

# 大学新入生の数値資料処理技能の現状を踏まえた 集計・図示支援用scriptの原型作成の試み

An Attempt Based on the Present Conditions of University Freshmen  
to Offer a Prototypic Script for Data Analysis and Visualization

加 藤 厚

まず、2008年度の基礎演習 I における実例と総合大学における実態調査の結果を踏まえ、「情報」既習の大学新入生における数値資料の要約・分析技能の不十分な修得と活用の現状、及びその 1 要因としての表計算ソフトの多機能性、そして「単純な単機能tool」による改善の可能性などが指摘された。

続いて、使用方法 (interface) がブラウザ (web browser) と共通で、処理結果 (作図用data) から文書処理ソフト上での自動作図が可能な「汎用分析支援tool」の原型としての分布集計・図示用scriptの検討と作成が、JavaScriptを用いて行われた。

作成された原型toolとJavaScriptの特長、及び今後の展開の可能性のいくつかを指摘して本論文の結びとした。

**キーワード**：大学新入生、情報、集計、図示、interface、JavaScript、表計算、文書処理

## 目 次

### I 問題と目的

- 1 文系大学生などにおける数値資料の要約・分析技能の現状
- 2 現状の規定要因としての「適切な支援toolの不在」
- 3 適切な支援toolの諸要件
- 4 本研究の目的

### II 方法と結果

- 1 支援toolの作成手段としてのJavaScriptの特長
- 2 入力方法の検討・決定と実装
- 3 平均値の算出・提示方法の検討と実装
- 4 階級別分布などの算出・提示方法の検討と実装
- 5 自動作図に必要なdata出力の検討と実装
- 6 自動作図の方法

III 考察  
文献

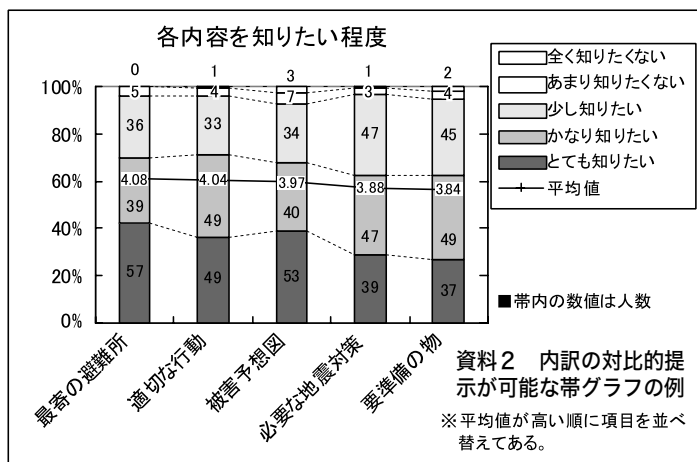
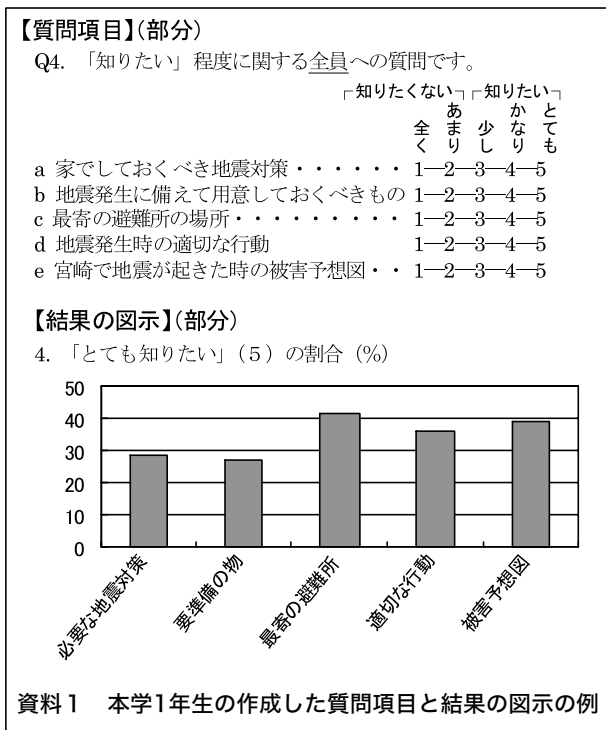
I 問題と目的

1 文系大学生などにおける数値資料の要約・分析技能の現状

資料1の内容は、本学の新生対象科目「基礎演習I」において、ある班が作成した質問紙の項目、及びその集計結果の図示の一部である。

設問と選択肢の内容・形式、図示の一般的技能などに関しては大学新生の平均的水準を満たす一方、回答として「知りたい程度」(=量的変数)を5段階尺度で把握しているながら、結果の図示においては5を選択した割合のみを提示している点、言い換えれば5以外の選択状況が不明である点が改善の余地として指摘できる。

本来、内訳を対比的に提示して各項目の特徴に関する理解を深めるためには帯グラフの利用が最適であり、また全体傾向の体系的提示には5段階尺度の平均値を基準として項目を並べ替えた整序提示が有効である。そして、これらの原則に従った資料2からは、「『とても知りたい』も多し一方『全く／あまり知らない』も比較的多い」という「被害予想図」の特徴が把握できる。しかし、高等学校で「情報」を履修してきた大学新生においても、数値資料の適切な図示によって最適な分析と判断を行うこ



これらの知識・技能の活用が必ずしも期待できないことは、資料1の例が示すとおりである。

加えて、看護士を対象に想定した統計入門書(田久・小島 2006)における「帯グラフで内訳を比較すべき集計結果を円グラフで列挙した結果、理解・説得が困難な状態に留まってしまう」(第3章)という、資料1に類似したエピソードの紹介を考慮するなら、「既有的知識・技能の不十分な活用」という状況は、必ずしも文系大学生に限定されず、より広い対象において存在することが危惧される。

2 現状の規定要因としての「適切な支援toolの不在」

「既有的知識・技能の不十分な活用」という状況の要因の1つとして、一般の学生・生徒などが容易に活用できる数値資料の要約・分析支援toolの不在が指摘できる。

資料3に示したとおり、2003(平成15)年度から実施されている現行学習指導要領では、高等学校の必修科目である「情報」において表計算/データベースソフトウェアの活用が求められている。

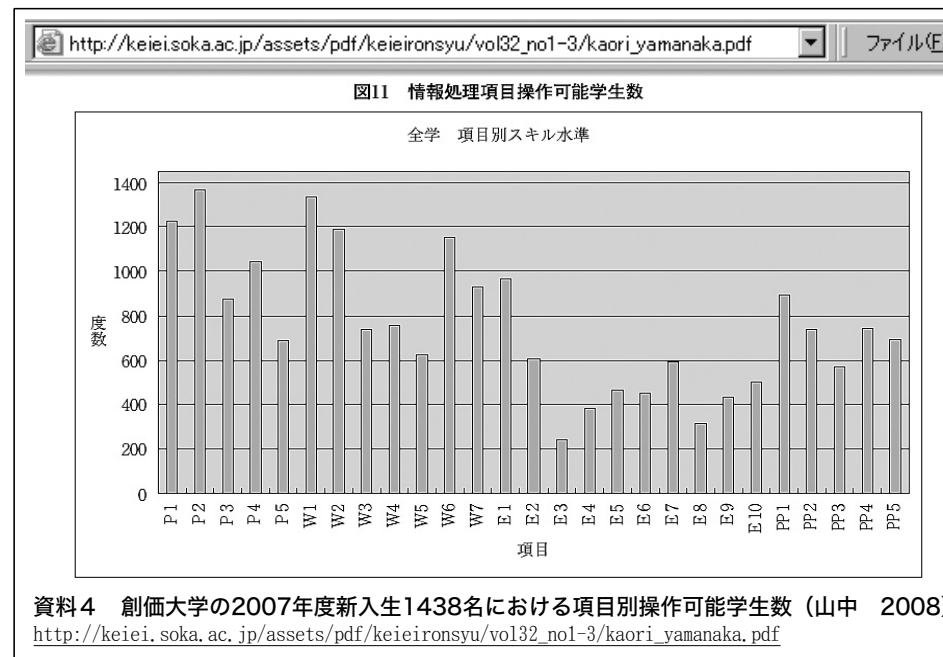
しかし、1私立総合大学の2007年度新生(経済・経営・法・文・教育・工の計6学部1438名:

**情報A 3(3):**  
(略)多様な形態の情報を統合的に活用することが必要な課題を設定し、文書処理、表計算、図形・画像処理、データベースなどのソフトウェアを目的に応じて使い分けたり組み合わせたりして活用する実習を中心に扱うようにする。

**情報B 2(3)イ:**  
情報を蓄積・管理するためのデータベースの概念を理解させ、簡単なデータベースを設計し、活用できるようにする。

**情報C 3(3):**  
(略)適切な題材を選び、情報の収集から分析・発信までを含めた一連の実習を中心に扱うようにする。情報の分析については、表計算ソフトウェアなどの簡単な統計分析機能やグラフ作成機能などを扱うようにする。

**資料3 高等学校学習指導要領 情報 における「表計算」・「データベース」などへの言及状況(文部省 1999)**  
[http://www.mext.go.jp/b\\_menu/shuppan/sonota/990301/03122603/011.htm](http://www.mext.go.jp/b_menu/shuppan/sonota/990301/03122603/011.htm)



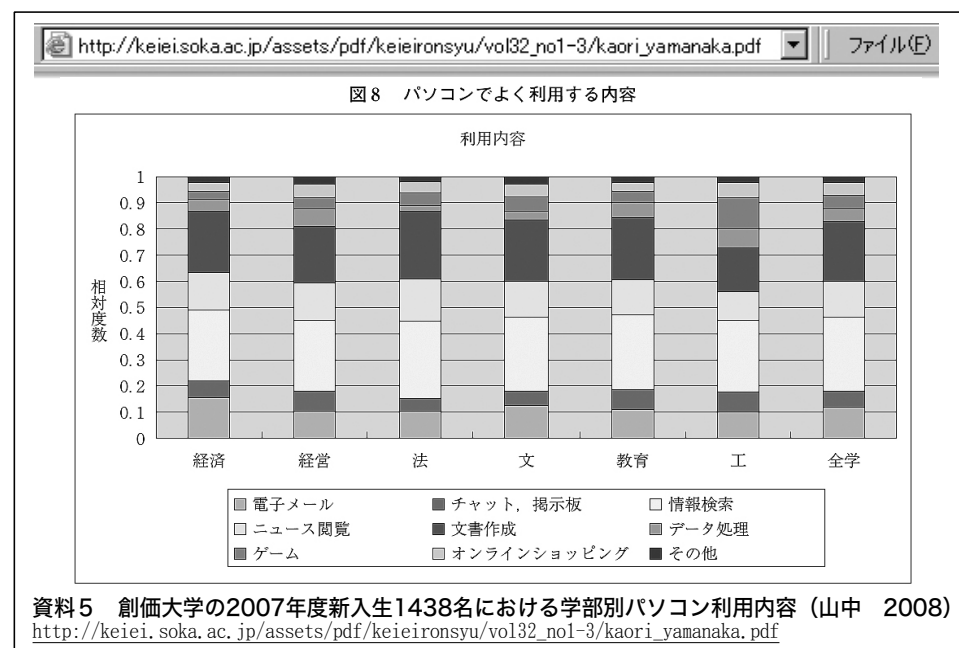
男性736名・女性702名)を対象とした調査結果(資料4)によれば、より基本的な表計算ソフトについても「Excelでは逆に半数を超した項目がE1『データの入力・編集』67.2%のみであり、その他の項目は全て半数以下の学生しかできない」(p.12)という低い技能修得状況が把握・報告されている(山中 2008)。なお、より高度なソフトウェアであるデータベース(DBMS)の修得状況については、同調査では取り上げられていない。

Excelに関する具体的な技能修得状況は、「できる」の値が高い順に「E1:データの入力・編集(文字データ, 数字データなど)」が67.2%、「E2:計算式の作成(加減乗除の記号の使用など)」が42.3%、「E7:表の印刷」が41.2%であり、以下「E10:ページ設定」が34.8%、「E5:表の編集(列や行の挿入・削除, 書式の変更など)」、「E6:関数を使った計算(合計SUM, 平均AVERAGEなど)」、「E9:グラフの作成・編集」、「E8:ワークシートの操作」が30%程度、そして「E4:オートフィル(連続データ, 曜日の作成など)」、「E3:絶対参照の計算式の作成(構成比の計算など)」については20%程度に留まっている。

従って、平均的な高校生～大学新入生に期待できるExcelなどの表計算ソフトの操作は「データの入力・編集」と「変数間の四則演算」・「印刷」程度までであり、その機能を十分活用した適切な分析と図示の遂行を期待するには、データベース(DBMS)は言うに及ばず、Excelに代表される現行の表計算ソフトも「複雑かつ多機能に過ぎる」ようである。

### 3 適切な支援toolの諸要件

多様な目的のために使用可能であることを追求した結果、多彩かつ高度な機能を完備するに到った現行の表計算ソフトが、皮肉にもその多機能性のあまり大学新入生などに使いこなせないの



なら、彼らにおける適切な処理遂行の支援にはむしろ機能を限定した単純な「単機能tool」の使用に一定の有効性が期待できる。そして、初心者利用を想定する場合、そのようなtoolには下記の諸要件が期待される。

- a. 処理内容の理解が容易な単機能toolであること
- b. 利用するために必要な新知識・技能が極力少ないこと
- c. 処理結果をレポートや発表用資料作成に容易に活用できること

ここで、大学新入生におけるパソコン利用内容の分布に基づいて彼らのソフトウェア利用状況を検討すると、資料5に示されているように、彼らが最もよく利用しているソフトウェアはブラウザ(情報検索, ニュース閲覧, チャット, 掲示板, 及び電子メールの一部などに使用)で全体の約5割を占め、次によく利用している文書処理ソフト(文書作成に使用)の2割強を加えると、これらの2つで全利用の約7割を占めている。従って、ブラウザと同様のinterface(使用方法)で数値資料の基本的整理・要約が行え、かつその処理結果を文書処理ソフトなどで利用可能なtoolを提供できるなら、それは一般の学生・生徒などにとってより現実的な意味で有効な支援となることが期待できる。

### 4 本研究の目的

以上の検討を踏まえ、本研究では「ブラウザと同様の使用方法(interface)で数値資料の整理・要約と図示が行える単機能tool」の原型として、一列分の数値入力を処理してその階級別分布と%, 平均値などを算出・提示し、併せて帯・円・折れ線グラフなどの自動作成に必要なdataを出力する支援toolの作成を目的とする。

## II 方法と結果

### 1 支援tool作成手段としてのJavaScriptの特長

今回の試みにおけるtool作成の手段としては、下記の3点の特長を有するJavaScript(資料6参照)を用いることとした。

資料6 ブラウザ用標準script言語としてのJavaScriptの解説(インセプト 2005) <http://e-words.jp/w/JavaScript.html>

- a. 本来、ブラウザ用のスクリプト言語であり、そのinterfaceはブラウザと同一である。
- b. 分析機能を備えたdata archiveの開発実績がある（加藤 2005）。
- c. 実体がテキストファイルであり、改良が容易に可能である。

## 2 入力方法の検討・決定と実装

JavaScriptによる直接読み込みが可能な外部ファイルは「外部JavaScript file」のみであり、加藤（2005）においては標準的汎用data書式であるCSV file（カンマ区切りファイル）を2次元配列の外部JavaScript fileに変換するtoolが作成・提示された。しかし、資料4の技能修得状況に基づけば、大学新入生などにおいては、CSV fileの理解と作成自体が容易ではないことが推測される。他方、表計算ソフトを用いた「データの入力・編集」については、彼らの7割近く（67.2%）が修得している（資料4）。そこで、今回作成するtoolでは、「テキストエリア」(textarea)への表計算ソフトからのcopy & pasteを数値資料の主な入力方法とした。

資料7は、上記を実現するscript（00.htm）の内容とその実行結果である。まずtitle部分（<title>〜</title>）に内容を説明する「題目」を、次にform部分にブラウザで表示される3つのテキストエリア（文字列入出力欄）とボタンを、そしてscript部分に処理の内容を記述している。


Formは記入用紙、その中のtextarea（テキストエリア）は複数行用入出力領域、buttonはclickされるとonClickに続くfunction（関数／機能：資料7の例では“処理（）”）が実行されるボタンであり、各textareaにはその行数(rows)と列数(cols)、及びmouseover（pointerが重なる）時にpopupされる説明(title)が追加指定されている。また、“入力欄”には初期値を提示して入力形式を例示するとともに、数値の変更・追加などによる機能確認の用に供している。

JavaScriptの本体はscript部分のfunctionであり、資料7ではその1行目で、変数「数値資料」への最初([0])のformの最初([0])のelement（要素：例では左のtextarea）の値（value=内容）である文字列「性別[改行]1[改行]2[改行]1[改行]2[改行]」の代入、2行目では2つ目のelement（＝

**【Script】**

```
<title>Pasteされた列をそのまま出力(仮資料付)</title>
<form>
<textarea rows="8" cols="8" title="入力欄">
性別
1
2
1
2
</textarea>
<textarea rows="8" cols="8" title="出力欄"></textarea>
<textarea rows="8" cols="8" title="不使用"></textarea>
<input type="button" value="実行" onClick="処理()">
</form>
<script>
function 処理()
{
  数値資料=document.forms[0].elements[0].value;
  document.forms[0].elements[1].value=数値資料;
}
</script>
```

**【実行結果】(仮資料のままの場合)**



資料7 Pasteされた1列分の数値資料をそのまま出力するJavaScript及びI.E.6.0 on Windows XPでの実行結果

中央のtextarea)の値への変数「数値資料」の値(=文字列)の代入が記述されている。

実行結果は資料7に示したとおりであり、第1要素(elements[0])に入力された値(文字列)がそのまま第2要素(elements[1])の値として出力(表示)される。なお、Internet Explorerで実行した場合の「アクティブコンテンツのブロック」は、title部分の前に<!-- saved from url=(0008)about:internet -->などの1行(mark of the web)を挿入することで回避可能である（資料17参照）。また、Firefox、Safariなどではブロックは行われない。

## 3 平均値の算出・提示方法の検討と実装

1列の数値資料、つまり一変数の整理・要約において直観的に最も有意義な処理結果である「階級別度数分布の算出と図示」への中間段階として、本節では平均値を算出するscriptを検討・実装する。

平均値(=総和÷件数)の算出・提示には、下記の各処理が必要であり、資料8はそれらを実現するscript（01.htm）の内容とその実行結果である。

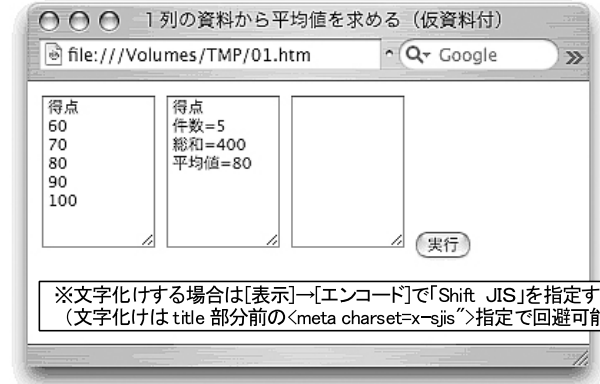
- a. Pasteされた“一塊”の文字列を「変数名」(1行目)と各数値に分割する。
- b. “存在する数値”を順に「総和」に加算し、同時に「件数」を1ずつ増加させる。
- c. 「総和」÷「件数」の値を出力欄の変数に代入する。

Script部分の第1行は、valueの要素中の全て(g=global)の¥r¥n(CR+LF:Windowsにおける改行)の¥n(LF:UNIX・Mac OS Xにおける改行)への置き換え(replace)、及び¥nによる

**【Script】**

```
<title>1列の資料から平均値を求める(仮資料付)</title>
<form>
<textarea rows="8" cols="8" title="1列の資料をpaste">
得点
60
70
80
90
100
</textarea>
<textarea rows="8" cols="8" title="処理結果を出力"></textarea>
<textarea rows="8" cols="8" title="不使用"></textarea>
<input type="button" value="実行" onClick="処理()">
</form>
<script>
function 処理()
{
  資料=document.forms[0].elements[0].value.replace(/¥r¥n/g,"¥n").split("¥n");
  件数=0;総和=0;
  for (var i in 資料)
  {
    if (0==i){変数名=資料[0];continue;}
    if (" "==資料[i]){continue;}
    総和=総和+parseFloat(資料[i]);件数++;
  }
  平均値=Math.round(1000*総和/件数)/1000;
  総和=Math.round(1000*総和)/1000;
  出力=変数名+"¥n 件数="+件数+"¥n 総和="+総和+"¥n 平均値="+平均値;
  document.forms[0].elements[1].value=出力;
}</script>
```

**【実行結果】(仮資料のままの場合)**



※文字化けする場合は[表示]→[エンコード]で「Shift JIS」を指定する。  
(文字化けはtitle部分前の<meta charset=x-sjis>指定で回避可能)

資料8 Pasteされた1列分の数値資料から平均値などを求めるJavaScript及びSafari 3.1 on Mac OS Xでの実行結果

分割(split)とその内容の「資料」への代入である。全ての¥r¥nの¥n置き換えにより改行の統一がなされ、また¥nによる分割によって「資料」は配列化され各要素には0から始まる連番が付けられる。

第2行は「件数」と「総和」への初期値(0)の代入、第3～7行のforブロックは「資料」の最初の要素(資料[0])の「変数名」への代入、及びそれ以降の存在する各要素の実数変換(parseFloat=(資料[i]-0))及び「総和」への加算と「件数」の1増(件数++)である。

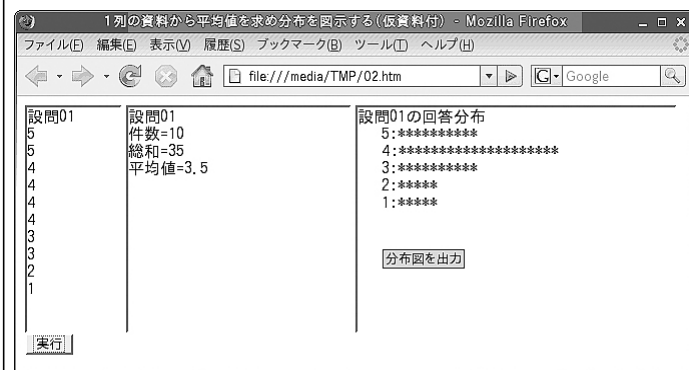
そして、最後の4行の記述により「変数名」、「件数」、及びMath.round(1000\*式)/1000(式の値を1000倍→四捨五入で整数化→1000で割る)によって小数第3位までに丸められた「総和」と「平均値」の値が、改行・説明文字列(“¥n件数=”などと共に第2要素(elements[1])に代入され、2つ目のtextareaに表示される。

```

【Script】
<title>1列の資料から平均値を求め分布を図示する(仮資料付)</title>
<form>
<textarea rows="12" cols="8" title="1列の資料を paste">
設問 01
5
【5改行を1行、4改行を4行、3改行を2行、2改行を1行省略】
1
</textarea>
<textarea rows="12" cols="24" title="処理結果を出力"></textarea>
<textarea rows="12" cols="36" title="分布図を出力"></textarea>
<input type="button" value="実行" onClick="処理()">
</form>
<script>
function 処理()
資料=document.forms[0].elements[0].value.replace(/¥r¥n/g,"¥n").split("¥n");
件数=0;総和=0;度数=new Array();for (i=-999; i<=9999; i++) 度数[i]=0;
for (var i in 資料)
if (0==) {変数名=資料[i]; continue;}
if ("¥n"==資料[i]) {continue;}
値=parseFloat(資料[i]);総和=総和+値;件数++;値=Math.round(値);度数[値]++;
}
平均値=Math.round(1000*総和/件数)/1000;総和=Math.round(1000*総和)/1000;
出力=変数名+"¥n 件数="+件数+"¥n 総和="+総和+"¥n 平均値="+平均値;
document.forms[0].elements[1].value=出力;
図="";
for (i=9999; -999<=i; i--)if (0==度数[i]) continue;
棒="";for (j=1; j<=Math.round(50*度数[j]/件数); j++) 棒=棒+"*";
空白="";for (k=1; k<=5-String(i).length; k++) 空白=空白+" ";
図=図+空白+"*"+棒+"¥n";
}
document.forms[0].elements[2].value=変数名+"の回答分布¥n"+図;
}
</script>

```

【実行結果】(仮資料のままの場合)



資料9 Pasteされた1列分の数値資料から平均値などを求め分布を図示するJavaScript及びFirefox 2.0 on Linux Mintでの実行結果

#### 4 階級別分布などの算出・提示方法の検討と実装

Pasteされた1列分の数値資料から階級別分布を算出し、各階級の割合を図示するためには、前節の処理に加え、下記の各処理が必要である。

- 各階級の度数計数用の配列を定義して初期値(0)を代入し、存在する数値を整数に丸めた値と一致する配列要素に1を加算する。
- 各階級の度数の%を計算し、その値に比例する図示用文字列を作成して「階級値・記号列・改行」の書式で降順(あるいは昇順)に出力用変数に追加する。

資料9は、a.・b.を実現するscript(02.htm)の内容とその実行結果である。

まず、資料8のscriptに「度数=new Array();for (i=-999; i<=9999; i++) 度数[i]=0;」が追加されて新配列「度数」の定義とその-999から9999の範囲の要素への初期値0の代入が行われ、また「値=Math.round(値);度数[値]++;」で各「値」を四捨五入により整数化した「値」と連番が一致する配列要素への1の加算が行われてa.を実現している。

次に、追加された2つ目のforブロックで9999から-999の範囲で降順に配列「度数」の要素(=各階級度数)の「件数」(=総度数)に対する割合の計算、それを50倍して四捨五入した数の"\*"(=2%につき\*1つ)の文字列「棒」への追加による図示用文字列の作成、及び5-階級値の桁数(String(i).length)分の「空白」+階級値(i)+":"と改行記号(¥n)で図示用文字列を挟んだ出力用文字列「図」の作成が行われ、「変数名」など+「図」の第3要素への代入によってb.が実現されている。

続いて、資料9の「図」への度数・%などの数値情報の追加表示をcheckboxによって選択可能としたscript(03.htm)を作成した。02.htmへの追加内容、及び03.htmの実行結果は資料10に示したとおりである。

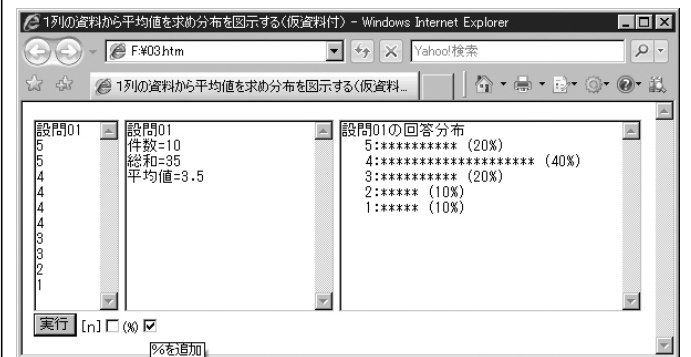
まず、form部分(ボタンの次)にチェックボックスを2つ追加し、次にfunction部分の2つ目のforブロックで小数第2位までに四捨五入で丸めた各階級の%の値を求めて「百分率」に代入している。そして、各チェックボッ

```

【Script】(追加部分のみ)
1. <input type="button" value="実行" onClick="処理0">の下に下記の2行
[n]<input type="checkbox" title="度数を追加">
(%)<input type="checkbox" title="%を追加">
2. 空白=""for...の行の上下に下記の計3行
百分率=Math.round(10000*度数[i]/件数)/100;
空白=""for (k=1; k<=5-String(i).length; k++) 空白=空白+" ";
if (document.forms[0].elements[4].checked==true) 棒=棒+"["+度数[i]+"]";
if (document.forms[0].elements[5].checked==true) 棒=棒+" (" +百分率+"%)";

```

【実行結果】(仮資料のままの場合)



資料10 分布図への度数・%の追加を可能にする追加script及びI.E.7.0 on Windows Vistaでの実行結果(%をcheckした場合)

クスに対応するformの連番4・5の要素がcheckedであれば、その階級の「度数[i]」・「百分率」を「棒」に追加することにより、度数・%などの追加表示が実現されている。

### 5 自動作図に必要なdata出力の検討と実装

3つ目のtextareaに提示された変数名、階級と棒、及び度数・%などは文字列であり、資料11に示したように、copy & pasteで文書処理ソフトに貼り付けて目盛・尺度などを追加すれば、簡素な「棒グラフ」としてレポートや発表用資料への利用が可能である。

しかし、本稿の「問題と目的」の1で述べた通り「内訳を対比的に提示して各項目の特徴に関する理解を深めるためには帯グラフの利用が最適」である。そして、資料の特徴や集計目的によっては、円・折れ線などのグラフによる提示が適切な場合も存在する。そこで、これらの基本的グラフの自動作成が可能なdata出力機能の追加を行った。

Wordの追加機能などでグラフを作成する場合、グラフの種類に応じた形式のdata作成が必要である。

資料12に示したように、Word上での帯グラフの作成には数値などの列

被害予想図について「知りたい」程度の回答分布

5:***** (38.69%)
4:***** (29.2%)
3:***** (24.82%)
2:*** (5.11%)
1:* (2.19%)

資料11 資料1のQ4-eへの回答の03.htmによる処理結果と「棒グラフ」としての利用例

※出力文字列を copy & pasteして文字列の目盛と尺度を追加し、MSゴシック9ポイント(行間10ポイント)で提示。

知りたい程度  とても知りたい  かなり  少し  あまり知りたくない  全く

被害予想図

53	40	34	7	3
----	----	----	---	---

資料12 資料11の処理結果のWord追加機能 (Microsoft graph) による帯・円・折れ線グラフ表示に必要な「データシート」の入力内容

件数	A	B	C	D	E
被害予想図	53	40	34	7	3

(縦) 方向での入力が、円・折れ線・棒グラフの作成には行(横)方向での入力が必要である。加えて、後者の場合、データシート上の各要素の左→右順は、円グラフでは時計回り順と、折れ線・棒グラフの横軸については左→右順と対応する(データシートの各要素の区切り記号は全てタブ=¥t)。従って、階級値と度数を行・列の各方向に階級値の昇・降順でタブ区切り出力する計4通りのdata出力機能を追加すれば、適切な部分の選択・copyとデータシートへのpasteにより、標準グラフの自動作成が可能となる。

資料13は、上述の計4通りのdata出力を実現するためのscriptを書き加えた04.htmのfunction部分(変更部分のみ)とその実行結果である。

出力用にd1~d6の変数を追加し、d1~d3については新たな降順forブロックで階級値(i)とその度数(度数[i])をタブ区切り(+“¥t”)で追記、d4~d6については既存の昇順forブロック内で同様に追記し、説明用の見出し([列・昇順: 帯用]など)と結合

```

【Script】(function 部分のみ)
function 処理()
  【9行省略】
  d1="回答¥t"+変数名+"¥n";d2="回答";d3=変数名;
  for (i=-999; i<=9999; i++){if (0=度数[i]) continue;
    d1=d1+i+"¥t"+度数[i]+"¥n";d2=d2+"¥t"+i;d3=d3+"¥t"+度数[i];
  }
  図="";d4="回答¥t"+変数名+"¥n";d5="回答";d6=変数名;
  for (i=9999; -999<=i; i--){if (0=度数[i]) continue;
    【6行省略】
    d4=d4+i+"¥t"+度数[i]+"¥n";d5=d5+"¥t"+i;d6=d6+"¥t"+度数[i];
  }
  出力 2="[列・昇順: 帯用]¥n"+d1+"¥n[列・降順: 帯用]¥n"+d4+"¥n[行・降順:";
  出力 3="円用¥n"+d5+"¥n"+d6+"¥n¥n[行・昇順: 線・棒用]¥n"+d2+"¥n"+d3;
  document.forms[0].elements[1].value=出力+"¥n¥n"+出力 2+出力 3;
  document.forms[0].elements[2].value=変数名+"の分布¥n"+図;
}
    
```

【実行結果】(仮資料のままの場合)

資料13 Word上でのグラフ自動作成用data出力機能を追加したscript及びI.E.6.0での実行結果 (nをcheckした場合)

資料14 2003以前のWordで作成されたdocに含まれるグラフはWord 2007上でも従来と同様の操作で修正・加工可能

の上「出力2・3」として従来の「出力」と共にelements[1].valueに代入している。

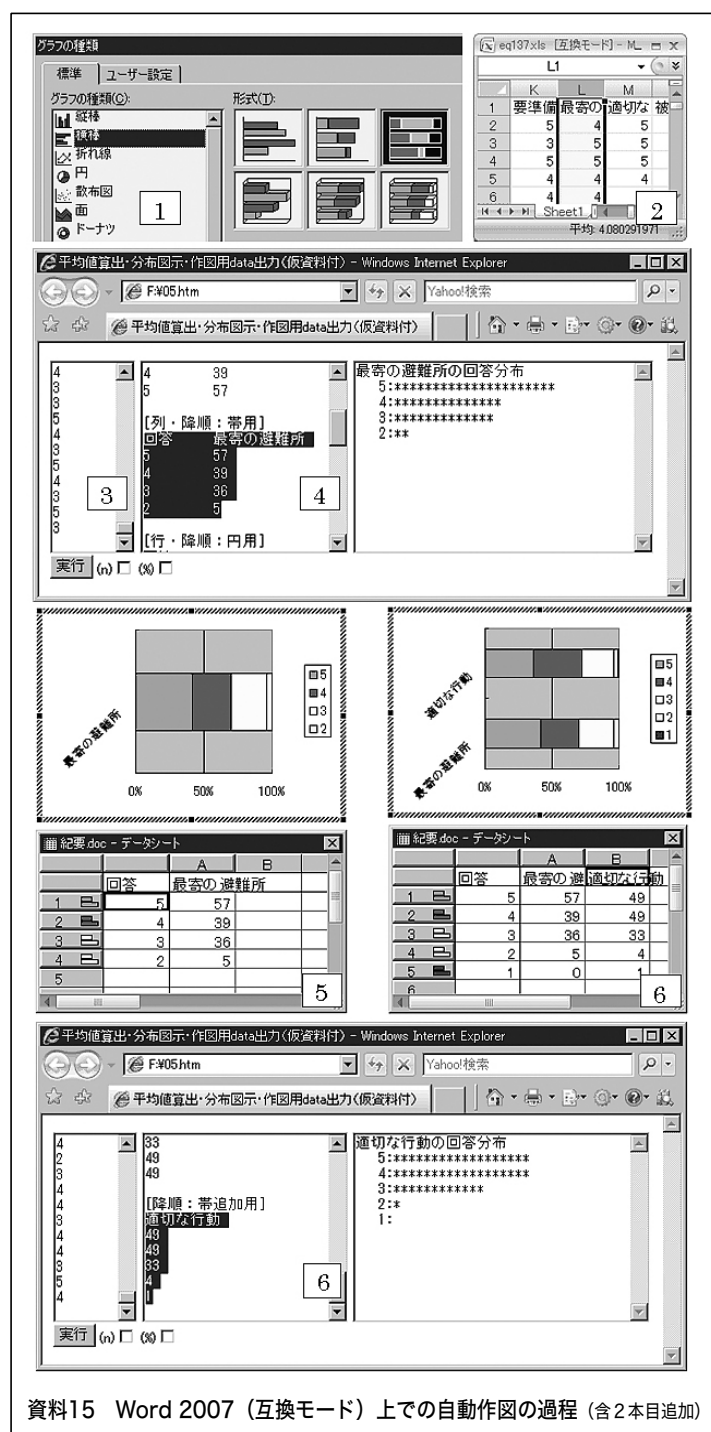
### 6 自動作図の方法

資料13にもその一部が示されているように、2つ目のtextareaには、計4通りの作図用dataが提示される。なお、各dataに対する[列・昇順：帯用]などの説明は、Wordの追加機能のMicrosoft graphにおける対応形式であり、作図ソフトによっては必ずしも該当しない。

資料14に示したとおり、2003以前のWordで作成したグラフを含むdocは、Word 2007では[互換モード]で開かれ、資料15に具体的に示したWord 2003までと同様の下記の操作が適用可能である。

1. [挿入]→[グラフ]→種類と形式を選択 (例：横棒→100%積み上げ横棒)。
2. Dataを含むfileをExcelで開く→当該data列文字 (例：L) click※→[Ctrl]+[C]などでcopy

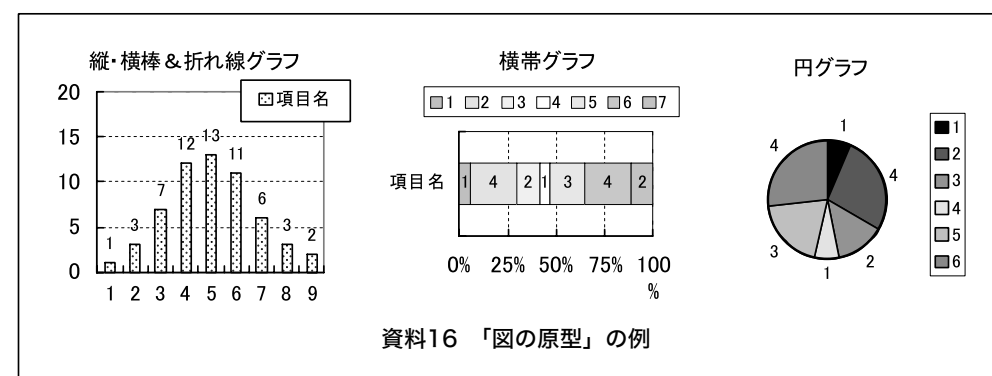
※当該data列文字のclickでは最終有効data以降の空行まで読み込んでしまう表計算ソフト



資料15 Word 2007 (互換モード) 上での自動作図の過程 (含2本目追加)

などの場合はdragによる正しい範囲指定が必要。

3. Script file (例：05.htm) を起動→[Ctrl]+[A]などで1つ目のtextareaを全範囲指定→[Ctrl]+[V]などでdataを上書きpaste→[実行]ボタンをクリック。
4. 2つ目のtextarea中の該当する作図用data (例：[列・降順：帯用]) をdragで反転→[Ctrl]+[C]などでcopy (2本目以降の帯については資料15と下記の6を参照)。
5. 1の「データシート」の左上隅のボタンをクリック(全範囲指定)→[Ctrl]+[V]などで作図用dataを上書きpaste→作図の自動実行。
6. 2本目以降の帯の追加には、階級値dataを省略した作図用data ([降順/昇順：帯追加用]) を利用 (Scriptは資料17：分布集計と自動作図.htmを参照)。



資料16 「図の原型」の例

資料16に一例を示したように、基本的なグラフについてその様式、凡例、文字サイズなどを整えた「図の原型」を作成・準備し、各「原型」のデータシートに作図用dataを上書きpasteすれば、修飾作業などはその大部分が省略可能となる。

### III 考察

資料10～15に示してきたとおり、今回JavaScriptを用いて作成したtool (分布集計と自動作図.htm) の使用方法 (interface) はブラウザと共通であり、「Excelなどの表計算ソフトから一列分の数値資料をcopy & pasteして[実行]ボタンなどをclickする」という単純な操作により、平均値と棒グラフ及び各階級の件数・%などの基本的な整理・要約結果の入手が可能である。また、平均値などに続いて出力される作図用dataをWordの追加機能であるMicrosoft graphの「データシート」などに貼り付けることにより、様々な種類のグラフの自動作成も可能である。従って、目的で述べた「ブラウザと同様の使用方法 (interface) で数値資料の整理・要約と図示が行える単機能tool」の原型としての「一列分の数値入力を処理してその階級別分布と%、平均値などを算出・提示し、併せて帯・円・折れ線グラフなどの自動作成に必要なdataを出力する支援toolの作成」は達成された。



さらに、今回の支援tool作成に使用したJavaScriptは下記の特長を備えており、今後の改良・発展、並びに広範囲かつ長期の活用の可能性が期待できる。

1. 改訂が容易:

変数名などに漢字など(2バイト文字)が使用可能なため可読性が高く、文法も簡明、かつテキスト・エディターのみで記述できるため、その理解・改訂などが容易である。例えば、textareaの大きさの変更、資料11に示したような「目盛と尺度」の自動出力、諸統計量(例:分散)の追加出力などは、それぞれrows・colsの数値の変更、「目盛と尺度」に対応する文字列変数の定義及び3つ目のtextareaの値への追加、必要な変数(例:平方和)と算出過程(例:分散=平方和/件数-(総和/件数)\*(総和/件数))の追加によって容易に可能である。

2. 発展性が大きい:

加藤(2005)のSPDAにおけるクロス集計、群別集計、散布図

```

<!-- saved from url=(0008)aboutinternet -->
<meta charset=x-sjis">
<title>分布集計と自動作図.htm: 平均値算出・分布図示・作図用 data 出力</title>
<form>
<textarea rows="16" cols="8" title="1列の資料を paste">
設問 01
5
【5改訂を1行、4改訂を4行、3改訂を2行、2改訂を1行省略】
1
</textarea>
<textarea rows="16" cols="24" title="処理結果を出力"></textarea>
<textarea rows="16" cols="36" title="分布図を出力"></textarea>
<input type="button" value="実行" onClick="処理()">
[n]<input type="checkbox" title="度数を追加">
(%)<input type="checkbox" title="%を追加">
<input type="button" value="説明" onClick="説明()">
</form>
<script>
function 処理()
資料=document.forms[0].elements[0].value.replace(/%r%rn/g,"%n").split("%n");
件数=0;総和=0;度数=new Array();for (i=-999; i<=9999; i++) 度数[i]=0;
for (var i in 資料)
if (0==i) {変数名=資料[0]; continue;}
if ("=="資料[i]) {continue;}
値=parseFloat(資料[i]);総和=総和+値;件数++;値=Math.round(値);度数[値]++;
}
平均値=Math.round(1000*総和/件数)/1000;総和=Math.round(1000*総和)/1000;
出力1=変数名+"%n 件数="+件数+"%n 総和="+総和+"%n 平均値="+平均値;
d1="回答%t"+変数名+"%n";d2="回答";d3=変数名;a1=変数名+"%n";
for (i=-999; i<=9999; i++)if (0==度数[i]) continue;
d1=d1+"%t"+度数[i]+"%n";d2=d2+"%t"+i;d3=d3+"%t"+度数[i];
a1=a1+度数[i]+"%n";
}
図="";d4="回答%t"+変数名+"%n";d5="回答";d6=変数名;a2=変数名+"%n";
for (i=9999; -999<=i; i--)if (0==度数[i]) continue;
棒="";for (j=1; j<=Math.round(50*度数[j]/件数); j++) 棒=棒+"*";
百分率=Math.round(10000*度数[j]/件数)/100;
空白="";幅="";余白=5-幅.length;for (k=1; k<余白; k++) 空白=空白+" ";
if (document.forms[0].elements[4].checked==true)棒=棒+"["+度数[j]+"]";
if (document.forms[0].elements[5].checked==true)棒=棒+"(%百分率%)";
図=図+空白+"%n";棒+"%n";
d4=d4+"%t"+度数[j]+"%n";d5=d5+"%t"+i;d6=d6+"%t"+度数[j];
a2=a2+度数[j]+"%n";
}
出力2="[列・昇順:帯用]%n"+d1+"%n[列・降順:帯用]%n"+d4+"%n[行・降順:"];
出力3="円用%rn"+d5+"%n"+d6+"%n%rn[行・昇順:線・棒用]%n"+d2+"%n"+d3+"%n";
出力4="%n[昇順:帯追加用]%n"+a1+"%n[降順:帯追加用]%n"+a2;
document.forms[0].elements[1].value=出力1+"%n%rn"+出力2+出力3+出力4;
document.forms[0].elements[2].value=変数名+"の回答分布%rn"+図;
}
function 説明()
alert("1. Excelなどの1列分の数値資料(第1行は変数名)を集計します。%n"+
"2. 件数[n]と%が追加可能でWindows/Mac OS X/Linux 汎用です。%n"+
"3. [列・昇順:帯用]などに続く data は word での自動作図用です。%n"+
"4. ご質問等は atsushikato.1958@gmail.com までお寄せください。%n");
}
</script>

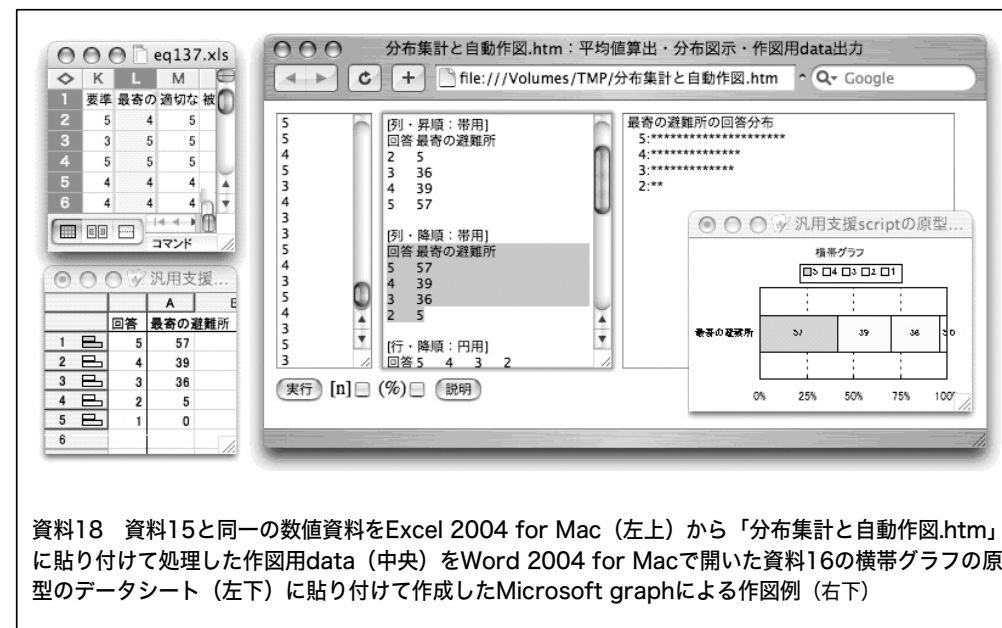
```

資料17 分布集計と自動作図.htmのscript

など、及び各種の基本的検定の実現が例示しているように、JavaScriptには数学関数を始めとする様々な機能が完備されている。従って、その活用による原型の発展可能性は大きく、多様で実用的な初心者向け「単機能tool」などの作成が期待できる。

3. 広範囲での長期利用が可能:

今回の原型tool作成に使用したJavaScriptは、情報通信システムの分野における国際標準化団体であるEcma Internationalの規格であり、資料6にも述べられているように、大多数のブラウザが対応している。その結果、資料8・9及び資料18に示したように、Windows以外の多様なOS上で使用可能であり、より広範・多様な場面でより長期間にわたり安定的に活用し続けることが可能である。



資料18 資料15と同一の数値資料をExcel 2004 for Mac (左上) から「分布集計と自動作図.htm」に貼り付けて処理した作図用data (中央)をWord 2004 for Macで開いた資料16の横帯グラフの原型のデータシート (左下)に貼り付けて作成したMicrosoft graphによる作図例 (右下)

上記の諸特長を活用し、例えば加藤(2004)のQPDの機能をJavaScriptで実現すれば、そのtoolはJavaScript対応ブラウザが存在する限り、Windows, Mac OS X, LinuxなどをOSとするパソコンに加え、PDA・携帯電話などを含む広範な機器上で、場面を選ばず長期間にわたり安定的に利用可能となる。初心者向け「単機能tool」の整備・充実という目的と併せ、その実現についても追求を進めたい。



## 文献

インセプト 2005 IT用語辞典 e-Words <http://e-words.jp/>

加藤厚 2005 二次分析機能を備えたonline data archiveの利点とその要件に関する検討  
宮崎公立大学人文学部紀要（第13巻） pp. 27-41. 宮崎公立大学  
<http://semi.miyazaki-mu.ac.jp/skato/spda/oda.doc>

加藤厚 2004 教育実践を支援する基礎的統計処理toolの必要性並びに現状とその1例としての  
QPDの過去・現在・未来  
宮崎公立大学人文学部紀要（第12巻） pp. 45-62. 宮崎公立大学  
<http://semi.miyazaki-mu.ac.jp/skato/qpd-a.doc>

田久浩志・小島隆矢 2006 マンガでわかるナースの統計学 オーム社

文部省 1999 高等学校学習指導要領 大蔵省印刷局  
[http://www.mext.go.jp/b\\_menu/shuppan/sonota/990301/03122603/011.htm](http://www.mext.go.jp/b_menu/shuppan/sonota/990301/03122603/011.htm)

山中 馨 2008 情報処理能力アンケート結果に基づいた初年次情報教育のカリキュラム設計  
創価経営論集 第32巻 pp.1-24. 創価大学経営学会  
[http://keiei.soka.ac.jp/assets/pdf/keieironsyu/vol32\\_no1-3/kaori\\_yamanaka.pdf](http://keiei.soka.ac.jp/assets/pdf/keieironsyu/vol32_no1-3/kaori_yamanaka.pdf)