Data Manipulation

Contents

1 Data Manipulation

Report examplifying the use of dplyr in data handling on the example of dsL.

1.1 Five basic functions in data handling

For a more detailed discussion of basic verbs and operations consult the R-Studio guide or internal vignette

```
# vignette("introduction",package="dplyr")
```

The following is a brief demonstration of dplyr syntax using dsL example

1.1.1 select()

selects variables into a smaller data set

```
require(dplyr)
ds<-dsL
dim(ds)
```

```
[1] 134760     60
```

```
ds<- select(ds,id,year, byear, attend, attendF)
head(ds,13)
```

```
   id year byear attend         attendF
1   1 1997  1981     NA            <NA>
2   1 1998  1981     NA            <NA>
3   1 1999  1981     NA            <NA>
4   1 2000  1981      1           Never
```

```
5   1 2001  1981      6 About once/week
6   1 2002  1981      2   Once or Twice
7   1 2003  1981      1           Never
8   1 2004  1981      1           Never
9   1 2005  1981      1           Never
10  1 2006  1981      1           Never
11  1 2007  1981      1           Never
12  1 2008  1981      1           Never
13  1 2009  1981      1           Never
```

```
dim(ds)
```

```
[1] 134760      5
```

1.1.2 filter()

Removes observations that do not meet criteria. The following code selects observation based on the type of sample

```
  sample         sampleF
1      1 Cross-Sectional
2      0      Oversample
```

and only between years 2000 and 2011, as only during those years the outcome of interest attend was recorded.

```
require(dplyr)
ds<- filter(dsL,sample==1, year %in% c(2000:2011))
ds<- select(ds,id, year, attend, attendF)
head(ds,13)
```

```
   id year attend         attendF
1   1 2000      1           Never
2   1 2001      6 About once/week
3   1 2002      2   Once or Twice
4   1 2003      1           Never
5   1 2004      1           Never
6   1 2005      1           Never
7   1 2006      1           Never
8   1 2007      1           Never
9   1 2008      1           Never
10  1 2009      1           Never
11  1 2010      1           Never
12  1 2011      1           Never
13  2 2000      2   Once or Twice
```

1.1.3 arrange()

Sorts observations

```
require(dplyr)
ds<- filter(dsL,sample==1, year %in% c(2000:2011))
ds<- select(ds,id, year, attend)
ds<- arrange(ds, year, desc(id))
head(ds,13)
```

```
     id year attend
1  9022 2000      1
2  9021 2000      2
3  9020 2000      2
4  9018 2000      4
5  9017 2000      6
6  9012 2000      5
7  9011 2000      6
8  9010 2000      1
9  9009 2000      2
10 9008 2000      6
11 8992 2000     NA
12 8991 2000      3
13 8987 2000      6
```

```
ds<- arrange(ds, id, year)
head(ds, 13)
```

```
   id year attend
1   1 2000      1
2   1 2001      6
3   1 2002      2
4   1 2003      1
5   1 2004      1
6   1 2005      1
7   1 2006      1
8   1 2007      1
9   1 2008      1
10  1 2009      1
11  1 2010      1
12  1 2011      1
13  2 2000      2
```

1.1.4 mutate()

Creates additional variables from the values of existing.

```
require(dplyr)
ds<- filter(dsL,sample==1, year %in% c(2000:2011))
ds<- select(ds,id, byear, year, attend)
ds<- mutate(ds,
            age = year-byear,
            timec = year-2000,
            linear= timec,
            quadratic= linear^2,
            cubic= linear^3)
head(ds,13)
```

```
   id byear year attend age timec linear quadratic cubic
1   1  1981 2000      1  19     0      0         0     0
2   1  1981 2001      6  20     1      1         1     1
3   1  1981 2002      2  21     2      2         4     8
4   1  1981 2003      1  22     3      3         9    27
5   1  1981 2004      1  23     4      4        16    64
6   1  1981 2005      1  24     5      5        25   125
7   1  1981 2006      1  25     6      6        36   216
8   1  1981 2007      1  26     7      7        49   343
9   1  1981 2008      1  27     8      8        64   512
10  1  1981 2009      1  28     9      9        81   729
11  1  1981 2010      1  29    10     10       100  1000
12  1  1981 2011      1  30    11     11       121  1331
13  2  1982 2000      2  18     0      0         0     0
```

1.1.5 summarize()

1.2 Grouping and Combining

The function group_by is used to identify groups in split-apply-combine (SAC) procedure: all possible interactions between the levels of supplied variables

```
require(dplyr)

ds<- filter(dsL,sample==1, year %in% c(2000:2011))
ds<- select(ds,id, year, attendF)

s <- group_by(ds, year,attendF)
s <- summarise(s, count = n())
s <- mutate(s, total = sum(count),
              percent= count/total)
head(s,10)

Source: local data frame [10 x 5]
Groups: year

   year              attendF count total  percent
1  2000                Never  1581  6748 0.234292
2  2000        Once or Twice  1304  6748 0.193242
3  2000 Less than once/month   775  6748 0.114849
4  2000     About once/month   362  6748 0.053646
5  2000    About twice/month   393  6748 0.058239
6  2000      About once/week  1101  6748 0.163159
7  2000   Several times/week   463  6748 0.068613
8  2000              Everyday    36  6748 0.005335
9  2000                    NA   733  6748 0.108625
10 2001                Never  1627  6748 0.241108
```

The same result can be achieved with the same result use a more elegant syntax that relies on %>% operator, in which x %>% f(y) turns into f(x, y) :

```
ds<-dsL %>%
  filter(sample==1, year %in% c(2000:2011)) %>%
```

```
  select(id, year, attendF) %>%
  group_by(year,attendF) %>%
    summarise(count = n()) %>%
    mutate(total = sum(count),
               percent= count/total)
head(ds,10)
```

```
Source: local data frame [10 x 5]
Groups: year
```

```
   year              attendF count total  percent
1  2000                Never  1581  6748 0.234292
2  2000        Once or Twice  1304  6748 0.193242
3  2000 Less than once/month   775  6748 0.114849
4  2000     About once/month   362  6748 0.053646
5  2000    About twice/month   393  6748 0.058239
6  2000      About once/week  1101  6748 0.163159
7  2000   Several times/week   463  6748 0.068613
8  2000              Everyday    36  6748 0.005335
9  2000                    NA   733  6748 0.108625
10 2001                Never  1627  6748 0.241108
```

verify that this is what we wanted to achieve:

```
summarize( filter(s, year==2000), should.be.one=sum(percent))
```

```
Source: local data frame [1 x 2]
```

```
  year should.be.one
1 2000             1
```

1.3 Base subsetting

Generally we can select any desired dataset by formula
dataset[*condition for rows* , *condition for columns*], and using **dataset$variableName** selector

```
ds<-dsL[dsL$year %in% c(2000:2011),c('id',"byear","year","attendF","ageyearF","agemon")]
print(ds[ds$id==1,])
```

```
   id byear year         attendF ageyearF agemon
4   1  1981 2000           Never       19    231
5   1  1981 2001 About once/week       20    243
6   1  1981 2002   Once or Twice       21    256
7   1  1981 2003           Never       22    266
8   1  1981 2004           Never       23    279
9   1  1981 2005           Never       24    290
10  1  1981 2006           Never       25    302
11  1  1981 2007           Never       26    313
12  1  1981 2008           Never       27    325
13  1  1981 2009           Never       28    337
14  1  1981 2010           Never       29    350
15  1  1981 2011           Never       29    360
```