Principles of I M P U T A T I O N M E T H O D S

University of Helsinki 2015

Seppo Laaksonen

Content

What is imputation, its purpose, concepts
Missingness (puuttuneisuus) mechanisms
Most common tools for missing item handling without real imputations
Missingness pattern
Targets for imputation
Imputation process
Imputation model
Imputation task
Single and multiple imputation (yksikertainen ja monikertainen imputointi)
Imputation model plus Imputation task in the case of the linear regression model
Imputation model plus Imputation task In the case of the response indicator model
Preserving associations in the case of missing data
General conclusion

After the introduction we go to details of each method including SAS codes and an instruction to SPSS multiple imputation.
A reference to my papers on the last page
And we will have Training all the time.

What is imputation?

It is to insert a value into the data in a more or less fabricated way ('best proxy'). Why?

- Since there is no value in this cell, that is, it is completely missing.
- Since the existing value is partially missing (like given as an interval) but this is desired to replace with a good unique value e.g. for distribution purposes.
- Since the existing value does not seem to be correct, and consequently, it is desired to get a more reliable value by replacing with a more plausible value.
- Since the current value seems to be too confidential, that is, and this individual unit should be disclosed. Motivation: the fabricated (imputed) value can be considered as less problematic even when told that it is no true value.

Imputation can be performed both for the macro and micro data but during this course I only consider the imputation methods of micro data. However, basically the same methods can be applied to macro data but usually this imputation is more limited, i.e. simpler methods are enough.

Purpose of imputation

To repeat: The purpose of imputation is twofold

-Either to replace a missing or partially missing or incorrect value with a such value that the estimate derived from this variable will be more valuable than without imputation. Thus if imputation is advantageous from an estimation point of view, use it. Naturally, there are in surveys several estimation tasks and can be possible that a certain imputation is not advantageous in all respects. Hence, it is possible that some estimates are computed without imputation and some others with imputation. On the other hand, a big question is which imputation is best for each estimation. It is good to notice also that a bad imputation may worsen the estimation. Be careful! You thus have to convince yourself or your client that imputation improves something.

- Or to make data more confidential. This leads to create certain incorrect values into the data that is not difficult but this should not be a purpose but to impute the confidential values so that their pattern gives opportunity to get as the reliable estimates as possible.

Use of imputation has increased

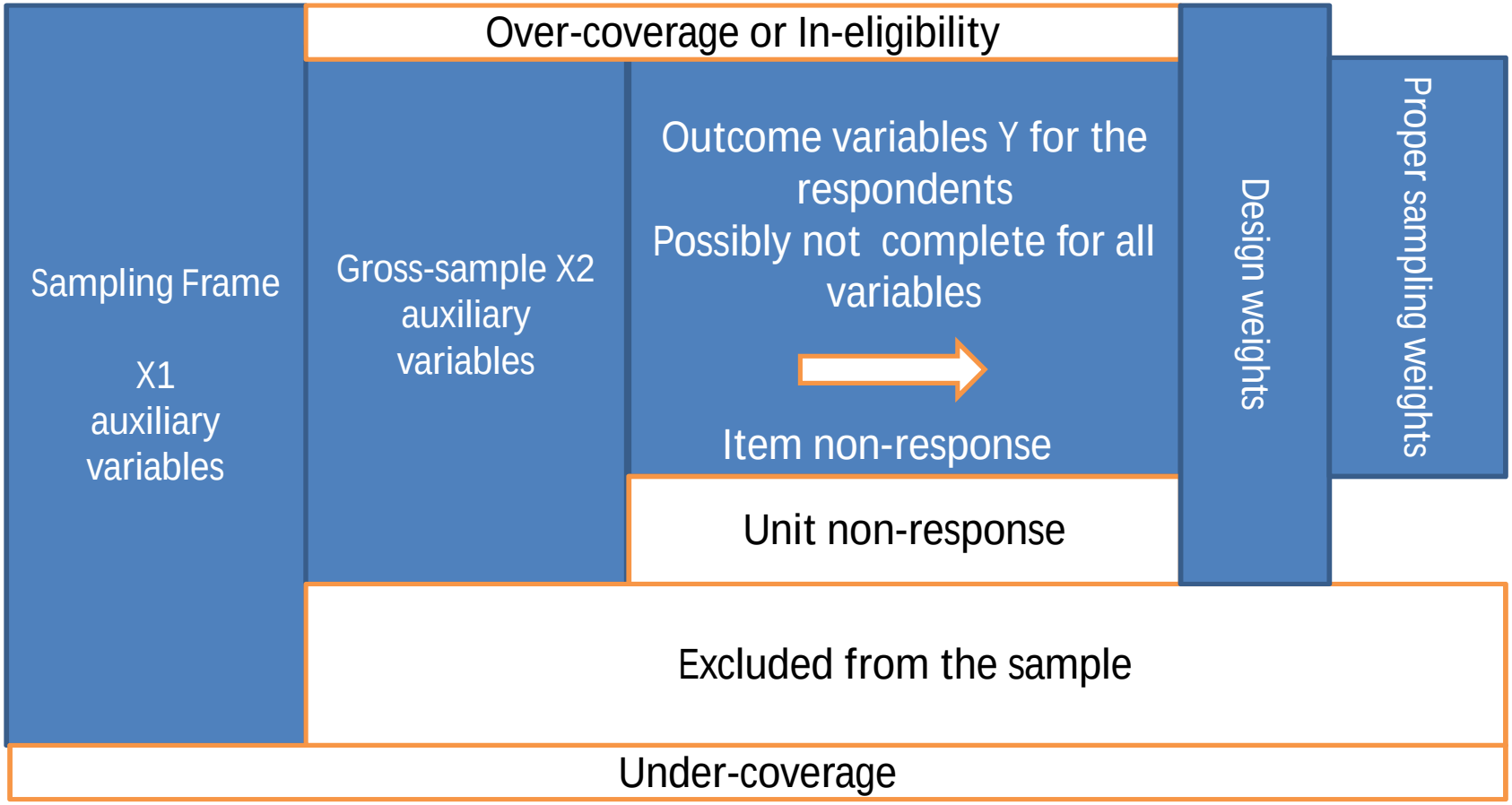
- Since missingness and data deficiencies have become more common and also statistical confidentiality is more important.

- Since methodology has been developed but its implementation into software is not satisfactory. Hence, many imputations in data institutions are still needed to do using a specific programming. Some methods are fortunately easy to program but some not. Most methods I will present are not difficult to perform with SAS codes that I use.

- Imputation research was flourishing in 1990's and early 2000's but recently very little new things have been invented. Interestingly, the results of the Euredit project (http://www.cs.york.ac.uk/euredit/) are still useful. This project in which I was involved, tested a big number of imputation techniques, called traditional and new methods, respectively. I will concentrate mainly on traditional methods. Since new projects have been missing, less new ideas have been developed but certain techniques have been however implemented in software, like SAS MI, SPSS, Solas, MICE. Any general imputation software do not exist. Thus if imputation is wished to use, the understanding of its methodology is necessary. Do not believe any automatic software even though you might get results without problems.

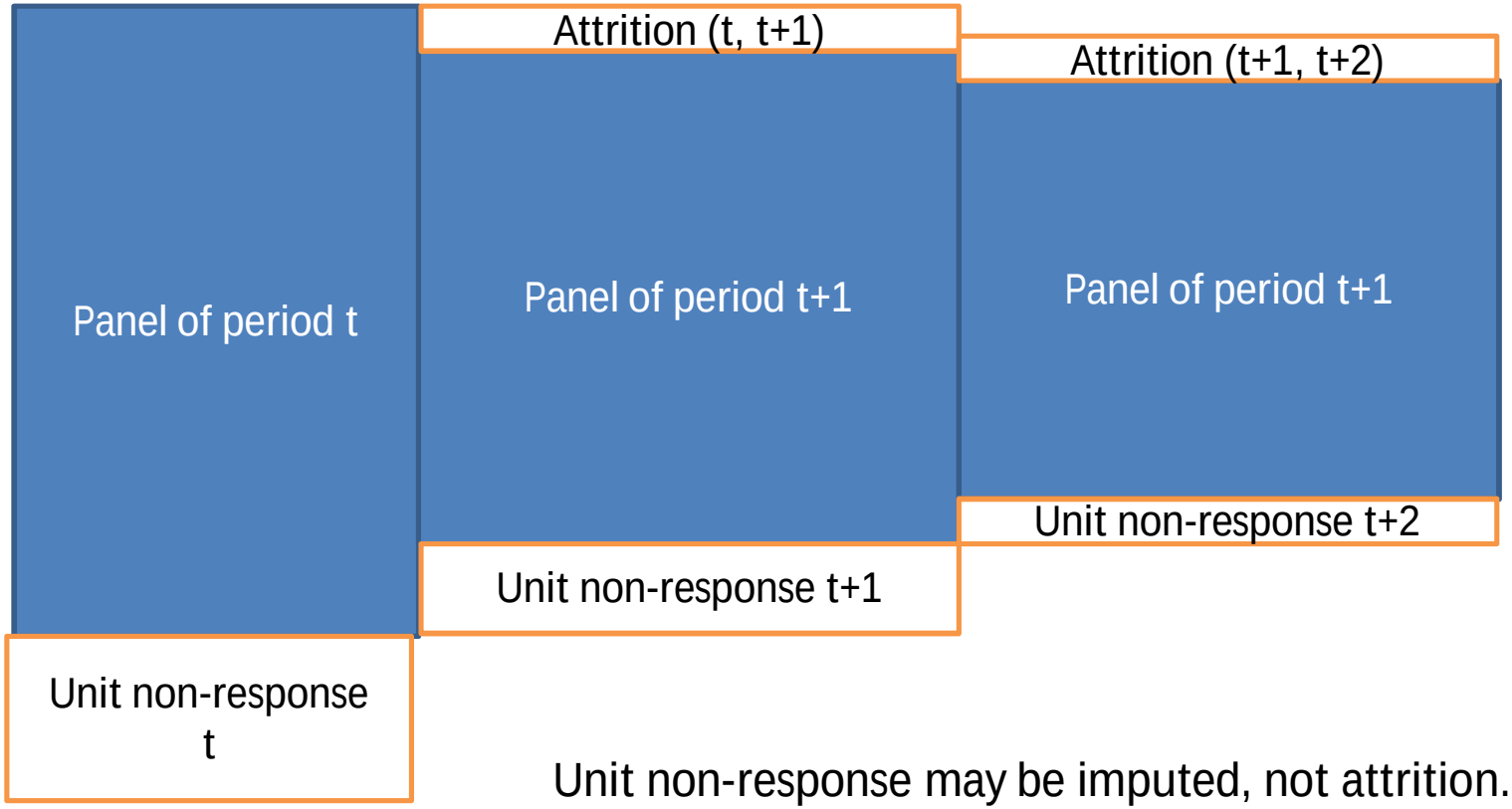
Micro data and Missingness

Now I focus on micro data where one can see various types of missingness. This is a cross-sectional case ('white boxes' are missing values and their values may be imputed using information from 'blue boxes'):



Micro data and Missingness

Cohort type of panel example



Unit non-response may be imputed, not attrition.

Information requirements for imputation

If any explanatory variable (auxiliary variable, covariate) does not exist, imputation can only be random based, i.e. guessing randomly missing values. This rarely works, but usually, it is needed auxiliary variables that predict missingness as well as possible. You can look the two previous pages and see which variables can be used, i.e. such variables that are non-missing. In panels or longitudinal data there are more such variables since e.g. the variables of the previous waves are available (a problem is still the fact that this variable has been changed).

What can be imputed due to missingness?

When looking for those schemes, we can find the following possible imputation affairs:

(i) Under-coverage that requires a new up-to-date frame. Very seldom possible.
(ii) Those units that are not selected into the sample. Done in theoretical (simulation) studies
(iii) Unit non-response, all or some variables. If done, called mass imputation. This is competitive to weighting methods.
(iv) Item non-response. This is the most common case.
(v) Deficient and sensitive values. Quite common.
(vi) Second, third etc wave missing values in cohort studies given that the previous value exists (or earlier imputed correctly enough).

Missingness mechanisms 1

Imputation requires useful auxiliary information. Without such data imputation is still possible but results are maybe bad. On the other hand, it is important to assess the missingness (or response) mechanism.
There are four basic mechanisms good to think and make assumptions before starting the imputation (usually only three of these are presented in literature):

MCAR (Missing Completely At Random): If this could be reality, it is rather easy to decide which methods to apply. Most methods are workable and you do not need auxiliary variables either. Simplest imputation methods follow this assumption.

MARS (Missing At Random Under Sampling Design): Now missingness only depends on the sampling design variables. This is often used so that one assume that MCAR holds true within strata (pre-strata, or even post-strata). Here imputation is performed by strata.

Missingness mechanisms 2

MAR (Missing At Random (Conditionally)): Now missingness depends on both the sampling design variables and all possible other auxiliary variables. This assumption is much used when good auxiliary variables are available. It is a basic assumption in imputations when implementing an imputation model (later).

MNAR (Missing Not At Random): Unfortunately this is the most common case in real-life to some extent. So, when all the auxiliary variables have been exploited, the quality of the estimates have been improved but still it is rather clear that our results are not ideal. So, it is good to interpret possible biases in results against general knowledge and lack of good auxiliaries (unfortunately).

Most common tools for missing item handling without real imputation

(i) In the case of mass missingness, the weighting or the reweighting is mostly exploited. This is possible only for the respondents. The respective imputed data thus covers the non-respondents too (or those non-respondents desired to include in estimation). Note that one imputation strategy is a kind of weighting method but its weights are more flexible than the standard reweighted sampling weights.

Most common tools for missing item handling without real imputation 2

(ii) Item-non-response is marked with a good and well-covered code, e.g.:

- -1 = respondent candidate not contacted (a problem here may be that we do not know whether this unit belongs to the target population). Very seldom these cases are imputed.
- -2 = respondent refused to answer (main reason for imputation)
- -3 = respondent was not able to give a correct answer
- -4 = missing for other reasons
- -6 = question was not asked from the respondent (imputation using logical rules)
- -9 = question does not concern the respondent

These codes are not much used but such as 7, 8, 9, 66, 77, 88, 99 instead. The negative values are easy to observe. Do not use a zero (0)!

Most common tools for missing item handling without real imputation 3

(ii) cont.

The good and illustrative codes are useful also when deciding the imputation methods itself. When going to impute, it is good to try a different imputation technique for each missingness code, since the nature of these units are different. I think that this is rarely applied in this way. Question: how to 'impute' cases with the codes -6 and -9?

Moreover, it is good to notice that the coded variable is full, without missing values. This kind of a categorical variable can be used as an explanatory variable in standard linear and linearised models, among others. But if desired to use it as continuous, real imputation is required.

Most common tools for missing item handling without real imputation 4

(iii) The values with missing codes are excluded from each analysis so that the observation number may vary by variable.

(iv) Close to case (iii) but now the units with missing values have been excluded from each analysis. In this latter case, there are always the same number of observations. The standard multi-dimensional analysis makes this automatically for those variable patterns that are used in the multidimensional analysis. This strategy gives consistent results with each other. This strategy does not give consistent results with each other. Called 'case deletion.' In think that this is still a fairly common strategy.

(v) Pair-wise analysis for multivariate purposes in such cases where e.g. the correlations are the basis for further analysis. This operation first computes pair-wise correlations like in case (iii) and when continues from the correlation matrix towards multivariate analysis. We lose less information here than in (iv).

Example: Item non-response

It is useful before imputation to examine how nonresponse vary. For the imputation of a pattern of possibly missing values is good to compute their item response rates. Here is an example using the European Social Survey and selecting some different variables. The below example with SAS codes illustrate the computation, first for creating the item response indicators:

```
libname aa ’z:\ess’;
data ess1_6; set aa.ess16_b ;
if hincfel<5 then sub_inc_res=1; else sub_inc_res=0;
if income<11 then income_res=1; else income_res=0;
if eisced<=5 then education_res=1; else education_res=0;
if happy<=10 then happy_res=1; else happy_res=0;
if imsmetn<=4 then immigration_res=1; else immigration_res=0;
if sclact<=5 then social_res=1; else social_res=0;
if lrscale<=10 then left_right_res=1; else left_right_res=0;
if cregion ne ' ' then region_res=1; else region_res=0;
if trstep <=10 then eu_parl_res=1; else eu_parl_res=0;
if vote=3 then vote_res=.; else if vote<=2 then vote_res=1; else vote_res=0;
```

Its is easiest to get the basic figures from these rates by calculating the means. Note that the last variable does not include those who were not eligible to vote. Hence the number of the observations is lower. The same might be concerned other variables in surveys like the satisfaction in job in which case whose not working is excluded.

The SAS System

The MEANS Procedure

Variable	N	Mean
sub_inc_res	291686	0.9821246
income_res	291686	0.7925646
education_res	291686	0.8486592
happy_res	291686	0.9924542
immigration_res	291686	0.9620208
social_res	291686	0.9761353
left_right_res	291686	0.8497322
region_res	291686	0.3587248
eu_parl_res	291686	0.8679059
vote_res	271362	0.9891289

These rates are one-dimensional but it is often good to know the same multidimensionally. In this case the pattern could be calculated as the next page shows.

Using SAS is the FREQ procedure is maybe the easiest way to get the whole pattern of the item response rates. I do not include all the previous variables here since the respective table would be too huge. The result can be seen from the file of out. In order to reduce the prints, ‘noprint’ option is used.

```
proc freq; tables sub_inc_res* income_res* education_res* happy_res*
immigration_res* social_res/noprint out=item_res;run;
```

The following page shows the ‘out’ file sorted by the frequency count. This helps in seeing different combinations better. If several variables is required to impute, this pattern helps in selecting the order. There is no definite order for this, but often it is good start from variables that do not needed many imputes and continue so that these imputed variables are used as covariates or auxiliary variables for the next variables being imputed. Another strategy is such in which best possible auxiliary variables can be used in each imputation. The compromise of both is maybe the ideal strategy. Think these questions but our practice later on does not include these sequential imputations.

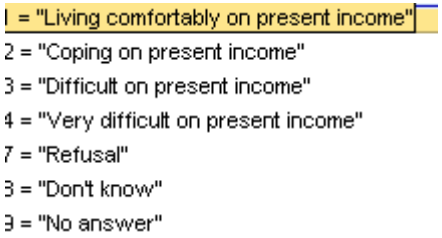
Item non-response pattern for some variables

5! = 5*4*3*2*1=
120 = the maximum number of combinations
>> 32

	sub_inc_res	income_res	education_res	happy_res	immigration_res	Frequency Count	Percent of Total Frequency
1	0	1	0	0	0	2	0.0006856688
2	0	1	0	0	1	3	0.0010285033
3	0	1	0	1	0	8	0.0027426753
4	0	1	1	0	0	14	0.0047996818
5	0	1	1	0	1	18	0.0061710195
6	1	0	0	0	0	18	0.0061710195
7	0	0	0	0	1	23	0.0078851916
8	1	1	0	0	0	35	0.0119992046
9	0	0	0	0	0	36	0.012342039
10	0	0	1	0	0	40	0.0137133767
11	0	0	1	0	1	48	0.0164560521
12	0	0	0	1	0	51	0.0174845553
13	1	0	0	0	1	87	0.0298265944
14	0	1	0	1	1	98	0.0335977729
15	1	1	0	0	1	142	0.0486824873
16	1	0	1	0	0	143	0.0490253218
17	0	1	1	1	0	154	0.0527965003
18	0	0	1	1	0	256	0.087765611
19	1	1	1	0	0	297	0.1018218221
20	1	0	1	0	1	354	0.1213633839
21	0	0	0	1	1	368	0.1261630658
22	1	0	0	1	0	386	0.1323340853
23	1	1	0	1	0	846	0.2900379175
24	1	1	1	0	1	941	0.3226071872
25	0	0	1	1	1	1989	0.6818976571
26	0	1	1	1	1	2106	0.722009284
27	1	0	1	1	0	2544	0.872170759
28	1	1	1	1	0	6248	2.1420294426
29	1	0	0	1	1	7750	2.6569667382
30	1	1	0	1	1	34291	11.756135022
31	1	0	1	1	1	46413	15.911973835
32	1	1	1	1	1	185977	63.759316525

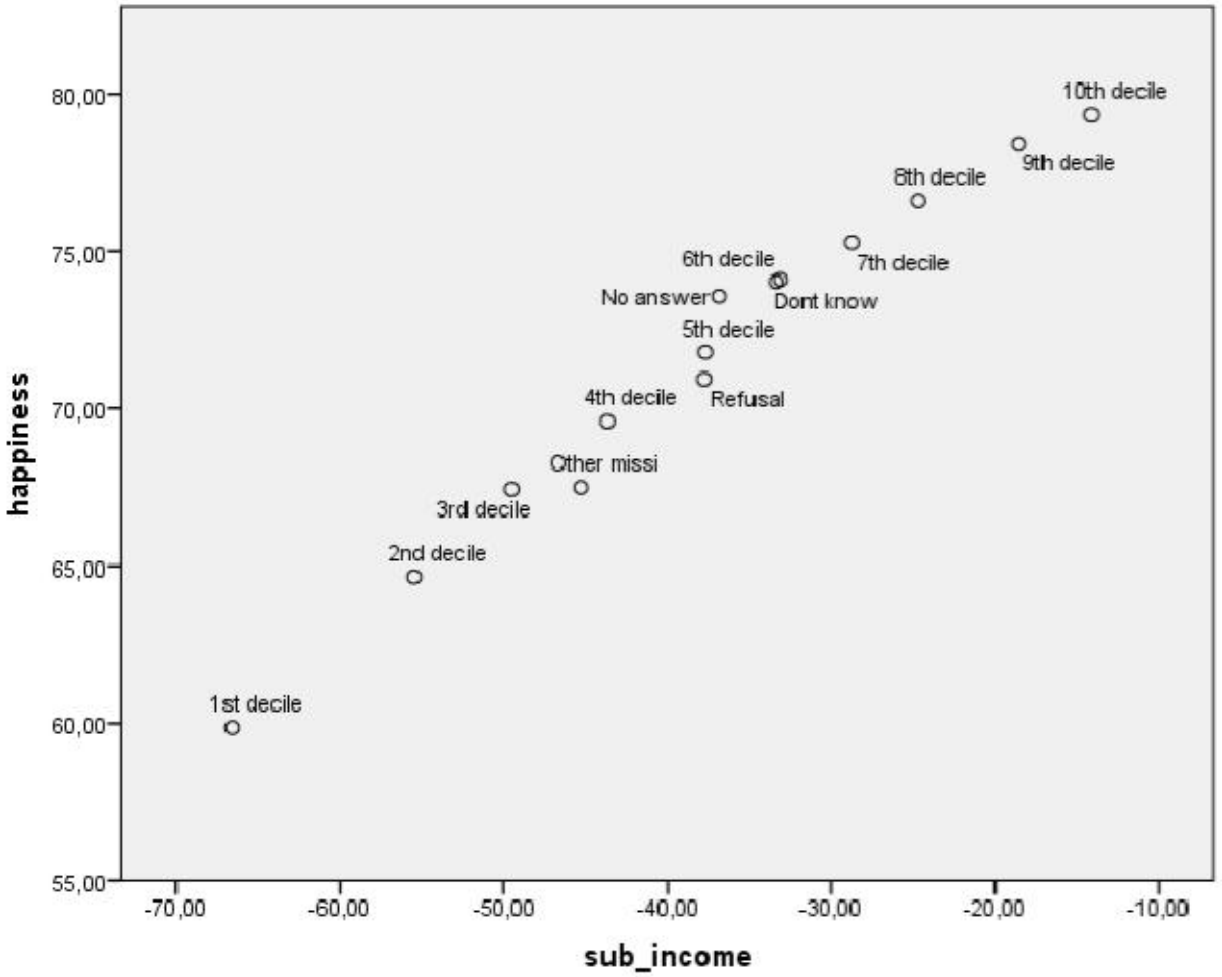
Example ’aggregate imputation’

This is also from the European Social Survey where we have found that the item non-response rates are low for the subjective income
= Feeling about household's income nowadays.
I have scaled this linearly into [0, 100].
The other variable is the happiness, scaled into [0, 10].



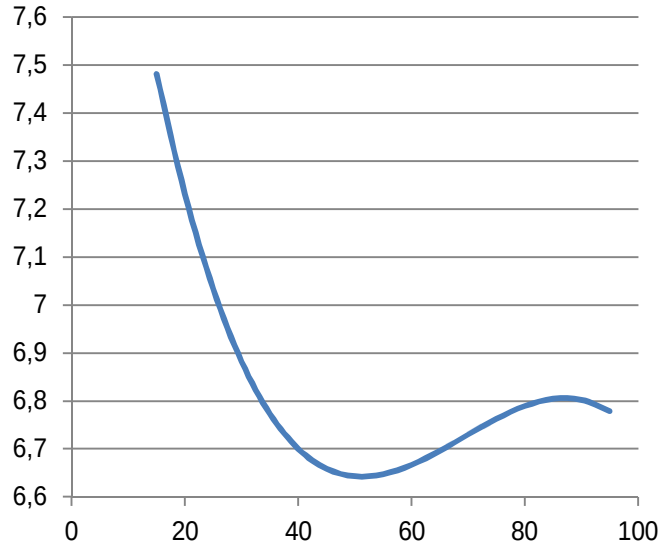
This rate is much higher for objective income. Hence it is possible to get some understanding about this income via those two former variables with low non-response.
The next page graph shows their scatter plot so that each plot is in the location in which the average of both variables. Try to interpret the missingness categories in particular.

Scatter plot subjective income vs happiness with plots of objective income groups ‘Aggregate imputation’



Example in which Happiness are explained by several variables, one being income as before (others, education, country, ESS round, gender). Some income values are lacking but the two missingness codes are here given. This thus is a kind of imputation since any big problems in regression models due to missingness does not occur. Now I can have two results below. Look at how the happiness is explained by income and how the happiness profile is by age (estimated using the age variables with the three powers, i.e. one, two and three

Income level	Regression Estimate	Std. error	p-value
Lowest	-0,9775	0,02067	<,0001
2	-0,57	0,01881	<,0001
3	-0,3015	0,01779	<,0001
4	-0,1774	0,01659	<,0001
5	-0,0497	0,01664	0,0028
6	0,09624	0,0171	<,0001
7	0,17428	0,01746	<,0001
8	0,26972	0,01781	<,0001
9	0,34471	0,01717	<,0001
10	0,46527	0,01907	<,0001
11	0,32817	0,0436	<,0001
Highest	0,46456	0,05285	<,0001
Refusal	0,03671	0,01575	0,0198
Other item-nonresponse	0	,	,



Happiness by age

Targets for imputation should be specified clearly

It is rather clear (except when imputation aims at protecting data)

(i) That a user is happy if the imputed values are as close as possible to the correct/true values. Success at individual level. Another point is that how to know how close they are, except in some cases. This may be often a too demanding target and hence

(ii) A user is still fairly happy if the distribution of the imputed values is close to the distribution obtained from true values. Success at distributional level. Of course this is hard to check but however easier than case (i).

(iii) The target to succeed at aggregate level is also satisfactory and specifically in NSI's or in other survey institutes where such estimates as average, total, ratio, median, point of decile and standard deviation are typical.

(iv) Some users hope to get the order of imputed values as correct as possible.

(v) Finally, success to preserve associations (like correlations) is also important in many studies.

The summary: it is most important to keep in mind the end use of the data set after imputation as well.

In our training

We are able to check everything since we know true values. Respectively we can look for the first three criteria, thus even though how well the imputation succeeds at individual level. For this purpose we compute the mean absolute error for the units with imputed values (the same can be done for the entire data but it is as illustrative.

$$MAE = \frac{\sum_{i=1}^{n-r} \left| y_{*i} - y_i \right|}{n-r}$$

in the formula y_* refers to an imputed value, *y to a respective true value,* r = the number of the observed units and *n* = the number of all the units

The same in SAS codes:

```
data imp2; set  imp;
mae=abs(income_imp-income);
proc means n mean; var mae; run;
```

In our training 2

This is not however very important criterion in our case especially since it is difficult to succeed well at individual level since we no good micro level auxiliary variable in our data set. This is usual in real life but sometimes its is possible to get a good tax variable that correlates well with income. In real life such a variable works with ordinary socio-economic groups but not with all.
It is fortunate the individual level success is not most important but the two other variable indicators in stead:

- the average income

and

- income differences.

The latter one may be considered even more important than the average. Income differences can be measured by various indicators but the simplest is the coefficient of variation (CV). This basic statistic is well correlated with the Gini coefficient e.g. that is the mostly used. The income differences can be looked via different distributional statistics as well. The similar criteria are appropriate for happiness as well.

Imputation process

Imputation is part of the data cleaning process. It can be considered to cover the following 6 actions:

(i) Basic data editing in which part the values desired to impute are also determined.

(ii) Auxiliary data acquisition and service incl. preliminary ideas to exploite these.

(iii) Imputation model(s): specification, estimation, outputs

(iv) Imputation task(s): use outputs of the model for imputation, possible re-editing if the imputed data are not clean and consistent.

(v) Estimation: point-estimates, variance estimation = sampling variance plus imputation variance.

(vi) Creation of the completed data (or several data): includes good meta data such as flagging of imputed values, documenting of the whole imputation procedure and deciding what to give outsiders.

Imputation model

Imputation model should be integrated strictly to the next step, that is, to imputation task. There are two options to determine the specification of the imputation model:

• To determine the model using smart information so that it predicts well the case required to impute. The model may a deterministic (or stochastic) function like $y = f(x)$ (+ e) or a rule (like in editing) such as 'if so and so but not so then it is that.'

• To estimate the model using either the same data required to impute or other data that is similar (at least the structure) to the present data.
The previous models are often used in simple (conservative) imputations and in the same step as editing. Next I will focus on the latter models.

A strategy: First, try to impute using the first alternative as well as possible = logical imputation, and second, to impute using the second alternative the rest; naturally if you will impute at all.

Imputation model 2

This second type of imputation model is always such in which it is purpose to predict something using auxiliary variables as independent variables.

The dependent variable of this imputation model can be of the two types only:

(i) either the variable being imputed itself

or

(ii) the missingness indicator of this variable.

Case (i) can cover all possible forms, categorical including binary and continuous but in case (ii) the variable is binary.

Imputation model 3

These two models are estimated from the two different data sets:
(i) From the respondents (observed units)

(ii) Both from the respondents and the non-respondents.

But of course, the explanatory variables should be available from both the respondents and the non-respondents. Note my earlier comment that a categorical variable with the missingness codes may work reasonably in the imputation but many such variables maybe not unless these are concerned the different units.

Note that in sequential imputation the number of non-respondents (missing value units) will be declining from one imputation to the next. In order to work well in this imputation, individual level success is important or such aggregate level that is important.

Imputation model 4

The model (i) is concerned a continuous variable (as income in our first training).

In this case the most common model is linear regression or its logarithmic version. Recently also mixed models are going to applied and these models may be better than linear if the measurements are from two levels for example. In this course we do work with mixed models since our training data are from one level, i.e. it is concerned individuals.

Regression models are easy to use and also the model fit (*R-square*) is a good indicator and it is good to look when searching for best auxiliary variables or covariates in the model specification phase. This will be the first real operation when going to imputation. Its result can be used in the imputation models (ii) as well. It is useful also for comparing different methods with each other.

Imputation model 5

The model (ii) is concerned a binary variable (1 = responded, 0 = not) but the same model can used for the model (i) if the dependent variable is binary (e.g. 1 = employed, 0 = unemployed).

You know how to work with the binary model to predict. First you have to choose a link function, that can be:

-logit
-probit
-complementary log-log
-log-log .

There are no dramatic differences in explaining models between those link functions but of course with some. Imputation thus requires to use this model for predicting the response propensities for all units (respondents and non-respondents). That is, the first outputs are those values between (0, 1).

Imputation model 6

In addition to ordinary models such as linear regression or probit regression, the imputation model can be nonlinear and nonparametric. An interesting example of the latter ones is *tree modeling*. If the dependent variable is categorical, we speak about *classification trees* (*random forests* is its newer version), whereas the model for continuous variable is *regression tree*. Moreover, neural nets often create analogous groups of the gross sample. This kind of a group is called in imputation terminology as *imputation class* or *imputation cell*.

Imputation cells can also be constructed manually or using smart statistical thinking. For example, strata or post-strata can be rather good imputation cells. Given that the imputation cells are homogenous from the imputational points of view (especially if MCAR holds true within cells), these offer many advantages. Imputation cells can be constructed with 'smart thinking', e.g. the model (i) or (ii) can be estimated two times by gender if though that the predictions vary by gender. Or regions and age groups can be good as well. If someone wishes to do so in our training that would be nice.

Imputation model 7

Both types of imputation models thus have been estimated in a best way in the sense that it predicts well so that the final target is imputation. The guru's of imputations have said that the imputation model should have a good predictability feature that is not necessarily easy to know what this means. We can say that this means at least that it is not necessary to concentrate on a model that is explaining well the dependent variable of the multivariate model. Naturally, it may be good if the estimated model coefficients of the explanatory (auxiliary) variables or covariates can be interpreted well since it helps in explaining for clients or reviewers why imputation is obviously working well. Keep still in mind the predictability. Hence we have to get the predicted values of the models before going on to the next step, imputation task.

Next page gives the basic SAS codes for both the linear regression model and for the binary regression model with the most common link functions, i.e. logit and probit.

Imputation model 8

SAS codes with predicted values in the output file

Symbols: y1 = continuous variable

y1_res = response indicator of y1

x1, x2, x3, ... = continuous auxiliary variables

z1, z2, z3, ... = categorical auxiliary variables or those used categorically, * =interaction between two variables

Linear regression

```
proc glm data=a.impucomplete; class z1 z2 z3 z4 ; model income2=z1
z2 z3 z3*z4 x1 x1*x1 /solution ;output out=new p=predicted; run;
proc means n mean min max cv data=new; var predicted; run;
```

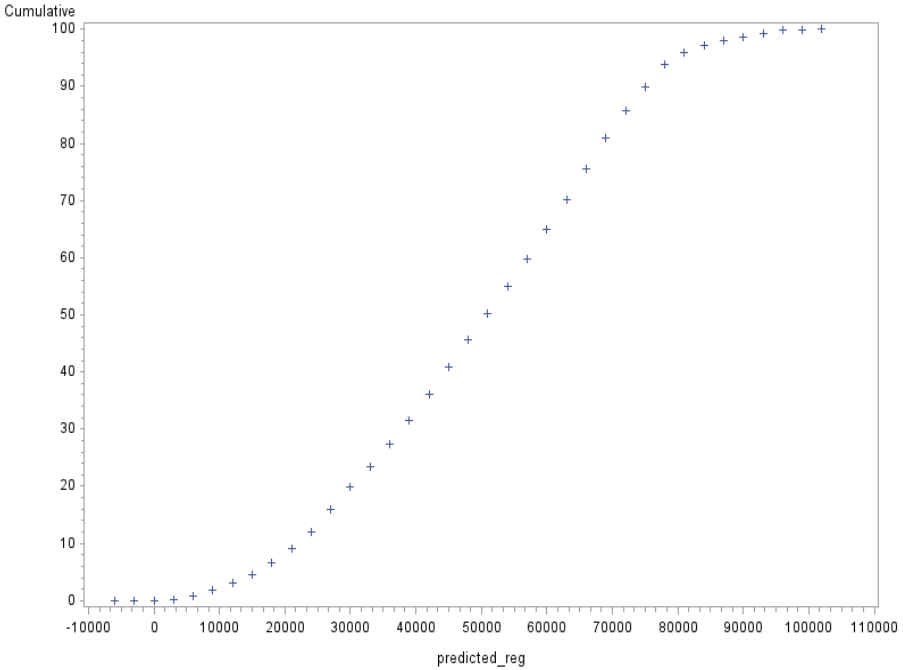
Logit regression or logistic regression

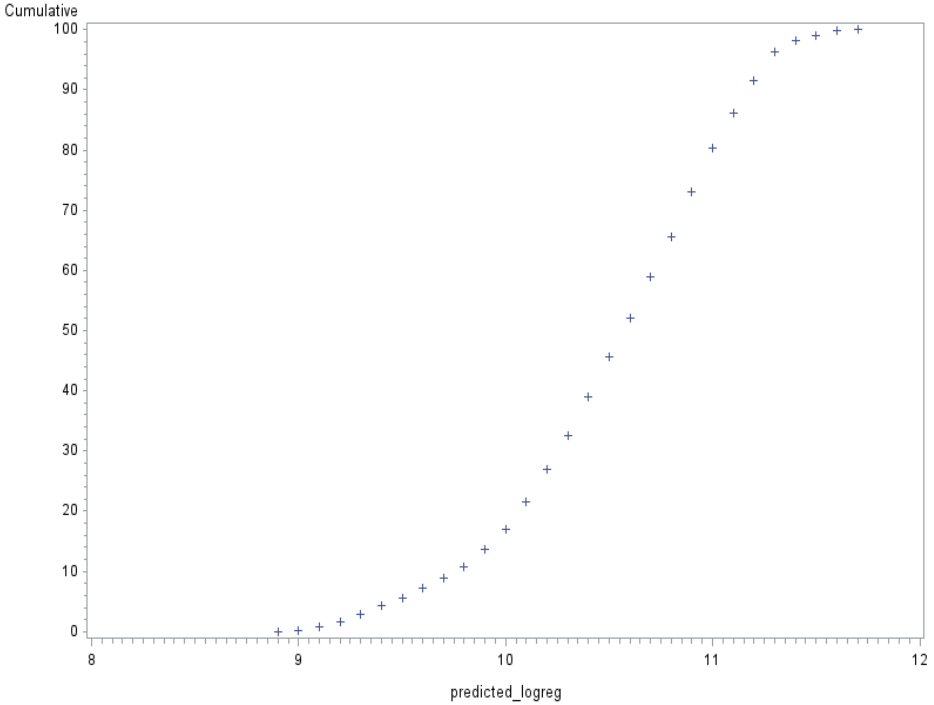
```
proc genmod data=a.impucomplete descending; class z1 z2 z3 z4 ;
model income_res=z1 z2 z3 z3*z4 x1 x1*x1 /link=logit dist=bin type3;
output out=new2 p=predicted; run;
proc means n mean min max cv data=new2; var predicted; run;
```

Probit regression as above but replace ‘logit’ with ‘probit’.

In our training, you thus have to choose the auxiliary variables. As seen, the same choice will be used in all models.

Cumulative frequencies of the predicted values for regression models

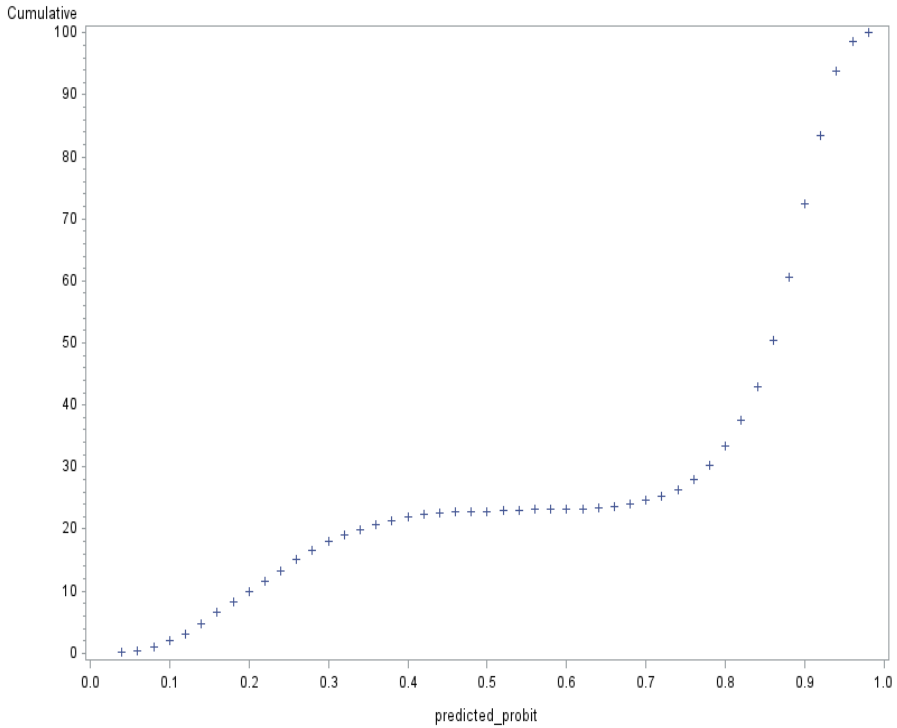




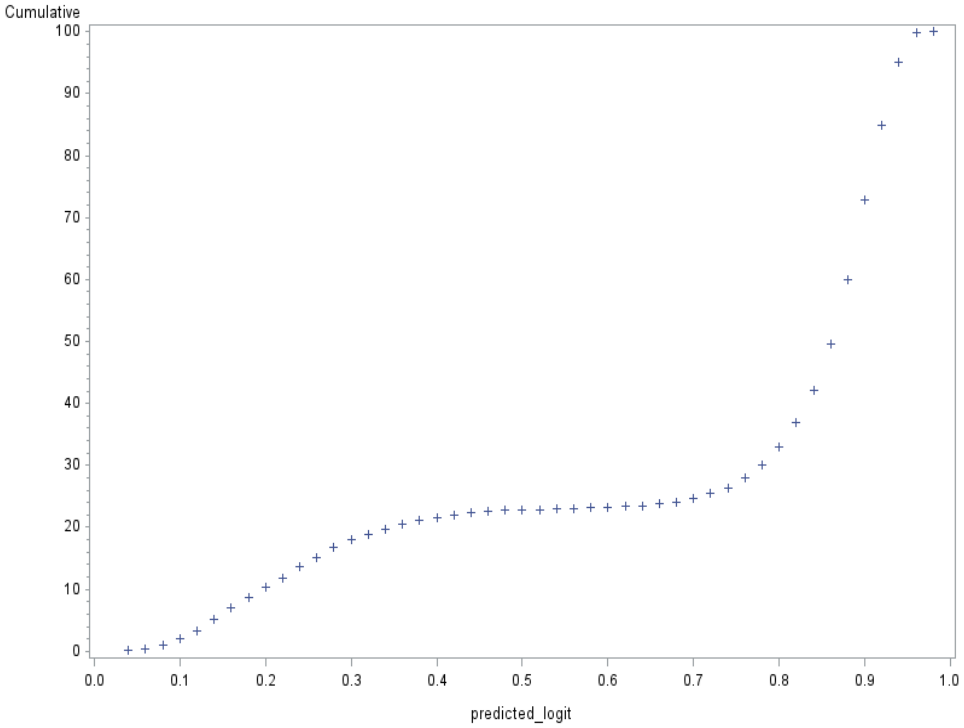
Linear regression

Loglinear regression

Cumulative frequencies of the predicted values for binary regression models

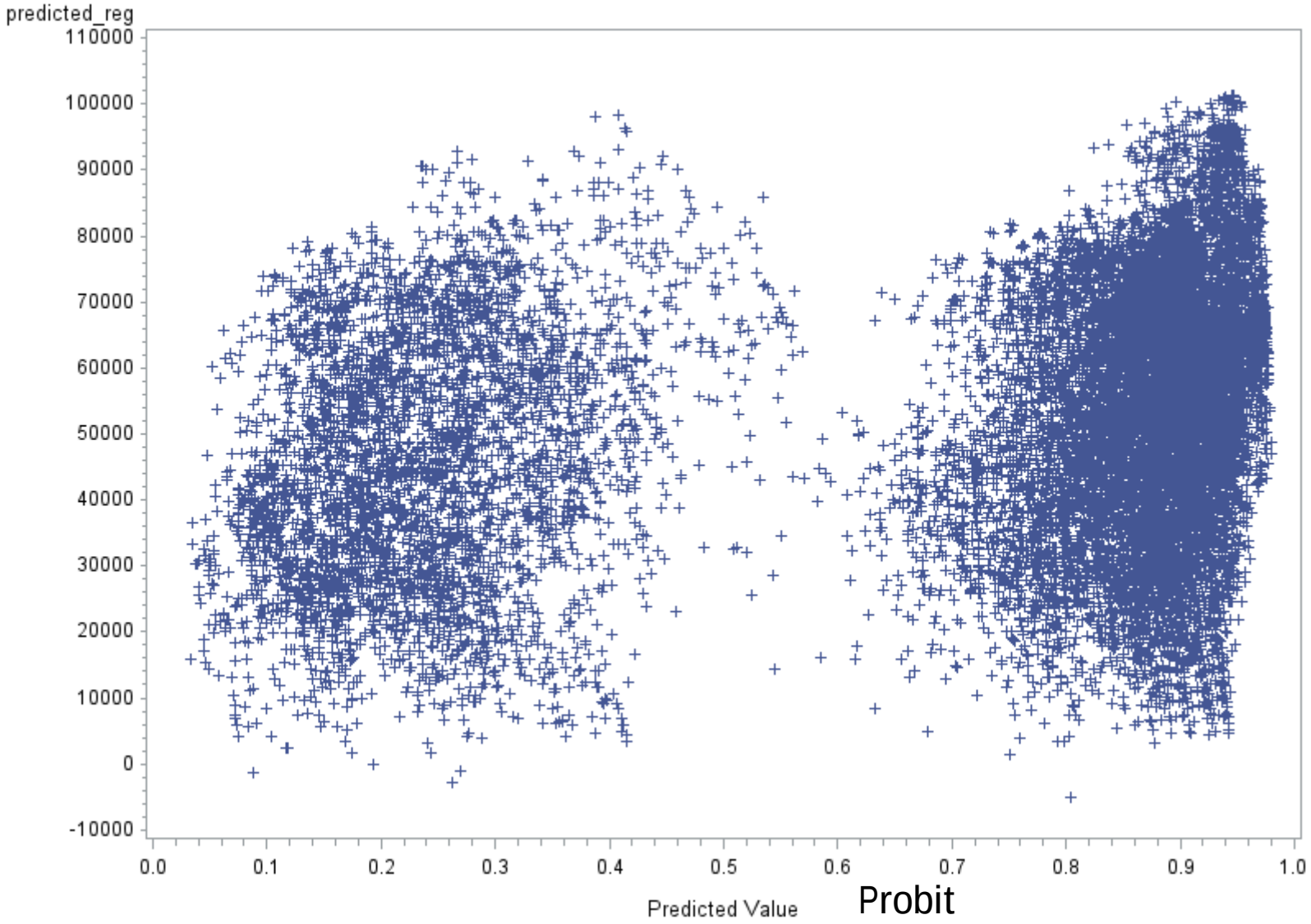


Probit regression

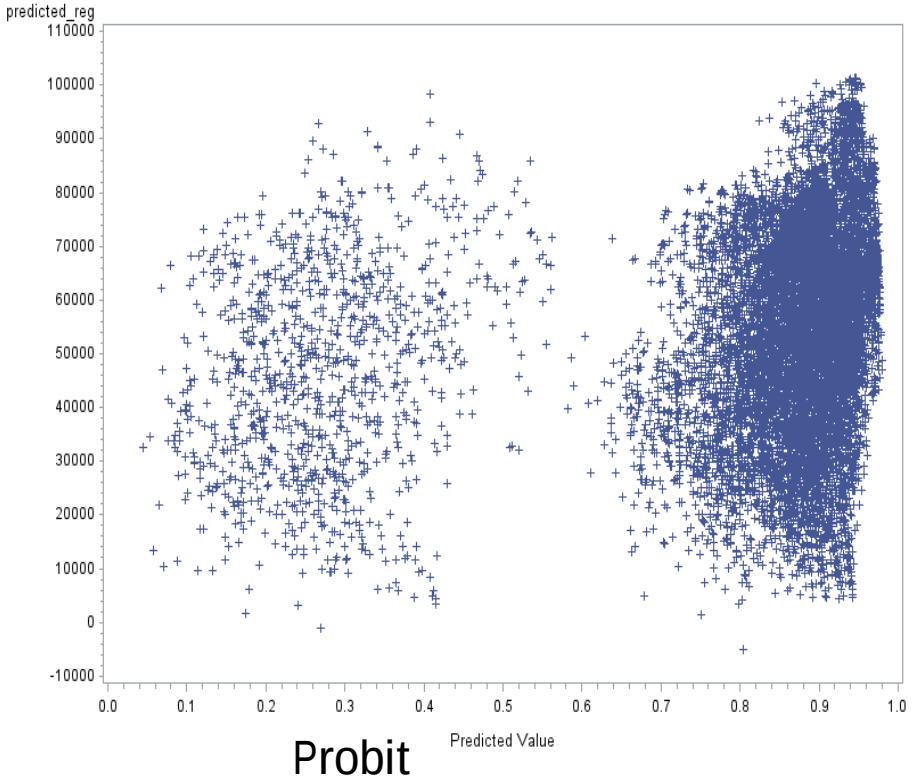


Logit regression

Scatter plots by predicted values: Both the respondent and the nonrespondents

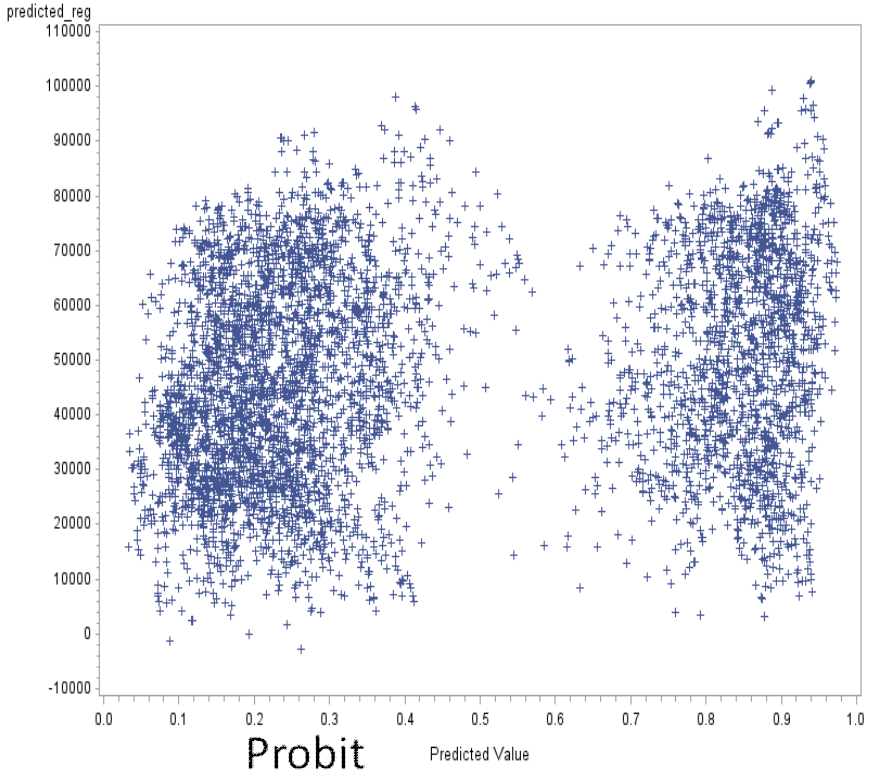


Scatter plots by predicted values:
For the respondent



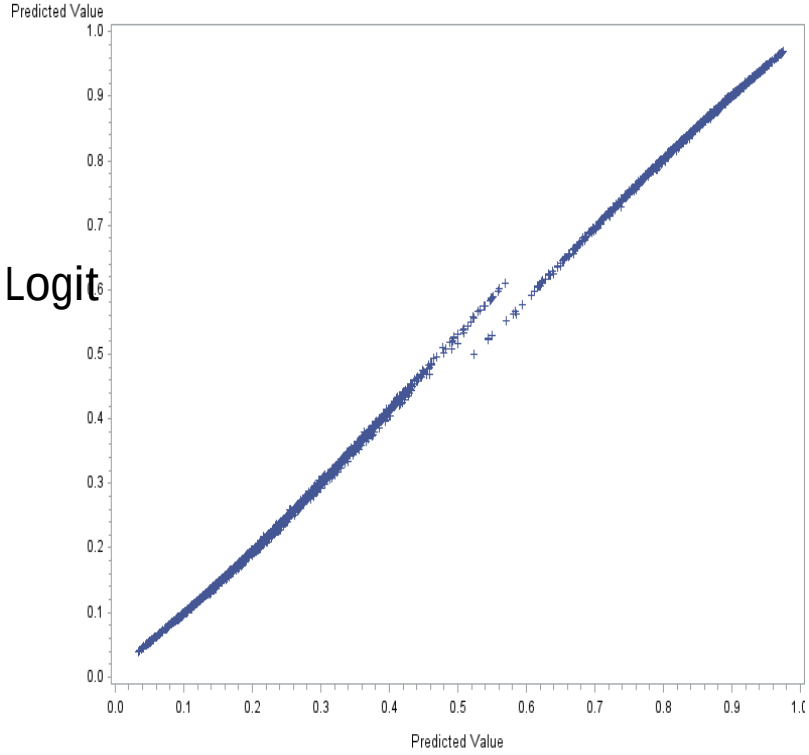
Probit

The nonrespondents

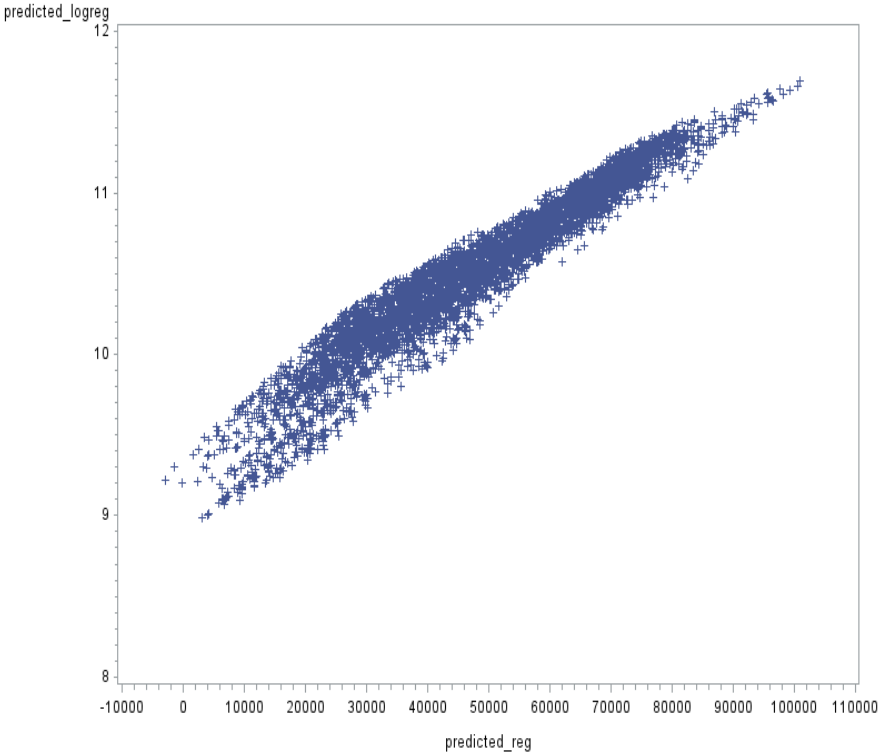


Probit

Scatter plots by predicted values: The nonrespondents
The similar models but with different ‘scales’
Predicting the response probability Predicting the income

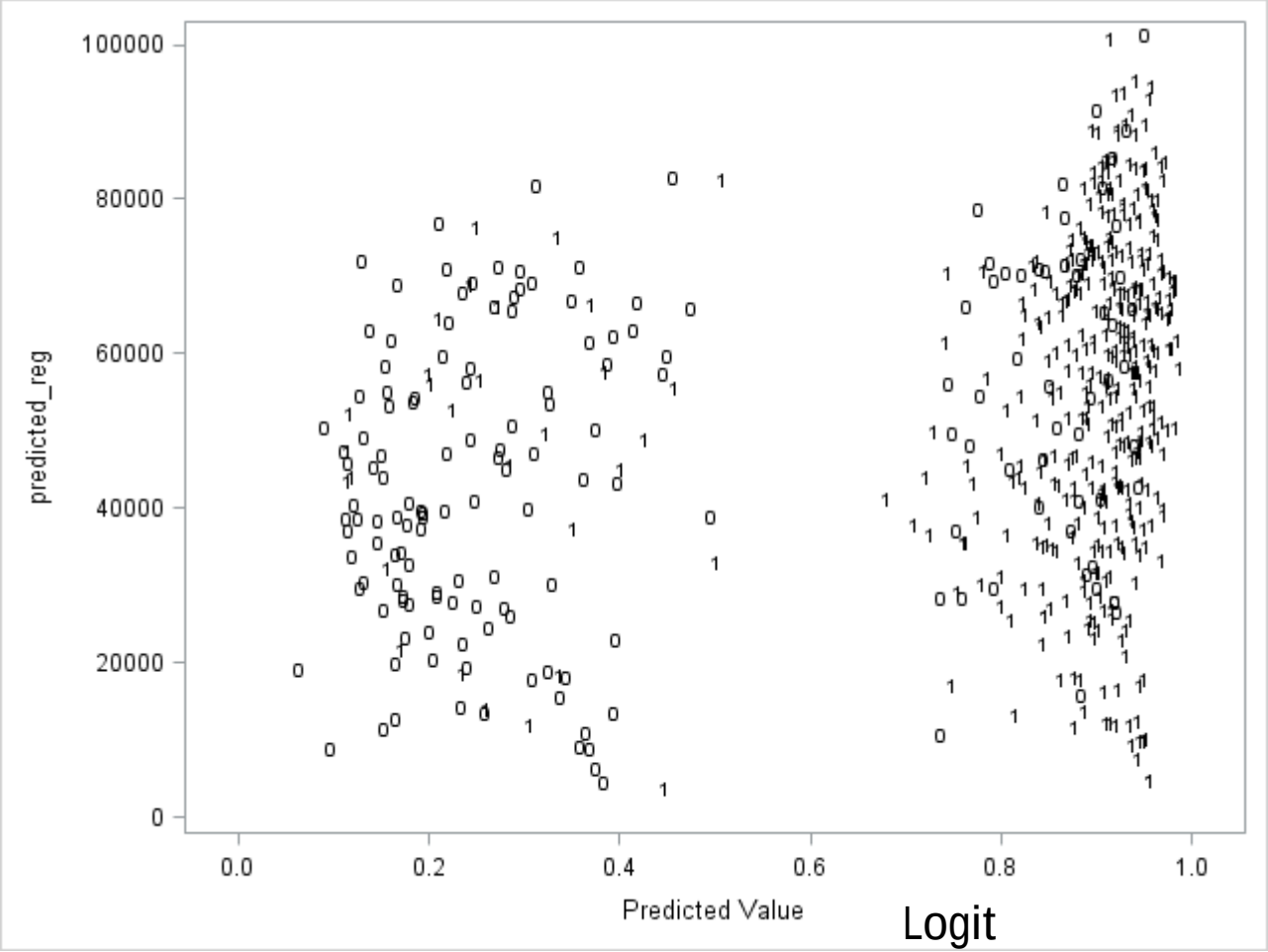


Logit

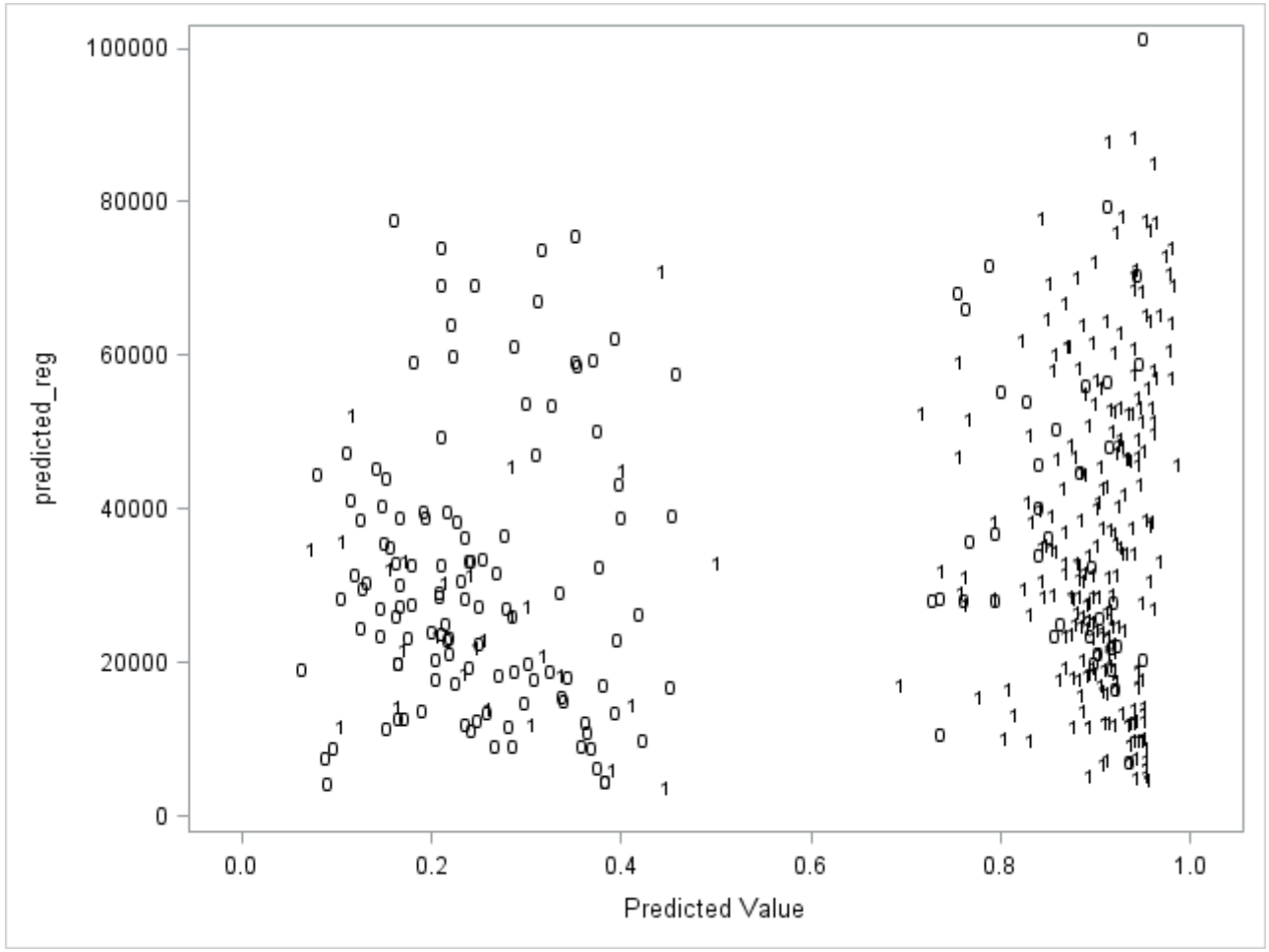


Probit

Scatter plot by two predicted values: A 3% random sample of the complete data so that the response indicator is marked: 1=observed, 0=not observed

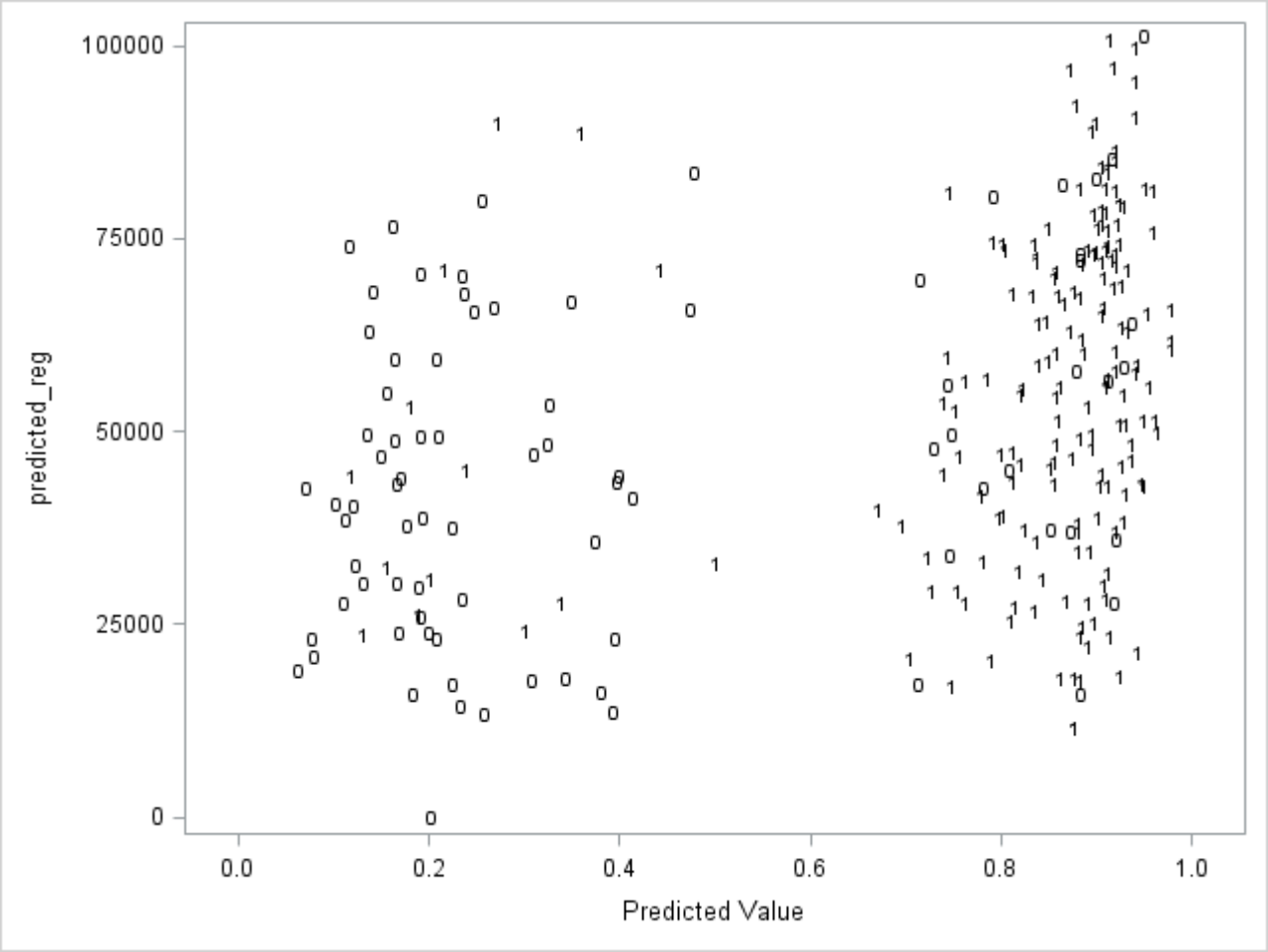


Scatter plot by two predicted values as the previous but for poor people, 10% random sample



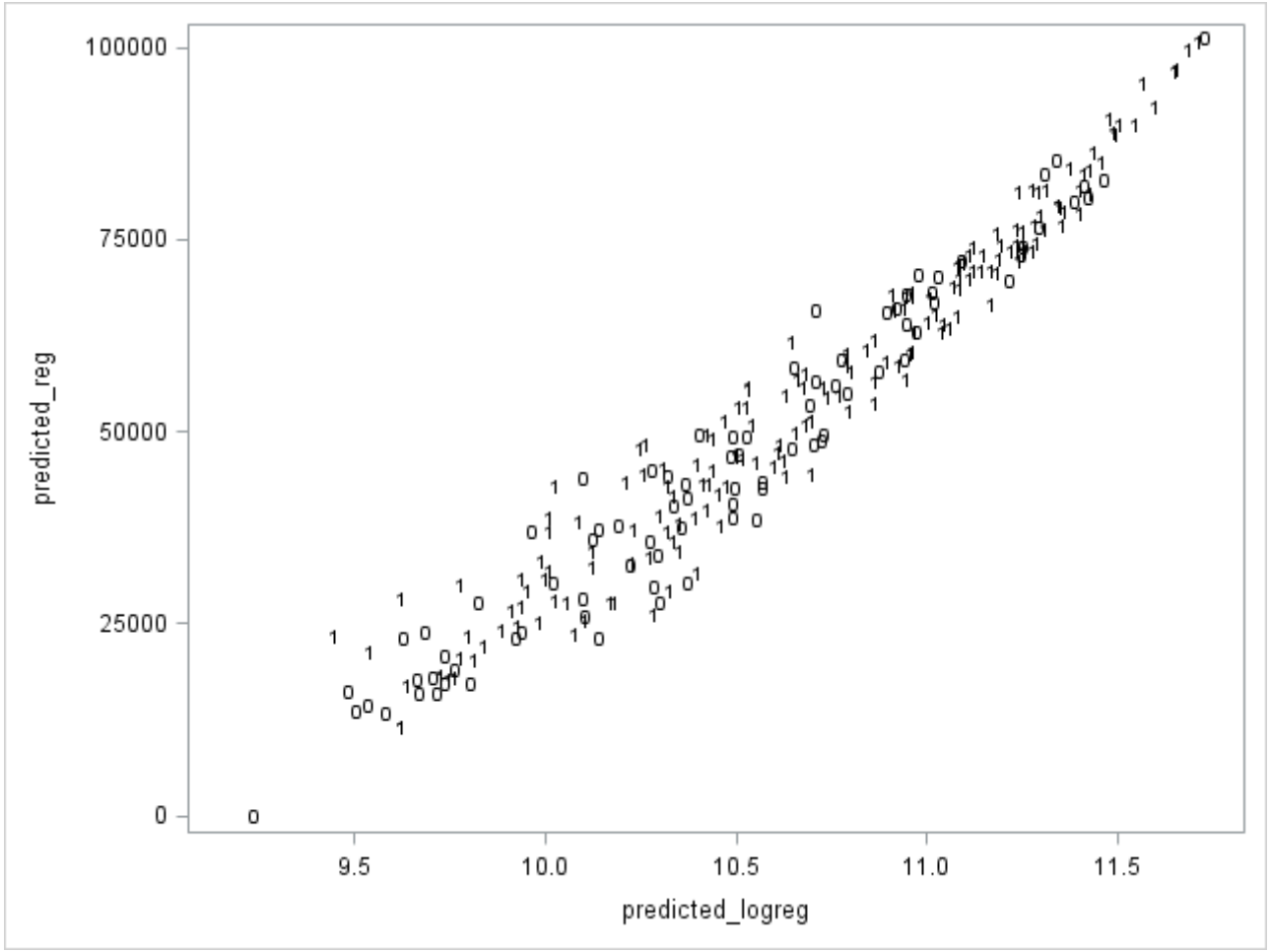
Logit

Scatter plot by two predicted values as the previous but for whose happiness is below 7, 10% random sample



Logit

Scatter plot by two regression-based predicted values as the previous but for whose happiness is below 7, 10% random sample



Concluding points about imputation models

The predicted values will have a big role when going to impute, that is, in the stage of the imputation task. The big point is that the predicted values should be available both for the respondents and for the non-respondents, i.e. the auxiliary variables should be complete as in our trainings. All the previous predictions are attempted. We have observed that there are many similarities but also essential differences and we cannot say surely which method is finally going to be the best if this will be found any way. However, it is expected that some methods are not good although used in real life. Our training data set is not easy that is good keep in mind for understanding difficulties of imputations. If the imputation model would be strong, that is, it is predicting well, most imputation task choices work quite well. Thus it does not matter which imputation task uses. But a usual real life application is not as easy and the imputation model thus does not fit very well. Nevertheless, imputations are good to perform. I hope that the examples of this course give some understanding about appropriate imputation methods, including both the model and the task that are connected to each other.

Imputation task

The two alternatives in general can be exploited after you have estimated the imputation model:

(a) Model-donor approach (malliluovuttaja) in which case the imputed values are computed deterministically (or stochastically) from the predicted values (adding noise) of the model.
(b) Real-donor approach (vastaajaluovuttaja) in which case the predicted values (or with adding noise) are used to find the nearest or a near neighbor of a unit with a missing value from whom an imputed value has been borrowed.

You see that the imputed values of case (b) are always observed values, observed at least once for respondents. The imputed values of case (a) are not necessarily observed except often for categorical variables (or they can be converted to possible values after preliminary imputation).

Imputation task 2

To integrate model and task you see that we have the following options. So, the predicted values of the missingness indicator cannot be used for model-donor imputation directly.

	(a) Model-donor approach	(b) Real-donor approach
(i) either the variable being imputed itself	Yes	Yes
(ii) the missingness indicator of this variable	No	Yes

Imputation task 3

Comment:
I use the term donor as it is used by many others but it is not general to use the term like model-donor. This methodology is often quite different, even spoken about model imputation when meant a type of model-donor imputation like when the imputation model is regression model and the imputation task is the direct predicted value. This is for me confusing since regression model can be used also for real-donor imputation. Model imputation is also strange since imputation always needs a model; so all imputations are model imputations.

The same confusion has been met often when speaking about logit imputation or probit imputation since this model can be used in both types of imputation tasks.

My term donor in task (a) means that the borrowing is derived from a group (group donors) that is a factual situation when modeling. The donor in task (b) is a unit, an individual.

Imputation task 4

Comment:
You will find from imputation literature the term ‘hot deck’ or ‘hot decking.’ This mystic term is derived from 1950’s I think when certain US surveyors randomly selected a donor from the observed values. This looked like ‘a hot deck’ in which those donors were moving their place and suddenly one was selected to replace a missing value. I do not like this term. It is historical and it is good to know origin. Later, I think, the term has been used also even though the donor selection is not random. E.g. when these real-donors are sorted in a certain order as we will do too. The title of my 2000 paper was e.g. ‘Regression-based nearest neighbor hot decking,’ but now this method could be ‘Nearest neighbor real-donor imputation when the imputation model is linear regression.’

We thus see that there is needed a certain near or nearest neighbor metrics for selecting a best donor whose observed value to be borrowed for imputing. We proceed to more details soon of this metrics.

Imputation task 5

Both imputation tasks use stochasticity or they can be applied deterministically. If stochasticity has been used in the imputation model, it follows that the imputation task should be automatically stochastic but it is still required to use certain random numbers in the imputation task. Stochasticity can be added also in the imputation task using appropriate random numbers. It is needed to assume how random numbers behave or what is their notional distribution (normal, lognormal, uniform)? If the real life data do not behave so, your imputation may violate your estimates.

Imputation task 6

The imputed value of the model-donor method is simply:
either
(•) Predicted value of the imputation model (*deterministic imputation)*
or
(••) Predicted value plus a noise term of the imputation model (*stochastic imputation)*.

I do not here (but later) go to details of the noise term but when using regression model it is often assumed its distribution to be normal with the mean = zero and the standard deviation = root mean square error. A problem is that there can be outliers in random values and consequently in imputed values. It requires to truncate outliers in some way. Another option, less problematic, is to use a pattern of observed residuals estimated for the respondents and then randomly draw these residuals to the noise for non-respondents. This strategy thus is a kind of a real-donor method.

Post-Editing after the model-donor method possibly

As known, the real-donor methods give observed values that are (or should) valid values. Hence nothing needed to do before the use of re-data.

But the model-donor imputed values thus are calculated and it is guaranteed that they are valid in all meanings. Sometimes they can still be used as such, but not always. Some examples:

- Our second variable in training is happiness that obtains the integer values from 0 to 10. When using model-donor methods, the imputes will be in most cases in decimal values. Any user does not accept it. A simple solution and sometimes used is to round them to integers.
- In SAS codes this can be done as follows:

```
data new2; set new;
if happy2 ne . then happy_imp_reg=happy2;
else happy_imp_reg=round(predicted_reg, 1);
mae=abs(happy- happy_imp_reg);
proc means data=new2 n mean cv min p1 p5 p25 p75 p95 max; where income_res=0;
var happy happy_imp_reg mae; run;
```

Post-Editing after the model-donor method possibly 2

The variable HAPPY thus is categorical but in the cases of a real continuous variable, the post-editing can also be important but its influence in the final results is not necessarily big. However, most clients do not like e.g. incomes with several decimals as we have obtained. Such values also indicate clearly for an expert that these are imputed. Thus: if the confidentiality is important as is often, a rounding is a good solution but what is the best rounding? My answer: the same as in the observed values. I looked at our data and found that the income values are in five euro's. Hence the rounding due to confidentiality and esthetic reasons can be as follows:

```
data new2; set new;
if income2 ne . then income_imp_reg=income2;
else income_imp_reg=round(predicted_reg, 5);
mae=abs(income- income_imp_reg);

proc means data=new2 n mean cv min p1 p5 p25 p75 p95 max; where income_res=0;
var income income_imp_reg mae; run;
```

Nearness metrics of real-donor methods

The imputed value of the real-donor method requires a metrics used to find an optimal unit donor from whom to borrow the imputed value.

This metrics can be derived from outside the data. The Mahalonobis distance is one such metrics used. Most typically, it is assumed that certain units (overall or within each imputation cell) are as close to each other. This means that a donor has been selected randomly (within the entire data or within an imputation cell). It is thus stochastic. This method is just the initial random hot deck method from 1950's.

Another common strategy is to use a smartly chosen other metrics and search for the nearest or a near donor from the data set. This because it is assumed that the units close to each other are similar. Of course, the success depends on those variables in this metrics.

Nearness metrics of real-donor methods 2

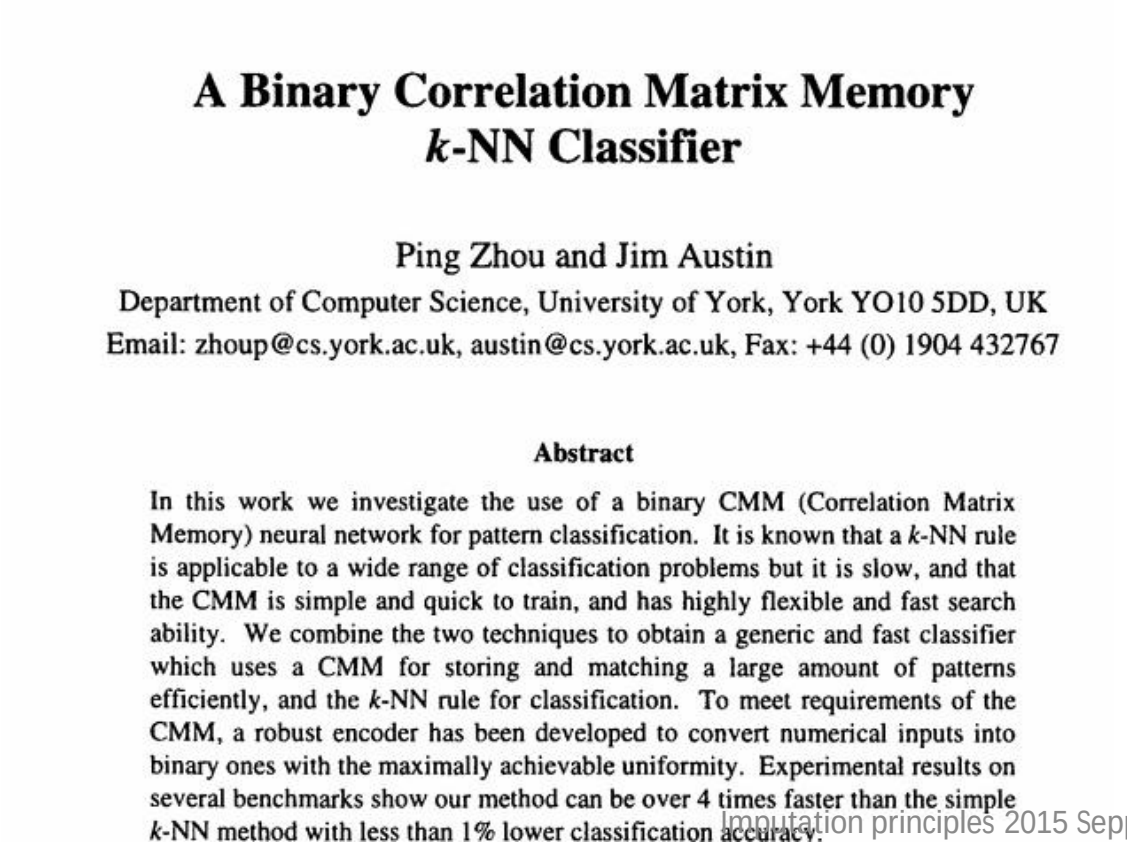
The third and most common metrics as guessed from the previous graphs is the metrics derived from the predicted values of the binary regression model (thus the link function should be chosen by the user). In the case of a stochastic selection, some random noise is needed to add but there are different options for this. We do not go to their details, but I want to mention a common tool from the Imputation book by Rubin:

- Classify the predicted values into a certain number of categories by their values, e.g. 10 to 20 categories, called imputation cells. These are fairly homogeneous and thus enough close to each other.
- Select randomly within each cell one observed value to replace a missing value. This method is called sometimes cell-based random hot deck.

The observations of this kind of imputation cells are called also 'donor pools.' There thus is a pool where to go to borrow a good value to replace a missing value. It is maybe good to create such donor pools in advance for imputing but the values of this pool should be from the same period at minimum.

Nearness metrics of real-donor methods 3

The cell-based random hot deck or ‘real-donor method using response propensity cells’ does not give any nearest neighbor but a near neighbor. There are literature e.g. such term as ‘*k*-Nearest Neighbors algorithm’ that is close to this idea so that this gives *k* nearest for each unit selected. The same idea was used in the Euredit project by the York University team.



A Binary Correlation Matrix Memory *k*-NN Classifier

Ping Zhou and Jim Austin
Department of Computer Science, University of York, York YO10 5DD, UK
Email: zhoup@cs.york.ac.uk, austin@cs.york.ac.uk, Fax: +44 (0) 1904 432767

Abstract

In this work we investigate the use of a binary CMM (Correlation Matrix Memory) neural network for pattern classification. It is known that a *k*-NN rule is applicable to a wide range of classification problems but it is slow, and that the CMM is simple and quick to train, and has highly flexible and fast search ability. We combine the two techniques to obtain a generic and fast classifier which uses a CMM for storing and matching a large amount of patterns efficiently, and the *k*-NN rule for classification. To meet requirements of the CMM, a robust encoder has been developed to convert numerical inputs into binary ones with the maximally achievable uniformity. Experimental results on several benchmarks show our method can be over 4 times faster than the simple *k*-NN method with less than 1% lower classification accuracy.

See the paper abstract in which the CMM is mentioned.
This method was not bad in the Euredit examinations.

Nearness metrics of real-donor methods 4

The York University method works but was not best. The better solution is to try to find a nearer neighbor than one randomly from a possibly large group. I have used (and we will use it in our training) such a method that

- (i) sorts all the units in the data by the predicted values from the largest to the smallest
- (ii) creates the lagged variables as many as needed (maybe even 10-20 lag variables), nearest= lag1, 2nd nearest= lag3, 3rd nearest=lag5, ...
- (iii) sorts this sorted data set to the opposite order, that is, from the smallest to the largest
- (iv) creates the lag variables similarly to (ii), 1st=lag2, 2nd=lag4, 3rd=lag6, ...
- (v) begins the imputation so that if a missing value is observed then it is first looked whether lag1 is non-missing; if 'yes', this value has been chosen as an imputed value; if 'no', then it is checked lag2 and so on as long as all the values are imputed.

This method works well except if gone too far from a nearest value to find an observed value. It means that the same real-donor will be used more than once as a real-donor. This approaches to a weighting method.

Nearness metrics of real-donor methods 5

The fourth good and rational strategy (like my regression-based nearest neighbor hot decking) in many situations is to use model-donor imputation values (that are predicted values of a regression model e.g.) over both the respondents and the non-respondents as the nearness metrics. This thus means that we impute technically the values for the respondents too, using the same strategy as for the non-respondents. It is not difficult and we made it already in graphs below. The next step is to work as in the previous case either to select the nearest donor, or a near donor that is usual when desired to randomize the procedure. Thus e.g. our nearness metrics is the previous model-donor output:

(•) Predicted value of the imputation model (*deterministic imputation of the entire data set)*
or
(••) Predicted value plus a noise term of the imputation model (*stochastic imputation)*.

Nearness metrics of real-donor methods 6

To make the previous point “Thus e.g. our nearness metrics can be the previous model-donor output” more clear:

We can thus work so that we first perform imputations using model-donor methodology but in this case also for the respondents (observed units) in addition to the non-respondents (not observed). Now we have the nearness metrics that is used – to find the nearest neighbor (or a reasonably near neighbor) for each non-respondent from the respondents and

- to insert this value to this unit.

This also gives opportunity to compare both strategies easily when estimating some figures from the imputed data set.

It is also possible to choose a model-donor imputed value for those units whose nearest neighbor is too far and thus not be plausible. In this case the final imputation is a mixed real&model-donor method. It is allowed.

Nearness metrics of real-donor methods 7

The imputed value of the real-donor method.

If the imputation model is based on the missingness/response indicator, the imputation is similar to that presented in previous pages, but now the values of the nearness metrics are thus within the interval (0,1). The SAS codes are thus similar in both cases but the values are not. Now we have automatically these propensity values both for the respondents and for the non-respondents. There are still several options to work with these values. These will be considered later in details. An interesting special case is such in which the variable being imputed is binary as well. Thus both variables (in imputation model and in analysis) are binary. This may arise confusion.

Single and multiple imputation

Imputation can be performed for each desired value of the non-complete variable just once, or several times. The first is called *single imputation (SI)* and the second *multiple imputation (MI).* These are not the two different imputation methods as often said, since multiple imputation means that single imputation has been repeated several times. So, each single imputation should aim at succeeding as well as possible e.g. avoiding the bias. There are the strict rules how to repeat imputation properly. The rules are not always clear and hence often criticized.

MI is in certain problems difficult to realize so that the users are happy. E.g. imputing values of large businesses this methodology may cause confusions. Instead, if imputation is concerned a big number of missing etc values for e.g. households and small/medium sized businesses (thus sample with large sampling weights) MI may be beneficial. Many details of MI are considered in the specific section of this course. MI is usually based on a Bayesian approach that is developed by Don Rubin (US), but non-Bayesian (called also repeated MI) is also used that I will prefer so far. Jan Björnstad (Norway) introduced this concept in 2007 (J. of Official Statistics).

Summary: Imputation model plus Imputation task in the case of the linear regression model

	Deterministic Single	Stochastic Single Multiple
Model-Donor	A. Regression model estimated and its predicted values are used as imputed values for missing items	C. Adding to the A model the normally distributed random numbers with the zero mean and with the Root_Mean-Square_ Error standard deviation. Or to add observed residuals.
Real-Donor	B. Regression model as in A but those predicted values are computed both for the respondents and for the non-respondents but now these are used as a nearness metrics.	D. Like B but applying to the C model. Multiple imputation by using several seeds for random numbers. This is concerned C too.

Summary: Imputation model plus Imputation task
In the case of the response indicator model

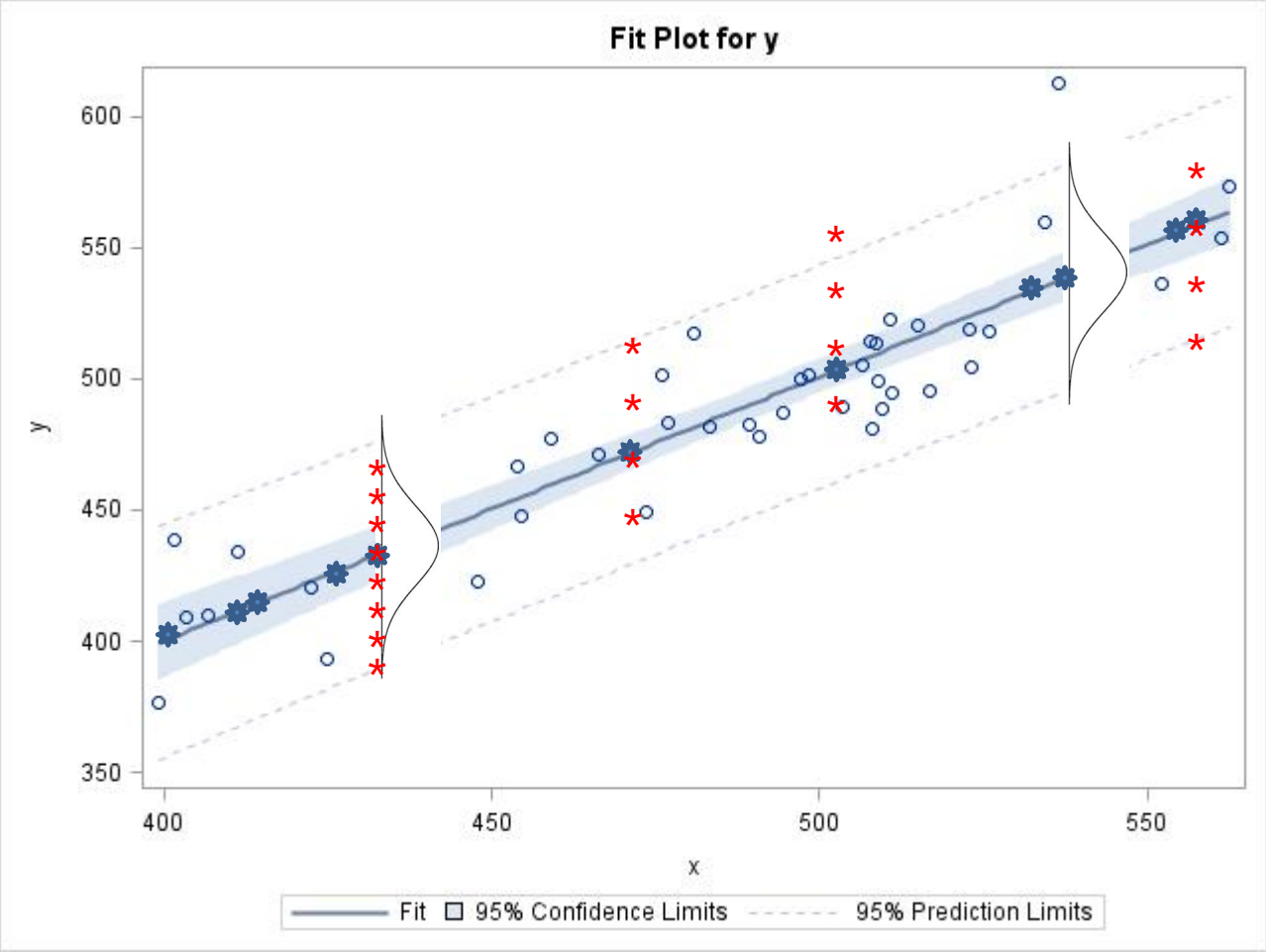
Model-Donor	Deterministic Single	Stochastic Single Multiple
Real-Donor	Nothing	Nothing
	E. Logit, probit or Comple-mentary log-log (CLL) reg-ression model with the respective explanatory variables as above. Those predicted values (response propensities) are computed both for the respondents and for the non-respondents that used as a nearness metrics.	F. Adding 'noise' in which different strategies can be used, always uniformly distributed random numbers, but I do not go now to details

Example, why and how to get adding to the A model the normally distributed random numbers with the zero mean and with the Root-Mean-Square-Error (RMSE) standard deviation.

This thus is derived from the model uncertainty (non fitting) that is simply measured by the residual and its standard deviation. As said above that if assumed that it normally distributed, it is possible that some 'residuals' are too big (i.e. above any observed residual): it that case it is good to think whether to truncate them.

The SAS codes in this case you will find after the next page illustration.

Illustration of the model-donor imputation with a simple regression. The random noise term N(0, RMSE) is added to the predicted values. It is a danger that the imputes are outside the plausible limits.



- A predicted value = Deterministic impute

* A possible impute with noise

y = imputed if missing
x = auxiliary variable

SAS codes for adding the noise with N(0, rmse) Continues for a next page

```
proc glm data=a.impucomplete; class z1 z2 z3 z4 ;
model income2=z1 z2 z3 z3*z4 x1 x1*x1 /solution ; output out=reg
p=predicted; r=residuals_reg; run;

/* It is needed to include those residuals and their minimum and the
maximum in the merged file. This can be made in varous ways but
this is my way: I create a new variable i and give the simplest
constant value. This same variable is needed in the initial output file
= reg in order to merge them together. This requires the sorting by
this variable. It looks maybe strange but it works. Next we thus
merge these files and we have constant values root mean square
error, and its minimum and maximum that are used to robust the
random number based residuals with the normal distribution that we
will get by the operator rr.*/
```

```
proc summary data=reg nway; var predicted_reg residuals_reg; output out=rmse
std(residuals_reg)=rmse min(residuals_reg)=min max(residuals_reg)=max
mean(predicted_reg)=mean;
data rmse; set rmse;
i=1; proc sort; by i;
data reg; set reg;
i=1; proc sort; by i;

/* Next we calculate these imputed values
This is one strategy for robusting imputes, that is, avoiding extreme values.*/
data reg2; merge reg rmse; by i;
rr=rannor(1); if rr<min/mean then rr=min/mean; if rr>max/mean then rr=max/mean;
if income_res=1 then income_imp=income2;
else income_imp=predicted_reg+rr*rmse;
mae=abs(income_imp-income);

/* And we get out results including true values);*/
proc means data=reg2 n mean cv min p1 p5 p25 p75 p95 max; where income_res=0;
var predicted_reg income_imp income mae; run;
```

Single and multiple imputation 2
Technics

Let
L = number of imputations *u*,
Θ = parameter being estimated,
and its point-estimate = *Q* (e.g. mean income and CV)
and variance estimate, respectively, = *B*
And then standard error of the mean = square root of the variance.

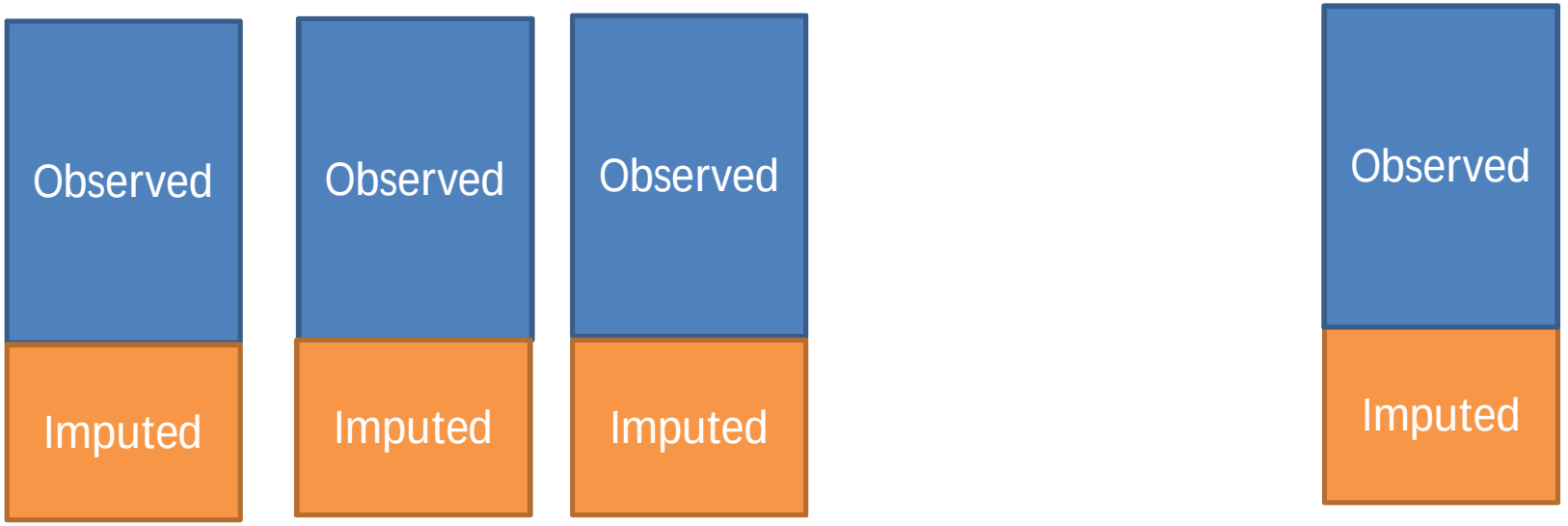
All these are calculated as usually so that the imputed values are included as such. The estimate may be whatever such as average, total, ratio, proportion, median, percentile, regression coefficient.

The number of imputations = *L* is in Rubin's initial book even as small as 3, but may work only with simple data sets. I think that *L*>=10 could be best to use in practice. Rubin's *L=3* is understood if remembers how inefficient the computers were in 1980's.

Single and multiple imputation 3

A simplified illustration of *L* single data sets with imputations (complete data)

Simple because the fractions of missing values may vary by variables. Here all have the same fraction.



Point and interval estimates from each data set

Single and multiple imputation 4

Now the multiply-imputed point-estimate is a simple average of multiply imputed estimates

$$Q_{MI} = \frac{\sum_u Q_u}{L}$$

Respectively, the variance can be calculated as the average of the variances of *L* complete data sets in which each variance is estimated using the formula that is valid for the sampling design of the survey. This is for the gross sample data set that also includes the units that are not needed to impute. But because a certain number is missing these are imputed and the average and the variance are calculated in a best way thus.

$$B = \frac{\sum_u B_u}{L}$$

Single and multiple imputation 5

The variance estimate is respectively $B_{MI} = \frac{\sum_u B_u}{L} + (k + \frac{1}{L}) \frac{1}{L-1} \sum_u (Q_u - Q_{MI})^2 =$

$k = \frac{1}{1-f}$ *f* = the fraction of missing and imputed values

If *k=1* or *f=0*, it is Rubin's formula, otherwise Björnstad's formula.

You see that the entire variance consists of the two components: (i) the average of variances (within-variance) and (ii) the between-variance that indicates how much multiply imputed estimates vary. If the variation is zero, this between-variance is zero too.

It is good to remind that multiple imputation is not any own imputation method but it consists of several single imputations. If single imputation is not working, multiple imputation is not either working. Some authors, unfortunately, are not speaking in this way. 'Multiple' requires thus a stochastic element.

Single and multiple imputation 6

The initial multiple imputation was developed by Donald Rubin. It was based on the Bayesian theory. This theory thus was reformulated by the Norwegian Jan Björnstad. A reason was that Rubin's strategy is not well working in many practical situations like in statistical offices. Hence he uses the term non-Bayesian.

It is not the only difference in these frameworks. The Bayesians use certain Bayesian rules in all imputation methods. Instead, the non-Bayesian framework uses simpler rules. A big question follows from this:
How good are these frameworks in practice?
And are the Bayesian rules really useful and better? Note that these rules are developed by Rubin and a user thus have to trust in him or his specifications. I have to say that I am not convinced about all the solutions?

Specialities for imputation of a categorical variable

This same framework is workable for categorical variables as well but the

	(a) Model-donor approach	(b) Real-donor approach
(i) either the variable being imputed itself	Yes	Yes
(ii) the missingness indicator of this variable	No	Yes

Alternatives of the first row are automatically different since the imputation model is can not be ideally any linear regression model. These cases are considered in following pages.

Fortunately, when using the binary missingness indicator as the dependent variable, the imputation task can be exactly similar as in the case of a continuous variable. That is, use the same nearness metrics in imputing missing values as above.

Model-donor imputation of a categorical variable

It is easiest to use this case so that each category has been imputed separately but this takes more time of course when comparing the real-donor imputation. This can be made using multinomial distribution and an available link function (logit, probit, cll). We do not concretize this methodology in this course, but apply in imputations a binary variable such as poor vs not poor or unemployed vs employed, sick vs not sick, happy vs not happy.

In this case, the imputation model is binary with an available link function as for the real-donor imputation above but this dependent variable is this binary variable being imputed, not any indicator.

The imputation model is estimated and the predicted values are respectively as usually. When going to imputation tasks on next pages, the variable of these predicted values is 'predicted_md' in which 'md' refers to 'model-donor.'
You remember that these values are within the interval (0, 1).

Model-donor imputation of a categorical variable 2

Alternatives:

(i) If the prediction is working well that is not guaranteed, it is easiest
- To give an imputed value = 1 if a predicted_md > 0.5
- an imputed value = 0, otherwise.

Thus if the binary model is concerned the variable 'poor' so that 1 = poor and 0 = not poor, this basically works but I have not seen any good empirical result on this. The following one instead works reasonably:

(ii) Create an uniformly distributed random variable within the same interval as the predicted values are i.e. (0,1). This is below the variable 'ran'.

```
data new3; set new2;
ran=ranuni(10);
if predicted_md>ran then poor_impu_md1=1; else poor_impu_md1=0;
if predicted_md>0.5 then poor_impu_md2=1; else poor_impu_md2=0;

proc means n mean cv min p1 p10 p75 p90 p95 p99 max ; where income_res=0;
var poor_impu_md1 poor_impu_md2 predicted_md; run;
```

As you see, the last imputed variable is for the alternative (i). The predicted values are computed for understanding.

Model-donor imputation of a categorical variable 3

As you see, the first alternative is deterministic but the second is stochastic. Hence the second can be used for non-Bayesian multiple imputation just changing the seed number that is now = 10. If changing this number the results vary to some extent but not in most cases much. Note that the second approach follows the Bernoulli distribution.

There are other deterministic solutions like learning about the observed results but I am not convinced about any. However, when calculating the average of the predicted_md this ‘aggregate imputation’ for the mean is fairly good but you cannot know any correct individual values. Thus if this aggregate only is needed, you can use this aggregate.

Preserving associations in the case of missing data

Associations like correlations are in some cases good to preserve or not violate dramatically when handling missing data. Here are some strategies:
(i) Do not impute at all, thus use data deletion. You will lose observations and your standard errors are larger. Also your results are biased to some extent. But it does not matter if you do not like to publish this paper.
(ii) Try to use such analysis method that takes missingness into account (the Nobel winner economist Heckman has developed a much cited strategy).
(iii) Adjust for missingness by a good reweighting method, also using auxiliary variables as much and well as possible.
(iv) Apply a real-donor methodology so that the whole (or essential) pattern of the variable values has been chosen from the same donor. You can put a bit random variation there, of course. This kind of pattern may also be relative such as relative distribution, not absolute values.
(v) Apply sequential imputation so that impute first variable y1, next impute y2 so that the imputed variable y1 is one additional auxiliary variable, and so on y3, …. all variables that are interest for you in this respect. Note that if the first imputation is not good, the next one may be worse, etc. but try nevertheless.

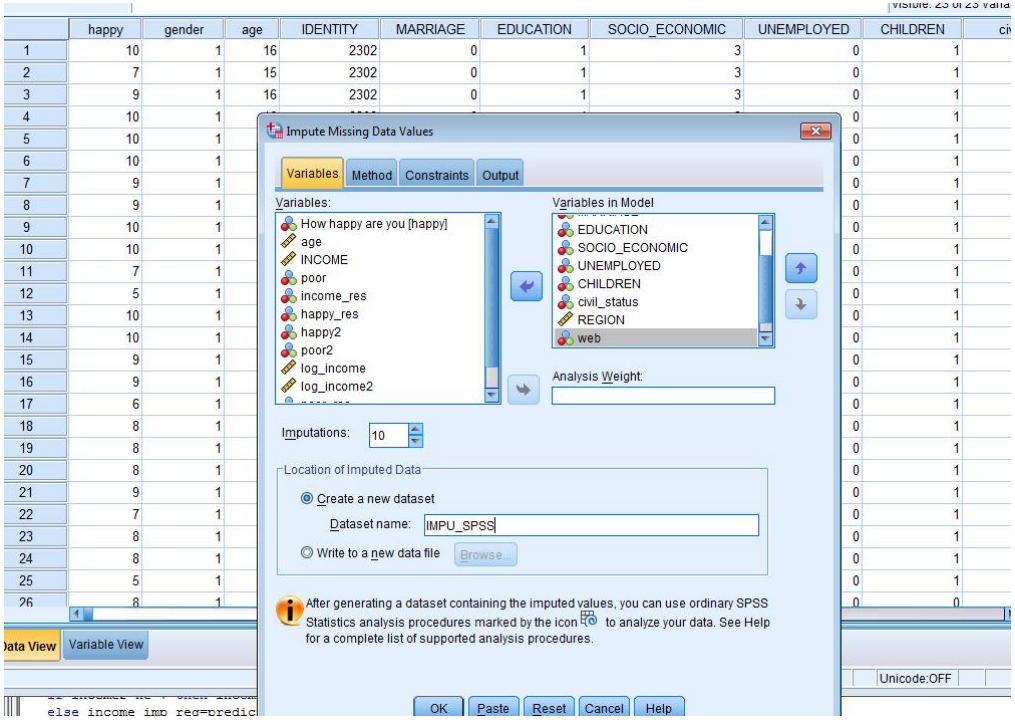
End comments

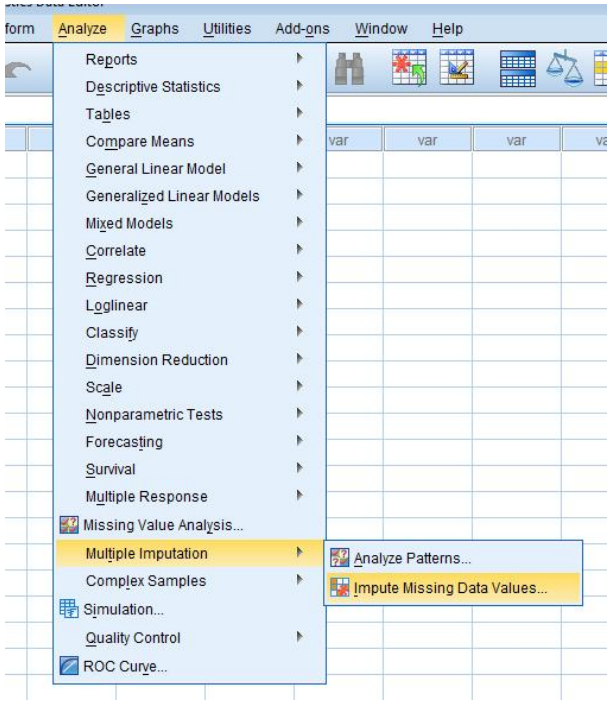
This ‘story’ covers my approach to imputation. Many things have also been trained and concretized respectively. I hope that you will keep in mind these principles when we go on to details.

An alternative could be to use ‘a black box software’ that gives your imputed values rather automatically. I would not be happy with such ‘boxes’ when working with real data since a client or reviewer is demanding and not without convincing statements believe all completed data. We will test SPSS black box software, and use the same data. The starting points for the SPSS training are in the following pages.

End comments and SPSS

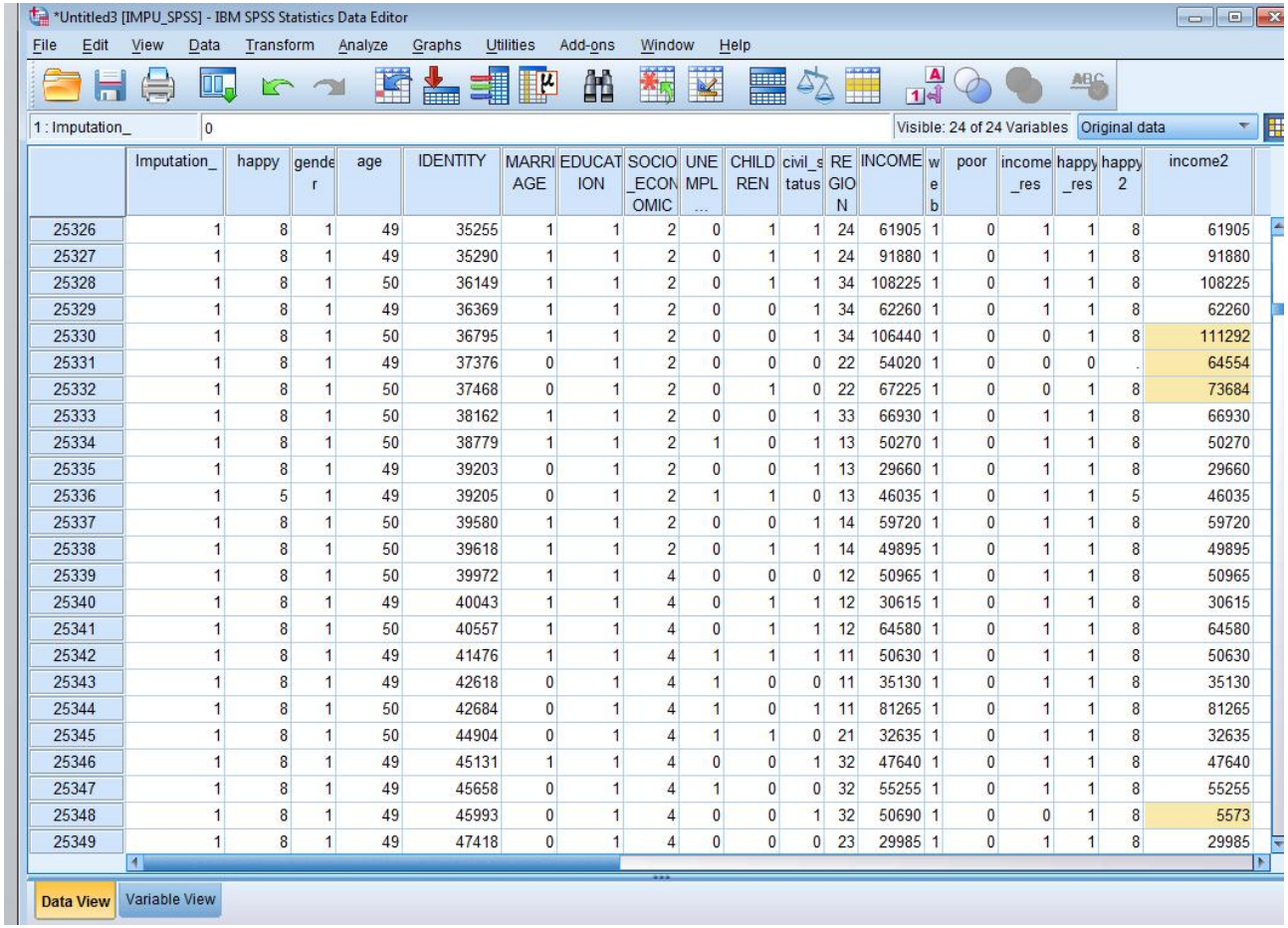
First push here right.
Next you have to select your SAS data that SPSS understands and select the variables (no income, poor or happy), and the number of imputations and the output file name. And then ‘ok’.





As you see, next you can choose a method and also constraints for the variable being imputed

The next page shows a part of the output where you see which values are imputed but the variable name is the same.



	Imputation_	happy	gender	age	IDENTITY	MARRIAGE	EDUCATION	SOCIO_ECONOMIC	UNEMPL...	CHILDREN	civil_status	REGION	INCOME	web	poor	income_res	happy_res	happy2	income2
25326	1	8	1	49	35255	1	1	2	0	1	1	24	61905	1	0	1	1	8	61905
25327	1	8	1	49	35290	1	1	2	0	1	1	24	91880	1	0	1	1	8	91880
25328	1	8	1	50	36149	1	1	2	0	1	1	34	108225	1	0	1	1	8	108225
25329	1	8	1	49	36369	1	1	2	0	0	1	34	62260	1	0	1	1	8	62260
25330	1	8	1	50	36795	1	1	2	0	0	1	34	106440	1	0	0	1	8	111292
25331	1	8	1	49	37376	0	1	2	0	0	0	22	54020	1	0	0	0	.	64554
25332	1	8	1	50	37468	0	1	2	0	1	0	22	67225	1	0	0	1	8	73684
25333	1	8	1	50	38162	1	1	2	0	0	1	33	66930	1	0	1	1	8	66930
25334	1	8	1	50	38779	1	1	2	1	0	1	13	50270	1	0	1	1	8	50270
25335	1	8	1	49	39203	0	1	2	0	0	1	13	29660	1	0	1	1	8	29660
25336	1	5	1	49	39205	0	1	2	1	1	0	13	46035	1	0	1	1	5	46035
25337	1	8	1	50	39580	1	1	2	0	0	1	14	59720	1	0	1	1	8	59720
25338	1	8	1	50	39618	1	1	2	0	1	1	14	49895	1	0	1	1	8	49895
25339	1	8	1	50	39972	1	1	4	0	0	0	12	50965	1	0	1	1	8	50965
25340	1	8	1	49	40043	1	1	4	0	1	1	12	30615	1	0	1	1	8	30615
25341	1	8	1	50	40557	1	1	4	0	1	1	12	64580	1	0	1	1	8	64580
25342	1	8	1	49	41476	1	1	4	1	1	1	11	50630	1	0	1	1	8	50630
25343	1	8	1	49	42618	0	1	4	1	0	0	11	35130	1	0	1	1	8	35130
25344	1	8	1	50	42684	0	1	4	1	0	1	11	81265	1	0	1	1	8	81265
25345	1	8	1	50	44904	0	1	4	1	1	0	21	32635	1	0	1	1	8	32635
25346	1	8	1	49	45131	1	1	4	0	0	1	32	47640	1	0	1	1	8	47640
25347	1	8	1	49	45658	0	1	4	1	0	0	32	55255	1	0	1	1	8	55255
25348	1	8	1	49	45993	0	1	4	0	0	1	32	50690	1	0	0	1	8	5573
25349	1	8	1	49	47418	0	1	4	0	0	0	23	29985	1	0	1	1	8	29985

This is from the first imputed data set. If the variable ‘Imputation_’=0, it is the initial non-imputed data set.
Now it is possible to calculate the results. This can be made using the SPSS but I use SAS so that this is first saved as a SAS file and e.g. the operations of the next page are used. This gives opportunity to compare these results with our earlier ones.

SPSS Multiple Imputation

The SAS codes for getting 10 multiply imputed results in the same way as earlier. I thus saved the former file with the name SPSS in my library ‘a’.

```
data spss; set a.spss;
if Imputation_ ne 0;
if income_res=0;
mae=abs(income-income2);
run;
proc means data=spss n mean cv min p1 p5
p25 p75 p95 max ;
class Imputation_; var income income2 mae;
run;
```

It is possible to calculate e.g. the average of all 10 imputations and get one estimate. This is not included in this material but if you can wish to do it, you can follow the formulas above (Rubin and Björnstad). In that stage it is good to use PROC SUMMARY. SPSS options can also be used but I do not give any model here.

SPSS Multiple Imputation 2

In the training, it is expected that you will do the three (or more) imputations so that

- The first one is simplest, i.e., the automatic that thus only requires to put the correct variables in the box (not true value variables) and the name of the output file
- The second one could be a custom method ('Monotone,' maybe not MCMC unless you do not know what it is) with Linear regression; as you see there is an interaction option and hence if you had used an interaction earlier do it similarly here
- The third expected method to be tested is predictive mean matching that is a Bayesian real-donor method and thus comparable with our real-donor methods.

SPSS has also an option 'Constraints' that gives opportunity to put there a minimum (maybe maximum) value. This could be best to test for your imputed variable = income2. You can put there e.g. a minimum>=0 and compare the results without this constraint. THUS: add one method with this constraint and compare.

Some notes about SAS MI

We had no time for real training with SAS multiple imputation for which purpose PROC MI is available. This follows Bayesian rules as SPSS. It includes about the same methods as SPSS but there are some more methods as well. You can look at details and codes with examples about SAS help and documentation (or even by googling with good words). SAS MI is not easy to well use and it is a bit old-fashioned. I below give one set of some SAS codes for getting some understanding. This is for linear regression and PMM = predictive mean matching (you can choose one of those options):

```
proc mi data=income round=.1  mu0= 0 35 45 nimpute=10
      seed=13951639 out=outex3  ;   class gender marriage education region
SOCIO_ECONOMIC unemployed children agegroup web;
                              monotone reg
                              regpmm (income2=gender marriage education region
SOCIO_ECONOMIC unemployed children agegroup web);
   var gender marriage education region SOCIO_ECONOMIC unemployed children
agegroup web income2;  run;
```

Solution to the first page imputation task.

Did you impute correctly or how close? Not difficult due to good auxiliary data.

What is this animal? In the aboriginal's language it means 'I do not understand' since the first visitors wondered this strange animal and asked 'what it is?'

The answer was 'kangaroo.'

Some of my publications on Imputation:

Laaksonen, Seppo (2009). Integrated Modelling Approach to Imputation with Empirical Examples. *NTTS - Conferences on New Techniques and Technologies for Statistics, Eurostat.* Brussels, 18-20 February 2009. Proceedings, 28-36.

Laaksonen, Seppo (2007). Discussion (pp. 467-475) to "Non-Bayesian Multiple Imputation" by J.F. Bjornstad (pp. 433-452) with his rejoinder (pp. 485-491). *Journal of Official Statistics* 23, 4.

Laaksonen, Seppo (2006). Need for High Quality Auxiliary Data Service for Improving the Quality of Editing and Imputation. In: United Nations Statistical Commission, "Statistical Data Editing", Volume 3,

Laaksonen, Seppo, Rässler, Susanne and Skinner, Chris and some other collaborators (2004). Documentation of Pseudo Code of Imputation Methods for the Simulation Study. Dacseis Project Research Papers under Workpackage 11 'Imputation and Nonresponse'. 51 pages. www.dacseis.de/deliverables.

Laaksonen, Seppo (2003). Section 4.6 "German Socio-Economic Household Panel (GSOEP)" for the Euredit Best Practice for Edit/imputation. 12 pages. http://www.cs.york.ac.uk/euredit/Private/index.html.

Laaksonen, Seppo (2003). Alternative Imputation Techniques For Complex Metric Variables. *The Journal of Applied Statistics,* 1009-1020.

Närhi, Vesa, Laaksonen, Seppo, Hietala, Risto, Ahonen, Timo and Lyytinen, Heikki (2001). Treating Missing Data in a Clinical Neuropsychological Dataset—Data Imputation. T*he Clinical Neuropsychologist, 380-392.*

Piela, Pasi and Laaksonen, Seppo (2001). Automatic Interaction Detection for Imputation – Tests with the WAID Software Package. Conference Proceedings of the Federal Committee on Statistical Methodology, Washington, November.

Chambers, R.L., Hoogland, J., Laaksonen, S., Mesa, D.M., Pannekoek, J., Piela, P., Tsai, P. and de Waal, T. (2001). The AUTIMP-project: Evaluation of Imputation Software. Research Paper 0122. Statistics Netherlands.

Laaksonen, Seppo (2000). Regression-Based Nearest Neighbour Hot Decking. *Computational Statistics*. 15,1, 65-71.

Laaksonen, Seppo (1991). Talletus- ja pörssiosaketietojen imputointi vuosien 1987-88 säästämis- ja velkaantumistutkimuksessa (Imputation of missing values of deposits and bonds in the Finnish two-year panel survey 1987-1988). I*n: Ahlqvist, Kirsti, Kangassalo, Pertti, Laaksonen, Seppo and Säylä, Markku. Raportti 1991.5, Statistics Finland*, Section 6, 44- 54.