#This program estimates a pooled curve with no age or Re-Apportionment adjustment

#load necessary R packages

library(minpack.lm)

library(Matrix)

library(metafor)

#define threshold concentration THRES, maximum concentration for plotting M, plotting increment bb

bb=1

#read in 2 files for each of the 15 cohorts with subject level data

#results file gives model parameter estimates and weights

# ap\_exp file gives concentrations for RR prediction

setwd ("C:/GEMM files")

dataACS<- read.csv("ACS results.csv", header = T)

dataxACS= read.csv("ACS ap\_exp.csv", header = T)

dataRome<- read.csv("Rome results.csv ", header = T)

dataxRome= read.csv("Rome ap.exp.csv ", header = T)

dataChinaH<- read.csv("China results.csv ", header=T)

dataChinaL<- read.csv("China IND COVARIATES results.csv ", header=T)

dataxChina= read.csv("China ap\_exp.csv", header = T)

dataHK<- read.csv("Hong Konk results.csv", header = T)

dataxHK= read.csv("Hong Kong ap\_exp.csv", header = T)

dataUK<- read.csv("England results.csv", header = T)

dataxUK= read.csv("England ap\_exp.csv", header = T)

dataCCHS<- read.csv("CCHS results.csv", header = T)

dataxCCHS<- read.csv("CCHS ap\_exp.csv", header = T)

dataAARP<- read.csv("AARP results.csv", header = T)

dataxAARP= read.csv("AARP ap\_exp.csv", header = T)

dataCanCHEC2001<- read.csv("CanCHEC 2001 results.csv", header = T)

dataxCanCHEC2001= read.csv("CanCHEC 2001 ap\_exp.csv", header = T)

dataBreast<- read.csv("Breast results.csv", header = T)

dataxBreast<- read.csv("Breast ap\_exp.csv", header = T)

dataNHS<- read.csv("NHS results.csv", header = T)

dataxNHS<- read.csv("NHS ap\_exp.csv", header = T)

dataNHIS<- read.csv("NHIS results.csv", header = T)

dataxNHIS<- read.csv("NHIS ap\_exp.csv", header = T)

dataCanCHEC1991<- read.csv("CanCHEC1991 results.csv", header = T)

dataxCanCHEC1991<- read.csv("CanCHEC1991 ap\_exp.csv", header = T)

dataCTS<- read.csv("CTS results.csv", header = T)

dataxCTS<- read.csv("CTS ap\_exp.csv", header = T)

dataVHM<- read.csv("VHM&PP results.csv", header = T)

dataxVHM<- read.csv("VHM&PP ap\_exp.csv", header = T)

dataDUELS <- read.csv(file = "DUELS results.csv", head=TRUE, sep=";", na.strings=c("."))

dataxDUELS <- read.csv(file = "DUELS ap\_exp.csv", head=TRUE, sep=";", na.strings=c("."))

#read in HR and CI for 18 ESCAPE cohorts not including VHM&PP cohort

dataESC<- read.csv("ESCAPE logHR se without VHM&PP.csv ", header = T)

#read in HR and CI for other cohorts

dataREST<- read.csv("HR CI Rest of World.csv ", header = T)

#read in cohort specific mortality rates for 5 GBD COD and all non-accidental by age group

rate<- read.csv("Mortality Rates by Cohort Age 80 plus.csv", header = T)

#set age as midpoint in 5 year age inteval

age=85

perc= rate[1:41,4]/100

medage=rate[1:41,5]

natM=rate[1:41,7]

CVM=rate[1:41,8]

natF=rate[1:41,14]

CVF=rate[1:41,15]

MCV=CVM/natM

FCV=CVF/natF

deltaCV=perc\*FCV+(1-perc)\*MCV

#this code adjusts the logHR for each cohort by proportion of non-accidental mortality rate that is CV

logrESC=dataESC[,2]

seESC=dataESC[,3]

denESC=dataESC[,5]

numESC=dataESC[,6]

logrESC=logrESC\*(numESC-denESC)

seESC=seESC\*(numESC-denESC)

ESCCVprop=c(deltaCV[10], deltaCV[24], deltaCV[25], deltaCV[26], deltaCV[27], deltaCV[28], deltaCV[7], deltaCV[22], deltaCV[23], deltaCV[13], deltaCV[9], deltaCV[14], deltaCV[29], deltaCV[12], deltaCV[17], deltaCV[19], deltaCV[18], deltaCV[15])

medageESC=c(medage[10], medage[24], medage[25], medage[26], medage[27], medage[28], medage[7], medage[22], medage[23], medage[13], medage[9], medage[14], medage[29], medage[12], medage[17], medage[19], medage[18], medage[15])

riskageESC=ESCCVprop\*exp(logrESC\*((age-110)/(medageESC-110))) + (1-ESCCVprop)\*exp(logrESC)

logrESC=log(riskageESC)

seESC=ESCCVprop\*seESC\*((age-110)/(medageESC-110)) + (1-ESCCVprop)\*seESC

denREST=dataREST[,5]

numREST=dataREST[,6]

logrREST=(log(dataREST[,2])/10)\*(numREST-denREST)

seREST=((log(dataREST[,4])-log(dataREST[,3]))/(10\*2\*1.96))\*(numREST-denREST)

RESTCVprop=c(deltaCV[36], deltaCV[37], deltaCV[38], deltaCV[21], deltaCV[39], deltaCV[30], deltaCV[11], deltaCV[40])

medageREST=c(medage[36], medage[37], medage[38], medage[21], medage[39], medage[30], medage[11], medage[40])

riskageREST=RESTCVprop\*exp(logrREST\*((age-110)/(medageREST-110))) + (1-RESTCVprop)\*exp(logrREST)

logrREST=log(riskageREST)

seREST=RESTCVprop\*seREST\*((age-110)/(medageREST-110)) + (1-RESTCVprop)\*seREST

xx=dataxDUELS[,2]

nxDUELS=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=18

mu=dataDUELS[a:e,2]

beta=dataDUELS[a:e,3]

se=dataDUELS[a:e,4]

tau=dataDUELS[a:e,5]

#ll=-dataDUELS[a:e,6]/2

f=as.character(dataDUELS[a:e,9])

weight=dataDUELS[a:e, 7]

output=cbind(f, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

#ll=as.numeric(out[,2])

wt=as.numeric(out[,2])

wt=matrix(wt)

mu=as.numeric(out[,3])

tau=as.numeric(out[,4])

beta=as.numeric(out[,5])

se=as.numeric(out[,6])

#wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VDUELS=diag(mvar, nx-1, nx-1)

MDUELS=rmean[2:nx]

DUELSden=xx[1]

DUELSnum=xx[2:nx]

rDUELS=mean(sd^2)/mean(sdw^2)

risk=deltaCV[20]\*exp(MDUELS\*((age-110)/(medage[20]-110))) + (1-deltaCV[20])\*exp(MDUELS)

MDUELS=log(risk)

mvar=(deltaCV[20]\*sqrt(mvar)\*((age-110)/(medage[20]-110)) + (1-deltaCV[20])\*sqrt(mvar))^2

VDUELS=diag(mvar, nx-1, nx-1)

xx=dataxCTS[,2]

nxCTS=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=19

mu=dataCTS[a:e,2]

beta=dataCTS[a:e,3]

se=dataCTS[a:e,4]

tau=dataCTS[a:e,5]

#ll=-dataCTS[a:e,6]/2

f=as.character(dataCTS[a:e,9])

weight=dataCTS[a:e, 7]

output=cbind(f, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

#ll=as.numeric(out[,2])

wt=as.numeric(out[,2])

wt=matrix(wt)

mu=as.numeric(out[,3])

tau=as.numeric(out[,4])

beta=as.numeric(out[,5])

se=as.numeric(out[,6])

#wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VCTS=diag(mvar, nx-1, nx-1)

MCTS=rmean[2:nx]

CTSden=xx[1]

CTSnum=xx[2:nx]

rCTS=mean(sd^2)/mean(sdw^2)

risk=deltaCV[35]\*exp(MCTS\*((age-110)/(medage[35]-110))) + (1-deltaCV[35])\*exp(MCTS)

MCTS=log(risk)

mvar=(deltaCV[25]\*sqrt(mvar)\*((age-110)/(medage[25]-110)) + (1-deltaCV[25])\*sqrt(mvar))^2

VCTS=diag(mvar, nx-1, nx-1)

xx=dataxVHM[,2]

nxVHM=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=20

mu=dataVHM[a:e,2]

beta=dataVHM[a:e,3]

se=dataVHM[a:e,4]

tau=dataVHM[a:e,5]

#ll=-dataVHM[a:e,6]/2

f=as.character(dataVHM[a:e,9])

weight=dataVHM[a:e, 7]

output=cbind(f, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

#ll=as.numeric(out[,2])

wt=as.numeric(out[,2])

wt=matrix(wt)

mu=as.numeric(out[,3])

tau=as.numeric(out[,4])

beta=as.numeric(out[,5])

se=as.numeric(out[,6])

#wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VVHM=diag(mvar, nx-1, nx-1)

MVHM=rmean[2:nx]

VHMden=xx[1]

VHMnum=xx[2:nx]

rVHM=mean(sd^2)/mean(sdw^2)

risk=deltaCV[1]\*exp(MVHM\*((age-110)/(medage[1]-110))) + (1-deltaCV[1])\*exp(MVHM)

MVHM=log(risk)

mvar=(deltaCV[1]\*sqrt(mvar)\*((age-110)/(medage[1]-110)) + (1-deltaCV[1])\*sqrt(mvar))^2

VVHM=diag(mvar, nx-1, nx-1)

xx=dataxBreast[,2]

nxBreast=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=19

mu=dataBreast[a:e,2]

beta=dataBreast[a:e,3]

se=dataBreast[a:e,4]

tau=dataBreast[a:e,5]

#ll=-dataBreast[a:e,6]/2

f=as.character(dataBreast[a:e,9])

weight=dataBreast[a:e, 7]

output=cbind(f, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

#ll=as.numeric(out[,2])

wt=as.numeric(out[,2])

wt=matrix(wt)

mu=as.numeric(out[,3])

tau=as.numeric(out[,4])

beta=as.numeric(out[,5])

se=as.numeric(out[,6])

#wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VBreast=diag(mvar, nx-1, nx-1)

MBreast=rmean[2:nx]

Breastden=xx[1]

Breastnum=xx[2:nx]

rBreast=mean(sd^2)/mean(sdw^2)

risk=deltaCV[2]\*exp(MBreast\*((age-110)/(medage[2]-110))) + (1-deltaCV[2])\*exp(MBreast)

MBreast=log(risk)

mvar=(deltaCV[2]\*sqrt(mvar)\*((age-110)/(medage[2]-110)) + (1-deltaCV[2])\*sqrt(mvar))^2

VBreast=diag(mvar, nx-1, nx-1)

xx=dataxCanCHEC1991[,2]

nxCanCHEC1991=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=20

mu=dataCanCHEC1991[a:e,2]

beta=dataCanCHEC1991[a:e,3]

se=dataCanCHEC1991[a:e,4]

tau=dataCanCHEC1991[a:e,5]

ll=-dataCanCHEC1991[a:e,6]/2

f=as.character(dataCanCHEC1991[a:e,9])

weight=dataCanCHEC1991[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VCanCHEC1991=diag(mvar, nx-1, nx-1)

MCanCHEC1991=rmean[2:nx]

CanCHEC1991den=xx[1]

CanCHEC1991num=xx[2:nx]

rCanCHEC1991=mean(sd^2)/mean(sdw^2)

risk=deltaCV[4]\*exp(MCanCHEC1991\*((age-110)/(medage[4]-110))) + (1-deltaCV[4])\*exp(MCanCHEC1991)

MCanCHEC1991=log(risk)

mvar=(deltaCV[4]\*sqrt(mvar)\*((age-110)/(medage[4]-110)) + (1-deltaCV[4])\*sqrt(mvar))^2

VCanCHEC1991=diag(mvar, nx-1, nx-1)

xx=dataxCanCHEC2001[,2]

nxCanCHEC2001=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=18

mu=dataCanCHEC2001[a:e,2]

beta=dataCanCHEC2001[a:e,3]

se=dataCanCHEC2001[a:e,4]

tau=dataCanCHEC2001[a:e,5]

ll=-dataCanCHEC2001[a:e,6]/2

f=as.character(dataCanCHEC2001[a:e,9])

weight=dataCanCHEC2001[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VCanCHEC2001=diag(mvar, nx-1, nx-1)

MCanCHEC2001=rmean[2:nx]

CanCHEC2001den=xx[1]

CanCHEC2001num=xx[2:nx]

rCanCHEC2001=mean(sd^2)/mean(sdw^2)

risk=deltaCV[3]\*exp(MCanCHEC2001\*((age-110)/(medage[3]-110))) + (1-deltaCV[3])\*exp(MCanCHEC2001)

MCanCHEC2001=log(risk)

mvar=(deltaCV[3]\*sqrt(mvar)\*((age-110)/(medage[3]-110)) + (1-deltaCV[3])\*sqrt(mvar))^2

VCanCHEC2001=diag(mvar, nx-1, nx-1)

xx=dataxAARP[,2]

nxAARP=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=21

mu=dataAARP[a:e,4]

beta=dataAARP[a:e,6]

se=dataAARP[a:e,7]

tau=dataAARP[a:e,5]

ll=-dataAARP[a:e,8]/2

f=as.character(dataAARP[a:e,2])

weight=dataAARP[a:e, 9]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "linear"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VAARP=diag(mvar, nx-1, nx-1)

MAARP=rmean[2:nx]

AARPden=xx[1]

AARPnum=xx[2:nx]

rAARP=mean(sd^2)/mean(sdw^2)

risk=deltaCV[31]\*exp(MAARP\*((age-110)/(medage[31]-110))) + (1-deltaCV[31])\*exp(MAARP)

MAARP=log(risk)

mvar=(deltaCV[31]\*sqrt(mvar)\*((age-110)/(medage[31]-110)) + (1-deltaCV[31])\*sqrt(mvar))^2

VAARP=diag(mvar, nx-1, nx-1)

xx=dataxCCHS[,2]

nxCCHS=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=19

mu=dataCCHS[a:e,2]

beta=dataCCHS[a:e,3]

se=dataCCHS[a:e,4]

tau=dataCCHS[a:e,5]

ll=-dataCCHS[a:e,6]/2

f=as.character(dataCCHS[a:e,9])

weight=dataCCHS[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VCCHS=diag(mvar, nx-1, nx-1)

MCCHS=rmean[2:nx]

CCHSden=xx[1]

CCHSnum=xx[2:nx]

rCCHS=mean(sd^2)/mean(sdw^2)

risk=deltaCV[5]\*exp(MCCHS\*((age-110)/(medage[5]-110))) + (1-deltaCV[5])\*exp(MCCHS)

MCCHS=log(risk)

mvar=(deltaCV[5]\*sqrt(mvar)\*((age-110)/(medage[5]-110)) + (1-deltaCV[5])\*sqrt(mvar))^2

VCCHS=diag(mvar, nx-1, nx-1)

xx=dataxUK[,2]

nxUK=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=19

mu=dataUK[a:e,4]

beta=dataUK[a:e,6]

se=dataUK[a:e,7]

tau=dataUK[a:e,5]

ll=-dataUK[a:e,8]/2

f=as.character(dataUK[a:e,2])

weight=dataUK[a:e, 9]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "linear"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VUK=diag(mvar, nx-1, nx-1)

MUK=rmean[2:nx]

UKden=xx[1]

UKnum=xx[2:nx]

rUK=mean(sd^2)/mean(sdw^2)

risk=deltaCV[8]\*exp(MUK\*((age-110)/(medage[8]-110))) + (1-deltaCV[8])\*exp(MUK)

MUK=log(risk)

mvar=(deltaCV[8]\*sqrt(mvar)\*((age-110)/(medage[8]-110)) + (1-deltaCV[8])\*sqrt(mvar))^2

VUK=diag(mvar, nx-1, nx-1)

xx=dataxHK[,2]

nxHK=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=20

mu=dataHK[a:e,2]

beta=dataHK[a:e,3]

se=dataHK[a:e,4]

tau=dataHK[a:e,5]

ll=-dataHK[a:e,6]/2

f=as.character(dataHK[a:e,9])

weight=dataHK[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VHK=diag(mvar, nx-1, nx-1)

MHK=rmean[2:nx]

HKden=xx[1]

HKnum=xx[2:nx]

rHK=mean(sd^2)/mean(sdw^2)

risk=deltaCV[41]\*exp(MHK\*((age-110)/(medage[41]-110))) + (1-deltaCV[41])\*exp(MHK)

MHK=log(risk)

mvar=(deltaCV[41]\*sqrt(mvar)\*((age-110)/(medage[41]-110)) + (1-deltaCV[41])\*sqrt(mvar))^2

VHK=diag(mvar, nx-1, nx-1)

xx=dataxACS[,2]

nxACS=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=21

mu=dataACS[a:e,2]

beta=dataACS[a:e,3]

se=dataACS[a:e,4]

tau=dataACS[a:e,5]

ll=-dataACS[a:e,6]/2

f=as.character(dataACS[a:e,9])

weight=dataACS[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VACS=diag(mvar, nx-1, nx-1)

MACS=rmean[2:nx]

ACSden=xx[1]

ACSnum=xx[2:nx]

rACS=mean(sd^2)/mean(sdw^2)

risk=deltaCV[33]\*exp(MACS\*((age-110)/(medage[33]-110))) + (1-deltaCV[33])\*exp(MACS)

MACS=log(risk)

mvar=(deltaCV[33]\*sqrt(mvar)\*((age-110)/(medage[33]-110)) + (1-deltaCV[33])\*sqrt(mvar))^2

VACS=diag(mvar, nx-1, nx-1)

xx=dataxChina[,2]

nxChina=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=18

mu=dataChinaH[a:e,2]

beta=dataChinaH[a:e,3]

se=dataChinaH[a:e,4]

tau=dataChinaH[a:e,5]

ll=-dataChinaH[a:e,6]/2

f=as.character(dataChinaH[a:e,9])

weight=dataChinaH[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

MChinaH=rmean[2:nx]

varH=sd^2

rChinaH=mean(sd^2)/mean(sdw^2)

xx=dataxChina[,2]

nxChina=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=18

mu=dataChinaL[a:e,2]

beta=dataChinaL[a:e,3]

se=dataChinaL[a:e,4]

tau=dataChinaL[a:e,5]

ll=-dataChinaL[a:e,6]/2

f=as.character(dataChinaL[a:e,9])

weight=dataChinaL[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

varL=sd^2

MChinaL=rmean[2:nx]

Chinaave=(MChinaH+MChinaL)/2

sdH=sqrt(varH[2:nx]+ (MChinaH-Chinaave)^2)

sdL=sqrt(varL[2:nx]+ (MChinaL-Chinaave)^2)

sdall=(sdL+sdH)/2

ss=matrix(0, nx-1, 1)

VV=(sdall)%\*%t(sdall)

for ( i in 1:nx-1) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VChina=diag(mvar, nx-1, nx-1)

MChinaL=rmean[2:nx]

Chinaden=xx[1]

Chinanum=xx[2:nx]

rChinaL=mean(sd^2)/mean(sdw^2)

MChina=(MChinaH+MChinaL)/2

rChina=(rChinaH+rChinaL)/2

risk=deltaCV[6]\*exp(MChina\*((age-110)/(medage[6]-110))) + (1-deltaCV[6])\*exp(MChina)

MChina=log(risk)

mvar=(deltaCV[6]\*sqrt(mvar)\*((age-110)/(medage[6]-110)) + (1-deltaCV[6])\*sqrt(mvar))^2

VChina=diag(mvar, nx-1, nx-1)

xx=dataxNHS[,2]

nxNHS=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=20

mu=dataNHS[a:e,2]

beta=dataNHS[a:e,3]

se=dataNHS[a:e,4]

tau=dataNHS[a:e,5]

ll=-dataNHS[a:e,6]/2

f=as.character(dataNHS[a:e,9])

weight=dataNHS[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VNHS=diag(mvar, nx-1, nx-1)

MNHS=rmean[2:nx]

NHSden=xx[1]

NHSnum=xx[2:nx]

rNHS=mean(sd^2)/mean(sdw^2)

risk=deltaCV[34]\*exp(MNHS\*((age-110)/(medage[34]-110))) + (1-deltaCV[34])\*exp(MNHS)

MHNS=log(risk)

mvar=(deltaCV[34]\*sqrt(mvar)\*((age-110)/(medage[34]-110)) + (1-deltaCV[34])\*sqrt(mvar))^2

VNHS=diag(mvar, nx-1, nx-1)

xx=dataxNHIS[,2]

nxNHIS=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=19

mu=dataNHIS[a:e,2]

beta=dataNHIS[a:e,3]

se=dataNHIS[a:e,4]

tau=dataNHIS[a:e,5]

ll=-dataNHIS[a:e,6]/2

f=as.character(dataNHIS[a:e,9])

weight=dataNHIS[a:e, 7]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

wt=exp((ll-max(ll)))/sum(exp(((ll-max(ll)))))

wt=matrix(wt)

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "z\*logit"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log(z)\*logit"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VNHIS=diag(mvar, nx-1, nx-1)

MNHIS=rmean[2:nx]

NHISden=xx[1]

NHISnum=xx[2:nx]

rNHIS=mean(sd^2)/mean(sdw^2)

risk=deltaCV[32]\*exp(MNHIS\*((age-110)/(medage[32]-110))) + (1-deltaCV[32])\*exp(MNHIS)

MHNIS=log(risk)

mvar=(deltaCV[32]\*sqrt(mvar)\*((age-110)/(medage[32]-110)) + (1-deltaCV[32])\*sqrt(mvar))^2

VNHIS=diag(mvar, nx-1, nx-1)

xx=dataxRome[,2]

nxRome=length(xx)-1

x=xx-min(xx)

nx=length(x)

r=max(x)-min(x)

a=1

e=18

mu=dataRome[a:e,4]

beta=dataRome[a:e,6]

se=dataRome[a:e,7]

tau=dataRome[a:e,5]

ll=-dataRome[a:e,8]/2

f=as.character(dataRome[a:e,2])

weight=dataRome[a:e, 9]

output=cbind(f, ll, weight, mu, tau, beta, se)

out=subset(output, weight>0)

f=out[,1]

ll=as.numeric(out[,2])

wt=as.numeric(out[,3])

wt=matrix(wt)

mu=as.numeric(out[,4])

tau=as.numeric(out[,5])

beta=as.numeric(out[,6])

se=as.numeric(out[,7])

nf=length(f)

T=matrix(0, nx, nf)

sd=matrix(0, nx, 1)

sdw=matrix(0, nx, 1)

sdb=matrix(0, nx, 1)

rmean=matrix(0, nx, 1)

meanrisk=matrix(0, nx, 1)

var=matrix(0,nx,nf)

varw=matrix(0,nx,nf)

varb=matrix(0,nx,nf)

upcl<-matrix(0, nx, 1)

lowcl<-matrix(0, nx, 1)

for (i in 1:nx) {

for (k in 1:nf) {

if (f[k]== "linear"){

T[i,k]<-x[i]/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

if (f[k] == "log"){

T[i,k]<-log(x[i]+1)/(1+exp(-(x[i]-mu[k])/(tau[k]\*r)))

}

}}

for (i in 1:nx) {rmean[i]=(T[i,]\*beta)%\*%wt}

for (k in 1:nf) { for (i in 1:nx) {

var[i,k]=(T[i,k]\*se[k])^2 + (T[i,k]\*beta[k]-rmean[i])^2

varw[i,k]=(T[i,k]\*se[k])^2

varb[i,k]=(T[i,k]\*beta[k]-rmean[i])^2 }}

for (i in 1:nx) {

sd[i]=sqrt(var[i,])%\*%wt

sdw[i]=sqrt(varw[i,])%\*%wt

sdb[i]=sqrt(varb[i,])%\*%wt

}

ss=matrix(0, nx, 1)

VV=(sd)%\*%t(sd)

for ( i in 1:nx) { ss[i]=sum(VV[i,])}

mvar= max(ss)

VRome=diag(mvar, nx-1, nx-1)

MRome=rmean[2:nx]

Romeden=xx[1]

Romenum=xx[2:nx]

rRome=mean(sd^2)/mean(sdw^2)

risk=deltaCV[16]\*exp(MRome\*((age-110)/(medage[16]-110))) + (1-deltaCV[16])\*exp(MRome)

MRome=log(risk)

mvar=(deltaCV[16]\*sqrt(mvar)\*((age-110)/(medage[16]-110)) + (1-deltaCV[16])\*sqrt(mvar))^2

VRome=diag(mvar, nx-1, nx-1)

ratio=c(rBreast, rCanCHEC1991, rACS, rNHS, rNHIS, rHK, rUK, rAARP, rRome, rCCHS, rCanCHEC2001, rVHM, rDUELS, rChina)

infl=mean(ratio)

vESC=seESC^2

VESC=bdiag(vESC[1], vESC[2], vESC[3], vESC[4], vESC[5], vESC[6], vESC[7], vESC[8], vESC[9], vESC[10], vESC[11], vESC[12], vESC[13], vESC[14], vESC[15], vESC[16], vESC[17], vESC[18])\*infl

vREST=seREST^2

VREST=vREST\*infl

VESC=VESC\*infl

VRESTUS=bdiag(vREST[1], vREST[2], vREST[3], vREST[5], vREST[8])

VUS=bdiag(VACS, VAARP, VNHS, VNHIS, VCTS, VRESTUS)

VRESTEur=bdiag(vREST[4], vREST[7])

VEurope=bdiag(VUK, VRome, VDUELS, VVHM, VESC, VRESTEur)

VRESTASIA=bdiag(VREST[6])

VAsia=bdiag(VHK, VChina, VRESTASIA)

VCanada=bdiag(VCanCHEC1991, VCanCHEC2001, VCCHS, VBreast)

Vall=bdiag(VCanada, VUS, VEurope, VAsia)

logr=c(MCanCHEC1991, MCanCHEC2001, MCCHS, MBreast, MACS, MAARP, MNHS, MNHIS, MCTS, logrREST[1], logrREST[2], logrREST[3], logrREST[5], logrREST[8], MUK, MRome, MDUELS, MVHM, logrESC, logrREST[4], logrREST[7], MHK, MChina, logrREST[6])

den=c(rep(CanCHEC1991den, nxCanCHEC1991), rep(CanCHEC2001den, nxCanCHEC2001), rep(CCHSden, nxCCHS), rep(Breastden, nxBreast), rep(ACSden, nxACS), rep(AARPden, nxAARP), rep(NHSden, nxNHS), rep(NHISden, nxNHIS), rep(CTSden, nxCTS), denREST[1], denREST[2], denREST[3], denREST[5], denREST[8], rep(UKden, nxUK), rep(Romeden, nxRome), rep(DUELSden, nxDUELS), rep(VHMden, nxVHM), denESC, denREST[4], denREST[7], rep(HKden, nxHK), rep(Chinaden, nxChina), denREST[6])

num=c(CanCHEC1991num, CanCHEC2001num, CCHSnum, Breastnum, ACSnum, AARPnum, NHSnum, NHISnum, CTSnum, numREST[1], numREST[2], numREST[3], numREST[5], numREST[8], UKnum, Romenum, DUELSnum, VHMnum, numESC, numREST[4], numREST[7], HKnum, Chinanum, numREST[6])

study=c(rep(1, nxCanCHEC1991), rep(2, nxCanCHEC2001), rep(3, nxCCHS), rep(4, nxBreast), rep(5, nxACS), rep(6, nxAARP), rep(7, nxNHS), rep(8, nxNHIS), rep(9, nxCTS), 10, 11, 12, 13, 14, rep(15, nxUK), rep(16, nxRome), rep(17, nxDUELS), rep(18, nxVHM), seq(19, 36), 37, 38, rep(39, nxHK), rep(40, nxChina), 41)

study=as.factor(study)

THRES=min(den)

numt=((num-THRES)+abs(num-THRES))/2

dent=((den-THRES)+abs(den-THRES))/2

diff=log(numt/1.6+1)/(1+exp(-(numt-15.5)/36.8)) - log(dent/1.6+1)/(1+exp(-(dent-15.5)/36.8))

fit=rma.mv(yi=logr, V=Vall, mods=~diff -1, random=list( ~ 1 | study), method="REML", intercept=FALSE,

struct="CS")

coef(fit)

sqrt(vcov(fit))