**Problem:**

After defining take\_any6 and take\_type, take\_any6==1 and take\_type missing should give no obs. But there are 113 of them.

take\_any6 = any of 6 leave types in either loop 1 or loop 2.

take\_type = type of most recent leave taken

**Investigation:**

Among 113, 99 have reason\_take = 21 (New Child), 12 have reason\_take = 20 (Unspecified)

reason\_take = most recent leave type

Among 12 with reason\_take = 20, they all have A20 = 2 so most recent (loop 2) is unspecified type. Need to impute.

Among 99 with reason\_take = 21, 7 have A20 = 1 and 92 have A20 missing so most recent (loop 1) should refer to reason\_take = 21. This is verified by the check that these 99 obs all have A5\_1\_CAT = 21. So why take\_type is missing for them?

First, take\_type is derived from take\_own, take\_matdis, … etc. So it must be that take\_[t] = 0 for all 6 types [t] for the 99 obs.

Questions boils down to:

If a row has A5\_1\_CAT = 21 and A20!=2, how can it end up with take\_[t] = 0 for all [t]?

To answer this, turn to definition of take\_[t]

take\_own relies on reason\_take = 1, so irrelevant

take\_illchild, reason\_take = 11

take\_illspouse, reason\_take = 12

take\_illparent, reason\_take = 13, all of these are irrelevant

A5\_1\_CAT = 21 can either go to take\_bond or take\_matdis

take\_bond = 1 if

A5\_1\_CAT = 21 (New Child)

AND

A11\_1 missing (seeDoctor = unknown) or   
GENDER\_CAT = 1 (male) or   
GENDER\_CAT = 2 (female) and A11\_1 = 2 (seeDoctor = no) and A5\_1\_CAT\_rev != 32 (non-maternity)

AND

A20!=2 (first loop = most recent) or A20 missing (first loop = most recent)

----------------------------------

So within A20 = 1 or missing and A5\_1\_CAT = 21 (New Child)

If sex = male, then take\_bond = 1

If sex = female, refer to below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A11\_1 (see doctor) | A5\_1\_CAT\_rev!=32 (non-maternity) | take\_bond | description | # cases / 99 |
| missing | Yes | 1 | non-maternity females caring ill baby at home | 0 |
| missing | No | 1 | maternity but maybe no doctor saw, count as take\_bond=1? | 0 |
| 1 | Yes | 0 | new child ill and doctor saw, why take\_bond = 0? | 66 |
| 1 | No | 0 | maternity and doctor saw, should refer to take\_matdis for further check | 26 |
| 2 | Yes | 1 | non-maternity females caring ill baby at home | 0 |
| 2 | No | 0 | maternity but no doctor saw, should refer to take\_matdis for further check | 7 |
| ***TOTAL*** | | | | **99** |

Up to here, we have consolidated why the 113 obs have take\_bond = 0.

Next we check take\_matdis

take\_matdis = 1 if

A5\_1\_CAT = 21 (New Child) and A11\_1 = 1 (seeDoctor = yes) and GENDER\_CAT = 2 (female) or

A5\_1\_CAT\_rev = 32 (maternity)

AND

A20! = 2 or A20 missing  
 (*note that there is no case with A5\_2\_CAT = 21 in FMLA 2012, so when define take\_matdis considering loop-1 reason is enough!)*

Problem identified! In the following generalization code (making no change to FMLA 2012 data cleaning but useful for generalizing code to later FMLA years), if condition check should assign existing take\_madis but not 0 to false cases

d['take\_matdis'] = np.where(  
 (((d['A5\_2\_CAT'] == 21) & (d['A11\_1'] == 1) & (d['GENDER\_CAT'] == 2)) | (d['A5\_1\_CAT\_rev'] == 32)) & (  
 d['A20'] == 1), 1, d['take\_matdis'])

After correction, running through code to after defining take\_any6 and take\_type, checking take\_type for take\_any6 = 1 using following

d[(d['take\_any6']==1) & (d['take\_type'].isna())][['take\_any6', 'take\_type']]

results in 14 cases:

d[(d['take\_any6']==1) & (d['take\_type'].isna())][['take\_any6', 'take\_type', 'A11\_1', 'A5\_1\_CAT\_rev', 'A5\_1\_CAT', 'GENDER\_CAT', 'A20','A5\_2\_CAT']]

take\_any6 take\_type A11\_1 A5\_1\_CAT\_rev A5\_1\_CAT GENDER\_CAT A20 A5\_2\_CAT  
91 1 NaN 1.0 13.0 13.0 2.0 2.0 20.0  
237 1 NaN 2.0 1.0 1.0 2.0 2.0 20.0  
410 1 NaN 1.0 1.0 1.0 2.0 2.0 20.0  
533 1 NaN 1.0 1.0 1.0 2.0 2.0 NaN  
536 1 NaN 1.0 1.0 1.0 1.0 2.0 20.0  
677 1 NaN 1.0 1.0 1.0 2.0 2.0 20.0  
728 1 NaN 1.0 1.0 1.0 1.0 2.0 20.0  
818 1 NaN 1.0 1.0 1.0 2.0 2.0 20.0  
819 1 NaN 1.0 31.0 21.0 2.0 2.0 20.0  
941 1 NaN 1.0 31.0 21.0 1.0 2.0 NaN  
1122 1 NaN 1.0 13.0 13.0 2.0 2.0 20.0  
1610 1 NaN 2.0 1.0 1.0 2.0 2.0 20.0  
1946 1 NaN 1.0 32.0 21.0 2.0 2.0 20.0  
2530 1 NaN 1.0 13.0 13.0 1.0 2.0 20.0

Can see that these are all A20 = 2 cases with A5\_2\_CAT = 20 (unspecified) or missing. So take\_type (most recent) would not be among the 6 leave types. We will force these 14 workers as taker with exactly one leave taken.