

Arizona State University

From the Selected Works of Joseph M Hilbe

September 15, 2016

Modeling Count Data Rcode update 15Sep2016

Joseph M Hilbe



Available at: https://works.bepress.com/joseph_hilbe/71/

Modeling Count Data, Cambridge University Press (2014)

Joseph M Hilbe hilbe@asu.edu

18 December, 2014 update

R Code

R code in text, with Chapter and section for which each code unit or table of R code is found. If you spot typos, problems, or have suggestions, please contact me at hilbe@asu.edu or j.m.hilbe@gmail.com. Alternative approaches, or providing R code for procedures or graphs which were not programmed in R for the book are most welcome.

Modeling Count Data was written for analysts, statisticians, students, and others interested in this area of statistics or data analysis who do not already have a background in the area. For those who wish to learn more details, including derivations, simulations, and so forth, I recommend Hilbe, *Negative Binomial Regression*, second edition (2011, Cambridge University Press).

See my website for updated MCD Errata and Comments, data sets formatted in Stata, R, SAS, and Excel, as well code: http://works.bepress.com/joseph_hilbe/

Joseph M. Hilbe

```
# =====
#           CHAPTER 1  Varieties of Count Data
# =====
#
# Section 1.2.1
# -----
#
# R CODE
sbp    <- c(131,132,122,119,123,115)
male   <- c(1,1,1,0,0,0)
smoker <- c(1,1,0,0,1,0)
age    <- c(34,36,30,32,26,23)
summary(reg1 <- lm(sbp~ male+smoker+age))
mu <- predict(reg1)
mu
cof <- reg1$coef
cof
xb <- cof[1] + cof[2]*male + cof[3]*smoker + cof[4]*age
xb
diff <- sbp - mu
diff

# Section 1.2.3
-----

# Table 1.2a R: Code for Figure 1.2a
# =====
library(gamlss)
obs <- 15
mu <- 4
y=seq(0:(obs-1))
```

```

alpha <- .5
amu <- mu*alpha
layout(1)
all.lines <- vector(mode = 'list', length = 5)
for (i in 1:length(mu)) {
  yp = exp(-mu[i])*(mu[i]^y)/factorial(y)
  ynb1 = exp( log(gamma(mu[i]/alpha + y))
             - log(gamma(y+1))
             - log(gamma(mu[i]/alpha))
             + (mu[i]/alpha)*log(1/(1+alpha))
             + y*log(1-1/(1+alpha)))
  ynb2 = exp( y*log(amu[i]/(1+amu[i]))
             - (1/alpha)*log(1+amu[i])
             + log( gamma(y +1/alpha) )
             - log( gamma(y+1) )
             - log( gamma(1/alpha) ))
  ypig = dPIG(y-1, mu, 1/alpha)
  ygp = exp( log((1-alpha)*mu[i])
            + (y-1)*log((1-alpha) * mu[i]+alpha*y)
            - (1-alpha)*mu[i]
            - alpha*y
            - log(gamma(y+1)))
  all.lines = list(yp = yp, ynb1 = ynb1, ynb2 = ynb2, ypig = ypig, ygp = ygp)
  ymax = max(unlist(all.lines), na.rm=TRUE)
  cols = c("red","blue","black","green","purple")
  plot(y, all.lines[[1]], ylim =
        c(0, ymax), type = "n", main="5 Count Distributions: mean=4; alpha=0.5")
  for (j in 1:5)
    lines(y, all.lines[[j]], ylim = c(0, ymax), col=cols[j],type='b',pch=19, lty=j)
    legend("topright",cex = 1.5, pch=19,
          legend=c("NB2","POI","PIG","NB1","GP"),
          col = c(1,2,3,4,5),
          lty = c(1,1,1,1,1),
          lwd = c(1,1,1,1,3))
}
# =====

# Section 1.4.2
-----

# Table 1.4 R: Poisson probabilities for y from 0 through 4
# =====
y <- c(4, 2, 0,3, 1, 2)
y0 <- exp(-2)* (2^0)/factorial(0)
y1 <- exp(-2)* (2^1)/factorial(1)
y2 <- exp(-2)* (2^2)/factorial(2)
y3 <- exp(-2)* (2^3)/factorial(3)
y4 <- exp(-2)* (2^4)/factorial(4)
poisProb <- c(y0, y1, y2, y3, y4); poisProb

# OR
dpois(0:4, lambda=2)

# CUMULATIVE
ppois(0:4, lambda=2)

# to plot a histogram
py <- 0:4
plot(poisProb ~ py, xlim=c(0,4), type="o", main="Poisson Prob 0-4: Mean=2")
# =====

```

```

# Table 1.5 R : Code for Figure 1.3
# =====
m<- c(0.5,1,3,5)           #Poisson means
y<- 0:11                   #Observed counts
layout(1)
for (i in 1:length(m)) {
  p<- dpois(y, m[i])       #poisson pdf
  if (i==1) {
    plot(y, p, col=i, type='l', lty=i)
  } else {
    lines(y, p, col=i, lty=i)
  }
}
# =====

# =====
#                               CHAPTER 2   Poisson Regression
# =====
#
# Section 2.3
# -----
#
# Table 2.4 R: Synthetic Poisson Model
# =====
library(MASS)
library(COUNT)
set.seed(4590)
nobs <- 50000
x1 <- runif(nobs)
x2 <- runif(nobs)
x3 <- runif(nobs)
py <- rpois(nobs, exp(1 + 0.75*x1 - 1.25*x2 + .5*x3))
cnt <- table(py)
dataf <- data.frame(prop.table(table(py) ) )
dataf$cumulative <- cumsum(dataf$Freq)
datafall <- data.frame(cnt, dataf$Freq*100, dataf$cumulative * 100)
datafall
summary(py)
summary(py1 <- glm(py ~ x1 + x2 + x3, family=poisson))
confint.default(py1)
py1$aic/(py1$df.null+1)
r <- resid(py1, type = "pearson")
pch2 <- sum(residuals(py1, type="pearson")^2)
disp <- pchi2/py1$df.residual; pchi2; disp
# =====

# Table 2.6 R: Monte Carlo Poisson
# =====
mysim <- function()
{
  nobs <- 50000
  x1 <- runif(nobs)
  x2 <- runif(nobs)
  x3 <- runif(nobs)
  py <- rpois(nobs, exp(2 + .75*x1 - 1.25*x2 + .5*x3))
  poi <- glm(py ~ x1 + x2 + x3, family=poisson)
  pr <- sum(residuals(poi, type="pearson")^2)
  prdisp <- pr/poi$df.residual
  beta <- poi$coef
  list(beta,prdisp)
}
B <- replicate(100, mysim())
apply(matrix(unlist(B[1,]),4,100),1,mean)

```

```

# =====
# Dispersion
mean(unlist(B[2,]))

# Section 2.4
# -----

# Table 2.7 R: Example Poisson Model and Associated Statistics
# =====
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
cage <- rwm1984$age - mean(rwm1984$age)
summary(poic <- glm(docvis ~ outwork + cage, family=poisson, data=rwm1984))
pr <- sum(residuals(poic, type="pearson")^2) # Pearson Chi2
pr/poic$df.residual # dispersion statistic
modelfit(poic)
cnt <- table(docvis)
dataf <- data.frame(prop.table(table(docvis) ) )
dataf$cumulative <- cumsum(dataf$Freq)
datafall <- data.frame(cnt, dataf$Freq*100, dataf$cumulative * 100)
datafall
# =====

# Table 2.8 R: Change Levels in Categorical Predictor
# =====
levels(rwm1984$edlevel) # levels of edlevel
elevel <- rwm1984$edlevel # new variable
levels(elevel)[2] <- "Not HS grad" # assign level 1 to 2
levels(elevel)[1] <- "HS" # rename level 1 to "HS"
levels(elevel) # levels of elevel
summary(tst2 <- glm(docvis ~ outwork + cage + female + married + kids
+ factor(elevel), family=poisson, data=rwm1984))
# =====

# Section 2.5.1
# -----

summary(pyq <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))

# Likelihood Profiling of SE
confint(pyq)

# Traditional Model-based SE
confint.default(pyq)

# Section 2.5.2
# -----

# Table 2.11 R: Poisson Model - Rate Ratio Parameterization
# =====
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
summary(poil <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))
pr <- sum(residuals(poil, type="pearson")^2) # Pearson Chi2
pr/poil$df.residual # dispersion statistic
poil$aic / (poil$df.null+1) # AIC/n
exp(coef(poil)) # IRR
exp(coef(poil))*sqrt(diag(vcov(poil))) # delta method
exp(confint.default(poil)) # CI of IRR
# =====

```

```

# Section 2.6
# -----

# Table 2.12 R: Poisson with Exposure
# =====
data(fasttrakg)
summary(fast <- glm(die ~ anterior + hcabg + factor(killip),
                  family=poisson,
                  offset=log(cases),
                  data=fasttrakg))

exp(coef(fast))
exp(coef(fast))*sqrt(diag(vcov(fast)))
exp(confint.default(fast))
modelfit(fast)
# =====

# Section 2.7
# -----

# R prediction
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
myglm <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984)
lpred <- predict(myglm, newdata=rwm1984, type="link", se.fit=TRUE)
up <- lpred$fit + (1.96 * lpred$se.fit)
lo <- lpred$fit - (1.96 * lpred$se.fit)
eta <- lpred$fit
upci <- myglm$family$linkinv(up)
mu <- myglm$family$linkinv(eta)
loci <- myglm$family$linkinv(lo)
summary(loci)
summary(mu)
summary(upci)
layout(1)
plot(eta, mu, col=1)
lines(eta, loci, col=2, type='p')
lines(eta, upci, col=3, type='p')

# Section 2.8.1
# -----

# Table 2.14 R: Marginal Effects at Mean
# =====
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
summary(pmem <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))
mout <- mean(rwm1984$outwork)
mage <- mean(rwm1984$age)
xb <- coef(pmem) [1] + coef(pmem) [2]*mout + coef(pmem) [3]*mage
dfdx <- exp(xb) * coef(pmem) [3]
mean(dfdx)
# =====

# Section 2.8.2
# -----

R CODE
mean(rwm1984$docvis) * coef(pmem) [3]

```

```

# Section 2.8.3
-----

# R CODE  discrete change
summary(pmem <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))
mu0 <- exp(pmem$coef[1] + pmem$coef[3]*mage)
mu1 <- exp(pmem$coef[1] + pmem$coef[2] + pmem$coef[3]*mage)
pe <- mu1 - mu0
mean(pe)

R CODE  avg partial effects
summary(pmem <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))
bout = coef(pmem)[2]
mu = fitted.values(pmem)
xb = pmem$linear.predictors
pe_out = 0
pe_out = ifelse(rwm1984$outwork == 0, exp(xb + bout)-exp(xb), NA)
pe_out = ifelse(rwm1984$outwork == 1, exp(xb)-exp(xb-bout), pe_out)
mean(pe_out)

# =====
#                               CHAPTER 3 Testing Overdispersion
# =====
#
# Section 3.1
# -----

# Table 3.1  R: Deviance Goodness-of-Fit Test
# =====
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
mymod <-glm(docvis ~ outwork + age, family=poisson, data=rwm1984)
mymod

dev<-deviance(mymod)
df<-df.residual(mymod)
p_value<-1-pchisq(dev,df)
print(matrix(c("Deviance GOF","D","df","p-value", " ",
              round(dev,4),df, p_value), ncol=2))
=====

# R CODE
mymod <-glm(docvis ~ outwork + age, family=poisson, data=rwm1984)
pr <- sum(residuals(mymod, type="pearson")^2) # get Pearson Chi2
pchisq(pr, mymod$df.residual, lower=F)      # calc p-value
pchisq(mymod$deviance, mymod$df.residual, lower= F) # calc p-vl

source("c:\\Rfiles\\P__disp.r") # use folder where you keep r files
P__disp(mymod)

# R CODE
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
mymod <-glm(docvis ~ outwork + age, family=poisson, data=rwm1984)
P__disp(mymod)

```

```

# Table 3.2 R: Function to Calculate Pearson Chi2 and Dispersion Statistics
# =====
P__disp <- function(x) {
  pr <- sum(residuals(x, type="pearson")^2)
  dispersion <- pr/x$df.residual
  cat("\n Pearson Chi2 = ", pr ,
      "\n Dispersion   = ", dispersion, "\n")
}
# =====

# Section 3.3.1
# -----

# Table 3.4 R: Z-Score Test
# =====
library(COUNT); data(rwm5yr); rwm1984 <- subset(rwm5yr, year==1984)
summary(poi <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))
mu <- predict(poi, type="response")
z <- ((rwm1984$docvis - mu)^2 - rwm1984$docvis) / (mu * sqrt(2))
summary(zscore <- lm(z ~ 1))
# =====

# Section 3.3.2
# -----

# Table 3.5 R: Lagrange Multiplier Test
# =====
obs <- nrow(rwm1984) # continue from Table 3.2
mmu <- mean(mu); nybar <- obs*mmu; musq <- mu*mu
mu2 <- mean(musq)*obs
chival <- (mu2 - nybar)^2 / (2*mu2); chival
pchisq(chival,1,lower.tail = FALSE)
# =====

# Section 3.3.3
# -----

# Table 3.7 R: Poisson Model with Ancillary Statistics
# =====
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
summary(poil <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))
pr <- sum(residuals(poil, type="pearson")^2) # Pearson Chi2
pr/poil$df.residual # dispersion statistic
poil$aic / (poil$df.null+1) # AIC/n
exp(coef(poil)) # IRR
exp(coef(poil))*sqrt(diag(vcov(poil))) # delta method
exp(confint.default(poil)) # CI of IRR
modelfit(poil) # same as Stata abic
sd(rwm1984$docvis)^2 # observed variance
xbp <- predict(poil) # xb, linear predictor
mup <- exp(xbp) # mu, fitted Poisson
mean(mup) # expected variance: mean=variance
# Table of observed vs expected counts
rbind(obs=table(rwm1984$docvis)[1:18],
      exp = round(sapply(0:17, function(x)sum(dpois(x, fitted(poil))))))
meany <- mean(rwm1984$docvis) # mean docvis
expect0 <- exp(-meany)*meany^0 / exp(log(factorial(0))) # expected prob of 0
zerodays <- (poil$df.null+1) *expect0 # expected zero days
obs=table(rwm1984$docvis)[1:18] # observed number values in each count 0-17
exp = round(sapply(0:17, function(x)sum(dpois(x, fitted(poil)))) #expected each count
chisq.test(obs, exp) # ChiSq test if obs & exp from same pop
# =====

```



```

# Section 3.4.1
# -----

# R: quasipoisson
# =====
data(medpar)
summary(poiql <- glm(los ~ hmo + white + hmo + factor(type),
                    family=quasipoisson, data=medpar))
# =====

# Table 3.8 R: Scaling SE Medpar Data
# =====
library(COUNT)
data(medpar)
attach(medpar)
summary(poi <- glm(los ~ hmo + white + factor(type), family=poisson,
                 data=medpar))
confint(poi) # profile confidence interval
pr <- sum(residuals(poi,type="pearson")^2) # Pearson statistic
dispersion <- pr/poi$df.residual; dispersion # dispersion
sse <- sqrt(diag(vcov(poi))) * sqrt(dispersion)
sse # model SE
# OR
poiQL <- glm(los ~ hmo + white + factor(type), family=quasipoisson,
            data=medpar)
coef(poiQL); confint(poiQL) # coeff & scaled SEs
modelfit(poi) # AIC,BIC statistics
# =====

# Section 3.4.2
# -----

# Table 3.10 R: Quasi-likelihood Poisson Standard Errors
# =====
poiQL <- glm(los ~ hmo+white+type2+type3, family=poisson, data=medpar)
summary(poiQL)
pr <-sum(residuals(poiQL, type="pearson")^2 )
disp <- pr/poiQL$df.residual # Pearson dispersion
se <-sqrt(diag(vcov(poiQL)))
QLse <- se/sqrt(disp)
QLse
# =====

# Section 3.4.3
# -----

# Table 3.12 R: Robust Standard Errors of medpar Model
# =====
library(sandwich)
poi <- glm(los ~ hmo + white + factor(type), family=poisson, data=medpar)
vcovHC(poi)
sqrt(diag(vcovHC(poi, type="HC0"))) # final HC0 = H-C-zero
# Clustering
poi <- glm(los ~ hmo + white + factor(type), family=poisson, data=medpar)
library(haplo.ccs)
sandcov(poi, medpar$provnum)
sqrt(diag(sandcov(poi, medpar$provnum)))
# =====

summary(poi1 <- glm(los ~ hmo+white+factor(type), family=poisson, data=medpar))

```

```

# Section 3.4.4
# -----

# Table 3.13 R: Bootstrap Standard Errors
# =====
library(COUNT)
library(boot)
data(medpar)
poi <- glm(los ~ hmo + white + factor(type), family=poisson, data=medpar)
summary(poi)
t <- function(x, i) {
xx <- x[i,]
  bsglm <- glm( los ~ hmo + white + factor(type), family=poisson, data=medpar)
  return(sqrt(diag(vcov(bsglm))))
}
bse <- boot(medpar, t, R=1000)
sqrt(diag(vcov(poi)))
apply(bse$t,2, mean)
# =====

detach(medpar)

# =====
#                               CHAPTER 4  Asssment of Fit
#                               =====
#
# Section 4.1
# -----

# R Pearson Chi2 and statistic and graph
summary(pexp <- glm(docvis ~ outwork + cage, family=poisson, data=rwm1984))
presid <- residuals(pexp, type="pearson")
pchi2 <- sum(residuals(pexp, type="pearson")^2) # Pearson Chi2
summary(rwm <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984))
P__disp(rwm)
mu <- predict(rwm)
grd <- par(mfrow = c(2,2))
plot(x=mu, y= rwm$docvis, main = "Response residuals")
plot(x=mu, y= presid, main = "Pearson residuals")

# Section 4.2.1
# -----

# Table 4.2 R: Likelihood Ratio Test
# =====
library(COUNT)
library(lmtest)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
poil <- glm(docvis ~ outwork + age, family=poisson, data=rwm1984)
poila <- glm(docvis ~ outwork, family=poisson, data=rwm1984)
lrtest(poil, poila)
drop1(poil, test="Chisq")
# =====

# Section 4.2.2
# -----

# Chi2 test boundary LR
pchisq(2.705,1, lower.tail=FALSE)/2

```

```

# Section 4.3.2
# -----
# Table 4.4 R: Version of Stata User Command, abic
# =====
modelfit <- function(x) {
  obs <- x$df.null + 1
  aic <- x$aic
  xvvars <- x$df.null - x$df.residual + 1
  rdof <- x$df.residual
  aic_n <- aic/obs
  ll <- xvvars - aic/2
  bic_r <- x$deviance - (rdof * log(obs))
  bic_l <- -2*ll + xvvars * log(obs)
  bic_qh <- -2*(ll - xvvars * log(xvvars))/obs
  c(AICn=aic_n, AIC=aic, BICqh=bic_qh, BICl=bic_l)
}
modelfit(x) # substitute fitted model name for x
# =====
library(COUNT)
data(medpar)
mymodel <- glm(los ~ hmo + white + factor(type), family=poisson, data=medpar)
modelfit(mymodel)

# =====
# CHAPTER 5 Negative Binomial Regression
# =====
#
# Section 5.3.1
# -----
# Table 5.4 R: rwm1984 Modeling Example
# =====
# make certain the appropriate packages are loaded
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
# USING glm.nb
summary(nbx <- glm.nb(docvis ~ outwork + age + married + female +
  edlevel2 + edlevel3 + edlevel4, data=rwm1984))
exp(coef(nbx))
exp(coef(nbx))*sqrt(diag(vcov(nbx)))
exp(confint.default(nbx))
alpha <- 1/nbx$theta
alpha
P_disp(nbx)
modelfit(nbx)
xbnb <- predict(poi1)
munb <- exp(xbnb)
# expected variance of NB model (using alpha where alpha=1/theta)
mean(munb) + (1/nbx$theta)*mean(munb)^2
round(sqrt(rbind(diag(vcov(nbx)), diag(sandwich(nbx)))), digits=4)
# USING nbinomial
summary(nb1 <- nbinomial(docvis ~ outwork + age + married + female +
  edlevel2 + edlevel3 + edlevel4, data=rwm1984))
modelfit(nb1)
# =====

# Table 5.5 R: rwm1984 Poisson and NB2 Models
# =====
library(COUNT)
library(msme)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)

```

```

# POISSON
poi <- glm(docvis ~ outwork + age + married + female +
           edlevel2 + edlevel3 + edlevel4,
           family = poisson, data = rwm1984)
summary(poi)

#NB2
summary(nb1 <- nbinomial(docvis ~ outwork + age + married + female +
                        edlevel2 + edlevel3 + edlevel4, data=rwm1984))
# NB1
library(gamlss)
summary(gamlss(formula = docvis ~ outwork + age + married + female +
               edlevel2 + edlevel3 + edlevel4, family = NBII, data = rwm1984))
# =====

# Section 5.4.3
# -----
# POISSON
library(COUNT)
data(nuts)
nut <- subset(nuts, dbh<.6)
sntrees <- scale(nut$sntrees)
sheight <- scale(nut$sheight)
scover <- scale(nut$scover)
summary(PO <- glm(cones ~ sntrees + sheight + scover, family=quasipoisson, data=nut))
table(nut$cones)
summary(nut$cones)

# NEGATIVE BINOMIAL
library(msme)
summary(NB <- nbinomial(cones ~ sntrees + sheight + scover, data=nut))

# HETEROGENEOUS NEGATIVE BINOMIAL
summary(HNB <- nbinomial(cones ~ sntrees + sheight + scover,
                        formula2 =~ sntrees + sheight + scover, data=nut, family = "negBinomial",
                        scale.link = "log_s"))
exp(coef(HNB))

# =====
#          CHAPTER 6 Poisson Inverse Gaussian Regression
# =====
#
# Section 6.1
# -----
library(gamlss)
dPIG(2, .5 ,1)
#
# Section 6.2.2
# -----

# Table 6.3 R: Poisson Inverse Gaussian - rwm1984
# =====
library(gamlss); library(COUNT); library(msme); library(sandwich)
data(rwm5yr); rwm1984 <- subset(rwm5yr, year==1984)
summary(nbmod <- glm.nb(docvis ~ outwork + age, data=rwm1984))
vcovHC(nbmod)
sqrt(diag(vcovHC(nbmod, type="HC0")))
pigmod <- gamlss(docvis ~ outwork + age, data=rwm1984, family=PIG)
summary(pigmod)
exp(coef(pigmod))
# =====
exp(1.344)

```

```

# Table 6.5 R: Poisson Inverse Gaussian - medpar
# =====
library(gamlss); library(COUNT); library(msme); library(sandwich)
data(medpar)
rwm1984 <- subset(rwm5yr, year==1984)
summary(nbmod1 <- glm.nb(los ~ hmo + white + factor(type), data=medpar))
vcovHC(nbmod1)
sqrt(diag(vcovHC(nbmod1, type="HC0")))
pigmod1 <- gamlss(los ~ hmo + white + factor(type), data=medpar, family=PIG)
summary(pigmod1)
exp(coef(pigmod1))
# =====

# Section 6.3
# -----
#
# page 171
#
# pigrng.r function
pigrng <- function(mu=2.0, disp=1, no=20) {
  pr=rep(0,no)
  cc=1+2*mu*disp
  cc1=sqrt(cc)
  pr[1]=exp((1-cc1)/disp)
  pr[2]=(mu/cc1)*pr[1]
  q1=2*mu*disp/cc
  q2=(mu^2)/cc
  for (i in 3:no){
    yv=i-1
    t1=1-3/(2*yv)
    t2=1/(yv*(yv-1))
    pr[i]=q1*t1*pr[yv] + q2*t2*pr[yv-1]
  }
  y=seq(0:(no-1))
  result=cbind(y-1, pr)
}
# Use as:
# source("c://rfiles/pigrng.r")
# yp <- pigrng(4, .5, 10) # or any other values
# ypig <- yp[,2]
# ypig
# plot(yp)
#
#
library(COUNT)
library(gamlss)
data(rwm1984)
summary(pigmod <- gamlss(docvis ~ outwork + age + married + female +
  edlevel3 + edlevel3 + edlevel4, data=rwm1984, family=PIG))
# PIG dispersion parameter
exp(1.323)

# Predicted probabilities
yp <- pigrng(mean(pigmod$mu.fv),exp(pigmod$sigma.coefficient), 11)
ypig <- yp[,2]
ypig

# predicted & observed counts
pigexp <- dim(rwm1984)[1]*ypig
pigexp
obs=table(rwm1984$docvis)[1:11]
obs

```

```

# table of observed and predicted counts from 0-10
rbind(obs, pigexp[1:11])
chisq.test(obs, pigexp[1:11])
# Figure 6.1
pigpred <- pigexp[1:11]
cnt <- 0:10
layout(1)
plot(cnt, obs)
lines(cnt, obs, type="b")
lines(cnt, pigpred, type="b", pch=24)
#

# =====
#          CHAPTER 7 Problems with Zero Counts
# =====
#
# R
# =====
exp(-3) * 3^0 / exp(log(factorial(0)))
100* (exp(-3) * 3^0 / exp(log(factorial(0))))
# =====

# Section 7.1.1
# -----

# Table 7.1 R: Poisson and Zero-truncated Poisson
# =====
library(msme)
library(gamlss.tr)
data(medpar)
poi <- glm(los~ white + hmo + factor(type), family=poisson, data=medpar)
summary(poi)
ztp <- gamlss(los~ white + hmo + factor(type),data=medpar, family=PO)
gen.trun(0, "POI", type="left", name = "lefttr")
lt0poi <- gamlss(los~white+hmo+ factor(type), data=medpar, family=POlefttr)
summary(lt0poi)
# =====

# Section 7.1.2
# -----

# Table 7.2 Zero-Truncated Negative Binomial
# =====
library(msme); library(gamlss.tr)
data(medpar)
nb <- nbinomial(los~ white + hmo + factor(type), data=medpar)
summary(nb)
ztnb <- gamlss(los~ white + hmo + factor(type),data=medpar, family=NBI)
gen.trun(0, "NBI", type="left", name = "lefttr")
lt0nb <- gamlss(los~white+hmo+ factor(type), data=medpar, family=NBlefttr)
summary(lt0nb)
# =====

# R: Calculate NB2 expected 0's for given ? and ?
# =====
a <- 1
mu <- 2
y <- 0
exp(y*log(a*mu/(1+a*mu)) - (1/a)*log(1+a*mu) +
  log(gamma(y + 1/a)) - log(gamma(y+1)) - log(gamma(1/a)))
# =====

```

```

# R: Proof that sum of y probabilities from 0 to 100 is 1
# =====
a <- 1
mu <- 2
y <- 0:100
ff <- exp(y*log(a*mu/(1+a*mu)) - (1/a)*log(1+a*mu) +
  log(gamma(y + 1/a)) - log(gamma(y+1)) - log(gamma(1/a)))
sum(ff)
# =====

# Section 7.2.1
# -----

# Table 7.3 R: Poisson-Logit Hurdle
# =====
library(pscl)
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
hpl <- hurdle(docvis ~ outwork + age, dist="poisson", data=rwm1984,
  zero.dist="binomial", link="logit")

summary(hpl)
AIC(hpl)
# =====

# Table 7.4: R Components to Poisson-Logit Hurdle
# =====
visit <- ifelse(rwm1984$docvis > 0, 1, 0)
table(visit)
logis <- glm(visit ~ outwork + age, data=rwm1984,
  family=binomial(link="logit"))

summary(logis)
library(pscl)
hpl2 <- hurdle(docvis ~ outwork + age, data=rwm1984,
  dist = "poisson", zero.dist="binomial", link="logit")
summary(hpl2)
logit <- glm(visit ~ outwork + age, data=rwm1984,
  family=binomial(link="logit"))
summary(logit)
# =====

# Table 7.5 R NB2-logit Hurdle <Assume Model from 7.3 Loaded>
# =====
hnbl <- hurdle(docvis ~ outwork + age, dist="negbin", data=rwm1984,
  zero.dist="binomial", link="logit")

summary(hnbl)
AIC(hnbl)
alpha <- 1/hnbl$theta
alpha
exp(coef(hnbl))
predhnbl <- hnbl$fitted.values
# =====

```

```

# Section 7.3.5
# -----
# Table 7.6. R - ZIP
# =====
library(pscl); library(COUNT); data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
poi <- glm(docvis ~ outwork + age, data=rwm1984, dist="poisson")
zip <- zeroinfl(docvis ~ outwork + age | outwork + age, data=rwm1984, dist="poisson")
summary(zip)
print(vuong(zip,poi))
exp(coef(zip))
round(colSums(predict(zip, type="prob")[,1:17])) # expected counts
rbind(obs=table(rwm1984$docvis)[1:18]) # observed counts
# =====

# R CODE
pred <- round(colSums(predict(zip, type="prob") [,1:13]))
obs <- table(rwm1984$docvis)[1:13]
rbind(obs, pred)

# Section 7.3.6
# -----

# Table 7.7. R - ZINB
# =====
library(pscl)
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
nb2 <- glm.nb(docvis ~ outwork + age, data=rwm1984)
zinb <- zeroinfl(docvis ~ outwork + age | outwork + age, data=rwm1984, dist="negbin")
summary(zinb)
print(vuong(zinb,nb2))
exp(coef(zinb))
pred <- round(colSums(predict(zinb, type="prob")[,1:13])) # expected counts
obs <- table(rwm1984$docvis)[1:13] # observed counts
rbind(obs, pred)
# =====

# Section 7.3.7
# -----

# R ZIPIG
# =====
library(gamlss)
data(rwm1984)
attach(rwm1984)
summary(zpig <- gamlss(docvis ~ outwork + age, sigma.fo= ~ 1,
  family=ZIPIG, data=rwm1984))
# code for calculating vuong, LR test, etc on book's website
# =====

# =====
# CHAPTER 8 Generalized Poisson
# =====
# =====
# CHAPTER 9 More Advanced Models
# =====
#
# 9.1 exact models

```



```

# R
# =====
library(COUNT)
data(azcabgptca)
attach(azcabgpca)
table(los)
table(procedure, type)
table(los, procedure)
summary(los)
summary(c91a <- glm(los ~ procedure+ type, family=poisson, data=azcabgptca))
modelfit(c91a)
summary(c91b <- glm(los ~ procedure+ type, family=quasipoisson, data=azcabgptca))
modelfit(c91b)
library(sandwich)
sqrt(diag(vcovHC(c91a, type="HC0")))
# =====

# R
# =====
library(gamlss.tr)
gen.trun(0,"PO", type="left", name="leftr")
summary(c91c <- gamlss(los~ procedure+type, data=azcabgptca, family=POleftr))
# =====

# Section 9.2.1 Truncated models
# -----

# Table 9.2 R Left-Truncated at 3 Poisson
# =====
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
summary(plt <- gamlss(docvis~outwork + age,data=rwm1984,family=PO))
library(gamlss)
library(gamlss.tr)
pltvis<-subset(rwm1984, rwm1984$docvis>3)
summary(lt3po <- gamlss(docvis~outwork+age, family=trun(3, "PO", "left"), data=pltvis))
# -----
pltvis<-subset(rwm1984, rwm1984$docvis>3)      # alternative method
gen.trun(3, "PO", "left")                      # saved globally for session
summary(lt3po <- gamlss(docvis~outwork+age, family=POleft, data=pltvis))
# =====

Table 9.3 R: Right-Truncated Poisson : cut=10
=====
rtp<-subset(rwm1984, rwm1984$docvis<10)
summary(rtpo <- gamlss(docvis~outwork + age, data=rtp,
  family=trun(10, "PO", type="right")))
=====

# Section 9.2.2
# -----
# Table 9.4 R Left Censored Poisson at Cut=3
# =====
library(gamlss.cens)
library(survival)
library(COUNT)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
lcvis <- rwm1984
cy <- with(lcvis, ifelse(docvist<3, 3, docvis))
ci <- with(lcvis, ifelse(docvis<=3, 0, 1))
Surv(cy,ci, type="left")[1:100]

```

```

cbind(Surv(cy,ci, type="left")[1:50], rwm1984$docvis[1:50])
lcmdvis <- data.frame(lcvis, cy, ci )
rm(cy,ci)
gen.cens("PO",type="left")
lcat30<-gamlss(Surv(cy, ci, type="left") ~ outwork + age,
  data=lcvis, family=POlc)
summary(lcat30)
# =====

# Table 9.5 R Right censored Poisson at 10
# =====
library(gamlss.cens)
library(survival)
rcvis <- rwm1984
cy <- with(rcvis, ifelse(docvis>=10, 9, docvis))
ci <- with(rcvis, ifelse(docvis>=10, 0, 1))
rcvis <- data.frame(rcvis, cy, ci )
rm(cy,ci)
gen.cens("PO",type="right")
summary(rcat30<-gamlss(Surv(cy, ci) ~ outwork + age,
  data=rcvis, family=POrc, n.cyc=100))
# =====

# Section 9.3
# -----
# Table 9.6 R: Poisson-Poisson Finite Mixture Model
# =====
# flexmix only allows "gaussian", "binomial", "poisson", and "Gamma" families
library(COUNT)
library(flexmix)
data(fishing)
attach(fishing)
fmm_pg <- flexmix(totabund~meandepth + offset(log(sweptarea)), data=rwm1984, k=2,
  model=list(FLXMRglm(totabund~., family="poisson"),
    FLXMRglm(totabund~., family="poisson")))
parameters(fmm_pg, component=1, model=1)
parameters(fmm_pg, component=2, model=1)
summary(fmm_pg)
# =====

# Section 9.4
# -----

#Table 9.7 R: GAM
# =====
library(COUNT)
library(mgcv)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
summary(pglm <- glm(docvis ~ outwork + age + female + married +
  edlevel2 + edlevel3 + edlevel4, family=poisson, data=rwm1984))
summary(pgam <- gam(docvis ~ outwork + s(age) + female + married +
  edlevel2 + edlevel3 + edlevel4, family=poisson, data=rwm1984))
plot(pgam)
# =====

```

```

# Section 9.6.1
#-----

# Table 9.8 R: GEE
# =====
library(COUNT)
library(gee)
data(medpar)
summary(pgee <- gee(los ~ hmo + white + age80 + type2 + type3,
                   data=medpar, id=medpar$provnum,
                   corstr='exchangeable', family=poisson))
# =====

# Section 9.6.2
# -----
# Table 9.9 R Random Intercept Poisson
# =====
library(gamlss.mx)
summary(rip <- gamlssNP(los ~ hmo + white + type2 + type3,
                       random=~1|provnum, data=medpar,
                       family=PO, mixture="gq", K=20))
# =====

# Section 9.7
# -----

# Table 9.10 R Generalized Waring Regression
# =====
library(COUNT)
library(GWRM)
data(rwm5yr)
rwm1984 <- subset(rwm5yr, year==1984)
war <- GWRM.fit(docvis ~ outwork + age + female + married, data=rwm1984)
GWRM.display(war)
# =====

# Section 9.8
# -----

# Table 9.11 R: Bayesian Poisson MCMC
# =====
library(COUNT)
library(MCMCpack)
data(medpar)
summary(poi <- glm(los ~ hmo + white + type2 + type3,
                  family=poisson, data=medpar))
confint.default(poi)
summary(poibayes <- MCMCpoisson(los ~ hmo + white + type2 + type3,
                                burnin = 5000, mcmc = 100000, data=medpar))
# =====

#EXTRA
# graphic of probabilities from counts of 0-8, with a mean of 4 and inverted alpha=2 (or .5)
library(gamlss)
x<-seq(0, 8, 1)
y<-dPIG(x, 4, 2)
plot(x, y, xlab="Count", ylab="Probability values", main=" PIG model with mu=4
and inverted alpha=0.5")

```