title: "HW 3" author: "Anindita Deb" date: "4/12/2022" output: pdf_document

####################Task 2 Modelling with Kmeans on simulated dataset### ########################### Simulated data set with 20 observations in each of################ ##################three classes (i.e. 60 observations total), and 50 variables followed by mean shift ##########and Principal Component Analysis### set.seed(100) target <- rep(c(1,2,3),20) data <- matrix(rnorm(60*50), ncol=50)

data[target==2,]= data[target==2,] - .6 data[target==3,]= data[target==3,] + .6

dimnames(data) <- list(rownames(data, do.NULL = FALSE, prefix = "row"),colnames(data, do.NULL = FALSE, prefix = "col")) library("cluster") library("bootcluster") library("fossil") library(ggpubr) library(factoextra) library(stats) library(ggplot2) data.fit <- prcomp(data,scale=FALSE) PCAdata <- as.data.frame(data.fit$x[, 1:2]) PCAdata <- cbind(PCAdata, target) colnames(PCAdata) <- c("PC1", "PC2", "target") percentVar <- round(100 * summary(iris.fit)$importance[2, 1:2], 0) ggplot(PCAdata) + aes(PC1, PC2, color = as.factor(target), shape = as.factor(target)) + geom_point(size = 2)+ xlab(paste0("PC1: ", percentVar[1], "% variance")) + ylab(paste0("PC2: ", percentVar[2], "% variance")) + ggtitle("Principal component analysis (PCA)") + theme(aspect.ratio = 1) plot(data.fit$x[,1:2], col=4-target, pch =19, xlab ="First principal component", ylab="Second principal component")

##############Modeling with K means for k=3 ##

Option nstart attempts 20 initial random centorids###

kmeans.fit <- kmeans(data, centers = 3, nstart = 20) table(kmeans.fit$cluster, target, dnn=c("Cluster","Class")) data_mod<-cbind(data[,c(1,2)],target,kmeans.fit$cluster) data_mod<-as.data.frame(data_mod) colnames(data_mod) <- c("col1", "col2", "target","KmeansCluster") data_mod$target<-as.factor(data_mod$target) data_mod$KmeansCluster<-as.factor(data_mod$KmeansCluster)

```
ggscatter( data_mod, x = "col1", y = "col2", color = "KmeansCluster", palette = "npg", ellipse = TRUE, ellipse.type = "convex", shape = "target", size = 1.5, legend = "right", ggtheme = theme_bw(), xlab = paste0("col1"), ylab = paste0("col2") ) + stat_mean(aes(color = "KmeansCluster"), size = 4)+ ggtitle("Kmeans clusters on col1 and col2")

################################################Modeling with K means for K=2############################################################################################# kmeans.fit <- kmeans(data, centers = 2, nstart = 20) table(kmeans.fit$cluster, target, dnn=c("Cluster","Class")) data_mod<-cbind(data[,c(1,2)],target,kmeans.fit$cluster) data_mod<-as.data.frame(data_mod) colnames(data_mod) <- c("col1", "col2", "target","KmeansCluster") data_mod$target<-as.factor(data_mod$target) data_mod$KmeansCluster<-as.factor(data_mod$KmeansCluster) ggscatter( data_mod, x = "col1", y = "col2", color = "KmeansCluster", palette = "npg", ellipse = TRUE, ellipse.type = "convex", shape = "target", size = 1.5, legend = "right", ggtheme = theme_bw(), xlab = paste0("col1"), ylab = paste0("col2") ) + stat_mean(aes(color = "KmeansCluster"), size = 4)+ ggtitle("Kmeans clusters on col1 and col2") ###############################Modeling with K means for K=4############################################################################################################## kmeans.fit <- kmeans(data, centers = 4, nstart = 20) table(kmeans.fit$cluster, target, dnn=c("Cluster","Class"))

###########################################Kmeans with Principal components with K =3 ############################################################# PCAdata <- as.data.frame(data.fit$x[, 1:2]) kmeans.fit <- kmeans(PCAdata, centers = 3, nstart = 20) summary(kmeans.fit) table(kmeans.fit$cluster, target, dnn=c("Cluster","Class")) PCAdata <- cbind(PCAdata, factor(target),factor(kmeans.fit$cluster)) colnames(PCAdata) <- c("PC1", "PC2", "target","KmeansCluster") ggscatter( PCAdata, x = "PC1", y = "PC2", color = "KmeansCluster", palette = "npg", ellipse = TRUE, ellipse.type = "convex", shape = "target", size = 1.5, legend = "right", ggtheme = theme_bw(), xlab = paste0("PC1: ", percentVar[1], "% variance"), ylab = paste0("PC2: ", percentVar[2], "% variance") ) + stat_mean(aes(color = KmeansCluster), size = 4)+ ggtitle("Kmeans clusters on PC1 and PC2") ############################Kmeans Modelling with K =3 on scaled data###########################################################################
#########################################Scaling the data################################################################################## data.scale <- scale(data, center = FALSE, scale = TRUE) kmeans.fit=kmeans(data.scale, 3, nstart =20) table(kmeans.fit$cluster, target, dnn=c("Cluster","Class")) ###First lets check if the data (data.scale) is scaled to Sd = 1. From below it appears like it is. We do notice the the columnwise means have shifted slightly.
```

for each variable/column, get the mean and sd before and after the scaling

```
check.col <- cbind(apply(data.scale,2,sd), apply(data,2,sd), apply(data.scale,2,mean), apply(data,2,mean)) colnames(check.col) <- c("Scaled \nColumn Sd","Original \nColumn Sd","Scaled \nColumn Mean","Original \nColumn Mean")
```

create a boxplot for the data to compare

```
boxplot(check.col, cex.axis=0.7)
```

##Lets examine the rowwise means to see how they compare after the scaling. The below plot shows a slight change in rowwise mean of each class however the general mean shift performed in part (a) of the question is very much kept intact. ###{r fig.width=8, fig.height=5}#####

for each row, get the mean before and after the scaling

```
check.row <- cbind(apply(data.scale,1,mean), apply(data,1,mean)) colnames(check.row) <- c("Scaled \nRow Mean","Original \nRow Mean") par(mfrow=c(1,3)) boxplot(check.row[target==1,], ylim=c(-1,1), main="Class 1") boxplot(check.row[target==2,], ylim=c(-1,1), main="Class 2") boxplot(check.row[target==3,], ylim=c(-1,1), main="Class 3")
```

####Now lets look at some data points. We can see that there has been a very small centering of the data points. However each class (or color) has its own center it is shifting into.

```
par(mfrow=c(1,1)) plot(data[,1:2], col =(4-target), pch=19, xlim=c(-3,3), ylim=c(-3,3)) points(data.scale[,1:2], col =(4-target), pch=1) legend(-3,3, c("Original","Scaled"), pch=c(19,1), cex=.8)
```

```
kmeans.fit <- kmeans(data.scale, centers = 3, nstart = 20) table(kmeans.fit$cluster, target, dnn=c("Cluster","Class")) data_mod<-cbind(data.scale[,c(1,2)],target,kmeans.fit$cluster) data_mod<-as.data.frame(data_mod) colnames(data_mod) <- c("col1", "col2", "target","KmeansCluster") data_mod$target<-as.factor(data_mod$target) data_mod$KmeansCluster<-as.factor(data_mod$KmeansCluster)
```

```
ggscatter( data_mod, x = "col1", y = "col2", color = "KmeansCluster", palette = "npg", ellipse = TRUE, ellipse.type = "convex", shape = "target", size = 1.5, legend = "right", ggtheme = theme_bw(), xlab = paste0("col1"), ylab = paste0("col2") ) + stat_mean(aes(color = "KmeansCluster"), size = 4)+ ggtitle("Kmeans clusters on col1 and col2")
```

```
############################### Task 3##############################################################################################
```

```
library("multtest") library("fpc") library("cluster") library("bootcluster") library("fossil") genedata<-read.csv("Ch12Ex13.csv",header=FALSE) sum(is.na(genedata)) is.null(genedata) d<-cor(genedata) hc <- hclust(d, method = "ave")
```

```
colnames(genedata)= paste("healthy_tissue", 1:20, sep = "") colnames(genedata[,21:40])=paste("diseased_tissue", 21:40, sep = "") rownames(mydata) = paste("Gene", 1:20, sep = "") genedata$ x<-str_extract(colnames(genedata), "healthy_tissue") colnames(genedata)[is.na(colnames(genedata))] <- paste("diseased_tissue", 21:40, sep = "") rownames(genedata) = paste("Gene", 1:1000, sep = "")
```

Pairwise correlation between tissue samples (columns)##

```
cols.cor <- cor(genedata, use = "pairwise.complete.obs", method = "pearson")
```

Pairwise correlation between rows (genes)###

```
rows.cor <- cor(t(genedata), use = "pairwise.complete.obs", method = "pearson")
```

Plotting the heatmap##

library("pheatmap") pheatmap(genedata, scale = "row", clustering_distance_cols = as.dist(1 - cols.cor), clustering_distance_rows = as.dist(1 - rows.cor[1:40,1:40]),angle_col=45,show_rownames = T, show_colnames = T) ####################Hierarchial Clustering with different distance linkage##
hclust.col <- hclust(as.dist(1-cols.cor)) hclust.row <- hclust(as.dist(1-rows.cor)) library("gplots") heatmap.2(as.matrix(genedata), scale = "row", col = bluered(100), trace = "none", density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row))

hclust.col <- hclust(as.dist(1-cols.cor),method="single") hclust.row <- hclust(as.dist(1-rows.cor),method="single") library("gplots") heatmap.2(as.matrix(genedata), scale = "row", col = bluered(100), trace = "none", density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row))

hclust.col <- hclust(as.dist(1-cols.cor),method="average") hclust.row <- hclust(as.dist(1-rows.cor),method="average") heatmap.2(as.matrix(genedata), scale = "row", col = bluered(100), trace = "none", density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row))

hclust.col <- hclust(as.dist(1-cols.cor),method="ward.D") hclust.row <- hclust(as.dist(1-rows.cor),method="ward.D") heatmap.2(as.matrix(genedata), scale = "row", col = bluered(100), trace = "none", density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row))

hclust.col <- hclust(as.dist(1-cols.cor),method="ward.D2") hclust.row <- hclust(as.dist(1-rows.cor),method="ward.D2") heatmap.2(as.matrix(genedata), scale = "row", col = bluered(100), trace = "none", density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row)) hclust.col <- hclust(as.dist(1-cols.cor),method="mcquitty") hclust.row <- hclust(as.dist(1-rows.cor),method="mcquitty") heatmap.2(as.matrix(genedata), scale = "row", col = bluered(100), trace = "none", density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row)) hclust.col <- hclust(as.dist(1-cols.cor),method="median") hclust.row <- hclust(as.dist(1-rows.cor),method="median") heatmap.2(as.matrix(genedata), scale = "row", col = bluered(100), trace = "none",

```
density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row) ) hclust.col <- hclust(as.dist(1-
cols.cor),method="centroid") hclust.row <- hclust(as.dist(1-rows.cor),method="centroid") heatmap.2(as.matrix(genedata), scale = "row", col
= bluered(100), trace = "none", density.info = "none", Colv = as.dendrogram(hclust.col), Rowv = as.dendrogram(hclust.row) )

########################################Task 4 - Hierarchial Random
Graphs#################################################################################

install.packages("igraphdata")

install.packages("igraph") library("igraphdata") library("igraph") data(package="igraphdata") ?fit_hrg data(karate)
###########################VIsulaizing the dataset########################################################
plot.igraph(karate, layout=layout.fruchterman.reingold, main='Karate Friends!', #vertex.label.dist=0.5,
vertex.label.color='black',
vertex.label.font=1,
vertex.label=V(karate)$name,
vertex.label.cex=0.75,
vertex.size=degree(karate)*1.5, edge.arrow.size=2 ) sort(degree(karate), decreasing = TRUE) average.path.length(karate)

shortPaths <- get.shortest.paths(karate, from="Actor 33") ##############Generating a random walk from a random Actor which is
Actor 33 in this case ######################

w <- random_walk(karate, start = "Actor 33", steps =1000) sort(table(w$name), decreasing = TRUE) probKarate <- table(w$name)/1000
########################Checking for influencers in the karate network#####################################

##################finding the actor who has the highest correlation with our drunken walk
probabilities########################## ########Let’s use some network centrality measures to see whether there any
differences################################## pr <- page.rank(karate) sort(pr$vector, decreasing = TRUE) nk <- V(karate)$name
cor(pr$vector[nk], probKarate[nk]) ec <- eigen_centrality(karate) sort(ec$vector, decreasing = TRUE) cor(ec$vector[nk], probKarate[nk]) hs
<- hub_score(karate) sort(hs$vector, decreasing = TRUE) cor(hs$vector[nk], probKarate[nk]) #######################Who’s the most
connected groups to each other? using Cliques######################################### karateCliques <- cliques(karate)
largeKarateCliques <- largest_cliques(karate) #########################Visulaizing the Kite
dataset######################################################### plot.igraph(kite, layout=layout.fruchterman.reingold,
```

main='Kite Friends!', #vertex.label.dist=0.5,
vertex.label.color='black',
vertex.label.font=1,
vertex.label=V(kite)$name,
vertex.label.cex=0.75,
vertex.size=degree(kite)*1.5, edge.arrow.size=2) average.path.length(kite) set.seed(123) x<-sample(E(karate),round(0.05*ecount(karate)))
karate_mod<-delete_edges(karate,x)

##Fit Hierarchcial Random Graphs on Karate Dataset

karate_mod_dendo <- fit_hrg(karate_mod) plot_dendrogram(karate_mod_dendo) { y=vertex_attr(karate)$label }

Predict missing edges

pred <- predict_edges(karate_mod) y<-as.data.frame(pred$edges)

> y<-as.data.frame(pred$edges) y[y$V1==3 & y$V2==29,] V1 V2 205 3 29 y[y$V1==19 & y$V2==33,] V1 V2 6 19 33 y[y$V1==1 & y$V2==20,] V1 V2 44 1 20 y[y$V1==27 & y$V2==30,] V1 V2 424 27 30

#####################################Rempved % percent of the edges on kite Dataset

##################Fit Hierarchcial Random Graphs on kite Dataset### set.seed(123) x<-sample(E(kite),round(0.05*ecount(kite))) kite_mod<-delete_edges(kite,x)

kite__dendo <- fit_hrg(kite) plot_dendrogram(kite__dendo) pred <- predict_edges(kite_mod) y<-as.data.frame(pred$edges) ###Randomly removing 15 % edges from karate and kite networks###################### ##Predicting the missing edges and checking

whether the edges are recovered########## set.seed(123) x<-sample(E(karate),round(0.15*ecount(karate))) karate_mod<-delete_edges(karate,x) pred <- predict_edges(karate_mod) y<-as.data.frame(pred$edges) y[y$V1==3 & y$V2==29,] y[y$V1==19 & y$V2==33,] y[y$V1=="H" & y$V2==20,] y[y$V1==27 & y$V2==30,] y[y$V1==9 & y$V2==31,] y[y$V1==16 & y$V2==20,] y[y$V1==16 & y$V2=="A",]
y[y$V1==9 & y$V2==33,] y[y$V1==32 & y$V2==33,] y[y$V1==3 & y$V2==4,] y[y$V1==28 & y$V2=="A",] y[y$V1==23 & y$V2=="A",] y[y$V1=="H" & y$V2==11,] set.seed(123) x<-sample(E(kite),round(0.15*ecount(kite))) kite_mod<-delete_edges(kite,x) pred <- predict_edges(kite_mod) y<-as.data.frame(pred$edges) y[y$V1==5 & y$V2==4,]
y[y$V1==8 & y$V2==7,] y[y$V1==9 & y$V2==7,] ###Randomly removing 40 % edges from karate and kite networks###################### ##Predicting the missing edges and checking whether the edges are recovered########## set.seed(123) x<-sample(E(karate),round(0.4*ecount(karate))) karate_mod<-delete_edges(karate,x) pred <- predict_edges(karate_mod) y<-as.data.frame(pred$edges) y[y$V1==3 & y$V2==29,] y[y$V1==19 & y$V2==33,] y[y$V1=="H" & y$V2==20,] y[y$V1==27 & y$V2==30,] y[y$V1==9 & y$V2==31,] y[y$V1==16 & y$V2==A,] y[y$V1==9 & y$V2==31,] y[y$V1==9 & y$V2==33,] y[y$V1==32 & y$V2==33,] y[y$V1==3 & y$V2==4,] y[y$V1==28 & y$V2=="A",] y[y$V1==23 & y$V2=="A",] y[y$V1=="H" & y$V2==11,]

set.seed(123) x<-sample(E(kite),round(0.4*ecount(kite))) kite_mod<-delete_edges(kite,x) pred <- predict_edges(kite_mod) y<-as.data.frame(pred$edges) y[y$V1==7 & y$V2==9,] y[y$V1==5 & y$V2==6,] y[y$V1==4 & y$V2==5,] y[y$V1==3 & y$V2==2,] y[y$V1==6 & y$V2==7,] y[y$V1==1 & y$V2==2,]