July 9, 2014

MCD-Figures-Code

Joseph M Hilbe, *Arizona State University*

Modeling Count Data

Cambridge University Press : 2014

Joseph M. Hilbe

hilbe@asu.edu or j.m.hilbe@gmail.com

Figure Code: R and Stata

7 July, 2014

```
Code for Figure 1.2a
==========================================
    PIG = (light green) sharp high initial mass with peak at about 1.0
Others: from highest probability at 0 to lowest
    GP    = purple
    NB2 =  black
    NB1 =  blue
    POI  =  red
=============================================00
```

FIG 1.2a   R

```
obs <- 15; mu <- 4; y <- (0:140)/10; 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 = exp( (-(y-mu[i])^2)/(alpha*2*y*mu[i]^2)) *
(sqrt(1/(alpha*2*pi*y^3)))
    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))
}
```

FIG 1.2a Stata

```
clear
set obs 15
gen byte mu = 4
gen byte y = _n-1
gen yp =  (exp(-mu)*mu^y)/exp(lngamma(y+1))
gen alpha = .5
gen amu = mu*alpha
gen ynb = exp(y*ln(amu/(1+amu)) - (1/alpha)*ln(1+amu) + lngamma(y
+1/alpha) /*
   */  - lngamma(y+1) - lngamma(1/alpha))
gen ygp = exp(ln((1-alpha)*mu) + (y-1)*ln((1-alpha)*mu+alpha*y) -  /*
   */  (1-alpha)*mu - alpha*y - lngamma(y+1))
gen ynb1 = exp(lngamma(mu/alpha + y) - lngamma(y+1) - lngamma(mu/alpha) +
/*
   */  (mu/alpha)*ln(1/(1+alpha)) + y*ln(1-1/(1+alpha)))
lab var yp "Poisson"
lab var ynb "NB2;alpha=.5"
lab var ynb1 "NB1; alpha=.5"
lab var ygp "generalized Poisson, alpha=.5"
graph twoway connected ygp ynb1 ynb yp  y, ms(s D T d) /*
   */  title("POI vs NB vs NB1 vs GP distributions: MEAN = 4; a=0.5")
```

FIG 1.2b Stata

```
* MCD Figure 1.2
clear
set obs 15
gen byte mu = 4
gen byte y = _n-1
gen yp =  (exp(-mu)*mu^y)/exp(lngamma(y+1))
gen alpha = .5
gen amu = mu*alpha
gen ynb = exp(y*ln(amu/(1+amu)) - (1/alpha)*ln(1+amu) + lngamma(y
+1/alpha) /*
   */  - lngamma(y+1) - lngamma(1/alpha))
gen ypig = exp((-(y-mu)^2)/(alpha*2*y*mu^2)) *
(sqrt(1/(alpha*2*_pi*y^3)))
gen ygp = exp(ln((1-alpha)*mu) + (y-1)*ln((1-alpha)*mu+alpha*y) -  /*
```

```
   */  (1-alpha)*mu - alpha*y - lngamma(y+1))
gen ynb1 = exp(lngamma(mu/alpha + y) - lngamma(y+1) - lngamma(mu/alpha) +
/*
   */  (mu/alpha)*ln(1/(1+alpha)) + y*ln(1-1/(1+alpha)))
lab var yp "Poisson"
lab var ynb "NB2"
lab var ynb1 "NB1"
lab var ypig "PIG"
lab var ygp "generalized Poisson"
graph twoway connected ygp ypig ynb1 ynb yp  y, ms(s D T d S) /*
   */  title("POI | NB | NB1 | GP |PIG distributions: MEAN = 4; a=0.5")
```

Fig 2.2   R

```
obs <- 15
mu <- 4
y <- (0:140)/10
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 = exp((-(y-mu[i])^2)/(alpha*2*y*mu[i]^2)) *
        (sqrt(1/(alpha*2*pi*y^3)))
    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")
    for (j in 1:5)
        lines(y, all.lines[[j]], ylim = c(0, ymax), col=cols[j], type='l',
lty=j)
}
```

Fig 3.1 Stata

```
* Hilbe, Modeling Count Data, Figure 3.1
use rwm1984, clear
qui glm docvis outwork age, fam(poisson)
predict mu
count
gen nobs = e(N)
local i 0
local newvar "pr`i'"
* Predicted probability at each day
 while `i' <=25 {
   local newvar "pr`i'"
   qui gen `newvar' =  poissonp(mu, `i')
   local i = `i' + 1
}
quietly gen cnt = .
quietly gen observ = .
quietly gen expect = .
local i 0
*: Observed and expected docvis
while `i' <=25 {
   local obs = `i' + 1
   replace cnt = `i' in `obs'
   tempvar obser
   gen `obser' = `e(depvar)' ==`i'  /* (docvis==`i') */
   sum `obser'
   replace observ = r(mean)* nobs in `obs'
   sum pr`i'
   replace expect = r(mean)* nobs in `obs'
   local i = `i' + 1
}
*: Preparation for table
gen byte count = cnt
gen diff = observ - expect
drop cnt pr0-pr25 nobs mu
list count observ expect diff in 1/21
lab var expect "Expected days"
lab var observ "Observed days"
label var count "Number visits to Physician"
twoway scatter expect observ count, c(l l) ms(T d) ///
      title(Observed vs Expected visits) ytitle(Count of patients)
```

Ch 4.5.1

```
qui {
qui nbreg docvis outwork age female married edlevel2 edlevel3 edlevel4
predict mu
gen alpha = 2.250628
local i 0
local newvar "pr`i'"
```

```
while `i' <=20 {
 local newvar "pr`i'"
 qui gen `newvar' = exp(`i'*log((alpha*mu)/(1+alpha*mu)) -
(1/alpha)*log(1+(alpha*mu)) + exp(lngamma(`i' + 1/alpha))-
exp(lngamma(`i'+1)) - exp(lngamma(1/alpha)))
 local i = `i' + 1
}
quietly gen cnt = .
quietly gen obpr = .
quietly gen prpr = .
local i 0
while `i' <=20 {
 local obs = `i' + 1
 replace cnt = `i' in `obs'
 tempvar obser
 gen `obser' = (`e(depvar)'==`i')
 sum `obser'
 replace obpr = r(mean) in `obs'
 sum pr`i'
 replace prpr = r(mean) in `obs'
 local i = `i' + 1
}
* Note: starred lines produce Table
* gen obsprop = obpr*100
* gen preprop = prpr*100
gen byte count = cnt
* gen diffprop = obsprop - preprop
* format obsprop preprop diffprop %8.3f
* list count obsprop preprop diffprop
label var prpr "Predicted days"
label var obpr "Observed days"
label var count "# of Visits"
}
twoway scatter prpr obpr count, c(l l) ms(T d) title(Observed vs Predicted
Probabilities) ytitle(Probability of Physician Visits per Year)
```

Fig 5.3 Stata

```
* Hilbe, Modeling Count Data, Fig 5.3
use rwm1984, clear
global xvar "outwork age female married edlevel2 edlevel3 edlevel4"
glm docvis $xvar, fam(poi) vce(robust) nolog nohead
countfit docvis $xvar, prm nbreg max(12)
```

Fig 5.4 Stata

```
* Hilbe, Modeling Count Data, Fig 5.4
use rwm1984, clear
qui {
qui nbreg docvis outwork age married female edlevel2 edlevel3 edlevel4
predict mu
local alpha = e(alpha)
gen amu = mu* e(alpha)
```

```
local i 0
local newvar "pr`i'"
while `i' <=15 {
local newvar "pr`i'"
qui gen `newvar' = exp(`i'*ln(amu/(1+amu)) - (1/`alpha')*ln(1+ amu) +
lngamma(`i' +1/`alpha') /*
  */  - lngamma(`i'+1) - lngamma(1/`alpha'))
local i = `i' + 1
}
quietly gen cnt = .
quietly gen obpr = .
quietly gen prpr = .
local i 0
while `i' <=15 {
 local obs = `i' + 1
 replace cnt = `i' in `obs'
 tempvar obser
 gen `obser' = (`e(depvar)'==`i')
 sum `obser'
 replace obpr = r(mean) in `obs'
 sum pr`i'
 replace prpr = r(mean) in `obs'
 local i = `i' + 1
}
gen byte count = cnt
label var prpr "NB2 - Predicted"
label var obpr "NB2 - Observed"
label var count "Count"
}
twoway scatter prpr obpr count, c(l l) ms(T d) title("docvis Observed vs
Predicted Probabilities") sub("Negative Binomial") ytitle(Probability of
Physician Visits)
```

Fig 6.1 Stata

```
* Hilbe, Modeling Count Data Fig 6.1
use rwm1984 clear
qui {
qui nbreg docvis outwork age married female edlevel2 edlevel3 edlevel4
predict mu
local alpha = e(alpha)
gen amu = mu* e(alpha)
local i 0
local newvar "pr`i'"
while `i' <=15 {
local newvar "pr`i'"
  qui gen `newvar' = exp((-(`i'-mu)^2)/(`alpha'*2*`i'*mu^2)) *
(sqrt(1/(`alpha'*2*_pi*`i'^3)))
local i = `i' + 1
}
quietly gen cnt = .
quietly gen obpr = .
```

```
quietly gen prpr = .
local i 0
while `i' <=15 {
 local obs = `i' + 1
 replace cnt = `i' in `obs'
 tempvar obser
 gen `obser' = (`e(depvar)'==`i')
 sum `obser'
 replace obpr = r(mean) in `obs'
 sum pr`i'
 replace prpr = r(mean) in `obs'
 local i = `i' + 1
}
gen byte count = cnt
label var prpr "NB2 - Predicted"
label var obpr "NB2 - Observed"
label var count "Count"
}
twoway scatter prpr obpr count, c(l l) ms(T d) title("docvis Observed vs
Predicted Probabilities") sub("PIG") ytitle(Probability of Physician
Visits)
```

Fig 8.1 Stata

```
* Hilbe, Modeling Coun Data  Figure 8.1
use azprocedure, clear
hist los if los<40, title("LOS for full Heart Procedure Data") discrete
xlab( 5 8.83 "mean" 10 15 20 25 30 35) percent
```

Fig 9.1 R

```
# Hilbe, Modeling Count Data  Figure 9.1
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)
```