Appendix

Contents

1 Foreword

Listed below are the models and survey questions used for the study. Effect plots are provided to assist in the interpretation of the longitudinal models.

The tables were created using the R package *stargazer* by Marek Hlavac.[1]

Our data set and R source code, used for the paper and to generate this appendix, is publicly available.[2]

[1] https://cran.r-project.org/web/packages/stargazer/vignettes/stargazer.pdf

[2] https://github.com/CMUSTRUDEL/npm-badges

2 Signals of Updated Dependencies

2.1 Freshness Scores

Given a list of releases for a *single* dependency, such as

(1.0.0, 1.0.1, 1.2.1, 2.0.1-alpha, 2.0.2)

we define the *freshness* to be a weighted distance from the current version to the newest version.

For example, if the current version is given by the package as 1.0.x, then the maximum version matching this in the list above is 1.0.1. Hence we are

1.0.1 ⇒ 1.2.1: one minor version (5 points)

1.2.1 ⇒ 2.0.1-alpha: one major version (20 points)

2.0.1-alpha ⇒ 2.0.2: one patch version (1 point)

= 26 points away from the latest version.

To handle the version strings used in package.json files, we use the *npm* packages *semver* and *semver-diff*. This allows us to find, for example, the latest release fulfulling a given version string. *semver* is used by *npm* itself.

We recognize that there are different "newest releases" depending on the time. To solve this, we created a database of releases for all *npm* packages and only considered those that occured before the time under consideration.

The per-project freshness scores in the paper are averaged over all of the project's dependencies.

2.1.0.1 DeLong's ROC Test

We test to see if there is a significant difference between the base and full logistic regression models for up-to-dateness/freshness using DeLong's ROC test.

```
##
##  DeLong's test for two correlated ROC curves
##
## data:  fitted(mod.fresh.badge.hurdle) and fitted(mod.fresh.base.hurdle) by mod.fresh.badge.hurdle$y (0, 1)
## Z = -108.47, p-value < 2.2e-16
## alternative hypothesis: true difference in AUC is not equal to 0
## sample estimates:
## AUC of roc1 AUC of roc2
##   0.7742829   0.7735593
```

2.2 Effect Plots

The adoption of both dependency manager badges and information badges seems to be correlated with an increase in dependency freshness. However, the effect lasts longer for dependency manager bages, as can be seen below.

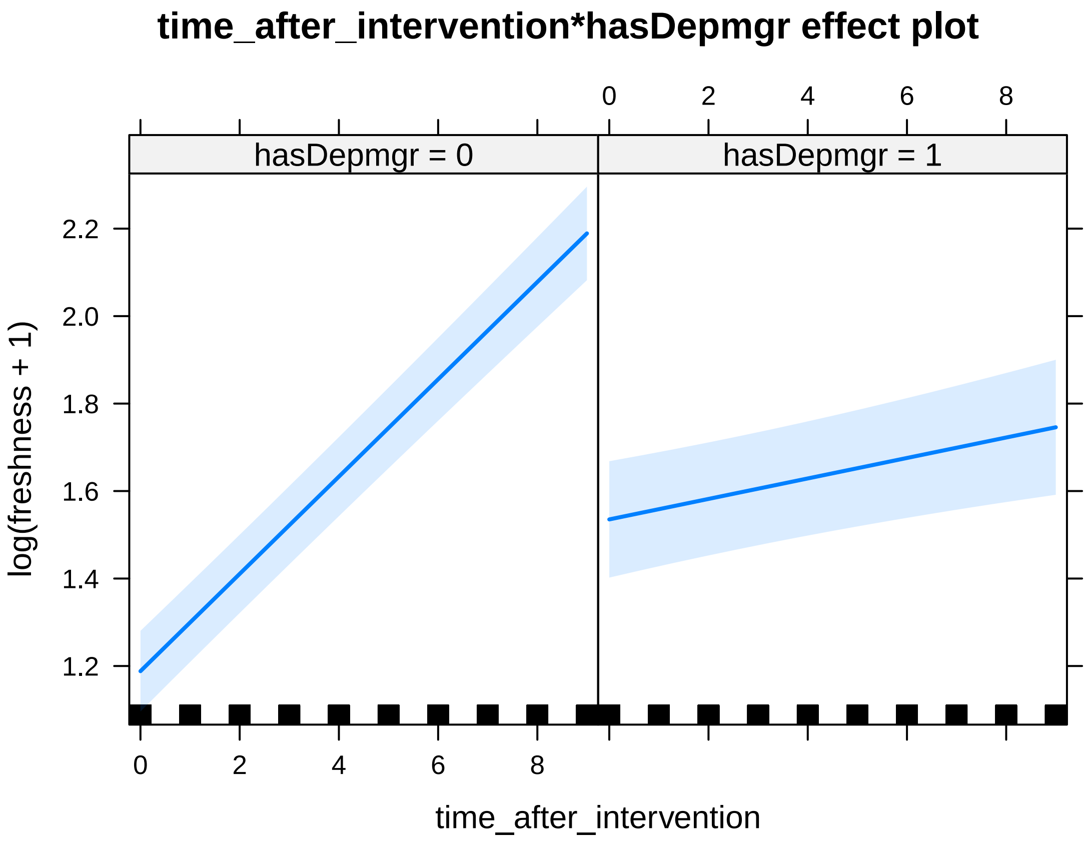
Recall that lower freshness is better (more up-to-date). Projects with dependency manager badges tend to stay at a similar level after the intervention, whereas the other classes correlate with decaying freshness (upward sloping).

```
## NOTE: time_after_intervention:hasDepmgr is not a high-order term in the model
```

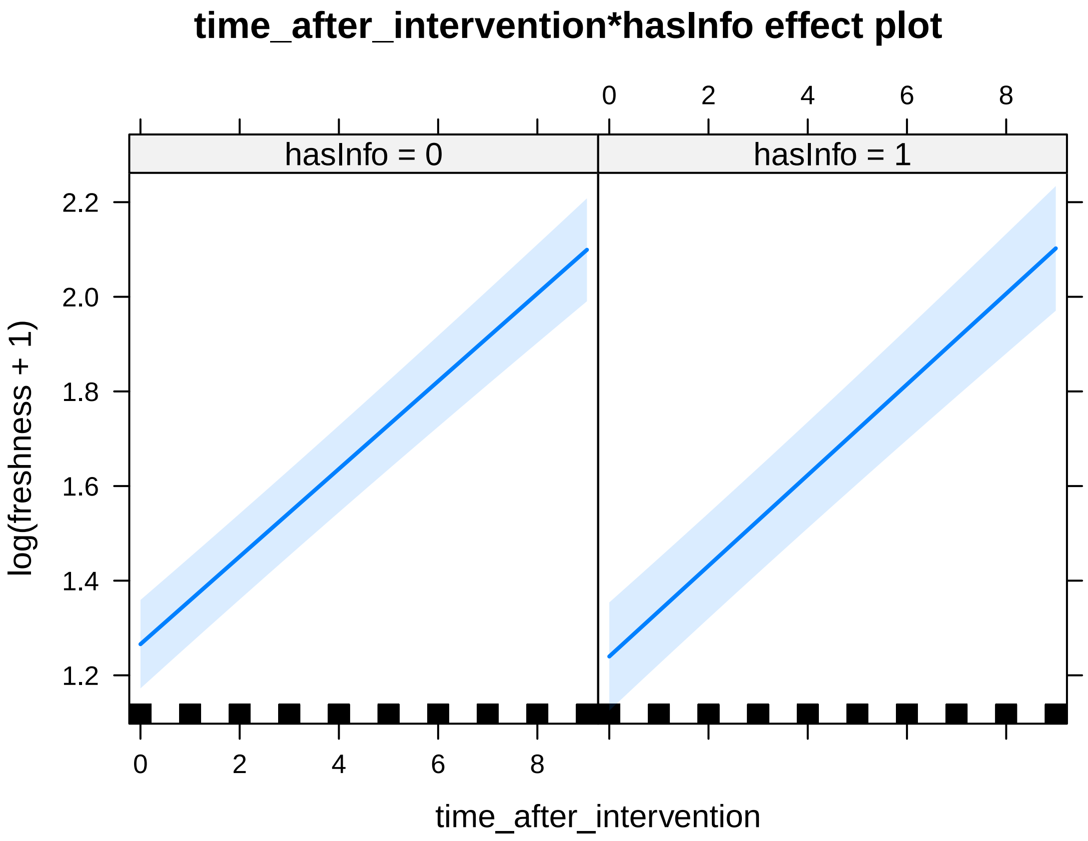
Table 1: Dependency “freshness” (1-4) at the time of study and (5) per month

	Dependent variable:				
	log(freshness + 1)		(freshness == 0)		log(freshness + 1)
	OLS		*logistic*		*linear mixed-effects*
	Base Model	Full Model	Base Model	Full Model	RDD
	(1)	(2)	(3)	(4)	(5)
log(dependencies + 1)	-0.272^{***}	-0.267^{***}	-1.771^{***}	-1.781^{***}	-0.040^{*}
log(dependents + 1)	-0.118^{***}	-0.110^{***}	0.216^{***}	0.207^{***}	-0.005
log(tsu + 1)	0.510^{***}	0.503^{***}	-0.658^{***}	-0.651^{***}	0.013
time					0.027^{***}
intervention1					-0.929^{***}
time_after_intervention					0.112^{***}
hasDepmgr1					0.446^{***}
hasInfo1					0.038
time_after_intervention:hasDepmgr1					-0.099^{***}
time_after_intervention:hasInfo1					-0.004
hasDepmgr1:hasInfo1					-0.320^{***}
time_after_intervention:hasDepmgr1:hasInfo1					0.035^{***}
log(stars + 1)	0.061^{***}	0.063^{***}	-0.085^{***}	-0.086^{***}	-0.0001
log(contributors + 1)	0.122^{***}	0.130^{***}	-0.239^{***}	-0.252^{***}	-0.041^{**}
hasExtra		-0.033^{***}		0.011	
hasDepMgrBadge1		-0.103^{***}		0.228^{***}	
hasInfoBadge1		-0.075^{***}		0.104^{***}	
hasDepMgrBadge1:hasInfoBadge1		-0.031		-0.047	
Constant	2.061^{***}	2.098^{***}	3.568^{***}	3.530^{***}	1.452^{***}
AIC	334670	334112	203930	203651	
Projects	293444	293444	293444	293444	1762(x19)
Adjusted R^2	0.224	0.228			
Pseudo-R^2			0.173	0.174	
R^2_m					0.0428
R^2_c					0.347

Note: *p<0.1; **p<0.05; ***p<0.01



```
## NOTE: time_after_intervention:hasInfo is not a high-order term in the model
```



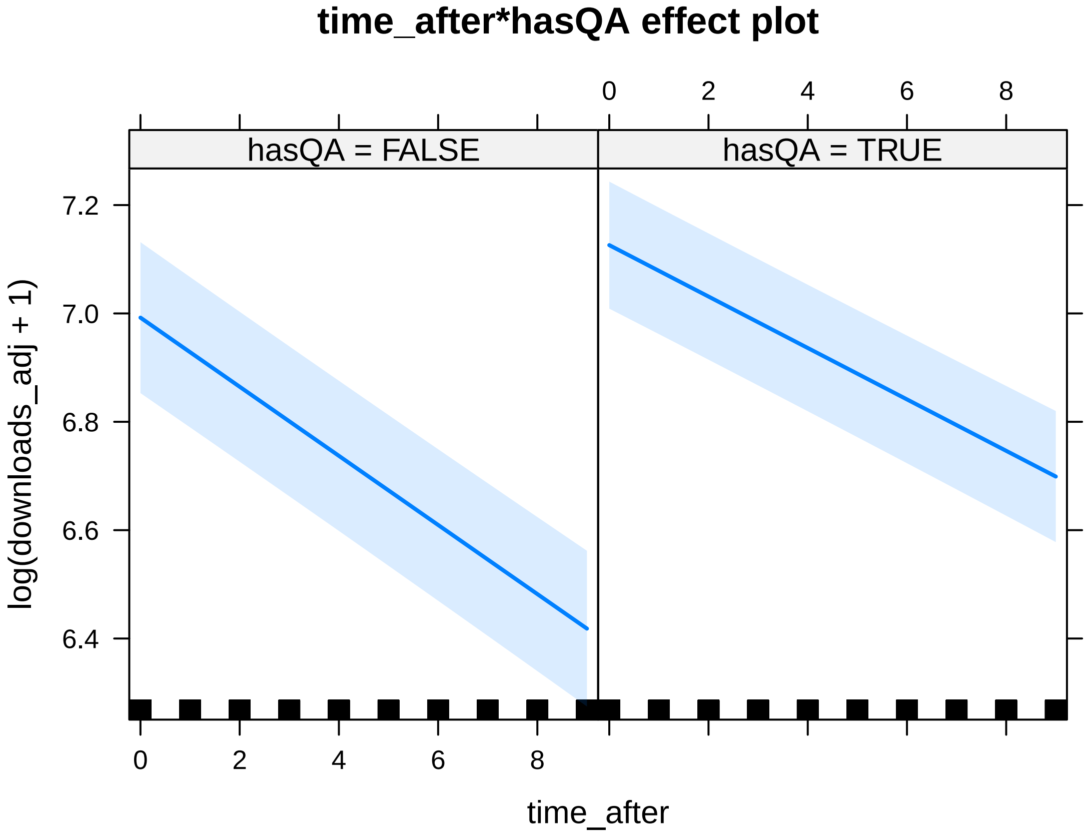
3 Signals of Popularity

Table 2: Download counts per (1-3) month of the study (4) month, as available

	Dependent variable:			
	downloads			log(downloads_adj + 1)
	negative binomial			*linear mixed-effects*
	Base Model	Full Model	Num. Badges	RDD
	(1)	(2)	(3)	(4)
log(age + 1)	−0.160***	−0.246***	−0.171***	
isPopular		4.954***		
log(size + 1)			−0.109***	
time				0.020***
time_after				−0.068***
intervention1				0.291***
log(stars + 1)	0.245***	0.121***	0.374***	0.103***
log(num_issues + 1)	0.240***	0.079***		
log(revisions + 1)	0.432***	0.517***	0.574***	0.682***
log(dependents + 1)	3.178***	1.172***	3.078***	0.947***
log(dependencies + 1)				−0.015
log(readme_size + 1)	0.109***	0.064***	0.116***	0.001
hasQABadge1		0.031***		
hasDepMgrBadge1		0.135***		
hasPopularityBadge1		0.135***		
hasInfoBadge1		−0.139***		
isPopularTRUE:hasQABadge1		0.820***		
isPopularTRUE:hasDepMgrBadge1		−0.480***		
isPopularTRUE:hasPopularityBadge1		−0.087***		
isPopularTRUE:hasInfoBadge1		−0.128***		
num_badges			0.302***	
num_badges_sq			−0.049***	
log(ght_age + 1)				0.544***
hasQA				0.134**
hasInfo1				−0.074
hasPopularity1				0.261***
hasOtherBadge				−0.032
time_after:hasQA				0.016***
time_after:hasInfo1				0.003
time_after:hasPopularity1				0.0003
time_after:hasOtherBadge				0.011***
Constant	3.418***	2.733***	4.131***	−0.301
AIC	3470725	3135433	3464672	
Projects	294042	294042	294042	294042
Pseudo-R^2	0.655	0.86	0.66	
R_m^2				0.552
R_c^2				0.932
Note:				*p<0.1; **p<0.05; ***p<0.01

3.1 Effect Plots

We find that the intervention effect correlated with adopting a badge lasts longer if a quality assurance badge is adopted within 15 days of the intervention. In the effect plot below, the slope is smaller for hasQA = 1.


time_after*hasQA effect plot
hasQA = FALSE
hasQA = TRUE
log(downloads_adj + 1)
7.2
7.0
6.8
6.6
6.4
0
2
4
6
8
time_after

4 Signals of Test Suite Quality

Table 3: Test suite size in bytes per project (1-4) at time of study (5) monthly

	Dependent variable:				
	test_bytes		(test_bytes >0)		log(testBytes + 1)
	negative binomial		*logistic*		*linear mixed-effects*
	Base Model	Full Model	Base Model	Full Model	RDD
	(1)	(2)	(3)	(4)	(5)
log(proj_bytes + 1)	0.957***	0.963***	0.183***	0.183***	
log(projBytes)					0.718***
log(stars + 1)	−0.008***	−0.015***	−0.027***	−0.079***	0.020
log(dependents + 1)	0.066***	0.060***	0.616***	0.501***	
log(revisions + 1)	−0.022***	−0.018***	0.044***	0.036***	0.113***
log(age + 1)	0.078***	0.106***	0.053***	0.188***	
hasQABadge		0.168***		2.898***	
hasInfoBadge1		−0.103***		−0.175***	
hasExtraBadge		−0.186***		−0.421***	
hasQABadge:hasInfoBadge1		0.184***		0.317***	
log(downloads + 1)					0.020**
time					0.014***
intervention1					0.087***
time_after_intervention					−0.006*
hasInfo1					0.052
hasQA					0.122
hasOther					−0.068
intervention1:hasInfo1					−0.062***
intervention1:hasQA					0.003
intervention1:hasOther					0.068***
Constant	−1.132***	−1.345***	−1.778***	−2.670***	0.738***
AIC	3628951	3627352	376165	304579	
Projects	293664	293664	293664	293664	2358(x19)
Pseudo-R^2	0.782	0.784	0.044	0.226	
R^2_m					0.575
R^2_c					0.955
Note:					*p<0.1; **p<0.05; ***p<0.01

4.1 Effect Plots

From the effect plots, we see that the adoption of a badge is correlated with an increase in test suite size at the intervention. If a quality assurance badge was adopted within 15 days of the intervention, we see a larger effect than for information badges.

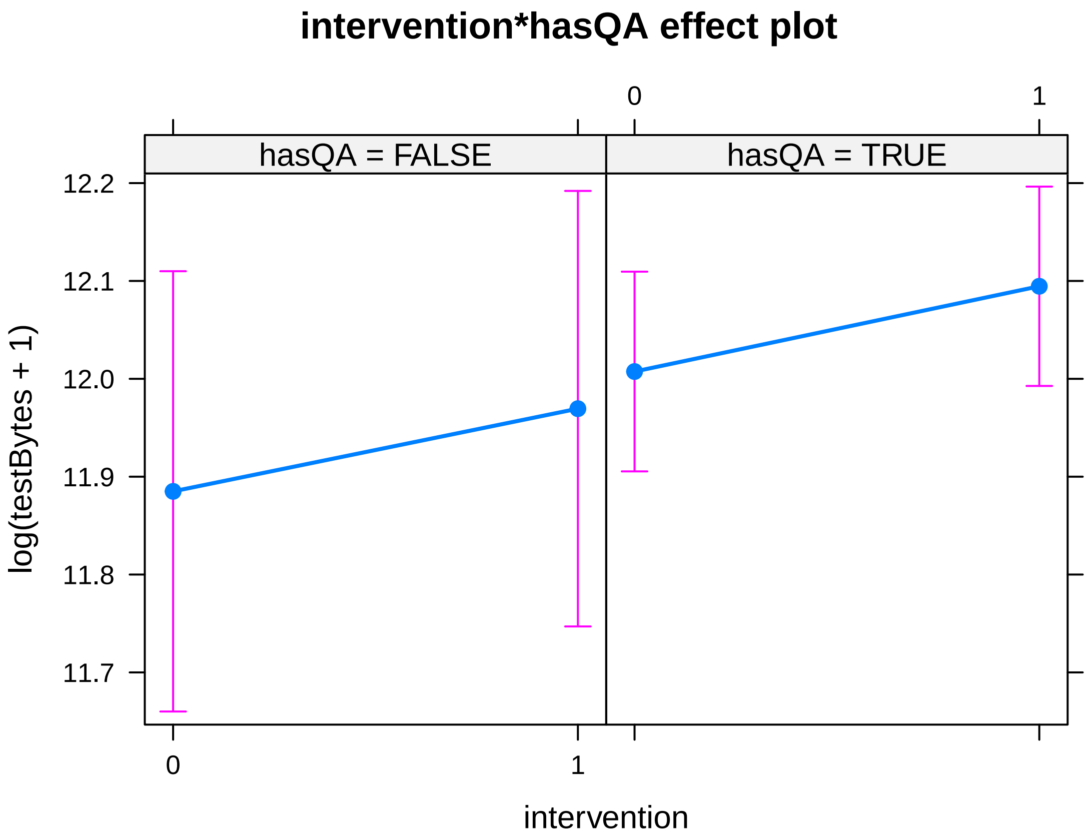
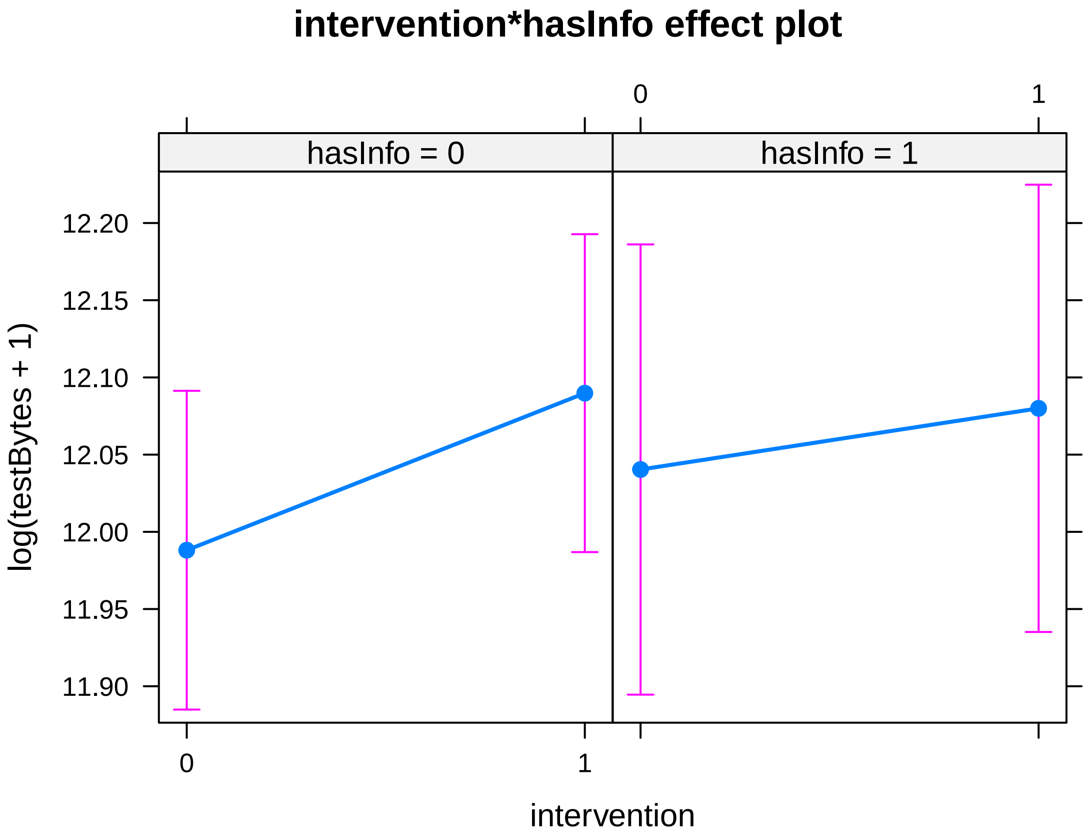

intervention*hasQA effect plot
hasQA = FALSE
hasQA = TRUE
log(testBytes + 1)
12.2
12.1
12.0
11.9
11.8
11.7
0
1
0
1
intervention

Figure 1:


intervention*hasInfo effect plot
hasInfo = 0
hasInfo = 1
log(testBytes + 1)
12.20
12.15
12.10
12.05
12.00
11.95
11.90
0
1
0
1
intervention

5 Signals of Contribution Quality

Table 4: Proportion of pull requests containing tests (1-2) per month at time of study (3) monthly

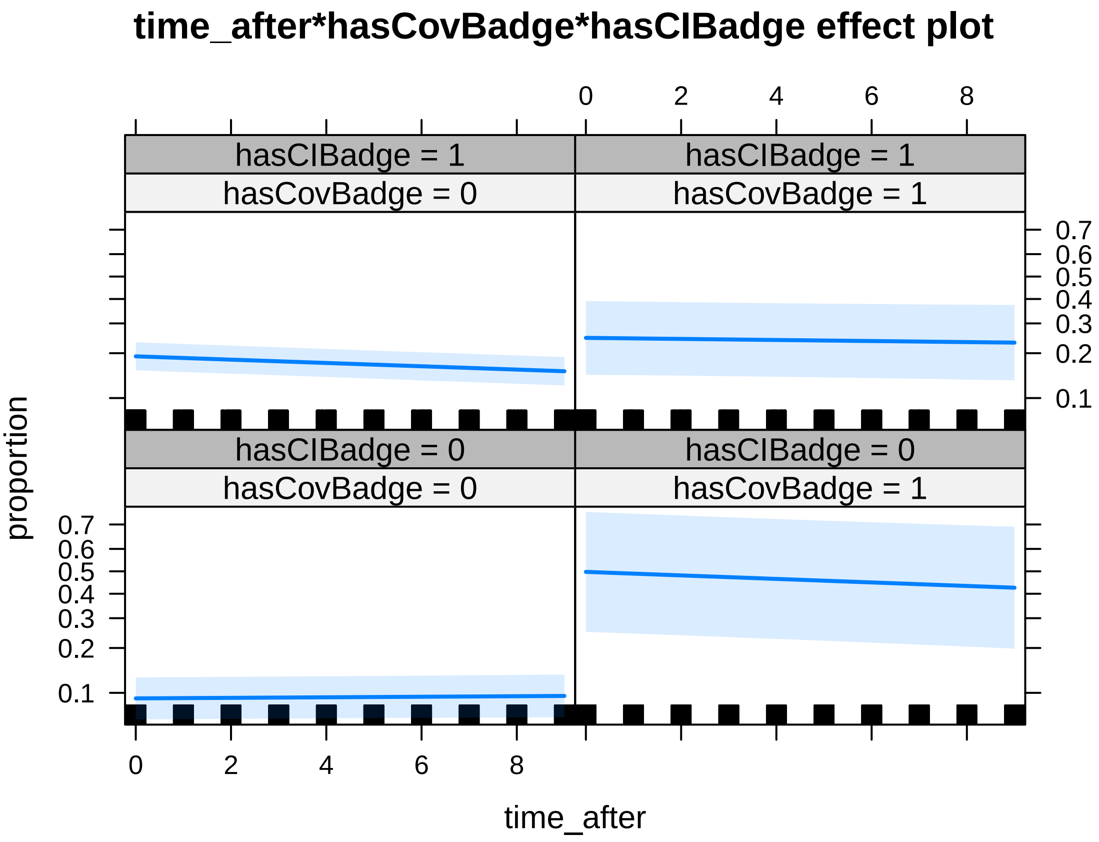
	Dependent variable:		
	proportion		
	logistic		*generalized linear mixed-effects*
	Base Model	Full Model	RDD
	(1)	(2)	(3)
log(contributors + 1)	0.235^{***}	0.229^{***}	0.048
log(revisions + 1)	0.160^{***}	0.159^{***}	0.015
log(stars + 1)	0.073^{***}	0.078^{***}	0.048
log(downloads + 1)			0.086^{***}
time_after			0.005
intervention1			0.282^{*}
hasCovBadge1		0.067^{**}	2.307^{***}
hasCIBadge1		0.216^{***}	0.849^{***}
hasOtherBadge		-0.179^{***}	
time_after:hasCovBadge1			-0.037^{**}
time_after:hasCIBadge1			-0.035^{***}
hasCovBadge1:hasCIBadge1		0.155^{***}	-1.866^{***}
intervention1:hasCovBadge1			-0.038
intervention1:hasCIBadge1			0.012
time_after:hasCovBadge1:hasCIBadge1			0.057^{***}
intervention1:hasCovBadge1:hasCIBadge1			-0.172
Constant	-2.783^{***}	-2.897^{***}	-4.305^{***}
AIC	158565	156445	
Projects	3344	3344	325(x19)
Pseudo-R^2	0.129	0.142	
Note:			*p<0.1; **p<0.05; ***p<0.01

5.1 Effect Plots

We find that the adoption of a quality assurance badge (coverage or continuous integration here) correlates with a small positive change in the proportion of pull requests with tests (second plot below).

Further, projects that adopt both a coverage and continuous integration badge tend to continue to have this increased proportion in the months following the intervention (upper right in the first plot below). All effect plots indicate that the effect decays, but in this case, the decay is slower, or less downward-sloping.

Note that the class (hasCov=1 and hasCI=0) has very few observations.


time_after*hasCovBadge*hasCIBadge effect plot
hasCIBadge = 1
hasCovBadge = 0
hasCIBadge = 1
hasCovBadge = 1
hasCIBadge = 0
hasCovBadge = 0
hasCIBadge = 0
hasCovBadge = 1
proportion
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2
4
6
8
time_after

6 Signals of Support

Since none of the variables of interest (*intervention* or *time_after_intervention*) have significant effects, we do not provide an effect plot for the longitudinal model.

Table 5: Average issue closing latencies (1-2) per month at time of study (3) monthly

	Dependent variable:		
	log(avgLat + 1)		log(avg + 1)
	OLS		*linear mixed-effects*
	Base Model	Full Model	RDD
	(1)	(2)	(3)
time			-0.077^{***}
intervention1			-0.043
log(contributors + 1)	0.184^{***}	0.163^{**}	0.304^{**}
log(age + 1)	-0.235^{**}	-0.141	-0.132
log(downloads + 1)	0.028	0.033	0.005
log(revisions + 1)	-0.027	-0.041	-0.194^{*}
log(num + 1)	0.178^{***}	0.178^{***}	
hasSupportBadge1		0.264^{**}	
hasInfoBadge1		0.122	
time_after			0.062
hasSupport1			-0.799^{***}
hasInfo1			-0.108
hasOther		0.054	-0.050
time_after:hasSupport1			-0.089^{*}
time_after:hasInfo1			-0.109^{*}
time_after:hasOther			0.050
Constant	4.292^{***}	3.777^{***}	7.913^{***}
AIC	2192	2191	
Projects	826	826	76(x19)
Pseudo-R^2	0.0492	0.0592	
R^2_m			0.192
R^2_c			0.448
Note:		*p<0.1; **p<0.05; ***p<0.01	

7 Signals of Security

This section was omitted from the paper due to space restrictions. Note that only 15084 projects have a non-zero security score.

For the additional information model, we see that the presence of a dependency management badge increases the chances of having no security vulnerabilities by 59%. Further, if the package has not been updated recently, it is more likely to have vulnerabilities, which agrees with intuition.

7.1 Security Scores

Using the public vulnerability databases of the *Node Security Project (nsp)* and *Snyk*, we created a security metric as follows. Each vulnerable dependency gets a score of 1, 5, or 20 depending on whether the database classified it as low, medium, or high priority. Unlike the freshness score, we do not average across all projects. Even a single vulnerability can be a significant fault, whereas out-of-date dependencies do not necessarily cause problems.

Historic security is calculated in the same way as freshness: we only consider those entries in the database happening before the date in history.

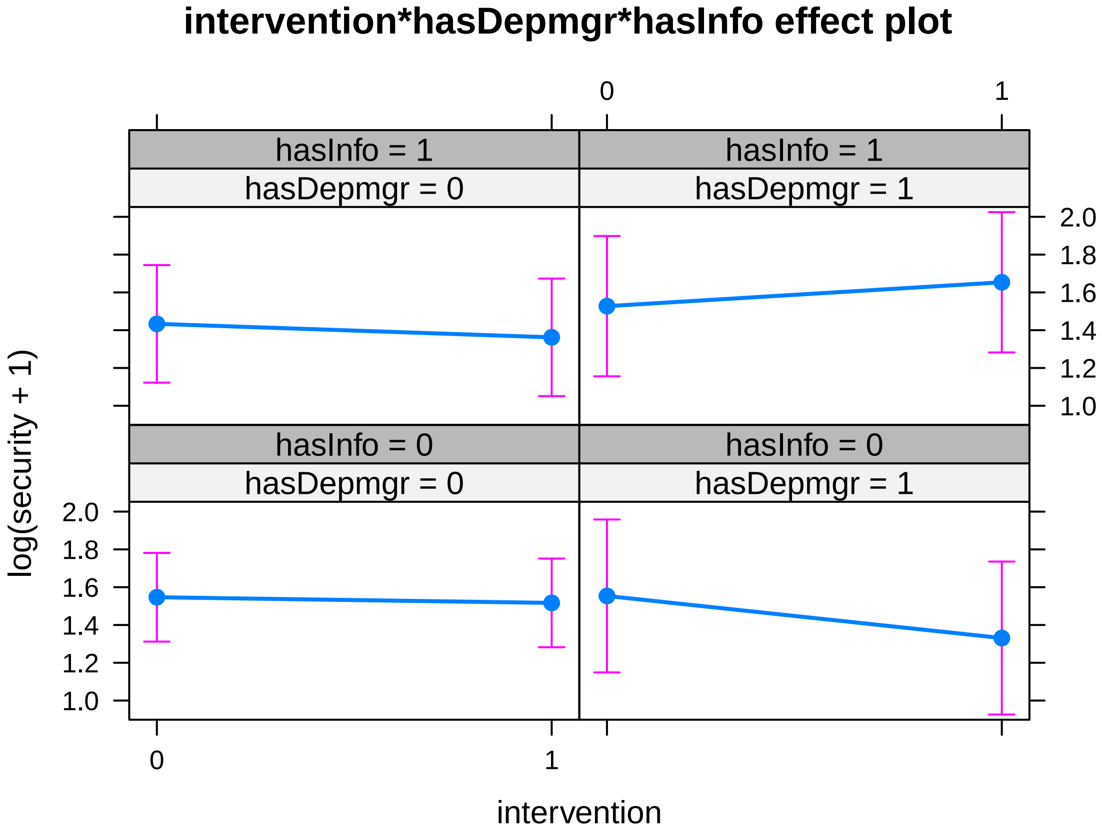
Table 6: Security scores (1-2) at the time of study and (3) monthly

	Dependent variable:		
	(security == 0)		log(security + 1)
	logistic		*linear* *mixed-effects*
	Base Model	Full Model	RDD
	(1)	(2)	(3)
log(dependencies + 1)	-1.190^{***}	-1.204^{***}	0.019
log(dependents + 1)	0.303^{***}	0.269^{***}	0.062^{*}
log(stars + 1)	-0.113^{***}	-0.122^{***}	0.005
log(contributors + 1)	-0.250^{***}	-0.285^{***}	0.112^{***}
log(tsu + 1)	-0.874^{***}	-0.850^{***}	0.181^{***}
hasExtra		0.288^{***}	
hasDepMgrBadge1		0.290^{***}	
hasInfoBadge1		0.109^{***}	
hasDepMgrBadge1:hasInfoBadge1		0.030	
time			-0.015^{*}
time_after_intervention			0.006
intervention1			−0.030
hasDepmgr1			0.007
hasInfo1			−0.113
intervention1:hasDepmgr1			−0.193
intervention1:hasInfo1			−0.042
hasDepmgr1:hasInfo1			0.087
intervention1:hasDepmgr1:hasInfo1			0.391^{**}
Constant	6.979^{***}	6.853^{***}	0.532^{**}
AIC	91618	91092	
Projects	207854	207854	239(x19)
Pseudo-R^2	0.153	0.158	
R^2_m			0.031
R^2_c			0.236

Note: *p<0.1; **p<0.05; ***p<0.01

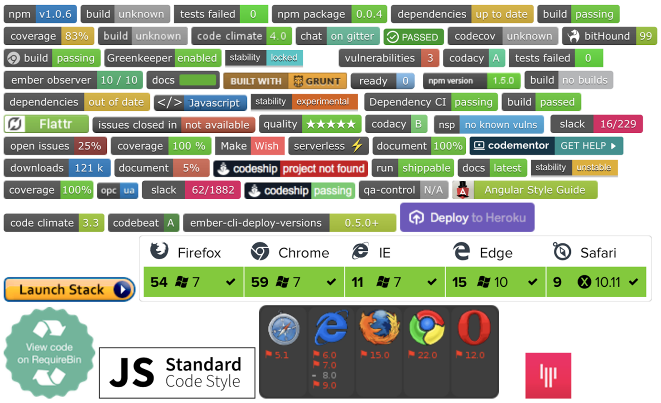
7.2 Effect Plots

From the significant variable *intervention* : *hasDepmgr* : *hasInfo*, we see that there is a significant positive intervention effect for projects with both dependency management and information badges. This is contrary to our expectations, since a lower security score is better. For other intervention terms, the effect is not significant.



8 Survey Questions

This graphic was displayed at the top of the survey.



8.1 Maintainers

8.1.1 Background Information

- How many years of experience do you have in open source software?
- How many npm packages are you a maintainer of?

- How many npm packages have you contributed to?
- Which project will your answers below refer to?

8.1.2 Badges in General

- What influenced your decision to display badges on your project's README file?
  1. They look nice
  2. They display useful information
  3. Advertisements by the badge-offering service itself
  4. Inspired by other projects
  5. Proposed by a collaborator
  6. Common practice in our organization
  7. Other:
- I consider the presence of badges in general to be an indicator of project quality. (five-level Likert scale)

8.1.3 Specific Badges

Here are some badges we found to be popular in the npm community:

- What do you intend to convey to contributors and users through your badges?
  - Please list the most important badges you use and what you are trying to say with them.
    1. Badge 1
    2. Badge 2
    3. Badge 3
- Besides the badges you listed above, are there any other badges you find important? What do they say about your project?
- What effects did you expect each badge would have on your project, contributors, and users?
  - Please refer to the badges that you listed above and explain your answer
    1. Badge 1
    2. Badge 2
    3. Badge 3
    4. Others

8.1.4 Perceived Effects

- Did you notice any differences in your project's practices, contributors, and users after adding badges? Do you attribute these differences to the badges or to other factors?
- Do you have any additional comments regarding badges?

8.2 Contributors

8.2.1 Background Information

- How many years of experience do you have in open source software?
- How many npm packages are you a maintainer of?
- How many npm packages have you contributed to?
- Which project will your answers below refer to?

8.2.2 Contributing to Projects

- When looking for a project to contribute to, I notice badges displayed in the README file. (five-level Likert scale)
- I consider the presence of badges in general to be an indicator of project quality. (five-level Likert scale)
- The presence of badges influenced my decision to contribute to this project. (five-level Likert scale)
- Please explain your previous answer:

8.2.3 Specific Badges

Here are some badges we found to be popular in the npm community:

- What do the badges that you find important tell you about the project?
  - Please name the most important badges that you look for and explain why.
    1. Badge 1
    2. Badge 2
    3. Badge 3
- Are there other badges that you did not list above that you look for? What do they tell you about the project?

8.2.4 Perceived Effects

- Do you tend to notice a difference in a project's practices or quality depending on the presence of badges? Does this apply to specific types of badges?