Mini-Project

Brittney Hayes

1. Preparing the data

```
#View(WisconsinCancer)

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

# Complete the following code to input the data and store as wisc.df
wisc.df <- read.csv(fna.data, row.names = 1)

 # View the first few rows of the dataframe
head(wisc.df)
```

```
                 X         X.1          X.2            X.3       X.4
id       diagnosis radius_mean texture_mean perimeter_mean area_mean
842302           M       17.99        10.38          122.8      1001
842517           M       20.57        17.77          132.9      1326
84300903         M       19.69        21.25            130      1203
84348301         M       11.42        20.38          77.58     386.1
84358402         M       20.29        14.34          135.1      1297
                     X.5              X.6            X.7                  X.8
id       smoothness_mean compactness_mean concavity_mean concave points_mean
842302            0.1184           0.2776         0.3001               0.1471
842517           0.08474          0.07864         0.0869              0.07017
84300903          0.1096           0.1599         0.1974               0.1279
84348301          0.1425           0.2839         0.2414               0.1052
84358402          0.1003           0.1328          0.198               0.1043
                   X.9                   X.10      X.11       X.12         X.13
id       symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
842302          0.2419                0.07871     1.095     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      1.156         3.445
84358402        0.1809                0.05883    0.7572     0.7813         5.438
            X.14          X.15            X.16         X.17             X.18
id       area_se smoothness_se compactness_se concavity_se concave points_se
842302     153.4      0.006399        0.04904      0.05373           0.01587
842517     74.08      0.005225        0.01308       0.0186            0.0134
84300903   94.03       0.00615        0.04006      0.03832           0.02058
84348301   27.23       0.00911        0.07458      0.05661           0.01867
84358402   94.44       0.01149        0.02461      0.05688           0.01885
                X.19                 X.20         X.21          X.22
id       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      0.0225             0.004571        23.57         25.53
84348301     0.05963             0.009208        14.91          26.5
84358402     0.01756             0.005115        22.54         16.67
                    X.23       X.24             X.25              X.26
id       perimeter_worst area_worst smoothness_worst compactness_worst
842302             184.6       2019           0.1622            0.6656
842517             158.8       1956           0.1238            0.1866
84300903           152.5       1709           0.1444            0.4245
84348301           98.87      567.7           0.2098            0.8663
84358402           152.2       1575           0.1374             0.205
                    X.27                 X.28           X.29
id       concavity_worst concave points_worst symmetry_worst
842302            0.7119               0.2654         0.4601
842517            0.2416                0.186          0.275
84300903          0.4504                0.243         0.3613
84348301          0.6869               0.2575         0.6638
84358402             0.4               0.1625         0.2364
                            X.30
id       fractal_dimension_worst
842302                    0.1189
842517                   0.08902
84300903                 0.08758
84348301                   0.173
84358402                 0.07678
```

```r
# We can use -1 here to remove the first column
wisc.data <- wisc.df[,-1]

# Create diagnosis vector for later
diagnosis <- wisc.df$X
```

1. Exploratory Data Analysis

Q1. How many observations are in this dataset?

```r
num_observations <- nrow(wisc.data)
```

Answer: 570

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

```r
num_malignant <- sum(diagnosis == "M")
```

Answer: 212

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

```r
num_mean_variables <- sum(grep("_mean", names(wisc.data)))
```

Answer: 0

2. Principal Component Analysis

```r
str(wisc.data)
```

```
'data.frame':   570 obs. of  30 variables:
 $ X.1 : chr  "radius_mean" "17.99" "20.57" "19.69" ...
 $ X.2 : chr  "texture_mean" "10.38" "17.77" "21.25" ...
 $ X.3 : chr  "perimeter_mean" "122.8" "132.9" "130" ...
 $ X.4 : chr  "area_mean" "1001" "1326" "1203" ...
 $ X.5 : chr  "smoothness_mean" "0.1184" "0.08474" "0.1096" ...
 $ X.6 : chr  "compactness_mean" "0.2776" "0.07864" "0.1599" ...
 $ X.7 : chr  "concavity_mean" "0.3001" "0.0869" "0.1974" ...
 $ X.8 : chr  "concave points_mean" "0.1471" "0.07017" "0.1279" ...
 $ X.9 : chr  "symmetry_mean" "0.2419" "0.1812" "0.2069" ...
 $ X.10: chr  "fractal_dimension_mean" "0.07871" "0.05667" "0.05999" ...
 $ X.11: chr  "radius_se" "1.095" "0.5435" "0.7456" ...
 $ X.12: chr  "texture_se" "0.9053" "0.7339" "0.7869" ...
 $ X.13: chr  "perimeter_se" "8.589" "3.398" "4.585" ...
 $ X.14: chr  "area_se" "153.4" "74.08" "94.03" ...
 $ X.15: chr  "smoothness_se" "0.006399" "0.005225" "0.00615" ...
 $ X.16: chr  "compactness_se" "0.04904" "0.01308" "0.04006" ...
 $ X.17: chr  "concavity_se" "0.05373" "0.0186" "0.03832" ...
 $ X.18: chr  "concave points_se" "0.01587" "0.0134" "0.02058" ...
 $ X.19: chr  "symmetry_se" "0.03003" "0.01389" "0.0225" ...
 $ X.20: chr  "fractal_dimension_se" "0.006193" "0.003532" "0.004571" ...
 $ X.21: chr  "radius_worst" "25.38" "24.99" "23.57" ...
 $ X.22: chr  "texture_worst" "17.33" "23.41" "25.53" ...
 $ X.23: chr  "perimeter_worst" "184.6" "158.8" "152.5" ...
 $ X.24: chr  "area_worst" "2019" "1956" "1709" ...
 $ X.25: chr  "smoothness_worst" "0.1622" "0.1238" "0.1444" ...
 $ X.26: chr  "compactness_worst" "0.6656" "0.1866" "0.4245" ...
 $ X.27: chr  "concavity_worst" "0.7119" "0.2416" "0.4504" ...
 $ X.28: chr  "concave points_worst" "0.2654" "0.186" "0.243" ...
 $ X.29: chr  "symmetry_worst" "0.4601" "0.275" "0.3613" ...
 $ X.30: chr  "fractal_dimension_worst" "0.1189" "0.08902" "0.08758" ...
```

```r
# Had issues "Error in colMeans(wisc.data) : 'x' must be numeric" Converting to numeric (i
wisc.data$X.1<- as.numeric(as.factor(wisc.data$X.1))
wisc.data$X.2<- as.numeric(as.factor(wisc.data$X.2))
wisc.data$X.3<- as.numeric(as.factor(wisc.data$X.3))
wisc.data$X.4<- as.numeric(as.factor(wisc.data$X.4))
wisc.data$X.5<- as.numeric(as.factor(wisc.data$X.5))
wisc.data$X.6<- as.numeric(as.factor(wisc.data$X.6))
wisc.data$X.7<- as.numeric(as.factor(wisc.data$X.7))
wisc.data$X.8<- as.numeric(as.factor(wisc.data$X.8))
wisc.data$X.9<- as.numeric(as.factor(wisc.data$X.9))
wisc.data$X.10<- as.numeric(as.factor(wisc.data$X.10))
wisc.data$X.11<- as.numeric(as.factor(wisc.data$X.11))
wisc.data$X.12<- as.numeric(as.factor(wisc.data$X.12))
wisc.data$X.13<- as.numeric(as.factor(wisc.data$X.13))
wisc.data$X.14<- as.numeric(as.factor(wisc.data$X.14))
wisc.data$X.15<- as.numeric(as.factor(wisc.data$X.15))
wisc.data$X.16<- as.numeric(as.factor(wisc.data$X.16))
wisc.data$X.17<- as.numeric(as.factor(wisc.data$X.17))
wisc.data$X.18<- as.numeric(as.factor(wisc.data$X.18))
wisc.data$X.19<- as.numeric(as.factor(wisc.data$X.19))
wisc.data$X.20<- as.numeric(as.factor(wisc.data$X.20))
wisc.data$X.21<- as.numeric(as.factor(wisc.data$X.21))
wisc.data$X.22<- as.numeric(as.factor(wisc.data$X.22))
wisc.data$X.23<- as.numeric(as.factor(wisc.data$X.23))
wisc.data$X.24<- as.numeric(as.factor(wisc.data$X.24))
wisc.data$X.25<- as.numeric(as.factor(wisc.data$X.25))
wisc.data$X.26<- as.numeric(as.factor(wisc.data$X.26))
wisc.data$X.27<- as.numeric(as.factor(wisc.data$X.27))
wisc.data$X.28<- as.numeric(as.factor(wisc.data$X.28))
wisc.data$X.29<- as.numeric(as.factor(wisc.data$X.29))
wisc.data$X.30<- as.numeric(as.factor(wisc.data$X.30))

# Check column means and standard deviations
apply(wisc.data,2,sd)
```

```
     X.1      X.2      X.3      X.4      X.5      X.6      X.7      X.8 
129.7166 133.6683 151.6142 155.2636 136.4445 154.3905 158.5430 159.7768 
     X.9     X.10     X.11     X.12     X.13     X.14     X.15     X.16 
118.4978 141.1332 155.3676 145.5543 152.5752 151.1658 157.6017 154.7020 
    X.17     X.18     X.19     X.20     X.21     X.22     X.23     X.24 
155.7535 146.3378 140.5687 156.2770 127.3364 145.1813 150.9367 157.8110 
    X.25     X.26     X.27     X.28     X.29     X.30 
109.8857 151.2856 157.8129 144.4117 142.2252 154.0813 
```

```r
numeric_data <- wisc.data[, c(1:30)]
means <- colMeans(numeric_data)
```

```r
# Perform PCA on wisc.data (now numeric_data)
wisc.pr <- prcomp(numeric_data, scale. = TRUE)

# Look at summary of results
summary(wisc.pr)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     3.4186 1.8552 1.72721 1.40694 1.31863 1.16598 1.06146
Proportion of Variance 0.3896 0.1147 0.09944 0.06598 0.05796 0.04532 0.03756
Cumulative Proportion  0.3896 0.5043 0.60372 0.66971 0.72767 0.77298 0.81054
                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
```

```
Standard deviation     0.97046 0.86234 0.82617 0.70804 0.68580 0.62796 0.57121
Proportion of Variance 0.03139 0.02479 0.02275 0.01671 0.01568 0.01314 0.01088
Cumulative Proportion  0.84193 0.86672 0.88947 0.90618 0.92186 0.93501 0.94588
                          PC15    PC16    PC17    PC18    PC19   PC20   PC21
Standard deviation     0.52326 0.48091 0.46611 0.45725 0.40475 0.3754 0.3332
Proportion of Variance 0.00913 0.00771 0.00724 0.00697 0.00546 0.0047 0.0037
Cumulative Proportion  0.95501 0.96272 0.96996 0.97693 0.98239 0.9871 0.9908
                          PC22   PC23    PC24    PC25    PC26    PC27    PC28
Standard deviation     0.23327 0.2191 0.20233 0.19205 0.18114 0.16743 0.13202
Proportion of Variance 0.00181 0.0016 0.00136 0.00123 0.00109 0.00093 0.00058
Cumulative Proportion  0.99260 0.9942 0.99556 0.99679 0.99789 0.99882 0.99940
                          PC29    PC30
Standard deviation     0.10866 0.07828
Proportion of Variance 0.00039 0.00020
Cumulative Proportion  0.99980 1.00000
```

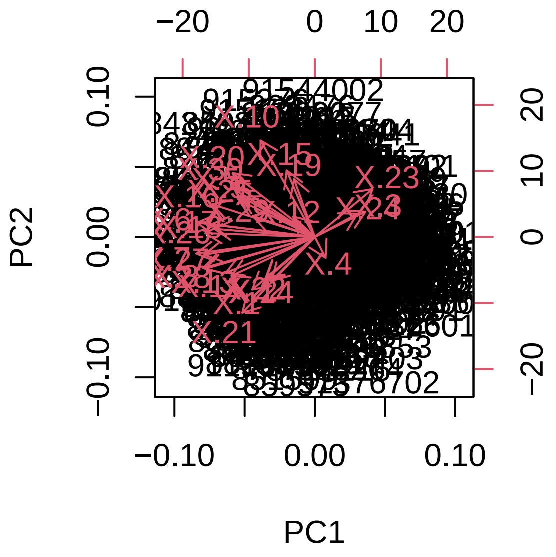
Q4. From your results, what proportion of the original variance is captured by the first principal components (PC1)? PC1 captures approximately 38.96% of the original variance.

Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data? 2 PCs are required to describe at least 70% of the original variance in the data.

Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data? 7 PCs are required to describe at least 90% of the original variance in the data.

2. Interpreting PCA Results

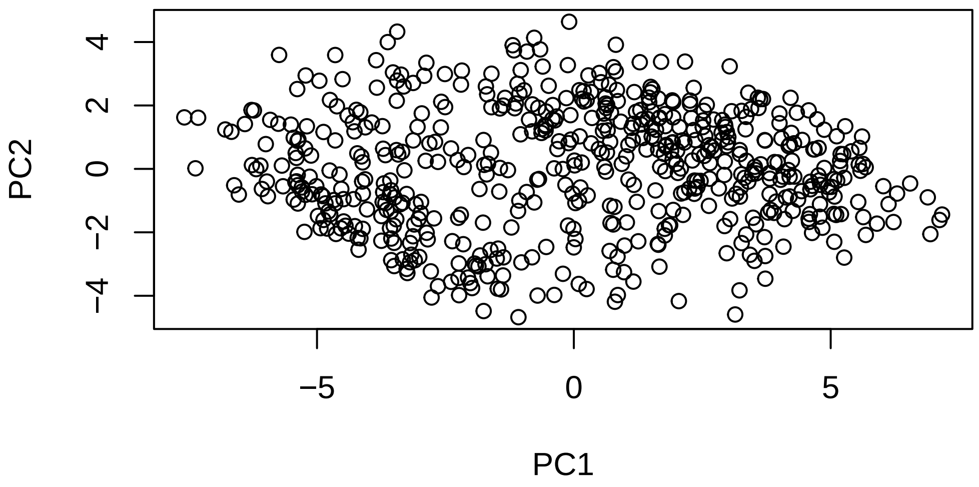
```
biplot(wisc.pr)
```



Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why? The center of the plot stands out to me only because it is so messy. It is difficult to understand given how cluttered it is.

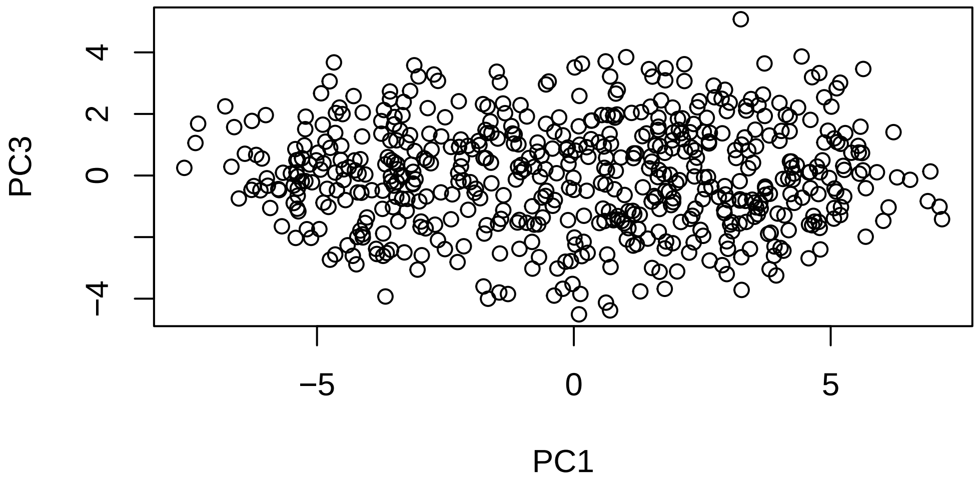
```r
# Scatter plot observations by components 1 and 2

plot(wisc.pr$x[, c(1, 2)],
     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[, c(1, 3)],
     xlab = "PC1", ylab = "PC3")
```



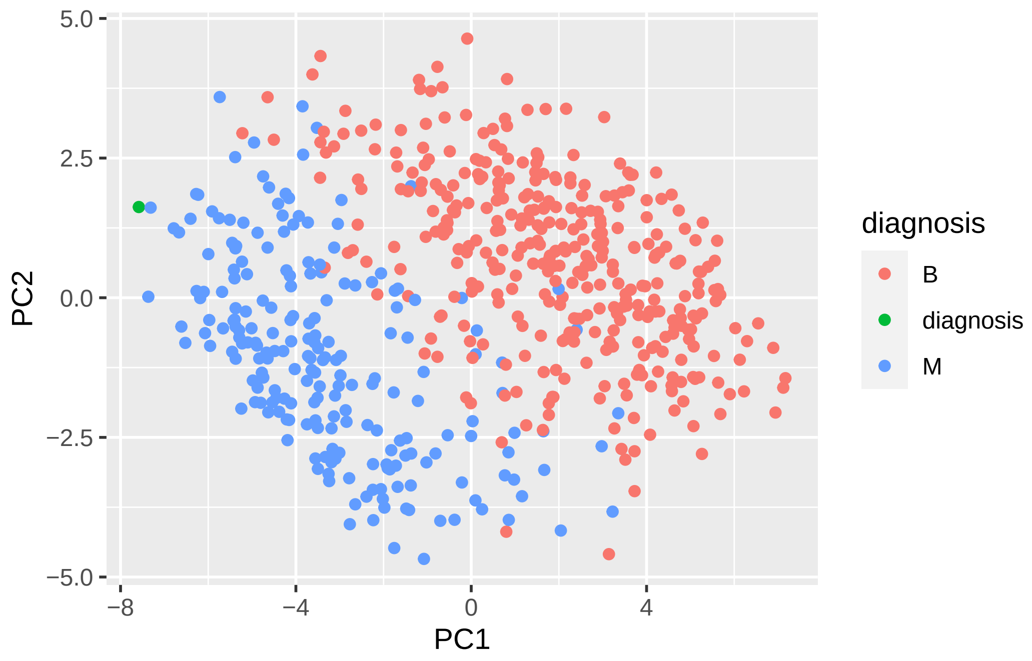
Answer: I was having issues with col=diagnosis so I omitted it in these plots (Error: unexpected symbol in: "plot(wisc.pr$x[, c(1, 3)], col = diagnosis xlab") & invalid color name 'diagnosis') but I notice that PC2 has more variance than PC3.

2.Variance Explained

```
# 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()
```



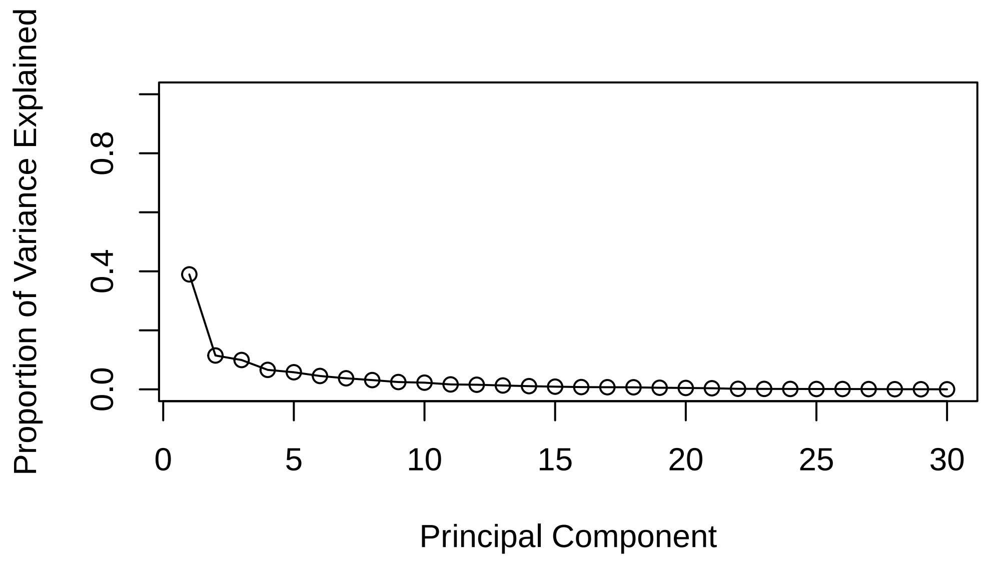
```r
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

```
[1] 11.686620  3.441871  2.983241  1.979494  1.738785  1.359503
```

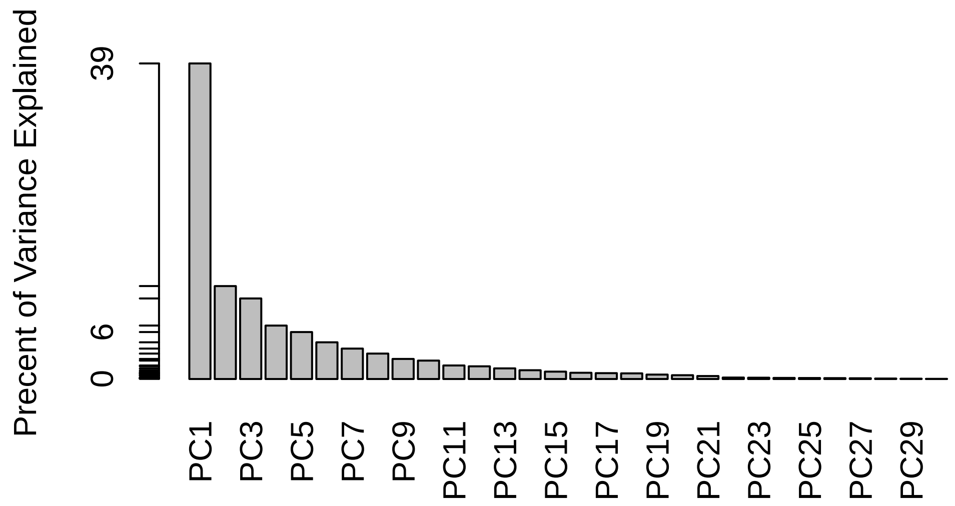
```r
# Calculate total variance explained
total_var <- sum(pr.var)

# Variance explained by each principal component: pve
pve <- pr.var / total_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 )
```



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?

```
# Loading vector component for feature concave.points_mean
loading_component <- wisc.pr$rotation[, 1]
loading_component
```

```
        X.1         X.2         X.3         X.4         X.5         X.6 
-0.14492806 -0.11319439  0.11734576  0.02507453 -0.18373394 -0.27308944 
        X.7         X.8         X.9        X.10        X.11        X.12 
-0.27475507 -0.25905963 -0.17075393 -0.12574144 -0.19310926 -0.05008037 
       X.13        X.14        X.15        X.16        X.17        X.18 
-0.20612259 -0.10000261 -0.06523108 -0.23750442 -0.24022059 -0.23232505 
       X.19        X.20        X.21        X.22        X.23        X.24 
-0.04820717 -0.19059524 -0.16802634 -0.11383556  0.13378332  0.09849681 
       X.25        X.26        X.27        X.28        X.29        X.30 
-0.17625086 -0.25367150 -0.25621658 -0.25391455 -0.14386065 -0.19384792 
```

Answer: X.8 is concave points mean, -0.25905963.

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

```r
# Cumulative proportion of variance explained
cumulative_pve <- cumsum(pve)

# Minimum number of principal components to explain 80% of the variance
min_components <- which.max(cumulative_pve >= 0.8)
```

Answer: 7

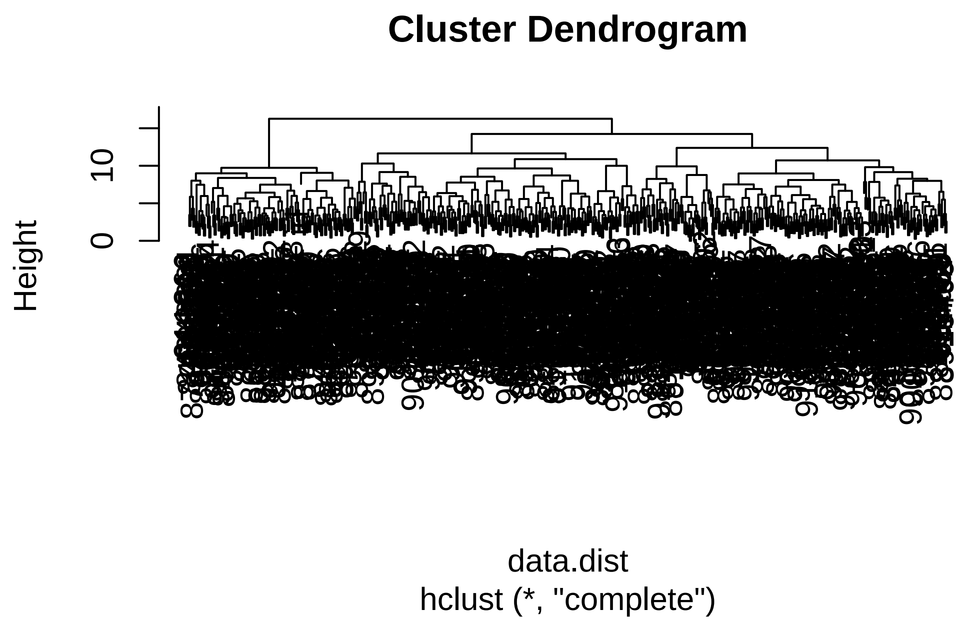
3. Hierarchical Clustering

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

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


# Create a hierarchical clustering model using complete linkage
wisc.hclust <- hclust(data.dist, method = "complete")
```

```r
# Plot the dendrogram
plot(wisc.hclust)
abline( col = "red", lty = 2)
```



Q11. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters? Around 10 according to my plot. However the plot on the website is different and is around 19.

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

  # Compare the cluster membership to the actual diagnoses.

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

```
                    diagnosis
wisc.hclust.clusters   B diagnosis   M
                   1  11         1 114
                   2   3         0  53
                   3 173         0  40
                   4 170         0   5
```

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

```
# Cut the tree into 10 clusters
  wisc.hclust.clusters <- cutree(wisc.hclust, k = 2)

  # Compare the cluster membership to the actual diagnoses.

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

```
                    diagnosis
wisc.hclust.clusters   B diagnosis   M
                   1  11         1 114
                   2 346         0  98
```

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

```
# Create hierarchical clustering models using different methods
wisc.hclust_single <- hclust(data.dist, method = "single")
wisc.hclust_complete <- hclust(data.dist, method = "complete")
wisc.hclust_average <- hclust(data.dist, method = "average")
wisc.hclust_ward <- hclust(data.dist, method = "ward.D2")

# Cut the trees into clusters (e.g., let's use 4 clusters for comparison)
wisc.hclust_single_clusters <- cutree(wisc.hclust_single, k = 4)
wisc.hclust_complete_clusters <- cutree(wisc.hclust_complete, k = 4)
wisc.hclust_average_clusters <- cutree(wisc.hclust_average, k = 4)
wisc.hclust_ward_clusters <- cutree(wisc.hclust_ward, k = 4)

# Compare cluster vs. diagnosis match for each method
print("Single linkage:")
```

```
[1] "Single linkage:"
```

```
print(table(wisc.hclust_single_clusters, diagnosis))
```

```
                           diagnosis
wisc.hclust_single_clusters   B diagnosis   M
                          1   0         1   0
                          2 355         0 211
                          3   0         0   1
                          4   2         0   0
```

```
print("\nComplete linkage:")
```

```
[1] "\nComplete linkage:"
```

```
print(table(wisc.hclust_complete_clusters, diagnosis))
```

```
                            diagnosis
wisc.hclust_complete_clusters   B diagnosis   M
                           1  11          1 114
                           2   3          0  53
                           3 173          0  40
                           4 170          0   5
```

```
print("\nAverage linkage:")
```

```
[1] "\nAverage linkage:"
```

```
print(table(wisc.hclust_average_clusters, diagnosis))
```

```
                           diagnosis
wisc.hclust_average_clusters   B diagnosis   M
                          1   0          1   0
                          2  19          0 182
                          3 331          0  29
                          4   7          0   1
```

```
print("\nWard linkage:")
```

```
[1] "\nWard linkage:"
```

```
print(table(wisc.hclust_ward_clusters, diagnosis))
```

```
                          diagnosis
wisc.hclust_ward_clusters   B diagnosis   M
                        1   1         1 177
                        2  58         0  30
                        3 165         0   3
                        4 133         0   2
```

I like the single clusters method most because the numbers are easier to look at.

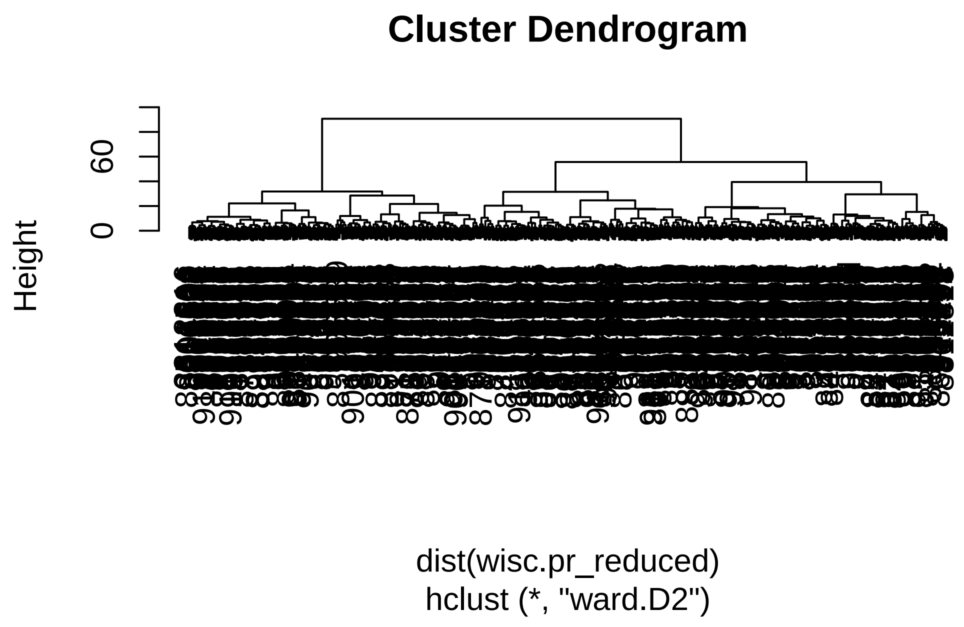
5. Combining Methods

```
# Find the minimum number of principal components required to describe at least 90% of the
cumulative_variance <- cumsum(wisc.pr$sdev^2 / sum(wisc.pr$sdev^2))
num_components_90 <- which(cumulative_variance >= 0.9)[1]

# Use the first num_components_90 principal components
wisc.pr_reduced <- wisc.pr$x[, 1:num_components_90]

# Create hierarchical clustering model with ward.D2 linkage method
wisc.pr.hclust <- hclust(dist(wisc.pr_reduced), method = "ward.D2")

plot(wisc.pr.hclust)
```



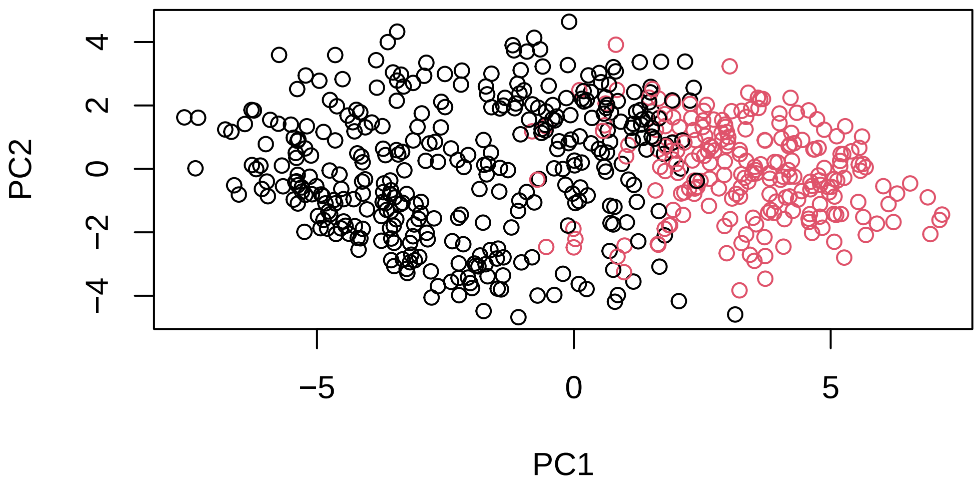
```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
  1   2 
352 218 
```

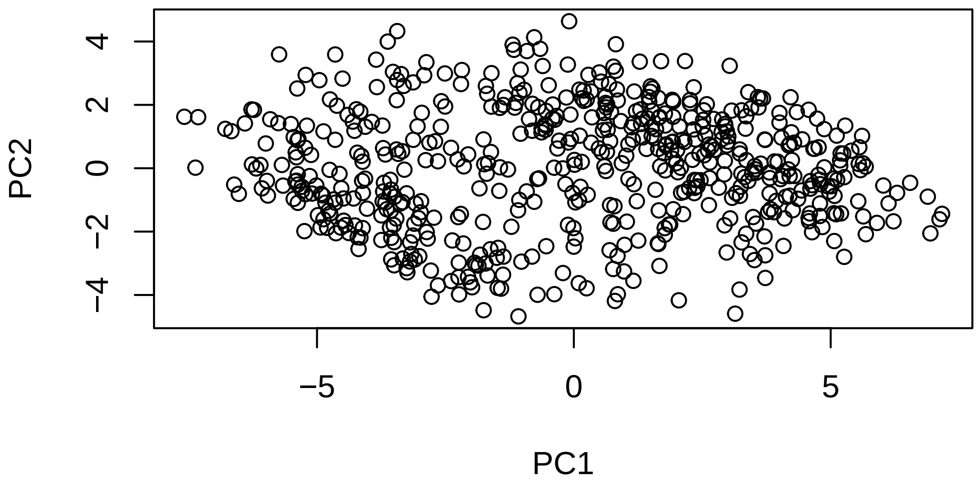
```
table(grps, diagnosis)
```

```
    diagnosis
grps   B diagnosis   M
   1 151         1 200
   2 206         0  12
```

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



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



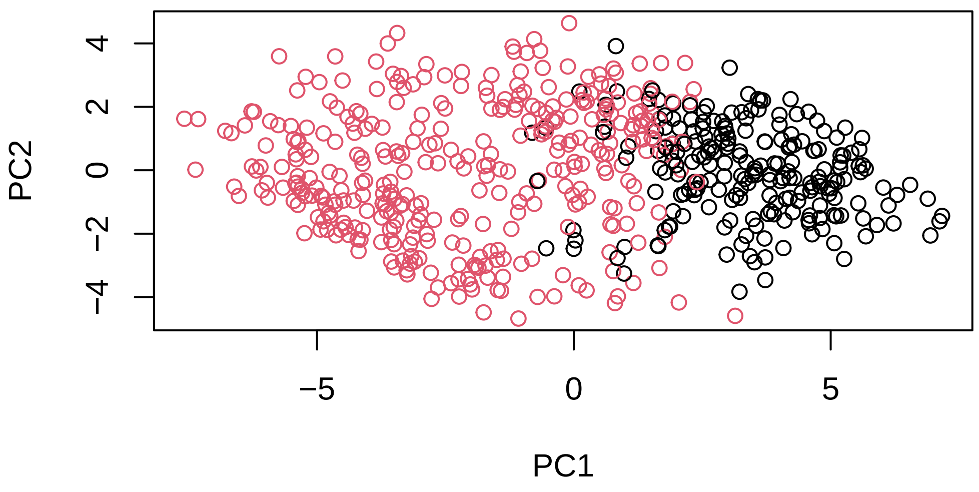
```r
g <- as.factor(grps)
levels(g)
```

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

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

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

```r
# Plot using our re-ordered factor
plot(wisc.pr$x[,1:2], col=g)
```



```r
# Use the distance along the first 7 PCs for clustering
wisc.pr.hclust <- hclust(dist(wisc.pr$x[, 1:7]), method = "ward.D2")

# Cut the hierarchical clustering model into 2 clusters
wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k = 2)
```

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

```r
# Compare the results from the new hierarchical clustering model with the actual diagnoses
table(wisc.pr.hclust.clusters, diagnosis)
```

```
                       diagnosis
wisc.pr.hclust.clusters   B diagnosis   M
                      1  16         1 185
                      2 341         0  27
```

The diagnoses are separated better in this model.

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.

```r
hclust_diagnosis_table <-table(wisc.hclust.clusters, diagnosis)
```

“Error: object ‘wisc.km’ not found.

6. Sensitivity/Specificity

```r
# Function to calculate specificity
calculate_specificity <- function(confusion_matrix) {
  true_negatives <- confusion_matrix[1, 1]
  false_positives <- confusion_matrix[1, 2]
  return(true_negatives / (true_negatives + false_positives))
}

# Function to calculate sensitivity
calculate_sensitivity <- function(confusion_matrix) {
  true_positives <- confusion_matrix[2, 2]
  false_negatives <- confusion_matrix[2, 1]
  return(true_positives / (true_positives + false_negatives))
}

# Calculate specificity and sensitivity for each clustering model
```

```r
hclust_specificity <- calculate_specificity(hclust_diagnosis_table)
hclust_sensitivity <- calculate_sensitivity(hclust_diagnosis_table)

# Display the results
print(paste("Hierarchical clustering specificity:", hclust_specificity))
```

```
[1] "Hierarchical clustering specificity: 0.916666666666667"
```

```r
print(paste("Hierarchical clustering sensitivity:", hclust_sensitivity))
```

```
[1] "Hierarchical clustering sensitivity: 0"
```

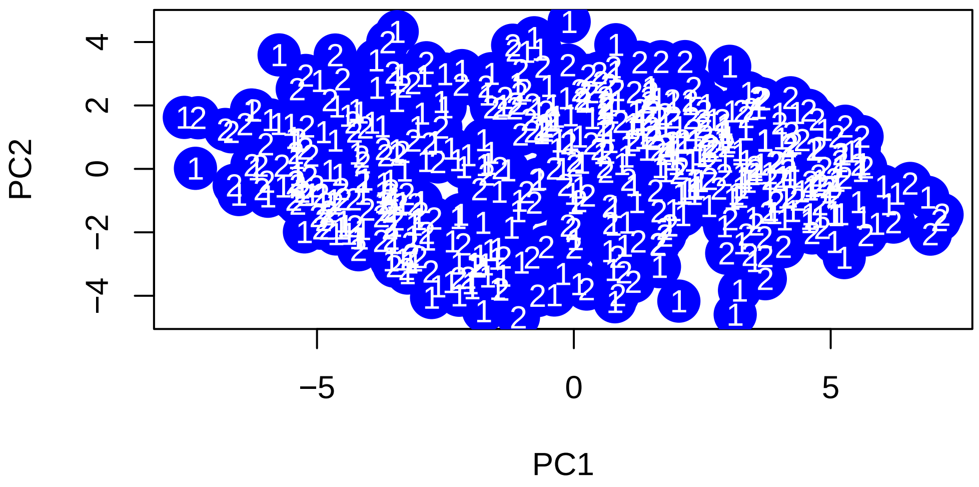
I keep getting errors and cannot figure them out. But based on what is on the website, I would say the second model is better at separating the diagnoses.

Q17. Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity? The second model is more specific and the second is more sensitive.

7. Prediction

```r
url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr)
```

```r
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? My plot came out too messy. Based on one on website I would say patients 1 and 2.