Week 5 Mini Project

David Alvarez

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

Q1. How many observations are in this dataset?

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

```
[1] 569  30
```

```
# Can be seen that there are 569 observations in this dataset
```

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

```
table(wisc.df$diagnosis)
```

```
  B   M
357 212
```

```
# There are 212 M diagnoses
```

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

```
variables <- colnames(wisc.df)
variables_with_mean <- grep("_mean", variables)
length(variables_with_mean)
```

```
[1] 10
```

Q4. From the results below, what proportion of the original variance is captured by PC1?

```
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
```

```
## The proportion of original variance captured by PC1 seems to be around 0.44 or 44%
```

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

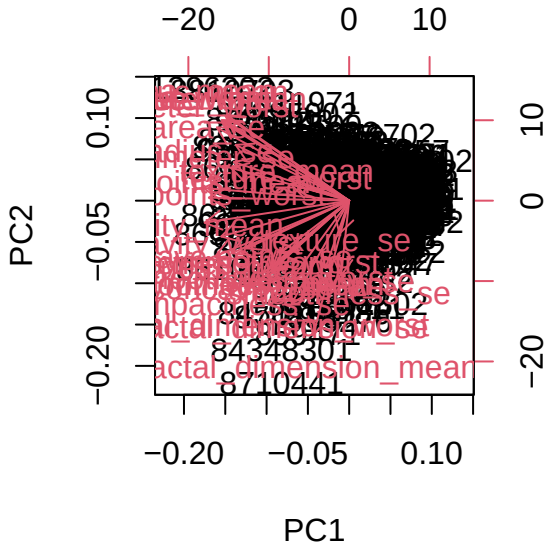
```
# To describe at least 70% of the original variance in the data, there are about 3 PCs req
```

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

```
# To describe at least 90% of the original variance in the data, at least around 7 PCs.
```

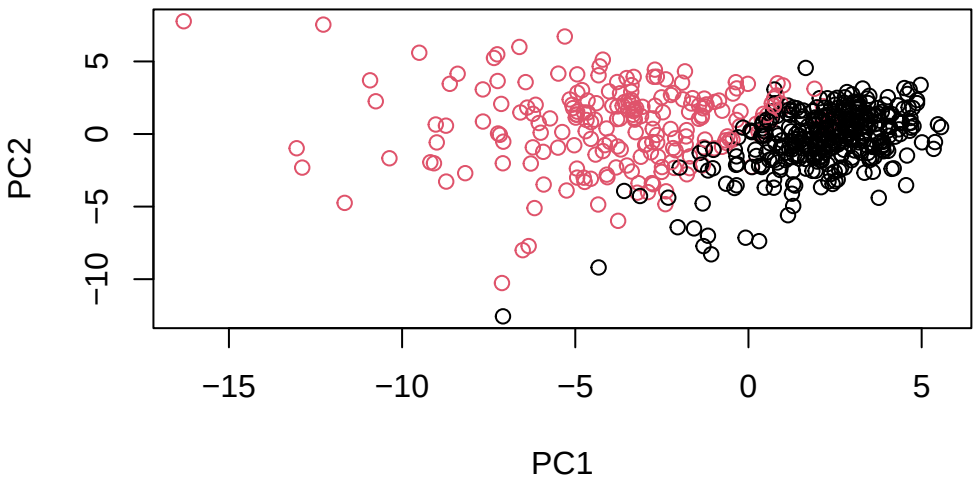
Q7. What stands out about the plot below? Is it easy or diffiucult to understand?

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



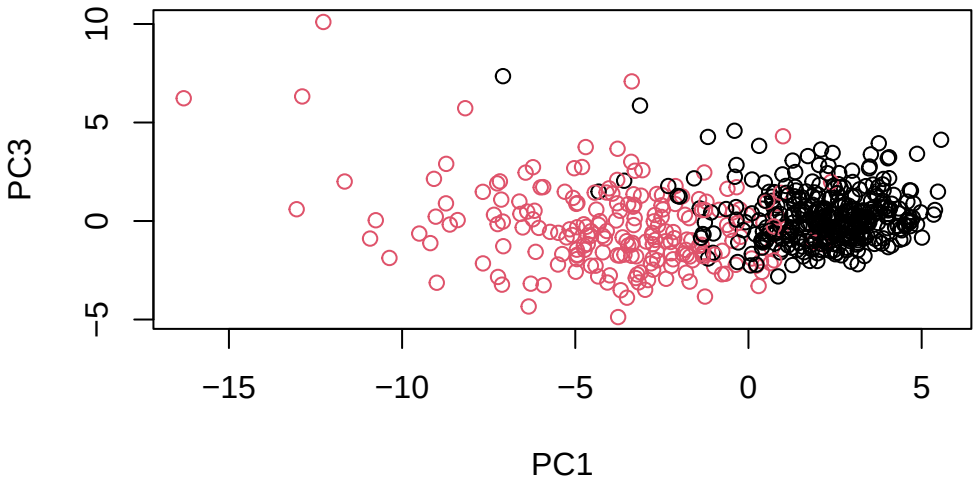

```
# There are too many things that stand out about this plot, such as it being a mix of colu
```

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



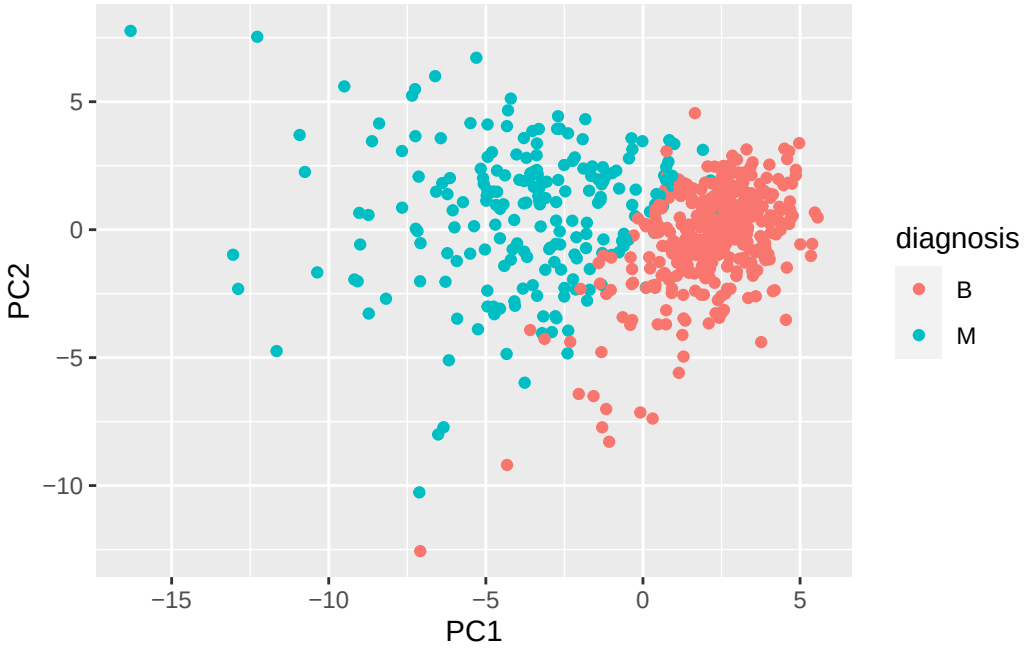
Q8. Generate a plot for principal components 1 and 3. What do you notice?

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



```
# I notice from these plots that they distinguish the points by diagnosis colors now quite
```

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

library(ggplot2)

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

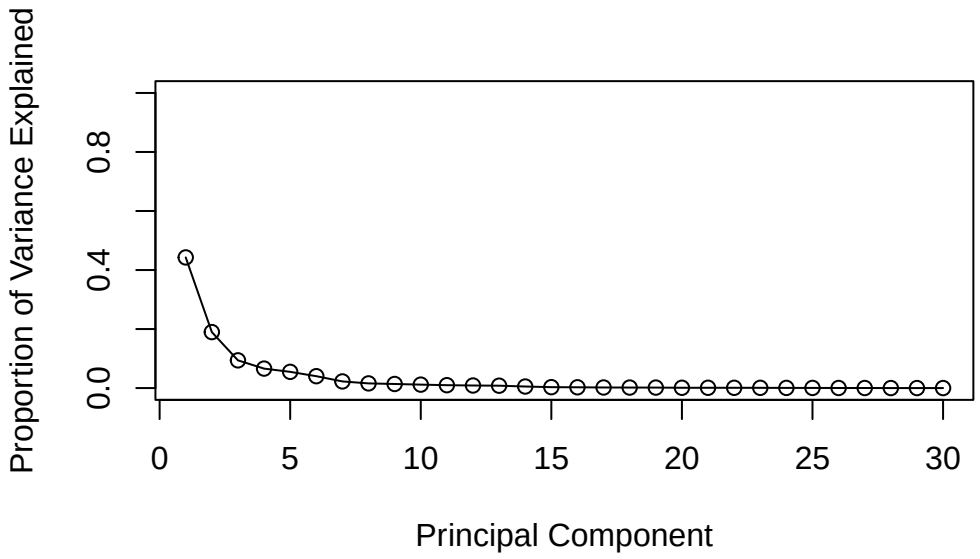



```
# Variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

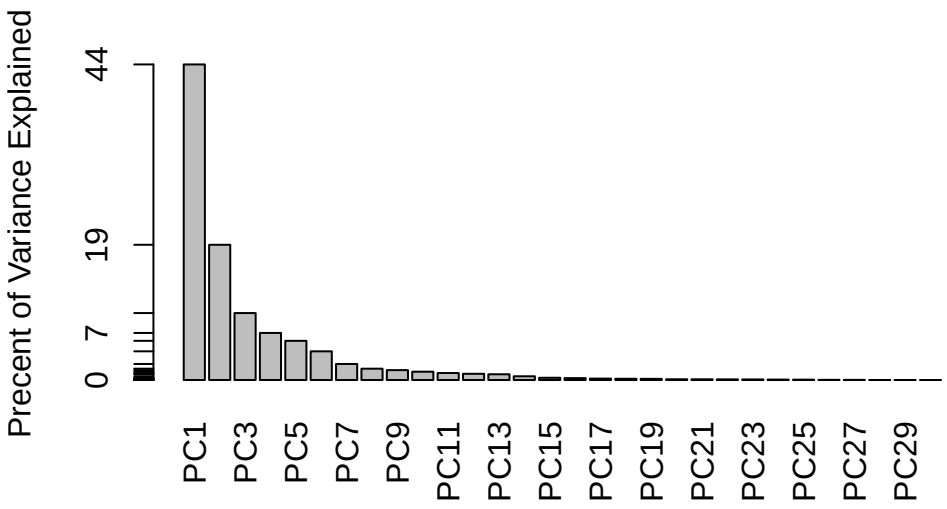
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
# Variance explained by each principal component

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

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




```
# Alternative screen plot of the same data

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 )
```

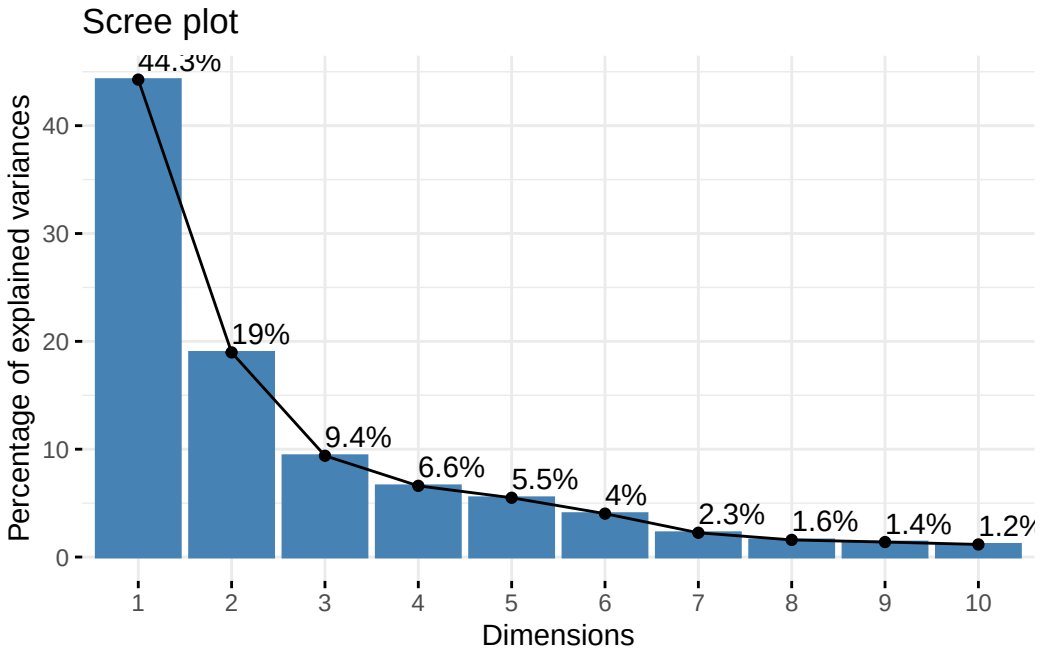



```
## ggplot based graph
library(factoextra)
```

```
Warning: package 'factoextra' was built under R version 4.3.2
```

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

```
fviz_eig(wisc.pr, addlabels = TRUE)
```



Q9. For the first principal component, what is the component of the loading vector for the feature 'concave.points_mean'?

```
pca_component <- wisc.pr$rotation[,1]
concave_points_mean_component <- pca_component["concave.points_mean"]
concave_points_mean_component
```

```
concave.points_mean
         -0.2608538
```

```
# Above is written out how to find the value for the feature 'concave.points_mean' and the
```

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
```

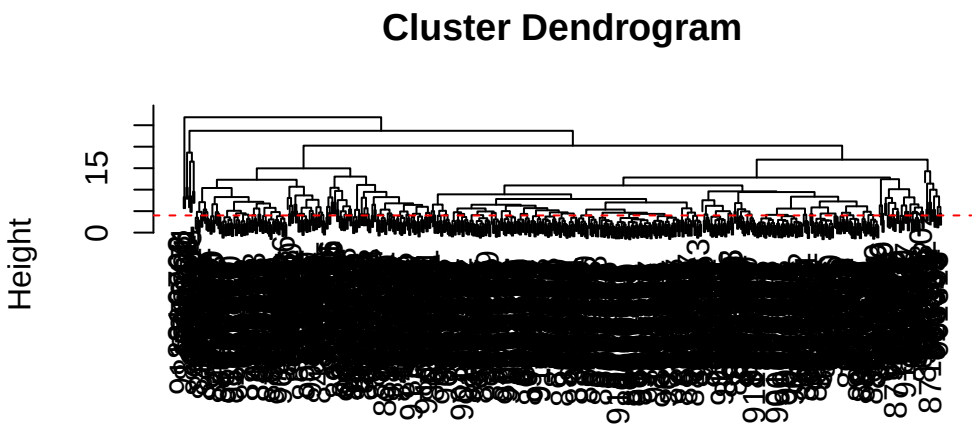
```
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
                           PC8    PC9    PC10   PC11    PC12    PC13    PC14
Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 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
```

```
# From looking at the data, to explain 80% of the variance, it would take a minimum of 5 p
```

```
# Scaling the 'wisc.data'

data.scaled <- scale(wisc.data)

data.dist <- dist(data.scaled)

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

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

```
plot(wisc.hclust)
abline(h=4, col="red", lty=2)
```




```
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?

```
wisc.hclust.clusters2 <- cutree(wisc.hclust, k=6)

table(wisc.hclust.clusters2, diagnosis)
```

```
                    diagnosis
wisc.hclust.clusters2   B   M
                   1  12 165
```

```
2   0   5
3 331  39
4   2   0
5  12   1
6   0   2
```

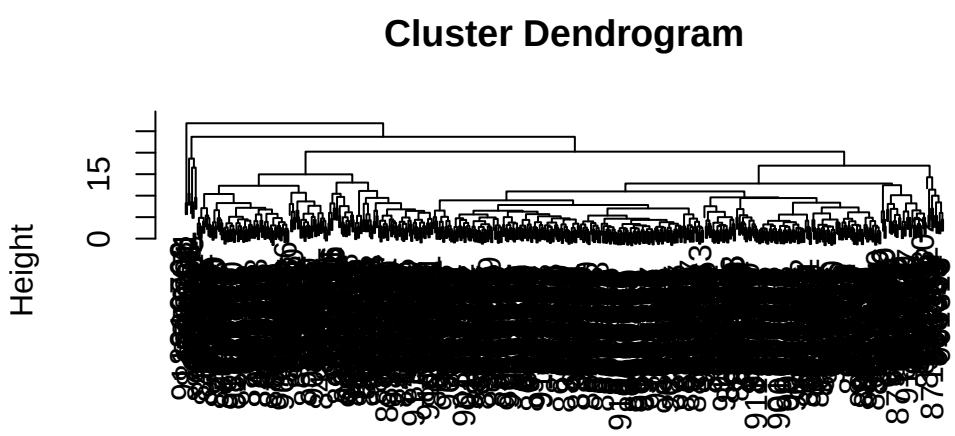
```
# By cutting them into different numbers of clusters between 2 and 10, there was not too b
```

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

```
hc.complete <- hclust(data.dist, method="complete")
hc.complete
```

```
Call:
hclust(d = data.dist, method = "complete")

Cluster method   : complete
Distance         : euclidean
Number of objects: 569
```

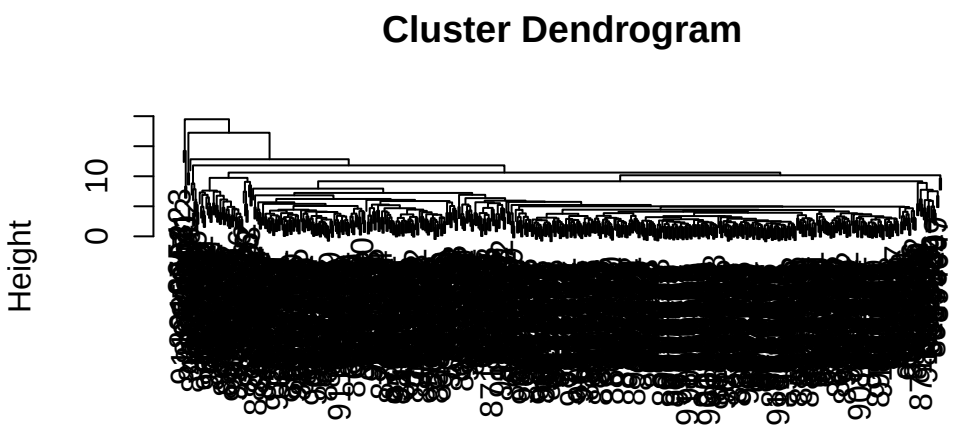
```
plot(hc.complete)
```



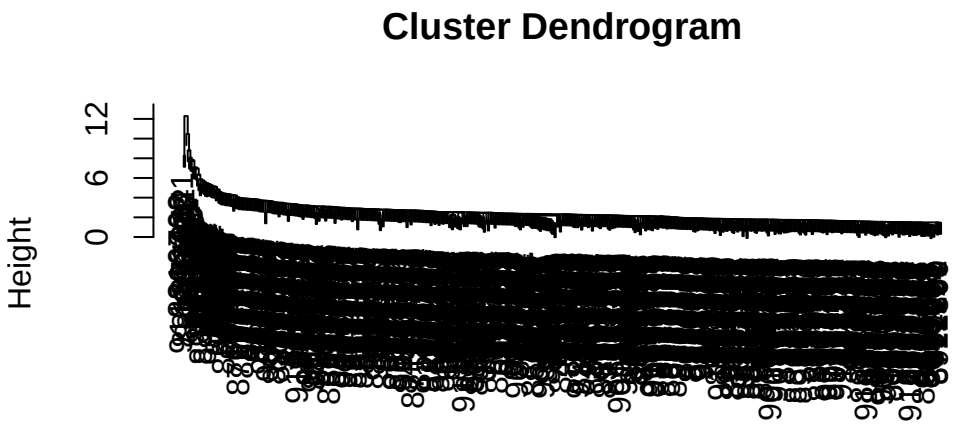

```
hc.average <- hclust(data.dist, method="average")
hc.average
```

```
Call:
hclust(d = data.dist, method = "average")

Cluster method   : average
Distance         : euclidean
Number of objects: 569
```

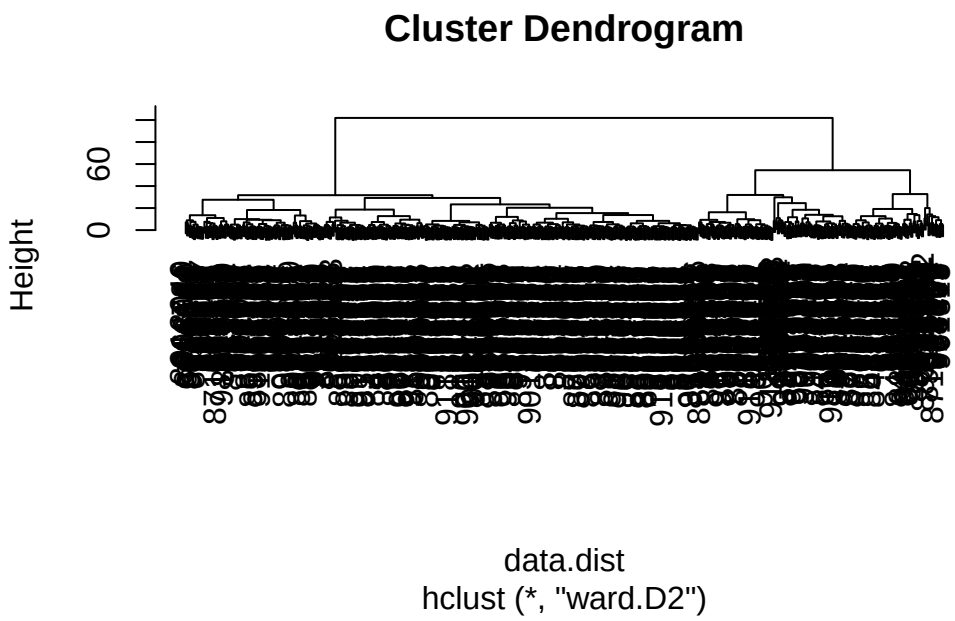
```
plot(hc.average)
```




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

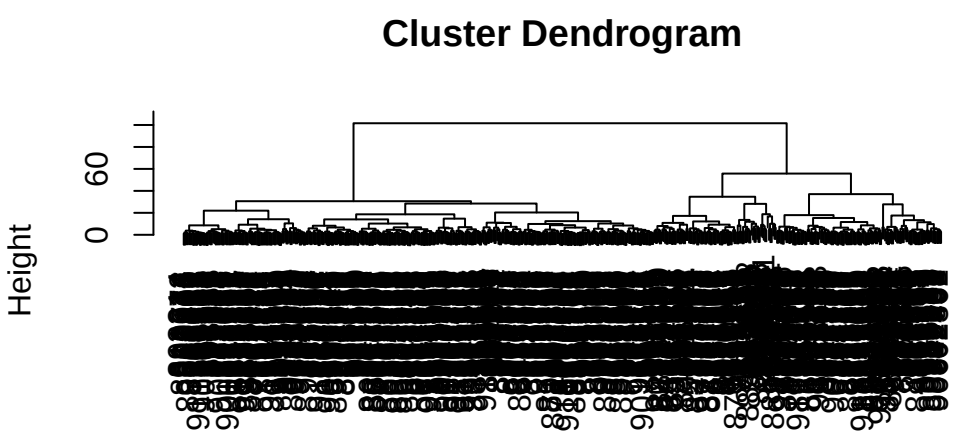


```
hc.ward <- hclust(data.dist, method="ward.D2")
plot(hc.ward)
```




```
# The method that seems to give the best result from the same 'data.dist' dataset is betwe
```

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



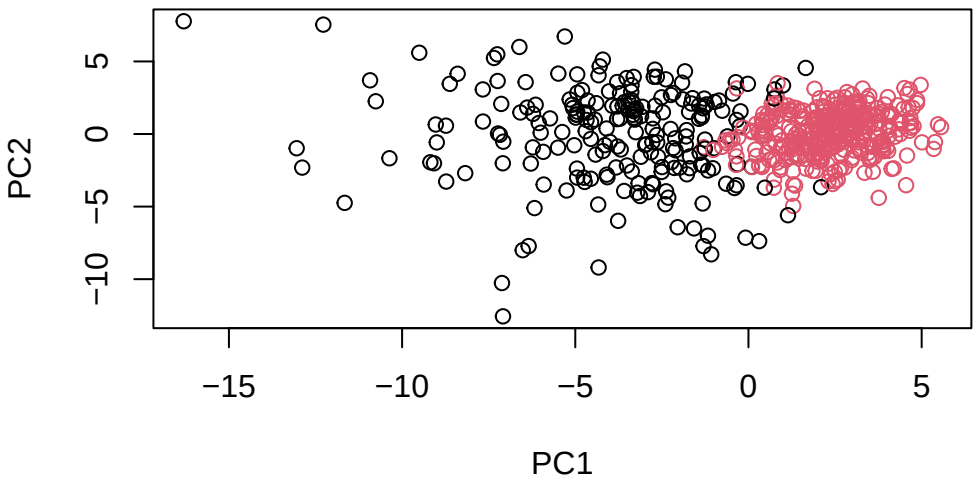

```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
  1   2
216 353
```

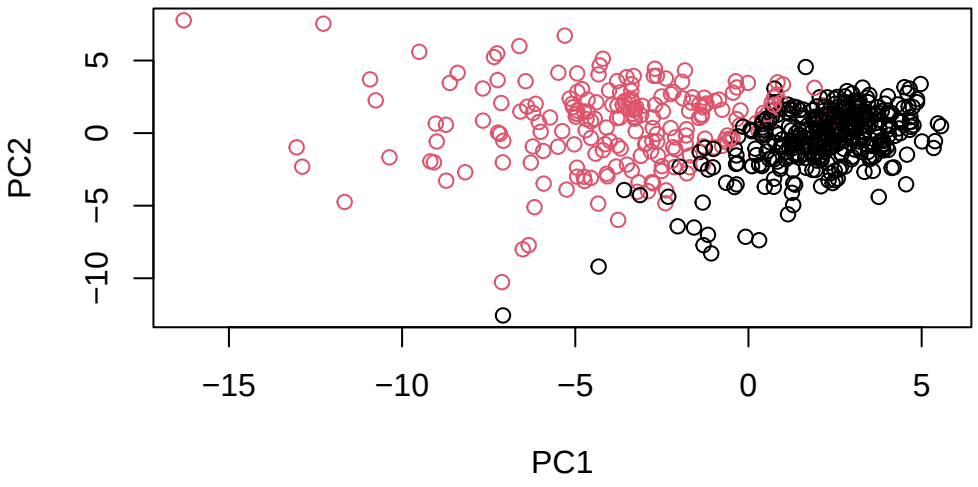
```
table(grps, diagnosis)
```

```
    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)
```



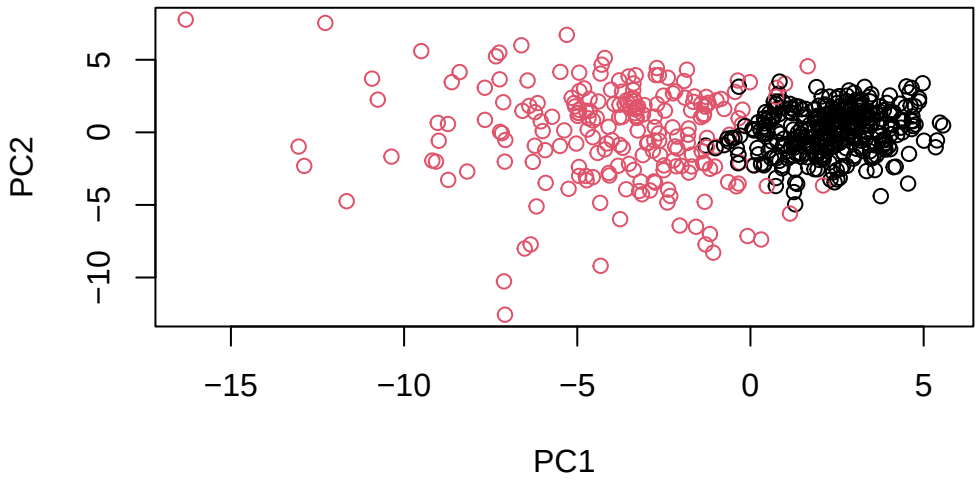
```
g <- as.factor(grps)
levels(g)
```

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

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

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

```
# Plot using re-ordered factor

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




```
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?

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

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

```
# Comparing this newly created model with two clusters compared to the previous four clust
```

Q16. How well do the k-means and hierarchical clustering models created in previous sections do in terms of separating the diagnoses?

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

```
table(wisc.km$cluster, diagnosis)
```

```
   diagnosis
      B   M
  1 343  37
  2  14 175
```

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

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

```
# From looking at both tables comparing the k-means and hierarchical clustering models, th
```

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

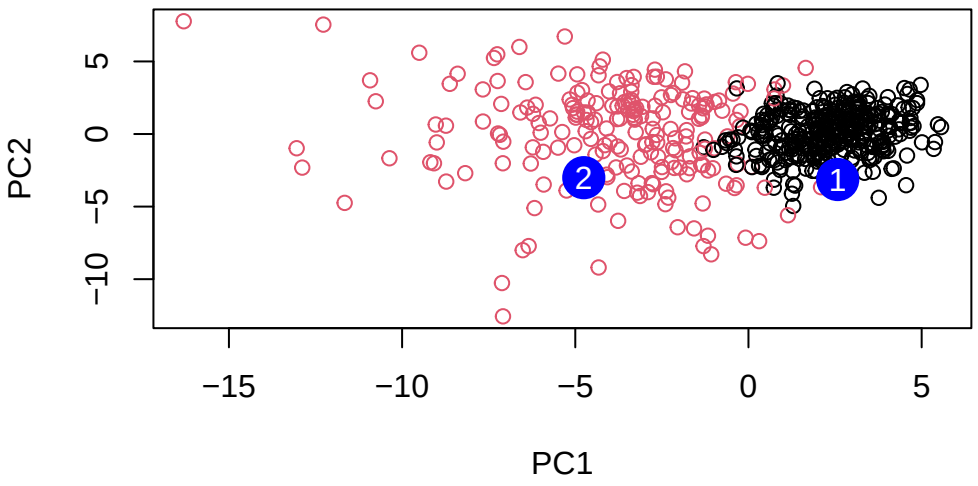
```
# plot(wisc.pr$x[,1:2], col=g), seems to produce a plot that shows the clusters that are p
```

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

```
#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")
```




```
# Based on the plot, the malignant results should be the points in red, so patient 2 shoul
```