

Class08 - PCA Mini Project

Dylan Mullaney (PID: A16869792)

Today we will do a complete analysis of some breast cancer biopsy data. First, let's revisit the main PCA function 'prcomp()' and see what 'scale=TRUE/FALSE' does.

```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Find the mean value per column.

```
apply(mtcars, 2, sd)
```

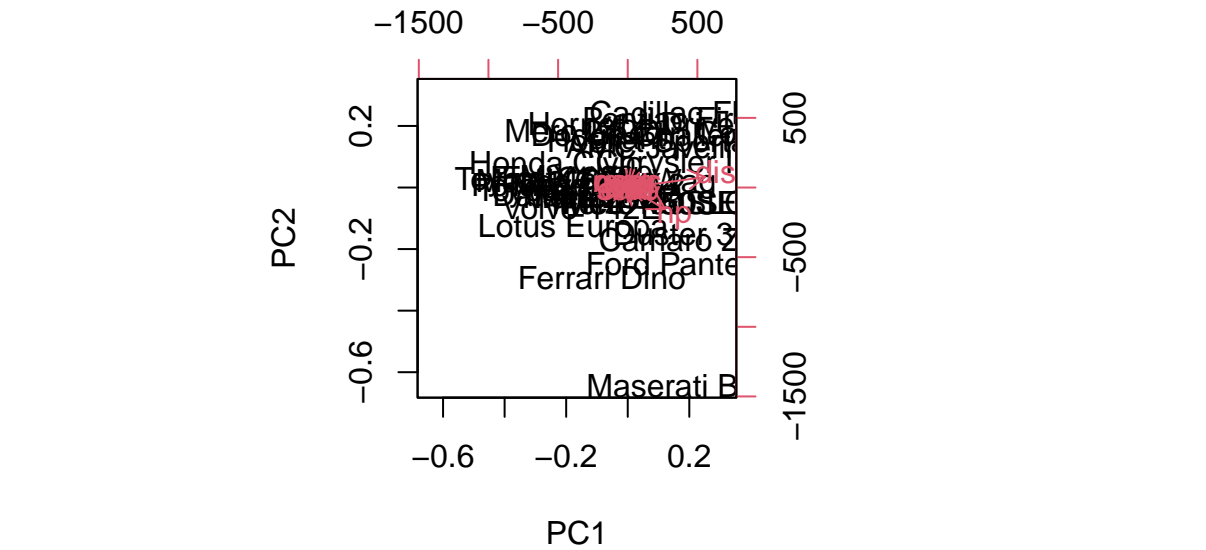
mpg	cyl	disp	hp	drat	wt
6.0269481	1.7859216	123.9386938	68.5628685	0.5346787	0.9784574
qsec	vs	am	gear	carb	
1.7869432	0.5040161	0.4989909	0.7378041	1.6152000	

```
#2 refers to columns
```

It's clear that displacement (disp) and horsepower (hp) have the highest mean values and highest standard deviations. They will likely dominate any analysis I do on this dataset.

```
pc.noscale <- prcomp(mtcars)
pc.scale <- prcomp(mtcars, scale=TRUE)
```

```
biplot(pc.noscale)
```



```
pc.noscale$rotation [,1]
```

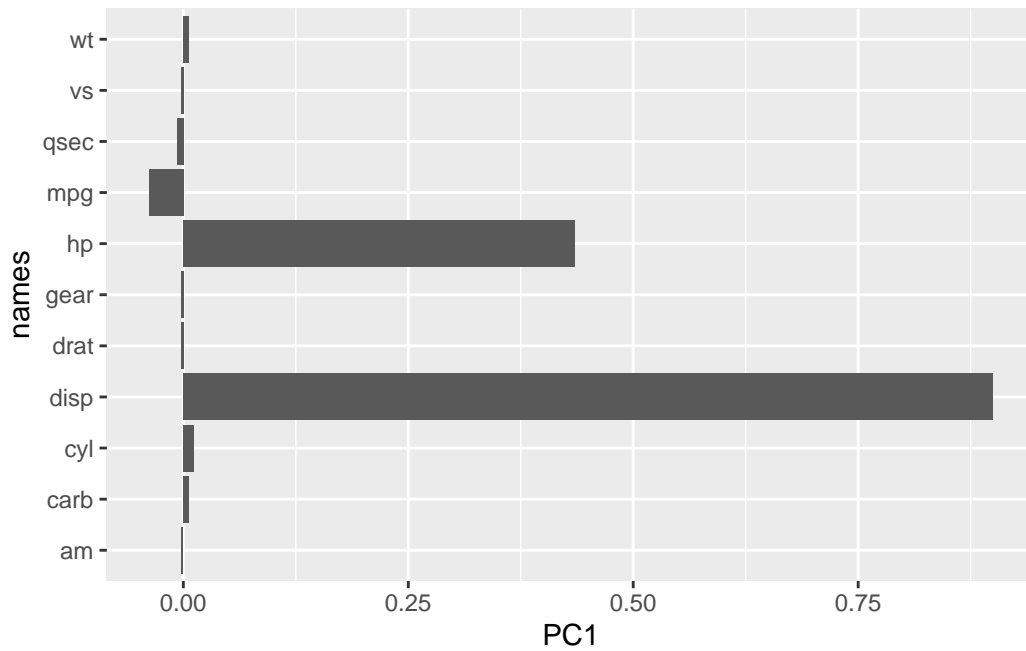
mpg	cyl	disp	hp	drat	wt
-0.038118199	0.012035150	0.899568146	0.434784387	-0.002660077	0.006239405
qsec	vs	am	gear	carb	
-0.006671270	-0.002729474	-0.001962644	-0.002604768	0.005766010	

Plot the loadings

```
library(ggplot2)

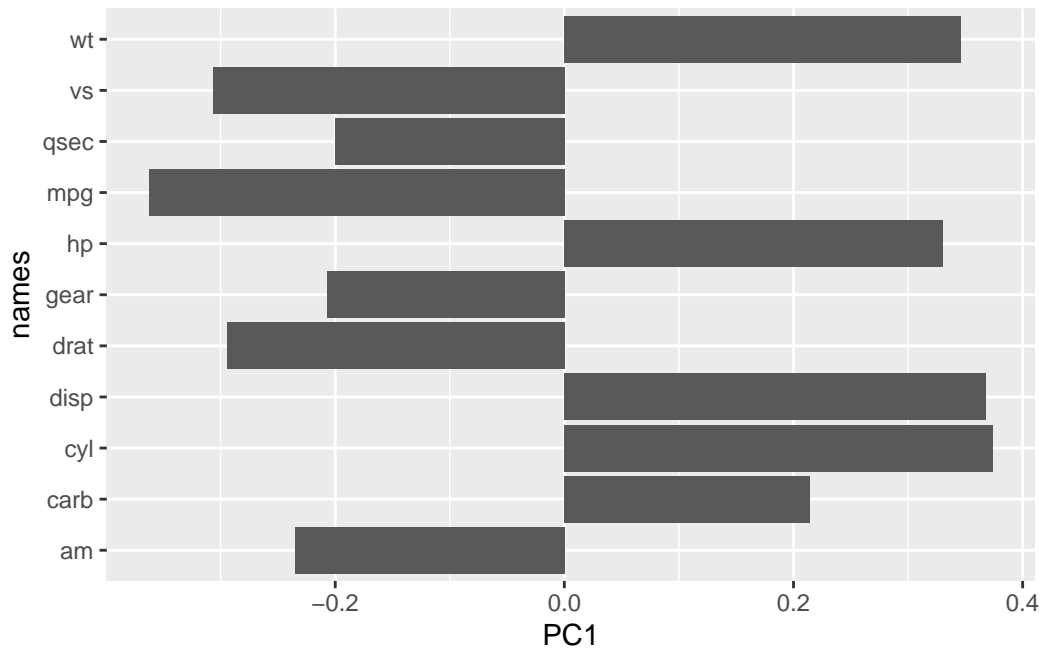
r1<- as.data.frame(pc.noscale$rotation)
r1$names <- row.names (pc.noscale$rotation)

ggplot (r1) +
  aes(PC1, names) +
  geom_col()
```

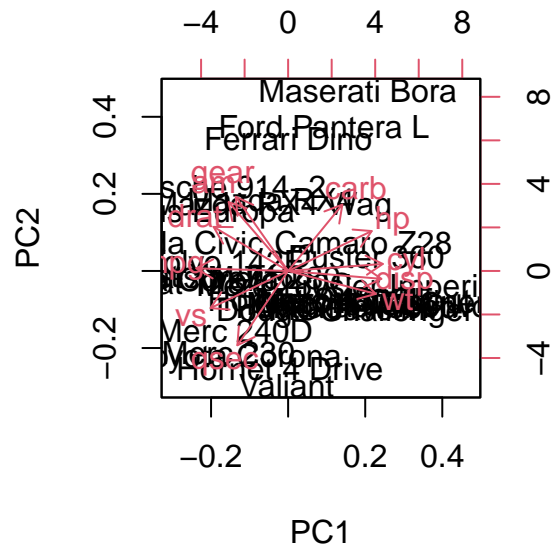


```
r2<- as.data.frame(pc.scale$rotation)
r2$names <- row.names (pc.scale$rotation)

ggplot (r2) +
  aes(PC1, names) +
  geom_col()
```



```
biplot(pc.scale)
```



Take-home: Generally, you always want to set `scale=TRUE` when we do this type

of analysis to avoid having our analysis be dominated by individual variables with the largest variance due to their unit of measurement.

FNA breast cancer data

Load the data into R, save data to same folder for it to be able to read.

```
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

Q1. How many observations are in this dataset?

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

```
[1] 569
```

```
#how many people are being measured?
```

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

```
wisc.df$diagnosis == "M" -> 'is this values equal to M, returns TF'
sum (wisc.df$diagnosis == "M")
```

```
[1] 212
```

The `table()` function is quicker and easier

```
table(wisc.df$diagnosis)
```

```
      B      M  
357 212
```

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

```
ncol(wisc.df)
```

```
[1] 31
```

```
colnames(wisc.df)
```

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

A useful function for this is `grep()`

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

```
[1] 2 3 4 5 6 7 8 9 10 11
```

```
#If you use sum, it will add the values of the numerical column outputs  
length(grep ("_mean", colnames(wisc.df)))
```

```
[1] 10
```

Before we go any further we need to exclude the diagnosis column from future analysis as to not bias our predictions. Diagnosis give the ‘answer’, benign versus malignant.

```
#Store this data separately so we don't lose it  
diagnosis <- as.factor(wisc.df$diagnosis)  
#Store as a factor because the variable has multiple factors  
head(diagnosis)
```

```
[1] M M M M M M  
Levels: B M
```

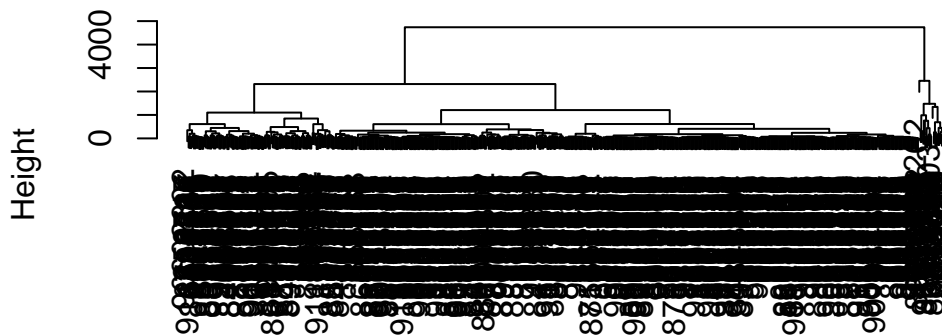
Create data minus diagnosis

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

Let’s see if we can cluster the ‘wisc.data’ to find some structure in the dataset.

```
hc <- hclust(dist(wisc.data))  
plot(hc)
```


Cluster Dendrogram



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

Principal Component Analysis (PCA)

Q4. From your results, what proportion of the original variance is captured by the first principal components (PC1)? 44% is captured in PC1, which is significantly higher than the other PCs.

Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data? Three - could be a combination of PC1 (44%), PC2 (19%), and PC3 (9%).

Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data? You would need to incorporate 7 PCs.

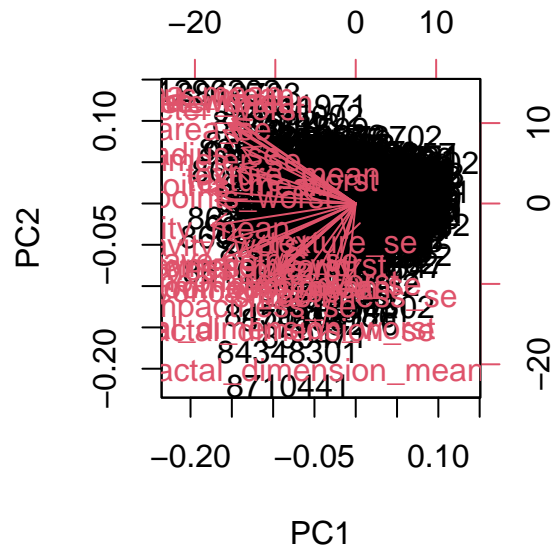
```
wisc.pr <- prcomp (wisc.data, scale=T)
summary(wisc.pr)
```

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					

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



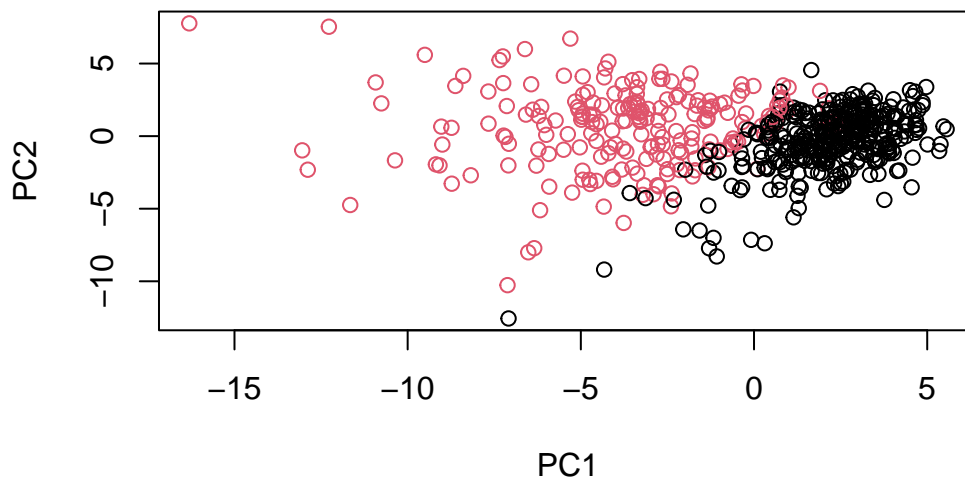
Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why? This plot has far too much information to interpret. I noticed that all the values seem to clump in a circle.

This biplot suck. We need to build our own PCA score plot of PC1 v PC2.

```
attributes(wisc.pr)
```

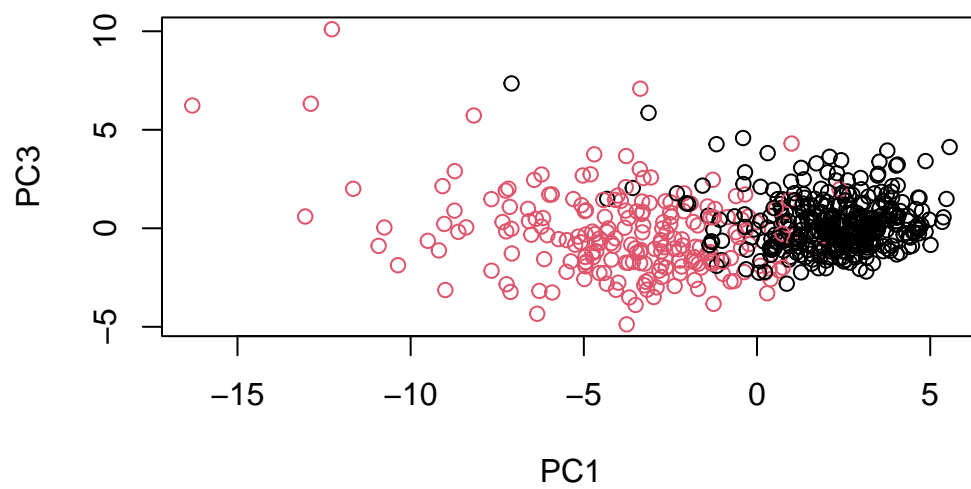
```
$names  
[1] "sdev"      "rotation" "center"    "scale"     "x"  
  
$class  
[1] "prcomp"
```

```
plot(wisc.pr$x[,1], wisc.pr$x[,2], col=diagnosis,  
      xlab= "PC1", ylab="PC2")
```



Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots? The shape of these plots feels very similar. The points appear to have moved slightly down and right as a group. There is slightly more overlap in the center of the dots.

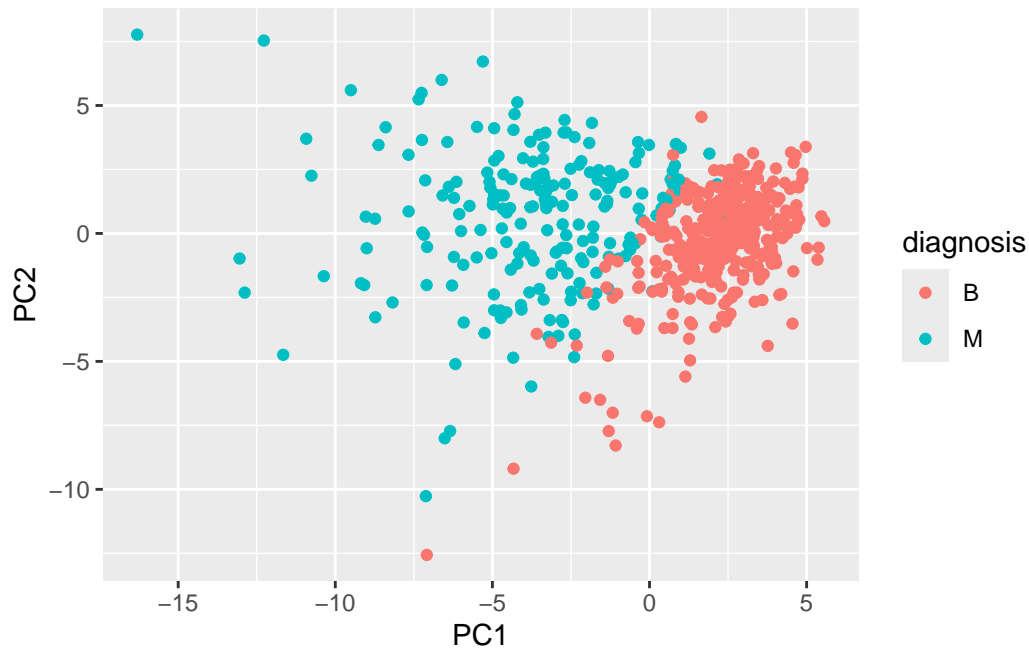
```
plot(wisc.pr$x[,1], wisc.pr$x[,3], col=diagnosis,  
      xlab= "PC1", ylab="PC3")
```



This is nice, make a ggplot of this data

```
pc <- as.data.frame (wisc.pr$x)

ggplot(pc) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```



PCA tries to find meaningful ways to flatten data with many inputs so that it can be analyzed reasonably.

Variance

Calculate the variance of each component. This is the variability among a group of data. This is done by squaring the stdev of the data.

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

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

Calculate the variance explained by each PC. This is done by dividing total variance by the PCs.

```
pve <- pr.var/wisc.pr$x
head(pve)
```

```
PC1      PC2      PC3      PC4      PC5
842302 -1.4460492 -0.0000683378 -6.672761e-04 4.377694e-04 5.778995e-03
```

842517	-2.3856094	3.5277835000	-2.515846e-04	6.702014e-04	-2.558379e-03	
84300903	-0.4918869	5.2980852242	-2.409307e+01	1.459976e-04	4.232194e-03	
84348301	-0.2783092	-0.2744785181	-1.762058e+00	8.714227e+01	4.497378e-05	
84358402	-0.4193277	1.0176131423	2.029426e+00	1.937117e+00	-2.431341e+01	
843786	-0.5076863	-0.4177749272	-6.754571e-01	2.997150e+00	5.394068e+00	
	PC6	PC7	PC8	PC9	PC10	
842302	5.798989e-03	0.0071756550	0.045357845	-0.155056705	-0.03130099	
842517	2.410124e-01	0.6127248096	-0.064297242	-0.025383862	0.02200756	
84300903	2.937907e-03	-0.0103365489	-0.084055881	0.643859521	0.03977960	
84348301	2.454499e-04	0.0011124725	-0.006518273	-0.005823683	-0.01387219	
84358402	-1.085711e-04	-0.0008005254	-0.002499680	-0.026180411	0.02166995	
843786	-2.947263e+01	0.0002715123	0.004530007	-0.011918029	-0.01302061	
	PC11	PC12	PC13	PC14	PC15	
842302	-0.11408498	-0.03630535	0.47898487	-0.07623736	0.09879027	
842517	-0.03377553	0.18996185	-0.03305337	-0.07578117	-5.86806264	
84300903	0.04022803	0.22079102	-0.07305729	1.86974044	-0.10243919	
84348301	0.01569317	0.02408966	-0.02942719	-0.06154745	0.18470370	
84358402	-0.02378854	-0.16351466	0.06279774	-0.05093571	-0.09667220	
843786	-0.07456600	0.19025796	-0.68777438	7.76683989	-0.15376730	
	PC16	PC17	PC18	PC19	PC20	PC21
842302	0.10727604	-0.3549083	-0.2859508	1.8058935	0.7564170	3.0479681
842517	-0.09163148	-4.6439767	0.2960056	-0.6347752	-2.1165631	-3.3803143
84300903	0.16199132	0.3113963	-0.9085892	-0.2399681	-0.7683282	0.7765854
84348301	0.96400407	0.1091201	-1.6571898	-2.9884186	-0.2027350	0.3619306
84358402	-0.20436194	0.3719413	-2.8141714	0.1288398	1.2204381	-0.8084247
843786	-2.35901940	0.1584000	-0.1664360	-0.4056133	-0.8345546	-33.2736930
	PC22	PC23	PC24	PC25	PC26	PC27
842302	-5.0980745	4.936921	2.721937	4.474997	-5.997014	-6.533172
842517	3.1103735	-1.612186	-36.958129	2.797700	-16.431661	6.666397
84300903	4.3341960	-3.959750	-3.415687	-2.437872	100.738130	13.633689
84348301	1.1884435	-2.106216	-1.917324	-4.529310	-1.516076	2.600192
84358402	8.9036390	1.732259	49.024962	-96.070644	8.941750	12.971387
843786	0.9312878	4.694117	-85.052439	-2.135717	-9.709053	-4.156081
	PC28	PC29	PC30			
842302	-58.45246	61.78684	120.76443			
842517	50.58154	-348.55554	1509.96537			
84300903	25.69692	524.55038	-2641.29438			
84348301	15.90738	-17.43882	82.76806			
84358402	-13.71338	134.14575	-56.96390			
843786	571.35589	-24.18886	-195.35171			

```
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

```
attributes(wisc.pr)
```

```
$names
```

```
[1] "sdev"      "rotation" "center"   "scale"    "x"
```

```
$class
```

```
[1] "prcomp"
```

```
# Variance isn't one of them, so we need to manipulate sdev to create variance
```

```
#pr.var <- wisc.pr$sdev^2
```

```
#pve <- round(pr.var/sum(pr.var),2)
```

```
#pve
```

```
#Round to 2 decimal points to make more sense of the results
```

```
#cumsum(pve)
```

```
#Cumulative Sum. This adds up the total variance gotten by each PC. Ie by the 4th PC, 79% of
```

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

```
pve
```

```
[1] 4.427203e-01 1.897118e-01 9.393163e-02 6.602135e-02 5.495768e-02  
[6] 4.024522e-02 2.250734e-02 1.588724e-02 1.389649e-02 1.168978e-02  
[11] 9.797190e-03 8.705379e-03 8.045250e-03 5.233657e-03 3.137832e-03  
[16] 2.662093e-03 1.979968e-03 1.753959e-03 1.649253e-03 1.038647e-03  
[21] 9.990965e-04 9.146468e-04 8.113613e-04 6.018336e-04 5.160424e-04  
[26] 2.725880e-04 2.300155e-04 5.297793e-05 2.496010e-05 4.434827e-06
```

```
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

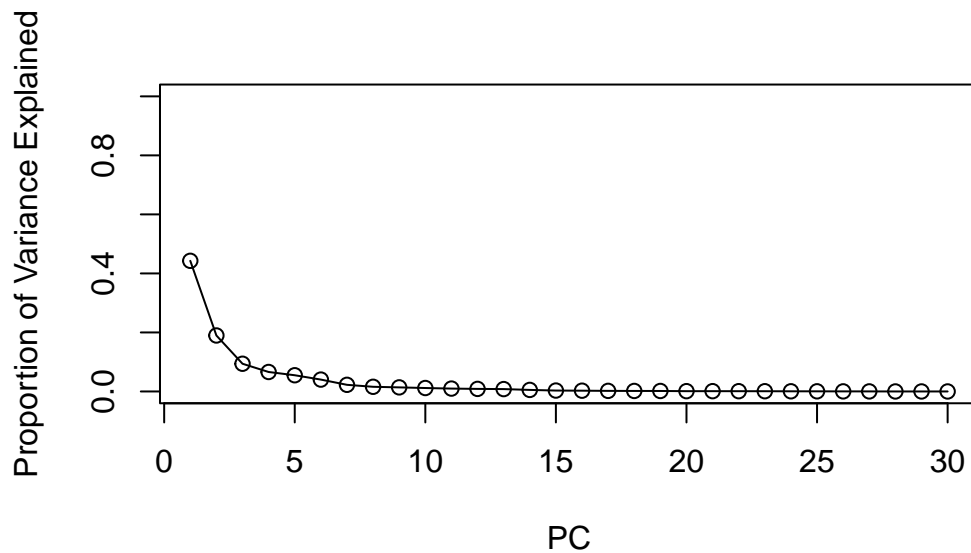
	PC29	PC30
Standard deviation	0.02736	0.01153
Proportion of Variance	0.00002	0.00000
Cumulative Proportion	1.00000	1.00000

```
# compare
```



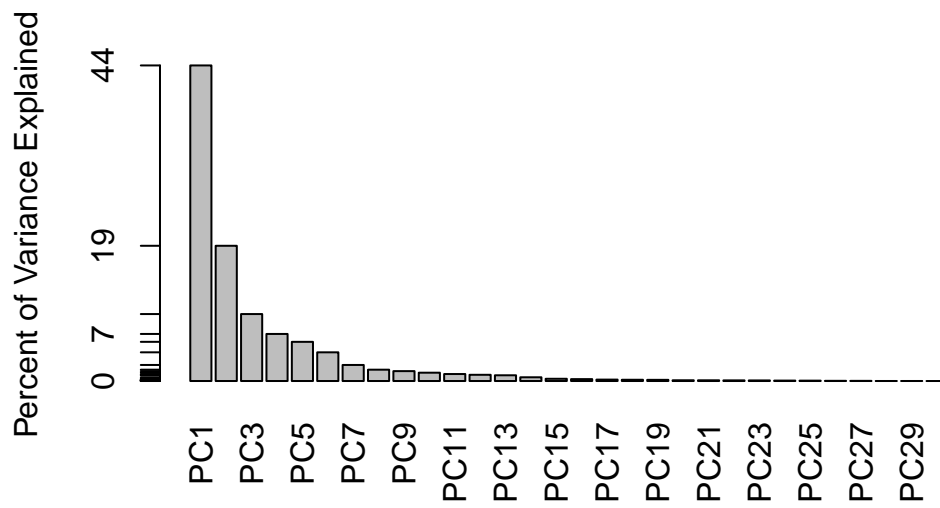
```
# When you call plot, check your data type and how many inputs you're giving it

plot(pve, xlab= "PC",
      ylab= "Proportion of Variance Explained",
      ylim = c(0,1), type= "o")
```



Another plot of the same data

```
barplot(pve, ylab = "Percent 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`?

```
#isolate pc1, find the vector you fed in to the equation to create the PCA that contributes 1
comp.load.vect <- wisc.pr$rotation["concave.points_mean",1]
comp.load.vect
```

```
[1] -0.2608538
```

Q10. What is the minimum number of principal components required to explain 80% of the variance of the data? 5 principal components

```
pve <- round(pr.var/sum(pr.var),2)
pve
```

```
[1] 0.44 0.19 0.09 0.07 0.05 0.04 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.00
[16] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```
cumsum (pve)
```

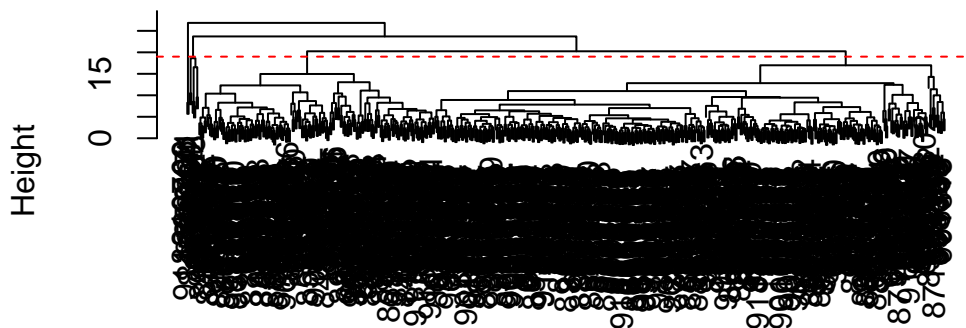
```
[1] 0.44 0.63 0.72 0.79 0.84 0.88 0.90 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.98  
[16] 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98
```

Hierarchical Clustering

First, scale the data. Calculate distances between all pairs of observations. Create a hierarchical clustering model with complete linkage.

```
data.scaled <- scale(wisc.data)  
  
data.dist <- dist(data.scaled)  
  
wisc.hclust <- hclust (data.dist, method ="complete")  
  
plot(wisc.hclust)  
abline(h=19 , col="red", lty=2)
```

Cluster Dendrogram



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

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

H=19

Selecting number of clusters

Use 'cutree()' to cut into 4 clusters and compare with diagnosis data.

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

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

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

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

You need at least 4 clusters to separate benign and malignant data - anything less yields almost all data in cluster 1. I like the distribution at 8 clusters as it separates the large group of 'M' results in cluster 1 in two different groups, one of which has no 'B' results.

```
wisc.hclust.clusters2 <- cutree(wisc.hclust, 8)

table(wisc.hclust.clusters2, diagnosis)
```

	diagnosis	
wisc.hclust.clusters2	B	M
1	12	86
2	0	79
3	0	3
4	331	39
5	2	0
6	12	1
7	0	2
8	0	2

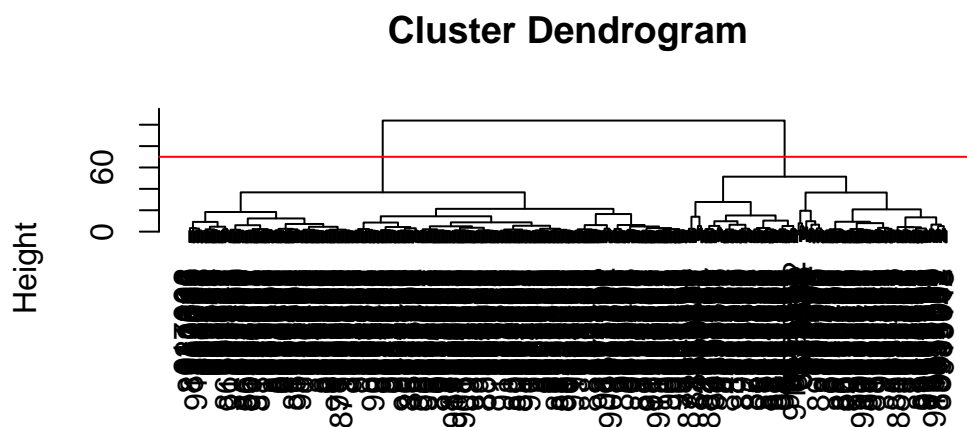
Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning. Options: "single", "complete", "average" and "ward.D2"

I like the ward.D2 method because it creates a smaller number of cleanly separated groups. This makes it easier to make more generalized predictions by the split between defined groups.

Clustering in PC space

```
hc <- hclust(dist(wisc.pr$x[,1:2]), method= "ward.D2")
plot(hc)

abline(h=70, col="red")
```



```
dist(wisc.pr$x[, 1:2])
hclust (*, "ward.D2")
```

Cluster membership vector, clustering on PCA results

```
grps <- cutree(hc, h=70)
table(grps)
```

```
grps
  1  2
195 374
```

```
table(diagnosis)
```

```
diagnosis
  B    M
357 212
```

Cross-table to see how my clustering groups compare to the expert diagnosis vector.

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

```
      diagnosis
grps   B    M
1    18 177
2   339  35
```

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

The new model does a fairly good job at separating the diagnoses. However, there are still a significant amount of false positive results (18, are diagnosed with cancer but don't have it) and false negative results (35, are not diagnosed with cancer but do have it).

Positive => cancer M Negative => non-cancer B

True = cluster/grp 1 False = grp 2

True Pos = 177 False Pos = 18 True Neg = 339 False Neg = 35

So we captured 177/212 of the cancer positive patients. -> sensitivity

Q16. How well do the k-means and 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.

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

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

Our new method does about as well at splitting up results as the hierarchical clustering method, with slight increases in some areas and slight decreases in values of others. K-means is optional component

Sensitivity/Specificity

Sensitivity: $\text{True Pos} / (\text{true pos} + \text{false neg})$

Specificity: $\text{True Neg} / (\text{true neg} + \text{false pos})$

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

```
# for hierarchical clustering method
#1,2,4 -> B: 12+2+0 (FP), M: 165+5+2 (TP)
#3 -> B: 343 (TN), M:40 (FN)
TP1 = sum(165+2+5)
FP1 = sum(12+2+0)
TN1 = 343
FN1 = 40

sens2 <- TP1/(TP1+FN1)
sens2
```

```
[1] 0.8113208
```

```
spec2 <- TN1 / (TN1+FP1)
spec2
```

```
[1] 0.9607843
```

```
# for new created method
TP1 = 177
FP1 = 18
TN1 = 339
FN1 = 35

sens1 <- TP1/(TP1+FN1)
sens1
```

```
[1] 0.8349057
```

```
spec1 <- TN1 / (TN1+FP1)
spec1
```

```
[1] 0.9495798
```

```
#hierarchical clustering- 1, 2, and 4 are M, 3 is B
#newly created method groups
```

Based on these calculations, our new method displays greater sensitivity (83% v 81%) and the hierarchical clustering methods shows greater specificity (96% v 95%). Overall, the differences are pretty minimal and insignificant between the two techniques.

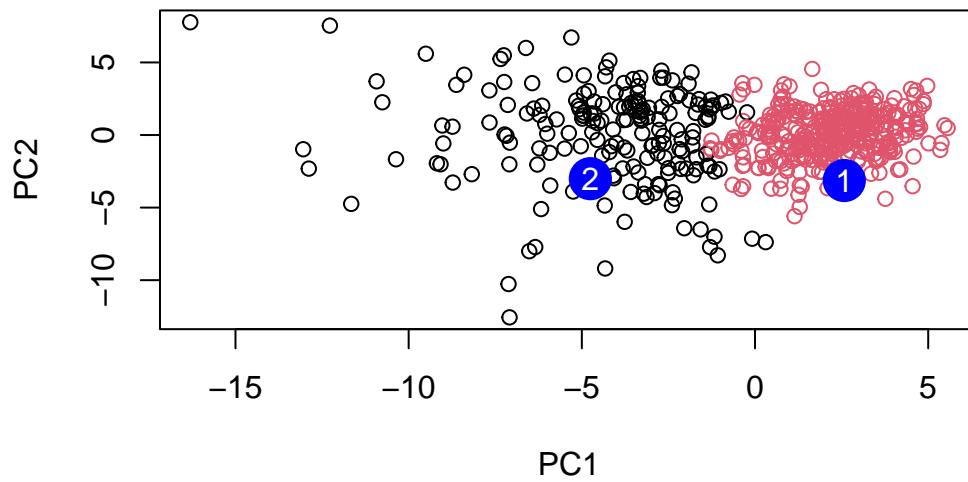
Predictions

We can use our PCA results (wisc.pr) to make predictions on new unseen data.

```
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=grps)
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?

I would prioritize patient number 2, as they fall much closer to the cluster of malignant data points.