【セミナー】

社会調査データ分析のための Stata 入門

― データ読み込みから基礎的分析まで ―

山本　圭三
YAMAMOTO Keizo

1　はじめに

本稿は、統計ソフト「Stata（ステータ）」を用いた社会調査データの分析方法について解説するものである。

社会学における計量分析にはこれまで SPSS が用いられることが多かったが、ここ最近では Stata を使用する機会も増えてきた。Stata は基本言語が英語であるソフトだが、コマンドと呼ばれるプログラムを記述して分析をおこなうことができる点は SPSS と同様である。しかも Stata の強みはこのプログラムの作りやすさにあり、同じ分析でも Stata は SPSS に比べより簡単なプログラムですませることができる。さらに、分析をおこなう処理速度も Stata の方が圧倒的に速く、それゆえ Stata にある程度慣れたユーザであれば、SPSS よりもテンポよく分析を進めていくことができるのである。

現在出版されている Stata の入門書は経済分析を念頭において書かれたものが多く、社会学でおこなわれる計量分析に必要な操作について詳述しているものは少ない。そこで、本稿では Stata を用いて社会調査データの読み取りから基本的な分析をおこなうまでの方法について解説していくことにする[1)]。なお、本稿では Stata を用いた具体的な操作方法について解説していくこととし、分析自体を理解するための解説は省略する。統計学や社会調査に関する知識については、他の文献を参照していただきたい。

2　Stata を使う際の基本事項

2.1 Stata のメイン画面

Stata を起動すると、図 1 のような画面が立ち上がる。メイン画面は、(1) Stata Command、(2) Stata Results、(3) Review、(4) Variables、という 4 つのウィンドウで構成されている。順に説明しよう。

(1) Stata Command ウィンドウ

Stata では「コマンド」と呼ばれるプログラムを書き、それを実行することで実際の分析がおこなわれる。分析者が、このコマンドを実際に入力していく部分が Stata Command ウィンドウ（以下、Command ウィンドウと略記）である。

データが読み込まれた状態で Command ウィンドウ内に何らかのコマンドを入力し、最後にエンターキーを押せばその命令が実行される[2)]。Command ウィンドウ内には、コマンドを直接入力することもできるし、他のテキストエディタで作成したものを貼り付けることもできる[3)]。Command ウィンドウ内では自動的に改行がなされるため、コマンドが長くなったとしても特に気にせず入力していけばよい。ただし、Command ウィンドウに一度に入力できるのは 1 つのコマンドだけである。複数の異なるコマンドを実行したい場合でも、1 つずつ順に入力・実行しなければならない（複数のコマンドを一括して実行する方法については、5.1 (3) を参照のこと）。

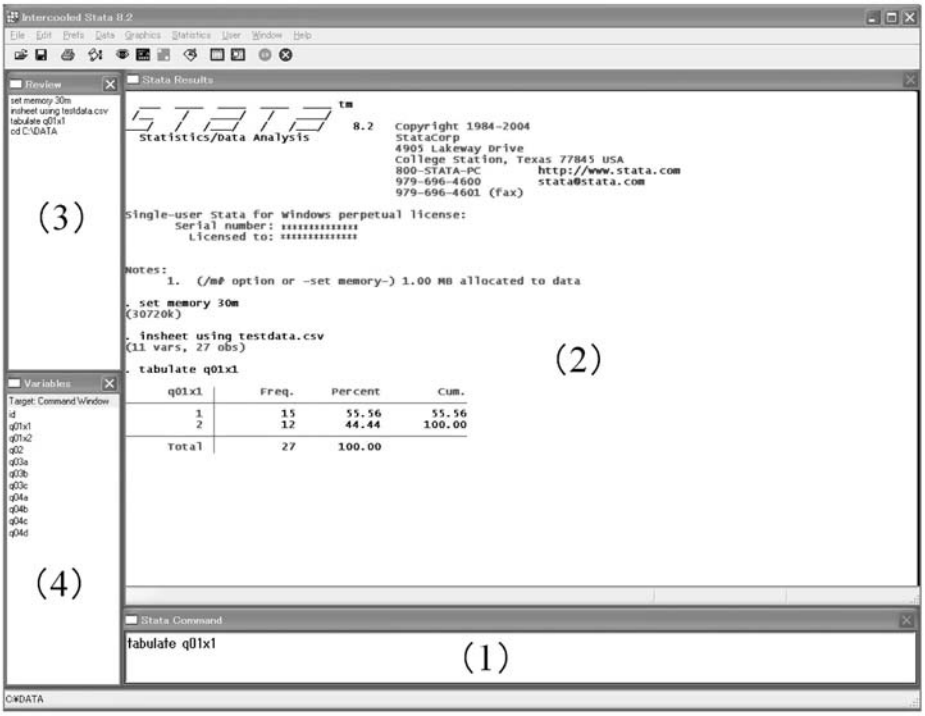


図 1　Stata のメイン画面

(2) Stata Results ウィンドウ

Stata Results ウィンドウ（以下、Results ウィンドウと略記）は、結果を表示する画面である。Command ウィンドウにプログラムを入力し実行すると、その命令に対する結果がこの Results ウィンドウに表示される。ただし、ここに文字などを入力することはできない。

分析結果を示す表などが大きくなった場合、Results ウィンドウには結果の一部分とともに「--more --」という記号が下部に表示される。この記号は、「結果をすべて表示できないため、一部分を隠している」ことを示すものであり、記号をクリックすると隠されている残りの部分が表示される。

(3) Review ウィンドウ

Review ウィンドウは、実行されたプログラムの一覧を示すものである。コマンドを実行するたび、そのコマンドが Review ウィンドウに追加されていく。Review ウィンドウを見れば、それまで自分がどのような分析をおこなってきたかがすぐにわかるようになっているのである。

また、Review ウィンドウに表示されているコマンドをクリックすると、Command ウィンドウに同じ文が入力される。類似したコマンドを続けて実行したい場合などは、これを利用すると便利である。

(4) Variables ウィンドウ

Variables ウィンドウは、変数のリストを表示する画面である。Stata のデフォルトでは、SPSS のように分析対象となるデータ自体を表示する画面が開かれない [4)]。データにどのような変数がある

のかを確認するためには、Variables ウィンドウを参照することになる。

また、Review ウィンドウと同様に、Variables ウィンドウに並ぶ変数名をクリックすると Command ウィンドウにその変数名が入力される。この機能を用いれば、コマンド入力の手間を省くことができる。

分析者は、以上の 4 画面を見ながらデータ分析を進めていくことになる。Command ウィンドウにコマンドを入力し、Results ウィンドウに出てくる分析結果を確認する。Review ウィンドウや Variables ウィンドウに表示されているリストを駆使しつつ、この作業を何度も繰り返していくのである。

2.2 分析をはじめる前に

(1) コマンドとオプションについて

Command ウィンドウに入力するコマンドは、基本的にメインのコマンド部分と、オプション部分から構成される。メインのコマンドとは、どのような変数を用いてどのような分析をおこなうのかを指定するものであり、先頭にコマンド名（処理内容の宣言）を入れ、その後に使用する変数を入力していくというかたちをとる。一方、オプションとはコマンドの内容をより細かく指定する際に使用するものである。オプションは、メインのコマンド部分の後にカンマを入れ、続けて入力していくことになる。

例えば、図 2 のコマンドは、「q01x1 と q04a によるクロス表を作成せよ。ただし、セル内には度数のほかに行%も入れ、かつχ2 乗値も算出せよ。」という意味であり、このうち下線部がオプションによって指定されている部分である。

図 2　コマンドとオプションの例

それぞれのコマンドにはいくつものオプションが設けられており、それぞれ詳しく紹介することは紙幅の関係上難しい。このため、本稿では基本的な分析をおこなうのに必要なオプションのみ紹介することにし、その他のオプションについては説明を省くことにしたい。

(2) コマンドの省略について

Stata では、コマンドの一部を省略できるようになっている。例えば、後で紹介する【tabulate】コマンドは「tab」、【generate】コマンドは「gen」といったように、かなり省略することができる。こうしたコマンドの省略は、実際に分析をおこなっていく際の時間短縮になるため非常に有効である。

本稿では、コマンドを紹介する際、省略可能なものについては直後のカッコ内に省略したかたちを記しておくことにする。読者はこうした記述を参考にしつつ、実際に短縮可能であることを確認していただきたい。

(3) 作業用フォルダについて

Stata をインストールすると、自動的に「作業用フォルダ」というものが作成される。作業用フォルダとは、Stata の処理に必要なファイルを保存しておくフォルダであり、具体的にはデータセットや2.3 で解説する Do ファイルなどを保存することになる。

現在どのフォルダを作業用フォルダとしているかは、Stata のメイン画面最下部（Variables ウィン

ドウと Command ウィンドウの下）に表記されている。インストール時に特に指定しない限り、作業用フォルダは「DATA」という名前で C ドライブに作成されている。作業用フォルダに何かファイルを追加したい場合には、マイコンピュータから C ドライブを選択し、「DATA」というフォルダを探してそこに保存すればよい。

（4）使用するメモリの設定

Stata では通常、パソコン上で処理をおこなう際に使用するメモリが 1MB に設定されている。このため、データセットにおけるサンプル数が大きいときや複雑な処理を一気におこなう場合などは、メモリ不足が起こり処理にかなり時間がかかってしまう。このような事態を避けるため、Stata を起動して分析をはじめる前には、【set】コマンドを用いて使用するメモリの容量を変更しておくとよい。

○コマンド： set

```
set memory メモリ容量の数値
```

◇コマンド例（1）： set

```
set memory 30m
```

「メモリ容量の数値」部分には、「○○m」という具合に設定したいメモリ容量の数値を具体的に入力する（「m」は MB という意味である）。メモリ容量をいくらに設定するのかは、分析者が決めればよい。コマンド例（1）は、「使用するメモリ容量を 30MB に設定せよ」という意味である。

2.3 Do ファイルについて

一般的に、データ分析は 1 日で終わることはあまりなく、何日もかけておこなうことのほうが多い。このような場合、以前おこなった分析コマンドをまとめて保存しておき、後日また同じものを実行できるようにしておくと、コマンドを入力する手間が省けて便利である。

こうした状況に備え、Stata には複数のコマンドをまとめて保存しておき、一度に実行できるような機能が設けられている。具体的には、「Do ファイル（拡張子は「.do」）」と呼ばれるコマンドをまとめたファイルを作成・保存しておくことで、一度に複数のコマンドが実行できるうえに、何度でも同じ分析が再現できるようになっているのである。

以下では、Do ファイルの作成する 2 つの方法と、保存されたファイルの実行方法について解説しよう。

（1）Review ウィンドウを用いる方法

2.1（3）で述べたように、Review ウィンドウにはそれまで実行されたコマンドの一覧が示されている。このとき、Review ウィンドウ上で右クリックし「Save Review Contents」を選択すると、保存するためのダイアログが出てくる（図 3）。

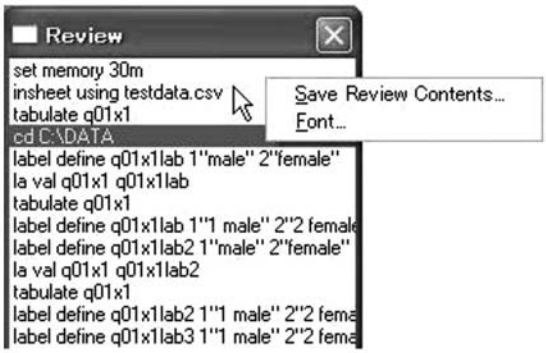


図 3　Review ウィンドウ内のコマンドを保存する

出てきたダイアログで適当な名前を入力して保存すると、2.2（3）で紹介した作業用フォルダに

Do ファイルが作成される。このようにして、実行した一連のコマンドを全て保存しておくことができる。

（2）外部のテキストエディタを用いる方法

Do ファイルは、基本的にテキスト情報からなるファイルである。このため、メモ帳や Sakura エディタなど外部のテキストエディタを用いて Do ファイルを作成することもできる。

外部エディタを用いて Do ファイルを作成する方法はごく簡単である。一連のコマンドを入力して「名前を付けて保存」する際に、ファイルの拡張子を「.do」にするだけでよい。Stata がインストールされているパソコンであれば、このようにして保存したファイルは Do ファイルとして認識される。

（3）保存された Do ファイルの実行

Do ファイルを Stata で実行するためには、まず Do ファイルを 2.2（3）で指定した作業用フォルダに入れておく必要がある。Do ファイルが作業用フォルダに入っている状態で、以下のコマンドを実行すれば、Do ファイルの内容がすべて実行される。

○コマンド： do

```
do do ファイル名
```

◇コマンド例（2）： do

```
do vardefcommands.do
```

コマンド例（2）は、「作業用フォルダに保存されている『vardefcommands.do』ファイルの内容を実行せよ」という意味である。

ちなみに、Do ファイル内において別の Do ファイルを実行させるコマンドを入れることもできる。Do ファイルのようなプログラムファイルは内容を読み取りやすいことも重要であるため、必要に応じてこうした機能を用いるとよいだろう。

3　分析用データの作成――読み込み・ラベル貼り付け・データ変容・保存

3.1 データの読み込み：insheet コマンド

分析を始める際、最初におこなうのは分析をするためのデータ（ここでは、「データセット」と呼ぼう）を Stata に読み込ませる作業である。まず、この方法から説明しよう。

社会調査では、データセットを作成するのにエクセルを用いるのが一般的である。エクセルでデータセットを作成した場合、たいていは図 4 のように 1 行目に変数名が、2 行目以降に実際のデー

	A	B	C	D	E	F	G
1	ID	q1x1	q1x2	q2	q3a	q3b	q3c
2	1	1	18	5000	7	5	0
3	2	1	19	50000	7	5	3
4	3	2	18	5000	5	6	1
5	4	2	18	40000	6	5	1
6	5	2	20	80000	6	4	2
7	6	2	18	30000	8	5	1
8	7	1	18	30000	7	6	0

図 4　エクセルで作成したデータセット（カンマ区切り形式）の例

タが入力された状態になる。データセットのファイルはエクセル形式（拡張子が「.xls」もしくは「.xlsx」）、あるいはカンマ区切り形式（拡張子が「.csv」）であることが多い。

このようなデータセットを Stata に読み込ませるためには、【insheet】コマンドを用いる。ここでは、カンマ区切り形式のファイルを読み込む方法を紹介しよう。

○コマンド： insheet

```
insheet using ファイルの場所¥ファイル名.csv
```

◇コマンド例（3）： insheet

```
insheet using c:¥data¥testdata.csv
```

「ファイルの場所」は、データセットが置かれているパソコン上の場所、「ファイル名」はデータセットにつけられている名前を指している。上のコマンド例（3）では、C ドライブの「data」フォルダにある「testdata.csv」というデータセットを読み込むことになる。

コマンドを実行すると、Stata にデータが読み込まれる。このとき、元のファイルで 1 行目に入力されていた文字列は自動的に変数名として認識される。読み込みがきちんとなされていれば、Variables ウィンドウに変数名が並んだ状態になる。

なお、変数名にハイフンやスペースを使用することはできない。これらは読み込み時に削除されてしまうため（例えば「q01-a」→「q01a」）、使用している場合は、あらかじめアンダーバーなどに変更しておくことが望ましい。また、Stata ではエクセル形式のファイルを読み込むことができない。エクセル形式でデータセットが作成されている場合、一度カンマ区切り形式に保存しなおした上で上記の方法をとることになる。

3.2 ラベルの貼り付け： label コマンド

（1）変数ラベルの貼り付け

データセットを作成する際には、変数名に「q01」や「q15a」などといった問番号を使用するのが一般的である。しかし、このような変数名ではその変数が何を意味するものであるかが一目で判断できない。仮にこのような名前で分析結果の表が出力されても、結果を読み取るのに時間がかかってしまうのである。

このため、データの読み込みを終えた後には、1 つ 1 つの変数に対してその内容を表す「ラベル」を貼り付けておく作業が必要になる。この作業には、【label】コマンドが用いられる。

○コマンド： label （変数ラベルの貼付）

```
label variable 変数名 変数ラベル
（la var 変数名 変数ラベル）
```

◇コマンド例（4）： label（変数ラベルの貼付）

```
label variable q01 sex
```

「variable」は、「変数にラベルを貼る」という宣言を意味する。「variable」の後にはラベルを貼りたい変数名を入力し、ラベルとして貼り付けたい文字列を変数名の後に入れる [5]。ラベルの部分は特にカッコなどで囲う必要はないが、ラベルに使用できるのは半角の英数字のみである。上のコマンド例（4）は、「『q01』という変数に対し、『sex』というラベルを貼り付けよ」という意味である。

また、一度貼り付けたラベルを削除したい場合も、同様のコマンドを用いる。ラベルを削除する場合には、ラベルの文字列を入力する部分を空白のままにしたコマンドを実行すればよい。下のコマンド例（5）は、「『q01』という変数に、空のラベルを貼り付けよ」、すなわち「『q01』という変数

に貼られているラベルを削除せよ」という意味のコマンドである。

◇コマンド例（5）： label（変数ラベルの削除）

```
label variable q01
```

（2）値ラベルの貼り付け：

ラベルの貼り付けは、変数だけでなく変数の値についてもおこなっておく必要がある。量的変数であれば特に問題はないが、質的変数の場合、値が何を意味するのかが一目では分かりにくいからである。

変数の値にラベルを貼り付ける作業にも、【label】コマンドが用いられる。ただし、変数ラベルの貼り付けとは異なり、（ア）値ラベルフォーマットの定義、（イ）変数に対する値ラベルフォーマットの適用、という 2 段階の作業をおこなうことになる。

まず、（ア）の段階において、値とそれに貼り付けるラベルの組み合わせをあらかじめ作成する。1 には○○というラベル、2 には××というラベル…という具合に、それぞれの値にどのようなラベルを貼り付けるのかについての規則（本稿ではこれを「値ラベルフォーマット」と呼ぶことにする）を先に決めておくのである。そして（イ）の段階で、定義された値ラベルフォーマットを、どの変数に適用するのかを設定する。この段階で、はじめて変数の値にラベルが貼られることになるのである。

（ア）の段階である値ラベルフォーマットの定義には、以下のようなコマンドが用いられる。

○コマンド： label （値ラベルフォーマットの作成）

```
label define 値ラベルフォーマット名 値 1"値ラベル 1" 値 2"値ラベル 2"
（la de 値ラベルフォーマット名 値 1"値ラベル 1" 値 2"値ラベル 2"）
```

「define」の後に入力された文字列が、ここで定義される値ラベルフォーマットの名前になる（名前は分析者が決めればよい）。フォーマット名に続いて、値と値ラベルを順に入力していくことになるが、ここでは値ラベルにあたる文字列を半角の二重引用符（”）で囲わなければならない（ラベルは半角英数のみ）。ちなみに、上では値 2 までしか書かれていないが、値が 3 つ以上ある場合は続けて入力していけばよい。

値ラベルフォーマットの定義ができたら、以下のようなコマンドで変数にフォーマットを適用させる。

○コマンド： label （変数に値ラベルフォーマットを適用）

```
label values 変数名 値ラベルフォーマット名
（la val 変数名 値ラベルフォーマット名）
```

「values」は、「値ラベルフォーマットの適用をおこなう」という宣言を意味する。変数名の後に、あらかじめ定義しておいた値ラベルフォーマット名を入力して実行すれば、値にラベルがつけられる[6)]。

◇コマンド例（6）： label（値ラベルフォーマットの作成・適用）

```
label define sexvalform 1"male" 2"female"

label values q01 sexvalform
```

コマンド例(6)では、まず 1 行目のコマンドで、1 に「male」、2 に「female」という値ラベルを与える「sexvalform」という値ラベルフォーマットが

定義されている。そして、2行目のコマンドで「q01に対して『sexvalform』という値ラベルフォーマットを適用せよ」という命令がなされている。

◇コマンド例（7）： label（複数の変数に値ラベルの適用）

```
label define agree4 1"agree" 2"modera
  tely agree" 3"moderately disagree"
  4"disagree"

label values q10a agree4

label values q10b agree4
```

コマンド例（7）は選択肢が共通している変数に対して値ラベルを貼り付けるコマンドの例である。最初のコマンドで、1に「agree」、2に「moderately agree」、3に「moderately disagree」、4に「disagree」というラベルを与える「agree4」という値ラベルフォーマットが定義されている。次のコマンドで、「q10a に対して『agree4』という値ラベルフォーマットを適用せよ」、最後のコマンドで「q10b に対して『agree4』という値ラベルフォーマットを適用せよ」という命令がなされている。

また、変数ラベルのときと同様に一度貼り付けた値ラベルを削除することもできる。コマンド例（8）は、「『q01』という変数に、空の値ラベルフォーマットを適用せよ」、すなわち「『q01』という変数に貼られている値ラベルを削除せよ」という意味である。

◇コマンド例（8）： label（ラベルの削除）

```
label values q01
```

3.3 データの変容

データ変容は、分析を進める際に頻繁におこなうことになる作業である。もともとの変数だけで分析がすむことはほとんどなく、変数を新しく作成したり、変数の値を変更したり、選択肢を統合したりする必要がかならずと言ってよいほどでてくる。次に、このデータ変容コマンドについて説明しよう。

（1）計算式を使用した新変数の作成：generate コマンド

もともとの変数から何らかの新しい変数を作成する場合には、【generate】コマンドが用いられる。

○コマンド： generate

```
generate 新変数名 = 命令文
（gen 新変数名 = 命令文）
```

「新変数名」は新たに作成される変数の名前であり、分析者の側で決めればよい（ただし半角英数のみ）。「命令文」の部分には、算術記号（＋、－、＊、／）やカッコを用いた計算式を用いることができる。例えば、「q21x1（本人の収入）」と「q21x2（配偶者の収入）」から、夫婦合算の収入（gassan）をあらわす変数を作成する場合、下のコマンド例（9）のようなコマンドを入力すればよい。

◇コマンド例（9）： generate（計算式による新変数の作成）

```
generate gassan = q21x1 + q21x2
```

また、命令文には関数を用いることもできる。関数によって変数の値を変換し、それを新しい変数として保存できるのである。

○コマンド： generate（関数を用いた新変数の作成）

```
generate 新変数名 = 関数名(変換に用いる変数)
(gen 新変数名 = 関数名(変換に用いる変数))
```

計算式の代わりに関数名を入れ、カッコ内に変換に用いる変数名を入力すれば、関数を用いた新変数の作成ができる。表 1 に挙げているような関数は、頻繁に用いられることになるだろう。

表 1　よく用いられる関数

関数	内容
abs(A)	Aの絶対値
sprt(A)	Aの平方根
log(A)	Aの自然対数
log10(A)	Aの常用対数
exp(A)	*e*のA乗

◇コマンド例（10）： generate （関数を用いた新変数の作成）

```
generate nq17 = log(q17)
```

コマンド例（10）は、量的変数である q17 の自然対数を値とする新変数を作成するコマンドである。具体的には、「q17 の値の自然対数を計算し、それを nq17 という変数名で保存せよ」という意味である。

（2）論理式を使用した新変数の作成：replace コマンド

新しい変数は、計算式や関数だけでなく論理式を用いて作成することもできる。この場合、【generate】コマンドに加え、【replace】コマンドを用いることになる。

○コマンド： replace

```
replace 変数名=変更後の値 if 論理式
```

【replace】コマンドは、既存の変数の値を変更するコマンドである。値を変更するときに、下の表 2 にあるような演算子を用いた論理式によって条件を設定することができる。

表 2　関係演算子および論理演算子

演算	Stataでの表現
A ＞ B	A > B
A ＜ B	A < B
A ≧ B	A >= B
A ≦ B	A <= B
A = B	A == B
A ≠ B	A != B
A かつ B	A & B
A または B	A | B

表 2 のうち、「＝」に関しては注意が必要である。いわゆる数学的な意味での等号（＝）は、Stata 上では「A＝＝B」などとイコールを 2 つ重ねて表記することになる。イコール 1 つで「A＝B」と記述した場合、「B を A として定義する」「B を A に代入する」といった意味になる（主として変数や値の作成・変換の際に用いられる）。

【replace】は値を変更するコマンドなので、新しい変数を作成する場合は（1）【generate】コマンドで新変数を作成する、（2）【replace】コマンドを用いて論理式に基づいた値の変更をおこなう、という作業をおこなうことになる。例えば、q01x1（性別の質問）と q31（婚姻状態の質問）から、「mtype（性別と婚姻状態の組み合わせ類型）」を作成する場合、コマンド例（11）のような文を入力することになる。

◇コマンド例（11）： generate replace（論理式を用いた新変数の作成）

```
generate mtype = .

replace mtype = 1 if q01x1==1 & q31==1
replace mtype = 2 if q01x1==1 & q31==2
```

```
replace mtype = 3 if q01x1==2 & q31==1
replace mtype = 4 if q01x1==2 & q31==2
```

コマンド例（11）では、1 行目で「mtype」という変数を作成している。ただしこの状態では、すべてのサンプルが「mtype」の値を持たない。2 行目～5 行目のコマンドによって、それぞれの条件に当てはまるサンプルに mtype の値が割り当てられることになる（if 以下で「＝＝」が用いられていることに注意すること）。例えば、2 行目は「q01x1 が 1 であり、なおかつ q31 が 1 であるサンプルは、mtype の値を 1 とせよ」という意味である。

（3）サンプルの等分・変数の標準化： egen コマンド

分析によっては、サンプルの等分や変数の標準化をおこなう場合もある。サンプルの等分とは、例えば世帯収入をもとに、サンプル全体を「上」、「中」、「下」と 3 分割するといった場合などである。このような場合、【egen】コマンドを用いることになる。【egen】は【generarte】コマンドと同様に関数を用いて新変数を作るものであるが、より複雑な関数を使うことができるコマンドである。

○コマンド： egen（サンプルの等分）

```
egen 新変数名=cut(もとの変数名), group(分割数)
```

サンプルの等分をおこなう場合、上のようなコマンドを用いる。「cut」も関数の一種であり、［group（分割数）］というオプションを伴なう。カッコ内に、分割したい数を入力すればよい。

◇コマンド例（12）： egen（サンプルの等分）

```
egen c3q38=cut(q38), group(3)
```

コマンド例（12）は、量的変数である q38 をもとに全体を 3 等分する場合の例である。具体的には、「q38 の値をもとに、全体を 3 等分した「c3q38」という新しい変数を作成せよ」という意味である。新しい変数の値は、もともとの変数の値が小さいグループから順に 0、1、2、…という具合に自動的に与えられる。

既存の変数を標準化したものを新変数として保存する際にも、【egen】コマンドを使用する。用いる関数は「std」である。

○コマンド： egen（変数の標準化）

```
egen 新変数名=std(もとの変数名)
```

◇コマンド例（13）： egen（変数の標準化）

```
egen zq35=std(q35)
```

コマンド例（13）は、量的変数である q35 を標準化する場合の例である。具体的には、「q35 の値を標準化した、zq35 という新しい変数を作成せよ」という意味である。

（4）値の再割り当て： recode コマンド

分析によっては、もともとの変数の値を別の値に変更したい場合も出てくる。例えば、ある質問に対して「1.そう思う」「2.ややそう思う」「3.あまりそう思わない」「4.そう思わない」という 4 段階で回答してもらったとしよう。この回答のままの状態だと、データとしては数値が大きくなるほどこの質問に対して「そう思わない」度合いが強いことを示すことになる。数値が大きくなるほど「そう思う」度合いが強いことを示すようにするためには、「そう思う」が 4、「ややそう思う」が 3、…「そう思わない」が 1、といった具合に値

を反転させなければならない。このような値の変更には、先に紹介した【replace】コマンドよりも【recode】コマンドの方が便利である。

○コマンド： recode （値の再割り当て）

```
recode 変数名 (変更前の値 = 変更後の値),
  gen(新変数名)
```

「recode」の後に値を変更したい変数名を入力し、その後に続けて具体的な変更の規則を入力していく。1 つの変数について、変更を複数おこないたい場合はカッコを並列して記述していけばよい。

カッコの後にある「, gen(新変数名)」の部分は、値の変更が行われた状態を、元の変数とは別の新しい変数として保存するための命令にあたる。【recode】コマンドによって変数の値を変更する場合、変更前の値を変更後の値に上書きする形で処理がなされる。つまり、ある変数に【recode】コマンドを実行した時点で、もともとの情報が変更後の情報に変わってしまい、変更前の状態に戻ることはできなくなってしまうのである。元の情報が失われてしまうような事態を避けるため、【recode】コマンドを実行する際には必ず「, gen(新変数名)」という部分も記述しておくとよい（名前は分析者が決める）。値の変更をおこなったものを別の変数として保存することで、もともとの情報も守っておくことができるからである[7]。

◇コマンド例（14）： recode（値の再割り当て）

```
recode q05 (1=5) (2=4) (3=3) (4=2) (5=1),
  gen(rq05)
```

コマンド例（14）は、「q05 の値をもとに、1 が 5 に、2 が 4 に、3 が 3 に、4 が 2 に、5 が 1 になるような rq05 という変数を作成せよ」という意味である。

また、複数の値を 1 つの値に統合する場合も【recode】コマンドを使うとよい。

○コマンド： recode （複数の値を 1 つの値に統合）

```
recode 変数名 (変更前の値 A 変更前の値 B
  = 変更後の値), gen(新変数名)
```

◇コマンド例（15）： recode（複数の値を 1 つの値に統合）

```
recode q05 (1 2=1) (3=2) (4 5=3),
  gen(nq05)
```

複数の値を 1 つに統合する場合は、統合したい値をスペースで区切って並列すればよい。上では 2 つの値を並列しているが、3 つ以上の値を並列することもできる。

コマンド例（15）は、「1.そう思う」「2.ややそう思う」「3.どちらでもない」「4.あまりそう思わない」「5.そう思わない」という 5 段階の回答を、「1.そう思う」「2.どちらでもない」「3.そう思わない」という 3 段階の回答に統合するコマンドの例である。具体的には、「q05 の値をもとに、1 と 2 が 1、3 が 2、4 と 5 が 3 になるような nq05 という変数を作成せよ」という意味である。

また、【recode】コマンドでは、変更したい値を範囲で指定することもできる。

○コマンド： recode （範囲で指定した値を 1 つの値に統合）

```
recode 変数名 (変更前の値 A / 変更前の値
  B = 変更後の値), gen(新変数名)
```

変更したい値を範囲で指定する場合は、範囲の初めと終わりの値をスラッシュ（/）でつなげれば

よい。範囲の指定には具体的な数値のほかに、変数の最大値（max）、最小値（min）なども使うことができる。

◇コマンド例（16）： recode（範囲で指定した値を 1 つの値に統合）

```
recode age (min/19=1)(20/29=2)(30/39=3)(40/max=4), gen(ageclass)
```

コマンド例（16）は、年齢をあらわす「age」をもちいて、20 歳未満、20 代、30 代、40 歳以上、という年齢階層をあらわす「ageclass」を作成するコマンドである。具体的には、「age の値をもとに、最小値～19 までが 1、20～29 までが 2、30～39 までが 3、40～最大値までが 4、という値をとるような ageclass という変数を作成せよ」という意味である。

3.4 データの保存： save コマンド

3.1（1）で、カンマ区切り形式のデータセットを読み込む方法を紹介したが、このとき、Stata は「Stata Dataset 形式（拡張子は「.dta」）」という独自のファイル形式でデータを認識している。この Stata Dataset 形式のデータを保存しておけば、いったん Stata を終了し、再び起動させる際にデータの読み込みや定義の作業を省略することができる。このようにデータを保存するためには、【save】コマンドを用いる。

○コマンド： save

save ファイル名
（sa ファイル名）

◇コマンド例（17）： save

```
save testdata
```

上のようなコマンドを実行すれば、2.2（3）で紹介した作業用フォルダに指定した名前のファイルが保存される。コマンド例（17）は、「現在のデータを、『testdata』という名前で作業フォルダに保存せよ」という意味である。作業用フォルダ以外に保存したい場合は、ファイル名の前に場所を示す情報を記述すればよい。

【save】コマンドで保存されたデータは、【use】コマンドで開くことができる。コマンド例（18）は、「作業用フォルダに保存されている『testdata』という名前のファイルを開け」という意味である。

○コマンド： use

use ファイル名
（u ファイル名）

◇コマンド例（18）： use

```
use testdata
```

ただし、上記のような方法でデータセットを保存することはあまり実用的ではない。というのも、分析後の状態でデータセットを保存したとしても、分析に際しておこなった変数加工の経緯を把握しておかなければ、データセットはわけのわからないものでしかないからである。

むしろ実際には、データセットよりも 2.3 で紹介した Do ファイルを保存しておくほうが便利である。Do ファイルがあれば、データの読み込みから加工、そして分析までを簡単に再現することができる。しかも Do ファイルは基本的にはテキスト形式であるため、容量が Stata Dataset 形式のファイルよりもずっと小さい。わざわざ容量の大きい Stata Dataset 形式のファイルを保存しておくよりも、Do ファイルを用いて読み込みから始めるほうが間違いをおかす可能性も低く、効率的なのである。

ID	q01x1 性別（1：男性、2:女性）	q01x2 年齢	q02 1ヶ月に自由に使えるお金	q03a 睡眠時間	q03b 大学での勉強時間	q03c 大学外での勉強時間	q04a 世の人々への信頼度	q04b 家族への信頼度	q04c 友人への信頼度	q04d 新聞への信頼度
							（1:信頼している～4:信頼していない）			
1	1	18	5000	7	5	0	2	1	2	1
2	1	19	50000	7	5	3	2	1	1	3
3	2	18	5000	5	6	1	3	1	1	3
4	2	18	40000	6	5	1	3	1	1	2
5	2	20	80000	6	4	2	3	2	2	2
6	2	18	30000	8	5	1	2	1	1	1
7	1	18	30000	7	6	0	2	1	1	2
8	1	19	50000	12	5	0	3	2	2	2
9	1	20	30000	6	5	1	3	3	2	2
10	1	18	30000	6	6	1	2	4	2	2
11	1	18	10000	2	2	8	1	2	1	2
12	2	19	5000	6	5.5	1	2	1	1	2
13	1	19	30000	8	6	1	3	2	2	3
14	1	20	30000	6	4	0	3	1	2	3
15	2	20	20000	6	6	1	2	1	1	2
16	1	18	40000	7	5	0	4	1	1	2
17	2	20	2000	6.5	4.5	1	2	1	1	2
18	2	18	10000	8	6	5	2	1	1	2
19	2	18	15000	6	6	0	3	1	2	1
20	2	20	100000	6	5.5	2.5	3	1	2	2
21	1	18	30000	6	3	1	2	2	2	4
22	2	18	40000	7	6	3	3	2	2	2
23	1	19	30000	6	6	0	3	3	3	3
24	1	18	20000	12	2	6	3	4	3	3
25	1	18	80000	10	5	0	3	2	2	2
26	2	20	100000	6	6	1	3	1	1	1
27	1	19	999999	8	4.5	2	4	3	3	4

図 5　架空のデータセット

4　基礎的な分析

ここからは具体的な分析のコマンドについて説明していく。以下では、図 5 に示すような架空のデータを用いて分析コマンドの例と分析結果の例を示しつつ解説していくことにする。

4.1 単純集計――1 変数についての分析

複雑な分析をはじめる前に、自分が使おうとしている変数がそもそもどのような分布をしているのかを把握しておくことは重要である。まず、このような単純集計の方法から説明することにしよう。

（1）度数分布表の作成： tabulate コマンド（1）

質的変数の単純集計には、度数分布表を用いるのが一般的である。度数分布表の作成には、【tabulate】コマンドを用いる。「tabulate」の後に変数名を入力し実行すれば、その変数についての度数分布表が出力される[8]。

○コマンド： tabulate

```
tabulate 変数名
（tab 変数名）
```

◇コマンド例（19）： tabulate

```
tabulate q01x1
```

コマンド例（19）は、q01x1（性別）の度数分布表を作成するためのコマンドである。このコマンドを実行した場合、図 6 のような表が出力される。表のうち、「Freq.」は度数、「Percent」は相対度数、「Cum.」は累積相対度数を示している。

```
. tabulate q01x1
```

q01x1	Freq.	Percent	Cum.
1 male	15	55.56	55.56
2 female	12	44.44	100.00
Total	27	100.00	

図 6　出力された度数分布表の例

```
. summarize q03a q03b q03c
```

Variable	Obs	Mean	Std. Dev.	Min	Max
q03a	27	6.907407	2.000178	2	12
q03b	27	5	1.160239	2	6
q03c	27	1.574074	1.974373	0	8

図 7　出力された記述統計量の例

(2) 記述統計量の算出： summarize コマンド

量的変数の単純集計には、平均値、分散、標準偏差、最大値、最小値などが用いられる。これらの記述統計量を算出するために用いるのが、【summarize】コマンドである。

○コマンド： summarize

```
summarize 変数名
(su 変数名)
```

◇コマンド例（20）： summarize

```
summarize q03a q03b q03c
```

「summarize」に続いて変数名を入力し実行すれば、その変数についての記述統計量が算出される。複数の変数について算出したい場合は、変数名をスペースで区切って並列させればよい。

コマンド例（20）は、q03a（睡眠時間）、q03b（大学での勉強時間）、q03c（大学以外での勉強時間）の記述統計量を算出させるコマンドである。このコマンドを実行した結果、図 7 のような表が出力される。表のうち、「Obs」は分析対象となったサンプル数、「Mean」は平均値、「Std. Dev.」は標準偏差、「Min」は最小値、「Max」は最大値を示している。

4.2 クロス表の作成――質的変数間の関係：tabulate コマンド（2）

分析に用いる変数の分布が把握できれば、次に変数間の関係を確認するための分析に移ることになる。変数間の関係といっても、用いる変数が質的変数であるか量的変数であるかによって、分析方法は異なってくる。ここではまず、質的変数間の関係を確認するためのクロス表を作成する方法について説明しよう。

クロス表を作成する際には、既に紹介した【tabulate】コマンドを用いる。ただし、変数の並べ方やオプションの指定などいくつか注意する点があるので、改めて述べることにしよう。

○コマンド： tabulate（クロス表の作成）

```
tabulate 変数名 1 変数名 2, row column
  chi2
(tab 変数名 1 変数名 2, r co ch)
```

```
. tab q01x1 q04c, row chi2
```

Key
frequency *row percentage*

q01x1	q04c 1 agree	2 mod. ag	3 mod. di	Total
1 male	4 26.67	8 53.33	3 20.00	15 100.00
2 female	8 66.67	4 33.33	0 0.00	12 100.00
Total	12 44.44	12 44.44	3 11.11	27 100.00

Pearson chi2(2) = 5.4000 Pr = 0.067

図 8 出力されたクロス表の例

「tabulate」の後に、スペースを空けて変数名を 2 つ並列させれば、クロス表を作成するコマンドになる。出力されるクロス表では、並列された変数のうち先に入力した方が表側に、後に入力した方が表頭に配置される[9]。

オプションを指定しない状態でコマンドを実行すると、度数のみのクロス表が出力されることになる。それゆえ、行パーセントが必要な場合は「row」、列パーセントが必要な場合は「column」をオプションで指定しておかなければならない（もちろんどちらか一方でよい）。また、オプションで「chi2」と指定すれば、$\chi 2$ 乗検定の結果もあわせて出力される。

◇コマンド例（21）： tabulate（クロス表の作成）

```
tabulate q01x1 q04c, row chi2
```

コマンド例（21）は、q01x1（性別）と q04c（友人への信頼度）のクロス表を作成するためのコマンドである。具体的には、「q01x1 を表側に、q04c を表頭においたクロス表を作成せよ。ただし、度数のほかに行パーセントも計算し、かつ $\chi 2$ 乗値も算出せよ。」という意味である。このコマンドを実行すると、図 8 のような結果が出力される。各セルの上の数値は度数、下の数値は行パーセント（オプションで「column」を指定した場合は列パーセント）をあらわす。

表の下には、「母集団において 2 変数間に関連はない」という帰無仮説に基づく検定結果が示されており、「Pearson chi2」が $\chi 2$ 乗値、カッコ内が自由度、「Pr」が有意確率をあらわしている。ここでの結果は、自由度が 2 で $\chi 2$ 乗値が 5.400、有意確率は 0.067 であり、仮に有意水準を 5%（0.05）とした場合帰無仮説は棄却できない。それゆえ、コマンド例（21）でおこなった検定の結果、「母集団において q01x1 と q04c の間に関連があるとはいえない」という結論が得られる[10]。

4.3 相関係数の算出——量的変数間の関係：pwcorr コマンド

量的変数間の関係を確認するためには、一般に相関係数が用いられる。相関係数を算出するためのコマンドは、【pwcorr】コマンドである。

○コマンド： pwcorr

```
pwcorr 変数名 1 変数名 2, obs sig
```

（pwcorr 変数名 1 変数名 2, o sig）

「pwcorr」の後に、スペースを空けて変数名を並列すれば、入力された変数どうしの相関係数が算出される。変数名の部分には 3 つ以上の変数を入力することもでき、その際には入力した数ぶんの行数、列数をもつ相関行列が出力されることになる。

オプションを指定しない状態でコマンドを実行すると、相関係数のみが記された相関行列が出力される。サンプル数が必要な場合は「obs」、検定の結果が必要な場合は「sig」とそれぞれオプションで指定しておかなければならない。

◇コマンド例（22）： pwcorr

```
pwcorr q04a q04b q04c, obs sig
```

コマンド例（22）は、q04a（世の人々への信頼度）、q04b（家族への信頼度）、q04c（友人への信頼度）の相関行列を作成するためのコマンドである。具体的には、「q04a と q04b、q04a と q04c、q04b と q04c の相関係数をそれぞれ算出せよ。ただし、相関係数にあわせてサンプル数、検定結果も算出せよ。」という意味である。このコマンドを実行すると、図 9 のような結果が出力される。各セルの 1 行目の数値が相関係数、2 行目が検定結果である有意確率、3 行目がサンプル数をあらわしている。

検定は、「母集団において、両変数間の相関係数は 0 である」という帰無仮説をもとにおこなわれている。コマンド例（22）でおこなった検定の結果のうち、例えば q04a と q04c であれば相関係数が 0.4665、有意確率は 0.0142 となっており（サンプル数は 27）、仮に有意水準を 5%（0.05）とする場合帰無仮説は棄却できる。それゆえ、「母集団において、q04a と q04c の間には（正の）相関関係があるといえる」という結論が得られることになる。

```
. pwcorr q04a q04b q04c ,obs sig
```

	q04a	q04b	q04c
q04a	1.0000		
	27		
q04b	0.1782	1.0000	
	0.3739		
	27	27	
q04c	0.4665	0.7324	1.0000
	0.0142	0.0000	
	27	27	27

図 9　出力された相関行列の例

4.4 2 グループ間での平均値の差——質的—量的変数間の関係 1

質的変数と量的変数の関係を確認するためには、平均値の差を検討していくことになる。このとき、検討に用いる質的変数が 2 つの値をとるものか、それとも 3 つ以上の値をとるものかで、分析法が変わってくる。まず、検討に用いる質的変数が 2 つの値をとる場合について、図 5 のデータのうち、q01x1（性別）と q02（自由に使えるお金）の関連を検討する例を用いて解説しよう。

2 つの値をとる質的変数と量的変数の関係を確認する際には、「t 検定」と呼ばれる平均値の差の検定がおこなわれる。非常に大ざっぱに言えば、質的変数の値ごとに人々を 2 つのグループに分け、それぞれの平均値に差があるかどうかを検討していくのである。ただし t 検定の方法にもいくつかあり、「2 つのグループの母分散が異なる」場合と「2 つのグループの母分散が等しい」場合で検定法が変わってくる。このため、t 検定をおこなう際には、まず（1）2 つのグループ間で母分散が異なるかどうかの検定をおこない、その後で（2）

2 つのグループ間で平均値が異なるかどうかの検定（すなわち t 検定）をおこなう、という手順をふむことになる。

(1) 2 つのグループ間での母分散の異同：sdtest コマンド

母分散がグループ間で異なるかどうかの検定をおこなうコマンドは、【sdtest】である。

○コマンド： sdtest

```
sdtest 量的変数名 , by(質的変数名)
```

c コマンド例（23）： sdtest

```
sdtest q02 , by(q01x1)
```

「sdtest」の後ろに、量的変数の名前を入力し、「by」の後ろには（2 つの値をとる）質的変数の名前を入力する[11]。コマンド例（23）は、「q02 ついて、q01x1 の値ごとに母分散を比較する検定をおこなえ」という意味である。このコマンドを実行すると、図 10 のような結果が出力される。

出力された結果は、上半分が記述統計量、下半分が検定結果という構成になっている。下半分の検定結果のうち、①の部分には帰無仮説（H_0）が示されている。ここでの帰無仮説は、「2 つのグループ（q01x1 の値が「1」のグループと「2」のグループ）の母分散は等しい」であるため、「sd(1 male) = sd(2 female)」となっている（sd は「Std. Dev.」の略）。さらに、いちばん下の 2 行には 3 種類の対立仮説（H_a）とそれに伴う検定結果（有意確率）が示されている。ここでの目的は、「2 つのグループ間で母分散が異なるかどうか」を検討することであるため、対立仮説が「sd(1) != sd(2)」である②の部分を見ることになる（「!=」は「≠」の意味である）。

コマンド例（23）でおこなった検定における有意確率は「0.0252」であり、仮に有意水準を 5%（0.05）とする場合帰無仮説は棄却できる。したがって、「2 つのグループ間で母分散は異なる」と判断できる。

```
. sdtest q02 ,by(q01x1)
Variance ratio test
```

Group	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf.	Interval]
1 male	14	33214.29	4905.904	18356.21	22615.72	43812.85
2 female	12	37250	10527.11	36466.98	14079.98	60420.02
combined	26	35076.92	5422.722	27650.57	23908.62	46245.23

```
① Ho: sd(1 male) = sd(2 female)
F(13,11) observed   = F_obs             =     0.253
F(13,11) lower tail = F_L   = F_obs     =     0.253
F(13,11) upper tail = F_U   = 1/F_obs   =     3.947
Ha: sd(1) < sd(2)        Ha: sd(1) != sd(2)        ② Ha: sd(1) > sd(2)
P < F_obs = 0.0108     P < F_L + P > F_U = 0.0252    P > F_obs = 0.9892
```

図 10　出力された sdtest の結果の例[12]

(2) 2 つのグループ間での平均値の差： ttest コマンド

母分散が異なるかどうかの確認を終えた後、t 検定に移る。t 検定をおこなうコマンドは、【ttest】である。

○コマンド：ttest（母分散が等しい場合の t 検定）

```
ttest 量的変数名 , by(質的変数名)
```

○コマンド：ttest（母分散が異なる場合の t 検定）

```
ttest 量的変数名 , by(質的変数名)
  unequal
（ttest 量的変数名 , by(質的変数名) une）
```

「ttest」の後ろに量的変数の名前を入力し、「by」の後ろには（2 つの値をとる）質的変数の名前を入力する[12]。母分散が等しい場合は「by(質的変数名)」までででよいが、母分散が異なる場合はさらにその後ろに「unequal」と入力することになる。

◇コマンド例（24）： ttest

```
ttest q02 , by(q01x1) unequal
```

コマンド例（24）は、「q02 について、q01x1 の値ごとに平均値を比較する t 検定をおこなえ。ただし、母分散は異なるものとする。」という意味である。このコマンドを実行すると、図 11 のような結果が出力される。

出力結果は、上半分が記述統計量、下半分が検定結果を示す、という構成になっている。記述統計量のうち、主として用いるのは①の部分であり、「Obs」が分析に用いたグループごとのサンプル数、「Mean」が平均値をあらわしている。

下半分の検定結果のうち、②の部分には帰無仮説（H_0）が示されている。ここでの帰無仮説は、「母集団において、2 つのグループ（q01x1 の値が「1」のグループと「2」のグループ）の平均値は等しい」であるため、「Mean(1 male) - Mean(2 female) = diff = 0」となっている（ここでは、Mean(1) と Mean(2)の差を「diff」と表現している）。

```
. ttest q02 ,by(q01x1)

Two-sample t test with equal variances
                  ①
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
  1 male |      14    33214.29    4905.904    18356.21    22615.72    43812.85
2 female |      12       37250    10527.11    36466.98    14079.98    60420.02
---------+--------------------------------------------------------------------
combined |      26    35076.92    5422.722    27650.57    23908.62    46245.23
---------+--------------------------------------------------------------------
    diff |           -4035.714    11071.37               -26885.91    18814.48
------------------------------------------------------------------------------
Degrees of freedom: 24
          ②
                    Ho: mean(1 male) - mean(2 female) = diff = 0

     Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
      t =  -0.3645      ③          t =  -0.3645                 t =  -0.3645
 P < t =   0.3593              P > |t| =   0.7187          P > t =   0.6407
```

図 11　出力された t 検定の結果の例[14]

いちばん下の 2 行には 3 種類の対立仮説（Ha）とそれに伴う検定結果（有意確率）が示されている。ここでの目的は、「2 つのグループ間で平均値が異なるかどうか」を検討することであるため、対立仮説が「diff != 0」である③の部分を見ることになる。③のうち、「P > ｜t｜」の部分に示されているのが有意確率である。

コマンド例（24）でおこなった検定における有意確率は「0.7187」であり、仮に有意水準を 5%（0.05）とする場合帰無仮説は棄却できない。したがって、検定の結果「母集団において、2 つのグループ間で平均値に差があるとは言えない（すなわち q02 と q01x1 の間に関連があるとは言えない）」という結論が得られる。

4.5 3 つ以上のグループ間での平均値の差——質的—量的変数間の関係 2： oneway コマンド

t 検定は 2 つのグループ間での平均値の差を検討する際に用いられるもので、3 つ以上のグループ間で差を検討する場合には適さない。3 つ以上の値をもつ質的変数を用いて平均値の差を検討する場合には、「一元配置の分散分析」をおこなうことになる。一元配置の分散分析をおこなうためのコマンドは、【oneway】である。

○コマンド： oneway

```
oneway 量的変数名 質的変数名, tabulate
(on 量的変数名 質的変数名, t)
```

◇コマンド例（25）： oneway

```
oneway q03c q04c, tabulate
```

「oneway」の後ろには平均値を比較したい量的変数、その後ろに質的変数の名前を入力する[15]。オプションとして「tabulate」と指定しておくと、一元配置の分散分析による検定の結果に加えて平均値等の記述統計量を示す表が出力されるようになる。

コマンド例（25）は、q04c（マスコミの信頼度）ごとに、q03c（大学以外での勉強時間）の平均値が異なっているかどうかの検定をおこなうためのコマンドである。具体的には、「q03c について、q04c の値ごとに平均値を比較する一元配置分散

```
. oneway q03c q04c, tabulate

                Summary of q03c
      q04c |      Mean   Std. Dev.       Freq.
-----------+------------------------------------
   1 agree |   1.9166667   2.3532698         12
2 mod. ag  |   .95833333    1.054392         12
3 mod. di  |   2.6666667   3.0550505          3
-----------+------------------------------------
     Total |   1.5740741   1.9743728         27

                          Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      9.53935185      2   4.76967593     1.25     0.3054
 Within groups        91.8125      24   3.82552083
------------------------------------------------------------------------
     Total          101.351852      26   3.89814815

Bartlett's test for equal variances:  chi2(2) =   6.9943  Prob>chi2 = 0.030
```

①

図 12　出力された一元配置分散分析の結果の例

分析をおこなえ。また、記述統計量の表もあわせて出力せよ。」という意味である。このコマンドを実行すると、図 12 のような結果が出力される。

出力結果のうち、上半分に示されているのが記述統計量である。「Mean」が平均値、「Std. Dev.」は標準偏差、「Freq.」は度数を示し、それぞれグループごとの値が算出されている。また、下半分には「母集団において、グループ間で平均値の差はない」という帰無仮説に基づいた検定の結果が算出されている。このうち、①の部分に示されているのが有意確率である。

コマンド例（25）でおこなった検定における有意確率は「0.3054」であり、仮に有意水準を 5%（0.05）とする場合帰無仮説は棄却できない。したがって、検定の結果「母集団において、グループ間で平均値に差があるとは言えない（すなわち q03c と q04c の間に関連があるとは言えない）」という結論が得られる。

5　分析に関する補足事項

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、分析を進めるにあたって、知っておいたほうがよいコマンドである。

5.1 分析の補助となるコマンド

（1）特定のサンプルの選択： if オプション

分析を進めていくなかでは、対象となるサンプルを限定して分析をおこないたい場合も出てくる（例えば、男性のみで年齢と収入の相関係数を算出したい場合など）。このような場合には、［if］オプションを用いる。

○コマンド： if オプション

分析コマンド　if 条件文

分析コマンドの後ろに［if］オプションをつけると、サンプルを限定した分析をおこなうことができる。具体的には、分析コマンドの後ろに「if」と入力し、さらにその後にサンプルを限定するための論理式を入力すればよい。

なお、サンプルを限定する［if］オプションは、他のオプションと異なり「if」の前にカンマを入れえる必要はない。他のオプションとともに［if］オプションを用いる場合は、メインのコマンドに続けて（カンマを入れずに）［if］オプションを入力し、その後にカンマを入力して他のオプションを入力すればよい。

◇コマンド例（26）： if オプションでサンプルを限定した pwcorr

```
pwcorr q04a q04b if q01x1 == 1, obs
  sig
```

コマンド例（26）は、q04a（世の人々への信頼度）と q04b（家族への信頼度）の相関係数を、男性のみで算出するためのコマンドである。具体的には、「q01x1 が『1』であるサンプルに限定して、04a と q04b の相関係数を算出せよ。ただし、相関係数にあわせてサンプル数、検定結果も算出せよ。」という意味である。

もちろんながら、［if］オプションは相関係数だけでなくほかの分析コマンド、データ変容コマンドでも使うことができる。

（2）カテゴリ別の分析： bysort コマンド

分析対象となるサンプルを限定する必要はないが、分析を何らかのグループごとにおこないたい場合もある（例えば、収入と生活満足度の相関係数を、男女で別々に算出したい場合など）。このような場合には、【bysort】コマンドを用いる。

○コマンド： bysort

```
bysort 変数名: 分析コマンド
```

◇コマンド例（27）： bysort

```
bysort q01x1: pwcorr q04a q04b, obs
  sig
```

分析コマンドの前に「bysort 変数名：」と入力しておくことで、指定された変数の値ごとに分析がおこなわれる（変数名の後のコロンを忘れないこと）。コマンド例（27）は、性別ごとに q04a（世の人々への信頼度）と q04b（家族への信頼度）の相関係数を算出するためのコマンドである。具体的には、「q01x1 の値ごとに、04a と q04b の相関係数を算出せよ。ただし、相関係数にあわせてサンプル数、検定結果も算出せよ。」という意味である。

［if］オプションと同様、【bysort】コマンドは相関係数以外の分析コマンドにも使うことができる。

（3）複数の変数に対する一括処理：for varlist コマンド

データ変容や分析をおこなう際、複数の変数に対して同一の処理がおこなわれることも少なくない（例えば、回答選択肢が共通である問に対して、同一の値ラベルを貼り付けるなど）。このようなときは、以下の 2 つのうちいずれかの方法で複数の変数を指定することによってコマンド入力の手間を省くことができる。

1 つは、変数と変数の間をハイフン（-）で結ぶ方法である。コマンド内で変数を指定する際「A-B」のように入力すると、「A から B まで」という意味になる。この場合、Variable ウィンドウ上で A から B までの間に含まれる変数がすべて指定されることになる。

◇コマンド例（28）： label（複数の変数に対する値ラベルフォーマットの適用）

```
label values q04a-q04d form2
```

コマンド例（28）は、複数の変数に対して同一の値ラベルフォーマットを適用させる場合の例である。具体的には、「q04a、q04b、q04c、q04d に対して、form2 という値ラベルフォーマットを適用せよ」という意味である。

2 つ目は、【for varlist】コマンドを用いる方法である。【for varlist】コマンドは、処理に用いたい変数リストをあらかじめ決めておくものである。

○コマンド： for varlist

```
for varlist 変数名 1 変数名 2: 変数部分を
  「X」とおいたコマンド
（for var 変数名 1 変数名 2: 変数部分を「X」
  とおいたコマンド）
```

コマンドを記述する前に、まず「for varlist」と入力し、その後ろに一括処理をおこないたい変数名を並列させるのである。変数名の並列が終わればコロンで区切り、その後に実際の処理に関するコマンドを入力する。このとき、コマンドのうち変数名を入力する部分に、変数名の代わりに大文字のエックス（X）を入力しておく。このようにしておくと、「for varlist」で並列された変数 1 つ 1 つが X の部分に順に代入され、それぞれに同じコマンドが実行されることになる。

◇コマンド例（29）： for varlist

```
for varlist q04a q04b q04d: label
  values X form3
```

コマンド例（29）は、q04a、q04b、q04d に対し

て共通の値ラベルを貼るためのコマンドである。具体的には、「q04a、q04b、q04d に対して、form3 という値ラベルフォーマットを適用せよ」という意味である。

ちなみに、【for varlist】コマンド内では、ハイフンを用いた複数変数の指定もできる。両者を活用して、コマンド入力の手間をなるべく省くとよいだろう。

（4）ヘルプ機能について

コマンドの書き方や、どのようなオプションがあるかなどについては、ヘルプ機能を用いると簡単に調べられる。Stata では、ヘルプを参照するための【help】コマンドというものが設けられている。

○コマンド： **help**

```
help コマンド名
```

◇コマンド例（30）： **help**

```
help oneway
```

「コマンド名」の部分には、ヘルプを見たいコマンド名を入力する。このコマンドを実行すると、Results ウィンドウにコマンドの説明文が出力される。コマンド例（30）は、「『oneway』コマンドのヘルプを出力せよ」という意味である。このコマンドを実行すると、図 13 のような結果が表示される。

なお、ヘルプの結果も他と同様に英語で出力される。読者の中には嫌がる方もおられるかと思うが、文章の中にはコマンドの使用例も含まれているので、これを見るだけでも参考になるはずである。コマンド入力で困った場合、ヘルプを参照する癖をつけておくとよいだろう[16]。

```
help for oneway                                                manual:  [R] oneway
                                                               dialog:  oneway

One-way analysis of variance

        oneway response_var factor_var [weight] [if exp] [in range] [, noanova
              nolabel missing wrap tabulate [no]means [no]standard [no]freq
              [no]obs bonferroni scheffe sidak ]

    by ... : may be used with oneway; see help by.

    aweights and fweights are allowed; see help weights.

Description

    oneway reports one-way analysis-of-variance (ANOVA) models and performs
    multiple-comparison tests.  If you wish to estimate more complicated ANOVA
    layouts, see help anova.

    See help loneway for an alternative to oneway with slightly different
    features.

Options

    noanova suppresses the display of the ANOVA table.

    nolabel causes the numeric codes to be displayed rather than the value
        labels in the ANOVA and multiple-comparison test tables.
```

図 13　【help】コマンド実行後（oneway について）

5.2 欠損値の除外

一般的な社会調査データでは、分析から除外すべき欠損値に対して独自の数値が割り当てられている。データセットを作成する際に、無回答には「9」や「99」、指定外には「7」や「97」、非該当には「8」や「98」といった数値を与え、有効回答と区別しているのである。ところが、Stata が欠損値として認識するのは空白（表示上はピリオド「.」）のみであり、99 や 97 といった数値は、欠損値としてではなく 1 つの回答データとして認識されることになる。このため、分析に際しては何らかのかたちで欠損値を除外する指定をおこなわなければならない。

欠損値の除外には、2 通りの方法がある。1 つは【recode】コマンドを使用しあらかじめ欠損値として定義しておく方法であり、もう 1 つは【if】コマンドを使用して分析から除外する方法である。順に説明しよう。

（1）recode コマンドを使用した欠損値の定義

【recode】コマンドを用いれば、欠損値の定義をおこなうことができる。具体的には、99 や 97 といった数値を、空白に置き換えるのである。

○コマンド： recode（欠損値の指定）

```
recode 変数名 (欠損値 =.), gen(新変数名)
```

◇コマンド例（31）： recode（欠損値の指定）

```
recode q04a (9=.), gen(nq04a)
```

コマンド例（31）では、もともとの欠損値の値である「9」を、Stata における欠損値である「.」に置き換えている。このとき、指定をおこなっていない他の値は元の変数と同じ状態が新しい変数にそのまま反映される。

ただし、単に置き換えるだけではもともとの変数の情報が失われてしまうため、欠損値の処理を施した新たな変数として保存しておくとよい。このようにしておくと、仮に欠損値の指定を解除したいような時に元の変数を使用できるからである。

（2）if オプションを使用した欠損値の除外

【recode】コマンドのようにあらかじめ欠損値を定義しておかなくても、5.1（1）で紹介した［if］オプションを用いれば欠損値を除外して分析をおこなうことができる。

○コマンド： if オプション（欠損値の除外）

```
コマンド  if 変数名!=欠損値
```

◇コマンド例（32）： if オプション（欠損値の除外）

```
summarize q04a if q04a != 9
```

コマンド例（32）のように、分析をおこなうサンプルを「9 以外」などと指定すれば、欠損値を含まない状態での分析ができる。

ここで紹介した 2 つの方法のうち、特にどちらが優れているとは言い難い。どちらを使用するかは分析者の判断に委ねたい。

5.3 分析結果の加工

2.1（2）で述べたとおり、分析の結果は Results ウィンドウに表示される。ただし、分析結果を実際に論文などで使用するためには、桁を調整したり罫線を引いたりして表を成形しなければならない。表を成形する作業は Results ウィンドウ上ではできないため、コピーしてエクセル等に貼り付けた上でおこなうことになる。

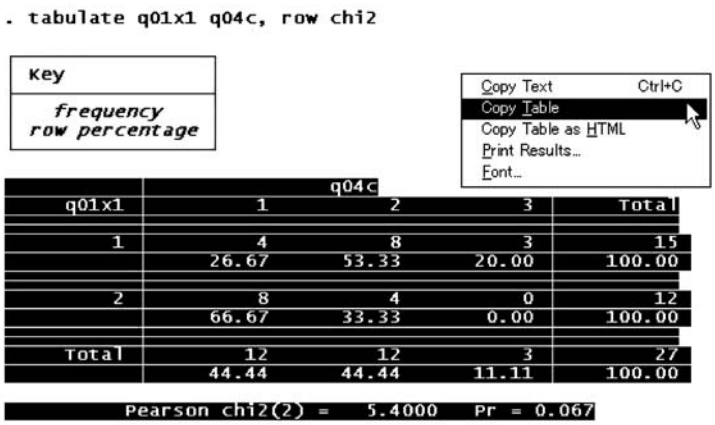


図 14　結果を選択してコピーする（Copy Table）

分析結果は「Ctrl＋C」ボタンなどでコピーしてエクセルに貼り付けることもできるが、この場合数値などがセルごとに分かれて入らず、成形しにくい状態になってしまう。成形しやすいかたちで分析結果をコピーするためには、次のような操作をおこなう。

(1)出力結果のうちコピーしたい部分をドラッグし、右クリックする。
(2) 出てきたダイアログのうち、「Copy Table」を選択する（図 14）。
(3) エクセルを立ちあげ、適当なところに貼り付ける。

以上の方法を取れば、結果の表に対応するかたちで行列がきれいにワークシートに収まる。その後、桁の調整や罫線を引くなどの作業をおこなうとよい。ただし、コピーする際に一度にたくさんの表をコピーしてしまうと、上記の方法でも貼り付けがうまくいかなく場合がある。少々面倒でも、表 1 つずつについてコピー&貼り付けをおこなうのがよいだろう。

6　おわりに

他の分析ソフトでも同じだが、Stata の操作に慣れるためにはとにかく数多くの分析をおこなってみることがいちばん手っ取り早い。読者の方々は、本稿での説明やヘルプ機能を参照しつつ様々な分析を試していただきたい。

なお、Stata ではこれ以外にも重回帰分析や主成分分析など多変量解析ももちろんおこなうことができる。多変量解析に関するコマンドについては、別稿で改めて解説していくことにしたい。

［注］

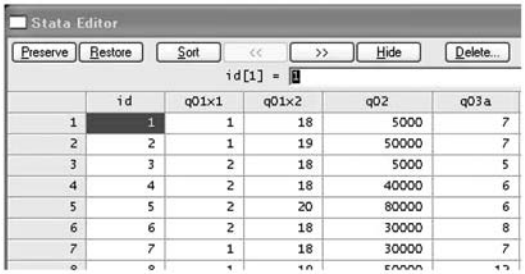
1） Stata にはいくつかヴァージョンがあるが、本稿での解説のために用いたのは Intercooled Stata 8 である。ただし、本稿で解説をおこなっている内容は、基本的に他のヴァージョンでも大きな違いはそれほどない。

2） Stata では、メニューバーから選択・指定してコマンドを実行することもできる。ただし、本稿では基本的に、こうしたメニューバーを用いた方法ではなく Command ウィンドウにコマンドを入力していく方法について解説していくことにする。というのも、筆者は以前、他の共著者らとシンタックスを用いた SPSS データ分析の方法についてまとめたことがあ

る。そこでは、メニューバーやダイアログを用いて分析する方法には、(1) 分析に対しておこなった細かい指定がわかりづらい点、(2) テキスト情報のみのシンタックスファイルと異なり、アウトプットファイルは容量が大きくなるので持ち運びにくい点、などがデメリットとして挙げられている（小林・雨森・山本 2009）。Stata においてメニューバーから選択・指定した分析をおこなう方法についても、これとほぼ同様のことが当てはまるからである。

3） 入力に使えるのは半角英数字のみである。

4） メニューバーの「Data」から「Data editor」を選択すれば、データを表示する「Data editor」ウィンドウが表示される（下図）。Data editor ウィンドウでは数値や文字を入力することもできる。データ自体に何らかの修正を加えたい場合は、この Data editor ウィンドウ上で作業をおこなうことになる。



	id	q01x1	q01x2	q02	q03a
1	1	1	18	5000	7
2	2	1	19	50000	7
3	3	2	18	5000	5
4	4	2	18	40000	6
5	5	2	20	80000	6
6	6	2	18	30000	8
7	7	1	18	30000	7

5） 「変数名」の部分に複数の変数を併記し、一括して変数ラベルの貼り付けをおこなうことはできない。【label variables】コマンドは、「variables」の後の文字列は変数名、その後ろがラベル、という具合に文字列が入力されている位置によって変数名とラベルの認識がなされるからである。複数の変数に対し一括して処理をおこなう方法については、5.1 (3) を参照のこと。

6） 「変数名」の部分に複数の変数を併記し、一括して値ラベルフォーマットの適用をおこなうことはできない。【label values】コマンドでは、「values」の後の文字列は変数名、その後ろが値ラベルフォーマット名、という具合に文字列が入力されている位置によって変数名とラベルフォーマット名の認識がなされるからである。複数の変数に対し一括して処理をおこなう方法については、5.1 (3) を参照のこと。

7） 【recode】コマンドでは変数名を指定する部分に複数の変数を入れることもできる。この場合、後ろにつける［gen］オプションのカッコ内にも新変数名を並列することになる。この方法を用いれば、複数の変数に対して値の再割り当てを一括しておこなうことができる。

8） 「変数名」の部分に複数の変数を併記し、一括して度数分布を出すことはできない。なぜなら、「変数名」の部分に複数の変数が入力されている場合、【tabulate】コマンドは「併記された変数からなるクロス表を作成せよ」という意味になるからである。複数の変数に対し一括して処理をおこなう方法については、5.1 (3) を参照のこと。

9） 「変数 1」あるいは「変数 2」の部分に複数の変数を併記し、複数のクロス表を一括して作成することはできない。複数の変数に対し一括して処理をおこなう方法については、5.1 (3) を参照のこと。

10） 正確に言えば、図 8 のクロス表には度数が 0 のセルがある上に期待度数 5 未満のセルが 2 つあるため、検定をおこなうのに適切な表ではない。ここでは、あくまでもコマンドと分析結果の例として用いている。

11） 「量的変数名」「質的変数名」の部分に複数の変数を併記し、一括して検定をおこなうことはできない。複数の変数に対し一括して分析をおこなう方法については、5.1 (3) を参照のこと。

12） 図 4 のデータでは、q02 が欠損値（999999）であるサンプルが 1 つ含まれている。図 10 の分析ではこのサンプルを欠損値として除外しているため、サンプル数が「26」になっている。なお、欠損値を除外する方法については 5.2 を参照のこと。

13） 「量的変数名」「質的変数名」の部分に複数の変数を併記し、一括して検定をおこなうことはできない。複数の変数に対し一括して分析をおこなう方法については、5.1 (3) を参照のこと。

14） 図 10 と同様に、図 11 の分析では欠損値（1 サンプル）を除外しているため、サンプル数が「26」になっている。

15） 「量的変数名」「質的変数名」の部分に複数の変数を併記し、一括して検定をおこなうことはできない。複数の変数に対し一括して処理をおこなう方法については、5.1 (3) を参照のこと。

16） 本稿ではコマンドの説明をおこなう際、コマンド名の部分をすみつきカッコ（【 】）でくくっている。それぞれのヘルプを見る際には、すみつきカッコ内の文字を【help】コマンドで入力すればよい。

［文献］

東尚弘・林野泰明・杉岡隆，2008『臨床研究のための Stata マニュアル』NPO 法人健康医療評価研究機構.
Cameron, A.C. & Trivedi, P. K.，2008，*Microeconometrics Using Stata*, Stata Corp.
石黒格編著，2008『Stata による社会調査データの分析――入門から応用まで』北大路書房.
小林久高・雨森聡・山本圭三，2009「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」『同志社社会学研究』第 13 号，45-76.
松浦寿幸，2010『Stata によるデータ分析入門』東京図書.

［参考になるウェブページ］

別所俊一郎「ネコでもわかる Stata 入門」（http://www.econ.hit-u.ac.jp/~bessho/paper/02/stata1.pdf#search=% 27stata%27）
松浦寿幸・佐々木明果・渡辺善次「経済分析のための Stata 入門」（http://park1.wakwak.com/~mt _tosiyuki/stata-manual.pdf）