データ統合について (第二回)

株式会社ライトストーン

2015 年 12 月 1 日

前回に引き続き、異なる構造を持つデータを組み合わせて、一つの分析用ワークファイルを作る方法について解説します。前回は移植先のワークファイルの方がデータがリッチである場合を考えましたが、今回は逆に、移植元の方がデータがリッチである場合を考えます。例えば、都道府県レベルのワークファイルで分析を行っている際に、市町村レベルのワークファイルで各都道府県に属する市町村のデータを合算し、都道府県ワークファイルへ移植したいような場合です。
以下では表記の簡単化のために、分析に使っているワークファイルを**移植先ワークファイル**、追加したいデータが入っているワークファイルを**ソースワークファイル**と呼びます。

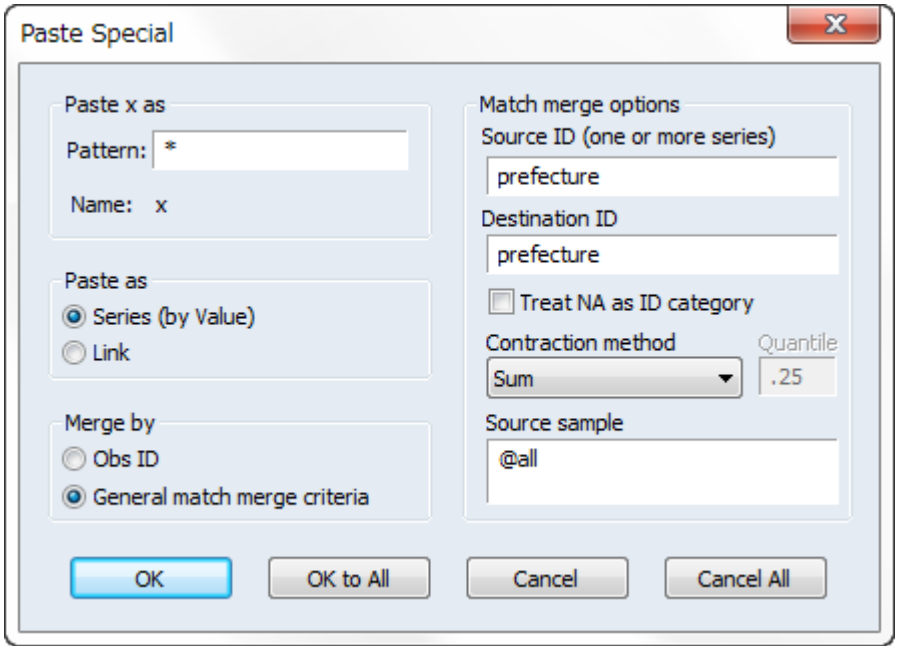
1 Many to one の場合

前回に引き続き、citydata.wf1 と prefecdata.wf1 をデータとして用います。一方で、**今回はそれぞれのワークファイルの役割を前回とは逆にします**。すなわち、移植先ワークファイルが prefecdata.wf1 で、ソースワークファイルが citydata.wf1 であると考えます。

例えば、citydata も prefecdata のどちらも、全国展開しているとあるコンビニエンスチェーンに関するデータであると想像してください。ここでは仮に、都道府県レベルの営業成績を比較したいため、一部のデータを都道府県毎に合算したいようなケースを考えます。例えば citydata 側の X が市町村単位の店舗人員数、Y が市町村単位の売上である場合に、各都道府県内で店舗人員数や売上の合計を求め、既に別のデータが存在する prefecdata 側に追加したいような場合です。このような場合も基本的な操作は前回と変わらないのですが、前回と比べて `Contraction method` の選択が重要性を増すことになります。

1.1 メニュー操作

何はともあれ、まずは和を取って移植する場合について実際の操作で体感してみたいと思います。操作は前回同様ですが、メニューで操作する方法を再度ご紹介します。それぞれのワークファイルが一度に開かれている状態で、citydata 側の X を prefecdata 側にドラッグアンドドロップするか、コピーアンドペーストしてください。すると、前回と同じダイアログが表示されますので、これも前回同様に General match merge criteria を選択してください。すると、ダイアログの右側が切り替わりますので、以下のように入力してください。



この状態で OK をクリックすると、データの移植が行われますので、確認してみます。

Tokyo	270
Saitama	210

とりあえず、何らかの数字が移植されたようです。そこで、計算が正しく行われているかを確認してみます。Citydata 側のデータは以下の通りです。

PREFECTURE	X
Tokyo	100
Saitama	90
Tokyo	80
Tokyo	90
Saitama	120

各都道府県毎に和を取ると、確かに数値は一致しますので、正しく移植が行われていることが確認できました。

上記は和を取ることが目的である場合の操作例でした。和以外を利用したい場合の操作方法はどうなるでしょうか。また、元データをそのまま移植することはできないでしょうか。これらはすべて、Contraction method の選択と関係しています。

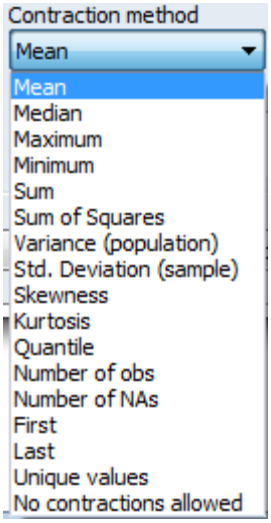
まず、元データをそのまま移植することから考えてみます。**上の画像二つをご覧いただければ分かるように、この場合、**Prefecture **の情報だけではソースワークファイルの“どの”データを移植してよいのかが分かりません。**たとえば Tokyo というキーだけが与えられている場合、100, 80, 90 という 3 つの移植候補があるため、この内どれを用いるべきなのかを EViews が判断できません。これが Many to one の Many の意味となります。**このような場合に元データをそのまま移植することはできません。**なんらかの Contraction か、あるいは分析目的/データの見直しと、それに伴う Souce ID/Destination ID の変更が必要になります。

一方で、和以外を用いることは容易に行えます。たとえば、上の操作例と同様に操作し、Contraction method を“Mean”にしてください。

Tokyo	90
Saitama	105

先ほどとは異なる値になっていることが分かります。手計算してみると、確かに平均になっていることが分かります。前回ご紹介した One to Many のケースではそれぞれの ID についてソースデータが一つしかないため、Sum や Mean など、ほとんどの Contraction method では結果的に No contraction allowed と同一の値（すなわち、ソースデータそのままの値）が移植されます。この意味で、Contraction method の選択はそれほど重要ではありませんでした。一方で Many to one の場合は、それぞれの ID についてソースデータが複数あるので、Contraction method の選択が本質的に重要となります。

EViews9 では、このデータの場合、Contraction method は以下からお選びいただけます[*1]。



ほとんどは名が体を表していますが、Unique values と No Contractions allowed の違いだけは分かりにくいのではないかと思います。これらは名前が示唆するようにほぼ同じものではあるのですが、No Contractions

[*1] なお、アルファオブジェクトの場合は、数字ではなく文字列なので利用できる Contraction method の種類が少なくなります。

allowed ではエラーが生じて移植が行われない一方、Unique values では移植が行われる特殊ケースがございます。つまり、以下のようなケースです。

PREFECTURE	S
Tokyo	150
Saitama	60
Tokyo	150
Tokyo	150
Saitama	60

このように、各 ID 毎に移植したいデータが同一の値を取っている場合、Unique values では移植が許可されます[*2]。一方で、No Contractions allowed ではこの場合も移植が認められません。

1.2 コマンド操作

コマンド操作の場合も、前回と操作はほぼ変わりませんので、まずは前回の内容をご確認ください。両方のデータを同じウィンドウ内で開いた状態で、まずは以下のコマンドをコピーアンドペーストして実行してみてください。

copy(c=none) citydata::citydata\x prefecdata::prefecdata\x @src prefecture @dest prefecture

エラーが生じることと思います。これは、`copy(c=none)` と、Contraction method として上記で説明した No Contractions allowed が選択されているためです。例えば、

copy(c=sum) citydata::citydata\x prefecdata::prefecdata\x @src prefecture @dest prefecture

というように指定すれば、エラーなしに（和の）移植が行われます。

c=オプションの指定として他にどのような値が使えるかについては、Copy コマンドのヘルプをご確認ください。

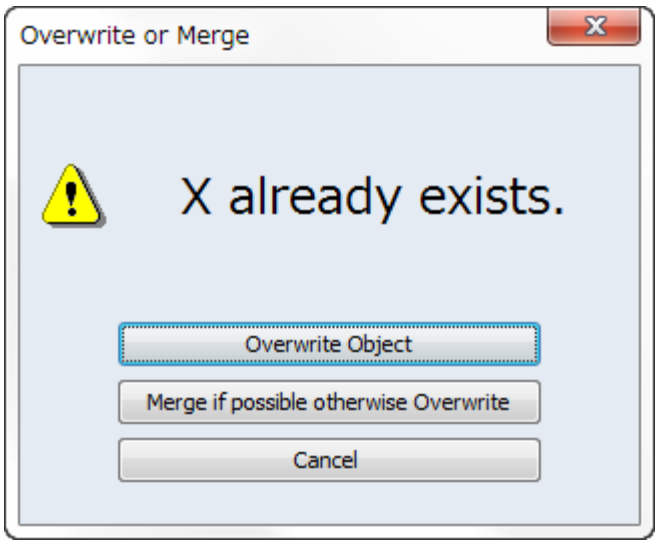
1.3 既に同名のオブジェクトが存在する場合

まずは以下のコマンドを続けて実行してみてください。

copy(c=sum) citydata::citydata\x prefecdata::prefecdata\x @src prefecture @dest prefecture
copy(c=mean) citydata::citydata\x prefecdata::prefecdata\x @src prefecture @dest prefecture

すると、2 番目のコマンドの実行の際に、次のようなダイアログが表示されます。

[*2] たとえば、Tokyo のいずれか一つの値だけが他と異なっている (Saitama の値は共通) ような場合も、Saitama も含め移植はまったく行われません



意味は読んで字のごとくで、既に X というオブジェクトが移植先ワークファイルに存在するので、どうするかを聞かれています。

最も簡単な解決策は、一度キャンセルして 2 番目のコマンドの `PREFECDATA::PREFECDATA\X` を `PREFECDATA::PREFECDATA\meanX` などとすることです (メニュー操作の場合は左上の Pattern 欄の*を消去し入力します)。この場合、移植先ワークファイルには meanX というオブジェクト名で移植されますので、上記の警告は表示されません。もちろん、名称は meanX でなくても構いません。和と平均を併存させて使いたいような場合に便利です。

`Overwrite Object` を選ぶと、移植先ワークファイルの既存オブジェクトは、新しい同名オブジェクトで上書きされます。既存オブジェクトを完全に置き換えることが可能です。

`Merge if possible otherwise Overwrite` を選ぶと、ソースワークファイル側に存在しないデータについては、移植先ワークファイルのデータがそのまま残ります。たとえば、データの入力状況が以下のような場合です。

Saitama	90
Saitama	120

ソースワークファイルの X

Tokyo	90
Saitama	NA

移植先ワークファイルの X

イメージとしては、元々東京のデータが入力されているワークファイルに、埼玉のデータを追加するようなケースです。

この場合、ソースワークファイルに Tokyo のデータがないので、Overwrite を選ぶと移植先ワークファイルの Tokyo の値が NA になってしまいます（例えば mean を選んだ場合は、順に NA, 105 となります)。別の

データを移植できるとしても、それで元々あるデータが消えてしまっては意味がありません。このような場合に Merge if を選択すると、ソース側に該当するデータがないものに関しては移植先のデータをそのまま用いることができます。この場合 Tokyo のデータがソースワークファイル側にないので、Saitama に対してのみ Contraction が行われます (mean を選んだ場合、順に 90, 105 となります)。ケースに応じて使い分けると便利です。

2 Many to Many の場合

上ではソースワークファイルだけが Many であるような状況を考えましたが、ここではソースワークファイルに加え、移植先ワークファイルも Many であるような場合に、上記と同じ操作を行うことを考えてみます。
prefecdatapanel.wf1 をダウンロードし、citydata.wf1 と同時に開いてください。
prefecdatapanel.wf1 はパネルデータになっているので、Tokyo, Saitama のどちらについても、2 年分のデータの保存先がございます。以下のコマンドを入力してください。

copy(c=mean) citydata::citydata\x prefecdatapanel::prefecdatapanel\x @src prefecture @dest prefecture

各年共に、X の都道府県平均値が移植されます。このように、Many to Many は、まず最初に Many to One の処理を行い、処理されたデータに対し One to Many の処理を行うことと同値です。

Macth Merge 機能は便利である反面、操作を間違えると意図しない結果となってしまうことがございます。また、操作が正しくても内部でお客様の意図と異なる計算が行われている可能性や (たとえば、計算において NA をどのように扱うかなど)、バグがある可能性もございます。**面倒かとは思いますが、操作を行った後、目視により移植が正しく行われているかを一度必ず確認してください。特に、もし不可能でなければ、本文書の最初の方で行ったように、いくつか候補を抜き出して手計算による結果と一致するか確かめてみてください。**
なお、EViews9 の新機能であるコマンドキャプチャ機能を使うと、行ったメニュー操作に対応するコマンドがコマンドキャプチャウィンドウに自動で記録されます。これによりいちいちヘルプメニューを調べる手間が省けます。一部キャプチャされない操作もございますが、何かと便利な機能ですので、EViews8 以前のバージョンをお使いの方はもしよろしければアップグレードをご検討ください。

今回の最後で、データに時点情報があるケースについて簡単に触れました。今回は紙面の都合上あまり詳しく述べられませんでしたが、時点情報の処理に関しては他にご説明したい点などもございます。ですので次回はパネルデータの場合の移植について、より詳しくご紹介したいと思います。