```
# Fitting the GB2 distribution to the Singapore Insurance claims data, Company 35 (ignoring Own Damages).
#
# Parameters are to be called alpha (parm[1]), reg_eqn, gamma1 (parm[2]) and gamma2 (parm[3]).
# No premium for a regressor
# read the data file

rm(list=ls())
PolClaims <- read.csv("C:/Users/valdezea/Desktop/Bogota_Colombia 2014/GB2EstimationCode-
Dec7/Co35polclaims_cleaned.csv")
attach(PolClaims)
Claims <- Tot_Payment

# define age categories (<25,25-35,35-45,45-55,55-65,65+)
AgeGroup<-1+(AgeInsured>20)+(AgeInsured>30)+(AgeInsured>40)+(AgeInsured>50)+(AgeInsured>60)+(AgeInsured>70)
# create year indicator (1996 or not);
YearI<-numeric(length(Claims)); YearI[which(Year==1996)]<-1
# define NCD0 - which is indicator if there is ANY no claim bonus;
NCD0<-rep(1,length(Claims)); NCD0[which(NCD==0)]<-0
# create marital variables;
MaritalM<-numeric(length(Claims)); MaritalM[which(Marital=="M")]<-1
# create gender variables
SexM<-numeric(length(Claims)); SexM[which(SexInsured=="M")]<-1
# create vehicle type (car indicator) variables
VTypeCar<-numeric(length(Claims));VTypeCar[which(VehicleType=="A")]<-1
# define comprehensive coverage
Comp<-numeric(length(Claims));Comp[which(CoverType=="C")]<-1;
# redefine the exposure
expose <- TLength

# For the regression, we must scale premium, as the optim hits a floating point limit;
PremW<-PremW/10000;

# density function of GB2
"dGB2" <- function(x,alpha,beta,gamma1,gamma2)
{
      num <- abs(alpha)*(x^(alpha*gamma1-1))*(beta^(alpha*gamma2))
      temp <- (beta^alpha+x^alpha)^(gamma1+gamma2)
      den <- beta(gamma1,gamma2)*temp
      result <- num/den
      return(result)
```

```
}
# cumulative distribution function of GB2
"pGB2" <- function(x,alpha,beta,gamma1,gamma2)
{
      temp <- (x/beta)^alpha
      result <- pbeta(temp/(1+temp),gamma1,gamma2)
      return(result)
}
# quantile function of GB2
"qGB2" <- function(q,alpha,beta,gamma1,gamma2)
{
      f <- function(x,q.q,alpha.q,beta.q,gamma1.q,gamma2.q){
      tmp1 <- (x/beta.q)^alpha.q
      tmp2 <- pbeta(tmp1/(1+tmp1),gamma1.q,gamma2.q)
      result <- tmp2-q.q
      return(result)
      }
      temp <- rep(0,length(q))
      for (i in 1:length(q)){
      tmpx <- ifelse(q[i]==0,0,ifelse(q[i]>0.9999999,72000000,uniroot(f, c(0,1000000), tol = 0.0001,
             q.q=q[i],alpha.q=alpha,beta.q=beta,gamma1.q=gamma1,gamma2.q=gamma2)$root))
      temp[i] <- tmpx
      }
      return(temp)
}
# negative log-likelihood of the GB2 distribution for the liability claims data
#  Inject on beta, use exponential function to ensure positivity;
"negll.GB2" <- function(parm, x) {
      alpha <- parm[1]
      gamma1 <- parm[2]
      gamma2 <- parm[3]
      int <- parm[4]
      bVTypeCar <- parm[5]
      bYear <- parm[6]
      bMaritalM <- parm[7]
      bSexM <- parm[8]
      bNCD <- parm[9]
      bComp <- parm[10]
      bAgeInsured <- parm[11]
      bClmCount <- parm[12]
```

```
    reg_eqn <- log(expose*Clm_Count) + int + bVTypeCar*VTypeCar + bYear*YearI +
          bMaritalM*MaritalM + bSexM*SexM + bNCD*NCD + bComp*Comp + bAgeInsured*AgeInsured;
    temp <- log(dGB2(x,alpha,exp(reg_eqn),gamma1,gamma2))
    result <- -sum(temp)
    return(result)
}

# now find the parameter estimates using (un)constrained optimization
# first set the initial parameter estimates
# The model is sensitive to small changes in the initial values

init.est <- c(0.07,220,150,-1,0.75,0.4,0.6,-0.2,-0.02,-0.1,0.1)
fit.GB2 <- optim(init.est, negll.GB2, NULL, x=Claims)
parm.hat <- fit.GB2$par
loglik.GB2 <- -fit.GB2$value
alpha.hat <- parm.hat[1]
gamma1.hat <- parm.hat[2]
gamma2.hat <- parm.hat[3]
int <- parm.hat[4]
bVTypeCar<-parm.hat[5]
bPremium<-parm.hat[6]
bMaritalM<-parm.hat[7]
bSexM<-parm.hat[8]
bNCD<-parm.hat[9]
bComp<-parm.hat[10]
bAgeInsured<-parm.hat[11]

# next estimate the standard errors.
library(nlme)
negll.GB2.Hess <- fdHess(parm.hat, negll.GB2, x=Claims)
inv.GB2.Hess <- solve(negll.GB2.Hess$Hessian)
parm.se <- sqrt(diag(inv.GB2.Hess))

# put together the model with the est, se, t, pval, AIC, BIC

dfe <- length(Claims)-length(parm.hat);
t_ratio<-parm.hat/parm.se;
##test if diff. from 1 t_ratio[1:3]<-(parm.hat[1:3]-1)/parm.se[1:3];
pval <- pf(t_ratio*t_ratio,df1=1,df2=dfe,lower.tail=F);
output <- cbind(parm.hat,parm.se,t_ratio,pval)
```

```
output <- round(output,digits=4)
rownames(output)<- c("alpha",
      "gamma1",
      "gamma2",
      "int",
      "bVTypeCar",
      "bYear",
      "bMaritalM",
      "bSexM",
      "bNCD",
      "bComp",
      "bAgeInsured")

colnames(output)<- c("estimate", "std error", "t-val","Pr>|t|");
cat("",fill=T)
print(output)
AIC<- 2*negll.GB2(parm.hat,Claims) + 2*length(parm.hat);
BIC<- 2*negll.GB2(parm.hat,Claims) + log(length(Claims))*length(parm.hat);
cat("",fill=T)
cat(paste("AIC estimated at ",round(AIC,2)),fill=T);
cat(paste("BIC estimated at ",round(BIC,2)),fill=T);

# do some graphical tests
par(mfrow=c(2,2),font.main=11,font.lab=11)

# obtain the residuals - remove effect of the regressors
#  first create the design matrix of regressors;
design.mat <-
matrix(c(rep(1,length(Claims)),VTypeCar,YearI,MaritalM,SexM,NCD,Comp,AgeInsured),length(Claims),8,byrow=F)
# Then obtain the residuals, R;
R <- (Claims*exp(t(-parm.hat[4:11]%*%t(design.mat))))/(expose*Clm_Count);

# the density plot of the residuals
hist(R,breaks=30,prob=T,ylim=c(0,0.0004))
lines(sort(R),dGB2(sort(R),alpha=alpha.hat,beta=1,gamma1=gamma1.hat,gamma2=gamma2.hat))

# the empirical distribution function
samp.pct <- (1:length(R)-0.5)/length(R)
plot(sort(R),samp.pct,type="s",xlab="Residuals",ylab="cdf",main="Empirical CDF")
abline(1,0,lty=3)
```

```
abline(0,0,lty=3)
lines(sort(R),pGB2(sort(R),alpha=alpha.hat,beta=1,gamma1=gamma1.hat,gamma2=gamma2.hat))

# the quantile-quantile plot
plot(qGB2(samp.pct,alpha=alpha.hat,beta=1,gamma1=gamma1.hat,gamma2=gamma2.hat),sort(R),
   xlab="theoretical quantiles",ylab="sample quantiles",main="Q-Q plot",cex=0.55,xlim=c(0,50000),ylim=c(0,50000))
abline(0,1)

# the probability-probability plot
plot(pGB2(sort(R),alpha=alpha.hat,beta=1,gamma1=gamma1.hat,gamma2=gamma2.hat),samp.pct,
   xlab="theoretical probability",ylab="sample probability",main="P-P plot",cex=0.55)
abline(0,1)

# the SBC (Schwarz Bayesian Criterion) test
numb.par <- length(parm.hat)
SBC.stat <- loglik.GB2 - (numb.par/2)*log(length(Claims))
out <- rbind(loglik.GB2,SBC.stat)
colnames(out) <- c("value")
rownames(out) <- c("Neg Log Likelihood","SBC criterion")
print(out)
```

```
OUTPUT

> source("C:\\Users\\valdezea\\Desktop\\Bogota_Colombia 2014\\GB2EstimationCode-
Dec7\\FitGB2.CO35ClaimsData.BetaRegression3.R")

             estimate std error   t-val Pr>|t|
alpha          0.0645       NaN     NaN    NaN
gamma1       220.2055       NaN     NaN    NaN
gamma2       164.6316       NaN     NaN    NaN
int            1.3238       NaN     NaN    NaN
bVTypeCar     -0.0236       NaN     NaN    NaN
bYear         -0.1364    0.0641 -2.1275 0.0335
bMaritalM     -0.6893    0.0949 -7.2606 0.0000
bSexM          0.0357       NaN     NaN    NaN
bNCD          -0.0111    0.0014 -7.9653 0.0000
bComp          0.3672    0.0344 10.6651 0.0000
bAgeInsured    0.0568    0.0020 29.0699 0.0000
```

```
AIC estimated at  49528.5
BIC estimated at  49593.25
                       value
Neg Log Likelihood -24753.25
SBC criterion      -24796.63
Warning message:
In sqrt(diag(inv.GB2.Hess)) : NaNs produced
```