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. Download the paper here

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: Poisson, negative binomial, zero-inflated Poisson, and zero-inflated negative binomial. The probability of being an extra zero is predicted with a logit link in the two zero-inflated variants. In SAS, I am still using GLIMMIX (even though these are not mixed-effects models), as well as GENMOD for the zero-inflated model variants. Further, because the relevant STATA options (using GLM to get conditional distribution fit, also using 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 function GLM, the glm.nb function from package MASS, and the zeroinfl function from package pscl (each also using z or χ^2). There are some inconsistencies in the results from R (as usual) where noted.

SAS Syntax for Data Import and Manipulation:

```
* Location for original files for these models - change this path;
* \\Client\ precedes path in Virtual Desktop outside H drive;
%LET filesave=C:\Dropbox\22_PSQF6270\PSQF6270_Example3;
LIBNAME filesave "&filesave.";

* Import Example 3 SAS data into work library;
DATA work.Example3; SET filesave.PSQF6270_Example3;
* Create indicator-dummy-coded predictor variables;
  NvC=.; NvS=.;  * Make 2 new empty variables;
  IF ercond=1 THEN DO; NvC=0; NvS=0; END; * Recode each if ercond=1=None;
  IF ercond=2 THEN DO; NvC=1; NvS=0; END; * Recode each if ercond=2=Cognitive Reappraisal;
  IF ercond=3 THEN DO; NvC=0; NvS=1; END; * Recode each if ercond=3=Suppression;
* Label all variables used in analysis;
  LABEL ipv=    "ipv: Inter-Partner Violence (0=N,1=Y)"
        ercond= "ercond: 1=None, 2=CogR, 3=Supp"
        aggr=   "aggr: Aggressive Verbalizations"
        NvC=    "NvC: Condition None=0 vs CogR=1"
        NvS=    "NvS: Condition None=0 vs Supp=1";
* Filter to only cases complete on all variables to be used below;
  IF NMISS(aggr, ipv, ercond)>0 THEN DELETE;
RUN;
```

STATA Syntax for Data Import and Manipulation:

```
// Defining global variable for file location to be replaced in code below
// \\Client\ precedes path in Virtual Desktop outside H drive;
global filesave "\\Client\C:\Dropbox\22_PSQF6270\PSQF6270_Example3"

// Import Example 3 Stata data
use "$filesave\PSQF6270_Example3.dta", 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 each if ercond=1=None
replace NvS=0 if ercond==1
replace NvC=1 if ercond==2  // Replace each if ercond=2=CogR
replace NvS=0 if ercond==2
replace NvC=0 if ercond==3  // Replace each 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 Data Import and Manipulation:

```
# Define variables for working directory and data name
filesave = "C:\\Dropbox/22_PSQF6270/PSQF6270_Example3/"
filename = "PSQF6270_Example3.sas7bdat"
setwd(dir=filesave)

# Import Example 3 SAS data
Example3 = read_sas(data_file=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 each for lower
Example3$NvS[which(Example3$ercond==1)]=0
Example3$NvC[which(Example3$ercond==2)]=1  # Replace each for middle
Example3$NvS[which(Example3$ercond==2)]=0
Example3$NvC[which(Example3$ercond==3)]=0  # Replace each for upper
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[ ,6:10]),]
```

Syntax and SAS Output for Data Description:

```
TITLE "SAS Cell Means for Aggressive Verbalizations Outcome";
PROC MEANS NONOBS NDEC=2 N MAX STDDEV MEAN STDERR DATA=work.Example3;
     CLASS ercond ipv; WAYS 2; VAR aggr;
RUN; TITLE;

TITLE "SAS Histogram for Aggressive Verbalizations Outcome";
PROC UNIVARIATE NOPRINT DATA=work.Example3; VAR aggr;
     HISTOGRAM aggr / MIDPOINTS=0 TO 24 BY 1;
RUN; QUIT; TITLE;

TITLE "SAS Histogram for Aggressive Verbalizations Outcome by Cell";
PROC SGPANEL DATA=work.Example3;
     PANELBY ipv ercond / ROWS=2 COLUMNS=3;
     HISTOGRAM aggr / BINSTART=0 BINWIDTH=1;
     LABEL ipv= "ipv" ercond="1=None, 2=CogR, 3=Supp";
RUN; QUIT; TITLE;
```

```
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 didn't figure out how to make a separate histogram for each cell
```

Analysis Variable : aggr aggr: Aggressive Verbalizations

ercond: 1=None, 2=CogR, 3=Supp	ipv: Inter-Partner Violence (0=N,1=Y)	N	Maximum	Std Dev	Mean	Std Error
1	0	53	24.00	4.32	**2.72**	0.59
	1	21	9.00	2.82	**3.05**	0.62
2	0	53	11.00	1.95	**0.92**	0.27
	1	20	1.00	0.37	**0.15**	0.08
3	0	54	14.00	3.30	**2.35**	0.45
	1	24	19.00	5.27	**4.46**	1.08

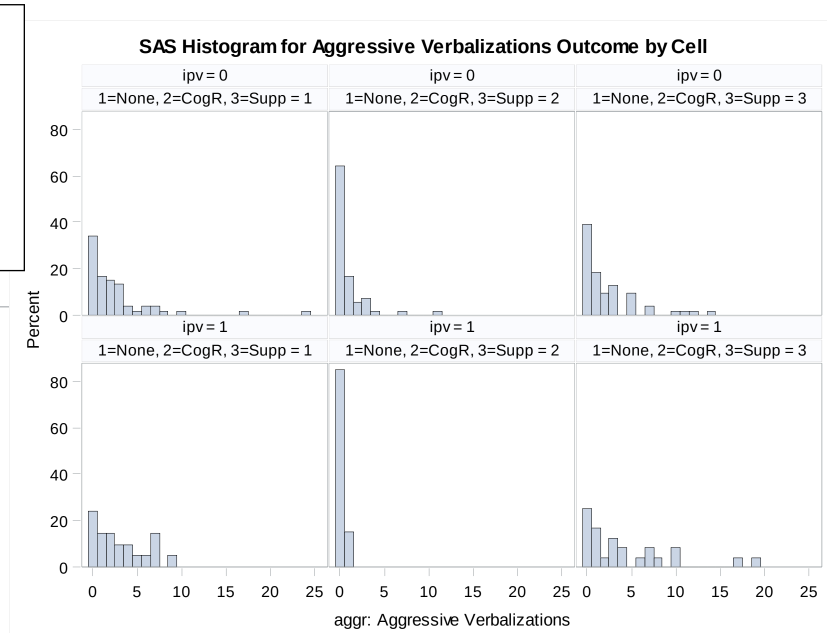
These are the outcome **cell means** (for each combination of IPV by condition) that our model is trying to capture….

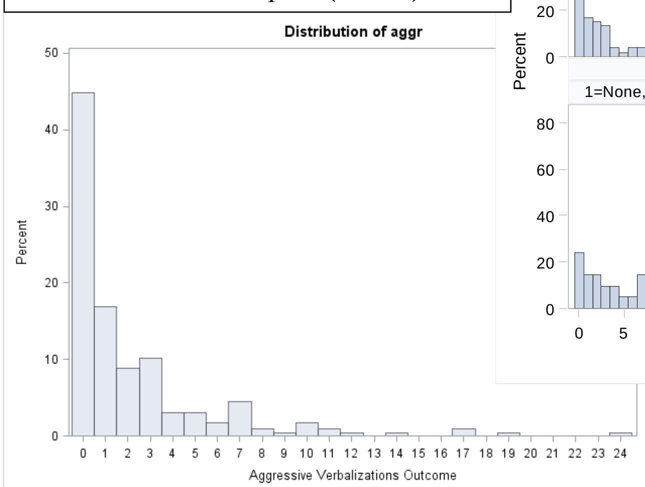
What we will see in this example is that the **cell means will stay the same** across models (via the same linear predictor).

What will change are the inferences about their 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 so good if we expect to use a general linear model (i.e., ANOVA here) to predict this outcome…

However, below is only the marginal distribution of Y. Maybe the residuals will look more normal?? Nope… (see 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 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:

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

Model for Aggressive Verbalizations using an Identity Link and a Normal Conditional Distribution ***(otherwise known as ANOVA, usually estimated with least squares which is equivalent to REML, here using ML instead for comparability with the count models to follow)***

```
TITLE1 "SAS Link=Identity Dist=Normal Model using GLIMMIX";
TITLE2 "Using ML instead of REML=OLS for Comparability with Subsequent Models";
PROC GLIMMIX DATA=work.Example3 GRADIENT NAMELEN=100 METHOD=MSPL PLOTS=(PEARSONPANEL);
MODEL aggr = ipv NvC NvS ipv*NvC ipv*NvS
              / SOLUTION CHISQ DDFM=NONE LINK=IDENTITY DIST=NORMAL;
CONTRAST "DF=5 Multiv Wald Test of Model" ipv 1, NvC 1, NvS 1, ipv*NvC 1, ipv*NvS 1 /CHISQ;
CONTRAST "DF=2 Multiv Wald Test of Interaction" ipv*NvC 1, ipv*NvS 1 / CHISQ;
* Yhat cell means in original count model scale per combo of ipv and condition;
ESTIMATE "IPV=No,  Cond=None"  int 1 ipv 0 NvC 0 NvS 0 ipv*NvC 0 ipv*Nvs 0 / CL;
ESTIMATE "IPV=Yes, Cond=None"  int 1 ipv 1 NvC 0 NvS 0 ipv*NvC 0 ipv*Nvs 0 / CL;
ESTIMATE "IPV=No,  Cond=CogR"  int 1 ipv 0 NvC 1 NvS 0 ipv*NvC 0 ipv*Nvs 0 / CL;
ESTIMATE "IPV=Yes, Cond=CogR"  int 1 ipv 1 NvC 1 NvS 0 ipv*NvC 1 ipv*Nvs 0 / CL;
ESTIMATE "IPV=No,  Cond=Supp"  int 1 ipv 0 NvC 0 NvS 1 ipv*NvC 0 ipv*Nvs 0 / CL;
ESTIMATE "IPV=Yes, Cond=Supp"  int 1 ipv 1 NvC 0 NvS 1 ipv*NvC 0 ipv*Nvs 1 / CL;
* Simple slopes of IPV per condition;
ESTIMATE "No vs Yes IPV: None"    ipv 1 ipv*NvC 0 ipv*Nvs 0;
ESTIMATE "No vs Yes IPV: CogR"    ipv 1 ipv*NvC 1 ipv*Nvs 0;
ESTIMATE "No vs Yes IPV: Supp"    ipv 1 ipv*NvC 0 ipv*Nvs 1;
* Simple slopes of condition per IPV;
ESTIMATE "None vs CogR: IPV=No"  NvC  1 NvS 0 ipv*NvC  0 ipv*Nvs 0;
ESTIMATE "None vs Supp: IPV=No"  NvC  0 NvS 1 ipv*NvC  0 ipv*Nvs 0;
ESTIMATE "CogR vs Supp: IPV=No"  NvC -1 NvS 1 ipv*NvC  0 ipv*Nvs 0;
ESTIMATE "None vs CogR: IPV=Yes" NvC  1 NvS 0 ipv*NvC  1 ipv*Nvs 0;
ESTIMATE "None vs Supp: IPV=Yes" NvC  0 NvS 1 ipv*NvC  0 ipv*Nvs 1;
ESTIMATE "CogR vs Supp: IPV=Yes" NvC -1 NvS 1 ipv*NvC -1 ipv*Nvs 1;
* Simple slopes for interaction contrasts;
ESTIMATE "No/Yes IPV differ btw None/Cog"  ipv*NvC  1 ipv*Nvs 0;
ESTIMATE "No/Yes IPV differ btw None/Sup"  ipv*NvC  0 ipv*Nvs 1;
ESTIMATE "No/Yes IPV differ btw Cog/Sup"   ipv*NvC -1 ipv*Nvs 1;
RUN; TITLE1; TITLE2;
```

```
display "STATA Link=Identity Dist=Normal Model using glm"
display "Using ML instead of REML=OLS for Comparability with Subsequent Models"
display "Add vfactor=(N-#fixed effects)/N to match SAS results for SEs"
display "Remove vfactor to match R GLM results for SEs"
glm aggr c.ipv c.NvC c.NvS c.ipv#c.NvC c.ipv#c.NvS, ///
         ml link(identity) family(gaussian) vfactor(0.973333333)
display "-2LL=" e(ll)*-2  // Print -2LL for model
estat ic, n(225)  // AIC and BIC do not match SAS and I do not know why
// DF=5 Multiv Wald Test of Model matches SAS
test (c.ipv=0)(c.NvC=0)(c.NvS=0)(c.ipv#c.NvC=0)(c.ipv#c.NvS=0)
```

```
// DF=2 Multiv Wald Test of Interaction matches SAS
test (c.ipv#c.NvC=0)(c.ipv#c.NvS=0)
// 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
```

From R GLM, the results are internally *inconsistent*—the −2LL matches that of ML, but the SEs and test statistics match the REML solution instead (that corrects for # fixed effects estimated)

```
print("R Link=Identity Dist=Normal Model using glm for ML Estimation")
print("SEs do not match SAS, and using t instead of z")
ModelNorm = glm(data=Example3, family=gaussian(link="identity"),
                formula=aggr~1+ipv+NvC+NvS +ipv:NvC +ipv:NvS); summary(ModelNorm)
print("Print ML -2LL, AIC, and BIC that match SAS")
-2*logLik(ModelNorm); AIC(ModelNorm); BIC(ModelNorm)

print("DF=5 Multiv Wald Test of Model -- matches SAS using REML")
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

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

print("Yhat cell means in original count model scale per condition")
print("SEs match SAS using REML instead")
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))

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),   # in order of fixed effects
  "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")))
NormSlopes
```

SAS Output for Normal:

Fit Statistics

-2 Log Likelihood	1184.26
AIC (smaller is better)	1198.26
AICC (smaller is better)	1198.77
BIC (smaller is better)	1222.17
CAIC (smaller is better)	1229.17
HQIC (smaller is better)	1207.91
Pearson Chi-Square	2544.23
Pearson Chi-Square / DF	**11.31**

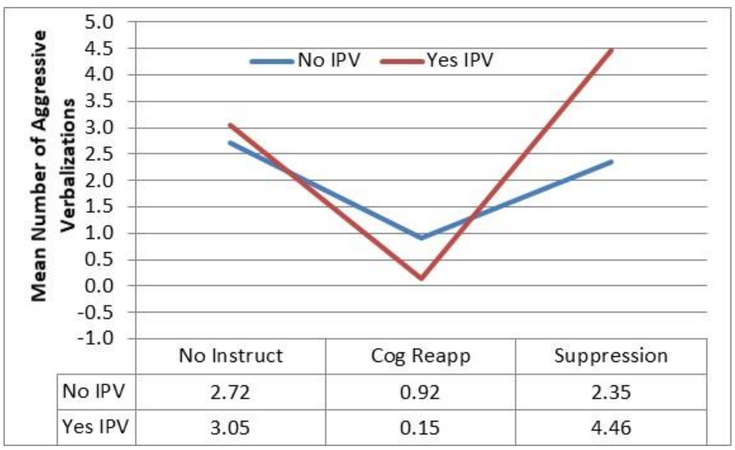
This is what I screwed up the last time I taught this class—for a normal conditional distribution, there is no denominator used for the expected SD by which to standardize the squared Pearson residual for each observation.
So the χ^2/DF value is just the residual variance, not an index of misfit!!!!

Btw, below is why variance differs across programs… ML *N* vs. REML *DDF*:

N	k	DDF = N-k	vfactor= DDF/N	Pearson chisq or residual deviance	chi-sq/DDF = scale factor using N	chi-sq/DDF = scale factor using DDF
225	6	219	0.973333	2544.228359	11.3076816	11.61748109

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Gradient	
Intercept	2.7170	0.4619	Infty	5.88	<.0001	-135E-17	**Beta0**
ipv	0.3306	0.8671	Infty	0.38	0.7030	8.88E-16	**Beta1**
NvC	-1.7925	0.6532	Infty	-2.74	0.0061	3.29E-15	**Beta2**
NvS	-0.3651	0.6502	Infty	-0.56	0.5744	-278E-19	**Beta3**
ipv*NvC	-1.1052	1.2372	Infty	-0.89	0.3717	6.52E-16	**Beta4**
ipv*NvS	1.7758	1.1968	Infty	1.48	0.1379	1.44E-15	**Beta5**
Scale	11.3077	1.0661	.	.	.	4.86E-16	**ML residual Variance**



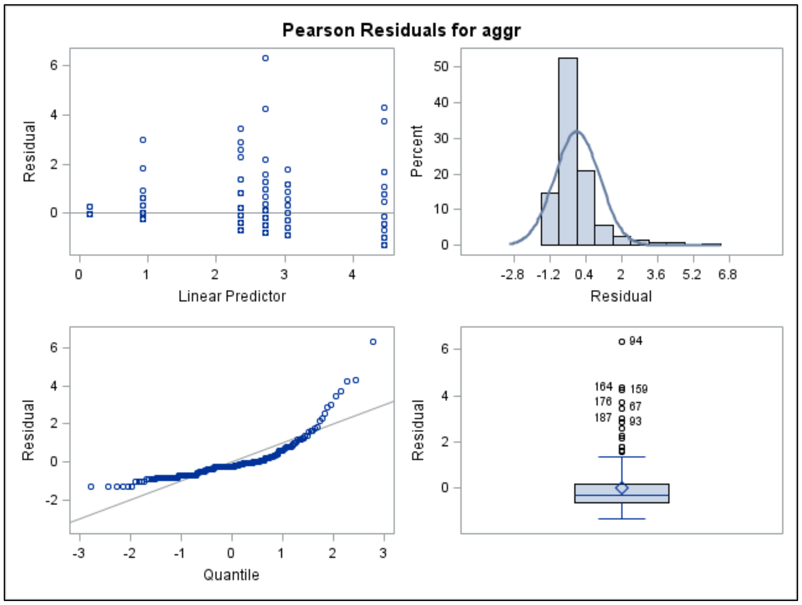
Here are the outcome cell means plotted in excel to help us understand the simple effects below…

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Lower	Upper	
IPV=No, Cond=None	2.7170	0.4619	Infty	5.88	<.0001	1.8117	3.6223	
IPV=Yes, Cond=None	3.0476	0.7338	Infty	4.15	<.0001	1.6094	4.4858	
IPV=No, Cond=CogR	0.9245	0.4619	Infty	2.00	0.0453	0.01922	1.8298	
IPV=Yes, Cond=CogR	0.1500	0.7519	Infty	0.20	0.8419	**-1.3237**	1.6237	**Uh oh**
IPV=No, Cond=Supp	2.3519	0.4576	Infty	5.14	<.0001	1.4550	3.2487	
IPV=Yes, Cond=Supp	4.4583	0.6864	Infty	6.50	<.0001	3.1130	5.8037	
No vs Yes IPV: None	0.3306	0.8671	Infty	0.38	0.7030	-1.3688	2.0301	
No vs Yes IPV: CogR	-0.7745	0.8825	Infty	-0.88	0.3801	-2.5041	0.9551	
No vs Yes IPV: Supp	2.1065	0.8250	Infty	2.55	0.0107	0.4896	3.7234	
None vs CogR: IPV=No	-1.7925	0.6532	Infty	-2.74	0.0061	-3.0728	-0.5122	
None vs Supp: IPV=No	-0.3651	0.6502	Infty	-0.56	0.5744	-1.6395	0.9092	
CogR vs Supp: IPV=No	1.4273	0.6502	Infty	2.20	0.0281	0.1530	2.7017	
None vs CogR: IPV=Yes	-2.8976	1.0506	Infty	-2.76	0.0058	-4.9568	-0.8384	
None vs Supp: IPV=Yes	1.4107	1.0048	Infty	1.40	0.1603	-0.5586	3.3801	
CogR vs Supp: IPV=Yes	4.3083	1.0181	Infty	4.23	<.0001	2.3129	6.3038	
No/Yes IPV differ btw None/Cog	-1.1052	1.2372	Infty	-0.89	0.3717	-3.5299	1.3196	
No/Yes IPV differ btw None/Sup	1.7758	1.1968	Infty	1.48	0.1379	-0.5699	4.1216	
No/Yes IPV differ btw Cog/Sup	2.8810	1.2080	Infty	2.38	0.0171	0.5134	5.2487	

p = .0535, seriously???

```
                                        Contrasts
                                       Num      Den
Label                                   DF       DF   Chi-Square   F Value      Pr > ChiSq   Pr > F
DF=5 Multiv Wald Test of Model           5    Infty        28.58      5.72          <.0001   <.0001
DF=2 Multiv Wald Test of Interaction     2    Infty         5.86      2.93          0.0535   0.0535
```



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).

Partial STATA Output for Normal:

```
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 residual 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
------------------------------------------------------------------------------
```

Partial R Output for Normal:

```
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)

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

These deviance values do not refer to LL or −2LL (as in previous examples). So I added separate commands to get −2LL, AIC, and BIC for each model.

Model for Aggressive Verbalizations using a Log Link and a Poisson Conditional Distribution

```
TITLE "SAS Link=Log Dist=Poisson Model using GLIMMIX";
PROC GLIMMIX DATA=work.Example3 GRADIENT NAMELEN=100 METHOD=MSPL;
MODEL aggr = ipv NvC NvS ipv*NvC ipv*NvS
             / SOLUTION CHISQ DDFM=NONE LINK=LOG DIST=POISSON;
CONTRAST "DF=5 Multiv Wald Test of Model" ipv 1, NvC 1, NvS 1, ipv*NvC 1, ipv*NvS 1 /CHISQ;
CONTRAST "DF=2 Multiv Wald Test of Interaction" ipv*NvC 1, ipv*NvS 1 / CHISQ;
* Yhat cell means in model and data scale per combination of ipv and condition;
ESTIMATE "IPV=No,  Cond=None"  int 1 ipv 0 NvC 0 NvS 0 ipv*NvC 0 ipv*Nvs 0 / ILINK;
ESTIMATE "IPV=Yes, Cond=None"  int 1 ipv 1 NvC 0 NvS 0 ipv*NvC 0 ipv*Nvs 0 / ILINK;
ESTIMATE "IPV=No,  Cond=CogR"  int 1 ipv 0 NvC 1 NvS 0 ipv*NvC 0 ipv*Nvs 0 / ILINK;
ESTIMATE "IPV=Yes, Cond=CogR"  int 1 ipv 1 NvC 1 NvS 0 ipv*NvC 1 ipv*Nvs 0 / ILINK;
ESTIMATE "IPV=No,  Cond=Supp"  int 1 ipv 0 NvC 0 NvS 1 ipv*NvC 0 ipv*Nvs 0 / ILINK;
ESTIMATE "IPV=Yes, Cond=Supp"  int 1 ipv 1 NvC 0 NvS 1 ipv*NvC 0 ipv*Nvs 1 / ILINK;
* Simple slopes of IPV per condition: EXP --> IRR effect sizes;
ESTIMATE "No vs Yes IPV: None"    ipv 1 ipv*NvC 0 ipv*Nvs 0 / EXP;
ESTIMATE "No vs Yes IPV: CogR"    ipv 1 ipv*NvC 1 ipv*Nvs 0 / EXP;
ESTIMATE "No vs Yes IPV: Supp"    ipv 1 ipv*NvC 0 ipv*Nvs 1 / EXP;
* Simple slopes of condition per IPV: EXP --> IRR effect sizes;
ESTIMATE "None vs CogR: IPV=No"  NvC  1 NvS 0 ipv*NvC  0 ipv*Nvs 0 / EXP;
ESTIMATE "None vs Supp: IPV=No"  NvC  0 NvS 1 ipv*NvC  0 ipv*Nvs 0 / EXP;
ESTIMATE "CogR vs Supp: IPV=No"  NvC -1 NvS 1 ipv*NvC  0 ipv*Nvs 0 / EXP;
ESTIMATE "None vs CogR: IPV=Yes" NvC  1 NvS 0 ipv*NvC  1 ipv*Nvs 0 / EXP;
ESTIMATE "None vs Supp: IPV=Yes" NvC  0 NvS 1 ipv*NvC  0 ipv*Nvs 1 / EXP;
ESTIMATE "CogR vs Supp: IPV=Yes" NvC -1 NvS 1 ipv*NvC -1 ipv*Nvs 1 / EXP;
* Simple slopes for interaction contrasts: EXP --> IRR effect sizes;
ESTIMATE "No/Yes IPV differ btw None/CogR"  ipv*NvC  1 ipv*Nvs 0 / EXP;
ESTIMATE "No/Yes IPV differ btw None/Supp"  ipv*NvC  0 ipv*Nvs 1 / EXP;
ESTIMATE "No/Yes IPV differ btw CogR/Supp"  ipv*NvC -1 ipv*Nvs 1 / EXP;
* Save tables with #parms and -2LL for FitTestG macro;
ODS OUTPUT OptInfo=ParmsP FitStatistics=FitP;
RUN; TITLE;
```

Add CL after slash to get confidence intervals too (used in online output)

```
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)
display "-2LL=" e(ll)*-2  // Print -2LL for model
estat ic, n(225)   // AIC and BIC match SAS
// 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)
// DF=2 Multiv Wald Test of Interaction
test  (c.ipv#c.NvC=0)(c.ipv#c.NvS=0)
// 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
// 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
// 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
```

We are using STATA GLM (as opposed to POISSON) see how well our residuals fit a Poisson conditional distribution.

Btw, to get the same fit as given by SAS (dividing by *N*), use this:

```
display
"Pearson chi2/DF using N= "
e(deviance_p)/e(N)
```

```
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)
// Simple slopes of IPV per condition
lincom c.ipv*1 + c.ipv#c.NvC*0 + c.ipv#c.NvS*0, irr  // No vs Yes IPV: None
lincom c.ipv*1 + c.ipv#c.NvC*1 + c.ipv#c.NvS*0, irr  // No vs Yes IPV: CogR
lincom c.ipv*1 + c.ipv#c.NvC*0 + c.ipv#c.NvS*1, irr  // 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, irr  // None vs CogR: IPV=No
lincom c.NvC*0  + c.NvS*1 + c.ipv#c.NvC*0  + c.ipv#c.NvS*0, irr  // None vs Supp: IPV=No
lincom c.NvC*-1 + c.NvS*1 + c.ipv#c.NvC*0  + c.ipv#c.NvS*0, irr  // CogR vs Supp: IPV=No
lincom c.NvC*1  + c.NvS*0 + c.ipv#c.NvC*1  + c.ipv#c.NvS*0, irr  // None vs CogR: IPV=Yes
lincom c.NvC*0  + c.NvS*1 + c.ipv#c.NvC*0  + c.ipv#c.NvS*1, irr  // None vs Supp: IPV=Yes
lincom c.NvC*-1 + c.NvS*1 + c.ipv#c.NvC*-1 + c.ipv#c.NvS*1, irr  // CogR vs Supp: IPV=Yes
// Simple slopes for interaction contrasts
lincom c.ipv#c.NvC*1  + c.ipv#c.NvS*0, irr  // No/Yes IPV differ btw None/CogR
lincom c.ipv#c.NvC*0  + c.ipv#c.NvS*1, irr  // No/Yes IPV differ btw None/Supp
lincom c.ipv#c.NvC*-1 + c.ipv#c.NvS*1, irr  // No/Yes IPV differ btw CogR/Supp


# R: save sample size and DDF for conditional distribution fit
DDFn=225     # What SAS uses
DDFk=DDFn-6  # What STATA uses

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)
summary(ModelPoisson); print("Print -2LL, AIC, and BIC")
-2*logLik(ModelPoisson); AIC(ModelPoisson); BIC(ModelPoisson)
print("Pearson Chi-Square / DF Index of Fit matching SAS and STATA")
sum(residuals(ModelPoisson, type="pearson")^2)/DDFn   # SAS
sum(residuals(ModelPoisson, type="pearson")^2)/DDFk   # STATA

print("DF=5 Multiv Wald Test of Model -- matches SAS and STATA")
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

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

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))

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))

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),   # in order of fixed effects
  "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),
```

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

```
  "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))
```

SAS Output for Log Link and Poisson:

```
       Fit Statistics (abbreviated)
-2 Log Likelihood            1155.76
AIC  (smaller is better)     1167.76
BIC  (smaller is better)     1188.26
Pearson Chi-Square           1028.54
Pearson Chi-Square / DF         4.57 → not low enough (1=good)
```

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Gradient	
Intercept	0.9995	0.08333	Infty	11.99	<.0001	6.74E-7	**Beta0**
ipv	0.1148	0.1502	Infty	0.76	0.4446	3.258E-7	**Beta1**
NvC	-1.0780	0.1654	Infty	-6.52	<.0001	4.652E-7	**Beta2**
NvS	-0.1443	0.1217	Infty	-1.19	0.2358	4.56E-8	**Beta3**
ipv*NvC	-1.9335	0.6134	Infty	-3.15	0.0016	3.187E-7	**Beta4**
ipv*NvS	0.5247	0.1995	Infty	2.63	0.0085	7.17E-9	**Beta5**

Now that's more like it! ☺

Label	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
DF=5 Multiv Wald Test of Model	5	Infty	111.08	22.22	<.0001	<.0001
DF=2 Multiv Wald Test of Interaction	2	Infty	20.61	**10.31**	**<.0001**	**<.0001**

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Mean	Standard Error Mean	Exponentiated Estimate
IPV=No, Cond=None	0.9995	0.08333	Infty	11.99	<.0001	2.7170	0.2264	.
IPV=Yes, Cond=None	1.1144	0.1250	Infty	8.91	<.0001	3.0476	0.3810	.
IPV=No, Cond=CogR	-0.07847	0.1429	Infty	-0.55	0.5828	0.9245	0.1321	.
IPV=Yes, Cond=CogR	-1.8971	0.5774	Infty	-3.29	0.0010	0.1500	0.08660	.
IPV=No, Cond=Supp	0.8552	0.08874	Infty	9.64	<.0001	2.3519	0.2087	.
IPV=Yes, Cond=Supp	1.4948	0.09667	Infty	15.46	<.0001	4.4583	0.4310	.
No vs Yes IPV: None	0.1148	0.1502	Infty	0.76	0.4446	Non-est	.	1.1217
No vs Yes IPV: CogR	-1.8186	0.5948	Infty	-3.06	0.0022	Non-est	.	0.1622
No vs Yes IPV: Supp	0.6396	0.1312	Infty	4.87	<.0001	Non-est	.	1.8957
None vs CogR: IPV=No	-1.0780	0.1654	Infty	-6.52	<.0001	Non-est	.	0.3403
None vs Supp: IPV=No	-0.1443	0.1217	Infty	-1.19	0.2358	Non-est	.	0.8656
CogR vs Supp: IPV=No	0.9337	0.1682	Infty	5.55	<.0001	Non-est	.	2.5438
None vs CogR: IPV=Yes	-3.0115	0.5907	Infty	-5.10	<.0001	Non-est	.	0.04922
None vs Supp: IPV=Yes	0.3804	0.1580	Infty	2.41	0.0161	Non-est	.	1.4629
CogR vs Supp: IPV=Yes	3.3919	0.5854	Infty	5.79	<.0001	Non-est	.	29.7222
No/Yes IPV differ btw None/CogR	-1.9335	0.6134	Infty	-3.15	0.0016	Non-est	.	0.1446
No/Yes IPV differ btw None/Supp	0.5247	0.1995	Infty	2.63	0.0085	Non-est	.	1.6900
No/Yes IPV differ btw CogR/Supp	2.4582	0.6091	Infty	4.04	<.0001	Non-est	.	11.6840

Partial STATA Output for Log Link and Poisson:

```
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
------------------------------------------------------------------------------
```

Partial R Output for Log Link and Poisson:

```
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)

[1] "Print -2LL, AIC, and BIC"
> -2 * logLik(ModelPoisson)
'log Lik.' 1155.7639 (df=6)
> AIC(ModelPoisson)
[1] 1167.7639
> BIC(ModelPoisson)
[1] 1188.2605
[1] "Pearson Chi-Square / DF Index of Fit matching SAS and STATA"
> sum(residuals(ModelPoisson, type = "pearson")^2)/DDFn
[1] 4.5712873
> sum(residuals(ModelPoisson, type = "pearson")^2)/DDFk
[1] 4.696528
```

The Poisson distribution has only one parameter—the mean, which is supposed to also be the conditional variance. 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 allow the variance to change as a quadratic function of the mean (called “NB2”), which seems to be the most common approach.

Model Predicting Aggressive Verbalizations using a Log Link and a Negative Binomial Conditional Distribution

```
TITLE "SAS Link=Log Dist=Negative Binomial Model using GLIMMIX";
PROC GLIMMIX DATA=work.Example3 GRADIENT NAMELEN=100 METHOD=MSPL;
MODEL aggr = ipv NvC NvS ipv*NvC ipv*NvS
            / SOLUTION CHISQ DDFM=NONE LINK=LOG DIST=NEGBIN;
* Syntax omitted for Wald tests, cell means, and simple effects (same as for Poisson);
* Save tables with #parms and -2LL for FitTestG macro;
ODS OUTPUT OptInfo=ParmsNB FitStatistics=FitNB;
* Save predicted counts per real person to dataset;
OUTPUT OUT=work.NegBinPred PREDICTED(ILINK)=PredCount;
RUN; TITLE;

* Call FitTestG macro to compute LRT between Poisson and NegBin;
%FitTestG(FitFewer=FitP, ParmsFewer=ParmsP, FitMore=FitNB, ParmsMore=ParmsNB);

* Save corr of pred count with aggr to square as R2;
TITLE "Correlation of Predicted and Actual Count";
PROC CORR NOSIMPLE DATA=work.NegBinPred OUT=work.Rpred;
     VAR aggr; WITH PredCount; RUN;
* Compute R2 in saved output;
DATA work.Rpred; SET work.Rpred;
      WHERE _TYPE_="CORR"; R2=aggr*aggr; RUN;
```

```
TITLE "R2 of Predicted and Actual Count";
PROC PRINT NOOBS DATA=work.Rpred; VAR R2; RUN; TITLE;

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

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)
display "-2LL=" e(ll)*-2  // Print -2LL for model
estat ic, n(225)  // AIC and BIC do not match SAS
// Syntax omitted for Wald tests, cell means, and simple effects (same as for Poisson)
// 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
```

Btw, to get the same fit as given by SAS (dividing by *N*), use this:

```
display
"Pearson chi2/DF using N= "
e(deviance_p)/e(N)
```

```
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)
summary(ModelNegBin); print("Print -2LL, AIC, and BIC")
-2*logLik(ModelNegBin); AIC(ModelNegBin); BIC(ModelNegBin)
print("Scale factor in same metric as SAS and STATA")
1/ModelNegBin$theta
print("Pearson Chi-Square / DF Index of Fit from SAS and STATA")
sum(residuals(ModelNegBin, type="pearson")^2)/DDFn  # SAS
sum(residuals(ModelNegBin, type="pearson")^2)/DDFk  # STATA

print("Getting 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

# Syntax omitted for Wald tests, cell means, and simple effects (same 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
```

SAS Output for Neg Bin:

Fit Statistics (abbreviated)

-2 Log Likelihood	819.58
AIC (smaller is better)	833.58
BIC (smaller is better)	857.49
Pearson Chi-Square	244.63
Pearson Chi-Square / DF	**1.09**

Poisson model $-2LL = 1155.76$
$-2\Delta LL(df = 1) = 1155.76 - 819.58 = 336.19$, $p < .001$

So the model fits significantly better from adding a "dispersion" (scale) parameter that allows the variance to exceed the mean:
$Variance = \mu + (1.601\mu^2)$ (as shown in STATA GLM output)

And the 1.09 means the fit to the distribution is close to the data! 😊

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Gradient	
Intercept	0.9995	0.1927	Infty	5.19	<.0001	-1E-7	**Beta0**
ipv	0.1148	0.3592	Infty	0.32	0.7492	-4.09E-8	**Beta1**
NvC	-1.0780	0.2963	Infty	-3.64	0.0003	-684E-12	**Beta2**
NvS	-0.1443	0.2733	Infty	-0.53	0.5974	-5.58E-8	**Beta3**
ipv*NvC	-1.9335	0.7701	Infty	-2.51	0.0120	1.5E-11	**Beta4**
ipv*NvS	0.5247	0.4925	Infty	1.07	0.2867	-3.59E-8	**Beta5**
Scale	1.6010	0.2427	.	.	.	-6.3E-6	**k dispersion**

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Mean	Standard Error Mean	Exponentiated Estimate
IPV=No, Cond=None	0.9995	0.1927	Infty	5.19	<.0001	2.7170	0.5237	.
IPV=Yes, Cond=None	1.1144	0.3031	Infty	3.68	0.0002	3.0476	0.9237	.
IPV=No, Cond=CogR	-0.07847	0.2250	Infty	-0.35	0.7272	0.9245	0.2080	.
IPV=Yes, Cond=CogR	-1.8971	0.6429	Infty	-2.95	0.0032	0.1500	0.09644	.
IPV=No, Cond=Supp	0.8552	0.1937	Infty	4.41	<.0001	2.3519	0.4556	.
IPV=Yes, Cond=Supp	1.4948	0.2758	Infty	5.42	<.0001	4.4583	1.2295	.
No vs Yes IPV: None	0.1148	0.3592	Infty	0.32	0.7492	Non-est	.	1.1217
No vs Yes IPV: CogR	-1.8186	0.6812	Infty	-2.67	0.0076	Non-est	.	0.1622
No vs Yes IPV: Supp	0.6396	0.3370	Infty	1.90	0.0577	Non-est	.	1.8957
None vs CogR: IPV=No	-1.0780	0.2963	Infty	-3.64	0.0003	Non-est	.	0.3403
None vs Supp: IPV=No	-0.1443	0.2733	Infty	-0.53	0.5974	Non-est	.	0.8656
CogR vs Supp: IPV=No	0.9337	0.2969	Infty	3.14	0.0017	Non-est	.	2.5438
None vs CogR: IPV=Yes	-3.0115	0.7108	Infty	-4.24	<.0001	Non-est	.	0.04922
None vs Supp: IPV=Yes	0.3804	0.4098	Infty	0.93	0.3532	Non-est	.	1.4629
CogR vs Supp: IPV=Yes	3.3919	0.6996	Infty	4.85	<.0001	Non-est	.	29.7222
No/Yes IPV differ btw None/CogR	-1.9335	0.7701	Infty	-2.51	0.0120	Non-est	.	0.1446
No/Yes IPV differ btw None/Supp	0.5247	0.4925	Infty	1.07	0.2867	Non-est	.	1.6900
No/Yes IPV differ btw CogR/Supp	2.4582	0.7600	Infty	3.23	0.0012	Non-est	.	11.6840

Contrasts

Not as optimistic as Poisson…

Label	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
DF=5 Multiv Wald Test of Model	5	Infty	41.22	8.24	<.0001	<.0001
DF=2 Multiv Wald Test of Interaction	2	Infty	10.47	**5.24**	**0.0053**	**0.0053**

Likelihood Ratio Test for FitP vs. FitNB → this is from my %FitTestG macro

Regular p-value uses DF, mixture p-value uses DF=DF,DF-1

Dev Fewer	DevMore	DFfewer	DFmore	Test Stat	Test DF	Reg Pvalue	Mix Pvalue
1155.76	819.577	6	7	**336.187**	1	<.0001	<.0001 → NegBin wins!!

Pearson Correlation, N = 225 → correlation of actual and predicted counts as model effect size

	aggr
PredCount	.33571 → R-square=.1127
Mu	<.0001

Partial STATA Output from nbreg for Neg Bin:

```
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
```

Partial STATA Output from glm for Neg Bin:

```
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
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.
```

Partial R Output for Neg Bin:

```
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

[1] "Print -2LL, AIC, and BIC"
> -2 * logLik(ModelNegBin)
'log Lik.' 819.5767 (df=7)
> AIC(ModelNegBin)
[1] 833.5767
> BIC(ModelNegBin)
[1] 857.48941

[1] "Scale factor in same metric as SAS and STATA"
> 1/ModelNegBin$theta
[1] 1.6010081

[1] "Pearson Chi-Square / DF Index of Fit from SAS and STATA"
> sum(residuals(ModelNegBin, type = "pearson")^2)/DDFn
[1] 1.087243
> sum(residuals(ModelNegBin, type = "pearson")^2)/DDFk
[1] 1.1170305

[1] "Test Statistic, Regular and Mixture P-values for DF=1"
'log Lik.' 336.18718 (df=6)
> RegPvalue
'log Lik.' 4.3176629e-75 (df=6)
> MixPvalue
'log Lik.' 2.1588315e-75 (df=6)
```

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 Zero-Inflated Poisson and Zero-Inflated Negative Binomial 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.

```
TITLE1 "SAS Zero-Inflated Poisson Model using GENDMOD";
TITLE2 "Only intercept in zero-inflation model (predict logit of extra zero)";
PROC GENMOD DATA=work.Example3 NAMELEN=100;
MODEL aggr = ipv NvC NvS ipv*NvC ipv*NvS / LINK=LOG DIST=ZIP;
ZEROMODEL / LINK=LOGIT;
ESTIMATE "Logit of extra zero" @ZERO intercept 1 / EXP;
CONTRAST "DF=5 Multiv Wald Test of Model" ipv 1, NvC 1, NvS 1, ipv*NvC 1, ipv*NvS 1 / WALD;
CONTRAST "DF=2 Multiv Wald Test of Interaction" ipv*NvC 1, ipv*NvS 1 / WALD;
RUN; TITLE1; TITLE2;
```

```
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)
display "-2LL=" e(ll)*-2  // Print -2LL for model
estat ic, n(225)  // AIC and BIC match SAS
nlcom 1/(1+exp(-1* _b[inflate:_cons])) // Probability of extra 0
// 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)
// DF=2 Multiv Wald Test of Interaction
test (c.ipv#c.NvC=0)(c.ipv#c.NvS=0)
```

I used "coeflegend" to see how STATA referred to the zero-inflation submodel intercept

```
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)
summary(ModelZIP); print("Print -2LL, AIC, and BIC")
-2*logLik(ModelZIP); AIC(ModelZIP); BIC(ModelZIP)
print("Pearson Chi-Square / DF Fit from SAS GENMOD and STATA ZIP")
sum(residuals(ModelZIP, type="pearson")^2)/DDFn   # SAS
sum(residuals(ModelZIP, type="pearson")^2)/DDFk   # STATA
print("Get probability of being an extra 0")
ZIPprob=1/(1+exp(-1*ModelZIP$coefficients$zero)); ZIPprob
# glht apparently does not work with pscl package, so no Wald tests included here
```

SAS Output for Zero-Inflated Poisson:

```
              Criteria For Assessing Goodness Of Fit
Deviance                                979.1702 → -2LL for model
Scaled Deviance                         979.1702
Pearson Chi-Square            218       451.6551         2.0718 → worse than NB model
Scaled Pearson X2             218       451.6551         2.0718
Log Likelihood                           68.5623
Full Log Likelihood                    -489.5851
AIC (smaller is better)                 993.1702 → worse than for NB model
AICC (smaller is better)                993.6863
BIC (smaller is better)                1017.0829 → worse than for NB model
```

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	1.3996	0.0859	1.2313	1.5680	265.51	<.0001
ipv	1	-0.0232	0.1537	-0.3244	0.2779	0.02	0.8799
NvC	1	-0.7697	0.2118	-1.1849	-0.3545	13.20	0.0003
NvS	1	-0.0763	0.1265	-0.3242	0.1716	0.36	0.5463

```
ipv*NvC        1     -2.0584     0.6469     -3.3264     -0.7904        10.12        0.0015
ipv*NvS        1      0.4809     0.2041      0.0808      0.8810         5.55        0.0185
Scale          0      1.0000     0.0000      1.0000      1.0000 → because ZIP has no extra variance

             Analysis Of Maximum Likelihood Zero Inflation Parameter Estimates
                                Standard     Wald 95% Confidence         Wald
Parameter    DF    Estimate       Error           Limits           Chi-Square    Pr > ChiSq
Intercept     1     -0.5453      0.1649     -0.8685     -0.2221         10.93        0.0009
```

-0.5453 gives the logit of the probability of being an extra 0 if mean = variance in Poisson. Btw, this Wald-test chi-square is still testing it against a null hypothesis of zero, which is useless to us.

```
                                                     Contrast Estimate Results
                                               Mean        Mean          L'Beta  Standard
Label                                      Estimate  Confidence Limits  Estimate     Error
Extra zero intercept (Zero Inflation)        0.3670   0.2956    0.4447   -0.5453    0.1649 Prob, Logit
Exp(Extra zero intercept (Zero Inflation))                                0.5797    0.0956 Odds
```

Partial STATA Output for Zero-Inflated Poisson:

```
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)
------------------------------------------------------------------------------
```

Partial R Output for Zero-Inflated Poisson:

```
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)

[1] "Print -2LL, AIC, and BIC"
> -2 * logLik(ModelZIP)
'log Lik.' 979.17018 (df=7)
> AIC(ModelZIP)
[1] 993.17018
> BIC(ModelZIP)
[1] 1017.0829
[1] "Pearson Chi-Square / DF Fit from SAS GENMOD and STATA ZIP"
> sum(residuals(ModelZIP, type = "pearson")^2)/DDFn
[1] 2.0073558
> sum(residuals(ModelZIP, type = "pearson")^2)/DDFk
[1] 2.0623518
```

```
TITLE1 "SAS Zero-Inflated Negative Binomial Model using GENMOD";
TITLE2 "Only intercept in zero-inflation model (predicted in logits)";
PROC GENMOD DATA=work.Example3 NAMELEN=100;
MODEL aggr = ipv NvC NvS ipv*NvC ipv*NvS / LINK=LOG DIST=ZINB;
ZEROMODEL / LINK=LOGIT;
ESTIMATE "Extra zero intercept" @ZERO intercept 1 / EXP;
CONTRAST "DF=5 Multiv Wald Test of Model" ipv 1, NvC 1, NvS 1, ipv*NvC 1, ipv*NvS 1 / WALD;
CONTRAST "DF=2 Multiv Wald Test of Interaction" ipv*NvC 1, ipv*NvS 1 / WALD;
RUN; TITLE1; TITLE2;
```

Stata blew up! See code online for what it should have looked like.

```
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)
summary(ModelZINB); print("Print -2LL, AIC, and BIC")
-2*logLik(ModelZINB); AIC(ModelZINB); BIC(ModelZINB)
print("Pearson Chi-Square / DF Index of Fit from SAS GENMOD")
sum(residuals(ModelZINB, type="pearson")^2)/DDFk # SAS
print("Scale factor in same metric as SAS and STATA")
1/ModelZINB$theta
print("Get probability of being an extra 0")
ZINBprob=1/(1+exp(-1*ModelZINB$coefficients$zero)); ZINBprob
# glht apparently does not work with pscl package, so no Wald tests included here
```

SAS Output for Zero-Inflated Neg Bin:

Algorithm converged. → but wait for it...

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance		**819.5767 → -2LL value**	
Scaled Deviance		819.5767	
Pearson Chi-Square	218	244.6297	**1.1222 → Comparable to NB model**
Scaled Pearson X2	218	244.6297	1.1222
Log Likelihood		-409.7884	
Full Log Likelihood		-409.7884	
AIC (smaller is better)		**835.5767 → worse than NB**	
AICC (smaller is better)		836.2434	
BIC (smaller is better)		**862.9055 → worse than NB**	

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	0.9995	0.1927	0.6217	1.3773	26.89	<.0001
ipv	1	0.1148	0.3592	-0.5892	0.8188	0.10	0.7492
NvC	1	-1.0780	0.2963	-1.6586	-0.4973	13.24	0.0003
NvS	1	-0.1443	0.2733	-0.6799	0.3913	0.28	0.5974
ipv*NvC	1	-1.9335	0.7701	-3.4428	-0.4242	6.30	0.0120
ipv*NvS	1	0.5247	0.4925	-0.4406	1.4901	1.14	0.2867
Dispersion	1	**1.6010**	0.2427	1.1894	2.1550		

NOTE: The negative binomial dispersion parameter was estimated by maximum likelihood.

Analysis Of Maximum Likelihood Zero Inflation Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	**-21.8760**	50332.76	-98672.3	98628.52	0.00	0.9997

The logit of being an “extra 0” = −21.8760 (with a crazy SE)! This is probability =.000000000316 of being an “extra 0”. So there are no extra 0 values in this distribution not already predicted by the negative binomial.

Partial R Output for Zero-Inflated Neg Bin:

```
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)
Theta = 0.62462 → 1/k dispersion

[1] "Print -2LL, AIC, and BIC"
> -2 * logLik(ModelZINB)
'log Lik.' 819.57674 (df=8)
> AIC(ModelZINB)
[1] 835.57674
> BIC(ModelZINB)
[1] 862.90554
[1] "Pearson Chi-Square / DF Index of Fit from SAS GENMOD"
> sum(residuals(ModelZINB, type = "pearson")^2)/DDFk
[1] 1.1170425
[1] "Scale factor in same metric as SAS and STATA"
[1] 1.6009695
[1] "Get probability of being an extra 0"
0.000014994787
```

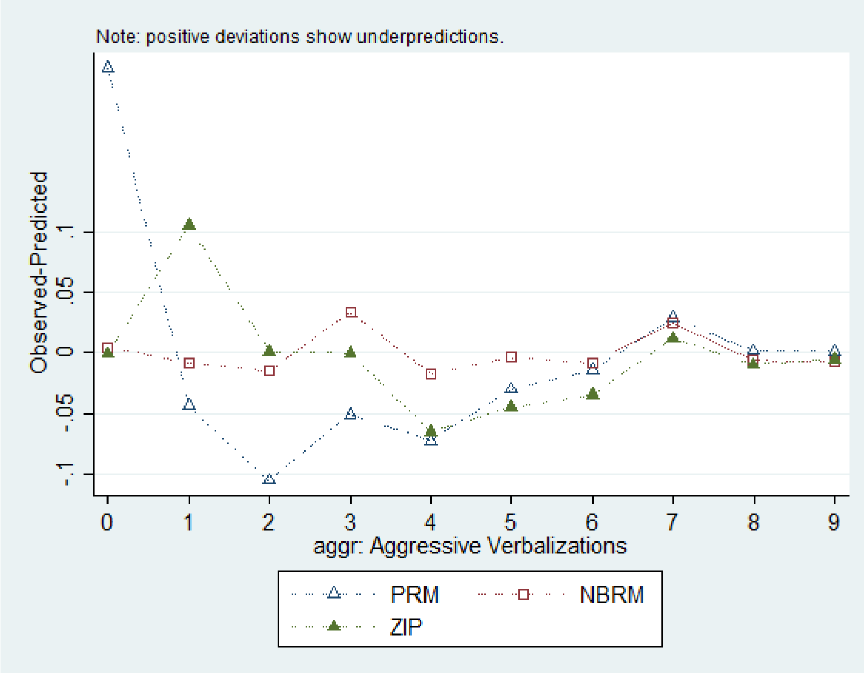
The STATA routine "countfit" can be used to compare count model conditional distributions:

```
// 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.

Below: Here is a comparison of the model-predicted cell means and their standard errors across models without zero-inflation. The means are exactly the same, but the SEs differ greatly.

Also below: Here is comparison of the simple slopes (for pairwise cell mean differences) and simple slope differences (for interactions). Poisson is way too optimistic with respect to the *p*-values because there is not enough variance included.

							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	0.46	2.72	0.23	2.72	0.52
IPV=Yes	**None**	21	9	2.82	3.05	0.62	3.05	0.73	3.05	0.38	3.05	0.92
IPV=No	**CogR**	53	11	1.95	0.92	0.27	0.92	0.46	0.92	0.13	0.92	0.21
IPV=Yes	**CogR**	20	1	0.04	0.15	0.08	0.15	0.75	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	0.69	4.46	0.43	4.46	1.23

	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	0.87	.703	0.11	0.15	.445	0.11	0.36	.749
No vs Yes IPV: CogR	-0.77	0.88	.380	-1.82	0.59	.002	-1.82	0.68	.008
No vs Yes IPV: Supp	2.11	0.83	.011	0.64	0.13	.000	0.64	0.34	.058
None vs CogR: IPV=No	-1.79	0.65	.006	-1.08	0.17	.000	-1.08	0.30	.000
None vs Supp: IPV=No	-0.37	0.65	.574	-0.14	0.12	.236	-0.14	0.27	.597
CogR vs Supp: IPV=No	1.43	0.65	.028	0.93	0.17	.000	0.93	0.30	.002
None vs CogR: IPV=Yes	-2.90	1.05	.006	-3.01	0.59	.000	-3.01	0.71	.000
None vs Supp: IPV=Yes	1.41	1.00	.160	0.38	0.16	.016	0.38	0.41	.353
CogR vs Supp: IPV=Yes	4.31	1.02	.000	3.39	0.59	.000	3.39	0.70	.000
No/Yes IPV differ by None/Cog	-1.11	1.24	.372	-1.93	0.61	.002	-1.93	0.77	.012
No/Yes IPV differ by None/Sup	1.78	1.20	.138	0.52	0.20	.009	0.52	0.49	.287
No/Yes IPV differ by Cog/Sup	2.88	1.21	.372	2.46	0.61	.002	2.46	0.76	.012

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 <u>Heterogeneous Negative Binomial</u> 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
estat ic, n(225)  // AIC and BIC match SAS
```

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
------------------------------------------------------------------------------
```

```
TITLE "SAS Heterogeneous Negative Binomial Model using NLMIXED";
PROC NLMIXED DATA=work.Example3 METHOD=GAUSS TECH=QUANEW GCONV=1e-12;
* Must list all parms to be estimated here with start values (from Stata here);
* Bs = fixed effects predicting the mean;
* Ds = fixed effects predicting the dispersion (scale factor);
PARMS B0 0.99 B1ipv  0.11 B2NvC -1.08 B3NvS -0.14 B4ipvxNvC  -1.93 B5ipvxNvS 0.52
      D0 0.48 D1ipv -0.81 D2NvC  0.64 D3NvS  0.05 D4ipvxNvC -11.95 D5ipvxNvS 0.60;
 * Linear predictor for the log count mean;
   LogCount = B0 + B1ipv*ipv + B2NvC*NvC + B3NvS*NvS
            + B4ipvxNvC*ipv*NvC + B5ipvxNvS*ipv*NvS;
   ExpCount = EXP(LogCount); * Inverse link;
 * Linear predictor for the log of the dispersion scale parameter;
   LogScale = D0 + D1ipv*ipv + D2NvC*NvC + D3NvS*NvS
            + D4ipvxNvC*ipv*NvC + D5ipvxNvS*ipv*NvS;
   ExpScale=EXP(LogScale); * Inverse Link;
 * Log-likelihood for NB-1 (constant) or NB-2 (multiplicative);
   * NB-1: LL = aggr*LOG(ExpScale) - (aggr+(1/(ExpScale/ExpCount)))*LOG(1+ExpScale)
              + lgamma(aggr+(1/(ExpScale/ExpCount)))
              - lgamma(1/(ExpScale/ExpCount)) - lgamma(aggr+1);
   * NB-2; LL = aggr*LOG(ExpScale*ExpCount)
              - (aggr+(1/ExpScale))*LOG(1+(ExpScale*ExpCount))
              + lgamma(aggr+(1/ExpScale)) - lgamma(1/ExpScale) - lgamma(aggr+1);
MODEL aggr ~ GENERAL(LL);
RUN;
```

SAS Output for Heterogeneous Neg Bin:

```
              Fit Statistics
-2 Log Likelihood                  813.3
AIC (smaller is better)            837.3
AICC (smaller is better)           838.8
BIC (smaller is better)            878.3

                                    Parameter Estimates
                          Standard                               95% Confidence
Parameter    Estimate        Error     DF   t Value    Pr > |t|        Limits          Gradient
B0             0.9995       0.1931    225      5.18      <.0001     0.6190    1.3801    0.000060
B1ipv          0.1148       0.2951    225      0.39      0.6975    -0.4666    0.6963    0.000074
B2NvC         -1.0780       0.3396    225     -3.17      0.0017    -1.7472   -0.4088    -0.00002
B3NvS         -0.1443       0.2765    225     -0.52      0.6023    -0.6893    0.4006    0.000042
B4ipvxNvC     -1.9335       0.7060    225     -2.74      0.0067    -3.3247   -0.5423    0.000015
B5ipvxNvS      0.5247       0.4395    225      1.19      0.2337    -0.3413    1.3907    0.000041
D0             0.4753       0.2697    225      1.76      0.0794   -0.05620    1.0067    -0.00003
D1ipv         -0.8075       0.5893    225     -1.37      0.1719    -1.9687    0.3536    -0.00001
D2NvC          0.6412       0.4645    225      1.38      0.1689    -0.2742    1.5566    -5.74E-6
D3NvS         0.04997       0.3927    225      0.13      0.8989    -0.7240    0.8239    -0.00003
D4ipvxNvC    -11.9500       236.93    225     -0.05      0.9598    -478.83    454.93    1.979E-6 → Uh oh...
D5ipvxNvS      0.6048       0.7515    225      0.80      0.4218    -0.8761    2.0858    0.000011
```

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 using SAS output [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 using maximum likelihood using SAS GLIMMIX 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. SAS ESTIMATE statements were used to request simple slopes and model-predicted outcomes.

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 $\chi^2/DF = 4.57$). 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, indicating that the observed 0 values were adequately captured within the negative binomial distribution (Pearson $\chi^2/DF = 1.09$). 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.