Class 8 Mini Project: PCA

Yujia Liu

It is important to consider scalling your sata before analysis such as 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) #Very different values
```

```
      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 
```

```
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
```

```
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
```

Key-point: It is usually always a good idea to scale your data before to PCA…

Breast Cancer Biopsy Analysis

```
# 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)
```

1. Exploratory data analysis

```
# We can use -1 here to remove the first column since the first column diagnosis contains the
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
```

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

```
table(diagnosis)
```

```
diagnosis
  B   M
357 212
```

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

```
variable_names <- colnames(wisc.data)
variable_names
```

```
 [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"
```

```
title_with_mean <- grep("_mean", variable_names, ignore.case = FALSE, value = TRUE)
length(title_with_mean)
```

```
[1] 10
```

2. Principle component analysis

```
# Check column means and standard deviations
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
```

```
# Perform PCA on scaled wisc.data
wisc.pr <- prcomp(wisc.data, scale = T )
```

```
# 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

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

3

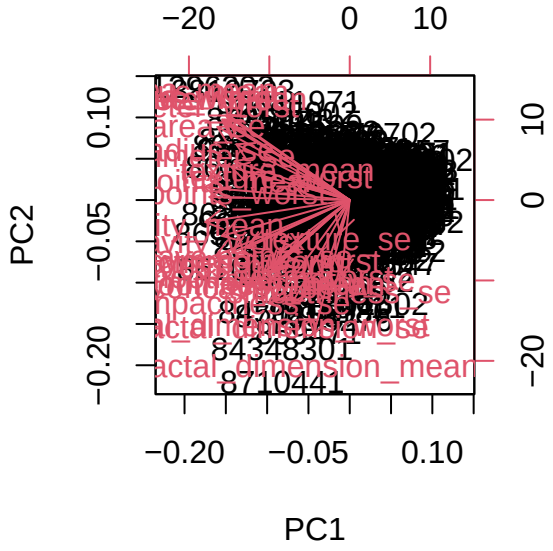
Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

7

Interpretation PCA results

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

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



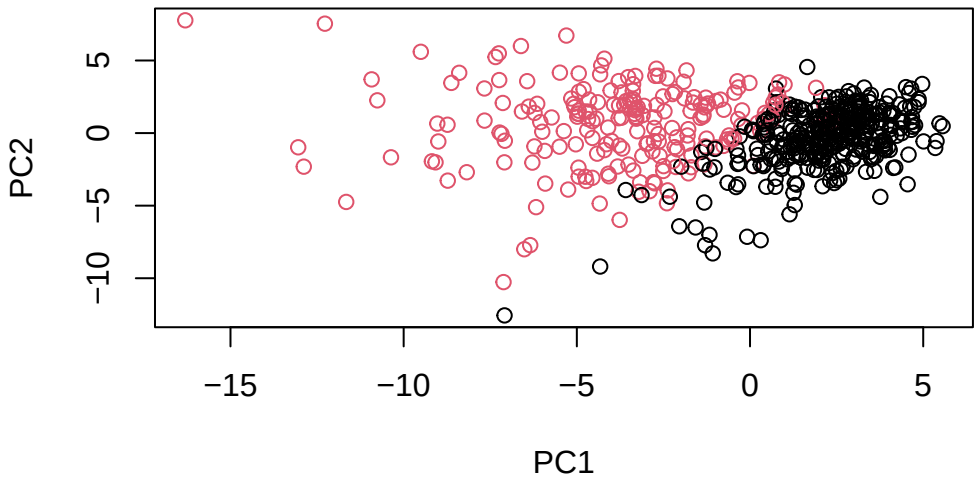
```
#Difficult to understand
```

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

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

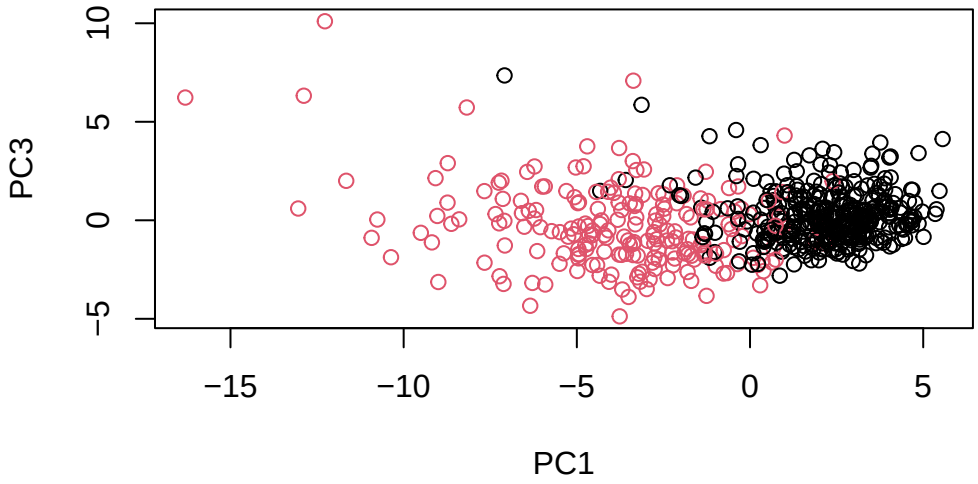
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

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



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

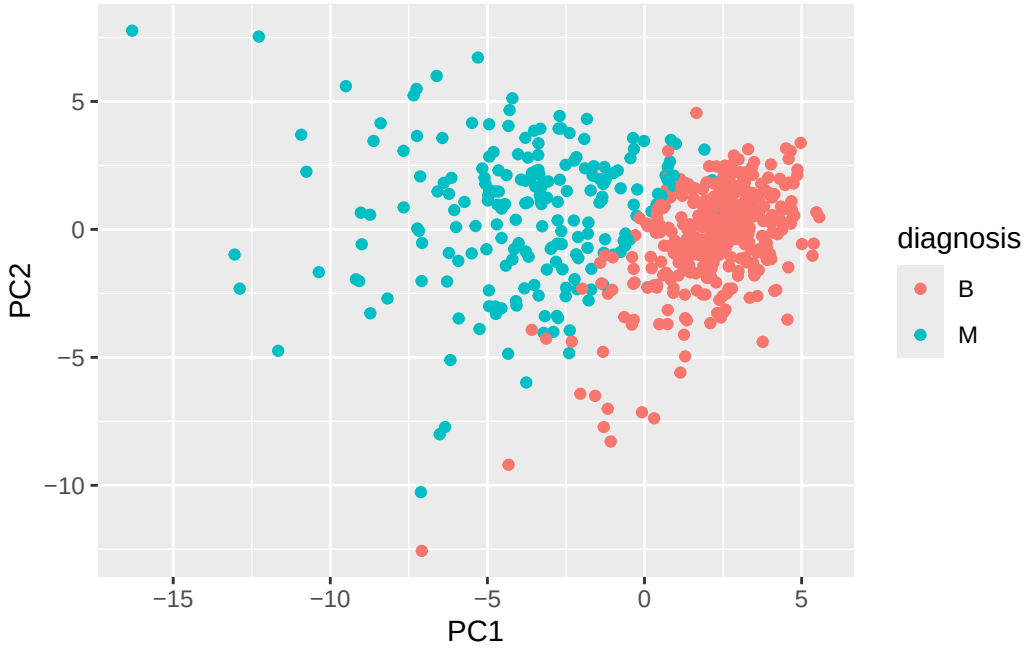
```
plot(wisc.pr$x[,1], wisc.pr$x[,3],
     col = as.factor(diagnosis),
     xlab = "PC1", ylab = "PC3")
```



PC2 explains more variance in the original data than PC3, the first plot "PC1 vs PC2" has a cleaner cut separating the two subgroups (malignant and benign). Overall, the plots indicate that PC1 is capturing a separation of malignant (red) from benign (black) samples.

As this is such a striking result let's see if we can use the ggplot2 package to make a more fancy figure of these results. Remember that `ggplot` requires a data.frame as input and we will also need to add our diagnosis vector as a column if we want to use it for mapping to the plot color aesthetic.

```
# 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()
```

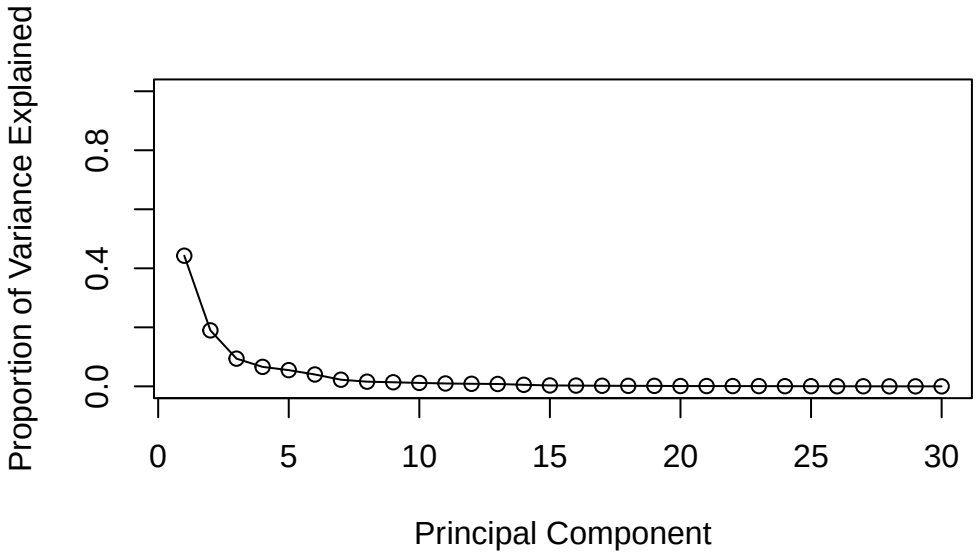


Calculate the variance of each principal component by squaring the sdev component of wisc.pr (i.e. wisc.pr$sdev^2). Save the result as an object called pr.var.

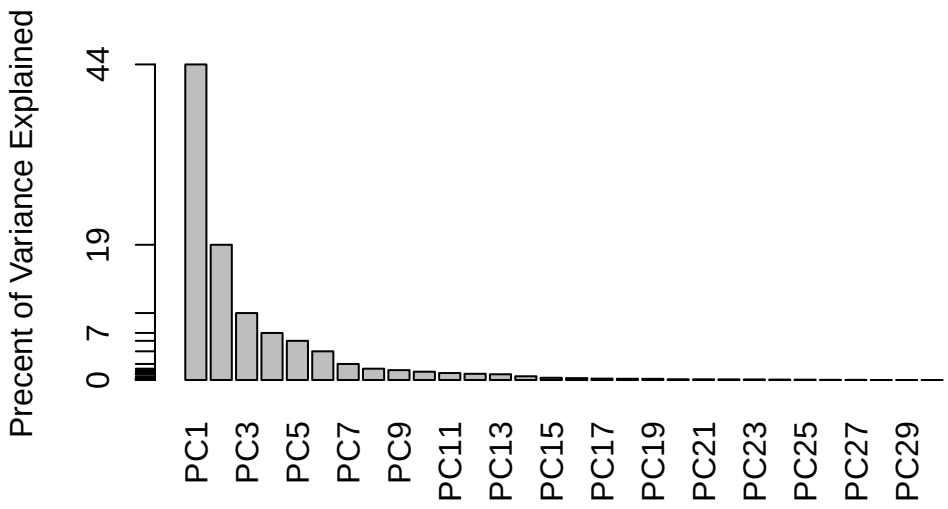
```
# Calculate variance of each component
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

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

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



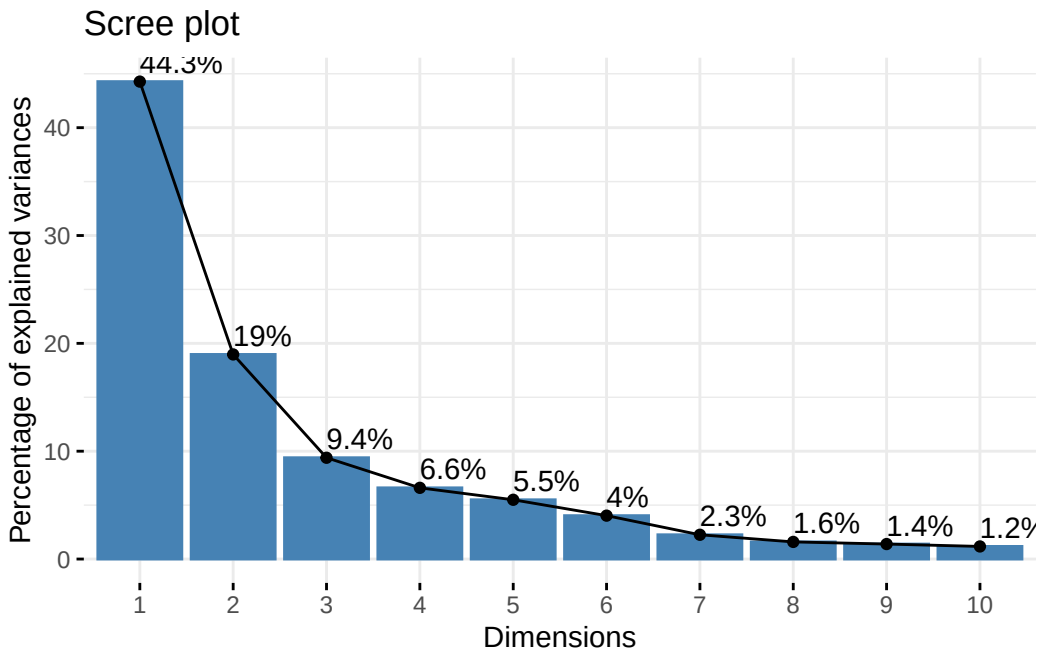

```
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```




```
## 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)
```



Communicating PCA results

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?

```
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
```

minimum of 5 is needed.

3.Hierarchical clustering

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

Calculate the (Euclidean) distances between all pairs of observations in the new scaled dataset and assign the result to "data.dist".

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

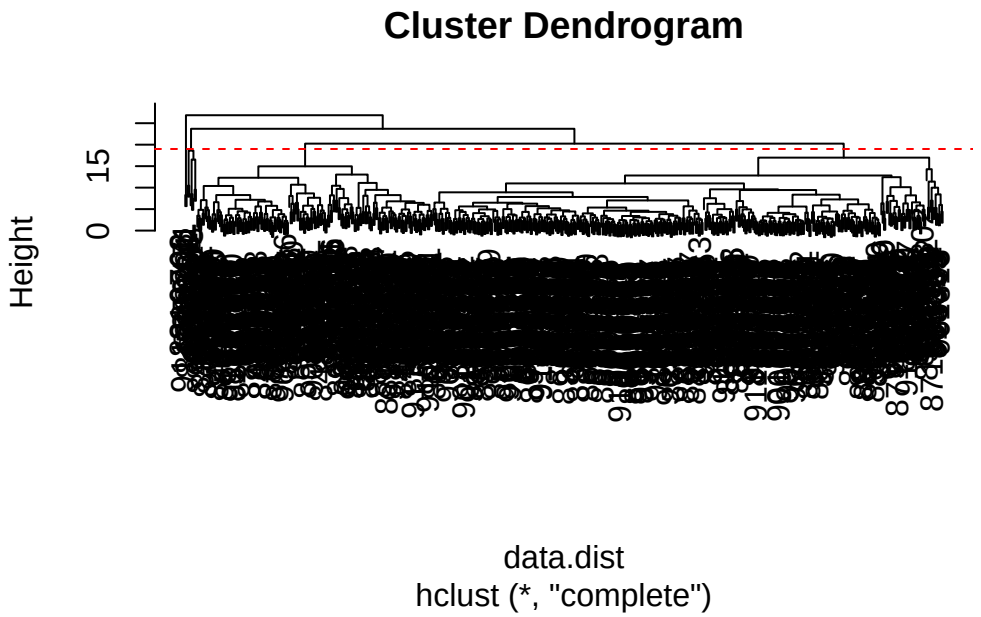
Create a hierarchical clustering model using complete linkage. Manually specify the method argument to hclust() and assign the results to wisc.hclust.

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

```
Call:
hclust(d = data.dist)

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?

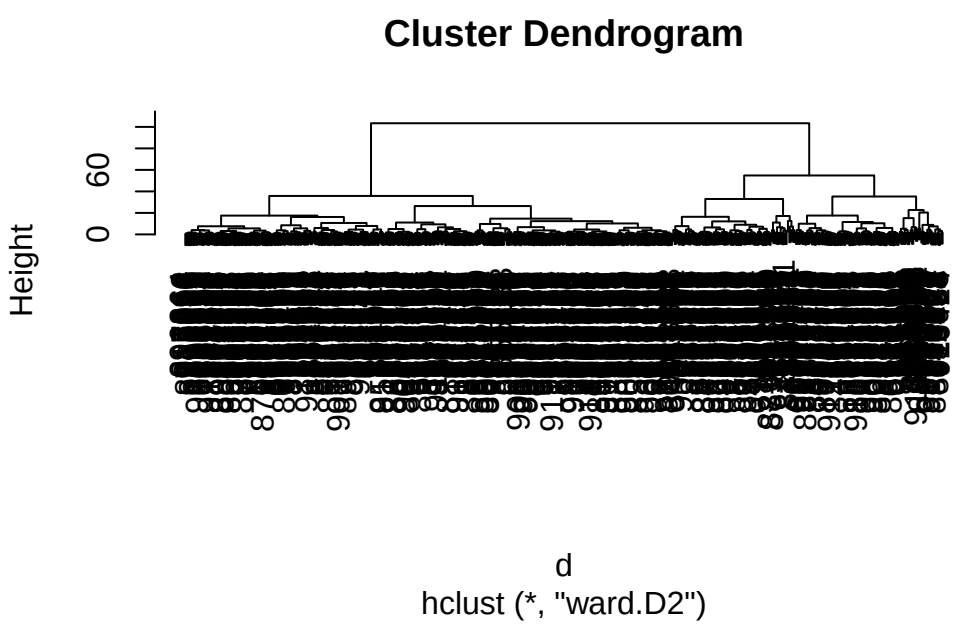
```
plot(wisc.hclust)
abline(h= 19, col="red", lty=2)
```



5. Combine PCA and clustering

Our PCA results were in `wisc.or$x`

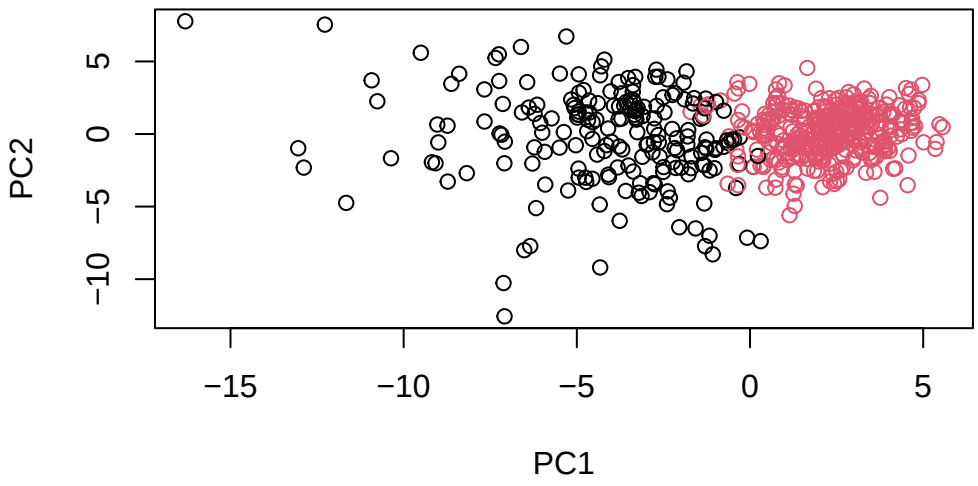
```
#distance matrix
d <- dist(wisc.pr$x[,1:3])
hc <- hclust(d, method = "ward.D2")
plot(hc)
```



Cut tree into two groups

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

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



Compare my clustering result (my `grps`) to the expert `diagnosis`

```
table(diagnosis)
```

```
diagnosis
  B   M
357 212
```

```
table(grps)
```

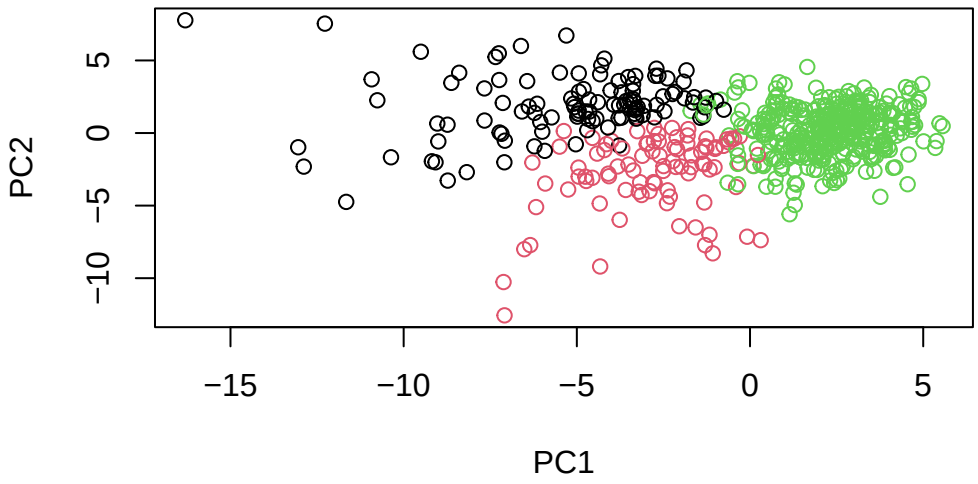
```
grps
  1   2
203 366
```

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

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

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

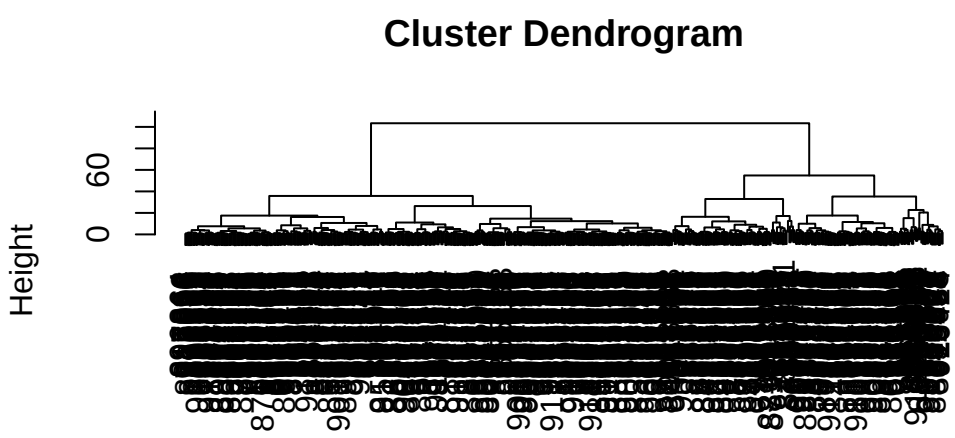
```
grps3 <- cutree(hc, k=3)
plot(wisc.pr$x, col = grps3)
```




```
number <- c(2:10)
func <- function(number){
  grps <- cutree(hc, k= number)
  plot(wisc.pr$x, col = grps)
}
```

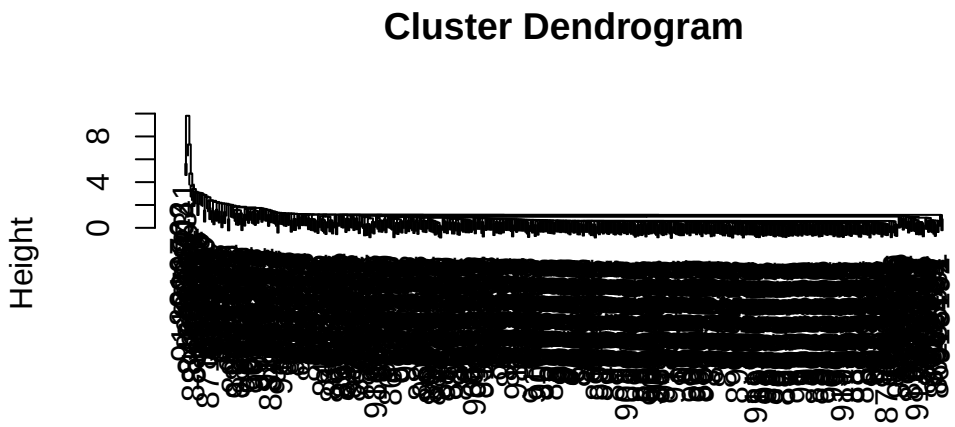
Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

```
plot(hclust(d, method = "ward.D2")) #This is the best! Gives the clear 2 groups
```



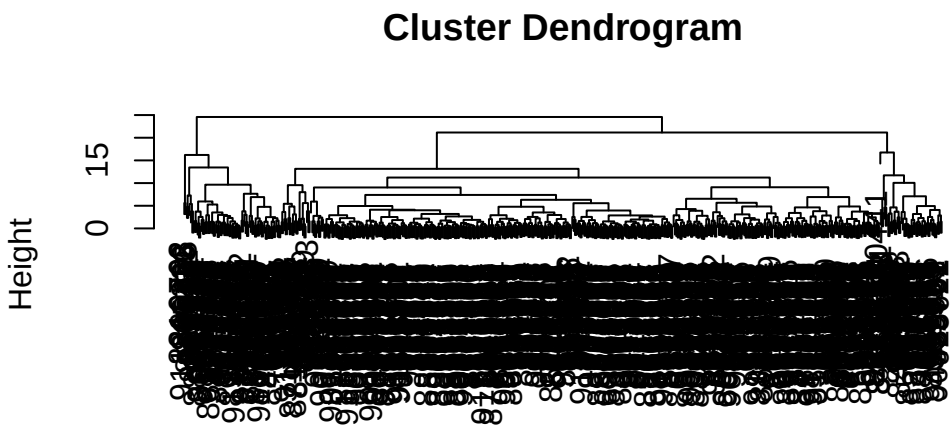
d
hclust (*, "ward.D2")

```
plot(hclust(d, method = "single"))
```



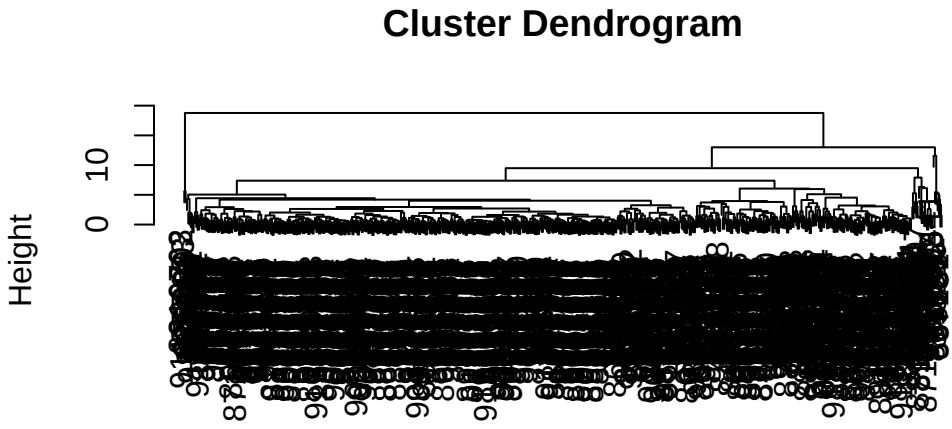
d
hclust (*, "single")

```r
plot(hclust(d, method = "complete"))
```



d
hclust (*, "complete")

```r
plot(hclust(d, method = "average"))
```



d
hclust (*, "average")

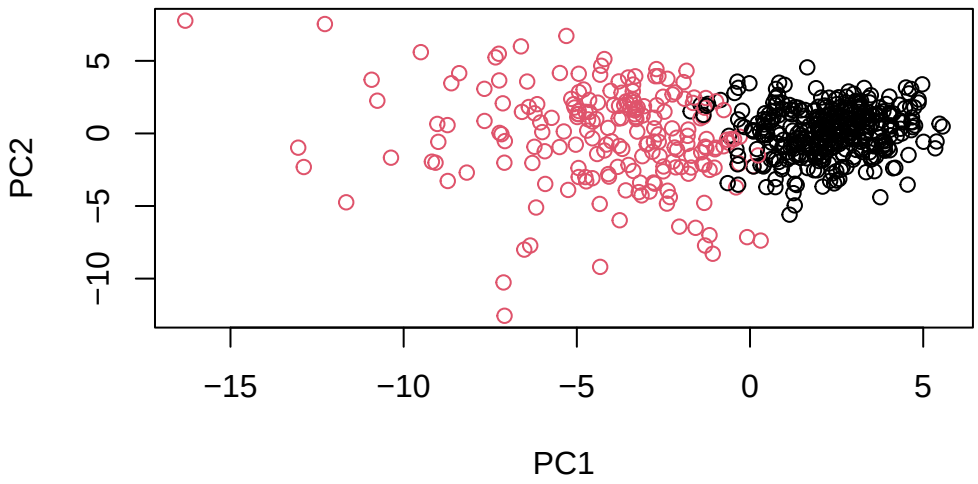
```
g <- as.factor(grps)
levels(g)
```

```
[1] "1" "2"
```

```
g <- relevel(g,2)
levels(g)
```

```
[1] "2" "1"
```

```
# Plot using our re-ordered factor
plot(wisc.pr$x[,1:2], col=g)
```




```
## Use the distance along the first 7 PCs for clustering i.e. wisc.pr$x[, 1:7]
wisc.pr.hclust <- hclust(d, 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?

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

```
                     diagnosis
wisc.pr.hclust.clusters   B   M
                      1  24 179
                      2 333  33
```

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

```
##    diagnosis
##       B   M
##   1  14 175
##   2 343  37

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

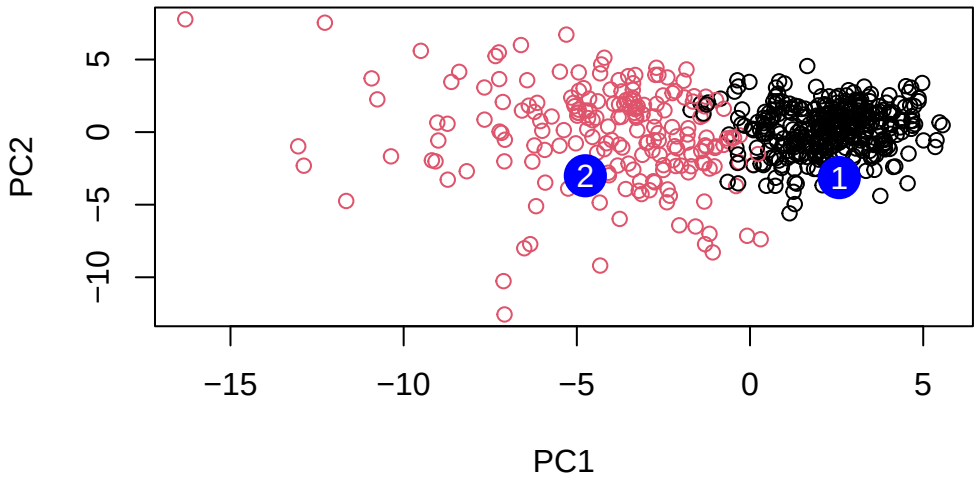
###Prediction

```
#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
```

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

We should prioritize Patient 1 for follow up as his/her symptoms(data) likely align with malignent diagnosis according to PCA.