

miniproject

Table of contents

Import and Prepare Data	2
PCA	5
Interpreting PCA results	7
Variance Explained	9
Hierarchical Clustering	11
Selecting Number of Clusters	15
Using Different methods	15
Combining methods	21
Sensitivity/Specificity	26
Prediction	26

Presets:

```
library(tidyverse)
```

Warning: package 'tidyverse' was built under R version 4.1.1

-- Attaching packages ----- tidyverse 1.3.1 --

```
v ggplot2 3.3.5    v purrr   0.3.4
v tibble  3.1.3    v dplyr   1.0.7
v tidyr   1.1.4    v stringr 1.4.0
v readr   2.1.1    v forcats 0.5.1
```

Warning: package 'tidyr' was built under R version 4.1.1

Warning: package 'readr' was built under R version 4.1.2

Warning: package 'purrr' was built under R version 4.1.1

Warning: package 'dplyr' was built under R version 4.1.1

Warning: package 'stringr' was built under R version 4.1.1

Warning: package 'forcats' was built under R version 4.1.1

```
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
```

Import and Prepare Data

```
fna.data <- "WisconsinCancer.csv"
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

Get rid of the diagnosis column as it is the “answer” to the question we want to answer with unsupervised learning.

```
wisc.data <- wisc.df[, -1]
```

New diagnosis vector to fill in later:

```
diagnosis <- as.factor(wisc.df[, 1])  
# Check it works  
head(diagnosis)
```

```
[1] M M M M M M
```

```
Levels: B M
```

Q1. How many observations are in the dataset?

```
nrow(wisc.data)
```

```
[1] 569
```

There are 569 rows (or observations) in this dataset.

Q2. How many observations have a malignant diagnosis?

```
table(diagnosis)
```

```
diagnosis  
  B    M  
357 212
```

There are 212 observations with a malignant diagnosis.

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

```
length(grep("_mean", names(wisc.data)))
```

```
[1] 10
```

There are 10 variables suffixed with `_mean`.

PCA

Check if data needs to be scaled

```
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

Do PCA:

```
wisc.pr <- prcomp(wisc.data, scale=T)
# Summarize 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 results, what proportion of original variance is captured by PC1?

From `summary(wisc.pr)` we know PC1 accounts for 44.27% of the original variance.

Q5. How many PCs are required to capture >70% of the variance?

3 PCs (1, 2, 3) are needed, at which point the Cumulative Proportion of variance captured is 72.6%.

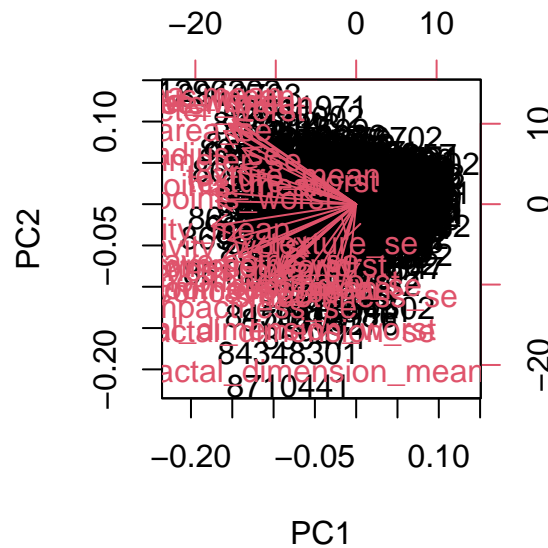
Q6. How many PCs are required to capture >90% of the variance?

The first 7 PCs are needed to capture >90% of the variance.

Interpreting PCA results

Making a biplot

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

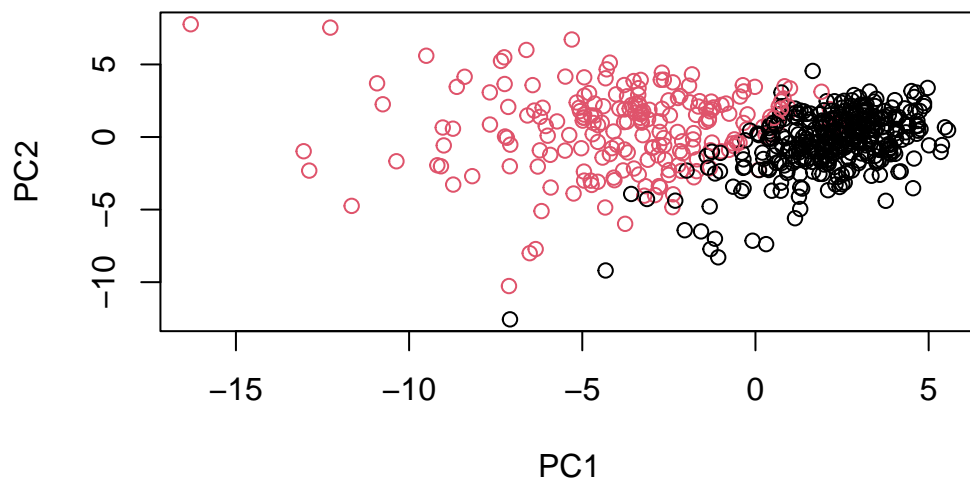


Q7. What stands out about this plot, is it easy to understand?

No the plot is incredibly busy with numbers and variables resulting in overcrowding/overplotting.

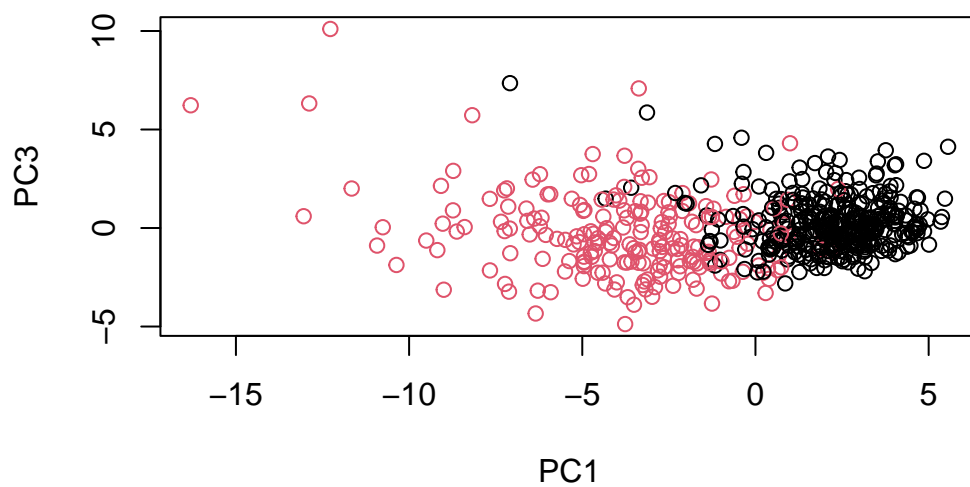
Making a scatter plot:

```
plot(wisc.pr$x, col = diagnosis, xlab = "PC1", ylab = "PC2")
```



Q8. Plot for PC1 and PC3, what do you notice?

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



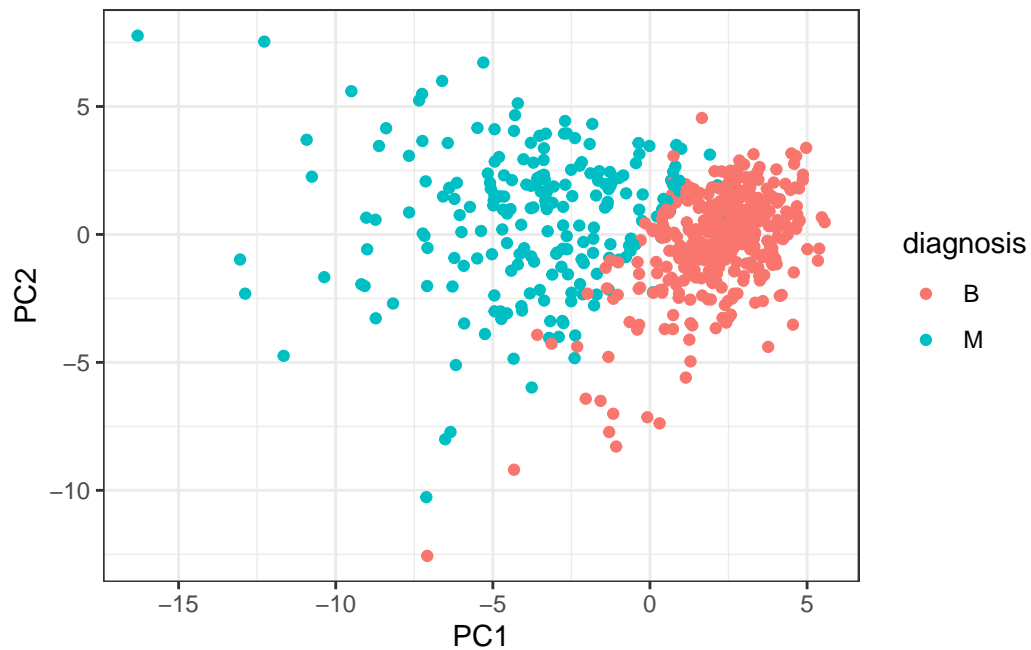
The data overlap more in the PC1/PC3 plot since PC3 explains less variance than PC2.

Using ggplot

```
# make dataframe
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis

# preset theme
theme_set(theme_bw())

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



Variance Explained

Calculate variance (sd squared)

```
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

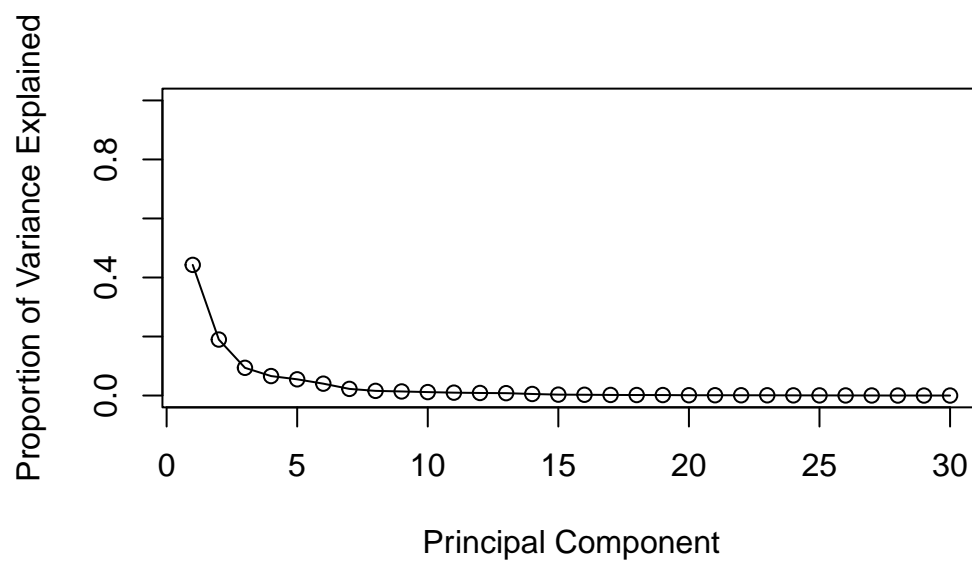
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

Calculate for each PC:

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

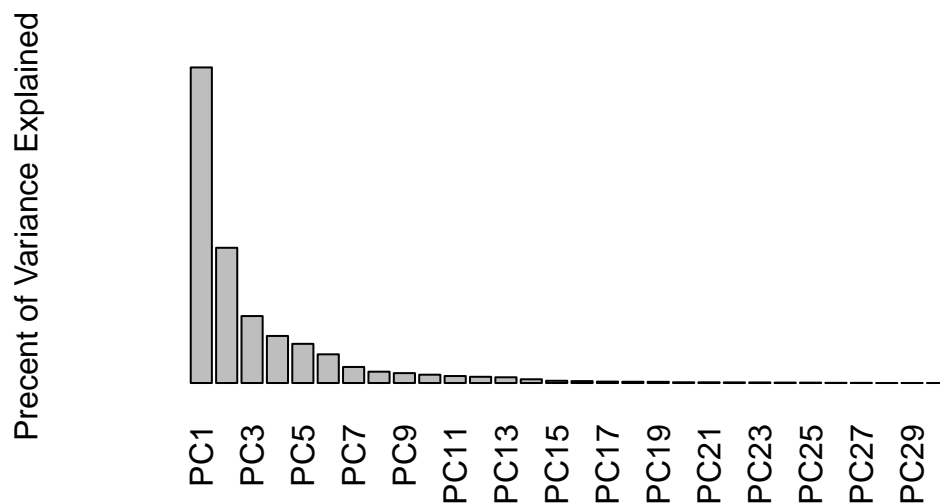
Plot

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



Alternative scree plot

```
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 )
```

Hierarchical Clustering

```
data.scaled <- scale(wisc.data)
head(data.scaled)
```

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
842302	1.0960995	-2.0715123	1.2688173	0.9835095	1.5670875
842517	1.8282120	-0.3533215	1.6844726	1.9070303	-0.8262354
84300903	1.5784992	0.4557859	1.5651260	1.5575132	0.9413821
84348301	-0.7682333	0.2535091	-0.5921661	-0.7637917	3.2806668
84358402	1.7487579	-1.1508038	1.7750113	1.8246238	0.2801253
843786	-0.4759559	-0.8346009	-0.3868077	-0.5052059	2.2354545
	compactness_mean	concavity_mean	concave.points_mean	symmetry_mean	
842302	3.2806281	2.65054179	2.5302489	2.215565542	
842517	-0.4866435	-0.02382489	0.5476623	0.001391139	
84300903	1.0519999	1.36227979	2.0354398	0.938858720	
84348301	3.3999174	1.91421287	1.4504311	2.864862154	

84358402	0.5388663	1.36980615	1.4272370	-0.009552062
843786	1.2432416	0.86554001	0.8239307	1.004517928
	fractal_dimension_mean	radius_se	texture_se	perimeter_se
842302	2.2537638	2.4875451	-0.5647681	2.8305403
842517	-0.8678888	0.4988157	-0.8754733	0.2630955
84300903	-0.3976580	1.2275958	-0.7793976	0.8501802
84348301	4.9066020	0.3260865	-0.1103120	0.2863415
84358402	-0.5619555	1.2694258	-0.7895490	1.2720701
843786	1.8883435	-0.2548461	-0.5921406	-0.3210217
	smoothness_se	compactness_se	concavity_se	concave.points_se
842302	-0.2138135	1.31570389	0.7233897	0.66023900
842517	-0.6048187	-0.69231710	-0.4403926	0.25993335
84300903	-0.2967439	0.81425704	0.2128891	1.42357487
84348301	0.6890953	2.74186785	0.8187979	1.11402678
84358402	1.4817634	-0.04847723	0.8277425	1.14319885
843786	0.1562093	0.44515196	0.1598845	-0.06906279
	symmetry_se	fractal_dimension_se	radius_worst	texture_worst
842302	1.1477468	0.90628565	1.8850310	-1.35809849
842517	-0.8047423	-0.09935632	1.8043398	-0.36887865
84300903	0.2368272	0.29330133	1.5105411	-0.02395331
84348301	4.7285198	2.04571087	-0.2812170	0.13386631
84358402	-0.3607748	0.49888916	1.2974336	-1.46548091
843786	0.1340009	0.48641784	-0.1653528	-0.31356043
	perimeter_worst	area_worst	smoothness_worst	compactness_worst
842302	2.3015755	1.9994782	1.3065367	2.6143647
842517	1.5337764	1.8888270	-0.3752817	-0.4300658
84300903	1.3462906	1.4550043	0.5269438	1.0819801
84348301	-0.2497196	-0.5495377	3.3912907	3.8899747
84358402	1.3373627	1.2196511	0.2203623	-0.3131190
843786	-0.1149083	-0.2441054	2.0467119	1.7201029
	concavity_worst	concave.points_worst	symmetry_worst	
842302	2.1076718	2.2940576	2.7482041	
842517	-0.1466200	1.0861286	-0.2436753	
84300903	0.8542223	1.9532817	1.1512420	
84348301	1.9878392	2.1738732	6.0407261	
84358402	0.6126397	0.7286181	-0.8675896	
843786	1.2621327	0.9050914	1.7525273	
	fractal_dimension_worst			
842302	1.9353117			
842517	0.2809428			
84300903	0.2012142			
84348301	4.9306719			
84358402	-0.3967505			

843786

2.2398308

Calculate Euclidean distance

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

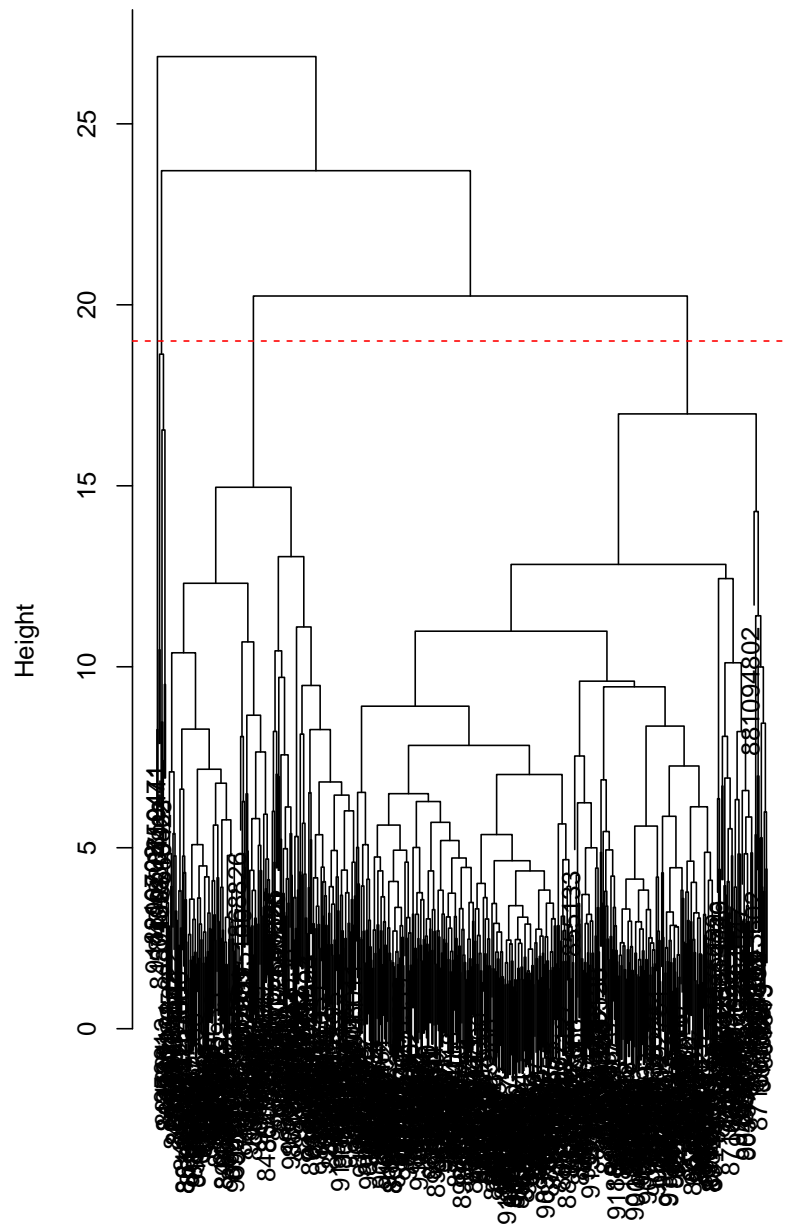
Create model using complete linkage

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

Q10. Plot and find the height where the model has 4 clusters

```
plot(wisc.hclust) + abline(h = 19, col = "red", lty = 2)
```

Cluster Dendrogram



data.dist
hclust (*, "complete")

```
integer(0)
```

Selecting Number of Clusters

```
wisc.hclust.clusters <- cutree(wisc.hclust, h = 19)
```

Compare cluster membership to actual diagnoses.

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

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

Using Different methods

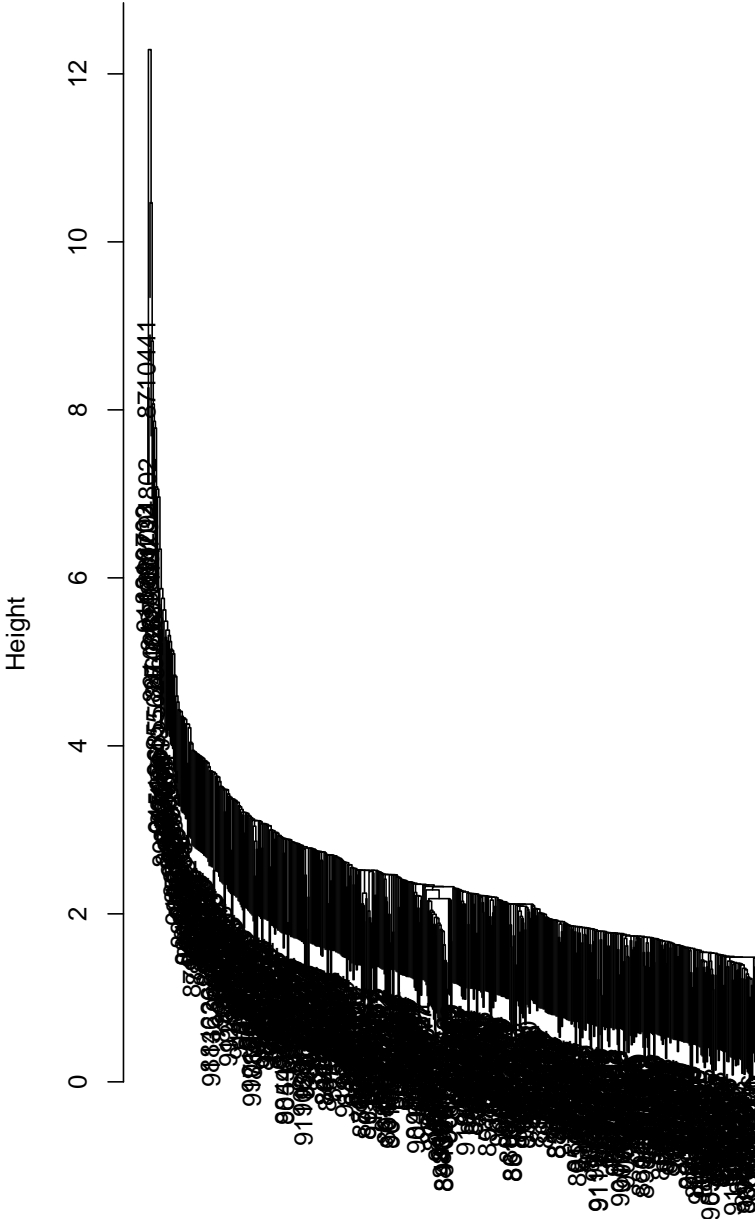
Methods to combine points during hierarchical clustering:

```
wisc.hclust.single <- hclust(data.dist, method = "single")  
wisc.hclust.average <- hclust(data.dist, method = "average")  
wisc.hclust.ward.D2 <- hclust(data.dist, method = "ward.D2")
```

Single method plot

```
plot(wisc.hclust.single) + abline(h = 19, col = "red", lty = 2)
```

Cluster Dendrogram



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

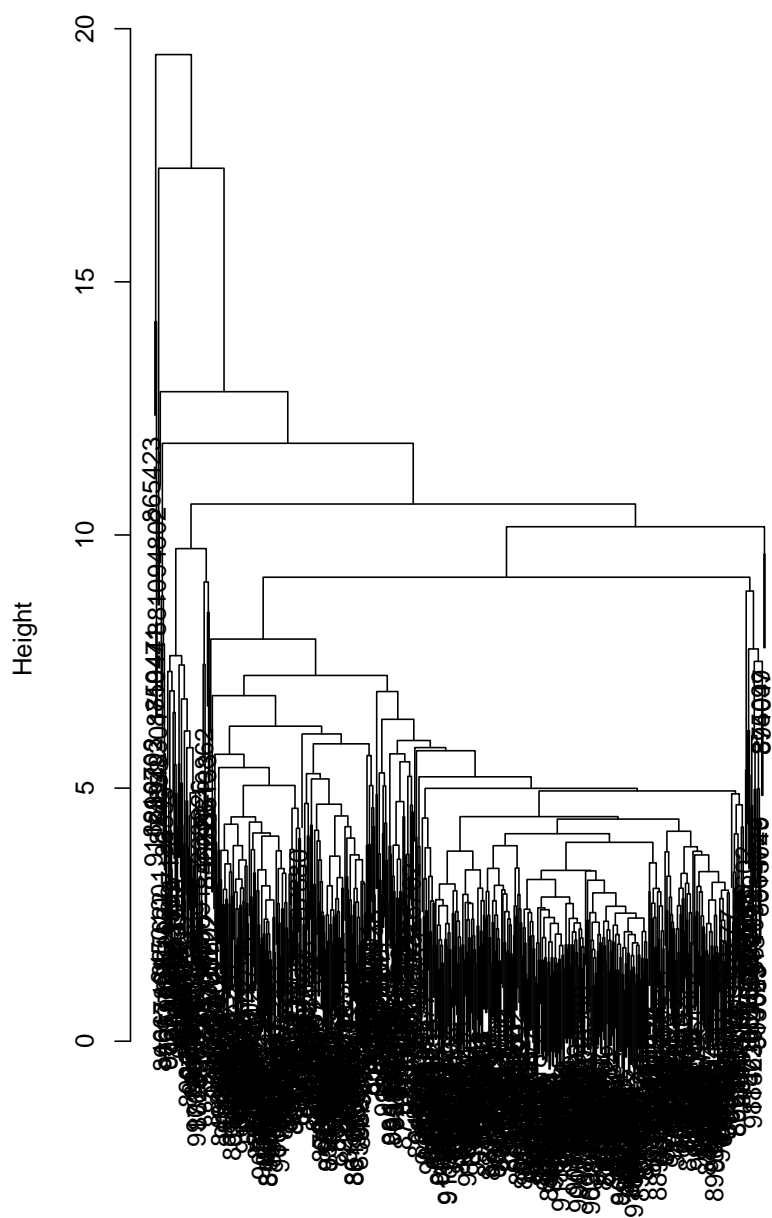


```
integer(0)
```

Average plot

```
plot(wisc.hclust.average)
```

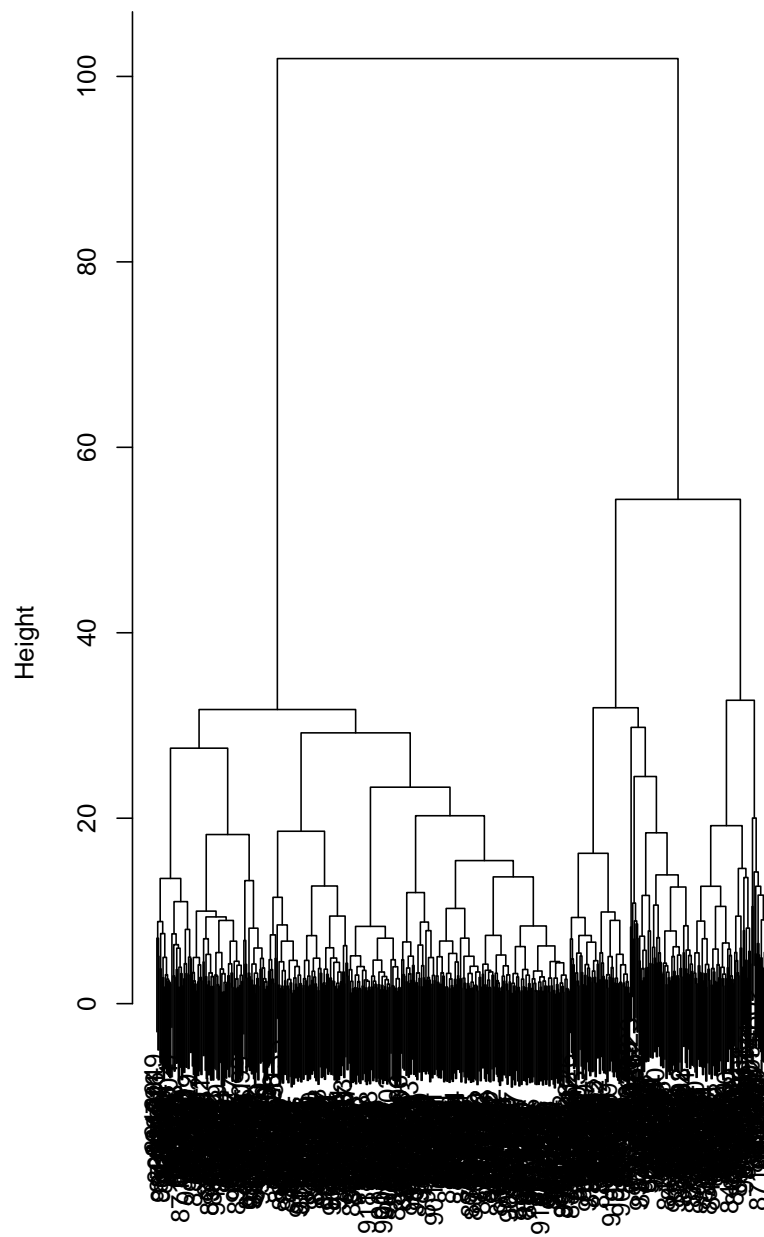
Cluster Dendrogram



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

```
plot(wisc.hclust.ward.D2)
```

Cluster Dendrogram



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

Q12. Which method gives your favorite results from data.dist?

Personally I liked that method ward.D2 because 1) the largest height vertical bars (largest Euclidean distance between sets) are in the first divide. This makes intuitive sense because from the diagnoses results we know there are 2 options, ideally most of the data should be clustered into these two groups. 2) The output tree is the easiest to read since it separates the groups more evenly, instead of separating one observation at a time.

Combining methods

Clustering on PCA results

```
# We only want the first 7 PCs
head(wisc.pr$x[,1:7])
```

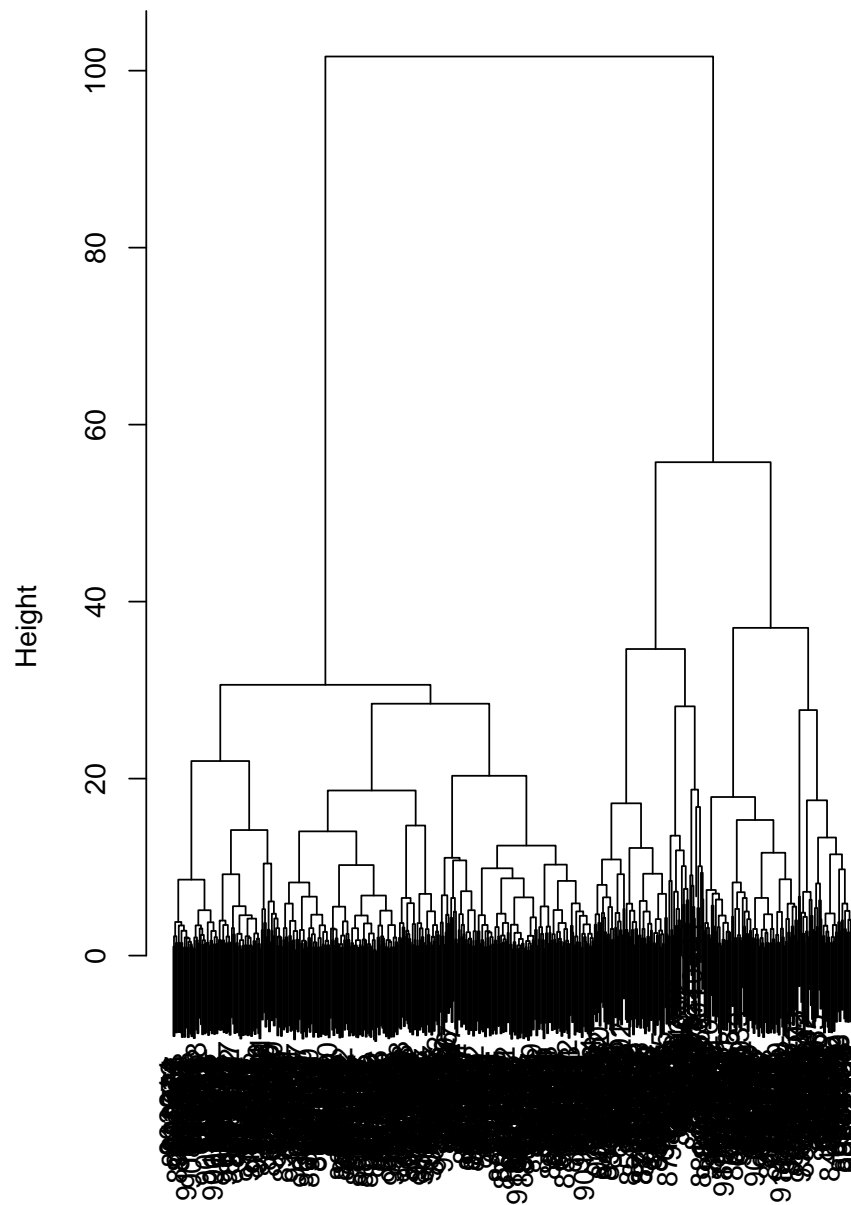
	PC1	PC2	PC3	PC4	PC5	PC6
842302	-9.184755	-1.946870	-1.1221788	3.6305364	1.1940595	1.41018364
842517	-2.385703	3.764859	-0.5288274	1.1172808	-0.6212284	0.02863116
84300903	-5.728855	1.074229	-0.5512625	0.9112808	0.1769302	0.54097615
84348301	-7.116691	-10.266556	-3.2299475	0.1524129	2.9582754	3.05073750
84358402	-3.931842	1.946359	1.3885450	2.9380542	-0.5462667	-1.22541641
843786	-2.378155	-3.946456	-2.9322967	0.9402096	1.0551135	-0.45064213
	PC7					
842302	2.15747152					
842517	0.01334635					
84300903	-0.66757908					
84348301	1.42865363					
84358402	-0.93538950					
843786	0.49001396					

```
# Find Euclidean distance
pca.dist <- dist(wisc.pr$x[,1:7])
# do clustering
wisc.pr.hclust <- hclust(pca.dist, method = "ward.D2")
```

Plot

```
plot(wisc.pr.hclust)
```

Cluster Dendrogram



pca.dist
hclust (*, "ward.D2")

Are the two main clusters the malignant and benign groups?

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

```
grps
 1  2
216 353
```

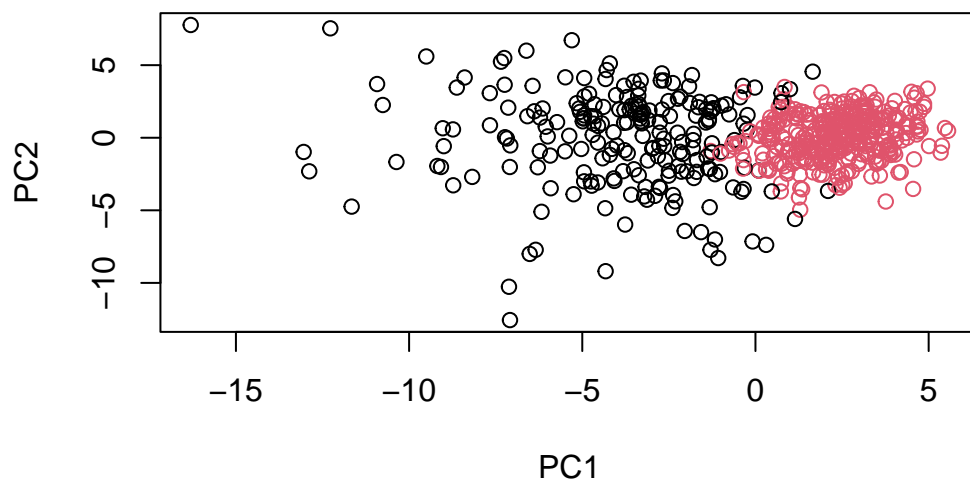
Match to diagnoses:

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

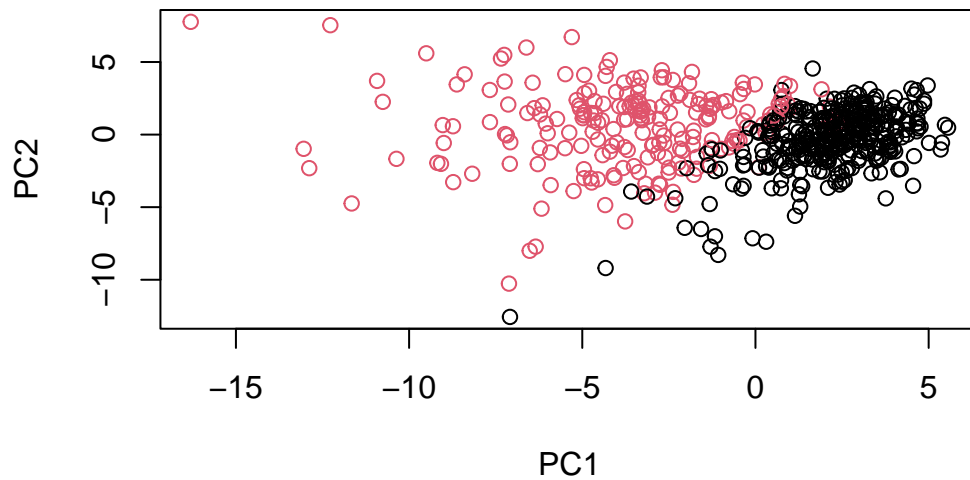
```
      diagnosis
grps   B    M
 1   28 188
 2  329   24
```

Visualize

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



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



Turn groups into factor

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

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

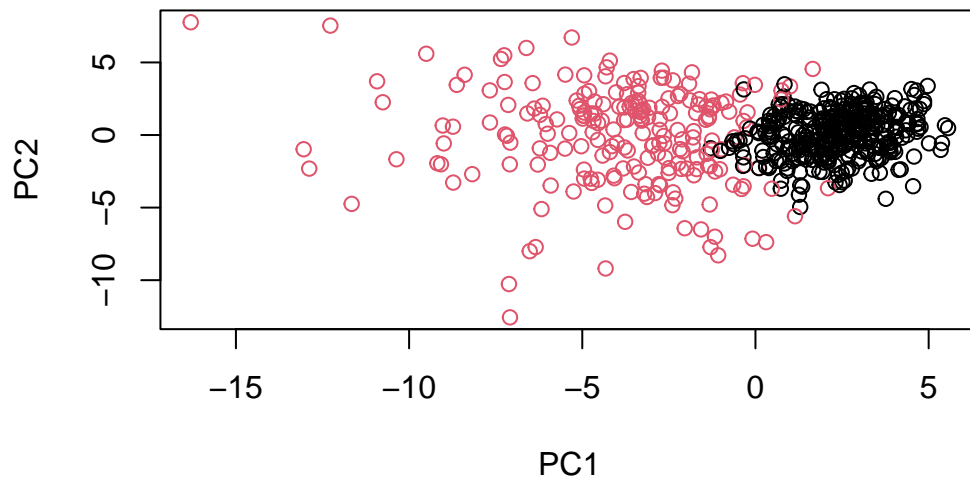
Reorder

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

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

New plot:

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

```
# Use the distance of first 7 PCs to cluster
wisc.pr.hclust <- hclust(pca.dist, method="ward.D2")
# cluster into 2 groups
wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k = 2)
# compare results from model with known diagnoses
table(wisc.pr.hclust.clusters, diagnosis)
```

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

Q13. What about clustering with 4 groups, do the results separate out the diagnoses well?

```
wisc.pr.hclust.clusters4 <- cutree(wisc.pr.hclust, k = 4)
table(wisc.pr.hclust.clusters4, diagnosis)
```

```
          diagnosis
wisc.pr.hclust.clusters4  B  M
1         0 45
```

```

2    2  77
3   26  66
4  329  24

```

Q14. How well do hierarchical clustering models from previous sections (before PCA) do in separating diagnoses?

```
head(wisc.hclust.clusters)
```

```

842302    842517 84300903 84348301 84358402    843786
      1          1          1          2          1          1

```

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

```

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

```

In this method there are fewer Malignant diagnoses in their own cluster. PCA clustering worked better.

Sensitivity/Specificity

Q15. Which analysis had best sensitivity? Which had best specificity?

Prediction

```

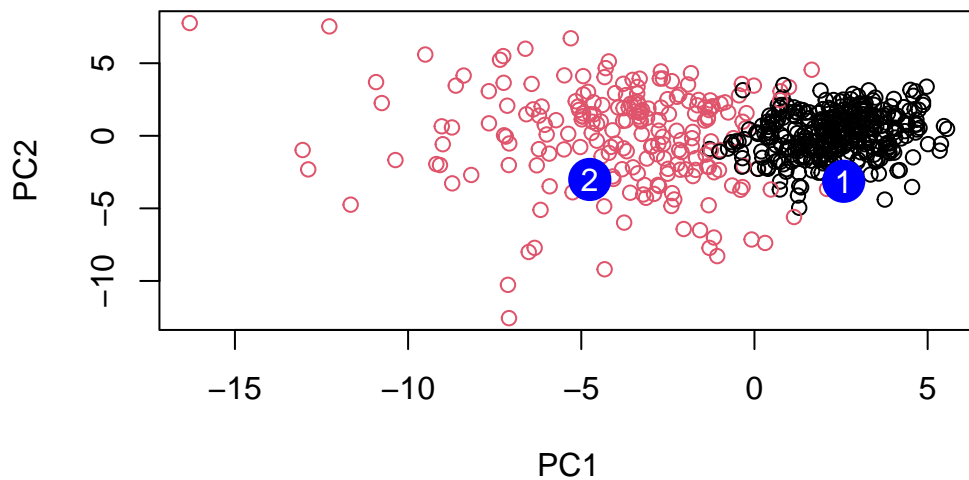
url <- "https://tinyurl.com/new-samples-CSV"
newdata <- read.csv(url)
npc <- predict(wisc.pr, newdata = newdata)
npc

```

	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			

Plotting new cancer data on the old PCA model

```
plot(wisc.pr$x[,1:2], col = g) + points(npc[,1], npc[,2], col = "blue", pch = 16, cex = 3)
```



integer(0)

Patient 2 should be prioritized as their data is clearly within the malignant cluster.

```
sessionInfo()
```

```
R version 4.1.0 (2021-05-18)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)
```

```
Matrix products: default
```

```
locale:
```

```
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] forcats_0.5.1  stringr_1.4.0  dplyr_1.0.7    purrr_0.3.4
[5] readr_2.1.1    tidyr_1.1.4    tibble_3.1.3   ggplot2_3.3.5
[9] tidyverse_1.3.1
```

```
loaded via a namespace (and not attached):
```

```
[1] tidymodels_1.1.1 xfun_0.29      haven_2.4.3    colorspace_2.0-2
[5] vctrs_0.3.8      generics_0.1.1 htmltools_0.5.4 yaml_2.2.2
[9] utf8_1.2.2       rlang_0.4.11   pillar_1.6.5    glue_1.4.2
[13] withr_2.4.3      DBI_1.1.2      dbplyr_2.1.1    modelr_0.1.8
[17] readxl_1.3.1     lifecycle_1.0.1 munsell_0.5.0   gtable_0.3.0
[21] cellranger_1.1.0 rvest_1.0.2    evaluate_0.14   labeling_0.4.2
[25] knitr_1.37       tzdb_0.2.0     fastmap_1.1.0   fansi_0.5.0
[29] broom_0.7.11     Rcpp_1.0.7     scales_1.1.1    backports_1.4.1
[33] jsonlite_1.7.3   farver_2.1.0   fs_1.5.2        hms_1.1.1
[37] digest_0.6.27    stringi_1.7.6  grid_4.1.0      cli_3.0.1
[41] tools_4.1.0      magrittr_2.0.1 crayon_1.4.2     pkgconfig_2.0.3
[45] ellipsis_0.3.2   xml2_1.3.3     reprex_2.0.1    lubridate_1.8.0
[49] rstudioapi_0.13  assertthat_0.2.1 rmarkdown_2.11  httr_1.4.2
[53] R6_2.5.1         compiler_4.1.0
```