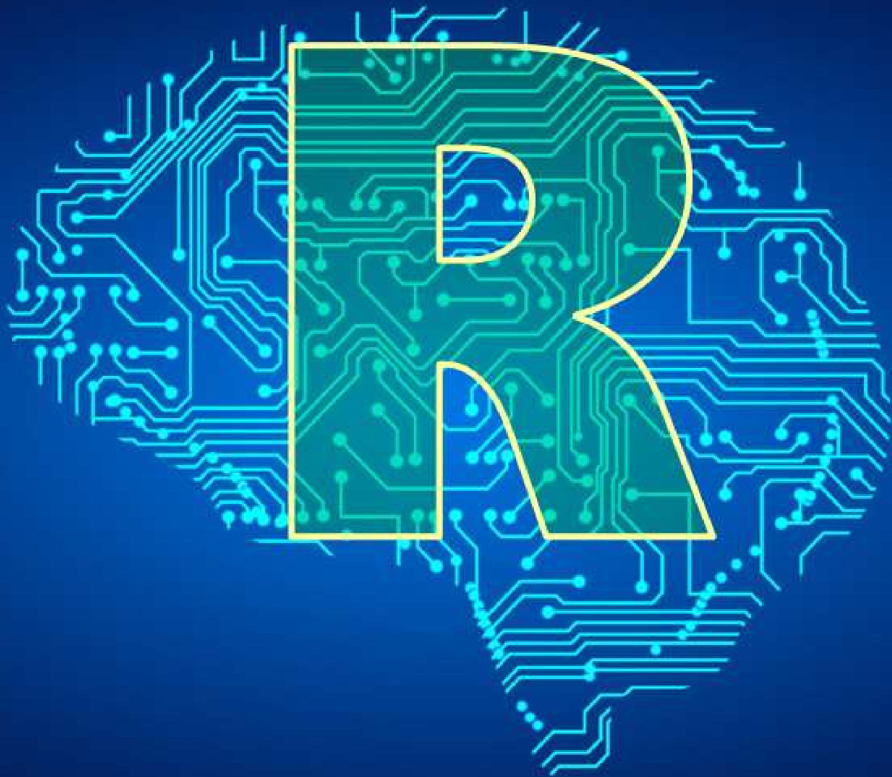
StatR-07

台灣人工智慧學校



缺失值處理

吳漢銘
國立臺北大學 統計學系

缺失值處理 - 大綱

- 主題1
  - 具缺失值資料 (Missing Data)
  - 缺失機制 (Missingness Mechanism)
    - Missing by Design
    - Missing Completely at Random (MCAR)
    - Missing at Random (MAR)
    - Missing Not at Random (MNAR)
- 主題2
  - R Packages for Dealing With Missing Values: VIM, MICE
  - Visualizing the Pattern of Missing Data
  - Traditional Approaches to Handling Missing Data
  - Imputation Methods: KNN
  - Which Imputation Method?

具缺失值資料 (Missing Data)

Missing data (missing values for certain variables for certain cases): **item non-response**.

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**.

	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

缺失值的處理

- The missing values may give clues to systematic aspects of the problem.

- **How to deal with missing values:**
  - Use a global constant to fill the value will misguide the mining process.
    (例如: 缺考給0分; 影像訊號=前景-背景)
  - 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)

缺失機制
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.

缺失機制
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:

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

Missing by Design
Missing Completely at Random

- Missing by Design
  - **Excluded** some participants from the analysis because they are not part of the population under investigation.
  - missingness codes: (i) refused to answer; (ii) answered don't know; (iii) had a valid skip or (iv) was skipped by an enumerator error.

- Missing Completely at Random (MCAR)
  - missingness is independent of their own unobserved values and the observed data.

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

  - *Example*: Miscoding or forgetting to log in answer.
  - **Imputation methods** rely on the missingness being of the MCAR type.

Missing at Random (MAR)
Missing Not at Random (MNAR)

- Missing at Random (MAR) $Pr(R|X) = Pr(R|X_o)$
  - missingness does not depend on their unobserved value but does dependent on the observed data.
  - *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.
  - We can ignore missing data (= omit missing observations) if we have MAR or MCAR.

- Missing Not at Random (MNAR)
  - Missingness that depends on the missing value itself.
  - *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).
  - MNAR data is a more serious issue. (not ignorable)

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 data points.
  - 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).

缺失值處理 - 大綱

- 主題1
  - 具缺失值資料 (Missing Data)
  - 缺失機制 (Missingness Mechanism)
    - Missing by Design
    - Missing Completely at Random (MCAR)
    - Missing at Random (MAR)
    - Missing Not at Random (MNAR)

- 主題2
  - R Packages for Dealing With Missing Values: VIM, MICE
  - Visualizing the Pattern of Missing Data
  - Traditional Approaches to Handling Missing Data
  - Imputation Methods: KNN
  - Which Imputation Method?

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 to
> x
[1] NA  4  7 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`**)

```
> 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
```

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.

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

- **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.

```
> 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

- `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:
  - 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

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.impute
[13] mice.impute
[16] mice.impute
[19] mice.impute
[22] mice.impute
[25] mice.theme
see '?methods' f
> ? 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	

Exploring Missing Data

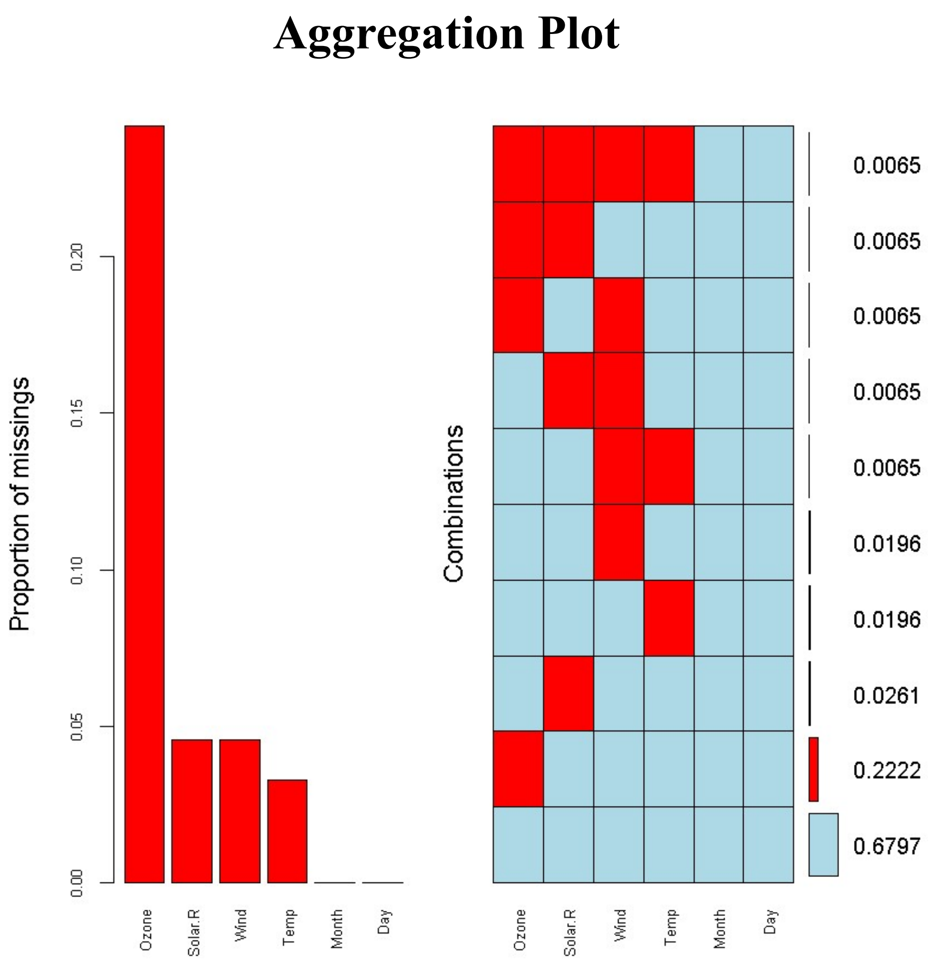
```
> head(airquality)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
> dim(airquality)
[1] 153   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

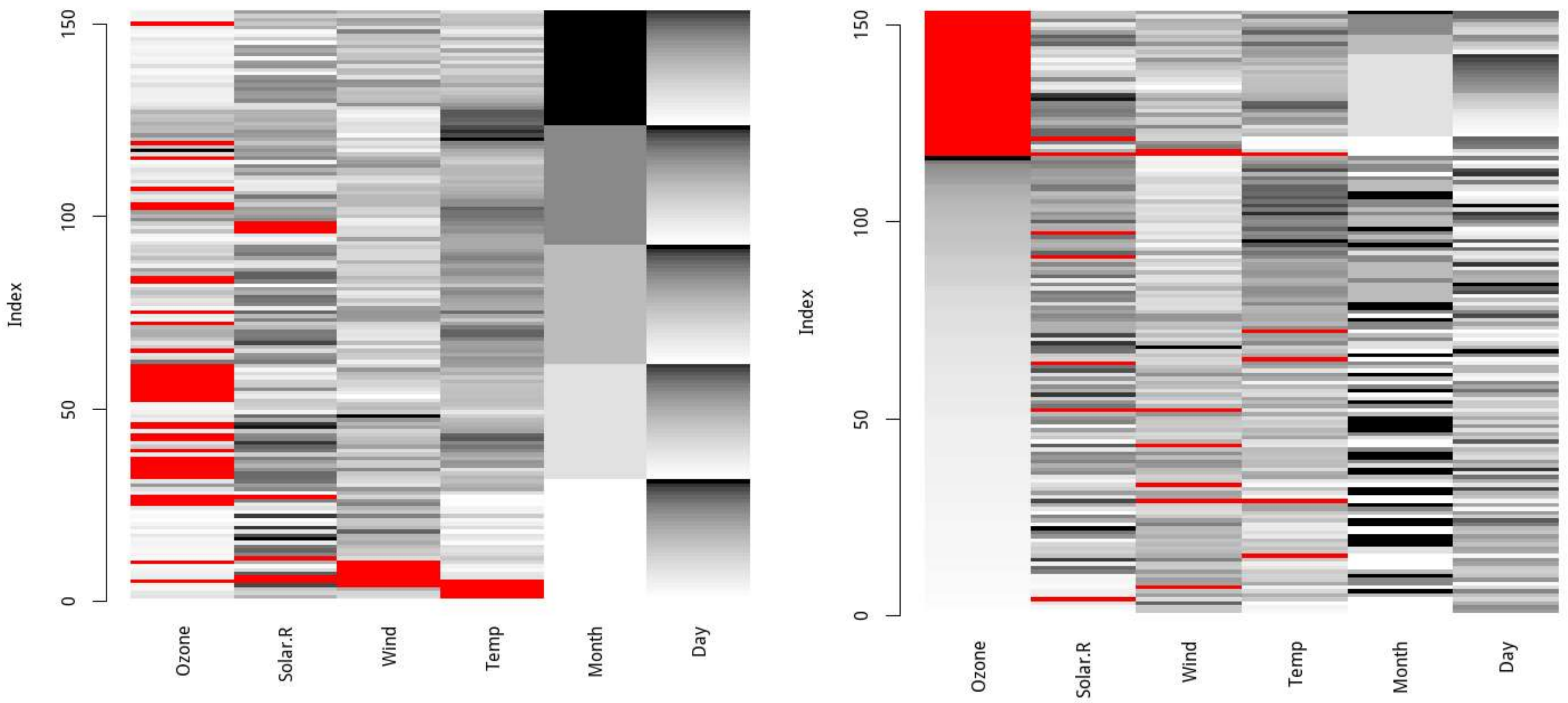
```
> library(mice)
> md.pattern(mydata)
    Month Day Temp Solar.R Wind Ozone
104     1   1    1       1    1     1  0
 34     1   1    1       1    1     0  1
  4     1   1    1       0    1     1  1
  3     1   1    1       1    0     1  1
  3     1   1    0       1    1     1  1
  1     1   1   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



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

<table>
<tr><td rowspan="2">V2</td><td>v</td><td>partial</td><td>complete</td></tr>
<tr><td>x</td><td>all missing</td><td>partial</td></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td>x</td><td>v</td></tr>
<tr><td colspan="2">V1</td></tr>
</table>

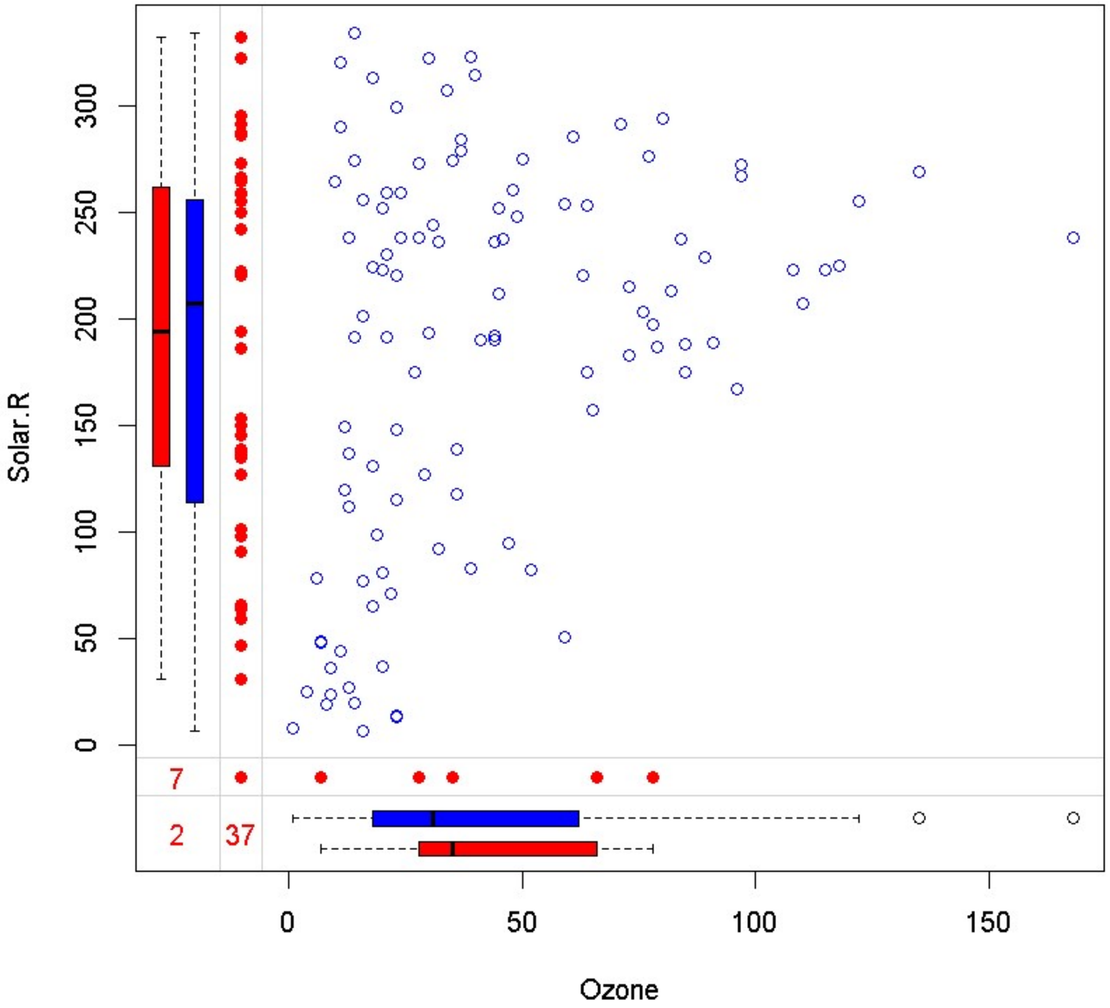
- **rr**: response-response, both variables are observed
- **rm**: response-missing, row observed, column missing
- **mr**: missing-response, row missing, column observed
- **mm**: missing-missing, both variables are missing

```
> 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
```

```
$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
```

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.
- If our assumption of MCAR data is correct, then we expect the red and blue box plots to be very similar.

List-wise Deletion

- Also called the **complete case analysis**.
- The use of this method is only justified if the missing data generation mechanism is **MCAR**.

```
> 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

Pairwise Deletion

- To compute a covariance matrix, each two cases will be used for which the values of both corresponding variables are available.
- 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.

```
> 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
> 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
```

Mean Substitution

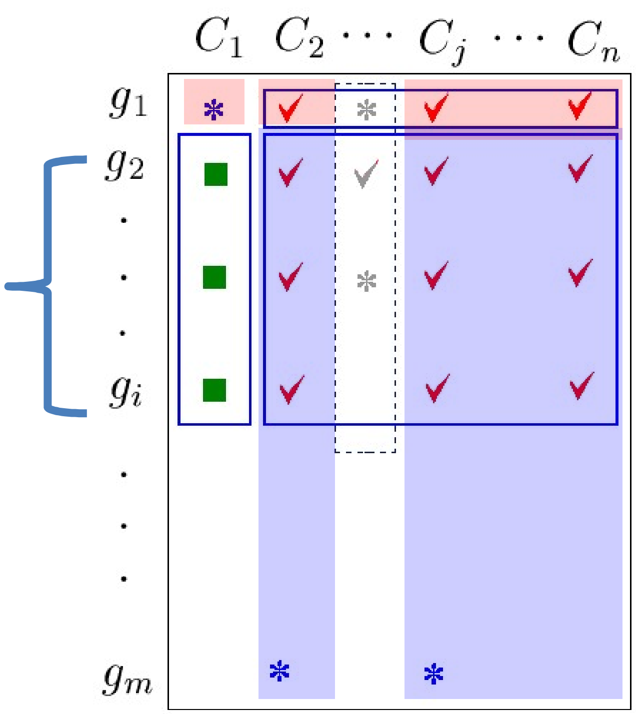
- A very simple but popular approach is to substitute means for the missing values.
- This method produces biased estimates and can severely distort the distribution of the variable in which missing values are substituted.
- Due to these **distributional problems**, it is often recommended to ignore 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
```

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:

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

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

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

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

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)
```

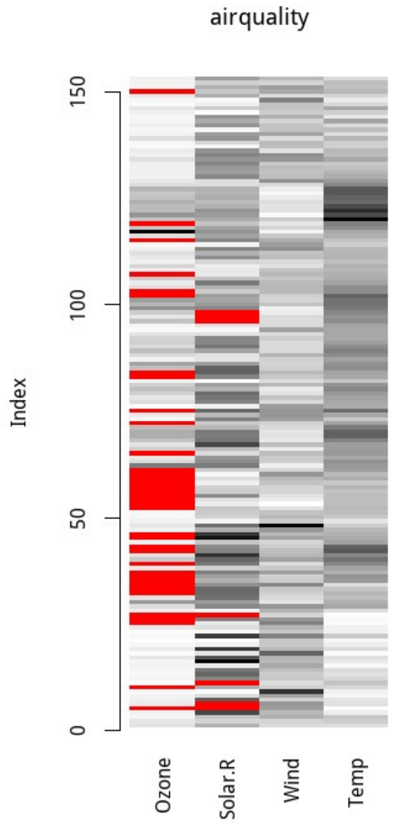
mean
weightedMean

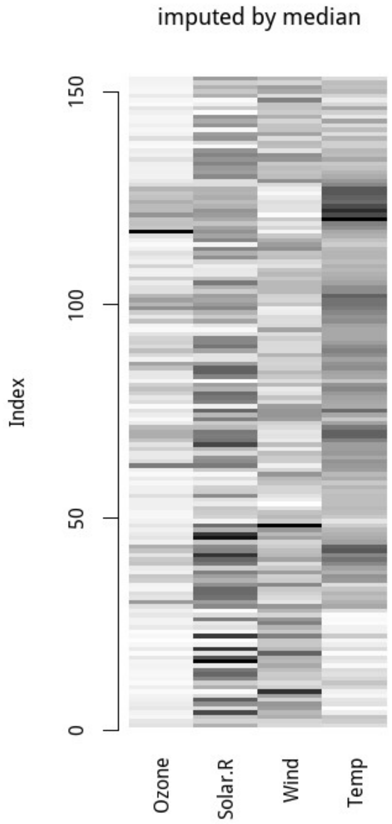
```
> names(airquality)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
> airquality.imp.median <- kNN(airquality[1:4], k=5)
> head(airquality.imp.median)
  Ozone Solar.R Wind Temp Ozone_imp Solar.R_imp Wind_imp Temp_imp
1    41     190  7.4   67     FALSE       FALSE    FALSE    FALSE
2    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.

matrixplot、自定平均函數

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





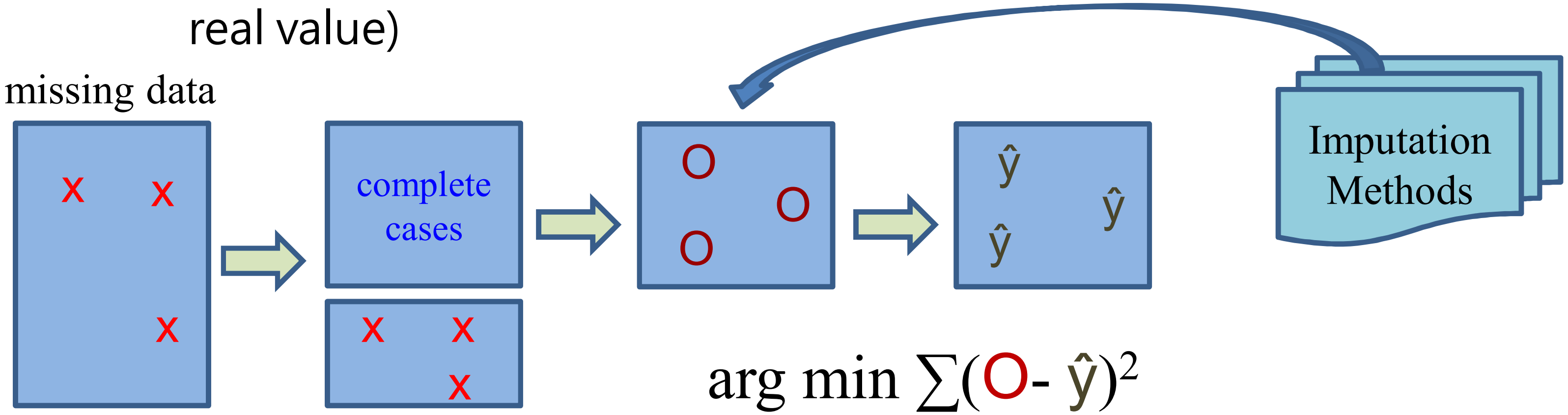
自定平均函數

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

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

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)



$$\arg\min \sum (O - \hat{y})^2$$