Supplementary Table 2

R code for the least squares (LS) estimation of β_0, β_1, σ by minimizing $\sum_t [\hat{S}(t) - S(t)]^2$, with the Nelson-Aalen survival function $\hat{S}(t)$, and the parametric survival function $S(t) = e^{-\beta_0 M(t)}[1 + \sigma^2 \beta_1 M(t)]^{-\sigma^{-2}}$.

```
################################
#S(t) contain M(t), the cumulative married mortality, a known function.
# M(t) is calculated from life tables for married, starting at M(0)=0
#these life table values are cumulated per day, and stored in an array: cum_married.array
#Since M(t) is a step function, our M(t) in the R-code is a taylor approximation
  #M(t)=m0*t+m1*t^2+m2*t^3

maxperiode<-7#years
alder<-matrix(c(55,64,65,74,75,84,85,94),nrow=2)#age groups
mx<-floor(maxperiode*365.25)+10#max t,   t=days
korttid<-seq(1,mx,1)

ai<-1##age group number 1,2,3,4
s<-1 ##male=1,female=2
a1<-alder[1,ai]
a2<-alder[2,ai]
yy<-cum_married.array[ai,s,1:mx]
res<-lm(yy~-1+korttid+I(korttid^2)+I(korttid^3))
m0<-res$coeff[1] ##coefficients in M(t)
m1<-res$coeff[2]
m2<-res$coeff[3]

library(survival)
#calculating Ŝ(t)
#survival data in data.df=data.frame(time,status,sex,age)
res.surv<-survfit(Surv(time,status)~1,data=data.df, subset=(age>=a1&age<=a2&sex==s),
type="fl")
ti<- res.surv $time[res.surv $time<=mx]# observed time points
Mt<-m0*ti+m1*ti^2+m2*ti^3 # a vector of M(t) at the observed time points
NAa<- -log(res.surv $surv[res.surv $time<=mx])#Nelson-Aalen estimator
St_hat<- res.surv $surv[res.surv $time<=mx]#Nelson-Aalen survival function Ŝ(t)


###############################
#########  defining some functions #####
###############################

#S(t) is now called StFrailty and calculated at t=tt
StFrailty<-function(tt)
{
	M<-m0*tt+m1*tt^2+m2*tt^3   #a singular value of M(t) at t=tt
	return(exp(-b0*M)*(1+b1*vari*M)^{-1/vari})
}
```

```
LSE<-function(start=c(1.2,.2,1),method=1)
{
	SS<-function(x)# x in a vector of length 3
	{
		b0<-x[1]
		b1<-x[2]
		delta<-x[3]# b1*vari
		sum((St_hat-exp(-b0*Mt)*(1+delta*Mt)^(-b1/delta))^2)
	}
	SS2<-function(x)#if b1=0
	{
		tau<-x[1] #tau=b0
		sum((St_hat-exp(-tau*Mt))^2)
	}
	SS1<-function(x)
	{
		b0<-x[1]
		b1<-x[2]
		delta<-x[3] # b1*vari
		##(Nelson-Aalen) -log(S(t))
		  sum((NAa-b0*Mt-(b1/delta)*log(1+delta*Mt))^2)
	}
	SS3<-function(x)#if b1=0
	{
		tau<-x[1] #tau=b0
		sum((NAa-tau*Mt)^2)
	}
	if (method==1)
	{
		res<-optim(start,SS,method="L-BFGS-
		B",lower=c(1,0,0.01),control=list(maxit=1000))
		res2<-nlm(SS2,1.2)
	}
	if (method==2)
	{
		res<-optim(start,SS1,method="L-BFGS-
		B",lower=c(1,0,0.01),control=list(maxit=1000))
		res2<-nlm(SS3,1.2)
	}
	if (res2$minimum>res$value)
	{
		#St_model<-exp(-res$par[1]*Mt)*(1+res$par[3]*Mt)^(-res$par[2]/res$par[3])
		sigma2<-res$par[3]/res$par[2]
		b0<-res$par[1]
		b1<-res$par[2]
		#delta<-res$par[3]
		conv<-res$convergence
		va<-res$value
	}
	if (res2$minimum<=res$value)
```

```
	{
		#St_model<-exp(-res2$estimate*Mt)
		sigma2<-0
		b0<-res2$estimate
		b1<-0
		#delta<-NA
		conv<-res2$code
		va<-res2$minimum
	}
	list(estb0=b0,estb1=b1,estsigma=sqrt(sigma2),va,conv)
	}
#### end function LSE######
##########################
##########################

######################
######run function LSE ####
###########################

value<-NULL
for (i in seq(100,2100,200))    #different start values for b1*vari (delta)
{
	tmp<-LSE(c(1.2,.3,i),1)
	##stores the sum of squares minimum value, and the start value i
	  value<-cbind(value,c(tmp[[4]],i))
}
i<-value[2,order(value[1,])[1]] # chose the best start value i
tmp<-LSE(c(1.2,.3,i))
####if b0 reach the lower limit 1, we change to method 2
	if (tmp[[1]]==1) tmp<-LSE(c(1.2,.3,i),2)


#################
####  results  ######
#################
data.frame(ai=ai,s=s
   ,estb0=tmp[[1]],estb1=tmp[[2]],estvari=(tmp[[3]])^2,convergence=tmp[[5]]))
```