La BD del proyecto de Bayesiana

Chávez Santiago, Rafael. Barra Melendrez, Enki Alonso.
Jeshua Romero, Guadarrama. Montaño Castro, David.

2022-05-17

R Markdown

Importacion de las posibles paqueterias a utilizar

```
library(MASS)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(dplyr)
library(datos)
library(htmlwidgets)
library(survival)
library(AUC)
```

```
## AUC 0.3.2
```

```
## Type AUCNews() to see the change log and ?AUC to get an overview.
```

```
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
## Attaching package: 'gdata'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     keep
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
library(dplyr)
library(DescTools)
```

```
## Registered S3 method overwritten by 'DescTools':
##   method         from
##   reorder.factor gdata
```

```
##
## Attaching package: 'DescTools'
```

```
## The following object is masked from 'package:gdata':
##
##     reorder.factor
```

```
library(openxlsx)
```

Datos a utilizar

Mandamos a llamar los datos que vamos a ocupar para el modelo

```
#Cargamos las bases de datos
tratamiento_art <- read.csv("/Users/enki/Documents/Modelo_Cox_Bayesiano/DBs/art_sim.csv")
informacion_basica <- read.csv("/Users/enki/Documents/Modelo_Cox_Bayesiano/DBs/basic_sim.csv")
seguimiento_paciente <- read.csv("/Users/enki/Documents/Modelo_Cox_Bayesiano/DBs/follow_sim.csv")
conteo_cd4 <- read.csv("/Users/enki/Documents/Modelo_Cox_Bayesiano/DBs/lab_cd4_sim.csv")
carga_viral <- read.csv("/Users/enki/Documents/Modelo_Cox_Bayesiano/DBs/lab_rna_sim.csv")
seguimiento_visitas <- read.csv("/Users/enki/Documents/Modelo_Cox_Bayesiano/DBs/visit_sim.csv")
```

Variables disponibles por cada archivo excel

DATOS DE art_sim.csv tratamiento_art

patient <– paciente site <– lugar de donde viene art_id <– tratamiento que sigue art_sd <– fecha de inicio de tratamiento art_ed <– fecha de termino del tratamiento (si no hay fecha de término el paciente continuó con dicho tratamiento) art_rs <– razón de cambio de tratamiento

DATOS DE basic_sim.csv informacion_basica

baseline_d <– fecha de enrrolamiento del paciente male <– 1 si es hombre, 0 si es mujer age <– edad del paciente birth_d <– fecha de nacimiento hivdiagnosis_d <– fecha de diagnostico mode <– modo de transmision de la enfermedad birth_d_a <– exactitud de la fecha registrada

DATOS DE follow_sim.csv seguimiento_paciente

l_alive_d <– ultima fecha en la que sabemos que el paciente aun se encuentra vivo, en contacto con el sistema death_y <– registro de su muerte, 1 si murio 0 si no murio death_d <– fecha de muerte

Sobre el CD4 Y RNA(CV)

Lo normal es que existan entre 500 y 1600 celulas CD4 por milimetro cubico de sangre rna es el numero de copias del virus por ml de sangre o por c/ml

DATOS DE lab_cd4_sim.csv conteo_cd4

cd4_d <– fecha en la que se realizo el conteo de cd4 cd4_v <– valor de cd4 correspondiente a su fecha en la que se realizo el conteo

DATOS DE lab_rna_sim.csv carga_viral

rna_d <– fecha del conteo de la carga viral rna_v <– valor de rna correspondiente a su fecha en la que se realizo el conteo si nos marca un rna_v = -40 nos informa que en ese momento era indectable

DATOS DE visit_sim.csv

visit_d <– fechas de visita del paciente ____

Fechas

Cambiamos las fechas para que aparezcan de una forma que nos sea facil tratarlas, i.e, cambiamos el formato en el cual aparecen y revisamos que estén en el formato actualizado

```
#Cambiamos el formato a tipo "Date"
tratamiento_art$art_sd <- as.Date(tratamiento_art$art_sd, "%Y-%m-%d")
class(tratamiento_art$art_sd)
```

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

```
tratamiento_art$art_ed <- as.Date(tratamiento_art$art_ed, "%Y-%m-%d")
class(tratamiento_art$art_ed)
```

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

```
informacion_basica$baseline_d <- as.Date(informacion_basica$baseline_d, "%Y-%m-%d")
class(informacion_basica$baseline_d)
```

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

```
informacion_basica$birth_d <- as.Date(informacion_basica$birth_d, "%Y-%m-%d")
class(informacion_basica$birth_d)
```

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

```
informacion_basica$hivdiagnosis_d <- as.Date(informacion_basica$hivdiagnosis_d, "%Y-%m-%d")
class(informacion_basica$hivdiagnosis_d)
```

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

```
seguimiento_paciente$l_alive_d <- as.Date(seguimiento_paciente$l_alive_d,  "%Y-%m-%d")
class(seguimiento_paciente$l_alive_d)
```

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

```
seguimiento_paciente$death_d <- as.Date(seguimiento_paciente$death_d,  "%Y-%m-%d")
class(seguimiento_paciente$death_d)
```

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

```
conteo_cd4$cd4_d <- as.Date(conteo_cd4$cd4_d, "%Y-%m-%d")
class(conteo_cd4$cd4_d)
```

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

```
carga_viral$rna_d <- as.Date(carga_viral$rna_d, "%Y-%m-%d")
class(carga_viral$rna_d)
```

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

```
seguimiento_visitas$visit_d <- as.Date(seguimiento_visitas$visit_d, "%Y-%m-%d")
class(seguimiento_visitas$visit_d)
```

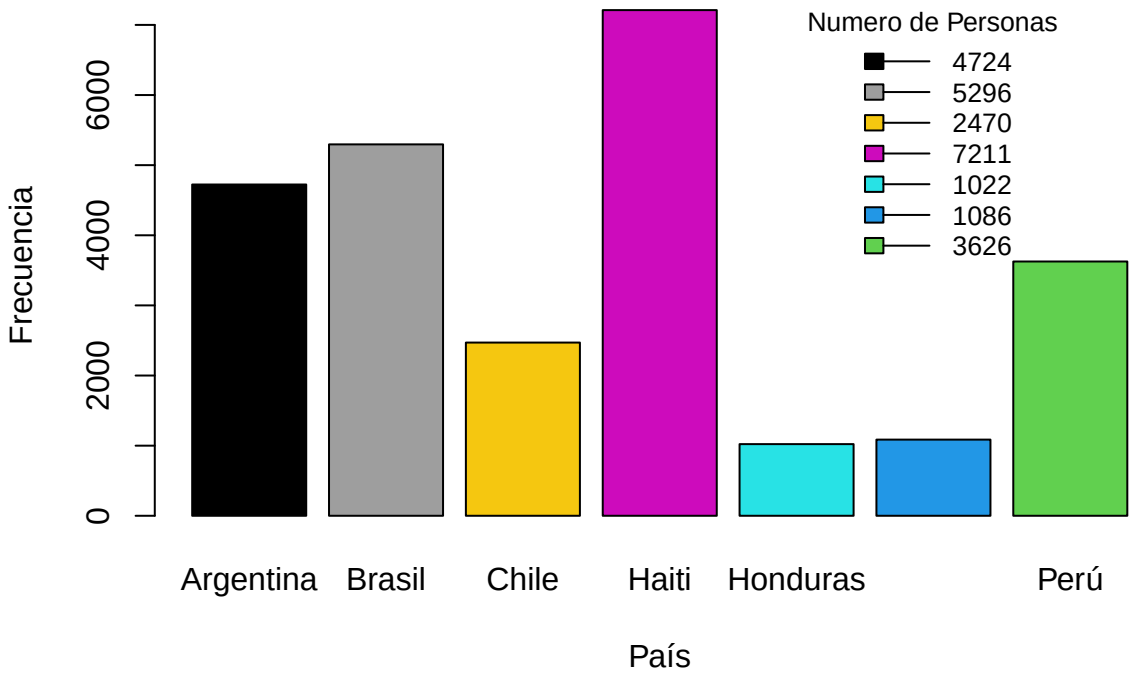
```
## [1] "Date"
```

Ya todas las variables estan en formato “Date”

Analisis de los datos base sin filtros. Lugares de origen de la poblacion de estudio.

Primero veremos de que lugares tenemos datos disponibles, dichos datos se encuentran en la tabla nombrada “informacion_basica”

```
# View(informacion_basica)

# Hacemos una tabla sobre la información por países
paises <- table(informacion_basica$site)
# View(paises)

#Cambiamos los nombres de la tabla para que estos esten a corde a cada país
names(paises)<- c("Argentina", "Brasil", "Chile", "Haiti", "Honduras", "México", "Perú")

#Generamos un grafico para visualizar la información
barplot(paises, col = 9:2, main = "Poblacion inicial en el estudio por país", ylab = "Frecuencia",
        xlab = "País")

#Veamos que clasificaiones generamos.
legend("topright", legend = paises, fill = 9:2, title = "Numero de Personas", cex = .8, xpd = TRUE,
       inset = c(.1, -.02), bty = "n", lwd = 1)
```



Análisis de los datos base sin filtros. Edades de la poblacion de estudio categorizadas.

Veamos que rangos de edades son los que tenemos disponibles para el estudio, datos disponibles en "informacion_basica"

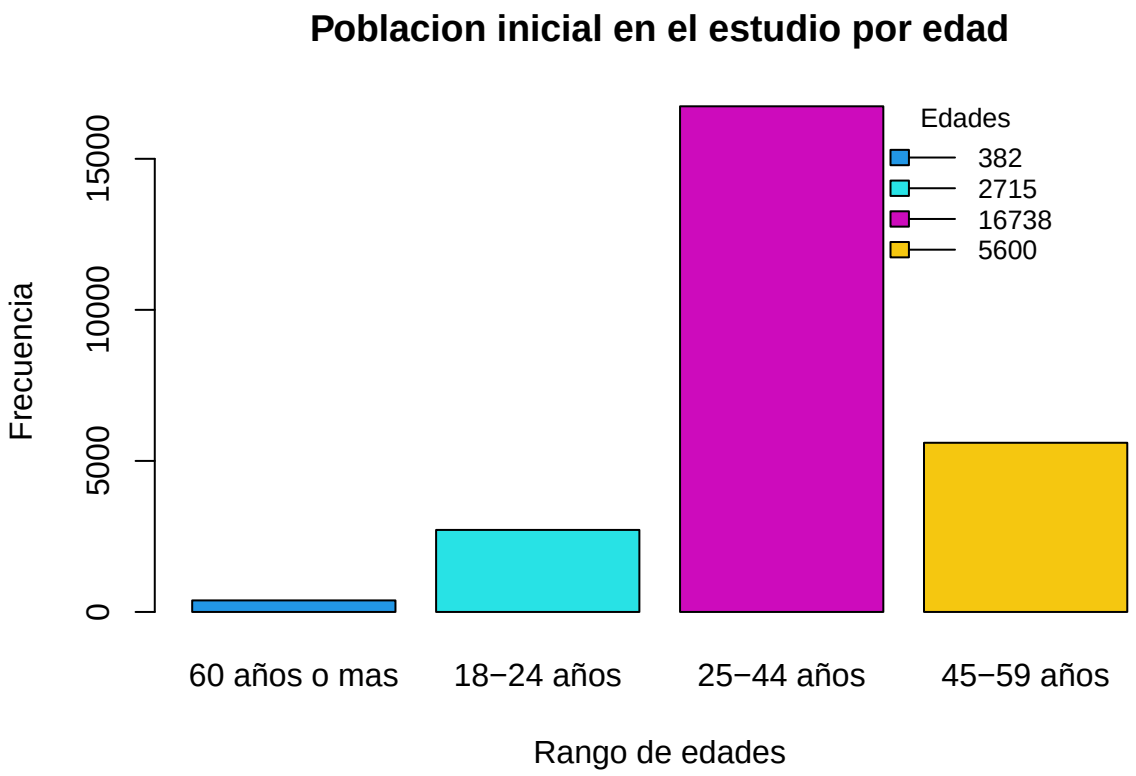
```
edades <- table(informacion_basica$age)
```

La tabla antes generada nos muestra edades diferentes. Se hará una agrupacion por edades clasificando a jóvenes de los 18 a los 24 años, adultos jóvenes antes de los entre los 25 y 44 años, adultos maduros entre los 45 y 59 años y finalmente los adultos mayores que tienen mas de 60 años

consultar link

```
# Primero añadimos una variable vacio a llenar de nulos para empezar la clasificación
informacion_basica$age_c = as.Date(NA)

# procedemos a realizar la separación por grupos
informacion_basica$age_c = factor(ifelse(informacion_basica$age<=24, "age18-24", ifelse(informacion_bas

#generamos la tabla de las edades categorizadas
categoria_edades <- table(informacion_basica$age_c)
# View(categoria_edades)

# Asignamos nombres a la tabla
names(categoria_edades) <- c("60 años o mas", "18-24 años", "25-44 años", "45-59 años")

# realizamos un grafico que nos permita ver los rangos de las edades
barplot(categoria_edades, col = 4:8, main = "Poblacion inicial en el estudio por edad", ylab = "Frecuen
        xlab = "Rango de edades")
```

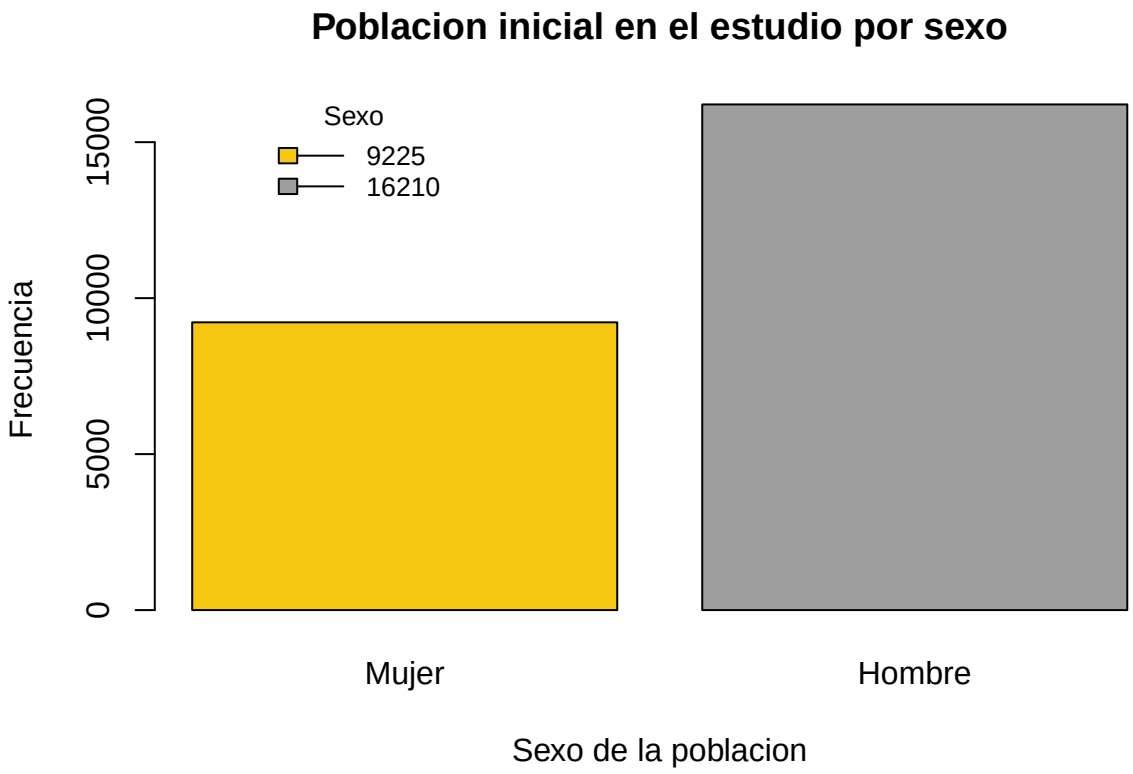
```
#veamos las clasificaciones generadas
legend("topright", legend = categoria_edades, fill = 4:8, title = "Edades", cex = .8, xpd = T, inset = c
```



Analisis de los datos base sin filtros. Sexo de la poblacion de estudio.

Veamos que porcentaje de hombres y mujeres se encuentran dentro de nuesta poblacion de estudio, datos disponibles en "informacion_basica"

```
#View(informacion_basica)
#Male es 1 si es hombre, 0 si es mujer

#Generamos la tabla de los sexos disponibles
sexo_poblacion <- table(informacion_basica$male)
# View(sexo_poblacion)

# asignamos clasificaion a los sexos
names(sexo_poblacion) <- c("Mujer", "Hombre")

# graficamos la tabla de sexos para visualizar la información
barplot(sexo_poblacion, col = 7:8, main = "Poblacion inicial en el estudio por sexo", ylab = "Frecuencia
        xlab = "Sexo de la poblacion")

#veamos las clasificaciones generadas
legend("topleft", legend = sexo_poblacion, fill = 7:8, title = "Sexo", cex = .8, xpd = T, inset = c(.1,
       bty = "n", lwd = 1)
```



Analisis de los datos base sin filtros. Tipo tratamiento de la poblacion de estudio categorizados.

Anteriormente separamos a la poblacion por grupo de edades y ahora lo haremos en base al tipo de tratamiento, estos grupos de tratamiento se separan en base al activo principal." Datos siponibles en "tratamiento_art"

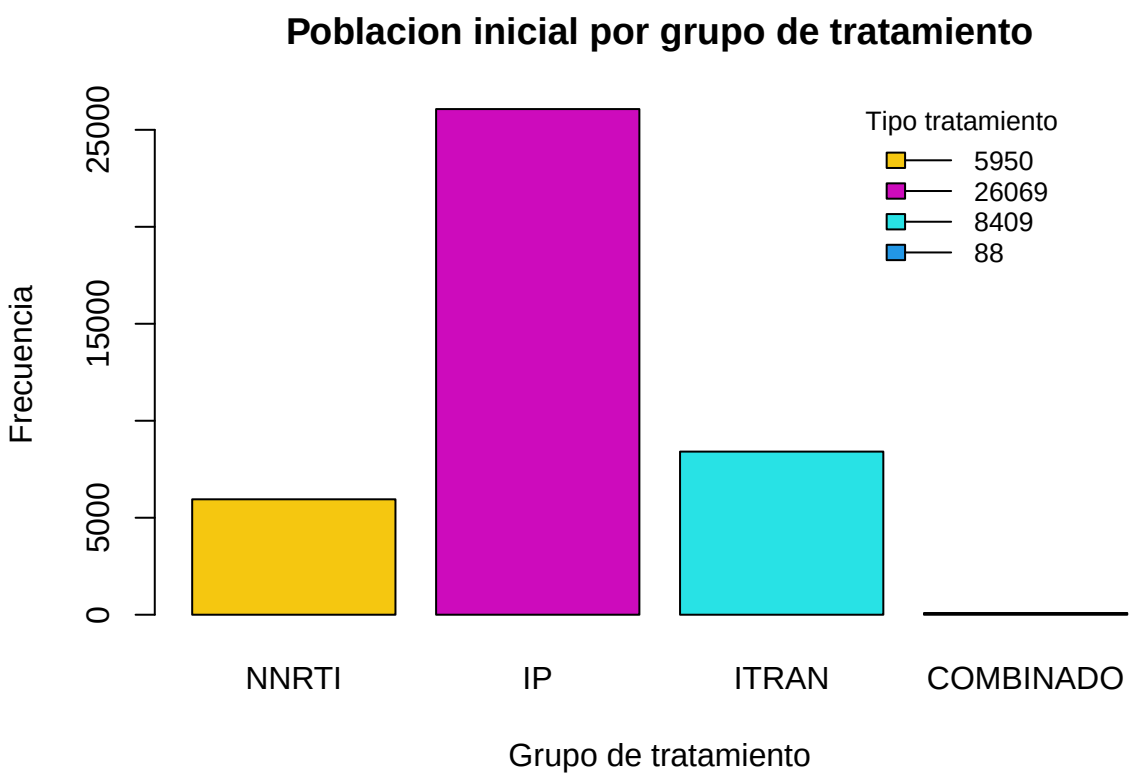
```
#Generamos variables dummy para separar los tipos de tratamiento disponibles.

# tratamientos
value_EFV = "EFV"
tratamiento_art$dummy_EFV <- grepl(value_EFV, tratamiento_art$art_id, fixed = TRUE)
value_NVP = "NVP"
tratamiento_art$dummy_NVP <- grepl(value_NVP, tratamiento_art$art_id, fixed = TRUE)
tratamiento_art$groupNNRTI <- ifelse(tratamiento_art$dummy_EFV==1 | tratamiento_art$dummy_NVP==1,1,0)

value_LPV = "LPV"
tratamiento_art$dummy_LPV <- grepl(value_NVP, tratamiento_art$art_id, fixed = T)
value_RTV = "RTV"
tratamiento_art$dummy_RTV <- grepl(value_RTV, tratamiento_art$art_id, fixed = T)
value_ATV = "ATV"
tratamiento_art$dummy_ATV <- grepl(value_ATV, tratamiento_art$art_id, fixed = T)
value_SQV = "SQV"
tratamiento_art$dummy_SQV <- grepl(value_SQV, tratamiento_art$art_id, fixed = T)
value_DRV = "DRV"
tratamiento_art$dummy_DVR <- grepl(value_DRV, tratamiento_art$art_id, fixed = T)
tratamiento_art$groupIP <- ifelse(tratamiento_art$dummy_LPV | tratamiento_art$dummy_RTV | tratamiento_ar

value_DLG ="DLG"
tratamiento_art$dummy_DLG <- grepl(value_DLG, tratamiento_art$art_id, fixed = T)
```

```
value_RAL = "RAL"
tratamiento_art$dummy_RAL <- grepl(value_RAL, tratamiento_art$art_id, fixed = T)

tratamiento_art$groupITRAN <- ifelse(tratamiento_art$dummy_DLG == 1 | tratamiento_art$dummy_RAL == 1, 1

tratamiento_art$group_art <- ifelse(tratamiento_art$groupNNRTI==1,1, ifelse(tratamiento_art$groupIP==1,1
                                ifelse(tratamiento_art$groupITRAN==1,3,0)))
```

Agrupamos en los tipos de tratamiento:

1. Aquellos que tienen EFV o NVP son un grupo (NNRTI)
2. Los que tienen LPV/RTV o ATV/RTV ó SQV ó DRV son otro grupo (IP).
3. Medicamentos como DLG o RAL, si es así ellos conforman otro grupo (ITRAN).
4. Los restantes se asignan a tratamiento combinado

```
# Hacemos la tabulacion de los datos por grupo de tratamiento
tipos_tratamiento <- table(tratamiento_art$group_art)
# View(tipos_tratamiento)

#asignamos nombres de cada grupo de tratamiento
names(tipos_tratamiento) <-c("NNRTI", "IP", "ITRAN", "COMBINADO")

# graficamos la tabla de los tratamientos
barplot(tipos_tratamiento, col = 7:3, main = "Poblacion inicial por grupo de tratamiento",
        ylab = "Frecuencia", xlab = "Grupo de tratamiento")

#veamos las clasificaciones generadas
legend("topright", legend = tipos_tratamiento, fill = 7:3, title = "Tipo tratamiento", cex = .8, xpd =
```



Limpieza_1 de variables que no usaremos en el modelo.

Comenzaremos con la limpieza de las tablas con las que contamos eliminando variables que no vamos a usar y la selección de los individuos que participaran en el estudio.

```
#Empezamos con la base de datos de tratamientos
tratamiento_art$art_ed <- NULL
tratamiento_art$pi <- NULL
tratamiento_art$nnrti1 <- NULL
tratamiento_art$nnrti2 <- NULL
tratamiento_art$nnrti <- NULL
tratamiento_art$nrti <- NULL
tratamiento_art$t20 <- NULL
tratamiento_art$ccr5 <- NULL
tratamiento_art$ii1 <- NULL
tratamiento_art$ii2 <- NULL
tratamiento_art$rtv_drug <- NULL
tratamiento_art$numdrugs <- NULL
tratamiento_art$art_class <- NULL
tratamiento_art$X <- NULL
tratamiento_art$dummy_ATV <- NULL
tratamiento_art$dummy_DLG <- NULL
tratamiento_art$dummy_DVR <- NULL
tratamiento_art$dummy_EFV <- NULL
tratamiento_art$dummy_LPV <- NULL
tratamiento_art$dummy_NVP <- NULL
tratamiento_art$dummy_RAL <- NULL
tratamiento_art$dummy_RTV <- NULL
tratamiento_art$dummy_SQV <- NULL
tratamiento_art$groupIP <- NULL
tratamiento_art$groupITRAN <- NULL
tratamiento_art$groupNNRTI <- NULL
tratamiento_art$art_rs <- NULL

# Seguimos con la base de datos de la informacion basica de cada paciente
informacion_basica$aids_y <- NULL
informacion_basica$aids.miss <- NULL
informacion_basica$aids_cl_d <- NULL
informacion_basica$aids_cl_y<-NULL
informacion_basica$recart_y<-NULL
informacion_basica$aids_d<-NULL
informacion_basica$mode_oth<-NULL
informacion_basica$clinicaltrial_y<-NULL
informacion_basica$baseline_d_num <- NULL
informacion_basica$hivdiagnosis_d_num <- NULL

# Continuamos con los datos de seguimiento de los pacientes
seguimiento_paciente$drop_rs_oth<-NULL
seguimiento_paciente$drop_rs<-NULL
seguimiento_paciente$death_d_a<-NULL
seguimiento_paciente$death_d <- NULL

# Para las cuentas de CD4
seguimiento_visitas$cdcstage<-NULL
seguimiento_visitas$whostage<-NULL
```

```r
# Visualizamos las variables de cada tabla que nos restan
head(carga_viral)
```

```
##   patient      site      rna_d  rna_v
## 1    ar.1 argentina 2007-01-19  74724
## 2    ar.1 argentina 2013-02-04    -40
## 3    ar.1 argentina 2011-11-03    399
## 4    ar.1 argentina 2010-11-16 407000
## 5    ar.1 argentina 2009-08-26    -50
## 6    ar.1 argentina 2008-05-07    -50
```

```r
head(conteo_cd4)
```

```
##   patient      site      cd4_d cd4_v time
## 1    ar.1 argentina 2007-01-19   405    0
## 2    ar.1 argentina 2008-05-07   490  474
## 3    ar.1 argentina 2009-08-26   238  950
## 4    ar.1 argentina 2010-11-16   451 1397
## 5    ar.1 argentina 2011-11-03   811 1749
## 6    ar.1 argentina 2013-02-04   238 2208
```

```r
head(informacion_basica)
```

```
##   patient      site baseline_d male      age    birth_d hivdiagnosis_d
## 1    ar.1 argentina 2007-04-13    1 34.16329 1973-02-12     2007-04-13
## 2    ar.2 argentina 2010-07-06    0 45.89359 1964-08-13     1999-01-07
## 3    ar.3 argentina 2011-03-28    1 47.23421 1964-01-02     1999-08-13
## 4    ar.4 argentina 2002-04-19    0 31.46904 1970-10-30     1996-09-14
## 5    ar.5 argentina 2004-12-27    1 36.88613 1968-02-07     2002-10-23
## 6    ar.6 argentina 2008-08-19    1 35.46968 1973-03-01     2008-08-19
##                                 mode birth_d_a    age_c
## 1                Homosexual contact         D age25-44
## 2               Injecting drug user         D age45-59
## 3              Heterosexual contact         D age45-59
## 4                           Unknown         D age25-44
## 5 Transfusion nonhemophilia related         D age25-44
## 6              Heterosexual contact         D age25-44
```

```r
head(seguimiento_paciente)
```

```
##   patient      site  l_alive_d death_y
## 1    ar.1 argentina 2013-02-04       0
## 2   ar.10 argentina 2013-02-13       0
## 3  ar.100 argentina 2013-07-12       0
## 4 ar.1000 argentina 2012-11-10       0
## 5 ar.1001 argentina 2013-06-21       0
## 6 ar.1002 argentina 2014-01-25       0
```

```r
head(seguimiento_visitas)
```

```
##   patient      site    visit_d
## 1    ar.1 argentina 2007-04-13
## 2    ar.2 argentina 2010-07-06
## 3    ar.3 argentina 2011-03-28
## 4    ar.4 argentina 2002-04-19
## 5    ar.5 argentina 2004-12-27
```

```
## 6    ar.6 argentina 2008-08-19
```

```
head(tratamiento_art)
```

```
##   patient      site      art_id     art_sd group_art
## 1    ar.1 argentina 3TC,AZT,NVP 2007-05-16         1
## 2    ar.1 argentina 3TC,AZT,EFV 2007-05-30         1
## 3    ar.1 argentina 3TC,ABC,AZT 2007-08-03         0
## 4   ar.10 argentina 3TC,AZT,EFV 2002-02-07         1
## 5  ar.100 argentina 3TC,ABC,AZT 2006-06-01         0
## 6  ar.100 argentina 3TC,AZT,EFV 2006-07-16         1
```

Limpieza de cada tabla de forma individual. conteo_cd4

Lo normal es que existan entre 500 y 1600 celulas CD4 por milimetro cubico de sangre

```
# Primer tabla a limpiar conteo_cd4

# cd4_d <-- fecha en la que se realizo el conteo de cd4
# cd4_v <-- valor de cd4 correspondiente a su fecha en la que se realizo el conteo

# Veamos que la tabla empieza sin valores NA's
sum(is.na(conteo_cd4$patient))
```

```
## [1] 0
```

```
sum(is.na(conteo_cd4$site))
```

```
## [1] 0
```

```
sum(is.na(conteo_cd4$cd4_d))
```

```
## [1] 1815
```

```
sum(is.na(conteo_cd4$cd4_v))
```

```
## [1] 1815
```

```
sum(is.na(conteo_cd4$time))
```

```
## [1] 1815
```

```
# Tenemos menos del 5% de datos faltantes en las cuentas asi que los eliminamos
conteo_cd4 <- na.omit(conteo_cd4)

# Comprobamos no tener NA's
sum(is.na(conteo_cd4$patient))
```

```
## [1] 0
```

```
sum(is.na(conteo_cd4$site))
```

```
## [1] 0
```

```
sum(is.na(conteo_cd4$cd4_d))
```

```
## [1] 0
```

```
sum(is.na(conteo_cd4$cd4_v))
```

```
## [1] 0
```

```r
sum(is.na(conteo_cd4$time))
```

```
## [1] 0
```

```r
# todo suma 0, no hay NA's

# Para conservar la prier cuenta de celulas cd4 basta con usar la variable "time" la cual nos esta midi

#Usamos un for para recorrer toda la variable "time", si tengo un tiempo distinto de 0 entonces no es e

for (i in 1:length(conteo_cd4$patient)) {
  if (conteo_cd4$time[i] != 0) {
    conteo_cd4$time[i] = NA
  }
}

#Eliminamos las tuplas/filas que tienen na's pues no son de nuestro interes
conteo_cd4 <- na.omit(conteo_cd4)

#Categorizamos los niveles de cd4 por niveles de celulas
conteo_cd4$cd4_v = factor(ifelse(conteo_cd4$cd4_v<200, "CD4_inicial <200", ifelse(conteo_cd4$cd4_v<=350

#la variable "time" ya la usamos y no nos sera de utilidad mas adelante asi que la eliminamos
conteo_cd4$time <- NULL

#Le damos una checada a la tabla para ver como va
#View(conteo_cd4)

#Revisamos si contienen valores na alguna de las variables en esta tabla
sum(is.na(conteo_cd4$patient))
```

```
## [1] 0
```

```r
sum(is.na(conteo_cd4$site))
```

```
## [1] 0
```

```r
sum(is.na(conteo_cd4$cd4_d))
```

```
## [1] 0
```

```r
sum(is.na(conteo_cd4$cd4_v))
```

```
## [1] 0
```

```r
#Todas nos arrojan una suma de 0 que nos indica la existencia de puros bulenaos FALSE asi que no hay NA
```

Limpieza de cada tabla de forma individual. carga_viral

rna es el numero de copias del virus por ml de sangre o por c/ml

```r
#Vamos a conservar el primer recuento de CV
#Primero veamos si tenemos na's
sum(is.na(carga_viral$patient))
```

```
## [1] 0
```

```r
sum(is.na(carga_viral$site))
```

```
## [1] 0
```

```
sum(is.na(carga_viral$rna_d))
```

```
## [1] 1815
```

```
sum(is.na(carga_viral$rna_v))
```

```
## [1] 2183
```

```
#Tenemos faltantes en los registros de CV asi que no podemos aplicar a ellos el modelo asi que los elim
carga_viral <-na.omit(carga_viral)

# comprovamos que no existan na's por segunda vez
sum(is.na(carga_viral$patient))
```

```
## [1] 0
```

```
sum(is.na(carga_viral$site))
```

```
## [1] 0
```

```
sum(is.na(carga_viral$rna_d))
```

```
## [1] 0
```

```
sum(is.na(carga_viral$rna_v))
```

```
## [1] 0
```

```
#todas suma 0, i.e., solo hay FALSE que nos indican que no hay na's

#Creamos una variable que nos servira de apoyo
carga_viral$eliminar = as.Date(NA)

# otra checadita a la tabla
# View(carga_viral)

# Generamos un vector de apoyo de los diferentes pacientes
patients <- unique(carga_viral$patient)

# Con un bucle for vamos a revisar cada entrada de la variable "rna_d" para obtener la fecha minima que

for (i in 1:length(patients)){
  reg = patients[i]
  datos_1 = carga_viral %>% filter(patient == reg) #submarco de datos obtendo por paciente
  datos_1$eliminar <- rep(0,length(datos_1$patient))
  datos_1$eliminar = as.Date(min(datos_1$rna_d))   # Fecha de inicio del paciente que cumple las condic
  carga_viral$eliminar[carga_viral$patient==reg] = datos_1$eliminar
}

# otra checadita a la tabla
# View(carga_viral)

# Ya tenemos una variable para hacer un comparativo, vamos a comparar las fechas en las que se sacaron
# vamos a comparar las fechas y si no coinciden vamos a asignar un NA para posteriormente eliminar toda

for (i in 1:length(patients)){# hice cambio
  if (carga_viral$rna_d[i] != carga_viral$eliminar[i]) {
```

```
    carga_viral$eliminar[i] = NA
  }
}

#otra checadita a la tabla
#View(carga_viral)

#Procedemos a eliminar los na's
carga_viral <- na.omit(carga_viral)

#otra checadita a la tabla
View(carga_viral)
#Ya tenemos los primeros recuentos de CV

#Procedemos a clasificar los niveles de CV
carga_viral$rna_v = factor(ifelse(carga_viral$rna_v<=50, "CV indetectable", ifelse(carga_viral$rna_v<500

#otra checadita mas a la tabla
#View(carga_viral)

#Desechamos la variable que usamos de apoyo pues ya no nos servira mas
carga_viral$eliminar <- NULL

#Ya tenemos la tabla de datos lista
```

aqui me quedé ## Limpieza de cada tabla de forma individual. tratamiento_art_primer_medicamento

Conservando el primer tratamiento al cual se somete el paciente

```
#View(tratamiento_art)
#Para conservar el primer tratamiento de cada paciente vamos a hacer algo parecido a lo que hicimos par

#Veamos si tenemos NA's
sum(is.na(tratamiento_art$patient))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$site))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$art_id))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$art_sd))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$group_art))
```

```
## [1] 0
```

```
#todo suma 0 lo que nos indica que no tenemos  NA's

#como no habia NA's no pasa nada al ejecutar el sig comando
tratamiento_art <- na.omit(tratamiento_art)

#Creamos la variable de apoyo
```

```
tratamiento_art$eliminar = as.Date(NA)

#otra checadita a la tabla
#View(tratamiento_art)

#Generamos un vector de apoyo de los diferentes pacientes
patients <- unique(tratamiento_art$patient)

# Con un for vamos a revisar cada entrada de la variable "art_sd" para obtener la fecha minima en la cu

for (i in 1:length(patients)) { #hice cambio
  reg = patients[i]
  datos_1 = tratamiento_art %>% filter(tratamiento_art$patient==reg)
  datos_1$eliminar=NA
  datos_1$eliminar = as.Date(min(datos_1$art_sd))   ### Fecha de inicio del paciente que cumple las con
  tratamiento_art$eliminar[tratamiento_art$patient==reg] = datos_1$eliminar
}

#otra checadita a la tabla
#View(tratamiento_art)

#Procedemos a eliminar los na's
tratamiento_art <- na.omit(tratamiento_art)

# Hacemos el comparativo para generar los NA's en la variable "eliminar" y despues usar "na.omit()"
for (i in 1:length(patients)) { #hice cambios
  if (tratamiento_art$art_sd[i] != tratamiento_art$eliminar[i]) {
    tratamiento_art$eliminar[i] = NA
  }
}

#otra checadita a la tabla
#View(tratamiento_art)

#Eliminamos los NA's y vamos a almacenar esta informacion en una tabla de primer medicamento asignado a
tratamiento_art_primer_medicamento <- na.omit(tratamiento_art)

#Una checadita a la tabla nueva que generamos
#View(tratamiento_art_primer_medicamento)

#Eliminamos la variable de apoyo que creamos dentro de las tablas "tratamiento_art" y "tratamiento_art_
tratamiento_art$eliminar <- NULL
tratamiento_art_primer_medicamento$eliminar <-NULL
```

Limpieza de cada tabla de forma individual. tratamiento_art_ultimo_medicamento

Conservando el primer tratamiento al cual se somete el paciente

```
#View(tratamiento_art)

# Para conservar el primer tratamiento de cada paciente vamos a hacer algo parecido a lo que hicimos pa

# Veamos si tenemos NA's
sum(is.na(tratamiento_art$patient))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$site))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$art_id))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$art_sd))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$group_art))
```

```
## [1] 0
```

```
# todo suma 0 lo que nos indica que no tenemos  NA's

#como no habia NA's no pasa nada al ejecutar el sig comando
tratamiento_art <- na.omit(tratamiento_art)


# Creamos la variable de apoyo
tratamiento_art$eliminar = as.Date(NA)

# otra checadita a la tabla
# View(tratamiento_art)

# Generamos un vector de apoyo de los diferentes pacientes
patients <- unique(tratamiento_art$patient)

# Con un for vamos a revisar cada entrada de la variable "art_sd" para obtener la fecha minima en la cu

for (i in 1:length(patients)){ #hice cambios
  reg = patients[i]
  datos_1 = tratamiento_art %>%  filter(tratamiento_art$patient==reg)
  datos_1$eliminar=NA
  datos_1$eliminar = as.Date(max(datos_1$art_sd))   ### Fecha de inicio del paciente que cumple las con
  tratamiento_art$eliminar[tratamiento_art$patient==reg] = datos_1$eliminar
}

# otra checadita a la tabla
# View(tratamiento_art)

# Hacemos el comparativo para generar los NA's en la variable "eliminar" y despues usar "na.omit()"
for (i in 1:length(patients)) {
  if (tratamiento_art$art_sd[i] != tratamiento_art$eliminar[i]) {
    tratamiento_art$eliminar[i] = NA
  }
}

# otra checadita a la tabla
# View(tratamiento_art)

# Eliminamos los NA's y vamos a almacenar esta informacion en una tabla de ultimo medicamento asignado
```

```
tratamiento_art_ultimo_medicamento <- na.omit(tratamiento_art)

# Una checadita a la tabla nueva que generamos
# View(tratamiento_art_ultimo_medicamento)

# Eliminamos la variable de apoyo que creamos dentro de las tablas "tratamiento_art" y "tratamiento_art_
tratamiento_art$eliminar <- NULL
tratamiento_art_ultimo_medicamento$eliminar <-NULL
```

Limpieza de cada tabla de forma individual. Comprobacion y Merge

Veamos que los datos almacenados en cada tabla de excel son unicos para cada paciente, para asi poderlas unir y formar nuestra tabla antes de empezar la seleccion de los datos para formar la coorte de estudio del proyecto

```
# Visualizamos las variables de cada tabla que nos restan
head(carga_viral)
```

```
##     patient      site      rna_d           rna_v
## 1      ar.1 argentina 2007-01-19         CV alta
## 21    ar.10 argentina 2002-09-30 CV indetectable
## 50   ar.100 argentina 2007-12-11 CV indetectable
## 64  ar.1000 argentina 2007-01-22 CV indetectable
## 97  ar.1001 argentina 2003-11-18         CV alta
## 106 ar.1002 argentina 2001-05-15 CV indetectable
```

```
head(conteo_cd4)
```

```
##     patient      site      cd4_d            cd4_v
## 1      ar.1 argentina 2007-01-19 CD4_inicial >350
## 7     ar.10 argentina 2002-09-30 CD4_inicial <200
## 38   ar.100 argentina 2007-12-11 CD4_inicial <200
## 59  ar.1000 argentina 2007-01-22 CD4_inicial >350
## 90  ar.1001 argentina 2003-11-18 CD4_inicial >350
## 102 ar.1002 argentina 2001-05-15 CD4_inicial <200
```

```
head(informacion_basica)
```

```
##   patient      site baseline_d male      age    birth_d hivdiagnosis_d
## 1    ar.1 argentina 2007-04-13    1 34.16329 1973-02-12     2007-04-13
## 2    ar.2 argentina 2010-07-06    0 45.89359 1964-08-13     1999-01-07
## 3    ar.3 argentina 2011-03-28    1 47.23421 1964-01-02     1999-08-13
## 4    ar.4 argentina 2002-04-19    0 31.46904 1970-10-30     1996-09-14
## 5    ar.5 argentina 2004-12-27    1 36.88613 1968-02-07     2002-10-23
## 6    ar.6 argentina 2008-08-19    1 35.46968 1973-03-01     2008-08-19
##                                 mode birth_d_a    age_c
## 1                Homosexual contact         D age25-44
## 2               Injecting drug user         D age45-59
## 3              Heterosexual contact         D age45-59
## 4                            Unknown         D age25-44
## 5 Transfusion nonhemophilia related         D age25-44
## 6              Heterosexual contact         D age25-44
```

```
head(seguimiento_paciente)
```

```
##   patient      site  l_alive_d death_y
```

```
## 1    ar.1 argentina 2013-02-04       0
## 2   ar.10 argentina 2013-02-13       0
## 3  ar.100 argentina 2013-07-12       0
## 4 ar.1000 argentina 2012-11-10       0
## 5 ar.1001 argentina 2013-06-21       0
## 6 ar.1002 argentina 2014-01-25       0
```

```
head(seguimiento_visitas)
```

```
##   patient      site    visit_d
## 1    ar.1 argentina 2007-04-13
## 2    ar.2 argentina 2010-07-06
## 3    ar.3 argentina 2011-03-28
## 4    ar.4 argentina 2002-04-19
## 5    ar.5 argentina 2004-12-27
## 6    ar.6 argentina 2008-08-19
```

```
head(tratamiento_art_primer_medicamento)
```

```
##    patient      site          art_id     art_sd group_art
## 1     ar.1 argentina     3TC,AZT,NVP 2007-05-16         1
## 4    ar.10 argentina     3TC,AZT,EFV 2002-02-07         1
## 5   ar.100 argentina     3TC,ABC,AZT 2006-06-01         0
## 8  ar.1000 argentina     3TC,AZT,EFV 2006-05-29         1
## 10 ar.1001 argentina 3TC,AZT,LPV,RTV 2004-12-05         2
## 13 ar.1002 argentina     3TC,AZT,NVP 1999-06-16         1
```

```
head(tratamiento_art_ultimo_medicamento)
```

```
##    patient      site          art_id     art_sd group_art
## 3     ar.1 argentina     3TC,ABC,AZT 2007-08-03         0
## 4    ar.10 argentina     3TC,AZT,EFV 2002-02-07         1
## 7   ar.100 argentina     3TC,AZT,NVP 2006-09-26         1
## 9  ar.1000 argentina     3TC,AZT,NFV 2006-12-05         0
## 12 ar.1001 argentina 3TC,AZT,LPV,RTV 2008-07-25         2
## 13 ar.1002 argentina     3TC,AZT,NVP 1999-06-16         1
```

Valores NA's de cada tabla individual

Veamos si las tablas por separado contienen valores NA's.

```
# comprovacion rapida de existencia de na's


sum(is.na(tratamiento_art$patient))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$site))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$art_id))
```

```
## [1] 0
```

```
sum(is.na(tratamiento_art$art_sd))
```

```
## [1] 0
```

```r
sum(is.na(tratamiento_art$art_ed))
```

```
## [1] 0
```

```r
sum(is.na(tratamiento_art$art_rs))
```

```
## [1] 0
```

```r
sum(is.na(tratamiento_art$group_art))
```

```
## [1] 0
```

```r
# todas suman 0 lo que nos iddica no hay NA's

sum(is.na(informacion_basica$patient))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$site))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$baseline_d))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$male))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$age))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$birth_d))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$hivdiagnosis_d))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$mode))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$birth_d_a))
```

```
## [1] 0
```

```r
sum(is.na(informacion_basica$age_c))
```

```
## [1] 0
```

```r
# todas suman 0 lo que nos iddica no hay NA's

sum(is.na(seguimiento_paciente$patient))
```

```
## [1] 0
```

```r
sum(is.na(seguimiento_paciente$site))
```

```
## [1] 0
```

```r
sum(is.na(seguimiento_paciente$l_alive_d))
```

```
## [1] 0
```

```r
sum(is.na(seguimiento_paciente$death_y))
```

```
## [1] 0
```

```r
# todas suman 0 lo que nos iddica no hay NA's

sum(is.na(carga_viral$patient))
```

```
## [1] 0
```

```r
sum(is.na(carga_viral$site))
```

```
## [1] 0
```

```r
sum(is.na(carga_viral$rna_d))
```

```
## [1] 0
```

```r
sum(is.na(carga_viral$rna_v))
```

```
## [1] 0
```

```r
# todas suman 0 lo que nos iddica no hay NA's

sum(is.na(conteo_cd4$patient))
```

```
## [1] 0
```

```r
sum(is.na(conteo_cd4$site))
```

```
## [1] 0
```

```r
sum(is.na(conteo_cd4$cd4_d))
```

```
## [1] 0
```

```r
sum(is.na(conteo_cd4$cd4_v))
```

```
## [1] 0
```

```r
# todas suman 0 lo que nos iddica no hay NA's

sum(is.na(seguimiento_visitas$patient))
```

```
## [1] 0
```

```r
sum(is.na(seguimiento_visitas$site))
```

```
## [1] 0
```

```r
sum(is.na(seguimiento_visitas$visit_d))
```

```
## [1] 0
```

```r
# todas suman 0 lo que nos iddica no hay NA's
```

Union para generar los datos con los que trabajaremos. Datos de cada tabla a usar con primer medicamento asignado

Finalmente con un único registro por cada paciente dentro de cada tabla procedemos a unirlas para generar nuestra "BD" con la que haremos el modelo

```
b1 <- merge(carga_viral, conteo_cd4, by=c("patient", "site"), all = T)
# View(b1)
# Comprovamos que no se nos generarn NA's
sum(is.na(b1$patient))
```

```
## [1] 0
```

```
sum(is.na(b1$site))
```

```
## [1] 0
```

```
sum(is.na(b1$rna_d))
```

```
## [1] 7299
```

```
sum(is.na(b1$rna_v))
```

```
## [1] 7299
```

```
sum(is.na(b1$cd4_d))
```

```
## [1] 0
```

```
sum(is.na(b1$cd4_v))
```

```
## [1] 0
```

```
b1 <- na.omit(b1)
# Vistaso rapido a la tabla
# View(b1)
```

Se generaron algunos na's al unir tabla lo que nos indica que tenemos pacientes con faltantes en los registros de CV, por esta razon no podemos usarlos pues son valores que no entran dentro de lo que necesitamos para el estudio y los omitimos por falta de información

```
b2 <- merge(informacion_basica, b1, by=c("patient", "site"), all = T)
# View(b2)

#Comprovamos que no se nos generarn NA's
sum(is.na(b2$patient))
```

```
## [1] 0
```

```
sum(is.na(b2$site))
```

```
## [1] 0
```

```
sum(is.na(b2$baseline_d))
```

```
## [1] 0
```

```
sum(is.na(b2$male))
```

```
## [1] 0
```

```
sum(is.na(b2$age))
```

```
## [1] 0
```

```
sum(is.na(b2$birth_d))
```

```
## [1] 0
```

```
sum(is.na(b2$hivdiagnosis_d))
```

```
## [1] 0
```

```
sum(is.na(b2$mode))
```

```
## [1] 0
```

```
sum(is.na(b2$birth_d_a))
```

```
## [1] 0
```

```
sum(is.na(b2$rna_d))
```

```
## [1] 9325
```

```
sum(is.na(b2$rna_v))
```

```
## [1] 9325
```

```
sum(is.na(b2$cd4_d))
```

```
## [1] 9325
```

```
sum(is.na(b2$cd4_v))
```

```
## [1] 9325
```

```
#Se generaron NA's ocacionando incompletitud de informacion en las variables, asi que descartamos las t
b2 <- na.omit(b2)
```

```
# Vistazo rapido a la tabla
# View(b2)

b3 <- merge(seguimiento_paciente, b2, by=c("patient", "site"), all = T)
# View(b3)
# Comprovamos que no se nos generarn NA's
sum(is.na(b3$patient))
```

```
## [1] 0
```

```
sum(is.na(b3$site))
```

```
## [1] 0
```

```
sum(is.na(b3$l_alive_d))
```

```
## [1] 0
```

```
sum(is.na(b3$death_y))
```

```
## [1] 0
```

```
sum(is.na(b3$baseline_d))
```

```
## [1] 9325
```

```
sum(is.na(b3$male))
```

```
## [1] 9325
```

```r
sum(is.na(b3$age))
```

```
## [1] 9325
```

```r
sum(is.na(b3$birth_d))
```

```
## [1] 9325
```

```r
sum(is.na(b3$hivdiagnosis_d))
```

```
## [1] 9325
```

```r
sum(is.na(b3$mode))
```

```
## [1] 9325
```

```r
sum(is.na(b3$birth_d_a))
```

```
## [1] 9325
```

```r
sum(is.na(b3$rna_d))
```

```
## [1] 9325
```

```r
sum(is.na(b3$rna_v))
```

```
## [1] 9325
```

```r
sum(is.na(b3$cd4_d))
```

```
## [1] 9325
```

```r
sum(is.na(b3$cd4_v))
```

```
## [1] 9325
```

```r
# Se generan tuplas/filas incompletas y no las vamos a poder usar en la coorte de estudio asi que las e
b3 <- na.omit(b3)

#
b4 <- merge(seguimiento_visitas, b3, by=c("patient", "site"), all = T)

# Si se generaron NA's los eliminamos por falta de informacion
b4 <- na.omit(b4)
```

Vamos a generar dos posibles tablas de datos para el modelo, una basada en el primer medicamento asignado a los pacientes y otra basada en el ultimo medicamento asignado a los pacientes

```r
#Primer medicamento
primer_medicamento_bd <- merge(tratamiento_art_primer_medicamento, b4, by=c("patient", "site"),
                               all = T)

# Omitimos tuplas incompletas que no nos sirven
primer_medicamento_bd <- na.omit(primer_medicamento_bd)

# Ultimo medicamento
ultimo_medicamento_bd <- merge(tratamiento_art_ultimo_medicamento, b4, by=c("patient", "site"),
                               all = T)

ultimo_medicamento_bd <- na.omit(ultimo_medicamento_bd)
```

Exportar los dataframe de R a excel para su use externo

Vamos a generar 2 archivos en excel para poder usarlos como BD alternas y no ejecutar todo este codigo para obtenerlas

```
write.csv(primer_medicamento_bd, "Primer_medicamento_base_Bayesiana.csv")

write.csv(ultimo_medicamento_bd, "Ultimo_medicamento_base_Bayesiana.csv")
```