【セミナー】

SPSSによる社会調査データ分析入門

― シンタックスの解説を中心に ―

小林　久高・雨森　聡・山本　圭三

KOBAYASHI Hisataka, AMENOMORI Satoshi,YAMAMOTO Keizo

1　はじめに

SPSSを使い始める理由は「授業での課題遂行やレポート・論文の執筆等でどうしても使わなければならないから」というものが大半だろう。当たり前の話だが、SPSSを一度も使ったことのない人にとって、それを使いこなすことは難しい。難しい上に、困ったときにヘルプを見ても書いてある意味がわからないことが多い。そもそも知りたいことが何であるかが自分でもわからないので、適切なヘルプにたどりつけないというのが実情かもしれない。

ヘルプを見ても問題が解決しないときにおこなうことは、参考になる図書を読むか、インターネットを利用して調べるかである。SPSSの使い方に関する図書はたくさん出版されているし、大学の研究室、とくに心理学や社会学の研究室のウェブサイトには参考になるものも多い。

ところで、出版されている図書やウェブサイトの解説では、分析に際してSPSSのメニューバーやダイアログが利用されることが多い。この方法は、分析手順が明確であり、視覚的にわかりやすいので、SPSS初心者にとっては非常にありがたい。しかし、メニューバーやダイアログを用いる方法以外にもシンタックスを用いて分析することも可能である。シンタックスとはSPSS上で作成するプログラムのことを指す。分析者は、シンタックス（プログラム）をシンタックスエディタに書き込み、それを実行することによって、データの変容や分析をおこなうことができる。

シンタックスを使用する利点のひとつは、分析の再現を容易にできる点である。レポートや論文を執筆する際、分析は繰り返しおこなわれるものである。幾度も分析をおこなっていると、どのような分析をおこなったかという記憶があいまいになる。記憶があいまいになったとしても、シンタックスが保存されていれば、簡単に分析を再現できるわけである。

シンタックスを使わずに、メニューバーやダイアログを用いて分析し、分析結果（アウトプット）を保存すればよいと思う人がいるかもしれない。しかし、分析結果だけでは、分析に対しておこなった細かい指定がわかりづらい。また、アウトプットファイルは、テキスト情報のみのシンタックスファイルと異なり、容量が大きくなるので持ち運びにくい。それゆえ、分析結果を保存する方法はあまりお勧めできない。

ところで、メニューバーやダイアログを用いる方法が普及してきたため、かつてはいくつか出版されていた「シンタックスの解説を含むSPSSのマニュアル」は現在手に入りにくい状態になっている。そこで、本稿ではシンタックスを用いたSPSSの使用方法について入門的な解説を行いたい。本稿の想定している読者は、社会調査実習などでSPSSを使う必要のある学生や、SPSSを使って間もない人である。

2　SPSS の基本事項

2.1 データエディタ

SPSS では、(1) SPSS データエディタ、(2) SPSS ビューア（出力）、(3) SPSS シンタックスエディタの 3 つのウィンドウが主に用いられる。

SPSS データエディタとは、SPSS が分析するデータを表示するものである（図 1)。このデータエディタは、SPSS を起動すると自動的に表示される。

SPSS はこのデータエディタにあるデータを分析する。このため、データエディタに直接データを入力するか、エクセルなどで保存されているデータをこのデータエディタに読み込ませるかしなければ、SPSS で分析することはできない。

2.2 ビューア

SPSS ビューアとは、分析結果を表示するウィンドウである（図 2)。分析を実行するとビューアは自動的に開かれ、分析結果はビューア内に出力される。

ビューアのウィンドウは左右 2 つの部分に分かれている。左側がいわば目次になっており、その目次をクリックすると、目次に対応した文字や分析結果が右側に表示される。

多くの分析結果は表形式で SPSS ビューアに出力される。これらの表はワードなどに画像として貼り付けることができるので都合がいい。さらに便利なのは、表を選択しエクセルに貼り付けると行列がきれいにワークシートに収まるということだ。印刷上の細かい設定が必要な表を作成する際には、(1) SPSS ビューアの表をエクセルに貼り付けて、(2) エクセル上で表の体裁を整え、(3) それを画像としてワードなどに貼り付けるのがよいだろう。

図 1　SPSS データエディタ

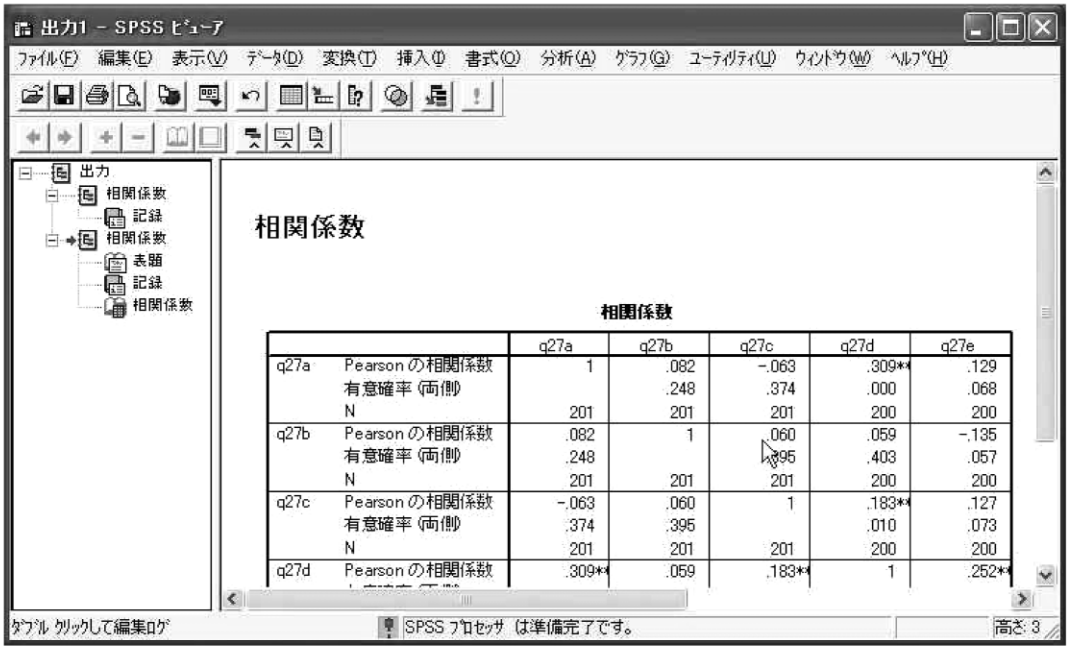

図 2　SPSS ビューア

学生のレポートだけでなく専門雑誌に収録されている論文においてさえ、ビューアに出力された表をそのまま本文に貼り付けられていることがある。ビューアに出力された表の中にある値は全てが必要でないことが多く、また、その表は見にくいものが多い。表には必要な情報だけを掲載し、値や罫線の見栄えをよくした表を作成したほうが、読者に対して親切だろう。ビューアに出力された表をそのまま本文に貼り付けるのではなく、エクセルで体裁を整えたものを貼り付けることをお勧めする。

2.3 シンタックスエディタ

先にも述べたように、シンタックスエディタに分析プログラムを書き、それを実行することで分析をおこなうことができる。このシンタックスエディタは、通常は自動的に表示されない。シンタックスエディタの起動方法は 2 通りある。1 つ目は、SPSS を起動させた後に、メニューバーの「ファイル」を開き、その中にある「新規作成」、さらにその中にある「シンタックス」を選択する方法である。いま 1 つは、作成済みのシンタックスファイルを起動する方法である。

シンタックスエディタに書くシンタックスは、たとえば図 3 のような形になる。ここで、「frequencies」の部分は、SPSS に実行させる命令のことで、コマンドと呼ばれている。frequencies コマンドは、「度数分布表を出力せよ」という意味である。

「variables」や「statistics」の部分は、サブコマンドと呼ばれている。コマンドにサブコマンドを付随させることによって、より細かい作業を命令することができる。「variables」は、コマンドを実行させる変数を指定するサブコマンドで、「statistics」は、コマンドを実行させる際に同時に出力させる統計量を指定するサブコマンドである。

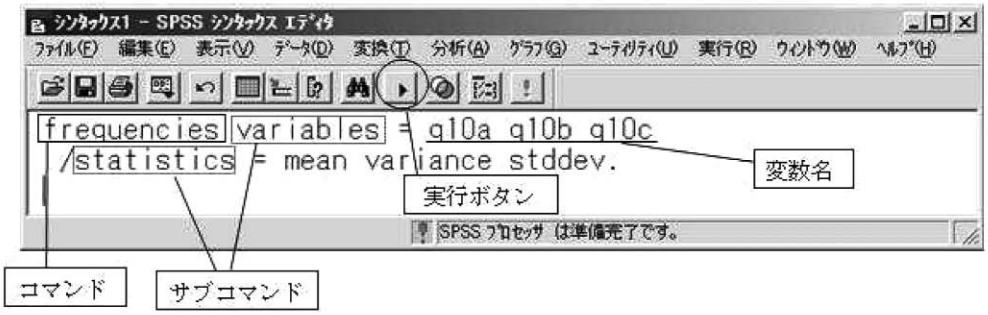

図 3　シンタックスの例

図 3 に示したシンタックスを読解すると、「q10a、q10b、q10c という名前の変数について、度数分布表を出力せよ。なお、平均、分散、標準偏差も同時に出力せよ。」という意味になる。

2.4 シンタックス使用時の基本事項

シンタックスを用いる際にはいくつか注意すべき点がある。

(1) 1 つのコマンドの終わりには、「.（ピリオド）」を入れる。

この「.」は、1 つのコマンドの終わりを SPSS に伝えるために必要な記号である。コマンドの末尾に「.」を入れずにコマンドを実行すると、SPSS にコマンドの終点を伝えることができず、結果的にエラーが表示される。

(2) シンタックスは、基本的に半角英数（大文字、小文字を問わない）の文字を用いる。

コマンドやサブコマンド内に全角文字があるとエラーが表示される。一見ミスのないコマンドを実行したのに、エラーが表示されるときがあるが、このときのたいていの原因は、どこかに全角のスペースが入っているという単純なものである。
変数の定義などのときに、全角文字を「'性別'」のように、半角の「'（クォーテーションマーク）」で囲んで用いることがある。このようにすれば、例外的に全角文字を用いることができる。

(3) サブコマンドを複数個入力する場合は、「/（スラッシュ）」で区切る。

コマンドによっては、サブコマンドをいくつか付け足して細かい指定をしたほうがいいときがある。このとき、図 3 の「statistics」の直前にある「/」のように、サブコマンドとサブコマンドの間に「/」を入力する必要がある。

(4) 分析結果（アウトプット）の中にシンタックスで実行したコマンドを表示する。

SPSS がインストールされた状態だと、コマンドを実行しても、どのようなコマンドが実行されたかアウトプットに表示されない。実行したコマンドをアウトプットに表示させるには、データエディタのメニューバーにある「編集」を開き、その中にある「オプション」を選択する。すると、オプションのダイアログが表示される。そのダイアログの「ビューア」タブを選択し、「ログの中にコマンドを表示」のチェックボックスをオンにする（図 4）。そうすれば、実行したコマンドがアウトプットに表示されるようになる。

図 4　コマンドをアウトプットに表示させる

では、コマンドの説明に入ろう。

3　データの読み込み・定義・変容・保存

3.1 データの読み込み

(1) get data

データは、SPSS で直接作成することもできるが、たいていの場合、エクセルなどで作成したものを読み込む。まずは、エクセル形式のファイルを読み込む方法について。

【コマンド：get data】

```
get data
    /type = xls
    /file = 'ファイルの場所¥ファイル名.xls'
    /sheet = name 'シート名'
    /cellrange = range '開始セル:終了セル'.
```

「type」は、読み込むデータの形式を指定するサブコマンドである。エクセル形式のデータを読み込む場合は、「xls」と指定すればよい。「file」は、読み込むファイルのある場所、読み込むファイル名を指定するサブコマンドである。「sheet」は、データが入力されているシート名を指定するサブコマンドである。エクセルファイルを指定しても、ブック内のどのシートを読み込むか、ということまでを指定しなければならない。「cellrange」は、データが入力されているセルの範囲を指定するサブコマンドである。「開始セル」にはエクセル上の左上端のセル番地（たいていは A1）を、「終了セル」は最終ケースの最後の変数が入っているセルのセル番地を入力すればよい。例えば終了セルが「II300」の場合、</cellrange=range' A1: II300'>となる。

シンタックスエディタに以上のコマンドを入力して実行すれば、SPSS のデータエディタにデータが読み込まれる。

(2) data list

次に、テキスト形式のファイルを読み込む場合について。ここで述べるテキスト形式のデータとは、テキストに数値のみが入力されたものである。エクセルなどで入力したデータを、タブ区切りのテキスト形式で保存したものとは異なる（図 5)。

図 5　テキスト形式のデータ

【コマンド：data list】

```
data list file='ファイルの場所¥ファイル名.txt'
    records=2
    /1      変数 A   1
            変数 B   2-4
            変数 C   5
    /2      変数 D   1-4.
```

「file」は、読み込むファイルのある場所とファイル名を指定するサブコマンドである。「records」は、1 ケースのデータが何行にわたっているかを示すものである。

「/1」と「/2」は、変数定義のサブコマンドである。図 5 のように、テキスト形式のデータでは、数値が変数の区別なく並ぶかたちになっている。このため、どの変数が何桁目から何桁目までなのかを、変数ごとに指定しなければならない。変数の範囲を指定し、変数名を付ける作業をおこなうのが、「/1」と「/2」のサブコマンドなのである。

「/1」の部分は変数の定義の対象となっているデータの行を指定している。「/1」の隣にあるのが定義する変数の名前であり、後ろの「1」はその変数の値とする範囲を桁番号で指定するものである。したがって、＜/1 変数 A 1＞とは「1 行目の 1 桁目を『変数 A』と定義する」という意味である。変数の範囲が 1 桁でない場合は、「2-4」のように間をハイフン (-) でつないで範囲の指定をすればよい。データの行が複数に渡っている場合は、例に示した「/2」のように、適宜スラッシュと次に定義の対象となる行番号を入力することになる。

以上のコマンドを実行すれば、SPSS のデータエディタにデータが読み込まれる。

3.2 データの定義

(1) variable labels～変数の定義

SPSS にデータが読み込まれた後は、変数にラベルを貼る作業が必要となる。変数にラベルを貼らなかった場合、分析結果はデータを読み込んだときのままの変数名で出力されるので、非常に読み取りにくい。変数にラベルを貼らなくても分析に支障はないが、どのような変数が使われているかを一目でわかるようにするためには、ラベルを貼っておいたほうがよいだろう。

【コマンド：variable labels】

```
variable labels     変数 1   '変数ラベル 1'/
                    変数 2   '変数ラベル 2'.
```

変数ラベルは、半角の引用符（‘）で囲う。この場合、「変数 1 には変数ラベル 1、変数 2 には変数ラベル 2 というラベルを貼れ。」という意味である。

【具体例】

```
variable labels     q01 '性別'/
                    q02 '居住形態'.
```

具体例は、「q01 には性別というラベルを、q02 には居住形態というラベルを貼れ。」という意味である。

(2) value labels～値ラベルの定義

変数だけでなく、変数の値に対してもラベルを貼ることができる。変数の値にラベルを貼れば、分析結果を見るときに、値が何を意味しているのかすぐに理解できる。値ラベルは、カテゴリカル変数には貼っておいたほうがよい。

【コマンド：value labels】

```
value labels      変数 1   値 1'値ラベル 1'
                           値 2'値ラベル 2'/
                  変数 2   値 1'値ラベル 1'
                           値 2'値ラベル 2'
                           値 3'値ラベル 3'.
```

変数ラベルと同様に、値ラベルも半角の引用符（‘）で囲う。上記の例の場合、「変数 1」の「値 1」に対し「値ラベル 1」、「値 2」に対して「値ラベル 2」というラベルを貼ることを意味している。

【具体例】

```
value labels      q01   1'男'
                        2'女'/
                  q02   1'自宅'
                        2'下宿'.
```

具体例は、「変数 q01 の値について、1 に男、2 に女というラベルを貼れ。また、変数 q02 の値について、1 に自宅、2 に下宿というラベルを貼れ。」という意味である。

(3) missing values～欠損値の定義

missing values コマンドは、特定の値（例えば、無回答）を欠損値として定義するときに用いられる。

【コマンド：missing values】①

```
missing values  変数(値).
missing values  変数 a  変数 b  変数 c(値).
missing values  変数(値 1,値 2).
```

指定する欠損値が同じ場合、スペースを空けて変数名を並列しておけば一度に指定することができる。また、複数の値を欠損値にしたい場合は、カッコ内にカンマで並列すればよい。ただし、欠損値として一度に指定できるのは 3 つまでである。

【具体例】

```
missing values q01 q02(9).
missing values q03(7,8,9).
```

具体例は、「q01 と q02 について、9 を欠損値とせよ。」、「q03 について、7、8、9 を欠損値とせよ。」という意味である。

欠損値の指定方法として、1 つずつ値で指定する方法の他に、「thru」を用いて特定の範囲で指定する方法がある。

【コマンド：missing values】②

```
missing values  変数(値 1,値 2,値 3 thru 値 5)
```

「値 3 thru 値 5」とは、「値 3 から値 5 までを欠損値とする」という意味である。もし 3 つ以上の値を欠損値として指定したい場合は、このように「thru」を用いて工夫する必要がある。

【具体例】

```
missing values q04(50,97 thru 99).
```

具体例は、「q04 について、50 および 97 から 99 を欠損値とせよ。」という意味である。

また、ある値をいったん欠損値に指定したとしても、解除することができる。

【コマンド：missing values】③
missing values 変数().

上記のように、missing values コマンドを用いて、カッコ内に何も入力しなければ、その変数の欠損値は「なし」となる。

【具体例】
missing values q05().

具体例は、「q05 について、欠損値はなしとせよ。」という意味である。

同じ変数に対して欠損値の指定を複数回おこなった場合、最後に実行したコマンドが有効になるので注意しよう。

3.3 データの変容

前項までは、分析する前に必要なコマンドについて触れてきた。それらのコマンドの他に、データの変容、例えば、新しい変数を作成、変数の値の入れ替え、カテゴリの統合などをおこなうコマンドを知っておいたほうがよい。なぜなら、分析が進むにつれて、もともとある変数では分析しにくくなるときがあるからである。

新しい変数を作成するときにその変数の名前をつけるわけだが、その名前は半角 8 文字以内であれば、どのようにつけてもよい。ただ、どの変数を変容して作成したかがわかる名前のほうがよいだろう。本稿では、ある変数の値を変容して作成した場合は頭文字に「n」を、値を逆転させた場合は「r」を便宜的につけて、新変数を名づけている。

では、データ変容のコマンドを紹介しよう。

（1）compute～変数の作成

例えば、本人の年収と配偶者の年収を用いて夫婦合算の年収を求めたいとき、手元にある変数だけでは対処できない。こういうときは、compute コマンドを用いて、新しい変数を作成すればよい。

【コマンド：compute】
compute 新変数=命令文.

命令文では、算術記号（＋、－、＊、／）を用いた計算やカッコを含む計算をすることができる。

【具体例】
compute gassan = q01+q02.

具体例は、「q01 と q02 の和を値とする gassan という変数を作成せよ」という意味である。

【具体例】
compute tuugaku = (q03*60)+q04.

時間（q03）と分（q04）が別々に訊ねられている通学時間の項目を、分に換算（tuugaku）することを例にして説明しよう。通学時間を分に換算するためには、「q03」に 60 を乗じて「q04」を加算する必要がある。具体例は、その計算をおこなっている。

四則演算のほかに、絶対値（< ABS >）や対数（< LG10 >、< LN >）なども用いることができる。

（2）recode～値の再割り当て

recode コマンドは、変数の値を他の値に変更するときに用いる。もともとの値を別の値にして、分析を進めたいときがある。例えば、ある質問が

「そう思う」～「そう思わない」の 5 件法で訊ねられており、その項目が、「そう思う」が「1」～「そう思わない」が「5」のように入力されているとする。この入力された値を反転させて、「そう思う」を「5」～「そう思わない」を「1」としたい場合に、recode コマンドを用いる。

【コマンド：recode】①
recode 変数(変容前の値＝変容後の値).

同じ変数であれば、1 つのコマンドで複数の値を変容することが可能である。その場合は、カッコを並列して変容の指定をしておけばよい。
【具体例】
recode q01 (1=3)(2=2)(3=1).

具体例は、「q01 の値について、1 を 3、2 を 2、3 を 1 とせよ。」という意味である。

また、「そう思う」と「ややそう思う」、「あまりそう思わない」と「そう思わない」をそれぞれ統合し、3 カテゴリのデータをつくる場合もある。このときも、recode コマンドを用いる。

【コマンド：recode】②
recode
変数名(変容前の値 1, 変容前の値 2＝変容後の値).

上のように、変容前の値をカンマ（,）で並べれば、複数の値を 1 つに統合することができる。

【具体例】
recode q02 (1,2=1)(3=2)(4,5=3).

具体例は、「q02 の値について、1 と 2 を 1 に、3 を 2 に､4 と 5 を 1 とせよ。」という意味である。

recode コマンドを実行する際には、注意しなければならないことがある。それは、recode コマンドを実行した変数の値は、それ以降、変容された値のままになってしまう点である。つまり、変容前の値を使いたくなっても、その値には既に新しい情報が上書きされているのである。

この問題を解決する方法は 2 つある。1 つは、recode コマンドを実行する前に、recode コマンドを実行する変数と同じ変数を別に作ってから、実行する方法である。

【具体例】
compute nq02=q02.
recode nq02 (1,2=1)(3=2)(4,5=3).

具体例は、「q02 と同じ nq02 という変数を作成せよ。その nq02 の値について、1 と 2 を 1 に、3 を 2 に、4 と 5 を 3 とせよ。」という意味である。

いま 1 つは、recode コマンドの内容を新変数に対して実行する方法である。

【具体例】
recode q01 (1=3)(2=2)(3=1) into rq01.

具体例は、「q01 の値をもとに、1 が 3、2 が 2、3 が 1 となるような rq01 という変数を作成せよ。」という意味である。

この 2 通りの方法でもって、元のデータの情報を守ることができる。

(3) if～条件文による変数の作成

compute コマンドと同様、この if コマンドでも新しい変数を作り出すことができる。if コマンドが compute コマンドと異なるのは、compute コマンドが計算式によって変数をつくるのに対し、if コマンドは論理式によって変数をつくる点である。例えば、年齢を「35 歳未満」、「35 歳以上 60 歳

未満」、「60 歳以上」という 3 つのカテゴリに分けたい場合、if コマンドを用いることになる。また、カテゴリの統合なども、recode コマンドだけでなく if コマンドを用いても可能である。

【コマンド：if】
if (論理式)新変数=新変数の値.

「論理式」にあたる部分に、新変数の条件を入力し、新変数の名前と値を指定する。論理式には、比較演算（<、>、≦、≧、=、）や論理演算（and、or、not）を用いる。

（比較）演算子は、文字に置き換えることもできる。表 1 は、演算子の記号と文字との対応関係を示したものである。

表 1　演算子による式と文字による式

演算子	文字	
a＞b	a gt b	(greater than)
a＜b	a lt b	(less than)
a≧b	a ge b	(greater equal)
a≦b	a le b	(less equal)
a＝b	a eq b	(equal)
a≠b	a ne b	(not equal)

【具体例】
if (q01 <= 2)nq01=1.
if (q01 = 3) nq01=2.
if (q01 >= 4) nq01=3.

この具体例では、q01 という変数から、nq01 という新変数をつくっている。コマンドは、「q01 が 2 以下ならば nq01 を 1、q01 が 3 ならば nq01 を 2、q01 が 4 以上ならば nq01 を 3 とせよ。」という意味である。

複数の値をもつ新変数をつくる場合には、その都度 if コマンドで指定しなければならない。if コマンドのどの条件にもあてはまらない場合は、その変数のシステム欠損値として処理される。

次に先ほど述べた、年齢を「35 歳未満」「35 歳以上 60 歳未満」「60 歳以上」という 3 つのカテゴリに分ける方法について述べよう。

【具体例】
if (q02 lt 35)sedai3=1.
if (q02 ge 35 and q02 lt 60)sedai3=2.
if (q02 ge 60)sedai3=3.

この具体例では、q02（年齢の項目）から sedai3 という新変数をつくっている。コマンドは、「q02 が 35 未満なら、sedai3 を 1、q02 が 35 以上かつ 60 未満なら sedai3 を 2、q02 が 60 以上なら sedai3 を 3 とせよ」という意味である。

また、論理式内では複数の変数の組み合わせも可能である。例えば、性別と婚姻状況（既婚・未婚）から、4 つの類型を得たい、といった場合である。

【具体例】
if (q03 eq 1 and q04 eq 1)marriage =1.
if (q03 eq 1 and q04 eq 2)marriage =2.
if (q03 eq 2 and q04 eq 1)marriage =3.
if (q03 eq 2 and q04 eq 2)marriage =4.

具体例では、q03（性別の項目）と q04（婚姻状況の項目）から marriage という新変数を作っている。コマンドは、「q03 が 1 で、かつ q04 が 1 ならば marriage を 1、q03 が 1 で、かつ q04 が 2 ならば marriage を 2 とせよ、q03 が 2 で、かつ q04 が 1 ならば marriage を 3 とせよ、q03 が 2 で、かつ q04 が 2 ならば marriage を 4 とせよ。」という意味である。

論理式では、変数の具体的な値から定義するだ

けでなく、変数どうしの関係からも定義できる。

【具体例】
if (q05 eq q06 and q07 eq q08)new=1.
if (q05 ne q06 or q07 ne q08)new=2.

具体例では、q05 と q06 および q07 と q08 から、new という新しい変数をつくっている。コマンドは、「q05 と q06 が等しく、かつ q07 と q08 が等しければ new を 1、q05q と 06 が等しくない、または q07 と q08 が等しくなければ new を 2 とせよ。」という意味である。

（4）rank～ケース等分による変数の作成

rank コマンドでも新変数をつくることができる。量的変数を用いて、ケースを等分するときに rank コマンドが用いられる。例えば、世帯収入をもとに全体を 3 分割し、「上」、「中」、「下」という値をもつ新変数をつくるときなどである。

【コマンド：rank】
rank variable=変数 1
/ntiles(分割数) into 新変数.

「変数 1」には、ケースを等分する変数を入力する。「ntiles」は、分割数を指定するサブコマンドである。（分割数）に入力した数字で「変数 1」を等分することになる。さらに、「ntiles」に< into 新変数>を追記することによって、等分した結果を新変数として保存できる。新変数の値は、こちらで指定する必要はなく、自動的に 1、2、3…という具合に与えられる。このとき、値は対象変数の値が小さいグループから順に与えられる。

【具体例】
rank variable=q01
/ntiles(3) into shotoku3.

具体例は、「q01 を用いてケースを 3 等分し、その結果を shotoku3 という新変数にせよ。」という意味である。

（5）execute～データ定義・変容コマンドの実行

frequencies などの分析コマンドが実行されるまで、データの定義、変容に関するコマンドの効力は保留される。分析コマンド実行前にデータ定義やデータ変容のコマンドの保留状態を解除したい場合、すなわち、データ定義やデータ変容のコマンドだけを実行したい場合、execute コマンドを用いればよい。execute コマンドを実行させる箇所は、データ定義やデータ変容のコマンドの直後である。

【コマンド：execute】
execute.

execute コマンドは、サブコマンドを指定する必要はない。先にも述べたように、データ定義やデータ変容コマンドとあわせて実行すれば、SPSS は分析コマンドの実行を待たずに、データの定義や変容だけをおこなってくれる。

3.4 データの保存・結合

（1）save～データの保存

SPSS に読み込んだデータや SPSS で入力したデータは、SPSS データファイルとして保存することができる。

【コマンド：save】
save outfile = ’データ保存場所¥ファイル名.sav’.

「outfile」は、ファイルの保存先やファイル名を指定するサブコマンドである。どこに、どういった名前で保存するかをoutfileサブコマンドで指定する。SPSSのデータファイルの拡張子は「sav」である。ファイル名の後に<.sav >と入力しておかなければならない。

(2) match files～データの統合

エクセルの1枚のシートには、256列まで入力することができる。列の数は変数の数と対応していることより、変数が256個までの調査ならデータ入力は容易におこなうことができる。しかし、調査によっては変数の数が256個以上になるときがある。そういう場合、2つのエクセルファイルを用いてデータ入力をおこなう必要がある。

get dataコマンドだけでは、データが入力された2つのエクセルファイルを読み込むことはできない。それでは、2つのエクセルファイルを、SPSS上で1つのデータに統合する方法を説明しよう。ここでは2つのデータをそれぞれaデータとbデータと呼ぶことにする。

まず、aデータをSPSSで読み込み、その読み込んだものをSPSSデータファイルとして保存する。次に、aデータと同様に、bデータをSPSSで読み込み、その読み込んだものをSPSSデータファイルとして保存する。最後に、別々に保存したSPSSデータファイルを、match fileコマンドを用いて統合する。

【コマンド：match files】

```
match files
    /file='ファイルの場所 1¥ファイル名 1.sav'
    /file='ファイルの場所 2¥ファイル名 2.sav'
    /by キー変数.
```

match fileコマンドは、SPSSデータファイルとして保存されているデータどうしを統合するものである。「file」は、統合するデータを指定するサブコマンドである。データの統合は、先に指定したファイルのデータに、後で指定したファイルのデータを追加する形でおこなわれる。例は、「『ファイル名1』のデータに『ファイル名2』のデータを追加せよ」という意味である。「by」は、統合する際に用いるキー変数を指定するサブコマンドである。データを統合するとき、統合するファイル間でケースを照合するための変数が必要となる。キー変数とは、このようにケースを照合する際に用いる変数のことである。通常、キー変数にはID（ケース番号）が用いられる。

このmatch fileコマンドは、エクセルで新たに作成した変数を今使用しているSPSSデータファイルに追加したいときにも有効である。

以上が、データの読み込み・定義・変容・保存に必要なコマンドである。

4　分析～基礎編～

4.1 単純集計

細かい分析に入る前に、単純集計を見て、データがどのように分布しているかを確認するのが常套手段である。単純集計を出させるコマンドとして、度数分布を出させるものや平均値などの統計量を出させるものがある。まず、度数分布を出させるコマンドについて説明することにする。

(1) frequencies～度数分布の表示

frequenciesは、度数分布表を得るためのコマンドである。

【コマンド：frequencies】

```
frequencies 変数
    /statistics=min max mean.
```

このコマンドを実行する前に、変数に対して欠損値の指定をしておけば、欠損値を除いた度数分布が出力される。「statistics」は、度数分布表とは別の表で記述統計量を出力させるサブコマンドである。上記の例では、最小値（< min >）、最大値（< max >）、平均値（< mean >）も出力させている。statistics サブコマンドはこれらの他に、中央値（< median >）、分散（< variance >）、標準偏差（< stddev >）も出力可能である。

この frequencies コマンドは、1 つの変数に対してだけではなく、複数の変数に対しても有効である。複数の変数の度数分布表を同時に出力させたい場合は、スペースを空けて変数名を並列しておけばよい。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、< iclude >である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

【具体例】

```
frequencies q01 q02
    /statistics=mean variance stddev.
```

具体例は、「q01 と q02 について度数分布表を出力せよ。同時に平均、分散、標準偏差も出力せよ。」という意味である。

（2）descriptives～記述統計量の表示

frequencies コマンドを、収入などの量的変数に対して実行すると、縦に長い度数分布表が出力される。この縦長の表は、データの分布を見るということでは意味のあるものだが、それ以外に意味はない。量的変数の分布は、平均値や中央値などの代表値で見るのが一般的である。frequencies コマンドは、カテゴリカルな変数に対して実行すれば、変数の値に応じて度数分布表を出力してくれるが、量的変数に対して実行するとあまり意味のない縦長の表を出力してしまう。変数の各値についての情報を出させずに、変数自体の記述統計量を出力させるには、descriptives コマンドを実行すればよい。

【コマンド：descriptives】①

```
descriptives 変数.
```

descriptives コマンドは、1 つの表で、欠損値を除いた度数と、最小値、最大値、平均値、標準偏差とを出力する。複数の変数に対して、このコマンドを実行したい場合は、スペースを空けて変数名を並列しておけばよい。

【具体例】

```
descriptives q01 q02.
```

具体例は、「q01 と q02 について、欠損値を除いた度数と、最小値、最大値、平均値、標準偏差を出力せよ。」という意味である。

最小値、最大値、平均値、標準偏差の他に、分散などの記述統計量を出力させたい場合は、statistics サブコマンドが必要である。

【コマンド：descriptives】②

```
descriptives 変数
    /statistics= min max mean stddev variance.
```

statistics サブコマンドを実行させるときには、注意が必要である。というのは、このサブコマンドをつけると、descriptives コマンドだけで実行していたときに出力されていた記述統計量が出なくなるからである。

【具体例】

```
descriptives q03 q04
    /statistics= min max mean stddev variance.
```

具体例は、「q03 と q04 の欠損値を除いた度数と、最小値、最大値、平均値、標準偏差と分散を出力せよ。」という意味である。

また、descriptives コマンドは、変数の標準得点を作成するときにも用いる。その場合は、save サブコマンドを使う。

【コマンド：descriptives】③

```
descriptives 変数
    /save.
```

このように save サブコマンドつければ、データエディタに標準得点が変数として追加される。このとき、変数名は「Z 変数名」という名前で保存される。

【具体例】

```
descriptives q05 q06
    /save.
```

具体例は、「q05 と q06 について、欠損値を除いた度数と、最小値、最大値、平均値、標準偏差を出力せよ。また、それぞれの標準得点を保存せよ。」という意味である。この場合、「q05」、「q06」の標準得点は、「Zq05」、「Zq06」として保存される。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、< variable >である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

frequencies コマンド、descriptives コマンド、どちらが使いやすいかは、それぞれ利点があるので判断できない。どちらを用いるかは、読者の判断に委ねたい。

（3）mult response～多重回答の度数分布

質問紙の中に、当てはまるもの全てにマルをつけるような質問がある。例えば、「あなたが働く理由として当てはまるものすべてにマルをしてください。」といった質問である。このような質問は、多重回答（multiple answer）の質問、またはマルチの質問などと呼ばれている。

多重回答の質問のデータ入力は、各項目、例でいうところの「働く理由」それぞれに対して、マルがあれば「1」、なければ「0」とする方法が一般的である。つまり、「0」と「１」の値を持った変数が、項目数個できることになる。

この項目数個ある変数をそれぞれ frequencies コマンドで単純集計しても、明らかになるのは各項目の単純集計である。各項目でなく、項目全て、すなわち質問全体の単純集計を見るには、mult response コマンドを実行すればよい。

mult response コマンドを用いれば、1 つのコマンドで複数の度数分布表やクロス表を作成する作業が指示できる。

【コマンド：mult response】①

```
mult response groups=グループ名 (変数 1 変数 2 変数 3 変数 4 (カウントする値))
    /frequencies=グループ名.
```

groups サブコマンドにおいて、多重回答の各項目を 1 つの多重回答グループとして定義し、さらに、「カウントする値」を指定しなければならない。「カウントする値」とは、文字通り数える値のことで、マルがあれば「1」、なければ「0」方式で入力されているデータのマルの数を数える場合、「(1)」と入力することになる。つまり、mult response コマンドを実行すれば、多重回答の質問

の各項目で「カウントする値」、すなわちマルがいくつあるかわかるようになるわけである。

そして、frequencies サブコマンドで、グループ名を指定し、度数分布表を出力させる。出力された表には、グループを構成する各項目について、「カウントする値」の数と、「カウントする値」の各項目の合計に占める各項目の「カウントする値」の数の割合と、各項目の計に占める「カウントする値」の数の割合とが示される。

文字だけだとわかりづらいので数式を用いて表現しよう。変数 1 について「カウントする値」（マルにしている人）の数を a、マルにしていない人を A とし、この関係と同様に変数 2、変数 3、変数 4 についても、それぞれ b と B、c と C、d と D とする。また、「カウントする値」の合計 a+b+c+d を X とする。表 2 は、先ほどの文字で表した関係を示したものである。

表 2　mult response コマンドで表示されるもの

	①	②	③
q01	a	a/X*100	a/(a+A)*100
q02	b	b/X*100	b/(b+B)*100
q03	c	c/X*100	c/(c+C)*100
q04	d	d/X*100	d/(d+D)*100
合計	X	100.0	

表 2 の①が「カウントする値」の数、②が「カウントする値」の各項目の合計に占める各項目の「カウントする値」の数の割合、③が各項目の計に占める「カウントする値」の数の割合に当たる。

【具体例】

```
mult response groups = reasons (q01x1 q01x2 q01x3
                                q01x4 (1) )
  /frequencies= reasons.
```

具体例は、「q01x1 から q01x4 までの 4 変数で reasons という多重回答のグループを構成し、q01x1 から q01x4 の『1』の数をカウントし、reasons の度数分布表を示せ。」という意味である。

また、mult response コマンドを用いて、多重回答のグループと他の変数とのクロス表をつくることもできる。

【コマンド：mult response】②

```
mult response groups=グループ名 (変数 1 変数 2
変数 3 変数 4 (カウントする値))
  /variables=クロスさせる変数(最大値，最小値)
  /tables=クロスさせる変数 by グループ名
  /cells=count column.
```

「variables」は、多重回答のグループとクロスさせる変数を指定するサブコマンドである。変数を指定すると同時に、その変数が取る値をカッコ内に入力する必要がある。「tables」は、表側と表頭に置く変数を指定するサブコマンドである。by の前に指定したものが表側に、by の後ろに指定したものが表頭に来る。「cells」は、表中のセル内にどのような値を表示させるかを指定するサブコマンドである。< count >と< column >と入力することによって、ケース数と列パーセントがクロス表に示されるようになる。

以上のサブコマンドを指定すれば、多重回答のグループと他の変数とのクロス表が得られる。しかし、cells サブコマンドで< column >を指定した場合は各列パーセントの計が、< row >を指定した場合はほとんどのパーセントが、正確に算出されないので、もしこの表を用いてレポートなり論文を書くときは、適宜エクセルなどで再計算する必要がある。注意してほしい。

【具体例】

```
mult response groups = reasons (q01x1 q01x2 q01x3
                                q01x4 (1) )
  /variables=q02(1,2)
  /tables=q02 by reasons
  /cells=count column.
```

具体例は、「q01x1 から q01x4 までの 4 変数で reasons という多重回答のグループを構成し、reasons と 1 から 2 の値を取る q02 とのクロス表を出力せよ。クロス表の表側は q02、表頭は reasons にし、表の中身にはケース数と列パーセントを表示せよ。」という意味である。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、＜table＞である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

SPSS のヴァージョンが古い場合、後で説明する partial correlations コマンドと同じように、mult response コマンドの出力結果がテキスト形式で表示されることがある。使用している SPSS のヴァージョン次第では、表を貼り付ける際に注意が必要である。

（4）means～平均値の比較

means コマンドは descriptives コマンドとよく似ている。似ているのは、記述統計量を出力する点である。異なるのは、means コマンドが平均値に特化している点と、カテゴリカル変数を加えることによってカテゴリ間の平均値の比較ができる点である。

【コマンド：means】①

```
means 変数.
```

means コマンドを実行すれば、度数、平均値、標準偏差が出力される。複数の変数に対して means コマンドを実行したい場合は、スペースを空けて変数名を並列すればよい。

【具体例】

```
means q01 q02.
```

具体例は、「q01 と q02 の度数、平均値、標準偏差を出力せよ。」という意味である。

先ほど述べた平均値の比較をおこなうときの means コマンドは、次の通りである。

【コマンド：means】②

```
means 変数 1 by 変数 2.
```

「means 変数 1」の後に「by 変数 2」と入力すれば、「変数 2」のカテゴリごとに「変数 1」平均値を算出してくれる。カテゴリという言葉を用いていることからわかるように、「変数 2」はカテゴリカル変数である必要がある。

【具体例】

```
means q03 by q04.
```

具体例は、「q04 のカテゴリごとに q03 の平均値を算出せよ。」という意味である。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、＜table＞である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

本項に挙げたコマンドは、基本的には度数分布や基礎統計量を見るために必要なものであった。1 つ 1 つの変数の基本的な情報を確認した後にすることは、変数間の関連を見ることである。以下では、変数間の関連を見るコマンドについて説明することにしよう。

4.2 2 変数間の関係と検定 1～平均の差の検定

means コマンドで平均値の比較はできるが、検定はおこなえない。本項で説明するのは、2 グループの平均値の差について検定をおこなうコマンドである。

（1）t-test～平均の差の検定

t-test コマンドについて説明しよう。

【コマンド：t-test】

```
t-test groups=変数 1(値 1,値 2)
    /variables=変数 2.
```

「groups」は、比較をおこなう 2 グループの元となる変数を指定するサブコマンドである。変数を指定し、その変数の中のどの値どうしを比較するのかを、カッコに入力する。性別のように 1 つの変数で 2 つの値しか取らないものは、カッコ内に値の指定をしなくても問題はないが、1 つの変数で 3 つ以上の値を取る場合はどのグループ間で比較するかを値で指定しなければならない。「variables」は、平均値を算出する変数を指定するサブコマンドである。他の変数の平均値も同時に算出するように命令するには、スペースを空けて変数を並列すればよい。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、＜analysis＞である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

【具体例】

```
t-test groups=q01(1,2)
    /variables=q02 q03 q04.
```

具体例は、「q02、q03、q04 の平均値を、q01 の 1 と 2 と別に算出し、その平均値の差の検定をおこなえ。」という意味である。

検定結果を見るには注意が必要である。検定結果は、「独立サンプルの検定」という表で出力される。「独立サンプルの検定」の表は、「等分散性のための Levene の検定」の部分と、「2 つの母平均の差の検定」の部分からなる。

「等分散性のための Levene の検定」では、「2 つのグループの母分散が等しい」という仮説の検定をおこなっている。この部分の仮説が棄却できる場合、すなわち母分散が等しいと仮定できない場合は、「等分散を仮定しない」の行、仮説が棄却できない場合、すなわち等分散が仮定できる場合は、「等分散を仮定する」の行の t 値および有意確率を見ればよい。見るべき t 値および有意確率は、「2 つの母平均の差の検定」に示されている。

（2）oneway～一元配置の分散分析

先ほど説明した t-test コマンドは、2 つのグループ間で平均の差を比較することに適しているが、3 つ以上のグループ間で平均の差を比較するには向いていない。3 つ以上のグループ間で平均の差を比較するときは、oneway コマンドを用いて一元配置の分散分析をおこなうことになる。言うまでもないが、oneway コマンドは、2 つのグループ間の平均の差の検定もおこなうことができる。

【コマンド：oneway】①

```
oneway 変数 1 by 変数 2.
```

「変数 1」には平均値を算出する変数、「変数 2」には比較をおこなう変数を入力する。このコマンドを実行すると分散分析の結果が出力される。分散分析の結果と同時に、記述統計量も出力させたい場合は、サブコマンドを入力する必要がある。

【コマンド：oneway】②
oneway 変数 1 by 変数 2
　/statistics=descriptives.

「statistics = descriptives」入力すれば、カテゴリごとに、度数、平均値、標準偏差、標準誤差、平均値の 95%信頼区間、最小値、最大値が出力される。他の変数の平均値も同時に算出するように命令するには、スペースを空けて変数を並列すればよい。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、＜analysis＞である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

【具体例】
oneway q01 by q02
　/statistics=descriptives.

具体例は、「q02 のカテゴリごとに q01 の平均値を算出し、その平均値の差の検定をおこなえ。また、度数、平均値、標準偏差、標準誤差、平均値の 95%信頼区間、最小値、最大値も出力せよ。」という意味である。

4.3 2 変数間の関係と検定 2～相関係数、クロス表

平均値の差を見る以外にも変数間の関係を見る方法はある。本項では、2 つの変数間の関係を見るコマンドとして、crosstabs コマンドと correleations コマンドを説明する。

まずは、crosstabs コマンドから説明しよう。

（1）crosstabs～クロス集計の作成

crosstabs コマンドを実行すれば、クロス集計表が出力される。サブコマンドをつければ、カイ 2 乗検定をおこなうこともできる。

【コマンド：crosstabs】①
crosstabs tables=変数 1 by 変数 2
　/cells=count row
　/statistics=chisq.

「tables」は、表側と表頭に置く変数を指定するサブコマンドである。by の前に指定したものが表側に、by の後ろに指定したものが表頭に来る。「cells」は、表中のセル内にどのような値を表示させるかを指定するサブコマンドである。例で用いている＜count＞は度数を、＜row＞は行パーセントを表示せよ、という意味である。列パーセントを表示したい場合は、＜column＞と入力する。

statistics サブコマンドは、指定した統計量を算出し検定をせよ、という指示を与えるものである。例にある＜ chisq ＞は、カイ 2 乗値を求めてカイ 2 乗検定をせよという意味である。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、＜table＞である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

クロス表を作成するときに気をつけなければならないのは、独立変数のカテゴリが 100 パーセントとなるようにしなければいけない点である。このことに関わるのは、tables サブコマンドで指定する変数の位置と、cells サブコマンドで指定するセル内に表示する値とである。独立変数を表頭にした場合は列パーセント（tables=従属変数 by 独立変数 /cells= column）が、独立変数を表側にした場合は行パーセント（tables=独立変数 by 従属変数 /cells= row）が出力されるようにする必要がある。

crosstabs コマンドを用いれば、2 変数からなる 2 重クロス表だけでなく、3 変数以上からなる多重クロス表をつくることも可能である。

【コマンド：crosstabs】②

```
crosstabs tables=変数 1 by 変数 2 by 変数 3
    /cells=count row
    /statistics=chisq.
```

例のように、表頭に来る変数（変数 2）の後にさらに「by 変数 3」とすれば、変数 3 でコントロールされた変数 1 と変数 2 のクロス表が出力される。「by 変数」でつなげることによって、コントロール変数はいくらでも増やすことができる。ただし、いうまでもないがクロス集計に用いる変数はカテゴリカル変数である必要がある。

（2）correleations～相関行列の作成

次は、correlations について。correleations コマンドを実行すれば、相関行列が出力される。

【コマンド：correlations】

```
correlations 変数 1 変数 2.
```

分析結果は、1 つの表で出力される。表のセル内には、相関係数、有意確率、度数が表示される。複数の変数の相関を見る場合は、スペースを空けて変数を並列すればよい。

なお、missing サブコマンド（欠損ケースの処理）のデフォルトは、＜pairwise＞である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

【具体例】

```
correlations q01 q02 q03.
```

具体例は、「q01 と q02、q01 と q03、q02 と q03 の度数と相関係数を出力せよ。また、その相関係数の検定をおこなえ。」という意味である。

2 変数の相関係数から第 3 変数の影響を除外した、偏相関係数を求めたいときは、partial correlations コマンドを用いる。

【コマンド：partial correlations】

```
partial correlations 変数 1 変数 2 by 変数 3.
```

by の後に影響を除外したい変数を入力する。partial correlations コマンドの分析結果は、correlations コマンドと同様に、1 つの表で出力される。表のセル内には、偏相関係数、有意確率、自由度（df）が表示される。SPSS のヴァージョンが古い場合、分析結果が表の形でなくテキスト形式で表示されることがある。分析結果に変わりはないが、エクセルへの表の貼り付け方が少々複雑になるので注意が必要である。

【具体例】

```
partial correlations q01 q02 q03 by q04.
```

具体例は、「q01 と q02、q01 と q03、q02 と q03 の偏相関係数と自由度を出力せよ。また、その偏相関係数の検定をおこなえ。」という意味である。

ただし、こちらもいうまでもないが、変数間の相関係数を見るときに用いる変数は、量的変数でなければならない。

4.4 その他のコマンド

本項で説明するコマンドは、分析には直接的な関係はない。しかし、分析を進めるにあって、知っておいたほうがよいコマンドではある。

（1）select if～特定のケースの選択

データのうちあるケース（例えば男性）に限定して分析をおこなうことがある。このときは、select if コマンドを用いる。

【コマンド：select if】
select if 論理式.
分析コマンド.

select if コマンドによってケースを選択した場合、それ以降の分析はすべて選択されたケースのみが対象となり、選択されなかったケースは分析から除外される。このため、select if コマンドを用いる際には注意が必要である。次に説明する temporary コマンドを用いれば、この問題は解消される。

【具体例】
select if q01=1.
correlations q02 q03.

具体例は、「q01 を 1 と回答しているケースだけを有効にせよ。そして、q01 の有効ケースを対象に q02 と q03 の度数と相関係数を出力せよ。また、その相関係数の検定をおこなえ。」という意味である。

(2) temporary～一時的なデータの変容

先ほど述べたように、select if コマンドによってケースを選択した場合、選択されなかったケースは以後の分析すべてにおいて除外されてしまう。select if コマンドで選択されなかったケースが除外されたままの変数では、あとでおこなう分析に支障をきたす。この問題を解決する方法は 2 つある。1 つは、compute コマンドや recode コマンドで同じ値を取る変数を作成し、その変数に対して select if コマンドを実行する方法である。いま 1 つは、select if などのデータ変容コマンドの効力を一時的なものにする方法である。後者の作業をおこなうコマンドが、temporary コマンドである。

【コマンド：temporary】
temporary.
データ変容コマンド.

データ変容コマンドの前に、temporary コマンドを実行することによって、それ以後のデータ変容コマンドは一時的なものとなる。temporary コマンドの効力は、次の分析コマンド（crosstabs や correlations など）を実行するまでである。分析コマンドが実行された時点で、一時的なデータ変容コマンドは解除される。tempoary コマンドが機能しうるデータ変容コマンドは、recode、compute、if、select if などである。

【具体例】
temporary.
select if q01=1.
correlations q01 q02.

具体例は、「q01 を 1 と回答しているケースだけを有効にせよ（ただし、このデータの変容は、次の分析後には無効になる）。そして、q01 の有効ケースを対象に q02 と q03 の度数と相関係数を出力せよ。また、その相関係数の検定をおこなえ。」という意味である。

(3) missing サブコマンド～欠損値をもつケースの扱いに関する指定

3.2 で紹介した missing values コマンドでは、特定の値を欠損値として定義することができ、定義された欠損値は、分析の対象から除外される。つまり、分析に用いようとしている変数に欠損値をもつケース（サンプル）は、その分析から除外されることになる。

ただし、除外の仕方にもいくつか方法があり、分析をおこなう際には missing サブコマンドを用

いて具体的に指定することになる。ここでは、下の表のようなデータを用いて相関係数を計算させる場合を例にしてその方法を紹介しよう（表 3：値は「1」～「5」までで、「9」は欠損値とする）。

表 3　架空のデータ（欠損値は 9）

	var_A	var_B	var_C
ケース1	1	3	2
ケース2	3	5	3
ケース3	4	2	1
ケース4	5	9	1
ケース5	3	4	9
ケース6	2	4	9

①listwise

missing サブコマンドで< listwise >を指定すると、その分析に用いることになっている（var サブコマンド以下に挙げられている）変数に 1 つでも欠損値をもつケースは、分析から除外される。つまり、< listwise >と指定した場合、分析に用いる変数に欠損値をもたないケースのみで分析がおこなわれるのである。

表 3 のデータにおいて変数 A～変数 C についてそれぞれの相関係数を出力するとき、欠損値処理を< listwise >とするシンタックスは次のようになる。

【具体例】

```
correlations var=var_A var_B var_C
 /missing=listwise.
```

4.3 の(2)で述べたとおり、上のような correlation コマンドを実行すると、var_A と var_B、var_A と var_C、var_B と var_C についての相関係数と度数、検定の結果が出力される。このときの分析結果としては、欠損値をもたないケース 1～ケース 3 のみで計算した値が出力される。つまり、var_B に欠損値をもつケース 4 と、var_B に欠損値をもつケース 5､6 が分析から除外されることになるのである。

②variable、pairwise

< listwise >を指定すると、分析に用いる変数に 1 つでも欠損値をもつケースは分析から除外される。このため、多くの変数を扱うような分析では除外されるケース数がかなり多くなってしまう、といった事態もしばしば起こる。しかし、調査対象者が少ない場合などは分析から除外されるサンプルを最小限にとどめておく必要も出てくる。こういった場合のために、< variable >や< pairwise >という指定の方法もある。

< variable >は分析に用いることになっている変数ごとにそれぞれ欠損値を処理する方法であり、descriptives コマンドのように 1 つの変数のみを扱う分析の際に用いられる。

< pairwise >は correlation コマンドなどのように 2 つの変数を同時に扱うような分析の際に用いられるもので、統計量を算出する変数のペアごとに欠損値を処理する方法である。つまり、欠損サンプルは 1 つの統計量を算出する際に関わる変数に欠損値をもつケースのみが除外されることになるのである。

表 3 のデータにおいて var_A、var_B、var_C についてそれぞれの相関係数を出すとき、欠損値処理を< pairwise >とするシンタックスは次のようになる。

【具体例】

```
correlations var=var_A var_B var_C
 /missing=pairwise.
```

具体例のようなコマンドを実行すると、除外さ

れるケースはvar_Aとvar_B、var_Aとvar_C、var_Bとvar_Cの相関係数を算出する場合それぞれで異なってくる。具体的には、var_Aとvar_Bの場合はケース4が、変数Aと変数Cの場合はケース5とケース6が、var_Bとvar_Cの場合はケース4とケース5とケース6が、それぞれ分析から除外される。したがって、結果の度数の部分にはそれぞれ「5」、「4」、「3」という値が出力されることになるのである。

③include

missingサブコマンドで< include >を指定すると、分析に用いる変数に欠損値をもつケースも、分析に含めることができる。

表3のデータにおいてvar_A、var_B、var_Cについてそれぞれの相関係数を出すとき、欠損値処理を< include >とするシンタックスは次のようになる。

【具体例】

```
correlations var=var_A var_B var_C
 /missing=include.
```

具体例のようなコマンドを実行すると、すべてのケースを分析の対象とした結果が出力される。結果の度数の部分にはすべて「6」という値が出力されることになるのである。

基本的な指定の方法は以上であるが、他にもいくつかあるので紹介しておこう。

④table

コマンドによっては、サブコマンドの指定によって扱う変数の数が1変数になったり2変数になったりするものがある（means コマンドや multi response コマンドなど）。このようなコマンドを実行する際に< variable >や< pairwise >と同様の処理をおこなうためには、< table >と指定すればよい。やや大雑把にいえば、< table >と指定すると、扱う変数が1変数の場合は< variable >、2変数以上を同時に扱うような場合は< pairwise >と同様の処理をおこなうことになるのである。

⑤analysis

また、コマンドによっては1つのコマンドで複数の分析を同時におこなうものもある（T-test コマンドなど）。このようなコマンドを実行する際に< pairwise >と同様の処理をおこなうためには、< analysis >と指定すればよい。こちらもやや大雑把にいえば、< analysis >と指定しておくと、複数ある分析ごとに欠損値を< pairwise >として処理することになるのである。

⑥exclude

< exclude >はある意味もっとも単純な指定方法である。< exclude >を指定すると、分析に扱う変数として挙げられていようがいまいが、とにかく欠損値を含むケースは分析から除外されることになる。

欠損値をもつケースの処理には、以上のような方法がある。分析結果の厳密性という観点からは、一括に欠損値を除外する< listwise >や< exclude >がもっとも安心だといえる。しかし、先に述べたようにそもそもデータのケースに制限がある場合などは、可能であれば< variable >や< pairwise >、< tables >、< analysis >などを指定しておくとよいだろう。

ちなみに、missingサブコマンドは特に指定をしなければデフォルトで指定されている方法で欠損ケースが処理されることになる。分析ごとのデフォルトはそれぞれの部分で記述しているので、そ

ちらを参照していただきたい。

5　分析～発展編：いくつかの多変量解析～

ここまで、基礎的な分析などについてのコマンドを説明してきた。レポートや論文に必要な分析が基礎的な分析のみで完了するのであれば、それは結構なことである。デュルケムの『自殺論』は、統計の分析といってもクロス集計だけで書かれている。つまり、素朴な分析だけでも、よいレポートやよい論文は書けるのである。

しかし、我々はなかなかデュルケムのように書けないし、また我々が明らかにしたい社会現象を基礎的な分析だけで解明することは難しい。場合によっては高度な分析も必要となる。基礎的な分析より高度なものとして、多変量解析がある。ここでは、いくつかの多変量解析のコマンドについて説明しよう。

5.1 分散分析

（1）どのようなときに用いるか

平均値の比較を行なうものとして、先ほど T 検定と一元配置の分散分析を挙げた。T 検定と一元配置の分散分析は、どちらも 1 変数についてその値（カテゴリ）ごとの平均の差を見るために用いられる。それらはたとえば、性別という変数の男女というカテゴリそれぞれについて、何らかの従属変数の値の平均が異なるかどうかを調べるものだ。独立変数が複数ある場合の平均値の比較をおこなう場合には、（多元配置の）分散分析をおこなう必要がある。例えば、「学歴」（値：初等、中等、高等）と、「婚姻状態」（値：未婚、既婚）の 2 つの独立変数によって、「生活満足度」に違いがあるかどうかを確かめるときに、分散分析をおこなうことになる。

（2）anova

【コマンド：anova】①

```
anova variables=従属変数 by 独立変数 1(最小値, 最大値) 独立変数 2(最小値, 最大値)
    /method=experimental.
```

「variable」は、分析に用いる変数を指定するサブコマンドである。指定の方法は、by の前に従属変数を先におき、by 以下に独立変数をおくというものである。2 つめ以降の独立変数を入力するときは、スペースを 1 つあけて入力すればよい。anova コマンドでは、各独立変数がいくつカテゴリをもっているのか、ということも指定しておかなければならない。各独立変数のうしろにカッコをつけ、そのカッコ内に各独立変数の最小値と最大値を入力すれば、カテゴリ数の指定ができる。

分散分析をおこなうと、独立変数の主効果と交互作用の値が示された分散分析表が出力される。主効果と交互作用の値は、他の変数の効果を除去する方法次第で変わる。この他の変数の効果を除去する方法を指定するのが、method サブコマンドである。SPSS では、一意的方法（< unique >）、実験的方法（< experimental >）、階層的方法（< hierarchical >）という 3 通りの除去の方法がある。anova コマンドは、これらのうち 1 つの方法を指定しなければならない。本稿では、これらの中でももっとも単純な実験的方法について説明する。

method サブコマンドを< exiperimental >と指定すれば、実験的方法を用いた結果を出力してくれる。< experimental >を指定した場合、独立変数の主効果の値は、他の独立変数の影響を除去したものが表示される。他方、交互作用の値は、それ以外の交互作用の効果と各独立変数の効果とを除去したものが表示される。

ちなみに、一意的方法は、他の変数すべての効果を除去した各変数の効果のみを取り出すもので、

階層的方法は、段階的に独立変数の効果をみていくものである。

anova コマンドは、variables サブコマンド、method サブコマンドを指定すれば、分散分析の結果を出力してくれる。この分散分析の結果以外にも、必要な統計量を出力させることができる。

【コマンド：anova】②

```
anova　variables=従属変数 by 独立変数 1(最大値, 最小値) 独立変数 2(最大値, 最小値)
    /method=experimental
    /statistics=mean.
```

「statistics」は、分散分析の結果のほかに統計量を計算させるサブコマンドである。平均値も一緒に計算させたい場合、上の例のように< mean >と指定しておけば、カテゴリ別の平均値を出力してくれる。

【具体例】

```
anova　variables=q01 by q02(1,3) q03(1,4)
    /method=experimental
    /statistics=mean.
```

具体例は、「独立変数を q02、q03、従属変数を q01 とする分散分析をせよ。なお、q02 は 1～3、q03 は 1～4 までの範囲である。他の変数の効果を除去する方法は実験的方法を用いよ。また、平均値の表も同時に出力せよ。」という意味である。

ちなみに、missing サブコマンドのデフォルトは、< exclude >である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

5.2 重回帰分析

（1）どのようなときに用いるか

変数間に因果的な関連が想定できる場合がある。このとき、因果的な関連が見られる両変数が量的変数であれば、相関係数を求めるだけでなく回帰分析もおこなうことができる。回帰分析とは、独立変数から従属変数を予測するというものである。1 つの独立変数から従属変数を説明するときに用いられるのが単回帰分析で、複数の独立変数から従属変数を説明しようというときに用いられるのが、重回帰分析である。例えば、「収入」と「睡眠時間」という独立変数から、「幸福度」という従属変数を説明しようとするときに、重回帰分析をおこなう。本項では、重回帰分析をおこなうコマンドについて説明しよう。

（2）regressions

【コマンド：regressions】

```
regressions variables=変数
    /dependent=従属変数
    /method=enter.
```

「variables」は、分析に用いる変数を指定するサブコマンドである。ここには、分析に用いるすべての変数、すなわち独立変数も従属変数も入力しておく。「dependent」は、従属変数を指定するサブコマンドである。variables サブコマンドであげた変数のうち、ここで指定されたものが従属変数として処理される。「method」は、変数の投入方法を指定するサブコマンドである。method サブコマンドを< enter >と指定することで、強制投入法による重回帰分析がおこなわれる。強制投入法のほかにも、投入する変数が自動的に操作されるステップワイズ法や変数除去法という方法もあるが、本稿ではもっとも単純な強制投入法のみを説明する。

【具体例】

```
regressions variables=q01 q02 q03
    /dependent=q01
    /method=enter.
```

具体例は、「q02、q03 を独立変数、q01 を従属変数とする重回帰分析をおこなえ。なお、変数の投入方法は、強制投入法を用いよ。」という意味である。

ちなみに、missing サブコマンドのデフォルトは、< listwise >となっている（missing サブコマンドに関しては、4.4（3）を参照のこと）。

5.3 主成分分析

（1）どのようなときに用いるか

たくさんある量的変数を同時に分析することは、手間であるし、視覚的にもみにくい。いくつかの量的変数をより少ない変数に要約することができれば、これらの問題は解消できる。例えば、「次の事柄は出世にどの程度必要か」という質問において、「才能」、「学歴」「知識」、「家柄」、「努力」、「幸運」、「処世術」という 7 つの項目それぞれが 5 件法で訊ねられているとする。これらの 7 項目をより少ない（合成）変数でとらえたいときに主成分分析をおこなう。

（2）factor

【コマンド：factor】①

```
factor
    /variables 変数
    /criteria factor(2)
    /rotation norotate.
```

「variables」は、分析に用いる変数を指定するサブコマンドである。主成分分析にかける変数は、すべてここに入力する。「criteria」は、主成分抽出するときの基準を指定するサブコマンドである。criteria サブコマンドに factor(主成分の数)としておけば、指定した個数の主成分を抽出してくれる。すなわち、< factor(2) >とすれば、「2 つの主成分を抽出せよ」という意味となる。「rotation」は、主成分を抽出するときに、その主成分を回転させるかどうかを指定するサブコマンドである。rotation サブコマンドを< norotate >とすれば、回転をせずに主成分を抽出することになる。

criteria サブコマンドは、抽出する主成分の数を基準の他に、抽出する主成分の固有値の下限を基準に指定することもできる。

【コマンド：factor】②

```
factor
    /variables 変数
    /criteria mineigen(0.8)
    /rotation norotate.
```

< mineigen >は抽出する主成分の固有値の下限を指定することを意味している。< mineigen(0.8)>は、「固有値 0.8 以上の主成分を抽出せよ」という意味である。

主成分が抽出されれば、主成分得点を求めることができる。save サブコマンドを用いれば、この主成分得点を新たな変数として保存できる。

【コマンド：factor】③

```
factor
    /variables 変数
    /criteria factor(2)
    /rotation norotate
    /save reg(all 新変数).
```

このように、save サブコマンドを< reg(all 新変数)>と指定しておけば、データエディタに主成分

得点が変数として保存される。その変数の名前は、「新変数」の部分に入力した文字に対応する。たとえば、</save　reg (all hk_)>とした場合、主成分得点の変数は「hk_1」「hk_2」として保存される。

ここでの説明では、criteria サブコマンドで 2 つの主成分が抽出されるように指定しているので、保存される変数の数は 2 つになっている。3 つ以上の主成分が抽出される場合は、主成分の数に応じて変数に記される数値も増減する。

さて、上記の例に従って主成分分析をおこない、その結果出力された主成分負荷量が示された表を眺めていても、抽出された主成分が何を意味しているのかいまいちわからないときがある。こういうとき、主成分の回転をおこなうことによって、主成分の解釈を容易にすることができる。その方法は、以下の通りである。

【コマンド：factor】④

```
factor
    /variables 変数
    /criteria factor(2)
    /rotation varimax.
```

rotation サブコマンドで< varimax >と指定すれば、バリマックス回転をおこなった結果が出力される。

コマンド実行後に出力される主成分負荷量を示す表において、変数を負荷量が高い順に並べて結果を見やすくすることもできる。

【コマンド：factor】⑤

```
factor
    /variables 変数
    /criteria factor(2)
    /rotation norotate
    /format sort.
```

format サブコマンドで< sort >と指定すれば、負荷量の高い順に変数を並べ替えた表が出力される。

【具体例】

```
factor
    /variables q01x01 q01x02 q01x03 q01x04 q01x05
               q01x06 q01x07
    / criteria mineigen(1)
    /rotation varimax.
```

具体例は、「q01x01、q01x02、q01x03、q01x04、q01x05、q01x06、q01x07 について、主成分分析をおこなえ。固有値 1 以上の主成分までを抽出し、主成分抽出時にはバリマックス回転をおこなえ。」という意味である。

ちなみに、missing サブコマンドのデフォルトは、< listwise >である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

以上が主成分分析をおこなう際に必要となる基本的なコマンドである。なお、主成分分析をおこなう際、注意しなければならない点がいくつかある。以下にあげておこう。

・相関係数が 1 の変数が含まれている
・変数数がケース数よりも多い
・変数間に論理的な関連がある
・欠損値を pairwise で処理している

上記のような点を考慮せずに factor コマンドを実行したとしても、致命的なエラーがなければ SPSS は主成分分析を最後までおこなってくれる。分析はしてくれるが、その分析は無理やりに実行されたものであるので、分析結果が誤っている可能性がある。したがって、主成分分析をおこなう場合には用いる変数に対して先のような点を注意しておく必要がある。

しておく必要がある。

(3) reliability

尺度を構成する場合、用いる項目の内的一貫性の高さを吟味しておく必要がある。このときにおこなうのが、信頼性分析である。尺度の信頼性はいくつかの側面から検討されるが、本稿ではクロンバッハのアルファ係数の算出方法を紹介する。

【コマンド：reliability】①

```
reliability
    /variables=変数
    /scale(alpha)=all
    /model=alpha.
```

「scale(alpha)」、「model」は、信頼性を検討する際の指標を指定するサブコマンドである。それぞれ、＜all＞、＜alpha＞とすれば、クロンバッハのアルファ係数が出力される。

尺度構成をおこなう場合、使用する項目間の相関も見なければならない。correlations コマンドを用いて出力してもよいが、reliability コマンドを用いても出力することができる。

【コマンド：reliability】②

```
reliability
    /variables=変数
    /scale(alpha)=all
    /model=alpha
    /statistics=corr.
```

「statistics」は、信頼性係数のほかに出力する統計量を指定するサブコマンドである。statistcs サブコマンドを＜ corr ＞と指定すれば項目間の相関係数が出力される。

【具体例】

```
reliability
    /variables=q01x01 q01x02 q01x03 q01x04 q01x05
            q01x06 q01x07
    /scale(alpha)=all
    /model=alpha
    /statistics=corr.
```

具体例は、「q01x01、q01x02、q01x03、q01x04、q01x05、q01x06、q01x07 について、信頼性分析をせよ。なお、信頼性係数としてクロンバッハのアルファ係数を算出せよ。また、項目間の相関行列も同時に出力せよ。」という意味である。

ちなみに、missing サブコマンドのデフォルトは、＜exclude＞となっている（missing サブコマンドに関しては、4.4（3）を参照のこと）。

(4) G-P 分析

尺度を構成する場合、項目分析の1つとしてG-P（Good-Poor）分析をおこなわなければならない。

G-P 分析では、次のような作業をおこなう。まず、ケースを総得点の順に並べ、得点の上位 25%と下位 25%との 2 つのグループに分ける。次に、各グループで得点の平均を算出し、平均の差の検定をおこなう。以上の工程を、尺度を構成している各項目でおこなう。なお、項目の反応が「賛成」「反対」といった 2 件法だった場合は、各グループでの比率をカイ 2 乗検定すればよい。

検定で有意差が得られなかった場合、他の項目との等質性が疑わしいと判断することになる。その項目は、尺度構成から外さなければならない。

SPSS 上での作業としては、rank コマンドを用いて当該項目を上位 25%と下位 25%の 2 グループに分け、その 2 グループ間で T 検定（t-test コマンド）をおこなう、ということになる。

5.4 クラスタ分析

（1）どのようなときに用いるか

類似性の指標を用いて対象をグルーピングし、対象がもつ構造を明らかにしたいときにクラスタ分析を用いる。クラスタ分析の方法は、主に階層的な方法と非階層的な方法2つに分けられる。本稿では、階層的方法によるクラスタ分析を説明する。

階層的方法によるクラスタ分析では、対象の類似性が高いものから順にクラスタ（集落）が形成されていく。クラスタを形成する対象は、ケースの場合もあるし、変数の場合もある。対象の類似性を表す指標として、ユークリッド距離や、ピアソンの積率相関係数、順位相関係数などが用いられる。距離を指標とする場合は対象間の距離が短いものから、相関を指標とする場合は対象間の相関が高いものからクラスタリングされる。

SPSSでクラスタ分析をおこなう際には、2つのコマンドを順に実行することになる。以下で順に紹介しよう。

（2）proximities

まず、何を対象とし、どういった指標を用いるのかを指定した上で、対象間の距離を算出する作業をしなければならない。これにあたるのが、proximitiesコマンドである。

【コマンド：proximities】

```
proximities 変数名
    /view=case
    /measure=euclid
    /matrix=out(*).
```

「view」は、距離を算出するべき対象を指定するサブコマンドである。先に述べたように、クラスタ分析の対象とされるものはケースの場合だけでなく変数の場合もある。したがって、viewサブコマンドにおいて何を対象とするのかを指定しておかなければならない。上の例は対象をケースとする場合であり、ケースではなく変数を対象とする場合は< variable >と指定すればよい。

「measure」は、対象間の距離の算出方法を指定するサブコマンドである。< euclid >とすれば、ユークリッド距離が算出される。この他にも、ピアソンの積率相関係数や順位相関係数など様々選べるので、必要に応じてヘルプを参照し使用していただきたい。

proximities コマンドを実行することで、ケース間もしくは変数間の距離が得られる。「matrix」は、この距離行列データの保存に関するサブコマンドである。分析の結果は「近接行列」という表に出力されると同時に、cluster コマンド実行のためにデータエディタにも保存される。上の例のようにmatrixサブコマンドを< out(*)>と指定すれば、現在使用しているデータエディタに距離行列の情報が保存される。ただし、この距離行列データはデータエディタに上書きされる形になるため元データは一時的に失われることとなる。したがって、proximities コマンドを使用した後のデータの保存には注意しておく必要がある。

【具体例】

```
proximities q01x01 q01x02 q01x03 q01x04 q01x05
    /view=variable
    /measure=euclid
    /matrix=out(*).
```

具体例は、「q01x01、q01x02、q01x03、q01x04、q01x05について、変数間のユークリッド距離を求めよ。また、距離行列はデータエディタに保存せよ。」という意味である。

ちなみに、missingサブコマンドのデフォルトは、

＜exclude＞である（missing サブコマンドに関しては、4.4（3）を参照のこと）。

（3）cluster

proximities コマンドによって対象間の距離は計算され、その情報が保存されている状態になった。次は、計算された距離をもとにクラスタを形成させる作業をおこなう。この作業にあたるのが、cluster コマンドである。cluster コマンドは、データエディタに距離情報が入力されている状態で実行するのが望ましい。したがって、proximities コマンドを先に実行してから、cluster コマンドを実行させるとよい。cluster コマンドのみでもケースを対象としたクラスタ分析はできるが、変数を対象としたクラスタ分析まではできない。本稿では proximities コマンドと cluster コマンドをあわせたクラスタ分析の方法を説明している。

【コマンド：cluster】

```
cluster
    /matrix=in(*)
    /plot=dendrogram.
```

「matrix」は、どういった距離情報をもとにクラスタ形成をおこなうかを指定するサブコマンドである。上の例のように< in(*) >と指定すれば、「現在データエディタにある情報をもとにクラスタ形成をせよ」という意味となる。つまり、先ほどの proximities コマンドの matrix サブコマンドにおいて、＜out(*)＞と指示したときに保存された距離行列の情報をもとにクラスタを形成するわけである。

クラスタ分析の過程を視覚的にわかりやすくするためには、一般的にデンドログラム（樹状図）を用いる（図 6）。デンドログラムを出力させるには、plot サブコマンドで、＜dendrogram＞と指定すればよい。

【具体例】

```
cluster
    /matrix=in(*)
    /plot=dendrogram.
```

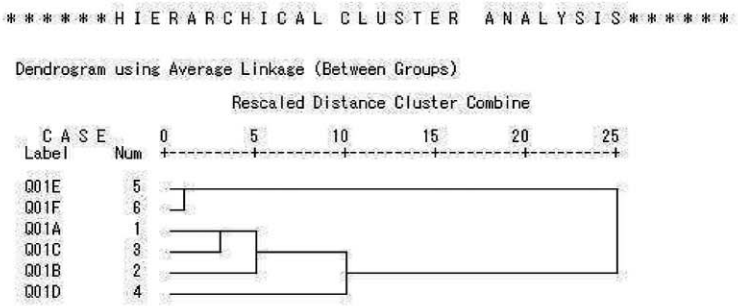

図 6　デンドログラムの例

具体例は、「現在データエディタにある距離行列データをもとに、クラスタ分析をせよ。なお、デンドログラムも同時に出力せよ。」という意味である。

ちなみに、missing サブコマンドのデフォルトは、<exclude>となっている（missing サブコマンドに関しては、4.4（3）を参照のこと）。

6　おわりに

以上が、主な分析にかかわるシンタックスについての説明である。本稿を一通り読まれた方に対して、若干のコメントを補足しておこう。

本稿で取り上げたコマンドのうちいくつかは、コマンドの文字を省略しても実行可能である。例えば、frequencies は freq、correleations は corr と省略できる。試しに実行してみるとよい。

また本稿では、コマンドには基本的にサブコマンドを付けるような説明をしてきたが、実はコマンドに対してサブコマンドがデフォルトで設定されていることがあるので、コマンドだけでも分析は可能な場合がある。それにもかかわらず、なぜサブコマンドについて説明したのかというと、どのような分析が SPSS 上でなされているかを把握してもらいたいと筆者たちは考えるからである。

昨今、使いやすい統計ソフトが入手しやすくなったおかげで、私たちは多変量解析などの複雑な分析を容易にできるようになった。分析するものにとって、このことは非常に喜ばしいことである。しかし、分析結果について質問を受ける教員やティーチング・アシスタントたちにとっては、少しやっかいなことでもある。

「こんな分析結果が出ましたけど、なんでですかねぇ」という質問に私たちはよく悩まされる。こうした質問が投げかけられる原因として、分析自体の理解不足だけでなく、統計ソフト、例えば SPSS 上でどのような分析がおこなわれているかを把握できていないことが考えられる。本稿はこのことを把握するのに役立つものであると思われる。なお、分析自体の理解については本稿の目的や主旨から外れるので、統計学ならびに社会調査関連の文献を参照されたい。

［参考文献］

雨森聡・山本圭三，2006，「SPSS マニュアル―データ読込から多変量解析まで」小林久高編『同志社大学 社会調査実習報告書』14（第 1 分冊）:391-425，同志社大学社会学科.

Durkheim, E., 1897, *Le suicide : etude de sociologie.* Paris : Felix Alcan.. （＝宮島喬訳，1985『自殺論』中央公論社）.

石村貞夫，2004，『SPSS による統計処理の手順［第 4 版］』東京図書.

————，2005，『SPSS によるカテゴリカルデータ分析の手順［第 2 版］』東京図書.

小林久高編，1993，『同志社大学 社会調査実習報告書』1，同志社大学文学部社会学専攻.

————，1998，「SPSS for Windows の使い方」『島根大学情報処理センター広報』9:39-50.

小林久高・雨森聡・山本圭三，2007，「SPSS を用いた社会調査データの分析マニュアル――シンタックスの解説を中心に」小林久高編『同志社大学 社会調査実習報告書』15（第 1 分冊）: 245-281，同志社大学社会学科.

Norusis,J,M, 1990, *The SPSS guide to data analysis*, SPSS Inc. （＝山本嘉一郎・森際孝司・藤本和子訳，1994『SPSS による統計学入門』東洋経済新報社）.

小野寺孝義・山本嘉一郎編，2004，『SPSS 事典　Base 編』ナカニシヤ出版.

武田祐佳，1993，「SPSS コマンド集」小林久高編『同志社大学 社会調査実習報告書』1: 359-64，同志社大学文学部社会学専攻.

太郎丸博，2005，『人文・社会科学のためのカテゴリカル・データ解析入門』ナカニシヤ出版.

土田昭司，1994，『社会調査のためのデータ分析入門』有斐閣.

内田治，2002，『すぐわかる SPSS によるアンケートの調査・集計・解析』東京図書.

————，2003，『すぐわかる SPSS によるアンケートの多変量解析』東京図書.

［参考になるウェブページ］

金沢大学文学部 岡田研究室（http://web.kanazawa-u.ac.jp/~tokada/spss/spss.htm）
関西大学社会学部 清水研究室（http://www2.ipcku.kansai-u.ac.jp/%7Eshimizu/spss.html）
國學院大學経済学部 小木曽研究室（http://www.socio.kyoto-u.ac.jp/info/spss.html）
筑波大学大学院システム情報工学研究科 石井研究室（http://infoshako.sk.tsukuba.ac.jp/~ishii/SPSS2003.pdf）