Subsetting and Merging

September 29, 2014

Data Preparation I: Subsetting and Merging

- Subsetting a vector
- Subsetting a matrix
- Subsetting a data frame
- Merging data

Subsetting a vector

```
x <- c(1:10)
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
## Use square brackets to get one element of the vector
x[1]
```

```
## [1] 1
```

```
x[5]
```

```
## [1] 5
```

```
## Use a logical argument
x[x>6]
```

```
## [1]  7  8  9 10
```

Subsetting a vector

```
## Using a logical vector to subset x
good <- x>6
good
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TR
```

```
x[good]
```

```
## [1]  7  8  9 10
```

Subsetting a matrix

```
x <- matrix(c(1:10), nrow=5, ncol=2)
x
```

```
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    8
## [4,]    4    9
## [5,]    5   10
```

```
## Get an element of the matrix
x[2,1]
```

```
## [1] 2
```

Subsetting a matrix

```
## Get a row of the matrix
x[3,]
```

```
## [1] 3 8
```

```
## Get a column of the matrix
x[,2]
```

```
## [1]  6  7  8  9 10
```

Subsetting a data frame

```
mydata
```

```
##     id age gender
## 1    1  23      1
## 2    2  19      1
## 3    3  34      0
## 4    4  31      1
## 5    5  28      1
## 6    6  29      0
## 7    7  20      0
## 8    8  30      0
## 9    9  26      1
## 10 10  19      1
```

Subsetting a data frame

```
## Subsetting variables
mydata[,2:3]
```

```
##     age gender
## 1    23      1
## 2    19      1
## 3    34      0
## 4    31      1
## 5    28      1
## 6    29      0
## 7    20      0
## 8    30      0
## 9    26      1
## 10   19      1
```

```
## Using logical argument
mydata[age > 25, ]
```

Subsetting a data frame

```
## Using logical argument
mydata[age > 25, ]
```

```
##    id age gender
## 3   3  34      0
## 4   4  31      1
## 5   5  28      1
## 6   6  29      0
## 8   8  30      0
## 9   9  26      1
```

Subsetting a data frame

Data with missing values

```
mydata
```

```
##     id age gender
## 1    1  23      1
## 2    2  19      1
## 3    3  34      0
## 4    4  31     NA
## 5    5  28      1
## 6    6  29      0
## 7    7  NA      0
## 8    8  30      0
## 9    9  26      1
## 10  10  19      1
```

Subsetting a data frame

Removing missing data. Two ways of doing it:

```
## Using complete.cases()
good <- complete.cases(mydata)
good
```

```
##   [1]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TR
```

```
mydata[good, ]
```

```
##    id age gender
## 1   1  23      1
## 2   2  19      1
## 3   3  34      0
## 5   5  28      1
## 6   6  29      0
## 8   8  30      0
## 9   9  26      1
```

Subsetting a data frame

```
## Using is.na()
bad <- is.na(mydata)
mydata[!bad[,2], ]
```

```
##     id age gender
## 1    1  23      1
## 2    2  19      1
## 3    3  34      0
## 4    4  31     NA
## 5    5  28      1
## 6    6  29      0
## 8    8  30      0
## 9    9  26      1
## 10 10  19      1
```

Subsetting a data frame

```
mydata[!is.na(mydata$age), ]
```

```
##     id age gender
## 1    1  23      1
## 2    2  19      1
## 3    3  34      0
## 4    4  31     NA
## 5    5  28      1
## 6    6  29      0
## 8    8  30      0
## 9    9  26      1
## 10  10  19      1
```

Merging data

Example 1

```
data1
```

```
##   id age
## 1  1  23
## 2  2  19
## 3  3  34
## 4  4  31
## 5  5  28
```

```
data2
```

```
##   id gender
## 1  1      1
## 2  2      1
## 3  3      0
## 4  4      0
## 5  5      1
```

Merging data

Example 1

```
merge(data1, data2, by="id")
```

```
##   id age gender
## 1  1  23      1
## 2  2  19      1
## 3  3  34      0
## 4  4  31      0
## 5  5  28      1
```

Merging data

Example 2

```
data2
```

```
##   id gender
## 1  1      1
## 2  2      1
## 3  3      0
## 4  4      0
```

```
merge(data1, data2, by="id")
```

```
##   id age gender
## 1  1  23      1
## 2  2  19      1
## 3  3  34      0
## 4  4  31      0
```

Merging data

Example 2

```
merge(data1, data2, by="id", all.x=TRUE)
```

```
##   id age gender
## 1  1  23      1
## 2  2  19      1
## 3  3  34      0
## 4  4  31      0
## 5  5  28     NA
```

Merging data

Example 3

```
data1
```

```
##    id.student id.class math.score gender
## 1           1      101         600      1
## 2           2      101         700      1
## 3           3      101         550      0
## 4           4      101         790      1
## 5           5      201         450      1
## 6           6      201         640      0
## 7           7      201         580      0
## 8           8      301         670      0
## 9           9      301         720      1
## 10         10      301         590      1
```

Merging data

Example 3

```
data2
```

```
##   id.class ses.class teach.exp
## 1      101         5        11
## 2      201         4         3
## 3      301         3         7
```

Merging data

Example 3

```
merge(data1, data2, by="id.class")
```

```
##    id.class id.student math.score gender ses.class teach
## 1       101          1         600      1         5
## 2       101          2         700      1         5
## 3       101          3         550      0         5
## 4       101          4         790      1         5
## 5       201          5         450      1         4
## 6       201          6         640      0         4
## 7       201          7         580      0         4
## 8       301          8         670      0         3
## 9       301          9         720      1         3
## 10      301         10         590      1         3
```