Unsupervised Learning Mini-Project

Core functions:

read.csv("YourFileName")

prcomp(x, scale = TRUE)

kmeans(x, centers = ?)

hclust(dist(x))

```r
# Make Available Data for Project
wisc.df <- read.csv("WisconsinCancer.csv", row.names = 1)
# Separate data and physician diagnosis
wisc.data <- (wisc.df[,-1])
diagnosis <- (wisc.df[,1])
```

Exploratory Data Analysis.

#Q1. There are 569 observations in this dataset.

```r
#Check how many observations have a malignant diagnosis.
x = 0
for(i in 1:length(diagnosis)) {
  if(diagnosis[i] == "M") {x <- x+1}
}
x
```

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

```r
#also 'table(diagnosis)'
#also 'sum(diagnosis == "M")
```

#Q2. There are 212 observations with a malignant diagnosis.

```r
# Check how many variables end with "_mean".
length(grep(pattern = "*_mean", x = colnames(wisc.data)))
```

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

#Q3. There are 10 variables suffixed with "_mean".

#Principal Component Analysis.

```
# Check column means and standard deviations (1 = rows, 2 = cols)
apply(wisc.data, 2, mean)
```

```
##             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
```

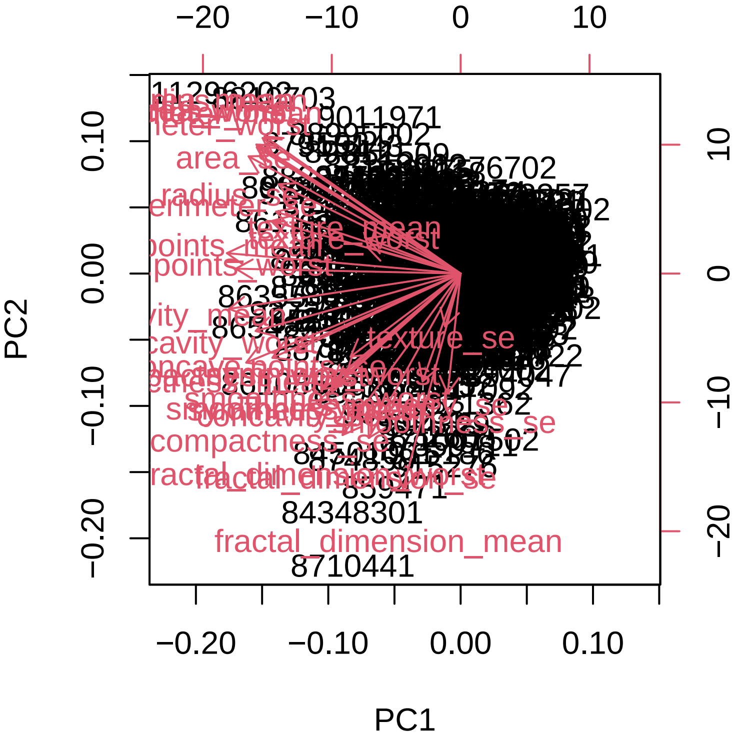
```
# Perform PCA on wisc.data by completing the following code
wisc.pr <- prcomp(wisc.data, scale=TRUE)
summary(wisc.pr)
```

```
## 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
```

```
# The more PCs you need to describe the data, the more all-over-the-place it is...
```

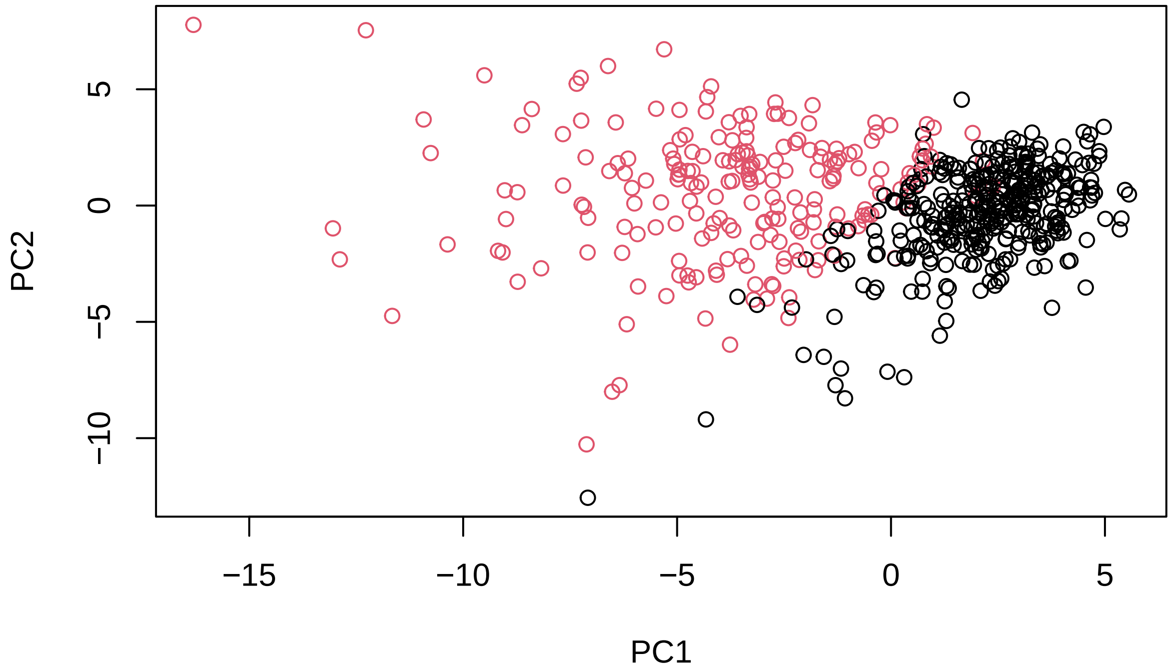
#Q4. The first principal component captures 44% of the variance in the data. #Q5. At least 3 PCs to describe >70% of the variance. #Q6. At least 7 PCs to describe >90% of the variance.

```
# Visualize PCA Results
biplot(wisc.pr)
```

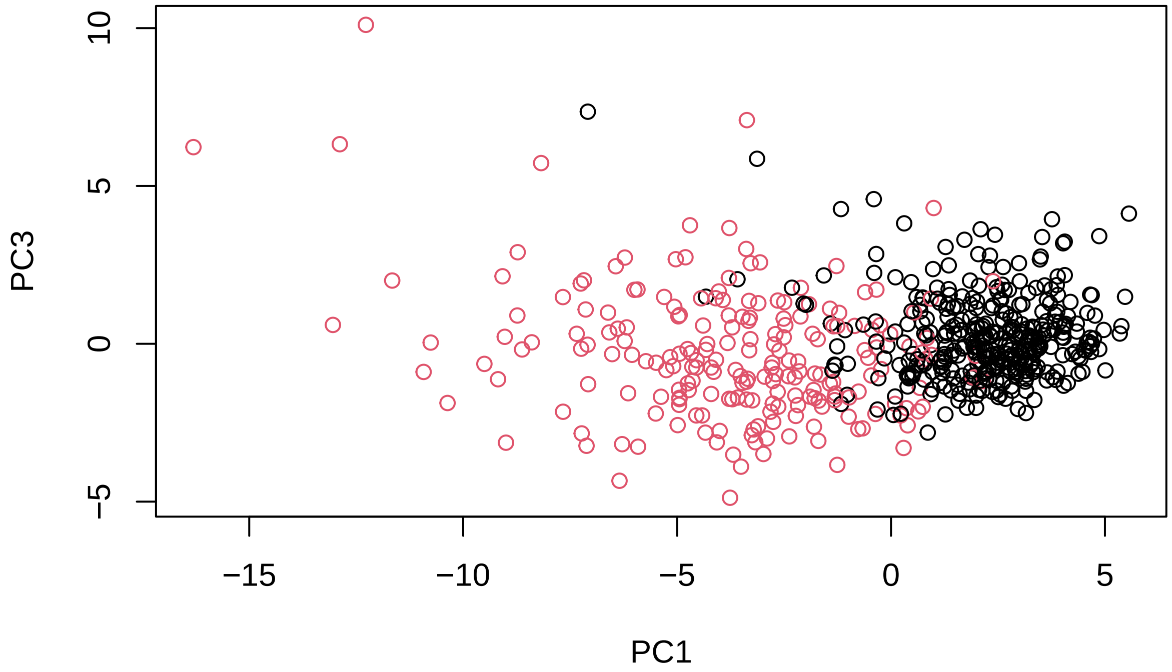


#Q7. The plot contains a vast amount of data, but there seems to be an overall leftward trend. The plot is not at all easily interpretable.

```
# Visualize PCA Results, but better
factor_diagnosis <- as.factor(diagnosis)
plot(wisc.pr$x[,1:2] , col=factor_diagnosis)
```



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



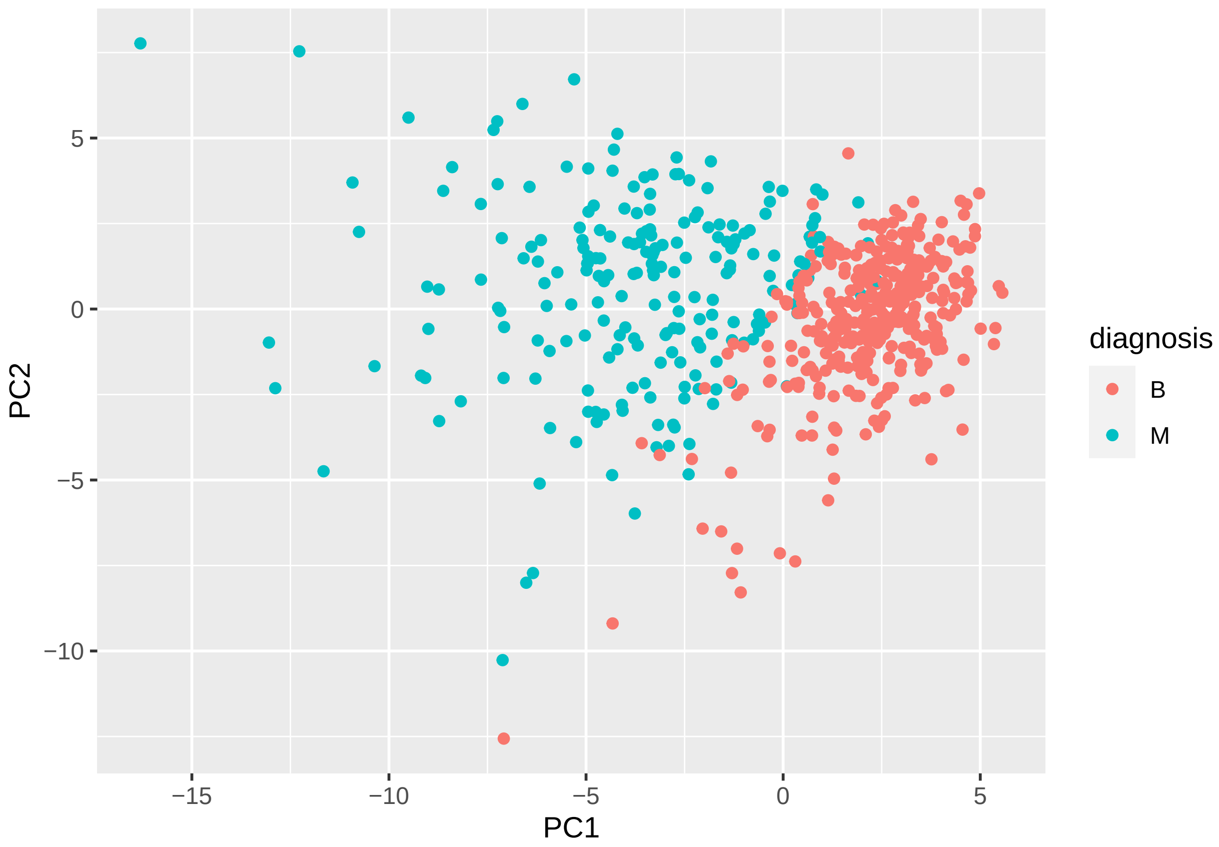
#Q8. The plots are largely similiar. The only notable differences are that the P1/P2 plot seems to have negative outliers, whereas the P1/P3 plot has positive ones. These outliers change the y-scale of the plot, and the separation seems to be a little clearer in the P1/P2 plot, as the black values in the P1/P3 plot cross far into the red cluster territory.

```
# Visualize data in ggplot

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

# Load the ggplot2 package
library(ggplot2)

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



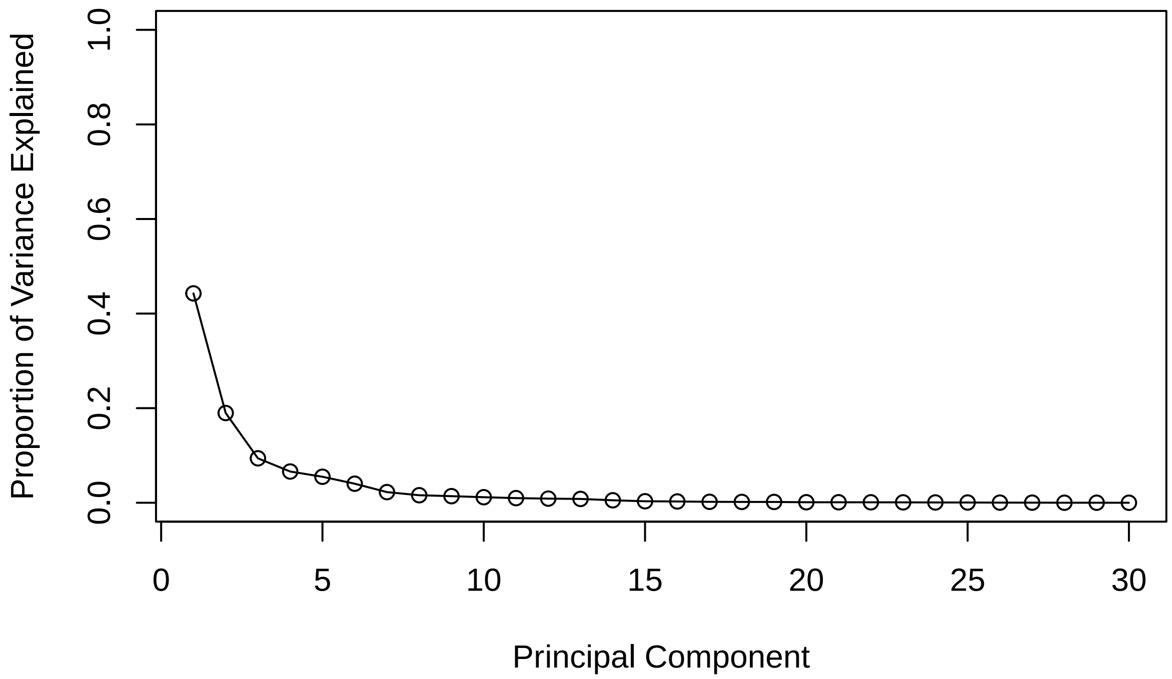
```
# Visualize Variance Capturing by PCs

# Calculate variance from standard deviation
pr.var <- wisc.pr$sdev^2
pr.var
```

```
##  [1] 1.328161e+01 5.691355e+00 2.817949e+00 1.980640e+00 1.648731e+00
##  [6] 1.207357e+00 6.752201e-01 4.766171e-01 4.168948e-01 3.506935e-01
## [11] 2.939157e-01 2.611614e-01 2.413575e-01 1.570097e-01 9.413497e-02
## [16] 7.986280e-02 5.939904e-02 5.261878e-02 4.947759e-02 3.115940e-02
## [21] 2.997289e-02 2.743940e-02 2.434084e-02 1.805501e-02 1.548127e-02
## [26] 8.177640e-03 6.900464e-03 1.589338e-03 7.488031e-04 1.330448e-04
```

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

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

#Q9. The component of the loading vector for "concave.points_mean" is -0.26. #Q10. At least 5 PCs to describe >80% of the variance.

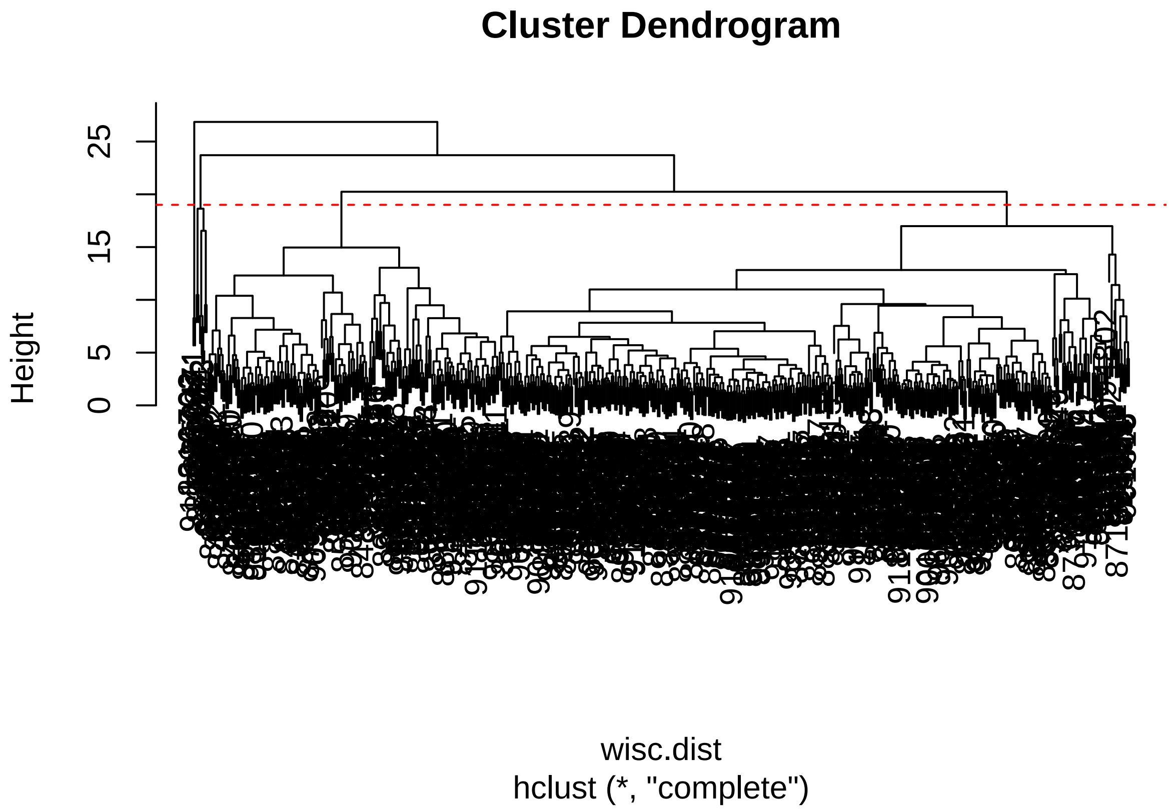
Hierarchical Clustering.

```
# Try 'complete' clustering method

# Calculate scaled Euclidean distances of data points
wisc.dist <- dist(scale(wisc.data))

# Cluster w/ 'hclust()'
wisc.hclust <- hclust(wisc.dist, method="complete")

# Visualize
plot(wisc.hclust)
abline(h=19, col="red", lty=2)
```



#Q11. The abline must be placed at h=19 to cut the tree into 4 groups.

```
# Extract Data
ans <- NULL
for(i in 2:10) {
  x <- cutree(wisc.hclust, k=i)
  ans <- rbind(ans, x)
}
#ans

for(i in 1:9) {
  print(table(ans[i,], diagnosis))
}
```

```
##    diagnosis
##       B   M
##   1 357 210
##   2   0   2
##    diagnosis
##       B   M
##   1 355 205
##   2   2   5
##   3   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   2   5
##   3 343  40
##   4   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   0   5
##   3 343  40
##   4   2   0
##   5   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   0   5
##   3 331  39
##   4   2   0
##   5  12   1
##   6   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   0   3
##   3 331  39
##   4   2   0
##   5  12   1
##   6   0   2
##   7   0   2
##    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
##    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
##     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
```

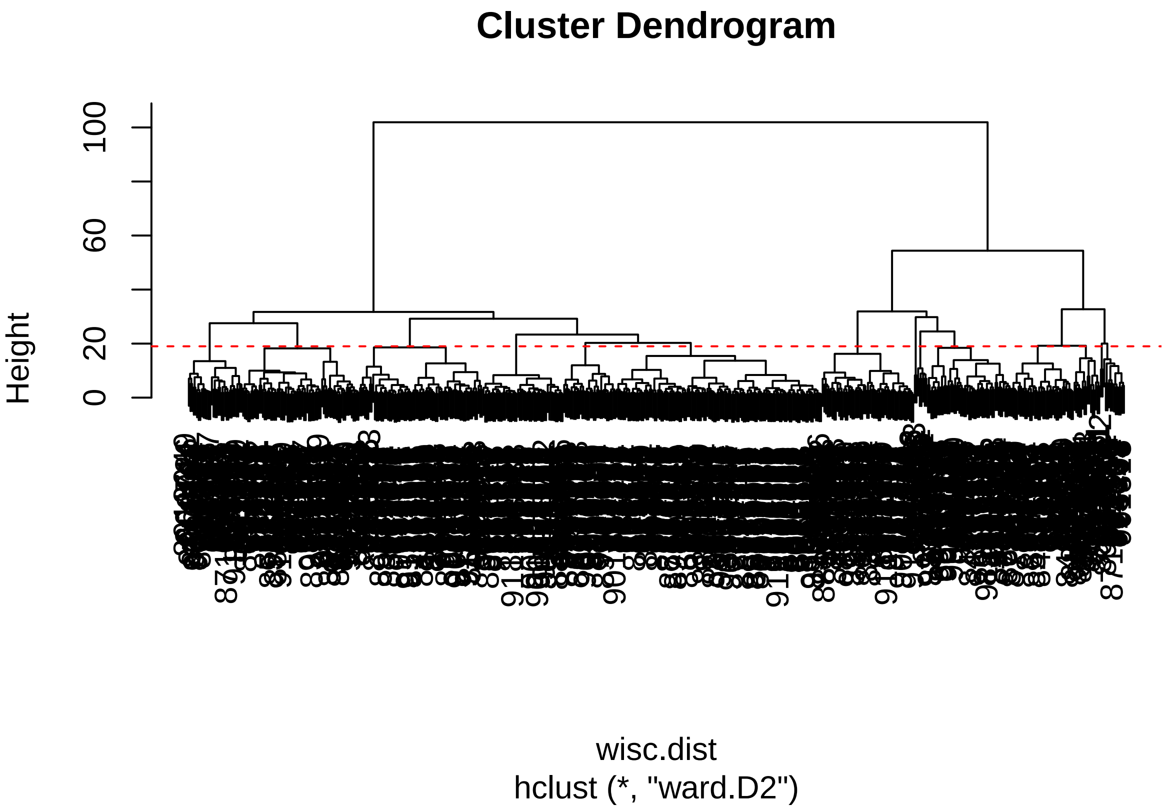
#Q12. No, cutting the tree into any less groups ignores the split between the two large groups of patients, and cutting it into any more groups than 4 unnecessarily divides groups.

```
# Try 'ward.D2' method

# Calculate scaled Euclidean distances of data points
wisc.dist <- dist(scale(wisc.data))

# Cluster w/ 'hclust()'
wisc.hclust <- hclust(wisc.dist, method="ward.D2")

# Visualize
plot(wisc.hclust)
abline(h=19, col="red", lty=2)
```



#Q13. Netiher the 'single' nor the 'average' method come up with any kind of clustering that separates the malignant and benign groups, but I don't think I can pick a favorite between the 'complete' and 'ward.D2' methods. While 'ward.D2' returns satisfactory results with separation into just two clusters, when 'complete' works, it separates more accurately (less erroneous results) than 'ward.D2'.

```
# See if k-means clustering gives same result

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

# Check result with table
table(wisc.km$cluster, diagnosis) #kmeans clustering result
```

```
##    diagnosis
##       B   M
##   1   1 130
##   2 356  82
```

```
table(ans [3,], diagnosis) #hclust result
```

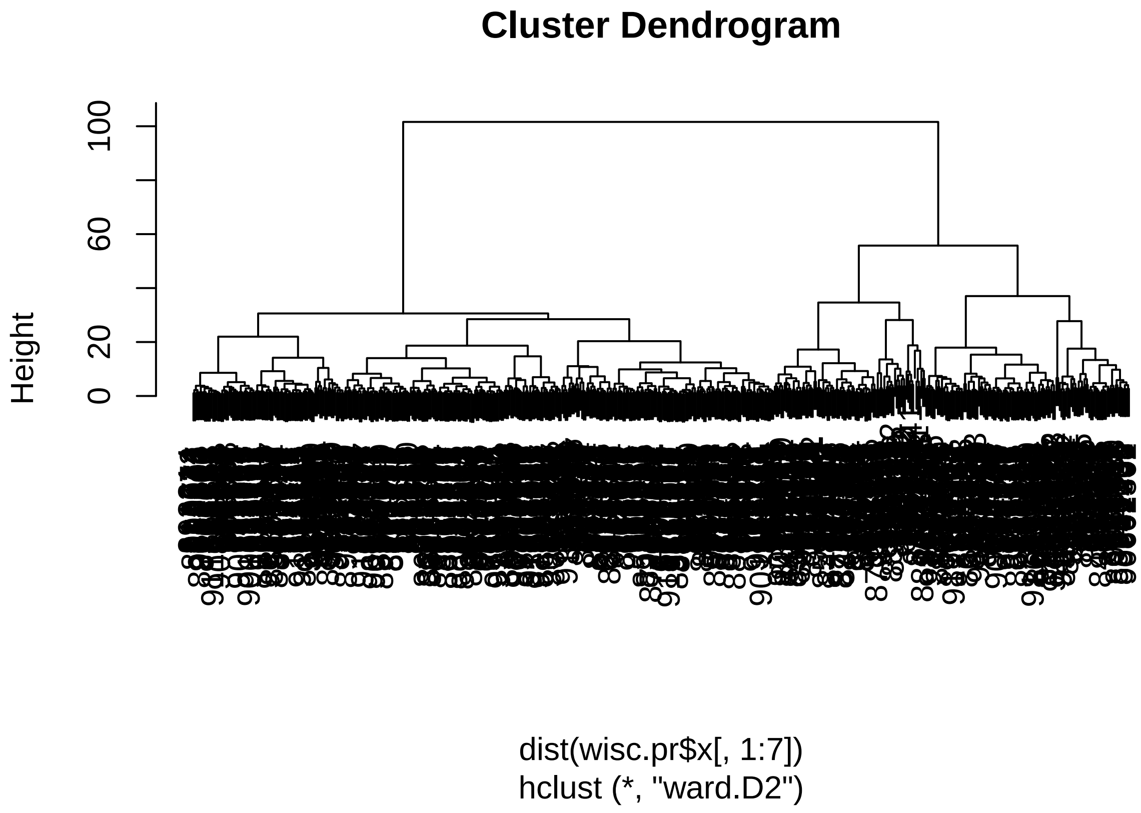
```
##    diagnosis
##       B   M
##   1  12 165
##   2   2   5
##   3 343  40
##   4   0   2
```

#Q14. K-means satisfactorily separates the diagnoses. It is not as accurate as hclust, but it is close behind. There are clearly two groups

Combining Methods

```
# Apply hierarchical clustering to PCA Analysis
wisc.pr.hclust <- hclust(dist(wisc.pr$x[,1:7]), method="ward.D2")

# Visualize
plot(wisc.pr.hclust)
```



```
# Split up the tree into groups
wisc.pr.hclust.cut <- cutree(wisc.pr.hclust, k=2)
table(wisc.pr.hclust.cut, diagnosis)
```

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

#Q15. This new model has an error rate of 9%. It is a slight improvement over the hierarchical clustering we did on the raw data. (52 errors now, over 61 errors previously).

```
# Compare all Results

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

```
##    diagnosis
##       B   M
##   1   1 130
##   2 356  82
```

```
## Hierarchical Clustering on Data
table(ans [3,], diagnosis)
```

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

```
## Hierarchical Clustering on PCA Analysis
table(wisc.pr.hclust.cut, diagnosis)
```

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

#Q16. Accuracy-wise overall, hierarchical clustering on PCA Analysis takes the win. It also identifies the lowest number of false negatives, which makes this the safest method for its applicaiton.

Sensitivity/ Specificity

#Q17. Total Malignant: 212, Total Benign: 357

K-means Clustering

Sensitivity: 130/212 = 61.3%

Specificity: 356/357 = 99.7%

Hierarchical Clustering

Sensitivity: 165/212 = 77.8%

Specificity: 343/357 = 96.1%

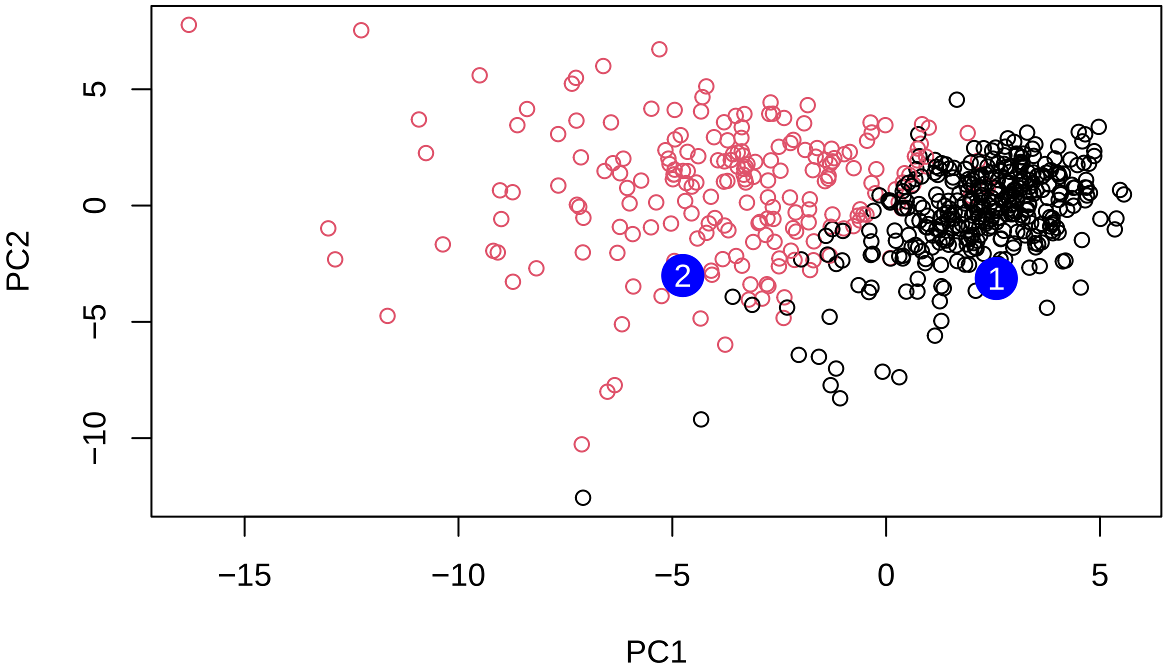
Hierarchical Clustering on Principal Components

Sensitivity: 188/212 = 88.7%

Specificity: 329/357 = 92.2%

```
# Import New Data Points
new <- read.csv("new_samples.csv")
npc <- predict(wisc.pr, newdata=new)

# Plot Old Data with Two New Points Overlay
plot(wisc.pr$x[,1:2], col=factor_diagnosis)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```



#Q18. Patient 2 should be prioritized as they probably have the malignant tumor.