Creating monotone missing data in SAS data step

First, read in the standard data file

**proc** **import** file= 'C:/Users/mclaina/OneDrive - University of South Carolina/Teaching/755\_Spring\_2022/Homework/ahead.csv' out=ahead dbms=csv replace;

getnames=Yes;

**run**;

Now, create lag variable to use in missingness model:

**data** ahead\_lag;

set ahead;

iadlany\_lag = lag(iadlany);

**run**;

**proc** **gee** data=ahead\_lag desc plots=histogram;

class id;

missmodel some missingness model / type=obslevel;

model iadlany = some model / dist=bin;

repeated subject=id / corr=cs;

**run**;

514

515 proc gee data=ahead\_lag desc plots=histogram;

516 class id;

517 missmodel some missingness model / type=obslevel;

518 model iadlany = some model / dist=bin;

519 repeated subject=id / corr=cs;

520 run;

**ERROR: Non-monotone missingness is not allowed.**

NOTE: The SAS System stopped processing this step because of errors.

NOTE: PROCEDURE GEE used (Total process time):

real time 0.21 seconds

cpu time 0.18 seconds

|  |
| --- |
| The SAS System |

The GEE Procedure

| **Model Information** | |
| --- | --- |
| **Data Set** | WORK.AHEAD\_LAG |
| **Distribution** | Binomial |
| **Link Function** | Logit |
| **Dependent Variable** | iadlany |

| **Missing Data Patterns** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Group** | **Occasion** | | | | **Freq** | **Percent** |
| **1** | **2** | **3** | **4** |
| **1** | X | X | X | X | 3814 | 57.34 |
| **2** | X | X | X | . | 872 | 13.11 |
| **3** | X | X | . | X | 79 | 1.19 |
| **4** | X | X | . | . | 925 | 13.91 |
| **5** | X | . | X | X | 90 | 1.35 |
| **6** | X | . | X | . | 34 | 0.51 |
| **7** | X | . | . | X | 28 | 0.42 |
| **8** | X | . | . | . | 809 | 12.16 |

**data** ahead\_mono;

set ahead\_lag;

iadlany\_mono = **.**;

if iadlany\_lag ne **.** then iadlany\_mono=iadlany;

if year = **0** then iadlany\_mono=iadlany;

**run**;

**proc** **gee** data=ahead\_mono desc plots=histogram;

class id;

missmodel some missingness model / type=obslevel;

model iadlany\_mono = some model / dist=bin;

repeated subject=id / corr=cs;

**run**;

|  |
| --- |
| The SAS System |

The GEE Procedure

| **Model Information** | |
| --- | --- |
| **Data Set** | WORK.AHEAD\_MONO |
| **Distribution** | Binomial |
| **Link Function** | Logit |
| **Dependent Variable** | iadlany\_mono |

| **Missing Data Patterns** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Group** | **Occasion** | | | | **Freq** | **Percent** |
| **1** | **2** | **3** | **4** |
| **1** | X | X | X | X | 3814 | 57.34 |
| **2** | X | X | X | . | 872 | 13.11 |
| **3** | X | X | . | . | 1004 | 15.10 |
| **4** | X | . | . | X | 90 | 1.35 |
| **5** | X | . | . | . | 871 | 13.10 |

**data** ahead\_mono2;

set ahead\_lag;

by id;

miss = **0**;

do i=**1** to **4**;

if first.id then miss = **0**;

if iadlany eq **.** then miss = **1**;

if lag(miss) eq **1** then miss = **1**;

iadlany\_mono = iadlany;

if miss = **1** then iadlany\_mono = **.**;

end;

output;

drop i miss;

**run**;

**proc** **gee** data=ahead\_mono2 desc plots=histogram;

class id;

missmodel some missingness model / type=obslevel;

model iadlany\_mono = some model / dist=bin;

repeated subject=id / corr=cs;

**run**;

|  |
| --- |
| The SAS System |

The GEE Procedure

| **Model Information** | |
| --- | --- |
| **Data Set** | WORK.AHEAD\_MONO2 |
| **Distribution** | Binomial |
| **Link Function** | Logit |
| **Dependent Variable** | iadlany\_mono |

|  |  |
| --- | --- |
| **Number of Observations Read** | 26604 |
| **Number of Observations Used** | 20841 |
| **Number of Events** | 5555 |
| **Number of Trials** | 20841 |
| **Number of Missing Values** | 5763 |

| **Class Level Information** | | |
| --- | --- | --- |
| **Class** | **Levels** | **Values** |
| **id** | 6651 | 1 10 100 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 101 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 102 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 103 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 104 1040 1041 1042 1043 1044 ... |

| **Response Profile** | | |
| --- | --- | --- |
| **Ordered Value** | **iadlany\_mono** | **Total Frequency** |
| **1** | 1 | 5555 |
| **2** | 0 | 15286 |

|  |
| --- |
| **PROC GEE is modeling the probability that iadlany\_mono='1'.** |

| **GEE Model Information** | |
| --- | --- |
| **Correlation Structure** | Exchangeable |
| **Subject Effect** | id (6651 levels) |
| **Number of Clusters** | 6651 |
| **Clusters With Missing Values** | 2837 |
| **Correlation Matrix Dimension** | 4 |
| **Maximum Cluster Size** | 4 |
| **Minimum Cluster Size** | 1 |

| **Missing Data Patterns** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Group** | **Occasion** | | | | **Freq** | **Percent** |
| **1** | **2** | **3** | **4** |
| **1** | X | X | X | X | 3814 | 57.34 |
| **2** | X | X | X | . | 872 | 13.11 |
| **3** | X | X | . | . | 1004 | 15.10 |
| **4** | X | . | . | . | 961 | 14.45 |

Review of wide to long:

**data** BIOS755.VOUCH\_long5;

  set BIOS755.VOUCH\_wide5;

  array A\_Y(1:28) (Y101-Y114 Y201-Y214);

  array A\_timepoint(**1**:**28**) (**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2**);

  array A\_measure(**1**:**28**) (**1 2 3 4 5 6 7 8 9 10 11 12 13 14 1 2 3 4 5 6 7 8 9 10 11 12 13 14**);

  do i=**1** to **28**;

   Y = A\_Y[i];

   timepoint = A\_timepoint[i];

   measure = A\_measure[i];

   output;

  end;

  drop Y101-Y114 Y201-Y214 A\_timepoint1 - A\_timepoint28 i A\_measure1 - A\_measure28;

**run**;

When analyzing imputed data you’ll use proc mixed  or proc glimmix . We have an example for proc glimmix in the notes. We don’t have one for proc mixed though. For proc mixed it would look like the following:

proc mixed data=w.long\_imputed noclprint;

by \_imputation\_;

class collegegrad id;

model headwage = time collegegrad time\*collegegrad / solution ddfm=bw;

random intercept time / type=un subject=id;

ods output solutionf=outcombine\_random;

run;

Then combine the results:

proc mianalyze parms(classvar=full)=outcombine\_random;

class collegegrad;

modeleffects intercept time collegegrad time\*collegegrad;

ods output parameterestimates=outcombine\_random\_a;

run;

Then you can print it out:

proc print noobs data=outcombine\_random\_a;

var parm collegegrad estimate stderr tvalue probt;

run;