Class 8: Breast Cancer Mini Project

Ruofan Kang (A17236920)

Today we will apply the machine learning methods we introduced in the last class on breast cancer biopsy data from fine needle iration (FNA).

Data input

The data is supplied on CSV format:

```
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
```

Now I will store the diagnosis column for later and exclude it from the data set I will actually do things with that I will call 'wisc.data'

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

Q1 How many people are in this data set?

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

```
[1] 569
```

Q2 How many of the observations have a malignant diagnosis?

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

```
  B   M
357 212
```

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

```
[1] 212
```

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

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

```
[1] 10
```

```
x
```

```
 [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"
```

Principal Component Analysis

We need to scale our input data before PCA as some of the columns are measured in terms of very different units with different means and different variances. The upshot here is we set 'scale-TRUE' argument to 'prcomp()'.

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

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
                           PC8    PC9    PC10   PC11    PC12    PC13    PC14
Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 0.00523
Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812 0.98335
                          PC15    PC16    PC17    PC18    PC19    PC20   PC21
Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
                          PC22    PC23   PC24    PC25    PC26    PC27    PC28
Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987
Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005
Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997
                          PC29    PC30
Standard deviation     0.02736 0.01153
Proportion of Variance 0.00002 0.00000
Cumulative Proportion  1.00000 1.00000
```

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

```
proportion_PC1 <- summary(wisc.pr)$importance["Proportion of Variance", "PC1"]
proportion_PC1
```

```
[1] 0.44272
```

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

```
cumulative_proportions <- summary(wisc.pr)$importance["Cumulative Proportion",]
pcs_70 <- which(cumulative_proportions >= 0.70)[1]
pcs_70
```

```
PC3
  3
```

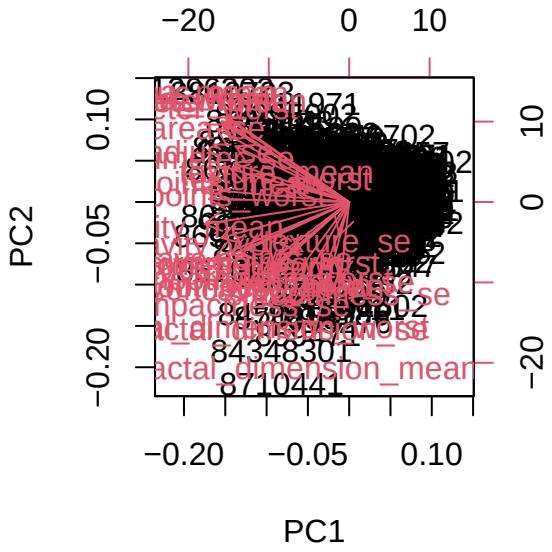
Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

```
pcs_90 <- which(cumulative_proportions >= 0.90)[1]
pcs_90
```

```
PC7
  7
```

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

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

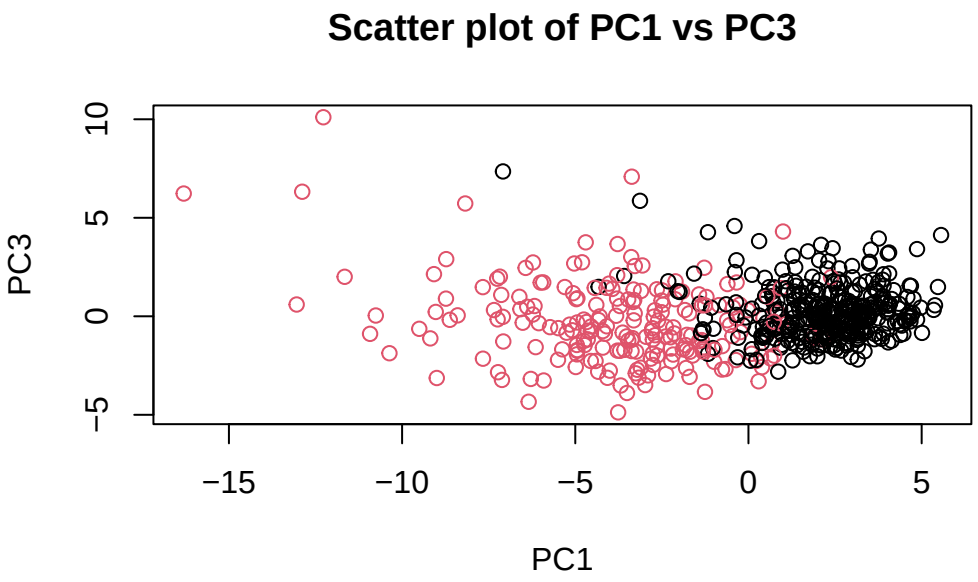


What strikes me about this plot is the confusion. The graph is very hard to understand. Because too many variables are plotted at the same time, the labels become overlapping, making the visualization difficult to interpret.

Generate one of our main result figures= the PC plot (a.k.a "score plot", "orientation plot","PC1 vs PC2 plot","PC plot", "projection plot",etc.) It is known by different names in different fields.

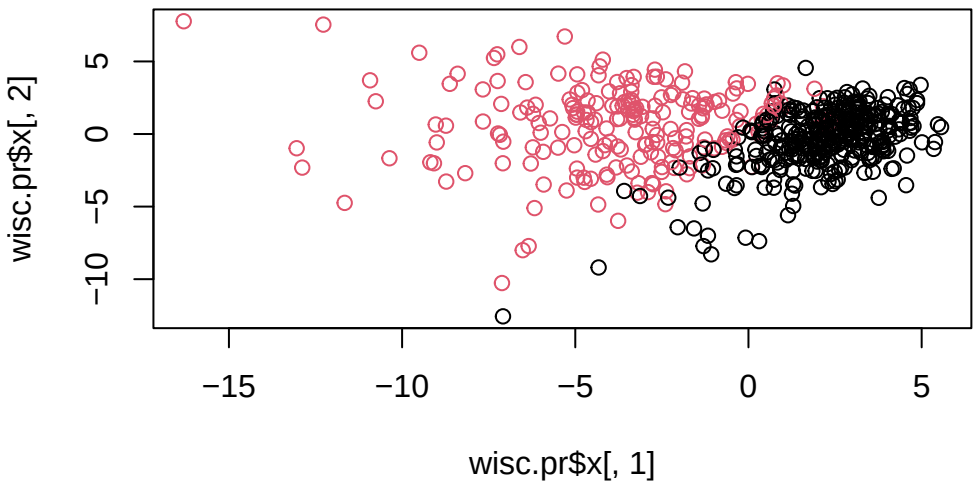
> Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

```
# Plot for PC1 vs PC3
plot(wisc.pr$x[, c(1, 3)], col = diagnosis,
     xlab = "PC1", ylab = "PC3", main = "Scatter plot of PC1 vs PC3")
```



I noticed two clusters for P1 and P3, and the clusters can distinguish where P1 and P3 are. However, there is also some overlap, and there may not be enough PC1 and PC3 to make a completely clear distinction.

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

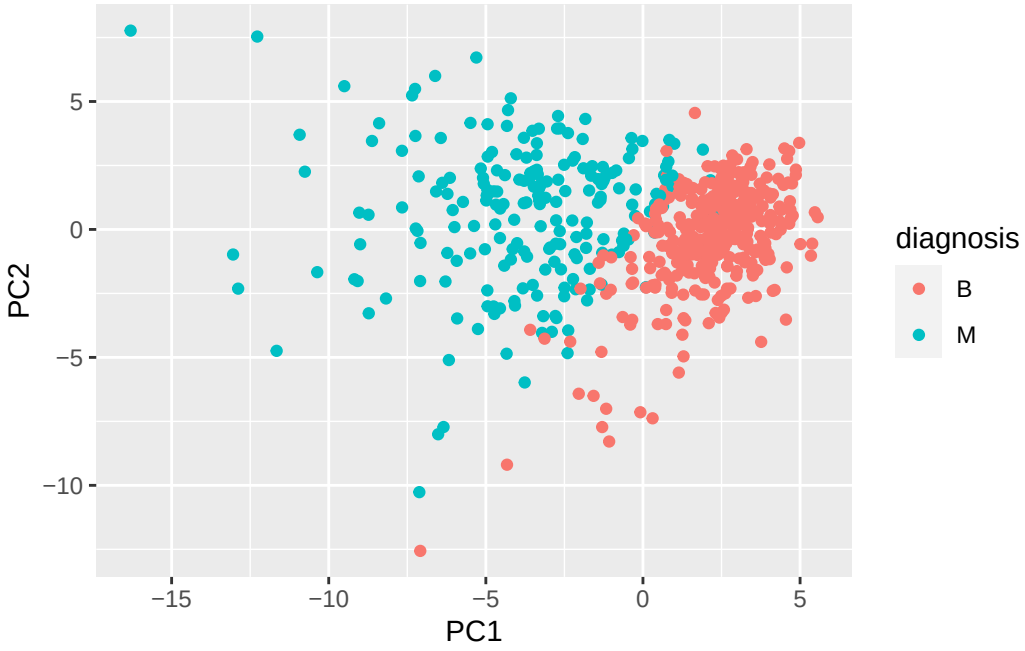


And a ggplot version

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

# Load the ggplot2 package
library(ggplot2)

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



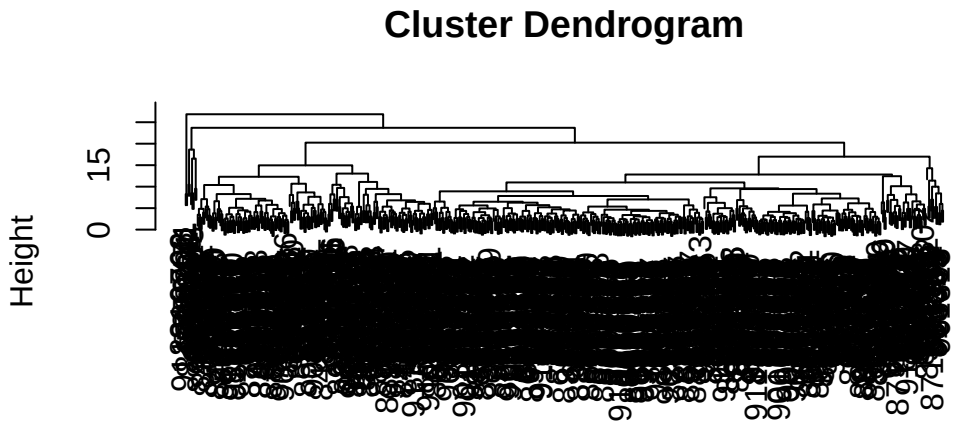
Hierarchical clustering

Can we just use clustering on the original data and get some insight into M vs B?

It is rather difficult, this "tree" looks like a hot mess...

```
#distance matrix needed for hclust
data.dist <- dist( scale(wisc.data) )

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

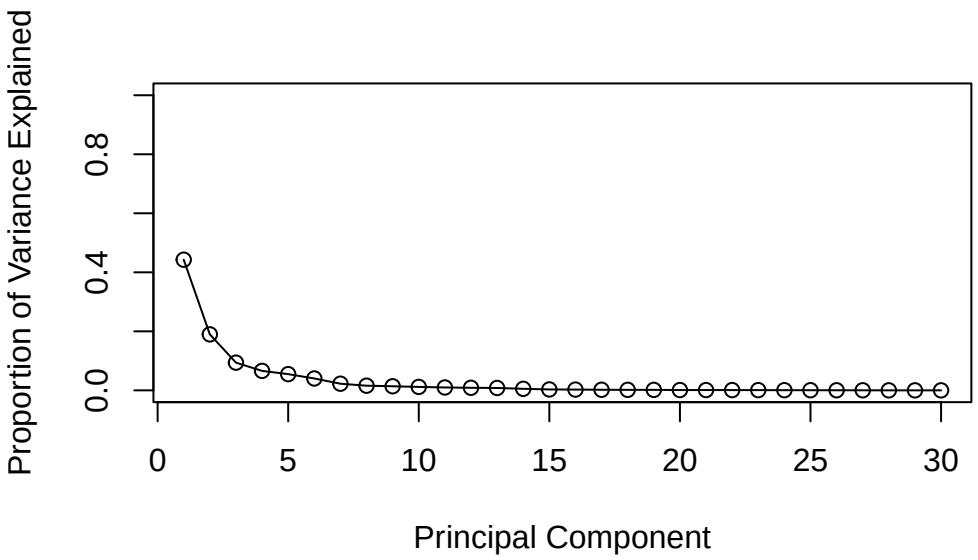



```
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

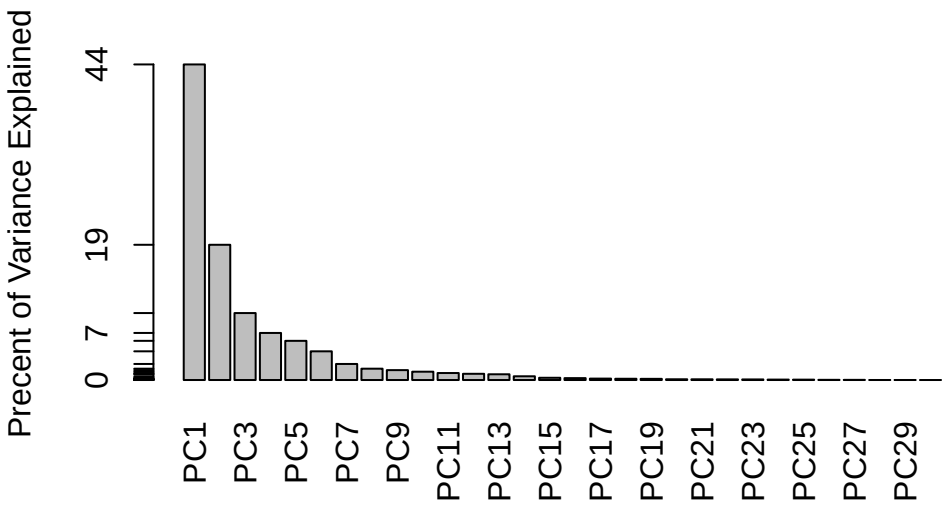
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
# 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")
```



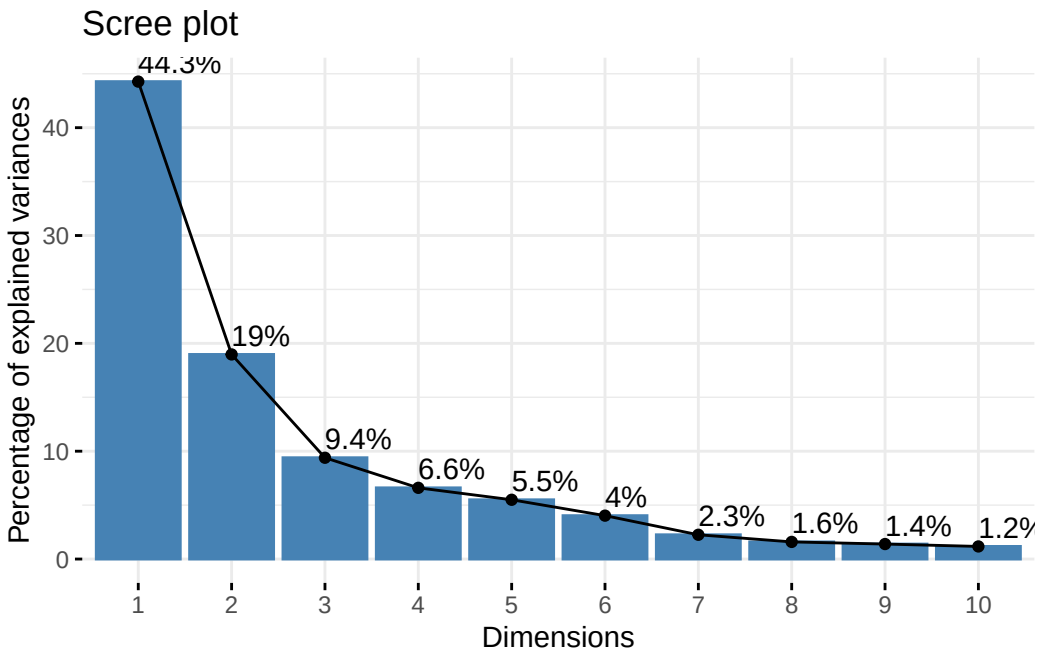

```
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```




```
## ggplot based graph
#install.packages("factoextra")
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(wisc.pr, addlabels = TRUE)
```



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?

```
loading_value <- wisc.pr$rotation["concave.points_mean", 1]
loading_value
```

```
[1] -0.2608538
```

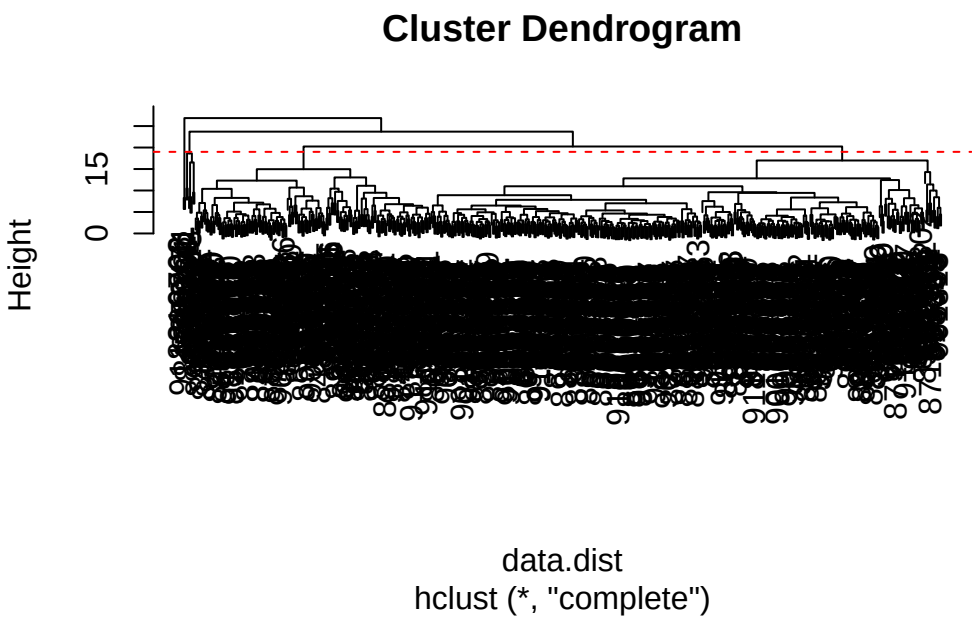
Q10. What is the minimum number of principal components required to explain 80% of the variance of the data?

```
cumulative_pve <- cumsum(pve)
min_pc <- which(cumulative_pve >= 0.80)[1]
min_pc
```

```
[1] 5
```

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

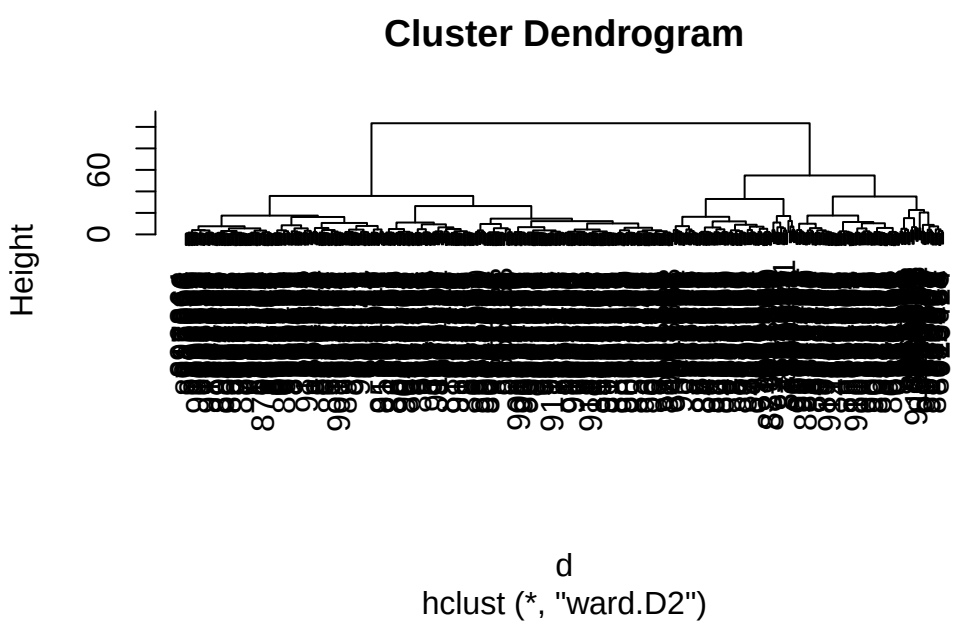
```
plot(wisc.hclust, main="Cluster Dendrogram")
abline(h=19, col="red", lty=2)
```



5.Combiming methods

This apprach will take not original data but our PCA results and work with them.

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



Generate 2 cluster groups from this hclust object.

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

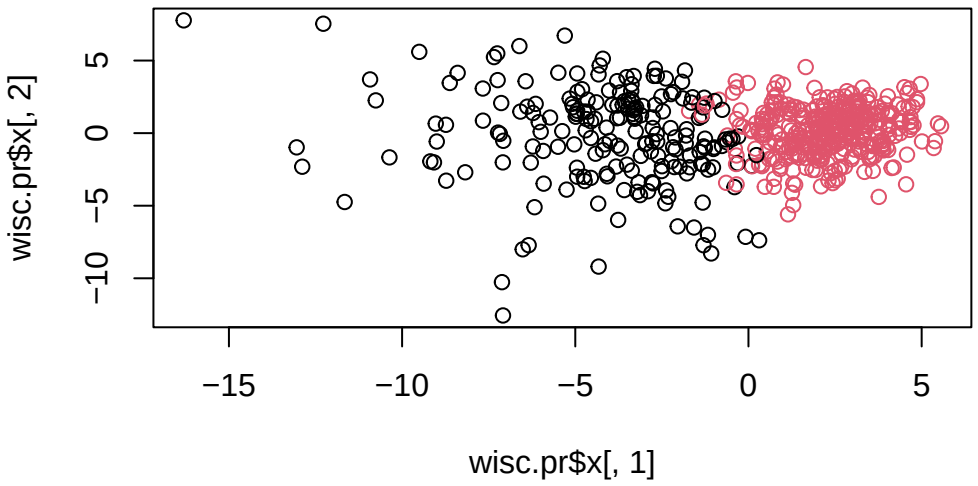
```
  842302    842517  84300903  84348301  84358402    843786    844359  84458202
       1         1         1         1         1         1         1         1
  844981  84501001    845636  84610002    846226    846381  84667401  84799002
       1         1         2         1         1         2         1         1
  848406  84862001    849014   8510426   8510653   8510824   8511133    851509
       2         1         1         2         2         2         1         1
  852552    852631    852763    852781    852973    853201    853401    853612
       1         1         1         1         1         2         1         1
85382601    854002    854039    854253    854268    854941    855133    855138
       1         1         1         1         1         2         2         1
  855167    855563    855625    856106  85638502    857010  85713702     85715
       2         1         1         1         2         1         2         1
  857155    857156    857343    857373    857374    857392    857438  85759902
       2         2         2         2         2         1         2         2
  857637    857793    857810    858477    858970    858981    858986    859196
       1         1         2         2         2         2         1         2
85922302    859283    859464    859465    859471    859487    859575    859711
```

1	1	2	2	1	2	1	1
859717	859983	8610175	8610404	8610629	8610637	8610862	8610908
1	2	2	2	2	1	1	2
861103	8611161	8611555	8611792	8612080	8612399	86135501	86135502
2	1	1	1	2	1	2	1
861597	861598	861648	861799	861853	862009	862028	86208
2	1	2	2	2	2	1	1
86211	862261	862485	862548	862717	862722	862965	862980
2	2	2	1	2	2	2	2
862989	863030	863031	863270	86355	864018	864033	86408
2	1	2	2	1	2	2	2
86409	864292	864496	864685	864726	864729	864877	865128
1	2	2	2	2	1	1	2
865137	86517	865423	865432	865468	86561	866083	866203
2	1	1	2	2	2	2	1
866458	866674	866714	8670	86730502	867387	867739	868202
1	1	2	1	1	2	1	2
868223	868682	868826	868871	868999	869104	869218	869224
2	2	1	2	2	2	2	2
869254	869476	869691	86973701	86973702	869931	871001501	871001502
2	2	1	2	2	2	2	1
8710441	87106	8711002	8711003	8711202	8711216	871122	871149
1	2	2	2	1	2	2	2
8711561	8711803	871201	8712064	8712289	8712291	87127	8712729
2	1	1	2	1	2	2	2
8712766	8712853	87139402	87163	87164	871641	871642	872113
1	2	2	2	1	2	2	2
872608	87281702	873357	873586	873592	873593	873701	873843
1	1	2	2	1	1	1	2
873885	874158	874217	874373	874662	874839	874858	875093
2	2	2	2	2	2	1	2
875099	875263	87556202	875878	875938	877159	877486	877500
2	1	1	2	1	1	1	1
877501	877989	878796	87880	87930	879523	879804	879830
2	1	1	1	2	2	2	2
8810158	8810436	881046502	8810528	8810703	881094802	8810955	8810987
1	2	1	2	1	1	1	1
8811523	8811779	8811842	88119002	8812816	8812818	8812844	8812877
2	2	1	1	2	2	2	1
8813129	88143502	88147101	88147102	88147202	881861	881972	88199202
2	2	2	2	2	1	1	2
88203002	88206102	882488	88249602	88299702	883263	883270	88330202
2	1	2	2	1	1	2	1

88350402	883539	883852	88411702	884180	884437	884448	884626
2	2	1	2	1	2	2	1
88466802	884689	884948	88518501	885429	8860702	886226	886452
2	2	1	2	1	1	1	1
88649001	886776	887181	88725602	887549	888264	888570	889403
1	1	1	1	1	2	1	2
889719	88995002	8910251	8910499	8910506	8910720	8910721	8910748
1	1	2	2	2	2	2	2
8910988	8910996	8911163	8911164	8911230	8911670	8911800	8911834
1	2	2	2	2	2	2	2
8912049	8912055	89122	8912280	8912284	8912521	8912909	8913
1	2	1	1	2	2	2	2
8913049	89143601	89143602	8915	891670	891703	891716	891923
1	2	1	2	2	2	2	2
891936	892189	892214	892399	892438	892604	89263202	892657
2	2	2	2	1	2	1	2
89296	893061	89344	89346	893526	893548	893783	89382601
2	2	2	2	2	2	2	2
89382602	893988	894047	894089	894090	894326	894329	894335
2	2	2	2	2	1	1	2
894604	894618	894855	895100	89511501	89511502	89524	895299
2	1	2	1	2	2	2	2
8953902	895633	896839	896864	897132	897137	897374	89742801
1	1	1	2	2	2	2	1
897604	897630	897880	89812	89813	898143	89827	898431
2	1	2	1	1	2	2	1
89864002	898677	898678	89869	898690	899147	899187	899667
2	2	2	2	2	2	2	1
899987	9010018	901011	9010258	9010259	901028	9010333	901034301
1	1	2	2	2	2	2	2
901034302	901041	9010598	9010872	9010877	901088	9011494	9011495
2	2	2	2	2	1	1	2
9011971	9012000	9012315	9012568	9012795	901288	9013005	901303
1	1	1	2	1	1	2	2
901315	9013579	9013594	9013838	901549	901836	90250	90251
1	2	2	1	2	2	2	2
902727	90291	902975	902976	903011	90312	90317302	903483
2	2	2	2	2	1	2	2
903507	903516	903554	903811	90401601	90401602	904302	904357
1	1	2	2	2	2	2	2
90439701	904647	904689	9047	904969	904971	905189	905190
1	2	2	2	2	2	2	2
90524101	905501	905502	905520	905539	905557	905680	905686

1	2	2	2	2	2	2	2
905978	90602302	906024	906290	906539	906564	906616	906878
2	1	2	2	2	1	2	2
907145	907367	907409	90745	90769601	90769602	907914	907915
2	2	2	2	2	2	1	2
908194	908445	908469	908489	908916	909220	909231	909410
1	1	2	1	2	2	2	2
909411	909445	90944601	909777	9110127	9110720	9110732	9110944
2	1	2	2	1	2	1	2
911150	911157302	9111596	9111805	9111843	911201	911202	9112085
2	1	2	1	2	2	2	2
9112366	9112367	9112594	9112712	911296201	911296202	9113156	911320501
2	2	2	2	1	1	2	2
911320502	9113239	9113455	9113514	9113538	911366	9113778	9113816
2	1	2	2	1	2	2	2
911384	9113846	911391	911408	911654	911673	911685	911916
2	2	2	2	2	2	2	1
912193	91227	912519	912558	912600	913063	913102	913505
2	2	2	2	2	1	2	1
913512	913535	91376701	91376702	914062	914101	914102	914333
2	2	2	2	1	2	2	2
914366	914580	914769	91485	914862	91504	91505	915143
1	2	1	1	2	1	2	1
915186	915276	91544001	91544002	915452	915460	91550	915664
1	1	2	2	2	1	2	2
915691	915940	91594602	916221	916799	916838	917062	917080
1	2	2	2	1	1	2	2
917092	91762702	91789	917896	917897	91805	91813701	91813702
2	1	2	2	2	2	2	2
918192	918465	91858	91903901	91903902	91930402	919537	919555
2	2	2	2	2	1	2	1
91979701	919812	921092	921362	921385	921386	921644	922296
1	2	2	2	2	1	2	2
922297	922576	922577	922840	923169	923465	923748	923780
2	2	2	2	2	2	2	2
924084	924342	924632	924934	924964	925236	925277	925291
2	2	2	2	2	2	2	2
925292	925311	925622	926125	926424	926682	926954	927241
2	2	1	1	1	1	2	1
92751							
2							

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



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

No, I cannot find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10. if I cut it below 4, one cluster would have both malignant cells and benign cells. if I cut it above 4, the data stratification becomes less distinct.

```
wisc.hclust.clusters <- cutree(wisc.hclust, k=4)
table(wisc.hclust.clusters, diagnosis)
```

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

```
wisc.hclust.clusters <- cutree(wisc.hclust, k=3)
table(wisc.hclust.clusters, diagnosis)
```

```
                    diagnosis
wisc.hclust.clusters   B   M
                   1 355 205
                   2   2   5
                   3   0   2
```

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

My personal favorite method is "ward.D2", especially for this dataset where the inherent cluster structures are not very clear-cut. The Ward's method provides a good balance between the shape and size of clusters. It has the ability to search for clusters that are coherent internally, but distinct from each other.

Q14. How well does k-means separate the two diagnoses? How does it compare to your hclust results?

With the given data and the interpretation, it seems that both clustering methods have their strengths. The k-means algorithm seems to be providing a simpler split with two clusters, while the hierarchical method offers more granularity with its four clusters, potentially capturing subgroups within the data.

```
scaled_data <- scale(wisc.data)
wisc.km <- kmeans(scaled_data, centers=2, nstart=20)
comparison_table <- table(wisc.km$cluster, diagnosis)
print(comparison_table)
```

```
  diagnosis
     B   M
  1 343  37
  2  14 175
```

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

The newly created model effectively distinguishes between the two diagnoses: benign (B) and malignant (M).The table shows that cluster 1 has 28 benign samples; 188 malignant samples. Most of the malignant samples are clustered in this cluster. Cluster 2 has 329 benign samples; 24 malignant samples and most of the benign samples are clustered in this cluster.In an ideal scenario, each cluster would exclusively represent one type of diagnosis, either benign or malignant. Despite this, the current model offers a clear categorization and remains a valuable tool.

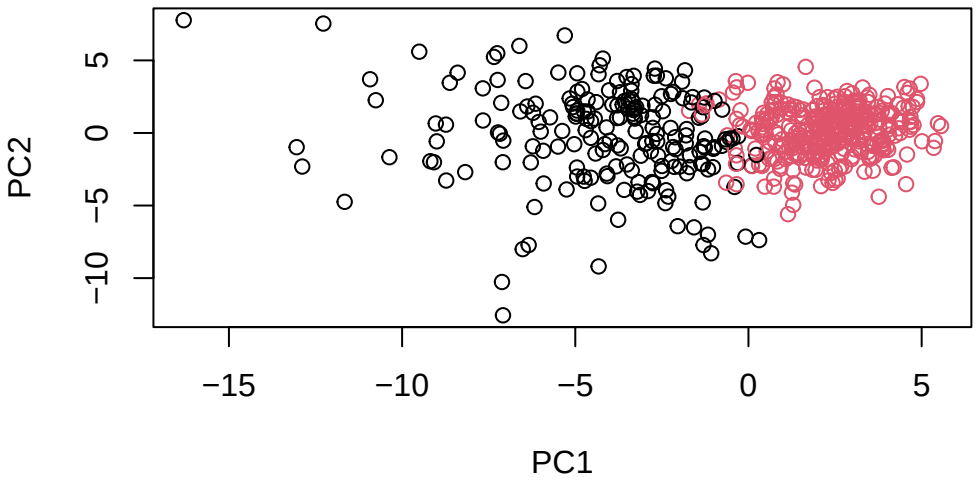
```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
  1   2
203 366
```

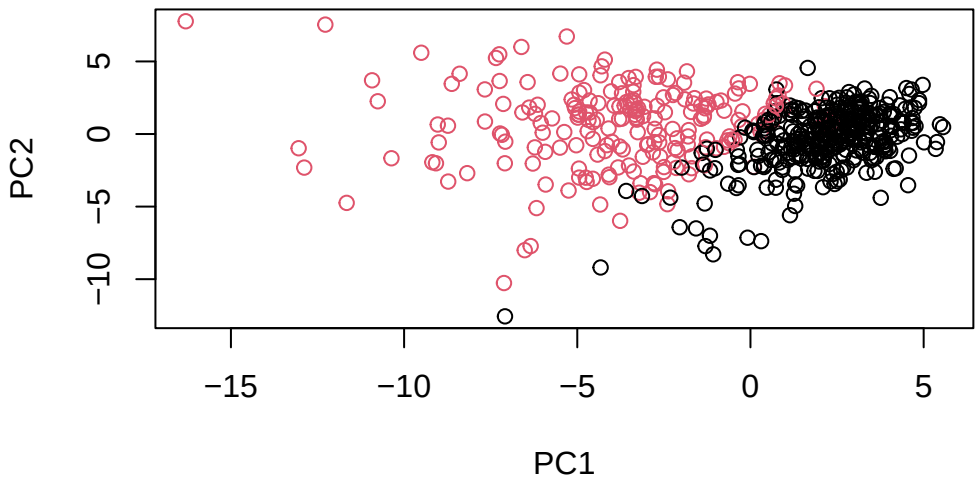
```
table(grps, diagnosis)
```

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

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




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



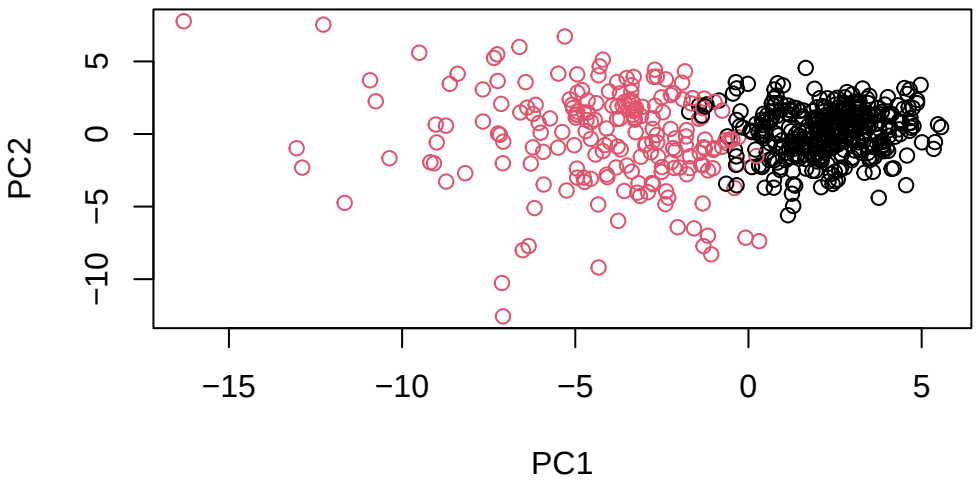

```
g <- as.factor(grps)
levels(g)
```

```
[1] "1" "2"
```

```
g <- relevel(g,2)
levels(g)
```

```
[1] "2" "1"
```

```
# Plot using our re-ordered factor
plot(wisc.pr$x[,1:2], col=g)
```




```
library(rgl)
plot3d(wisc.pr$x[,1:3], xlab="PC 1", ylab="PC 2", zlab="PC 3", cex=1.5, size=1, type="s",
```

```
dist.matrix <- dist(wisc.pr$x[, 1:7])
```

```
wisc.pr.hclust <- hclust(dist.matrix, method="ward.D2")
```

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

```
comparison.table <- table(wisc.pr.hclust.clusters, diagnosis)
print(comparison.table)
```

```
                     diagnosis
wisc.pr.hclust.clusters   B   M
                      1  28 188
                      2 329  24
```

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.

The k-mean clustering model distinguishes between two different diagnoses, with Cluster 1 grouping primarily malignant samples but still having 28 benign samples present. Cluster 2 groups mainly benign samples, but 24 malignant samples are still present. The hierarchical clustering model, on the other hand, uses four clusters, which is more detailed in comparison. Cluster 1 is predominantly malignant samples, but there are still a few benign samples present. Cluster 2 is mixed with a very small overall number, Cluster 3 is predominantly benign but a small number of malignant samples are present, and Cluster 4 is completely malignant but has a sample size of only two. Although neither sample is perfect, it is still possible to distinguish the separation of the two samples and be able to analyze the data in a valuable way.

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

```
   diagnosis
      B   M
  1 343  37
  2  14 175
```

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

```
                    diagnosis
wisc.hclust.clusters   B   M
                   1 355 205
                   2   2   5
                   3   0   2
```

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

The best specificity:K-means clustering model The best sensitivity: hierarchical clustering model

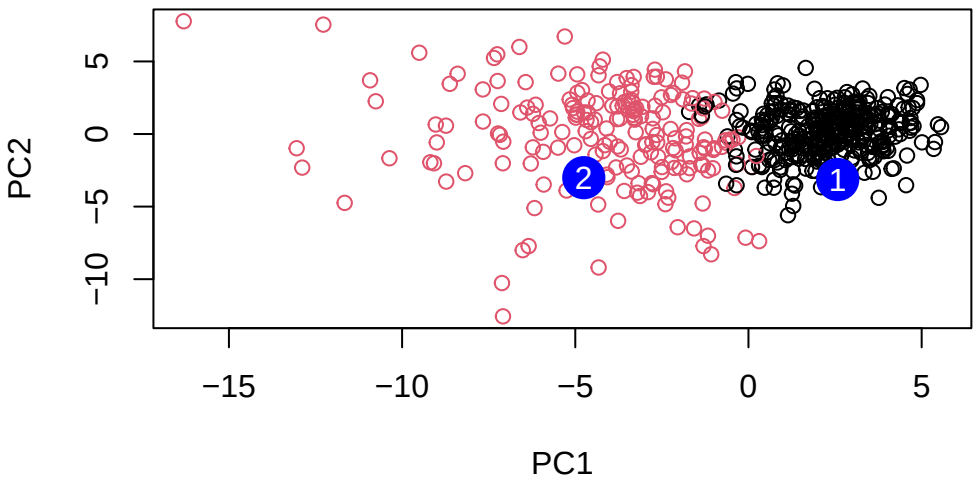
Q18. Which of these new patients should we prioritize for follow up based on your results?

Patient 1 was prioritized for follow-up. This is because patient 1 appears to be closer to dense clusters of dots based on the PCA plot, which may represent that the patient is a malignant sample and requires faster treatment.

```
#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=g)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```


PC2
5
0
−5
−10
−15
−10
−5
0
5
PC1
2
1