# Multiple Imputation for Missing Data

Missing data arises in virtually every study. No matter what type of study you are doing, missing data is practically inevitable:

- Survey – Nonresponse arises for many reasons, including lack of interest, lack of time, deliberate decision to not participate due to subject matter, nonsensical or out of range responses are often found, coding errors in data entry or data transfer, etc.
- Data base study – Almost always missing items, for reasons including missed appointments in clinical data base studies, incomplete or highly inaccurate RAMQ data (missing diagnoses since physicians get paid by medical act, not by diagnosis), not everyone captured in data base (cancer registries are incomplete, more so for some cancers compared to others), etc.
- Clinical trials – Dropouts, loss to follow-up, machine failures, missed or missing doses, subject error, data entry errors, etc.

When missing data occurs, it can cause bias in any analyses, as well as loss of statistical efficiency due to lowering of sample size. It is therefore important to consider whether one needs to adjust for missing data or not. For very small percentages of missing data with no large bias expected, can sometimes simply ignore missing data. In probably the majority of cases, however, one should investigate possible biases.

Various ways to adjust, including adjusted maximum through use of the EM algorithm, single and multiple imputation, and various methods proposed over the years for special situations for example, "last value carried forward" in clinical trials).

We will learn about multiple imputation, the "gold standard" method for dealing with missing data. Multiple imputation is rather easy to carry out in practice, and can be used in virtually any missing data problem. Further, as we will see, it can be used for both "ignorable" and "non-ignorable" missing data problems.

**Important:** No missing data technique is perfect. All methods carry assumptions, and almost always, these assumptions are **unverifiable**. While adjusting for missing data remains a good idea, one should consider these adjustments as "helpful" in investigating the problem, and not as providing "definitive" answers. Good statistical practice dictates that one does an analysis without adjustment, and then usually **several** adjusted analyses, to investigate a range of possible inferences, and to investigate robustness to the missing data and assumptions about the missing data. This topic is delved into in detail in the article by Kmetic et al, which follows these introductory notes.

We will first see see various types of missing data mechanisms, i.e., ways that data can go missing, each of which has a different effect on the analyses. We will then see multiple imputation as a possible solution.

## Simulated Regression Data Set

We will look at two different data sets, one simulated in R, the other a real data set from the Canadian Multicentre Osteoporosis Study (CaMos, see article by Kmetic et al).

We will use a simulated data set, so that we know the correct inferences, and so can see the effect of missing data. The Kmetic et al article uses a real data set, so that we can learn about dealing with missing data in a realistic context, and the extra issues that arise in practice.

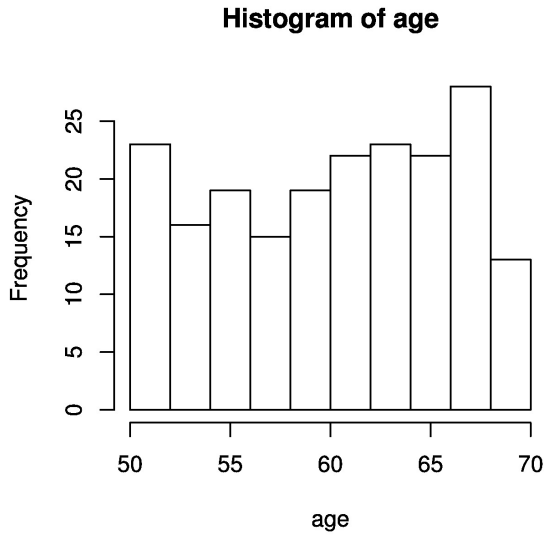
Suppose we have the following (assumed exactly true) regression equation (CRP is a measure of inflammation, of interest to cardiologists for predicting future MI's . . . this is **not** a particularly realistic data set):

$$CRP = 5 + 0.2 \times age + 0.5 \times sex$$

Assume sex is coded as 1 = male, and 0 = female, so males have slightly higher values than females, and CRP tends to rise with age, by about 2 points per decade. Suppose that the residual standard deviation is $\sigma = 1$.

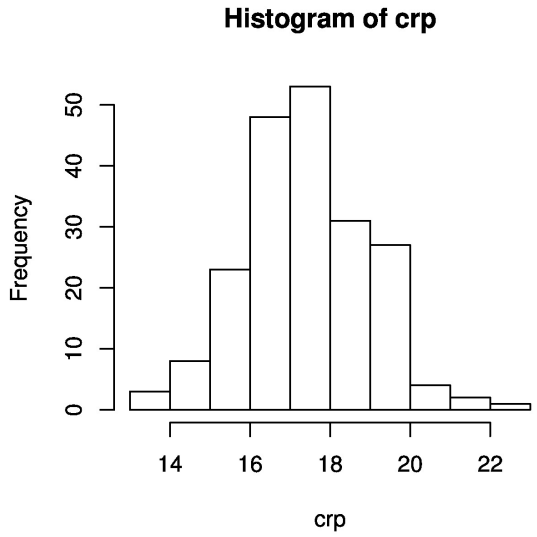
We will now simulate data on 200 subjects, aged between 50 and 70 years old, and evenly divided between males and females. We can simulate such a data set in R as follows:

```
> age <- sample(50:70, size=200, replace=T)
> age
  [1] 58 68 61 59 62 60 69 50 53 56 55 65 55 69 70 56 66 56 64 52
 [21] 68 62 52 66 54 60 51 50 63 55 50 63 53 63 60 65 68 68 66 51
 [41] 63 68 60 60 67 64 63 70 68 60 62 58 65 59 67 61 50 65 57 65
 [61] 64 68 64 64 70 66 50 59 66 62 55 56 68 65 57 56 64 54 69 68
 [81] 65 52 54 65 62 69 63 67 62 68 69 64 56 51 50 54 62 67 51 63
[101] 55 67 62 58 53 56 63 54 64 57 67 69 61 54 56 59 63 55 62 59
[121] 67 57 54 51 65 61 68 56 62 56 68 64 55 66 57 50 68 67 67 66
[141] 58 59 57 58 54 54 50 64 70 60 60 51 59 63 56 54 51 65 66 52
[161] 68 68 52 60 62 65 59 57 70 68 62 59 50 68 68 60 64 56 57 61
[181] 53 62 54 62 61 69 52 66 65 57 57 63 62 70 67 65 64 51 61 54
> hist(age)
```



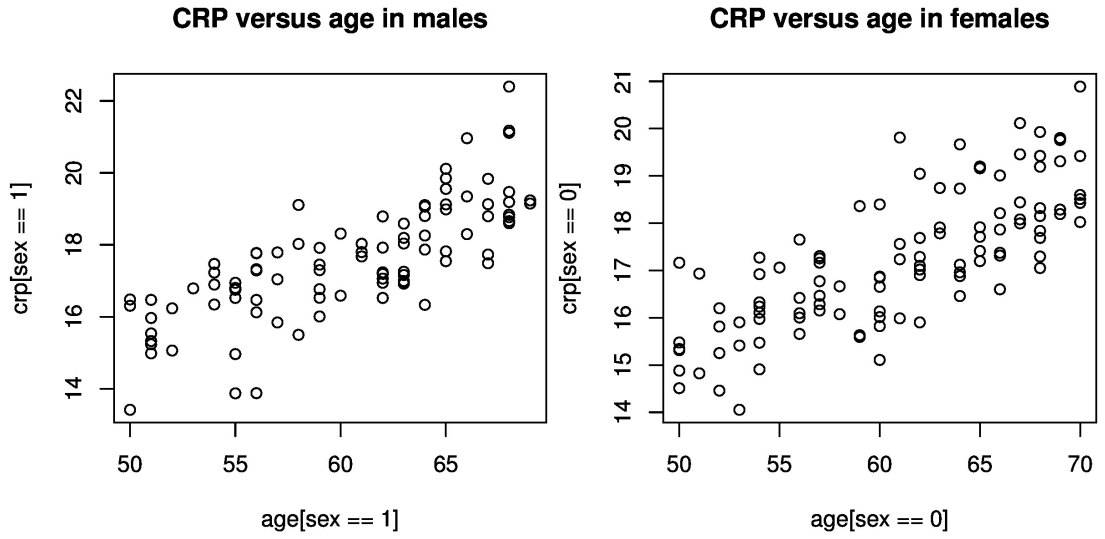

```
> sex <-rbinom(200, 1, prob=0.5)
> sex
  [1] 1 0 1 1 1 1 1 0 1 1 0 0 1 0 0 1 0 1 0 1 1 0 0 1 0 0 1 0 1 1
 [31] 1 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0
 [61] 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0 0 1 0 0 1 1 1 0
 [91] 1 1 1 1 1 1 0 1 0 1 1 0 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 1 1 1
[121] 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 0
[151] 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 1 1 1 0 1
[181] 0 1 1 1 1 0 0 1 0 0 1 1 0 0 1 1 0 0 0 1
> mean(age)
[1] 60.66
> mean(sex)
[1] 0.465
> crp <- 5 + 0.2*age + 0.5*sex + rnorm(200, mean=0, sd=1)
> crp
  [1] 15.49734 17.68776 17.79758 16.52655 17.19128 16.58653
  [7] 19.23419 14.51001 16.78875 16.46769 17.06029 17.19751
 [13] 16.94474 19.79995 18.02299 16.12339 19.00835 17.76363
........ etc ..... messy to look at, so ..................
> round(crp,2)
  [1] 15.50 17.69 17.80 16.53 17.19 16.59 19.23 14.51 16.79 16.47
 [11] 17.06 17.20 16.94 19.80 18.02 16.12 19.01 17.76 17.12 16.23
 [21] 21.17 19.04 14.46 20.96 17.27 15.82 14.98 17.16 17.16 14.96
 [31] 16.30 18.74 14.05 17.91 16.13 17.41 19.42 22.40 17.37 15.32
 [41] 17.25 18.77 16.87 16.65 18.00 16.46 16.96 19.42 18.32 18.40
 [51] 17.02 19.11 19.20 15.59 19.83 17.56 15.48 19.12 16.77 17.91
 [61] 18.80 18.84 19.67 17.87 20.89 17.86 15.31 15.63 16.60 16.90
 [71] 13.87 17.65 19.47 17.54 17.30 17.29 16.96 15.97 19.31 19.19
 [81] 19.85 15.81 16.23 17.82 17.02 19.76 16.99 17.73 16.52 17.84
```

```
 [91] 19.15 16.33 17.32 15.97 16.48 17.47 17.29 19.13 14.82 18.03
[101] 16.80 18.44 16.94 16.66 15.90 15.66 16.92 16.92 18.26 17.79
[111] 19.46 18.19 19.81 17.23 16.00 16.01 18.19 16.75 17.23 16.77
[121] 18.08 17.25 14.91 15.23 20.11 15.99 21.11 13.88 15.90 16.09
[131] 17.29 16.88 16.52 18.21 16.15 14.88 18.66 18.79 20.12 17.31
[141] 18.02 18.36 17.04 16.07 16.32 15.47 13.41 19.06 18.60 15.11
[151] 16.85 16.47 17.45 17.78 16.42 16.11 15.54 18.98 19.34 15.06
[161] 18.15 17.05 15.25 16.01 17.06 17.71 17.29 17.16 18.51 18.60
[171] 17.69 17.92 15.34 19.19 19.93 18.31 19.10 17.77 16.46 17.68
[181] 15.41 18.79 16.89 17.92 18.03 18.29 16.20 18.30 19.16 16.29
[191] 15.84 18.59 17.10 18.43 17.49 19.55 18.73 16.93 17.24 16.34
```




We can examine the scatter plots in both males and females:

```
> plot(age[sex==1], crp[sex==1], main = "CRP versus age in males")
> plot(age[sex==0], crp[sex==0], main = "CRP versus age in females")
```




Finally, we can do a frequentist regression analysis in R to check that the two coefficients are well estimated by least squares, and with no missing data:

```
> crp.results <-lm(crp ~ age + sex)
> summary(crp.results)

Call: lm(formula = crp ~ age + sex)

Residuals:
     Min       1Q   Median       3Q      Max
-2.83628 -0.63559 -0.02836  0.66047  3.31742

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.2049     0.7373   7.060 2.79e-11 ***
age           0.1973     0.0120  16.442  < 2e-16 ***
sex           0.4587     0.1400   3.276  0.00125 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9865 on 197 degrees of freedom Multiple
R-Squared: 0.5839,     Adjusted R-squared: 0.5797
F-statistic: 138.2 on 2 and 197 DF,  p-value: < 2.2e-16
```

So everything seems to have worked extremely well. This is not at all surprising, since we simulated data via an exactly linear equation, and then estimated coefficients from a model that was exactly correct.

What happens if we have missing data?

Suppose that, in particular, 1/3 of the age and 1/3 of the sex data are in fact missing. First suppose they are missing completely at random (MCAR). As the name implies, under MCAR, the data points are simply missing completely at random, with no relation of the probability of being missing to any values in the data set (or outside of the data set).

To create a data set that is MCAR, we simply need to delete some items in each of the age and sex variables. To do this in R:

```
> mcar.for.age <- rbinom(200, 1, prob=0.33333)
> mcar.for.age
  [1] 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 0 0 1 1 0 0 1 0 1
 [31] 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
 [61] 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 1 0 0
 [91] 1 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0
[121] 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 0
[151] 0 1 0 0 0 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1
[181] 0 1 1 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0
> mcar.for.sex <- rbinom(200, 1, prob=0.33333)
> mcar.for.sex
  [1] 0 1 0 0 0 1 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 1 1 0 0 1
 [31] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0
 [61] 1 1 1 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 1 0 1
 [91] 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 0 0 0 0 1 1 0 0 0
[121] 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1
[151] 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 1 0 1 0 0 0
[181] 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0
> age.mcar <- age
> sex.mcar <- sex
> age.mcar[mcar.for.age==1] <- NA
> age.mcar
  [1] 58 68 NA 59 62 60 69 NA 53 56 55 65 NA 69 NA 56 66 NA NA NA
 [21] 68 62 52 NA NA 60 51 NA 63 NA 50 63 NA 63 60 65 NA 68 66 51
 [41] 63 68 60 60 NA 64 63 70 68 NA 62 NA 65 59 67 61 50 65 57 65
 [61] NA 68 64 64 70 66 50 59 NA 62 55 NA 68 NA 57 56 64 NA 69 NA
 [81] NA NA NA 65 62 NA 63 NA 62 68 NA 64 56 NA 50 NA NA 67 NA 63
[101] NA 67 62 58 NA 56 63 54 64 57 67 NA NA NA 56 59 63 NA 62 59
[121] 67 57 NA 51 65 61 NA 56 NA 56 NA NA 55 NA 57 NA NA 67 67 66
[141] NA NA 57 58 NA 54 NA 64 70 60 60 NA 59 63 56 54 51 NA 66 NA
[161] 68 NA 52 NA NA NA 59 57 70 68 62 59 50 68 68 NA 64 NA 57 NA
[181] 53 NA NA 62 61 69 52 NA 65 57 57 63 NA NA 67 NA 64 NA 61 54
> sex.mcar[mcar.for.sex==1] <- NA
> sex.mcar
  [1]  1 NA  1  1  1 NA  1  0 NA  1  0 NA NA  0  0  1 NA  1  0 NA
```

```
 [21] NA  0  0  1  0 NA NA  0  1 NA  1  0  0 NA  0  0  0  1  0  1
 [41]  1  1  0  0  0 NA  1 NA  0  0 NA  1 NA  0  1  0  0  1  0  0
 [61] NA NA NA  1 NA  0  0  0 NA  0  1  0 NA  1  0 NA  0  0  0  1
 [81] NA  0  0 NA  0  0 NA NA  1 NA  1 NA NA  1  1  1 NA  1  0  1
[101] NA NA NA NA  0 NA NA  0  1 NA NA  0  0  1  0 NA NA  1  1  1
[121] NA NA NA  1 NA NA NA  1  0  0  0 NA NA NA NA NA NA  1  0 NA
[141]  1  0 NA  0  0 NA  1  1  0 NA NA NA  1 NA NA  0 NA  1  1  1
[161]  0  0  0  0  1 NA NA  0 NA  1 NA  1 NA  0 NA  1 NA  1  0  1
[181]  0  1  1  1 NA  0 NA NA  0  0  1  1  0  0  1 NA  0 NA  0  1
```

Now try the regression again:

```
> crp.results.mcar <-lm(crp ~ age.mcar + sex.mcar)
> summary(crp.results.mcar)

Call: lm(formula = crp ~ age.mcar + sex.mcar)

Residuals:
     Min       1Q   Median       3Q      Max
-2.45138 -0.57833 -0.05643  0.53546  3.43273

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.88059    1.13373   3.423  0.00100 **
age.mcar     0.21977    0.01855  11.847  < 2e-16 ***
sex.mcar     0.14001    0.20568   0.681  0.49813
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9118 on 76 degrees of freedom Multiple
R-Squared: 0.6505,     Adjusted R-squared: 0.6413
F-statistic: 0.74 on 2 and 76 DF,  p-value: < 2.2e-16
```

Note two effects of the MCAR missing data: First, much “power” is lost, as estimation now based on only 76 DF, rather than the 197 we had before, i.e., only 79 subjects were used in the analysis. Thus, standard errors have increased considerably. Age still reasonably well estimated, but not sex. 95% CI still easily includes both effects of age and sex, however, so no evidence of bias in the analysis.

In general, under MCAR, expect loss of power, but no bias. We can do a similar analysis, but with MAR rather than MCAR. In MCAR, the missing data probabilities are completely unrelated to the values of the missing items or any other values in the data set. In contrast, for MAR missing data, the probability that an item is missing

can be related to values, but only *observed* values, not unobserved values. The idea is that under MAR data, one can use observed values to "recapture" the essence of the missing data, and so still derive valid inferences.

Both MCAR and MAR missing data mechanisms are termed *ignorable*, because conditional on the observed data set, one can derive valid inferences. Missing data mechanisms are termed *non-ignorable* if the missing data can depend on unobserved values, so that even conditioning on the observed data does not produce valid inferences.

In practice, one can never know if missing data mechanisms are ignorable or not, so it is wise to present several analyses considering both cases, and investigate robustness of inferences to various assumptions. Again, see paper by Kmetic et al. for an example of this.

Going back to our MCAR data, note that CRP itself is still well estimated, in terms of its mean, but with increased SE:

```
> t.test(crp)
95 percent confidence interval:
 17.17438 17.59874
sample estimates: mean of x
 17.38656

> t.test(crp[mcar.for.age==0 & mcar.for.sex==0 ])
95 percent confidence interval:
 16.96929 17.65130
sample estimates: mean of x
 17.31030
```

Note that the width of the CI went from about 0.42 (no missing data) to 0.68 with the missing data, an increase of more than 50%, but there was no bias, the mean being about 17.3 in both cases.

To simulate a data set that is MAR, we need to delete missing data not at random, but in a systematic and biased fashion. To remain ignorable, we need to delete values such that the bias can (more or less, in real practice) be corrected with the observed data at hand. So, what happens if we delete CRP values from higher age values, and delete more CRP values from males compared to females? Clearly, this will cause a bias in estimated CRP values (but not necessarily in the regression values, as regression coefficients are the same regardless of age, sex, or CRP values). Can we adjust back to recapture the "true" mean values of CRP?

First, let's create a data set with an MAR missing mechanism, where CRP values tend to be missing from persons with higher ages. Age ranges from 50 to 70 in our data set. We will delete CRP values in the range from 50 to 59 with a rate of 10%,

while CRP values from ages in the range 60 to 70 will be deleted with a probability of 60%. In R, we can code this as:

```
> mar.for.crp <- rep(NA, 200) #  Just to create a blank vector
> for(i in 1:200)
+  {
+  if(age[i] < 60) { mar.for.crp[i] <- rbinom(1, 1, prob=0.1) }
+  if(age[i] >= 60) { mar.for.crp[i] <- rbinom(1, 1, prob=0.6) }
+ }
> mean(mar.for.crp)
[1] 0.43
```

So our overall rate of deletions is 43%, close to expectations. Our CRP data set now looks like this:

```
> crp.mar <- crp
> crp.mar[mar.for.crp==1] <- NA
> round(crp.mar,2)
  [1]    NA    NA    NA 16.53 17.19 16.59    NA 14.51 16.79 16.47
 [11]    NA    NA 16.94 19.80    NA 16.12    NA    NA 17.12 16.23
 [21] 21.17 19.04 14.46    NA 17.27 15.82 14.98 17.16    NA 14.96
 [31] 16.30    NA 14.05    NA 16.13 17.41    NA 22.40    NA 15.32
 [41]    NA    NA    NA    NA    NA    NA    NA    NA    NA 18.40
 [51] 17.02 19.11 19.20 15.59    NA 17.56 15.48    NA 16.77 17.91
 [61] 18.80    NA    NA 17.87    NA    NA 15.31 15.63    NA 16.90
 [71] 13.87 17.65    NA 17.54 17.30    NA 16.96 15.97 19.31    NA
 [81]    NA    NA 16.23 17.82    NA    NA 16.99 17.73 16.52    NA
 [91] 19.15 16.33 17.32    NA 16.48 17.47    NA 19.13 14.82    NA
[101] 16.80 18.44    NA 16.66 15.90 15.66    NA 16.92    NA 17.79
[111]    NA    NA    NA    NA 16.00 16.01    NA    NA 17.23    NA
[121] 18.08 17.25 14.91 15.23 20.11    NA    NA 13.88    NA 16.09
[131] 17.29 16.88 16.52    NA    NA 14.88    NA 18.79    NA    NA
[141] 18.02 18.36 17.04    NA 16.32 15.47 13.41    NA 18.60    NA
[151]    NA 16.47 17.45    NA 16.42 16.11 15.54    NA    NA 15.06
[161] 18.15    NA 15.25 16.01 17.06    NA 17.29 17.16    NA 18.60
[171]    NA 17.92 15.34 19.19    NA 18.31    NA 17.77    NA    NA
[181] 15.41    NA    NA    NA    NA 18.29 16.20    NA 19.16    NA
[191] 15.84 18.59    NA    NA    NA    NA    NA 16.93 17.24 16.34
```

Note that we now have very biased estimation of CRP, if no adjustments are made:

```
> t.test(crp.mar)
```

```
95 percent confidence interval:
 16.65344 17.21744
sample estimates:
mean of x
 16.93544
```

Note that the mean point estimate is quite far from the"true" value of 17.39, and even the 95% CI misses the true value by quite a large margin.

So, here we have seen two problems associated with missing data: Lack of precision is estimated values due to lower sample size, and biased estimation due to data not being missing completely at random.

We will now see how multiple imputation can solve both of these problems, as well as seeing how easy it is to program multiple imputation is WinBUGS.

In anticipation of using WinBUGS, we will now prepare our data set for use in WinBUGS:

```
crp.list <- list(crp=crp, age=age, sex=sex, crp.mar=crp.mar,
age.mcar = age.mcar, sex.mcar = sex.mcar)
> dput(crp.list,
   file="c://lawrence//work//courses//677//notes//missing data//crp.txt")
```

You may need to slightly edit this file, but crp.txt is essentially ready for importation of the data into WinBUGS.

## Multiple Imputation

Multiple imputation is a very easy technique, both in theory and in practice. The basic idea is to fill in the missing data with a "best guess", based on any information at hand. Here, for example, we assume a relationship between age, CRP, and sex, so any of these can be predicted from a model using the other two.

Of course, any predictions will not be perfectly correct. That is why single imputation does not work well, as it assumes you know more about the missing data than you really do. Thus, any inferences are "too accurate", with SD's that are too small. In contrast, multiple imputation predicts several missing values for each missing item, creating several "complete" data sets. In each of these *different* "complete" data sets, an analysis is carried out. Final inferences are then created by averaging each of these separate analyses, and the mistake of "too accurate" inferences is avoided by

summing the variance within each analysis with the variance of parameter estimates across each analysis.

**Basic multiple imputation steps are:**

1. Make a model that predicts every missing data item from whatever other information is at hand that can be useful for that prediction. Model can be linear regression, logistic regression, non-linear models, etc, anything goes.
2. Use the above models to create a "complete" data set. Since no model is perfect, imputing each single item requires a two-step process: First, draw a set of parameter values from the models to be used for prediction, second, use those parameter values to make a prediction of any missing data.
3. Each time you create a "complete" data set, do an analysis of that complete data set, keeping the mean and SE of each parameter of interest.
4. Repeat this anywhere between 2 and tens of thousands of times (two or four is often enough, but need to worry about convergence if WinBUGS is used, so often do many thousands).
5. To form final inferences, for each repetition (iteration in WinBUGS), average across means, and sum the within and between variances for each parameter. WinBUGS essentially does this automatically, this step is only needed if programming the entire process yourself in another program (and, of course, conceptually).

WinBUGS performs multiple imputation as the default analysis, as we will now see. We will run several WinBUGS programs, as follows:

1. Using all of the data, first show that WinBUGS can more or less exactly recreate the R regression analysis of the full data set.
2. Using the age.mcar and sex.mcar data sets, show that not too much precision is lost if multiple imputation is used, compared to ignoring the missing data (i.e., case deletion, which is the default in R, SAS, SPSS, Stata, and almost every other statistics package except WinBUGS).
3. In using the crp.mar data, show that both bias is reduced and accuracy is gained when using multiple imputation compared to an analysis that ignores the missing data.

# Multiple Imputation in WinBUGS

Here is a WinBUGS program (and the results) for regression of all the data (note that I have edited the data saved from R to just use the items needed in this program, and deleted the rest):

```
model
{
for (i in 1:200)
{
crp.mean[i] <-  alpha + beta.age*age[i] + beta.sex*sex[i]
crp[i] ~ dnorm(crp.mean[i], tau.crp)
}
alpha ~ dnorm(0, 0.0001)
beta.age ~ dnorm(0, 0.0001)
beta.sex ~ dnorm(0, 0.0001)
tau.crp <- 1/(sd.crp*sd.crp)
sd.crp ~ dunif(0.00001, 10)
}

#  Inits

list(alpha = 0, beta.sex = 0, beta.age = 0, sd.crp = 5)

# Data

list(crp = c(15.4973371597009, 17.6877628162003, 17.7975848672349,
16.5265487264035, 17.1912821304663, 16.5865348180380, 19.2341946673035,
14.5100078038022, 16.7887532493483, 16.4676879355468, 17.0602948092325,
17.1975083550762, 16.9447418113014, .....etc...))
```

| node | mean | sd | MC error | 2.5% | median | 97.5% | start | sample |
|---|---|---|---|---|---|---|---|---|
| alpha | 5.214 | 0.7407 | 0.006965 | 3.745 | 5.222 | 6.66 | 1001 | 10000 |
| beta.age | 0.1971 | 0.01204 | 1.156E-4 | 0.1735 | 0.197 | 0.2209 | 1001 | 10000 |
| beta.sex | 0.4595 | 0.1411 | 0.001432 | 0.1815 | 0.4592 | 0.7366 | 1001 | 10000 |
| sd.crp | 0.9926 | 0.05103 | 5.17E-4 | 0.8999 | 0.9898 | 1.099 | 1001 | 10000 |
| tau.crp | 1.023 | 0.1043 | 0.001037 | 0.8276 | 1.021 | 1.235 | 1001 | 10000 |

As expected, we see that the results here very closely match (almost identical) those from the standard regression results we saw above from R. Next, let's compare the

results from the regression with and without multiple imputation when the data are MCAR:

Here is the WinBUGS program and results:

```
model
{
for (i in 1:200)
{
#  Basic Regression

crp.mean[i] <-  alpha.crp + beta.age*age.mcar[i] +
                beta.sex*sex.mcar[i]
crp[i] ~ dnorm(crp.mean[i], tau.crp)

#  Regression for Imputing Missing Age Variables

age.mean[i] <-  alpha.age + beta.age.crp*crp[i]
age.mcar[i] ~ dnorm(age.mean[i], tau.age)

# Logistic Regression for Imputing Missing Sex Variables

sex.mcar[i] ~ dbern(p.sex[i])
logit(p.sex[i]) <- alpha.sex + beta.sex.crp * crp[i]

}
alpha.crp ~ dnorm(0, 0.0001)
beta.age ~ dnorm(0, 0.0001)
beta.sex ~ dnorm(0, 0.0001)
tau.crp <- 1/(sd.crp*sd.crp)
sd.crp ~ dunif(0.00001, 10)

alpha.age ~ dnorm(0, 0.0001)
beta.age.crp ~ dnorm(0, 0.0001)
tau.age <- 1/(sd.age*sd.age)
sd.age ~ dunif(0.00001, 20)

alpha.sex ~ dnorm(0, 0.01)
beta.sex.crp ~ dnorm(0, 0.01) }

#  Inits

list(alpha.crp = 0, beta.sex = 0, beta.age = 0, sd.crp = 5,
```

```
alpha.age = 0, beta.age.crp = 0, alpha.sex = 0, beta.sex.crp = 0,
sd.age = 5)

# Data

list(crp = c(15.4973371597009, 17.6877628162003, 17.7975848672349,
16.5265487264035, 17.1912821304663, 16.5865348180380,
19.2341946673035, 14.5100078038022, 16.7887532493483, ...etc...
16.3402398342744), age.mcar = c(58, 68, NA, 59, 62, 60, 69, NA,
53, 56, 55, 65, NA, 69, NA, 56, 66, NA, NA, NA, 68, 62, 52, NA,
NA, 60, 51, NA, 63, NA, ...etc..., NA, 61, 54), sex.mcar = c(1,
NA, 1, 1, 1, NA, 1, 0, NA, 1, 0, NA, NA, 0, 0, 1, NA, 1, 0, NA,
NA, 0, 0, 1, 0, NA, NA, 0, 1, ...etc... 0, NA, 0, 1))
```

| node | mean | sd | MC error | 2.5% | median | 97.5% | start | sample |
|---|---|---|---|---|---|---|---|---|
| alpha.age | 5.827 | 3.011 | 0.02805 | -0.1028 | 5.864 | 11.69 | 1001 | 20000 |
| alpha.crp | 3.928 | 0.6937 | 0.007073 | 2.571 | 3.928 | 5.273 | 1001 | 20000 |
| alpha.sex | -3.882 | 2.192 | 0.1676 | -8.545 | -3.855 | 0.5329 | 1001 | 20000 |
| beta.age | 0.2184 | 0.01138 | 1.25E-4 | 0.1962 | 0.2184 | 0.2406 | 1001 | 20000 |
| beta.age.crp | 3.179 | 0.1726 | 0.001608 | 2.843 | 3.177 | 3.518 | 1001 | 20000 |
| beta.sex | 0.2514 | 0.1663 | 0.004195 | -0.08144 | 0.2553 | 0.5685 | 1001 | 20000 |
| beta.sex.crp | 0.2145 | 0.1259 | 0.009613 | -0.03951 | 0.2134 | 0.4823 | 1001 | 20000 |
| sd.age | 3.195 | 0.1798 | 0.001423 | 2.864 | 3.187 | 3.566 | 1001 | 20000 |
| sd.crp | 0.8311 | 0.04514 | 4.432E-4 | 0.7481 | 0.8294 | 0.9263 | 1001 | 20000 |
| tau.age | 0.09892 | 0.01108 | 8.638E-5 | 0.07866 | 0.09842 | 0.122 | 1001 | 20000 |
| tau.crp | 1.46 | 0.1575 | 0.001509 | 1.165 | 1.454 | 1.787 | 1001 | 20000 |

Note that all true parameter values of the original regression are well within their 95% credible intervals, and in this case the results are not all that much different compared to simply leaving out all cases with missing data. This is not surprising, given that the data were missing completely at random, and an exact model was used for the MCAR analyses. In other cases there will be more gained, especially if the data are NOT completely missing at random (see this week's assignment for an example of this phenomenon).

One could argue in the above example that multiple imputation was hardly worth the trouble, and this sometimes is the case with MCAR data. However, now consider estimating the mean value of CRP in the population, using the CRP data that were MAR, rather than MCAR. Recall that standard analyses, ignoring the missing data,

did very badly in this case, with the 95% interval missing the true parameter value.

```
model
{
mean.crp <- mean(crp.mean[])

for (i in 1:200)
{

#  Basic Regression

crp.mean[i] <-  alpha.crp + beta.age*age[i] + beta.sex*sex[i]
crp.mar[i] ~ dnorm(crp.mean[i], tau.crp)

}
alpha.crp ~ dnorm(0, 0.01)
beta.age ~ dnorm(0, 0.01)
beta.sex ~ dnorm(0, 0.01)
tau.crp <- 1/(sd.crp*sd.crp)
sd.crp ~ dunif(0.01, 10)

}

#  Inits

list(alpha.crp = 0, beta.sex = 0, beta.age = 0, sd.crp = 1)

# Data

list(age = c(58, 68, 61, 59, 62, 60,
69, 50, 53, 56, 55, 65, 55, 69, 70, 56, 66, 56, 64, 52, 68, 62,
52, 66, 54, 60, 51, 50, ...etc...
61, 54),
sex = c(1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
1, 0, 1, 0, 1, 1, ...etc...1, 1, 0, 0, 0, 1),
crp.mar=c(NA, NA, NA, 16.5265487264035, 17.1912821304663, 16.5865348180380,
NA, 14.5100078038022, 16.7887532493483, 16.4676879355468, NA,
NA, 16.9447418113014, 19.7999455207252, NA, 16.1233945211970,
NA, NA, 17.1211685477801, 16.2339325019515, 21.1724598527071,
19.0446227087453, ....etc....  18.5890535269482,
NA, NA, NA, NA, NA, 16.9331262768319, 17.2364188500370, 16.3402398342744
))
```

The results are:

| node | mean | sd | MC error | 2.5% | median | 97.5% | start | sample |
|---|---|---|---|---|---|---|---|---|
| alpha.crp | 5.391 | 0.9821 | 0.01003 | 3.465 | 5.384 | 7.305 | 1001 | 10000 |
| beta.age | 0.195 | 0.01665 | 1.7E-4 | 0.162 | 0.1952 | 0.2272 | 1001 | 10000 |
| beta.sex | 0.2857 | 0.1923 | 0.001943 | -0.09571 | 0.2856 | 0.6611 | 1001 | 10000 |
| mean.crp | 17.35 | 0.1027 | 0.00101 | 17.15 | 17.35 | 17.55 | 1001 | 10000 |
| sd.crp | 1.028 | 0.07162 | 7.522E-4 | 0.8986 | 1.024 | 1.18 | 1001 | 10000 |
| tau.crp | 0.9604 | 0.1327 | 0.001387 | 0.7179 | 0.9543 | 1.239 | 1001 | 10000 |

Note the near perfect reult for mean CRP, 17.35, 95% CrI (17.15, 17.55), and compare it to the non-imputed result we had before, of 16.93, 95% CI (16.65, 17.21), which missed the true value completely. The imputed value even compares very well to the value with NO missing data, which was 17.39, 95% CI (17.17, 17.60). Note that the imputed interval is *narrower* than when the whole data set was used, even though more than 40% of the data went missing!! This has occurred because the model was MAR, so assumptions behind multiple imputation were perfectly satisfied, and because we used a good model.

## Concluding Comments

We have seen two extremes of the use of multiple imputation: In the first case, not much was gained, as with MCAR data the case deletion method did quite well. In the second case multiple was extremely useful, adjusting for bias in the MAR data, AND increasing precision beyond what was in the original data set by itself. Conclusion: While multiple imputation is sometimes useful, it can also sometimes do worse than other methods. Further, except for simulated data sets, one cannot usually tell which sort of situation we are in.

All of the above have assumed MCAR or MAR missing data mechanisms. Under MCAR or MAR, the missing data mechanism is "ignorable", meaning in practice that the data analyst can derive valid inferences from the data, provided a technique like multiple imputation is used. The following article discusses these concepts in further detail, and demonstrates something the analyst can do in the case of non-ignorable missing data. In practice, one never knows if data are MAR or not, so checking assumptions is very important.