Manejo avanzado de datos

Caso estudio

ISCIII

18 de noviembre de 2021

1 Importación/Exportación

1.1 Importa los datos del caso estudio (datos.caso.estudio.txt)

```
rm(list=ls())
datos<-read.table("ruta/a/mi/directorio/datos.caso.estudio.txt",header=TRUE,sep="\t")
```

1.2 Crea una base de datos que contenga los hombres casados fumadores de la base de datos y exportalo con el nombre "HCF.txt"

```
unique(datos$"sexo")
```

```
## [1] "Mujer"  "Hombre"
```

```
unique(datos$"estado.civil")
```

```
## [1] "Casado"     "Soltero"    "Divorciado"
```

```
unique(datos$"fumador")
```

```
## [1] "No" "Si"
```

```
datos_hcf=subset(datos, sexo=="Hombre" & estado.civil=="Casado" & fumador=="Si")

dim(datos_hcf)
```

```
## [1] 20 15
```

```
write.table(datos_hcf,file="data/HCF.txt",sep="\t",quote=FALSE,row.names=FALSE)
```

2 Codificación de variable

2.1 Crea una nueva variable de estado civil donde los casados y divorciados pertenecen a la misma categoria

```
levels.old=c("Casado","Soltero","Divorciado")
levels.new=c("Casado/divorciado","Soltero","Casado/divorciado")
datos$estado.civil.new <- factor(datos$estado.civil, levels=levels.old,labels=levels.new)

table(datos$estado.civil.new,datos$estado.civil)
```

```
##
##                     Casado Divorciado Soltero
##   Casado/divorciado     50         75       0
##   Soltero                0          0      75
```

2.2 Crea una variable de grupos de edad de acuerdo a los intervalos : [0,25) , [25,60), [60,85]

```
datos$grupo.edad <-cut(datos$edad,breaks=c(0,25,60,85),right=FALSE,include.lowest=TRUE)

table(datos$grupo.edad)
```

```
##
##  [0,25) [25,60) [60,85]
##      56      88      56
```

2.3 Crea una variable de fecha a partir de la variable `fdiag_cm`

```
class(datos$fdiag_cm)
```

```
## [1] "character"
```

```
datos$fecha.CM <- as.Date(datos$fdiag_cm)
class(datos$fecha.CM)
```

```
## [1] "Date"
```

```
datos$fecha.CM[1:6]
```

```
## [1] "1977-05-05" NA           "2007-09-28" NA           NA
## [6] NA
```

2.4 Crea una variable de fecha a partir de la variable `fdiag_cp`

```
class(datos$fdiag_cp)
```

```
## [1] "character"
```

```
datos$fecha.CP <- as.Date(datos$fdiag_cp, format="%d.%m.%y")
class(datos$fecha.CP)
```

```
## [1] "Date"
```

```
datos$fecha.CP[1:6]
```

```
## [1] NA           NA           "1996-06-17" "2006-08-12" NA
## [6] "2002-06-28"
```

2.5 Crea una variable de fecha a partir de la variable `fdef`

```
class(datos$fdef)
```

```
## [1] "character"
```

```
datos$fecha.DF <- as.Date(datos$fdef,format="%Y.%m.%d")
class(datos$fecha.DF)
```

```
## [1] "Date"
```

```
datos$fecha.DF[1:6]
```

```
## [1] "1977-05-14" NA           "2007-12-13" "2006-10-06" NA
## [6] "2002-08-21"
```

2.6 Crea una variable tipo caracter a partir de la variable `fdef` de tal manera que figure solo el mes y el año (ejemplo: mayo/1980)

```
datos$fecha.DF.mes <-as.character( format(datos$fecha.DF,format="%B/%Y") )
datos$fecha.DF.mes[1:6]
```

```
## [1] "mayo/1977"      NA               "diciembre/2007" "octubre/2006"
## [5] NA               "agosto/2002"
```

2.7 Crea una variable que cuenta el número de días entre la fecha de diagnostico de cáncer de mama y la fecha de defunción

```
datos$supervivencia.cm <-difftime(datos$fecha.DF,datos$fecha.CM,units="days")
class(datos$supervivencia)
```

```
## [1] "difftime"
```

```
head(datos[,c("ID","fecha.DF","fecha.CM","supervivencia.cm")])
```

```
##           ID   fecha.DF   fecha.CM supervivencia.cm
## 1 137--EXT_1 1977-05-14 1977-05-05           9 days
## 2 174--MAD_1       <NA>       <NA>          NA days
## 3 200--AND_2 2007-12-13 2007-09-28          76 days
## 4   23--GA_2 2006-10-06       <NA>          NA days
## 5   39--GA_1       <NA>       <NA>          NA days
## 6  90--EXT_2 2002-08-21       <NA>          NA days
```

2.8 Crea una variable que cuenta el número de semanas entre la fecha de diagnostico de cáncer de prostata y la fecha de defunción

```
datos$supervivencia.cp <-difftime(datos$fecha.DF,datos$fecha.CP,units="week")
class(datos$supervivencia.cp)
```

```
## [1] "difftime"
```

```
head(datos[,c("ID","fecha.DF","fecha.CP","supervivencia.cp")])
```

```
##           ID   fecha.DF   fecha.CP supervivencia.cp
## 1 137--EXT_1 1977-05-14       <NA>          NA weeks
## 2 174--MAD_1       <NA>       <NA>          NA weeks
## 3 200--AND_2 2007-12-13 1996-06-17 599.428571 weeks
## 4   23--GA_2 2006-10-06 2006-08-12   7.857143 weeks
## 5   39--GA_1       <NA>       <NA>          NA weeks
## 6  90--EXT_2 2002-08-21 2002-06-28   7.714286 weeks
```

3 Manipulación de caracteres

3.1 Comprueba el número de caracteres de la variable ID de los datos

```
nchar(datos$ID)
```

```
##   [1] 10 10 10  8  8  9  8 10  9  9  9  9  9  9 10 10 10 10  8 10  9  8  9 10 10
##  [26] 10 10 10 10 10 10  9 10  8 10  9 10 10  9  9 10  9 10  9 10  8  9  9  8  9
##  [51] 10  9  8  8  9 10 10  9  7  9 10 10  9  9  9  9  9  9  9 10  9 10  8  9  9
##  [76]  9 10  8 10  9  9 10  9 10 10  9  9 10 10  9 10 10 10  9  9 10 10 10  9 10
## [101] 10  9 10 10 10  9  9 10 10 10  9 10 10  9 10 10  9  8  9  8  8  9  9  9  9
## [126]  9  9 10  9 10 10  9  8  9  8  9  9  8  9 10  9  9  9  8  9 10 10 10  9  9
## [151]  9 10 10  7 10 10  9  8  9 10 10  8  9  8 10  9  8  8  9  9  9  9  9 10  9
## [176]  9  8  8 10 10  9 10  9 10 10 10  8 10  9 10  9  9  9  9 10  9  9  9 10 10
```

```r
unique(nchar(datos$ID))
```

```
## [1] 10  8  9  7
```

```r
table(nchar(datos$ID),exclude=NULL)
```

```
##
##  7  8  9 10
##  2 27 89 82
```

3.2 Comprueba si existen valores repetidos en la variable ID de los datos

```r
nrow(datos)
```

```
## [1] 200
```

```r
length(unique(datos$"ID"))
```

```
## [1] 200
```

3.3 Crea una nueva variable que corresponda a los valores que tiene la variable ID antes del primer "--"

```r
temp=strsplit(datos$ID,split="--")
datos$ID.num=sapply(temp,function(x) x[[1]])


head(datos[,c("ID","ID.num")])
```

```
##           ID ID.num
## 1 137--EXT_1    137
## 2 174--MAD_1    174
## 3 200--AND_2    200
## 4   23--GA_2     23
## 5   39--GA_1     39
## 6  90--EXT_2     90
```

3.4 Crea una nueva variable llamada CCAA que corresponda a los caracteres entre "--" y "_" de los valores de la variable ID

```r
temp=strsplit(datos$ID,split="--|_")
datos$CCAA=sapply(temp,function(x) x[[2]])


head(datos[,c("ID","CCAA")])
```

```
##           ID CCAA
## 1 137--EXT_1  EXT
## 2 174--MAD_1  MAD
## 3 200--AND_2  AND
## 4   23--GA_2   GA
## 5   39--GA_1   GA
## 6  90--EXT_2  EXT
```

3.5 Crea una nueva variable que sea igual a la variable ID, pero donde se hayan eliminado los caracteres "--" y "_"

```
datos$ID_new <-gsub("--|_","",datos$ID)
```

4 Asignación condicionada

4.1 Crea una nueva variable que tome el valor "Si" cuando en la variable `complicaciones` se mencione algo referido a Cáncer y "No" en caso contrario

```
unique(datos$complicaciones)
```

```
##  [1] "Cancer pancreas;Cancer pancreas" "EPOC;Cancer gástrico"
##  [3] "Infarto;Infarto"                 "Cancer gástrico;Fiebre"
##  [5] "Fiebre;Cancer pancreas"          "Cancer pancreas;Cancer gástrico"
##  [7] "Infarto;Cancer pancreas"         "Infarto;Diarrea"
##  [9] "Diarrea;Diarrea"                 "Fiebre;Cancer gástrico"
## [11] "Cancer pancreas;Infarto"         "Fiebre;Diarrea"
## [13] "Diarrea;Cancer gástrico"         "Diarrea;EPOC"
## [15] "EPOC;Infarto"                    "Cancer gástrico;Infarto"
## [17] "Infarto;EPOC"                    "Diarrea;Fiebre"
## [19] "Diarrea;Infarto"                 "EPOC;Diarrea"
## [21] "Cancer gástrico;Cancer gástrico" "Infarto;Cancer gástrico"
## [23] "Cancer pancreas;Fiebre"          "Fiebre;Fiebre"
## [25] "Cancer pancreas;Diarrea"         "EPOC;Cancer pancreas"
## [27] "Fiebre;Infarto"                  "EPOC;Fiebre"
## [29] "Infarto;Fiebre"                  "EPOC;EPOC"
## [31] "Cancer gástrico;EPOC"            "Diarrea;Cancer pancreas"
## [33] "Cancer pancreas;EPOC"            "Cancer gástrico;Cancer pancreas"
## [35] "Fiebre;EPOC"                     "Cancer gástrico;Diarrea"
```

```
test <- grepl("Cancer",datos$complicaciones)
datos$cancer.secundario = ifelse(test,"Si","No")
```

4.2 Crea una variable de *sobrepeso* que tome el valor "Sí" cuando el índice de masa corporal (peso/altura^2) sea superior a 25 kg/m^2

```r
datos$imc <- datos$peso/(datos$altura/100)^2
datos$sobrepeso <- ifelse(datos$imc>25,"Si","No")

table(datos$"sobrepeso",exclude=NULL)
```

```
##
##  Si
## 200
```

5 Funciones

5.1 Crea una funcion que revise la base de datos

Si se encuentra algún registro de hombre, casado con nivel de estudios bajo o medio, muestre el siguiente mensaje "REVISA LA BASE DE DATOS" y proporcione los identificadores de los registros que cumplan estas condiciones

```r
depuracion<-function(DF){
  ids_mirar<-NA
    mirar <- subset(DF, sexo=="Hombre" & nivel.estudios!="Alto" & estado.civil == "Casado")
    if(nrow(mirar)>0){
        print("REVISA LA BASE DE DATOS")
        ids_mirar<-mirar$ID
      }
    ids_mirar
}

depuracion(datos)
```

```
## [1] "REVISA LA BASE DE DATOS"
```

```
##  [1] "200--AND_2" "20--AND_1"  "41--MAD_2"  "50--MAD_2"  "29--AND_1"
##  [6] "88--EXT_1"  "83--MAD_1"  "96--EXT_2"  "66--MAD_2"  "38--EXT_2"
## [11] "80--AND_1"  "82--EXT_1"  "70--AND_1"  "84--EXT_2"  "77--GA_1"
## [16] "167--EXT_2" "136--AND_1" "73--GA_1"
```

6 Combinación y remodelación

6.1 Importa las bases de datos `datos_caso_estudio_genes.txt` y los datos `datos.caso.biomarcadores.txt`

```r
genes <-read.table("ruta/a/mi/directorio/datos.caso.estudio_genes.txt",header=TRUE,sep="\t")

biomarcadores<-read.table("ruta/a/mi/directorio/datos.caso.estudio_biomarcadores.txt",header=TRUE,sep="
```

6.2 Recodifica los SNPS de la base de datos de genes de tal forma que los genotipos de los SNPs no tengan espacios vacios en sus valores

```
genes[1:10,1:10]
```

```
##            ID SNP_1 SNP_2 SNP_3 SNP_4 SNP_5 SNP_6 SNP_7 SNP_8 SNP_9
## 1   104--GA_2   A A   A A   A A   G A   A A   T G   G G   A A   A C
## 2   88--EXT_1   G A   A A   A A   G A   A A   G G   T T   C C   A C
## 3  163--EXT_2   A A   A A   C C   G G   A C   T G   T G   C C   C C
## 4   81--MAD_1   A A   A A   A C   G G   C C   G G   G G   A A   A A
## 5  103--MAD_1   G A   G G   A C   G A   A A   T G   T G   C C   C C
## 6  111--EXT_2   G G   A A   A C   A A   C C   T T   G G   A C   C C
## 7    1--MAD_2   G A   A A   A A   G A   A C   G G   G G   A A   C C
## 8  100--AND_1   G G   G G   A C   G G   C C   T G   T T   C C   C C
## 9  179--AND_1   G A   G G   A A   A A   A C   G G   T G   A C   A A
## 10  93--MAD_2   G A   G A   C C   A A   A C   T T   T T   A A   A A
```

```
genes[,-1]<-apply(genes[,-1],2,function(x) gsub(" ","",x))

# head(genes)
```

6.3 Une las tres bases de datos (`datos`, `genes` y `biomarcadores`) en una sola base mamteniendo los registros de la base `datos`

```
dim(datos);dim(genes);dim(biomarcadores)
```

```
## [1] 200  29
```

```
## [1] 200 101
```

```
## [1] 140   8
```

```
length(intersect(datos$ID,genes$ID))
```

```
## [1] 200
```

```
setdiff(datos$ID,biomarcadores$ID)
```

```
##  [1] "164--GA_1"  "84--EXT_2"  "93--MAD_2"  "77--GA_1"   "118--GA_1"
##  [6] "170--MAD_1" "188--MAD_2" "199--EXT_1" "53--MAD_1"  "75--AND_2"
## [11] "33--AND_2"  "158--EXT_2" "159--MAD_2" "9--GA_1"    "181--AND_1"
## [16] "107--MAD_1" "126--GA_1"  "97--GA_1"   "89--EXT_1"  "167--EXT_2"
## [21] "129--AND_1" "6--EXT_1"   "63--EXT_2"  "56--GA_2"   "136--AND_1"
## [26] "59--AND_2"  "1--MAD_2"   "2--EXT_1"   "65--MAD_2"  "51--AND_2"
## [31] "180--GA_1"  "28--AND_2"  "127--GA_1"  "182--MAD_1" "141--GA_2"
## [36] "61--EXT_2"  "73--GA_1"   "35--GA_2"   "134--MAD_2" "152--AND_2"
## [41] "37--MAD_1"  "111--EXT_2" "81--MAD_1"  "113--EXT_1" "142--MAD_2"
## [46] "191--MAD_2" "68--GA_2"   "168--AND_1" "85--MAD_1"  "131--AND_2"
## [51] "52--EXT_1"  "135--GA_2"  "123--GA_1"  "43--MAD_1"  "119--AND_2"
## [56] "42--MAD_2"  "145--GA_2"  "57--EXT_2"  "169--AND_2" "143--AND_2"
```

```
temp<-merge(datos,genes,by="ID")

todo<-merge(temp,biomarcadores,by="ID",all.x=TRUE)
```

6.4 Crea una base de datos en formato "long" con una unica columna para los SNPs y una unica columna para los biomarcadores

```
require(reshape)

columnas.snp=grep("^SNP_",names(todo))
temp=melt(todo,measure=columnas.snp,variable_name = "SNPs")

index=grep("value",names(temp))
names(temp)[index] <- "genotipo" #sustituye el nombre "value" por el nombre "genotipo"

columnas.bio=names(biomarcadores)[-1]
long=melt(temp,measure=columnas.bio,variable_name = "Biomarcador")

head(subset(long,select=c(ID:sexo,SNPs:value)))
```

```
##           ID edad   sexo  SNPs genotipo Biomarcador value
## 1   1--MAD_2   21 Hombre SNP_1       GA          Al    NA
## 2   10--GA_2    6  Mujer SNP_1       AA          Al  1.24
## 3 100--AND_1   69  Mujer SNP_1       GG          Al  3.30
## 4 101--EXT_2   47 Hombre SNP_1       GA          Al  1.20
## 5 102--EXT_2   40  Mujer SNP_1       GG          Al  2.10
## 6 103--MAD_1   67  Mujer SNP_1       GA          Al  2.63
```