classs099miniproject

Nicole Manuguid

10/26/2021

Mini Project

Exploratory Data Analysis

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

```
#save input data file into project directory

fna.data <- "WisconsinCancer.csv"
```

```
#input data and store as wisc.df

wisc.df <- read.csv(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
```

```
# omit first column
wisc.data <- wisc.df[,-1]
```

```
# create vector for diagnosis
diagnosis <- as.factor(wisc.df$diagnosis)
```

> Q1. Ho many observations are in this dataset?

569 observations

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

```
table(diagnosis)
```

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

There are 212 observations with malignant diagnosis.

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

```
library(stringr)
colnames(wisc.data)
```

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

```
sum(str_count(colnames(wisc.data), "_mean"))
```

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

There are 10 variables with "_mean".

```
#can also use grep() to find the number of variables with suffix "mean"
length(grep("mean", colnames(wisc.df)))
```

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

Principal Component Analysis

```
#check column means 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
```

```
#perform PCA on wisc.data
wisc.pr <- prcomp(wisc.data)
```

```
#summary of results
summary(wisc.pr)
```

```
## Importance of components:
##                            PC1      PC2      PC3     PC4     PC5     PC6   PC7
## Standard deviation     666.170 85.49912 26.52987 7.39248 6.31585 1.73337 1.347
## Proportion of Variance   0.982  0.01618  0.00156 0.00012 0.00009 0.00001 0.000
## Cumulative Proportion    0.982  0.99822  0.99978 0.99990 0.99999 0.99999 1.000
##                           PC8    PC9   PC10   PC11    PC12    PC13    PC14
## Standard deviation     0.6095 0.3944 0.2899 0.1778 0.08659 0.05623 0.04649
## Proportion of Variance 0.0000 0.0000 0.0000 0.0000 0.00000 0.00000 0.00000
## Cumulative Proportion  1.0000 1.0000 1.0000 1.0000 1.00000 1.00000 1.00000
##                           PC15   PC16    PC17    PC18    PC19    PC20     PC21
## Standard deviation     0.03642 0.0253 0.01936 0.01534 0.01359 0.01281 0.008838
## Proportion of Variance 0.00000 0.0000 0.00000 0.00000 0.00000 0.00000 0.000000
## Cumulative Proportion  1.00000 1.0000 1.00000 1.00000 1.00000 1.00000 1.000000
##                           PC22     PC23     PC24     PC25     PC26     PC27
## Standard deviation     0.00759 0.005909 0.005329 0.004018 0.003534 0.001918
## Proportion of Variance 0.00000 0.000000 0.000000 0.000000 0.000000 0.000000
## Cumulative Proportion  1.00000 1.000000 1.000000 1.000000 1.000000 1.000000
##                            PC28     PC29      PC30
## Standard deviation     0.001688 0.001416 0.0008379
## Proportion of Variance 0.000000 0.000000 0.0000000
## Cumulative Proportion  1.000000 1.000000 1.0000000
```

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

98.2%

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

```
# need to scale because data is on different scales, we will use scale = TRUE
summary(prcomp(wisc.data, scale = TRUE))
```

```
## 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
```

At PC3

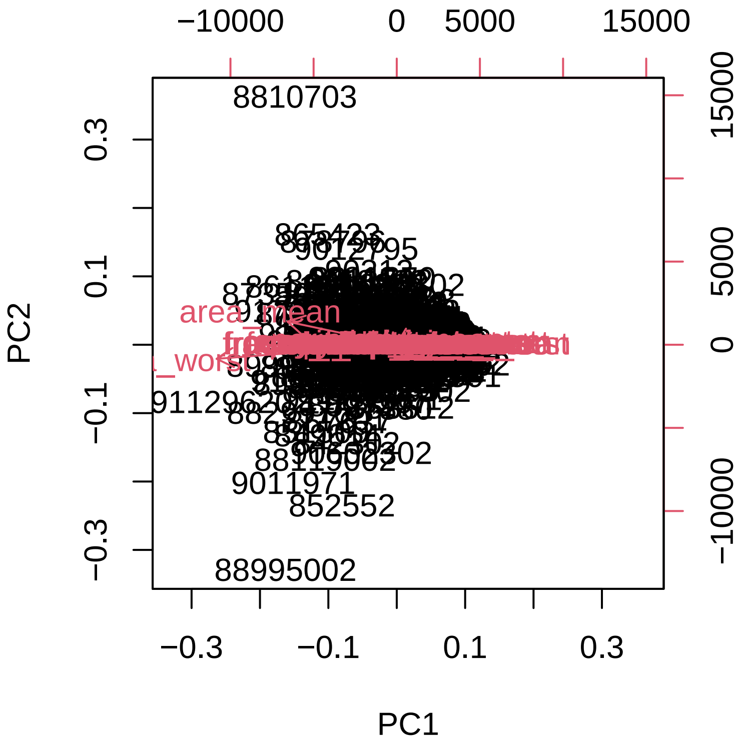
> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

At PC7

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

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

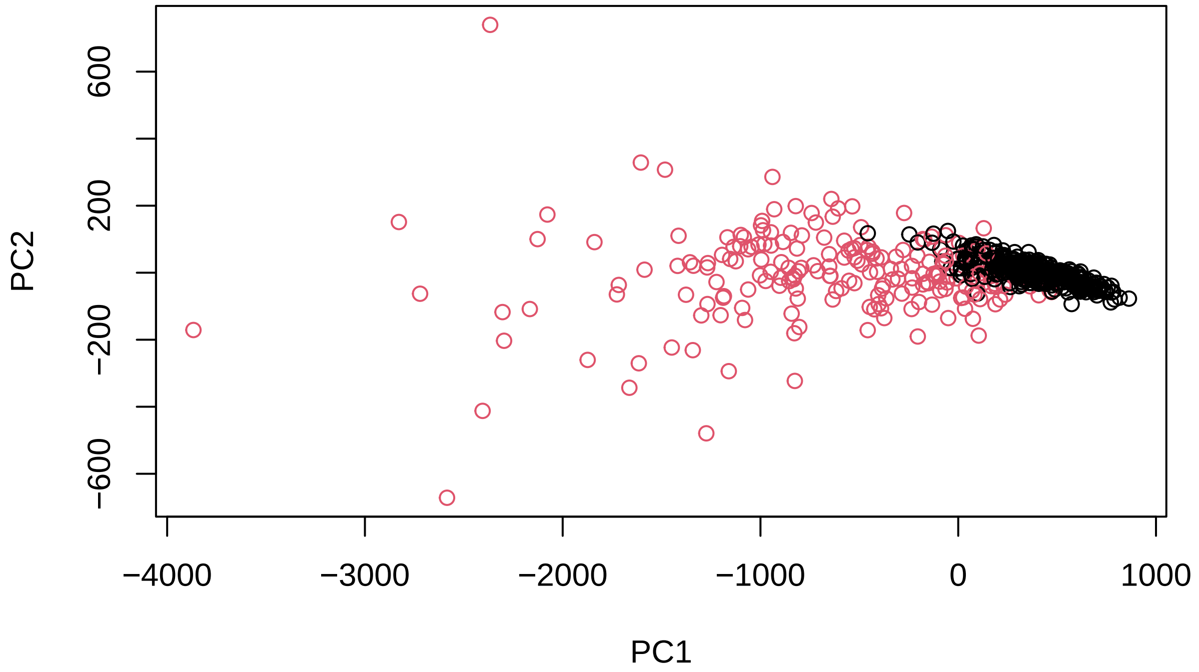
```
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
```



Two sections stand out to me, which are colored into a red and black section. As of right now the plot is difficult to understand. It looks like the red data from PC2 is coming out of PC1.

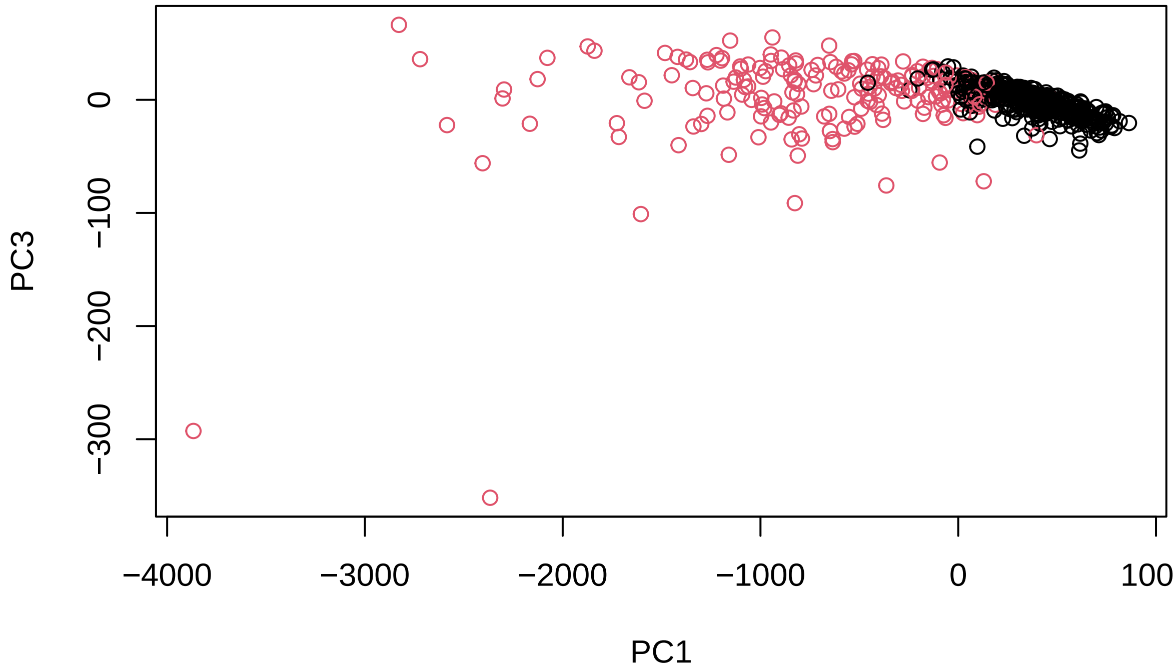
To make this plot ourselves we need access the PCA scores data.

```
# lets make a better plot
# scatter plot observations by components 1 and 2.



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



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")
```

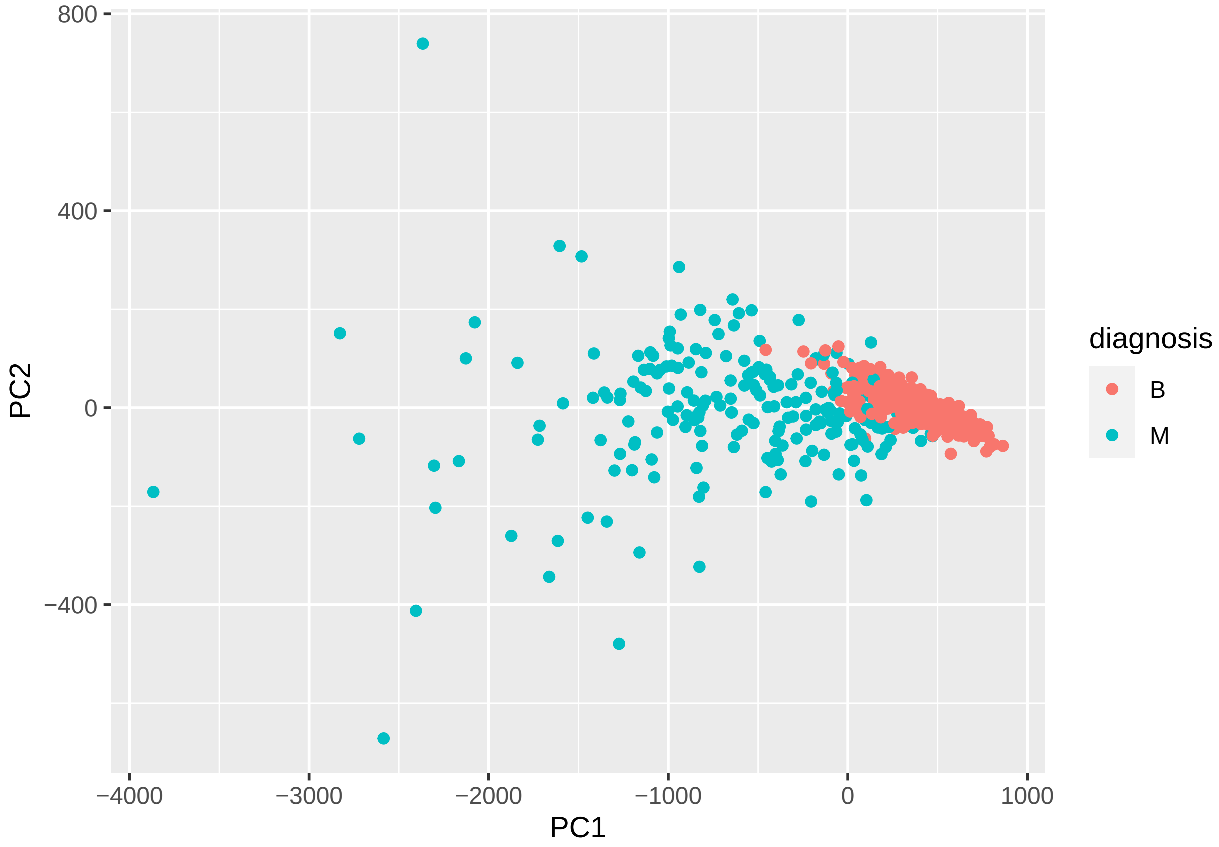


Let's see a ggplot

```
#create a data.frame for ggplot
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis
```

```
#load ggplot package
library(ggplot2)
```

```
#make a scatter plot by diagnosis
ggplot(df) + aes(PC1, PC2, col = diagnosis) + geom_point()
```

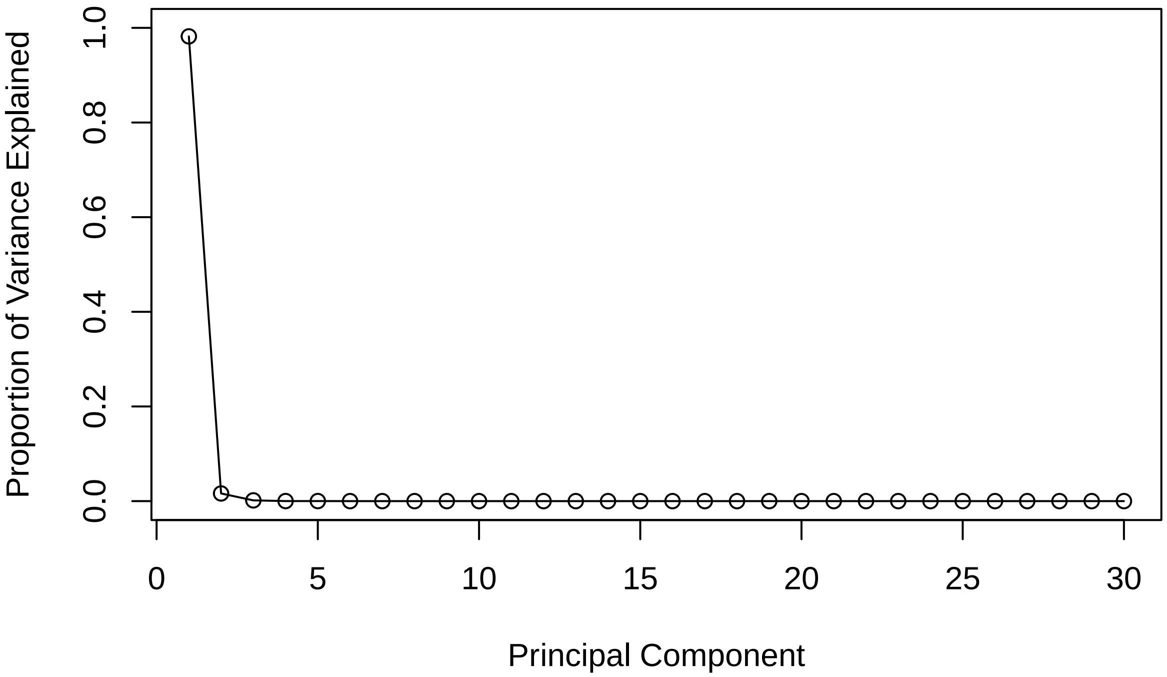


```
# calculate variance of each component

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

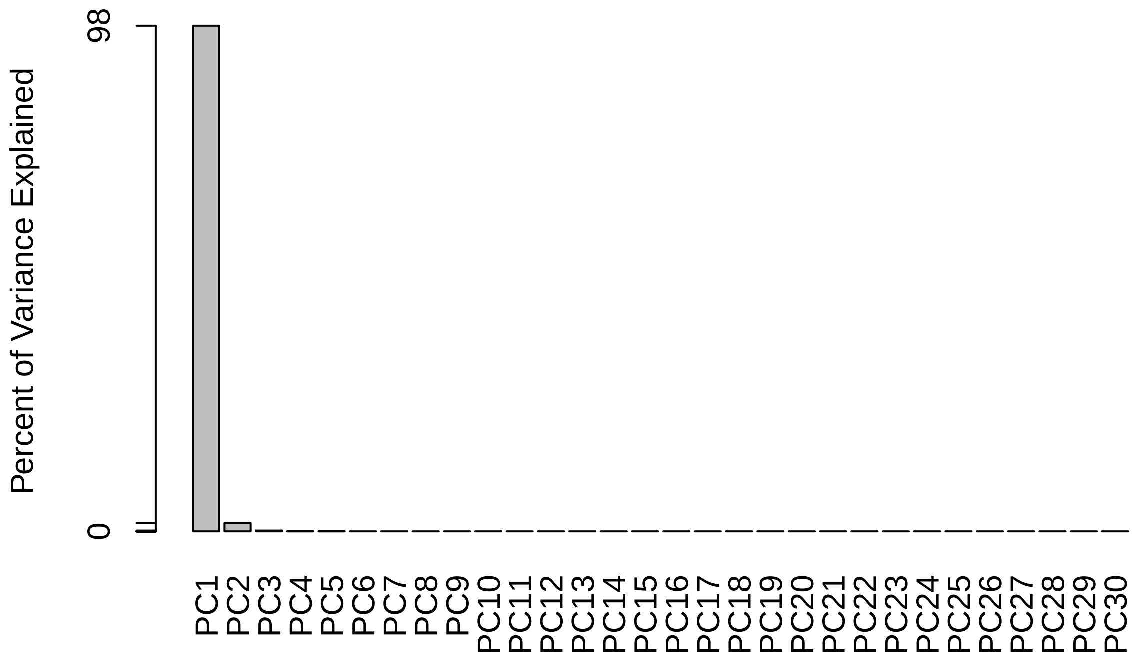
```
## [1] 4.437826e+05 7.310100e+03 7.038337e+02 5.464874e+01 3.989002e+01
## [6] 3.004588e+00
```

```
# variance explained by each principal component

pve <- pr.var / sum(pr.var)
```

```
#plot variance explained by each principal component

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



```
# alternative scree plot of the same data, note date driven y-axis

barplot(pve, ylab = "Percent of Variance Explained", names.arg=paste0("PC",1:length(pve)), las=2, axes =
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["concave.points_mean", 1]
```

```
## [1] -4.778078e-05
```

10. What is the minimum number of principal components required to explain 80% of the variance of the data?

```
var <-summary(wisc.pr)
sum(var$importance[3,]< 0.8)
```

```
## [1] 0
```

```
summary(wisc.pr)
```

```
## Importance of components:
##                             PC1      PC2      PC3     PC4     PC5     PC6   PC7
## Standard deviation     666.170 85.49912 26.52987 7.39248 6.31585 1.73337 1.347
## Proportion of Variance   0.982  0.01618  0.00156 0.00012 0.00009 0.00001 0.000
## Cumulative Proportion    0.982  0.99822  0.99978 0.99990 0.99999 0.99999 1.000
##                            PC8    PC9   PC10   PC11    PC12    PC13    PC14
## Standard deviation     0.6095 0.3944 0.2899 0.1778 0.08659 0.05623 0.04649
## Proportion of Variance 0.0000 0.0000 0.0000 0.0000 0.00000 0.00000 0.00000
## Cumulative Proportion  1.0000 1.0000 1.0000 1.0000 1.00000 1.00000 1.00000
##                           PC15   PC16    PC17    PC18    PC19    PC20     PC21
## Standard deviation     0.03642 0.0253 0.01936 0.01534 0.01359 0.01281 0.008838
## Proportion of Variance 0.00000 0.0000 0.00000 0.00000 0.00000 0.00000 0.000000
## Cumulative Proportion  1.00000 1.0000 1.00000 1.00000 1.00000 1.00000 1.000000
##                           PC22     PC23     PC24     PC25     PC26     PC27
## Standard deviation     0.00759 0.005909 0.005329 0.004018 0.003534 0.001918
## Proportion of Variance 0.00000 0.000000 0.000000 0.000000 0.000000 0.000000
## Cumulative Proportion  1.00000 1.000000 1.000000 1.000000 1.000000 1.000000
##                            PC28     PC29      PC30
## Standard deviation     0.001688 0.001416 0.0008379
## Proportion of Variance 0.000000 0.000000 0.0000000
## Cumulative Proportion  1.000000 1.000000 1.0000000
```

Need at least 5 components (until PC5)

#Hierarchal clustering

```r
# scale the wisc.data using the "scale()" function
data.scaled <- scale(wisc.data)
```

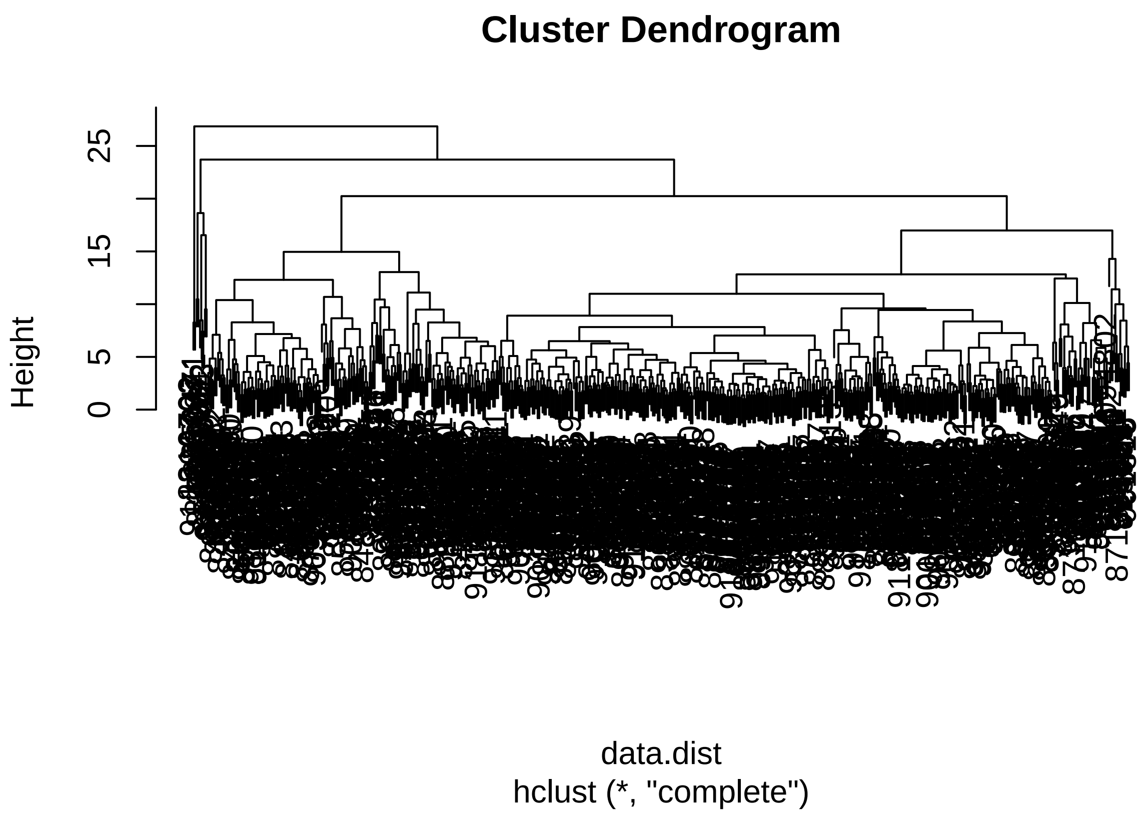
```r
#calculate Euclidean distances
data.dist <- dist(data.scaled)
```

```r
#create hierarchal clustering model
wisc.hclust <- hclust(data.dist)
```

#results of hierarchal clustering

> Q11. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?

```r
plot(wisc.hclust)
```



```
#cut the tree into 4 groups
wisc.hclust.clusters <- cutree(wisc.hclust, k =4)
```

Compare to diagnosis results

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

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

> Q12. Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10?

```
wisc.hclust.clusters <- cutree(wisc.hclust, k =2)
```

k = 4 still works the best

5. COmbining Methods

We take the results of our PCA analysis and cluster in this space 'wisc.pr$x'

```
summary(wisc.pr)
```

```
## Importance of components:
##                             PC1      PC2      PC3     PC4     PC5     PC6   PC7
## Standard deviation     666.170 85.49912 26.52987 7.39248 6.31585 1.73337 1.347
## Proportion of Variance   0.982  0.01618  0.00156 0.00012 0.00009 0.00001 0.000
## Cumulative Proportion    0.982  0.99822  0.99978 0.99990 0.99999 0.99999 1.000
##                           PC8    PC9   PC10   PC11    PC12    PC13    PC14
## Standard deviation     0.6095 0.3944 0.2899 0.1778 0.08659 0.05623 0.04649
## Proportion of Variance 0.0000 0.0000 0.0000 0.0000 0.00000 0.00000 0.00000
## Cumulative Proportion  1.0000 1.0000 1.0000 1.0000 1.00000 1.00000 1.00000
##                           PC15   PC16    PC17    PC18    PC19    PC20     PC21
## Standard deviation     0.03642 0.0253 0.01936 0.01534 0.01359 0.01281 0.008838
## Proportion of Variance 0.00000 0.0000 0.00000 0.00000 0.00000 0.00000 0.000000
## Cumulative Proportion  1.00000 1.0000 1.00000 1.00000 1.00000 1.00000 1.000000
##                           PC22     PC23     PC24     PC25     PC26     PC27
## Standard deviation     0.00759 0.005909 0.005329 0.004018 0.003534 0.001918
## Proportion of Variance 0.00000 0.000000 0.000000 0.000000 0.000000 0.000000
## Cumulative Proportion  1.00000 1.000000 1.000000 1.000000 1.000000 1.000000
##                            PC28     PC29      PC30
## Standard deviation     0.001688 0.001416 0.0008379
## Proportion of Variance 0.000000 0.000000 0.0000000
## Cumulative Proportion  1.000000 1.000000 1.0000000
```

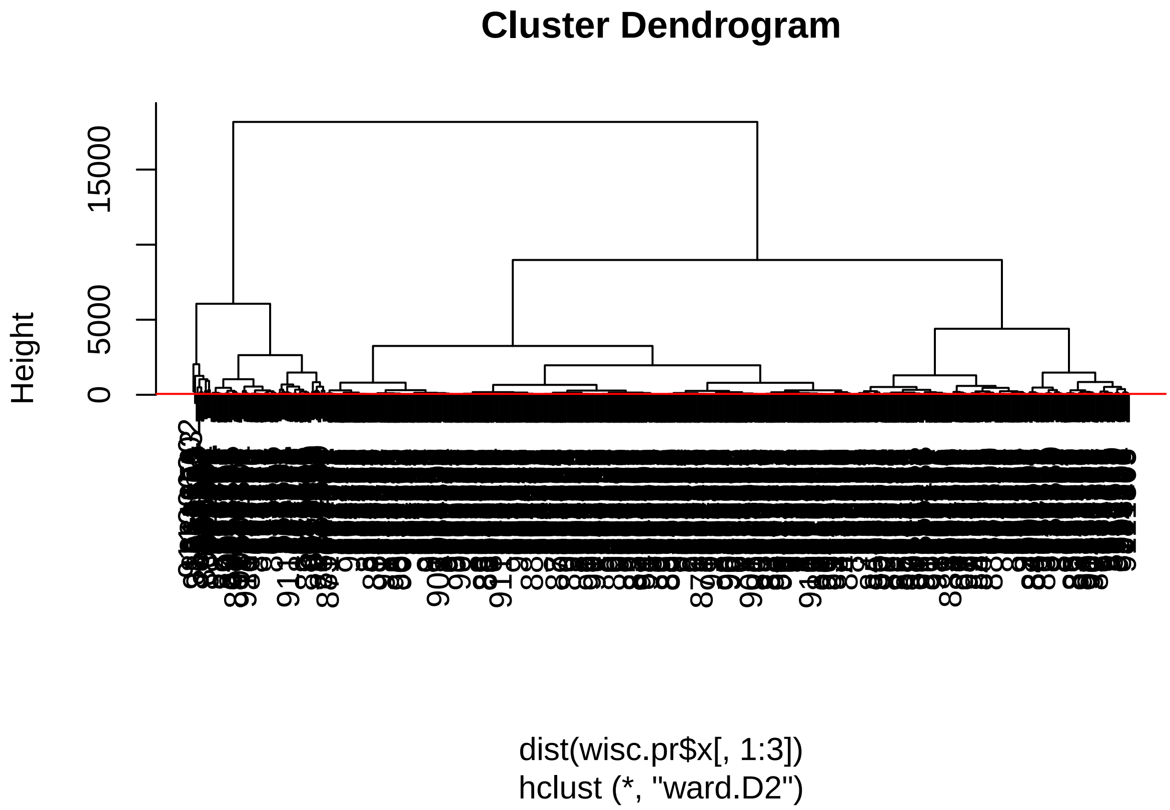
> Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

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

"ward.D2" is able to create groups that have variance minimized within clusters

Plot my dendrogram

```
plot(wisc.pc.hclust )
abline (h=60, col = "red")
```



Cut the tree into k=2 groups

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

```
## grps
##   1   2
##  81 488
```

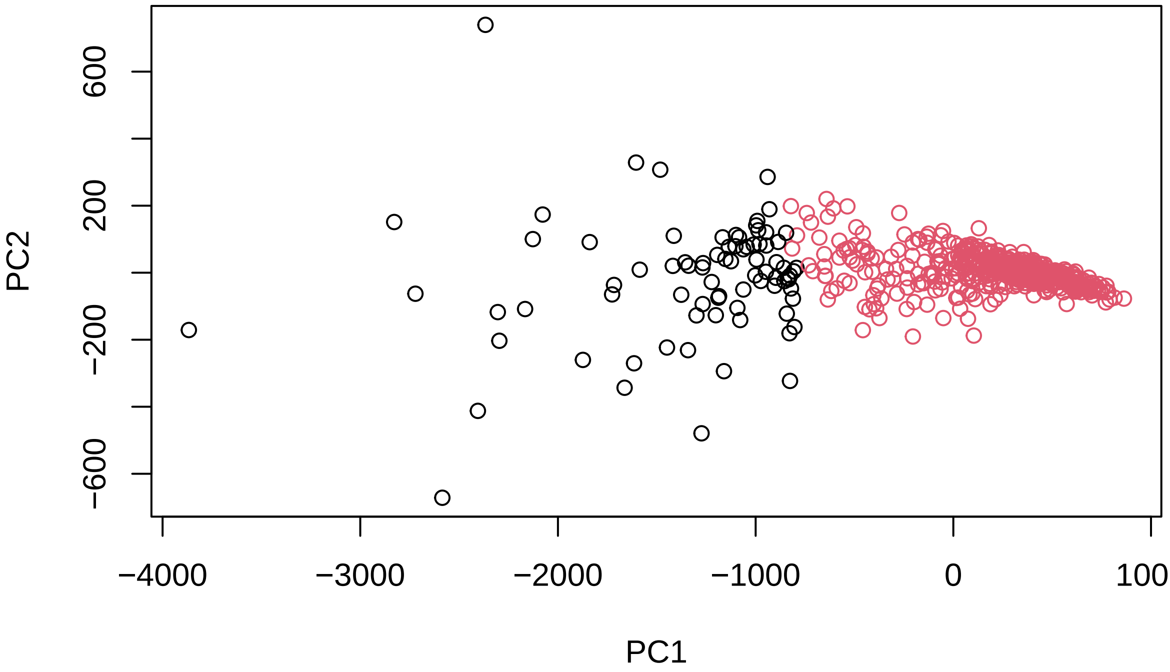
Cross table compare of diagnosis and my cluster groups

> Q15. How well does the newly created model with four clusters separate out the two diagnoses?

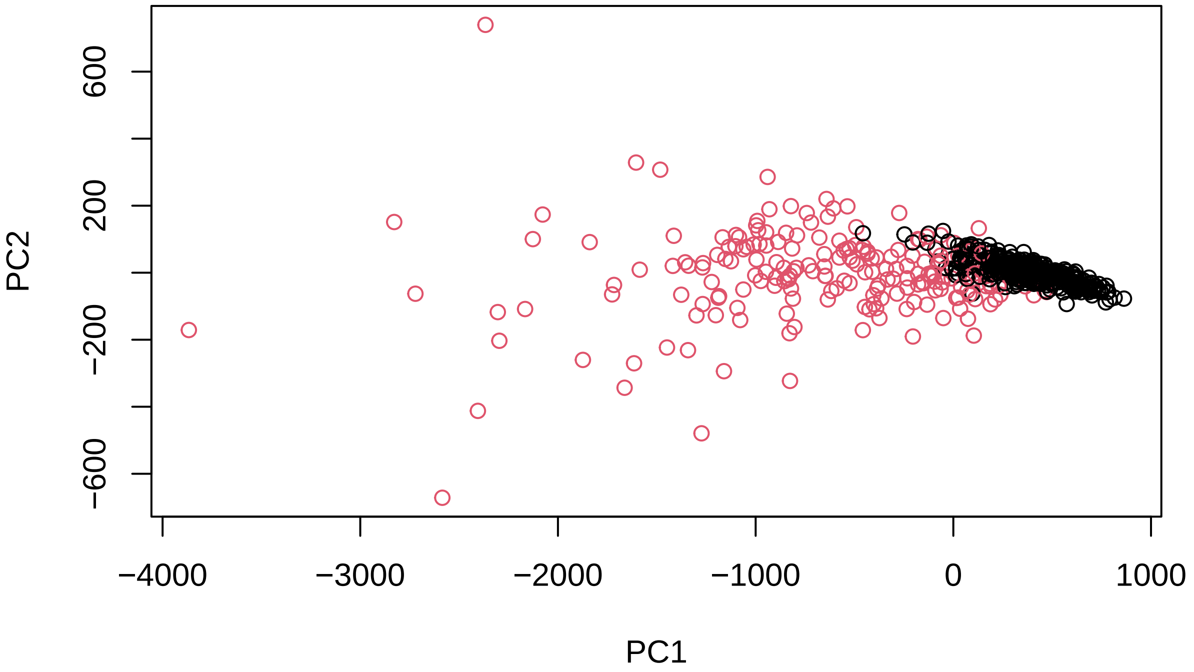
```
table(grps, diagnosis)
```

```
##     diagnosis
## grps   B   M
##    1   0  81
##    2 357 131
```

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



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



Q16. How well do the k-means and 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(grps, diagnosis)
```

```
##     diagnosis
## grps   B   M
##    1   0  81
##    2 357 131
```

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

```
##                     diagnosis
## wisc.hclust.clusters   B   M
##                    1 357 210
##                    2   0   2
```

Sensitivity/ Specificity

Accuracy What proportion did we get correct if we call cluster 1 M and cluster 2 B

```
(333+ 179)/nrow(wisc.data)
```

```
## [1] 0.8998243
```

Sensitivity

```
179/(179+33)
```

```
## [1] 0.8443396
```

Specificity

```
333/(333+24)
```

```
## [1] 0.9327731
```

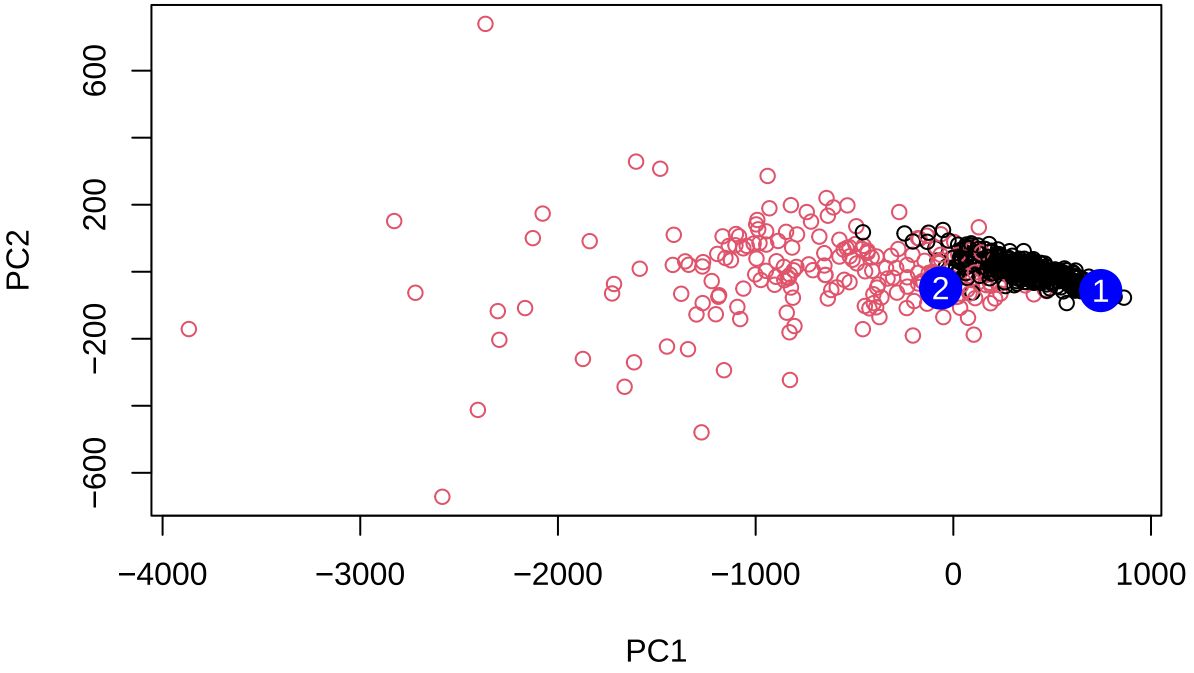
> Q17. Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

7. Prediction

```
#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,] 745.60081 -56.16454 -21.15609 -3.330663  9.355518  2.317462 -1.147268
## [2,] -64.40839 -48.46996 -15.93413 12.089591 -4.636008 -1.045210 -0.295228
##             PC8         PC9        PC10      PC11        PC12         PC13
## [1,] -0.7644759  0.11704582  0.06401851 0.1191717 -0.05611973 -0.040020096
## [2,] -0.7454142 -0.09167106 -0.76173550 0.3206674  0.02602751  0.005023528
##             PC14         PC15        PC16        PC17        PC18         PC19
## [1,]  0.01354667 -0.018755904 -0.01050870 -0.01183961 0.020946097  0.030567858
## [2,] -0.11943490  0.008958015  0.03391077 -0.02468455 0.008002482 -0.006896744
##              PC20         PC21        PC22         PC23         PC24
## [1,] -0.007960122 -0.003773165 0.018561168 0.0001875602 -0.005463212
## [2,]  0.007001178 -0.022182056 0.008725155 0.0075849336  0.004619616
##              PC25        PC26         PC27         PC28         PC29
## [1,] -0.005992320 0.005357732 4.550233e-05  0.003252776 0.0012510265
## [2,]  0.002804663 0.003229335 1.977351e-03 -0.002261832 0.0009130702
##               PC30
## [1,] -0.0009794321
## [2,] -0.0009078383
```

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



Q18. Which of these new patients should we prioritize for follow up based on your results?

We should prioritize patients 2 because the red cluster signifies malignant.