Mini Project

Max Strul

10/21/22

Table of contents

Mini Project

Unsupervised learning analysis of breast cancer cells

The data itself comes from the Wisconsin Breast Cancer Diagnostic Data Set first reported by K. P. Benne and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets".

```
wisc.df <- read.csv("WisconsinCancer.csv", 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  X
```

```
842302                    0.11890 NA
842517                    0.08902 NA
84300903                  0.08758 NA
84348301                  0.17300 NA
84358402                  0.07678 NA
843786                    0.12440 NA
```

```
wisc.data <- wisc.df[,-1]
#no more diagnosis column
wisc.data <- wisc.data[,-31]
#got rid of white space
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
```

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

```
[1] 569
```

```
ncol(wisc.data)
```

```
[1] 30
```

We can use diagnosis as a factor to sum over specific factors

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

How many individuals have a diagnosed cancer?

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

```
  B   M
357 212
```

How many variables have a suffix of "mean"

The `grep()` function will be used here

```
#obtain column names:
col.names.vector <- colnames(wisc.data)
#this is a vector
length(grep("_mean",col.names.vector))
```

```
[1] 10
```

Starting the principal component analysis

Lets try PCA on this data to see what major features might be hidden in this large dimensional dataset.

Functions we want to use: `prcomp()`

`hclust(dist(x))`

`cutree(x,k=?)`

Initial data analysis **scaling**

```
round(colMeans(wisc.data), 2)
```

```
            radius_mean            texture_mean          perimeter_mean
                  14.13                   19.29                   91.97
              area_mean         smoothness_mean        compactness_mean
                 654.89                    0.10                    0.10
         concavity_mean     concave.points_mean           symmetry_mean
                   0.09                    0.05                    0.18
 fractal_dimension_mean               radius_se              texture_se
                   0.06                    0.41                    1.22
           perimeter_se                 area_se           smoothness_se
                   2.87                   40.34                    0.01
```

```
       compactness_se            concavity_se       concave.points_se
                 0.03                    0.03                    0.01
         symmetry_se    fractal_dimension_se            radius_worst
                 0.02                    0.00                   16.27
       texture_worst         perimeter_worst              area_worst
                25.68                  107.26                  880.58
    smoothness_worst       compactness_worst         concavity_worst
                 0.13                    0.25                    0.27
 concave.points_worst          symmetry_worst fractal_dimension_worst
                 0.11                    0.29                    0.08
```

```
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
```

```
scale_false <- prcomp(wisc.data, scale = FALSE)
scale_true <- prcomp(wisc.data, scale = TRUE)
summary(scale_false)
```

```
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
```

```r
summary(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
```

```
#We will continue with scale = true

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

An example of when you don't want to re-scale is when you have all of the same units data (example a .pdb structure file, all the data is the same between proteins, and if you re-scale then you are altering the data!)

We see here that if we are un-scaled then PC1 covers 98 percent of the variance, but when we scale its only 44%

```
#we can perform a score plot
#aka a P.C. plot or
#ordination plots

head(wisc.pr$x)
```

```
               PC1        PC2        PC3       PC4        PC5         PC6
842302   -9.184755  -1.946870 -1.1221788 3.6305364  1.1940595  1.41018364
842517   -2.385703   3.764859 -0.5288274 1.1172808 -0.6212284  0.02863116
84300903 -5.728855   1.074229 -0.5512625 0.9112808  0.1769302  0.54097615
84348301 -7.116691 -10.266556 -3.2299475 0.1524129  2.9582754  3.05073750
84358402 -3.931842   1.946359  1.3885450 2.9380542 -0.5462667 -1.22541641
843786   -2.378155  -3.946456 -2.9322967 0.9402096  1.0551135 -0.45064213
                PC7         PC8         PC9        PC10       PC11        PC12
842302    2.15747152  0.39805698 -0.15698023 -0.8766305 -0.2627243 -0.8582593
842517    0.01334635 -0.24077660 -0.71127897  1.1060218 -0.8124048  0.1577838
84300903 -0.66757908 -0.09728813  0.02404449  0.4538760  0.6050715  0.1242777
84348301  1.42865363 -1.05863376 -1.40420412 -1.1159933  1.1505012  1.0104267
84358402 -0.93538950 -0.63581661 -0.26357355  0.3773724 -0.6507870 -0.1104183
843786    0.49001396  0.16529843 -0.13335576 -0.5299649 -0.1096698  0.0813699
               PC13         PC14         PC15        PC16        PC17
842302    0.10329677 -0.690196797  0.601264078  0.74446075 -0.26523740
842517   -0.94269981 -0.652900844 -0.008966977 -0.64823831 -0.01719707
84300903 -0.41026561  0.016665095 -0.482994760  0.32482472  0.19075064
84348301 -0.93245070 -0.486988399  0.168699395  0.05132509  0.48220960
84358402  0.38760691 -0.538706543 -0.310046684 -0.15247165  0.13302526
843786   -0.02625135  0.003133944 -0.178447576 -0.01270566  0.19671335
               PC18        PC19        PC20         PC21        PC22
```

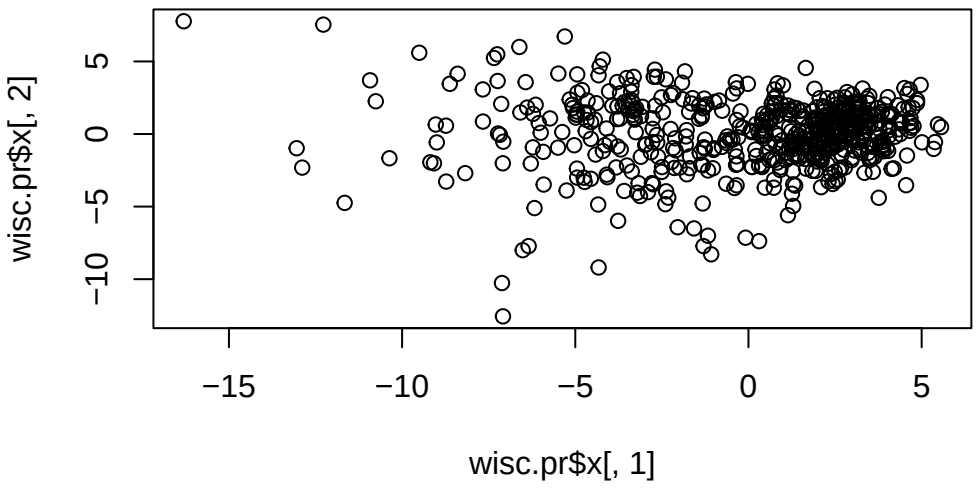
```
842302   -0.54907956  0.1336499  0.34526111  0.096430045 -0.06878939
842517    0.31801756 -0.2473470 -0.11403274 -0.077259494  0.09449530
84300903 -0.08789759 -0.3922812 -0.20435242  0.310793246  0.06025601
84348301 -0.03584323 -0.0267241 -0.46432511  0.433811661  0.20308706
84358402 -0.01869779  0.4610302  0.06543782 -0.116442469  0.01763433
843786   -0.29727706 -0.1297265 -0.07117453 -0.002400178  0.10108043
               PC23         PC24         PC25         PC26        PC27
842302    0.08444429  0.175102213  0.150887294 -0.201326305 -0.25236294
842517   -0.21752666 -0.011280193  0.170360355 -0.041092627  0.18111081
84300903 -0.07422581 -0.102671419 -0.171007656  0.004731249  0.04952586
84348301 -0.12399554 -0.153294780 -0.077427574 -0.274982822  0.18330078
84358402  0.13933105  0.005327110 -0.003059371  0.039219780  0.03213957
843786    0.03344819 -0.002837749 -0.122282765 -0.030272333 -0.08438081
                PC28         PC29          PC30
842302   -0.0338846387  0.045607590  0.0471277407
842517    0.0325955021 -0.005682424  0.0018662342
84300903  0.0469844833  0.003143131 -0.0007498749
84348301  0.0424469831 -0.069233868  0.0199198881
84358402 -0.0347556386  0.005033481 -0.0211951203
843786    0.0007296587 -0.019703996 -0.0034564331
```

```
head(wisc.pr$rotation)
```

```
                        PC1         PC2          PC3         PC4         PC5
radius_mean      -0.2189024  0.23385713 -0.008531243  0.04140896 -0.03778635
texture_mean     -0.1037246  0.05970609  0.064549903 -0.60305000  0.04946885
perimeter_mean   -0.2275373  0.21518136 -0.009314220  0.04198310 -0.03737466
area_mean        -0.2209950  0.23107671  0.028699526  0.05343380 -0.01033125
smoothness_mean  -0.1425897 -0.18611302 -0.104291904  0.15938277  0.36508853
compactness_mean -0.2392854 -0.15189161 -0.074091571  0.03179458 -0.01170397
                         PC6         PC7          PC8          PC9        PC10
radius_mean       0.018740790 -0.12408834  0.007452296 -0.223109764  0.09548644
texture_mean     -0.032178837  0.01139954 -0.130674825  0.112699390  0.24093407
perimeter_mean    0.017308445 -0.11447706  0.018687258 -0.223739213  0.08638562
area_mean        -0.001887748 -0.05165343 -0.034673604 -0.195586014  0.07495649
smoothness_mean  -0.286374497 -0.14066899  0.288974575  0.006424722 -0.06929268
compactness_mean -0.014130949  0.03091850  0.151396350 -0.167841425  0.01293620
                        PC11        PC12       PC13         PC14        PC15
radius_mean      -0.04147149  0.05106746 0.01196721  0.059506135 -0.05111877
texture_mean      0.30224340  0.25489642 0.20346133 -0.021560100 -0.10792242
perimeter_mean   -0.01678264  0.03892611 0.04410950  0.048513812 -0.03990294
```

```
area_mean         -0.11016964  0.06543751 0.06737574  0.010830829  0.01396691
smoothness_mean    0.13702184  0.31672721 0.04557360  0.445064860 -0.11814336
compactness_mean   0.30800963 -0.10401704 0.22928130  0.008101057  0.23089996
                        PC16        PC17         PC18        PC19        PC20
radius_mean       -0.1505839  0.20292425  0.146712338  0.22538466 -0.04969866
texture_mean      -0.1578420 -0.03870612 -0.041102985  0.02978864 -0.24413499
perimeter_mean    -0.1144540  0.19482131  0.158317455  0.23959528 -0.01766501
area_mean         -0.1324480  0.25570576  0.266168105 -0.02732219 -0.09014376
smoothness_mean   -0.2046132  0.16792991 -0.352226802 -0.16456584  0.01710096
compactness_mean   0.1701784 -0.02030771  0.007794138  0.28422236  0.48868633
                       PC21        PC22          PC23        PC24        PC25
radius_mean       -0.06857001 -0.07292890 -0.0985526942 -0.18257944 -0.01922650
texture_mean       0.44836947 -0.09480063 -0.0005549975  0.09878679  0.08474593
perimeter_mean    -0.06976904 -0.07516048 -0.0402447050 -0.11664888  0.02701541
area_mean         -0.01844328 -0.09756578  0.0077772734  0.06984834 -0.21004078
smoothness_mean   -0.11949175 -0.06382295 -0.0206657211  0.06869742  0.02895489
compactness_mean   0.19262140  0.09807756  0.0523603957 -0.10413552  0.39662323
                       PC26        PC27          PC28         PC29
radius_mean       -0.12947640 -0.13152667  2.111940e-01  0.211460455
texture_mean      -0.02455666 -0.01735731 -6.581146e-05 -0.010533934
perimeter_mean    -0.12525595 -0.11541542  8.433827e-02  0.383826098
area_mean          0.36272740  0.46661248 -2.725083e-01 -0.422794920
smoothness_mean   -0.03700369  0.06968992  1.479269e-03 -0.003434667
compactness_mean   0.26280847  0.09774871 -5.462767e-03 -0.041016774
                        PC30
radius_mean        0.702414091
texture_mean       0.000273661
perimeter_mean    -0.689896968
area_mean         -0.032947348
smoothness_mean   -0.004847458
compactness_mean   0.044674186
```

```
plot(wisc.pr$x[,1],wisc.pr$x[,2])
```



** each dot here is a patient! **

We can now color based on diagnosis

```
plot(wisc.pr$x[,1],wisc.pr$x[,2],col=diagnosis,title="Malginant (red) vs Non Malignant (Bl
```

```
Warning in plot.window(...): "title" is not a graphical parameter
```

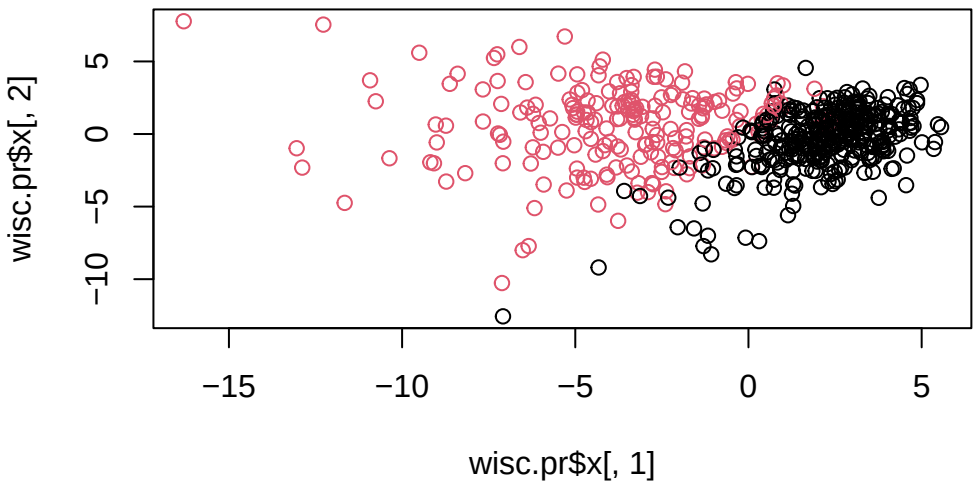
```
Warning in plot.xy(xy, type, ...): "title" is not a graphical parameter
```

```
Warning in axis(side = side, at = at, labels = labels, ...): "title" is not a
graphical parameter
```

```
Warning in axis(side = side, at = at, labels = labels, ...): "title" is not a
graphical parameter
```

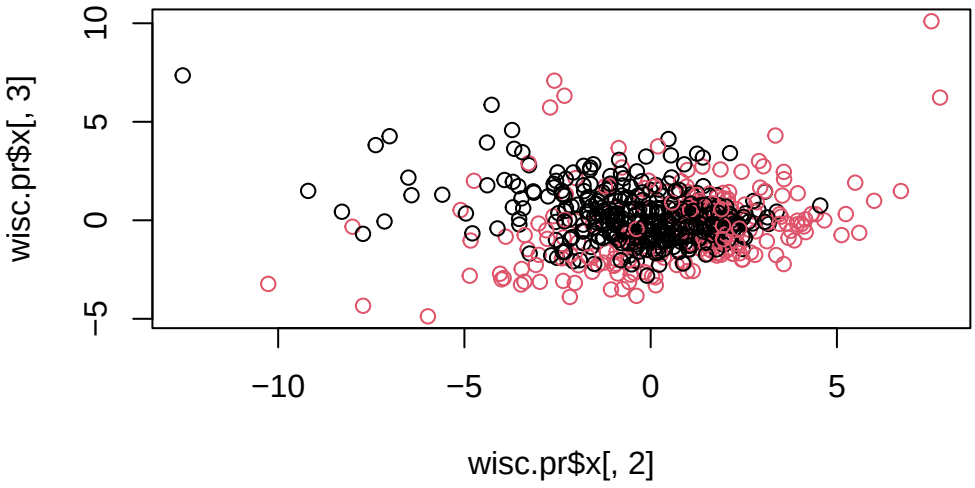
```
Warning in box(...): "title" is not a graphical parameter
```

```
Warning in title(...): "title" is not a graphical parameter
```

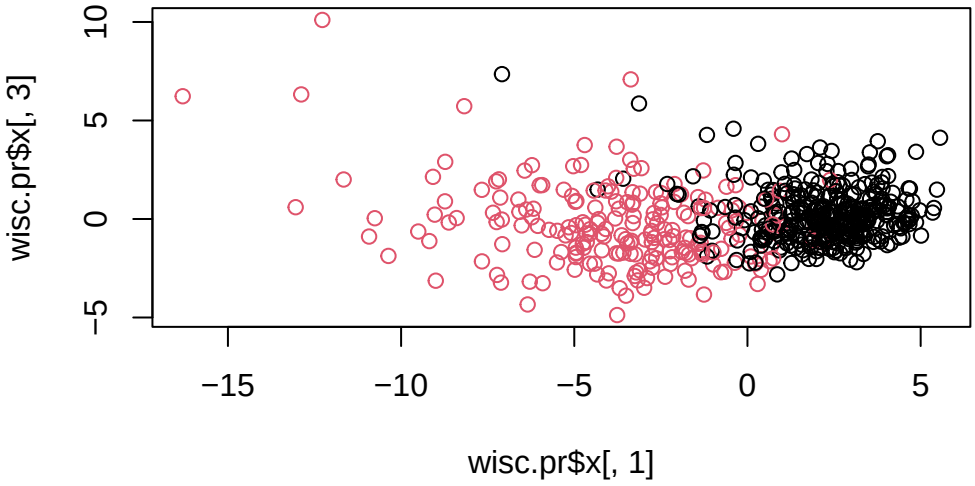


Can we plot other PCAs to see if we see such a stark contrast?

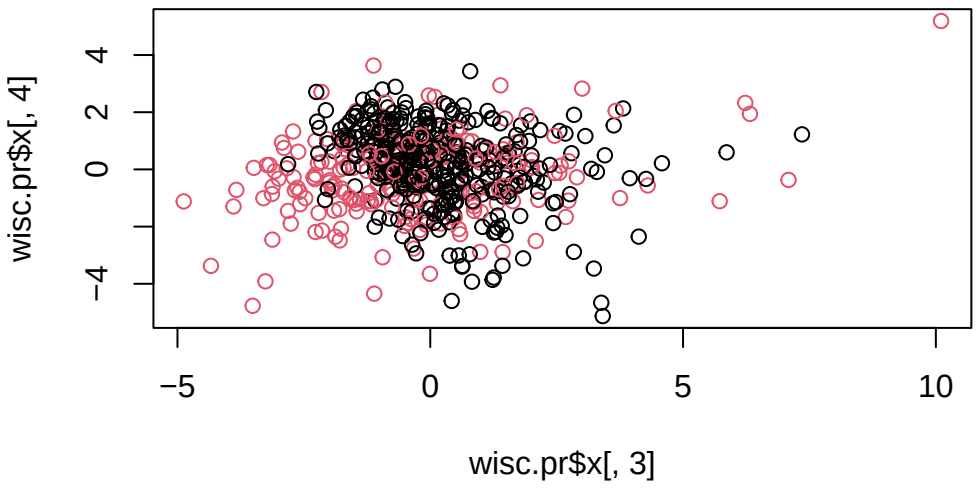
```
plot(wisc.pr$x[,2],wisc.pr$x[,3],col=diagnosis)
```




```
plot(wisc.pr$x[,1],wisc.pr$x[,3],col=diagnosis)
```



```
plot(wisc.pr$x[,3],wisc.pr$x[,4],col=diagnosis)
```

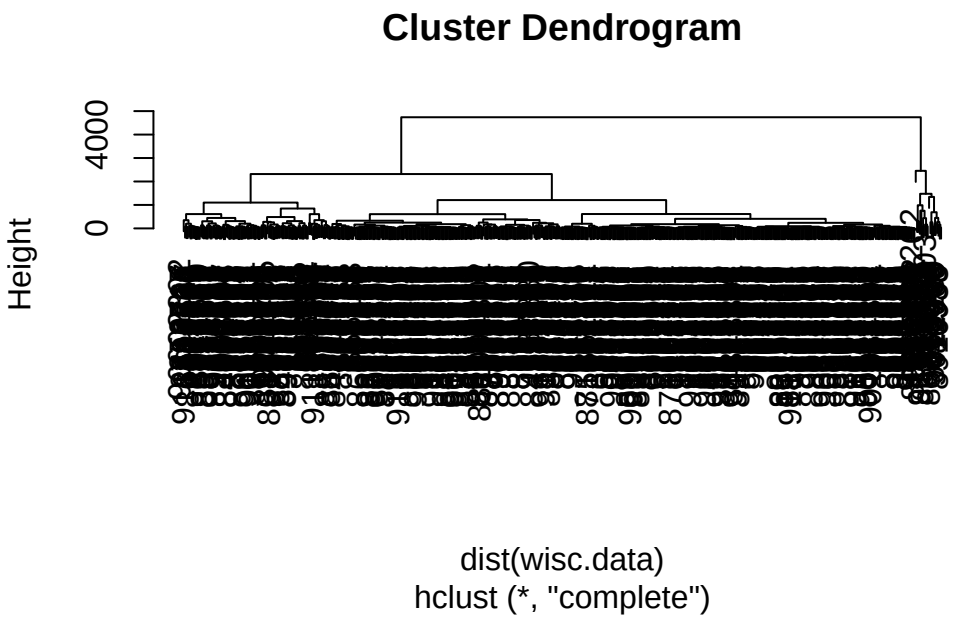


Comparing PCA1 and PCA2 are the best. 1 and 3 also work.

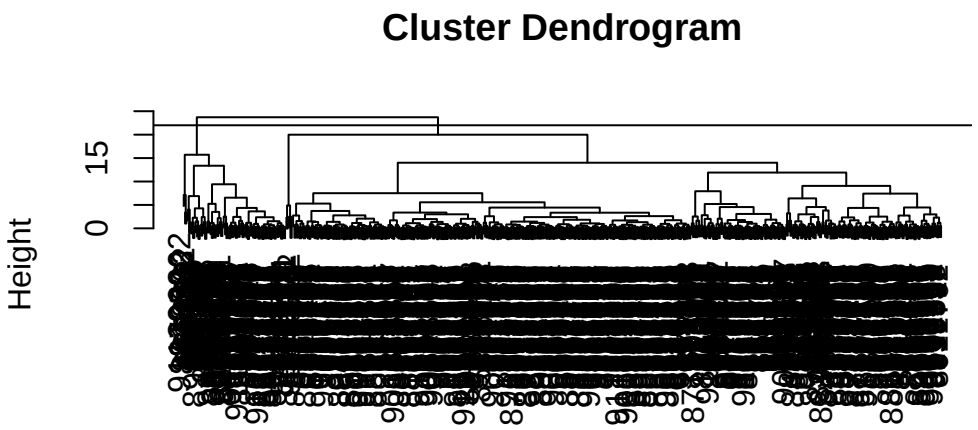
So now we want to eventually use these graphs and auto generate this so that we can find out which gives us the greatest distance between the two groups

we can use the distance function to do so?

```
wisc.pr.pca.1.2 <- cbind(x=wisc.pr$x[,1],y=wisc.pr$x[,2])

wisc.pr.tree <- hclust(dist(wisc.pr.pca.1.2))

wisc.pr.tree.original <- hclust(dist(wisc.data))

plot(wisc.pr.tree.original)
```



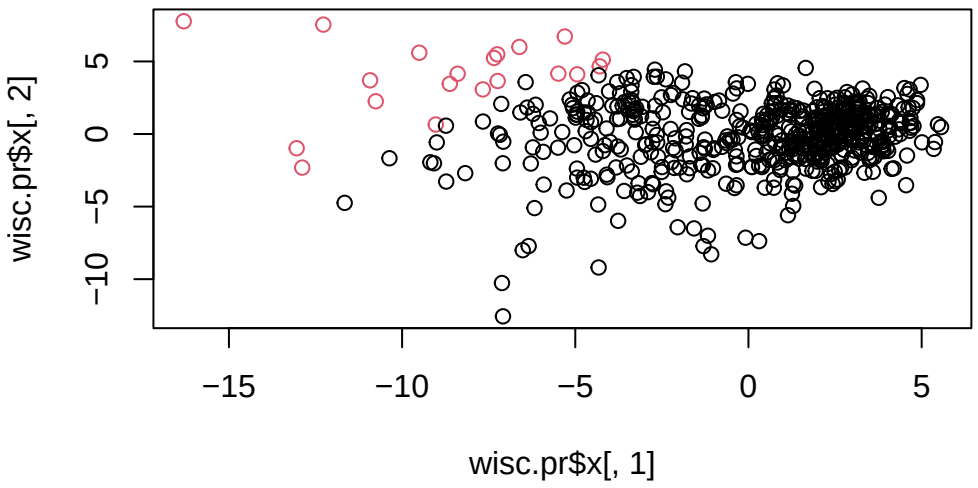

```
plot(wisc.pr.tree)

#it looks like height of 22 looks like a good point to cut
abline(h=22)
```




```
tree_groups <- cutree(wisc.pr.tree.original,k=2)

#Here we see that of the two groups,
#the group separation was very poor.

table(diagnosis, tree_groups)
```

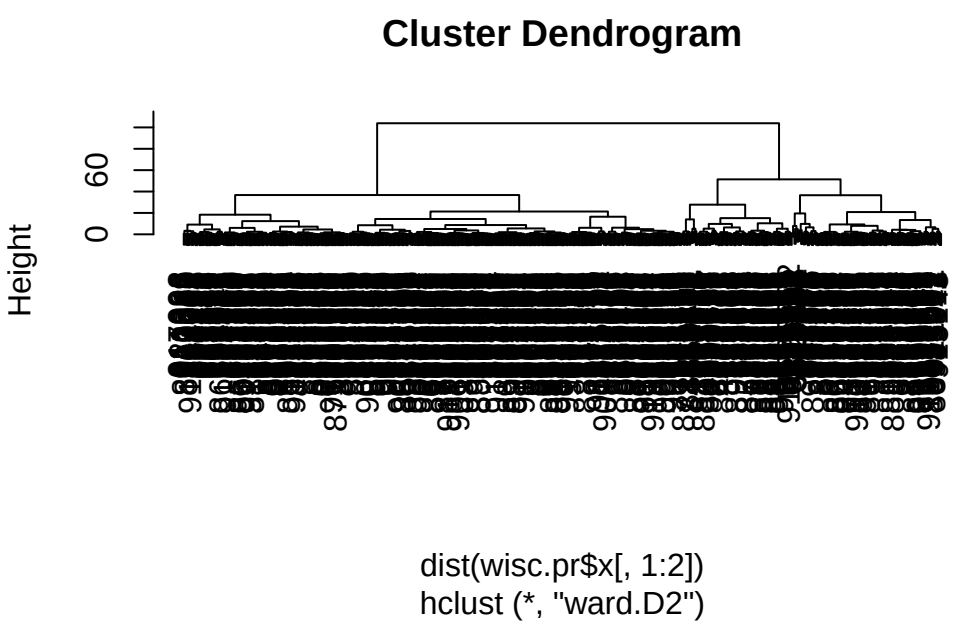
```
         tree_groups
diagnosis   1   2
        B 357   0
        M 192  20
```

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




```
#k=2 here becauase we want two groups
#km.pr <- kmeans(wisc.pr.pca.1.2, centers = 2, nstart = 20)

#plot(wisc.pr.pca.1.2, col=km.pr$cluster)
```

```
newtree.pca <- hclust(dist(wisc.pr$x[,1:2]),method="ward.D2")
plot(newtree.pca)
```




```
table(diagnosis, cutree(newtree.pca,k=2))
```

```
diagnosis   1   2
        B  18 339
        M 177  35
```

Here we see that the three PCAs allow us to develop a table much better!

Infact the sensitivtiy of this algorithm might be good, lets find out!

```
summary(diagnosis)
```

```
  B   M
357 212
```

```
tn <- 339
tp <- 177
fn <- 18
fp <- 35
sensitivity <- tp/(tp+fn)
```

```r
specifcity <- tn/(fp+tn)
#now we want to find out the values:

sensitivity
```

```
[1] 0.9076923
```

```r
specifcity
```

```
[1] 0.9064171
```

You can explore through utilizing more of the PCAs, however the sensitivty and specificty get worse! For example:

sensitivity and specificity for using [,1:3]:

sens: 0.88

spec: 0.91

sensitivity and specificity for using [,1:2]:

sens: 0.91

spec: 0.91

So actually, using just the first two PCAs provides a better model!

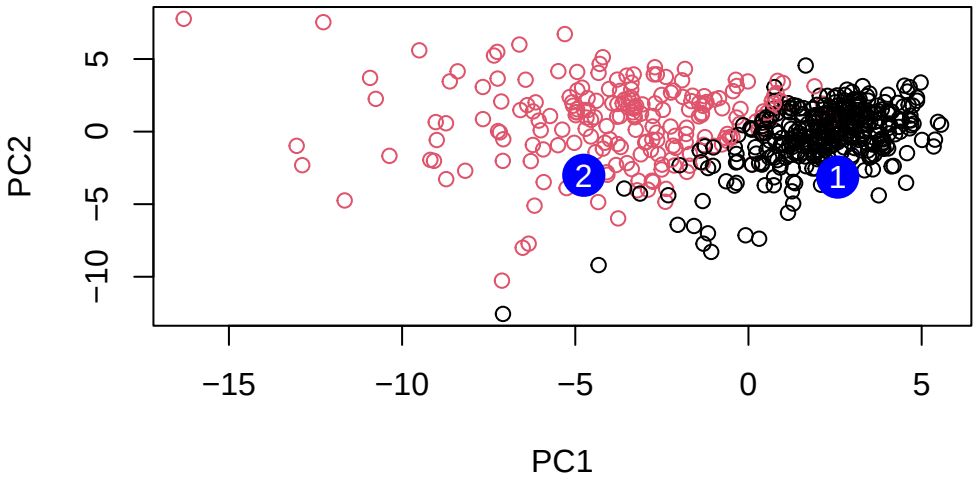
We can now use this as a predictive method!

`predict()` will take our PCA model from before and you can add new data to project onto our PCA space

```r
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
```

```r
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")
```



I would follow up with patient 2, since the red group is the malignant group and patient 1 is closer to the non malignant group.

```r
sessionInfo()
```

```
R version 4.2.1 (2022-06-23)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Monterey 12.5.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.2.1  magrittr_2.0.3  fastmap_1.1.0   cli_3.4.1
 [5] tools_4.2.1     htmltools_0.5.3 yaml_2.3.5      stringi_1.7.8
 [9] rmarkdown_2.16  knitr_1.40      stringr_1.4.1   xfun_0.33
[13] digest_0.6.29   jsonlite_1.8.2  rlang_1.0.6     evaluate_0.17
```

Remember our goal is to: 1.) Reduce dimensionality 2.) Visualize multidimensional data 3.) Choose the most useful variables (features or PCA0s 4.) Identify groupings of objects 5.) Identify outliars and remove if necessary