Generating siblings

SDC
http://bendixcarstensen.com/
June 2017
1

Compiled Wednesday 21st June, 2017, 14:34
from: /home/bendix/sdc/coll/jebe/r/siblings.tex

Bendix Carstensen Steno Diabetes Center, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
bcar0029@regionh.dk b@bxc.dk
http://BendixCarstensen.com

Claus Thorn EkstrÃ¿m Department of Biostatistics, University of Copenhagen

Contents

1 The problem

We have informtion on persons' id and the the id of the persons' mother and father. From this we want to derive the id of the persons' full respectively half siblings.

This is an indexing problem. Kindly solved by Claus Ekstrøm.

2 The solution

2.1 Some clunky demo code

Here is some sample code that read a small data frame, followed by a demonstration of the workings.

```
> cat('
+ id fid mid
+   1   NA  0
+   2   9  NA
+   3   1   2
+   4   1   2
+  12   7   6
+  13   5   6
+  14   5   6
+  15   5   8', file="netwk.txt" )
> ( indata <- read.table("netwk.txt", header=TRUE) )
  id fid mid
1  1  NA   0
2  2   9  NA
3  3   1   2
4  4   1   2
5 12   7   6
6 13   5   6
7 14   5   6
8 15   5   8
> idat <- indata
> ( foffspring <- by(indata, indata$fid, function(i) { i$id }, simplify=FALSE) )
indata$fid: 1
[1] 3 4
------------------------------------------------------------------------
indata$fid: 5
[1] 13 14 15
------------------------------------------------------------------------
indata$fid: 7
[1] 12
------------------------------------------------------------------------
indata$fid: 9
[1] 2
> ( moffspring <- by(indata, indata$mid, function(i) { i$id }, simplify=FALSE) )
indata$mid: 0
[1] 1
------------------------------------------------------------------------
indata$mid: 2
[1] 3 4
------------------------------------------------------------------------
indata$mid: 6
[1] 12 13 14
------------------------------------------------------------------------
indata$mid: 8
[1] 15
```

```
> foffspring[2]
$`5`
[1] 13 14 15
> foffspring[[2]]
[1] 13 14 15
```

Any sib included

```
> sibs <- sapply( 1:nrow(indata),
+                 function(i) {
+  res <- c()
+  if( !is.na(indata$fid[i]) ) res <- c(res, unlist(foffspring[paste0(indata$fid[i])]))
+  if( !is.na(indata$mid[i]) ) res <- c(res, unlist(moffspring[paste0(indata$mid[i])]))
+  unique(res[res != indata$id[i]])
+                             },
+                 simplify=TRUE )
> sibs
[[1]]
integer(0)

[[2]]
integer(0)

[[3]]
[1] 4

[[4]]
[1] 3

[[5]]
[1] 13 14

[[6]]
[1] 14 15 12

[[7]]
[1] 13 15 12

[[8]]
[1] 13 14
```

Sibs by father:

```
> fsibs <- sapply( 1:nrow(indata),
+                  function(i) {
+  res <- c()
+  if( !is.na(indata$fid[i]) ) res <- c(res, unlist(foffspring[paste0(indata$fid[i])]))
+  unique(res[res != indata$id[i]])
+                             },
+                  simplify=TRUE )
> fsibs
[[1]]
NULL

[[2]]
integer(0)

[[3]]
[1] 4

[[4]]
[1] 3
```

```
[[5]]
integer(0)

[[6]]
[1] 14 15

[[7]]
[1] 13 15

[[8]]
[1] 13 14
```

Sibs by mother:

```
> msibs <- sapply( 1:nrow(indata),
+                  function(i) {
+  res <- c()
+  if( !is.na(indata$fid[i]) ) res <- c(res, unlist(moffspring[paste0(indata$mid[i])]) )
+  unique(res[res != indata$id[i]])
+                              },
+                  simplify=TRUE )
> msibs
[[1]]
NULL

[[2]]
NULL

[[3]]
[1] 4

[[4]]
[1] 3

[[5]]
[1] 13 14

[[6]]
[1] 12 14

[[7]]
[1] 12 13

[[8]]
integer(0)
```

Full sibs only:

```
> Fsibs <- sapply( 1:nrow(indata),
+                  function(i) {
+  res <- hasf <- hasm <- c()
+  if( !is.na(indata$fid[i]) ) hasf <- unlist(foffspring[paste0(indata$fid[i])])
+  if( !is.na(indata$mid[i]) ) hasm <- unlist(moffspring[paste0(indata$mid[i])])
+  res <- c( res, intersect( hasf, hasm ) )
+  unique(res[res != indata$id[i]])
+                              },
+                  simplify=TRUE )
> Fsibs
[[1]]
integer(0)

[[2]]
NULL
```

```
[[3]]
[1] 4

[[4]]
[1] 3

[[5]]
integer(0)

[[6]]
[1] 14

[[7]]
[1] 13

[[8]]
integer(0)
```

Full sibs only:

```
> Hsibs <- sapply( 1:nrow(indata),
+                  function(i) {
+  res <- hasf <- hasm <- c()
+  if( !is.na(indata$fid[i]) ) hasf <- unlist(foffspring[paste0(indata$fid[i])])
+  if( !is.na(indata$mid[i]) ) hasm <- unlist(moffspring[paste0(indata$mid[i])])
+  res <- c( res, setdiff( union(hasf,hasm), intersect(hasf,hasm) ) )
+  unique(res[res != indata$id[i]])
+                               },
+                  simplify=TRUE )
> Hsibs
[[1]]
integer(0)

[[2]]
integer(0)

[[3]]
integer(0)

[[4]]
integer(0)

[[5]]
[1] 13 14

[[6]]
[1] 15 12

[[7]]
[1] 15 12

[[8]]
[1] 13 14
```

Finally we name the resulting lists with the correct ids:

```
> names(  sibs ) <-
+ names( Hsibs ) <-
+ names( Fsibs ) <-
+ names( msibs ) <-
+ names( fsibs ) <- indata$id
> sibs
```

```
$`1`
integer(0)

$`2`
integer(0)

$`3`
[1] 4

$`4`
[1] 3

$`12`
[1] 13 14

$`13`
[1] 14 15 12

$`14`
[1] 13 15 12

$`15`
[1] 13 14
```

```
> Hsibs
```

```
$`1`
integer(0)

$`2`
integer(0)

$`3`
integer(0)

$`4`
integer(0)

$`12`
[1] 13 14

$`13`
[1] 15 12

$`14`
[1] 15 12

$`15`
[1] 13 14
```

```
> Fsibs
```

```
$`1`
integer(0)

$`2`
NULL

$`3`
[1] 4

$`4`
[1] 3

$`12`
integer(0)
```

```
$`13`
[1] 14

$`14`
[1] 13

$`15`
integer(0)
```

whereby we can construct a dataframe with sibs / helfsibs etc:

```
> mx <- 5
> NAfill <- function( x ) c(x,rep(NA,mx))[1:mx]
> db <- data.frame( id = as.numeric(names(sibs)),
+                   do.call( rbind, lapply( sibs, NAfill ) ) )
> names( db )[-1] <- paste("s",1:mx,sep="")
> db
   id s1 s2 s3 s4 s5
1   1 NA NA NA NA NA
2   2 NA NA NA NA NA
3   3  4 NA NA NA NA
4   4  3 NA NA NA NA
12 12 13 14 NA NA NA
13 13 14 15 12 NA NA
14 14 13 15 12 NA NA
15 15 13 14 NA NA NA

> ms <- data.frame( id = as.numeric(names(msibs)),
+                   do.call( rbind, lapply( msibs, NAfill ) ) )
> names( ms )[-1] <- paste("m",1:mx,sep="")
> ms

   id m1 m2 m3 m4 m5
1   1 NA NA NA NA NA
2   2 NA NA NA NA NA
3   3  4 NA NA NA NA
4   4  3 NA NA NA NA
12 12 13 14 NA NA NA
13 13 12 14 NA NA NA
14 14 12 13 NA NA NA
15 15 NA NA NA NA NA

> fs <- data.frame( id = as.numeric(names(fsibs)),
+                   do.call( rbind, lapply( fsibs, NAfill ) ) )
> names( ms )[-1] <- paste("f",1:mx,sep="")
> fs

   id X1 X2 X3 X4 X5
1   1 NA NA NA NA NA
2   2 NA NA NA NA NA
3   3  4 NA NA NA NA
4   4  3 NA NA NA NA
12 12 NA NA NA NA NA
13 13 14 15 NA NA NA
14 14 13 15 NA NA NA
15 15 13 14 NA NA NA

> Fs <- data.frame( id = as.numeric(names(Fsibs)),
+                   do.call( rbind, lapply( Fsibs, NAfill ) ) )
> names( Fs )[-1] <- paste("F",1:mx,sep="")
> Hs <- data.frame( id = as.numeric(names(Hsibs)),
+                   do.call( rbind, lapply( Hsibs, NAfill ) ) )
> names( Hs )[-1] <- paste("H",1:mx,sep="")
> merge( Fs, Hs )
```

```
  id F1 F2 F3 F4 F5 H1 H2 H3 H4 H5
1  1 NA NA NA NA NA NA NA NA NA NA
2  2 NA NA NA NA NA NA NA NA NA NA
3  3  4 NA NA NA NA NA NA NA NA NA
4  4  3 NA NA NA NA NA NA NA NA NA
5 12 NA NA NA NA NA 13 14 NA NA NA
6 13 14 NA NA NA NA 15 12 NA NA NA
7 14 13 NA NA NA NA 15 12 NA NA NA
8 15 NA NA NA NA NA 13 14 NA NA NA
```

3 Based on `data.table`

```
> indata <- idat
> library(data.table)
> setDT(indata)[,msib  := .(list(id)), by = "mid"][
+               ,msibs := mapply(setdiff, msib, id)][
+               ,fsib  := .(list(id)), by = "fid"][
+               ,fsibs := mapply(setdiff, fsib, id)][
+               ,sibs  := mapply(union, msibs, fsibs)][
+               ,c("msib","msibs", "fsib", "fsibs") := NULL]
```

```
   id fid mid     sibs
1:  1  NA   0
2:  2   9  NA
3:  3   1   2        4
4:  4   1   2        3
5: 12   7   6    13,14
6: 13   5   6 12,14,15
7: 14   5   6 12,13,15
8: 15   5   8    13,14
```

```
> indata
```

```
   id fid mid     sibs
1:  1  NA   0
2:  2   9  NA
3:  3   1   2        4
4:  4   1   2        3
5: 12   7   6    13,14
6: 13   5   6 12,14,15
7: 14   5   6 12,13,15
8: 15   5   8    13,14
```

```
> str( indata )
```

```
Classes ‘data.table’ and 'data.frame':        8 obs. of  4 variables:
 $ id  : int  1 2 3 4 12 13 14 15
 $ fid : int  NA 9 1 1 7 5 5 5
 $ mid : int  0 NA 2 2 6 6 6 8
 $ sibs:List of 8
  ..$ : int
  ..$ : int
  ..$ : int 4
  ..$ : int 3
  ..$ : int  13 14
  ..$ : int  12 14 15
  ..$ : int  12 13 15
  ..$ : int  13 14
 - attr(*, ".internal.selfref")=<externalptr>
```

Here are the full sibs:

```
> (DT <- data.table(indata))
```

```
   id fid mid     sibs
1:  1  NA   0
2:  2   9  NA
3:  3   1   2        4
4:  4   1   2        3
5: 12   7   6    13,14
6: 13   5   6 12,14,15
7: 14   5   6 12,13,15
8: 15   5   8    13,14
```

```
> ( sibDT = DT[!is.na(fid) & !is.na(mid),
+              CJ(id = id, sid = id)[id != sid]
+             ,by=.(fid, mid)] )
```

```
   fid mid id sid
1:   1   2  3   4
2:   1   2  4   3
3:   5   6 13  14
4:   5   6 14  13
```

This should give full sibs:

```
> hsibDT = melt(DT, id = id)[!is.na(value),
+               CJ(id = id, hsid = id)[id != hsid]
+               , by=.(ptype = variable, pid = value)][!sibDT, on=.(id, hsid = sid)]
```

3.1 Implemented as an R function properly modulized

Well. . .

4 The practicalities

Well, well

References