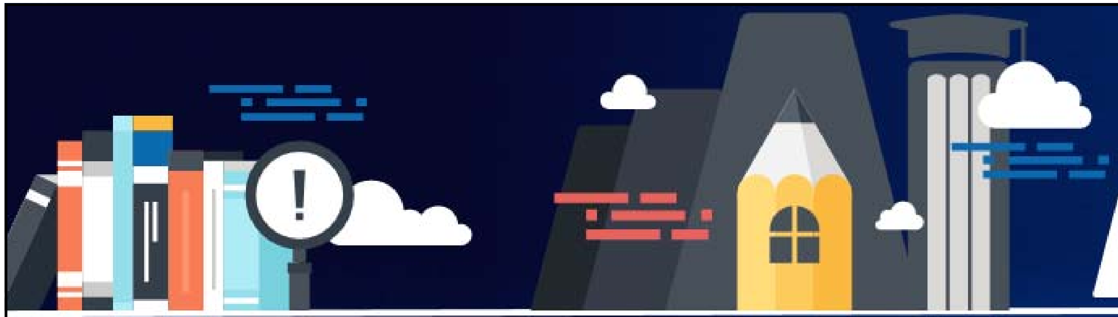
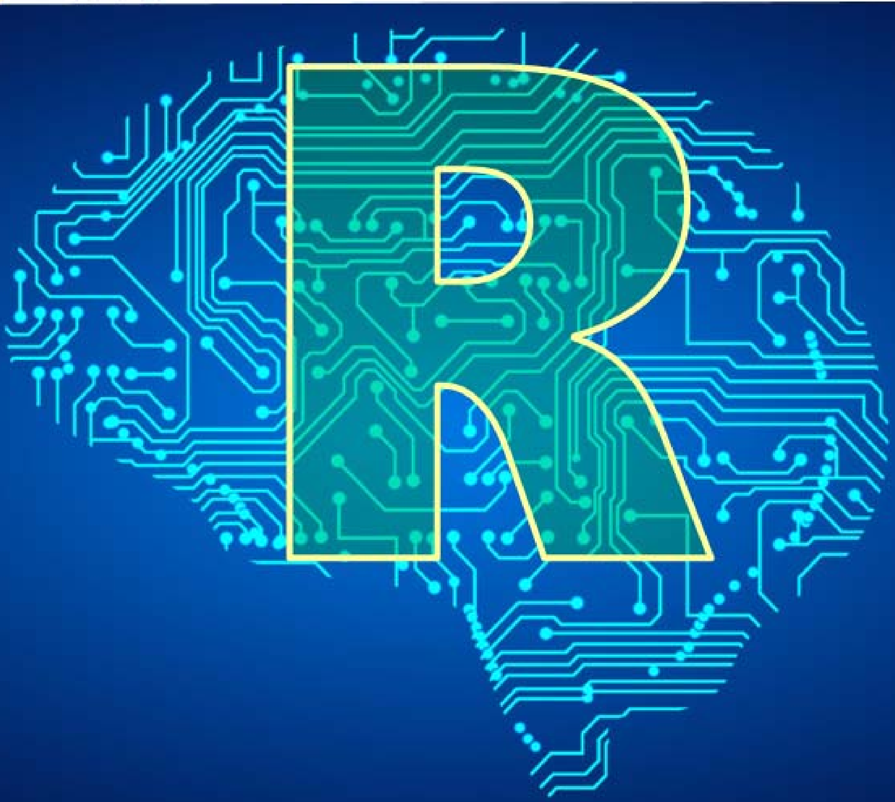


D2-2

台灣人工智慧學校



遺失值處理

吳漢銘

國立臺北大學 統計學系

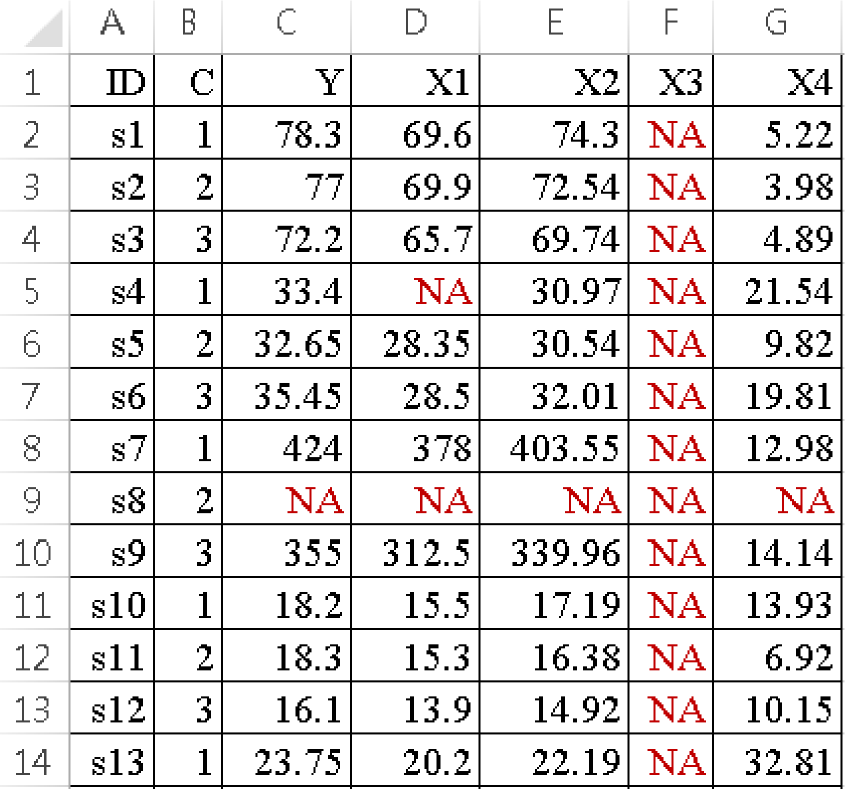
http://www.hmwu.idv.tw

本章大綱

- 遺失值 (Missing Data)
- **Missingness Mechanism**
  - Missing by Design，Missing Completely at Random，Missing at Random (MAR), Missing Not at Random (MNAR)，
- **Missing Values in R**
- **Traditional Approaches to Handling Missing Data**
- **Advanced Imputation Methods**
- **R Packages for Dealing With Missing Values**
  - **VIM, MICE**，Amelia，mi, Hmisc

遺失值 (Missing Data)

- When data are missing for a variable for all cases: **latent** or **unobserved.**
- When data are missing for all variables for a given case: **unit non-response.**
- Missing data (missing values for certain variables for certain cases): **item non-response.**



	A	B	C	D	E	F	G
1	ID	C	Y	X1	X2	X3	X4
2	s1	1	78.3	69.6	74.3	NA	5.22
3	s2	2	77	69.9	72.54	NA	3.98
4	s3	3	72.2	65.7	69.74	NA	4.89
5	s4	1	33.4	NA	30.97	NA	21.54
6	s5	2	32.65	28.35	30.54	NA	9.82
7	s6	3	35.45	28.5	32.01	NA	19.81
8	s7	1	424	378	403.55	NA	12.98
9	s8	2	NA	NA	NA	NA	NA
10	s9	3	355	312.5	339.96	NA	14.14
11	s10	1	18.2	15.5	17.19	NA	13.93
12	s11	2	18.3	15.3	16.38	NA	6.92
13	s12	3	16.1	13.9	14.92	NA	10.15
14	s13	1	23.75	20.2	22.19	NA	32.81

遺失值的處理

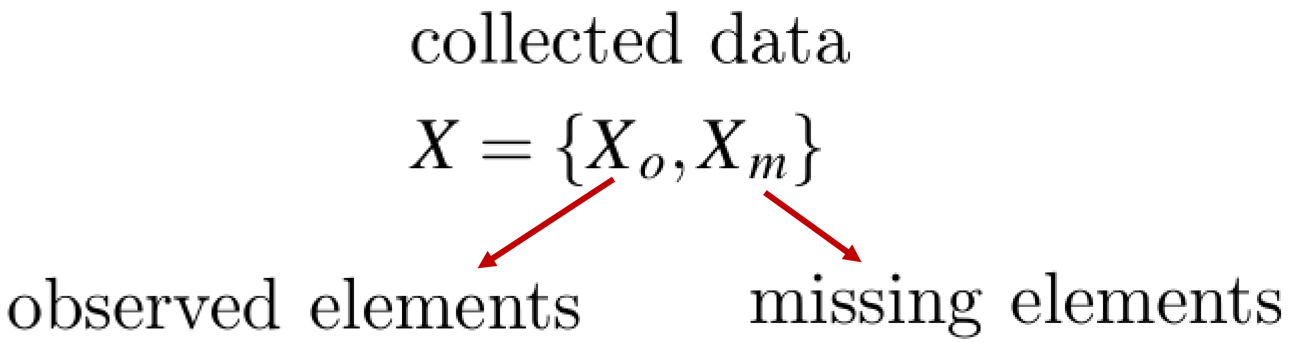
- The missing values may give clues to systematic aspects of the problem. Ignore the tuple, you cannot make use of the remaining values except the missing one.

- **How to deal with missing values:**
  - Use a global constant to fill the value will misguide the mining process.
  - Use a measure for a central tendency for the attribute to fill the missing value for symmetric data distribution.
  - Use the attribute mean or median for all samples belonging to the same class as the given tuple.
  - **補值 (Missing value imputation) (most popular)**

Missingness Mechanism

- The presence of missing data can
  - effect the properties of the estimates (e.g. means, percentages, percentiles, variances, ratios, regression parameters, etc.).
  - affect inferences, (e.g., the properties of tests and confidence intervals.)
- The missingness mechanism (Little and Rubin, 1987)
  - The way in which the probability of an item missing depends on other observed or non-observed variables as well as on its own value.
- It helpful to classify missing values on the basis of the stochastic mechanism that produces them.

Missingness Mechanism



collected data

$$X = \{X_o, X_m\}$$

observed elements

missing elements

The missingness indicator matrix R corresponds X,

and each element of R is 1 if the corresponding element of X is missing,

and 0 otherwise.

define the missingness mechanism as

the probability of R conditional on

the values of the observed and missing elements of X:

$$Pr(R|X_o, X_m)$$

(M1) Missing by Design

- **Excluded** some participants from the analysis because they are not part of the population under investigation.
  - Eg., valid skips: when a question is not answered because it is not applicable to the given unit.

- In many surveys different missingnes codes are applied indicating the reason why the respondent did not provide an answer:
  - (i) refused to answer; (ii) answered don't know; (iii) had a valid skip or (iv) was skipped by an enumerator error.
  - Depending on the code one can decide whether the corresponding values are to be imputed or not.

(M2) Missing Completely at Random

- Missing Completely at Random (MCAR)
  - missingness is independent of their own unobserved values and the observed data.
  - the pattern of missing values is totally random and does not depend on any variable, which may or may not be included in the analysis.

$$Pr(R|X) = Pr(R)$$

- *Example*: Miscoding or forgetting to log in answer.

- For most data sets, the MCAR assumption is unlikely to be satisfied, one exception being the case when data are missing by design.
- For MCAR, no bias is introduced when omitting those observations with missing values.
- **Imputation methods** rely on the missingness being of the MCAR type.

(M3) Missing at Random (MAR)

- **MAR:** missingness does not depend on their unobserved value but does dependent on the observed data.

$$Pr(R|X) = Pr(R|X_o)$$

遺失狀況跟觀察變數有關: 例如問男生是否沮喪? 可能不回回答
問女生體重?可能不會回答

- *Example 1*: **male participants (observed data) are more likely to refuse to fill out the depression survey, but it does not depend on the level of their depression (unobserved value).**
- *Example 2*: **if men are more likely to tell you their weight than women, weight is MAR.**

- MAR can never be tested on any given data set because it can be that some unobserved variables are causing the missing pattern.
- MCAR is a special case of MAR.
- We can ignore missing data (= omit missing observations) if we have MAR or MCAR.

(M4) Missing Not at Random (MNAR)

- MNAR means that an unknown process is generating the missing values.
  - Missingness that depends on unobserved predictors
  - Missingness that depends on the missing value itself.

故意的遺失:跟遺失本身有相關，例如薪水高低

- *Example*: question about **income**, where the high rate of missing values (usually 20%~50%) is related to the value of the income itself (very high and very low values will not be answered).

(M4) Missing Not at Random (MNAR)

- MNAR data is a more serious issue.
  - Check the data gathering process further and try to understand why the information is missing.
  - Eg., if most of the people in a survey did not answer a certain question, why did they do that? Was the question unclear?

- MNAR: the missing-data mechanism is not ignorable, and a valid estimation requires the missing-data mechanism to be modeled as part of the estimation process. The results can be very sensitive to the model choice (Little and Rubin, 1987).

Some Notes

- Assuming data is **MCAR**, too much missing data can be a problem.
  - Usually a safe maximum threshold is 5% of the total for large datasets.
  - If missing data for a certain feature or sample is more than 5% then you probably should leave that feature or sample out.

- If some variable is missing almost 25% of the datapoints.
  - Consider either dropping it from the analysis or gather more measurements.
  - Keep the other variables are below the 5% threshold.

- For categorical variables, replacing categorical variables is usually not advisable. Some common practice include replacing missing categorical variables with the mode of the observed ones (questionable).

Missing Values in R

- **NA**: a missing value ("not available"), **"NA"**: a string.
- **x[1]== NA** is not a valid logical expression and will not return **FALSE** as one would expect but will return **NA**.

```
> myvector <- c(10, 20, NA, 30, 40)
> myvector
[1] 10 20 NA 30 40
> mycountry <- c("Austria", "Australia", NA, NA, "Germany", "NA")
> mycountry
[1] "Austria"   "Australia" NA          NA          "Germany"   "NA"
> is.na(myvector)
[1] FALSE FALSE  TRUE FALSE FALSE
> which(is.na(myvector))
[1] 3
> x <- c(1, 4, 7, 10)
> x[4] <- NA # sets the 4th element to NA
> x
[1]  1  4  7 NA
> is.na(x) <- 1 # sets the first element t
> x
[1] NA  4  7 NA
```

```
#Recoding Values to Missing
mydata$v1[mydata$v1==99] <- NA
```

```
> set.seed(12345)
> mydata <- matrix(round(rnorm(20), 2), ncol=5)
> mydata[sample(1:20, 3)] <- NA
> mydata
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,]  0.59  0.61    NA  0.37    NA
[2,]  0.71 -1.82 -0.92  0.52 -0.33
[3,] -0.11  0.63 -0.12 -0.75  1.12
[4,] -0.45 -0.28  1.82    NA  0.30
> which(colSums(is.na(mydata)) > 0)
[1] 3 4 5
```

NOTE: **NULL** denotes something which never existed and cannot exist at all.

NA in Summary Functions

- Most of the statistical summary functions (`mean, var, sum, min, max`, etc.) accept an argument called `na.rm`, which can be set to `TRUE` if you want missing values to be removed before the summary is calculated. (default : `FALSE`)

- For functions that don't provide such an argument, the negation operator (`!`) can be used in an expression like `x[!is.na(x)]` to create a vector which contains only the nonmissing values in `x`.

```
> x <- c(1, 4, NA, 10)
>  summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
    1.0     2.5     4.0     5.0     7.0    10.0       1
>  mean(x)
[1] NA
>  sd(x)
[1] NA
>  mean(x, na.rm=TRUE)
[1] 5
>  sd(x, na.rm=TRUE)
[1] 4.582576
> x[!is.na(x)]
[1]  1  4 10
```

移除遺失值

NA in Modeling Functions

- The statistical modeling functions (**`lm, glm, gam`**, etc.) all have an argument called **`na.action=`**, which allows you to specify a function that will be applied to the data frame specified by the **`data=`** argument before the modeling function processes the data.
  - **`na.fail()`** - issue an error if the object contains missing values.
  - **`na.omit()`** - exclude the missing values and return the rest of the object. (The **`complete.cases`** function may also be useful to achieve the same task)
  - **`na.exclude()`** - same as **`na.omit()`**but will result in different behavior of some functions (like **`napredict()`** and **`naresid()`**)
  - **`na.pass()`** - return also the missing values (the object remains unchanged)

NA in Modeling Functions

```
> mydata <- as.data.frame(matrix(sample(1:20, 8), ncol = 2))
> mydata[4, 2] <- NA
> names(mydata) <- c("y", "x")
> mydata
   y  x
1  1 19
2  6 12
3 10  2
4  4 NA
> lm(y~x, data = mydata)

Call:
lm(formula = y ~ x, data = mydata)

Coefficients:
(Intercept)            x
    11.3927      -0.5205

> lm(y~x, data = mydata, na.action = na.omit)

Call:
lm(formula = y ~ x, data = mydata, na.action = na.omit)

Coefficients:
(Intercept)            x
    11.3927      -0.5205

> lm(y~x, data = mydata, na.action = na.fail)
Error in na.fail.default(list(y = c(1L, 6L, 10L, 4L), x = c(19L, 12L,  :
  missing values in object
```

Other Special Values in R

- `NaN` : "not a number" which can arise for example when we try to compute the undeterminate 0/0.
- `Inf` which results from computations like 1/0.
- Using the functions `is.finite()` and `is.infinite()` we can determine whether a number is finite or not.

```
> x <- c(1, 0, 10)
> x/x
[1]   1 NaN   1
> is.nan(x/x)
[1] FALSE  TRUE FALSE
```

```
> 1/x
[1] 1.0 Inf 0.1
> is.finite(1/x)
[1]  TRUE FALSE  TRUE
>
> -10/x
[1]  -10 -Inf   -1
> is.infinite(-10/x)
[1] FALSE  TRUE FALSE
```

```
> exp(-Inf)
[1] 0
> 0/Inf
[1] 0
> Inf - Inf
[1] NaN
> Inf/Inf
[1] NaN
```

R Packages for Dealing With Missing Values

https://cran.r-project.org/web/packages/package-name/

- `Amelia (Amelia II)`: A Program for Missing Data
- `hot.deck:` Multiple Hot-Deck Imputation
- `HotDeckImputation`: Hot Deck Imputation Methods for Missing Data
- `impute`: (Bioconductor) Imputation for Microarray Data
- `mi`: Missing Data Imputation and Model Checking
- `mice`: Multivariate Imputation by Chained Equations
- `missForest:` Nonparametric Missing Value Imputation using Random Forest
- `missMDA`: Handling Missing Values with Multivariate Data Analysis (e.g., imputePCA, imputeMCA,)
- `mitools`: Tools for Multiple Imputation of Missing Data
- `norm:` Analysis of Multivariate Normal Datasets with Missing Values
- `VIM`: Visualization and Imputation of Missing Values
- R packages support for missing values imputation.
  - `Hmisc:` Harrell Miscellaneous
  - `survey`: analysis of complex survey samples
  - `Zelig`: Everyone's Statistical Software
  - `rfImpute{randomForest}`: Imputations by randomForest
  - `imputation{rminer}`: Data Mining Classification and Regression Methods, Missing data imputation (e.g. substitution by value or hotdeck method).
  - `impute.svd{bcv}`: Cross-Validation for the SVD (Bi-Cross-Validation), Missing value imputation via a low-rank SVD approximation estimated by the EM algorithm.
  - `mlr`: Machine Learning in R provides several imputation methods.
    https://mlr-org.github.io/mlr-tutorial/release/html/index.html

Package "`imputation`" was removed from the CRAN. (Archived on 2014-01-14)

R Package: MICE

- `mice`: Multivariate Imputation by Chained Equations in R by Stef van Buuren.

- Imputing missing values on mixed data.
  - Continuous data: Predictive mean matching, Bayesian linear regression, Linear regression ignoring model error, Unconditional mean imputation etc.
  - Binary data: Logistic Regression, Logistic regression with bootstrap
  - Categorical data (More than 2 categories) - Polytomous logistic regression, Proportional odds model etc.
  - Mixed data (Can work for both Continuous and Categorical) - CART, Random Forest, Sample (Random sample from the observed values).

Source: http://www.listendata.com/2015/08/missing-imputation-with-mice-package-in.html

Imputation using MICE Package

```
> head(airquality)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
> dim(airquality)
[1] 153   6
> mydata <- airquality
> mydata[4:10,3] <- rep(NA,7)
> mydata[1:5,4] <- NA
>
> # Use numerical variables as examples here.
> # Ozone is the variable with the most missing datapoints.
> summary(mydata)
     Ozone           Solar.R           Wind             Temp           Month            Day      
 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :57.00   Min.   :5.000   Min.   : 1.0  
 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:73.00   1st Qu.:6.000   1st Qu.: 8.0  
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00   Median :7.000   Median :16.0  
 Mean   : 42.13   Mean   :185.9   Mean   : 9.806   Mean   :78.28   Mean   :6.993   Mean   :15.8  
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00   3rd Qu.:8.000   3rd Qu.:23.0  
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00   Max.   :9.000   Max.   :31.0  
 NA's   :37       NA's   :7       NA's   :7        NA's   :5                                       
```

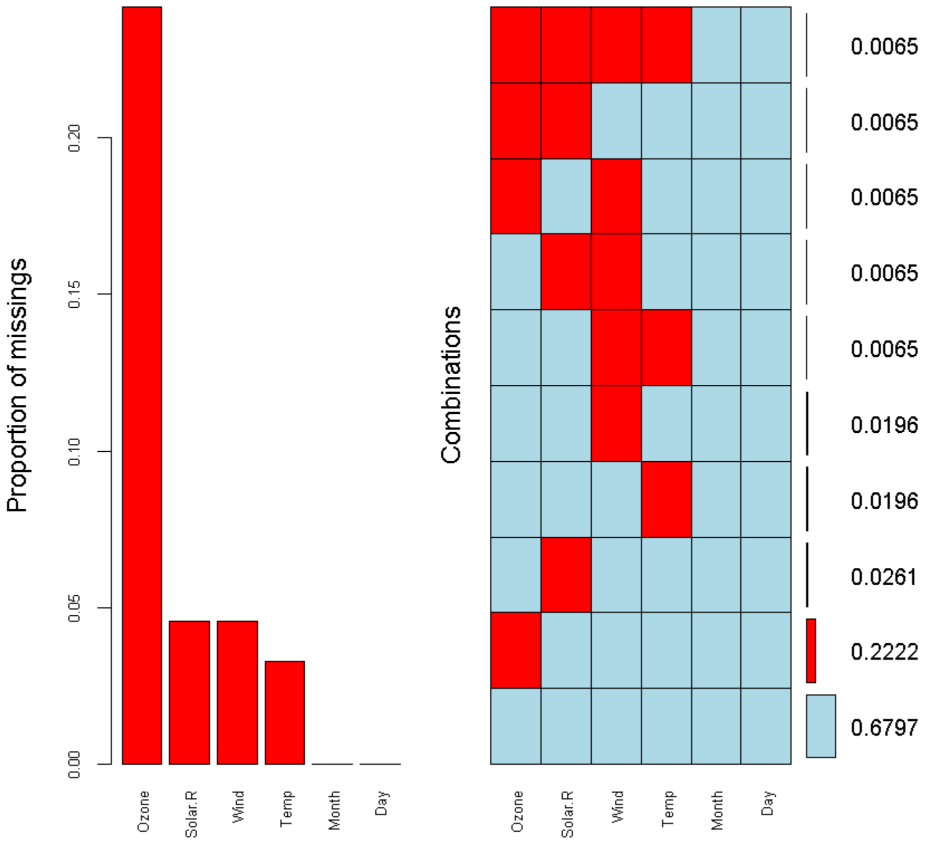
Sourec: http://www.r-bloggers.com/imputing-missing-data-with-r-mice-package/

Visualizing the Pattern of Missing Data

```
> library(mice)
> md.pattern(mydata)
    Month Day Temp Solar.R Wind Ozone
104     1   1    1       1    1     1  0
 34     1   1    1       1    1     0  1
  4     1   1    1       0    1     1  1
  3     1   1    1       1    0     1  1
  3     1   1    0       1    1     1  1
  1     1   1    1       0    1     0  2
  1     1   1    1       1    0     0  2
  1     1   1    1       0    0     1  2
  1     1   1    0       1    0     1  2
  1     1   1    0       0    0     0  4
        0   0    5       7    7    37 56
```

```
> library(VIM)
> mydata.aggrplot <- aggr(mydata,
col=c('lightblue','red'), numbers=TRUE,
prop = TRUE, sortVars=TRUE,
labels=names(mydata), cex.axis=.7, gap=3)

 Variables sorted by number of missings:
 Variable      Count
    Ozone 0.24183007
  Solar.R 0.04575163
     Wind 0.04575163
     Temp 0.03267974
    Month 0.00000000
      Day 0.00000000
```

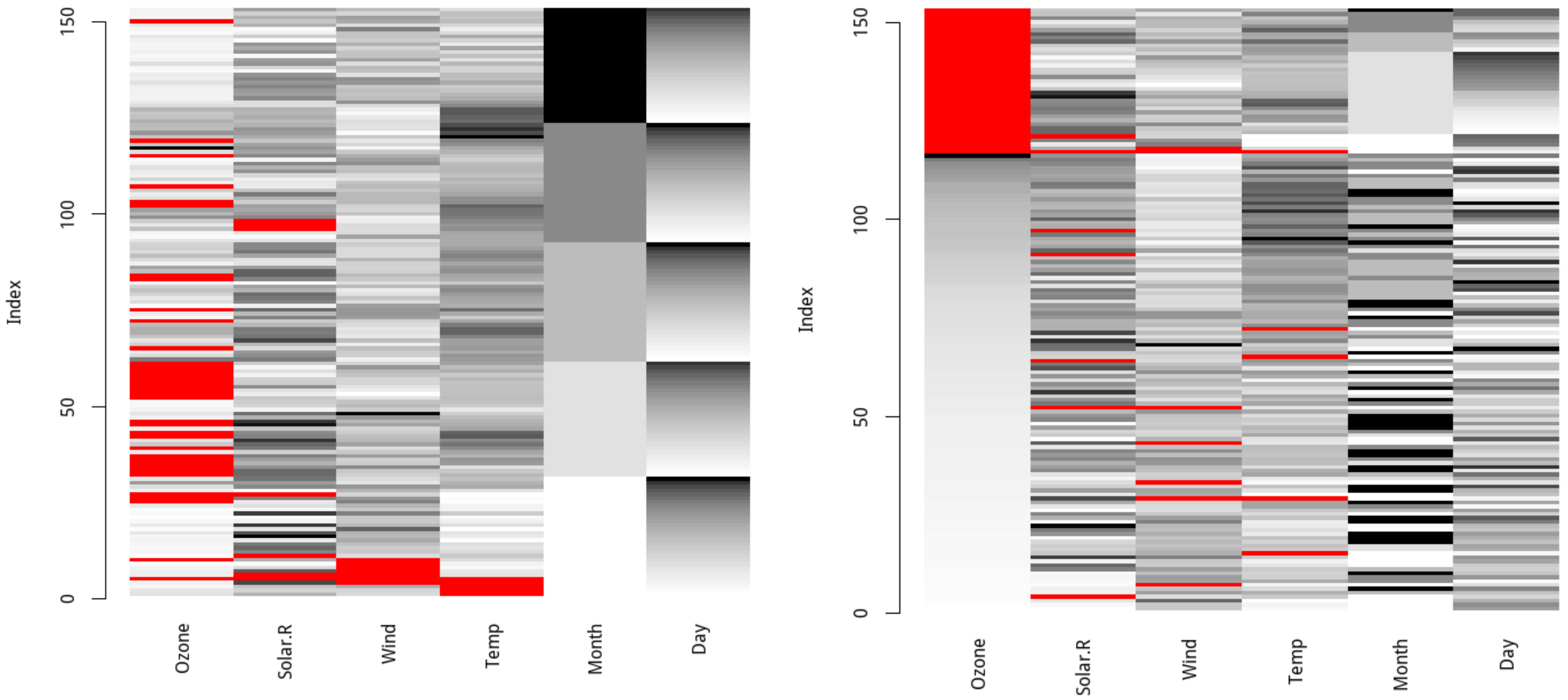
#104 samples are complete, 34 samples miss only the Ozone measurement, 4 samples miss only the Solar.R value and so on.



Aggregation Plot

Matrix Plot

```
> matrixplot(mydata)

Click in a column to sort by the corresponding variable.
To regain use of the VIM GUI and the R console, click outside the plot region.

Matrix plot sorted by variable 'Ozone'.
```

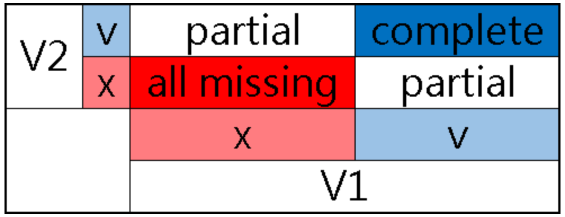


Number of Observations Per Patterns for All Pairs of Variables

```
> md.pairs(mydata)
$rr
        Ozone Solar.R Wind Temp Month Day
Ozone     116     111  111  112   116 116
Solar.R   111     146  141  142   146 146
Wind      111     141  146  143   146 146
Temp      112     142  143  148   148 148
Month     116     146  146  148   153 153
Day       116     146  146  148   153 153

$rm
        Ozone Solar.R Wind Temp Month Day
Ozone       0       5    5    4     0   0
Solar.R    35       0    5    4     0   0
Wind       35       5    0    3     0   0
Temp       36       6    5    0     0   0
Month      37       7    7    5     0   0
Day        37       7    7    5     0   0
```

- **rr**: response-response, both variables are observed
- **rm**: response-missing, row observed, column missing
- **mr**: missing-response, row missing, column observed
- **mm**: missing-missing, both variables are missing

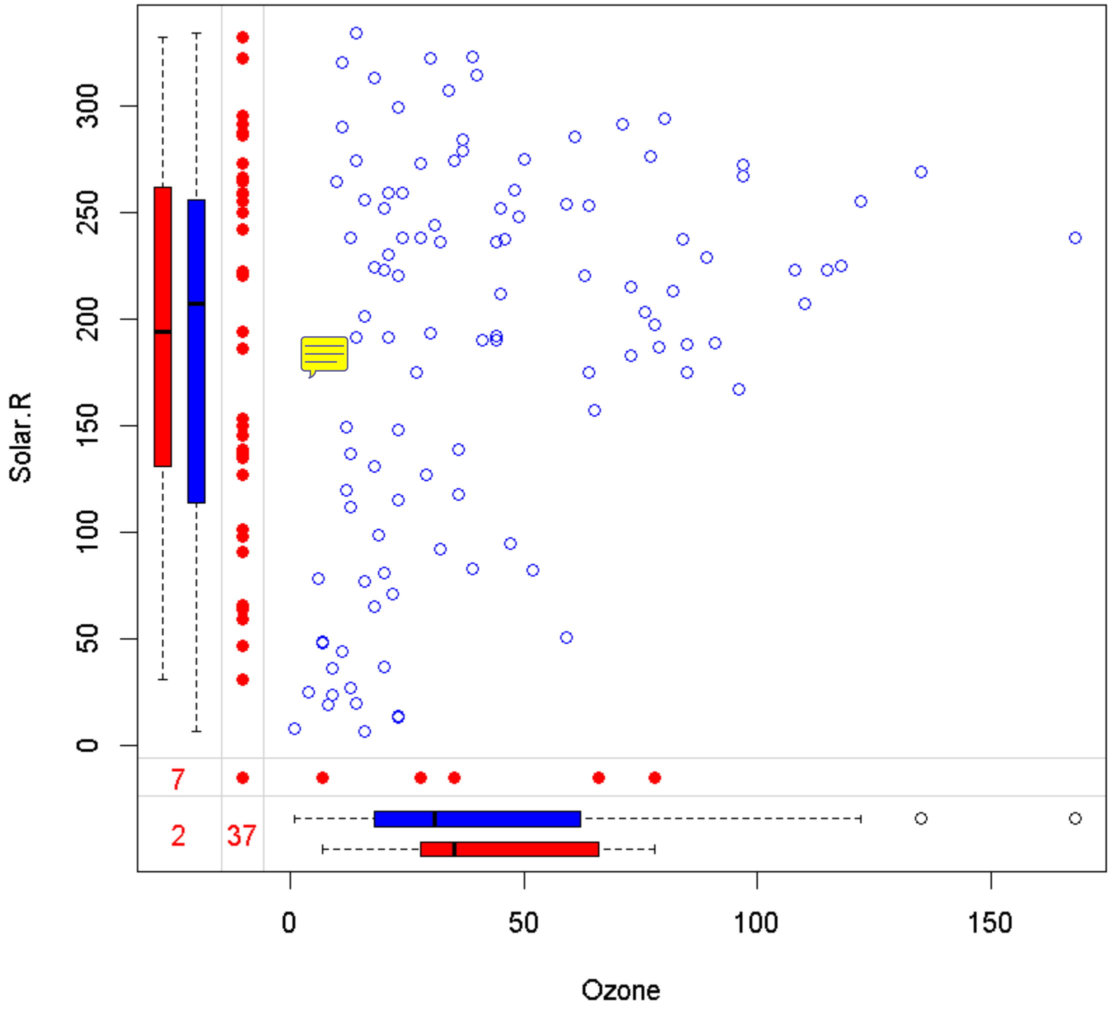
```
$mr
        Ozone Solar.R Wind Temp Month Day
Ozone       0      35   35   36    37  37
Solar.R     5       0    5    6     7   7
Wind        5       5    0    5     7   7
Temp        4       4    3    0     5   5
Month       0       0    0    0     0   0
Day         0       0    0    0     0   0

$mm
        Ozone Solar.R Wind Temp Month Day
Ozone      37       2    2    1     0   0
Solar.R     2       7    2    1     0   0
Wind        2       2    7    2     0   0
Temp        1       1    2    5     0   0
Month       0       0    0    0     0   0
Day         0       0    0    0     0   0
```



V2	v	partial	complete
	x	all missing	partial
		x	v
		V1	

Marginplot

```
> marginplot(mydata[,c("Ozone", "Solar.R")], col = c("blue", "red"))
```



- The blue box plot located on the left and bottom margins shows the distribution of the non-missing datapoints.
- The red box plot on the left shows the distribution of Solar.R with Ozone missing.
- Likewhise for the Ozone box plots at the bottom of the graph.
- If our assumption of MCAR data is correct, then we expect the red and blue box plots to be very similar.

Traditional Approaches to Handling Missing Data

- If missing values do occur by chance among a set of replicates, the observed members of the set can stand in for the missing, albeit with some loss of statistical precision.
- Traditional Approaches to Handling Missing Data

  (T1) List-wise deletion

  (T2) Pairwise deletion

  (T3) Non-response weighting

  (T4) Mean substitution

  (T5) Regression substitution.

  (T6) Last value carried forward.

  (T7) Using information from related observations.

  (T8) Dummy variable adjustment

  (T9) Deterministic imputation.

(T1) List-wise Deletion

- Also called the **complete case analysis**.
- All units with missing data for a variable are removed and the analysis is performed with the remaining units (complete cases).
- This is the default approach in most statistical packages.
- The use of this method is only justified if the missing data generation mechanism is **MCAR**.
- In R, using the function `na.omit()` or extract complete observations using the function `complete.cases()`.

```
> mdata <- matrix(rnorm(15), nrow=5)
> mdata[sample(1:15, 4)] <- NA
> mdata <- as.data.frame(mdata)
> mdata
           V1         V2          V3
1 -0.62222501  1.0807983          NA
2  0.07124865  0.5216675 -0.08334454
3  1.70707399  0.1004917  0.88197789
4          NA -0.6595201 -0.08387860
5          NA  1.6138847          NA
> (x1 <- na.omit(mdata))
          V1        V2          V3
2 0.07124865 0.5216675 -0.08334454
3 1.70707399 0.1004917  0.88197789
> (x2 <- mdata[complete.cases(mdata),])
          V1        V2          V3
2 0.07124865 0.5216675 -0.08334454
3 1.70707399 0.1004917  0.88197789
> mdata[!complete.cases(mdata),]
         V1         V2         V3
1 -0.622225  1.0807983         NA
4        NA -0.6595201 -0.0838786
5        NA  1.6138847         NA
```

快速分析一下，得知資料大概狀況

(T2) Pairwise Deletion

- To compute a covariance matrix, each two cases will be used for which the values of both corresponding variables are available. In R,
  - `use="everything"` (default): use all observations will result in a covariance matrix most likely consisting of NAs.
  - `use="all.obs"`: the presence of missing observations will produce an error.
  - `use="complete.obs"`: missing values are handled by list-wise deletion (and if there are no complete cases, an error appears).
  - `use="pairwise.complete.obs"`: the covariance between each pair of variables is computed using all complete pairs of observations on those variables.
- This can result in covariance or correlation matrices which are not positive semi-definite, as well as NA entries if there are no complete pairs for the given pair of variables.

```
> cov(mdata)
   V1        V2 V3
V1 NA        NA NA
V2 NA 0.7694197 NA
V3 NA        NA NA
> cov(mdata, use = "all.obs")
Error in cov(mdata, use = "all.obs") :
missing observations in cov/cor
> cov(mdata, use = "complete.obs")
           V1          V2         V3
V1  1.3379623 -0.34448500  0.7895494
V2 -0.3444850  0.08869452 -0.2032852
V3  0.7895494 -0.20328521  0.4659237
```

```
> cov(mdata, use = "na.or.complete")
           V1          V2         V3
V1  1.3379623 -0.34448500  0.7895494
V2 -0.3444850  0.08869452 -0.2032852
V3  0.7895494 -0.20328521  0.4659237
> cov(mdata, use = "pairwise")
           V1          V2         V3
V1  1.4304107 -0.56002326 0.78954945
V2 -0.5600233  0.76941970 0.05468712
V3  0.7895494  0.05468712 0.31078774
```

(T4) Mean Substitution

- A very simple but popular approach is to substitute means for the missing values.
- The method preserves sample size and does not reduce the statistical power associated with sample size in comparison with list-wise or pairwise deletion.
- This method produces biased estimates and can severely distort the distribution of the variable in which missing values are substituted.
- This results in underestimates of the standard deviations and distorts relationships between variables (estimates of the correlation are pulled toward zero).

Due to these **distributional problems**, it is often recommended to ignore missing values rather than impute values by mean substitution (Little and Rubin, 1989.)

```
mean.subst <- function(x) {
   x[is.na(x)] <- mean(x, na.rm = TRUE)
   x
}
```

```
> mdata
           V1        V2          V3
1 -0.62222501  1.0807983          NA
2  0.07124865  0.5216675 -0.08334454
3  1.70707399  0.1004917  0.88197789
4          NA -0.6595201 -0.08387860
5          NA  1.6138847          NA
> mdata.mip <- apply(mdata, 2, mean.subst)
> mdata.mip
              V1         V2          V3
[1,] -0.62222501  1.0807983  0.23825158
[2,]  0.07124865  0.5216675 -0.08334454
[3,]  1.70707399  0.1004917  0.88197789
[4,]  0.38536588 -0.6595201 -0.08387860
[5,]  0.38536588  1.6138847  0.23825158
```

Advanced Imputation Methods

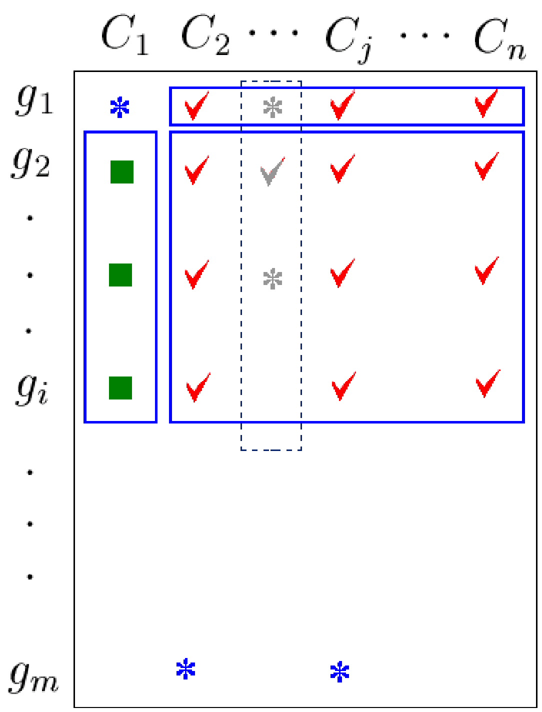
- Univariate methods: column-wise (conditional) mean imputation.

- Multivariate methods: using the linear dependencies between variables.
  - **data-ordering and distance-based** imputation methods such as hot-deck methods and k-nearest neighbour imputation.
  - **covariance-based methods** such as the approaches by Verboven et al (2007) or Serneels and Verdonck (2008), and
  - **model-based methods** approaches such as regression imputation (Raghunathan et al, 2001; Templ et al, 2010) or depth-based imputation (B´ eguin and Hulliger, 2004).

- The assumption of elliptical distributions is necessary for all covariance-based methods, but not for depth-based ones.

(A1) K-Nearest Neighbour Imputation

- KNN imputation searches for the k-nearest observations (respective to the observation which has to be imputed) and replaces the missing value with the mean of the found *k* observations.
- It is recommended to use the (weighted) median instead of the arithmetic mean.
- **KNN** minimize data modeling assumptions and take advantage of the correlation structure of the data.



KNNimpute

Model:

$$\{g_{(k)}, k = 1, 2, \cdots, K\} = \underset{k}{\text{args}} \ \max_{i \in \text{C}} \text{Corr}(g_1, g_i)$$

$$\{g_{(k)}, k = 1, 2, \cdots, K\} = \underset{k}{\text{args}} \ \min_{i \in \text{C}} \text{Dist}(g_1, g_i)$$

C: Observed C_i's without missing values

Imputation:

Average $\widehat{C_1(g_1)} = \frac{1}{K} \sum_{k=1}^{K} C_1(g_k)$

Weighted Average $\widehat{C_1(g_1)} = \frac{\sum_{k=1}^{K} w_k C_1(g_k)}{\sum_{k=1}^{K} w_k}$

$$w_k = \frac{1}{\sum_{j \in \text{C}} [C_j(g_k) - C_1(g_1)]^2}$$

kNN {VIM}:
k-Nearest Neighbour Imputation

Description

k-Nearest Neighbour Imputation based on a variation of the Gower Distance for numerical, categorical, ordered and semi-continous variables.

Usage

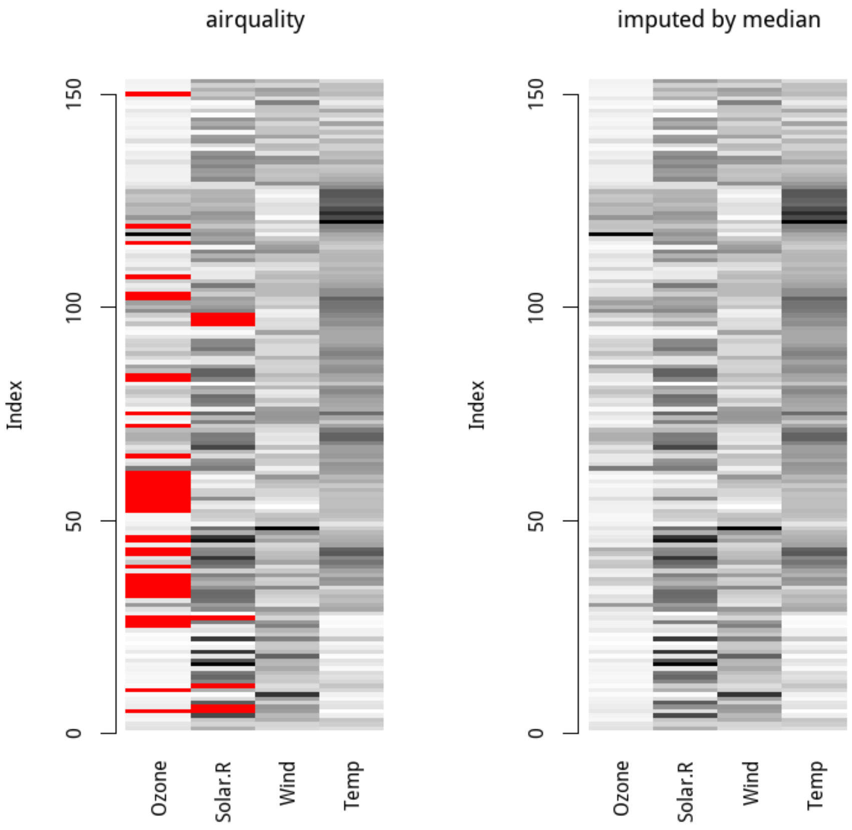
```
kNN(data, variable = colnames(data), metric = NULL, k = 5,
  dist_var = colnames(data), weights = NULL, numFun = median,
  catFun = maxCat, makeNA = NULL, NAcond = NULL, impNA = TRUE,
  donorcond = NULL, mixed = vector(), mixed.constant = NULL,
  trace = FALSE, imp_var = TRUE, imp_suffix = "imp", addRandom = FALSE,
  useImputedDist = TRUE, weightDist = FALSE)
```

```
> names(airquality)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
> airquality.imp.median <- kNN(airquality[1:4], k=5)
> head(airquality.imp.median)
  Ozone Solar.R Wind Temp Ozone_imp Solar.R_imp Wind_imp Temp_imp
1    41     190  7.4   67     FALSE       FALSE    FALSE    FALSE
2    36     118  8.0   72     FALSE       FALSE    FALSE    FALSE
3    12     149 12.6   74     FALSE       FALSE    FALSE    FALSE
4    18     313 11.5   62     FALSE       FALSE    FALSE    FALSE
5    35      92 14.3   56      TRUE        TRUE    FALSE    FALSE
6    28     242 14.9   66     FALSE        TRUE    FALSE    FALSE
```

- Gower JC, 1971, A General Coefficient of Similarity and Some of Its Properties. Biometrics, 857–871.
- Alexander Kowarik and Matthias Templ, 2016, Imputation with the R Package VIM, Journal of Statistical Software, Volume 74, Issue 7.

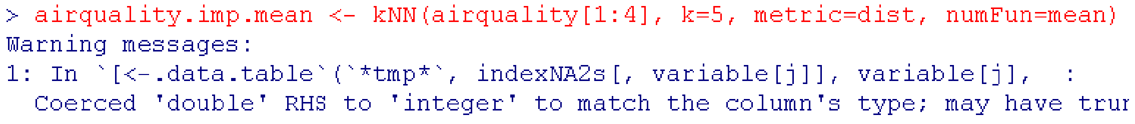
kNN {VIM}:
k-Nearest Neighbour Imputation

```
> matrixplot(airquality[1:4], interactive = F, main="airquality")
> matrixplot(airquality.imp.median[1:4], interactive = F, main="imputed by median")
```



```
trim_mean <- function(x){
   mean(x, trim = 0.1)
}
```

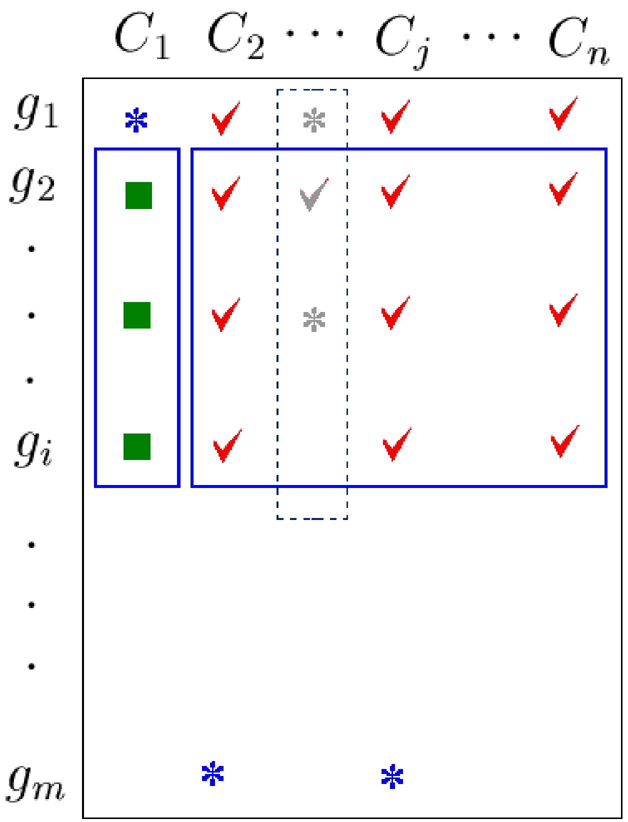
```
> airquality.imp.mean <- kNN(airquality[1:4],
+   k=5, metric=dist, numFun=mean)
> airquality.imp.tmean <- kNN(airquality[1:4],
+   k=5, numFun=trim_mean)
```



```
> airquality.imp.mean <- kNN(airquality[1:4], k=5, metric=dist, numFun=mean)
Warning messages:
1: In `[<-.data.table`(`*tmp*`, indexNA2s[, variable[j]], variable[j],  :
  Coerced 'double' RHS to 'integer' to match the column's type; may have trur
```

(A2) Regression Methods

- Using fitted regression values to replace missing values.
- The model must be chosen so that it does not yields invalid fitted values. e.g., negative values.
- This technique might be more accurate than simply substituting a measure of central tendency, since the imputed value is based on other input variables.



Regression

Model:

$$C_1 = \beta_0 + \sum_{j \in \mathrm{C}} \beta_j C_j$$

C: Observed C_i's without missing values

Imputation:

$$\widehat{C_1(g_1)} = \hat{\beta}_0 + \sum_{j \in \mathrm{C}} \hat{\beta}_j C_j(g_1)$$

regressionImp {VIM}: Regression Imputation

Description

Impute missing values based on a regression model.

Usage

```
regressionImp(formula, data, family = "AUTO", robust = FALSE,
  imp_var = TRUE, imp_suffix = "imp", mod_cat = FALSE)
```

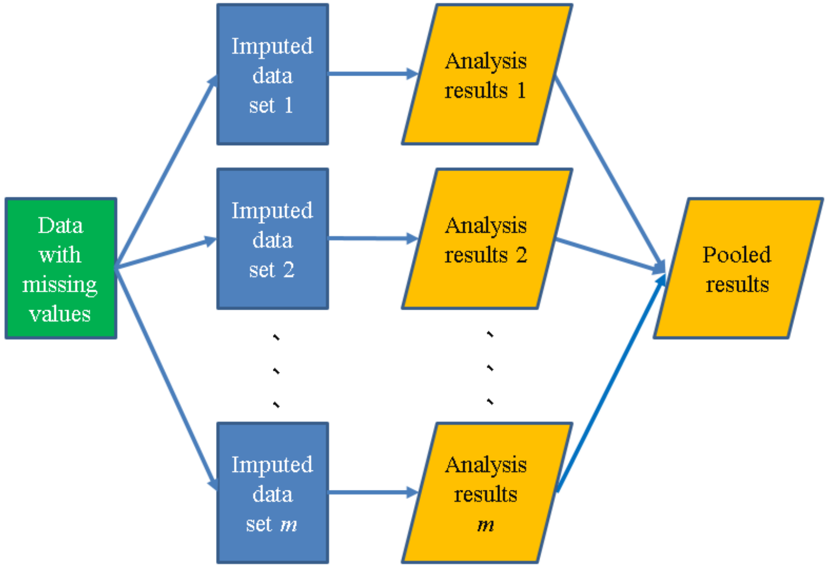
```
> airquality.imp.lm <- regressionImp(Ozone ~ Wind + Temp, data=airquality)
Error in regressionImp_work(formula = formula, data = data, family = family,  : 
  找不到物件 'nLev'
>
> data(sleep)
> summary(sleep)
    BodyWgt            BrainWgt            NonD            Dream           Sleep
 Min.   :   0.005   Min.   :   0.14   Min.   : 2.100   Min.   :0.000   Min.   : 2.60
 1st Qu.:   0.600   1st Qu.:   4.25   1st Qu.: 6.250   1st Qu.:0.900   1st Qu.: 8.05
 Median :   3.342   Median :  17.25   Median : 8.350   Median :1.800   Median :10.45
 Mean   : 198.790   Mean   : 283.13   Mean   : 8.673   Mean   :1.972   Mean   :10.53
 3rd Qu.:  48.203   3rd Qu.: 166.00   3rd Qu.:11.000   3rd Qu.:2.550   3rd Qu.:13.20
 Max.   :6654.000   Max.   :5712.00   Max.   :17.900   Max.   :6.600   Max.   :19.90
                                      NA's   :14       NA's   :12      NA's   :4
      Span              Gest             Pred            Exp            Danger
 Min.   :  2.000   Min.   : 12.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
 1st Qu.:  6.625   1st Qu.: 35.75   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
 Median : 15.100   Median : 79.00   Median :3.000   Median :2.000   Median :2.000
 Mean   : 19.878   Mean   :142.35   Mean   :2.871   Mean   :2.419   Mean   :2.613
 3rd Qu.: 27.750   3rd Qu.:207.50   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :100.000   Max.   :645.00   Max.   :5.000   Max.   :5.000   Max.   :5.000
 NA's   :4         NA's   :4
```

regressionImp {VIM}: Regression Imputation

```
> sleep.imp.lm <- regressionImp(Dream + NonD ~ BodyWgt + BrainWgt, data=sleep)
> summary(sleep.imp.lm)
    BodyWgt            BrainWgt            NonD             Dream             Sleep
 Min.   :   0.005   Min.   :   0.14   Min.   :-11.733   Min.   :-0.6897   Min.   : 2.60
 1st Qu.:   0.600   1st Qu.:   4.25   1st Qu.:  6.525   1st Qu.: 1.0000   1st Qu.: 8.05
 Median :   3.342   Median :  17.25   Median :  8.500   Median : 1.9312   Median :10.45
 Mean   : 198.790   Mean   : 283.13   Mean   :  8.335   Mean   : 1.9326   Mean   :10.53
 3rd Qu.:  48.203   3rd Qu.: 166.00   3rd Qu.: 10.550   3rd Qu.: 2.2750   3rd Qu.:13.20
 Max.   :6654.000   Max.   :5712.00   Max.   : 17.900   Max.   : 6.6000   Max.   :19.90
                                                                          NA's   :4
      Span              Gest             Pred            Exp            Danger
 Min.   :  2.000   Min.   : 12.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
 1st Qu.:  6.625   1st Qu.: 35.75   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
 Median : 15.100   Median : 79.00   Median :3.000   Median :2.000   Median :2.000
 Mean   : 19.878   Mean   :142.35   Mean   :2.871   Mean   :2.419   Mean   :2.613
 3rd Qu.: 27.750   3rd Qu.:207.50   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :100.000   Max.   :645.00   Max.   :5.000   Max.   :5.000   Max.   :5.000
 NA's   :4         NA's   :4
 Dream_imp        NonD_imp
 Mode :logical   Mode :logical
 FALSE:50        FALSE:48
 TRUE :12        TRUE :14
 NA's :0         NA's :0
```

(A7) Multiple Imputation

- Multiple imputation requires three steps
  - Imputation: impute the missing entries of the incomplete data sets *m* times. Imputed values are drawn for a distribution (that can be different for each missing entry). This step results is *m* complete data sets.
  - Analysis: Analyze each of the *m* completed data sets. This step results in *m* analyses.
  - Pooling: Integrate the *m* analysis results into a final result.
- Rubin (1987) has shown that if the method to create imputations is inferences will be statistically valid.



Multiple Imputation Online: www.multiple-imputation.com

Rubin, D.B. (1987), Multiple Imputation for Nonresponse in Surveys, New York: John Wiley & Sons, Inc.
Little, R.J.A. and Rubin, D.B. (1987), Statistical Analysis with Missing Data, New York: John Wiley & Sons, Inc.

Comparison

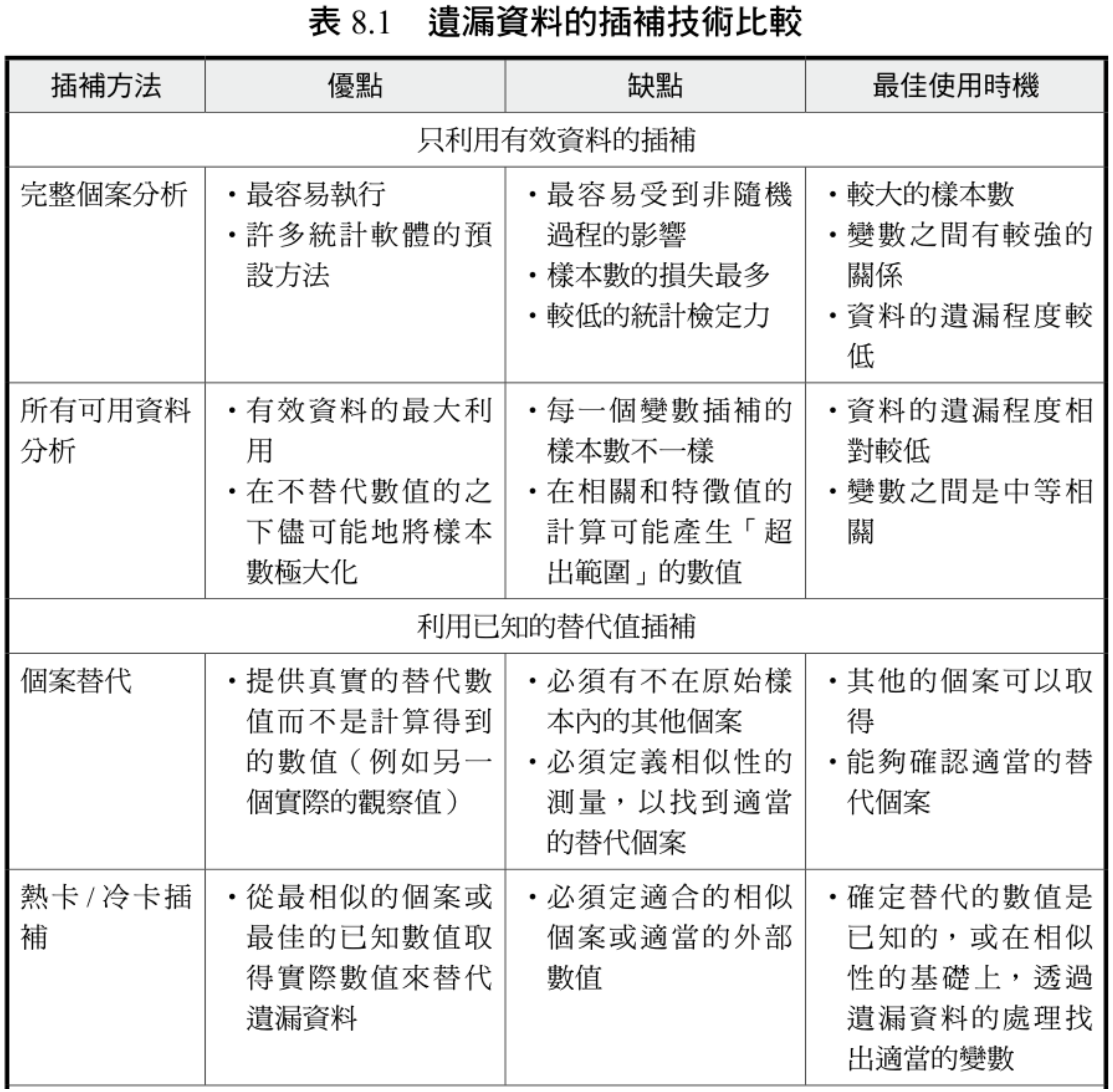


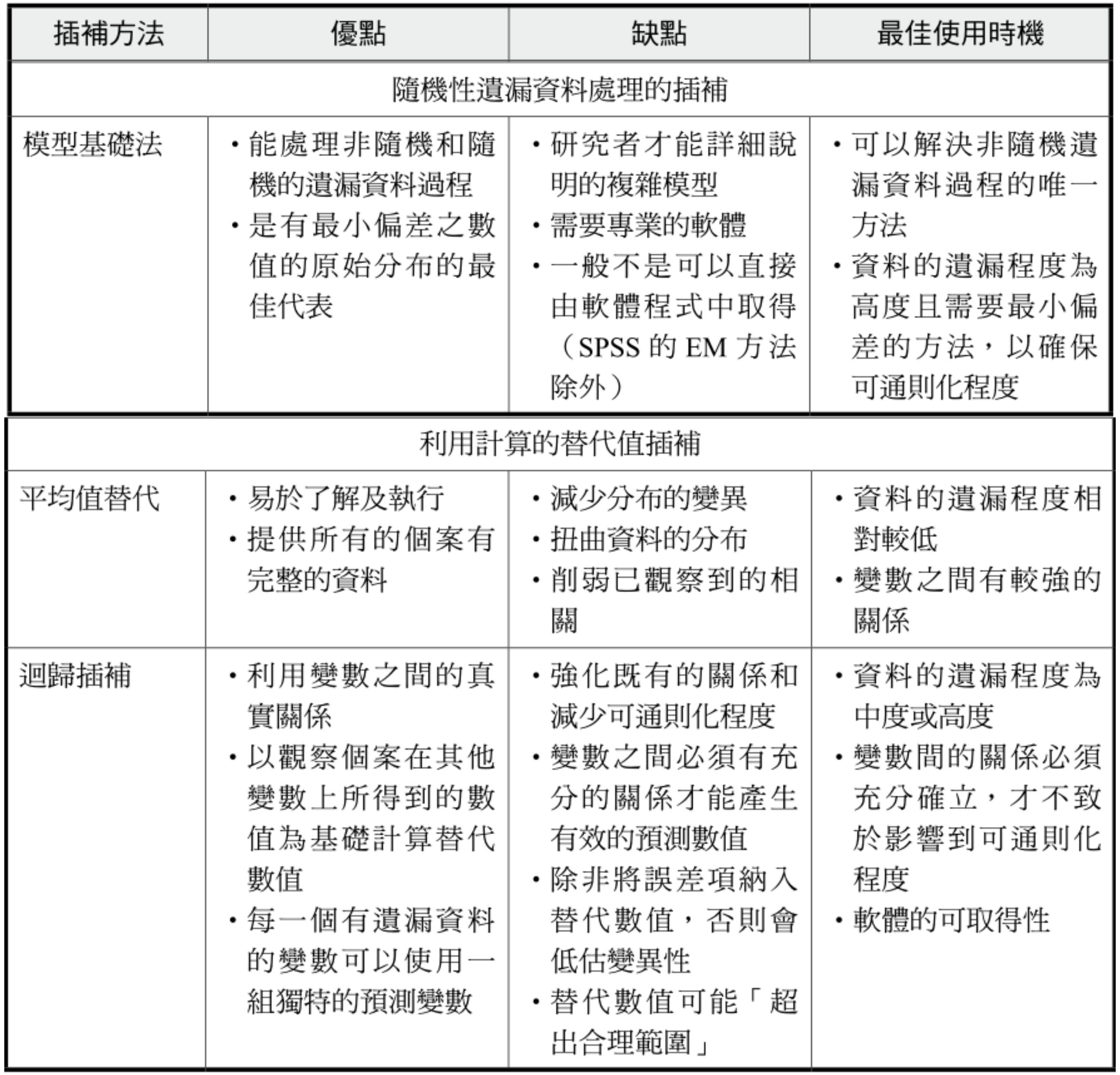
表 8.1　遺漏資料的插補技術比較

插補方法	優點	缺點	最佳使用時機
只利用有效資料的插補			
完整個案分析	• 最容易執行 • 許多統計軟體的預設方法	• 最容易受到非隨機過程的影響 • 樣本數的損失最多 • 較低的統計檢定力	• 較大的樣本數 • 變數之間有較強的關係 • 資料的遺漏程度較低
所有可用資料分析	• 有效資料的最大利用 • 在不替代數值的之下儘可能地將樣本數極大化	• 每一個變數插補的樣本數不一樣 • 在相關和特徵值的計算可能產生「超出範圍」的數值	• 資料的遺漏程度相對較低 • 變數之間是中等相關
利用已知的替代值插補			
個案替代	• 提供真實的替代數值而不是計算得到的數值（例如另一個實際的觀察值）	• 必須有不在原始樣本內的其他個案 • 必須定義相似性的測量，以找到適當的替代個案	• 其他的個案可以取得 • 能夠確認適當的替代個案
熱卡 / 冷卡插補	• 從最相似的個案或最佳的已知數值取得實際數值來替代遺漏資料	• 必須定適合的相似個案或適當的外部數值	• 確定替代的數值是已知的，或在相似性的基礎上，透過遺漏資料的處理找出適當的變數

劉正山, 莊文忠, 2012, 項目無反應資料的多重插補分析, 第八章, 臺灣選舉與民主化調查(TEDS)方法論之回顧與前瞻 (黃紀 主編。) pp. 276-305.

Comparison

MLE, EM



插補方法	優點	缺點	最佳使用時機
隨機性遺漏資料處理的插補			
模型基礎法	・能處理非隨機和隨機的遺漏資料過程 ・是有最小偏差之數值的原始分布的最佳代表	・研究者才能詳細說明的複雜模型 ・需要專業的軟體 ・一般不是可以直接由軟體程式中取得（SPSS 的 EM 方法除外）	・可以解決非隨機遺漏資料過程的唯一方法 ・資料的遺漏程度為高度且需要最小偏差的方法，以確保可通則化程度
利用計算的替代值插補			
平均值替代	・易於了解及執行 ・提供所有的個案有完整的資料	・減少分布的變異 ・扭曲資料的分布 ・削弱已觀察到的相關	・資料的遺漏程度相對較低 ・變數之間有較強的關係
迴歸插補	・利用變數之間的真實關係 ・以觀察個案在其他變數上所得到的數值為基礎計算替代數值 ・每一個有遺漏資料的變數可以使用一組獨特的預測變數	・強化既有的關係和減少可通則化程度 ・變數之間必須有充分的關係才能產生有效的預測數值 ・除非將誤差項納入替代數值，否則會低估變異性 ・替代數值可能「超出合理範圍」	・資料的遺漏程度為中度或高度 ・變數間的關係必須充分確立，才不致於影響到可通則化程度 ・軟體的可取得性

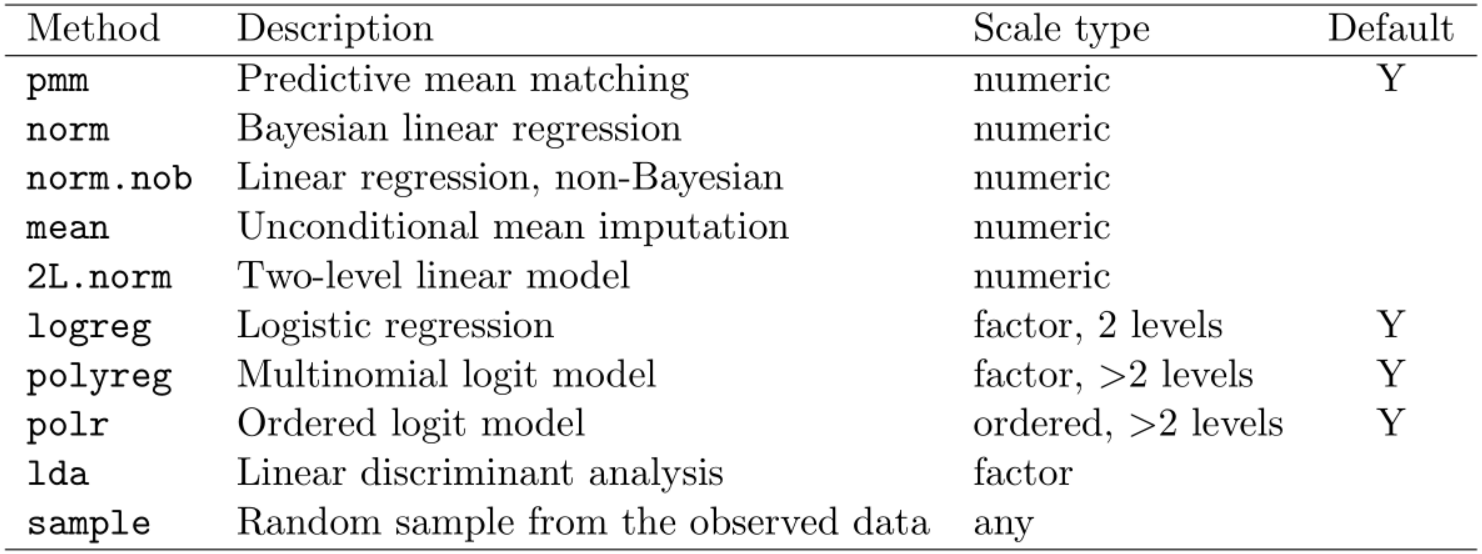
資料來源：Hairs et al. (2010, 55)

劉正山, 莊文忠, 2012, 項目無反應資料的多重插補分析, 第八章, 臺灣選舉與民主化調查(TEDS)方法論之回顧與前瞻 (黃紀 主編。) pp. 276-305.

Generates Multivariate Imputations by Chained Equations (MICE)

```
mice(data, m = 5, method = vector("character", length = ncol(data)),
  predictorMatrix = (1 - diag(1, ncol(data))),
  visitSequence = (1:ncol(data))[apply(is.na(data), 2, any)],
  form = vector("character", length = ncol(data)),
  post = vector("character", length = ncol(data)), defaultMethod = c("pmm",
  "logreg", "polyreg", "polr"), maxit = 5, diagnostics = TRUE,
  printFlag = TRUE, seed = NA, imputationMethod = NULL,
  defaultImputationMethod = NULL, data.init = NULL, ...)
```

```
> methods(mice)
 [1] mice.impute.2l.norm      mice.impute.2l.pan       mice.impute.2lonly.mean
 [4] mice.impute.2lonly.norm  mice.impute.2lonly.pmm   mice.impute.cart
 [7] mice.impute.fastpmm      mice.impute.lda          mice.impute.logreg
[10] mice.imput
[13] mice.imput
[16] mice.imput
[19] mice.imput
[22] mice.imput
[25] mice.theme
see '?methods' 
> ? mice
```



Method	Description	Scale type	Default
pmm	Predictive mean matching	numeric	Y
norm	Bayesian linear regression	numeric	
norm.nob	Linear regression, non-Bayesian	numeric	
mean	Unconditional mean imputation	numeric	
2L.norm	Two-level linear model	numeric	
logreg	Logistic regression	factor, 2 levels	Y
polyreg	Multinomial logit model	factor, >2 levels	Y
polr	Ordered logit model	ordered, >2 levels	Y
lda	Linear discriminant analysis	factor	
sample	Random sample from the observed data	any	

Impute Missing Values

```
> mydata.ip <- mice(mydata, m=5, maxit=50, meth='pmm', seed=500)

 iter imp variable
  1   1  Ozone  Solar.R  Wind  Temp
  1   2  Ozone  Solar.R  Wind  Temp
...
  50   4  Ozone  Solar.R  Wind  Temp
  50   5  Ozone  Solar.R  Wind  Temp
> summary(mydata.ip)
Multiply imputed data set
Call:
mice(data = mydata, m = 5, method = "pmm", maxit = 50, seed = 500)
Number of multiple imputations:  5
Missing cells per column:
  Ozone Solar.R    Wind    Temp   Month     Day
     37       7       7       5       0       0
Imputation methods:
  Ozone Solar.R    Wind    Temp   Month     Day
  "pmm"   "pmm"   "pmm"   "pmm"   "pmm"   "pmm"
VisitSequence:
  Ozone Solar.R    Wind    Temp
      1       2       3       4
PredictorMatrix:
        Ozone Solar.R Wind Temp Month Day
Ozone       0       1    1    1     1   1
Solar.R     1       0    1    1     1   1
Wind        1       1    0    1     1   1
Temp        1       1    1    0     1   1
Month       0       0    0    0     0   0
Day         0       0    0    0     0   0
Random generator seed value:  500
```

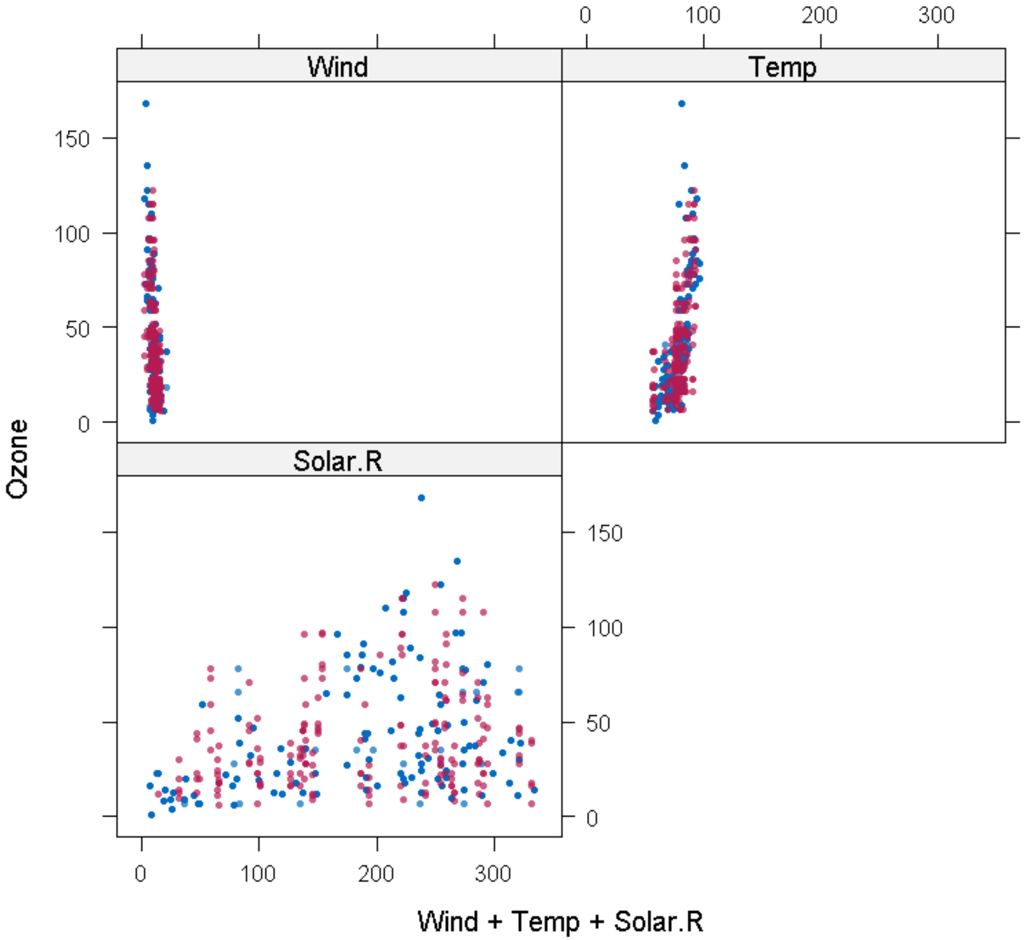
```
> mydata.ip$imp$Ozone
>  1   2   3   4   5
5     59  85  20 108  18
10    11   7  27  14  21
...
150    9  34  27  12  22
```

The output shows the imputed data for each observation (first column left) within each imputed dataset (first row at the top).

```
> # get back the first completed dataset out of 5
> mydata.completed <- complete(mydata.ip, 1)
```

Compare the Distributions of Original and Imputed data

```
> library(lattice)
> xyplot(mydata.ip, Ozone ~ Wind + Temp + Solar.R, pch=16, cex=0.5)
```

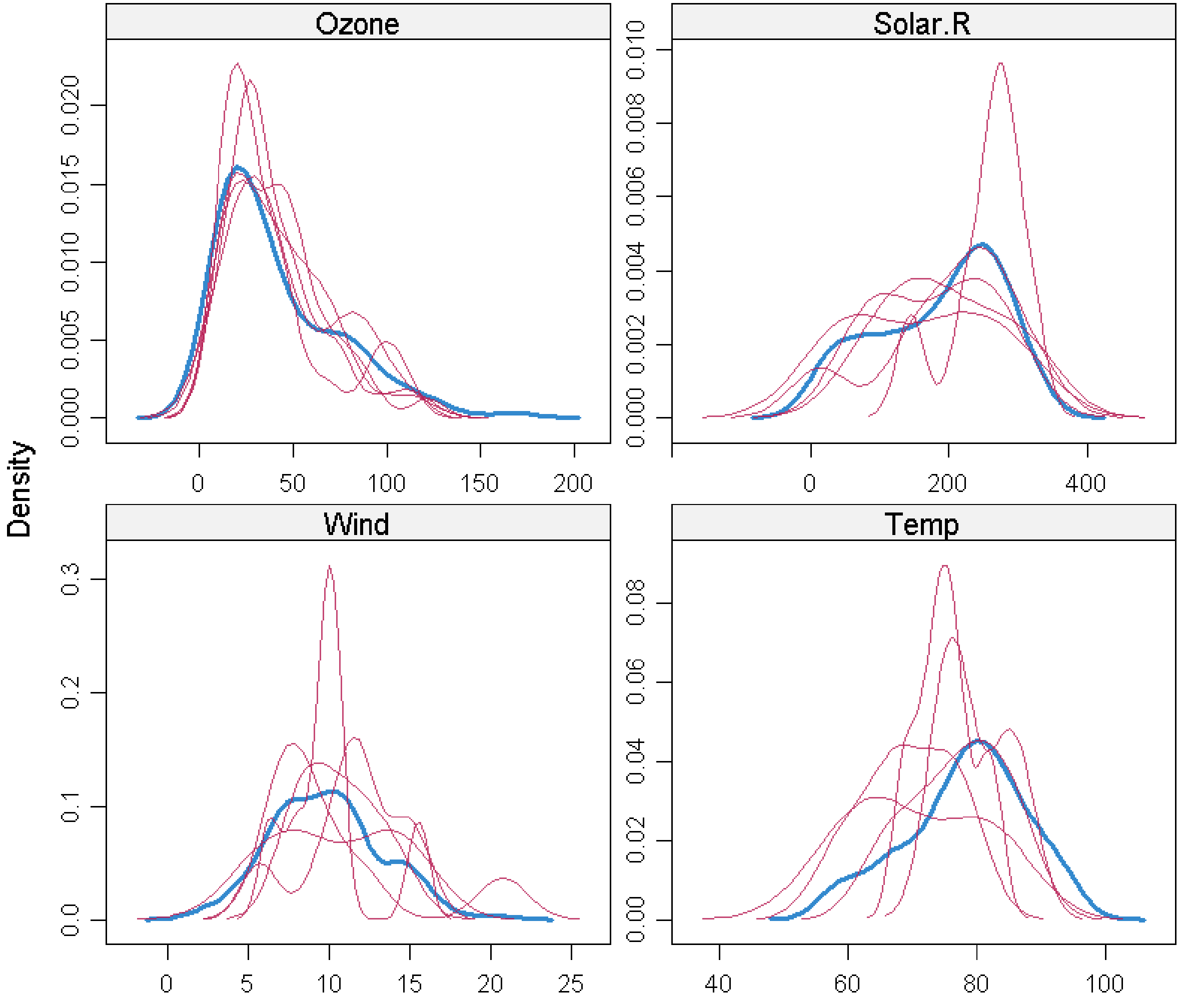


- Check if the shape of the imputed points (magenta) matches the shape of the observed (blue) ones (observed).

- The matching shape means the imputed values are indeed "plausible values".

Density Plot

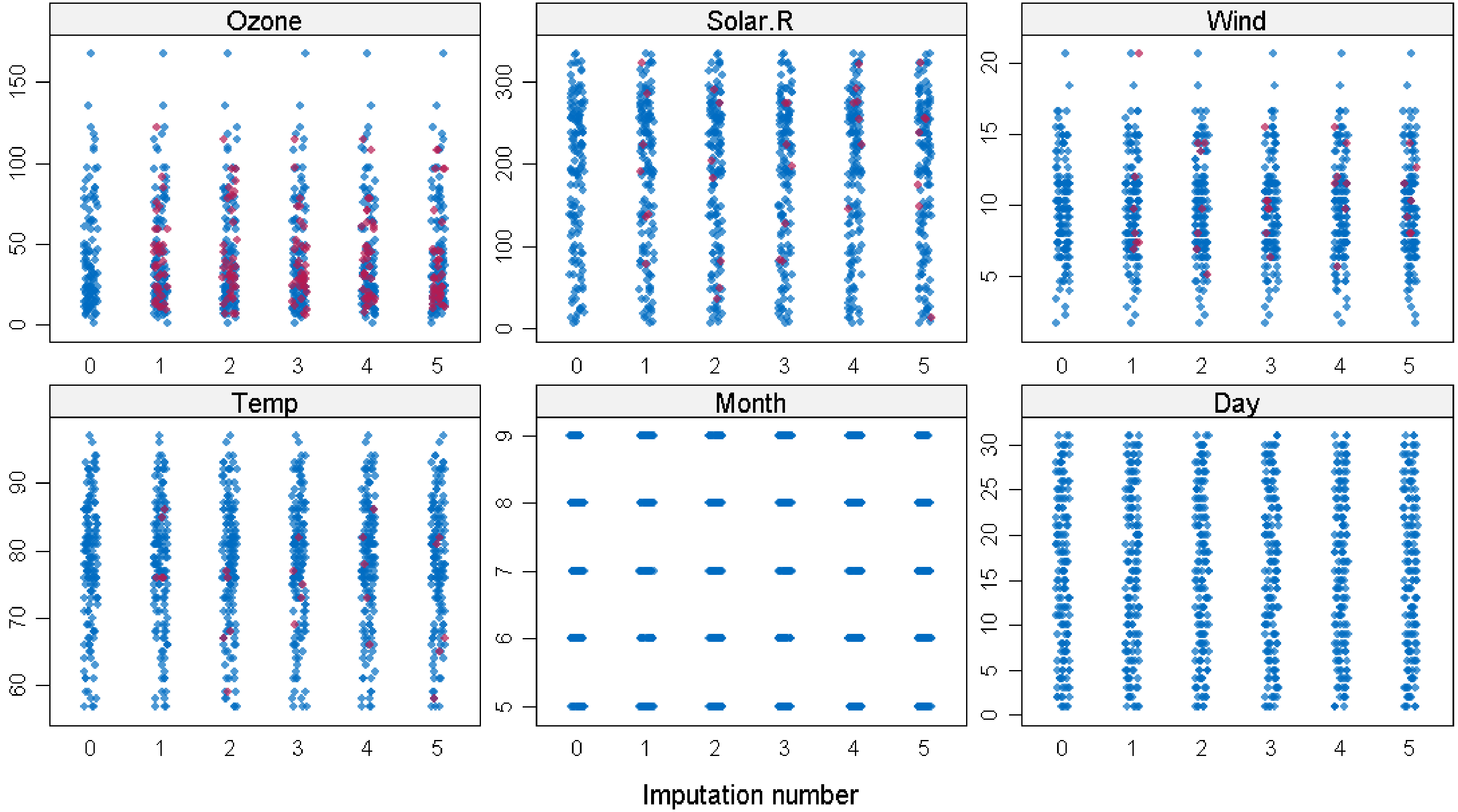
```
> densityplot(mydata.ip)
```



The density of the imputed data for each imputed dataset is showed in magenta while the density of the observed data is showed in blue. Under MCAR, we expect the distributions to be similar.

Shows the Distributions of the Variables as Individual Points

```
> stripplot(mydata.ip, pch = 16, cex = 0.6)
```



Pooling

- Next step: fit a linear model to the data.
- **mice** fit a model to each of the imputed dataset and then pool the results together.

```
> # linear regression for each imputed data set - 5 regression are run
> modelFit1 <- with(mydata.ip, lm(Temp~ Ozone + Solar.R+Wind))
> # pool coefficients and standard errors across all 5 regression models
> summary(pool(modelFit1))
                    est          se          t       df     Pr(>|t|)        lo 95
(Intercept) 71.11418579 2.840129171 25.0390674 85.04465 0.000000e+00 65.467290906
Ozone        0.17412083 0.025108183  6.9348239 72.90551 1.383136e-09  0.124079199
Solar.R      0.01004273 0.007163085  1.4020115 87.03503 1.644683e-01 -0.004194599
Wind        -0.21504110 0.222484210 -0.9665454 61.98616 3.375274e-01 -0.659782671
                  hi 95 nmis       fmi    lambda
(Intercept) 76.76108067   NA 0.1459648 0.1261138
Ozone        0.22416246   37 0.1734348 0.1510666
Solar.R      0.02428005    7 0.1418215 0.1223252
Wind         0.22970047    7 0.2026905 0.1773735
```

To reduce the effect of the random seed initialization, we can impute a higher number of dataset, by changing the default **m=5** parameter in the **mice()** function.

```
mydata.ip2 <- mice(mydata, m=50, seed=245435)
modelFit2 <- with(mydata.ip2,lm(Temp ~ Ozone + Solar.R + Wind))
summary(pool(modelFit2))
```

Quick Tutorial on MICE Package

```
> # Generate 10% missing values at Random
> iris.mis <- prodNA(iris, noNA = 0.1)  # library(missForest)
> # Check missing values introduced in the data
> summary(iris.mis)
> iris.mis <- subset(iris.mis, select = -c(Species))
> summary(iris.mis)
>
> #  A tabular form of missing value present in each variable
> library(mice)
> md.pattern(iris.mis)
> # Visualization
> library(VIM)
> mice_plot <- aggr(iris.mis, col=c('navyblue','yellow'), numbers=TRUE, sortVars=TRUE,
                    labels=names(iris.mis), cex.axis=.7,
                    gap=3, ylab=c("Missing data","Pattern"))

> #  Imputation
> imputed.Data <- mice(iris.mis, m=5, maxit = 50, method = 'pmm', seed = 500)
> summary(imputed.Data)
> # Check imputed values
> imputed.Data$imp$Sepal.Width
> # Get complete data (2nd out of 5)
> completeData <- complete(imputed.Data,2)
> # Build predictive model
> fit <- with(data = imputed.Data, exp = lm(Sepal.Width ~ Sepal.Length + Petal.Width))
> # Combine results of all 5 models
> combine <- pool(fit)
> summary(combine)
```

Source: http://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/

Which Imputation Method?

- KNN is the most widely-used.
- Characteristics of data that may affect choice of imputation method:
  - dimensionality
  - percentage of values missing
  - experimental design (time series, case/control, etc.)
  - patterns of correlation in data
- Suggestion!!
  - add (**same percentage**) artificial missing values to your (**complete cases**) data set
  - impute them with various methods
  - see which is best (since you know the real value)