

Class 8: Breast Cancer Mini Project

Abraham Rachlin

Background

Data import

Our data come from the U. of Wisconsin Medical Center

```
wisc.df <- read.csv("WisconsinCancer.csv", row.names = 1)
```

Q. How many patients/samples are in this dataset?

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

```
[1] 569
```

How many of the observations have a malignant diagnosis?

```
wisc.df$diagnosis
```

```
[1] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M"
[19] "M" "B" "B" "B" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M"
[37] "M" "B" "M" "M" "M" "M" "M" "M" "M" "M" "B" "M" "B" "B" "B" "B" "B" "M"
[55] "M" "B" "M" "M" "B" "B" "B" "B" "M" "B" "M" "M" "B" "B" "B" "B" "M" "B"
[73] "M" "M" "B" "M" "B" "M" "M" "B" "B" "B" "M" "M" "B" "M" "M" "M" "B" "B"
[91] "B" "M" "B" "B" "M" "M" "B" "B" "B" "M" "M" "B" "B" "B" "B" "M" "B" "B"
[109] "M" "B" "B" "B" "B" "B" "B" "B" "B" "M" "M" "M" "B" "M" "M" "B" "B" "B"
[127] "M" "M" "B" "M" "B" "M" "M" "B" "M" "M" "B" "B" "M" "B" "B" "M" "B" "B"
[145] "B" "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "B" "M"
[163] "M" "B" "M" "B" "B" "M" "M" "B" "B" "M" "M" "B" "B" "B" "B" "M" "B" "B"
[181] "M" "M" "M" "B" "M" "B" "M" "B" "B" "B" "M" "B" "B" "M" "M" "B" "M" "M"
[199] "M" "M" "B" "M" "M" "M" "B" "M" "B" "M" "B" "B" "M" "B" "M" "M" "M" "M"
```

```

[217] "B" "B" "M" "M" "B" "B" "B" "M" "B" "B" "B" "B" "B" "M" "M" "B" "B" "M"
[235] "B" "B" "M" "M" "B" "M" "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "M" "B"
[253] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "B" "B" "B" "B"
[271] "B" "B" "M" "B" "M" "B" "B" "M" "B" "B" "M" "B" "M" "M" "B" "B" "B" "B"
[289] "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "M" "B" "B" "B"
[307] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "M" "B" "M"
[325] "B" "B" "B" "B" "M" "M" "M" "B" "B" "B" "B" "M" "B" "M" "B" "M" "B" "B"
[343] "B" "M" "B" "B" "B" "B" "B" "B" "B" "M" "M" "M" "B" "B" "B" "B" "B" "B"
[361] "B" "B" "B" "B" "B" "M" "M" "B" "M" "M" "M" "B" "M" "M" "B" "B" "B" "B"
[379] "B" "M" "B" "B" "B" "B" "B" "M" "B" "B" "B" "M" "B" "B" "M" "M" "B" "B"
[397] "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "B" "B"
[415] "M" "B" "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B"
[433] "M" "M" "B" "M" "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "M" "B" "B" "M"
[451] "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "M" "M" "B" "B" "B" "B" "B" "B"
[469] "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "B"
[487] "B" "M" "B" "M" "B" "B" "M" "B" "B" "B" "B" "B" "M" "M" "B" "M" "B" "M"
[505] "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "M" "B" "M" "M" "B" "B" "B" "M"
[523] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "M" "M" "B" "B" "B"
[541] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
[559] "B" "B" "B" "B" "M" "M" "M" "M" "M" "M" "M" "B"

```

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

```

  B    M
357 212

```

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

```
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"
```

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

```
[1] 10
```

There is a diagnosis column that is the clinician consensus that I want to exclude from any further analysis. We will come back later and compare our results to this diagnosis.

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

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

Now we can remove it from the `wisc.df`

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

Clustering

```
kmeans(wisc.data, centers=2)
```

K-means clustering with 2 clusters of sizes 131, 438

Cluster means:

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
1	19.37992	21.69458	128.23130	1185.9298	0.1012946
2	12.55630	18.57037	81.12347	496.0619	0.0948845
	compactness_mean	concavity_mean	concave.points_mean	symmetry_mean	
1	0.14861298	0.17693947		0.10069878	0.1915397
2	0.09109982	0.06243776		0.03343254	0.1780580
	fractal_dimension_mean	radius_se	texture_se	perimeter_se	area_se

1	0.06060290	0.7428038	1.222538	5.250580	95.67817
2	0.06345402	0.3041909	1.215153	2.152881	23.78529
	smoothness_se	compactness_se	concavity_se	concave.points_se	symmetry_se
1	0.006598687	0.03217669	0.04241977	0.01567398	0.02030397
2	0.007173263	0.02347469	0.02874551	0.01063632	0.02061358
	fractal_dimension_se	radius_worst	texture_worst	perimeter_worst	area_worst
1	0.003953389	23.70947	28.91267	158.49618	1753.0229
2	0.003747503	14.04390	24.70954	91.93751	619.6479
	smoothness_worst	compactness_worst	concavity_worst	concave.points_worst	
1	0.1404247	0.3577577	0.4493061	0.19243107	
2	0.1299591	0.2233118	0.2192149	0.09132984	
	symmetry_worst	fractal_dimension_worst			
1	0.3118817	0.08616550			
2	0.2835537	0.08328194			

Clustering vector:

842302	842517	84300903	84348301	84358402	843786	844359	84458202
1	1	1	2	1	2	1	2
844981	84501001	845636	84610002	846226	846381	84667401	84799002
2	2	2	1	1	2	2	2
848406	84862001	849014	8510426	8510653	8510824	8511133	851509
2	1	1	2	2	2	2	1
852552	852631	852763	852781	852973	853201	853401	853612
1	1	2	1	1	1	1	2
85382601	854002	854039	854253	854268	854941	855133	855138
1	1	1	1	2	2	2	2
855167	855563	855625	856106	85638502	857010	85713702	85715
2	2	1	2	2	1	2	2
857155	857156	857343	857373	857374	857392	857438	85759902
2	2	2	2	2	1	2	2
857637	857793	857810	858477	858970	858981	858986	859196
1	2	2	2	2	2	2	2
85922302	859283	859464	859465	859471	859487	859575	859711
2	2	2	2	2	2	1	2
859717	859983	8610175	8610404	8610629	8610637	8610862	8610908
1	2	2	1	2	1	1	2
861103	8611161	8611555	8611792	8612080	8612399	86135501	86135502
2	2	1	1	2	1	2	1
861597	861598	861648	861799	861853	862009	862028	86208
2	2	2	2	2	2	2	1
86211	862261	862485	862548	862717	862722	862965	862980
2	2	2	2	2	2	2	2
862989	863030	863031	863270	86355	864018	864033	86408

2	2	2	2	1	2	2	2
86409	864292	864496	864685	864726	864729	864877	865128
2	2	2	2	2	2	1	1
865137	86517	865423	865432	865468	86561	866083	866203
2	1	1	2	2	2	2	1
866458	866674	866714	8670	86730502	867387	867739	868202
2	1	2	2	2	2	1	2
868223	868682	868826	868871	868999	869104	869218	869224
2	2	2	2	2	1	2	2
869254	869476	869691	86973701	86973702	869931	871001501	871001502
2	2	2	2	2	2	2	2
8710441	87106	8711002	8711003	8711202	8711216	871122	871149
2	2	2	2	1	2	2	2
8711561	8711803	871201	8712064	8712289	8712291	87127	8712729
2	1	1	2	1	2	2	1
8712766	8712853	87139402	87163	87164	871641	871642	872113
1	2	2	2	2	2	2	2
872608	87281702	873357	873586	873592	873593	873701	873843
2	2	2	2	1	1	1	2
873885	874158	874217	874373	874662	874839	874858	875093
2	2	1	2	2	2	2	2
875099	875263	87556202	875878	875938	877159	877486	877500
2	2	2	2	2	1	1	2
877501	877989	878796	87880	87930	879523	879804	879830
2	1	1	2	2	2	2	1
8810158	8810436	881046502	8810528	8810703	881094802	8810955	8810987
2	2	1	2	1	2	2	2
8811523	8811779	8811842	88119002	8812816	8812818	8812844	8812877
2	2	1	1	2	2	2	2
8813129	88143502	88147101	88147102	88147202	881861	881972	88199202
2	2	2	2	2	2	1	2
88203002	88206102	882488	88249602	88299702	883263	883270	88330202
2	1	2	2	1	1	2	1
88350402	883539	883852	88411702	884180	884437	884448	884626
2	2	2	2	1	2	2	2
88466802	884689	884948	88518501	885429	8860702	886226	886452
2	2	1	2	1	1	1	2
88649001	886776	887181	88725602	887549	888264	888570	889403
1	2	1	2	1	1	1	2
889719	88995002	8910251	8910499	8910506	8910720	8910721	8910748
1	1	2	2	2	2	2	2
8910988	8910996	8911163	8911164	8911230	8911670	8911800	8911834
1	2	1	2	2	1	2	2

8912049	8912055	89122	8912280	8912284	8912521	8912909	8913
1	2	1	2	2	2	2	2
8913049	89143601	89143602	8915	891670	891703	891716	891923
2	2	2	2	2	2	2	2
891936	892189	892214	892399	892438	892604	89263202	892657
2	2	2	2	1	2	1	2
89296	893061	89344	89346	893526	893548	893783	89382601
2	2	2	2	2	2	2	2
89382602	893988	894047	894089	894090	894326	894329	894335
2	2	2	2	2	1	2	2
894604	894618	894855	895100	89511501	89511502	89524	895299
2	1	2	1	2	2	2	2
8953902	895633	896839	896864	897132	897137	897374	89742801
2	2	2	2	2	2	2	1
897604	897630	897880	89812	89813	898143	89827	898431
2	1	2	1	2	2	2	1
89864002	898677	898678	89869	898690	899147	899187	899667
2	2	2	2	2	2	2	2
899987	9010018	901011	9010258	9010259	901028	9010333	901034301
1	2	2	2	2	2	2	2
901034302	901041	9010598	9010872	9010877	901088	9011494	9011495
2	2	2	2	2	1	1	2
9011971	9012000	9012315	9012568	9012795	901288	9013005	901303
1	1	2	2	1	1	2	2
901315	9013579	9013594	9013838	901549	901836	90250	90251
2	2	2	2	2	2	2	2
902727	90291	902975	902976	903011	90312	90317302	903483
2	2	2	2	2	1	2	2
903507	903516	903554	903811	90401601	90401602	904302	904357
1	1	2	2	2	2	2	2
90439701	904647	904689	9047	904969	904971	905189	905190
1	2	2	2	2	2	2	2
90524101	905501	905502	905520	905539	905557	905680	905686
1	2	2	2	2	2	2	2
905978	90602302	906024	906290	906539	906564	906616	906878
2	1	2	2	2	2	2	2
907145	907367	907409	90745	90769601	90769602	907914	907915
2	2	2	2	2	2	2	2
908194	908445	908469	908489	908916	909220	909231	909410
1	1	2	2	2	2	2	2
909411	909445	90944601	909777	9110127	9110720	9110732	9110944
2	1	2	2	1	2	1	2
911150	911157302	9111596	9111805	9111843	911201	911202	9112085

2	1	2	1	2	2	2	2
9112366	9112367	9112594	9112712	911296201	911296202	9113156	911320501
2	2	2	2	1	1	2	2
911320502	9113239	9113455	9113514	9113538	911366	9113778	9113816
2	2	2	2	1	2	2	2
911384	9113846	911391	911408	911654	911673	911685	911916
2	2	2	2	2	2	2	2
912193	91227	912519	912558	912600	913063	913102	913505
2	2	2	2	2	2	2	1
913512	913535	91376701	91376702	914062	914101	914102	914333
2	2	2	1	1	2	2	2
914366	914580	914769	91485	914862	91504	91505	915143
2	2	1	1	2	2	2	1
915186	915276	91544001	91544002	915452	915460	91550	915664
2	2	2	2	2	2	2	2
915691	915940	91594602	916221	916799	916838	917062	917080
2	2	2	2	1	1	2	2
917092	91762702	91789	917896	917897	91805	91813701	91813702
2	1	2	2	2	2	2	2
918192	918465	91858	91903901	91903902	91930402	919537	919555
2	2	2	2	2	1	2	1
91979701	919812	921092	921362	921385	921386	921644	922296
2	2	2	2	2	2	2	2
922297	922576	922577	922840	923169	923465	923748	923780
2	2	2	2	2	2	2	2
924084	924342	924632	924934	924964	925236	925277	925291
2	2	2	2	2	2	2	2
925292	925311	925622	926125	926424	926682	926954	927241
2	2	2	1	1	1	2	1
92751							
2							

Within cluster sum of squares by cluster:

[1] 49383423 28559677

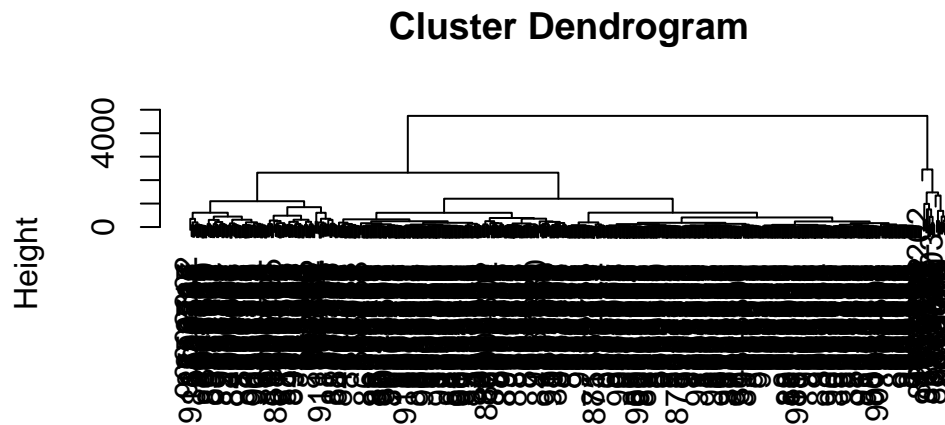
(between_SS / total_SS = 69.6 %)

Available components:

[1] "cluster"	"centers"	"totss"	"withinss"	"tot.withinss"
[6] "betweenss"	"size"	"iter"	"ifault"	

Let's try a hclust

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



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

We can extract clusters from this rather poor dendrogram/tree with the `cutree()`

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

How many individuals in each cluster

```
table(grps)
```

```
grps
 1  2
549 20
```

```
table(diagnosis)
```

```
diagnosis
 B  M
357 212
```


We can generate a cross-table that compares our cluster `grps` vector without `diagnosis` vector values

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

```
      grps  
diagnosis 1  2  
B 357    0  
M 192   20
```

Principal Component Analysis

The importance of data scaling

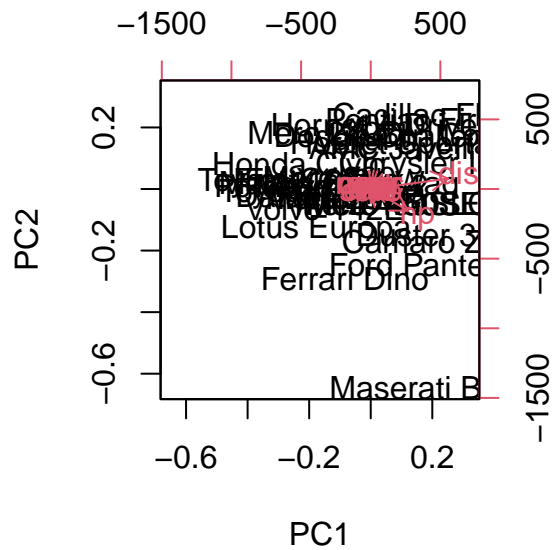
The main function for PCA in base R is `prcomp()` it has a default input parameter of `scale=F`

```
#prcomp()  
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

We could do a PCA of this data as is and it could be misleading...

```
pc <- prcomp(mtcars)  
biplot(pc)
```



Let's look at the mean values of each column and their standard deviation.

```
colMeans(mtcars)
```

mpg	cyl	disp	hp	drat	wt	qsec
20.090625	6.187500	230.721875	146.687500	3.596563	3.217250	17.848750
vs	am	gear	carb			
0.437500	0.406250	3.687500	2.812500			

```
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	

We can “scale” this data before PCA to get a much better representation and analysis of all the columns.

```
mtscale <- scale(mtcars)
```

```
round(colMeans(mtscale))
```

```
mpg  cyl  disp   hp drat    wt  qsec    vs  am  gear  carb
0     0     0     0   0     0     0     0   0   0     0
```

```
apply(mtscale, 2, sd)
```

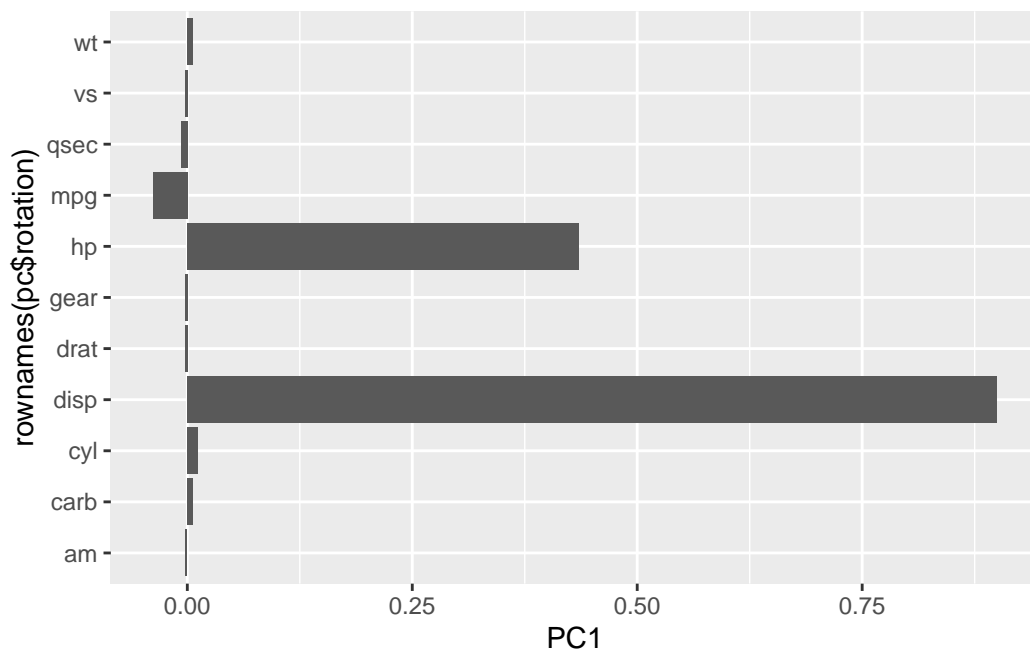
```
mpg  cyl  disp   hp drat    wt  qsec    vs  am  gear  carb
1     1     1     1   1     1     1     1   1   1     1
```

```
pr.scale <- prcomp(mtscale)
```

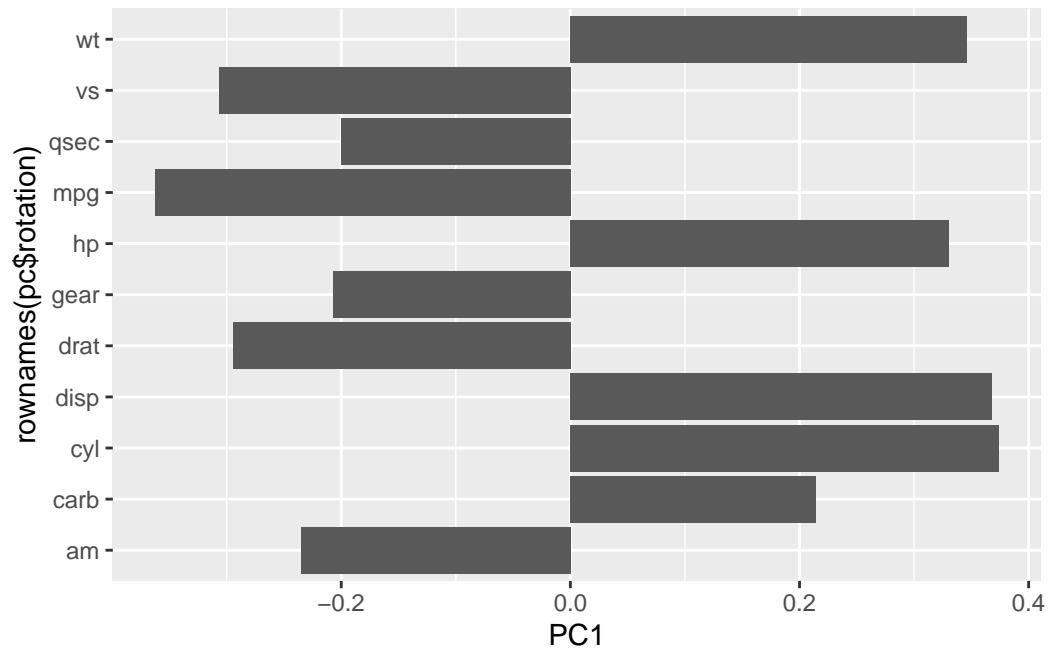
We can look at the two main results figures from PCA- the “PC plot” (a.k.a. score plot, ordination plot, or PC1 vs PC2 plot). The “loadings plot” how the original variables contribute to the new PCs.

A loadings plot of the unscaled PCA results

```
library(ggplot2)
ggplot(pc$rotation) +
  aes(PC1, rownames(pc$rotation)) +
  geom_col()
```

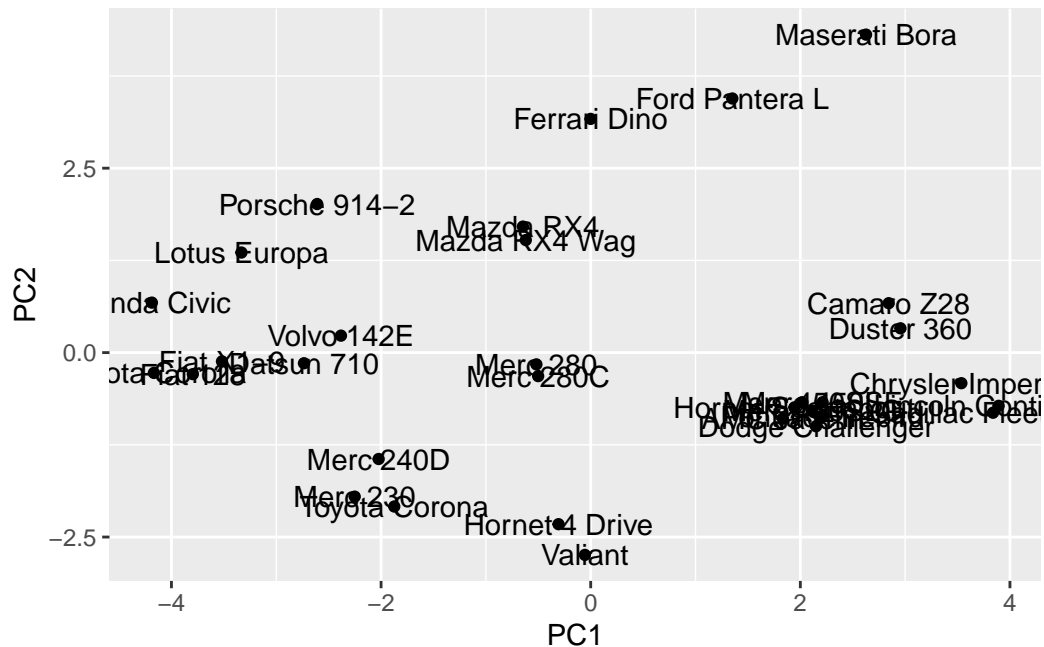


```
ggplot(pr.scale$rotation) +
  aes(PC1, rownames(pc$rotation)) +
  geom_col()
```



PC plot of scaled PCA results

```
ggplot(pr.scale$x) +
  aes(PC1, PC2, label=rownames(pr.scale$x)) +
  geom_point() +
  geom_text()
```



Key point: In general, we will set `scale=TRUE` when we do PCA. This is not the default but probably should be...

We can check the SD and mean of the different columns in `wisc.data` to see if we need to scale - hint, we do!

PCA of wisc data

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

To see how well PCA is doing here in terms capturing the variance (or spread) in the data, we can use the `'summary()'` function.

```
summary(wisc.pr)
```

Importance of components:

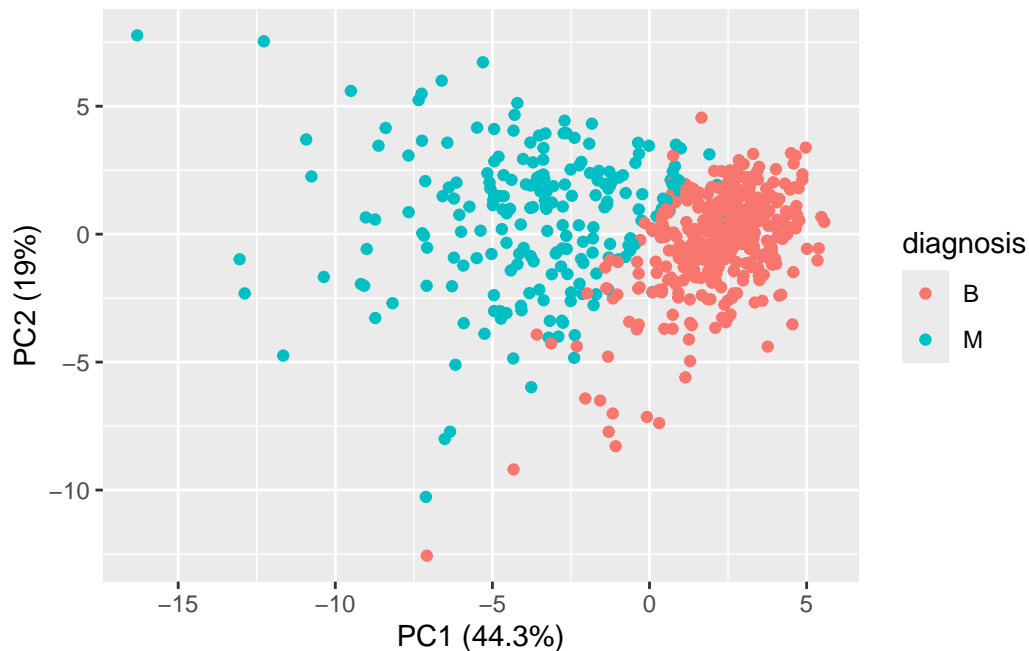
	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	3.6444	2.3857	1.67867	1.40735	1.28403	1.09880	0.82172
Proportion of Variance	0.4427	0.1897	0.09393	0.06602	0.05496	0.04025	0.02251
Cumulative Proportion	0.4427	0.6324	0.72636	0.79239	0.84734	0.88759	0.91010

	PC8	PC9	PC10	PC11	PC12	PC13	PC14
Standard deviation	0.78113	0.65752	0.59254	0.53846	0.48960	0.44689	0.41020
Proportion of Variance	0.02251	0.01944	0.01754	0.01602	0.01454	0.01334	0.01222
Cumulative Proportion	0.93261	0.95205	0.96959	0.98561	0.99994	1.00000	1.00000

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					

Let's make the main PC1 vs PC2

```
ggplot(wisc.pr$x) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point() +
  xlab("PC1 (44.3%)") +
  ylab("PC2 (19%)")
```



Q4. From your results, what proportion of the original variance is captured by the first principal components (PC1)?

0.4427 or 44.27%.

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

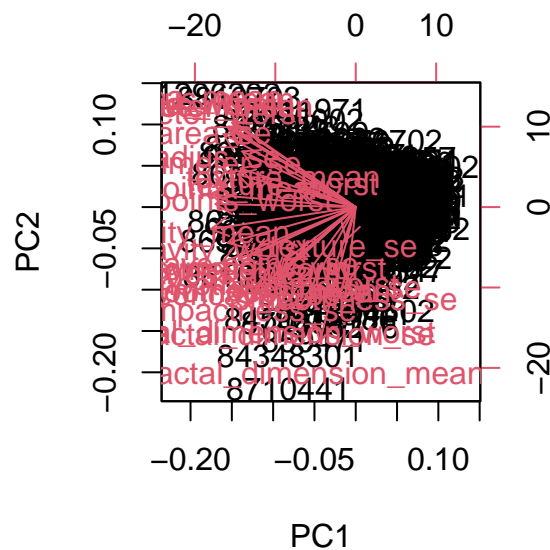
3 PCs are required to get to 70%.

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

You need about 7 principal components to reach 90% of the original variance.

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

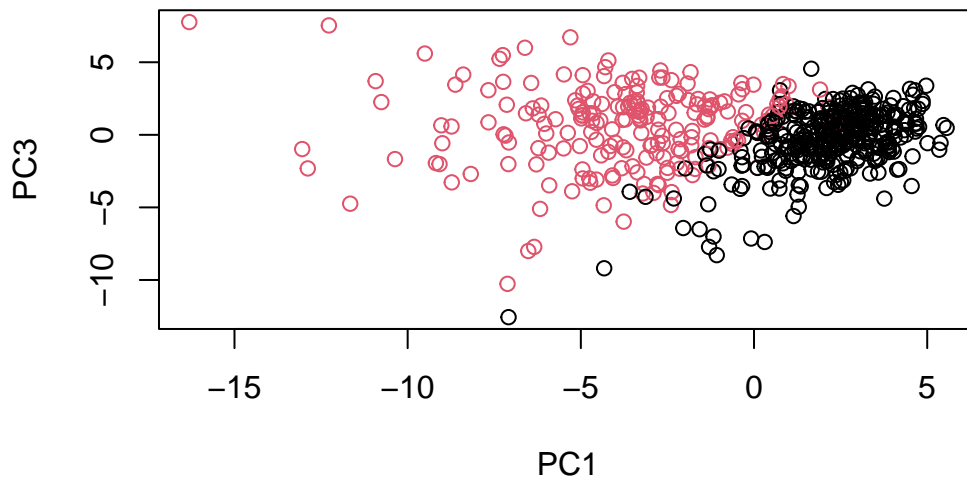
```
biplot(wisc.pr)
```



What stands out to me about this plot is the fact that it is a massive mess. This plot seems very difficult to understand, as everything is very unorganized.

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

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



What I can notice about this graph is that it is much cleaner than the previous. This plot shows two clumps of data, separating the two groups by diagnosis.

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
```

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

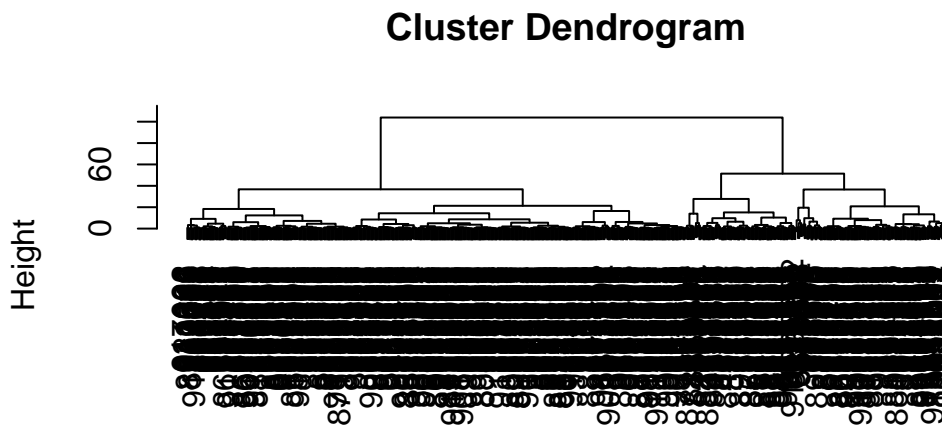
The minimum number of principal components to explain 80% of variance is five principal components.

5. Combining methods

We can take our PCA results and use them as a basis set for other analysis such as clustering.

Clustering on PCA Results

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



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

We can “cut” this tree to yield our cluster (groups):

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

```
pc.grps  
 1    2  
195 374
```

How do my cluster grps compare to the expert diagnosis

```
table(diagnosis, pc.grps)
```

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

```
table(diagnosis)
```

```
diagnosis
  B   M
357 212
```

q15. How well does the newly created model with four clusters separate out the two diagnoses?

The newly created model with four clusters separates the diagnosis well. It helps visualize how the benign groups and metastacized groups differ across the different groups.

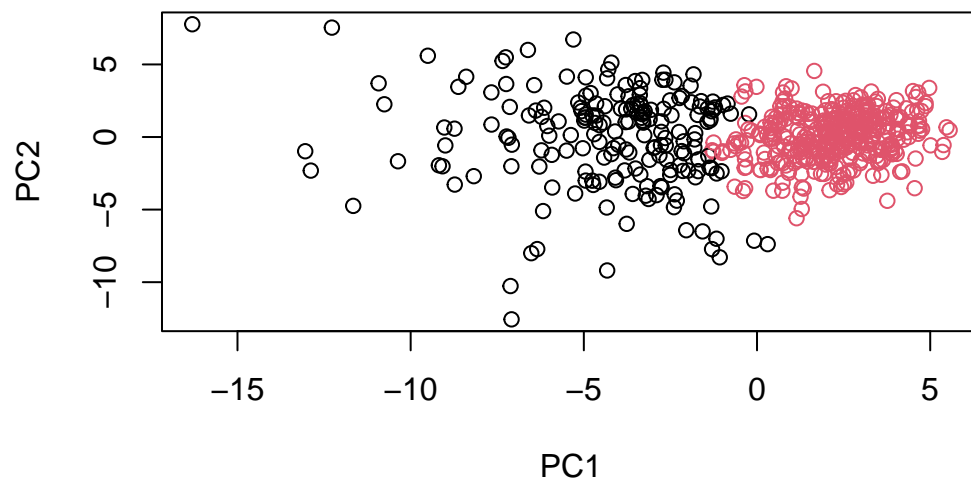
```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
 1   2
195 374
```

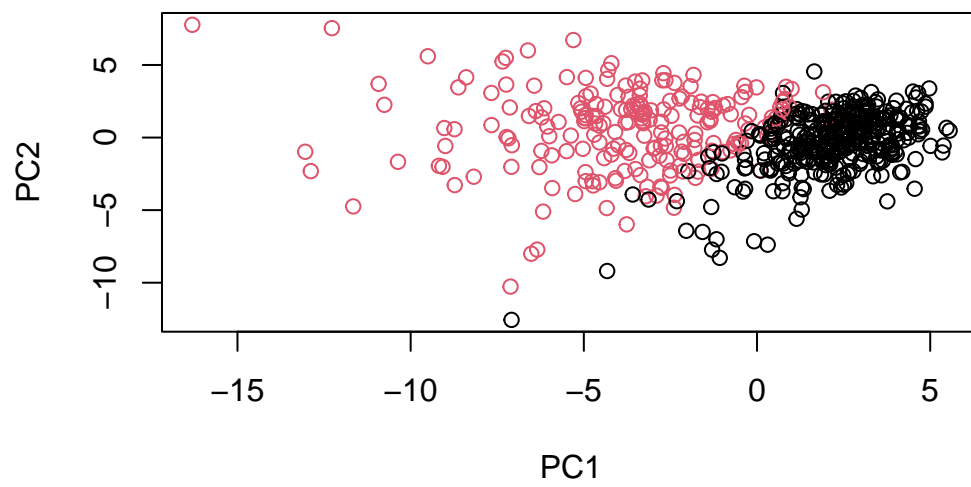
```
table(grps, diagnosis)
```

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

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



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



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

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

```

      diagnosis
wisc.pr.hclust.clusters  B  M
1    18 177
2   339  35

```

Q16. 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.

I think they do a good job at separating the models well, as you could see the two groups separated well. However, it could be better, as it could be in much more detail.

They did really poorly, we do much better after PCA- the new PCA variables give us much better separation of M and B.

7. Prediction

We can use our model PCA model for the analysis of new “unseen” data. In this case fro, U. Mich.

```

#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

```

```

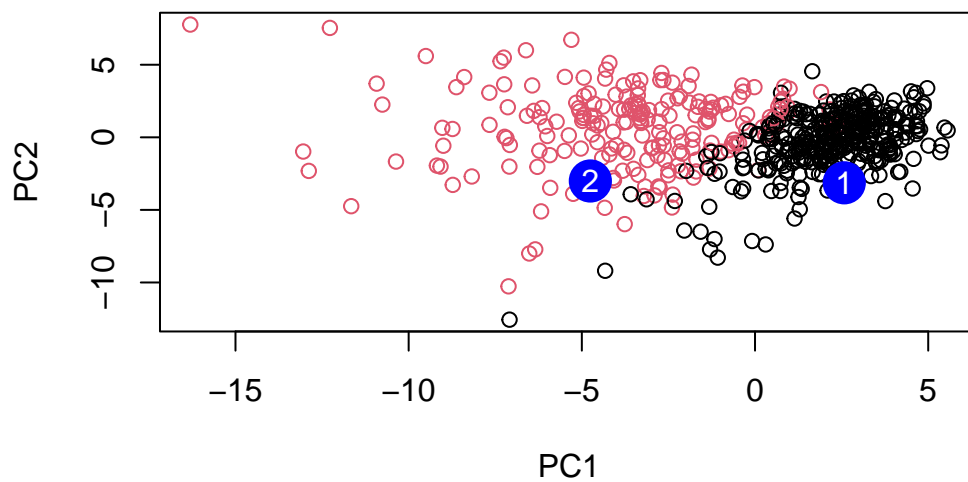
[1,] 0.1228233 0.09358453 0.08347651 0.1223396 0.02124121 0.078884581
[2,] -0.1224776 0.01732146 0.06316631 -0.2338618 -0.20755948 -0.009833238
      PC27      PC28      PC29      PC30
[1,] 0.220199544 -0.02946023 -0.015620933 0.005269029
[2,] -0.001134152 0.09638361 0.002795349 -0.019015820

```

```

plot(wisc.pr$x[,1:2], col=diagnosis)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")

```



For follow-up, patients in group 2 should be asked for a follow-up.