Class08

Rachel, PID:A16037641

The goal of today's mini project is to explore a complete analysis using the unsupervies learning techniques covered in the last class. We will extend what we learned by combining PCA asa. preprocessing step to clustering using data that consist of measurements of cell nuclei of human bresat masses. The data itself comes from the Wisconsin Breast Cancer

Save your input data file into your Project directory

fna.data <- "WisconsinCancer.csv"

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

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

wisc.df <- ____(fna.data, row.names=1)

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

Make sure we do not include sample ID or the diagnosis column in further analysis

```
diagnois <- as.factor(wisc.df$diagnosis)
wisc.data <- wisc.df[,-1]
dim(wisc.data)
```

```
[1] 569  30
```

```
head(wisc.data)
```

```
         radius_mean texture_mean perimeter_mean area_mean smoothness_mean
842302         17.99        10.38         122.80    1001.0         0.11840
842517         20.57        17.77         132.90    1326.0         0.08474
84300903       19.69        21.25         130.00    1203.0         0.10960
84348301       11.42        20.38          77.58     386.1         0.14250
84358402       20.29        14.34         135.10    1297.0         0.10030
843786         12.45        15.70          82.57     477.1         0.12780
         compactness_mean concavity_mean concave.points_mean symmetry_mean
842302            0.27760         0.3001             0.14710        0.2419
842517            0.07864         0.0869             0.07017        0.1812
84300903          0.15990         0.1974             0.12790        0.2069
84348301          0.28390         0.2414             0.10520        0.2597
84358402          0.13280         0.1980             0.10430        0.1809
843786            0.17000         0.1578             0.08089        0.2087
         fractal_dimension_mean radius_se texture_se perimeter_se area_se
842302                  0.07871    1.0950     0.9053        8.589  153.40
842517                  0.05667    0.5435     0.7339        3.398   74.08
84300903                0.05999    0.7456     0.7869        4.585   94.03
84348301                0.09744    0.4956     1.1560        3.445   27.23
84358402                0.05883    0.7572     0.7813        5.438   94.44
843786                  0.07613    0.3345     0.8902        2.217   27.19
         smoothness_se compactness_se concavity_se concave.points_se
842302        0.006399        0.04904      0.05373           0.01587
842517        0.005225        0.01308      0.01860           0.01340
84300903      0.006150        0.04006      0.03832           0.02058
84348301      0.009110        0.07458      0.05661           0.01867
84358402      0.011490        0.02461      0.05688           0.01885
843786        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
```

Q.1 How many observations are in this dataset?

If we use the 'nrow(wisc.data) we get an answer of 569 and 30, so their are 569 observations in this dataset

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

```
[1] 569
```

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

There are 212 malignant diagnosis

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

```
[1] 212
```

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

```
  B   M
357 212
```

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

There are 10 variables suffixed with _mean.

```
length(grep("_mean", colnames(wisc.data)))
```

```
[1] 10
```

Principal Component Analysis

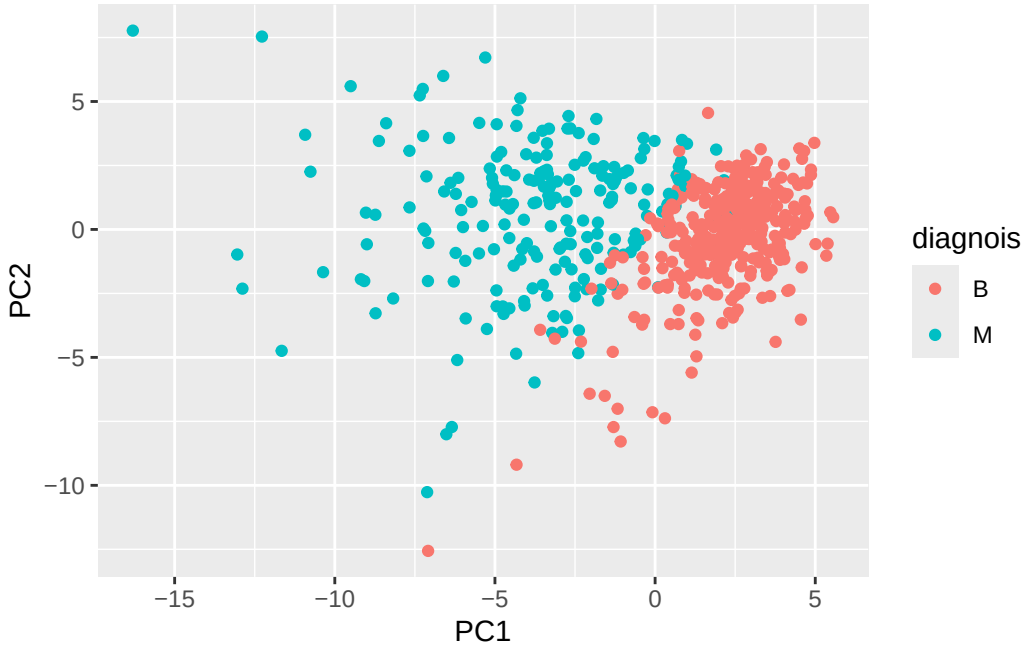
The main function in base R for PCA is called `prcomp()`. An optional argument `scale` should nearly alwyas be switched to `scale=TRUE` for this function.

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

Let's make our main result figure - the "PC Plot" or "Score Plot", or "Ordienation plot"...

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



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

44 percent of the original variance is captured by the first principal component.

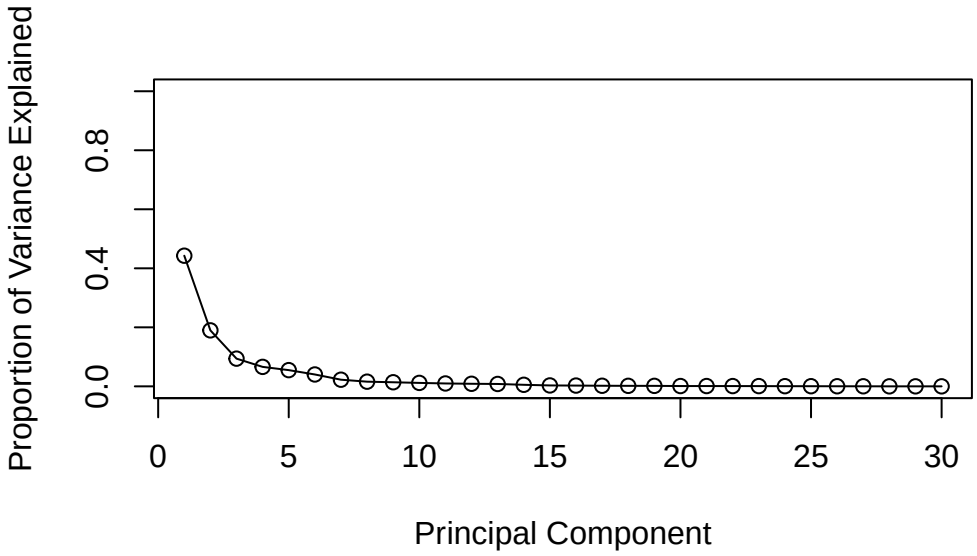
```
pr.var <- wisc.pr$sdev^2
pve <- pr.var / sum(pr.var)
head(pr.var)
```

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

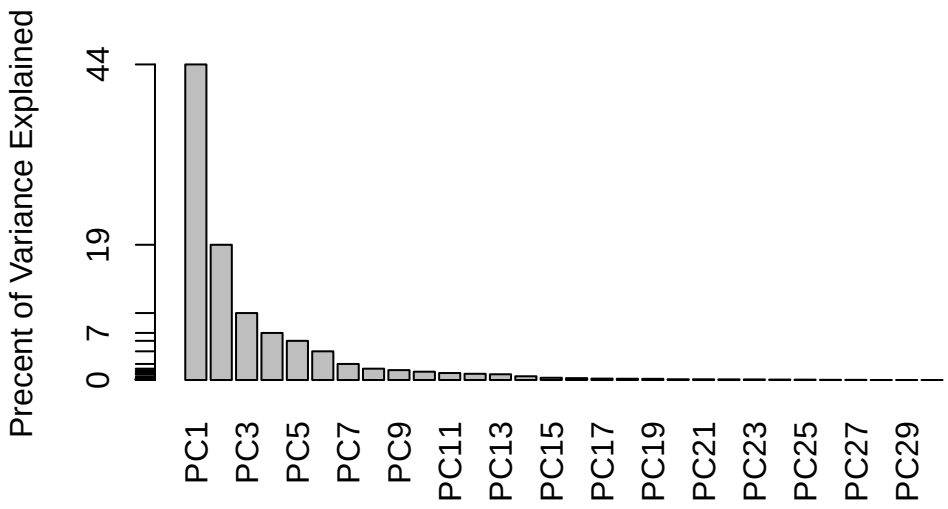
```
head(pve*100)
```

```
[1] 44.272026 18.971182  9.393163  6.602135  5.495768  4.024522
```

```
# 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 )
```

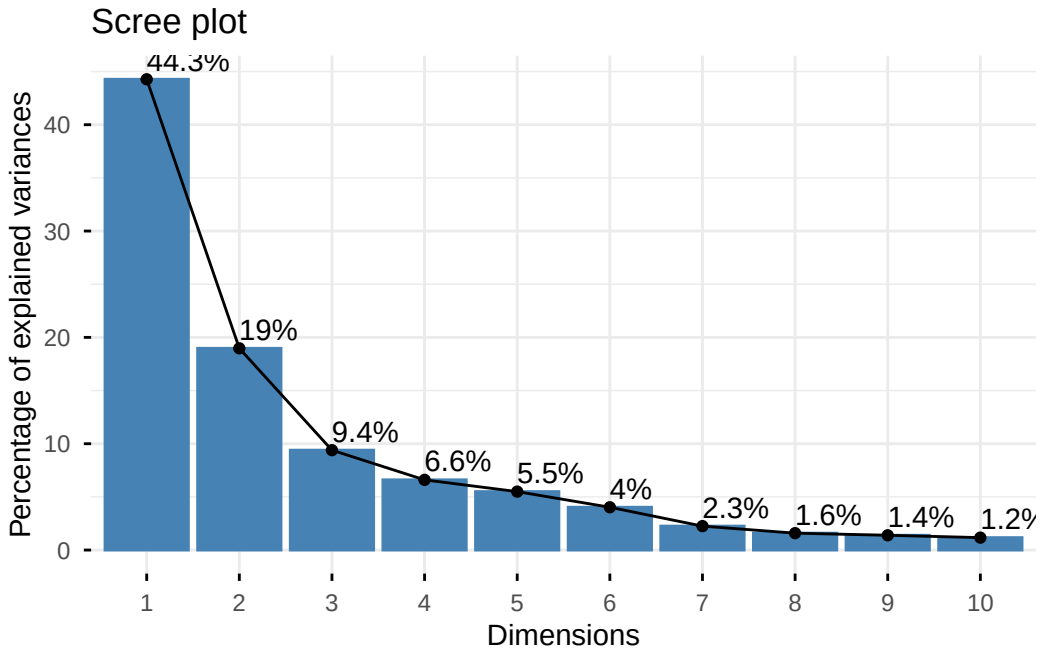



```
## ggplot based graph
library(factoextra)
```

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

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

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```



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

Three are required

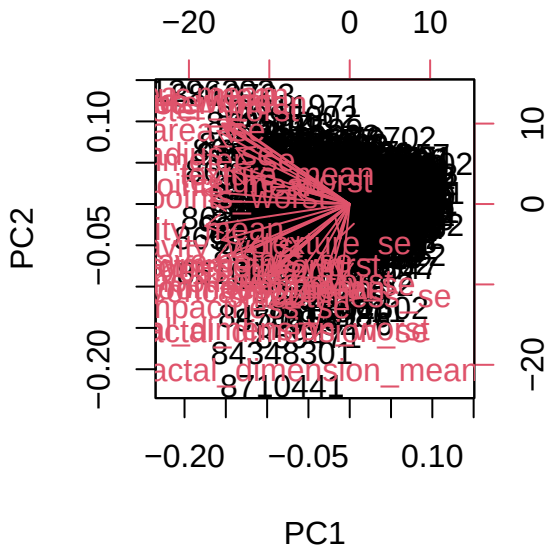
Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

There are seven required

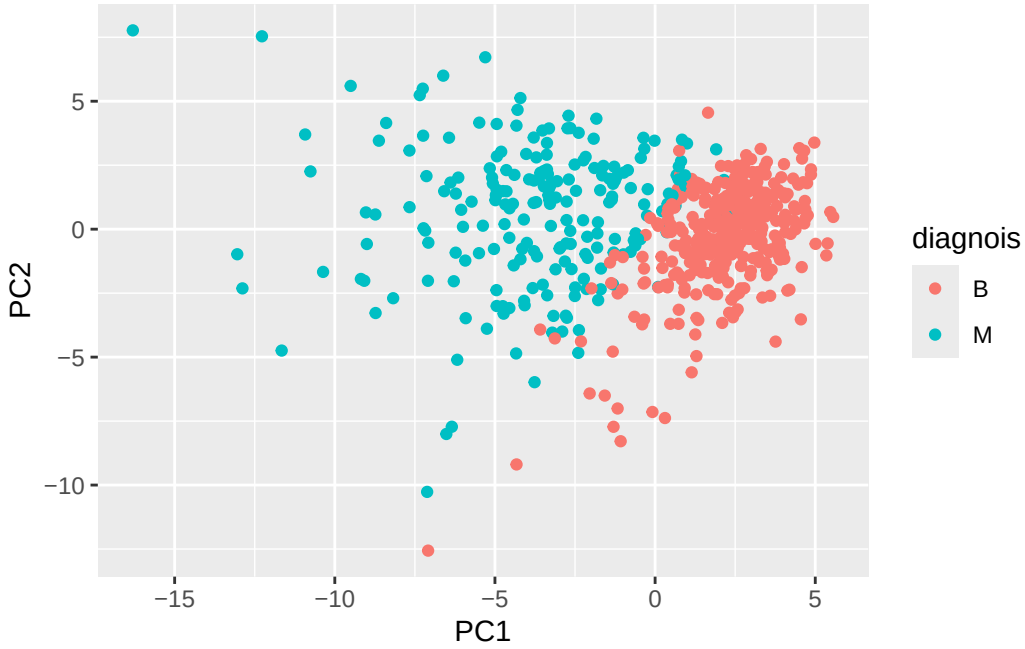
Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

This plot is incredibly hard to read, it seems to be labeling every point with patient identifyers

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



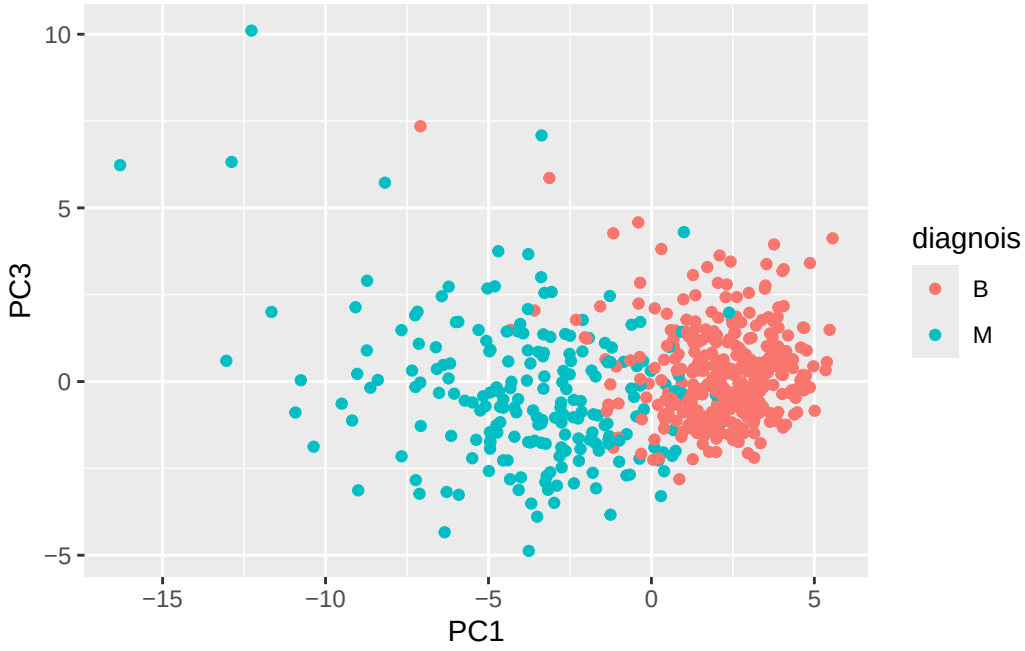

```
ggplot(wisc.pr$x) +
  aes(PC1, PC2, col=diagnois) +
  geom_point()
```



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

These plots show that component 1 is capturing a seperation of malignant (red) from benign (black).

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



Calculate the variance of each principal component by squaring the sdev component of wisc.pr (i.e. wisc.pr$sdev^2).

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.

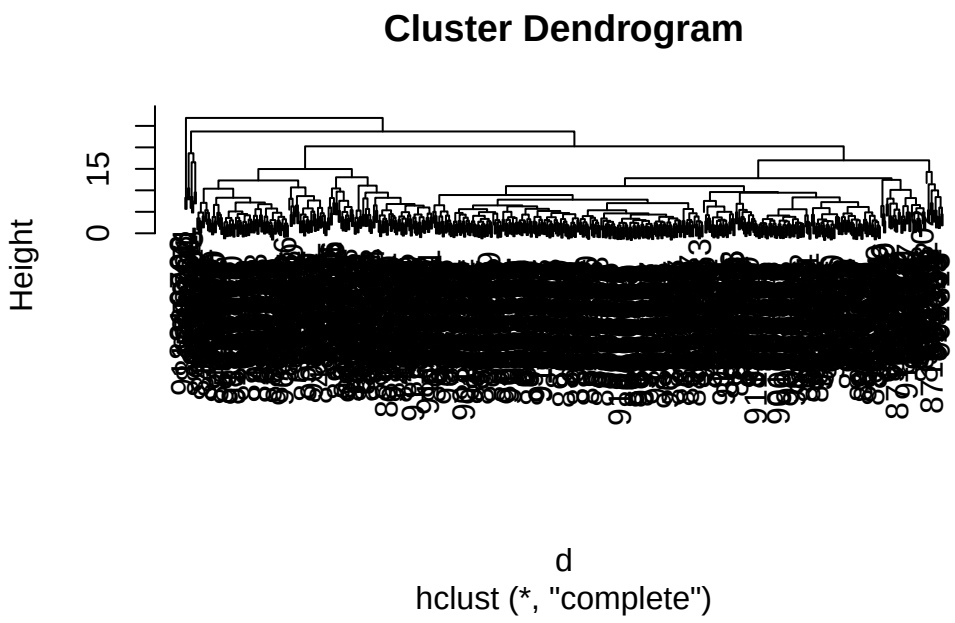
```
wisc.pr$rotation["concave.points_mean", 1]
```

```
[1] -0.2608538
```

The value -0.2608538 is the loading for concave.points_mean in the first principal component. It shows how much this feature contributes to PC1. A larger absolute value means a stronger influence; the negative sign indicates the direction of the relationship with PC1

Hierarchal Clustering

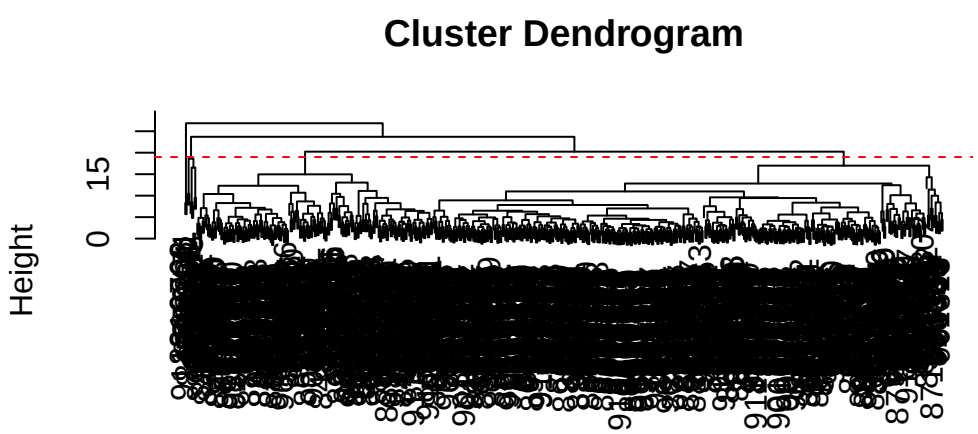
```
d <- dist(scale(wisc.data))
h <- hclust(d)
plot(h)
```



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

Around the height of 19, the clustering has 4 clusters

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

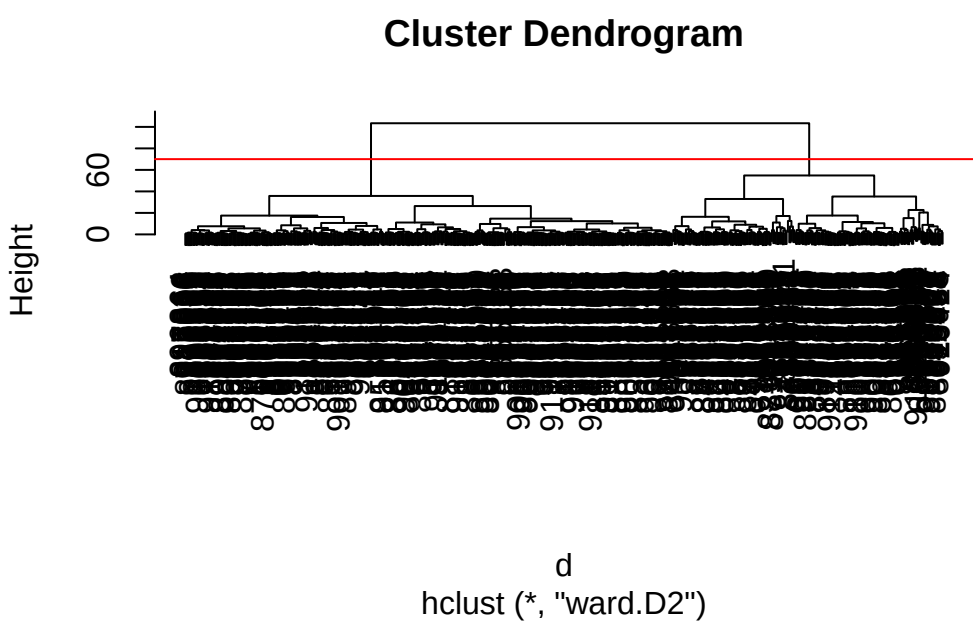


Combining PCA and clustering

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

I like using `ward.D2`, as seen below the method gives a cleaner dendrogram.

```
d <- dist(wisc.pr$x[,1:3])
wisc.pr.hclust <- hclust(d, method="ward.D2")
plot(wisc.pr.hclust)
abline(h=70, col="red")
```



Get my cluster membership vector

```
groups <-cutree(wisc.pr.hclust, h=70)
table(groups)
```

```
groups
  1   2
203 366
```

```
table(groups, diagnois)
```

```
      diagnois
groups   B   M
     1  24 179
     2 333  33
```

```
wisc.hclust.clusters <- cutree(wisc.pr.hclust, h=30)
```

```
table(diagnois)
```

```
diagnois
  B   M
357 212
```

Q13. How well does the newly created model with four clusters separate out the two diagnoses?

The newly created model seems to create four well separate clusters Make a "cross-table"

```
table(groups, diagnois)
```

```
      diagnois
groups   B   M
     1  24 179
     2 333  33
```

Q14. How well do the 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.

The previously made hierarchical clustering models do not seem to separate the diagnoses out as well as the PCA models we made.

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

```
                    diagnois
wisc.hclust.clusters   B   M
                   1   0  33
                   2   0  78
                   3  13   5
                   4  11  63
                   5 184  32
                   6 149   1
```

Q15. OPTIONAL: Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

Our PCA and clustering combination seemed to yield the best results. To find malignant cases, TP: 179 FP: 24 TN: 333 FN: 33

Sensitivity: TP/(TP+FN): 179/(179+33) = 0.84 Specificity: TN/(TN+FN): 333/(333+33) = 0.91

Prediction

We will use the predict() function that will take our PCA model from before and new cancer cell data and project that data onto our PCA space.

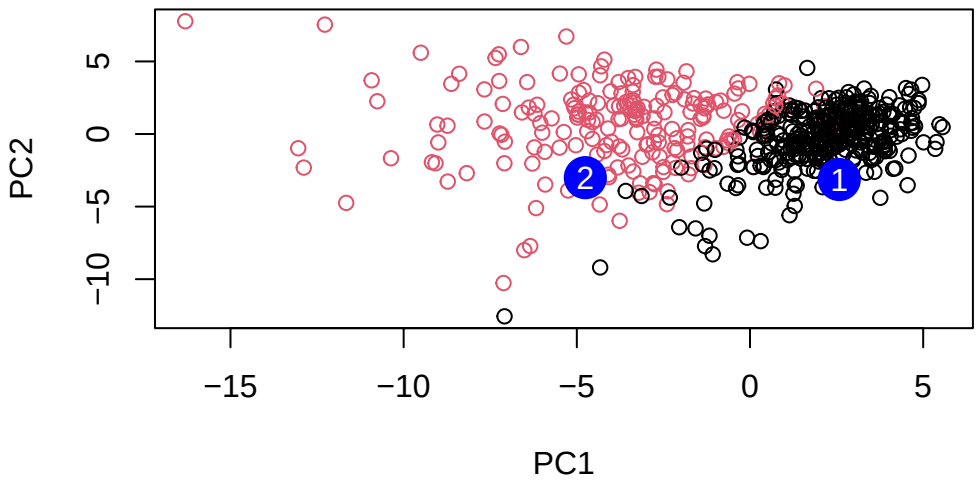
```
#url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
npc
```

```
          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
```

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

Based on these results we should prioritize patient two for a follow up.

```
plot(wisc.pr$x[,1:2], col=diagnois)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```


PC2
5
0
−5
−10
−15
−10
−5
0
5
PC1
2
1