

Selecting Quality Data

Tim Menzies (Ph.D.)
Assoc Prof; LCSEE, WVU

tim@menzies.us

304-376-2859

Open issues in data quality (for software engineering)

- Conclusion Instability
- Locality
- Best way to find significant data
 - How to import data from other sites
 - Relevancy filtering
 - Feature selection

Background

The PROMISE Experiment

- 2005, ... 2011
 - Repeatable, improvable, maybe even refutable experiments in software engineering
 - Put up or shut up
 - [Menzies07]
- If you publish a paper, it is strongly encouraged that you also offer the data on which the conclusion was made
- 130 data sets:
 - Defect prediction
 - Effort estimation
 - Model-based reasoning
 - Text mining
 - etc

<http://promisedata.org/data>

The image displays two side-by-side screenshots of the PROMISE website. The left screenshot shows the main "Data Repository" page at <http://promisedata.org/?cat=11>. It features a logo with a blue question mark and orange line graph, followed by the text "PROMISE The 2010 International Conference on Predictive Models in Software Engineering". A navigation menu includes Home, About, Data, Papers, People, PROMISE '06, PROMISE '07, PROMISE '08, PROMISE '09, and PROMISE '10. A search bar is present. A sidebar titled "News" lists links to All presentations online, Pictures – PROMISE 2008, Promise Presentations Online, Workshop Updates, and 2008 Workshop Updated. Below the menu is a bar chart showing the number of datasets in various categories: DefectPrediction (57), EffortEstimation (18), General (9), Model-BasedSE (7), and TextMining (9). A note states, "In 2006, the repository held 23 data sets." Another note says, "In 2008, at last update, the repository holds 100 data sets in the following areas:" followed by the same category list. A download bar at the bottom shows files 10.1.1.79.8022.pdf, usatravelagreement.zip, and thesis (8).pdf. The right screenshot shows the "Important Dates" page for PROMISE 2010 at <http://promisedata.org/2010/dates>. It features the same logo and navigation menu. The "Important Dates" section lists submission and notification deadlines. A sidebar on the right shows a photo of a large fountain at night.

Welcome to the Promise Data Repository!

In 2006, the repository held 23 data sets.

In 2008, at last update, the repository holds 100 data sets in the following areas:

- Defect Prediction (57)
- Effort Estimation (18)
- General (9)
- Model-based SE (7)
- Text Mining (9)

10.1.1.79.8022.pdf usatravelagreement.zip thesis (8).pdf Show all down...

PROMISE data sets ... AND COUNTING

108
107
106

News

- All presentations online
- Pictures – PROMISE 2008
- Promise Presentations Online
- Workshop Updates
- 2008 Workshop Updated

PROMISE 2010

The 6th International Conference on Predictive Models in Software Engineering
Co-located with ICSM'10, at Timisoara, Romania
Sept 12-13, 2010

[Home](#) | [Program](#) | [Venue](#) | [Registration](#) | [Call for papers](#) | [Dates](#) | [Committees](#) | [Keynotes](#) | [Submit](#) | [Promote](#)

Important Dates

Papers:

- Abstract Submission Deadline: May 28, 2010
- Paper Submission Deadline: June 4, 2010
- Student Symposium Submission Deadline: June 4, 2010
- Notification of Results: July 9th, 2010
- Camera Ready Copy Submission Deadline: July 23th, 2010

Registration:

- Early registration deadline: August 16, 2010

Conference:

- Main conference: September 12 and 13, 2010
- Student symposium: September 13, afternoon/evening.
(Note: symposium attendees must register for the main conference.)

Special issue:

- Invitations to submit: October 1, 2010
- Paper submission deadline: Dec 31, 2010
- Notifications of first round reviewing: March 31, 2011
- Publication: late 2011 (planned)

Hotel. PROMISE: 2005, 2006, 2007, 2008, 2009 | Contact: mail (at) promisedata.org

Repository + Annual conference + Journal special issues

See you there?

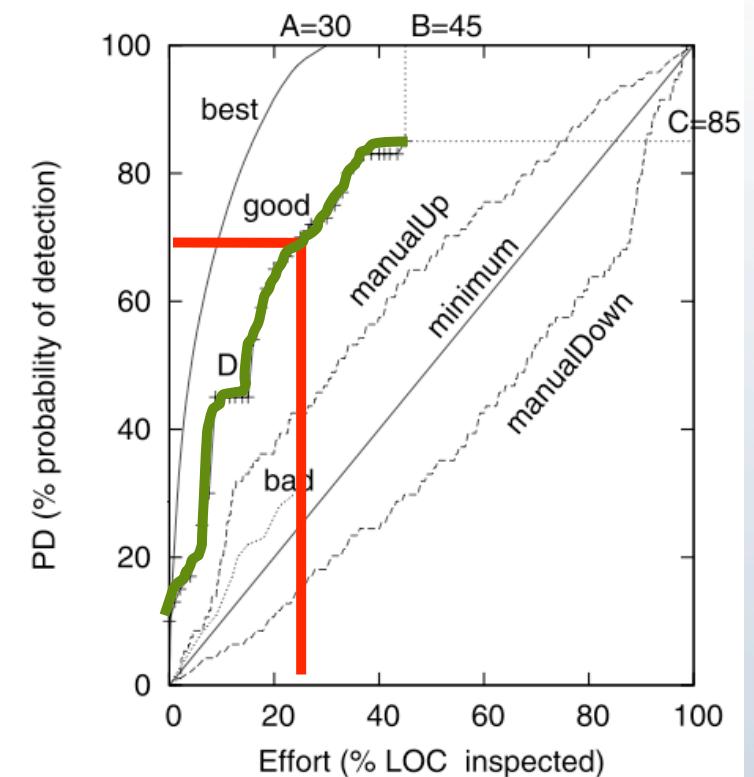
Relevance to quality

Other repos	Our repo
<ul style="list-style-type: none">● Extensive software support for the data storage<ul style="list-style-type: none">● E.g. [Gentleman07, Leischo2]● Elaborate design of research questions:<ul style="list-style-type: none">● E.g. CeBASE [Basilio02]● None of those SE repos are still on-line	<ul style="list-style-type: none">● No restrictions on research questions● Send us something,<ul style="list-style-type: none">● We'll place it on-line● Let many groups analyzing that data<ul style="list-style-type: none">● Extensive list of conclusions● Still an active, on-going research initiative.

Some results:

Learning to Organize Testing

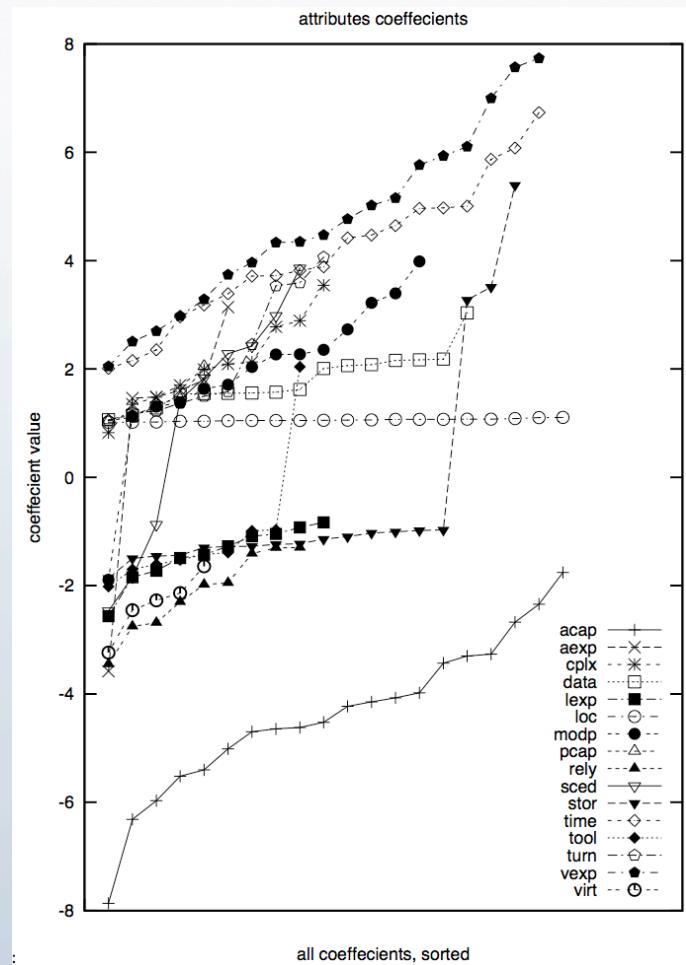
- Inspect X% of the code, find Y% of the bugs
 - But which X%?
 - Ask a data miner
- Typical results:
 - [Ostrand10]: X= 20% of files , Y= 75 to 93%
 - [Tosun10] : X= 25% . Y= 88%
 - [Menzies10a] X= 25%, Y= 70%
 - Or any other point you like



Conclusion Instability

20 * 90% sub-samples

- Linear regression on COCOMO data
- Wild variance in the learned model
 - Specifically, in the variable coefficients



Locality

[Brady10]: case-based reasoning to tame instability

- Seek best changes to a project
 - Tames conclusion instability, but conclusions project specific
- Beware the snake oil salesman telling you that “it worked there, it will work here”
- Best to assess policies w.r.t. local data
- (Are you collecting local data?)

cases	query	acap	apex	ltx	ltx	plex	pmat	pmat	sced	sced	stor	time	tool	# of Changes
		3	3	3	4	3	3	4	2	3	3	3	3	
nasa93	ground					100%	55%				85%			3
nasa93	flight					95%	70%				100%			3
nasa93	osp	95%	90%										100%	3
nasa93	osp2				100%			80%	85%					3
coc81	flight						60%					65%		2
coc81	osp2			55%	55%		65%			100%				4
coc81	ground					80%					100%			2
coc81	osp					65%			65%					2
Overall:		12%	11%	7%	19%	24%	49%	10%	11%	21%	23%	21%	13%	

How to find significant data

Some data is better than others

- **Relevancy filtering**
 - Given a test set T
 - Relevant = the k-th nearest neighbors in the train set
 - Only train on this “relevant” set
- **Data subset selection**
 - Reveal the core content



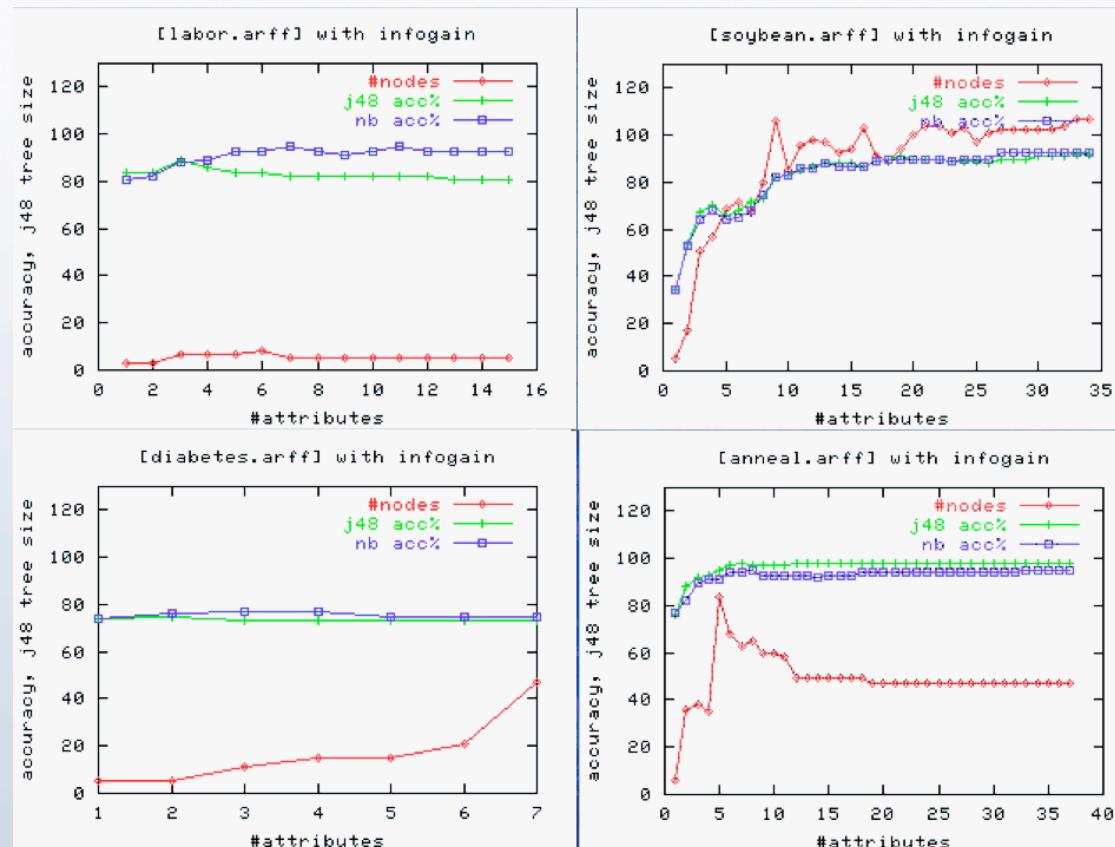
Relevancy filtering:

How to use other people's data

- Learning
 - (a) software development estimation or
 - (b) software defect predictors
- Suppose you don't have local data
 - Find relevant data from other companies
- Works as well as if you had local data
 - For defect prediction [Turhan09]
 - For effort estimation [Menzies10]
- An experiment with N contractors
 - Imported:
 - train on them, test on me
 - Local:
 - train on one, test of one
 - Relevant:
 - find nearest neighbors in them, train on just those
 - test on me
- P_d = detection, pf = false alarm
 - Imported $P_d, pf = 94 (!), 68$
 - Local: $P_d, pf = 75, 29$
 - Relevant: $P_d, pf = 69, 27$

Feature selection: strip away spurious details

X-axis
sorted by entropy
 $\text{sum}(-p \cdot \log(p))$



Useful = 6/16, 10/35, 1/7, 5/45 = 38%, 26%, 14%, 11%

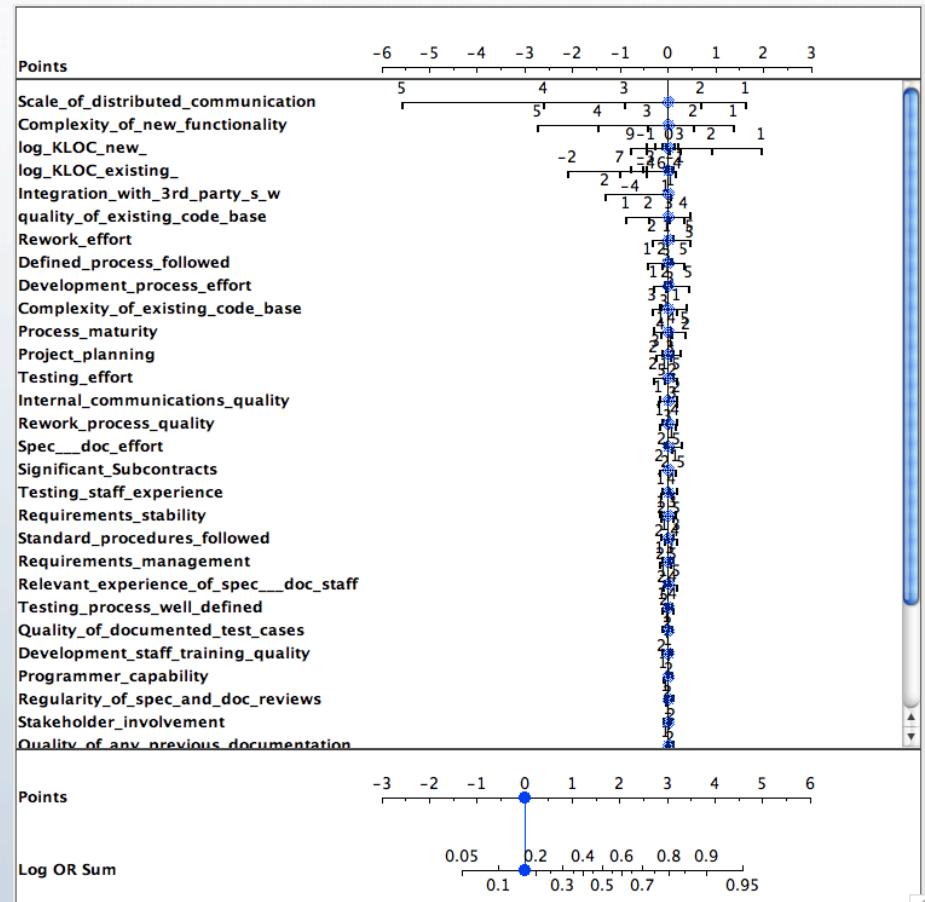
Select features with Nomograms

- Nomograms [Mozinao4]:

- Discretize every feature.
- For all pairs of target / other classes of size C_1, C_2 count frequency of range N_1, N_2 in each class
- $\text{Log}(\text{odds ratio}) = \log((N_1/C_1) / (N_2/C_2))$
- If positive, then more frequent in target

- Data from Fenton's Bayes Nets

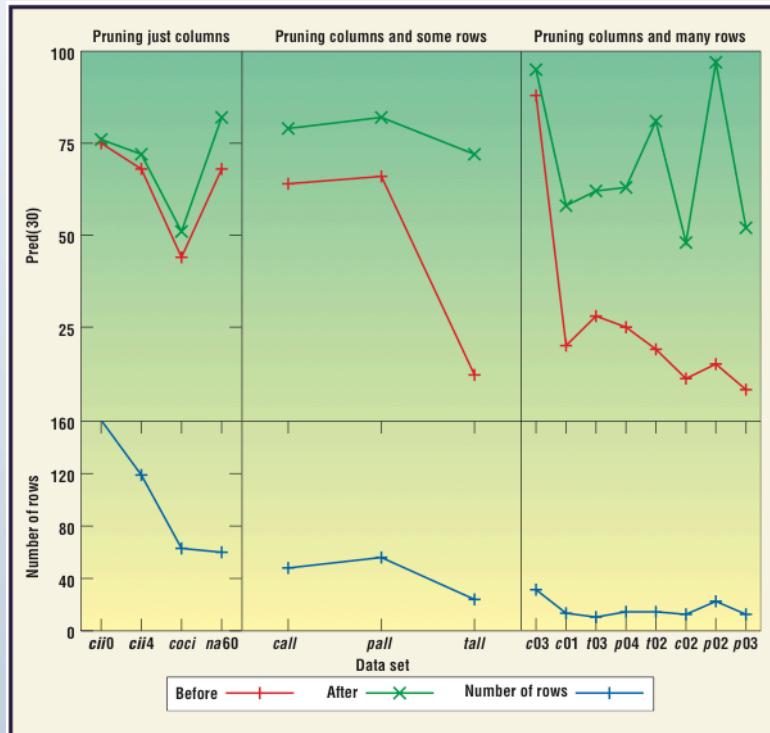
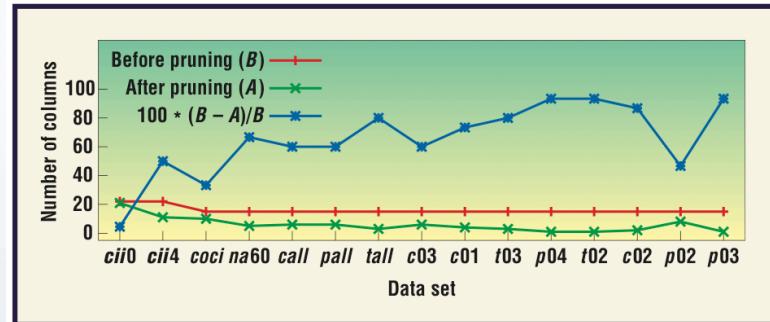
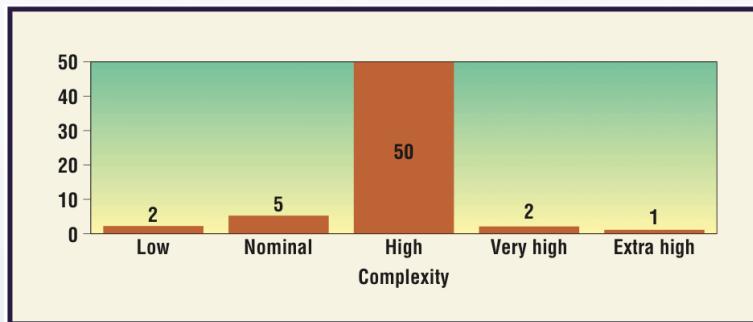
- [Fentono8]
- Target class: worst defects
- Useful subset of attributes: 7%
- And of those attributes,
 - Only a few ranges matter



Caution

- **Simple entropy, nomograms,**
 - running in linear time
 - not the best feature selector
- **Best = WRAPPER**
 - A search through all subsets of F features
 - Warning: may be impractical for large data sets
- **Mixed strategies**
 - Send out the scouts (e.g. nomograms) to quickly prune
 - Before firing the big guns (WRAPPER)

Example 1: Select columns with WRAPPER for effort estimation



- Reference: [Menzies05]
- Useful = 65%

Example 2: Select columns with WRAPPER for defect prediction

- 10-way cross-val, LSR, on [Bno8]

Defects =

82.2602 * S₁=L,M,VH +
158.6082 * S₁=M,VH +
249.407 * S₁=VH +
41.0281 * S₂=L,H +
68.9153 * S₂=H +
151.9207 * S₃=M,H +
125.4786 * S₃=H +
257.8698 * S₄=H,M,VL +
108.1679 * S₄=VL +
134.9064 * S₅=L,M +
-385.7142 * S₆=H,M,VH +
115.5933 * S₆=VH +
-178.9595 * S₇=H,L,M,VL +
...

[50 lines deleted]

R² = 0.45 (mean on x-val)

- WRAPPER:

- Search for attributes that matter, 10 times on 90% of the data
- Only build model from attributes selected at least 50% of the time.

Defects =

876.3379 * S₇=VL +
-292.9474 * D₃=L,M +
483.6206 * P₅=M +
5.5113 * KLoC +
95.4278

Uses 4/53 = 8% of the data

R²=0.98 (mean on x-val)

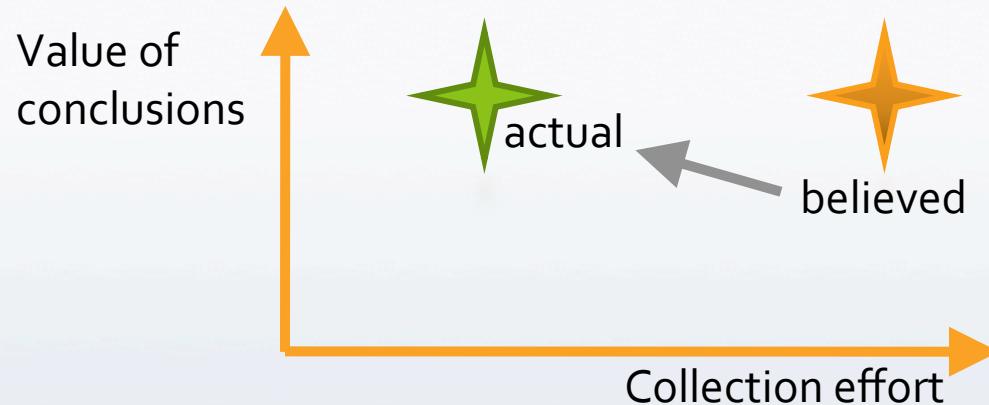
Implications for data quality planning

- A warning: it may be a waste of effort to:
 - Expend great effort to precisely define all possible data points,
 - Then spend much time collecting data according to those definitions.
- Rather, more effective to:
 1. Quickly implement a draft definition/collection plan.
 2. As soon as any data is available, apply feature selection.
 3. Only elaborate data that feature selection reports are significant.



Summary

Explaining the surprising conclusions of PROMISE



- Given the local conditions of a particular project,
 - Only a small number of features are insignificant:
 - E.g. 38%, 26%, 14%, 11%, 63%, 7%, 7%
- So in a large pool of messy data
 - They may exist a subset that is still useful for prediction
- So real-world data is good, providing you get enough of it, and throw most of it away before you learn

Open issues in data quality (for software engineering)

- Conclusion Instability
- Locality
- Best way to find significant data
 - How to import data from other sites
 - Relevancy filtering
 - Feature selection

References

- [Basilio02] Victor Basili, Roseanne Tesoriero, Patricia Costa, Mikael Lindvall, Ioana Rus, Forrest Shull, and Marvin Zelkowitz. Building an experience base for software engineering: A report on the first cebase workshop. In in ProFES (Product Focused Software Process Improvement), pages 110–125, 2001.
- [Bno08] <http://icde.googlecode.com/svn/trunk/share/data/arff/bn.arff>
- [Brady10] Adam Brady and Tim Menzies. Case-Based Reasoning vs. Parametric Models for Software Quality Optimization, PROMISE'10
- [Fenton08] Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction Norman Fenton, Martin Neil, William Marsh, Peter Hearty, Lukasz Radlinski and Paul Krause., PROMISE 2008
- [Gentleman07] R. Gentleman and D. Temple Lang. Statistical analyses and reproducible research. *Journal of Computational and Graphical Statistics*, 16(1):1–23, March 2007.
- [Leisch02] F. Leisch. SWeave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz, editors, *Proceedings in Computational Statistics* (2002), pages 575–580. Physica 2002.
- [Menzies05] Tim Menzies with Scott Chen and Dan Por and Barry Boehm, Finding the Right Data for Software Cost Modeling , IEEE Software Nov/Dec 2005
- [Menzies06] T. Menzies, K. Lum, and J. Hihn. The deviance problem in effort estimation. In PROMISE, 2006,
- [Menzies07] T. Menzies with G. Boetticher, and T. Ostrand. PROMISE Repository of empirical software engineering data. West Virginia University, Department of Computer Science, 2007. <http://promisedata.org/>.
- [Menzies10] Ekrem Kocaguneli, Gregory Gay, Tim Menzies, Ye Yang, Jacky W. Keung: When to use data from other projects for effort estimation. ASE 2010: 321-324
- [Menzies10a] Tim Menzies, Zach Milton, Burak Turhan, Bojan Cukic et al. Defect prediction from static code features: current results, limitations, new approaches Automated Software Engineering, Dec 2010
- [Mozina04] Martin Mozina,, Janez Demsar, Michael Kattan, and Blazzupan, Nomograms for Visualization of Naïve Bayesian Classifier
- [Ostrand10] Programmer-based Fault Prediction Thomas J. Ostrand, Elaine J. Weyuker, Robert M. Bell. PROMISE'10
- [Tosun10] Tosun, A., Bener, A.: AI-based software defect predictors: Applications and benefits. In: IAAI'10 (2010)
- [Turhan09] Turhan, B., Menzies, T., Bener, A., Distefano, J.: On the relative value of cross-company and within-company data for defect prediction. *Empir. Softw. Eng.* 68(2), 278–290 (2009). Available from <http://menzies.us/pdf/o8ccvc.pdf>



**Questions?
Comments?**