Example 3: Predicting Count Outcomes using 4 Types of Poisson and Negative Binomial Models
(syntax and output available for SAS, STATA, and R electronically)

The data for this example come from a study about the effects of emotion regulation strategy (none=control, cognitive reappraisal, or suppression) in predicting the aggressive verbalizations of persons with or without a history of perpetrating intimate partner violence (IPV). The planned analysis was a two-way between-groups ANOVA for 3 levels of strategy condition by 2 levels of IPV history. Here is the paper published about these data (with similar results, although their models included covariates and so their sample differed slightly):

Maldonado, R. C., DiLillo, D., & **Hoffman, L.** (2015). Can college students alter their intimate partner aggression-risk behaviors using emotion regulation strategies? An examination using I3 Theory. *Psychology of Violence, 5*(1), 46–55.

This example will examine the results of the same linear predictor using a general linear model (identity link + normal conditional distribution), as well as four types of generalized linear models with log links but different distributions: Poisson, negative binomial, zero-inflated Poisson, and zero-inflated negative binomial. In addition, the probability of being an extra zero is predicted with a logit link in the two zero-inflated variants. Because the relevant STATA options (using GLM to get conditional distribution fit, also NBREG, ZIP, and ZINB here) do not have denominator degrees of freedom, they were set to "none" in SAS GLIMMIX so that the SAS Wald test results (still labeled as t or F) will match those of STATA (using z or χ^2). In R, I am using the base R function GLM, the glm.nb function from package MASS, and the zeroinfl function from package pscl (each also using z or χ^2). In SAS (as shown in the online files only), I am still using GLIMMIX (even though these are not mixed-effects models) to get conditional distribution fit, as well as GENMOD for the zero-inflated model variants.

STATA Syntax for Importing and Preparing Data for Analysis:

```
// Defining global variable for file location to be replaced in code below
// \\Client\ precedes path in Virtual Desktop outside H drive;
global filesave "C:\Dropbox\24_PSQF6270\PSQF6270_Example3"

// Import Example 3 Excel data
import excel "$filesave\Excel_Example3.xlsx", clear

// STATA code to create indicator-dummy-coded predictor variables
gen NvC=.  // Make 2 new empty variables
gen NvS=.
replace NvC=0 if ercond==1  // Replace if ercond=1=None
replace NvS=0 if ercond==1
replace NvC=1 if ercond==2  // Replace if ercond=2=CogR
replace NvS=0 if ercond==2
replace NvC=0 if ercond==3  // Replace if ercond=3=Supp
replace NvS=1 if ercond==3
label variable IPV    "IPV: Inter-Partner Violence (0=N,1=Y)"
label variable ercond "ercond: 1=None, 2=CogR, 3=Supp"
label variable aggr   "aggr: Aggressive Verbalizations"
label variable NvC    "NvC: Condition None=0 vs. CogR=1"
label variable NvS    "NvS: Condition None=0 vs. Supp=1"
// Filter to only cases complete on all variables to be used below
egen nummiss = rowmiss(aggr IPV ercond)
drop if nummiss>0
```

R Syntax for Importing and Preparing Data for Analysis (after loading packages *readxl*, *TeachingDemos*, *psych*, *multcomp*, *prediction*, *MASS*, and *pscl*, as shown online):

```
# Define variables for working directory and data name
filesave = "C:\\Dropbox/24_PSQF6270/PSQF6270_Example3/"
filename = "Excel_Example3.xlsx"
setwd(dir=filesave)
```

```
# Import Example 3 Excel data
Example3 = read_excel(paste0(filesave,filename))
# Convert to data frame without labels to use for analysis
Example3 = as.data.frame(Example3)

# R code to create indicator-dummy-coded binary predictors
Example3$NvC=NA; Example3$NvS=NA  # Make 2 new empty variables
Example3$NvC[which(Example3$ercond==1)]=0  # Replace if ercond=None=1
Example3$NvS[which(Example3$ercond==1)]=0
Example3$NvC[which(Example3$ercond==2)]=1  # Replace if ercond=CogR=2
Example3$NvS[which(Example3$ercond==2)]=0
Example3$NvC[which(Example3$ercond==3)]=0  # Replace if ercond=Supp=3
Example3$NvS[which(Example3$ercond==3)]=1
# Label variables as comments only (not actually added to data)

# Filter to only cases complete on all variables to be used below
Example3 = Example3[complete.cases(Example3[ , c("aggr","IPV","ercond")]),]
```

Syntax and Condensed Output for Data Description:

```
display "STATA Cell Means for Aggressive Verbalizations"
bysort ercond IPV: tabstat aggr, statistics(n max sd mean semean)

display "STATA Histogram for Aggressive Verbalizations"
hist aggr, percent normal discrete width(1) start(0)
graph export "$filesave\STATA Overall Histogram.png", replace

display "STATA Histogram for Aggressive Verbalizations by Cell"
hist aggr, by(IPV ercond) percent normal discrete width(1) start(0)
graph export "$filesave\STATA By Cell Histogram.png", replace

print("R Cell Means for Aggressive Verbalizations Outcome")
describeBy(x=Example3$aggr,list(Example3$IPV,Example3$ercond))

# to save a plot: open a file, create the plot, then close the file
png(file = "R Histogram for Aggressive Verbalizations.png")  # open file
hist(x=Example3$aggr, freq=FALSE,
     ylab="Density",xlab="aggr: Aggressive Verbalizations") # axis labels
dev.off()  # close file
# I did not figure out how to make a separate histogram for each cell
```

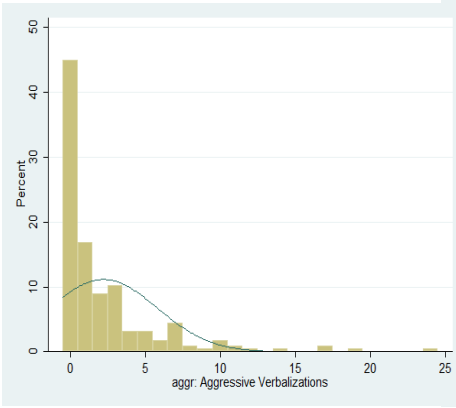
							Exp(Log Link)					
					Raw Data		Normal Dist		Poisson Dist		NegBin Dist	
IPV	Cond	N	Max	SD	Mean	SE	Mean	SE	Mean	SE	Mean	SE
IPV=No	None	53	24	4.32	2.72	0.59	2.72	0.47	2.72	0.23	2.72	0.52
IPV=Yes	None	21	9	2.82	3.05	0.62	3.05	0.74	3.05	0.38	3.05	0.92
IPV=No	CogR	53	11	1.95	0.92	0.27	0.92	0.47	0.92	0.13	0.92	0.21
IPV=Yes	CogR	20	1	0.04	0.15	0.08	0.15	0.76	0.15	0.09	0.15	0.10
IPV=No	Supp	54	14	3.30	2.35	0.45	2.35	0.46	2.35	0.21	2.35	0.46
IPV=Yes	Supp	24	19	5.27	4.46	1.08	4.46	0.70	4.46	0.43	4.46	1.23

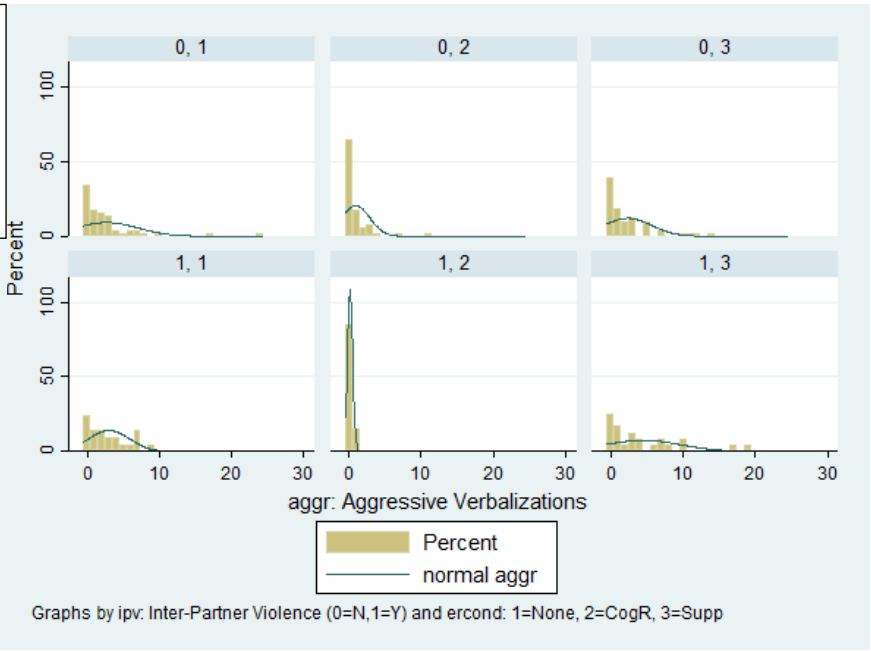
Above are the outcome **cell means** for each combination of IPV by emotion regulation condition that our model is trying to capture, along with the per-cell maximum and SD.

What we will see in this example is that the **cell means will stay the same** across models—this is because we will use the same linear predictor (while changing the link and conditional distribution) across all models.

Thus, what will change across models are the inferences about these cell differences (which come from their standard errors, which result from the conditional distribution chosen).

Look how non-aggressive our sample is! That's great for them, but not if we expect to use a general linear model (i.e., ANOVA) to predict this outcome… However, below is only the marginal distribution of Y. Maybe the residuals will look more normal?? Nope… (left)





Same Linear Predictor to be used across ALL models:

$$\widehat{Aggr}_i = \beta_0 + \beta_1(IPV_i) + \beta_2(NoneVsCogR_i) + \beta_3(NoneVsSupp_i) + \beta_4(IPV_i)(NoneVsCogR_i) + \beta_5(IPV_i)(NoneVsSupp_i)$$

Model-implied slope of IPV history (no vs. yes) per ER condition:

$$IPV\ slope = \beta_1 + \beta_4(NoneVsCogR_i) + \beta_5(NoneVsSupp_i)$$

Model-implied slope for condition differences (none, cognitive reappraisal, suppression) per IPV:

$$None\ vs.CogR\ slope = \beta_2 + \beta_4(IPV_i)$$
$$None\ vs.Supp\ slope = \beta_3 + \beta_5(IPV_i)$$
$$CogR\ vs.Supp\ slope = [\beta_3 + \beta_5(IPV_i)] - [\beta_2 + \beta_4(IPV_i)]$$

STATA GLM: Model using an Identity Link and a Normal Conditional Distribution

This model would otherwise be known as ANOVA if it were estimated using ordinary least squares, which is equivalent to residual ML (REML). Although the LL value is using ML estimation, the SEs and Wald tests use the REML estimate of the residual variance instead for comparability with R GLM.

```
display "STATA Link=Identity Dist=Normal Model using glm"
display "SEs use REML=OLS instead for Comparability with R"
glm aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, ml link(identity) family(gaussian) nolog
```

```
Generalized linear models                         No. of obs         =        225
Optimization     : ML                             Residual df        =        219
                                                  Scale parameter    =   11.61748
Deviance         =  2544.228359                   (1/df) Deviance    =   11.61748
Pearson          =  2544.228359                   (1/df) Pearson     =   11.61748 → REML res variance
Variance function: V(u) = 1                       [Gaussian]
Link function    : g(u) = u                       [Identity]
                                                  AIC                =   5.316693 → not usual AIC!
Log likelihood   = -592.1279267                   BIC                =   1358.102 → not usual BIC!
```

```
------------------------------------------------------------------------------
             |                 OIM
        aggr |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         IPV |   .3306379   .8670718     0.38   0.703    -1.368792    2.030067  Beta1
         NvC |  -1.792453   .6532266    -2.74   0.006    -3.072753   -.5121523  Beta2
         NvS |  -.3651293   .6501953    -0.56   0.574    -1.639489    .9092302  Beta3
 c.IPV#c.NvC |  -1.105166   1.237154    -0.89   0.372    -3.529944    1.319611  Beta4
 c.IPV#c.NvS |   1.775844   1.196816     1.48   0.138    -.5698726     4.12156  Beta5
       _cons |   2.716981   .4619009     5.88   0.000     1.811672     3.62229  Beta0
------------------------------------------------------------------------------
```

```
display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 1184.2559

// DF=5 Multiv Wald Test of Model
test (c.IPV=0)(c.NvC=0)(c.NvS=0)(c.IPV#c.NvC=0)(c.IPV#c.NvS=0)

           chi2(  5) =   27.82
         Prob > chi2 =    0.0000

// DF=2 Multiv Wald Test of Interaction
test (c.IPV#c.NvC=0)(c.IPV#c.NvS=0)

           chi2(  2) =    5.70
         Prob > chi2 =    0.0579
```

In the intended ANOVA model, this interaction using denominator DF was $F(2, 219) = 2.85$, $p = .0600$!

Btw, F * # slopes = χ^2

```
// Yhat cell means in original count model scale per condition
margins, at(c.IPV=(0(1)1) c.NvC=0 c.NvS=0) predict(xb) // None
margins, at(c.IPV=(0(1)1) c.NvC=1 c.NvS=0) predict(xb) // CogR
margins, at(c.IPV=(0(1)1) c.NvC=0 c.NvS=1) predict(xb) // Supp

// Simple slopes of IPV per condition
lincom c.IPV*1 + c.IPV#c.NvC*0 + c.IPV#c.NvS*0  // No vs Yes IPV: None
lincom c.IPV*1 + c.IPV#c.NvC*1 + c.IPV#c.NvS*0  // No vs Yes IPV: CogR
lincom c.IPV*1 + c.IPV#c.NvC*0 + c.IPV#c.NvS*1  // No vs Yes IPV: Supp

// Simple slopes of condition per IPV
lincom c.NvC*1  + c.NvS*0 + c.IPV#c.NvC*0  + c.IPV#c.NvS*0  // None vs CogR: IPV=No
lincom c.NvC*0  + c.NvS*1 + c.IPV#c.NvC*0  + c.IPV#c.NvS*0  // None vs Supp: IPV=No
lincom c.NvC*-1 + c.NvS*1 + c.IPV#c.NvC*0  + c.IPV#c.NvS*0  // CogR vs Supp: IPV=No
lincom c.NvC*1  + c.NvS*0 + c.IPV#c.NvC*1  + c.IPV#c.NvS*0  // None vs CogR: IPV=Yes
lincom c.NvC*0  + c.NvS*1 + c.IPV#c.NvC*0  + c.IPV#c.NvS*1  // None vs Supp: IPV=Yes
lincom c.NvC*-1 + c.NvS*1 + c.IPV#c.NvC*-1 + c.IPV#c.NvS*1  // CogR vs Supp: IPV=Yes

// Simple slopes for interaction contrasts
lincom c.IPV#c.NvC*1  + c.IPV#c.NvS*0  // No/Yes IPV differ btw None/CogR
lincom c.IPV#c.NvC*0  + c.IPV#c.NvS*1  // No/Yes IPV differ btw None/Supp
lincom c.IPV#c.NvC*-1 + c.IPV#c.NvS*1  // No/Yes IPV differ btw CogR/Supp
```

(Voluminous STATA output for margins and lincom not shown; see R output below)

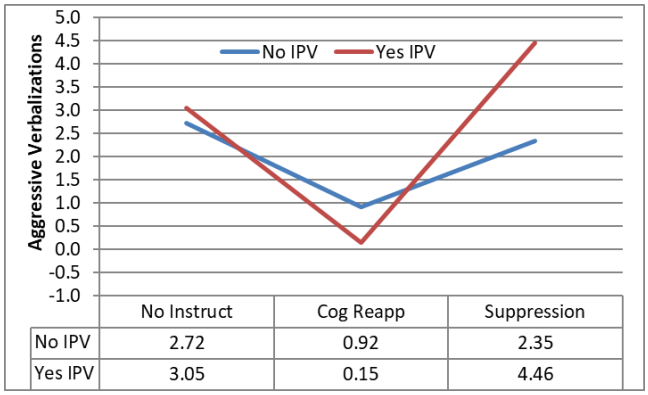
R GLM: Model using an Identity Link and a Normal Conditional Distribution

```
print("R Link=Identity Dist=Normal Model using glm for ML Estimation")
print("Uses t instead of z for Univariate Wald tests")
ModelNorm = glm(data=Example3, family=gaussian(link="identity"),
                formula=aggr~1+IPV+NvC+NvS +IPV:NvC +IPV:NvS)
print("Print results with -2LL"); summary(ModelNorm)

Coefficients:
            Estimate Std. Error t value      Pr(>|t|)
(Intercept)  2.71698    0.46819  5.8032 0.00000002262  Beta0
IPV          0.33064    0.87887  0.3762      0.707126  Beta1
NvC         -1.79245    0.66211 -2.7072      0.007321  Beta2
NvS         -0.36513    0.65904 -0.5540      0.580123  Beta3
IPV:NvC     -1.10517    1.25399 -0.8813      0.379110  Beta4
IPV:NvS      1.77584    1.21310  1.4639      0.144658  Beta5
(Dispersion parameter for gaussian family taken to be 11.617481) → REML residual variance
```

```
    Null deviance: 2867.40 on 224 DDF → sum of squared Pearson residuals: empty model
Residual deviance: 2544.23 on 219 DDF → sum of squared Pearson residuals: this model

-2*logLik(ModelNorm)
'log Lik.' 1184.2559 (df=7) → -2LL

print("DF=5 Multiv Wald Test of Model with 8 digits")
NormR2 = glht(model=ModelNorm, linfct=c("IPV=0","NvC=0","NvS=0","IPV:NvC=0","IPV:NvS=0"))
print(summary(NormR2, test=Chisqtest()), digits="8") # Joint chi-square test

      Chisq DF     Pr(>Chisq)
1 27.817321  5 0.000039517063

print("DF=2 Multiv Wald Test of Interaction with 8 digits")
NormInt = glht(model=ModelNorm, linfct=c("IPV:NvC=0","IPV:NvS=0"))
print(summary(NormInt, test=Chisqtest()), digits="8") # Joint chi-square test

      Chisq DF  Pr(>Chisq)
1 5.6992729  2 0.057865353
```

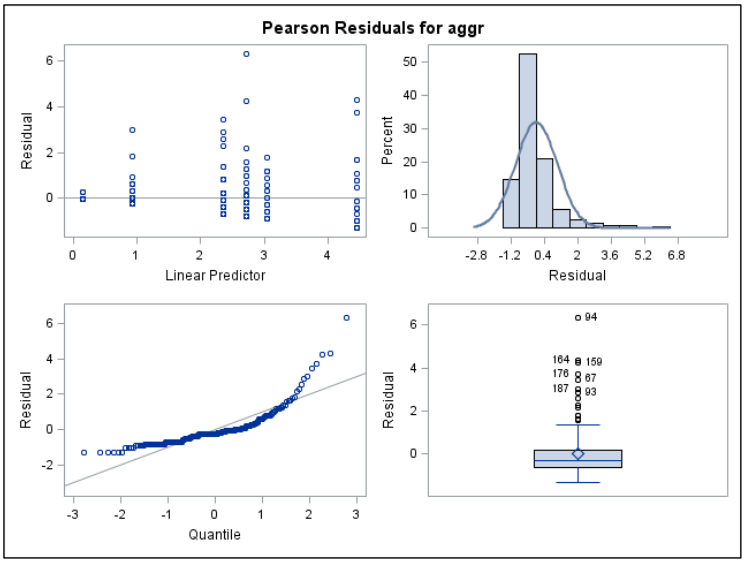
In the intended ANOVA model, this interaction using denominator DF was $F(2, 219) = 2.85$, $p = .0600$!

```
print("Yhat cell means in original count model scale per condition")
NormPredN = prediction(model=ModelNorm, type="response", at=list(IPV=0:1,NvC=0,NvS=0))
NormPredC = prediction(model=ModelNorm, type="response", at=list(IPV=0:1,NvC=1,NvS=0))
NormPredS = prediction(model=ModelNorm, type="response", at=list(IPV=0:1,NvC=0,NvS=1))
summary(rbind(NormPredN,NormPredC,NormPredS))

at(IPV) at(NvC) at(NvS) Prediction     SE      z              p     lower upper
      0       0       0     2.7170 0.4682 5.8032 0.000000006506  1.799354 3.635
      1       0       0     3.0476 0.7438 4.0975 0.000041771484  1.589831 4.505
      0       1       0     0.9245 0.4682 1.9747 0.048301670177  0.006901 1.842
      1       1       0     0.1500 0.7438 0.2017 0.840173402631 -1.307788 1.608 Uh-oh
      0       0       1     2.3519 0.4682 5.0233 0.000000507826  1.434225 3.269
      1       0       1     4.4583 0.7438 5.9941 0.000000002046  3.000545 5.916

print("Simple slopes: condition per IPV,
       IPV per condition, interactions")
print("SEs match SAS using REML instead")
NormSlopes = (summary(glht(model=ModelNorm,
linfct=rbind(
"No vs Yes IPV: None"    = c(0,1, 0,0, 0,0),
"No vs Yes IPV: CogR"    = c(0,1, 0,0, 1,0),
"No vs Yes IPV: Supp"    = c(0,1, 0,0, 0,1),
"None vs CogR: IPV=No" =  c(0,0, 1,0, 0,0),
"None vs Supp: IPV=No" =  c(0,0, 0,1, 0,0),
"CogR vs Supp: IPV=No" =  c(0,0,-1,1, 0,0),
"None vs CogR: IPV=Yes" = c(0,0, 1,0, 1,0),
"None vs Supp: IPV=Yes" = c(0,0, 0,1, 0,1),
"CogR vs Supp: IPV=Yes" = c(0,0,-1,1,-1,1),
"No/Yes IPV diff btw None/CogR" = c(0,0,0,0, 1,0),
"No/Yes IPV diff btw None/Supp" = c(0,0,0,0, 0,1),
"No/Yes IPV diff btw CogR/Supp" = c(0,0,0,0,-1,1))),
test=adjusted("none"))); NormSlopes
```



	No Instruct	Cog Reapp	Suppression
No IPV	2.72	0.92	2.35
Yes IPV	3.05	0.15	4.46

```
Linear Hypotheses:
                                          Estimate Std. Error z value   Pr(>|z|)
No vs Yes IPV: None == 0                   0.33064    0.87887  0.3762   0.706762
No vs Yes IPV: CogR == 0                  -0.77453    0.89447 -0.8659   0.386539
No vs Yes IPV: Supp == 0                   2.10648    0.83618  2.5192   0.011763

None vs CogR: IPV=No == 0                 -1.79245    0.66211 -2.7072   0.006786
None vs Supp: IPV=No == 0                 -0.36513    0.65904 -0.5540   0.579558
CogR vs Supp: IPV=No == 0                  1.42732    0.65904  2.1658   0.030330

None vs CogR: IPV=Yes == 0                -2.89762    1.06494 -2.7209   0.006510
None vs Supp: IPV=Yes == 0                 1.41071    1.01847  1.3851   0.166011
CogR vs Supp: IPV=Yes == 0                 4.30833    1.03196  4.1749 0.00002981

No/Yes IPV differ btw None/CogR == 0 -1.10517    1.25399 -0.8813   0.378144
No/Yes IPV differ btw None/Supp == 0  1.77584    1.21310  1.4639   0.143224
No/Yes IPV differ btw CogR/Supp == 0  2.88101    1.22445  2.3529   0.018627
```



RESIDUAL Output (from SAS): What about that whole non-normal residuals thing? Yep, it's still an issue… in addition, **the variance appears to grow with the mean**.

A data transformation is not going to make this better. What should we do instead? **We get a new model.** Let's try using a log link (to keep the predicted counts positive) and start with a Poisson conditional distribution (in which the conditional variance is supposed to be the same as the conditional mean, which means it is non-constant across the predicted count outcome).

STATA GLM: Model using a Log Link and a Poisson Conditional Distribution

I am using STATA GLM (instead of the direct POISSON function) to obtain the Pearson χ^2 / DF statistic that indicates how well our residuals fit a Poisson conditional distribution.

```
display "STATA Link=Log Dist=Poisson Model using glm"
glm aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, ml link(log) family(poisson) nolog

Generalized linear models                         No. of obs      =        225
Optimization     : ML                             Residual df     =        219
                                                  Scale parameter =          1
Deviance         =  793.6859856                   (1/df) Deviance =   3.624137
Pearson          =  1028.539634                   (1/df) Pearson  =   4.696528 → too high (1=good)
Variance function: V(u) = u                       [Poisson]
Link function    : g(u) = ln(u)                   [Log]
                                                  AIC             =   5.190062 → Not usual AIC!
Log likelihood   =  -577.881941                   BIC             =    -392.44 → Not usual BIC!
------------------------------------------------------------------------------
             |                 OIM
        aggr |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         IPV |   .1148393   .1502313     0.76   0.445    -.1796087    .4092872  Beta1
         NvC |  -1.077993   .1653862    -6.52   0.000    -1.402144   -.7538419  Beta2
         NvS |  -.1443183   .1217311    -1.19   0.236    -.3829069    .0942702  Beta3
 c.IPV#c.NvC |  -1.933488   .6134418    -3.15   0.002    -3.135811   -.7311636  Beta4
 c.IPV#c.NvS |   .5247327   .1994724     2.63   0.009     .1337739    .9156915  Beta5
       _cons |   .9995214   .0833333    11.99   0.000     .8361911    1.162852  Beta0
------------------------------------------------------------------------------
display "-2LL= " e(ll)*-2  // Print -2LL for model
-2LL= 1155.7639

// DF=5 Multiv Wald Test of Model R2
test (c.IPV=0)(c.NvC=0)(c.NvS=0)(c.IPV#c.NvC=0)(c.IPV#c.NvS=0)
           chi2(  5) =  111.08
         Prob > chi2 =    0.0000

// DF=2 Multiv Wald Test of Interaction
test (c.IPV#c.NvC=0)(c.IPV#c.NvS=0)
           chi2(  2) =   20.61
         Prob > chi2 =    0.0000

// Yhat cell means in log count model scale per condition
margins, at(c.ipv=(0(1)1) c.NvC=0 c.NvS=0) predict(xb) // None
margins, at(c.ipv=(0(1)1) c.NvC=1 c.NvS=0) predict(xb) // CogR
margins, at(c.ipv=(0(1)1) c.NvC=0 c.NvS=1) predict(xb) // Supp
```

From previous normal model: χ^2 (5) = 27.82, p < .0001

From previous normal model: χ^2 (2) = 5.70, p = .0579

```
// Y hat cell means in expected count data scale per ercond
margins, at(c.ipv=(0(1)1) c.NvC=0 c.NvS=0) // None
margins, at(c.ipv=(0(1)1) c.NvC=1 c.NvS=0) // CogR
margins, at(c.ipv=(0(1)1) c.NvC=0 c.NvS=1) // Supp
```

Exp(count) **margins** matches pred means from previous normal model

(All lincom statements are the same as before; see R output below)

I also added the option **irr** to these lincom statements (online)

```
display "STATA Link=Log Dist=Poisson Model using glm"
display "Request Incidence-Rate Ratios (via eform or irr)"
glm aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, eform ml link(log) family(poisson)
```

R GLM: Model using a Log Link and a Poisson Conditional Distribution

```
# R: save sample size and DDF for conditional distribution fit
DDFn=225     # What SAS GLIMMIX uses by default
DDFk=DDFn-6  # What STATA and SAS GENMOD use (N - # fixed effects)

print("R Link=Log Dist=Poisson Model using glm")
ModelPoisson = glm(data=Example3, family=poisson(link="log"),
                   formula=aggr~1+IPV+NvC+NvS +IPV:NvC +IPV:NvS)
print("Print results with -2LL"); summary(ModelPoisson)

Coefficients:
             Estimate Std. Error z value         Pr(>|z|)
(Intercept)  0.999521   0.083333 11.9943        < 2.2e-16  Beta0
IPV          0.114839   0.150231  0.7644         0.444619  Beta1
NvC         -1.077993   0.165386 -6.5180 0.00000000007123  Beta2
NvS         -0.144318   0.121731 -1.1856         0.235800  Beta3
IPV:NvC     -1.933488   0.613442 -3.1519         0.001622  Beta4
IPV:NvS      0.524733   0.199472  2.6306         0.008523  Beta5

(Dispersion parameter for poisson family taken to be 1)

-2*logLik(ModelPoisson)
'log Lik.' 1155.7639 (df=6)

print("Pearson Chi-Square / DF Index of Fit")
sum(residuals(ModelPoisson, type="pearson")^2)/DDFk  # STATA
[1] 4.696528

print("DF=5 Multiv Wald Test of Model with 8 digits")
PoissonR2 = glht(model=ModelPoisson,
                 linfct=c("IPV=0","NvC=0","NvS=0","IPV:NvC=0","IPV:NvS=0"))
print(summary(PoissonR2, test=Chisqtest()), digits="8") # Joint chi-square test

      Chisq DF     Pr(>Chisq)
1 111.07725  5 2.4255681e-22
```

I put these Wald test results inside a print function to ensure enough precision in its reporting for homework.

From previous normal model: χ^2 (5) = 27.82, p < .0001

```
print("DF=2 Multiv Wald Test of Interaction")
PoissonInt = glht(model=ModelPoisson, linfct=c("IPV:NvC=0","IPV:NvS=0"))
print(summary(PoissonInt, test=Chisqtest()), digits="8") # Joint chi-square test

      Chisq DF      Pr(>Chisq)
1 20.614194  2 0.000033395245
```

From previous normal model: χ^2 (2) = 5.70, p = .0579

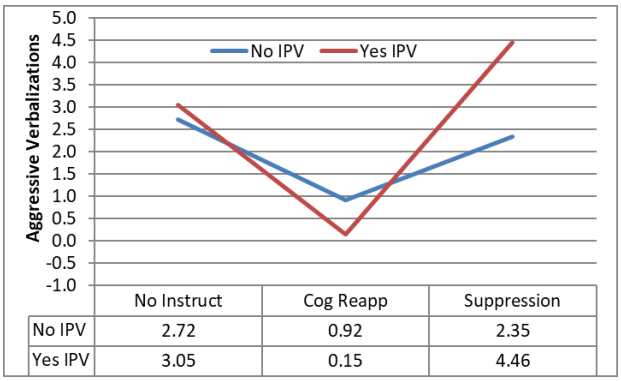
```
print("Yhat cell means in log count model scale per condition")
PoissonLogN = prediction(model=ModelPoisson, type="link", at=list(IPV=0:1,NvC=0,NvS=0))
PoissonLogC = prediction(model=ModelPoisson, type="link", at=list(IPV=0:1,NvC=1,NvS=0))
PoissonLogS = prediction(model=ModelPoisson, type="link", at=list(IPV=0:1,NvC=0,NvS=1))
summary(rbind(PoissonLogN,PoissonLogC,PoissonLogS))

at(IPV) at(NvC) at(NvS) Prediction      SE        z         p   lower    upper
      0       0       0    0.99952 0.08333  11.9943 3.808e-33  0.8362  1.16285
      1       0       0    1.11436 0.12500   8.9149 4.883e-19  0.8694  1.35936
      0       1       0   -0.07847 0.08333  -0.9417 3.464e-01 -0.2418  0.08486
      1       1       0   -1.89712 0.12500 -15.1770 5.025e-52 -2.1421 -1.65212
      0       0       1    0.85520 0.08333  10.2624 1.040e-24  0.6919  1.01853
      1       0       1    1.49478 0.12500  11.9582 5.882e-33  1.2498  1.73977
```

```
print("Yhat cell means in expected count data scale per condition")
PoissonCountN = prediction(model=ModelPoisson,type="response",at=list(IPV=0:1,NvC=0,NvS=0))
PoissonCountC = prediction(model=ModelPoisson,type="response",at=list(IPV=0:1,NvC=1,NvS=0))
PoissonCountS = prediction(model=ModelPoisson,type="response",at=list(IPV=0:1,NvC=0,NvS=1))
summary(rbind(PoissonCountN,PoissonCountC,PoissonCountS))
```

```
at(IPV) at(NvC) at(NvS) Prediction     SE       z          p   lower  upper
      0       0       0     2.7170 0.2264 12.0000 3.553e-33  2.2732 3.1607
      1       0       0     3.0476 0.3810  8.0000 1.244e-15  2.3010 3.7943
      0       1       0     0.9245 0.2264  4.0833 4.439e-05  0.4808 1.3683
      1       1       0     0.1500 0.3810  0.3938 6.938e-01 -0.5967 0.8967
      0       0       1     2.3519 0.2264 10.3873 2.831e-25  1.9081 2.7956
      1       0       1     4.4583 0.3810 11.7031 1.228e-31  3.7117 5.2050
```

Exp(count) **response** matches pred means from previous normal model

```
print("Simple slopes: condition per IPV,
       IPV per condition, interactions")
PoissonSlopes = (summary(glht(model=ModelPoisson,
linfct=rbind(
"No vs Yes IPV: None"    = c(0,1, 0,0, 0,0),
"No vs Yes IPV: CogR"    = c(0,1, 0,0, 1,0),
"No vs Yes IPV: Supp"    = c(0,1, 0,0, 0,1),
"None vs CogR: IPV=No" =  c(0,0, 1,0, 0,0),
"None vs Supp: IPV=No" =  c(0,0, 0,1, 0,0),
"CogR vs Supp: IPV=No" =  c(0,0,-1,1, 0,0),
"None vs CogR: IPV=Yes" = c(0,0, 1,0, 1,0),
"None vs Supp: IPV=Yes" = c(0,0, 0,1, 0,1),
"CogR vs Supp: IPV=Yes" = c(0,0,-1,1,-1,1),
"No/Yes IPV differ btw None/CogR" = c(0,0,0,0, 1,0),
"No/Yes IPV differ btw None/Supp" = c(0,0,0,0, 0,1),
"No/Yes IPV differ btw CogR/Supp" = c(0,0,0,0,-1,1))),
test=adjusted("none"))); PoissonSlopes
print("IRR effect sizes for simple slopes")
data.frame(OR=exp(PoissonSlopes$test$coefficients))
```



	No Instruct	Cog Reapp	Suppression
No IPV	2.72	0.92	2.35
Yes IPV	3.05	0.15	4.46

Comparing results across models: Identify/Normal vs. Log/Poisson:

	Identity Link, Normal Dist			Log Link, Poisson Dist		
Model Slope in Log Count	**Est**	**SE**	**p-value**	**Est**	**SE**	**p-value**
No vs Yes IPV: None	0.33	0.88	.707	0.11	0.15	.445
No vs Yes IPV: CogR	-0.77	0.89	.387	-1.82	0.59	.002
No vs Yes IPV: Supp	2.11	0.84	.012	0.64	0.13	.000
None vs CogR: IPV=No	-1.79	0.66	.007	-1.08	0.17	.000
None vs Supp: IPV=No	-0.37	0.66	.580	-0.14	0.12	.236
CogR vs Supp: IPV=No	1.43	0.66	.030	0.93	0.17	.000
None vs CogR: IPV=Yes	-2.90	1.06	.007	-3.01	0.59	.000
None vs Supp: IPV=Yes	1.41	1.02	.166	0.38	0.16	.016
CogR vs Supp: IPV=Yes	4.31	1.03	.000	3.39	0.59	.000
No/Yes IPV differ by None/CogR	-1.11	1.25	.378	-1.93	0.61	.002
No/Yes IPV differ by None/Supp	1.78	1.21	.143	0.52	0.20	.009
No/Yes IPV differ by CogR/Supp	2.88	1.22	.019	2.46	0.61	.000

The Poisson distribution has only one parameter—the mean, which is supposed to also be the conditional variance. But our Pearson χ^2 / DF = 4.697 result says that the average residual is 4.697 times as large as the Poisson distribution predicts (from conditional SD), indicating that this distribution fit is not good enough yet.

In count data it is often more reasonable to allow the variance to differ from the mean (usually to be greater, known as "over-dispersion"). There are multiple ways to do this; here we will use a negative binomial model to allow the residual variance to change as a quadratic function of the mean (called "NB-2"), which seems to be the most accepted approach.

STATA: Model using a Log Link and a Negative Binomial Conditional Distribution

```
display "STATA Link=Log Dist=Negative Binomial Model using nbreg"
display "nbreg gives LRT for scale factor that distinguishes NB from Poisson"
nbreg aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, nolog

Negative binomial regression                    Number of obs     =        225
                                                LR chi2(5)        =      42.88
Dispersion     = mean                           Prob > chi2       =     0.0000
Log likelihood = -409.78835                     Pseudo R2         =     0.0497
------------------------------------------------------------------------------
        aggr |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         IPV |   .1148393   .3591874     0.32   0.749    -.5891552    .8188337  Beta1
         NvC |  -1.077993   .2962567    -3.64   0.000    -1.658645   -.4973406  Beta2
         NvS |  -.1443183   .2732663    -0.53   0.597    -.6799104    .3912737  Beta3
 c.IPV#c.NvC |  -1.933488   .7700748    -2.51   0.012    -3.442807   -.4241687  Beta4
 c.IPV#c.NvS |   .5247327   .4925367     1.07   0.287    -.4406215    1.490087  Beta5
       _cons |   .9995214   .1927489     5.19   0.000     .6217405    1.377302  Beta0
-------------+----------------------------------------------------------------
    /lnalpha |   .4706335   .1516165                      .1734706    .7677963 log(k)
-------------+----------------------------------------------------------------
       alpha |   1.601008   .2427392                      1.189426    2.155012 k dispersion
------------------------------------------------------------------------------
LR test of alpha=0: chibar2(01) = 336.19                  Prob >= chibar2 = 0.000 → NB wins

display "STATA Link=Log Dist=Negative Binomial Model using glm"
display "glm gives conditional fit, scale factor estimated by ML"
glm aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, ml link(log) family(nbinomial ml)

Generalized linear models                       No. of obs      =        225
Optimization     : ML                           Residual df     =        219
                                                Scale parameter =          1
Deviance         =   219.006464                 (1/df) Deviance =    1.00003
Pearson          =  244.6296746                 (1/df) Pearson  =    1.11703 → Close to 1, hooray!
Variance function: V(u) = u+(1.601)u^2          [Neg. Binomial]
Link function    : g(u) = ln(u)                 [Log]
                                                AIC             =   3.695896 → not usual AIC!
Log likelihood   = -409.7883514                 BIC             =  -967.1195 → not usual BIC!
------------------------------------------------------------------------------
             |                 OIM
        aggr |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         IPV |   .1148393   .3591874     0.32   0.749    -.5891552    .8188337  Beta1
         NvC |  -1.077993   .2962567    -3.64   0.000    -1.658645   -.4973406  Beta2
         NvS |  -.1443183   .2732663    -0.53   0.597    -.6799104    .3912737  Beta3
 c.IPV#c.NvC |  -1.933487   .7700748    -2.51   0.012    -3.442806   -.4241685  Beta4
 c.IPV#c.NvS |   .5247327   .4925367     1.07   0.287    -.4406215    1.490087  Beta5
       _cons |   .9995214   .1927489     5.19   0.000     .6217405    1.377302  Beta0
------------------------------------------------------------------------------
Note: Negative binomial parameter estimated via ML and treated as fixed once estimated.

display "-2LL=" e(ll)*-2  // Print -2LL for model
-2LL = 819.5767
```

Poisson model −2LL =1155.76
−2ΔLL(df = 1) = 1155.76 – 819.58
= 336.19, $p < .001$
And the 1.12 fit above means the average residual is close what is predicted by NegBin!

(All margins and lincom statements are the same as for Poisson)

```
// DF=5 Multiv Wald Test of Model R2
test (c.IPV=0)(c.NvC=0)(c.NvS=0)(c.IPV#c.NvC=0)(c.IPV#c.NvS=0)
           chi2(  5) =  41.22
         Prob > chi2 =   0.0000
```

From normal model: $\chi^2(5) = 27.82$, $p < .0001$
From Poisson model: $\chi^2(5) = 111.08$, $p < .0001$

```
// DF=2 Multiv Wald Test of Interaction
test (c.IPV#c.NvC=0)(c.IPV#c.NvS=0)
           chi2(  2) =  10.47
         Prob > chi2 =   0.0000
```

From normal model: $\chi^2(2) = 5.70$, $p = .0579$
From Poisson model: $\chi^2(2) = 20.61$, $p < .0001$

```
// Save predicted counts per real person to dataset
predict predcount
corr predcount aggr     // Get corr of pred count with aggr
display "R2=" r(rho)^2  // Print R2 relative to empty model
R2=.11270409

display "STATA Link=Log Dist=Negative Binomial Model"
display "Request Incidence-Rate Ratios (via eform or irr)"
glm aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, eform ml link(log) family(nbinomial ml) nolog
```

I also added the option **irr** to the lincom statements (online)

R: Model using a Log Link and a Negative Binomial Conditional Distribution

```
print("R Link=Log Dist=Negative Binomial Model")
print("Using glm.nb add-on to glm from MASS package")
ModelNegBin = glm.nb(data=Example3, link=log,
                     formula=aggr~1+IPV+NvC+NvS +IPV:NvC +IPV:NvS)
print("Print results with -2LL"); summary(ModelNegBin)

Coefficients:
            Estimate Std. Error z value     Pr(>|z|)
(Intercept)  0.99952    0.19275  5.1856 0.0000002153  Beta0
IPV          0.11484    0.35919  0.3197     0.749181  Beta1
NvC         -1.07799    0.29626 -3.6387     0.000274  Beta2
NvS         -0.14432    0.27327 -0.5281     0.597414  Beta3
IPV:NvC     -1.93349    0.77007 -2.5108     0.012047  Beta4
IPV:NvS      0.52473    0.49254  1.0654     0.286710  Beta5

(Dispersion parameter for Negative Binomial(0.6246) family taken to be 1) → is 1/k instead
              Theta:  0.6246
          Std. Err.:  0.0947

-2*logLik(ModelNegBin)
'log Lik.' 819.5767 (df=7)

print("Scale factor in same k metric as SAS and STATA")
ModelNegBin$theta
[1] 1.6010081

print("Pearson Chi-Square / DF Index of Fit")
sum(residuals(ModelNegBin, type="pearson")^2)/DDFk  # STATA
[1] 1.1170305

print("Likelihood Ratio Test for Poisson vs NegBin")
DevTest=-2*(logLik(ModelPoisson)-logLik(ModelNegBin))
RegPvalue=pchisq((DevTest), df=1, lower.tail=FALSE); MixPvalue=RegPvalue/2
print("Test Statistic, Regular and Mixture P-values for DF=1")
DevTest; RegPvalue; MixPvalue
'log Lik.' 336.18718 (df=6)
'log Lik.' 4.3176629e-75 (df=6)
'log Lik.' 2.1588315e-75 (df=6)

print("DF=5 Multiv Wald Test of Model with 8 digits")
NegBinR2 = glht(model=ModelNegBin,
                linfct=c("IPV=0","NvC=0","NvS=0","IPV:NvC=0","IPV:NvS=0"))
print(summary(NegBinR2, test=Chisqtest()), digits="8") # Joint chi-square test

      Chisq DF     Pr(>Chisq)
1 41.217936  5 0.000000084780759
```

From normal model: χ^2 (5) = 27.82, p < .0001
From Poisson model: χ^2 (5) = 111.08, p < .0001

```
print("DF=2 Multiv Wald Test of Interaction")
NegBinInt = glht(model=ModelNegBin, linfct=c("IPV:NvC=0","IPV:NvS=0"))
print(summary(NegBinInt, test=Chisqtest()), digits="8") # Joint chi-square test

      Chisq DF      Pr(>Chisq)
1 10.47018  2    0.0053263459
```

From normal model: χ^2 (2) = 5.70, p = .0579
From Poisson model: χ^2 (2) = 20.61, p < .0001

```
# Syntax omitted for cell means, and simple effects (same syntax as for Poisson)
```

```
print("Save predicted counts and correlate with aggr")
Example3$PredCount = predict(ModelNegBin, type="response")
rPred = cor.test(Example3$PredCount, Example3$aggr, method="pearson")
print("R2"); rPred$estimate^2
0.11270409
```

Comparing results across models: Identify/Normal vs. Log/Poisson vs. Log/Negative Binomial:

	Identity Link, Normal Dist			Log Link, Poisson Dist			Log Link, Neg Bin Dist		
Model Slope in Log Count	**Est**	**SE**	**p-value**	**Est**	**SE**	**p-value**	**Est**	**SE**	**p-value**
No vs Yes IPV: None	0.33	0.88	.707	0.11	0.15	.445	0.11	0.36	.749
No vs Yes IPV: CogR	-0.77	0.89	.387	-1.82	0.59	.002	-1.82	0.68	.008
No vs Yes IPV: Supp	2.11	0.84	.012	0.64	0.13	.000	0.64	0.34	.058
None vs CogR: IPV=No	-1.79	0.66	.007	-1.08	0.17	.000	-1.08	0.30	.000
None vs Supp: IPV=No	-0.37	0.66	.580	-0.14	0.12	.236	-0.14	0.27	.597
CogR vs Supp: IPV=No	1.43	0.66	.030	0.93	0.17	.000	0.93	0.30	.002
None vs CogR: IPV=Yes	-2.90	1.06	.007	-3.01	0.59	.000	-3.01	0.71	.000
None vs Supp: IPV=Yes	1.41	1.02	.166	0.38	0.16	.016	0.38	0.41	.353
CogR vs Supp: IPV=Yes	4.31	1.03	.000	3.39	0.59	.000	3.39	0.70	.000
No/Yes IPV differ by None/CogR	-1.11	1.25	.378	-1.93	0.61	.002	-1.93	0.77	.012
No/Yes IPV differ by None/Supp	1.78	1.21	.143	0.52	0.20	.009	0.52	0.49	.287
No/Yes IPV differ by CogR/Supp	2.88	1.22	.019	2.46	0.61	.000	2.46	0.76	.001

Given the large amount of zero values, we should examine whether we have adequately addressed them—let's compare our currently winning negative binomial model with models using <u>Zero-Inflated Poisson</u> and <u>Zero-Inflated Negative Binomial</u> conditional distributions. These add a separate "zero-inflation" submodel that predicts the logit of being an "extra" zero relative to what is expected given a Poisson or Negative Binomial conditional distribution. Here, we are fitting empty zero-inflation models that contain only an intercept for the probability of being an extra 0—if it's small enough, we don't need a zero-inflation submodel at all!

<u>STATA</u> ZIP: Model using a Log Link and a Zero-Inflated Poisson Conditional Distribution

```
display "STATA Zero-Inflated Poisson Model"
display "Only intercept in zero-inflation model (predict logit of extra zero)"
zip aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, inflate(_cons)

Zero-inflated Poisson regression                Number of obs     =        225
                                                Nonzero obs       =        124
                                                Zero obs          =        101
Inflation model = logit                         LR chi2(5)        =      64.17
Log likelihood  = -489.5851                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
        aggr |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
aggr         |
         IPV |  -.0232234   .1536578    -0.15   0.880    -.3243872    .2779404
         NvC |  -.7697161   .2118299    -3.63   0.000    -1.184895   -.3545372
         NvS |  -.0763171   .1264847    -0.60   0.546    -.3242226    .1715883
 c.IPV#c.NvC |  -2.058436   .6469452    -3.18   0.001    -3.326425   -.7904467
 c.IPV#c.NvS |   .4808622   .2041397     2.36   0.018     .0807558    .8809686
       _cons |   1.399623   .0858963    16.29   0.000     1.231269    1.567977
-------------+----------------------------------------------------------------
inflate      |
       _cons |  -.5453023   .1649153    -3.31   0.001    -.8685302   -.2220743 → logit extra 0
------------------------------------------------------------------------------    ignore p-value

display "-2LL=" e(ll)*-2  // Print -2LL for model
-2LL= 979.17018
```

I used "coeflegend" to get the name of zero-inflation submodel intercept

```
nlcom 1/(1+exp(-1* _b[inflate:_cons])) // Probability of extra 0
------------------------------------------------------------------------------
        aggr |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _nl_1 |    .366955   .0383097     9.58   0.000     .2918695    .4420405 37% extra 0 values
------------------------------------------------------------------------------
```

R ZIP: Model using a Log Link and a Zero-Inflated Poisson Conditional Distribution

```
print("R Link=Log Dist=Zero-Inflated Poisson Model using pscl package")
print("Only intercept in zero-inflation model, predict logit of extra zero")
ModelZIP = zeroinfl(data=Example3, dist="poisson",link="logit",
                    formula=aggr~1+IPV+NvC+NvS +IPV:NvC +IPV:NvS | 1)
print("Print results with -2LL"); summary(ModelZIP)

Count model coefficients (poisson with log link):
             Estimate Std. Error z value  Pr(>|z|)
(Intercept)  1.399623   0.085896 16.2943 < 2.2e-16
IPV         -0.023224   0.153658 -0.1511 0.8798668
NvC         -0.769716   0.211830 -3.6337 0.0002794
NvS         -0.076317   0.126485 -0.6034 0.5462621
IPV:NvC     -2.058437   0.646945 -3.1818 0.0014637
IPV:NvS      0.480863   0.204140  2.3556 0.0184949

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value  Pr(>|z|)
(Intercept) -0.54530    0.16492 -3.3066 0.0009445 → logit extra 0 (ignore p-value)

-2*logLik(ModelZIP);
'log Lik.' 979.17018 (df=7)

print("Pearson Chi-Square / DF Fit")
sum(residuals(ModelZIP, type="pearson")^2)/(DDFk-1) # Not given in STATA
[1] 2.0623518

print("Get probability of being an extra 0")
ZIPprob=1/(1+exp(-1*ModelZIP$coefficients$zero)); ZIPprob
(Intercept)
 0.36695499
```

R ZINB: Model using a Log Link and a Zero-Inflated Negative Binomial Conditional Distribution (because STATA blew up!)

```
print("R Link=Log Dist=Zero-Inflated Negative Binomial Model using pscl package")
print("Only intercept in zero-inflation model, predict logit of extra zero")
ModelZINB = zeroinfl(data=Example3, dist="negbin",link="logit",
                     formula=aggr~1+IPV+NvC+NvS +IPV:NvC +IPV:NvS | 1)
print("Print results with -2LL"); summary(ModelZINB)

Count model coefficients (negbin with log link):
             Estimate Std. Error z value     Pr(>|z|)
(Intercept)  0.99953    0.19278  5.1849 0.0000002162
IPV          0.11485    0.35919  0.3198    0.7491508
NvC         -1.07798    0.29626 -3.6387    0.0002741
NvS         -0.14433    0.27327 -0.5282    0.5973929
IPV:NvC     -1.93361    0.77009 -2.5109    0.0120430
IPV:NvS      0.52476    0.49254  1.0654    0.2866862
Log(theta)  -0.47061    0.15180 -3.1003    0.0019334

Zero-inflation model coefficients (binomial with logit link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -11.108    230.777 -0.0481   0.9616 logit of extra 0 (ignore p-value)
Theta = 0.62462 → 1/k dispersion

-2*logLik(ModelZINB)
'log Lik.' 819.57674 (df=8) → Nearly identical to original negative binomial model
```

```
print("Pearson Chi-Square / DF Index of Fit")
sum(residuals(ModelZINB, type="pearson")^2)/(DDFk-1)
[1] 1.1170425 → Very similar to original negative binomial

print("Scale factor in same k metric as SAS and STATA")
1/ModelZINB$theta
[1] 1.6009695 → Very similar to original negative binomial

print("Get probability of being an extra 0")
ZINBprob=1/(1+exp(-1*ModelZINB$coefficients$zero)); ZINBprob
0.000014994787
```

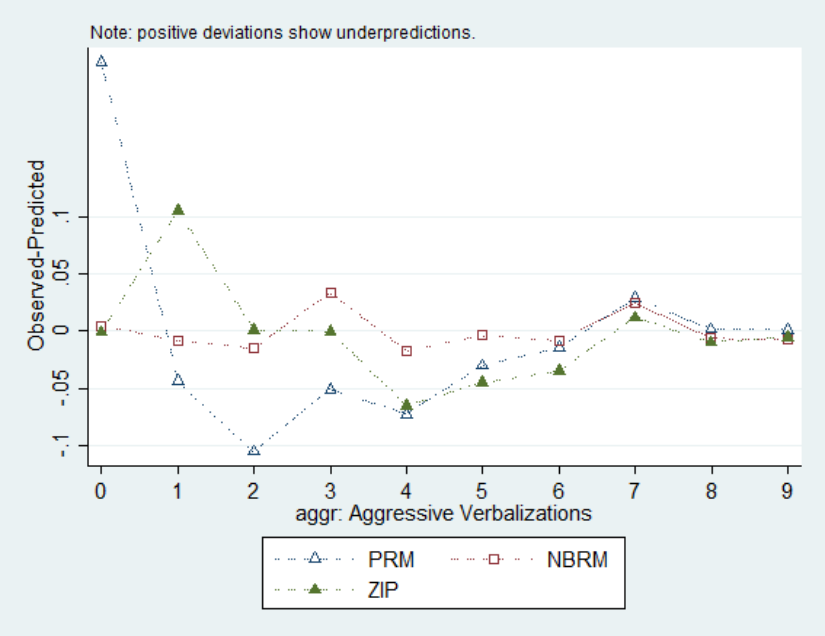
The logit of being an "extra 0" = −11.108 (with a crazy SE)! This is probability =.000014994787 of being an "extra" 0. So there are no extra 0 values in this distribution not already predicted by the negative binomial.

STATA user routine "countfit" can be used to compare count model conditional distributions:

```
// Run search below, then install from window that pops up
search countfit

display "STATA Countfit to Compare Fit of Alternative Count Model Distributions"
display "prm=Poisson, nbreg=Negative Binomial, zip=Zero-Inflated Poisson"
display "Results suggest NegBin fits best"
countfit aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, prm nbreg zip replace
graph export "$filesave\STATA Predicted Counts from Countfit.png", replace
```

Tests and Fit Statistics

PRM (Poisson)	BIC= 1188.260	AIC= 1167.764	Prefer	Over	Evidence	
vs NBRM	BIC= 857.489	dif= 330.771	NBRM	PRM	Very strong	
	AIC= 833.577	dif= 334.187	NBRM	PRM		
	LRX2= 336.187	prob= 0.000	NBRM	PRM	p=0.000	
vs ZIP	BIC= 1031.774	dif= 156.487	ZIP	PRM	Very strong	
	AIC= 990.780	dif= 176.983	ZIP	PRM		
	Vuong= .	prob= .	ZIP	PRM	p=.	
NBRM (NegBin)	BIC= 857.489	AIC= 833.577	Prefer	Over	Evidence	
vs ZIP	BIC= 1031.774	dif= -174.284	NBRM	ZIP	Very strong	
	AIC= 990.780	dif= -157.204	NBRM	ZIP		
ZIP	BIC= 1031.774	AIC= 990.780	Prefer	Over	Evidence	



Left: This plot from STATA countfit shows the match between the model-predicted counts and the actual counts for three models: PRM=Poisson, NBRM=Negative Binomial, and ZIP=zero-inflated Poisson. I did not add ZINB given that it blew up.

The NBRM has the closest match to the observed counts (smallest deviations), consistent with our Pearson χ^2 / DF results.

One last model—does the log of the negative binomial scale parameter need to differ by the same linear predictor used for the log of the expected count? Let's see—this is called a Heterogeneous Negative Binomial model. It is not directly available in SAS, but it can be programmed in NLMIXED, as I found here by Robin High. I searched but did not find it in R (although I'm sure it's in there somewhere).

```
display "STATA Heterogeneous Negative Binomial Model"
display "lnalpha gives linear model to predict log of scale factor"
gnbreg aggr c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS, ///
       lnalpha(c.IPV c.NvC c.NvS c.IPV#c.NvC c.IPV#c.NvS)
display "-2LL=" e(ll)*-2  // Print -2LL for model
```

STATA Output for Heterogeneous Neg Bin:

```
Generalized negative binomial regression        Number of obs     =        225
                                                LR chi2(5)        =      22.98
                                                Prob > chi2       =     0.0003
Log likelihood = -406.66217                     Pseudo R2         =     0.0275
------------------------------------------------------------------------------
        aggr |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
aggr         |
         IPV |   .1148393    .295082     0.39   0.697    -.4635109    .6931894
         NvC |  -1.077993   .3395887    -3.17   0.002    -1.743575   -.4124115
         NvS |  -.1443183   .2765458    -0.52   0.602    -.6863381    .3977014
 c.IPV#c.NvC |  -1.933485   .7057295    -2.74   0.006    -3.316689   -.5502803
 c.IPV#c.NvS |   .5247327   .4394642     1.19   0.232    -.3366012    1.386067
       _cons |   .9995214    .193111     5.18   0.000     .6210309    1.378012
-------------+----------------------------------------------------------------
lnalpha      |
         IPV |  -.8075255   .5892574    -1.37   0.171    -1.962449    .3473977
         NvC |   .6411665   .4645302     1.38   0.168    -.2692959    1.551629
         NvS |   .0499735   .3927454     0.13   0.899    -.7197934    .8197403
 c.IPV#c.NvC |  -12.20546   23.22818    -0.53   0.599    -57.73185    33.32094 → Uh oh…
 c.IPV#c.NvS |   .6048024   .7515379     0.80   0.421    -.8681849     2.07779
       _cons |   .4752475   .2696956     1.76   0.078    -.0533463    1.003841
------------------------------------------------------------------------------
```

After re-estimating the model removing the interaction terms in predicting the scale factor, none of the effects predicting different scale factors by condition or IPV are significant and the information criteria are higher (worse). This indicates the original negative binomial with a constant scale factor is likely to be sufficient.

One other idea—given the lack of quantitative predictors to be given linear slopes (that could create predicted counts below zero without a link function) we could also estimate a general linear model in which the residual variance is allowed to differ across the six conditions. This would also address the problem of non-constant residual variance by tying it to the linear predictor rather than the predicted mean per se. This type of heterogeneous variance model could be done in SAS MIXED or GLIMMIX, as well as STATA MIXED and R LME. But given our focus on generalized linear models, I will leave that idea for another example…

Sample results section [notes what else should be included]:

We examined the extent to which how the count of aggressive verbalizations in the experimental condition differed across three strategy conditions (none, cognitive reappraisal, or suppression) as a function of whether participants had a history of intimate partner violence (IPV; no, yes). We estimated generalized linear models in [software] using maximum likelihood without denominator degrees of freedom. Effect sizes are provided using incident-rate ratios (IIR), which are exponentiated slope coefficients interpreted similarly to odds ratios: IRR values between 0 and 1 indicate negative effects, an IRR value of 1 indicates no effect, and IRR values > 1 indicate positive effects.

Before examining the results, we first examined the fit of the conditional distribution to the model residuals. As expected given the highly skewed observed count distribution, a model specifying an identity link function and normal residuals (i.e., a standard analysis of variance) resulted in confidence intervals for the cell means that included negative (impossible) count values. An alternative model specifying a log link function and Poisson conditional distribution (in which the conditional mean and variance are the same) did not appear to fit the observed distribution (Pearson $\chi^2/DF = 4.70$). This is because the conditional variance significantly exceeded the conditional mean, as indicated by a significant likelihood ratio test for a model specifying a negative binomial distribution instead (i.e., that included a scale factor to allow over-dispersion as a quadratic function of the mean, NB2), $-2\Delta LL(1) = 336.19$, $p < .0001$. Adding a zero-inflation parameter did not improve model fit, $-2\Delta LL(1) = 0$, indicating that the observed 0 values were adequately captured within the negative binomial distribution (Pearson $\chi^2/DF = 1.12$). Finally, we examined the potential for group differences in the log of the dispersion scale factor using the same linear predictor as for the log count, but no main effects or interactions were significant, suggesting the original negative binomial with a single scale factor is likely to be sufficient.

The overall model explained a significant amount of variance in aggressive verbalizations, $\chi^2(5) = 41.22$, $p < .0001$. The correlation between the predicted and actual counts was .336 ($R^2 = .113$); dispersion parameter = 1.601. As expected, there was a significant interaction between strategy condition and history of IPV, $\chi^2(2) = 10.47$, $p = .005$. [Figure 1 depicts the adjusted cell means for the log counts in panel A, and the expected counts in panel B. Table 1 provides simple slopes and slope differences within the interaction].

Let us first consider the pattern of the interaction with IPV as a moderator of the effect of strategy condition. The number of aggressive verbalizations was significantly lower when using a cognitive reappraisal strategy than when using no strategy (control) or a suppression strategy, and this was true for persons with or without a history of IPV. However, these benefits of a cognitive reappraisal strategy (relative to control or suppression) were significantly stronger in persons with a history of IPV than persons without a history of IPV.

Let us next consider the pattern of the interaction with strategy condition as a moderator of the effect of IPV. There were no significant IPV group differences when using no strategy (although aggressive verbalizations were marginally higher in persons with a history of IPV than without when using a suppression strategy). Surprisingly, the number of aggressive verbalizations when using a cognitive reappraisal strategy was significantly lower in persons with a history of IPV than without a history of IPV. This IPV group difference was significantly larger for participants using a cognitive reappraisal strategy than those using no strategy, and the IPV effect differed significantly between the cognitive reappraisal and suppression strategy conditions.