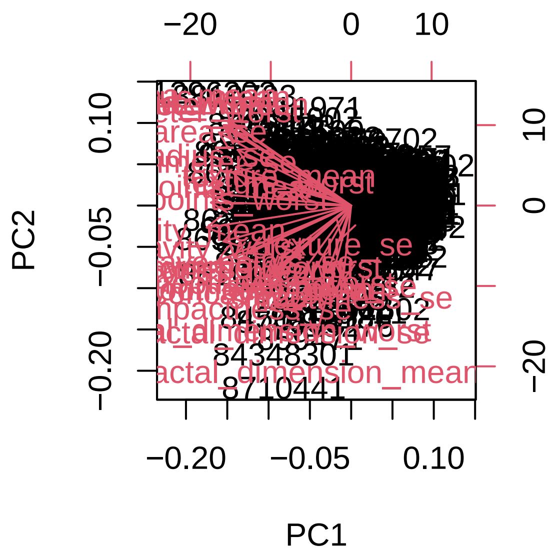
3 PCs

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

7 PCs

Interpreting PCA results

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

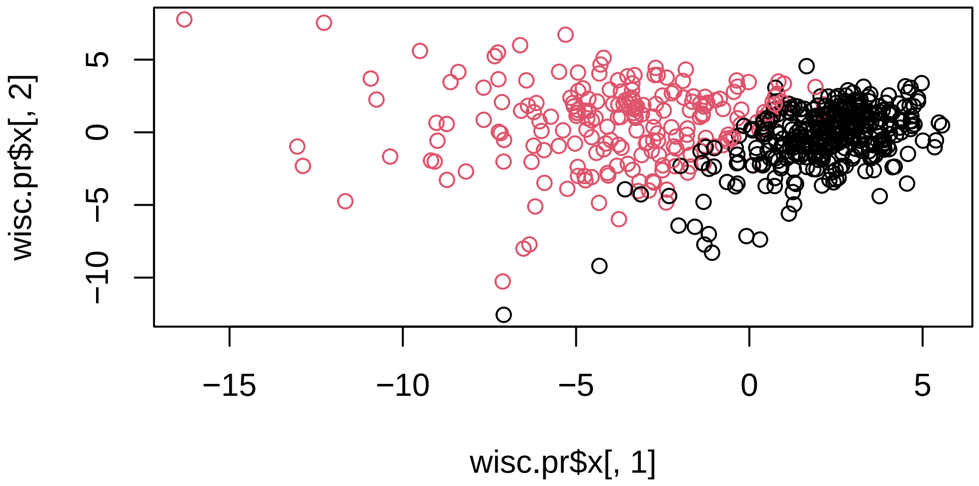


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

What stands out is everything! It is very difficult to understand because everything is dense and overlapping, with long names instead of just points.

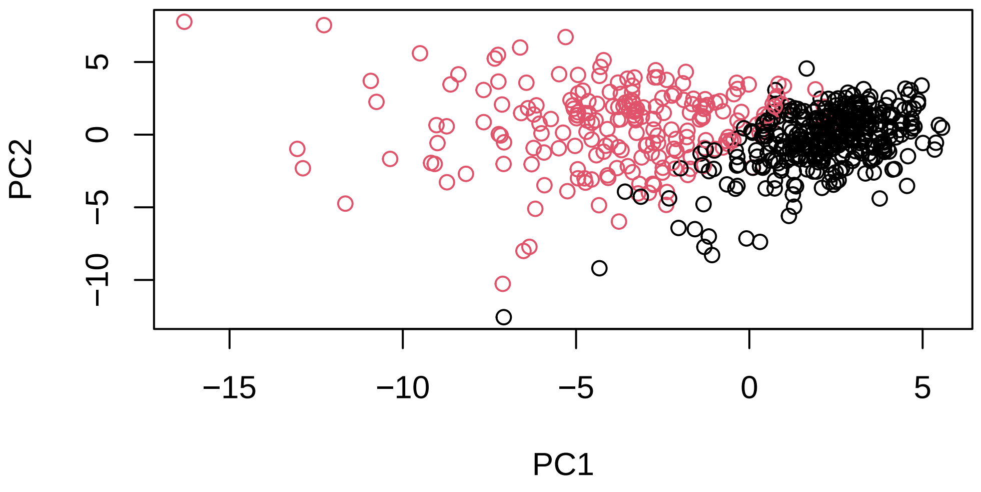
Now we plot our PCA data and color by diagnosis:

```
plot(wisc.pr$x[,1], wisc.pr$x[,2], col=diagnosis1)
```



To add labels:

```
plot(wisc.pr$x[,1], wisc.pr$x[,2], col=diagnosis1, xlab = "PC1", ylab = "PC2")
```

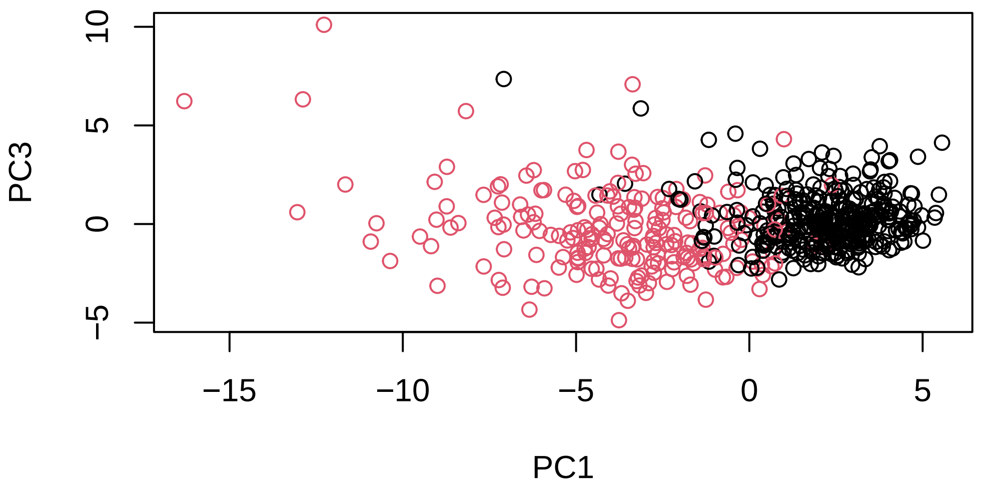


We can see that the diagnoses are starkly separated on the plot, which is notable. The idea of PCA plots here is that more similar cells will be clustered. It's a method for compressing a lot of data into something that represents the essence of the data.

You can create a point to represent a cluster of data in the PCA, for example (from class) using the original data and the PCA data for all rows to get a value for each column.

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

```r
plot(wisc.pr$x[, 1 ], wisc.pr$x[, 3 ], col = diagnosis1,
     xlab = "PC1", ylab = "PC3")
```



With PC3, the points appear less clustered than in PC2, and the M and B diagnoses overlap more.

Create a data.frame for ggplot

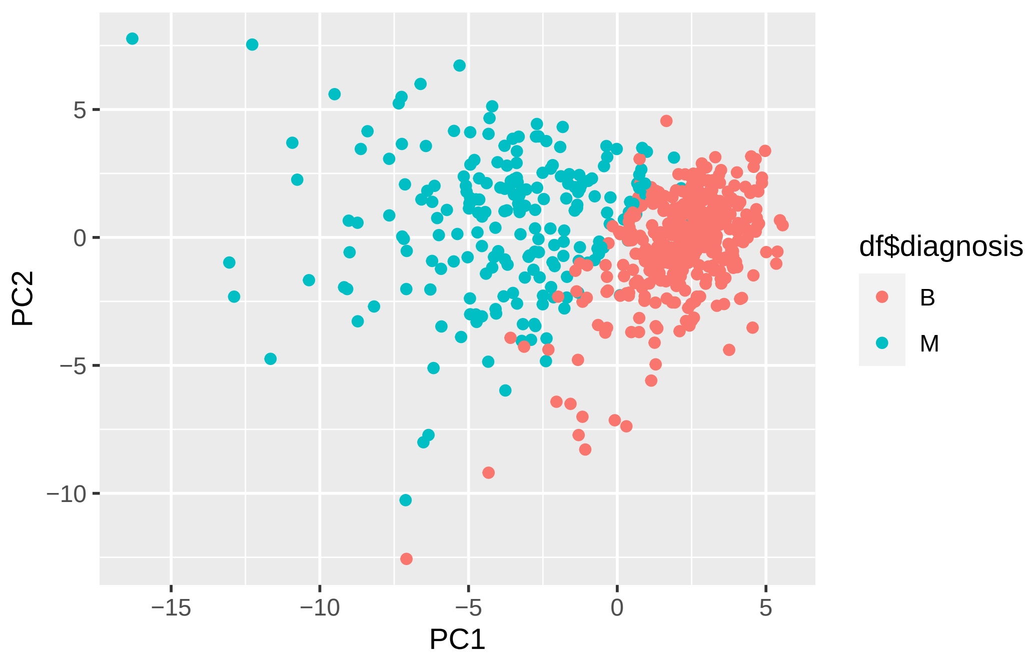
```r
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis1
```

Load the ggplot2 package

```r
library(ggplot2)
```

Make a scatter plot:

```r
ggplot(df) +
  aes(PC1, PC2, col=df$diagnosis) +
  geom_point()
```



Here we use SD squared to calculate the variance of each PCA component:

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

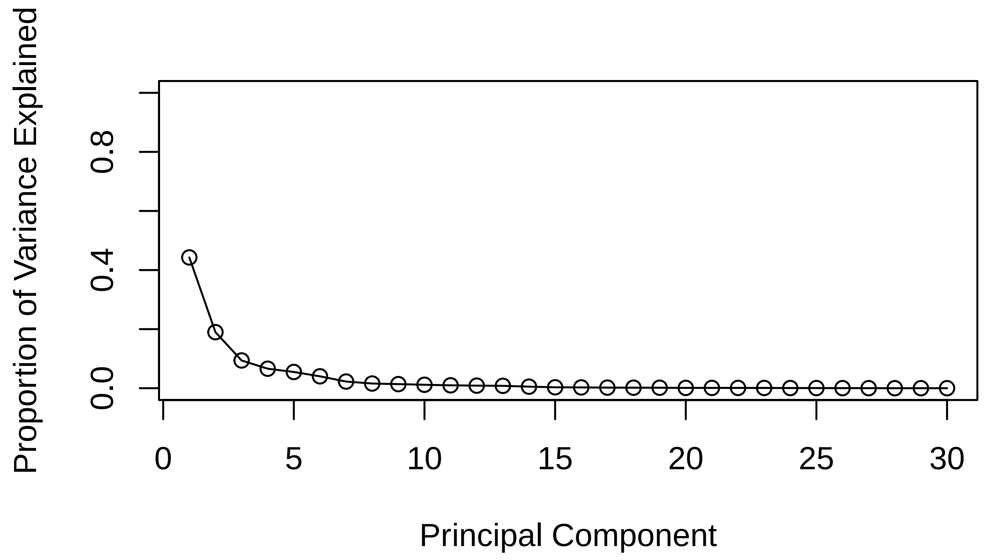
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

Variance explained by each principal component: pve

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

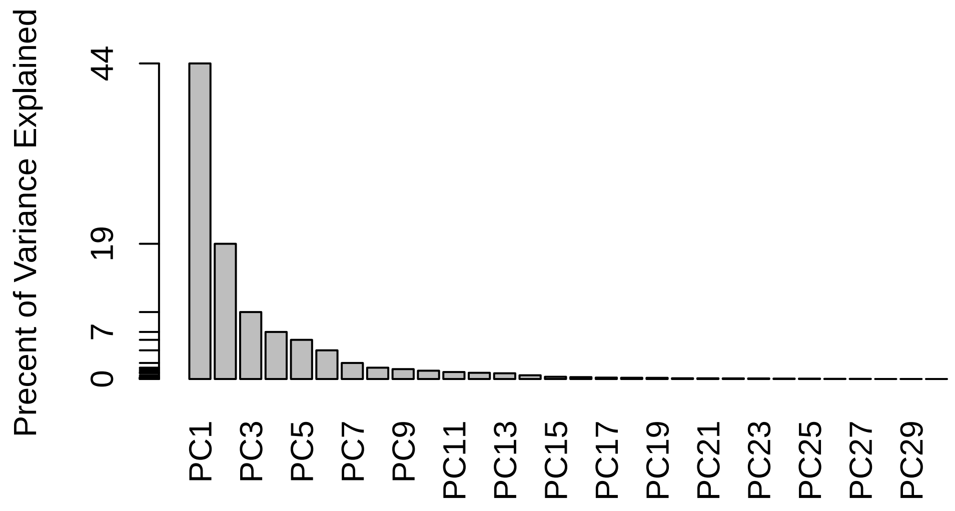
Plot variance explained for each principal component

```
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 )
```



```r
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
```

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?

This is just asking for the value of `wisc.pr$rotation` row `concave.points_mean`

```
wisc.pr$rotation["concave.points_mean",1]
```

```
[1] -0.2608538
```

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

5. PCA 1-5 explain 80% of the variance.

```
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
```

```
sum(pve[1:5])
```

```
[1] 0.8473427
```

Hierarchical Clustering

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

data.scaled <- _____(wisc.data)

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

Calculate the (Euclidean) distances between all pairs of observations in the new scaled dataset and assign the result to data.dist.

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

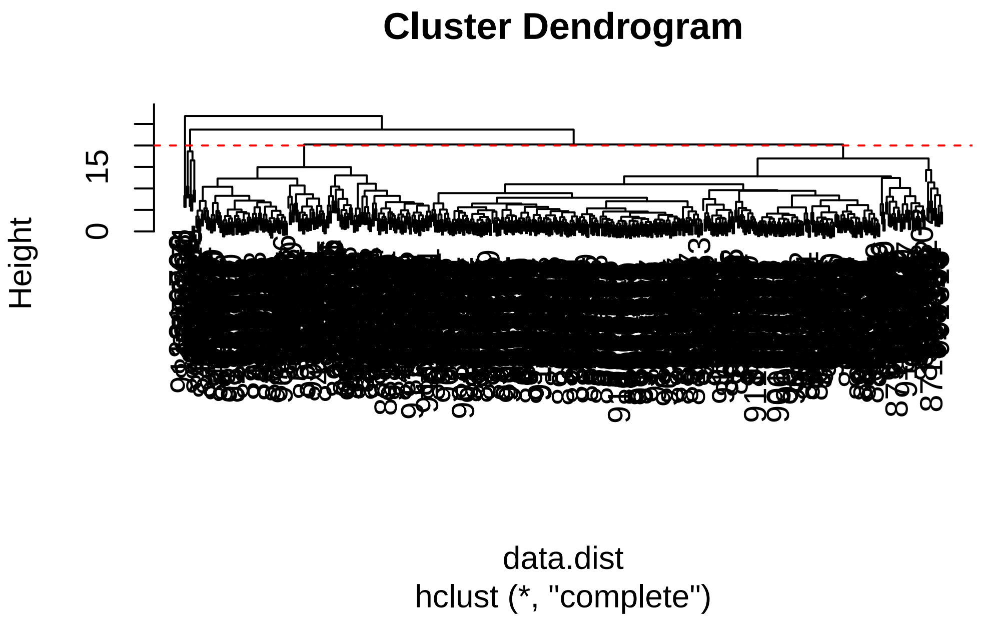
Create a hierarchical clustering model using complete linkage. Manually specify the method argument to hclust() and assign the results to wisc.hclust.

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

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

I used h = 20

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



Selecting number of clusters

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

table(wisc.hclust.clusters, diagnosis1)
```

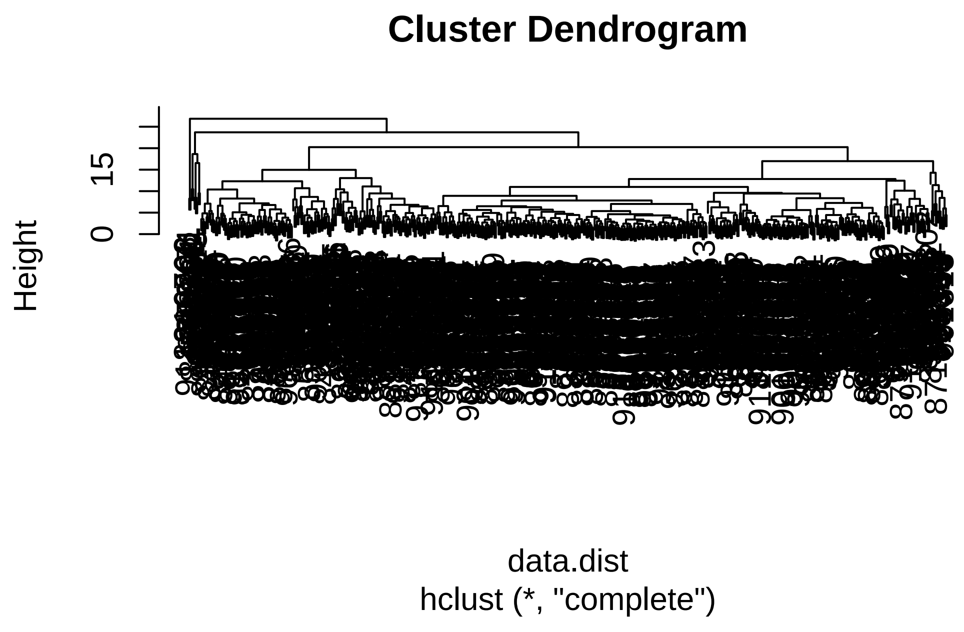
```
                    diagnosis1
wisc.hclust.clusters   B   M
                   1  12 165
                   2   2   5
                   3 343  40
                   4   0   2
```

Q12. Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10?

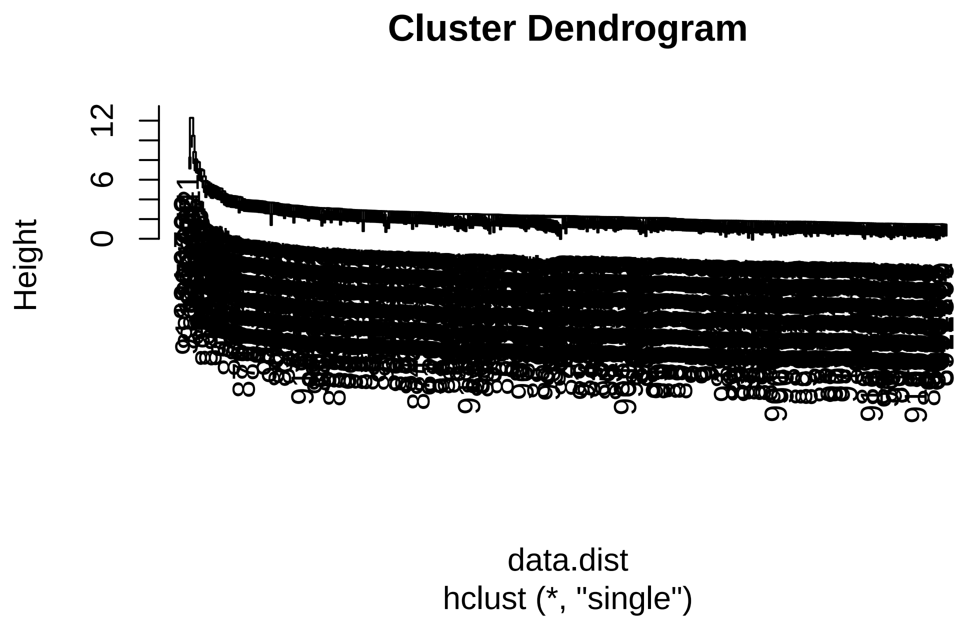
The best match is k=4 because then the majority of the B and M diagnoses are separated into different rows (rows 1 and 3 above).

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

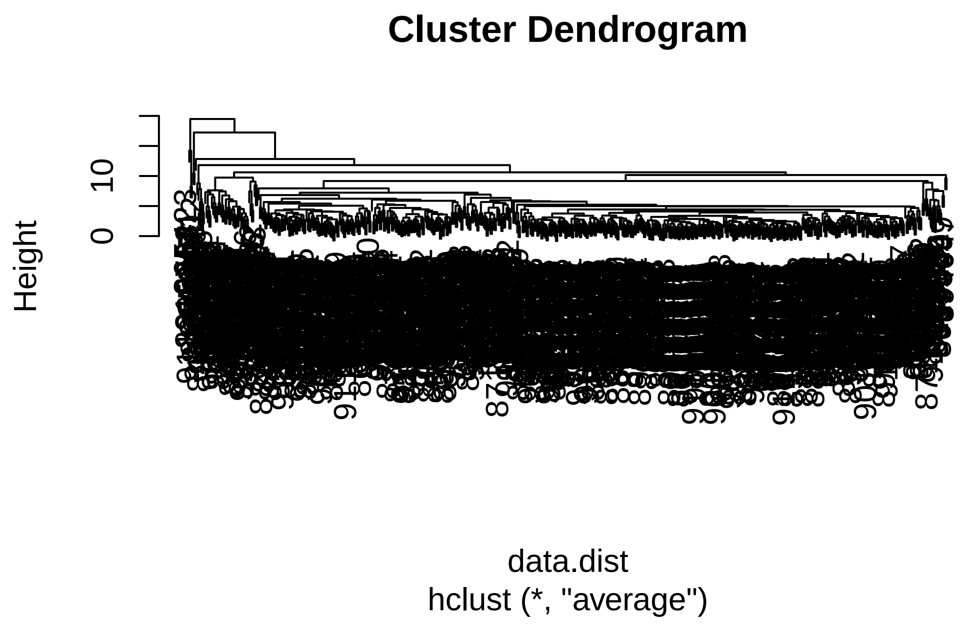
```r
plot(hclust(data.dist, method = "complete"))
```



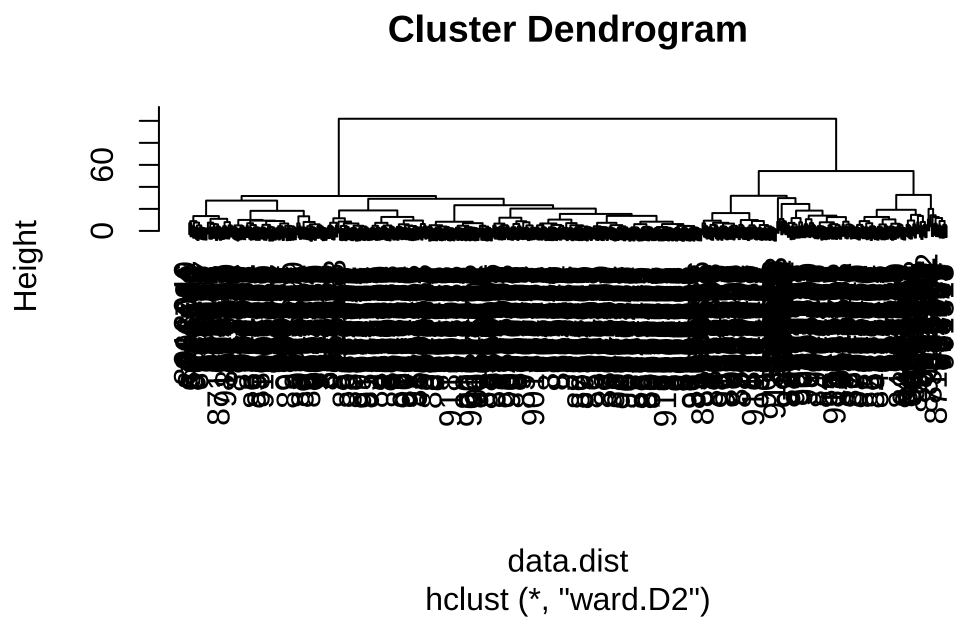
```r
plot(hclust(data.dist, method = "single"))
```



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



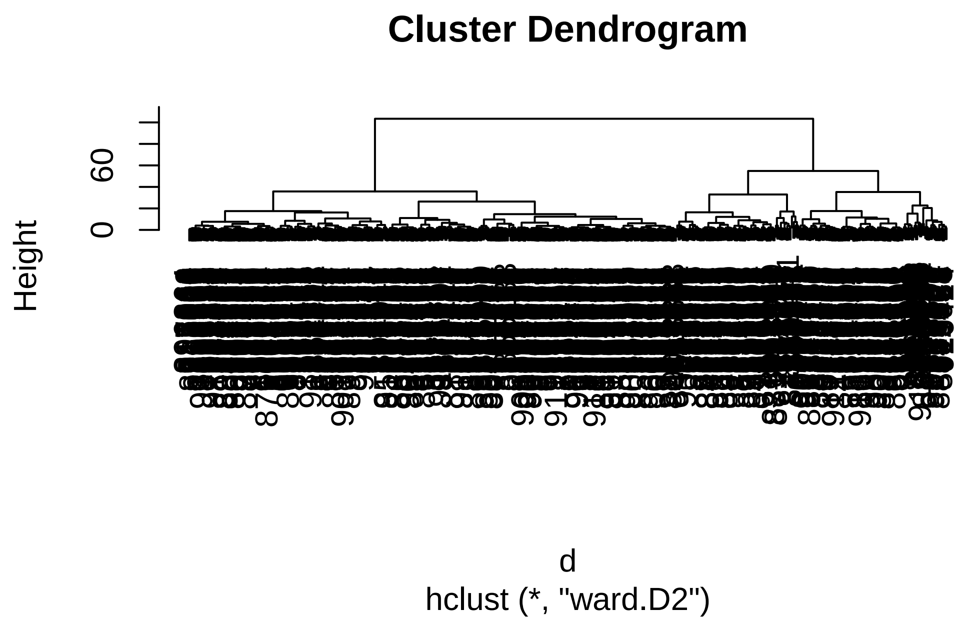
```r
plot(hclust(data.dist, method = "ward.D2"))
```



I definitely prefer the "ward.D2" clustering. It presents the cleanest groupings, and is easiest to read.

Combining methods

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



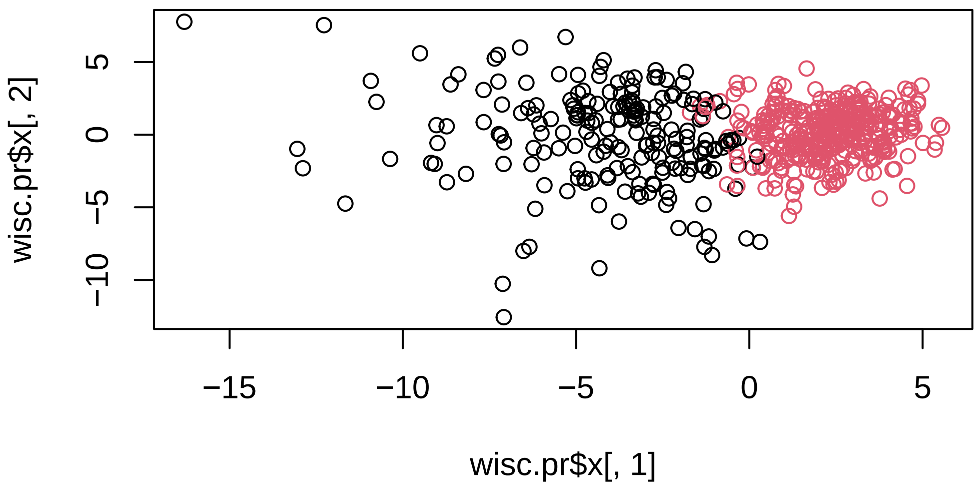
Generate 2 cluster groups from this hclust at the height for the number of clusters we want, here it's 2.

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

```
grps
  1   2 
203 366 
```

Plotting with color from `grps` instead of the expert diagnosis from before, we see that we have a very similar result!

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



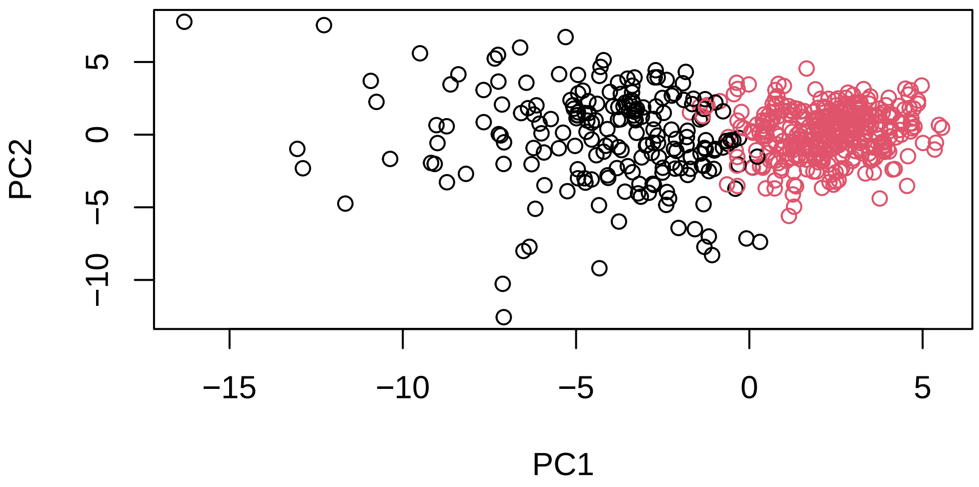
Let’s compare them:

```r
table(grps, diagnosis1)
```

```
    diagnosis1
grps   B   M
   1  24 179
   2 333  33
```

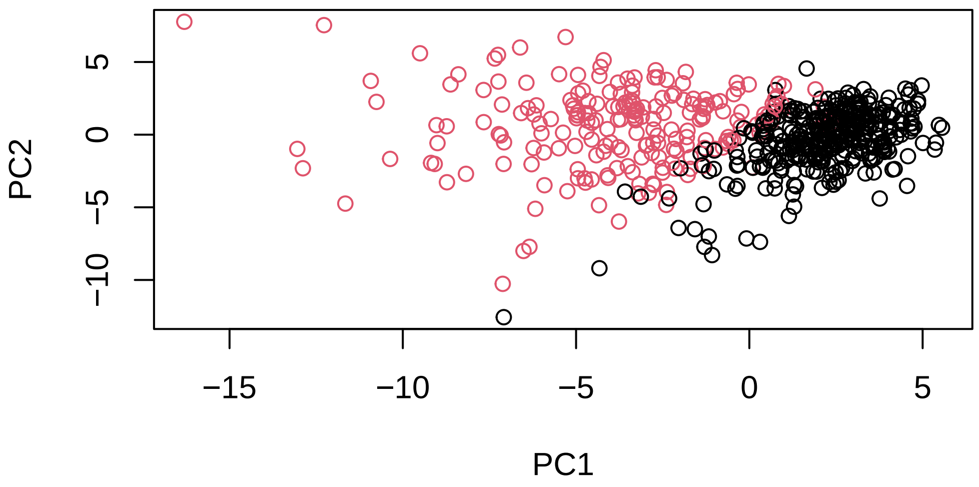
Here we color by groups:

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



Here we color by diagnosis:

```
plot(wisc.pr$x[,1:2], col=diagnosis1)
```



Changing the colors to match: (turn `grps` into a factor)

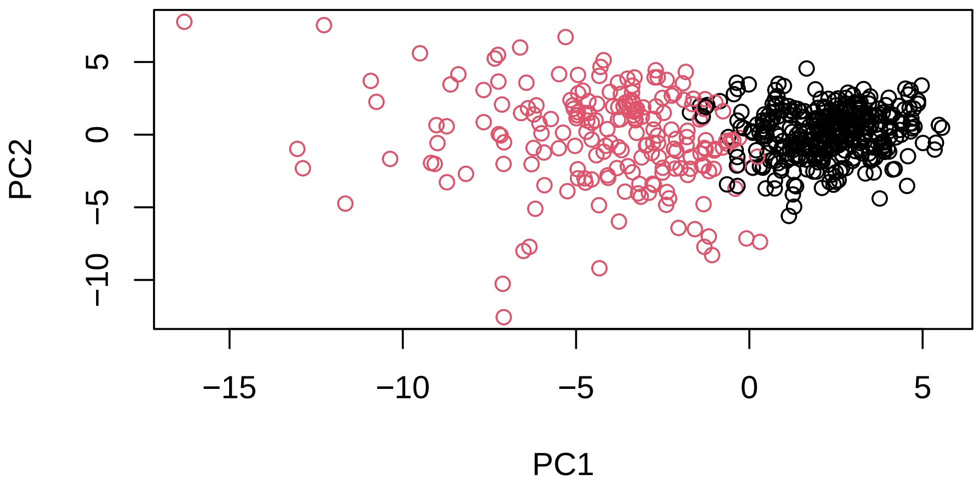
```
g <- as.factor(grps)
levels(g)
```

```
[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?

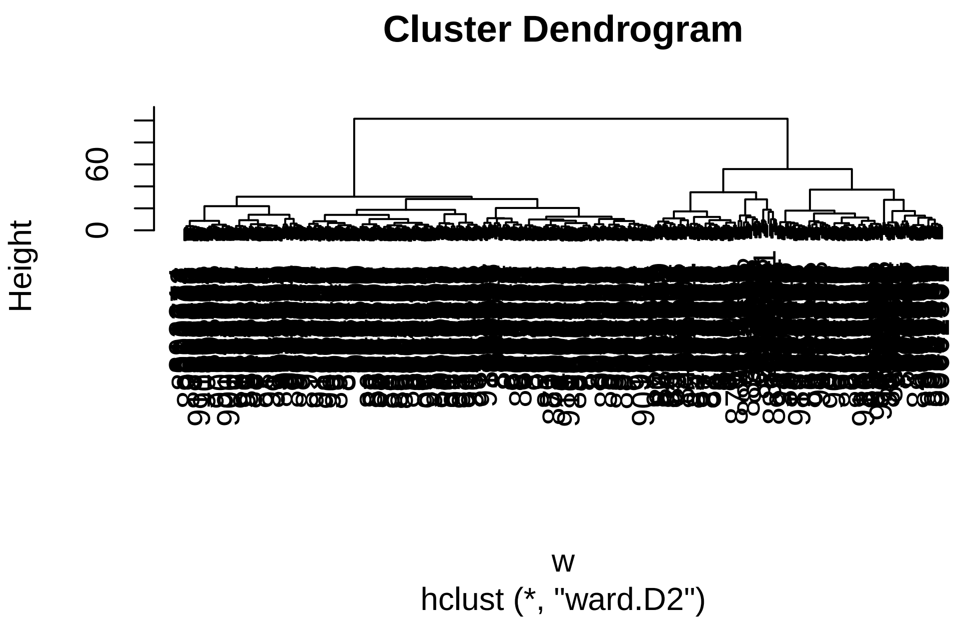
```
wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k=2)

table(wisc.pr.hclust.clusters, diagnosis1)
```

```
                       diagnosis1
wisc.pr.hclust.clusters   B   M
                      1  24 179
                      2 333  33
```

THis model works fairly well because the majority of diagnoses are separated, but there are still some potential false positives and false negatives in each cluster.

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


Cluster Dendrogram
Height
0
60
w
hclust (*, "ward.D2")