Universidad de Valladolid

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Estadística

Imputación múltiple de datos ausentes

Autora: Melissa Serrador Ledo

Tutora: Lourdes Barba Escribá

Junio 2017

Contenido

Resumen

La imputación múltiple es una metodología para rellenar valores ausentes (o missing) en conjuntos de datos incompletos y hacer estimaciones e inferencias usando los datos ya imputados. En el presente trabajo se estudian diferentes métodos de imputación múltiple dependiendo del tipo de variables a imputar y del patrón de datos missing, centrándose en el uso de patrones monótonos y arbitrarios. Se exponen tres ejemplos en los que se aplica la imputación múltiple con SAS y en los que se reflejan los tres pasos claves en el proceso MI: definición del modelo de imputación, análisis de los conjuntos de datos completos MI, y estimación e inferencia de los conjuntos de datos imputados de forma múltiple.

Abstract

Multiple imputation is a methodology for filling missing values into incomplete data sets and making estimates and inferences using imputed data. In the present project we study different methods of multiple imputation depending on the type of variables to be imputed and the pattern of missing data, focusing on the use of monotone and arbitrary patterns. We include three examples in which the multiple imputation with SAS is applied and in which the three key steps in the MI process are reflected: definition of the imputation model, analysis of the complete MI data sets, estimation and inference of imputed data set of multiple form.

Palabras clave

MI, imputación múltiple, datos ausentes, missing, PROC MI, PROC MIANALYZE

1. Introducción

La imputación múltiple es una metodología para rellenar valores ausentes (o missing) en conjuntos de datos incompletos y hacer estimaciones e inferencias usando los datos ya imputados. El mecanismo que emplea es la imputación de los missing varias veces, generando así varios conjuntos de datos completos, después realizar las estimaciones sobre estos datos por separado y finalmente hacer una síntesis global. Estos son los tres pasos básicos de la MI (Multiple Imputation). Así, al hacer la imputación más de una vez, se están generando distintos valores que podría tomar un mismo missing basándose en los datos observados, recogiendo más información que si solo se hace una sola imputación, con el objetivo de mejorar los resultados de las estimaciones finales.

La imputación pudo originarse en el Consejo Nacional de Investigadores. Antes de mediados de los 80, se indicaban los missing en un conjunto de datos, pero no se adoptaban medidas para darles un valor. Durante los 80, EEUU. donde la imputación fue promovida por programas como la Encuesta de Ingresos y la Participación en Programas, SIPP, y Canadá emplearon imputación por regresión. A partir de los años 90, se produce una demanda de métodos para problemas cada vez mayores, surgiendo así en SAS procedimientos PROC MI y el análisis estadístico de datos imputados con PROC MIANALYZE.

El concepto de Imputación Múltiple fue formulado por Rubin, un profesor de estadística de la Universidad de Harvard, en 1987. Buscó un procedimiento para que el imputador utilizara los datos disponibles y los métodos de imputación, y el analista analizara los datos imputados y obtuviese inferencias. Pero surgían problemas como que el imputador no incluyese en el modelo una variable importante, que asuma relaciones lineales cuando no las hay o que no incorpore interacciones importantes, de manera que el analista esté sujeto a sesgo. Por eso, se consideró que la mejor opción era que el imputador y el analista fuesen la misma persona, y surgen así estos métodos de imputación múltiple para solventar este problema.

En este trabajo, se describen los tipos de missing, patrones de datos ausentes, y métodos de imputación múltiple. Además, se van a desarrollar algunos ejemplos a partir de conjuntos de datos que se tomaron completamente observados y en los que, aleatoriamente, se han suprimido algunos valores con el objetivo de realizar el proceso de imputación múltiple con diferentes patrones y tipos de variables a imputar

1.1. Atributos de los métodos MI

- Basados en modelos: el modelo de imputación normalmente es explícito, pero también puede ser bueno uno implícito.
- Estocásticos: imputan basándose en los parámetros del modelo y en los errores de la distribución predictiva de los datos ausentes.
- Multivariantes: conservan propiedades distributivas y asociaciones entre variables que se pueden incluir en el modelo.
- Emplean múltiples repeticiones independientes del procedimiento de imputación que permite la estimación de la varianza de los parámetros estimados atribuible a la imputación.
- Robustos contra pequeñas desviaciones.
- Muy útiles en aplicaciones reales.

2. Pasos MI

Se siguen tres pasos clave en el proceso de imputación múltiple:

2.1. Paso 1. Definición del modelo de imputación

En el proceso MI, el primer paso consiste en la elección del modelo de imputación, basándonos en el patrón de datos missing, y en la cantidad y tipo de las variables a imputar. Este método se aplicará tantas veces como número de imputaciones que se hayan fijado (*m*), generando así *m* conjuntos de datos completos.

El orden de las variables a incluir en el modelo de imputación en PROC MI es el siguiente:

i. Incluir variables clave, aunque no tengan missing

ii. Incluir otras variables correlacionadas con las anteriores

iii. Incluir variables que predicen datos ausentes

En caso de duda, es mejor incluir más variables.

Imputar consiste en un sorteo aleatorio de un valor imputado a partir de la distribución predictiva a posteriori $f(Y_{miss}| Y_{obs}, \Theta)$. La imputación requiere obtener la función (paso P) y después imputar valores aleatorios de esa distribución, basándose en relaciones con otras variables (paso I).

Para visualizar el patrón de missing, utilizamos PROC MI con NIMPUTE=0, de manera que no realice ninguna imputación. A continuación, para imputar, se usa el mismo procedimiento, pero ampliando el número de imputaciones, o no añadiendo dicha opción de manera que el número de imputaciones sería, por defecto, 5. Este número de imputaciones se ha comprobado que funciona bien, pero se podría ajustar en función del porcentaje de datos ausentes. También se especificará el método de imputación en cada caso en particular junto con el nombre de la variable a imputar. Finalmente, en la opción OUT= escribiremos el nombre del fichero que almacenará los *m* conjuntos de datos ya imputados.

2.2. Paso 2. Análisis de los conjuntos de datos completos MI

En este segundo paso se realiza el análisis de los *m* conjuntos de datos completos e independientes por separado con un procedimiento estándar, obteniendo estadísticos estimados y errores estándar para cada uno. Se usa dicho procedimiento estándar que utiliza como datos de entrada los de la salida del procedimiento de imputación PROC MI, junto con la sentencia BY _IMPUTATION_ para obtener los resultados asociados a cada uno de los *m* conjuntos de datos

imputados. Aquí, se genera un conjunto de datos de salida con la sentencia ODS OUTPUT que servirá como datos de entrada del procedimiento PROC MIANALYZE del paso 3.

2.3. Paso 3. Estimación e inferencia de los conjuntos de datos imputados de forma múltiple

Finalmente, es el turno de la síntesis de resultados, generando estimaciones, errores estándar, intervalos de confianza, etc. Los estimadores MI son el promedio de los estimadores de las *m* repeticiones del algoritmo de imputación.:

$$\bar{\theta} = \frac{1}{M}\sum_{m=1}^{M}\widehat{\theta_m}$$

Fórmula 1. Media de las estimaciones

donde $\widehat{\theta_m}$ es el estimador de θ para el conjunto de datos completo m=1,…,M.

La varianza de la imputación múltiple se calcula combinando la componente de la varianza dentro de la imputación, como la media de las varianzas estimadas de los *m* análisis, y la componente de la varianza MI.

La varianza dentro de la imputación se calcula como la media de las varianzas estimadas de los $\widehat{\theta_m}$ de los m=1,…,M conjuntos de datos completos:

$$\bar{W} = \frac{1}{M}\sum_{m=1}^{M}\bar{W}_m = \frac{1}{M}\sum_{m=1}^{M} var(\widehat{\theta_m})$$

Fórmula 2. Varianza dentro de la imputación

donde $var(\widehat{\theta_m})$ es la estimación de la varianza de $\widehat{\theta_m}$ para la repetición MI m=1,…,M.

La varianza entre las distintas imputaciones se estima como:

$$B = \frac{1}{M-1}\sum_{m=1}^{M}\left(\widehat{\theta_m} - \bar{\theta}\right)^2$$

Fórmula 3. Varianza entre imputaciones

La varianza total de la estimación Mi de θ se calcula usando la siguiente fórmula combinada de Rubin:

$$var(\bar{\theta}) = \bar{W} + \left(\frac{M+1}{M}\right) * B$$

Fórmula 4. Varianza total

En SAS se usa el procedimiento PROC MIANALYZE con MODELEFFECTS en el que escribiremos 'intercept' seguido de los regresores en el orden correcto en caso de que se quiera estimar una regresión, o 'mean' en caso de que se desee estimar una media.

3. Tipos de datos ausentes y patrones

3.1. Tipos de datos ausentes

Existen tres diferentes tipos de datos ausentes: (1)

- **MCAR** (missing completely at random – datos ausentes completamente aleatorios): se da cuando los datos ausentes son una muestra aleatoria simple de la población sin un proceso subyacente que tiende a sesgar los datos observados.

- **MAR** (missing at random – datos ausentes aleatorios): el patrón de los datos ausentes en una variable Y no es aleatorio, sino que depende de otras variables de la muestra X. Pero para cada valor de X, los valores observados de Y sí representan una muestra aleatoria de Y. Por ejemplo, si X es el sexo del encuestado e Y es su renta, se tendría un proceso MAR si existen más valores ausentes de Y en hombres que en mujeres y, sin embargo, los datos son aleatorios para ambos sexos en el sentido de que, tanto en los hombres como en las mujeres, el patrón de ausentes es completamente aleatorio. Si, además, tampoco existiesen diferencias por sexos, los datos ausentes serían MCAR.

- **MNAR** (missing not at random – datos ausentes no aleatorios): surgen cuando existen patrones sistemáticos en el proceso de datos ausentes

3.2. Tipos de patrones

Suponemos que tenemos conjuntos de datos dados en la forma de casos con sus variables, ya sean datos de encuestas o mediciones de diferentes características en individuos de una población. Los datos missing se pueden distribuir de diferentes formas dentro del conjunto de datos: puede haber valores ausentes en una sola variable, o en varias. Existen tres tipos de patrones ('X' representa dato observado y '.' representa dato missing):

- **Patrón monótono**

Es un patrón en el que, si en un individuo no se observa la variable Y2, no se observará tampoco la Y3; y si no se observa la variable Y1, tampoco se observarán las variables Y2 y Y3. Se comenzará imputando la variable Y2 utilizando las relaciones entre los valores de Y2 con sus correspondientes valores de Y1. Esto puede ocurrir en datos de encuestas donde hay diferentes bloques de preguntas y el encuestado puede que no desee contestar a preguntas relacionadas con algún tema en particular, o simplemente decida no terminar la encuesta.

Obs	Y1	Y2	Y3
1	X	X	X
2	X	X	X
3	X	X	.
4	X	.	.
5	.	.	.

Tabla 1

- **Patrón generalizado o arbitrario**

Este patrón, la mayoría de las veces, es primero transformado a un patrón monótono, imputando el número necesario de valores ausentes para ello, y después utilizando un método de imputación para patrones monótonos.

Obs	Y1	Y2	Y3
1	X	X	.
2	.	.	X
3	.	X	.
4	X	.	X
5	.	X	X

Tabla 2

- **Patrón matricial**

Este patrón surge en datos de encuestas donde hay preguntas básicas que se les hace a todos los individuos, y otras más específicas asignadas aleatoriamente a submuestras de individuos.

Obs	Y1	Y2	Y3
1	X	.	X
2	X	.	X
3	X	X	.
4	X	X	.
5	X	X	.

Tabla 3

En este trabajo nos centraremos en el uso de patrones arbitrarios y monótonos.

4. Métodos de imputación

Existen diferentes métodos de imputación, dependiendo del patrón de datos ausentes y del tipo de variables a imputar:

Patrón	Tipo de variable	Método
Monótono	Continua	Regresión lineal, método de emparejamiento predictivo de la media, propensity score
	Binaria /ordinal	Regresión logística
	Nominal	Función discriminante
Arbitrario	Continua, con covariables continuas	Método monótono MCMC
	Continua, con mixtura de covariables	Regresión FCS, método de emparejamiento predictivo de la media FCS
	Binaria /ordinal	Regresión logística FCS
	Nominal	Función discriminante FCS

Tabla 4

4.1. Métodos para patrones monótonos

En caso de tener un patrón monótono se usa MONOTONE, que imputa una variable a la vez basándose en la distribución a posteriori.

• **Regresión lineal**: para imputar variables continuas. Recordemos el patrón monótono (tabla 1). Si Y2 es continua el método MONOTONE REGRESSION usará el modelo estándar

$$Y2=\beta_0+\beta_1*Y1+\varepsilon$$

Fórmula 5

En el paso P, la regresión obtendrá las estimaciones $\hat{\beta}=(\hat{\beta}_0, \hat{\beta}_1)$, la varianza residual $\hat{\sigma}_2^2$ y V2 (suma de cuadrados de la matriz de productos cruzados de la regresión de Y2 sobre Y1). Esto define la distribución a posteriori. En el paso I se imputa Y2, y después se continua con el resto de variables.

En PROC MI se especifica como método de imputación MONOTONE REGRESSION seguido de la variable a imputar. Si además añadimos '/details', obtenemos los parámetros estimados para cada predictor e imputación.

- **Emparejamiento predictivo de la media o predictive mean matching**: para imputar variables continuas. Define la distribución a posteriori de la misma manera que la regresión lineal. En el paso I, en cambio, busca los *k* (por defecto 5, aunque también lo podemos fijar nosotros mismos) vecinos más próximos de cada valor ausente, es decir, los *k* casos observados que más se le parecen, y entre ellos elige uno al azar.

En SAS, el método de imputación se escribiría como MONOTONE REGPMM seguido del nombre de la variable a imputar, y también permite la opción '/details'.

- **Puntuaje de propensión o propensity score**: para imputar variables continuas. Es un método univariante que no incorpora asociaciones entre variables.

- **Regresión logística**: para imputar variables binarias u ordinales. El proceso de imputar valores bajo este modelo sigue una secuencia de pasos P y pasos I similar a la del método de regresión lineal. En el paso P se aplica la regresión logística a los casos observados de las variables dependientes y los predictores, obteniendo el modelo logístico ajustado para estimar la probabilidad de que el valor missing pertenezca a cada categoría de la variable de clasificación.

Supongamos que la variable dependiente es binaria. El modelo ajustado es:

$$logit\big(p(Y_4 = 1)\big) = \log\left(\frac{p}{1-p}\right) = \widehat{\beta_0} + \widehat{\beta_1} * Y_1 + \widehat{\beta_2} * Y_2 + \widehat{\beta_3} * Y_3$$

Fórmula 6

El modelo de regresión logística ajustado produce estimaciones de los parámetros de la regresión logística y la matriz de covarianzas V. La distribución a posteriori de los parámetros de regresión logística se asume normal multivariante.

Seguidamente se calcula la inversa del logit:

$$p(Y_4) = \exp\left(\frac{Y_{j<4}\beta_*}{1 + \exp\big(Y_{j<4}\beta_*\big)}\right)$$

Fórmula 7

donde $\beta_* = \hat{\beta} + \sqrt{V} * Z$ con Z= vector (k+1)-dimensional de normales independientes.

En el paso I, se extrae un número aleatorio de una uniforme U(0,1) para determinar si la variable dependiente vale 0 o 1. Por ejemplo, si el modelo logístico predice que la probabilidad de que la variable dependiente valga 0 es *p* y el valor extraído de la uniforme es mayor que *0*, entonces se imputará 1, y en el caso de que sea menor que *x* se imputará 0.

- **Función discriminante**: para imputar variables nominales. En el paso P simula los parámetros o probabilidades de grupo de la distribución multinomial a posteriori para Y, y las relaciona con un vector de covariables observadas X. Dentro de cada grupo de categoría, el vector de covariables se asume aproximadamente normal multivariante, y la matriz de varianzas-covarianzas se supone constante para todos los grupos. La distribución predictiva de los missing es multinomial con probabilidad de pertenecer a la categoría g:

$$\mathrm{p}\left(Y_{i,mis} = \mathrm{g}|x_i\right) = \frac{e\exp(-0.5*D_g^2(x_i)}{\sum_g \exp\left(-0.5*D_g^2(x_i)\right))}$$

Fórmula 8

donde $D_g^2(x_i)$ es el cuadrado de la distancia del valor discriminante para el caso i y el grupo g.

En el paso I, extrae un número aleatorio de una uniforme U(0,1) que se compara con las probabilidades de las g categorías para determinar la categoría del valor imputado.

4.2. Métodos para patrones arbitrarios

MCMC y FCS usan algoritmos iterativos diseñados para simular sorteos de la distribución predictiva a posteriori multivariante de variables con patrón arbitrario de datos ausentes.

El método de la Cadena de Markov de Monte Carlo (MCMC), se diseñó para imputar valores en un vector de variables continuas que se asume que están distribuidas conjuntamente como normales multivariantes. Es utilizado para imputar variables continuas con covariables también continuas. Se usa el método de Monte Carlo para simular $p(\mu,\Sigma|Y_{obs})$ y supone una secuencia iterativa de pasos I y P. En el paso I en cada iteración genera imputaciones de la distribución predictiva; en el paso P se actualizan los valores de los parámetros de la distribución predictiva. El algoritmo converge al hacer muchas iteraciones de estos dos pasos, y este número de iteraciones dependerá del número de variables y del patrón de datos ausentes. PROC MI de SAS hace por defecto 200 iteraciones (NBITER=200), proporciona gráficos para evaluar la convergencia del algoritmo, y permite trazar estimaciones posteriores de medias y varianzas.

Además, este método también es usado para imputar variables con tasa de missing baja para transformar el patrón arbitrario a monótono y después usar un método monótono para imputar el resto de datos ausentes.

El algoritmo de regresión secuencial o regresiones encadenadas (FCS) no hace suposiciones fuertes sobre la naturaleza de la distribución conjunta de las variables y maneja fácilmente mixturas de variables continuas y categóricas. En cada iteración se mueve a una de las variables del modelo de imputación. Para cada iteración y cada variable hay un paso P y un paso I. En el primero, usa los valores observados e imputados en la iteración actual para obtener la distribución predictiva de los

datos missing para la variable objetivo. Para modelar la distribución predictiva condicional de los Y_k, usa los mismos métodos de regresión o función discriminante descritos para patrones monótonos. Las imputaciones son generadas a partir de la distribución predictiva. Cuando la última variable ha sido imputada, repite los pasos hasta que converge.

Antes de que existieran los algoritmos FCS, se imputaban mixturas de variables categóricas y continuas como normales multivariantes. Los valores imputados a las variables categóricas eran redondeados a valores de categoría enteros, pero parece más apropiado utilizar un método que incorpore las propiedades de la variable a imputar.

5. Imputación múltiple para el análisis de datos complejos de encuestas por muestreo

Los diseños de encuestas por muestreo que incorporan estratificación y agrupamiento (o clustering) pueden requerir ponderación de cada caso para tener en cuenta las diferencias en la selección de la muestra y la probabilidad de respuesta para las unidades observadas. A veces es necesario incorporar estas características del diseño de la recolección de datos en la imputación, estimación e inferencia.

Según Rubin[2], en caso de tener una muestra con estratos y pesos de cada caso, se deben incluir estos en la imputación MI, ya que incluir predictores no importantes no supone un gran coste. A través de un modelo mixto lineal o multinivel, se puede modelar la relación de la estratificación o clustering:

$$Y_i = \beta * x_i + \Upsilon_{str,i} + u_{cl(str),i} + \varepsilon_{i(str,cl)}$$

Fórmula 9

donde:

x_i son los efectos fijos de las covariables de interés

$\Upsilon_{str,i}$ es el efecto fijo para el estrato al que pertenece cada individuo

$u_{cl(str),i}$ es el efecto fijo para el cluster

El modelo reducido, que incluye un efecto fijo para el peso y otro correspondiente a la interacción entre estrato y cluster, evita los problemas de inestabilidad que se pueden encontrar en el modelo completo. Es la expresión que se emplea en PROC MI:

$$Y_i = \beta * x_i + \delta * w_i + \Upsilon_{\text{strata x cluster}} + \varepsilon_i$$

Fórmula 10

donde:

w_i es el peso para el caso *i*

$\Upsilon_{\text{strata x cluster}}$ es el efecto fijo correspondiente al estrato y cluster al que pertenece el caso

Supongamos que tenemos un conjunto de datos y queremos estimar la media de una variable. Existen tres enfoques diferentes:

5.1. Incorporando las variables de diseño y ponderación en el modelo de imputación

En este caso, se incluyen la variable combinada estrato-cluster, una variable que se crea como combinación de los dígitos de la variable de estrato y de la variable de cluster (obteniendo así una variable categórica con tantas categorías como combinaciones de estos dígitos) y la variable de peso en el modelo de imputación.

Los datos de muestras complejas imponen dos requisitos para asegurar que las estimaciones e inferencias sean robustas y reflejen los efectos de la estratificación, clustering y ponderación:

1. Se debe utilizar PROC SURVEY para generar las estimaciones y errores estándar para cada uno de los *m* conjuntos de datos, con las sentencias STRATA, CLUSTER y WEIGHT.

2. Se debe usar la opción EDF de PROC MIANALYZE para determinar correctamente los grados de libertad (que aproximadamente son nº_clusters – nº_estratos) de los datos completos usados en la construcción de intervalos de confianza.

5.2. Análisis MI ignorando las variables de diseño y ponderación

Se hace la imputación omitiendo las variables de diseño complejas y los pesos. Sin embargo, sí que las incluimos en la estimación MI e inferencia (pasos 2 y 3) en STRATA, CLUSTER y WEIGTH en PROC SURVEYMEANS, además de añadir la opción EDF para fijar los grados de libertad en PROC MIANALYZE.

5.3. Análisis MI incluyendo las variables de estrato y cluster en el modelo de imputación y los pesos en la sentencia FREQ de PROC MI

Este último procedimiento consiste en usar la variable combinada estrato-cluster como predictor lineal y el peso en la sentencia FREQ de PROC MI. En caso de tener un peso que no es entero, se multiplicaría por una constante de manera que lo convirtamos en entero para FREQ.

Una vez ejecutados los tres procedimientos, se ha comprobado que, en las estimaciones de la media de la variable de interés, los errores estándar y los intervalos de confianza de cada uno no existen diferencias significativas en la mayoría de los casos, pero esto no significa que las características de diseño sean ‘no-informativas’. De hecho, en muestras pequeñas donde las correlaciones intra-clases son grandes, no se deberían omitir las variables de diseño.

6. Ejemplo 1: imputación múltiple de variables contínuas con patrón arbitrario y covariables contínuas

(Datos extraídos del apéndice 4 del libro "Applied Linear Statistical Models")

El director de una universidad quiere ver si las calificaciones medias de los alumnos al final de su primer año se pueden predecir a partir de los resultados de las pruebas de ingreso y el rango de la escuela secundaria.

Se tiene información de un total de 705 alumnos universitarios entre los años 1996 y 2000, y las variables son:

Id: identificador de estudiante

GPA: promedio de las calificaciones del primer año, continua con missing

High_school_class_rank: rango de la escuela secundaria dado como percentil: los percentiles inferiores significan altos rangos en la clase, continua y completa

ACT_score: calificación en el examen de ingreso, continua con missing

Academic_year: año en el que el estudiante ingresó en la universidad, discreta con 6 valores distintos, sin missing

Utilizando el software SAS, vamos a realizar los tres pasos del procedimiento MI para imputar los missing y estimar el modelo, donde la variable independiente será *GPA* y los regresores serán *High_school_class_rank* y *ACT_score* según desea el director.

```
data apendice4;
  input id GPA High_school_class_rank ACT_score Academic_year;
  cards;
1 0.98 61 NA 1996
2 1.13 84 20 1996
3 1.25 74 19 1996
…
705 4 99 32 2000
; run;
```

El primer paso del proceso MI es obtener el patrón de datos missing con el procedimiento PROC MI y la opción NIMPUE=0 de manera que no realice ninguna imputación, junto con las estadísticas univariantes y correlaciones gracias a la opción SIMPLE:

```
proc mi data=apendice4 nimpute=0 simple;
  var GPA High_school_class_rank ACT_score Academic_year;
run;
```

Missing Data Patterns						
Group	**GPA**	**High_school_class_rank**	**ACT_score**	**Academic_year**	**Freq**	**Percent**
1	X	X	X	X	646	91.63
2	X	X	.	X	39	5.53
3	.	X	X	X	20	2.84

Tabla 5

Missing Data Patterns				
Group	**Group Means**			
	GPA	**High_school_class_rank**	**ACT_score**	**Academic_year**
1	2.986889	76.886997	24.586687	1997.996904
2	2.795718	77.717949	.	1998.000000
3	.	77.600000	25.000000	1998.100000

Tabla 6

Si nos fijamos en el patrón de datos missing (tabla 5), vemos que un 2.84% de los estudiantes tienen valor ausente en la variable *GPA* (*promedio de las calificaciones del primer año*) y un 5.53% no tienen asignado un valor en la variable *ACT_score* (*calificación en el examen de ingreso*), ambas variables continuas. En el resto de estudiantes se han observado todas las variables. De esta manera, nos encontramos ante un patrón de datos ausentes arbitrario con tres grupos de datos:

- ✓ casos en los que se tienen todas las variables completas (646 alumnos)
- ✓ casos con missing en la variable *GPA* (20 alumnos)
- ✓ casos con missing en la variable *ACT_score* (39 alumnos)

En la tabla 6 podemos ver las medias por grupos. Por ejemplo, para nuestra variable de interés, *GPA*, la media del grupo de individuos que tiene valor en todas las variables es de 2.986889, mientras que la media en el grupo con valor missing en la variable *ATC_score* es de 2.795718.

En la tabla de estadísticas univariantes (tabla 7), se nos proporciona información sobre la media, desviación estándar, mínimo, máximo, número de missing, porcentaje de missing, y número de casos con valor en esa variable.

Univariate Statistics							
Variable	**N**	**Mean**	**Std Dev**	**Minimum**	**Maximum**	**Missing Values**	
						Count	**Percent**
GPA	685	2.97600	0.63336	0.51000	4.00000	20	2.84
High_school_class_rank	705	76.95319	18.63394	4.00000	99.00000	0	0.00
ACT_score	666	24.59910	4.05006	13.00000	35.00000	39	5.53
Academic_year	705	1998	1.40413	1996	2000	0	0.00

Tabla 7

La media del *promedio de las calificaciones del primer año* de los alumnos es de 2.976 con una desviación estándar de 0.63336, siendo el promedio mínimo de 0.51 y el máximo de 4. Por otro lado, la media de la variable *calificación en el examen de ingreso* es de 24.59910, siendo la mínima nota un 13 y la máxima, 35. El *rango de la escuela secundaria* está entre 4 y 99.

Pairwise Correlations				
	GPA	**High_school_class_rank**	**ACT_score**	**Academic_year**
GPA	1.000000000	0.387062731	0.351888361	0.024519245
High_school_class_rank	0.387062731	1.000000000	0.441361171	-0.015363873
ACT_score	0.351888361	0.441361171	1.000000000	0.014117990
Academic_year	0.024519245	-0.015363873	0.014117990	1.000000000

Tabla 8

En la tabla de correlaciones (tabla 8) encontramos valores bajos, todos por debajo de 0.5, lo que nos indica que las variables no están muy correlacionadas y que los cambios en una de ellas no influirán de manera significativa en el resto.

Seguidamente, y continuando en el primer paso del proceso MI, procedemos a la imputación de los valores que no han sido observados en la muestra. Empleamos el algoritmo MCMC utilizado para imputar variables continuas con patrón de missing arbitrario y covariables también continuas. Solicitamos, además, dos plots con los que podremos comprobar la convergencia del algoritmo de imputación: uno con la traza enfrentando las medias de nuestras dos variables a imputar contra el número de iteración, y otro con el ACF. La opción OUT= nos crea un fichero con los 5 conjuntos de datos imputados que usaremos más adelante. Se eligen 5 imputaciones puesto que contamos con un porcentaje pequeño de datos ausentes.

```
proc mi data=apendice4 out=outap4 seed=192 nimpute=5
      min= 0.51 . 13
      max= 4 . 35
      round = 0.01 . 1;
  var GPA High_school_class_rank ACT_score;
  mcmc
  plots=(trace(mean(GPA) mean(ACT_score))
      acf(mean(GPA) mean(ACT_score)));
run;
```

De estas líneas de código, se obtienen:

- trazas de la media de cada variable a imputar frente al número de iteración
- ACF de las dos variables
- tabla de información de la varianza: varianza entre las distintas imputaciones, varianza dentro de las imputaciones, varianza total, incremento relativo de la varianza, fracción de información perdida debido a la imputación, eficiencia relativa
- tabla de parámetros estimados para las dos variables: media, error estándar, intervalo de confianza, mínimo, máximo, test para contrastar H0: $\mu=\mu_0=0$

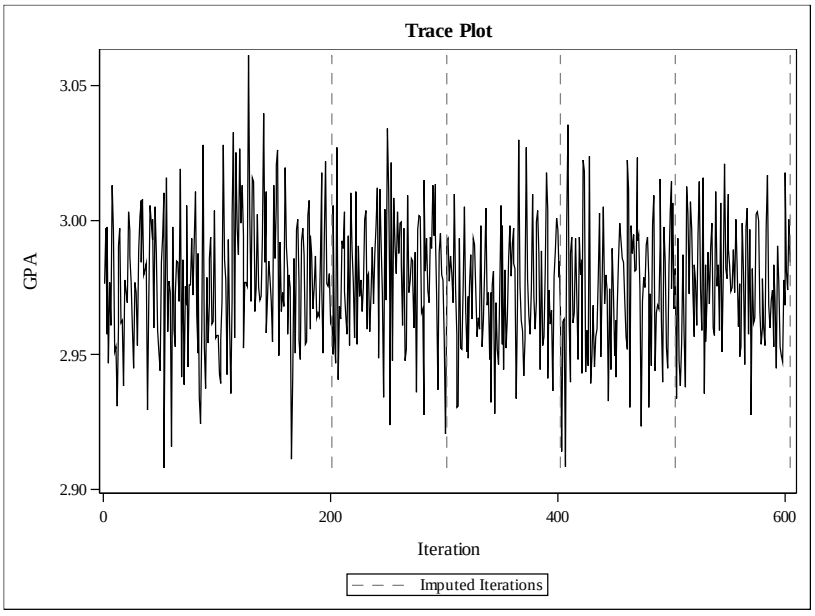

Grafico 1

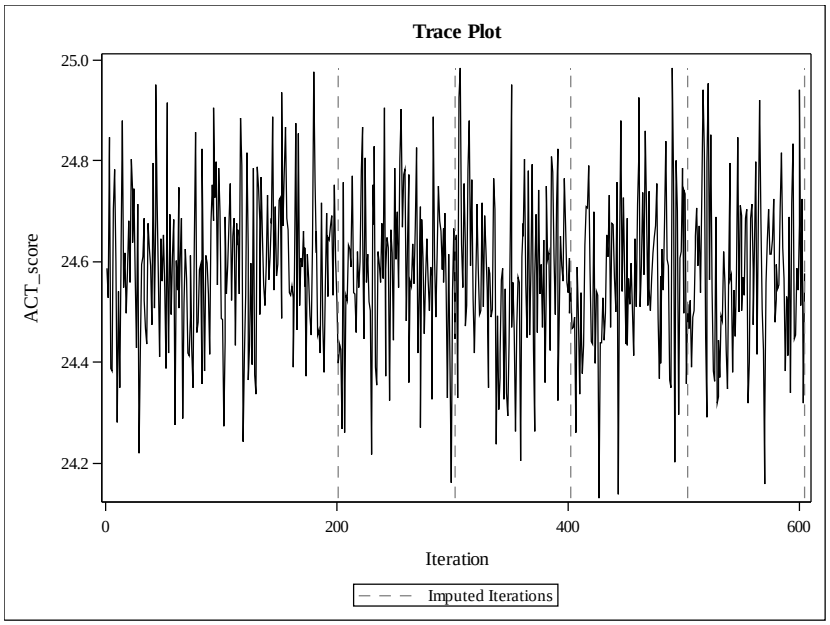

Grafico 2

Las trazas de las medias de *GPA* frente al número de iteración (gráfico 1), vemos que fluctúan aleatoriamente en torno a un valor medio de 2.97 aproximadamente sin tendencia creciente ni decreciente, al igual que las trazas de las medias de las *notas del examen de ingreso* (gráfico 2), en torno a 24.6. Estas características nos indican aleatoriedad en los datos y no un patrón sistemático.

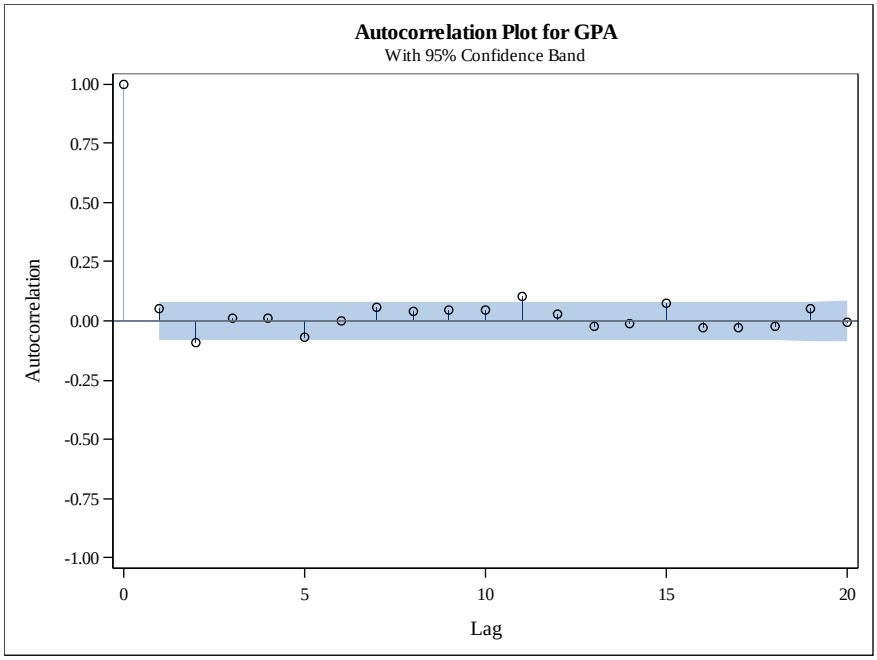

Grafico 3

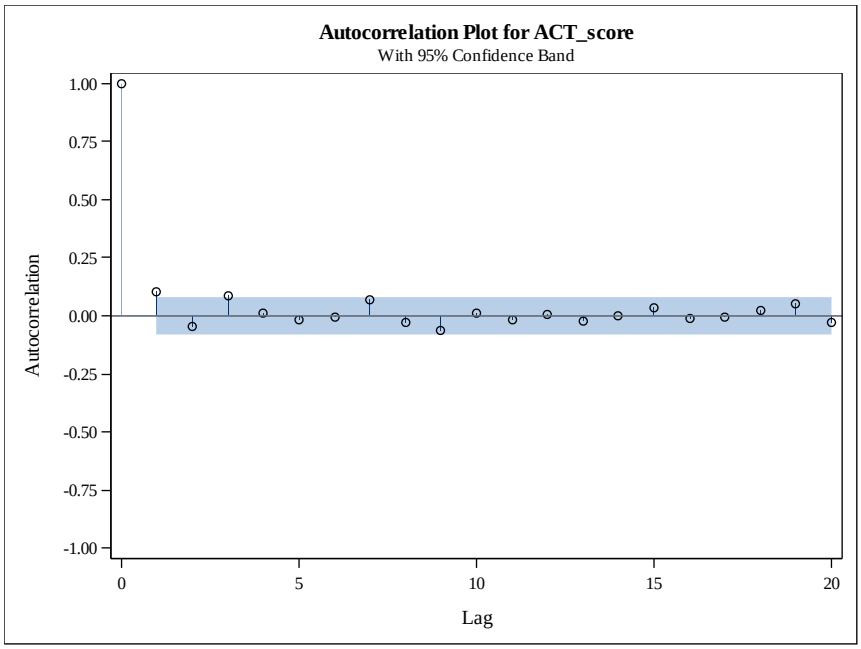

Grafico 4

Los autocorrelogramas de la media de las variables (gráficos 3 y 4) muestran 20 retardos por defecto, aunque esto se puede cambiar con la opción NLAG=. Aquí vemos que las autocorrelaciones son despreciables al variar en torno al 0, de manera que no hay problemas de convergencia ni independencia de las repeticiones.

Variance Information							
Variable	**Variance**			**DF**	**Relative Increase in Variance**	**Fraction Missing Information**	**Relative Efficiency**
	Between	**Within**	**Total**				
GPA	0.000002086	0.000568	0.000571	696.59	0.004407	0.004397	0.999121
ACT_score	0.001418	0.023077	0.024778	369.25	0.073727	0.070853	0.986027

Tabla 9

En la tabla de Información de la Varianza (tabla 9), obtenemos los valores de la varianza dentro de la imputación, entre las imputaciones y total (calculadas con las fórmulas 2, 3 y 4 respectivamente), para cada una de las dos variables a imputar. Además, nos proporciona los incrementos relativos de la varianza para cada una de las dos variables imputadas, que resultan ser muy bajos; la fracción de información perdida debida a la imputación que también es baja para ambas variables; y la eficiencia relativa, con valores elevados, muy próximos a 1.

Parameter Estimates										
Variable	**Mean**	**Std Error**	**95% Confidence Limits**		**DF**	**Min**	**Max**	**Mu0**	**t for H0: Mean=Mu0**	**Pr > |t|**
GPA	2.974	0.0238	2.927	3.021	696.59	2.972	2.97	0	124.52	<.0001
ACT_score	24.591	0.1574	24.281	24.900	369.25	24.54	24.6	0	156.22	<.0001

Tabla 10

La última tabla que nos devuelven estas instrucciones de SAS es la de los Parámetros Estimados (tabla 10). La media estimada para la variable *promedio de las notas del primer año* es de 2.974 con un intervalo de confianza del 95% de (2.927, 3.021). El test para contrastar la hipótesis nula de que la media de dicha variable sea igual a 0 devuelve un p-valor menor que 0.0001, de manera que se rechaza la hipótesis en favor de la alternativa de que esa media sea distinta de 0. Lo mismo ocurre para la variable *calificación en el examen de ingreso*, donde su media estimada resulta ser 24.591 y su intervalo de confianza (24.281, 24.900).

Con un PROC MEANS vamos a ver las medias de las dos variables imputadas para cada una de las cinco imputaciones y así comprobar que no hay ningún error en las imputaciones:

```
proc means data=outap4 nonobs;
  class _imputation_;
  var GPA;
run;
```

Analysis Variable : GPA					
Imputation Number	**N**	**Mean**	**Std Dev**	**Minimum**	**Maximum**
1	705	2.9742028	0.6358216	0.5100000	4.0000000
2	705	2.9739050	0.6308200	0.5100000	4.0000000
3	705	2.9726993	0.6323207	0.5100000	4.0000000
4	705	2.9766709	0.6312916	0.5100000	4.0000000
5	705	2.9742312	0.6340392	0.5100000	4.0000000

Tabla 11

Para la variable *nota media del primer año*, la media para los datos que se observaron era de 2.976 con una desviación estándar de 0.633, mientras que la media de esta misma variable para los datos que han sido imputados, por ejemplo, en la primera iteración es de 2.974 con una desviación de 0.636. Los valores medios del resto de imputaciones también resultan ser muy similares a los de los datos observados.

```
proc means data=outap4 nonobs;
  class _imputation_;
  var ACT_score;
run;
```

Analysis Variable : ACT_score					
Imputation Number	**N**	**Mean**	**Std Dev**	**Minimum**	**Maximum**
1	705	24.5687943	4.0513244	13.0000000	35.0000000
2	705	24.5460993	4.0418183	13.0000000	35.0000000
3	705	24.6354610	4.0186556	13.0000000	35.0000000
4	705	24.6241135	4.0479004	13.0000000	35.0000000
5	705	24.5815603	4.0077270	13.0000000	35.0000000

Tabla 12

Por otro lado, la media de los datos observados de la variable *nota en el examen de ingreso*, al igual que ocurría con *GPA*, las medias de las 5 repeticiones de los datos imputados no distan de manera significativa (tampoco pueden ser idénticas debido a la aleatoriedad del proceso de imputación) de la observada 24.599. En definitiva, las imputaciones presentan normalidad.

En el paso 2, se debe emplear un procedimiento estándar a los conjuntos de datos imputados para estimar nuestro objetivo en cada uno de ellos. En este caso es la regresión de *GPA* sobre *High_school_class* y *ATC_score* con un PROC REG. Indicamos con la opción OUTEST= el nombre que queremos dar al conjunto de datos donde se almacenarán las 10 regresiones, y que servirán como entrada en el PROC MIANALYZE del paso 3.

```
proc reg data=outap4 outest=out_est_ap4 covout;
  model GPA = High_school_class_rank ACT_score;
  by _imputation_;
run;
```

Imputation number=1

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	2	54.67728	27.33864	83.47	<.0001
Error	702	229.92813	0.32753		
Corrected Total	704	284.60541			

Tabla 13

Root MSE	0.57230	**R-Square**	0.1921
Dependent Mean	2.97420	**Adj R-Sq**	0.1898
Coeff Var	19.24230		

Tabla 14

Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	1	1.33957	0.13708	9.77	<.0001
High_school_class_rank	1	0.00975	0.00129	7.55	<.0001
ACT_score	1	0.03600	0.00593	6.07	<.0001

Tabla 15

De las salidas que nos producen estas órdenes, solo muestro el modelo estimado para los datos de la primera imputación, que resulta ser:

$$GPA=1.33+0.00975*High\_chool\_class\_rank+0.036*ATC\_score$$

Todos los regresores resultan ser significativos, y cabe destacar el valor tan bajo de R^2, un 0.1921, de manera que la proporción de variabilidad explicada por dicho modelo es reducida.

SAS también proporciona gráficos habituales, como el de residuos y normalidad, pero que no incluyo porque no son el objetivo de este trabajo.

El tercer y último paso del proceso MI es la síntesis de resultados. Para ello, empleamos el procedimiento PROC MIANALYZE con MODELEFFECTS en el que enumeramos los predictores del modelo después de 'intercept'.

```
proc mianalyze data=out_est_ap4;
 modeleffects intercept High_school_class_rank ACT_score;
run;
```

Parameter Estimates							
Parameter	**Estimate**	**Std Error**	**95% Confidence Limits**		**DF**	**Minimum**	**Maximum**
intercept	1.338804	0.138224	1.067859	1.609748	10884	1.309574	1.355503
High_school_class_rank	0.009776	0.001305	0.007218	0.012335	3718.4	0.009432	0.010017
ACT_score	0.035917	0.006008	0.024139	0.047694	5855.5	0.034915	0.036869

Tabla 16

El modelo MI estimado es:

$$GPA=1.339+0.009776*High\_school\_class\_rank+0.035917*ATC\_score$$

7. Ejemplo 2: imputación múltiple de variables continuas con patrón arbitrario y mixtura de covariables

(Datos extraídos del apéndice 7 del libro "Applied Linear Statistical Models")

Partimos de un conjunto de 522 viviendas de una ciudad de medio oeste de las cuales se recogen una serie de variables:

sales_price: precio de la vivienda (en dólares), continua con missing

finished_square_feet: área construida (en pies cuadrados), continua con missing

bedrooms: número de habitaciones, continua y completa

bathrooms: número de baños, continua y completa

air_conditioning: 1-si tiene aire acondicionado, 0-si no, binaria y completa

garage_size: capacidad del garaje (en número de vehículos), continua y completa

pool: 1-si tiene piscina, 0-si no, binaria y completa

real_built: año en que fue construida, continua y completa

quality: 1-calidad de la construcción alta, 2-calidad media, 3-calidad baja, categórica y completa

style: estilo, categórica con 11 categorías y completa

lot_size: tamaño de solar (en pies cuadrados), continua y completa

adjacent_to_highway: 1-si está próxima a la autopista, 0-si no, binaria y completa

El objetivo es predecir los *precios de venta* (*sales_price*) de casas residenciales como una función de varias características de estas, restringiéndonos a las viviendas de calidad media, y además estimar la media de dicha variable.

```
data apendice7;
  input id sales_price finished_square_feet bedrooms bathrooms air_conditioning
garage_size pool real_built quality style lot_size adjacent_to_highway;
  cards ;
1 360000 3032 4 4 1 2 0 1972 2 1 22221 0
2 340000 2058 4 2 1 2 0 1976 2 1 22912 0
3 250000 1780 4 3 1 2 0 1980 2 1 21345 0
…
522 95500 1184 2 1 0 1 0 1951 3 1 14786 0
;run;
```

Con un PROC MEANS vamos a examinar la cantidad de missing, la media, el mínimo y el máximo de cada variable, seleccionando con un WHERE únicamente las viviendas de calidad media:

```
proc means data=apendice7 n nmiss mean min max;
  where quality=2;
  var sales_price finished_square_feet bedrooms bathrooms air_conditioning
garage_size pool real_built quality style lot_size adjacent_to_highway;
run;
```

Variable	N	N Miss	Mean	Minimum	Maximum
sales_price	238	52	276491.39	150000.00	675000.00
finished_square_feet	260	30	2324.00	1384.00	4150.00
bedrooms	290	0	3.6517241	1.0000000	6.0000000
bathrooms	290	0	2.9034483	1.0000000	7.0000000
air_conditioning	290	0	0.9310345	0	1.0000000
garage_size	290	0	2.1517241	0	5.0000000
pool	290	0	0.0758621	0	1.0000000
real_built	290	0	1970.49	1908.00	1998.00
quality	290	0	2.0000000	2.0000000	2.0000000
style	290	0	3.7758621	1.0000000	11.0000000
lot_size	290	0	24793.76	6734.00	86830.00
adjacent_to_highway	290	0	0.0206897	0	1.0000000

Tabla 17

Al seleccionar el subconjunto de viviendas de calidad media, nos quedamos únicamente con 290 de las 522. Si nos fijamos, la variable *precio de la vivienda* contiene un total de 52 valores ausentes, su media es de 276491.39 dólares y los valores mínimo y máximo de los datos observados son 150000 y 675000 respectivamente. Estos valores los usaremos más adelante para acotar la imputación de esta variable al rango de los datos observados. La otra variable que solo se observó en 260 individuos, es la del *área construida*, con media 2324 pies cuadrados, y mínimo y máximo 1384 y 4150.

Examinamos el patrón y la cantidad de missing de todas las variables del problema:

```
proc mi nimpute=0 data=apendice7;
  where quality=2;
  var sales_price finished_square_feet bedrooms bathrooms air_conditioning
garage_size pool real_built quality style lot_size adjacent_to_highway;
run;
```

Missing Data Patterns													
Group	**Sales price**	**Finished square feet**	**bedrooms**	**Bathrooms**	**Air conditioning**	**Garage size**	**pool**	**Real built**	**style**	**Lot size**	**Adjacent to highway**	**Freq**	**Percent**
1	X	X	X	X	X	X	X	X	X	X	X	214	73.79
2	X	.	X	X	X	X	X	X	X	X	X	24	8.28
3	.	X	X	X	X	X	X	X	X	X	X	46	15.86
4	.	.	X	X	X	X	X	X	X	X	X	6	2.07

Tabla 18

El conjunto de datos contiene dos variables con missing que son ambas continuas:

- *finished_square_feet* con 24+6=30 missing
- *sales_price* con 46+6=52 missing

Se tiene un patrón arbitrario con cuatro grupos de datos:

✓ 214 viviendas en las que todas las variables se observaron (un 73.79%)

✓ 24 viviendas en las que no se observa la variable *finished_square_feet* (un 8.28%)

✓ 46 viviendas en las que no se observa la variable *sales_price* (un 15.86%)

✓ 6 viviendas en las que no se observa la variable *finished_square_feet* ni *sales_price* (un 2.07%)

Teniendo en cuenta que hay dos variables con missing, que son ambas continuas, que el patrón de datos ausentes es arbitrario y que tenemos mixtura de covariables continuas y categóricas, tendremos que emplear FCS para imputar, puesto que, al incluir variables de clasificación en el modelo de imputación como *style*, se incumplen las suposiciones de MCMC de que sean normales multivariantes.

FCS REGRESSION

A continuación, imputamos con el método FCS REGRESSION y obtenemos los parámetros estimados para cada predictor e imputación con la opción DETAILS:

```
proc mi data=apendice7 nimpute=10 seed=400 out=ap7_imp
  min= 150000 1384 . . . . . .
  max= 675000 4150  . . . . . .
  round= 1 1 . . . . . .;
  where quality=2;
  var  sales_price  finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway;
  class style; *las variables binarias no es necesario incluirlas en class;
  fcs     nbiter=10     regression     (sales_price     /details)     regression
(finished_square_feet);
run;
```

Después, comprobamos que las estimaciones de los parámetros para las 10 imputaciones no difieran mucho con un PROC MEANS, que nos da la cantidad, media, desviación estándar, mínimo y máximo de los datos separados por número de imputación. No es necesario seguir poniendo el WHERE, puesto que utilizamos los datos almacenados en *ap7_imp* que solo contienen datos de viviendas de clase media.

```
proc means data=ap7_imp;
  class _imputation_;
  var sales_price;
run;
```

Analysis Variable : sales_price				
Imputation Number	**Mean**	**Std Dev**	**Minimum**	**Maximum**
1	278473.47	87511.93	150000.00	675000.00
2	276226.41	84936.12	150000.00	675000.00
3	275826.81	85769.72	150000.00	675000.00
4	274876.01	85584.28	150000.00	675000.00
5	276464.24	84811.55	150000.00	675000.00
6	274739.12	85556.03	150000.00	675000.00
7	275905.32	86495.37	150000.00	675000.00
8	276703.12	88776.39	150000.00	675000.00
9	275284.02	85319.93	150000.00	675000.00
10	275604.50	86733.57	150000.00	675000.00

Tabla 19

Observamos que las estimaciones de los parámetros del modelo de imputación para las 10 imputaciones son muy similares, lo que sugiere estabilidad en las imputaciones, y además no distan demasiado de la media de los valores observados, 276491.39.

```
proc means data=ap7_imp;
  class _imputation_;
  var finished_square_feet;
run;
```

Analysis Variable : finished_square_feet				
Imputation Number	**Mean**	**Std Dev**	**Minimum**	**Maximum**
1	2311.51	546.3886422	1384.00	4150.00
2	2318.01	535.4750437	1384.00	4150.00
3	2319.73	539.7448823	1384.00	4150.00
4	2307.53	533.0445042	1384.00	4150.00
5	2318.28	533.1770054	1384.00	4150.00
6	2316.24	535.0529469	1384.00	4150.00
7	2319.34	542.0108897	1384.00	4150.00
8	2320.29	549.3910945	1384.00	4150.00
9	2311.62	544.3524709	1384.00	4150.00
10	2325.09	540.4046430	1384.00	4150.00

Tabla 20

En las imputaciones de la variable *área construida* tampoco se observan irregularidades, teniendo en cuenta que la media observada era de 2324.

Para cada conjunto de imputación, en el paso 2 solicitamos el modelo de regresión para estimar *sales_price*:

```
proc surveyreg data=ap7_imp;
  by _imputation_;
  class style;
  model sales_price=finished_square_feet bedrooms bathrooms air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway / solution;
  ods output parameterestimates=ap7imp_regparms;
run;
```

Imputation number=1

Estimated Regression Coefficients				
Parameter	**Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	-2917632.5	509357.797	-5.73	<.0001
finished_square_feet	111.1	10.492	10.59	<.0001
bedrooms	610.1	4752.624	0.13	0.8980
bathrooms	12443.0	6482.102	1.92	0.0559
air_conditioning	-7988.0	15111.241	-0.53	0.5975
garage_size	509.4	7254.717	0.07	0.9441
pool	20093.5	13012.754	1.54	0.1236
real_built	1420.4	259.219	5.48	<.0001
style 1	75018.3	13062.993	5.74	<.0001
style 2	65176.7	13195.257	4.94	<.0001
style 3	74343.1	13296.307	5.59	<.0001
style 4	32719.4	37971.258	0.86	0.3896
style 5	29494.6	30645.188	0.96	0.3366
style 6	70547.7	24795.801	2.85	0.0048
style 7	47448.9	14822.262	3.20	0.0015
style 11	0.0	0.000	.	.
lot_size	1.8	0.438	4.05	<.0001
adjacent_to_highway	11128.3	13025.395	0.85	0.3936

Tabla 21

Aquí muestro el modelo de regresión para la primera imputación. Los coeficientes de algunas variables como *bedrooms*, *air_conditioning*, *garaje_size*, *pool*, *adjacent_to_highway* son claramente no significativos, por lo que procedemos a eliminarlos progresivamente, quitando en cada paso el menos significativo. La variable *bathrooms*, aunque en la primera imputación no resulta significativa a nivel 0.05, está al límite de serlo y, además, en otras imputaciones sí lo es, por lo que no será suprimida.

```
proc surveyreg data=ap7_imp;
  by _imputation_;
  class style;
  model sales_price=finished_square_feet bathrooms real_built style lot_size /
solution;
  ods output parameterestimates=ap7imp_regparms;
run;
```

Imputation number=1

Estimated Regression Coefficients				
Parameter	**Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	-2825215.3	493668.401	-5.72	<.0001
finished_square_feet	113.0	9.706	11.64	<.0001
bathrooms	13090.3	6290.070	2.08	0.0383
real_built	1368.1	248.888	5.50	<.0001
style 1	77128.2	7647.354	10.09	<.0001
style 2	69231.0	7989.407	8.67	<.0001
style 3	76383.9	7853.425	9.73	<.0001
style 4	31236.1	35719.436	0.87	0.3826
style 5	29007.7	28110.899	1.03	0.3030
style 6	69579.0	23646.656	2.94	0.0035
style 7	48928.9	10730.503	4.56	<.0001
style 11	0.0	0.000	.	.
lot_size	1.8	0.413	4.34	<.0001

Tabla 22

Con el nuevo modelo, se obtienen coeficientes significativos excepto para alguna categoría de la variable *style*. Estas órdenes de SAS han generado un conjunto de datos de salida llamado *ap7imp_regparms*, con los 10 modelos estimados, para el PROC MIANALYZE.

A continuación, muestro unas órdenes para modificar el renombrado que hace SAS a las categorías de la variable *style*, puesto que con PROC MIANALYZE no podemos especificar nombres variables con espacios.

```
data ap7imp_regparms2;
set ap7imp_regparms;
 if Parameter='style 1' then Parameter='style1';
 if Parameter='style 2' then Parameter='style2';
 if Parameter='style 3' then Parameter='style3';
 if Parameter='style 4' then Parameter='style4';
 if Parameter='style 5' then Parameter='style5';
 if Parameter='style 6' then Parameter='style6';
 if Parameter='style 7' then Parameter='style7';
 if Parameter='style 11' then Parameter='style11';
run;
```

Por último, en el paso 3, se utiliza el conjunto de datos anterior para obtener las estimaciones MI.

```
proc mianalyze parms=ap7imp_regparms2;
  modeleffects intercept finished_square_feet bathrooms real_built style1 style2
style3 style4 style5 style6 style7 style11 lot_size ;
run;
```

Parameter Estimates							
Parameter	**Estimate**	**Std Error**	**95% Confidence Limits**		**DF**	**Minimum**	**Maximum**
Intercept	-2796582	641892	-4073864	-1519301	80.513	-3172841	-1922934
finished_square_feet	108.013147	12.306440	84	132	91.963	98.335623	119.051493
Bathrooms	16193	7222.145102	1933	30452	165.93	12585	21064
real_built	1355.452503	322.417847	714	1997	83.014	918.954168	1546.291659
style1	75559	8367.825144	59119	91999	503.93	68735	78623
style2	60306	8971.196172	42446	78165	78.171	53652	69231
style3	73644	9011.023719	55889	91400	227.95	67074	77129
style4	96195	40810	11522	180869	21.828	31236	144402
style5	33189	29479	-24609	90988	3484.9	25504	49059
style6	64809	26805	12018	117599	251.59	44977	82702
style7	42967	11720	19951	65984	613.72	37230	48929
style11	0	0	.	.	.	0	0
lot_size	1.753802	0.418102	1	3	1685.3	1.544467	1.849282

Tabla 23

En la tabla de Parámetros Estimados (tabla 24), la primera columna muestra las estimaciones de los coeficientes de los regresores del modelo para predecir el *precio de la vivienda.*

Vamos ahora a estimar la media de los *precios de las viviendas*, de manera que volvemos al paso 2, en el que usamos el procedimiento estándar PROC SURVEYMEANS para la estimación de la media de la variable de interés para cada conjunto de datos imputados. El paso 1 no es necesario repetirlo, puesto que las imputaciones sí que nos son válidas.

```
proc surveymeans data=ap7_imp;
  by _imputation_;
  var sales_price ;
  ods output Statistics=ap7imp_mean;
run;
```

Imputation	Mean	StdErr	LowerCLMean	UpperCLMean
1	278473,4724	5138,877365	268359,1009	288587,8439
2	276226,4138	4987,620562	266409,7469	286043,0807
3	275826,8069	5036,571182	265913,7951	285739,8187
4	274876,0138	5025,682215	264984,4337	284767,5939
5	276464,2379	4980,305627	266661,9683	286266,5075
6	274739,1172	5024,02279	264850,8033	284627,4312
7	275905,3241	5079,183157	265908,4432	285902,2051
8	276703,1207	5213,129391	266442,6059	286963,6355
9	275284,0241	5010,158846	265422,9973	285145,051
10	275604,4966	5093,1708	265580,085	285628,9081

Tabla 24

Por último, PROC MIANALYZE para combinar los resultados y obtener la estimación MI de la media, que resulta ser 276010 con un intervalo de confianza (265846.9, 286173.7).

```
proc mianalyze data=ap7imp_mean;
  modeleffects mean;
  stderr stderr;
run;
```

Parameter Estimates							
Parameter	**Estimate**	**Std Error**	**95% Confidence Limits**		**DF**	**Minimum**	**Maximum**
mean	276010	5183.910585	265846.9	286173.7	3992.3	274739	278473

Tabla 25

FCS REGPMM

Imputamos ahora con el método FCS emparejamiento predictivo de la media, obteniendo de nuevo 10 conjuntos de datos imputados:

```
proc mi data=apendice7 nimpute=10 seed=400 out=ap7_imp_regpmm
  min= 150000 1384 . . . . . .
  max= 675000 4150  . . . . . .
  round=  1 1 . . . . . .;
  where quality=2;
  var  sales_price  finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway;
  class style; *las variables binarias no es necesario incluirlas en class;
  fcs regpmm (sales_price /details) regpmm (finished_square_feet);
run;
```

Aquí también se ha comprobado que las medias de los datos imputados para las variables *sales_price* y *finished_square_feet* no difieran mucho del valor medio observado, aunque no muestro las tablas.

```
proc surveyreg data=ap7_imp_regpmm;
  by _imputation_;
  class style  ;
  model  sales_price=finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway / solution;
  ods output parameterestimates=ap7imp_regpmmparms;
run;
```

Imputation number=1

Estimated Regression Coefficients				
Parameter	**Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	-2865118.2	543862.757	-5.27	<.0001
finished_square_feet	113.1	10.630	10.64	<.0001
Bedrooms	3361.0	4403.686	0.76	0.4460
Bathrooms	11843.5	6560.658	1.81	0.0721
air_conditioning	-4049.3	14616.182	-0.28	0.7819
garage_size	837.9	7237.904	0.12	0.9079
Pool	21920.1	11445.063	1.92	0.0564
real_built	1388.2	276.670	5.02	<.0001
style 1	68071.8	12293.010	5.54	<.0001
style 2	39775.0	12169.564	3.27	0.0012
style 3	67206.4	12973.373	5.18	<.0001
style 4	110773.7	38145.076	2.90	0.0040
style 5	7750.0	33590.009	0.23	0.8177
style 6	64887.3	25133.854	2.58	0.0103
style 7	31940.3	14020.648	2.28	0.0235
style 11	0.0	0.000	.	.
lot_size	1.8	0.408	4.51	<.0001
adjacent_to_highway	20047.8	13716.157	1.46	0.1449

Tabla 26

Al hacer la regresión para la primera imputación, resultan no significativas las mismas variables que cuando empleábamos el método de imputación FCS REGRESSION. Las vamos eliminando una a una para obtener el nuevo modelo. La variable *pool* resulta significativa en algunas

de las imputaciones, pero optamos por eliminarla del modelo para así hacer coincidir los regresores con del modelo anterior y comparar los estimadores de los coeficientes.

```
proc surveyreg data=ap7_imp_regpmm;
  by _imputation_;
  class style  ;
  model sales_price=finished_square_feet bathrooms real_built style lot_size /
solution;
  ods output parameterestimates=ap7imp_regpmmparms2;
run;
```

Imputation number=1

Estimated Regression Coefficients				
Parameter	**Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	-2779216.9	516939.964	-5.38	<.0001
finished_square_feet	116.5	9.953	11.70	<.0001
bathrooms	13563.2	6218.191	2.18	0.0300
real_built	1340.4	261.077	5.13	<.0001
style 1	76264.2	7627.741	10.00	<.0001
style 2	50279.9	7149.237	7.03	<.0001
style 3	74844.7	8089.410	9.25	<.0001
style 4	114596.8	38664.939	2.96	0.0033
style 5	13043.5	31162.269	0.42	0.6758
style 6	68431.0	23574.397	2.90	0.0040
style 7	38488.1	11140.646	3.45	0.0006
style 11	0.0	0.000	.	.
lot_size	1.8	0.396	4.67	<.0001

Tabla 27

Ya tenemos el modelo con todas las variables significativas, excepto algún nivel de la variable *style* como ocurría en el caso anterior.

```
data ap7imp_regpmmparms2;
set ap7imp_regpmmparms;
 if Parameter='style 1' then Parameter='style1';
 if Parameter='style 2' then Parameter='style2';
 if Parameter='style 3' then Parameter='style3';
 if Parameter='style 4' then Parameter='style4';
 if Parameter='style 5' then Parameter='style5';
 if Parameter='style 6' then Parameter='style6';
 if Parameter='style 7' then Parameter='style7';
 if Parameter='style 11' then Parameter='style11';
run;
```

En el paso 3 hacemos la síntesis de los resultados de las diferentes imputaciones.

```
proc mianalyze parms=ap7imp_regpmmparms2;
  modeleffects intercept finished_square_feet bathrooms pool real_built style1
style2 style3 style4 style5 style6 style7 style11 lot_size;
run;
```

Parameter Estimates							
Parameter	**Estimate**	**Std Error**	**95% Confidence Limits**		**DF**	**Minimum**	**Maximum**
intercept	-3057500	639107	-4319467	-1795532	163.51	-3721592	-2664815
finished_square_feet	108.835030	12.028283	85	133	148.05	97.098549	115.127861
bathrooms	11528	7075.048548	-2390	25445	332.88	8338.054017	16185
real_built	1487.835291	323.690061	849	2127	174.21	1289.334001	1814.798112
style1	66685	13009	41133	92237	568.81	56849	71335
style2	46522	13037	20862	72182	288.04	39775	55454
style3	65240	13393	38947	91533	748.61	56360	70026
style4	102258	42656	15345	189171	31.753	46794	138816
style5	22087	34004	-44687	88862	630.31	6236.671726	34860
style6	61942	27136	8618	115267	467.55	43921	75245
style7	35134	15231	5146	65122	268.98	21696	43107
style11	0	0	.	.	.	0	0
lot_size	1.880119	0.437964	1	3	700.94	1.644252	2.168390

Tabla 28

La columna ‘estimate’ da los coeficientes de los regresores del modelo MI.

Vamos ahora a estimar la media de los *precios de las viviendas* para las 10 imputaciones:

```
proc surveymeans data=ap7_imp_regpmm;
  by _imputation_;
  var sales_price ;
  ods output Statistics=ap7imp_mean_regpmm;
run;
```

Imputation	Mean	StdErr	LowerCLMean	UpperCLMean
1	275415,7655	5183,627042	265213,3174	285618,2136
2	277358,6621	5107,675261	267305,7027	287411,6214
3	273834,4621	5098,510987	263799,5399	283869,3842
4	275681,5586	5014,059401	265812,8546	285550,2626
5	276228,7655	5106,892992	266177,3459	286280,1852
6	274442,0759	5162,950221	264280,324	284603,8277
7	277470,1862	5415,028384	266812,2925	288128,0799
8	277113,531	5332,667352	266617,7408	287609,3212
9	273542,7379	5065,349259	263573,0849	283512,3909
10	275437,7655	5038,298013	265521,3549	285354,1761

Tabla 29

```
proc mianalyze data=ap7imp_mean_regpmm;
  modeleffects mean;
  stderr stderr;
run;
```

Parameter Estimates							
Parameter	**Estimate**	**Std Error**	**95% Confidence Limits**		**DF**	**Minimum**	**Maximum**
Mean	275653	5363.207543	265132.6	286172.5	1537.4	273543	277470

Tabla 30

Por último, PROC MIANALYZE se emplea para la estimación MI de la media, que resulta ser 275653 con un intervalo de confianza (265132.6, 286172.5).

Comparación de resultados

A continuación, se muestran los modelos estimados con cada uno de los métodos de imputación:

	REGRESSION	REGPMM
intercept	-3057500	-2796582
finished_square_feet	108.835030	108.013147
bathrooms	11528	16193
real_built	1487.835291	1355.452503
style1	66685	75559
style2	46522	60306
style3	65240	73644
style4	102258	96195
style5	22087	33189
style6	61942	64809
style7	35134	42967
style11	0	0
lot_size	1.880119	1.753802

Tabla 31

A simple vista, existen pequeñas diferencias, pero lo vemos más claramente con un ejemplo. Supongamos que queremos estimar el precio de una vivienda de:

- 2000 pies cuadrados de área construida
- 4 baños
- construida en 1910
- estilo 5
- 20000 pies cuadrados de solar

El modelo obtenido con los datos imputados con el método de regresión daría la siguiente estimación:

Sales_price=-3057500+108.835*2000+11528*4+1487.835*1910+22087+1.88*20000= 107733.9 dólares

mientras que la estimación MI cuando se empleó imputación con el método de emparejamiento predictivo de la media es:

Sales_price= -2796582+108.013*2000+16193*4+1355.453*1910+33189+1.75*20000= 141320.2 dólares

Hay una diferencia de más de 33000 dólares.

Por último, se comparan las medias MI de los *precios de las viviendas*:

Método de imputación	**Media *sales_price***	**Error estándar**	**Intervalo Confianza 95%**		**GL**	**Mínimo**	**Máximo**
REGRESSION	276010	5183.910585	265846.9	286173.7	3992.3	274739	278473
REGPMM	275653	5363.207543	265132.6	286172.5	1537.4	273543	277470

Tabla 32

Las medias no difieren demasiado. Nos fijamos también en los errores estándar donde resulta menor el obtenido con el método FCS REGRESSION.

8. Ejemplo 3: imputación múltiple de variables categóricas con patrón arbitrario y mixtura de covariables

(Datos extraídos del apéndice 10 del libro "Applied Linear Statistical Models")

Se tiene un conjunto de 196 personas seleccionadas de dos sectores de una ciudad y de las que se toma la siguiente información:

Id: identificador del individuo

Age: edad de la persona en años, continua y sin missing

Sector: sector dentro de la ciudad, 1-sector 1, 2-sector 2, categórica sin missing

Savings_account_status: 1-tiene cuenta de ahorro, 0-no tiene cuenta de ahorro, categórica sin missing

Socioeconomic_status: 1-alto, 2-medio, 3-bajo, categórica con missing

Disease_status: 1-con enfermedad, 0-sin enfermedad, categórica con missing

El objetivo es modelar la probabilidad de tener enfermedad como función del resto de variables.

```
data apendice10;
  input Id Age Sector Savings_account_stat Socioeconomic_status Disease_status;
  cards;
1 33 1 1 1 0
2 35 1 1 1 0
3 6 1 0 1 0
…
196 24 1 0 NA 0
; run;
```

De acuerdo con los tres pasos MI, lo primero que vamos a hacer va a ser examinar el patrón de missing, elegir un modelo de imputación y realizar las imputaciones:

```
proc mi data=apendice10 nimpute=0;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
run;
```

Missing Data Patterns							
Group	**Age**	**Sector**	**Savings_account_stat**	**Socioeconomic_status**	**Disease_status**	**Freq**	**Percent**
1	X	X	X	X	X	154	78.57
2	X	X	X	X	.	15	7.65
3	X	X	X	.	X	20	10.20
4	X	X	X	.	.	7	3.57

Tabla 33

En la tabla 34 se da el patrón de missing. Se trata de un patrón arbitrario con missing en las variables *Socioeconomic_status* y *Disease_status*, ambas categóricas. Existen cuatro grupos de datos:

- ✓ 154 personas de las que se conocen todas las variables (78.57%)
- ✓ 15 personas de las que no se sabe si están enfermas (7.65%)
- ✓ 20 personas de las que no se conoce el estatus socioeconómico (10.2%)
- ✓ 7 personas de las que no se sabe ni si están enfermas ni el estatus socioeconómico (3.57%)

Se va a emplear un procedimiento multipaso, que primero usa MCMC MONOTONE, imputando solo algunos missing de manera que el nuevo patrón sea monótono, y luego MONOTONE LOGISTIC para completar el proceso de imputación.

En estos datos, si se imputan los 20 valores de la variable *Socioeconomic_status* se consigue un patrón monótono.

Imputamos con MCMC MONOTONE para conseguir patrón monótono:

```
proc mi data=apendice10 nimpute=10 seed=20 out=ap10_imp_1ststep
      round= . . . 1 1
      min=   . . . 1 0
      max=   . . . 3 1;
  mcmc impute=monotone;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
run;
```

Missing Data Patterns							
Group	**Age**	**Sector**	**Savings_account_stat**	**Socioeconomic_status**	**Disease_status**	**Freq**	**Percent**
1	X	X	X	X	X	154	78.57
2	X	X	X	X	O	15	7.65
3	X	X	X	.	X	20	10.20
4	X	X	X	O	O	7	3.57

Tabla 34

En la tabla 34 se muestra un patrón en el que las 'X' indican datos observados, los '.' significan que los missing fueron imputados por el método MCMC, y las 'O' representan missing no imputado. De esta manera, hay tres grupos de datos:

- ✓ personas de las que se conocen todas las variables
- ✓ personas de las que no se conoce si están enfermas
- ✓ personas de las que no se sabe ni si están enfermas ni el estatus socioeconómico

Se ve más claramente a continuación:

```
proc mi data=ap10_imp_1ststep nimpute=0;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
run;
```

Missing Data Patterns							
Group	**Age**	**Sector**	**Savings_account_stat**	**Socioeconomic_status**	**Disease_status**	**Freq**	**Percent**
1	X	X	X	X	X	1740	88.78
2	X	X	X	X	.	150	7.65
3	X	X	X	.	.	70	3.57

Tabla 35

En la tabla 35 se tiene el nuevo patrón. La frecuencia se ha multiplicado por 10, puesto que el conjunto de datos que toma como entrada es el que contiene las 10 imputaciones del método MCMC.

Imputamos el resto de missing con LOGISTIC, pero ahora debemos utilizar una sola imputación, porque el conjunto de datos de entrada es el que contiene las 10 imputaciones anteriores.

```
proc mi data=ap10_imp_1ststep nimpute=1 seed=2013 out=ap10_imp_2ndstep;
  by _imputation_;
  class Sector Savings_account_stat Socioeconomic_status Disease_status;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
  monotone logistic (Socioeconomic_status) logistic (Disease_status);
run;
```

En el paso 2, se establece un formato para la variable *Socioeconomic_status* y se utiliza el procedimiento estándar PROC SURVEYLOGISTIC para modelar la probabilidad de tener enfermedad como función del resto de variables mediante la regresión logística.

```
proc format;
  value seef 1='1=Alto' 2='2=Medio' 3='3=Bajo';
run;

proc surveylogistic data=ap10_imp_2ndstep;
  by _imputation_;
  class Sector Socioeconomic_status / param=reference;
  model Disease_status (event='1') = Age Sector Savings_account_stat
Socioeconomic_status;
  format Socioeconomic_status seef.;
  ods output parameterestimates=ap10_2steps_est;
run;
```

Imputation number=1

Variable	ClassVal0	Estimate	StdErr	WaldChiSq	ProbChiSq	tValue	ProbT
Intercept		-0,90188	0,462712	3,7991098	0,0512798	-1,9491	052714
Age		0,033702	0,008839	14,53584	0,0001375	3,8125	0,00018
Sector	1	-1,06084	0,361126	8,6295851	0,0033074	-2,9376	0,00370
Savings account stat		195	0,406615	0,5163193	0,4724159	-0,7185	0,47327
Socioeconomic status	1=Alto	-0,07287	0,453527	0,0258208	0,8723385	-0,1606	0,87250
Socioeconomic status	2=Medio	-0,26947	0,446931	0,3635315	0,5465514	-0,6029	0,54725

Tabla 36

En el fichero que se especifica en la sentencia ODS OUTPUT se almacenan los parámetros estimados de los modelos de regresión de los 10 conjuntos de datos imputados (en la tabla 36 se muestra solo el modelo para la primera imputación). Estos datos serán la entrada de PROC MIANALYZE para obtener el modelo final.

```
proc mianalyze parms(classvar=classval) = ap10_2steps_est;
  class Sector Socioeconomic_status;
  modeleffects intercept Age Sector Savings_account_stat Socioeconomic_status;
run;
```

Parameter Estimates						
Parameter	**Sector**	**Socioeconomic_status**	**Estimate**	**Std Error**	**95% Confidence Limits**	
Intercept			-0.821582	0.528866	-1.86627	0.04765
Age			0.027908	0.009988	0.00817	-0.42800
Sector	1		-1.214334	0.399646	-2.00067	0.58936
Savings_account_stat			-0.212931	0.409131	-1.01522	0.85786
Socioeconomic_status		1=Alto	-0.124568	0.498317	-1.10700	0.86973
Socioeconomic_status		2=Medio	-0.111398	0.498390	-1.09252	0.86973

Tabla 37

El modelo logístico sería:

$$\text{Logit}(P(\textit{Disease\_estatus}=1)) = -0.821582 + 0.027908*\textit{Age} - 1.214334*\text{Sector1} - 0.212931*\textit{Saving\_account\_stat1} - 0.124568*\textit{Socioeconomic\_status1} - 0.111398*\textit{Socioeconomic\_status2}$$

Pero algunos regresores como *Saving_account_stat* y *Socioeconomic_status* resultaban ser no significativos en la tabla 36, por lo que eliminamos primero *Socioeconomic_status* por ser menos significativa y observamos que en el nuevo modelo la variable *Saving_account_stat* también es no significativa (aunque en este trabajo no se expone). De igual modo la eliminamos y volvemos de nuevo al paso 2:

```
proc surveylogistic data=ap10_imp_2ndstep;
  by _imputation_;
  class Sector Socioeconomic_status / param=reference;
  model Disease_status (event='1')=Age Sector  ;
  format Socioeconomic_status seef.;
  ods output parameterestimates=ap10_2steps_est2;
run;
```

Imputation number=1

Variable	ClassVal0	Estimate	StdErr	WaldChiSq	ProbChiSq	tValue	ProbT
Intercept		-1,172590177	0,3253110	12,992567	0,00031273	-3,60452	0,0003970
Age		0,031780279	0,0081049	15,374941	8,81496E-05	3,9210893	0,0001220
Sector	1	-0,958665494	0,3373748	8,074373169	0,004489563	-2,8415441	0,0049667

Tabla 38

Ahora sí que todos los coeficientes resultan significativos. Estimamos el nuevo modelo MI con PROC MIANALYZE:

```
proc mianalyze parms(classvar=classval)=ap10_2steps_est2;
  class Sector Socioeconomic_status;
  modeleffects intercept Age Sector;
run;
```

Parameter Estimates								
Parameter	**Sector**	**Estimate**	**Std Error**	**95% Confidence Limits**		**DF**	**Minimum**	**Maximum**
intercept		-1.018088	0.349901	-1.70653	-0.32965	315.15	-1.199560	-0.772058
Age		0.026278	0.009224	0.00803	0.04453	126.4	0.017546	0.033344
Sector	1	-1.130452	0.371795	-1.86154	-0.39937	371.77	-1.356442	-0.956230

Tabla 39

De la tabla 39 se deduce que el modelo final es:

Logit(P(*Disease_estatus*=1)) = -1.018088+ 0.026278**Age*-1.130452*Sector1

9. Conclusiones

La imputación múltiple no es un simple procedimiento para imputar missing a variables incompletas. Sigue tres pasos claros en los que la imputación en sí solo se realiza en el primero de ellos, imputando el mismo conjunto de datos repetidas veces, para después, en el segundo paso realizar las estimaciones propuestas como objetivo a cada conjunto por separado. En el último paso, se extraen las estimaciones MI finales, promediando las de las repeticiones anteriores.

Se ha visto que existen métodos MI específicos para cada caso. Por ejemplo, para imputar variables continuas se emplean regresión lineal o el método de emparejamiento predictivo de la media en caso de tener patrón monótono de missing. Si el patrón es arbitrario, habría que diferenciar si hay covariables continuas o mixtas, donde se emplearían algoritmos MCMC y FCS respectivamente. Para variables binarias u ordinales es adecuada la regresión logística, y para imputar valores nominales la opción correcta es la función discriminante.

Existen diferentes formas de realizar la imputación de forma múltiple a datos complejos de encuestas por muestreo: incorporando las variables de diseño y muestreo en el modelo de imputación, realizar el análisis MI ignorando estas variables, y realizar el análisis MI incluyendo las variables de estrato y cluster en el modelo de imputación, y los pesos en la sentencia FREQ de PROC MI. Otros estudios han demostrados que lo habitual es que los resultados sean similares, pero es recomendable el uso de estas variables de diseño.

En los ejemplos desarrollados cabe destacar el segundo, debido a la diferencia en los modelos estimados cuando las imputaciones se realizaban con dos métodos diferentes: regresión lineal y el método de emparejamiento predictivo de la media. En dicho ejemplo, se desea predecir el precio de una vivienda con determinadas características, y este precio resulta muy diferente de un método a otro. En caso de inclinarnos por alguno de los dos métodos de imputación, lo haríamos por el de regresión porque el error estándar al estimar la media resultaba menor. Fijándonos únicamente en los modelos no podríamos decidir qué método es más correcto.

A lo largo de la realización del trabajo, se han detectado algunos problemas mínimos con el software SAS, como se pudo ver en el ejemplo 2, donde hubo que renombrar la variable categórica *style* del conjunto de datos de la salida del PROC SURVEYREG para introducirlos en PROC MIANALYZE.

Futuras ampliaciones

En el presente trabajo se ha visto el proceso de imputación múltiple en problemas en los que las variables a imputar eran todas continuas o todas categóricas. Por ello, este trabajo se podría extender en el estudio del mecanismo de imputación múltiple en problemas que contengan missing en variables de diferentes tipos, en los que, por ejemplo, sean algunas categóricas y otras continuas.

Además, se podrían estudiar más en profundidad otros métodos de imputación que se mencionan en este trabajo, pero no se han empleado, como el método de función discriminante para imputar variables de tipo nominal con patrón de missing monótono, o propensity score para rellenar valores ausentes en variables continuas también con patrón monótono.

Bibliografía y referencias

Bibliografía

- ✓ Patricia Berglund and Steven Heeringa. 2014. Multiple Imputation of Missing Data Using SAS
- ✓ https://support.sas.com/documentation/
- ✓ M H. Kutner, C.J. Nachtsheim, J. Neter, W. Li. 2005. Applied Linear Statistical Models

Referencias

(1) Francisco M. Ocaña Peinado. Técnicas estadísticas en Nutrición y Salud. Tratamiento estadístico de outliers y datos faltantes.

(2) Rubin, D.B. 1996. Múltiple imputation after 18+ years (whith discussion). *Journal of the American Statistical Association*

Anexos

Anexo 1

```
data apendice4;
  input id GPA High_school_class_rank ACT_score Academic_year ;
  cards;
1 0.98 61 NA 1996
2 1.13 84 20 1996
3 1.25 74 19 1996
…
705 4 99 32 2000
; run;

*Paso 1;
proc mi data=apendice4 nimpute=0 simple;
  var GPA High_school_class_rank ACT_score Academic_year;
run;

proc mi data=apendice4 out=outap4 seed=192 nimpute=5
      min= 0.51 . 13
      max= 4 . 35
      round = 0.01 . 1;
  var GPA High_school_class_rank ACT_score;
  mcmc
  plots=(trace(mean(GPA) mean(ACT_score))
      acf(mean(GPA) mean(ACT_score)));
run;

proc means data=outap4 nonobs;
  class _imputation_;
  var GPA;
run;

proc means data=outap4 nonobs;
  class _imputation_;
  var ACT_score;
run;

*Paso 2;
proc reg data=outap4 outest=out_est_ap4 covout;
  model GPA = High_school_class_rank ACT_score;
  by _imputation_;
run;

*Paso 3;
proc mianalyze data=out_est_ap4;
 modeleffects intercept High_school_class_rank ACT_score;
run;
```

Anexo 2

```
data apendice7;
 input id sales_price finished_square_feet bedrooms bathrooms air_conditioning
garage_size pool real_built quality style lot_size adjacent_to_highway;
  cards;
1 360000 3032 4 4 1 2 0 1972 2 1 22221 0
2 340000 2058 4 2 1 2 0 1976 2 1 22912 0
3 250000 1780 4 3 1 2 0 1980 2 1 21345 0
…
522 95500 1184 2 1 0 1 0 1951 3 1 14786 0
; run;

proc means data=apendice7 n nmiss mean min max;
  where quality=2;
  var  sales_price  finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built quality style lot_size adjacent_to_highway;
run;

*Paso 1;
proc mi nimpute=0 data=apendice7;
  where quality=2;
  var  sales_price  finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built quality style lot_size adjacent_to_highway;
run;

proc mi data=apendice7 nimpute=10 seed=400 out=ap7_imp
  min= 150000 1384 . . . . . .
  max= 675000 4150  . . . . . .
  round= 1 1 . . . . . .;
  where quality=2;
  var  sales_price  finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway;
  class style; *las variables binarias no es necesario incluirlas en class;
  fcs     nbiter=10     regression     (sales_price     /details)     regression
(finished_square_feet);
run;

proc means data=ap7_imp;
  class _imputation_;
  var sales_price;
run;

proc means data=ap7_imp;
  class _imputation_;
  var finished_square_feet;
run;

*Paso 2;
proc surveyreg data=ap7_imp;
  by _imputation_;
  class style;
  model  sales_price=finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway / solution;
  ods output parameterestimates=ap7imp_regparms;
run;
```

```
proc surveyreg data=ap7_imp;
  by _imputation_;
  class style;
  model sales_price=finished_square_feet bathrooms real_built style lot_size /
solution;
  ods output parameterestimates=ap7imp_regparms;
run;

data ap7imp_regparms2;
set ap7imp_regparms;
 if Parameter='style 1' then Parameter='style1';
 if Parameter='style 2' then Parameter='style2';
 if Parameter='style 3' then Parameter='style3';
 if Parameter='style 4' then Parameter='style4';
 if Parameter='style 5' then Parameter='style5';
 if Parameter='style 6' then Parameter='style6';
 if Parameter='style 7' then Parameter='style7';
 if Parameter='style 11' then Parameter='style11';
run;

*Paso 3;
proc mianalyze parms=ap7imp_regparms2;
  modeleffects intercept finished_square_feet bathrooms real_built style1 style2
style3 style4 style5 style6 style7 style11 lot_size ;
run;

*Paso 2;
proc surveymeans data=ap7_imp;
  by _imputation_;
  var sales_price ;
  ods output Statistics=ap7imp_mean;
run;

*Paso 3;
proc mianalyze data=ap7imp_mean;
  modeleffects mean;
  stderr stderr;
run;

*Paso 1;
proc mi data=apendice7 nimpute=10 seed=400 out=ap7_imp_regpmm
  min= 150000 1384 . . . . . .
  max= 675000 4150  . . . . . .
  round=  1 1 . . . . . .;
  where quality=2;
  var  sales_price  finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway;
  class style; *las variables binarias no es necesario incluirlas en class;
  fcs regpmm (sales_price /details) regpmm (finished_square_feet);
run;
```

```
*Paso 2;
proc surveyreg data=ap7_imp_regpmm;
  by _imputation_;
  class style  ;
  model  sales_price=finished_square_feet  bedrooms  bathrooms  air_conditioning
garage_size pool real_built style lot_size adjacent_to_highway / solution;
  ods output parameterestimates=ap7imp_regpmmparms;
run;

proc surveyreg data=ap7_imp_regpmm;
  by _imputation_;
  class style  ;
  model sales_price=finished_square_feet bathrooms real_built style lot_size /
solution;
  ods output parameterestimates=ap7imp_regpmmparms2;
run;

data ap7imp_regpmmparms2;
set ap7imp_regpmmparms;
 if Parameter='style 1' then Parameter='style1';
 if Parameter='style 2' then Parameter='style2';
 if Parameter='style 3' then Parameter='style3';
 if Parameter='style 4' then Parameter='style4';
 if Parameter='style 5' then Parameter='style5';
 if Parameter='style 6' then Parameter='style6';
 if Parameter='style 7' then Parameter='style7';
 if Parameter='style 11' then Parameter='style11';
run;

*Paso 3;
proc mianalyze parms=ap7imp_regpmmparms2;
  modeleffects intercept finished_square_feet bathrooms pool real_built style1
style2 style3 style4 style5 style6 style7 style11 lot_size;
run;

*Paso 2;
proc surveymeans data=ap7_imp_regpmm;
  by _imputation_;
  var sales_price ;
  ods output Statistics=ap7imp_mean_regpmm;
run;

*Paso 3;
proc mianalyze data=ap7imp_mean_regpmm;
  modeleffects mean;
  stderr stderr;
run;
```

Anexo 3

```
data apendice10;
  input Id Age Sector Savings_account_stat Socioeconomic_status Disease_status;
  cards;
1 33 1 1 1 0
2 35 1 1 1 0
3 6 1 0 1 0
…
196 24 1 0 NA 0
; run;

*Paso 1;
proc mi data=apendice10 nimpute=0;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
run;

proc mi data=apendice10 nimpute=10 seed=20 out=ap10_imp_1ststep
round= . . . 1 1
min=   . . . 1 0
max=   . . . 3 1;
  mcmc impute=monotone;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
run;

proc mi data=ap10_imp_1ststep nimpute=0;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
run;

proc mi data=ap10_imp_1ststep nimpute=1 seed=2013 out=ap10_imp_2ndstep;
  by _imputation_;
  class Sector Savings_account_stat Socioeconomic_status Disease_status;
  var Age Sector Savings_account_stat Socioeconomic_status Disease_status;
  monotone logistic (Socioeconomic_status) logistic (Disease_status);
run;

proc format;
  value seef 1='1=Alto' 2='2=Medio' 3='3=Bajo';
run;

*Paso 2;
proc surveylogistic data=ap10_imp_2ndstep;
  by _imputation_;
  class Sector Socioeconomic_status / param=reference;
  model  Disease_status  (event='1')  =  Age  Sector  Savings_account_stat
Socioeconomic_status;
  format Socioeconomic_status seef.;
  ods output parameterestimates=ap10_2steps_est;
run;

*Paso 3;
proc mianalyze parms(classvar=classval) = ap10_2steps_est;
  class Sector Socioeconomic_status;
  modeleffects intercept Age Sector Savings_account_stat Socioeconomic_status;
run;
```

```
*Paso 2;
proc surveylogistic data=ap10_imp_2ndstep;
  by _imputation_;
  class Sector Socioeconomic_status / param=reference;
  model Disease_status (event='1')=Age Sector  ;
  format Socioeconomic_status seef.;
  ods output parameterestimates=ap10_2steps_est2;
run;

*Paso 3;
proc mianalyze parms(classvar=classval)=ap10_2steps_est2;
  class Sector Socioeconomic_status;
  modeleffects intercept Age Sector;
run;
```