Lab Mini Project: Unsupervised Learning Analysis of Breast Cancer Cells

Nhi To

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

```r
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 of this dataset?

```r
apply(mtcars, 2, mean)
```

```
       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
```

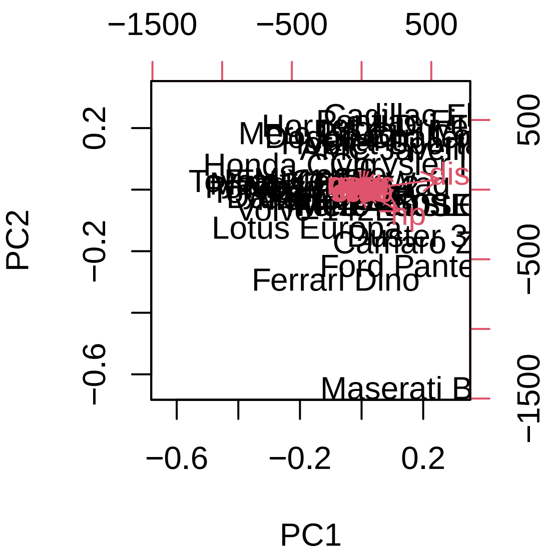
```r
apply(mtcars, 2, sd)
```

```
        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
```

It is clear that displacement and horsepower have the highest mean values and highest standard deviation here. They will likely dominate any analysis I do on this dataset. Let's see.

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

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



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

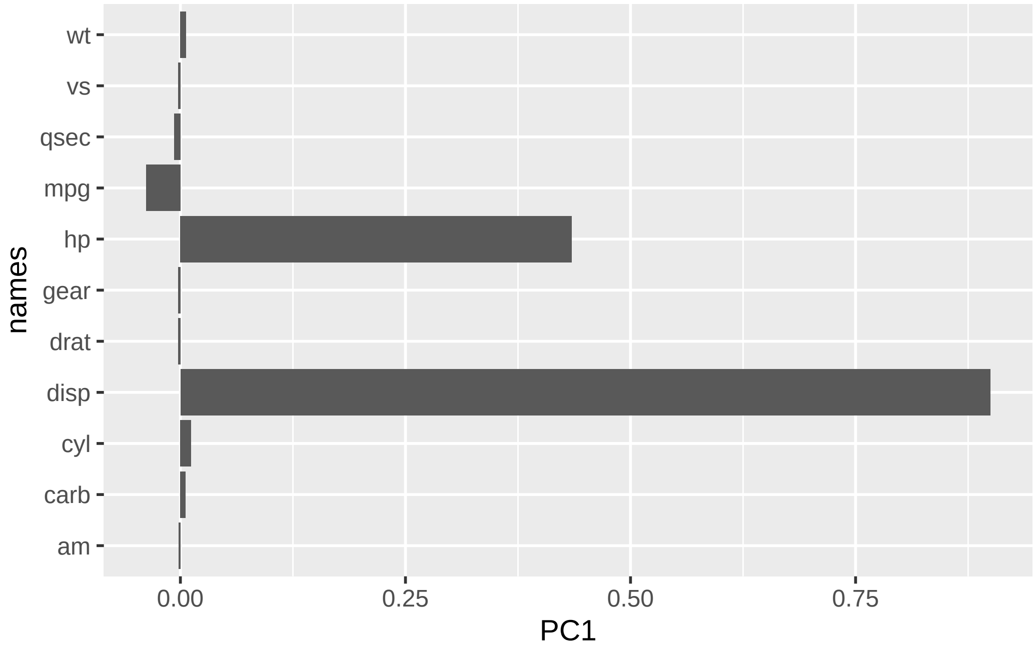
```
          mpg           cyl          disp            hp          drat            wt 
-0.038118199   0.012035150   0.899568146   0.434784387 -0.002660077   0.006239405 
         qsec            vs            am          gear          carb 
-0.006671270 -0.002729474 -0.001962644 -0.002604768   0.005766010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

Plot the loadings

```r
library(ggplot2)

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

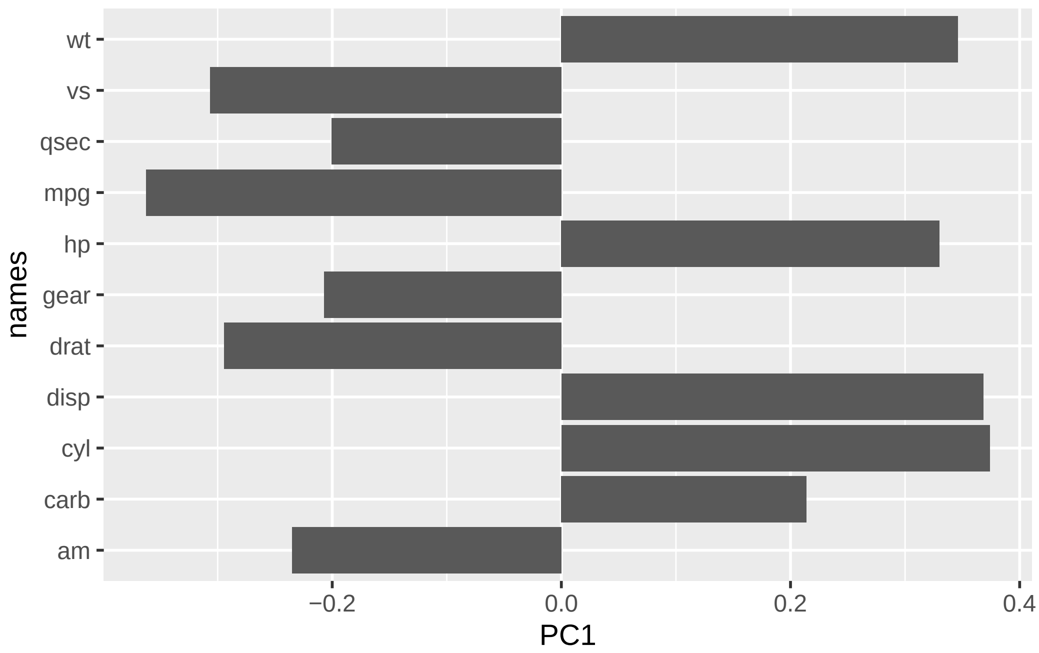
```
ggplot(r1)+
  aes(PC1, names) +
  geom_col()
```



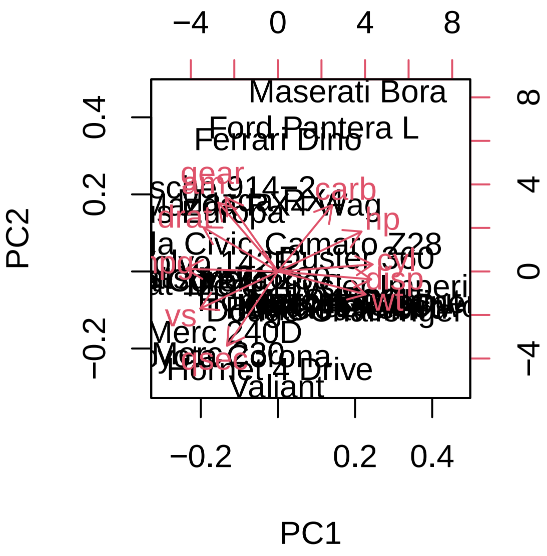
```
library(ggplot2)

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

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



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



Take-Home: Generally we always want to set ‘scale=TRUE’ when we do this type of analysis to avoid our analysis being dominated by individual variables with the largest variance just due to their unit of measurement.

#FNA breast cancer data

Load the data into R.

```r
wisc.df <-read.csv("WisconsinCancer.csv")
head(wisc.df)
```

```
        id diagnosis radius_mean texture_mean perimeter_mean area_mean
1   842302         M       17.99        10.38         122.80    1001.0
2   842517         M       20.57        17.77         132.90    1326.0
3 84300903         M       19.69        21.25         130.00    1203.0
4 84348301         M       11.42        20.38          77.58     386.1
5 84358402         M       20.29        14.34         135.10    1297.0
6   843786         M       12.45        15.70          82.57     477.1
  smoothness_mean compactness_mean concavity_mean concave.points_mean
1         0.11840          0.27760         0.3001             0.14710
2         0.08474          0.07864         0.0869             0.07017
3         0.10960          0.15990         0.1974             0.12790
4         0.14250          0.28390         0.2414             0.10520
5         0.10030          0.13280         0.1980             0.10430
6         0.12780          0.17000         0.1578             0.08089
  symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
1        0.2419                0.07871    1.0950     0.9053        8.589
2        0.1812                0.05667    0.5435     0.7339        3.398
3        0.2069                0.05999    0.7456     0.7869        4.585
4        0.2597                0.09744    0.4956     1.1560        3.445
5        0.1809                0.05883    0.7572     0.7813        5.438
6        0.2087                0.07613    0.3345     0.8902        2.217
  area_se smoothness_se compactness_se concavity_se concave.points_se
1  153.40      0.006399        0.04904      0.05373           0.01587
2   74.08      0.005225        0.01308      0.01860           0.01340
3   94.03      0.006150        0.04006      0.03832           0.02058
4   27.23      0.009110        0.07458      0.05661           0.01867
5   94.44      0.011490        0.02461      0.05688           0.01885
6   27.19      0.007510        0.03345      0.03672           0.01137
  symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
1     0.03003             0.006193        25.38         17.33          184.60
2     0.01389             0.003532        24.99         23.41          158.80
3     0.02250             0.004571        23.57         25.53          152.50
4     0.05963             0.009208        14.91         26.50           98.87
```

```
5     0.01756             0.005115        22.54          16.67          152.20
6     0.02165             0.005082        15.47          23.75          103.40
  area_worst smoothness_worst compactness_worst concavity_worst
1     2019.0           0.1622            0.6656          0.7119
2     1956.0           0.1238            0.1866          0.2416
3     1709.0           0.1444            0.4245          0.4504
4      567.7           0.2098            0.8663          0.6869
5     1575.0           0.1374            0.2050          0.4000
6      741.6           0.1791            0.5249          0.5355
  concave.points_worst symmetry_worst fractal_dimension_worst
1               0.2654         0.4601                 0.11890
2               0.1860         0.2750                 0.08902
3               0.2430         0.3613                 0.08758
4               0.2575         0.6638                 0.17300
5               0.1625         0.2364                 0.07678
6               0.1741         0.3985                 0.12440
```

> Q1. How many observations are in this dataset?
>
> **ANSWER**: There are 569 observations in this dataset.

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

```
[1] 569
```

> Q2. How many of the observations have a malignant diagnosis?
>
> **ANSWER**:There are 212 observations that have a malignant diagnosis.

```r
sum(wisc.df$diagnosis == "M")
```

```
[1] 212
```

The 'table()' function is super useful here

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

```

  B   M 
357 212 
```

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

ANSWER: 10 variables/features in the data were suffixed with "_mean".

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

```
[1] 32
```

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

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

A useful function for this is 'grep()'

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

```
[1] 10
```

Before we go any further, we need to exclude the diagnosis column from any future analysis - this tells us whether a sample to cancer or non-cancer

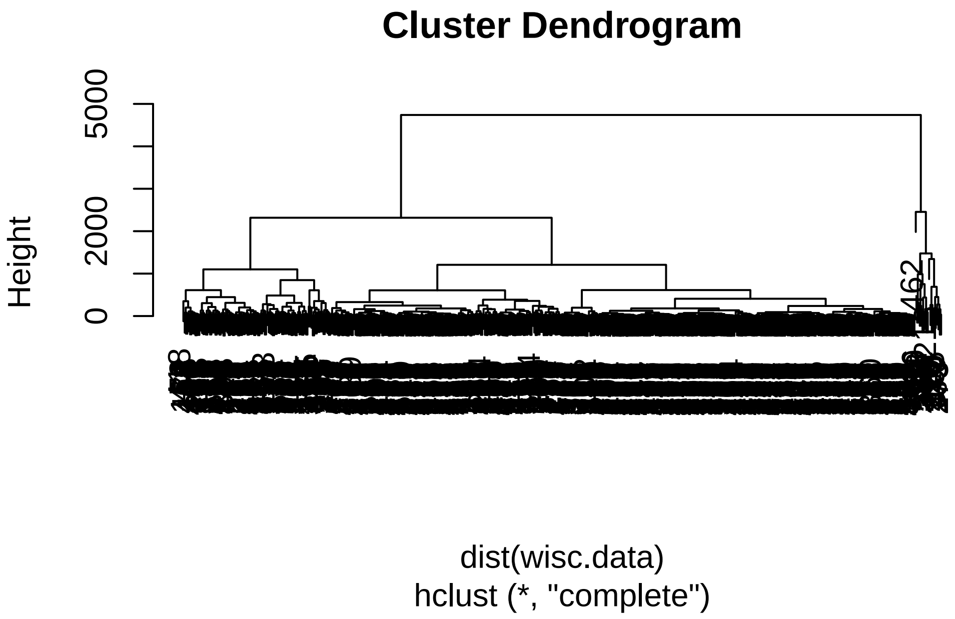
```r
diagnosis <- as.factor(wisc.df$diagnosis)
head(diagnosis)
```

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

```
wisc.data<- wisc.df[,-1:-2]
```

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

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



Principal Component Analysis (PCA)

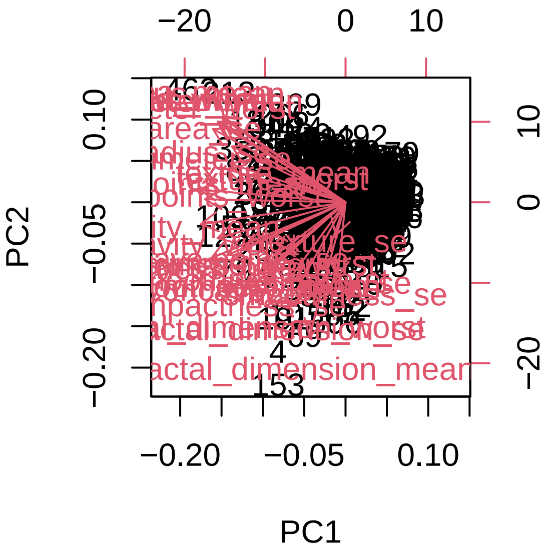
```
wisc.pr <- prcomp(wisc.data, scale=T)

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
```

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



This biplot sucks! We need to build our own PCA score plot of PC1 vs PC2

```r
head(wisc.pr$x)
```

```
              PC1         PC2         PC3        PC4         PC5          PC6
```

```
[1,] -9.184755  -1.946870 -1.1221788 3.6305364  1.1940595  1.41018364
[2,] -2.385703   3.764859 -0.5288274 1.1172808 -0.6212284  0.02863116
[3,] -5.728855   1.074229 -0.5512625 0.9112808  0.1769302  0.54097615
[4,] -7.116691 -10.266556 -3.2299475 0.1524129  2.9582754  3.05073750
[5,] -3.931842   1.946359  1.3885450 2.9380542 -0.5462667 -1.22541641
[6,] -2.378155  -3.946456 -2.9322967 0.9402096  1.0551135 -0.45064213
             PC7         PC8         PC9       PC10       PC11       PC12
[1,]  2.15747152  0.39805698 -0.15698023 -0.8766305 -0.2627243 -0.8582593
[2,]  0.01334635 -0.24077660 -0.71127897  1.1060218 -0.8124048  0.1577838
[3,] -0.66757908 -0.09728813  0.02404449  0.4538760  0.6050715  0.1242777
[4,]  1.42865363 -1.05863376 -1.40420412 -1.1159933  1.1505012  1.0104267
[5,] -0.93538950 -0.63581661 -0.26357355  0.3773724 -0.6507870 -0.1104183
[6,]  0.49001396  0.16529843 -0.13335576 -0.5299649 -0.1096698  0.0813699
            PC13         PC14         PC15        PC16        PC17        PC18
[1,]  0.10329677 -0.690196797  0.601264078  0.74446075 -0.26523740 -0.54907956
[2,] -0.94269981 -0.652900844 -0.008966977 -0.64823831 -0.01719707  0.31801756
[3,] -0.41026561  0.016665095 -0.482994760  0.32482472  0.19075064 -0.08789759
[4,] -0.93245070 -0.486988399  0.168699395  0.05132509  0.48220960 -0.03584323
[5,]  0.38760691 -0.538706543 -0.310046684 -0.15247165  0.13302526 -0.01869779
[6,] -0.02625135  0.003133944 -0.178447576 -0.01270566  0.19671335 -0.29727706
           PC19        PC20         PC21        PC22        PC23         PC24
[1,]  0.1336499  0.34526111  0.096430045 -0.06878939  0.08444429  0.175102213
[2,] -0.2473470 -0.11403274 -0.077259494  0.09449530 -0.21752666 -0.011280193
[3,] -0.3922812 -0.20435242  0.310793246  0.06025601 -0.07422581 -0.102671419
[4,] -0.0267241 -0.46432511  0.433811661  0.20308706 -0.12399554 -0.153294780
[5,]  0.4610302  0.06543782 -0.116442469  0.01763433  0.13933105  0.005327110
[6,] -0.1297265 -0.07117453 -0.002400178  0.10108043  0.03344819 -0.002837749
             PC25         PC26        PC27          PC28         PC29
[1,]  0.150887294 -0.201326305 -0.25236294 -0.0338846387  0.045607590
[2,]  0.170360355 -0.041092627  0.18111081  0.0325955021 -0.005682424
[3,] -0.171007656  0.004731249  0.04952586  0.0469844833  0.003143131
[4,] -0.077427574 -0.274982822  0.18330078  0.0424469831 -0.069233868
[5,] -0.003059371  0.039219780  0.03213957 -0.0347556386  0.005033481
[6,] -0.122282765 -0.030272333 -0.08438081  0.0007296587 -0.019703996
              PC30
[1,]  0.0471277407
[2,]  0.0018662342
[3,] -0.0007498749
[4,]  0.0199198881
[5,] -0.0211951203
[6,] -0.0034564331
```

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

ANSWER: The proportion of the original variance is 0.4427 that is captured by the first principal components (PC1).

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

ANSWER: PC1, PC2, and PC3 are the three principal compeonts (PCs) are required to describe at least 70% of the original variance in the data

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

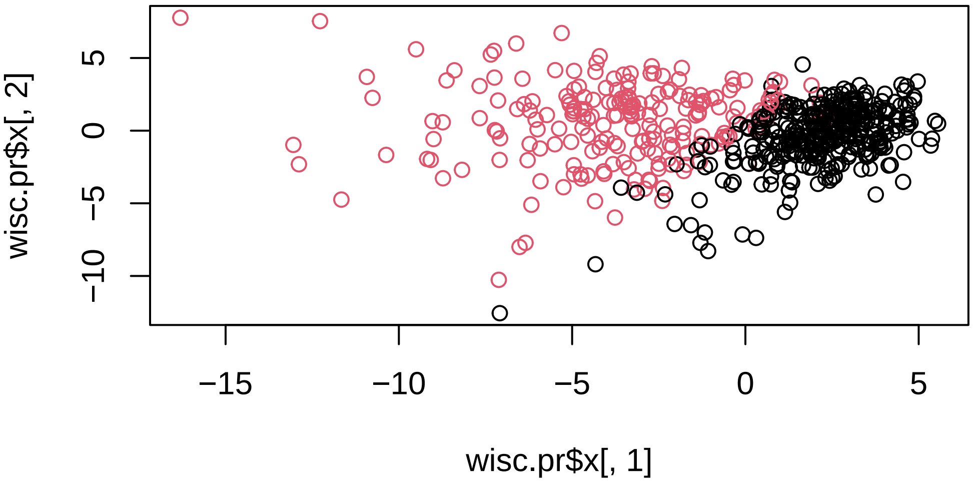
ANSWER: PC1, PC2, PC3, PC4, PC5, PC6, and PC7 are the seven principal components (PCs) that are required to describe at least 90% of the original variance in the data?

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

ANSWER: There are two red and black distinguished grouped data. The red grouped data represent the malignant (cancer cells), while the black grouped data represents the non-cancer cells. This plot is very difficult to understand because the data is so grouped, and the name values are heavily obscuring any analysis to be done on this graph.

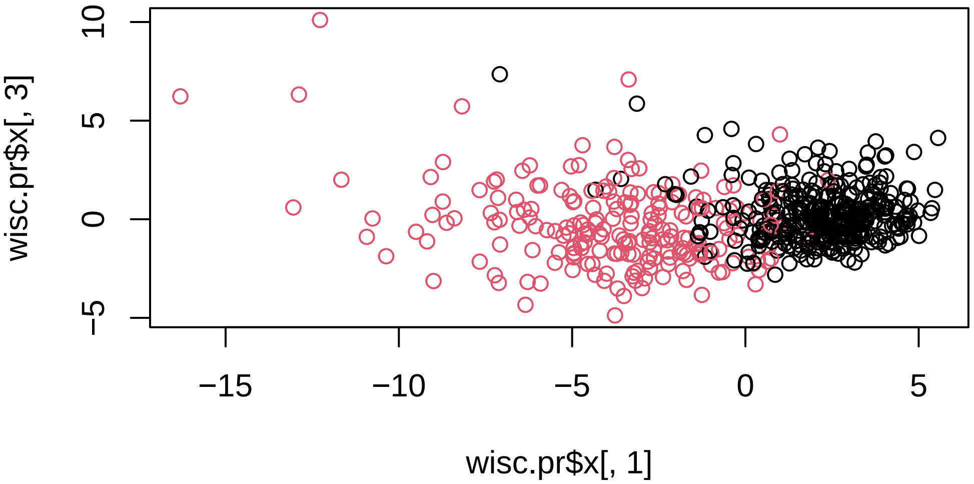
PLot of PC1 vs PC2 the first two columns

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



PLot of PC1 vs PC3 the first two columns

```
plot(wisc.pr$x[,1], wisc.pr$x[,3], col=diagnosis)
```



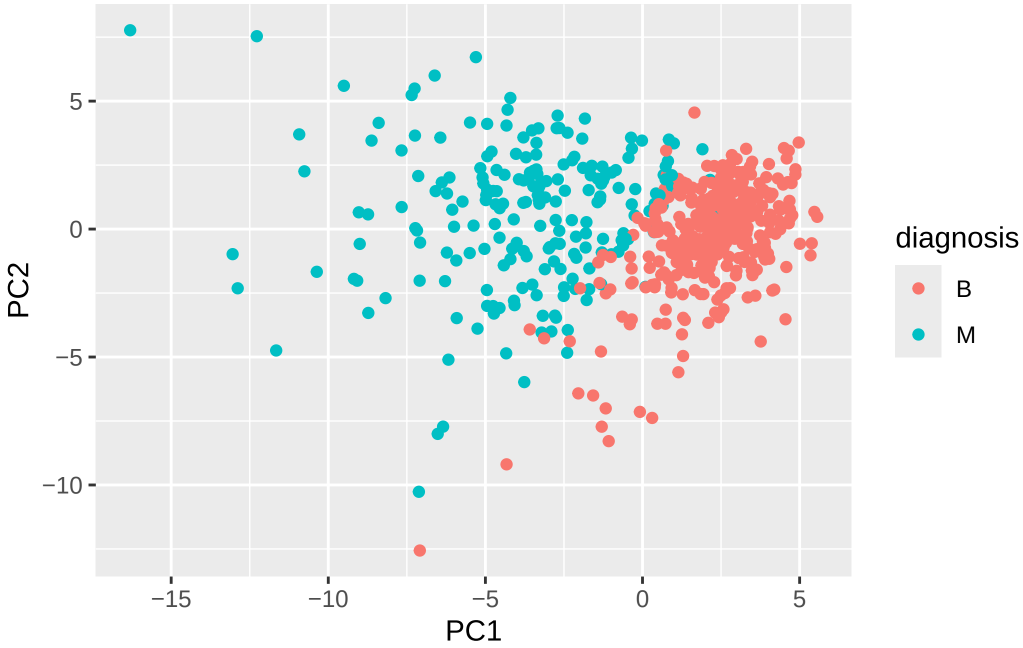
Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

ANSWER: I notice that these values are now more clustered and overlapping. In PC1 vs PC2, there were two clear distinct groups. There are still two distinct groups in PC1 vs PC3, however more of the data overlap now, indicating that there are more similarities between principal components 1 and 3.

Make a ggplot version of this score plot

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

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

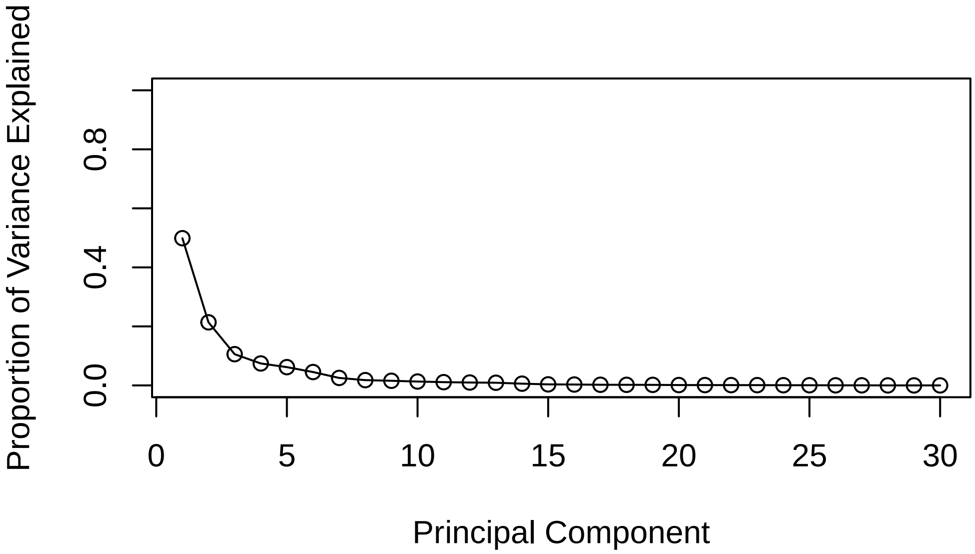


```r
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

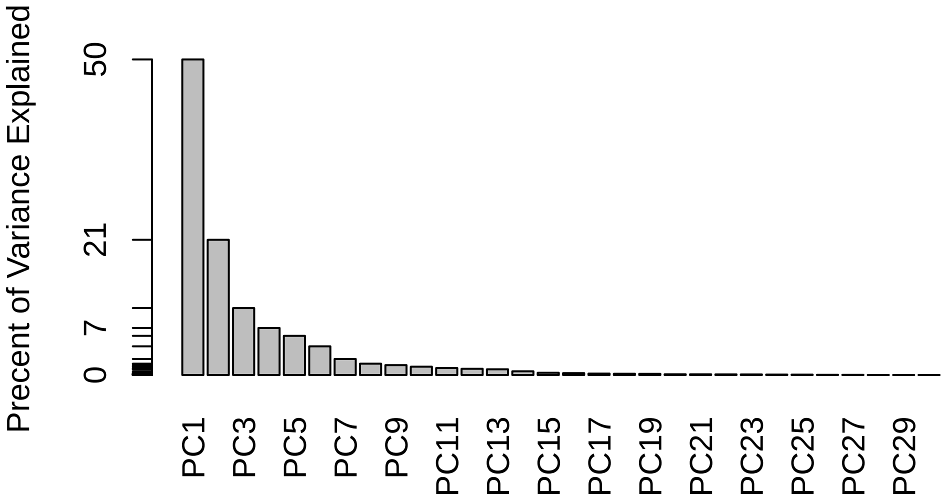
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

```
# Variance explained by each principal component: pve
pve <- pr.var / sum(head(pr.var))

# Plot variance explained for each principal component
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "o")
```



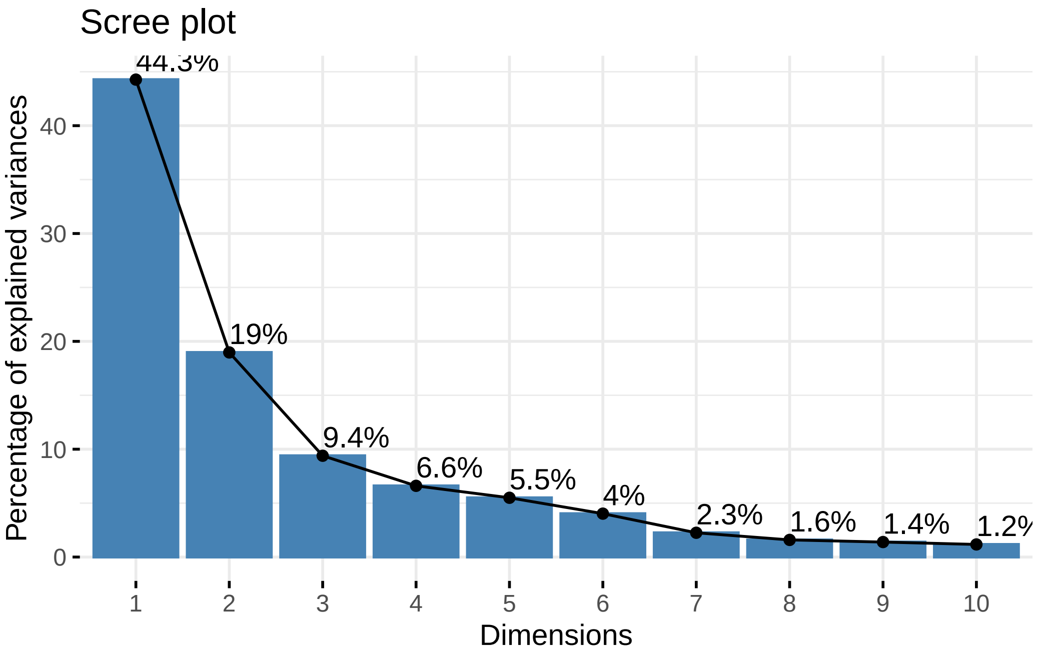
```
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```



```
## ggplot based graph
#install.packages("factoextra")
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(wisc.pr, addlabels = TRUE)
```



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?

ANSWER: The component of the loading vector wisc.pr$rotation[,1] for the feature concave.points_mean is -0.2608538.

```
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?

ANSWER: The minimum number of principal components required to explain 80% of the variance of the data are seven PCs (PC1-5).

```
# Scale the wisc.data data using the "scale()" function
data.scaled <- scale(wisc.data)
```

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

```
disc.hclust <- hclust(data.dist, method="complete", members=NULL)
disc.hclust
```

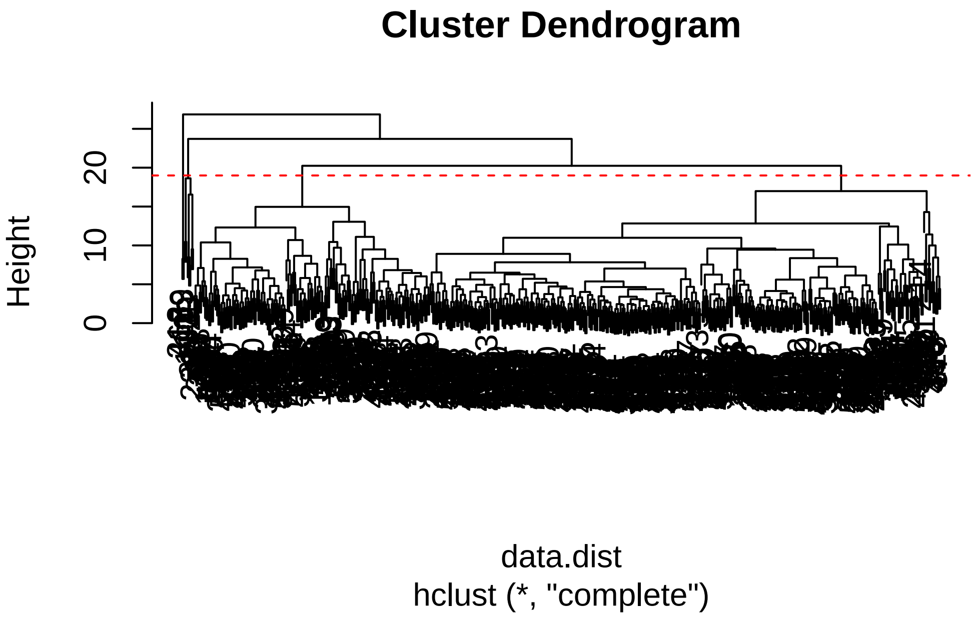
```
Call:
hclust(d = data.dist, method = "complete", members = NULL)

Cluster method   : complete
Distance         : euclidean
Number of objects: 569
```

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

ANSWER: The height at which the clustering model has 4 clusters is when height is 19.

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



Selecting the number of clusters

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

```
  [1] 1 1 1 2 1 1 1 1 1 2 3 1 1 3 1 1 3 1 1 3 3 3 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
 [38] 3 3 1 3 1 1 1 1 1 3 1 3 3 3 3 3 1 3 3 1 1 3 3 3 3 1 3 1 1 3 3 2 3 1 3 1 1
 [75] 3 3 3 1 2 3 3 1 1 1 3 1 3 1 3 1 3 3 3 3 1 1 3 3 3 3 3 3 3 3 3 1 1 3 1 3 3
[112] 3 3 3 3 3 3 1 1 3 3 1 2 3 3 3 1 3 1 1 3 1 1 3 1 3 3 3 1 3 3 3 3 3 3 3 1 3
[149] 3 3 3 3 2 3 3 3 1 3 3 3 3 1 1 3 1 3 3 3 1 3 3 3 1 3 3 3 3 1 3 3 1 1 1 3 1
[186] 3 3 3 3 3 2 3 3 1 1 3 1 3 1 1 3 3 1 1 3 3 3 3 1 3 1 3 4 3 1 1 3 3 1 1 3 3
[223] 3 1 3 3 3 3 3 1 1 3 3 1 3 3 1 1 3 1 3 3 3 3 1 3 3 3 3 3 1 3 1 3 1 3 1 1 1
[260] 1 1 3 1 3 1 1 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 1 3 3 3 3 3 3 3 3 3 3 3 3
[297] 3 3 3 3 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 1 3 3 3 3 1 1 1 1 3
[334] 3 3 1 3 1 3 1 3 3 3 1 3 3 3 3 3 3 3 1 1 1 3 3 3 3 3 3 3 3 3 3 3 1 1 3 1 1
[371] 1 3 1 1 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 1 3 3 1 1 3 3 3 3 3 3 1 3 3 3 3 3 3
[408] 3 1 3 3 3 3 3 3 3 3 1 3 3 3 1 3 3 3 3 3 3 3 3 1 3 1 1 3 1 3 3 3 3 3 3 3 3
[445] 3 3 1 3 3 1 3 1 3 3 3 3 3 3 3 3 1 4 3 3 3 3 3 3 1 1 3 3 3 3 3 3 3 3 3 1 3
[482] 3 3 3 3 3 3 1 3 3 3 3 1 3 3 3 1 3 1 1 3 1 3 1 3 3 3 3 3 1 3 3 1 3 3 3 1 1
[519] 3 3 3 1 3 3 3 3 1 3 3 3 3 3 3 1 3 1 3 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3
[556] 3 3 3 3 3 3 3 1 1 1 1 3 1 3
```

```
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?

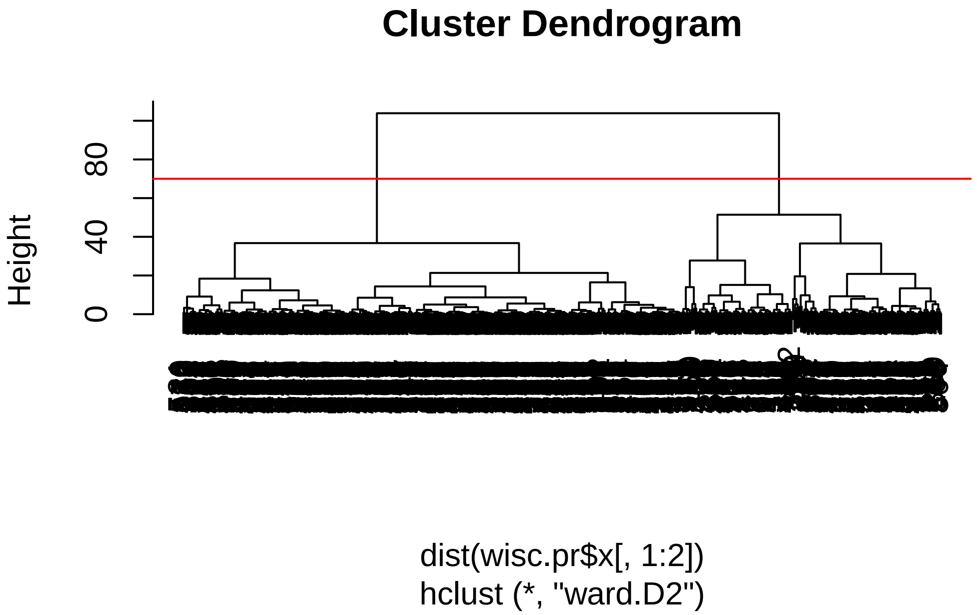
ANSWER: No I could not find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10. When I tried clusters 2-10, the main clusters, 1 and 3 still remained prominent, and did not change significantly.

Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

ANSWER: “ward.D2” is my most favorite resuls for the same data.dsit dataset, beause the data seems cleaners. The method “single” was very messy, which did not allow for any data analysis to be done.

```
q13 <- hclust(dist(wisc.pr$x[,1:2]), method= "ward.D2")

plot(q13)
abline(h=70, col="red")
```



OPTIONAL: K-means clustering- K means clustering and comparing results

```
wisc.km <- kmeans(wisc.data, centers=2, nstart= 20)

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

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

```
table(wisc.hclust.clusters, wisc.km$cluster)
```

```
wisc.hclust.clusters   1   2
                   1  68 109
```

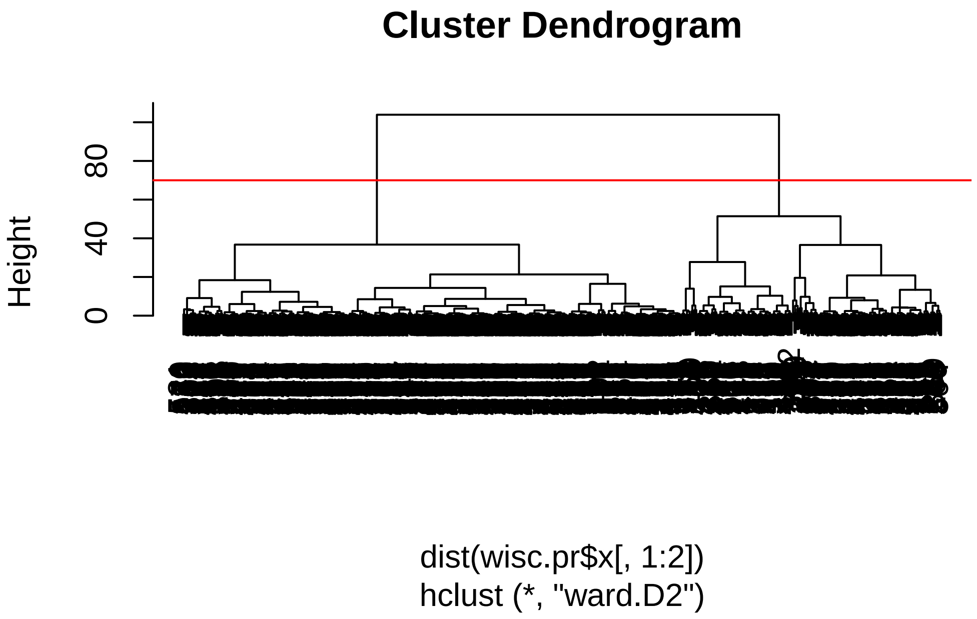
```
  2   5   2
  3 365  18
  4   0   2
```

Combining Methods

Clustering in PC space

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

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



Cluster membership vector

```
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 correspond to the expert diagnosis vector of M and B values

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

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

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

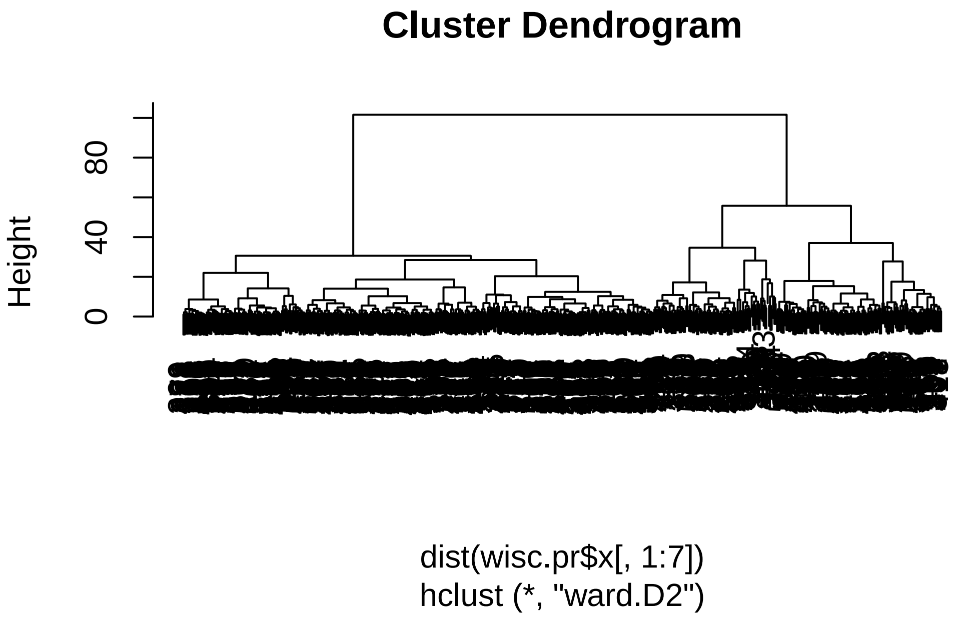
True= cluster/group 1 False= cluster/group 2

True Positive is 177 False Positive is 35

True Negative is 18 False Negative is 339

Combining Methods: Clustering on PCA results

```
wisc.pr.hclust<- hclust(dist(wisc.pr$x[,1:7]), method= "ward.D2")
plot(wisc.pr.hclust)
```



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

```
grps
  1   2 
216 353 
```

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

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

```r
## Use the distance along the first 7 PCs for clustering i.e. wisc.pr$x[, 1:7]
wisc.pr.hclust <- hclust(dist(wisc.pr$x[, 1:7]), method="ward.D2")

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

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

ANSWER: The newly created model with four clusters vs two clusters is about the same, because there are apparent two groups. The table with four cluster groups that are distinct in the table with two custer groups are clusters 1 and 3. Therefore, we conclude, that the two tables separate with around the same results.

```
# Compare to actual diagnoses
table(wisc.pr.hclust.clusters, diagnosis)
```

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

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.

ANSWER: The k-means model did a good job in terms of separating the diagnoses. In the table with the wisc.km$cluster, we see that there is only 1 benign vs 12 benign in the hiearchial cluster table. In addition, we see that there are 2 benign and 5 malignant in the hiarchial cluster table, which is varies on one's interpretation. It is prefered to have more distinct groups because that means the table filtered out ones that may be considered an outier.

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

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

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

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

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

> **ANSWER**: Using the sensitivity formula (TP)/(TP+FN) and the specificity formula (TN)/(TN+FP), I can conclude that PCA clustering is better than Km. Km's senitivity value is [(130)/(130+356)]=0.2675, while PCA is [(188)/(188+329)]=0.3636. PCA having a higher sensitivity value means that it detects more malignant cases (TP) than Km, which means there are fewer fewer malignant cases being misidentified as benign (FN). Km's specificity value is [(1)/(1+82)]=0.012, while PCA is [(28)/(28+24)]=0.538. PCA having a higher specifcity value means there are less benign cases being misidentified as malignant (FP). Km's very small value of 0.012 means that this method is not very specific, and thus most benign cases are being identified as malignant.

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

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

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

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

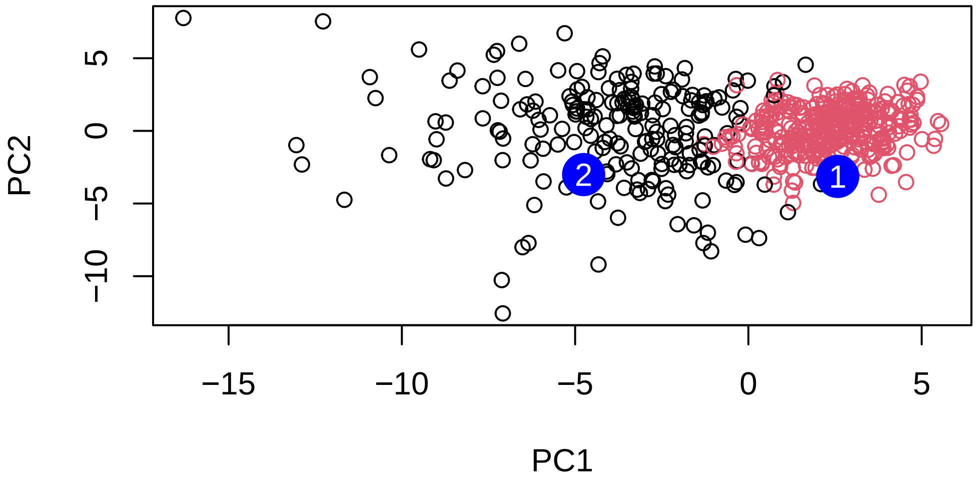
7. Prediction We can use our PCA results(wisc.pr) to make predictions on new unseen data.

```
#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=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?

ANSWER:Since patient 2 is clearly clustered with the malignant (black) cluster compared to patient 1 that is clustered with the benign (red) cluster, we should prioritize patient 2. Patient 2 is the patient that most likely has malignant cells that should be treated sooner than patient 1 with benign cells.