Script

In [ ]:

```
### 18 buono
aplicacion_actuarial <- function( aplicacion = 'tabla'){
  # Zittiendo le avvertenze
  duepalle <- getOption('warn')
  options(warn = -1)
  list.of.packages <- c('plotly', 'httr', 'data.table')
  new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,'Package'])]
  if(length(new.packages)) install.packages(new.packages)
  suppressMessages(library(data.table))
  suppressMessages(library(plotly))
  suppressMessages(library(httr))
    # caricamento del dataset
  cohortes_g20 <- read.csv('lifeTable.csv')
    # definizione dei valori delle femmine
  paises <- colnames(cohortes_g20)
  hembra <- paises[c(30:length(paises))]
  uomini <- paises[c(2:29)]
  n2 <- c(57, 56, 34, 39, 43, 40, 44, 51)
  hembrasg8 <- paises[n2]
  hembra <- cbind(hembrasg8, n2)
  # Chiamando alla funzione di imputazione
  m1 <- readline(prompt = 'Ingrese un Pais del G8: ')
  if(m1 %in% hembra){

    z1 <- as.numeric(hembra[which(hembra==m1),2])
    f1 <- cohortes_g20[,z1]

  x1 <- shift(f1, -1); age <- 0:110
  x1 <- nafill(x1, fill = 0)
  px <- x1/f1;  qx <- 1- px
  dx <- f1*qx;  Lx <- (x1+f1)/2
  mx <- dx/Lx; #mx[length(mx)] <- 0 Alabama
  url <- "https://ipinfo.io/"
  a1 <- GET(url);a2 <- content(a1)
  a3 <- paste('Lugar de ejecuciòn:',a2$city,',',a2$region, a2$country)
  a4 <- paste('Fecha de ejecuciòn:',format(Sys.time(), "%A, %b %d %X %Y"))
  e <- c()
  for (i in 1:111){
    lex <- f1[i]
    e <- append(e,(sum(f1[i:length(f1)])/lex)-1)
}
  tab <- data.frame(matrix(c(age,f1,dx, px, qx, Lx, mx, round(e,2)), ncol = 8))
  colnames(tab) <- c('edad','lx', 'dx', 'px', 'qx', 'Lx', 'mx', 'ex')

      if((aplicacion!='tabla')|(aplicacion!='grafico')|(aplicacion!='conmutaciones')){
  if(aplicacion =='tabla'){

      print(a3);print(toupper(a4))
      return(tab)
  }else if (aplicacion =='grafico'){
    m11 <- readline(prompt = 'Desea el Plot logaritmico de qx: ')
    print(a3);print(toupper(a4))
    if (m11 == 'no'){

      fig1 <- plot_ly(tab, x=~edad, y= ~lx, line = list(color='red'),
        type = 'scatter', mode='lines', text=~ex)
      fig1 <- fig1 %>% layout(title="Sobreviventes - Esperanza de vida")
      return(fig1)

    }else{

      fig2 <- plot_ly(tab, x=~edad, y= ~qx, line = list(color='red'),
                      type = 'scatter', mode='lines', text=~ex)
      fig2 <- fig2 %>% layout(title="Tasa mortalidad Log - Esperanza de vida")
      fig2 <- layout(fig2, yaxis = list(type = "log"))
      # htmlwidgets::saveWidget(as_widget(fig2), "fig2.html") salvare
      return(fig2)

    }

  }else if (aplicacion =='conmutaciones'){

      enter <- as.numeric(readline(prompt = "Por favor, teclee una tasa de interés: "))
      i <- enter
      ind <- 1:111
      Cx <- ((1+i)**(-ind)*dx)
      Dx <- (1+i)^(-age)*f1
      Nx <- c()

      for(h in ind){
        Nx <- append(Nx, sum(Dx[h:length(Dx)]))
}

      Mx <- c()
      for(j in ind){
        Mx <- append(Mx, sum(Cx[j:111]))
}

      Sx <- c()
      for(w in ind){
        Sx <- append(Sx, sum(Nx[w:110]))
}

      Rx <- c()
      for(u in ind){
        Rx <- append(Rx, sum(Mx[u:111]))
}
      con <- matrix(c(f1, Cx, Dx, Nx, Mx, Rx, Sx), ncol = 7)
      ros <- c('lx', 'Cx', 'Dx', 'Nx', 'Mx', 'Rx', 'Sx')
      colnames(con) <- ros
      print(a3);print(toupper(a4))
      return(con)
  }
      }
    options(warn = duepalle)
}else{
      print('Solo se permite un pais del G8.
            Por favor, vuelva a teclear su elecciòn.')
  }
}
```