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

**# Llista de FUNCIONS PROPIES per: #**

**# Organitzar / simplificar l'analisi de dades #**

**# #**

**# Author: Jordi Real Gatius #**

**# E-mail: jordireal@gmail.com #**

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

**# Canvi local aviam que tal**

**# Funcions propies LLEPALI Project -------------------------------**

**# Jordi Real**

**# jordireal@gmail.com**

**list.of.packages <- c("compareGroups","survminer", "data.table","MatchIt","survival","dplyr","lubridate","purrr","stringr","readxl","Hmisc","knitr","SNPassoc","DiagrammeR","pROC","ResourceSelection")**

**new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]**

**if(length(new.packages)) install.packages(new.packages)**

**# Llibreries necessaries**

**library("data.table")**

**library("SNPassoc")**

**library("htmlwidgets")**

**library("compareGroups")**

**library("foreign")**

**library("lattice")**

**library("Hmisc")**

**# library("ggplot2")**

**library("pander")**

**library("readxl")**

**library("knitr")**

**library("data.table")**

**library("MatchIt")**

**library("survival")**

**library("dplyr")**

**# library("survminer")**

**library("purrr")**

**library("stringr")**

**library("tidyr")**

**#**

**#**

**# Canviar directori de treball subdirectori ------------**

**directori\_treball<-function(subdirectori,directori) {**

**# directori=c("C:/Users/Jordi/Google Drive",**

**# "C:/Users/usuari/Google Drive",**

**# "C:/Users/43728088M/Google Drive",**

**# "C:/Users/jreal/Google Drive",**

**# "D:/Google Drive")**

**# subdirectori="CIBERDEM/GEDAPS/Cohort\_RD"**

**# directori[file.exists(directori)] %>%**

**pp<-file.path(directori[file.exists(directori)],subdirectori)**

**setwd(pp)**

**}**

**#**

**# Etiquetar les variables de les dades #####**

**###**

**etiquetar<-function(d=dadestotal,taulavariables="variables\_R.xls",camp\_descripcio="descripcio") {**

**# d=dades**

**# taulavariables = conductor\_variables**

**# camp\_descripcio="descripcio"**

**# Llegir etiquetes i variables a analitzar ####**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**# selecciono els camps necessaris (camp i descripcio) i amb etiqueta**

**camp\_descripcio<-sym(camp\_descripcio)**

**variables<-variables %>% dplyr::select(camp,descripcio=!!camp\_descripcio)**

**# Els que no tenen etiquet assignar el mateix nom del camp**

**variables<-variables %>% mutate(descripcio=ifelse(descripcio=="0",camp,descripcio))**

**#**

**# Etiquetar variables**

**seleccio<-variables**

**camp<- as.vector(seleccio$camp) #**

**descripcio<- as.vector(seleccio$descripcio) #**

**### etiquetar variables seleccionades #**

**for (i in 1:length(descripcio)){if (any(colnames(d) == camp[i])) {Hmisc::label(d[[camp[i]]]) <- descripcio[i]}}**

**d**

**}**

**# Etiquetar valors ------------------------------------------**

**## Retorna Data frame etiquetat en funció d'un conductor ##**

**## dataframe dades, conductor\_variables**

**etiquetar\_valors<-function(dt=dades,variables\_factors=conductor\_variables,fulla="etiquetes",**

**camp\_etiqueta="etiqueta"){**

**# dt=dades**

**# variables\_factors=conductor\_variables**

**# fulla="etiquetes\_valors"**

**# camp\_etiqueta="etiqueta"**

**# Llegir conductor#**

**variables\_factors<-readxl::read\_excel(variables\_factors,sheet=fulla)**

**#**

**# Split**

**camp\_etiqueta<-sym(camp\_etiqueta)**

**k<-variables\_factors%>%dplyr::select(camp, valor,!!camp\_etiqueta)**

**pepe<-k %>% base::split(list(.$camp))**

**#**

**noms\_variables<-names(pepe)**

**num\_vars<-length(noms\_variables)**

**# Elimina espais en blanc de totes les variables factor / character (treu nivells) tot el data\_frame**

**dt[sapply(dt,is.factor)] <- lapply(dt[sapply(dt,is.factor)], trimws)**

**dt[sapply(dt,is.character)] <- lapply(dt[sapply(dt,is.character)], trimws)**

**for (i in 1:num\_vars) {**

**# i<-1**

**if (noms\_variables[i] %in% colnames(dt)) {**

**etiquetes\_valors<-pepe[[i]] %>% pull(!!camp\_etiqueta)**

**dt[noms\_variables[i]]<-lapply(dt[noms\_variables[i]],function(y) factor(y,levels=pepe[[i]]$valor,labels=etiquetes\_valors))**

**}**

**}**

**dt**

**}**

**#**

**# Etiquetar Taula ------------------**

**# Llanço taula i camp que vull etiquetar i cambia nom del camp en funció d'etiqueta**

**etiquetar\_taula<-function(taula=resumtotal,camp="variable",taulavariables="variables\_R.xls",camp\_descripcio="descripcio") {**

**# taula=T3.MI**

**# taulavariables=conductor\_variables**

**# camp="variables\_taula"**

**# camp\_descripcio="descripcio"**

**#### Llegir etiquetes i variables a analitzar ####**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**#**

**# Canviar nom de camp de variables al de la taula**

**colnames(variables)[colnames(variables)=="camp"] <- camp**

**# Canviar arguments per ser evaluats**

**camp\_eval<-sym(camp)**

**camp\_descripcio\_eval<-sym(camp\_descripcio)**

**# Canviar el format de la taula**

**taula<-taula %>% left\_join(dplyr::select(variables,c(!!camp\_eval,camp\_descripcio)),by=quo\_name(camp\_eval)) %>%**

**mutate(!!camp\_eval:=descripcio) %>%**

**dplyr::select(-descripcio)**

**}**

**# dades<-etiquetar(dades)**

**# dades<-etiquetar(dades,"variables\_R.xls")**

**# FORMULA A PARTIR DE VARIABLES----------------------**

**##### hi envio la columna de variables amb que vull generar la formula pel compare**

**formula=function(x="taula1",y="grup",eliminar=c("idp",y)) {**

**pepito<-paste("as.vector(variables[variables$",x,"==1,]$camp)[!as.vector(variables[variables$",x,"==1,]$camp)%in%eliminar]",sep="")**

**llistataula<-eval(parse(text=pepito))**

**y<-as.formula(paste(y, paste(llistataula, collapse=" + "), sep=" ~ "))**

**}**

**# FORMULA MILLORADA --------------------------**

**#**

**# Te en compte l'Ordre que està posada en el conductor taulavariables**

**#**

**#**

**formula\_compare=function(x="taula1",y="grup",elimina=c("IDP"),taulavariables="variables\_R.xls", dt="No") {**

**# x="table5"**

**# y="grup"**

**# taulavariables =conductor\_variables**

**# elimina=c("IDP")**

**# dt=dades**

**# 1. Llegir conductor analisis**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**# 2. DATA table filtrar ordenar llista de camps**

**polio<-data.table::data.table(variables)**

**x<-sym(x)**

**mua<-polio[camp!=elimina] %>%**

**dplyr::filter(!!x>0) %>%**

**dplyr::arrange(!!x) %>%**

**dplyr::select(camp) %>% as.vector()**

**# 1.2. Filtrar per variables que realment existeixen en la base de dades**

**if (is.data.frame(dt)) {mua<-mua %>% semi\_join(data.frame(camp=names(dades)),by="camp")}**

**# 3. Generar formula**

**# y<-as.formula(paste(y, paste(llista$camp, collapse=" + "), sep=" ~ "))**

**y<-as.formula(paste(y, paste(mua$camp, collapse=" + "), sep=" ~ "))**

**y**

**}**

**# Llistat de taules a partir de Llista de factors de Y's i em retorna una llista de taules -----**

**llista.compare.Ys<-function(dt=dades,llista.y=c("CODGLP1","CKDEPI\_cat2"),llista.x=c("canvi612.pes.perc","canvi612M.pes"),show.ratio=F,byrow=T,show.n=T,show.all=T,show.descr=T,digits=NA,digits.ratio=NA,hide.no = c('NA','No'),ref.no=NA){**

**# dt=dt.matched**

**# llista.y = c("event")**

**# llista.x=llistaPS**

**# show.ratio=F**

**# byrow=T**

**restab.llista<-list()**

**# 3. Generar formula**

**for (i in 1:length(llista.y)) {**

**# i<-1**

**restab.llista[[i]]<-as.formula(paste(llista.y[[i]], paste(llista.x, collapse=" + "), sep=" ~ ")) %>%**

**compareGroups(data=dt,include.miss = F,include.label=T,byrow = byrow,ref.no=ref.no) %>%**

**createTable(show.ratio = show.ratio , hide.no = hide.no, show.p.overall=T,show.n=show.n,show.all=show.all,show.descr=show.descr,digits=digits,digits.ratio=digits.ratio)**

**}**

**restab.llista**

**}**

**# Selector de Variables -------**

**#**

**selectorvariables=function(taula="table1",taulavariables="variables\_R.xls",dt=dadestotal) {**

**# taula = "dades\_imputacio2"**

**# taulavariables="variables\_v2.xls"**

**# dt=dades\_test**

**vector\_variables<-extreure.variables(taula=taula,taulavariables = taulavariables)**

**# Selecciono les que no existeixen en DT**

**variables.no.existeixen<-vector\_variables[!is.element(vector\_variables,names(dt))]**

**# Elimino les que no existeixen**

**vector\_variables<-vector\_variables[is.element(vector\_variables,names(dt))]**

**moco<-dt %>% dplyr::select\_at(vector\_variables)**

**message(paste0("Llista de variables que no existeixen en el dataset:",paste0(variables.no.existeixen ,collapse = ", ")))**

**moco**

**}**

**# Extreure.Variables: Selector de variables TAULA DE--------**

**#**

**extreure.variables=function(taula="table1",taulavariables="variables\_R.xls") {**

**# taula="dates"**

**# taulavariables = conductor\_variables**

**#### Llegir etiquetes i variables a analitzar ####**

**variables <- data.frame(readxl::read\_excel(taulavariables))**

**variables[is.na(variables)]<- 0**

**taula<-rlang::sym(taula)**

**# filtratge**

**kk<-variables %>% dplyr::filter(!!taula>0) %>% dplyr::arrange(!!taula) %>% dplyr::select(camp) %>% as.vector()**

**kk<-as.vector(kk[[1]])**

**purrr::set\_names(kk,kk)**

**}**

**# factoritzar NO.Yes ------------------**

**########## factoritzar NO.Yes llista de variables "factor" situades a la taulavariables camp=factor**

**factoritzar.NO.YES<-function(dt=dadesDF,columna="factor",taulavariables="variables\_FELIPE.xls"){**

**# dt=dades**

**# columna="factor.YESNO"**

**# taulavariables=conductor\_variables**

**# Extreure variables**

**x<-extreure.variables(taula=columna,taulavariables=taulavariables)**

**# Seleccionar només variables que estan en dt**

**if (!x[!x%in%names(dt)] %>% length()<1) {print("No existeixen en dt:");print(x[!x%in%names(dt)])}**

**# Selecciono nomes les vars en bdades**

**x<-x[x%in%names(dt)]**

**### Factoritzar-les**

**dt[x]<-lapply(dt[x],function(y) factor(y,levels=c(0,1), labels=c("No","Yes")))**

**dt**

**}**

**# factoritzar vector ------------**

**# factoritzar una llista de variables donades unes dades i un vector de variables**

**factoritzar<-function(dt=dades,variables=c("grup","situacio")) {**

**# dt=dades**

**# variables=c("grup","situacio","kk","sexe")**

**# Només si variable existeix la variable en dt**

**variables<-variables[variables %in% names(dades)]**

**factoritzacio<-function(x) {if (!is.factor(x)) x<-as.factor(x) else x<-x}**

**dt<-dt %>% mutate\_at(variables,factoritzacio)**

**}**

**# Recodifico EN FUNCIÓ DE UN CAMP -------------------**

**### RETORNA DADES AMB RECODIFICACIÓ**

**# Recodifico EN FUNCIÓ DE de llista de camps -------------------**

**### RETORNA DADES AMB RECODIFICACIÓ**

**recodificar<-function(dt=dades,taulavariables="VARIABLES.xls",criteris="recode1",missings=F,prefix=NA){**

**# dt=dades**

**# taulavariables = conductor\_variables**

**# criteris = "recodes"**

**# missings=F**

**## Llegeix criteris de variables**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**criteris\_sym<-sym(criteris)**

**variables<-variables %>% dplyr::filter(!!criteris\_sym!=0)**

**## 0. Filtro taula variables només variables implicades en el filtre i el genero**

**caracter\_quartil<-"Q"**

**maco<-variables %>%**

**dplyr::select(camp,criteris) %>%**

**filter(!str\_detect(eval(parse(text=criteris)), caracter\_quartil))**

**## Generar recodificació en base info**

**maco\_lista<-maco %>% base::split(list(.$camp))**

**num\_recodes<-length(maco\_lista)**

**# Assignar a primer element (A partir d'aquí fer un for)**

**for (i in 1:num\_recodes) {**

**# i<-2**

**maco<-maco\_lista[[i]]**

**mamon<-stringr::str\_split(maco[criteris],"/") %>%**

**unlist() %>%**

**as.numeric()**

**mamon<-c(-Inf,mamon,Inf)**

**##### Fer la recodificació en base el rang generat**

**nomcamp<-maco["camp"] %>% as.character()**

**nomrecode<-paste0(nomcamp,".cat",length(mamon))**

**if (!is.na(prefix)) {nomrecode<-paste0(nomcamp,".cat",prefix,length(mamon)) }**

**# Si la variables ja existeix la elimino i la sobrescric**

**if (nomrecode%in%names(dt)) {dt<-dt %>% select\_(paste0("-",nomrecode))}**

**dt<-dt %>% mutate\_(camp=nomcamp)**

**dt<-dt %>% mutate(popes=cut(camp,breaks = mamon))**

**# Si missings --> generar a una categoria missing**

**if (missings==T) {dt<-missings\_to\_level(dt,"popes")}**

**colnames(dt)[colnames(dt)=="popes"] <- nomrecode**

**dt<-dt %>% dplyr::select(-camp)**

**print(paste0("Generada: ",nomrecode))**

**}**

**dt**

**}**

**# Retorna objecte Surv en dt a partir de dades (dt), event("20150531" / NA), dtindex(Date), dtsortida(20171231),**

**generar\_Surv<-function(dt,event,dtindex="dtindex",dtsortida="sortida"){**

**# dt=dades\_dt**

**# event="DG.MCV"**

**# dtindex="dtindex"**

**# dtsortida="data\_sortida"**

**x<-sym(event)**

**dtindex<-sym(dtindex)**

**sortida<-sym(dtsortida)**

**if(class(dt[[x]])!="Date" & class(dt[[sortida]])!="Date") {**

**temp<-dt %>% dplyr::select(!!dtindex,!!x,!!sortida) %>%**

**mutate(**

**event=case\_when(as.Date(as.character(!!x),"%Y%m%d")>0~1,**

**is.na(!!x)~0),**

**data\_final=case\_when(as.Date(as.character(!!x),"%Y%m%d")>0~as.Date(as.character(!!x),"%Y%m%d"),**

**is.na(!!x)~as.Date(as.character(!!sortida),"%Y%m%d")))**

**}**

**if(class(dt[[x]])=="Date" & class(dt[[sortida]])=="numeric") {**

**temp<-dt %>% dplyr::select(!!dtindex,!!x,!!sortida) %>%**

**mutate(**

**event=case\_when(!!x>0~1,**

**is.na(!!x)~0),**

**data\_final=case\_when(!!x>0~!!x,**

**is.na(!!x)~as.Date(as.character(!!sortida),"%Y%m%d")))**

**}**

**temp<- temp %>% mutate(temps=(data\_final-dtindex) %>% as.numeric())**

**# Genero l'objecte Surv**

**temp$event\_surv<-Surv(temp$temps,temp$event)**

**# Selecciono i renombro**

**nom\_surv=paste0(event,".surv")**

**temp<-temp %>% dplyr::select(event\_surv)**

**colnames(temp)=nom\_surv**

**temp**

**}**

**# missing\_to\_level (Recodifica variable amb una categoria missing) -------**

**missings\_to\_level<-function(dades,variable="popes") {**

**# dades=temp**

**# variable="val\_CKDEPI.cat5"**

**# Subset columnes de d**

**d\_temp<-dades %>% select\_("temp"=variable)**

**# names(dt)[names(dt)==variable]<-"variable\_temporal"**

**# dt<-dt %>% rename\_("variable\_temporal\_provisional"=variable)**

**levels\_nous <- levels(d\_temp$temp)**

**levels\_nous[length(levels\_nous) + 1] <- "None"**

**d\_temp$temp<-factor(d\_temp$temp,levels = levels\_nous)**

**d\_temp$temp[is.na(d\_temp$temp)]<-"None"**

**#**

**dades <- dades %>% select\_(paste0("-",variable))**

**# Canviar el nom al origen**

**names(d\_temp)[names(d\_temp) == "temp"] <- variable**

**dades <-cbind(dades,d\_temp)**

**}**

**# formula COX ajustat per event="Yes" -------------**

**#**

**### incorpora efecte cluster**

**### incorpora variables a evaluar a=Llista de variables a avaluar ###**

**formulaCOX=function(x="v.ajust",event="event",temps="temps",elimina="",cluster="",a="",taulavariables="variables.xls",codievent='1') {**

**variables <- data.frame(readxl::read\_excel(taulavariables))**

**variables[is.na(variables)]<- 0**

**variables<-variables %>% arrange\_(x)**

**pepito<-paste("as.vector(variables[variables$",x,">0,]$camp)[!as.vector(variables[variables$",x,">0,]$camp)%in%c('idp')]",sep="")**

**llistataula<-eval(parse(text=pepito))**

**if (a!="") llistataula<-c(llistataula,a)**

**# resposta<-paste("Surv(",temps,", as.integer(",event,"=='Si'))")**

**# resposta<-paste("Surv(",temps,", as.integer(",event,"=='Yes'))")**

**resposta<-paste0("Surv(",temps,", as.integer(",event,"=='",codievent,"'))")**

**#**

**if (cluster!="") kk<-paste(paste(llistataula,collapse=" + "),paste("cluster(",cluster,")",sep=""),sep="+")**

**if (cluster=="") kk<-paste(llistataula,collapse=" + ")**

**#**

**# y<-as.formula(paste(resposta, paste(llistataula, collapse=" + "), sep=" ~ "))**

**if (sum(elimina==llistataula)>0) y<-as.formula(paste(paste(resposta, kk , sep=" ~ "),elimina,sep=" - "))**

**if (sum(elimina==llistataula)==0) y<-as.formula(paste(resposta, kk , sep=" ~ "))**

**#**

**y**

**}**

**# Retorna Ngran, Events, coef, HR, IC95, IC95, se.coef, p ---------**

**HRadj=function(x="v.ajust",event="EV.INSUF\_CARD",t="tmp\_insuf\_card",e="",c="",d=dadesDF,taulavariables="variables.xls",codievent='1') {**

**# x="v.ajust"**

**# event="exitusCV"**

**# t="temps\_seguiment"**

**# d=dades**

**# taulavariables = conductor\_variables**

**# e=""**

**# c=""**

**# codievent='Si'**

**pepito<-paste("sum(d$",t,")",sep="")**

**PT<-eval(parse(text=pepito))**

**if (c=="") posicio\_p=5**

**if (c!="") posicio\_p=6**

**result=tryCatch({**

**pp<-coxph(formulaCOX(x=x,event=event,temps=t,elimina=e,cluster=c,taulavariables = taulavariables,codievent=codievent),data=d)**

**cbind(PT.Year=PT/365.25,**

**N=pp$n,**

**EVENTS=pp$nevent,**

**coef=summary(pp)$coef[1,1],**

**HR=summary(pp)$coef[1,2],**

**IC951=summary(pp)$conf.int[1,3],**

**IC952=summary(pp)$conf.int[1,4],**

**se.coef=summary(pp)$coef[1,3],**

**p=summary(pp)$coef[1,posicio\_p])}**

**,error = function(e) {**

**cbind(PT.Year=PT/365.25,**

**N=0,**

**EVENTS=0,**

**coef=NA,**

**HR=NA,**

**IC951=NA,**

**IC952=NA,**

**se.coef=NA,**

**p=NA)})**

**result**

**}**

**# HRestratificats ----------------------**

**### FUNCIiÓ QUE LLANÇO event, temps adjusted i em retorna un data frame amb tot global+ estratificat ###**

**### ENVIO exitus, temps i dades i em retorna data frame amb estratificats**

**#### camp estratificat conte variables estratificades tipo="v.ajust" / "crude"**

**HRestratificats<-function(event="exitus",t="temps",tipo="v.ajust",c="",taulavariables='variables.xls') {**

**HRestratificats=data.frame()**

**outDf<-data.frame(Subgroup="Total",HRadj(x=tipo,event=event,t=t,d=dades,c=c))**

**variables2 <- data.frame(readxl::read\_excel(taulavariables))**

**variables2[is.na(variables2)]<- 0**

**# row.names(outDf)<-label(dades$exitus)**

**row.names(outDf)<-eval(parse(text=paste("Hmisc::label(dades$",event,")",sep="")))**

**HRestratificats <-rbind(HRestratificats,outDf)**

**N<-length(variables2[variables2$estrat==1,]$camp)**

**for (i in 1:N) {**

**outDf <-ddply(dades, variables2[variables2$estrat==1,]$camp[i], function(df) HRadj(x=tipo,event=event,t=t,d=df,c=c))**

**row.names(outDf)<-c(paste(Hmisc::label(eval(parse(text=paste("dades$",names(outDf)[1],sep="")))),"Adj1",sep=""),**

**paste(Hmisc::label(eval(parse(text=paste("dades$",names(outDf)[1],sep="")))),"Adj2",sep=""))**

**names(outDf)[1]<-paste("Subgroup")**

**HRestratificats <-rbind(HRestratificats,outDf)**

**}**

**# retorna**

**return(HRestratificats)**

**}**

**# Formula.LOGIT segons LLISTA DE VARIABLES D'AJUST #######################**

**# hi envio la columna de variables amb que vull generar la formula pel compare**

**##### x= variables d'ajust / y = resposta / eliminar / a = Avaluar**

**formula.LOGIT=function(x="taula1",y="resposta",eliminar=c("IDP"), a="",taulavariables='variables.xls') {**

**# x="regicor\_alone"**

**# y="event"**

**# taulavariables = conductor\_variables**

**# eliminar=c("IDP")**

**# a=""**

**# Llegir variables**

**variables <- data.frame(readxl::read\_excel(taulavariables))**

**variables[is.na(variables)]<- 0**

**x\_sym<-sym(x)**

**llistataula<-variables %>%**

**dplyr::filter(!!x\_sym>0) %>%**

**dplyr::arrange(!!x\_sym) %>%**

**dplyr::filter("camp"!=eliminar) %>%**

**pull(camp)**

**if (a!="") llistataula<-c(a,llistataula)**

**y<-as.formula(paste(y, paste(llistataula, collapse=" + "), sep=" ~ "))**

**}**

**# Formula segos LLISTA DE VARIABLES D'AJUST #######################**

**##### hi envio la columna de variables amb que vull generar la formula pel compare**

**##### x= variables d'ajust / y = resposta / eliminar / a = Avaluar**

**formula.text=function(x="taula1",y="resposta",eliminar=c("IDP"), a="",taulavariables='variables.xls') {**

**variables <- data.frame(readxl::read\_excel(taulavariables))**

**variables[is.na(variables)]<- 0**

**x<-sym(x)**

**variables<-variables %>%**

**dplyr::filter(!!x>0) %>%**

**dplyr::arrange(!!x)**

**pepito<-paste("as.vector(variables[variables$",x,">0,]$camp)[!as.vector(variables[variables$",x,">0,]$camp)%in%eliminar]",sep="")**

**llistataula<-eval(parse(text=pepito))**

**# if (a!="") llistataula<-c(llistataula,a)**

**if (a!="") llistataula<-c(a,llistataula,a)**

**y<-paste(y, paste(llistataula, collapse=" + "), sep=" ~ ")**

**}**

**# formula\_vector(vector,y) ##########**

**formula\_vector<-function(vector=c("sex","age"),y="y",logit=F){**

**if (!logit) {formula=as.formula(paste(y, paste(vector, collapse=" + "), sep=" ~ "))}**

**if (logit) {formula=paste0("as.factor(",y,")~ ", paste(vector, collapse=" + ")) %>% as.formula()}**

**formula**

**}**

**# OR.ajustats(x,ajust,y) ###########**

**#**

**OR.ajustats=function(x="lipos",ajust="V.ajust",y="prediabetis",d=dadestotal,taulavariables='variables.xls') {**

**#**

**# d=dades**

**# taulavariables = "VARIABLES.xls"**

**# x="lipos"**

**# ajust="v.ajust"**

**# y="Prediabetes"**

**# d=dades**

**# taulavariables="VARIABLES.xls"**

**# x="lipos2"**

**# ajust="v.ajust"**

**# y="Prediabetes"**

**#**

**variables <- data.frame(readxl::read\_excel(taulavariables))**

**variables[is.na(variables)]<- 0**

**# inicialitzar**

**num<-paste("length(variables[variables$",x,">0,]$camp)",sep="")**

**num<-eval(parse(text=num))**

**ORadj<-matrix(data=NA,ncol=4,nrow = num)**

**# noms de columnes en matriu ORadj**

**listvariables<-paste("list(variables[variables$",x,">0,]$camp[1:",num,"],c('OR','Linf','Lsup','p valor'))",sep="")**

**dimnames(ORadj)<-eval(parse(text=listvariables))**

**#**

**#### extrec la variable que vull ajustar**

**xtext<-paste("variables[variables$",x,">0,]",sep="")**

**#**

**## inicio bucle amb totes les variables que vull ajustar**

**for (i in 1:num) {**

**# i=1**

**xeval<-eval(parse(text=xtext))$camp[i]**

**# genero la forumla del model**

**# myFormula<-paste(y,"~",xeval,"+",variables.ajust(x=ajust),sep="")**

**myFormula<-formula.LOGIT(x=ajust,y=y,eliminar="",a=xeval)**

**# ajusto models**

**model<-glm(formula= myFormula, family = binomial, data=d)**

**model**

**# extrec Coeficients dels models i IC i coloco dins de ORadj**

**lolo<-cbind(OR=exp(summary.glm(model)$coef[,1]),Linf=exp(summary.glm(model)$coef[,1]-1.96\*summary.glm(model)$coef[,2]),Lsup=exp(summary.glm(model)$coef[,1]+1.96\*summary.glm(model)$coef[,2]),p\_value=summary.glm(model)$coef[,4])**

**ORadj[i,]<-cbind(OR=exp(summary.glm(model)$coef[2,1]),Linf=exp(summary.glm(model)$coef[2,1]-1.96\*summary.glm(model)$coef[2,2]),Lsup=exp(summary.glm(model)$coef[2,1]+1.96\*summary.glm(model)$coef[2,2]),p\_value=summary.glm(model)$coef[2,4])**

**}**

**ORadj<-rownames(ORadj) %>% cbind(ORadj)**

**ORadj<-as\_tibble(ORadj)**

**nomscol<-c("Variable","OR","Linf","Lsup","pvalor")**

**ORadj<-ORadj %>% setNames(nomscol)**

**ORadj**

**}**

**# Variables.ajust -----------------**

**##### hi envio la columna de variables amb que vull generar la formula pel compare**

**# FUNCIO variables.ajust**

**variables.ajust=function(x="taula1",variables=variables) {**

**pepito<-paste("as.vector(variables[variables$",x,"==1,]$camp)[!as.vector(variables[variables$",x,"==1,]$camp)%in%c('idp','grup')]",sep="")**

**llistataula<-eval(parse(text=pepito))**

**z<-paste(llistataula, collapse=" + ")**

**}**

**# GLM COEFICIENTS ###########################################################**

**################# EXTREU COEFICIENTS glm, IC95 , p valors GLM a partir de llista d'outcomes, X, i llista de v.ajust**

**extreure\_coef\_glm<-function(dt=dades,outcomes="OFT\_WORST",x="DM",z="",taulavariables="variables\_R.xls"){**

**# dt=dades**

**# outcomes="lipos"**

**# x="MCD"**

**# z="variables\_ajust"**

**# taulavariables=conductor\_variables**

**# Número de categories de X**

**Ncat.x<-sum(table(dt[x])!=0)**

**if (is.numeric(dt[[x]])) Ncat.x=1**

**### Si hi ha variables d'ajust genero llista**

**if (z!="") mam<-names(selectorvariables(z,dt=dt,taulavariables=taulavariables)) ### Genero llista de variables**

**if (z!="") x<-paste0(paste0(mam,collapse = "+"),"+",x)**

**models1\_oft<-names(selectorvariables(outcomes,dt=dt,taulavariables=taulavariables))%>%**

**paste('~',x) %>%**

**purrr::map(~glm(as.formula(.x), data= dt))%>%**

**purrr::map(summary) %>%**

**purrr::map(coefficients)**

**if (Ncat.x>1) noms\_var\_X<-models1\_oft[[1]] %>%**

**rownames %>% #**

**tail(Ncat.x-1) # Capturo nom categories de X**

**if (Ncat.x==1) noms\_var\_X<-models1\_oft[[1]] %>%**

**rownames %>% tail(1)**

**# names(table(dt[x]))[2:Ncat.x]**

**if (Ncat.x==1) models1\_oft<-models1\_oft %>% # Si es continua només un coef de X**

**purrr::map(tail,Ncat.x) %>%**

**purrr::map\_dfr(data.table)**

**if (Ncat.x>1) models1\_oft<-models1\_oft %>% ## Select només num de coeficients necessaris de X**

**purrr::map(tail,Ncat.x-1) %>%**

**purrr::map\_dfr(data.table)**

**if (Ncat.x>1) variables<-names(selectorvariables(outcomes,taulavariables,dt=dt)) %>% ## Noms dels outcomes**

**rep(each=Ncat.x-1) %>% ## Cada Num de coeficients**

**data.table() # Outcomes**

**if (Ncat.x==1) variables<-names(selectorvariables(outcomes,taulavariables,dt=dt)) %>% ## Noms dels outcomes**

**data.table() # Outcomes**

**colnames(variables)<-"Outcome"**

**models\_taula<-cbind(variables,Cat.X=noms\_var\_X,models1\_oft)**

**models\_taula<-models\_taula %>% dplyr::select(-c("t value")) ## Elimino t value**

**list(coef=models\_taula,caption=paste("Coeficient ajustat per:", x))**

**}**

**# GLM (Logistic o Lineal) dades imputades --------------------**

**## Retorn de coeficients glm() amb dades imputades d'una variable independent X ~ Y**

**extreure\_coef\_glm\_mi<-function(dt=tempData,outcome="valor612M.GLICADA",x="SEXE") {**

**# dt=tempData**

**# outcome="canvi612M.glicadaCAT2"**

**# x="IMC\_cat4"**

**# Outcome es factor?**

**outcome\_es\_factor<-data\_long[[outcome]] %>% class=="factor" | data\_long[[outcome]] %>% class=="character"**

**# Si Outcome (Y) es factor --> glm-Logistica**

**if (outcome\_es\_factor) {**

**pepe<-paste0(outcome,"~",x)**

**resum<-with(tempData,glm(eval(parse(text=pepe)),family = binomial(link="logit"))) %>% pool %>% summary**

**resum\_model<-tibble(categoria=row.names(resum)) %>%**

**cbind(resum) %>%**

**mutate(OR=estimate %>% exp,**

**Linf=(estimate-std.error) %>% exp,**

**Lsup=(estimate+std.error) %>% exp)**

**}**

**# Si outcome (Y) es numeric --> GLM lineal**

**if (!outcome\_es\_factor) {**

**pepe<-paste0(outcome,"~",x)**

**model\_mi<-with(dt,lm(eval(parse(text=pepe))))**

**resum<-summary(mice::pool(model\_mi))**

**resum\_model<-tibble(categoria=row.names(resum)) %>% cbind(resum)**

**}**

**# Si X es cat afegir categoria de referencia**

**es\_factor<- data\_long[[x]] %>% class=="factor" | data\_long[[x]] %>% class=="character"**

**num\_categories<-data\_long[[x]] %>% table %>% length**

**if (es\_factor) {**

**resumtotal<-tibble(categoria=row.names(resum),outcome=outcome) %>%**

**add\_row (categoria=paste0(x,".Ref"),outcome=outcome)**

**}**

**# Si no es factor**

**if (!es\_factor) {resumtotal<-tibble(categoria=row.names(resum),outcome=outcome) }**

**# Afegir categoria**

**resumtotal<-resumtotal %>% left\_join(resum\_model,by="categoria")**

**# Només en GLM calcular la mitjana estimada per categoria**

**if (es\_factor & outcome\_es\_factor==F) {**

**resumtotal<-resumtotal %>% mutate (beta0=resumtotal$estimate[1],estimate=ifelse(is.na(estimate),0,estimate)) %>%**

**mutate(mean=ifelse(categoria!="(Intercept)", beta0+estimate,NA))**

**}**

**# Seleccionar columnes**

**resumtotal**

**}**

**# Coeficients GLM(lineal/logistica) MICE estratificats ---------------------**

**# Arguments: Objecte MICE i data\_list imputats, vector de X , Y , logit=T/F**

**extreure\_coef\_mice\_estrats<-function(tempData,data\_list,X=c("bmi","hyp"),Y="chl",grups="age",logit=F) {**

**# tempData**

**# data\_list**

**# X=X**

**# Y="bmi"**

**# grups="age"**

**# logit=F**

**# .data=data\_list[[1]]**

**fitting=function(.data,frm,logit=F) {**

**if(logit) {model=glm(frm,data=.data,family=binomial(link="logit"))}**

**if(!logit){model=lm(frm, data =.data)}**

**model**

**}**

**# Cada llista de datasets separat per grups**

**data\_list\_splitted<-data\_list %>% map(~base::split(.x,.x[[grups]]))**

**# numero de grups**

**num\_grups<-tempData$data[[grups]] %>% table %>% length**

**# Aplica models n una llista**

**models\_list<-lapply(1:num\_grups, function(NSPLIT) data\_list\_splitted %>%**

**lapply(nth, NSPLIT) %>%**

**lapply(fitting, formula\_vector(X,Y,logit),logit=logit) %>%**

**mice::as.mira() %>%**

**mice::pool() %>%**

**summary() %>%**

**tibble::rownames\_to\_column("variable"))**

**# Posar noms als grups**

**names(models\_list)<-names(data\_list\_splitted[[1]])**

**# Ho posa en un data set**

**models\_dt<-bind\_rows(models\_list, .id = "Grup") %>% as\_tibble**

**models\_dt**

**}**

**# K-M plot #####**

**plotKM=function(y=exitus.surv,grup=grup,d=dades,caption="",llegenda=c("No","Yes")) {**

**# y=dadesDF$exitus\_surv**

**# grup=dadesDF$hta**

**# d=dadesDF**

**# caption=Hmisc::label(dadesDF$hta)**

**# llegenda=c("No","Yes")**

**# y=dadesDF$exitus\_surv**

**# grup=dadesDF$edad\_cat6**

**# d=dadesDF**

**# llegenda=c("<45", "[45-55)", "[55-65)", "[65-75)", "[75-85)","85+")**

**# Basic survival curves**

**p <- survminer::ggsurvplot(survfit(y ~ grup, data = d), data = d,**

**main = "Survival curve",**

**title= caption,**

**size = 0.5,**

**ylim = c(0,1),**

**xlim = c(0,60),**

**break.x.by=12,**

**xlab = "Time in months",**

**risk.table = F,**

**censor.shape="|", censor.size = 1**

**,legend.labs=llegenda)**

**p**

**}**

**# K-M plot #####**

**plotKM\_Incidence=function(y=exitus.surv,grup=grup,d=dades,caption="",llegenda=c("No","Yes")) {**

**# caption=""**

**# llegenda=c("No","Yes")**

**# y=dadesDF$exitus\_surv**

**# grup=dadesDF$edad\_cat6**

**# d=dadesDF**

**# llegenda=c("85+","[75-85)", "[65-75)", "[55-65)", "[45-55)","<45")**

**# Basic survival curves**

**p <- survminer::ggsurvplot(survfit(y ~ grup, data = d), data = d,**

**main = "Survival curve",**

**title= caption,**

**size = 0.5,**

**ylim = c(0,1),**

**xlim = c(0,60),**

**break.x.by=12,**

**linesize="strata",**

**xlab = "Time in months",**

**risk.table = F,**

**censor.shape=".",**

**censor.size = 0.5,**

**legend.labs=llegenda,**

**legend="right",**

**fun="event",**

**ggtheme = theme\_bw(),**

**palette = c("black","black","black","black","black","black"))**

**p**

**}**

**# Box-plot -----------------**

**boxplot\_variables\_grup<-function(dt=dades,variables="OFT\_WORST",grup="DM", taulavariables="variables\_R.xls") {**

**# dt=dades**

**# variables="OFT\_WORST"**

**# grup="DM"**

**# taulavariables="variables\_R.xls"**

**### extrect variables**

**paco<-extreure.variables(variables,taulavariables=taulavariables)**

**### Genero taula llarga**

**popes<-dt %>%**

**dplyr::select(c(paco,grup)) %>%**

**gather\_(key=variables,value="valor",setdiff(paco, grup))**

**### FAi ggplot**

**figura1<-popes %>% ggplot2::ggplot(aes\_string(x=variables, y="valor",fill=grup))+geom\_boxplot()**

**figura1**

**}**

**# Figura Spline Y~x per grups --------------------**

**# Spline Y ~ x (continua) estratificat per grups)**

**# Requereix Y, X, grup y dades**

**ggplot\_grups<-function(Y="DIS\_estatina",dt=dades,X="edat",grup="sexe") {**

**figuragamX<-ggplot(dt, aes\_string(x=X, y=Y,group=grup,shape=grup, color=grup))+**

**geom\_smooth(method = lm, formula = y ~ splines::bs(x, 3), se = T)+**

**xlab(Hmisc::label(dades[X]))+**

**ylab(Hmisc::label(dades[Y]))+**

**theme\_bw()+**

**labs(colour =grup)+**

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

**figuragamX**

**}**

**# Retorna un mapa temporal (datainicial-datafinal per grups) Individus a partir de:**

**# dades, datainicial, data final, id, grup color, grup linea, finestra (porca1,porca2)**

**MAP\_ggplot<-function(dades=dt,datainicial="data",datafinal="datafi",id="idp\_temp",grup\_color=NA,grup\_linea=NA,lim\_inf=-Inf,lim\_sup=Inf) {**

**# dades=dadestemp**

**# datainicial="dtinclusio"**

**# datafinal="dat\_sit\_2014"**

**# id="id"**

**# grup\_color="sexe"**

**# grup\_linea="sc\_bcn"**

**# lim\_inf=-Inf**

**# lim\_sup=+Inf**

**if (is.na(grup\_linea)) dades<- dades %>% mutate(Overall="Overall")**

**if (is.na(grup\_linea)) grup\_linea<- "Overall"**

**if (is.na(grup\_color)) dades<- dades %>% mutate(Overall2="Overall2")**

**if (is.na(grup\_color)) grup\_color<- "Overall2"**

**# # Configuro limits finestra**

**if (lim\_inf==-Inf) porca1<-min(dades %>% pull(datainicial) %>% lubridate::ymd())**

**if (lim\_sup==+Inf) porca2<-max(dades %>% pull(datafinal) %>% lubridate::ymd())**

**# #**

**if (lim\_inf!=-Inf) porca1<-lim\_inf**

**if (lim\_sup!=+Inf) porca2<-lim\_sup**

**porca1=lubridate::ymd(porca1)**

**porca2=lubridate::ymd(porca2)**

**# Conversió a Sym per evaluació**

**datainicial<-rlang::sym(datainicial)**

**datafinal<-rlang::sym(datafinal)**

**id<-rlang::sym(id)**

**grup\_color<-rlang::sym(grup\_color)**

**grup\_linea<-rlang::sym(grup\_linea)**

**# Calculo dies de duració**

**dades<-dades %>%**

**mutate(**

**dia0=lubridate::ymd(!!datainicial),**

**diaf=lubridate::ymd(!!datafinal),**

**days\_duration=lubridate::interval(dia0,diaf) %>% lubridate::as.duration()/lubridate::ddays()**

**)**

**# Gráfico el tema**

**ggplot(dades,aes(x =dia0,y =!!id, color=!!grup\_color,group=!!grup\_linea,linetype=!!grup\_linea))+**

**geom\_segment(aes(x =dia0, xend=diaf, y =!!id, yend = !!id),arrow = arrow(length = unit(0.03, "npc"))) +**

**geom\_point(aes(dia0, !!id)) +**

**geom\_text(vjust = -0.5, hjust=0, size = 3,aes(x =dia0, y = !!id,label = paste(round(days\_duration, 2), "days")))+**

**scale\_colour\_brewer(palette = "Set1")+**

**xlim(porca1,porca2)+**

**theme(legend.position="top",legend.background = element\_rect(fill="gray80",size=1, linetype="solid", colour ="black"))**

**}**

**# Retorna llista amb dos data\_frames de farmacs i dos plots pre i post**

**Gaps<-function(dt=dades,K=14,Nmostra=10,finestraX=c(NA,NA),llavor=123){**

**# dt=temp\_dades**

**# K=14**

**# Nmostra=5**

**# finestraX=c(NA,NA)**

**# llavor=123**

**# if (Nmostra==Inf) Nmostra=10**

**# Si Nmostra es infinit o mes gran que la mostra agafo el màxim**

**Nmostra\_maxim<- dt %>% distinct(idp) %>% nrow()**

**if (Nmostra==Inf | Nmostra>Nmostra\_maxim) Nmostra<- Nmostra\_maxim**

**farmacs\_list<-dt %>%distinct(agr)%>%dplyr::pull()**

**dt<-dt%>% mutate(agr=factor(agr))**

**set.seed(llavor) # S'ha d'actualitzar**

**id\_sample<-dt %>% distinct(idp) %>%sample\_n(size=Nmostra)**

**dt<-id\_sample %>% left\_join(dt,by="idp")**

**dt<-dt%>%dplyr::select(idp,agr,data=dat,datafi,FACTPRESC=tipus)**

**# Calculo dies de duració**

**dt<-dt %>%**

**mutate(**

**data=lubridate::ymd(data),**

**datafi=lubridate::ymd(datafi),**

**days\_duration=lubridate::interval(data,datafi) %>% as.duration()/ddays())**

**dt<-dt %>% mutate (idp2=idp, idp=paste0(idp,agr,".",str\_sub(FACTPRESC,1,1)))**

**dt<-dt%>%dplyr::select(idp,agr,data,datafi,days\_duration,idp2,FACTPRESC)**

**# Genera mapa origen (n)**

**dt<-dt %>% mutate (idp\_temp=paste0(stringr::str\_sub(dt$idp,1,6),agr,".",str\_sub(FACTPRESC,1,1)))**

**if (is.na(finestraX[1])) porca1<-lubridate::ymd(min(dt$data))**

**if (is.na(finestraX[2])) porca2<-lubridate::ymd(max(dt$datafi))**

**if (!is.na(finestraX[1])) porca1<-lubridate::ymd(finestraX[1])**

**if (!is.na(finestraX[2])) porca2<-lubridate::ymd(finestraX[2])**

**dt<-dt %>% mutate(datafi =case\_when(porca2<=datafi ~ porca2,TRUE ~ datafi))**

**# Recalcular intervals en dies a partir de les finetres!**

**dt<-dt%>%mutate(days\_duration=interval(data,datafi)%>%as.duration()/ddays())**

**MAP<-MAP\_ggplot(dades=dt,datainicial="data",datafinal="datafi",id="idp\_temp",grup\_color="agr",grup\_linea="FACTPRESC",lim\_inf=porca1,lim\_sup=porca2)**

**dt<-dt%>%arrange(idp,data,datafi)**

**dt<-mutate(dt,data=ymd(data),datafi=ymd(datafi))**

**dt<-dt%>%group\_by(idp)%>% mutate(gap=(data-lag(datafi)))**

**dt<-dt%>%mutate(gap2=case\_when(gap>K ~1, TRUE ~0))**

**dt<-dt%>%group\_by(idp)%>%mutate(gap3=(cumsum(gap2)))%>%ungroup()**

**# Agregate**

**dt2<-dt %>%**

**dplyr::select(idp,data,datafi,gap3,agr,idp2, FACTPRESC) %>%**

**group\_by(idp,agr,gap3)%>%**

**summarise(data= min(data), datafi= max(datafi),idp2=min(idp2),FACTPRESC=min(FACTPRESC))%>%**

**ungroup**

**#**

**# Tornem a Recalcular intervals en dies a partir dels Gaps i Fienstra!.**

**dt2<-dt2%>%mutate(days\_duration=interval(data,datafi)%>%as.duration()/ddays())**

**dt2<-dt2 %>% mutate(idp\_temp=paste0(stringr::str\_sub(dt2$idp,1,6),agr,".",str\_sub(FACTPRESC,1,1)))**

**MAP2<MAP\_ggplot(dades=dt2,datainicial="data",datafinal="datafi",id="idp\_temp",grup\_color="agr",grup\_linea="FACTPRESC",lim\_inf=porca1,lim\_sup=porca2)**

**#MAP2**

**dt2<-dt2 %>% dplyr::select(idp2,idp,agr,data,datafi,FACTPRESC)**

**#dt2**

**list(dades1=dt,dades2=dt2,Mapa\_pre=MAP,Mapa\_post=MAP2)**

**}**

**#**

**# Historic de farmacs: idp, datinici,datafi, gap**

**# Elimina solapaments i discontinuitats petites i retorna dades sense discontinuitats ni solapaments amb igual o menys registres**

**# Retorna dades amb : id, datainici i datafi amb menys registres, havent eliminat solapaments i gaps (discontinuitat petita)**

**agregar\_solapaments\_gaps<-function(dt=dades,id="idp",datainici="data",datafinal="datafi",gap=5){**

**# dt=FX.FACTURATS\_PRESCRITS\_GRUPS**

**# gap=60**

**# datainici="dat"**

**# datafinal="datafi"**

**# id="idp"**

**# Conversió a Sym per evaluació**

**datainici\_sym<-rlang::sym(datainici)**

**datafinal\_sym<-rlang::sym(datafinal)**

**idp\_sym=rlang::sym(id)**

**# Seleccionar dades necessaries amb noms sense sym**

**dt<-dt %>% dplyr::select(idp=!!idp\_sym, data=!!datainici\_sym,datafi=!!datafinal\_sym)**

**# 1. Eliminar solapaments**

**dt<-dt %>%**

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

**arrange(data) %>%**

**mutate(indx = c(0, cumsum(as.numeric(lead(data)) >**

**cummax(as.numeric(datafi)))[-n()])) %>%**

**group\_by(idp, indx) %>%**

**summarise(data = min(data), datafi = max(datafi)) %>%**

**dplyr::select(-indx)**

**# 2. ELiminar Gaps (discontinuitats)**

**# MAP\_ggplot(dades=dt,datainicial="data",datafinal="datafi",id="idp",grup\_color=NA,grup\_linea=NA)**

**dt<-dt %>%**

**mutate(gap1=(data-lag(datafi))) %>%**

**mutate(gap2=case\_when(gap1 > gap ~1, TRUE ~0)) %>%**

**mutate(gap3=(cumsum(gap2))) %>%**

**ungroup()**

**# 3. Agregate per idp-gap**

**dt2<-dt %>% mutate (idp2=idp) %>%**

**dplyr::select(idp,data,datafi,gap3,idp,idp2) %>%**

**group\_by(idp,gap3)%>%**

**summarise(datainici= min(data), datafi= max(datafi),idp2=min(idp2)) %>%**

**ungroup()**

**dt2<-dt2 %>% dplyr::select("idp","datainici","datafi")**

**# MAP\_ggplot(dades= dt2 %>% head(10),datainicial="datainici",datafinal="datafi",id=id)**

**# Renombro noms dels camps originals**

**colnames(dt2)<-c(idp\_sym,datainici\_sym,datafinal\_sym)**

**dt2**

**}**

**# Dibuixa mapa temporal univariant per verificar solapaments**

**MAP\_ggplot\_univariant<-function(dades=dt,datainicial="data",datafinal="datafi",id="idp\_temp", Nmostra=Inf) {**

**# dades=farmacs\_dt\_sense\_gaps %>% filter(GRUP=="IDPP4")**

**# datainicial="dat"**

**# datafinal="datafi"**

**# id="idp"**

**# Nmostra=2**

**# Conversió a Sym per evaluació**

**datainicial<-rlang::sym(datainicial)**

**datafinal<-rlang::sym(datafinal)**

**id\_sym<-rlang::sym(id)**

**# mostrejo**

**dades<-mostreig\_ids(dt=dades,id=id,n\_mostra = Nmostra)**

**# if (Nmostra!=Inf) id\_sample<-dades %>% distinct(!!id) %>% sample\_n(size=Nmostra)**

**# dt<-id\_sample %>% left\_join(dt,by=quo\_name(id)) #**

**# Calculo dies de duració**

**dades<-dades %>% mutate(dia0=!!datainicial,diaf=!!datafinal,days\_duration=diaf-dia0)**

**# Gráfico el tema**

**ggplot2::ggplot(dades,ggplot2::aes(x =dia0,y =!!id\_sym))+**

**ggplot2::geom\_segment(ggplot2::aes(x =dia0, xend=diaf, y =!!id\_sym, yend = !!id\_sym),arrow = ggplot2::arrow(length = ggplot2::unit(0.03, "npc"))) +**

**ggplot2::geom\_point(ggplot2::aes(dia0, !!id\_sym)) +**

**ggplot2::geom\_text(vjust = -0.5, hjust=0, size = 3, ggplot2::aes(x =dia0, y = !!id\_sym,label = paste(round(days\_duration, 2), "days")))+**

**ggplot2::scale\_colour\_brewer(palette = "Set1")+**

**ggplot2::theme(legend.position="top",legend.background = ggplot2::element\_rect(fill="gray80",size=1, linetype="solid", colour ="black"))**

**}**

**# Analitiques (Y=Individu, X=data, Tamany=Valor, Color=tipus analitica) -----------------**

**#**

**MAP\_punts\_ggplot<-function(**

**dt=mostra50,**

**id="idp",**

**datainicial="dat",**

**val="val",**

**grup\_color="agr",**

**Nmostra=Inf,**

**llavor=123,**

**finestraX=c(-Inf,+Inf),**

**id\_AGG=F**

**)**

**{**

**# dt=VARIABLES**

**# id="idp"**

**# datainicial ="dat"**

**# val="val"**

**# grup\_color = "cod"**

**# Nmostra = 2**

**# finestraX=c(-Inf,+Inf)**

**# llavor=126**

**# id\_AGG=T**

**if (finestraX[1]==-Inf) porca1<-min(dt %>% pull(datainicial)) %>% ymd()**

**if (finestraX[2]==+Inf) porca2<-max(dt %>% pull(datainicial)) %>% ymd()**

**if (finestraX[1]!=-Inf) porca1<-finestraX[1] %>% ymd()**

**if (finestraX[2]!=+Inf) porca2<-finestraX[2] %>% ymd()**

**# Interpretacio com a parametre**

**grup\_color<-rlang::sym(grup\_color)**

**datainicial<-rlang::sym(datainicial)**

**id<-rlang::sym(id)**

**val<-rlang::sym(val)**

**# Converteix data a data inicial**

**dt<-dt %>% mutate(dat=ymd(!!datainicial))**

**# Cal estandarditzar valor**

**# Llista de nombre d'analitiques**

**analitiques\_list<-dt%>%distinct(!!grup\_color)%>%dplyr::pull()**

**set.seed(llavor) # S'ha d'actualitzar**

**#**

**id\_sample<-dt %>% distinct(!!id) %>% sample\_n(size=Nmostra)**

**dt<-id\_sample %>% left\_join(dt,by=quo\_name(id)) #**

**# Construccio del identificador id-grup**

**if (id\_AGG){**

**dt<-dt%>%mutate(id\_plot=paste0(stringr::str\_sub(!!id,1,6),!!grup\_color),id\_num=as.numeric(factor(!!id)))**

**}**

**if (id\_AGG ==F) {**

**dt<-dt%>%mutate(id\_plot=paste0(stringr::str\_sub(!!id,1,6)),id\_num=as.numeric(factor(!!id))) }**

**ggplot(dt,aes(x =!!datainicial,y =id\_plot,color=!!grup\_color))+**

**geom\_point(aes(!!datainicial, id\_plot)) +**

**geom\_point(aes(size = !!val))+**

**labs(title = "Històric de determinacions")+theme(plot.title = element\_text(size=30,hjust = 0.5))+**

**theme(axis.text = element\_text(colour = "black",size = 10))+**

**theme(panel.grid.major = element\_line(colour = "grey80",size=0.001))+**

**theme(axis.line = element\_line(colour = "black",size = 0.9))+**

**scale\_colour\_brewer(palette = "Set1")+**

**xlim(porca1,porca2)+**

**geom\_text(vjust = -0.5, hjust=0, size = 3,aes(x =!!datainicial, y =id\_plot,label = paste(round(!!val, 2),""))) +**

**theme(legend.position="top",legend.background = element\_rect(fill="gray80",size=1, linetype="solid", colour ="black"))+**

**scale\_y\_discrete(breaks= dt %>% pull(id\_plot),labels=dt %>% pull(id\_num))**

**#**

**}**

**# Analitiques (Y=Individu, X=data, Tamany=Valor, Color=tipus analitica) -----------------**

**MAP\_valor\_ggplot<-function(**

**dt=mostra50,**

**id="idp",**

**datainicial="dat",**

**val="val",**

**grup\_color="agr",**

**Nmostra=1,**

**finestraX=c(-Inf,Inf),**

**llavor=123**

**)**

**{**

**# dt=VARIABLES %>% filter(cod %in% c("HBA1C"))**

**# datainicial ="dat"**

**# id="idp"**

**# val="val"**

**# grup\_color = "cod"**

**# Nmostra = 4**

**# finestraX=c(-Inf,Inf)**

**# llavor=126**

**if (finestraX[1]==-Inf) {porca1<-min(dt %>% pull(datainicial)) %>% ymd()}**

**if (finestraX[2]==+Inf) {porca2<-max(dt %>% pull(datainicial)) %>% ymd()}**

**if (finestraX[1]!=-Inf) {porca1<-finestraX[1] %>% ymd()}**

**if (finestraX[2]!=+Inf) {porca2<-finestraX[2] %>% ymd()}**

**# Interpretacio com a parametre**

**grup\_color<-rlang::sym(grup\_color)**

**datainicial<-rlang::sym(datainicial)**

**id<-rlang::sym(id)**

**val<-rlang::sym(val)**

**# Formatejo a data**

**dt<-dt %>% mutate(dat=lubridate::ymd(!!datainicial))**

**# Llistat de codis d'analitiques**

**analitiques\_list<-dt%>%distinct(!!grup\_color)%>%dplyr::pull()**

**set.seed(llavor) # S'ha d'actualitzar**

**# Seleccionar sample**

**id\_sample<-dt%>% distinct(!!id) %>%sample\_n(size=Nmostra)**

**dt<-id\_sample %>% left\_join(dt,by=quo\_name(id)) #**

**#**

**# Construcció del identificador id-grup**

**dt<-dt%>%mutate(id\_plot=paste0(stringr::str\_sub(!!id,1,6),!!grup\_color))**

**# Grafica plot de la variable**

**ggplot(dt,aes(x =!!datainicial,y =id\_plot,color=id\_plot))+**

**geom\_line(aes(!!datainicial, !!val))+**

**geom\_point(aes(!!datainicial, !!val),color="black")+**

**labs(title = "Evolució de valors")+ theme(plot.title = element\_text(size=30,hjust = 0.5))+**

**theme(axis.text = element\_text(colour = "black",size = 10))+**

**theme(panel.grid.major = element\_line(colour = "grey80",size=0.001))+**

**theme(axis.line = element\_line(colour = "black",size = 0.9))+**

**scale\_colour\_brewer(palette = "Set1")+**

**xlim(porca1,porca2)+**

**geom\_text(vjust = -0.5, hjust=0, size = 3,aes(x =!!datainicial, y =!!val,label = paste(round(!!val, 2),""))) +**

**theme(legend.position="top",legend.background = element\_rect(fill="gray80",size=1, linetype="solid", colour ="black"))**

**}**

**# HR.COX --------------------**

**#### funció que retorna MATRIU-->Ngran, Events, HR, IC951, IC952, p**

**HR.COX=function(x="v.ajust",event="EV.INSUF\_CARD",t="tmp\_insuf\_card",e="",d=dadesDF,taulavariables="variables.xls") {**

**# x="ajust3\_obj2"**

**# event="EV\_IAM\_DIC"**

**# t="temps\_EV\_IAM"**

**# d=dadestotal**

**# taulavariables = conductor\_variables**

**# e=""**

**pepito<-paste("sum(d$",t,")",sep="")**

**PT<-eval(parse(text=pepito))**

**result=tryCatch({**

**pp<-survival::coxph(formulaCOX(x=x,event=event,temps=t,elimina=e,taulavariables = taulavariables),data=d)**

**cbind(N=pp$n,**

**EVENTS=pp$nevent,**

**HRadjusted=summary(pp)$coef[,2],**

**IC951=summary(pp)$conf.int[,3],**

**IC952=summary(pp)$conf.int[,4],**

**p=summary(pp)$coef[,5])}**

**,error = function(e) {**

**cbind(N=0,**

**EVENTS=0,**

**HRadjusted=NA,**

**IC951=NA,**

**IC952=NA,**

**p=NA)}**

**)**

**result**

**}**

**# HR CRUS ------------------**

**HR.COX.CRU=function(x="lipos",event="EVENT\_MCV",t="temps\_exitus",e="",d=dadesDF,variables="variables\_R.xls",evento="Si") {**

**# x="Baseline"**

**# event="RD"**

**# t="TEMPS\_RD2"**

**# d=dadestotal**

**# variables=conductor\_variables**

**# evento="1"**

**bd.camps<-selectorvariables(x,dt=d,taulavariables=variables)**

**camps<-names(bd.camps)**

**num\_camps<-length(names(bd.camps))**

**poco<-cbind()**

**for (i in 1:num\_camps) {**

**# i<-1**

**xx<-camps[i]**

**rr<-paste("Surv(",t,", as.integer(",event," == ",evento,"))~",xx,sep="")**

**pp<-coxph(eval(parse(text=rr)),data=d)**

**mama<-cbind(N=pp$n,**

**EVENTS=pp$nevent,**

**HRcrude=summary(pp)$coef[,2],**

**IC951=summary(pp)$conf.int[,3],**

**IC952=summary(pp)$conf.int[,4],**

**p=summary(pp)$coef[,5])**

**rownames(mama)<-names(pp$coefficients)**

**poco<-rbind(poco,mama)**

**}**

**poco**

**}**

**################# FUNCIO PER EXTREURE CORRELACIONS, P VALORS ENTRE var1 i llista de quantis de dades**

**extreure\_cor=function(var1="CD36",var="quantis",d="dades",taulavariables="VARIABLES.xls") {**

**# var1="HbA1c"**

**# var="lipos2"**

**# d="dades"**

**# taulavariables="VARIABLES.xls"**

**## Llegeix criteris de variables**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**llistavariables<-eval(parse(text=paste("variables$camp[variables$",var,">0]",sep="")))**

**# llistavariables<-variables$camp[variables$var==1]**

**x<-eval(parse(text=paste(d,"$",var1,sep="")))**

**ppp<-cbind()**

**for (i in 2:length(llistavariables)) {**

**var2<-paste(d,llistavariables[i],sep="$")**

**y<-eval(parse(text=var2))**

**cor.test(x,y)$estimate**

**correlacio<-cor.test(x,y)$estimate**

**pvalor<-cor.test(x,y)$p.value**

**pp<-cbind(correlacio,pvalor)**

**row.names(pp)<-llistavariables[i]**

**ppp<-rbind(ppp,pp)**

**}**

**ppp**

**}**

**# Extreure OR (segons formula, i dades) --------------------**

**# LLANÇO UNA FORMULA les dades per executar un model i retorno OR , CI95% i p-valor en una tibble()**

**extreure\_OR<- function (formu="AnyPlaqueBasal~CD5L",dades=dt,conditional=F,strata="caseid") {**

**# formu<-formula.LOGIT(x="article.model",y="canvi312M.GLICADA.inputCAT2",taulavariables='variables\_v2.xls')**

**# dades=tempData**

**# formu=formula**

**# dades=dades**

**# conditional=F**

**# strata="caseid"**

**modelcomplet=T**

**dades\_resum<-as\_tibble()**

**# Si dades NO son dades imputadesl**

**if (class(dades)[1]!="mids") {**

**# Model logistic / logistic condicional**

**if (conditional==F) {**

**fit<-stats::glm(formu, family = binomial, data=dades)**

**} else {**

**formu<- paste0(formu,"+ strata(",strata,")")**

**fit<-survival::clogit(as.formula(formu),data=dades)}**

**# Extrec info total del model**

**my\_coefficients <- fit %>% coef**

**ci<-fit %>% confint**

**OR<-my\_coefficients %>% exp()**

**OR\_linf<-ci %>% exp()**

**pvalors<-coef(summary(fit))[,'Pr(>|z|)']**

**coeficients<-cbind(OR,OR\_linf,pvalors) %>% as\_tibble**

**ret\_val <- tibble::enframe(row.names(ci)) %>% bind\_cols(coeficients)**

**colnames(ret\_val) <- c("id","Categoria","OR","Linf", "Lsup", "p.value")**

**dades\_resum<-ret\_val %>% as\_tibble**

**}**

**dades\_resum**

**# Si son dades imputades tipo mids de MICE**

**if (class(dades)[1]=="mids"){**

**pepe<-paste(formu[2],formu[3],sep='~')**

**resum<-with(tempData,glm(eval(parse(text=pepe)),family = binomial(link="logit"))) %>% mice::pool() %>% summary ()**

**ret\_val<-cbind(categoria=row.names(resum)) %>% cbind(resum) %>% as\_tibble**

**# Capturar OR, etc...**

**dades\_resum<-ret\_val %>% mutate(OR=estimate %>% exp,**

**Linf=(estimate-std.error) %>% exp,**

**Lsup=(estimate+std.error) %>% exp) %>%**

**dplyr::select(categoria,OR,Linf,Lsup,p.value)**

**}**

**dades\_resum**

**}**

**# Taula variables segons formula i dades genera la taula de coeficients**

**generar\_taula\_variables\_formula<-function(formu="AnyPlaqueBasal~CD5L",dades=dt) {**

**# formu=formu**

**# dt=dades**

**taula\_editada<-**

**all.vars(formu)[-1] %>%**

**map(~paste0(.x,levels(dades[[.x]]),"/",.x)) %>%**

**unlist() %>%**

**tibble() %>% rename("var"=".") %>%**

**separate(col=var, into=c("Categoria","Variable"), sep = "/") %>%**

**mutate(nivell=stringr::str\_remove(Categoria,Variable),**

**tipo=if\_else(nivell=="","Continua","Cat"))**

**}**

**# Retorno model amb ORs, curva ROC , auc IC95% etc... a partir de formula glm , i dades**

**extreure\_model\_logistic<-function(x="OS4\_GSK",y="canvi6M.glipesCAT2",taulavariables=conductorvariables,dades=dades,elimina=c("IDP"),a="", valor\_outcome="Yes",conditional=F,strata="caseid") {**

**# x="grup"**

**# y="HBA1C.dif324m.cat"**

**# taulavariables=conductor\_variables**

**# dades=dades**

**# elimina=c("IDP")**

**# a=""**

**# valor\_outcome="Yes"**

**# conditional = F**

**# strata = "caseid"**

**# Ojo que variables no factoritzades --> error**

**formu=formula.LOGIT(x=x,y=y,taulavariables=taulavariables)**

**formu\_text<-formula.text(x=x,y=y,taulavariables=taulavariables)**

**resposta<-all.vars(formu)[1]**

**fit<-stats::glm(formu, family = binomial, data=dades)**

**if (conditional==F) {**

**taula\_OR<-extreure\_OR(formu=formu,dades=dades,conditional=conditional,strata=strata)**

**} else {**

**taula\_OR<-extreure\_OR(formu=formu\_text,dades=dades,conditional=conditional,strata=strata)**

**fit\_c<-survival::clogit(as.formula(paste0(formu\_text,"+ strata(",strata,")")),data=dades)**

**}**

**taula\_editada<-generar\_taula\_variables\_formula(formu,dades)**

**# juntar taula\_OR + taula editada --> etiquetar i editar**

**taula\_editada<-taula\_editada %>%**

**left\_join(taula\_OR,by="Categoria") %>%**

**mutate(nivell=if\_else(is.na(OR),paste0(" Ref:",nivell),nivell),**

**OR=if\_else(is.na(OR),1,OR),**

**Linf=if\_else(is.na(Linf),1,Linf),**

**Lsup=if\_else(is.na(Lsup),1,Lsup),**

**nivell=stringr::str\_trim(nivell)) %>%**

**filter (!is.na(id)) %>% # Eliminar cat de referencia**

**etiquetar\_taula("Variable",taulavariables,"descripcio") %>%**

**mutate(Variable=if\_else(tipo=="Cat",paste0(Variable,":",nivell),Variable)) %>%**

**dplyr::select(Categoria=Variable,OR,Linf,Lsup,p.value)**

**forest\_plot<-forest.plot(taula\_editada)**

**dades\_prediccio<-**

**data.frame(prediccio=predict(fit,dades, type=c("response")),known.truth=dades %>% pull(resposta)) %>%**

**tibble::as\_tibble() %>%**

**mutate(event=as.numeric(known.truth==valor\_outcome)) %>%**

**filter(!is.na(event) & !is.na(prediccio))**

**if (conditional) {**

**predict\_clogit<-data.frame(logit\_pred=predict(fit\_c,type = "lp")) %>%**

**mutate(prob\_pred=boot::inv.logit(logit\_pred))**

**dades\_prediccio<-dades\_prediccio %>%**

**cbind(predict\_clogit) %>% dplyr::select(-prediccio) %>% rename(prediccio=prob\_pred)**

**}**

**g <- pROC::roc(event ~ prediccio, data = dades\_prediccio)**

**auc=pROC::auc(g)**

**auc\_ci=pROC::ci(g)**

**plot\_curve<-**

**ggplot(dades\_prediccio, aes(d = event, m = prediccio)) +**

**plotROC::geom\_roc(n.cuts = 0)**

**plot\_curve<- plot\_curve +**

**# annotate("text", x = .75, y = .25, label = paste("AUC =", round(plotROC::calc\_auc(plot\_curve)["AUC"], 2))) +**

**annotate("text", x = .75, y = .25, label = paste("95 CI%:",round(auc\_ci[2],2),"-",round(auc\_ci[3],2)))**

**HL\_test<-ResourceSelection::hoslem.test(dades\_prediccio$event, dades\_prediccio$prediccio, g = 10)**

**popes<-list(taula\_OR=taula\_editada,forest\_plot=forest\_plot,ggplot\_ROC=plot\_curve,auc=auc,auc\_ci=auc\_ci,HL\_test=HL\_test)**

**}**

**#**

**# Resum d'un data.table (Mitjana, DT, N etc...) --------------------**

**###### RESUM D'UN DATA.TABLE**

**### LLANÇO UN DT, VARIABLE I UNA ESTRATIFICACIó I EM TORNA UN DT AMB un resum**

**### mitjana, DT, N etc... per cada ESTRAT**

**resum3<-function(dt=dades,x="val\_last.HBA1C",estrat="constant"){**

**dt$constant<-1**

**e<-parse(text=x)**

**resum3<-dt[, .(**

**Mean=mean(eval(e),na.rm=T),**

**SD=sd(eval(e),na.rm=T),**

**Nmenor7=sum(eval(e)<7,na.rm=T),**

**Perc\_menor7=(sum(eval(e)<7,na.rm=T)/length(which(eval(e) != "NA")))\*100,**

**N=length(eval(e))**

**)**

**,by=estrat]**

**resum3**

**}**

**# Resum quanti -------------------------**

**##### funció que retorna un summary (mean, sd) de y en funció d'un grup**

**resum\_quanti<-function(dt=dades,y="valor\_basal.GLICADA",grup="constant") {**

**dt$constant=1**

**# dt=data\_long**

**# y="valor\_basal.GLICADA"**

**# grup="SEXE"**

**### extrect p valor**

**pepito=paste0("summary(aov(",y,"~",grup,",data=dt))[[1]][['Pr(>F)']]",sep="")**

**pvalor<-eval(parse(text=pepito))[1]**

**summ1 <- paste0('mean(', y, ',na.rm=T)')**

**summ2<-paste0('sd(',y,',na.rm=T)')**

**dt %>% dplyr::group\_by\_(grup) %>%**

**dplyr::summarise\_(mean=summ1,**

**sd=summ2,**

**n="n()") %>%**

**dplyr::mutate(p=pvalor) %>%**

**rename("group"=grup)**

**}**

**# ESTADISTICS RESUMS x grup x estrat ----------------------**

**# RETORNA ESTADISTICS RESUMS (mean, sd, p-valor --> ANOVA/t-test) X GRUP X ESTRAT**

**resum\_quanti\_estrat<-function(dt=dades,y="valor\_basal.GLICADA",grup="CODGLP1",estrat="HBA1C\_cat4"){**

**# dt=dades**

**# y="valor\_basal.GLICADA"**

**# grup="CODGLP1"**

**# estrat="HBA1C\_cat4"**

**# dt<-dt %>% dplyr::select\_if(names(.)%in%c(y,grup,estrat)) select\_if no funciona**

**dt<-dt %>% dplyr::select(c(y,grup,estrat))**

**if (!"estrat" %in% colnames(dt)) {**

**dt<-dt %>% mutate (overall="Overall")**

**estrat="overall"}**

**dt %>%**

**tidyr::drop\_na(y) %>%**

**dplyr::group\_by\_(estrat) %>%**

**dplyr::do(resum\_quanti(dt=.,y=y,grup=grup))**

**}**

**# Resum events ----------------------**

**################### Llan?o dades, event i temps i me fa un resum**

**resum\_events<-function(dades=dadestotal,evento="RD",temps="temps",valorevent="Si") {**

**# dades=dadesDF**

**# evento="EVENT\_MORT2014"**

**# temps="temps\_mortalitat"**

**# valorevent="1"**

**# dades=dades**

**# evento="RD"**

**# temps="TEMPS\_RD2"**

**Patients=length(dades[[evento]])**

**PYears=sum(dades[[temps]])**

**temps\_seguiment=mean(dades[[temps]])**

**N.Events=sum(dades[[evento]]==valorevent)**

**Event.rate=((N.Events/PYears)\*100)**

**IA=(N.Events/Patients)**

**resum<-cbind(Patients,PYears,temps\_seguiment,N.Events,Event.rate,IA)**

**resum**

**}**

**# Resum events ----------------------**

**resum\_events\_v2<-function(dades=dades,evento="RD",temps="temps") {**

**# dades=dadestotal**

**# evento="RD"**

**# temps="TEMPS\_RD2"**

**Patients=length(dades[[evento]])**

**PYears=sum(dades[[temps]])**

**temps\_seguiment=mean(dades[[temps]])**

**N=mean(dades[["N\_BREAK"]])**

**min=min(dades[["N\_BREAK"]])**

**max=max(dades[["N\_BREAK"]])**

**N.Events=sum(dades[[evento]])**

**Event.rate=(N.Events/PYears)\*100**

**IA=(N.Events/Patients)\*100**

**### Fusionar tot**

**resum<-cbind(Patients,PYears,temps\_seguiment,N,min,max,N.Events,Event.rate,IA)**

**resum**

**}**

**# Resum events per grup ------------------**

**########## Llanço dades, event, temps , grup i retorno un resum d'events per grups**

**resum\_events\_grup=function(d=dadestotal,evento="RD",temps="TEMPS\_RD2",grup="sexe") {**

**# d=dadestotal**

**# evento="RD"**

**# temps="TEMPS\_RD2"**

**# grup="sexe"**

**# valorevent="1"**

**pepito=paste0("as.factor(d$",grup,")")**

**dadesgrups<-d %>% split(eval(parse(text=pepito)))**

**temp<- dadesgrups %>%**

**map(~resum\_events\_v2(dades=.x,evento=evento,temps=temps)) %>%**

**map(as.data.frame) %>%**

**map\_df(bind\_rows,.id = "Group") %>%**

**as\_tibble()**

**}**

**# Llistat de Taules compare ------------------**

**# LLISTA DE noms de taules i retorna llista de taules comparatives**

**# Llanço una LLISTA de noms de taules que estan en el Conductor Variables i em retorna una llista de taules ###**

**llistadetaules.compare<-function(tablero=c("taula1","taula2","taula3","taula4","taula5"),y="sexe",variables = "variables.xls",dt=dades){**

**restab.llista<-list()**

**for (i in 1:length(tablero)) {**

**restab.llista[[i]]<-tablero[i] %>%**

**formula\_compare(y=y,taulavariables = variables) %>%**

**compareGroups(data=dt,include.miss = F,include.label=T) %>%**

**createTable(show.ratio = F, hide.no = c('NA','No'), show.p.overall=T,show.n=T,show.all=T)**

**}**

**restab.llista**

**}**

**# P-valors ajustats segons multiple test Comparations desde un objecte Compare groups ------------------**

**### Llanço un objecte compare groups i em retorna els p-valors + els ajustats en una taula**

**# p.adjust.methods**

**# c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY",**

**# "fdr", "none")**

**## Ajust BH**

**# The "BH" (aka "fdr") and "BY" method of Benjamini, Hochberg, and Yekutieli control the false discovery rate,**

**# the expected proportion of false discoveries amongst the rejected hypotheses.**

**# The false discovery rate is a less stringent condition than the family-wise error rate, so these methods are more powerful than the others.**

**Pvalors\_ajustats\_compare<-function(objecte\_compare=T1.1.2, metodo="BH",p="p.overall",Sig="No") {**

**# objecte\_compare=T2\_Lipos**

**# metodo = "bonferroni"**

**# metodo = "BH"**

**# p="p.overall"**

**# p="p.mul"**

**# Sig="No"**

**# 1. Extrect els p-valors**

**pvalors <- compareGroups::getResults(objecte\_compare, p)**

**# 2. Taula de p vals**

**pvals<-data.table(pvalors)**

**# 4. Ajusta p- valors**

**# pvals$Adjpvalor<-stats::p.adjust(pvalors, method = metodo)**

**pvals<-pvals[,1:ncol(pvals)] %>% map\_df(stats::p.adjust,method = metodo)**

**# # 5. Punt de tall**

**# pvals<-pvals %>% mutate\_all(sigBH=ifelse(Adjpvalor<0.05,"Sig","NS"))**

**# 5. Canviar a punts de tall si argument Sig="Yes"**

**if (any(Sig==c("Yes","Si",1))) pvals<-pvals %>%**

**mutate\_all(funs(ifelse(.<0.05,"Sig","NS")))**

**# 3. Posa noms**

**pvals$variable<-rownames(pvalors)**

**if (is.null(rownames(pvalors))) pvals$variable<-names(pvalors)**

**pvals %>% dplyr::select(variable,starts\_with('p'))**

**# # 6. Canviar noms**

**# pvals<-pvals %>% setNames(c("P.crude","Variable",paste0("Padj.",substr(metodo, 1,3)), paste0("Sig.",substr(metodo, 1,3))))**

**}**

**Pvalors\_ajustats\_taula<-function(objecte\_taula=OR.ajust, p.valors='p valor', metodo="BH") {**

**# objecte\_taula=taulacoef**

**# p.valors='P\_adj'**

**# metodo="bonferroni"**

**# objecte\_taula=pvals**

**# p.valors="p.No vs Yes"**

**# metodo="bonferroni"**

**# 0 Genero noms de l'objecte a crear**

**nomsnous<-c(names(objecte\_taula),paste0(p.valors,".",substr(metodo, 1,3)),paste0(p.valors,".Sig.",substr(metodo, 1,3)))**

**# 1. Extrec p-valors**

**pvalors <-objecte\_taula[[p.valors]]**

**p.num<-pvalors %>% as.numeric()**

**# 2. Calculo els p valors ajustats**

**pvals\_adj<-stats::p.adjust(p.num, method = metodo)**

**# 3. Ho fusiono amb la taula**

**objecte\_taula<-objecte\_taula %>% cbind(pvals\_adj)**

**# 4. Punt de tall**

**objecte\_taula<-objecte\_taula %>% mutate (sigBH=ifelse(pvals\_adj<0.05,"Sig","NS"))**

**# 6. Canviar noms**

**objecte\_taula<-objecte\_taula %>% setNames(nomsnous)**

**objecte\_taula %>% as\_tibble()**

**}**

**# Afegeix dataindex Dinamica o / Constant si no existeix------------**

**###### Funció que Afegeix dataindex Dinamica o / Constant si no existeix**

**### Entra BD Historic i surt BD Historic + dataindex**

**afegir\_dataindex<-function(dt\_historic,bd.dindex="20161231") {**

**# dt\_historic=dt**

**# bd.dindex=bd.dindex**

**# Si es una constant generar una columna constant**

**if (is.numeric(bd.dindex) | is.character(bd.dindex)){**

**rrr<-dt\_historic %>%**

**dplyr::mutate(dtindex=bd.dindex) %>%**

**data.table**

**}**

**# Si es una bd amb data index fusionar data index**

**if (!(is.numeric(bd.dindex) | is.character(bd.dindex))) {**

**# Fusionar a l'historic la data index movil**

**rrr<-dt\_historic %>%**

**dplyr::inner\_join(bd.dindex, by="idp") %>%**

**rename(dtindex=tidyselect::last\_col()) %>% ## Renomenar dtindex (última columna de bd.index)**

**data.table**

**}**

**rrr**

**}**

**# Agregar analitiques -----------------**

**#################### FUNCIÓ QUE LLANÇO 1. Data.table,**

**# 2. dataindex constant, / o data.frame amb idp + dataindex (caracter) ,**

**# 3. finestra de temps previ en dies**

**#################### RETORNA UN data.table amb dades agregades**

**agregar\_analitiques<-function(dt=ANALITIQUES,bd.dindex="20161231",finestra.dies=c(-Inf,Inf),sufix = c(".valor", ".dies")){**

**# dt =VARIABLES**

**# bd.dindex =dt\_index**

**# finestra.dies=c(-365,0)**

**# sufix = c(".valor", ".dies")**

**#### Afegir + data index (+dtindex) en l'historic de variables**

**print("Afegint dt.index")**

**dt<-afegir\_dataindex(dt,bd.dindex)**

**# Convertir dates a numeric**

**print ("Convertir dates a numeric")**

**if (class(dt$dat)!="Date") dt$dat=as.Date(as.character(dt$dat),format="%Y%m%d") %>% as.numeric()**

**if (class(dt$dat)=="Date") dt$dat=as.numeric(dt$dat)**

**if (class(dt$dtindex)!="Date") dt$dtindex=as.Date(as.character(dt$dtindex),format="%Y%m%d") %>% as.numeric()**

**if (class(dt$dtindex)=="Date") dt$dtindex=as.numeric(dt$dtindex)**

**##### filtrar per intervals de dates**

**print("Filtrant dates")**

**dt<-dt %>% dplyr::filter(dat>= dtindex +finestra.dies[1] &**

**dat<= dtindex +finestra.dies[2])**

**print ("Seleccionant unic registre per variable-id")**

**## Selecciono un unic registre i agrego amb valor valid més proper dins de la finestra**

**paco<- dt %>% filter(val!=-9) %>%**

**dplyr::mutate(dies=dtindex -dat) %>% # Calculo els dies fins data index**

**dplyr::group\_by(idp,dtindex,cod) %>% # Agafo fila tal que dies sigui mínim (Valor mes proper)**

**dplyr::slice(which.min(dies)) %>% # Fila que els dies sigui més propera a data index**

**dplyr::ungroup()**

**print ("Reshaping")**

**# RESHAPE valors d'Analitiques**

**analitiques.valor <- paco[,c("idp","dtindex","cod","val")] %>%**

**tidyr::spread(cod,val)**

**# RESHAPE Dies**

**analitiques.dies <- paco[,c("idp","dtindex","cod","dies")] %>%**

**tidyr::spread(cod,dies)**

**print ("Join: valor+dies")**

**# JOINT Valors i dies**

**analitiques.idp<-full\_join(analitiques.valor, analitiques.dies, by=c("idp","dtindex"),suffix = sufix)**

**analitiques.idp**

**}**

**# Agregar\_problemes -----------------**

**################### LLANCO PROBLEMES LONG + UNA DATA INDEX / BD ab data index, finestra temporal ->**

**# RETORNO UNA TAULA AGREGADA / TAMBÉ POSO LA TAULA CATALEG**

**agregar\_problemes<-function(dt=PROBLEMES,bd.dindex="20161231",dt.agregadors=CATALEG,finestra.dies=c(-Inf,0),prefix="DG.",camp\_agregador="agr") {**

**# dt=HISTORIC\_EVENTOS\_TOTAL**

**# bd.dindex =IDS\_CIPS**

**# dt.agregadors=dt\_cataleg**

**# finestra.dies=c(-Inf,0)**

**# finestra.dies = c(0,+Inf)**

**# prefix = "DG."**

**# camp\_agregador = "Agrupador1"**

**## afegir en dataindex de BDINDEX si bd.dindex<>""**

**#### Afegir + data index (+dtindex) en l'historic de problemes**

**dt<-afegir\_dataindex(dt,bd.dindex)**

**## filtrar per intervals de dates**

**# Convertir dates a numeric**

**if (class(dt$dat)=="Date") dt$dat\_num=as.numeric(dt$dat)**

**if (class(dt$dtindex)=="Date") dt$dtindex\_num=as.numeric(dt$dtindex)**

**if (class(dt$dat)!="Date") dt$dat\_num=as.Date(as.character(dt$dat),format="%Y%m%d") %>% as.numeric()**

**if (class(dt$dtindex)!="Date") dt$dtindex\_num=as.Date(as.character(dt$dtindex),format="%Y%m%d") %>% as.numeric()**

**dt<-dt %>% as\_tibble()**

**##### filtrar per intervals de dates**

**dt<-dt %>% dplyr::filter(dat\_num>= dtindex\_num +finestra.dies[1] &**

**dat\_num<= dtindex\_num +finestra.dies[2])**

**# dt<-dt[data.table::between(**

**# lubridate::ymd(dat),**

**# lubridate::ymd(dtindex)+finestra.dies[1],**

**# lubridate::ymd(dtindex)+finestra.dies[2])]**

**## Filtrar CATALEG PER CAMP AGREGADOR**

**camp\_agregador\_sym<-sym(camp\_agregador)**

**dt.agregadors<-dt.agregadors %>%**

**dplyr::select(cod,agr=!!camp\_agregador\_sym) %>%**

**filter(!is.na(agr))**

**## Capturar agregador**

**dt.temp<-dt %>%**

**# camps mínims que necessito per agregar**

**dplyr::select(c(idp,dtindex,cod,dat)) %>% # Selecciono camps mínims**

**# Capturo Agregador de CATALEG**

**dplyr::inner\_join(dplyr::select(dt.agregadors,c(cod,agr)), by="cod") %>% # Capturo agregador del cataleg**

**# Eliminar duplicats agafant el primer registre (dat minima)**

**# Agrupar= unic reg per idp-agr (mes antic segons data)**

**dplyr::group\_by(idp,dtindex,agr) %>% # Agrupo per idp agr**

**dplyr::slice(which.min(dat)) %>% # Selecciono més antic**

**dplyr::ungroup() # desagrupo**

**# RESHAPE una data per agregador**

**# seleccionar camps i Reshape**

**dt.agregat<-dt.temp %>%**

**dplyr::select(idp,agr,dat,dtindex) %>% # Selecciono agregador i data**

**# RESHAPE per agregador i em quedo la data**

**tidyr::spread(agr,dat,sep=".") # Reshape**

**names(dt.agregat) <- sub("agr.", prefix, names(dt.agregat)) # Afegir prefix en noms de variables**

**dt.agregat**

**}**

**# Agregar\_problemes un sol agregador -----------------**

**################### LLANCO PROBLEMES LONG + UNA DATA INDEX / Un agregador / BD ab data index, finestra temporal ->**

**# RETORNO UNA TAULA AGREGADA / DAta i Codi/ TAMBÉ POSO LA TAULA CATALEG**

**agregar\_problemes\_agr<-function(dt=PROBLEMES,agregador="ECV",camp\_agregador="AGR\_TER",bd.dindex="20161231",dt.agregadors=CATALEG,finestra.dies=c(-Inf,0),prefix="") {**

**# bd.dindex="20161231"**

**# bd.dindex=bd\_dtindex**

**# dt=PROBLEMES**

**# agregador ="prevalent"**

**# dt.agregadors=CATALEG**

**# finestra.dies=c(-Inf,0)**

**# prefix=""**

**# camp\_agregador="exposed"**

**## afegir en dataindex de BDINDEX si bd.dindex<>""**

**#### Afegir + data index (+dtindex) en l'historic de problemes**

**dt<-afegir\_dataindex(dt,bd.dindex)**

**## filtrar per intervals de dates**

**# # Convertir dates() a numeric**

**# dt<-dt %>% mutate(**

**# dat=as.Date(as.character(dat),format="%Y%m%d") %>% as.numeric(),**

**# dtindex=as.Date(as.character(dtindex),format="%Y%m%d") %>% as.numeric()) %>%**

**# as\_tibble()**

**# Convertir dates a numeric si son numeriques**

**if (class(dt$dat)=="Date") dt$dat\_num=as.numeric(dt$dat)**

**if (class(dt$dtindex)=="Date") dt$dtindex\_num=as.numeric(dt$dtindex)**

**if (class(dt$dat)!="Date") dt$dat\_num=as.Date(as.character(dt$dat),format="%Y%m%d") %>% as.numeric()**

**if (class(dt$dtindex)!="Date") dt$dtindex\_num=as.Date(as.character(dt$dtindex),format="%Y%m%d") %>% as.numeric()**

**##### filtrar per intervals de dates**

**dt<-dt %>% dplyr::filter(dat\_num>= dtindex\_num +finestra.dies[1] &**

**dat\_num<= dtindex\_num +finestra.dies[2])**

**## Filtrar CATALEG**

**camp\_agregador\_sym<-sym(camp\_agregador)**

**dt.agregadors<-dt.agregadors %>%**

**dplyr::select(cod,agr=!!camp\_agregador\_sym) %>%**

**filter(agr==agregador)**

**## Capturar agregador**

**dt.temp<-dt %>%**

**# camps mínims que necessito per agregar**

**dplyr::select(c(idp,cod,dat)) %>% # Selecciono camps mínims**

**# Capturo Agregador de CATALEG**

**dplyr::inner\_join(dplyr::select(dt.agregadors,c(cod,agr)), by="cod") %>% # Capturo agregador del cataleg**

**# Eliminar duplicats agafant el primer registre (dat minima)**

**# Agrupar= unic reg per idp-agr (mes antic segons data)**

**dplyr::group\_by(idp,agr) %>% # Agrupo per idp agr**

**dplyr::slice(which.min(dat)) %>% # Selecciono més antic**

**dplyr::ungroup() # desagrupo**

**# RESHAPE una data per agregador**

**# seleccionar camps i Reshape**

**dt.agregat<-dt.temp %>%**

**dplyr::select(idp,agr,dat,cod) %>% # Selecciono agregador i data**

**# RESHAPE per agregador i em quedo la data**

**tidyr::spread(agr,dat,sep=".") # Reshape**

**names(dt.agregat) <- sub("agr.", prefix, names(dt.agregat)) # Afegir prefix en noms de variables**

**dt.agregat**

**}**

**# agregar\_prescripcions ----------------------**

**# Retorna tibble (data.table) amb el temps de prescripció en una finestra o primera data per idp-dataindex / primera data**

**# Arguments: Historic de PRESCRIPCIONS, data index constant o data.table, agregadors de codis (tibble:cod agr), finestra de temps en dies (-365,0)**

**# Requereix:(idp,cod,dat,dbaixa(yyyymmdd)) i Cataleg d'agrupadors amb cod, agr**

**#**

**agregar\_prescripcions<-function(dt=PRESCRIPCIONS,bd.dindex=20161231,dt.agregadors=CATALEG,prefix="FP.",finestra.dies=c(0,0),camp\_agregador="agr",agregar\_data=F){**

**# camp\_agregador="agr"**

**# agregar\_data=F**

**# dt=FX.PRESCRITS**

**# prefix = "FD."**

**# bd.dindex =bdades\_index**

**# finestra.dies=c(-45,-45)**

**# dt.agregadors=CATALEG**

**# Recode numeros infinits**

**finestra.dies=ifelse(finestra.dies==+Inf,99999,finestra.dies)**

**finestra.dies=ifelse(finestra.dies==-Inf,-99999,finestra.dies)**

**## afegir en dataindex de BDINDEX si bd.dindex<>""**

**#### Afegir + data index (+dtindex) en l'historic de problemes**

**dt<-afegir\_dataindex(dt,bd.dindex)**

**##### Arreglar dades**

**dt<-dt %>% mutate(**

**dat=lubridate::ymd(dat),**

**dbaixa=ifelse(is.na(dbaixa),30160101,dbaixa),**

**dbaixa=lubridate::ymd(dbaixa),**

**dtindex=lubridate::ymd(dtindex))**

**## arreglar CATALEG**

**dt.agregadors<-dt.agregadors %>% select\_("cod","agr"=camp\_agregador)**

**dt.agregadors<-dt.agregadors %>% filter(!is.na(agr))**

**prescripcions\_agr<-dt %>%**

**dplyr::select(idp,dtindex,cod,dat,dbaixa) %>%**

**# Calculo els dies de solapament per codi (cod)**

**dplyr::mutate(overlap = pmax(pmin(dtindex+lubridate::days(finestra.dies[2]), dbaixa) - pmax(dtindex+lubridate::days(finestra.dies[1]), dat) + 1,0),**

**overlap=as.numeric(overlap)) %>%**

**filter(overlap>0) # Elimino els que no xafen la finestra (overlap==0)**

**# Capturo l'agregador cataleg i elimino repetits**

**prescripcions\_agr<-prescripcions\_agr %>%**

**dplyr::inner\_join(dplyr::select(dt.agregadors,c(cod,agr)), by="cod") %>% # Capturo agregador del cataleg**

**dplyr::distinct(idp,dtindex,cod,agr,.keep\_all = TRUE) # Eliminar duplicats PER idp-dtindex-cod-agr**

**# Agregació de temps acumulats (dies) o primera data dins finestra**

**if (!(agregar\_data)) {**

**# suma dies acumulats**

**prescripcions\_agr<-prescripcions\_agr %>%**

**dplyr::group\_by(idp,dtindex,agr) %>%**

**dplyr::summarise(FX=sum(overlap,na.rm=T)) %>%**

**dplyr::ungroup() }**

**# Si s'ha d'agregar la primera data de prescripció dins finestra de temps**

**if (agregar\_data) {**

**# Selecciono primera data dins de l'interval**

**prescripcions\_agr <- prescripcions\_agr %>%**

**dplyr::mutate (**

**int1=dtindex+lubridate::days(finestra.dies[1]),**

**data0=ifelse(dat>=int1,dat,int1), # Si solapament inclou tota la finestra afago limit inferior de la finestra**

**data0=lubridate::as\_date(data0)) %>%**

**as\_tibble() %>%**

**dplyr::select(idp,dtindex,agr,dat=data0) %>%**

**dplyr::group\_by(idp,dtindex,agr) %>%**

**dplyr::slice(which.min(dat)) %>% #**

**dplyr::ungroup() %>%**

**dplyr::rename(FX=dat)}**

**# Aplanamenta**

**prescripcions\_agr<-prescripcions\_agr %>% tidyr::spread(agr,FX,sep=".")**

**# Canvi de noms**

**names(prescripcions\_agr) <- sub("agr.", prefix, names(prescripcions\_agr)) # Afegir prefix en noms de variables**

**prescripcions\_agr**

**}**

**# agregar\_facturacio -------------------**

**# Retorna tibble (data.table) amb la suma d'envasos o data primera dispensació dins d'una finestra de temps per idp-dataindex**

**# Arguments: historic de facturacions (PRESCRIPCIONS) , data index constant o data.table, agregadors de codis (tibble:cod agr), finestra de temps en dies (-365,0)**

**# Requereix dt=(idp,cod,env,dat(yyyymm)) i Cataleg d'agrupadors amb cod, agr**

**agregar\_facturacio<-function(dt=PRESCRIPCIONS,finestra.dies=c(-365,0),dt.agregadors=CATALEG,bd.dindex="20161231",prefix="FD.",camp\_agregador="agr", agregar\_data=F){**

**# dt=FX.FACTURATS\_PRESCRITS**

**# finestra.dies = c(-Inf,+Inf)**

**# camp\_agregador="GRUP"**

**# dt.agregadors = conductor\_variables**

**# bd.dindex = "20171231"**

**# prefix="FD."**

**# agregar\_data=T**

**agregador\_sym<-sym(camp\_agregador)**

**## Filtrar CATALEG per agrupador per camp\_agregador**

**dt.agregadors<-dt.agregadors %>% dplyr::select(cod,agr=!!agregador\_sym)**

**dt.agregadors<-dt.agregadors %>% filter(!is.na(agr))**

**# filtrar dt farmacs només per agregadors d'interes (camp\_agregador)**

**print("Filtrant per farmac agregador")**

**dt<-dt %>% semi\_join(dt.agregadors, by="cod")**

**#### Afegir data index en l'historic de farmacs**

**print("Afegint data index en historic de farmacs")**

**dt<-afegir\_dataindex(dt,bd.dindex)**

**# Si no existeix agr el creo**

**if (!("agr" %in% colnames(dt))) { dt<-dt %>% mutate(agr=NA) }**

**#### Filtrar dt per finestra temporal i genero data i datafi**

**print("Filtrant historic per finestra temporal i generant data i datafi")**

**# Recode els numeros infinits**

**finestra.dies=ifelse(finestra.dies==+Inf,99999,finestra.dies)**

**finestra.dies=ifelse(finestra.dies==-Inf,-99999,finestra.dies)**

**##**

**# Elimino funcions de llibreria lubridate**

**pepito<-dt %>% dplyr::mutate (**

**data=paste0(as.character(dat),"15") %>% as.Date('%Y%m%d'), # Data arrodonida al dia 15**

**datafi=data+(env\*30), # Genero data fi en funció dels envasos**

**dtindex=as.Date(dtindex,'%Y%m%d'))**

**# Estimo el nombre d'envasos de solapament per codi i agrego per codi diferent**

**print ("Estimo el nombre d'envasos de solapament per codi i agrego per codi diferent")**

**pepito<-pepito %>%**

**dplyr::mutate(interval2=dtindex+finestra.dies[2],**

**interval1=dtindex+finestra.dies[1],**

**overlap = pmax(pmin(interval2, datafi) - pmax(interval1,data) + 1,0),**

**overlap=as.numeric(overlap),**

**env2=overlap/30) %>%**

**dplyr::select(-agr,-dat,-interval2,-interval1,env,-env,env=env2) %>% # Netejo variables**

**filter(env>0.05) # Selecciono files amb solapament d'envasos dins finestra (Elimino env>0.05)**

**# Capturo Agregador de CATALEG i elimino duplicats**

**print("Capturo Agregador de CATALEG i elimino duplicats")**

**pepito<- pepito %>%**

**dplyr::inner\_join(dplyr::select(dt.agregadors,c(cod,agr)), by="cod") %>% # Capturo agregador del cataleg**

**dplyr::distinct(idp,dtindex,cod,agr,data,datafi,.keep\_all = TRUE) %>% # Elimino duplicats per idp-dtindex-cod-agr**

**as\_tibble()**

**# Agregació de nombre d'envasos per defecte**

**print("Agregant facturacio")**

**if (!(agregar\_data)) {**

**dt\_agregada <- pepito %>% # Agrego --> Suma de numero d'envasos per idp-dtindex-agr**

**dplyr::select(c(idp,dtindex,agr,env)) %>%**

**as\_tibble() %>%**

**dplyr::group\_by(idp,dtindex,agr) %>%**

**dplyr::summarise(FX=sum(env,na.rm=T)) %>%**

**dplyr::ungroup()**

**}**

**# Si s'ha d'agregar data primera Facturació**

**if (agregar\_data){**

**dt\_agregada <- pepito %>% # Agrego --> data mínima**

**mutate(**

**int1=dtindex+finestra.dies[1], # Si solapament inclou tota la finestra afago limit inferior de la finestra**

**data0=ifelse(data>=int1,data,int1)) %>%**

**as\_tibble() %>%**

**dplyr::select(c(idp,dtindex,agr,data=data0)) %>%**

**dplyr::group\_by(idp,dtindex,agr) %>%**

**dplyr::slice(which.min(data)) %>%**

**dplyr::ungroup() %>%**

**dplyr::rename(FX=data) %>%**

**mutate(FX=as.Date(FX,origin = "1970-01-01"))**

**}**

**# Aplanamenta**

**print("Aplanamenta")**

**dt\_agregada<-dt\_agregada %>%**

**tidyr::spread(agr,FX,sep=".") %>%**

**mutate\_if(is.numeric, funs(ifelse(is.na(.), 0, .)))**

**# mutate\_if(is.numeric, list(ifelse(is.na(.), 0, .)))**

**names(dt\_agregada) <- sub("agr.", prefix, names(dt\_agregada)) # Afegir prefix a noms de variables**

**dt\_agregada**

**}**

**# AGREGADOR DE VISITES ###############**

**### Envio la historic de visites i retorno numero de visites en la finestra de temps**

**agregar\_visites<-function(dt=VISITES,bd.dindex=20161231,finestra.dies=c(-365,0)){**

**# dt=VISITES**

**# bd.dindex = bdades\_index**

**# finestra.dies=c(-365,-45)**

**## Afegir en dataindex (+dtindex) en historic de Visites**

**dt<-afegir\_dataindex(dt,bd.dindex)**

**##### filtrar per intervals de dates**

**# Convertir dates a numeric**

**dt<-dt %>% mutate(dat=as.Date(as.character(dat),format="%Y%m%d") %>% as.numeric(),**

**dtindex=dtindex %>% as.numeric()) %>% as\_tibble()**

**##### filtrar per intervals de dates**

**dt<-dt %>% dplyr::filter(dat>= dtindex +finestra.dies[1] &**

**dat<= dtindex +finestra.dies[2])**

**# dt<-dt[data.table::between(**

**# lubridate::ymd(dat),**

**# lubridate::ymd(dtindex)+finestra.dies[1],**

**# lubridate::ymd(dtindex)+finestra.dies[2])]**

**##### Agregar (Suma de visites en interval independentment del tipus)**

**paco<-dt %>%**

**dplyr::group\_by(idp,dtindex,cod) %>% # Agrupo per id**

**dplyr::count() %>% # Conto el numero visites per codi**

**dplyr::ungroup()**

**# RESHAPE per idp**

**visites <- paco[,c("idp","dtindex","cod","n")] %>%**

**dplyr::select(idp,dtindex,visites=cod,n) %>%**

**tidyr::spread(visites,n,sep = "\_")**

**paco <- paco %>%**

**dplyr::select(idp,dtindex,visites=cod,n) %>%**

**tidyr::spread(visites,n,sep = "\_")**

**# NA = 0**

**visites[is.na(paco)]<-0**

**### Computo visites globals**

**paco<-paco %>% dplyr::select(idp,dtindex) # Separo id de visites**

**visites<-visites %>% # Sumo totes les visites**

**dplyr::select(starts\_with("visites")) %>%**

**mutate(visites\_TOT=rowSums(.) )**

**paco<-paco %>% cbind(visites) %>% as\_tibble()**

**paco**

**}**

**# APLICA CRITERIS D'EXCLUSIÓ A dades -----------------------**

**criteris\_exclusio<-function(dt=dades,taulavariables="VARIABLES\_R3b.xls",criteris="exclusio1") {**

**# dt=dades**

**# taulavariables="VARIABLES\_R3b.xls"**

**# criteris="exclusio1"**

**## Llegeix criteris de variables**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**## 0. Filtro taula variables només variables implicades en el filtre i el genero**

**maco<-variables %>%**

**dplyr::filter\_(paste0(criteris,"!=0")) %>%**

**dplyr::select\_("camp",criteris) %>%**

**# Genero la llista de filtres**

**tidyr::unite\_("filtres", c("camp", criteris),sep="")**

**# Concateno condicions amb un OR**

**maco<-str\_c(maco$filtres,collapse=" | ")**

**## 1. Genera filtre en base a columna exclusio1 popes**

**popes<-str\_c("!(",maco,")")**

**## 2. Eliminar els espais en blanc de les variables factors del data.frame**

**dt<-dt %>%**

**dplyr::mutate\_if(is.factor,funs(str\_trim(.)))**

**## 3. Aplicar filtre: popes a dt**

**dt<-dt %>%**

**dplyr::filter(eval(parse(text=popes)))**

**}**

**# FLOW-CHART A partir de criteris d'exclusió en taulavariable -----------------------------------**

**criteris\_exclusio\_diagrama<-function(dt=dades,taulavariables="VARIABLES\_R3b.xls",criteris="exclusio1",**

**pob\_lab=c("Pob inicial","Pob final"),etiquetes="etiqueta\_exclusio",ordre="exc\_ordre",grups=NA){**

**# dt=dadesinicials**

**# taulavariables = conductor\_variables**

**# criteris = "exclusio"**

**# etiquetes="exc\_lab"**

**# grups="grup"**

**# pob\_lab=c("Pob inicial","Pob final")**

**# ordre="exc\_ordre"**

**### Si hi ha grups capturar el nombre categories**

**# Per defecte UN sol grup**

**ngrups=1**

**# ngrups>1**

**if (!is.na(grups)) {**

**ngrups=length(table(dt[grups]))**

**Etiqueta\_pob\_inicial=pob\_lab[1]**

**Npob\_inicial=dt %>% count() %>% as.numeric()**

**}**

**## Llegeixo criteris de variables i selecciono variables amb filtres**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**variables<-variables %>% dplyr::filter\_(paste0(criteris,"!=0"))**

**# variables<-variables %>% select\_if(names(.)%in%c("camp",criteris,etiquetes,ordre)) select\_if no funciona**

**variables<-variables %>% dplyr::select(c("camp",criteris,etiquetes,ordre))**

**## Elimino els espais en blanc de les variables factor**

**dt<-dt %>% dplyr::mutate\_if(is.factor,funs(str\_trim(.))) %>% as.data.table()**

**## Selecciono dades només de les variables implicades en el filtres**

**llista\_camps<-variables["camp"]**

**## Dades amb variables implicades en els filtres**

**if (is.na(grups)) {dt<-dt %>% mutate(grup="constant")}**

**if (is.na(grups)) {grups="grup"}**

**datatemp<-dt %>% dplyr::select(c(variables[["camp"]],grups)) %>% as\_tibble %>% rename\_("grup"=grups)**

**# Genero filtres**

**maco\_noms<-variables["camp"]**

**# Genero la llista de filtres**

**maco\_criteris<-variables %>%**

**dplyr::select\_("camp",criteris) %>%**

**tidyr::unite\_("filtres", c("camp", criteris),sep="")**

**maco\_criteris<-maco\_noms %>% cbind(maco\_criteris) %>% mutate(tipus\_cri="pur")**

**maco\_miss<-variables %>%**

**dplyr::select\_("camp") %>%**

**dplyr::mutate(filtres=paste0("is.na(",OR2=camp,")",sep="")) %>% dplyr::select(filtres)**

**maco\_miss<-maco\_noms %>% cbind(maco\_miss) %>% mutate(tipus\_cri="missing")**

**maco\_criteris<-maco\_criteris %>% rbind(maco\_miss) %>% arrange\_("camp")**

**## Generar taula amb dades per cada criteri d'exclusió**

**num\_criteris<-data.frame()**

**## Generar dades dels critersi criteris**

**for (i in 1: length(maco\_criteris$filtres)){**

**dades\_criteris<-datatemp %>%**

**dplyr::filter\_(as.character(maco\_criteris[i,]$filtres)) %>%**

**group\_by(grup) %>% summarise (n=n(),any(n)) %>% mutate(criteri=i) %>%**

**mutate(camp=maco\_criteris[i,]$camp,**

**filtre\_tipus=maco\_criteris[i,]$tipus\_cri,**

**filtre\_forma=maco\_criteris[i,]$filtres**

**) %>%**

**ungroup**

**num\_criteris<-num\_criteris %>% rbind(dades\_criteris)**

**}**

**# Expandir per tenir una fila per criteri amb valor**

**taula\_criteris<-num\_criteris %>%**

**expand(grup,camp,filtre\_tipus) %>%**

**left\_join(num\_criteris,by=c("grup","camp","filtre\_tipus"))**

**# Netejar aquelles files que no tinguin cap 0 en cap dels grups**

**temp<-taula\_criteris %>% mutate(n=ifelse(is.na(n),0,n)) %>%**

**group\_by(camp,filtre\_tipus) %>%**

**summarise(suma\_n=sum(n))**

**taula\_criteris<-taula\_criteris %>%**

**left\_join(temp,by=c("camp","filtre\_tipus")) %>%**

**filter(suma\_n!=0) %>%**

**dplyr::select(-suma\_n) %>%**

**mutate (n=ifelse(is.na(n),0,n))**

**# Afegir etiquetes a num\_criteris**

**# Etiquetes dels criteris d'inclusio**

**taula\_etiquetes<- variables %>% dplyr::select\_("camp",etiquetes) %>% rename\_("etiqueta\_exclusio"=etiquetes)**

**taula\_criteris<-taula\_criteris %>% left\_join(taula\_etiquetes,by="camp")**

**taula\_criteris<-taula\_criteris %>% mutate(etiqueta\_exclusio=ifelse(filtre\_tipus=="missing",paste0("Excluded NA:",camp),etiqueta\_exclusio))**

**taula\_criteris<-taula\_criteris %>% arrange(camp,filtre\_tipus,grup)**

**## I ara passar informació generada a vectors per passar-ho al diagrameR**

**# N d'esclusions**

**n\_exc<-taula\_criteris[c("n","grup")] %>% split(.$grup)**

**# Etiquetes d'exclusions**

**lab\_exc<-taula\_criteris[c("etiqueta\_exclusio","grup")] %>% split(.$grup)**

**lab\_exc<-lab\_exc[[1]]$etiqueta\_exclusio %>% as\_vector**

**# Calcular N població final per cada grup (3x1)**

**# Generar FILTRE**

**filtre\_total<-stringr::str\_c(maco\_criteris$filtres,collapse=" | ")**

**filtre\_total<-stringr::str\_c("!(",filtre\_total,")")**

**# Eliminar els espais en blanc de les variables factors del data.frame**

**datatemp<-datatemp %>% dplyr::mutate\_if(is.factor,funs(str\_trim(.)))**

**# Aplicar FILTRE**

**datatemp<-datatemp %>% dplyr::filter(eval(parse(text=filtre\_total)))**

**# Generar N per cada grup Inicial i Final x grup**

**pob.i<-dt %>% group\_by(grup) %>% summarise (n=n()) %>% dplyr::select(n) %>% as.vector**

**pob.f<-datatemp %>% group\_by(grup) %>% summarise (n=n()) %>% dplyr::select(n) %>% as.vector**

**n\_pob1<-c(pob.i$n[1],pob.f$n[1])**

**n\_pob2<-c(pob.i$n[2],pob.f$n[2])**

**n\_pob3<-c(pob.i$n[3],pob.f$n[3])**

**# Generar Etiquetes: Pob inicial i final x grup**

**# Etiquetes grups**

**pob\_lab\_grup1<-c(paste0("Group: ",taula\_criteris[["grup"]][1]),paste0("Group: ",taula\_criteris[["grup"]][1]))**

**pob\_lab\_grup2<-c(paste0("Group: ",taula\_criteris[["grup"]][2]),paste0("Group: ",taula\_criteris[["grup"]][2]))**

**pob\_lab\_grup3<-c(paste0("Group: ",taula\_criteris[["grup"]][3]),paste0("Group: ",taula\_criteris[["grup"]][3]))**

**# Si només hi ha un grup pob inicial es parametres inicials**

**if (ngrups==1) {pob\_lab\_grup1=pob\_lab}**

**if (ngrups==1) {Npob\_inicial=n\_pob1}**

**# Crido diagrama**

**diagrama<-diagramaFlowchart(grups=ngrups,**

**pob=Npob\_inicial,**

**pob\_lab=Etiqueta\_pob\_inicial,**

**exc1=n\_exc[[1]]$n %>% as\_vector,**

**exc2=n\_exc[[2]]$n %>% as\_vector,**

**exc3=n\_exc[[3]]$n %>% as\_vector,**

**exc\_lab1=lab\_exc,**

**exc\_lab2=lab\_exc,**

**exc\_lab3=lab\_exc,**

**pob1=n\_pob1,**

**pob2=n\_pob2,**

**pob3=n\_pob3,**

**pob\_lab1=pob\_lab\_grup1,**

**pob\_lab2=pob\_lab\_grup2,**

**pob\_lab3=pob\_lab\_grup3**

**)**

**diagrama**

**}**

**# CALCULA LA PROPORCIÓ -- RETORNA N I % fila ----------------**

**calcular\_proporcio<-function(dt=dades,factor="canvi612M.glicadaCAT2"){**

**# dt=dades**

**# factor="canvi612M.glicadaCAT2"**

**# cat="Yes"**

**moco<-dt %>%**

**tidyr::drop\_na(factor) %>%**

**dplyr::group\_by\_(factor) %>%**

**dplyr::summarise\_(n="n()") %>%**

**dplyr::mutate\_(freq="n/sum(n)\*100")**

**moco**

**}**

**# CALCULA PROPORCIO PER GRUPS I RETORNA P VALOR --------------**

**proporcions\_grups<-function(dt=dades,factor="canvi612M.glicadaCAT2",estrat="SEXE"){**

**# dt=dades**

**# factor="canvi612M.glicadaCAT2"**

**# estrat="CODGLP1"**

**## extrec p-valor**

**pepito=paste0("chisq.test(dt$",factor,",dt$",estrat,")$p.value",sep="")**

**pvalor<-eval(parse(text=pepito))**

**resultat<-**

**dt %>%**

**tidyr::drop\_na(factor) %>%**

**dplyr::group\_by\_(estrat) %>%**

**dplyr::do(calcular\_proporcio(dt=.,factor=factor)) %>%**

**dplyr::mutate(p=pvalor)**

**resultat**

**}**

**# RETORNA UNA LLISTA DE TAULES DE PROPORCIONS PER GRUPS ESTRATIFICAT PER estratificat ----------**

**proporcio\_grups\_estratificat<-function(dt=dades,factor.Y="canvi612M.glicadaCAT2",grup=c("SEXE","CODGLP1","anys\_DMcat4"),estratificat="HBA1C\_cat4") {**

**# dt=dades**

**# factor.Y="canvi612M.glicadaCAT2"**

**# grup=c("SEXE","anys\_DMcat4")**

**# estratificat="HBA1C\_cat4"**

**pepe<-list()**

**for (i in 1:length(grup)) {**

**pepe[[i]]<-**

**dt %>%**

**tidyr::drop\_na(estratificat) %>%**

**dplyr::group\_by\_(estratificat) %>%**

**dplyr::do(proporcions\_grups(dt=.,factor=factor.Y,estrat=grup[i]))**

**}**

**pepe**

**}**

**# REDUCCIÓ AJUSTADA DIFERENTS METODES D'AJUST-----------------**

**## BASAL , POST I RETORNA LA DIFERENCIA AJUSTA SEGONS EL BASAL I ERROR ESTANDARD**

**reduccio\_ajustada<-function(dt=dades,v.basal,v.final,mean.basal=NA) {**

**library(mgcv)**

**# # parametres**

**# dt=dades**

**# v.basal="HBpreADD"**

**# v.final="HBpostADD"**

**# mean.basal=9.02**

**## Si no poso la mitjana basal poso la mitjana de la base de dades**

**if (is.na(mean.basal)) mean.basal=mean(dt[,v.basal],na.rm=T)**

**# Calculo la variable canvi**

**dt<-dt %>%**

**dplyr::mutate(canvi=dt[,v.basal]-dt[,v.final])**

**# Genero quintils que no els faré servir de moment**

**dt<-dt %>%**

**dplyr::mutate(basal\_cat5=cut2(dt[,v.basal], g=5))**

**## Elimino missings de taula i selecciono variables**

**dt<-dt %>%**

**tidyr::drop\_na(canvi) %>%**

**dplyr::select\_(v.basal,v.final,"canvi","basal\_cat5")**

**## canvio noms que tampoc caldria**

**names(dt)<-c("pre","post","dif","basal\_cat")**

**## model cru (descriptiu bàsic,+ mean, se )**

**taula<-dt %>% summarise(**

**n=n(),**

**mean.basal=mean(pre),**

**mean.canvi=mean(dif),**

**se=sd(dif)/sqrt(n())**

**)**

**### arguments de funcions dels models amb les dades, junt amb la mean.basal**

**pre<-dt$pre**

**dif<-dt$dif**

**# funcions dels models**

**model.lineal.w<-function(y=y,x=x)glm(y~x,weights =x,family = gaussian)**

**model.lineal<-function(y=y,x=x) glm(y~x,family = gaussian)**

**model.nolineal<-function(y=y,x=x) glm(y~x+I(x^2)+I(x^3),family = gaussian)**

**model.gam1<-function(y=y,x=x) gam(y~s(x),family = gaussian)**

**model.gam2<-function(y=y,x=x) gam(y~s(x,bs="cc",k=12),family = gaussian)**

**# Genero els models que els poso en una llista**

**llista.models<-list(**

**lineal.w=model.lineal.w(x=pre,y=dif),**

**# lineal=model.lineal(x=pre,y=dif),**

**nolineal=model.nolineal(x=pre,y=dif),**

**gam1=model.gam1(x=pre,y=dif)**

**# , gam2=model.gam2(x=pre,y=dif)**

**)**

**predict(llista.models[[1]],data.frame(x=mean.basal))**

**## Genero les prediccions () en el punt basal mitg i guardo el la SE**

**maquina<-llista.models %>%**

**purrr::map\_df(predict,data.frame(x=mean.basal),se.fit=T) %>%**

**as.data.frame()**

**## Calculo Rquadrat per cada model**

**Rquadrat<-llista.models %>%**

**purrr::map\_dbl(function(modelaco) (1-(modelaco$deviance/modelaco$null.deviance))) %>%**

**as.data.frame()**

**## Combino informació Rquadrat + prediccions de cada model**

**taula.models<-cbind(Rquadrat,maquina)**

**## poso els noms dels models com una columna**

**taula.models$model<-row.names(taula.models)**

**## enganxo la taula dels valors mitjans i la N**

**taula<-cbind(taula.models,taula,v.basal)**

**## Hauria de canviar el nom de les variables una mica i eliminar coses que no serveixen i tal pascual**

**names(taula)[1] <- "R.Square"**

**# Drop variables with -**

**# taula<-select(taula, -("residual.scale"))**

**taula**

**}**

**# Predicció ajustada amb dades imputades -----------------**

**# Envio un dades generades amb MICE , X Y i retorna les prediccions amb ES ###**

**glance.prediction = function(x) {**

**data.frame(term = 'prediction',**

**estimate = x[['fit']],**

**std.error = x[['se.fit']],**

**df.residual = x[['df']]) }**

**tidy.prediction = function(x, effects = "fixed", exponentiate = FALSE)**

**{glance.prediction(x)}**

**retorn\_prediccio\_MI<-function(data\_imp=tempData,x="HBpreADD",y="canvi\_ADD",dades\_origen=dades) {**

**# data\_imp=tempData**

**# x="1"**

**# y="canvi\_ADD"**

**# dades\_origen=dades**

**imp<-tempData**

**nimp<-imp$m**

**if (x!="1") mean.basal=mean(dades\_origen[,x],na.rm=T)**

**if (x=="1") mean.basal=0**

**df.pred<-as.data.frame(mean.basal) %>% setNames(x)**

**texto=paste0(y,"~",x)**

**mods.imp = lapply(1:nimp, function(.nimp){**

**# m = lm(canvi\_ADD~HBpreADD, data = complete(imp, .nimp))**

**m=with(complete(imp,.nimp),lm(eval(parse(text=texto))))**

**mm = predict(m, newdata=df.pred, se.fit = TRUE)**

**structure(**

**mm,**

**class = 'prediction')**

**})**

**pp<-summary(mice::pool(as.mira(mods.imp)))**

**pp**

**}**

**retorn\_prediccio\_MI\_STR<-function(data\_imp=tempData,x="HBpreADD",y="canvi\_ADD",dades\_origen=dades,valor\_subset="<8",var\_subset="HBpreADD") {**

**# data\_imp=tempData**

**# x="1"**

**# y="canvi\_ADD"**

**# dades\_origen=dades**

**# valor\_subset="<8"**

**# var\_subset="HBpreADD"**

**subset<-paste0(var\_subset,valor\_subset)**

**imp<-tempData**

**nimp<-imp$m**

**if (x!="1") mean.basal=mean(dades\_origen[,x],na.rm=T)**

**if (x=="1") mean.basal=0**

**df.pred<-as.data.frame(mean.basal) %>% setNames(x)**

**texto=paste0(y,"~",x) # texte model**

**texte\_subset<-paste0("subset(complete(imp,.nimp),",subset,")")**

**mods.imp = lapply(1:nimp, function(.nimp){**

**# m = lm(canvi\_ADD~HBpreADD, data = complete(imp, .nimp))**

**# m=with(complete(imp,.nimp),lm(eval(parse(text=texto))))**

**m=with(eval(parse(text=texte\_subset)),lm(eval(parse(text=texto))))**

**mm = predict(m, newdata=df.pred, se.fit = TRUE)**

**structure(**

**mm,**

**class = 'prediction')**

**})**

**pp<-summary(mice::pool(as.mira(mods.imp)))**

**pp**

**}**

**retorn\_prediccio\_MI\_STR2<-function(data\_imp=tempData,x="HBpreADD",y="canvi\_ADD",dades\_origen=dades,valor\_subset1=">=8",valor\_subset2="<=10",var\_subset="HBpreADD") {**

**# data\_imp=tempData**

**# x="1"**

**# y="canvi\_ADD"**

**# dades\_origen=dades**

**# valor\_subset="<8"**

**# var\_subset="HBpreADD"**

**# valor\_subset1=">=8"**

**# valor\_subset2="<=10"**

**subset<-paste0(var\_subset,valor\_subset1," & ",var\_subset,valor\_subset2)**

**imp<-tempData**

**nimp<-imp$m**

**if (x!="1") mean.basal=mean(dades\_origen[,x],na.rm=T)**

**if (x=="1") mean.basal=0**

**df.pred<-as.data.frame(mean.basal) %>% setNames(x)**

**texto=paste0(y,"~",x) # texte model**

**texte\_subset<-paste0("subset(complete(imp,.nimp),",subset,")")**

**mods.imp = lapply(1:nimp, function(.nimp){**

**# m = lm(canvi\_ADD~HBpreADD, data = complete(imp, .nimp))**

**# m=with(complete(imp,.nimp),lm(eval(parse(text=texto))))**

**m=with(eval(parse(text=texte\_subset)),lm(eval(parse(text=texto))))**

**mm = predict(m, newdata=df.pred, se.fit = TRUE)**

**structure(**

**mm,**

**class = 'prediction')**

**})**

**pp<-summary(mice::pool(as.mira(mods.imp)))**

**pp**

**}**

**#**

**# PLOT dispersió segons PRE-POST , FA DISPERSIÓ DE PRE VS CANVI I SOBREPOSA AJUST------**

**plot.dispersio.reduccio <-function(dt=dades,v.basal="HBpreADD",v.final="HBpostADD") {**

**# library(mgcv)**

**# # parametres**

**# dt=dades**

**# v.basal="HBpreADD"**

**# v.final="HBpostADD"**

**dt <-dt %>% dplyr::mutate(canvi=dt[,v.basal]-dt[,v.final])**

**# Genero quintils**

**dt<-dt %>% dplyr::mutate(basal\_cat5=cut2(dt[,v.basal], g=5))**

**## Elimino missings de taula i selecciono variables**

**dt<-dt %>%**

**tidyr::drop\_na(canvi) %>%**

**dplyr::select\_(v.basal,v.final,"canvi","basal\_cat5")**

**## poso noms**

**names(dt)<-c("pre","post","dif","basal\_cat")**

**lineal<-glm(dif~pre,weights = pre,family = gaussian, data=dt) %>% predict()**

**lineal2<-glm(dif~pre,family = gaussian, data=dt) %>% predict()**

**gam1<-mgcv::gam(dif~s(pre),family = gaussian, data=dt) %>% predict()**

**gam2<-mgcv::gam(dif~s(pre,bs="cc",k=12),family = gaussian, data=dt) %>% predict()**

**model.nolineal<-glm(dif~pre+I(pre^2)+I(pre^3),family = gaussian,data=dt) %>% predict()**

**figuraZ<-dt %>%**

**ggplot2::ggplot(aes(x=pre, y=dif)) +**

**geom\_point() +**

**ylab("Change at 6-12 months:HbA1c (%)") +**

**xlab("HbA1c (%) Baseline") +**

**geom\_point(aes(y=lineal),color="red")+**

**geom\_point(aes(y=gam1),color="blue") +**

**geom\_point(aes(y=model.nolineal),color="green")+**

**ggtitle(paste0(v.basal," Versus ",v.final)) # for the main title**

**figuraZ**

**}**

**# Forest.plot --------------------**

**# A partir de taula amb OR's / Betas genera Forest Plot**

**# La taula ha de contenir els seguents camps:Categoria,OR,Linf,Lsup**

**forest.plot<-function(dadesmodel=ramo,label=dadesmodel$Categoria,mean=dadesmodel$OR,lower=dadesmodel$Linf,upper=dadesmodel$Lsup,label\_X="OR (95% CI)", intercept=1) {**

**# dadesmodel=taula\_coefs**

**# label=taula\_editada$Categoria**

**# mean=taula\_editada$OR**

**# lower=taula\_editada$Linf**

**# upper=taula\_editada$Lsup**

**# label\_X="OR (95% CI)"**

**# intercept=1**

**dadesmodel<-dadesmodel %>% mutate(id=seq(length(label),1))**

**fp <- ggplot(data=dadesmodel,aes(x=dadesmodel$id, y=dadesmodel$OR, ymin=dadesmodel$Linf, ymax=dadesmodel$Lsup)) +**

**geom\_pointrange() +**

**geom\_hline(yintercept=intercept, lty=2) + # add a dotted line at x=1 after flip**

**coord\_flip() + # flip coordinates (puts labels on y axis)**

**xlab("Label") + ylab(label\_X) +**

**scale\_x\_continuous(breaks=dadesmodel %>% pull(id) ,labels=dadesmodel %>% pull(Categoria))**

**fp**

**}**

**# DATA RANDOM ENTRE DUES DATES (dataini i datafi) ---------------**

**data.random <- function(dataini=20120101, datafi=20121231) {**

**# dataini=20120101**

**# datafi=20161231**

**dataini <- as.POSIXct(lubridate::ymd(dataini))**

**datafi <- as.POSIXct(lubridate::ymd(datafi))**

**temps <- as.numeric(difftime(datafi,dataini,unit="sec"))**

**# Genera Data sumant temps random a dataini**

**rt <- dataini + runif(1, 0, temps)**

**}**

**# RETORNA UNA DATA A STRING ------------------**

**data.to.string<-function(data) {**

**data.string=paste0(year(data),**

**str\_pad(lubridate::month(data),2,"left","0"),**

**str\_pad(lubridate::day(data),2,"left","0"))**

**}**

**# Data R Lubridate a partir de data UTC -----------**

**# Dades i x=Variable o vector de variables**

**dataUTC\_to\_Rdata<-function(x,dt) {**

**# dt<-dades**

**# x=c("data\_inici\_HD","ANT1\_ARTER\_PERI","ANT2\_ARTE\_PERI","ANT1\_CI")**

**# Seleccionar nom del camp si es tipo caracter**

**vector\_caracter<-dt %>% dplyr::select\_if(~!any(class(.)!="character",na.rm=F)) %>% names()**

**# Vectors de variables UTC (data POSIXct)**

**x\_UTC<-x [!x %in% vector\_caracter]**

**# Vector de variables caracter ("37712")**

**x\_text<-x [x %in% vector\_caracter]**

**# Funcio que converteix UTC data a date ymd**

**data\_convert\_UTC<-function(x){**

**x<-format(as.POSIXct(x, origin='1970-01-01'), format='%Y/%m/%d')**

**x<-lubridate::ymd(x)}**

**# Funcio que converteix data caracter ("37712) a date ymd () "2003-04-01"**

**data\_convert\_text<-function(x){**

**x<-as.Date(as.numeric(x), origin = "1899-12-30") %>%**

**lubridate::ymd()}**

**# Aplicar conversions als dos tipos de dates**

**dt<-dt %>% purrr::modify\_at(x\_UTC,~data\_convert\_UTC(.x)) # UTC ->date**

**dt<-dt %>% purrr::modify\_at(x\_text,~data\_convert\_text(.x)) # text->date**

**dt**

**}**

**# Funcio que converteix data caracter ("37712") a date ymd () "2003-04-01"**

**data\_convert\_text<-function(x){**

**x<-as.Date(as.numeric(x), origin = "1899-12-30") %>%**

**lubridate::ymd()}**

**# Funcio que converteix de numeric (15784) a Date "2013-03-20"**

**data\_convert\_numeric<-function(x){ x<-as.Date(x, origin = "1970-01-01")}**

**# Funcio que converteix UTC data a date ymd**

**data\_convert\_UTC<-function(x){**

**x<-format(as.POSIXct(x, origin='1970-01-01'), format='%Y/%m/%d')**

**x<-lubridate::ymd(x)}**

**# CONVERTEIX FORMAT TEXT A DATA -------------**

**#**

**# Format YYYYMMDD (Format text -> data)**

**# Input : dades, conductorvariables, campdata (com a indicadora (0/1))**

**convertir\_dates<-function(d=dadestotal,taulavariables="variables\_R.xls",campdata="dates")**

**{**

**#### Llegir etiquetes i variables a analitzar ##**

**variables <- readxl::read\_excel(taulavariables)**

**variables[is.na(variables)]<- 0**

**#**

**#**

**# etiquetar variables**

**seleccio<-variables**

**camp<- as.vector(seleccio$camp) #**

**for (i in 1:length(camp)){if (seleccio$dates[i]==1) {**

**pepito<-paste0("as.Date(d[[camp[",i,"]]], '%Y%d%m')")**

**d[[camp[i]]]<-eval(parse(text=pepito))**

**} }**

**d**

**}**

**# Passa data de SPSS a Rdata ----------------------------**

**# 13481683200 --> 2010-01-01**

**dataSPSS\_to\_Rdata <- function(x) {**

**y<-as.Date(x/86400, origin = "1582-10-14") %>%**

**lubridate::ymd() }**

**#**

**# Random dates i marcar potencials CONTROLS-----------**

**#**

**# Genero N dates random entre 2010-2016 el mateix nombre que**

**dt\_index\_data\_random<-function(dt=PACIENTS) {**

**# dt=PACIENTS**

**# Necessito camp dtsortida (ymd)**

**#### Genero una data random entre 01/01/2010 i 31/12/2016**

**set.seed(123)**

**data\_index\_data<-dt %>%**

**nrow() %>% runif(as.Date("10/01/01", "%y/%m/%d"), as.Date("16/12/31", "%y/%m/%d")) %>%**

**data.table() %>%**

**setNames(.,c("dtindex.random")) %>%**

**mutate (**

**dtindex.random=as.Date(dtindex.random, origin = "1970-01-01")**

**)**

**# Fusiono amb idp i selecciono POTENCIALS CONTROLS dins de periode de seguiment**

**BD\_PAC\_DINDEX<-dt %>%**

**dplyr::select(idp,dtsortida) %>%**

**cbind(data\_index\_data) %>% # Fusiono dates random**

**filter(dtindex.random<=lubridate::ymd(dtsortida)) %>% # Filtro només aquells que dins de la data de seguiment**

**select (idp,dtindex.random) %>%**

**as\_tibble()**

**}**

**#**

**# GENERA UNA DATA INDEX SEGONS UNA DETERMINACIÓ ----------------------**

**## RETORNA DADES AMB idp + dtindex.semirandom**

**dt\_index\_data\_semirandom<-function(dt=PACIENTS,dt.variables=VARIABLES,codi="EK201"){**

**# dt=PACIENTS**

**# dt.variables=VARIABLES**

**# codi="EK201"**

**# b) SEMI.RANDOM (amb un màxim de data a data sortida)**

**# Una data entre tots Colesterol total (prèvies a data sortida)**

**# Si no hi ha cap Colesterol alguna V clínica període (Random)**

**set.seed(123)**

**### Per cada pacient selecciono una dat random de entre tots els COLESTEROLS (2010-2016)**

**UN.COLESTEROL<-dt.variables %>%**

**filter(cod==codi) %>% # selecciono colesterols (Validar que es EK201)**

**dplyr::left\_join(dt,by="idp") %>% # Junto pacients**

**dplyr::select(idp,cod,dat,dtsortida) %>% # Selecciono camps necessaris**

**filter(!is.na(dtsortida)) %>% # Filtro només pacients (amb dtsortida)**

**filter (dat>=20100101 & dat<=dtsortida) %>% # filtro Dates dins periode de seguiment**

**group\_by(idp) %>% # Agafo un colesterol per cada idp**

**sample\_n(size = 1) %>% # Random**

**ungroup %>%**

**dplyr::select(idp, dat) %>%**

**rename(dat\_col=dat)**

**### Per cada pacient selecciono una dat random entre totes les VARIABLES**

**UNA.VARIABLE<-dt.variables %>% # totes les variables**

**dplyr::left\_join(dt,by="idp") %>% # Junto pacients**

**dplyr::select(idp,dat,dtsortida) %>% # Selecciono camps necessaris**

**filter(!is.na(dtsortida)) %>% # Filtro només pacients amb dtsortida**

**filter (dat>=20100101 & dat<=dtsortida) %>% # Dates possibles dins el seguiment**

**group\_by(idp) %>% # Agafo unA fila per cada idp**

**sample\_n(size = 1) %>% # RAndom**

**ungroup() %>%**

**dplyr::select(idp, dat) %>%**

**rename(dat\_var=dat)**

**### Fusió d'ambdos fitxers i selecciono una d'elles preferentment colesterol**

**BDADES\_DT\_INDEX<-UNA.VARIABLE %>%**

**left\_join(UN.COLESTEROL,by="idp") %>%**

**mutate(dtindex.semirandom=ifelse(is.na(dat\_col),dat\_var,dat\_col)) %>%**

**dplyr::select(idp,dtindex.semirandom)**

**}**

**# MATCHING CAS-CONTROL SEGONS MÉTODE DENSITY-INCIDENCE ------------------**

**## Retorna Subset matxejat per grup (event) en data index (dtindex.random, control) DE dt\_pacients\_dindex**

**## Llista de variables variables.ps**

**matching\_case\_control<-function(dt=PACIENTS,variables.ps=llistaPS,dt\_pacients\_dindex=BD\_PAC\_DINDEX) {**

**# dt=PACIENTS**

**# variables.ps=c("edat","dtindex","sexe") # covaribles**

**# dt\_pacients\_dindex=BD\_PAC\_DINDEX**

**# Es neceseciten camps com <dtsortida idp event> + llista de variables a matxejar**

**# <idp, dtindex.random, control> en BD\_PAC\_DINDEX**

**# 2 Fusionar events i controls en una sola taula**

**dt <-dt %>%**

**left\_join(dt\_pacients\_dindex,by="idp") # dt + dtindex.random (data random generada + de control**

**# Selecciono events i mutar dataindex (event=1) en data d'event (dtsortida)**

**dtevents<-dt %>% filter(event==1) %>% mutate(dtindex=lubridate::ymd(dtsortida), event=1) ## Els events data de sortida**

**# Seleccionar controls i mutar dataindex en data index random**

**dtcontrols<-dt %>% filter(control==1) %>% mutate(dtindex=dtindex.random, event=0) ## Els controls data random**

**# Fusionar events + controls**

**dt.total<-dtevents %>% rbind(dtcontrols)**

**# 3 Agregar en data index (Edat)**

**# Agrego en dtindex**

**dt.total<-dt.total %>%**

**mutate (edat=as.numeric((dtindex-lubridate::ymd(dnaix))/365.25)) # Calculo edat en dataindex**

**# 4 Fer matching**

**# preparar dades per matching (idp + Llista matching)**

**dadesmatching<-dt.total %>% dplyr::select(idp,edat,dtindex,event,sexe)**

**# Genero llista de covaraibles**

**formulaPS<-as.formula(paste("event", paste(variables.ps, collapse=" + "), sep=" ~ "))**

**dt.matched<-formulaPS %>%**

**matchit(method="nearest",data=dadesmatching,ratio=4,caliper=0.01,distance = "logit") %>% # FAig el matching 4 a 1**

**weights() %>% # Guardo els pesos**

**data.table() %>%**

**'colnames<-'(c("PS")) %>%**

**bind\_cols(dt.total) %>% # Ho junto al dt.total**

**filter(PS==1) %>%**

**as\_tibble()**

**}**

**# FLOW CHART FINAL (diagramaFlowchart) ---------------------**

**#----------------------------------------------------------------------------------------------#**

**# FLOW-CHART FINAL**

**#----------------------------------------------------------------------------------------------#**

**# Diagramer**

**diagramaFlowchart<-function(**

**grups=1,**

**pob=c(1700),**

**pob\_lab=c("Poblaci? Alt Pened?s"),**

**pob1=c(1000,500),**

**pob2=c(400,200),**

**pob3=c(300,100),**

**pob\_lab1=c("A INICIAL","A FINAL"),**

**pob\_lab2=c("B Inicial","B Final"),**

**pob\_lab3=c("C Inicial","C Final"),**

**exc1=c(50,300,150),**

**exc2=c(100,50,50),**

**exc3=c(100,50,50),**

**exc\_lab1=c('Edat>90 anys','M.Cardio','M.Pulmonar'),**

**exc\_lab2=c('Edat>90 anys','M.Cardio','M.Pulmonar'),**

**exc\_lab3=c('Edat>90 anys','M.Cardio','M.Pulmonar'),**

**colors=c('white','grey'),**

**forma=c('ellipse','box'))**

**{**

**if (grups<1)**

**{print("Error, posa els GRUPS, si us plau! al Flowchart!") }**

**else if (grups==1)**

**{diagramaFlowchart1G(**

**pob1=pob1,**

**pob\_lab1=pob\_lab1,**

**exc1=exc1,**

**exc\_lab1=exc\_lab1,**

**colors=colors,**

**forma=forma) }**

**else if (grups==2)**

**{diagramaFlowchart2G(**

**pob=pob,**

**pob\_lab=pob\_lab,**

**pob1=pob1,**

**pob\_lab1=pob\_lab1,**

**exc1=exc1,**

**exc\_lab1=exc\_lab1,**

**pob2=pob2,**

**pob\_lab2=pob\_lab2,**

**exc2=exc2,**

**exc\_lab2=exc\_lab2,**

**colors=colors,**

**forma=forma ) }**

**else if (grups==3)**

**{diagramaFlowchart3G(**

**pob=pob,**

**pob\_lab=pob\_lab,**

**pob1=pob1,**

**pob\_lab1=pob\_lab1,**

**exc1=exc1,**

**exc\_lab1=exc\_lab1,**

**pob2=pob2,**

**pob\_lab2=pob\_lab2,**

**exc2=exc2,**

**exc\_lab2=exc\_lab2,**

**pob3=pob3,**

**pob\_lab3=pob\_lab3,**

**exc3=exc3,**

**exc\_lab3=exc\_lab3,**

**colors=colors,**

**forma=forma) }**

**else if (grups>3)**

**{**

**print("Error no podem fer m?s de 3 Grups pel Flowchart!")**

**}**

**}**

**# FLOW-CHART 1 GRUP ----------------------------------**

**#----------------------------------------------------------------------------------------------#**

**# FLOW-CHART 1 GRUP**

**#----------------------------------------------------------------------------------------------#**

**diagramaFlowchart1G<-function(**

**pob\_lab1=c("Pob Inicial","Pob Final"),**

**pob1=c(1000,50),**

**exc1=c(10,1),**

**exc\_lab1=c('Edat>90 anys','kkk'),**

**colors=c('white','grey'),**

**forma=c('box','box'))**

**{**

**if (length(exc1)<=10)**

**{**

**m1<-""**

**for (i in 1:(length(exc1) ))**

**{**

**m1<-paste0(m1,'->A',i)**

**i=i+1 }**

**m1**

**#"->A1->A2"#**

**#---------------------------------------------------------------------------------------#**

**m2<-""**

**for (i in 1:(length(exc1)))**

**{**

**m2<-paste0(m2,' A',i,'->','E',i,'[color = black,dir=none]')**

**i=i+1 }**

**m2**

**#A1->E1[color = black,dir=none]A2->E2[color = black,dir=none]**

**#---------------------------------------------------------------------------------------#**

**m3<-""**

**for (i in 1:(length(exc1)))**

**{**

**m3<-paste0(m3,' A',i,';')**

**i=i+1 }**

**m3**

**#A1;A2**

**#---------------------------------------------------------------------------------------#**

**m4<-""**

**for (i in 1:(length(exc1)))**

**{**

**m4<-paste0(m4,' E',i,';')**

**i=i+1 }**

**m4**

**#E1; E2;**

**#---------------------------------------------------------------------------------------#**

**m5<-""**

**for (i in 1:(length(exc1)))**

**{**

**m5<-paste0(m5,' subgraph {rank = same;',' A',i,';','E',i,'}')**

**i=i+1 }**

**m5**

**#subgraph {rank = same; A1;E1} subgraph {rank = same; A2;E2}**

**#---------------------------------------------------------------------------------------#**

**m6<-""**

**for (i in 1:(length(exc1)))**

**{**

**m6<-paste0(m6,'A',i,' [label =', "'@@",i+2,"']",';')**

**i=i+1 }**

**m6**

**#A1 [label ='@@3'];A2 [label ='@@4'];**

**#---------------------------------------------------------------------------------------#**

**m7<-""**

**for (i in 1:(length(exc1)))**

**{**

**m7<-paste0(m7,'E',i,' [label =', "'@@",i+(length(exc1)+2),"']",';')**

**i=i+1 }**

**m7**

**#E1 [label ='@@5'];E2 [label ='@@6'];**

**#---------------------------------------------------------------------------------------#**

**m8<-""**

**for (i in 1:(length(exc1)) )**

**{**

**m8<-paste0(m8," \n ","[",i+2,"]:paste0(' ')")**

**i=i+1 }**

**m8**

**#" \n [3]:paste0(' ') \n [4]:paste0(' ')**

**#---------------------------------------------------------------------------------------#**

**paramet<-c(m1,m2,m3,m4,m5,m6,m7,m8)**

**#---------------------------------------------------------------------------------------#**

**makao1<-paste0("digraph rai {","graph[layout = dot]",**

**"node[shape=",forma[1],",","fontsize=12,fontname=Helvetica,width=0.9,**

**penwidth=0.9,color=black,style=filled,fillcolor=",colors[1],"]",**

**"P1;P2;", "node[shape=point,width =0,penwidth=0,color=black]",**

**paramet[3],**

**"node[shape=",forma[2],",","fontsize=8,fontname='Courier New',width=0.5,penwidth=0.5,style=filled,fillcolor=",colors[2],"]",**

**paramet[4],**

**" \n ","P1 [label = '@@1']","P2 [label = '@@2']",**

**paramet[6],**

**paramet[7],**

**" \n ","edge[width=0.5,penwidth=0.5,arrowhead=vee]",**

**" \n ","P1",**

**paramet[1],**

**"->P2[color = black,dir=none] ",**

**" \n ",**

**paramet[2],**

**" \n ",**

**paramet[5],"}",**

**"\n[1]:paste0('", pob\_lab1[1], " \\n ", "[N = ',", pob1[1], ",']')" ,**

**"\n[2]:paste0('", pob\_lab1[2], " \\n ", "[N = ',", pob1[2], ",']')" ,**

**paramet[8],**

**"\n[5]:paste0('", exc\_lab1[1], " \\n ", "[N = ',", exc1[1], ",']')" ,**

**"\n[6]:paste0('", exc\_lab1[2], " \\n ", "[N = ',", exc1[2], ",']')" ,**

**"\n[7]:paste0('", exc\_lab1[3], " \\n ", "[N = ',", exc1[3], ",']')" ,**

**"\n[8]:paste0('", exc\_lab1[4], " \\n ", "[N = ',", exc1[4], ",']')" ,**

**"\n[9]:paste0('", exc\_lab1[5], " \\n ", "[N = ',", exc1[5], ",']')" ,**

**"\n[10]:paste0('", exc\_lab1[6], " \\n ", "[N = ',", exc1[6], ",']')" ,**

**"\n[11]:paste0('", exc\_lab1[7], " \\n ", "[N = ',", exc1[7], ",']')" ,**

**"\n[12]:paste0('", exc\_lab1[8], " \\n ", "[N = ',", exc1[8], ",']')" ,**

**"\n[13]:paste0('", exc\_lab1[9], " \\n ", "[N = ',", exc1[9], ",']')" ,**

**"\n[14]:paste0('", exc\_lab1[10], " \\n ", "[N = ',", exc1[10], ",']')"**

**)**

**#---------------------------------------------------------------------------------------#**

**DiagrammeR::grViz(makao1)**

**#---------------------------------------------------------------------------------------#**

**}**

**else**

**{print("ERROR!, Les Exlusions han de ser iguals o inferiors a 10 !")**

**}**

**}**

**#---------------------------------------------------------------------------------------#**

**#----------------------------------------------------------------------------------------------#**

**# 2 GRUPS**

**#----------------------------------------------------------------------------------------------#**

**# FLOW-CHART 2 GRUPS ----------------------------------**

**diagramaFlowchart2G<-function(**

**pob\_lab=c("Poblaci? Total"),**

**pob\_lab1=c("Poblaci? Inicial A","Poblaci? Final A"),**

**pob\_lab2=c("Poblaci? Inicial B","Poblaci? Final B"),**

**pob=c(70211123),**

**pob1=c(10088,50),**

**exc1=c(1021,111,9),**

**exc\_lab1=c('Edat>90 anys','Cardio','J'),**

**pob2=c(19002,599),**

**exc2=c(1002,150,90),**

**exc\_lab2=c('Edat>76 anys','Rata','U'),**

**colors=c('white','grey'),**

**forma=c('box','box') )**

**{**

**if (length(exc1)<=10 && length(exc2)<=10)**

**{**

**#-----------------------------------------------------------------------------------#**

**m1a<-""**

**for (i in 1:(length(exc1) ))**

**{**

**m1a<-paste0(m1a,'->A',i)**

**i=i+1 }**

**m1a**

**#->A1->A2->A3**

**#-----------------------------------------------------------------------------------#**

**m1b<-""**

**for (i in 1:(length(exc2) ))**

**{**

**m1b<-paste0(m1b,'->B',i)**

**i=i+1 }**

**m1b**

**#->B1->B2->B3**

**#-----------------------------------------------------------------------------------#**

**m2a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m2a<-paste0(m2a,' A',i,'->','E\_A',i,'[color = black,dir=none]')**

**i=i+1 }**

**m2a**

**#A1->E\_A1[color = black,dir=none] A2->E\_A2[color = black,dir=none] A3->E\_A3[color = black,dir=none]**

**#-----------------------------------------------------------------------------------#**

**m2b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m2b<-paste0(m2b,' B',i,'->','E\_B',i,'[color = black,dir=none]')**

**i=i+1 }**

**m2b**

**#B1->E\_B1[color = black,dir=none] B2->E\_B2[color = black,dir=none] B3->E\_B3[color = black,dir=none]**

**#-----------------------------------------------------------------------------------#**

**m3a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m3a<-paste0(m3a,' A',i,';')**

**i=i+1 }**

**m3a**

**#A1; A2; A3;**

**#-----------------------------------------------------------------------------------#**

**m3b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m3b<-paste0(m3b,' B',i,';')**

**i=i+1 }**

**m3b**

**#" B1; B2; B3;"**

**#-----------------------------------------------------------------------------------#**

**m4a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m4a<-paste0(m4a,' E\_A',i,';')**

**i=i+1 }**

**m4a**

**#E\_A1; E\_A2; E\_A3;**

**#-----------------------------------------------------------------------------------#**

**m4b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m4b<-paste0(m4b,' E\_B',i,';')**

**i=i+1 }**

**m4b**

**#E\_B1; E\_B2; E\_B3;**

**#-----------------------------------------------------------------------------------#**

**m5a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m5a<-paste0(m5a,' subgraph {rank = same;',' A',i,';','E\_A',i,';','}')**

**i=i+1 }**

**m5a**

**# subgraph {rank = same; A1;E\_A1;} subgraph {rank = same; A2;E\_A2;} subgraph {rank = same; A3;E\_A3;}**

**#-----------------------------------------------------------------------------------#**

**m5b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m5b<-paste0(m5b,' subgraph {rank = same;',' B',i,';','E\_B',i,';','}')**

**i=i+1 }**

**m5b**

**#subgraph {rank = same; B1;E\_B1;} subgraph {rank = same; B2;E\_B2;} subgraph {rank = same; B3;E\_B3;}**

**#-----------------------------------------------------------------------------------#**

**m6a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m6a<-paste0(m6a,'A',i,'[label=', "'@@",i+5,"']",';')**

**i=i+1 }**

**m6a**

**#A1 A1[label='@@6'];A2[label='@@7'];A3[label='@@8'];**

**#-----------------------------------------------------------------------------------#**

**m6b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m6b<-paste0(m6b,'B',i,'[label=', "'@@",(i+length(exc2))+5,"']",';')**

**i=i+1 }**

**m6b**

**#B1[label='@@9'];B2[label='@@10'];B3[label='@@11'];**

**#-----------------------------------------------------------------------------------#**

**m7a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m7a<-paste0(m7a,'E\_A',i,' [label =', "'@@",19+i,"']",';')**

**i=i+1 }**

**m7a**

**#E\_A1 [label ='@@20'];E\_A2 [label ='@@21'];E\_A3 [label ='@@22'];**

**#-----------------------------------------------------------------------------------#**

**m7b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m7b<-paste0(m7b,'E\_B',i,' [label =', "'@@",29+i,"']",';')**

**i=i+1 }**

**m7b**

**#E\_B1 [label ='@@30'];E\_B2 [label ='@@31'];E\_B3 [label ='@@32'];**

**#-----------------------------------------------------------------------------------#**

**#-----------------------------------------------------------------------------------#**

**paramet2<-c(m1a,m1b,m2a,m2b,m3a,m3b,m4a,m4b,m5a,m5b,m6a,m6b,m7a,m7b)**

**#-----------------------------------------------------------------------------------#**

**makao2<-paste0("digraph rai {","graph[layout = dot]",**

**"node[shape=",forma[1],",","fontsize=12,fontname=Helvetica,width=0.9,penwidth=0.9,color=black,style=filled,fillcolor=",colors[1],"]"," P\_T;PA\_I;PA\_F;PB\_I;PB\_F ",**

**"node[shape=",forma[2],",","fontsize=8,fontname='Courier New',width=0,penwidth=0,style=filled,fillcolor=",colors[2],"]",**

**paramet2[7],paramet2[8],**

**"node[shape=point,width =0,penwidth=0,color=black,fontname='Courier New']",paramet2[5],paramet2[6],**

**" \n ","P\_T[label='@@1']","PA\_I[label='@@2']","PA\_F[label='@@4']","PB\_I[label='@@3']","PB\_F[label='@@5']",**

**paramet2[13],paramet2[14],**

**paramet2[11],paramet2[12],**

**" \n ","edge[width=0.5,penwidth=0.5,arrowhead=vee]",**

**" \n ","P\_T->PA\_I[color = black,arrowhead=vee]",**

**" \n ","P\_T->PB\_I[color = black,arrowhead=vee]",**

**" \n ","PA\_I",paramet2[1],"->PA\_F[color = black,dir=none] ",**

**" \n ",paramet2[3],**

**" \n ","PB\_I",paramet2[2],"->PB\_F[color = black,dir=none] ",**

**" \n ",paramet2[4],**

**" \n ",paramet2[9],**

**" \n ",paramet2[10],"}",**

**" \n[1]:paste0('", pob\_lab[1], " \\n "," [N = ',", pob[1], ",']')",**

**" \n[2]:paste0('", pob\_lab1[1]," \\n ", " [N = ',", pob1[1], ",']')",**

**" \n[3]:paste0('", pob\_lab2[1]," \\n ", " [N = ',", pob2[1], ",']')",**

**" \n[4]:paste0('", pob\_lab1[2]," \\n ", " [N = ',", pob1[2], ",']')",**

**" \n[5]:paste0('", pob\_lab2[2]," \\n ", " [N = ',", pob2[2], ",']')",**

**" \n[6]:paste0('')"," \n[7]:paste0('')"," \n[8]:paste0('')"," \n[9]:paste0('')",**

**" \n[10]:paste0('')"," \n[11]:paste0('')"," \n[12]:paste0('')"," \n[13]:paste0('')",**

**" \n[14]:paste0('')"," \n[15]:paste0('')"," \n[16]:paste0('')"," \n[17]:paste0('')",**

**" \n[18]:paste0('')"," \n[19]:paste0('')",**

**" \n[20]:paste0('", exc\_lab1[1]," \\n ", "[N = ',", exc1[1], ",']')",**

**" \n[21]:paste0('", exc\_lab1[2]," \\n ", "[N = ',", exc1[2], ",']')",**

**" \n[22]:paste0('", exc\_lab1[3]," \\n ", "[N = ',", exc1[3], ",']')",**

**" \n[23]:paste0('", exc\_lab1[4]," \\n ", "[N = ',", exc1[4], ",']')",**

**" \n[24]:paste0('", exc\_lab1[5]," \\n ", "[N = ',", exc1[5], ",']')",**

**" \n[25]:paste0('", exc\_lab1[6]," \\n ", "[N = ',", exc1[6], ",']')",**

**" \n[26]:paste0('", exc\_lab1[7]," \\n ", "[N = ',", exc1[7], ",']')",**

**" \n[27]:paste0('", exc\_lab1[8]," \\n ", "[N = ',", exc1[8], ",']')",**

**" \n[28]:paste0('", exc\_lab1[9]," \\n ", "[N = ',", exc1[9], ",']')",**

**" \n[29]:paste0('", exc\_lab1[10]," \\n ", "[N = ',",exc1[10], ",']')",**

**" \n[30]:paste0('", exc\_lab2[1]," \\n ", "[N = ',", exc2[1], ",']')",**

**" \n[31]:paste0('", exc\_lab2[2]," \\n ", "[N = ',", exc2[2], ",']')",**

**" \n[32]:paste0('", exc\_lab2[3]," \\n ", "[N = ',", exc2[3], ",']')",**

**" \n[33]:paste0('", exc\_lab2[4]," \\n ", "[N = ',", exc2[4], ",']')",**

**" \n[34]:paste0('", exc\_lab2[5]," \\n ", "[N = ',", exc2[5], ",']')",**

**" \n[35]:paste0('", exc\_lab2[6]," \\n ", "[N = ',", exc2[6], ",']')",**

**" \n[36]:paste0('", exc\_lab2[7]," \\n ", "[N = ',", exc2[7], ",']')",**

**" \n[37]:paste0('", exc\_lab2[8]," \\n ", "[N = ',", exc2[8], ",']')",**

**" \n[38]:paste0('", exc\_lab2[9]," \\n ", "[N = ',", exc2[9], ",']')",**

**" \n[39]:paste0('", exc\_lab2[10]," \\n ", "[N = ',",exc2[10], ",']')"**

**)**

**#---------------------------------------------------------------------------------------#**

**DiagrammeR::grViz(makao2)**

**#---------------------------------------------------------------------------------------#**

**}**

**else**

**{print("ERROR!, Les Exlusions han de ser iguals o inferiors a 10 !")**

**}**

**}**

**#---------------------------------------------------------------------------------------#**

**#----------------------------------------------------------------------------------------------#**

**# 3 GRUPS**

**#----------------------------------------------------------------------------------------------#**

**# FLOW-CHART 3 GRUPS ----------------------------------**

**diagramaFlowchart3G<-function(**

**pob\_lab=c("Poblaci? Total"),**

**pob\_lab1=c("Poblaci? Inicial A","Poblaci? Final A"),**

**pob\_lab2=c("Poblaci? Inicial B","Poblaci? Final B"),**

**pob\_lab3=c("Poblaci? Inicial C","Poblaci? Final C"),**

**pob=c(70211123),**

**pob1=c(10088,50),**

**exc1=c(1021,111,9),**

**exc\_lab1=c('Edat>90 anys','Cardio','J'),**

**pob2=c(19002,599),**

**exc2=c(1002,150,90),**

**exc\_lab2=c('Edat>76 anys','Rata','U'),**

**pob3=c(19002,599),**

**exc3=c(1002,150,0),**

**exc\_lab3=c('Edat>91 anys','Pulm?','L'),**

**colors=c('white','grey'),**

**forma=c('box','box') )**

**{**

**if (length(exc1)<=10 && length(exc2)<=10 && length(exc3)<=10 )**

**{**

**#-----------------------------------------------------------------------------------#**

**m1a<-""**

**for (i in 1:(length(exc1) ))**

**{**

**m1a<-paste0(m1a,'->A',i)**

**i=i+1 }**

**m1a**

**#"->A1->A2->A3"#**

**#-----------------------------------------------------------------------------------#**

**m1b<-""**

**for (i in 1:(length(exc2) ))**

**{**

**m1b<-paste0(m1b,'->B',i)**

**i=i+1 }**

**m1b**

**#"->B1->B2->B3"**

**#-----------------------------------------------------------------------------------#**

**m1c<-""**

**for (i in 1:(length(exc3) ))**

**{**

**m1c<-paste0(m1c,'->C',i)**

**i=i+1 }**

**m1c**

**#"->C1->C2->C3"**

**#-----------------------------------------------------------------------------------#**

**m2a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m2a<-paste0(m2a,' A',i,'->','E\_A',i,'[color = black,dir=none]')**

**i=i+1 }**

**m2a**

**#A1->E\_A1[color = black,dir=none] A2->E\_A2[color = black,dir=none] A3->E\_A3[color = black,dir=none]**

**#-----------------------------------------------------------------------------------#**

**m2b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m2b<-paste0(m2b,' B',i,'->','E\_B',i,'[color = black,dir=none]')**

**i=i+1 }**

**m2b**

**#B1->E\_B1[color = black,dir=none] B2->E\_B2[color = black,dir=none]**

**#-----------------------------------------------------------------------------------#**

**m2c<-""**

**for (i in 1:(length(exc3)))**

**{**

**m2c<-paste0(m2c,' C',i,'->','E\_C',i,'[color = black,dir=none]')**

**i=i+1 }**

**m2c**

**#C1->E\_C1[color = black,dir=none] C2->E\_C2[color = black,dir=none] C3->E\_C3[color = black,dir=none]**

**#-----------------------------------------------------------------------------------#**

**m3a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m3a<-paste0(m3a,' A',i,';')**

**i=i+1 }**

**m3a**

**# A1; A2; A3;**

**#-----------------------------------------------------------------------------------#**

**m3b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m3b<-paste0(m3b,' B',i,';')**

**i=i+1 }**

**m3b**

**#B1; B2; B3;**

**#-----------------------------------------------------------------------------------#**

**m3c<-""**

**for (i in 1:(length(exc3)))**

**{**

**m3c<-paste0(m3c,' C',i,';')**

**i=i+1 }**

**m3c**

**#C1; C2; C3;**

**#-----------------------------------------------------------------------------------#**

**m4a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m4a<-paste0(m4a,' E\_A',i,';')**

**i=i+1 }**

**m4a**

**#E\_A1; E\_A2; E\_A3;**

**#-----------------------------------------------------------------------------------#**

**m4b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m4b<-paste0(m4b,' E\_B',i,';')**

**i=i+1 }**

**m4b**

**#E\_B1; E\_B2; E\_B3;"**

**#-----------------------------------------------------------------------------------#**

**m4c<-""**

**for (i in 1:(length(exc3)))**

**{**

**m4c<-paste0(m4c,' E\_C',i,';')**

**i=i+1 }**

**m4c**

**#E\_C1; E\_C2; E\_C3;**

**#-----------------------------------------------------------------------------------#**

**m5a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m5a<-paste0(m5a,' subgraph {rank = same;',' A',i,';','E\_A',i,';','}')**

**i=i+1 }**

**m5a**

**#subgraph {rank = same; A1;E\_A1;} subgraph {rank = same; A2;E\_A2;} subgraph {rank = same; A3;E\_A3;}"**

**#-----------------------------------------------------------------------------------#**

**m5b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m5b<-paste0(m5b,' subgraph {rank = same;',' B',i,';','E\_B',i,';','}')**

**i=i+1 }**

**m5b**

**#subgraph {rank = same; B1;E\_B1;} subgraph {rank = same; B2;E\_B2;} subgraph {rank = same; B3;E\_B3;}**

**#-----------------------------------------------------------------------------------#**

**m5c<-""**

**for (i in 1:(length(exc3)))**

**{**

**m5c<-paste0(m5c,' subgraph {rank = same;',' C',i,';','E\_C',i,';','}')**

**i=i+1 }**

**m5c**

**# subgraph {rank = same; C1;E\_C1;} subgraph {rank = same; C2;E\_C2;} subgraph {rank = same; C3;E\_C3;}**

**#-----------------------------------------------------------------------------------#**

**m6a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m6a<-paste0(m6a,'A',i,'[label=', "'@@",i+7,"']",';')**

**i=i+1 }**

**m6a**

**#A1[label='@@8'];A2[label='@@9'];A3[label='@@10'];**

**#-----------------------------------------------------------------------------------#**

**m6b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m6b<-paste0(m6b,'B',i,'[label=', "'@@",(i+length(exc2))+7,"']",';')**

**i=i+1 }**

**m6b**

**#B1[label='@@11'];B2[label='@@12'];B3[label='@@13'];**

**#-----------------------------------------------------------------------------------#**

**m6c<-""**

**for (i in 1:(length(exc3)))**

**{**

**m6c<-paste0(m6c,'C',i,'[label=', "'@@",(i+length(exc3))+10,"']",';')**

**i=i+1 }**

**m6c**

**#C1[label='@@14'];C2[label='@@15'];C3[label='@@16'];"**

**#-----------------------------------------------------------------------------------#**

**m7a<-""**

**for (i in 1:(length(exc1)))**

**{**

**m7a<-paste0(m7a,'E\_A',i,' [label =', "'@@",19+i,"']",';')**

**i=i+1 }**

**m7a**

**#"E\_A1 [label ='@@20'];E\_A2 [label ='@@21'];E\_A3 [label ='@@22']"**

**#-----------------------------------------------------------------------------------#**

**m7b<-""**

**for (i in 1:(length(exc2)))**

**{**

**m7b<-paste0(m7b,'E\_B',i,' [label =', "'@@",29+i,"']",';')**

**i=i+1 }**

**m7b**

**#E\_B1 [label ='@@30'];E\_B2 [label ='@@31'];E\_B3 [label ='@@32']**

**#-----------------------------------------------------------------------------------#**

**m7c<-""**

**for (i in 1:(length(exc3)))**

**{**

**m7c<-paste0(m7c,'E\_C',i,' [label =', "'@@",39+i,"']",';')**

**i=i+1 }**

**m7c**

**#"E\_C1 [label ='@@40'];E\_C2 [label ='@@41'];E\_C3 [label ='@@42'];"**

**#-----------------------------------------------------------------------------------#**

**paramet2<-c(m1a,m1b,m1c,m2a,m2b,m2c,m3a,m3b,m3c,m4a,m4b,m4c,m5a,m5b,m5c,m6a,m6b,m6c,m7a,m7b,m7c)**

**#-----------------------------------------------------------------------------------#**

**makao3<-paste0("digraph rai {","graph[layout = dot]",**

**"node[shape=",forma[1],",","fontsize=12,fontname=Helvetica,width=0.9,penwidth=0.9,color=black,style=filled,fillcolor=",colors[1],"]"," P\_T;PA\_I;PA\_F;PB\_I;PB\_F;PC\_I;PC\_F ",**

**"node[shape=",forma[2],",","fontsize=8,fontname='Courier New',width=0,penwidth=0,style=filled,fillcolor=",colors[2],"]",**

**paramet2[10],paramet2[11],paramet2[12],**

**"node[shape=point,width =0,penwidth=0,color=black,fontname='Courier New']",paramet2[7],paramet2[8],paramet2[9],**

**" \n ","P\_T[label='@@1']","PA\_I[label='@@2']","PA\_F[label='@@5']","PB\_I[label='@@3']","PB\_F[label='@@6']","PC\_I[label='@@4']","PC\_F[label='@@7']",**

**" \n ",paramet2[19],paramet2[20],paramet2[21],**

**" \n ",paramet2[16],paramet2[17],paramet2[18],**

**" \n ","edge[width=0.5,penwidth=0.5,arrowhead=vee]",**

**" \n ","P\_T->PA\_I[color = black,arrowhead=vee]",**

**" \n ","P\_T->PB\_I[color = black,arrowhead=vee]",**

**" \n ","P\_T->PC\_I[color = black,arrowhead=vee]",**

**" \n ","PA\_I",paramet2[1],"->PA\_F[color = black,dir=none] ",**

**" \n ",paramet2[4],**

**" \n ","PB\_I",paramet2[2],"->PB\_F[color = black,dir=none] ",**

**" \n ",paramet2[5],**

**" \n ","PC\_I",paramet2[3],"->PC\_F[color = black,dir=none] ",**

**" \n ",paramet2[6],**

**" \n ",paramet2[13],**

**" \n ",paramet2[14],**

**" \n ",paramet2[15],"}",**

**" \n[1]:paste0('", pob\_lab[1], " \\n ", " [N = ',", pob[1], ",']')",**

**" \n[2]:paste0('", pob\_lab1[1]," \\n ", " [N = ',", pob1[1], ",']')",**

**" \n[3]:paste0('", pob\_lab2[1]," \\n ", " [N = ',", pob2[1], ",']')",**

**" \n[4]:paste0('", pob\_lab3[1]," \\n ", " [N = ',", pob3[1], ",']')",**

**" \n[5]:paste0('", pob\_lab1[2]," \\n ", " [N = ',", pob1[2], ",']')",**

**" \n[6]:paste0('", pob\_lab2[2]," \\n ", " [N = ',", pob2[2], ",']')",**

**" \n[7]:paste0('", pob\_lab3[2]," \\n ", " [N = ',", pob3[2], ",']')",**

**" \n[8]:paste0('')"," \n[9]:paste0('')"," \n[10]:paste0('')"," \n[11]:paste0('')",**

**" \n[12]:paste0('')"," \n[13]:paste0('')"," \n[14]:paste0('')"," \n[15]:paste0('')",**

**" \n[16]:paste0('')"," \n[17]:paste0('')"," \n[18]:paste0('')"," \n[19]:paste0('')",**

**" \n[20]:paste0('", exc\_lab1[1]," \\n ", "[N = ',", exc1[1], ",']')",**

**" \n[21]:paste0('", exc\_lab1[2]," \\n ", "[N = ',", exc1[2], ",']')",**

**" \n[22]:paste0('", exc\_lab1[3]," \\n ", "[N = ',", exc1[3], ",']')",**

**" \n[23]:paste0('", exc\_lab1[4]," \\n ", "[N = ',", exc1[4], ",']')",**

**" \n[24]:paste0('", exc\_lab1[5]," \\n ", "[N = ',", exc1[5], ",']')",**

**" \n[25]:paste0('", exc\_lab1[6]," \\n ", "[N = ',", exc1[6], ",']')",**

**" \n[26]:paste0('", exc\_lab1[7]," \\n ", "[N = ',", exc1[7], ",']')",**

**" \n[27]:paste0('", exc\_lab1[8]," \\n ", "[N = ',", exc1[8], ",']')",**

**" \n[28]:paste0('", exc\_lab1[9]," \\n ", "[N = ',", exc1[9], ",']')",**

**" \n[20]:paste0('", exc\_lab1[10]," \\n ", "[N = ',",exc1[10], ",']')",**

**" \n[30]:paste0('", exc\_lab2[1]," \\n ", "[N = ',", exc2[1], ",']')",**

**" \n[31]:paste0('", exc\_lab2[2]," \\n ", "[N = ',", exc2[2], ",']')",**

**" \n[32]:paste0('", exc\_lab2[3]," \\n ", "[N = ',", exc2[3], ",']')",**

**" \n[33]:paste0('", exc\_lab2[4]," \\n ", "[N = ',", exc2[4], ",']')",**

**" \n[34]:paste0('", exc\_lab2[5]," \\n ", "[N = ',", exc2[5], ",']')",**

**" \n[35]:paste0('", exc\_lab2[6]," \\n ", "[N = ',", exc2[6], ",']')",**

**" \n[36]:paste0('", exc\_lab2[7]," \\n ", "[N = ',", exc2[7], ",']')",**

**" \n[37]:paste0('", exc\_lab2[8]," \\n ", "[N = ',", exc2[8], ",']')",**

**" \n[38]:paste0('", exc\_lab2[9]," \\n ", "[N = ',", exc2[9], ",']')",**

**" \n[39]:paste0('", exc\_lab2[10]," \\n ","[N = ',", exc2[10], ",']')",**

**" \n[40]:paste0('", exc\_lab3[1]," \\n ", "[N = ',", exc3[1], ",']')",**

**" \n[41]:paste0('", exc\_lab3[2]," \\n ", "[N = ',", exc3[2], ",']')",**

**" \n[42]:paste0('", exc\_lab3[3]," \\n ", "[N = ',", exc3[3], ",']')",**

**" \n[43]:paste0('", exc\_lab3[4]," \\n ", "[N = ',", exc3[4], ",']')",**

**" \n[44]:paste0('", exc\_lab3[5]," \\n ", "[N = ',", exc3[5], ",']')",**

**" \n[45]:paste0('", exc\_lab3[6]," \\n ", "[N = ',", exc3[6], ",']')",**

**" \n[46]:paste0('", exc\_lab3[7]," \\n ", "[N = ',", exc3[7], ",']')",**

**" \n[47]:paste0('", exc\_lab3[8]," \\n ", "[N = ',", exc3[8], ",']')",**

**" \n[48]:paste0('", exc\_lab3[9]," \\n ", "[N = ',", exc3[9], ",']')",**

**" \n[49]:paste0('", exc\_lab3[10]," \\n ", "[N = ',",exc3[10], ",']')"**

**)**

**#---------------------------------------------------------------------------------------#**

**DiagrammeR::grViz(makao3)**

**#---------------------------------------------------------------------------------------#**

**}**

**else**

**{print("ERROR!, Les Exlusions han de ser iguals o inferiors a 10 !")**

**}**

**}**

**# NETEJA NOMS DE VARIABLES DE CARACTERS EXTRANYS ("/","(".....) ---------------**

**netejar.noms.variables<-function(dt=LIPOS\_EORTEGA){**

**paco<-names(dt) %>%**

**iconv("UTF-8","ASCII","") %>%**

**stringr::str\_replace("/","") %>%**

**stringr::str\_replace\_all("\\(","") %>%**

**stringr::str\_replace\_all("\\)","") %>%**

**stringr::str\_replace\_all("\\/","") %>%**

**stringr::str\_trim() %>%**

**stringr::str\_replace\_all(" ","\_") %>%**

**stringr::str\_replace\_all("-","\_")**

**names(dt)<-paco**

**dt**

**}**

**# Funció que elimina accents dels noms de les variables**

**netejar.accents.variables <- function(dt=LIPOS\_EORTEGA){**

**paco<-names(dt) %>%**

**iconv(to="ASCII//TRANSLIT")**

**names(dt)<-paco**

**dt**

**}**

**# Comptar\_valors(dt, vector\_variables, valor) ##################**

**# en funció de vector de variables, i un valor("Yes")**

**comptar\_valors<-function(dt=dadesevents,variables=c("EV.TER.ARTER\_PERIF","EV.TER.AVC"),valor="Yes"){**

**# dt=dades**

**# variables=c("EV1\_ULCERES", "EV2\_ULCERES", "EV3\_ULCERES", "EV4\_ULCERES")**

**# valor="Yes"**

**# Concateno valors**

**pepito<-paste0("paste0(",paste0(variables,collapse = ","),")")**

**dt<-dt %>%**

**mutate\_("combi\_vars"=pepito) %>%**

**mutate(**

**num\_valors=str\_count(combi\_vars,valor)) %>%**

**dplyr::select(-combi\_vars)**

**}**

**# mostreig\_ids () Mostreja ids d'una base de dades ---------------------**

**mostreig\_ids<-function(dt,id="idp",n\_mostra=100) {**

**# n\_mostra<-100**

**# dt<-dades**

**# id="idp"**

**if (n\_mostra!=Inf) {**

**id\_sym<-sym(id)**

**id\_sample<-dt %>% distinct(!!id\_sym) %>%sample\_n(size=n\_mostra)**

**dt<-id\_sample %>% left\_join(dt,by=id)**

**} else { dt<-dt}**

**dt**

**}**

**#**

**# Funció per calcular el risc REGICOR (regicor) -----------------**

**#**

**# age: númerica (anys)**

**# sex: text, 'H' homes i 'D' dones**

**# smoker, diabetes: binària (0 no i 1 si)**

**# coltot i colhdl: en mg/dL**

**# sbp i dbp: númeric (mmHg)**

**regicor <- function(age, sex, smoker, diabetes, coltot, colhdl, sbp, dbp, divide = 1){**

**n <- length(age)**

**diabetes <- as.numeric(diabetes)**

**bp\_opti <- ifelse(sbp < 120 & dbp < 80, 1, 0)**

**bp\_high <- ifelse((130 <= sbp & sbp < 140) | (85 <= dbp & dbp < 90), 1, 0)**

**bp\_i <- ifelse((140 <= sbp & sbp < 160) | (90 <= dbp & dbp < 100), 1, 0)**

**bp\_ii <- ifelse(160 <= sbp | 100 <= dbp, 1, 0)**

**i\_bp\_ii <- (bp\_ii == 1)**

**bp\_opti[i\_bp\_ii] <- bp\_high[i\_bp\_ii] <- bp\_i[i\_bp\_ii] <- 0**

**i\_bp\_i <- (bp\_i == 1)**

**bp\_opti[i\_bp\_i] <- bp\_high[i\_bp\_i] <- 0**

**i\_bp\_high <- (bp\_high == 1)**

**bp\_opti[i\_bp\_high] <- 0**

**c\_160 <- ifelse(coltot < 160, 1, 0)**

**c200\_239 <- ifelse(200 <= coltot & coltot < 240, 1, 0)**

**c240\_279 <- ifelse(240 <= coltot & coltot < 280, 1, 0)**

**c280\_ <- ifelse(280 <= coltot, 1, 0)**

**h\_35 <- ifelse(colhdl < 35, 1, 0)**

**h35\_44 <- ifelse(35 <= colhdl & colhdl < 45, 1, 0)**

**h45\_49 <- ifelse(45 <= colhdl & colhdl < 50, 1, 0)**

**h50\_59 <- ifelse(50 <= colhdl & colhdl < 60, 1, 0)**

**h60\_ <- ifelse(60 <= colhdl, 1, 0)**

**men <- (sex == 'H')**

**l\_chol = rep(0, n)**

**l\_chol[men] <- (0.04826\*age - 0.65945\*c\_160 + 0.17692\*c200\_239 + 0.50539\*c240\_279 +**

**0.65713\*c280\_ + 0.49744\*h\_35 + 0.24310\*h35\_44 - 0.05107\*h50\_59 - 0.48660\*h60\_ -**

**0.00226\*bp\_opti + 0.28320\*bp\_high + 0.52168\*bp\_i + 0.61859\*bp\_ii +**

**0.42839\*diabetes + 0.52337\*smoker)[men]**

**l\_chol[!men] <- (0.33766\*age - 0.00268\*(age^2) - 0.26138\*c\_160 + 0.20771\*c200\_239 +**

**0.24385\*c240\_279 + 0.53513\*c280\_ + 0.84312\*h\_35 + 0.377096\*h35\_44 +**

**0.19785\*h45\_49 - 0.42951\*h60\_ - 0.53363\*bp\_opti - 0.06773\*bp\_high +**

**0.26288\*bp\_i + 0.46573\*bp\_ii + 0.59626\*diabetes + 0.29246\*smoker)[!men]**

**g\_chol = rep(0, n)**

**g\_chol[men] <- 3.489**

**g\_chol[!men] = 10.279**

**b\_chol <- exp(l\_chol - g\_chol)**

**result <- rep(0,n)**

**result[men] <- (1 - (1 -(1 - 0.951)/divide)^b\_chol[men])\*100**

**result[!men] <- (1 - (1 - (1 - 0.978)/divide)^b\_chol[!men])\*100**

**result**

**}**

**#30.Setembre.2019#**

**# FI GENERAR FUNCIONs**