class08

Jihyun In (PID: A16955363)

Table of contents

Background

This source provides materials for a class mini-project focused on unsupervised learning analysis of human breast cancer cell data. Students will conduct principal component analysis (PCA) for dimensionality reduction and then apply hierarchical and k-means clustering techniques. The project involves exploratory data analysis, interpreting PCA results, evaluating clustering performance by comparing cluster assignments to actual diagnoses, and optionally combining PCA with clustering. The goal is to identify potential groupings within the cell data based on their characteristics without prior knowledge of malignancy, and the project concludes with an application of the PCA model to classify new patient samples.

Data import

Our data come from the U of Wisconsin Medical Center:

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

Q. How many patients/samples are in this dataset?

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

```
[1] 569
```

Q2. How many ofthe ovservations have a malignant diagnosis?

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

```
  B   M
357 212
```

```
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"
```

```
length(grep("_mean", colnames(wisc.df), value = T))
```

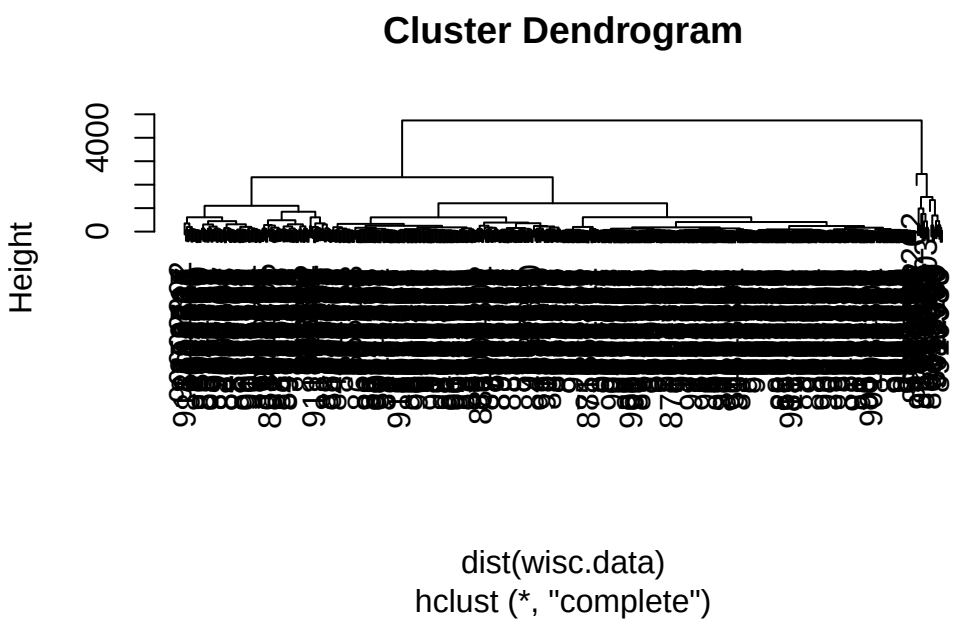
```
[1] 10
```

There is a diagnosis column that is the clinician consensusthat I want to exclude form any further analysis. We will come back later and compare our results to this diagnosis.

```
diagnosis <- as.factor(wisc.df$diagnosis)
wisc.data <- wisc.df[,-1]
```

Clustering

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

plot(hc)
```



We can extract clusters form this rather poor dendrogram/tree with the `cutree()` command

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

How many individuals in each cluster?

```
table(grps)
```

```
grps
  1   2
549  20
```

```
table(diagnosis)
```

```
diagnosis
  B   M
357 212
```

We can generate a cross-table that compares our cluster `grps` vector with our `diagnosis` vector values.

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

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

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
```

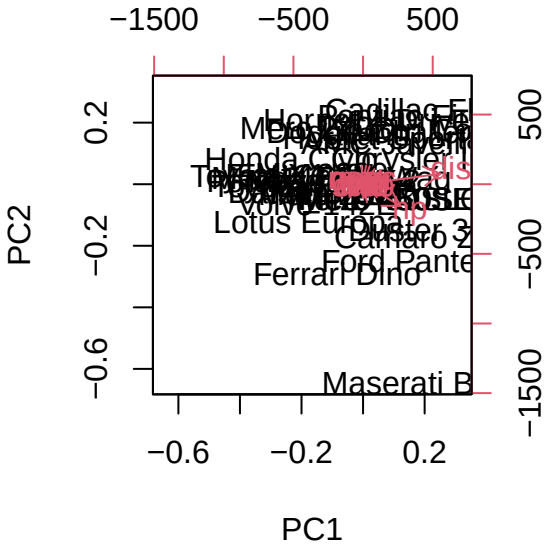
The main function fo rPCA in base R is `prcomp()`. It has a default input parameter of `scale=FALSE`.

```
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

We could do a PCA of this data as is, and it would be misleading…

```
pc <- prcomp(mtcars)
biplot(pc)
```



Let's look at the mean values of each column nd their standard deviations.

```
colMeans(mtcars)
```

```
       mpg        cyl       disp         hp       drat         wt       qsec
 20.090625   6.187500 230.721875 146.687500   3.596563   3.217250  17.848750
        vs         am       gear       carb
  0.437500   0.406250   3.687500   2.812500
```

```
apply(mtcars, 2, sd)
```

```
       mpg        cyl       disp         hp       drat         wt
 6.0269481  1.7859216 123.9386938 68.5628685  0.5346787  0.9784574
      qsec         vs         am       gear       carb
 1.7869432  0.5040161  0.4989909  0.7378041  1.6152000
```

we can "scale" tis data before PCA to get a much better representation an danalysis of all the columns.

```
mtscale <- scale(mtcars)
```

```
round(colMeans(mtscale))
```

```
 mpg  cyl disp   hp drat   wt qsec   vs   am gear carb
   0    0    0    0    0    0    0    0    0    0    0
```

```
apply(mtscale, 2, sd)
```

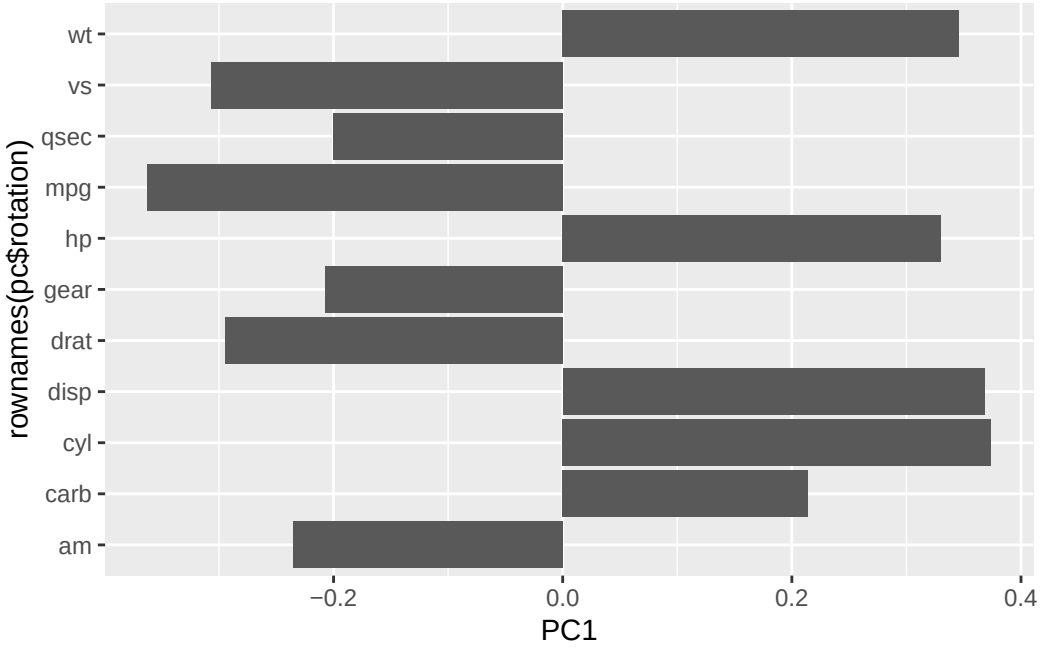
```
 mpg  cyl disp   hp drat   wt qsec   vs   am gear carb
   1    1    1    1    1    1    1    1    1    1    1
```

```
pc.scale <- prcomp(mtscale)
```

We can look at the two main results figures from PCA - the "PC plot" (a.k.a. score plot, orientation plot, or PC1 vs PC2 plot). The "loadings plot" show how the original variables contribute to the new PCs.

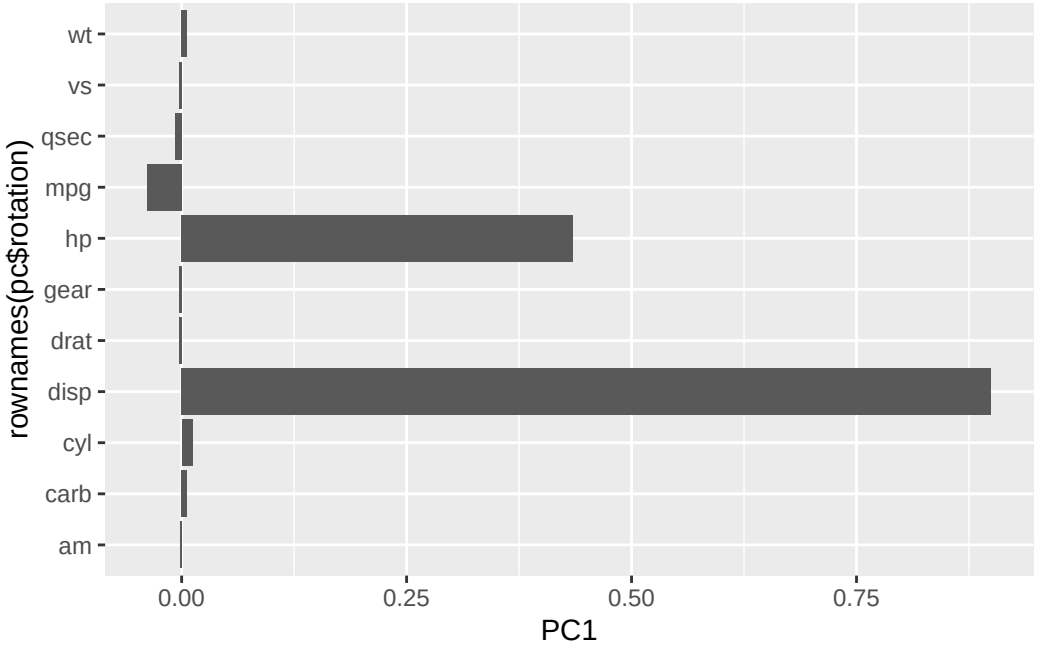
A loadings plot of the unscaled PCA results

```
library(ggplot2)

ggplot(pc.scale$rotation) +
  aes(PC1, rownames(pc$rotation)) +
  geom_col()
```



Scaled one:

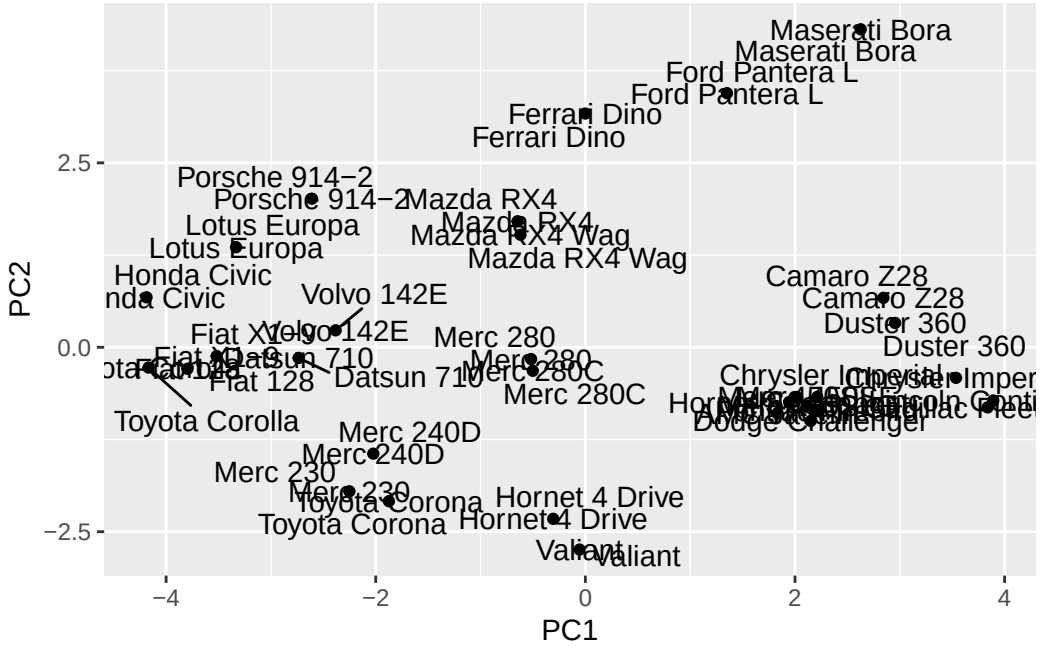
```
library(ggplot2)

ggplot(pc$rotation) +
  aes(PC1, rownames(pc$rotation)) +
  geom_col()
```



PC plot of scaled PCA results

```
library(ggrepel)
ggplot(pc.scale$x) +
  aes(PC1, PC2, label=rownames(pc.scale$x)) +
  geom_point() +
  geom_text() + geom_text_repel()
```

```
Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Key point: In general we will set `scale=TRUE` when we do PCA. This is not the default but probably should be...

PCA of wisc.data

Now with our actual data: We can check the sd and mean of the different columns in `wisc.data` to see if we need to scale - hint: we do!)

```
colMeans(wisc.data)
```

```
            radius_mean            texture_mean          perimeter_mean
           1.412729e+01            1.928965e+01            9.196903e+01
              area_mean         smoothness_mean        compactness_mean
           6.548891e+02            9.636028e-02            1.043410e-01
         concavity_mean      concave.points_mean           symmetry_mean
           8.879932e-02            4.891915e-02            1.811619e-01
 fractal_dimension_mean               radius_se              texture_se
           6.279761e-02            4.051721e-01            1.216853e+00
           perimeter_se                 area_se           smoothness_se
           2.866059e+00            4.033708e+01            7.040979e-03
         compactness_se          concavity_se     concave.points_se
           2.547814e-02            3.189372e-02            1.179614e-02
            symmetry_se    fractal_dimension_se            radius_worst
```

```
          2.054230e-02            3.794904e-03            1.626919e+01
         texture_worst         perimeter_worst              area_worst
          2.567722e+01            1.072612e+02            8.805831e+02
       smoothness_worst       compactness_worst         concavity_worst
          1.323686e-01            2.542650e-01            2.721885e-01
  concave.points_worst          symmetry_worst fractal_dimension_worst
          1.146062e-01            2.900756e-01            8.394582e-02
```

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

```
           radius_mean            texture_mean          perimeter_mean
          3.524049e+00            4.301036e+00            2.429898e+01
             area_mean         smoothness_mean        compactness_mean
          3.519141e+02            1.406413e-02            5.281276e-02
        concavity_mean     concave.points_mean           symmetry_mean
          7.971981e-02            3.880284e-02            2.741428e-02
 fractal_dimension_mean               radius_se              texture_se
          7.060363e-03            2.773127e-01            5.516484e-01
          perimeter_se                 area_se           smoothness_se
          2.021855e+00            4.549101e+01            3.002518e-03
        compactness_se            concavity_se       concave.points_se
          1.790818e-02            3.018606e-02            6.170285e-03
           symmetry_se    fractal_dimension_se            radius_worst
          8.266372e-03            2.646071e-03            4.833242e+00
         texture_worst         perimeter_worst              area_worst
          6.146258e+00            3.360254e+01            5.693570e+02
       smoothness_worst       compactness_worst         concavity_worst
          2.283243e-02            1.573365e-01            2.086243e-01
  concave.points_worst          symmetry_worst fractal_dimension_worst
          6.573234e-02            6.186747e-02            1.806127e-02
```

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

To see how well PCA is doing here in terms of capturing the variance (or spread) in the data we can use the `summary()` function.

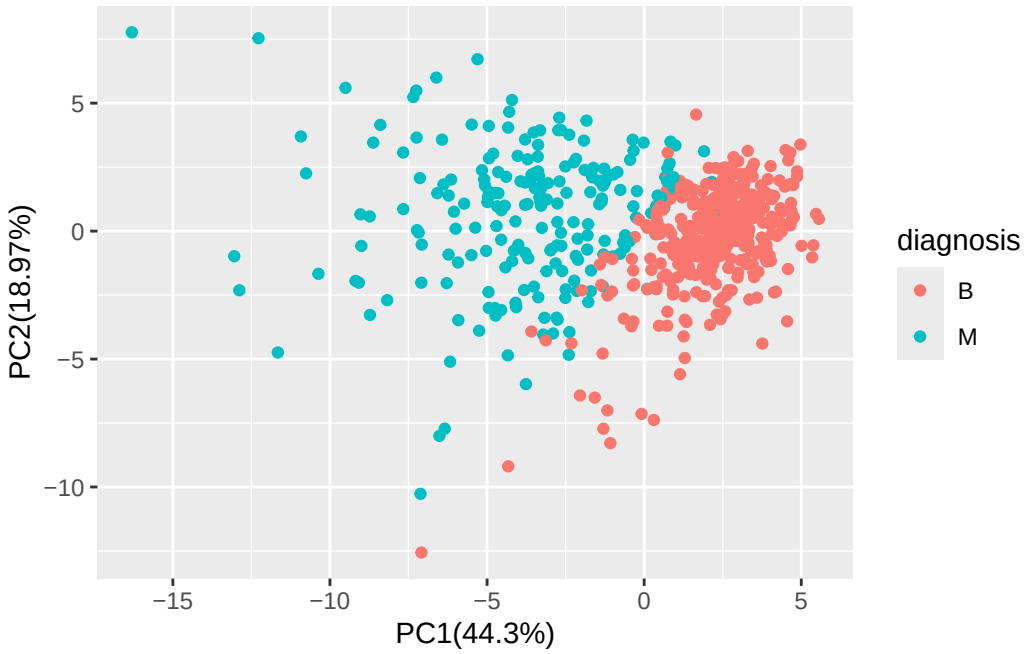
```
summary(wisc.pr)
```

```
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
```

Let's make the main PC1 vs. PC2

```
ggplot(wisc.pr$x) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point() +
  xlab("PC1(44.3%)") +
  ylab("PC2(18.97%)")
```



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

PC1 captures 44.27% of the variance.

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

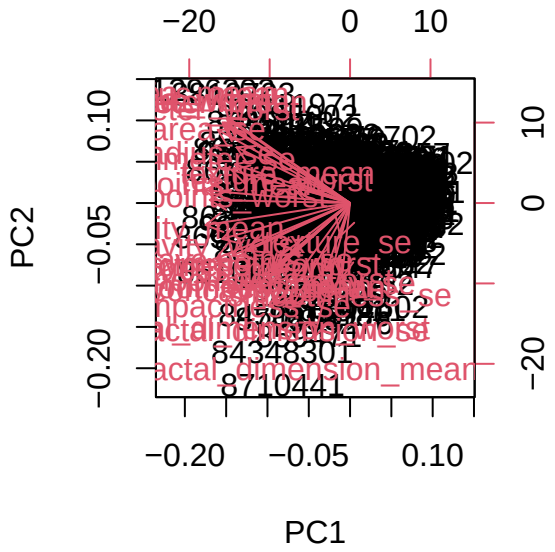
3 PCs are required to describe 70% of the original variance in the data.

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

7 PCs are required to describe 90% of the original variance in the data.

Creating the biplot:

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

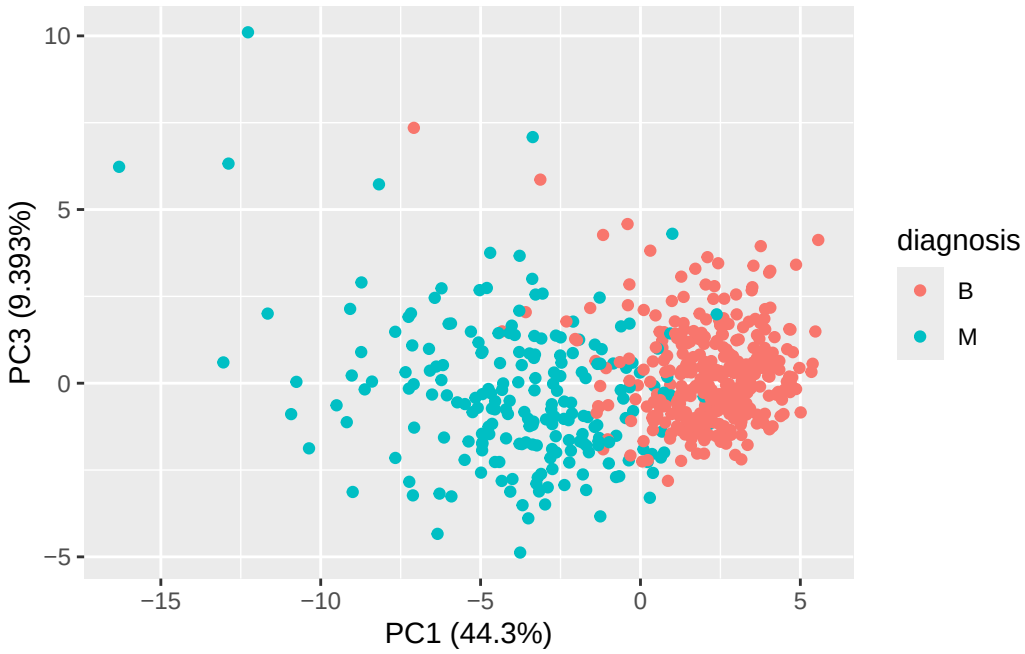


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

This is difficult to understand because there is a lot of data points. It is hardto gain any sorts of information from this plot.

Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

```
ggplot(wisc.pr$x) +
  aes(PC1, PC3, col=diagnosis) +
  geom_point() +
  xlab("PC1 (44.3%)") +
  ylab("PC3 (9.393%)")
```

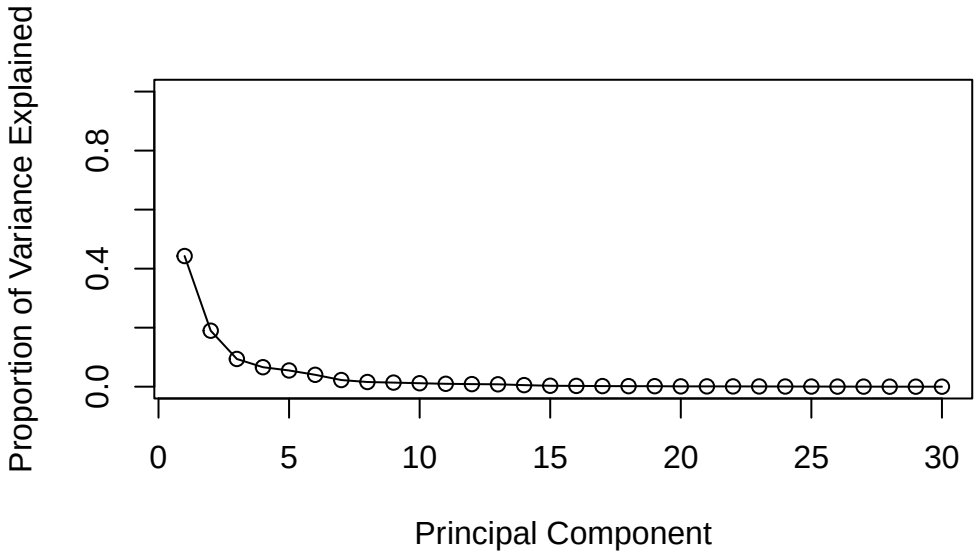


###Variance Explained

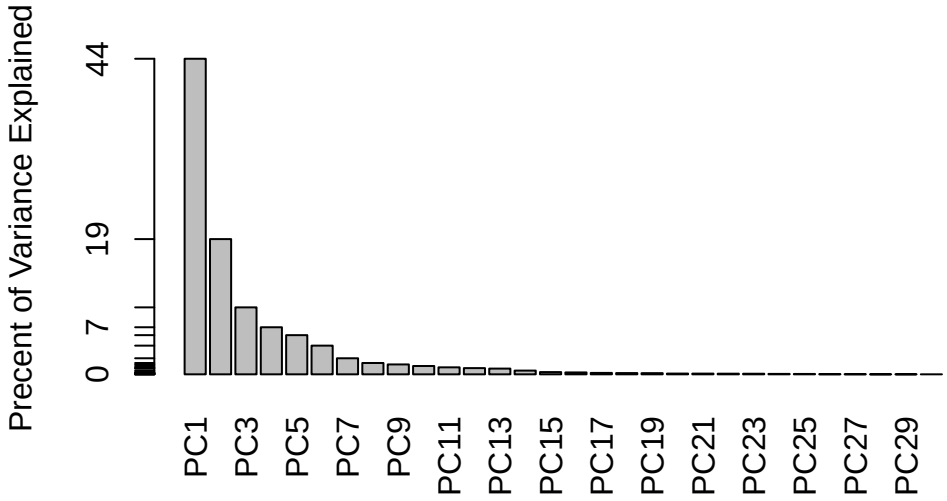
```
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

```
# Variance explained by each principal component: pve
pve <- pr.var / sum(pr.var)

# Plot variance explained for each principal component
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "o")
```




```
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```



Q9. For the first principal component, what is the component of the loading vector (i.e. wisc.pr$rotation[,1]) for the feature concave.points_mean?

```
wisc.pr$rotation[,1]
```

```
radius_mean             texture_mean           perimeter_mean
```

```
            -0.21890244             -0.10372458             -0.22753729
              area_mean         smoothness_mean        compactness_mean
            -0.22099499             -0.14258969             -0.23928535
         concavity_mean     concave.points_mean           symmetry_mean
            -0.25840048             -0.26085376             -0.13816696
 fractal_dimension_mean               radius_se              texture_se
            -0.06436335             -0.20597878             -0.01742803
           perimeter_se                 area_se           smoothness_se
            -0.21132592             -0.20286964             -0.01453145
         compactness_se            concavity_se       concave.points_se
            -0.17039345             -0.15358979             -0.18341740
            symmetry_se    fractal_dimension_se             radius_worst
            -0.04249842             -0.10256832             -0.22799663
          texture_worst         perimeter_worst              area_worst
            -0.10446933             -0.23663968             -0.22487053
       smoothness_worst       compactness_worst         concavity_worst
            -0.12795256             -0.21009588             -0.22876753
   concave.points_worst          symmetry_worst fractal_dimension_worst
            -0.25088597             -0.12290456             -0.13178394
```

The component is: -0.26085376

> Q10. What is the minimum number of principal components required to explain 80% of the variance of the data?

5 PCs are required to explain 80% of the data.

Combining Methods

We can take our PCA results and use them as a basis set for othe ranalysis such as clustering

Clustering on PCA results

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

```
                PC1           PC2
842302     -9.18475521  -1.946870030
842517     -2.38570263   3.764859063
84300903   -5.72885549   1.074228589
84348301   -7.11669126 -10.266555635
```

84358402	-3.93184247	1.946358977
843786	-2.37815462	-3.946456430
844359	-2.23691506	2.687666414
84458202	-2.14141428	-2.338186649
844981	-3.17213315	-3.388831138
84501001	-6.34616284	-7.720380945
845636	0.80970132	2.656937673
84610002	-2.64876984	-0.066509405
846226	-8.17783882	-2.698602007
846381	-0.34182514	0.967428026
84667401	-4.33856172	-4.856809832
84799002	-4.07207318	-2.974443983
848406	-0.22985277	1.563382114
84862001	-4.41412695	-1.417423150
849014	-4.94435304	4.110716528
8510426	1.23597583	0.188049490
8510653	1.57677384	-0.572304625
8510824	3.55420904	-1.661487969
8511133	-4.72904972	-3.302058265
851509	-4.20482441	5.123858059
852552	-4.94528075	1.542395143
852631	-7.09232236	-2.016835734
852763	-3.50717666	-2.169715994
852781	-3.06136021	1.874902631
852973	-4.00374127	-0.536769860
853201	-1.71380176	1.522365502
853401	-6.05411853	0.756511800
853612	-2.89968469	-4.001774373
85382601	-4.55077848	-0.337239419
854002	-4.98621538	1.131593223
854039	-2.98271631	-0.757756497
854253	-2.76393718	0.354044421
854268	-1.29505925	-0.912393466
854941	3.74601730	1.412230504
855133	0.99719148	3.348346731
855138	-0.76459136	-0.885464837
855167	2.14906252	1.922300194
855563	0.09324934	-2.258764532
855625	-9.08001023	-2.016898441
856106	-0.98958304	-0.984064148
85638502	0.29328849	0.136978564
857010	-5.37620991	0.134758404
85713702	4.57790860	-1.482915508

85715	-1.69851237	-2.350203862
857155	2.13456708	-0.095745364
857156	1.56610099	1.207370855
857343	3.53979091	1.281368110
857373	3.15503795	1.687473796
857374	3.44745515	0.497780720
857392	-3.29964761	1.129943574
857438	0.67402621	2.114549137
85759902	2.85564577	-0.152588874
857637	-4.64465207	2.308301508
857793	-2.17494916	-0.971261438
857810	3.71818738	1.786070097
858477	4.13232693	-2.401679261
858970	2.38373822	-2.755233814
858981	2.57661610	-3.135912650
858986	-4.75492832	-3.009032875
859196	2.31209785	-3.265116721
85922302	-1.69012080	-1.539322116
859283	-1.81071217	-0.722104815
859464	2.78347559	-2.308617372
859465	3.51555502	0.657730736
859471	-4.32619605	-9.194435552
859487	3.25841241	0.937013647
859575	-2.70221851	4.433240985
859711	0.30758513	-7.381317396
859717	-5.49886689	-0.937500513
859983	0.36139121	-0.119633821
8610175	2.62766457	0.696696349
8610404	-1.42691206	1.965372092
8610629	0.83378424	-1.963876860
8610637	-6.22541880	-0.919260690
8610862	-11.65845644	-4.744442593
8610908	2.01980045	0.254675746
861103	1.63694460	-1.714440587
8611161	-1.16643527	-2.512305286
8611555	-10.75977535	2.255997858
8611792	-5.03038488	-0.773728357
8612080	2.17255269	-0.496441169
8612399	-3.28534463	1.666770480
86135501	-0.60707378	-0.162071919
86135502	-3.58041324	2.204721916
861597	0.93333687	-0.926885919
861598	-1.25849730	-1.014684255

861648	1.58686770	1.618232954
861799	-0.25227559	0.530884425
861853	2.84492835	2.891103732
862009	1.96322268	0.964308498
862028	-2.77342525	-0.557509977
86208	-4.39236314	2.121640892
86211	2.58980407	-0.213446120
862261	3.90090576	-1.189020421
862485	2.81575365	-0.367560567
862548	-0.61573908	-0.638349988
862717	0.43247968	1.390820523
862722	4.55102442	-3.525683788
862965	3.44515217	1.423370040
862980	2.11410008	-1.847747782
862989	2.68898618	-1.418813334
863030	-3.21109134	-4.043198634
863031	0.74861266	-1.796058175
863270	3.15622889	1.034838878
86355	-13.04464395	-0.980650357
864018	2.22672053	-0.666823902
864033	2.49063954	-2.596401708
86408	0.10325957	-2.278139015
86409	-3.58813694	-3.922881433
864292	1.34776279	-3.553098748
864496	2.50791098	-3.248461200
864685	2.04423086	-0.304616836
864726	2.09522559	-3.663872062
864729	-3.10779316	-1.568009873
864877	-4.95236801	-2.382749650
865128	-0.85026612	2.304707468
865137	2.96339173	-0.371179883
86517	-3.33120209	1.324392029
865423	-12.88327621	-2.314585873
865432	0.77006610	0.064052860
865468	2.20057925	0.734958544
86561	3.14064881	1.875758557
866083	0.63831885	0.910564828
866203	-1.91744627	3.534971966
866458	-1.40762980	-1.303782215
866674	-4.63960895	1.480714297
866714	1.87581684	-1.421979452
8670	-1.43081046	1.048681220
86730502	-1.35143790	1.153126842

867387	0.70849143	1.566853657
867739	-2.16950708	2.823776101
868202	1.97510247	0.419018755
868223	2.59850310	0.481911436
868682	3.25514318	0.417953651
868826	-3.77870160	-0.859625218
868871	1.99028348	-1.328186508
868999	5.01059989	-0.574194748
869104	-1.34383800	1.273650278
869218	2.45380972	-0.897665441
869224	1.83566648	0.090946742
869254	4.34266862	0.892786708
869476	0.73216868	-3.698927716
869691	-2.39788795	-4.833735642
86973701	-0.39275308	-1.082115915
86973702	0.41196533	0.389288784
869931	3.04724369	2.235817239
871001501	1.44129554	-0.305601552
871001502	-0.08311297	-7.144073768
8710441	-7.08707084	-12.562140869
87106	3.74011364	-0.250281306
8711002	0.96832079	-0.944113582
8711003	2.41659356	-0.005547532
8711202	-4.09718264	0.378470806
8711216	0.75094226	3.067949187
871122	3.65143365	0.674055644
871149	4.67609710	1.102886823
8711561	0.59725624	-1.784081072
8711803	-3.38435289	2.908477953
871201	-6.14447970	2.015878994
8712064	1.32386648	-1.468002337
8712289	-5.48932277	4.162167037
8712291	2.99476813	2.736453261
87127	4.38287625	-0.006866240
8712729	-1.21256768	2.037246395
8712766	-5.06520424	1.783582969
8712853	1.97596280	1.841159394
87139402	2.51266554	-0.114151857
87163	0.94665494	1.683272444
87164	-2.81688978	-1.263990953
871641	3.48092345	-1.618268077
871642	4.65463399	0.222718846
872113	5.34691339	-1.025855214

872608	-1.16986842	-7.008319996
87281702	-2.95370292	-0.705800822
873357	4.97132782	3.383227983
873586	4.06045289	1.245070643
873592	-9.50430706	5.598558052
873593	-8.99924714	-0.580520281
873701	-0.75821134	1.607118481
873843	2.65702172	-0.539461132
873885	0.38967611	0.988372689
874158	3.88564657	-0.815354285
874217	-0.36455538	3.571318842
874373	2.83339783	0.398379162
874662	3.30737348	-0.155604280
874839	3.35435293	1.102762109
874858	-6.51738201	-8.004126824
875093	1.71622508	0.542688019
875099	5.56084294	0.477427493
875263	-1.77809695	-2.774146213
87556202	-2.60918908	-1.560049030
875878	2.81656023	0.969257722
875938	-2.49624245	-2.276479790
877159	-1.27590451	2.441111044
877486	-3.47014398	2.275846405
877500	-1.25557034	-0.382057094
877501	1.47213604	-0.116787496
877989	-1.64801062	2.100442902
878796	-9.02864522	0.654596849
87880	-4.55058603	-3.083925626
87930	0.78050321	-0.652275325
879523	0.22289648	0.701204504
879804	3.45185811	-1.305789765
879830	-0.44615183	2.785257008
8810158	-0.31416180	-2.075734301
8810436	2.05738166	2.470613908
881046502	-4.80474158	3.026440019
8810528	2.99607430	0.396429212
8810703	-12.27421974	7.536778599
881094802	-3.36923202	-2.585550416
8810955	-2.50656080	-2.612349885
8810987	-1.31690973	-2.152585453
8811523	0.38841453	-2.274796314
8811779	2.75450931	-1.085879226
8811842	-4.93923501	2.845820436

```
88119002   -4.29429880   4.662172958
8812816     2.37141521   0.732757814
8812818     0.96634003  -0.438052741
8812844     2.97318270  -1.809381711
8812877    -1.80166175  -0.166314618
8813129     2.47618073   1.417329346
88143502    0.82531310   1.249148354
88147101    3.89127342  -0.538097276
88147102    0.38923560   0.613875324
88147202    1.45604019   0.201549415
881861     -2.75314552  -3.462726769
881972     -3.25439115   0.125090426
88199202    4.01560693   1.353056834
88203002    3.67318128   1.290493562
88206102   -3.31340866   3.935689268
882488      4.03305870  -1.161524089
88249602    2.33218128   1.347793816
88299702   -8.39645911   4.150252991
883263     -2.64524110   3.947699392
883270      0.68116951   1.134992457
88330202   -3.34735048   2.152675899
88350402    2.22576208   1.213931555
883539      4.47364832   1.739598699
883852     -1.32145867  -4.785266054
88411702    1.84846341   1.582735509
884180     -3.79222888   1.025330459
884437      2.27931254  -2.075859442
884448      3.22593164   1.171043084
884626     -1.02499890  -2.359186109
88466802    2.58028546  -0.728572047
884689      2.48369783  -0.460191483
884948     -7.13756174   2.073018195
88518501    3.02981522   0.648897838
885429     -7.07878207  -0.527758800
8860702    -1.31294375   1.773865017
886226     -3.70838653   2.805008998
886452     -0.46007427  -0.393819769
88649001   -6.38703001   1.821491443
886776     -5.25092469  -3.891107226
887181     -8.72618338  -3.276993966
88725602   -3.68807276  -1.064709236
887549     -3.37528098   3.368264175
888264      0.83947101   3.497008034
```

888570	-3.28148526	0.989802279
889403	1.90709089	3.119201237
889719	-1.61336234	2.472397898
88995002	-6.61390681	5.997998811
8910251	1.46338831	-1.685149061
8910499	2.01038350	1.102516460
8910506	2.28233204	-0.009485293
8910720	0.73909276	-3.149749158
8910721	4.50260070	3.166486414
8910748	3.15597737	-0.409941090
8910988	-7.66940121	3.072602737
8910996	3.63593673	-1.588189625
8911163	-0.33704893	3.141609173
8911164	1.27546111	-0.848306563
8911230	4.34219338	0.321694999
8911670	-0.01741547	3.456302110
8911800	3.48935577	2.631766723
8911834	1.86490793	0.901271186
8912049	-4.95590744	1.339944038
8912055	2.46855062	0.137904923
89122	-3.78173258	1.900235409
8912280	-2.63493731	-0.576647607
8912284	1.37147827	-0.005800358
8912521	4.13524399	1.375298123
8912909	0.68633639	-1.693519506
8913	4.30833870	1.976746804
8913049	-0.40295486	-3.720160654
89143601	3.00096526	-0.353848878
89143602	-3.13125303	-4.269702670
8915	0.41746954	0.807865734
891670	1.41897459	-1.392752638
891703	2.86806475	0.268408338
891716	3.36288672	0.806890637
891923	3.31208587	1.440908382
891936	4.76551701	0.542484530
892189	2.38378528	0.823391074
892214	2.78464581	2.530807159
892399	3.39085817	-0.753301131
892438	-6.58444233	1.483557076
892604	1.45405111	-0.591422478
89263202	-7.17661843	-0.055097580
892657	3.57568917	-0.890799182
89296	3.05434514	0.179055836

893061	3.07173303	0.305789373
89344	3.85015201	1.523386003
89346	5.38551771	-0.555766489
893526	4.65358767	3.062692587
893548	4.02400169	2.540671520
893783	3.34487880	0.068502773
89382601	3.29694105	3.135950089
89382602	2.13452598	0.004388987
893988	4.03446724	0.240486452
894047	3.76210430	-4.394333943
894089	4.86695697	2.337158334
894090	4.74307511	1.796820963
894326	-1.89365006	2.390134936
894329	-1.57154834	-6.503377021
894335	4.02308037	1.400973753
894604	1.29372783	-3.467757378
894618	-1.83322742	4.317723823
894855	1.80878902	-0.395858988
895100	-7.23012357	0.035670411
89511501	3.14288478	0.741873771
89511502	2.88472980	0.464323069
89524	3.14688506	1.769691255
895299	4.62431058	1.834004631
8953902	-2.24853190	0.348229535
895633	-2.10594822	-1.120987593
896839	-1.78437991	0.268984957
896864	0.52723271	-1.264758769
897132	3.21067592	-1.102473082
897137	4.38256677	0.760434627
897374	3.80666840	0.909118626
89742801	-3.31675558	1.575666422
897604	2.39117199	-0.989160421
897630	-4.67605183	0.967741443
897880	3.06499827	-1.134839030
89812	-7.34675421	5.238014242
89813	-0.29505357	-0.226735533
898143	1.93726491	-2.542750760
89827	2.07535283	-1.804931340
898431	-4.97078859	1.331400687
89864002	2.17500311	-0.958068260
898677	2.42941383	-3.444173357
898678	3.36500493	0.562434015
89869	1.20553704	1.315886497

898690	3.32315522	-0.474753848
899147	1.92709976	-1.460184989
899187	4.03828183	1.355724562
899667	-6.17684877	-5.103514894
899987	-10.92468559	3.699998472
9010018	-2.11663452	-0.296368297
901011	3.00954689	-0.243545681
9010258	0.98178471	-0.796551513
9010259	-0.35757923	-2.125968004
901028	3.19111003	1.847526445
9010333	2.99060189	-1.629958015
901034301	3.19110631	-0.578830133
901034302	4.58792496	2.758699732
901041	2.48373073	1.187894183
9010598	2.39655663	0.250524355
9010872	0.77892982	2.122475872
9010877	3.13722800	1.486714010
901088	-2.73260737	3.941699950
9011494	-6.22155168	1.388887616
9011495	2.22182078	0.356846723
9011971	-5.30247440	6.717504288
9012000	-7.24162919	3.652255530
9012315	-4.20352196	-1.175208822
9012568	2.49743889	2.016761090
9012795	-3.63181933	1.954720962
901288	-3.51615443	3.855214937
9013005	2.61264945	1.101505251
901303	0.09602363	0.129733431
901315	-2.04305926	-6.421169887
9013579	3.06405348	2.180177374
9013594	1.60217670	-0.292338995
9013838	-3.76262144	-5.980033549
901549	0.92475292	-2.300457270
901836	3.28545874	-0.201134359
90250	1.55348525	-0.978996316
90251	0.39842934	-2.159312724
902727	2.29854534	0.931258175
90291	0.54242409	1.315910048
902975	2.05700950	-0.320153445
902976	3.42150749	2.443368505
903011	0.47199125	-3.699574203
90312	-4.69936579	0.195783486
90317302	3.64017727	-0.786168171

903483	3.59152918	-2.602797030
903507	-4.14544978	-0.766814129
903516	-7.66539681	0.859727681
903554	2.00861205	-0.429311811
903811	2.91826573	1.698977554
90401601	0.50029759	-0.106637635
90401602	1.90981814	0.650821226
904302	3.78177167	0.325859645
904357	2.70601296	0.219472629
90439701	-6.28414409	-2.034757025
904647	3.42775463	0.991737410
904689	1.99962900	0.293068594
9047	2.47111877	0.334752480
904969	3.64405137	1.240273815
904971	2.34412427	-0.682141105
905189	1.14199209	1.960446508
905190	1.34966974	-0.369249002
90524101	-2.76094145	1.077685260
905501	1.96759233	0.175693916
905502	2.92861019	0.494421191
905520	2.92380175	-0.377000536
905539	3.99267514	-0.959012260
905557	-0.15500167	0.437623166
905680	0.93326963	2.104094210
905686	1.92792536	-0.891682104
905978	2.62003325	-2.499901968
90602302	-5.99833869	0.090949196
906024	2.91930912	0.009204582
906290	3.33900683	0.022205192
906539	2.01314368	-0.777767778
906564	-1.98671401	-2.314011967
906616	1.66813615	-0.861560770
906878	0.37085180	-0.113512779
907145	1.84733447	-2.538995601
907367	4.69890766	0.431922385
907409	1.28205907	-2.548666752
90745	2.31221232	-0.401678043
90769601	4.65498928	0.781611021
90769602	3.94000659	2.028231657
907914	-4.94550694	-3.003421435
907915	0.91499842	-2.476833776
908194	-4.54511193	0.815283142
908445	-4.43843746	0.991566424

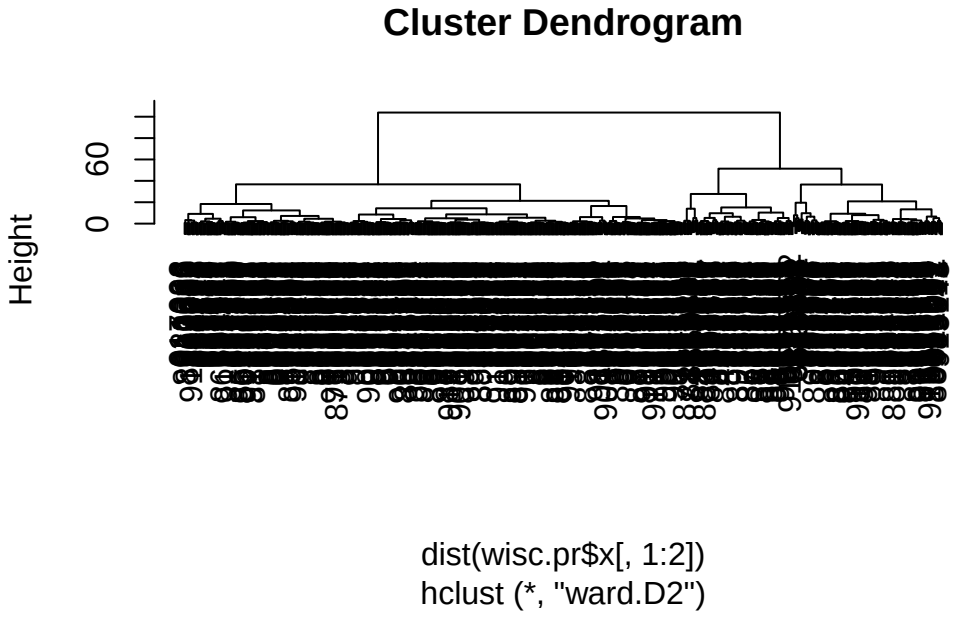
908469	2.19305464	1.803766114
908489	-0.66385679	-0.436476802
908916	2.23890605	0.454189872
909220	2.12331215	1.193746541
909231	2.61301823	1.830449830
909410	3.20814515	2.233177743
909411	0.30648073	-2.183077955
909445	-2.47018342	1.498506224
90944601	3.45168126	2.134583340
909777	3.89709781	-0.730133530
9110127	-0.98086448	2.208486712
9110720	1.25614629	-1.066752885
9110732	-3.24256054	1.776795240
9110944	1.55585043	1.037014643
911150	1.24815985	1.587442008
911157302	-4.32784530	4.045772565
9111596	1.07650412	-1.802032564
9111805	-2.50954497	2.526584455
9111843	2.21514321	-0.029865010
911201	1.17279871	0.474422064
911202	2.83380799	1.017230750
9112085	1.85545330	1.570011674
9112366	1.32744636	-0.776412499
9112367	2.80030738	1.664836753
9112594	3.17177797	2.073908798
9112712	4.08348433	0.484283961
911296201	-3.47243837	1.671413723
911296202	-16.30488665	7.769016888
9113156	2.55786437	2.491852880
911320501	2.96627878	0.068786611
911320502	2.75500634	1.792341696
9113239	-1.36913518	-2.108156394
9113455	0.41729731	-0.116417262
9113514	3.83502421	-0.898388162
9113538	-5.92575082	-1.227216861
911366	-0.64491119	-3.422999542
9113778	2.68148698	-1.442286469
9113816	2.03986918	0.902461858
911384	1.39806620	1.770668244
9113846	3.53330636	1.245758018
911391	1.98877953	-1.897792693
911408	1.99642872	0.206189711
911654	0.52020039	0.972512103

911673	3.16958159	2.087215768
911685	2.20077374	-1.284908734
911916	-3.82107090	-2.303209735
912193	2.98286009	0.672747298
91227	2.47887396	2.361880555
912519	1.27099312	-0.509385643
912558	2.23319191	1.297562775
912600	0.05613371	0.227183495
913063	-2.31535334	-4.385275755
913102	2.28232306	2.464672105
913505	-4.75043499	1.488114177
913512	1.72479328	-0.997300316
913535	0.74112965	2.449879331
91376701	2.89176621	0.977084588
91376702	1.65306732	4.551659142
914062	-3.10079133	1.235066483
914101	4.06069761	0.560461075
914102	2.79087498	1.076940539
914333	1.47362330	1.589695437
914366	0.22919297	-1.514570755
914580	2.55101378	0.763228924
914769	-3.71243855	1.057755506
91485	-5.09036286	2.017432706
914862	0.60008810	0.837717560
91504	-2.78794673	-3.382584454
91505	1.40753273	-1.504446838
915143	-7.25279714	5.490904302
915186	-1.29772469	-7.723057325
915276	-1.07574673	-8.287483337
91544001	1.24866183	-1.594281055
91544002	1.25582341	-4.113567176
915452	1.16550969	1.664229389
915460	-4.09054331	-2.800463137
91550	1.88597978	-1.670721680
915664	2.76500279	2.159148972
915691	-2.22503669	-1.939927827
915940	1.13058210	1.409761524
91594602	0.73312738	1.941988438
916221	2.33161916	-0.789451973
916799	-2.69485257	1.942001189
916838	-3.37604410	2.331377578
917062	0.19969487	-1.075413752
917080	1.17499204	-1.010481061

917092	1.29150016	-4.959860548
91762702	-8.62316838	3.456410638
91789	4.42556234	0.785345015
917896	0.61929383	-0.635789842
917897	3.24945659	-1.284794677
91805	3.34997942	-2.670950443
91813701	0.85904712	-0.096763381
91813702	3.15384692	0.870776432
918192	-0.34622347	-1.539870352
918465	2.45924278	-0.600319751
91858	1.52875558	-0.404861554
91903901	1.77202587	-0.803503014
91903902	2.67808519	1.483403662
91930402	-4.02489378	2.938844225
919537	2.13447312	-1.517245623
919555	-5.16086993	2.380108987
91979701	-0.53546080	-0.380380451
919812	-0.34282174	-3.531373815
921092	4.19339024	-2.365311059
921362	1.14182718	-5.594535868
921385	1.66401100	-2.387517361
921386	-1.01082308	-1.091429307
921644	1.29978604	1.819814057
922296	2.37134219	1.680097929
922297	1.66440651	0.213774641
922576	1.92598353	1.136739705
922577	4.23349159	-0.184110499
922840	2.67551655	-2.313756961
923169	3.83312511	-0.495813665
923465	2.54919727	-0.228129228
923748	4.69079604	0.766803238
923780	2.02325691	-1.260133116
924084	2.89340232	1.450359601
924342	3.49912218	-1.799249342
924632	2.15201013	0.829339088
924934	2.05327740	-1.615038205
924964	3.87388097	-1.083301553
925236	4.06028949	-0.122061034
925277	0.09858059	0.213372093
925291	1.08841850	-1.291711328
925292	0.48134743	0.177863190
925311	4.86602793	2.129232607
925622	-5.91241029	-3.479575000

```
926125     -8.73365338   0.573350185
926424     -6.43365455   3.573672989
926682     -3.79004753   3.580897052
926954     -1.25507494   1.900624364
927241    -10.36567336  -1.670540206
92751       5.47042990   0.670047220
```

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



We can "cut this tree to yield out clusters:

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

```
pc.grps
  1   2
195 374
```

How do my cluster grps compare to the diagnosis?

```
table(diagnosis, pc.grps)
```

```
         pc.grps
diagnosis   1   2
        B  18 339
        M 177  35
```

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

It does not separate out the two diagnoses very well. There are still different benign and malignant diagnoses in all the groups, mixed together.

> 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.

They did really badly. We do much better after PCA - the new pca variables (what we call a basis set) give us a much better separation of M and B.

7. Prediction

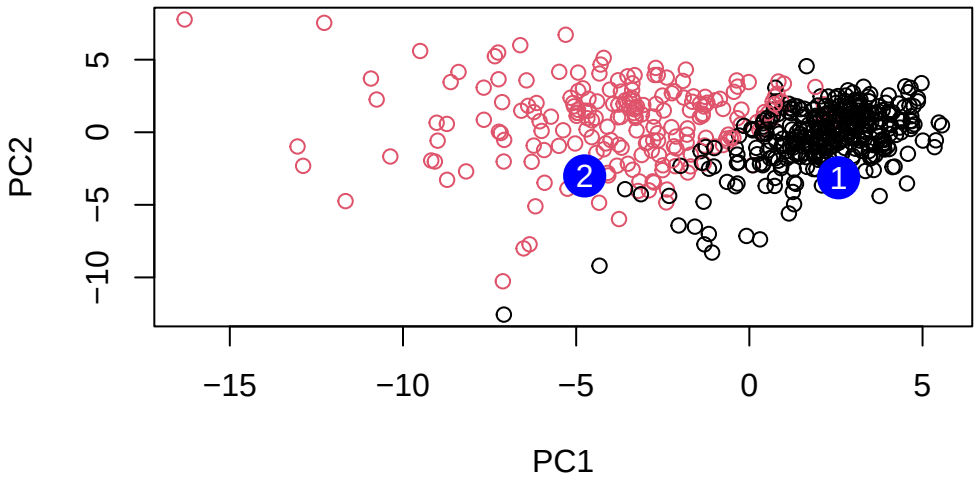
We can use our PCA model for the analysis of new "unseen" data. In this case form U Mich.

```
#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,]  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(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 patient 2 for followups, since patient 2 has a profile similar to that of malignant patients.