Task Specific Imputation Method

Park.S

Department of Applied Statistics

1. Introduction

Imputation is necessary because it is difficult to analyze data with missing values. So I learned multivariate imputation by chained equations method to impute missing values and analyzed the results in class. The result analysis was mostly focused on seeing plots of imputed data to check the bias. However, there are many kinds of tasks that I could do with one imputed data. For instance, with consumers' consumption data I would be able to predict the amount of their consumption or whether they are buying or not. I was wondering if one imputed data would be appropriate for all tasks if these various tasks were created. So, I would like to know if a kind of greedy solution is a global solution. I planned this project to solve these questions. I would like to apply several imputation methods which have different algorithms on the missing data and compare their performance when the each imputed data are used in regression and classification tasks.

2. Exploratory Data Analysis

2.1 Data description

I used Boston housing data set, which has a total of 14 variables and 506 records. This data illustrates information about houses in Boston. The variables show how old the house is, how many rooms it has, and how much it costs. Since the Boston housing data has less categorical variability, I randomly converted the *PTRATIO* into a categorical variable, with the top 50% being 1 and the bottom 50% being zero, so that it became a binary variable.

Variable	Description
INDUS	proportion of non-retail business acres per town
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
CRIM	per capita crime rate by town
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
B	1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.

Table 1
Variable description

I conducted a total of two tasks through the data, one is regression that predicts house prices, that is, the value of *MEDV* through linear regression. And the other task is classification that

predicts the category of *MEDV* through Support Vector Machine. To compare those predicting performance, I splitted full data into 70% training set and 30% test set.

2.2 Experiment Design

Since the Boston data set was data without missing value, I had to generate missing values on the data. I planned to satisfy the MAR assumption because in MAR assumption I could impute data depending on the observed values. To decide the missing variables, I checked the correlation table and chose *RM, RAD, LSTAT* which have high correlation with *MEDV*. Also, I set the assumption that missing values occurred on *CRIM, PTRATIO* due to nonresponse of residents, because those information are seneitive part to answer. Therefore, I decided to select a total of 5 variables(*RM, RAD, LSTAT, CRIM, PTRATIO*) as missing variables and make them to have a monotone missing pattern while satisfying MAR assumptions.

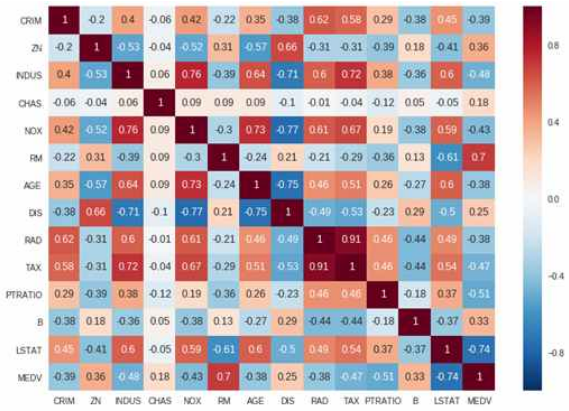


Table 2
Correlation between variables

2.3 Missing process

I made a latent variable Z and used it to generate missing values. Missing indicator (0 or 1) was sampled from *Ber(θ)*.

$$\theta = \frac{1}{1+e^{-z}}, \qquad z = c + \alpha^{\top} X$$

For example, if I generate missing values of *RM,* the latent variable was like below.

$$.5\, X_{TAX} + .6\, X_{AGE} + .5\ X_{DIS} + 1.2\, X_{PTRATIO} - 1.5$$

I applied it to the sigmoid function to obtain the probability of being missed in the *RM.* And I applied the probability to the Bernoulli distribution to generate a missing indicator for *RM.*

$$p(M_{RM}|X) = \frac{1}{1+e^{.5\, X_{TAX} + .6\, X_{AGE} + .5\ X_{DIS} + 1.2\, X_{PTRATIO} - 1.5}}$$

Since I planned to make missing sequentially to let the missing pattern 'monotone', I generated missing only when the previous variable misses.

$$p(M_{LSTAT}|M_{RM}, X) = \frac{1}{1+e^{.5X_{CRIM} + .6X_{ZN} + .25X_{AGE} + .6X_{DIS} + 1.2X_B - 2.5}}\ 1_{M_{RM}=1}$$

I proceeded the monotone pattern, as shown in the left figure, with the missing ratio for each variable in order of *RM, LSTAT, RAD, CRIM,* and *PTRTIO*. The missing rate of each variable was that *RM* had 52.77%, *LSTAT* had 47.23%, *RAD* had 33.00%, *CRIM* had 22.13%, and *PTRATIO* had 16.40% of missing. The overall missing pattern is illustrated in Figure 1.

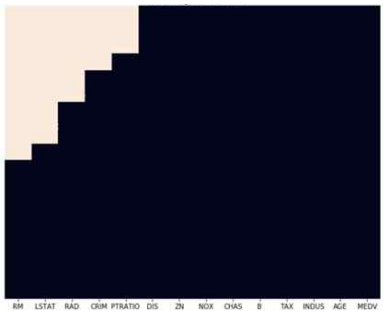


Figure 1
Missing pattern

To determine if the MAR assumption is correct, I drew a histogram of the data before and after missing was generated. Figure 2 shows that the distribution before and after missing was genereates are different. Therefore, I was able to confirm that the missing data was not an MCAR assumption.

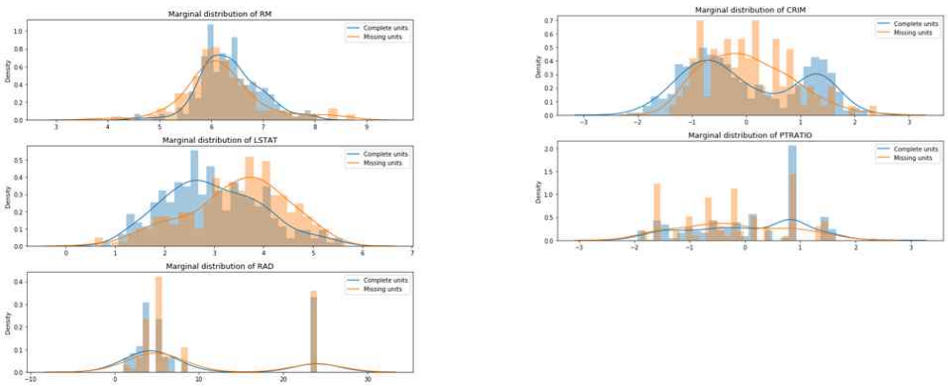
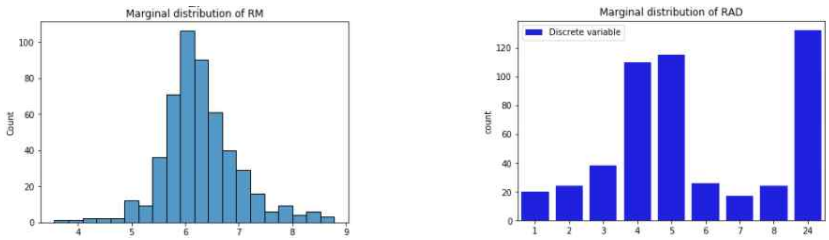


Figure 2
Distribution of Complete units and Missing units

2.4 Variable transformation for Normality

Some multiple imputation R packages such as “Amelia” and “Norm” require normality assumption. To utilize those packages in imputation steps, I drew histograms to see if the missing variables follow normal distribution.



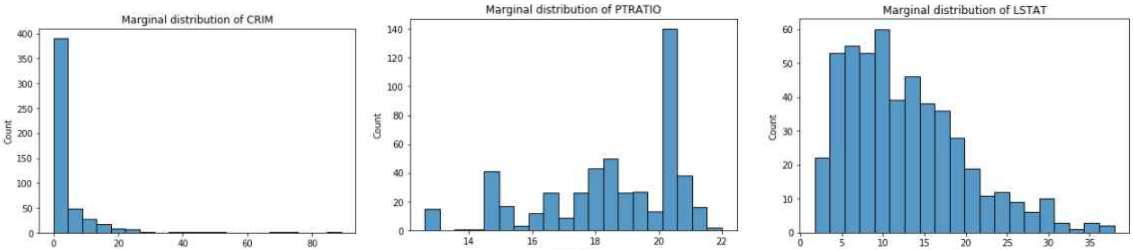


Figure 3
Normality of variables

From Figure 3, only *RM* showed that it follows normal distribution well, so I decided to transform *CRIM, LSTAT, PTRATIO, RAD.* However, *RAD* is categorical variable so I decided to transform without this variable.

Since *CRIM, PTRATIO,* and *LSTAT* are the variables representing the proportions, I first used Logit function to mapp those values to the real number interval. And I transformed the variable by using thwo methods. I proceeded *RM* with Box-Cox transformation because it only had positive values, but *CRIM, PTRATIO,* and *LSTAT* had also negative values, so I proceeded their transformation with Yeo-Johnson transformation. The procedure follows equations below.

$$\psi yj(\lambda, x) = \begin{cases} \dfrac{\{(x+1)^{\lambda} - 1\}}{\lambda} & (x \geq 0,\ \lambda \neq 0) \\ \log(x+1) & (x \geq 0,\ \lambda = 0) \\ -\dfrac{(-x+1)^{2-\lambda} - 1}{2-\lambda} & (x < 0,\ \lambda \neq 2) \\ -\log(-x+1) & (x < 0,\ \lambda = 2) \end{cases}$$

Yeo-Johnson Transformation

$$\psi bc(\lambda, x) = \begin{cases} \dfrac{\left(x^{\lambda} - 1\right)}{\lambda} & (\lambda \neq 0) \\ \log(x) & (\lambda = 0) \end{cases}$$

Box-Cox Transformation

From Figure 4, *RM* and *LSTAT* appeared to follow normal distribution, but *CRIM* and *PTRATIO* do not follow normal distribution yet. I have tried many of these variables to follow normal distribution, but they didn't change much, so I decided to proceed imputation under this condition.

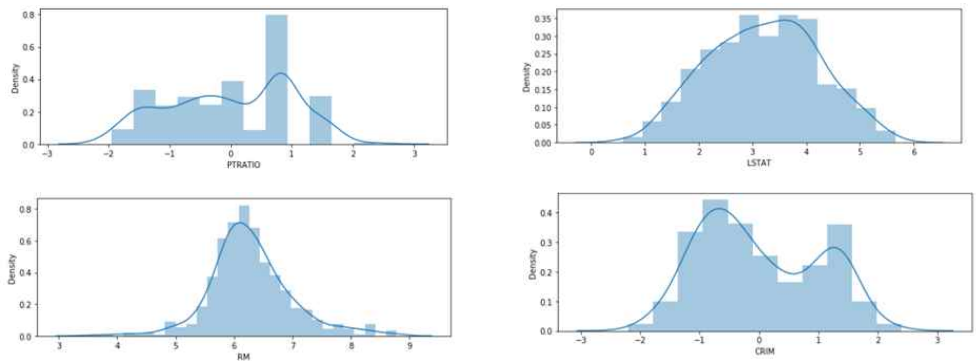


Figure 4
Distribution after transformation

3. Imputation processing

3.1 Imputation methods

I planned to apply multiple imputation methods to compare performance of the methods. I decided to apply EM algorithm, Data augmentation, and imputation by Chained-equations. In particular, in the case of EM algorithm and Data augmentation, I decided to compare performance of two packages, the MLMI package which proceed imputation based on maximum likelihood estimation and the NORM package which assumes multivariate normal distribution.

3.2 Simulation design

I planned a simulation to compare the imputation methods more accurately. I first created 100 missing data sets with the same missing mechanism, which is shown in the section 2. Mean and variance of the missing variables were obtained from each data set, and a linear regression was performed to predict the *MEDV*. And I also applied Support Vector Machine skill to predict the categories of *MEDV*.

3.3 EM algorithm

I first applied the EM algorithm through the AMELIA package in R. The algorithm first creates a bootstrapped version of the original data, estimates the sufficient statistics (with priors if specified) by EM on this bootstrapped sample, and then imputes the missing values of the original data using the estimated sufficient statistics. It repeats this process "m" times to produce the m complete data sets where the observed values are the same and the unobserved values are drawn from their posterior distributions. For each simulation data, "m", that is the number of imputed data sets to create, was determined to be 5 and imputation was performed. Table 3 shows the sample data of imputed results from the EM algorithm with AMELIA.

RM	LSTAT	RAD	CRIM	PTRATIO	DIS	ZN	NOX	CHAS	B	TAX	INDUS	AGE	MEDV
7.408049	1.461482	5	-1.542938	-1.546704	6.0622	0.0	0.4580	0	396.90	222	2.18	54.2	36.2
5.723967	3.530537	3	-1.091328	1.026592	4.2579	0.0	0.5380	0	386.75	307	8.14	81.7	17.5
5.601107	3.714877	5	0.149672	-0.958858	3.9769	0.0	0.5380	0	396.90	307	8.14	91.7	15.2
6.327672	3.501619	7	-0.167309	-1.001809	3.9900	0.0	0.5380	0	232.60	307	8.14	82.0	13.2
5.834947	3.363702	4	-0.341897	-0.829245	3.7872	0.0	0.5380	0	358.77	307	8.14	95.0	13.1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7.820000	1.545878	5	-1.230580	-1.498629	4.6947	20.0	0.4429	0	387.31	216	3.33	64.5	45.4
7.853000	1.563767	4	-1.240922	-1.547015	5.1180	95.0	0.4161	0	392.78	224	2.68	33.2	48.5

Table 3
Imputation with Amelia package

RM	LSTAT	RAD	CRIM	PTRATIO	DIS	ZN	NOX	CHAS	B	TAX	INDUS	AGE	MEDV
6	2.725136	0.844451	-0.163215	-0.297829	2.5194	0.0	0.5440	0	350.45	304	9.90	37.8	16.1
6	2.513492	3.437269	-0.562581	-0.354972	3.2721	0.0	0.5070	0	396.90	307	6.20	80.8	30.1
1	3.824915	12.344065	0.006203	-1.349114	1.6232	0.0	0.8710	0	261.95	403	19.58	98.5	19.4
1	3.181807	14.304439	0.075331	-0.860116	1.5916	0.0	0.8710	0	341.60	403	19.58	100.0	19.6
5	4.866569	10.843899	0.877865	0.973692	2.4259	0.0	0.6050	0	292.29	403	19.58	94.6	17.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7	1.545878	5.000000	-1.230580	-1.498629	4.6947	20.0	0.4429	0	387.31	216	3.33	64.5	45.4
7	1.563767	4.000000	-1.240922	-1.547015	5.1180	95.0	0.4161	0	392.78	224	2.68	33.2	48.5
7	1.235390	4.000000	-1.754162	-1.615990	5.6484	80.0	0.4220	0	394.23	255	0.46	32.0	50.0

Table 4
Imputation with mlmi package

I also applied the EM algorithm with MLMI package in R. This algorithm performs multiple imputation under a general location model as described by Schafer (1997), using the mix package. Imputation can either be performed using conditional on the maximum likelihood estimate of the model parameters, referred to as maximum likelihood multiple imputation by von Hippel (2018). Similarly, for each simulation data, m was determined to be 5 and 20 and imputation was performed. Table 4 shows the sample data of imputed results from the EM algorithm with MLMI.

As a result, Figure 5 shows the imputed result using EM algorithm with AMELIA package for one simulation data. The blue histogram is the original data, and the red histogram which is imputed data with AMELIA shows the similar distribution with the blue one.

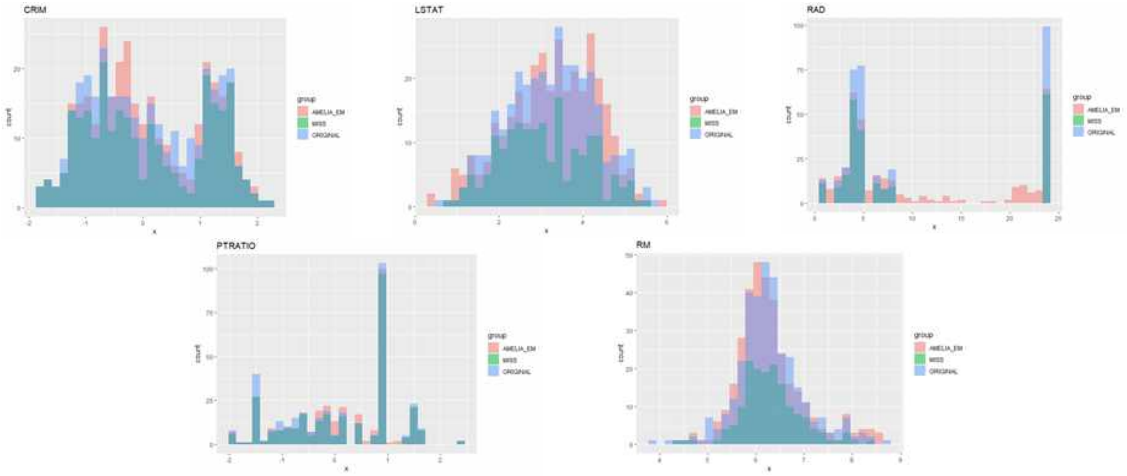


Figure 5
Simulation results from EM algorithm with Amelia

Also Figure 6 shows the imputed result using EM algorithm with MLMI package for the same one simulation data. The blue histogram is the actual data, and the green one is the data imputed with MLMI. And I drew a plot with a count, which looks different, but when I think about density, I can see that it has similar distribution. But in the case of *RM*, it is concentrated in a particular section. So I could see that the distribution is different for *RM*.

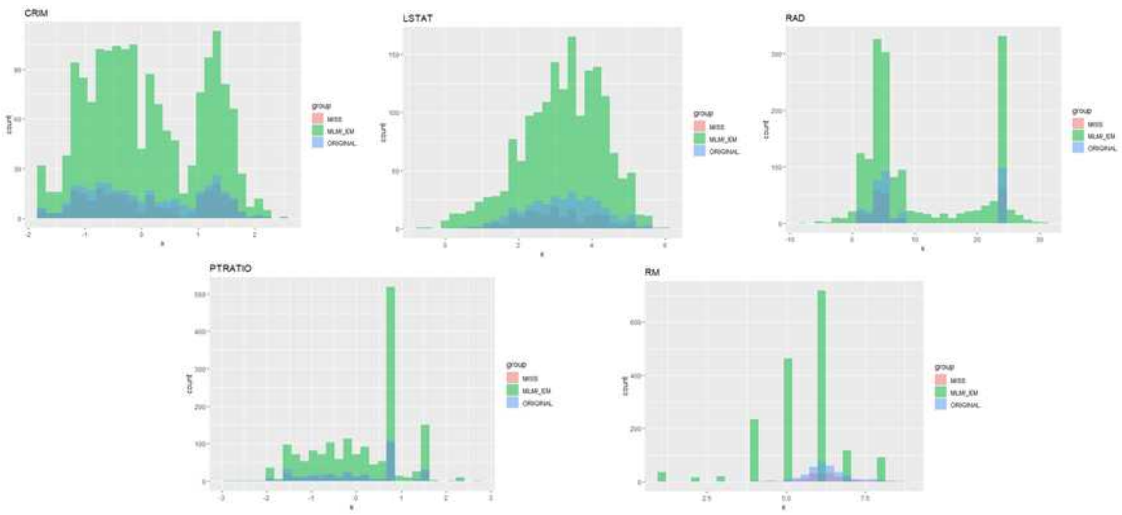


Figure 6
Simulation results from EM algorithm with mlmi

3.4 Data augmentation

I applied data augmentation through the NORM package in R. Data augmentation was performed under a normal-inverted Wishart prior. The usual noninformative prior for the multivariate normal distribution was used as a prior. NORM package simulates several iterations of a single Markov chain. Each iteration consists of a random imputation of the missing data given the observed data and the current parameter value (I-step), followed by a draw from the posterior distribution of the parameter given the observed data and the imputed data (P-step). For each simulation data, m was determined to be 1, 5, 10, and iteration number was determined to be 10 and 100 for each simulation data set. I also put Maximum likelihood estimator as a initial value. Table 5 shows the sample data of imputed results from the data augmentation. Table 5 shows the sample

data of imputed results from data augmentation with NORM package.

RM	LSTAT	RAD	CRIM	PTRATIO	DIS	ZN	NOX	CHAS	B	TAX	INDUS	AGE	MEDV
5.221337	3.768884	1	-0.689818	0.282515	2.5194	0.0	0.544	0	350.45	304	9.90	37.8	16.1
7.371190	2.005999	6	-0.154704	-0.167915	3.2721	0.0	0.507	0	396.90	307	6.20	80.8	30.1
6.041522	3.388254	16	1.540843	1.093526	1.6232	0.0	0.871	0	261.95	403	19.58	98.5	19.4
5.936874	3.774250	10	0.304944	0.071641	1.5916	0.0	0.871	0	341.60	403	19.58	100.0	19.6
6.067186	3.913312	8	0.599777	1.326307	2.4259	0.0	0.605	0	292.29	403	19.58	94.6	17.4

Table 5
Imputation with norm package

RM	LSTAT	RAD	CRIM	PTRATIO	DIS	ZN	NOX	CHAS	B	TAX	INDUS	AGE	MEDV
6	2.541324	1.432867	-0.783756	0.494788	2.5194	0.0	0.5440	0	350.45	304	9.90	37.8	16.1
6	3.004674	9.288626	-0.097815	-0.610609	3.2721	0.0	0.5070	0	396.90	307	6.20	80.8	30.1
1	1.421242	12.245786	0.930376	-1.686010	1.6232	0.0	0.8710	0	261.95	403	19.58	98.5	19.4
1	0.167185	16.159849	0.277741	-1.119867	1.5916	0.0	0.8710	0	341.60	403	19.58	100.0	19.6
6	3.519021	1.912360	-0.861445	-1.599383	2.4259	0.0	0.6050	0	292.29	403	19.58	94.6	17.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Table 6
Imputation with mlmi package

I also applied the data augmentation with MLMI package in R. Imputation was performed using posterior draws. Similarly, for each simulation data, m was determined to be 5 and 20 and input was performed. Table 6 shows the sample data of imputed results from data augmentation with MLMI package.

As a result, Figure 7 shows the imputed result using data augmentation with NORM package for same one simulation data. The blue histogram is the original data, and the green histogram which is imputed data with NORM package. Likewise, I draw plots with the count and they look different, but when I think about density, I can see that they have a similar distribution.

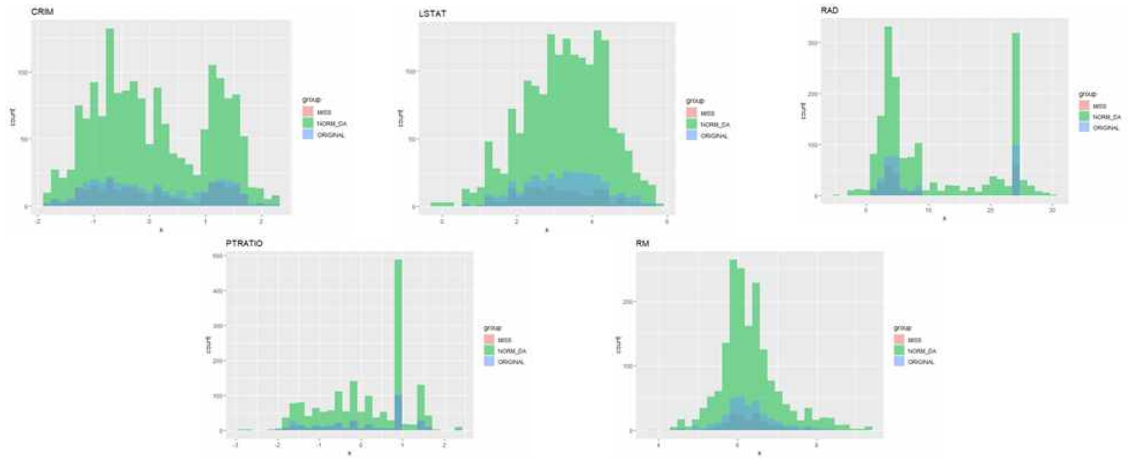
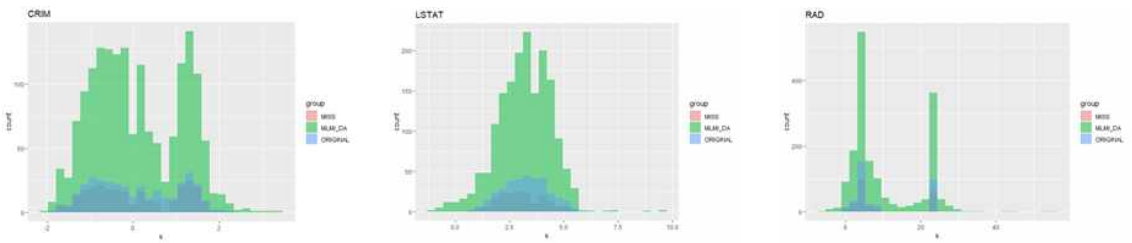


Figure 7
Simulation results from data augmentation with norm

Also Figure 8 shows the imputed result using data augmentation with MLMI package for the same one simulation data. The blue histogram is the actual data, and the green one is the data imputed with MLMI. I could see that *RM* is concentrated in a specific interval, the same result as the previous imputation of EM algorithm in the MLMI package. So I could see that the distribution is different for the *RM*.



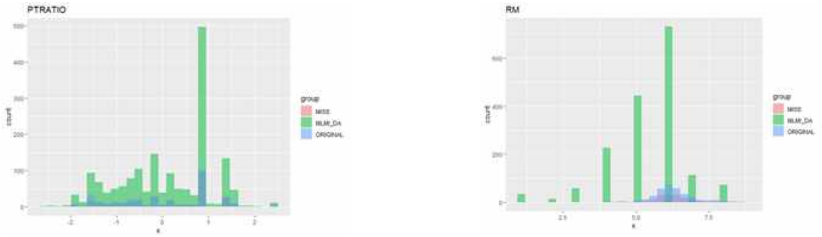


Figure 8
Simulation results from data augmentation with mlmi

3.5 Multivariate imputation by chained equations

I applied multivariate imputation by chained equations method with MICE package. This package creates multiple imputations for multivariate missing data. The method is based on Fully Conditional Specification, where each incomplete variable is imputed by a separate model. MICE package can impute continuous two-level data, and maintain consistency between mice 73 imputations by means of passive imputation. I put m=5 and implemented Polytomous logistic regression method for the categorical variable *RAD*, and predictive mean matching method for the remaining continuous variables. Table 7 shows the sample data of imputed results from MICE package.

X1.RM	X1.LSTAT	X1.RAD	X1.CRIM	X1.PTRATIO	X1.DIS	X1.ZN	X1.NOX	X1.CHAS	X1.B	X1.TAX	X1.INDUS	X1.AGE	X1.MEDV
7.420	2.279159	3	-0.486092	-0.142361	6.0622	0.0	0.4580	0	396.90	222	2.18	54.2	36.2
5.836	3.674847	5	-0.325437	-0.244016	4.2579	0.0	0.5380	0	386.75	307	8.14	81.7	17.5
5.968	2.995029	6	-0.360853	-0.481156	3.9769	0.0	0.5380	0	396.90	307	8.14	91.7	15.2
6.162	3.288434	5	-0.502433	0.187986	3.9900	0.0	0.5380	0	232.60	307	8.14	82.0	13.2
5.987	4.340107	5	-0.480534	0.187986	3.7872	0.0	0.5380	0	358.77	307	8.14	95.0	13.1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7.820	1.545878	5	-1.230580	-1.498629	4.6947	20.0	0.4429	0	387.31	216	3.33	64.5	45.4
7.853	1.563767	4	-1.240922	-1.547015	5.1180	95.0	0.4161	0	392.78	224	2.68	33.2	48.5
7.875	1.235390	4	-1.754162	-1.615990	5.6484	80.0	0.4220	0	394.23	255	0.46	32.0	50.0
7.923	1.315413	1	-1.707497	-1.780461	5.8850	90.0	0.4010	1	395.52	198	1.21	24.8	50.0
8.034	1.196098	4	-1.545492	-1.547015	5.1180	95.0	0.4161	0	390.55	224	2.68	31.9	50.0

Table 7
Imputation with mice package

As a result, Figure 9 shows the imputed result of multivariate imputation by chained equations with MICE package for same one simulation data. The blue histogram is the actual data, and the red one is the imputed data with MICE, and both showed similar distribution in the plots.

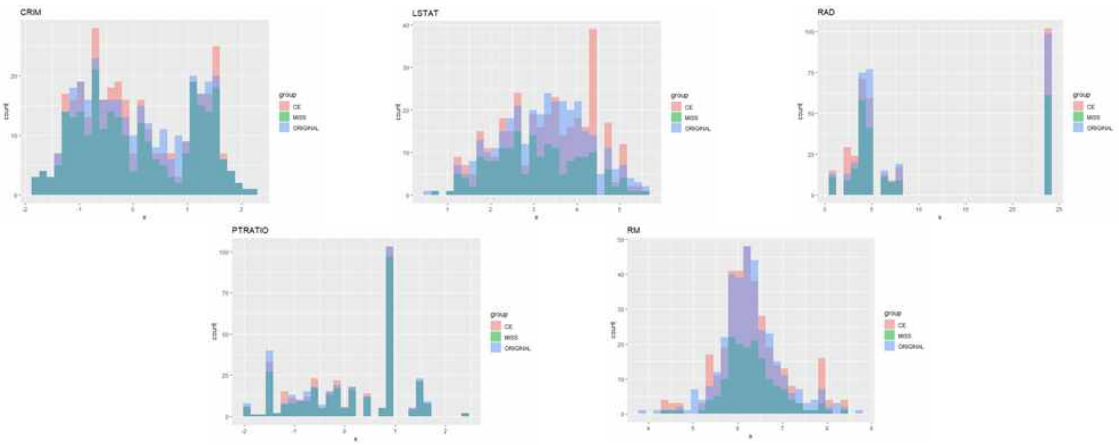


Figure 9
Simulation results from MICE

4. Summary and Conclusion

4.1 Comparison of real data

I compared one of the imputed values from each method with original value. *RAD* is a categorical variable with order, so it was rounded. From Table 8, each variable had different methods that imputed closet to the true values, but data augmentation with MLMI showed the best performance overall.

Methods		RM	LSTAT	RAD	CRIM	PTRATIO
DA.norm	m=5, i=10	5.221337	3.768884	1	-0.68982	0.282515
DA.mlmi	m=5	6	2.541324	1	-0.78376	0.494787
	m=20	6	3.32933	3	-0.51714	0.466609
EM.Amelia	m=5	5.793354	3.436601	8	-0.4251	-1.379
EM.mlmi	m=5	6	2.725136	1	-0.16321	-0.29783
	m=20	5	3.548495	4	-0.04546	0.482403
Chained-Equation (m=5)		6.174	4.274116	2	-0.6571	-1.42226
True data		4.973	3.433946	4	0.845128	-0.24402

Table 8
Real data comparison

4.2 Mean Comparison

Table 9 shows the final conclusion from 100 data sets of simulation. The mean was obtained by averaging 100 means from each simulation. I could see that data augmentation with NORM package, EM algorithm with Amelia package, and chained equations with MICE package imputed the mean close to the actual data.

Mean		RM	LSTAT	RAD	CRIM	PTRATIO
DA.norm	m=5, i=10	6.291214	3.322144	10.48525	0.081658	0.0638
DA.mlmi	m=5	5.416912	3.256702	10.47597	0.090710	0.072018
	m=20	5.395133	3.252968	10.42985	0.087838	0.079648
EM.Amelia	m=5	6.294920	3.276103	9.843379	0.004492	0.028692
EM.mlmi	m=5	5.395132	3.25296	10.42985	0.087837	0.079647
	m=20	5.457463	3.271918	10.43026	0.089126	0.080832
Chained-Equation (m=5)		6.294858	3.283044	9.739704	0.003979	0.020853
True data		6.284634	3.213478	9.549407	7.25E-17	-1.36E-13

Table 9
Mean comparison

4.3 Variance Comparison

Table 10 shows the average of variance from 100 data sets of simulation. From Table 10, the variance from EM algorithm with Amelia package was close to the variance of the actual data.

Variance		RM	LSTAT	RAD	CRIM	PTRATIO
DA.norm	m=5, i=10	0.519665	1.103588	79.62875	1.00449	1.004673
DA.mlmi	m=5	1.660532	1.38776	86.95124	1.005717	1.014679
	m=20	1.789033	1.277400	85.88700	1.029416	1.001261
EM.Amelia	m=5	0.486234	1.052581	72.146422	0.998470	0.976242
EM.mlmi	m=5	1.570668	1.08884	80.08223	0.974880	1.005234
	m=20	1.634719	1.147139	79.24630	0.962822	0.999432
Chained-Equation (m=5)		0.469793	1.078810	77.885019	1.011281	0.997531
True data		0.493670	1.040509	75.81637	1.00198	1.00198

Table 10
Variance Comparison

4.4 The best method for regression

I performed linear regression to check the performance of predicting values of *MEDV*. For each simulation training sets, I applied imputation methods. Then I established a linear regression model from each simulation data sets. And I obtained MSE values by using the test set. Table 11 shows mean and standard deviation of the MSE. Data augmentation with NORM package showed the best performance. In addition, EM algorithm with Amelia package also showed a good prediction.

Methods		MSE	STD
Chained-Equation		22.84	2.22
EM.Amelia		19.72	3.20
EM.mlmi	**m=5**	22.71	2.43
	m=20	24.06	3.40
DA.norm	**m=1, i=10**	20.07	2.64
	m=1, i=100	20.28	2.89
	m=5, i=10	19.17	2.14
	m=5, i=100	19.72	2.48
	m=20, i=10	19.18	2.15
	m=20, i=100	**18.90**	**2.05**
DA.mlmi	**m=5**	26.98	5.13
	m=20	25.12	4.46

Table 11
Linear regression MSE comparison

From Table 12, the difference between the coefficients of linear regression model from the original data and the coefficients of the same model from each simulation data. EM algorithm with NORM package showed close values to the actual true linear regression model.

Methods		Intercept	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
Em(norm)		1.264	0.21	-0.93	-1.04	1.29	-14.07	5.77	-1.41	-1.99	-0.90	-1.13	-2.32	-1.07	-5.12
EM (Mlmi)	m=5	33.08	-0.52	-1.27	-1.17	3.08	-5.35	1.04	-0.95	-3.39	-0.86	-1.21	-2.87	-1.12	-4.55
	m=20	23.50	3.21	-5.01	2.337	4.58	-6.09	4.31	0.55	-6.59	2.058	0.52	-7.21	2.17	-4.27
DA (Norm)	m=1,i=10	9.27	-0.19	-0.93	-1.07	1.76	-14.93	4.62	-1.42	-2.16	-0.89	-1.13	-2.55	-1.07	-4.73
	m=1,i=100	8.51	-0.04	-0.93	-1.08	1.76	-14.74	4.66	-1.42	-2.16	-0.89	-1.13	-2.56	-1.07	-4.69
	m=5,i=10	11.34	-0.25	-0.93	-1.07	2.29	-12.99	4.17	-1.41	-2.13	-0.88	-1.13	-2.69	-1.07	-5.13
	m=5,i=100	7.90	-0.06	-0.93	-1.09	1.62	-14.76	4.67	-1.42	-2.17	-0.90	-1.13	-2.58	-1.07	-4.63
	m=20,i=10	11.26	-0.24	-0.93	-1.07	2.28	-13.11	4.19	-1.41	-2.13	-0.88	-1.13	-2.68	-1.07	-5.13
	m=20,i=100	10.39	-0.06	-0.93	-1.09	1.81	-14.96	4.39	-1.42	-2.19	-0.88	-1.13	-2.55	-1.07	-4.78
DA (mlmi)	m=5	28.79	-0.64	-1.33	-1.09	4.62	-8.3	3.97	-1.74	-2.71	-1.27	-1.07	-4.34	-1.29	-2.54
	m=20	21.00	3.23	-5.00	2.29	5.75	-5.60	4.23	0.54	-6.68	2.05	0.52	-7.41	2.18	-3.40

Table 12
Linear regression coefficients comparison

4.5 The best method for classification

I transformed *MEDV* to binary variable, to create a classification task for the category of *MEDV*. Table 13 is the results of performing Support Vector Machine to predict the categories of *MEDV*, whether it is the top 50% group or the bottom 50% group. SVM was conducted for each simulation data. And the data was applied to the test sets to obtain the AUROC for each data. I looked at the mean and standard deviation for this AUROC, and I could see that ooo best predicted the categories of *MEDV*. CC stands for Complete-Case Analysis, and I included this result to compare if the imputation was meaningful.

ROC AUC	CC	EM	CE	DA
Mean	0.769	0.811	0.804	0.808
Std.	0.0774	0.0288	0.0358	0.0338

Table 13
AUROC results

I implemented various imputation methods using real data. In my experiment, I found that is EM algorithm is a good method in regression and classification overall. The regression coefficients were smallest when using EM algorithm. I would like to conclude that the best imputation method can be changed depending on which task I do, but EM algorithm showed good performance for regression and classification tasks. For the specific task, the comparison of performance of different imputation methods would be necessary.

5. Reference

James H. R Package 'Amelia' decription (2019)
Jonathan B. R Package 'mlmi' decription (2019)
Joseph L. R Package 'norm' decription (2013)
Stef B. R Package 'mice' decription (2011)