eAppendix 1. R code to find SE and CI for the MMT of the temperature-mortality spline.

eAppendix 2. Temperature-mortality associations for the 52 provincial capital cities in Spain (with 95% CI shaded grey). Dashed vertical lines are unconstrained minimum mortality temperatures (MMT). Average daily mortality count is indicated in parentheses after the city name. RR=relative risk.

eAppendix 3. Sensitivity analysis for the cities where unconstrained minimum mortality temperatures (MMT) are at or close to one of the imprecisely estimated tails of the curve. RR=relative risk. I2=index of heterogeneity.

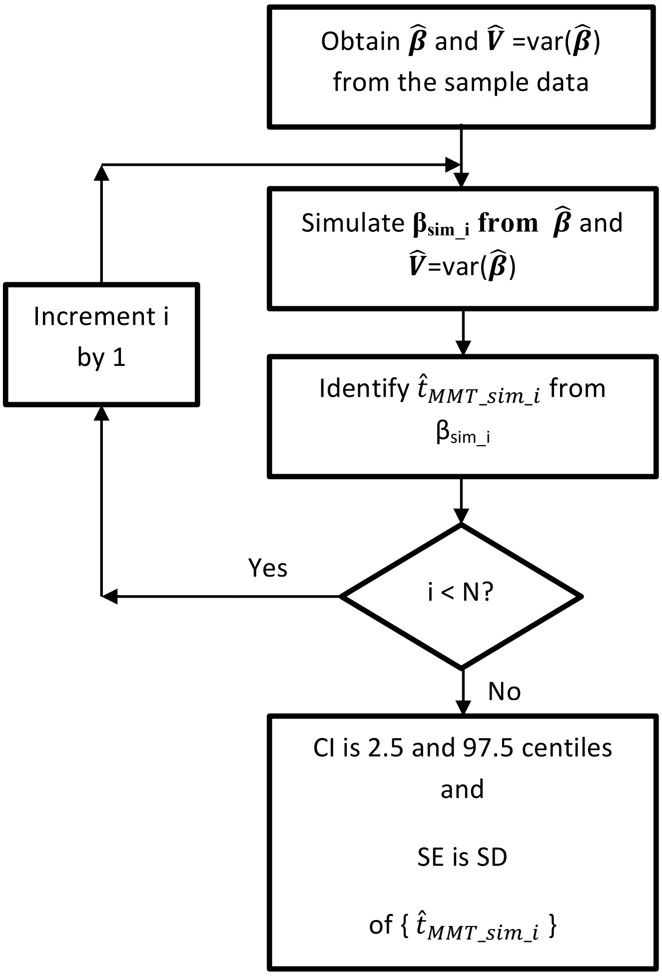
eAppendix 4. Sensitivity analysis for DLNM choice of knots and lag period. I2=index of heterogeneity. Res. I2=residual heterogeneity.

Algorithm to find SE and CI for the minimum mortality temperature of the temperature-mortality spline.

Suppose we have a parameter vector $\boldsymbol{\beta}$ of a function of **t** (typically a spline of temperature, but the algorithm proposed is more general):

$$y = a + f(t, \boldsymbol{\theta}) \quad (1)$$

Denote the value of t at which y is minimum as t_{MMT}, and the minimum y as y_{MMT}. Using the ^ notation to denote a sample estimate of a parameter, we know $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{V}}$=var($\widehat{\boldsymbol{\beta}}$). We seek a CI and SE for $\hat{t}_{MMT}$.



CIs and SEs obtained from the above algorithm may be termed an approximate parametric bootstrap. ***Parametric*** because we generate resampled data from parameters rather than actually resampling data, and ***approximate*** because we generate resampled values of the estimated spline rather than of the data themselves.

```
####################################################################################
# FUNCTION TO ESTIMATE MINIMUM OF A EXPOSURE-RESPONSE FUNCTION FROM A FITTED MODEL
####################################################################################
#
# DISCLAIMER:
#   THE CODE COMPOSING THIS FUNCTION HAS NOT BEEN SYSTEMATICALLY TESTED. THE
#   PRESENCE OF BUGS CANNOT BE RULED OUT. ALSO, ALTHOUGH WRITTEN GENERICALLY
#   FOR WORKING IN DIFFERENT SCENARIOS AND DATA, THE FUNCTION HAS NOT BEEN
#   TESTED IN CONTEXTS DIFFERENT THAN THE EXAMPLE INCLUDED IN THE PAPER.
#   IT IS RESPONSIBILITY OF THE USER TO CHECK THE RELIABILITY OF THE RESULTS IN
#   DIFFERENT APPLICATIONS.
#
# 19 FEB 2016
####################################################################################

findmin <- function(basis,model=NULL,coef=NULL,vcov=NULL,at=NULL,from=NULL,
                    to=NULL,by=NULL,sim=FALSE,nsimboot=10000) {

  ####################################################################################
  #   ARGUMENTS:
  #   - basis: A SPLINE OR OTHER BASIS FOR AN EXPOSURE x CREATED BY DLNM FUNCTION
  #            CROSSBASIS OR ONEBASIS
  #   - model: THE FITTED MODEL
  #   - coef AND vcov: COEF AND VCOV FOR basis IF model IS NOT PROVIDED
  #
  #   - at: A NUMERIC VECTOR OF x VALUES OVER WHICH THE MINIMUM IS SOUGHT
  #   OR
  #   - from, to: RANGE OF x VALUES OVER WHICH THE MINIMUM IS SOUGHT.
  #   - by: INCREMENT OF THE SEQUENCES x VALUES OVER WHICH THE MINIMUM IS SOUGHT
  #
  #   - sim: IF BOOTSTRAP SIMULATION SAMPLES SHOULD BE RETURNED
  #   - nsimboot: NUMBER OF SIMULATION SAMPLES
  ####################################################################################

  ####################################################################################
  # CREATE THE BASIS AND EXTRACT COEF-VCOV
  #
  # CHECK AND DEFINE BASIS
  if(!any(class(basis)%in%c("crossbasis","onebasis")))
    stop("the first argument must be an object of class 'crossbasis' or 'onebasis'")
  #
  # INFO
  one <- any(class(basis)%in%c("onebasis"))
  attr <- attributes(basis)
  #
  # PREDICTION VALUES
  if(is.null(at)) {
    if(is.null(from)) from <- attr$range[1]
    if(is.null(to)) to <- attr$range[2]
    if(is.null(by)) by <- 0.1
    nobs <- max(1,diff(attr$range)/by)
    pretty <- pretty(c(from,to),n=nobs)
    predvar <- pretty[pretty>=from&pretty<=to]
  } else predvar <- sort(unique(at))
  #
  # CREATE THE BASIS FOR PREDICTION
  basisnew <- if(!one) {
    Q <- matrix(rep(predvar,diff(attr$lag)+1),length(predvar))
    do.call(crossbasis,list(x=Q,lag=attr$lag,argvar=attr(basis,"argvar"),
                            arglag=attr(basis,"arglag")))
  } else do.call(onebasis,c(list(x=predvar,fun=(attr$fun)),
                            attr[match(names(formals(attr$fun)),names(attr),nomatch=0)]))
  #
  # EXTRACT COEF-VCOV
  name <- deparse(substitute(basis))
  cond <- if(one) paste(name,"[[:print:]]*b[0-9]{1,2}",sep="") else
    paste(name,"[[:print:]]*v[0-9]{1,2}\\.l[0-9]{1,2}",sep="")
  if(ncol(basis)==1L) cond <- name
  if(!is.null(model)) {
    model.class <- class(model)
    coef <- dlnm:::getcoef(model,model.class,cond)
    vcov <- dlnm:::getvcov(model,model.class,cond)
  }
  #
  # CHECK COEF AND VCOV
  if(length(coef)!=ncol(basis) || length(coef)!=dim(vcov)[1] ||
       any(is.na(coef))|| any(is.na(vcov))) {
    stop("number of estimated parameters does not match number of basis
         Possible reasons:
         1) 'model' and 'basis' objects do not match
```

```
        2) wrong 'coef' or 'vcov' arguments or methods
        3) model dropped some basis variables because of collinearity (set to NA)
        4) name of basis matrix matches other parameters in the model formula
        Unlikely, but in this case change the name of basis")
  }

  ##################################################################################
  # FIND THE MINIMUM
  #
  pred <- drop(basisnew%*%coef)
  ind <- which.min(pred)
  min <- predvar[ind]

  ##################################################################################
  # APPROXIMATE PARAMETRIC BOOTSTRAP SIMULATIONS
  #
  if(sim) {
    # SIMULATE COEFFICIENTS
    k <- length(coef)
    eigen <- eigen(vcov)
    X <- matrix(rnorm(length(coef)*nsimboot),nsimboot)
    coefsim <- coef + eigen$vectors %*% diag(sqrt(eigen$values),k) %*% t(X)
    # COMPUTE MINIMUM
    minsim <- apply(coefsim,2,function(coefi) {
      pred <- drop(basisnew%*%coefi)
      ind <- which.min(pred)
      return(predvar[ind])
    })
  }

  ##################################################################################
  #
  res <- if(sim) minsim else min
  return(res)
  }

###################### END OF FINDMIN FUNCTION ###################################
```

```
######### EXAMPLES OF ESTIMATION OF THE MMT USING PUBLIC DOMAIN DATA ###########
#
# The required data file london.csv is in the public domain, and
# available from http://www.ag-myresearch.com/bmcmrm2014.html
#
################################################################################

library(dlnm) ; library(splines)
#source("findmin.R")

################################################################################
# FIRST EXAMPLE: SUBSET OF DATA, SIMPLE MODEL, UNIDIMENSIONAL EXPOSURE-RESPONSE

# LOAD A SUBSET OF THE DATA: LONDON JULY-DEC 2005
# (CONVENIENCE SAMPLE, LARGISH ASIMMETRIC CI)
data <- subset(read.csv("london.csv"), year==2005 & month>6)
head(data)

# A UNIDIMENSIONAL 4DF SPLINE
b1 <- onebasis(data$tmean,df=4)

# SIMPLE MODEL WITH NO CONTROL FOR CONFOUNDING
m1 <- glm(death~b1,family=quasipoisson,data)

# ESTIMATE MMT, WITH CI AND STANDARD ERROR
(min1 <- findmin(b1,m1))
(min1ci <- quantile(findmin(b1,m1,sim=T),c(2.5,97.5)/100))
(min1se <- sd(findmin(b1,m1,sim=T)))

# ESTIMATE THE MINIMUM WITHIN A SPECIFIED RANGE (15-16: MEANINGLESS ILLUSTRATION)
(min1b <- findmin(b1,m1,from=15,to=16))

# ALTERNATIVE BOOTSTRAP POINT ESTIMATE = MEAN OF BS SAMPLES
(min1alt <- mean(findmin(b1,m1,sim=T)))

# PLOT
plot(crosspred(b1,m1),ylab="RR",xlab="Temperature",xlim=c(0,25),
     ylim=c(0.9,1.3),lwd=1.5)
abline(v=min1)
abline(v=min1ci,lty=2)

################################################################################
# SECOND EXAMPLE: WHOLE DATA, FULL MODEL, BI-MENSIONAL EXPOSURE-LAG-RESPONSE

# LOAD DATA: LONDON 1993-2006
data <- read.csv("london.csv")
head(data)

# BI-DIMENSIONAL EXPOSURE-LAG-RESPONSE SPLINE
vk <- equalknots(data$tmean,fun="bs",df=4,degree=2)
lk <- logknots(25,3)
cb2 <- crossbasis(data$tmean, lag=25, argvar=list(fun="bs",degree=2,knots=vk),
                  arglag=list(knots=lk))

# FULL MODEL WITH CONTROL FOR CONFOUNDING
m2 <- glm(death~cb2+ns(time,10*14)+dow,family=quasipoisson(),data)

# ESTIMATE MMT, WITH CI AND STANDARD ERROR; ALSO ALTERNATIVE BS POINT ESTIMATE
(min2 <- findmin(cb2,m2))
(min2ci <- quantile(findmin(cb2,m2,sim=T),c(2.5,97.5)/100))
(min2se <- sd(findmin(cb2,m2,sim=T)))
(min2alt <-  mean(findmin(cb2,m2,sim=T)))

# IN PERCENTILE SCALE
sum(data$tmean<min2)/nrow(data)*100

# PLOT
cb2 <- crossbasis(data$tmean, lag=25, argvar=list(fun="bs",degree=2,
                                                  knots=vk,cen=min2), arglag=list(knots=lk))
plot(crosspred(cb2,m2),"overall",ylab="RR",xlab="Temperature",xlim=c(-5,35),
     ylim=c(0.5,3.5),lwd=1.5)
abline(v=min2)
abline(v=min2ci,lty=2)

#################### END OF EXAMPLES #########################################
```

```
################################################################################
# SIMULATION STUDY TO EVALUATE BIAS AND COVERAGE OF THE FUNCTION findmin()
#
# The required data file london.csv is in the public domain, and
# available from http://www.ag-myresearch.com/bmcmrm2014.html
#
################################################################################

library(dlnm) ; library(splines); library(MASS)
#source("findmin.R")

################################################################################
# DEFINE A KNOWN EXPOSURE-RESPONSE ASSOCIATION

# LOAD A SUBSET OF THE DATA: LONDON JULY-DEC 2005
# (CONVENIENCE SAMPLE, LARGISH ASYMMETRIC CI)
data <- subset(read.csv("london.csv"), year==2005 & month>6)
head(data)

# A UNIDIMENSIONAL 4DF SPLINE
b <- onebasis(data$tmean,df=4)

# SIMPLE MODEL, WITH PREDICTED OUTCOME
m <- glm(death~b,family=poisson,data)
deathpred <- predict(m,type="response")

# REAL MINIMUM
(min <- findmin(b,m))

################################################################################
# RUN SIMULATION

# NUMBER OF SIMULATIONS
nsim <- 1000  ## reduce for quick illustration of code

# GRID
summary(data$tmean)
at <- seq(0,24.3,by=0.1)

# CREATE THE OBJECT TO STORE THE INFO
res <- matrix(NA,nsim,8,dimnames=list(seq(nsim),

c("est_min","bias","covered","true_at_CI_edge","boundary","est_SE","altest_min",("bias_altest") )))

# RUN THE LOOP
set.seed(13041975)
for(i in seq(nsim)) {

  cat(i,"")

  # GENERATE SIMULATED DATA (CODE FOR NEGTIVE BIN FOR FLEXIBILITY; FOR POISSON USE PHI=1 )
  # DEFINE OVERDISPERSION PARAMETER

  phi <- 1       # SET TO 1 FOR POISSON. ABOVE 1 WILL GIVE NEG BIN, INFLATING MEAN DISPERSION BY PHI
  if(phi==1)  {
    deathsim <- rpois(length(deathpred),deathpred)
    } else {
    # DEFINE THETA (PHI = 1 + MU/THETA, SO THETA = MU/(PHI-1) )
    theta <- deathpred/(phi-1)
    # SIMULATE FROM A NEGATIVE BINOMIAL DISTRIBUTION
    deathsim <- rnegbin(length(deathpred),deathpred,theta)
  }

  # FIT THE MODEL
  msim <- glm(deathsim~b,family=quasipoisson,data)

  # FIND MIN, SAMPLE OF BOOTSTRAP ESTIMATES, AND CI AND SE FROM THOSE
  minsim   <- findmin(b,msim,at=at)
  minsimbs <- findmin(b,msim,,at=at,sim=T)
  mincisim <- quantile(minsimbs,c(2.5,97.5)/100)
  minsesim <- sd(minsimbs)
  minsimalt <- mean(minsimbs)

  # STORE THE DATA
  res[i,1] <- minsim
  res[i,2] <- min-minsim
  res[i,3] <- mincisim[1]<=min & mincisim[2]>=min
  res[i,4] <- mincisim[1]==min | mincisim[2]==min
  res[i,5] <- any(mincisim == range(at))
  res[i,6] <- minsesim
```

```
  res[i,7] <- minsimalt
  res[i,8] <- min-minsimalt
}

################################################################################
# ASSESS RESULTS

# BIAS
mean(res[,2])

# COVERAGE
mean(res[,3])*100

# % OF CIS WITH TRUE MIN AT EDGE} (DISCRETE SPACE ISSUE)
mean(res[,4])*100

# PERCENTAGE OF CI AT BOUNDARY OF X SPACE
mean(res[,5])*100

# TRUE SAMPLING SD OF ESTIMATED MINIMA (EMPIRICAL SE)
# AND DISTRIBUTION OF ESTIMATED SES
sd(res[,1])
quantile(res[,1], c(.025,.975))

# AND DISTRIBUTION OF ESTIMATED SES
summary(res[,6])
mean(res[,6])

# ALTERNATIVE BOOTSTRAP POINT ESTIMATES
  # BIAS
    mean(res[,8])
  # SAMPLING DISTRIBUTION
    sd(res[,7])
    quantile(res[,7], c(.025,.975))
#################### END OF SIMULATION ########################################
```

Table of performance for proposed bootstrap CI and alternative point estimates for minimum mortality temperature in above 1,000 simulations

Statistic	Model distribution: Poisson	Model distribution: Negative Binomial (phi=1.2)
Coverage (%) of 95% CIs (% of CIs including or touching true MMT)	96.4%	96.8%
of which, % with MMT on CI lower or upper limit	1.2%	0.9%
Mean SE (MMT) estimated by bootstrap	1.35 ºC	1.72 ºC
Median SE (MMT) estimated by bootstrap	0.92 ºC	1.23 ºC
Actual empirical *'true'* SE (SD of MMTs over simulations)	0.96 ºC	1.13 ºC
Sampling distribution of usual MMT estimates and of alternative bootstrap estimates		
Usual estimates		
Bias (difference from true MMT=16.1)	-0.05 ºC	-0.10 ºC
SD	0.96 ºC	1.13 ºC
95% fractile interval	15.1-17.5 ºC	15.0-17.6 ºC
Alternative bootstrap estimates		
Bias (difference from true MMT=16.1)	-0.07 ºC	-0.10 ºC
SD	0.83 ºC	1.06 ºC
95% fractile interval	14.6-17.5 ºC	13.9-17.9 ºC

MMT = minimum mortality temperature; *CI* = confidence interval; *SE* = standard error; *SD* = standard deviation.


Burgos (5.4)
-9.8 4.8 30.0
Leon (5.9)
-8.5 20.0 28.0
Soria (1.6)
-7.2 20.0 28.0
Avila (2.8)
-6.3 14.0 29.0
Vitoria (5.0)
-6.4 12.0 31.0
Palencia (3.4)
-6.7 30.0
Lugo (4.6)
-4.1 15.0 29.0
Teruel (1.6)
-11 28
Salamanca (5.9)
-5.1 14.0 29.0
Segovia (2.2)
-6.3 5.5 31.0
Valladolid (8.8)
-4.8 21.0 31.0
Pamplona (7.4)
-5.20 32.00
Zamora (2.7)
-4.3 23.0 31.0
Cuenca (2.2)
-4.4 30.0
Guadalajara (2.6)
-3.8 22.0 29.0
Oviedo (9.4)
-1.6 20.0 28.0
San Sebastian (8.6)
-1.7 12.0 30.0
Logroño (4.3)
-4.9 32.0
Huesca (2.0)
-5.5 32.0
Albacete (4.5)
-11 13 32
Santander (7.6)
1.1 16.0 28.0
Bilbao (10.7)
-0.4 21.0 32.0
Girona (4.0)
-2.3 23.0 30.0
Pontevedra (4.1)
2.4 20.0 30.0
A Coruña (9.9)
2.4 19.0 29.0
Ourense (5.1)
-1.3 12.0 32.0
Madrid (75.2)
-1.8 13.0 32.0
Lleida (4.9)
-7.6 14.0 31.0
Zaragoza (18.7)
-4.8 8.2 33.0
Granada (10.1)
-3.1 24.0 33.0
Ciudad Real (3.0)
-3.2 25.0 34.0
Toledo (4.0)
-2 25 34
Barcelona (47.7)
1.4 23.0 31.0
Caceres (2.9)
0.1 34.0
P. Mallorca (10.8)
1.8 15.0 32.0
Jaen (4.7)
-3.1 16.0 35.0
Badajoz (4.9)
1.6 24.0 34.0
Castellon (5.4)
3.4 26.0 32.0
Tarragona (3.4)
-0.1 27.0
Huelva (5.6)
3.8 20.0 36.0
Cordoba (9.3)
1 23 36
Alicante (6.7)
4 25 32
Valencia (27.9)
3.4 25.0 34.0
Cadiz (5.3)
3.9 23.0 33.0
Malaga (15.2)
4.4 21.0 34.0
Ceuta (1.3)
4.1 36.0
Melilla (1.1)
3.8 14.0 36.0
Murcia (10.1)
3.8 26.0 36.0
Almeria (5.4)
5.4 20.0 36.0
Sevilla (23.1)
2.7 20.0 37.0
Las Palmas (11.2)
14 23 33
Tenerife (6.8)
13 34
RR
0 1 2 3 4 5
Mean temperature

	No boundaries			1st-99th			5th-95th			10th-90th		
Cities modiffied	MMT	(95% CI)	Centile	MMT	(95% CI)	Centile	MMT	(95% CI)	Centile	MMT	(95% CI)	Centile
Teruel	-10.9	(-10.9 , 4.6)	0.0	-2.1	(-10.9 , 3.7)	1.0	1.8	(-10.9 , 6.3)	5.9	3.4	(-10.9 , 4.5)	11.1
Palencia	-6.7	(-6.7 , 19.9)	0.0	12.6	(-6.7 , 19.1)	55.9	12.6	(-6.7 , 19.7)	55.9	12.6	(-6.7 , 20.1)	55.9
Huesca	-5.5	(-5.5 , 32.0)	0.0	-0.5	(-5.5 , 32.0)	1.0	2.8	(-5.5 , 32.0)	5.0	4.8	(-5.5 , 32.0)	10.2
Logroño	-4.9	(-4.9 , 31.5)	0.0	0.6	(-4.9 , 31.5)	1.0	3.6	(-4.9 , 31.5)	5.1	5.6	(-4.9 , 31.5)	10.8
Cuenca	-4.4	(-4.4 , 23.5)	0.0	-0.1	(-4.4 , 25.0)	1.0	15.8	(-4.4 , 29.8)	63.0	15.8	(-4.4 , 29.8)	63.0
Pamplona	-0.4	(-5.2 , 31.6)	1.0				3.2	(-5.2 , 31.6)	6.0	4.9	(-5.2 , 31.6)	10.9
Tenerife	13.4	(13.4 , 25.2)	0.0	23.5	(13.4 , 25.3)	68.8	23.5	(13.4 , 25.2)	68.8	23.5	(13.4 , 25.0)	68.8
Oviedo	20.2	(12.5 , 28.4)	93.1							19.4	(12.0 , 28.4)	89.2
Murcia	25.7	(13.1 , 36.1)	90.2							25.5	(3.8 , 36.0)	89.2
Castellon	25.9	(12.5 , 32.0)	90.3							25.8	(12.9 , 32.0)	88.9
Caceres	34.1	(0.1 , 34.1)	100.0	20.0	(0.1 , 34.1)	67.0	20.0	(0.1 , 34.1)	67.0	20.0	(0.1 , 34.1)	67.0
Ceuta	36.1	(4.1 , 36.1)	100.0	19.4	(4.1 , 36.1)	55.8	19.4	(4.1 , 36.1)	55.8	19.4	(4.1 , 16.1)	55.8
Meta-analysis estimates												
Pooled mean	18.7			19.7			21.2			21.4		
(95% CI)	(16.6 , 20.7)			(18.1 , 21.4)			(20.0 , 22.4)			(20.3 , 22.5)		
I2	47%			24%			2%			0%		

Knots	Lag	Pooled mean*			Mean increase by 1ºC of mean temp.**		
		MMT	(95% CI)	I2	ΔMMT	(95% CI)	Res. I2
at 10, 75 and 90th centiles	14	18.8	(17.3 , 20.3)	29.0	1.01	(0.48 , 1.53)	12.3
	21	19.7	(18.1 , 21.4)	24.0	1.09	(0.56 , 1.63)	2.8
	28	22.9	(21.7 , 24.3)	9.9	1.08	(0.59 , 1.57)	0.0
at 25, 50 and 75th centiles	14	19.4	(18.3 , 20.6)	38.4	1.02	(0.73 , 1.32)	0.0
	21	20.2	(19.0 , 21.5)	28.0	1.10	(0.73 , 1.48)	0.0
	28	21.1	(19.8 , 22.2)	12.0	1.23	(0.75 , 1.71)	0.0

* Random effects meta-analysis estimates for city-specific MMT constrained to the 1st-99th centile range.

** Random effects meta-regression estimates for the association between city-specific MMT and mean annual temperature.