Web掲載資料では
一部スライドをカットしております

今さら聞けないソフトウェア品質データ分析の基本

東洋大学経営学部　野中　誠

2014年9月2日

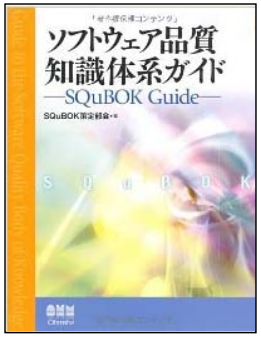
情報処理学会ソフトウェア工学研究会
ソフトウェアエンジニアリングシンポジウム (SES2014)
チュートリアル

東洋大学

- **所属**
  - 東洋大学 経営学部 経営学科　教授
- **背景**
  - 工業経営／経営システム工学，ソフトウェア工学，品質マネジメント
- **主な学外活動**
  - IPA/SEC 高信頼性定量化部会主査(『ソフトウェア開発データ白書』など)
  - 日本科学技術連盟 SQiPソフトウェア品質委員会 運営委員長
  - 日本SPIコンソーシアム 外部理事
  - 国立情報学研究所 特任研究員（トップエスイー講座，メトリクス講義担当）

野 中　　誠

- **主な著書**
  - 野中・小池・小室著『データ指向のソフトウェア品質マネジメント』日科技連出版社（2012）
    2013年度日経品質管理文献賞受賞
  - 野中・鷲崎訳『演習で学ぶ ソフトウェアメトリクスの基礎』日経BP社（2009）
  - SQuBOK策定部会編著『ソフトウェア品質知識体系ガイドーSQuBOK　Guideー』オーム社（2007）
- **研究・教育**
  - ソフトウェア品質マネジメント(メトリクスを中心に)
  - 情報システムと経営の関わり



講演概要

ソフトウェア品質と生産性の組織的な向上には，技術による解決だけでなく，
プロジェクト実績データを調べて実態を把握し，プロセスの弱みを明らかにし，課題を共有し，
目標を定めて継続的に改善していくといった組織的な取り組みが必要です。

また，最近は，見積りの妥当性やプロジェクトの品質状況などについて，
顧客などに実績データをもとに説明する必要性も高まっています。

しかし，そもそも実績データを記録していなかったり，データの可視化方法が分からなかったり，
分析スキルが足りなかったり，根拠のない定型フォーマットが長年にわたって使われ続けて
いたりします。

このチュートリアルでは，そうした疑問や課題の解決方法について，
『データ指向のソフトウェア品質マネジメント』の著者であり，
IPA/SEC『ソフトウェア開発データ白書』の検討部会主査を務める講師が，
情報処理学会ソフトウェア工学研究会の知見を活かしながら解説します。

品質データ分析の基礎知識に自信のない方にもご理解いただけるよう平易に解説します。

※ 高度な統計解析の話，交差検証，機械学習などの話はほとんどいたしません

品質データ分析は改善の推進力

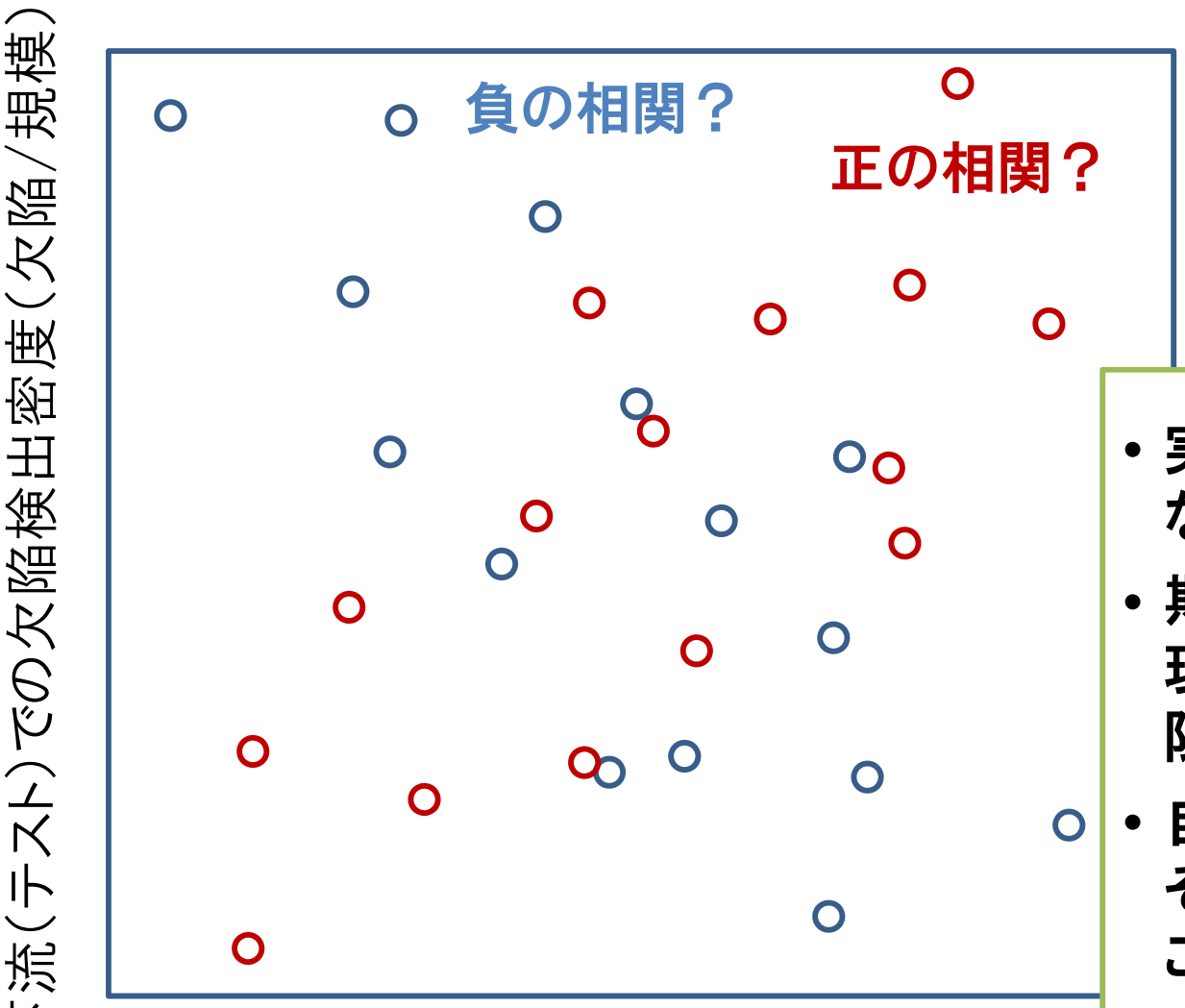
- **品質管理の基本原則「事実に基づく管理(fact control)」はソフトウェアにおいても追求すべき基本原則である**

- **しかし，ソフトウェアにおける「事実に基づく管理」は課題が多い**
  - 測定プロセスが**人に大きく依存**しており，データの量と質に問題がある
  - 多くのメトリクスは二面性があり，単一では品質の良し悪しを判断できない
  - コード行数に対する批判など，古典的な問題が解決されていない

- **課題を踏まえながらも「事実に基づく管理」を推進することは，継続的な品質改善に向けた有効な施策である**
  - データが示す**客観的事実の力**を活用し，実態を把握する
  - データを手がかりに原因を特定し，対策を講じ，その効果を確認する

  そして，
  - 再発防止・未然防止に向けてプロセスを改善し，失敗コスト比率を下げる
  - 顧客満足度の向上につながる活動に注力し，「品質」の向上を図る

「データが示す客観的事実の力」の例：上流欠陥摘出と下流欠陥検出は正・負の相関のどちらか？



- 実際には「正の相関」になることも（その方が多いかも）
- 期待している因果関係と現実の状況が一致しているとは限らない
- 自組織の状況をデータで把握し，その結果を組織内で共有することが改善への第一歩

品質データ測定・分析のアクションプラン

- **上位レベルの目的を定める**
  - **顧客満足**のために，製品・サービスとして重点管理すべき**品質特性**は何か
    例)ISO/IEC 25010:2011 システム／ソフトウェア製品品質モデル
    機能適合性，性能効率性，互換性，使用性，信頼性，セキュリティ，保守性，移植性
  - **事業戦略および製品・サービス戦略**の上で重視すべき KPI は何か
- **品質データ測定・分析の直接の目的を定める**
  - 製品・サービスに対する顧客の評価(またはその代替指標)を把握したい
  - 開発中のプロジェクトの品質状況を把握したい
  - 品質計画のための標準値を定めたい
  - 開発工数の見積りの妥当性を評価したい　など
- **目的達成に必要なメトリクスとその可視化・分析方法を検討する**
  - 製品・サービスのリリース後の状況を把握する指標として何を測定すればよいか
  - プロセス指標・プロダクト指標として何を測定すればよいか
  - どのような統計量／グラフを用いれば必要な情報が得られるか
  - どのような分析手法(統計解析手法など)を適用すれば解が得られるか
- **可視化・分析によって得られた情報に基づきアクションをとる**
  - データを通じて把握できた事実に基づき，どのような施策につなげるか

参考：品質／ソフトウェア品質の基本概念
… 品質の良し悪しは顧客満足で決まる

- **日本的品質管理で培われてきた品質の考え方：徹底的な消費者指向**
  - 「品質の良し悪しは**顧客の満足**で決まる」「提供する側の評価で決まるものではない」
    ⇒ 品質論の原点は「相対的認識」であり，他人に認められてはじめて評価が定まる
- **ニーズを満たす度合いで評価する**
  - 「明示された条件で使用する場合に，ソフトウェア製品が**明示的ニーズ**及び**暗黙のニーズ**を満たす度合い」（ISO/IEC 25040:2011）
  - 「**ニーズ**に関わる対象の特徴の全体像」飯塚 悦功

  ※ ACSIモデルでは「知覚品質」と「知覚価値」を分け，価格に対する効用を知覚価値に含めている
- **立場によってニーズは異なる**
  - 「品質は**誰かにとっての価値**である」G. M. Weinberg
    ⇒ ユーザやシステム管理者など，立場によって価値は異なる（品質の相対性）
- **信頼性と機能適合性に絞って捉える**
  - 「ソフトウェアの停止や，不正確な出力結果の原因となる**欠陥**が残存している度合い」C. Jones

顧客満足を原点として，ニーズを満たす度合いで品質を捉える

まずは，リリース後の状況に関わる品質データについて把握する：リリース後欠陥密度のベンチマークデータ

日・米・欧・印のパフォーマンス比較（Cusumano *et. al.*, 2003）

項目	インド	日本	米国	欧州他	合計・平均
調査対象のプロジェクト数（件）	24	27	31	22	104（合計）
開発規模の中央値（KLOC）	209	469	270	436	374
リリース後欠陥密度の中央値（KLOC当たり）	0.263	**0.020**	0.400	0.225	0.150

IPA/SEC『ソフトウェア開発データ白書2012-2013』

項目	新規開発 (N = 520)	改良開発 (N = 349)
リリース後不具合密度の中央値（KLOC当たり）	0.016	0.000
リリース後不具合密度の平均値（KLOC当たり）	0.106	0.123

- この指標を組織横断的に比較するのは「眉唾」かもしれない
- 測定方法がほぼ揃った同一組織において，製品を提供した後の状況を把握し，その分布や変化を知ることが品質改善の第一歩

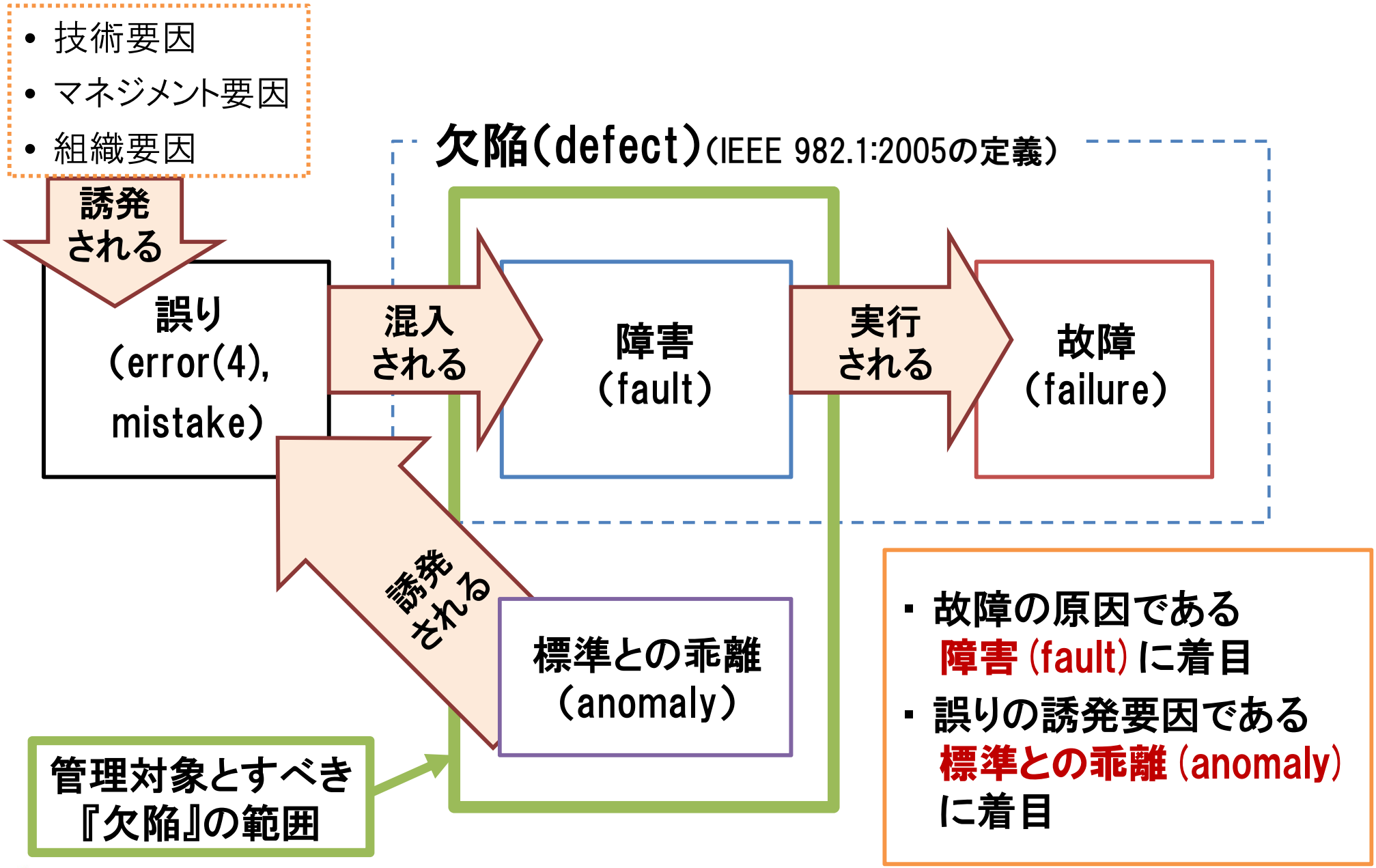
Cusumano, M., *et. al.*, “Software Development Worldwide: The State of the Practice,” *IEEE Software*, Nov/Dec 2003, pp.28-34.

欠陥？ 不具合？ IEEE標準の用語は？？

用語	定義(筆者訳)	出典のIEEE標準
欠陥 (defect)	障害(fault)または故障(failure)の**いずれかを言及できる一般的用語**。 ※個人的には，曖昧な定義であり好ましくないと考えます	982.1-2005
障害 (fault)	(1) ハードウェアデバイスまたはコンポーネントの欠陥(defect)。 (2) コンピュータープログラム内の，**不正確なステップ，プロセス，またはデータ定義**。一般には，エラー(error)やバグ(bug)をこの意味に用いる。	610.12-1990(R2002)
故障 (failure)	システムまたはコンポーネントが，指定された性能要求の範囲内で**要求された機能を果たせないこと**。	610.12-1990(R2002)
エラー， 誤り (error)	(1) 正しい結果と，観測された結果の相違。狭義のerror。 (2) 不正確なステップ，プロセス，またはデータ定義。狭義のfault。 (3) 不正確な結果。狭義のfailure。 (4) 不正確な結果を生み出した**人間の行為。狭義のmistake**。	610.12-1990(R2002)
不正 (anomaly)	要求仕様，設計文書，ユーザ文書，標準，または誰かの認識もしくは体験など，これらに基づく期待から乖離した状態。	1044-1993(R2002) 1028-2008
	文書において，またはソフトウェアもしくはシステムの運用において観測された，期待から乖離したものすべて。ただし，期待は，検証済みのソフトウェア製品，リファレンス文書，または指定された振る舞いを記述したその他の情報源に基づく。	829-2008

人の誤り(mistake) ―<混入>→ **原因箇所(fault)** ―<実行>→ **現象(failure)**

何を欠陥として捉えているのか？



欠陥の数をカウントするにあたって，欠陥分類が標準化されているとよい

機能性欠陥
(functional defects)

障害（fault)に相当

- **大規模欠陥**：機能性の欠如，大幅な追加･修正が必要
- **サポート**：ライブラリの欠陥
- **タイミング**：マルチスレッド環境で生じる問題
- **チェック**：妥当性確認ミス，不正な値の検出ミス
- **リソース**：データ，変数，リソース初期化，解放などの誤り
- **ロジック**：比較演算子，制御フロー，計算などの誤り
- **インタフェース**：ライブラリ，デバイス，DB，OS等とのインタフェース不整合

発展性欠陥
(evolvability defects)

標準との乖離（anomaly)に相当

- **構造**：コード構造
- **視覚的表現**：コードの読みやすさ
- **ドキュメンテーション**：コードの説明

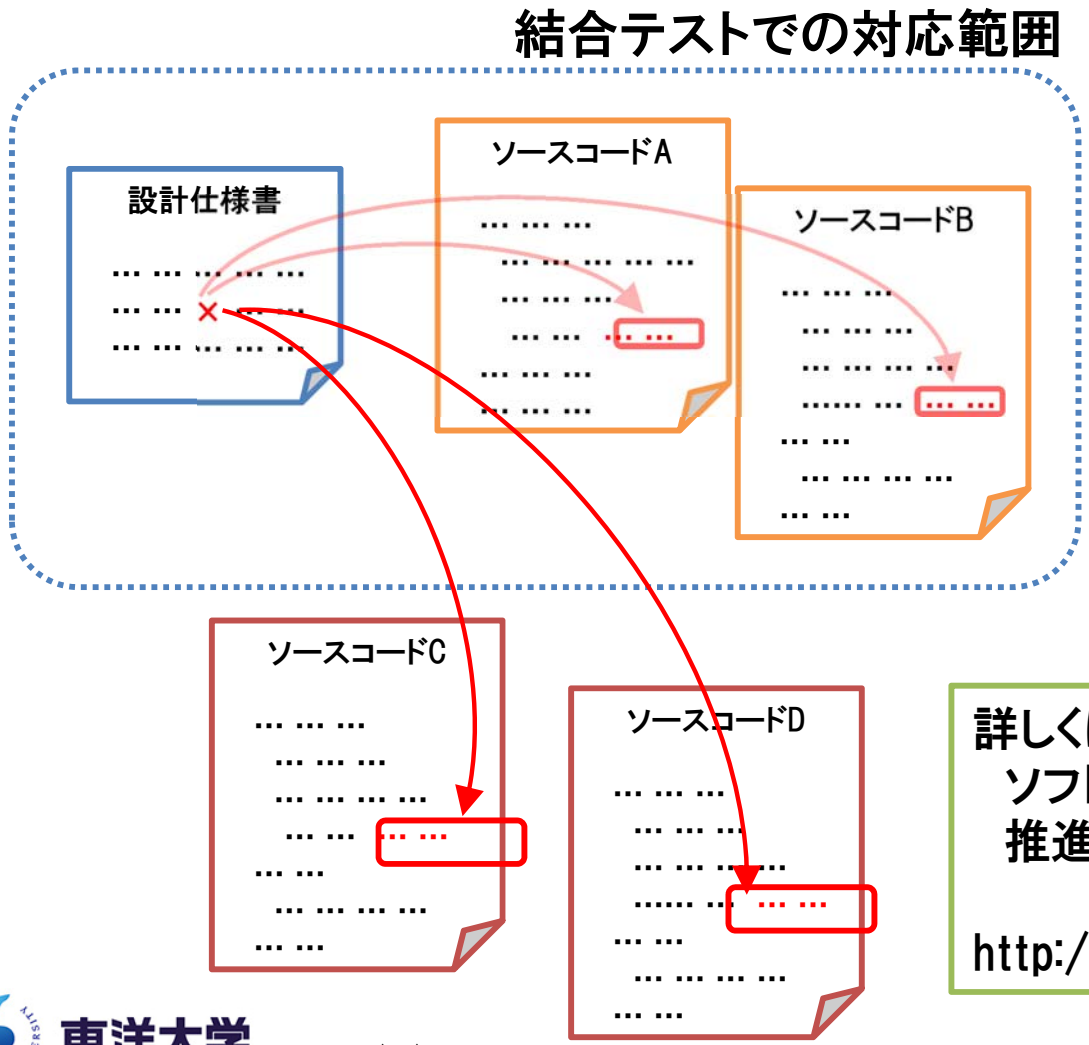
参考：コードインスペクション指摘の約70%は，発展性欠陥を検出
(Mäntylä, 2009)

- **欠陥の定量的管理の前提として，欠陥の分類を考えておく必要がある**
- **欠陥除去の前倒しについて議論するときは，機能性欠陥に着目した方が望ましい**
- **将来の機能性欠陥につながる発展性欠陥も，その混入･摘出状況を把握し，早期に摘出できていることを確認した方が望ましい**
- **欠陥の分類に着目して，プロジェクトの状況を把握する**

Mäntylä, M. V. and Lassenius, C., “What Types of Defects Are Really Discovered in Code Reviews?,” *IEEE Trans. Softw. Eng.*, vol. 35, no. 3, pp. 430-448, 2009.

欠陥の数をカウントするにあたって，その方法が標準化されているとよい

システムテストで，結合テストで対応をし忘れたソースコード2個所を発見し，
これを修正した。



この場合，全体として何件の
欠陥としてカウントしますか？

詳しくは「ソフトウェアジャパン2014
ソフトウェア品質データ分析を通じた組織的改善の
推進」をご覧ください。

http://www.youtube.com/watch?v=lU7cqPHUuEY

データから情報を読み取る

- **グラフに表す**
  - ヒストグラム，箱ひげ図，散布図，散布図行列などの基本的なグラフ
  - 独自の工夫を施したグラフ
  - 分布の形状を知る（正規分布，歪んだ分布，極端に0の多い分布，…）
  - 層別の必要性をチェックする（例：リリース後欠陥密度が多い・少ない）
  - 外れ値がないかをチェックする（例：工数が増えずに期間だけ増える）
- **数値で表す**
  - 基本統計量を見る：平均値，中央値，最頻値，比率，標準偏差など
  - 相関係数を見る　：相関係数，偏相関係数
  - 比率で表す　　：リリース後欠陥密度，レビュー密度　など
- **2軸または3軸で表す**
  - クロス集計表，層ごとの散布図
- **相関分析，回帰分析，検定などの手法を適用する**

数値の例：欠陥除去比率（DRE: Defect Removal Efficiency）

ある欠陥除去工程の開始時点で残存していた欠陥を，その工程でどれだけ除去できたかを表す

除去＼混入	要求定義	機能設計	詳細設計	コーディング	単体テスト	結合テスト	システムテスト	運用	合計	DRE
機能設計インスペクション	49	681							730	74.4%
詳細設計インスペクション	6	42	681						729	61.3%
コードインスペクション	12	28	114	941					1095	54.8%
単体テスト	21	43	43	223	2				332	36.7%
結合テスト	20	41	61	261	—	4			387	67.1%
システムテスト	6	8	24	72	—	—	1		111	58.1%
運用	8	16	16	40	—	—	—	1	81	
合計	122	859	939	1537	2	4	1	1	3465	97.7%

例：詳細設計インスペクションの DRE

DRE = 729 / (122+859+939 − 730)

プロセス全体の DRE

欠陥除去能力の低い工程を特定し，重点改善対象の候補とする

Laird, L. M. and Brennan, M. C., *Software Measurement and Estimation: A Practical Approach*, John-Wiley and Sons, 2006 (野中誠, 鷲崎弘宜訳：演習で学ぶソフトウェアメトリクスの基礎, 日経BP社, 2009).
Kan, S. H., *Metrics and Models in Software Quality Engineering*, 2nd ed., Addison-Wesley, 2002 (古山恒夫, 冨野壽監訳, ソフトウェア品質工学の尺度とモデル, 共立出版, 2004).

参考：相関係数の種類と，共分散の構造

・ピアソンの積率相関係数 (Pearson's product-moment correlation coefficient)
- 一般にいわれる相関係数，データが正規分布に従っているときに適用する

・順位相関係数 (rank correlation coefficient)
- データが正規分布に従っていない，曲線関係のとき，順序尺度のデータの場合に適用する
- 順序尺度のデータにも適用する

偏相関係数 (partial correlation coefficient)
- 着目している2つの変数以外の変数の影響を除外して，相関係数を求めるときに適用する

ピアソンの積率相関係数

$$r = \frac{s_{xy}}{s_x s_y}$$

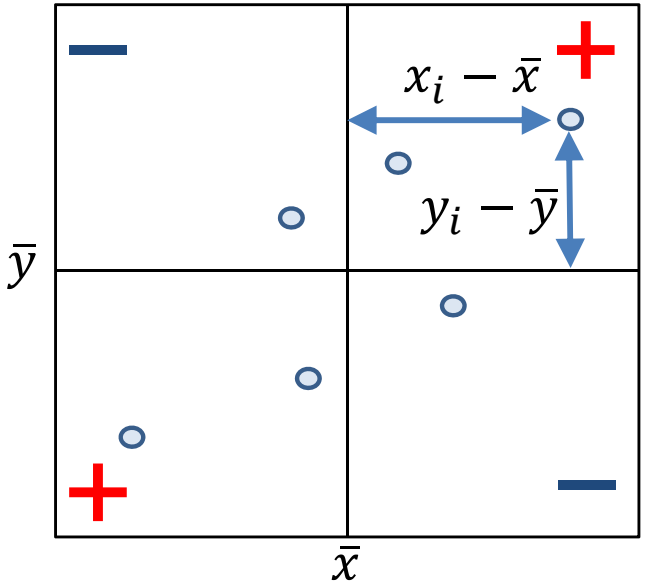
xとyの共分散を，
xおよびyの標準偏差
で割った値

共分散

$$s_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n}$$

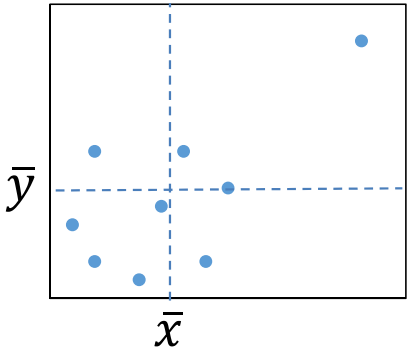
s_x : xの標準偏差

s_y : yの標準偏差



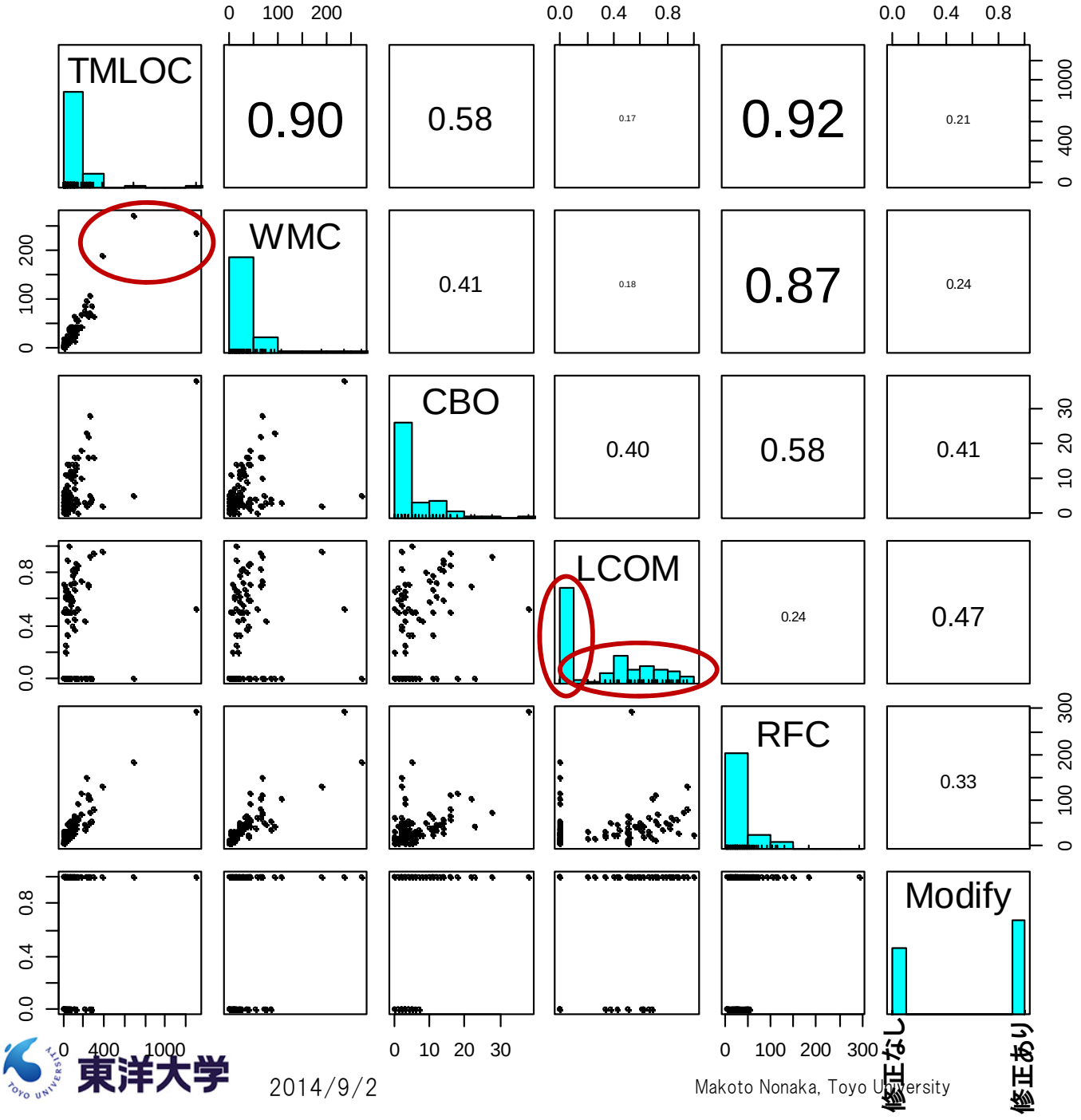
右上と左下にデータが多く分布すれば
共分散はプラスになっていく

左上と右下にデータが多く分布すれば
共分散はマイナスになっていく



なので，
上記のような分布でも，
相関係数は 0.67 という
大きな値になってしまう

散布図行列の例：クラスのプロダクトメトリクスと修正有無の関係



n = 102

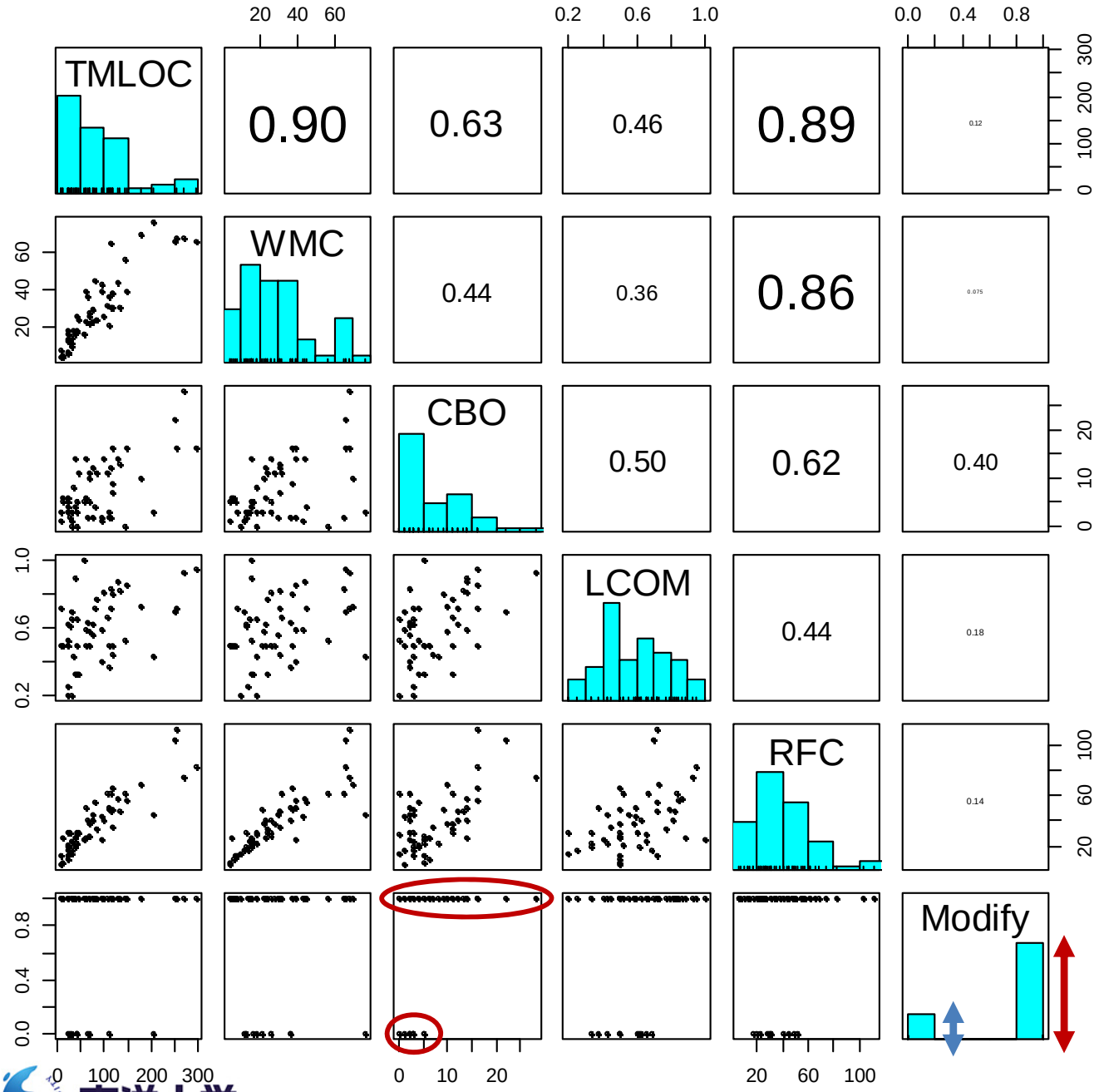
どこに着目しますか？

- 規模の大きいクラス
- LCOMの分布

参考：
R の psych ライブラリの pairs.panels関数で作図

```
pairs.panels(
    data,
    smooth=FALSE,
    density=FALSE,
    ellipses=FALSE,
    scale=TRUE
)
```

散布図行列の例：
規模について外れ値を除外し，LCOM > 0 で絞りこんだデータ

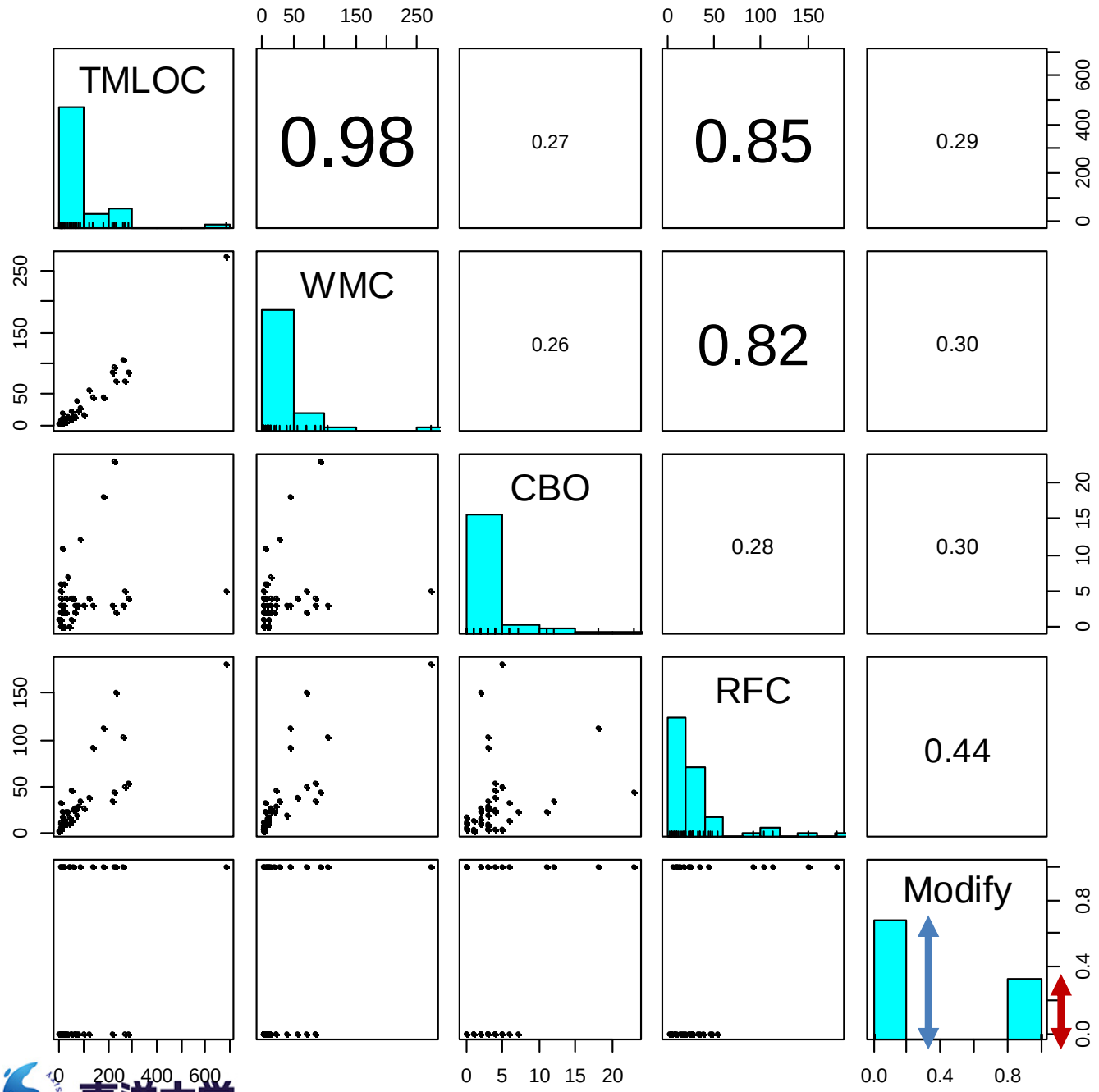


n = 53

この散布図行列から
何が読み取れますか？

- 修正有無の比率
  ⇒ 修正率が高い
- CBOと修正有無の関係

散布図行列の例：LCOM = 0で絞り込んだデータ



n = 47

この散布図行列から
何が読み取れますか？

・修正有無の比率
⇒ 修正率が低い

分析によって得られた知見

- コード行数が**380行以上**という大きいクラスは3個あり，これらはすべて
  リリース後に欠陥修正があった**(欠陥修正率3/3 = 1.0)**。
- **凝集度が欠如していない**クラスは46個あり，このうちリリース後に欠陥修正が
  あったのは15個であった**(欠陥修正率15/46 = 0.33)**。
- 一方，**凝集度が欠如した**クラスは53個あり，このうちリリース後に欠陥修正が
  あったのは42個であった**(欠陥修正率42/53 = 0.79)**。
- 凝集度が欠如していたクラス53個について，**結合度**の値が**中央値の5以下**
  では**欠陥修正率が17/28 = 0.61**であったのに対して，
  中央値より大きい場合では**欠陥修正率が25/25 = 1.0**であった。

**レビューやテストの重点化を進めるにあたって，
このようなデータから得られた客観性の高い知見は
プロセス改善の推進力になる**

目的変数が修正有無という二値なので，ロジスティック回帰分析を適用する方法もある

回帰分析の例：このようなデータをもとに欠陥数をどうやって予測する？

Prj	Effort	KLOC	STDefects	Productivity	STDefDensity	S1	S2	S3	S4	S5	S6	S7	F1	F2	F3	D1	D2	D3	D4
1	7109	6	148	0.844	24.67	4	3	5	4	3	4	2	3	2	3	2	4	4	4
2	1308	1	31	0.765	31.00	4	4	5	4	3	4	4	2	1	3	2	4	4	4
3	18170	54	209	2.972	3.87	4	4	5	4	4	5	4	4	4	5	4	5	4	5
5	9434	14	373	1.484	26.64	4	3	4	3	4	NA	3	4	4	5	2	3	4	4
6	9441	14	167	1.483	11.93	5	3	5	3	4	NA	4	3	3	5	3	4	3	3
7	13888	21	204	1.512	9.71	2	3	5	4	4	2	3	2	1	3	3	5	4	4
8	8822	6	53	0.680	8.83	3	3	4	3	4	2	4	3	2	3	4	4	3	3
9	2192	3	17	1.369	5.67	4	5	5	4	5	3	5	2	2	3	4	5	5	4
10	4410	5	29	1.134	5.80	4	4	4	3	4	3	4	3	2	3	4	4	4	4
11	14196	4	71	0.282	17.75	4	3	4	3	4	4	4	4	4	4	4	4	4	4
12	13388	19	90	1.419	4.74	4	3	4	3	3	3	2	4	4	4	5	3	3	4
13	25450	49	129	1.925	2.63	5	3	3	2	3	4	2	4	4	4	4	4	4	4
14	33472	58	672	1.733	11.59	4	4	4	4	4	4	4	5	4	4	4	4	4	4
15	34893	154	1768	4.413	11.48	4	4	4	4	4	5	1	4	4	3	4	4	4	4
16	7121	27	109	3.792	4.04	4	4	4	4	4	4	3	2	1	3	4	4	4	4
17	13680	33	688	2.412	20.85	5	4	3	2	4	4	3	2	1	3	3	3	4	4
18	32366	155	1906	4.789	12.30	3	4	4	4	4	5	1	5	5	4	3	4	4	4
19	12388	87	476	7.023	5.47	4	3	4	4	4	4	3	4	4	4	4	4	4	4
20	52660	50	928	0.949	18.56	2	2	3	1	2	3	1	5	4	5	1	1	2	4
21	18748	22	196	1.173	8.91	4	4	4	3	2	3	3	2	3	5	4	4	4	4
22	28206	44	184	1.560	4.18	2	2	3	3	3	3	2	3	3	5	4	3	2	4
23	53995	61	680	1.130	11.15	3	4	5	4	2	3	3	4	5	5	2	4	4	4
24	24895	99	1597	3.977	16.13	3	3	3	4	3	4	2	3	3	4	3	4	4	3
25	6906	23	546	3.330	23.74	3	4	NA	4	3	3	3	4	1	4	3	4	3	4
27	14602	52	412	3.561	7.92	4	3	5	3	3	5	3	4	5	5	3	2	3	4
28	8581	36	881	4.195	24.47	4	2	5	3	3	3	2	5	5	5	3	2	4	4
29	3764	11	91	2.922	8.27	4	3	5	4	3	3	5	3	3	4	5	5	4	4
30	1976	1	5	0.506	5.00	4	4	5	4	4	3	5	2	2	3	4	4	4	4
31	15691	33	653	2.103	19.79	NA	4	4	3	3	4	3	3	3	4	4	4	4	4

Fenton, N., et al. *Project Data Incorporating Qualitative Facts for Improved Software Defect Prediction. in Predictor Models in Software Engineering, 2007. PROMISE'07: ICSE Workshops 2007. International Workshop on. 2007.*

分析対象のデータ

- **定量データ**
  - Effort：開発工数
  - KLOC：ソースコード行数
  - STDefects：システムテスト検出欠陥数（IV&Vで発見された機能欠陥の数）
  - STDefDensity = STDefects / KLOC ： システムテスト検出欠陥密度
- **定性データ ～ それぞれの要因についての5段階評価（1 ～ 5）**

S: 要求定義プロセス

S1: 当該工程に関するメンバーの経験
S2: ユーザ要求文書の品質
S3: 要求・仕様・テスト仕様レビューの充足度
S4: 要求レビュープロセスの効率性
S5: 要求レビューの有効性
S6: 仕様レビュー指摘欠陥の多さ
S7: 要求の安定性

F: 新規開発機能

F1: 新規開発機能の複雑度
F2: 新規開発機能の規模
F3: ベース規模に対する新規規模の大きさ

D: 開発プロセス

D1: 当該工程に関するメンバーの経験
D2: プログラマの能力
D3: 設計レビュープロセスの効率性
D4: 開発メンバーのモチベーション

T: テスト，手戻り

T1: テストプロセスの効率性
T2: 結合テストに関するメンバーの経験
T3: 統合テストに関するメンバーの経験
T4: テストケースの質

P: プロジェクト管理

P1: 開発メンバーのトレーニングの質
P2: 構成管理プロセスの効率性
P3: プロジェクト計画の適切さ
P4: 開発拠点/グループの多さ
P5: 主要なステークホルダの関与度
P6: 顧客の関与度
P7: 協力会社マネジメントの有効性
P8: チーム内コミュニケーション
P9: プロセス成熟度

Fenton, N., et al. *Project Data Incorporating Qualitative Facts for Improved Software Defect Prediction. in Predictor Models in Software Engineering, 2007. PROMISE'07: ICSE Workshops 2007. International Workshop on. 2007.*

回帰分析 (Regression Analysis)

- **回帰分析により解きたい問題**

  ・間隔尺度の量的変数 X と量的変数 Y の間の直線的な関係を把握したい
  ・X がある値をとるとき，Y はどのような期待値をとるのか予測したい

  例）開発規模とテスト不具合数の関係は？
  例）規模がある値のとき，テスト不具合数の期待値はどのくらいか？
  注）X が順序尺度のとき，Y の値の予測は意味をなさない

- **モデル式：** $\log(\text{STDefects}) = f(\log(\text{KLOC})) = \alpha + \beta X + \varepsilon$　　**εは誤差**

- **分析結果**

```
RegModel.1 <- lm(log(Failures)~log(KLOC), data=data1)
summary(result1)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.46056    0.32469   7.578 3.76e-08 ***
log(KLOC)    0.95636    0.09792   9.766 2.35e-10 ***

Multiple R-squared: 0.7794, Adjusted R-squared: 0.7712
F-statistic: 95.38 on 1 and 27 DF,  p-value: 2.351e-10
```

直線の係数の推定値

p値：係数がゼロという帰無仮説の下で，このようなデータが得られる確率

寄与率（調整済みR^2）
当てはめた直線によって説明されるyの変動の割合

変数が増えれば増えるほどR^2の値は増えてしまうが，調整済みR^2ではその影響を除外している

回帰分析を行うには前提条件がある

- **データの独立性** (independence)
  - 均質なデータ集合の中から無作為に抽出されていること
  - 同一固体から繰り返し測定する場合に，適切な代表値が得られていること
- **分散の均一性** (homogeneity of variance)
  - 当てはめられたモデルの周りでのデータの散らばり具合はどこも等しいこと
- **誤差の正規性** (normality of error)
  - 当てはめられたモデルの周りでの残差が正規分布に従うこと
- **線形性（加法性）** (linearity / additivity)
  - 目的変数と説明変数の間に線形な関係を推定できること
  - 変数変換により線形性を確保してもよい

例） $Effort = A * Size^{B}$ → 両辺の対数をとる

$\log(Effort) = \log(A) + B * \log(Size)$ → これも「線形」と考えてよい

Alan Grafen and Rosie Hails, 野間口謙太郎, 野間口眞太朗訳：一般線形モデルによる生物科学のための現代統計学, 共立出版, 2007.

重回帰分析：
説明変数を増やして回帰モデルの説明力を高める

- **目的変数に影響していそうな変数を見つける**
  - 散布図と相関係数に基づいて「あたり」をつける
  - 経験則に基づいて，変数間の因果関係を考える
  - ピアソンの積率相関係数，スピアマンの順位相関係数を調べる
  - それらについて，無相関の検定を行う
- **層別した上でさらに分析する**
  - 性質の異なるデータであれば層別して分析する

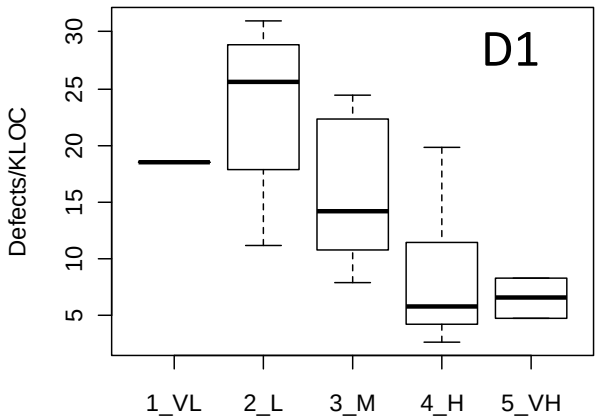
そのほかの定性的なデータも使って予測精度を高めたい

相関係数を使って，有用そうな変数にあたりをつける

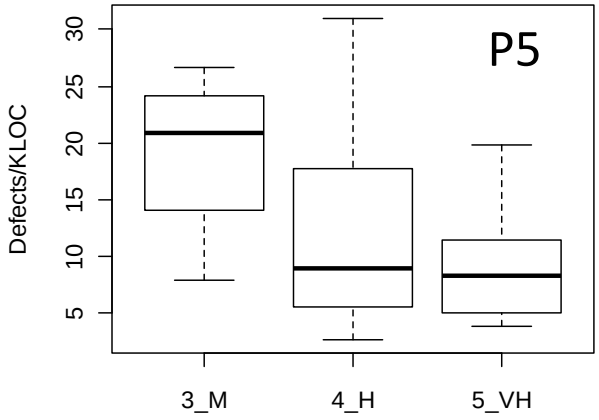
データ項目	欠陥密度	不具合数
D1：開発スタッフ経験	-0.693	-0.408
P5：ステークホルダ関与度	-0.435	-0.254
P6：顧客関与度	-0.374	-0.137

※ スピアマンの順位相関係数を使っています

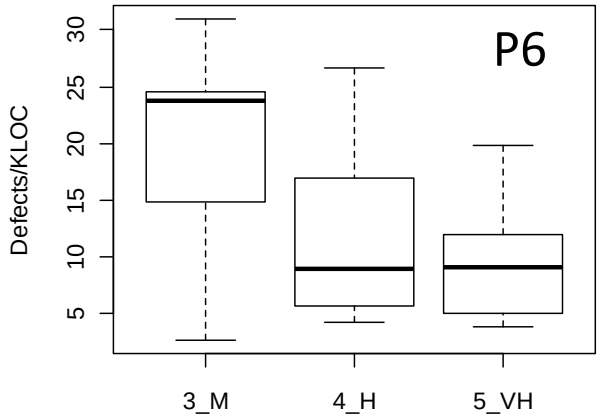
箱ひげ図を描いて，欠陥数(欠陥密度)に影響する要因を探る
もちろん，要因と効果の因果関係も検討しなければならない



Development Staff Experience



Stakeholder involvement



Customer involvement

ステップワイズ変数選択：意味のある変数を自動で選ぶ

```
log(STDefects) ~ log(KLOC) + D1 + P5 + P6
               Df Sum of Sq    RSS     AIC
- P6            1    0.0182 1.3088 -81.848
<none>                     1.2906 -80.253
- P5            1    0.1113 1.4019 -79.854
- D1            1    0.6333 1.9240 -70.674
- log(KLOC)     1    8.0395 9.3301 -24.887
```

```
RegModel.1 <-
lm(log(STDefects)~log(KLOC)+D1+P
5+P6, data=Dataset)

RegModel.1 <- step(RegModel.1)

summary(RegModel.1)
```

```
lm(formula = log(STDefects) ~ log(KLOC) + D1 + P5 + P6, data = data))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.04474    0.27477   7.442 8.56e-08 ***
log(KLOC)    0.93585    0.07479  12.513 2.91e-12 ***
D1          -0.17664    0.05014  -3.523  0.00167 **
P5          -0.08503    0.06325  -1.344  0.19089

Multiple R-squared:  0.882,     Adjusted R-squared:  0.8678
F-statistic: 62.26 on 3 and 25 DF,  p-value: 9.746e-12
```

寄与率が0.77から0.87まで向上した

説明変数を選択するときの留意事項

- **p値に着目，統計的有意か否かを調べる**
  - `Pr(>|z|)` → 係数がゼロという帰無仮説の下で，このような係数の値が推定される確率
- **変数の符号に着目，論理的に妥当な関係か否かを調べる**
  - 例）ツール使用スキルが高いと，工数は増える？ 減る？
  - 説明変数同士の相関が強いと多重共線性が生じ，符号が逆転する
- **論理的に意味のある変数かどうか調べる**
  - 目的変数を予測するのに，値が時間的に後に決まる説明変数はNG
- **変数の自動選択に委ねすぎてはいけない**
  - 「みせかけの相関」の変数をモデル式に含めないようにする
- **変数の増やしすぎに注意**
  - 変数が増えると，寄与率が自ずと高まる ⇒ 調整済みR2に着目
  - 本来なら有意ではない変数なのに，確率的に有意という結果が出ることも

順序尺度のデータを質的変数として扱う

- **変数を質的変数として指定する**
  - Rでデータセット編集 → 変数名を選んで「Character」を指定

```
lm(formula = log(STDefects) ~ log(KLOC) + D1, data = Dataset)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.18186    0.62186   5.117 3.49e-05 ***
log(KLOC)    0.93332    0.08195  11.389 6.23e-11 ***
D1[T.2]      0.04382    0.61327   0.071   0.9436
D1[T.3]     -0.24678    0.56547  -0.436   0.6666
D1[T.4]     -1.05327    0.55741  -1.890   0.0715 .
D1[T.5]     -1.16958    0.66047  -1.771   0.0898 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared: 0.8889,     Adjusted R-squared: 0.8648
F-statistic: 36.81 on 5 and 23 DF,  p-value: 3.085e-10
```

p値が大きく，
統計的に有意といえない

⇒ VL, L, M は併合して
一つの値でも十分かも

VL → VH に上がるにつれて，
係数が適切な方向に変化していることに注目

調整済みR2乗の値は
間隔尺度のモデルとほとんど変化なし

順序尺度に変換することにより，ダミー変数が導入される

元データ

D1
1_VL
4_H
4_H
2_L
3_M
3_M
2_L
3_M
3_M
5_VH
4_H
4_H

ダミー変数導入後

D1 2_L	D1 3_M	D1 4_H	D1 5_VH
0	0	0	0
0	0	1	0
0	0	1	0
1	0	0	0
0	1	0	0
0	1	0	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	0	1
0	0	1	0
0	0	1	0

※ 文字列データの名義変数は修正しなくてよい → Rが勝手にダミー変数にしてくれる

応用：ポアソン回帰分析による欠陥数の予測

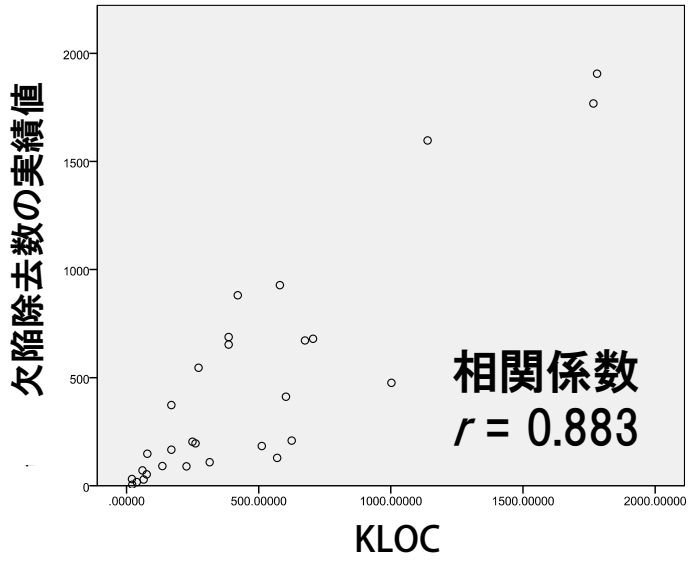
ポアソン分布 … （パラメータ次第だが）右裾を長く引く分布

分析対象のデータ(N = 29)

区分	記号	メトリクス
定量データ	Defects	欠陥検出数
	KLOC	ソースコード行数
定性データ（5段階評価）	D1	開発メンバーの経験
	P5	ステークホルダーの関与度
	P6	顧客とのコミュニケーション

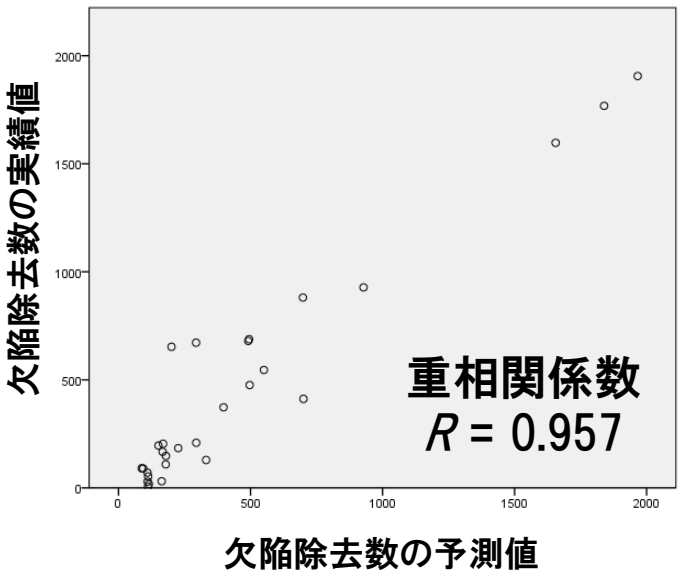
適切な統計手法を用いることで，欠陥数を高い精度で予測できるモデルを構築できる（場合もある）

規模と欠陥数の関係



重回帰分析

Defects = *f*(KLOC, D1, P5, P6)



（データの出典）Fenton, N., Neil, M., Marsh, W., Hearty, P. and Radlinski, L.: On the effectiveness of early life cycle defect prediction with Bayes Nets, *Empirical Software Engineering*, pp. 499-537, Springer, June 2008. http://www.springerlink.com/content/kg06211161305k1t/ よりダウンロード可能

野中（2011）「計数データとしてのソフトウェア欠陥の予測」『ウィンターワークショップ2011・イン・修善寺 論文集』pp.111-112, 情報処理学会

*"I'd rather be **vaguely right** than **precisely wrong**."*

精密に誤るよりも，漠然と正しくありたい

– John Maynard Keynes

- **私たちのデータ分析は「精密に誤って」いないか？**（自戒の念も込めて）
  - 目新しい高度な手法を適用することで満足していないか
  - 精度の低いデータなのに必要以上のミクロな分析をしていないか
  - 本来の目的からそれた分析結果を出そうとしていないか

- **「漠然と正しい」情報が得られているか？**
  - 正しい意思決定に役立つ，正しい方向感の情報を生み出せたか
  - データ分析の結果に期待する「顧客」に満足してもらえたか