UNIVERSITÉ Concordia UNIVERSITY

Soen 6611

An Empirical Study on relation between Design-Defects and C&K Metrics Suite

Submitted to: Dr. Rodrigo Morales

Submitted By: **Team 10**

1.Outline

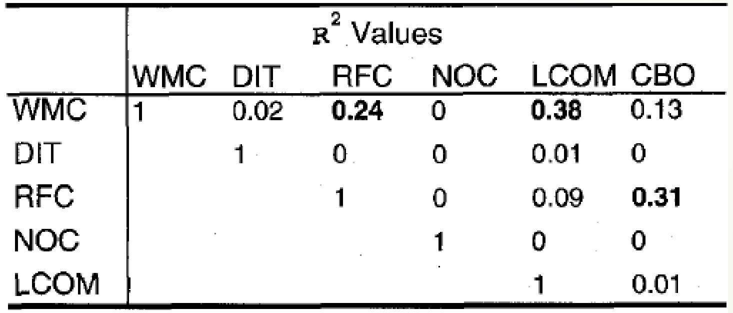
- Introduction.
- Literature Review.
- Research Methods.
- Findings.
- Discussion.
- Conclusion.
- Future Research.
- References.

2.Introduction

- What are design defects ?
- Code Smells: “Smells are certain structures in the code that indicate violation of fundamental design principles and negatively impact design quality.“ [1].
- We quantify the software systems by their attributes, using various metrics.
- In this study we use a well-known Object Oriented metric suite, CK metrics.
- The goal of this research is to study relation between CK metrics and Code Smells.

3. Literature Review

- Basilli in his paper[2], found that linear Pearson's correlations between the studied oo metrics are, in general, very weak. Three coefficients of determination appear somewhat more significant.



	R^2 Values					
	WMC	DIT	RFC	NOC	LCOM	CBO
WMC	1	0.02	**0.24**	0	**0.38**	0.13
DIT		1	0	0	0.01	0
RFC			1	0	0.09	**0.31**
NOC				1	0	0
LCOM					1	0.01

- From their results, five out of the six CK OO metrics appear to be useful to predict class fault-proneness during the high- and low-level design phases of the life-cycle

Literature Review

- They used logistic regression, a standard technique to analyze the relationships between metrics and the fault proneness of classes.
- R.Subramanyam [6] validated the WMC, CBO, and DIT metrics as predictors of the error counts in a class.
- Satwinder Singh [3] in his research paper attempted to find probabilistic threshold values for each metric to predict fault-proneness.

4. Research Questions

- Are Ck metrics useful in detecting Design defects ?
- If yes, are there any specific metrics which are better at doing so ?

5. Research Methods

5.1. Data Collection:

- 3 releases for each of the 3 selected large scale systems developed in Java were analysed.

Mockito	Ant	Wildfly
• 2.25.0	• 1.01.0	• 16.0
• 2.21.1	• 1.05.2	• 12.0
• 2.17.6	• 1.10.5	• 8.0

- Tools used:
  - CK metrics data : Scitool's Understand.
  - Code Smells : Designite Java.
  - Programming : Python, R.

Research Methods

- Code Smells :

Design Smells	Implementation Smells
Abstraction Design smells	Long Method
Encapsulation Design smells	Complex Method
Modularisation Design smells	Long Parameter List
Hierarchy Design smells	Long Identifier
	Long Statement
	Complex Conditional
	Virtual Method Call from Constructor
	Empty Catch Block
	Magic Number
	Duplicate Code
	Missing Default

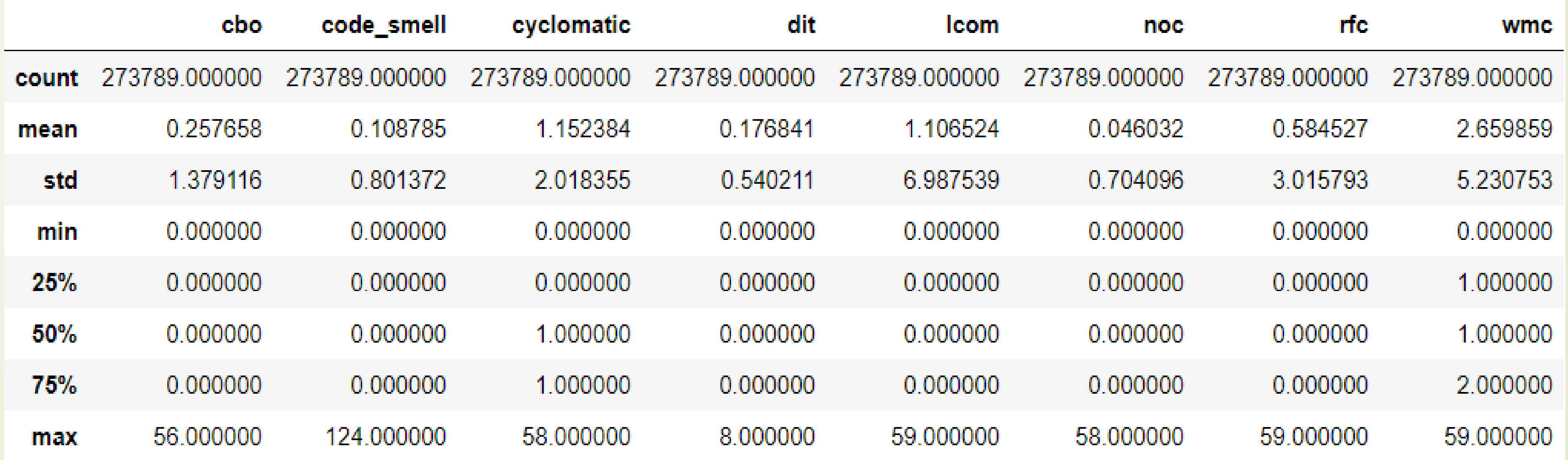
Research Methods

5.2. Method (Quantitative):

- Custom function written for merging the data.
- Descriptive Analysis.
- Feature Engineering.
- Inferential Analysis.

6. Findings

- Descriptive Analysis



	cbo	code_smell	cyclomatic	dit	lcom	noc	rfc	wmc
count	273789.000000	273789.000000	273789.000000	273789.000000	273789.000000	273789.000000	273789.000000	273789.000000
mean	0.257658	0.108785	1.152384	0.176841	1.106524	0.046032	0.584527	2.659859
std	1.379116	0.801372	2.018355	0.540211	6.987539	0.704096	3.015793	5.230753
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
50%	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000
75%	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	2.000000
max	56.000000	124.000000	58.000000	8.000000	59.000000	58.000000	59.000000	59.000000

Figure 6.1

Findings

- Outliers

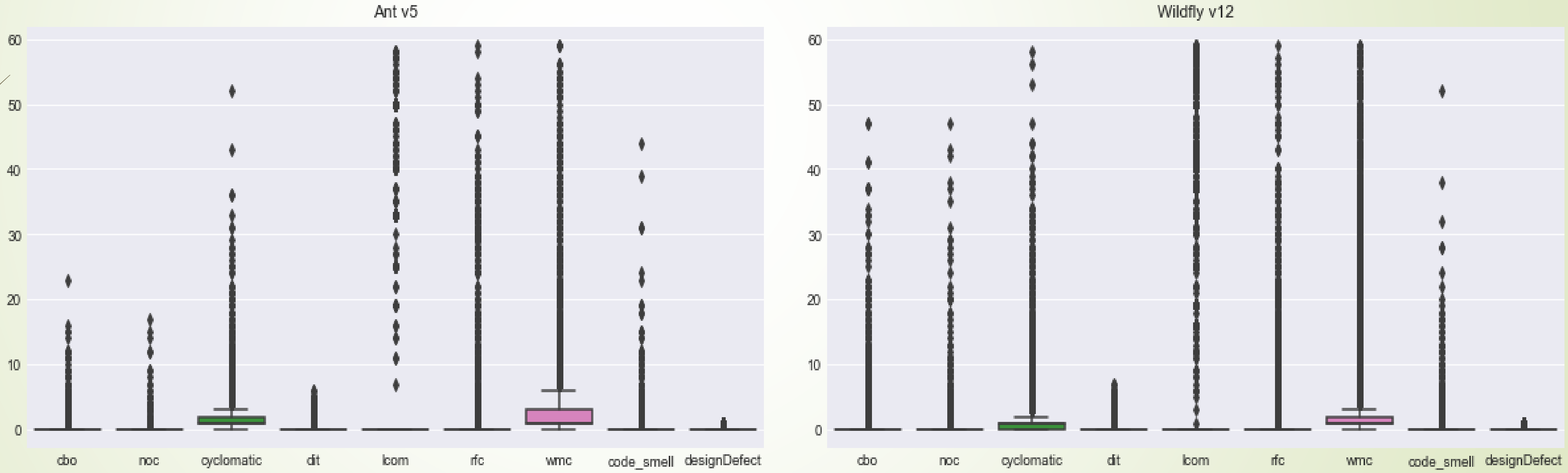
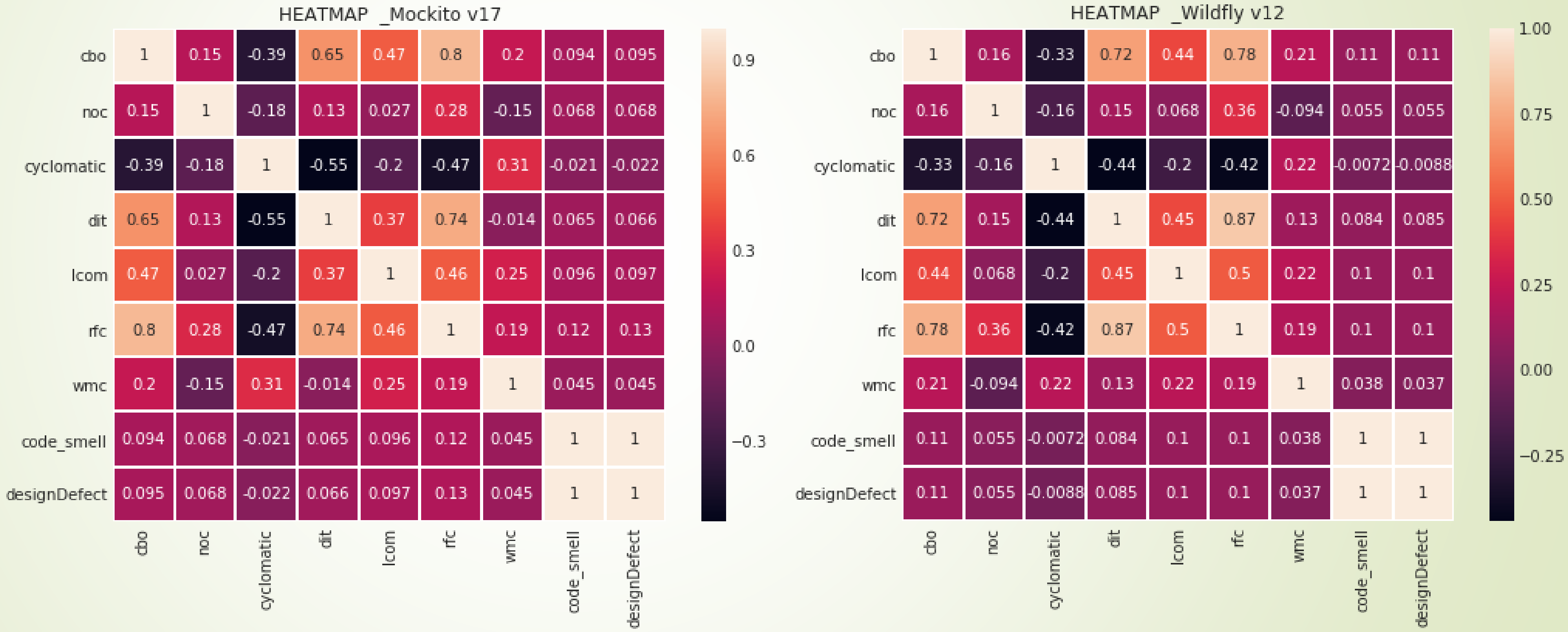


Figure 6.2

Findings

- Heatmaps



HEATMAP _Mockito v17

	cbo	noc	cyclomatic	dit	lcom	rfc	wmc	code_smell	designDefect
cbo	1	0.15	-0.39	0.65	0.47	0.8	0.2	0.094	0.095
noc	0.15	1	-0.18	0.13	0.027	0.28	-0.15	0.068	0.068
cyclomatic	-0.39	-0.18	1	-0.55	-0.2	-0.47	0.31	-0.021	-0.022
dit	0.65	0.13	-0.55	1	0.37	0.74	-0.014	0.065	0.066
lcom	0.47	0.027	-0.2	0.37	1	0.46	0.25	0.096	0.097
rfc	0.8	0.28	-0.47	0.74	0.46	1	0.19	0.12	0.13
wmc	0.2	-0.15	0.31	-0.014	0.25	0.19	1	0.045	0.045
code_smell	0.094	0.068	-0.021	0.065	0.096	0.12	0.045	1	1
designDefect	0.095	0.068	-0.022	0.066	0.097	0.13	0.045	1	1

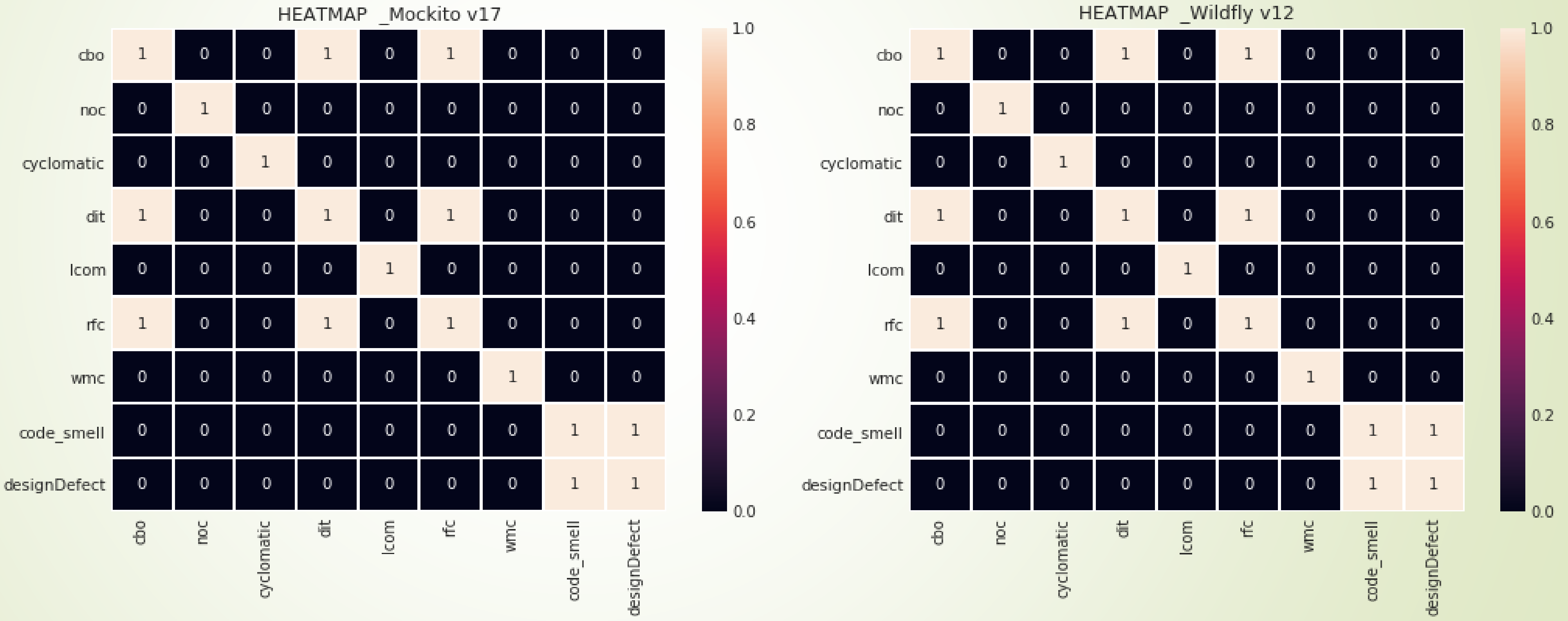
HEATMAP _Wildfly v12

	cbo	noc	cyclomatic	dit	lcom	rfc	wmc	code_smell	designDefect
cbo	1	0.16	-0.33	0.72	0.44	0.78	0.21	0.11	0.11
noc	0.16	1	-0.16	0.15	0.068	0.36	-0.094	0.055	0.055
cyclomatic	-0.33	-0.16	1	-0.44	-0.2	-0.42	0.22	-0.0072	-0.0088
dit	0.72	0.15	-0.44	1	0.45	0.87	0.13	0.084	0.085
lcom	0.44	0.068	-0.2	0.45	1	0.5	0.22	0.1	0.1
rfc	0.78	0.36	-0.42	0.87	0.5	1	0.19	0.1	0.1
wmc	0.21	-0.094	0.22	0.13	0.22	0.19	1	0.038	0.037
code_smell	0.11	0.055	-0.0072	0.084	0.1	0.1	0.038	1	1
designDefect	0.11	0.055	-0.0088	0.085	0.1	0.1	0.037	1	1

Figure 6.3

Findings

- Heatmaps



HEATMAP _Mockito v17

	cbo	noc	cyclomatic	dit	lcom	rfc	wmc	code_smell	designDefect
cbo	1	0	0	1	0	1	0	0	0
noc	0	1	0	0	0	0	0	0	0
cyclomatic	0	0	1	0	0	0	0	0	0
dit	1	0	0	1	0	1	0	0	0
lcom	0	0	0	0	1	0	0	0	0
rfc	1	0	0	1	0	1	0	0	0
wmc	0	0	0	0	0	0	1	0	0
code_smell	0	0	0	0	0	0	0	1	1
designDefect	0	0	0	0	0	0	0	1	1

HEATMAP _Wildfly v12

	cbo	noc	cyclomatic	dit	lcom	rfc	wmc	code_smell	designDefect
cbo	1	0	0	1	0	1	0	0	0
noc	0	1	0	0	0	0	0	0	0
cyclomatic	0	0	1	0	0	0	0	0	0
dit	1	0	0	1	0	1	0	0	0
lcom	0	0	0	0	1	0	0	0	0
rfc	1	0	0	1	0	1	0	0	0
wmc	0	0	0	0	0	0	1	0	0
code_smell	0	0	0	0	0	0	0	1	1
designDefect	0	0	0	0	0	0	0	1	1

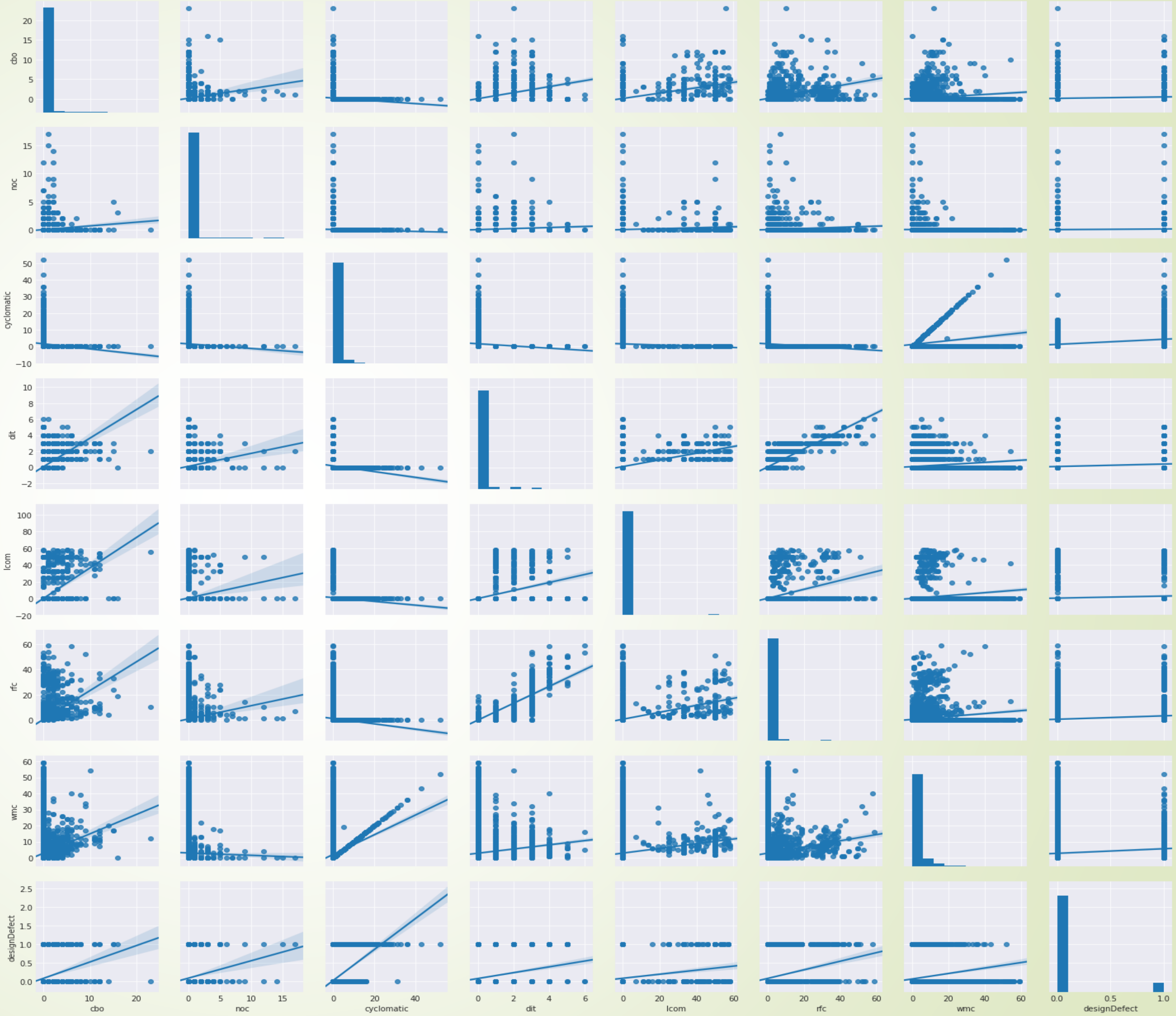
Figure 6.4

Findings

Regression plots

- Presented for Ant.

Figure 6.5

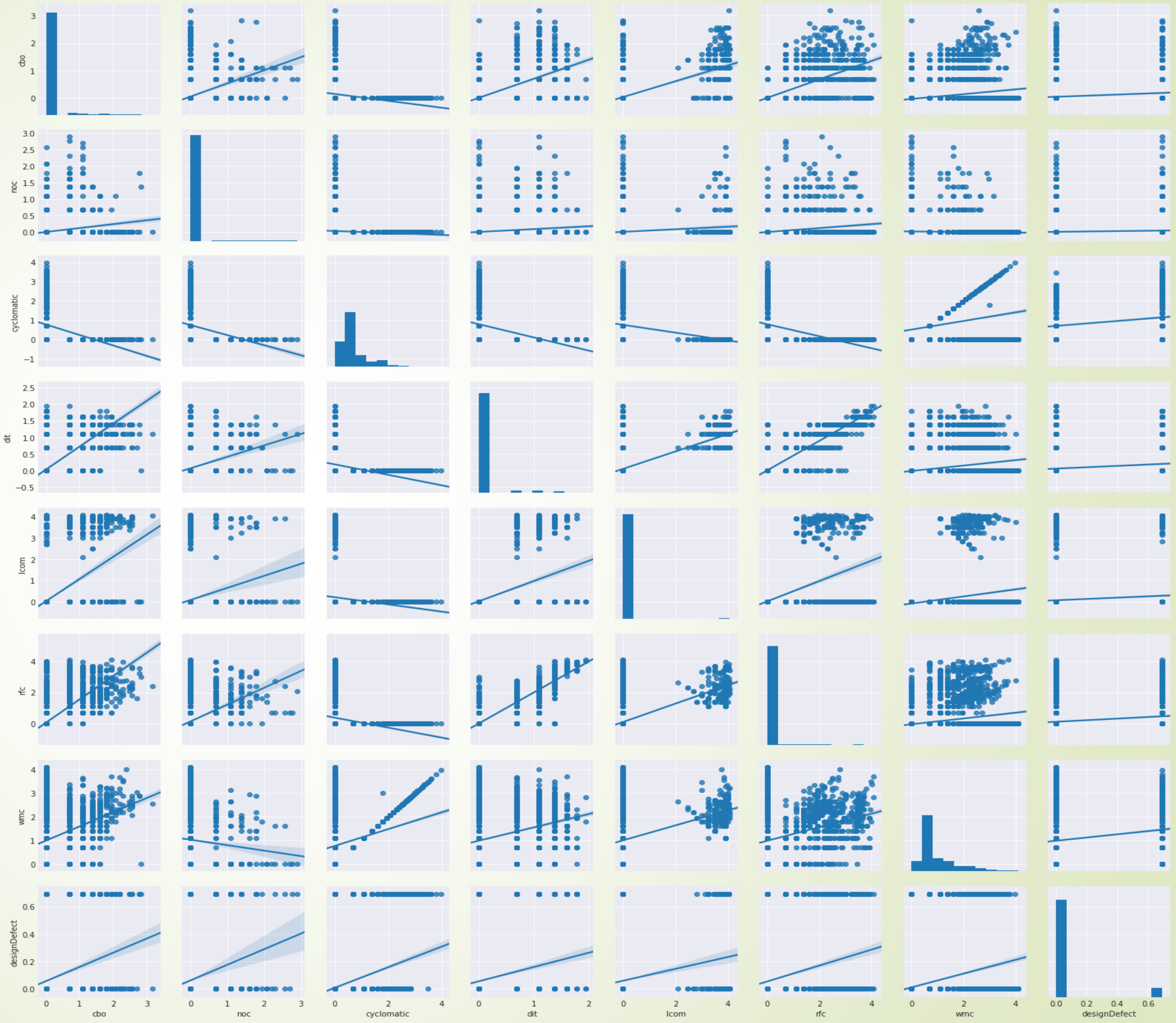


Findings

Log transformed Regression plots.

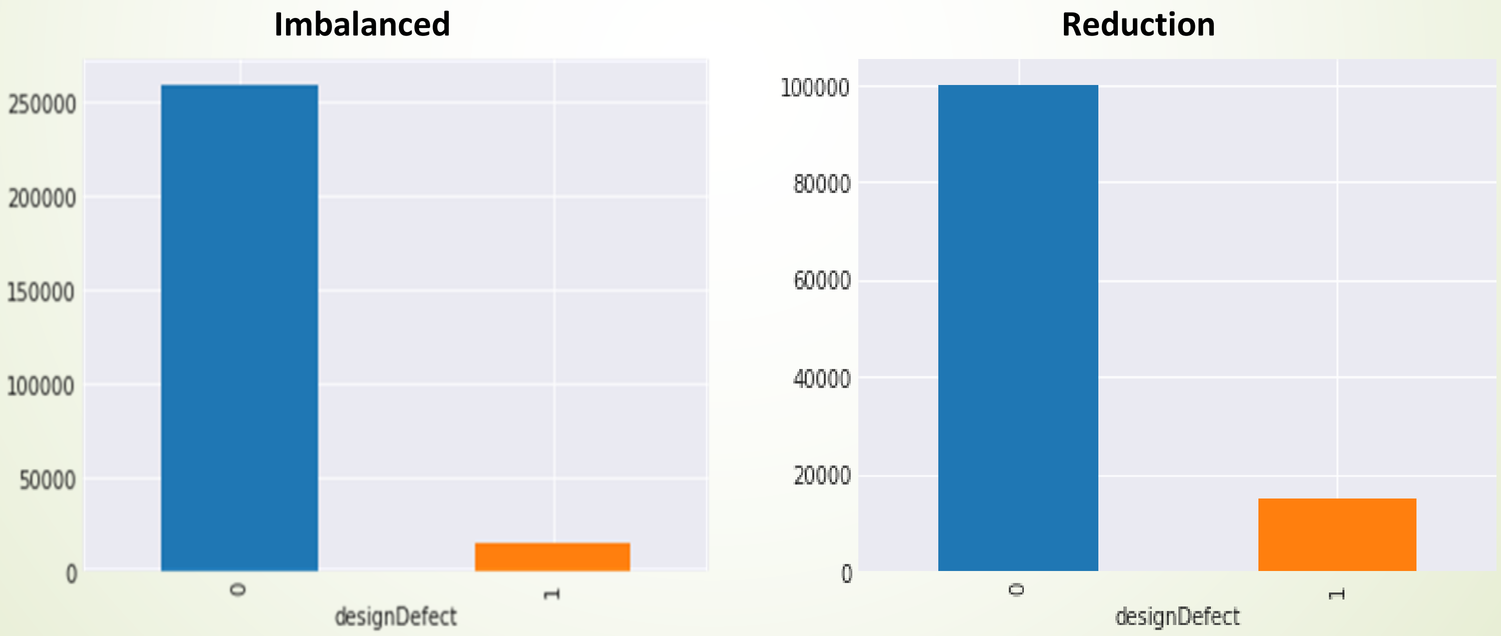
- Presented for Ant.

Figure 6.6



Findings

- A classic problem of imbalanced dataset and reduction.



Imbalanced

Reduction

Figure 6.7

7. Discussion

- In fig 6.1, We can see that the dataset is clean.
- Descriptive analysis shows that the design of the system is good.
- We see a lot of Outliers in [fig 6.2], perform additional analysis to determine if the outliers will degrade the analysis or actually hold some meaning.
- Spearman correlation heatmaps [fig6.3] show high correlation between certain metrics, use this data to eliminate certain variables if necessary.
- We observe that none of the metrics execute linear relationship [fig6.5].
- To identify hidden relation, do a log transformation on the data [fig 6.6].

Discussion

- Model Variations :

	Without outliers	Raw Data (Imbalanced)	Smote Treatment (Balanced)
Accuracy	94.41%	70.04%	59.19%
Sensitivity	99.7%	93.26%	71.44%
Specificity	2%	23.95%	42.86%
P-value	0.8134	3.823e-13	0.0014

8. Conclusion

- From all the analysis we conclude that, there is some correlation between CK metrics and presence of Design defects, but it is not significant for commercial use.
- We accept the null hypothesis, Ck metrics are not useful in detecting Design defects.

Future research

- In the study presented, we have combined the two subdomains of code smells viz. implementation and design smells. An interesting analysis would be to analyse the effect of the two types separately.
- The systems studied are professionally built, under strict design policies and hence tend to be clean. For a clear analysis more of open source systems of mid size or indie applications can be studied.

Summary

- 3 large size systems, with 3 releases for each were analysed all together.
- Datasets are combined and balancing of dependent variable was performed.
- Analysis of outliers showed they were erroneous data, removal improved the model performance.
- There was some but not significant correlation between CK metrics and Design Defects.
- We accepted null hypothesis : Ck metrics are not useful in detecting Design defects.

References

[1] https://en.wikipedia.org/wiki/Code_smell

[2] A Validation of Object-Oriented Design Metrics as Quality Indicators Victor R. Basili, *Fellow, I€€€,* Lionel C. Briand, and Walcelio L. Melo, *Member, IEEE Computer Society1.*

[3] Analysis of CK Metrics to predict Software Fault-Proneness using Bayesian Inference. Heena Kapila , Satwinder singh.

[4] Comments on “A Metrics Suite for Object Oriented Design” Neville I. Churcher and Martin J. Shepperd

[5] DECOR: A Method for the Specification and Detection of Code and Design Smells Naouel Moha, Yann-Gae¨l Gue´he´neuc, Laurence Duchien, and Anne-Franc¸oise Le Meur

[6] Ramanath Subramanyam and M.S. Krishnan,“Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 29, NO. 4, APRIL 2003