欠損値の補完　多重補完法

**調査における欠損値とはなにか**

実際のデータ分析を行う上では、欠損値の問題は避けては通れないものである。日本語の解説としては松田・伴・美添を参照すること。

**欠損値を補完するべきか、しないべきか**

通常の分析手続きをより煩雑にする欠損値を補完する作業は、出来る事であれば避けたいというのは多くの人が共通して望むことであろう。ここで簡単に欠損値を補完することの是非について論じておきたい。まず補完するべき理由であるが、適切な方法で補完することでより正確な推定値と分析結果が得られることが期待される。インフォーマルな分析であればともかく、出版して世間に広く流通させることを目的とする分析や学位論文であれば出来るだけ正確な分析をすることを目指すのは多くの研究者が望むことであろう。また、学術雑誌によってはレビューの過程で欠損値を補完することを求められる機会も増えている。それらは欠損値を補完する事の十分な理由であると思える。

しかしながら、研究活動という知的生産もコストパフォーマンスを無視しては行えない。すべての集計や予備的な分析で欠損値の補完をすることは非効率であり現実的でない。手間をかける事を鑑みると、世間に公表する少数の重要な推定量（たとえば失業率・所得・得票のシェアの推定）や論文の鍵となる重要な多変量解析などでは欠損値を補完し、その他の予備的な分析では欠損値を補完しないまま分析するのが現実的である。

補完しないリスクについても述べておこう。欠損値を補完したデータを用いた推定量はより正確であると先に述べたが、ではどのような状況で推定量が不正確になるのだろうか。Little and Rubinによると欠損のパターンとは概ね三つの場合に分けられる。  
  
Missing at completely random (MCAR)    
  
これは欠損が他の観測される変数に依存していない場合である。例えば、アクシデントで調査票のページが破れて幾人かの回答が失われた場合（調査票が束になっているその順序はランダムであると仮定する）、その欠損のプロセスは回答の値とも、回答者の他の質問への回答とも依存関係がない。この場合、欠損を無視しても、推定量にはバイアスは生じない。  
  
Mising at random (MAR)  
  
これは欠損が観測される他の変数には依存しているものの、回答そのものの値とは無関係な場合である。例としては、世帯収入を尋ねる質問があるとする。プライベートな情報である収入に関する質問は通常欠損が多い。またこの欠損は、比較的に高学歴の層と、低学歴や高齢の層に多い。この場合、欠損を無視した収入の分布は低所得者と高所得者の割合が過小評価されてしまうことになる。しかし、教育程度・年齢と収入の間には正の相関があり教育程度と年齢を考慮した条件付き分布であれば欠損によるバイアスの量は小さい。MARの仮定が正しければ、欠損のメカニズムをよく近似する変数による条件付き分布を見る限りMCARが成り立っているということである。  
  
無視できない欠損（Missing not at random)  
  
これは観測値の欠損がその値に依存している場合で、この場合はデータを用いた推定にはバイアスがあり、かつ補完もできない。例えば大学入学の試験の際の成績と、入学後の成績の相関関係に関心があるとする。しかし入学試験試の際の得点が低い層はそもそも入学が許可されないために、入学後の成績は観測されず、両者の相関関係の推定値を妥当な形で求めることはできない。このような標本選択が起きている場合は通常の方法で推定を行ったり保管をすることは不可能である。  
  
まとめると、もし欠損がMCARで起きているのであれば欠損を無視した分析をしてもバイアスはなく何の影響もないが、欠損がMARで起きているのであれば補完をすることでバイアスを減らすことができる。

**様々な欠損に対するアプローチと諸問題**

欠損値は無視する場合。欠損値のある標本は無視して、可能な限りの標本を用いて分析をする(complete data analysis)というのも一つの方法である。この方法の欠点は、特に多変量解析を行う時など標本数が少なくなることである。標本数が少なくなる事の問題は以下の３つに集約される。一つは手間をかけたデータが使えなくなるというコストパフォーマンス上の問題。二つ目は質問に無回答のある標本が分析から欠落することで偏った標本の分析になり、分析による知見の妥当性が脅かされるという事である。最後に、標本数が減ることで標本誤差が増えて統計的検定を行った際に有意な結果が出にくくなるという事もある。

欠損値に何らかの値を代入する場合。これを欠損値補完 (imputation)というが、補完には様々な方法がある。大別して分けると、欠損値に平均値や中位点などの値を補完する方法と、回帰分析などで予測値を求め、それに乱数を加えて複数の値を補完する(multiple imputation)方法に分けられる。

前者のように一つの値を一意に補完する方法にはいくつかの問題がある。具体的な例を挙げてみて見よう。本書で扱っている調査データでは生年月日を尋ねている。この質問には欠損は無いのであるが、敢て人工的に欠損を発生させて欠損を平均や中位点などで補完することのデメリットを見てみよう。まず３章で扱ったように、生年月日から年齢を計算する。

recode q2 999 = .  
gen age = 104 - (q2+25)  
tab age,m

次に年齢(age)変数のコピーとして新しい変数age\_missをつくる。ここではランダムに25%の回答者の年齢を欠損にしている。runiform()関数は０から１の間で一様分布に従う実数の乱数を発生させる。こうして作られた変数RANDを利用して、年齢変数を欠損にすることでランダムに年齢変数を欠損に変えることができる。ちなみにこの原理は、単純無作為抽出をシミュレーションする際に良く使われる方法である。  
  
set seed 39374  
gen RAND = runiform()  
gen age\_miss = age  
replace age\_miss = . if RAND <= .3  
su age age\_miss

こうして作られたage\_miss変数とオリジナルのage変数を比べてみると、平均も標準偏差もほぼ等しいことが見て取れる。

Variable | Obs Mean Std. Dev. Min Max  
-------------+--------------------------------------------------------  
 age | 1445 51.71419 13.51143 24 74  
 age\_miss | 1032 51.59787 13.37993 24 74

次に、ナイーブな補完の方法として平均値による補完を試みる。summarize コマンドで、欠損のある年齢変数の平均を求める。Stataは毎回コマンドを走らせる度に、内部のマクロと行列に各種統計量を保存するのである。summarize コマンドの後にreturn listを打つことで、内部のマクロに保存された値を確認することができる。この場合は平均値である51.56がr(mean)という名前のローカルマクロに保存されているので、replaceコマンドを用いて欠損値をこの平均値で置き換えている。ちなみに、Stataでローカルマクロを参照する際には最初の括弧に「`」を用い、次の括弧に「'」を使う必要がある。これはStataの独自の性質で注意を要する。また、ローカルマクロはセッションが終わると消えてしまうので一度のセッションでローカルマクロを計算する元になったコマンドと、ローカルマクロを用いた計算が実行される必要がある。

そうした後、再びオリジナルの年齢変数と、平均補完をした年齢変数を比較してみると、平均値は相変わらず変わらないのであるが、補完をした変数の標準偏差が小さくなっていることがわかる。これは、25%の標本に平均値が補完されたために、データのばらつきが減り、標準偏差が下のデータよりも小さく推定されてしまっている。

summarize age\_miss  
return list  
display `r(mean)'  
replace age\_miss = `r(mean)' if age\_miss == .  
su age age\_miss

出力例

. summarize age\_miss

Variable | Obs Mean Std. Dev. Min Max

-------------+--------------------------------------------------------

age\_miss | 1032 51.59787 13.37993 24 74

. return list

scalars:

r(N) = 1032

r(sum\_w) = 1032

r(mean) = 51.59786821705426

r(Var) = 179.0224202813555

r(sd) = 13.37992601927811

r(min) = 24

r(max) = 74

r(sum) = 53249

. display `r(mean)'

51.597868

. replace age\_miss = `r(mean)' if age\_miss == .

su age age\_miss

Variable | Obs Mean Std. Dev. Min Max

-------------+--------------------------------------------------------

age | 1445 51.71419 13.51143 24 74

age\_miss | 1445 51.59787 11.30575 24 74

ヒストグラムを比較してみると、平均を補完した変数（右パネル）の分布が歪んでいることがわかる。この歪みは二つの問題を引き起こす。一つは補完された変数を用いて分散や標準誤差を計算した場合、欠損値に同じ値が代入されたことで、分散を減らし標準誤差の過小推定につながるということである。これは、変数の点推定と標準誤差の推定を主目的とする家計調査や人口統計学的な調査の場合には深刻な欠点である。第二に変数間の相関を調べることを主眼とする社会調査においては、この補完は実質測定誤差が増えることに対応し、通常回帰係数が過小に評価されることにつながる(Fuller, xxx)。



これはすこし進んだ内容になるが、平均値を代入することのデメリットを示すために簡単なシミュレーションを行ってみる。

* 年齢変数を取り、ランダムに5%, 10%から60%まで人工的に欠損値を発生させ、平均値で補完する。
* 学歴変数と欠損を補完した年齢変数の間の相関係数を求める
* 各欠損の割合について60回試行を繰り返す。

下の図は相関係数（Y軸）と欠損の割合（X軸）の関係を図示したものである。欠損が少ない場合、学歴と年齢の間には -.38程度の相関がある。これは戦後、高等教育の機会が拡大したために、年齢と学歴の間に負の相関が見られるのである。しかし、年齢変数の欠損が増えるにつれて、相関係数の絶対値で見た大きさが小さくなることが見て取れる。これは相関の希薄化(attenuation)と呼ばれる現象である。

これがおきた理由は、ランダムに発生させた欠損値に対して、学歴変数の値にかかわらずに強制的に平均値を補完したことで、データの相関構造が歪んでしまったのである。次の節では、このようなデータ分析上の問題を回避するための、より妥当な方法による欠損の補完方法について解説する。



**Stataを用いた多重補完のやりかた**

サンプルデータを用いて多重補完法を行ってみよう。ここではStata ver.12以降を使うことを前提にする。サンプルデータの学歴・職業および年収を補完する事を試みる。以下に簡単に欠損の程度をまとめる。慣れない間は、以下のような関係をまとめることを強く薦める。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 質問番号 | 質問 | 欠損数 | 依存関係 | 変数の種類 |
| Q1 | 性別 | なし | なし | 離散(０か１) |
| Q2 | 年齢 | なし | なし | 離散 |
| Q3 | 学歴 | 16 | なし | 順序 |
| Q5 | 就業経験 | 5 | なし | 名義 |
| Q5x1 | 職業 | 23 | Q5 で1,2,3と回答 | 名義 |
| Q5x2 | 産業 | 30 | Q5 で1,2,3と回答 | 名義 |
| Q6 | 収入 | 123 |  | 順序 |

まず、依存関係の無い学歴(Q3)と就業経験(Q5)および収入(Q6)を補完する事を試みる。最初に多重補完をする際の環境変数を設定する必要がある。mi set flong は補完したデータをどのように保存するかを指定するコマンドである。Stataは複数の補完データの保存法をサポートしているが、筆者が勧めるのはflongスタイルである。次に mi set M = 5 の行では補完データの数を指定している。コンピューターが非力であった時代には、メモリや計算時間を節約するために出来るだけ小さな数に留めて置きたい動機があった。ルービン(1987)はM=5でたいていの場合は十分であると述べている。計算機の発達に現在ではこの数を20や50しても十分に対応できるであろう[[1]](#footnote-1)。set seed 9875783 は乱数のシードナンバーを与えている。これを設定することで、分析を繰り返したときでも補完する値を再現することが出来るようになる。プログラムにミスなどがあり、シンタックスを修正したときにこれを設定していないと毎回値が少し変わるために、シンタックスを修正した結果として値が変わっているのか、それとも乱数のために値が変わっているのかが判別で不便であるからである。ユーザは分析の際に任意の値を設定することを勧める[[2]](#footnote-2)。

mi set flong

mi set M = 5

set seed 9875783

次に分析で用いる変数の欠損値処理を行う。Q5の就業経験については、Q5をそのまま補完するのではなくて離散をつくる。これは補完をするときの統計モデルの選択に関わる理由によるもので後に詳しく説明する。

recode q3 999 = .

recode q4 999 = .

recode q5 999 = .

recode q6 999 = .

gen EverWorked = q5

recode EverWorked 1/3 = 1 4 = 0

そして mi register imputedコマンドで補完をする変数を指定してStataに教えてあげる。

mi register imputed q3 EverWorked q6

そして以下のコマンドが欠損を補完する部分である。ここでは学歴(Q3)と収入(Q6)を順序変数とみなす２行目の (ologit , ascontinuous ) q3 q6 というのは従属変数がQ3もしくはQ6で補完のモデルは順序ロジットモデルを選択しているということである。オプションのascontinuousというのは、学歴や収入を独立変数として使う場合には連続量として回帰分析に投入することを指定している。３行目の(logit) EverWorked はかつての就業経験(EverWorked)を従属変数にロジステックモデルで補完を行うことをしている。４行目の= q1 q2 , replace noisilyはQ1（性別）およびQ2（生まれた年）を独立変数にしている。

mi impute chained ///

(ologit , ascontinuous ) q3 q6 ///

(logit ) EverWorked ///

 = q1 q2 , replace noisily

ここで興味深いこととしてはQ2の生まれた年については、昭和の元号の年のままでも構わないし、実際の年齢に変換しても結果は変わらないということである。通常の多変量分析であれば、推定された回帰係数の値と解釈に関心があるわけであるが、欠損値の補完においては回帰係数の大小やその値には関心が無い。その為に、その変数の統計的な意味での扱いが正しければ（例えば新聞購読などの名義変数はダミー変数に変換してから用いる）、回帰係数の解釈を分かりやすくするための変換をする必要は無い。Q1の性別もこれはオリジナルの値のままで、０－１の離散変数に変えることが通常であろうが無視して構わない。

**補完をする際のモデルのspecificationの推奨される方法**

欠損値を補完する際は、出来るだけ補完する対象の変数の尺度を考慮した推定をすることが望ましい。例えば性別であれば。０か１の値をとる離散変数にしたうえで、ロジットモデルで補完するなどである。ここで、性別を補完する際に線形回帰などを用いると、線形回帰の従属変数は理論的には実数であり負の無限大から正の無限大までの範囲をとりうるので、性別変数に-3などの値が補完されることになり都合が悪い。同様に、補完する変数が連続量とみなせる変数（年齢や体重）であればロジットモデルやポワソンモデルよりも、線形回帰が望ましい。また、年齢や体重であれば変数が、整数でなく実数(46.54歳)でもかまない。

したの表に変数のタイプと、推奨されるモデルの特定化の方法をまとめた。Stataのマニュアルを読めば、さらにいくつか特殊なモデルが可能となっているが基本的には以下のモデルで殆どのケースを網羅できると考える。他の多くの分析と同様に、異なる方法を用いても同じ目的を達成出来ることがあり、欠損地の補完も例外ではない。例えば、どの新聞を購読するかという質問については、複数の離散変数を用いてそれをロジットモデルで補完することも出来るし、購読している新聞名という名義尺度のまま、多項ロジステックモデルで補完することも出来る。Stataのmlogitはやや不安定であるので、筆者としては名義尺度はダミー化してロジステックモデルで補完することを勧める。

筆者の経験では、出来るだけ推定の方法を単純にしたほうが時間がかからないだけでなく、無用のエラーも減り、望ましい。例えば、新聞紙の名前変数を分析に使うために補完をしたいとする。しかし、分析上は「朝日新聞読者」か「それ以外」にしか関心がないとする。その場合は名義尺度として扱うよりも、2値の離散変数にしてしまってから補完をすることを勧める。

また、補完モデルは回帰分析そのものなので、基本的には単純であればあるほどロバストなモデルになる。具体的には、独立変数の数（推定される係数の数）をむやみに増やさない。外れ値などは処理しておくである。回帰分析なので、必要に応じて交互作用変数なども投入できるが、モデルの複雑さを増すので無闇にたくさん入れればよいというものでもない。

mi impute chainedのメリットは補完の対象になっている従属変数が、自動的にほかの補完モデルにおいては独立変数として扱われることである。ここで注意すべきは、ologit, mlogitを用いて補完された変数は独立変数となる際に、名義変数として扱われることである。

|  |  |  |  |
| --- | --- | --- | --- |
| 変数のタイプ | 補完モデル | 補完される値 | 例 |
| 離散変数 | logit | 整数 ０か１ | 性別、新聞購読、投票したか否か |
| 連続変数 | regress | 実数 | 年齢、体重、収入の額 |
| 順序変数 | ologit | Stataでは通常は整数として扱われる | 順序変数として尋ねられた学歴・収入・態度など |
| 名義変数 | mlogit | Stataでは通常は整数として扱われる | 相互背反な名義尺度。人種・職業・居住地など |
| ポワソン分布に従う変数 | poisson | 正の整数 | 回数として数えられるもの。過去の投票数・運動回数など |

**多重補完されたデータセットを用いた分析**

1. 標本数が十万や百万件のデータを扱いかつMの数を20などに増やす場合は、欠損データの保存方法をflongよりもmlongにしたほうが効率が良い。 [↑](#footnote-ref-1)
2. シードは0から2,147,483,647までの整数を自由に決められる。これはStataがC言語で書かれているのでC言語の32ビットの整数型を受け継いでいるからである。 [↑](#footnote-ref-2)