

Relational Event Models 2.0

Mixed effect additive REMs - Tutorial Section: Sheet

Boschi, Martina

2024-06-24

Mixed Effect Additive REMs: Application to Alien Species Invasions

0. Preparatory Steps: Introducing Alien Species Invasions

0.1. Installing libraries

In the following section, we will install the necessary packages for this tutorial. The provided code ensures that the packages are installed only if they are not already present on your system.

```
if (!require("mgcv", quietly = TRUE)) {  
  # If not installed, install it  
  install.packages("mgcv")  
  # Load the package  
  library("mgcv")  
} else {  
  if (!require("splines", quietly = TRUE)) {  
    install.packages("splines")  
    library("splines")  
  } else {  
    if (!require("ggplot2", quietly = TRUE)) {  
      install.packages("ggplot2")  
      library("ggplot2")  
    } else {  
      if (!require("tidyverse", quietly = TRUE)) {  
        install.packages("tidyverse")  
        library("tidyverse")  
      } else {  
        if (!require("RColorBrewer", quietly = TRUE)){  
          install.packages("RColorBrewer")  
        } else {  
          if (!require("mgcViz", quietly = TRUE)){  
            install.packages("mgcViz")  
          } else {  
            library("mgcv")  
            library("splines")  
            library("ggplot2")  
            library("tidyverse")  
            library("RColorBrewer")  
            library("mgcViz")  
          }  
        }  
      }  
    }  
  }  
}
```

```

    }
  }
}

```

During the tutorial, we will refer to the following color palettes to improve the visualization of the results.

```

pal.blue <- brewer.pal(9, "Blues")
pal.rose <- brewer.pal(9, "RdPu")
colors <- c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
            "#D55E00", "#CC79A7", "#999999", "#66C2A5", "#FC8D62")

```

0.2. Introducing alien species invasions



Demo 1:

D1.1: Load the input data "input00.RData".

```
load(file="01-Data/01-Inputs/input00.RData")
```

D1.2: Inspect the original data contained in the object FR. Have a look at the columns: "LifeForm", "Taxon", "Region", "FirstRecord", "Source".

```
head(FR[,c("LifeForm", "Taxon",
           "Region", "FirstRecord",
           "Source")])
```

##	LifeForm	Taxon	Region	FirstRecord
## 1	Algae	Acanthophora muscoides	Turkey	1986
## 2	Algae	Acanthophora nayadiformis	Cyprus	1997

```
## 3      Algae Acanthophora nayadiformis      Turkey      1970
## 4      Algae Acanthophora spicifera Hawaiian Islands      1952
## 5      Algae Acetabularia caliculus      Israel      1943
## 6      Algae Acetabularia caliculus      Spain      1957
##              Source
## 1      Cinar et al. (2005)
## 2              DAISIE
## 3      Cinar et al. (2005)
## 4 Carlton & Eldrege (2009)
## 5              DAISIE
## 6              DAISIE
```

D1.3: Interpret this data as a **relational event network**. Define **sender set**, **receiver set**, and **time-window**.

D1.4: Inspect the object `native`. A species' **native range** (NR) is the collection of areas where it is indigenous. How would you relate this concept to that of **risk set**?

```
native[sample(1:nrow(native), 10),c("species", "region")]
```

```
##              species      region
## 996157 Aonidomytilus albus      Cuba
## 1033011 Vespula vulgaris      China
## 1009074 Hylastes ater      Denmark
## 1010145 Lilioceris lillii      Slovakia
## 1021700 Pheidole megacephala      South Africa
## 998489 Bombus terrestris      Corse
## 1009158 Bostrichus ligniperda      Russia
## 995569 Anastrepha obliqua      Nicaragua
## 1001341 Chilo suppressalis      India
## 1001809 Cinara cupressi      United States
```

D1.5: Inspect the object `first_records`. Describe this relational event network and identify why it can be considered a subset of the previous one.

```
head(first_records[,c("year", "lf",
                      "species",
                      "region")])
```

```
##      year      lf      species      region
## 9337 1880 Insects Adelges nordmannianae      Switzerland
## 15937 1881 Insects Pheidole megacephala      France
## 16225 1884 Insects Pineus pini      New Zealand
## 12792 1886 Insects Eulachnus rileyi      United States
## 14019 1887 Insects Linepithema humile      Portugal
## 11797 1889 Insects Ctenarytaina eucalypti      New Zealand
```

D1.6: Read the commented function `invaded.regions` that allows computing which regions the species invaded before the current year. Which type of covariates does this function allow to compute: **exogenous**, **endogenous**, or **global**?

```
invaded.regions <- function(sp.n, r.n, y, native, first_records){
  # Convert input arguments to numeric type if not already
  sp.n <- as.numeric(sp.n)
  r.n <- as.numeric(r.n)
  y <- as.numeric(y)
```

```

# Get unique combinations of species number and region number from native data
t <- unique(as.vector(subset(native, sp.num == sp.n, r.num)))

# Get region numbers from first records data where species number matches
pr <- as.vector(subset(first_records, sp.num == sp.n, r.num))

# If the invasion is both present in first records data and in native range
# consider the former as actual piece of information
t <- setdiff(t, pr)

# Find indices of first records occurring before end date for the species
set.sp <- which(first_records$sp.num == sp.n & first_records$year < y)

# Combine regions in native range with regions in first records before date
t <- na.omit(c(t, first_records$r.num[set.sp]))

# Do not consider the involved region
inv <- unlist(setdiff(t, r.n))

# Return invaded regions
return(inv)
}

```

0.3. Building the Case-Control Dataset

Demo 2:

D2.1: Consider the smaller relational event network described by the data in `first_records`. Begin by defining the `sender` set and storing it in `spec`, followed by defining the `receiver` set and storing it in `reg.lf`.

```

# Define sender set - species
spec <- unique(first_records$species)
(s <- length(spec))

```

```
## [1] 114
```

```

# Define receiver set - regions
reg.lf <- unique(first_records$region)
(r <- length(reg.lf))

```

```
## [1] 159
```

The dynamic network of FRs consists of two sets of nodes: the set of species and the set of regions. Additionally, in the dataset, we refer to a numerical formulation of species and regions, which correspond to the set of unique observed species in the data and the set of regions reported in the matrix of geographical distances, respectively.

```
reg <- colnames(data_distance)
```

D2.2: Review the commented function `creating_case_control_dataset`, which utilizes information from the relational event network, noting that the relational event is non-recurrent.

```

creating_case_control_dataset <- function(first_records,
                                          spec, reg.lf, reg,
                                          seed=1234
){

```

```

reg.lf.num <- match(reg.lf, reg)
s <- length(spec)
r <- length(reg.lf)

## POSSIBLE (S,R) INTERACTIONS ####
# Initialize a vector to keep track of the risk set size over time
at.risk <- NULL
# Create a matrix to indicate possible (species, region) interactions
alien.occ <- matrix(0, nrow = s, ncol = r)
rownames(alien.occ) <- spec
colnames(alien.occ) <- reg.lf
for (n.sp in 1:s){
  # Identify native regions for each species
  nat.id <- unique(native$r.num[native$sp.num == n.sp])
  # Identify regions where the species is not native
  possible.to <- setdiff(reg.lf.num, nat.id)
  # Mark these regions as possible invasion sites for the species
  alien.occ[n.sp, reg[possible.to]] <- 1
}

## COLLECTING INFORMATION ####
dat.gam <- data.frame(matrix(NA, nrow=nrow(first_records), ncol=6))
colnames(dat.gam) <- c("y", "year",
                      "sp1", "r1",
                      "sp2", "r2")
# The response is fixed and equal to 1
dat.gam[,1] <- rep(1, nrow(first_records))

set.seed(seed)
# For each FR:
for (i in 1:nrow(first_records)){

  ### INFORMATION CONCERNING THE EVENT ###
  # year of the invasion event
  dat.gam[i,2] <- year <- first_records[i,"year"]
  # invading species
  dat.gam[i,3] <- s.ev <- first_records[i,"species"]
  # invaded country
  dat.gam[i,4] <- r.ev <- first_records[i,"region"]

  ### POSSIBLE EVENTS ###
  # Events occurred at the same time of the considered event
  # are removed from the risk set
  sub_stp <- first_records[first_records$year==year,
                           c("species", "region")]
  ni <- nrow(sub_stp)
  for (j in 1:ni){
    # Mark these (species, region) pairs as not at risk
    alien.occ[sub_stp[j,1], sub_stp[j,2]] <- 0
  }
  at.risk <- c(at.risk, sum(alien.occ==1))

  ### SAMPLING THE NON-EVENT ###

```

```

sr.nv<-sample(which(alien.occ!=0),1)
# species non-event
dat.gam[i,5] <- s.nv <- spec[(sr.nv-1)%/s+1]
# region non-event
dat.gam[i,6] <- r.nv <- reg.lf[(sr.nv-1)%/s+1]
}

return(dat.gam)
}

```

D2.3: Use the function `creating_case_control_dataset` with `first_records` and generate the case-control dataset. Additionally, determine the numerical codification of species and regions.

```

dat.gam <- creating_case_control_dataset(first_records,
                                         spec, reg.lf, reg)

```

```

dat.gam$sp1.num <- match(dat.gam$sp1, spec)
dat.gam$r1.num <- match(dat.gam$r1, reg)
dat.gam$sp2.num <- match(dat.gam$sp2, spec)
dat.gam$r2.num <- match(dat.gam$r2, reg)

```

```

head(dat.gam[c("year", "sp1", "r1", "sp2", "r2")])

```

##	year	sp1	r1	sp2
## 1	1880	Adelges nordmannianae	Switzerland	Epiphyas postvittana
## 2	1881	Pheidole megacephala	France	Chilo suppressalis
## 3	1884	Pineus pini	New Zealand	Coccinella septempunctata
## 4	1886	Eulachnus rileyi	United States	Agrilus planipennis
## 5	1887	Linepithema humile	Portugal	Tremex fuscicornis
## 6	1889	Ctenarytaina eucalypti	New Zealand	Xyleborus glabratus
##		r2		
## 1		Venezuela		
## 2		Benin		
## 3		Gambia		
## 4		Benin		
## 5		Guinea-Bissau		
## 6		Hawaiian Islands		

```

head(dat.gam[c("year", "sp1.num", "r1.num", "sp2.num", "r2.num")])

```

##	year	sp1.num	r1.num	sp2.num	r2.num
## 1	1880	1	90	22	77
## 2	1881	2	11	34	195
## 3	1884	3	31	66	208
## 4	1886	4	37	106	195
## 5	1887	5	25	103	211
## 6	1889	6	31	108	3

1. Time-Varying Effect of Trade on Species Invasions



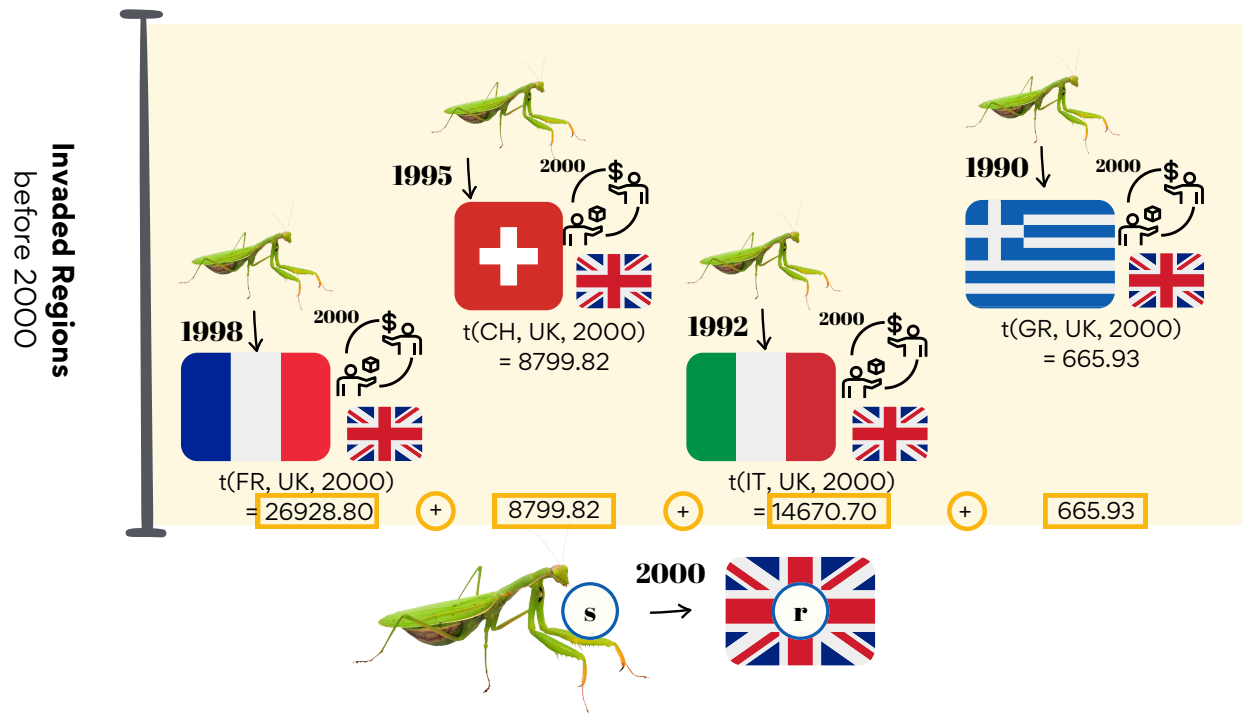
1.1. Trade: Covariate Computation

Demo 3 :

D3.1 : Load the input data "input01.RData".

```
load("01-Data/01-Inputs/input01.RData")
```

D3.2 : Write a function `log_trade` that, given a relational event (and all the relevant information required), computes the annual trade between the involved region and the previously invaded regions by the species. Apply the function to the events and the sampled non-events, and compute the difference. Refer to the figure in the following slide to understand how the covariate is computed.



```
t <- which(data_trade$transfer < 0)
data_trade$transfer[t] <- 0
```

```
trade.funct <- function(inv, r.n, y, reg, data_trade){

  # Convert input arguments to numeric type if not already
  r.n <- as.numeric(r.n)
  y <- as.numeric(y)

  # Check if there are already invaded countries
  if(length(inv)!=0){

    # Find rows that involve invaded countries as sending trade
    u <- which(data_trade$FromRegion %in% reg[inv])
    # Find rows that involve region of interest as receiving trade
    v <- which(data_trade$ToRegion == reg[r.n])
    # Find the intersection of the two sets
    w <- intersect(u,v)
    x <- data_trade[w,]
    # Consider the trade instances occurred before or at the time of interest
    x <- x[x$year<=y,]
    trade_value <- NULL
    # If there are rows in the filtered dataset
    if(nrow(x)>0){
      # For each invaded country, the maximum year is recorded
      o <- aggregate(x$year, list(x$FromRegion), FUN=max)
      # For each of them, the corresponding transfer is stored
      for (o.i in 1:nrow(o)){
        trade_value <- c(trade_value,
                          x$transfer[x$FromRegion==o[o.i,1] &
                                    x$year==o[o.i,2]])}
    }
  }
}
```



```

    }
  } else {
    # If there are not already invaded countries, trade is set equal to 0
    trade_value <- 0
  }
  # Compute the log-transformed sum of trade values (with an added constant 1)
  log_trade.value <- ifelse(length(trade_value)>0,
                           log(sum(trade_value, na.rm =T)+1),0)

  # Return the computed log-transformed trade value
  return(log_trade.value)
}

```

```

log_trade <- function(sp.n, r.n, y, native, first_records, reg, data_trade){

  inv <- invaded.regions(sp.n = sp.n,
                        r.n = r.n,
                        y = y,
                        native = native,
                        first_records = first_records)

  log_trade.value <- ifelse(r.n==match("USACanada", reg),
                           mean(trade.funct(inv = inv,
                                             r.n = match("United States",reg),
                                             y = y,
                                             reg = reg,
                                             data_trade = data_trade),
                               trade.funct(inv = inv,
                                             r.n = match("Canada",reg),
                                             y = y,
                                             reg = reg,
                                             data_trade = data_trade)),
                           trade.funct(inv = inv,
                                         r.n = r.n,
                                         y = y,
                                         reg = reg,
                                         data_trade = data_trade))

  return(log_trade.value)
}

```

```

dat.gam$tr1 <- apply(dat.gam[,c("sp1.num", "r1.num", "year")], 1,
                    function(x) log_trade(sp.n = x[1],
                                           r.n = x[2],
                                           y = x[3],
                                           native = native,
                                           first_records =
                                             first_records,
                                           reg = reg,
                                           data_trade = data_trade))

dat.gam$tr2 <- apply(dat.gam[,c("sp2.num", "r2.num", "year")], 1,
                    function(x) log_trade(x[1], x[2], x[3],
                                           native = native,
                                           first_records =

```

```

                                first_records,
                                reg = reg,
                                data_trade = data_trade))

dat.gam$str = dat.gam$str1 - dat.gam$str2

```

D3.3: Inspect the nature of the covariate. Is it a monadic or dyadic covariate?

D3.4: Is it an endogenous or exogenous covariate?

1.2. Effect of Trade as Spline function of Time

D4.0 Choose the basis-dimension q .

```
q = 10
```

D4.1 Select the spline basis functions types

```

bspline <- function(x, k, i, m = 2) {
  # ith B-spline basis function of order m at the values in x
  # given knot locations in k

  if (m == -1) {
    # Base case of the recursion:
    # when m is -1, we are at the lowest order basis function
    res <- as.numeric(x < k[i + 1] & x >= k[i])
    # Returns 1 if x is within the interval [k[i], k[i+1]]
  } else {

    # Recursive case:
    # B-spline basis function from lower order basis functions
    # Calculate the first term's coefficient
    z0 <- (x - k[i]) / (k[i + m + 1] - k[i])
    # Calculate the second term's coefficient
    z1 <- (k[i + m + 2] - x) / (k[i + m + 2] - k[i + 1])

    # Recursive calls to the lower order basis functions
    res <- z0 * bspline(x, k, i, m - 1) + z1 * bspline(x, k, i + 1, m - 1)
  }
  return(res) # Return the evaluated B-spline basis function
}

```

D4.2 Given range of the variable, select the basis evaluation points.

```

# Equally spaced knots from 1880 to 2005
n_knots = 14.
knots = seq(from = min(dat.gam$year),
            to = max(dat.gam$year),
            length.out = n_knots)

```

D4.3 Evaluate the q basis functions of time $b_l(t)$.

```

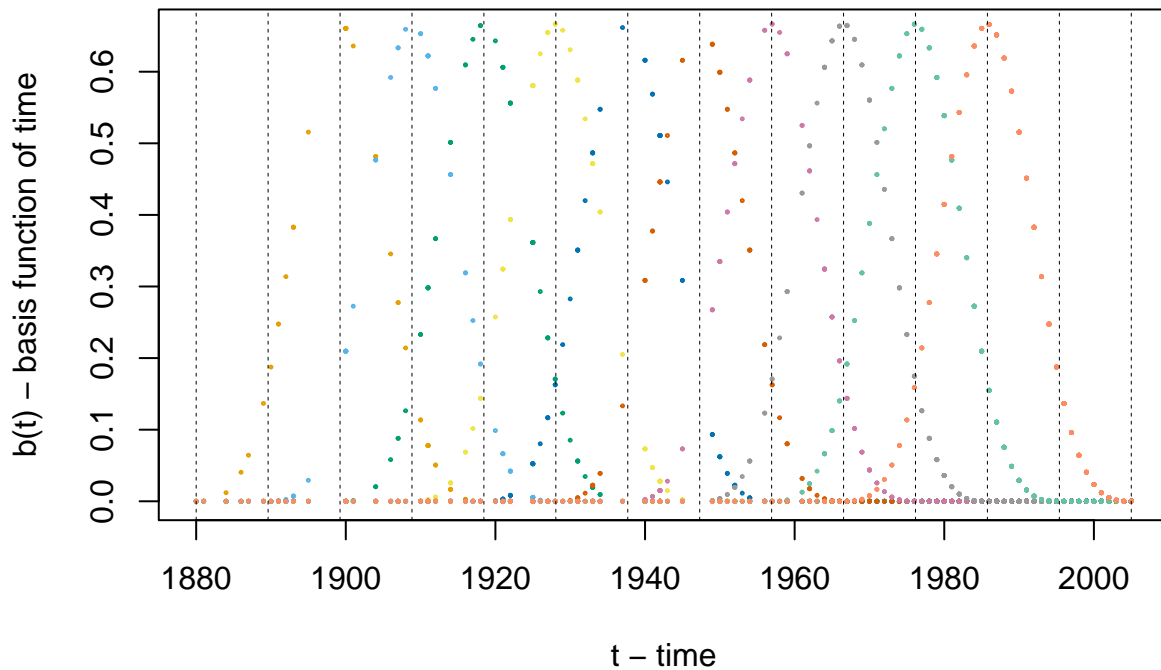
m = 2
basis = matrix(0, nrow = length(dat.gam$year), ncol = q)
for (j in 1:q) {
  basis[, j] = bspline(dat.gam$year, k = knots, i = j)
}

```

```

plot(y = basis[, 1],
     x = dat.gam$year,
     ylab = 'b(t) - basis function of time', xlab = 't - time',
     col = 0,
     cex = 0.2)
for (j in 1:10) {
  points(y = basis[, j],
        x = dat.gam$year,
        cex = 0.2,
        col = colors[j])
}
for (k in knots) {
  abline(v=k, lty=2, lwd=0.5)
}

```



1.3. Determine Trade's Contribution

- To the **log-hazard of an interaction**}

$$f_{sr} = \dots + \sum_{j=1}^q \theta_j b_j(t) tr_{sr}(t) + \dots$$

- To the **sampled likelihood function**}

$$PL(\theta) = \prod_{i=1}^n \frac{e^{\left[\sum_{j=1}^q \theta_j b_j(t)\right] \Delta tr_i}}{1 + e^{\left[\sum_{j=1}^q \theta_j b_j(t)\right] \Delta tr_i}}$$

1.4. Estimate the Coefficients of the Spline

Demo 5:

D5.1 Let $\mathbf{x.ev}$ and $\mathbf{x.nv}$ be $n \times 1$ vectors of covariate evaluated for events & non-events. Then: $\mathbf{x} = \mathbf{x.ev} - \mathbf{x.nv}$;

```
x.ev <- dat.gam$tr1
x.nv <- dat.gam$tr2
x <- x.ev - x.nv
```

D5.2 Let \mathbf{stp} be $n \times 1$ vector of event-times.

```
stp <- dat.gam$year
```

D5.3 Fit the model incorporating a time-varying effect for \mathbf{x} .

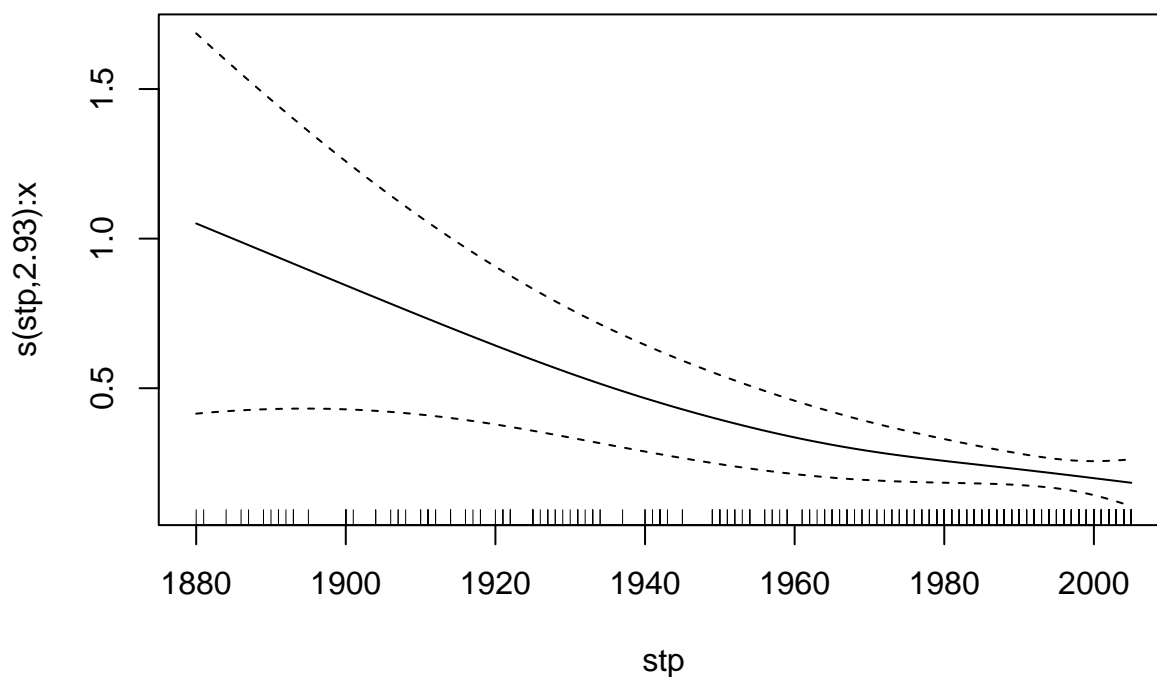
REMARK: default spline's type for `gam` in `mgcv` consists of *thin plate regression spline*.

```
gam_tr.only <- gam(y ~ s(stp, by=x) - 1,
  family="binomial"(link = 'logit'),
  method="REML", data=dat.gam)
```

1.5. Interpreting Trade's Effect

Demo 6

```
plot(gam_tr.only)
```

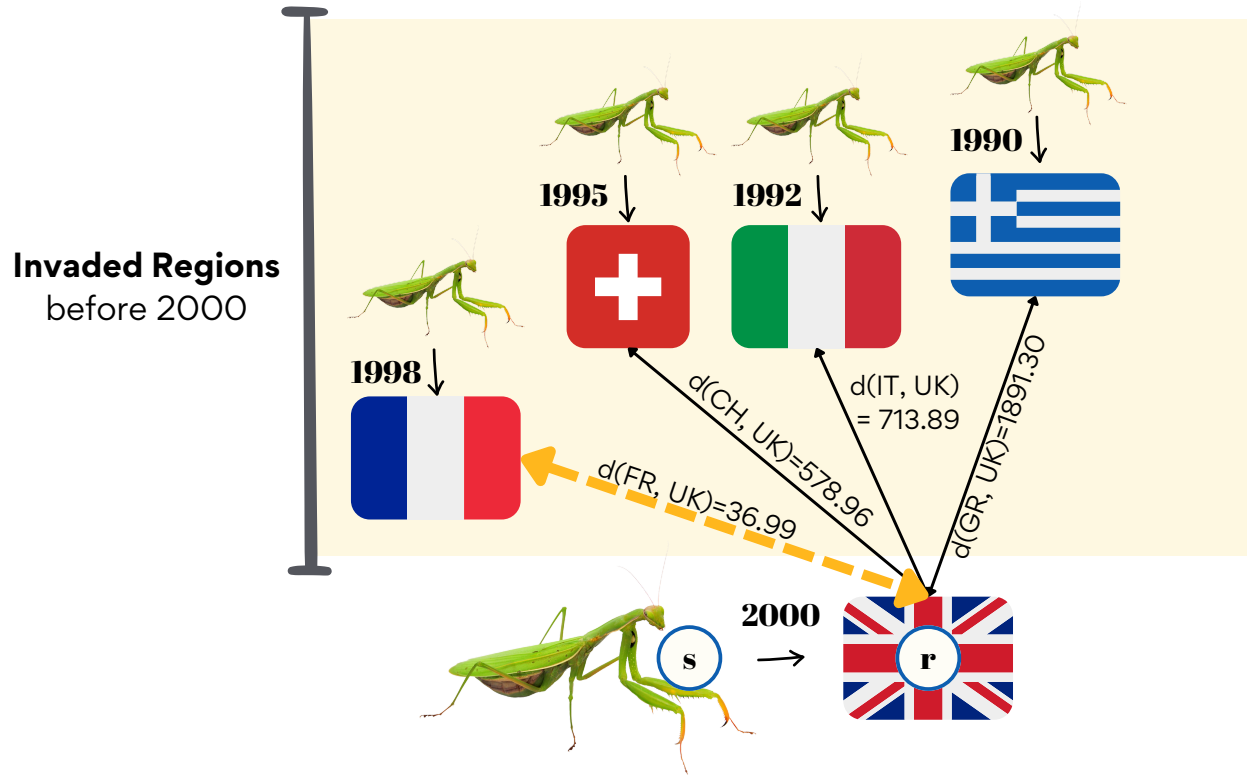


2. Non-Linear Effect of Distance on Species Invasions



2.1. Distance: Covariate Computation

Supplementary 2



2.2. Effect: Spline function of Distance

Supplementary 3

2.3. Determine Distance's Contribution

- To the **log-hazard of an interaction**

$$f_{sr} = \dots + \sum_{j=1}^q \theta_j b_j [d_{sr}(t)] + \dots$$

- To the **sampled likelihood function**

$$PL(\theta) = \prod_{i=1}^n \frac{e^{\left[\sum_{j=1}^q \theta_j b_j [d_{s_i r_i}(t_i)] - b_j [d_{s_i^* r_i^*}(t_i)] \right]}}{1 + e^{\left[\sum_{j=1}^q \theta_j b_j [d_{s_i r_i}(t_i)] - b_j [d_{s_i^* r_i^*}(t_i)] \right]}}$$

2.4. Estimate the Coefficients of the Spline

Exercise 1 :

E1.1 : Load the input data "input02.RData".

E1.2 : Let **x.ev** and **x.nv** be $n \times 1$ vectors of covariate evaluated for events & non-events.

E1.2 : Let **unit** be $n \times 1$ unit vector.

E1.3 : Define **X**, combining **x.ev** and **x.nv** and **I** combining **unit** and **-unit**.

E1.4 Fit the model incorporating a non-linear effect.

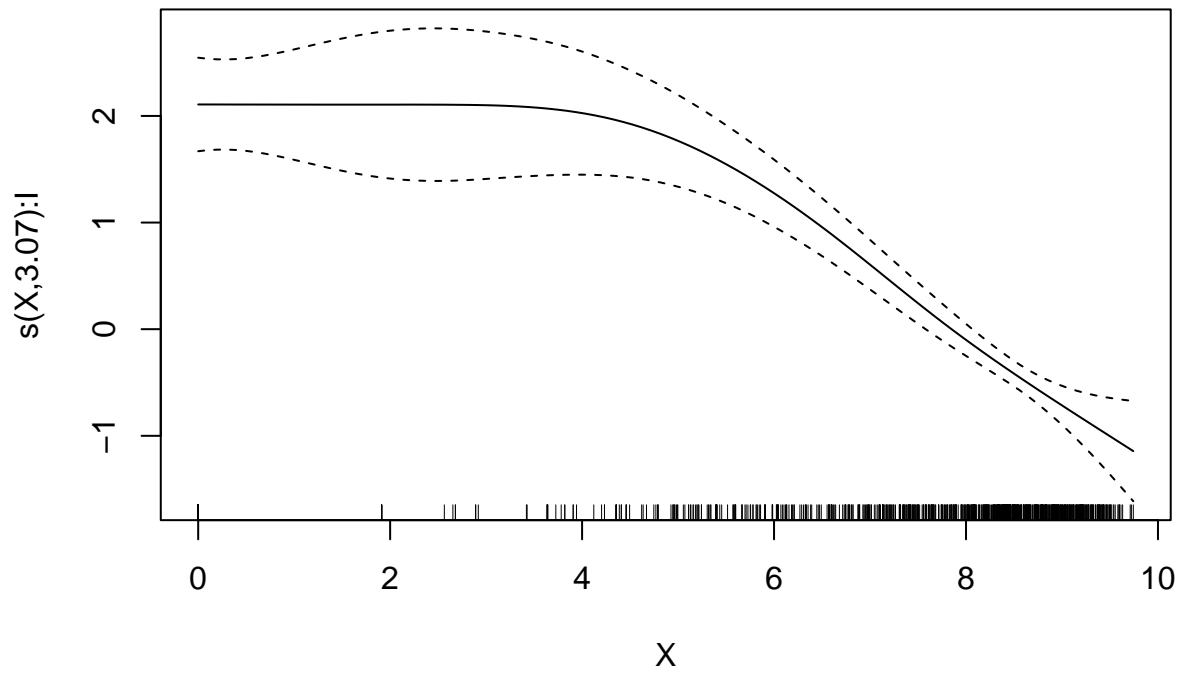
REMARK: default spline's type for **gam** in **mgcv** consists of **thin plate regression spline**.

2.5. Interpreting Distance's Effect

Demo 7

```
load(file="01-Data/02-Gam-Fits/gam_d.only.RData")
```

```
plot(gam_d.only)
```



3. Species Invasiveness Random Effect



3.1. Species Invasiveness: Covariate Computation

Demo 8:

D8.1: Load the input data "input03.RData".

```
load("01-Data/01-Inputs/input03.RData")
```

D8.2: Define the covariate: $z_s(t)$: [involvement of species \$s\$](#) in first record occurring at t . Create a $n \times 2$ matrix `sp` containing in the first column `spec` levels related to species involved in the events and in the second those involved in the non-events.

```
sp1 <- dat.gam$sp1
sp2 <- dat.gam$sp2
sp <- factor(c(sp1, sp2))
dim(sp) <- c(length(sp1), 2)
```

D7.3: Inspect the nature of the covariate. Is it a [monadic](#) or [dyadic](#) covariate? It is just reporting the label of species involved s , so the covariate $z_s(t)$ is monadic.

3.2. Random Effects as Smooths

$$\gamma' z_s(t) = \sum_{s' \in S} \gamma_{s'} 1\{s = s'\}$$

3.3. Species Invasiveness' Contribution

- To the [log-hazard of an interaction](#)

$$f_{sr} = \dots + \sum_{s' \in S} \gamma_{s'} 1\{s = s'\} + \dots$$

- To the **sampled likelihood function**

$$PL(\gamma) = \prod_{i=1}^n \frac{e^{\gamma_{s_i} - \gamma_{s_i}^*}}{1 + e^{\gamma_{s_i} - \gamma_{s_i}^*}} - \frac{1}{2\sigma_{sp}^2} \gamma^T \gamma$$

3.4. Estimate the Coefficients of the Spline

Exercise 3 :

E3.1 : Let `unit` be $n \times 1$ unit vector. Define `I` combining `unit` and `-unit`.

E3.2 Fit the model incorporating a random effect.

REMARK: default spline's type for `gam` in `mgcv` consists of **thin plate regression spline**. Instead, we need to specify that we aim to fit a random effect.

3.5. Interpreting Species Invasiveness Random Effects

Demo 8 :

```
load(file="01-Data/02-Gam-Fits/gam_sp.only.RData")
```

```
re.species <- coefficients(gam_sp.only)
names(re.species) <- levels(sp)
```

```
sort(re.species, decreasing = TRUE)[1:5]
```

```
## Frankliniella occidentalis      Aphis spiraeicola
##              2.411066              1.848464
##      Cinara cupressi      Phenacoccus manihoti
##              1.789825              1.749139
##      Apis mellifera
##              1.571748
```

```
sort(re.species)[1:5]
```

```
## Xylosandrus compactus      Solenopsis richteri      Scolytus schevyrewi
##              -1.5031237              -1.2676923              -1.0597829
## Archips fuscocupreanus      Agrilus planipennis
##              -1.0562797              -0.9503508
```

4. Fit a model with all the components included

Exercise 4 :

E4.1 : Load the input data **"input04.RData"**.

```
load("01-Data/01-Inputs/input04.RData")
```

```
unit <- rep(1, nrow(dat.gam))
stp = dat.gam$year
X = cbind(dat.gam$d1, dat.gam$d2)
I = cbind(unit, -unit)
```

```
gam_complete <- gam(y ~ dt +
                    s(stp, by=tr) +
                    s(X, by=I) +
                    s(sp, by=I, bs="re") - 1,
                    family="binomial"(link = 'logit'),
                    method="REML", data=dat.gam)
```

```
load(file="01-Data/02-Gam-Fits/gam_dt.only.RData")
load(file="01-Data/02-Gam-Fits/gam_tr.only.RData")
load(file="01-Data/02-Gam-Fits/gam_d.only.RData")
load(file="01-Data/02-Gam-Fits/gam_sp.only.RData")
```

```
AIC(gam_dt.only)
```

Model Selection

```
## [1] 742.7089
```

```
AIC(gam_tr.only)
```

```
## [1] 630.9921
```

```
AIC(gam_d.only)
```

```
## [1] 550.0876
```

```
AIC(gam_sp.only)
```

```
## [1] 642.3556
```

```
AIC(gam_complete)
```

```
## [1] 436.6027
```

```
sort(re.species, decreasing = TRUE)[1:5]
```

Changes in random effect predictions

```
## Frankliniella occidentalis      Aphis spiraeicola
##                2.411066                1.848464
##          Cinara cupressi      Phenacoccus manihoti
##                1.789825                1.749139
##          Apis mellifera
##                1.571748
```

```
sort(re.species)[1:5]
```

```
## Xylosandrus compactus      Solenopsis richteri      Scolytus schevyrewi
##                -1.5031237                -1.2676923                -1.0597829
## Archips fuscocupreanus      Agrilus planipennis
##                -1.0562797                -0.9503508
```

```
re.species_complete <- coefficients(gam_complete)[21:length(coefficients(gam_complete))]
names(re.species_complete) <- levels(sp)
sort(re.species_complete, decreasing = TRUE)[1:5]
```



```
##      Bactrocera invadens      Phenacoccus manihoti Maconellicoccus hirsutus
##              1.2972472              1.1558752              0.9404922
##      Aphis spiraecola      Wasmannia auropunctata
##              0.8913869              0.8754983
```

```
sort(re.species_complete)[1:5]
```

```
##      Phloeomyzus passerinii      Curculio conicus      Xylosandrus compactus
##              -1.0033272              -0.9886417              -0.8874932
##      Xylosandrus crassiusculus      Bostrichus ligniperda
##              -0.8056282              -0.7592472
```

```
sort(re.species)[1:3]
```

```
##      Xylosandrus compactus      Solenopsis richteri      Scolytus schevyrewi
##              -1.503124              -1.267692              -1.059783
```

```
sort(re.species_complete)[1:3]
```

```
##      Phloeomyzus passerinii      Curculio conicus      Xylosandrus compactus
##              -1.0033272              -0.9886417              -0.8874932
```