統計数理（1991）
第39巻 第1号 85-96

共同研究報告

EJDAにおけるデータ記述

慶應義塾大学 理工学部　柴田　里程・渋谷　政昭

（1991年6月　受付）

統計学の学術雑誌を1つの電子ジャーナルとして実現するEJDA（Electronic Journal of Data Analysis）プロジェクトにおける中心的な課題が「データとその記述の一体化，標準化」のルール設定である．本論文では，そのために著者らが考案したD＆Dについて，その意義と価値，設計方針，構造を具体例を交えながら解説する．D＆Dは特定のハードウェア，ソフトウェアに依存しない，テキスト形式の「データとその記述」で，1つのリスト構造を成している．

1. EJDAとD＆D

EJDAは統計学の学術雑誌を電子メディア，特に近年急速に発達してきたネットワークを通じて実現しようというものである．この発想は現在ではもはや特に目新しいものではなくなってきている．ネットワークとワークステーションの普及によって具体的なシステムとして運用するための技術的条件は整いつつあり，著作権，査読者の確保など運用上の諸問題は残っているものの，そう重大な障害ではない．

データ主体の電子雑誌である以上，データとその背景情報と属性の記述をどう扱うかが主要な問題である．したがって，以下で解説するD＆D(Data and Description)の必要性はEJDAの構想のきわめて初期の段階で認識された．「データとその記述の一体化，標準化」を何らかの形で行なわずにデータの流通，蓄積を始めても，混乱を増すだけで，あまり役に立たないことは明らかである．

まず，この一体化，標準化の方式として，すでに用いられている方式で有望なものはないかと探したが，すべて，オペレーティング・システム，解析ソフトウェアに固有のもので，汎用性がなく，我々の目的には合わなかった．また，データとその記述の一体化という点でも，何らかの形で一体化の努力はしているものの，組織立ったものはなかった．そこで，独自に1つの方式を開発することになった．もちろんその際留意したのは，特定のハードウェア，ソフトウェアからの独立性であり，汎用性である．これらの条件から導かれる自然な形態は，人間にとっても可読な，いわゆるテキスト形式で，それぞれのデータの記述，関係の記述などかなり複雑なものまで表現できるものということになる．しかし，D＆Dは可読であるからといって必ずしも人間がそのままの形で読み，理解し，利用することを期待したものではなく，計算機の間の1つの媒体としての役割を担うものである．したがって，1つの形式的な文法に従って記述される必要もある．このように，特定のソフトウェア，ハードウェアに依存しない記述言語という点では，プリント出力用のポストスクリプト言語（アドビ・システムズ（1988））と似ている．ただし，D＆Dの場合はデータの記述であるので，操作の記述を含まず，言語としては簡単で，いわば準言語（sublanguage）と呼ぶべきものである．

著者らが先に発表した，“Electronic Journal of Data Analysis”の構想（渋谷・柴田（1987））に関し，幸い統計数理研究所の理解も得て，共同研究（62-共研-6，63-共研-1，1-共研-6，2-共研-6，3-共研-14）として，継続的な検討，研究を行なってきた．本稿はその中間報告であり，またEJDAの実働化という目標に向けての出発点である．

2. D & Dの概略

設計の初期段階でまず議論する必要性のあったことは，基本的なデータ構造として何を採用するかであった．統計解析でよく現れ用いられる形式は行列で，各列が1つの変量に対応する，いわゆるデータ行列と呼ばれるものである．しかし，行列だけでは型の異なるデータを混在させられず，均整のとれていない構造のデータ，複雑な関係をもったデータを表現するには無理がある．行列より一般的なデータ構造が関係形式（relational scheme）で，データ・ベクトル（値の並び）の集まりとして構成する方式である．この場合，各ベクトル中の要素の型は同一でなければならないが，ベクトルの間では必ずしもそうでなくてもよい．原理的には，どのようなデータ・セットも，この形式で表現できることが知られている．いわゆるリレーショナル・データベースはこの関係形式に基礎をおいて構成されたデータベースである．

この形式を採用すれば，行列も，たとえば列ベクトルの集まりとして表現でき，行列の形式は特に必要でなくなる．しかし，こうすると行という概念がないので，行に関する記述を行なうには別にidとなるベクトルを追加して間接的に記述しなければならない．さらに，行列の行方向についての記述，たとえば和が1というような記述を行なうにも，行列の要素を1列に並べたベクトルとそれに対応するidのベクトルを作り，それを通じて記述しなければならない．いずれにしても，冗長で，人間にとって読みやすい形ではなくなることが多い．また，均整のとれた実験計画がなされた観測データの場合には，行列データの形にしておいた方が意味がはっきりすることも確かである．

そこで，いろいろな議論の末，基本的なデータの表現形式としては，行列，より一般的には「配列」と「関係形式」を両方併用することにした．ただし，行列であっても，値は列をつなげた1つのベクトルとし，行数，列数などの記述は記述部で行なうことにしたので，データ本体（Data.body）を見ただけではこの2つの形式が混在していることには気付かない．また，データ本体は，（名前付きの）ベクトルの集合であるが，各ベクトルの値は実数値のみとし，それ以外の論理型，カテゴリなどは，記述部でコーディング規則を記述することにして，型の均整化を計った．さらに，大量の外部データなどはD & D本体には含めず，外部定義とすることが可能な仕組みも取り入れた．

さて，記述部は大きく3部に分かれる．前書き（Preface），概説（Overview），データ記述（Data Description）の各部分である．本でいうと，これに本文としてデータ本体を加えたものが1冊の本，すなわち1つのD & Dとなる．前書きは表題（Title），日付（Date），著者（Contributor）などより成る．この部分は検索あるいは解析の参考になる事柄である．概説は実験計画あるいは標本設計（Design），ランダム化と標本抽出のメカニズム（Observation.mechanism），可能な解析手法（Possible.analysis）の記述より成る．この部分はデータの統計的側面を記述する部分である．データ記述の部分は構造記述（Data.structure）と，行列の軸に関する記述（Axis），データ本体の各ベクトルについての記述（Data）より成る．行列の形式の場合には，軸については対応するデータがデータ本体にはないので，別に記述する必要がある．構造記述はAxis, Dataの各要素の名前を引用することによって行ない，行列などもここで構造化されて初めてその姿を現す．

ここで，1つの具体的なD＆Dの例を眺めてみよう．次の例はCox and Snell (1981) のExample OをD＆Dとして書き直したものである．以下の章で与えるほとんどの例も，同様にして作成したD＆Dの一部である（Shibata et al. (1990b) を参照のこと）．

```
Cox.O=
(
  Title="Atomic weight of iodine"
  Date=(created="1988-11-30" modified="1990-01-27")
  Contributor=(investigator=("Baxter and Landstredt (1940)"  "Brownlee (1965)")
      assembler="M. Takagiwa, Keio Univ.")
  Research.field="chemistry"
  Purpose="Accurate determination of atomic weight of iodine from the ratios of reacting
      weight of iodine and silver, using five batches of silver and two batches of iodine."
  Source="O ; Cox and Snell (1981), Applied Statistics, Chapman and Hall"
  Explanation=("References: Armitage, P. (1971), Statistical Methods in Medical
      Research, Section 8.7, and Snedecor, G.W. and Cochran, W.G. (1967), Statistical
      Methods, Section 6.7"
      "Silver batch C in <silver> is a repurification of batch B,
      which in turn is a repurification of batch A.")
  Primary.model="Analysis of unbalanced data"
  Possible.analysis=(
      ANOVA=(
          explanatory=("silver" "iodine")
          response="weight")
                    )
  Data.structure=(
      ratios=(Long.name="Ratios of reacting weight of iodine"
          Columns=("weight" "silver" "iodine")
                  )
                  )
  Data=(
      weight=(Long.name="Ratio of reacting weight of silver and iodine"
          Transform=(
                RATIO=(denominator=1)
                LIN=(location=1.176399 scale=10E6)
                      )
                    )
      silver=(Long.name="Silver batch"
          Dtype="category"
          Code=("A" "B" "C" "D" "E")
                    )
      iodine=(Long.name="Iodine batch"
          Dtype="category"
```

```
        Code=("I" "II")
                )
        )
  Data.body=(weight=(23 26 42 42 30 21 38 50 51 56 0 41 19 24 14 62)
      silver=(1 1 2 2 3 3 3 4 4 5 1 1 1 2 2 4)
      iodine=(10*1 6*2)
              )
)
```

このように，D＆Dは特別な補足説明なしでもデータの背景，属性がわかるように構成されている．さまざまなキーワードについては以下で順次説明する．

以上で，D＆Dの全体像はだいたいおわかりになったことと思う．あとはこれを実際にどう具体的に表現するかが問題があったが，最終的な形としては「名前＝値」を基本的な要素とし，かっこで囲むことによりそれらの要素の並びをまとめ，ふたたび値として用いることのできる1つのリスト構造をもったLISPに似た形式とした．ここであえて，「名前＝値」の形式にこだわったのは，人間にも読みやすい形式の方が何かと都合がよいだろうという配慮による．並びの区切りとして，特にコンマを用いずに単に空白部（空白，復帰，改行，タブ，コンマ）としたのも同じ理由による．

このように記述部とデータ本体から成る1つの構造を1つのD＆Dと呼ぶ．一般に，D＆Dファイルは複数のD＆Dより成り，ふたたび，各D＆Dを値とする「名前＝値」の並びとなっている．このようなD＆Dファイルの先頭あるいは各D＆Dの直前には，計算機で読み込むときに必要となる基本的な条件，名前の文字列の最大の長さ，外部データの参照，さらには日本語の場合には用いられている漢字コードなどの情報を，規約（protocol）として付加する．

3. D＆Dの意義と価値

まず，データ収集の現場での役割である．データを収集するときに多かれ少なかれデータのさまざまな属性，背景情報を記録しておくが，その様式は必ずしも一定ではなく，必須項目といったものも特に定まっていないことが多い．計算機可読な形で入力されていたとしても，単なる説明文であることが多い．D＆Dはこれらについて1つの指針を与え，汎用な記述を可能にする．もちろん，D＆Dを人間が直接書くのではなく，それぞれの現場に応じたインタフェースを用意し，それを介して生成することになる．以上のような問題はデータ収集と解析が同一人物でない場合に顕著であるが，特に解析のコンサルティングといったことを考えた場合にはもっと切実な問題である．

データ収集の現場でのD＆Dの利用形態としては逆も考えられる．すでに生成されたD＆Dがあれば，あるいは，類似したデータ収集例を適当なライブラリから入手できれば，そこから諸条件を引き出し，「こういう目的で，このような解析を行ないたいときには，どういう条件でデータ収集を設計すればよいか」といったことが即座にわかり，後の解析に対して過不足のない記述も可能になる．これは解析のステージから収集のステージへのフィードバックである．この両ステージの間が両方向にスムーズに流れて初めて豊富な解析結果が生まれるわけであるが，D＆Dはこの流れを担う媒体となりうる．

D＆Dのもう1つの重要な役割は統計解析，データ解析の高度化の基盤としてである．これはEJDAプロジェクトの目標とも一致する役割であるが，1つのデータをさまざまな人間が解

析してみるということが，解析手法のみならず，理論面の進歩にもつながることは間違いない．これは科学としての基本要件である「追試ができる」点からも重要である．さらに進めば，データ解析の体系化，自動化が見えてくる．データとその記述を一体化し，構造化した，人間にも機械にも可読なD＆Dはこのような基盤としても大きな役割を果たすことができる．

4. D＆Dルール

以下この章ではD＆Dルールを各部分に分けて説明する．D＆Dルールはルールであると同時に記述すべき，あるいは記述しておいた方がよい項目を述べており，拡張可能である．今後の普及に伴なって項目をより豊富にする必要がある．

4.1 D＆Dルールの読み方

ルールを間違いなく述べるには，どうしても形式的な表現が必要となる．以下でそのような表現の規則(メタ文法)を説明する．さらに形式的な定義は字句解析プログラムlexと構文解析プログラムyaccにより記述されている．A：Bは「左辺のAをBで定義する」という意味であり，右辺で｜が用いられれば，排他選択を意味する．たとえば，A：a｜b｜cは「Aはa, b, cのいずれか」を意味する．

先にも述べたように，D＆Dの基本的な様式は，「名前＝値」を並べ，かっこで囲んでそれをふたたび「値」として用いて入れ子状にし，1つの構造としたものである．したがって，「値」はさらに入れ子になっていない終端の場合と，入れ子の中の非終端の場合がある．以下の説明では非終端の場合の値はイタリックを用いて区別している．「名前」は英字の単語で，大文字で始まる名前はD＆D処理システムがキーワードとして理解する名前であり，大文字だけの名前は解析システムが理解する名前である．すべて小文字の名前はユーザ定義の名前か，局所的なパラメータ名である．ユーザ定義の名前はData.structure, Axis, Data.bodyのいずれかの要素名として現れたもので，どこで定義された名前かを区別するため，それぞれsname, aname, dnameで表している．これらの名前は他の要素としては二重引用符で囲んで引用する．たとえば，Dataで定義された名前dnameは必ずData.bodyにも現れ，対応するデータ・ベクトルの記述を与える．なお，これらの名前の並びを表すのに，次のような名前を用いることもある．

anames：(”aname”...)
dnames：(”dname”...)
qnames：(qname ...)
qname：”dname”｜”aname”

ちなみに，qnameはquoted nameの略である．

「値」に現れる終端記号は，次の4つのいずれかである．

NUMBERS：(NUMBER｜数列 ...)
STRINGS：(STRING ...)
NUMBER：数値
STRING：二重引用符で囲んだ文字列

「数値」は通常の浮動小数表示で，指数部はDあるいはEで表す．数値としてはこの他に2文字NAも許される．NAはNot Availableの略で欠損値を表す．どのような欠損値であるかの詳細はData.structureのStypeを用いれば表せる．「数列」は演算子：または＊を用いた規則的

な数列である．m：nはmからnまで公差1または−1の数列，m＊nはnをm回くり返し並べた数列である．

4.2 D & Dの最上位の構造

D & Dの最上位の構造は以下の通りであるが，†印の付いた7要素だけが必須で，残りの9要素は任意である．

D & D：（
Title＝STRING†
Date＝*date*†
Contributor＝*contributor*†
Research.field＝STRING | STRINGS†
Keyword＝STRING | STRINGS
Purpose＝STRING | STRINGS
Source＝STRING | STRINGS
Explanation＝STRING | STRINGS
Primary.model＝*primary.model*

Design＝*design*
Observation.mechanism＝*observation.mechanism*
Possible.analysis＝*analyses*

Data.structure＝(*sname*＝*struct* ...)†
Axis＝(*aname*＝*axis* ...)
Data＝(*dname*＝*attr* ...)†

Data.body＝(*dname*＝*value* ...)†
）

TitleからPrimary.modelまでが前書き部，DesignからPossible.analysisまでが概説部，Data.structureからDataまでがデータ記述部，最後のData.bodyがデータ本体である．以下の節でこれらの各部を順に説明する．より厳密な，詳しい説明については，Shibata et al.(1990a)，あるいはEJDAシステムにある最新版を参照されたい．

4.3 前書き部

前書きの部分はSTRINGあるいはSTRINGSによる説明である（例外はDate，Contributor, Primary.model）．名前の示す通り，Research.fieldはこのD & Dに関連した研究分野や仕事などの記述，Keywordはキーワード，Purposeはデータ収集の目的，Sourceは，データが他の出典からの引用であるとき，その出典の記述である．Explanationは他の要素で形式的に表現できなかったことを自由文形式で記録しておくためにある．参考文献などもこの要素を利用して記述する．

Dateは日付であるが，履歴を記録するため，いくつかの日付の並びとなっている．

date：（collected＝STRING created＝STRING modified＝STRINGS）

日付は，ISO 8601 に従い，"yyyy-mm-dd" の形式の文字列で与える．Contributor もデータのいくつかの収集者と記述者が異なる場合を考慮し，いくつかの氏名（連絡先）の並びとなっている．

contributor : (investigator=STRINGS assembler=STRINGS)

Primary.model は解析の際のモデル構築のヒントになる事柄を与えるもので，文字列あるいは文字列の並びである．後述の Possible.analysis と異なり，この情報はソフトウェアよりは解析する人間のためであるので，特に形式，用語は定めない．以下はいくつかの例で，統計学の分野，トピックスに相当する．

primary.model : *model* | (*model* ...)
model : "accelerated test" | "classification" | "contingency"
| "dose response" | "factorial experiment"
| "factor analysis" | "functional relationship"
| "point process" | "sampling survey" | "survival analysis"

4.4 概説部

この部分はデータの統計的側面をなるべく形式的に記述することを目的とした部分で，Design, Observation.mechanism, Possible.analysis より成る．Design はいわゆる実験計画が行なわれた観測データに関する記述で，現在のところ以下のようなものが定義されている．この部分などは，データ収集の段階，あるいはその準備段階でぜひとも用意すべき部分である．

design: CR=(factors=*qnames* ftypes=NUMBERS
response=*dnames*)
| RB=(factors=*qnames* ftypes=NUMBERS
response=*dnames*)
| BIB=(v=NUMBER k=NUMBER b=NUMBER
r=NUMBER factors=*qnames*
ftypes=NUMBERS response=*dnames*)
| OA=(n=NUMBER m=NUMBER s=NUMBER
d=NUMBER factors=*qnames*
assign="*dname*" ftypes=NUMBERS
response=*dnames*)
| SPLIT=(factors=*qnames* ftypes=NUMBERS
response=*dnames*)
| NESTED=(factors=*qnames* ftypes=NUMBERS
response=*dnames*)

名前の示すように，順に完備型，乱塊法，釣合型不完備ブロック，直交配列，2 方分割，枝分かれ型の実験計画である．ftypes は因子の分類で 1 から 6 までの数字で示す．順に，制御因子，標示因子，ブロック因子，補助因子，変動因子，繰り返し因子である．最後の繰り返し因子は，通常，因子としては分類されないが，次の例でもわかるように行列の繰り返し軸を記述するために必要となるので，あえて因子の 1 つとして含めた．なお，名前 ftypes が後述の Stype, Atype, Dtype と異なりすべて小文字であるのは，D & D 処理システムとしては特別な扱いをしないからである．

例. (Cox and Snell; H)

```
Design=(RB=(factors=("process" "batches" "purity")
            ftypes=(1 3 4) response="fault"))
```

例. (Cox and Snell; J)

```
Design=(CR=(factors=("length" "amplitude" "load")
            ftypes=(1 1 1) response="cycles"))
```

例. (Cox and Snell; M)

```
Design=(BIB=(v=8 k=4 b=4 r=2
            factors=("factor1" "factor2" "block" "grade")
            ftypes=(2 2 3 4) response="count"))
```

例. (Cox and Snell; Q)

```
Design=(NESTED=(factors=("yarn" "bobbin" "samples")
            ftypes=(1 5 6) response="strength"))
```

例. (Cox and Snell; R)

```
Design=(SPLIT=(factors=("day" "sex" "treatment.a" "treatment.b")
            ftypes=(3 2 1 1) response="amount"))
```

Observation.mechanism は，実験計画の各因子のランダム化の記述と標本調査における任意抽出法の記述を行なう部分である．

observation.mechanism : (*obs* ...)
obs : RND=(which=*qname* given=*qnames*
from=*qname* rate="*dname*"
size=NUMBER | STRING replace=NUMBER)
| SYS=(which=*qname* given=*qnames*
from=*qname* rate="*dname*"
size=NUMBER | STRING start=NUMBERS
pattern=*patterns* | STRINGS)
| RNDCENSOR=(which=*qname* censor=*dname*)
| CENSOR=(which=*qname* censor.time="*dname*"
censor.number=NUMBER)
| POPULATION=(which="*dname*" frame=STRINGS
from="*dname*" size=NUMBER | STRING)
patterns : (*pattern* 1=NUMBERS *pattern* 2=NUMBERS ...)

上から順に，ランダム化のメカニズム，システマティックな項，ランダム・センサリング，センサリング，有限母集団の記述である．given を用いれば条件付きでランダムな項，条件付きでシステマティックな項などの記述も行なえる．

例. (柴田 (1989); 家計調査)

```
Observation.mechanism=(
    RND=(which="city" given="city.strata" size=1)
    RND=(which="block" given="city" size="unfixed")
    SYS=(which="unit" given=("start.year" "block" "city")
```

```
        size=”unfixed but maximum 12”)
    SYS=(which=”start.month” given=(”unit” ”start.year” ”block” ”city”)
        size=12
        pattern=(a=(1 1 1 1 1 1 7 7 7 7 7 7) b=(2 2 2 2 2 2 8 8 8 8 8 8)
        c=(3 3 3 3 3 3 9 9 9 9 9 9) d=(4 4 4 4 4 4 10 10 10 10 10 10 )
        e=(5 5 5 5 5 5 11 11 11 11 11 11) f=(6 6 6 6 6 6 12 12 12 12 12 12))
        )
    RND=(which=”household”
        given=(”classification” ”start.month” ”unit” ”start.year”
        ”block” ”city”) size=6))
```

例. (Cox and Snell ; Q)

```
Observation.mechanism=(
    SYS=(which=”yarn”)
    RND=(which=”bobbin” given=”yarn” size=6)
    RND=(which=”samples” given=(”bobbin” ”yarn”) size=4))
```

Possible.analysis は可能な解析を羅列する部分で，今のところ次のような解析法が定義されているが，今後ふやしていく予定である．

analyses : (*analysis*...)
analysis : ANOVA=(explanatory=*qnames* response=”*dname*”)
| LOG.LIN=(explanatory=*qnames* response=”*dname*”)
| LOGIS.LIN=(explanatory=*qnames* response=*binary.response*)
| REG=(explanatory=*qnames* response=*dnames*)
binary.response : ”*dname*”
| ”*aname*”
| (positive=”*dname*” total=”*dname*”)
| (negative=”*dname*” positive=”*dname*”)

上から順に，分散分析，対数線形，ロジスティック線形，回帰モデルである．

4.5 データ記述部

この部分は Data.body に与えられた各データ・ベクトルの属性を記述し，それらの間の関係を記述する部分である．Data の各要素は Data.body の要素と名前で 1 対 1 に対応し，それぞれの属性を記述する(4.5.3 節 ; データ記述)．Axis は行列の軸とその属性を記述する(4.5.2 節 ; 軸定義)．Data.structure は Data.body を構造化し，また複数のデータ・ベクトルの間の関係を記述する (4.5.1 節 ; データ構造)．

4.5.1 データ構造

Data.structure には関係形式の記述と配列の記述の 2 通りの形式がある．

struct : *str* | (*str*...)
str : (Long.name=STRING Columns=*dnames* Stype=*stypes*)
| (Long.name=STRING Axes=*anames* Value=”*dname*”
Stype=*stypes*)

前半が関係形式の記述で，Columns に Data.body の要素名を与えることにより，1つの関係を定義する．後半は配列の記述で，Axis で定義された要素名を Axes に与え，Value には Data.body の要素名を与えて，配列の値を定義する．いずれの形式も任意個定義してよい．

このようにして定義した関係，配列の Columns や Axes に与えられたデータ・ベクトルや軸の間の関係を Stype によって記述する．Stype には次のように構造をもった名前付きの並びを与える．

stypes : *stype* |(*stype* ...)
stype : SEQ=(which=*qnames* given=*qnames*)
| COORD=(x=*qname* y=*qname* z=*qname*)
| (r=*qname* theta=*qname*)
| (r=*qname* theta=*qname* phi=*qname*)
| (longitude=*qname* latitude=*qname* system=STRING)
| RADIX=(year=*qname* month=*qname*
day=*qname* hour=*qname*
minute=*qname* second=*qname*)
| (degree=*qname* second=*qname*
system=STRING)
| ASSOCIATION=(which=(*aname aname*)
from=*aname* to=*aname* what=STRINGS)
| PERM=(which="*dname*" given=*qnames*)
| TIME.SERIES=(time=*qnames* value=*qnames*)
| SUM=(which=*qnames* given=*qnames*
total=NUMBER | total="*dname*"
| out.of=NUMBER | out.of="*dname*")
| INTERVAL=(which=*qname* min="*dname*" max="*dname*")
| SUBSET=(from=*anames*)
| GRAPH=(vertex="*aname*" adjacent="*dname*")
| INDEX=(which="*dname*" of=*qname*)
| INVALID=(which="*dname*" where=NUMBERS | "*dname*"
what=NUMBERS | "*dname*" code=STRINGS)
| PRECISION=(which=*dnames* precision=NUMBER | "*dname*")
| TRUNCATION=(which="*dname*" where=NUMBERS | "*dname*"
left=NUMBERS | "*dname*" right=NUMBERS | "*dname*")
| *user.defined.stype*=(which=*qname*
where=NUMBERS | "*dname*" *anything*)

上から順に，データの順序に意味がある，座標系，基数系，連関表，並べかえ，時系列，和の制約，値が区間，値が集合，グラフ，添字であることを表し，INVALID 以下は，欠損値，精度，打ち切りの記述である．最後の *user.defined.stype* は新たな Stype の導入のために用意してある．

Stype と，後に説明する Atype あるいは Dtype との違いは，Stype はいくつかのデータ・ベクトルをまとめて初めて意味のある属性，あるいは属性を記述するのに複数のデータ・ベクトルを必要とするものの表現であるのに対し，Atype, Dtype は配列の各軸あるいはデータ・ベク

トル固有の属性である.

4.5.2 軸定義

Axis は配列の軸の定義と記述である.

axis : (Long.name=STRING Atype=*atype*
Unit=STRING Levels=NUMBERS | STRINGS)

Atype は軸の属性の記述で, "variates", "rank", "logical", "category", "ordered category", "count", "id", "sequence", "equispaced", "specified values", "interval class" のいずれかである. "variates" と "rank" が軸特有の属性で, 残りの属性は Dtype でも用いられる. "variates" はその軸が変量であることを示し, "rank" はその軸が嗜好などのような順位付けであることを示す. Unit は, Atype が "equispaced", "specified values", "interval class" のいずれかであるときに用いる要素で, その軸の単位を表す. Levels には水準を数字か文字列のベクトルで与える.

4.5.3 データ記述

最後の Data は Data.body の各要素の記述である.

attr : (Long.name=STRINGS Dtype=*dtype*
Length = NUMBER | NUMBERS
Unit=STRING Code=STRINGS
Range=*range* Transform=*transforms*)

Length には対応するデータ・ベクトルの長さ, あるいは配列の値の場合には各軸の大きさを与える. Dtype は対応するデータの型で, "source", "sorted", "logical", "category", "ordered category", "count", "id", "sequence", "equispaced", "specified values", "interval class" のいずれかであり, "score" と "sorted" 以外は Atype と共通である. "score" は成績のような1つの階級データであることを示し, "sorted" はデータ収集の段階ですでに整列化されたデータであることを示す. Unit は単位を表し, Code は対応するデータのコーディングを与える. Range はデータの値の可能な範囲を示す.

range : (min=NUMBER max=NUMBER) | NUMBERS | STRINGS

Transform はデータにすでに何らかの変換が施されているときに用いる.

transforms : (*transform* ...)
transform : LIN=(location=NUMBER scale=NUMBER
orig.unit=STRING)
| LOG=(base=NUMBER orig.unit=STRING)
| BOX.COX=(lambda=NUMBER orig.unit=STRING)
| CUT=(breaks=NUMBERS orig.unit=STRING)
| PW=(alpha=NUMBER orig.unit=STRING)
| RATIO=(denominator=NUMBER orig.unit=STRING)
| INDEX=(denominator=NUMBER orig.unit=STRING)

上から順に, 線形変換, 対数変換, Box-Cox 変換, 打ち切り, 巾乗変換, 比, 指数変換である.

4.6 データ本体

Data.body はきわめて単純で，単なる数値の並びである．

value : NUMBERS

値 *value* が省略されたときは，そのデータは外部データであることを示す．その存在はD&Dファイルの先頭の規約（protocol）で示されているはずである．

参 考 文 献

アドビ・システムズ(1988)．『PostScript リファレンス・マニュアル』(石田晴久 監修，松村邦仁 訳)，アスキー，東京．

Cox, D.R. and Snell, E.J. (1981). *Applied Statistics* : *Principles and Applications*, Chapman and Hall, London.

柴田里程（1989）．統計データとその属性記述の形式化，統計情報，6月号，18-24.

Shibata, R., Sibuya, M. and Takagiwa, M. (1990a). Data and description rule, Research Report, KSTS/RR-90/001, Department of Mathematics, Keio University.

Shibata, R., Sibuya, M. and Takagiwa, M. (1990b). D&D examples, Research Report, KSTS/RR-90/002, Department of Mathematics, Keio University.

渋谷政昭，柴田里程（1987）．電子ジャーナル "Electronic Journal of Data Analysis" の構想，統計数理，**35**，81-87.