The University of Osaka
Institutional Knowledge Archive

Title	JsStat : Java Scriptによる統計学
Author(s)	吉田, 光雄
Citation	大阪大学人間科学部紀要. 1998, 24, p. 271-302
Version Type	VoR
URL	https://doi.org/10.18910/6867
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

JsStat : JavaScript による統計学

吉田　光雄

目　次

はじめに
Ⅰ．JvaScript とは
Ⅱ．JavaScript の文法(抜粋)
Ⅲ．「統計学入門」：電子化統計学教科書の例
Ⅳ．統計学/心理統計法のためのユーザー関数一覧
Ⅴ．まとめ
文献

JsStat : JavaScript による統計学

吉田　光雄

はじめに

統計学があらゆる科学に浸透して、データ解析を裏面から支えている。急激な勢いで高性能化されつつあるパーソナル・コンピュータと相俟って、データ処理の両輪をなしている。しかし、皮相な見方をすれば、コンピュータが普及し、強力な統計処理アプリケーションが開発されればされるほど、ユーザーサイドにおける統計学の理論と技法に対する理解は薄らいでゆくように思われる。ユーザーフレンドリなコンピュータ OS と統計ソフトの出現は、アプリケーション・ユーザーの数を増大させてはいるものの、その中味がブラックボックス化され、インプットとアウトプットのみをつないで事足れりとする傾向が、次第に助長されていくように思われるからである。

多くの大学で、専門分野の如何を問わず、コンピュータ・リテラシーと統計学に関する科目が、高いウエイトを置いて学習されているが、初等統計学を学習中の学生にとって、早々と統計アプリケーションに慣れ親しむことは望ましいことではない。かっての学生が経験したように、例えば、電卓を用いて、ひとつひとつ数字を入力して、時には入力ミスに悩まされながら、統計処理の公式を理解していくことは、今日においても、なお統計学学習の基本であろう。学生のコンピュータ環境が豊かになり、すぐれた表計算や総合統計処理アプリケーションが氾濫しているが、原点に立ち返った統計学の学習が望まれるところである。

望ましいデータ処理の一つの方法は、電卓を用いることであるが、誤入力が多く、大量のデータ処理には不向きである。反復計算では再度のデータ入力が要求され、高度の計算には耐え得ない。そこで、思い起こされるのが、コンピュータ言語によるプログラミングである。統計処理の中味（計算手順）の理解なくしては、プログラミングはできないが、コンピュータ利用法は、かってのメインフレーム時代のプログラム全盛の時代から、パーソナル・コンピュータが普及し、アプリケーション万能の時代を迎えて、プログラミングの困難さを乗り越える学生は少数派となった。特殊な場合を除いて、情報処理科目でコンピュータ言語（かっての Fortran, Pascal, C 言語など）が教えられることはない。現在の花盛りはワープロ、電子メール、インターネットである。ホームページの構築用に HTML や、高度の文書整形に LaTeX が学習されるが、これらは中級以上のクラスにおいてであろう。

本稿では、そうしたネットワーク時代におけるコンピュータ・ツールとして、JavaScript を取り上げる。

I. JavaScript とは

インターネットが急速に普及し、多くのところで Web ページが作成され、情報の交換が盛んに行われている。その際使用される言語が HTML (Hyper Text Markup Language) であるが、HTML の文法を超えて、ユーザーカスタマイズに便利なように開発されたスクリプト言語が JavaScript である。代表的なインターネット・ブラウザ、例えば、N. Navigator、M. Internet Explorer などの最新バージョンには、標準搭載されており、インストールなどの作業を必要とせず、手軽に使用することができる。

JavaScript は、汎用の簡易言語として、プログラミング言語の知識がなくても十分使用可能である。かっての BASIC 言語より、文法は簡単であり、プログラミング未経験者にとってもなじみやすいものである。インターネット上でサーバーとクライアント側を結んで情報を処理するための言語として、1995年に開発された。先行の JAVA の簡易版とも言えるが、両者は全く異なるものであり、本稿で取り上げるのは JavaScript の方である。

JAVA は型チェックやセキュリティを厳密に行う高機能言語であるが、JavaScript は単純な機能に限って、手軽に使用できるように設計されたものである。しかし、構造は簡単であっても、最新のオブジェクト指向方式を取り込んだ、新しいタイプの言語である。開発・拡張が目覚しく、最新の仕様の確定は困難であるが、情報はインターネットで公開されている(例えば18, 19)。 もともと、Web 画面の修飾を目的に開発されたため、そうした機能が豊富に用意されているが、数値計算、数学的処理を行うための、コマンドやオブジェクト、関数なども多数用意されており、それらを利用することにより、統計学の学習や、データ処理などにも活用することができる。JavaScript で大量のデータ処理を行うためには、ハード・ソフト両面での制約が多いが、逆にコンピュータ・ネットワークの時代には、機種を超えて遠隔地からの処理を可能とすることなど、統計処理アプリケーションとは異なる次元での長所も多い。

プログラムは HTML ドキュメント中に、タグ<script>~</script>を用いて書けばよく、インタプリタとして、翻訳、実行される。エラーは直ちに診断され、その場所が知らされるので、修正は容易である。コンピュータと対話しつつ、デバッグを済ませ、計算を行うことができる。 電卓の場合は数値キーを叩いて入力しなければならないが、誤入力も多く、複雑な計算には不向である。JavaScript の場合は、統計処理の内容をプログラミングし、事前に電子化されたデータを用意して処理するため、こうした難点は解決される。エディタと連動して利用すれば、計算の反復、結果の保存・編集なども容易で

ある。

JavaScript を用いて、統計学を学習し、また実際の研究用データ処理にも活用されるシステムとして、 JsStat (Statistics by JavaScript)を構築したので(29)、それについて報告する。

JsStat の構成

プラットホーム

インターネット・ブラウザの搭載されたコンピュータ(Macintosh, Windows 機)

URL　　http : //koko15. hus. osaka-u. ac. jp/~yoshida/jsstat/

構成内容

[1]. JavaScript の文法解説

[2].「統計学入門」：電子化統計学教科書の例

[3]. 初等統計学ユーザ関数

[4]. 多変量解析ユーザ関数

[5]. 心理学的測定法ユーザ関数

[6]. 統計処理用サンプルデータ集

[7]. 他サイトへのリンク (StatLib, Online Statistics, Departments of Statistics)

[8]. その他

II. JavaScript の文法(抜粋)

JsStat に必要な JavaScript の文法について述べる。最小限の内容であり、以下のユーザー関数を理解するには十分であるが、更に詳細については成書 (1～8) を参照。

(1). プログラムの構成

一連の処理を JavaScript の文法に従って書いたものを JavaScript のプログラムという。HTML ファイルの中の一部として書き、従って、プログラムの拡張子として ～.html を使用する。

プログラム部分の冒頭に <script language=JavaScript> ないしは <script>、末尾に </script> のタグが必要である。すなわち、<script>~</script> で囲まれた部分が JavaScript で書かれたプログラムとして、解読され、処理される。<head>部、<body>部のいずれに挿入してもよい。また、HTML タグ中にイベントハンドラとして書き込むこともできる。

HTML のタグ(のいくつか)は有効で、JavaScript に優先して、HTML タグとして解読される。処理される数値(データ)もプログラム中で与えられ、処理結果は画面に出力される。データを外部ファイルで与えたり、処理結果を外部ファイルに書き込んだりす

ることはできず、一見不便のようであるが、出力をファイルに落としたいときは、処理後、プルダウンメニュから保存を選ぶか、コピー／ペーストで他のエディターに取ればよく、実質的に不便はない。ただ、データの入力に関しては、プログラム中で行わなければならず、大量のデータの場合には、やや手間であるが、他の言語と同様、配列が使用できるので、それを用いて行えばよい。

JavaScript で用意された、処理の単位を文という。プログラムは文の集まりである。

空行、空白は処理には無関係である。従って、多くの場合、空行を挿入して関連する一連の処理をまとめ、さらに行内にインデントを与えて処理の深度を示し、プログラムを分かり易くする工夫がなされる。かっての BASIC のようなスパゲッティ・プログラムは、今日、特別の場合を除いて書かれることはない。JavaScript はオブジェクト指向を取り入れた最新の言語である。

(2). 予約語

JavaScript の処理系側で予め使用されており、ユーザー側で使用できない用語である。一般によく使用されると思われるのは次のものであり、これらを変数・定数の名称として使用することはできない。

boolean, break, byte, case, char, class, const, continue, default, do, double, else, extends, false, final, float, for, function, goto, if, implement, import, in, int, interface, long, new, null, private, public, return, static, true, var, void, while, with など。

(3). 変数・定数(リテラル)

他の言語処理系のように、実数、整数の区別をする必要はなく、入力された値により、自動的に判定される。従って事前にその宣言をする必要はない。変数の名前は、英字で始まる英数字で、特に長さの制限はない。ただし、大文字、小文字は区別されるので、注意が必要である。

実数、整数の数値、および文字列を定数として取り扱うことができる。文字列は“～”のようにダブルクオテーションで囲む。

(4). 代入

“=”で右辺の値が左辺の変数に代入される。左右の変数が同じ値かどうかの判定は“==”が用いられる。

例： x=a； if (x==a) ～

(5). 配列

類似の変数の集まりは配列としてまとめた方が処理しやすい。配列を最初に使用するとき、配列の大きさの宣言をしておかねばならない。実際のデータが宣言されたサイズより大きいとき、配列は自動的に拡大され、データが失われることはない。

初期設定の例：

a=new Array(3)　　大きさ 3 の配列が用意される。値は null。

b=new Array(0,0,0)　　大きさ 3 の配列が用意され、値として 0 が代入される。

配列要素の参照

配列要素の番号は 0 から始まるので、大きさ n の配列 A の第 i 要素は A[i-1] である。

A[0], A[1], A[3], …, A[n-1]

配列要素は１次元のベクトルとして格納されており、添字をコントロールすることにより、仮想的に、２次元あるいはそれ以上の次元の配列としても取り扱うことができる。他言語のように、直接高次の配列を書けないが、変換の関数を用意して、仮想的に使用することができる。

配列の要素を戻り値として返す関数を作成してもよいし、配列の添え字を返す関数を作成してもよい。次はその例である。

例：２次元配列 A(a x b) の第(i,j)要素は A[i*b+j] で参照される。

```
function a2(aa,na,nb,ii,jj) {
// 大きさ(na,nb)の仮想配列 aa の
// 第(ii,jj)要素を取り出す。
return aa[ii*nb + jj] }
function b2(nb,ii,jj) {
// 大きさ (na,nb) の仮想配列の
// 第(ii,jj)要素の添え字を返す。
return ii*nb + jj }
```

例：３次元配列 B(a x b x c) の第(i,j,k)要素は A[i*nb*nc + j*nc + k] で参照される。

```
function a3(aa,na,nb,nc,ii,jj,kk) {
// 大きさ (na,nb,nc)の仮想配列 aa の
// 第(ii,jj,kk)要素を取り出す。
return aa[ii*nb*nc + jj*nc + kk]}function
b3(nb,nc,ii,jj,kk) }
// 大きさ (na,nb,nc) の仮想配列の
// 第(ii,jj,kk)要素の添え字を返す。
return ii*nb*nc + jj*nc + kk }
// main
aa = new Array(1,2,3,    4,5,6)
na = 2 ; nb = 3
document.write("<br>a=",a2(aa,na,nb,1,
1))
document.write("<br>a= ", aa[b2(nb,1,
1)])
bb=new Array(1,2,3,4,   11,12,13,14,
21,22,23,24, 31,32,33,34, 41,42,43,44,
51,52,53,54)
na=2 ; nb=3 ; nc=4
document.write("<br>b=",a3(bb,na,nb,
nc,1,1,2))
document.write("<br>b=",bb[b3(n b,nc,1,
1,2)])
// end
```

(6). 演算

演算は、算術演算(四則演算等)、論理演算(真偽の判断)、比較演算(大小の判断等)、文字列操作演算などが定義されている。

算術演算	演算子	例、備考
	加算	+
	減算	-

乗算	*	
除算	/	商は少数点つきの実数
剰余	%	整数を整数で割ったときの余り
増分	++	1 を加える
減分	--	1 を減ずる

論理演算	演算子	例、備考
AND	&&	
OR	||	
NOT	!	

等しい	==	
不等	!=	
大きい	>	
大 or 等	>=	
小さい	<	
小 or 等	<=	

文字列操作	演算子
連 結	+ “abc” + “def”→ “abcdef”

比較演算　演算子　真の時，値　true を返す

(7).制御　プログラムの中で処理は書かれた順番に従って行われるが、処理の流れを変更、制御する文として、for, if, while 文がある。

(処理)の内容が複数個の文のとき、間に；を置く。一連の処理は｛｝で括られるが、処理が単独の文のとき｛｝を省略してもよいし、見やすくするため、書いても差し支えない。

(7−1). for 文（処理の反復）

for(初期値；終了判定；増分)　{処理}

初期値から、終了判定の条件が満たされる限り、制御パラメタを増分させながら、{処理}　を反復する。

例：1 から n までの整数の和

```
sum = 0;for (i=1; i<= n; i++) sum=sum + i
```

(7−2). if 文（処理の分岐）

if（条件）　{処理}

（条件）の判定を行い、それが“ 真 ”のとき、{処理} を行った後次の文を実行し、“偽” のとき、{処理} を飛ばして次の文を実行する。

例：偶数かどうかの判定

```
if (n % 2 == 0) document.write("<br>",n, "= 偶数")
```

if（条件）　{処理1}　else　{処理2}

前記の if 判定を少し複雑にし、（条件）が “ 真 ” のとき {処理1} を行い、（条件）が“ 偽 ” のとき else 以下の {処理2} を行って、次の文に進む。

例：偶数か奇数かの判定

```
if (n % 2 == 0) document.write("<br>",n, " = 偶数")
else document.write("<br>", n, " = 奇数")
```

if-else 文を簡単にした代入文として、次のものがある。

```
変数 = (条件) ? val1 : val2
```

(条件)が真のとき、変数の値は val1 となり、偽のとき、val2 となる。

(7−3). while 文（処理の反復）

while（条件） {処理}

（条件）が満たされる限り {処理} を反復し、（条件）が " 偽 " となった時点で反復を終了する。

例：1 から n までの和

```
sum = 0 ; i = 0
while (i <= n) {sum = sum + i ; i++}
document.write("<br>sum= ", sum)
```

(7−4). break 文（反復処理の終了）

処理の中断終了を行う。for, while 文中で用いられ、反復処理の流れを変更して、終了する。

(7−5). continue 文（処理のスキップ）

for, while 文中で用いられ、それ以後の残りの処理をスキップして、ループブロックの先頭に処理を移す。

(8). プリント

画面への出力は

document.write("
文字列",処理結果,"<p>")を用いる。文字列の出力は "〜" （ダブルクオテーション）で囲み、数字の演算結果はそのまま書く。引用中の
は改行、<p>は２行改行を示す。

(9). 関数

一連の処理をメインの処理から切り離し、関数として定義しておけば、同一の処理を何度も反復して書く必要がなくなる。

関数との値のやりとりは引数を用いる。また、return 文で処理結果を戻すこともできる。関数を呼ぶ方の引数と、呼ばれる関数の側の引数は、型(変数、配列、文字列等)と個数があっていなければならないし、書かれた順に従って、対応がなされる。従って、名前は違っていてもよい。

変数の値が保持される範囲をスコープという。関数内でのみ有効なものをローカル(局所)変数、全プログラムを通じて有効なものをグローバル(全域)変数という。目的に応じて両者を使い分ける必要があるが、同一の名前はグローバルに使用され、実引数と仮引数で異なる名前を使用しておけば、仮引数の値は関数内でのみ有効である。また、関

数内の変数、配列に対して var 宣言をしておけば、それは関数内のみで、使用される。

よく使用される次の関数(Math オブジェクト）は、予め処理系に組み込まれているので常時使用が可能である。

(9—1). 数学関数(メソッド)

絶対値	Math.abs(x)	| x |
平方根	Math.sqrt(x)	$\sqrt{x}$
ベキ乗	Math.pow(x,k)	x^k
対数	Math log(x)	
指数	Math.exp(x)	e^x, exp(x)
sin	Math.sin(x)	sin(x)
cos	Math.cos(x)	cos(x)
tan	Math.tan(x)	tan(x)
arc sin	Math.asin(x)	$\sin^{-1}(x)$
arc cos	Math.acos(x)	$\cos^{-1}(x)$
arc tan	Math.atan(x)	$\tan^{-1}(x)$
最大値(2数の)	Math.max(x1,x2)	max
最小値(2数の)	Math.min(x1,x2)	min
最大整数値	Math.floor(x1,x2)	
乱数	Math.random()	

数学定数(プロパティ)

e	Math.E	2.718281828459045
π	Math.PI	3.141592653589793
$\sqrt{2}$	Math.SQRT2	1.414213562373095
$\sqrt{1/2}$	Math.SQRT1_2	0.707106781186547

(9—2). 文字列関数

実数化	parseFloat(string)	文字を浮動少数点に変換
整数化	parseInt(string)	文字を整数に変換

(9—3).ユーザー関数

処理系で定義されていない処理はユーザー自らが作成せざるを得ない。例えば、除算演算子(/)は整数を整数で割り、割り切れないときは実数の答えを返すが、整数の商を整数で得たい時もある。他の言語処理系のように、この演算が定義されていないので、ユーザー関数として作成する他はない。

さらに、実数の計算結果は普通、桁数は10～20と多いので、任意の有効数字の出力のために、ユーザー関数を作成しておくと便利である。必要な小数点以下の桁数を指定することができる。

```
function div(a,b) {
// 整数の除算(a/b の商を、小数点以下を
// 切り捨て、整数で返す)
return Math.round(a/b-0.5) }
function dec(aa,dc) {
// 実数 aa のうち、小数点以下を dc 桁として
// その値を返す
maxdec=Math.pow(10,dc)
bb=Math.round(aa*maxdec) / maxdec
document.write(" ",bb, " ")
return }
```

普通、関数は小さい処理を一つのまとまりとして書かれることが多く、大きい処理は、多数の関数を組み合わせて書かれることとなる。次節において、統計処理に関するユーザー関数を集め、解説を加える。

(10). コメント

// 以下から行末までの内容は、コメントとして、処理からは外される /* 複数の行にまたがるときは、スラッシュとアスタリスクを組み合わせて 使用してもよい。 */

プログラム、データの内容、その他のコメントを多く書いておいた方が内容の理解、整理に役立つ。空白行は処理されない。文は一行内に書いてもよいし、複数行にまたがってもよい。空白の字も処理されない。全角の空白は文字列内で用いられる以外は、エラーとなる。

(11). デバッグ

プログラムを書いた後、デバッグ(虫とり)の作業が必要である。よく見られるエラーとしては次のようなものがある。

1. 括弧 { }, ( ) の誤対応
2. 引用符 ",' の誤対応
3. 大文字、小文字の区別
4. 変数、配列、関数などの未定義 (定義する前に参照)
5. 変数、配列、関数名の重複
6. 区切り , ; の誤使用
7. 全角空白の使用
8. 変数、配列、関数名の重複
9. ゼロの除算、マイナスの平方根
10. 実行時エラーの見過ごし

例えば、インデント、コメント等を多用して、変数や処理の中味を分かりやすくすることが大切である。かっては、職人芸を競い、メモリの少ない、計算時間の速い、凝ったプログラムが好んで書かれたが、計算機環境の向上した今日、要求されるのは分かり易い、デバッグの容易なプログラムであろう。

文法的なエラーが消えても、実行時にエラーが発生する場合もある。計算がストップすれば、エラーの存在はすぐ分かるが、アルゴリズムの間違いのため、計算は行われても、正しい結果が得られていない場合は厄介である。既存のアプリケーションで処理結果を確認する、などの注意が必要であろう。

(12). プログラムの実行

プログラムを実行するには、予めデータを、HTML ファイルの中に書き込んでおき、それを読めばよい。処理(計算)結果が画面に出力される。 やや洗練された方法として、以下の例に示すように、「計算」ボタンを押すことにより関数が実行され、結果が次ページに書き込まれる、とすることもできる。ウインドウを更に一つ開いて、そこに計算結果を出力することも可能である。本来、JavaScript はそうした個性的な画面(ホームページ)の作成用として開発されたものであり、ユーザー・カスタマイズの道具として、

活用されている。ここでは、統計処理が目的であり、個性的な画面の構成を必ずしも意図していない。必要に応じて、作成されればよい。

例：myclick1.html ----------

```
<html><head>
<script>
function my(){
document.write("<br>10！は")nfact=1
for (i=1; i<=10 ; i++)nfact=nfact*i
document.write(nfact,"です。")
document.write("<p>Back ボタンで前ページに戻ります。<p>") }
</script></head>
<body>
10！ を計算します。<p>
ボタンを押して下さい。
<form>
<INPUT TYPE="button" VALUE="計算"
onClick="my()">
</form>
</body></html>
```

次の例は２つの作業をボタンで選択するものである。

例： myclick2.html ----------

```
<html><head>
<script>
function my1(){
nsum=0
for(i=1; i<=10 ; i++)nsum=nsum+i
document.write("<br>Sum10=", nsum,
"<p>")
document.write("<p>Back ボタンで前ページに戻ります。<p>") }
function my2(){
nfact=1
for (i=1; i<=10 ; i++) nfact=nfact*i
document.write("<br>10！=", nfact, "<p
>")
document.write("<p>Back ボタンで前ページに戻ります。<p>") }
</script>
</head>

<body bgcolor="#ffffff" >
<p>Sum(10)「1から10までの整数の和」
あるいは 10！ を計算します。<p>
どちらかのボタンを押して下さい。
<form>
<INPUT TYPE="button" VALUE="Sum
(10)" onClick="my1()">
<INPUT TYPE="button"VALUE="10！"
onClick="my2()">
</form>
</body></html>
```

III. 「統計学入門」：電子化統計学教科書の例

インターネットの普及とともに、それを統計学教育にも取り入れようとする試みが現われ始めた。 例えば、Rice University(Houston, USA)における例(30)であるが、HyperStat Online(Lane, D.M.)の名のもとに、

Introduction
Describing Univariate Data
Describing Bivariate Data
Introduction to Probability
Normal Distribution
Sampling Distributions
Point Estimation
Confidence Intervals
The Logic of Hypothesis Testing
Testing Hypotheses with Standard Errors
Power
Introduction to Between-Subjects ANOVA
Factorial Between-Subjects ANOVA
Within-Subjects ANOVA
Prediction
Chi Square
Distribution-Free Tests
Measuring Effect Size

など、統計学全般についての解説が電子化情報として提供されている。その他にも

Networked Classroom (Lane, D.M. and Atlas, R.S., Rice University)
Statistics : The study of stability in variation (Leeuw, J. de, UCLA)
Introductory Statistics : Concepts, models, and applications (Stockburger, D.W., Missouri State University)
SurfStat (Dear, K., The University of New Castle)
Statistics 30X class notes (Wang, C.N. and Whiting, Texsas A&M)
Introduction to Quantitative Methods (Glass, G., Arizona State University)
Statistics Lectures (Mosier, C., Clarkson College)
MAT220X Outline (Biafore, C., New Hampshire College)

など(31～39)に、講義要項が詳細にわたって公開されている(30X, 220X 等は講義番号)。これらはいずれも、統計学の書物としても十分自習できる程度に分かりやすく、しかも体系的に書かれており、新しい教育の方法を示唆するものであろう。

これらの主旨にそう一例として、統計学教科書(5)を取り上げ、電子化を試みた。主として大学の文科系クラスで使用されており、「数学を抜きにして真の統計学の理解はありえない」、との方針のもとに、最小限の数学に絞って、統計学の理論と技法が解説されたものである。 多くの例題、演習問題は、電卓を用いて計算されるという前提で、問題が選ばれているが、近年、大学のコンピュータ環境は急速に改善され、国公私立大学のいずれを問わず、多くの大学で豊かな環境が提供されるに至っている。この環境の

激変に対応して、本稿はより進んだ計算の道具としてコンピュータの使用を想定する。その際、まず思いつくのが、コンピュータにおける統計ソフトの使用であろう。統計処理専用ソフトに加え、表計算アプリケーションにも多くの統計関数が用意され、便利にデータ処理を行うことができる。

本学の１年生に限って言えば、コンピュータ・リテラシー科目、「情報活用基礎」は第Ｉセメスタで必修科目として課せられており、情報処理教育センターで学生の使用可能なアプリケーションは、SAS(総合統計解析システム)，Mathematica(数式処理ソフト)，LQ，Wingz(表計算) などである。また、多くのパソコンに搭載されている Excel(表計算)も個人レベルでの普及度も高い。

しかし、問題はコンピュータにより、インストールされているアプリケーションが様々である点と、さらに統計処理の手順がユーザーの目から隠され、処理の中味を知ることなく、インプットとアウトプットに注目するだけで、データ処理が可能なことである。統計学の学習を始めたばかりの学生にとって、こうした状況は望ましくなく、単なる計算の便法としてしか統計学が学習されない恐れがある。統計学も他の科学(学問)と同様、理論の体系であって、単なる計算技法の集まりではない。

文科系学生にとって道は険しくとも、初期学習者にこのことを伝えるのは大切なことである。そうした観点から、手計算で数字キーを押す負担から解放するが、計算の中味をなおざりにはしない方法としてコンピュータ・プログラミングを取り上げる。

コンピュータの使用は、初期のメインフレームの頃における数値計算から、次第に多様なアプリケーションが開発され、ワープロに代表されるような、机上の知的作業を支援する、電子文房具としての役割を担うに至っている。かってのコンピュータ言語全盛は昔日の感が強いが、しかし、いかなる時代にあってもコンピュータ使用の原点は言語である。

そして、上述のごとく、今一つの爆発的使用が、インターネットに代表されるコンピュータ・ネットワーキングである。良質のインターネット・ブラウザが提供されたことと相俟って、ホームページの作成は、集団・組織を越えて、個人のレベルでの普及を見るに至っている。かってのパソコンに搭載された Rom–Basic と同じように、パソコン購入の時点でブラウザがプレインストールされており、手軽にインターネットに入ることができる。加えて、Rom–Basic とは比較すべくもないほどの良質のブラウザが提供されている。

本節はそうした、インターネット・ブラウザに標準搭載されている JavaScript を用いて統計学の学習ならびにデータの統計処理を行うものである。JavaScript を使用するためのプラットホームは容易に提供されていると見て、上記、初級統計学の教科書を電子化し、学習者の自発的な学習を支援する教材を提供し、計算に関わる部分をプログラミングして埋め込んだ。記載のフォーマットを例示するのみであり、実際のホームペー

ジは、HTML 言語で書かれており、ブラウザにより閲覧可能である。いくつかの個所について、要点を抜粋、例示する。

第２章　データ処理（抜粋例示）

記述統計学について解説された章であるが、度数分布、平均、分散（標準偏差）、相関係数について、JavaScript による計算法を示す。

§1. 度数分布表

大量のデータについて、全体の傾向を知る方法は、視覚的に分布の状況を眺めることである。まず、データを分類して表の形に整理し、そのグラフを描く。統計学でよく用いられる方法は、粗(raw)データをいくつかの階級に分類し、度数分布表(frequency table)に整理することである。数値例として、女子学生の身長のデータを用いる。

【数値例 データ 1】女子学生の身長(n=60)

149	153	153	155	153	165	155	160	165	155
157	161	161	155	155	155	155	151	159	157
157	154	165	157	154	157	151	164	155	155
158	163	160	154	160	163	156	163	155	147
162	152	158	156	166	158	160	155	161	155
159	158	159	158	148	153	167	172	168	153

(1)計算の実行

度数分布　実行

度数分布表算出へのリンクがはられているので、計算を行う時は、実行をクリックすればよい。飛んだ先で２つのボタンが現われるので、いずれかを選択し、クリックすれば、計算が行われる。

度数分布(階級幅に分類)

度数分布(整数で集計)

(2)ソースの表示

プルダウンメニュの

Netscape　　View/Document Source

Internet Explorer 表示/ソース表示

を選択して実行する。画面が代わって、ソースプログラムが現われる。

(3)データを更新して計算を実行

サーバー側で計算を実行する場合には、ソースを見るだけでよいが、クライアント側

でデータを変更して計算を行うためには、新しいデータの書き込まれたファイルをユーザーが作成しなければならない。エディターあるいはワープロで編集・作成した後、クライアント・コンピュータの適当なディレクトリに保存する。そのファイルをブラウザに取り込めば、新しいデータで計算が行われる。

ソースプログラムは以下の通りである。<head>に関数群を置き、<body>部にデータおよび関数の呼び出しのための文を書くが、すべてを<body>部に置いてもよい。HTML ファイルのいずれの部分でも、タグ<script>~</script>ではさまれた部分が、JavaScript のプログラムとして解読される。

例題の場合、事前にデータの視察により、度数分布表に集計するためには、階級値が必要であるので、第１階級の下限値と階級幅(a1=147.5 ; ci=3 ;)を与えておく。

```
ファイル名：chap2__1.html--------
<html><head><script>
function digittab(x) {
// Digit Tabulation
document.write(" <p>===Digit
 Tabulation ===<p>")
x=new Array(3,2,4,8,9,12,0,1,8,4,2,6,
5)
// x=height
f=new Array(0,0,)
n=x.length ; xmin=x[0] ; xmax=x[0]
for (i=1; i<n ;i++) {
 if (xmin > x[i]) xmin=x[i]
 if (xmax < x[i]) xmax=x[i] }
for (i=xmin-1; i<=xmax-1 ; i++) f[i]=0
 for (i=0 ; i<n ; i++) { k=x[i] ; f[k]=f[k]+1 }
document.write("<p>Input Data<p>")
for (i=0 ; i<n ; i++) { document.write(x
[i], " ")
 if (((i+1) % 10) ==0)
 document.write("<br>") }
document.write("<p>Frequency<p>")
ns=0 ; for (i=xmin ; i<=xmax ; i++) {
document.write("<br>", i, " ", f[i])
ns=ns+f[i] }
 document.write("<p> 計", n)
document.write("<p>Digit Tabulation
End")
document.write("<p>Back ボタンで前ペ
ージに戻ります。<p>")
return }

function tabulation(x,a1,ci) {
var ns,i,xmax,xmin,k
document.write("<br>==== 度数分布 =
==<br>")
// Input Data write
// x=data, a1=Lower Bound, c=Class
interval
document.write("<br>Input Data : <br
>")
ns=x.length
for (i=0 ; i<ns ; i++) {
document.write(x[i]+",")
 if (((i+1) % 10) == 0) document.
write("<br>") }
```

```
document.write("<br>Data size=", ns,
"<p>")
// Frequecy Distribution
document.write("<br>Frequency
 Distribution<br>")
 a=new Array(3) ; b=new Array(3) ;
 xc=new Array(3) ; f=new Array(3) ;
 xmin=x [0] ; xmax=x [0]
 for (i=1 ; i<ns ; i++) {
 xmin=Math.min(xmin, x[i])
 xmax=Math.max(xmax, x[i])}
 document.write("<br>Min=", xmin,
 "<br>Max=", xmax, "<br>")
  k=Math.ceil((xmax − a1) / ci) +1
a[1] =a1 ; a[0] =a[1] − ci
 for (j=2 ; j<=k ; j++){ a[j]=a[j−1] + ci }
 document.write("<br> A1=", a1, "<br
>Class
Interval=", ci, "<br>K=", k, "<br>")
 for (j=0 ; j<k ; j++){
 f[j]=0 ; b[j]=a[j] +ci ; xc[j]=a[j]+ci/2 }
 for (i=0 ; i<x.length ; i++){
 kk=Math.floor((x[i] − (a1 − ci)) / ci)
f[kk]=f[kk]+1 ; }
document.write("<br>a=lower bound, b
=upper bound, x=class mark<br>")
document.write("<br>Frequency Table<
br>")
document.write("No. a-- x(CM) --b
Freq<br>")
for (j=0 ; j<k ; j++)
document.write(j, "  ", a[j], "  ", xc[j],
"  ", b[j], "  ", f[j], " <br>")
document.write("<br>Frequency Table
End")
document.write("<p>Back ボタンで前ページに戻ります。<p>")
return }
</script></head>

<body>
<h3>女子学生の身長データを、度数分布表に集計します。</h3>
<form>
<INPUT TYPE="button" VALUE="
job2_1 度数分布(階級幅)" onClick="job
2_1()">
<INPUT TYPE="button" VALUE="
job2_2 度数分布(整数)" onClick="job2_
2()"></form>

<script>
function job2__1() {
// 女子学生の身長データ
height=new Array(
149,153,153,155,153,165,155,160,165,155,
157,161,161,155,155,155,155,151,159,157,
157,154,165,157,154,157,151,164,155,155,
158,163,160,154,160,163,156,163,155,147,
162,152,158,156,166,158,160,155,161,155,
159,158,159,158,148,153,167,172,168,153)
a1=147.5 ; ci=3 ;
tabulation(height, a1, ci)
return }

function job2__2() {
height=new Array(
149,153,153,155,153,165,155,160,165,155,
 前項と同じにつき、中略
159,158,159,158,148,153,167,172,168,153)
digittab(height)
return }
</script></body></html>
```

§2. 分布の特性値

データの統計処理として、その特徴を把握するために、例えば、平均、分散（標準偏差）が算出される。

粗データ(raw data)をそのまま処理する

度数分布表に集計したもの(tabulated data)を用いる

場合とに分けて述べる。

2—1. 平均

分布の全体的傾向を示す指標を代表値という。全データを１つのスコアで代表させるものであり、ふつう平均が用いられる。

女子学生の身長のデータについて、平均を求める。

$x_m = 1/60(149+153+\cdots+153) = 157.58$　　粗データの場合

$x_m = 1/60(146 \cdot 1+149 \cdot 2+\cdots+170 \cdot 1) = 157.60$　　度数分布表データの場合

2—2. 分散

データのもつ情報として次に重要なのが、ばらつき(dispersion)の程度である。よく用いられるのは、分散、ないしは標準偏差であるが、分散は平均からの偏差の２乗の平均であり、標準偏差はその平方根である。

身長のデータについて分散を計算すると

$s^2 = (149^2+\cdots+153^2)/60 - 157.583^2 = 24.687$　　粗データの場合

$s^2 = (146^2 \cdot 1+\cdots+170^2 \cdot 1)/60 - 157.60^2 = 24.40$　　度数分布表の場合

となる。

ファイル名：chap2_2.html------

```
<html><body>
<script>
function job2_3(){
document.write("<p>身長データ(粗データ)")
height=new Array(
149,153,153,155,153,165,155,160,165,155,
        中略
159,158,159,158,148,153,167,172,168,153)
result = meansd(height)
document . write ( " < p > Result ( Raw Data)")
document . write ( " < br > Mean = " , result [0])
document.write("<br>Var=",result[1])
document.write("<br>SD=",result[2])
document.write("<p>End<p>")
document.write("<p>Back ボタンで前ページに戻ります。<p>")
return }
function job2_4(){
document.write("<p>身長データ(度数分布)")
```

```
xh=new
Array(146,149,152,155,158,161,164,167,
170)
xf=new Array(1,2,8,17,13,8,7,3,1)
result = freqmsd(xh,xf)
document.write("<p>Result(Tabulated
Data")
document.write("<br>Mean= ",result
[0])
document.write("<br>Var= ",result[1])
document.write("<br>SD= ",result[2])
document.write("<p>End<p>")
document.write("<p>Back ボタンで前ページに戻ります。<p>")
 return }
</head></script>
<body>
<form>
<INPUT TYPE="button" VALUE="job2_3 平均・分散(粗データ)" onClick="job2_3()">
<INPUT TYPE="button" VALUE="job2_4 平均・分散(度数分布)" onClick="job2_4()">
</form>
</body></html>
```

身長のデータについて、平均・分散・標準偏差を計算する。

§3. 相関関係

【数値例 データ 2】女子学生の（身長、体重)(n=60)

(149、46)	(153、45)	(153、42)	(155、44)	(153、48)	(165、50)
(155、50)	(160、50)	(165、55)	(155、48)	(157、51)	(161、51)
(161、60)	(155、45)	(155、49)	(155、52)	(155、56)	(151、43)
(159、49)	(157、51)	(157、48)	(154、53)	(165、58)	(157、49)
(154、48)	(157、50)	(151、47)	(164、55)	(155、46)	(155、54)
(158、53)	(163、51)	(160、58)	(154、52)	(160、43)	(163、49)
(156、46)	(163、52)	(155、53)	(147、43)	(162、53)	(152、47)
(158、52)	(156、46)	(166、54)	(158、46)	(160、44)	(155、50)
(161、52)	(155、44)	(159、47)	(158、49)	(159、53)	(158、55)
(148、45)	(153、48)	(167、54)	(172、56)	(168、70)	(153、44)

3—1. 共分散・相関係数

データが 2 変数(x, y)について与えられているとき、両変数の関係を見ることができる。【データ 2】 は先の身長のデータ（表2.1）に体重を追加したものであり、両者をプロットしたものを散布図という。身長と体重の関係を視覚的に見ることができる。

2変数の共変関係を共分散で求め、それを標準偏差で規準化して、相関係数とする。

まず、粗データから共分散・相関係数を計算する。粗データについて、共分散と相関係数を求めると、

$$s_{xy} = 1/60\ (149 \cdot 46 + 153 \cdot 45 + \cdots + 153 \cdot 44) - 157.58 \cdot 50.03 = 115.364$$

$$r = 15.364/(4.968 \cdot 4.923) = 0.6283$$

となり、相関の程度はあまり高くはないが、身長と体重の間にはプラスの相関が見られる。

3—2. 相関表

粗データではなく、階級に分類したデータを用いて、2次元的に分類したものが相関表であり、相関表から、共分散、相関係数を計算することもできる。【データ 3】は身長ー体重についての相関表である。

【数値例 データ 3】身長・体重の相関表

身長		146	149	152	155	158	161	164	167	170	計
体重	43	1		3	2		2				8
	46		2	3	4	2					11
	49			2	5	5	1	2			15
	52				4	5	3	2			14
	55				2	1		2	2	1	8
	58						1	1			2
	61						1				1
	64								1		1
		1	2	8	17	13	8	7	3	1	60

問題 1 次の英語・数学の成績について、相関係数を求めよ。

英語
83, 80, 48, 68, 70, 45, 72, 28, 51, 32,
42, 38, 52, 80, 52, 78, 32, 60, 54, 49
数学
55, 42, 32, 71, 67, 60, 63, 51, 49, 51,
64, 15, 73, 71, 32, 68, 42, 55, 62, 31

問題 2. 次の相関表のデータについて、相関係数を求めよ。

	x	10	20	30	40	50
y	8	2	3	1		
	12	2	4	1		
	16		2	5	3	
	20		1	2	3	1

少しプログラムが大きくなるが、相関係数を計算する諸プログラムを合併させる。

ファイル名：chap2_3.html--------

```
<html><head><script>
function meansd(x) { s1=0 ; s2=0
 for (i=0 ; i<x.length ; i++) {
 s1=s1+x[i] ; s2=s2+x[i]*x[i] }
xm=s1/x.length
xsd=Math.sqrt(s2/x.length - xm*xm)
msd=new Array(xm,xsd)
return msd }
function vecsum(x) { sum=0
for (i=0 ; i<x.length ; i++)
sum=sum+x[i]
return sum }
function vecsum2(x) { sum2=0
for (i=0 ; i<x.length ; i++)
```

```
sum2=sum2+x[i]*x[i]
return sum2 }
function covariance(x,y) { cov=0
for (i=0; i<x.length; i++)
 cov=cov+x[i]*y[i]]
 cov=cov/x.length-meansd(x)[0]*
meansd(y)[0] }
 return cov }
function correlation(x,y) {
r=covariance(x,y)/(meansd(x)[1]*
meansd(y)[1])
 document.write(" <p>corr=",r)
return }
function fmeansd(xx,ff) {
 s1=0;s2=0;ns=vecsum(ff)
 for (i=0;i<xx.length;i++) {
 s1=s1+xx[i]*ff[i];s2=s2+xx[i]*xx[i]*ff[i] }
 xm=s1/ns
 xsd=Math.sqrt(s2/ns - xm*xm)
 msd=new Array(xm,xsd)
return msd}
function afmarginal(na,nb,f){
 fa=new Array(0,0)
 for (j=0;j<na;j++) {
fa[j]=0;for (k=0;k<nb;k++)
fa[j]=fa[j]+f[j*nb+k] }
return fa }
function bfmarginal(na,nb,f) {
 fb=new Array(0,0)
for (k=0;k<nb;k++) {
fb[k]=0;for (j=0;j<na;j++)
fb[k]=fb[k]+f[j*nb+k] }
return fb }
function cortab(x,y,f) {
na=x.length;nb=y.length
fa=new Array(2);fb=new Array(2)
 fa=afmarginal(na,nb,f)
 fb=bfmarginal(na,nb,f)
 ns=vecsum(f)
s12=0;
for (j=0;j<na;j++)
for (k=0;k<nb;k++)
 s12=s12+x[j]*y[k]*f[j*nb+k]
 cov=s12/ns—fmeansd(x,fa)[0]*
 fmeansd(y,fb)[0]
 r=cov/(fmeansd(x,fa)[1] *fmeansd(y,
fb)[1])

document.write("r=",r)
document.write("<p>")
 return }
</script></head>

<body><form><h3>女子学生の身長・体
重データから、相関係数を計算します。
</h3><INPUT TYPE="button" VALUE=
"job2_5 身長・体重
 (粗データ)" onClick="job2_5()">
<INPUT TYPE="button" VALUE="job2
_6 身長・体重 (相関表データ)" onClick
="job2_6()">
<br>
<h3> 粗データ、相関表データから、相
関係数を計算します。</h3>
<INPUT TYPE="button" VALUE="job2
_7 英語・数学(粗データ)" onClick="job
2_7()">
<INPUT TYPE="button" VALUE="job2
_8 X・Y(相関表データ)" onClick="job2_
8()">
</form>
```

```
<script>
function job2_3) {
// 女子学生の身長・体重データ
height=new Array(
149,153,153,155,153,165,155,160,165,155,
    前項と同じにつき、中略
159,158,159,158,148,153,167,172,168,153)
weight=new Array(
46,45,42,44,48,50,50,50,55,48,
 51,51,60,45,49,52,56,43,49,51,
 48,53,58,49,48,50,47,55,46,54,
 53,51,58,52,43,49,46,52,53,43,
 53,47,52,46,54,46,44,50,52,44,
 47,49,53,55,45,48,54,56,70,44)
correlation(height,weight)
document.write("<p>Back ボタンで前ページに戻ります。<p>")
return }
function job2_4) {
// sincho
xh=new Array(
146,149,152,155,158,161,164,167,170)
yw=new Array(
43,46,49,52,55,58,61,64)
cortabf=new Array(
 1,0,3,2,0,2,0,0,0,
 0,2,3,4,2,0,0,0,0,
 0,0,2,5,5,1,2,0,0,
 0,0,0,4,5,3,2,0,0,
 0,0,0,2,1,0,2,2,1,
 0,0,0,0,0,1,1,0,0,
 0,0,0,0,0,1,0,0,0,
 0,0,0,0,0,0,0,1,0)
cortab(xh,yw,cortabf) document.write
("<p>Back ボタンで前ページに戻ります。
<p>")
return }
function job2_7() {
eng=new Array(
83,80,48,68,70,45,72,28,51,32,42,38,5
2,80,52,78,32,60,54,49)
math=new Array(
55,42,32,71,67,60,63,51,49,51,64,15,7
3,71,32,68,42,55,62,31)
correlation(eng,math)
document.write("<p>Back ボタンで前ページに戻ります。<p>")
return }

function job2_8() {
x=new Array(10,20,30,40,50)
y=new Array(8,12,16,20)
cortabf=new Array(2,3,1,0,0,
                  2,4,1,0,0,
                  0,2,5,3,0,
                  0,1,2,3,1)
cortab(x,y,cortabf)
document.write("<p>Back ボタンで前ページに戻ります。<p>")
return }
</script></body></html>
```

IV. 統計学／心理統計法のためのユーザー関数一覧

第 II 部では、統計学教科書を中心に初等統計処理の関数を提供したが、ここでは、さらに幅広く、各種の統計処理について、ユーザー関数を一覧するものである。部分的に前項と重なって再録されているが、本稿はデータベース的使用の便宜を図ったものである。基礎統計学(検定を含む)、多変量解析、心理学的測定法、およびそれらの下位項目について、分類する。

見出し後の HTML ファイル名は、類似の関数を一括保存するためのファイル名である。それのないものは、例えば、多変量解析など、一つ一つが膨大であるので、個々の手法名で保存されている。ディレクトリ名で、区別されている場合もある。

現在構築中であり、完成されたものではない。必要に応じて今後追加される。また、十分にデバッグを行ったつもりであるが、皆無とは言えず、今後努力が必要である。従って、関数群は常時、書き換えられている。

ここでは、紙数の関係から各々についての詳細な説明は省略するが、関数の冒頭部にコメントがあり、関数名のフルネーム、引数、出力などに関して、簡単に解説されている。ソースを閲覧することにより、内容を知ることができよう。

引数のない関数は、関数内部でデータを発生させるものであり、出力も関数内でプリントして終了する場合と、戻り値として、呼ばれたルーチンに返すものとがある。個々にソースに書かれている。引数 x は、スカラ、ベクトル、行列、およびそれらの複合など、様々である。個々の関数毎に注意が必要である。

JavaScript はあくまでも簡易言語であって、本格的言語ほど強力ではない。メモリ、速度、信頼性（時として、文法チェックを間違い、ハングアップする）にも難点があるので、そのことを意識して使用しなければならない。

電卓代わりとして、小さい計算のチェックには十分であるが、大きいデータの処理には不向きである。

以下、主なものいくつかについて、関数名一覧と解説を試みる。

[1]. 基礎統計学（basestat. html）

初等統計学で学習される統計計算をまとめた。前半は準備としての数学的関数、行列演算と記述統計の関連する部分であり、後半は検定などの推測統計学に関するものである。

実際の計算に際しては、必要な関数を取り込み、処理するデータと、関数を呼ぶための文を書いたメインルーチンを用意しなければならない。すなわち、次のような構成の HTML ファイルを作成し、そのファイルをブラウザでローカルに読み込めば、処理が行われる。

計算のための HTML ファイルの構成

```
<html><body>                y=data
<script>                    e=abc(y)
funcion abc(x) {             </script>
 ---                         </body></html>
return  d }
```

メインルーチンでデータを用意し、処理は関数内で行う。結果は、関数内でプリントして終了してもよいし、関数の戻り値として、呼ばれたルーチンに持ち返ってもよい。返す値が複数個あるときは、それらをまとめて配列とする。

1．数学的基礎(math.html)

1—1．関数

sum1n(n)　1から n までの整数の和
sumab(a,b)　a から b までの整数の和
factorial(n)　整数の階乗
gamma(x)　ガンマ関数
beta(a,b)　ベータ関数

1—2．積分

simpson(n,x,f)　シンプソンの積分

1—3．反復法

sqrt(a)　Newton=Raphason 法

1—4．ベクトル・行列演算(matrix.html)

sum(x)　ベクトルの要素の和
sum2(x)　ベクトルの要素の 2 乗和
x.length　ベクトル x の大きさ
minimum(x)　ベクトル x の要素の最大値
maxmum(x)　ベクトル x の要素の最小値
TakeCol(x,na,nb,i,jj) 2 次元配列から特定の列を抽出
TakeRow(x,na,nb,ii,j)) 2 次元配列から特定の行を抽出
MatTrans(x,xt,II1,JJ2)　行列の転置
VecProduct(x,y)　ベクトルの内積
MatAdd(a,b,c,l,m)　行列の和
MatMul(a,b,c,l,m,n)　行列の積
MatInv(a,ai,n)　逆行列
QuadForm(n,aa,xx,bb)　2 次形式

2．データの整理(base.html)

2—1．1変数処理

freq(x,a1,c)　度数分布表の作成
mean(x)　平均
xsd(x)　標準偏差
meansd(x)　平均・標準偏差
pmom(x,p)　p 次の中心積率
amean(x)　算術平均
gmean(x)　幾何平均
hmean(x)　調和平均
freqmean(x,f)　度数分布表の平均
freqsd(x,f)　度数分布表の標準偏差
freqmeansd(x,f)　度数分布表の平均・標準偏差
zscore(xx,xm,xsd)　データの標準化
zs(n,p,x)　正規近似

2—2．変数処理

covariance(x,y)　共分散
corr(x,y)　相関係数
cortab(x,na,nb)　相関表の処理、

平均・標準偏差・相関係数

2—3. 多変数処理

covmat(ns, np, x)　分散・共分散行列
cormat(ns, np, x)　相関行列

2—4. 順序量

sort(x)　ベクトル要素の並べ換え
ptile(x)　パーセンタイル順位
med(x)　メディアン
Q(x)　Q1, Q2, Q3

3. 統計学的分布(pdist. html)

3—1. 確率分布

binom(n, p, x)　二項分布
poisson(m)　ポアッソン分布
geometry(p, x)　幾何分布
hypgeo(n, m, a, x)　超幾何分布
negbinom(a, p)　負の二項分布

3—2. 標本分布(sdist. html)

zs(z)　z-score
zlprob(z)　z—下側確率
chidist(chi, df)　χ^2—分布
tdist(t, df)　t—分布
fdist(f, df1, df2)　F—分布
normal()　数表(正規分布表)

4. 区間推定(interval. html)

mint1(x, sig2, n, g, z)　母平均の区間推定(母分散既知)
mint2(x, s2, n, g, t)　母平均の区間推定(母分散未知)
sint(s2, n, g, chi1, chi2)　母分散の区間推定

5. 検定 (stest. html)

5—1. 平均・平均の差

mtest1(x, sig2, n, m)
平均の検定(母分散既知)
mtest2(x, s2, n, m)
平均の検定(母分散未知)
mdiftest1(x1, sig1, n1, x2, sig2, n2)
平均の差の検定(母分散既知)
mdiftest2(x1, s1, n1, x2, s2, n2)
平均の差の検定(母分散未知, 等分散)
mdiftest3(x1, s1, n1, x2, s2, n2)
平均の差の検定(母分散未知, Welch)
mdiftest4(x1, s1, n1, x2, s2, n2)
平均の差の検定(サンプル数大)

5—2. 分散・分散比

function vtest(s2, n, sig2)　分散の検定
vratiotest(s1, s2, n1, n2)　分散比の検定
bartlet(v, nn)　分散の等質性の検定(Bartlet)

5—3. 相関係数

corttest(r, n)　相関係数の検定
fcorztest(r, ro, n)　相関係数の検定(Z—変換)
cordiftest(r1, r2, n1, n2)
相関係数の差の検定

5—4. 比率

pratio(p, n, p0)　比率の検定
pdiff(p1, p2, n1, n2)　比率の差の検定

5—5. クロス表

ctab22(a, b, c, d)　(2x2)分割表の独立

	性検定
ctabab(x, a, b)	(axb)分割表の独立性検定
goodfit(f, e)	適合度検定
fisher(x)	Fisher の直接確率

6. 分散分析（anova.html）

anova(na, nn, ax)	一元配置分散分析
contrast(na, nn, ax2)	多重比較
anova2(na, nb, nn, ax)	二元配置分散分析
ancova(na, nn, ax, ns)	一元配置共分散分析

[2]. 多変量解析

多変量解析の手法は多くのアプリケーションに用意されており、今さら屋上屋を重ねる必要もないが、簡単なものについて、プログラミングを行った。

多変量解析といっても行列演算の組み合わせであって、行列の積、逆行列、固有値を計算する関数を用意しておけば、ユーザー関数として作成することは、さほど困難な作業ではない。しかし、JavaScript ではやや計算は重く、多くのオプションを備えた総合的なものは困難であるので、よく使用されるもののみを取り上げ、最も簡単なメニュに限ってプログラミングを行った。計算のアルゴリズムを理解するための、教育的目的の関数群である。

7. ベクトル・行列演算2(matrix2.html)

det(n, x)	行列式
minor(x)	小行列
cofactor(x)	余因子
trimat(x)	三角行列
Jacobi(n, a)	固有値・固有ベクトル

8. 多変量解析(mulva.html)

hotell(x)	Hotelling's T^2 統計量
mahadist(n, xi, xj, vc)	Mahalanobis の距離
mndist(x)	多変量(標準)正規分布
regression(x, y)	回帰分析
mreg(x, y, ns, np)	重回帰分析
disc(ns,np,ng,gr,xx)	判別分析
pca(ns, np, xx)	主成分分析
varimax(np, nf, a)	バリマックス回転
procrustes(np, nf, a, c)	プロクラテス回転
score(ns, np, nf, x, a)	因子得点
cluster(x)	クラスター分析
manova(x)	多変量分散分析

[3]. 心理学的測定法

統計学の心理学における応用の一例として、心理学的測定法を取り上げる。心理学実験および心理学的社会調査によって得られた各種のデータは、その特性に応じて様々な統計処理が行われるが、代表的な方法についてプログラミングを行い、データ処理の便に付した。領域として、精神物理学的測定法、尺度構成法、テスト理論を取り上げるが、問題はこれらの領域のみに尽きるものではない。本稿での例をもとに、個別のテーマに応じて、各自ユーザー関数を作成し、効率的なデータ処理が行われることが望まれる。

9. 精神物理学的測定法（physics, html）

mconst(x)	恒常法
mlimit(x)	極限法
madjust(x)	調整法

10. 尺度構成法（scaling. html）

mpair(x)	一対比較法
mlikert(x)	リッカート法
morder(x)	順位法

11. テスト理論（ttheory. html）

alpha(x)	アルファ係数
kr20(x)	K=R 公式
sb(x)	S=B 公式
gpanal(x)	G-P 分析

心理学的実験の制御、学習過程の分析等に関して研究されているが(11～17)、言語 JavaScript は、コンピュータのハードウエアにタッチできず、必ずしも実験制御に最適に構成されているとは思われないので、ここでは、データ処理にのみ止める。現在においても、実験の制御に適した言語は、BASIC ないしは C－言語であろうか。データが送られてきた後、 JavaScript によりデータ処理が始まるという仮定で、以上の諸関数が活用される。

[4]. 休憩室(rest. html)

subj()	主観的確率	主観的確率に関する仮想実験
mpai(n)	π の値(モンテカルロ法他)	π の値の算出
randnormal(n)	正規乱数	正規乱数の発生
clt()	中心極限定理	定理のシミュレーション
randpoisson(n)	ポアッソン乱数	ポアッソン乱数の発生と再生性
randchi(n)	χ^2－乱数	カイ２乗乱数の発生と再生性
randt(n)	t－乱数	t－乱数の発生

上記ユーザー関数およびそれらを組み合わせて作成したプログラムの中から、いくつかを例として取り上げる。ソースは紙数の関係上省略するが、ブラウザで閲読可能である。

例１：平均の差に関する検定

正規母集団による平均の差の検定は大きく４つに分けられる。

1. 母分散が既知の場合	正規検定
2. 母分散未知で等しい場合	t-検定
3. 母分散未知で等しくない場合	t-検定(Welch)の方法
4. サンプル数大の場合	正規検定

それぞれの場合における、数値例(5より)をメインルーチンから読み込む。平均、分散、サンプル数である。数値例に対応する個所を、当該処理用データに置き換えればよい。1～4 のいずれの公式を用いるかは事前に設定しておく。

例２：重回帰分析

多変量解析のうちの代表的な手法は、重回帰分析であろう。連続量の独立変数 X から、連続量の従属変数 y を線型予測して求めものである。回帰係数 $b=(X'X)^{-1}X'y$、y の予測値= $\hat{y}$ 、重相関係数=R_m、残差平方和=$R^2{}_0$、などの基本的なもののみを出力する。正規モデルを想定して、重相関係数、回帰係数の検定も行う。また、独立変数の選択に際して必要な AIC(r) も算出しておく。

例３：一対比較法

選好比率行列をもとに、Thurstone の比較判断の法則により、刺激の尺度値を算出する。

先ず CaseV により尺度値を求め、期待比率を算出して、角変換により原データとの適合度検定を行う。時としてあてはめが悪いので、その時は、分散一定の仮定をはずし、分散を推定して CaseIII の場合について解く。

尺度値の算出に際して、正規変換を行うので、標準正規分布の z－値，分布関数を算出する関数が必要である。また、粗データにおいては、

A>B で B>C のとき、C>A

と循環する反応もあり、一概に誤反応とも言えないが、一応そうした回答を発見・抽出する関数 concheck(np, nf) も作成した。

例４：シミュレーション：χ^2－分布

一様乱数をもとに各種乱数を発生させることができる。ここでは、その一例として

一様乱数 --> 正規乱数 --> カイ2乗乱数

を発生させ、ヒストグラムを描く。

1. (0,1)一様乱数を np1 個発生させて、それらを加える。
2. その 平均=np1/2、分散=np1/12 であるので、それを用いて標準化を行う。
3. この変数は中心極限定理により、正規乱数である。
4. 自由度=n のカイ２乗変数の定義は、正規変数の n 個の２乗和であるので、この正規乱数を２乗して np2 個加える

   $Z \sim N(0,1)$ のとき $X^2 \equiv (\mathrm{def}) \Sigma Z^2$
5. これはカイ２乗乱数であるので、それを np3 個発生させ、度数分布に集計し、ヒストグラムを描く。

6. 平均、分散を計算し、カイ２乗乱数となっているか検討 （V(X)＝2 E(X)）。

V. まとめ

統計学学習支援およびデータ解析用ツールとして JsStat(Statistics by JavaScript) を構築した。 JavaScript は主要インターネットブラウザに標準搭載されている、Web ページ構築用の簡易言語であり、入手が容易である上に、プログラミング技法も簡単で、かっての Basic よりも簡単にプログラミングできる。

さらに、それに加えて、数学関数も強力に用意されており、それらを用いれば、統計計算をも容易に行うことができる。 多くの統計計算は、コンピュータの普及により統計ソフトを多用して処理されているが、計算の中身はブラックボックス化され、ユーザーには見えず、こうした状況は統計学の学習に際しては適切ではない。

本システムは、統計学の学習に平行して、必要な計算は直ちに、自らプログラミングして、処理することが出来るよう開発されているので、それらを組み合わせて使用することにより、教育効果が発揮される。また、高度の関数群を組み合わせることにより、中程度までのデータ解析を行うこともできる。

主な内容は以下のごとくである。

統計学電子化教科書：一例として、データの整理（平均、分散、相関計数）の章を例示
統計学ユーザー関数：記述統計学、確率分布、標本分布、推定、統計的検定、多変量解析の基礎的手法、統計シミュレーション、心理統計法（尺度構成、精神物理学的方法）
他サイトへのリンク（オンライン統計学書、統計処理ソフト、主要統計学研究室）など。

文 献

（1）Darnel, R.，ネティズム工房訳，JavaScript クイックリファレンス，プレンティスホール，1997.
（2）Goodman, D. 田邉茂也，各務辰哉訳，JavaScript ハンドブック，オーム社，1996.
（3）川崎克己，JavaScript マスターバイブル，みずき，1996.
（4）Kent, P. & Kent, J.，インターナショナル・トムソン・パブリッシング・ジャパン訳，Official Netscape JavaScript，BNN，1996.
（5）Lemay, L. and Moncur, M.，ネティズム工房訳，JavaScript 1.1 プログラミング入門，プレンティスホール，1997.
（6）Shobe, M.，Ritchney, T，SE 編集部訳，入門 JavaScript，翔泳社，1996.
（7）Simpson, A.，インターナショナル・トムソン・パブリッシング・ジャパン訳，

Official Netscape Navigator Gold 3.0, BNN, 1996.
(8) 上田学, JavaScript, オーム社, 1997.
(9) 稲垣宣生・山根芳知・吉田光雄, 統計学入門, 裳華房, 1992.
(10) Lane, D.M., HyperStat : Hypermedia for Learning Statistics and Analyzing Data., Academic Press, 1993.
(11) Atlas, R. et al., The use of animation in Software Training : Pitfalls and Benefits. In Quinos, M. and Dutta, A., (Eds.) Training for 21st Century Technology, 1997.
(12) Guadagno, N. et al., The predictability of commands in a spreadsheet program. Interacting with Computers, 2, 75−82, 1990.
(13) Napier, A. et al., Knowledge of command usage in a spreadsheet program : Impact on user interface design and training, Data BASE, 23, 13−21, 1992.
(14) Lane, D.M. & Ashby, B., PsychLib : A library of machine-language routines for controlling psychology experiments on the Apple Macintosh Computer. Behavior Research Methods Instrumentation and computers, 19, 246−248, 1987.
(15) Martel, R. et al., Male-female differences : A computer simulation. American Psychologist, 51, 157−158, 1996.
(16) Pirolli, P., & Anderson, J.R., The role of learning from examples in acquissition of recursive programming skills., Canadian Journal of Psychology, 39, 240−272. 1985.
(17) VanLehn, K., Arithmetic procedures are induced from examples. In Hiebert, J. (Ed.), Conceptual and procedural knowledge : The case of mathematics, 130−180, Hillsdale, 1986.

URL(インターネット関連サイト。一例であり、体系的にリストアップするものではない。また、随時書き換えられる場合がある。)

JavaScript 情報
(18). http : //home. netscape. com/eng/mozilla/3. 0/handbook/javascript/
(19). http : //www. sapphire. co. uk/javascript/ − JavaScript Index

JavaScript レファレンス・プログラム集
(20). http : //www. shiojiri. or. jp/~openspc/JavaScript/
(21). http : //www. ueda. info. waseda. ac. jp/~gaku/js/
(22). http : //www. geocities. com/ResearchTriangle/1828/resources. html
(23). http : //www. webreference. com/javascript/
(24). http : //www. math. grin. edu/~rebelsky/Tutorials/javascript/EdMedia97/refs. html
(25). http : //www. acmuiuc. edu/webmonkeys/javascript/objects. html
(26). http : //www. starlingtech. com/books/javascript/examples. html
(27). http : //user. cnet. or. jp/o/openspc/OpenSpace/JavaScript/lib/JSlib. html#top
(28). http : //www. acm. uiuc. edu/webmonkeys/javascript/objects. html

オンライン統計学
(29). http : //koko15. hus. osaka-u. ac. jp/~yoshida/jsstat/

(30). http : //www. ruf. rice. edu/~lane/hyperstat/intro. html
(31). http : //www. ruf. rice. edu/~lane/network_classroom/network_classroom. html
(32). http : //www. stat. ucla. edu/textbook/
(33). http : //www. psychstat. smsu. edu/sbk00. htm/
(34). http : //surfstat. newcastle. edu. au/surfstat/gateway. html
(35). http : //surfstat. newcastle. edu. au/surfstat/main/surfstat–main. html
(36). http : //stat. tamu. edu/~jlethen/30Xnotes/trydouble2. html
(37). http : //olam. ed. asu. edu/%7eglass/502/home. html
(38). http : //phoenix. som. clarkson. edu/~cmosier/statistics/lectures. html
(39). http : //www. dist–ed. nhc. edu/facmats/cbiafore/mat220/m220htm/outline. htm

統計学 Library・アプリケーション

(40). http : //lib. stat. cmu. edu/
(41). http : //www. stat. unipg. it/IASC/SW–archives. html
(42). http : //www. stat. unipg. it/iasc. html
(43). http : //www. stat. ucla. edu/consult/theforum/w–agora. phtml

統計学研究室(世界の主要 Departments of Statistics へのリンク)

(44). http : //lib. stat. cmu. edu/otherplaces/. index. html

JsStat : Introduction to Statistics by JavaScript

Mitsuo Yoshida

JavaScript is a script language to customize user Web homepages working on Internet browsers such as N. Navigator and M. Internet Explorer of the recent versions, used as a powerful tool to modify letters, figures, background colors, and every design and layout of the pages.

A variety of mathematical functions and all kinds of numerical opereations enough to statistical purposes are also defined and on that reason any necessary ones are easily constructed as user functions when insufficient.

Java (different system from JavaScript) is a full platform with sufficient security and object-oriented programming styles, which means difficult for common computer users. However, JavaScript is an easy language and offers a good platform for beginners as a former language, BASIC, popular with early personal computers.

A statistical system, named JsStat (Statistics by JavaScript) has constructed for an elementary statistics textbook and as a collection of user-functions for general data analyses especially for many methods of psychological statistics.

They are already debugged and tested working on any browsers with JavaScript, but some bugs are still supposed to be left in the system.

The main contents are as follows : (1). Electronic elementary statistics textbook,
(2). Statistical user-functions, including descriptive statistics, estimation, hypothesis testing, fundamental methods of multivariate analysis, statistics simulations, and psychological methods (scaling and psychophysics), and (3). Links to online statistics books, application libraries and main university's departments of statistics.