Code

Illustrative code is attached.

Simulation

The following is the code for the simulation study for lightning perils. Both the regression approach and the maximum likelihood approach for deductibles ratemaking are implemented here.

```
# Figure 1 and Figure 2.

library(actuar)
library(tweedie)
library(MASS)
library(stats)
library(stats4)
library(statmod)

CreateData <- function(B) {
  idx <- function(i,Ns){
    if(Ns[i]==0) {return(NA)} else
      if(i==1){return(1:Ns[1])} else
      { return((sum(Ns[1:(i-1)])+1):
                 (sum(Ns[1:(i-1)])+Ns[i]))}
  }
  data <- data.frame(1:B); colnames(data) <- "i"
  data$d <- as.matrix(t(rmultinom(
    B, size=1, prob=c( 0.46153846, 0.21514423, 0.11778846,
                       0.09495192, 0.01081731,
                       sum(0.01081731, 0.05649038,
                           0.02163462, 0.01081731))))) %*%
    c(500, 1000, 2500,
      5000, 10000, 25000)
  gg <- function(x,d) {ifelse(x>d,x-d,0)}
  data$N <- rpois(n=B,lambda=N.mu)
  claims <- data.frame(1:sum(data$N));
  colnames(claims) <- "claim"
  claims$Y <- rpareto(n=sum(data$N),
                      shape=Yalpha, scale=Y.mu*(Yalpha-1))
  temp2 <- NULL
  for (i in 1:B) {
    temp <- idx(i,data$N)
    if(!any(is.na(temp))) {
      temp2 <-
        rbind(temp2,cbind(rep(i,length(temp)),temp))
    }
  }
  colnames(temp2) <- c("i","claim")
```

```
  temp2 <- data.frame(temp2)
  claims <- merge(claims,temp2,by.x="claim",
                  by.y="claim",all.x=T)
  claims <- merge(claims,data[,c("i","d")],
                  by.x="i",by.y="i",all.x=T)
  claims <- claims[,c("claim","i","d","Y")]
  head(claims)
  claims$Yc <- gg(claims$Y,claims$d)
  claimstrunc <- subset(claims,Yc>0); nrow(claimstrunc)
  data <- merge(data,aggregate(Y~i,data=claimstrunc,FUN=length),
                by="i",all.x=T)
  colnames(data)[ncol(data)] <- "Nstar"
  data[,c("Nstar")][is.na(data[,c("Nstar")])] <- 0
  data <- merge(data,aggregate(Y~i,data=claimstrunc,FUN=sum),
                by="i",all.x=T)
  colnames(data)[ncol(data)] <- "Claim"
  data[,c("Claim")][is.na(data[,c("Claim")])] <- 0
  data$lnd <- log(data$d)
  claimstrunc$lnd <- log(claimstrunc$d)
  data$lndlambda <- log(data$d+Y.mu*(Yalpha-1))
  claimstrunc$lndlambda <- log(claimstrunc$d + Y.mu*(Yalpha-1))
  l <- list()
  l$claimstrunc <- claimstrunc
  l$data <- data
  l$claims <- claims
  return(l)
}

GetRels <- function() {

  sim.pois.reg<-glm(Nstar ~ lnd, data=out$data,
                    family=poisson(link = "log"))
  sim.gamma.reg<-glm(Yc ~ lnd, data=out$claimstrunc,
                     family=Gamma(link = "log"))
  lik <- function(param) {
    beta   <- param[1:2]
    a      <- param[3]
    X      <- as.matrix(cbind(1,out$claimstrunc$lnd))
    mean   <- exp(X%*%beta)
    lambda <- mean * (a-1)
    if(any(lambda<=0)){return(10^10)}
    if(a<=0){return(10^10)}
    dparet <- a * lambda^a / (lambda + out$claimstrunc$Yc)^(a+1)
    return(-sum(log(dparet)))
  }
  op <- optim(c(0.1,0.1,1.1),lik,method="L-BFGS-B",hessian=T)
  out$data$NstarPred <- exp(as.matrix(cbind(1,out$data$lnd)) %*%
                            coef(sim.pois.reg))
```

```
  out$data$Ymu <- Y.mu
  out$data$Yalpha <- Yalpha
  out$data$Yd <- sapply(1:B,function(i){integrate(function(d){
    1-ppareto(d,shape=Yalpha,scale=Y.mu*(Yalpha-1))},
    lower=out$data[i,]$d, upper=Inf)$val})
  out$data$Fd  <- ppareto(out$data$d,
                        shape=Yalpha,scale=Y.mu*(Yalpha-1))
  out$data$YstarTrue <- out$data$Yd/(1-out$data$Fd)
  out$data$YstarPred <- exp(as.matrix(cbind(1,out$data$lnd)) %*%
                            coef(sim.gamma.reg))
  out$data$YstarPred2 <- exp(as.matrix(cbind(1,out$data$lnd)) %*%
                             op$par[1:2])*(op$par[3]-1)
  out$data$NstarPred <- exp(as.matrix(cbind(1,out$data$lnd)) %*%
                            coef(sim.pois.reg))
  out2 <- list()
  out2$claimstrunc <- out$claimstrunc
  out2$claims <- out$claims
  out2$data <- out$data
  out2$coefpois <- coef(sim.pois.reg)
  out2$coefgamma <- coef(sim.gamma.reg)
  out2$coefpareto <- op$par
  out2$X <- as.matrix(cbind(1,out$data$lnd))

  deds <- seq(500,25000,length=50)
  Xs <- NULL
  for(i in 1:length(deds)) {
    Xs <- rbind(Xs,c(1,deds[i],deds[i]^2,deds[i]^3))
  }
  rels1 <- (deds/500)^(out2$coefpois[2]+out2$coefpareto[2])
  # rels2 <- (deds/500)^(coef(sim.tweedie.reg)[2])
  rels3 <- NULL
  denominator <- numerator <- numeric(length(deds))
  scales <- rep(Y.mu * (Yalpha - 1),50)
  for (i in 1:50) {
    for (dd in 1:length(deds)) {
      numerator[dd] <- integrate(function(d){
        1-ppareto(d,shape=Yalpha,scale=scales[i])},
        lower=deds[dd],upper=Inf)$value
      denominator[dd] <- integrate(function(d){
        1-ppareto(d,shape=Yalpha,scale=scales[i])},
        lower=500,upper=Inf)$value
    }
    newrel <- numerator/denominator
    rels3 <- rbind(rels3,newrel)
  }
  list(rels1=rels1,rels3=rels3)
}
```

```
GetRelsNew <- function() {

  sim.pois.reg<-glm(Nstar ~ lndlambda, data=out$data,
                    family=poisson(link = "log"))
  sim.gamma.reg<-glm(Yc ~ lndlambda, data=out$claimstrunc,
                     family=Gamma(link = "log"))
  lik <- function(param) {
    beta   <- param[1:2]
    a      <- param[3]
    X      <- as.matrix(cbind(1,out$claimstrunc$lndlambda))
    mean   <- exp(X%*%beta)
    lambda <- mean * (a-1)
    if(any(lambda<=0)){return(10^10)}
    if(a<=0){return(10^10)}
    dparet <- a * lambda^a / (lambda + out$claimstrunc$Yc)^(a+1)
    return(-sum(log(dparet)))
  }
  op <- optim(c(0.1,0.1,1.1),lik,method="L-BFGS-B",hessian=T)
  out$data$NstarPred <- exp(as.matrix(cbind(1,out$data$lndlambda)) %*%
                             coef(sim.pois.reg))
  out$data$Ymu <- Y.mu
  out$data$Yalpha <- Yalpha
  out$data$Yd <- sapply(1:B,function(i){integrate(function(d){
    1-ppareto(d,shape=Yalpha,scale=Y.mu*(Yalpha-1))},
    lower=out$data[i,]$d, upper=Inf)$val})
  out$data$Fd  <- ppareto(out$data$d,
                         shape=Yalpha,scale=Y.mu*(Yalpha-1))
  out$data$YstarTrue <- out$data$Yd/(1-out$data$Fd)
  out$data$YstarPred <- exp(as.matrix(cbind(1,out$data$lndlambda)) %*%
                             coef(sim.gamma.reg))
  out$data$YstarPred2 <- exp(as.matrix(cbind(1,out$data$lndlambda)) %*%
                              op$par[1:2])*(op$par[3]-1)
  out$data$NstarPred <- exp(as.matrix(cbind(1,out$data$lndlambda)) %*%
                             coef(sim.pois.reg))
  out2 <- list()
  out2$claimstrunc <- out$claimstrunc
  out2$claims <- out$claims
  out2$data <- out$data
  out2$coefpois <- coef(sim.pois.reg)
  out2$coefgamma <- coef(sim.gamma.reg)
  out2$coefpareto <- op$par
  out2$X <- as.matrix(cbind(1,out$data$lndlambda))
  lines(density(rpareto(
    B,shape=Yalpha,scale=exp(
      out2$X%*%out2$coefpareto[1:2])*(Yalpha-1))))
  deds <- seq(500,25000,length=50)
  Xs <- NULL
```

```
  for(i in 1:length(deds)) {
    Xs <- rbind(Xs,c(1,deds[i],deds[i]^2,deds[i]^3))
  }
  rels1 <- ((deds+Y.mu*(Yalpha-1))/(500+Y.mu*(Yalpha-1)))^(out2$coefpois[2]+
                    out2$coefpareto[2])
  rels3 <- NULL
  denominator <- numerator <- numeric(length(deds))
  scales <- rep(Y.mu * (Yalpha - 1),50)
  for (i in 1:50) {
    for (dd in 1:length(deds)) {
      numerator[dd] <- integrate(function(d){
        1-ppareto(d,shape=Yalpha,scale=scales[i])},
        lower=deds[dd],upper=Inf)$value
      denominator[dd] <- integrate(function(d){
        1-ppareto(d,shape=Yalpha,scale=scales[i])},
        lower=500,upper=Inf)$value
    }
    newrel <- numerator/denominator
    rels3 <- rbind(rels3,newrel)
  }

  list(rels1=rels1,rels3=rels3)
}


# Standard Parametrization

Y.mu <- 11087.29
Yalpha <- 2.5534937
N.mu <- 1
B <- 10000
set.seed(12345)
out <- CreateData(B)
out1 <- GetRels()

Y.mu <- 2000
Yalpha <- 2.5534937
N.mu <- 1
B <- 10000
set.seed(12345)
out <- CreateData(B)
out2 <- GetRels()

Y.mu <- 500
Yalpha <- 2.5534937
N.mu <- 1
B <- 10000
set.seed(12345)
```

```
out <- CreateData(B)
out3 <- GetRels()


# Figure 1. Relativities From Regression Using ln(d)

par(mfrow=c(1,3))
deds <- seq(500,25000,length=50)

plot(deds,out1$rels1,type="l", ylim=c(0,1),
     lty=2, lwd=3, xlab="Deductible", ylab="Relativity", col="black",
     main="E[Y]=11,087")
for(i in 1:10) {lines(deds,out1$rels3[1,],lty=1,lwd=1,col="black")}

plot(deds,out2$rels1,type="l", ylim=c(0,1),
     lty=2, lwd=3, xlab="Deductible", ylab="Relativity", col="black",
     main="E[Y]=2,000")
for(i in 1:10) {lines(deds,out2$rels3[1,],lty=1,lwd=1,col="black")}

plot(deds,out3$rels1,type="l", ylim=c(0,1),
     lty=2, lwd=3, xlab="Deductible", ylab="Relativity", col="black",
     main="E[Y]=500")
for(i in 1:10) {lines(deds,out3$rels3[1,],lty=1,lwd=1,col="black")}


legend("topright",c(
  "True Relativity",
  "Frequency-Severity Relativity"),
  lty=c(1,2), lwd=c(1,3), cex=1.5, col=c("black","black"))


# New Parametrization

Y.mu <- 11087.29
Yalpha <- 2.5534937
N.mu <- 1
B <- 10000
set.seed(12345)
out <- CreateData(B)
outnew1 <- GetRelsNew()

Y.mu <- 2000
Yalpha <- 2.5534937
N.mu <- 1
B <- 10000
set.seed(12345)
out <- CreateData(B)
outnew2 <- GetRelsNew()
```

```
Y.mu <- 500
Yalpha <- 2.5534937
N.mu <- 1
B <- 10000
set.seed(12345)
out <- CreateData(B)
outnew3 <- GetRelsNew()


# Figure 2. Relativities From Regression Using ln(d+lambda)

par(mfrow=c(1,3))
deds <- seq(500,25000,length=50)

plot(deds,outnew1$rels1,type="l", ylim=c(0,1),
     lty=2, lwd=2, xlab="Deductible", ylab="Relativity", col="black",
     main="E[Y]=11,087")
for(i in 1:10) {lines(deds,out1$rels3[1,],lty=1,lwd=1,col="black")}

plot(deds,outnew2$rels1,type="l", ylim=c(0,1),
     lty=2, lwd=2, xlab="Deductible", ylab="Relativity", col="black",
     main="E[Y]=2,000")
for(i in 1:10) {lines(deds,out2$rels3[1,],lty=1,lwd=1,col="black")}

plot(deds,outnew3$rels1,type="l", ylim=c(0,1),
     lty=2, lwd=2, xlab="Deductible", ylab="Relativity", col="black",
     main="E[Y]=500")
for(i in 1:10) {lines(deds,out3$rels3[1,],lty=1,lwd=1,col="black")}


legend("topright",c(
  "True Relativity",
  "Frequency-Severity Relativity"),
  lty=c(1,2), lwd=c(1,3), cex=1.5, col=c("black","black"))
```

Likelihood by Peril

The following is the code for truncated maximum likelihood estimation for the lightning peril type using the exponential, gamma, and Pareto distribution. This provides the practitioner with an idea of how truncated maximum likelihood estimation can be performed.

```
library(BB)
library(MASS)
library(pscl)
library(actuar)

# Figure 3. Histogram of lightBC$LossBeforeDeductible
# (Fitted Distributions of Table 12 Overlayed)

# Table 2. Coefficients from Regression Approach
# (Lightning Peril Type Shown for Illustration)

# Poisson
regBC.pois.light.ded<-glm(count2light ~ CoverageBC+lnDeductBC,
                          data=freqinBCnew,family=poisson(link = "log"))
summary(regBC.pois.light.ded)
# (Intercept) -9.75691    0.34380  -28.38   <2e-16 ***
# CoverageBC   0.72915    0.02723   26.78   <2e-16 ***
# lnDeductBC  -0.82191    0.04276  -19.22   <2e-16 ***
# AIC: 3791.1

# Gamma
regBC.gamma.light.ded<-glm(LossAfterDeduct ~ CoverageBC+lnDeductBC,
                           data=lightBCtrunc,family=Gamma(link = "log"))
summary(regBC.gamma.light.ded); gamma.dispersion(regBC.gamma.light.ded)
# (Intercept) 11.71305    0.77493  15.115  < 2e-16 ***
# CoverageBC   0.22590    0.05963   3.788 0.000164 ***
# lnDeductBC   0.48943    0.09730   5.030 6.19e-07 ***
# Dispersion: 1.445765
# AIC: 14523


# Figure 4-5. (Shown in Next Section)

# Table 3-5, 6-11. (Shown in Next Section)

# Table 12. Exponential, Gamma, Pareto Model Coefficient Estimates
```

```
# Exponential Model
explik_light <- function(param) {
  x <- as.matrix(cbind(1,lightBCtrunc[,c("CoverageBC")]))
  mu <- exp(x %*% param)
  if(any(mu<=0)){return(10^20)}
  denom <- 1 - pexp(lightBCtrunc$DeductBC*1000000,rate=1/mu)
  if(any(denom<=0)){return(10^20)}
  ldenom <- log(denom)
  lik <- dexp( lightBCtrunc$LossAfterDeduct +
                  lightBCtrunc$DeductBC*1000000 , rate=1/mu)/denom
  llik <- log(lik)
  return(-sum(llik))
}
op.exp.light<-optim(c(5,1),explik_light,method=c("L-BFGS-B"),hessian=T)
op.exp.light<-BBoptim(op.exp.light$par,explik_light)
op.exp.light<-nlminb(op.exp.light$par,explik_light)
op.exp.light<-optim(op.exp.light$par,explik_light,method=c("L-BFGS-B"), hessian=T)
cbind(op.exp.light$par,sqrt(diag(solve(op.exp.light$hessian))))
op.exp.light$value


# Gamma Model
gammalikBCtrunc.light <- function(param) {
  xsev <- as.matrix(cbind(1,lightBCtrunc[,c("CoverageBC")]))
  mu <- exp(xsev %*% param[1:2])
  if(param[3]<=0) {return(10^20)}
  if(any(mu<=0)){return(10^20)}
  denom <- 1 - pgamma(lightBCtrunc$DeductBC*1000000,
                    shape=1/param[3], scale=mu*param[3])
  lik <- dgamma( lightBCtrunc$LossAfterDeduct +
                 lightBCtrunc$DeductBC*1000000 ,
               shape=1*param[3], scale=mu/param[3], log=T) -
                  log(denom)
  return(-sum(lik))
}
op.gamma.light<-optim(c(7,0.5,0.5),gammalikBCtrunc.light,
                            method=c("L-BFGS-B"),hessian=T)
op.gamma.light<-BBoptim(op.gamma.light$par,gammalikBCtrunc.light)
op.gamma.light<-nlminb(op.gamma.light$par,gammalikBCtrunc.light)
op.gamma.light<-optim(op.gamma.light$par,gammalikBCtrunc.light,
   method=c("L-BFGS-B"), hessian=T)
cbind(op.gamma.light$par,sqrt(diag(solve(op.gamma.light$hessian))))
op.gamma.light$value
```

```
# Pareto Model
likpareto3light_covar_ded <- function(param) {
  betasev <- param[1:3]
  a       <- param[4]
  xsev <- as.matrix(cbind(1,lightBCtrunc[,c("CoverageBC","lnDeductBC")]))
  lambda <- exp(xsev %*% betasev) *(a-1)
  if(any(lambda<=0)){return(10^10)}
  if(a<=0){return(10^10)}
  y <- lightBCtrunc$LossAfterDeduct + lightBCtrunc$DeductBC
  dpareto <- a * lambda^a / (lambda + y)^(a+1)
  dpareto <- dpareto / (1 - ppareto(lightBCtrunc$DeductBC,
                                  shape=a,scale=lambda))
  return(-sum(log(dpareto)))
}

op_pareto3light_covar_ded <- optim(c(5,1,1,1),
likpareto3light_covar_ded, method=c("L-BFGS-B"))
op_pareto3light_covar_ded<-BBoptim(
op_pareto3light_covar_ded$par,likpareto3light_covar_ded)
op_pareto3light_covar_ded<-nlminb(
op_pareto3light_covar_ded$par,likpareto3light_covar_ded)
op_pareto3light_covar_ded<-optim(
op_pareto3light_covar_ded$par,likpareto3light_covar_ded,
method=c("L-BFGS-B"), hessian=T)
cbind(op_pareto3light_covar_ded$par,
sqrt(diag(solve(op_pareto3light_covar_ded$hessian))))
op_pareto3light_covar_ded$value


# Note: Figures 6, 7, 8, 9, 10, 11, 12, 13, 14 are created using the above
# three fitted models for each of the nine peril types.
```

```
# Table 13. Poisson Frequency Model Coefficient Estimates
# (Lightning Peril Type Shown for Illustration)

v.exp.light <- 1 - pexp(
  freqinBCnew$DeductBC*1000000,
  rate=1/exp(as.matrix(cbind(1,freqinBCnew[,c("CoverageBC")]))%*%
             op.exp.light$par[1:2]))

v.gamma.light <- 1 - pgamma(
  freqinBCnew$DeductBC*1000000,
  shape=1/op.gamma.light$par[3], scale=
    exp(as.matrix(cbind(
      1,freqinBCnew[,c("CoverageBC")]))%*%
        op.gamma.light$par[1:2])*
    op.gamma.light$par[3])

v.pareto.light <- 1 - ppareto(
  freqinBCnew$DeductBC*1000000,
  shape=op_pareto3light_covar$par[3],scale=
    exp(as.matrix(cbind(1,freqinBCnew[,c("CoverageBC")]))%*%
          op_pareto3light_covar$par[1:2])*(op_pareto3light_covar$par[3]-1))

likpois.light.trunc.gamma<-function(param){
  x <- as.matrix(cbind(1,freqinBCnew[,c("CoverageBC")]))
  la <- exp(x %*% param + log(v.gamma.light))
  l <- -sum(log(dpois(freqinBCnew$count2light,
                      lambda=la)))
  return(l)
}
op.pois.light.gamma<-optim(c(0,0),likpois.light.trunc.gamma,
                                    method=c("BFGS"), hessian=T)
op.pois.light.gamma<-BBoptim(op.pois.light.gamma$par,likpois.light.trunc.gamma)
op.pois.light.gamma<-nlminb(op.pois.light.gamma$par,likpois.light.trunc.gamma)
op.pois.light.gamma<-optim(op.pois.light.gamma$par,likpois.light.trunc.gamma,
                                    method=c("L-BFGS-B"), hessian=T)
cbind(op.pois.light.gamma$par,sqrt(diag(solve(op.pois.light.gamma$hessian))))
op.pois.light.gamma$value # 1985.285
# [1,] -3.2419850 0.09578267
# [2,]  0.5184758 0.02495184

likpois.light.trunc.pareto<-function(param){
  x <- as.matrix(cbind(1,freqinBCnew[,c("CoverageBC")]))
  la <- exp(x %*% param + log(v.pareto.light))
  l <- -sum(log(dpois(freqinBCnew$count2light,
                      lambda=la)))
  return(l)
}
```

```
op.pois.light.pareto<-optim(c(0,0),likpois.light.trunc.pareto,
                                  method=c("BFGS"), hessian=T)
op.pois.light.pareto<-BBoptim(op.pois.light.pareto$par,likpois.light.trunc.pareto)
op.pois.light.pareto<-nlminb(op.pois.light.pareto$par,likpois.light.trunc.pareto)
op.pois.light.pareto<-optim(op.pois.light.pareto$par,likpois.light.trunc.pareto,
                                  method=c("L-BFGS-B"), hessian=T)
op.pois.light.pareto$value # 1948.613
cbind(op.pois.light.pareto$par,
      sqrt(diag(solve(op.pois.light.pareto$hessian))))
# [1,] -3.2447638 0.09721341
# [2,]  0.5604956 0.02539026

likpois.light.trunc.exp<-function(param){
  x <- as.matrix(cbind(1,freqinBCnew[,c("CoverageBC")]))
  la <- exp(x %*% param + log(v.exp.light))
  l <- -sum(log(dpois(freqinBCnew$count2light,
                      lambda=la)))
  return(l)
}
op.pois.light.exp<-optim(c(0,0),likpois.light.trunc.exp,
                              method=c("BFGS"), hessian=T)
op.pois.light.exp<-BBoptim(op.pois.light.exp$par,likpois.light.trunc.exp)
op.pois.light.exp<-nlminb(op.pois.light.exp$par,likpois.light.trunc.exp)
op.pois.light.exp<-optim(op.pois.light.exp$par,likpois.light.trunc.exp,
                              method=c("L-BFGS-B"), hessian=T)
op.pois.light.exp$value # 2090.805
cbind(op.pois.light.exp$par,sqrt(diag(solve(op.pois.light.exp$hessian))))
# [1,] -3.3998134 0.09323772
# [2,]  0.4839632 0.02416907
```

Main Code

Below is the code for all outputs shown in the main text. Data generation code has been omitted; however, the variable names are self-explanatory.

```
claiminBC <- read.csv("data/claiminBC.csv")
claiminBCtrunc <- read.csv("data/claiminBCtrunc.csv")
claimoutBC <- read.csv("data/claimoutBC.csv")
claimoutBCtrunc <- read.csv("data/claimoutBCtrunc.csv")
freqinBCnew <- read.csv("data/freqinBCnew.csv")
freqoutBCnew <- read.csv("data/freqoutBCnew.csv")
claiminBC.low <- read.csv("data/claiminBC.low.csv")
claiminBC.medium <- read.csv("data/claiminBC.medium.csv")
claiminBC.high <- read.csv("data/claiminBC.high.csv")
claiminBCtrunc.low <- read.csv("data/claiminBCtrunc.low.csv")
claiminBCtrunc.medium <- read.csv("data/claiminBCtrunc.medium.csv")
claiminBCtrunc.high <- read.csv("data/claiminBCtrunc.high.csv")

fireBC <- subset(claiminBC,PerilFire==1); nrow(fireBC) # 171
fireBCtrunc <- subset(claiminBCtrunc,PerilFire==1); nrow(fireBCtrunc) # 141
fireoutBC <- subset(claimoutBC,PerilFire==1); nrow(fireoutBC) # 46
fireoutBCtrunc <- subset(claimoutBCtrunc,PerilFire==1); nrow(fireoutBCtrunc) # 35
vandalismBC <- subset(claiminBC,PerilVandalism==1); nrow(vandalismBC) # 1774
vandalismBCtrunc <- subset(claiminBCtrunc,PerilVandalism==1);
nrow(vandalismBCtrunc) # 494
vandalismoutBC <- subset(claimoutBC,PerilVandalism==1);
nrow(vandalismoutBC) # 310
vandalismoutBCtrunc <- subset(claimoutBCtrunc,PerilVandalism==1);
nrow(vandalismoutBCtrunc) # 52
lightBC <- subset(claiminBC,PerilLightning==1); nrow(lightBC) # 832
lightBCtrunc <- subset(claiminBCtrunc,PerilLightning==1);
nrow(lightBCtrunc) # 722
lightoutBC <- subset(claimoutBC,PerilLightning==1); nrow(lightoutBC) # 123
lightoutBCtrunc <- subset(claimoutBCtrunc,PerilLightning==1);
nrow(lightoutBCtrunc) # 103
windBC <- subset(claiminBC,PerilWind==1); nrow(windBC) # 296
windBCtrunc <- subset(claiminBCtrunc,PerilWind==1);
nrow(windBCtrunc) # 247
windoutBC <- subset(claimoutBC,PerilWind==1); nrow(windoutBC) # 107
windoutBCtrunc <- subset(claimoutBCtrunc,PerilWind==1);
nrow(windoutBCtrunc) # 90
hailBC <- subset(claiminBC,PerilHail==1); nrow(hailBC) # 76
hailBCtrunc <- subset(claiminBCtrunc,PerilHail==1);
nrow(hailBCtrunc) # 70
hailoutBC <- subset(claimoutBC,PerilHail==1); nrow(hailoutBC) # 18
hailoutBCtrunc <- subset(claimoutBCtrunc,PerilHail==1);
nrow(hailoutBCtrunc) # 16
vehicleBC <- subset(claiminBC,PerilVehicle==1); nrow(vehicleBC) # 852
```

```
vehicleBCtrunc <- subset(claiminBCtrunc,PerilVehicle==1);
nrow(vehicleBCtrunc) # 632
vehicleoutBC <- subset(claimoutBC,PerilVehicle==1); nrow(vehicleoutBC) # 227
vehicleoutBCtrunc <- subset(claimoutBCtrunc,PerilVehicle==1);
nrow(vehicleoutBCtrunc) # 154
waternwBC <- subset(claiminBC,PerilWaterNW==1); nrow(waternwBC) # 202
waternwBCtrunc <- subset(claiminBCtrunc,PerilWaterNW==1);
nrow(waternwBCtrunc) # 158
waternwoutBC <- subset(claimoutBC,PerilWaterNW==1); nrow(waternwoutBC) # 67
waternwoutBCtrunc <- subset(claimoutBCtrunc,PerilWaterNW==1);
nrow(waternwoutBCtrunc) # 41
waterwBC <- subset(claiminBC,PerilWaterW==1); nrow(waterwBC) # 426
waterwBCtrunc <- subset(claiminBCtrunc,PerilWaterW==1);
nrow(waterwBCtrunc) # 352
waterwoutBC <- subset(claimoutBC,PerilWaterW==1); nrow(waterwoutBC) # 38
waterwoutBCtrunc <- subset(claimoutBCtrunc,PerilWaterW==1);
nrow(waterwoutBCtrunc) # 27
miscBC <- subset(claiminBC,PerilMisc==1); nrow(miscBC) # 362
miscBCtrunc <- subset(claiminBCtrunc,PerilMisc==1);
nrow(miscBCtrunc) # 273
miscoutBC <- subset(claimoutBC,PerilMisc==1); nrow(miscoutBC) # 103
miscoutBCtrunc <- subset(claimoutBCtrunc,PerilMisc==1);
nrow(miscoutBCtrunc) # 62

x.low <- as.matrix(rep(1,nrow(claiminBC.low)))
x.medium <- as.matrix(cbind(1,claiminBC.medium[,"CoverageBC"]))
x.high <- as.matrix(cbind(1,claiminBC.high[,"CoverageBC"]))
x.low.trunc <- as.matrix(rep(1,nrow(claiminBCtrunc.low)))
x.medium.trunc <- as.matrix(cbind(1,claiminBCtrunc.medium[,"CoverageBC"]))
x.high.trunc <- as.matrix(cbind(1,claiminBCtrunc.high[,"CoverageBC"]))


# Table 3, 4, 5 (Shown Within Code)

# Figure 4, 5 (Shown Within Code)

# Table 6. Summary Statistics of BC (Primary Coverage) Claims
cbind(round(aggregate(DeductBC*1000000 ~ Year, data=freqinBCnew, mean),0),
      round(aggregate(count1 ~ Year, data=freqinBCnew, mean),3),
      round(aggregate(count2 ~ Year, data=freqinBCnew, mean),3),
      round(aggregate(Claim1 ~ Year, data=freqinBCnew, sum),0),
      round(aggregate(Claim2 ~ Year, data=freqinBCnew, sum),0),
      round(aggregate(Claim2 ~ Year, data=freqinBCnew, length),0))

# Table 7. Summary Statistics of BC Claims by Deductible
round(aggregate(count1 ~ DeductBC, data=freqinBCnew, mean),3)
round(aggregate(count2 ~ DeductBC, data=freqinBCnew, mean),3)
round(aggregate(Claim1 ~ DeductBC, data=freqinBCnew, mean),0)
```

```
round(aggregate(Claim2 ~ DeductBC, data=freqinBCnew, mean),0)
round(aggregate(Claim2 ~ DeductBC, data=freqinBCnew, length),3)


# Table 8. BC Policies, 2006-2010
round(summary(freqinBCnew$CoverageBC),3)

# Table 9. BC Claims, 2006-2010
summary(claiminBC$CoverageBC)

# Table 10. BC Losses, 2006-2010
summary(claiminBCtrunc$CoverageBC)

# Table 11. Peril Types of BC Losses
aggregate(LossBeforeDeductible ~ PerilSimple, data=claiminBC, sum)

# Table 12, 13: (See Likelihood for Peril Section)

# Table 14. Average Loss and Claim Severity by Peril
# Summarize: claiminBC.low, claiminBC.medium, claiminBC.high (Omitted)

# Table 15. Poisson-Gamma Regression (Model A)

regBC.pois<-glm(count2 ~ CoverageBC+lnDeductBC+NoClaimCreditBC+
                  TypeCity+TypeCounty+TypeMisc+TypeSchool+TypeTown,
                  data=freqinBCnew,family=poisson(link = "log"))
regBC.gamma<-glm(LossAfterDeduct ~ CoverageBC + TypeCity + TypeCounty +
                   TypeMisc + TypeSchool + TypeTown,
                 data=claiminBCtrunc, family=Gamma(link = "log"),
                 control=list(epsilon=1e-08,maxit=50,trace=FALSE))

summary(regBC.pois)

summary(regBC.gamma,dispersion=gamma.dispersion(regBC.gamma))

regBC.pois2<-glm(count2 ~ NoClaimCreditBC+
                   TypeCity+TypeCounty+TypeMisc+TypeSchool+TypeTown,
                 data=freqinBCnew,family=poisson(link = "log"),
                 offset = offset)
regBC.gamma2<-glm(LossAfterDeduct ~ TypeCity + TypeCounty +
                    TypeMisc + TypeSchool + TypeTown,
                  data=claiminBCtrunc, family=Gamma(link = "log"),
                  offset=offset,
                  control=list(epsilon=1e-08,maxit=50,trace=FALSE))
```

```
# Table 16. Regression Approach With Offset

summary(regBC.pois2)

summary(regBC.gamma2,dispersion=gamma.dispersion(regBC.gamma))

# Predicted scores using the two regression approaches.
regBC.pois.score <-
  exp(as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC","lnDeductBC","NoClaimCreditBC","TypeCity",
    "TypeCounty","TypeMisc","TypeSchool","TypeTown")])) %*% coef(regBC.pois))
regBC.gamma.score <-
  exp(as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC","TypeCity","TypeCounty","TypeMisc","TypeSchool",
    "TypeTown")])) %*% coef(regBC.gamma))
regBC.pois.nodeduct <-
  exp(as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC")],log(1/1000000),freqoutBCnew[,c(
      "NoClaimCreditBC","TypeCity","TypeCounty","TypeMisc",
      "TypeSchool","TypeTown")])) %*% coef(regBC.pois))


# Claim1: Aggregate claims before deductible
# Claim2: Aggregate claims after deductible
par(mfrow=c(1,2))
plot(log(regBC.pois.nodeduct * regBC.gamma.score+1),
     log(freqoutBCnew$Claim1+1), xlim=c(0,15),ylim=c(0,15),
     xlab="Losses Score", ylab="Losses"); abline(0,1)
cor(regBC.pois.nodeduct * regBC.gamma.score, freqoutBCnew$Claim1,
    method="spearman") # 0.4084289
plot(log(regBC.pois.score * regBC.gamma.score+1),
     log(freqoutBCnew$Claim2+1), xlim=c(0,15),ylim=c(0,15),
     xlab="Claims Score", ylab="Claims"); abline(0,1)
cor(regBC.pois.score * regBC.gamma.score, freqoutBCnew$Claim2,
    method="spearman") # 0.3922297


# Change deduct to any d.
# Model assumes deductible amounts in million
regBC.deduct.prem <- function(d) {
  sum(exp(as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC")],log(d/1000000),freqoutBCnew[,c(
      "NoClaimCreditBC","TypeCity","TypeCounty","TypeMisc",
      "TypeSchool","TypeTown")])) %*% coef(regBC.pois)) *
      regBC.gamma.score)
}
```

```
# MLE Approach

# Gamma Likelihood
gammalikBC <- function(param) {
  # print(param)
  xsev <- as.matrix(cbind(1,claiminBCtrunc[,c(
    "CoverageBC","TypeCity","TypeCounty",
    "TypeMisc","TypeSchool","TypeTown")]))
  betasev <- param[1:ncol(xsev)]
  phi <- 1/param[length(param)]
  if(phi<=0) {return(99999999999)}
  mu <- exp(xsev %*% betasev)
  if(any(mu<=0)){return(9999999999)}
  #llik <- dgamma( claiminBCtrunc$LossAfterDeduct ,
  #                shape=1/phi, scale=mu*phi , log=TRUE )
  ldenom <- 1 - pgamma(claiminBCtrunc$DeductBC*1000000,
                       shape=1/phi, scale=mu*phi)
  llik <- dgamma( claiminBCtrunc$LossAfterDeduct +
   claiminBCtrunc$DeductBC*1000000 ,
                  shape=1/phi, scale=mu*phi , log=TRUE ) -
    ifelse(ldenom==1,0.999999999,ldenom)
  return(-sum(llik))
  remove(llik,mu,phi,betasev,xsev)
}

op.gamma.trunc<-optim(c(regBC.gamma$coefficient,
                        gamma.dispersion(regBC.gamma)),
                        gammalikBC,method=c("Nelder-Mead"),hessian=T)
op.gamma.trunc<-BBoptim(op.gamma.trunc$par,gammalikBC)
op.gamma.trunc<-nlminb(op.gamma.trunc$par,gammalikBC)
op.gamma.trunc<-optim(op.gamma.trunc$par,gammalikBC,
                      method=c("L-BFGS-B"), hessian=T)

op.gamma.trunc$value # 35320.35


# Table 17. Poisson-Gamma Maximum Likelihood (Model B)

# Gamma Truncated Estimation
cbind(op.gamma.trunc$par,sqrt(diag(solve(op.gamma.trunc$hessian))))

# Poisson Truncated Estimation.
v.gamma <- 1 - pgamma(
  freqinBCnew$DeductBC*1000000,
  shape=1/op.gamma.trunc$par[8], scale=
    exp(as.matrix(cbind(1,freqinBCnew[,c(
      "CoverageBC","TypeCity","TypeCounty","TypeMisc",
      "TypeSchool","TypeTown")]))%*%op.gamma.trunc$par[1:7])*
```

```
    op.gamma.trunc$par[8])
likelihood.pois.trunc<-function(param){
  beta<-param
  x2 <- as.matrix(cbind(1,freqinBCnew[,c(
    "CoverageBC","NoClaimCreditBC",
    "TypeCity","TypeCounty","TypeMisc","TypeSchool","TypeTown")]))
  mean.pois.BC <- exp(x2 %*% beta + log(v.gamma))
  l <- -sum(log(dpois(freqinBCnew$count2,
                      lambda=mean.pois.BC)))
  return(l)
}
op.pois.trunc<-optim(c(0,0,0,0,0,0,0,0),
                     likelihood.pois.trunc,method=c("BFGS"), hessian=T)
op.pois.trunc<-BBoptim(op.pois.trunc$par,likelihood.pois.trunc)
op.pois.trunc<-nlminb(op.pois.trunc$par,likelihood.pois.trunc)
op.pois.trunc<-optim(op.pois.trunc$par,likelihood.pois.trunc,
                     method=c("L-BFGS-B"), hessian=T)
op.pois.trunc$value # 5133.119

cbind(op.pois.trunc$par,sqrt(diag(solve(op.pois.trunc$hessian))))


# Alternative way to obtain Table 17

# Offset Approach
regBC.pois3<-glm(count2 ~ CoverageBC+NoClaimCreditBC+
                   TypeCity+TypeCounty+TypeMisc+TypeSchool+TypeTown,
                 data=freqinBCnew, family=poisson(link = "log"),
                 offset=log(v.gamma))
summary(regBC.pois3)


# Table 4. Out-of-Sample (2011) Performance of Each Approach
# Apply Rating Formulas (Use Out-of-Sample Data)


mu.out <- exp(as.matrix(cbind(1,freqoutBCnew[,c(
  "CoverageBC","TypeCity","TypeCounty","TypeMisc",
  "TypeSchool","TypeTown")]))%*%op.gamma.trunc$par[1:7])

mod.out <- numeric(nrow(freqoutBCnew))
for (i in 1:length(mod.out)) {
  mod.out[i] <- integrate(function(d){1-pgamma(
    d,shape=1/op.gamma.trunc$par[8],
    scale=
      exp(as.matrix(cbind(1,freqoutBCnew[i,c(
        "CoverageBC","TypeCity","TypeCounty","TypeMisc",
        "TypeSchool","TypeTown")]))%*%op.gamma.trunc$par[1:7])*
```

```
      op.gamma.trunc$par[8])},
    lower=0, upper=freqoutBCnew$DeductBC[i]*1000000)$val
}

S <- exp(as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC","NoClaimCreditBC","TypeCity","TypeCounty","TypeMisc",
    "TypeSchool","TypeTown")])) %*% op.pois.trunc$par) * (mu.out)

SS <- exp(as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC","NoClaimCreditBC","TypeCity","TypeCounty","TypeMisc",
    "TypeSchool","TypeTown")])) %*% op.pois.trunc$par) * (mu.out - mod.out)

cor(SS, freqoutBCnew$Claim2, method="pearson") # 0.3358046

cor(SS, freqoutBCnew$Claim2, method="spearman") # 0.3846728


# Differences
cor(S - SS, freqoutBCnew$Claim1 -
freqoutBCnew$Claim2, method="pearson") # 0.7201838
cor(S - SS, freqoutBCnew$Claim1 -
freqoutBCnew$Claim2, method="spearman") # 0.4125738



# Severity Model by Peril type.
sevmodelBCgamma.low <- glm(LossBeforeDeductible~ 1,
                           data=claiminBC.low, family=Gamma(link = "log"))
sevmodelBCgamma.medium <- glm(LossBeforeDeductible~ 1 + CoverageBC,
                              data=claiminBC.medium, family=Gamma(link = "log"))
sevmodelBCgamma.high <- glm(LossBeforeDeductible~ 1 + CoverageBC,
                            data=claiminBC.high, family=Gamma(link = "log"))
summary(sevmodelBCgamma.low); gamma.dispersion(sevmodelBCgamma.low)
summary(sevmodelBCgamma.medium); gamma.dispersion(sevmodelBCgamma.medium)
summary(sevmodelBCgamma.high); gamma.dispersion(sevmodelBCgamma.high)


# Low/Medium/High Categorization
likgb2BC.low <- function(param) {
  k <- 1
  alpha <- param[1:k]
  sig <- param[k+1]
  a1 <- param[k+2]
  a2 <- param[k+3]
  meansevBC <- x.low * alpha
  if(sig<=0) {return(10^10)}
  if(a1<=0|a1>500) {return(10^10)}
  if(a2<=0|a2>500){return(10^10)}
```

```
  # if(a2-sig < 0){return(10^10)}
  sevBC <- claiminBC.low$LossBeforeDeductible
  yt <- (log(sevBC)-meansevBC)/sig
  logexpyt<-ifelse(yt>23,yt,log(1+exp(yt)))
  logdens <- a1*yt - log(sig) - log(beta(a1,a2)) - (a1+a2)*logexpyt
  return(-sum(logdens))
  remove(k,alpha,sig,a1,a2,meansevBC,sevBC,yt,logexpyt,logdens)
}

likgb2BC.medium <- function(param) {
  k <- 2
  alpha <- param[1:k]
  sig <- param[k+1]
  a1 <- param[k+2]
  a2 <- param[k+3]
  meansevBC <- x.medium %*% alpha
  if(sig<=0) {return(10^10)}
  if(a1<=0|a1>500) {return(10^10)}
  if(a2<=0|a2>500){return(10^10)}
  # if(a2-sig < 0){return(10^10)}
  sevBC <- claiminBC.medium$LossBeforeDeductible
  yt <- (log(sevBC)-meansevBC)/sig
  logexpyt<-ifelse(yt>23,yt,log(1+exp(yt)))
  logdens <- a1*yt - log(sig) - log(beta(a1,a2)) - (a1+a2)*logexpyt
  return(-sum(logdens))
  remove(k,alpha,sig,a1,a2,meansevBC,sevBC,yt,logexpyt,logdens)
}

likgb2BC.high <- function(param) {
  k <- 2
  alpha <- param[1:k]
  sig <- param[k+1]
  a1 <- param[k+2]
  a2 <- param[k+3]
  meansevBC <- x.high %*% alpha
  if(sig<=0) {return(10^10)}
  if(a1<=0|a1>500) {return(10^10)}
  if(a2<=0|a2>500){return(10^10)}
  # if(a2-sig < 0){return(10^10)}
  sevBC <- claiminBC.high$LossBeforeDeductible
  yt <- (log(sevBC)-meansevBC)/sig
  logexpyt<-ifelse(yt>23,yt,log(1+exp(yt)))
  logdens <- a1*yt - log(sig) - log(beta(a1,a2)) - (a1+a2)*logexpyt
  return(-sum(logdens))
  remove(k,alpha,sig,a1,a2,meansevBC,sevBC,yt,logexpyt,logdens)
}

indexperilBC<-c(which(names(claiminBC)=="TypeCity"),
```

```
                 which(names(claiminBC)=="TypeCounty"),
                 which(names(claiminBC)=="TypeMisc"),
                 which(names(claiminBC)=="TypeSchool"),
                 which(names(claiminBC)=="TypeTown"))
x.omega <- as.matrix(cbind(1,claiminBC[,indexperilBC]))
n.omega <- ncol(x.omega)

indexperilBC.out<-c(which(names(claimoutBC)=="TypeCity"),
                     which(names(claimoutBC)=="TypeCounty"),
                     which(names(claimoutBC)=="TypeMisc"),
                     which(names(claimoutBC)=="TypeSchool"),
                     which(names(claimoutBC)=="TypeTown"))
x.omega.out <- as.matrix(cbind(1,claimoutBC[,indexperilBC.out]))
n.omega.out <- ncol(x.omega.out)

indexperilBC.trunc<-c(which(names(claiminBCtrunc)=="TypeCity"),
                       which(names(claiminBCtrunc)=="TypeCounty"),
                       which(names(claiminBCtrunc)=="TypeMisc"),
                       which(names(claiminBCtrunc)=="TypeSchool"),
                       which(names(claiminBCtrunc)=="TypeTown"))
x.omega.trunc <- as.matrix(cbind(1,claiminBCtrunc[,indexperilBC.trunc]))
n.omega.trunc <- ncol(x.omega.trunc)

indexperil.freqinBC<-c(which(names(freqinBCnew)=="TypeCity"),
                        which(names(freqinBCnew)=="TypeCounty"),
                        which(names(freqinBCnew)=="TypeMisc"),
                        which(names(freqinBCnew)=="TypeSchool"),
                        which(names(freqinBCnew)=="TypeTown"))
x.omega.freqin <- as.matrix(cbind(1,freqinBCnew[,indexperil.freqinBC]))
n.omega.freqin <- ncol(x.omega.freqin)

indexperil.new<-c(which(names(freqinBCnew)=="TypeCity"),
                   which(names(freqinBCnew)=="TypeCounty"),
                   which(names(freqinBCnew)=="TypeMisc"),
                   which(names(freqinBCnew)=="TypeSchool"),
                   which(names(freqinBCnew)=="TypeTown"))
x.omega.new <- as.matrix(cbind(1,freqinBCnew[,indexperil.new]))
n.omega.new <- ncol(x.omega.new)

indexperilout.new<-c(which(names(freqoutBCnew)=="TypeCity"),
                      which(names(freqoutBCnew)=="TypeCounty"),
                      which(names(freqoutBCnew)=="TypeMisc"),
                      which(names(freqoutBCnew)=="TypeSchool"),
                      which(names(freqoutBCnew)=="TypeTown"))
x.omegaout.new <- as.matrix(cbind(1,freqoutBCnew[,indexperilout.new]))
n.omegaout.new <- ncol(x.omegaout.new)
```

```
op <- optim(c(sevmodelBCgamma.low$coefficients,0.1,0.1,0.1),
            likgb2BC.low,method=c("BFGS"),hessian=T)
gb2obj.low <- list(alpha=op$par[1],sig=op$par[1+1],
                   a1=op$par[1+2],a2=op$par[1+3],op=op)
cbind(gb2obj.low$op$par,sqrt(diag(solve(gb2obj.low$op$hessian))))

op <- optim(c(sevmodelBCgamma.medium$coefficients,0.1,0.1,0.1),
            likgb2BC.medium,method=c("BFGS"),hessian=T)
gb2obj.medium <- list(alpha=op$par[1:2],sig=op$par[2+1],
                      a1=op$par[2+2],a2=op$par[2+3],op=op)
cbind(gb2obj.medium$op$par,sqrt(diag(solve(gb2obj.medium$op$hessian))))

op <- optim(c(sevmodelBCgamma.high$coefficients,0.1,0.1,0.1),
            likgb2BC.high,method=c("BFGS"),hessian=T)
gb2obj.high <- list(alpha=op$par[1:2],
                    sig=op$par[2+1],
                    a1=op$par[2+2],
                    a2=op$par[2+3],
                    op=op)
cbind(gb2obj.high$op$par,sqrt(diag(solve(gb2obj.high$op$hessian))))


# Truncated GB2 by Peril Type.
likgb2BC.simp <- function(param) {

  k1 <- 1; k2 <- 2; k3 <- 2;
  alphaL <- param[1:k1]
  alphaM <- param[(k1+1):(k1+k2)]
  alphaH <- param[(k1+k2+1):(k1+k2+k3)]
  sig <- param[k1+k2+k3+1]
  a1  <- param[k1+k2+k3+2]
  a2  <- param[k1+k2+k3+3]
  sigL <- sigM <- sigH <- sig
  a1L <- a1M <- a1H <- a1
  a2L <- a2M <- a2H <- a2

  if(sigL<=0) {return(10^23)}
  if(sigM<=0) {return(10^23)}
  if(sigH<=0) {return(10^23)}
  if(a1L<=0|a1L>19) {return(10^23)}
  if(a2L<=0|a2L>19){return(10^23)}
  if(a1M<=0|a1M>19) {return(10^23)}
  if(a2M<=0|a2M>19){return(10^23)}
  if(a1H<=0|a1H>19) {return(10^23)}
  if(a2H<=0|a2H>19){return(10^23)}

  # Low
  meansevBCL <- x.low * alphaL
```

```
  sevBCL <- claiminBC.low$LossBeforeDeductible
  ytL <- (log(sevBCL)-meansevBCL)/sigL
  logexpytL <- ifelse(ytL>23,ytL,log(1+exp(ytL)))
  bL <- exp(meansevBCL)

  # Medium
  meansevBCM <- x.medium %*% alphaM
  sevBCM <- claiminBC.medium$LossBeforeDeductible
  ytM <- (log(sevBCM)-meansevBCM)/sigM
  logexpytM <- ifelse(ytM>23,ytM,log(1+exp(ytM)))
  bM <- exp(meansevBCM)

  # High
  meansevBCH <- x.high %*% alphaH
  sevBCH <- claiminBC.high$LossBeforeDeductible
  ytH <- (log(sevBCH)-meansevBCH)/sigH
  logexpytH <- ifelse(ytH>23,ytH,log(1+exp(ytH)))
  bH <- exp(meansevBCH)

  # Log Densities
  logdensL <- a1L*ytL - log(sigL) - log(beta(a1L,a2L)) -
    (a1L+a2L)*logexpytL
  logdensM <- a1M*ytM - log(sigM) - log(beta(a1M,a2M)) -
    (a1M+a2M)*logexpytM
  logdensH <- a1H*ytH - log(sigH) - log(beta(a1H,a2H)) -
    (a1H+a2H)*logexpytH

  return(-sum(logdensL)-sum(logdensM)-sum(logdensH))
}


likperil<-function(a){
  omega.medium    <-a[1:n.omega]
  omega.high      <-a[(n.omega+1):(2*n.omega)]
  exp1<-exp(x.omega %*% omega.medium)
  exp2<-exp(x.omega %*% omega.high)
  p0 <- 1 / (1 + exp1 + exp2)
  p1 <- exp1 / (1 + exp1 + exp2)
  p2 <- exp2 / (1 + exp1 + exp2)
  l <- -sum(log(pmax(p1*claiminBC$PerilMedium+
                      p2*claiminBC$PerilHigh+
                      p0*claiminBC$PerilLow,10^-10)))
  return(l)
}

op.simp <- optim(c(1,1,1,1,1, 0.1,0.1,0.1),likgb2BC.simp,method=c("BFGS"),hessian=T)
op.simp <- optim(op.simp$par,likgb2BC.simp,method="Nelder-Mead")
op.simp <- BBoptim(op.simp$par,likgb2BC.simp)
```

```
op.simp <- optim(op.simp$par,likgb2BC.simp,method="L-BFGS-B",hessian=T)
cbind(op.simp$par,sqrt(diag(solve(op.simp$hessian))))


# Mixture Likelihood for GB2
lik.simp.final <- function(param) {
  k1 <- 1; k2 <- 2; k3 <- 2
  alphaL <- param[1:k1]
  alphaM <- param[(k1+1):(k1+k2)]
  alphaH <- param[(k1+k2+1):(k1+k2+k3)]
  sig <- param[k1+k2+k3+1]
  a1  <- param[k1+k2+k3+2]
  a2  <- param[k1+k2+k3+3]
  omega.medium    <- param[(k1+k2+k3+3+1):(k1+k2+k3+3+n.omega)]
  omega.high      <- param[(k1+k2+k3+3+n.omega+1):(k1+k2+k3+3+2*n.omega)]
  sigL <- sigM <- sigH <- sig
  a1L <- a1M <- a1H <- a1
  a2L <- a2M <- a2H <- a2
  if(sigL<=0) {return(10^23)}
  if(sigM<=0) {return(10^23)}
  if(sigH<=0) {return(10^23)}
  if(a1L<=0|a1L>10) {return(10^23)}
  if(a2L<=0|a2L>10){return(10^23)}
  if(a1M<=0|a1M>10) {return(10^23)}
  if(a2M<=0|a2M>10){return(10^23)}
  if(a1H<=0|a1H>10) {return(10^23)}
  if(a2H<=0|a2H>10){return(10^23)}
  exp1.t <- exp(x.omega.trunc %*% omega.medium)
  exp2.t <- exp(x.omega.trunc %*% omega.high)
  pM.t <- exp1.t / (1 + exp1.t + exp2.t)
  pH.t <- exp2.t / (1 + exp1.t + exp2.t)
  pL.t <- 1 - pM.t - pH.t

  # Need these to calculate probability weights.
  Fd.L <- pgb2(claiminBCtrunc$DeductBC,
              shape1=1/sigL, scale=exp(alphaL),
              shape2=a1L,shape3=a2L)
  Fd.M <- pgb2(claiminBCtrunc$DeductBC,
              shape1=1/sigM, scale = exp(as.matrix(cbind(
                1,claiminBCtrunc[,"CoverageBC"])) %*% alphaM),
              shape2=a1M, shape3=a2M)
  Fd.H <- pgb2(claiminBCtrunc$DeductBC,
              shape1=1/sigH, scale = exp(as.matrix(cbind(
                1,claiminBCtrunc[,"CoverageBC"])) %*% alphaH),
              shape2=a1H, shape3=a2H)

  # Probability Weights.
  denom   <- pL.t/(1-Fd.L) + pM.t/(1-Fd.M) + pH.t/(1-Fd.H)
```

```
pLtrunc <- (pL.t/(1-Fd.L))/denom
pMtrunc <- (pM.t/(1-Fd.M))/denom
pHtrunc <- (pH.t/(1-Fd.H))/denom
# NOTE: Likelihood for Perils removed.
#       Just added the severity likelihoods.
# Severity Part:
# Low
meansevBCL <- x.low.trunc * alphaL
sevBCL <- claiminBCtrunc.low$LossAfterDeduct +
  claiminBCtrunc.low$DeductBC
ytL <- (log(sevBCL)-meansevBCL)/sigL
logexpytL <- ifelse(ytL>23,ytL,log(1+exp(ytL)))
bL <- exp(meansevBCL)
# Medium
meansevBCM <- x.medium.trunc %*% alphaM
sevBCM <- claiminBCtrunc.medium$LossAfterDeduct +
  claiminBCtrunc.medium$DeductBC
ytM <- (log(sevBCM)-meansevBCM)/sigM
logexpytM <- ifelse(ytM>23,ytM,log(1+exp(ytM)))
bM <- exp(meansevBCM)
# High
meansevBCH <- x.high.trunc %*% alphaH
sevBCH <- claiminBCtrunc.high$LossAfterDeduct +
  claiminBCtrunc.high$DeductBC
ytH <- (log(sevBCH)-meansevBCH)/sigH
logexpytH <- ifelse(ytH>23,ytH,log(1+exp(ytH)))
bH <- exp(meansevBCH)
# Log densities
logdensL <- a1L*ytL - log(sigL) - log(beta(a1L,a2L)) -
  (a1L+a2L)*logexpytL  +
  log(pLtrunc[claiminBCtrunc$PerilSimple=="PerilLow"]) -
  log(1 -Fd.L[claiminBCtrunc$PerilSimple=="PerilLow"])
logdensM <- a1M*ytM - log(sigM) - log(beta(a1M,a2M)) -
  (a1M+a2M)*logexpytM  +
  log(pMtrunc[claiminBCtrunc$PerilSimple=="PerilMedium"]) -
  log(1 -Fd.M[claiminBCtrunc$PerilSimple=="PerilMedium"])
logdensH <- a1H*ytH - log(sigH) - log(beta(a1H,a2H)) -
  (a1H+a2H)*logexpytH  +
  log(pHtrunc[claiminBCtrunc$PerilSimple=="PerilHigh"]) -
  log(1 -Fd.H[claiminBCtrunc$PerilSimple=="PerilHigh"])
return(-sum(logdensL)-sum(logdensM)-sum(logdensH))
remove(k1,k2,k3, alphaL,sigL,a1L,a2L, alphaM,sigM,a1M,a2M,
       alphaH,sigH,a1H,a2H, exp1.t, exp2.t, pM.t,pH.t,pL.t,
       Fd.L,Fd.M,Fd.H,denom, pLtrunc,pMtrunc,pHtrunc,
       meansevBCL,sevBCL,ytL,logexpytL,bL,
       meansevBCM,sevBCM,ytM,logexpytM,bM,
       meansevBCH,sevBCH,ytH,logexpytH,bH,
       logdensL,logdensM,logdensH)
```

```
  remove(omega.medium,omega.high,exp1,exp2,p0,p1,p2,
         Fd.low,Fd.medium,Fd.high,Fd,denom,p0trunc,p1trunc,p2trunc,l,
         f.low,f.medium,f.high)
}


op <- optim(c(sevmodelBCgamma.low$coefficients,0.1,0.1,0.1),
            likgb2BC.low,method=c("BFGS"),hessian=T)
gb2obj.low <- list(alpha=op$par[1],sig=op$par[1+1],
a1=op$par[1+2],a2=op$par[1+3],op=op)
op <- optim(c(sevmodelBCgamma.medium$coefficients,0.1,0.1,0.1),
            likgb2BC.medium,method=c("BFGS"),hessian=T)
gb2obj.medium <- list(alpha=op$par[1:2],sig=op$par[2+1],
                      a1=op$par[2+2],a2=op$par[2+3],op=op)
op <- optim(c(sevmodelBCgamma.high$coefficients,0.1,0.1,0.1),
            likgb2BC.high,method=c("BFGS"),hessian=T)
gb2obj.high <- list(alpha=op$par[1:2],sig=op$par[2+1],
                    a1=op$par[2+2],a2=op$par[2+3],op=op)
op.simp <- optim(c(1,1,1,1,1, 0.1,0.1,0.1),likgb2BC.simp,
                 method=c("BFGS"),hessian=T)
op.simp <- optim(op.simp$par,likgb2BC.simp,method="Nelder-Mead")
op.simp <- BBoptim(op.simp$par,likgb2BC.simp)
op.simp <- optim(op.simp$par,likgb2BC.simp,method="L-BFGS-B",hessian=T)
op.peril<-optim(rep(0,2*n.omega),likperil,method=c("BFGS"))
op.peril<-optim(op.peril$par,likperil,method=c("L-BFGS-B"),hessian=T)
op.simp.final <- optim(c(op.simp$par, op.peril$par),lik.simp.final,
                       method=c("BFGS"),  hessian=T)
op.simp.final <- optim(op.simp.final$par,lik.simp.final,
method="L-BFGS-B",hessian=T)
op.simp.final <- optim(op.simp.final$par,lik.simp.final,
method="Nelder-Mead",hessian=T)


# Table 18: GB2 Maximum Likelihood (Model C)
cbind(op.simp.final$par,sqrt(diag(solve(op.simp.final$hessian))))


# Predict Peril Types using Underlying Losses
exp1.in <- exp(x.omega[,1:n.omega] %*% op.peril$par[1:n.omega])
exp2.in <- exp(x.omega[,1:n.omega] %*% op.peril$par[(n.omega+1):(2*n.omega)])
pMedium.in <- exp1.in / (1 + exp1.in + exp2.in)
pHigh.in <- exp2.in / (1 + exp1.in + exp2.in)
pLow.in <- 1 - pMedium.in - pHigh.in


# Simple Model using Underlying Losses
ptemp <-
  pLow.in * pgb2(claiminBC$LossBeforeDeductible,
```

```
                 shape1 = 1/op.simp$par[6],
                 scale = exp(op.simp$par[1]),
                 shape2 = op.simp$par[7], shape3 = op.simp$par[8]) +
  pMedium.in * pgb2(claiminBC$LossBeforeDeductible,
                    shape1 = 1/op.simp$par[6],
                    scale = exp(
                      as.matrix(cbind(1,claiminBC[,"CoverageBC"])) %*%
                        c(op.simp$par[2:3])),
                    shape2 = op.simp$par[7], shape3 = op.simp$par[8]) +
  pHigh.in * pgb2(claiminBC$LossBeforeDeductible,
                  shape1 = 1/op.simp$par[6],
                  scale = exp(
                    as.matrix(cbind(1,claiminBC[,"CoverageBC"])) %*%
                      op.simp$par[4:5]),
                  shape2=op.simp$par[7], shape3=op.simp$par[8])


# Final Truncated Estimation
# Truncated Estimation of Peril and Severity Together
e1t <- exp(x.omega %*% op.simp.final$par[ 9:14])
e2t <- exp(x.omega %*% op.simp.final$par[15:20])
p1t <- e1t / (1 + e1t + e2t)
p2t <- e2t / (1 + e1t + e2t)
p0t <- 1 - p1t - p2t
ptemp.perilsev <-
  p0t * pgb2(claiminBC$LossBeforeDeductible,
             shape1 = 1/op.simp.final$par[6],
             scale = exp(op.simp.final$par[1]),
             shape2 = op.simp.final$par[7],
             shape3 = op.simp.final$par[8]) +
  p1t * pgb2(claiminBC$LossBeforeDeductible,
             shape1 = 1/op.simp.final$par[6],
             scale = exp(as.matrix(cbind(
               1,claiminBC[,"CoverageBC"])) %*%
                 op.simp.final$par[2:3]),
             shape2 = op.simp.final$par[7],
             shape3 = op.simp.final$par[8]) +
  p2t * pgb2(claiminBC$LossBeforeDeductible,
             shape1 = 1/op.simp.final$par[6],
             scale = exp(as.matrix(cbind(
               1,claiminBC[,"CoverageBC"])) %*%
                 op.simp.final$par[4:5]),
             shape2=op.simp.final$par[7],
             shape3=op.simp.final$par[8])
remove(e1t,e2t,p0t,p1t,p2t)


# Figure 5. Comparison of Severity Distributions (Q-Q Plot)
```

```
# Gamma
par(mfrow=c(1,3))
regBC.gamma.raw<-glm(LossBeforeDeductible ~
CoverageBC + TypeCity + TypeCounty +
                        TypeMisc + TypeSchool + TypeTown,
                      data=claiminBC, family=Gamma(link = "log"),
                      control=list(epsilon=1e-08,maxit=50,trace=FALSE))
qqnorm(qnorm(pgamma(claiminBC$LossBeforeDeductible,
                    shape=1/gamma.dispersion(regBC.gamma.raw),
                    scale=fitted(regBC.gamma.raw)*gamma.dispersion(regBC.gamma.raw))),
       xlim=c(-4,4), ylim=c(-4,4), main="Gamma Losses")
qqline(qnorm(pgamma(claiminBC$LossBeforeDeductible,
                    shape=1/gamma.dispersion(regBC.gamma.raw),
                    scale=fitted(regBC.gamma.raw)*gamma.dispersion(regBC.gamma.raw))),
       xlim=c(-4,4), ylim=c(-4,4))


# GB2 Mixture Using Underlying Losses
qqnorm(qnorm(ptemp),
       xlim=c(-4,4), ylim=c(-4,4), main="GB2 Losses")
qqline(qnorm(ptemp),
       xlim=c(-4,4), ylim=c(-4,4))


# GB2 Mixture Using Truncated Estimation
qqnorm(qnorm(ptemp.perilsev),
       xlim=c(-4,4), ylim=c(-4,4), main="GB2 Truncated Estimation")
qqline(qnorm(ptemp.perilsev), xlim=c(-4,4), ylim=c(-4,4))


# Kolmogorov-Smirnov Statistics
ks.test(qnorm(pgamma(claiminBC$LossBeforeDeductible,
                     shape=1/gamma.dispersion(regBC.gamma.raw),
                     scale=fitted(regBC.gamma.raw)*
                       gamma.dispersion(regBC.gamma.raw))),pnorm,0,1)
ks.test(qnorm(ptemp),pnorm,0,1)
ks.test(qnorm(ptemp.perilsev),pnorm,0,1)


# write.csv(op.pois.trunc$par,"data/par.pois.trunc.csv", row.names=F)
# write.csv(op.gamma.trunc$par,"data/par.gamma.trunc.csv", row.names=F)
# write.csv(op.simp.final$par,"data/par.simp.final.csv", row.names=F)

# write.csv(x.omega.new,"data/x.omega.new.csv", row.names=F)
# write.csv(x.omegaout.new,"data/x.omegaout.new.csv", row.names=F)
```

```
# 0-1 Inflated Model
zeroBC<-zeroinfl(FreqBC~ CoverageBC+NoClaimCreditBC+
                   TypeCity+TypeCounty+TypeMisc+TypeSchool+
                   TypeTown|1+ CoverageBC+NoClaimCreditBC,
                 data=freqinBCnew,dist = c("poisson"),link = c("logit"))
likelihood01<-function(a){
  n=8; m=3;
  beta<-a[1:n]
  gamma0<-a[(n+1):(n+m)]
  gamma1<-a[(n+m+1):(n+m+m)]
  z <- as.matrix(cbind(1,freqinBCnew[,c(
    "CoverageBC","NoClaimCreditBC")]))
  x2 <- as.matrix(cbind(1,freqinBCnew[,c(
    "CoverageBC","NoClaimCreditBC","TypeCity",
    "TypeCounty","TypeMisc","TypeSchool","TypeTown")]))
  zgamma0<-as.matrix(z)%*%gamma0
  zgamma1<-as.matrix(z)%*%gamma1
  pzeroBC<-1/(exp(-zgamma0)+exp(zgamma1-zgamma0)+1)
  poneBC<-1/(exp(-zgamma1)+exp(zgamma0-zgamma1)+1)
  x2beta<-as.matrix(x2)%*%beta
  meanpoissonBC<-exp(x2beta)
  l<--sum(log(pmax((pzeroBC)*(freqinBCnew$count1==0)+
                     poneBC*(freqinBCnew$count1==1)+(1-pzeroBC-poneBC)*
                     dpois(freqinBCnew$count1,meanpoissonBC),10^-10)))
  return(l)
}
opp01BCbfgs<-optim(c(zeroBC$coefficients$count,
                     zeroBC$coefficients$zero,0,0,0),
                   likelihood01,method=("BFGS"))
opp01BCnm<-optim(opp01BCbfgs$par,likelihood01,method=("Nelder-Mead"))
opp01BCBB<-BBoptim(opp01BCnm$par,likelihood01)
opp01BCnm<-optim(opp01BCBB$par,likelihood01,method=("Nelder-Mead"))
opp01BCbfgs<-optim(opp01BCnm$par,likelihood01,method=("BFGS"),hessian=T)
cbind(opp01BCbfgs$par,sqrt(diag(solve(opp01BCbfgs$hessian))))


# GB2 Model
e1t <- exp(x.omega.new %*% op.simp.final$par[ 9:14])
e2t <- exp(x.omega.new %*% op.simp.final$par[15:20])
p1t <- e1t / (1 + e1t + e2t)
p2t <- e2t / (1 + e1t + e2t)
p0t <- 1 - p1t - p2t
v.gb2 <- 1 - (
  p0t * pgb2(freqinBCnew$DeductBC*1000000,
             shape1 = 1/op.simp.final$par[6],
             scale = exp(op.simp.final$par[1]),
             shape2 = op.simp.final$par[7],
             shape3 = op.simp.final$par[8]) +
```

```
    p1t * pgb2(freqinBCnew$DeductBC*1000000,
               shape1 = 1/op.simp.final$par[6],
               scale = exp(as.matrix(cbind(
                 1,freqinBCnew[,"CoverageBC"])) %*%
                   op.simp.final$par[2:3]),
               shape2 = op.simp.final$par[7],
               shape3 = op.simp.final$par[8]) +
    p2t * pgb2(freqinBCnew$DeductBC*1000000,
               shape1 = 1/op.simp.final$par[6],
               scale = exp(as.matrix(cbind(
                 1,freqinBCnew[,"CoverageBC"])) %*%
                   op.simp.final$par[4:5]),
               shape2=op.simp.final$par[7],
               shape3=op.simp.final$par[8]))
remove(e1t,e2t,p0t,p1t,p2t)
mod.out2 <- numeric(nrow(freqoutBCnew))
for (i in 1:length(mod.out2)) {
  mod.out2[i] <- integrate(function(d){
    e1t <- exp(x.omegaout.new[i,] %*% op.simp.final$par[ 9:14])
    e2t <- exp(x.omegaout.new[i,] %*% op.simp.final$par[15:20])
    p1t <- e1t / (1 + e1t + e2t)
    p2t <- e2t / (1 + e1t + e2t)
    p0t <- 1 - p1t - p2t
    v <- 1 - (
      p0t * pgb2(d, shape1 = 1/op.simp.final$par[6],
                 scale = exp(op.simp.final$par[1]),
                 shape2 = op.simp.final$par[7],
                 shape3 = op.simp.final$par[8]) +
        p1t * pgb2(d, shape1 = 1/op.simp.final$par[6],
                   scale = exp(as.matrix(cbind(
                     1,freqoutBCnew[i,"CoverageBC"])) %*%
                       op.simp.final$par[2:3]),
                   shape2 = op.simp.final$par[7],
                   shape3 = op.simp.final$par[8]) +
        p2t * pgb2(d, shape1 = 1/op.simp.final$par[6],
                   scale = exp(as.matrix(cbind(
                     1,freqoutBCnew[i,"CoverageBC"])) %*%
                       op.simp.final$par[4:5]),
                   shape2=op.simp.final$par[7],
                   shape3=op.simp.final$par[8]))
    return(v)
    remove(e1t,e2t,p0t,p1t,p2t)
  },
  lower=0, upper=freqoutBCnew$DeductBC[i]*1000000)$val
}

ful.out2 <- numeric(nrow(freqoutBCnew))
ulim.out2 <- numeric(nrow(freqoutBCnew))
```

```
for (i in 1:length(mod.out2)) {
  e1t <- exp(x.omegaout.new[i,] %*% op.simp.final$par[ 9:14])
  e2t <- exp(x.omegaout.new[i,] %*% op.simp.final$par[15:20])
  p1t <- e1t / (1 + e1t + e2t)
  p2t <- e2t / (1 + e1t + e2t)
  p0t <- 1 - p1t - p2t
  ful.out2[i] <- integrate(function(d){
    v <- 1 - (
      p0t * pgb2(d, shape1 = 1/op.simp.final$par[6],
                 scale = exp(op.simp.final$par[1]),
                 shape2 = op.simp.final$par[7],
                 shape3 = op.simp.final$par[8]) +
        p1t * pgb2(d, shape1 = 1/op.simp.final$par[6],
                   scale = exp(as.matrix(cbind(
                     1,freqoutBCnew[i,"CoverageBC"])) %*%
                       op.simp.final$par[2:3]),
                   shape2 = op.simp.final$par[7],
                   shape3 = op.simp.final$par[8]) +
        p2t * pgb2(d, shape1 = 1/op.simp.final$par[6],
                   scale = exp(as.matrix(cbind(
                     1,freqoutBCnew[i,"CoverageBC"])) %*%
                       op.simp.final$par[4:5]),
                   shape2=op.simp.final$par[7],
                   shape3=op.simp.final$par[8]))
    return(v)
  },
  lower=0, upper=exp(freqoutBCnew$CoverageBC[i])*1000000)$val
  ulim.out2[i] <-
    exp(freqoutBCnew$CoverageBC[i])*1000000 *
    (1-(p0t * pgb2(exp(freqoutBCnew$CoverageBC[i])*1000000,
                   shape1 = 1/op.simp.final$par[6],
                   scale = exp(op.simp.final$par[1]),
                   shape2 = op.simp.final$par[7],
                   shape3 = op.simp.final$par[8]) +
          p1t * pgb2(exp(freqoutBCnew$CoverageBC[i])*1000000,
           shape1 = 1/op.simp.final$par[6],
                      scale = exp(as.matrix(cbind(
                        1,freqoutBCnew[i,"CoverageBC"])) %*%
                         op.simp.final$par[2:3]),
                     shape2 = op.simp.final$par[7],
                     shape3 = op.simp.final$par[8]) +
          p2t * pgb2(exp(freqoutBCnew$CoverageBC[i])*1000000,
           shape1 = 1/op.simp.final$par[6],
                      scale = exp(as.matrix(cbind(
                        1,freqoutBCnew[i,"CoverageBC"])) %*%
                         op.simp.final$par[4:5]),
                     shape2=op.simp.final$par[7],
                     shape3=op.simp.final$par[8])))
```

```
  remove(e1t,e2t,p0t,p1t,p2t)
}
any(is.na(ful.out2))# FALSE
any(is.na(ulim.out2))# FALSE


# Add Upper Limits.
ful.ulim.out2 <- ful.out2 + ulim.out2


# Frequency Score (Poisson)
regBC.pois4 <- glm(count2 ~ CoverageBC+NoClaimCreditBC+
                     TypeCity+TypeCounty+TypeMisc+TypeSchool+TypeTown,
                   data=freqinBCnew, family=poisson(link = "log"),
                   offset=log(v.gb2))
summary(regBC.pois4)

pois4.out <-
  exp(as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC","NoClaimCreditBC","TypeCity","TypeCounty","TypeMisc",
    "TypeSchool","TypeTown")]))%*% coef(regBC.pois4))


S_2 <- pois4.out * ful.ulim.out2
SS_2 <- pois4.out * (ful.ulim.out2 - mod.out2)


#  Comparison
cbind(
  rbind(sum(SS),sum(regBC.pois.score*regBC.gamma.score),
   sum(SS_2),sum(freqoutBCnew$Claim2)),
  rbind(
    cor(SS,freqoutBCnew$Claim2,method="pearson"),
    cor(regBC.pois.score*regBC.gamma.score,freqoutBCnew$Claim2,
     method="pearson"),
    cor(SS_2,freqoutBCnew$Claim2,method="pearson"),1),
  rbind(
    cor(SS,freqoutBCnew$Claim2,method="spearman"),
    cor(regBC.pois.score*regBC.gamma.score,freqoutBCnew$Claim2,
     method="spearman"),
    cor(SS_2,freqoutBCnew$Claim2,method="spearman"),1)
)


# Table 19. Comparison of Coefficients for Frequency Models

# Poisson Truncated Estimation
likelihoodpoistrunc<-function(a){
```

```
  n=8;
  beta<-a[1:n]
  x2 <- as.matrix(cbind(1,freqinBCnew[,c(
    "CoverageBC","NoClaimCreditBC","TypeCity",
    "TypeCounty","TypeMisc","TypeSchool","TypeTown")]))
  x2beta<-as.matrix(x2)%*%beta
  meanpoissonBC<-exp(x2beta)*v.gb2
  l<--sum(log(dpois(freqinBCnew$count2,lambda=meanpoissonBC)))
  return(l)
}
oppoistrunc<-optim(c(zeroBC$coefficients$count), likelihoodpoistrunc,
method=("BFGS"))
oppoistrunc<-optim(oppoistrunc$par,likelihoodpoistrunc,method=("Nelder-Mead"))
oppoistrunc<-BBoptim(oppoistrunc$par,likelihoodpoistrunc)
oppoistrunc<-optim(oppoistrunc$par,likelihoodpoistrunc,method=("Nelder-Mead"))
oppoistrunc<-optim(oppoistrunc$par,likelihoodpoistrunc,method=("BFGS"),hessian=T)
cbind(oppoistrunc$par,sqrt(diag(solve(oppoistrunc$hessian))))


# 0-1 Inflated Poisson Truncated Estimation
likelihood01poistrunc<-function(a){
  n=8; m=3;
  beta<-a[1:n]
  gamma0<-a[(n+1):(n+m)]
  gamma1<-a[(n+m+1):(n+m+m)]
  z <- as.matrix(cbind(1,freqinBCnew[,c(
    "CoverageBC","NoClaimCreditBC")]))
  x2 <- as.matrix(cbind(1,freqinBCnew[,c(
    "CoverageBC","NoClaimCreditBC","TypeCity",
    "TypeCounty","TypeMisc","TypeSchool","TypeTown")]))
  zgamma0<-as.matrix(z)%*%gamma0
  zgamma1<-as.matrix(z)%*%gamma1
  pzeroBC<-1/(exp(-zgamma0)+exp(zgamma1-zgamma0)+1)
  poneBC<-1/(exp(-zgamma1)+exp(zgamma0-zgamma1)+1)
  ptwoBC<-1-(pzeroBC+poneBC)
  x2beta<-as.matrix(x2)%*%beta
  meanpoissonBC<-exp(x2beta)*v.gb2
  Plv <- exp(-exp(x2beta)*(1-v.gb2))

  zeropoissonBC <- pzeroBC +
                   ptwoBC * exp(-exp(x2beta)) +
                   poneBC*(1-v.gb2) +
                   ptwoBC * exp(-meanpoissonBC)*(1-Plv)

  onepoissonBC <-poneBC +
                  ptwoBC * meanpoissonBC*exp(-meanpoissonBC)*(1-Plv)

  l<--sum(log((freqinBCnew$count2==0)*zeropoissonBC+
```

```
              (freqinBCnew$count2==1)*onepoissonBC+
              ptwoBC*dpois(freqinBCnew$count2,lambda=meanpoissonBC)))
  return(l)
}
op01poistrunc<-optim(c(zeroBC$coefficients$count,zeroBC$coefficients$zero,0,0,0),
                     likelihood01poistrunc,method=("BFGS"))
op01poistrunc<-optim(op01poistrunc$par,likelihood01poistrunc,method=("Nelder-Mead"))
op01poistrunc<-BBoptim(op01poistrunc$par,likelihood01poistrunc)
op01poistrunc<-optim(op01poistrunc$par,likelihood01poistrunc,method=("Nelder-Mead"))
op01poistrunc<-optim(op01poistrunc$par,likelihood01poistrunc,method=("BFGS"),hessian=T)
cbind(op01poistrunc$par,sqrt(diag(solve(op01poistrunc$hessian))))


# Table 5. Comparison of Predicted Counts Using Validation Sample
# (Compare Frequency Models)
#  1. Poisson
#  2. 0-1 Inflated Poisson
#  3. Poisson with GB2 Assumption
#  4. 0-1 Inflated Poisson with GB2 Assumption

# Frequency Score (Poisson)
regBC.pois0 <- glm(count1 ~ CoverageBC+NoClaimCreditBC+
                     TypeCity+TypeCounty+TypeMisc+TypeSchool+TypeTown,
                   data=freqinBCnew, family=poisson(link = "log"))
summary(regBC.pois0)

x2 <- as.matrix(cbind(1,freqoutBCnew[,c(
  "CoverageBC","NoClaimCreditBC","TypeCity","TypeCounty","TypeMisc",
  "TypeSchool","TypeTown")])
pois0.out <-
  exp()%*% coef(regBC.pois0))
numrow=20
exp1<-rep(0,numrow)
exp1[1]<-sum(dpois(0,lambda=pois0.out))
exp1[2]<-sum(dpois(1,lambda=pois0.out))
for(i in 3:numrow){
  exp1[i]<-sum(dpois((i-1),lambda=pois0.out))
}

# Frequency Score (0-1 Inflated Poisson Model)
n=8; m=3;
z <- as.matrix(cbind(1,freqoutBCnew[,c(
  "CoverageBC","NoClaimCreditBC")]))
x2 <- as.matrix(cbind(1,freqoutBCnew[,c(
  "CoverageBC","NoClaimCreditBC","TypeCity",
  "TypeCounty","TypeMisc","TypeSchool","TypeTown")]))
par<-opp01BCbfgs
beta<-par$par[1:n]
```

```
gamma0<-par$par[(n+1):(n+m)]
gamma1<-par$par[(n+m+1):(n+m+m)]
pzeroBC<-exp(as.matrix(z)%*%gamma0)/
  (1+exp(as.matrix(z)%*%gamma0)+exp(as.matrix(z)%*%gamma1))
poneBC<-exp(as.matrix(z)%*%gamma1)/
  (1+exp(as.matrix(z)%*%gamma1)+exp(as.matrix(z)%*%gamma0))
meanpoissonBC<-exp(as.matrix(x2)%*%beta)
numrow=20
exp2<-rep(0,numrow)
exp2[1]<-sum(pzeroBC+(1-pzeroBC-poneBC)*dpois(0,meanpoissonBC))
exp2[2]<-sum(poneBC+(1-pzeroBC-poneBC)*dpois(1,meanpoissonBC))
for(i in 3:numrow){
  exp2[i]<-sum((1-pzeroBC-poneBC)*dpois((i-1),meanpoissonBC))
}

# Frequency Score (Poisson Truncated Estimation with GB2 Assumption)
n=8;
x2 <- as.matrix(cbind(1,freqoutBCnew[,c(
  "CoverageBC","NoClaimCreditBC","TypeCity",
  "TypeCounty","TypeMisc","TypeSchool","TypeTown")]))
par<-oppoistrunc
beta<-par$par[1:n]
meanpoissonBC<-exp(as.matrix(x2)%*%beta)
numrow=20
exp3<-rep(0,numrow)
exp3[1]<-sum(dpois(0,meanpoissonBC))
exp3[2]<-sum(dpois(1,meanpoissonBC))
for(i in 3:numrow){
  exp3[i]<-sum(dpois((i-1),meanpoissonBC))
}

# Frequency Score (0-1 Inflated Poisson Truncated Estimation with GB2 Assumption)
n=8; m=3;
  z <- as.matrix(cbind(1,freqoutBCnew[,c(
    "CoverageBC","NoClaimCreditBC")]))
x2 <- as.matrix(cbind(1,freqoutBCnew[,c(
  "CoverageBC","NoClaimCreditBC","TypeCity",
  "TypeCounty","TypeMisc","TypeSchool","TypeTown")]))
par<-op01poistrunc
beta<-par$par[1:n]
gamma0<-par$par[(n+1):(n+m)]
gamma1<-par$par[(n+m+1):(n+m+m)]
pzeroBC<-exp(as.matrix(z)%*%gamma0)/
  (1+exp(as.matrix(z)%*%gamma0)+exp(as.matrix(z)%*%gamma1))
poneBC<-exp(as.matrix(z)%*%gamma1)/
  (1+exp(as.matrix(z)%*%gamma1)+exp(as.matrix(z)%*%gamma0))
meanpoissonBC<-exp(as.matrix(x2)%*%beta)
numrow=20
```

```
exp4<-rep(0,numrow)
exp4[1]<-sum(pzeroBC+(1-pzeroBC-poneBC)*dpois(0,meanpoissonBC))
exp4[2]<-sum(poneBC+(1-pzeroBC-poneBC)*dpois(1,meanpoissonBC))
for(i in 3:numrow){
  exp4[i]<-sum((1-pzeroBC-poneBC)*dpois((i-1),meanpoissonBC))
}

# Empirical Count
numrow<-20
emp<-rep(0,numrow)
for(i in 1:numrow){
  emp[i]<-sum(freqoutBCnew$count1==(i-1))
}

round(cbind(exp1,exp2,exp3,exp4,emp))


# By Peril Relativities
# (code for each peril type are omitted)

source("byperilcode.R")


# Table 3. Comparison of Relativities
# for Regression Approach and Selected MLE Approaches

Table3a <- round(rbind(
  reg.BC.fire.rels,
  reg.BC.vandalism.rels,
  reg.BC.light.rels,
  reg.BC.wind.rels,
  reg.BC.hail.rels,
  reg.BC.vehicle.rels,
  reg.BC.waternw.rels,
  reg.BC.waterw.rels,
  reg.BC.misc.rels),3)

Table3b <- round(rbind(
  temp.fire,
  temp.vandalism,
  temp.light,
  temp.wind,
  temp.hail,
  temp.vehicle,
  temp.waternw,
  temp.waterw,
  temp.misc),3)
```

```
Table3c <- round(rbind(
  temp.pareto.fire,
  temp.pareto.vandalism,
  temp.pareto.light,
  temp.pareto.wind,
  temp.pareto.hail,
  temp.pareto.vehicle,
  temp.pareto.waternw,
  temp.pareto.waterw,
  temp.pareto.misc),3)

colnames(Table3a) <- colnames(Table3b) <- colnames(Table3c) <-
  round(sort(unique(
  freqinBCnew$DeductBC*1000000))[2:8],0)
rownames(Table3a) <- rownames(Table3b) <- rownames(Table3c) <-
  c("Fire","Vandalism","Lightning",
  "Wind","Hail","Vehicle",
  "WaterNW","WaterW","Misc")

Table3a
Table3b
Table3c


# Figure 4. Plot of Relativities for Regression Approach and Selected MLE Approaches

plot(c(1000,2500,5000,10000,15000,25000,50000),Table3a[1,1:7],
     type="l", xlim=c(1000,50000), ylim=c(0,1))
for (i in 2:9) { lines(c(1000,2500,5000,10000,15000,25000,50000),
                       Table3a[i,1:7], type="l", lty=i, lwd=i, col=i) }
legend("topright",c("Fire","Vandalism","Lightning","Wind","Hail",
                    "Vehicle","Water(NW)","Water(W)","Misc"),lty=1:9,
                    lwd=1:9, col=1:9, bg="white")

plot(c(1000,2500,5000,10000,15000,25000,50000),Table3b[1,1:7],
     type="l", xlim=c(1000,50000), ylim=c(0,1))
for (i in 2:9) { lines(c(1000,2500,5000,10000,15000,25000,50000),
                       Table3b[i,1:7], type="l", lty=i, lwd=i, col=i) }
legend("topright",c("Fire","Vandalism","Lightning","Wind","Hail",
                    "Vehicle","Water(NW)","Water(W)","Misc"),lty=1:9,
                    lwd=1:9, col=1:9, bg="white")

plot(c(1000,2500,5000,10000,15000,25000,50000),Table3c[1,1:7],
     type="l", xlim=c(1000,50000), ylim=c(0,1))
for (i in 2:9) { lines(c(1000,2500,5000,10000,15000,25000,50000),
                       Table3c[i,1:7], type="l", lty=i, lwd=i, col=i) }
legend("bottomleft",c("Fire","Vandalism","Lightning","Wind","Hail",
                    "Vehicle","Water(NW)","Water(W)","Misc"),lty=1:9,
```

```
                    lwd=1:9, col=1:9, bg="white")

mains <- c("Fire","Vandalism","Lightning","Wind","Hail",
           "Vehicle","Water(NW)","Water(W)","Misc")
par(mfrow=c(1,3))

for (i in 1:3) {
  plot(c(1000,2500,5000,10000,15000,25000,50000),Table3a[i,1:7],
       type="l", xlim=c(1000,50000), ylim=c(0,1), xlab=c("Deductible"),
        ylab=c("Relativity"),
       main=mains[i],lty=3, lwd=2, col=2)
  lines(c(1000,2500,5000,10000,15000,25000,50000),Table3b[i,1:7],
  lty=5, lwd=1, col=3)
  lines(c(1000,2500,5000,10000,15000,25000,50000),Table3c[i,1:7],
  lty=1, lwd=1, col=1) }
legend("topright",c("Regression","Gamma MLE","Pareto MLE"), lty=c(3,5,1),
lwd=c(2,1,1), col=c(2,3,1))

for (i in 4:6) {
  plot(c(1000,2500,5000,10000,15000,25000,50000),Table3a[i,1:7],
       type="l", xlim=c(1000,50000), ylim=c(0,1), xlab=c("Deductible"),
       ylab=c("Relativity"),
       main=mains[i],lty=3, lwd=2, col=2)
  lines(c(1000,2500,5000,10000,15000,25000,50000),Table3b[i,1:7],
  lty=5, lwd=1, col=3)
  lines(c(1000,2500,5000,10000,15000,25000,50000),Table3c[i,1:7],
  lty=1, lwd=1, col=1) }
legend("topright",c("Regression","Gamma MLE","Pareto MLE"),
lty=c(3,5,1), lwd=c(2,1,1), col=c(2,3,1))

for (i in 7:9) {
  plot(c(1000,2500,5000,10000,15000,25000,50000),Table3a[i,1:7],
       type="l", xlim=c(1000,50000), ylim=c(0,1), xlab=c("Deductible"),
       ylab=c("Relativity"),
       main=mains[i],lty=3, lwd=2, col=2)
  lines(c(1000,2500,5000,10000,15000,25000,50000),Table3b[i,1:7],
  lty=5, lwd=1, col=3)
  lines(c(1000,2500,5000,10000,15000,25000,50000),Table3c[i,1:7],
  lty=1, lwd=1, col=1) }
legend("topright",c("Regression","Gamma MLE","Pareto MLE"),
lty=c(3,5,1), lwd=c(2,1,1), col=c(2,3,1))

round(rbind(
  c( coef(regBC.pois.fire.ded)["lnDeductBC"] ,
   coef(regBC.gamma.fire.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.vandalism.ded)["lnDeductBC"] ,
   coef(regBC.gamma.vandalism.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.light.ded)["lnDeductBC"] ,
```

```
   coef(regBC.gamma.light.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.wind.ded)["lnDeductBC"] ,
   coef(regBC.gamma.wind.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.hail.ded)["lnDeductBC"] ,
   coef(regBC.gamma.hail.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.vehicle.ded)["lnDeductBC"] ,
   coef(regBC.gamma.vehicle.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.waternw.ded)["lnDeductBC"] ,
   coef(regBC.gamma.waternw.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.waterw.ded)["lnDeductBC"] ,
   coef(regBC.gamma.waterw.ded)["lnDeductBC"] ),
  c( coef(regBC.pois.misc.ded)["lnDeductBC"] ,
   coef(regBC.gamma.misc.ded)["lnDeductBC"] )),4)

# Shape Parameters
round(t(t(c(1-op_pareto3fire_covar$par[3],
      1-op_pareto3vandalism_covar$par[3],
      1-op_pareto3light_covar$par[3],
      1-op_pareto3wind_covar$par[3],
      1-op_pareto3hail_covar$par[3],
      1-op_pareto3vehicle_covar$par[3],
      1-op_pareto3waternw_covar$par[3],
      1-op_pareto3waterw_covar$par[3],
      1-op_pareto3misc_covar$par[3]))),4)
```