Data Analysis

Sridhar

28 January 2019

#install.packages("ggplot2")  
#install.packages("ggvis")  
#install.packages("mice")  
library(ggplot2)  
library(ggvis)

##   
## Attaching package: 'ggvis'

## The following object is masked from 'package:ggplot2':  
##   
## resolution

library(knitr)  
library(mice)

## Loading required package: lattice

##   
## Attaching package: 'mice'

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(caret)  
library(doParallel)

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel

## Load the Data

Breast Cancer Wisconsin(Diagnosis) data is loaded and the column names are defined based on the attribute information.

### Dataset 1

The outcome variable is the “classes” and it has following category of data

* Malignant or
* Benign breast mass

The phenotypes for characterization are \* Sample ID (code number) \* Clump thickness \* Uniformity of cell size \* Uniformity of cell shape \* Marginal adhesion \* Single epithelial cell size \* Number of bare nuclei \* Bland chromatin \* Number of normal nuclei \* Mitosis \* Classes, i.e. diagnosis

diagnosis\_data <- read.csv("~/Desktop/BreastCancer/Data/breast-cancer-wisconsin.data.csv", header = FALSE)  
colnames(diagnosis\_data) <- c("sample\_code\_number", "clump\_thickness", "uniformity\_of\_cell\_size", "uniformity\_of\_cell\_shape", "marginal\_adhesion", "single\_epithelial\_cell\_size",   
 "bare\_nuclei", "bland\_chromatin", "normal\_nucleoli", "mitosis", "classes")

# impute missing data  
  
diagnosis\_data[,2:10] <- apply(diagnosis\_data[, 2:10], 2, function(x) as.numeric(as.character(x)))

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

dataset\_impute <- mice(diagnosis\_data[, 2:10], print = FALSE)  
diagnosis\_data <- cbind(diagnosis\_data[, 11, drop = FALSE], mice::complete(dataset\_impute, 1))  
  
diagnosis\_data$classes <- as.factor(diagnosis\_data$classes)  
  
# how many benign and malignant cases are there?  
summary(diagnosis\_data$classes)

## 2 4   
## 458 241

The data is presented below.

head(diagnosis\_data)

## classes clump\_thickness uniformity\_of\_cell\_size uniformity\_of\_cell\_shape  
## 1 2 5 1 1  
## 2 2 5 4 4  
## 3 2 3 1 1  
## 4 2 6 8 8  
## 5 2 4 1 1  
## 6 4 8 10 10  
## marginal\_adhesion single\_epithelial\_cell\_size bare\_nuclei  
## 1 1 2 1  
## 2 5 7 10  
## 3 1 2 2  
## 4 1 3 4  
## 5 3 2 1  
## 6 8 7 10  
## bland\_chromatin normal\_nucleoli mitosis  
## 1 3 1 1  
## 2 3 2 1  
## 3 3 1 1  
## 4 3 7 1  
## 5 3 1 1  
## 6 9 7 1

The properties of data are given,

str(diagnosis\_data)

## 'data.frame': 699 obs. of 10 variables:  
## $ classes : Factor w/ 2 levels "2","4": 1 1 1 1 1 2 1 1 1 1 ...  
## $ clump\_thickness : num 5 5 3 6 4 8 1 2 2 4 ...  
## $ uniformity\_of\_cell\_size : num 1 4 1 8 1 10 1 1 1 2 ...  
## $ uniformity\_of\_cell\_shape : num 1 4 1 8 1 10 1 2 1 1 ...  
## $ marginal\_adhesion : num 1 5 1 1 3 8 1 1 1 1 ...  
## $ single\_epithelial\_cell\_size: num 2 7 2 3 2 7 2 2 2 2 ...  
## $ bare\_nuclei : num 1 10 2 4 1 10 10 1 1 1 ...  
## $ bland\_chromatin : num 3 3 3 3 3 9 3 3 1 2 ...  
## $ normal\_nucleoli : num 1 2 1 7 1 7 1 1 1 1 ...  
## $ mitosis : num 1 1 1 1 1 1 1 1 5 1 ...

### Dataset 2

The outcome variable is the “classes” and it has following category of data Attribute Information:

1. ID number
2. Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area
5. smoothness (local variation in radius lengths)
6. compactness (perimeter^2 / area - 1.0)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. fractal dimension (“coastline approximation” - 1)

diagnosis\_data\_2 <- read.csv("~/Desktop/BreastCancer/Data/wdbc.data.csv", header = FALSE)  
  
phenotypes <- rep(c("radius", "texture", "perimeter", "area", "smoothness", "compactness", "concavity", "concave\_points", "symmetry", "fractal\_dimension"), 3)  
types <- rep(c("mean", "se", "largest\_worst"), each = 10)  
  
colnames(diagnosis\_data\_2) <- c("ID", "diagnosis", paste(phenotypes, types, sep = "\_"))  
head(diagnosis\_data\_2)

## 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\_largest\_worst  
## 1 0.03003 0.006193 25.38  
## 2 0.01389 0.003532 24.99  
## 3 0.02250 0.004571 23.57  
## 4 0.05963 0.009208 14.91  
## 5 0.01756 0.005115 22.54  
## 6 0.02165 0.005082 15.47  
## texture\_largest\_worst perimeter\_largest\_worst area\_largest\_worst  
## 1 17.33 184.60 2019.0  
## 2 23.41 158.80 1956.0  
## 3 25.53 152.50 1709.0  
## 4 26.50 98.87 567.7  
## 5 16.67 152.20 1575.0  
## 6 23.75 103.40 741.6  
## smoothness\_largest\_worst compactness\_largest\_worst  
## 1 0.1622 0.6656  
## 2 0.1238 0.1866  
## 3 0.1444 0.4245  
## 4 0.2098 0.8663  
## 5 0.1374 0.2050  
## 6 0.1791 0.5249  
## concavity\_largest\_worst concave\_points\_largest\_worst  
## 1 0.7119 0.2654  
## 2 0.2416 0.1860  
## 3 0.4504 0.2430  
## 4 0.6869 0.2575  
## 5 0.4000 0.1625  
## 6 0.5355 0.1741  
## symmetry\_largest\_worst fractal\_dimension\_largest\_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

str(diagnosis\_data\_2)

## 'data.frame': 569 obs. of 32 variables:  
## $ ID : int 842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...  
## $ diagnosis : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...  
## $ radius\_mean : num 18 20.6 19.7 11.4 20.3 ...  
## $ texture\_mean : num 10.4 17.8 21.2 20.4 14.3 ...  
## $ perimeter\_mean : num 122.8 132.9 130 77.6 135.1 ...  
## $ area\_mean : num 1001 1326 1203 386 1297 ...  
## $ smoothness\_mean : num 0.1184 0.0847 0.1096 0.1425 0.1003 ...  
## $ compactness\_mean : num 0.2776 0.0786 0.1599 0.2839 0.1328 ...  
## $ concavity\_mean : num 0.3001 0.0869 0.1974 0.2414 0.198 ...  
## $ concave\_points\_mean : num 0.1471 0.0702 0.1279 0.1052 0.1043 ...  
## $ symmetry\_mean : num 0.242 0.181 0.207 0.26 0.181 ...  
## $ fractal\_dimension\_mean : num 0.0787 0.0567 0.06 0.0974 0.0588 ...  
## $ radius\_se : num 1.095 0.543 0.746 0.496 0.757 ...  
## $ texture\_se : num 0.905 0.734 0.787 1.156 0.781 ...  
## $ perimeter\_se : num 8.59 3.4 4.58 3.44 5.44 ...  
## $ area\_se : num 153.4 74.1 94 27.2 94.4 ...  
## $ smoothness\_se : num 0.0064 0.00522 0.00615 0.00911 0.01149 ...  
## $ compactness\_se : num 0.049 0.0131 0.0401 0.0746 0.0246 ...  
## $ concavity\_se : num 0.0537 0.0186 0.0383 0.0566 0.0569 ...  
## $ concave\_points\_se : num 0.0159 0.0134 0.0206 0.0187 0.0188 ...  
## $ symmetry\_se : num 0.03 0.0139 0.0225 0.0596 0.0176 ...  
## $ fractal\_dimension\_se : num 0.00619 0.00353 0.00457 0.00921 0.00511 ...  
## $ radius\_largest\_worst : num 25.4 25 23.6 14.9 22.5 ...  
## $ texture\_largest\_worst : num 17.3 23.4 25.5 26.5 16.7 ...  
## $ perimeter\_largest\_worst : num 184.6 158.8 152.5 98.9 152.2 ...  
## $ area\_largest\_worst : num 2019 1956 1709 568 1575 ...  
## $ smoothness\_largest\_worst : num 0.162 0.124 0.144 0.21 0.137 ...  
## $ compactness\_largest\_worst : num 0.666 0.187 0.424 0.866 0.205 ...  
## $ concavity\_largest\_worst : num 0.712 0.242 0.45 0.687 0.4 ...  
## $ concave\_points\_largest\_worst : num 0.265 0.186 0.243 0.258 0.163 ...  
## $ symmetry\_largest\_worst : num 0.46 0.275 0.361 0.664 0.236 ...  
## $ fractal\_dimension\_largest\_worst: num 0.1189 0.089 0.0876 0.173 0.0768 ...

diagnosis\_data\_3 <- read.csv("~/Desktop/BreastCancer/Data/wpbc.data.csv", header = FALSE)  
  
colnames(diagnosis\_data\_3) <- c("ID", "outcome", "time", paste(phenotypes, types, sep = "\_"), "tumor\_size", "lymph\_node\_status")  
diagnosis\_data\_3[diagnosis\_data\_3 == "?"] <- NA  
head(diagnosis\_data\_3)

## ID outcome time radius\_mean texture\_mean perimeter\_mean area\_mean  
## 1 119513 N 31 18.02 27.60 117.50 1013.0  
## 2 8423 N 61 17.99 10.38 122.80 1001.0  
## 3 842517 N 116 21.37 17.44 137.50 1373.0  
## 4 843483 N 123 11.42 20.38 77.58 386.1  
## 5 843584 R 27 20.29 14.34 135.10 1297.0  
## 6 843786 R 77 12.75 15.29 84.60 502.7  
## smoothness\_mean compactness\_mean concavity\_mean concave\_points\_mean  
## 1 0.09489 0.1036 0.1086 0.07055  
## 2 0.11840 0.2776 0.3001 0.14710  
## 3 0.08836 0.1189 0.1255 0.08180  
## 4 0.14250 0.2839 0.2414 0.10520  
## 5 0.10030 0.1328 0.1980 0.10430  
## 6 0.11890 0.1569 0.1664 0.07666  
## symmetry\_mean fractal\_dimension\_mean radius\_se texture\_se perimeter\_se  
## 1 0.1865 0.06333 0.6249 1.8900 3.972  
## 2 0.2419 0.07871 1.0950 0.9053 8.589  
## 3 0.2333 0.06010 0.5854 0.6105 3.928  
## 4 0.2597 0.09744 0.4956 1.1560 3.445  
## 5 0.1809 0.05883 0.7572 0.7813 5.438  
## 6 0.1995 0.07164 0.3877 0.7402 2.999  
## area\_se smoothness\_se compactness\_se concavity\_se concave\_points\_se  
## 1 71.55 0.004433 0.01421 0.03233 0.009854  
## 2 153.40 0.006399 0.04904 0.05373 0.015870  
## 3 82.15 0.006167 0.03449 0.03300 0.018050  
## 4 27.23 0.009110 0.07458 0.05661 0.018670  
## 5 94.44 0.011490 0.02461 0.05688 0.018850  
## 6 30.85 0.007775 0.02987 0.04561 0.013570  
## symmetry\_se fractal\_dimension\_se radius\_largest\_worst  
## 1 0.01694 0.003495 21.63  
## 2 0.03003 0.006193 25.38  
## 3 0.03094 0.005039 24.90  
## 4 0.05963 0.009208 14.91  
## 5 0.01756 0.005115 22.54  
## 6 0.01774 0.005114 15.51  
## texture\_largest\_worst perimeter\_largest\_worst area\_largest\_worst  
## 1 37.08 139.70 1436.0  
## 2 17.33 184.60 2019.0  
## 3 20.98 159.10 1949.0  
## 4 26.50 98.87 567.7  
## 5 16.67 152.20 1575.0  
## 6 20.37 107.30 733.2  
## smoothness\_largest\_worst compactness\_largest\_worst  
## 1 0.1195 0.1926  
## 2 0.1622 0.6656  
## 3 0.1188 0.3449  
## 4 0.2098 0.8663  
## 5 0.1374 0.2050  
## 6 0.1706 0.4196  
## concavity\_largest\_worst concave\_points\_largest\_worst  
## 1 0.3140 0.1170  
## 2 0.7119 0.2654  
## 3 0.3414 0.2032  
## 4 0.6869 0.2575  
## 5 0.4000 0.1625  
## 6 0.5999 0.1709  
## symmetry\_largest\_worst fractal\_dimension\_largest\_worst tumor\_size  
## 1 0.2677 0.08113 5.0  
## 2 0.4601 0.11890 3.0  
## 3 0.4334 0.09067 2.5  
## 4 0.6638 0.17300 2.0  
## 5 0.2364 0.07678 3.5  
## 6 0.3485 0.11790 2.5  
## lymph\_node\_status  
## 1 5  
## 2 2  
## 3 0  
## 4 0  
## 5 0  
## 6 0

diagnosis\_data\_3[,3:35] <- apply(diagnosis\_data\_3[,3:35], 2, function(x) as.numeric(as.character(x)))  
dataset\_impute <- mice(diagnosis\_data\_3[,3:35], print = FALSE)

## Warning: Number of logged events: 25

diagnosis\_data\_3 <- cbind(diagnosis\_data\_3[, 2, drop = FALSE], mice::complete(dataset\_impute, 1))  
  
# how many recurring and non-recurring cases are there?  
summary(diagnosis\_data\_3$outcome)

## N R   
## 151 47

# plotting theme\*  
  
custom\_theme <- function(base\_size = 11,  
 base\_family = "",  
 base\_line\_size = base\_size / 170,  
 base\_rect\_size = base\_size / 170){  
 theme\_minimal(base\_size = base\_size,   
 base\_family = base\_family,  
 base\_line\_size = base\_line\_size) %+replace%  
 theme(  
 plot.title = element\_text(  
 color = rgb(25, 43, 65, maxColorValue = 255),   
 face = "bold",  
 hjust = 0),  
 axis.title = element\_text(  
 color = rgb(105, 105, 105, maxColorValue = 255),  
 size = rel(0.75)),  
 axis.text = element\_text(  
 color = rgb(105, 105, 105, maxColorValue = 255),  
 size = rel(0.5)),  
 panel.grid.major = element\_line(  
 rgb(105, 105, 105, maxColorValue = 255),  
 linetype = "dotted"),   
 panel.grid.minor = element\_line(  
 rgb(105, 105, 105, maxColorValue = 255),  
 linetype = "dotted",   
 size = rel(4)),   
   
 complete = TRUE  
 )  
}  
theme\_set(custom\_theme())

# function for PCA plotting  
library(pcaGoPromoter)

## Loading required package: ellipse

##   
## Attaching package: 'ellipse'

## The following object is masked from 'package:graphics':  
##   
## pairs

## Loading required package: Biostrings

## Loading required package: BiocGenerics

##   
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':  
##   
## clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,  
## clusterExport, clusterMap, parApply, parCapply, parLapply,  
## parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:mice':  
##   
## cbind, rbind

## The following objects are masked from 'package:stats':  
##   
## IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':  
##   
## anyDuplicated, append, as.data.frame, basename, cbind,  
## colMeans, colnames, colSums, dirname, do.call, duplicated,  
## eval, evalq, Filter, Find, get, grep, grepl, intersect,  
## is.unsorted, lapply, lengths, Map, mapply, match, mget, order,  
## paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,  
## Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,  
## table, tapply, union, unique, unsplit, which, which.max,  
## which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##   
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:base':  
##   
## expand.grid

## Loading required package: IRanges

## Loading required package: XVector

##   
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':  
##   
## strsplit

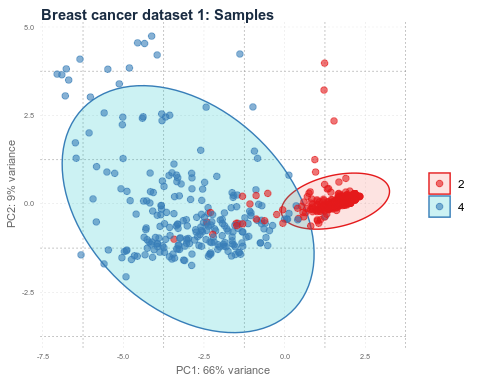
library(ellipse)  
  
pca\_func <- function(data, groups, title, print\_ellipse = TRUE) {  
   
 # perform pca and extract scores  
 pcaOutput <- pca(data, printDropped = FALSE, scale = TRUE, center = TRUE)  
 pcaOutput2 <- as.data.frame(pcaOutput$scores)  
   
 # define groups for plotting  
 pcaOutput2$groups <- groups  
   
 # when plotting samples calculate ellipses for plotting (when plotting features, there are no replicates)  
 if (print\_ellipse) {  
   
 centroids <- aggregate(cbind(PC1, PC2) ~ groups, pcaOutput2, mean)  
 conf.rgn <- do.call(rbind, lapply(unique(pcaOutput2$groups), function(t)  
 data.frame(groups = as.character(t),  
 ellipse(cov(pcaOutput2[pcaOutput2$groups == t, 1:2]),  
 centre = as.matrix(centroids[centroids$groups == t, 2:3]),  
 level = 0.95),  
 stringsAsFactors = FALSE)))  
   
 plot <- ggplot(data = pcaOutput2, aes(x = PC1, y = PC2, group = groups, color = groups)) +   
 geom\_polygon(data = conf.rgn, aes(fill = groups), alpha = 0.2) +  
 geom\_point(size = 2, alpha = 0.6) +   
 scale\_color\_brewer(palette = "Set1") +  
 labs(title = title,  
 color = "",  
 fill = "",  
 x = paste0("PC1: ", round(pcaOutput$pov[1], digits = 2) \* 100, "% variance"),  
 y = paste0("PC2: ", round(pcaOutput$pov[2], digits = 2) \* 100, "% variance"))  
   
 } else {  
   
 # if there are fewer than 10 groups (e.g. the predictor classes) I want to have colors from RColorBrewer  
 if (length(unique(pcaOutput2$groups)) <= 10) {  
   
 plot <- ggplot(data = pcaOutput2, aes(x = PC1, y = PC2, group = groups, color = groups)) +   
 geom\_point(size = 2, alpha = 0.6) +   
 scale\_color\_brewer(palette = "Set1") +  
 labs(title = title,  
 color = "",  
 fill = "",  
 x = paste0("PC1: ", round(pcaOutput$pov[1], digits = 2) \* 100, "% variance"),  
 y = paste0("PC2: ", round(pcaOutput$pov[2], digits = 2) \* 100, "% variance"))  
   
 } else {  
   
 # otherwise use the default rainbow colors  
 plot <- ggplot(data = pcaOutput2, aes(x = PC1, y = PC2, group = groups, color = groups)) +   
 geom\_point(size = 2, alpha = 0.6) +   
 labs(title = title,  
 color = "",  
 fill = "",  
 x = paste0("PC1: ", round(pcaOutput$pov[1], digits = 2) \* 100, "% variance"),  
 y = paste0("PC2: ", round(pcaOutput$pov[2], digits = 2) \* 100, "% variance"))  
   
 }  
 }  
   
 return(plot)  
   
}

library(gridExtra)

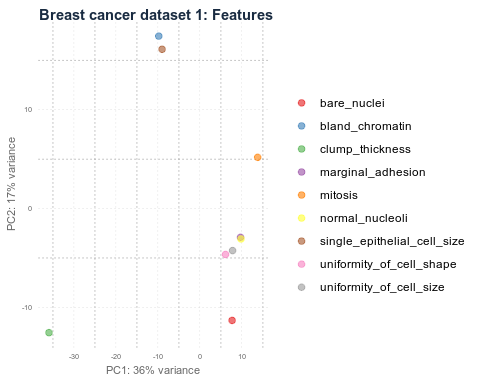
##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:BiocGenerics':  
##   
## combine

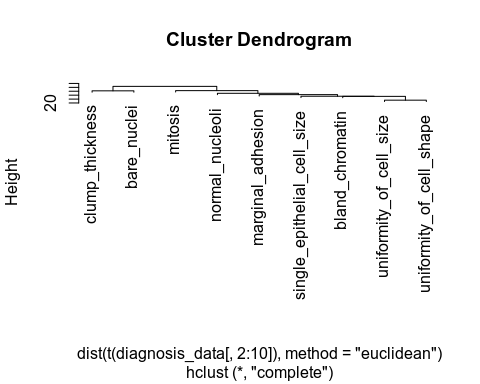
library(grid)  
p1 <- pca\_func(data = t(diagnosis\_data[, 2:10]), groups = as.character(diagnosis\_data$classes), title = "Breast cancer dataset 1: Samples")  
p2 <- pca\_func(data = diagnosis\_data[, 2:10], groups = as.character(colnames(diagnosis\_data[, 2:10])), title = "Breast cancer dataset 1: Features", print\_ellipse = FALSE)  
plot(p1)



plot(p2)



h\_1 <- hclust(dist(t(diagnosis\_data[, 2:10]), method = "euclidean"), method = "complete")  
plot(h\_1)



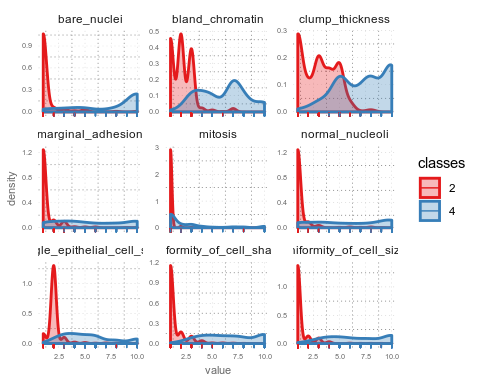
library(tidyr)

##   
## Attaching package: 'tidyr'

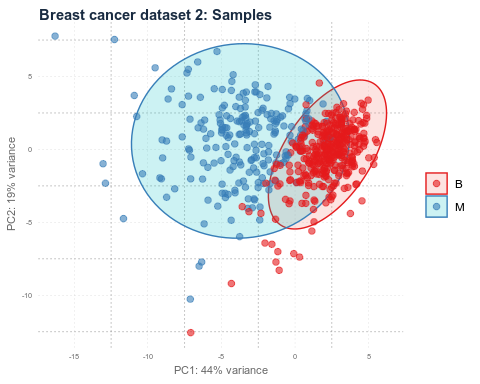
## The following object is masked from 'package:S4Vectors':  
##   
## expand

## The following object is masked from 'package:mice':  
##   
## complete

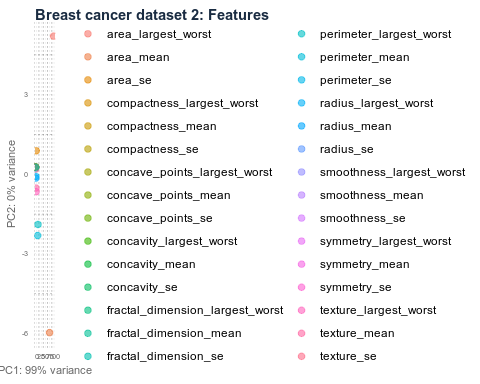
diagnosis\_data\_gather <- diagnosis\_data %>%  
 gather(measure, value, clump\_thickness:mitosis)  
  
ggplot(data = diagnosis\_data\_gather, aes(x = value, fill = classes, color = classes)) +  
 geom\_density(alpha = 0.3, size = 1) +  
 geom\_rug() +  
 scale\_fill\_brewer(palette = "Set1") +  
 scale\_color\_brewer(palette = "Set1") +  
 facet\_wrap( ~ measure, scales = "free\_y", ncol = 3)



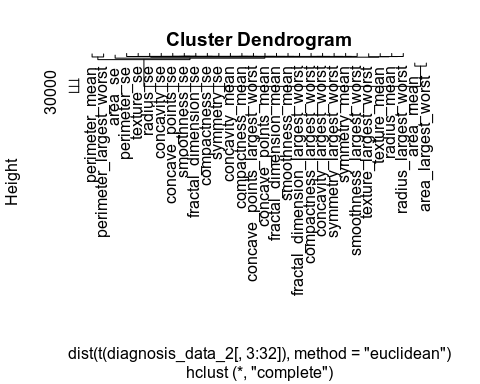
p1 <- pca\_func(data = t(diagnosis\_data\_2[, 3:32]), groups = as.character(diagnosis\_data\_2$diagnosis), title = "Breast cancer dataset 2: Samples")  
p2 <- pca\_func(data = diagnosis\_data\_2[, 3:32], groups = as.character(colnames(diagnosis\_data\_2[, 3:32])), title = "Breast cancer dataset 2: Features", print\_ellipse = FALSE)  
plot(p1)



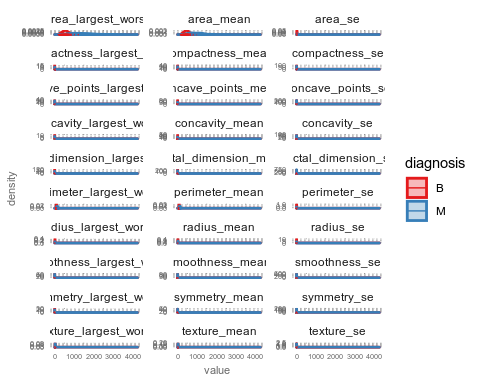
plot(p2)



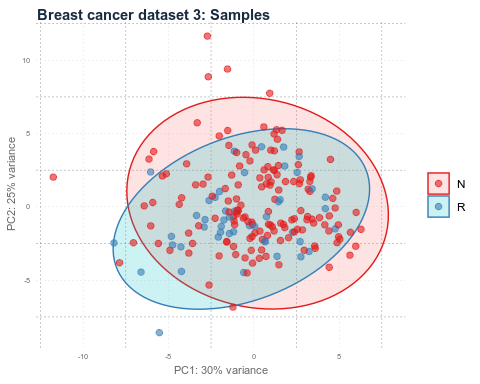
h\_2 <- hclust(dist(t(diagnosis\_data\_2[, 3:32]), method = "euclidean"), method = "complete")  
plot(h\_2)



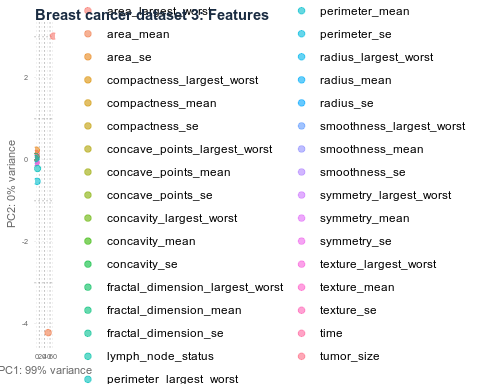
diagnosis\_data\_2\_gather <- diagnosis\_data\_2[, -1] %>%  
 gather(measure, value, radius\_mean:fractal\_dimension\_largest\_worst)  
  
ggplot(data = diagnosis\_data\_2\_gather, aes(x = value, fill = diagnosis, color = diagnosis)) +  
 geom\_density(alpha = 0.3, size = 1) +  
 geom\_rug() +  
 scale\_fill\_brewer(palette = "Set1") +  
 scale\_color\_brewer(palette = "Set1") +  
 facet\_wrap( ~ measure, scales = "free\_y", ncol = 3)



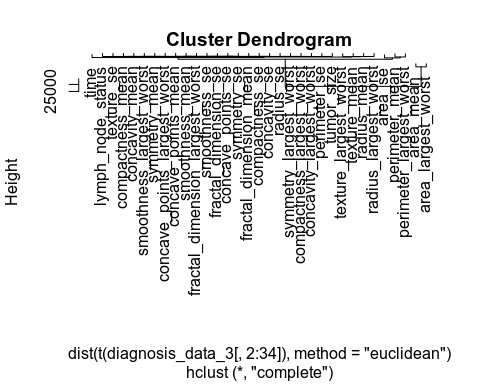
p1 <- pca\_func(data = t(diagnosis\_data\_3[, 2:34]), groups = as.character(diagnosis\_data\_3$outcome), title = "Breast cancer dataset 3: Samples")  
p2 <- pca\_func(data = diagnosis\_data\_3[, 2:34], groups = as.character(colnames(diagnosis\_data\_3[, 2:34])), title = "Breast cancer dataset 3: Features", print\_ellipse = FALSE)  
plot(p1)



plot(p2)



h\_3 <- hclust(dist(t(diagnosis\_data\_3[,2:34]), method = "euclidean"), method = "complete")  
plot(h\_3)



# parallel processing  
registerDoParallel()  
  
# prepare training scheme  
control <- trainControl(method = "repeatedcv", number = 10, repeats = 10)  
  
feature\_imp <- function(model, title) {  
   
 # estimate variable importance  
 importance <- varImp(model, scale = TRUE)  
   
 # prepare dataframes for plotting  
 importance\_df\_1 <- importance$importance  
 importance\_df\_1$group <- rownames(importance\_df\_1)  
   
 importance\_df\_2 <- importance\_df\_1  
 importance\_df\_2$Overall <- 0  
   
 importance\_df <- rbind(importance\_df\_1, importance\_df\_2)  
   
 plot <- ggplot() +  
 geom\_point(data = importance\_df\_1, aes(x = Overall, y = group, color = group), size = 2) +  
 geom\_path(data = importance\_df, aes(x = Overall, y = group, color = group, group = group), size = 1) +  
 theme(legend.position = "none") +  
 labs(  
 x = "Importance",  
 y = "",  
 title = title,  
 subtitle = "Scaled feature importance",  
 caption = "\nDetermined with Random Forest and  
 repeated cross validation (10 repeats, 10 times)"  
 )  
   
 return(plot)  
   
}

# train the model  
set.seed(27)  
imp\_1 <- train(classes ~ ., data = diagnosis\_data, method = "rf", preProcess = c("scale", "center"), trControl = control)  
p1 <- feature\_imp(imp\_1, title = "Breast cancer dataset 1")  
set.seed(27)  
imp\_2 <- train(diagnosis ~ ., data = diagnosis\_data\_2[, -1], method = "rf", preProcess = c("scale", "center"), trControl = control)  
p2 <- feature\_imp(imp\_2, title = "Breast cancer dataset 2")  
set.seed(27)  
imp\_3 <- train(outcome ~ ., data = diagnosis\_data\_3, method = "rf", preProcess = c("scale", "center"), trControl = control)  
p3 <- feature\_imp(imp\_3, title = "Breast cancer dataset 3")  
grid.arrange(p1, p2, p3, ncol = 3, widths = c(0.3, 0.35, 0.35))

