class08: PCA Mini Project

Saba Heydari Seradj (PID: A17002175

It is important to consider scaling your data before PCA.

For example:

```
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
```

```
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 
```

```r
x <- scale(mtcars)
head(x)
```

```
                          mpg        cyl        disp         hp       drat
Mazda RX4           0.1508848 -0.1049878 -0.57061982 -0.5350928  0.5675137
Mazda RX4 Wag       0.1508848 -0.1049878 -0.57061982 -0.5350928  0.5675137
Datsun 710          0.4495434 -1.2248578 -0.99018209 -0.7830405  0.4739996
Hornet 4 Drive      0.2172534 -0.1049878  0.22009369 -0.5350928 -0.9661175
Hornet Sportabout  -0.2307345  1.0148821  1.04308123  0.4129422 -0.8351978
Valiant            -0.3302874 -0.1049878 -0.04616698 -0.6080186 -1.5646078
                             wt       qsec         vs         am       gear
Mazda RX4          -0.610399567 -0.7771651 -0.8680278  1.1899014  0.4235542
Mazda RX4 Wag      -0.349785269 -0.4637808 -0.8680278  1.1899014  0.4235542
Datsun 710         -0.917004624  0.4260068  1.1160357  1.1899014  0.4235542
Hornet 4 Drive     -0.002299538  0.8904872  1.1160357 -0.8141431 -0.9318192
Hornet Sportabout   0.227654255 -0.4637808 -0.8680278 -0.8141431 -0.9318192
Valiant             0.248094592  1.3269868  1.1160357 -0.8141431 -0.9318192
                         carb
Mazda RX4           0.7352031
Mazda RX4 Wag       0.7352031
Datsun 710         -1.1221521
Hornet 4 Drive     -1.1221521
Hornet Sportabout  -0.5030337
Valiant            -1.1221521
```

```r
round(colMeans(x),2)
```

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

```r
round(apply(x,2,sd),2)
```

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

The mean has been scaled to 0, and the SD to 1.

Unsupervised Learning Analysis of Human Breast Cancer Cells

1) Exploratory Data Analysis

```r
# Save your input data file into your Project directory
fna.data <- "WisconsinCancer.csv"

# Complete the following code to input the data and store as wisc.df
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
```

We don't want to pass the 'diagnosis' to the PCA, that is just the expert answer that we will later compare our analysis results to. So, we will remove it and also make a vector called 'diagnosis'.

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

```
# Create diagnosis vector for later
diagnosis <- wisc.df$diagnosis
```

- **Q1**. How many observations are in this dataset?

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

```
[1] 569
```

There are 569 observations in the dataset.

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

```r
# Use table to see how many observations have malignant diagnosis
table(diagnosis)['M']
```

```
  M
212
```

There are 212 malignant diagnoses.

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

```r
mean_names <- grep("_mean$", colnames(wisc.df))
length(mean_names)
```

```
[1] 10
```

10 columns have `_mean` in their name.

PCA

```r
# Check column means and standard deviations
round(colMeans(wisc.data),2)
```

```
            radius_mean            texture_mean          perimeter_mean
                  14.13                   19.29                   91.97
              area_mean         smoothness_mean        compactness_mean
                 654.89                    0.10                    0.10
         concavity_mean     concave.points_mean           symmetry_mean
                   0.09                    0.05                    0.18
 fractal_dimension_mean               radius_se              texture_se
                   0.06                    0.41                    1.22
           perimeter_se                 area_se           smoothness_se
                   2.87                   40.34                    0.01
         compactness_se            concavity_se       concave.points_se
```

```
                 0.03                     0.03                     0.01
          symmetry_se     fractal_dimension_se             radius_worst
                 0.02                     0.00                    16.27
        texture_worst          perimeter_worst               area_worst
                25.68                   107.26                   880.58
     smoothness_worst        compactness_worst          concavity_worst
                 0.13                     0.25                     0.27
 concave.points_worst           symmetry_worst  fractal_dimension_worst
                 0.11                     0.29                     0.08
```

```r
round(apply(wisc.data,2,sd),2)
```

```
          radius_mean             texture_mean           perimeter_mean
                 3.52                     4.30                    24.30
            area_mean          smoothness_mean         compactness_mean
               351.91                     0.01                     0.05
       concavity_mean      concave.points_mean            symmetry_mean
                 0.08                     0.04                     0.03
fractal_dimension_mean                radius_se               texture_se
                 0.01                     0.28                     0.55
         perimeter_se                  area_se            smoothness_se
                 2.02                    45.49                     0.00
       compactness_se             concavity_se        concave.points_se
                 0.02                     0.03                     0.01
          symmetry_se     fractal_dimension_se             radius_worst
                 0.01                     0.00                     4.83
        texture_worst          perimeter_worst               area_worst
                 6.15                    33.60                   569.36
     smoothness_worst        compactness_worst          concavity_worst
                 0.02                     0.16                     0.21
 concave.points_worst           symmetry_worst  fractal_dimension_worst
                 0.07                     0.06                     0.02
```

```r
# Perform PCA on wisc.data
wisc.pr <- prcomp(wisc.data, scale = TRUE)
# Look at summary of 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 your results, what proportion of the original variance is captured by the first principal components (PC1)?

0.4427 of the original variance is captured by PC1.

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

Looking at the cumulative proportion, we can see that 3 PCs are requird to describe at least 70% of the original variance.

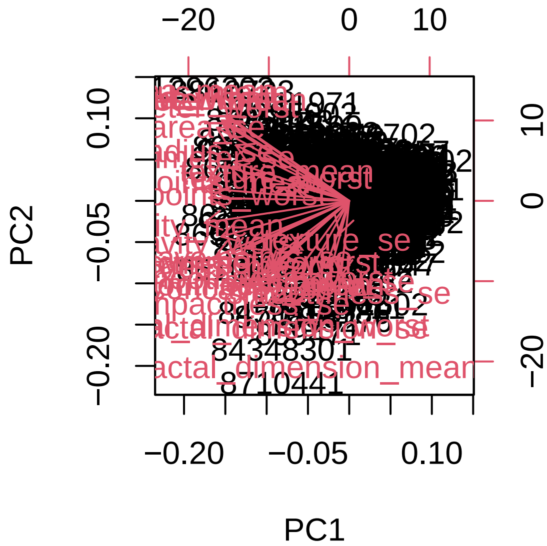
> **Q6.** How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

Looking at the cumulative proportion, we can see that 7 PCs are requird to describe at least 70% of the original variance.

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

It is very messy and difficult to understand.

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



Let's look at what is in this `wisc.pr`

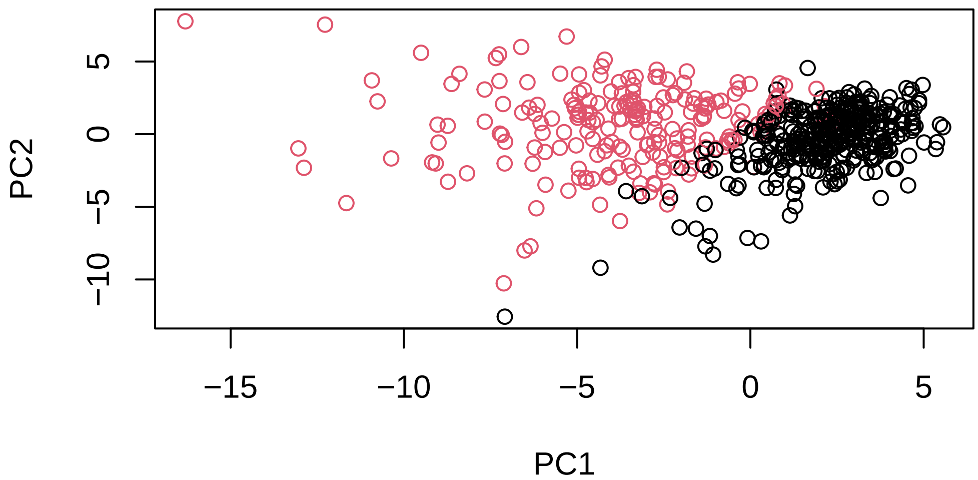
```
attributes(wisc.pr)
```

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

$class
[1] "prcomp"
```

Main 'PC score plot', 'PC1 vs PC2 plot'

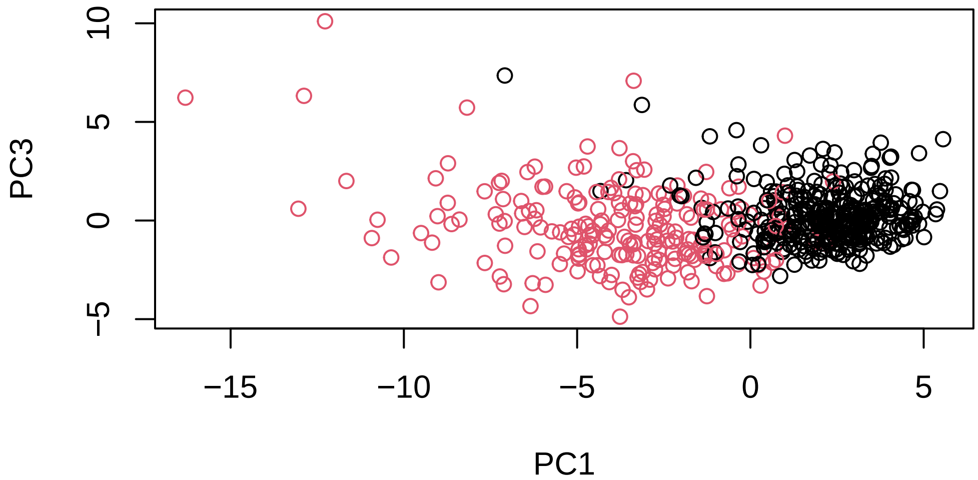
```
plot(wisc.pr$x[,1],wisc.pr$x[,2],col=as.factor(diagnosis),xlab = "PC1", ylab = "PC2")
```




```
# or plot(wisc.pr$x,col=as.factor(diagnosis))
```

Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots? It is kind of similar to PC1 vs PC2 plot, but there is more overlap.

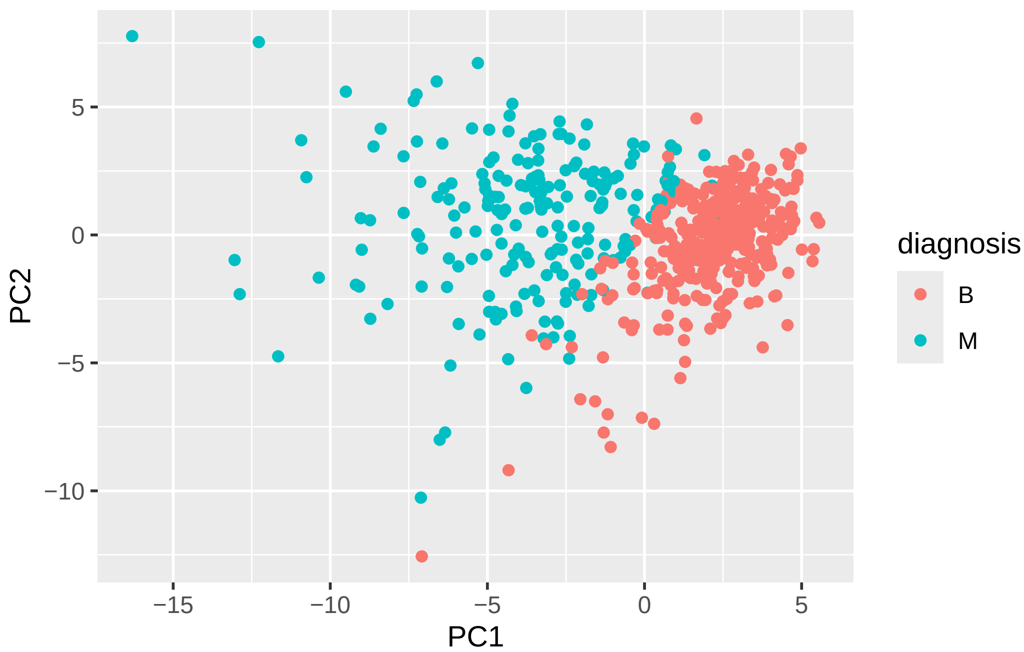
```
plot(wisc.pr$x[,1],wisc.pr$x[,3],col=as.factor(diagnosis),xlab = "PC1", ylab = "PC3")
```




```
# Create a data.frame for ggplot
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis

# Load the ggplot2 package
library(ggplot2)

# Make a scatter plot colored by diagnosis
ggplot(df) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```

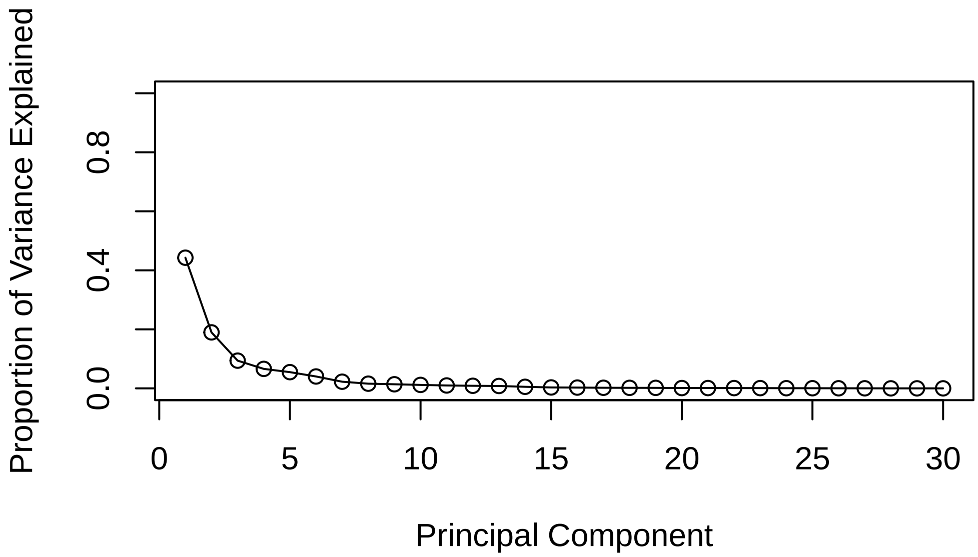



```r
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

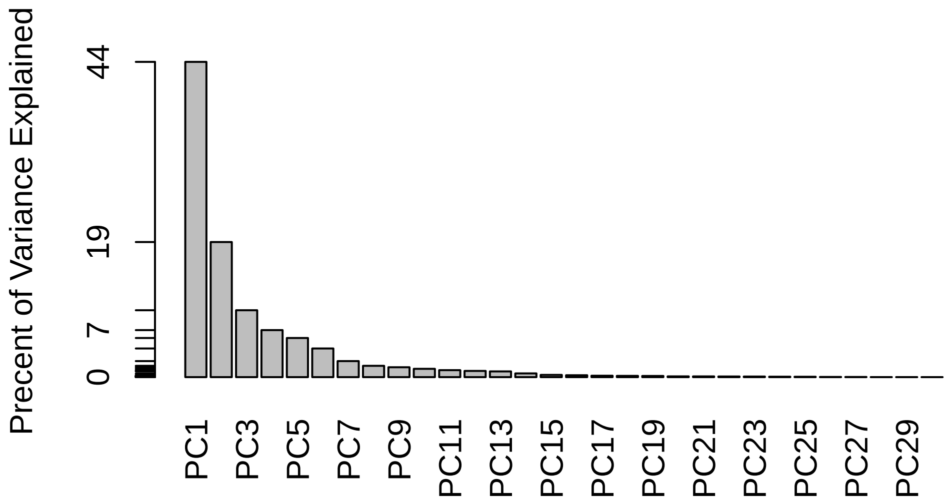
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

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



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?

It is -0.2608538.

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

```
                        PC1         PC2          PC3         PC4         PC5
radius_mean      -0.2189024  0.23385713 -0.008531243  0.04140896 -0.03778635
texture_mean     -0.1037246  0.05970609  0.064549903 -0.60305000  0.04946885
perimeter_mean   -0.2275373  0.21518136 -0.009314220  0.04198310 -0.03737466
area_mean        -0.2209950  0.23107671  0.028699526  0.05343380 -0.01033125
smoothness_mean  -0.1425897 -0.18611302 -0.104291904  0.15938277  0.36508853
compactness_mean -0.2392854 -0.15189161 -0.074091571  0.03179458 -0.01170397
                          PC6         PC7          PC8          PC9        PC10
radius_mean       0.018740790 -0.12408834  0.007452296 -0.223109764  0.09548644
texture_mean     -0.032178837  0.01139954 -0.130674825  0.112699390  0.24093407
perimeter_mean    0.017308445 -0.11447706  0.018687258 -0.223739213  0.08638562
area_mean        -0.001887748 -0.05165343 -0.034673604 -0.195586014  0.07495649
smoothness_mean  -0.286374497 -0.14066899  0.288974575  0.006424722 -0.06929268
compactness_mean -0.014130949  0.03091850  0.151396350 -0.167841425  0.01293620
                        PC11        PC12       PC13         PC14        PC15
radius_mean      -0.04147149  0.05106746 0.01196721  0.059506135 -0.05111877
```

```
texture_mean      0.30224340  0.25489642 0.20346133 -0.021560100 -0.10792242
perimeter_mean   -0.01678264  0.03892611 0.04410950  0.048513812 -0.03990294
area_mean        -0.11016964  0.06543751 0.06737574  0.010830829  0.01396691
smoothness_mean   0.13702184  0.31672721 0.04557360  0.445064860 -0.11814336
compactness_mean  0.30800963 -0.10401704 0.22928130  0.008101057  0.23089996
                       PC16        PC17         PC18        PC19        PC20
radius_mean      -0.1505839  0.20292425  0.146712338  0.22538466 -0.04969866
texture_mean     -0.1578420 -0.03870612 -0.041102985  0.02978864 -0.24413499
perimeter_mean   -0.1144540  0.19482131  0.158317455  0.23959528 -0.01766501
area_mean        -0.1324480  0.25570576  0.266168105 -0.02732219 -0.09014376
smoothness_mean  -0.2046132  0.16792991 -0.352226802 -0.16456584  0.01710096
compactness_mean  0.1701784 -0.02030771  0.007794138  0.28422236  0.48868633
                        PC21        PC22          PC23        PC24        PC25
radius_mean      -0.06857001 -0.07292890 -0.0985526942 -0.18257944 -0.01922650
texture_mean      0.44836947 -0.09480063 -0.0005549975  0.09878679  0.08474593
perimeter_mean   -0.06976904 -0.07516048 -0.0402447050 -0.11664888  0.02701541
area_mean        -0.01844328 -0.09756578  0.0077772734  0.06984834 -0.21004078
smoothness_mean  -0.11949175 -0.06382295 -0.0206657211  0.06869742  0.02895489
compactness_mean  0.19262140  0.09807756  0.0523603957 -0.10413552  0.39662323
                        PC26        PC27          PC28         PC29
radius_mean      -0.12947640 -0.13152667  2.111940e-01  0.211460455
texture_mean     -0.02455666 -0.01735731 -6.581146e-05 -0.010533934
perimeter_mean   -0.12525595 -0.11541542  8.433827e-02  0.383826098
area_mean         0.36272740  0.46661248 -2.725083e-01 -0.422794920
smoothness_mean  -0.03700369  0.06968992  1.479269e-03 -0.003434667
compactness_mean  0.26280847  0.09774871 -5.462767e-03 -0.041016774
                         PC30
radius_mean       0.702414091
texture_mean      0.000273661
perimeter_mean   -0.689896968
area_mean        -0.032947348
smoothness_mean  -0.004847458
compactness_mean  0.044674186
```

```r
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?

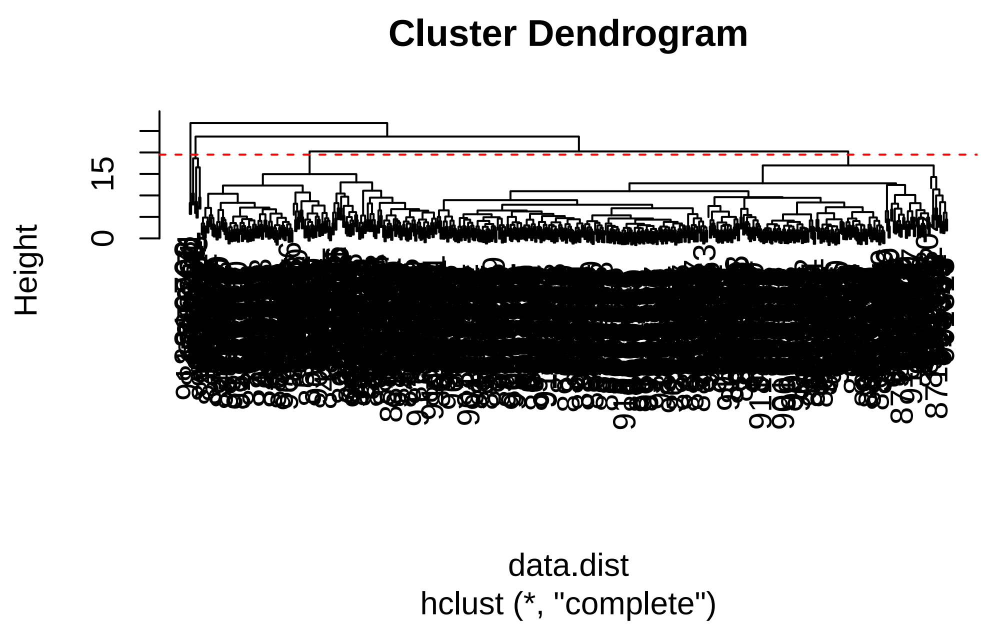
We need 5 PCs to explain 80% of the data.

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

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

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

```r
plot(wisc.hclust)
abline(h = 19.5, col = "red", lty = 2)
```



Q10. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?

It is around 19-20.

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

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

```
                    diagnosis
wisc.hclust.clusters   B   M
```

```
                    1  12 165
                    2   2   5
                    3 343  40
                    4   0   2
```

> Q11. Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10?

```r
wisc.hclust.clusters2 <- cutree(wisc.hclust, k =2)
table(wisc.hclust.clusters, diagnosis)
```

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

Two clusters would not work.

```r
wisc.hclust.clusters3 <- cutree(wisc.hclust, k =3)
table(wisc.hclust.clusters, diagnosis)
```

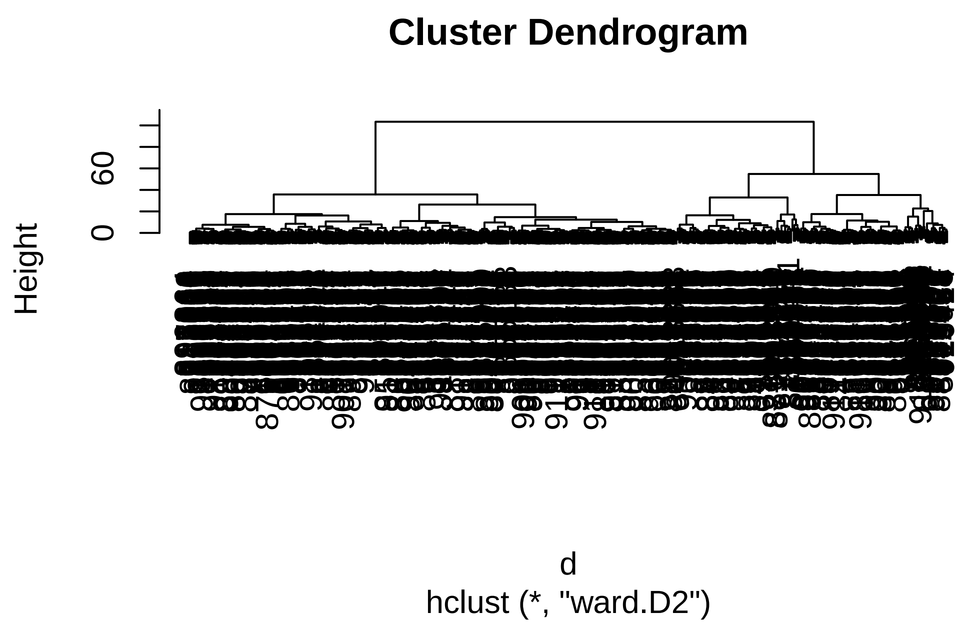
```
                    diagnosis
wisc.hclust.clusters   B   M
                   1  12 165
                   2   2   5
                   3 343  40
                   4   0   2
```

Three clusters is also not going to work. It seems 4 is the smallest number of clusters we can get here that would actually represent the seperation.

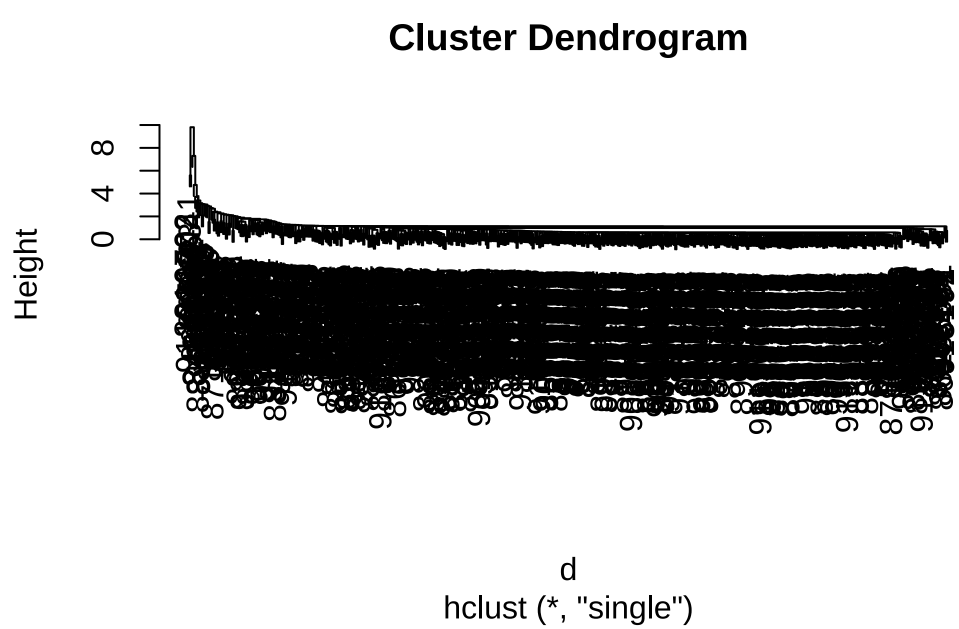
> Q12. Which method gives your favorite results for the same data.dist dataset?

Looking at single, complete, average and ward.D2 methods, my favorite is ward.D2. It is the cleanest-looking tree with good seperation and long branch heights. Ward.D2 method minimizes variance within clusters.

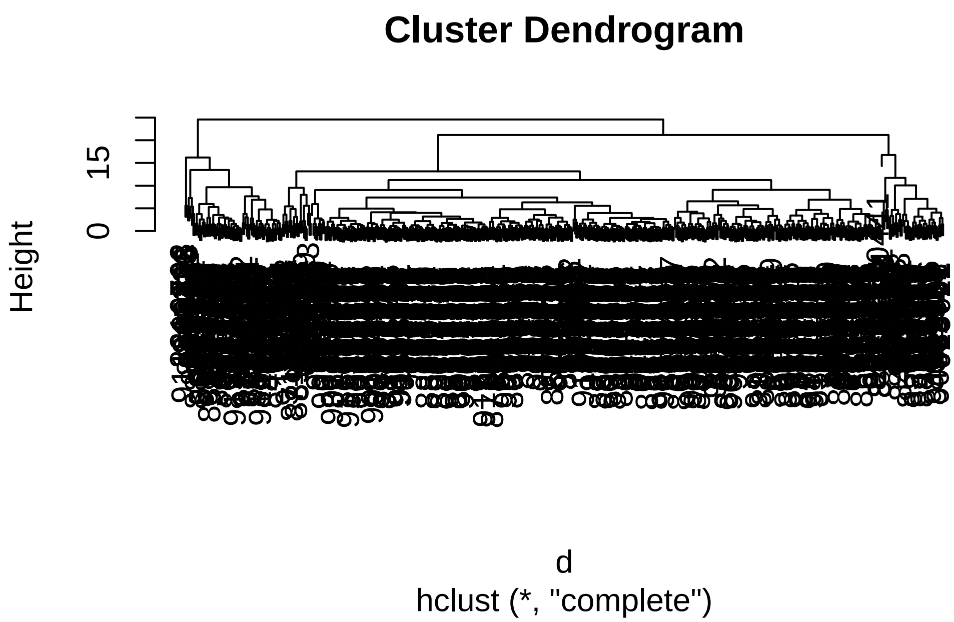
```r
d <- dist(wisc.pr$x[,1:3])
hc <- hclust(d, method='ward.D2')
plot(hc)
```



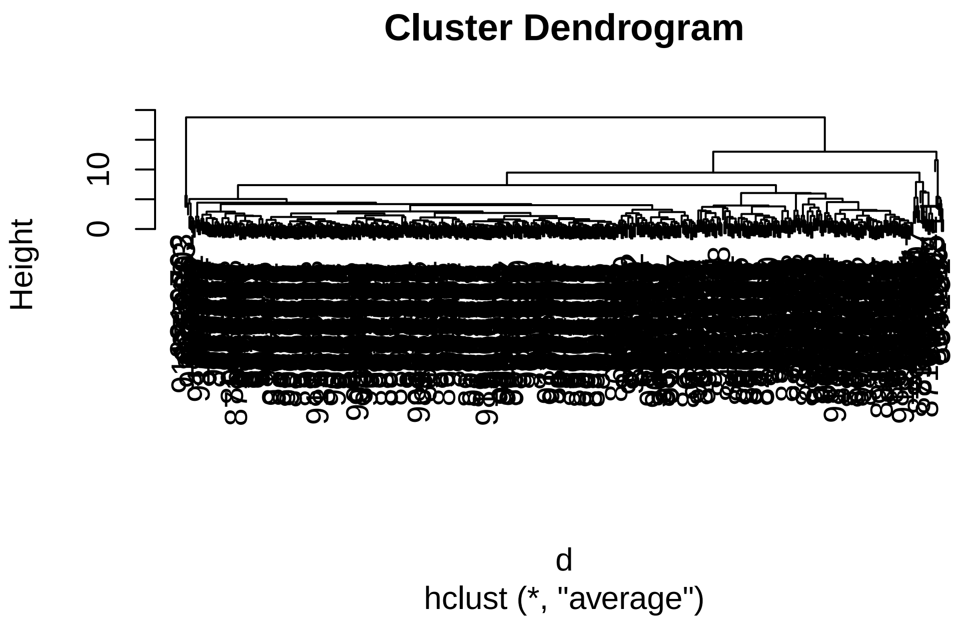

```r
hc2 <- hclust(d, method='single')
plot(hc2)
```



```r
hc3 <- hclust(d, method='complete')
plot(hc3)
```



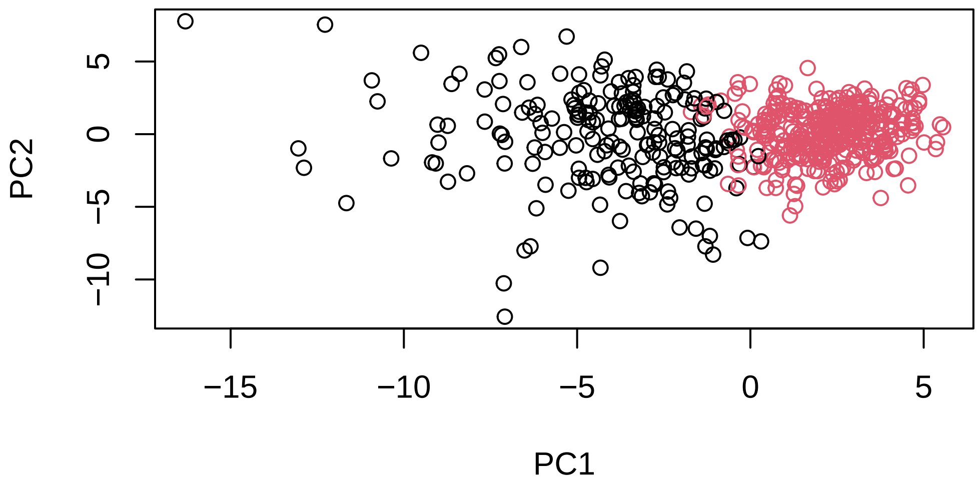

```r
hc4 <- hclust(d, method='average')
plot(hc4)
```




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

```
    diagnosis
grps   B   M
   1  24 179
   2 333  33
```

```
plot(wisc.pr$x, col=grps)
```




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

```
         grps
diagnosis   1   2
        B  24 333
        M 179  33
```

> Q13. How well does the newly created model separate out the two diagnoses?

The model using PCA with two clusters looks pretty good and as we can see from the table results it has done quite a good job in correctly clustering the datapoints.

> Q14. 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

As we can see, the model without PCA does a poor job at separating the diagnoses and requires 4 cluster minimum to yield any meaningful results like below. The output after PCA however looks much better and can separate samples well with two clusters.

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

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

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

```
         grps
diagnosis   1   2
        B  24 333
        M 179  33
```