breast_cancer_mini_project

Kelly_F

10/27/2021

Exploratory Analysis

```
# Import Raw Data
wisc.df <- read.csv("./WisconsinCancer.csv",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  X
## 842302                   0.11890 NA
## 842517                   0.08902 NA
## 84300903                 0.08758 NA
## 84348301                 0.17300 NA
## 84358402                 0.07678 NA
## 843786                   0.12440 NA
```

```
# Remove diagnosis column
wisc.data <- wisc.df[,-1]
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  X
## 842302                   0.11890 NA
## 842517                   0.08902 NA
## 84300903                 0.08758 NA
## 84348301                 0.17300 NA
## 84358402                 0.07678 NA
## 843786                   0.12440 NA
```

```
# Remove "X" column of NA values
dim(wisc.data)
```

```
## [1] 569  31
```

```
wisc.data <- wisc.data[, 1:30]

# Create diagnosis vector & factor data
diagnosis <- as.factor(wisc.df$diagnosis)
str(diagnosis)
```

```
##  Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
```

Q1. How many observations are in this dataset?

569 observations

```
dim(wisc.df)
```

```
## [1] 569  32
```

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

212 are malignant

```
#Option 1
length(grep("M", diagnosis))
```

```
## [1] 212
```

```
#Option 2
dim(subset(wisc.df, diagnosis=="M"))
```

```
## [1] 212  32
```

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

10

```
colnames(wisc.df)
```

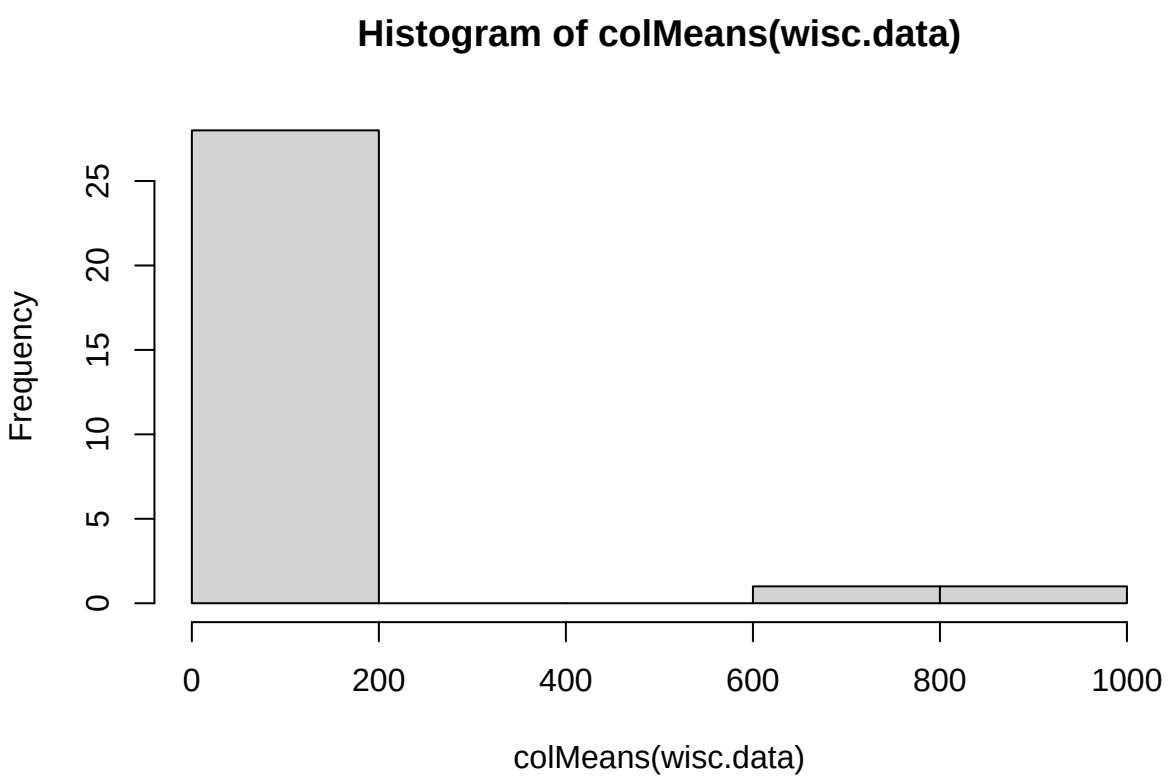
```
##  [1] "diagnosis"               "radius_mean"
##  [3] "texture_mean"            "perimeter_mean"
##  [5] "area_mean"               "smoothness_mean"
##  [7] "compactness_mean"        "concavity_mean"
##  [9] "concave.points_mean"     "symmetry_mean"
## [11] "fractal_dimension_mean"  "radius_se"
## [13] "texture_se"              "perimeter_se"
## [15] "area_se"                 "smoothness_se"
## [17] "compactness_se"          "concavity_se"
## [19] "concave.points_se"       "symmetry_se"
## [21] "fractal_dimension_se"    "radius_worst"
## [23] "texture_worst"           "perimeter_worst"
## [25] "area_worst"              "smoothness_worst"
## [27] "compactness_worst"       "concavity_worst"
## [29] "concave.points_worst"    "symmetry_worst"
## [31] "fractal_dimension_worst" "X"
```

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

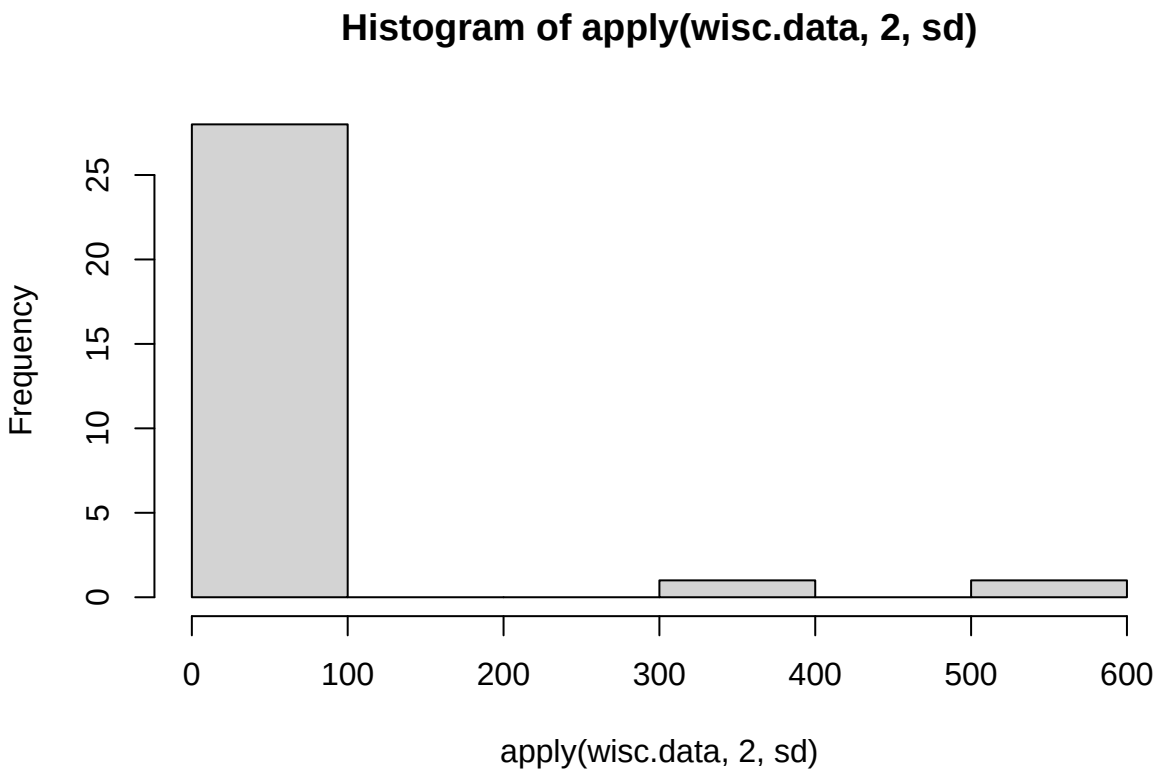
```
## [1] 10
```

Performing PCA

```
# Check column means and standard deviations
hist(colMeans(wisc.data))
```




```
hist(apply(wisc.data,2,sd))
```




```
# Perform PCA on wisc.data and transform data due to large variation
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)?

44.27%

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

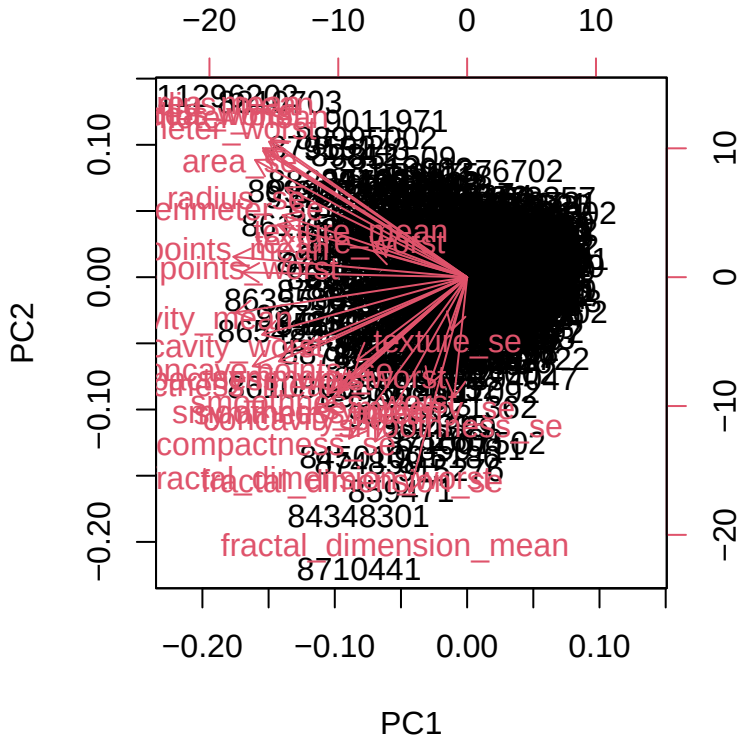
Three

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

Seven

Interpreting PCA Results

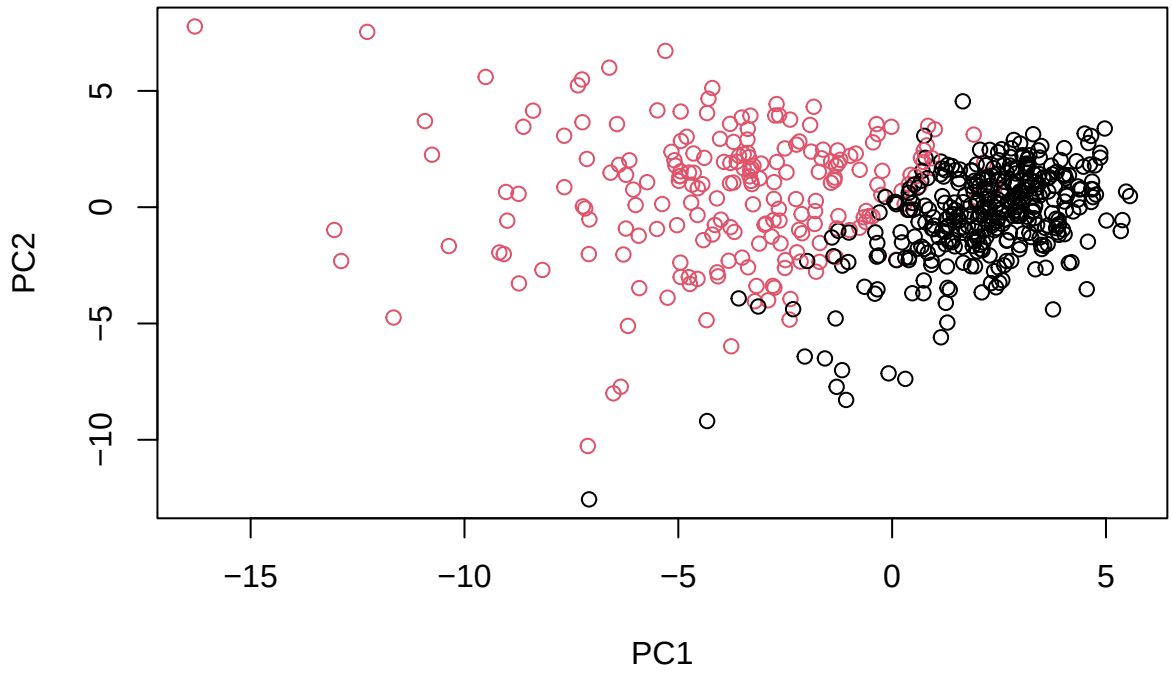
```
# Generate biplot
biplot(wisc.pr)
```



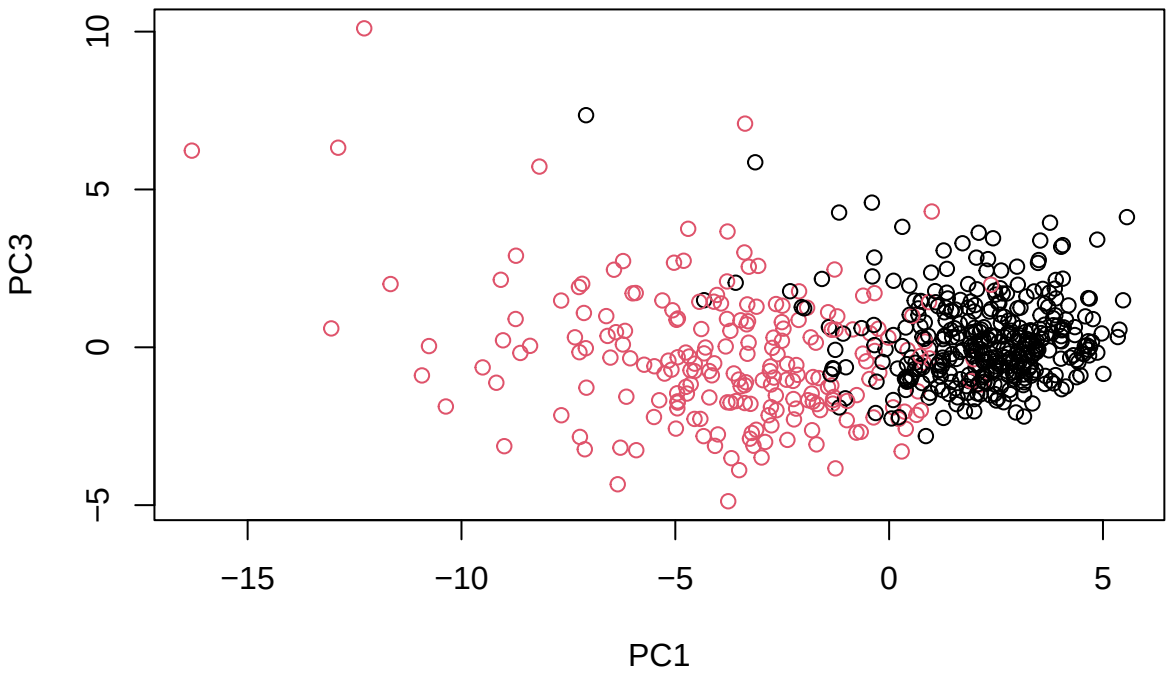
> Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

It is difficult to understand. There are many overlapping points with too many text labels present on the plot. With the current plot, its difficult to tell which features may be driving separation in the data

```
# R base plot: plot PC1 & PC1, color by diagnosis
plot(wisc.pr$x[, 1:2], col=diagnosis)
```




```
# Repeat for components 1 and 3
plot(wisc.pr$x[, c(1,3)], col = diagnosis,
     xlab = "PC1", ylab = "PC3")
```



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

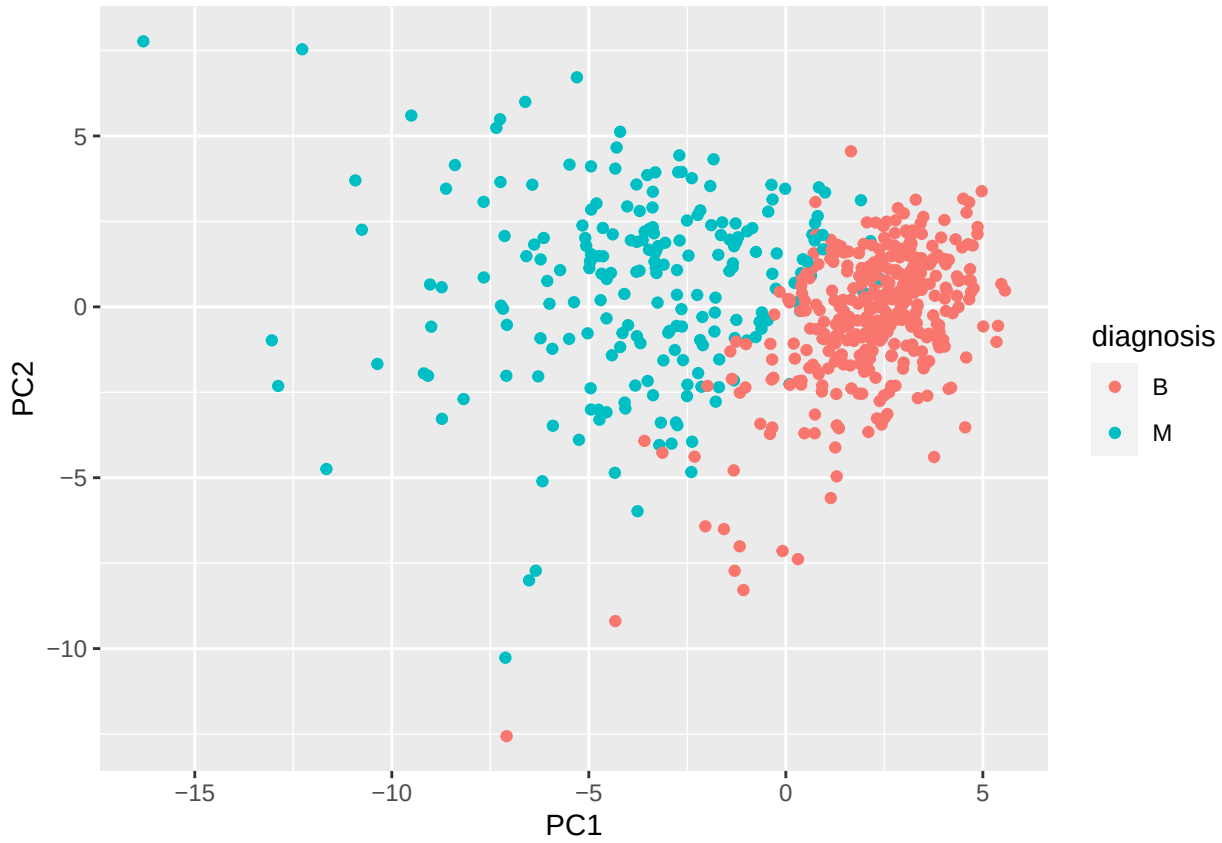
There is clearer separation of points in the graph of PC1 vs PC2, due to the fact that more variance is explained in the plot when compared to PC1 vs PC3

```
# Create clearer graphs in ggplot
library(ggplot2)

# Create dataframe of PC values
head(wisc.pr$x)
```

```
##                PC1        PC2        PC3        PC4         PC5         PC6
## 842302   -9.184755  -1.946870 -1.1221788 3.6305364  1.1940595  1.41018364
## 842517   -2.385703   3.764859 -0.5288274 1.1172808 -0.6212284  0.02863116
## 84300903 -5.728855   1.074229 -0.5512625 0.9112808  0.1769302  0.54097615
## 84348301 -7.116691 -10.266556 -3.2299475 0.1524129  2.9582754  3.05073750
## 84358402 -3.931842   1.946359  1.3885450 2.9380542 -0.5462667 -1.22541641
## 843786   -2.378155  -3.946456 -2.9322967 0.9402096  1.0551135 -0.45064213
##                 PC7         PC8         PC9        PC10        PC11        PC12
## 842302    2.15747152  0.39805698 -0.15698023 -0.8766305 -0.2627243 -0.8582593
## 842517    0.01334635 -0.24077660 -0.71127897  1.1060218 -0.8124048  0.1577838
## 84300903 -0.66757908 -0.09728813  0.02404449  0.4538760  0.6050715  0.1242777
## 84348301  1.42865363 -1.05863376 -1.40420412 -1.1159933  1.1505012  1.0104267
## 84358402 -0.93538950 -0.63581661 -0.26357355  0.3773724 -0.6507870 -0.1104183
```

```
## 843786    0.49001396  0.16529843 -0.13335576 -0.5299649 -0.1096698  0.0813699
##                PC13         PC14         PC15        PC16        PC17
## 842302    0.10329677 -0.690196797  0.601264078  0.74446075 -0.26523740
## 842517   -0.94269981 -0.652900844 -0.008966977 -0.64823831 -0.01719707
## 84300903 -0.41026561  0.016665095 -0.482994760  0.32482472  0.19075064
## 84348301 -0.93245070 -0.486988399  0.168699395  0.05132509  0.48220960
## 84358402  0.38760691 -0.538706543 -0.310046684 -0.15247165  0.13302526
## 843786   -0.02625135  0.003133944 -0.178447576 -0.01270566  0.19671335
##                PC18       PC19        PC20         PC21        PC22
## 842302   -0.54907956  0.1336499  0.34526111  0.096430045 -0.06878939
## 842517    0.31801756 -0.2473470 -0.11403274 -0.077259494  0.09449530
## 84300903 -0.08789759 -0.3922812 -0.20435242  0.310793246  0.06025601
## 84348301 -0.03584323 -0.0267241 -0.46432511  0.433811661  0.20308706
## 84358402 -0.01869779  0.4610302  0.06543782 -0.116442469  0.01763433
## 843786   -0.29727706 -0.1297265 -0.07117453 -0.002400178  0.10108043
##                PC23         PC24         PC25         PC26        PC27
## 842302    0.08444429  0.175102213  0.150887294 -0.201326305 -0.25236294
## 842517   -0.21752666 -0.011280193  0.170360355 -0.041092627  0.18111081
## 84300903 -0.07422581 -0.102671419 -0.171007656  0.004731249  0.04952586
## 84348301 -0.12399554 -0.153294780 -0.077427574 -0.274982822  0.18330078
## 84358402  0.13933105  0.005327110 -0.003059371  0.039219780  0.03213957
## 843786    0.03344819 -0.002837749 -0.122282765 -0.030272333 -0.08438081
##                  PC28         PC29          PC30
## 842302   -0.0338846387  0.045607590  0.0471277407
## 842517    0.0325955021 -0.005682424  0.0018662342
## 84300903  0.0469844833  0.003143131 -0.0007498749
## 84348301  0.0424469831 -0.069233868  0.0199198881
## 84358402 -0.0347556386  0.005033481 -0.0211951203
## 843786    0.0007296587 -0.019703996 -0.0034564331
```

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

ggplot(df.pc) +
  aes(x=PC1, y=PC2, col=diagnosis) +
  geom_point()
```



Variance Explained

```
# Calculate variance of each principal component
wisc.pr$sdev
```

```
##  [1] 3.64439401 2.38565601 1.67867477 1.40735229 1.28402903 1.09879780
##  [7] 0.82171778 0.69037464 0.64567392 0.59219377 0.54213992 0.51103950
## [13] 0.49128148 0.39624453 0.30681422 0.28260007 0.24371918 0.22938785
## [19] 0.22243559 0.17652026 0.17312681 0.16564843 0.15601550 0.13436892
## [25] 0.12442376 0.09043030 0.08306903 0.03986650 0.02736427 0.01153451
```

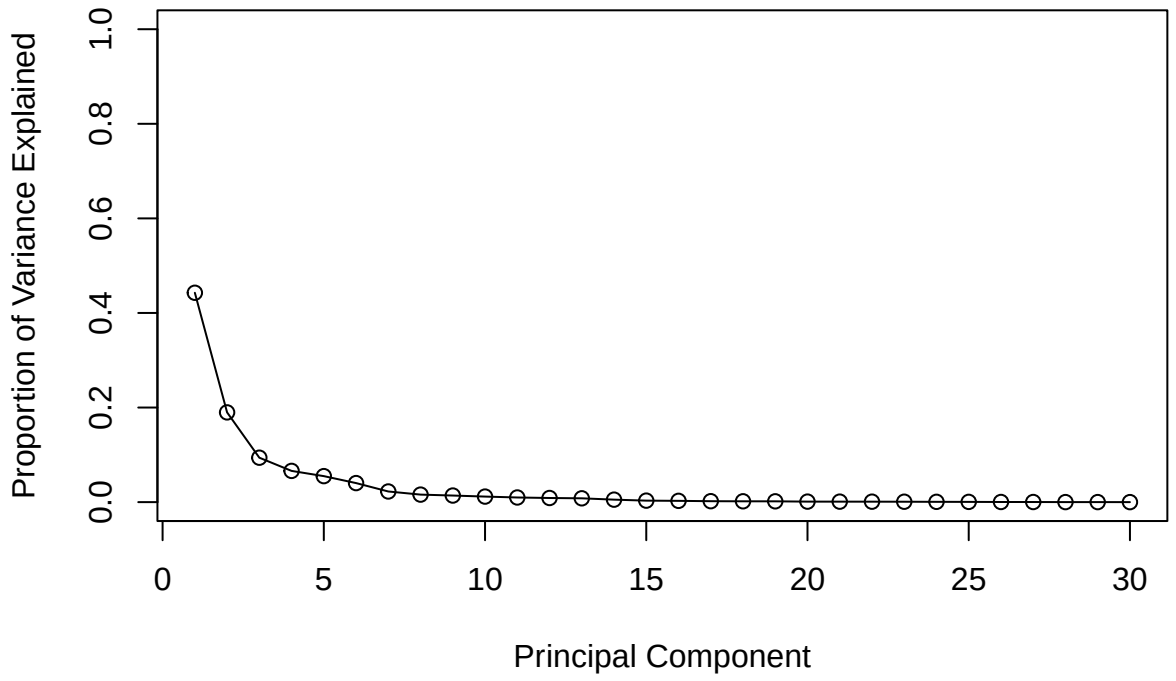
```
pr.var <- wisc.pr$sdev^2 #calculate variance
head(pr.var)
```

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

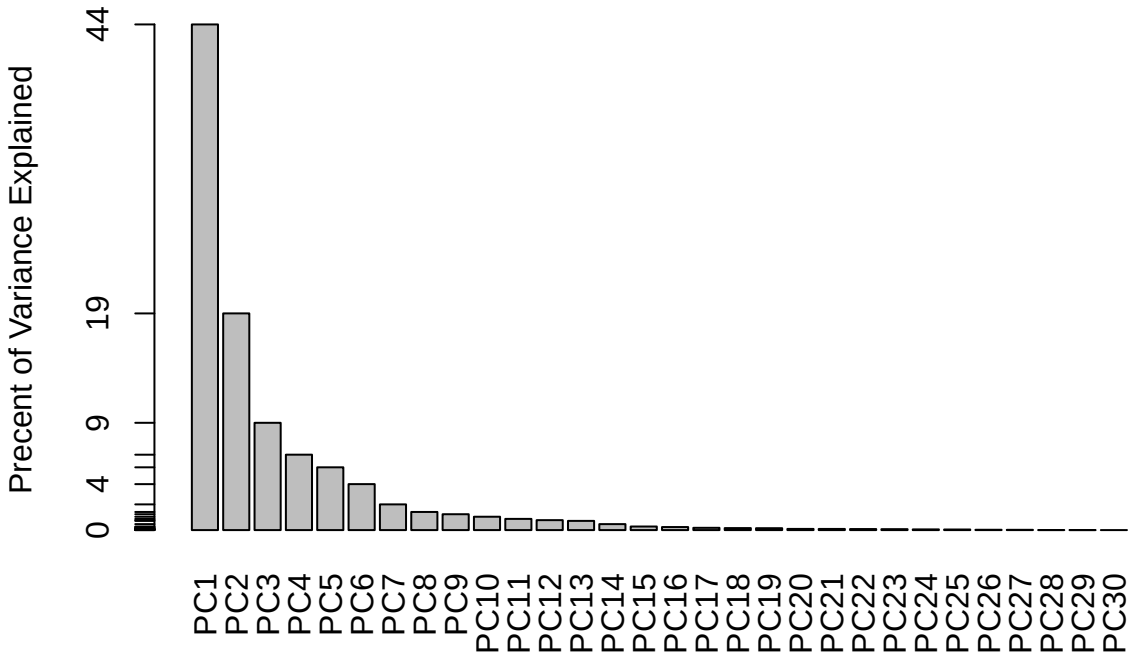
```
# Variance explained by each principal component: pve
pve <- pr.var / sum(pr.var)
head(pve)
```

```
## [1] 0.44272026 0.18971182 0.09393163 0.06602135 0.05495768 0.04024522
```

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



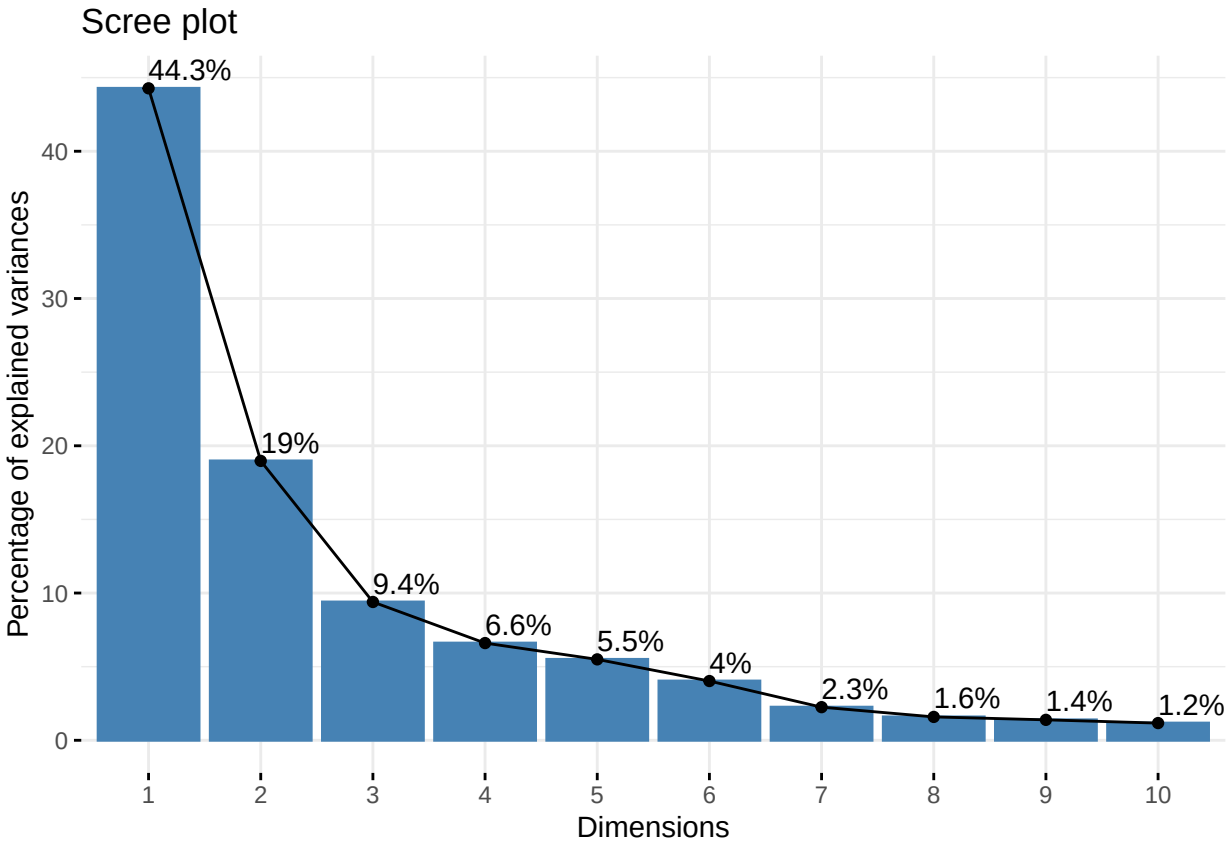

```r
# Alternative scree plot of the same data, note data driven y-axis
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```




```
## 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?

-0.2608538 (This # is the influence [relative magnitude] of this feature on the PC in question)

```
wisc.pr$rotation["concave.points_mean", "PC1"]
```

```
## [1] -0.2608538
```

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

5 principal componenets

```
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
```

Hierarchical clustering

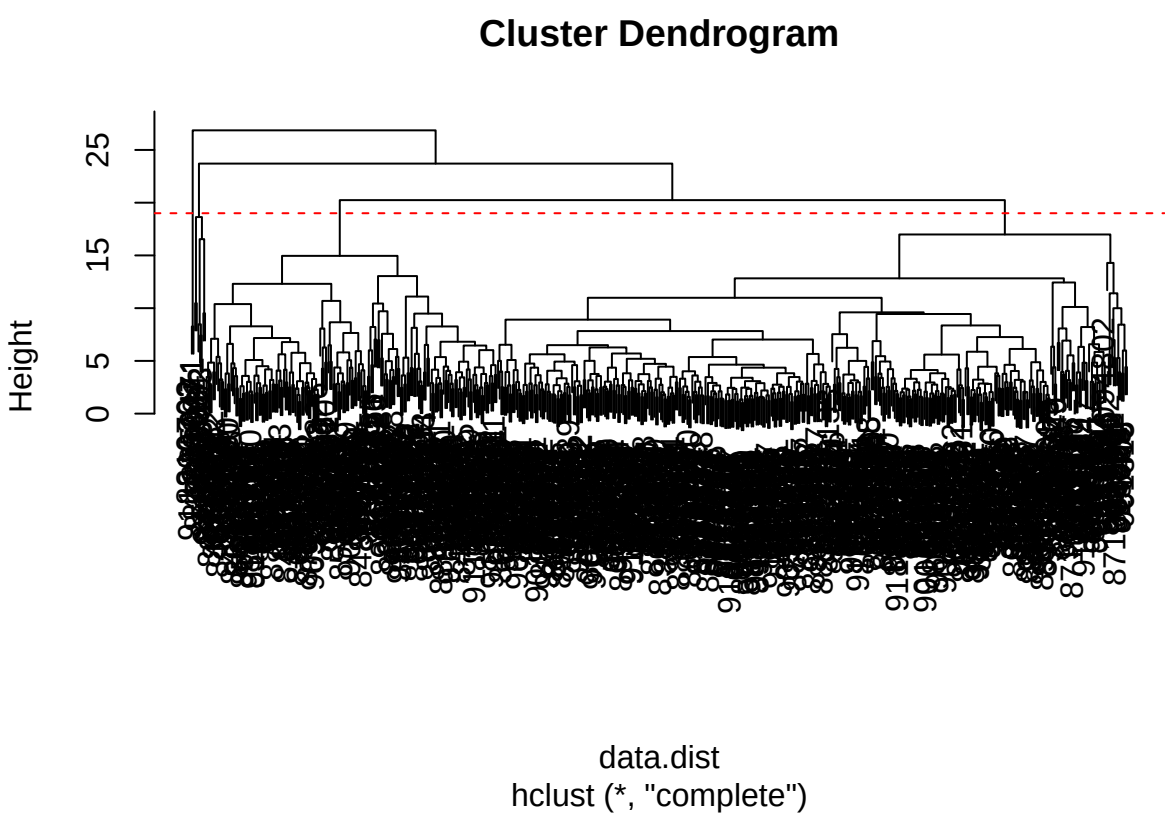
```
# Scale the wisc.data
data.scaled <- scale(wisc.data)

# Calculate Euclidean distance between all points
data.dist <- dist(data.scaled)

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

Results of HClust

Q11. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?

19

```
# Plot Cluster Dendrogram
plot(wisc.hclust)
abline(h=19, col="red", lty=2) #lty=2 specifies a dashed line
```



Selecting number of clusters

```
# Cut hclust data into 4 clusters
wisc.hclust.clusters <- cutree(wisc.hclust, k=4)
head(wisc.hclust.clusters)
```

```
##   842302   842517 84300903 84348301 84358402   843786
##        1        1        1        2        1        1
```

```
# Compare cluster membership to diagnosis
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?

No, 4 is the optimal cluster number. If you decrease below 4 clusters, all M & B are in the same cluster (#1). Increasing beyond 5 clusters further fragments the M & B cases into non-useful clusters.

```
for (i in 2:10){
  table <- table(cutree(wisc.hclust, k=i), diagnosis)
  print(table)
}
```

```
##    diagnosis
##       B   M
##   1 357 210
##   2   0   2
##    diagnosis
##       B   M
##   1 355 205
##   2   2   5
##   3   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   2   5
##   3 343  40
##   4   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   0   5
##   3 343  40
##   4   2   0
##   5   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   0   5
##   3 331  39
##   4   2   0
##   5  12   1
##   6   0   2
##    diagnosis
##       B   M
##   1  12 165
##   2   0   3
##   3 331  39
##   4   2   0
##   5  12   1
##   6   0   2
##   7   0   2
##    diagnosis
##       B   M
##   1  12  86
##   2   0  79
##   3   0   3
##   4 331  39
##   5   2   0
##   6  12   1
##   7   0   2
```

```
##   8   0   2
##    diagnosis
##       B   M
##   1  12  86
##   2   0  79
##   3   0   3
##   4 331  39
##   5   2   0
##   6  12   0
##   7   0   2
##   8   0   2
##   9   0   1
##     diagnosis
##        B   M
##   1   12  86
##   2    0  59
##   3    0   3
##   4  331  39
##   5    0  20
##   6    2   0
##   7   12   0
##   8    0   2
##   9    0   2
##   10   0   1
```

Using Different Methods

> Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning. Try the following methods: "single", "complete", "average", "ward.D2".

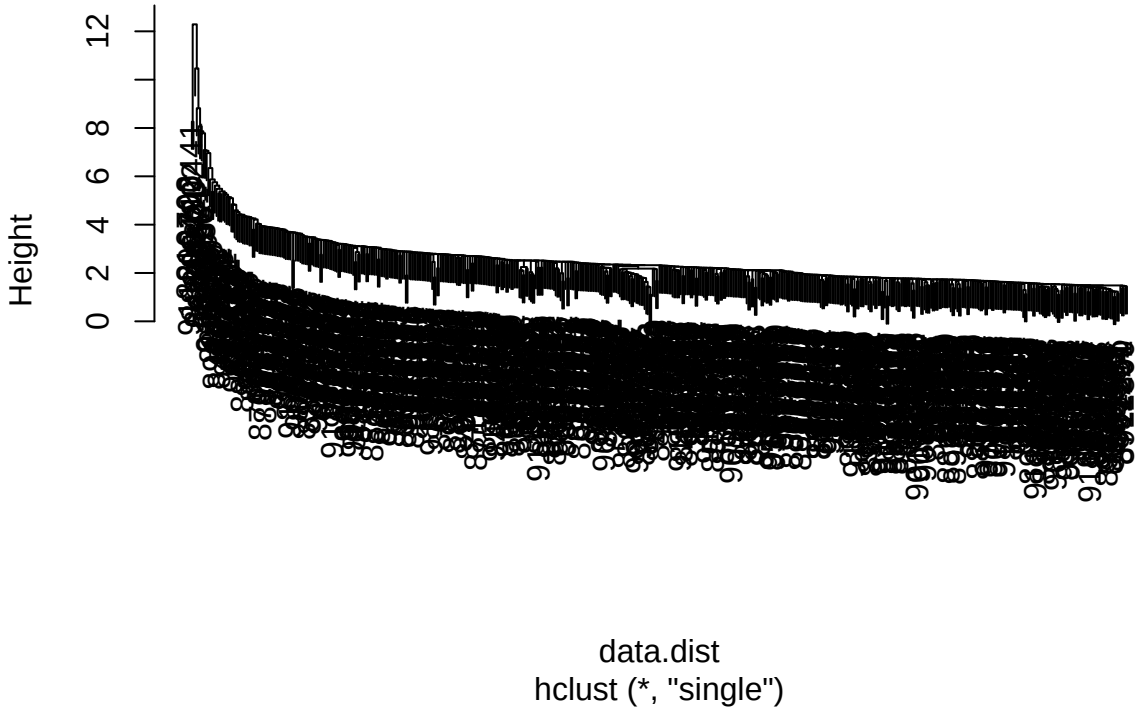
My favorite results come from "ward.D2".

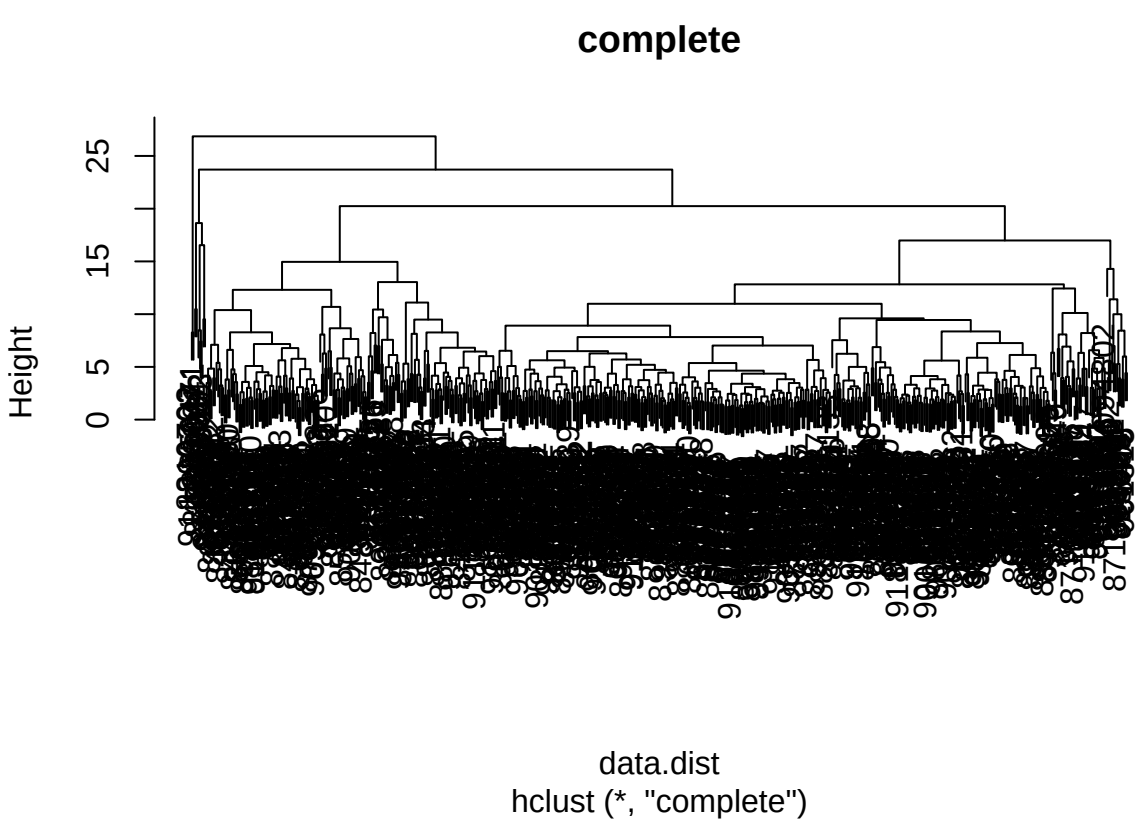
By visually inspecting the graphs, we see that the Dendrogram can be clearly cut into two clusters. When the results of hclust w/ data grouped into two clusters are plotted against diagnosis, nearly every case segregates into one of the two clusters:

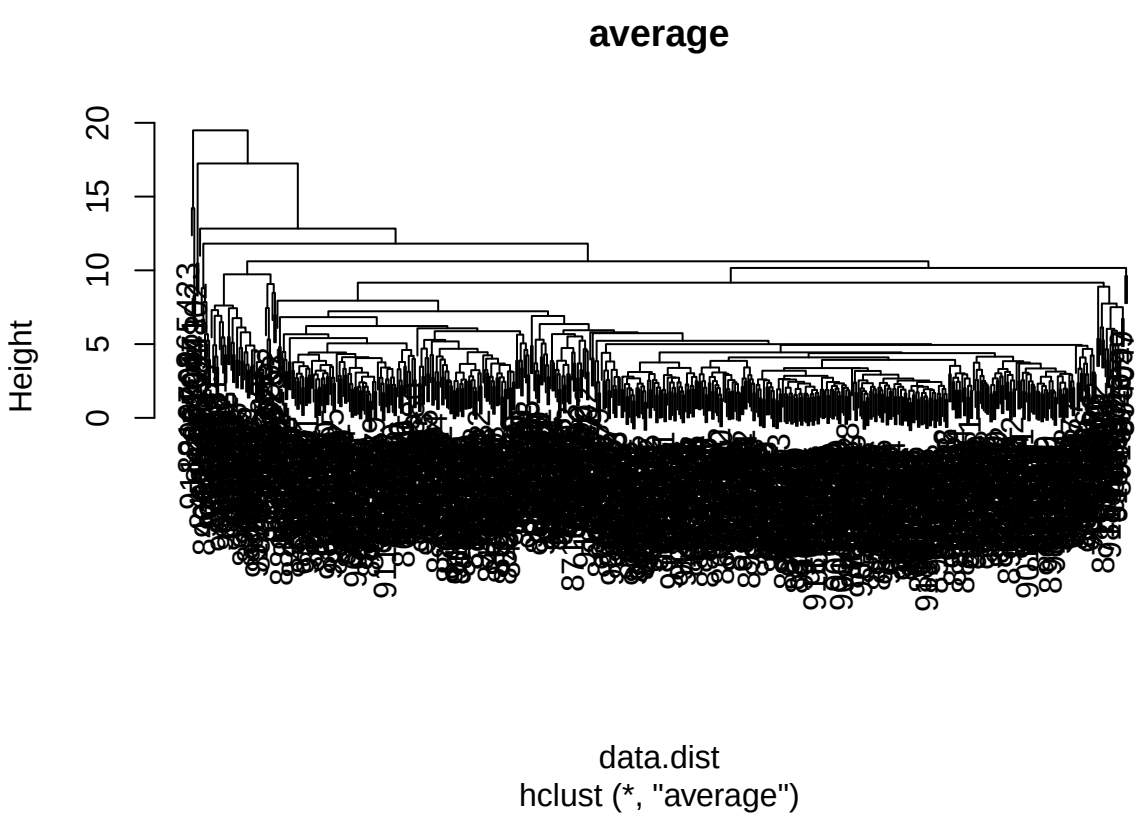
164 malignant in cluster 1 & 337 benin in cluster 2. Only 68 samples unassigned.

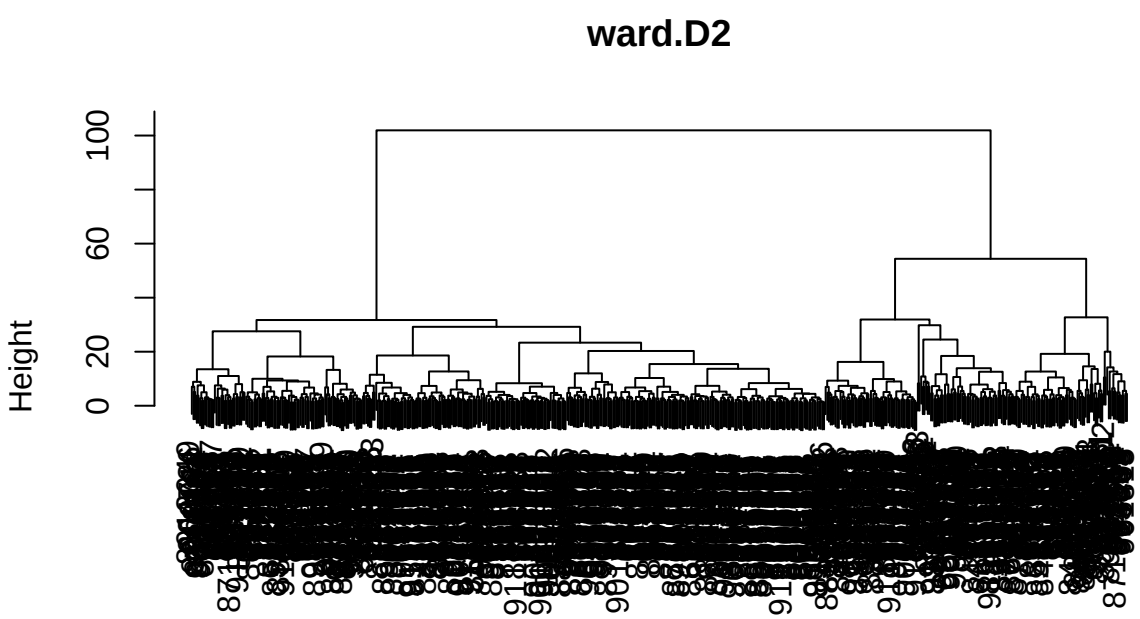
For the other methods, it is hard to "cut the tree" into any meaningful clusters that segregate the data into malignant vs benign clusters.

```
# Compare all 4 hclust clustering methods

# Create list of method names for for loop
hclust_methods <- c("single", "complete", "average", "ward.D2")

for (i in 1:length(hclust_methods)){
  wisc.hclust <- hclust(data.dist, method=hclust_methods[i])
  plot(wisc.hclust, main=hclust_methods[i])

}
```


single
Height
0
2
4
6
8
12
data.dist
hclust (*, "single")


complete
Height
0
5
15
25
data.dist
hclust (*, "complete")


average
Height
0
5
10
15
20
data.dist
hclust (*, "average")




```
# ward.D2 looks the best (all points break into two clusters), so lets inspect the clustering results v

table(cutree(hclust(data.dist, method="ward.D2"), k=2), diagnosis)
```

```
##    diagnosis
##       B   M
##   1  20 164
##   2 337  48
```

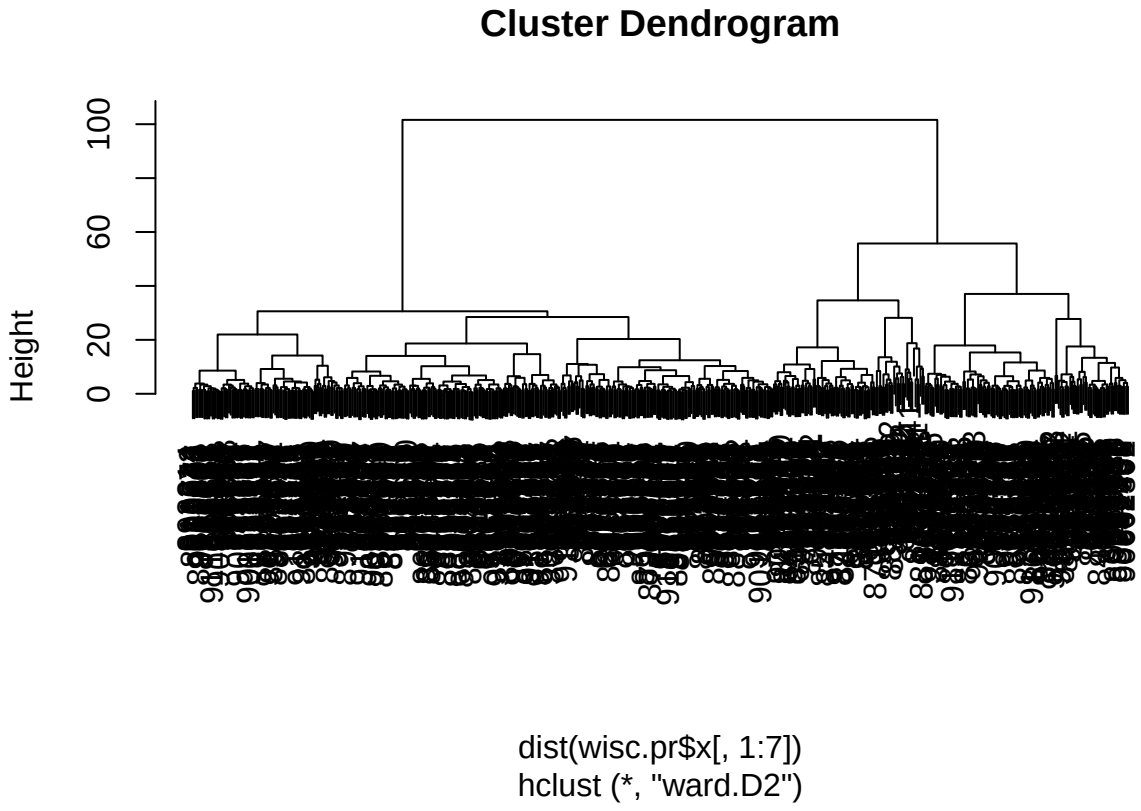
Combining Methods

```
# Determine number of principal components to describe at least 90% of variability
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
```

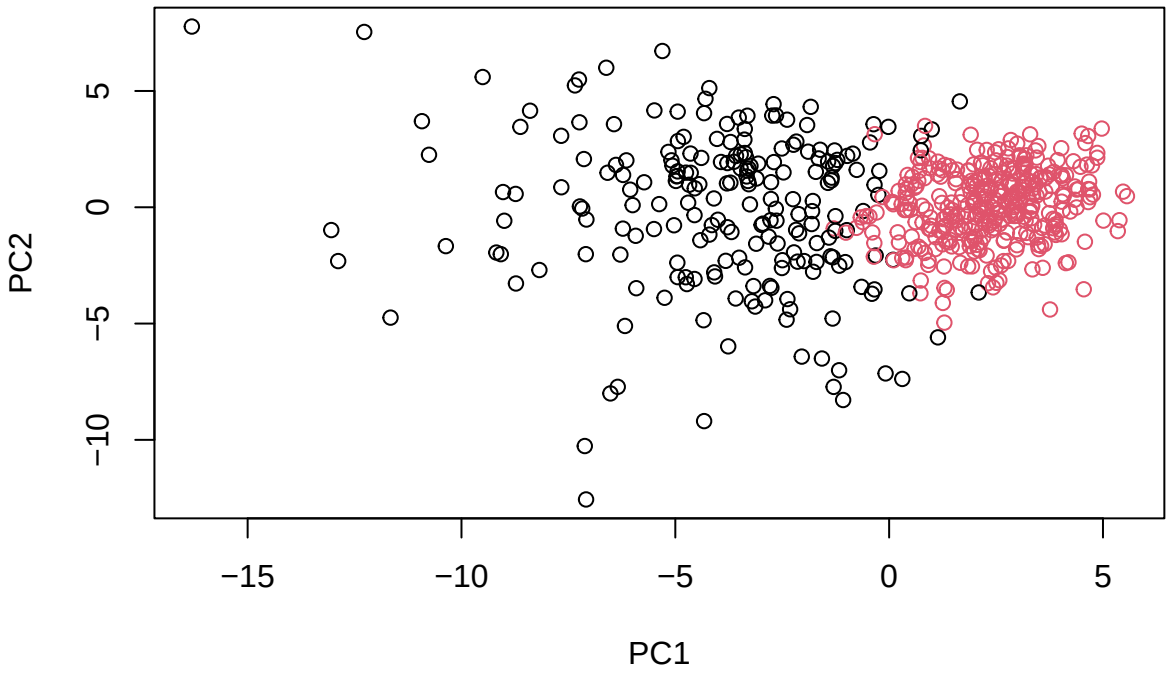
```
# Complete hclust with method="ward.D2"
# Perform hclust on distance matrix of first 7 principal components of "wisc.pr"

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

# Plot hclust dendrogram
plot(wisc.pr.hclust)
```



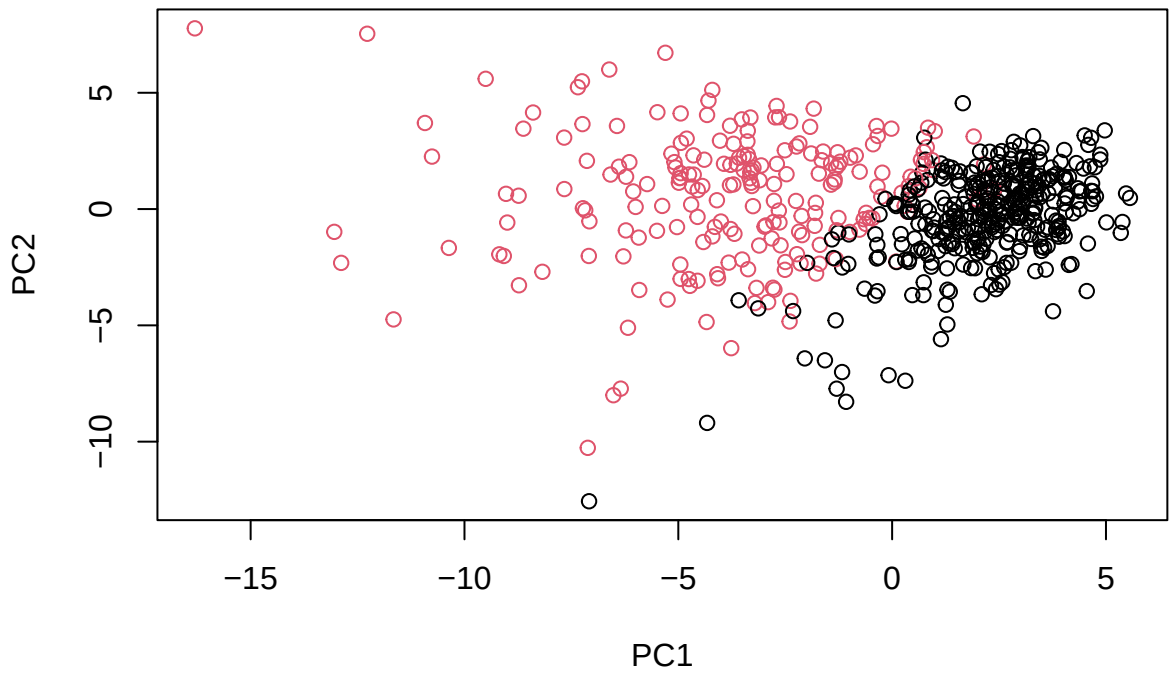

```
# Cut dendrogram into two branches to see if the data is clustering by diagnosis
grps <- cutree(wisc.pr.hclust, k=2)
table(grps, diagnosis)
```

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

```
# Plot PC1 & 2 and color by hclust cluster membership
plot(wisc.pr$x[,1:2], col=grps)
```




```
# Plot PC1 & 2 and color by diagnosis
plot(wisc.pr$x[,1:2], col=diagnosis)
```



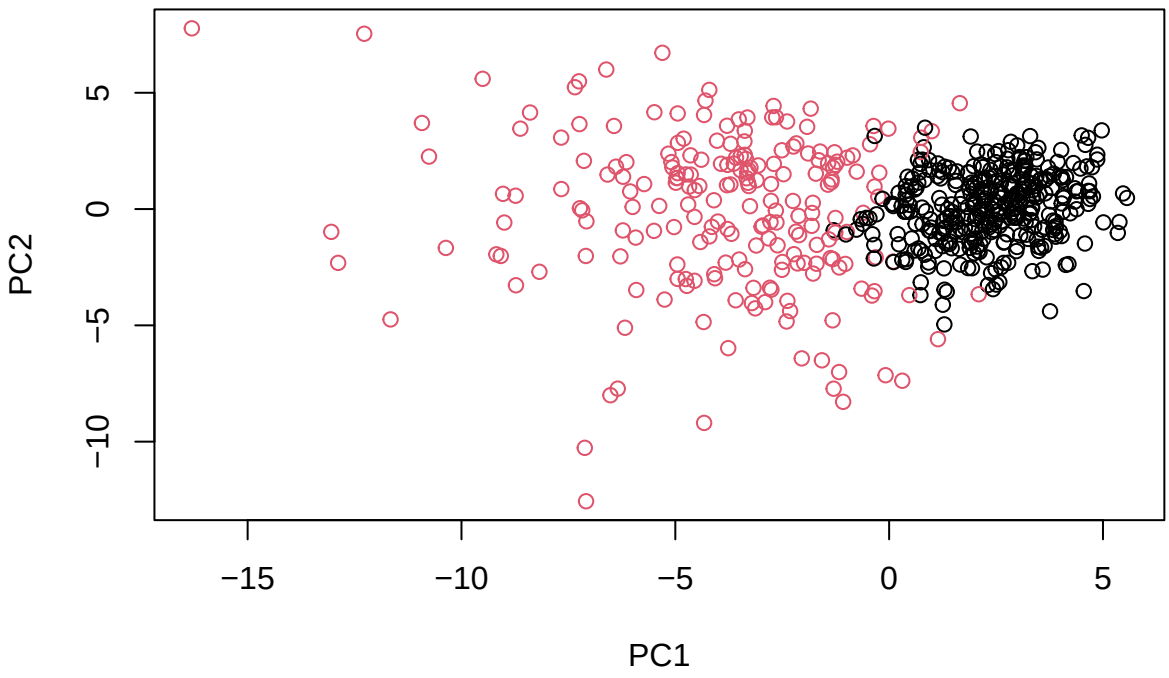

```
# Factor and reorder groups so that colors of diagnosis and cluster membership plots line up
g <- as.factor(grps)
levels(g)
```

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

```
# Reorder factors
g <- relevel(g,2)
levels(g)
```

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

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




```
# Cut "wisc.pr.hclust" clustering model from above into 2 clusters and assign to new variable

wisc.pr.hclust.clusters <- cutree(wisc.pr.hclust, k=2)

# Compare hclust results with diagnoses
table(wisc.pr.hclust.clusters, diagnosis)
```

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

Q15. How well does the newly created model with four clusters separate out the two diagnoses?

The new model built on the PCA results does a pretty good job of separating out the two diagnosis. 188 malignant cases are assigned to cluster 1 and 329 benign cases assigned to cluster 2. Overall, only 52 cases don't match the "right" cluster. This performs better than the hclust model built on the euclidean distances of the scaled wisc.data (question 13).

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.

Both hclust methods w/ models built on euclidean distances from the scaled wisc.data perform fairly well at separating the diagnosis.

When using the complete method, at k=4 clusters, 165 malignant cases are assigned to cluster 1, and 343 benign cases are assigned to cluster 3. When using the ward.d2 method on the wisc.disc distance matrix, fewer clusters (k=2) are needed to segregate out malignant and benign samples, w/ 164 malignant cases assigned to cluster 1 and 337 benign cases assigned to cluster 2.

The hclust model built on PCA results performs the "best" overall. With two clusters, its able to separate 188 malignant cases into cluster 1 and 329 benign cases into cluster 2.

```
# Compare how well different clustering methods work for separating diagnosis
# Note- kmeans section was optional.

# Hclust w/ euclidean distances, method = complete
hclust_euclidean_complete <- table(wisc.hclust.clusters, diagnosis)
hclust_euclidean_complete
```

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

```
# Hclust w/ euclidean distances, method = ward.D2
hclust_euclidean_ward <-table(cutree(hclust(data.dist, method="ward.D2"), k=2), diagnosis)
hclust_euclidean_ward
```

```
##    diagnosis
##       B   M
##   1  20 164
##   2 337  48
```

```
# clust w/ PCA, method = ward.D2
hclust_pca_ward <- table(wisc.pr.hclust.clusters, diagnosis)
hclust_pca_ward
```

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

Sensitivity and Specificity

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

The hclust clustering model built on the first 7 PCs (PCA model) with ward.d2 method. 88.67% sensitivity & 93.2% specificity.

Note:

Sensitivity= ability to correctly detect ill patients who do have the condition. TP/(TP+FN)

Specificity= ability to correctly reject healthy patients w/o a condition. TN/(TN+FN)

```
# True number of malignant and benign cases
length(grep("M", diagnosis))
```

```
## [1] 212
```

```
length(grep("B", diagnosis))
```

```
## [1] 357
```

```
# hclust_euclidean_complete
# Sensitivity
165/212
```

```
## [1] 0.7783019
```

```
# Specificity
343/(343+40)
```

```
## [1] 0.8955614
```

```
# hclust_euclidean_ward
# Sensitivity
164/212
```

```
## [1] 0.7735849
```

```
# specificity
337/(337+48)
```

```
## [1] 0.8753247
```

```
# hclust_pca_ward
# sensitivity
188/(212)
```

```
## [1] 0.8867925
```

```
# specificity
329/(329+24)
```

```
## [1] 0.9320113
```

```
# Note... need to find a better way to calculate this w/o hard coding
```

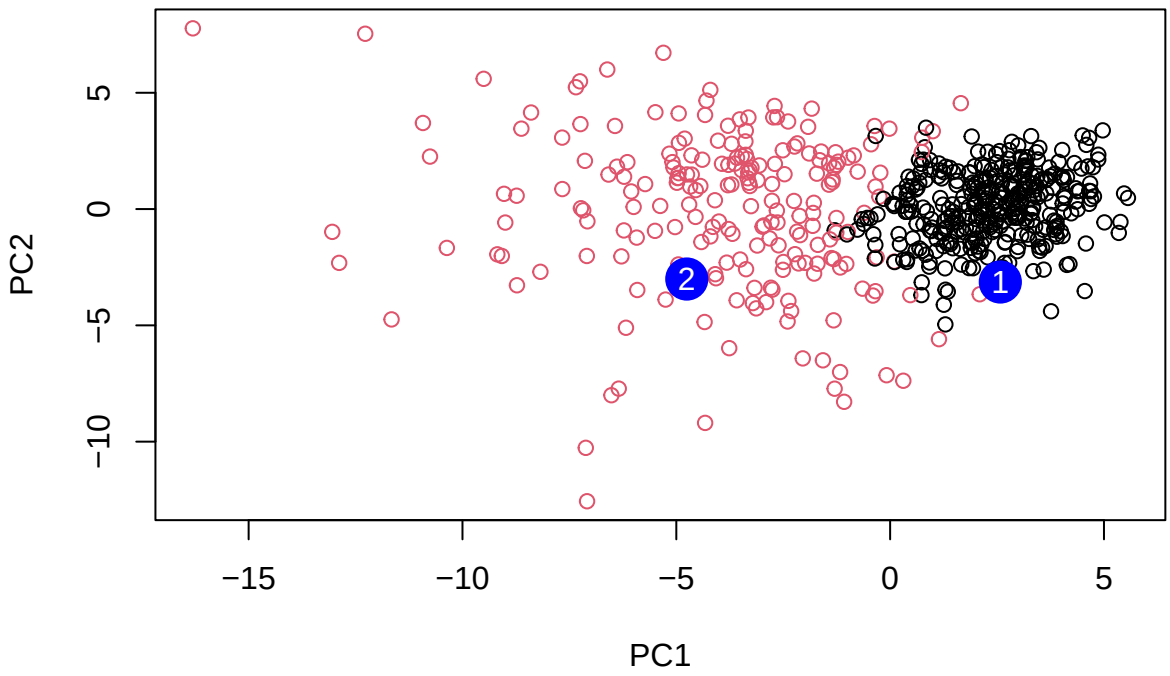
Prediction

Project new cancer cell data on previous PCA space

```r
#url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url) # Read in new dataset
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
# Generate plot
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 prioritie patient #2, who is in the red (malignant) cluster.

Session Info

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/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
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] factoextra_1.0.7 ggplot2_3.3.5
```

```
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.24         purrr_0.3.4       haven_2.4.1
##  [5] carData_3.0-4     colorspace_2.0-2  vctrs_0.3.8       generics_0.1.0
##  [9] htmltools_0.5.1.1 yaml_2.2.1        utf8_1.2.1        rlang_0.4.11
## [13] pillar_1.6.1      ggpubr_0.4.0      foreign_0.8-81    glue_1.4.2
## [17] withr_2.4.2       DBI_1.1.1         readxl_1.3.1      lifecycle_1.0.0
## [21] stringr_1.4.0     cellranger_1.1.0  munsell_0.5.0     ggsignif_0.6.2
## [25] gtable_0.3.0      zip_2.2.0         evaluate_0.14     labeling_0.4.2
## [29] knitr_1.33        rio_0.5.27        forcats_0.5.1     curl_4.3.2
## [33] fansi_0.5.0       highr_0.9         broom_0.7.8       Rcpp_1.0.7
## [37] scales_1.1.1      backports_1.2.1   abind_1.4-5       farver_2.1.0
## [41] hms_1.1.0         digest_0.6.27     openxlsx_4.2.4    stringi_1.6.2
## [45] rstatix_0.7.0     dplyr_1.0.7       ggrepel_0.9.1     grid_4.1.1
## [49] tools_4.1.1       magrittr_2.0.1    tibble_3.1.2      crayon_1.4.1
## [53] tidyr_1.1.3       car_3.0-11        pkgconfig_2.0.3   ellipsis_0.3.2
## [57] data.table_1.14.0 assertthat_0.2.1  rmarkdown_2.9     R6_2.5.0
## [61] compiler_4.1.1
```