

```
#####
##### Section 2
#####

#Figure 2a and 2b
meanG <- 1; varG= 1
s <- varG/meanG
a <- meanG^2/varG
x <- seq(from=0,to=6,length.out=1000)
t <- seq(from=0, to = 10, length.out=1000)
hazardRatio <- function(t,l=10,r=1,delta=1){
  rt <- rt(t,l=1,r=r)
  nominator <- 1 + delta * t
  denominator <- 1 + t + t*r - 0.5 * t^2 / l * r
  return(rt * nominator / denominator)
}
rt <- function(t,l=10,r=1){
  return(1 + r*(1-t/l))
}
hazardr <- hazardRatio(t,l=10)
par(mfrow=c(1,2))
plot(x,dgamma(x, shape = a, scale = s),type="l",xlab="Frailty", ylab =
"Probability density",main="A")
plot(t, hazardr, type = "l",ylim=c(0.6,2),xlim=c(0,10),ylab="Hazard
ratio",xlab="Time",main="B")
lines(t, rt(t),lty=2)
abline(h=1,lty=3)
legend("topright",c("Observed population estimate","Individual causal effect"),
lty = 1:2)
#Figure 2a and Figure 2b finished

#Function to find parameters of the Compound Poisson distribution
findMandV <- function(prop, exp, var){
  p <- -log(prop)
  v <- 1 / (var/exp - exp/p)
  #v <- exp / var * (1/(1-exp^2/(var*p)))
  m <- exp * v / p
  return(c(m,v))
}

#Function to find parameters of the inverse Gaussian function
findVandPinvGaussian <- function(exp, var){
  m = -0.5
  v <- exp * (m+1) / var
  p = exp * v / m
  return(c(v,p))
}

### get risks if frailty is Compound Poisson distributed
riskRatio6.15<- function(p,m,v,A,r){
  nominator <- r * (1 + A / v)^(m + 1)
  denominator <- (1 + r*A / v)^(m + 1)
  return(nominator/denominator)
}
```

```

riskRatioCompoundPoisson <- function(prop,exp, var,r,A){
  mv <- findMandV(prop,exp, var)
  print(mv)
  m <- mv[1]; v=mv[2]
  p = -log(prop)
  return(riskRatio6.15(p,m,v,A,r))
}
getPoissonRisks <- function(prop=0.970,exp=1, var=35,r=1.81,a=0.0033,years=1:8){
  output <- NULL
  for(i in 1:length(years)){
    output[i] <- riskRatioCompoundPoisson(prop,exp, var,r,a*years[i])
  }
  return(output)
}

#####
### Section 2.1
#####
par(mfrow=c(1,1))

#Imported data from WHI results on Hormone Therapy and Cardiovascular risk
(published in New England Journal of medicine)
dataNEJM <- matrix(ncol=3,byrow=T,c(1.81,1.09,3.01,1.34
,0.82,2.18,1.27,0.64,2.50,1.25,0.74,2.12,1.45,0.81,2.59,0.70,0.42,1.14))

#Figure 3a
pRisks <- getPoissonRisks(0.970,1,35,1.81,0.0033)
plot(0:7,c(1.81,pRisks[1:7]),ylim=c(0.3,3.0),lwd=2,type="l",lty=2,ylab="Hazard
ratio",xlab="Time since treatment initiation (years)",xlim=c(0,6))
lines(0:7,dataNEJM[c(1:6,6,6),1],lwd=2,col=2,type="s")
#lines(0:7,c(1.81,gammaRisks[1:7]),lty=3,lwd=2)
legend(x="topright",c("Observed HR by WHI","Compound Poisson
frailty"),lty=c(1,2), col=c(2, 1),cex=0.75)
#legend(x="topright",c("Observed HR","Compound Poisson","Gamma"),lty=c(1,1,2),
col=c(2, 1,1),cex=0.75)
for(time in 1:length(dataNEJM[,1])){
  rect(time-1,dataNEJM[time,2],time, dataNEJM[time,3],col=rgb(1, 0.1, 0.0, 0.15,
names = NULL, maxColorValue = 1),border=NA)
}

# Code to plot the distribution of the hazard rate (i.e. the heterogeneous
hazard rate)
rr = 1.81; variance2 = 1
xval = seq(from=1,to=3,length.out=1000)
yval <- dgamma(xval,shape=(rr-1)/variance2,scale=variance2) + 1
plot(xval, yval,type="l")

#####
# Simulations of scenarios to obtain Figure 3b and Figure 3c
#####
r = 1.81 #For the estimate of the underlying risk in the hormone therapy example
# Function to simulate a population of n individuals that experience the 3
counterfactual scenarios

```

```

getRateRatios <- function(n=1e6, variance1=40, variance2=2,rr=1.81,alpha=3.3e-
3,numberOfYears=7,p=0.970,shapeG=4.5794193,scaleG=0.1394855,gamma=T,uniform=F){
  #n: Number of individuals in the simulations
  #variance1: Variance of the Gamma frailty
  #Variance2: of the Gamma distribution that describes the relative risk (r),
  which has mean rr
  if(gamma==T) z <- rgamma(n,shape=1/variance1,scale=variance1)
  else{
    pois <- rpois(n,-log(p))
    nonZeros <- which(pois != 0)
    z <- rep(0,n)
    for(i in nonZeros){
      z[i] <- sum(rgamma(n=pois[i],shape=shapeG,scale=1/scaleG))
    }
  }
  if(uniform==FALSE) r <- rgamma(n,shape=(rr-1)/variance2,scale=variance2) + 1
  else r <- runif(n,min=1,max=1.81+0.81)
  #Make a matrix "years" which shows the events for each individual (row) in
  each year (col)

  #simulate from a Compound poisson
  year0 <- rep(0,n)
  yearX <- matrix(ncol=numberOfYears,nrow=n,NA)
  years <- cbind(year0,yearX)
  yearsNoRisk <- years
  yearsConstantRisk <- years

  for(year in 1:numberOfYears){
    selected <- which(years[,year]==0)
    for(sel in selected){
      years[sel,year+1] <- rbinom(1,1,min(1,z[sel]*r[sel]*alpha))
    }
    selected2 <- which(yearsNoRisk[,year]==0)
    for(sel2 in selected2){
      yearsNoRisk[sel2,year+1] <- rbinom(1,1,min(1,z[sel2]*alpha))
    }
    selected3 <- which(yearsConstantRisk[,year]==0)
    for(sel3 in selected3){
      yearsConstantRisk[sel3,year+1] <- rbinom(1,1,min(1,z[sel3]*rr*alpha))
    }
  }
  return(list(years,yearsConstantRisk,yearsNoRisk))
}

#Returns risk Hazard rates for mat1 (col1), mat2(col2),mat3 (col3), Hazard ratio
of heterogen risk vs no risk (col4), homogen risk vs no risk (col5), and the
actual, underlying population mean of the relative risk (col6)
findEmpiricalRatios <- function(mat1,mat2,mat3){
  ys <- length(mat2[1,])
  output <- matrix(NA, nrow=ys,ncol=6)
  for(i in 1:ys){
    output[i,1] <- sum(na.omit(mat1[,i])==1)/length(na.omit(mat1[,i]))
    output[i,2] <- sum(na.omit(mat2[,i])==1)/length(na.omit(mat2[,i]))
    output[i,3] <- sum(na.omit(mat3[,i])==1)/length(na.omit(mat3[,i]))
  }
}

```

```

    output[i,4] <- output[i,1]/output[i,3]
    output[i,5] <- output[i,2]/output[i,3]
    output[i,6] <- sum(r[which(!is.na(mat1[,i]))]) / sum(!is.na(mat1[,i]))
  }
  return(output)
}
ratioGamma <- getRateRatios(gamma=F)
comparedRatiosGamma <-
findEmpiricalRatios(ratioGamma[[1]],ratioGamma[[2]],ratioGamma[[3]])

getRiskRatiosClean <- function(n=1e6, variance1=35,
variance2=2,rr=1.81,alpha=3.3e-
3,numberOfYears=7,p=0.970,shapeG=4.5794193,scaleG=0.1394855,gamma=F,uniform=F) {
  temp1 <-
getRateRatios(n,variance1,variance2,rr,alpha,numberOfYears,p,shapeG,scaleG,gamma
=gamma,uniform=uniform)
  temp2 <- findEmpiricalRatios(temp1[[1]],temp1[[2]],temp1[[3]])
  return(temp2)
}

findMeans <- function(list1){
  size <- length(list1)
  output <- matrix(0,nrow=7,ncol=2)
  for(listNr in 1:size){
    thisList <- list1[[listNr]]
    output <- output + thisList[2:8,4:5]
  }
  return(output/size)
}

#Output to be plotted
outputCompPois3 <- lapply(1:10,function(x) getRiskRatiosClean(gamma=F))
outputComp3 <- lapply(1:10,function(x) getRiskRatiosClean(gamma=T))
theseMeansCP3 <- findMeans(outputCompPois3)
theseMeans3 <- findMeans(outputComp3)

#Figure 3b
plot(0:6,dataNEJM[c(1:6,6),1],lwd=3,col=2,type="s",ylim=c(0.3,3.0),lty=1,ylab="H
azard ratio",xlab="Time since treatment initiation (years)",xlim=c(0,6))
#lines(0:7,c(1.81,gammaRisks[1:7]),lty=3,lwd=2)
#legend(x="topright",c("Observed HR by WHI","Compound Poisson
frailty"),lty=c(1,2), col=c(2, 1),cex=0.75)
legend(x="topright",c("Observed HR","Homogeneous risk","Heterogeneous
risk"),lty=c(1,2,2), col=c(2, 1,4),cex=0.75)
for(time in 1:length(dataNEJM[,1])){
  rect(time-1,dataNEJM[time,2],time, dataNEJM[time,3],col=rgb(1, 0.1, 0.0, 0.15,
names = NULL, maxColorValue = 1),border=NA)
}
lines(0:6,c(theseMeansCP3[1:7,1]),type="s",lty=5,lwd=2,col=4)
lines(0:6,c(theseMeansCP3[1:7,2]),type="s",lty=5,lwd=2,col=1)

```

```

#Figure 3c
plot(0:6,dataNEJM[c(1:6,6),1],lwd=3,col=2,type="s",ylim=c(0.3,3.0),lty=1,ylab="Hazard ratio",xlab="Time since treatment initiation (years)",xlim=c(0,6))
#lines(0:7,c(1.81,gammaRisks[1:7]),lty=3,lwd=2)
#legend(x="topright",c("Observed HR by WHI","Compound Poisson frailty"),lty=c(1,2), col=c(2, 1),cex=0.75)
legend(x="topright",c("Observed HR","Homogeneous risk","Heterogeneous risk"),lty=c(1,2,2), col=c(2, 1,4),cex=0.75)
for(time in 1:length(dataNEJM[,1])){
  rect(time-1,dataNEJM[time,2],time, dataNEJM[time,3],col=rgb(1, 0.1, 0.0, 0.15, names = NULL, maxColorValue = 1),border=NA)
}
lines(0:6,c(theseMeans3[1:7,1]),type="s",lty=5,lwd=2,col=4)
lines(0:6,c(theseMeans3[1:7,2]),type="s",lty=5,lwd=2,col=1)

#####
#Formulas for the Alzheimer's disease example, Section 3
#####
varG = 2.5; c = 1.8
findNominator <- function(A1,A2,delta){
  return(1 + delta*(A1+A2))
}
findNominator(-log(0.55),-log(0.83), varG) / findNominator(-log(0.55)*c,-log(0.83), varG)
findNominator(-log(0.77),-log(0.97), varG) / findNominator(-log(0.77)*c,-log(0.97), varG)

#####
## Formulas behind results in Section 4 (which are elaborated in Appendix 1.3 and 1.4)
#####
ltk <- function(a,d,c,t){
  nom <- a*(1+d*c+d*a*t)^(-1-1/d) + k*(1+d*c+d*a*t)^(-1/d)
  denom <- a*(1+d*a*t)^(-1-1/d) + k*(1+d*a*t)^(-1/d)
  return(nom/denom)
}
dltk <- function(a,d,c,t){
  nom <- (-1-1/d)*a*d*(1+d*c+d*a*t)^(-2-1/d) + (-1/d)*k*d*(1+d*c+d*a*t)^(-1-1/d)
  denom <- a*(1+d*a*t)^(-1-1/d) + k*(1+d*a*t)^(-1/d)
  return(nom/denom)
}
lt <- function(a,d,c,t){
  nom <- (1+d*c+d*a*t)^(-1-1/d)
  denom <- (1+d*a*t)^(-1-1/d)
  return(nom/denom)
}
dlt <- function(a,d,c,t){
  nom <- (-1-1/d)*d*(1+d*c+d*a*t)^(-2-1/d)
  denom <- (1+d*a*t)^(-1-1/d)
  return(nom/denom)
}

muks <- function(s,m,g,a,d,t){

```

```

    return(m + g*(-dltk(a,d,g*s,t))/ltk(a,d,g*s,t))
}
mus <- function(s,m,g,a,d,t){
  return(g*(-dlt(a,d,g*s,t))/lt(a,d,g*s,t))
}

#Explicit calculations of the scenarios in Section 4 (i.e. Figure 4a and 4b)
s <- seq(0,5,length.out=1000)
t = 1; k =1; a = 1; d = 5; m = 0; g = 1;
output1 <- sapply(s, muks, m=m,g=g,a=a,d=d,t=t )
output2 <- sapply(s, mus, m=m,g=g,a=a,d=d,t=t )
output3 <- sapply(s, muks, m=0.1,g=g,a=a,d=d,t=t )
output4 <- sapply(s, mus, m=0.1,g=g,a=a,d=d,t=t )
head(output)
myExpression <- expression("hazard Rate", 'E'[2])

#Plotting of Figure 4a and 4b
par(mfrow=c(1,2))
#plot(s,output2,col=2,type="l",ylim=c(0,1), main="A",lwd=2, ylab="Hazard rate",
xlab=paste("s (time in years since ",myExpression,")",sep=""))
plot(s,output2,col=2,type="l",ylim=c(0,1), main="A",lwd=2, ylab="Hazard rate",
xlab=expression(paste("s (time in years since ','E'[1]','"))))
lines(s,output1,type="l",lwd=2)
legend("topright",c("High risk group","Low risk group"), col=1:2,lwd=2)
plot(s,output4,col=2,type="l",ylim=c(0,1), main="B",lwd=2, ylab="Hazard rate",
xlab=expression(paste("s (time in years since ','E'[1]','"))))
lines(s,output3,type="l",lwd=2)
legend("topright",c("High risk group","Low risk group"), col=1:2,lwd=2)

#####
#Appendix Figure 1
#####
t <- seq(from=0,to=10,length.out=1000)
#R implementation of formulas in Aalen et al, 2008.
riskRatio6.20 <- function(delta, A, r,t=1){
  return( r * (1+ delta*A*t) / (1 + delta * r * A*t) )
}

riskRatio6.21 <- function(v,t,A,r,m){
  return(r * ( (v+A*t) / (v + r*A*t) )^(m+1) )
}
gammaVal <- sapply(t, riskRatio6.20, delta = 1, A = 0.2, r = 2)
pvcVal <- sapply(t, riskRatio6.21, v = 6, m = 5, A=0.2, r = 2)
plot(t, gammaVal, type="l",ylim = c(0.4,3), ylab="Hazard ratio",xlab="Time")
lines(t, pvcVal,col =2)
abline(h=2, lty = 2, col =1)
abline(h=1, lty = 3, col =1)
legend("topright",c("Conditional on frailty", "Observed estimate, Compound
Poisson", "Observed estimate, Gamma"),col=c(1,2,1), lty = c(2,1,1))
#####
##### End
#####

```