第 14 章: 因子物件與類別變數

14: Factor Object

因子物件 或 因數物件 (factor) 爲處理 類別資料 (categorical data), 提供的一種有效的方法. 在科學測量上, 類別尺度 (categorical scale) 或 類別變數 (categorical variable) 將變數值分成互斥的類別水準, 在類別變數定義中的特定幾種類別水準是否有大小差距, 又可分類成 名目變數 (nominal variable) 與 有序變數 (ordinal variable), 或稱 名目尺度 與 順序尺度. 其中 名目尺度 是最簡單的測量, 名目尺度將變數值分成互斥的類別水準, 同一變數內的類別水準並無量化大小的差別. 名目尺度用文字或數字表示或標記, 這些數字本身並無任何意義, 例如 "左側" 爲 1, "右側" 爲 0; 1 = "拇指", 2 = "食指", 3 = "中指", 4 = "無名指", 5 = "小指". 在名目尺度中, 特別的情形是變數內只有 2 個類別水準, 如存活或死亡, 統計習慣上會分別標記爲 0 或 1, 感染或無感染, 這類變數常稱做 二元變數 (dichotomous variable, binary variable), 二元變數通常有特殊的統計分析方法. 有些時候, 對二元變數會先計算變數內每一水準的數目. 形成分類表格.

順序尺度 (ordinal scale) 是指同一變數內的類別水準有輕重, 大小, 強弱, 好壞等級順序之資料. 例如, 疼痛情形有 4 種情境: "無", "輕度", "中度", "重度", 癌症分期爲 I, II, III, IV 等 4 期. 雖然順序尺度用數字表示或標記, 但是數字本身通常不能用來做運算, 只能比較相對大小或高低次序, 順序之間的實際差異並無法從標記的數

字差異得知.

統計中的 類別資料 (categorical data), 在數理統計的討論爲 離散變數 (discrete variable), 包含 名義變數 (nominal variable) 與 有序變數 (ordinal variable), 有各種不同表示方法, 在資料儲存時常常是以文字爲變數值, 但使用上文字較占空間且無法精準傳達類別變數的概念. 因此在 R 中特別使用 因子 (factor) 物件來表示 類別變數 (categorical variable). 因素 或 因子 是一種特殊的文字向量, 文字向量中的每一個元素, 取一個離散值, 因子物件 有一個特殊屬性, 稱爲 層次, 水平, 水準 或 類別水準 (levels), 表示這組所有可能的離散值. 因素 或 因子常常是用 文字或字串輸入, 有時會使用數值或整數代表, 一但變數設定爲 因素 或 因子 向量, R 在列印或輸出時, 並不會加上雙引號 ", 且數值或有大小順序的文字, R 在統計分析上都必須特別處理.

14.1 因子物件函式 factor()

在 R 中若要設定因子, 可以簡單地用函式 `factor()` 產生無序因子物件.

```
1 > factor(x = character(), levels, labels = levels,
2          exclude = NA, ordered = is.ordered(x), nmax = NA)
```

其中引數爲

- `x` 爲原始向量, 通常爲文字向量, 若是數值向量, R 先轉換成文字向量.
- `levels` 設定類別水準.
- `labels` 類別水準的標記文字.
- `exclude = NA` 排除缺失值或某一特定值爲一類別水準.
- `ordered = is.ordered(x)` 設定因子物件類別水準的順序, 仍是無序因子物件.
- `nmax = NA` 是否類別水準的最大數目.

```
> ## factor()
> sex <- c("Male", "Female", "Male", "Male", "Female")
> sex
[1] "Male"   "Female" "Male"   "Male"   "Female"
> class(sex)
[1] "character"
> sex <- factor(sex)
> sex
[1] Male   Female Male   Male   Female
Levels: Female Male
> class(sex)
[1] "factor"
> #
> ## factor() + levels + labels
> x.chr = c("male", "male", "female", "female")
> factor(x.chr, levels = c("male", "female", "bisex"))
[1] male   male   female female
Levels: male female bisex
> factor(x.chr, levels = c("male", "female", "bisex"),
+               labels = c("m", "f", "b"))
[1] m m f f
Levels: m f b
> #
> pain <- c("none", "mild", "moderate", "severe", NA)
> factor(pain) # NA is NOT a level.
[1] none     mild     moderate severe   <NA>
Levels: mild moderate none severe
> factor(pain, exclude = NA) # NA is NOT a level.
[1] none     mild     moderate severe   <NA>
Levels: mild moderate none severe
> factor(pain, exclude = c(NA)) # NA is NOT a level.
[1] none     mild     moderate severe   <NA>
Levels: mild moderate none severe
> factor(pain, exclude = NULL) # NA is a level.
[1] none     mild     moderate severe   <NA>
Levels: mild moderate none severe <NA>
> factor(pain, exclude = "mild") # NA is a level.
[1] none     <NA>     moderate severe   <NA>
Levels: moderate none severe <NA>
> pain <- factor(pain, exclude = c("mild", NA))
> pain # mild and NA are NOT levels.
[1] none     <NA>     moderate severe   <NA>
Levels: moderate none severe
> #
> pain <- c("none", "mild", "moderate", "severe", NA)
> pain <- factor(pain, ordered = is.ordered(pain))
> pain # not an ordered variable
[1] none     mild     moderate severe   <NA>
Levels: mild moderate none severe
> class(pain)
```

```
[1] "factor"
> #
> income <- c("Lo", "Mid", "Hi", NA)
> income <- factor(income, ordered = is.ordered(income))
> income # not an ordered variable
[1] Lo   Mid  Hi   <NA>
Levels: Hi Lo Mid
> #
> stage <- c(1, 3, 4, 2, NA)
> factor(stage)
[1] 1    3    4    2    <NA>
Levels: 1 2 3 4
> stage # not an ordered variable
[1]  1  3  4  2 NA
```

14.2 無序因子物件 與 名義變數

Unordered Factor

R 中的 無序因子, 無序因素 (unordered factor) 類似於統計分析中的 名義變數 (nominal variable), 無序因子 中的類別水準 (level), 其離散值無大小順序的關係, 如性別的男與女. R 列印無序因子物件時, 類別水準 內建型式依照文字字母順序, 可以使用函式 `levels()` 查看 無序因子物件 的 類別水準; 也可以使用函式 `levels()` 設定列印無序因子物件 類別水準 的顯示次序, R 內建顯示次序是依照文字字母或數字排次序, 可以使用函式 `levels()` 指令, 改變顯示因子物件的次序或方向.

在統計模型中常使用 類別變數 作爲解釋變數, 常常必須令 無序因子物件 或 類別變數的某一個類別水準爲 參照水準 (reference level), 以便建構類別型解釋變數內不同類別水準的 對照比較 (contrast comparison). 使用函式 `relevel()`, 可以改變無序因子 類別水準 的參考水準.

建構類別水準函式爲

```
> levels(x)
> relevel(x, ref, ...)
```

其中引數

- x: 爲原始變數向量, 通常爲文字向量.
- ref: 設定參考水準.

使用 as.integer() 可將類別變數轉換成數值整數, 此數值整數從 1 到 類別水準的整數, 且依照原有的類別水準依序給整數, 使用上必須注意此差異.

```
> ## unorder factor
> ## level(), relevel()
> gender <- c("M", "F", "M", "M", "F")
> gender <- factor(gender)
> gender
[1] M F M M F
Levels: F M
> levels(gender)
[1] "F" "M"
> #
> levels(gender) <- c("Female", "Male")
> gender
[1] Male   Female Male   Male   Female
Levels: Female Male
> #
> hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
> hypertension <- factor(hypertension)
> hypertension
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Hi Lo Mod
> relevel(hypertension, ref = "Lo") # reset a reference level
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Lo Hi Mod
> #
> # keep same level words
> hypertension
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Hi Lo Mod
> levels(hypertension)
[1] "Hi"  "Lo"  "Mod"
> levels(hypertension) <- c("High", "Low", "Moderate")
> hypertension
[1] Low      Moderate High     Moderate Low      High     Low
Levels: High Low Moderate
> #
> ## convert to numerical values
> hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
> hypertension <- factor(hypertension)
> levels(hypertension)
[1] "Hi"  "Lo"  "Mod"
> hypertension
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
```

```
Levels: Hi Lo Mod
> as.integer(hypertension)
[1] 2 3 1 3 2 1 2
> #
> ## convert to numerical values
> hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
> hypertension <- factor(hypertension)
> hypertension
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Hi Lo Mod
> levels(hypertension) <- list("Low" = "Lo",
                               "Moderate" = "Mod",
                               "High" = "Hi")
> hypertension
[1] Low      Moderate High     Moderate Low      High     Low
Levels: Low Moderate High
> as.integer(hypertension)
[1] 1 2 3 2 1 3 1
> ## convert to numerical values
> pain <- c(7, 8, 6, 6, 8, 7)
> pain <- factor(pain)
> pain
[1] 7 8 6 6 8 7
Levels: 6 7 8
> as.integer(pain)
[1] 2 3 1 1 3 2
```

14.3 有序因子物件 與 順序變數 Ordered Factor

R 中的 有序因子 (ordered factor) 類似於統計分析中的 順序變數 (ordinal variable), 因子物件 或 順序變數的 類別水準 (level) 通常有自己的自然大小順序差異, 並且有可能在統計分析中進一步考慮將大小順序概念納入統計模型中. 可以使用函式 `ordered()` 來產生有序因子物件. 函式 `ordered()` 與函式 `levels()` 二者基本上類似.

```
> ordered(x, levels, ...)
```

其中引數

- x: 爲原始變數向量, 通常爲文字向量.

- 內建是依照文字字母排大小順序或數字排大小順序.
- `level`: 設定或改變大小順序水準.

R 內建大小的順序是依照文字字母排大小順序或數字排大小順序, 可以使用函式內引數 `levels` 指令, 改變大小順序或大小方向. R 可以使用函式 `level()` 合併函式 `ordered` 改變大小順序或大小方向. 多數情況下, 有序因子物件 和 無序因子物件 的唯一差別, 在於 有序因子物件 在 R 中顯示的時候, 對應了各別類別水準的順序, 且在應用在統計模型分析時, 這種差異的意義更明顯.

```
> ## ordered  factor
> hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
> hypertension <- factor(hypertension)
> hypertension
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Hi Lo Mod
> hypertension <- ordered(hypertension)
> hypertension # not as a natural order
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Hi < Lo < Mod
> hypertension <- ordered(hypertension,
                          levels = c("Lo", "Mod", "Hi"))
> hypertension # reset as a natural order
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Lo < Mod < Hi
> #
> hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
> hypertension <- factor(hypertension) # an unordered factor
> levels(hypertension) <- c("Low", "Moderate", "High")
> hypertension # an unordered factor
[1] Moderate High     Low      High     Moderate Low      Moderate
Levels: Low Moderate High
> hypertension <- ordered(hypertension)
> hypertension # reset as a natural order
[1] Moderate High     Low      High     Moderate Low      Moderate
Levels: Low < Moderate < High
> #
> hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
> hypertension <- factor(hypertension,
                         ordered = is.ordered(hypertension))
> hypertension # not as a natural order
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Hi Lo Mod
> #
> hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
> hypertension <- factor(hypertension,
```

```
                        levels = c("Lo", "Mod", "Hi"),
                        ordered = is.ordered(hypertension))
> hypertension # reset as a natural order
[1] Lo  Mod Hi  Mod Lo  Hi  Lo
Levels: Lo Mod Hi
```

14.4 連續型變數轉換成類別變數函式: cut()

在資料分析中, 常常將 連續型變數 (continuous variable) 轉換成 類別型變數 (discrete variable), 但許多統計學者並不這樣建議. 使用函式 `cut()` 可以將 連續型變數) (continuous variable 分割成 類別型變數 (discrete variable). 配合使用函式 `factor()` 或 `ordered`, 可將連續型變數轉換成各種類別變數. 函式 `cut(x,...)` 將連續型數值向量轉變成因子向量:

```
> cut(x, breaks, labels = NULL,
      include.lowest = FALSE, right = TRUE, dig.lab = 3,
      ordered_result = FALSE, ...)
```

`cut(x,...)` 常見的引數:

- `x`: 爲連續型數值向量.
- `breaks`:
  - 若 `breaks = k`, $k \geq 2$, 則表示將 `x` 分成 k 類別水準.
  - 若 `breaks = y`, `y` 爲向量, 代表類別水準的切割點.
- `labels`: 代表類別水準的標記, R 設定爲 (a, b], a, b 爲分割點數值.
- `include.lowest = TRUE`: 代表類別水準的最左切割區間爲包含最小值.
- 配合 `right = FALSE`.
- `right = FALSE`: 代表類別水準的切割區間爲 $[y_i, y_{i+1})$, 左側爲中括號, 右側爲小括號.

- `right = TRUE`: 代表類別水準的切割區間爲 $(y_i, y_{i+1}]$, 左側爲小括號, 右側爲中括號, 最右切割區間爲包含最大值.
- `dig.lab = 3` 呈現分割點數值的小數位數.
- `ordered_result = FALSE`: R 設定代表類別水準設定爲無序因子, 若改成 `ordered_result = TRUE` 代表類別水準設定爲有序因子.

```
> ## cut() unordered facor
> x.vec <- 0:12
> y.vec <- cut(x.vec, breaks = 4)
> y.vec
 [1] (-0.012,3] (-0.012,3] (-0.012,3] (-0.012,3] (3,6]      (3,6]
 [7] (3,6]      (6,9]      (6,9]      (6,9]      (9,12]     (9,12]
[13] (9,12]
Levels: (-0.012,3] (3,6] (6,9] (9,12]
> y.vec <- cut(x.vec, breaks = seq(0, 12, by = 4))
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] (4,8] (8,12]
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12))
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] (4,8] (8,12]
> y.vec <- cut(x.vec, breaks = 5)
> y.vec
 [1] (-0.012,2.4] (-0.012,2.4] (-0.012,2.4] (2.4,4.8]   (2.4,4.8]
 [6] (4.8,7.2]    (4.8,7.2]    (4.8,7.2]    (7.2,9.6]   (7.2,9.6]
[11] (9.6,12]     (9.6,12]     (9.6,12]
Levels: (-0.012,2.4] (2.4,4.8] (4.8,7.2] (7.2,9.6] (9.6,12]
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12), include.lowest = T)
> y.vec
 [1] [0,4]  [0,4]  [0,4]  [0,4]  [0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: [0,4] (4,8] (8,12]
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12), include.lowest = F)
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] (4,8] (8,12]
> #
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12), right = F)
> y.vec
 [1] [0,4)  [0,4)  [0,4)  [0,4)  [4,8)  [4,8)  [4,8)  [4,8)  [8,12) [8,12)
[11] [8,12) [8,12) <NA>
```

```
Levels: [0,4) [4,8) [8,12)
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12), right = T)
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] (4,8] (8,12]
> #
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12),
               include.lowest = T, right = F)
> y.vec
 [1] [0,4)  [0,4)  [0,4)  [0,4)  [4,8)  [4,8)  [4,8)  [4,8)  [8,12] [8,12]
[11] [8,12] [8,12] [8,12]
Levels: [0,4) [4,8) [8,12]
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12),
               include.lowest = F, right = T)
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] (4,8] (8,12]
> #
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12),
+              include.lowest = T, right = T)
> y.vec
 [1] [0,4]  [0,4]  [0,4]  [0,4]  [0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: [0,4] (4,8] (8,12]
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12),
               include.lowest = F, right = F)
> y.vec
 [1] [0,4)  [0,4)  [0,4)  [0,4)  [4,8)  [4,8)  [4,8)  [4,8)  [8,12) [8,12)
[11] [8,12) [8,12) <NA>
Levels: [0,4) [4,8) [8,12)
> #
> y.vec <- cut(x.vec, breaks = 5,
               include.lowest = T, right = F, dig.lab = 0)
> y.vec
 [1] [-0.012,2.4) [-0.012,2.4) [-0.012,2.4) [2.4,4.8)    [2.4,4.8)
 [6] [4.8,7.2)    [4.8,7.2)    [4.8,7.2)    [7.2,9.6)    [7.2,9.6)
[11] [9.6,12]     [9.6,12]     [9.6,12]
Levels: [-0.012,2.4) [2.4,4.8) [4.8,7.2) [7.2,9.6) [9.6,12]
> y.vec <- cut(x.vec, breaks = 5,
               include.lowest = F, right = T, dig.lab = 0)
> y.vec
 [1] (-0.012,2.4] (-0.012,2.4] (-0.012,2.4] (2.4,4.8]    (2.4,4.8]
 [6] (4.8,7.2]    (4.8,7.2]    (4.8,7.2]    (7.2,9.6]    (7.2,9.6]
[11] (9.6,12]     (9.6,12]     (9.6,12]
Levels: (-0.012,2.4] (2.4,4.8] (4.8,7.2] (7.2,9.6] (9.6,12]
> #
> y.vec <- cut(x.vec, breaks = 5,
               include.lowest = T, right = T, dig.lab = 0)
```

```
> y.vec
 [1] [-0.012,2.4] [-0.012,2.4] [-0.012,2.4] (2.4,4.8]    (2.4,4.8]
 [6] (4.8,7.2]    (4.8,7.2]    (4.8,7.2]    (7.2,9.6]    (7.2,9.6]
[11] (9.6,12]     (9.6,12]     (9.6,12]
Levels: [-0.012,2.4] (2.4,4.8] (4.8,7.2] (7.2,9.6] (9.6,12]
> y.vec <- cut(x.vec, breaks = 5,
               include.lowest = F, right = F, dig.lab = 0)
> y.vec
 [1] [-0.012,2.4) [-0.012,2.4) [-0.012,2.4) [2.4,4.8)    [2.4,4.8)
 [6] [4.8,7.2)    [4.8,7.2)    [4.8,7.2)    [7.2,9.6)    [7.2,9.6)
[11] [9.6,12)     [9.6,12)     [9.6,12)
Levels: [-0.012,2.4) [2.4,4.8) [4.8,7.2) [7.2,9.6) [9.6,12)
>
> ## cut() ordered factor
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12))
> y.vec <- ordered(y.vec, labels = levels(y.vec))
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] < (4,8] < (8,12]
> ##
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12), include.lowest = T)
> y.vec
 [1] [0,4]  [0,4]  [0,4]  [0,4]  [0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: [0,4] (4,8] (8,12]
> y.vec <- ordered(y.vec, labels = levels(y.vec))
> y.vec
 [1] [0,4]  [0,4]  [0,4]  [0,4]  [0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: [0,4] < (4,8] < (8,12]
> #
> y.vec <- cut(x.vec, breaks = c(0, 4, 8, 12), include.lowest = F)
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] (4,8] (8,12]
> y.vec <- ordered(y.vec, labels = levels(y.vec))
> y.vec
 [1] <NA>   (0,4]  (0,4]  (0,4]  (0,4]  (4,8]  (4,8]  (4,8]  (4,8]  (8,12]
[11] (8,12] (8,12] (8,12]
Levels: (0,4] < (4,8] < (8,12]
> #
> y.vec <- cut(x.vec, breaks = 5, include.lowest = T)
> y.vec
 [1] [-0.012,2.4] [-0.012,2.4] [-0.012,2.4] (2.4,4.8]    (2.4,4.8]
 [6] (4.8,7.2]    (4.8,7.2]    (4.8,7.2]    (7.2,9.6]    (7.2,9.6]
[11] (9.6,12]     (9.6,12]     (9.6,12]
Levels: [-0.012,2.4] (2.4,4.8] (4.8,7.2] (7.2,9.6] (9.6,12]
> y.vec <- ordered(y.vec, labels = levels(y.vec))
```

```
> y.vec
 [1] [-0.012,2.4] [-0.012,2.4] [-0.012,2.4] (2.4,4.8]    (2.4,4.8]
 [6] (4.8,7.2]    (4.8,7.2]    (4.8,7.2]    (7.2,9.6]    (7.2,9.6]
[11] (9.6,12]     (9.6,12]     (9.6,12]
5 Levels: [-0.012,2.4] < (2.4,4.8] < (4.8,7.2] < ... < (9.6,12]
> y.vec <- cut(x.vec, breaks = 5, include.lowest = F)
> y.vec
 [1] (-0.012,2.4] (-0.012,2.4] (-0.012,2.4] (2.4,4.8]    (2.4,4.8]
 [6] (4.8,7.2]    (4.8,7.2]    (4.8,7.2]    (7.2,9.6]    (7.2,9.6]
[11] (9.6,12]     (9.6,12]     (9.6,12]
Levels: (-0.012,2.4] (2.4,4.8] (4.8,7.2] (7.2,9.6] (9.6,12]
> y.vec <- ordered(y.vec, labels = levels(y.vec))
> y.vec
 [1] (-0.012,2.4] (-0.012,2.4] (-0.012,2.4] (2.4,4.8]    (2.4,4.8]
 [6] (4.8,7.2]    (4.8,7.2]    (4.8,7.2]    (7.2,9.6]    (7.2,9.6]
[11] (9.6,12]     (9.6,12]     (9.6,12]
5 Levels: (-0.012,2.4] < (2.4,4.8] < (4.8,7.2] < ... < (9.6,12]
```

14.5 因子物件函式: gl()

因子物件函式 `gl()` 與另一個與類似的函式 `rep()`, 可以產生因子物件, 因爲 2 種函式都可產生重複性數值, 且可有許多不同的變化.

```
> gl(n, k, length = n*k, labels = 1:n, ordered = FALSE)
```

函式 `gl()` 內的引數:

`n` 整數, 設定因子變數的類別水準之數目.

`k` 整數, 設定因子變數的類別水準之重複數目.

`labels` 設定因子變數的類別水準之名稱, 內設爲 `1:n`.

`ordered = FALSE` 設定類別水準沒有大小順序, 若改成 `ordered = TRUE` 則產生順序變數.

```
> ## gl()
> ## first control, then treatment:
> gl(n = 2, k = 3)
[1] 1 1 1 2 2 2
Levels: 1 2
```

```
> gl(2, 3)
[1] 1 1 1 2 2 2
Levels: 1 2
> gl(n = 2, k = 3, label = c("Control", "Treat"))
[1] Control Control Control Treat   Treat   Treat
Levels: Control Treat
> #
> ## alternating 1s and 2s
> gl(n = 2, k = 1, length = 5)
[1] 1 2 1 2 1
Levels: 1 2
> #
> ## alternating pairs of 1s and 2s
> gl(n = 2, k = 2, length = 5)
[1] 1 1 2 2 1
Levels: 1 2
> gl(2, 2, 5)
[1] 1 1 2 2 1
Levels: 1 2
> ## Clinical Trials
> gl(n = 3, k = 4, length = 12,
+    label = c("Control", "Active", "Test"),
+    order = FALSE)
 [1] Control Control Control Control Active  Active
 [7] Active  Active  Test    Test    Test    Test
Levels: Control Active Test
> gl(n = 3, k = 4, length = 20,
+    label = c("D0", "D1", "D2"),
+    order = TRUE)
 [1] D0 D0 D0 D0 D1 D1 D1 D1 D2 D2 D2 D2 D0 D0 D0 D0 D1 D1
[19] D1 D1
Levels: D0 < D1 < D2
```