Class10: MacineLearningProject_take2

Rachael McVicar

2/7/2020

Get our input data

Our data for today come from the Wisconsin Breast Cancer Diagnostic Data Set

```
wisc.df <- read.csv("WisconsinCancer.csv")
head(wisc.df)
```

```
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         M       17.99        10.38         122.80    1001.0
## 2   842517         M       20.57        17.77         132.90    1326.0
## 3 84300903         M       19.69        21.25         130.00    1203.0
## 4 84348301         M       11.42        20.38          77.58     386.1
## 5 84358402         M       20.29        14.34         135.10    1297.0
## 6   843786         M       12.45        15.70          82.57     477.1
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
## 4         0.14250          0.28390         0.2414             0.10520
## 5         0.10030          0.13280         0.1980             0.10430
## 6         0.12780          0.17000         0.1578             0.08089
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.2419                0.07871    1.0950     0.9053        8.589
## 2        0.1812                0.05667    0.5435     0.7339        3.398
## 3        0.2069                0.05999    0.7456     0.7869        4.585
## 4        0.2597                0.09744    0.4956     1.1560        3.445
## 5        0.1809                0.05883    0.7572     0.7813        5.438
## 6        0.2087                0.07613    0.3345     0.8902        2.217
##   area_se smoothness_se compactness_se concavity_se concave.points_se
## 1  153.40      0.006399        0.04904      0.05373           0.01587
## 2   74.08      0.005225        0.01308      0.01860           0.01340
## 3   94.03      0.006150        0.04006      0.03832           0.02058
## 4   27.23      0.009110        0.07458      0.05661           0.01867
## 5   94.44      0.011490        0.02461      0.05688           0.01885
## 6   27.19      0.007510        0.03345      0.03672           0.01137
##   symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
## 1     0.03003             0.006193        25.38         17.33          184.60
## 2     0.01389             0.003532        24.99         23.41          158.80
## 3     0.02250             0.004571        23.57         25.53          152.50
## 4     0.05963             0.009208        14.91         26.50           98.87
## 5     0.01756             0.005115        22.54         16.67          152.20
## 6     0.02165             0.005082        15.47         23.75          103.40
##   area_worst smoothness_worst compactness_worst concavity_worst
## 1     2019.0           0.1622            0.6656          0.7119
## 2     1956.0           0.1238            0.1866          0.2416
## 3     1709.0           0.1444            0.4245          0.4504
## 4      567.7           0.2098            0.8663          0.6869
## 5     1575.0           0.1374            0.2050          0.4000
## 6      741.6           0.1791            0.5249          0.5355
##   concave.points_worst symmetry_worst fractal_dimension_worst  X
## 1               0.2654         0.4601                 0.11890 NA
## 2               0.1860         0.2750                 0.08902 NA
## 3               0.2430         0.3613                 0.08758 NA
## 4               0.2575         0.6638                 0.17300 NA
## 5               0.1625         0.2364                 0.07678 NA
## 6               0.1741         0.3985                 0.12440 NA
```

```
wisc.data <- as.matrix(wisc.df[,3:32])
head(wisc.data)
```

```
##      radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## [1,]       17.99        10.38         122.80    1001.0         0.11840
## [2,]       20.57        17.77         132.90    1326.0         0.08474
## [3,]       19.69        21.25         130.00    1203.0         0.10960
## [4,]       11.42        20.38          77.58     386.1         0.14250
## [5,]       20.29        14.34         135.10    1297.0         0.10030
## [6,]       12.45        15.70          82.57     477.1         0.12780
##      compactness_mean concavity_mean concave.points_mean symmetry_mean
## [1,]          0.27760         0.3001             0.14710        0.2419
## [2,]          0.07864         0.0869             0.07017        0.1812
## [3,]          0.15990         0.1974             0.12790        0.2069
## [4,]          0.28390         0.2414             0.10520        0.2597
## [5,]          0.13280         0.1980             0.10430        0.1809
## [6,]          0.17000         0.1578             0.08089        0.2087
##      fractal_dimension_mean radius_se texture_se perimeter_se area_se
## [1,]                0.07871    1.0950     0.9053        8.589  153.40
## [2,]                0.05667    0.5435     0.7339        3.398   74.08
## [3,]                0.05999    0.7456     0.7869        4.585   94.03
## [4,]                0.09744    0.4956     1.1560        3.445   27.23
## [5,]                0.05883    0.7572     0.7813        5.438   94.44
## [6,]                0.07613    0.3345     0.8902        2.217   27.19
##      smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## [1,]      0.006399        0.04904      0.05373           0.01587     0.03003
## [2,]      0.005225        0.01308      0.01860           0.01340     0.01389
## [3,]      0.006150        0.04006      0.03832           0.02058     0.02250
## [4,]      0.009110        0.07458      0.05661           0.01867     0.05963
## [5,]      0.011490        0.02461      0.05688           0.01885     0.01756
## [6,]      0.007510        0.03345      0.03672           0.01137     0.02165
##      fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
## [1,]             0.006193        25.38         17.33          184.60     2019.0
## [2,]             0.003532        24.99         23.41          158.80     1956.0
## [3,]             0.004571        23.57         25.53          152.50     1709.0
## [4,]             0.009208        14.91         26.50           98.87      567.7
## [5,]             0.005115        22.54         16.67          152.20     1575.0
## [6,]             0.005082        15.47         23.75          103.40      741.6
##      smoothness_worst compactness_worst concavity_worst concave.points_worst
## [1,]           0.1622            0.6656          0.7119               0.2654
## [2,]           0.1238            0.1866          0.2416               0.1860
## [3,]           0.1444            0.4245          0.4504               0.2430
## [4,]           0.2098            0.8663          0.6869               0.2575
## [5,]           0.1374            0.2050          0.4000               0.1625
## [6,]           0.1791            0.5249          0.5355               0.1741
##      symmetry_worst fractal_dimension_worst
## [1,]         0.4601                 0.11890
## [2,]         0.2750                 0.08902
## [3,]         0.3613                 0.08758
## [4,]         0.6638                 0.17300
## [5,]         0.2364                 0.07678
## [6,]         0.3985                 0.12440
```

Q. How many patients are there in this dataset?

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

```
## [1] 569
```

Q. How many cancer and non-cancer patients are there?

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

```
##
##   B   M
## 357 212
```

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

```
## [1] 212
```

Q. How many columns are "_mean" values

```
colnames(wisc.df)
```

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

We can use the `grep()` function to see this

```
grep("_mean", colnames(wisc.data), value=TRUE)
```

```
##  [1] "radius_mean"            "texture_mean"            "perimeter_mean"
##  [4] "area_mean"              "smoothness_mean"         "compactness_mean"
##  [7] "concavity_mean"         "concave.points_mean"     "symmetry_mean"
## [10] "fractal_dimension_mean"
```

We can take the `length()` of this to find how many matches there are

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

```
## [1] 10
```

```
#View(wisc.data)
```

Enter Principal Component Analysis

First we need to check whether our input data should be scaled. Lets check the sd() and mean() of all our columns in `wisc.data`

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

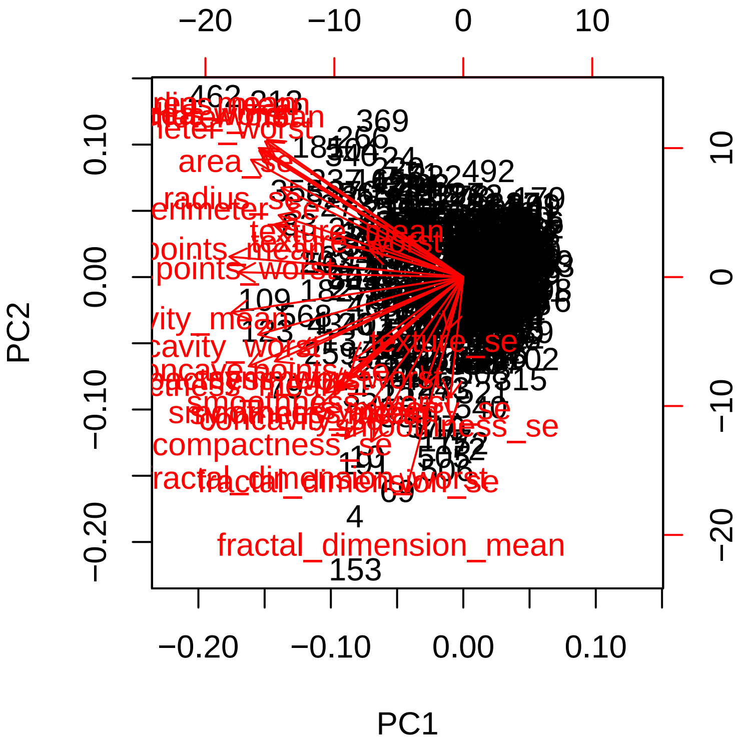
```
##             radius_mean            texture_mean          perimeter_mean 
##                    3.52                    4.30                   24.30 
##               area_mean         smoothness_mean        compactness_mean 
##                  351.91                    0.01                    0.05 
##          concavity_mean     concave.points_mean           symmetry_mean 
##                    0.08                    0.04                    0.03 
##  fractal_dimension_mean               radius_se              texture_se 
##                    0.01                    0.28                    0.55 
##            perimeter_se                 area_se           smoothness_se 
##                    2.02                   45.49                    0.00 
##          compactness_se            concavity_se       concave.points_se 
##                    0.02                    0.03                    0.01 
##             symmetry_se    fractal_dimension_se            radius_worst 
##                    0.01                    0.00                    4.83 
##           texture_worst         perimeter_worst              area_worst 
##                    6.15                   33.60                  569.36 
##        smoothness_worst       compactness_worst         concavity_worst 
##                    0.02                    0.16                    0.21 
##    concave.points_worst          symmetry_worst fractal_dimension_worst 
##                    0.07                    0.06                    0.02
```

after class break

```
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
```

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



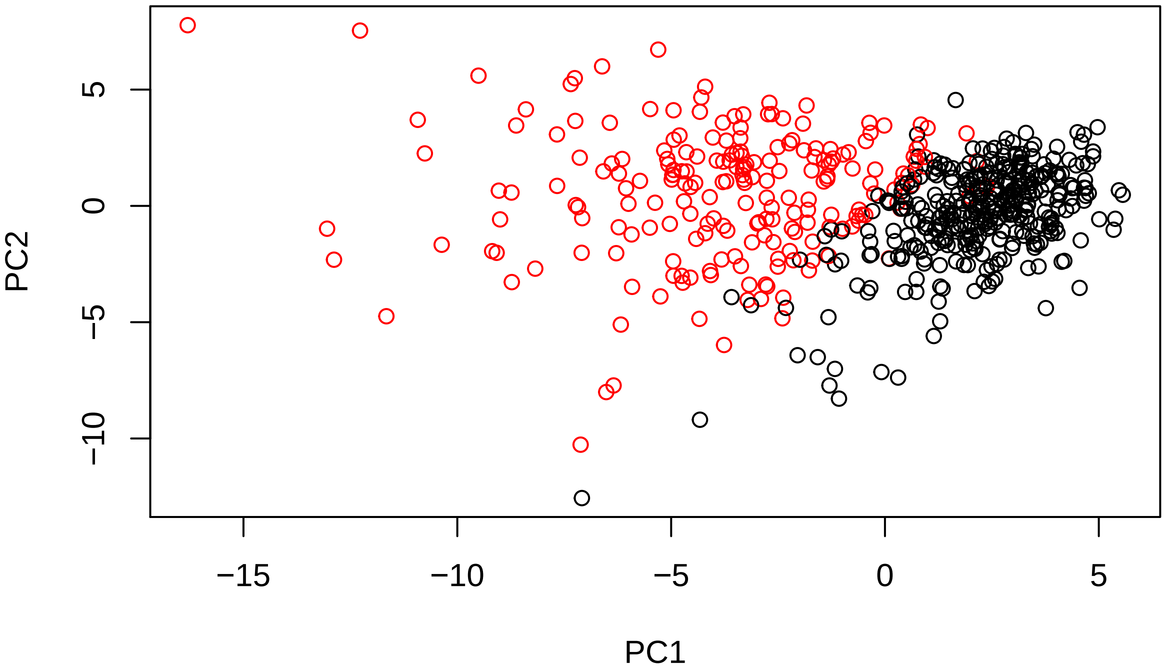
Biplot just doesn't cut it, time to look at this using PCA. First need to access the results within the `wisc.pr` object.

```
attributes(wisc.pr)
```

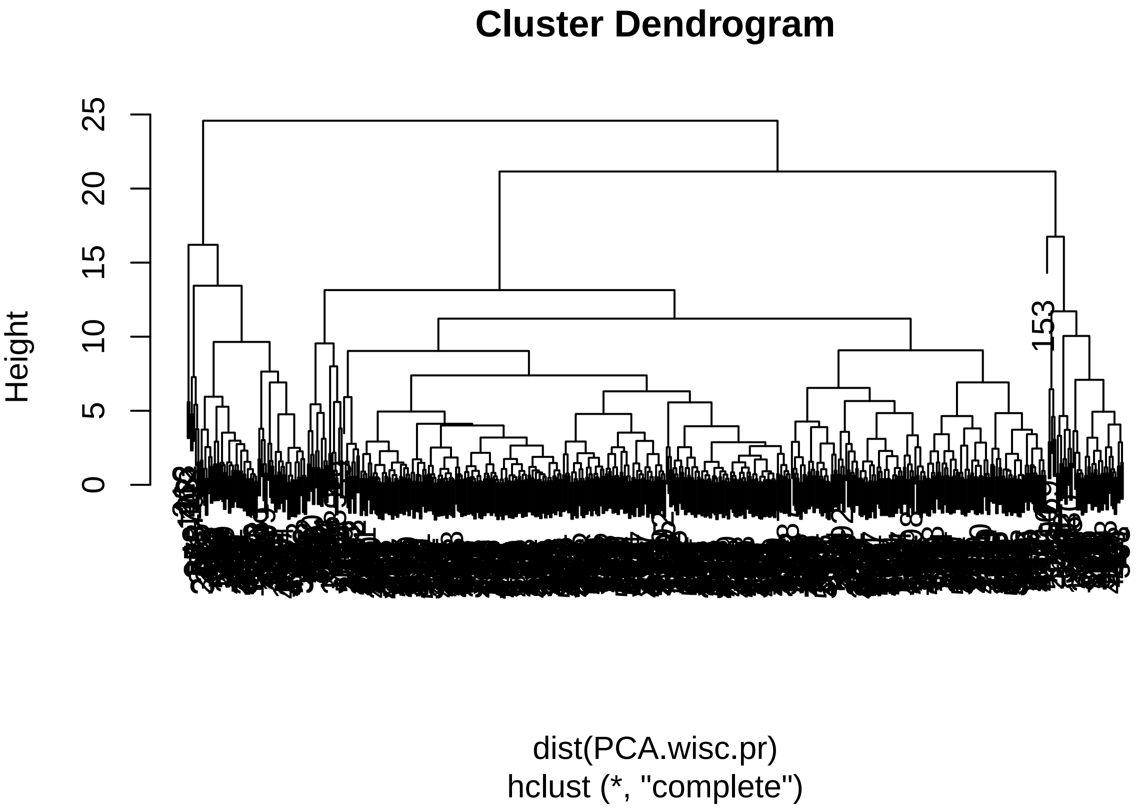
```
## $names
## [1] "sdev"     "rotation" "center"   "scale"    "x"
##
## $class
## [1] "prcomp"
```

We want the $x component to make our PCA plot!

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

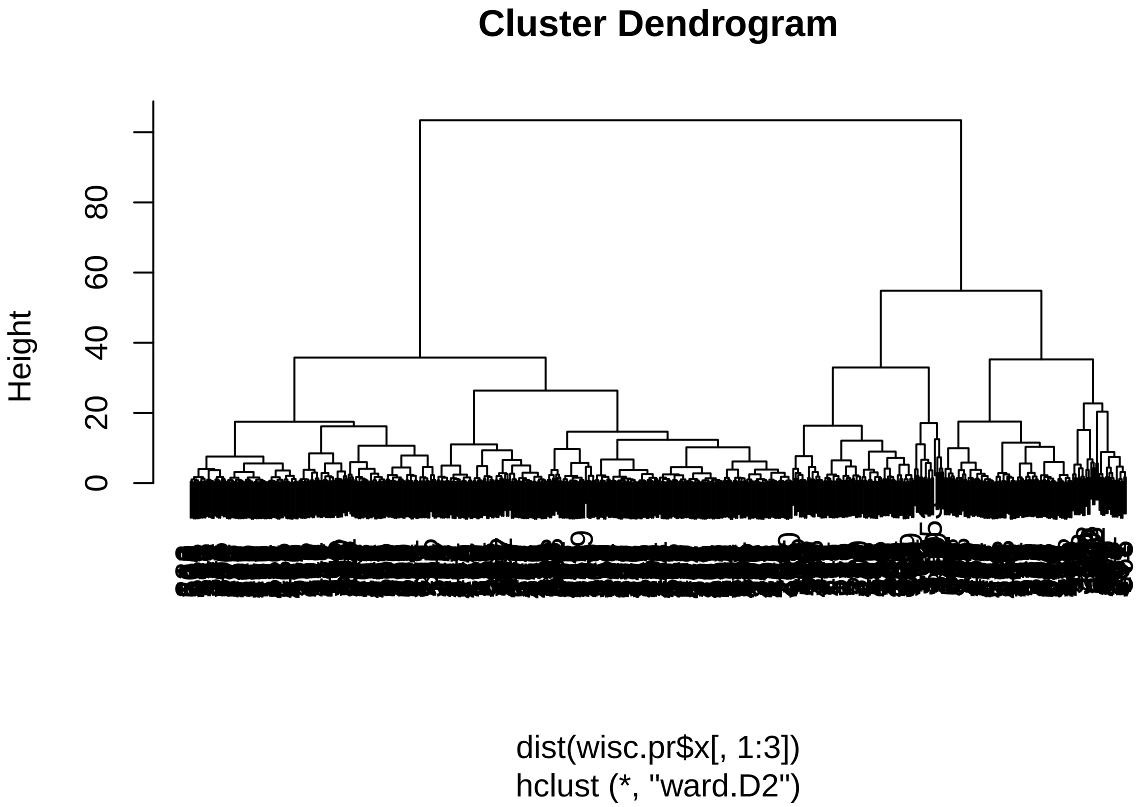


```r
PCA.wisc.pr<- wisc.pr$x[,1:3]
hclust.wisc.pr<- hclust(dist(PCA.wisc.pr))
#View(hclust.wisc.pr)
plot(hclust.wisc.pr)
```



I don't know where it is good to *cut* a tree like this…

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



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

```
## grps
##   1   2
## 203 366
```

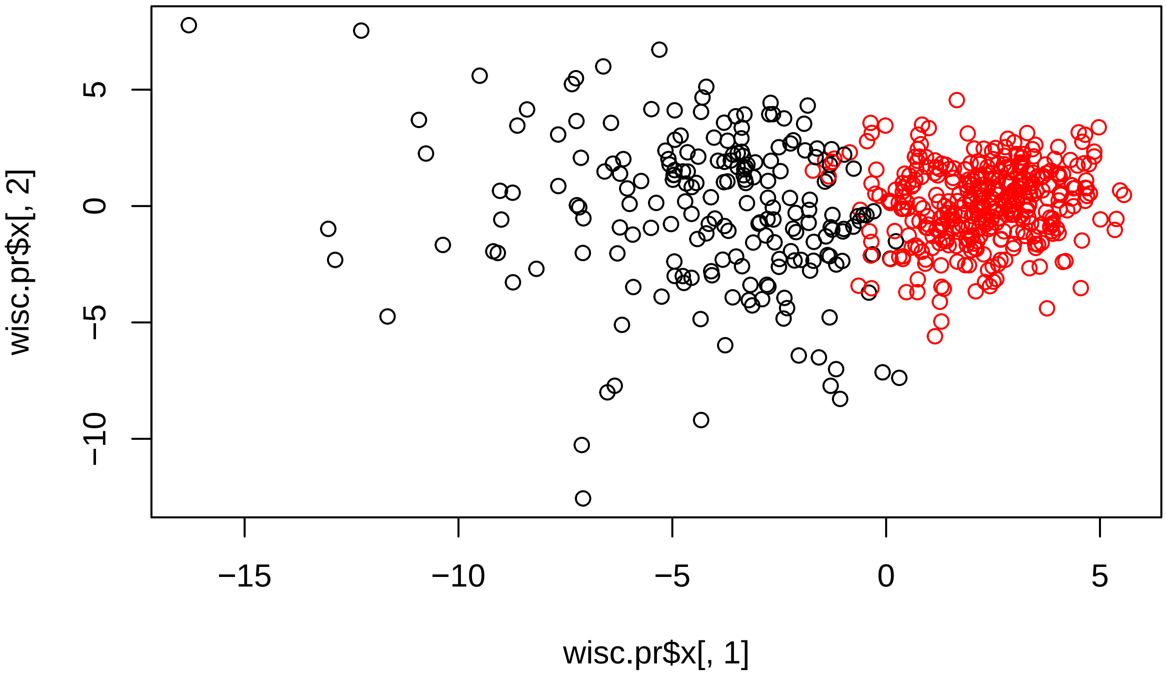
```r
table(grps, wisc.df$diagnosis)
```

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

```r
kmeans(wisc.data, centers = 2)
```

```
## 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:
##   [1] 2 2 2 1 2 1 2 1 1 1 1 2 2 1 1 1 1 2 2 1 1 1 1 2 2 2 1 2 2 2 2 1 2 2 2 2 1
##  [38] 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1
##  [75] 1 2 1 2 2 1 1 1 2 2 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
## [112] 1 1 1 1 1 1 1 2 2 1 2 2 1 1 1 1 2 1 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 2 1 1 1 1 2 2 1 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1
## [186] 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 2 1 1 2 1 2 1 1 1 1 1 2 2 1 1
## [223] 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 2 1 2 1 1 1 1 2 1 1 1 1 1 2 1 2 2 2 1 2 1 2
## [260] 1 2 2 2 1 2 2 1 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
## [297] 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1
## [334] 1 1 2 1 2 1 2 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2
## [371] 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1
## [408] 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1
## [445] 2 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
## [482] 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2
## [519] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [556] 1 1 1 1 1 1 1 1 2 2 2 1 2 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"
```

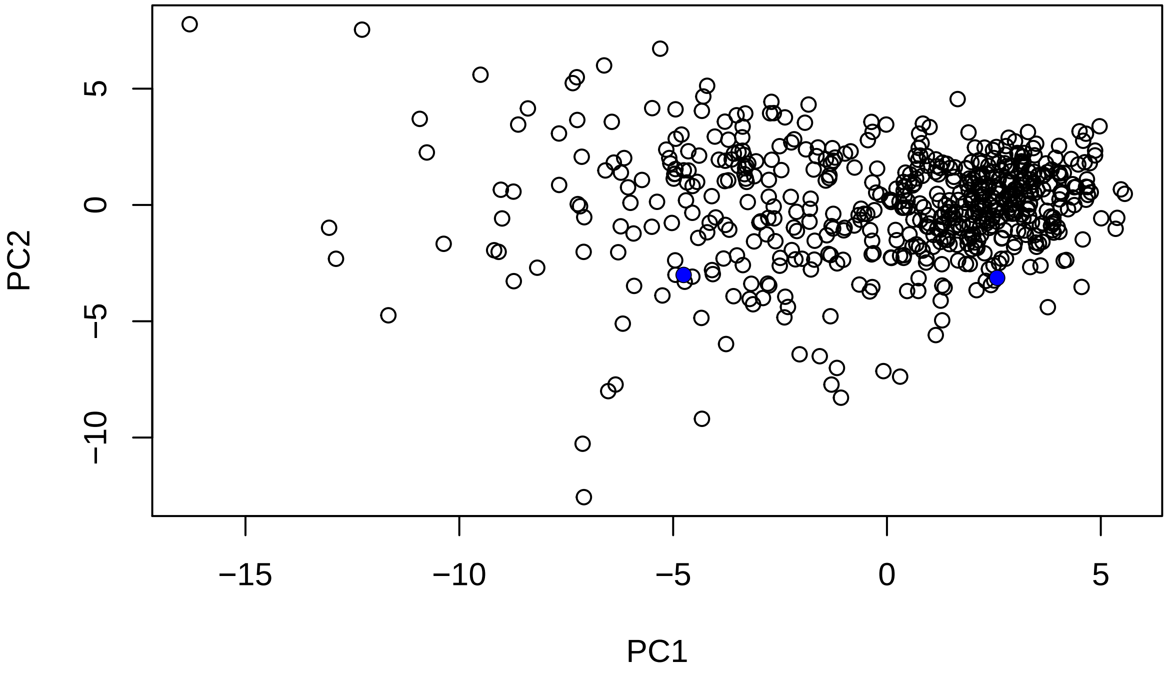
```r
plot(wisc.pr$x[,1], wisc.pr$x[,2], col=grps)
```



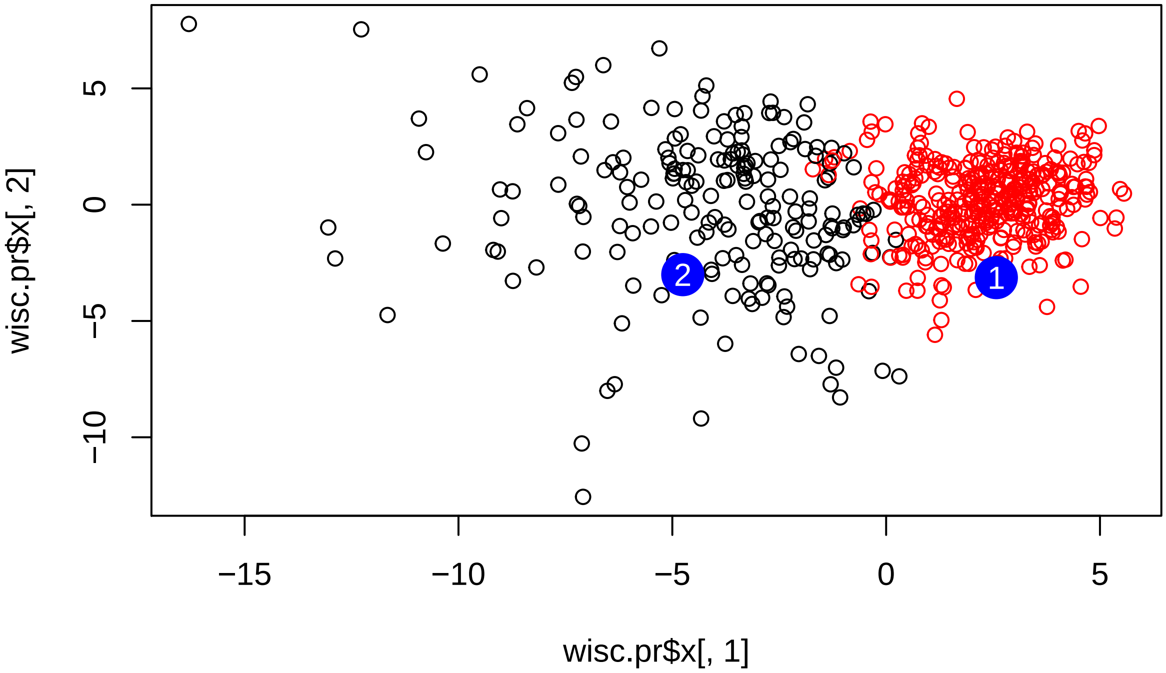
one last hoorah

Predicting Malignancy Of New samples

```
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
plot(wisc.pr$x[,1:2])
points(npc[,1], npc[,2], col="blue", pch=16)
```



```
plot(wisc.pr$x[,1], wisc.pr$x[,2], col=grps)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], labels=c(1,2), col="white")
```


wisc.pr$x[, 2]
wisc.pr$x[, 1]