まえがき

産業は「材」をめぐってさまざまな変遷を遂げてきた．農産物，水産物，林産物，畜産物などを材とする第一次産業，鉄や銅，石炭や石油を材とする第二次産業，物流やサービスを材とする第三次産業と変遷してきた産業も，ここで，データを材とする産業の出現によって新たな変貌を遂げることになる．データは質量ゼロで，ほとんどコストをかけずにコピーできるが，材として規格化するのは困難である．そこが，これまでの産業とは大きく性格を異にする．この新しい産業を支えるのが，データサイエンスであり，データエンジニアリングであることは一つの共通認識になりつつあるといってもよい．

しかし，すくなくとも今の日本は，データサイエンスを便利なバズワード (buzzword) として用いるものの，誰もその姿を一向にはっきりさせようとしない，情けない状況にある．たとえば，[17] では「データサイエンティストには科学者という含みはなく，高度なスキルを連想させて人々を惹きつけやすいからサイエンティストという言葉を使っているにすぎない」と言い切り，[24] では「今までの科学とは違う第４の科学」という逃げとも聞こえる言い方で済ましている．これは，[12] を含め「ビジネス上の必要に応じてデータを利用するパラダイムをデータサイエンスと称する」という T.H.Davenport and D.J.Patil [10] の提唱[1)]をそのまま引き継いでいるだけにしか見えない．それならそれで，**データ・サイエンティスト** (data scientist) と**データサイエンスティスト** (datascientist) の区別ぐらいは必要ではなかろうか．

この状況は，著者の学生時代に始まった「情報科学」の流れとよく似ている．その当時，東京工業大学で日本初の情報科学科設立に携わった先生方に「情報とはなんですか」，「情報科学とはどんな科学なんですか」と尋ねても，みな沈

[1)] データを利用する科学者をデータ・サイエンティストと呼ぼうというのが，彼らの提唱である．

黙するだけで何の答えも返ってこなかった．しかし，言語学者であった父にこの話をしたら，情報は「情けで得る報せ」で，要は「スパイの報告」のことなんだよと教えてくれた．すでにこのような説明は国語辞典からも消えているが，かといって情報という言葉の簡潔な説明は一向に見当たらず，辞典によってもその説明は異なる．つまり情報は相変わらずあいまいなまま便利に使われている言葉でしかない．これでは，情報科学が一つの科学分野として成立するのは望むべくもない．いまから思えば，コンピュータがまだ発展段階にあった当時，計算機科学やコンピュータサイエンスでは人々を惹きつけにくいと考え，情報科学というバズワードをつくりだしたに違いない．

しかし，データは情報よりも具体的な存在であり，枠組みさえ明確にすれば，データサイエンスは一つの有力な科学として成立しうる．著者らはこう信じて20 年以上に渡り「サイエンスとしてのデータサイエンス」を追い求めてきた．「なぜ」という素朴な疑問から出発し，対象に関する原理や原則を明らかにしていくのが**科学** (science) の本質だとしたら，そのような原理や原則をできるだけ汎用にまとめ上げたものが「理論」である．理論あってこその科学であり，バズワードの呪縛から逃れ，着実な進歩を重ねるためにもなくてはならない存在である．ひとつのパラダイムとして独立するには，理論という核なしでは済まないことは，他の科学をみれば明らかであろう．

このような問題意識から，データサイエンスの理論と実践をできる限りわかりやすくまとめたのが本書である．といっても堅苦しく構えないでいただきたい．データを活用する上で押さえるべき基本といった程度のことであり，題名を「データサイエンスの作法」としたのもそんな思いからである．

作法 (practice) は失敗しないために必要な心得である．それが食事の作法やお茶の作法といった一つの形式として独り歩き始めると，本来の目的は忘れられ，ただやみくもに習得すべき**所作** (manners) としてしか理解されなくなってしまう．これは不幸でしかない．本書で述べるデータサイエンスの作法が，今後このような所作に陥ることがないよう願うばかりである．

本書が，本格的なデータサイエンスの理論展開にあたっての一つの出発点として役立つだけでなく，明確な枠組みと共通な言葉を確立することで，データサイエンス実践におけるコミュニケーションを円滑にし，目標に到達する道を

すばやく見出し効率よく進むのにも役立つならば望外の幸せである．執筆にあたっては，大学の初年度での教科書として，あるいは企業研修でのテキストとしても使えるよう配慮した．

本書では，データサイエンスの理論と実践との橋渡しとしてTRAD [2)]を用いる．エンジニアリングが，できる限り，人間ではなく機械に任せようという営みなのに対し，サイエンスはすべからく人間の営みである．そのため，データのサイエンスは，人間が，さまざまな思い込みから解放され，データを的確に把握しようと思わない限り始まらない．もちろん，データはそのままでは無味乾燥な存在であり，数字や記号の並びを眺めていても退屈なだけである．人間の直観を呼び覚ますなんらかの仕掛けがない限り，暗中模索の旅を続けるしかない．TRAD はそこに一つの有力な答えを与える．どんなデータでも一枚の視覚表示によって人間の直観を呼び覚ませば活かし切ることができる．そんな環境が TRAD であり，長年のデータサイエンス研究の成果でもある．

TRAD はまたプレゼンテーションの道具としても優れている．他人とのコミュニケーションの場では,「そのまま見せても退屈」というだけの理由で，データについては外形的な説明に留めてしまうことが多い．しかし，これがコミュニケーションギャップの始まりである．いわばデータが宙に浮いた状態で，何等かの結果なり結論だけが報告される．そこにデータの誤用や不正の余地も生まれる．TRAD はもとのデータをそのまま視覚表示し，さまざまな属性や背景も即座に読み取れるように工夫されている．このような視覚表示で，参加者にデータの全体像を見せることからプレゼンテーションを始めれば，すぐ議論が沸き起こるに違いない．そこには誤用や不正の余地もない．

視覚表示を介したデータとの対話で必要となる操作が，変量のフィルタリングと記録のフィルタリング，そして型などによる表示形式の変更である．これらは，それぞれ，**視点**，**視野**，**視覚**を変えることに相当し，データブラウジングの三大要素を構成する．

また，何らかの結論に至るまでにデータはさまざまな変容を遂げる．その過程をトレースできることも重要である．TRAD はその機能を備え，視覚表示で

[2)] `http://datascience.jp/TRAD.html` より自由にダウンロードできる無償のソフトウェアである．その詳細については 2.1 節などを参照されたい．

容易にトレースできる．さらには，高度な解析やモデルの当てはめをおこなう必要が生じることもあるだろう．そのようなときには，データ解析ソフトウェア R との切れ目ない連携機能もおおいに役立つ．

本書は，著者のこれまでの活動で遭遇したさまざまな課題をもとに，いくつかのデータ例を中心に話を展開する．しっかりした理論に根差した実践の力強さを実感していただきたい．たまったデータの活用に悩んでいる方，データの活用を試みたものの途方に暮れている方，データサイエンスの実体が見えず当惑している方など，データを巡るさまざまな悩みを抱える方々の一助になれば望外の幸せである．本書を読み，作法を身に着けることで，少しでも悩みを解消していただきたい．実際に TRAD を使いながら読み進めていただくと，なお効果的である．

本書は拙著『データ分析とデータサイエンス』[21] の姉妹書の性格も有している．著書 [21] は高等学校の必修科目「数学 I」の副読本の性格をもった第 I 部から始まり，データサイエンスの姿を明らかにする大学レベルの第 II 部へと続く構成で，数学が苦手の方には少しとっつきにくかったかもしれない．そこで，本書では，数式を用いずに，データサイエンスの理論と実践の全体像が理解できるように配慮した．本書の内容で不満の残る読者には，[21] も一読されることをお勧めする．

島津秀康君と力丸佑紀さんには原稿を読んでいただき貴重なご意見をいただいた．ここに深く感謝する．また，近代科学社の小山透氏には，いつもながら，遅筆の小生を辛抱強く励まし，本書の題名をはじめとして数々の有用な助言をいただいた．この卓越した編集者なしでは，本書が日の目をみることはなかったに違いない．感謝しても感謝しきれない．

2020 年 9 月　　柴田里程

本書のカバーには，[21] の『いまきたこの道』と『富士夕景』に引き続き，中神潔氏の油彩『無題』と『赤い月』を使わせていただいた．横浜の山下公園がモチーフで，『いまきたこの道』の少女もすっかり大人である．

目 次

データ例

アイリスデータ 視覚表示（第 2 章）→ R での「種」の判別（8.4.1 節）

脊柱後弯症データ 変量のフィルタリング（3.1 節）→ R でのモデル化（8.4.2 節）

車の評価 変量のフィルタリングと記録のフィルタリング（3.3 節）→ R での線形モデルのあてはめ（8.4.3 節）

賃貸集合住宅の満足度 記録度数型への変更による視覚表示の変化（4.5 節）

大学ランキングデータ Web からのデータ取得と読込み（5.1 節）

宿泊旅行調査第 1 表 Excel ファイルの読込み（5.2.1 節）→ 射影（6.2.1 節）→ 正規化（7.1.1 節）

宿泊旅行調査第 2 表 Excel ファイルの読込み（5.2.2 節）→ 射影（6.2.2 節）→ 正規化（7.1.2 節）

患者調査 複数の表を含む csv ファイルの読込み（5.3 節）→ 正規化（7.1.3 節）

LIBOR データ 表を転置しての読込み（5.4 節）

民力データ 射影（6.1 節）→ R での相関係数のチェック（8.4.4 節）

実験データ 視覚表示と R による要因探索（8.4.5 節）

呼吸量心拍データ 共通変量と同等変量（あとがき）

第1章　資料，情報，データ

日常生活では，資料，情報，データの違いなどあまり意識せずに済んでしまうが，その活用となると，また話は別である．何をどうしたいのかあいまいなままでは，活用どころか，議論すらままならない．

資料 (document) は，本棚の本，書類の山，たまったレシートなど，なんらかの記録の集まりで，まずは存在することに意味がある．電子化された資料であっても，とりあえずスキャンした書籍など，その時点では明確な利用目的がなくても必要になるかもしれないということで保管 (archive) されているのが資料であるといってもよい．

情報 (information) は，なんらかの目的をもってまとめられた資料である．そこには，それを役立てたいという明確な意思が存在する．このような情報の塊を日常的には「データベース」と呼ぶこともあるが，正確には「情報ベース」とでも呼ぶべきものであろう．

データ (data) は「記号で表された，推論の根拠となる事実の反映」で，資料や情報より具体的な存在であり，推論の根拠という目的も明確である．「推論の根拠」を抜きにして「記号で表された事実の反映」というだけなら，資料や情報もデータになってしまう．もちろん，推論の根拠としたい場合でも，まだ推論の方向性がはっきり定まらないため，厳密にはデータと呼べるかどうか微妙なケースもある．たとえば，**テキストマイニング** (text mininig) では，このような「記号で表された事実の反映」に現れた単語を手掛かりに，なんらかの事実を浮かび上がらせようとするが，その段階では推論の方向性がはっきりしていないことのほうが多い．テキストマイニングをおこなって初めて推論の方向性が浮かび上がり，「記号で表された事実の反映」を何らかの形に整えた上で，その文脈や背景を突き詰める作業に進むのが普通である．したがって，その段

第8章　Rとその利用

8.1　R

ここまでお読みになれば，データの視覚表示によって人間の直観をうまく呼び覚ますだけでも，かなりの事がわかってくることを実感されたに違いない．もちろん，いつもそれで済むわけではない．おかしな値を具体的な数値で確かめたくなることもあるだろうし，対数変換などを始めとするさまざまな変換をデータに施すことではじめて見えてくる姿もある．また，ヒストグラムや散布図，時系列図といった古典的なグラフィック表示を得たいこともあるだろう．

すでにおわかりのように，視覚表示からわかることは，どうしても「… であろう」とか「… を示唆している」といった感触に近いことが中心になる．実際にはそれで十分なことも多いが，さらに検証が必要なことも多い．よく知られたモデルを当てはめてみたり，仮設[1)]検定をおこないたいこともあるかもしれない．

そこで，TRAD では，次のステップへ進むための道として， R とのシームレスな連携機能を用意している．名前からもわかるように， TRAD は当初から R を利用すれば済む処理や機能は R に任せ，視覚表示を中心にした人間の直感を呼び覚ます機能を**TAD**(タッド)(TextilePlot and DandD) [2)] が担うといったハイブリット構成を基本に開発してきた．

すでに 1.3 節でも触れたように，前身となる S から数えれば 40 年近くの歴史を持つ R は，そのシンプルな構成と汎用な設計が相まって，広く用いられるようになり，また，ユーザ作成の多くのライブラリが加わることで，いまやデータ解析にはなくてはならない道具になっている．著者自身，S の開発に当

[1)] ここで仮説ではなく仮設を用いている理由については [21] を参照されたい．
[2)] TRAD から R を除いた残りを TAD と呼ぶ．

初からかかわり，R も含め，これまで教育や研究，さらには業務に活用してきた経験から，その便利さ，強力さは身に染みてわかっているつもりである．

しかし，R の使用にはいつもある程度のストレスが伴う．R のヒューマンインタフェースは，すべてをコマンド行の入力で進める **CUI**(character user interface) で，マウス操作だけでほとんどが済む **GUI**(graphical user interface) に比べれば，習熟が必要で，操作の全貌も見えにくく，データの全体像もつかみにくい．R での作業は細かい操作の積み重ねであり，そのときは全貌をつかんだつもりでも，少し時間が経つと，何をやっていたのか思い出せなかったり，他のデータに同じ操作を施そうとしてもうまく再現できなかったりする．それを再現性のある形で人に伝えるのはさらに容易ではない．R を駆使したデータ解析はどうしても個人的な職人芸になってしまい，プレゼンテーションも結果だけの説明に終始しがちになるのも当然である．

もちろん，CUI の利点は小回りのきくところにある．R なら，関数を書き換えることで操作をカスタマイズすれば，いくらでも自分にとって使いやすい環境を整えられる．それだけでなく，いつも先を考えながらコマンド行を入力することが，自然にデータやその背後にある現象への洞察を深めることになることも見逃せない．これがクリックの繰返しだけでは得られない CUI の利点である．GUI の TAD と CUI の R をうまく組み合わせれば，両者の利点を活かした理想的な環境ができるに違いないという信念が TRAD 開発の裏にはある．

8.2 R のインタフェース

R を利用する上でのストレスは CUI だけではない．すべてを **R 式** (R formula) [3] の評価という形で完結しようとした S の時代からの基本的な設計方針にもその原因がある．たとえば，**R オブジェクト** (R object) [4] の編集ですら，関数 `edit` 呼出しの副作用として立ち上がるエディタを利用するようになっている．これは，設計方針としては一貫していて，それなりに美しいが，編集結果に構文エラーなどがあれば，関数 `edit` を呼び出し直し，エラーの起きた箇

[3] R の構文に従って作られた式を R 式という．
[4] R 上の，関数，データなどすべてのものをまとめて R オブジェクトと呼ぶ．

所を探すことから始めなければならない．当然，編集の終了まで R の操作は一切できず，小回りはきかない．また，現在どんな R オブジェクトが作られているのかも，作業を振り返る意味で常時チェックしたいが，そのたびに関数 `objects` あるいは `ls` を呼び出す必要があるのも慣れないと面倒である．

Python

ディープラーニング（deep learning, 深層学習）を簡単に試せることもあって，プログラミング言語 Python の利用が広まっている．当然，学習データの処理や解析もすべて Python で済ませてしまいたくなる．そのため，データ解析には Python か R かといった疑問も生まれてきた．Python は，R と同じ対話型のプログラミング言語なのでよく似てはいるが違う．違いを一言でいえば「コンピュータ寄りのプログラミング言語」か「データ寄りの解析環境」かである．Python は「簡単に汎用なプログラムが作成できる」が売りであるのに対し，R は「簡単に汎用なデータの扱いができる」が売りである．したがって，どちらに重きを置くかで選択も変わってくるに違いない．最近では，Python と R を必要に応じて使い分けるハイブリッドな使い方もできるようになってきたが，いずれもユーザインタフェースが CUI のプログラミング言語であることには変わりがない．データを総体的に理解する環境として構築してきた TRAD は，GUI では対応しきれないさまざまな計算やモデルの当てはめなどを，データを中心に据えた R に任せることで補完している．

これらは，RStudio という額縁をダウンロード（無償と有償あり）し，インストールすることである程度までは改善される．それでも，R オブジェクトリストや履歴リストが常時表示されるようになったり，パッケージの一覧管理といった機能が加わるだけで，額縁の中身である R のユーザインタフェースはほとんどそのままである．日本語の入力が不自然だったり，行列やデータフレームの表示で，日本語が含まれていると列が乱れたりするといった問題は解決されておらず，R オブジェクトエディタの貧弱さも変わらない．

そのためであろうか，RStudio では，R オブジェクトの編集はあまりおこなわず，ほとんどすべての操作を R スクリプトで記述し，それを一挙に実行させる**バッチ処理** (batch processing) [5]的な使い方が主流になっているようである．しかし，そうなると，どうしても定形的な処理が中心になってしまい，データの本当の姿がとらえられているのかどうか，その保証もないまま満足するしかなくなる．データの単なる利用ならばこれでも済むかもしれないが，活用となると難しい．少しでも処理が入り組んだら，R スクリプトの**デバッグ** (debug) だけでも大変であるし，効率が悪い．

何よりも，せっかくの R の良さが生かされない．データ解析の本質は**さまざまな方向からデータを探り試行錯誤を繰り返すことにある**という認識にもとづいて開発されたのが S であり R である．それは 40 年近く経った今でも変わらない．データの利用から活用に進もうと思ったら，バッチ処理的な使い方だけで済むわけがないことは，読者諸兄姉はすでに十分おわかりのことに違いない．

8.3 TAD と R

TRAD では，R を一つの計算エンジンとして位置づけ，独自のユーザインタフェースを用意している．ユーザ作成の R オブジェクトを常に一覧できるオブジェクトパネル，履歴を有機的に利用できる履歴パネルのほか，本格的なエディタも組み込まれている．このエディタは R オブジェクトの編集やデバッグの快適な環境を提供するだけでなく，R スクリプトの作成や編集もでき，R オブジェクトの一つである **構文解析済みオブジェクト** (expression object) として保存するので，R スクリプトを別途保存する必要はなく，散逸も防げる．

TAD と R は JRI (Java R Interface) で結び付いているので，ユーザは，その境目をほとんど意識することなく，図 8.1 のように TAD と R の間を自由に行き来できる．TAD から R へのデータの受け渡しは比較的簡単である．データテーブルを R のデータフレームとして受け渡せばよい．しかし，問題は逆向きで，R のデータフレームの属性は圧倒的に少なく，ほとんど裸に近い形の

[5] コンピュータが高価な時代に，非対話型に一括処理することで効率化を図ったのがバッチ処理である．

索 引