Software defect prediction with Zero-inflated Poisson models

MADSESE 2019

Madrid 5 de Junio 2019

Daniel Rodríguez, Javier Dolado, Javier Tuya, Dietmar Pfahl
UAH, UPV/EHU, UniOvi, U. Tartu

TIN2016-76956-C3-R, QARE, BadgePeople, TESTEAMOS

Software defect prediction with Zero-inflated Poisson models

- Motivation
- Equinox dataset
- Several approaches to fitting regression models. ZIP model.
- Conclusions

Motivation

- The number of *Software Defects* found in a software product can be assimilated to the *"count data"* concept that is used in many disciplines, because the outcome, number of defects of whatever software process, is a count.
- We take the data that is available in public repositories
- There are several ways of analyzing count data. The classical Poisson or negative binomial regression model can be augmented with zero-inflated Poisson and zero-inflated negative binomial models to cope with the excess of zeros in the count data.
- There are many packages and new proposals for analyzing Zero-inflated data. We wanted to compare them on a dataset.

Equinox dataset

- This dataset is part of the Bug prediction dataset and corresponds to a Java Framework included the Eclipse project. Many variables can be selected.
- Only a few are relevant

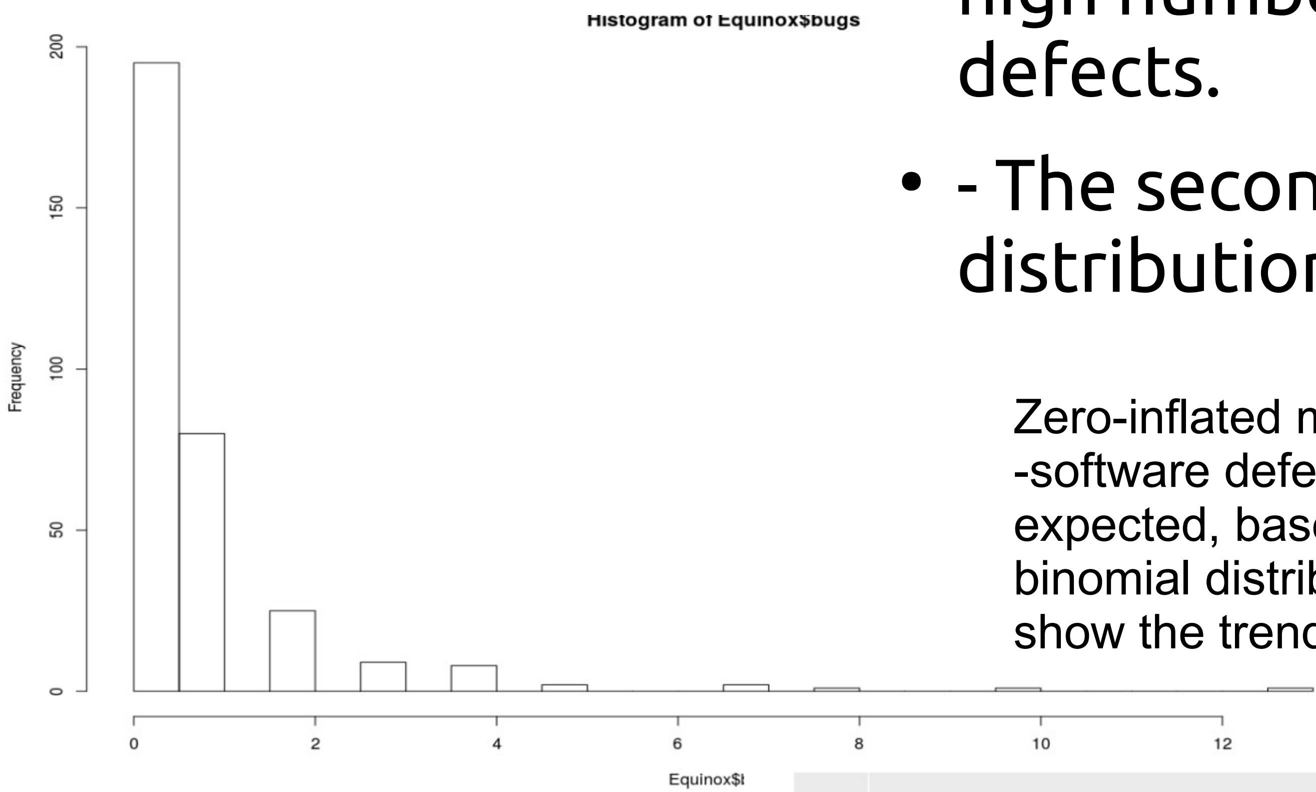
classname	cbo	dit	fanIn	fanOut	lcom	noc	numberOfAt	numberOfAttr
ext::framework::a::importer::Act	6	1	0	6	3	0	0	0
org::eclipse::osgi::framework::ir	14	1	3	11	300	0	25	0
org::osgi::framework::ServiceEv	4	1	4	0	3	0	6	0
org::eclipse::osgi::framework::ir	1	2	1	0	0	0	1	0
substitutes::z::Fz	0	1	0	0	0	0	0	0
circularity::test::Activator	2	1	0	2	1	0	0	0
org::eclipse::osgi::framework::ir	12	2	3	9	45	0	6	0
org::eclipse::osgi::internal::mod	3	2	2	1	1	0	0	2
org::eclipse::osgi::internal::reso	2	2	1	1	10	0	0	2
org::eclipse::osgi::internal::mod	10	1	8					
org::osgi::framework::ServiceP	7	1	1					
org::eclipse::osgi::framework::ir	22	1	18					
nativetest::d::Activator	4	1	0					
substitutes::y::Ay	0	1	0					
org::eclipse::osgi::framework::ir	0	2	0					
substitutes::x::Kx	0	1	0					
org::eclipse::equinox::launcher:	40	1	3					
nativetest::b2::Activator	4	1	0					
org::eclipse::osgi::internal::base	12	2	1					

numberO	numberOfP	numberO	rfc	wmc	bugs	nonTrivialBugs	majorBugs
1	0	2	14	3	0	0	0
7	0	7	172	115	0	0	0
0	0	3	3	3	1	0	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	2	8	2	0	0	0
1	0	4	34	39	0	0	0
0	0	0	3	3	0	0	0
0	0	5	9	4	1	0	0
0	0	5	12	7	0	0	0
3	0	5	29	22	1	0	0
0	0	4	17	8	0	0	0
[illegible]	0	2	8	2	0	0	0
[illegible]	0	0	0	0	0	0	0
[illegible]	137	0	0	0	0	0	0
[illegible]	0	0	0	0	0	0	0
[illegible]	2	3	836	410	3	0	1
[illegible]	0	2	8	2	0	0	0
[illegible]	0	17	54	38	0	0	0
[illegible]	0	4	4	4	0	0	0
[illegible]	0	0	0	0	1	0	1
0	0	2	10	5	0	0	0

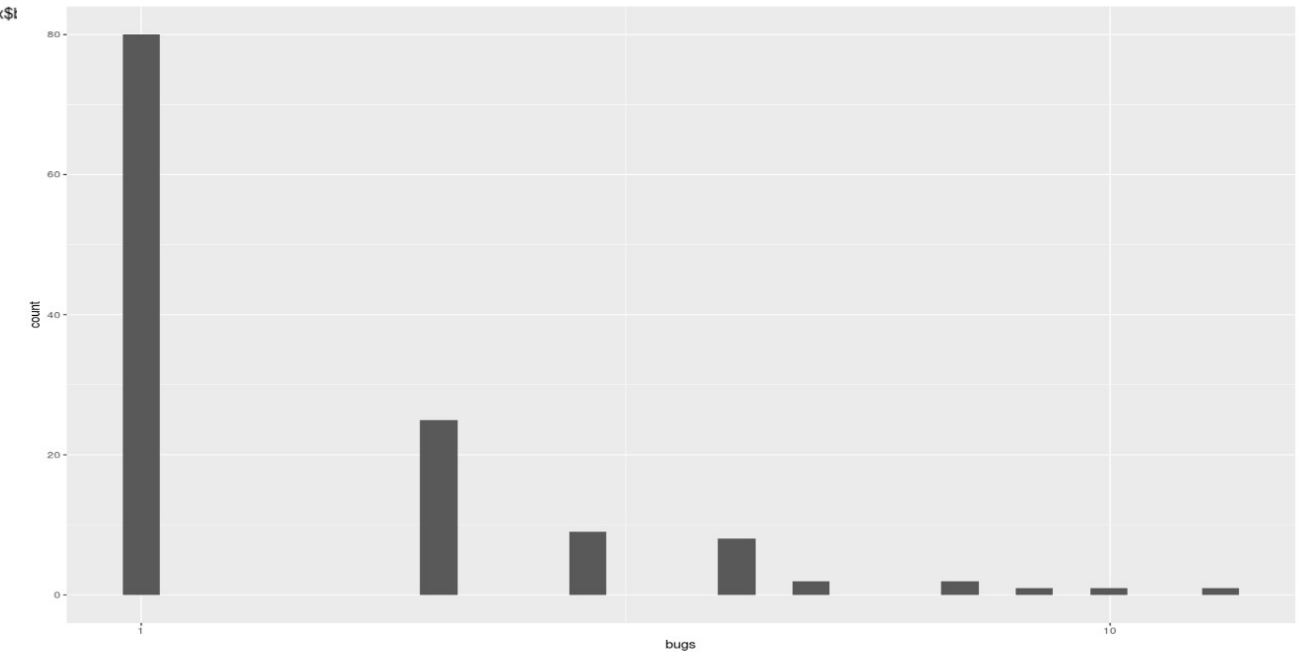
```
(bugs~ wmc+rfc+cbo+lcom,
data=equinox,
ziformula=~numberOfLinesOfCode,
family=poisson)
```

Equinox dataset

- \- The first histogram shows the high number of modules with no defects.
- \- The second histogram shows the distribution of the non-zero values.



Zero-inflated means that the response variable -software defects- contains more zeros than expected, based on the Poisson or negative binomial distribution. A simple histogram may show the trend.



Methods and R

- We analyzed the Equinox dataset using frequentist analysis and Bayesian analysis.
- We explored several models: Poisson, Negative Binomial, Zero Inflated Poisson, and Zero Inflated Negative Binomial
- There are many R packages that can be used to fit regression models:
  - MASS
  - pscl
  - R2Jags (Bayesian)
  - mgvc
  - glmmTMB (relatively new)

Results

Table 1. Summary of the results obtained with different R packages.

Method	*AIC*	*BIC*	*R Package*	*# Bugs predicted*
Regression	**904.8354**	927.5198	MASS	97.76806
Poisson	**632.1547**	651.0584	pscl	188.7356
Poisson	**632.2**	651.1	glmmTMB	n.a
Poisson	**632.1547**	-	mgvc	-
Neg. binom.	**644.5**	-	MASS	195.8165
Neg. binom.	**628.6**	651.2	glmmTMB	n.a.
Neg. binom.	**628.5507**	-	mgvc	-
ZIP	**606.9155**	633.3807	pscl	195.7924
ZIP	**606.9**	633.4	glmmTMB	n.a.
ZIP	**602.9** wmc	629	glmmTMB	n.a.
ZIP	-	**DIC=622.5**	Bayes RJAGS	-
ZIP	**653.4149**	-	mgvc	-
ZIP	**647.9201** wmc	-	mgvc	-
ZINB	**607.5639**	637.8098	pscl	198.2048

Conclusions

- We have build several models to fit one small dataset.
- We have run several R packages with different approaches to Zero-inflated models.
- We can say that for small datasets the method used is not important respect to the cost in time. Bayesian simulation takes time but it does not prevent getting results.
- Precision is good for ZIP models.
- But the questions remain: how to build a good strategy for collecting relevant data and estimating defects in actual software settings.