Class 08: Mini Project

A16442048

1.Exploratory Data Analysis

```
fna.data <- "WisconsinCancer.csv"
```

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

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

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

```
  [1] "M" "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" "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" "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" "B"
```

Q1. How many observations are in this dataset?

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

```
[1] 569
```

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

```
mal <- "M"
sum(diagnosis == "M")
```

```
[1] 212
```

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

```
all_col_names <- colnames(wisc.df)
mean <- "mean"
```

```
mean_suffix <- grep(mean,all_col_names,value ="TRUE")
mean_suffix
```

```
 [1] "radius_mean"            "texture_mean"           "perimeter_mean"
 [4] "area_mean"              "smoothness_mean"        "compactness_mean"
 [7] "concavity_mean"         "concave.points_mean"    "symmetry_mean"
[10] "fractal_dimension_mean"
```

10

2. 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
```

```
wisc.pr <- prcomp(wisc.data)
```

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

```
Importance of components:
                          PC1      PC2      PC3     PC4     PC5     PC6   PC7
Standard deviation     666.170 85.49912 26.52987 7.39248 6.31585 1.73337 1.347
Proportion of Variance   0.982  0.01618  0.00156 0.00012 0.00009 0.00001 0.000
Cumulative Proportion    0.982  0.99822  0.99978 0.99990 0.99999 0.99999 1.000
                           PC8    PC9   PC10   PC11    PC12    PC13    PC14
Standard deviation     0.6095 0.3944 0.2899 0.1778 0.08659 0.05623 0.04649
Proportion of Variance 0.0000 0.0000 0.0000 0.0000 0.00000 0.00000 0.00000
Cumulative Proportion  1.0000 1.0000 1.0000 1.0000 1.00000 1.00000 1.00000
                          PC15   PC16    PC17    PC18    PC19    PC20     PC21
Standard deviation     0.03642 0.0253 0.01936 0.01534 0.01359 0.01281 0.008838
Proportion of Variance 0.00000 0.0000 0.00000 0.00000 0.00000 0.00000 0.000000
Cumulative Proportion  1.00000 1.0000 1.00000 1.00000 1.00000 1.00000 1.000000
                          PC22     PC23     PC24     PC25     PC26     PC27
Standard deviation     0.00759 0.005909 0.005329 0.004018 0.003534 0.001918
Proportion of Variance 0.00000 0.000000 0.000000 0.000000 0.000000 0.000000
Cumulative Proportion  1.00000 1.000000 1.000000 1.000000 1.000000 1.000000
                          PC28     PC29      PC30
Standard deviation     0.001688 0.001416 0.0008379
Proportion of Variance 0.000000 0.000000 0.0000000
Cumulative Proportion  1.000000 1.000000 1.0000000
```

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

44.27%

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

3 PCs

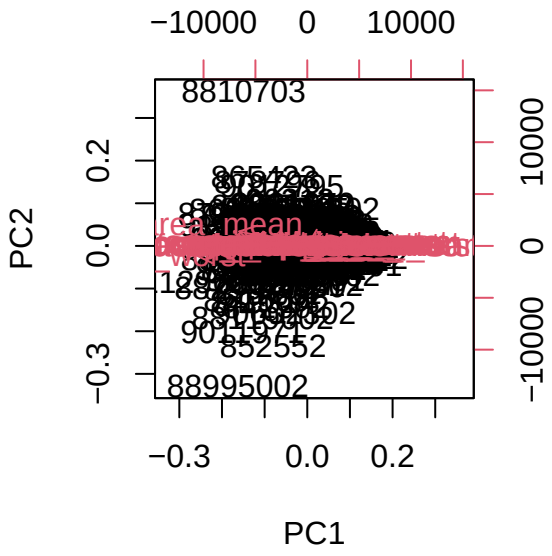
> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

7 PCs

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

```
Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped
```

```
Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped

Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
arrow.len): zero-length arrow is of indeterminate angle and so skipped
```

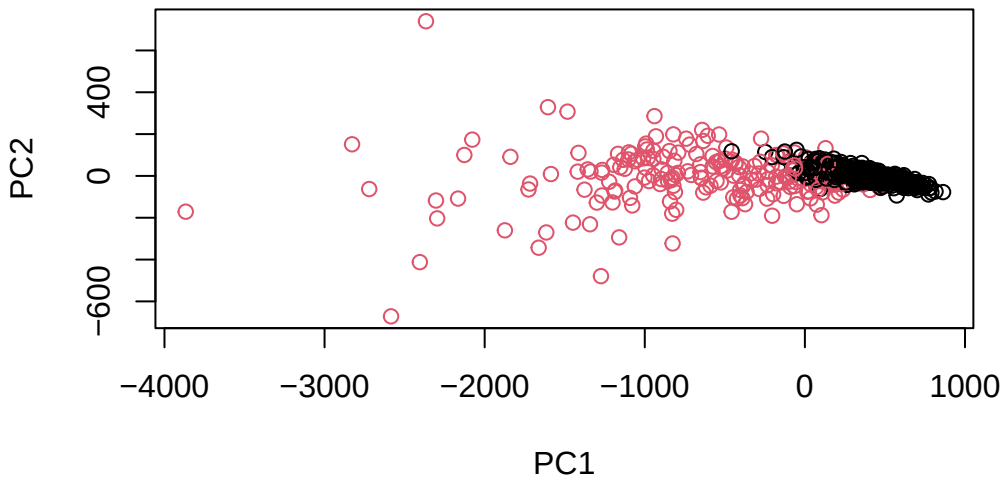


Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

It has 4 axes, everything is large but the plot si small so there is much overlap which makes it hard to understand.

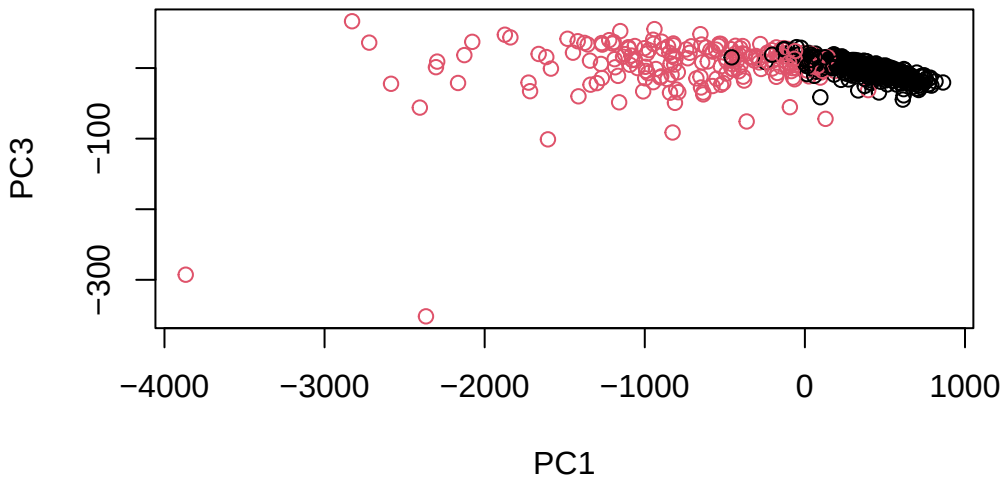
```
library(ggplot2)
```

```
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?

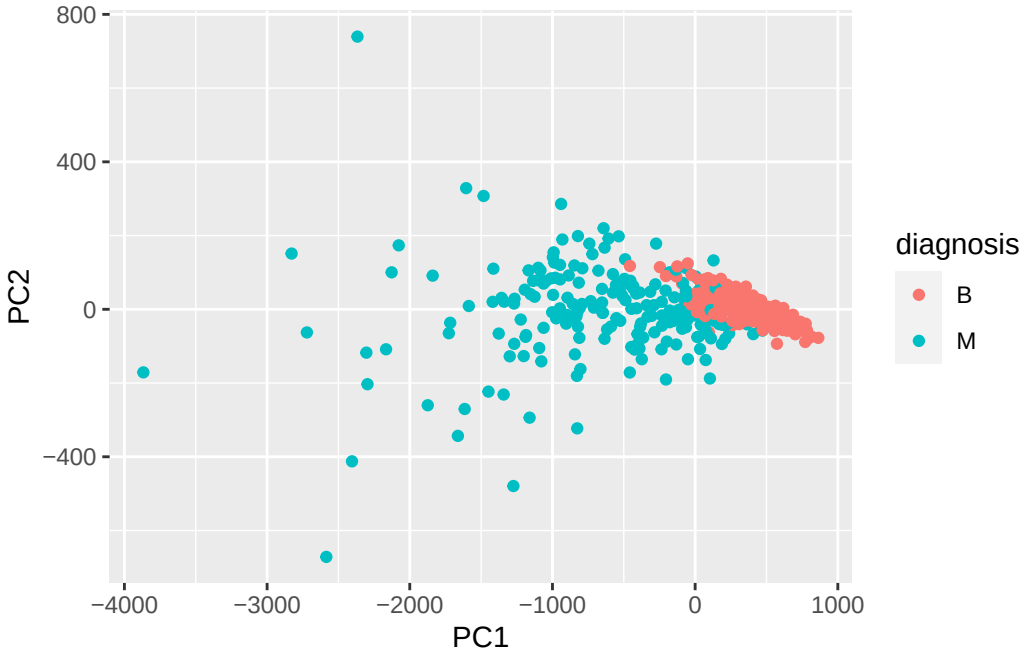
```
plot(wisc.pr$x[,1],wisc.pr$x[,3] , col=as.factor(diagnosis),
     xlab = "PC1", ylab = "PC3")
```



When PC1 is plotted against PC3 rather than PC2, it leans higher on the y axis. THey are also both skewed to the right of the plot a lot. They also both separate benign form malignant diagnoses.

```
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis
```

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




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

```
[1] 4.437826e+05 7.310100e+03 7.038337e+02 5.464874e+01 3.989002e+01
[6] 3.004588e+00
```

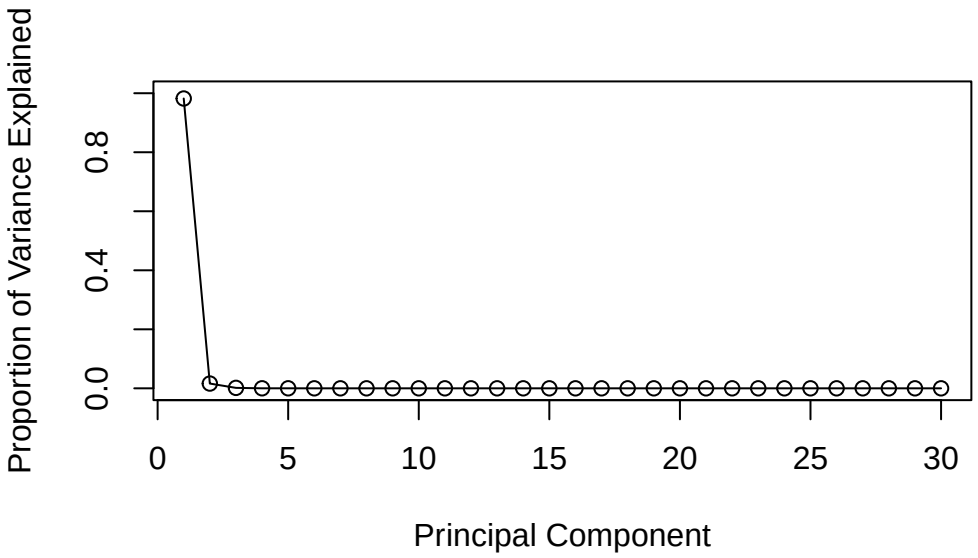
```
total_var <- sum(wisc.pr$sdev^2)
total_var
```

```
[1] 451896.6
```

```
pve <- (wisc.pr$sdev^2) / total_var
pve
```

```
 [1] 9.820447e-01 1.617649e-02 1.557511e-03 1.209320e-04 8.827245e-05
 [6] 6.648840e-06 4.017137e-06 8.220172e-07 3.441353e-07 1.860187e-07
[11] 6.994732e-08 1.659089e-08 6.996416e-09 4.783183e-09 2.935492e-09
[16] 1.416849e-09 8.295777e-10 5.204059e-10 4.084640e-10 3.633134e-10
[21] 1.728497e-10 1.274875e-10 7.726830e-11 6.283577e-11 3.573023e-11
[26] 2.763960e-11 8.144523e-12 6.302115e-12 4.436669e-12 1.553447e-12
```

```
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "o")
```



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?

-4.778078e-05

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

```
cumulative_pve <- cumsum(pve)
which(cumulative_pve >= 0.8)[1]
```

```
[1] 1
```

3.Hierarchical Clustering

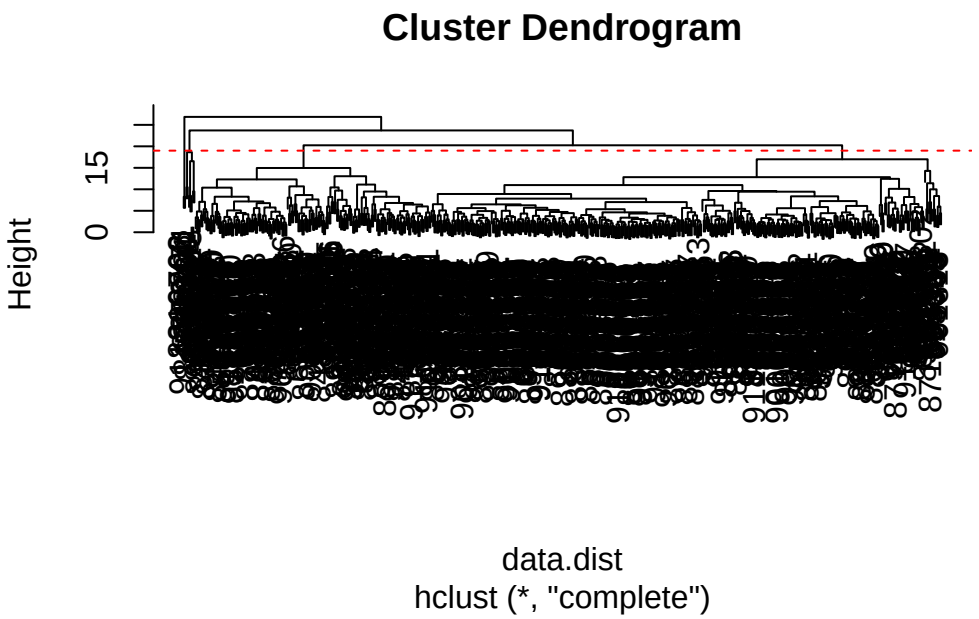
```
scaled <- scale(wisc.data)
```

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

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

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)
```



19

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

```
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?

You can try multiple numbers between 2 and 10 and then decide which one is a better match.

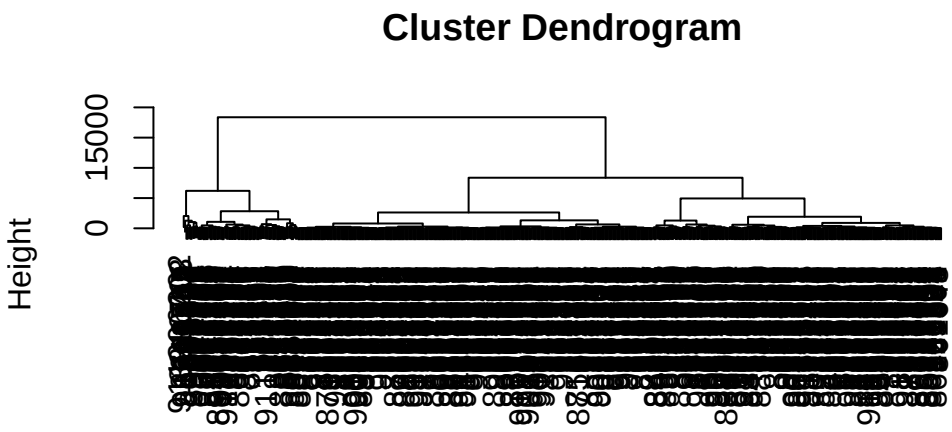
> Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

"ward.D2" because it shows the cleanest clusters and is the easiest to interpret for me

5.Combining Methods

```
cum.var <- cumsum(summary(wisc.pr)$importance[2, ])
num.comp <- which.max(cum.var >= 0.9)
```

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



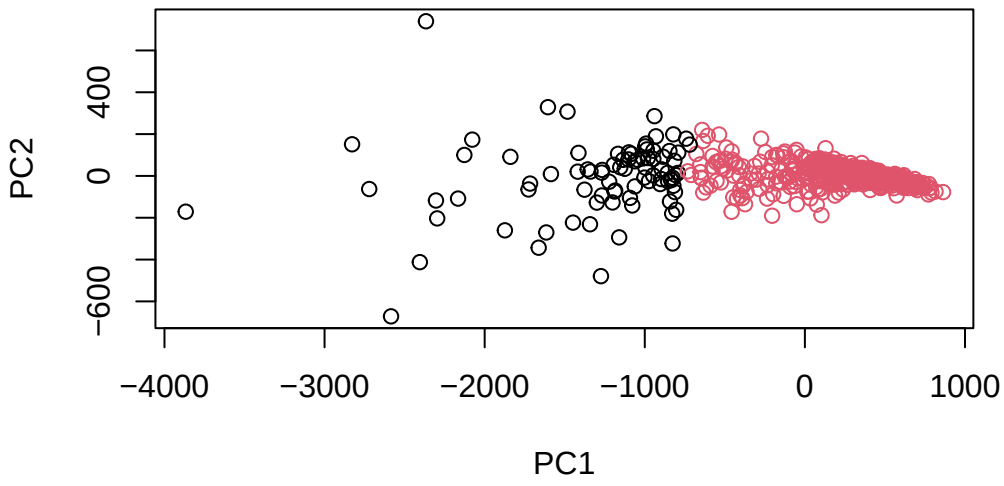

```
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
  1   2
 86 483
```

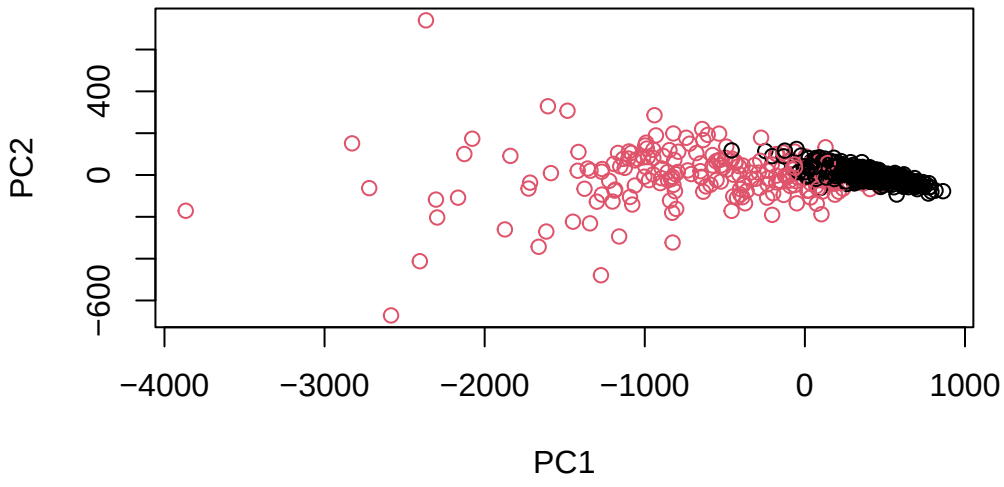
```
table(grps, diagnosis)
```

```
    diagnosis
grps   B   M
   1   0  86
   2 357 126
```

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




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



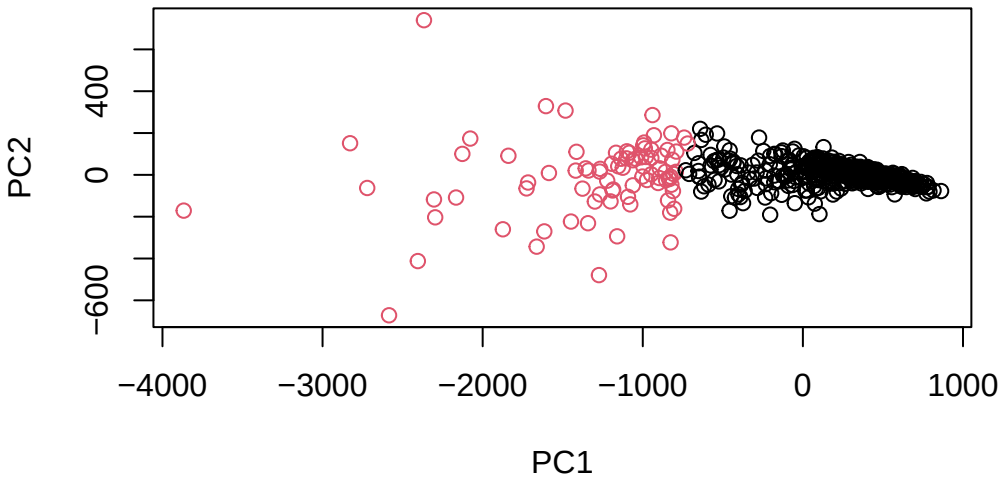
```
g <- as.factor(grps)
levels(g)
```

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

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

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

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




```
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?

```
compare <- table(wisc.pr.hclust.clusters, as.numeric(factor(diagnosis)))
print(compare)
```

```
wisc.pr.hclust.clusters   1   2
                      1   0  86
                      2 357 126
```

> Q17. 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?

The `wisc.km$cluster` is better in my opinion because the `wisc.hclust.clusters` breaks it down into more groups but with less meaning, the numbers for each group are very insignificantly changed from the first.

6. Sensitivity/Specificity

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

```
#K-Means

#Sensitivity:
  37 / (37 + 175)
```

```
[1] 0.1745283
```

```
#Specificity:
  14 / (14 + 343)
```

```
[1] 0.03921569
```

```
#Hierarchical

# Sensitivity:
  5 / (5 + 165)
```

```
[1] 0.02941176
```

```
# Specificity:
12 / (12 + 343)
```

```
[1] 0.03380282
```

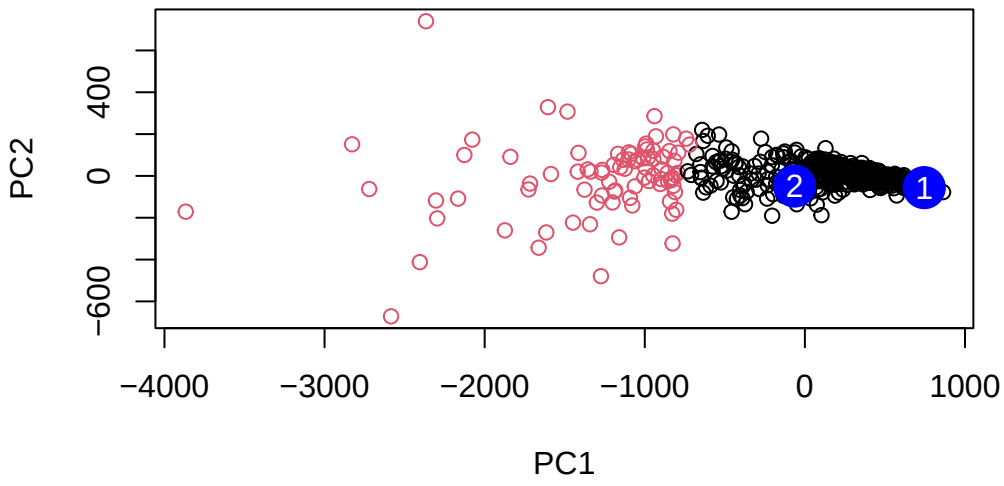
K-means for both.

7. 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,] 745.60081 -56.16454 -21.15609 -3.330663  9.355518  2.317462 -1.147268
[2,] -64.40839 -48.46996 -15.93413 12.089591 -4.636008 -1.045210 -0.295228
           PC8         PC9        PC10       PC11        PC12         PC13
[1,] -0.7644759  0.11704582  0.06401851 0.1191717 -0.05611973 -0.040020096
[2,] -0.7454142 -0.09167106 -0.76173550 0.3206674  0.02602751  0.005023528
           PC14         PC15        PC16        PC17        PC18         PC19
[1,]  0.01354667 -0.018755904 -0.01050870 -0.01183961 0.020946097  0.030567858
[2,] -0.11943490  0.008958015  0.03391077 -0.02468455 0.008002482 -0.006896744
            PC20         PC21        PC22         PC23         PC24
[1,] -0.007960122 -0.003773165 0.018561168 0.0001875602 -0.005463212
[2,]  0.007001178 -0.022182056 0.008725155 0.0075849336  0.004619616
            PC25        PC26         PC27         PC28         PC29
[1,] -0.005992320 0.005357732 4.550233e-05  0.003252776 0.0012510265
[2,]  0.002804663 0.003229335 1.977351e-03 -0.002261832 0.0009130702
             PC30
[1,] -0.0009794321
[2,] -0.0009078383
```

```
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?

The ones labeled in red.

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.5

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```

```
other attached packages:
[1] ggplot2_3.4.4

loaded via a namespace (and not attached):
 [1] vctrs_0.6.5      cli_3.6.2        knitr_1.45       rlang_1.1.3
 [5] xfun_0.41        generics_0.1.3   jsonlite_1.8.8   labeling_0.4.3
 [9] glue_1.7.0       colorspace_2.1-0 htmltools_0.5.7  scales_1.3.0
[13] fansi_1.0.6      rmarkdown_2.25   grid_4.3.2       evaluate_0.23
[17] munsell_0.5.0    tibble_3.2.1     fastmap_1.1.1    yaml_2.3.8
[21] lifecycle_1.0.4  compiler_4.3.2   dplyr_1.1.4      pkgconfig_2.0.3
[25] farver_2.1.1     digest_0.6.34    R6_2.5.1         tidyselect_1.2.0
[29] utf8_1.2.4       pillar_1.9.0     magrittr_2.0.3   withr_3.0.0
[33] tools_4.3.2      gtable_0.3.4
```