

Mini-Project

Arturo Agramont

Mini-Project

We will start by reading the data

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

We will remove the first column

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

diag_levels = c("B", "M")
diagnosis = factor(wisc.df$diagnosis, levels =diag_levels)
diagnosis
```

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

Levels: B M

- **Q1.** How many observations are in this dataset?

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

```
[1] 569
```

There are 569 observations in this data set.

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

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

```
  B    M  
357 212
```

212 observations were given a malignant diagnosis

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

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

```
integer(0)
```

10 of the variables are suffixed with `_mean`

PCA

Checking columns and standard deviations

```
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

```
wisc.pr = prcomp(scale(wisc.data))
```

```
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 is the proportion of PC1.

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

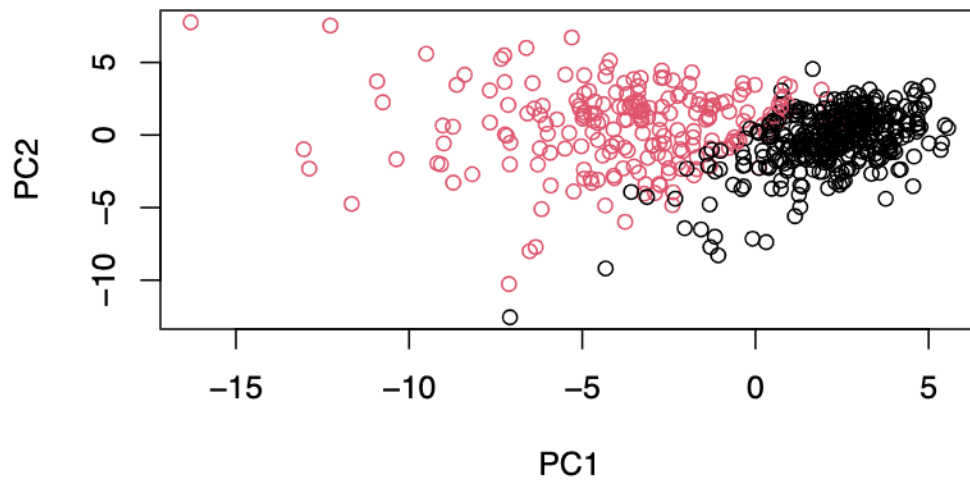
3 principle components are needed. PC1, PC2, and PC3

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

7 PCs are needed to describe 90% of variance.

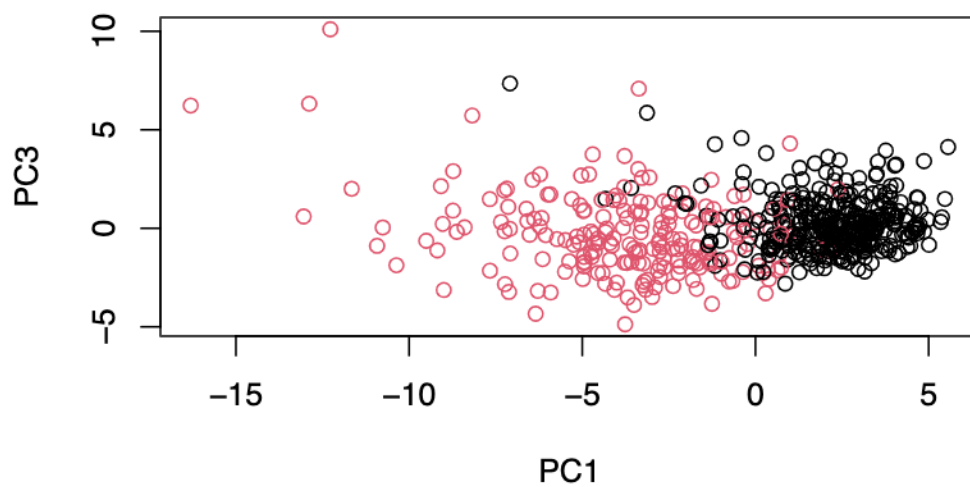
We will now create a biplot

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

- **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,  
     xlab = "PC1", ylab = "PC3")
```



There is a greater distinction between points in plot 1 than plot 2

Now using ggplot2

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

library(ggplot2)

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



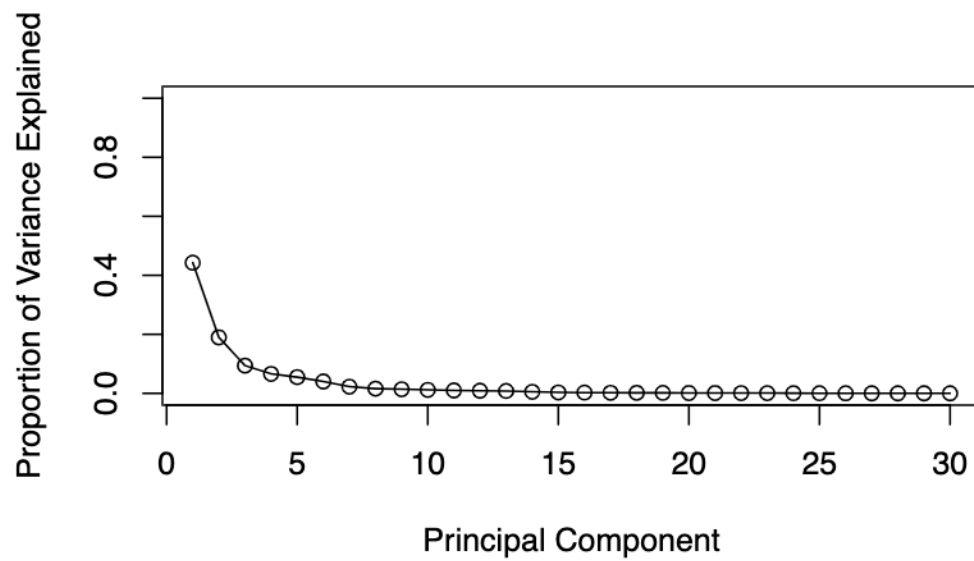
Calculating variance of each principal component

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

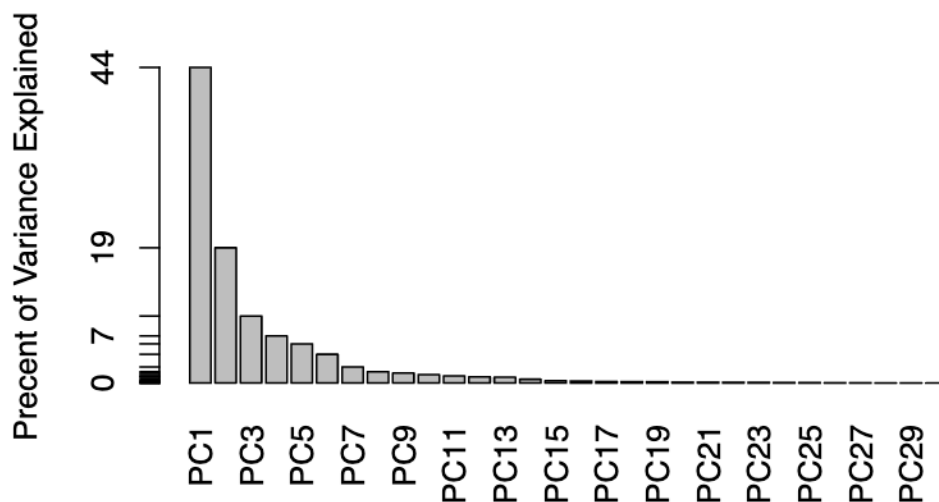
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
pve = pr.var/sum(pr.var)
```

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

```
# 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 )
```



- **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`?

```
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

<code>concave.points_worst</code>	<code>symmetry_worst</code>	<code>fractal_dimension_worst</code>
-0.25088597	-0.12290456	-0.13178394

It tells how much the original feature contributes to the first PC.

Hierarchical clustering

First we will scale `wisc.data` and assign to `data.scaled`

```
data.scaled = scale(wisc.data)
```

calculating euclidean distance between pairs in scaled data

```
data.dist= dist(data.scaled)
```

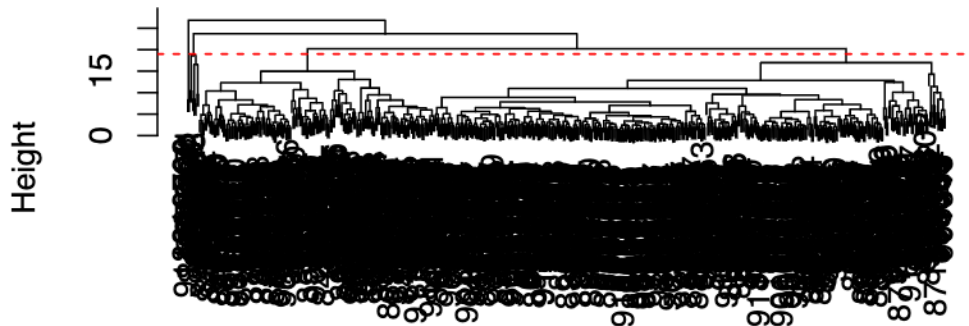
Creating a hierarchical clustering model

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

Cluster Dendrogram



data.dist
hclust (*, "complete")

Using function `cutree()` in order to make the tree have 4 clusters

```
wisc.hclust.clusters = cutree(wisc.hclust, k = 4, h=19)
```

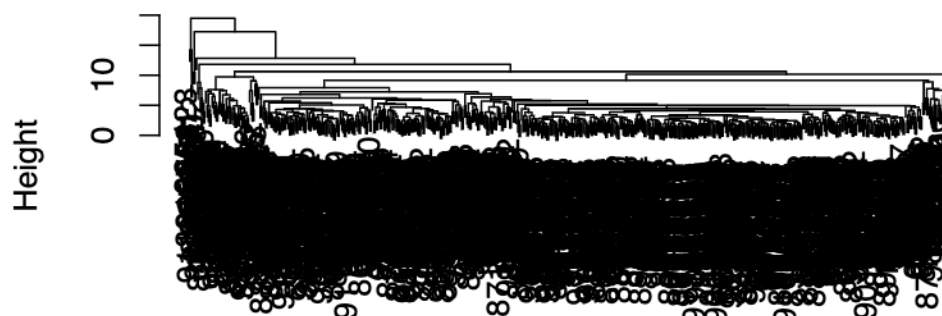
```
table(wisc.hclust.clusters, diagnosis)
```

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

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

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

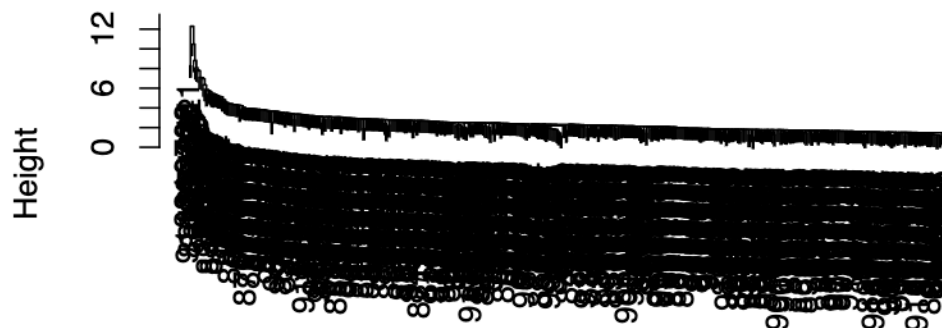
Cluster Dendrogram



```
data.dist  
hclust (*, "average")
```

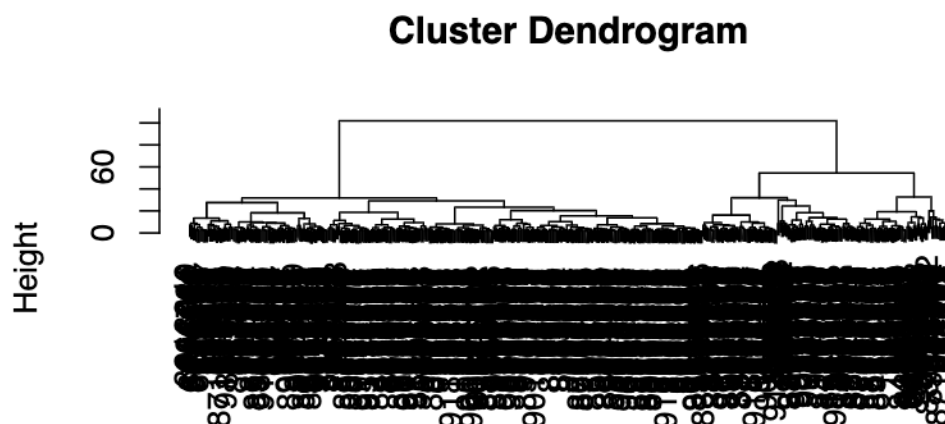
```
plot(wisc.hclust.single)
```

Cluster Dendrogram



```
data.dist  
hclust (*, "single")
```

```
plot(wisc.hclust.ward)
```



```
data.dist
hclust (*, "ward.D2")
```

Either ward or complete are my favorite results for the same data set because they provide easier to digest views of the data..

Now we will look into the 2 groups of the ward tree

```
data_dist = dist(wisc.pr$x[,1:7])
wisc.pr.hclust = hclust(data_dist, method = "ward.D2" )
grps = cutree(wisc.pr.hclust, k=2)
table(grps)
```

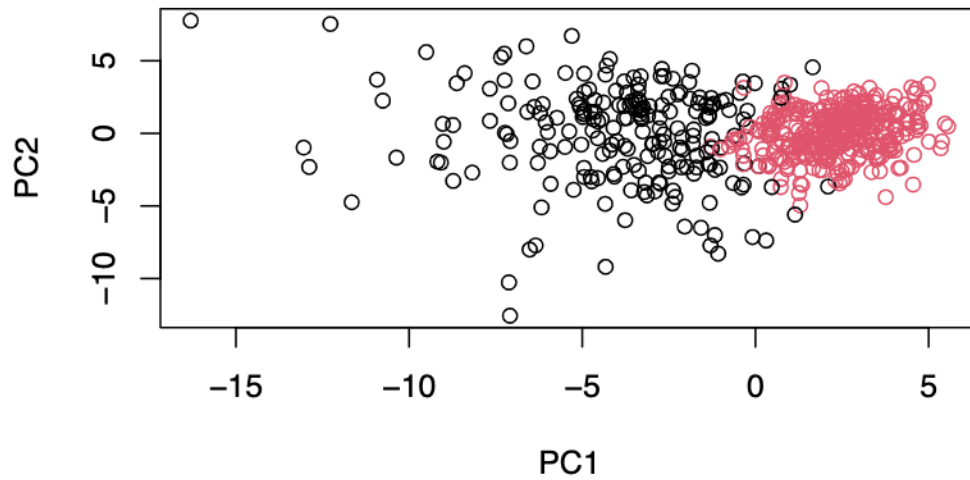
```
grps
 1  2
216 353
```

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

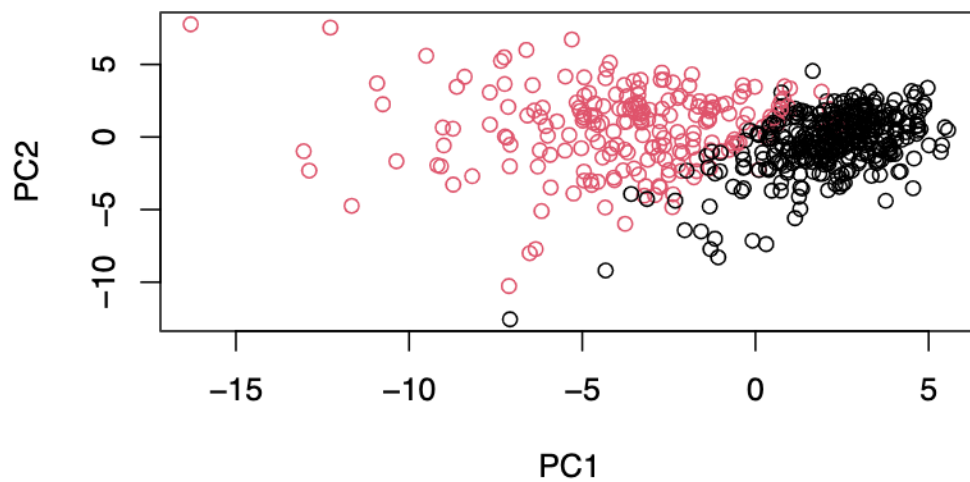
```
diagnosis
grps    B    M
```

```
1 28 188
2 329 24
```

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



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



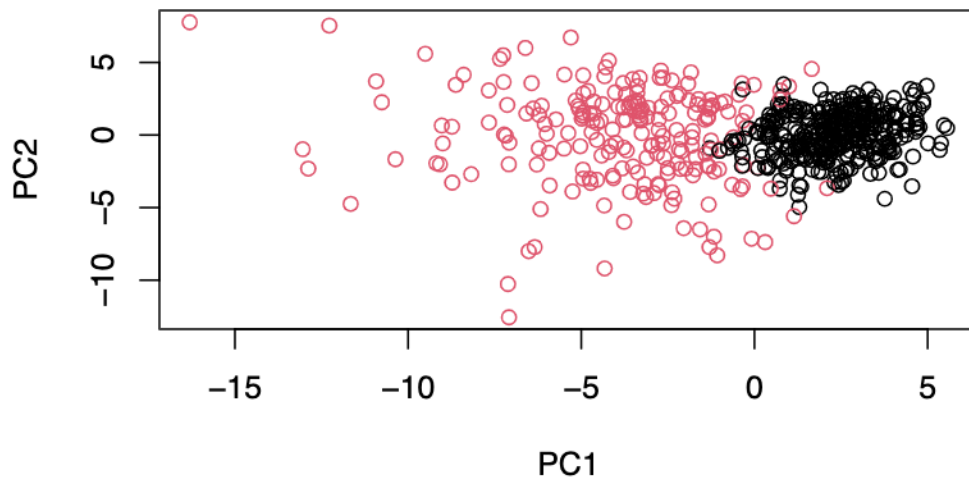
```
g = as.factor(grps)
levels(g)
```

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

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

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

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

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

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

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

It separates the diagnoses more where diagnosis B is more in group 2 and diagnosis M is more in group 1.

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.hclust.clusters, diagnosis)
```

```

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

```

This created 4 groups in place of 2 that don't have a more distinct formation of groups as the other model.

Prediction

Using `predict()` function to take PCA model from before onto new cancer data

```

#url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
npc

```

```

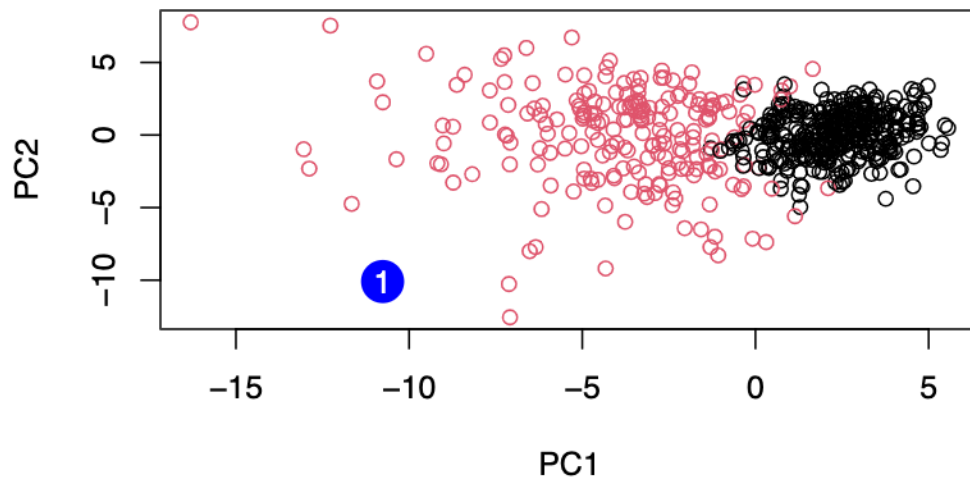
              PC1          PC2          PC3          PC4          PC5          PC6          PC7
[1,] -10.76452 -10.093978 -0.5897994 -4.164748 10.61922 -1.630738 0.03566861
[2,] -18.09606 -9.967098 -2.1549431 -4.006848 6.69687 -2.034714 1.25088149
              PC8          PC9          PC10          PC11          PC12          PC13          PC14
[1,] 0.7308658 -1.580861 3.166451 -0.7167150 3.850569 -0.8259764 1.0195729
[2,] 0.6308585 -1.155629 3.608207 -0.3405375 2.288732 -0.3976672 0.1347203
              PC15          PC16          PC17          PC18          PC19          PC20          PC21
[1,] 3.735687 -4.068783 1.0877034 0.9985959 1.022760 -2.430215 -1.295749
[2,] 3.543905 -3.749616 0.7613603 1.1763217 1.366702 -2.609643 -1.541050
              PC22          PC23          PC24          PC25          PC26          PC27          PC28
[1,] -1.348026 -0.7388274 -1.083000 -0.4220831 -1.892993 -1.176056 0.05527974
[2,] -1.424290 -0.7591376 -1.439202 -0.6508838 -1.981711 -1.397390 0.18112357
              PC29          PC30
[1,] 0.2658028 0.05162840
[2,] 0.2842191 0.02734355

```

```

plot(wisc.pr$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?

1 should be prioritized