

Class_09_MiniProject

Angelita Rivera (PID A15522236)

10/26/2021

```
# Save your 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)
```

```
head(wisc.df)
```

```
##      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
```


Exploratory Data Analysis

Q1. How many observations are in this dataset?

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

```
## 'data.frame': 569 obs. of 30 variables:
## $ radius_mean : num 18 20.6 19.7 11.4 20.3 ...
## $ texture_mean : num 10.4 17.8 21.2 20.4 14.3 ...
## $ perimeter_mean : num 122.8 132.9 130 77.6 135.1 ...
## $ area_mean : num 1001 1326 1203 386 1297 ...
## $ smoothness_mean : num 0.1184 0.0847 0.1096 0.1425 0.1003 ...
## $ compactness_mean : num 0.2776 0.0786 0.1599 0.2839 0.1328 ...
## $ concavity_mean : num 0.3001 0.0869 0.1974 0.2414 0.198 ...
## $ concave.points_mean : num 0.1471 0.0702 0.1279 0.1052 0.1043 ...
## $ symmetry_mean : num 0.242 0.181 0.207 0.26 0.181 ...
## $ fractal_dimension_mean : num 0.0787 0.0567 0.06 0.0974 0.0588 ...
## $ radius_se : num 1.095 0.543 0.746 0.496 0.757 ...
## $ texture_se : num 0.905 0.734 0.787 1.156 0.781 ...
## $ perimeter_se : num 8.59 3.4 4.58 3.44 5.44 ...
## $ area_se : num 153.4 74.1 94 27.2 94.4 ...
## $ smoothness_se : num 0.0064 0.00522 0.00615 0.00911 0.01149 ...
## $ compactness_se : num 0.049 0.0131 0.0401 0.0746 0.0246 ...
## $ concavity_se : num 0.0537 0.0186 0.0383 0.0566 0.0569 ...
## $ concave.points_se : num 0.0159 0.0134 0.0206 0.0187 0.0188 ...
## $ symmetry_se : num 0.03 0.0139 0.0225 0.0596 0.0176 ...
## $ fractal_dimension_se : num 0.00619 0.00353 0.00457 0.00921 0.00511 ...
## $ radius_worst : num 25.4 25 23.6 14.9 22.5 ...
## $ texture_worst : num 17.3 23.4 25.5 26.5 16.7 ...
## $ perimeter_worst : num 184.6 158.8 152.5 98.9 152.2 ...
## $ area_worst : num 2019 1956 1709 568 1575 ...
## $ smoothness_worst : num 0.162 0.124 0.144 0.21 0.137 ...
## $ compactness_worst : num 0.666 0.187 0.424 0.866 0.205 ...
## $ concavity_worst : num 0.712 0.242 0.45 0.687 0.4 ...
## $ concave.points_worst : num 0.265 0.186 0.243 0.258 0.163 ...
## $ symmetry_worst : num 0.46 0.275 0.361 0.664 0.236 ...
## $ fractal_dimension_worst : num 0.1189 0.089 0.0876 0.173 0.0768 ...
```

```
str(diagnosis)
```

```
## Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
```

There are 569 observations total in this dataset.

Q2. How many observations have a malignant diagnosis?

```
table(diagnosis)
```

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

212 observations have a malignant diagnosis.

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

```
wisc.colnames <- c(colnames(wisc.data))
wisc.colnames
```

```
## [1] "radius_mean"      "texture_mean"
## [3] "perimeter_mean"   "area_mean"
## [5] "smoothness_mean"  "compactness_mean"
## [7] "concavity_mean"   "concave.points_mean"
## [9] "symmetry_mean"    "fractal_dimension_mean"
## [11] "radius_se"        "texture_se"
## [13] "perimeter_se"     "area_se"
## [15] "smoothness_se"    "compactness_se"
## [17] "concavity_se"     "concave.points_se"
## [19] "symmetry_se"      "fractal_dimension_se"
## [21] "radius_worst"     "texture_worst"
## [23] "perimeter_worst"  "area_worst"
## [25] "smoothness_worst" "compactness_worst"
## [27] "concavity_worst"  "concave.points_worst"
## [29] "symmetry_worst"   "fractal_dimension_worst"
```

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

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

There are 10 variables/features suffixed with `_mean` in this dataset.

Principal Component Analysis

```
# 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
```

```
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)
```

```
# Look at summary of results
```

```
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)?

The proportion of original variance captured by the first principal components of PC1 is 0.4427.

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

We need 3 principal components 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?

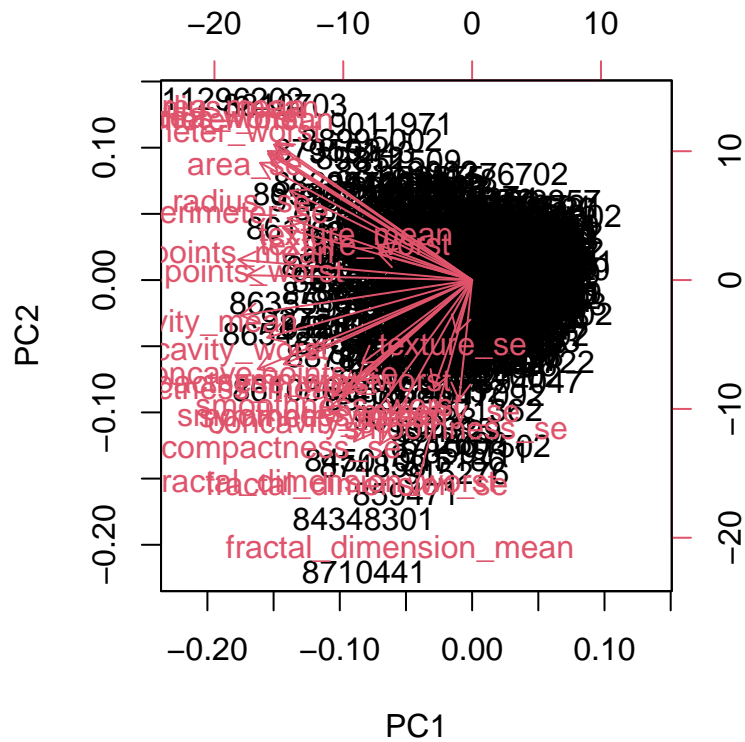
We need 7 principal components to describe at least 90% of the original variance in the data.

Interpreting PCA Results

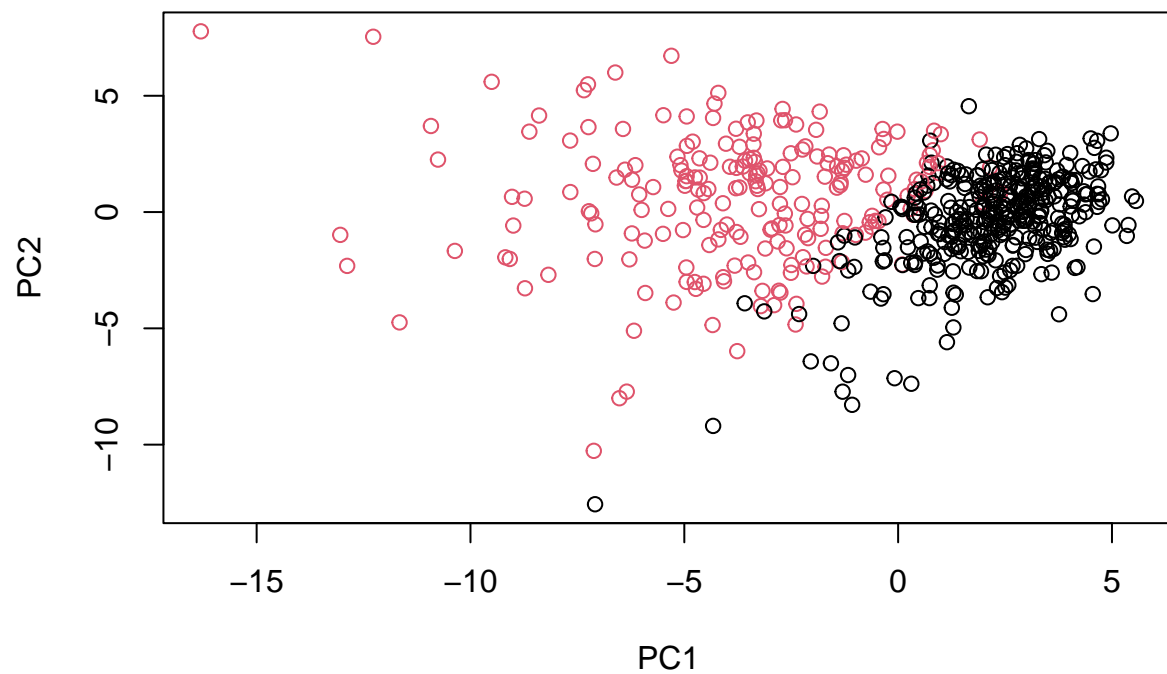
Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

The biplot is not useful for a large data set like we have. It is only useful for a smaller data set (10-15 variables). It uses labels as plots, which, because we have so much data, is near impossible to read. It is way too compacted. We need to create a better plot; something that allows us to plot/view the data in a more clear, readable way.

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

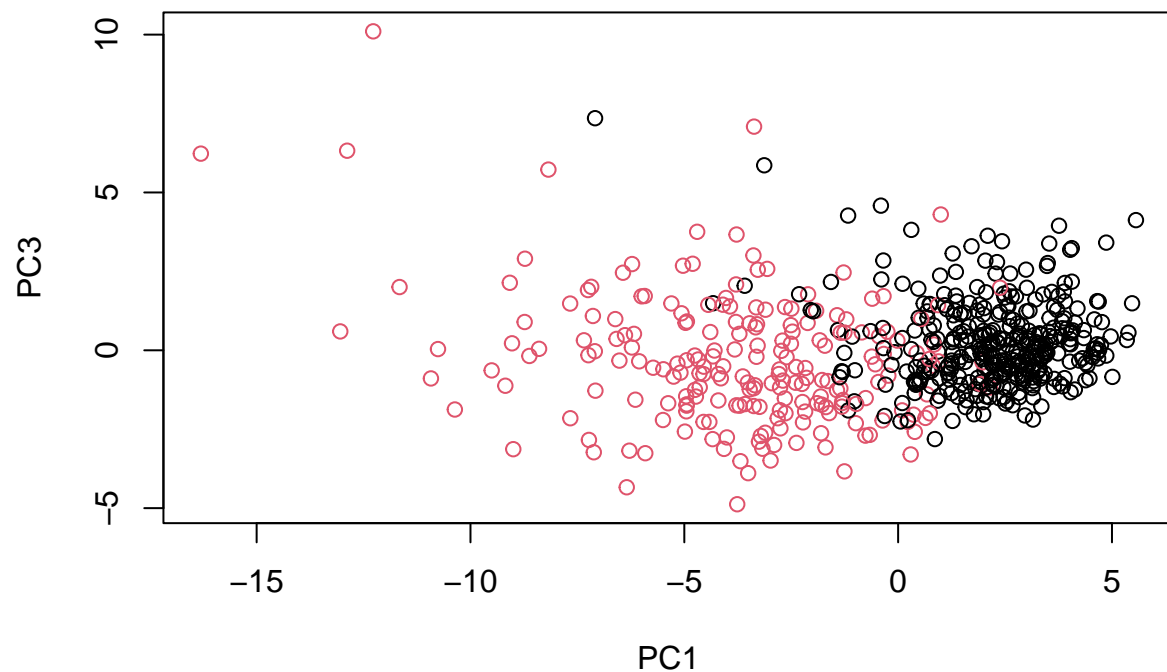


```
# Scatter plot observations by components 1 and 2
plot(wisc.pr$x, col = diagnosis,
     xlab = "PC1", ylab = "PC2")
```



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

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

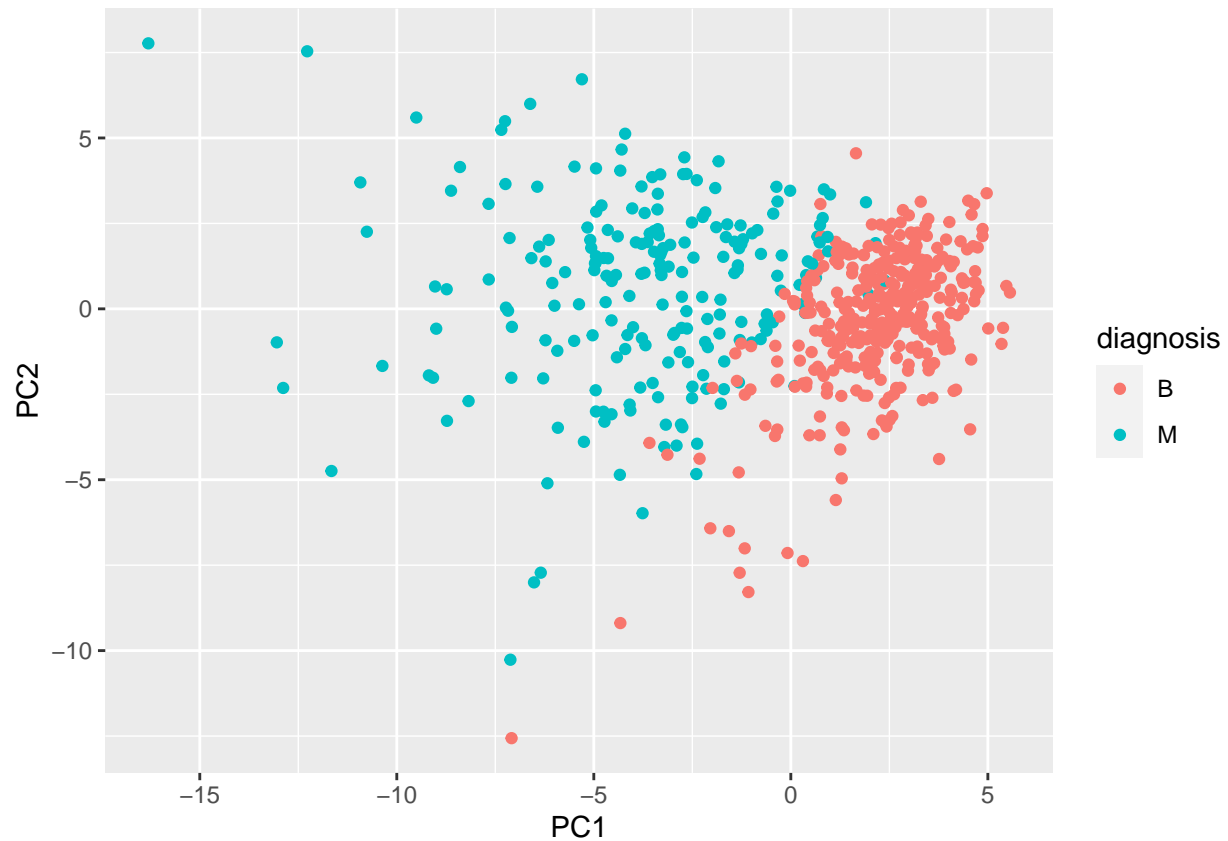
The plot is much easier to read. And, it appears to be separated by benign (black) and malignant (red) tumors.

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()
```

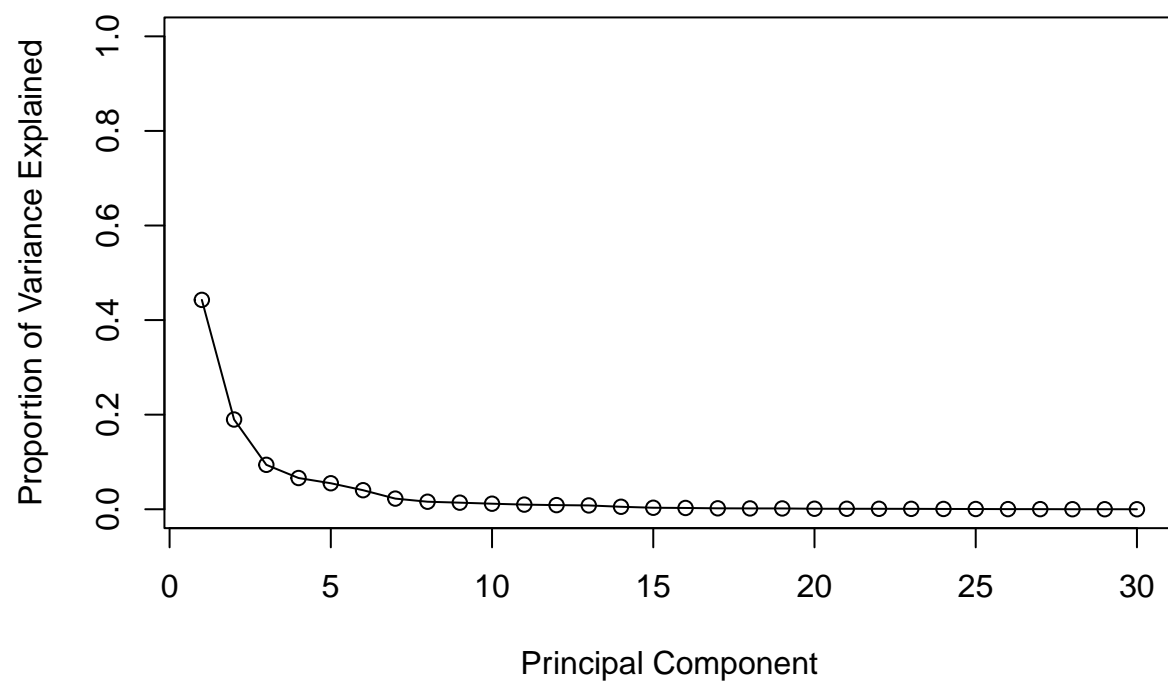


```
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

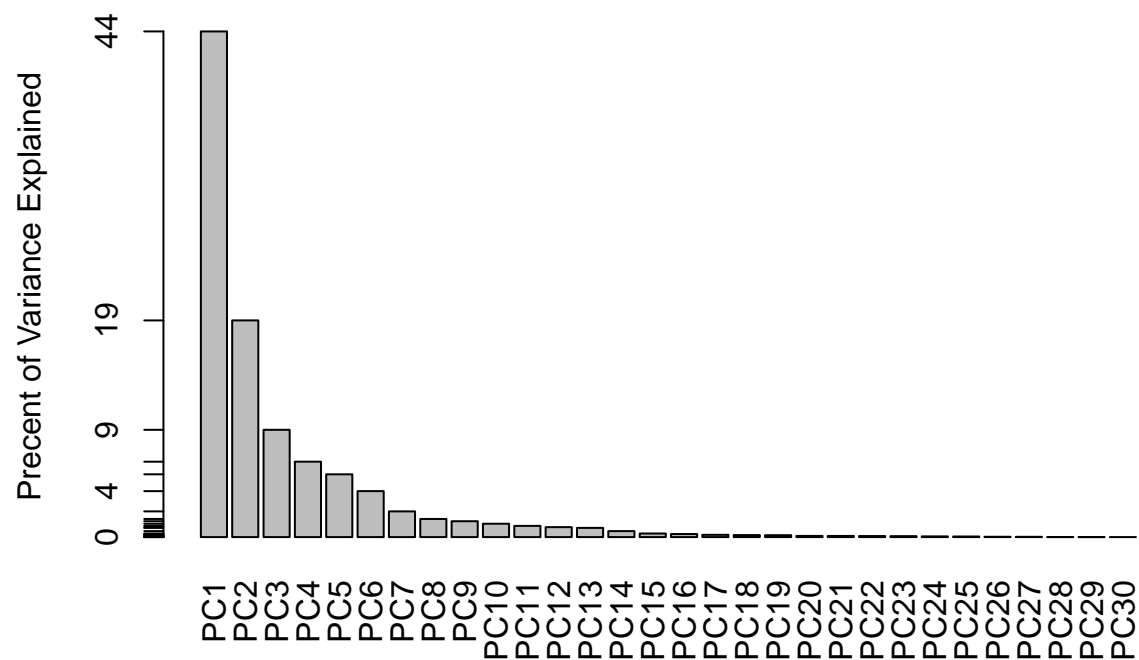
```
## [1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
# 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")
```



```
# 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 )
```

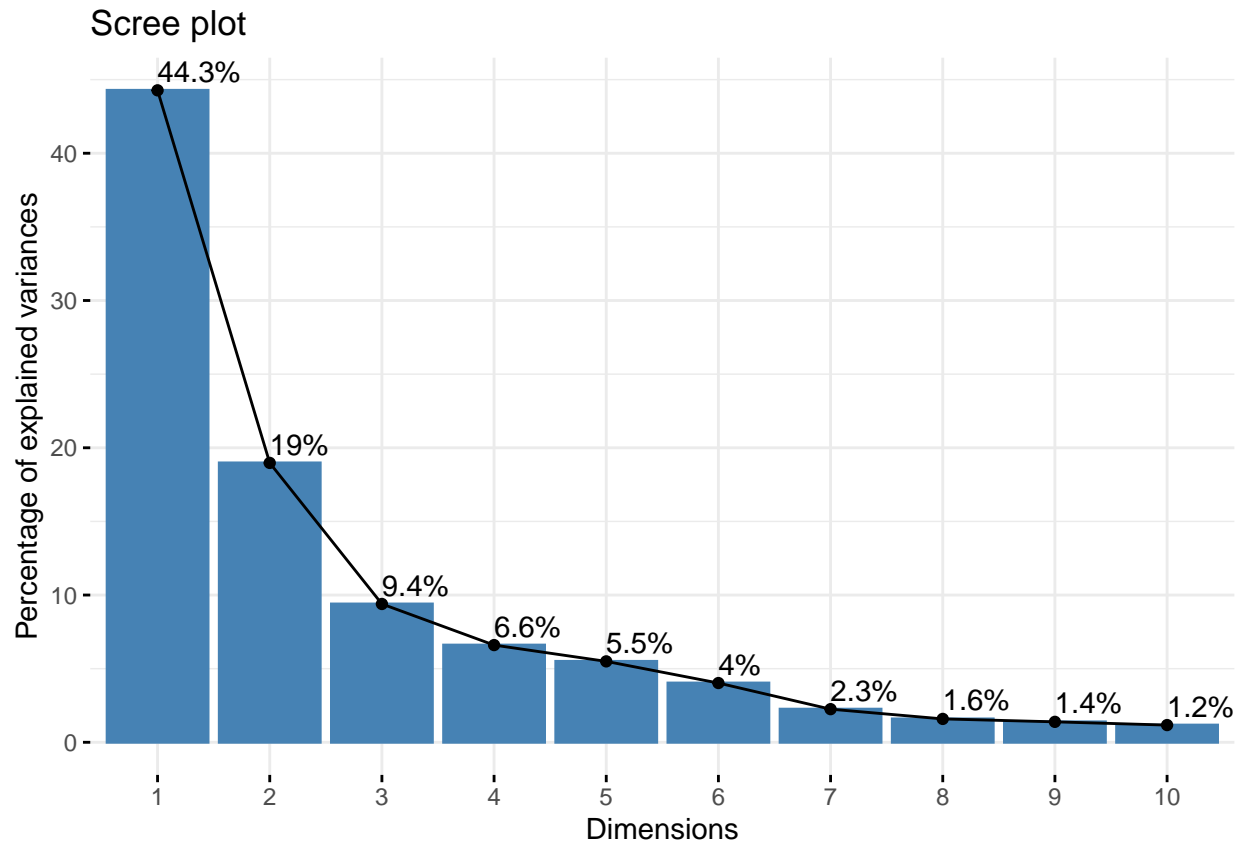


Optional

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

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

```
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`?

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

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

According to the function above; the component of the loading vector is -0.2608538.

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

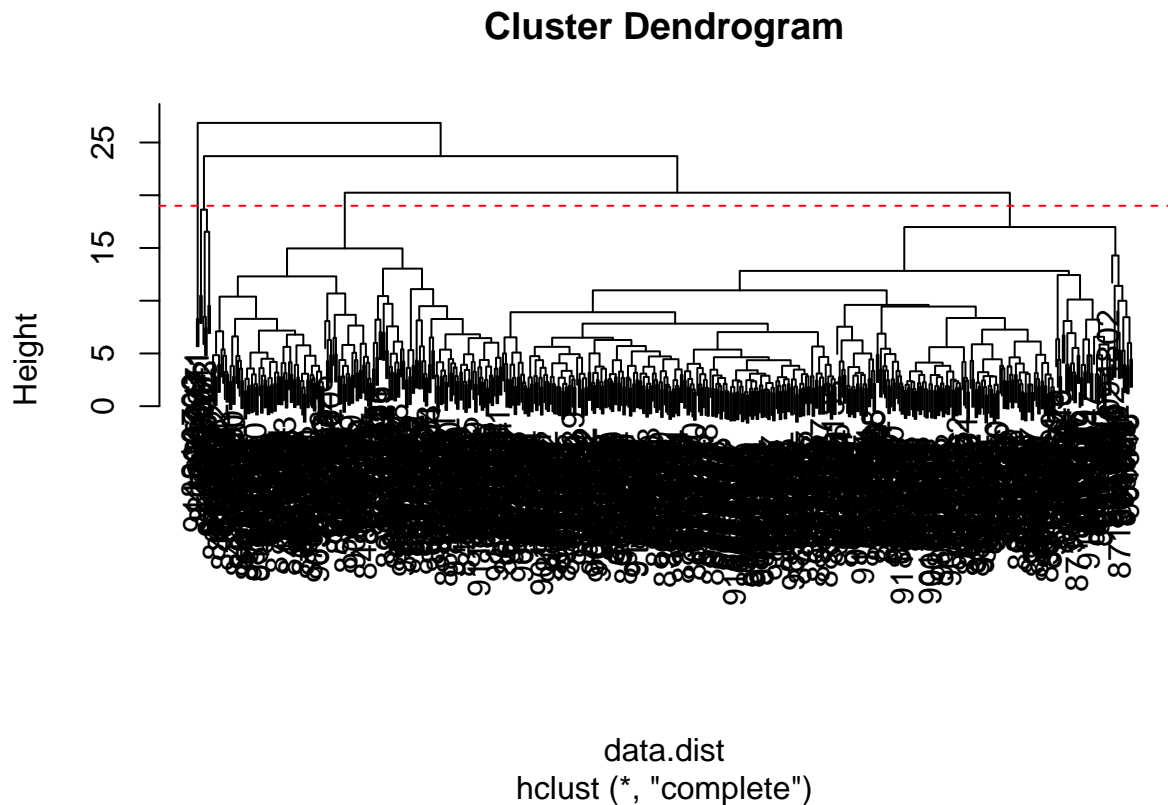
```
var <- summary(wisc.pr)
sum(var$importance[3,] < 0.8)
```

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

4 principal components are required to explain 80% of the variance data.

Hierarchical clustering

```
# Scale the wisc.data data using the "scale()" function
data.scaled <- scale(wisc.data)
data.dist <- dist(data.scaled)
wisc.hclust <- hclust(data.dist)
plot(wisc.hclust)
abline(h = 19, col = "red", lty = 2)
```



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

The height at which the clustering model has 4 clusters is 19.

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

No, 4 clusters appears to be the best suited number of clusters for this dataset.

Cut the tree into 4 groups

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

Combining methods; Clustering on PCA results

We take the results of our PCA analysis and cluster in this space 'wisc.pr\$x'

```
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
```

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

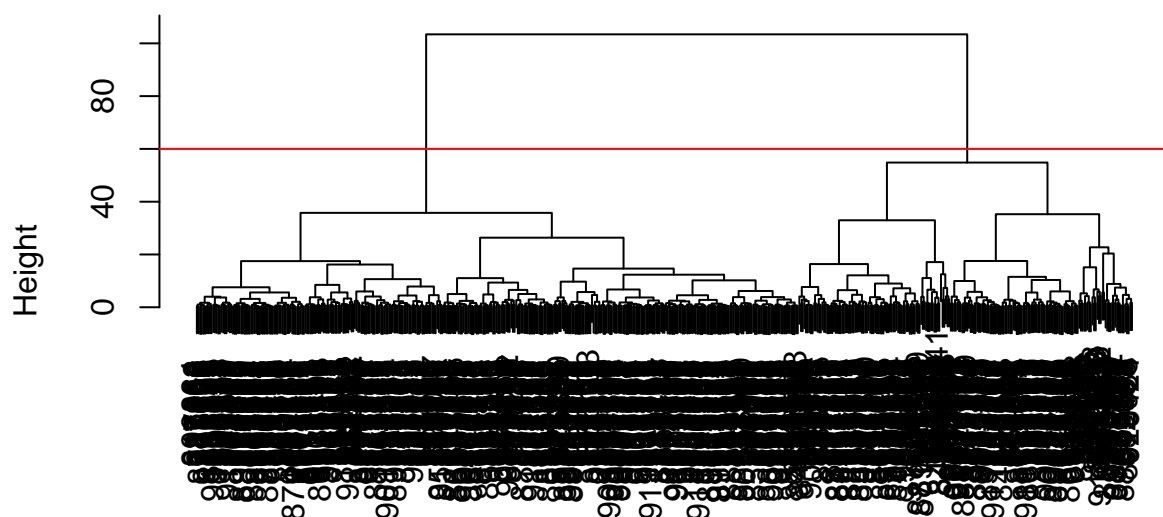
Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

I liked the method we did above the best because it takes a “bottom up” approach to clustering, which allows us to easily see clearly defined clusters. I thought the method above (and below) gave us the best clustering/plot because the variance was minimized within each cluster.

Plot my dendrogram

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

Cluster Dendrogram



```
dist(wisc.pr$x[, 1:3])
hclust (*, "ward.D2")
```

Cut the tree into k=2 groups

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

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

Cross table compare of diagnosis and my cluster groups

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

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

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

Yes, it does a good job!

Sensitivity/Specificity

Accuracy What proportion did we get correct if we call cluster 1 M and cluster 2 B?

```
(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: $TP/(TP+FN)$.

```
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: $TN/(TN+FN)$.

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

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

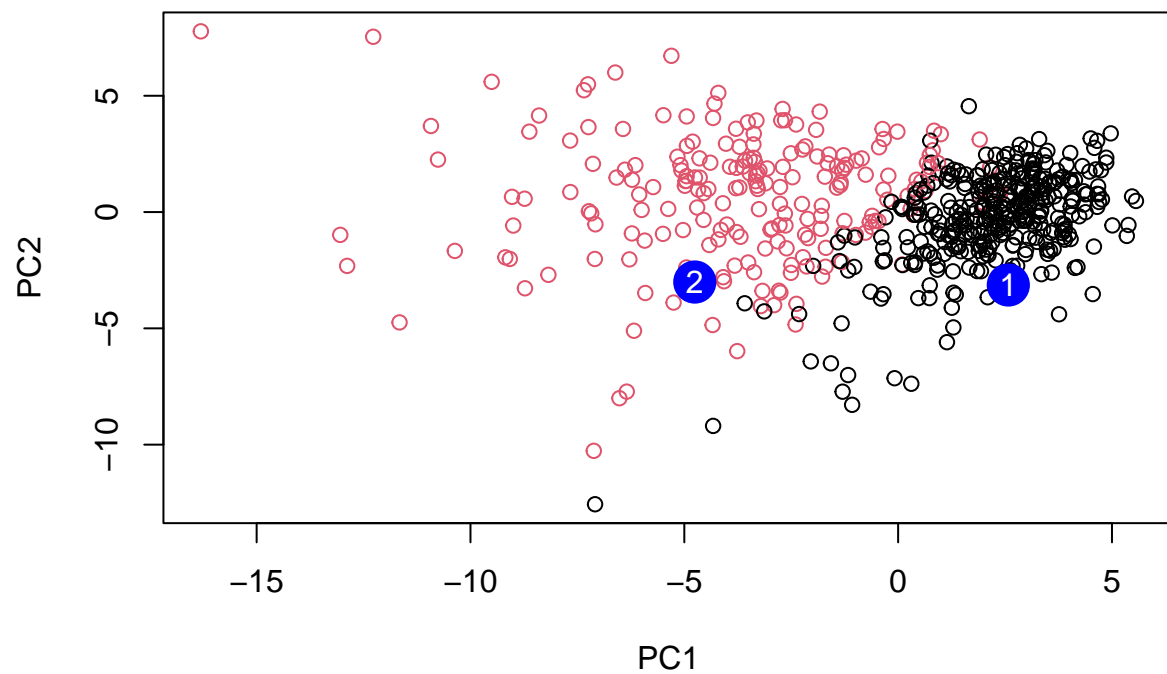
Prediction

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

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

The PCA clustering method has the best specificity and sensitivity.

```
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?

You should prioritize patient #2, because their diagnosis falls within the malignant cluster.