miniprojectR

Rachel Kraft

2/13/2022

Download and import data set

```
fna.data <- "C:/Users/Rachel/Desktop/WisconsinCancer.csv"
wisc.df <- read.csv(fna.data, row.names=1)
```

Remove diagnosis column

```
wisc.data <- wisc.df[,-1]
```

Vector for info from diagnosis column

Make a factor for diagnosis to assign the possible levels to the diagnosis column

```
diagnosis_levels <- c("B", "M")
diagnosis <- factor(c(wisc.df[,1]), levels=diagnosis_levels)
```

Q1) How many observations are in this dataset?

```
dim(wisc.data)
```

```
## [1] 569  30
```

Q2) How many of the observations have malignant diagnosis?

```
## grep() search for character match in a character vector
length(grep("M", diagnosis))
```

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

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

- look for which columns have mean in them

```
length(grep("mean", colnames(wisc.data)))
```

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

PCA

- check mean and stdev to see if it needs to be scaled

```
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=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)?

When you look at the PCA results above, the proportion of variance for PC1 is 0.4427

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

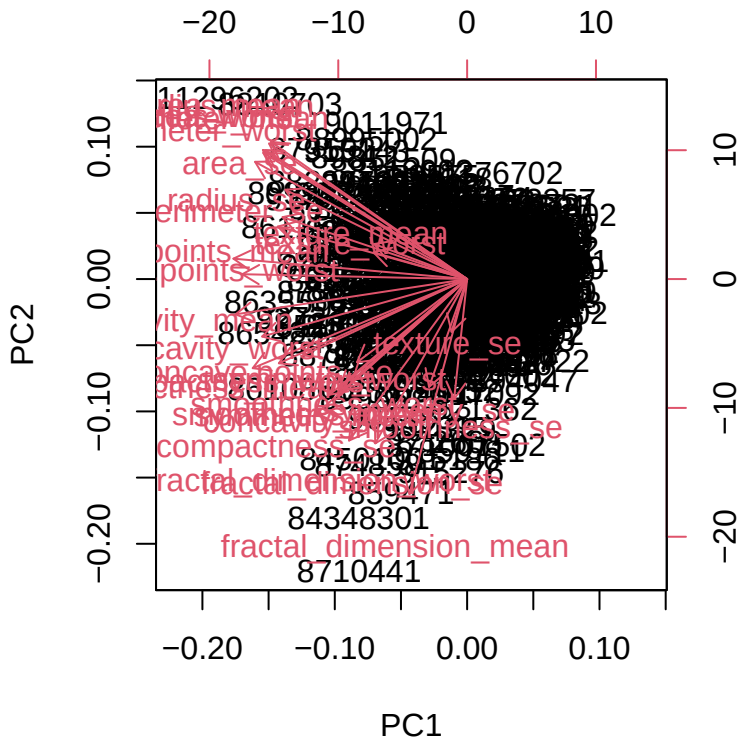
To find this, look at the cumulative proportion row for each PC column above. The first one over 0.7 is PC3, so 3 principal components are required to describe at least 70% of the original variance

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

Cumulative proportion is at least 0.9 at PC7, so 7 principal components are required to describe at least 90% of original variance.

> Make a biplot for our PCA results

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

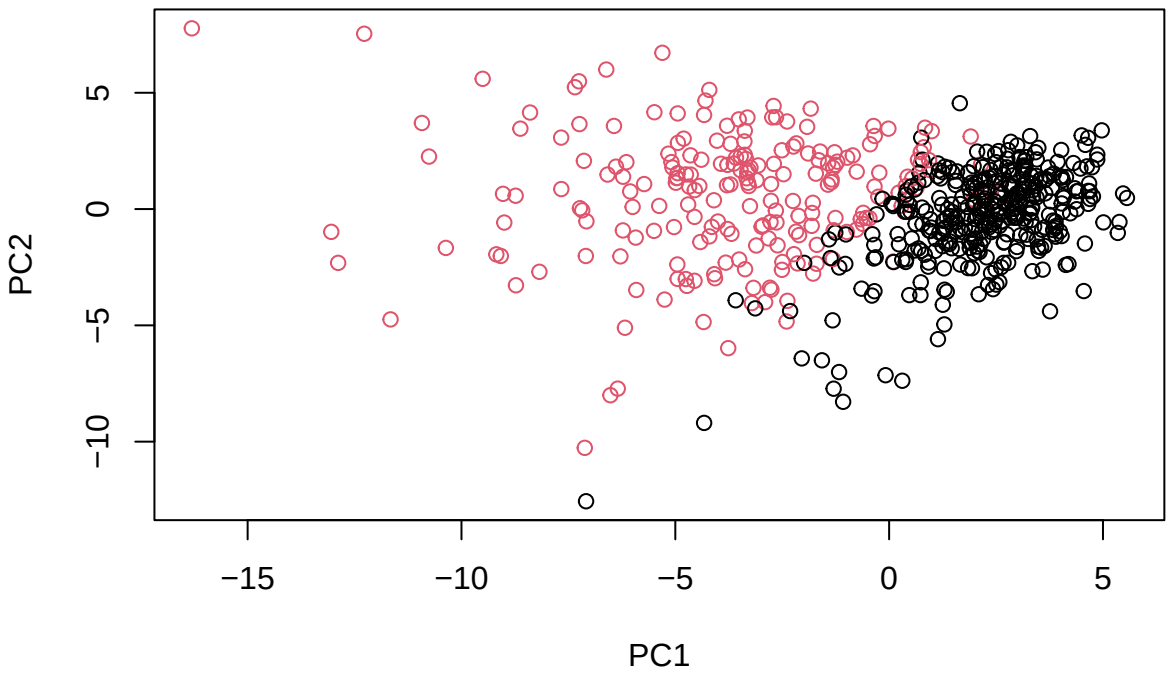


Q7) What stands out to you about this plot? Is it easy or difficult to understand? Why?

It is too cluttered and chaotic, very difficult to understand.

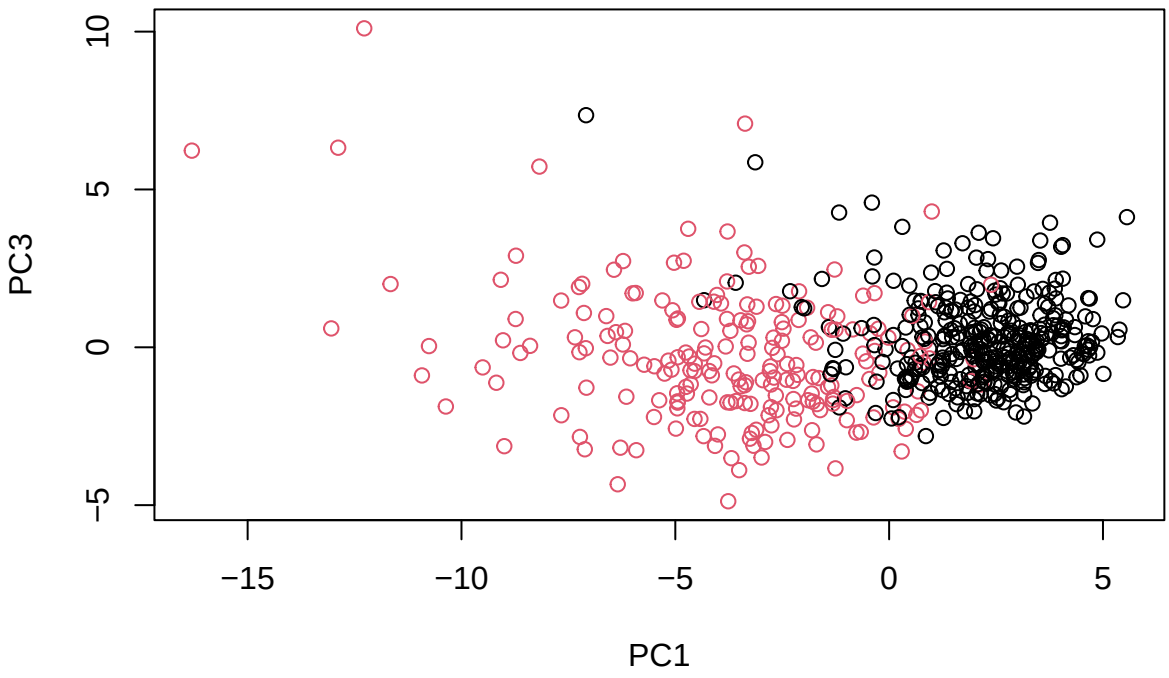
Make scatter plot for each observation along PC1 and 2

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



Use ggplot2 for a better figure

```
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis

library(ggplot2)

ggplot(df)+aes(PC1, PC2, col=df$diagnosis) + geom_point()
```

```
## Warning: Use of ‘df$diagnosis‘ is discouraged. Use ‘diagnosis‘ instead.
```

Calculate variance of each PC by squaring sdev

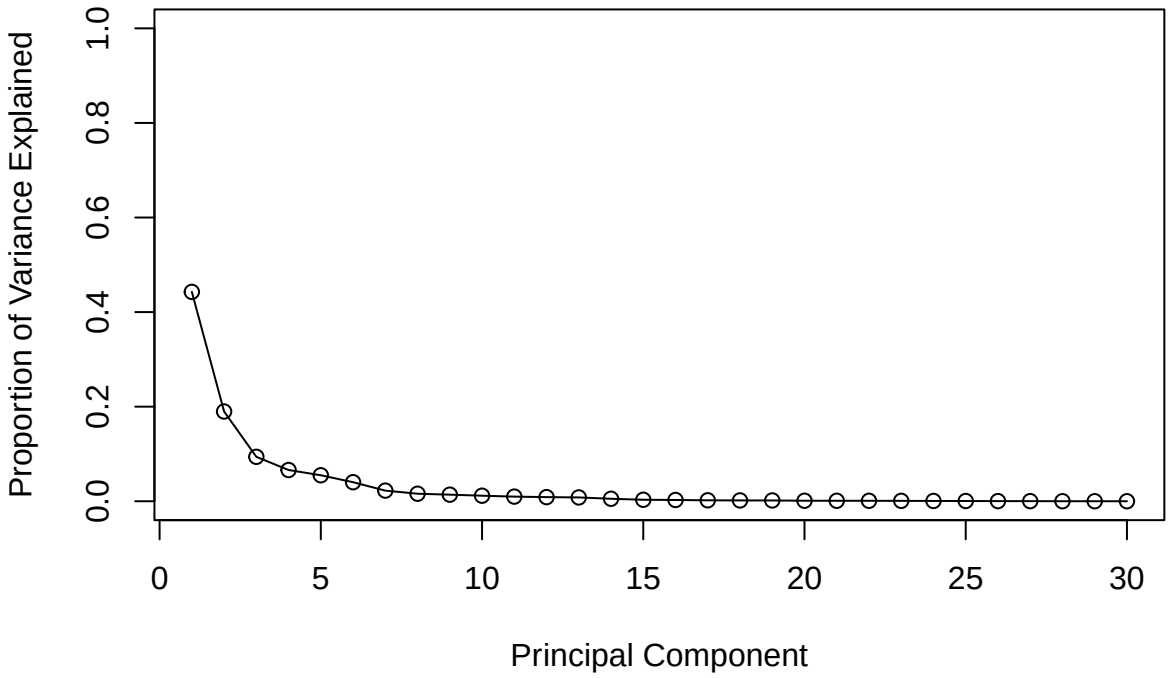
```
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

Calculate variance explained by each PC and plot

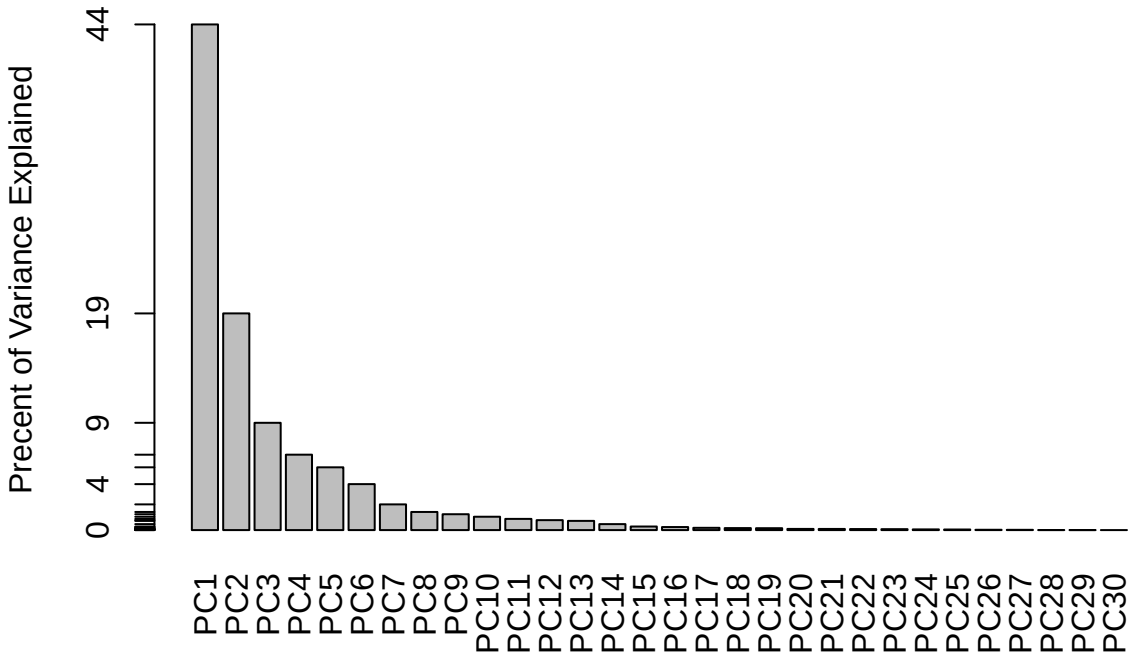
```
pve <- pr.var/sum(pr.var)

plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "o")
```



Alternative plot

```
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
```

component for concave.points_mean is -0.26085376

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

5 principal components are required to explain 80% of variance of the data according to the PCA analysis

Hierarchical clustering

- scale the data

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

- calculate distances

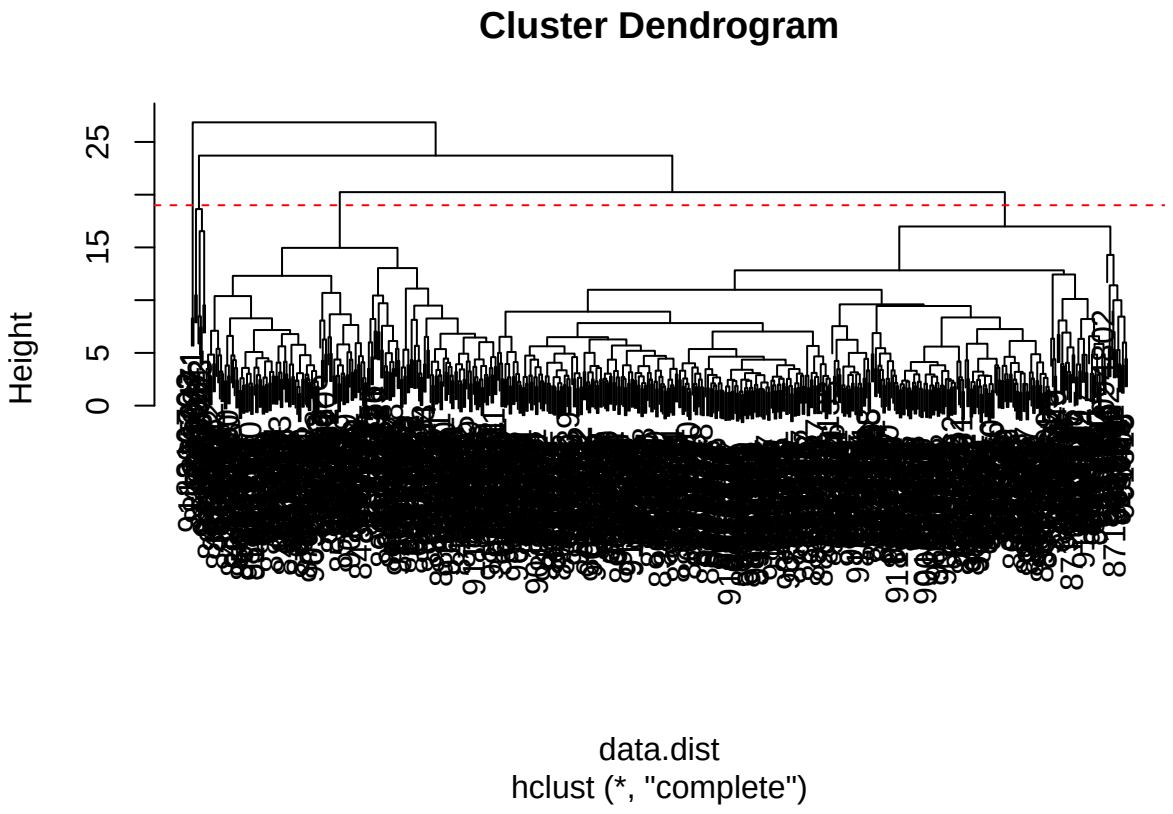
```
data.dist <- dist(data.scaled)
```

- create clustering model

```
wisc.hclust <- hclust(data.dist, method="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=19, col="red", lty=2)
```



- cut into 4 clusters

```
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
```

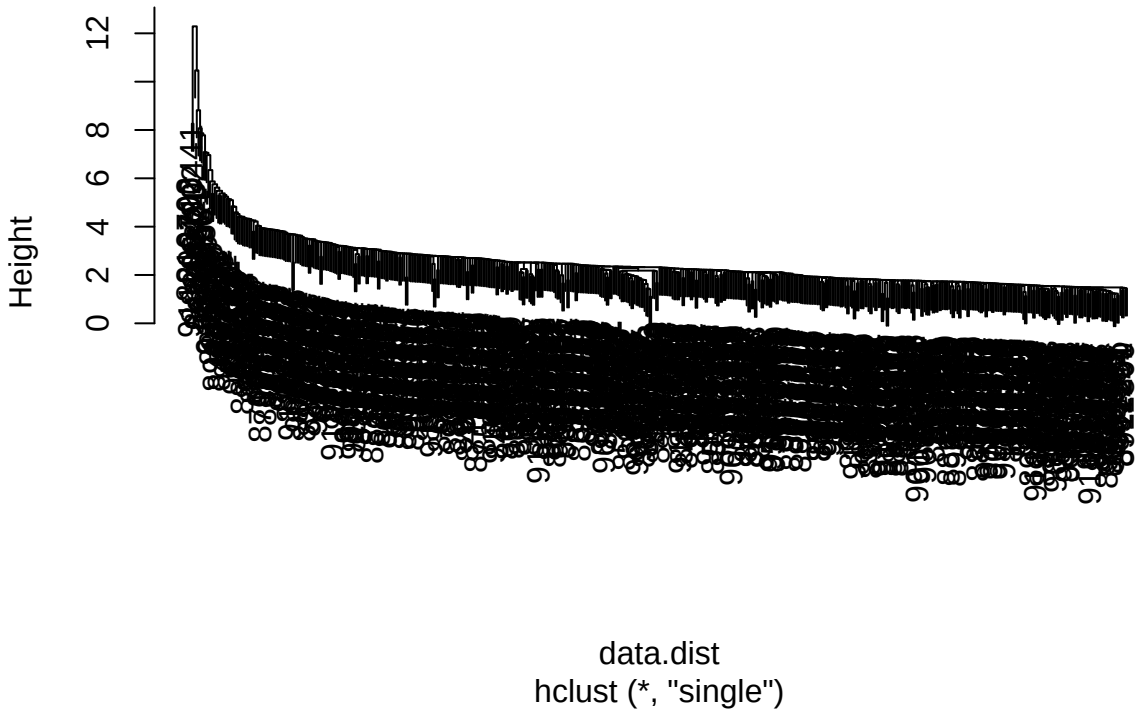
Q12) Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10?

```
for (i in c(2:10)) {
  wisc.hclust.clusters <- cutree(wisc.hclust, k=i)
  table(wisc.hclust.clusters, diagnosis)
}
wisc.hclust.clusters <- cutree(wisc.hclust, k=4)
```

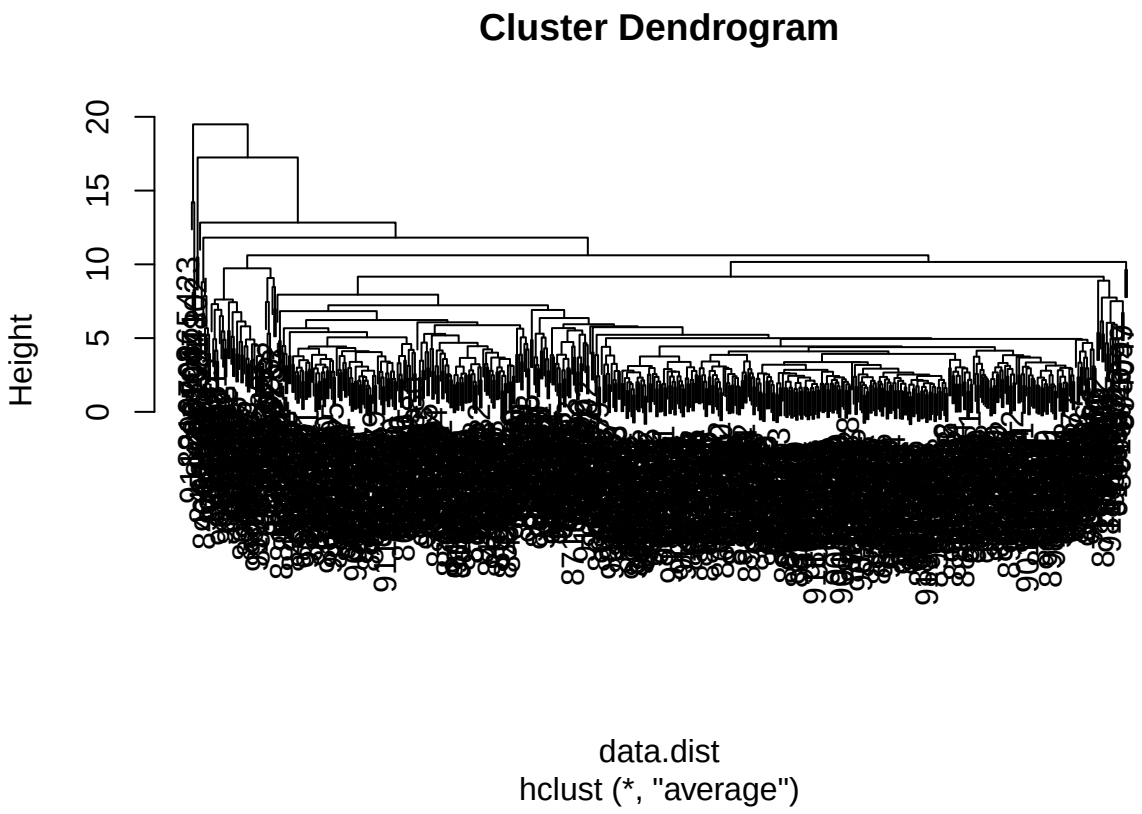
Cutting into 5 clusters may be a good match too

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

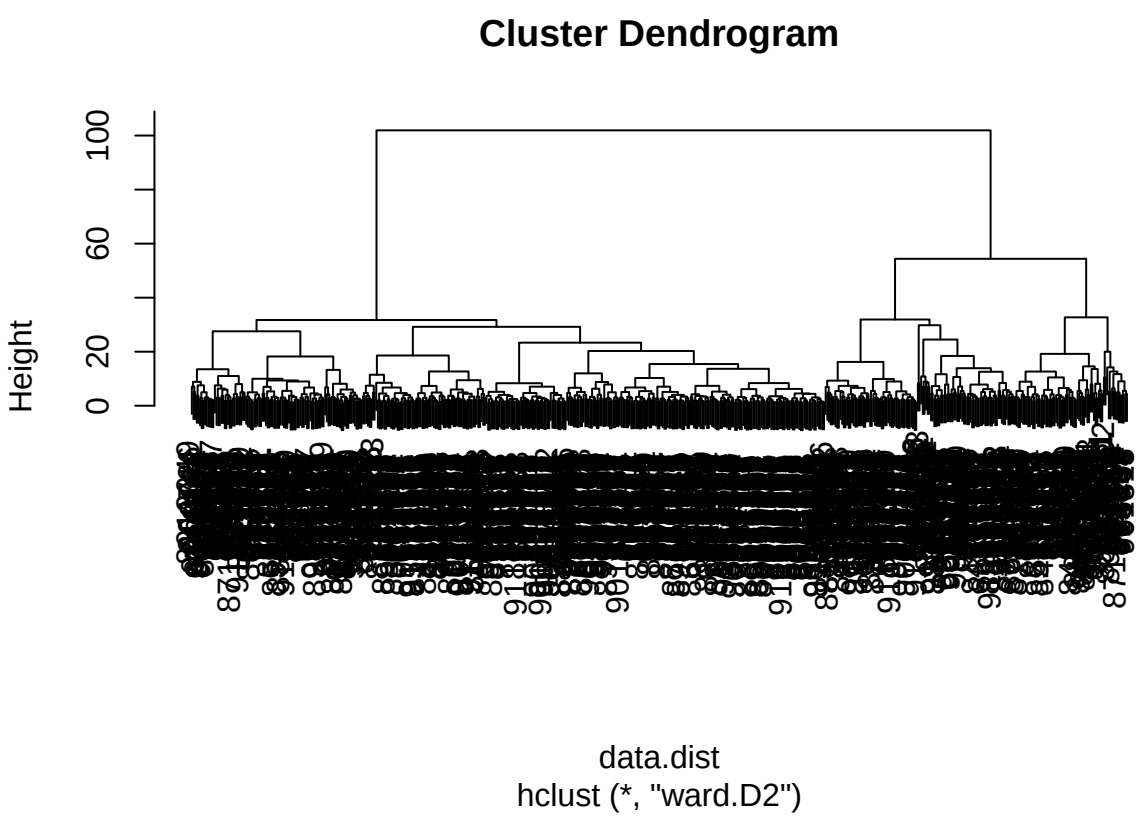
```
wisc.hclust <- hclust(data.dist, method="single")
plot(wisc.hclust)
```




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



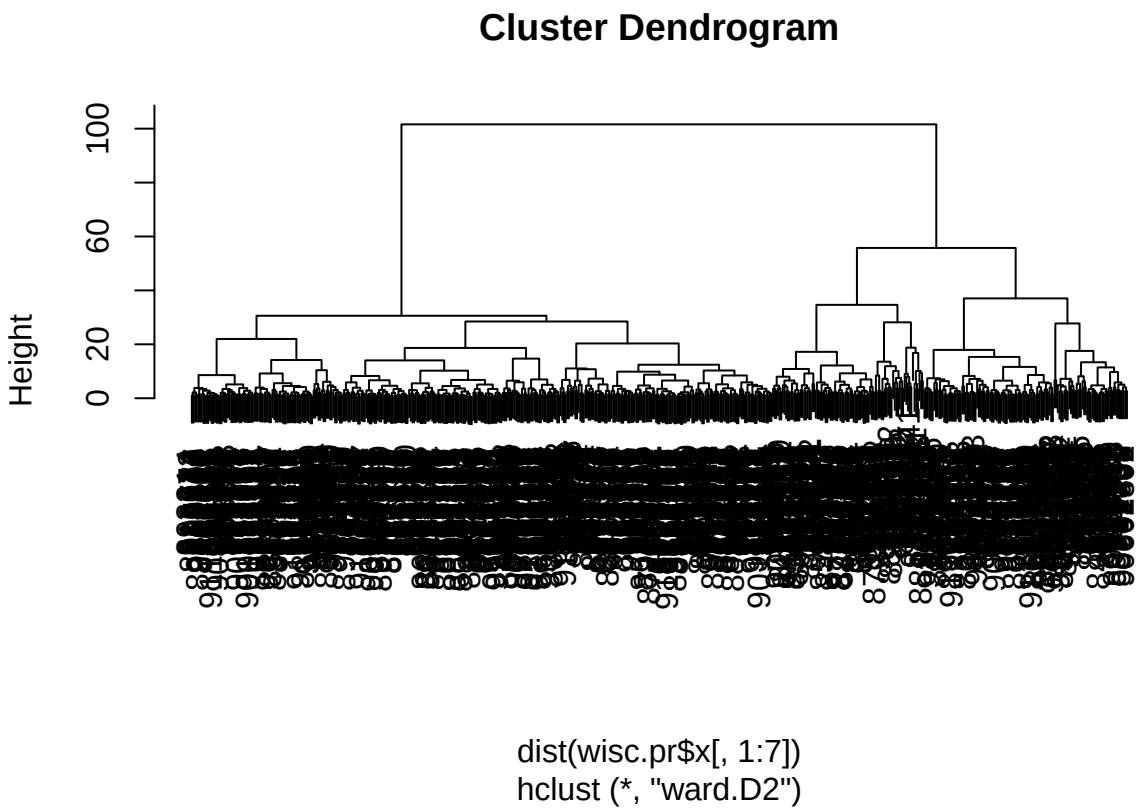

```
wisc.hclust <- hclust(data.dist, method="ward.D2")
plot(wisc.hclust)
```



I like the results from method="Ward.D2" because it looks the cleanest and allows me to see the clusters more clearly.

Use min PA to describe at least 90% of variability and method="ward.D2"

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



Are these 2 clusters malignant and benign?

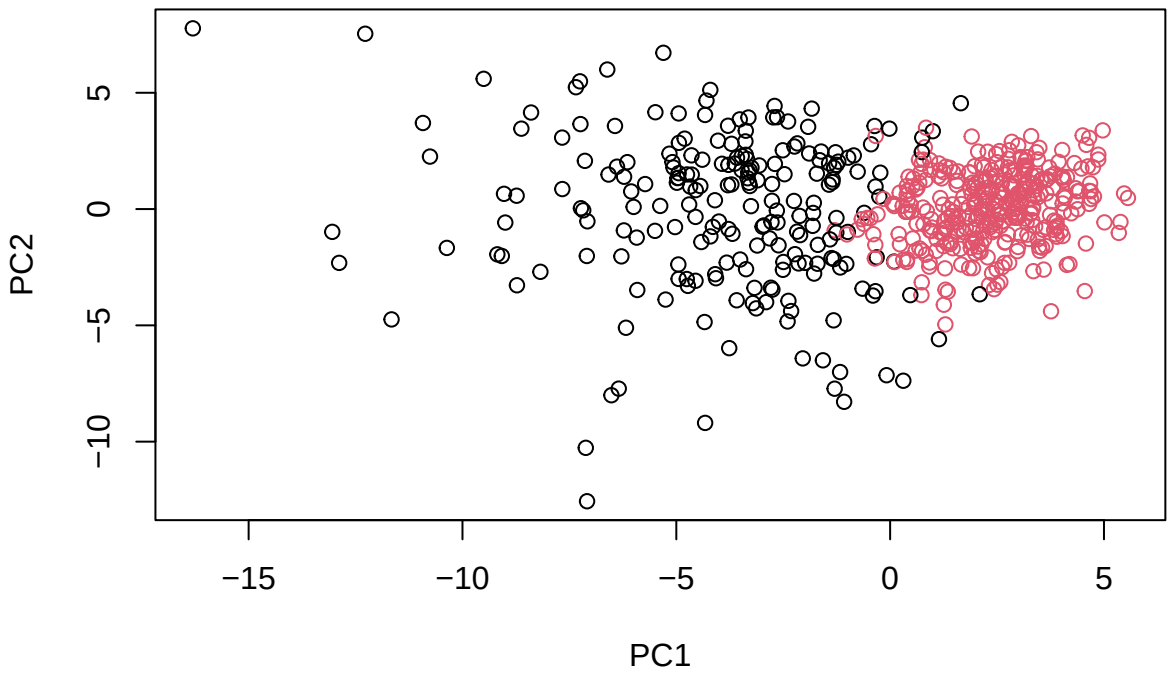
```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

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

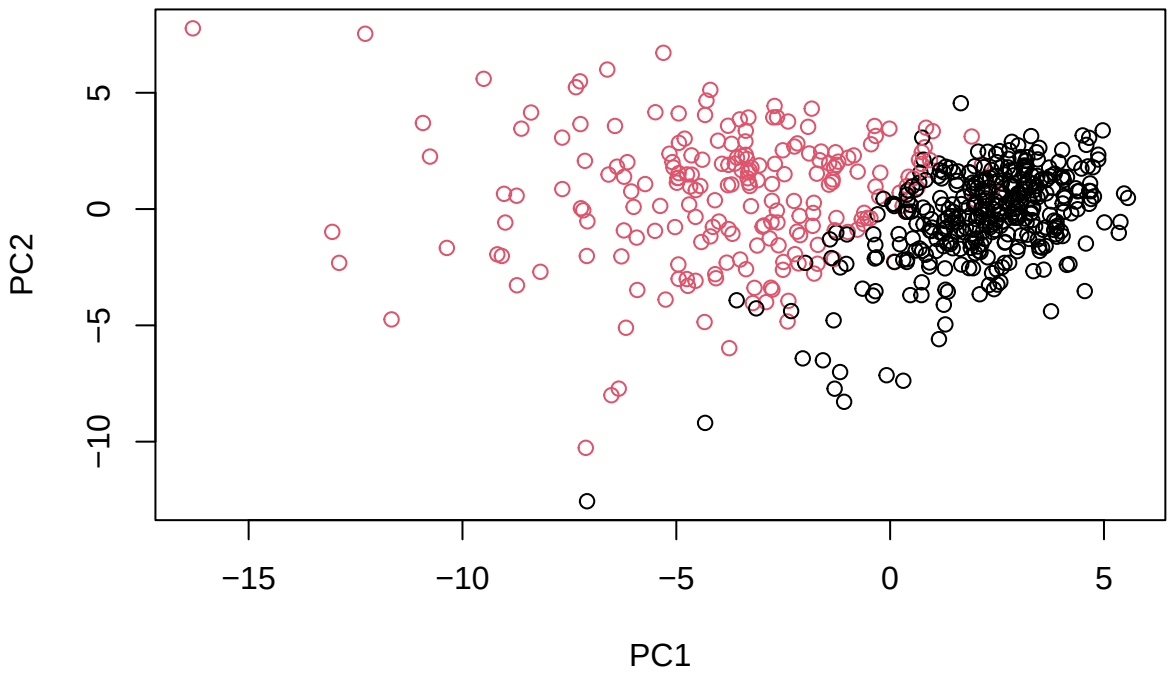
```
table(grps, diagnosis)
```

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

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




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



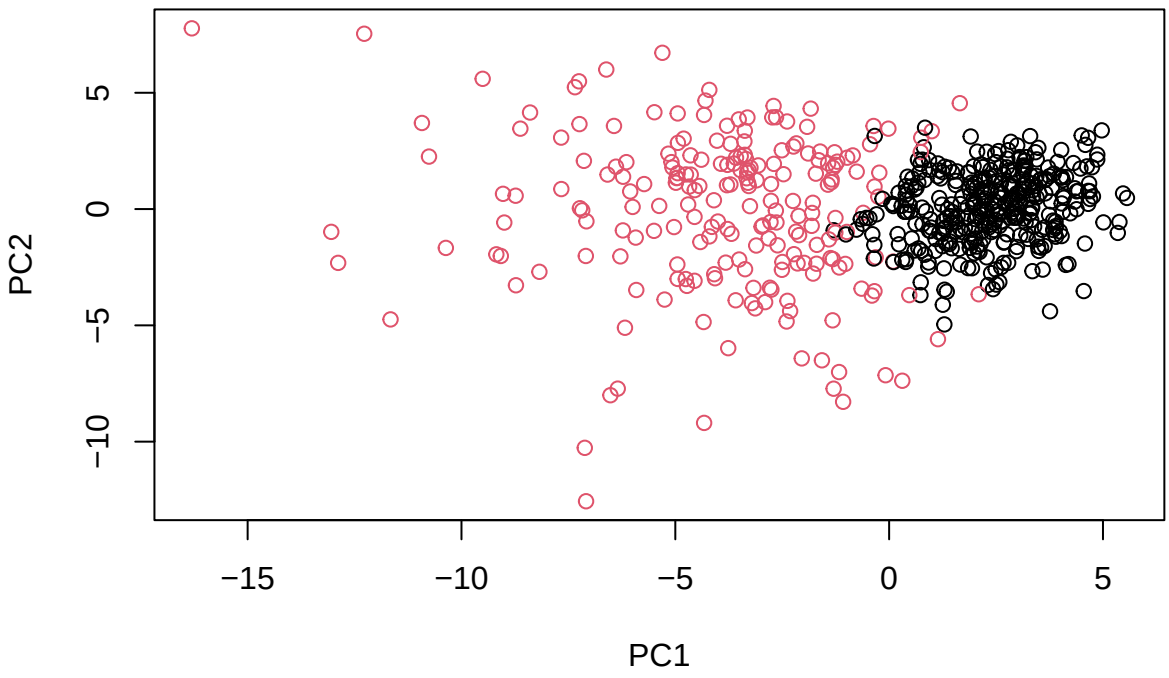

```
g <- as.factor(grps)
levels(g)
```

```
## [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)
```




```
## Use the distance along the first 7 PCs for clustering i.e. wisc.pr$x[, 1:7]

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

## Cut this hierarchical clustering model into 2 clusters and assign the results to wisc.pr.hclust.clust

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?

This method separates the two diagnoses by majority, but i think could be better

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?

It seems that the k-means clustering model from the example did a better job at separating the diagnoses because there is a higher proportion of malignant samples in the cluster where malignant is the majority than out hierarchical clustering model. But, the hierarchical model still separated them pretty well.

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

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

-sensitivity

wisc.km$cluster

```
175/(175+37)
```

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

wisc.hclust.clusters

```
165/(165+5+40+2)
```

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

The wisc.km$cluster analysis has a higher sensitivity

-specificity

wisc.km$cluster

```
343/(343+14)
```

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

wisc.hclust.clusters

```
343/(12+2+343)
```

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

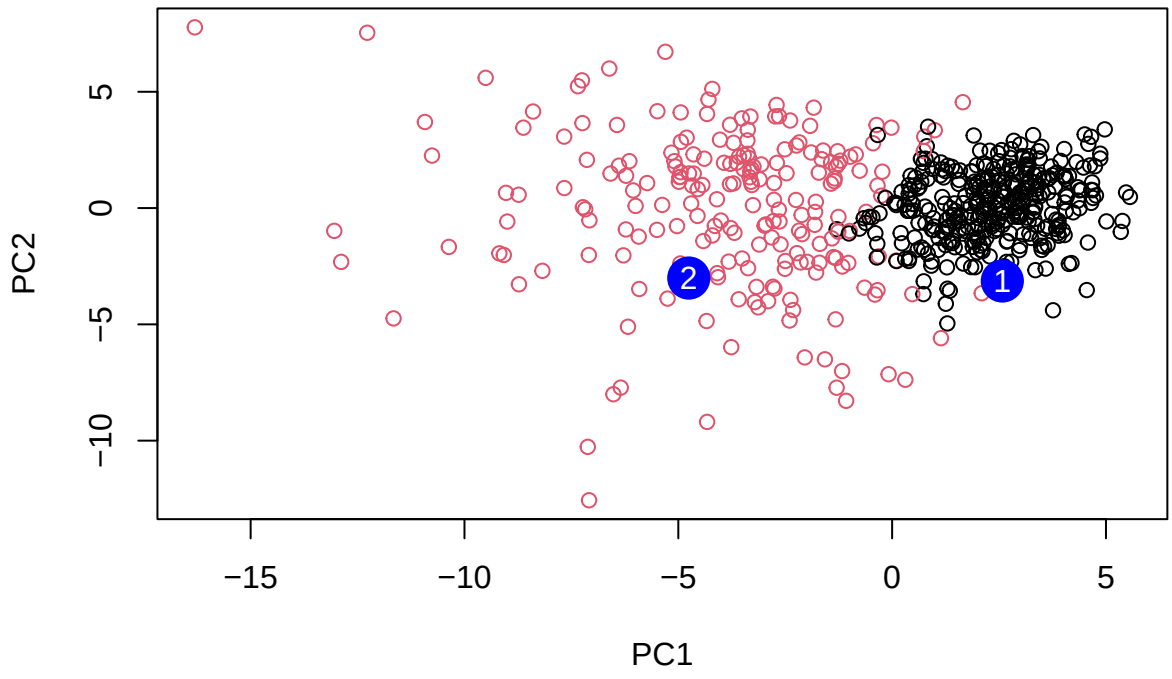
The sensitivity of the two analyses is the same

Prediction

```
#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?

We should follow up on patient 2