Class 008

R.D. Carias (A18573289)

Table of contents

Background

Today, we are using all the R techniques we've reviewed thus far, including the machine learning methods of clustering and PCA - to analyze a breast cancer data set that came from the university of Wisconsin Medical Center.

```r
wisc.df <- read.csv("WisconsinCancer.csv", row.names = 1)
head(wisc.df, 3)
```

```
         diagnosis radius_mean texture_mean perimeter_mean area_mean
842302           M       17.99        10.38          122.8      1001
842517           M       20.57        17.77          132.9      1326
84300903         M       19.69        21.25          130.0      1203
         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
```

```
         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
         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
         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
         perimeter_worst area_worst smoothness_worst compactness_worst
842302             184.6       2019           0.1622            0.6656
842517             158.8       1956           0.1238            0.1866
84300903           152.5       1709           0.1444            0.4245
         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
         fractal_dimension_worst
842302                   0.11890
842517                   0.08902
84300903                 0.08758
```

Q1. How many observations are in this data set?: 569 observations for these data.

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

```
[1] 569
```

569 observations are present within this data frame.

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

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

```
[1] 212
```

There are 212 observation with malignant diagnosis.

> Q3. How many variables/features in the data are suffixed with _mean?: There are 10 variables with the "mean" siffix.

```
length(grep("mean", colnames(wisc.df),))
```

```
[1] 10
```

Now, we need to remove the `diagnosis` column before we do an further analysis of these data. We don't want to pass the diagnosis data to the PCA. We will save it as new vector that we can use later to compare our findings to that of the researchers.

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

diagnosis <- wisc.df$diagnosis
```

Principal Component Analysis (PCA)

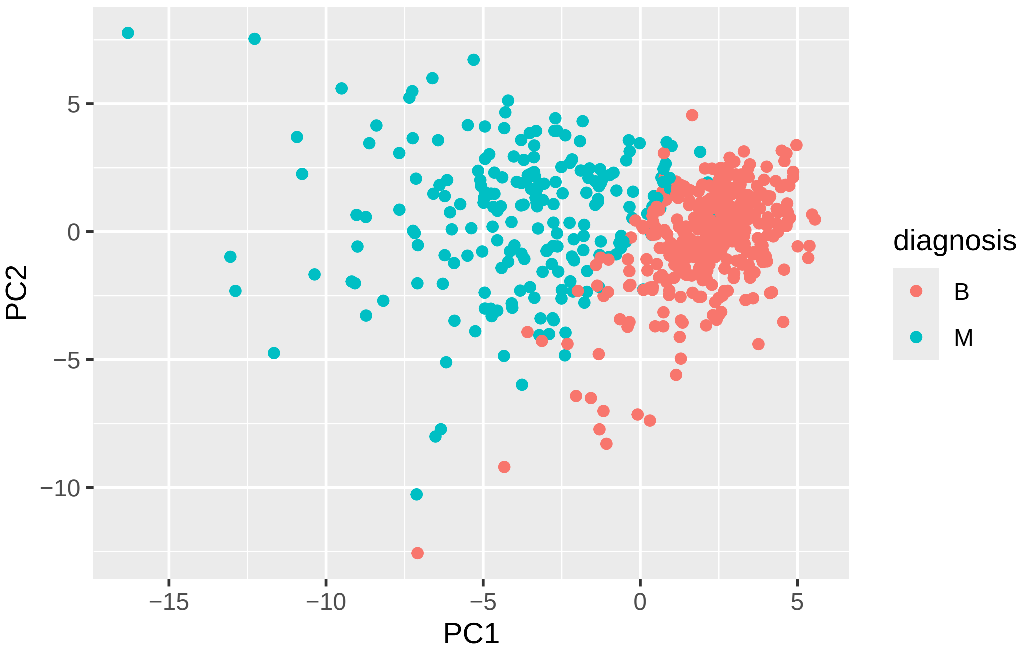
The main function in base R is called `prcomp()`. Scaling effects the results but center does not. We still use the optional argument `scale=TRUE` here as the data columns/features/dimensions are on very different scales in the OG data set.

```
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
```

```
library(ggplot2)
ggplot(wisc.pr$x) + aes(PC1,PC2, col=diagnosis) + geom_point()
```



Remarks: This plot shows PC1 versus PC2. **Each point** represents a single biopsy sample summarized across all measured features. Samples that appear close together in this plot have **similar** overall multivariate characteristics.

PCA is a dimensionality-reduction technique that **transforms a dataset with many correlated variables into a smaller set of new variables** called `principal components`. These components are linear combinations of the original features and are constructed to capture as much variation in the data as possible.

Before performing PCA, we often need to normalize (scale) the variables. **This is important when features are measured in different units or ranges.** For example, if one variable is measured in miles and another in centimeters, the variable with the larger numerical scale would dominate the variance and heavily influence PC1 purely due to unit size, not biological importance.

PC1 represents the direction in feature space along which the data show the greatest overall variance. **PC2 represents the direction of the next greatest variance, constrained to be orthogonal (perpendicular) to PC1.** These components are determined mathematically from the covariance (or correlation) structure of the entire dataset, not by visually defined clusters. However, clusters may become visible in the PCA plot because samples with similar feature patterns project to similar locations in this reduced space.

`wisc.pr$x` does not mean “column x”; it accesses the scores matrix stored inside the `prcomp` PCA object. That matrix contains the projected values of each observation on PC1, PC2, etc. Passing it to ggplot() means you are giving ggplot the PCA scores as the dataset, not selecting a single column.

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

Proportion of Variance captured by our principal component analysis (PCA1) is 0.4427 or 44.27%. This can be determined by using the `summary()` of our ‘pr’ results.

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

We require 3 PCAs to describe at least 70% of the original variance. Results: Cumulative Proportion 0.72636

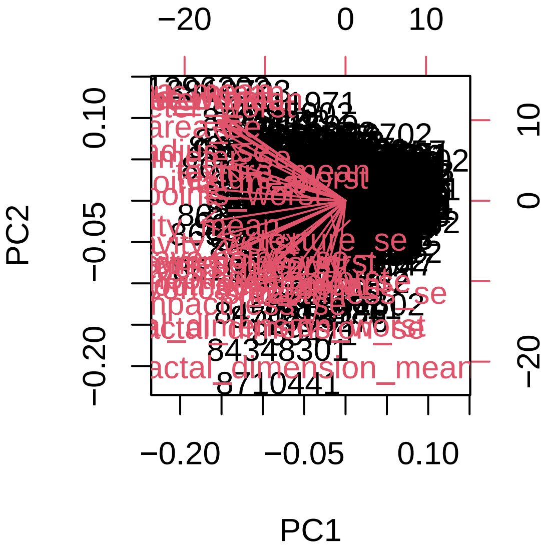
> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?:

We require 7 PCAs to describe at least 90% of the original variance. Results: Cumulative Proportion 0.91010

Interpreting Results of PCA

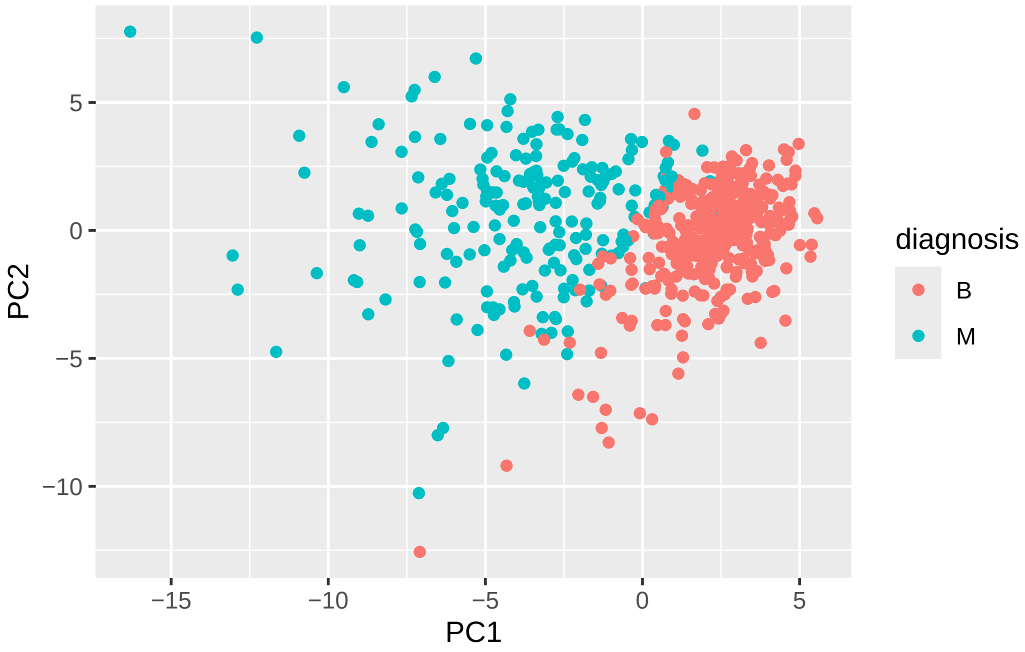
> Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?:

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



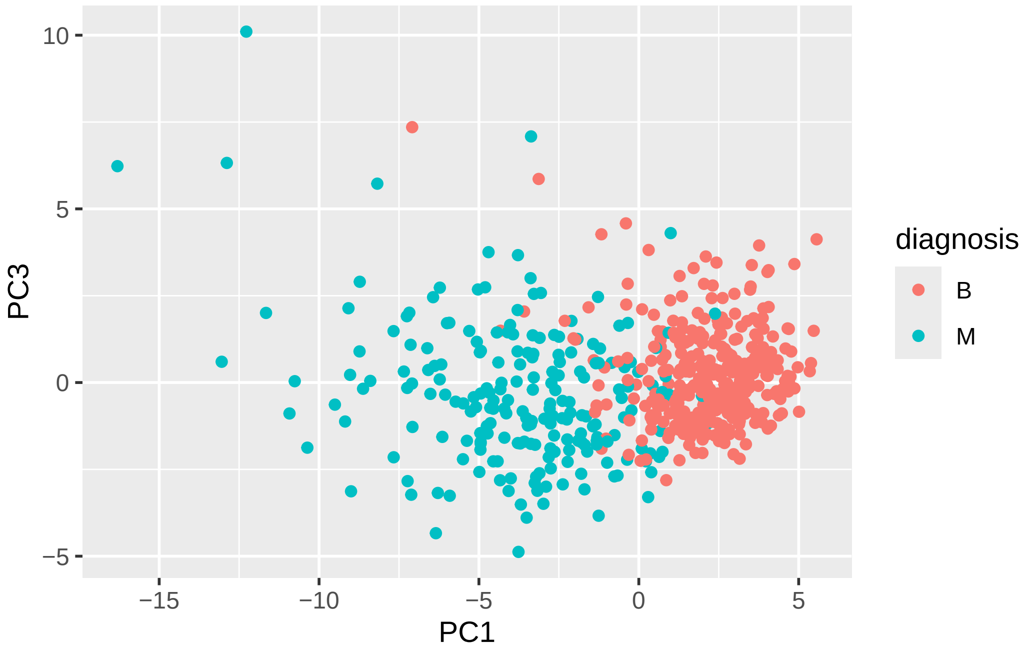
If we use `biplot()`, it is difficult to understand due to over crowding of information presented on the entire `wisc.pr`. This biplot is over saturated with information, text, etc.

```
# Scatter plot observations by components 1 and 2
library(ggplot2)

ggplot(wisc.pr$x) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```



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

```
library(ggplot2)
ggplot(wisc.pr$x) + aes(PC1,PC3, col=diagnosis) + geom_point()
```

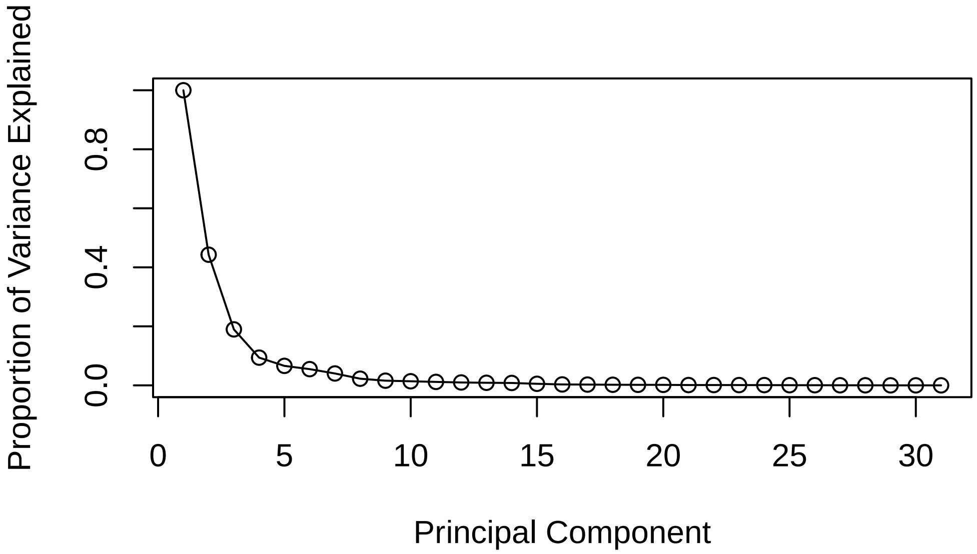


This plot shows PC1 versus PC3. Because principal components are ordered by the amount of variance they explain, PC3 captures substantially less variance than PC1. As a result, separation between clusters is less pronounced. This is expected.

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

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

```
# Variance explained by each principal component: pve
pve <- (wisc.pr$sdev^2) / sum(wisc.pr$sdev^2)



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

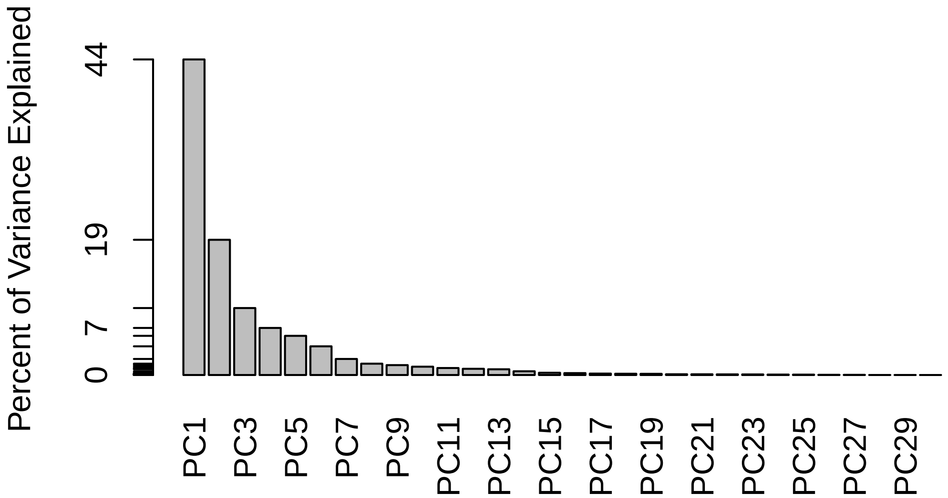


```
# Statistical view:
# For random variables X and Y:
# Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)
# PCA constructs principal components that are uncorrelated:
# Cov(PC_i, PC_j) = 0 for i  j
# Therefore, all covariance terms vanish and total variance
# is simply the sum of individual PC variances (Σ  _i)
```

Remarks:

The code takes the PCA output and computes how much variance each principal component explains by squaring the PCA standard deviations and dividing by the total variance. It then creates a scree-style plot where the y-axis shows the proportion of variance explained and the x-axis indexes the components. The leading 1 in c(1, pve) is added only to anchor the plot at total variance, not to represent an actual principal component

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



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? This tells us how much this original feature contributes to the first PC. Are there any features with larger contributions than this one?:

```
wisc.pr$rotation["concave.points_mean", 1]
```

```
[1] -0.2608538
```

```
sort(abs(wisc.pr$rotation[,1]), decreasing = TRUE)
```

```
    concave.points_mean          concavity_mean    concave.points_worst
             0.26085376              0.25840048              0.25088597
       compactness_mean         perimeter_worst         concavity_worst
             0.23928535              0.23663968              0.22876753
           radius_worst          perimeter_mean              area_worst
             0.22799663              0.22753729              0.22487053
              area_mean             radius_mean            perimeter_se
             0.22099499              0.21890244              0.21132592
      compactness_worst               radius_se                 area_se
             0.21009588              0.20597878              0.20286964
      concave.points_se          compactness_se            concavity_se
```

```
             0.18341740              0.17039345              0.15358979 
        smoothness_mean           symmetry_mean fractal_dimension_worst 
             0.14258969              0.13816696              0.13178394 
       smoothness_worst          symmetry_worst           texture_worst 
             0.12795256              0.12290456              0.10446933 
           texture_mean    fractal_dimension_se  fractal_dimension_mean 
             0.10372458              0.10256832              0.06436335 
            symmetry_se              texture_se           smoothness_se 
             0.04249842              0.01742803              0.01453145 
```

Numerically, no single feature contributes meaningfully more to PC1 than concave.points_mean, as the top loadings differ only marginally in magnitude.

Biologically, the top few features, concave points, concavity, and related shape measures, capture the same tumor morphology and should be interpreted as comparably important drivers of the first principal component.

Remarks:

Loadings `wisc.pr$rotation` represent the weights of each original feature in a principal component. By sorting the absolute PC1 loadings in decreasing order, we identify which features contribute most strongly to PC1. Comparing these values shows whether any features contribute more to PC1 than `concave.points_mean`.

First, we explicitly extract the PC1 loading for concave.points_mean to see how strongly that single feature contributes. When all PC1 loadings are then sorted by absolute value, it appears first because it truly has the largest magnitude, not because it was selectively prioritized.

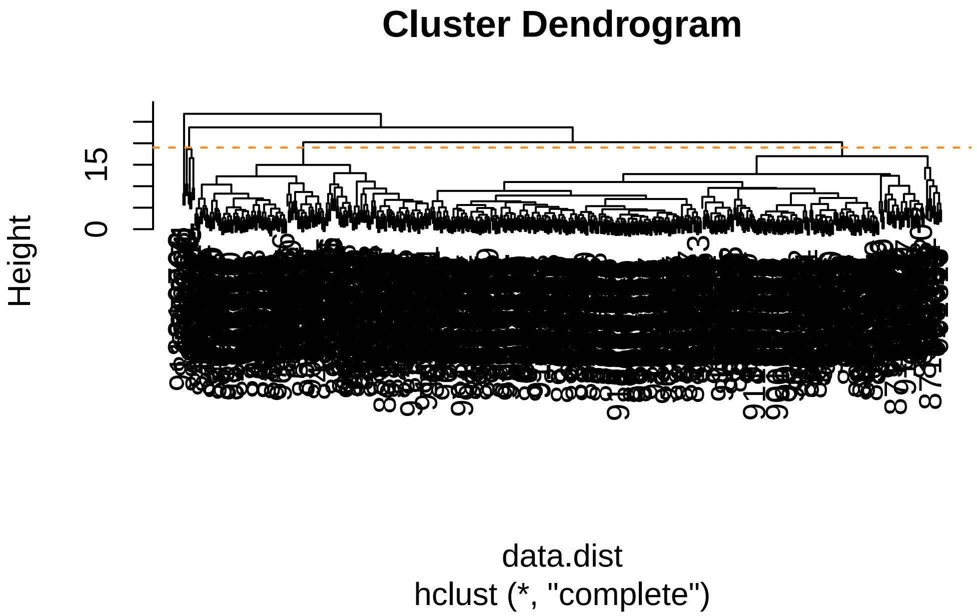
Hierarchical clustering

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

data.dist <- dist(data.scaled)

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

Results of hierarchical clustering

> Q10. 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="darkorange", lty=2)
```



The height at which we get 4 cluster is 19.

Selecting number of clusters

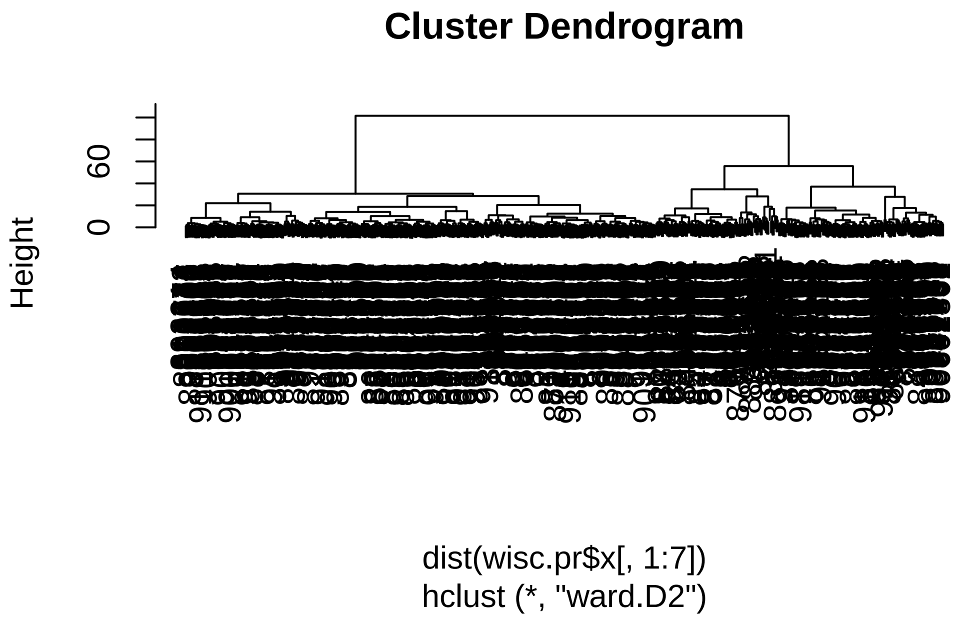
```
wisc.hclust.clusters <- cutree(wisc.hclust, k = 4)

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

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

Interpretation: Cluster 1 is enriched for malignant (M) samples, while cluster 3 is enriched for benign (B) samples. This indicates that the unsupervised hierarchical clustering captures biologically meaningful separation between the two diagnoses.

Using Different Method

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

# plot it
plot(wisc.pr.hclust)
```



```
wisc.pr.hclust.cutree <- cutree(wisc.pr.hclust, k = 3)
table(wisc.pr.hclust.cutree, diagnosis)
```

```
                     diagnosis
wisc.pr.hclust.cutree   B   M
                    1   2 122
                    2  26  66
                    3 329  24
```

```
wisc.pr.hclust.cutree <- cutree(wisc.pr.hclust, k = 4)
table(wisc.pr.hclust.cutree, diagnosis)
```

```
                     diagnosis
wisc.pr.hclust.cutree   B   M
                    1   0  45
                    2   2  77
                    3  26  66
                    4 329  24
```

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

As Ward.D2 minimizes within-cluster variance and operates in a bottom-up manner, it provides a strong starting point for our analysis. In this dataset, Ward.D2 defined a 100% pure malignant cluster, which was improved separation compared to the "complete linkage" method. While isolating more ambiguous cases into larger clusters than that of our complete linkage results. For this reason, comparison with single, complete, and average linkage remains important.

Combining Methods

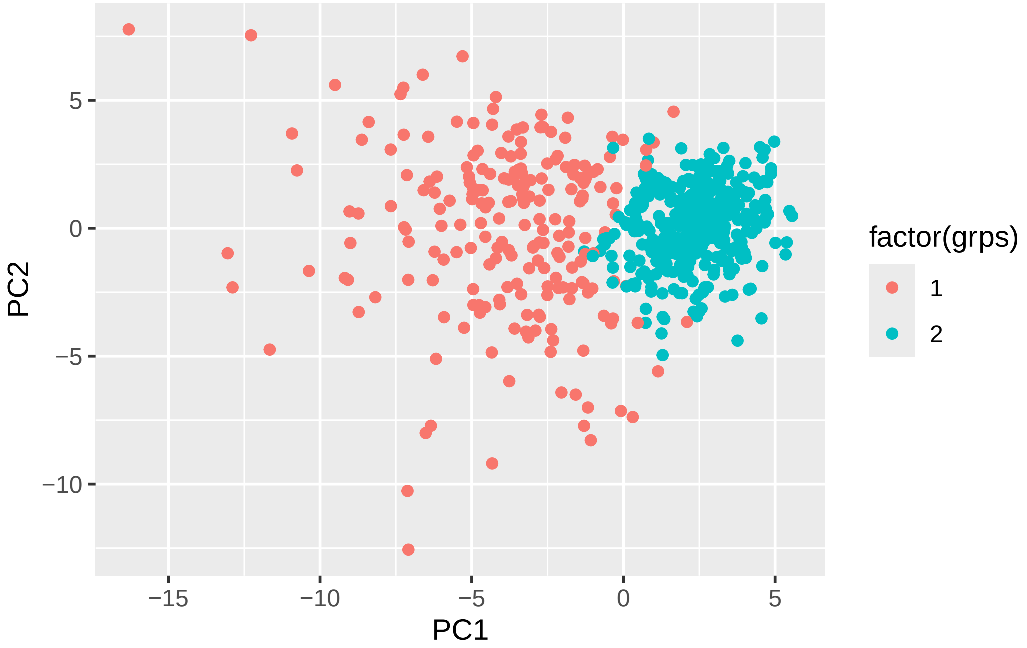
```r
# Cut into 2 clusters
grps<- cutree(wisc.pr.hclust, k = 2)

# Check cluster sizes
table(grps)
```

```
grps
  1   2 
216 353 
```

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

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

```r
ggplot(as.data.frame(wisc.pr$x)) +
  aes(PC1, PC2, col = factor(grps)) +
  geom_point()
```



```r
wisc.pr.hclust <- hclust(dist(wisc.pr$x[, 1:7]), method = "ward.D2")
wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k = 2)
```

> Q13. How well does the newly created hclust model with two clusters separate out the two "M" and "B" diagnoses?

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

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

Our results produce one cluster that is strongly Malignant and the other is Strongly Benign. Same as last time.

> Q14. How well do the hierarchical clustering models you created in the previous sections (i.e. without first doing PCA) do in terms of separating the diagnoses? Again, use the table() function to compare the output of each model (wisc.hclust.clusters and wisc.pr.hclust.clusters) with the vector containing the actual diagnoses.

```
# Compare clustering WITHOUT PCA (complete linkage on original data)
table(wisc.hclust.clusters, diagnosis)
```

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

```
# Compare clustering WITH PCA (Ward.D2 on first 7 PCs)
table(wisc.pr.hclust.clusters, diagnosis)
```

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

Our complete linkage clusters produced 4 clusters, of mixed results. When we cluster and include the first 7 PCA, encompassing 90% of the variance, we are able to obtain two clusters where either malignant or benign diagnosis dominate.

Sensitivity

Sensitivity Equation : TP/(TP+FN)

```
179/(179+33)
```

```
[1] 0.8443396
```

Specificity: TN/(TN+FP)

```
333/(333+24)
```

```
[1] 0.9327731
```

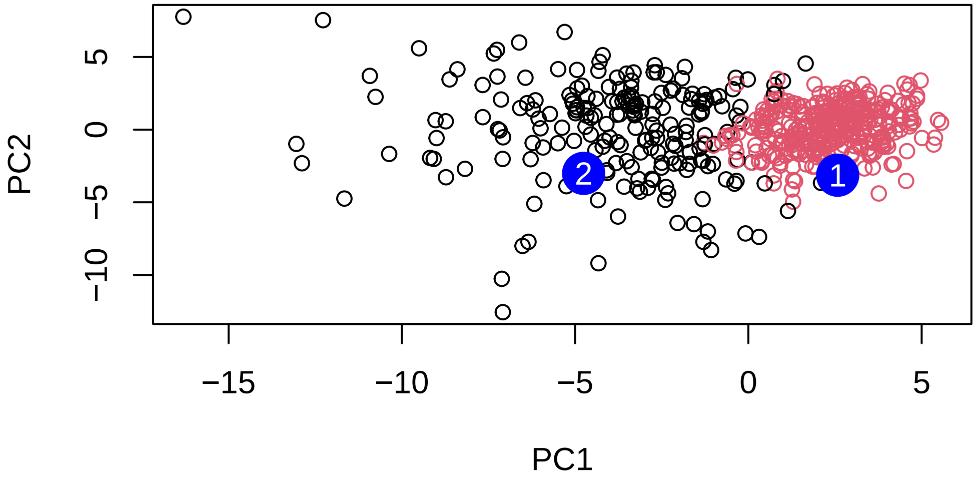
Prediction

We can use our PCA model of prediction of new unseen cases:

This is a NEW data set from **UMich**, NOT the **UWisc** data we have used above.

```r
#url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
```

```r
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")
```



> Q16. Which of these new patients should we prioritize for follow up based on your results?:

When these new data are projected onto our PCA analysis we would advise that patients cluster 1 be prioritized for follow up. New patients assigned to cluster 1 should be prioritized for follow-up because this cluster is highly "enriched" for malignant diagnoses in the training data.

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

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