2016 年 9 月
株式会社ライトストーン

Stata14 における文字のエンコード形式の変更について

Stata 14 では、拡張 ASCII 文字のエンコード形式に変更があります。

同変更により、Stata13 やそれ以前の Stata で保存したファイル（拡張子が dta、do、ado、smcl、gph、stsem、stpr のファイル）について、拡張 ASCII 文字が含まれる場合、Stata14 では表示の文字化けが発生します。文字化けが発生すると、do ファイルなどは正常に動作しないことがあります。この文字化けを回避するには、エンコード形式が UTF-8 である新たなファイルへデータを変換する必要があります。変換方法については、一つの方法を本資料に記述します。

Stata で、ASCII 文字すなわち半角英数のみを使用している場合、この変更による影響は特にありません。引き続き以前の Stata のデータを使用できます。

変更履歴

日付	変更内容
2015.5	初版
2016.9	表題を含むフォーマットの変更。 変換が必要なものと不要なものを明記。 テキストファイルのインポート・エクスポートのコマンドを修正。 テキストファイルのインポート時にエンコード形式を指定する方法を追記。

背景

Stata 14 では、前バージョンである 13 からの仕様変更の一つとして、ASCII 文字（半角の 0-9、A-Z、a-z、および>.+-などの記号）以外のいわゆる拡張 ASCII 文字のエンコード形式を、それまでのプラットフォーム依存でなく、新たに Unicode(UTF-8)へ統一する変更が含まれております。同変更により、Stata13 やそれ以前の Stata で保存した dta ファイル（拡張子が dta のファイル）、do ファイル（同）、ado ファイル（同）について、拡張 ASCII 文字が含まれる場合、Stata14 では表示の文字化けが発生します。これらの文字を正しく表示するには、エンコード形式が UTF-8 である新たなファイルへデータを変換する必要があります。

Stata で、ASCII 文字すなわち半角英数のみを使用している場合、この変更による影響は特にありません。引き続き以前の Stata のデータを使用できます。

Excel ファイルをインポートする場合、この変更による影響は特にありません。

テキストファイル（拡張子が txt や csv などのファイル）をインポートする場合、インポートする際に読み込むエンコード形式を指定できます（詳しくはこちらを参照ください）。事前の UTF-8 形式への変換は必要ありません。

変換が必要なものと不要なものについてはこちらを参照ください。

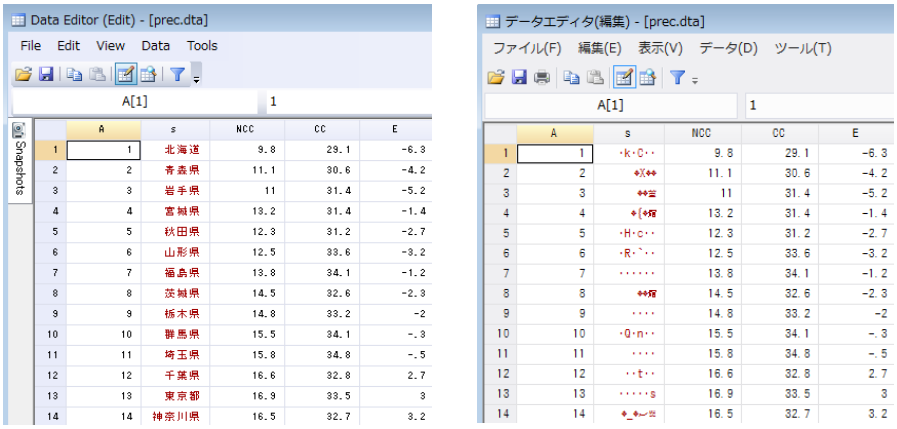


図 1. Stata13 で保存した日本語文字（左）は、Stata14 でファイルをそのまま開くと文字化けします（右）

一般的に、拡張 ASCII 文字は、保存されたコードからは使用したエンコード形式を明確に判別できないこともあり、すべて自動で完璧に行うことができません。また、文字コードの衝突などから、ここで説明する方法を用いても、変換が完全には行えない可能性があります。

変換が必要なもの

1. **Stata13**、またはそれ以前の Stata で保存した以下のもので、

   拡張 ASCII 文字（日本語フォントなど）が含まれるもの

   dta ファイル
   do ファイル
   ado ファイル
   smcl ファイル

   （※gph ファイル、stsem ファイル、stpr ファイルは変換できません。何卒ご了承ください。）
   （※ひとたび変換をおこなうと、Stata13、またはそれ以前の Stata では文字化けの生じるファイルとなります。）

変換が不要なもの

1. Stata13、またはそれ以前の Stata で保存した以下のもので、

   拡張 ASCII 文字（日本語フォントなど）が含まれないもの

2. Stata14、またはそれ以降※で保存した以下のもの

   （本資料の作成時点では、Stata14 より新しいバージョンはありません）

3. インポート前の Excel ファイル、csv ファイル、テキストファイル

「背景」に戻る

UTF-8 への変換法

本資料では、Stata14 に用意されたツールを使用した変換法を説明します。拡張 ASCII 文字を UTF-8 へ変換する方法は幾通りも考えられ、本資料で説明する方法がただ一つの方法という訳ではありません。

なお、Stata14 以外による変換については、こちらを参照ください。

Stata 14 で変換を行う方法

Stata 14 には拡張 ASCII 文字を Unicode に変換するための新たなコマンド `unicode` が用意されています。変換作業の大まかな流れは以下です。（詳細は、`help unicode translate` をご覧ください。）

0. 前準備
1. `unicode analyze` による分析
2. `unicode encoding set` による読み込み形式の設定
3. `unicode translate` による変換
4. 変換の検証

作業の一助となる操作の一覧は以下です。

A. ファイルを変換前の状態へ戻す
B. データを変更せずその他の情報のみ変換する
C. バックアップファイルを削除する
D. ログの開始/停止/表示
E. テキストファイルへのエクスポート
F. テキストファイルのインポート

0. 前準備

変換をする前に、以下の2つを実施してください。

①メモリ上のデータのクリア

Stata14 で既にデータセットを開いている場合、必要があれば保存し、その後 `clear` コマンドを実行してメモリ上から一掃します。

```
clear
```

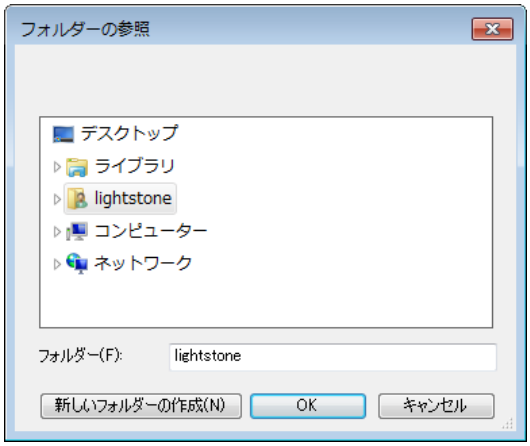
②作業フォルダの移動

変換の対象となるファイルが現在の作業フォルダに存在しない場合、変換の対象となるファイルがあるフォルダまで作業フォルダを移動してください。作業フォルダの移動は、以下のコマンドを実行するか、またはメニューから操作します。

`cd` *変換の対象となるファイルがあるフォルダ名*

または、

[ファイル]－[作業フォルダの変更...]を選択して以下のようなダイアログを開き、変換の対象となるファイルのあるフォルダを選択



1. `unicode analyze` による分析

変換の対象となるファイルを分析します。Stata のコマンドウィンドウで、以下を実行してください。

`unicode analyze` *ファイル名*

*ファイル名*には、変換の対象となるファイルを.dta や.do、.ado などの拡張子付きで指定します。二重引用符(`" "`)で囲んでも問題ありません。ただし、別のフォルダにあるファイルは指定できません。

例として、prec.dta というファイルを指定して実行した結果は、以下です。

```
. unicode analyze prec.dta

  File summary (before starting):
        1  file(s) specified
        1  file(s) to be examined ...

  File prec.dta (Stata dataset)
       11 variable labels need translation
        1 str# variable needs translation
          ----------------------------------------------------------------------
          File needs translation.  Use unicode translate on this file.

  File prec.dta needs translation

  File summary:
        1 file(s) need translation

.
```

上記の結果のように赤字で「File needs translation.」と表示された場合、UTF-8 で未定義のコードが発見されたことなどを理由に、変換が必要と判定されたことになります。水平線より上にある記述は、詳細な分析結果です。上の例では、「11 variable labels need translation」と「1 str# variable needs translation」とあり、11 個の値ラベルと 1 個の str#型の変数に変換が必要であることが示されています。

一方でもし、以下の実行結果のように、「does not need translation」と表示された場合、ファイルを変換する必要はありません。

```
. unicode analyze auto.dta

  File summary (before starting):
        1  file(s) specified
        1  file(s) to be examined ...

  File auto.dta (Stata dataset)
          ----------------------------------------------------------------------
          File does not need translation

  File summary:
      all files okay

.
```

上記の unicode analyze を実行すると、作業フォルダに bak.stunicode というフォルダが新たに作成され、ここに分析結果が保存されます。また同フォルダには、変換の際、変換前のファイルがバックアップとして保存されます。

なお、unicode analyze は一度に複数のファイルを指定したり、*を用いた形で指定したりすることもできます。

「1. unicode analyze による分析法」の先頭に戻る

「Stata 14 で変換を行う方法」の先頭に戻る

2. unicode encoding set による読み込み形式の設定

次に、ファイルの読み込みに用いる文字コードの形式を指定します。前述のように、ファイルに保存された文字コードそのものから使用された形式を判定することは不可能です。しかし、日本語でよく使用される形式は限られています。日本語における Unicode 以外の代表的なものは以下です。

Shift_JIS	Windows で用いられる形式。Windows-932 などもこれに類似。
EUC-JP	Unix で用いられる形式。
JIS	電子メールで用いられる形式。ISO-2022-JP はこれに同じ。

なお、Mac では標準で Unicode を用いています。ファイルで用いられた形式が正確に分からない場合、ひとまず上記のどれかであると当たりをつけます。

文字コードの形式は、以下で設定します。

```
unicode encoding set "形式名"
```

"*形式名*"は、"Shift_JIS"、"EUC-JP"、"JIS"など、二重引用符付きで入力します。大文字と小文字の区別はありません。また、アンダーバー(_)とハイフン(-)の区別もありません。

例として、"Shift_JIS"への設定を行った結果は以下です。

```
. unicode encoding set "Shift_JIS"
  (default encoding now Shift_JIS)

.
```

なお、Stata14 でサポートしている文字コードの一覧を表示するには、help encodings を実行します。

「2. unicode encoding set による読み込み形式の設定」の先頭に戻る

「Stata 14 で変換を行う方法」の先頭に戻る

3. unicode translate による変換

次に、変換を実施するため以下のコマンドを実行します。

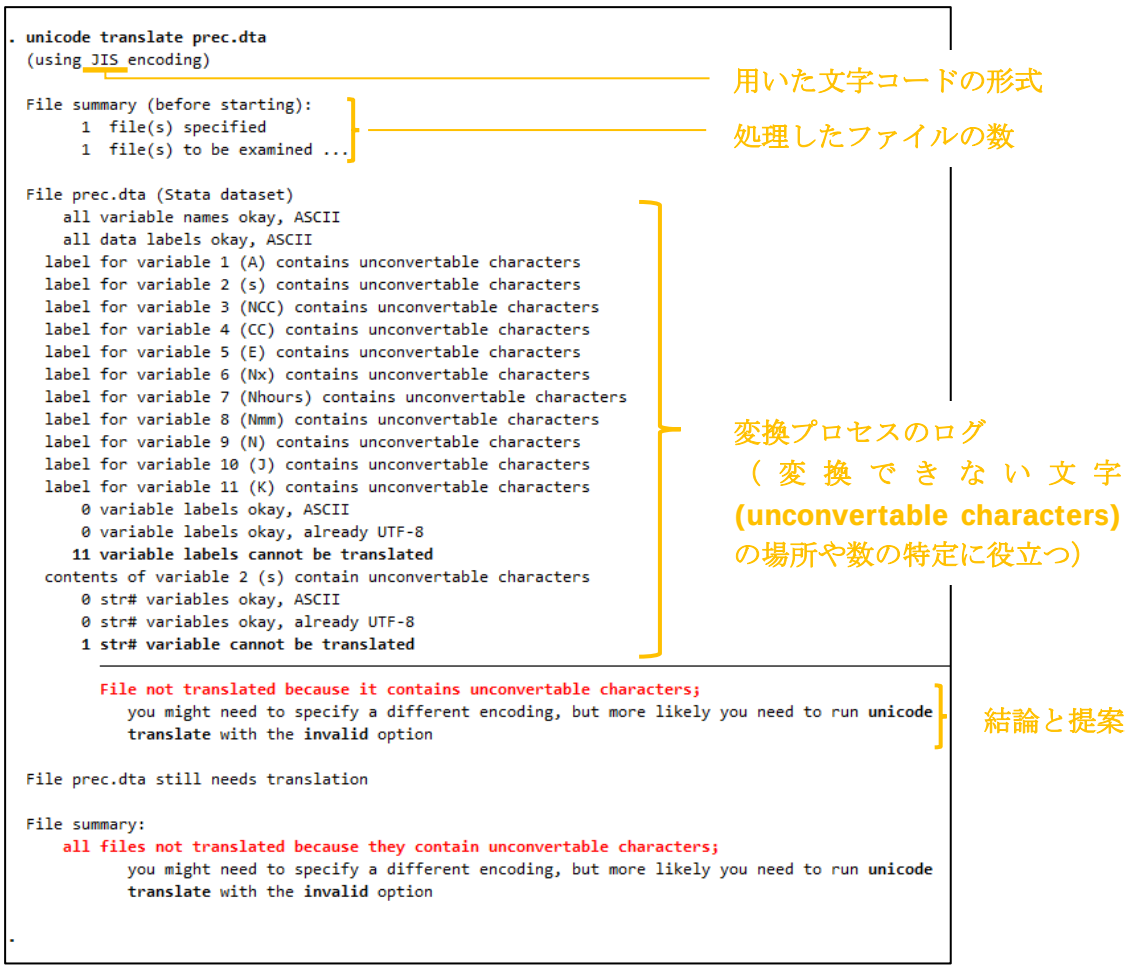
```
unicode translate ファイル名
```

先ほど分析したファイルを指定して実行すると、次のような結果が出ます。

```
. unicode translate prec.dta
  (using Shift_JIS encoding)

  File summary (before starting):
        1  file(s) specified
        1  file(s) to be examined ...

  File prec.dta (Stata dataset)
      all variable names okay, ASCII
      all data labels okay, ASCII
      all variable labels translated
      all str# variables translated
          ------------------------------------------------------------
          File successfully translated

  File summary:
      all files successfully translated

.
```

実行結果で、水平線のすぐ下に「File successfully translated」と表示された場合、無事に変換が行われています。水平線より上の記述は、変換プロセスの詳細なログです。たとえば、「all variable names okay, ASCII」からは、「すべての variable name（変数名）が ASCII 文字」であったことが分かります。ASCII 文字はそのまま表示できるので、このコマンドでの変換は行われません。また、「all variable labels translated」からは、「すべての variable label（変数ラベル）が変換された」ことが分かります。変換した文字は、実際にファイルを開いて目で確認してください（「4. 変換の検証」へ移動する）。

一方、変換が失敗すると、次のような結果が出ます。



上記の結果のように赤字で表示された「File not translated」との記述がある場合、何らかの理由により変換は行われていません。赤字で続けて表示されている「because it contains unconvertable characters」は、その理由が「変換できない文字があったため」であることを示しています。水平線より上の記述は変換プロセスのログです。たとえば、「label for variable 1 (A) contains unconvertable characters」は、「変数 1（名前 A）の変数ラベルが変換できない文字を含んでいた」ことを示します。

上記のような失敗の結果は、今後の作業を効率化するために、メモ帳やログに保存し、いつでも見直せるようにすることをお勧めします。

変換が失敗する原因は様々です。今の場合のように変換できない文字が数多くある場合、

原因① 読み込みに使用した文字コードの形式が適切でなかった

ということが考えられます。この場合、別の形式へ設定することで問題が解決する可能性があります。「2. unicode encoding set による読み込み形式の設定」へ戻り、先ほどとは別の形式を選択してから、「3. unicode translate による変換」を実行します。文字コードの形式は数多くあり、他の形式に若干の追加を行ったもの、呼び方のみ異なるものなど様々あります。Stata14 で変換できる文字コードの形式の一覧を表示するには、help encodings を実行してください。

いくつかの形式を試しても、なかなか変換に成功する形式が見つからない場合、

原因② 一つの形式の文字コードでは絶対に変換ができない

ということが考えられます。`unicode translate` は読み込みに用いる形式は一度につき一つしか設定できません。この原因がある場合、一度変換を行った後、そこから更に変換を行うことで問題が解決する可能性があります。

変換できない文字があっても強制的に変換を行うには、以下を実行します。変換が実施されファイルの内容が置き換わる際、変換前のファイルが bak.stunicode に保存され、変換後においても `unicode restore` *ファイル名*を実行することにより元の状態に戻すことができます。ファイルを強制的に変換するには、以下のコマンドのうちどれかを実行します。

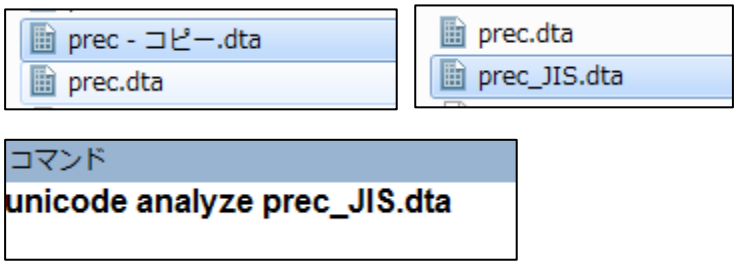
```
unicode translate ファイル名, invalid(mark)

unicode translate ファイル名, invalid(ignore)

unicode translate ファイル名, invalid(escape)
```

`invalid(mark)`を指定すると、変換できない文字は Unicode で公式の置き換え用の文字（Replacement character、U+fffd）で置き換えて変換を実施します。Replacement character は□や  という表示になります。

`invalid(ignore)`を指定すると、変換できない文字は削除して変換を実施します。

`invalid(escape)`を指定すると、変換できない文字は%X##というエスケープシーケンスコードで置き換えて変換を実施します。##には変換できなかった文字の 16 進数コードです。

変換後は、実際にファイルを開き、文字化けしていない文字が見られるかを確認してください。全く見られないようであれば、再び文字コードの形式の設定からやり直します。正しい文字が見られ、部分的にでも変換が成功しているようであれば、ファイルのコピーを作成し、必要であれば半角英数のみの名前にファイル名を変更した後、そのファイルについて、再び「1. `unicode analyze` による分析法」から実行します。あるいは、変換できない文字の数が少数である場合、無理に対応形式を探さず、データエディタなどから手動で修正する方法も考えられます。



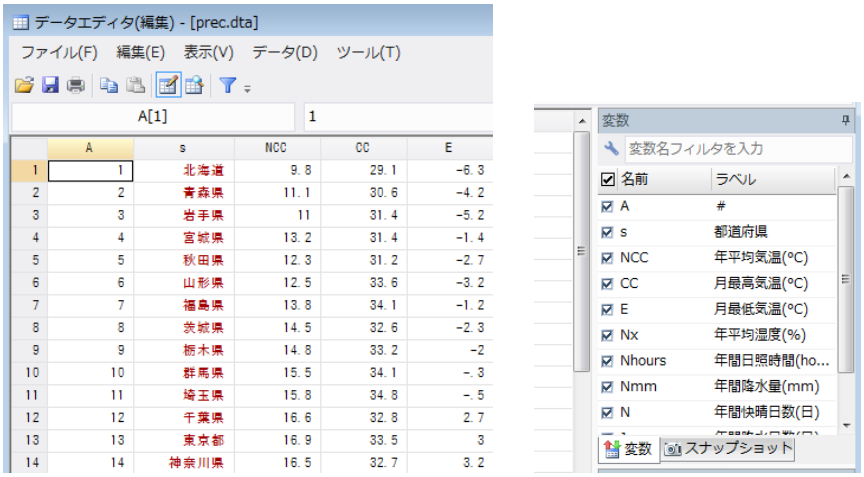
「3. unicode translate による変換」の先頭に戻る

「Stata 14 で変換を行う方法」の先頭に戻る

4. 変換の検証

変換が成功したら、実際に Stata14 で開いてみて、本当に正しい変換が行われたかを確認します。

```
. use prec.dta

. edit

.
```



目で確認して文字化けがなさそうであれば、正しい形式を選択して変換を行ったと考えられます。

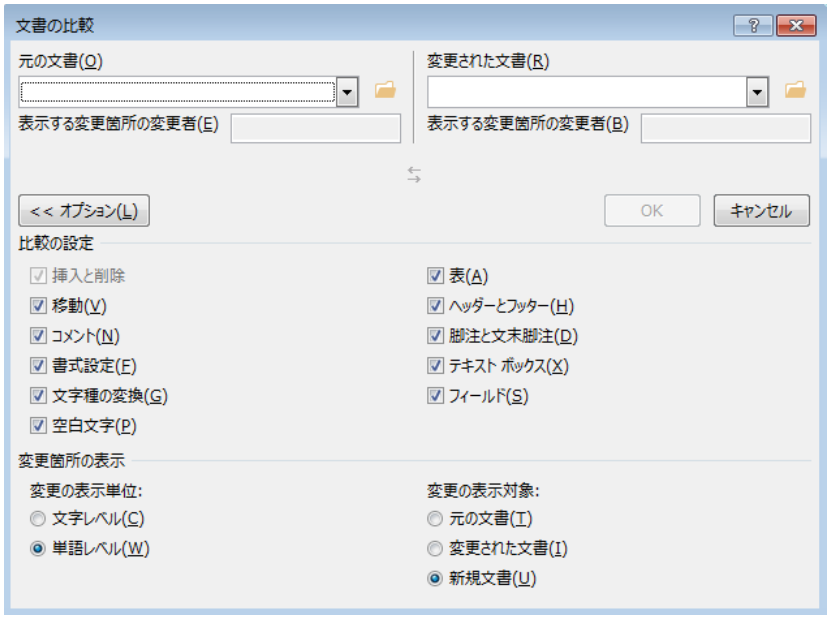
全体的に文字化けが見られるようであれば、「2. unicode encoding set による読み込み形式の設定」へ戻り、形式の設定からやり直してください。

変換が成功した場合でも、本当に変換が正しく行われたのかの確認は行いたいところです。変換の正当性の検証は、別の方法による変換結果と比較する方法が考えられますが、残念ながら Stata のデータセット形式を認識した上で拡張 ASCII 文字の変換を行える別の方法は存在せず、一刀両断的に結論が与えられるような検証方法がありません。ただし、ある程度範囲を絞った上で、条件付きで検証を行う方法が考えられます。以下は、そのうちのいくつかです。

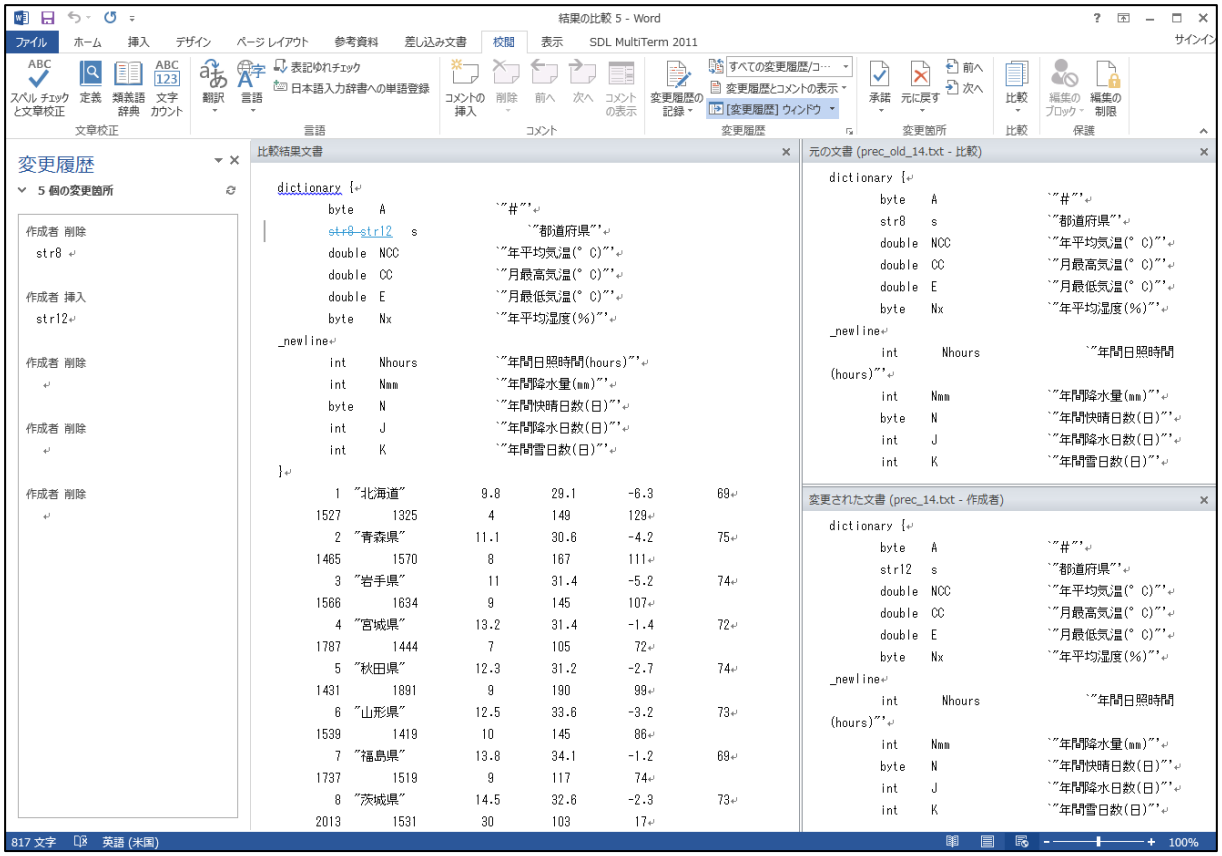
検証法① 変換前と変換後の.dta ファイルをテキストデータに出力して比較する
検証法② 変換前と変換後の.dta ファイルを以前の Stata と Stata14 で開き、文字を表示させた状態をスクリーンショットで保存するなどして、両者の文字を目で見て比較する

検証法③ 変換前と変換後の.dta ファイルについて、summarize コマンドを実行し、結果を比較する

以下、検証法①のみを取り上げて説明します。まず、Stata で.dta ファイルを開き、テキストデータへ出力します。テキストファイルへのエクスポートの方法については、テキストファイルへのエクスポートをご覧ください。出力したファイルは、Microsoft 社の Word 文書で「比較」という機能を使って比較できます。Word 2013 では、メニューから[校閲]－[比較]－[比較]を選択すると、以下のようなダイアログが開きます。



左上にある[元の文書]でボックスの右にあるフォルダマークをクリックし、比較するテキストファイルのうちの一つを開きます。エンコードを選択するウィンドウが出る場合、そのままの状態で OK を選択します。次に同様に、右上にある[変更された文書]でボックスの右にあるフォルダマークをクリックし、比較するテキストファイルのうちのもう一つを開きます。すると次のような画面になります。



上の例では、変換での変更になった点が、エクスポート上の仕様によるわずかなスペースの違いを除けば、変数 s の型が str8 から str12 に変更されたのみであることが分かります。ちなみに、この変更はエンコード形式の違いにより必要なバイト数が変わったためであると考えられます。より重要なのは、両者のファイルで日本語の文字にも ASCII 文字に違いが見られないという結果が得られたことであり、これにより、少なくとも変数の値、変数名、変数ラベルにおいては、変換が正しく行われたことが検証されたと言えます。

「4. 変換の検証」の先頭に戻る

「Stata 14 で変換を行う方法」の先頭に戻る

A. ファイルを変換前の状態へ戻す

一度変換したファイルを、変換前の状態に戻すには、以下のコマンドを実行します。

```
unicode restore ファイル名
```

「Stata 14 で変換を行う方法」の先頭に戻る

B. データを変更せずその他の情報のみ変換する

変数の値として格納された文字は、可読性よりもコードの値が一定であることが重要になる場合があります。変数の値に含まれた文字を変換せず、変数ラベル、値ラベルなどのその他にある文字列のみを変換するには、以下のコマンドを実行します。

```
unicode translate ファイル名, nodata
```

「Stata 14 で変換を行う方法」の先頭に戻る

C. バックアップファイルを削除する

変換の実施により作成されたファイルを削除するには、以下のコマンドを実行します。

```
unicode erasebackups, badidea
```

「Stata 14 で変換を行う方法」の先頭に戻る

D. ログの開始/停止/表示

ログを開始するには、以下のコマンドを実行またはメニューを選択します。

```
log using ファイル名
```

[ファイル]－[ログ]－[開始]を選択してファイル名を指定

ログを終了するには、以下のコマンドを実行またはメニューを選択します。

```
log close
```

[ファイル]－[ログ]－[終了]を選択

ログを表示するには、以下のコマンドを実行またはメニューを選択します。

```
view ファイル名
```

[ファイル]－[ログ]－[開始]を選択してファイル名を指定

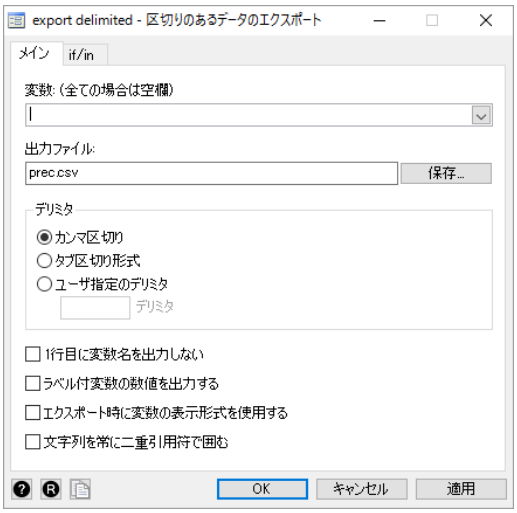
「Stata 14 で変換を行う方法」の先頭に戻る

E. テキストファイルへのエクスポート

まず、変換するファイルをテキストファイルへエクスポートします。Stata で変換するファイルを読み込んだ後、コマンドウィンドウで以下を実行するか、またはメニューから以下を選択してください。

```
export delimited ファイル名, replace [nolabel]
```

[ファイル]－[エクスポート]－[テキストデータ(デリメタ、.csv 等)]



*ファイル名*には出力先のファイル名を指定します。.csv や.txt などの拡張子付きで指定します。二重引用符(" ")で囲んでも問題ありません。ただ、別のフォルダにあるファイルは指定できません。nolabel を指定した場合、値ラベルを適用したデータについて、値ラベルでなく数値データが出力されます。指定しない場合、数値ラベルでなく、値ラベルが出力されます。たとえば、prec.dta を開いた後、prec.csv というファイルへエクスポートすると、以下のように、結果ウィンドウには特に表示は出ません。

```
. export delimited prec.csv, replace
(note: file prec.csv not found)
file prec.csv saved

.
```

「4. 変換の検証」の先頭に戻る

「Stata 14 で変換を行う方法」の先頭に戻る

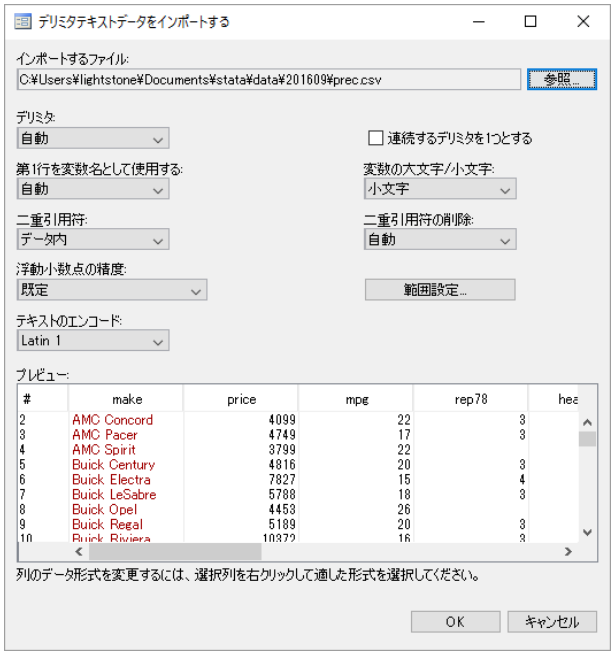
「Stata 14 以外での変換」の先頭に戻る

F. テキストファイルのインポート

次に、変換したテキストファイルを Stata にインポートします。コマンドウィンドウで以下を実行するか、またはメニューから以下を選択してください。

```
import delimited ファイル名, clear [encoding("エンコード形式")]
```

[ファイル]－[インポート]－[テキストデータ(デリメタ、.csv 等)]



*ファイル名*には入力するファイル名を指定します。.csv や.txt などの拡張子付きで指定します。二重引用符(" ")で囲んでも問題ありません。ただ、別のフォルダにあるファイルは指定できません。encoding("*エンコード形式*")を指定した場合、*エンコード形式*に入力した形式でファイルを読み込みます。たとえば、encoding("Shift_JIS")と指定すると、シフト JIS 形式で保存されたテキストファイルを文字化けなくインポートできます。メニュー操作をおこなう際、ダイアログに表れる「テキストのエンコード」では、"Latin 1"、"MacRoman"、"UTF-8"、"UTF-16"の 4 つのみしか指定できません。それ以外を指定する場合、コマンド操作をご利用ください。

「4. 変換の検証」の先頭に戻る

「Stata 14 で変換を行う方法」の先頭に戻る

「Stata 14 以外での変換」の先頭に戻る

Stata 14 以外での変換

文字の変換は、必ずしも Stata14 を利用して行う必要はありません。文字の変換を行う方法は何通りも考えられます。ただ、Stata14 以外での変換に当たっては、以下の点に留意する必要があります。

- .dta ファイルにおいては、Stata14 以外を用いて Stata 独自のデータセット形式を認識しながら行うことができません。したがって、変数の値、変数名、変数ラベル、値ラベルなど、すべての情報を失わずに変換を行うのは困難な作業になることが見込まれます。

- .do ファイルや.ado ファイルにおいては、それらが純粋にテキストファイルであるため、一般的なテキストエディタなどを用いて変換を行える可能性があります。

- Stata14 以外で変換を行う方法の一つとして、①.dta ファイルをテキストファイルへエクスポート、②テキストファイルを UTF-8 で保存、③保存したテキストファイルをインポート、という手順で実施する方法が考えられます。この方法は比較的短い作業で、かつ広く一般的に用いられている方法で変換を行えるという長所があります。一方で、前述のように変数の値と変数ラベル、変数名以外のデータが失われてしまいます（逆に言えば、.dta ファイルでそれら以外を利用していない場合は、極めて有用な手段になります）。

- テキストファイルを UTF-8 で保存する場合、テキストとして表示されない BOM というデータをファイルに含める方法と含めない方法の 2 通りが存在します。Stata 用に変換を行う場合、この BOM がない方法で保存が行われる必要があります。もし、BOM を含んだ方法で保存したファイルを Stata で利用しようとすると、ファイルがインポートできなかったり、コマンドが実行できなかったりします。Windows のメモ帳は UTF-8 での保存ができる大変便利なツールですが、BOM を含めない方法で保存することができないため、この変換には利用できません。

以下には、関連する機能についての説明です。

E. テキストファイルへのエクスポート
F. テキストファイルのインポート

「UTF-8 への変換法」の先頭に戻る