

# Class 09 mini-project

Neva Olliffe PID: A69026930

## Load data

```
# 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
84358402	0.01756		0.005115	22.54	16.67
843786	0.02165		0.005082	15.47	23.75
	perimeter_worst	area_worst	smoothness_worst	compactness_worst	
842302	184.60	2019.0	0.1622		0.6656
842517	158.80	1956.0	0.1238		0.1866
84300903	152.50	1709.0	0.1444		0.4245
84348301	98.87	567.7	0.2098		0.8663
84358402	152.20	1575.0	0.1374		0.2050
843786	103.40	741.6	0.1791		0.5249
	concavity_worst	concave.points_worst	symmetry_worst		
842302	0.7119		0.2654	0.4601	
842517	0.2416		0.1860	0.2750	
84300903	0.4504		0.2430	0.3613	
84348301	0.6869		0.2575	0.6638	
84358402	0.4000		0.1625	0.2364	
843786	0.5355		0.1741	0.3985	
	fractal_dimension_worst	X			
842302		0.11890	NA		
842517		0.08902	NA		
84300903		0.08758	NA		
84348301		0.17300	NA		
84358402		0.07678	NA		
843786		0.12440	NA		

## Separate the diagnosis from the rest of the data

```
# We can use -1 here to remove the first column
wisc.data <- wisc.df[,-1]
wisc.data <- wisc.data[,-31]

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

### Q1. How many observations are in this dataset?

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

```
[1] 569 30
```

569 patients are included in this dataset.

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

```
sum(wisc.df$diagnosis == "M")
```

```
[1] 212
```

212 patients have a malignant diagnosis.

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

```
length(grep("__mean", colnames(wisc.df)))
```

```
[1] 10
```

10 different columns are suffixed with \_\_mean

## 2. PCA

```
# 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

```
print("SD")
```

```
[1] "SD"
```

```
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 = T)
```

```
# 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)?**

44.27% of the variance.

```
v <- summary(wisc.pr)
pcvar <- v$importance[3,]
which(pcvar>= 0.7)[1]
```

PC3

3

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

The 3 PCs are required to describe 70% of variance.

```
v <- summary(wisc.pr)
pcvar <- v$importance[3,]
which(pcvar>= 0.9)[1]
```

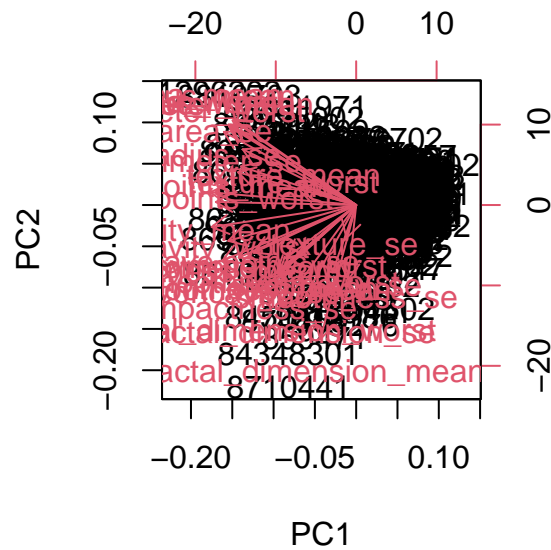
PC7

7

**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 90% of the variance of the data.

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

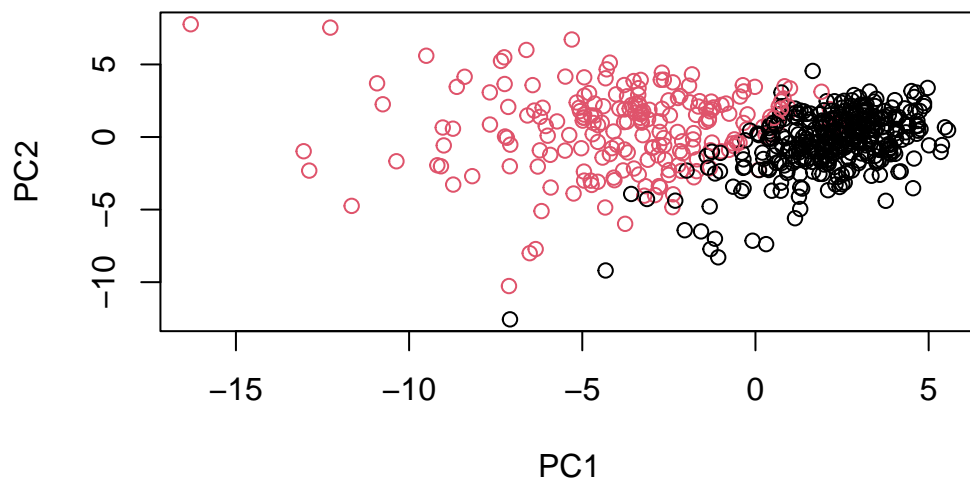


**Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?**

This plot is very difficult to understand. All of the data points are labeled so it's very hard to see them, and the labeling of the vectors is confusing.

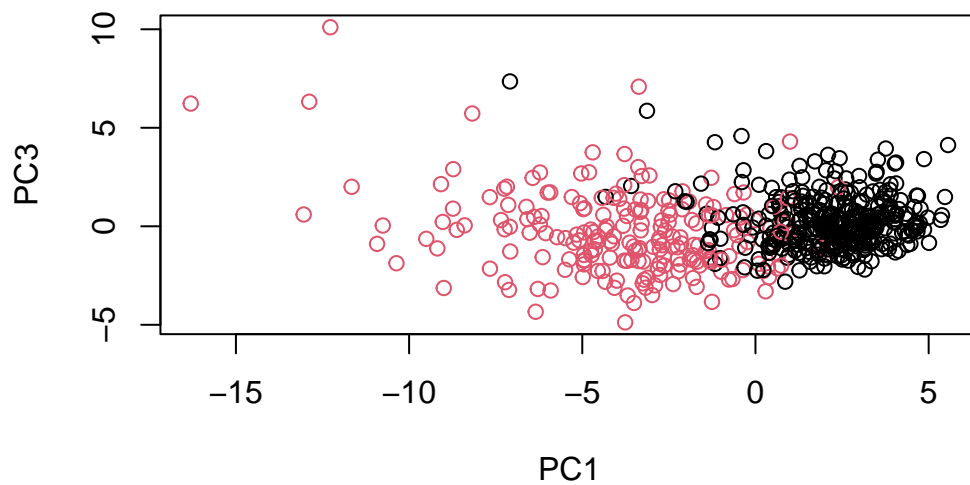
```
# Change diagnosis to color

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



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





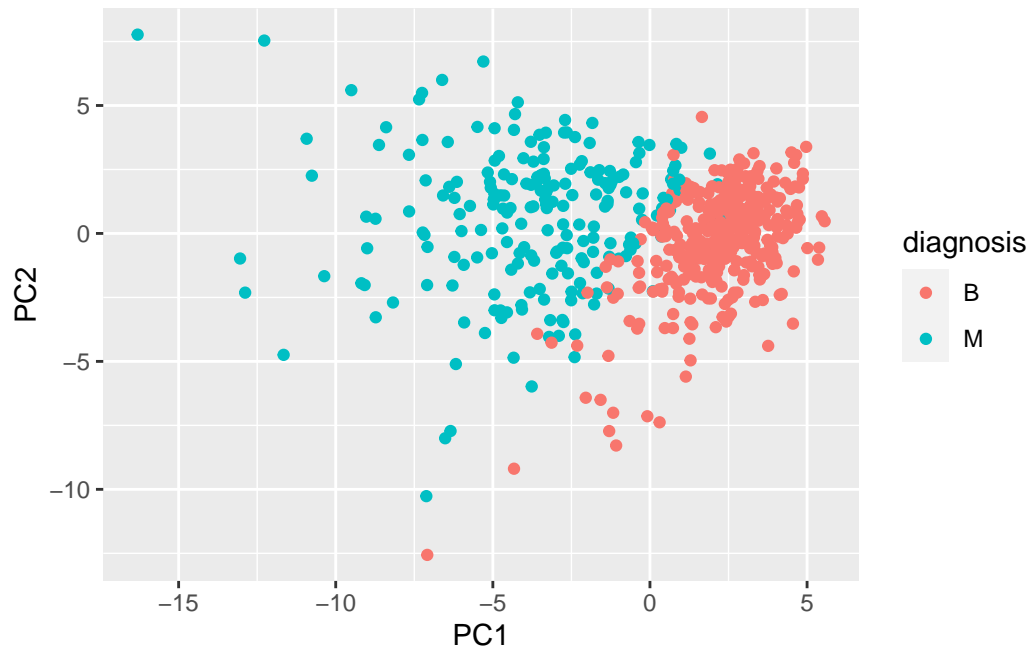
**Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?**

PC1 is doing more separation of benign and malignant factors than either PC1 or PC3. In both plots, the vast majority of sample separation is on the x axis.

```
# 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()
```

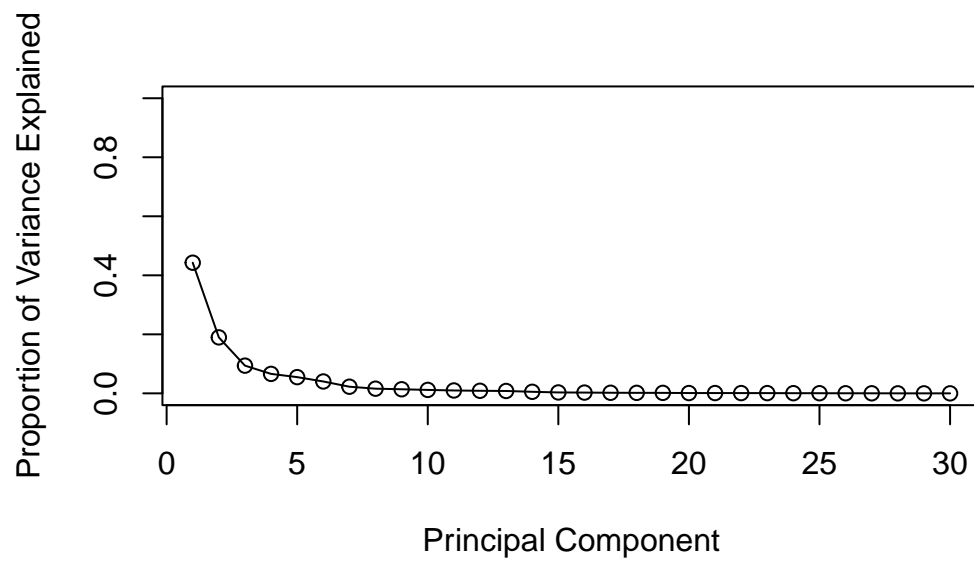


```
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

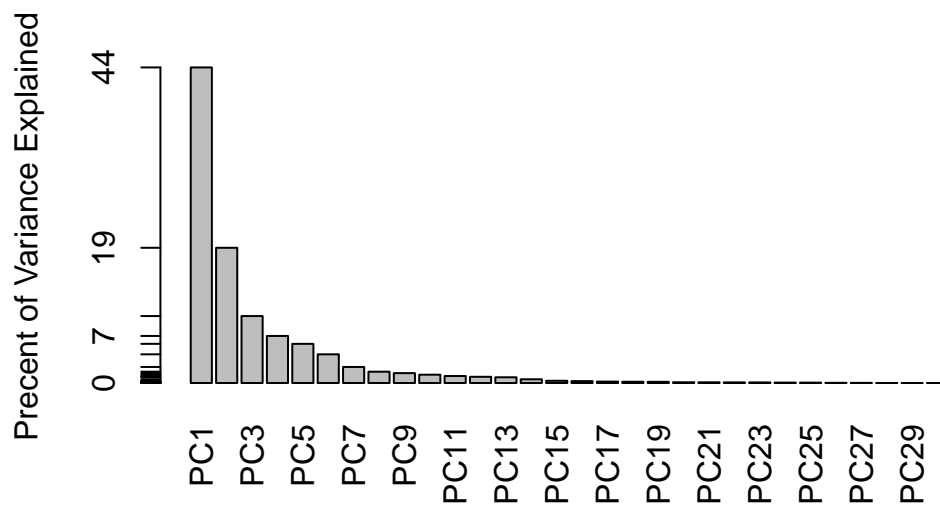
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
# 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")
```



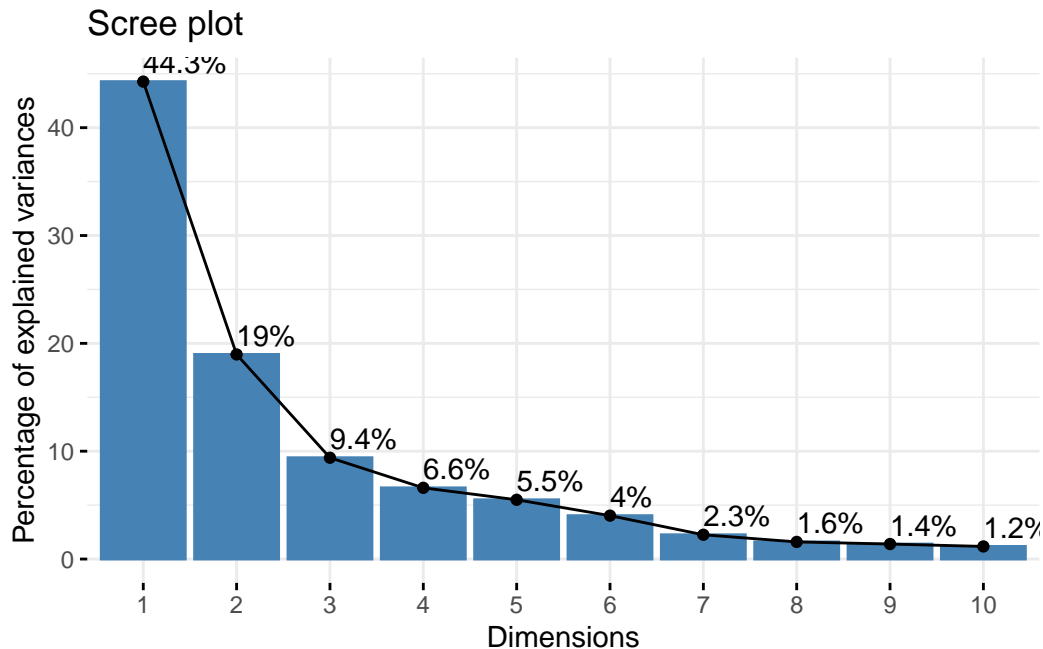
```
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
        names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```



```
## ggplot based graph
#install.packages("factoextra")
library(factoextra)
```

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 (i.e. `wisc.pr$rotation[,1]`) for the feature `concave.points_mean`? This tells us how much this original feature contributes to the first PC.**

The loading vector for `concave.points_mean` is -0.26086. Because this factor is negative and malignant cells are plotting negatively on the x axis, this tells us that a high mean of concave points contributes to a malignant diagnosis.

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

```
[1] -0.2608538
```

### 3. Hierarchical clustering

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

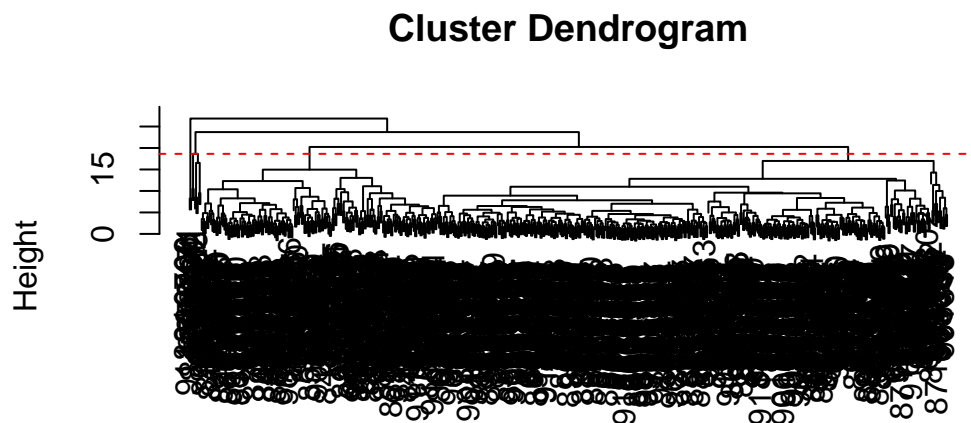
# Find Euclidian distances
data.dist <- dist(data.scaled)
```

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

**Q10. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?**

```
sorted_clust <- sort(-wisc.hclust$height)

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



```
data.dist
hclust (*, "complete")
```

```
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

**Q11. OPTIONAL: Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10? How do you judge the quality of your result in each case?**

**Q12. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.**

I prefer ward.D2, which is easier to understand visually as a plot and gives the cleanest separation between benign and malignant samples using only 2 clusters.

```

methods <- c("single", "complete", "average", "ward.D2")

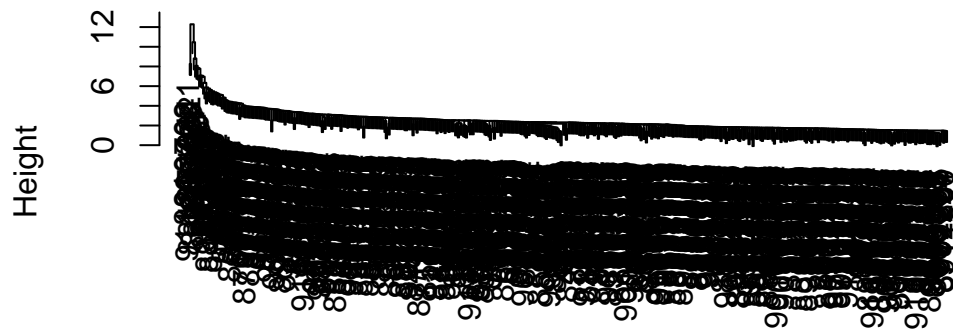
for (m in methods) {
  # Re-cluster for each method
  temp.wisc.hclust <- hclust(data.dist, method = m)
  # plot each method
  plot(temp.wisc.hclust, main = paste("method =", m))

  # Make the table for 2 clusters
  temp.wisc.hclust.clusters <- cutree(temp.wisc.hclust, k = 2)

  print(paste("method =", m))
  print(table(temp.wisc.hclust.clusters, diagnosis))
}

```

method = single

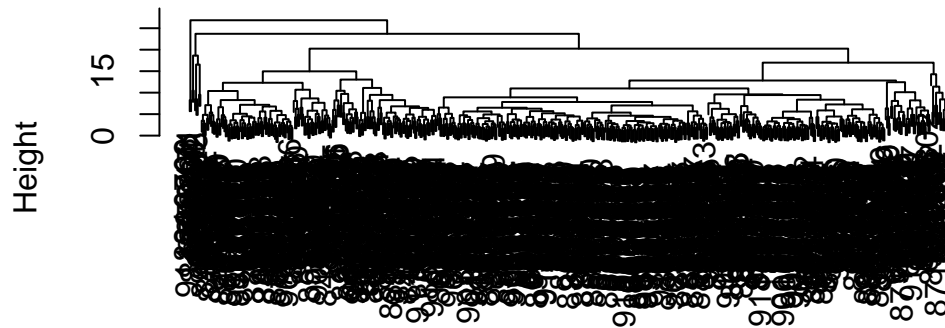


data.dist  
hclust (\*, "single")

```
[1] "method = single"
      diagnosis
temp.wisc.hclust.clusters  B  M
      1 357 210
      2   0   2
```



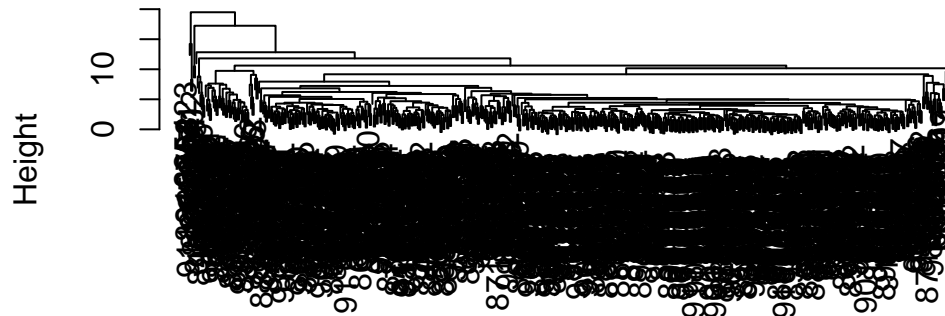
**method = complete**



data.dist  
hclust (\*, "complete")

```
[1] "method = complete"
      diagnosis
temp.wisc.hclust.clusters  B  M
      1 357 210
      2   0   2
```

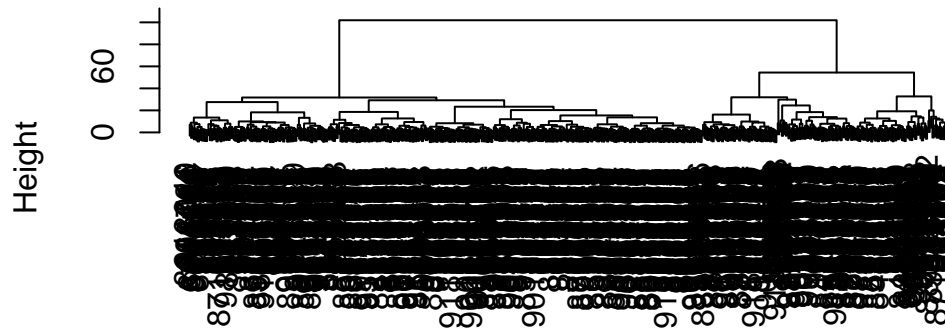
method = average



```
data.dist  
hclust (*, "average")
```

```
[1] "method = average"  
      diagnosis  
temp.wisc.hclust.clusters  B  M  
      1 357 209  
      2   0   3
```

method = ward.D2



data.dist  
hclust (\*, "ward.D2")

```
[1] "method = ward.D2"
      diagnosis
temp.wisc.hclust.clusters  B  M
      1  20 164
      2 337  48
```

## 4. Combining methods

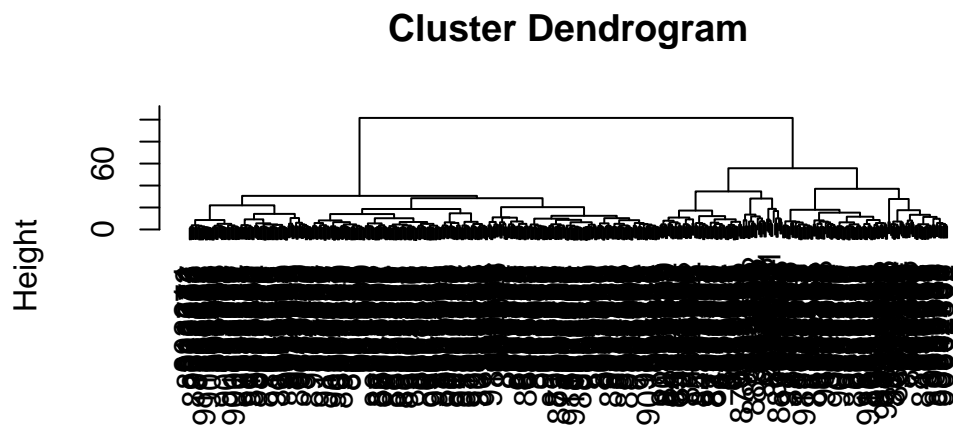
```
# Determine how many PCs are required to explain at least 90% of the variance.
pcvar<- summary(wisc.pr)$importance[3,]
which(pcvar>=0.9)[1]
```

```
PC7
7
```

```
## Redo euclidean distances for pr data
wisc.pr.dist <- dist(wisc.pr$x[,1:7])

# Perform clustering and plot
```

```
wisc.pr.hclust <- hclust(wisc.pr.dist, method = "ward.D2")
plot(wisc.pr.hclust)
```



```
wisc.pr.dist
hclust (*, "ward.D2")
```

```
##Check clustering results compared to diagnosis
grps <- cutree(wisc.pr.hclust, k=2)
table(grps, diagnosis)
```

```
diagnosis
grps  B  M
1   28 188
2   329 24
```

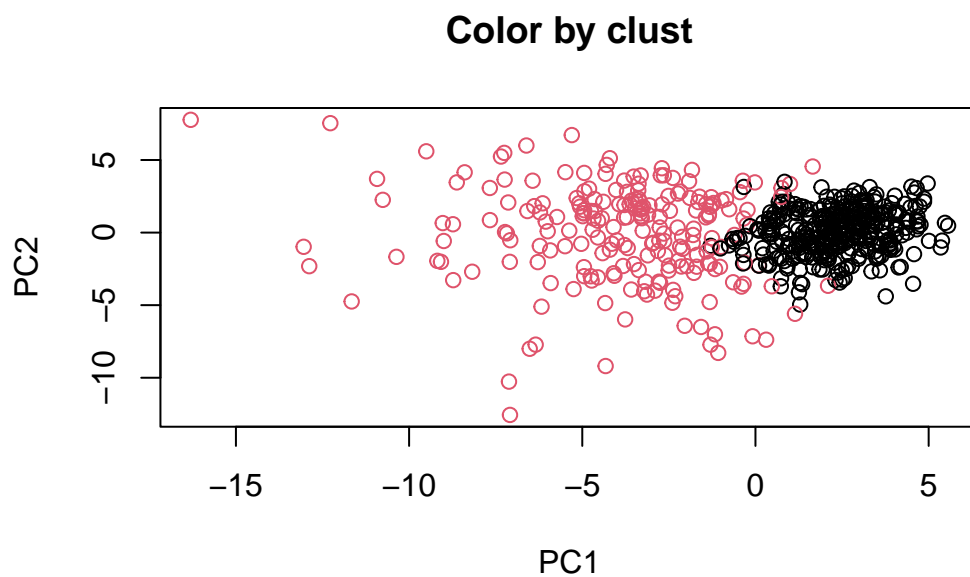
```
# Adjust the colors
g <- as.factor(grps)
levels(g)
```

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

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

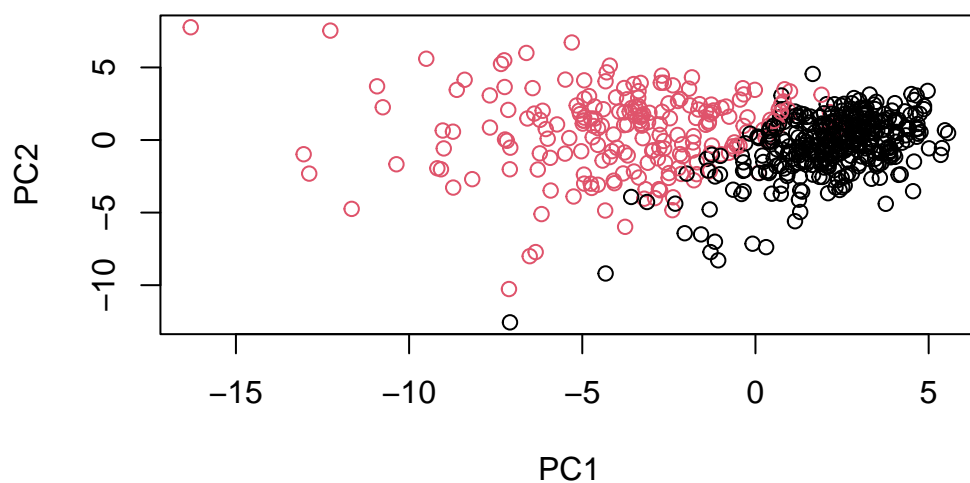
```
[1] "2" "1"
```

```
# Plot results from pca then cluster
plot(wisc.pr$x[,1:2], col=g, main = "Color by clust")
```



```
plot(wisc.pr$x[,1:2], col=diagnosis, main = "Color by diagnosis")
```

### Color by diagnosis



```
wisc.pr.hclust.2clusters <- cutree(wisc.pr.hclust, k=2)
# Compare to actual diagnoses
table(wisc.pr.hclust.2clusters, diagnosis)
```

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

**Q13. How well does the newly created model with four clusters separate out the two diagnoses?**

This model separates out fairly well, giving us 28 false positives and 24 false negatives.

**Q14. How well do the 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.**

```
methods <- c("single", "complete", "average", "ward.D2")

for (m in methods) {
  # Re-cluster for each method
  temp.wisc.hclust <- hclust(data.dist, method = m)

  # Make the table for 4 clusters
  temp.wisc.hclust.clusters <- cutree(temp.wisc.hclust, k = 4)

  print(paste("method =", m))
  print(table(temp.wisc.hclust.clusters, diagnosis))
}
```

```
[1] "method = single"
              diagnosis
temp.wisc.hclust.clusters  B  M
1 356 209
2   1   0
3   0   2
4   0   1
```

```
[1] "method = complete"
              diagnosis
temp.wisc.hclust.clusters  B  M
1  12 165
2   2   5
3 343  40
4   0   2
```

```
[1] "method = average"
              diagnosis
temp.wisc.hclust.clusters  B  M
1 355 209
2   2   0
3   0   1
4   0   2
```

```
[1] "method = ward.D2"
              diagnosis
```

```
temp.wisc.hclust.clusters  B    M
                        1    0 115
                        2    6  48
                        3 337  48
                        4   14   1
```

When clustering into 4 groups, the **single** and **average** methods perform very poorly - they put most benign and malignant samples in the same cluster. The **complete** and **ward.D2** clusters perform better, but it's hard to say which is best, since both generate a cluster with mostly benign and a cluster with mostly malignant, and then 2 clusters with very few samples.

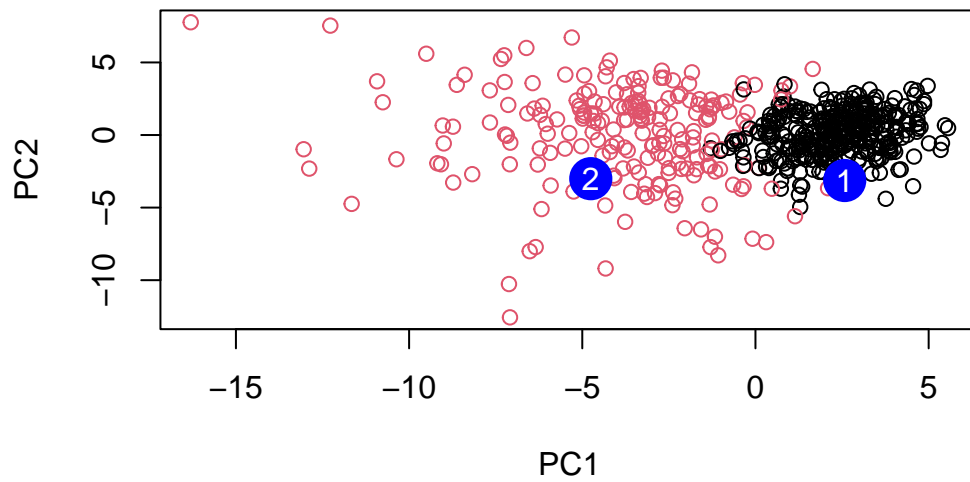
## #6. Prediction

```
#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")
```





**Q16. Which of these new patients should we prioritize for follow up based on your results?**

We should prioritize patient 2 because they are clustering with the patient samples diagnosed malignant.