Class08

Brooke Clements

Background

In todays class we will apply the methods and techniques clustering and PCA to help make sense of real world breast cancer FNA biopsy data set.

Data import

We start by importing our data. It is a CSV file so we will use the `read.csv()` function.

```
fna.data <- "https://bioboot.github.io/bimm143_S20/class-material/WisconsinCancer.csv"
```

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

```
View(wisc.df)
```

Have a peak at the first few entries.

```
head(wisc.df, 4)
```

```
         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
         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
         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
          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
          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
          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
          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
          fractal_dimension_worst
842302                    0.11890
842517                    0.08902
84300903                  0.08758
84348301                  0.17300
```

Make sure to remove the first `diagmosis` column - I don'y want to use this for my machine learning models. We will use it later to compare our resutlts to the expert diagnois.

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

Q1. How many observations are in this dataset?

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

```
[1] 569
```

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

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

```
  B   M
357 212
```

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

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

```
[1] 10
```

Principal Component Analysis

The main function here is `prcomp()` and we want to make sure we set the optional argument `scale=TRUE`

```
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 component (PC1)?

0.4427

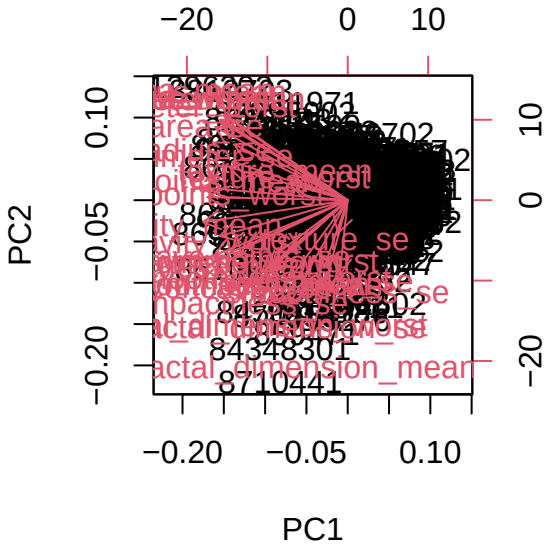
Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data?

We need 3 PCs to describe 72.636% 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?

We need 7 PCs to describe 91.01% of the original variance in the data.

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



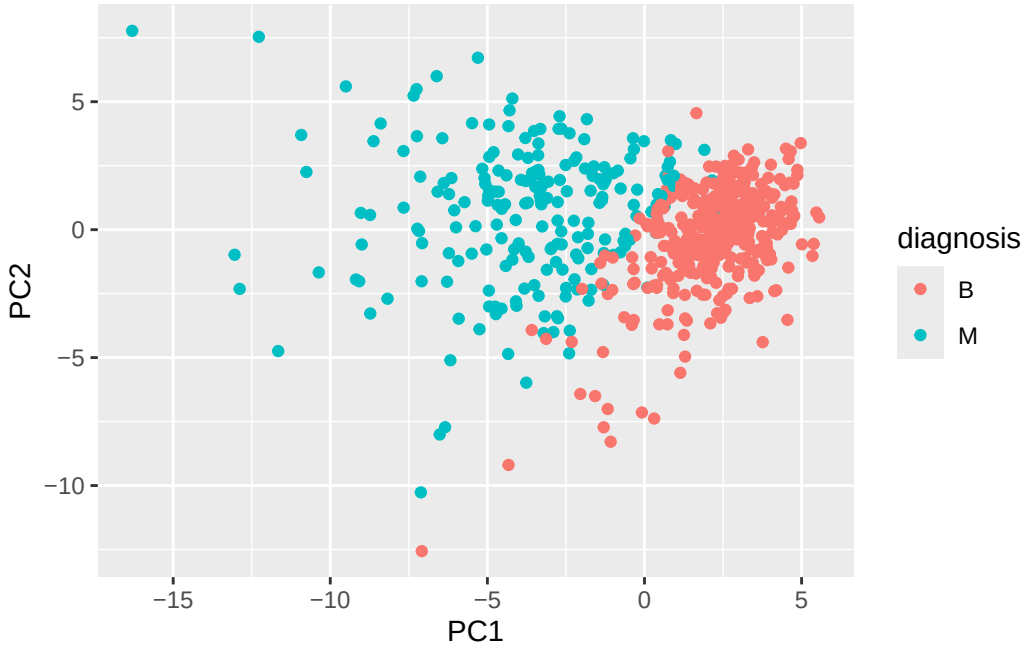
Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

The first thing that stands out about this grapg is how overwhelming it is. This graph is difficult to understand due to all the overlaping data points.

Our main PCS score plot or "PC plot" of results:

```
library(ggplot2)
```

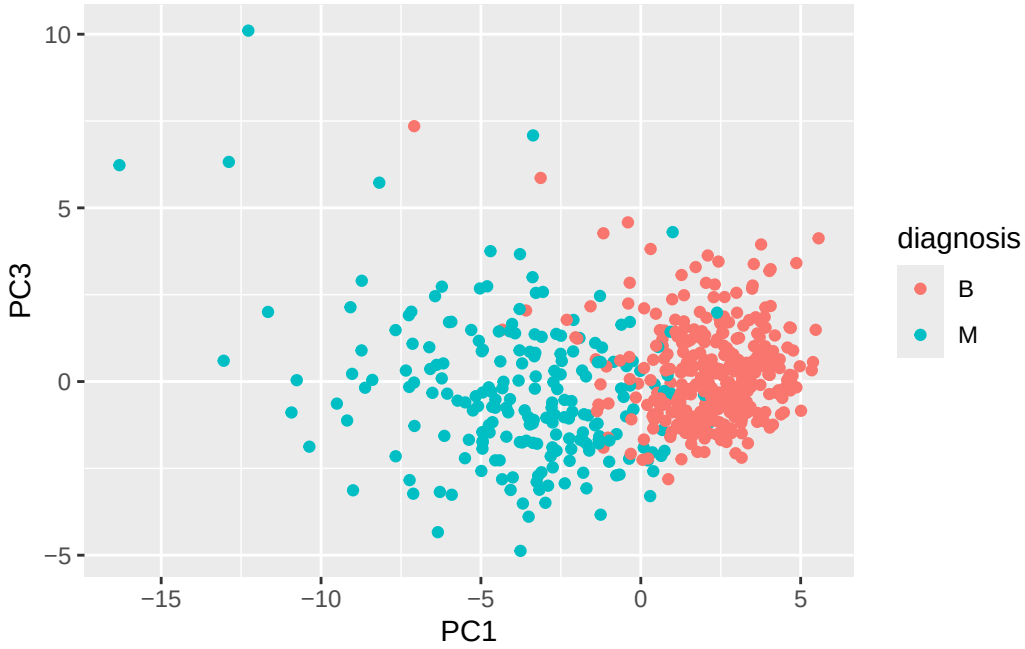
```
ggplot(wisc.pr$x) +
  aes(PC1,PC2, col=diagnosis)+
  geom_point()
```



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

Componemts 1 and 2 have a more clear seperation that 1 and 3.

```
ggplot(wisc.pr$x) +
  aes(PC1,PC3, col=diagnosis)+
  geom_point()
```



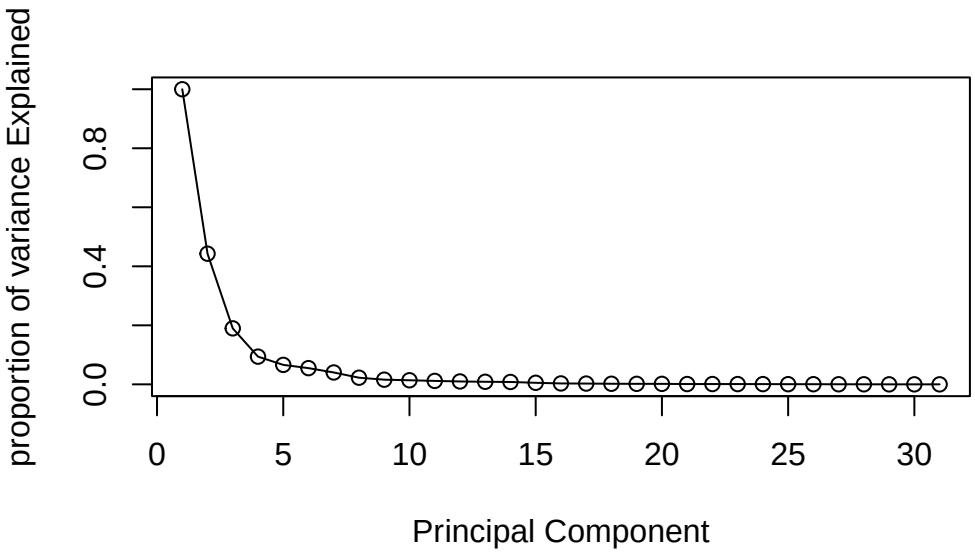
#Calculate variance of each component

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

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

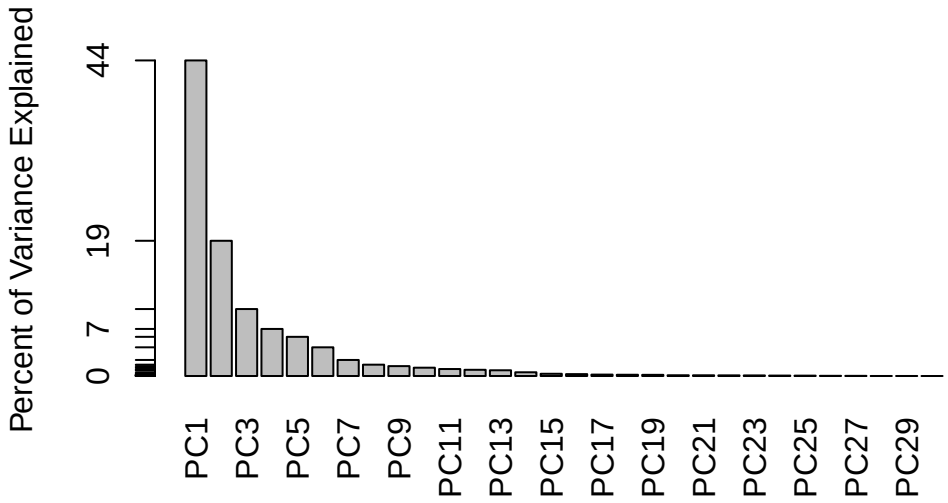
#Variance explained by each principal component: pve

```
pve <- pr.var/30
plot(c(1,pve),xlab="Principal Component", ylab="proportion of variance Explained", ylim=c(0,
```



#Alternative scree plot of the same data, note data driven y-axis

```
barplot(pve,ylab="Percent of Variance Explained",
        names.arg=paste0("PC",1:length(pve)), las=2, axes=FALSE) +
axis(2, at=pve, labels=round(pve,2)*100 )
```




```
            [,1]
 [1,]  0.7000044
 [2,]  1.9000250
 [3,]  3.1000530
 [4,]  4.3002300
 [5,]  5.5002726
 [6,]  6.7005160
 [7,]  7.9006018
 [8,]  9.1008114
 [9,] 10.3009146
[10,] 11.5009991
[11,] 12.7010386
[12,] 13.9016493
[13,] 15.1017540
[14,] 16.3019800
[15,] 17.5026621
[16,] 18.7031378
[17,] 19.9052337
[18,] 21.1080452
[19,] 22.3087054
[20,] 23.5097972
[21,] 24.7116898
[22,] 25.9138965
```

```
[23,] 27.1158872
[24,] 28.3225073
[25,] 29.5402452
[26,] 30.7549577
[27,] 31.9660213
[28,] 33.1939316
[29,] 34.4897118
[30,] 35.9427203
```

```
If cells in the nucleus are deeply indented ("concave"), irregular non circular ("compactanc
```

```
>Q9. For the first principal component, what is the component of the loading vector (i.e. wi
```

```
::: {.cell}

```{.r .cell-code}
wisc.pr$rotation["concave.points_mean",1]
```

[1] -0.2608538

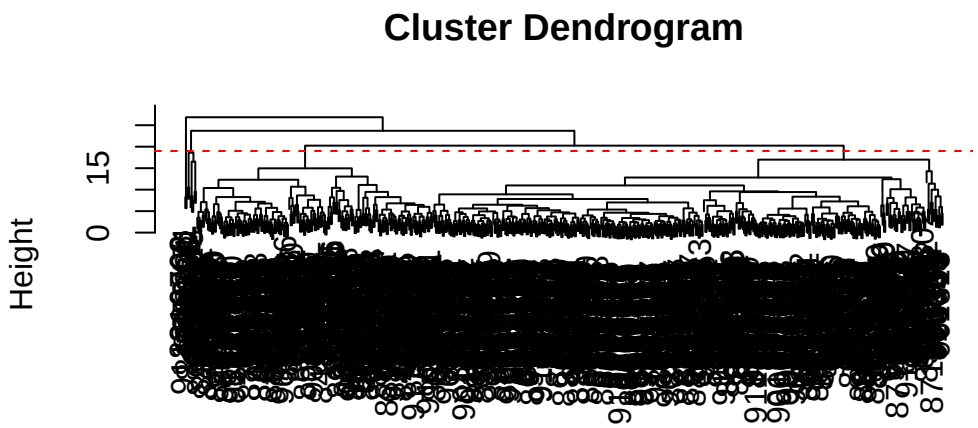
:::
```

##Hierarchical Clustering First scale the data (with the `scale()`) function , then caculate a distance matrix (with the `dist()`function). Then cluster `hclust()` function and plot:

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

> 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, col="red", lty=2)
```




```
integer(0)
```

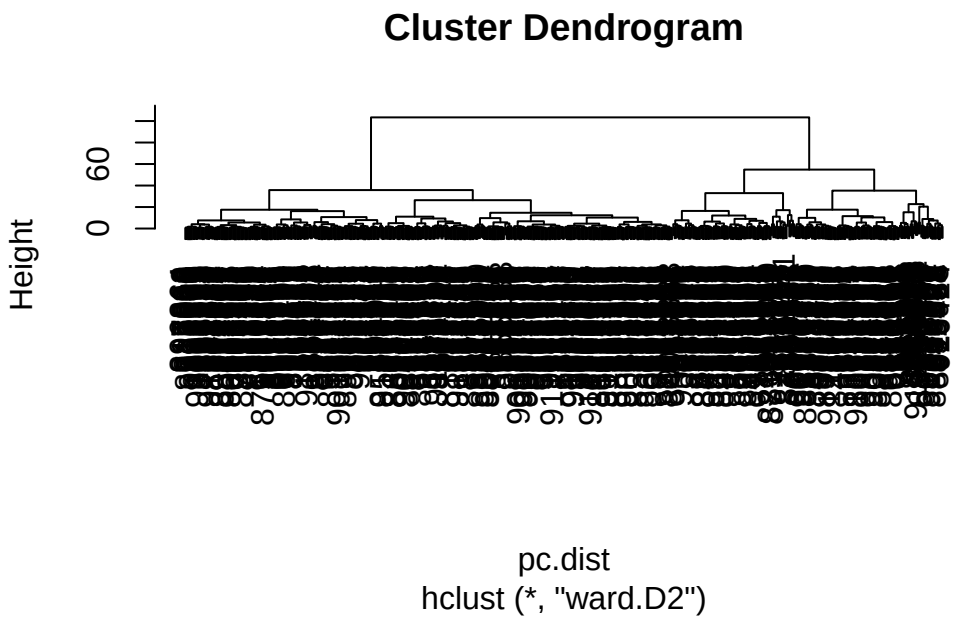
```
wisc.hclust.cluster <- cutree(wisc.hclust, k=4)
table(wisc.hclust.cluster, diagnosis)
```

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

Combinging methods

Here we will take our PCA results and use those as inputs for clusintering. In other words our `wisc.pr$x` scores that we plotted above (the main output from PCA - how the data lie on our new principal componet axis / variables) and use a subset of these PCs that capture the most variance as input for `hclust()`

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



Cut the dendrogram/tree into two main groups/clusters:

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

```
grps
  1   2
203 366
```

I want to know how the clustering in `grps` with values of 1 or 2 correspond the expert `diagnosis`

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

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

My clustering **groups 1** are mostly "M" diagnosis (179) and my clustering **group 2** are mostly "B" diagnosis.