Lecture 8 Breast Cell Analysis

Forrest Wang

#1. Exploratory Data Analysis Preparing the Data:

```
fna.data <- "https://bioboot.github.io/bimm143_F22/class-material/WisconsinCancer.csv"
wisc.df <- read.csv(fna.data, row.names = 1)
```

Use -1 to remove 1st column here

```
wisc.df
```

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean
842302	M	17.990	10.38	122.80	1001.0
842517	M	20.570	17.77	132.90	1326.0
84300903	M	19.690	21.25	130.00	1203.0
84348301	M	11.420	20.38	77.58	386.1
84358402	M	20.290	14.34	135.10	1297.0
843786	M	12.450	15.70	82.57	477.1
844359	M	18.250	19.98	119.60	1040.0
84458202	M	13.710	20.83	90.20	577.9
844981	M	13.000	21.82	87.50	519.8
84501001	M	12.460	24.04	83.97	475.9
845636	M	16.020	23.24	102.70	797.8
84610002	M	15.780	17.89	103.60	781.0
846226	M	19.170	24.80	132.40	1123.0
846381	M	15.850	23.95	103.70	782.7
84667401	M	13.730	22.61	93.60	578.3
84799002	M	14.540	27.54	96.73	658.8
848406	M	14.680	20.13	94.74	684.5
84862001	M	16.130	20.68	108.10	798.8
849014	M	19.810	22.15	130.00	1260.0
8510426	B	13.540	14.36	87.46	566.3
8510653	B	13.080	15.71	85.63	520.0

```
922577                  0.07399
922840                  0.09479
923169                  0.07920
923465                  0.07626
923748                  0.06592
923780                  0.08032
924084                  0.06484
924342                  0.07393
924632                  0.07242
924934                  0.08283
924964                  0.06742
925236                  0.06969
925277                  0.08004
925291                  0.08732
925292                  0.08321
925311                  0.05905
925622                  0.14090
926125                  0.09873
926424                  0.07115
926682                  0.06637
926954                  0.07820
927241                  0.12400
92751                   0.07039
```

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

Questions

Q1. How many oservations are in this dataset? A1: 569 observations

```
dim(wisc.data)
```

```
[1] 569  30
```

Q2: How many of the observations have a malignant diagnosis? A2: 212 observations have malignant diagnosis

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

```
  B   M
357 212
```

Q3: How many variables/features in the data suffixed w/ _mean? A3: 10 variables

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

```
[1] 10
```

#2. Principal Component Analysis Check column means & stddev

```
colMeans(wisc.data)
```

```
            radius_mean            texture_mean          perimeter_mean
           1.412729e+01            1.928965e+01            9.196903e+01
              area_mean         smoothness_mean        compactness_mean
           6.548891e+02            9.636028e-02            1.043410e-01
         concavity_mean     concave.points_mean          symmetry_mean
           8.879932e-02            4.891915e-02            1.811619e-01
 fractal_dimension_mean               radius_se              texture_se
           6.279761e-02            4.051721e-01            1.216853e+00
           perimeter_se                 area_se          smoothness_se
           2.866059e+00            4.033708e+01            7.040979e-03
         compactness_se           concavity_se       concave.points_se
           2.547814e-02            3.189372e-02            1.179614e-02
            symmetry_se    fractal_dimension_se            radius_worst
           2.054230e-02            3.794904e-03            1.626919e+01
          texture_worst         perimeter_worst              area_worst
           2.567722e+01            1.072612e+02            8.805831e+02
       smoothness_worst       compactness_worst         concavity_worst
           1.323686e-01            2.542650e-01            2.721885e-01
   concave.points_worst          symmetry_worst fractal_dimension_worst
           1.146062e-01            2.900756e-01            8.394582e-02
```

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

```
          radius_mean           texture_mean         perimeter_mean
         3.524049e+00           4.301036e+00           2.429898e+01
            area_mean        smoothness_mean       compactness_mean
         3.519141e+02           1.406413e-02           5.281276e-02
       concavity_mean    concave.points_mean         symmetry_mean
         7.971981e-02           3.880284e-02           2.741428e-02
fractal_dimension_mean              radius_se             texture_se
         7.060363e-03           2.773127e-01           5.516484e-01
         perimeter_se                area_se         smoothness_se
         2.021855e+00           4.549101e+01           3.002518e-03
       compactness_se           concavity_se      concave.points_se
         1.790818e-02           3.018606e-02           6.170285e-03
          symmetry_se   fractal_dimension_se            radius_worst
         8.266372e-03           2.646071e-03           4.833242e+00
        texture_worst        perimeter_worst              area_worst
         6.146258e+00           3.360254e+01           5.693570e+02
     smoothness_worst      compactness_worst         concavity_worst
         2.283243e-02           1.573365e-01           2.086243e-01
  concave.points_worst         symmetry_worst fractal_dimension_worst
         6.573234e-02           6.186747e-02           1.806127e-02
```

Perform PCA on wisc.data by completing the following code:

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

Examine result summary

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

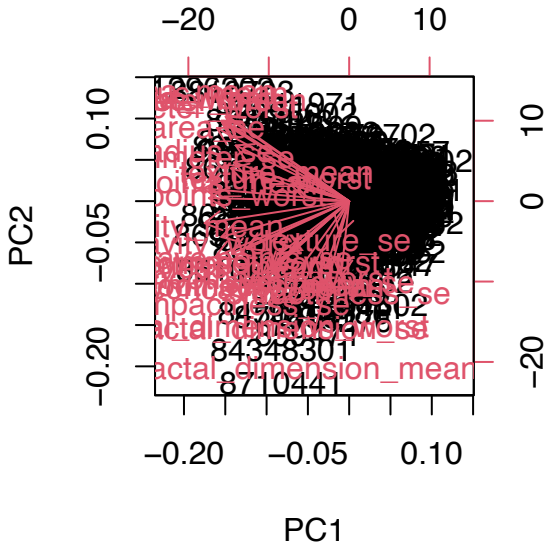
```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
Cumulative Proportion  0.4427 0.6324 0.72636 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
```

Q5: How many PCs required to describe >70% of original variance in data? A5: 3PCs

Q6: What stands out to you about this plot? Is it easy/difficult to interpret? A5: This plot is extremely difficult to examine and extract meaning from because none of the points are seperated, but overlapping and is unreadable

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

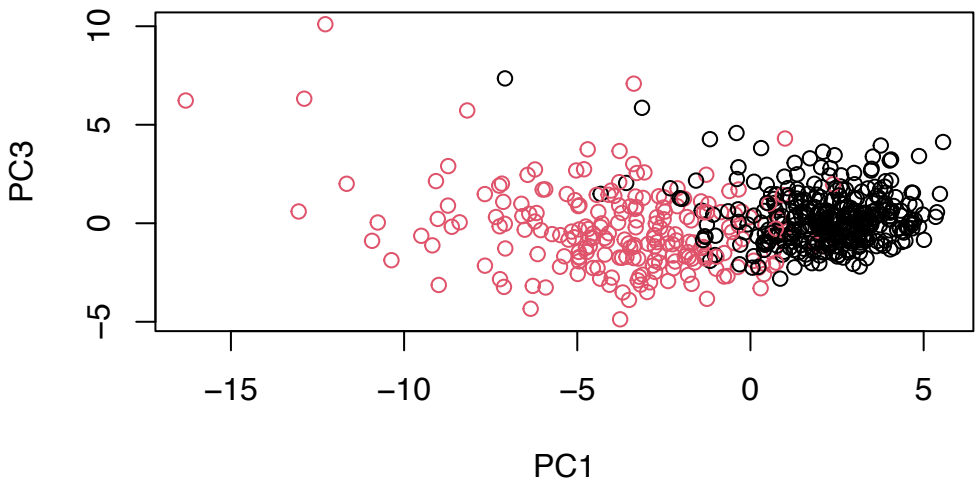


#Scatter plot observations by components 1 & 2

Q8: Generate similar plot for principal components 1 & 3. What do you notice about these plots? A8: The 2 components and their separation are closely resembling, although PC1 illustrates additional separation

Repeat for components 1 & 3

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



Create a data.frame for ggplot

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

Load the ggplot 2 package

```
library(ggplot2)
```

Calculate variance of each component

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

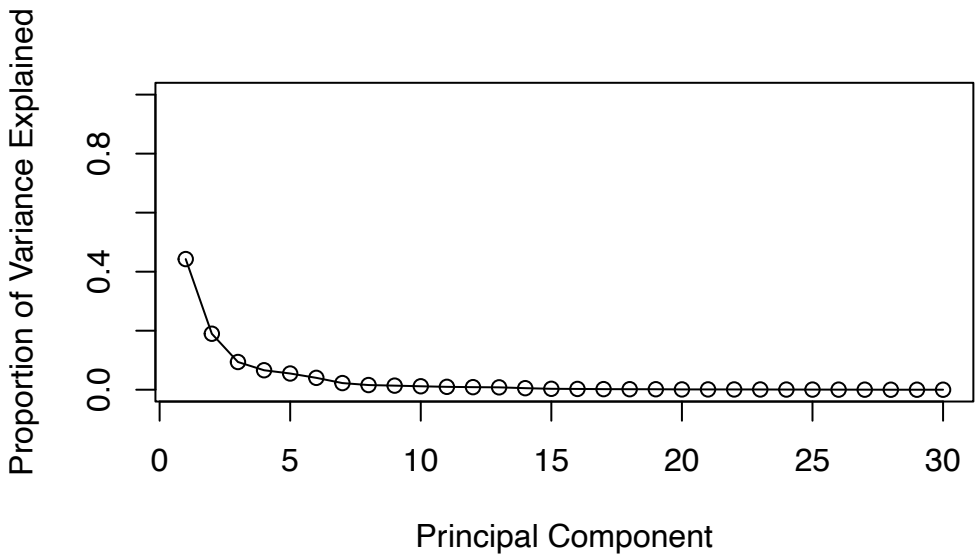
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

Variance explained by each principal component: pve

```
pve <- pr.var / sum(pr.var)
```

Plot variance explained for each principal

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



#3. Hierarchical clustering Scale the wisc.data using the "scale()" func

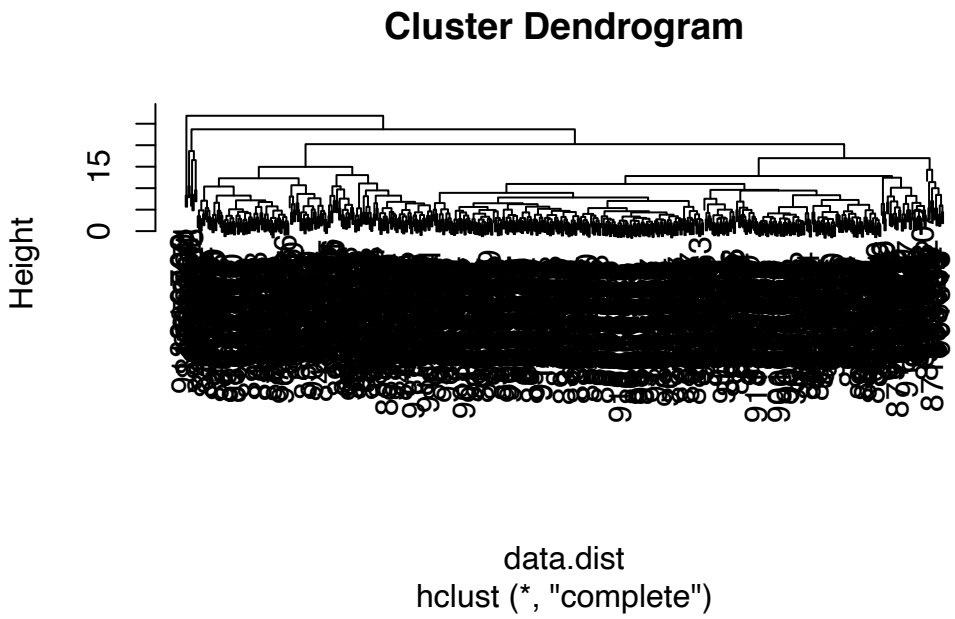
```
data.scaled <-scale(wisc.data)
```

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

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

Q11: Using the plot() and aline() functions, what is the height at which the clustering model has 4 clusters? A11: The height is between 19 and 20 for 4 clusters

```
plot(wisc.hclust)
```



Q12: Can you find a better cluster vs diagnosis match by cutting into a different number of clusters between 2 and 10? A12: Yes. A better cluster amount would be either 4 or 5 in which separation's significant impact is lost when the cluster number exceed 5. Conversely, when cluster#<5, separation doesn't exist.

Q13: Which method gives your favorite result for the same data.disc dataset? Explain your reasoning? A13: "Ward.D2" gives my favorate results because the two groups are separated more distinctly

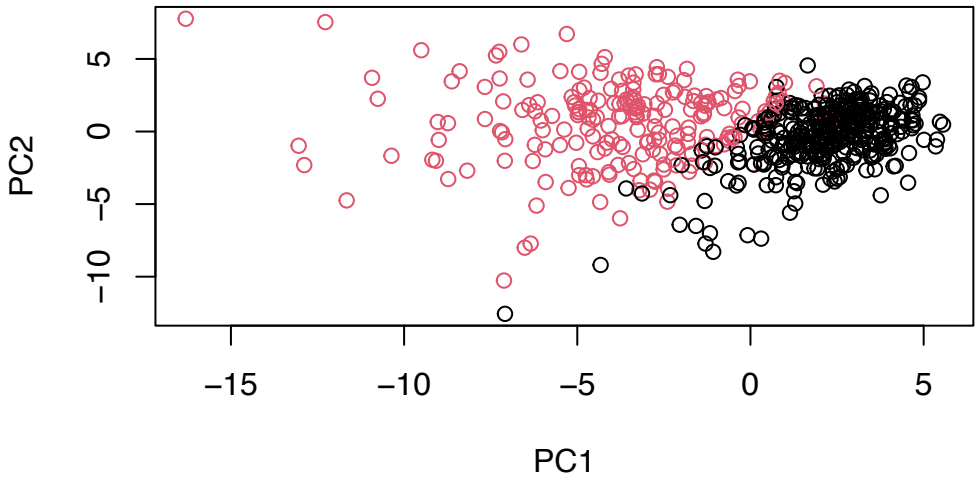
```
wisc.pr.hclust <- hclust(data.dist, method = "ward.D2")
grps <- cutree(wisc.pr.hclust, k=2)
table(grps)
```

```
grps
  1   2
184 385
```

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

```
grps
  1   2
184 385
```

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




```
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 4 clusters separate out the diagnoses? A15: The newer model illustrates much better variation and supports said diagnostic findings.

Compare to actual diagnoses

Q16: How well does the k-means & heirarchical clustering models you created in previous section do in terms of separating the diagnoses? A16: K_means & heirarchical clustering models created in previous section is useful in terms of separating diagnoses but it is much better interpreted visually through a graph in my opinion.

#6. Sensitivity/Specificity Q17. Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity

1. In terms of specificity, I believe pca hclust demonstrates the greatest visual measurements

```
HclustSpec <- 343/(343+40)
```

kmeans

```
kmeansSpec <- 343/(343+37)
```

pca hclust

```
pcaHclustSpec <- 329/(329+24)
```

2. In terms of sensitivity, I believe hclust has the greatest results

```
HclustSens <- 165/(165+12)
```

Kmeans

```
kmeansSense <- 175/(175+14)
```

pca hclust

```
pcaHclustSens <- 188/(188+28)
```