Comparison of different imputation methods with focus on point estimate

Yunkun Yang

20602444

STAT 454

Table of Contents

Abstract

Missing data issue has become one of the most serious problem in statistical analysis. To deal with those missing data problems, a variety methods have been developed and applied such as imputation. In this report, different ways of imputation methods will be discussed under the traditionally statistical setting and I will also run a simulation study to compare those different methods using a real-life dataset ‘California Housing’, which is publically published for analysis. [1] The illustration can be considered as not only the comparison between methods, but also a contrast between theoretical analysis and the real-life analysis.

Introduction

Missing data happens almost everywhere. No matter how detailed the experiments are designed, how carefully the investigators ask their questions, or even how patient respondents try to answer those questions. It just happens. The most serious issue is that once a data is missing, it is almost impossible to recover the actual missing value and it may also lead to more missing values as a result. Imputation can be used to solve some of the problems caused by missing values. Before the discussion of different imputation methods, missing mechanisms should be introduced.

Mechanism of Missing Data

Rubin (1976) and his colleagues (Little and Rubin, 2002) established the foundation of missing data theory. They classified the missing data problems into three categories: (1) missing completely at random (MCAR), (2) missing at random (MAR), (3) not missing at random. [2]

MCAR refers to the data where the messiness mechanism does not depend on the variable of interest, or any other variable observed. Mathematical notation can be denoted by $\mathrm{p_i} = \mathrm{p}\ (\text{for all i})$. Another interpretation of MCAR is that the data are collected or observed at random with the probability p from the last formula. Missing due to systematic error can be one of the reason why MCAR takes place.

MAR means that the missing data is conditional on some other independent variables observed, but not depends on the variable itself. It can also be called as missing conditionally at random, which ironically is also MCAR. [3] The mathematical notation can be presented as $\mathrm{p_i} = \mathrm{p(x_i)}$, the probability of observation of Unit i is depends on the value of other features of Unit i.

NMAR occurs when the missing mechanism depends both on the actual value of the missing data and other variables. $\mathrm{p_i} = \mathrm{p(x_i, y_i)}$ indicates that the chance of observing Unit i depends on both itself and some other features. One of the most obvious example for NMAR is that some rich people may not be willing to reveal their income during the experiment. Due to the difficulty of analyzing and prediction NMAR, I will not focus too much on such mechanism in the simulation section.

Imputation Methods

As mentioned above, there are a lot of imputation methods that are quite efficient and useful. I would like to discuss some of the single imputation methods such as mean imputation and regression imputation etc.

Complete Case (Case Deletion)

One of the most frequently used approach is to ignore those observations with missing data and only focus on the complete data. It is called complete case analysis or listwise deletion. Some researchers insist that deletion of incomplete records may introduce bias in estimation of the parameters. However, as long as MCAR assumption (random observed) is satisfied, a listwise deletion can produce unbiased estimates and conservative results. [4] If MCAR assumption is somehow not fulfilled, such method can cause the bias in estimates of the parameter. [5]

Mean Imputation

This method is simply substituting all the missing values with the mean of the all observed values. It is the easiest and most conservative approach method because it does not change the sample mean, but it basically ignores the effect of other variables and decrease the variance of the variable. As a result, some correlations between variables are also decreased. [6] That is the reason why mean imputation should be used under the MCAR assumption.

From the point of estimation perspective (with focus only on μ_y), mean imputation and complete case are basically the same process. Intuitively, the mean imputation has only information about the mean of the observed response variable y and the complete case also has only information about the observed response variable y. We can also prove those two methods can always give us same μ_y, but it may disregard all other independent information.

Settings: $\begin{cases} \mathrm{y_r}(\text{for y is not missing}) \quad \text{size: } \mathrm{n_r} \\ \mathrm{y_m} \ (\text{for y is missing}) \quad \text{size: } \mathrm{n - n_r} \end{cases}$

For Complete Case: $\mu_y = \bar{y} = \frac{1}{n_r} * \sum_{\text{all not missing}} y_r$

For Mean Imputation: $\mu_y = \bar{y} = \frac{1}{n}\left(\sum_{\text{all not missing}} y_r + \frac{\sum_{\text{all not missing}} y_r}{n_r} * (n - n_r)\right) =$

$\frac{1}{n}(\frac{n}{n_r} \sum_{\text{all not missing}} y_r) = \frac{1}{n_r} * (\sum_{\text{all not missing}} y_r)$

Hot Deck Imputation

Hot deck imputation involves replacing missing value with the observed values that is similar. [7] For $j \in S_m$, impute $y_j^* = y_k$where k is random selected from S_r. Because some of the different samples may be selected, the imputed values vary in each case. However, the estimate of those missing values is still the mean of the population mean.

$$E[y] = \frac{n_r}{n}E[y_r] + \frac{n - n_r}{n}E[y_m] = \frac{n_r}{n}E[y_r] + \frac{n - n_r}{n} E[y_r] = E[y_r] = \mu_y$$

Regression Imputation

Unlike the previous three methods, which disregards the effect of the independent variables, regression imputation uses more of the available information in the dataset to generate imputed values [6]. The predicted value y is the best predictor if the regression model holds for the respondents and non-respondents.

$$y_i = x_i'\beta + \varepsilon_i$$

and β can be achieved from fitting the model using $\{ (y_i, x_i) \ \ for\ i \ \in S_R\}$

The simple regression, also called deterministic regression, ignores the residuals terms because under the perfect setting of regression, $E(\varepsilon_i) = 0$. On the other hand, random regression imputation adds such residual terms so the fitted values can include more information as the complete records.

For the random regression,

$$y_j^* = x_j'\hat{\beta} + \widehat{\varepsilon_j^*}, \qquad \text{for } j \in S_m$$

and the $\widehat{\varepsilon_j^*}$ is selected from fitted residual

$$r_j = y_j - x_j\hat{\beta}, \qquad \text{for } j \in S_r$$

Nearest neighbor imputation

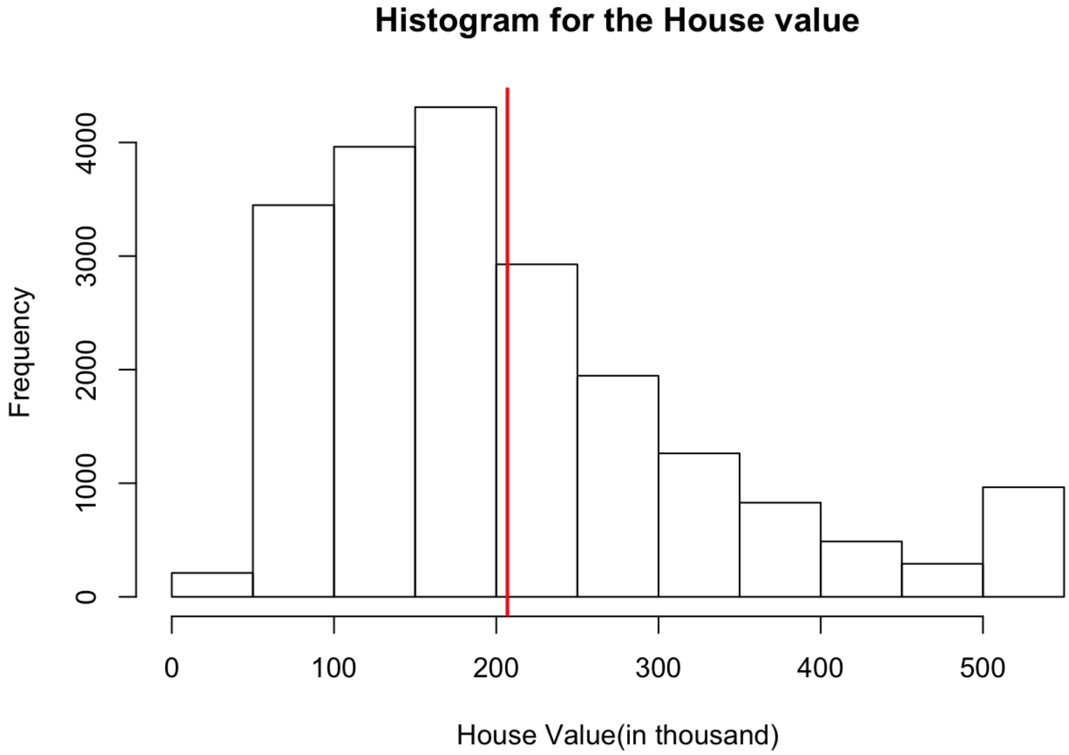
Nearest neighbor imputation is predicting the missing values by finding the k closest observed neighbors. Nearest neighbor imputation is one of the hot deck methods. However, it is more efficient than hot deck methods since NNI makes use of auxiliary information given by x's. Compared to the regression model, NNI doesn't use an explicit model to relate response and other information, which makes it more robust against the model violations than those methods based on explicit models. [8]

Imputations allows users to save the resources from collecting more data and they are able to analyze the complete dataset using their standard techniques. However, different types of imputation may have different impact on the survey result. Imputed data is not real, and the variance estimated should be considered to reflect the uncertainty. [9]

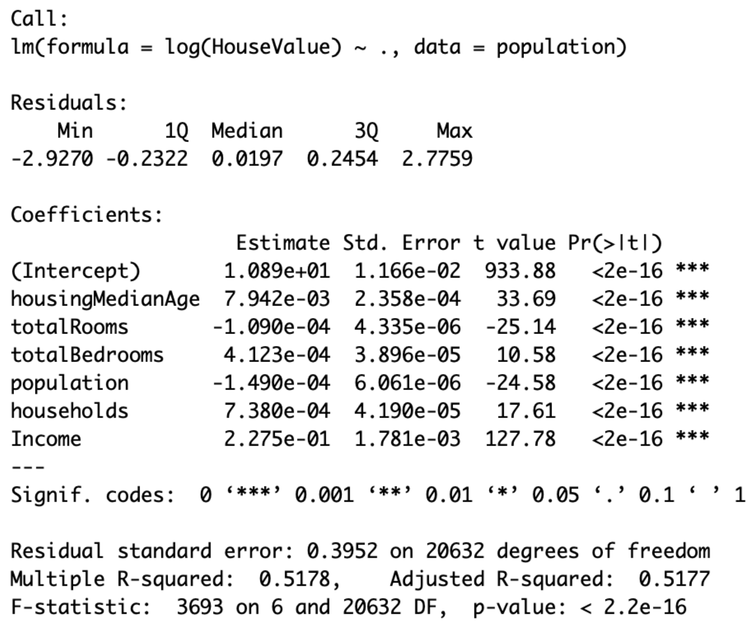
Simulation process

Setups

We will use the California Housing data which is provided by SKlearn.[1] The dataset includes the data of houses and some general information about certain district (given by longitude and latitude). There are a few columns in the dataset, longitude, latitude, housingMedianAge, total rooms, total bedrooms, populations, number of households, median income and median house value of the district. The dataset contains 20639 rows and 9 variables. We will consider each district as one record, hence we can ignore the longitude and latitude variables.



We will first plot the histogram of the House value so we are able to see the general trend of the response variable. The red straight line indicates the population mean. The shape is not a typical bell-shaped trend. It has a long right handed tail, which contains some outliers over 500k. The extreme values can cause some issue during the later simulation process.



```
Call:
lm(formula = log(HouseValue) ~ ., data = population)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.9270 -0.2322  0.0197  0.2454  2.7759 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1.089e+01  1.166e-02  933.88   <2e-16 ***
housingMedianAge  7.942e-03  2.358e-04   33.69   <2e-16 ***
totalRooms       -1.090e-04  4.335e-06  -25.14   <2e-16 ***
totalBedrooms     4.123e-04  3.896e-05   10.58   <2e-16 ***
population       -1.490e-04  6.061e-06  -24.58   <2e-16 ***
households        7.380e-04  4.190e-05   17.61   <2e-16 ***
Income            2.275e-01  1.781e-03  127.78   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

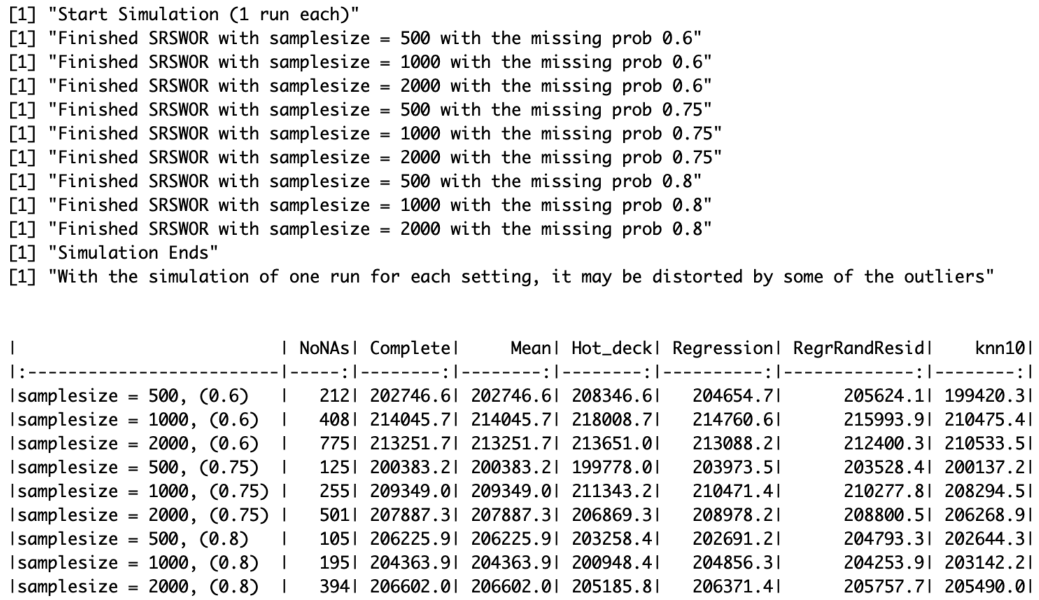
Residual standard error: 0.3952 on 20632 degrees of freedom
Multiple R-squared:  0.5178,	Adjusted R-squared:  0.5177 
F-statistic:  3693 on 6 and 20632 DF,  p-value: < 2.2e-16
```

For our better understanding, a linear model can be constructed on the entire dataset and all variables are quite significant to the housing value.

Imputation

We will use 6 methods of imputation as mentioned above, complete case, mean imputation, random hot deck imputation, simple regression imputation, random regression imputation and knn imputation (in this case we choose k = 10 so it won’t be too computationally complex but still capture information from some of the neighbors).

MCAR with 1 run each



```
[1] "Start Simulation (1 run each)"
[1] "Finished SRSWOR with samplesize = 500 with the missing prob 0.6"
[1] "Finished SRSWOR with samplesize = 1000 with the missing prob 0.6"
[1] "Finished SRSWOR with samplesize = 2000 with the missing prob 0.6"
[1] "Finished SRSWOR with samplesize = 500 with the missing prob 0.75"
[1] "Finished SRSWOR with samplesize = 1000 with the missing prob 0.75"
[1] "Finished SRSWOR with samplesize = 2000 with the missing prob 0.75"
[1] "Finished SRSWOR with samplesize = 500 with the missing prob 0.8"
[1] "Finished SRSWOR with samplesize = 1000 with the missing prob 0.8"
[1] "Finished SRSWOR with samplesize = 2000 with the missing prob 0.8"
[1] "Simulation Ends"
[1] "With the simulation of one run for each setting, it may be distorted by some of the outliers"
```

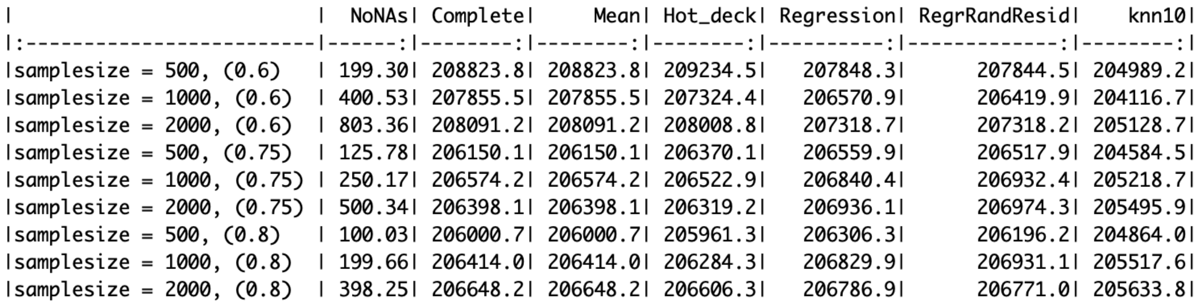
```
|                         | NoNAs| Complete|     Mean| Hot_deck| Regression| RegrRandResid|    knn10|
|:------------------------|-----:|--------:|--------:|--------:|----------:|-------------:|--------:|
|samplesize = 500, (0.6)  |   212| 202746.6| 202746.6| 208346.6|   204654.7|      205624.1| 199420.3|
|samplesize = 1000, (0.6)  |   408| 214045.7| 214045.7| 218008.7|   214760.6|      215993.9| 210475.4|
|samplesize = 2000, (0.6)  |   775| 213251.7| 213251.7| 213651.0|   213088.2|      212400.3| 210533.5|
|samplesize = 500, (0.75)  |   125| 200383.2| 200383.2| 199778.0|   203973.5|      203528.4| 200137.2|
|samplesize = 1000, (0.75) |   255| 209349.0| 209349.0| 211343.2|   210471.4|      210277.8| 208294.5|
|samplesize = 2000, (0.75) |   501| 207887.3| 207887.3| 206869.3|   208978.2|      208800.5| 206268.9|
|samplesize = 500, (0.8)   |   105| 206225.9| 206225.9| 203258.4|   202691.2|      204793.3| 202644.3|
|samplesize = 1000, (0.8)  |   195| 204363.9| 204363.9| 200948.4|   204856.3|      204253.9| 203142.2|
|samplesize = 2000, (0.8)  |   394| 206602.0| 206602.0| 205185.8|   206371.4|      205757.7| 205490.0|
```

mean table of different imputation methods with the sample size = (500,1000,2000) and inclusion probability = (0.6, 0.75, 0.8) with 1 runs of each simulation.

The first simulation for MCAR with missing probability 0.4,0.25 and 0.2, the sample sizes are 500,1000,2000. In the table, there are number of Nas, the mean under complete case, mean imputation, random hot deck, regression imputation, random imputation and knn imputation. As we anticipated, the point estimate of mean imputation and complete case are the same for all cases.

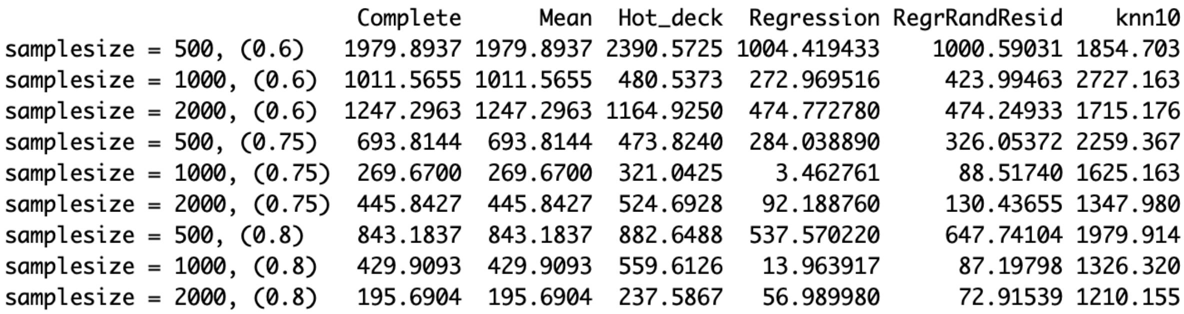
The population mean is 206843.91, we can see the differences between each imputation method are quite significant. It can be caused by either the original distribution of the model or the chance of random selection (because we only take each sample one time). To fix the issue brought by the random selection, we run each sampling 100 times and due to the statistic theory, the mean of each point estimate will still convergence to the true mean.

MCAR with 100 run each



```
|                          |  NoNAs| Complete|     Mean| Hot_deck| Regression| RegrRandResid|    knn10|
|:-------------------------|------:|--------:|--------:|--------:|----------:|-------------:|--------:|
|samplesize = 500, (0.6)   | 199.30| 208823.8| 208823.8| 209234.5|   207848.3|      207844.5| 204989.2|
|samplesize = 1000, (0.6)  | 400.53| 207855.5| 207855.5| 207324.4|   206570.9|      206419.9| 204116.7|
|samplesize = 2000, (0.6)  | 803.36| 208091.2| 208091.2| 208008.8|   207318.7|      207318.2| 205128.7|
|samplesize = 500, (0.75)  | 125.78| 206150.1| 206150.1| 206370.1|   206559.9|      206517.9| 204584.5|
|samplesize = 1000, (0.75) | 250.17| 206574.2| 206574.2| 206522.9|   206840.4|      206932.4| 205218.7|
|samplesize = 2000, (0.75) | 500.34| 206398.1| 206398.1| 206319.2|   206936.1|      206974.3| 205495.9|
|samplesize = 500, (0.8)   | 100.03| 206000.7| 206000.7| 205961.3|   206306.3|      206196.2| 204864.0|
|samplesize = 1000, (0.8)  | 199.66| 206414.0| 206414.0| 206284.3|   206829.9|      206931.1| 205517.6|
|samplesize = 2000, (0.8)  | 398.25| 206648.2| 206648.2| 206606.3|   206786.9|      206771.0| 205633.8|
```

mean table of different imputation methods with the sample size = (500,1000,2000) and inclusion probability = (0.6, 0.75, 0.8) with 100 runs of each simulation.



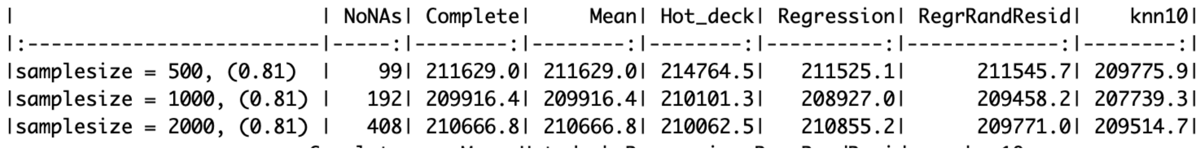
```
                          Complete      Mean  Hot_deck  Regression RegrRandResid    knn10
samplesize = 500, (0.6)   1979.8937 1979.8937 2390.5725 1004.419433    1000.59031 1854.703
samplesize = 1000, (0.6)  1011.5655 1011.5655  480.5373  272.969516     423.99463 2727.163
samplesize = 2000, (0.6)  1247.2963 1247.2963 1164.9250  474.772780     474.24933 1715.176
samplesize = 500, (0.75)   693.8144  693.8144  473.8240  284.038890     326.05372 2259.367
samplesize = 1000, (0.75)  269.6700  269.6700  321.0425    3.462761      88.51740 1625.163
samplesize = 2000, (0.75)  445.8427  445.8427  524.6928   92.188760     130.43655 1347.980
samplesize = 500, (0.8)    843.1837  843.1837  882.6488  537.570220     647.74104 1979.914
samplesize = 1000, (0.8)   429.9093  429.9093  559.6126   13.963917      87.19798 1326.320
samplesize = 2000, (0.8)   195.6904  195.6904  237.5867   56.989980      72.91539 1210.155
```

Difference Table (between the simulated mean and population μ)

After run the simulation 100 times for the same settings, we have 100 means for different imputation methods. By CLT, they should all have the mean μ of the population. Because there are a few extreme values in the original distribution, by taking multiple runs, the effect of the extreme values cannot be erased, but somehow buffered.

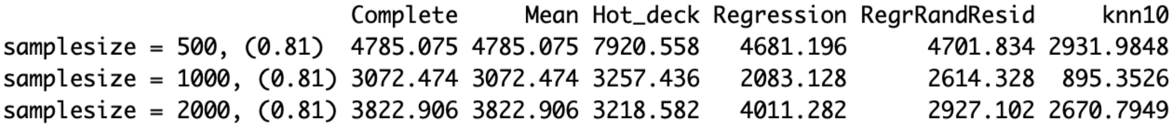
We present the second table and calculate the difference between the average of those means and the population mean. We can see that regression imputation method outruns other imputation methods. One of the reason that Mean imputation and Random Hot Deck imputation do not work in this case can be that they cannot ignore the effect of the outliers. The regression works well because the response variable in the population is somehow correlated with other covariates as we already discussed a little in the previous section.

MAR simulation with 1 run



```
|                          | NoNAs| Complete|     Mean| Hot_deck| Regression| RegrRandResid|    knn10|
|:-------------------------|-----:|--------:|--------:|--------:|----------:|-------------:|--------:|
|samplesize = 500, (0.81)  |    99| 211629.0| 211629.0| 214764.5|   211525.1|      211545.7| 209775.9|
|samplesize = 1000, (0.81) |   192| 209916.4| 209916.4| 210101.3|   208927.0|      209458.2| 207739.3|
|samplesize = 2000, (0.81) |   408| 210666.8| 210666.8| 210062.5|   210855.2|      209771.0| 209514.7|
```

MAR mean table of different imputation methods with the sample size = (500,1000,2000) and inclusion probability = 0.81 (Derived by response rate function) with 1 runs of each simulation.



```
                          Complete     Mean Hot_deck Regression RegrRandResid     knn10
samplesize = 500, (0.81)  4785.075 4785.075 7920.558   4681.196      4701.834 2931.9848
samplesize = 1000, (0.81) 3072.474 3072.474 3257.436   2083.128      2614.328  895.3526
samplesize = 2000, (0.81) 3822.906 3822.906 3218.582   4011.282      2927.102 2670.7949
```

Difference Table (between the simulated mean and population μ)

In the MAR mechanism, a response function is required to calculate the missing probability of each response. We propose a function Response Rate = 1 + 0.05 * log(households) + 0.1 * log(Income) indicating that there is a higher chance the respondents may not respond in a larger community and rich people may not willing to reveal the house value of their family. Due to the dataset contains median value of the income, we set the coefficient 0.1 to buffer the missing chance. With inverse logistic model, we are able to control the missing rata as 0.81.

We noticed that compared to MCAR, under MAR mechanism, the imputed values lead to a much higher difference even with the regression imputation or random regression imputation.

However, the knn method does dive a slightly better result. It is still a sign that the response variable is related to the independent variables. It is very likely that with more tuning on the number k, the imputed values can be improved. To work with MAR mechanism, more advanced imputation methods are required or a different technique can better work.

Conclusion & Remarks

After the simulation with the real-life data, we realize that the simulations are not satisfiable as we anticipated. The main reason for that is the data does not belong to a bell shaped distribution. Assumption to MCAR are not always realistic, MAR and even NMAR take place much more often in the real life. A more advanced model is required to run the imputation such as some machine learning models or EM imputation method.

Also, in this case, we did not generate the missing values in the X's data. It happens frequently and it can be more difficult if there are multiple missing values in a record. Multiple imputation can be used to deal with such case. Multiple imputation is an iterative form of stochastic imputation. The distribution of observed data is used to estimate multiple values and the estimated value is used to predict other values. [10]

Reference:

[1]: Sklearn dataset California Housing. Scikit Learn. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html#sklearn.datasets.fetch_california_housing

[2]: Shinichi Nakagawa. Missing data: mechanisms, methods, and messages, P 83. Retrieved from http://www.i-deel.org/uploads/5/2/4/1/52416001/chapter_4.pdf

[3]: KAREN GRACE-MARTIN. What is the difference between MAR and MCAR missing data? The Analysis Factor. Retrieved from https://www.theanalysisfactor.com/mar-and-mcar-missing-data/

[4]: Hyun Kang. The prevention and handling of the missing data. Korean J Anesthesiol. 2013 May; 64(5): 402–406. Published online 2013 May 24. Retrieved from: 10.4097/kjae.2013.64.5.402

[5]: Donner A. The relative effectiveness of procedures commonly used in multiple regression analysis for dealing with missing values. Am Stat. 1982;36:378–381. Retrieved from https://amstat.tandfonline.com/doi/abs/10.1080/00031305.1982.10483055?casa_token=Q9hXu5V9qk4AAAAA:dzWjwR_u37AicyXyTPA6fxq0AJ0mtOMZLp6tobn452xT9VCzmfzMdvCb9pAuUInGFm8Vt_-jvoyK#.Xo0w_JNKhQI

[6]: Carol M. Musil. A Comparison of Imputation Techniques for Handling Missing Data. Western Journal of Nursing Research,2002,24(7),815-829. Retrieved from https://journals.sagepub.com/doi/pdf/10.1177/019394502762477004

[7]: Andridge, R. R., & Little, R. J. (2010). A Review of Hot Deck Imputation for Survey Non-response. International statistical review = Revue internationale de statistique, 78(1), 40–64. Retrieved from https://doi.org/10.1111/j.1751-5823.2010.00103.x

[8]: Jiahua Chen, Jun Shao. Nearest Neighbor Imputation for Survey Data. Journal of Of®cial Statistics, Vol. 16, No. 2, 2000, pp. 113±131. Retrieved from https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/nearest-neighbor-imputation-for-survey-data.pdf

[9]: Judi Scheffer. Dealing with missing data. Res. Lett. Inf. Math. Sci. (2002) 3, 153-160. Retrieved from http://www.massey.ac.nz/~wwiims/research/letters/

[10]: Bruin, J. 2006. newtest: command to compute new test. UCLA: Statistical Consulting Group. Retrieved from https://stats.idre.ucla.edu/stata/ado/analysis/.

Appendix:

~~~
---
title: "STAT454_final_proj"
output: html_document
author: Yunkun Yang
---


```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
setwd("~/Documents/Files/2020Winter/STAT454/Final_Project")
library(dplyr)
library(DMwR)
library(knitr)
rm(list = ls())

```
~~~

##Setups

We will use the Califonia Housing data which is provided by SKlearn.[1] The dataset includes the data of houses and some general information about certain district (given by longitutde and latitude). There are a few columns in the dataset, longitute, latitude, housingMedianAge, total rooms, total bedrooms, populations, number of households, median income and median house value of the district. The dataset contains 20639 rows and 9 variables. We will consider each district as one record, hence we can ignore the longitute and latitude variables.

~~~
```{r loading data}
set.seed(611)
population <- read.csv('CaliforniaHousing/cal_housing.data')
pop_colname <-
c('longitude','latitude',"housingMedianAge","totalRooms","totalBedrooms","population",
"households","Income","HouseValue")
names(population) <- pop_colname
population $longitude <- NULL
population$latitude <- NULL
N <- nrow(population)

pop_mu <- mean(population$HouseValue)
hist(population$HouseValue/1000,freq = TRUE,main = 'Histogram for the House value',xlab =
'House Value(in thousand)')
abline(v = pop_mu/1000, col = "red", lwd = 2)

```
~~~

We will first plot the histogram of the House value so we are able to see the general trend of the response variable. The red stright line indicates the population mean. The shape is not a typical bell-shaped trend, and it contains some outliers over 500k.

~~~
```{r pop linear model}
pop_model <- lm(log(HouseValue) ~ . , data = population)
summary(pop_model)
```
~~~

For our better understanding, a linear model can be constructed on the entire dataset and all variables are quite siginificant to the housing value.

##Imputation

We will use 6 methods of imputation as mentioned above, complete case, mean imputation, random hot deck imputation, simple regression imputation, random regression imputation and knn imputation (in this case we choose k = 10 so it won't be too computationally complex but still capture information from some of the neighbors).

~~~
```{r imputation code}
remove_miss <- function(y){
  new_y <- y[! is.na(y)]
  return(new_y)
}
~~~

```
imp_mean <- function(y){
  return( ifelse( is.na(y) , mean(y,na.rm = TRUE) , y) )
}


imp_hot_deck <- function(y){
  y_complete <- y[! is.na(y)]
  y_miss_len <- length(y[is.na(y)])
  y_imp <- sample(y_complete,y_miss_len)
  new_y <- ifelse(is.na(y),y_imp,y)
  return(new_y)
}

## With x
imp_regress <- function(x,y,random_residual = FALSE){
  training <- x[! is.na(y),]
  train_y <- y[! is.na(y)]
  training_dt <- cbind(training,train_y)
  model1 <- lm(train_y ~ ., data = training_dt)
  testing <- x[is.na(y),]
  pred <- predict(model1,testing)
```

~~~
  if(random_residual == TRUE){
    residual <- residuals(model1)
    random_residual <- sample(residual,length(pred))
    pred <- pred + random_residual
  }
  new_y <- c(train_y,pred)
  new_x <- rbind(training,testing)
  return(list(new_x,new_y))
}


imp_knn <- function(x,y,k = 10){
  dt <- cbind(x,y)
  dt <- knnImputation(dt,k=k)
  return(list(dt[,-ncol(dt)],dt[,ncol(dt)]))
}
```
```{r}
imputation_simulate <- function(dt_pop, samplesize, runs = 500,
                              k=10, method= c("MCAR","MAR","NMAR")){
  n <- samplesize
  result <- data.frame()
  for (i in seq(1,runs)) {
    sam <- sample(N,n)
~~~

```
    dt <- dt_pop[sam,]
    y <- ifelse(dt[,method] == 1,dt$HouseValue,NA)
    x <- dt[,c(1,2,3,4,5,6)]
    na_no <- sum(is.na(y))
    remove_miss_ybar <- remove_miss(y) %>% mean()
    imp_mean_ybar <- imp_mean(y) %>% mean()
    imp_hot_deck_ybar <- imp_hot_deck(y) %>% mean()
    imp_regress_ybar <- imp_regress(x,y)[[2]] %>% mean()
    imp_regress_wresid_ybar <- imp_regress(x,y,random_residual = TRUE)[[2]] %>% mean()
    imp_knn10_ybar <- imp_knn(x,y,k=k)[[2]] %>% mean()

    temp <- list('NoNAs'= na_no, 'Complete'=remove_miss_ybar, 'Mean'= imp_mean_ybar,
                 'Hot_deck'= imp_hot_deck_ybar,
                 'Regression'= imp_regress_ybar,
                 'RegrRandResid'= imp_regress_wresid_ybar,
                 'knn10'= imp_knn10_ybar)
    result <- rbind(result,temp)
  }
  return(result)
}
```

## MCAR Simulation

~~~
### MCAR with 1 run each


```{r}
simulation <- data.frame()
samplesize <- c(500,1000,2000)
prob <- c(0.6,0.75,0.8)
set.seed(611)


## MCAR (Missing completely at random)
print('Start Simulation (1 run each)')
for (p in prob) {
  MCAR <- rep(1,N)
  mis <- sample(N,(1-p)*N)
  MCAR[mis] <- 0
  dt_pop <- cbind(population,MCAR)


  for (i in samplesize) {
    temp <- imputation_simulate(dt_pop, i,runs=1,method = "MCAR")
    #temp <- as.data.frame(colMeans(temp)) %>% t()
    rownames(temp) <- paste0('samplesize = ', i,', (', p,')')
    simulation<-rbind(simulation,temp)#c(simulation,temp)
    print(paste0('Finished SRSWOR with samplesize = ', i,' with the missing prob ', p))
~~~

~~~
  }
}
print('Simulation Ends')
print('With the simulation of one run for each setting, it may be distorted by some of the
outliers')
kable(simulation)
```
~~~

The first simulation for MCAR with missing probability 0.4,0.25 and 0.2, the sample size are 500,1000,2000. In the table, there are number of nas, the mean under complete case, mean imputation, random hot deck, regression imputation, random imputation and knn imputation. The population mean is ```r pop_mu```, we can see the differences between each imputation method are quite significant. It can be caused by either the original distribution of the model or the chance of random selection (because we only take the sample one time).

### MCAR with 100 run each

~~~
```{r}
simulation <- c()
simulation <- data.frame()
samplesize <- c(500,1000,2000)
prob <- c(0.6,0.75,0.8)
set.seed(611)
~~~

~~~~
## MCAR (Missing completely at random)
print('Start Simulation (100 run each)')
for (p in prob) {
  MCAR <- rep(1,N)
  mis <- sample(N,(1-p)*N)
  MCAR[mis] <- 0
  dt_pop <- cbind(population,MCAR)

  for (i in samplesize) {
    temp <- imputation_simulate(dt_pop, i,runs=100,method = "MCAR")
    temp <- as.data.frame(colMeans(temp)) %>% t()
    rownames(temp) <- paste0('samplesize = ', i,', (', p,')')
    simulation<-rbind(simulation,temp)#c(simulation,temp)
    print(paste0('Finished SRSWOR with samplesize = ', i,' with the missing prob ', p))
  }
}
print('Simulation Ends')
```


```{r}
kable(simulation)
print('Difference table abs(Mean after imputation - pop_mu)')
~~~~

~~~
abs(simulation[,-1]-pop_mu)
```
~~~

After run the simulation 100 times for the same settings, we have 100 means for different imputation methods. By CLT, they should all have the mean $\mu$ of the population. Because there are a few extreme values in the original distribution, by taking multiple runs, the effect of the extreme values cannot be erased, but somehow buffered.

We present the second table and calculate the difference between the average of those means and the population mean. We can see that regression imputations outrun other imputation methods. One of the reason that Mean imputation and Random Hot Deck imputation do not work in this case can be that they cannot ignore the effect of the outliers. The regression works well because the original values in the population can be modelled using proper regression model as we discussed in the previous section.

## MAR simulation with 1 run

~~~
```{r}
## MAR (Missing at random)
# Response Propensity
simulation <- data.frame()
samplesize <- c(500,1000,2000)
~~~

~~~
set.seed(611)
print('Start Simulation (1 run each)')
res <- 1 + 0.05 * log(population$households) + 0.1*log(population$Income)
rp <- exp(res) / (1+exp(res))
MAR <- rbinom(N, 1, rp)
dt_pop <- cbind(dt_pop,MAR)
p_mar <- round(mean(rp),digits = 2)
for (i in samplesize) {
  temp <- imputation_simulate(dt_pop, i,runs=1,method = "MAR")
  temp <- as.data.frame(temp)
  rownames(temp) <- paste0('samplesize = ', i,', (', p_mar,')')
  simulation<-rbind(simulation,temp)#c(simulation,temp)
  print(paste0('Finished SRSWOR with samplesize = ', i,' with the missing prob ', p_mar))
}
print('Simulation Ends')
kable(simulation)
print('Difference table abs(Mean after imputation - pop_mu)')
simulation[,-1]-pop_mu %>% abs
```
~~~

In the MAR mechnism, a response function is required to calculate the missing probability of each response. We propose a function $Response Rate = 1 + 0.05 * log(households) + 0.1 * log(Income)$ indicating that there are higher chance the respondents may not respond in a larger community and rich people may not willing to reveal the house value of their family. Due to the dataset contains median value of the income, we set the coefficient 0.1 to buffer the missing chance. With a inverse logistic model, we are able to control the missing rata as ```r p_mar```.

We noticed that compared to MCAR, under MAR mechnism, the imputed values did not contribute too much, and the difference between the calculated mean and population mean is much bigger. To work with MAR mechnism, more advanced imputation methods are required or a different techniques can better work.

~~~
```{r,echo=FALSE}
## NMAR (Not missing at random)
p <- 0.95
sort.y = sort(population$HouseValue, decreasing=TRUE)
ceiling   = sort.y[ceiling((1-p)*N)]
NMAR <- ifelse(population$HouseValue > ceiling,0,1)
dt_pop <- cbind(dt_pop,NMAR)
```
~~~