Supplement of Adv. Geosci., 44, 9–13, 2017
http://www.adv-geosci.net/44/9/2017/
doi:10.5194/adgeo-44-9-2017-supplement

Advances in Geosciences
Open Access

Supplement of

Human-flood interactions in Rome over the past 150 years

Giuliano Di Baldassarre et al.

Correspondence to: Giuliano Di Baldassarre (giuliano.dibaldassarre@geo.uu.se)

Supplementary material: 1) Data and 2) Model code

1. Data

Year	*Flood peak levels (m)*
1805	16.42
1846	16.25
1870	17.22
1900	16.17
1915	16.08
1937	16.84

Year	*Floodplain Population*
1871	109 210
1901	173 629
1931	334 116
1951	473 802
1981	609 857
1991	582 819
2001	583 286
2014	567 959

2. Model code in R

```
library(deSolve)

################################ Setting up the functions ###########################
# modelEQ_new and Weventfun_new simulate the socio-hydrological model as in
# Di Baldassarre et al. (2015)

# modelEQ_new simulates the three differential equations as if
# there were no pertubations from external forces (e.g. floods)
modelEQ_new <- function (t, state, parameters) {
  with(as.list(c(state, parameters)),{
    dMdt <- -mu_S*M                             # Society
    dHdt <- -kappa_T*H                          # Technology
    dDdt <- rho_D*(1 - D*(1 + alpha_D*M))      # Demography
    # return the rate of change
    list(c(dDdt, dHdt, dMdt))
  })
}
# Weventfun_new simulate the action of flood events on the socio-hydrological system
Weventfun_new <- function(t, y, parms){
  with (as.list(y),{
    W <- ifelse(any(Wevent$tW == t), Wevent$W[Wevent$tW == t], 0)
    if (W + parms["xi_H"]*H > H) {
      L <- D*(1 - exp(-(W + parms["xi_H"]*H)/(parms["alpha_H"])))
      R <- parms["epsilon_T"]*(W + parms["xi_H"]*H - H)
      if (D > parms["delta_E"]) {
        D <- D - L
        H <- H + R
        M <- M + L
      } else {
        D <- D - L
        M <- M + L
      }
    }
    return(c(D, H, M))
```

```
  })
}
# Function to terminate the process if D goes to 0
gameoverfun_new <- function (t, y, parms) {
  with (as.list(y),{
    return(D)
  })
}

################################# Uploading the data ###############################
data <- read.csv("Rome_data.csv") # read the data
years <- data[,1]                  # years
Wr <- data[,2]                # event magnitude
h <- data[,3]                 # levee height
Pmax <- data[,4]              # max population density
Pmin <- data[,5]              # min population density
Pma <- cbind(Pmax,Pmin)
P <- apply(Pma,1,mean)        # mean population density

pP <- P[!is.na(P)]           # aggregate population density discarding Na values
yP <- years[P %in% pP]       # select years associated to non-Na population density values
hh <- h[!is.na(h)]           # aggregate levee heights discarding Na values
hh[1:2] <- c(NA,NA)          # discard level of the year 1800 equal to 16 m, assumed as a
                             # starting point of the analysis. Moreoever, the year 1880 has
                             # associated two levee height values, being the year where levees
                             # where built. In plotting the data, we consider the higher only.
Wevent <- data.frame(years[!is.na(Wr)],
                     Wr[!is.na(Wr)]) # aggregate flood event magnitude data discarding Na values
                                    # and associate the corresponding years
names(Wevent) <- c("tW","W")

############################### Performing the simulation ###########################
# Parameters
alpha_H = 30                 # Parameter related to flood depth-damage curve


Rho_D=c(0.008,abs(pP[2]-pP[1])/ (yP[2]-yP[1]),
        abs(pP[3]-pP[2])/ (yP[3]-yP[2]),
        abs(pP[4]-pP[3])/ (yP[4]-yP[3]),
        abs(pP[5]-pP[4])/ (yP[5]-yP[4]),
        abs(pP[6]-pP[5])/ (yP[6]-yP[5]),
        abs(pP[7]-pP[6])/ (yP[7]-yP[6]),
        abs(pP[8]-pP[7])/ (yP[8]-yP[7])) # relative population density values

rho <- mean(Rho_D)           # mean relative population density
xi_H=0                       # High water level enhancement due to presence of flood walls
alpha_D=20;                  # Ratio between preparedness and awareness
epsilon_T=1.1;               # Safety factor for flood walls
kappa_T=0                    # Protection level decay rate
mu_S=0.06                    # Memory loss rate (half-life)
delta_E = 0.13               # this value discriminates whether levees will be built or not
parms=c(rho_D=rho, alpha_D=alpha_D, delta_E=delta_E, epsilon_T=epsilon_T, kappa_T=kappa_T,
        mu_S=mu_S, alpha_H=alpha_H, xi_H=xi_H) # vector of parameters
#Initial conditions:
H0 <- 0                      # initial levee height
M0 <- 0                      # no awareness is assumed at the beginning
D0 = 0.001                   # initial population density
y=c(D=D0, H=H0, M=M0)

# Perform the simulation:
times <- seq(min(years),max(years), by=1) # timespan over which the analysis will be carried out
sim <- dede(y=y, t = times, func=modelEQ_new,parms=parms,
            events=list(func=Weventfun_new, time=Wevent$tW), rootfun=gameoverfun_new)
```