

# Class 8: Breast Cancer Mini-Project

Andres Vasquez (PID: A16278181)

## About

Today's lab we will work with fine needle aspiration (FNA) of breast mass data from the University of Wisconsin

## Data Import

```
# 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					
842302		0.11890				
842517		0.08902				
84300903		0.08758				
84348301		0.17300				
84358402		0.07678				
843786		0.12440				

How many patients are in this dataset?

```
nrow(wisc.df)
```

```
[1] 569
```

Q1. How many observations are in this dataset?

```
wisc.data <- wisc.df[, -1]  
ncol(wisc.data)
```

```
[1] 30
```

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

```
diagnosis <- as.factor(wisc.df$diagnosis)  
diagnosis
```

```
[1] 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  
[38] B 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 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 B M B B B B B M B 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  
[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 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 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 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 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  
[556] B B B B B B B M M M M M M B
```

Levels: B M

```
table(diagnosis)
```

```
diagnosis  
B      M  
357 212
```

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

```
inds <- grep("_mean", colnames(wisc.data))
length(inds)
```

```
[1] 10
```

```
grep("_mean", colnames(wisc.data), value = T)
```

```
[1] "radius_mean"          "texture_mean"         "perimeter_mean"
[4] "area_mean"           "smoothness_mean"      "compactness_mean"
[7] "concavity_mean"      "concave.points_mean"  "symmetry_mean"
[10] "fractal_dimension_mean"
```

## Initial Analysis

Before analysis I want to take out the expert diagnosis column (aka the answer) from our dataset

```
wisc.data <- wisc.df[,-1]
head(wisc.data)
```

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
842302	17.99	10.38	122.80	1001.0	0.11840
842517	20.57	17.77	132.90	1326.0	0.08474
84300903	19.69	21.25	130.00	1203.0	0.10960
84348301	11.42	20.38	77.58	386.1	0.14250
84358402	20.29	14.34	135.10	1297.0	0.10030
843786	12.45	15.70	82.57	477.1	0.12780
	compactness_mean	concavity_mean	concave.points_mean	symmetry_mean	
842302	0.27760	0.3001		0.14710	0.2419
842517	0.07864	0.0869		0.07017	0.1812
84300903	0.15990	0.1974		0.12790	0.2069
84348301	0.28390	0.2414		0.10520	0.2597
84358402	0.13280	0.1980		0.10430	0.1809
843786	0.17000	0.1578		0.08089	0.2087
	fractal_dimension_mean	radius_se	texture_se	perimeter_se	area_se
842302		0.07871	1.0950	0.9053	8.589 153.40
842517		0.05667	0.5435	0.7339	3.398 74.08
84300903		0.05999	0.7456	0.7869	4.585 94.03
84348301		0.09744	0.4956	1.1560	3.445 27.23

84358402	0.05883	0.7572	0.7813	5.438	94.44
843786	0.07613	0.3345	0.8902	2.217	27.19
	smoothness_se	compactness_se	concavity_se	concave.points_se	
842302	0.006399	0.04904	0.05373	0.01587	
842517	0.005225	0.01308	0.01860	0.01340	
84300903	0.006150	0.04006	0.03832	0.02058	
84348301	0.009110	0.07458	0.05661	0.01867	
84358402	0.011490	0.02461	0.05688	0.01885	
843786	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				
842302	0.11890				
842517	0.08902				
84300903	0.08758				
84348301	0.17300				
84358402	0.07678				
843786	0.12440				

## Clustering

We can try `kmeans()` clustering first

```
km <- kmeans(wisc.data, center = 2)
```

```
table(km$cluster)
```

```
1  2  
131 438
```

Cross table

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

```
diagnosis  
  B  M  
1  1 130  
2 356  82
```

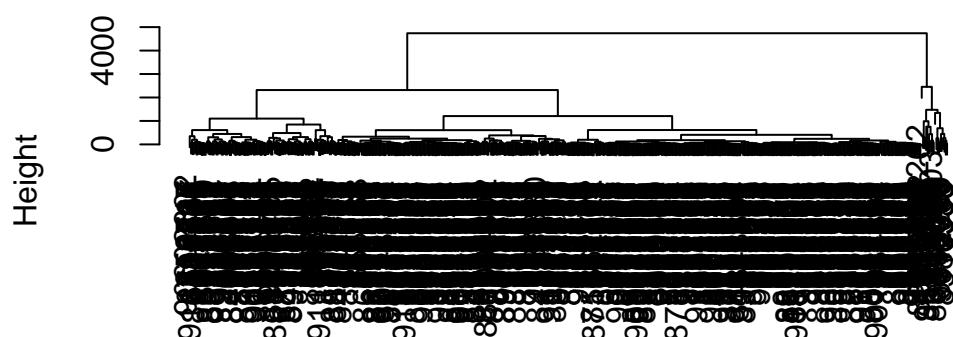
```
table(diagnosis)
```

```
diagnosis  
  B  M  
357 212
```

Let's try `hclust()`. The key input required for `hclust()` is a distance matrix as produced by the `dist()` function.

```
hc <- hclust(dist(wisc.data))  
# I can make a tree life figure (dendrogram)  
plot(hc)
```

## Cluster Dendrogram



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

```
grps <- cutree(hc, k=2)
table(grps)
```

```
grps
  1  2
549 20
```

## PCA

Do we need to scale the data?

We can look at the sd of each column (original variable)

```
round(apply(wisc.data, 2, sd))
```

radius_mean	texture_mean	perimeter_mean
4	4	24
area_mean	smoothness_mean	compactness_mean
352	0	0
concavity_mean	concave.points_mean	symmetry_mean

0	0	0
fractal_dimension_mean	radius_se	texture_se
0	0	1
perimeter_se	area_se	smoothness_se
2	45	0
compactness_se	concavity_se	concave.points_se
0	0	0
symmetry_se	fractal_dimension_se	radius_worst
0	0	5
texture_worst	perimeter_worst	area_worst
6	34	569
smoothness_worst	compactness_worst	concavity_worst
0	0	0
concave.points_worst	symmetry_worst	fractal_dimension_worst
0	0	0

Yes, we need to scale. We will run `prcomp()` with `scale = T`

```
wisc.pr <- prcomp(wisc.data, scale = T)
summary(wisc.pr)
```

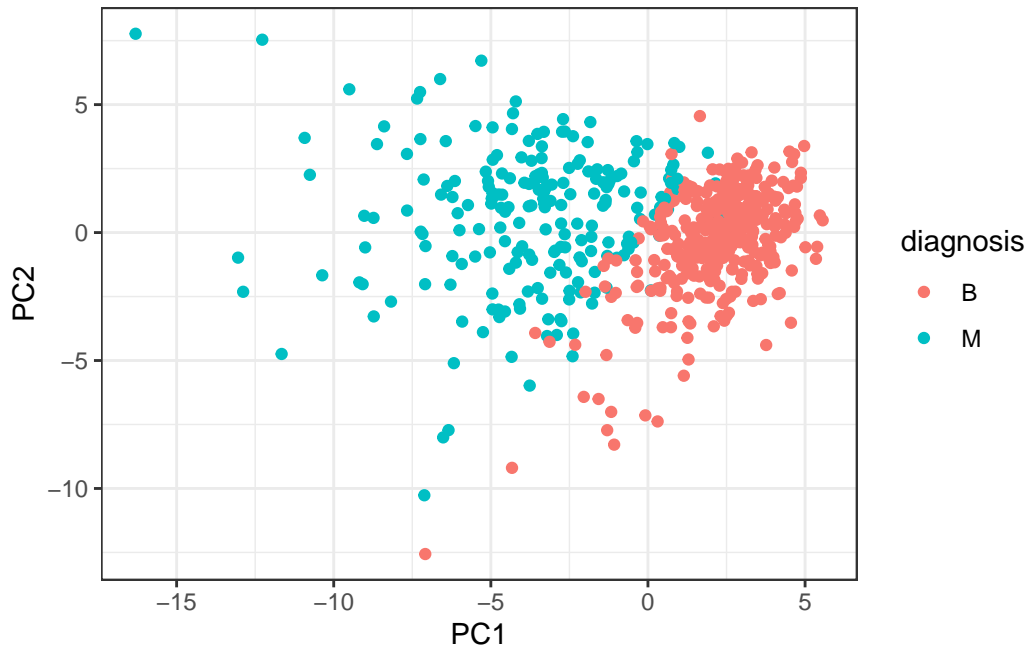
Importance of components:

	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					



Generate our main PCA plot (score plot, PC1 vs PC2, etc)

```
library(ggplot2)
res <- as.data.frame(wisc.pr$x)
ggplot(res) + aes(PC1, PC2, col = diagnosis) + geom_point() + theme_bw()
```



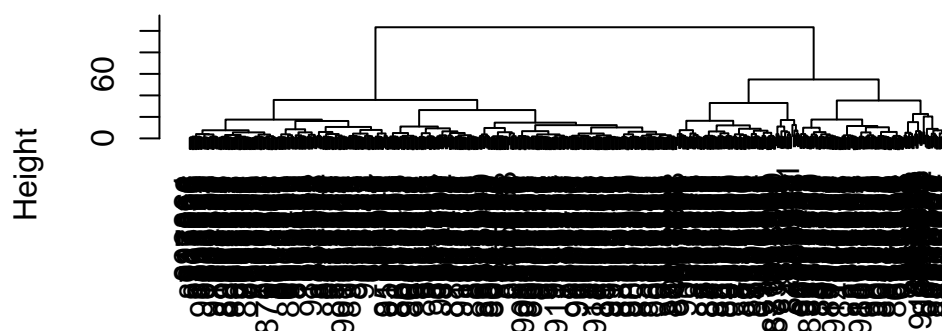
## Combining Methods

Clustering on PCA results

Using the minimum number of principal components required to describe at least 90% of the variability in the data, create a hierarchical clustering model with the linkage method="ward.D2". We use Ward's criterion here because it is based on multidimensional variance like principal components analysis. Assign the results to wisc.pr.hclust.

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

## Cluster Dendrogram



```
d
hclust (*, "ward.D2")
```

To get my clustering result/membership vector I need to “cut” the tree with the `cutree()` function

How many patients in each group?

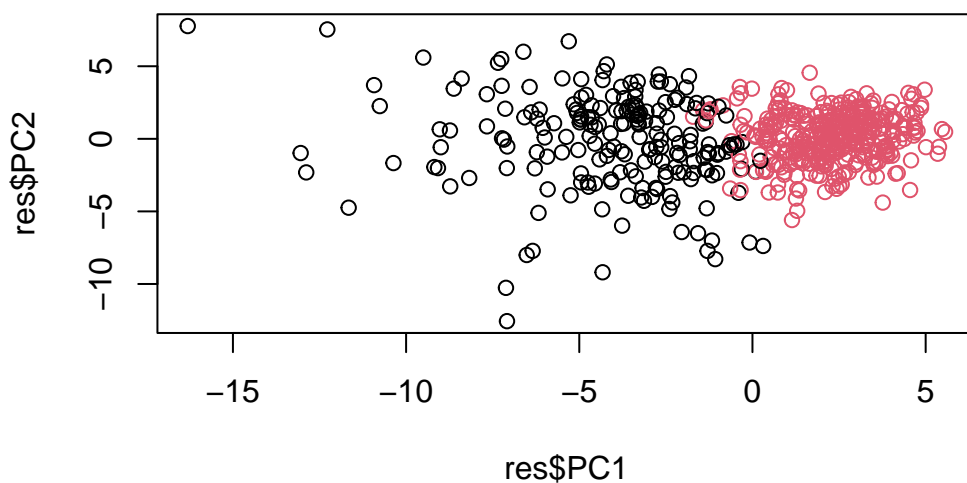
```
grps <- cutree(hc, k=2)
table(grps)
```

```
grps
 1  2
203 366
```

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

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

```
plot(res$PC1,res$PC2, col = grps)
```



## Prediction

We can use our PCA result (model) to do predictions, that is take new unseen data and project it onto our new PC variables

```
#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	

```

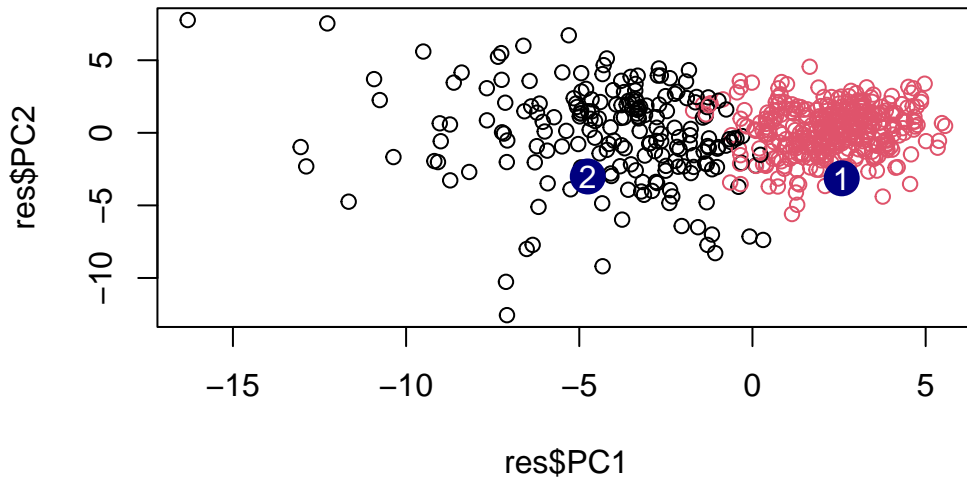
[2,] 0.1299153 0.1448061 -0.40509706 0.06565549 0.25591230 -0.4289500
      PC21      PC22      PC23      PC24      PC25      PC26
[1,] 0.1228233 0.09358453 0.08347651 0.1223396 0.02124121 0.078884581
[2,] -0.1224776 0.01732146 0.06316631 -0.2338618 -0.20755948 -0.009833238
      PC27      PC28      PC29      PC30
[1,] 0.220199544 -0.02946023 -0.015620933 0.005269029
[2,] -0.001134152 0.09638361 0.002795349 -0.019015820

```

```

plot(res$PC1,res$PC2, col = grps)
points(npc[,1], npc[,2], col = "navy", pch = 16, cex = 2.5)
text(npc[,1], npc[,2], c(1,2), col="white")

```



## Summary

Principal Component Analysis (PCA) is a super useful method for analyzing large datasets. It works by finding new variables (PCs) that capture the most variance from the original variables in your datasets. - dimensionality reduction method