ソフトウェア工学III

プロジェクト特性データの分析I
—— 可視化 ——

ソフトウェア工学講座
門田暁人
akito-m@is.naist.jp
B303室，内線5311

プロジェクト特性データの例

プロジェクトID	開発種別	業種	アーキテクチャ	開発言語(第1言語)	OS
1	a: 新規開発	a: 銀行	a: クライアントサーバ	d: VISUAL BASIC	g: WINDOWS NT
2	a: 新規開発	a: 銀行	b: スタンドアロン	f: PL/I	c: MVS
3	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
4	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
5	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
6	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
7	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
8	a: 新規開発	a: 銀行	c: 混合	d: VISUAL BASIC	c: MVS
…	…	…	…	…	…
73	b: 改修・保守		c: 混合	c: COBOL	

要求仕様_明確度合	開発期間(月数)	ピーク要員数	FP計測手法	規模(FP)	開発工数(人時)
	15	15	a: IFPUG	556	24690
c: ややあいまい	8	6	a: IFPUG	80	825
	6	1	a: IFPUG	77	758
a: 非常に明確	4	6	a: IFPUG	255	2119
	6	0	a: IFPUG	349	2741
d: 非常にあいまい	1	3	a: IFPUG	69	1090
b: かなり明確	4	11	a: IFPUG	375	1855
	6		a: IFPUG	271	1747
b: かなり明確	12	4	a: IFPUG	439	2007

導入 —— この講義の狙い

- データ分析に先立つ知識
  - 変数の定義の厳密さ，計測の信頼性や計測時期
  - 欠損値の扱い，外れ値の発見
- データ分析手法
  - データの分布を視覚的に調べる方法
    - 散布図，ヒストグラム，箱ひげ図，平行座標プロットなど
  - 検定
    - 標本から母集団の統計学的な性質を推測する
  - 予測
    - 重回帰分析，ケースベース推論，タグチメソッドなど
  - 発見的な手法
    - アソシエーション分析

何のための可視化・分析か？

- 過去の実績を現行プロジェクトの計画立案に役立てる.
  - プロジェクト計画とは
    1. 規模を見積もる.
    2. プロジェクト特性を決定する.
       - 開発期間
       - アーキテクチャ
       - プログラミング言語
       - 外注率，外注先
    3. 1., 2.から工数（人月）を見積もる.
- 過去の実績から分かること
  - 開発期間，開発要員数が現実的かどうか.
  - 生産性がどのくらいになりそうか.
  - テストにどのくらいの期間・工数をかければよいか. . . etc.

分析者の立場からみたデータ種別

- ＜個体×変数＞型のデータ（静的）
  [例] 多数のプロジェクトの特性値（開発者数，開発期間．．．）
  　　多数のモジュールのメトリクス（行数，ループ数．．．）
  - 変数間の相関や分布の調査，予測，ルール発見
  - 様々な統計的手法が使える．

- 時系列データ（動的）
  [例] ソースコード規模の時間変化
  　　バグ発見数の時間変化
  - 変動要因の解析，予測
  - 自己回帰モデルなどの統計的手法があるが，人間が介在するデータの解析は難しい．

＜個体×変数＞型のデータ

- 情報処理推進機構（IPA）ソフトウェアエンジニアリングセンター
  - 1419プロジェクト×約400変数
  - 日本のソフトウェアベンダー19社から収集したもの
- International Software Benchmarking Standards Group (ISBSG)
  - 3024プロジェクト×99変数
  - 20カ国から収集したもの
- NASA IV&V機構Metrics Data Program
  - 約14000モジュール×27変数 など
  - 10プロジェクトのデータが公開されている．

大手ソフトウェアベンダーの多くは，これらに類似するデータを収集し，品質保証部門や生産管理部門で活用している．

題材

- 79プロジェクト，10変数のデータセット project1.csv
  - 演習用に作成した典型的なもの
  - 欠損値を含む
  - 欠損値にマイナスの値を埋めたもの project2.csv
- 77プロジェクト，11変数のデータセット canada.csv
  - カナダのソフトウェア企業のもの
  - 欠損値なし

ソフトウェア工学IIIのWebサイトよりダウンロードしてください．

canada.csvに含まれる変数

- case-ID：プロジェクトID
- Duration：開発期間（月数）…. 1ヶ月＝約160時間
- ExpEquip：開発チーム経験年数
- ExpProjMan：プロジェクトマネージャ経験年数
- Transactions：トランザクション数
- RawFPs：規模（調整前FP）
- Adj Factor：FP調整係数
- Adj FPs：規模（調整後FP）
- Dev Env：開発環境
- Year Fin：開発終了年
- Entities：エンティティ数
- ActualEffort：開発工数（人時）

導出可能な変数

生産性＝Adj FPs ÷ ActualEffort （1人時で開発可能な規模）
平均要員数＝ActualEffort ÷ (Duration × 160)

データ分析ツール

- Weka
  - ニュージーランドのWaikato大学で開発されているフリーのデータマイニングツール
- DAVIS
  - 韓国の成均館（ソンギュングァン）大学で開発されているフリーのデータマイニングツール
- 群馬大学 青木繁伸先生による統計ツールセット
  - Black-Box（Webベース）
    http://aoki2.si.gunma-u.ac.jp/BlackBox/BlackBox.html
  - JavaScript による統計計算
    http://aoki2.si.gunma-u.ac.jp/JavaScript/index.html
- その他（Excel, R, Rコマンダー, SPSS等）

データ分析に先立つ知識

- 変数の定義の厳密さ
- 計測の信頼性
- 変数の種類（尺度）
- 計測時期
- 欠損値の扱い
- 外れ値の発見・除去

変数の定義の厳密さ

- 定義が厳密でないものがある.
  - 厳密さ＝高：アーキテクチャ，言語など
  - 厳密さ＝中：開発種別，開発要員数など
    - 新規か再開発かは区別しづらいことも多い.
    - どこまでを開発要員に含めるかは難しい.
  - 厳密さ＝低：要求仕様の明確さなど
    - 「非常に明確」「かなり明確」「ややあいまい」「非常にあいまい」の4カテゴリからなる順序尺度であるが，厳密な定義を与えることは難しい.

各変数の定義の厳密さ念頭において分析を進める必要がある.

計測の信頼性

- 計測値の信頼性には大きなばらつきがある.
  - 信頼性＝高：アーキテクチャ，言語など
    - 計測ミスが起こりにくい.
  - 信頼性＝中：規模（FP）など
    - 計測する人や組織によって多少のばらつきがある.
    - FP計測手法は，組織毎に少しずつカスタマイズされている.
  - 信頼性＝低い：開発要員数（平均，ピーク）など
    - 外注先の開発要員数は，非常に計測しづらい.
    - 外注率＝ゼロのプロジェクトだと，信頼性は高い.

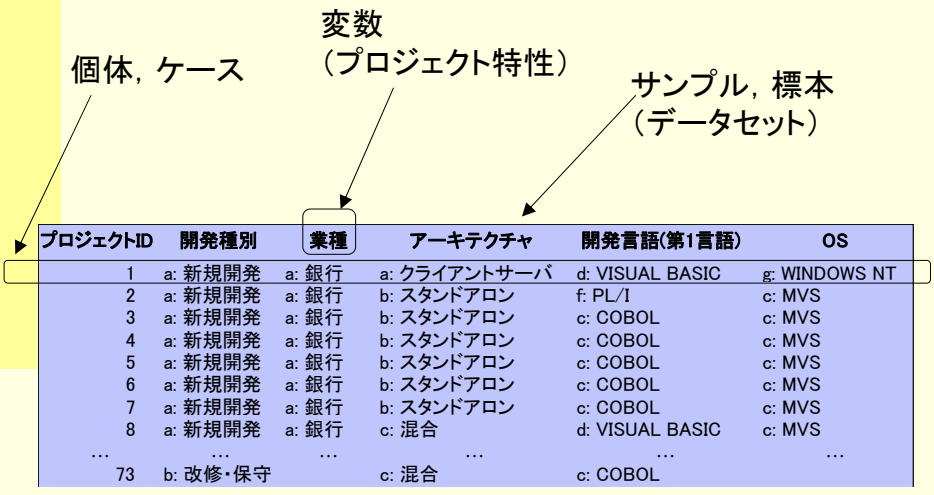
各計測値の信頼性を念頭において分析を進める必要がある.

計測の実態を把握せよ

- 定義が厳密で，計測が信頼できるとしても，得られた計測値が役に立たないこともある．
  - 例：ソースコード行数
    - 一般に，行数が多い＝規模が大きい≒作業量が多い．
    - COBOLなどでは実作業量を反映しやすいが，Javaなどの最近の言語では自動生成された行数を多く含む．
    - 自動生成された行数を除外してカウントするのは難しい．
    - それ以前に，コメント行を除外してカウントすることが意外に難しい．Google → “source code counter”
    - JavaのGUIアプリケーションの場合，ソースコード行数よりも，「画面数」の方が規模や作業量の指標として役に立つという考え方もある（かなり乱暴ではあるが）．

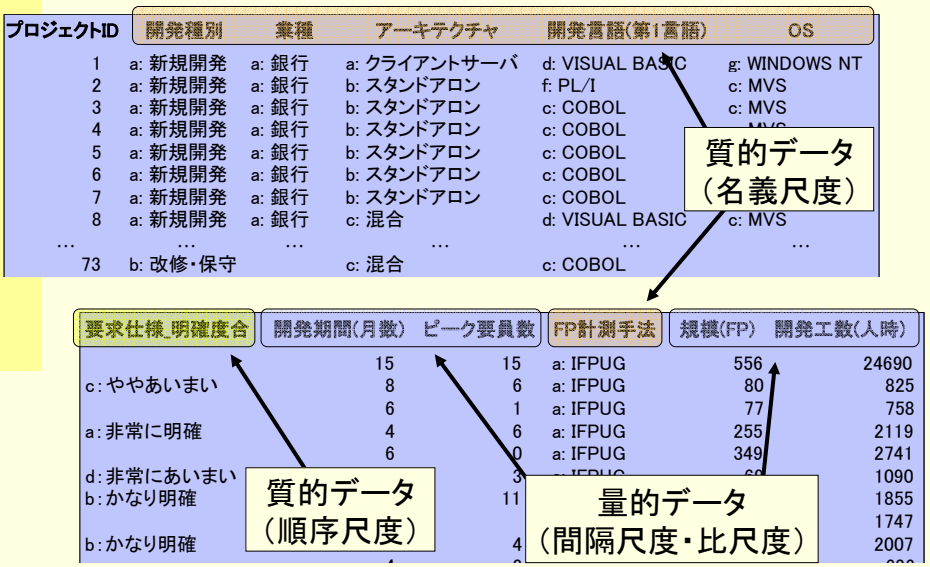
計測の実態を知った上で，計測値を利用することが望ましい．

用語



プロジェクトID	開発種別	業種	アーキテクチャ	開発言語(第1言語)	OS
1	a: 新規開発	a: 銀行	a: クライアントサーバ	d: VISUAL BASIC	g: WINDOWS NT
2	a: 新規開発	a: 銀行	b: スタンドアロン	f: PL/I	c: MVS
3	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
4	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
5	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
6	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
7	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
8	a: 新規開発	a: 銀行	c: 混合	d: VISUAL BASIC	c: MVS
…	…	…	…	…	…
73	b: 改修・保守		c: 混合	c: COBOL	

変数の種類



変数の種類

- 量的データ（数値変数）
  - 比尺度
  - 間隔尺度

  } 適用すべき統計手法に決定的違いはない．順序尺度への変換が可能である．
- 質的データ（カテゴリ変数）
  - 順序尺度
    - 実用上は量的データ（間隔尺度）と見なす場合もあるが，平均値や分散を算出しても意味がない．
  - 名義尺度
    - 2値データ（ダミー変数）
    - 多値データ → 複数の2値データに変換可能

    } 2値データは数値変数として扱うこともある．

欠損値の存在

プロジェクトID	開発種別	業種	アーキテクチャ	開発言語(第1言語)	OS
1	a: 新規開発	a: 銀行	a: クライアントサーバ	d: VISUAL BASIC	g: WINDOWS NT
2	a: 新規開発	a: 銀行	b: スタンドアロン	f: PL/I	c: MVS
3	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
4	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
5	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
6	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
7	a: 新規開発	a: 銀行	b: スタンドアロン	c: COBOL	c: MVS
8	a: 新規開発	a: 銀行	c: 混合	d: VISUAL BASIC	c: MVS
…	…	…	…	…	…
73	b: 改修・保守			c: COBOL	

欠損値

要求仕様_明確度合	開発…	…員数	FP計測手法	規模(FP)	開発工数(人時)
	15	15	a: IFPUG	556	24690
c: ややあいまい	8	6	a: IFPUG	80	825
	6	1	a: IFPUG	77	758
a: 非常に明確	4	6	a: IFPUG	255	2119
	6	0	a: IFPUG	349	2741
d: 非常にあいまい	1	3	a: IFPUG	69	1090
b: かなり明確	4	11	a: IFPUG	375	1855
	6		a: IFPUG	271	1747
b: かなり明確	12	4	a: IFPUG	439	2007

欠損値の取り扱い

- 欠損値をなくす
  - 多くの統計手法（特に，重回帰分析などの予測手法）では，欠損値のないデータセットを予め作成することが求められる．
  - 欠損値処理法
    - 平均値挿入法：欠損値に対して，当該変数の平均値を挿入する．
    - リストワイズ除去法：欠損値を一つでも含むケースを削除する．
    - 他に，ペアワイズ除去法，ホットデック法，k-NN法などがある．
- 欠損値をそのままにしておく
  - 欠損値を埋めない方がよいことも多い．
    - 基礎統計量（平均値，中央値，分散など）の算出
    - 散布図，相関係数
    - アソシエーション分析　など

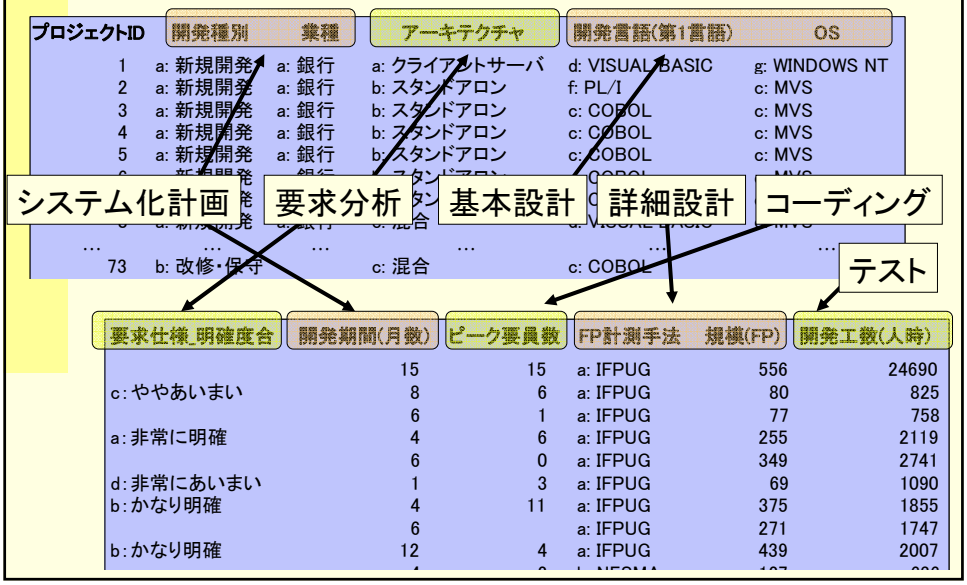
ホットデック法による欠損値挿入の例

要求仕様_明確度合	開発期間(月数)	ピーク要員数	FP計測手法	規模(FP)	開発工数(人時)
	15	15	a: IFPUG		
c: ややあいまい	8	6	a: IFPUG		
	6	1	a: IFPUG	77	758
a: 非常に明確	5	6	a: IFPUG	255	2119
	6	0	a: IFPUG	349	2741
d: 非常にあいまい	1	3	a: IFPUG	69	1090
b: かなり明確	4	11	a: IFPUG	375	1855
	6		a: IFPUG	271	1747
b: かなり明確	12	4	a: IFPUG	439	2007
	4	2	b: NESMA	127	636
	7	8	a: IFPUG	413	9033
	12	8			
	8	8	a: IFPUG	455	8520
	12	5	a: IFPUG	307	3652
	7	10	a: IFPUG	102	1252

値が比較的似ている.

平均値や中央値で埋めるよりは一般に精度がよい.
具体的なアルゴリズムは複数ある.

変数の計測時期

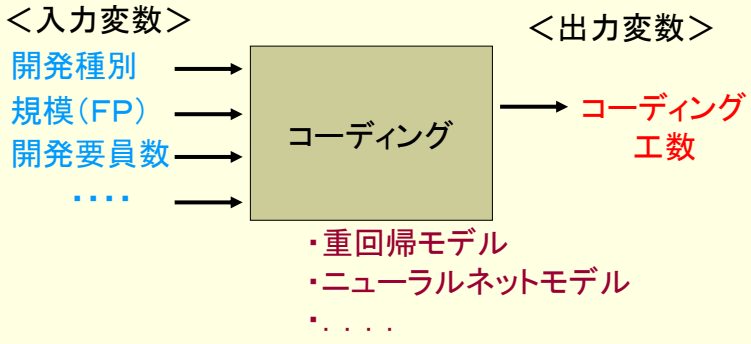


変数の計測時期を考慮した分析

- よい例，悪い例
  - ○ 開発種別，業種，規模(FP)から開発工数を予測する.
  - × 開発工数から開発規模を予測する.
  - × テスト工数からコーディング工数を予測する.
- 各変数の計測時期を考慮して，
  - 説明変数と目的変数を決める.
  - 仮説を決める.
    - 例：規模が大きいと生産性が悪い.

説明変数と目的変数

- 分析対象を，入力と出力の関係によりモデル化する.



- 統計では，入力変数を説明変数（独立変数），出力変数を目的変数(従属変数)と呼ぶ.

データの分布を調べよう

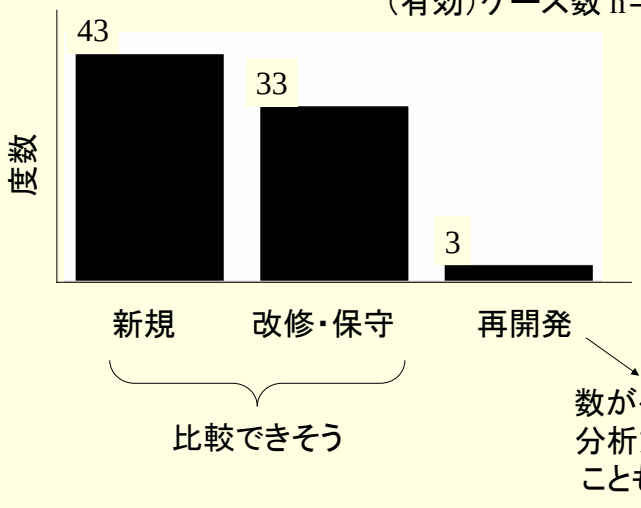
- 様々な分析手法（予測手法）の適用を考える前に，まず対象データを「理解」することが必要である．
  - １変数
    - 度数分布グラフ（棒グラフ，円グラフ） … カテゴリ変数の場合
    - ヒストグラム … 数値変数の場合
  - ２変数以上
    - 相関係数（行列），箱ひげ図

データを理解することで，計測ミスと思われる「おかしな値」を見つけたり，分析の方針を立てることが可能となる．

度数分布グラフの例１

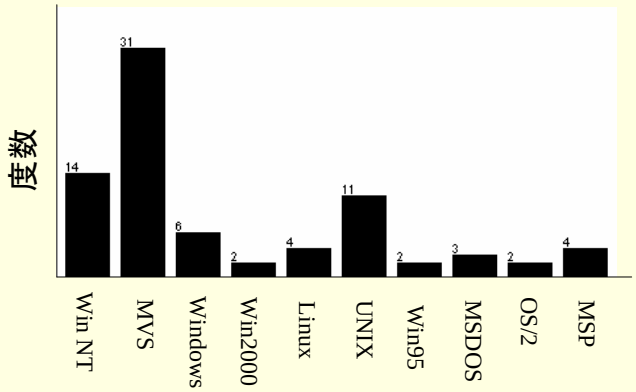
- 開発種別の棒グラフ

（有効）ケース数 n＝43+33+3=79



度数分布グラフの例2

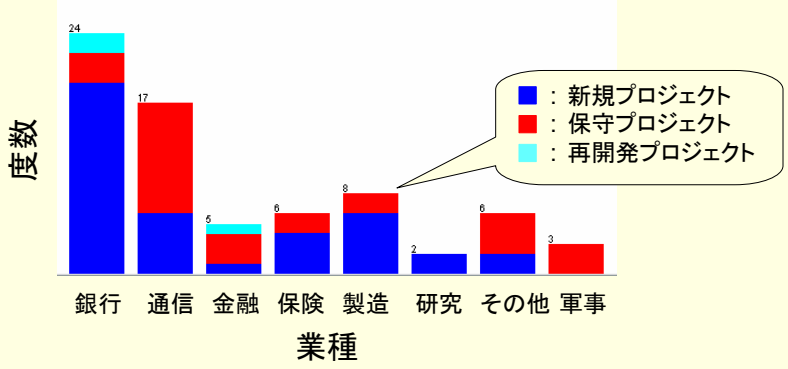
- OSの棒グラフ



細かくカテゴライズしすぎている.
より少数のカテゴリ(Windows, UNIX, DOS, メインフレーム)に集約させてから分析することが望ましい.

度数分布グラフの例3

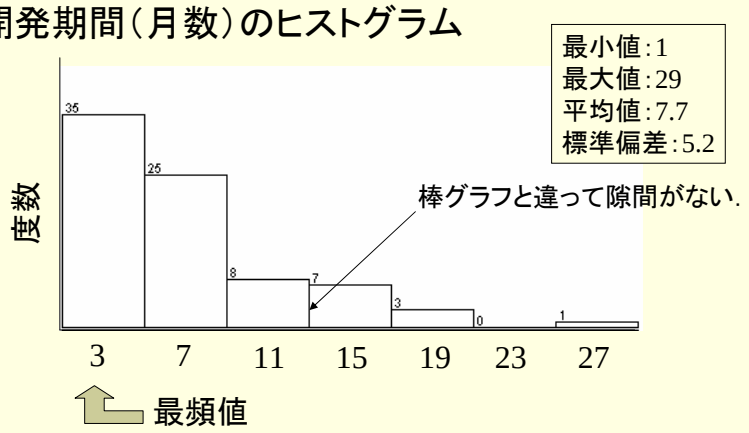
- 開発種別で色分けした業種の棒グラフ



ある変数の値で色分けすることで，2変数の関係がわかる.

ヒストグラムの例１

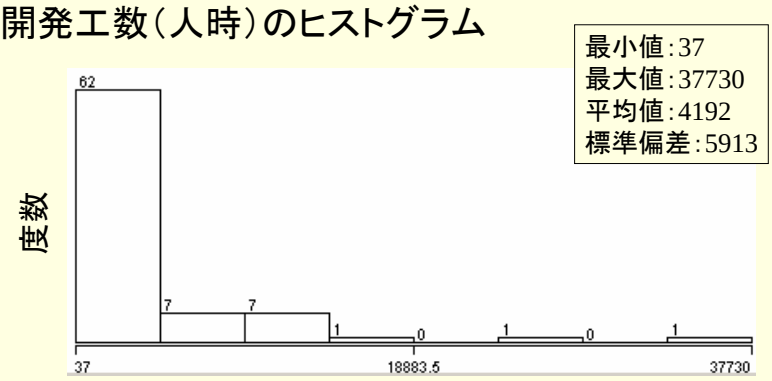
- 開発期間（月数）のヒストグラム



値の分布を大まかに把握したり，異常値を発見することができる．
階級の決め方に決定版はない．きりのよいところで切る．

ヒストグラムの例２

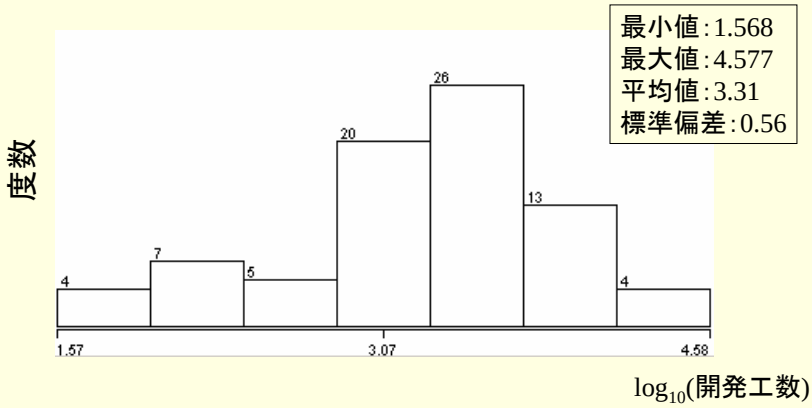
- 開発工数（人時）のヒストグラム



値の分布にかなりの偏りが見られる．（平均値は役立たない）
対数変換した方が分析しやすいことが多い．

ヒストグラムの例3

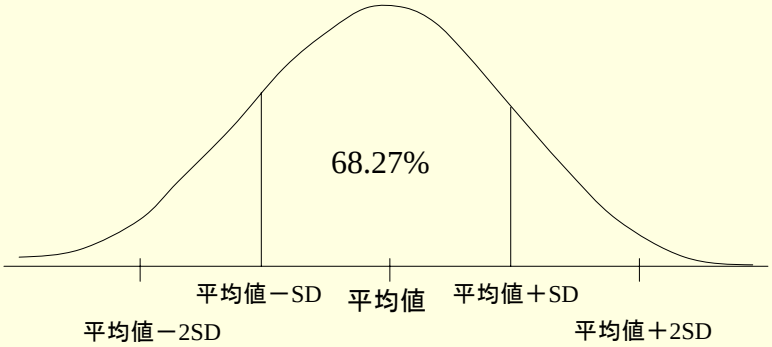
- 対数変換した開発工数（人数）のヒストグラム



値の分布が正規分布に近くなり，様々な統計手法を適用しやすい．
その反面，値の解釈が難しい．

正規分布における標準偏差の意味

平均値 ± 1 標準偏差（SD）の範囲内に全データの 68.27% が含まれる．



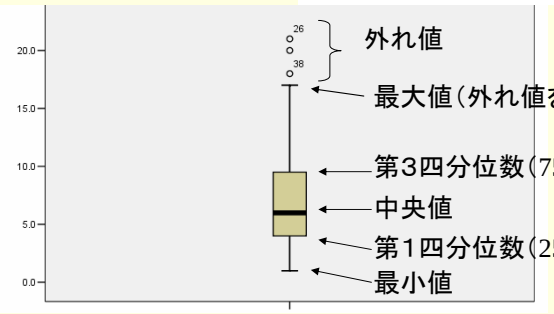
（平均値 ± 2SDでは95.45%，± 3SDでは99.73% ）

箱ひげ図（boxplot)

- 開発期間（月数）

処理したケースの要約

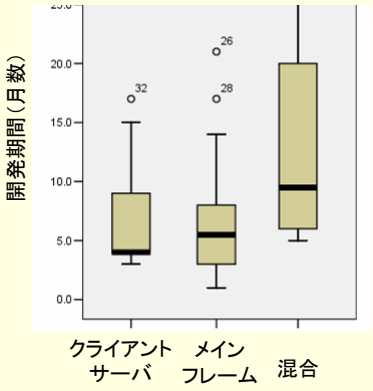
	ケース					
	有効		欠損値		合計	
	度数	パーセント	度数	パーセント	度数	パーセント
開発期間月数	75	94.9%	4	5.1%	79	100.0%



中央値からは，中心傾向が分かる．
箱の長さから，値の広がりやばらつきが分かる．
ヒゲの長さは，およその正常範囲を示す（ヒゲの両端の決め方にはいくつかの流儀がある）．

箱ひげ図（boxplot)による層別分析

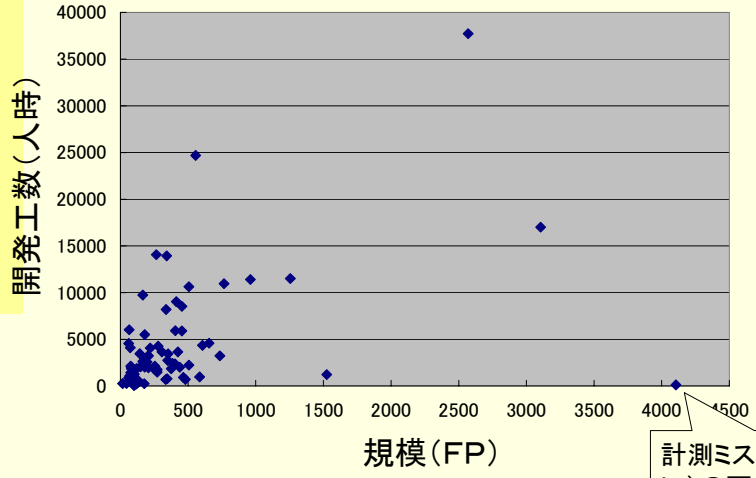
- アーキテクチャ（カテゴリ変数）×開発期間（数値変数）



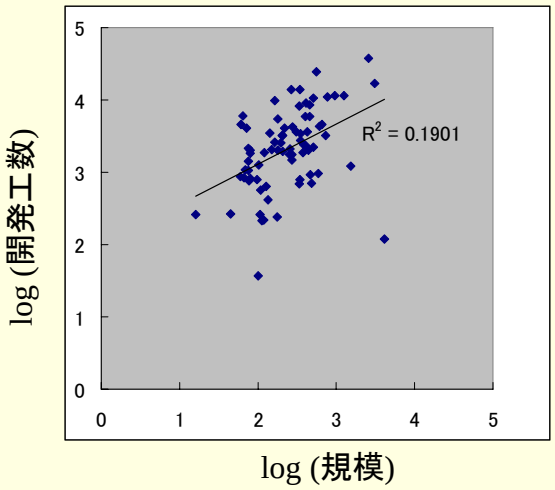
箱ひげ図による層別分析は，データ分析の基本であり，多くの報告書や論文で活用されている．

散布図

- 開発規模×開発工数

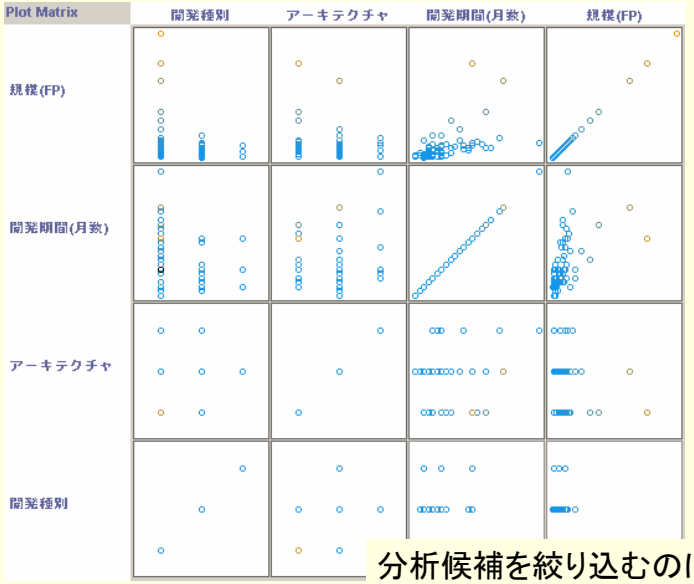


散布図（両対数グラフ）



常用対数 $\log_{10}X$ を用いる.
この例では，2変数間に明確な関係は見られない.

散布図マトリクス



分析候補を絞り込むのに役立つ.

相関係数行列(ピアソンの積率相関係数)

	開発期間(月数)	ピーク要員数	規模(FP)	開発工数(人時)
開発期間(月数)	1			
ピーク要員数	0.285025945	1		
規模(FP)	0.513136543	0.523879826	1	
開発工数(人時)	0.52297278	0.804970641	0.4456138	1

相関係数の解釈の目安:

0～0.2　ほとんど相関がない.
0.2～0.4　やや相関がある.
0.4～0.7　かなり相関がある.
0.7～1.0　強い相関がある.

順序尺度に対しては, 順位相関係数を用いることができる.

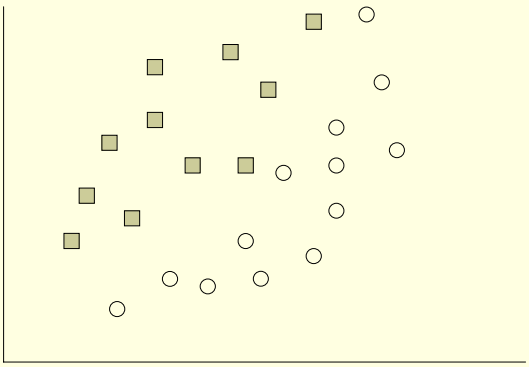
名義尺度同士の相関は？

- 2つの名義尺度間の関連の強さを表す指標として
  - ファイ係数
  - コンティンジェンシー係数
  - クラメールのV　　などがある.

この3つの中では，クラメールのVがよく使われる.

一方，名義尺度—量的データ間の関連の指標としては，分散分析における寄与率が挙げられる.

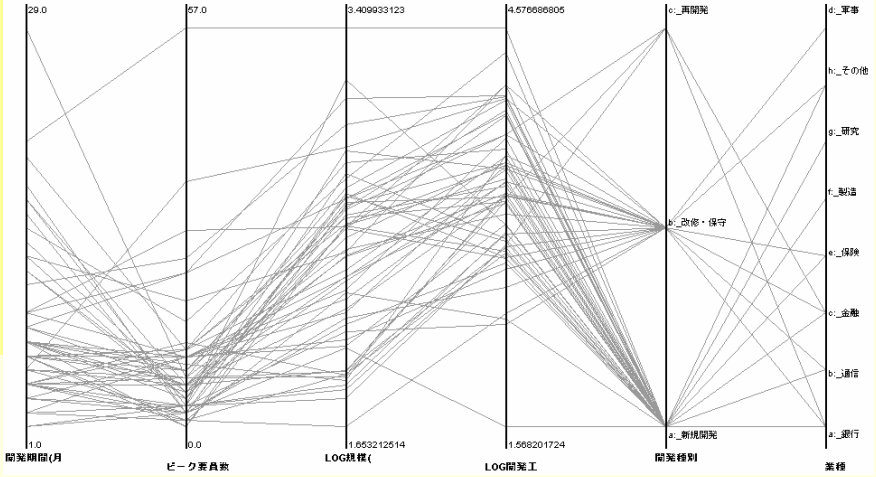
相関係数を用いるときの注意点



性質の異なるグループの個体が混在されている場合に，
全体として相関が見られなくなる.

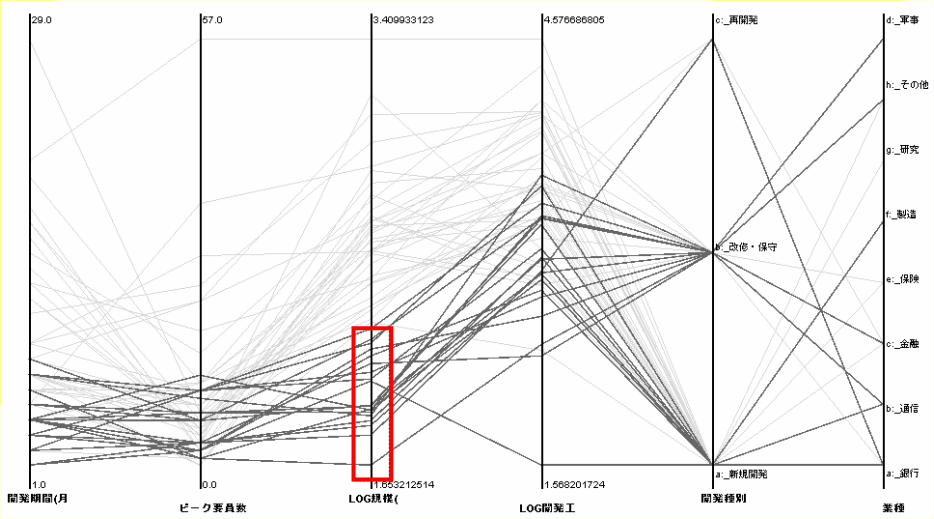
平行座標プロット（Parallel Coordinates Plot）

- 複数の変数の関係を調べるのに役立つ．



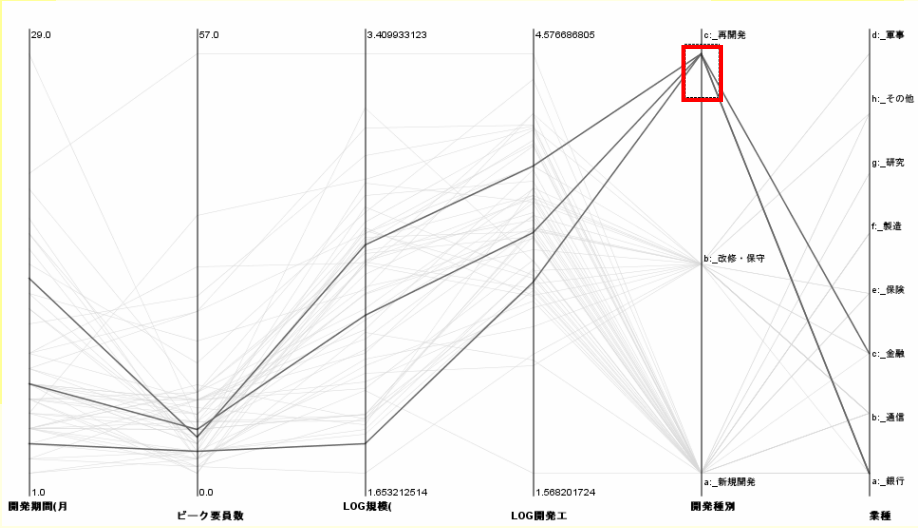
1本の線が1つの個体（プロジェクト）を表す．
欠損値を予めなくしておく必要がある（-1で埋めるなど）．

平行座標プロットの例1



規模が小さいと，開発期間が短く，ピーク要員数も小さい．開発工数は必ずしも小さくない．

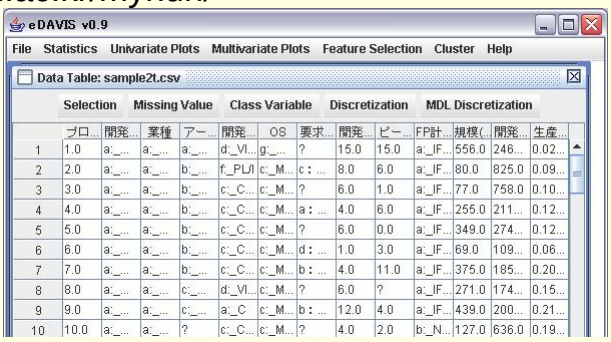
平行座標プロットの例2



再開発プロジェクトは，ピーク要員数が小さく，業種は金融と銀行のみである．

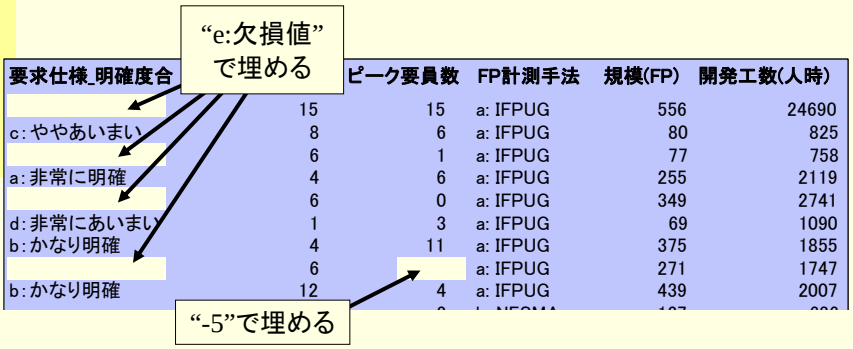
平行座標プロット　－　DAVIS

- DAVIS
  - 韓国の成均館（ソンギュングァン）大学の許文烈氏が開発したフリーのデータ視覚化ツール
  - Java VM上で動作する．
  - http://stat.skku.ac.kr/myhuh/



平行座標プロット　－　DAVIS

- 入力
  - csv形式のファイル
  - 空白セルを「欠損であることを示す値」で予め埋めておく
  - 変数によっては対数変換（log）を予め行っておく



要求仕様_明確度合		ピーク要員数	FP計測手法	規模(FP)	開発工数(人時)
	15	15	a: IFPUG	556	24690
c: ややあいまい	8	6	a: IFPUG	80	825
	6	1	a: IFPUG	77	758
a: 非常に明確	4	6	a: IFPUG	255	2119
	6	0	a: IFPUG	349	2741
d: 非常にあいまい	1	3	a: IFPUG	69	1090
b: かなり明確	4	11	a: IFPUG	375	1855
	6		a: IFPUG	271	1747
b: かなり明確	12	4	a: IFPUG	439	2007

平行座標プロット　－　利点と課題

- 多変数の関連を見るのに役立つ．
  - ただし，８変数ぐらいまでが限界．
- 定量的な分析は別途行う必要がある．
  - 箱ひげ図，t検定，分散分析