金融データの欠損値補完

田　中　謙一郎

目次

はじめに

データ分析において欠損値の存在とその処理は多くの研究者の頭を悩ます事象の一つであった。初期における対処法としてはリストワイズ削除やペアワイズ削除など欠損値のあるオブザベーションを除去するアプローチと平均値や中央値などで欠損した部分を補間する方法があった。しかし、1976年にD.B. Rubinの論文が『Biometrika』誌に掲載された頃より様々な補完の方法が試みられてきた。これは、使用するデータの大型化や電子計算機の性能向上と無縁なことではないように思える。連続変数の欠損を補完する場合、欠損が単調であればFCS（fully conditional specification)モデルのベイズ回帰法と予測平均マッチング法、あるいはノンパラメトリックな傾向スコア法などが使用される。一方、欠測パターンが非単調な場合はMICE（Multiple Imputation by Chained Equations）アルゴリズムやデータに多変量正規分布を仮定するMCMC（Markov Chain Monte Carlo）法が一般的である。この他、多重代入法で利用されるEMB（Expectation-Maximization

with Bootstrapping）法などがあり、それぞれSAS、IBM SPSS Statistics, 同Amos, M-Plus, Stataなどの商用ソフトウェアに実装されているだけでなく、フリーソフトウェアの代表格であるRのパッケージAmeliaやMICEなども活用可能である。

1.　欠損値の分類

1つ目はMCAR (Missing Completely At Random)と呼ばれる完全にランダムに起こってしまう欠損である。2つ目はMAR (Missing At Random)と呼ばれる他の変数が原因で発生してしまう欠損（医療の現場では薬の投与の研究をしていて、患者が転院してしまうとそれ以上データが取れなくなって欠損してしまうという時間変数に依存してしまう場合である。また、未婚か既婚かを尋ねるアンケートで女性が答えにくかったり、特定の年齢層が答えてくれなかったりする場合に相当する。）である。3つ目はMNAR (Missing Not At Random)と呼ばれ、Non-ignorable Missing(無視できない欠損)の意味である。これは欠損した値に依存するもので、扱いが難しいといわれている。たとえば、年収の場合欠損が年収そのものの値に依存しているということである。

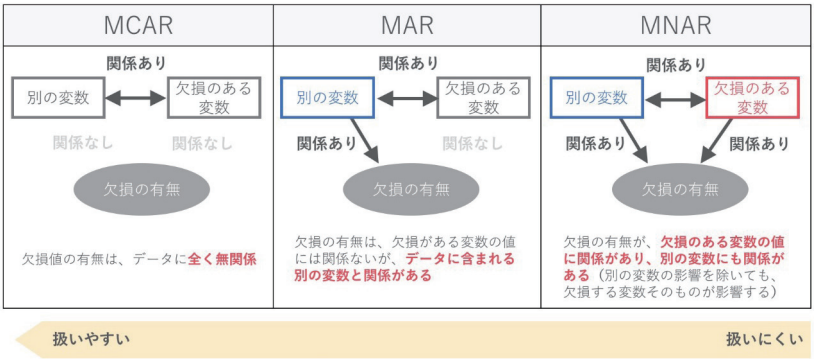


図1-1　3種類の欠損値

【出所：http://qiita.com/fujit33/items/fe38c6e67d6511168f6f】

次に簡単な数値例で人為的に欠損値を発生させ、それを補完（代入）して、従来の単一代入法を比較してみる。ここで、雑誌『金融ビジネス』1997年3月号（東洋経済新報社）による、14の都市銀行の９つの財務指標（V3からV11まで）と銀行の実力度を判別するFactor変数（V2－1：上位；0：下位）から成るデータセット”TestData”の中身を見てみる。ここで、V1は銀行名、V2は銀行経営の実力度が上位か下位かのFactor変数である。V3からV11まではV2の判断材料となる財務指標を表わす。これを倒産分析のデータと見なすことも可能である。ちなみに、V3は総資金業務純益率、V4は1人当たり資金益、V5は自己資本比率、V6は資金量平残、V7は株主資本純益率である。さらに、V8は粗利経費率であり、V9は国内総資金利鞘、V10は不良債権比率、V11は不良債権引当金となる。

```
> TestData
            V1 V2  V3  V4  V5  V6  V7  V8   V9 V10  V11
1       Taisho  1 100 293 277   2 151 626  936 841  642
2    Tokushima  1 440 344 342  12 204 414 1000 985 1000
3      Hokkoku  1 407 411 773  42 185 424  702 873  349
4       Suruga  1 407 409 499  49 106 459  702 843  409
5      Chugoku  1 308 474 474  76 184 450  628 946  131
6    Yamaguchi  1 286 386 386  69 127 433  564 907  312
7     Shizuoka  1 187 369 369 120 142 329  309 957  329
8       Naniwa  0 560 195 195   5  73 440  638 439  162
9      Hanshin  0 385 246 246  17  81 392  606 550  202
10    Hokkaido  0 220 303 303  54  88 305  383 536  175
11     Kokumin  0 440 124 124   7  62 382  394 204  176
12 ChibaKogyo  0  88 203 203  33  92 189  330 595  107
13       Osaka  0  99 107 107  28  96 130   64 688  222
14 KyotoKyoei  0 187  68  68   4  68 191  277 189  167
```

まず、統計解析向けのプログラミング言語Rのパッケージのひとつである ForImpのmissingmat2コマンドにより、V2-V11にランダムに10個の欠損値（NA）を代入する。

```
> MissData
           V1 V2  V3  V4  V5  V6  V7  V8   V9 V10  V11
1      Taisho  1  NA 293 277   2 151 626  936 841  642
2   Tokushima NA 440 344 342  12 204 414 1000 985 1000
3     Hokkoku NA 407 411 773  42 185 424  702 873  349
4      Suruga  1 407 409 499  49 106 459  702  NA  409
5     Chugoku  1 308 474 474  76 184 450  628 946  131
6   Yamaguchi  1 286 386 386  69 127 433  564 907  312
7    Shizuoka  1 187 369 369 120 142 329  309 957  329
8      Naniwa  0 560 195 195   5  73 440   NA 439  162
9     Hanshin  0  NA 246 246  17  81 392  606 550  202
10   Hokkaido NA 220 303 303  54  88 305  383 536  175
11    Kokumin NA 440 124 124   7  62 382  394 204  176
12 ChibaKogyo  0  88  NA 203  33  92 189  330 595  107
13      Osaka  0  99 107 107  28  96 130   64 688  222
14 KyotoKyoei  0 187  68  68   4  68 191   NA 189  167
```

同じくRのパッケージForImpのmissingnessコマンドにより、データの欠損値情報を表示することが出来る。

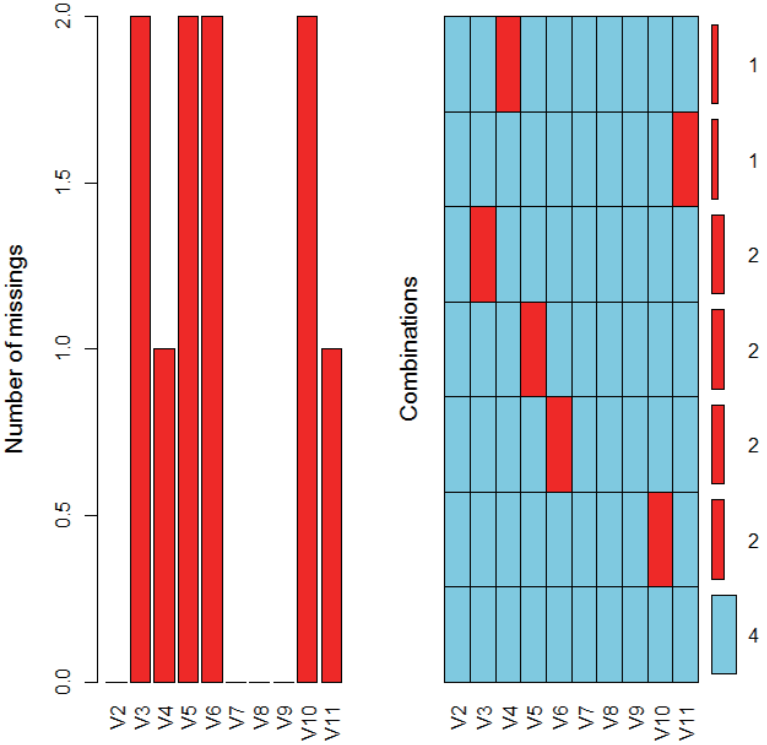
```
> missingness(MissData)
$number_of_missing_values
[1] 10

$missing_values_per_unit

 0  1
 4 10

$missing_values_per_variable
 V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11
  0   4   2   1   0   0   0   0   2   1   0
```

RのパッケージVIMによって欠損値パターンの可視化を行ってみる。単一のデータ項目に対する欠損値の件数の棒グラフ、複数のデータ項目の欠損値のパターンの可視化を行ってみる。



NA値に各列データの中央値を代入する場合、Rのパッケージ ForImpの medianimpコマンドが有効である。

```
> medianimp(MissData[,-1])
   V2  V3  V4  V5  V6  V7  V8   V9 V10  V11
1   1 286 293 277   2 151 626  936 841  642
2   0 440 344 342  12 204 414 1000 985 1000
3   0 407 411 773  42 185 424  702 873  349
4   1 407 409 499  49 106 459  702 688  409
5   1 308 474 474  76 184 450  628 946  131
6   1 286 386 386  69 127 433  564 907  312
7   1 187 369 369 120 142 329  309 957  329
8   0 560 195 195   5  73 440  564 439  162
9   0 286 246 246  17  81 392  606 550  202
10  0 220 303 303  54  88 305  383 536  175
11  0 440 124 124   7  62 382  394 204  176
12  0  88 303 203  33  92 189  330 595  107
13  0  99 107 107  28  96 130   64 688  222
14  0 187  68  68   4  68 191  564 189  167
```

また、NA値に各列データの平均値を代入する場合、meanimpコマンドがあるがこちらの出力結果は省略する。さらに、Rの欠損値パッケージにはimputeMissingsもある。ここでは、2列目の変数V2をfactor変数に変換しておくことが肝要である。

```
> str(MissData)
'data.frame':   14 obs. of  11 variables:
 $ V1 : Factor w/ 14 levels "ChibaKogyo","Chugoku",..: 12 13 5 11 2 14 10 8 3 4
 $ V2 : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 1 1 ...
 $ V3 : int  NA 440 407 407 308 286 187 560 NA 220 ...
 $ V4 : int  293 344 411 409 474 386 369 195 246 303 ...
 $ V5 : int  277 342 773 499 474 386 369 195 246 303 ...
 $ V6 : int  2 12 42 49 76 69 120 5 17 54 ...
 $ V7 : int  151 204 185 106 184 127 142 73 81 88 ...
 $ V8 : int  626 414 424 459 450 433 329 440 392 305 ...
 $ V9 : int  936 1000 702 702 628 564 309 NA 606 383 ...
 $ V10: int  841 985 873 NA 946 907 957 439 550 536 ...
 $ V11: int  642 1000 349 409 131 312 329 162 202 175 ...
```

このパッケージの補完コマンドimputeはmethodオプションでmedian/modeとrandomForest/modeを選択することが出来る。ただし、後者は数値データのみに有効で、factor変数に対する補完は出来ない。

```
> impute(MissData, method = "median/mode")
            V1 V2  V3  V4  V5  V6  V7  V8   V9 V10  V11
1       Taisho  1 297 293 277   2 151 626  936 841  642
2    Tokushima  1 440 344 342  12 204 414 1000 985 1000
3      Hokkoku  1 407 411 773  42 185 424  702 873  349
4       Suruga  1 407 409 499  49 106 459  702 688  409
5      Chugoku  1 308 474 474  76 184 450  628 946  131
6    Yamaguchi  1 286 386 386  69 127 433  564 907  312
7     Shizuoka  1 187 369 369 120 142 329  309 957  329
8       Naniwa  0 560 195 195   5  73 440  585 439  162
9      Hanshin  0 297 246 246  17  81 392  606 550  202
10    Hokkaido  0 220 303 303  54  88 305  383 536  175
11     Kokumin  0 440 124 124   7  62 382  394 204  176
12 ChibaKogyo  0  88 303 203  33  92 189  330 595  107
13       Osaka  0  99 107 107  28  96 130   64 688  222
14 KyotoKyoei  0 187  68  68   4  68 191  585 189  167
```

```
> impute(MissData[,3:11], method = "randomForest")
         V3       V4  V5  V6  V7  V8        V9       V10  V11
1  343.3077 293.0000 277   2 151 626  936.0000 841.0000  642
2  440.0000 344.0000 342  12 204 414 1000.0000 985.0000 1000
3  407.0000 411.0000 773  42 185 424  702.0000 873.0000  349
4  407.0000 409.0000 499  49 106 459  702.0000 776.5002  409
5  308.0000 474.0000 474  76 184 450  628.0000 946.0000  131
6  286.0000 386.0000 386  69 127 433  564.0000 907.0000  312
7  187.0000 369.0000 369 120 142 329  309.0000 957.0000  329
8  560.0000 195.0000 195   5  73 440  581.9312 439.0000  162
9  332.9773 246.0000 246  17  81 392  606.0000 550.0000  202
10 220.0000 303.0000 303  54  88 305  383.0000 536.0000  175
11 440.0000 124.0000 124   7  62 382  394.0000 204.0000  176
12  88.0000 255.7704 203  33  92 189  330.0000 595.0000  107
13  99.0000 107.0000 107  28  96 130   64.0000 688.0000  222
14 187.0000  68.0000  68   4  68 191  435.2600 189.0000  167
```

このほか、Rパッケージzooによる補完機能も表示可能である。欠損値を直近の値で補完した場合は以下の通りである。欠損値より以前の値が存在しない場合は補完できない。

```
> na.locf(MissData)
           V1 V2   V3  V4  V5  V6  V7  V8   V9 V10  V11
1      Taisho  1 <NA> 293 277   2 151 626  936 841  642
2   Tokushima  1  440 344 342  12 204 414 1000 985 1000
3     Hokkoku  1  407 411 773  42 185 424  702 873  349
4      Suruga  1  407 409 499  49 106 459  702 873  409
5     Chugoku  1  308 474 474  76 184 450  628 946  131
6   Yamaguchi  1  286 386 386  69 127 433  564 907  312
7    Shizuoka  1  187 369 369 120 142 329  309 957  329
8      Naniwa  0  560 195 195   5  73 440  309 439  162
9     Hanshin  0  560 246 246  17  81 392  606 550  202
10   Hokkaido  0  220 303 303  54  88 305  383 536  175
11    Kokumin  0  440 124 124   7  62 382  394 204  176
12 ChibaKogyo  0   88 124 203  33  92 189  330 595  107
13      Osaka  0   99 107 107  28  96 130   64 688  222
14 KyotoKyoei  0  187  68  68   4  68 191   64 189  167
```

それとは反対に、欠損値を次の時点の値で補完した場合は逆の現象が起こる。

```
> na.locf(MissData,fromLast=TRUE)
           V1 V2  V3  V4  V5  V6  V7  V8   V9 V10  V11
1      Taisho  1 440 293 277   2 151 626  936 841  642
2   Tokushima  1 440 344 342  12 204 414 1000 985 1000
3     Hokkoku  1 407 411 773  42 185 424  702 873  349
4      Suruga  1 407 409 499  49 106 459  702 946  409
5     Chugoku  1 308 474 474  76 184 450  628 946  131
6   Yamaguchi  1 286 386 386  69 127 433  564 907  312
7    Shizuoka  1 187 369 369 120 142 329  309 957  329
8      Naniwa  0 560 195 195   5  73 440  606 439  162
9     Hanshin  0 220 246 246  17  81 392  606 550  202
10   Hokkaido  0 220 303 303  54  88 305  383 536  175
11    Kokumin  0 440 124 124   7  62 382  394 204  176
12 ChibaKogyo  0  88 107 203  33  92 189  330 595  107
13      Osaka  0  99 107 107  28  96 130   64 688  222
14 KyotoKyoei  0 187  68  68   4  68 191 <NA> 189  167
```

次に考えられるのが線形補間であり、これはna.approx関数で実行できる。この数値例ではV2だけが連続2コマ欠損となっていたが、一般に連続2コマあるいは3コマ欠損値が続く場合、中央値や平均値の設定を個別に行うことはデータの次元が増えた場合、極めて非効率的である。そこで、線形補間という手法が有効となるわけである。

```
> na.approx(MissData[,3:11])
       V3    V4  V5  V6  V7  V8     V9    V10  V11
 [1,]  NA 293.0 277   2 151 626  936.0 841.0  642
 [2,] 440 344.0 342  12 204 414 1000.0 985.0 1000
 [3,] 407 411.0 773  42 185 424  702.0 873.0  349
 [4,] 407 409.0 499  49 106 459  702.0 909.5  409
 [5,] 308 474.0 474  76 184 450  628.0 946.0  131
 [6,] 286 386.0 386  69 127 433  564.0 907.0  312
 [7,] 187 369.0 369 120 142 329  309.0 957.0  329
 [8,] 560 195.0 195   5  73 440  457.5 439.0  162
 [9,] 390 246.0 246  17  81 392  606.0 550.0  202
[10,] 220 303.0 303  54  88 305  383.0 536.0  175
[11,] 440 124.0 124   7  62 382  394.0 204.0  176
[12,]  88 115.5 203  33  92 189  330.0 595.0  107
[13,]  99 107.0 107  28  96 130   64.0 688.0  222
[14,] 187  68.0  68   4  68 191     NA 189.0  167
```

その次が、よりなめらかなスプライン補間であり、na.spline関数で実行した。

```
> na.spline(MissData[,3:11])
           V3        V4  V5  V6  V7  V8        V9      V10  V11
 [1,] 665.3747 293.00000 277   2 151 626  936.0000 841.000  642
 [2,] 440.0000 344.00000 342  12 204 414 1000.0000 985.000 1000
 [3,] 407.0000 411.00000 773  42 185 424  702.0000 873.000  349
 [4,] 407.0000 409.00000 499  49 106 459  702.0000 902.627  409
 [5,] 308.0000 474.00000 474  76 184 450  628.0000 946.000  131
 [6,] 286.0000 386.00000 386  69 127 433  564.0000 907.000  312
 [7,] 187.0000 369.00000 369 120 142 329  309.0000 957.000  329
 [8,] 560.0000 195.00000 195   5  73 440  453.2777 439.000  162
 [9,] 411.3720 246.00000 246  17  81 392  606.0000 550.000  202
[10,] 220.0000 303.00000 303  54  88 305  383.0000 536.000  175
[11,] 440.0000 124.00000 124   7  62 382  394.0000 204.000  176
[12,]  88.0000  70.11455 203  33  92 189  330.0000 595.000  107
[13,]  99.0000 107.00000 107  28  96 130   64.0000 688.000  222
[14,] 187.0000  68.00000  68   4  68 191 -503.0801 189.000  167
```

さらに、補完するデータの両端での2次微係数が0となるような制約を置いたスプライン補間である自然スプライン補間を選択することもできる。

```
> na.spline(MissData[,3:11],method="natural")
           V3        V4  V5  V6  V7  V8        V9       V10  V11
 [1,] 491.6536 293.00000 277   2 151 626  936.0000 841.0000  642
 [2,] 440.0000 344.00000 342  12 204 414 1000.0000 985.0000 1000
 [3,] 407.0000 411.00000 773  42 185 424  702.0000 873.0000  349
 [4,] 407.0000 409.00000 499  49 106 459  702.0000 897.9756  409
 [5,] 308.0000 474.00000 474  76 184 450  628.0000 946.0000  131
 [6,] 286.0000 386.00000 386  69 127 433  564.0000 907.0000  312
 [7,] 187.0000 369.00000 369 120 142 329  309.0000 957.0000  329
 [8,] 560.0000 195.00000 195   5  73 440  453.0587 439.0000  162
 [9,] 411.1996 246.00000 246  17  81 392  606.0000 550.0000  202
[10,] 220.0000 303.00000 303  54  88 305  383.0000 536.0000  175
[11,] 440.0000 124.00000 124   7  62 382  394.0000 204.0000  176
[12,]  88.0000  75.93561 203  33  92 189  330.0000 595.0000  107
[13,]  99.0000 107.00000 107  28  96 130   64.0000 688.0000  222
[14,] 187.0000  68.00000  68   4  68 191 -244.4573 189.0000  167
```

スプライン補間や自然スプライン補間で代入された値は、マイナスの値を含むという難点がある。これらを、商用プログラミング言語SASで実行し、回帰分析まで行うと以下のような出力となる。

Bank Evaluation 1997

OBS	bank	evaluation	X1	X2	X3	X4	X5	X6	X7	X8	X9
1	Taisho	1	100	293	277	2	151	626	936	841	642
2	Tokushim	1	440	344	342	12	204	414	1000	985	1000
3	Hokkoku	1	407	411	773	42	185	424	702	873	349
4	Suruga	1	407	409	499	.	106	459	702	843	409
5	Chugoku	1	308	474	491	.	184	450	628	946	131
6	Yamaguch	1	286	386	545	69	127	433	564	907	312
7	Shizuoka	1	187	369	878	120	142	329	309	957	329
8	Naniwa	0	560	195	200	5	73	440	638	439	162
9	Hanshin	0	385	246	177	.	81	392	606	550	202
10	Hokkaido	0	220	303	250	54	88	305	383	536	175
11	Kokumin	0	440	124	154	7	62	382	394	204	176
12	ChibaKog	0	88	203	296	33	92	189	330	595	107
13	Osaka	0	99	107	193	28	96	130	64	688	222
14	KyotoKyo	0	187	68	232	4	68	191	277	189	167

Bank Evaluation 1997

The MI Procedure

Model Information	
Data Set	WORK.GINKOU3
Method	FCS
Number of Imputations	10
Number of Burn-in Iterations	20
Seed for random number generator	14823

FCS Model Specification	
Method	Imputed Variables
Regression	X1 X2 X3 X4 X5 X6 X7 X8 X9
Discriminant Function	evalution

Missing Data Patterns

Group	evalution	X1	X2	X3	X4	X5	X6	X7	X8	X9	Freq	Percent	Group Means X1	X2	X3	X4	X5	X6	X7	X8	X9
1	X	X	X	X	X	X	X	X	X	X	11	78.57	274.000000	254.818182	376.363636	343.818818	117.090909	351.181818	506.818182	655.818182	331.000000
2	X	X	X	X	.	X	X	X	X	X	3	21.43	366.666667	376.333333	389.000000	.	.	123.666667	433.666667	645.333333	779.666667	247.333333

Variance Information

Variable	Variance Between	Variance Within	Variance Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
X4	967.089238	1265781	1276419	11.279	0.008404	0.008350	0.999166

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	MuO	t for H0: Mean=MuO	Pr > |t|
X4	3.377499	1129.787213	-2475.78	2482.533	11.279	-276.486966	36.780537	0	0.00	0.9977

Bank Evaluation 1997

OBS	_Imputation_	bank	evaluation	X1	X2	X3	X4	X5	X6	X7	X8	X9
1	1	Taisho	1	100	293	277	2.000	151	626	936	841	642
2	1	Tokushim	1	440	344	342	12.000	204	414	1000	985	1000
3	1	Hokkoku	1	407	411	773	42.000	185	424	702	873	349
4	1	Suruga	1	407	409	499	76.922	106	459	702	843	409
5	1	Chugoku	1	308	474	491	13.742	184	450	628	946	131
6	1	Yamaguch	1	286	386	545	69.000	127	433	564	907	312
7	1	Shizuoka	1	187	369	878	120.000	142	329	309	957	329
8	1	Naniwa	0	560	195	200	5.000	73	440	638	439	162
9	1	Hanshin	0	385	246	177	28.113	81	392	606	550	202
10	1	Hokkaido	0	220	303	250	54.000	88	305	383	536	175
11	1	Kokumin	0	440	124	154	7.000	62	382	394	204	176
12	1	ChibaKog	0	88	203	296	33.000	92	189	330	595	107
13	1	Osaka	0	99	107	193	28.000	96	130	64	688	222
14	1	KyotoKyo	0	187	68	232	4.000	68	191	277	189	167

Bank Evaluation 1997

REG プロシジャ
モデル：MODEL1
従属変数：X4

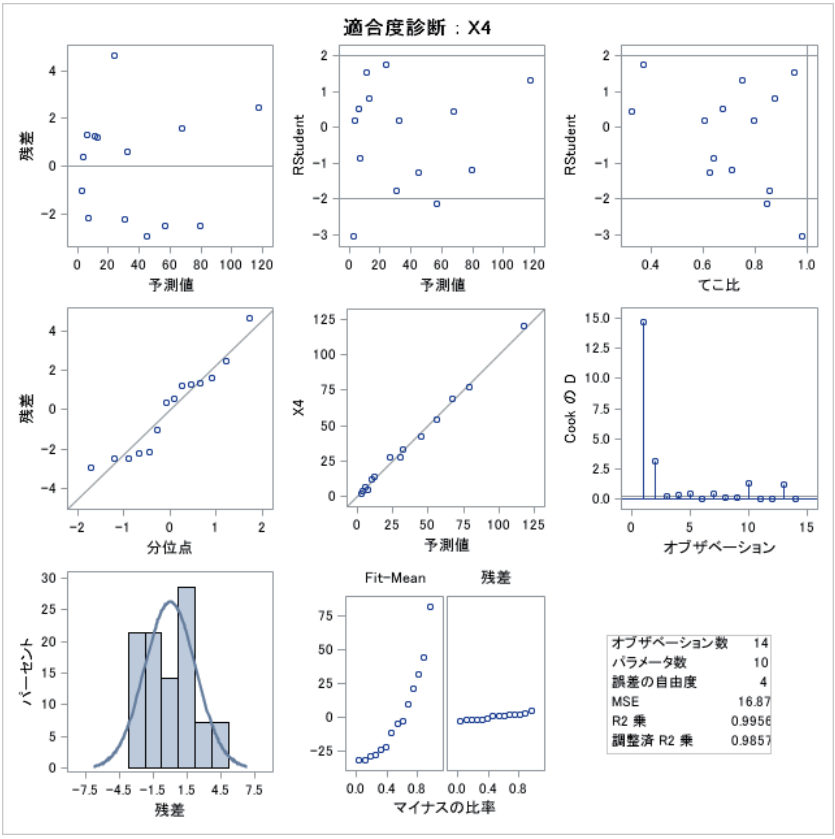
Imputation Number=1

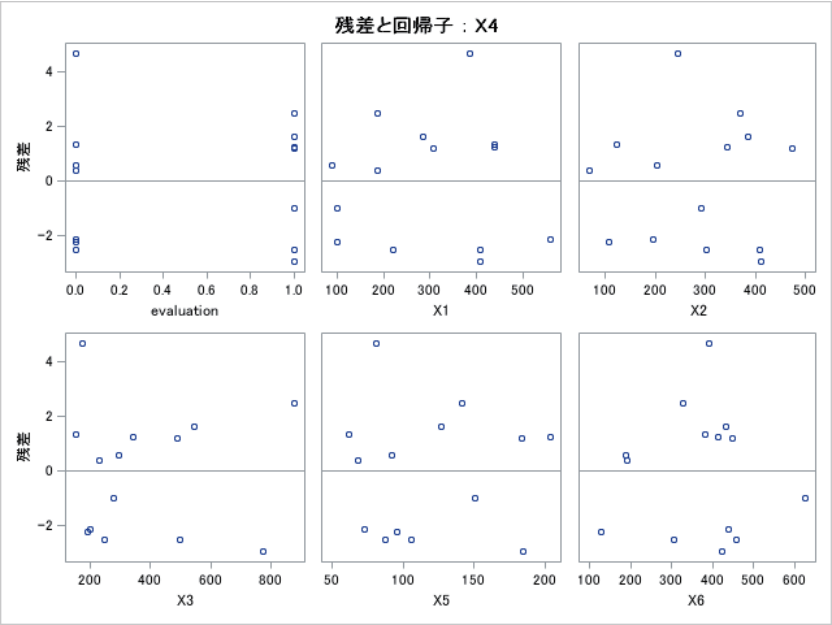
読み込んだオブザベーション数	14
使用されたオブザベーション数	14

分散分析					
要因	自由度	平方和	平均平方	F 値	Pr > F
Model	9	15295	1699.40582	100.74	0.0002
Error	4	67.47812	16.86953		
Corrected Total	13	15362			

Root MSE	4.10725	R2 乗	0.9956
従属変数の平均	35.34118	調整済 R2 乗	0.9857
変動係数	11.62172		

パラメータ推定値					
変数	自由度	パラメータ推定値	標準誤差	t 値	Pr > |t|
Intercept	1	6.11967	9.23562	0.66	0.5438
evaluation	1	−33.61515	8.93180	−3.76	0.0197
X1	1	−0.01553	0.01066	−1.46	0.2188
X2	1	0.24552	0.03001	8.18	0.0012
X3	1	0.12643	0.01207	10.48	0.0005
X5	1	−0.79738	0.06112	−13.05	0.0002
X6	1	0.02984	0.02526	1.18	0.3028
X7	1	−0.08547	0.01708	−5.00	0.0075
X8	1	0.04073	0.01310	3.11	0.0359
X9	1	0.11305	0.01214	9.31	0.0007


適合度診断：X4
残差
予測値
RStudent
予測値
RStudent
てこ比
残差
分位点
X4
予測値
Cook の D
オブザベーション
パーセント
残差
Fit−Mean
残差
マイナスの比率
オブザベーション数 14
パラメータ数 10
誤差の自由度 4
MSE 16.87
R2 乗 0.9956
調整済 R2 乗 0.9857



2．データの構成と欠損パターン

本稿で用いたデータは本学図書館保有のオンライン・データベースである日本経済新聞社「日経NEEDS Financial Quest Ver. 2.0」である。そのデータベースから事業所数1475社の７つの変数を抜き出した。事業所名(name)、企業コード(code)、売上高・営業収益(sales)、資産合計(assets)、資本金(capital)、売上原価(cost)、期末従業員数(employees)の7変数である。期間は1996年度から2016年度の20年間である。よって単純計算から、行数としては1475×20＝29500行となる。ただし、数値データが5変数とも欠落しているデータは補完には用いられないので、この分をあらかじめ削除すると27855行となる。

ところが、欠損値の補完を調べる場合、欠損の割合があまりに小さいと、

アルゴリズム同士の相異が明確には出てこない。そこで、高橋・伊藤[1]に従って、人為的に欠損データを発生させることにした。各々の財務データは大きさがまちまちのため、各データ系列を自然対数に置き換え、それに標準正規乱数を加えたものを昇順に並べ替え、小さなものから順に1%から10％を除き、対応する原系列同士で回帰分析を行った。売上高は10％、資産、資本金、売上原価は5％、期末従業員数は1％を欠損値とした。さらに、最初のSequential No.によって並び替えると欠損パターンは以下のようになる。各変数で人為的に発生させた欠損値との誤差があるのは、元々の欠損値との合計であるからである。いずれもIBM SPSS Statistics Ver.25のオプションである多重代入法からの出力である。

表2.-1　変数の要約

変数	欠損		有効値		
	度数	パーセント	N	平均値	標準偏差
sales	2811	10.10%	25045	293116.5	1238731.7
cost	1891	6.80%	25965	210787.6	991818.9
capital	1418	5.10%	26438	20044.8	66694.3
assets	1418	5.10%	26438	339058.6	1532320.1
employees	1042	3.70%	29814	5636.0	21838.0

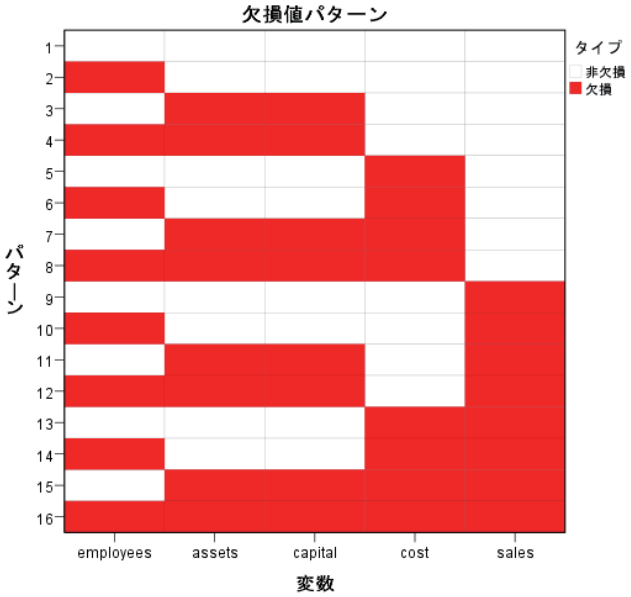


図2-1　欠損値パターン

3．単一代入法と多重代入法

欠損値に対する処理としてはいくつかの対応が考えられる。まず考えられるのが、欠損値の削除であり、リストワイズ法とペアワイズ法がある。前者は欠損レコードそのものを除去する方法であり、後者は相関係数など2変数を用いて計算を行う際に、対象の変数が 欠損している場合に計算対象から除外するものである。次に考えられるのが、欠損値に計算した値を代入（補完）方法であり、それには大きく分けて単一代入法と多重代入法がある。後者は、欠損値に代入したデータセットを複数作成し、各データセットに対して分析を実行し、その結果を統合するにより欠損値を補完する方法である。

欠損値の補完に対してもっとも簡便な方法は平均値を代入する方法であ

り、次には回帰分析による推定値を代入する方法である。欠損値が完全にランダムに欠損しているかどうか (MCAR) を検定するための Roderick J. A. Little のカイ 2 乗統計量が0.05 以下の場合、欠損値のパターンは観測データのみに関連する（MAR）となる。この場合は完全情報最尤推定法（Full Information Maximum Likelihood Estimation, FIML）が適している。それはサンプルごとに欠損パターンに応じた尤度関数を仮定して最尤推定を実施して得られる多変量正規分布を用いて平均値や分散共分散行列を推定する方法である。また、EM法（Expectation–Maximization algorithm）や期待値最大化法とは、統計学において、確率モデルのパラメータを最尤推定する手法の一つであり、観測不可能な潜在変数に確率モデルが依存する場合に用いられる。近年ではEM法にブートストラップ法を加味した手法がとられ、EMB(Expectation–Maximization with Bootstrapping)法と呼ばれることもある。EMB法は特に多重代入に適用する場合、ブートストラップによりコレスキー分解を回避しており、計算効率が高いと期待されるが、こちらも多変量正規分布を仮定しており、その前提が満たせない場合の性能には必ずしも保証はない。その代表格としてRパッケージのAmelia Ⅱがあり、米国ハーバード大学のJames Honaker, Gary King and Matthew Blackwellによって開発された。女性名Ameliaにちなんで、このソフトウェアに命名されたのは、高名な米国女性飛行士Amelia Mary Earhartが1937年南太平洋において行方不明になったことによるものと考えられる。

図3-1　AmeliaのGUI版 ”AmeliaView”の起動画面

多重代入については他にMCMC（マルコフ連鎖モンテカルロ）法と完全条件仕様連鎖方程式(Fully Conditional Specified Chained Equations：FCS)法がある。これらは、いずれもSAS Ver.9.3以降では両方のやり方で多重代入を補完できる。しかし、前者のアルゴリズムを実行できるソフトウェアとしてはM-plusがあり、後者のアルゴリズムを実行できるソフトウェアとしてはオランダのユトレヒト大学のStef van Buurenを中心とするグループによるRパッケージのMICE(Multivariate Imputation by Chained Equations)がある。FCS法には、適切な多変量分布が存在していなくても補完が可能であるという利点があるためと考えられる。さらに、統計ソフトとして一般的なStataもデータ拡大法として知られるMCMC法には”mi impute mvn”コマンドで対応し、FCS法には”mi impute chained”コマンドで実行可能となっている。

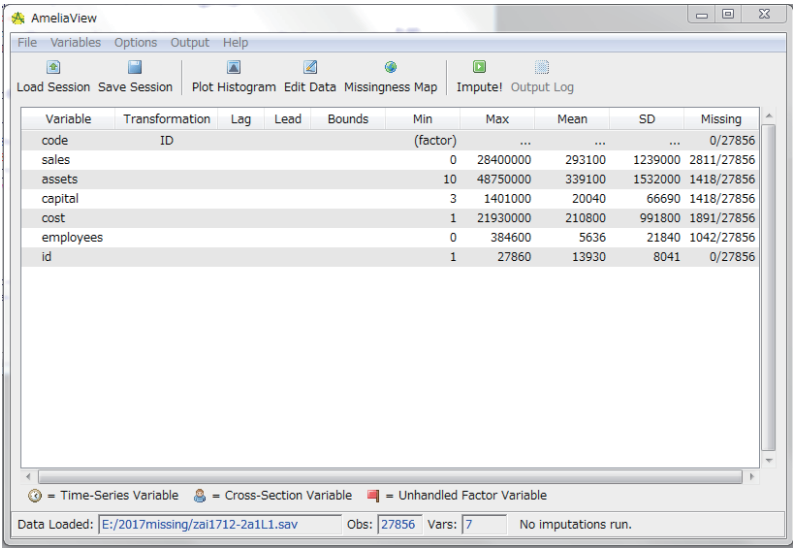


図3-2　AmeliaにSPSSの入力ファイル”zai1712-2aL1.sav”を読み込んだ画面

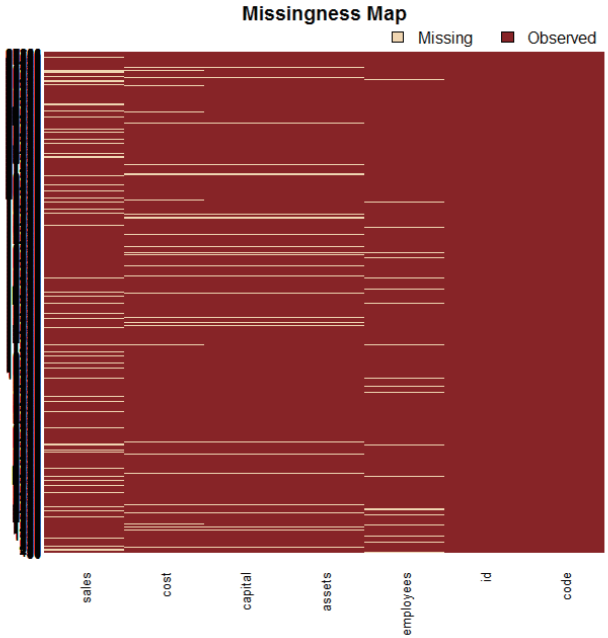


図3-3　AmeliaViewによる欠損パターンの表示

謝辞

本稿の統計ソフトウェアSASによる欠損値の補完計算において、九州大学情報基盤研究センターの多目的サーバー”najima”を利用させていただいた。

参考文献

1. 高橋将宜・伊藤孝之「様々な多重代入法アルゴリズムの比較～大規模経済系データを用いた分析～」、『統計研究彙報』第71号、2014年。
2. 阿部貴之『欠測データの統計解析』（統計解析スタンダードシリーズ）、朝倉書店、2016年。
3. 高柳慎一・井口亮・水木栄「金融データ解析の基礎」（『Useful R』シリーズVol. 8 金明哲編）共立出版、2014年。
4. 福島真太郎 『データ分析プロセス』(Useful Rシリーズ Vol. 2 金明哲編)、共立出版、2015年。
5. Patricia Berglund & Steven Heeringa, “Multiple Imputation of Missing Data Using SAS,” SAS Institute Inc., 2014.
6. Stef van Buuren,”Flexible Imputation of Missing Data,” CRC Press, 2012.