

# Breast Cancer Lab

Ethan Lai

2/11/2022

## Importing Data

We import the data from “WisconsinCancer.csv” and perform some basic processing.

```
fna.data <- read.csv("WisconsinCancer.csv")
wisc.df <- data.frame(fna.data, row.names=1)
diagnosis <- wisc.df[1]
wisc.data <- wisc.df[,-1,-31]
```

Q1. How many observations are in this dataset? Q2. How many of the observations have a malignant diagnosis? Q3. How many variables/features in the data are suffixed with `_mean`?

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

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

```
length(diagnosis[diagnosis=="M"])
```

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

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

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

There are 569 observations 212 are malignant 10 column names contain `_mean`

#PCA

Check the mean and standard deviation of the features (i.e. columns) of the `wisc.data` to determine if the data should be scaled. Use the `colMeans()` and `apply()` functions like you’ve done before.

```
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
##          X
##          NA
```

```
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
##          X
##          NA
```

There is a fair amount of variability in column variance, so we should probably set scaling to true.

Perform PCA on dataset:

```
wisc.pr <- prcomp( x=wisc.data[-31], scale.=TRUE)
```

Let's examine the summary for this PCA:

```
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)?

0.442

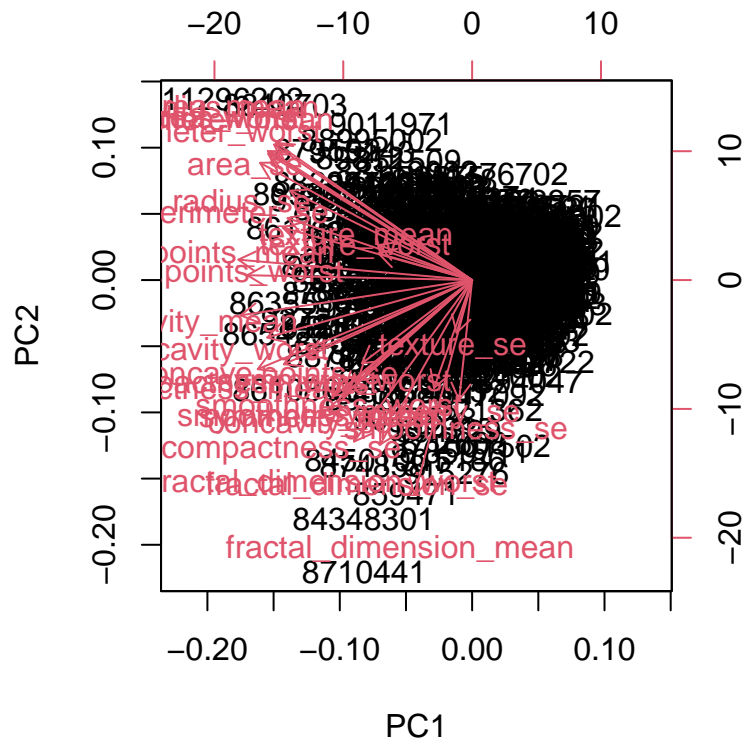
###Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data? PC1, 2, and 3 combined capture about 72.6% of the variance

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

7 PCs

## Now we Plot of PCA analysis as a biplot

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

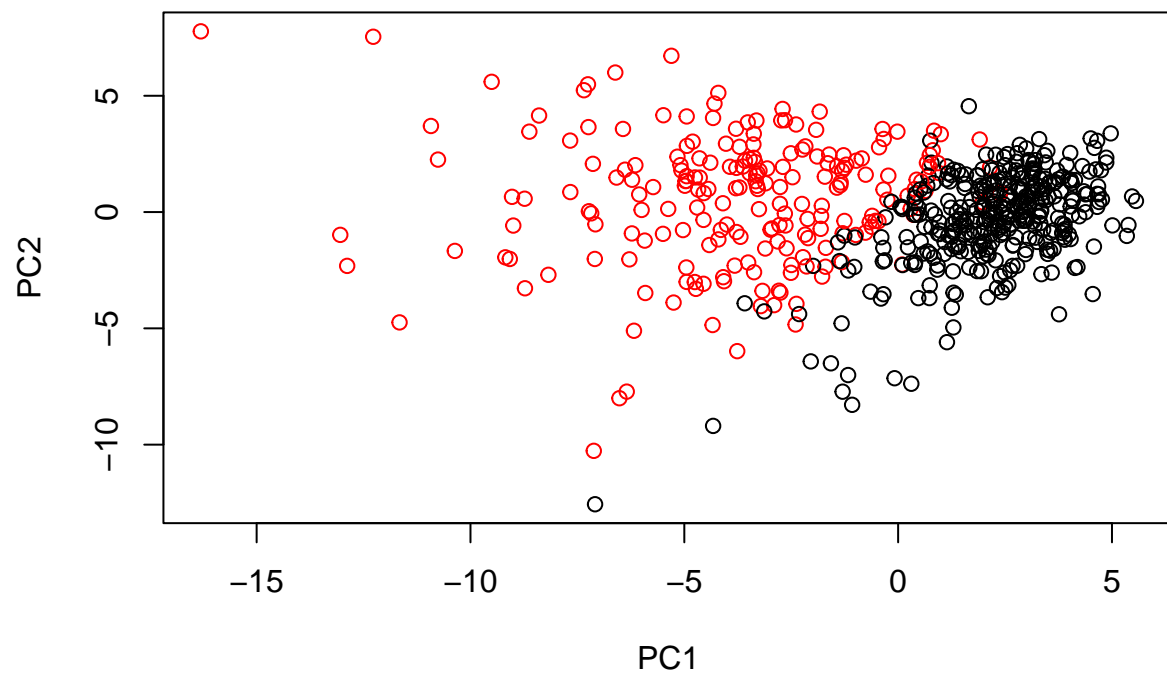


###Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

Default plot is extremely dense and chaotic!

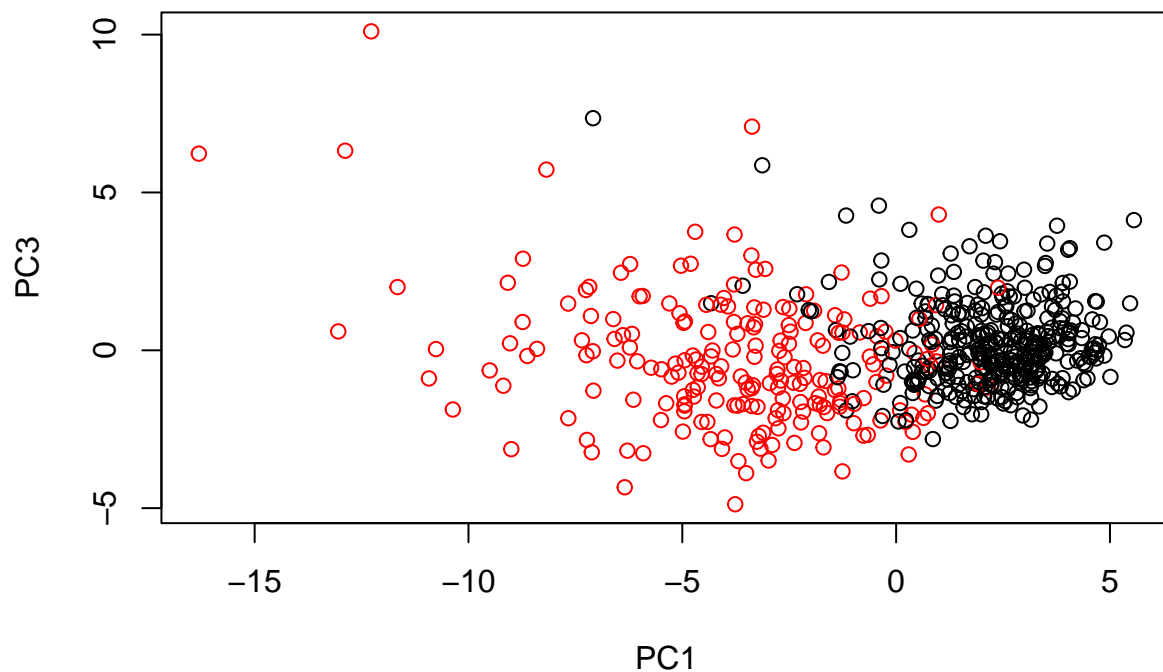
#Let's use base R to generate a scatter plot of PC1 vs PC2 instead.

```
# Scatter plot observations by components 1 and 2
plot( wisc.pr$x[,1], wisc.pr$x[,2], col = ifelse(diagnosis == "M", 'red', 'black'),
      xlab = "PC1", ylab = "PC2")
```



Now we generate a similar plot for PC1 and PC3:

```
# Scatter plot observations by components 1 and 2
plot( wisc.pr$x[,1], wisc.pr$x[,3], col = ifelse(diagnosis == "M", 'red', 'black'),
      xlab = "PC1", ylab = "PC3")
```



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

First plot has better separation of groups, which makes sense as PC2 explains more variance than PC3.

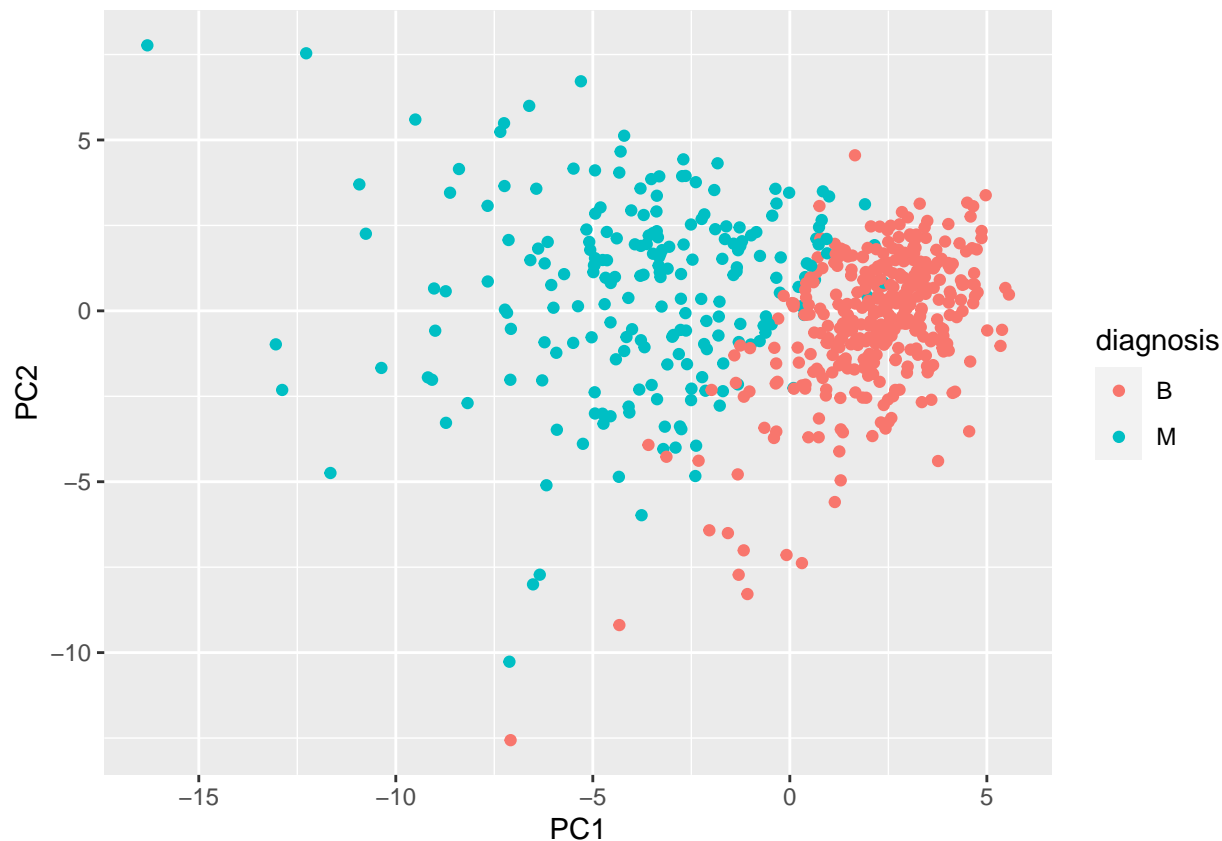
Overall, the plots indicate that principal component 1 is capturing a separation of malignant (red) from benign (black) samples!

Let's plot these results in ggplot2

```
# 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
df$diagnosis <- unlist(df$diagnosis)
ggplot(df, aes(PC1, PC2, col=diagnosis)) + geom_point()
```



## Variance explained

In this exercise, you will produce scree plots showing the proportion of variance explained as the number of principal components increases. The data from PCA must be prepared for these plots, as there is not a built-in function in base R to create them directly from the PCA model.

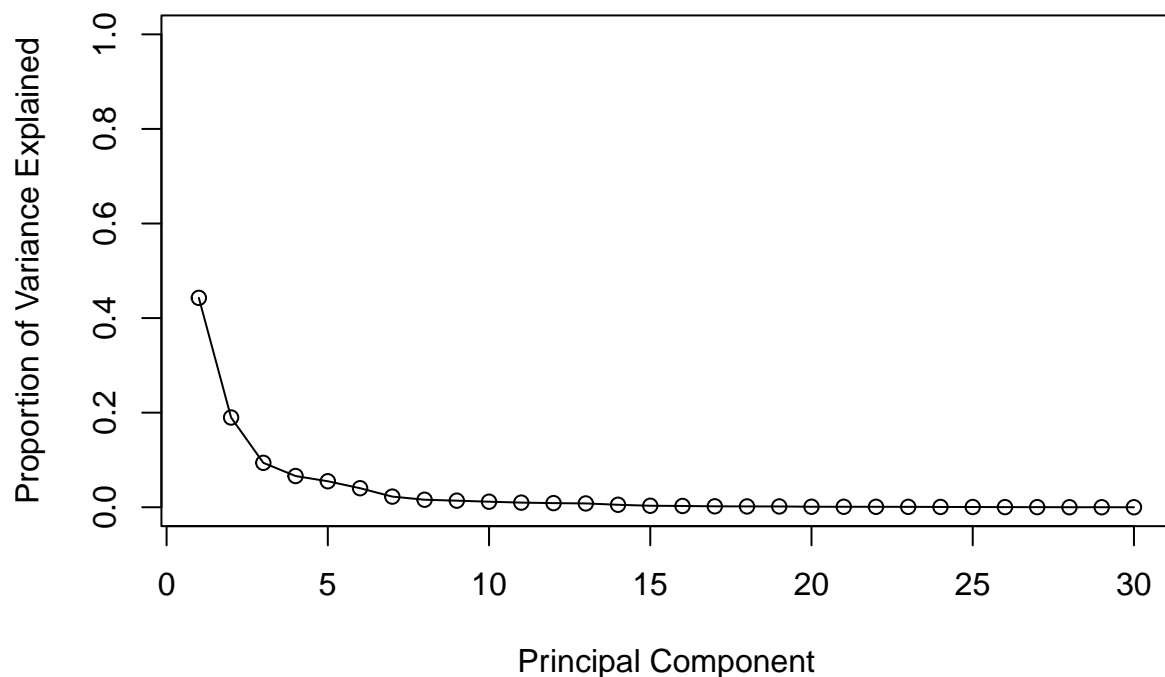
```
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

Calculate the variance explained by each principal component by dividing by the total variance explained of all principal components. Assign this to a variable called pve and create a plot of variance explained for each principal component.

```
# 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")
```



#### #Communicating PCA results

In this section we will check your understanding of the PCA results, in particular the loadings and variance explained. The loadings, represented as vectors, explain the mapping from the original features to the principal components. The principal components are naturally ordered from the most variance explained to the least variance explained.

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][8]
```

```
## concave.points_mean
## -0.2608538
```

```
-.26085
```

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

```
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
```

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

## Hierarchical clustering

The goal of this section is to do hierarchical clustering of the original data.

First scale the `wisc.data` data and assign the result to `data.scaled`.

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

Calculate the (Euclidean) distances between all pairs of observations in the new scaled dataset and assign the result to `data.dist`.

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

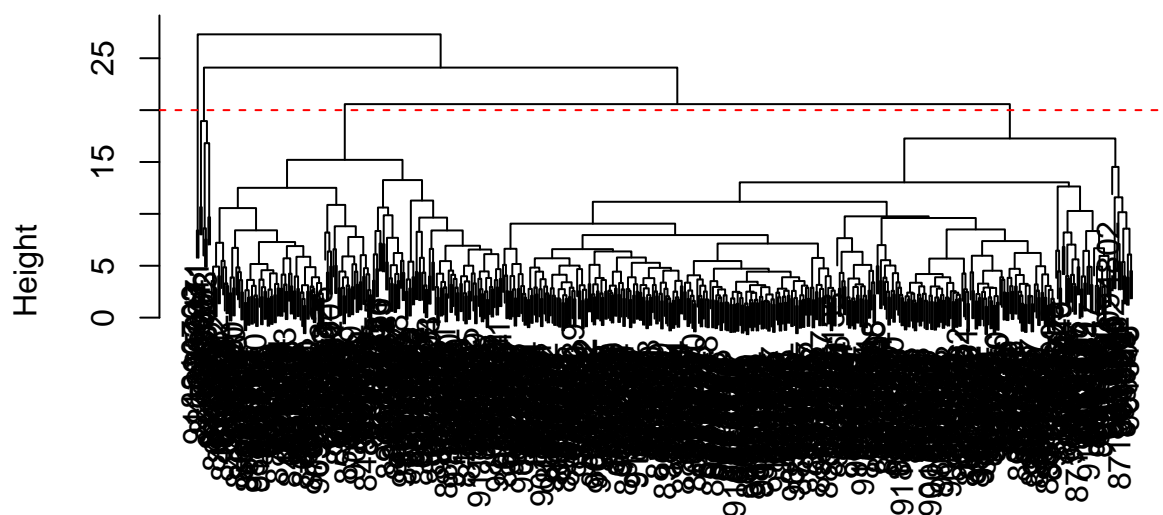
Create a hierarchical clustering model using complete linkage. Manually specify the `method` argument to `hclust()` and assign the results to `wisc.hclust`.

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

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)
```

## Cluster Dendrogram



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

Roughly height = 20.

#Selecting number of clusters In this section, you will compare the outputs from your hierarchical clustering model to the actual diagnoses.

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

We can use the table() function to compare the cluster membership to the actual diagnoses.

```
table(wisc.hclust.clusters, unlist(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?

```
# Clunky but not sure how for looks interface with R markdown?
table(cutree(wisc.hclust,k=2), unlist(diagnosis))
```

```
##
```

```
##      B  M
##  1 357 210
##  2   0   2
```

```
table(cutree(wisc.hclust,k=3), unlist(diagnosis))
```

```
##
##      B  M
##  1 355 205
##  2   2   5
##  3   0   2
```

```
table(cutree(wisc.hclust,k=4), unlist(diagnosis))
```

```
##
##      B  M
##  1  12 165
##  2   2   5
##  3 343  40
##  4   0   2
```

```
table(cutree(wisc.hclust,k=5), unlist(diagnosis))
```

```
##
##      B  M
##  1  12 165
##  2   0   5
##  3 343  40
##  4   2   0
##  5   0   2
```

```
table(cutree(wisc.hclust,k=6), unlist(diagnosis))
```

```
##
##      B  M
##  1  12 165
##  2   0   5
##  3 331  39
##  4   2   0
##  5  12   1
##  6   0   2
```

```
table(cutree(wisc.hclust,k=7), unlist(diagnosis))
```

```
##
##      B  M
##  1  12 165
##  2   0   3
##  3 331  39
##  4   2   0
##  5  12   1
##  6   0   2
##  7   0   2
```

```
table(cutree(wisc.hclust,k=8), unlist(diagnosis))
```

```
##
##      B  M
##  1 12 86
##  2  0 79
##  3  0  3
##  4 331 39
##  5  2  0
##  6 12  1
##  7  0  2
##  8  0  2
```

```
table(cutree(wisc.hclust,k=9), unlist(diagnosis))
```

```
##
##      B  M
##  1 12 86
##  2  0 79
##  3  0  3
##  4 331 39
##  5  2  0
##  6 12  0
##  7  0  2
##  8  0  2
##  9  0  1
```

```
table(cutree(wisc.hclust,k=10), unlist(diagnosis))
```

```
##
##      B  M
##  1 12 86
##  2  0 59
##  3  0  3
##  4 331 39
##  5  0 20
##  6  2  0
##  7 12  0
##  8  0  2
##  9  0  2
## 10  0  1
```

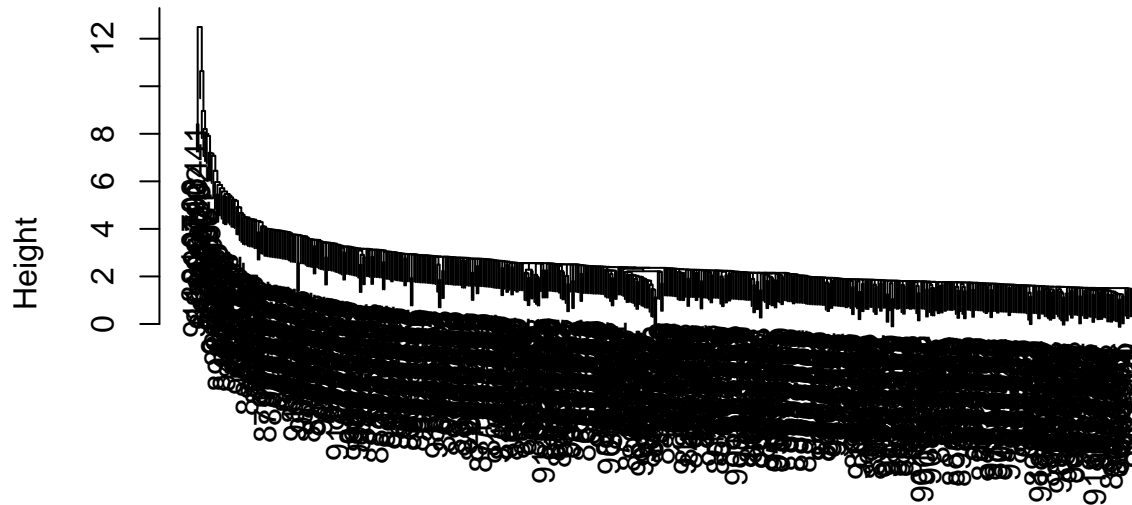
k=4 does seem to be one of the better scenarios, though k=3 and k=5 aren't bad either.

As we discussed in our last class videos there are number of different “methods” we can use to combine points during the hierarchical clustering procedure. These include “single”, “complete”, “average” and (my favorite) “ward.D2”.

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

```
wisc.hclust2 <- hclust(data.dist, "single")  
plot(wisc.hclust2)
```

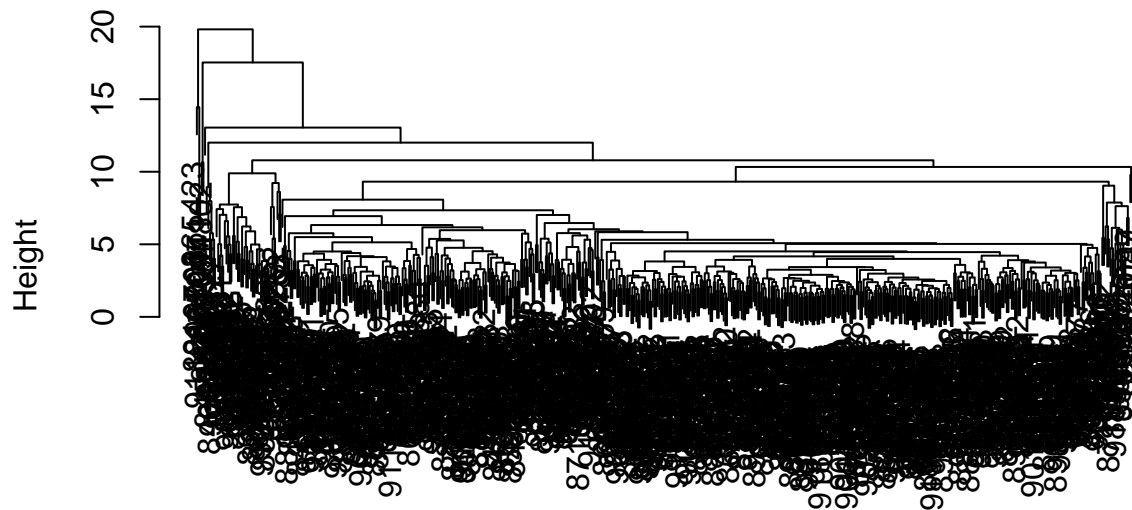
## Cluster Dendrogram



data.dist  
hclust (\*, "single")

```
wisc.hclust3 <- hclust(data.dist, "average")  
plot(wisc.hclust3)
```

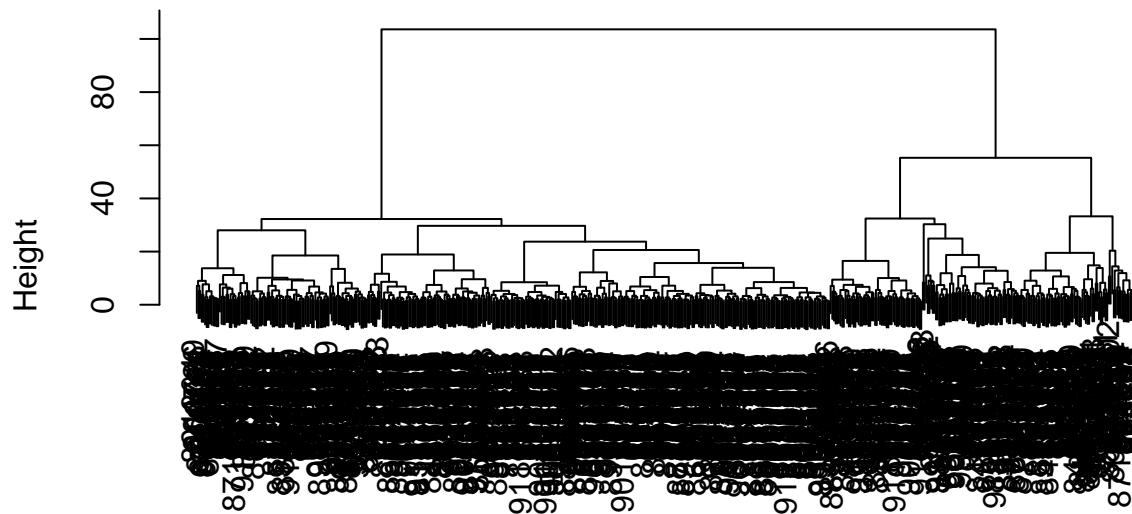
## Cluster Dendrogram



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

```
wisc.hclust4 <- hclust(data.dist, "ward.D2")  
plot(wisc.hclust4)
```

## Cluster Dendrogram



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

ward.D2 produces two very clean main branches in the dendrogram!

Let's see how it corresponds to diagnosis

```
wisc.hclust4.clusters <- cutree(wisc.hclust4,k=2)
table(wisc.hclust.clusters, unlist(diagnosis))
```

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

Pretty good clustering! ward.D2 seems to cluster with very high correlation to actual diagnosis.

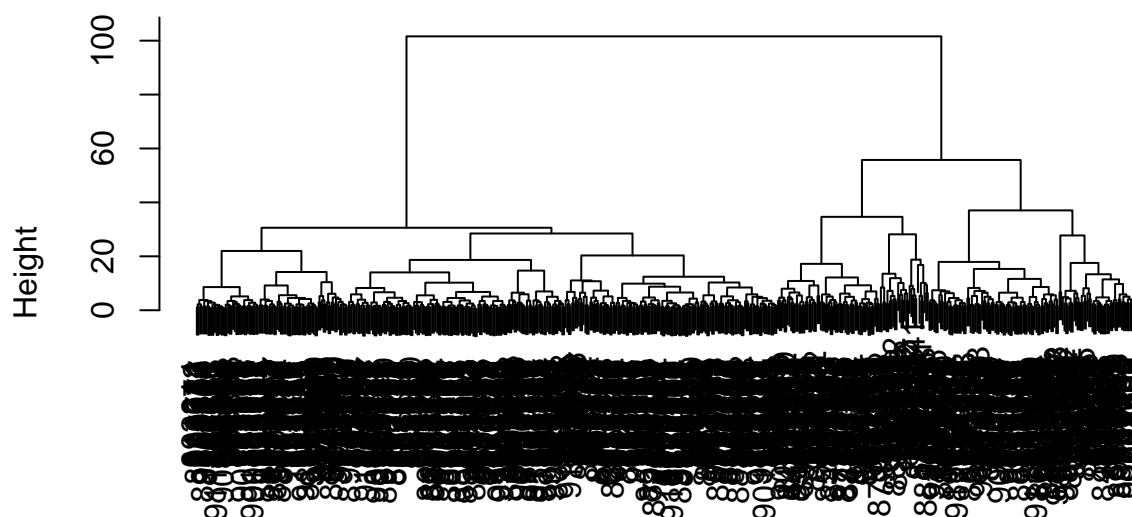
#5. Combining methods

##Clustering on PCA results

Using the minimum number of principal components required to describe at least 90% of the variability in the data, create a hierarchical clustering model with the linkage method="ward.D2". We use Ward's criterion here because it is based on multidimensional variance like principal components analysis. Assign the results to wisc.pr.hclust.

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

## Cluster Dendrogram



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

Do these two major branches correspond to malignant vs benign?

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

```
## grps
## 1 2
## 216 353
```

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

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

Let's try plotting this data:

Mapping color to grps:

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

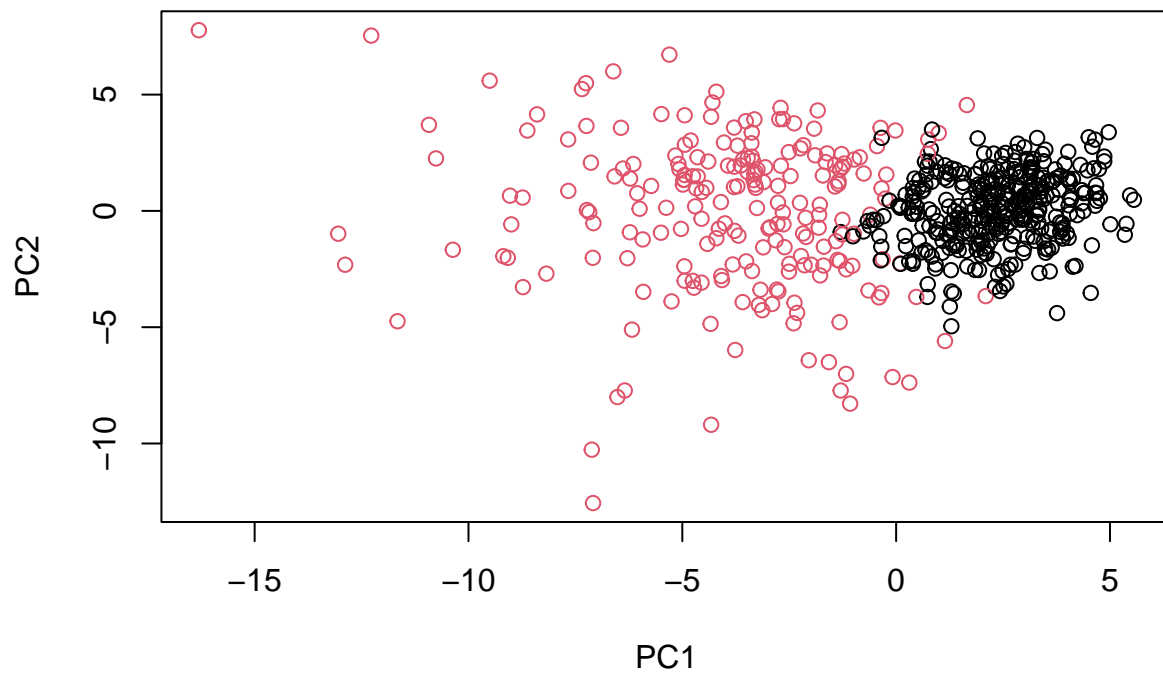
```
## [1] "1" "2"
```



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

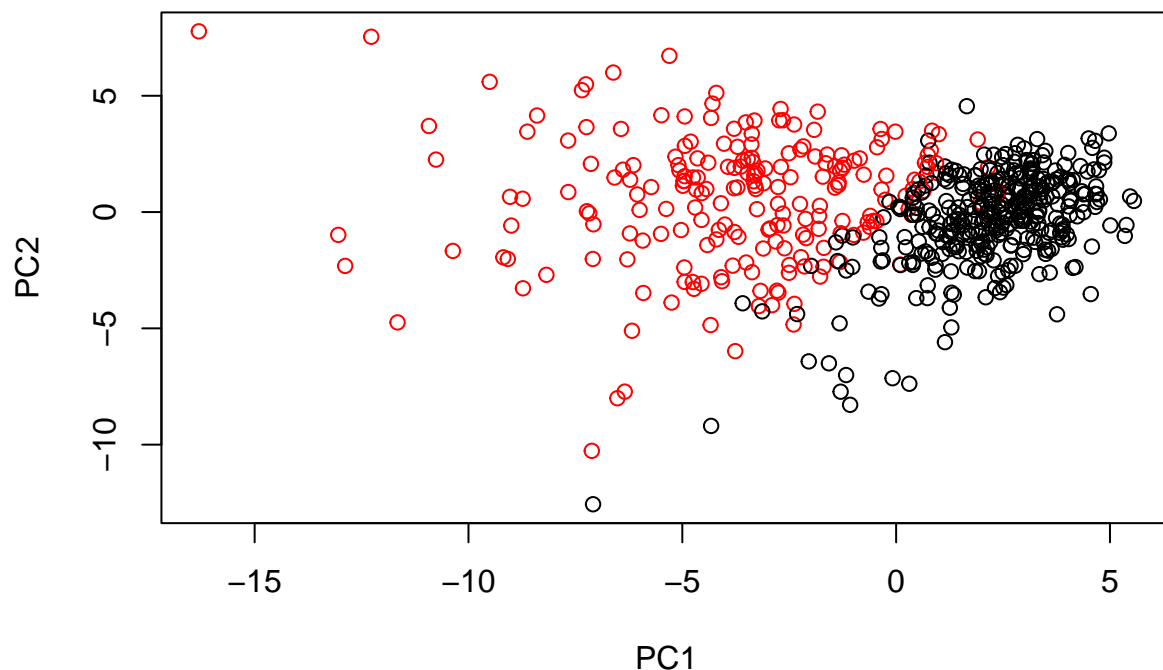
```
## [1] "2" "1"
```

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



And to actual diagnosis:

```
plot(wisc.pr$x[,1:2], col = ifelse(diagnosis == "M", 'red', 'black'))
```



It seems that this new clustering/PCA approach quite accurately predicts actual diagnosis!

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

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.

```
# New model
wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k=2)
table(wisc.pr.hclust.clusters, unlist(diagnosis))
```

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

```
# old hierarchical clustering model
table(wisc.hclust.clusters, unlist(diagnosis))
```

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

The new model does quite a bit better than the old hierarchical clustering model!

#### #6. 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. In other words:  $TP/(TP+FN)$ .

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. In other words:  $TN/(TN+FP)$ .

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

We'll just manually calculate these scores for the sake of brevity

```
# New model sensitivity
188/(188+24)
```

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

```
# New model specificity
329/(329+28)
```

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

```
#Old model sensitivity
165/(165+5+40+2)
```

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

```
#Old model specificity
343/(343+14)
```

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

So the new model is more sensitive, but the old is actually marginally more specific.

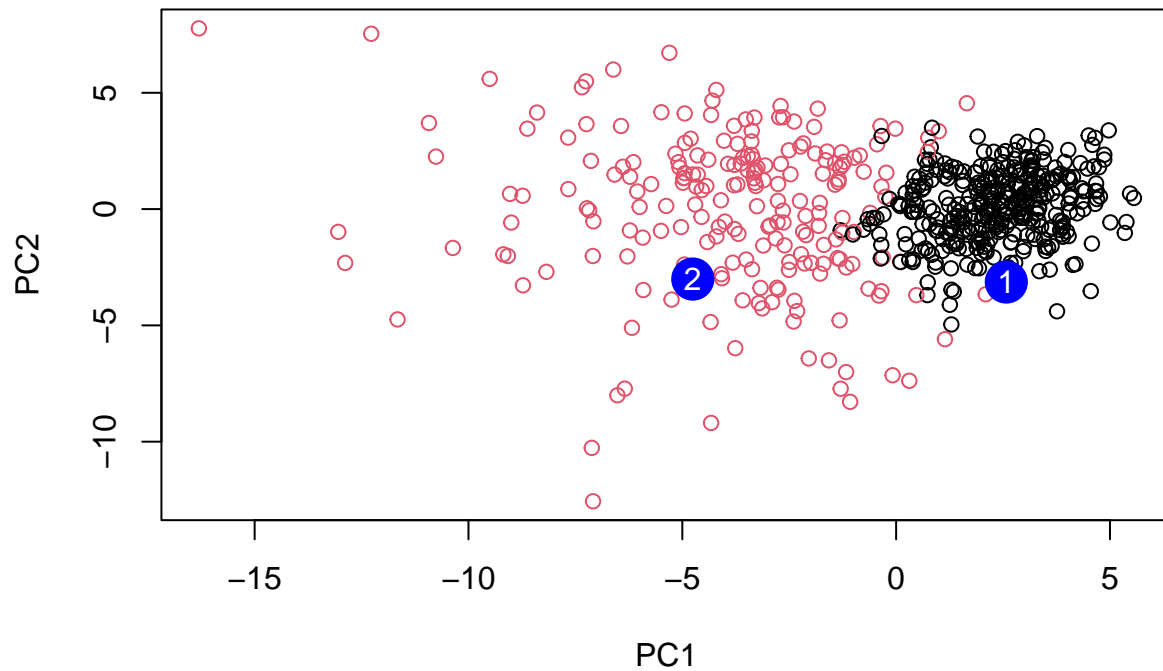
#7. Prediction We will use the `predict()` function that will take our PCA model from before and new cancer cell data and project that data onto our PCA space.

```
#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")
```



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

Patient 2- their biopsy falls into the malignant cluster.