Lab 9: Mini Project

Taylor Darby

10/27/2021

1. Exploratory data analysis

Preparing the data

```r
# Save input data file into Project directory
fna.data <- "WisconsinCancer.csv"

# read the csv file
wisc.d <- read.csv(fna.data, row.names = 1)

# examine input data
head(wisc.d)
```

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

```
# **WARNING** determined there was an issue with the file. A last empty column was inserted and needs t

ncol(wisc.d)
```

```
## [1] 32
```

```
# There are 32 columns so we will remove the last column
wisc.df <- wisc.d[,-32]

ncol(wisc.df)
```

```
## [1] 31
```

```
# Remove the first column of the dataset because essentially this is our "answer" in our unsupervised a

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

# Setup a separate vector with data from the "diagnosis" column only ('as.factor()' function makes usin
diagnosis <- as.factor(wisc.df$diagnosis)
diagnosis
```

```
##   [1] M M M M M M M M M M M M M M M M M M M B B B M M M M M M M M M M M M M M M
##  [38] B M M M M M M M M B M B B B B B M M B M M B B B B M B M M B B B B M B M M
##  [75] B M B M M B B B M M B M M M B B B M B B M M B B B M M B B B B M B B M B B
## [112] B B B B B B M M M B M M B B B M M B M B M M B M M B B M B B M B B B B M B
## [149] B B B B B B B B M B B B B M M B M B B M M B B M M B B B B M B B M M M B M
## [186] B M B B B M B B M M B M M M M B M M M B M B M B B M B M M M M B B M M B B
## [223] B M B B B B B M M B B M B B M M B M B B B B M B B B B B M B M M M M M M M
## [260] M M M M M M M B B B B B B M B M B B M B B M B M M B B B B B B B B B B B B
## [297] B M B B M B M B B B B B B B B B B B B B B M B B B M B M B B B B M M M B B
## [334] B B M B M B M B B B M B B B B B B B M M M B B B B B B B B B B B M M B M M
## [371] M B M M B B B B B M B B B B B M B B B M B B M M B B B B B B M B B B B B B
## [408] B M B B B B B M B B M B B B B B B B B B B B B M B M M B M B B B B B M B B
## [445] M B M B B M B M B B B B B B B B M M B B B B B B M B B B B B B B B B B M B
## [482] B B B B B B M B M B B M B B B B B M M B M B M B B B B B M B B M B M B M M
## [519] B B B M B B B B B B B B B B B M B M M B B B B B B B B B B B B B B B B B B
## [556] B B B B B B B M M M M M M B
## Levels: B M
```

Exploratory data analysis

> **Q1.** How many observations are in this dataset?

There are 569 observations in this dataset.

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

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

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

There are 212 malignant diagnosis.

```r
# generate a table of the "diagnosis" data to sum each categorical value
table(diagnosis)
```

```
## diagnosis
##   B   M
## 357 212
```

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

There are 10 variables/features suffixed with '_mean'

```r
# use grep to determine where the pattern '_mean" occurs and use 'length' to count how many times the p

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

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

2. Principal Component Analysis

Performing PCA

```
# Check the columns of the 'wisc.data' to determine if the data should 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
```

```r
# Perform PCA on wisc.data by completing the following code --> determined data needs to be scaled
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 of Variance PC1 = 0.4427

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

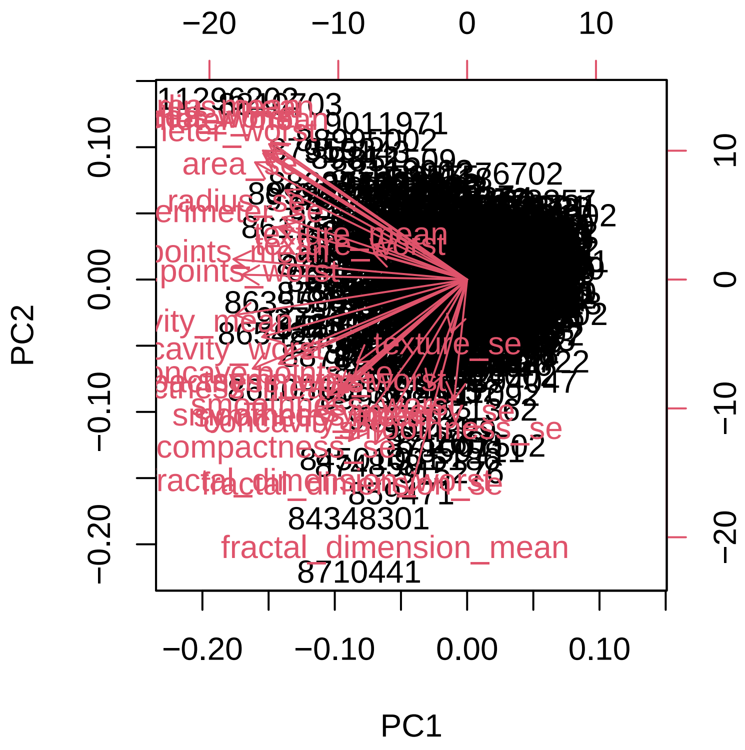
Three. In PC3 at least 70% of the original variance is described

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

Seven. In PC7 at least 90% of the original variance is described

Interpreting PCA results

```r
# Create a biplot of the PCA data (wisc.pr)
biplot(wisc.pr)
```

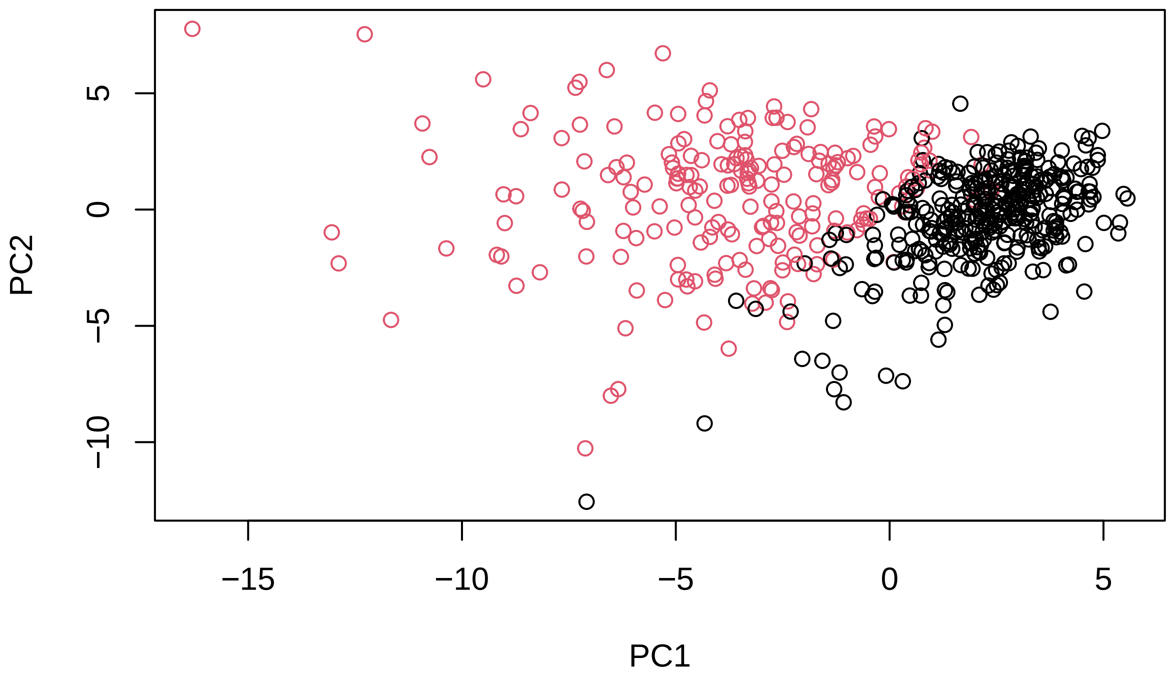


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

There are way too many datapoints to be able to understand what is going on.

Let's find a solution to this:

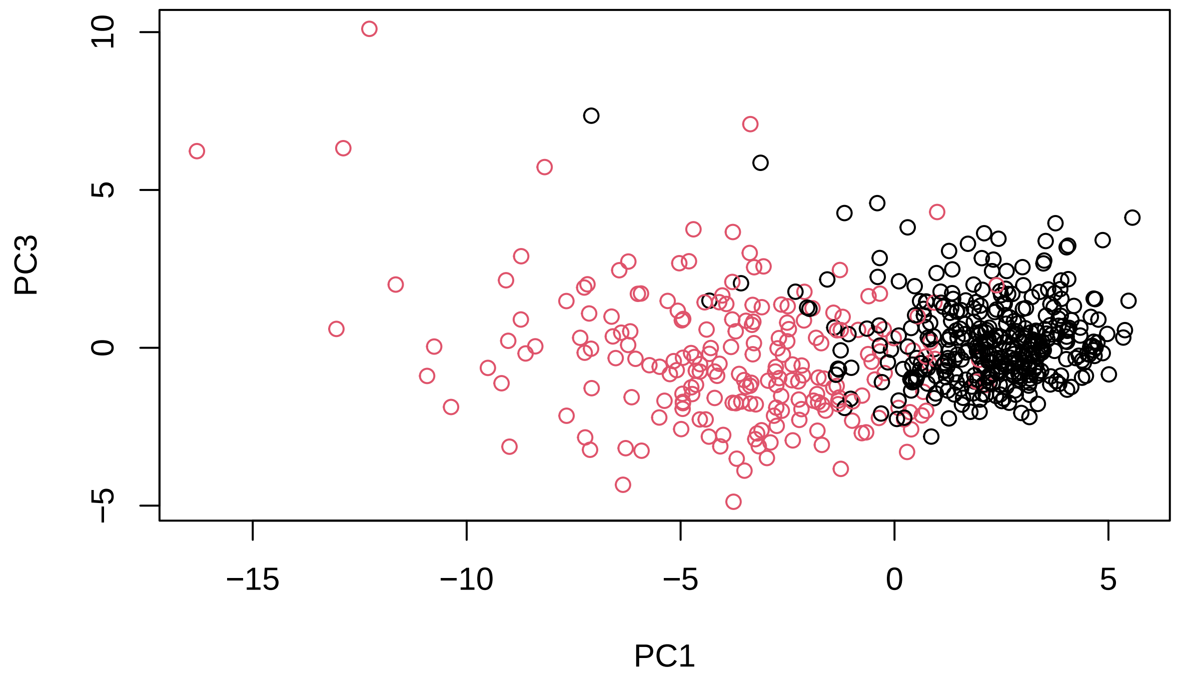
```
# Generate a more standard scatter plot of each observation along principal components 1 and 2 and colo

plot(wisc.pr$x, col=diagnosis, xlab="PC1", ylab="PC2")
```



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

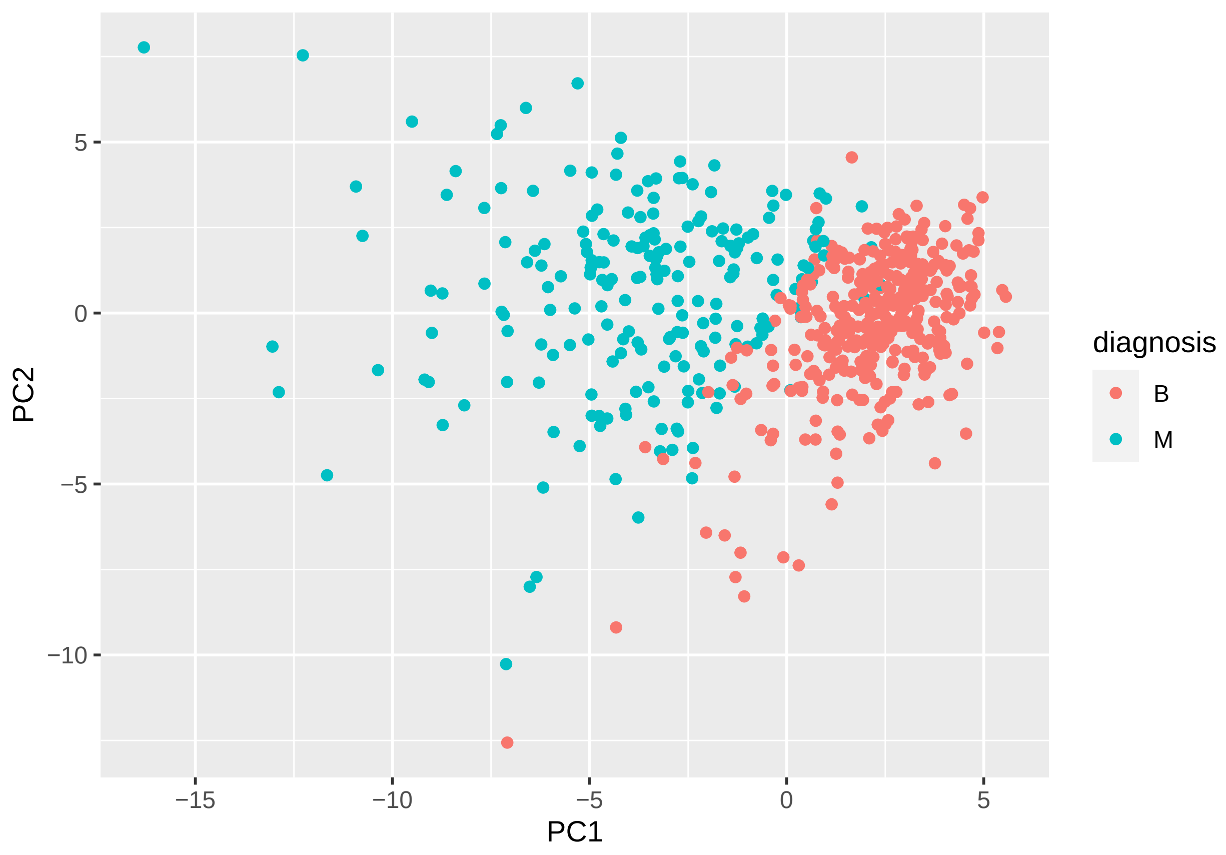
The plots look very similar but the PC3 values are shifted down a bit.

```r
# Repeat for components 1 and 3
plot(wisc.pr$x[,c("PC1","PC3")], col = diagnosis,
     xlab = "PC1", ylab = "PC3")
```



Let's try in **ggplot**

```
# 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()
```



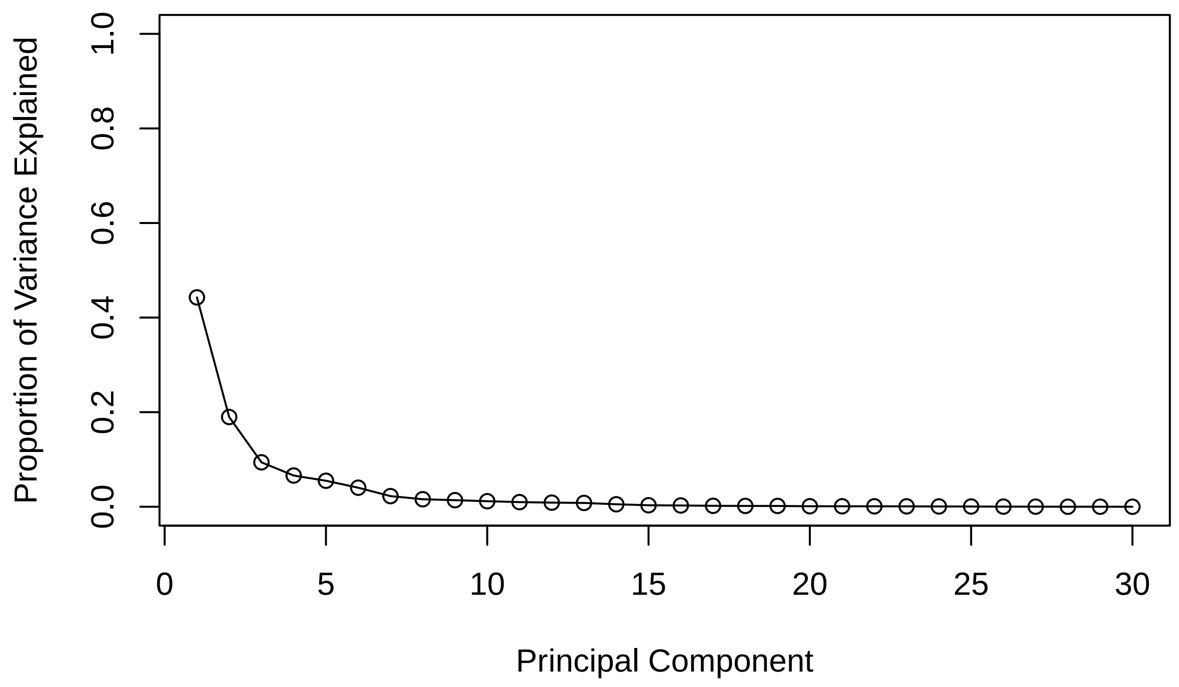
Variance explained

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

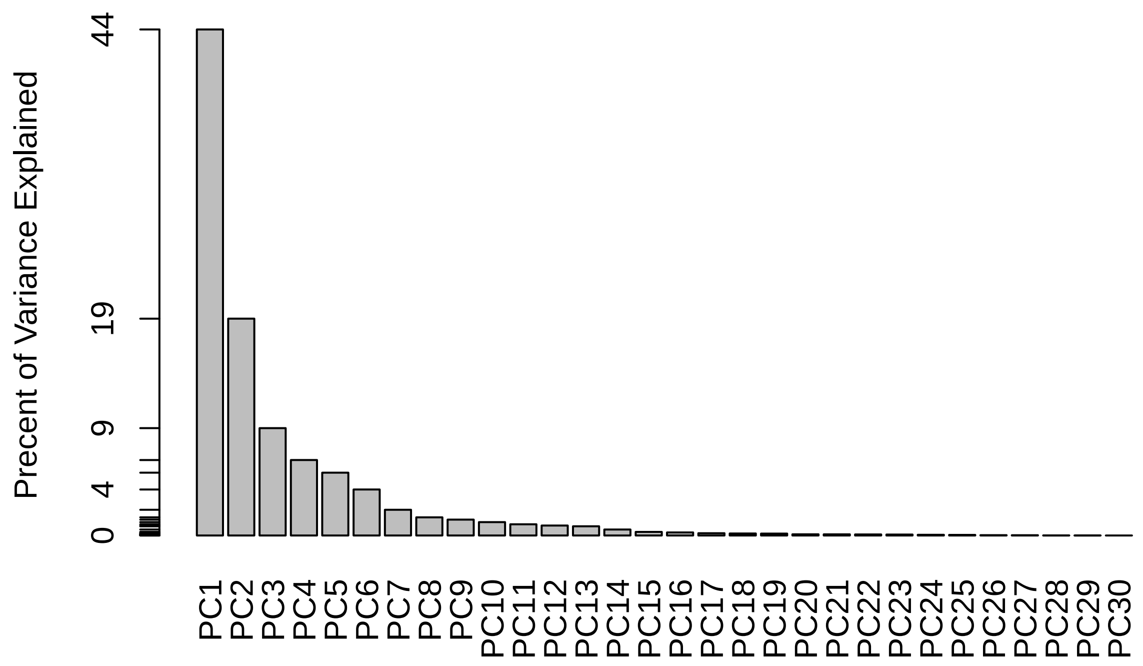
```
## [1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

Calculate the variance explained by each principal component by dividing by the total variance explained of all principal components.

```
# 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 )
```

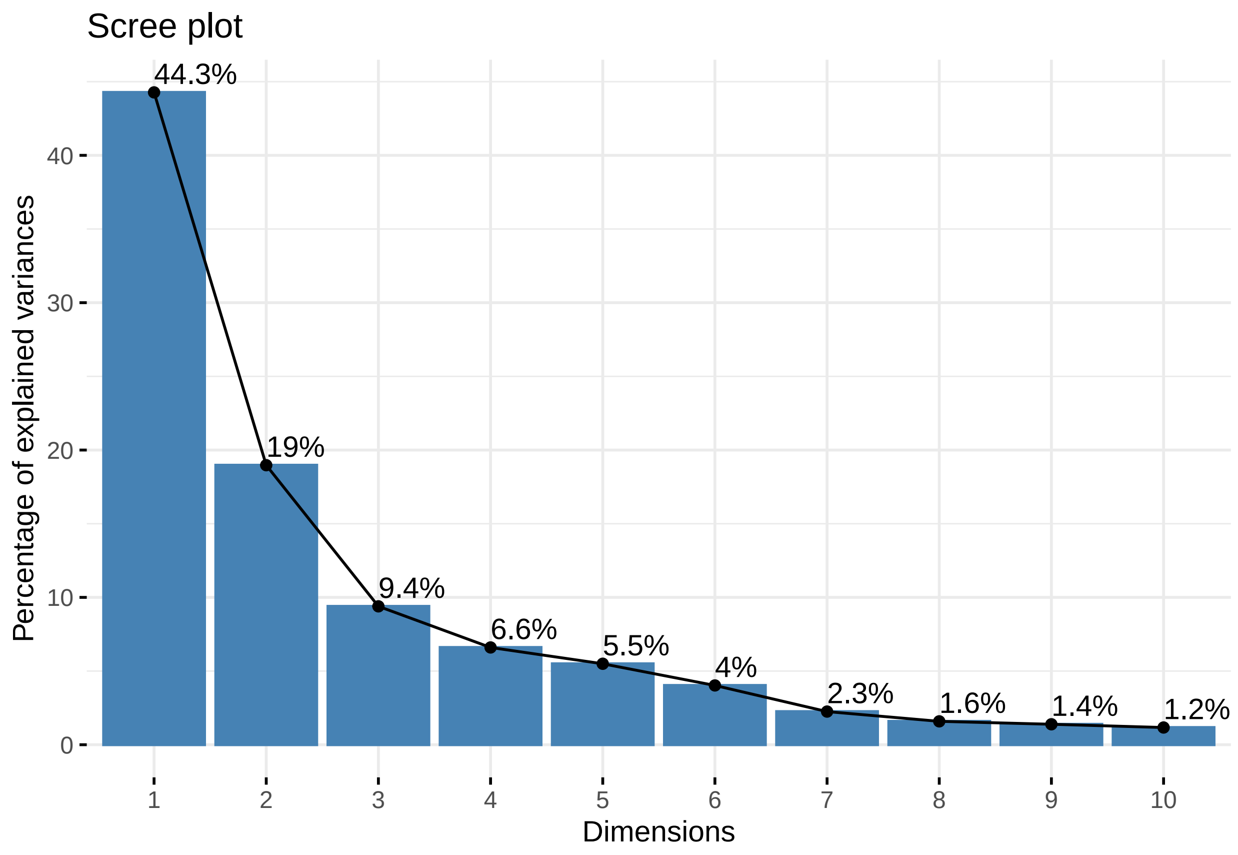


OPTIONAL: There are quite a few CRAN packages that are helpful for PCA. This includes the factoextra package. Feel free to explore this package.

```r
# install.packages("factoextra")
library(factoextra)
```

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

```r
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?

-0.2608538

```
# call the first column of row "concave.points_mean" from the "rotation" dataset of 'wisc.pr'
wisc.pr$rotation["concave.points_mean",1]
```

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

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

4

There are four PCs with less than 80% variance of the data. That means it takes 5 PCs to explain 80% or more of the variance of the data.

```r
var <- summary(wisc.pr)
sum(var$importance[3,] < 0.8)
```

```
## [1] 4
```

3. Hierarchical clustering

```r
# Scale the wisc.data data using the "scale()" function
data.scaled <- scale(wisc.data)
```

```r
# Calculate the (Euclidean) distances between all pairs of observations in the new scaled dataset and a
data.dist <-  dist(data.scaled)
```

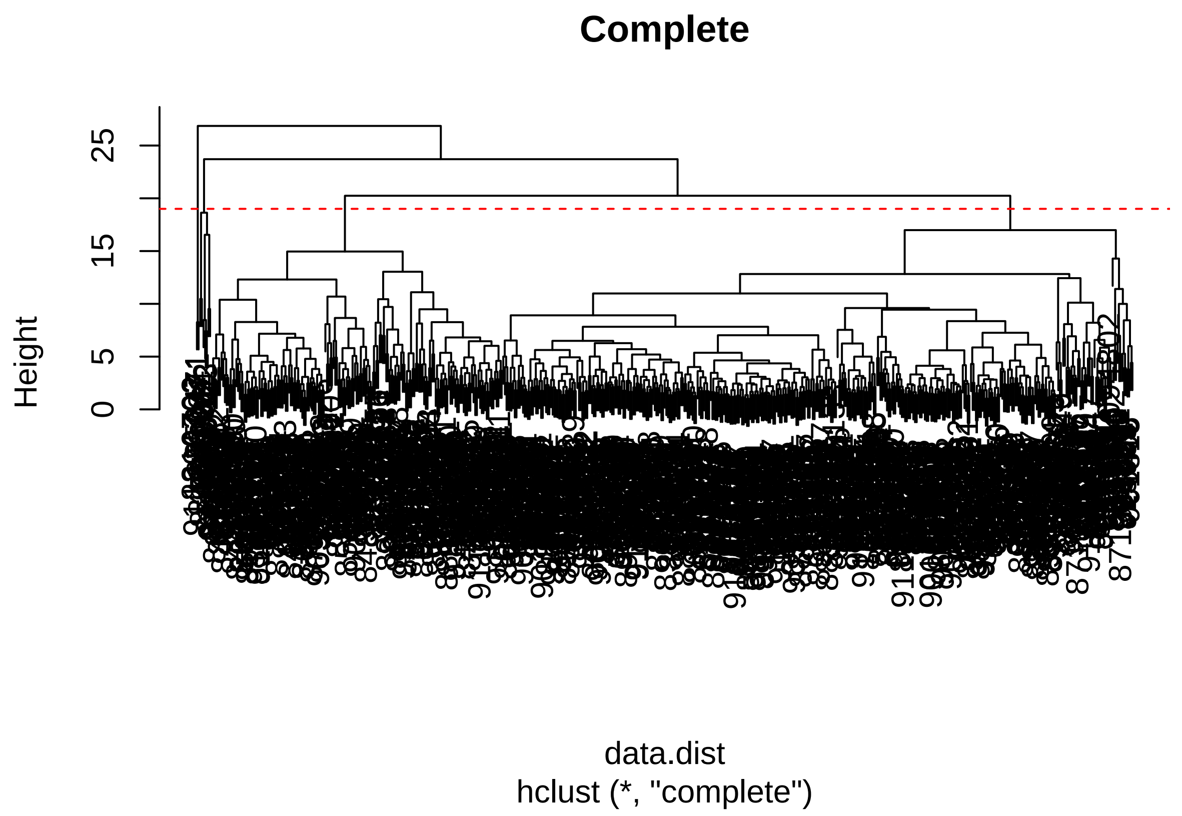
```r
# Create a hierarchical clustering model using complete linkage. Manually specify the method argument t
wisc.hclust <- hclust(data.dist, method = "complete")
```

Results of hierarchical clustering

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

At about a height of 19 the clustering model has 4 clusters.

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



Use ‘cutree()’ to cut the tree so that it has 4 clusters. Assign the output to the variable wisc.hclust.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?

Four clusters seems to be the best, but 6 clusters results in a third, very small group of 12 benign cells.

```
wisc.hclust.clusters.2 <- cutree(wisc.hclust, k = 6)
table(wisc.hclust.clusters.2, diagnosis)
```

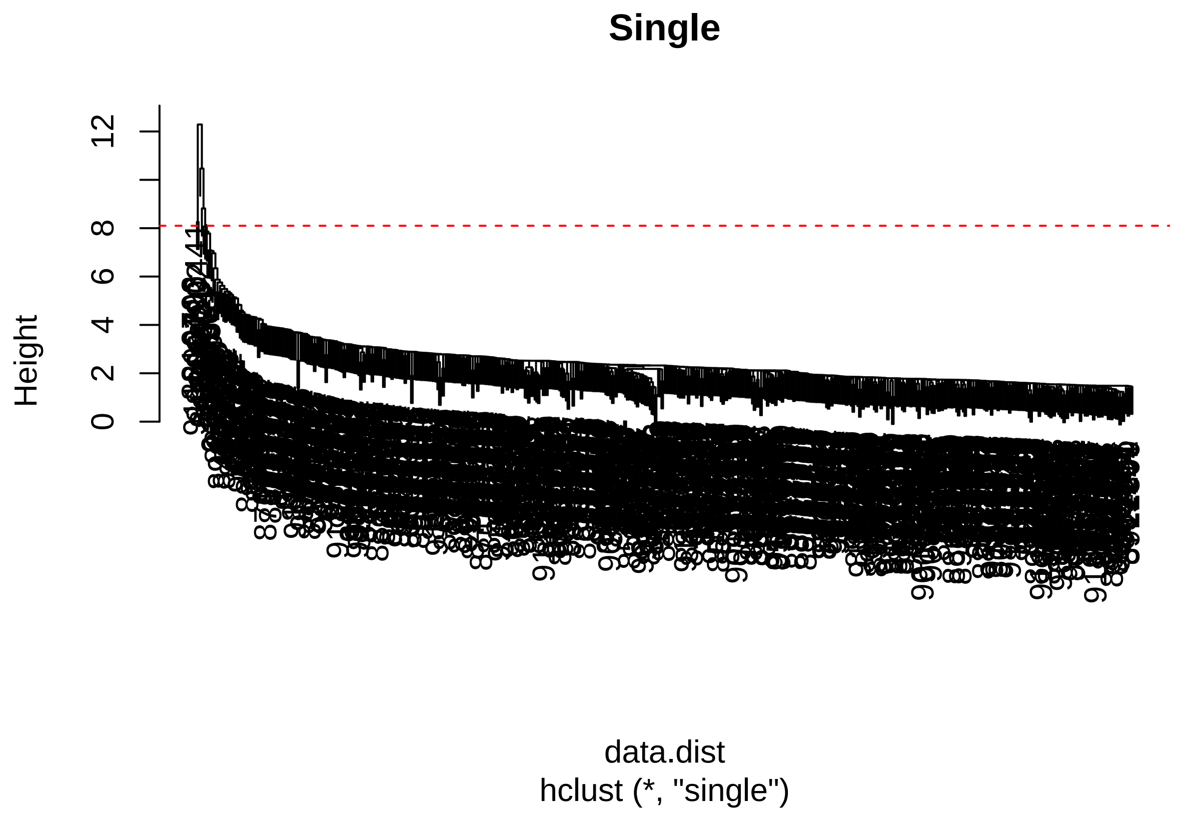
```
##                       diagnosis
## wisc.hclust.clusters.2   B   M
##                          1  12 165
##                          2   0   5
##                          3 331  39
##                          4   2   0
##                          5  12   1
##                          6   0   2
```

Using different methods

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

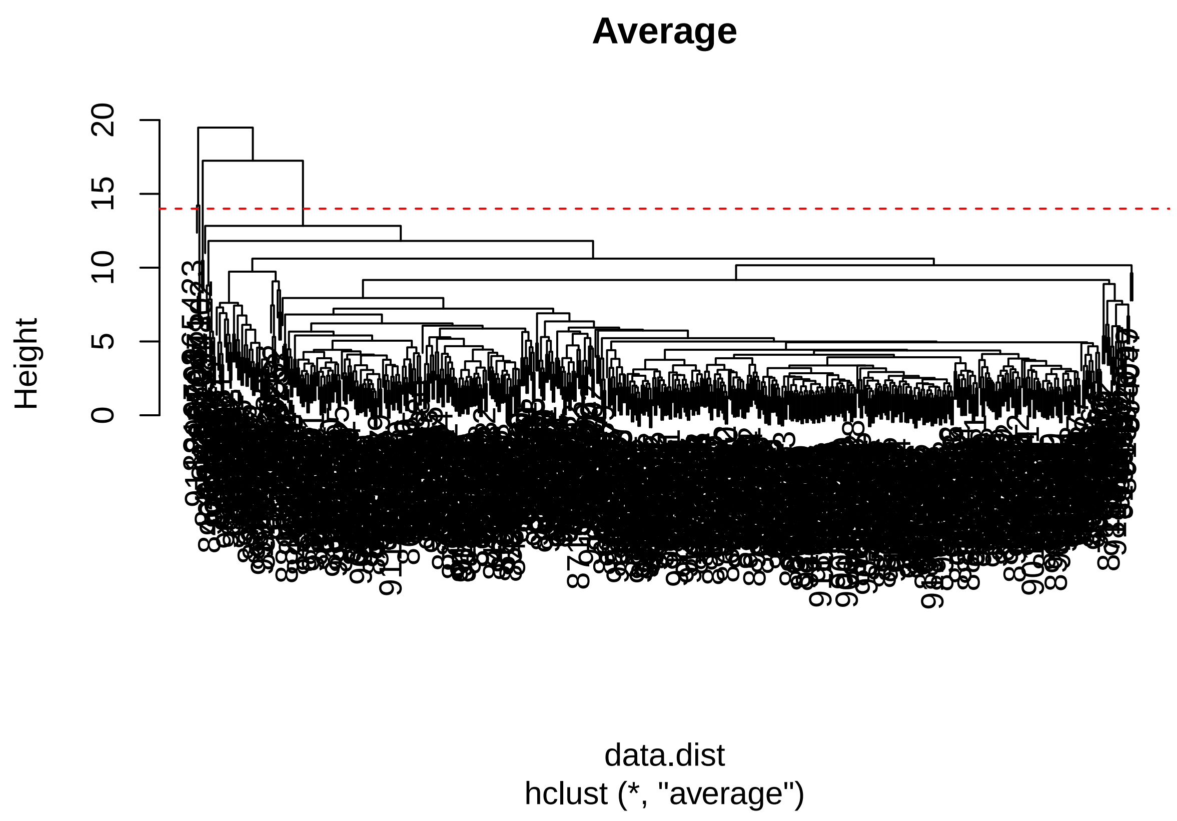
I think the 'ward.D2' is the best method because we are plotting PCA results and 'ward.D2' clusters based on variance (is based on multidimensional variance) like PCA

```
# "Single" method cut at 4 clusters
wisc.hclust.single <- hclust(data.dist, method = "single")
plot(wisc.hclust.single, main="Single")
abline(h=8.1 , col="red", lty=2)
```

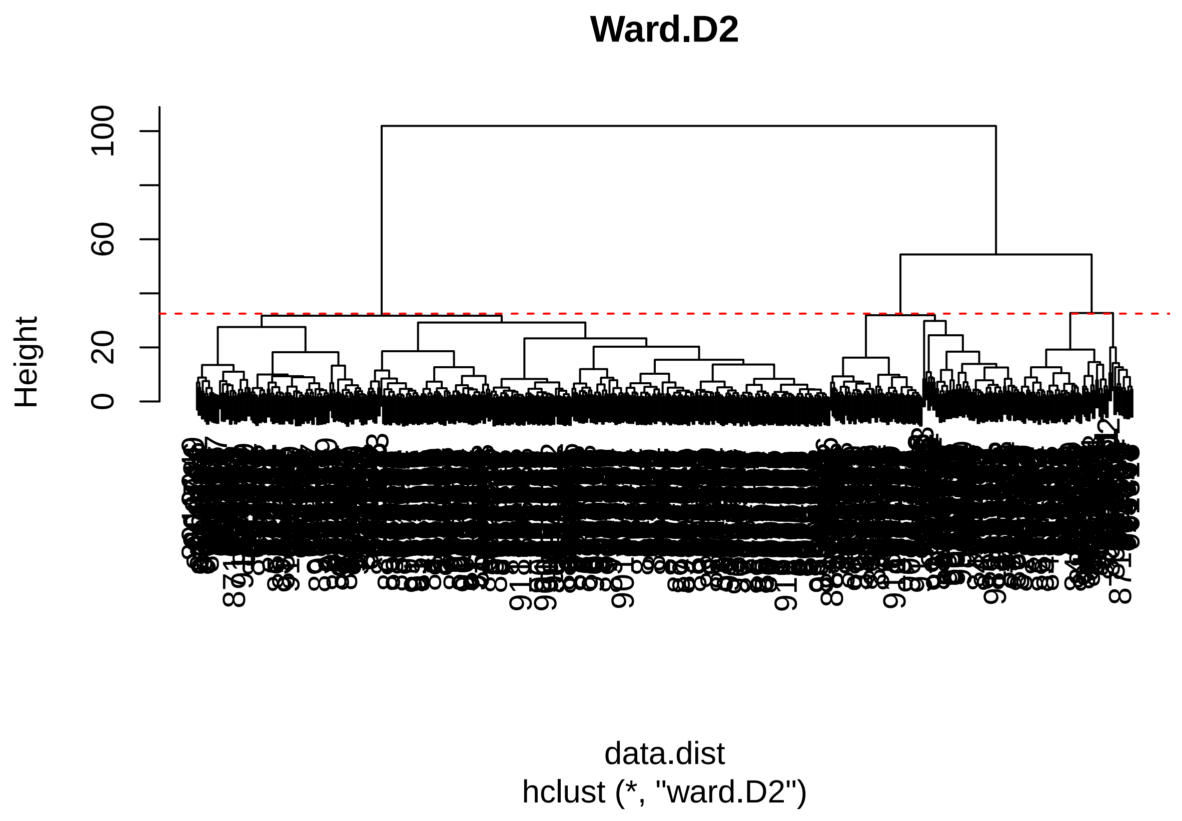


```
# "Average" method cut at 4 clusters
wisc.hclust.average <- hclust(data.dist, method = "average")
```

```r
plot(wisc.hclust.average, main="Average")
abline(h=14 , col="red", lty=2)
```



```r
# "Ward.D2" method cut at 4 clusters
wisc.hclust.ward <- hclust(data.dist, method = "ward.D2")
plot(wisc.hclust.ward, main="Ward.D2")
abline(h=32.5 , col="red", lty=2)
```



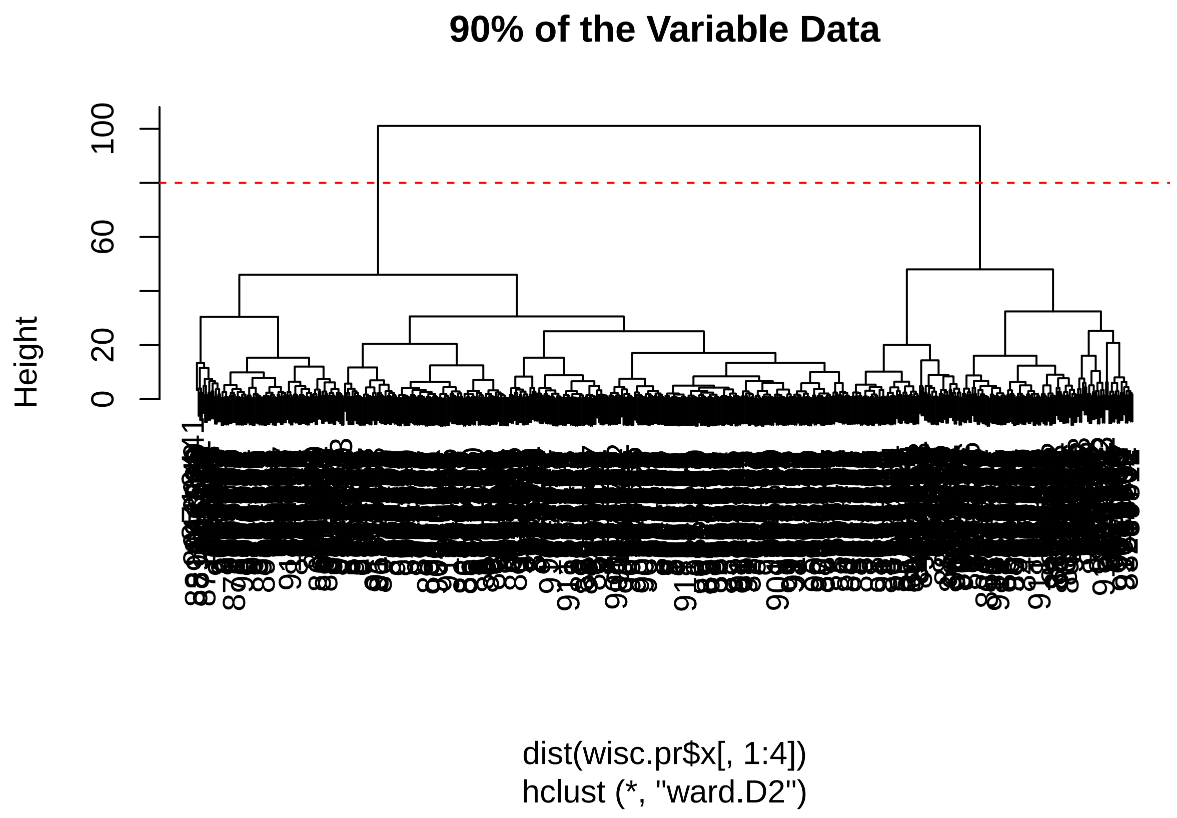
Q14. (OPTIONAL) - SKIPPED

Started this section of Lab 9 the folowing class Oct. 29, 2021

5. Combining methods

I will use 4 PCs this time and 'hclust()' and 'dist()' as input

```
wisc.pr.hclust <- hclust(dist(wisc.pr$x[,1:4]), method="ward.D2")
plot(wisc.pr.hclust, main="90% of the Variable Data")
abline(h=80 , col="red", lty=2)
```



Let’s find our cluster membership vector by cutting this tree into k=2 groups.

```
# two main branches of or dendrogram indicating two main clusters - maybe these are malignant and benig
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
## grps
##   1   2
## 171 398
```

Now let’s compare to the expert M and B vector

```
table(diagnosis)
```

```
## diagnosis
##   B   M
## 357 212
```

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

Very well. See table below.

We can do a cross-table by giving the ‘table()’ function to two inputs.

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

```
##     diagnosis
## grps   B   M
##    1   6 165
##    2 351  47
```

> **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.

I didn't do the optional k-means section but here you can see before ward.D2 we need to generate 4 groups before seeing separation. After ward.D2 the separation is clearer.

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

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

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

```
##     diagnosis
## grps   B   M
##    1   6 165
##    2 351  47
```

6. Sensitivity/Specificity

Accuracy: essentially how many did we get correct?

```
# pre-ward.d2
(165+343) / nrow(wisc.data)
```

```
## [1] 0.8927944
```

```
# post-ward.d2
(165+351) / nrow(wisc.data)
```

```
## [1] 0.9068541
```

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

PCA scaled and grouped into 2 groups using ward.D2 is more specific, but not as sensitive as the data before ward.D2 analysis.

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).

```
# pre-ward.d2
(165/(165+12))
```

```
## [1] 0.9322034
```

```
# post-ward.d2
(165/(165+6))
```

```
## [1] 0.9649123
```

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).

```
# pre-ward.d2
(343/(343+40))
```

```
## [1] 0.8955614
```

```
# post-ward.d2
(351/(351+47))
```

```
## [1] 0.8819095
```

7. Prediction

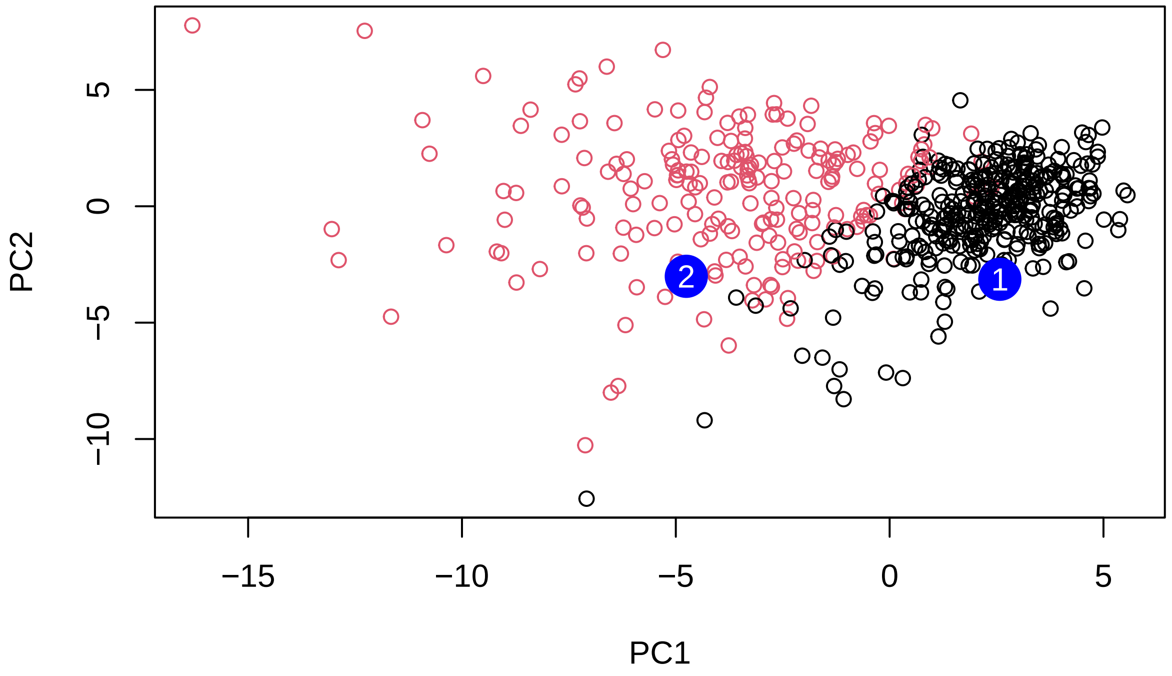
We will use the predict() function that will take our PCA model from before and new cancer cell data and project that data onto our PCA space.

```
#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
```

Now add these new samples to our PCA plot

```
plot(wisc.pr$x[,1:2], col=diagnosis)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```



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

I would follow up with patient 2 because the analysis predicted their test results were consistent with the malignant profile according to our previous PCA analysis.

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] factoextra_1.0.7 ggplot2_3.3.5
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.26         purrr_0.3.4       haven_2.4.3
##  [5] carData_3.0-4     colorspace_2.0-2  vctrs_0.3.8       generics_0.1.0
##  [9] htmltools_0.5.2   yaml_2.2.1        utf8_1.2.2        rlang_0.4.11
## [13] pillar_1.6.3      ggpubr_0.4.0      foreign_0.8-81    glue_1.4.2
## [17] withr_2.4.2       readxl_1.3.1      lifecycle_1.0.1   stringr_1.4.0
## [21] cellranger_1.1.0  munsell_0.5.0     ggsignif_0.6.3    gtable_0.3.0
## [25] zip_2.2.0         evaluate_0.14     labeling_0.4.2    knitr_1.36
## [29] rio_0.5.27        forcats_0.5.1     fastmap_1.1.0     curl_4.3.2
## [33] fansi_0.5.0       highr_0.9         broom_0.7.9       Rcpp_1.0.7
## [37] scales_1.1.1      backports_1.3.0   abind_1.4-5       farver_2.1.0
## [41] hms_1.1.1         digest_0.6.28     stringi_1.7.5     openxlsx_4.2.4
## [45] rstatix_0.7.0     dplyr_1.0.7       ggrepel_0.9.1     grid_4.1.1
## [49] tools_4.1.1       magrittr_2.0.1    tibble_3.1.5      crayon_1.4.1
## [53] tidyr_1.1.4       car_3.0-11        pkgconfig_2.0.3   ellipsis_0.3.2
## [57] data.table_1.14.2 rmarkdown_2.11    R6_2.5.1          compiler_4.1.1
```