class10

##PRCOMP(x, SCALE = T)

##get our input date. our data for today comes from Winsconsin Breat

wisc.df <- read.csv("WisconsinCancer.csv")  
head(wisc.df)

## id diagnosis radius\_mean texture\_mean perimeter\_mean area\_mean  
## 1 842302 M 17.99 10.38 122.80 1001.0  
## 2 842517 M 20.57 17.77 132.90 1326.0  
## 3 84300903 M 19.69 21.25 130.00 1203.0  
## 4 84348301 M 11.42 20.38 77.58 386.1  
## 5 84358402 M 20.29 14.34 135.10 1297.0  
## 6 843786 M 12.45 15.70 82.57 477.1  
## smoothness\_mean compactness\_mean concavity\_mean concave.points\_mean  
## 1 0.11840 0.27760 0.3001 0.14710  
## 2 0.08474 0.07864 0.0869 0.07017  
## 3 0.10960 0.15990 0.1974 0.12790  
## 4 0.14250 0.28390 0.2414 0.10520  
## 5 0.10030 0.13280 0.1980 0.10430  
## 6 0.12780 0.17000 0.1578 0.08089  
## symmetry\_mean fractal\_dimension\_mean radius\_se texture\_se perimeter\_se  
## 1 0.2419 0.07871 1.0950 0.9053 8.589  
## 2 0.1812 0.05667 0.5435 0.7339 3.398  
## 3 0.2069 0.05999 0.7456 0.7869 4.585  
## 4 0.2597 0.09744 0.4956 1.1560 3.445  
## 5 0.1809 0.05883 0.7572 0.7813 5.438  
## 6 0.2087 0.07613 0.3345 0.8902 2.217  
## area\_se smoothness\_se compactness\_se concavity\_se concave.points\_se  
## 1 153.40 0.006399 0.04904 0.05373 0.01587  
## 2 74.08 0.005225 0.01308 0.01860 0.01340  
## 3 94.03 0.006150 0.04006 0.03832 0.02058  
## 4 27.23 0.009110 0.07458 0.05661 0.01867  
## 5 94.44 0.011490 0.02461 0.05688 0.01885  
## 6 27.19 0.007510 0.03345 0.03672 0.01137  
## symmetry\_se fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst  
## 1 0.03003 0.006193 25.38 17.33 184.60  
## 2 0.01389 0.003532 24.99 23.41 158.80  
## 3 0.02250 0.004571 23.57 25.53 152.50  
## 4 0.05963 0.009208 14.91 26.50 98.87  
## 5 0.01756 0.005115 22.54 16.67 152.20  
## 6 0.02165 0.005082 15.47 23.75 103.40  
## area\_worst smoothness\_worst compactness\_worst concavity\_worst  
## 1 2019.0 0.1622 0.6656 0.7119  
## 2 1956.0 0.1238 0.1866 0.2416  
## 3 1709.0 0.1444 0.4245 0.4504  
## 4 567.7 0.2098 0.8663 0.6869  
## 5 1575.0 0.1374 0.2050 0.4000  
## 6 741.6 0.1791 0.5249 0.5355  
## concave.points\_worst symmetry\_worst fractal\_dimension\_worst X  
## 1 0.2654 0.4601 0.11890 NA  
## 2 0.1860 0.2750 0.08902 NA  
## 3 0.2430 0.3613 0.08758 NA  
## 4 0.2575 0.6638 0.17300 NA  
## 5 0.1625 0.2364 0.07678 NA  
## 6 0.1741 0.3985 0.12440 NA

#clean data. select from column 3 to 32

wisc.data <- as.matrix(wisc.df[ , 3:32])  
head(wisc.data)

## radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean  
## [1,] 17.99 10.38 122.80 1001.0 0.11840  
## [2,] 20.57 17.77 132.90 1326.0 0.08474  
## [3,] 19.69 21.25 130.00 1203.0 0.10960  
## [4,] 11.42 20.38 77.58 386.1 0.14250  
## [5,] 20.29 14.34 135.10 1297.0 0.10030  
## [6,] 12.45 15.70 82.57 477.1 0.12780  
## compactness\_mean concavity\_mean concave.points\_mean symmetry\_mean  
## [1,] 0.27760 0.3001 0.14710 0.2419  
## [2,] 0.07864 0.0869 0.07017 0.1812  
## [3,] 0.15990 0.1974 0.12790 0.2069  
## [4,] 0.28390 0.2414 0.10520 0.2597  
## [5,] 0.13280 0.1980 0.10430 0.1809  
## [6,] 0.17000 0.1578 0.08089 0.2087  
## fractal\_dimension\_mean radius\_se texture\_se perimeter\_se area\_se  
## [1,] 0.07871 1.0950 0.9053 8.589 153.40  
## [2,] 0.05667 0.5435 0.7339 3.398 74.08  
## [3,] 0.05999 0.7456 0.7869 4.585 94.03  
## [4,] 0.09744 0.4956 1.1560 3.445 27.23  
## [5,] 0.05883 0.7572 0.7813 5.438 94.44  
## [6,] 0.07613 0.3345 0.8902 2.217 27.19  
## smoothness\_se compactness\_se concavity\_se concave.points\_se symmetry\_se  
## [1,] 0.006399 0.04904 0.05373 0.01587 0.03003  
## [2,] 0.005225 0.01308 0.01860 0.01340 0.01389  
## [3,] 0.006150 0.04006 0.03832 0.02058 0.02250  
## [4,] 0.009110 0.07458 0.05661 0.01867 0.05963  
## [5,] 0.011490 0.02461 0.05688 0.01885 0.01756  
## [6,] 0.007510 0.03345 0.03672 0.01137 0.02165  
## fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst area\_worst  
## [1,] 0.006193 25.38 17.33 184.60 2019.0  
## [2,] 0.003532 24.99 23.41 158.80 1956.0  
## [3,] 0.004571 23.57 25.53 152.50 1709.0  
## [4,] 0.009208 14.91 26.50 98.87 567.7  
## [5,] 0.005115 22.54 16.67 152.20 1575.0  
## [6,] 0.005082 15.47 23.75 103.40 741.6  
## smoothness\_worst compactness\_worst concavity\_worst concave.points\_worst  
## [1,] 0.1622 0.6656 0.7119 0.2654  
## [2,] 0.1238 0.1866 0.2416 0.1860  
## [3,] 0.1444 0.4245 0.4504 0.2430  
## [4,] 0.2098 0.8663 0.6869 0.2575  
## [5,] 0.1374 0.2050 0.4000 0.1625  
## [6,] 0.1791 0.5249 0.5355 0.1741  
## symmetry\_worst fractal\_dimension\_worst  
## [1,] 0.4601 0.11890  
## [2,] 0.2750 0.08902  
## [3,] 0.3613 0.08758  
## [4,] 0.6638 0.17300  
## [5,] 0.2364 0.07678  
## [6,] 0.3985 0.12440

#Q. How many patients are there in this dataset?

nrow(wisc.df)

## [1] 569

#Q. How many cancere and non-cancer patients? access with dollar sign

wisc.df$diagnosis

## [1] M M M M M M M M M M M M M M M M M M M B B B M M M M M M M M M M M M M M M  
## [38] B M M M M M M M M B M B B B B B M M B M M B B B B M B M M B B B B M B M M  
## [75] B M B M M B B B M M B M M M B B B M B B M M B B B M M B B B B M B B M B B  
## [112] B B B B B B M M M B M M B B B M M B M B M M B M M B B M B B M B B B B M B  
## [149] B B B B B B B B M B B B B M M B M B B M M B B M M B B B B M B B M M M B M  
## [186] B M B B B M B B M M B M M M M B M M M B M B M B B M B M M M M B B M M B B  
## [223] B M B B B B B M M B B M B B M M B M B B B B M B B B B B M B M M M M M M M  
## [260] M M M M M M M B B B B B B M B M B B M B B M B M M B B B B B B B B B B B B  
## [297] B M B B M B M B B B B B B B B B B B B B B M B B B M B M B B B B M M M B B  
## [334] B B M B M B M B B B M B B B B B B B M M M B B B B B B B B B B B M M B M M  
## [371] M B M M B B B B B M B B B B B M B B B M B B M M B B B B B B M B B B B B B  
## [408] B M B B B B B M B B M B B B B B B B B B B B B M B M M B M B B B B B M B B  
## [445] M B M B B M B M B B B B B B B B M M B B B B B B M B B B B B B B B B B M B  
## [482] B B B B B B M B M B B M B B B B B M M B M B M B B B B B M B B M B M B M M  
## [519] B B B M B B B B B B B B B B B M B M M B B B B B B B B B B B B B B B B B B  
## [556] B B B B B B B M M M M M M B  
## Levels: B M

#use the table version

table(wisc.df$diagnosis)

##   
## B M   
## 357 212

table(wisc.df$diagnosis == "M")

##   
## FALSE TRUE   
## 357 212

#use a graph to see how many of the columns have “mean” component ##graphs look for patterns within a vector

##\*\*use colnames()

colnames(wisc.df)

## [1] "id" "diagnosis"   
## [3] "radius\_mean" "texture\_mean"   
## [5] "perimeter\_mean" "area\_mean"   
## [7] "smoothness\_mean" "compactness\_mean"   
## [9] "concavity\_mean" "concave.points\_mean"   
## [11] "symmetry\_mean" "fractal\_dimension\_mean"   
## [13] "radius\_se" "texture\_se"   
## [15] "perimeter\_se" "area\_se"   
## [17] "smoothness\_se" "compactness\_se"   
## [19] "concavity\_se" "concave.points\_se"   
## [21] "symmetry\_se" "fractal\_dimension\_se"   
## [23] "radius\_worst" "texture\_worst"   
## [25] "perimeter\_worst" "area\_worst"   
## [27] "smoothness\_worst" "compactness\_worst"   
## [29] "concavity\_worst" "concave.points\_worst"   
## [31] "symmetry\_worst" "fractal\_dimension\_worst"  
## [33] "X"

##use *grep()*, search in the help box, and use the ones without the equal sign

grep("\_mean", colnames(wisc.df))

## [1] 3 4 5 6 7 8 9 10 11 12

##to just get the columns with the \*\*\_mean\*\*

grep("\_mean", colnames(wisc.df), value = TRUE)

## [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"

#the number of columns there are with \*\_mean\*

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

## [1] 10

##Enter Principal of Component Analysis

##First check whether our input should be scaled. let’s check the sd() and mean() of all our columns in wisc.data

##APPLY(x, 2, function you want to apply like Standard deviation)

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

#try to round this up. use “2” to command it to show 2 decimal points

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

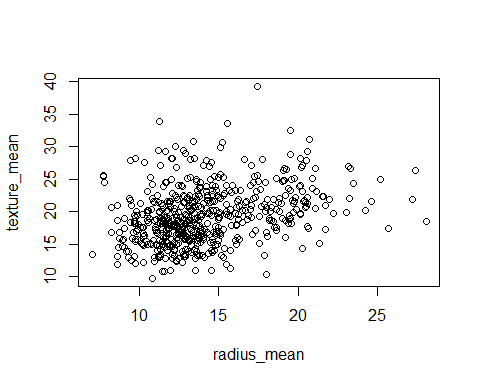
## radius\_mean texture\_mean perimeter\_mean   
## 3.52 4.30 24.30   
## area\_mean smoothness\_mean compactness\_mean   
## 351.91 0.01 0.05   
## concavity\_mean concave.points\_mean symmetry\_mean   
## 0.08 0.04 0.03   
## fractal\_dimension\_mean radius\_se texture\_se   
## 0.01 0.28 0.55   
## perimeter\_se area\_se smoothness\_se   
## 2.02 45.49 0.00   
## compactness\_se concavity\_se concave.points\_se   
## 0.02 0.03 0.01   
## symmetry\_se fractal\_dimension\_se radius\_worst   
## 0.01 0.00 4.83   
## texture\_worst perimeter\_worst area\_worst   
## 6.15 33.60 569.36   
## smoothness\_worst compactness\_worst concavity\_worst   
## 0.02 0.16 0.21   
## concave.points\_worst symmetry\_worst fractal\_dimension\_worst   
## 0.07 0.06 0.02

#perform PCA on wisc.data by completing the following

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

plot(wisc.data)



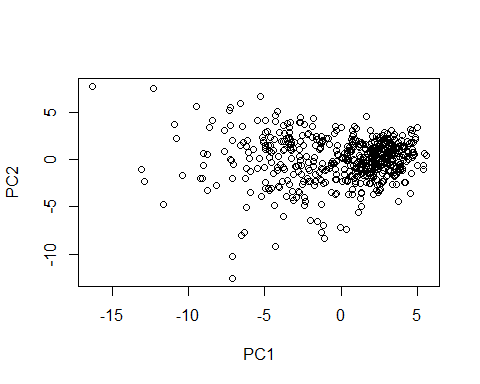
#this is a hot mess! we need to cook our own PCA plot. need to access the results within the wisc.pr

attributes(wisc.pr)

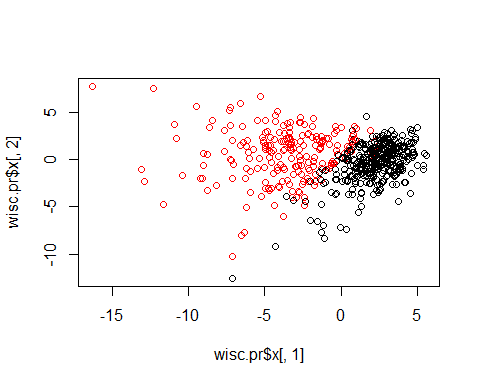
## $names  
## [1] "sdev" "rotation" "center" "scale" "x"   
##   
## $class  
## [1] "prcomp"

##want ot use $x component to make our PCA. plot column PC1 to PC2

plot(wisc.pr$x[ ,1:2])



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

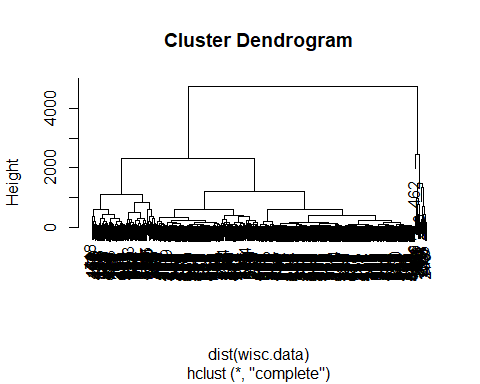


table(wisc.df$diagnosis)

##   
## B M   
## 357 212

#this is super important to do clusters

hc <- hclust (dist(wisc.data))  
plot(hc)

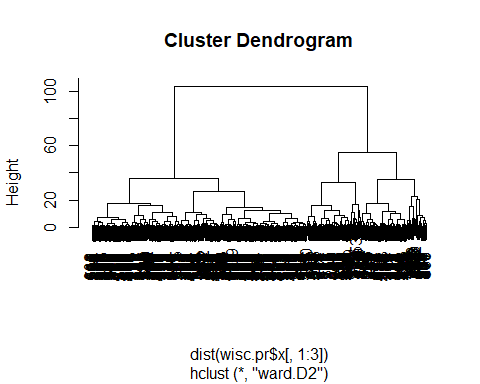


#use result of PCA fo do my clustering

hclust(dist(wisc.pr$x[ , 1:3]))

##   
## Call:  
## hclust(d = dist(wisc.pr$x[, 1:3]))  
##   
## Cluster method : complete   
## Distance : euclidean   
## Number of objects: 569

wisc.pr.hc <- hclust(dist(wisc.pr$x[ , 1:3]), method="ward.D2")  
plot(wisc.pr.hc)



#where to cut this clustogram. need the cutree()

grps <- cutree(wisc.pr.hc, k=2)  
grps

## [1] 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1  
## [38] 2 2 1 2 1 1 1 2 1 2 1 2 2 2 2 2 1 2 2 1 1 2 2 2 2 1 2 1 1 2 2 1 2 1 1 1 2  
## [75] 2 2 2 1 1 2 2 1 1 1 2 1 2 1 2 1 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 1 2 2 1 2 2  
## [112] 2 1 2 2 2 2 1 1 2 2 1 1 2 2 2 2 1 1 1 2 1 1 2 1 2 2 2 1 2 2 2 2 2 2 2 1 2  
## [149] 2 2 2 1 1 2 2 2 1 2 2 2 2 1 1 2 1 2 2 2 1 2 2 2 1 2 2 2 1 1 2 2 1 1 1 2 2  
## [186] 2 2 2 2 2 1 2 2 1 1 2 1 1 1 1 2 1 1 1 2 2 2 2 1 2 1 2 1 1 1 1 2 2 1 1 2 2  
## [223] 2 1 2 2 2 2 2 1 1 2 2 1 2 2 1 1 2 1 2 2 1 2 1 2 2 1 2 2 1 2 1 1 1 1 1 1 1  
## [260] 1 1 2 1 2 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 2 2 2 2 2  
## [297] 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 2 1 2 2 2 2 1 1 1 2 2  
## [334] 2 2 1 2 1 2 1 1 2 2 1 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1  
## [371] 1 2 1 1 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2  
## [408] 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 1 2 1 1 2 1 2 2 2 2 2 1 2 2  
## [445] 1 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2  
## [482] 2 2 2 2 1 2 1 2 2 2 2 1 2 2 2 1 2 1 1 2 1 2 1 1 1 2 2 2 1 2 2 1 2 2 2 1 1  
## [519] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [556] 2 2 2 2 2 2 2 1 1 1 1 2 1 2

#how many 1s and 2s? use a table

table(grps)

## grps  
## 1 2   
## 203 366

#ho wmnay are benign (B) and how many are malignant (M)

table(grps, wisc.df$diagnosis)

##   
## grps B M  
## 1 24 179  
## 2 333 33

#24 and 33 are false positives

#Now look at NEW DATA FILE

#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

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

