Count Data

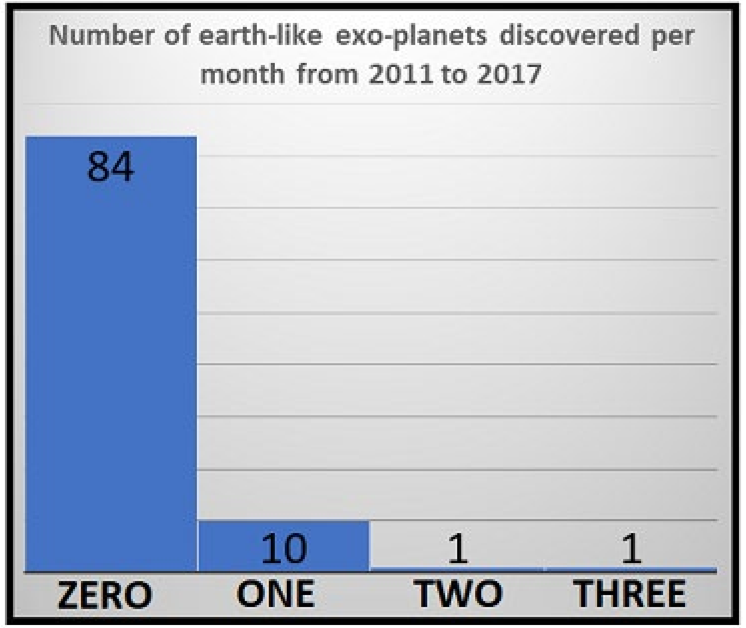
Rahul Telang

Count data

- How do you account for heterogeneity in your sample where the mean rate (λ) is not same for all observations?
- If there are excessive zeros in the sample, how do you account for those?

Poisson and zero inflation

- What do to when count data has too many zeros?
  - Number of times a machine fails each month
  - Number of exoplanets discovered each year
  - The number of billionaires living in every single city in the world.



Too many zeros

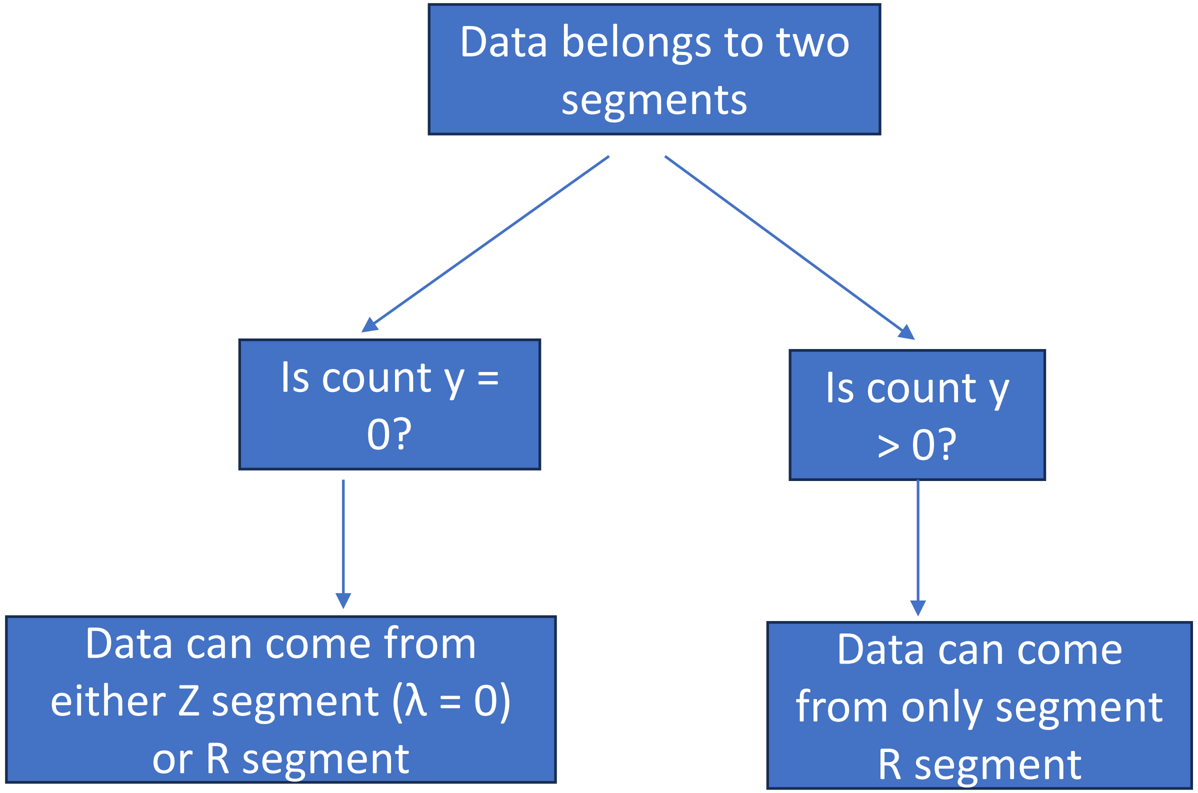
- Excessive zeros could be because some observations are always zero and may not follow Poisson data generation process, but they are part of the sample.
- There must be a mechanism to classify such observations and exclude them from analysis. Otherwise, the results will be erroneous.

Zero count

- However, observing y = 0 does not signal that it does not follow Poisson distribution and hence should be excluded. Recall P(y=0) is also possible when mean rate ($\lambda > 0$).
- To accommodate zero counts, we need to allow the model to explicitly classify data (y = 0) which does not follow Poisson distribution.

Zero inflated model structure

- The model introduces two segments (regular - R or zero - Z)-



Likelihood function

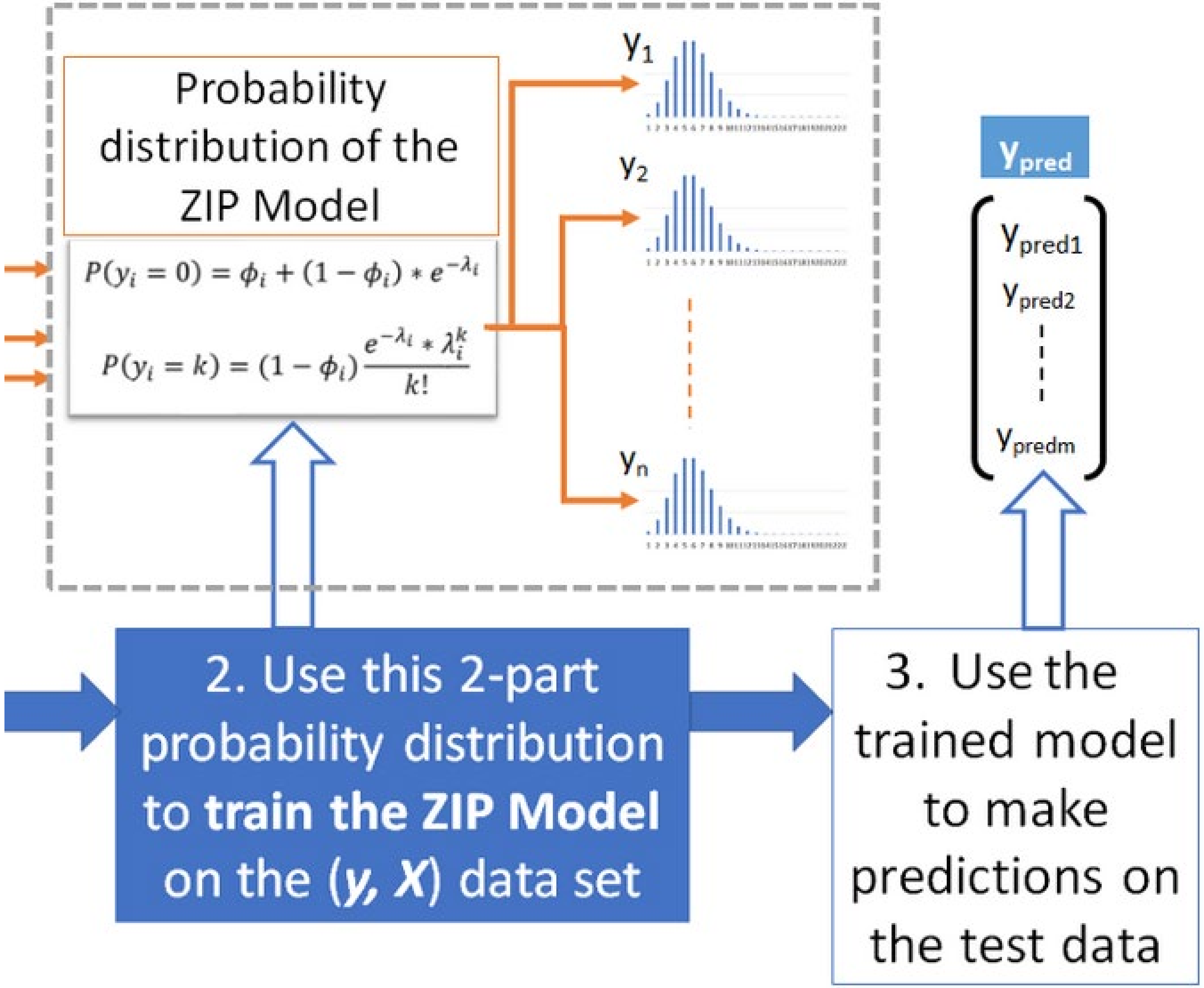
- Observations belong to zero segment (Z) with *probability* φ and belong to regular segment (R) with *probability* $(1- \varphi)$.
- We write the likelihood of observing data y_i,
  - if $y_i = 0$, then data can come from both segments with respective probabilities
    - $P(y_i =0) = \phi + (1- \phi) \frac{e^{-\lambda} * \lambda^0}{0!} = \phi + (1- \phi)\, e^{-\lambda}$
  - When we observe $y_i > 0$, it can only come from R segment
    - $P(y_i =1,..,n) = (1- \phi) \frac{e^{-\lambda} * \lambda^{y_i}}{y_i!}$
- As before λ is affected by covariates Z. So $\lambda = \lambda_0\, e^{\delta Z}$

Zero inflated model

- We need a classification tool which can classify an observation belonging to zero segment.
- Probability ϕ can be written as a logistics distribution such that
  - $\phi = \frac{e^{X\beta}}{1+e^{X\beta}}$ where X is the covariates which are used for classifying segments and β are the estimates which capture the impact of X on classification.
- With these probabilities, one can readily write the likelihood function and maximize it to recover β and δ.

Zero inflated model

- Once we have φ in hand, we can fit Poisson model.
- Statsmodel provide ZeroinflatedPoisson() function to estimate the parameters in GLM models.

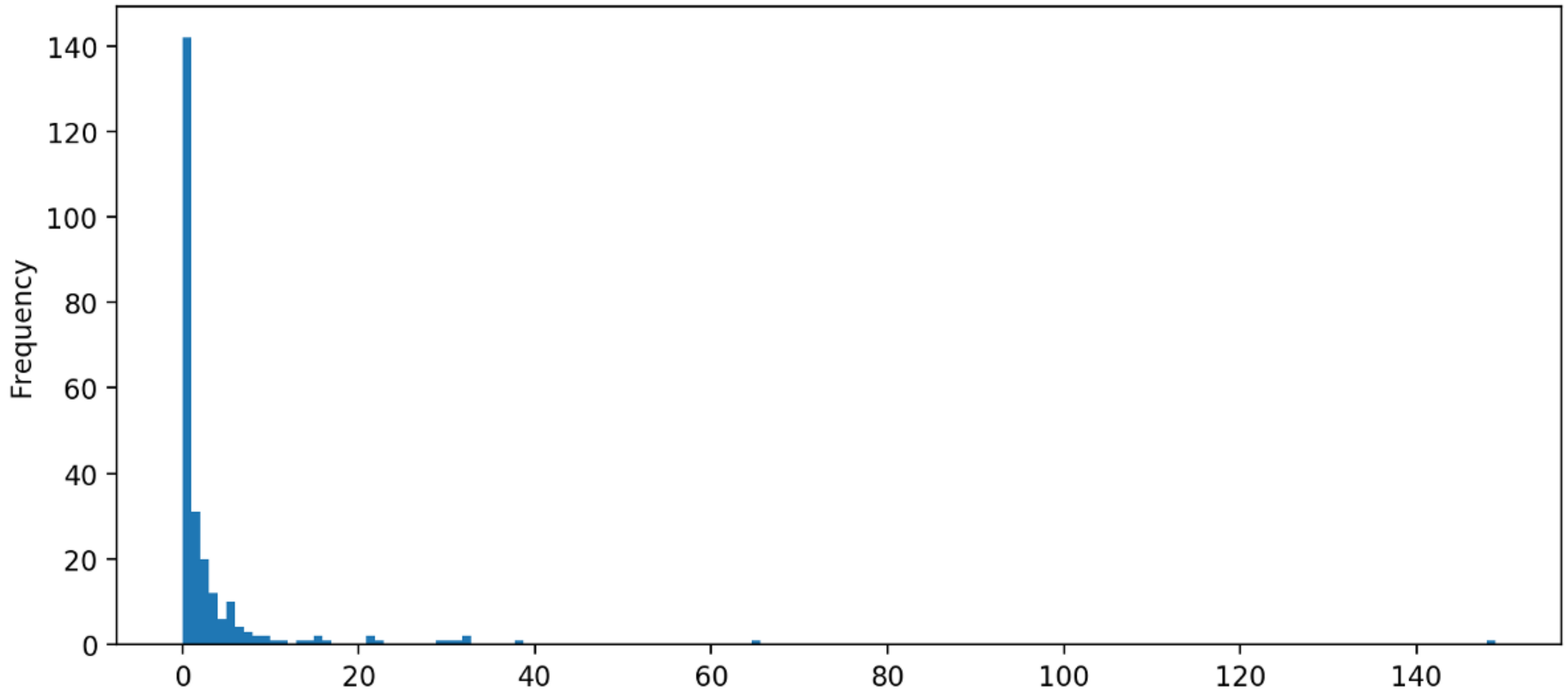


An Example

- See the reading material on fish purchase.
- Take a different example : During camping trips people also may for fishing. We have data on number of fish caught by groups who go for camping trips.
  - data consists of camping trips taken by 250 groups of people:
- Variables in the data set
  - **FISH_COUNT:** The number of fish that were caught. This will be our dependent variable ***y***.
  - **LIVE_BAIT:** A binary variable indicating whether live bait was used.
  - **CAMPER:** Whether the fishing group used a camper van.
  - **PERSONS:** Total number of people in the fishing group. Note that in some groups, none of them may have fished.
  - **CHILDREN:** The number of children in the camping group.

Data

- No. of fish caught looks like this. Clearly there are lots of zeros



Model

- We want to predict number of fish caught.
- Since this is count data, Poisson will be a good starting point. However, there are many of zeros, zero inflated python can be suitable choice.
- In a zero inflated model
  - We model the number of fish caught (using Poisson distribution)
  - The fish is caught only after campers go for fishing. We need a classifier that can classify campers who for fishing. We use logistic distribution.
- Fortunately, GLM provides a function which can model zero inflated Poisson
  - sm.ZeroInflatedPoisson(endog=y_train, exog=X_train, exog_infl=X_train, inflation='logit').fit()
- exog_infl - is the list of covariates for logistics model. We can define the covariates in this vector.

Results

- It does not converge. We specify maxiter=100 in the fit function when we estimate the equation. See next slide.

ZeroInflatedPoisson Regression Results

Dep. Variable:	FISH_COUNT	No. Observations:	205
Model:	ZeroInflatedPoisson	Df Residuals:	200
Method:	MLE	Df Model:	4
Date:	Wed, 27 Sep 2023	Pseudo R-squ.:	0.3747
Time:	13:23:38	Log-Likelihood:	-619.57
converged:	False	LL-Null:	-990.77
Covariance Type:	nonrobust	LLR p-value:	2.280e-159

	coef	std err	z	P>|z|	[0.025	0.975]
inflate_Intercept	0.3461	0.976	0.354	0.723	-1.567	2.259
inflate_LIVE_BAIT	-0.1522	0.836	-0.182	0.856	-1.790	1.486
inflate_CAMPER	-0.2632	0.400	-0.658	0.511	-1.047	0.521
inflate_CHILDREN	1.6484	0.329	5.014	0.000	1.004	2.293
inflate_PERSONS	-0.4904	0.209	-2.341	0.019	-0.901	-0.080
Intercept	-2.2451	0.308	-7.282	0.000	-2.849	-1.641
LIVE_BAIT	1.5296	0.285	5.369	0.000	0.971	2.088
CAMPER	0.6587	0.101	6.517	0.000	0.461	0.857
CHILDREN	-1.1055	0.094	-11.798	0.000	-1.289	-0.922
PERSONS	0.8644	0.045	19.392	0.000	0.777	0.952

Results

- sm.ZeroInflatedPoisson(endog=y_train, exog=X_train, exog_infl=X_train, inflation = 'logit'). fit(maxiter=100)
- For the logit part - children and persons are significant. Campers with children are more likely to be classified as λ=0.
- One unit increase in children increases the log odds of λ=0 by 72% Similarly when one more person decreases the log odds of being classified as λ=0 by 12%
- For the count model, all covariates are significant and have the same interpretation as Poisson model. A unit increase in live bait increases the count by exp(1.70) =547% .

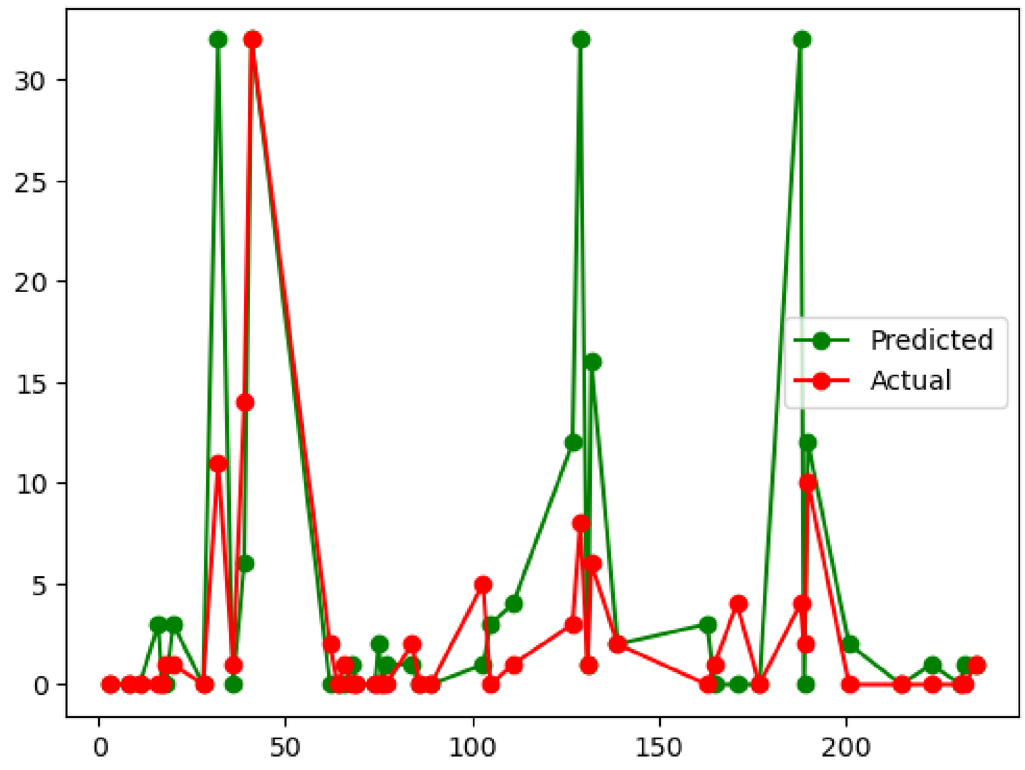
```
                     ZeroInflatedPoisson Regression Results
==============================================================================
Dep. Variable:                FISH_COUNT   No. Observations:                  200
Model:               ZeroInflatedPoisson   Df Residuals:                      195
Method:                              MLE   Df Model:                            4
Date:                   Wed, 27 Sep 2023   Pseudo R-squ.:                  0.3276
Time:                           13:31:42   Log-Likelihood:                -465.27
converged:                          True   LL-Null:                       -691.92
Covariance Type:               nonrobust   LLR p-value:                 8.391e-97
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
inflate_Intercept     1.1474      1.021      1.124      0.261      -0.853       3.148
inflate_LIVE_BAIT     0.5275      0.890      0.593      0.553      -1.217       2.272
inflate_CAMPER       -0.9817      0.394     -2.489      0.013      -1.755      -0.209
inflate_CHILDREN      1.7247      0.346      4.990      0.000       1.047       2.402
inflate_PERSONS      -0.8855      0.223     -3.979      0.000      -1.322      -0.449
Intercept            -1.7911      0.288     -6.226      0.000      -2.355      -1.227
LIVE_BAIT             1.7026      0.247      6.881      0.000       1.218       2.188
CAMPER                0.1928      0.100      1.923      0.054      -0.004       0.389
CHILDREN             -0.9827      0.102     -9.598      0.000      -1.183      -0.782
PERSONS               0.7105      0.048     14.659      0.000       0.616       0.806
```

Individual predictions

- It does pretty well.
- zip_predictions = zip_training_results.predict(X_test,exog_infl =X_test, which = 'mean’)

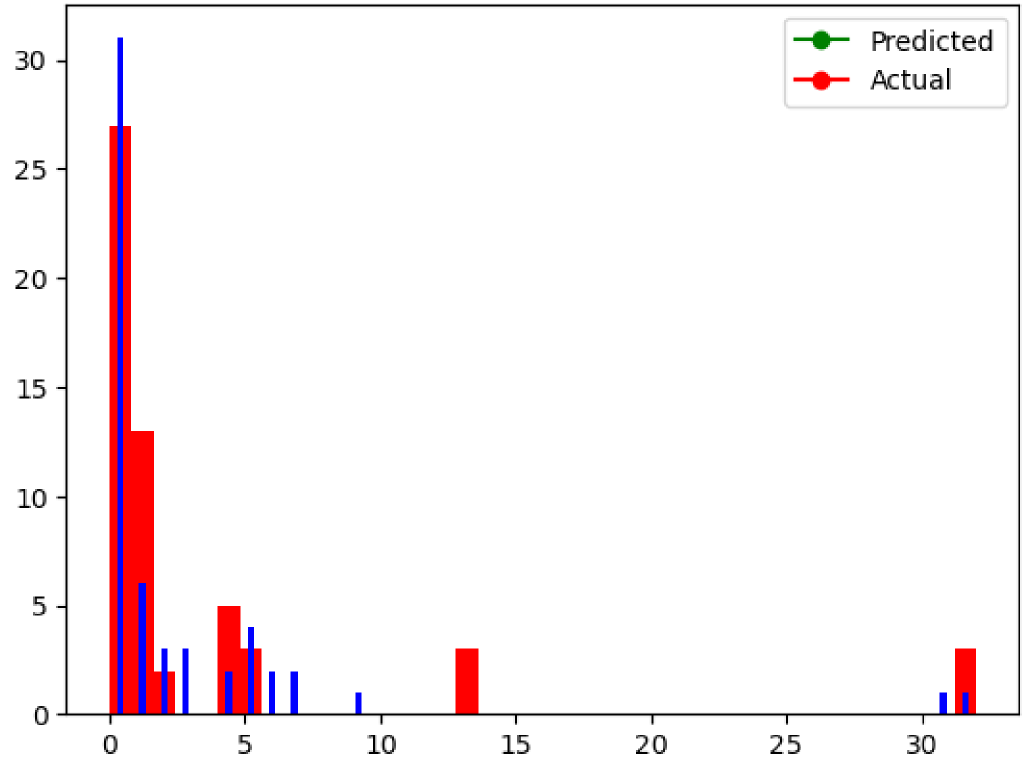
- One can output different predictions - see here
- https://www.statsmodels.org/dev/generated/statsmodels.discrete.count_model.ZeroInflatedPoisson.predict.html

Predicted versus actual counts using the ZIP model



Aggregate Predictions

- We can aggregate the data in bins
- Here we have divided the fish count in 40 bins



Heterogeneity

- In all the models we have studied (timing, count and even choice), we assume that all observation are homogeneous and come from same distribution. In Poisson model, we estimate one λ for all observations.
- However, observations are heterogeneous, and data could be come from two different Poisson distribution with parameters λ_1 and λ_2.
- GLM focuses more on the estimating the impact of covariates (β) than the parameter of the distribution
- Estimating two λ is non-trivial.

Heterogeneity

- We can readily accommodate heterogeneity by assuming two segments (Seg1 and Seg2). Probability of observing data (outcome y) will be a weighted average of Seg1 and Seg2.
- The data comes from segment 1 with probability ϕ, and Segment 2 with probability (1- ϕ).
  - $P(y_i) = (\phi) \frac{e^{-\lambda_1} * \lambda_1^{y_i}}{y_i!} + (1-\phi) \frac{e^{-\lambda_2} * \lambda_2^{y_i}}{y_i!}$
- One can readily write the likelihood expression and estimate parameters λ_1 and $\lambda_{2.}$ However, GLM does not provide a function to estimate λ_1 and $\lambda_{2.}$

Negative Binomial

- However, if we allow λ to be heterogeneous in a more general way, GLM offers a model
- λ is heterogeneous not in discrete segments but captured by continuous distribution (gamma distribution). The probability of observing a data

$f^m(y) = f(y/\lambda)*f(\lambda)$

where $f(y/\lambda)$ is Poisson distribution (y conditional on λ)

$f(\lambda)$ is a distribution of λ (assumed to be gamma).

Recall when we assume two segments, $f(\lambda)$ is a discrete distribution of $f(\lambda) = \phi \lambda_1 + (1- \phi) \lambda_2$

- It turns out that mixture of Poisson $f(y/\lambda)$ with gamma $f(\lambda)$, leads to a widely used negative binomial distribution.

Negative Binomial

- The probability distribution for NBD is
  - $P(y; p, r) = \frac{(y + r - 1)!}{(y!(r-1)!)} p^r (1-p)^r$

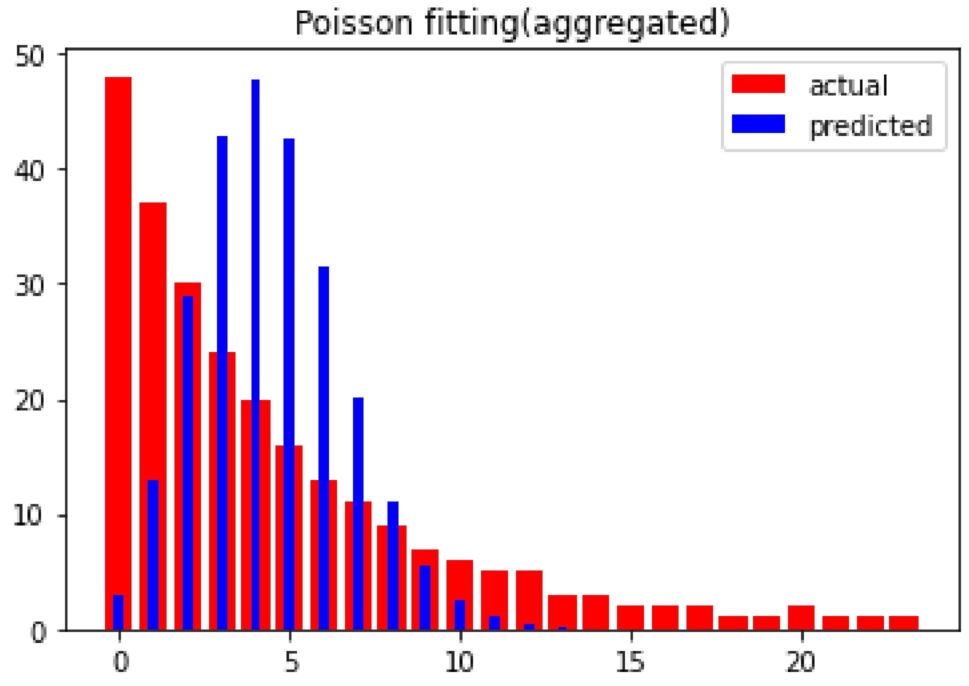
  where y is number of failures, r is number of success and p is the probability of success.
- A critical assumption in Poisson model is that the mean μ = λ is equal to the variance σ^2 = λ.
- Clearly in reality this assumption does not always hold true.
- When we let λ to be heterogeneous, we do not estimate Poisson distribution but instead use negative binomial distribution

NBD

- Statsmodel in python allows us to estimate NBD model like Poisson model.
  - sm.negativebinomial( )
- Along with covariates, it also estimate a parameters α where variance
  - Variance = mean + α mean^2
- Higher value of α signals how different mean is from the variance.
- Interpretation of estimates remain same as in Poisson. A Unit increase in X leads to β unit increase in $\log(\lambda)$.
- Going back to the billboard exposure problem we solved using Poisson

Aggregate Predictions using Poisson

- We calculate P(x=0,1…).
- 48 users had 0 exposure. We can calculate P(x=0). Since we have 250 users in the sample, number of people who have 0 exposure is 250*p(x=0).
- As one can see, the fit is not great. One possibility is that assumption of homogeneous λ in too restrictive.
- What if we allow the distribution as negative Binomial?



NBD

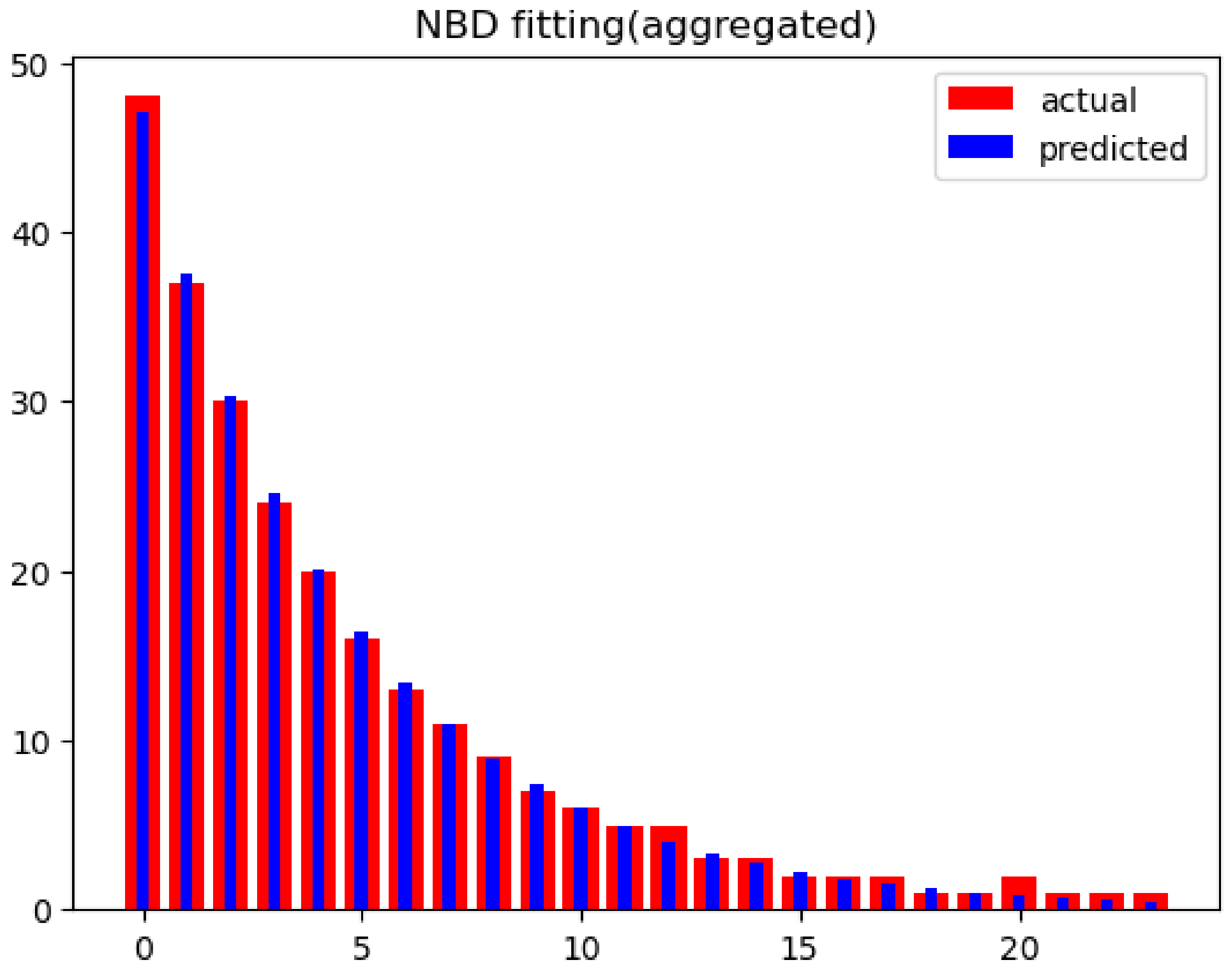
- NBD_results = sm.negativebinomial("exposures~1' , df_l).fit()
- Estimate for Intercept, which is λ_0, remains the same.
- Estimate for α is large and statistically significant. This suggest that mean ≠ variance.
- We want to make prediction using NBD. Recall the mean prediction is still going to be $\lambda^{1.49}$ = 4.35.
- We can use negative binomial pmf to calculate the probability of each number of exposure

```
                     NegativeBinomial Regression Results
==============================================================================
Dep. Variable:              exposures   No. Observations:                  250
Model:               NegativeBinomial   Df Residuals:                      249
Method:                           MLE   Df Model:                            0
Date:                Sun, 01 Oct 2023   Pseudo R-squ.:               6.499e-12
Time:                        11:18:21   Log-Likelihood:                -649.69
converged:                       True   LL-Null:                       -649.69
Covariance Type:            nonrobust   LLR p-value:                       nan
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.4943      0.071     21.080      0.000       1.355       1.633
alpha          1.0317      0.121      8.539      0.000       0.795       1.269
==============================================================================
```

NBD predictions

- NBD_pmf = stats.nbinom.pmf(x_range, n, p)
- One needs to convert estimates intercept (which is mean) mu and alpha into n and p.
- mu, alpha = NBD_results.params
- It can be shown that
- n = 1/alpha
  p = 1/(1+alpha*exp(mu))
-

  NBD_pmf = stats.nbinom.pmf(x_range, n, p)



NBD for Fish count

- We estimated zero inflated Poisson model for number of fish caught and see evidence of excessive zeros.
- What if we estimate NBD model? How does the prediction look?

- We use the function
  - sm.NegativeBinomial(endog=y_train, exog=X_train).fit()
- We are using NBD instead of Poisson. Going back to our example

Dep. Variable:	FISH_COUNT	No. Observations:	194
Model:	NegativeBinomial	Df Residuals:	189
Method:	MLE	Df Model:	4
Date:	Mon, 02 Oct 2023	Pseudo R-squ.:	0.1805
Time:	10:46:14	Log-Likelihood:	-294.54
converged:	True	LL-Null:	-359.44
Covariance Type:	nonrobust	LLR p-value:	4.318e-27

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-3.0022	0.546	-5.503	0.000	-4.071	-1.933
LIVE_BAIT	1.1990	0.474	2.531	0.011	0.271	2.127
CAMPER	1.1848	0.259	4.572	0.000	0.677	1.693
CHILDREN	-2.1643	0.225	-9.623	0.000	-2.605	-1.724
PERSONS	1.0317	0.118	8.747	0.000	0.800	1.263
alpha	1.5294	0.290	5.277	0.000	0.961	2.097

For Statsmodel, Python used this pdf for NBD.

$$P(y|\alpha, \beta_0, \beta_1, \ldots) = \frac{\Gamma(1/\alpha+y)}{\Gamma(1/\alpha)y!}\left(\frac{1}{\alpha\exp(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots)+1}\right)^{1/\alpha}\left(\frac{\alpha\exp(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots)}{\alpha\exp(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots)+1}\right)^{y}$$

Estimates

- $\alpha > 0$ suggest that variance is bigger than the mean.
- $\alpha > 1.52$ and significant, $\alpha > 0$ suggests that variance is different than mean
  - $Var(\lambda) = mean + \alpha\ mean^2$
- The interpretation remains the same as in Poisson. A unit change in X will impact β unit change in log(y) - all else constant. So, a unit change in LIVE_BAIT changes the log of fish count by 1.19 or one can tale the exponent and hence by ~($e^{1.19}$)% = 328%
- Intercept is the mean of Poisson with other covariates held at zero so $\lambda = \exp(-3.0) = 0.05$.

What about Zero inflated NBD?

- In a zero inflated model, expected number of outcome (say visits) is
  - E(visits) = P(visits=0) *0 + P(visits>0)*E(visits)
- In Zero-inflated Poisson model, we calculate the probability of visits=0 and find the expected number E(.) based on Poisson distribution. We can do the same thing using NBD.
- In zero inflated NBD, we follow the same process and outline NBD as the distribution.

Zero inflated NBD

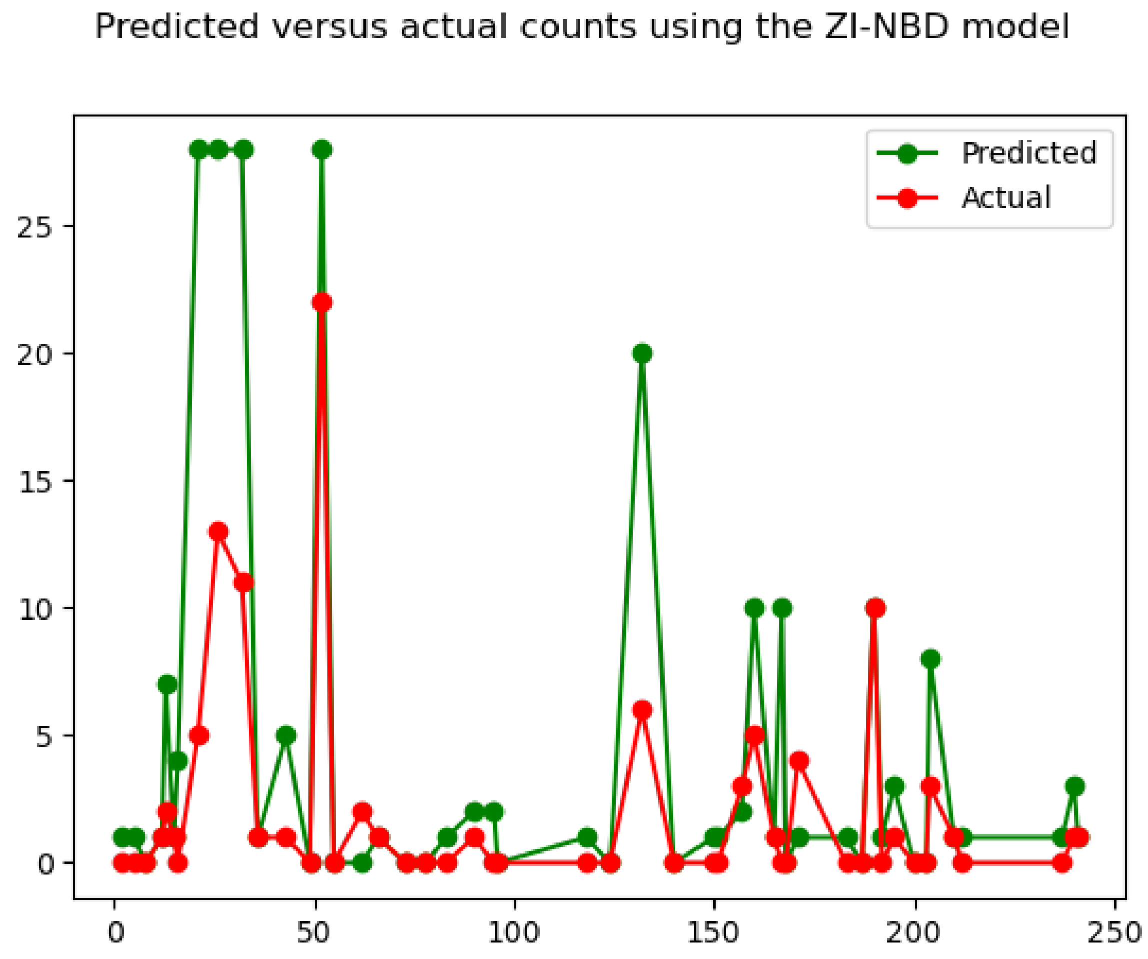
- In Python sm.ZeroInflatedNegativeBinomialP(endog=y_train, exog=X_train, exog_infl=X_train, inflation='logit').fit(maxiter=100)

	coef	std err	z	P>|z|	[0.025	0.975]
inflate_Intercept	1.6393	9.288	0.177	0.860	-16.564	19.843
inflate_LIVE_BAIT	0.5837	7.440	0.078	0.937	-13.998	15.165
inflate_CAMPER	-1.8998	3.152	-0.603	0.547	-8.077	4.277
inflate_CHILDREN	-8.9837	316.294	-0.028	0.977	-628.908	610.941
inflate_PERSONS	-2.4099	3.521	-0.684	0.494	-9.310	4.491
Intercept	-2.7423	0.584	-4.698	0.000	-3.886	-1.598
LIVE_BAIT	1.4577	0.461	3.159	0.002	0.553	2.362
CAMPER	0.1864	0.282	0.661	0.508	-0.366	0.739
CHILDREN	-1.8049	0.205	-8.822	0.000	-2.206	-1.404
PERSONS	1.0972	0.153	7.177	0.000	0.798	1.397
alpha	1.8402	0.365	5.042	0.000	1.125	2.55

Predictions

- These are different commands one can use to see various output
- https://www.statsmodels.org/dev/generated/statsmodels.discrete.count_model.ZeroInflatedPoisson.html

Predictions



Summary

- For any count data, Poisson is the starting distribution. In GLM Poisson fits in the exponential family.
- GLM estimate the impact of covariates and parameters of distribution.
- Many times, there are too many zeros and we use Logit to classify zeros and positive numbers before applying Poisson distribution.
- Poisson imposes the restriction that mean=variance. We use Negative binomial distribution to relax this assumption.