Using apply functions in R

Svetlana Eden

April 25, 2012

1 Load Hmisc

```
>   library(Hmisc)
```

2 Prepare dataframe

```
>   data = data.frame(id = c(333, 333, 333, 11, 4444, 4444, 1, 22, 22, 55, 55, 55, 55, 55,
+                            666, 666, 666),
+                     date=c("2006/04/07", "2005/04/17", "2001/12/13", NA, NA, "1994/08/27",
+                            "1995/09/09", "1992/05/15", "1998/11/19", NA, "2000/04/13",
+                            "1968/04/16", NA, "2011/07/07", "2012/10/22", "2011/02/14",
+                            "2005/05/24"))
>   data$date = as.Date(data$date)
>   data

     id       date
1   333 2006-04-07
2   333 2005-04-17
3   333 2001-12-13
4    11       <NA>
5  4444       <NA>
6  4444 1994-08-27
7     1 1995-09-09
8    22 1992-05-15
9    22 1998-11-19
10   55       <NA>
11   55 2000-04-13
12   55 1968-04-16
13   55       <NA>
14   55 2011-07-07
15  666 2012-10-22
16  666 2011-02-14
17  666 2005-05-24
```

2.1 Set some values to NA randomly

```
>   set.seed(20120425)
>   data$q1 = abs(round(10*rnorm(nrow(data))))
>   data$q1[sample(1:nrow(data), 4)] = NA
>   data$q2 = abs(round(10*rnorm(nrow(data))))
>   data$q2[sample(1:nrow(data), 6)] = NA
>   data$q3 = abs(round(10*rnorm(nrow(data))))
```

```
>   data[13, c("q1", "q2")] = NA
>   data
```

```
     id       date q1 q2 q3
1   333 2006-04-07 NA  9  1
2   333 2005-04-17  6 NA  7
3   333 2001-12-13  5 10 11
4    11       <NA>  7 24 10
5  4444       <NA>  6  8  7
6  4444 1994-08-27 13  4 15
7     1 1995-09-09  6  0  1
8    22 1992-05-15 11 NA 23
9    22 1998-11-19 10 NA 10
10   55       <NA>  2  5  2
11   55 2000-04-13 NA NA 10
12   55 1968-04-16  3  3  1
13   55       <NA> NA NA  1
14   55 2011-07-07  6 16  7
15  666 2012-10-22  2 NA  9
16  666 2011-02-14 NA  9  5
17  666 2005-05-24 NA  3 11
```

2.2 Assign labels

```
>   label(data$id) = "Patient ID"
>   label(data$date) = "Observation date"
>   label(data$q1) = "qestion 1"
>   label(data$q2) = "qestion 2"
>   label(data$q3) = "qestion 3"
```

3 Print Its Labels Using sapply()

3.1 Print Labels Without sapply()

```
>   label(data$q1)
```

```
[1] "qestion 1"
```

```
>   n = "q1"
>   label(data[[n]])
```

```
[1] "qestion 1"
```

```
>   for (n in names(data)){
+     cat(n, ":\n   ", label(data[[n]]), "\n")
+   }
```

```
id :
    Patient ID
date :
    Observation date
q1 :
    qestion 1
q2 :
    qestion 2
q3 :
    qestion 3
```

3.2 Use sapply()

```
>   sout1 = sapply(data, label)
>   sout1

             id              date                q1                q2
  "Patient ID" "Observation date"       "qestion 1"       "qestion 2"
             q3
   "qestion 3"

>   sout1[1:2]

             id              date
  "Patient ID" "Observation date"

>   sout1[c("id", "date")]

             id              date
  "Patient ID" "Observation date"

>
```

4 Calculate proportion of missing using sapply()

4.1 Without sapply()

```
>   is.na(data$q1)

 [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
[13]  TRUE FALSE FALSE  TRUE  TRUE

>   sum(is.na(data$q1))

[1] 5

>   100*sum(is.na(data$q1))/nrow(data)

[1] 29.41176

>   for (n in names(data)){
+     tmp = 100*sum(is.na(data[[n]]))/nrow(data)
+     cat(n, ": ", tmp, "\n", sep="")
+   }

id: 0
date: 23.52941
q1: 29.41176
q2: 35.29412
q3: 0

>
```

4.2 Still without sapply(), if we want to keep the info about NAs

```
>   percMiss = rep(NA, ncol(data))
>   names(percMiss) = names(data)
>   for (n in names(data)){
+     percMiss[n] = 100*sum(is.na(data[[n]]))/nrow(data)
+   }
>   round(percMiss, 1)

  id date   q1   q2   q3
 0.0 23.5 29.4 35.3  0.0

>
```

4.3 With sapply()

```
>   sout2 = sapply(data, function(x){100*sum(is.na(x))/nrow(data)})
>   round(sout2, 1)

  id date   q1   q2   q3
 0.0 23.5 29.4 35.3  0.0
```

5 Find Percent Missing Using apply()

```
>   aout1 = apply(data, 2, function(x){100*sum(is.na(x))/nrow(data)})
>   aout1

      id     date       q1       q2       q3
 0.00000 23.52941 29.41176 35.29412  0.00000

>
```

6 Find Complete Records Using apply()

```
>   nonMissingRecords = apply(data, 1, function(x){all(!is.na(x))})
>   nonMissingRecords

 [1] FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE
[13] FALSE  TRUE FALSE FALSE FALSE

>   ccdata = data[nonMissingRecords, ]
>   ccdata

     id       date q1 q2 q3
3   333 2001-12-13  5 10 11
6  4444 1994-08-27 13  4 15
7     1 1995-09-09  6  0  1
12   55 1968-04-16  3  3  1
14   55 2011-07-07  6 16  7

>
```

7 Calculate Mean Score $(q1 + q2 + q3)$ Using apply()

7.1 Simple mean

```
>   data$totalscore = apply(data[, c("q1", "q2", "q3")], 1, mean)
>   data
```

```
     id       date q1 q2 q3 totalscore
1   333 2006-04-07 NA  9  1         NA
2   333 2005-04-17  6 NA  7         NA
3   333 2001-12-13  5 10 11   8.666667
4    11       <NA>  7 24 10  13.666667
5  4444       <NA>  6  8  7   7.000000
6  4444 1994-08-27 13  4 15  10.666667
7     1 1995-09-09  6  0  1   2.333333
8    22 1992-05-15 11 NA 23         NA
9    22 1998-11-19 10 NA 10         NA
10   55       <NA>  2  5  2   3.000000
11   55 2000-04-13 NA NA 10         NA
12   55 1968-04-16  3  3  1   2.333333
13   55       <NA> NA NA  1         NA
14   55 2011-07-07  6 16  7   9.666667
15  666 2012-10-22  2 NA  9         NA
16  666 2011-02-14 NA  9  5         NA
17  666 2005-05-24 NA  3 11         NA
```

7.2 Mean and argument na.rm

```
>   data$totalscore = apply(data[, c("q1", "q2", "q3")], 1, mean, na.rm=TRUE)
>   data
```

```
     id       date q1 q2 q3 totalscore
1   333 2006-04-07 NA  9  1   5.000000
2   333 2005-04-17  6 NA  7   6.500000
3   333 2001-12-13  5 10 11   8.666667
4    11       <NA>  7 24 10  13.666667
5  4444       <NA>  6  8  7   7.000000
6  4444 1994-08-27 13  4 15  10.666667
7     1 1995-09-09  6  0  1   2.333333
8    22 1992-05-15 11 NA 23  17.000000
9    22 1998-11-19 10 NA 10  10.000000
10   55       <NA>  2  5  2   3.000000
11   55 2000-04-13 NA NA 10  10.000000
12   55 1968-04-16  3  3  1   2.333333
13   55       <NA> NA NA  1   1.000000
14   55 2011-07-07  6 16  7   9.666667
15  666 2012-10-22  2 NA  9   5.500000
16  666 2011-02-14 NA  9  5   7.000000
17  666 2005-05-24 NA  3 11   7.000000
```

7.3 Assigning score based on the number of missing items

```
>   data$totalscore = apply(data[, c("q1", "q2", "q3")], 1,
+                           function(x){if (sum(is.na(x))>=2) NA else mean(x, na.rm=TRUE)})
>   data
```

```
     id       date q1 q2 q3 totalscore
1   333 2006-04-07 NA  9  1   5.000000
2   333 2005-04-17  6 NA  7   6.500000
3   333 2001-12-13  5 10 11   8.666667
4    11       <NA>  7 24 10  13.666667
5  4444       <NA>  6  8  7   7.000000
6  4444 1994-08-27 13  4 15  10.666667
7     1 1995-09-09  6  0  1   2.333333
8    22 1992-05-15 11 NA 23  17.000000
9    22 1998-11-19 10 NA 10  10.000000
10   55       <NA>  2  5  2   3.000000
11   55 2000-04-13 NA NA 10         NA
12   55 1968-04-16  3  3  1   2.333333
13   55       <NA> NA NA  1         NA
14   55 2011-07-07  6 16  7   9.666667
15  666 2012-10-22  2 NA  9   5.500000
16  666 2011-02-14 NA  9  5   7.000000
17  666 2005-05-24 NA  3 11   7.000000

>
>
```

8 Calculate Maximum Date Using tapply()

8.1 Simple max()

```
>   maxdate = tapply(data$date, data$id, max)
>   maxdate

    1    11    22    55   333   666  4444
 9382    NA 10549    NA 13245 15635    NA

>
```

8.2 Convert back into date

```
>   as.Date(maxdate, origin="1970-01-01")

           1           11           22           55          333          666
"1995-09-09"           NA "1998-11-19"           NA "2006-04-07" "2012-10-22"
        4444
          NA

>
```

8.3 max() with argument na.rm

```
>   maxdate = tapply(data$date, data$id, max, na.rm=TRUE)
>   maxdate = as.Date(maxdate, origin="1970-01-01")
>   maxdate

           1           11           22           55          333          666
"1995-09-09"           NA "1998-11-19" "2011-07-07" "2006-04-07" "2012-10-22"
        4444
"1994-08-27"

>
```

8.4 Merging back into the dataframe

```
>   maxdate = tapply(data$date, data$id, max, na.rm=TRUE)
>   maxdate = as.Date(maxdate, origin="1970-01-01")
>   tmp = data.frame(id=as.numeric(names(maxdate)), maxdate = maxdate)
>   data1 = merge(data, tmp, by="id", all.x=TRUE)
>   data1

     id       date q1 q2 q3 totalscore    maxdate
1     1 1995-09-09  6  0  1   2.333333 1995-09-09
2    11       <NA>  7 24 10  13.666667       <NA>
3    22 1992-05-15 11 NA 23  17.000000 1998-11-19
4    22 1998-11-19 10 NA 10  10.000000 1998-11-19
5    55       <NA>  2  5  2   3.000000 2011-07-07
6    55 2000-04-13 NA NA 10         NA 2011-07-07
7    55 1968-04-16  3  3  1   2.333333 2011-07-07
8    55       <NA> NA NA  1         NA 2011-07-07
9    55 2011-07-07  6 16  7   9.666667 2011-07-07
10  333 2006-04-07 NA  9  1   5.000000 2006-04-07
11  333 2005-04-17  6 NA  7   6.500000 2006-04-07
12  333 2001-12-13  5 10 11   8.666667 2006-04-07
13  666 2012-10-22  2 NA  9   5.500000 2012-10-22
14  666 2011-02-14 NA  9  5   7.000000 2012-10-22
15  666 2005-05-24 NA  3 11   7.000000 2012-10-22
16 4444       <NA>  6  8  7   7.000000 1994-08-27
17 4444 1994-08-27 13  4 15  10.666667 1994-08-27
```

8.5 Alternative (better) way of merging

```
>   maxdate[c(1, 2, 3)]

           1           11           22
"1995-09-09"           NA "1998-11-19"

>   maxdate[c("1", "11", "22")]

           1           11           22
"1995-09-09"           NA "1998-11-19"

>   maxdate[as.character(data$id)]

         333          333          333           11         4444         4444
"2006-04-07" "2006-04-07" "2006-04-07"           NA "1994-08-27" "1994-08-27"
           1           22           22           55           55           55
"1995-09-09" "1998-11-19" "1998-11-19" "2011-07-07" "2011-07-07" "2011-07-07"
          55           55          666          666          666
"2011-07-07" "2011-07-07" "2012-10-22" "2012-10-22" "2012-10-22"

>   maxdate = tapply(data$date, data$id, max, na.rm=TRUE)
>   maxdate = as.Date(maxdate, origin="1970-01-01")
>   data$maxdate = maxdate[as.character(data$id)]
>   data

     id       date q1 q2 q3 totalscore    maxdate
1   333 2006-04-07 NA  9  1   5.000000 2006-04-07
2   333 2005-04-17  6 NA  7   6.500000 2006-04-07
```

```
3   333 2001-12-13  5 10 11   8.666667 2006-04-07
4    11       <NA>  7 24 10  13.666667       <NA>
5  4444       <NA>  6  8  7   7.000000 1994-08-27
6  4444 1994-08-27 13  4 15  10.666667 1994-08-27
7     1 1995-09-09  6  0  1   2.333333 1995-09-09
8    22 1992-05-15 11 NA 23  17.000000 1998-11-19
9    22 1998-11-19 10 NA 10  10.000000 1998-11-19
10   55       <NA>  2  5  2   3.000000 2011-07-07
11   55 2000-04-13 NA NA 10         NA 2011-07-07
12   55 1968-04-16  3  3  1   2.333333 2011-07-07
13   55       <NA> NA NA  1         NA 2011-07-07
14   55 2011-07-07  6 16  7   9.666667 2011-07-07
15  666 2012-10-22  2 NA  9   5.500000 2012-10-22
16  666 2011-02-14 NA  9  5   7.000000 2012-10-22
17  666 2005-05-24 NA  3 11   7.000000 2012-10-22
```