Supplementary Material

Dimitra-Kleio Kipourou[a*], Maja Pohar Perme[b], Bernard Rachet[a], Aurélien Belot[a]

[a] *Cancer Survival Group, Faculty of Epidemiology and Population Health,*
Department of Non-Communicable Disease Epidemiology,
London School of Hygiene & Tropical Medicine,
London, WC1E 7HT, UK.
[b] *Faculty of Medicine,*
Institute for Biostatistics and Medical Informatics,
University of Ljubljana,
Ljubljana, Slovenia

dimitra-kleio.kipourou@lshtm.ac.uk

We illustrate the method using a simulated dataset that includes information on vital status, covariables (age, sex, etc.), and the expected mortality rate coming from the lifetable from the general population. For the user to be able to apply the method, 4 steps are needed:

1. Prepare the data.

2. Compute the pseudo-observations using the non-parametric estimator for the crude probabilities and the number of life years lost for each cause.

3. Fit the models for each indicator and for each cause and derive the covariable effects.

4. Predict the population point estimates and confidence intervals.

The data used in this document are available at `https://github.com/pseudorel/supp_material.git`. The R-packages needed for the following calculations are `survival`, `relsurv` and `geepack`. Please note here that if you use the version 2.2.1 of `relsurv`, you need to update the `survival` package to version 2.42.6.

---

*To whom correspondence should be addressed.

```
# Change appropriately the folder path where simdatn2.RData

#setwd()


# Install the needed packages

reqPcks <- c("relsurv", "survival","geepack")


for(p in reqPcks){

if(!require(p, character.only=TRUE)) {

install.packages(p)

library(p, character.only = TRUE)}

}


packageVersion("relsurv")

## [1] '2.2.1'

packageVersion("survival")

## [1] '2.42.6'
```

1. STEP 1: PREPARE THE DATA

We start first by exploring our data (simdatn2). Data consist of 10 variables including the continuous

variables age at diagnosis (age), year at diagnosis (year) and, survival time (timesurv), all of them

expressed in years. agecr is defined as $\frac{(age-70)}{10}$ and yearcr as $\frac{(year-2002)}{10}$. There are also the binary

variables sex (sex, $\{1,2\}$) and vital status (vstat, $\{0,1\}$). cause is a variable denoting the cause of

death, which although is not used in the calculations, we will use it later to help us understand the pseudo-

observations. Lastly, `popmrate` corresponds to the expected mortality rate for a given individual, while

`expectedrates.RT` is a ratetable object showing the event rates for a given year, age, and sex.

```r
# Load data
load("data_pseudo_tutorial2.RData")


str(simdatn2)

## 'data.frame': 1000 obs. of  10 variables:
##  $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ sex     : num  1 2 2 2 2 1 1 1 1 1 ...
##  $ year    : num  2000 2000 2000 2000 2000 ...
##  $ age     : num  34.7 43 43.4 47.3 50.3 ...
##  $ agecr   : num  -3.53 -2.7 -2.66 -2.27 -1.97 ...
##  $ yearcr  : num  -0.172 -0.183 -0.179 -0.187 -0.187 ...
##  $ timesurv: num  0.548 0.572 0.31 0.602 0.409 ...
##  $ vstat   : num  1 1 1 1 0 1 1 0 1 0 ...
##  $ cause   : num  1 1 1 1 0 1 1 0 2 0 ...
##  $ popmrate: num  0.00113 0.00133 0.00133 0.00198 0.00265 ...

head(simdatn2)

##   id sex     year      age      agecr     yearcr  timesurv
## 1  1   1 2000.278 34.71650 -3.528350 -0.1722157 0.5479875
## 2  2   2 2000.172 43.04669 -2.695331 -0.1827627 0.5719410
## 3  3   2 2000.215 43.44366 -2.655634 -0.1785245 0.3095095
## 4  4   2 2000.127 47.34635 -2.265365 -0.1873047 0.6024045
## 5  5   2 2000.125 50.32838 -1.967162 -0.1874797 0.4091484
```

```
## 6  6   1 2000.340 55.86828 -1.413172 -0.1660094 0.6528223

##    vstat cause    popmrate

## 1     1    1 0.001128884

## 2     1    1 0.001331712

## 3     1    1 0.001331712

## 4     1    1 0.001979022

## 5     0    0 0.002654055

## 6     1    1 0.007801306
```

For the application, we need to have in our dataset (or create if needed) those variables with which we will link the data to the ratetable. We start with exploring our ratetable.

```
# Explore the ratetable
str(expectedrates.RT)

##  ratetable [1:100, 1:2, 1:35] 3.34e-05 2.87e-06 1.26e-06 9.00e-07 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:100] "0" "1" "2" "3" ...
##   ..$ : chr [1:2] "1" "2"
##   ..$ : chr [1:35] "1981" "1982" "1983" "1984" ...
##  - attr(*, "dimid")= chr [1:3] "AGE.RT" "SEX.RT" "YEAR.RT"
##  - attr(*, "factor")= num [1:3] 0 1 0
##  - attr(*, "cutpoints")=List of 3
##   ..$ : num [1:100] 0 365 730 1096 1461 ...
##   ..$ : NULL
##   ..$ : num [1:35] 7671 8036 8401 8766 9132 ...
```

Our ratetable, includes 3 variables:

- `AGE.RT`: age ranging from 0 to 99 years, expressed in days.

- `SEX.RT`: sex is binary with values 1,2.

- `YEAR.RT`: year is ranging from 1981 to 2015, expressed as the difference in days between 01/01/1960 and year of diagnosis.

Except for sex, none of these variables exist in our data, so we need to generate them. In addition, we create also another variable called `timesurvD`, which corresponds to the survival time expressed in days, as this is necessary for the `cmp.rel` command used in the next step.

```r
N <- nrow(simdatn2)


# Convert (1) age at diagnosis, (2) time from 01-01-1960 until diag

# date in days.

#----------------------------------------------------------------

# (1) Age at diagnosis in days

simdatn2$agediagdays <- round(simdatn2$age*365.241)


# (2) Transform continuous year of diagnosis the date of diagnosis

# and subtract it from 01-01-1960.

simdatn2$diag_year<- floor(simdatn2$year)

simdatn2$diag_moCont <- as.numeric(substr(simdatn2$year,5,

nchar(simdatn2$year)))

simdatn2$diagDate <- as.Date(paste(simdatn2$diag_year,

"-01-01",sep=""))+

round(simdatn2$diag_moCont*365.241)
```

```
simdatn2$diagdays1960 <- as.numeric(simdatn2$diagDate-

as.Date("01/01/1960",

format="%d/%m/%Y"))



# (3) Survival time in days

simdatn2$timesurvD <- floor(simdatn2$timesurv*365.241)
```

2. STEP 2: CALCULATION OF PSEUDO-OBSERVATIONS

For the next step, we need to decide on the timepoints, at which pseudo-observations will be calculated. We chose 6 time-points (any number of time-points between 5-10 time-points should be adequate) based on a random selection of quantiles of the time-to-any-event distirbution.

```
times_c <- quantile(simdatn2$timesurv, probs=seq(0.15,1,0.15))

times_c <- round(times_c,1)

print(times_c)


##   15%   30%   45%   60%   75%   90%

##   0.4   1.0   2.3   4.8   9.4  10.0
```

Then, we use the leave-one-out estimator as was described in eq.1 (Section 2.1) where each $\widehat{\theta}_i$ is calculated using the non-parametric estimator which is shown in Section 2.3.1. In R-software this estimator is provided from R-package relsurv.

We run the cmp.rel command once, where all individuals are included and then we run it another n times (with n being the sample size), where each time one individual is excluded from the dataset (leave-one-out estimator). The non-parametric estimator (provided with cmp.rel) gives estimates for both cancer and population estimates and thus, 2 sets of pseudo-observations are created; one correspond-

ing to cancer (pseudo_CPr.1, pseudo_lyl.1) and another to the population (pseudo_CPr.2, pseudo_lyl.2). We show here how we can derive the pseudo-observations for both crude probabilities of death and life years lost from each cause at the same time. We have to note here though, that the pseudo-observations for the crude probabilities are estimated for all the time-points we mentioned before, while the pseudo-observations for the life years lost are only derived for the maximum time point, *i.e.* $t=10$ years.

```r
#-------------------------------------------------------------------
# L E A V E   O N E   O U T   E S T I M A T O R
#-------------------------------------------------------------------


# Thetas based on the whole sample
fit_all <- cmp.rel(Surv(timesurvD,vstat)~1+ratetable
(AGE.RT=agediagdays,
SEX.RT=sex,
YEAR.RT=diagdays1960),
ratetable=expectedrates.RT,data=simdatn2,tau=3652.41,
conf.int=0.95)


results_relsurv<- list(summary(fit_all, times = times_c)$est,
cbind(fit_all$causeSpec$area, fit_all$population$area))



ls <- list()


for (y in 1:nrow(simdatn2)){
```

```r
fit <- cmp.rel(Surv(timesurvD,vstat)~ratetable(AGE.RT=agediagdays,

SEX.RT=sex,YEAR.RT=diagdays1960),

ratetable=expectedrates.RT,data=simdatn2[-y,],tau=3652.41,

conf.int=0.95)

ls[[y]] <- list(summary(fit, times = times_c)$est,

cbind(fit$causeSpec$area, fit$population$area)) # to be stored

}


#!!!!!!

# TIP:

# An alternative using foreach from package doSNOW can be found in

# code.R found here-> https://github.com/pseudorel/supp_material.git



# Separate estimates based on the indicator and the cause


CPr.1 <- t(sapply(1:N, function (x) ls[[x]][[1]][1,]))

#dataframe dimensions: (N x times_c)

CPr.2 <- t(sapply(1:N, function (x) ls[[x]][[1]][2,]))


lyl.1 <- sapply(1:N, function (x) ls[[x]][[2]][,1])

#dataframe dimensions: (N x 1)

lyl.2 <- sapply(1:N, function (x) ls[[x]][[2]][,2])
```

```r
# Final step: leave - one - out estimator


pseudo_CPr.1<-data.frame(matrix(1,nrow(simdatn2),length(times_c)))

pseudo_CPr.2<-data.frame(matrix(1,nrow(simdatn2),length(times_c)))

colnames(pseudo_CPr.1)<- colnames(pseudo_CPr.2)<-

paste(times_c,"y",sep="")


pseudo_lyl.1<-data.frame(matrix(1,nrow(simdatn2),1))

pseudo_lyl.2<-data.frame(matrix(1,nrow(simdatn2),1))

colnames(pseudo_lyl.1)<- colnames(pseudo_lyl.2)<-

paste(max(times_c),"y",sep="")


for(y in 1:length(times_c)){

for (x in 1:N){

pseudo_CPr.1[x,y]<- N*results_relsurv[[1]][1,y]-(N-1)*CPr.1[x,y]

pseudo_CPr.2[x,y]<- N*results_relsurv[[1]][2,y]-(N-1)*CPr.2[x,y]

}

}


for (x in 1:N){

pseudo_lyl.1[x,]<- N*results_relsurv[[2]][,1]-(N-1)*lyl.1[x]

pseudo_lyl.2[x,]<- N*results_relsurv[[2]][,2]-(N-1)*lyl.2[x]


}
```

With this way, we summarised all the pseudo-observations into 4 dataframes, based on the indicator

and the cause of death. Before moving to the next step, it would be helpful to conceptualise the pseudo-observations and how they behave depending on the censoring time and status. According to Andersen & Pohar-Perme (2010), pseudo-observations calculated within the cause-specific setting in the case of censoring, tend to be negative at first and then jump above 1 in case of failing from cause of interest or remain negative (and decrease) in the case of failing from the other cause; while if they are censored, the pseudo-observations start increasing at the first next failure corresponding to the cause in question. To investigate if the same applies for the pseudo-observations that were derived through the relative survival setting, we provide 4 examples of individuals who experienced: 1. early censoring, 2. cancer death, 3. death from other causes, and 4. administrative censoring. t We examine the behaviour of the pseudo-observations that were calculated for cancer (pseudo_CPr.1) and the results are shown below.

[Table 1 about here.]

The first and the second cases agree with the previous statement. However, in the third case, although we would expect the pseudo-observations to remain negative and decrease, we notice a similar pattern with that of case 2, with the only exception being that the pseudo-observations in this case did not reach 1. Lastly, for those who were administratively censored, the pseudo-observations are negative and constantly decrease over time.

3. STEP 3: FIT THE MODELS FOR EACH INDICATOR AND FOR EACH CAUSE AND PROVIDE THE COVARIABLE EFFECTS.

For this step we used a new dataset called b which is an "extended" version of the initial dataset simdatn2. That means that the data were expanded in such way that now each individual instead of having one row of information, they have as many rows as the timepoints that the pseudo-observations were calculated (6). In this example, we used 6 timepoints for the pseudo-observations for the crude probabilities for each of the N individuals, thus the b dataset consisted of $N \times 6$ rows. One can think of this new dataset, as a balanced longitudinal dataset with the outcome (pseudo-observations) being measured at the same time-

points. As opposed to that, the dimensions of b corresponding to the pseudo-observations for the number of life years lost are the same with simdatn2.

As we have 2 sets of pseudo-observations for each indicator, one for cancer and one for other causes, we need to define 2 models to match each cause. For simplicity, we used the same covariates for all causes, an identity link and an independent working covariance structure. The model used in both cases included the variables agecr, sex and yearcr.

3.1 *Models for the Crude Probabilities of death from cancer and other causes*

```
linkfunc <- "identity"

covastr <- "independence"


for (h in 1:2){

pseudo <- get(paste("pseudo_CPr",h,sep="."))


b <- NULL

for (it in 1:length(times_c)) {

b <- rbind(b, cbind(simdatn2,

pseudo = pseudo[,it],

tpseudo = times_c[it],

id = 1:nrow(simdatn2)))

}

b <- b[order(b$id), ]

assign(paste("b_cpd", h, sep="."),b)
```

```r
#Put sex always 2nd and interaction before any variables!


pseudo_fit <- geese(pseudo ~ as.factor(tpseudo) +

agecr+sex+yearcr, data = b, id = id,

jack = TRUE, scale.fix = TRUE,

family = gaussian, mean.link = linkfunc,

corstr = covastr)
assign(paste("pseudo_fit_cpd", h, sep="."),pseudo_fit)

}


print(paste("Cause: Cancer"))

## [1] "Cause: Cancer"

print(summary(pseudo_fit_cpd.1)$mean)

##                        estimate      san.se       ajs.se
## (Intercept)          0.18592665 0.043376612 0.043319037
## as.factor(tpseudo)1    0.12214868 0.011010724 0.010966499
## as.factor(tpseudo)2.3  0.22912820 0.014575255 0.014516712
## as.factor(tpseudo)4.8  0.30213387 0.016839044 0.016771409
## as.factor(tpseudo)9.4  0.35894016 0.019110520 0.019033761
## as.factor(tpseudo)10   0.36339510 0.019357682 0.019279930
## agecr                0.06379199 0.009106758 0.009114273
## sex                 -0.01734549 0.027392288 0.027358494
## yearcr               0.17928122 0.162787334 0.162726497
```

```
##                           wald            p

## (Intercept)           18.3726603 1.816457e-05

## as.factor(tpseudo)1   123.0681789 0.000000e+00

## as.factor(tpseudo)2.3 247.1296018 0.000000e+00

## as.factor(tpseudo)4.8 321.9318649 0.000000e+00

## as.factor(tpseudo)9.4 352.7760493 0.000000e+00

## as.factor(tpseudo)10  352.4126075 0.000000e+00

## agecr                  49.0687160 2.471467e-12

## sex                     0.4009741 5.265866e-01

## yearcr                  1.2129095 2.707567e-01
```

```r
print(paste("Cause: Other causes"))
```

```
## [1] "Cause: Other causes"
```

```r
print(summary(pseudo_fit_cpd.2)$mean)
```

```
##                         estimate      san.se       ajs.se

## (Intercept)           0.05470728 0.0087119594 0.008701479

## as.factor(tpseudo)1   0.01346564 0.0004206486 0.000418959

## as.factor(tpseudo)2.3 0.03891401 0.0013402614 0.001334878

## as.factor(tpseudo)4.8 0.08101439 0.0031833026 0.003170517

## as.factor(tpseudo)9.4 0.14918598 0.0068458200 0.006818323

## as.factor(tpseudo)10  0.15728598 0.0073426576 0.007313165

## agecr                 0.03443196 0.0018134048 0.001813052

## sex                  -0.02570726 0.0056628217 0.005654986

## yearcr               -0.06548895 0.0333602341 0.033343488

##                           wald            p
```

```
## (Intercept)              39.43288 3.395310e-10

## as.factor(tpseudo)1   1024.74383 0.000000e+00

## as.factor(tpseudo)2.3  843.01036 0.000000e+00

## as.factor(tpseudo)4.8  647.69197 0.000000e+00

## as.factor(tpseudo)9.4  474.90315 0.000000e+00

## as.factor(tpseudo)10    458.85288 0.000000e+00

## agecr                   360.52379 0.000000e+00

## sex                      20.60847 5.634635e-06

## yearcr                    3.85370 4.963637e-02
```

3.2   *Models for the Number of Life Years Lost due to death from cancer and other causes*

```r
linkfunc <- "identity"

covastr <- "independence"

times_lyl<- max(times_c)


for (h in 1:2){

pseudo <- get(paste("pseudo_lyl",h,sep="."))


b <- NULL

for (it in 1:length(times_lyl)) {

b <- rbind(b, cbind(simdatn2,

pseudo = pseudo[,it],

tpseudo = times_lyl[it],

id = 1:nrow(simdatn2)))
```

```
}

b <- b[order(b$id), ]

assign(paste("b_lyl", h, sep="."),b)


pseudo_fit <- geese(pseudo ~ agecr+sex+yearcr, data = b, id = id,

jack = TRUE, scale.fix = TRUE,

family = gaussian, mean.link = linkfunc,

corstr = covastr)

assign(paste("pseudo_fit_lyl", h, sep="."),pseudo_fit)

}


print(paste("Cause: Cancer"))

## [1] "Cause: Cancer"

print(summary(pseudo_fit_lyl.1)$mean)

##               estimate    san.se    ajs.se       wald

## (Intercept)  4.6540300 0.4879353 0.4885423 90.9774649

## agecr        0.6967687 0.1009452 0.1012960 47.6437187

## sex         -0.2238582 0.3013415 0.3017322  0.5518592

## yearcr       2.0723053 1.7882753 1.7921353  1.3428847

##                         p

## (Intercept) 0.000000e+00

## agecr       5.111578e-12

## sex         4.575590e-01

## yearcr      2.465259e-01
```

```
print(paste("Cause: Other causes"))

## [1] "Cause: Other causes"

print(summary(pseudo_fit_lyl.2)$mean)

##                estimate      san.se      ajs.se        wald
## (Intercept)   1.4294683  0.10742974  0.10754160  177.051685
## agecr         0.3805871  0.01934978  0.01939555  386.861907
## sex          -0.2989296  0.06023483  0.06030396   24.628760
## yearcr       -0.6943286  0.35731594  0.35803635    3.775942
##                           p
## (Intercept)  0.000000e+00
## agecr        0.000000e+00
## sex          6.950661e-07
## yearcr       5.199463e-02
```

4. STEP 4: PREDICT THE POPULATION POINT ESTIMATES AND CONFIDENCE INTERVALS.

We start with the function pseudo_pred which is needed in order to predict the individual/population

point estimates and confidence intervals of the cumulative probabilities. The function is shown below:

```
pseudo_pred <- function(mod, tpseudo, var1, var2,var3,data,p)
```

where mod is the GEE model (run with function geese), tpseudo is a vector defining the timepoints

where pseudo-observations were calculated, var1-var3 stand for the non-time-dependent variables in

the model (we have only 3 variables), data are the data for which we do the predictions, times are

the time-points in which the pseudo-observations have been calculated in Step 3, and p is the number of

non-time-dependent variables in the model. The function can be extended to time-dependent variables.

```r
#Find the variance, cod, 95%ci

pseudo_pred <- function(mod, tpseudo, var1, var2, var3, data, p,
times){

xis  <-data.frame(data[,c(tpseudo,var1, var2,var3)])


Ntim <- length(times)

N <- nrow(data)/Ntim

xiss <-data.frame(matrix(1,nrow(data),Ntim))

xis<-cbind(xis,xiss)


betas<- mod$beta

cov_vars <- mod$vbeta.ajs


pred<-data.frame(matrix(1,N,Ntim))

var<-NULL


y=p


if (Ntim!=1){

for (i in 2:Ntim){

u<-y+1+i

xis[,u]<- ifelse(xis[,tpseudo]==times[i],1,0)

}


xis_transf <- as.matrix(cbind(xis[,c((ncol(xis)-
```

```r
Ntim+1):ncol(xis))],

xis[,c(2:(ncol(xis)-Ntim))]]))

#Define separately the first point

xis_transf.1 <- subset(xis_transf,xis_transf[,1]==1 &

rowSums(xis_transf[,2:Ntim])==0)

weights.1 <- matrix(rep(1/N,nrow(xis_transf.1)), nrow =1)

new_xis_trans.1 <- weights.1%*%xis_transf.1

var[1]<- as.numeric(new_xis_trans.1%*%cov_vars%*%t(new_xis_trans.1))

pred[,1] <- as.numeric(betas%*%t(xis_transf.1))


for (l in 2:Ntim){

xis_transf.l <- subset(xis_transf, xis_transf[,l]==1)

weights.l <- matrix(rep(1/N,nrow(xis_transf.l)), nrow =1)

new_xis_trans.l <- weights.l%*%xis_transf.l

var.l<- as.numeric(new_xis_trans.l%*%cov_vars%*%t(new_xis_trans.l))

pred.l <- as.numeric(betas%*%t(xis_transf.l))



pred[,l] <- pred.l

var[l] <- var.l

}


} else if (Ntim==1){

xis_transf <- as.matrix(cbind(xis[,ncol(xis)],

xis[,c(2:(ncol(xis)-1))]]))
```

```r
weights.1 <- matrix(rep(1/N,nrow(xis_transf)), nrow =1)

new_xis_trans.1 <- weights.1%*%xis_transf

var[1]<- as.numeric(new_xis_trans.1%*%cov_vars%*%t(new_xis_trans.1))

pred[,1] <- as.numeric(betas%*%t(new_xis_trans.1))

}


if (pseudo_fit$model$mean.link=="cloglog"){

est <- colMeans(1-exp(-exp(pred)))

lower <- 1-exp(-exp(log(-log(1-est))-qnorm(0.975)*sqrt(var)))

upper <- 1-exp(-exp(log(-log(1-est))+qnorm(0.975)*sqrt(var)))

var1 <- var*((1-est)*log(1-est))^2

} else if (pseudo_fit$model$mean.link=="log"){

est <- colMeans(exp(pred))

upper <- exp(log(est)+qnorm(0.975)*sqrt(var))

lower<-  exp(log(est)-qnorm(0.975)*sqrt(var))

var1 <- var*(est^2 )

} else if (pseudo_fit$model$mean.link=="identity"){

est <- colMeans(pred)

lower <- est-qnorm(0.975)*sqrt(var)

upper <- est+qnorm(0.975)*sqrt(var)

var1 <- var

}

return(as.data.frame(t(data.frame(rbind(est,lower,upper,var1)))))

}
```

Using the regression parameter estimates from the previous models and the function shown above, we predict the mean point estimates along with their confidence intervals at population level; although, this can be adopted for an individual or a subgroup. A comparison of these estimates to the non-parametric estimates (provided by `cmpr.rel`, see object `NP_fit`) could provide a goodness-of-fit check. For the crude probabilities of death, we provide also a graph except for the point estimates, for visual inspection.

```r
# N o n - P a r a m e t r i c

NP_fit <- cmp.rel(Surv(timesurvD,vstat)~ratetable(AGE.RT=agediagdays,

SEX.RT=sex,

YEAR.RT=diagdays1960),

ratetable=expectedrates.RT,data=simdatn2,

tau=max(times_c)*365.241,conf.int=0.95)




# C r u d e   p r o b a b i l i t i e s


for (h in 1:2){

pseudo_fit <- get(paste("pseudo_fit_cpd",h,sep="."))

b <- get(paste("b_cpd",h,sep="."))


pseudo_data<-pseudo_pred(mod=pseudo_fit, tpseudo="tpseudo",

var1="agecr", var2="sex", var3="yearcr",

data=b,p=3, times=times_c)

rownames(pseudo_data)<- paste(paste(paste(paste("CPr",h,sep="."),

"at",sep=" "), times_c, sep=" "),

"y",sep="")
```

```r
assign(paste("pseudo_data_cpd", h, sep="."),pseudo_data)

}



# L i f e   Y e a r s   L o s t


for (h in 1:2){

pseudo_fit <- get(paste("pseudo_fit_lyl",h,sep="."))

b <- get(paste("b_lyl",h,sep="."))


pseudo_data<-pseudo_pred(mod=pseudo_fit, tpseudo="tpseudo",

var1="agecr", var2="sex", var3="yearcr",

data=b,p=3, times=max(times_c))

assign(paste("pseudo_data_lyl", h, sep="."),pseudo_data)

}



# Plot to compare non-parametric and model estimates for CPr

plot(fit, conf.int = TRUE, ylab="Crude probabilities of death",

ylim=c(0,0.5))

points(times_c*365.241,pseudo_data_cpd.1$est, col="red", pch=19)

arrows(times_c*365.241,pseudo_data_cpd.1$lower,

times_c*365.241,pseudo_data_cpd.1$upper,

length=0.05, angle=90, code=3, col="red")

points(times_c*365.241,pseudo_data_cpd.2$est, col="red", pch=19)
```
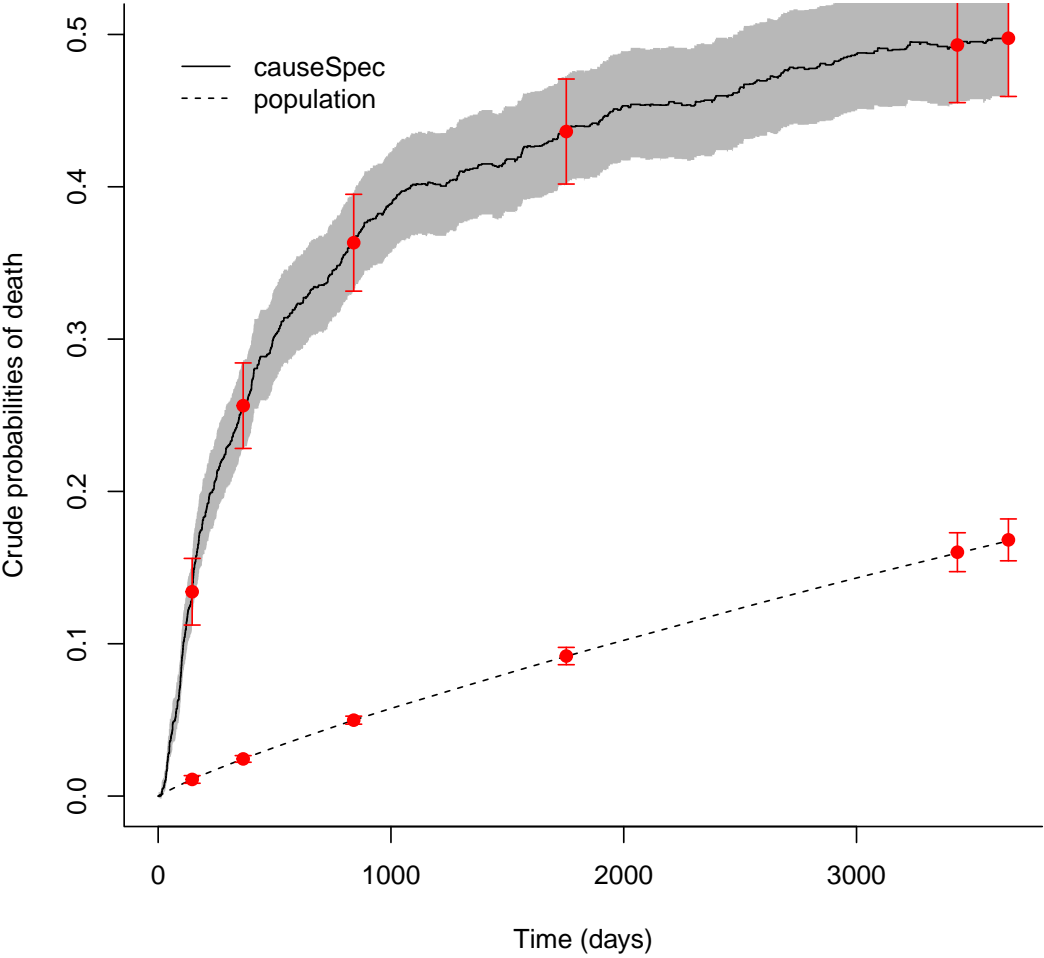
```
arrows(times_c*365.241,pseudo_data_cpd.2$lower,

times_c*365.241,pseudo_data_cpd.2$upper,

length=0.05, angle=90, code=3, col="red")
```


[Table 2 about here.]

	cens_time	cause	0.4y	1y	2.3y	4.8y	9.4y	10y
1	1.642	0	-0.0111	-0.0307	0.0113	0.1262	0.2157	0.2227
2	3.807	1	-0.0056	-0.0157	-0.0389	1.0636	1.0507	1.0497
3	3.540	2	-0.0151	-0.0408	-0.1017	0.9550	0.9432	0.9422
4	10.000	0	-0.0120	-0.0321	-0.0800	-0.1957	-0.5141	-0.5704

	est	lower	upper	Non Parametric
CPr.1 at 0.4y	0.13	0.11	0.16	0.13
CPr.1 at 1y	0.26	0.23	0.28	0.26
CPr.1 at 2.3y	0.36	0.33	0.40	0.36
CPr.1 at 4.8y	0.44	0.40	0.47	0.44
CPr.1 at 9.4y	0.49	0.46	0.53	0.49
CPr.1 at 10y	0.50	0.46	0.54	0.50
LYL.1 at 10y	4.03	3.74	4.33	4.03
CPr.2 at 0.4y	0.01	0.01	0.01	0.01
CPr.2 at 1y	0.02	0.02	0.03	0.02
CPr.2 at 2.3y	0.05	0.05	0.05	0.05
CPr.2 at 4.8y	0.09	0.09	0.10	0.09
CPr.2 at 9.4y	0.16	0.15	0.17	0.16
CPr.2 at 10y	0.17	0.15	0.18	0.17
LYL.2 at 10y	0.92	0.86	0.98	0.92

Table 1. Model estimates and confidence intervals (est, lower, upper) vs. non-parametric estimates for both measures and both causes