

第 1 部 プログラム編

第1章 AUTOCODEプログラムについて(佐藤 裕)

AUTOCODE for Windows Ver. 1.01

1. プログラムの概要

アンケート調査など自記式質問紙調査における自由回答項目の計量的分析は、データの数量化 - コーディングに大きな困難があります。体験したことのある方はご存じでしょうが、文章とコード表を照らし合わせながらコーディングしていくのはかなり大変な作業ですし、複数の人間でコードする場合には判断基準の意志統一が難しいし、判断に困るような例はどんどん出て来るし、大量のデータだと、コードしているうちにコードルールを変更したくなったり、だんだんと判断が変わってきて、結局もう一度最初からやり直したり……。

AUTOCODE は、こういった問題のいくつかを解決するために開発したプログラムです。コンピュータによる自動コーディングはコーディングにかかる労力を節約し、コーディングの一貫性を高め、コーディングルールを明示化することによってチェックを容易にします。

しかし、最初に断っておきたいのですが、AUTOCODE はコーディングを「完全に」自動化するプログラムではありません。AUTOCODE の主な機能は2つです。

コーディング機能

まず1つは、コーディングルールと自由回答の文章を照らし合わせて、それぞれの回答にコードを割り当てる機能です。コーディングルールを作成することにより、手作業によるコーディングでは曖昧になってしまうことの多いコーディングの基準を明確にすることができ、それを第三者に評価してもらうこともできるようになります。またコーディングルールさえできてしまえば、コーディング作業は一瞬の間にできてしまいますので、コーディングルールの作成に十分な手間をかけることができます。

コーディングルール作成支援機能

コーディングルールの作成は大変な作業です。AUTOCODE の場合は回答文の中に出現する文字列を検索してそれに対応するコードを出力するのですが、そのためには検索する文字列のリストを実際の回答文から抽出する作業が必要になります。MS-DOS 版の AUTOCODE ではこの作業はエディタなどを用いて行っていたのですが、膨大な時間がかかっていました。

Windows 版の AUTOCODE は、コーディングルール作成の省力化を主要な目的として開発しました。省力化といっても自動的にルールを作成してくれるわけではなく、検索すべき文字列の選択はあくまでも分析者が考えなくてはなりません。しかし、コーディング結果をリアルタイムに表示させながら、回答文から検索すべき文字列を抽出してゆく AUTOCODE の

ユーザーインターフェイスは、エディタなどを用いるよりもずっとコーディング作業を楽なものにしてくれるはずです。

2. コンピュータコーディングとは？

コンピュータを使わないコーディング

コンピュータを使わずに自由回答のコーディングを行うには、分析者が1つ1つの回答を見てそれにあてはまるコードを記述していくことになります。その際問題になるのはコーディングの基準です。

単語で答えるような質問の場合はそれほど問題にはなりませんが、文章で答えるような場合だと、どのような回答であればどのようにコードするのかをあらかじめ厳密に決めておくのは困難です。通常はコード作業をしながら曖昧なケースについてそのつど考えていくことになるでしょう。しかし、そのつど考えた新しい基準をすべて記録しておくことは困難であり、実際には「このような回答はどう判断したんだっけ？」と前にコードした文章を見なおすこともあります。これが面倒になってくるとあやふやな記憶に頼って判断してしまうというようなこともあると思います。複数の人が分担してコーディングをするとコードの基準はさらに問題になります。

また手作業によるコーディングを用いた分析を見る方にしてみても、どのような回答をどのように読み取ったのかが明らかにされていないと、分析結果の判断がしにくいことがあります。

また、手作業によるコーディングは労力も膨大になります。個々の回答を見ながらコードを割り当て、もう1度（あるいは2度、3度）全体をチェックして、それから割り当てたコードをコンピュータに入力し、さらに入力されたデータをチェックする。きちんとやろうとするとこれぐらいの作業は必要でしょう。

コンピュータコーディングのメリット

コンピュータによるコーディングの1番のメリットはコーディングの基準を明確にできるということです。融通の効かないコンピュータにコーディングをさせるわけですから、その基準は明確でなければなりません。AUTOCODE の場合はコード名と抽出文字列を結び付けるコーディングルールを作成し、それによってコンピュータに指示を与えます。

コーディングルールの作成は、複数の分析者がデータを共有する場合や、第三者にコーディングの基準を示す場合にも役に立ちます。これによって、コーディングの信頼性を判断してもらうことが可能です。

いったんコーディングルールを作成してしまえば、あとはコンピュータがデータセットまで作ってくれますので余計な入力作業は必要がないし、その際に発生するかもしれない転記ミスなども防止できます。コーディングのチェックも容易ですし、コーディング基準

の修正も比較的簡単にできます。

コンピュータコーディング(AUTOCODE)の限界

しかし、コンピュータによるコーディングも万能ではありません。

まず、コーディングルールの作成はかなり大変な作業になります。手作業でのコーディングでは頭の中にあっただけのルールをいちいち記述してゆく作業だと考えてよいでしょう。

作業の量は回答文の種類や分析目的によってかなり変わりますが、場合によってはコーディングルールが回答文の半分くらいの分量になってしまうこともあり得ます。

そこで、AUTOCODE の Windows 版ではコーディングルールの作成を支援する機能を重視しています。それでもまだ大変な作業であることには変わりありませんが、従来に比べて大幅にルール作成にかかる労力を減らすことができるようになったと思います。

また、AUTOCODE では文字列の検索によってコードを割り当てていますが、そのことによる限界もあります。つまり回答文全体の文脈を判断してコードを割り当てているのではないということです。ただし、抽出文字列の長さには制限がありませんので、長い文字列を抽出することによってある程度は対応することができます(コーディングルールは見にくくなりますが)。

3. コーディングの方法

3.1. AUTOCODEのコーディングルール

AUTOCODE のコーディングルールはきわめて単純です。基本的には回答文中から文字列を検索し、それが見つかったかどうかを報告しているだけです。しかし、それだけでは使いものにならないので、ちょっとした工夫を2つほどしています。

まず、異なった表現で同じ意味が表されることがあることを考える必要があります。例えば「コンピュータ」を「電子計算機」という言葉で表す人もいるだろうし、「パソコン」という言葉も用いられるでしょう。あるいは「サーバ」「ホスト」なども分析目的によってはひとまとめにしたほうがいいかもしれません。そこで、AUTOCODE では、1つの「コード名」に複数の「抽出文字列」を対応させるルールを用います。上の例であれば、「コンピュータ」というコード名に「電子計算機」「パソコン」など(もちろん「コンピュータ」自体も)を対応させるわけです。このようなルールを受け取った AUTOCODE は、回答文を検索して「電子計算機」や「パソコン」などの文字列を発見すると「コンピュータ」というコードが見つかったと報告します。

次に文字列を検索してコーディングする際に問題になるのは、抽出したい文字列が他の意味の言葉の一部分になっていたり、前後の文脈から抽出すべきでないとは判断される場合です。

前者は特に短い文字列で頻発します。例えば「親」という言葉を母親とか父親の意味で抜き出したいときに、これをそのまま抽出文字列として使うと「親切」「親身になって」「親しい」などの言葉がすべて抜き出されてしまいます。

後者は、叙述的な言葉の後に否定の表現が現れるときなどに発生します。例えば「悲しい」という文字列を抜き出したい場合に「悲しいことはない」という表現まで抜き出されては困るわけです。

AUTOCODE はこの問題にごく単純な方法で対応しています。それぞれの抽出文字列に対して、もし前後に特定の文字列が隣接していた場合にはコードの割り当てを回避するというルールです。このような文字列を「回避文字列」と呼びます。

「親」の場合には、後ろに「切」「身になって」「しい」などの文字列が隣接していた場合には、コードの割り当てを回避するルールを作ればよいのです。この場合回避文字列は、「>切」「>身になって」「>しい」となります。それぞれの文字列の最初の「>」は「後ろにつく」という意味です。前につく文字列の場合は「<」を最初につけます。

なお、AUTOCODE のコーディングルールは回答を分類するようなものではなく、いくつかの項目についてあてはまるか否かを調べるルールです。選択肢を用いた質問では複数回答を許容する質問に当たります。

3.2. コーディングのプロセス

AUTOCODE による自由回答のコーディングのプロセスは、だいたい以下のような手順になります。

テキストファイルの準備

まず、自由回答文のテキストファイルを準備します。

コーディングルールの作成

自由回答の文章を参照しながら、コーディングルールを作成します。

コーディングルールの編集

コーディング結果を見ながらコーディングルールを修正します。

コーディング結果の出力

AUTOCODE はごく簡単な統計計算が可能ですが、本格的な分析のためには他の統計プログラムを利用する必要があります。AUTOCODE は他のプログラムで利用可能なデータ形式でコーディング結果を出力します。

3.3. テキストファイルの準備

まず、自由回答の文章をコンピュータに入力する必要があります。入力はテキストファイルが出力できるプログラムなら何で行っても結構ですが、ワープロで入力を行う場合にはワープロ独自の文書形式ではなくテキストファイル形式で保存してください。

入力はケースID番号と回答文が交互になる形式で行います。具体例を以下に示します。

```
0001
自由回答のコーディングは大変面倒だ
0002
自由回答はコードをつけるのが難しいし、分析もわずらわしい
0004
自由回答の分析は面白い
...
1234
これが最後の回答です
```

1行目の「0001」、3行目の「0002」などがID番号です。これはそれが誰の回答なのかを識別するために必要で、自由回答以外のデータと結合するときのキーとして用います。従って、重複していないことが最低限必要です（重複していてもエラーにはなりません）。フォーマットは半角数字で桁数をそろえてください（最大が4桁なら「1」は「0001」とする）。順番はどのようになっていても結構ですし、番号が飛んでいても問題ありません。無回答の場合はID番号だけを入力しても結構ですが、ID番号もない方が後の編集作業が楽です。

2、4、6行目などが回答文です。文節で切り分けたりせずにそのまま入力していただいて結構です。表記を統一したり（仮名を漢字にするとか）すると後の処理が楽にはなりますが、この段階で完全になおすのは困難でしょうし、「明らかな誤字に気がつけば訂正する」という程度でかまわないと思います。

このようにデータ入力の作業がかなり単純になっていることがAUTOCODEを使ったコーディングの1つの特徴です。この程度ならアルバイトにまかせたりデータ入力業社に外注したりしてもほとんど問題なくできるはずです。

回答文の長さには制限がありませんが、半角換算で10,000字以上になるとSPSSシンタックスファイルの出力が正常に行われません。途中で改行を入れても結構です。ケース数は5,000ケースが上限となっています。これは便宜的なものですから、もしそれ以上の数の回答を処理したい方が居られましたら作者までご連絡ください。

自由回答のテキストファイルのファイル名は拡張子も含めて自由です。

自由回答のテキストファイルができたら、AUTOCODEを起動し、「ファイル」メニューの「テキストを開く」を選んでテキストファイルを読み込んでください。

読み込みが終了すると、「テキスト」と表示されたエリアに最初のケースの回答文が表示

されます。その上のIDボックスには最初のケースのID番号（テキストファイルに記載された番号）が表示されているはずですが、また右下の「テキストファイル」と表示されたボックスにはファイル名、その下の「件数」のボックスには読み込まれたケース数が表示されています。まずこれらを確認してください。

テキスト表示エリアの左には、「TOP」「」「」「BTM」と表示されたボタンがあります。これらで表示するテキストを移動することができます。「」ボタンは一つ後のケースに、「」は一つ前のケース、「TOP」と「BTM」はそれぞれ最初と最後のケースに移動します。これらのボタンを押して、うまく読み込まれていないケースがないかどうかチェックしてください。

「BTM」ボタンの下のボックスにはそのケースの通し番号が表示されています。これはテキストファイル内でのそのケースの位置を示しており、ID番号とは異なります。ID番号は途中が抜けていたり小さいものから順に並んでいなくても良いことになっていますので、AUTOCODE は管理用に通し番号をつけています。このボックスに適当な数値を半角で入力し、その下の「GO」ボタンを押すと、その番号のケースにジャンプします。これはいったん中断したコーディング作業を再開するときに便利な機能です。

ここまでできれば、次はコーディングルールを作成することになります。

注意とヒント

自由回答のテキストファイルは1つの質問に対する回答しか入力することはできません。もしコーディングしたい質問が2つ以上ある場合は、それぞれについてファイルを作成する必要があります。しかし、回答を入力する際には別々に入力しては手間がかかりますので、1度に入力して（例えば、ID番号、質問Aの回答、質問Bの回答、というように）後から分割するという方法が考えられます。これはAWKなどのプログラムについての知識があればそう難しいことではありませんが、もし要望があれば専用のプログラムを作成します。

自由回答以外の部分と結合しようとする場合、ID番号の打ち間違いは致命的です。回答文の方は後で何度も読むことになるし、その時にチェックできるのですが、ID番号はチェックのしようもありません。そのため、入力時にはID番号を正確に入力することに特に気を配ってください。また、自由回答以外の部分をデータ入力する際に、自由回答の質問については、例えば回答あり - 「1」、回答なし - 「0」というようにコーディングして入力しておく、それをAUTOCODEが出力するデータと突き合わせてチェックすることができます。

3.4. コーディングルールの作成

3.4.1. コーディングルールの作成

このプロセスが、AUTOCODEを使ったコーディングにおいて、もっとも重要であり、もっとも手間のかかる部分です。

まず、分析しようとするデータをよく検討して、どのような文字列を抜き出せばよいの

かを考えます。例えば、前節であげた回答文例が、「自由回答法についてどう思うか？」という質問に対する回答だとして、コードルールを作ることを考えてみます。

まず、「自由回答法について」という質問なので、「自由回答(は/の)」という文字列は抜き出す必要がありません。それに対する評価が知りたいので、「コーディングは大変面倒だ」とか「分析は面白い」という部分が抜き出したい部分です。そこでまず考えられるのが、このような文字列をそのまま抜き出すという方法です。

また、「領域(コーディングや分析など)」と「評価」を別々に調べて、後で関連を見たいのなら、「コーディング」「分析」「面白い」「面倒」などの文字列を抜き出すこともできます(以下の説明はこちらの方法をとったということです)。

AUTOCODE は「単語」とか「文節」「文」「命題」などの単位を理解できませんので、どのような単位で文字列を抜き出すかは利用者に全くまかされています。

抜き出す文字列(「抽出文字列」と呼びます)が決まれば、つぎにそれを「同じ意味」あるいは「同じカテゴリーに属する」ものどうしまとめて、コード名をつけます。この例だと、「コーディング」と「コードをつける」は同じ意味ですので、1つのコードとしてまとめて、「コーディング」というコード名をつけます。

まず、どのような「コード名」にどのような「抽出文字列」を対応させるかについて、ある程度考えてから実際のコーディング作業に入ってください。

コーディングルールは、コーディングルールファイルとしていつでも保存、読み込みができます。保存するには「ファイル」メニューから「ルールファイルの保存」を、読み込むには「ファイル」メニューから「ルールファイルを開く」を選択します。コーディングルールファイルのデフォルト拡張子は「.cod」です。保存の際に拡張子を省略すると「.cod」が付けられます。

3.4.2. コード名の作成

ある程度コーディングの方針が定まったら、まずコード名から作成していきます。

「コード名」という名前のついたボックスが AUTOCODE の左下にあります。これがコード名を表示するボックスです。初期状態では「(未定義)」という文字だけが表示されています。この意味は2つ後の項目で説明します。

新しいコード名を作成するには、コード名ボックスの下の1行だけのボックスにその名前を入力し、そのすぐ上の「」ボタンをクリックします。するとコード名ボックスにそのコード名が表示されるはずですが。

新しいコード名は選択されているコード名の後(選択されていなければ最後)に入ります。類似したコード名をまとめておいた方が見やすくなると思います。

このようにして、あらかじめ決めたコード名を次々と入力していきます。

現在のバージョンではコード名は100件まで作成することができます。

コード名はこの段階ですべて入力しなくても、いつでも新たに作成することができます。

また名前を変えたり順序を入替えることもできます。(コード名の編集)

3.4.3. 抽出文字列の割り当て

次に回答文から抽出する文字列をコード名に割り当てます。抽出文字列の作成は、キーボードから入力するよりも実際の回答文から切り出してくる方がミスも少なく確実です。

そのため、この作業はテキストファイルを読み込んだ状態で行います(以下の作業はすべて同じです)。

テキストファイルを読み込み、順に表示させながら、抽出したい部分を探します。抽出したい部分が見つければ、まずそれに対応するコード名をクリックして選択します。次にテキストボックスの抽出したい部分をマウスかキーボードで選択し(反転表示されます)、抽出文字列と表示されたボックスの上の「 」ボタンを押します。すると抽出文字列のボックスに選択した文字列が加えられます。これで割り当ては完了です。

割り当てが終わってからテキストボックスを見ると、今抽出した箇所が青色で表示されているはずですが、これはこの部分がコーディングに用いられたということを示しています。そして、それに対応するコード名がその右の「コーディング結果」ボックスに表示されません。

このように、AUTOCODE ではコーディング結果の表示がリアルタイムに行われます。

この手順をくり返し、どんどんと抽出文字列を割り当てていきます。

抽出文字列が増えるに従って、以前抽出した文字列によって、新しく見る回答文がすでにコーディングされていることが増えてくるはずですが、その場合は新たにその文字列を抽出する必要はありません。

この操作はキーボードからも行うことができます(キーボードによる操作)。

もし抽出したい文字列に対応するコード名がなければいつでも追加することができます。また、抽出したいのだけどどのようにカテゴリー化すればよいか決めかねる場合は、一時的に「(未定義)」というコード名に割り当てておいて、あとで割り当てを変更することもできます。

抽出文字列は回答文から抽出すべきですが、キーボードから入力することもできるようになっていますし、編集も可能です。その方法はコード名の場合と同じですが、作成の場合はあらかじめ対応するコード名を選択してから行ってください(「コード名の作成」)。

抽出文字列表示用ボックスには、抽出文字列が文字コードの順に表示されます。これは同じ文字列(または先頭部分が共通の文字列)を重複して登録してしまわないようにするためです。

全く同じ抽出文字列を別のコード名に割り当てることもできます。ただし、その2つのコード名の間関係が曖昧になってしまう可能性もありますので、注意してください。

注意とヒント

抽出文字列の切り出し方は、分析者が自由に判断すればよいことですが、作者の考え方を紹介しておきます。

AUTOCODE のコーディングルールは、単なるコンピュータに与える指示ではなく、複数の分析者がコーディングの基準を共有したり、第三者に対してコーディングの基準を示す役割も持っていると考えられています。そのため、抽出文字列や回避文字列は、「意味がとれる」範囲を抜き出すべきだと思います。

例えば「困」という字が「困った」と「困る」という文字列の中だけで使われているなら、「困」という文字だけを抽出文字列として指定することにより、1つのルールで「困った」にも「困る」にも対応できます。しかし、第三者などが見た場合には、「困る」「困った」を抜き出したいのか「困難」を抜き出したのかわからないので、面倒でも「困った」と「困る」の両方を抽出文字列にした方がよいということなのです。

抽出文字列を追加した時は、その文字列で回答文の検索を行ったほうがよいでしょう。検索の結果都合の悪いケースが見つければ、すぐに回避文字列を設定することができます。

3.4.4. 未定義コードの利用

コード名ボックスに始めからある「(未定義)」という名前のコードは、割り当て先の決まらない抽出文字列を一時的に保管しておくためのコード名です。もちろんこのコードは実際のコーディングには使用されません。このコード名は名前を変更することも削除することもできません。

このコード名を活用して、コード名をまったく作らないで文字列の抽出を開始することもできます。そしてある程度抽出文字列がたまった段階でコード名を作りはじめるというのも一つの方法です。

抽出文字列の割り当てを後から変更する方法は、「抽出文字列の割り当て変更」を参照してください。

3.4.5. コーディングルールファイルの保存と読み出し

コーディングルールは「コーディングルールファイル」として、いつでも保存し、読み出すことができます。

保存するには、「ファイル」メニューの「ルールファイルの保存」、読み出すには「ファイル」メニューの「ルールファイルを開く」を選択してください。

ルールファイルの拡張子は「.cod」に固定されています。ファイル名を入力する際には拡張子なしで入力してください。

3.5. コーディングルールの編集

3.5.1. コーディング結果のチェック

コーディング結果のチェックはいつでも画面上で行うことができます。コーディングが

一応最後まで終わってからもよいし、ある程度まとまった段階でも、1つのケースで抽出が終了することに行っても結構です。

コーディング結果のチェックはテキストボックスとコーディング結果ボックスで行います。テキストボックスではコーディングに用いられた箇所が青色で表示されており、コーディング結果ボックスには割り当てられたコード名が表示されています。この結果を見て、不足しているコードがないか、あるいはコードされるべきではないのにコードが割り振られてしまっている箇所はないかをチェックします。

コードが不足している場合には、抽出文字列を追加します。

不適切なコードが割り振られてしまっている場合には、回避文字列の設定が必要です。

コーディング結果のチェックは、画面上だけでなく、ファイルに出力し（それを印刷して）行うこともできます。「出力」メニューの「コード対照リストの出力」を選択してください。出力はテキストファイル形式なので、エディタやワープロで読み込み、必要なら印刷してください。かなりの分量になると思いますので、縮小印刷ができればその方がいいでしょう。

3.5.2. 回避文字列の設定

回避文字列は抽出文字列の前後に特定の文字列が出現した場合にはコードの割り当てを回避するというルールを表現したものです。

例えば父親とか母親というの意味で「親」という文字列を抽出したいのに、回答文中に「親切」という文字列が出現すると、「親」というコードを割り当ててしまいます。そこで、後ろに「切」という文字列が隣接している場合はコードを回避するというルールを作ります。

まず、テキストボックスに問題となる箇所を表示させます。次に回避ルールを割り当てたいコード名と抽出文字列をそれぞれのボックスで選択します。テキストボックスで回避文字列として指定したい部分（上の例では「切」）をマウスかキーボードで選択し、回避文字列ボックスの上の「>」か「<」のボタンを押します。「>」ボタンをその文字列が抽出文字列の後ろに隣接する場合にコードを回避するというルール、「<」のボタンは前に隣接する場合のルールを割り当てます。「親切」の場合は「>」です。

テキストボックスとコーディング結果ボックスで回避ルールの効果を確認してください。テキストボックスの青色が消え、結果ボックスからもコード名はなくなっているはずですが。

この操作はキーボードから行うことができます（キーボードによる操作）。

回避文字列もキーボードから直接入力したり編集することができます。その方法は基本的にコード名の場合と同じです（「コード名の作成」）。ただし、あらかじめ対応する抽出文字列を選択し、半角の「>」か「<」に続けて回避文字列を入力して下さい。

回避文字列表示用ボックスには、回避文字列が文字コードの順に表示されます。これは同じ文字列（または先頭部分が共通の文字列）を重複して登録してしまわないようにする

ためです。

3.5.3. コード名の編集

もしコード名を打ち間違えたり、後で名前を変えなくなったら、ボックスに表示されているコード名をダブルクリックしてください。すると入力用ボックスにそのコード名が表示され、「 」ボタンが「変更」に変わります。名前を変えたらその「変更」ボタンをクリックしてください。名前を変える前に他のコード名ボックスを操作すると「変更」が「 」に戻ってしまいます。名前を変更する際に、名前をすべて消去してしまうと「変更」が「 」に戻ってしまいます（ 既知の問題点）

コード名は分析者が作業しやすいように、また他の人が理解しやすいように、自由に並び替えることができます。

コード名の並び替えは、コード名をマウスでドラッグするだけです。ドロップした箇所にあるコード名の前にドラッグしたコード名が移動します。もちろん、そのコード名に割り当てられている抽出文字列と回避文字列も同時に移動します。（未定義）コードは移動することができません。

3.5.4. 抽出文字列の割り当て変更

抽出文字列を間違ったコード名に割り当ててしまった場合や、「(未定義)」コードを利用した場合、あるいはコーディングルールの構造を変更したい場合には、抽出文字列の割り当てを変更することができます。

その方法は簡単です。割り当てを変更したい抽出文字列を、変更先のコード名にドラッグするだけです。

このような方法で、自由にコードルールを変更することができます。

3.5.5. コード名、抽出文字列、回避文字列の削除

コード名、抽出文字列、回避文字列はいずれも削除することができます。削除したい項目を選択し、それぞれが表示されているボックスの下の「×」ボタンを押します。確認のウィンドウが表示されるので、「OK」を押すと削除されます。いったん削除すると復活させることはできませんので注意してください。コード名を削除した時はそれに対応する抽出文字列と回避文字列が、抽出文字列を削除した時には対応する回避文字列が、それぞれすべて削除されます。

3.5.6. コードや文字列の検索

コード検索

あるコード名がどの回答文に割り当てられているかをチェックするには、「コード検索」を用いて下さい。検索したいコード名を選択して、左下の「コード検索」という文字の上下にあるボタンを押すと、そのコードが割り当てられているテキストを検索して表示します。検索はひとつ前または後のテキストから行いますので、現在表示されているテキストは検索しません。

文字列検索

特定の文字列を検索するには「文字列検索」を用いてください。右上の「文字列検索」という表示の下にあるボックスに検索したい文字列を入力してその上下のボタンを押します。これも現在表示されているテキストは検索しません。

抽出文字列の検索

抽出文字列を検索するには、検索したい抽出文字列を選択し、抽出文字列ボックスの上の「 Search」ボタンを押します。すると「文字列検索」ボックスにその抽出文字列が入力されますので、文字列検索で前後の検索を行うことができます。

3.6. コーディングルールファイルのフォーマット

AUTOCODE で使用するコーディングルールファイルはテキスト形式ですので、ワープロやエディタで編集することもできますし、それを印刷して第三者などに示すこともできます。

コーディングルールファイルのフォーマットは以下のようになっています。

(未定義)

未定義の抽出文字列 1

未定義の抽出文字列 1 の回避文字列 1

未定義の抽出文字列 2

*

コード名 1

コード名 1 の抽出文字列 1

コード名 1 の抽出文字列 1 の回避文字列 1

コード名 1 の抽出文字列 1 の回避文字列 2

コード名 1 の抽出文字列 2

コード名 1 の抽出文字列 2 の回避文字列 1

コード名 1 の抽出文字列 3

*

コード名 2

コード名 2 の抽出文字列 1

コード名 2 の抽出文字列 1 の回避文字列 1
コード名 2 の抽出文字列 2
コード名 2 の抽出文字列 3
*

それぞれのコード名のブロックは全角の「*」で区切られています。まず最初にコード名が書かれ、それに続いて対応する抽出文字列が順に並びます。回避文字列はそれが対応する抽出文字列の直後に順に書かれます。回避文字列の先頭は「>」か「<」になっているので区別がつかます。

なお、コーディングルールファイルを他のプログラムで編集して再度 AUTOCODE に読み込む際には、「(未定義)」というコード名とそれに対応する抽出文字列、回避文字列をすべて削除してください。そうすれば抽出文字列と回避文字列のソートを自動的に行ってくれます。

「(未定義)」を残したままだとソートが行われませんので、もし抽出文字列や回避文字列を変更していると、AUTOCODE の動作がおかしくなります。

4. コーディング結果の利用

4.1. SPSSでの利用

SPSS で AUTOCODE のコーディング結果を利用するには、AUTOCODE が出力したデータファイルと SPSS シンタックスファイルを利用します。このファイルを SPSS で読み込んで実行すると、自由回答部分についてはすぐに統計計算を行えます。

「出力」メニューの「SPSS データファイルの出力」を選んでください。まずデータファイル、次にシンタックスファイルのファイル名を入力してそれぞれ出力します。データファイルのデフォルト拡張子は「.dat」、シンタックスファイルは「.sps」です。

データファイルは固定カラム形式です。それぞれの変数のカラム位置はシンタックスファイルを参照してください。

変数名は、ID 番号が「ID」、自由回答の長さ(半角換算)が「LENGTH」、それぞれのコードが「FREEXXX」(XXX は 0 0 1 からの通し番号)となっています。コードの値は見つかった場合に「1」、見つからなければ「0」とコードされています。

自由回答と他の部分のデータを結合して分析するには、自由回答部分のデータをいったん保存し、他の部分のデータとマージします。その方法は SPSS のマニュアルを見てください。

4.2. その他のプログラムでの利用

SPSS 以外でも固定カラム形式やタブ区切り形式のデータを読み取れるプログラムなら、AUTOCODE の出力を簡単に利用できます。

固定カラム形式

SAS など固定カラム形式のデータを読み取れるプログラムで利用するには、「出力」メニューの「SPSS データファイルの出力」を選択し、データファイルと SPSS シンタックスファイルを出力します。SPSS シンタックスファイルをエディタなどで読むと、そこにどのカラムがどの変数と対応しているかが書かれていますので、それを参考に読み取りプログラムを作成してください。

タブ区切り形式

表計算プログラムやデータベースプログラムなど、タブ区切り形式のデータを読み取れるプログラムで利用するには、「出力」メニューの「タブ区切りファイルの出力」を選択します。

タブ区切りファイルは ID 番号、自由回答の長さ（半角換算）、各コードの順にタブで区切られ、それぞれのレコードは改行で区切られています。コードの値は見つかった場合に「1」、見つからなければ「0」とコードされています。

1 行めは見出し行で、ここにコード名が出力されています。見出し行の各フィールドは二重引用符（"）で囲まれています。デフォルトの拡張子は「tab」です。

5. 統計計算

5.1. 統計計算について

AUTOCODE は単純集計といくつかの簡単な統計計算を表示 / 出力することができます。

しかし、統計計算は「おまけ」のようなものなので、その精度については保障しかねます。あくまでも目安としてお使いください。

5.2. 単純集計

単純集計を出力するには、「統計」メニューの「単純集計」を選んでください。すると単純集計が表示されます。データの大きさとコンピュータの計算能力によっては少し待たされます（メッセージはありません）が、いくら長くても 10 数秒くらいだと思います。

最初に表示されるのは、コード名順の表示ですが、「度数順」ボタンを押すと、度数の大きいものから順に表示することもできます。

テキストファイル形式でファイル出力することもできます。

「戻る」ボタンで戻ります。

5.3. 関連度

AUTOCODE はカテゴリー間の関連の強さの度合を計算して表示します。必要ならファイル出力もできます。

「統計」メニューから「関連度」を選んでください。

関連の強さの指標としては、相関係数と修正相関係数をサポートしています。

相関係数

ピアソンの積率相関係数です。2 × 2 表ではこれは四分点相関係数と一致しますので、実際の値は四分点相関係数の計算式で求めています。四分点相関係数の計算式は、2つのコードが共にあてはまる場合を a、どちらかがあてはまる場合を b、c 共にあてはまらない場合を d とすると、

$$(ad-bc)/\sqrt{(a+b)(c+d)(a+c)(b+d)}$$

です。

a=d=0 の場合に最小値 - 1 を、b=d=0 の場合に最大値 1 をとります。

もしまるめた後の値が ± 1 ならば、表示は「++」または「--」となります（まずあり得ないことですが）。

修正相関係数

「修正相関係数」という名前は私が勝手につけたものです。特に理論的な根拠もないので（たぶん）AUTOCODE が出力していることを根拠にして論文などで用いないでください。

相関係数の最大・最小値は ± 1 ですが、周辺度数を固定すると必ずしもこの値にはなりません。特に、2つの変数の周辺分布が著しく異なる場合には、相関係数の最大・最小値の絶対値はかなり小さくなります。例えば片方が 10 と 990、もう片方が 100 と 900 という周辺度数の場合、周辺度数を固定したときの相関係数の最大値は 0.3 程度です。

ということは、あてはまるケース数が大きく異なる2つのコードの間の相関は低く見積もられてしまっているのではないかと考えられます（たぶん）。

そこで、相関係数に以下のような修正を加えてみました。

$$\text{修正相関係数} = \text{実際の相関係数} \div \text{周辺度数を固定したときの最大または最小の相関係数}$$

これを出力してみれば分かることですが、両方がコードされているケースがゼロであれば、修正相関係数は - 1 になります（「--」と表示される）。また片方だけがコードされているケースのうちどちらかがゼロ、すなわち片方がもう片方に「含まれている」場合は修正相関係数は 1 になります（「++」と表示される）。

まあ、これは半分お遊びみたいなものですから、あまり深く考えないでください。

5.4. 関連グラフ

「関連グラフ」とは、コード間の相関係数を視覚的にわかりやすく表示したグラフです。コード名の番号を円周上に等間隔に表示し、相関係数がある基準値以上になる組み合わせに線を引くというものです。まあこれも半分ぐらい「お遊び」だと思ってください。

「関連グラフ」を表示するには、「統計」メニューから「関連グラフ」を選んでください。すると別のウィンドウが開きますので、基準値をセットして「実行」ボタンを押します。

右のグラフの数字と左下のボックスの数字が対応していますので、それでコード名を確認してください。

基準値は0から1まで（両端は含まず）の値を入力することができます。基準値をいろいろと変えてグラフの変化を楽しんでください。

今のところウィンドウのサイズは変えられないようになっています。そのため、あまりコード名の数が多くなると何だか分からない表示になってしまいます。50 くらいが限界でしょうか。

6. その他

6.1. キーボードによる操作

AUTOCODE の操作の一部は、キーボードからも行えるようになっています。

ただし、現バージョンでは、キーボードでできるのはごく一部の操作だけであり、またキーアサインも確定していません。キーアサインについては、ご意見があればぜひ反映させたいと思います。

抽出文字列の割り当て

おそらく AUTOCODE によるコードルールの作成でもっとも時間がかかるのがこの作業だと思いますので、これに関連する操作をまず優先しました。以下、「C t r l +」は C t r l キーと同時に押すという意味です。

一つ前のテキストへ：	C t r l + 「Y」
一つ後ろのテキストへ：	C t r l + 「B」
最初のテキストへ：	C t r l + 「Home」
最後のテキストへ：	C t r l + 「End」
コード名の選択を一つ前へ：	C t r l + 「U」
コード名の選択を一つ後ろへ：	C t r l + 「N」
抽出文字列の選択を一つ前へ：	C t r l + 「I」
抽出文字列の選択を一つ後へ：	C t r l + 「M」
選択した文字列を抽出文字列にする：	C t r l + 「J」
選択した文字列を回避文字列（前置き）にする：	C t r l + 「K」
選択した文字列を回避文字列（後置き）にする：	C t r l + 「L」

ボックスへの直接入力

「B T M」ボタンの下のテキスト番号を入力するボックスと、下の3つの編集用ボックスに入力した後「E n t e r」キーを押しても警告音が鳴るだけで何も起こりません。これは今のところ回避できないのですが、「T a b」キーで移動する順序を整理しましたので、これらのボックスに入力した後「T a b」キーを1回押すと関連するボタンにフォーカスが移動するようになりました。

つまり、入力した後、「T a b」キー、「E n t e r」キーの順で押すとその結果が反映されます。

6.2. 各種制限

現バージョンではコードの数などに以下のような制限があります。

- ・テキストファイルの最大ケース数： 5,000
- ・コード名の最大数： 100 ((未定義)も含む)
- ・1つのコード名に対応する抽出文字列の最大数： 500
- ・1つの抽出文字列に対応する回避文字列の最大数： 100

この制限が実際に問題になるケースはあまりないと思いますが、もしあれば簡単な修正で制限を緩和できますので、遠慮なく作者までご連絡ください。

1ケースの回答文の長さに制限はありませんが、半角換算で10,000字以上になるとSPSSシンタックスファイルの出力が正常に行われません。

コード名の長さに制限はありませんが、あまり長いとSPSSシンタックスファイルを利用する際にSPSSのラベルの長さの制限を超えてしまう可能性があります。

抽出文字列、回避文字列の長さには制限がありません。

6.3. 既知の問題点

コード名や文字列の変更

コード名や文字列を変更しようとして該当する部分をダブルクリックし、ボックスの中の文字を編集しているときに、文字をすべて削除してしまうと挿入モードに戻ってしまいます。すべて削除しないように操作してください。

同じ抽出文字列が2つ以上ある場合

1つの回答文に同じ抽出文字列が2つ以上ある場合は、最初のものしか青色で表示されません。これは表示上の問題であって、コーディング結果には全く影響はありません。

6.4. DOS版との互換性

Windows版AUTOCODEのコーディングルールファイルはDOS版との互換性がある程度あり

ます。

DOS 版のコーディングルールは、「#」「*」「@」を使ったルールを用いていない限りそのまま使えます。DOS 版のコーディングルールを読み込むと抽出文字列と回避文字列を自動的にソートし、コード名の最初に「(未定義)」を付け加えます。

「#」「*」「@」を使ったルールを用いていても読み込み自体は問題なく行われますが、これらは単なる文字列の一部として扱われ、コードとしての機能は果たしません。

また Windows 版のコーディングルールファイルを DOS 版で使う場合には（そんな人が居るとは思えないが）、エディタなどで「(未定義)」を削除してください。

6.5. プログラムの使用条件等

著作権

本プログラムの著作権は佐藤裕が所持しています。

使用条件

AUTOCODE はフリーウェアです。特に使用条件はありません。ただし、AUTOCODE を用いてデータ解析をした結果を発表する際は、そのことを明記してください。

また、その場合は作者に連絡してもらえると大変ありがたいです。

6.6. サポート

サポートといっても素人プログラマなのでどの程度のことができるか心許ないですが、何か問題が生じた場合には、作者まで連絡いただければできる限りのことはしようと思えます。

作者連絡先

富山大学人文学部 佐藤裕
E-MAIL: satoh@hmt.toyama-u.ac.jp

AUTOCODE の最新版は以下のホームページから入手できるようにしておくつもりです。

<http://jinbun1.hmt.toyama-u.ac.jp/socio/satoh/autocode/index.html>

また、上記のホームページでは AUTOCODE に関する情報の提供も行っていきたいと思えます。

6.7. 履歴

1998.4.9 Ver. 0.20b
最初の公開版
1998.4.10 Ver. 0.21b

コード名を追加したときの表示の乱れを修正

1998.4.12 Ver. 0.22b

コード名と抽出文字列を追加した時の表示の乱れを修正

見えない範囲へのドラッグをスムーズに

「×」ボタンでの終了でも保存確認を

1998.4.15 Ver. 0.30b

キーボードによる操作を部分的に可能にした

「関連グラフ」の新設

1999.2.27 Ver. 1.00

度数順の単純集計を新設

タブ区切りファイルの出力を新設

「終了」ボタンを削除

その他バグ修正

1999.6.24 Ver. 1.01

単純集計のファイル出力を追加