

Excel VBA を使用した入力データのチェックプログラム

——Data Checker for Social Research——

山本 圭三・金 政芸・猿渡 壮

YAMAMOTO Keizo, KIM Jeongwoon, SARUWATARI Takeshi

1 はじめに

調査票を用いた計量的調査において、調査票が回収された後には回答データをパソコンに入力する作業をおこなうことになる。データを入力する作業自体は単純なものであるが、このときキーの打ち間違いなどの入力ミスはどうしても起こってしまう。この作業でミスが起こってしまうとせっかく得られた回答が台無しになってしまうため、ミス発生の予防、あるいはミスが起こった場合の修正を可能な限りおこなわなければならない。

入力ミスを防ぐ手立てはいくつか考えられるが、その1つとしてデータ入力を2回おこなう方法がある（小林・雨森・山本 2008; 小林・山本・金・猿渡 2011）。このアイデアは、簡単に言えば、得られた回答を2度別々に入力し、1回目の入力と2回目の入力が一致しているかどうかでミスの発生を判断する、というものである。1回入力するだけではミスに気づきにくい、2回同じものを入力しそれらを照合することで、見落とししていたミスを検出できるのである。ただし、ミスが起こっている箇所を発見しそれを修正する作業をすべて手作業でおこなおうとすると、かなり時間がかかってしまう。入力する数が多くなればなるほど作業はますます煩雑になっていくため、作業の最中に新たなミスが生じてしまうこともある。

こうした点をふまえ、筆者らは入力時のミスを発見し修正するためのデータチェックプログラム「Data Checker for Social Research 1.0（以下、DCSR）」を作成した。このプログラムは、データ

入力作業がエクセルでおこなわれることが多いことを鑑み、エクセルのVBA（Visual Basic for Applications）を用いて作成されたものである。本稿では、このDCSRの紹介とともに、DCSRを用いて入力済みのデータをチェックする方法について解説していく。

本稿は、以下のような構成になっている。まず2で、DCSRを用いた作業の前段階として、入力時のエラーを防ぐ基本的なアイデアについて簡単に述べる。次に、3でDCSRの準備から操作方法までを、具体的な例とともに紹介する。4で注意点を述べ、末尾にDCSRのプログラムを記載する。入力時のエラーを防ぐ方法について熟知しておられる読者は、3から読み始めてもらえればよい。なお、DCSRはエクセルのいわゆるマクロにあたるものであるが、本稿ではマクロについての一般的な説明は省略する。マクロについて基礎から知りたい場合は、他の文献にあたっていただきたい。

2 入力時エラーを防ぐためのアイデア

2.1 データの2回入力

まず、データの2回入力の方法についてまとめておく。以前に筆者らが用いた例を参考にしつつ、基本的なアイデアを紹介しよう（小林・山本・金・猿渡 2011）。

まず、全回答者分のデータをエクセルの1枚の

シートに入力する。すべてのデータを入力し終わったら、もう一度全回答者分のデータを別のシートに入力する。図1は1度目と2度目の入力結果を示したものである（上：1度目の入力（Sheet1）、下：2度目の入力（Sheet2））。

図1 データの2回入力

	A	B	C	D	E	F
1	ID	q01x1	q01x2	q02x1	q02x2	q02x3
2	1	2	18	社会	社会福祉	9999
3	2	2	20	社会	社会	2009
4	3	2	20	社会	メディア	2008
5	4	2	20	社会	社会	2007
6	5	1	21	社会	産業関係	2008
7	6	2	20	社会	社会	2009
8	7	2	20	社会	社会	2008
9	8	1	20	社	社	2010

	A	B	C	D	E	F
1	ID	q01x1	q01x2	q02x1	q02x2	q02x3
2	1	2	18	社会	社会福祉	9999
3	2	2	20	社会	社会	2009
4	3	2	20	社会	メディア	2008
5	4	2	22	社会	社会	2009
6	5	1	21	社会	産業関係	2008
7	6	2	20	社会	社会	2009
8	7	2	20	社会	社会	2008
9	8	1	20	社	社	2010

2度目の入力を終わったら、別のシート（Sheet3）において、2つのシートの同じセル番地に同じ値が入力されているかを照合する。入力された値が一致していなければ、入力時にミスが起きていることになる。たとえばSheet1とSheet2のB2に同じ値が入力されているかを照合するためには、Sheet3のB2に「=IF(Sheet1!B2=Sheet2!B2, "OK", "NG")」という式を入力すればよい。この式をコピーシート全体に貼り付ければ、入力ミスがどこで生じているのかわかる。

図2は、2回の入力データの照合をおこなった結果の例である。図から、C5とF5のセルで入力ミスが起きていることがわかる。Sheet3でミスが起きている箇所がわかれば、Sheet1もしくはSheet2に入力されている値を修正すればよい。

ミスが修正されればSheet3での照合結果も自動

的に変更され、『OK』と表記されるようになる。データ全体にわたってSheet1とSheet2の値がすべて一致すれば、入力ミスの修正作業は完了となる。

図2 入力された値の照合

	A	B	C	D	E	F
1	ID	q01x1	q01x2	q02x1	q02x2	q02x3
2	1	OK	OK	OK	OK	OK
3	2	OK	OK	OK	OK	OK
4	3	OK	OK	OK	OK	OK
5	4	OK	NG	OK	OK	NG
6	5	OK	OK	OK	OK	OK
7	6	OK	OK	OK	OK	OK
8	7	OK	OK	OK	OK	OK
9	8	OK	OK	OK	OK	OK

以上のように、データを2回入力することで、かなりの割合で単純な入力ミスを防ぐことができる。ただし、このチェック方法では、2回とも入力ミスをした場合は検出できない。それゆえ、2回入力をおこなう際には、1人で2度の入力をおこなうよりも、2人1組でそれぞれ1回ずつ入力するほうが望ましい（小林・雨森・山本 2008；小林・山本・金・猿渡 2011）。

2.2 DCSRによる作業の簡略化

入力された値を修正する作業は、3つのシート間を行き来しながら進めることになる。Sheet3でミスが発生しているセルを発見したら、調査票を見ながらSheet1およびSheet2で当該セルの値を確認し、ミスが起こっているほうの値を修正するのである。このような作業は、文章で書くと簡単に見えるが、実際に行なってみるとかなり煩雑で時間もかかる。このため、時には正しい値を誤って変更してしまったり、ミスを見逃してしまったりすることもある。

こうした煩雑な操作をより簡単に、なおかつ正確におこなうためのツールが、本稿で紹介するDCSRである。DCSRは、具体的に以下のような作業を自動でおこなうことができる。

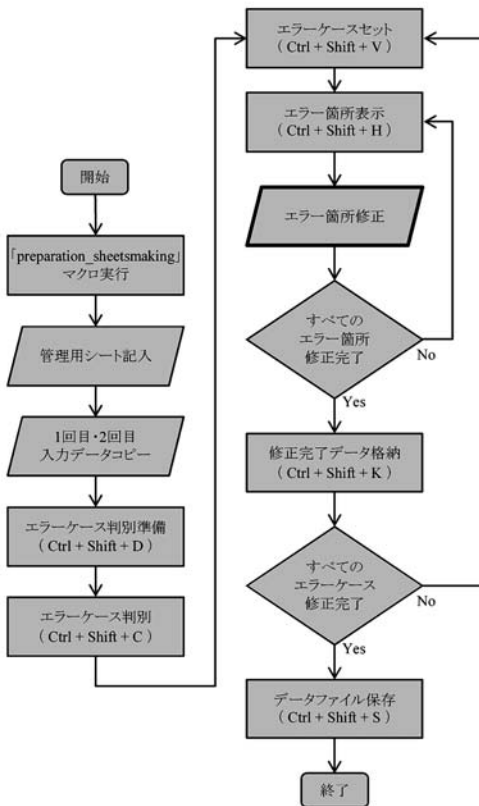
- ・入力ミス発生ケースの判別
- ・入力ミスの発生件数のカウント
- ・入力ミス発生箇所の自動表示
- ・修正完了済みデータの作成

以下では、DCSR の具体的な使い方について解説しよう。

3 DCSR の操作方法

先にも述べたように、DCSR はエクセルの VBA を用いて作成したデータチェックプログラムである。データチェックは、以下のような流れでおこなわれる (図 3)。

図 3 DCSR における作業の流れ



以下で、DCSR の具体的な操作方法について解

説しよう。なお、DCSR は基本的にエクセル起動時のデフォルト状態を想定して作成されたものである。このため、以下で説明する操作は、あらかじめデータを入力したブックとは別の新しいブックでおこなっていただきたい。

3.1 準備

DCSR を用いるにあたっては、いくつかの準備が必要になる。この作業から説明しよう。

(1) Visual Basic Editor の起動とプログラム入力

DCSR を使用するためには、まずエクセルに DCSR のプログラムを入力しなければならない。プログラムの入力には「Visual Basic Editor (以下、VBE)」においておこなうが、その方法は次のとおりである。

エクセルを起動すると通常は図 4 のような画面が立ち上がる。VBE やマクロの操作をおこなうボタンは「開発」タブ内にあるが、通常はこの「開発」タブは表示されていない。表示するためには、以下の操作をおこなう¹⁾。

- (1) リボンの「ファイル」タブから「オプション」を選択する。
- (2) 開いたダイアログの左側のメニューから「リボンのユーザー設定」を選択する。
- (3) 2つの大きな表のうち、右側の「リボンのユーザー設定」の表にある「開発」のチェックボックスをオンにする (図 5)。
- (4) 「OK」ボタンを押す。

リボンの「開発」タブを選択し、左端にある「Visual Basic」を選択すると (図 6)、VBE の編集画面が開く。開いた画面では右側の大きなウィンドウがグレーになっているが、この状態でメニューバーから「挿入」→「標準モジュール」を選

図 4 エクセルの通常画面（「開発」タブ非表示）

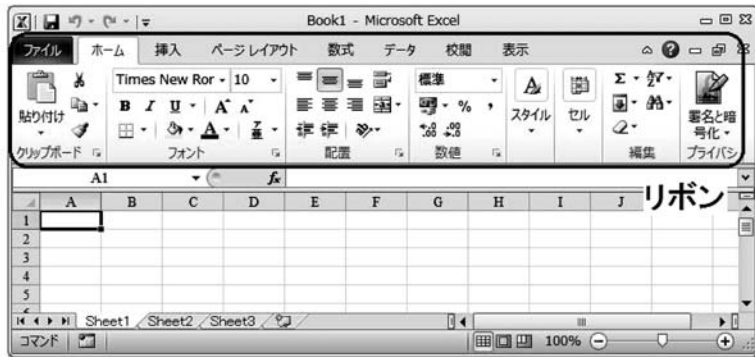


図 5 エクセルのオプション画面

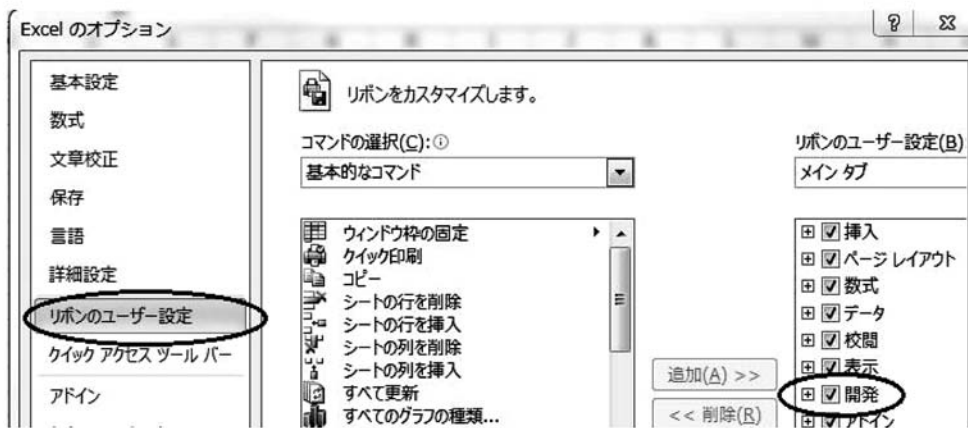


図 6 「開発」タブ

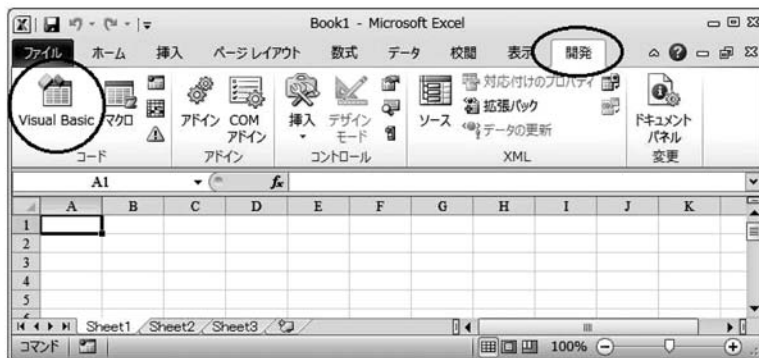
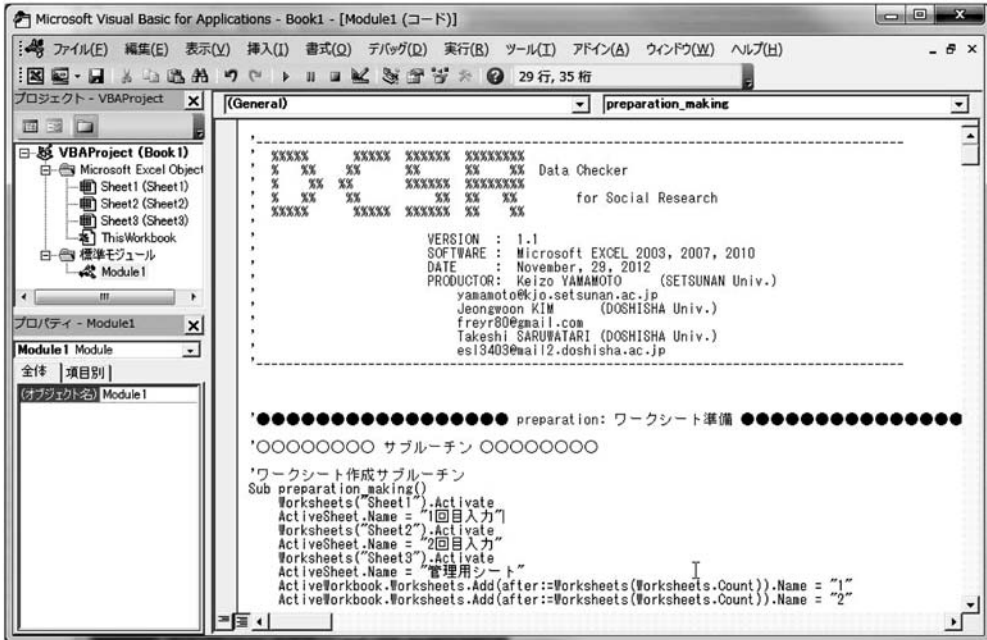


図 7 プログラム入力画面



択すると、右側に白色のウィンドウが開く。DCSRのプログラムは、この白色のウィンドウ内に入力することになる (図 7) 2)。

プログラムを入力し終えたら、メニューバーの「ファイル」から「終了して Microsoft Excel に戻る」を選択し、VBE の編集画面を閉じる。VBE の編集画面を閉じた時点で、入力されたプログラムは実行可能になる。

(2) 一部のプログラムの実行

DCSR には、作業の準備をおこなうための「preparation_sheetsmaking」という名前のプログラム (マクロ) が用意されている。実際に DCSR を操作するためには、このプログラムを先に実行しておかなければならない。

プログラムの実行は、以下の要領でおこなう。リボンの「開発」タブをクリックし、左側にある「マクロ」を選択すると、実行するプログラムを

図 8 「preparation_sheetsmaking」の実行



選択するダイアログが開く (図 8)。マクロ名の中から「preparation_sheetsmaking」を選択し「実行」をクリックすると、プログラムが実行される。

プログラムが実行されると、Sheet1 が「管理用シート」、Sheet2 が「1 回目入力」、Sheet3 が「2

図 9 管理用シート

	A	B	C	D	E
1					
2	管理用メニュー	黄色の部分に情報を入力してください。		【操作メニュー】	【ショートカットキー】
3				エラーケース判別準備	Ctrl + Shift + D
4				エラーケース判別	Ctrl + Shift + C
5	作業担当者1氏名→ 社会太郎			エラーケースセット	Ctrl + Shift + V
6	作業担当者2氏名→ 調査次郎			エラー箇所表示	Ctrl + Shift + H
7				修正完了データ格納	Ctrl + Shift + K
8	先頭ケース番号→		1	データファイル保存	Ctrl + Shift + S
9	末尾ケース番号→		300		
10					

回目入力」という名前に変わる。また、「1」「2」「判別用シート」「要修正_1回目入力」「要修正_2回目入力」「修正用シート」「チェック完了データ」の各シートが追加される。この状態になれば、DCSRの準備は完了である。

(3) 管理用シートの記入

社会調査のデータは、変数やケースの数が膨大であるため、たいてい数人から数十人が分担して入力作業をおこない、それを後で1つのデータファイルに統合する。「管理用シート」は、データファイルを統合するときミスが起こらないように、作業担当者名と担当したケース番号を記入するためのシートである。

「preparation_sheetsmaking」を実行すると、新しく作成された「管理用シート」が開かれた状態になる。この段階で、1回目入力をおこなった作業者の氏名を「作業担当者1氏名」に、2回目入力をおこなった作業者の氏名を「作業担当者2氏名」に記入しよう。また、担当したデータの先頭ケース番号と末尾ケース番号も記入しておこう(図9)。

(4) データのコピー

次に、エラーチェックをおこなうデータを、DCSRのプログラムが設定されているブックにコピーする。1回目の入力データは「1回目入力」シートに、2回目の入力データは「2回目入力」シ

トにそれぞれコピーして貼り付ける。データをコピーする際には、データ全体が選択されているかを確認しながらコピーするように注意しよう。

なお、ここでコピーし貼り付けるデータには、ケースの通し番号をあらわす「ID」を先頭の列(A列)に必ず入れておく必要がある。IDが入力されていない場合、以降のチェック作業がきちんと進まないからである。コピー元に通し番号が入力されていない場合は、事前に入力しておこう。

(5) ショートカットキーを使ったプログラムの実行について

VBEに入力されたプログラムは、先のようにダイアログから選択することで実行できるが、ショートカットキーを使って実行することも可能である。DCSRでも、効率よく作業ができるように、操作に関わるプログラムはショートカットキーで実行できるようにしている。以下では、ショートカットキーを使った方法で説明する。

3.2 エラーケースの判別——チェック第1段階

入力したケースのなかには、入力ミスが起こっているケースもあれば、起こっていないケースもある。ここでは、入力ミスが起こっているケースのことを「エラーケース」と呼ぶことにしよう。修正が必要なのはエラーケースのみであるため、

実際の修正をおこなう前に、エラーケースとそれ以外を判別し、エラーケースだけを取り出しておくほうが効率的である。このため、DCSR では具体的なエラーケースの修正をおこなう前に、エラーケースの判別作業をおこなうことになっている。

エラーケースを判別するためには、(1) 判別作業準備プログラム、(2) エラーケース判別プログラムの2つを実行することになる。

(1) エラーケース判別の準備

「1回目入力」および「2回目入力」シートにデータが入力されている状態で「Ctrl + Shift + D」キーを押すと、エラーケース判別の準備がおこなわれる。具体的には、「判別用シート」に変数名などの情報が入力されるとともに、「1」シートに「1回目入力」のデータが、「2」シートに「2回目入力」のデータがコピーされる。作業が終わると図10のようなダイアログが現れ、判別作業のための準備を完了したことを知らせる³⁾。

図 10 判別作業の準備が完了



なお、この時点で「1回目入力」「2回目入力」シートは編集ができなくなっている。これは、入力した元データを自動的に保護しておくための設定である。チェックや修正作業のなかで何か問題が起こった場合は、保護されているこの元のデータを用いて最初から作業をおこなえばよい。

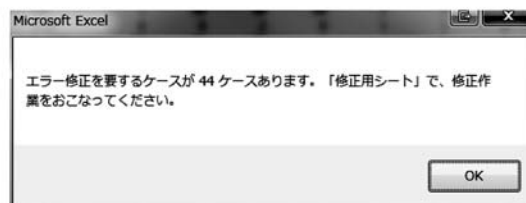
(2) 判別作業

準備が終われば、実際の判別作業に移る。エラーケース判別プログラムを実行するショートカットキーは「Ctrl + Shift + C」キーである。

キーを押すと、それぞれのケースについて、1回目に入力した値と2回目に入力した値が一致しているかどうかチェックされていく。1つでも一致しない値があれば、当該ケースの「1回目入力」の値がすべて「要修正_1回目入力」にコピーされ、「2回目入力」の値が「要修正_2回目入力」にコピーされる。すべての値が一致しているケース、すなわち入力エラーのないケースは「チェック完了データ」に格納される。

ケースの判別作業が終わると、エラーケースの数を知らせる図11のようなダイアログが現れる。「OK」をクリックすると判別作業は完了である。

図 11 エラーケース数の表示



3.3 エラーの修正作業——チェック第2段階

(1) 修正用シートの説明

実際にデータを修正する作業は「修正用シート」でおこなう。エラーケースの判別作業が完了した後に「Ctrl + Shift + V」キーを押すと、エラーケースのうち1ケース分が「修正用シート」にセットされる。図12が「Ctrl + Shift + V」キーを押した後の「修正用シート」の画面である。

図 12 修正用シート

	A	B	C	D	E	F	G	H
1	ID ↓							
2		2						
3		q01 x1	q01 x2	q02 x1	q02 x2	q02 x3	q03	q04
4	1回目入力値→	2	20	社会	社会福祉	2009	1	15000
5	2回目入力値→	2	20	社会	社会	2009	1	15000
6	一致:0、不一致:1	0	0	0	1	0	0	0
7								
8	修正後の値→	2	20	社会	要修正	2009	1	15000
9								
10	入力時エラー数 ↓							
11		6						
12								
13	修正が必要なセル数 ↓							
14		6						

「ID」の部分には、修正が必要なケースの番号が示されている。「1回目入力値」「2回目入力値」と書かれた行には、1回目と2回目に入力された値がそのままコピーされている。その下にある「一致：0、不一致：1」と書かれた行には、1回目に入力した値と2回目に入力した値が一致しているかどうか記されており、一致していれば0、一致していなければ1が黄色で表示されている。つまり、この行に1が表示されている場合、その部分は修正が必要だということになる。

「入力時エラー数」の部分には、修正前の段階でこのケースにエラーがいくつあるかが示されている。例えば、図12ではIDが2のケースで計6個のエラーが生じているとわかる。

(2) エラーの修正

図12で「一致：0、不一致：1」の行を見ていくと、E6で入力値に不一致があることがわかる（1回目の入力値は「社会福祉」、2回目の入力値は「社会」）。不一致が見つかったら、調査票やコードブックを確かめながら、正しい値を入力する。このとき、正しい値は「修正後の値」の行に入力する（図13）。不一致が生じている場合はここに「要

図 13 修正後の値を入力する場所

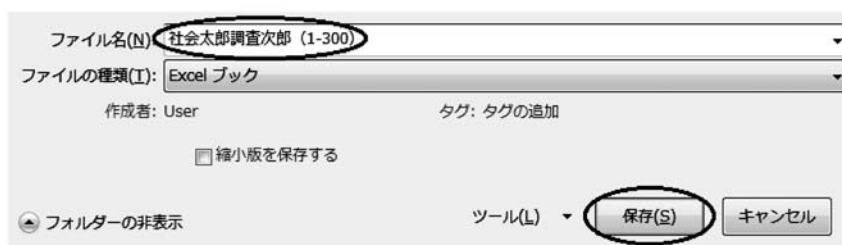
1回目入力値→	2	20	社会	社会福祉	
2回目入力値→	2	20	社会	社会	
一致:0、不一致:1	0	0	0	1	
修正後の値→	2	20	社会	要修正	

修正」という文字が赤色で表示されているので、この文字を消して正しい値を入力すればいい。1つの修正が完了したら、再度修正が必要な箇所を探し、同様の方法で修正をおこなっていく。

ところで、修正箇所を探すためには画面を右にスクロールさせながら、修正が必要なセルを1つ1つ見つけ出していかなければならない。こうした作業を簡略化したい場合は、「Ctrl + Shift + H」を押せばいい。キーを押すと、自動的に次の修正箇所までカーソルが移動する。「Ctrl + Shift + H」で修正箇所を探し、正しい値を入力し、再度「Ctrl + Shift + H」を押して修正箇所を探すという作業を繰り返すことで、修正に要する時間は大きく短縮できる。

このような方法で正しい値を入力していくと、「修正が必要なセル数」(A14)の部分の数値が1つずつ減っていく。1つのエラーケースの修正作業は、「修正が必要なセル数」が0になるまでおこなう。

図 14 保存するファイルの名前を入力



(3) 修正済みデータの格納

1 つのケースの修正作業が終わったら、「Ctrl + Shift + K」を押す。これでデータは「チェック完了データ」シートに格納される。

格納が済んだら、次のエラーケースの修正にとりかかる。「Ctrl + Shift + V」を押し、次のエラーケースを「修正用シート」にセットし、同様の方法でエラーの修正をおこなっていけばいい。

3.4 エラー修正後のデータを保存

すべてのケースの修正が完了した状態で「Ctrl + Shift + V」キーを押すと、「修正を要するケースがありません」というメッセージが表示される。これはすべてのエラーケースの修正が完了したことを意味している。すべての修正が完了したら、修正後のデータを保存しよう。データの保存方法は以下の通りである。

まず、「管理用シート」に作業担当者名、先頭ケース番号、末尾ケース番号が入力されていることを確認した上で、「Ctrl+ Shift+S」を押す。すると、名前をつけてファイルを保存する画面があらわれる(図 14)。保存するファイルの名前は「作業担当者名 (担当範囲)」という形で自動的に設定されるので、適当な保存場所を選択した上で「保存」ボタンを押せばいい。

チェックを複数の作業グループでおこなった場

合、チェック済みのファイルが作業グループのぶんだけできる。各作業グループで作成されたチェック完了データを1つにまとめれば、入力エラー修正済みデータファイルのできあがりとなる。

4 操作上の注意点

DCSR を操作する上では、いくつか注意しなければならない点がある。

4.1 データの先頭列には「ID」を入れておく

DCSR は、チェックすべきデータにケースの通し番号 (ID) がふられている状態を前提で作成されている。このため、3.1 (4) でも述べたが、「1 回目入力」「2 回目入力」シートに貼り付けるデータには、必ず A 列に「ID」を入れておかなければならない。A 列に入力されているのが ID ではない場合、チェック作業の段階で何らかの不具合が生じてしまうので注意しておこう。

4.2 作業の中断と再開について

エラーケースの数が多ければ、1 回の作業ですべてを修正することが難しい場合もある。だが、修正作業が途中のままで「Ctrl+ Shift+ S」を押してデータを保存しようとしても、警告メッセージ

が出て保存ができない。「Ctrl+ Shift+ S」キーは、あくまでもチェックが完了した際に使用するものとして設定されているからである。

DCSR プログラムが入力されているファイルも基本的にはエクセルファイルであるため、リボンの「ファイル」を選択し、「名前を付けて保存」することも可能である。作業を途中で中断したい場合はこのようにしてファイルを保存すればよい。この時、「標準モジュール」に記述した DCSR のプログラムも同時に保存される。

作業を再開する場合も、通常のエクセルファイルと同様に当該ファイルのアイコンをダブルクリックすればよい。ただし、エクセルの設定によっては作業を再開した時にショートカットキーが無効になっている場合がある。このような場合は、「開発」タブにある「マクロ」ボタンを押し、ダイアログのなかから「preparation_shortcutkey」を選択し「実行」をクリックするとよい。

4.3 シートの保護解除について

DCSR では、実際の修正作業に直接関連しないセルやシートは編集作業ができないよう保護設定がなされている。たとえば、もともと入力したデータは「1回目入力」「2回目入力」シートに入っているが、これらのシートは判別作業をおこなった時点でシートごと保護され編集ができなくなっている(3.2 (1)を参照)。だが、何らかのミスが生じたときなどは、修正をほどこす前のデータに立ち戻って作業をおこなわなければならない。このような場合には、シートの保護を解除する必要がある。

DCSR には、シートの保護解除を実行する「stepx_control_allsheetunprotect」というマクロも準備してある。「stepx_control_allsheetunprotect」を実行すると、すべてのシートの保護設定が解除さ

れる。保護されているシートに何らかの手を加えたい場合は、これを実行しておくといよい。

4.4 マクロのエラーが出た場合の対処

データの修正作業をおこなう中では、不意に DCSR のプログラム自体に何らかの変更を加える操作をしてしまうことがある。この場合、ショートカットキーを押しても警告メッセージが出てうまく動かなくなってしまうこともある(図 15)。

図 15 マクロのエラーを示す警告



このような場合は作業を中断し、正しい DCSR プログラムを貼り付けなおして作業を再開することになる。具体的には、作業の進み具合に応じて次のように対処すればよい。

エラーケースの判別をしている最中に警告メッセージが出てきた場合は、「終了」あるいは「OK」を押したのち、ファイルを保存せず作業を中止する。そして、改めて新規のワークシートを立ち上げ、3.1 の作業から順におこなっていけばよい。

エラー修正作業に入り、すでに何件か修正をお

こなっている際にエラーが生じた場合は、少々面倒だが以下の要領で対処するのがよい。

- [1] 「終了」あるいは「OK」を押し、画面に戻る。
- [2] 「名前を付けて保存」で、適当な名前を付けて保存する（「チェック完了データ」にあるチェック済みデータをとりあえず保存）。
- [3] 新規ワークシートを立ち上げ、3.1の(1)～(3)までの作業をおこなう。
- [4] 入力したデータのうちチェックが必要なデータのみを「1回目入力」「2回目入力」のシートに貼り付けをする（同じ作業の繰り返しを避けるため）。
- [5] チェック作業完了後、3.4のようにファイルを保存する（エラー発生前作業分と発生後作業分の2ファイルができる）。
- [6] 2ファイルに分かれている「チェック完了データ」の内容を別ファイルにコピー&貼り付けをし、名前を付けて保存する。

以上の作業を経ることで、入力エラー修正済みデータファイルができあがる。

5 おわりに

最後に、その他の注意点について述べておこう。

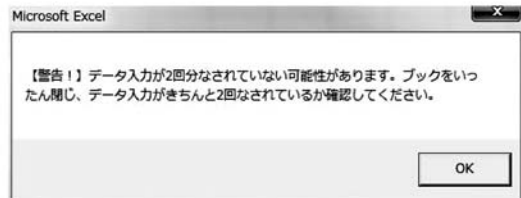
DCSR では、修正後の値を入力する際に間違っただけを入力してしまった場合、そのミスを見出すことはできない。そのため、修正値を入力する際には、調査票やコードブックをよく確認し、慎重におこなっていただきたい。また DCSR では、1回目と2回目に同じ間違っただけが入力されている場合も、ミスを見出すことはできない。したがって、DCSRでのチェックを終えた後は、単純集計を見てこうしたミスがないかを確認することが必要である。

入力エラーの修正を終えたら、次はロジカルエ

ラーの修正が待っている。ロジカルエラーの修正に関しては、別稿（小林・雨森・山本 2008, 小林・山本・金・猿渡 2011）を参照されたい。

【注】

- 1) ここで説明している手順は、Excel のバージョンが 2010 の場合を想定している。2007 の場合は、以下のとおりの操作をおこなうことになる。
 - (1) 画面左上にある Office ボタンを押し、「Excel のオプション」を選択する。
 - (2) 開いたダイアログの左側のメニューから「基本設定」を選択する。
 - (3) 「Excel の使用に関する基本オプション」の部分にある「開発タブをリボンに表示する」にチェックを入れる。
 - (4) 「OK」ボタンを押す。
 また、2003 以前のバージョンの場合そもそもリボンが存在しない。このバージョンのときは、ツールバーから「ツール」→「マクロ」→「Visual Basic Editor」と選択すれば、VBE の編集画面が開く。
- 2) [付記] で述べているが、DCSR のプログラムは筆者の1人が運営するホームページ上で公開されている。ここでは便宜上「プログラムを入力する」と表現しているが、実際にはホームページからプログラムをダウンロードし、その内容をコピー&貼り付けをすればよい。
- 3) DCSR は、2回分の入力がかちとなされていることを前提とするプログラムである。このため、「1回目入力」シートと「2回目入力」シートに同じ数のデータが入力されていない、もしくは1回目と2回目で異なる ID のデータが入力されている場合、エラーケース判別の時点で下図のような警告がなされる。



この警告が出た際は DCSR での作業をいったん中断し、入力済みのデータに立ち戻って確認しよう。すべてのケースについて2回分きちんと入力されていることが確認できれば、再度 3.1 (1) からの手順で作業を進めればよい。

[付記]

DCSR のプログラムは、下記ウェブページにて公開しております。希望される方は、ウェブページからお申し込みください。

【DCSR 公開用ウェブページ】

<http://yamamoke.net/reports/research/dcsr>

また、プログラムのバグを発見された場合は、下記連絡先にお伝えいただければ幸いです。

山本 圭三: yamamoto@kjo.setsunan.ac.jp
金 政芸: freyr80@gmail.com
猿渡 壮: tsrwtr@gmail.com

[文献]

小林久高・雨森聡・山本圭三, 2008「社会調査データの入力とチェックの方法」『同志社社会学研究』12: 41-50.
小林久高・山本圭三・金政芸・猿渡壮, 2011「社会調査のためのエクセル利用法」『同志社社会学研究』15, 77-96.

[資料] Data Checker for Social Research (DCSR) プログラム

```
-----  
' *****  
' % % % % % % % % % % % % % % Data Checker  
' % % % % % % % % % % % % % %  
' % % % % % % % % % % % % % % for Social Research  
' *****  
'  
' VERSION : 1.1  
' SOFTWARE : Microsoft EXCEL 2003, 2007, 2010  
' DATE : February, 02, 2013  
' PRODUCOR: Keizo YAMAMOTO (SETSUNAN Univ.)  
' yamamoto@kjo.setsunan.ac.jp  
' Jeongwoon KIM (DOSHISHA Univ.)  
' freyr80@gmail.com  
' Takeshi SARUWATARI (DOSHISHA Univ.)  
' tsrwtr@gmail.com  
-----
```

' ●●●●●●●●●●●●●●●● preparation: ワークシート準備 ●●●●●●●●●●●●●●●●

' ○○○○○○○○ サブルーチン ○○○○○○○○

' ワークシート作成サブルーチン

```
Sub preparation_making()  
  With ActiveWorkbook  
    .Worksheets("Sheet1").Name = "管理用シート"  
    .Worksheets("Sheet2").Name = "1 回目入力"  
    .Worksheets("Sheet3").Name = "2 回目入力"  
    .Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "1"  
    .Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "2"  
    .Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "判別用シート"  
    .Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "要修正_1 回目入力"  
    .Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "要修正_2 回目入力"  
    .Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "修正用シート"  
    .Worksheets.Add(after:=Worksheets(Worksheets.Count)).Name = "チェック完了データ"  
  End With  
End Sub
```

' シート保護サブルーチン

```
Sub preparation_sheetprotect()  
  Worksheets("1").Protect UserInterfaceOnly:=True  
  Worksheets("2").Protect UserInterfaceOnly:=True  
  Worksheets("判別用シート").Protect UserInterfaceOnly:=True  
  Worksheets("要修正_1 回目入力").Protect UserInterfaceOnly:=True  
  Worksheets("要修正_2 回目入力").Protect UserInterfaceOnly:=True  
  Worksheets("修正用シート").Protect UserInterfaceOnly:=True  
  Worksheets("チェック完了データ").Protect UserInterfaceOnly:=True
```

```
End sub
```

```
'管理用シート作成サブルーチン
```

```
Sub preparation_controlsheetmake()
  Worksheets("管理用シート").Activate
  With ActiveSheet
    .Cells(2, 1).Value = "管理用メニュー"
    .Cells(2, 2).Value = "黄色の部分に情報を入力してください。"
    .Cells(5, 1).Value = "作業担当者 1 氏名→"
    .Cells(6, 1).Value = "作業担当者 2 氏名→"
    .Cells(8, 1).Value = "先頭ケース番号→"
    .Cells(9, 1).Value = "末尾ケース番号→"
    .Cells(2, 4).Value = "【操作メニュー】"
    .Cells(2, 5).Value = "【ショートカットキー】"
    .Cells(3, 4).Value = "エラーケース判別準備"
    .Cells(4, 4).Value = "エラーケース判別"
    .Cells(5, 4).Value = "エラーケースセット"
    .Cells(6, 4).Value = "エラー箇所表示"
    .Cells(7, 4).Value = "修正完了データ格納"
    .Cells(8, 4).Value = "データファイル保存"
    .Cells(3, 5).Value = "Ctrl + Shift + D"
    .Cells(4, 5).Value = "Ctrl + Shift + C"
    .Cells(5, 5).Value = "Ctrl + Shift + V"
    .Cells(6, 5).Value = "Ctrl + Shift + H"
    .Cells(7, 5).Value = "Ctrl + Shift + K"
    .Cells(8, 5).Value = "Ctrl + Shift + S"
    .Columns("A:A").ColumnWidth = 21
    .Columns("B:B").ColumnWidth = 45
    .Columns("D:D").ColumnWidth = 30
    .Columns("E:E").ColumnWidth = 28
    .Range("D2:E8").Borders.LineStyle = xlContinuous
    .Cells(2, 4).HorizontalAlignment = xlCenter
    .Cells(2, 5).HorizontalAlignment = xlCenter
    .Cells(5, 1).HorizontalAlignment = xlRight
    .Cells(6, 1).HorizontalAlignment = xlRight
    .Cells(8, 1).HorizontalAlignment = xlRight
    .Cells(9, 1).HorizontalAlignment = xlRight
    .Range("D2:E8").Interior.ColorIndex = 4
    .Cells(5, 2).Interior.ColorIndex = 6
    .Cells(6, 2).Interior.ColorIndex = 6
    .Cells(8, 2).Interior.ColorIndex = 6
    .Cells(9, 2).Interior.ColorIndex = 6
    .Cells(2, 2).Interior.ColorIndex = 6
  End With
End Sub
```

```
'ショートカットキー割り当てサブルーチン
```

```
Sub preparation_shortcutkey()
  Application.OnKey "+^{D}", "step1_dataset"
  Application.OnKey "+^{C}", "step1_distinction"
  Application.OnKey "+^{V}", "step2_datacorrection"
  Application.OnKey "+^{H}", "step2_showerrors"
  Application.OnKey "+^{K}", "step2_datastorage"
  Application.OnKey "+^{S}", "step3_savefile"
End Sub
```

```
' OOOOOOOO メイン OOOOOOOO
```

```
Sub preparation_sheetsmaking()
  Call preparation_making
  Call preparation_sheetprotect
  Call preparation_shortcutkey
  Call preparation_controlsheetmake
  Application.MoveAfterReturnDirection = xlToRight
  Worksheets("管理用シート").Activate
End sub
```

```
' ●●●●●●●●●●●●●●●● step1: エラーケース判別準備(dataset) ●●●●●●●●●●●●●●●●

' OOOOOOOO サブルーチン OOOOOOOO

' データ並べ替えサブルーチン
Sub step1_imputdata_sort()
  Worksheets("1 回目入力").Activate
  Range("A1").CurrentRegion.Select
  Selection.Offset(1).Resize(Selection.Rows.Count - 1).Select
  Selection.Sort key1:=Range("A1"), order1:=xlAscending
  Worksheets("2 回目入力").Activate
  Range("A1").CurrentRegion.Select
  Selection.Offset(1).Resize(Selection.Rows.Count - 1).Select
  Selection.Sort key1:=Range("A1"), order1:=xlAscending
End Sub

Sub step1_idcollation()
  Worksheets("1 回目入力").Activate
  Columns("A").Select
  Selection.Copy
  Worksheets("管理用シート").Activate
  Columns("L").Select
  Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:=False, Transpose:=False
  Application.CutCopyMode = False
  Worksheets("2 回目入力").Activate
  Columns("A").Select
  Selection.Copy
  Worksheets("管理用シート").Activate
  Columns("M").Select
  Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:=False, Transpose:=False
  ActiveSheet.Range("N2").Value = "=IF(EXACT(L2,M2))=TRUE,0,1)"
  ActiveSheet.Range("N2").Select
  Selection.Copy
  Dim i As Integer
  i = 3
  Do Until ActiveSheet.Cells(i, 13) = ""
    ActiveSheet.Cells(i, 14).Select
    ActiveSheet.Paste
    i = i + 1
  Loop
  Application.CutCopyMode = False
  Worksheets("管理用シート").Activate
  With ActiveSheet
    .Cells(2, 9).Value = "↓ID 不一致ケース数"
    .Cells(3, 9).Value = "=sum(N2:N10000)"
  End With
End Sub

' 入力データ複製サブルーチン
Sub step1_dataset_copy()
  Worksheets("1").Activate
  ActiveSheet.Unprotect
  Sheets("1 回目入力").Select
  Cells.Select
  Selection.Copy
  Sheets("1").Select
  Cells.Select
  Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:=False, Transpose:=False
  Worksheets("2").Activate
  ActiveSheet.Unprotect
  Sheets("2 回目入力").Select
  Cells.Select
  Selection.Copy
  Sheets("2").Select
  Cells.Select
```

```

Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:=False, Transpose:=False
Application.CutCopyMode = False
Worksheets("1 回目入力").Protect UserInterfaceOnly:=True
Worksheets("2 回目入力").Protect UserInterfaceOnly:=True
End Sub

```

'判別用シート設定サブルーチン

```

Sub step1_dataset_checksheets1()
Worksheets("判別用シート").activate
ActiveSheet.Unprotect
Worksheets("1").Activate
ActiveSheet.Rows("1:1").Copy
Worksheets("判別用シート").Activate
ActiveSheet.Range("A1").Select
ActiveSheet.Paste
ActiveSheet.Range("B4").Value = "=if(B2=B3,0,1)"
ActiveSheet.Range("B4").Select
Selection.Copy
Dim i As Integer
i = 3
Do Until ActiveSheet.Cells(1, i) = ""
    ActiveSheet.Cells(4, i).Select
    ActiveSheet.Paste
    i = i + 1
Loop
Application.CutCopyMode = False
ActiveSheet.Rows("4:4").Select
Selection.FormatConditions.Delete
Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
    Formula1:="1"
Selection.FormatConditions(1).Interior.ColorIndex = 3
Worksheets("判別用シート").Cells(6, 1).Value = "エラー数↓"
Worksheets("判別用シート").Cells(7, 1).Value = "=sum(B4:IV4)"
ActiveSheet.Range("A1").Select
End Sub

```

'エラーケース用シート設定サブルーチン

```

Sub step1_dataset_revise()
Worksheets("要修正_1 回目入力").activate
ActiveSheet.Unprotect
Worksheets("1").Activate
ActiveSheet.Rows("1:1").Copy
Worksheets("要修正_1 回目入力").Activate
ActiveSheet.Range("A1").Select
ActiveSheet.Paste
Application.CutCopyMode = False
Worksheets("要修正_2 回目入力").activate
ActiveSheet.Unprotect
Worksheets("1").Activate
ActiveSheet.Rows("1:1").Copy
Worksheets("要修正_2 回目入力").Activate
ActiveSheet.Range("A1").Select
ActiveSheet.Paste
Application.CutCopyMode = False
End sub

```

'データ修正用シート設定サブルーチン

```

Sub step1_dataset_checksheets2()
Worksheets("修正用シート").Activate
ActiveSheet.Unprotect
Worksheets("1").Activate
ActiveSheet.Rows("1:1").Copy
Worksheets("修正用シート").Activate
ActiveSheet.Range("A3").Select
ActiveSheet.Paste
Application.CutCopyMode = False
Worksheets("修正用シート").Activate

```

```

ActiveSheet.Range("B6").Value = "=if(B4=B5,0,1)"
ActiveSheet.Range("B6").Select
Selection.Copy
Dim i As Integer
i = 3
Do Until ActiveSheet.Cells(3, i) = ""
    ActiveSheet.Cells(6, i).Select
    ActiveSheet.Paste
    i = i + 1
Loop
Application.CutCopyMode = False
Worksheets("修正用シート").Activate
With ActiveSheet
    .Range("A1").Value = "ID ↓"
    .Range("A3").Value = ""
    .Range("A4").Value = "1 回目入力値→"
    .Range("A5").Value = "2 回目入力値→"
    .Range("A6").Value = "一致:0, 不一致:1"
    .Range("A10").Value = "入力時エラー数 ↓"
    .Range("A11").Value = "=sum(B6:IV6)"
    .Range("A13").Value = "修正が必要なセル数 ↓"
    .Range("A14").Value = "=countif(B8:IV8, ""要修正"*)"
    .Range("A16").Value = "【操作メニュー】"
    .Range("A17").Value = "エラーケースセット"
    .Range("A18").Value = "→Ctrl + Shift + V"
    .Range("A19").Value = "エラー箇所表示"
    .Range("A20").Value = "→Ctrl + Shift + H"
    .Range("A21").Value = "修正完了データ格納"
    .Range("A22").Value = "→Ctrl + Shift + K"
    .Range("A23").Value = "データファイル保存"
    .Range("A24").Value = "→Ctrl + Shift + S"
    .Columns("A:A").ColumnWidth = 23
    .Range("A16:A24").Borders.LineStyle = xlContinuous
    .Range("A16").HorizontalAlignment = xlCenter
    .Range("A1").HorizontalAlignment = xlRight
    .Range("A3").HorizontalAlignment = xlRight
    .Range("A6").HorizontalAlignment = xlRight
    .Range("A8").HorizontalAlignment = xlRight
    .Range("A10").HorizontalAlignment = xlRight
    .Range("A11").HorizontalAlignment = xlRight
    .Range("A13").HorizontalAlignment = xlRight
    .Range("A14").HorizontalAlignment = xlRight
    .Range("A16:A24").Interior.ColorIndex = 4
End With
Worksheets("修正用シート").Activate
ActiveSheet.Rows("6:6").Select
Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, Formula1:=""1"
Selection.FormatConditions(1).Interior.ColorIndex = 6
ActiveSheet.Rows("8:8").Select
Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, Formula1:=""要修正""
Selection.FormatConditions(1).Interior.ColorIndex = 3
ActiveSheet.Range("B4").Select
ActiveWindow.FreezePanes = True
ActiveSheet.Range("A1").Select
End Sub

'チェック完了データ用シート作成サブルーチン
Sub step1_dataset_checkeddata()
    Worksheets("チェック完了データ").Activate
    ActiveSheet.Unprotect
    Worksheets("1").Activate
    ActiveSheet.Rows("1:1").Copy
    Worksheets("チェック完了データ").Activate
    ActiveSheet.Range("A1").Select
    ActiveSheet.Paste
End Sub

```


' 00000000 サブルーチン呼出 00000000

```
Sub step1_dataset_subcall()
    Call step1_dataset_copy
    Call step1_dataset_checksheet1
    Call step1_dataset_revise
    Call step1_dataset_checksheet2
    Call step1_dataset_checkeddata
    Call stepx_control_allsheetprotect
    Worksheets("管理用シート").Activate
End sub
```

' 00000000 メイン 00000000

```
Sub step1_dataset()
    If Worksheets("1").Cells(2, 1).Value = "" Then
        If Worksheets("要修正_1回目入力").Cells(2, 1).Value = "" Then
            If Worksheets("チェック完了データ").Cells(2, 1).Value = "" Then
                Call step1_imputdata_sort
                Call step1_idcollation()
                If Worksheets("管理用シート").Cells(3, 9).Value = "0" Then
                    Call step1_dataset_subcall
                    MsgBox "エラーケース判別の準備ができました。実際の判別作業に移ってください。"
                Else:
                    MsgBox "【警告！】データ入力が2回分なされていない可能性があります。ブックをいったん閉じ、データ入力がきちんと2回 _
                        なされているか確認してください。"
                End if
            Else:
                MsgBox "エラーケース判別は完了しています。"
            End if
        Else:
            MsgBox "エラーケース判別は完了しています。"
        End If
    Else:
        MsgBox "エラーケース判別の準備は完了しています。"
    End if
End sub
```

' ●●●●●●●●●●●●●●●●●●● step1: エラーケース判別(distinction) ●●●●●●●●●●●●●●●●●●●●

' 00000000 サブルーチン 00000000

```
' データチェックサブルーチン
Sub step1_distinction_copy()
    Worksheets("1").Activate
    ActiveSheet.Unprotect
    ActiveSheet.Rows(2).Copy
    Worksheets("判別用シート").Activate
    ActiveSheet.Unprotect
    ActiveSheet.Range("A2").Select
    ActiveSheet.Paste
    Worksheets("2").Activate
    ActiveSheet.Unprotect
    ActiveSheet.Rows(2).Copy
    Worksheets("判別用シート").Activate
    ActiveSheet.Range("A3").Select
    ActiveSheet.Paste
    Worksheets("判別用シート").Activate
End Sub
```

```
' 完成データ格納サブルーチン
Sub step1_distinction_storage()
```

```
Worksheets("チェック完了データ").Activate
ActiveSheet.Unprotect
Worksheets("判別用シート").Activate
Rows("2:2").Copy
Worksheets("チェック完了データ").Activate
Rows("2:2").Insert
Application.CutCopyMode = False
Worksheets("判別用シート").Activate
Rows("2:3").Select
Selection.ClearContents
Worksheets("1").Activate
Range("2:2").Delete
Worksheets("2").Activate
Range("2:2").Delete
End Sub
```

'エラーケース移動サブルーチン

```
Sub step1_distinction_move()
Worksheets("判別用シート").Activate
Rows("2:2").Copy
Worksheets("要修正_1回目入力").Activate
Rows("2:2").Insert
Application.CutCopyMode = False
Worksheets("判別用シート").Activate
Rows("3:3").Copy
Worksheets("要修正_2回目入力").Activate
Rows("2:2").Insert
Application.CutCopyMode = False
Worksheets("判別用シート").Activate
Rows("2:3").Select
Selection.ClearContents
Worksheets("1").Activate
Range("2:2").Delete
Worksheets("2").Activate
Range("2:2").Delete
Worksheets("判別用シート").Activate
End sub
```

'要修正シート並べ替えサブルーチン

```
Sub step1_distinction_sort()
Worksheets("要修正_1回目入力").Activate
Range("A1").CurrentRegion.Select
Selection.Offset(1).Resize(Selection.Rows.Count - 1).Select
Selection.Sort key1:=Range("A1"), order1:=xlAscending
Worksheets("要修正_2回目入力").Activate
Range("A1").CurrentRegion.Select
Selection.Offset(1).Resize(Selection.Rows.Count - 1).Select
Selection.Sort key1:=Range("A1"), order1:=xlAscending
End sub
```

'OOOOOOOO サブルーチン呼出 OOOOOOOO

```
Sub step1_distinction_subcall()
Call step1_distinction_copy
Worksheets("判別用シート").Activate 'シート1アクティヴ化
ActiveSheet.Range("A7").Activate
If Worksheets("判別用シート").Cells(7, 1).Value > "0" Then
Call step1_distinction_move
Else:
Call step1_distinction_storage
End If
End Sub
```

```
'OOOOOOOOO メイン OOOOOOOOO
Sub step1_distinction()
  If Worksheets("1").Cells(2, 1).Value = "" Then
    MsgBox "エラーケース判別のためのデータがありません。エラーケースの判別が完了しているか、もしくは判別のためのデータがセッ _
           していません。"
  Else:
    If Worksheets("要修正_1回目入力").Cells(2, 1).Value = "" Then
      If Worksheets("チェック完了データ").Cells(2, 1).Value = "" Then
        Do Until Worksheets("1").Cells(2, 1).Value = ""
          Call step1_distinction_subcall
        Loop
        If Worksheets("要修正_1回目入力").Cells(2, 1).Value = "" Then
          MsgBox "エラー修正を要するケースはありません。データを保存してください。"
        Else:
          Call step1_distinction_sort
          Worksheets("要修正_1回目入力").Activate
          Range("A1").CurrentRegion.Select
          MsgBox "エラー修正を要するケースが " & Selection.Rows.Count - 1 & " ケースあります。「修正用シート」で、 _
                 修正作業をおこなってください。"
          Call stepx_control_allsheetprotect
          Worksheets("修正用シート").Activate
        End if
      Else:
        MsgBox "エラーケース判別は完了しています。"
      End if
    Else:
      MsgBox "エラーケース判別は完了しています。"
    End if
  End if
End sub
```

'●●●●●●●●●●●●●●●●●●●● step2: 修正データセット(datacorrection) ●●●●●●●●●●●●●●●●●●●●

```
'OOOOOOOOO サブルーチン OOOOOOOOO
'チェック用データコピーサブルーチン
Sub step2_datacorrection_copy()
  Call stepx_control_allsheetunprotect
  Worksheets("要修正_1回目入力").Activate 'シート1 アクティブ化
  ActiveSheet.Rows(2).Copy
  Worksheets("修正用シート").Activate
  ActiveSheet.Range("A4").Select
  ActiveSheet.Paste
  Worksheets("要修正_2回目入力").Activate
  ActiveSheet.Rows(2).Copy
  Worksheets("修正用シート").Activate
  ActiveSheet.Range("A5").Select
  ActiveSheet.Paste
  Worksheets("修正用シート").Activate
  ActiveSheet.Range("A4").select
  Selection.Copy
  ActiveSheet.Range("A2").select
  ActiveSheet.Paste
  ActiveSheet.Range("A4").Value = "1回目入力値→"
  ActiveSheet.Range("A5").Value = "2回目入力値→"
  ActiveSheet.Range("A4").HorizontalAlignment = xlRight
  ActiveSheet.Range("A5").HorizontalAlignment = xlRight
End Sub
```

```
'修正用データ入力部分設定サブルーチン
sub step2_datacorrection_sheetset()
  Worksheets("修正用シート").Activate
  ActiveSheet.Range("B8").Value = "=if(B4=B5,B4,""要修正"")"
  ActiveSheet.Range("B8").Select
  Selection.Copy
```

```

    Dim i As Integer
    i = 3
    Do Until ActiveSheet.Cells(3, i) = ""
        ActiveSheet.Cells(8, i).Select
        ActiveSheet.Paste
        i = i + 1
    Loop
    Application.CutCopyMode = False
    ActiveSheet.Range("A8").Value = "修正後の値→"
    ActiveSheet.Range("A1").Select
End Sub

'入力部分以外への入力拒否設定サブルーチン
Sub step2_datacorrection_protect()
    Worksheets("修正用シート").Activate
    With ActiveSheet
        .Range("A8:IV8").Locked = False
        .Range("A30:IV30").Locked = False
    End With
    Worksheets("修正用シート").Protect UserInterfaceOnly:=True
End sub

'エラー数表示サブルーチン
Sub step2_datacorrection_show()
    ActiveSheet.Range("A9").Activate
    If Worksheets("修正用シート").Cells(11, 1).Value > "0" Then
        MsgBox "ID:" & Range("A2").Value & " 入力時エラーが" & Range("A11").Value & "箇所あります"
        ActiveSheet.Range("B8").Activate
    Else:
        MsgBox "入力時エラー箇所はありません。データを格納してください。"
    End If
End Sub

'○○○○○○○○ サブルーチン呼出 ○○○○○○○○

Sub step2_datacorrection_subcall()
    Call step2_datacorrection_copy
    Call step2_datacorrection_sheetset
    Call step2_datacorrection_protect
    Call step2_datacorrection_show
End sub

'○○○○○○○○ メイン ○○○○○○○○

Sub step2_datacorrection()
    If Worksheets("要修正_1回目入力").Cells(2, 1).Value = "" Then
        MsgBox "修正を要するケースがありません。"
    ElseIf Worksheets("修正用シート").Cells(4, 2).Value = "" Then
        Call step2_datacorrection_subcall
    Else:
        MsgBox "前ケースの修正が終了していません。前ケースの修正を完了し、データを格納してください。"
    End If
End sub

'●●●●●●●●●●●●●●●●●●●● step2: エラー箇所表示(showerrors) ●●●●●●●●●●●●●●●●●●●●

'○○○○○○○○ サブルーチン ○○○○○○○○

Sub step2_showerrors_main()
    Worksheets("修正用シート").Activate
    Dim i As Integer

```

```

i = 1
Do Until Worksheets("修正用シート").Cells(8, i).Value = "要修正"
    ActiveSheet.Range("B8").Activate
    ActiveCell.Offset(0, i - 1).Select
    i = i + 1
Loop
End Sub

' OOOOOOOO メイン OOOOOOOO

Sub step2_showerrors()
    If Worksheets("修正用シート").Cells(4, 2).Value = "" Then
        MsgBox "表示可能なケースがありません。修正用のケースをセットしてください。"
    ElseIf Worksheets("修正用シート").Cells(14, 1).Value > "0" Then
        Call step2_showerrors_main
    Else
        MsgBox "修正は完了しています。データを格納してください。"
    End If
End Sub

' ●●●●●●●●●●●●●●●●●●●●●●●● step2: データ格納(datastorage) ●●●●●●●●●●●●●●●●●●●●●●●●

' OOOOOOOO サブルーチン OOOOOOOO

' データ格納サブルーチン
Sub step2_datastorage_copy()
    Worksheets("修正用シート").Activate
    Rows("8:8").Copy
    ActiveSheet.Range("A30").select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    ActiveSheet.Range("A2").select
    selection.copy
    ActiveSheet.Range("A30").select
    ActiveSheet.paste
    Rows("30:30").Copy
    Worksheets("チェック完了データ").Activate
    Rows("2:2").Insert
    Application.CutCopyMode = False
End sub

' データ削除サブルーチン
Sub step2_datastorage_delete()
    Worksheets("修正用シート").Activate
    Rows("4:5").Select
    Selection.ClearContents
    Rows("30:30").Select
    Selection.ClearContents
    Rows("8:8").Select
    Selection.ClearContents
    ActiveSheet.Cells(8, 1).Value = "修正後の値→"
    ActiveSheet.Range("A8").HorizontalAlignment = xlRight
    Worksheets("要修正_1 回目入力").Activate
    Range("2:2").Delete
    Worksheets("要修正_2 回目入力").Activate
    Range("2:2").Delete
End Sub

' OOOOOOOO サブルーチン呼出 OOOOOOOO

Sub step2_datastorage_subcall()
    Call stepx_control_allsheetunprotect

```

```

Call step2_datastorage_copy
Call step2_datastorage_delete
Call stepx_control_allsheetprotect
End Sub

```

' OOOOOOOO メイン OOOOOOOO

```

Sub step2_datastorage()
If Worksheets("修正用シート").Cells(4, 2).Value = "" Then
MsgBox "格納可能なデータがありません。修正用のケースをセットしてください。"
Else:
If Worksheets("修正用シート").Cells(14, 1).Value > "0" Then
MsgBox "エラー修正が完了していません。エラー箇所を修正してください。"
Else:
Call step2_datastorage_subcall
ActiveSheet.Range("B8").Activate
End If
End If
Worksheets("修正用シート").Activate
End Sub

```

' ●●●●●●●●●●●●●●●●●●●●●●●● step3: データ保存 (savefile) ●●●●●●●●●●●●●●●●●●●●●●●●

' OOOOOOOO サブルーチン OOOOOOOO

```

' データ並べ替えサブルーチン
Sub step3_savefile_datasort()
Worksheets("チェック完了データ").Activate
Range("A1").CurrentRegion.Select
Selection.Offset(1).Resize(Selection.Rows.Count - 1).Select
Selection.Sort key1:=Range("A1"), order1:=xlAscending
End Sub

```

```

' データ保存サブルーチン
Sub step3_savefile_datasave()
Worksheets("管理用シート").Activate
ActiveSheet.Range("B11").Value = Range("B5").Value & Range("B6").Value & "(" & Range("B8").Value & _
"-" & Range("B9").Value & ")"
Application.Dialogs(xlDialogSaveAs).Show arg1:=Range("B11").Value
End Sub

```

' OOOOOOOO メイン OOOOOOOO

```

Sub step3_savefile
If Worksheets("要修正_1回目入力").Cells(2, 1).Value = "" Then
If Worksheets("チェック完了データ").Cells(2, 1).Value = "" Then
Msgbox "エラーの修正作業が完了していません。修正作業をおこなった後に保存してください。"
Else:
Call step3_savefile_datasort
Call step3_savefile_datasave
End if
Else:
Msgbox "エラー未修正のケースが残っています。全ケースの修正を完了してください。"
End if
End sub

```

' ●●●●●●●●●●●●●●●●●●●●●●●● stepx: 管理用 ●●●●●●●●●●●●●●●●●●●●●●●●

```

' シート保護
Sub stepx_control_allsheetprotect()

```

```
Worksheets("1 回目入力").Protect UserInterfaceOnly:=True
Worksheets("2 回目入力").Protect UserInterfaceOnly:=True
Worksheets("1").Protect UserInterfaceOnly:=True
Worksheets("2").Protect UserInterfaceOnly:=True
Worksheets("要修正_1 回目入力").Protect UserInterfaceOnly:=True
Worksheets("要修正_2 回目入力").Protect UserInterfaceOnly:=True
Worksheets("チェック完了データ").Protect UserInterfaceOnly:=True
End Sub
```

'シート保護解除

```
Sub stepx_control_allsheetunprotect()
    Worksheets("1 回目入力").activate
    ActiveSheet.Unprotect
    Worksheets("2 回目入力").activate
    ActiveSheet.Unprotect
    Worksheets("1").activate
    ActiveSheet.Unprotect
    Worksheets("2").activate
    ActiveSheet.Unprotect
    Worksheets("要修正_1 回目入力").activate
    ActiveSheet.Unprotect
    Worksheets("要修正_2 回目入力").activate
    ActiveSheet.Unprotect
    Worksheets("チェック完了データ").activate
    ActiveSheet.Unprotect
End sub
```