Class 8: Breast Cancer Mini-Project

Kevin Tan (PID: A16774162)

About

In today's lab we will work with fine needle aspiration (FNA) of breast mass data from the University of Wisconsin.

Data Import

```
fna.data <- read.csv("WisconsinCancer.csv")

wisc.df <- data.frame(fna.data, 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
```

Q. How many observations/patients/indivduals/samples 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.df)
```

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

```
inds <- grep("_mean", colnames(wisc.df))
length(inds)
```

```
[1] 10
```

Initial Analysis

Before analysis I want to take out the expert diagnosis column (aka the answer) from our dataset.

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

```
[1] M M M M M M
Levels: B M
```

```
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
```

Clustering

We can try a kmeans() clustering first

```
km <- kmeans(wisc.data, centers = 2)
km
```

```
K-means clustering with 2 clusters of sizes 438, 131

Cluster means:
  radius_mean texture_mean perimeter_mean area_mean smoothness_mean
1    12.55630     18.57037       81.12347  496.0619       0.0948845
```

```
2    19.37992     21.69458      128.23130 1185.9298       0.1012946
  compactness_mean concavity_mean concave.points_mean symmetry_mean
1       0.09109982     0.06243776          0.03343254     0.1780580
2       0.14861298     0.17693947          0.10069878     0.1915397
  fractal_dimension_mean radius_se texture_se perimeter_se  area_se
1             0.06345402 0.3041909   1.215153     2.152881 23.78529
2             0.06060290 0.7428038   1.222538     5.250580 95.67817
  smoothness_se compactness_se concavity_se concave.points_se symmetry_se
1   0.007173263     0.02347469   0.02874551        0.01063632  0.02061358
2   0.006598687     0.03217669   0.04241977        0.01567398  0.02030397
  fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
1          0.003747503     14.04390      24.70954        91.93751   619.6479
2          0.003953389     23.70947      28.91267       158.49618  1753.0229
  smoothness_worst compactness_worst concavity_worst concave.points_worst
1        0.1299591         0.2233118       0.2192149           0.09132984
2        0.1404247         0.3577577       0.4493061           0.19243107
  symmetry_worst fractal_dimension_worst
1      0.2835537              0.08328194
2      0.3118817              0.08616550

Clustering vector:
   842302    842517  84300903  84348301  84358402    843786    844359  84458202
        2         2         2         1         2         1         2         1
   844981  84501001    845636  84610002    846226    846381  84667401  84799002
        1         1         1         2         2         1         1         1
   848406  84862001    849014   8510426   8510653   8510824   8511133    851509
        1         2         2         1         1         1         1         2
   852552    852631    852763    852781    852973    853201    853401    853612
        2         2         1         2         2         2         2         1
 85382601    854002    854039    854253    854268    854941    855133    855138
        2         2         2         2         1         1         1         1
   855167    855563    855625    856106  85638502    857010  85713702     85715
        1         1         2         1         1         2         1         1
   857155    857156    857343    857373    857374    857392    857438  85759902
        1         1         1         1         1         2         1         1
   857637    857793    857810    858477    858970    858981    858986    859196
        2         1         1         1         1         1         1         1
 85922302    859283    859464    859465    859471    859487    859575    859711
        1         1         1         1         1         1         2         1
   859717    859983   8610175   8610404   8610629   8610637   8610862   8610908
        2         1         1         2         1         2         2         1
   861103   8611161   8611555   8611792   8612080   8612399  86135501  86135502
        1         1         2         2         1         2         1         2
```

861597	861598	861648	861799	861853	862009	862028	86208
1	1	1	1	1	1	1	2
86211	862261	862485	862548	862717	862722	862965	862980
1	1	1	1	1	1	1	1
862989	863030	863031	863270	86355	864018	864033	86408
1	1	1	1	2	1	1	1
86409	864292	864496	864685	864726	864729	864877	865128
1	1	1	1	1	1	2	2
865137	86517	865423	865432	865468	86561	866083	866203
1	2	2	1	1	1	1	2
866458	866674	866714	8670	86730502	867387	867739	868202
1	2	1	1	1	1	2	1
868223	868682	868826	868871	868999	869104	869218	869224
1	1	1	1	1	2	1	1
869254	869476	869691	86973701	86973702	869931	871001501	871001502
1	1	1	1	1	1	1	1
8710441	87106	8711002	8711003	8711202	8711216	871122	871149
1	1	1	1	2	1	1	1
8711561	8711803	871201	8712064	8712289	8712291	87127	8712729
1	2	2	1	2	1	1	2
8712766	8712853	87139402	87163	87164	871641	871642	872113
2	1	1	1	1	1	1	1
872608	87281702	873357	873586	873592	873593	873701	873843
1	1	1	1	2	2	2	1
873885	874158	874217	874373	874662	874839	874858	875093
1	1	2	1	1	1	1	1
875099	875263	87556202	875878	875938	877159	877486	877500
1	1	1	1	1	2	2	1
877501	877989	878796	87880	87930	879523	879804	879830
1	2	2	1	1	1	1	2
8810158	8810436	881046502	8810528	8810703	881094802	8810955	8810987
1	1	2	1	2	1	1	1
8811523	8811779	8811842	88119002	8812816	8812818	8812844	8812877
1	1	2	2	1	1	1	1
8813129	88143502	88147101	88147102	88147202	881861	881972	88199202
1	1	1	1	1	1	2	1
88203002	88206102	882488	88249602	88299702	883263	883270	88330202
1	2	1	1	2	2	1	2
88350402	883539	883852	88411702	884180	884437	884448	884626
1	1	1	1	2	1	1	1
88466802	884689	884948	88518501	885429	8860702	886226	886452
1	1	2	1	2	2	2	1
88649001	886776	887181	88725602	887549	888264	888570	889403

2	1	2	1	2	2	2	1
889719	88995002	8910251	8910499	8910506	8910720	8910721	8910748
2	2	1	1	1	1	1	1
8910988	8910996	8911163	8911164	8911230	8911670	8911800	8911834
2	1	2	1	1	2	1	1
8912049	8912055	89122	8912280	8912284	8912521	8912909	8913
2	1	2	1	1	1	1	1
8913049	89143601	89143602	8915	891670	891703	891716	891923
1	1	1	1	1	1	1	1
891936	892189	892214	892399	892438	892604	89263202	892657
1	1	1	1	2	1	2	1
89296	893061	89344	89346	893526	893548	893783	89382601
1	1	1	1	1	1	1	1
89382602	893988	894047	894089	894090	894326	894329	894335
1	1	1	1	1	2	1	1
894604	894618	894855	895100	89511501	89511502	89524	895299
1	2	1	2	1	1	1	1
8953902	895633	896839	896864	897132	897137	897374	89742801
1	1	1	1	1	1	1	2
897604	897630	897880	89812	89813	898143	89827	898431
1	2	1	2	1	1	1	2
89864002	898677	898678	89869	898690	899147	899187	899667
1	1	1	1	1	1	1	1
899987	9010018	901011	9010258	9010259	901028	9010333	901034301
2	1	1	1	1	1	1	1
901034302	901041	9010598	9010872	9010877	901088	9011494	9011495
1	1	1	1	1	2	2	1
9011971	9012000	9012315	9012568	9012795	901288	9013005	901303
2	2	1	1	2	2	1	1
901315	9013579	9013594	9013838	901549	901836	90250	90251
1	1	1	1	1	1	1	1
902727	90291	902975	902976	903011	90312	90317302	903483
1	1	1	1	1	2	1	1
903507	903516	903554	903811	90401601	90401602	904302	904357
2	2	1	1	1	1	1	1
90439701	904647	904689	9047	904969	904971	905189	905190
2	1	1	1	1	1	1	1
90524101	905501	905502	905520	905539	905557	905680	905686
2	1	1	1	1	1	1	1
905978	90602302	906024	906290	906539	906564	906616	906878
1	2	1	1	1	1	1	1
907145	907367	907409	90745	90769601	90769602	907914	907915
1	1	1	1	1	1	1	1

```
   908194    908445    908469    908489    908916    909220    909231    909410
        2         2         1         1         1         1         1         1
   909411    909445  90944601    909777   9110127   9110720   9110732   9110944
        1         2         1         1         2         1         2         1
   911150 911157302   9111596   9111805   9111843    911201    911202   9112085
        1         2         1         2         1         1         1         1
  9112366   9112367   9112594   9112712 911296201 911296202   9113156 911320501
        1         1         1         1         2         2         1         1
911320502   9113239   9113455   9113514   9113538    911366   9113778   9113816
        1         1         1         1         2         1         1         1
   911384   9113846    911391    911408    911654    911673    911685    911916
        1         1         1         1         1         1         1         1
   912193     91227    912519    912558    912600    913063    913102    913505
        1         1         1         1         1         1         1         2
   913512    913535  91376701  91376702    914062    914101    914102    914333
        1         1         1         2         2         1         1         1
   914366    914580    914769     91485    914862     91504     91505    915143
        1         1         2         2         1         1         1         2
   915186    915276  91544001  91544002    915452    915460     91550    915664
        1         1         1         1         1         1         1         1
   915691    915940  91594602    916221    916799    916838    917062    917080
        1         1         1         1         2         2         1         1
   917092  91762702     91789    917896    917897     91805  91813701  91813702
        1         2         1         1         1         1         1         1
   918192    918465     91858  91903901  91903902  91930402    919537    919555
        1         1         1         1         1         2         1         2
 91979701    919812    921092    921362    921385    921386    921644    922296
        1         1         1         1         1         1         1         1
   922297    922576    922577    922840    923169    923465    923748    923780
        1         1         1         1         1         1         1         1
   924084    924342    924632    924934    924964    925236    925277    925291
        1         1         1         1         1         1         1         1
   925292    925311    925622    926125    926424    926682    926954    927241
        1         1         1         2         2         2         1         2
    92751
        1

Within cluster sum of squares by cluster:
[1] 28559677 49383423
 (between_SS / total_SS =  69.6 %)

Available components:
```

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```
table(km$cluster)
```

```
  1   2
438 131
```

Cross-table

```
table(km$cluster, diagnosis)
```

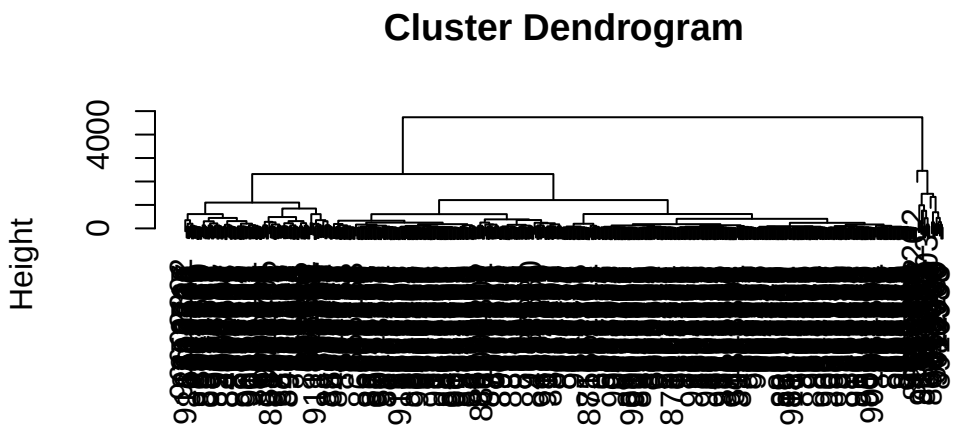
```
   diagnosis
      B   M
  1 356  82
  2   1 130
```

Let's try `hclust()` the key input required for `hclust()` is a distance matrix as produced by the `dist()` function

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

I cna make a tree like figure

```
plot(hc)
```



PCA

Do we need to scale the data?

We can look a the sd of each column (original variable)

```
round(apply(wisc.data, 2, sd))
```

```
            radius_mean            texture_mean          perimeter_mean
                      4                       4                      24
              area_mean         smoothness_mean        compactness_mean
                    352                       0                       0
         concavity_mean     concave.points_mean          symmetry_mean
                      0                       0                       0
fractal_dimension_mean               radius_se              texture_se
                      0                       0                       1
          perimeter_se                 area_se           smoothness_se
                      2                      45                       0
         compactness_se            concavity_se       concave.points_se
                      0                       0                       0
           symmetry_se     fractal_dimension_se            radius_worst
                      0                       0                       5
```

```
         texture_worst         perimeter_worst              area_worst
                     6                      34                     569
     smoothness_worst       compactness_worst         concavity_worst
                     0                       0                       0
  concave.points_worst         symmetry_worst fractal_dimension_worst
                     0                       0                       0
```

Yes we need to scale. We will run `prcomp()` with `scale=TRUE`.

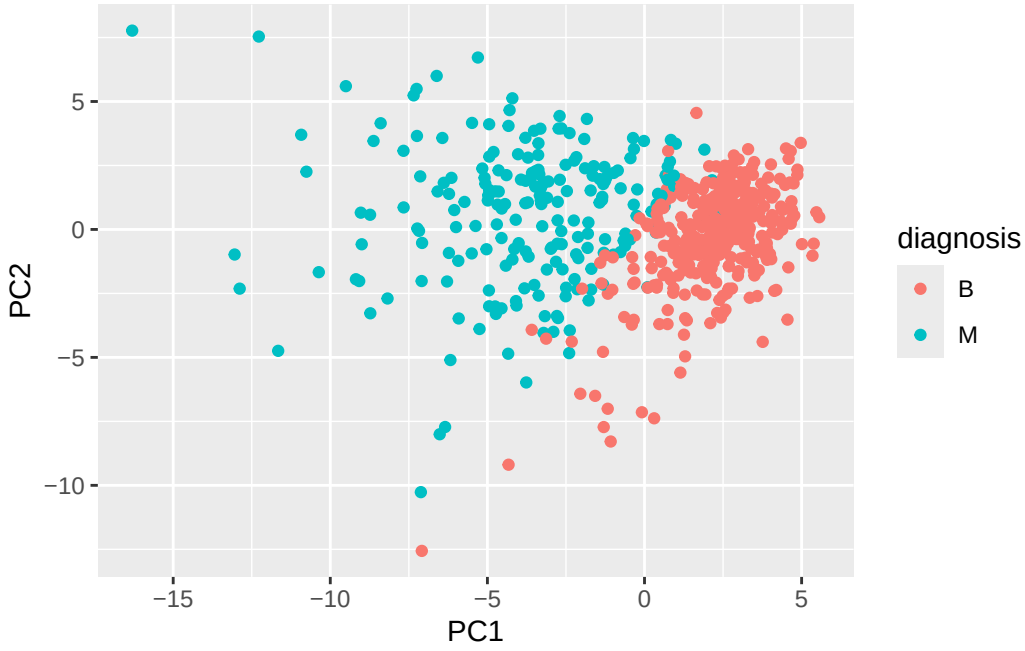
```
wisc.pr <- prcomp(wisc.data, scale=T)
summary(wisc.pr)
```

```
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
```

Generate our main PCA plot (score plot, PC1 vs PC2 plot)...

```
library (ggplot2)

res <- as.data.frame(wisc.pr$x)
ggplot(res) +
  aes(x=PC1, y=PC2, col = diagnosis) +
  geom_point()
```



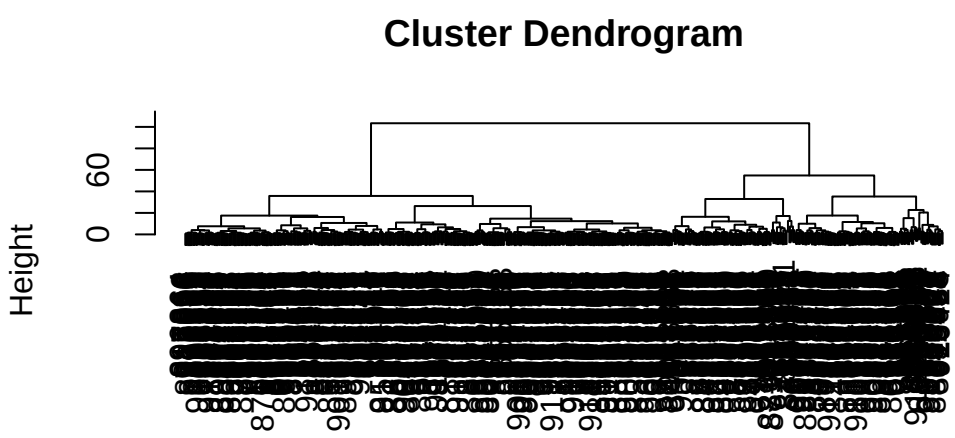
Combininng Methods

Clustering on PCA results

Using the minimum number of principal components required to describe at least 90% of the variability in the data, create a hierarchical clustering model with the linkage method="ward.D2". We use Ward's criterion here because it is based on multidimensional variance like principal components analysis. Assign the results to wisc.pr.hclust.

We need 7 PCs to account for 90% of the variability

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



To get my clustering result/membership factor I need to "cut" the tree with the `cutree()` function.

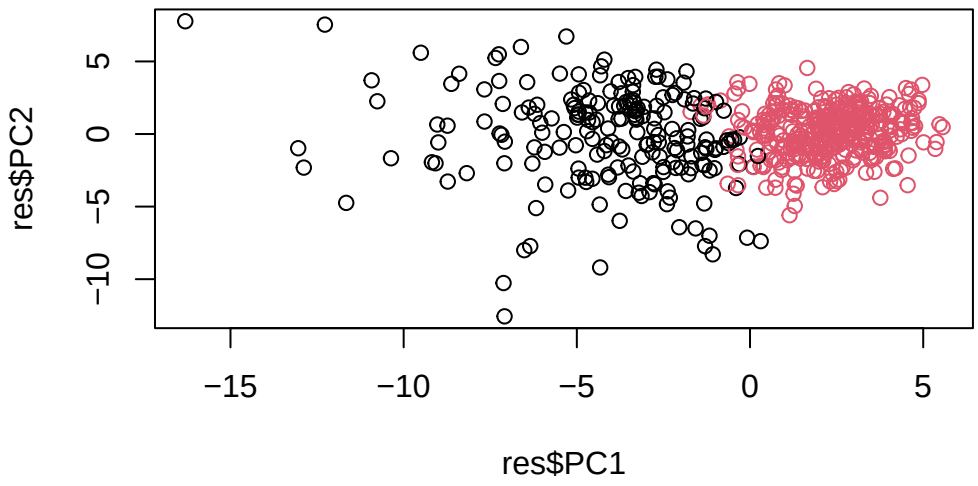
```
grps <- cutree(hc, k=2)
```

Q. How many patients are in each cluster group?

```
table(grps)
```

```
grps
  1   2
203 366
```

```
plot(res$PC1, res$PC2, col=grps)
```



Prediction

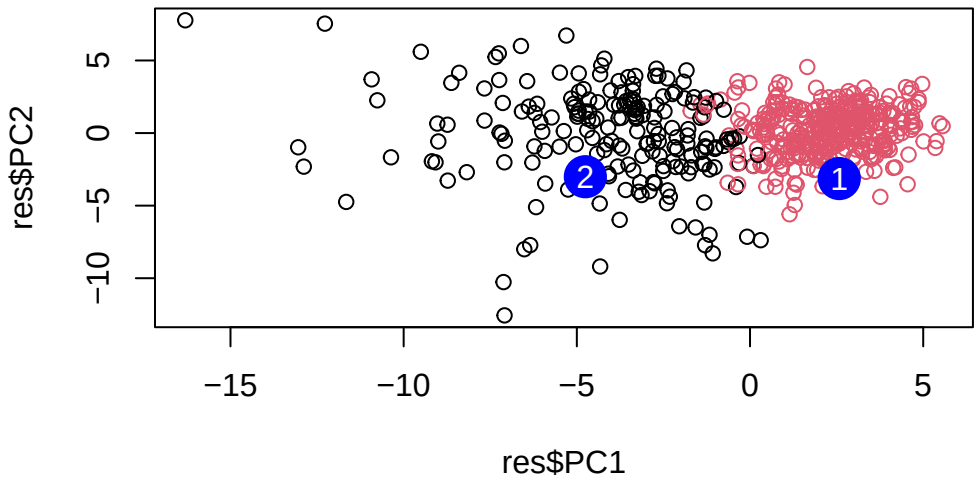
We can use our PCA result (model) to do predictions, that is take new unseen data and project it onto our new PC variables

```
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,]  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(res$PC1, res$PC2, col=grps)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], labels=c(1,2), col="white")
```



Q18. Which of these patients should we prioritize for follow up?

Patient 2

Summary

Principal Component Analysis (PCA) is a super useful method for analyzing large datasets. This is used to reduce the dimensions that you are measuring. It works by finding new variables (PCs) that capture the most variance from the original variables in your dataset.