**Analisis No Supervisado (18/12/2022)**

* This document: [**https://tinyurl.com/ulao2he**](https://tinyurl.com/ulao2he)
* Data directory: [**tinyurl.com/bdhcf3s3**](http://tinyurl.com/bdhcf3s3)

**# =======================================================**

**Iremos copiando progresivamente al documento los ejemplos que vayamos usando en la sesión ….**

* **Ir instalando las librerías (adjunto código R)**
* **Ir descargando en una carpeta local de vuestro ordenador el directorio datos**
* **Actualizar las variables path\_source y path\_data de vuestro ordenador**

**#########################################################**

**# Non Supervised analysis**

**# Javier M. Elicegui**

**#########################################################**

**#============================ Initialization ===============================**

**{**

**rm(list=ls())**

**par(ask=F)**

**# indicar directorio carpeta local**

**path\_source <- "/…/Big\_Data/src"**

**path\_data <- "/…/Big\_Data/data\_course"**

**setwd(path\_data)**

**libs <- c("tidyverse","ggplot2","datasets","plyr", "ggmap","dplyr","RColorBrewer","vcd",**

**"factoextra", "corrplot", "reshape2", "lattice", "gridExtra","RgoogleMaps",**

**"htmlwidgets", "Hmisc", "SnowballC", "kohonen", "maptools","ggpubr","circlize",**

**"dummies","rgeos", "sp", "wordcloud", "deldir","NbClust","mixtools","utils",**

**"PerformanceAnalytics", "graphics", "FactoMineR", "ppcor", "cluster",**

**"arulesViz","arules","leaflet", "devtools","htmltools","mixtools","gplots" )**

**for (i in libs){**

**print(i)**

**if(!require(i, character.only = TRUE))**

**{ install.packages(i, dependencies=TRUE); library(i) }**

**}**

**library(devtools)**

**if(!require("ggbiplot", character.only = TRUE))**

**install\_github("vqv/ggbiplot")**

**}**

**# =================== Rare Distributions: Anscombe, DatasaurusDozen. ============================**

**{**

**# https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf**

**# https://www.autodeskresearch.com/publications/samestats (Dino)**

**# read**

**{**

**# https://en.wikipedia.org/wiki/Anscombe's\_quartet**

**df <- read.table(file.path(path\_data,'Anscombe.txt'), header=T,**

**stringsAsFactors = FALSE)**

**df %>% group\_by(case) %>% filter(!is.na(x)) %>%**

**dplyr::summarise(**

**n = n(),**

**mean\_x = mean(x),**

**variance\_x = var(x),**

**mean\_y = mean(y),**

**variance\_y = var(y),**

**corr\_ = cor(x,y)**

**)**

**}**

**# Points**

**ggplot(df, aes(x=x, y=y, color=case)) + geom\_point() + facet\_wrap( ~ case, ncol = 2)**

**# Histogram, Density**

**# https://en.wikipedia.org/wiki/Kernel\_density\_estimation**

**# contraste funciones densidad**

**ggplot(df, aes(x=y, y=..count..)) +**

**geom\_histogram(fill="cornsilk", colour="grey60", binwidth=1) +**

**geom\_density()**

**p1 <- ggplot(df, aes(x=x)) + geom\_density() + xlim(0, 20) + facet\_grid(case ~ .)**

**p2 <- ggplot(df, aes(x=y)) + geom\_density() + xlim(0, 20) + facet\_grid(case ~ .)**

**grid.arrange(p1, p2 , ncol=2)**

**# función densidad 2D**

**ggplot(filter(df, case=='A'), aes(x=x, y=y)) + geom\_point() +**

**stat\_density2d() + xlim(0,20) +ylim(0,15)**

**ggplot(filter(df, case=='A'), aes(x=x, y=y)) + geom\_point() +**

**stat\_density2d(aes(fill = ..level..), geom = "polygon") + xlim(0,20) +ylim(0,15)**

**# >>> to do: DatasaurusDozen.**

**# https://www.autodesk.com/research/publications/same-stats-different-graphs**

**# install.packages("datasauRus")**

**require(datasauRus) # datasaurus\_dozen**

**{**

**datasaurus\_dozen %>%**

**group\_by(dataset) %>%**

**dplyr::summarise(**

**mean\_x = mean(x),**

**mean\_y = mean(y),**

**std\_dev\_x = sd(x),**

**std\_dev\_y = sd(y),**

**corr\_x\_y = cor(x, y)**

**)**

**ggplot(datasaurus\_dozen, aes(x = x, y = y, colour = dataset))+**

**geom\_point()+**

**theme\_void()+**

**theme(legend.position = "none")+**

**facet\_wrap(~dataset, ncol = 3)**

**}**

**}**

**hade=TRUE)**

**# Chi-Square test**

**# http://www.sthda.com/english/wiki/chi-square-test-of-independence-in-r**

**# Null hypothesis (H0): rows and columns variables of the contingency table are independent.**

**# Alternative hypothesis (H1): row and column variables are dependent**

**xtabs(~ smoking, data = alzheimer) %>% prop.table %>% round(4)**

**xtabs(~ disease, data = alzheimer) %>% prop.table %>% round(4)**

**xtabs(~ smoking+disease, data = alzheimer) %>% prop.table %>% round(2)**

**sum(alz) # 538**

**#Expected: .368\*.0520\*538 = 10.3 (Alzheimer & < 10)**

**(chisq <- chisq.t# =========================== Maps ===============================================**

**{**

**# fuente: ggplot2: A backstage tour, Hadley Wickham**

**# houston crime**

**require(ggplot2)**

**require(reshape2)**

**require(RgoogleMaps)**

**require(ggmap)**

**require(grid)**

**# map 1**

**{**

**theft <- subset(crime, offense == "theft" & between(lat, 29,30.2) & lon > -95.8)**

**lonr <- range(theft$lon)**

**latr <- range(theft$lat)**

**# https://rpubs.com/jiayiliu/ggmap\_examples**

**# https://console.cloud.google.com/google/maps-apis/credentials?project=tutorial-168315**

**# h\_map <- GetMap.bbox(lonr, latr, size = c(640, 640), API\_console\_key=key)**

**h\_map <- get(load("crime\_map.rda"))**

**ggplot(theft, aes(lon, lat)) +**

**annotation\_custom((rasterGrob(h\_map$myTile,**

**width = unit(1,"npc"),**

**height = unit(1,"npc"))), lonr[1], lonr[2], latr[1], latr[2]) +**

**geom\_density2d(colour = "black") +**

**geom\_point(size=0.1, color="grey", alpha=0.2)**

**}**

**addLayersControl(baseGroups = names(esri),**

**options = layersControlOptions(collapsed = FALSE)) %>%**

**addMiniMap(tiles = esri[[3]], toggleDisplay = TRUE,**

**position = "bottomleft") %>%**

**htmlwidgets::onRender("**

**function(el, x) {**

**var myMap = this;**

**# map2**

**{**

**# https://leafletjs.com/examples/quick-start/**

**require(leaflet)**

**leaflet <- leaflet() %>% setView(0,0,3)**

**esri <- grep("^Esri", providers, value = TRUE)**

**for (provider in esri) {**

**leaflet <- leaflet %>% addProviderTiles(provider, group = provider)**

**}**

**leaflet %>%**

**myMap.on('baselayerchange',**

**function (e) {**

**myMap.minimap.changeLayer(L.tileLayer.provider(e.name));**

**})**

**}")**

**library(htmltools)**

**aux\_ <- 1:30**

**leaflet(theft) %>% addTiles() %>%**

**addMarkers(~lon[aux\_], ~lat[aux\_], label = ~htmlEscape(address[aux\_]))**

**}**

**# >>> to do: rape, burglary, murder ...**

**}**

**#======================= usefull representations ====================**

**{ # https://rpubs.com/hadley/ggplot-intro**

**# book: R graphics Cookbook (O'Reilly)**

**help(mpg)**

**ggplot(mpg, aes(x=hwy, y=..density..)) +**

**geom\_histogram(fill="cornsilk", colour="grey60", binwidth=1) +**

**geom\_density()**

**quantile(mpg$hwy, 0.1) # median(mpg$hwy)**

**ecdf(mpg$hwy)(20) # 38%**

**# Smoothed conditional means**

**# https://ggplot2.tidyverse.org/reference/geom\_smooth.html**

**qplot(displ, hwy, data=mpg, facets = . ~ year) + geom\_smooth(method="loess")**

**# Boxplots**

**# https://en.wikipedia.org/wiki/Box\_plot**

**# https://www.researchgate.net/figure/The-main-components-of-a-boxplot-median-quartiles-whiskers-fences-and-outliers\_fig6\_303779929**

**qplot(class, hwy, data = mpg, geom = 'boxplot')**

**qplot(class, hwy, data = mpg, geom = 'violin')**

**# Dot Plots**

**# http://www.statmethods.net/graphs/dot.html**

**help(mtcars)**

**aux\_ <- mtcars %>% rownames\_to\_column %>%**

**arrange(mpg) %>%**

**mutate(cyl = as.factor(cyl)) %>%**

**mutate(color = case\_when(**

**cyl==4 ~ "red",**

**cyl==6 ~ "blue",**

**cyl==8 ~ "darkgreen"))**

**dotchart(aux\_$mpg,labels=aux\_$rowname,cex=.7,groups= aux\_$cyl,**

**main="Gas Milage for Car Models\ngrouped by cylinder",**

**xlab="Miles Per Gallon", gcolor="black", color=aux\_$color)**

**# Scatter Plot**

**# http://www.statmethods.net/graphs/scatterplot.html**

**require("PerformanceAnalytics")**

**chart.Correlation(mtcars[,c(1,3,4,5,6)], histogram=TRUE, pch=19, method="pearson")**

**# https://rpkgs.datanovia.com/ggpubr/ easy-to-use functions**

**p <- mtcars %>% mutate(am=as.factor(am)) %>%**

**ggboxplot(x = "cyl", y = "mpg",**

**color = "am", palette =c("#00AFBB", "#E7B800"),**

**add = "jitter")#, shape = "gear")**

**my\_comparisons <- list( c("4", "6"), c("6", "8"), c("4", "8") )**

**p + stat\_compare\_means(comparisons = my\_comparisons, method="t.test")+ # Add pairwise comparisons p-value**

**stat\_compare\_means(label.y = 50)**

**# chord diagrams and complex heatmaps**

**# https://jokergoo.github.io/circlize\_book/book/a-complex-example-of-chord-diagram.html**

**# https://jokergoo.github.io/ComplexHeatmap-reference/book/more-examples.html**

**{**

**library(circlize)**

**set.seed(999)**

**mat = matrix(sample(18, 18), 3, 6)**

**rownames(mat) = paste0("S", 1:3)**

**colnames(mat) = paste0("E", 1:6)**

**mat**

**df = data.frame(from = rep(rownames(mat), times = ncol(mat)),**

**to = rep(colnames(mat), each = nrow(mat)),**

**value = as.vector(mat),**

**stringsAsFactors = FALSE)**

**df**

**chordDiagram(mat)**

**}**

**}**

**#======================== Visualize contingence tables ===========**

**{**

**# Mosaic Plot Example**

**# http://www.statmethods.net/advgraphs/mosaic.html**

**# https://cran.r-project.org/web/packages/vcd/vignettes/strucplot.pdf**

**# HairEyeColor**

**{**

**require(graphics)**

**require(corrplot)**

**require(factoextra)**

**require(vcd)**

**mosaicplot(HairEyeColor, shade = TRUE)**

**}**

**# alzheimer**

**{**

**data("alzheimer", package = "coin")**

**alz <- xtabs(~ smoking + disease, data = alzheimer)**

**mosaicplot(alz, shade = TRUE, las=2, main = "alzheimer")**

**assoc(alz, sest(alz))**

**(chisq$observed)**

**(round(chisq$expected,2))**

**(round(chisq$residuals, 3)) # Pearson residuals (obs - exp / sqrt(exp)**

**# Chi-Square graphic**

**sum(chisq$residuals^2)**

**curve(dchisq(x, df = 6), from = 0, to = 40)**

**pchisq(q=28, df=6, lower.tail=FALSE)**

**x\_vector <- seq(15, 40)**

**p\_vector <- dchisq(x\_vector, df = 6)**

**polygon(c(x\_vector, rev(x\_vector)), c(p\_vector, rep(0, length(p\_vector))),**

**col = adjustcolor('red', alpha=0.3), border = NA)**

**}**

**# housetasks**

**{**

**# housetasks http://www.sthda.com/english/wiki/chi-square-test-of-independence-in-r**

**data(housetasks)**

**file\_path <- "http://www.sthda.com/sthda/RDoc/data/housetasks.txt"**

**housetasks <- read.delim(file\_path, row.names = 1)**

**dt <- as.table(as.matrix(housetasks))**

**mosaicplot(dt, shade = TRUE, las=2, main = "housetasks")**

**assoc(head(dt, 5), shade = TRUE, las=3)**

**# other representation**

**chisq <- chisq.test(housetasks)**

**corrplot(chisq$residuals, is.cor = FALSE)**

**}**

**# >> to do: data(children)**

**# >>> to do: Titanic**

**titanic=get(load("titanic.raw.rdata"))**

**}**

**cor.test(mtcars$gear, mtcars$am)**

**chart.Correlation(mtcars[, c(10,9,4,5,6)], histogram=TRUE, pch=19, method="pearson")**

**# tests the H0 hypothesis that the samples came from a Normal distribution**

**shapiro.test(rnorm(1000, mean = 5, sd = 1))**

**# failure to reject the null (that the data are normal).**

**shapiro.test(runif(50, min = 2, max = 4))**

**# reject the null (that the data are normal).**

**shapiro.test(mtcars$mpg) # normality**

**plot(density(mtcars$mpg))**

**shapiro.test(mtcars$am) # no normality**

**plot(density(mtcars$am))**

**# table format**

**require("Hmisc")**

**flattenCorrMatrix <- function(cormat, pmat) {**

**ut <- upper.tri(cormat)**

**data.frame(**

**row = rownames(cormat)[row(cormat)[ut]],**

**column = rownames(cormat)[base::col(cormat)[ut]],**

**cor =(cormat)[ut],**

**p = pmat[ut]**

**)**

**}**

**res2 <- rcorr(as.matrix(mtcars[,1:7]))**

**flattenCorrMatrix(res2$r, res2$P)**

**# ============= Correlations ==============================**

**{**

**# spearman**

**df <- data.frame(IQ=c(106,86,100,101,99,103,97,113,112,110),**

**TV=c(7,0,27,50,28,29,20,12,6,17)) %>% arrange(IQ)**

**plot(df$IQ, df$TV, type='l')**

**df$rank1 <- rank(df$IQ)**

**df$rank2 <- rank(df$TV)**

**cor(df$IQ,df$TV, method='pearson')**

**cor(df$IQ,df$TV, method='spearman')**

**cor(df$rank1,df$rank2, method='pearson')**

**cor(df$IQ,df$TV, method='kendall')**

**# http://www.sthda.com/english/wiki/print.php?id=61**

**require(corrplot)**

**# Third Correlogram Example**

**mcor <- cor(mtcars)**

**round(mcor, digits=2)**

**corrplot(mcor, method="shade", shade.col=NA, tl.srt=45,**

**addCoef.col="black", order="AOE", type='lower')**

**cor(mtcars$gear, mtcars$am)**

**# partial correlation**

**{**

**# Relationship between weight and number of meals intake while controlling age**

**# http://www.css.cornell.edu/faculty/dgr2/teach/R/R\_corregr.pdf**

**require(ggm)**

**require(ppcor)**

**require(psych)**

**# short example**

**df <- data.frame(X = c(2,4,15,20), Y = c(1,2,3,4), Z = c(0,0,1,1))**

**mm1 <- lm(X~Z, df)**

**res1 <- mm1$residuals**

**mm2 <- lm(Y~Z,df)**

**res2 <- mm2$residuals**

**cor(res1,res2) # 0.9191 - partial correlation**

**cor(df$X, df$Y) # 0.9695 - correlation**

**corrplot(cor(df, method="pearson"),**

**method="shade", shade.col=NA, tl.srt=45,**

**addCoef.col="black", order="AOE", type='lower')**

**corrplot(ppcor::pcor(df, method="pearson")$estimate,**

**method="shade", shade.col=NA, tl.srt=45,**

**addCoef.col="black", order="AOE", type='lower')**

**# mtcars example**

**corrplot(cor(mtcars, method="pearson"),**

**method="shade", shade.col=NA, tl.srt=45,**

**addCoef.col="black", order="AOE", type='lower')**

**corrplot(ppcor::pcor(mtcars, method="pearson")$estimate,**

**method="shade", shade.col=NA, tl.srt=45,**

**addCoef.col="black", order="AOE", type='lower')**

**}**

**# no linear correlation**

**{**

**aux\_ <- data.frame(dg=1:365, x=sin(1:365\*pi/180), y=cos(1:365\*pi/180))**

**cor(aux\_$x, aux\_$y)**

**plot(aux\_$x, aux\_$y)**

**aux\_ <- data.frame(dg=1:365, x=sin(1:(365\*2)\*pi/180), y=sin(1:(365\*2)\*pi\*2/180))**

**cor(aux\_$x, aux\_$y)**

**plot(aux\_$x, aux\_$y)**

**}**

**# >>> to do: marks, states**

**data(marks) # algebra, statistics, mechanics, ...**

**states <- state.x77[,1:6] # Population, Income, Murder, ...**

**}**

**#================ Principal Components PCA ======================**

**{**

**# http://www.sthda.com/english/wiki/practical-guide-to-principal-component-methods-in-r**

**library(ggbiplot)**

**require(factoextra)**

**# simulation**

**{**

**n <- 500**

**set.seed(1)**

**aux1\_ <- rnorm(n, 15, 15)**

**aux2\_ <- rnorm(n, 5, 1)**

**df <- data.frame(d=aux1\_+3\*aux2\_+5, e=aux2\_+2\*aux1\_+8)**

**ggplot(df, aes(x = d, y = e)) +geom\_point(size=2, color='magenta')**

**cor(df$d, df$e)**

**df.pca <- prcomp(df, scale=TRUE)**

**ggplot(df.pca$x %>% as.data.frame, aes(x = PC1, y = PC2)) +geom\_point(size=2, color="blue")**

**cor(df.pca$x[,1], df.pca$x[,2]) %>% round(3)**

**}**

**# wine**

**{**

|  | **wine <-read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data",sep=",")** |
| --- | --- |

**wine.pca <- prcomp(wine, scale = TRUE)**

**ggbiplot(wine.pca, obs.scale = 1, var.scale = 1, # labels=rownames(wine),**

**groups = wine.class, ellipse = TRUE, circle = TRUE) +**

**scale\_color\_discrete(name = '') +**

**theme(legend.direction = 'horizontal', legend.position = 'top')**

**fviz\_screeplot(wine.pca, addlabels = TRUE, ylim = c(0, 50))**

**get\_eigenvalue(wine.pca)**

**}**

**# USArrests**

**{**

**# http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/118-principal-component-analysis-in-r-prcomp-vs-princomp/#package-for-pca-visualization**

**pca <- prcomp(USArrests, scale = TRUE)**

**fviz\_pca\_biplot(pca, repel = TRUE,**

**col.var = "#2E9FDF", # Variables color**

**col.ind = "#696969") # Individuals color**

**plot(pca)**

**}**

**# mtcars**

**{**

**# https://www.datacamp.com/community/tutorials/pca-analysis-r**

**mtcars.pca <- prcomp(mtcars[,-c(8,9)], center = TRUE,scale. = TRUE)**

**ggbiplot(mtcars.pca, labels=rownames(mtcars))**

**mtcars.country <- c(rep("Japan", 3), rep("US",4), rep("Europe", 7),rep("US",3),**

**"Europe", rep("Japan", 3), rep("US",4),**

**rep("Europe", 3), "US", rep("Europe", 3))**

**ggbiplot(mtcars.pca, ellipse=TRUE, labels=rownames(mtcars), groups=mtcars.country)**

**ggbiplot(mtcars.pca,ellipse=TRUE,choices=c(3,4),**

**labels=rownames(mtcars), groups=mtcars.country)**

**}**

**# >> to do: decathon, iris**

**# library(FactoMineR); data(decathlon)**

**}**

**# Correspondence Analysis**

**# http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/113-ca-correspondence-analysis-in-r-essentials/**

**{**

**require(FactoMineR)**

**res.ca <- CA(housetasks, graph = FALSE)**

**fviz\_screeplot(res.ca, addlabels = TRUE, ylim = c(0, 50))**

**fviz\_ca\_biplot(res.ca, repel = TRUE)**

**# other representation**

**chisq <- chisq.test(housetasks)**

**corrplot(chisq$residuals, is.cor = FALSE)**

**}**

**#================= CLusters =============================**

**{**

**require(cluster)**

**require(NbClust)**

**require(factoextra)**

**# 1D, 2D**

**{**

**# 1D**

**library(mixtools)**

**wait = faithful$waiting**

**hist(wait, freq=F)**

**lines(density(wait))**

**mixmdl = normalmixEM(wait)**

**plot(mixmdl,which=2)**

**lines(density(wait), lty=2, lwd=2)**

**# 2D**

**n\_clusters <- 2**

**km.res <- kmeans(faithful, n\_clusters, nstart = 25)**

**table(km.res$cluster)**

**plot(faithful, col = km.res$cluster, pch = 19, frame = FALSE,**

**main = "K-means with k = 2", cex=1)**

**points(km.res$centers, col = 5:7, pch = 8, cex = 3)**

**}**

**# distances**

**{**

**# https://r-snippets.readthedocs.io/en/latest/real\_analysis/metrics.html**

**# https://people.revoledu.com/kardi/tutorial/Similarity/index.html**

**# Mahalanobis distance: https://www.charlesgauvin.ca/post/distances-and-outlier-detection/**

**aux\_ <- matrix(**

**c(1,4,8,9,2,3,**

**9,4,1,2,4,7,**

**1,7,9,3,2,8,**

**2,1,4,7,8,9,**

**1,4,8,3,9,2,**

**3,7,8,6,5,9), nrow=6,ncol=6)**

**dist(aux\_, method = "minkowski", p = 2)**

**# text**

**a <- c('potato', 'tomotto', 'chips', 'baloon')**

**b <- c('car', 'chips', 'bird', 'salt')**

**jaccard <- function(a, b) {**

**intersection = length(intersect(a, b))**

**union = length(a) + length(b) - intersection**

**return (intersection/union)**

**}**

**jaccard(a, b)**

**}**

**# USArrest example**

**{**

**# Number clustering with different methods**

**# http://www.sthda.com/english/wiki/determining-the-optimal-number-of-clusters-3-must-known-methods-unsupervised-machine-learning**

**set.seed(123)**

**gap\_stat <- clusGap(scale(USArrests), FUN = kmeans, nstart = 25,**

**K.max = 10, B = 50)**

**plot(gap\_stat, frame = FALSE, xlab = "Number of clusters k")**

**set.seed(123)**

**km.res <- kmeans(scale(USArrests), 4, nstart = 25)**

**fviz\_cluster(km.res, USArrests) # frame.type = "t", frame.alpha = 0, frame.level = 0.7**

**# k-means clustering is highly sensitive to outliers**

**# A more robust algorithm is provided by PAM algorithm (Partitioning Around Medoids)**

**pam.res <- pam(scale(USArrests), 4)**

**fviz\_cluster(pam.res)**

**# Voronoi Diagrams**

**# http://flowingdata.com/2016/04/12/voronoi-diagram-and-delaunay-triangulation-in-r/**

**# http://letstalkdata.com/2014/05/creating-voronoi-diagrams-with-ggplot/**

**{**

**pca <- stats::prcomp(scale(USArrests), scale = FALSE, center = FALSE)**

**ind <- facto\_summarize(pca, element = "ind", result = "coord", axes = 1:2)**

**colnames(ind)[2:3] <- c("x", "y")**

**clustcent <- stats::aggregate(ind[, 2:3], by =**

**list(cluster = km.res$cluster), mean)**

**x<-clustcent$x**

**y<-clustcent$y**

**voronoi<-deldir(x,y, rw=c(min(ind$x), max(ind$x), min(ind$y), max(ind$y)))**

**last\_plot() + geom\_segment(**

**aes(x = x1, y = y1, xend = x2, yend = y2), size = 1,**

**data = voronoi$dirsgs, linetype = 1, color= "#FFB958")**

**}**

**}**

**# Hierarchical clustering**

**{**

**data(nutrient, package="flexclust")**

**res.hc <- eclust(scale(nutrient), "hclust") # compute hclust**

**fviz\_dend(res.hc, rect = TRUE) # dendrogam**

**# cluster USArrestes**

**require(fastcluster)**

**require(graphics)**

**hc <- hclust(dist(USArrests), "ave")**

**plot(hc)**

**plot(hc, hang = -1)**

**}**

**# >> ejercicio iris, animals, plantTraits**

**# https://people.sc.fsu.edu/~jburkardt/datasets/hartigan/hartigan.html datasets**

**{**

**df <- read.table('https://people.sc.fsu.edu/~jburkardt/datasets/hartigan/file02.txt',**

**skip = 26, header=T, stringsAsFactors = FALSE)**

**{**

**rownames(df) <- df$Name**

**df$Name <- NULL**

**res.hc <- eclust(scale(df), "hclust") # compute hclust**

**fviz\_dend(res.hc, rect = TRUE) # dendrogam**

**}**

**}**

**}**

**#================= Textos =============================**

**{**

**# http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know**

**# Load**

**require("tm")**

**require("SnowballC")**

**require("wordcloud")**

**require("RColorBrewer")**

**require(cld3)**

**detect\_language("this is a sample")**

**detect\_language("esto es un ejemplo")**

**detect\_language("Dies ist ein Beispiel")**

**filePath <- "http://www.sthda.com/sthda/RDoc/example-files/martin-luther-king-i-have-a-dream-speech.txt"**

**text <- readLines(filePath)**

**# Load the data as a corpus**

**docs <- Corpus(VectorSource(text))**

**toSpace <- content\_transformer(function (x , pattern ) gsub(pattern, " ", x))**

**docs <- tm\_map(docs, toSpace, "/")**

**docs <- tm\_map(docs, toSpace, "@")**

**docs <- tm\_map(docs, toSpace, "\\|")**

**# Convert the text to lower case**

**docs <- tm\_map(docs, content\_transformer(tolower))**

**# Remove numbers**

**docs <- tm\_map(docs, removeNumbers)**

**# Remove english common stopwords**

**docs <- tm\_map(docs, removeWords, stopwords("english"))**

**# Remove your own stop word**

**# specify your stopwords as a character vector**

**docs <- tm\_map(docs, removeWords, c("will","let"))**

**# Remove punctuations**

**docs <- tm\_map(docs, removePunctuation)**

**# Eliminate extra white spaces**

**docs <- tm\_map(docs, stripWhitespace)**

**# Text stemming**

**# docs <- tm\_map(docs, stemDocument)**

**dtm <- TermDocumentMatrix(docs)**

**m <- as.matrix(dtm)**

**v <- sort(rowSums(m),decreasing=TRUE)**

**d <- data.frame(word = names(v),freq=v)**

**head(d, 10)**

**set.seed(1234)**

**wordcloud(words = d$word, freq = d$freq, min.freq = 1,**

**max.words=200, random.order=FALSE, rot.per=0.35,**

**colors=brewer.pal(8, "Dark2"))**

**}**

**#==================== Association Rules ====================**

**{ # https://cran.r-project.org/web/packages/arules/vignettes/arules.pdf**

**# https://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf**

**# Rules Groceries**

**{**

**require("arules")**

**require("arulesViz")**

**data("Groceries")**

**inspect(Groceries[1:3])**

**(rules <- apriori(Groceries,**

**parameter=list(support=0.001, confidence=0.7)))**

**inspect(head(sort(rules, by ="lift"),3))**

**{**

**length(Groceries) # Total: 98352**

**# Support: % rhs over Total**

**19/9835 # 0.001932 count / Total**

**# Covorage: lhs over Total**

**0.002135 ... 0.002135\*9835 = 21**

**# Confidence:**

**0.001932/0.002135 o 19/21 # 0.9049 % Support / Covarage**

**itemFrequency(Groceries)["bottled beer"]\*9835 # "bottled beer" 792**

**792/9835 = 0.08053 # "bottled beer" / total**

**0.9048/ 0.08053 # 11.24 lift Confidence / % Total"bottled beer"**

**}**

**plot(rules, measure=c("support", "lift"),**

**shading="confidence")**

**inspect(head(rules, by="lift", 2))**

**(subrules <- rules[quality(rules)$confidence > 0.8])**

**plot(rules, method="grouped")**

**sel <- plot(rules, method="grouped", interactive=TRUE)**

**}**

**# Rules income Adults**

**{**

**data("AdultUCI")**

**dim(AdultUCI)**

**AdultUCI[1:2, ]**

**AdultUCI[["fnlwgt"]] <- NULL**

**AdultUCI[["education-num"]] <- NULL**

**AdultUCI[["age"]] <- ordered(cut(AdultUCI[["age"]],**

**c(15,25, 45, 65, 100)), labels = c("Young", "Middle-aged", "Senior", "Old"))**

**AdultUCI[["hours-per-week"]] <- ordered(cut(AdultUCI[["hours-per-week"]],**

**c(0, 25, 40, 60, 168)), labels = c("Part-time", "Full-time",**

**"Over-time", "Workaholic"))**

**AdultUCI[["capital-gain"]] <- ordered(cut(AdultUCI[["capital-gain"]],**

**c(-Inf, 0, median(AdultUCI[["capital-gain"]][AdultUCI[["capital-gain"]] > 0]), Inf)),**

**labels = c("None", "Low", "High"))**

**AdultUCI[["capital-loss"]] <- ordered(cut(AdultUCI[["capital-loss"]],**

**c(-Inf, 0, median(AdultUCI[["capital-loss"]][AdultUCI[["capital-loss"]] > 0]),**

**Inf)),**

**labels = c("none", "low", "high"))**

**(Adult <- as(AdultUCI, "transactions"))**

**inspect(Adult[1:3])**

**itemFrequencyPlot(Adult, support = 0.1, cex.names = 0.8)**

**rules <- apriori(Adult, parameter = list(support = 0.02, confidence = 0.6))**

**rules**

**rulesIncomeLarge <- subset(rules, subset = rhs %in% "income=small" )**

**inspect(head(sort(rulesIncomeLarge, by = "confidence"), n = 3))**

**# >>> buscar cosas curiosas (ej: relationship=Own-child)**

**}**

**# Titanic: https://rpubs.com/JanpuHou/283047**

**{**

**titanic.raw <- get(load("titanic.raw.rdata"))**

**rule <- apriori(titanic.raw,**

**# min support & confidence**

**parameter=list(minlen=3, supp=0.1, conf=0.7),**

**appearance = list(default="lhs", rhs=c("Survived=No","Survived=Yes")))**

**inspect(rule)**

**plot(rule, method="graph",**

**control=list(nodeCol="red", edgeCol="blue", type="items"))**

**plot(rule, method="grouped")**

**subrules2 <- head(sort(rule, by="lift"), 2)**

**plot(subrules2, method="paracoord", control=list(alpha=1, col=rainbow(4)))**

**}**

**}**