Complete Synthesis of the ACS data carried out for the HLG-MOS Challenge

Gillian Raab 28/01/22

Summary

- A complete synthesis of the ACS data was carried out. Including preprocessing and evaluation, this took 5 person hours of analyst's work, most of it spent in familiarising and pre-processing the data. The synthesis was carried out with the synthpop package and took 25 minutes on a Windows laptop. After an initial evaluation some tuning was carried out and the synthesis rerun, as explained below, with similar results. This took an additional 3 hours of analyst time – including doing this write up. An additional 2 hours was spent adding a somewhat ad-hoc section on disclosure control. The package *synthpop* was used for all analyses.
- A smaller report was also prepared and is submitted that carried out an incomplete synthesis of the same data.

Contents

Description of use cases and approach

The National Records of Scotland (NRS) are the Data Custodians of a wide range of statistical data for Scotland. These include demographic data, census records as well as many others. A research project (the Scottish Longitudinal Study - SLS) is situated within NRS and provides Longitudinal census data linked to other sources to researchers. The Three members of team synScotland are from different teams in NRS who have used synthetic data in different ways.

- Gillian Raab has been involved in providing synthetic data to researchers at the SLS to carry out preliminary analyses outside the safe setting within NRS and also for training courses on how to use SLS data.
- Liam Calvin comes from a team with responsibility for Census outputs. This team has used synthetic data based on the last Census (2011) to allow users to evaluate possible outputs being planned for this one.
- Christopher McCrum is involved with statistical disclosure control of Census tabulations. Synthetic data has been used by this team to evaluate rules to be applied to carry out SDC on tables.

These three use cases come under items 1, 2 and 3 of the purposes you describe in the Challenge Document. All require data with high utility. We have selected the ACS data to synthesise because it most resembles the data held in NRS,

The ACS (American Community Survey) data

I understand that the ACS is a long-form return that is part of the US Census. The ACS data were taken from the US Census Bureau's Public Use Micro data site https://usa.ipums.org/usa/index.shtml

Preprocessing

This was much the most time consuming task and involved tabulating and visualising the data. There are 1,035,201 variables and 37 variables in the original data. Also carried out a number of exploratory tables to try to understand the data. A few findings:

- Data are for ages 21+
- Totals increased from 13K in 2012 to 14K in 2018
- Important variable PUMA (Public use area) defines 181 areas with between 3K and 14K records each.
- Some very strong dependencies e.g. MARST and AGE, also RACE and HISPAN which may have structural zeros (e.g. you can't be Hispanic and only native American. Also strong relationships between income variables.
- The variable INCTOT was not the total of the other income variables – assume other sources not detailed here
- Some people had a yes for any health insurance but no other insurance was ticked. Presumably other types of insurance

Variables removed before synthesis

The data are from the public use microdata (PUMS) available for the years 2012 to 2018. I am not too sure exactly what extract they are from but they appear to have originally had some sort of synthetic longitudinal structure identified by the last variable in the data set . But we are told to ignore that. Just as well as it would have been a much more difficult synthesising job – though challenging. But we are told to ignore this so I have removed that variable and also the two weight variables. Two further variables (EMPSTAT and LABFORCE) were removed because they could be derived from EMPSTATD.

Changing variable types and assigning missing values

All variables are provided as numeric but many have codes that can be found in the documentation. For those that appear to be factors, I have converted into R factors and added levels where it was easy and helped interpretation.

I looked for missing values in numeric variables. These seem mostly to be zeros which is not best practice. The income variables were checked out and it appears that a zero there means no income from that source. I thought this was different from a low income from that source so decided to code it as a missing value. This means that during synthesis it will be synthesised first and then the other non zero values synthesised from the rest of the distribution. Some preliminary runs showed that this gave much better utility for relationships with other variables. Similarly zeros in ARRIVES and DEPARTS (the majority of all records) were set to missings. See below

There were a set of 5 variables about different types of health insurance. A composite variable was created that might make the synthesis easier.

Arrival and departure time

These were given as hours and minutes on a24 hour clock. Something is not right with them because I would have expected some departures before midnight with arrivals after. This was not the case. All departures in the 15 minutes before midnight had travel times of < 15 minutes. Some data sorting has been done to try to make all travel times positive that may have distorted tha=e data. I have ignored that but for synthesis I have used two variables ARIVAL time in minutes past midnight and time in minutes for arrival 0 departure.

Data set for synthesis

Now consisted of 27 variables, 18 categorical and 9 numeric.

Synthesis methods

First method

The synthesis was carried out using the *synthpop* function *syn.strata* with the strata defined by the 181 PUMA areas. This will make synthesis easier and also preserve the relationships of all variables with PUMA areas. The order of variables were just as they appeared as supplied or after pre-processing. The default CART synthesis was used for all variables except the first.

The code to produce the synthetic data is:

```
## values of continuous variables to be treated as special (in
##addition to any missing values)
cont.na <- list(POVERTY = 501, INCTOT = 0, INCWAGE = 0,
                INCWELFR = 0, INCINVST = 0, INCEARN =0)

t1 <- Sys.time()
stratsyn <- syn.strata(data= tidy, cont.na = cont.na, strata =
"PUMA", seed = 92345)
t2 <- Sys.time()
cat(t2-t1)  ##  took 25 minutes
```

First utility evaluation

The synthpop compare function was run for all the variables. Everything appeared OK visually, but the one=way utility measure standardises-pMSE (S_pMSE) was above the threshold we usually use of 10.o for 3 variables, The plots for the 4 variables with the worst S_pMSE don't look too bad, See Fig 1:

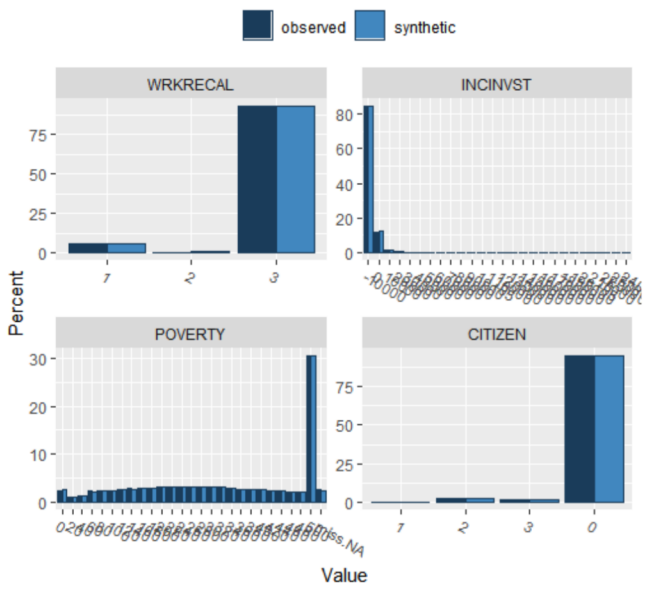


Fig1 Compare plots for the 4 variables with the worst S_pMSE.

Differences are not very evident in plots though examining the data showed that it was level 2 of WRKRECAL that seemed to have the biggest difference. For POVERTY the big spike was at 501 (treated as a special value) that included about a 3rd of all indicating an income of over 5 times the poverty threshold. A value of zero (assumed missing) was also taken out as a special value. It was noted that there were also a very large number of records with a POVERTY value of 1, and this was where the biggest discrepancy happened.

Moving on to two-way relationships, the Fig 2 gives the plot from utility.tables for the S_pMSE for all two-way tables.

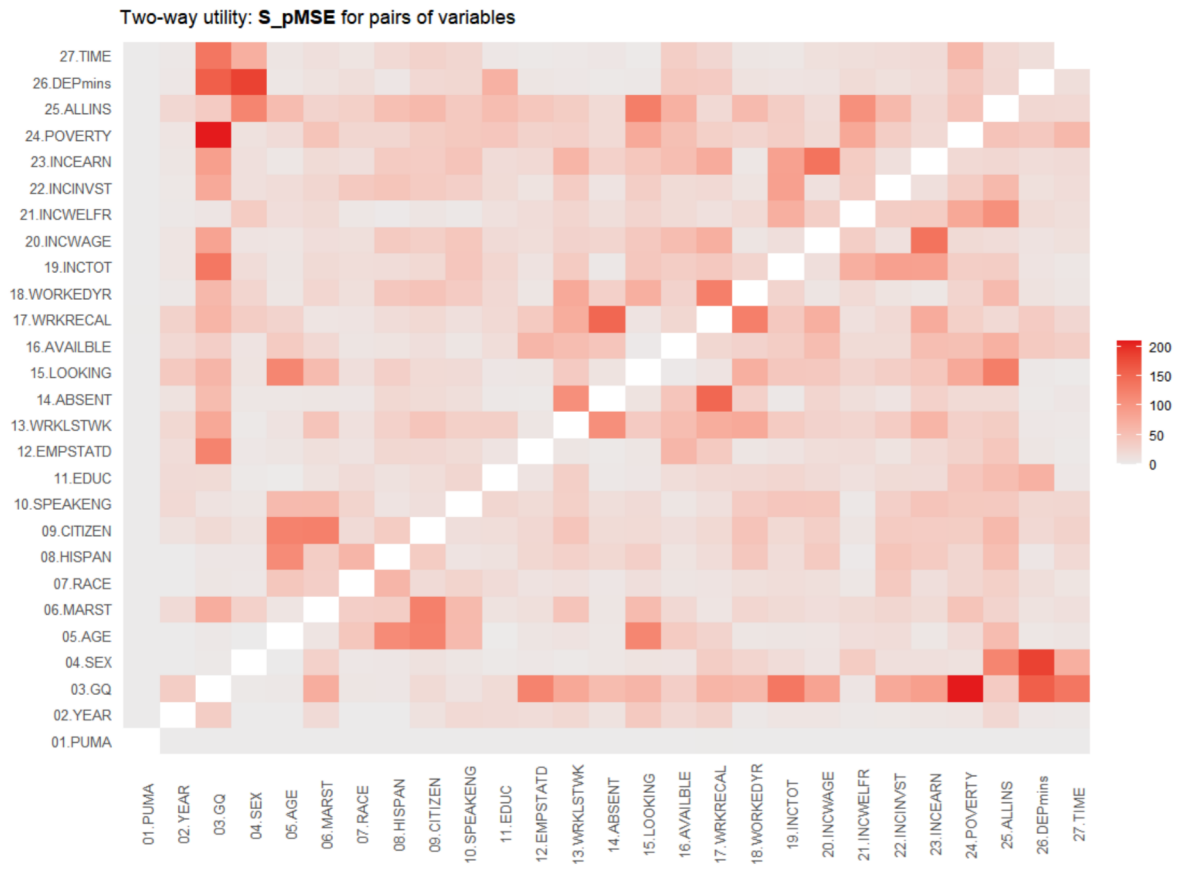


Figure 2: plot of two way relationships from first synthesis

This points to some relationships that were not well maintained. The worst was that between GQ (living in group quarters) and POVERTY as well as others. Looking at just this we can compare tables with the *utility.tab* function.

```
            GQ
POVERTY          1       2        3       4       5
  [1,126)   130531     358        0    6103      85
  [126,216) 139299     121        0     859      41
  [216,299) 137696      68        0     324      17
  [299,389) 138263      53        0     194      13
  [389,500] 138989      41        0     127       8
  501       314458      60        0     112      22
  <NA>           0       0    24839    2520       0

Synthesised:
($tab.syn)
            GQ
POVERTY          1       2        3       4       5
  [1,126)   132767     228       36    3948      53
  [126,216) 137250     119       32    1189      36
  [216,299) 137034      69       37     850      22
  [299,389) 138136      63       44     667      20
```

[389,500]	139745	59	83	590	25
501	313991	148	819	928	26
<NA>	282	2	23771	2132	0

We can see right away at least a bit of what went wrong. Living in GQ 3 sets POVERTY to NA (this had been recoded from the original 0), This could be avoided by setting a rule to force POVERTY to missing for GQ 3.

Finally the relationship between the income variables were compared for the original and synthetic data (Fig 3). In each case a random sample of 5,000 records is plotted. There are some differences, especially INCEARN and INCWAGE, which was identified as the 5th worst in the two way plot above but overall doesn't look too bad.

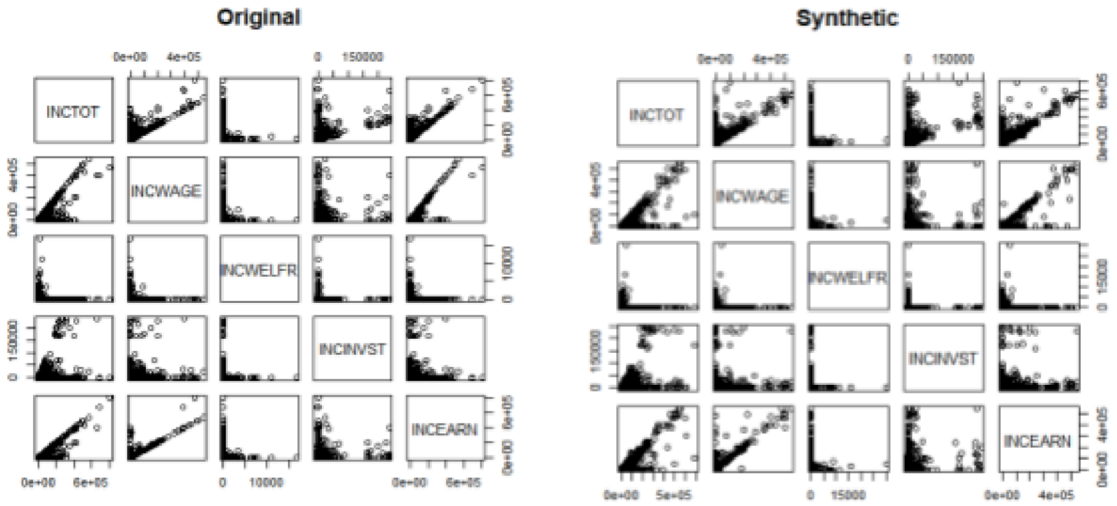


Figure 3 Pairs plot for a sample of 500 observations for 5 income variables

Second synthesis

Ideally all of these possible problems could have been investigated. Synthesis can be improved by the following methods

1. Changing the order of synthesis to place relationships of major importance near the start
2. Setting rules for tables with structural zeros
3. Pulling some values out of continuous variables if they have a meaning beyond their numeric value (999 for missing is a common example.

Given the shortage of time and my scant knowledge of the details of the data set I have only tried to fix one aspect. This is the worst two-way S_pMSE value for POVERTY and GQ. A new special value of 1 was defined for POVERTY and rules were set to force POVERTY to missing for GQ values of 3.

The univariate utilities for the variables in Figure 2 were improved for 2 of the 3 variables, though not by much and were slightly worse for one (CITIZEN). The equivalent of Figure 3 for the new synthesis looked little changed although the GQ-POVERTY measure was improved, but still not satisfactory. Looking at the comparison with *utility.tables* we can see that the rule has fixed the problem with GQ value 3, but the POVERTY profile for GQ 4 is still not correctly represented.

```
            GQ
POVERTY          1      2      3      4      5
  [2,136)   130647    291      0   4633     81
  [136,222) 134908    113      0    735     40
  [222,303) 134382     71      0    320     16
  [303,392) 136286     48      0    184     12
  [392,500] 134577     41      0    117      8
  1          13978     77      0   1618      7
  501       314458     60      0    112     22
  <NA>           0      0  24839   2520      0

Synthesised:
($tab.syn)
            GQ
POVERTY          1      2      3      4      5
  [2,136)   131342    190      0   2942     49
  [136,222) 133328    103      0   1163     30
  [222,303) 133606     73      0    776     19
  [303,392) 135834     73      0    714     21
  [392,500] 135548     76      0    601     10
  1          15241     57      0   1117     10
  501       314450    133      0    973     30
  <NA>         134      5  24526   2026      1
```

With more work to explore some of the models used in the synthesis, and with a knowledge of which of these relationships might be important this synthesis could be improved. But meanwhile this is our best effort for now.

Statistical disclosure control

The *sdc* function in synthpop should be used to create a data set to release. It allows top and bottom coding and smoothing of continuous variables, as well as the exclusion of unique records that are also unique in the original data to be excluded. It also adds labels to the synthetic data to make to make it clear that it is synthetic. The continuous variables in the ACS data had already been top-coded and banded, so this is not relevant here, But a label was added and 1,998 (0.19%) replicated- uniques were removed from the data to be released.

Evaluation of disclosure risk for one example

For completely synthetic data there are few, if any, agreed measures of disclosure risk. Most approaches define pseudo-identifiers (also called keys) that are things that might be known about a data subject as well as target variables whose value might be found by interrogating the synthetic data,

An ad-hoc approach was taken here that might mimic what someone might do to extract information from the synthetic data. Imagine this scenario:

- Someone has access to the synthetic data
- The following 18 key variables (pseudo-identifiers) are available from public sources about certain individuals "PUMA" "YEAR" "GQ" "SEX" "AGE" "MARST" "RACE" "HISPAN" "CITIZEN" "SPEAKENG" "EDUC" "EMPSTATD" "WRKLSTWK" "ABSENT"

"LOOKING" "AVAILBLE" "WRKRECAL" "WORKEDYR"

- The knows an individual who is in the original data with an unusual combination of keys
- They identify such a person in the synthetic data and find that they are unique
- They want to find out what their income is

To mimic this we can find how many individuals are unique on these 18 keys in the synthetic data that are also unique in the original. This turns out to be 92,931 (9% of all records).
We then align the original and synthetic data for these 92,931 records and compare the synthetic income with the actual income for each one.
Here are some results for the variable INCTOT. Summarising it for all the original data gives:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-12600   12000   28000   41530   52000 1178000
```

To present the results the original and synthetic incomes have been grouped into 10 categories, with the following boundaries
[-9200,5000) [12000,18400) [18400,25000) [25000,32000) [32000,40000) [40000,50000)

The grouped variables were then cross-tabulated and the original group corresponding to each unique combination of the keys expressed as the % of each synthetic group.

	Original income group									
Syn grp	grp 1	grp 2	grp 3	grp 4	grp 5	grp 6	grp 7	grp 8	grp 9	grp 10
grp 1	**57.1**	8.2	6.2	3.6	2.8	2.6	15.1	2.3	1.3	1.0
grp 2	8.3	**25.4**	15.2	10.8	7.9	6.1	15.5	5.5	3.2	2.2
grp 3	6.0	13.1	**25.9**	12.4	9.4	8.3	10.6	7.2	4.5	2.5
grp 4	4.1	9.7	13.1	**24.6**	10.9	10.6	7.4	9.6	6.2	3.8
grp 5	3.3	8.5	11.5	12.6	**22.1**	11.3	6.1	11.7	7.9	4.8
grp 6	2.7	6.4	9.2	10.6	10.3	**23.0**	5.4	13.8	11.2	7.4
grp 7	15.0	14.4	11.2	7.6	5.4	4.6	**34.4**	3.5	2.3	1.5
grp 8	1.9	4.3	7.1	8.7	9.4	13.2	3.3	**27.1**	14.2	10.8
grp 9	1.6	2.8	4.5	6.2	7.4	11.2	2.4	16.4	**27.7**	19.9
grp 10	1.2	1.7	2.9	4.1	4.4	7.3	1.7	12.0	17.4	**47.3**

As expected the variables are correlated. The numbers that are allocated to the correct decile of INCTOT are shown in red, For example, of those records found to be in the highest income group in the synthetic data, only 47.3% would be found to be in that group in the original data.

Data for release

The SDC function was used to add a label to the records for release.

A random sample of 6 records from this data set are shown here.

	flag	PUMA	YEAR	GQ	SEX	AGE	MARST	RACE	HISPAN
985214	FALSE DATA	39-901	2016	1	Female	83	Wid	White alone	0
848666	FALSE DATA	39-4602	2017	1	Female	52	Mar_wspouse	White alone	0
498736	FALSE DATA	17-501	2012	1	Male	31	Sing_nevm	White alone	0
560050	FALSE DATA	39-1300	2012	1	Female	81	Div	White alone	0
725907	FALSE DATA	39-3400	2012	1	Male	27	Sing_nevm	White alone	0
232673	FALSE DATA	17-3107	2018	1	Male	53	Mar_wspouse	White alone	0

	CITIZEN	SPEAKENG	EDUC	EMPSTATD	WRKLSTWK	ABSENT	LOOKING	AVAILBLE
985214	0	3	6	30	1	4	1	5
848666	0	3	6	30	1	1	1	5
498736	0	3	6	10	2	4	3	5
560050	0	3	6	30	1	1	1	5
725907	0	3	6	10	2	4	2	4
232673	0	3	10	10	2	4	3	5

	WRKRECAL	WORKEDYR	INCTOT	INCWAGE	INCWELFR	INCINVST	INCEARN	POVERTY
985214	3	2	24000	0	0	0	0	183
848666	3	2	0	0	0	0	0	314
498736	3	3	13000	13000	0	0	13000	109
560050	3	1	14000	0	0	0	0	113
725907	1	3	17500	19000	0	0	18400	238
232673	3	3	120000	120000	0	0	120000	501

	ALLINS	DEPmins	TIME
985214	Yes_Yes_No_No_Yes	0	0
848666	Yes_Yes_No_No_No	0	0
498736	Yes_Yes_Yes_No_No	372	42
560050	Yes_Yes_Yes_No_Yes	0	0
725907	No_No_No_No_No	332	22
232673	Yes_Yes_Yes_No_No	362	42

If these data were to be used in training then a post-processing step that will reverse the pre-processing described above would usually be used to make the data similar to what a person using the PUMS data might get.

The R code to Carry out these syntheses can be made available.

Incomplete Synthesis carried out for the HLG-MOS Challenge

Gillian Raab 26/01/22

Summary

- An incomplete synthesis of the ACS data was carried out. It uses the same preprocessing as described for complete synthesis. The report of complete synthesis describes
- Partial synthesis was carried out for only the 5 income variables. All other variables remaining the same as in the original

Incomplete synthesis of ACS data

Synthesis method

The complete synthesis of the ACS was carried out using the *synthpop* function *syn.strata* with the strata defined by the 181 PUMA areas. This approach would not work for partial/incomplete synthesis because it first generates a synthetic version of PUMA that includes different numbers of units in each synthetic PUMA. Instead the following code was used to carry out partial synthesis in each PUMA and combine all the results.

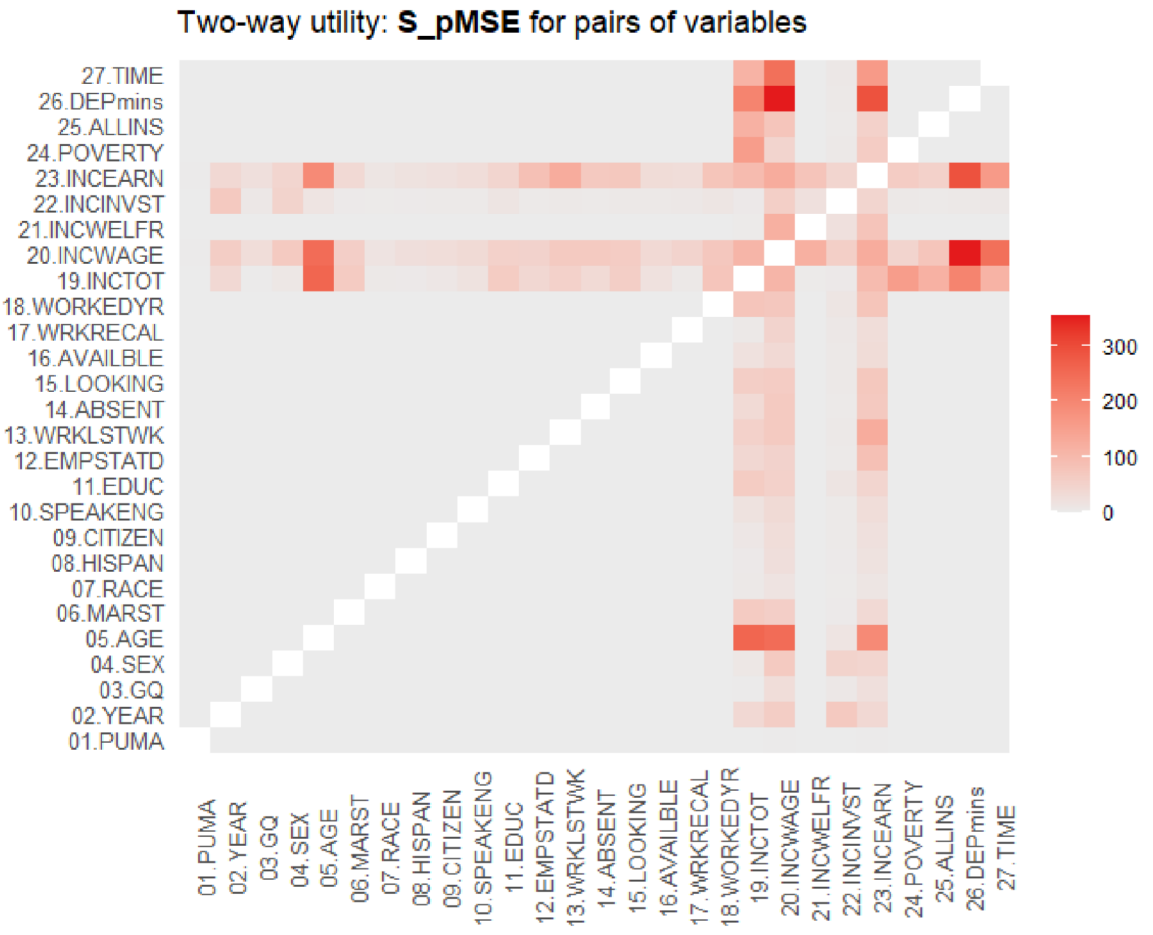
```
## tidy is the cleaned up original data
part_syn <- tidy

## this puts all the other variables before the income variables in
## the visit sequence
vseq <- c(1:18,24:27,19:23)

## sets the nethod to “” (Unchanged) to all except the
## income variables
method <- c(rep("",18),rep("ctree",5),rep("",4))

## a loop to do a partial synthesis for each PUMA and replace the
## values in the original data
for ( i in 1:181) {
  cat(i,"Now at ",i,"out of 181\n")
  res <- syn(tidy[tidy$PUMA == levels(tidy$PUMA)[i],],
  cont.na = cont.na, method = method, visit.sequence = vseq)
  part_syn[part_syn$PUMA == levels(part_syn$PUMA)[i],19:23 ] <-
                          res$syn[,19:23]
}
```

This code took 7 minutes for one run 10 minutes another time, to produce the synthetic data.

A brief look at utility

Fig 2 gives the plot from utility.tables for the S_pMSE for all two-way tables from the partial synthesis. As expected the unsynthesised variables all score zero on the S_pMSE (low is good).

Figure 2: plot of two way relationships for partial synthesis



This points to some relationships that were not well maintained between income and some other variables, especially departure time and sex. Reordering the variables and adjusting the predictor variables and perhaps further, or different, stratification might fix this.

Time did not allow this nor any work on assessing disclosure risk – but something similar to what was done for complete synthesis might be helpful.