Missing data Imputation

Regression Substitution Method

- We use the complete data points to calculate the regression of the incomplete variable on the other complete variables
- Then we substitute the predicted mean for each unit with a missing value
- In this way we use information from the joint distribution of the variables to make the imputation
- Regression mean imputation can generate unbiased estimates of means, associations and regression coefficients in a much wider range of settings than simple mean imputation
- However, one important problem remains. The variability of the imputations is too small, so the estimated precision of regression coefficients will be wrong and inferences will be misleading

Regression Imputation: An Example

- We consider the *fitness* data set.
- The variable Oxygen has complete data
- The variable RunTime has three observations missing
- The variable RunPulse has three observations (4, 11, 14) missing together with RunTime and five on its own (5, 8, 18, 19, 25)
- So we develop three regression lines as follows:
  - RunTime on Oxygen to predict missing observations 4, 11, 14
  - RunPulse on Oxygen to predict missing observations 4, 11, 14
  - RunPulse on Oxygen and RunTime to predict missing observations 5, 8, 18, 19, 25

Regression Imputation Example: R Output

```
        1         2         3
 7.733491  9.827755 10.585679

       1        2        3
159.1726 167.2775 170.2106

       1        2        3        4        5
168.7270 167.9277 171.5581 171.1831 172.2053
```

Completed Data Set

```
predictOxygen RunTime RunPulse
44.609 11.37 178
45.313 10.07 185
54.297  8.65 156
59.571 7.73  159
49.874  9.22 169
44.811 11.63 176
45.681 11.95 176
49.091 10.85 168
39.442 13.08 174
60.055  8.63 170
50.541  9.82 167
37.388 14.03 186
44.754 11.12 176
47.273 10.59 17
51.855 10.33 166
49.156  8.95 180
40.836 10.95 168
46.672 10.00 172
46.774 10.25 171
50.388 10.08 168
39.407 12.63 174
46.080 11.17 156
45.441  9.63 164
```

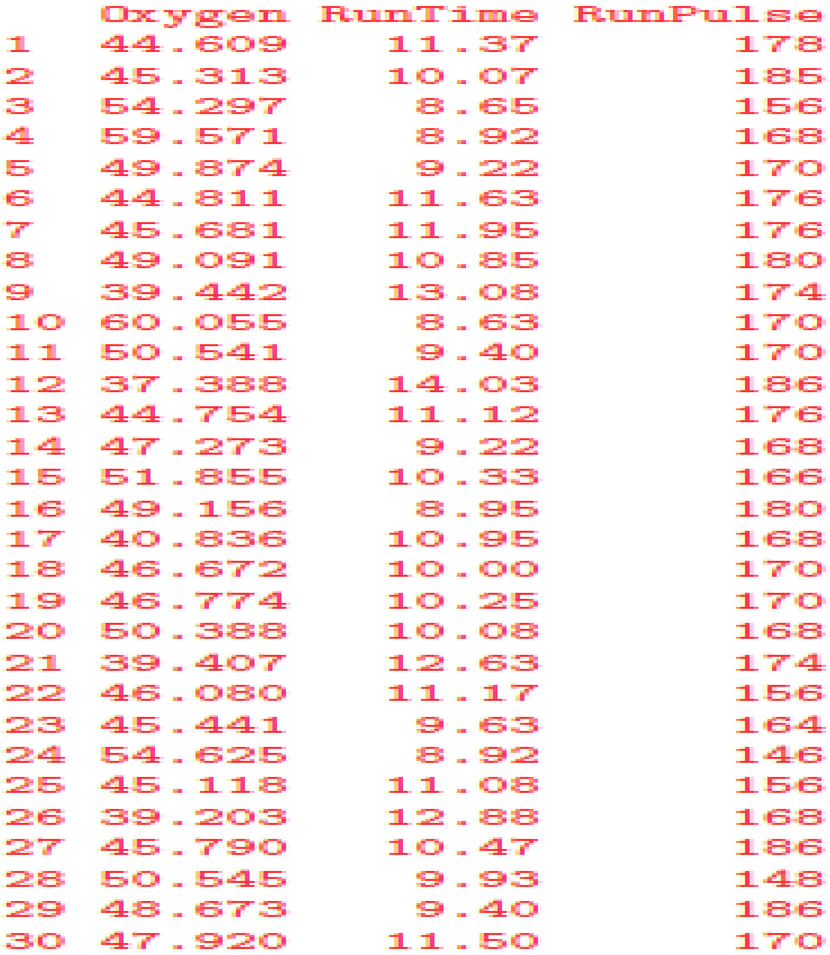
k-Nearest Neighbor Approach

- Another way of dealing with missing data is the k nearest neighbor (knn) approach
- This method is quite simple in principle but is effective and often preferred over some of the more sophisticated methods described above
- Nearest neighbors are records that have similar completed data patterns; the average of the k-nearest neighbors' completed data are used to impute the value for a variable that is missing its value
- k can be set by the analyst
- It has been shown that a k ranging from 5 to 10 is adequate
- The advantage of the knn approach is that it assumes data are missing at random (MAR) meaning, missing data only depends on the observed data; which in turn means, the knn approach is able to take advantage of multivariate relationships in the completed data
- The disadvantage of this approach is it does not include a component to model random variation; consequently uncertainty in the imputed value is underestimated

Multiple Imputation

- An additional method for imputing values for missing observations is known as multiple imputation (MI)
- There are a number of ways of performing MI, though they all involve the use of random components to overcome the problem of underestimation of standard errors
- The parameter estimates using this approach are nearly unbiased
- The interesting thing about MI is that the word "multiple" refers not to the iterative nature of the process involved in imputation, but to the fact that we impute multiple complete data sets and run whatever analysis is appropriate on each data set in turn
- We then combine the results of those multiple analyses using fairly simple rules
- In a way it is like running multiple replications of an experiment and then combining the results across the multiple analyses
- But in the case of MI, the replications are repeated simulations of data sets based upon parameter estimates from the original study

```
iter imp variable
 1   1  RunTime  RunPulse
 1   2  RunTime  RunPulse
 1   3  RunTime  RunPulse
 1   4  RunTime  RunPulse
 1   5  RunTime  RunPulse
 2   1  RunTime  RunPulse
 2   2  RunTime  RunPulse
 2   3  RunTime  RunPulse
 2   4  RunTime  RunPulse
 2   5  RunTime  RunPulse
```



```
     Oxygen RunTime RunPulse
1    44.609   11.37      178
2    45.313   10.07      185
3    54.297    8.65      156
4    59.571    8.92      168
5    49.874    9.22      170
6    44.811   11.63      176
7    45.681   11.95      176
8    49.091   10.85      180
9    39.442   13.08      174
10   60.055    8.63      170
11   50.541    9.40      170
12   37.388   14.03      186
13   44.754   11.12      176
14   47.273    9.22      168
15   51.855   10.33      166
16   49.156    8.95      180
17   40.836   10.95      168
18   46.672   10.00      170
19   46.774   10.25      170
20   50.388   10.08      168
21   39.407   12.63      174
22   46.080   11.17      156
23   45.441    9.63      164
24   54.625    8.92      146
25   45.118   11.08      156
26   39.203   12.88      168
27   45.790   10.47      186
28   50.545    9.93      148
29   48.673    9.40      186
30   47.920   11.50      170
```