Class_08_020124

Johann Tailor

Goal

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.

Our data will be sourced from the site:

**Sometimes the data is not a url, in that case you can download it in the directory and then launch it using `read.csv()` or use the following code chunk:

```
wisc.df <- read.csv(url("https://bioboot.github.io/bimm143_S20/class-material/WisconsinCan
```

> Q1: How many observations/samples/patient# are in your data? Answer: 569

You can use this also (in-text running code):

569

```
nrow(wisc.df)
```

```
[1] 569
```

ANSWER: 569

> Q2: Whats in the `$diagnosis` column? How many of each types? Answer: Benign: 357 M: 212

Ways you can do this: Calculate T/F and count?

You can also use the table function:

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

```
[1] 212
```

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

```
[1] 357
```

```r
#the best one:
table(wisc.df$diagnosis)
```

```

  B   M 
357 212 
```

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

Answer: 10

```r
grep("_mean", colnames(wisc.df))
```

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

```r
length(grep("_mean", colnames(wisc.df)))
```

```
[1] 10
```

We will save the diagnosis for later:

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

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

We will now delete the diagnosis column so that we dont know the answer.

```r
wisc.data <- wisc.df [,-1]
dim(wisc.data)
```

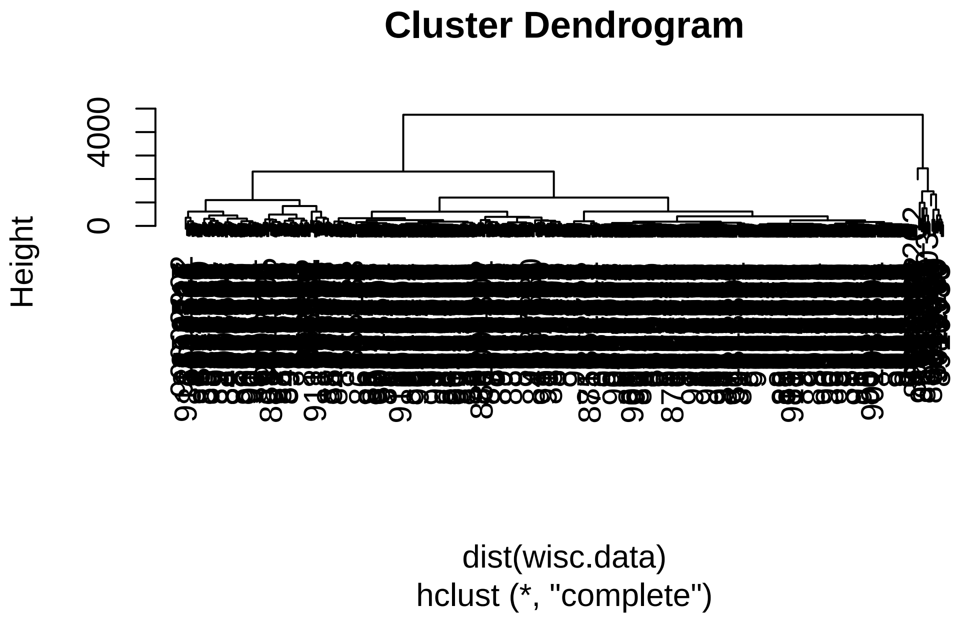
```
[1] 569  30
```

Section 2: Using PCA

Let's try clustering this data:

The format: hclust(d, method = "complete", members = NULL)

```r
wisc.hc <- hclust(dist(wisc.data))
plot(wisc.hc)
```



The data as is when clustered doesn't look good.

Let's try PCA

But first lets see if we have to scale the data.

```
apply(mtcars, 2, sd)
```

```
        mpg         cyl        disp          hp        drat          wt 
  6.0269481   1.7859216 123.9386938  68.5628685   0.5346787   0.9784574 
       qsec          vs          am        gear        carb 
  1.7869432   0.5040161   0.4989909   0.7378041   1.6152000 
```

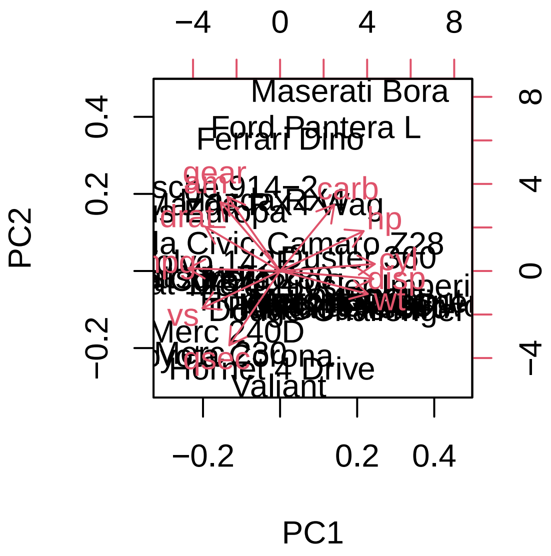
In this example, display since its ST.DEV is very high, it will dominate the whole PCA. Therefore, we need to scale it.

```
pc.scale <- prcomp(mtcars, scale=T)
summary(pc.scale)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6    PC7
Standard deviation     2.5707 1.6280 0.79196 0.51923 0.47271 0.46000 0.3678
```

```
Proportion of Variance 0.6008 0.2409 0.05702 0.02451 0.02031 0.01924 0.0123
Cumulative Proportion  0.6008 0.8417 0.89873 0.92324 0.94356 0.96279 0.9751
                           PC8    PC9    PC10   PC11
Standard deviation     0.35057 0.2776 0.22811 0.1485
Proportion of Variance 0.01117 0.0070 0.00473 0.0020
Cumulative Proportion  0.98626 0.9933 0.99800 1.0000
```

```
biplot(pc.scale)
```



Back to cancer dataset:

```
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 see that the variance is very different so we will scale it.

```
wisc.pc.scale <- prcomp(wisc.data, scale=T)
```

How well is the PCs captured from the original data set:

```
summary(wisc.pc.scale)
```

```
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
```

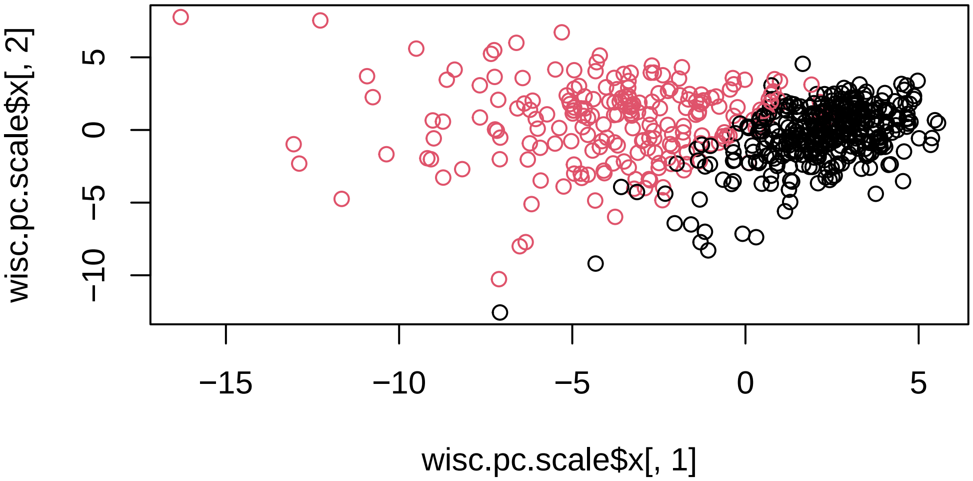
```
Cumulative Proportion  1.00000 1.00000
```

Now, lets get our main PC score plot (a.k.a PC1 Vs. PC2 plot):

```
# these are the attributes of the PCA plot. They will be standard.

attributes(wisc.pc.scale)
```

```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

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



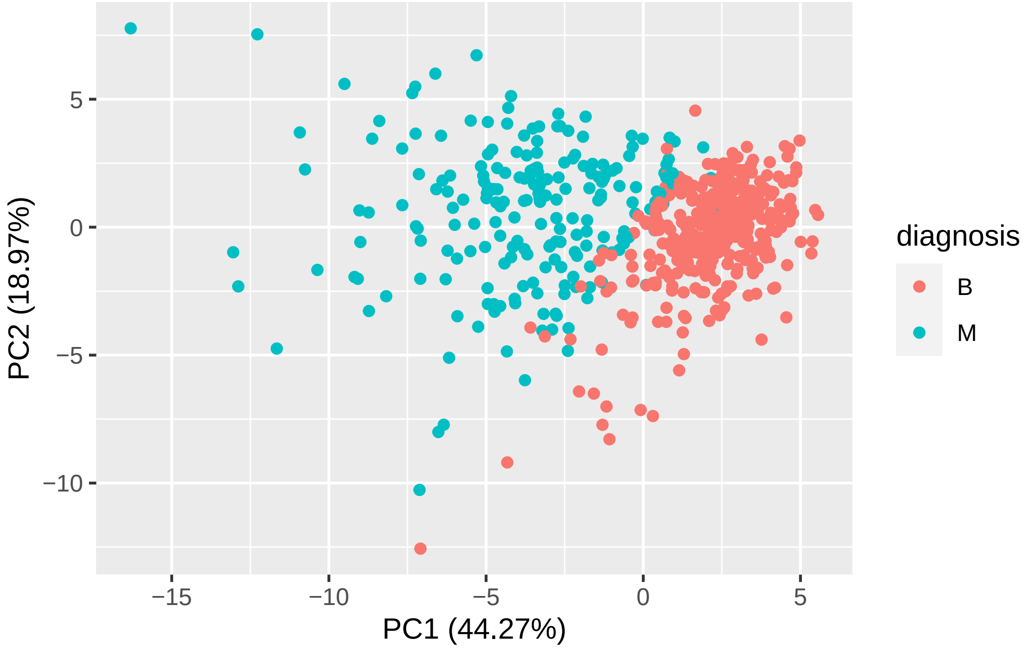
```
# 1 here stands for PC1 nad 2 stands for PC2.
```

Now, lets make nice ggplot

```
pc <- as.data.frame(wisc.pc.scale$x)
dim(pc)
```

```
[1] 569  30
```

```
library(ggplot2)

ggplot(pc, aes(x= pc$PC1, y= pc$PC2, col=diagnosis)) + geom_point() + labs(x = "PC1 (44.27
```



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

Answer: 44.27%

```
summary(wisc.pc.scale)
```

```
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
```

ANSWER: It cover 44.27% of the variance.

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

Answer: PC1 and PC2 cover about 63.24% (closest to 70%). The summary shown above was used to calculate it.

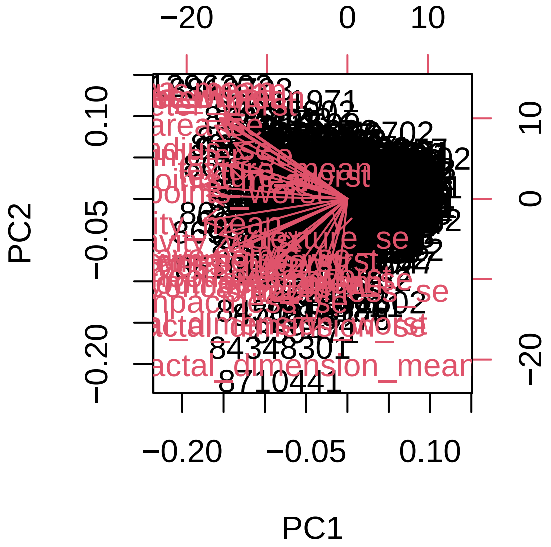
> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

PC1 through PC6 cover 88.759% of data (closest to 90%)

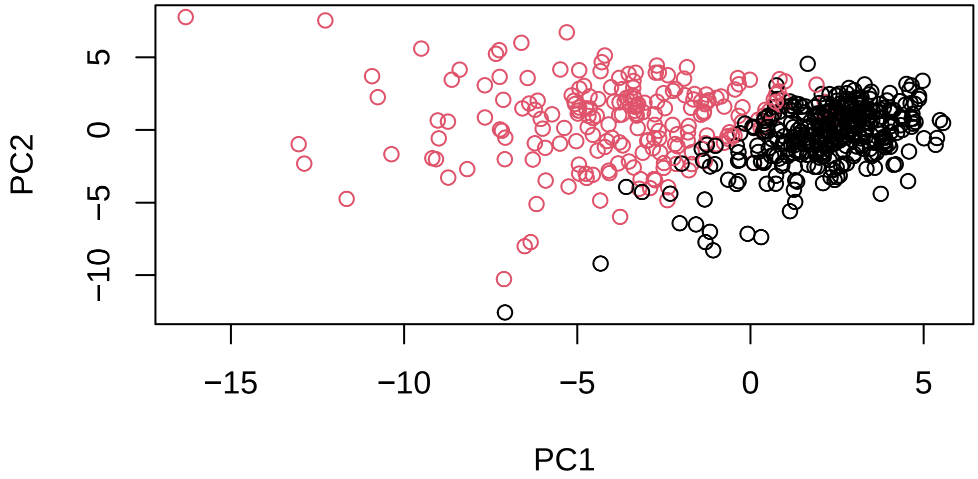
> Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

Answer: Its very crowded and has all patient infomration. It needs to be put in terms of variance via PCA plots.

```
biplot(wisc.pc.scale)
```



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

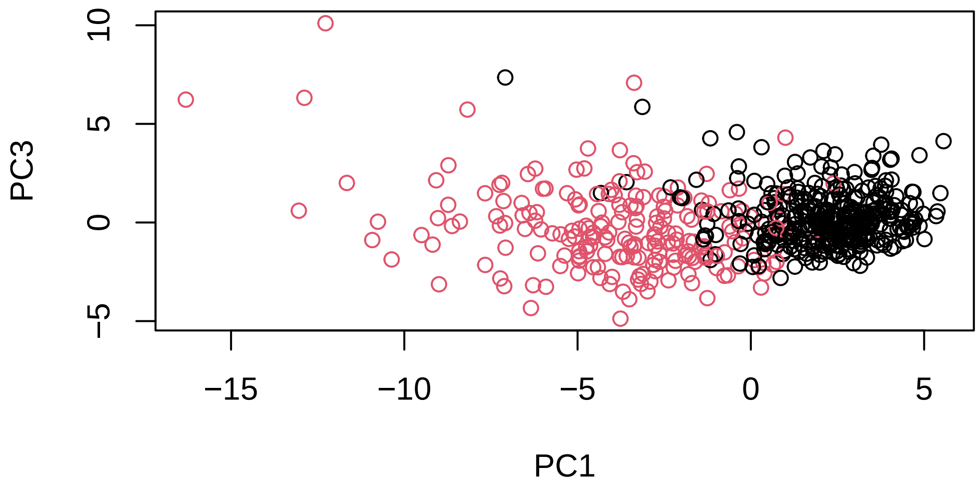


Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

Answer: they are very closely clustered. No start or visual difference.

Use this: plot(wisc.pc.scale$x[, 1], wisc.pc.scale$x[, 2], col=diagnosis)

```
plot(wisc.pc.scale$x[, 1], wisc.pc.scale$x[, 3], col=diagnosis,
     xlab = "PC1", ylab = "PC3")
```



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.

Answer: -0.2608538

```
loading_component <- wisc.pc.scale$rotation["concave.points_mean", 1]

# Print the result
print(loading_component)
```

```
[1] -0.2608538
```

hierarchical clustering

```
#squaring the standard deviation for each column:

wisc.pc.scale$sdev^2
```

```
 [1] 1.328161e+01 5.691355e+00 2.817949e+00 1.980640e+00 1.648731e+00
 [6] 1.207357e+00 6.752201e-01 4.766171e-01 4.168948e-01 3.506935e-01
[11] 2.939157e-01 2.611614e-01 2.413575e-01 1.570097e-01 9.413497e-02
[16] 7.986280e-02 5.939904e-02 5.261878e-02 4.947759e-02 3.115940e-02
[21] 2.997289e-02 2.743940e-02 2.434084e-02 1.805501e-02 1.548127e-02
[26] 8.177640e-03 6.900464e-03 1.589338e-03 7.488031e-04 1.330448e-04
```

```
#saving the variance of each PC as pv.rar
pr.var <- wisc.pc.scale$sdev^2
head(pr.var)
```

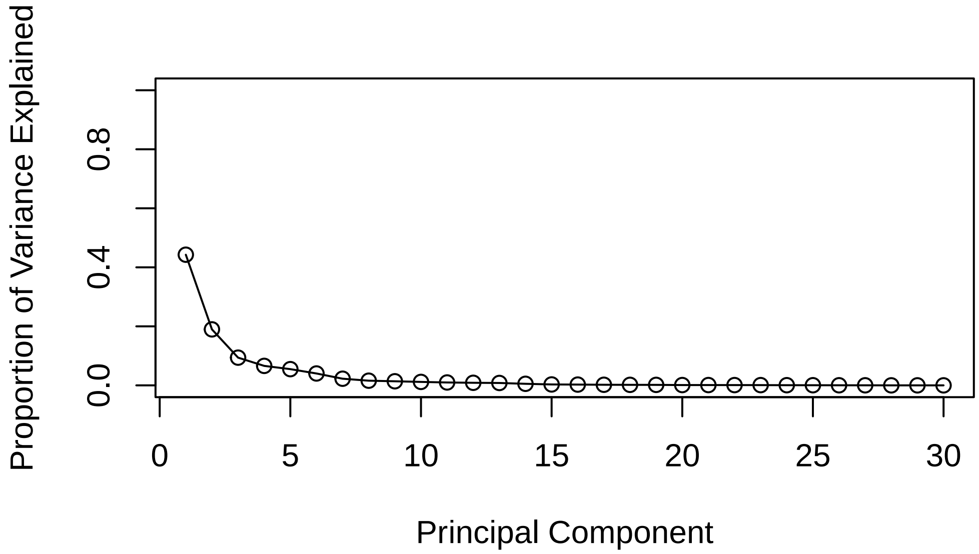
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
#pve will divide each PC with the total variance
pve <- (pr.var)/ (sum(pr.var))
pve
```

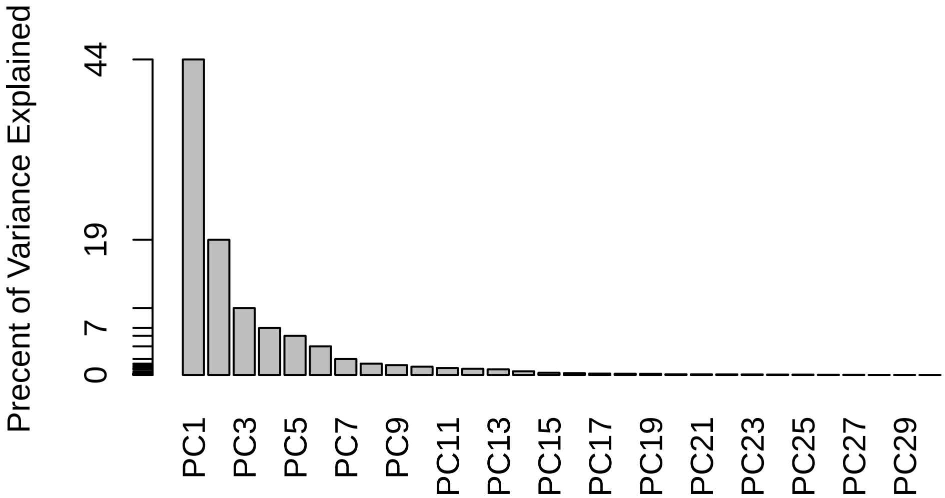
```
 [1] 4.427203e-01 1.897118e-01 9.393163e-02 6.602135e-02 5.495768e-02
 [6] 4.024522e-02 2.250734e-02 1.588724e-02 1.389649e-02 1.168978e-02
[11] 9.797190e-03 8.705379e-03 8.045250e-03 5.233657e-03 3.137832e-03
[16] 2.662093e-03 1.979968e-03 1.753959e-03 1.649253e-03 1.038647e-03
[21] 9.990965e-04 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
```

Plotting the scree plot:

```
# Plot variance explained for each principal component

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



```
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 )
```

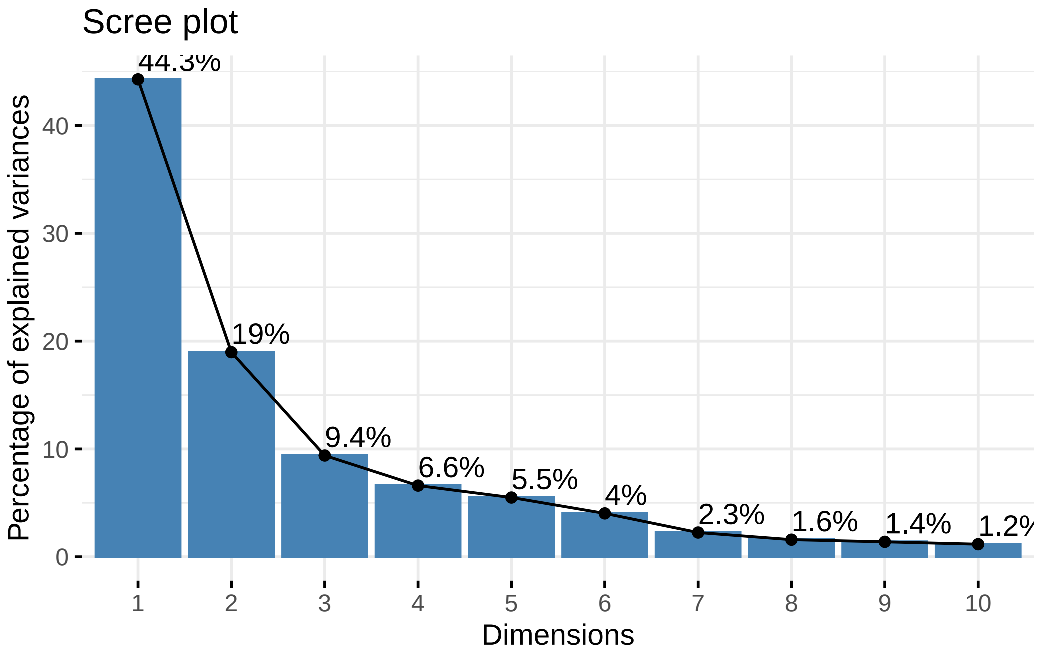


Another way:

```r
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

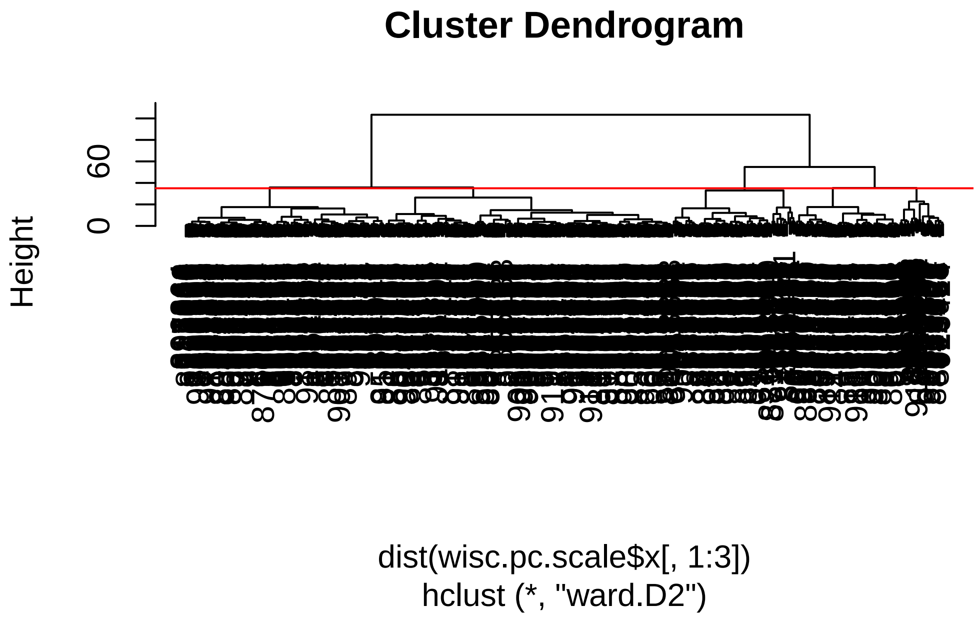
```r
fviz_eig(wisc.pc.scale, addlabels = TRUE)
```



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

Answer: at the height of 35, I get 4 clusters as shown by the graph below.

```
# taking onlt the first three PCs

#wisc.pc.scale$x[,1:3]

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

plot(wisc.pr.hclust)

#lets cut the dendogram to get bigger clusters:

abline(h=35, col="red")
```



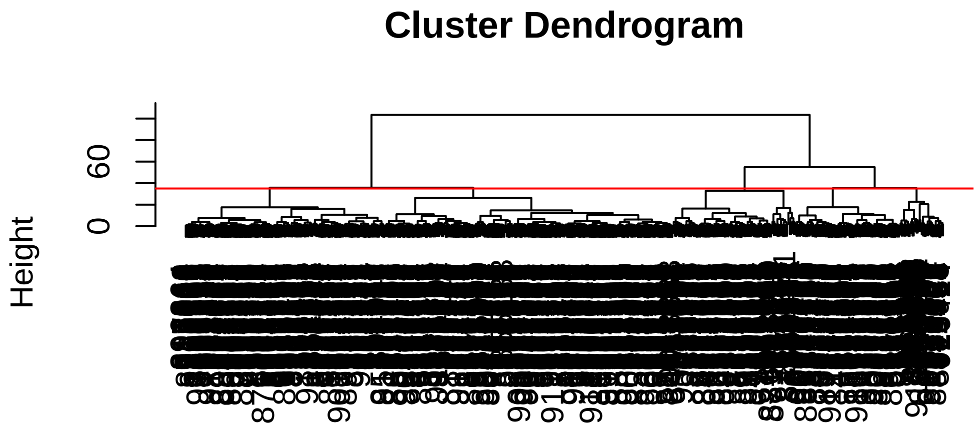
Q12. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

Answer: After looking at them all, I think `ward.D2` shows the data in an understandable manner. It shows clear clusters and the data is represented in botttom up heirachchial manner.

combining methods

Here we will use the results of the PCA as a the input to a clustering analysis:

```
# taking onlt the first three PCs

#wisc.pc.scale$x[,1:3]

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

plot(wisc.pr.hclust)

#lets cut the dendogram to get bigger clusters:

abline(h=35, col="red")
```



dist(wisc.pc.scale$x[, 1:3])
hclust (*, "ward.D2")

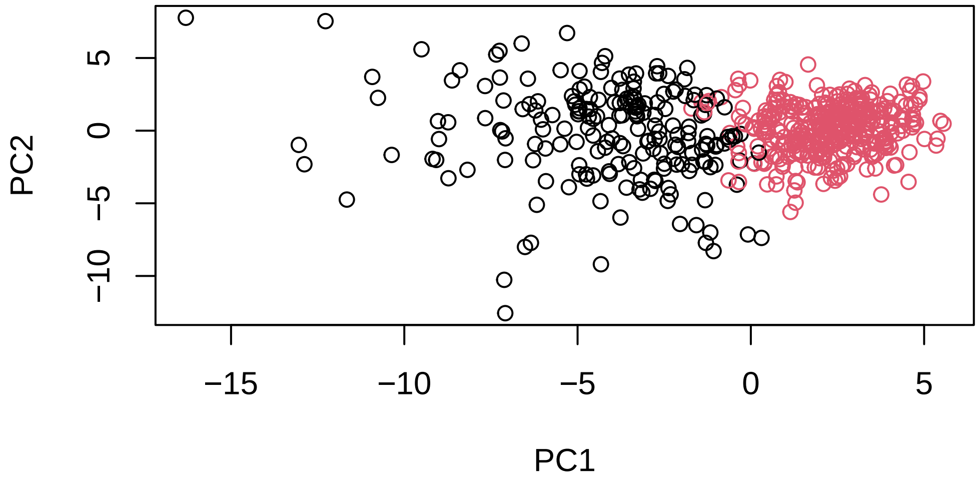
```
groups <- cutree(wisc.pr.hclust, k=2)
table(groups)
```

```
groups
  1   2 
203 366 
```

```
table(groups, diagnosis)
```

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

```
plot(wisc.pc.scale$x[,1:2], col=groups)
```



Q.Q11. OPTIONAL: Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10? How do you judge the quality of your result in each case?

```
#now lets find how many patients are involved in these two groups:

table(groups)
```

```
groups
  1   2 
203 366 
```

```
table(diagnosis)
```

```
diagnosis
  B   M 
357 212 
```

```
#This will combine the two data sets to give you a cross-reference:
```

```
table(groups, diagnosis)
```

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

changing groups into clusters:

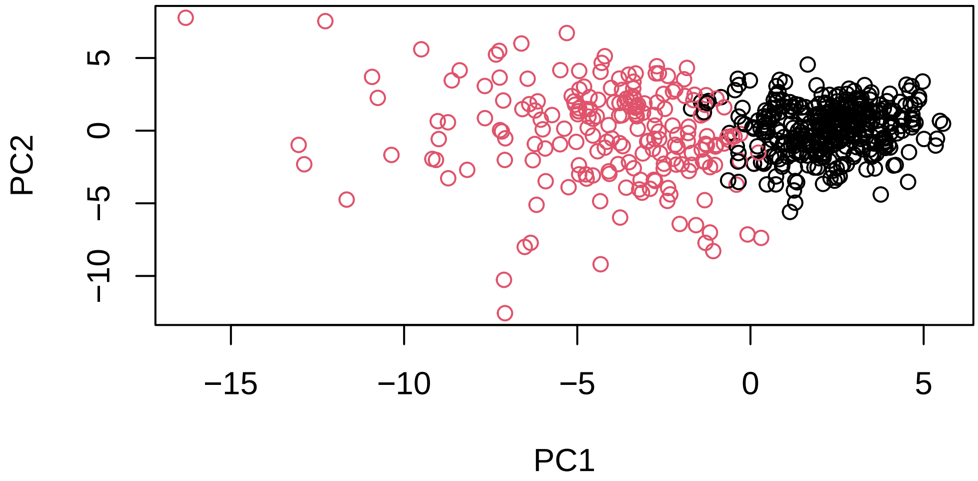
```
g <- as.factor(groups)
levels(g)
```

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

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

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

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



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)
```

```
# Compare to actual diagnoses
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?

Answer: The newly created model with 2 clusters is far more accurate than the previous one w/o clustering.

> 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.

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

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

Section 5

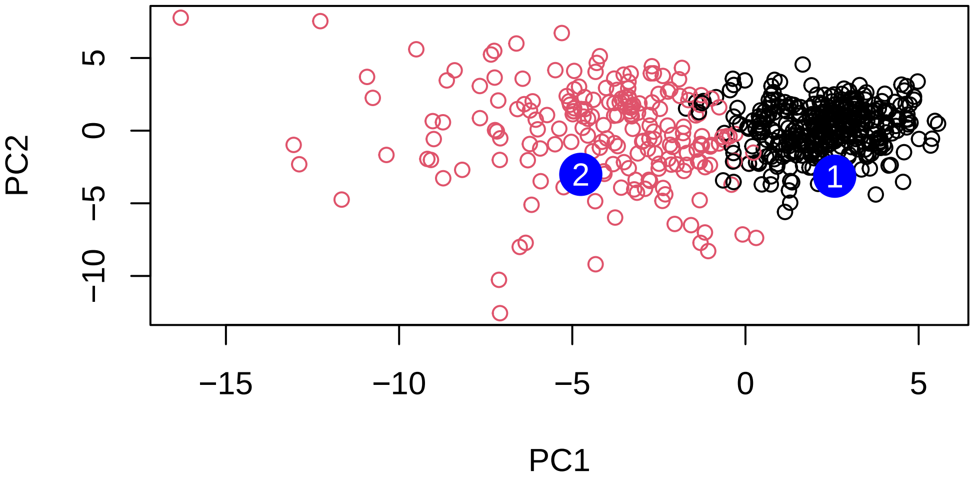
179/212 = sensitivity True Negative = NON MALIGNANT

Section 6: Prediction

```
#url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pc.scale, 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.pc.scale$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")
```



Q16. Which of these new patients should we prioritize for follow up based on your results?

Answer: I think patient 1, as its clustering is very solid with the black group and it has a PC1 value that is positive (~2.5)