

Class08 Lab

Nickolas Beam

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

We will here remove the first column (the expert diagnosis) as we will not use this in our analysis.

```
#We can use -1 here to remove the first column.
wisc.data <- wisc.df[,-1]
diagnosis <- as.factor(wisc.df[,1])
```

Q1 How many observations are in this dataset?

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

```
[1] 569
```

Q2 How many of the observations have a malignant diagnosis?

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

```
  B    M
357 212
```

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

```
colnames(wisc.data)
```

```

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

```

Use `grep()` to search for

```

matches <- grep("_mean", colnames(wisc.df))
length(matches)

```

```
[1] 10
```

Principal component analysis

```
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

We need to scale with 'scale=TRUE' argument with prcomp()

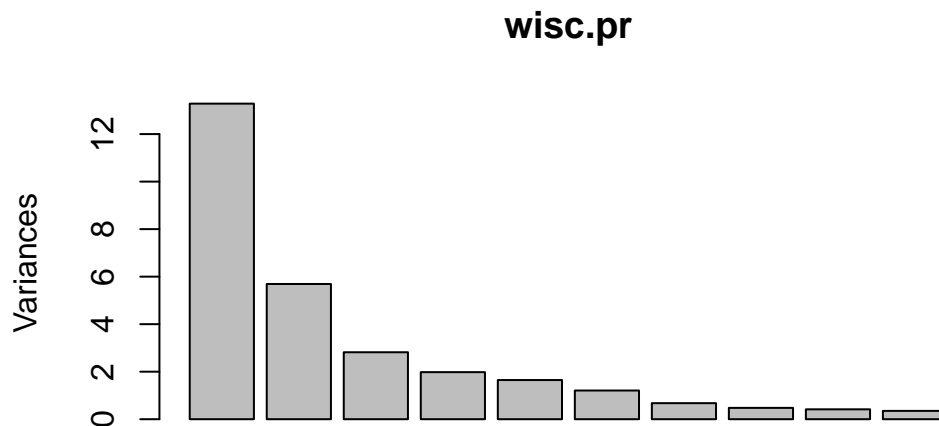
```
wisc.pr <- prcomp(wisc.data, scale = TRUE)
summary(wisc.pr)
```

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

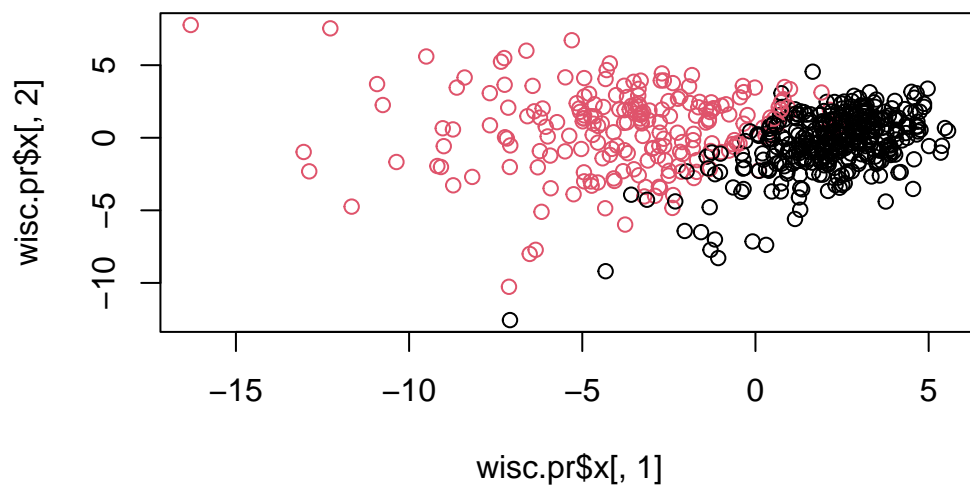
Proportion of Variance	0.4427	0.1897	0.09393	0.06602	0.05496	0.04025	0.02251
Cumulative Proportion	0.4427	0.6324	0.72636	0.79239	0.84734	0.88759	0.91010
	PC8	PC9	PC10	PC11	PC12	PC13	PC14
Standard deviation	0.69037	0.6457	0.59219	0.5421	0.51104	0.49128	0.39624
Proportion of Variance	0.01589	0.0139	0.01169	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					

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



Let's make a PC plot (a.k.a "score plot" or "PC1 vs PC2" etc. plot)

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



```
y <- summary(wisc.pr)
attributes(y)
```

```
$names
[1] "sdev"      "rotation"  "center"    "scale"     "x"
[6] "importance"
```

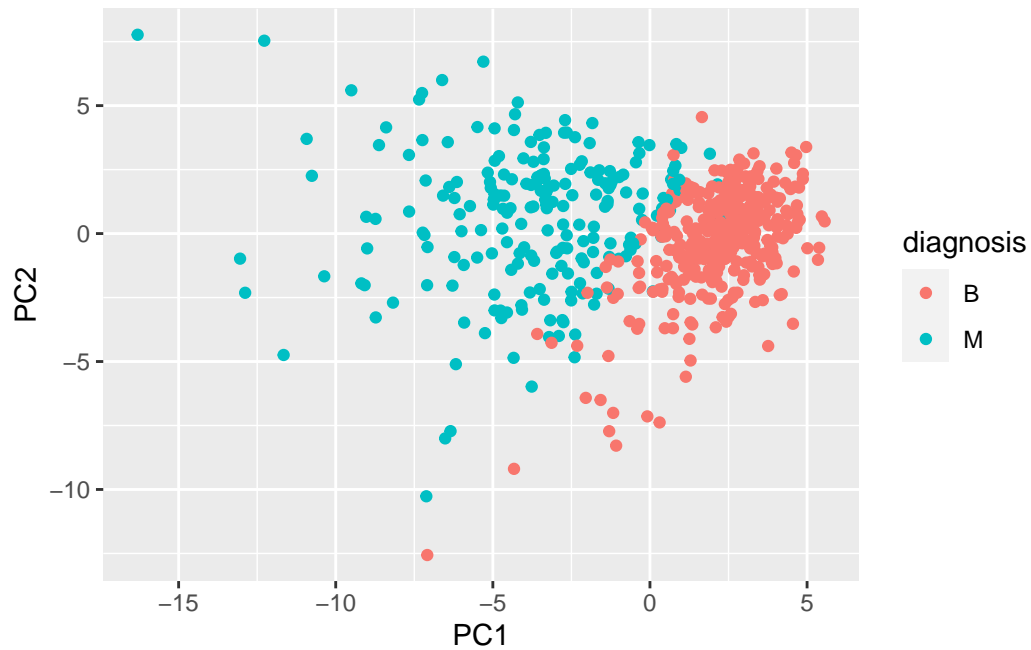
```
$class
[1] "summary.prcomp"
```

How many PCs do I need to capture 80% of the original variance in my dataset?

```
sum(y$importance[3,] <= 0.8)
```

```
[1] 4
```

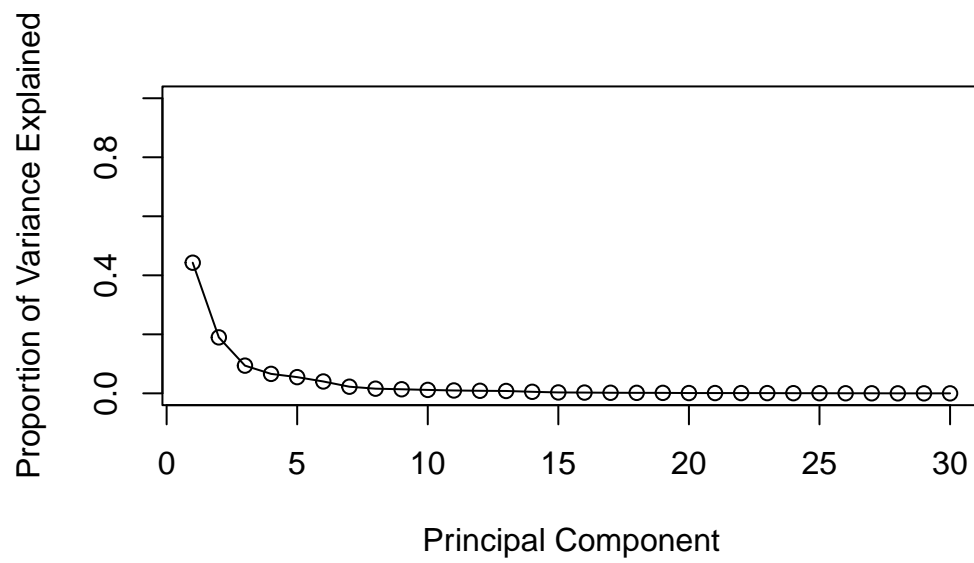
```
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis
library(ggplot2)
ggplot(df) + aes(PC1, PC2, col=diagnosis) + geom_point()
```



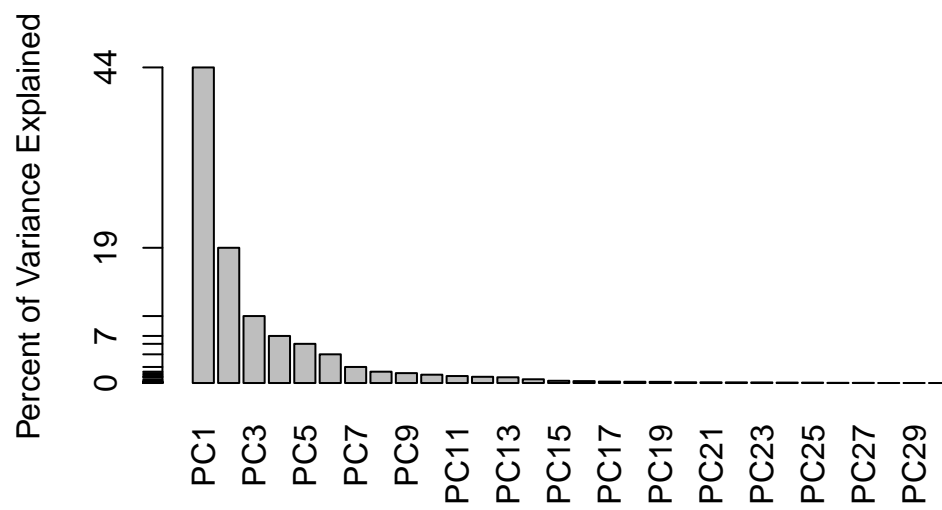
```
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

```
pve <- pr.var/sum(pr.var)
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0,1), type = "o")
```



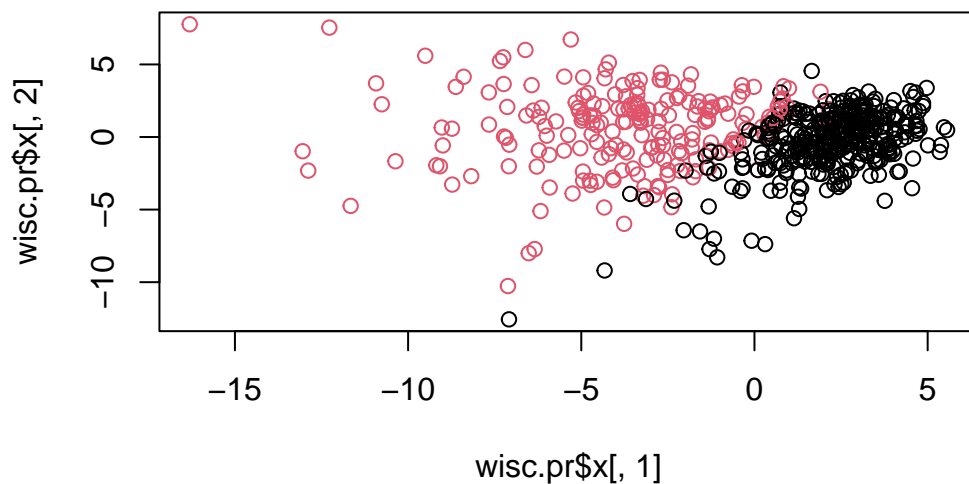
```
barplot(pve, ylab = "Percent of Variance Explained",  
        names.arg=paste0("PC", 1:length(pve)), las=2, axes = FALSE)  
axis(2, at=pve, labels=round(pve,2)*100)
```



Combine PCA with clustering

I want to cluster in “PC space”.

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

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

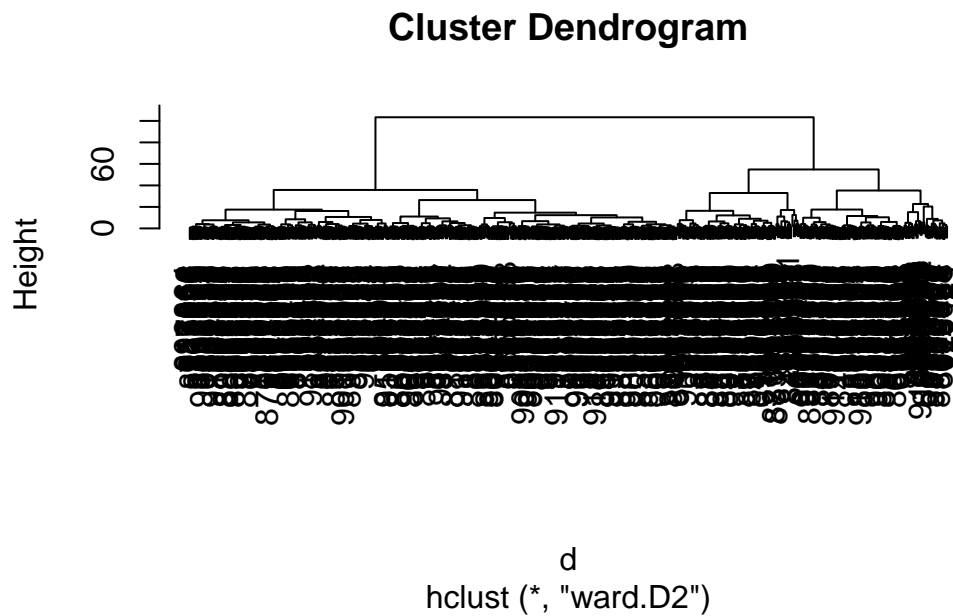
Importance of components:

	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

The `hclust()` function wants a distance matrix as input...

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



Find my cluster membership vector with the `cutree()` function.

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

```
grps
 1  2
203 366
```

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

	grps	
diagnosis	1	2
B	24	333
M	179	33