

Lab 08 Mini-Project: Analysis of Human Breast Cancer Cells

Divya Shetty (A15390408)

2/13/2022

Exploratory Data Analysis

Read the input csv data file to extract the data for analysis.

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

wisc.df <- read.csv(fna.data, row.names = 1)
head(wisc.df)
```

```
##      diagnosis radius_mean texture_mean perimeter_mean area_mean
## 842302         M      17.99      10.38      122.80      1001.0
## 842517         M      20.57      17.77      132.90      1326.0
## 84300903        M      19.69      21.25      130.00      1203.0
## 84348301         M      11.42      20.38       77.58       386.1
## 84358402         M      20.29      14.34      135.10      1297.0
## 843786          M      12.45      15.70       82.57       477.1
##      smoothness_mean compactness_mean concavity_mean concave.points_mean
## 842302      0.11840      0.27760      0.3001      0.14710
## 842517      0.08474      0.07864      0.0869      0.07017
## 84300903      0.10960      0.15990      0.1974      0.12790
## 84348301      0.14250      0.28390      0.2414      0.10520
## 84358402      0.10030      0.13280      0.1980      0.10430
## 843786      0.12780      0.17000      0.1578      0.08089
##      symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 842302      0.2419      0.07871      1.0950      0.9053      8.589
## 842517      0.1812      0.05667      0.5435      0.7339      3.398
## 84300903      0.2069      0.05999      0.7456      0.7869      4.585
## 84348301      0.2597      0.09744      0.4956      1.1560      3.445
## 84358402      0.1809      0.05883      0.7572      0.7813      5.438
## 843786      0.2087      0.07613      0.3345      0.8902      2.217
##      area_se smoothness_se compactness_se concavity_se concave.points_se
## 842302     153.40      0.006399      0.04904      0.05373      0.01587
## 842517      74.08      0.005225      0.01308      0.01860      0.01340
## 84300903      94.03      0.006150      0.04006      0.03832      0.02058
## 84348301      27.23      0.009110      0.07458      0.05661      0.01867
## 84358402      94.44      0.011490      0.02461      0.05688      0.01885
## 843786      27.19      0.007510      0.03345      0.03672      0.01137
##      symmetry_se fractal_dimension_se radius_worst texture_worst
## 842302      0.03003      0.006193      25.38      17.33
## 842517      0.01389      0.003532      24.99      23.41
## 84300903      0.02250      0.004571      23.57      25.53
```

## 84348301	0.05963	0.009208	14.91	26.50
## 84358402	0.01756	0.005115	22.54	16.67
## 843786	0.02165	0.005082	15.47	23.75
##	perimeter_worst	area_worst	smoothness_worst	compactness_worst
## 842302	184.60	2019.0	0.1622	0.6656
## 842517	158.80	1956.0	0.1238	0.1866
## 84300903	152.50	1709.0	0.1444	0.4245
## 84348301	98.87	567.7	0.2098	0.8663
## 84358402	152.20	1575.0	0.1374	0.2050
## 843786	103.40	741.6	0.1791	0.5249
##	concavity_worst	concave.points_worst	symmetry_worst	
## 842302	0.7119	0.2654	0.4601	
## 842517	0.2416	0.1860	0.2750	
## 84300903	0.4504	0.2430	0.3613	
## 84348301	0.6869	0.2575	0.6638	
## 84358402	0.4000	0.1625	0.2364	
## 843786	0.5355	0.1741	0.3985	
##	fractal_dimension_worst			
## 842302		0.11890		
## 842517		0.08902		
## 84300903		0.08758		
## 84348301		0.17300		
## 84358402		0.07678		
## 843786		0.12440		

Remove the “diagnosis” column and store it in a separate data frame. The column indicates the expert diagnosis of whether a cell sample is malignant or benign.

```
#removing diagnosis from df
wisc.data <- wisc.df[,-1]

#store diagnosis in vector
diagnosis <- factor(wisc.df[,1])
```

Q1. How many observations are in this dataset?

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

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

There are 569 observations in the dataset.

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

```
length(grep("M", diagnosis))
```

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

There are 212 observations with a malignant diagnosis.

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

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

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

There are 10 variable suffixed with “_mean” in the data.

Principal Component Analysis

Performing PCA

Check the mean and standard deviation of the dataset features to determine if the data needs to be scaled before creating a PCA plot.

```
#mean
```

```
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
```

```
#s.d.
```

```
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
```

If the input variables have different units of measurement or significantly different variances, the data should be scaled.

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

```
#summary of variation
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)?

The proportion of variance captured by PC1 is about 44.27%.

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

The cumulative proportion for PC3 is 72.636%, the first to be greater than or equal to 70%. Therefore, 3 principal components are required.

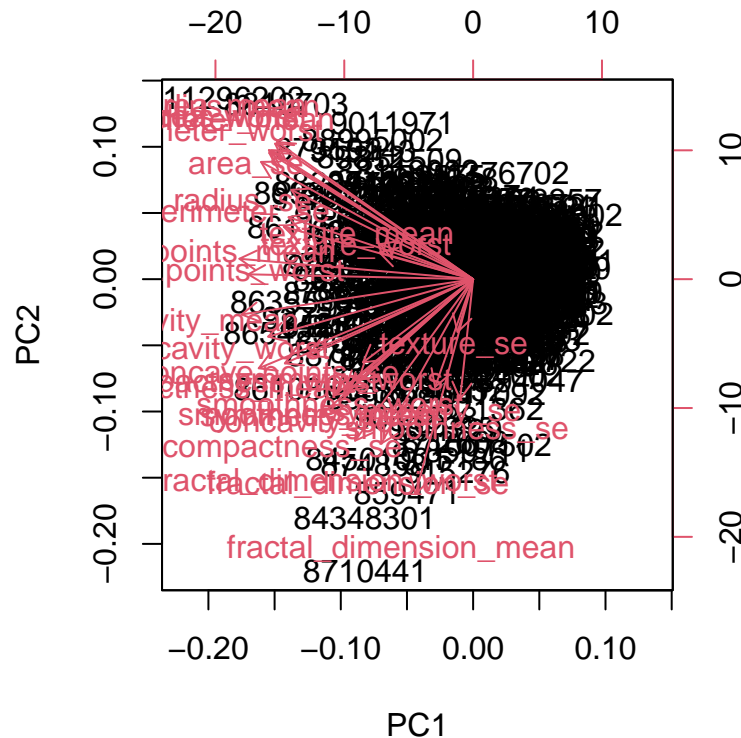
Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

The cumulative proportion for PC7 is 91.010%, which is greater than 90%. Therefore, 7 principal components are required.

Interpreting PCA Results

Various visualizations of the PCA results will be generated in order to determine the best way to interpret the model.

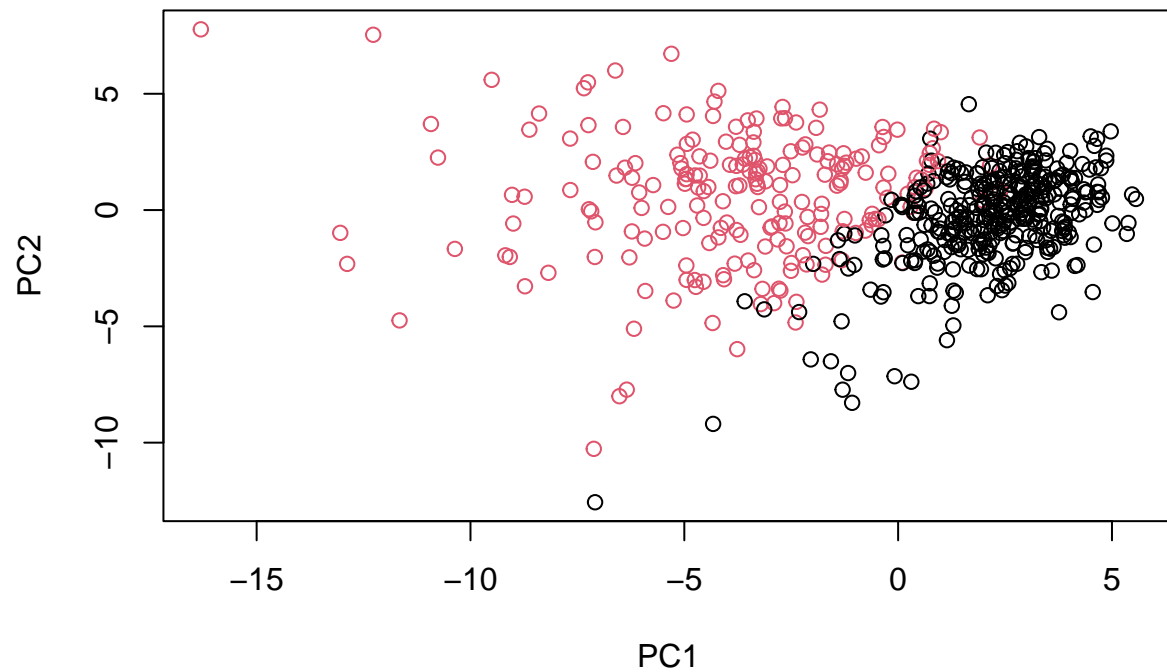
```
#biplot of PCA
biplot(wisc.pr)
```



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

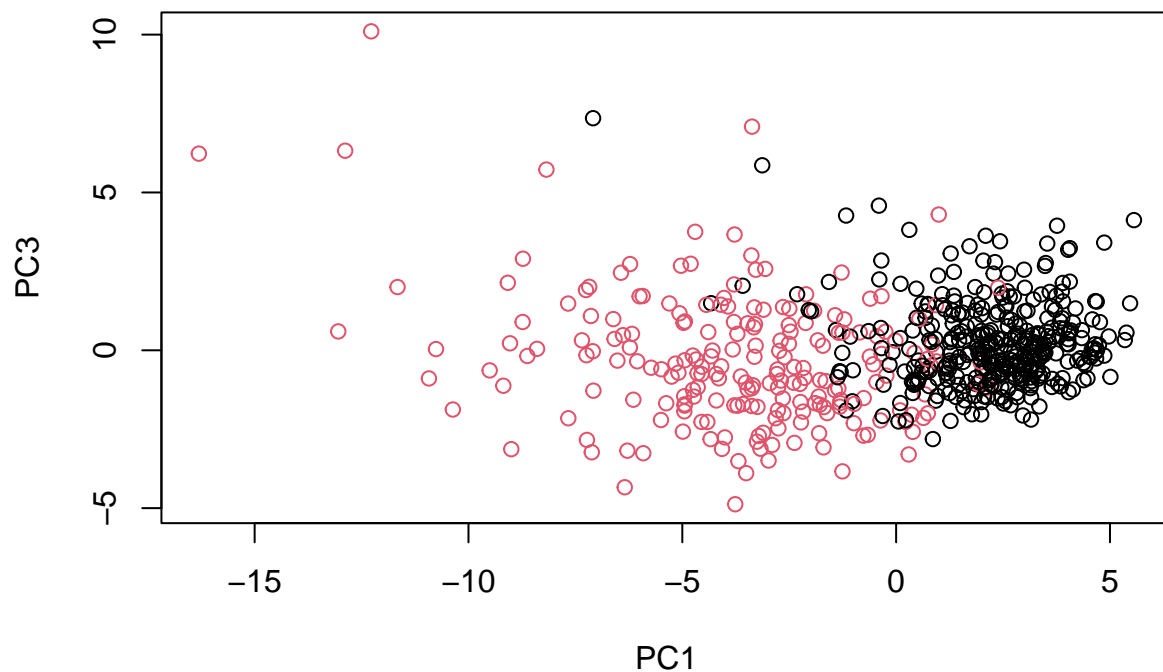
The biplot is very messy. Since there are so many values, it's difficult to read the plot and make any kind of analysis with it.

```
#scatterplot wiith PC1 and PC2
plot(wisc.pr$x[,1], wisc.pr$x[,2], col = diagnosis,
      xlab = "PC1", ylab = "PC2")
```



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

```
plot(wisc.pr$x[,1], wisc.pr$x[,3], col = diagnosis,  
     xlab = "PC1", ylab = "PC3")
```



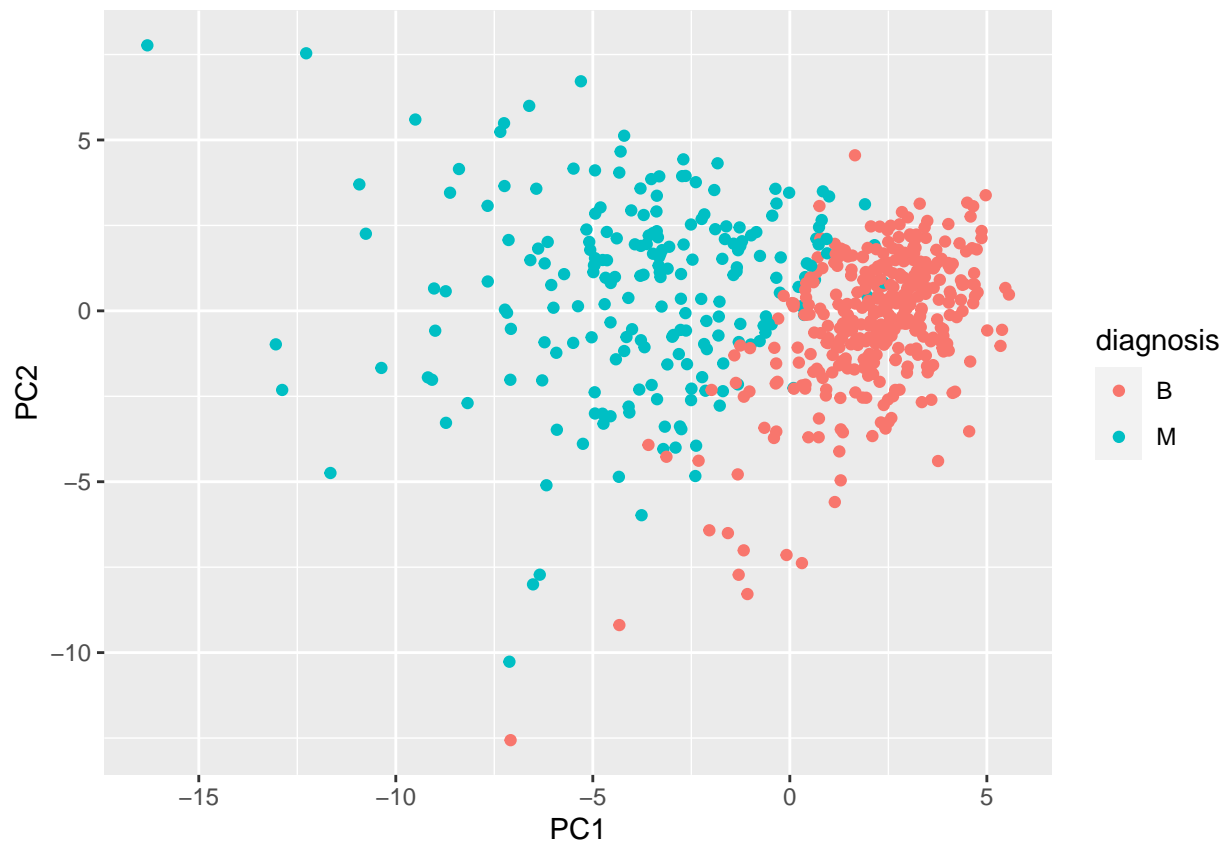
Since both scatterplots use points and not text like the biplot, it's easier to visualize the PCA results. There is a clearer divide between the subgroups (red = malignant, black = benign) for the first scatterplot, most likely because PC2 has a greater proportion of variance than PC3. Overall, both plots show that PC1 captures a separation of malignant and benign samples.

Make a ggplot for a fancier visualization!

```
#load package
library(ggplot2)

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

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



Variance Explained

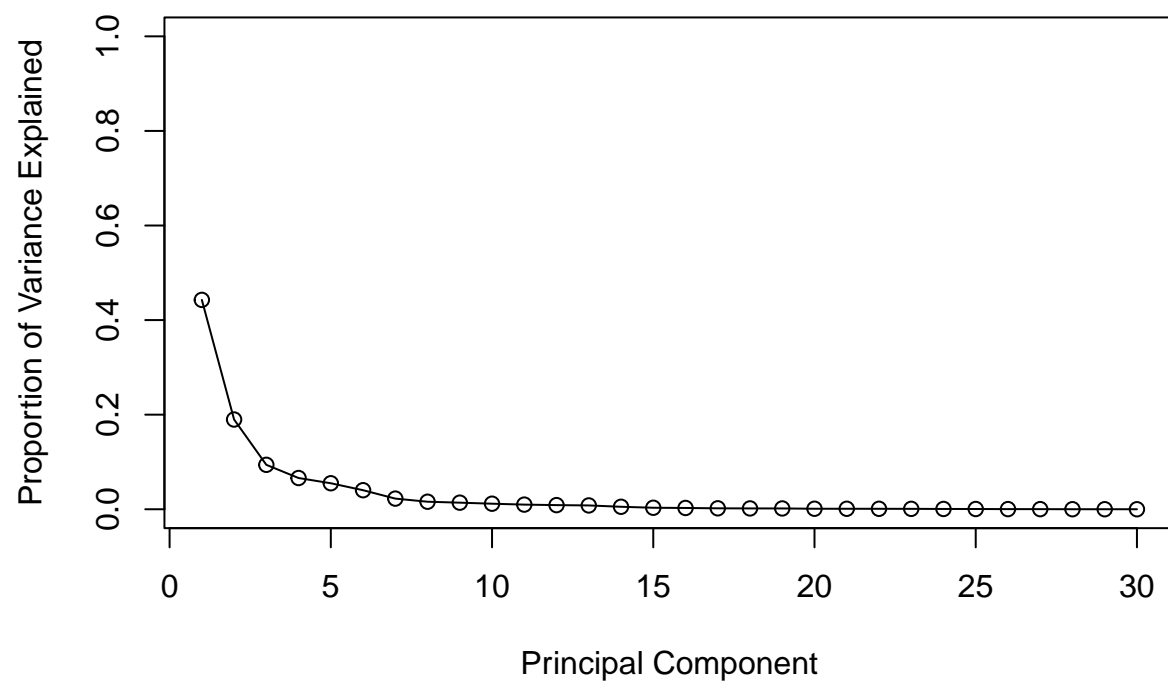
Scree plots show the proportion of variance explained as the number of principal components increases. If there's an 'elbow' in the amount of variance explained, this may indicate a natural number of principal components. The variance of each principal component can be calculated by squaring the sdev component of "wisc.pr".

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

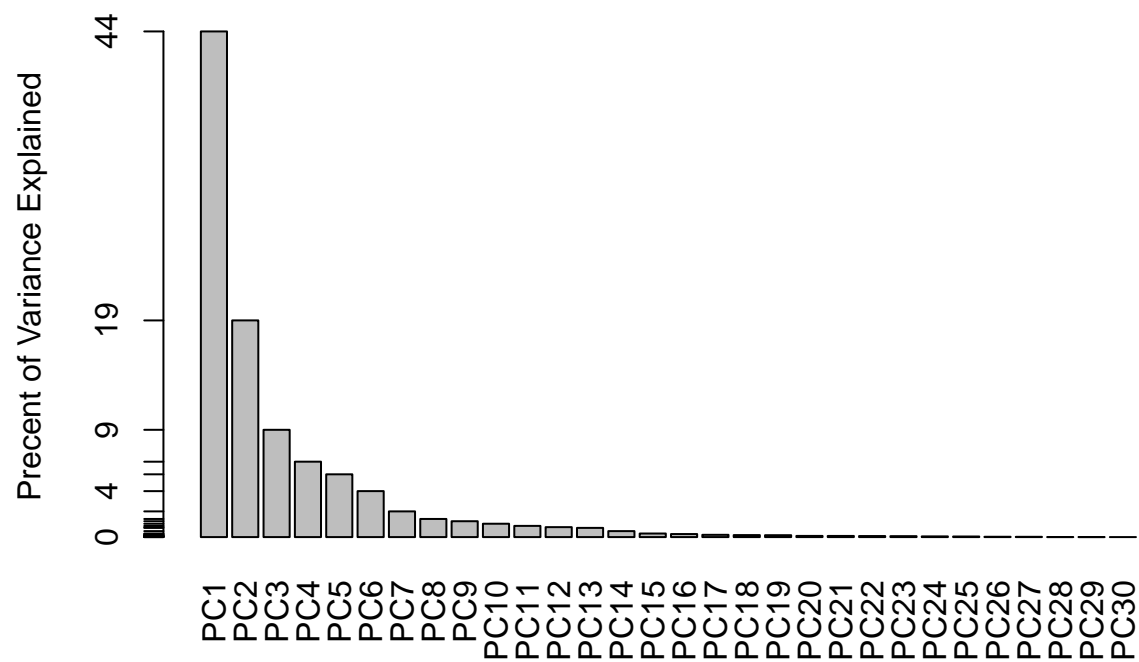
```
## [1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

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

```
#plot of 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 )
```

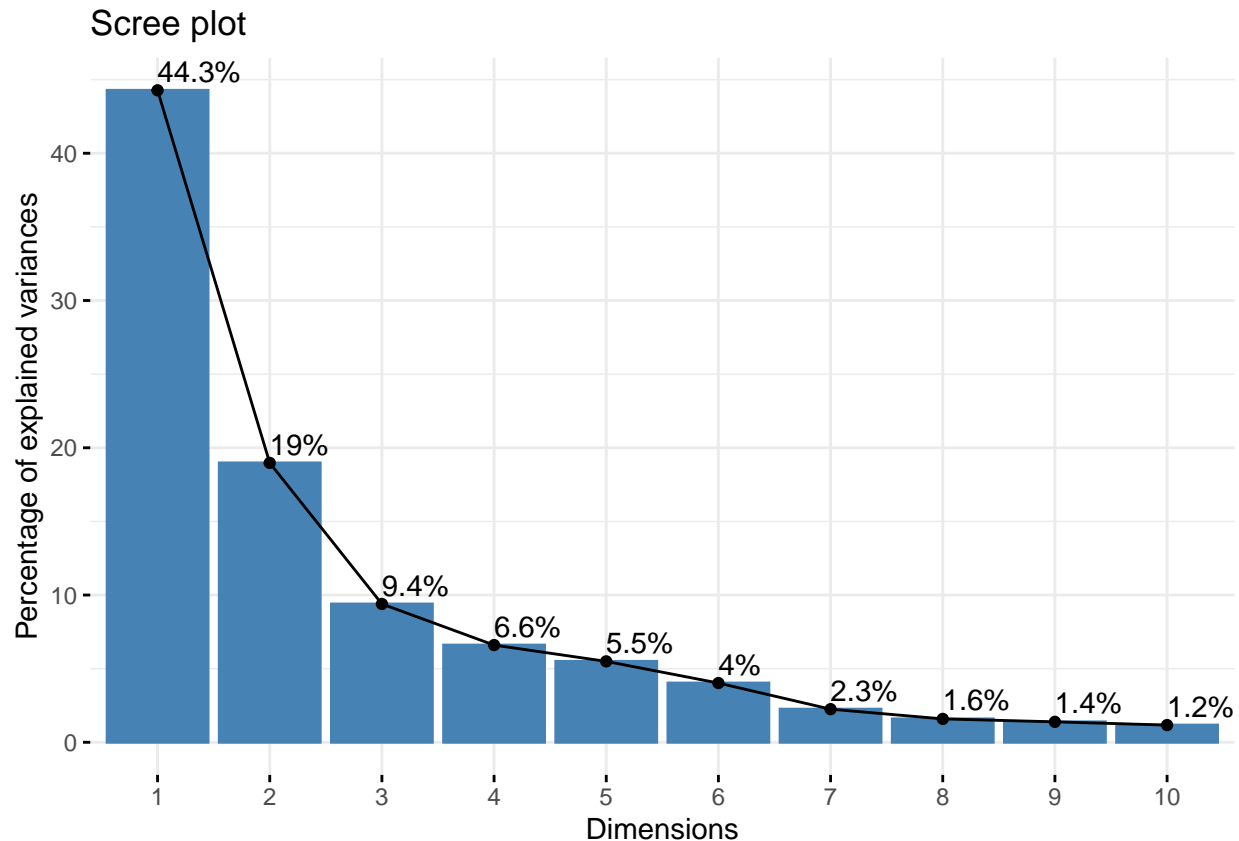


OPTIONAL: exploring the factoextra package!

```
#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)
```



Communicating PCA Results

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

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

```
## [1] -0.2608538
```

The component of the loading vector is -0.2608538. The loading score is a bit larger compared to the values of the other variables, so “concave.point_mean” most likely has a greater influence on PC1.

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

Considering all the proportion of variance covered by each principal component (shown in scree plots), the minimum number of principal components to explain 80% of the variance in the data is 5, since it has a cumulative proportion of 84.8%.

Hierarchical Clustering

The data must first be scaled.

```
data.scaled <- scale(wisc.data)
```

Then the distances between all pairs of observations must be computed.

```
data.dist <- dist(data.scaled)
```

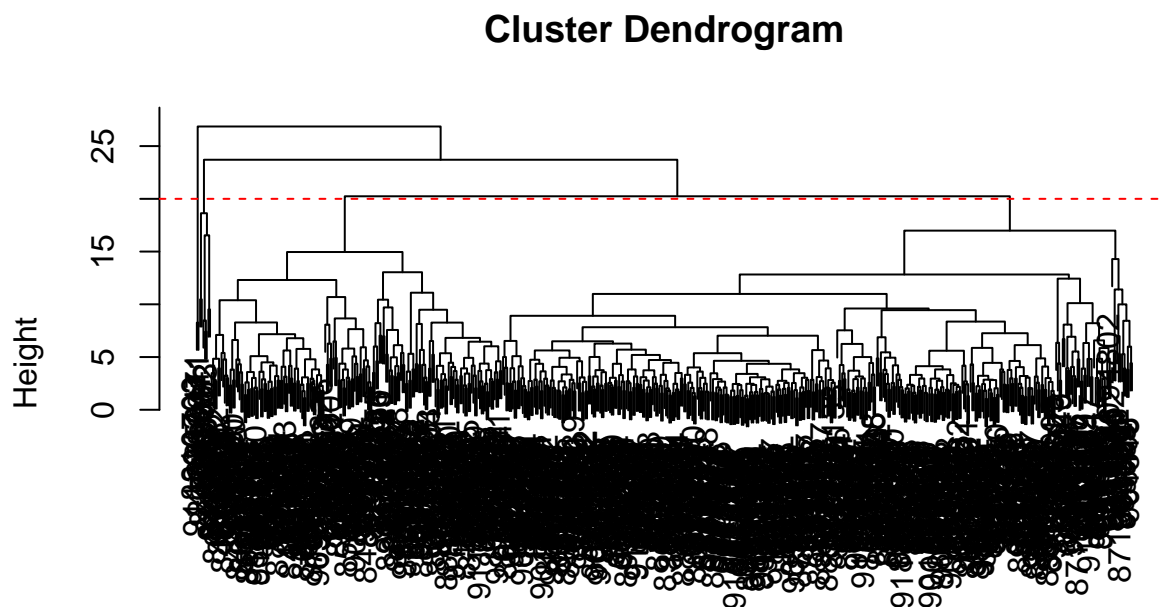
Now a hierarchical clustering model can be made (using complete linkage).

```
wisc.hclust <- hclust(data.dist, method = "complete")
```

Results of Hierarchical Clustering

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

```
plot(wisc.hclust)
abline(h = 20, col = "red", lty = 2)
```



data.dist
hclust (*, "complete")

The clustering model has four clusters at the height of 20.

Selecting Number of Clusters

Since we have the actual diagnoses, we can compare the output of the hierarchical clustering to determine the performance of the model with a certain number of clusters.

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

#compare results to diagnoses
table(wisc.hclust.clusters, diagnosis)
```

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

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

Having a smaller number of clusters makes it more difficult to determine which cluster matches with what diagnosis because the diagnoses are close to evenly split in just cluster 1. Having a greater number of clusters doesn't affect the overall conclusions drawn from having four clusters too much, though there are other clusters that could be associated with a particular diagnosis.

Using Different Methods

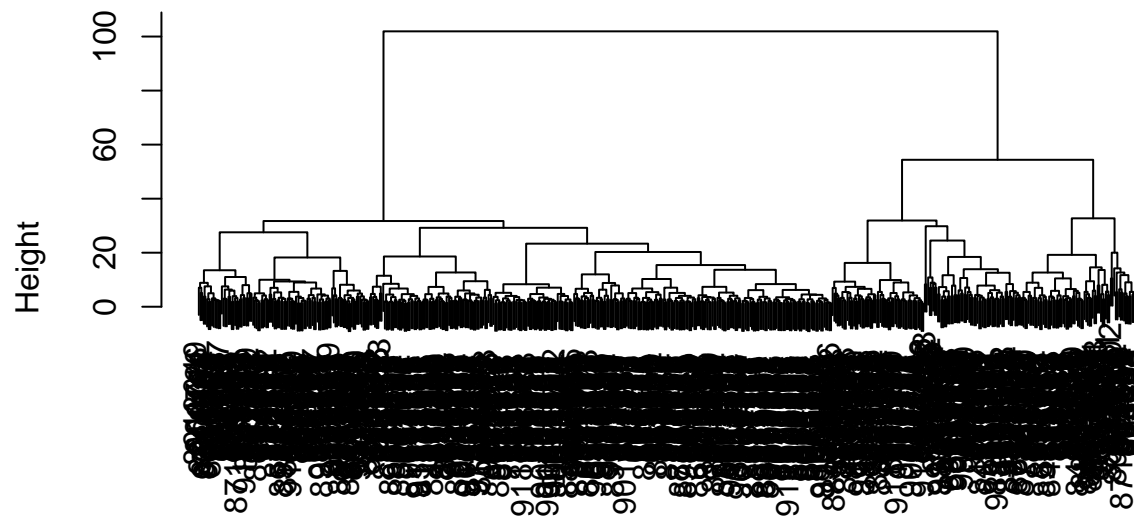
Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

The “ward.D2” method or “average” method are both preferable methods since they can minimize variance when creating clusters compared to using the “complete” or “single” method. For this specific dataset, “ward.D2” appears to show clearer clusters that may correspond to the two diagnoses whereas “average” is more difficult to analyze.

“One of the problems with Cluster Analysis is that different methods may produce different results – There is generally no universally accepted ‘best’ method”

```
#ward.D2 method
wisc.hclust.ward <- hclust(data.dist, method = "ward.D2")
plot(wisc.hclust.ward)
```

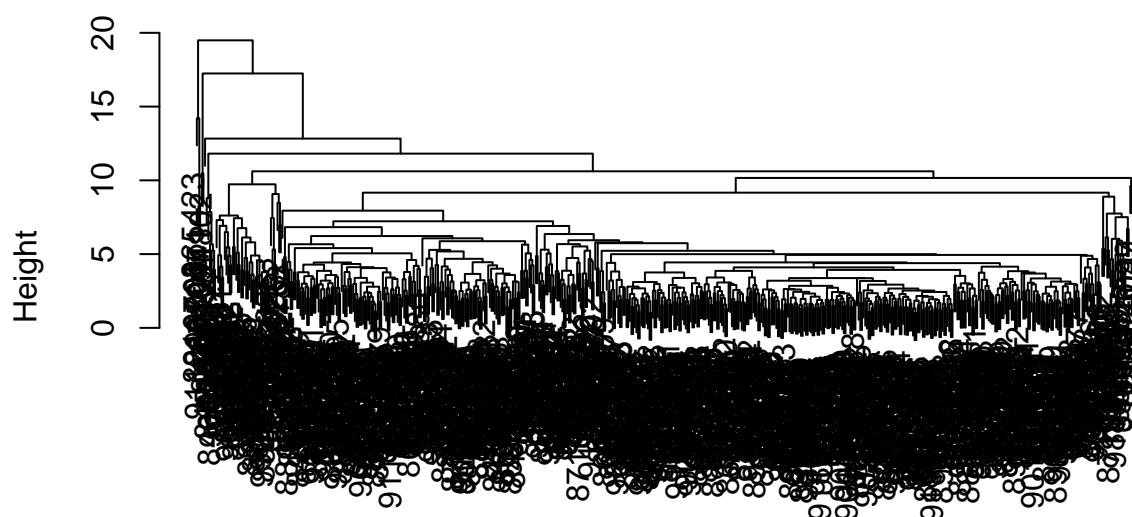
Cluster Dendrogram



```
data.dist  
hclust (*, "ward.D2")
```

```
#average method  
wisc.hclust.ward <- hclust(data.dist,method = "average")  
plot(wisc.hclust.ward)
```

Cluster Dendrogram



```
data.dist
hclust (*, "average")
```

OPTIONAL: K-Means Clustering

A k-means model for the scaled dataset was created with 2 clusters, with the algorithm repeated 20 times to find a well performing model.

```
wisc.km <- kmeans(data.scaled, centers = 2, nstart = 20)
```

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

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

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

```
#k-means
table(wisc.km$cluster, diagnosis)
```

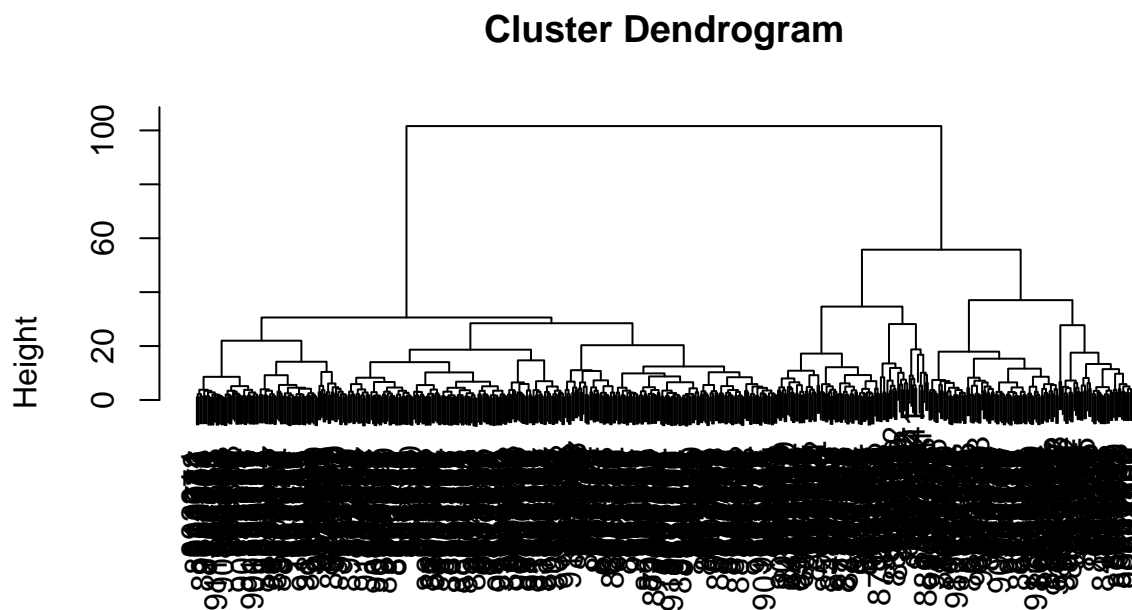
```
##      diagnosis
##      B      M
##    1 343   37
##    2  14  175
```

The k-means model separate the diagnoses more clearly compared to the hierarchical clustering since it uses 2 clusters to separate them, whereas the hierarchical clustering needs 4 clusters to demonstrate the same level of separation. As stated before, having less than 4 clusters leads to less separation of diagnoses in clusters.

Combining Methods

Combine the PCA model with hierarchical clustering!

```
wisc.pr.hclust <- hclust(dist(wisc.pr$x[,1:7]), method = "ward.D2")
plot(wisc.pr.hclust)
```



```
dist(wisc.pr$x[, 1:7])
hclust (*, "ward.D2")
```

There are two main clusters in this model, which may be associated with malignant and benign diagnoses. The table() function can be used to determine if they're related.

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

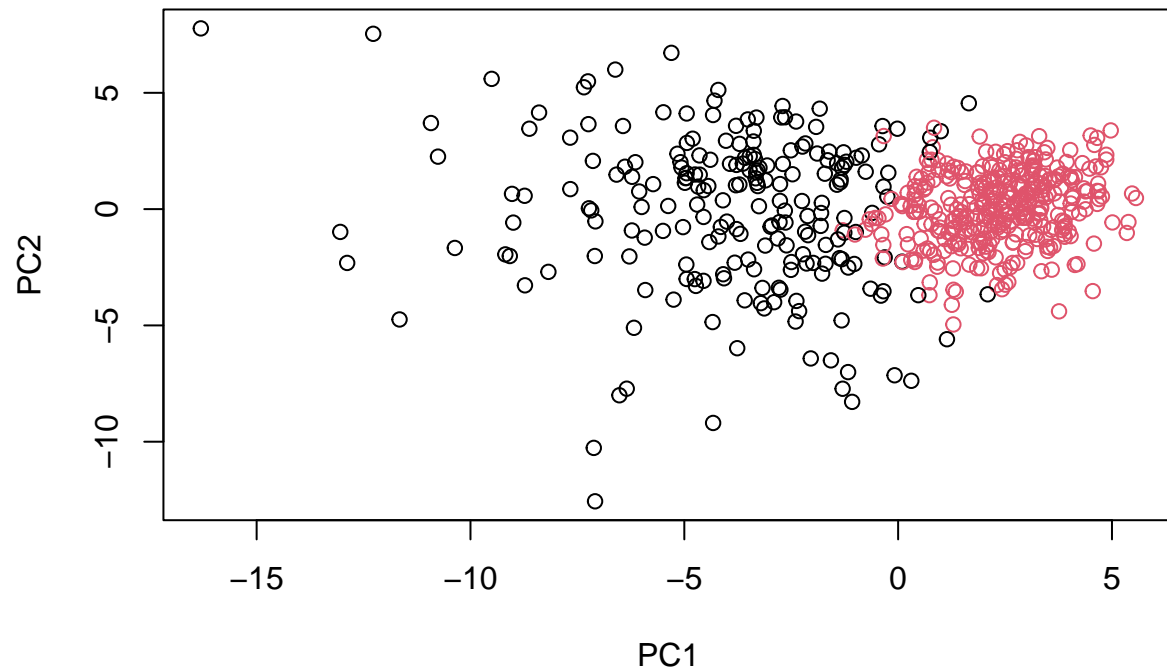
```
##      diagnosis
```



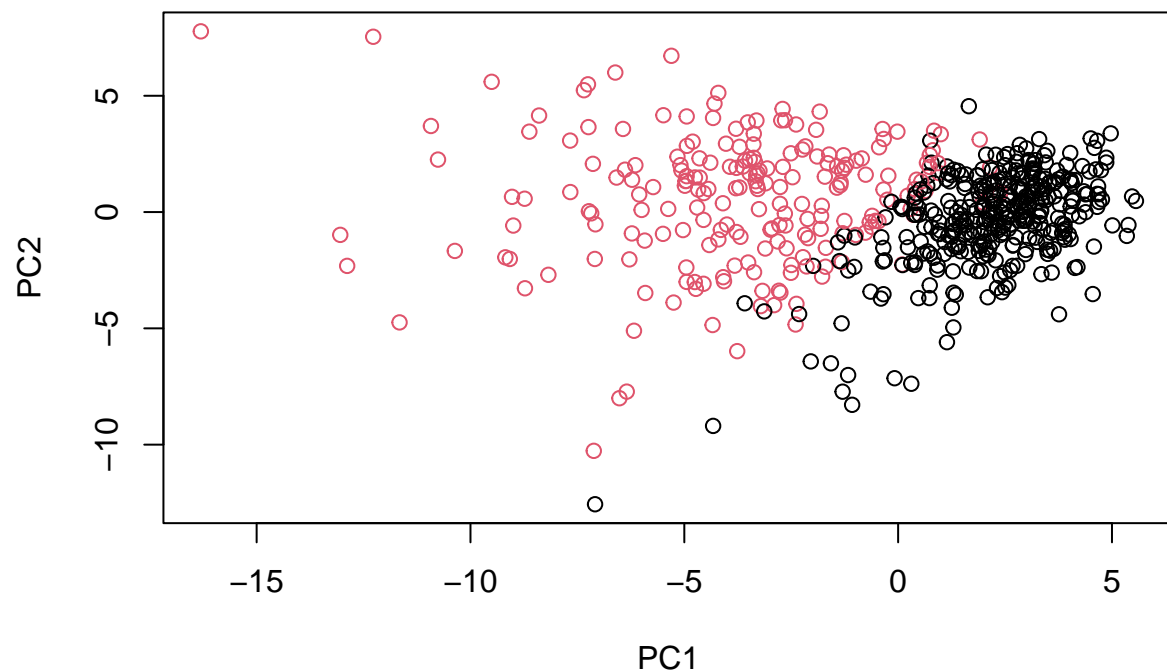
```
## grps    B    M
##      1  28 188
##      2 329  24
```

Use a scatterplot to visualize the separation!

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

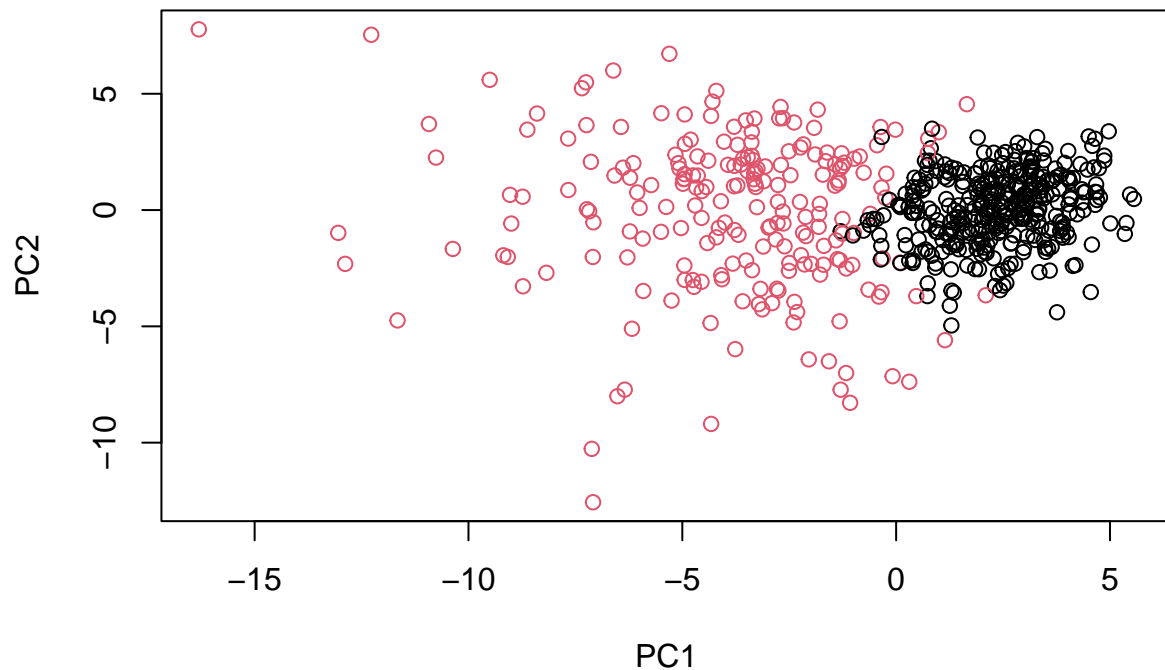


```
plot(wisc.pr$x[,1:2], col = diagnosis)
```



OPTIONAL: Reorder the levels so cluster 2 (Benign) will be colored black and cluster 1 (Malignant) will be colored red so that it matches the coloring of “col = diagnosis”.

```
#make grps into a factor  
g <- as.factor(grps)  
  
#reorder for color swap  
g <- relevel(g,2)  
  
#plot using re-ordered factor  
plot(wisc.pr$x[,1:2], col = g)
```



Now, compare the results of the new model with the actual diagnoses.

```
#cut into two clusters
wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k = 2)

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

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

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

The new model separates the two diagnoses clearly since there's significantly more benign diagnoses in cluster 2 and more malignant diagnoses in cluster 1.

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?

```
#k-means
table(wisc.km$cluster, diagnosis)
```

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

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

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

The new model appears to perform similarly to the k-means model in that they both clearly separate the two diagnoses to a similar extent. The same is true for hierarchical clustering as well, though since the hierarchical clustering uses 4 clusters, the separation is a little less clearly divided.

Sensitivity & Specificity

Sensitivity refers to a test's ability to correctly detect ill patients who do have the condition. In our example here the sensitivity is the total number of samples in the cluster identified as predominantly malignant (cancerous) divided by the total number of known malignant samples.

Specificity relates to a test's ability to correctly reject healthy patients without a condition. In our example specificity is the proportion of benign (not cancerous) samples in the cluster identified as predominantly benign that are known to be benign.

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

```
#k-means sensitivity
175/212
```

```
## [1] 0.8254717
```

```
#hierarchical sensitivity
165/212
```

```
## [1] 0.7783019
```

```
#combined sensitivity
188/212
```

```
## [1] 0.8867925
```

The PCA-hierarchical clustering model had the greatest sensitivity.

```
#k-means specificity
343/357
```

```
## [1] 0.9607843
```

```
#hierarchical specificity
343/357
```

```
## [1] 0.9607843
```

```
#combined specificity
329/357
```

```
## [1] 0.9215686
```

The k-means model had the greatest specificity.

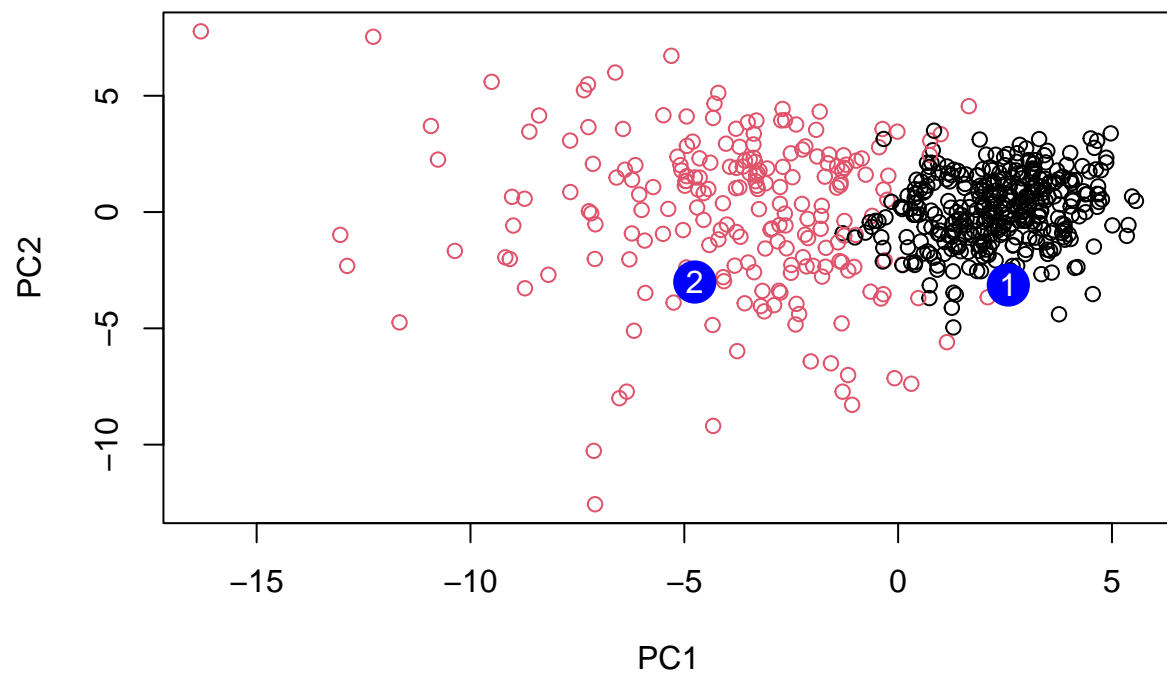
Prediction

Use the `predict()` function to project new cancel cell data on our current PCA model.

```
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")
```



Q18. Which of these new patients should we prioritize for follow up based on your results?
We should prioritize follow ups with patients from cluster 2, since cluster 2 was determined to be those predicted to have a malignant diagnosis with a low chance of it being a false positive.