class09_mini_project.Rmd

Anu Chaparala

10/26/2021

##Preparing the PCA data

```r
# Save your input data file into your Project directory
setwd("~/Downloads")
fna.data <- read.csv("WisconsinCancer.csv")
#fna.data print

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

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

```r
# Create diagnosis vector for later
diagnosis <- as.factor(wisc.df[,1])
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
```

> Q1. How many observations are in this dataset?

There are 32 observations per patient, and 569 patients in total.

```r
dim(fna.data)
```

```
## [1] 569  32
```

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

There are 212 malignant diagnoses.

```r
table(diagnosis)
```

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

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

There are 10 variables/featurse with suffix _mean.

```r
length(grep("_mean", colnames(wisc.data)))
```

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

##PCA

```r
# Check column means and standard deviations
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
```

```r
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
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)?

44.27% variance captured by PC1.

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

PC1-PC3, should be able to account for about 72% of the original variance in the data.

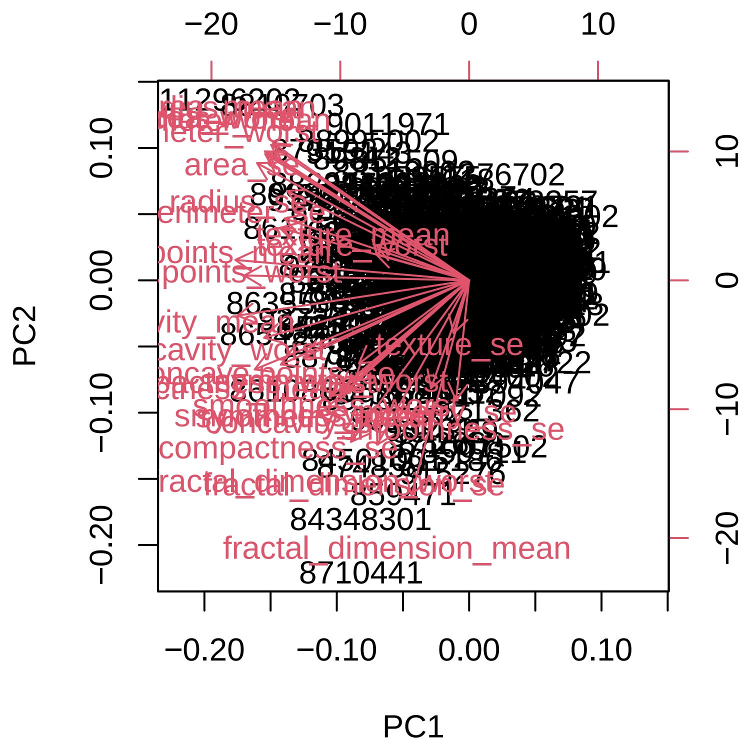
> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

PC1-PC7, should be able to account for about 91% of the original variance in the data.

##Interpreting PCA Results

We want the score plot (a.k.a. "Biplot", "PCA plot", "PC1 vs. PC2", etc.).

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

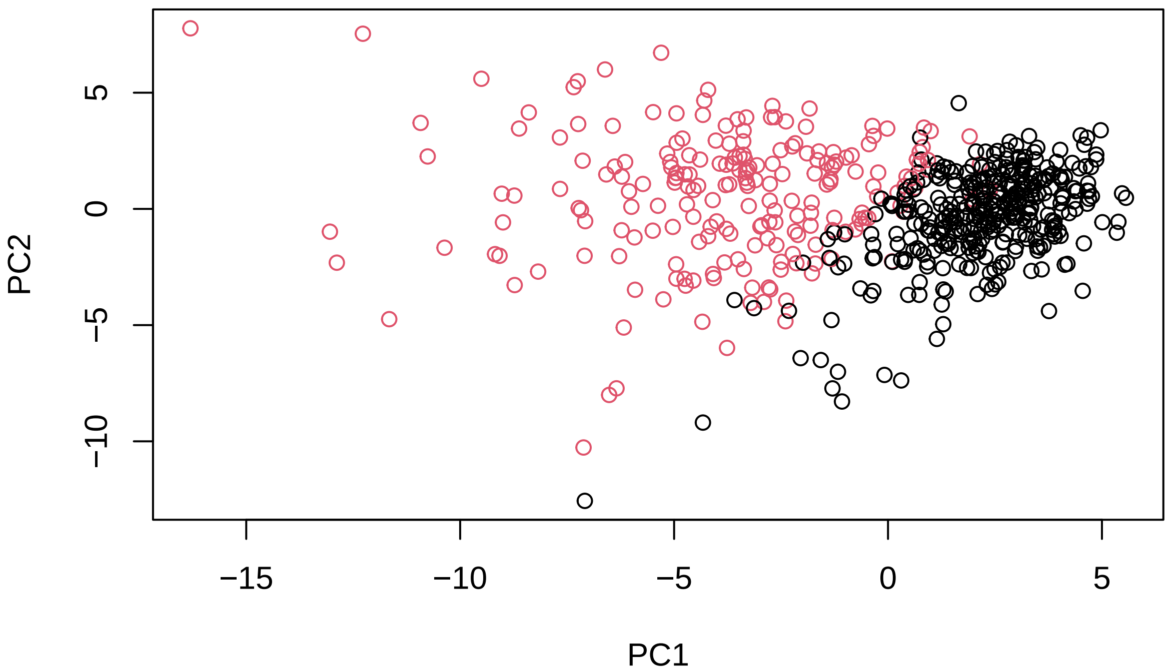


> Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

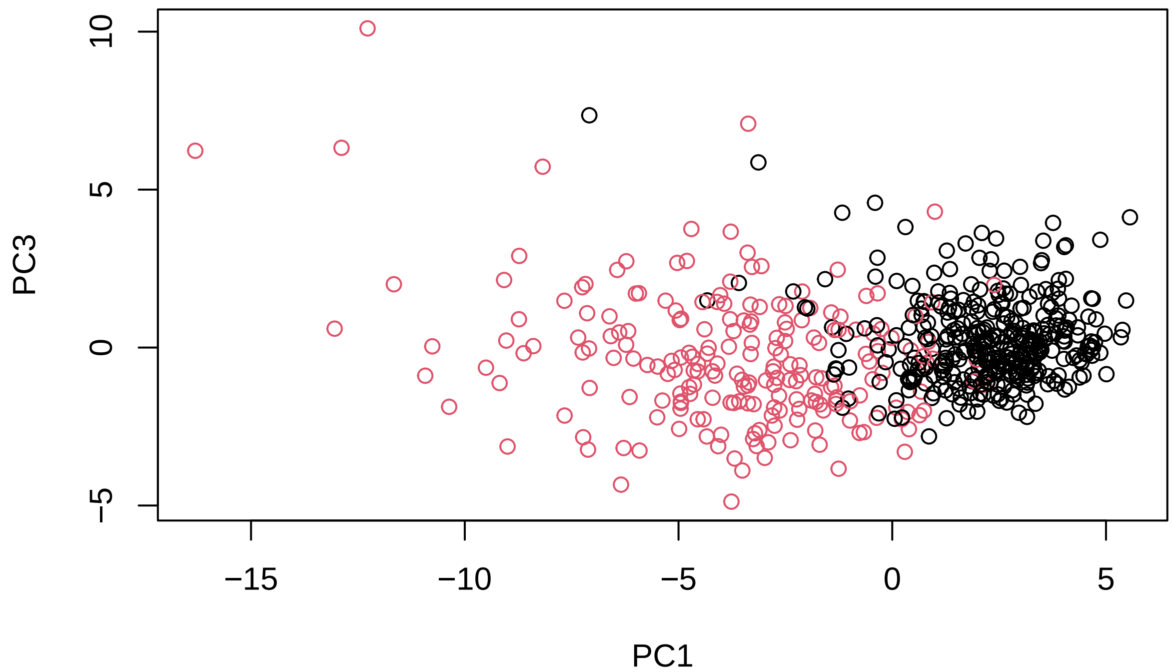
This plot is very incoherent. We can barely see and individual point, much less understand what it means or find any possible trends. Bottom line, it's a mess.

To make this plot ourselves we need to access the PCA scores data.

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




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



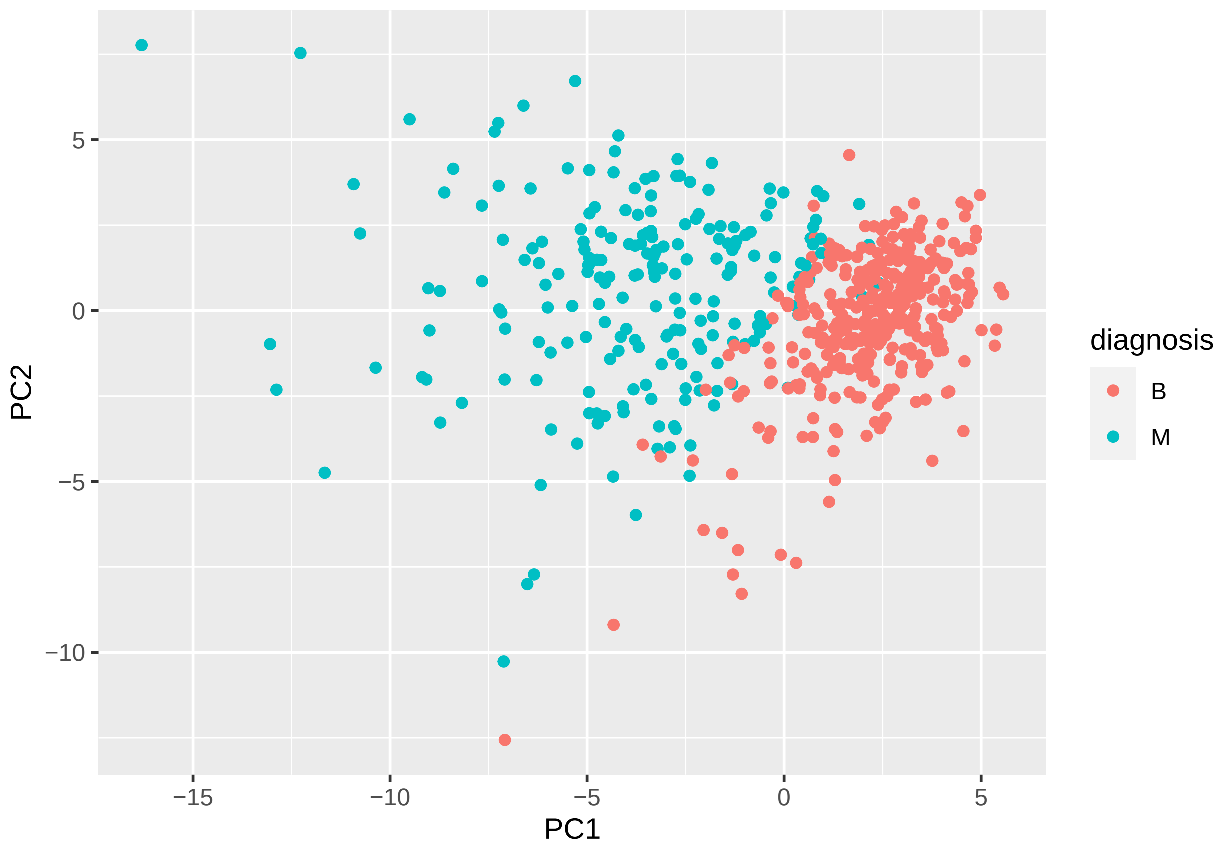
Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

There is very little clustering with regards to the different diagnosis in the PC1 by PC3 plot, as compared to the PC1 by PC2 plot. This is the case because PC2 explains more of the variance than PC3, showing a cleaner separation between the two diagnoses groups.

```
# 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()
```



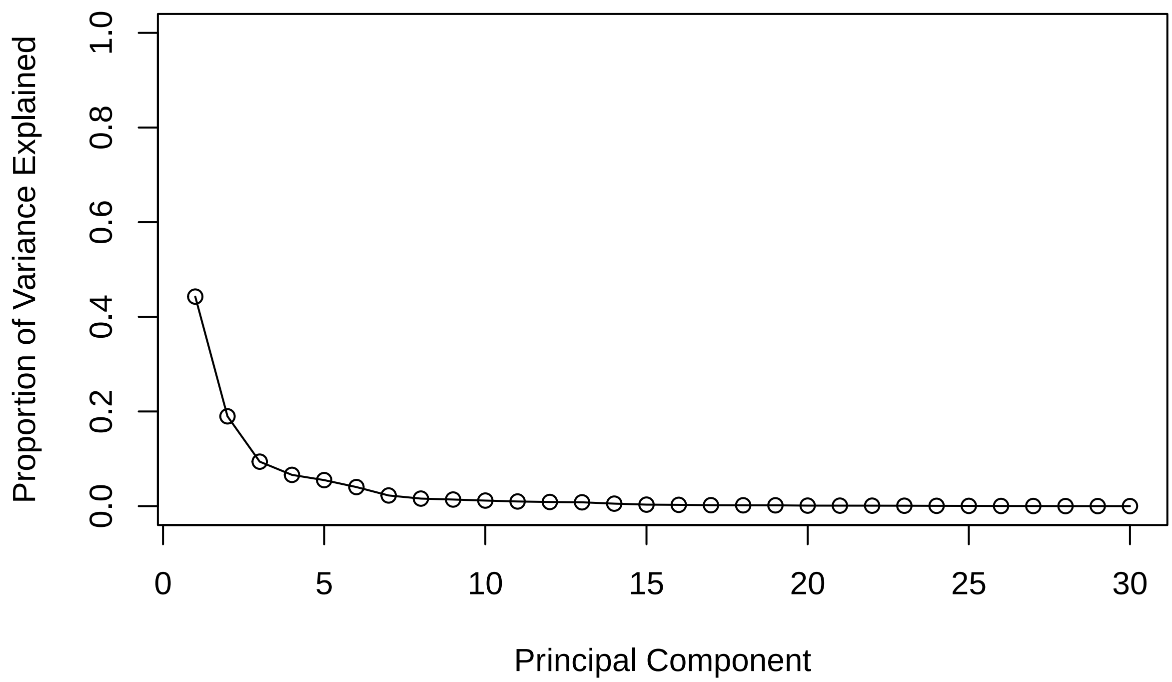
##Understanding Variance

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

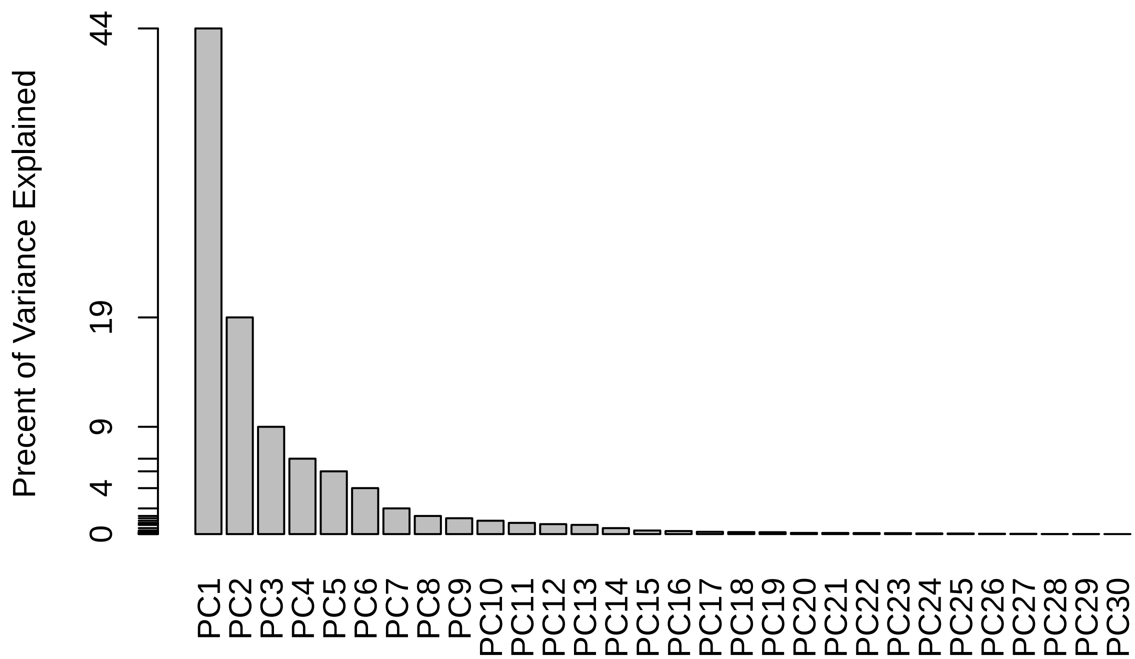
```
## [1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
# 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")
```




```r
# 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?

-0.26085376: concave.points_mean feature

```
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
```

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

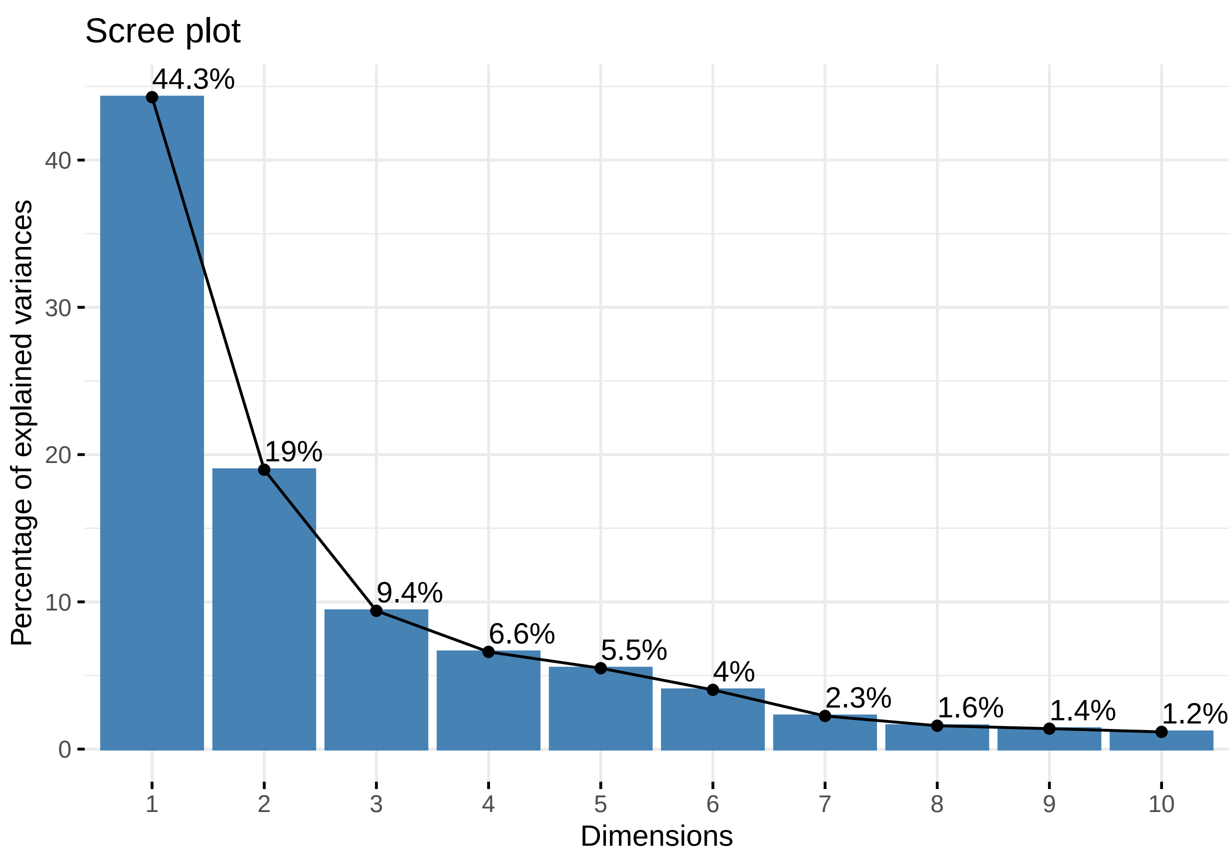
PC1-PC5 should be able to explain at least 80% of the variance of the data. (Found using summary(wisc.pr)).

Additional PCA package.

```r
## ggplot based graph
#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)
```

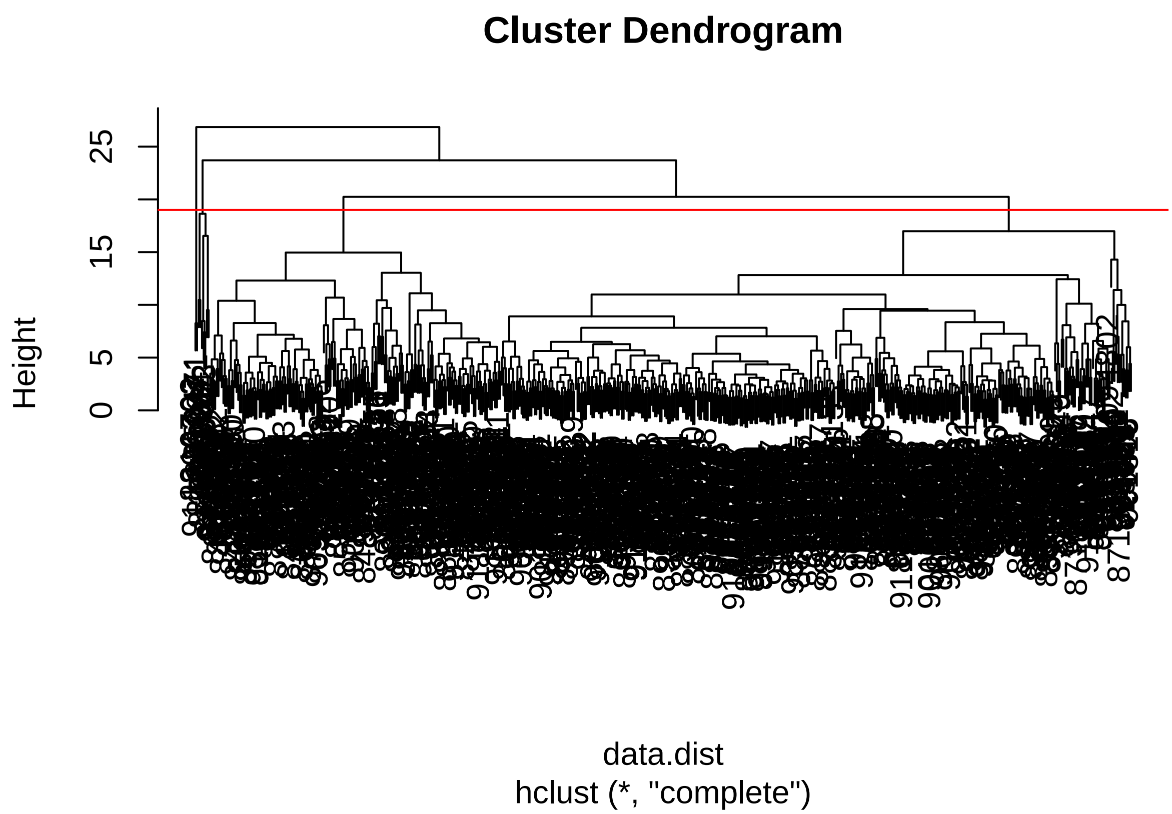


##Hierarchal Clustering

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

```r
#Create a hierarchical clustering model using complete linkage.
wisc.hclust <- hclust(data.dist, method="complete")
plot(wisc.hclust)
abline(h=19, col="red")
```



##Results of hierarchical clustering

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

The height is 19.

> Q12. Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10? (we used 4)

```r
wisc.hclust.clusters <- cutree(wisc.hclust, k=4)
table(wisc.hclust.clusters)
```

```
## wisc.hclust.clusters
##   1   2   3   4
## 177   7 383   2
```

We take the results of our PCA analysis and cluser in this space. ‘wisc.pr$x’

```r
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
```

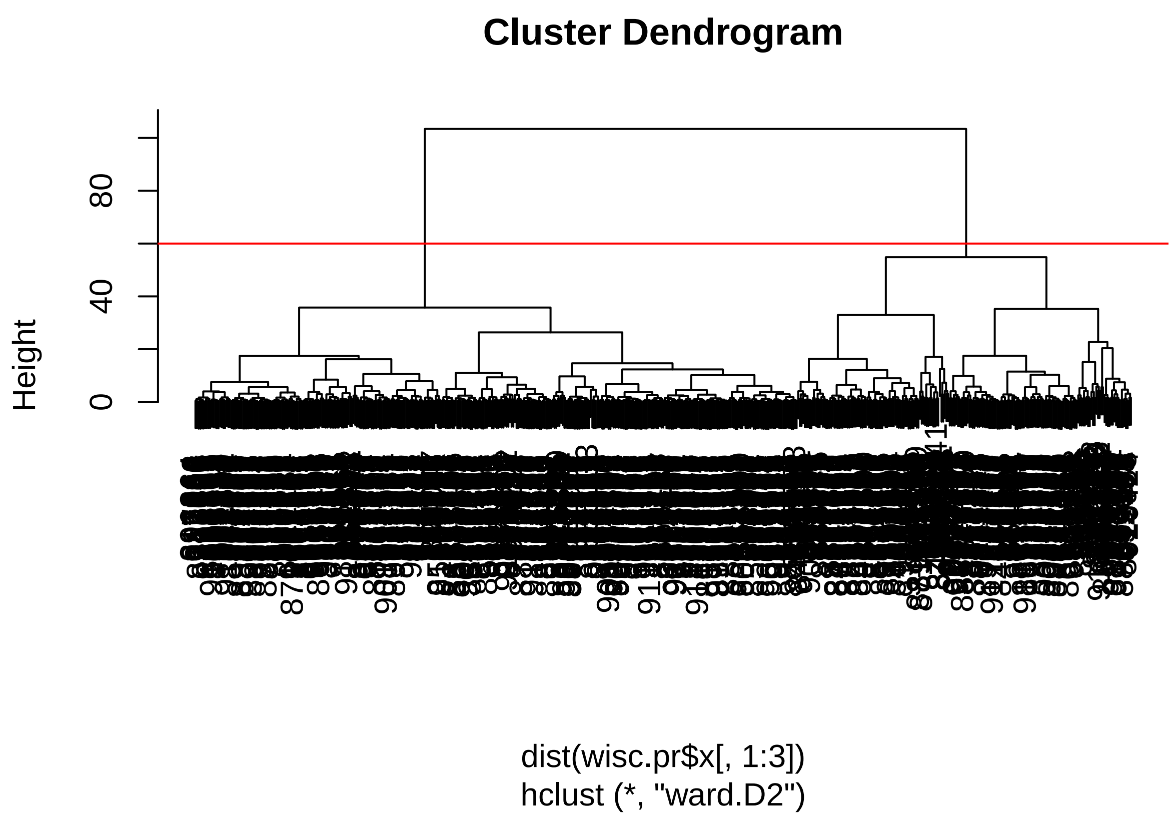
> Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

Method "ward.D2" works well because it allows me to clearly see heights that I could cutree at and use the resulting groups.

```r
wisc.pc.hclust <- hclust (dist(wisc.pr$x[, 1:3]), method = "ward.D2")
```

Plot my dendrogram

```r
plot(wisc.pc.hclust)
abline(h=60, col="red")
```



Cut the tree into k=2 groups.

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

```
## grps
##   1   2
## 203 366
```

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

Very well, we are able to clearly compare the true positives, false positives, true negatives, and false negatives with this new 4 cluster model. Cross table compare of diagnosis and my cluser groups

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

```
##          grps
## diagnosis   1   2
##         B  24 333
##         M 179  33
```

Sensitivity/Specificity

AccuracyWhat proprortion did we get correct if we call cluster 1 M and cluster 2 B?

Almost 90% accuracy.

```r
(333 + 179)/nrow(wisc.data)
```

```
## [1] 0.8998243
```

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).

Approximately 84% sensitivity.

```r
(179/(179+33))
```

```
## [1] 0.8443396
```

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).

Approximately 93% specificity.

```r
(333/(333+24))
```

```
## [1] 0.9327731
```

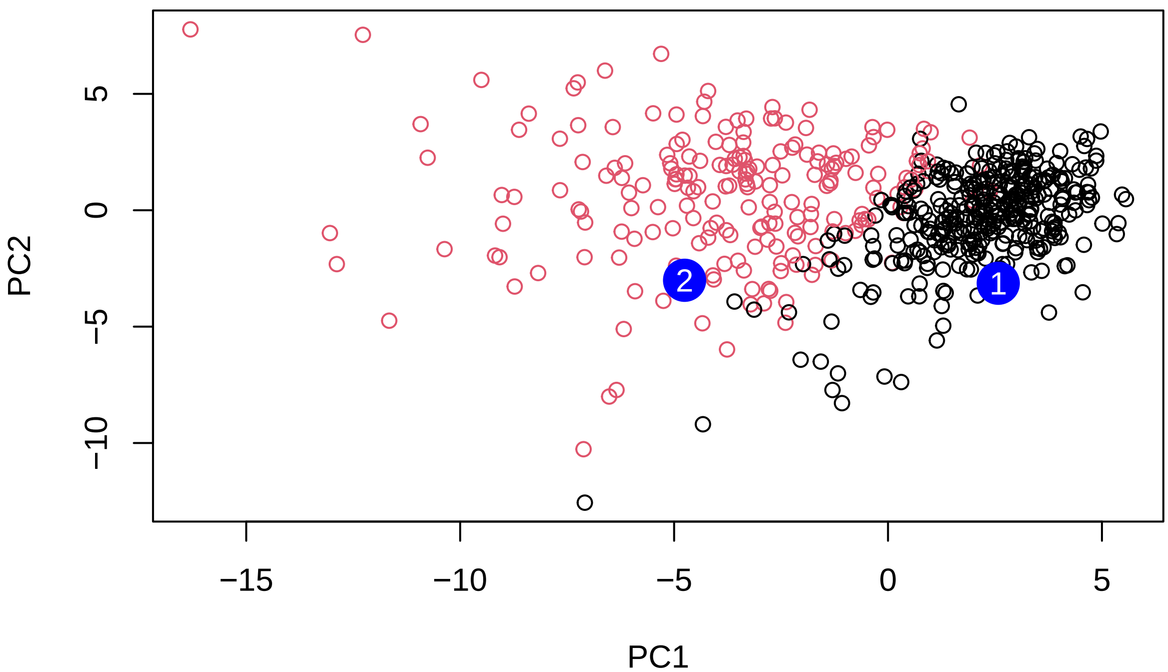
> Q16. (diregard per professor) Q17. Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

I suspect that the hierarchical clustering model gave us the highest accuracy, specificity, and sensitivity.

```r
#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
```

```r
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?

Patient 2