STA261 Manajemen Data Relasional
Teknik Wrangling Data Frame di R

Dr. Agus M Soleh

IPB University
Bogor Indonesia

https://www.stat.ipb.ac.id/agusms/
agusms@apps.ipb.ac.id

Outline

Pendahuluan

Munging/wrangling data di R diutamakan untuk pengelolaan data frame dengan cara:

- Menambahkan kolom/peubah baru dalam data frame
- Subsetting data
- Mengurutkan data
- Meng-kode ulang data
- Menggabungkan data
- Mengubah format data frame

Menambahkan kolom/peubah baru dalam data frame

- Kadang diperlukan suatu peubah yang merupakan operasi dari objek yang ada
- Dilakukan menggunakan operator assignment $ atau [] atau [[]]
  Misal telah terdapat data frame dtku, kemudian ingin menambahkan peubah `baru` maka gunakan `dtku$baru <- ekspresi`
- Hati-hati jika menggunakan operator [] atau [[]]
  jika indeksnya sudah terisi maka akan terupdate untuk kolom tersebut

Subsetting data

- Hal penting untuk subsetting data adalah membuat vektor logical seperti yg diinginkan.
- Harus dapat menterjemahkan idea rumit ke dalam vektor logic
- Misal terdapat data berikut:

```
> a
   gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10
1       f  8  9  5  9  9  9  5  9 11   8
2       f 15  8  9  5 10  8 10 12  9  15
3       f  8 13 12  6  7  9 14 12 12   9
4       m 14  9  8 NA 11 10  6 11 11   8
5       m  8  2  9 16  8 10  8  9  8   9
6       f 10  6 10  9 10 10  9  7  7  11
7       m  9  9  7 13  9 12 10  9 11   7
8       m  9 10  8 10  5 10  6 12  6   8
9       f 10 13  6  7 10 12 13  9  6   9
10      m 12 10  9  8 17  7  9 10  7   7
```

Subsetting data

- Fungsi dan operator yang bermanfaat untuk digunakan:

  `==, !=, >, >=, <, <=, %in%, duplicated, is.na, is.null, is.numeric, dll`

- Contoh: jika hanya diinginkan data khusus untuk gender *females*

```
> idx <- a$gender=="f"
> a[idx,]
  gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10
1      f  8  9  5  9  9  9  5  9 11   8
2      f 15  8  9  5 10  8 10 12  9  15
3      f  8 13 12  6  7  9 14 12 12   9
6      f 10  6 10  9 10 10  9  7  7  11
9      f 10 13  6  7 10 12 13  9  6   9
```

Subsetting data

- Misalkan diinginkan dataset baru yang memenuhi syarat:
  - untuk *females*: v1 di atas 7 dan v10 di bawah 10
  - untuk *males*: pada v4 selain *missing value*

```
> myidx <- (a$gender=="f" & a$v1>7 & a$v10<10) |
           (a$gender=="m" & !is.na(a$v4))
> a.new <- a[myidx,]
> a.new
   gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10
1       f  8  9  5  9  9  9  5  9 11   8
3       f  8 13 12  6  7  9 14 12 12   9
5       m  8  2  9 16  8 10  8  9  8   9
7       m  9  9  7 13  9 12 10  9 11   7
8       m  9 10  8 10  5 10  6 12  6   8
9       f 10 13  6  7 10 12 13  9  6   9
10      m 12 10  9  8 17  7  9 10  7   7
```

Mengurutkan data

- Membuat index seperti dalam subsetting, tetapi indeks yang dibuat adalah vektor integer
- Beberapa fungsi yang bermanfaat:

```
order()
sort()
which()
rev()
unique()
dll
```

Mengurutkan data

- Contoh: Mengurutkan berdasarkan gender secara ascending

```
> idx <- order(a$gender)
> a[idx,]
   gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10
1       f  8  9  5  9  9  9  5  9 11   8
2       f 15  8  9  5 10  8 10 12  9  15
3       f  8 13 12  6  7  9 14 12 12   9
6       f 10  6 10  9 10 10  9  7  7  11
9       f 10 13  6  7 10 12 13  9  6   9
4       m 14  9  8 NA 11 10  6 11 11   8
5       m  8  2  9 16  8 10  8  9  8   9
7       m  9  9  7 13  9 12 10  9 11   7
8       m  9 10  8 10  5 10  6 12  6   8
10      m 12 10  9  8 17  7  9 10  7   7
```

Mengurutkan data

Ingin diurutkan dengan urutan:

- gender (ascending)
- v1 jika male dan v2 jika female (ascending)

```
> newvec <- (a$gender=="m")*a$v1 + (a$gender=="f")*a$v2
> idx <- order(a$gender, newvec)
> a[idx,]
   gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10
6       f 10  6 10  9 10 10  9  7  7  11
2       f 15  8  9  5 10  8 10 12  9  15
1       f  8  9  5  9  9  9  5  9 11   8
3       f  8 13 12  6  7  9 14 12 12   9
9       f 10 13  6  7 10 12 13  9  6   9
5       m  8  2  9 16  8 10  8  9  8   9
7       m  9  9  7 13  9 12 10  9 11   7
8       m  9 10  8 10  5 10  6 12  6   8
10      m 12 10  9  8 17  7  9 10  7   7
4       m 14  9  8 NA 11 10  6 11 11   8
```

Meng-kode ulang data

- Pada kasus ini diinginkan variabel yang merupakan hasil transformasi dari variabel yang sudah ada
- Penambahan variabel dilakukan seperti pada proses menciptakan variabel baru
- Umumnya menggunakan pernyataan dan fungsi `if()`, `ifelse()`, `switch()` dan ekspresi logical
- Contoh:
  - Menggunakan logical:
    ```
    dta$grupumur<-1*(dta$AGE<=30)+2*(dta$AGE>30)
    ```
  - Menggunakan fungsi ifelse:
    ```
    dta$grupumur<-ifelse(dta$AGE %in% 1:30,1,2)
    ```

Menggabungkan data

- Dapat menggunakan fungsi `cbind()` atau `rbind()`
- Jika melakukan proses join, dapat menggunakan fungsi `merge()`
- Misal terdapat data frame sebagai berikut:

```
> a3
   id1 gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10
1  A.0      f 10  6 10  9 10 10  9  7  7  11
2  B.1      f 15  8  9  5 10  8 10 12  9  15
3  C.0      f  8  9  5  9  9  9  5  9 11   8
4  D.1      f  8 13 12  6  7  9 14 12 12   9
5  E.1      f 10 13  6  7 10 12 13  9  6   9
6  F.0      m  8  2  9 16  8 10  8  9  8   9
7  G.0      m  9  9  7 13  9 12 10  9 11   7
8  H.0      m  9 10  8 10  5 10  6 12  6   8
9  I.0      m 12 10  9  8 17  7  9 10  7   7
10 J.0      m 14  9  8 NA 11 10  6 11 11   8
```

```
> a4
  X1 X2 id2
1  3  2 A.0
2  4  4 B.1
3  1  4 C.1
4  3  3 D.0
5  2  1 E.0
```

Menggabungkan data

```
> merge(a3,a4,by.x=1,by.y=3,all=FALSE)
  id1 gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 X1 X2
1 A.0      f 10  6 10  9 10 10  9  7  7  11  3  2
2 B.1      f 15  8  9  5 10  8 10 12  9  15  4  4
> merge(a3,a4,by.x=1,by.y=3,all=TRUE)
   id1 gender v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 X1 X2
1  A.0      f 10  6 10  9 10 10  9  7  7  11  3  2
2  B.1      f 15  8  9  5 10  8 10 12  9  15  4  4
3  C.0      f  8  9  5  9  9  9  5  9 11   8 NA NA
4  D.1      f  8 13 12  6  7  9 14 12 12   9 NA NA
5  E.1      f 10 13  6  7 10 12 13  9  6   9 NA NA
6  F.0      m  8  2  9 16  8 10  8  9  8   9 NA NA
7  G.0      m  9  9  7 13  9 12 10  9 11   7 NA NA
8  H.0      m  9 10  8 10  5 10  6 12  6   8 NA NA
9  I.0      m 12 10  9  8 17  7  9 10  7   7 NA NA
10 J.0      m 14  9  8 NA 11 10  6 11 11   8 NA NA
11 C.1   <NA> NA NA NA NA NA NA NA NA NA  NA  1  4
12 D.0   <NA> NA NA NA NA NA NA NA NA NA  NA  3  3
```

Mengubah format data frame

- Mengubah format data frame adalah proses mengubah format data frame:
  - *Long to wide*
  - *Wide to long*
- Menggunakan fungsi `reshape()`
- Misal terdapat data berikut:

```
> df1
   school class ggg       score t2
1       1     9   1  0.37934875  1
2       1    10   1  0.05293721  2
3       1     9   2  1.38667892  3
4       1    10   2  0.20765320  4
5       2     9   1  0.65889853  1
6       2    10   1 -0.15820791  2
7       2     9   2 -0.86448228  3
8       2    10   2 -1.09312537  4
9       3     9   1  0.14485712  1
10      3    10   1  0.03916625  2
11      3     9   2  0.31687016  3
12      3    10   2  0.10621393  4
```

Long to wide data format

- Dua argumen yang harus digunakan:
  - idvar = minimal 1 variabel yang barisnya menunjukkan individu yang sama
  - timevar = 1 variabel untuk dijadikan variabel dalam format wide
- ilustrasi:

```
> wd <- reshape(df1, idvar=c("school","class"), timevar=
      "ggg", direction="wide")
> wd
   school class     score.1 t2.1    score.2 t2.2
1       1     9  0.37934875    1  1.3866789    3
2       1    10  0.05293721    2  0.2076532    4
5       2     9  0.65889853    1 -0.8644823    3
6       2    10 -0.15820791    2 -1.0931254    4
9       3     9  0.14485712    1  0.3168702    3
10      3    10  0.03916625    2  0.1062139    4
```

Long to wide data format

- Ilustrasi lain:

```
> reshape(df1, idvar="school", timevar="t2", direction=
    "wide", drop="ggg")
  school class.1   score.1 class.2     score.2
1      1       9 0.3793487      10  0.05293721
5      2       9 0.6588985      10 -0.15820791
9      3       9 0.1448571      10  0.03916625
  class.3    score.3 class.4    score.4
1       9  1.3866789      10  0.2076532
5       9 -0.8644823      10 -1.0931254
9       9  0.3168702      10  0.1062139
```

Wide to Long data format

- Dua argumen yang harus digunakan:
  - varying = objek list dari nama variabel dalam format wide
  - v.names = objek vektor dari nama kolom baru dalam format long

```
> reshape(wd, varying=list(c("score.1","score.2"),c("t2.1","t2.2")),
    direction="long", v.names=c("new1", "new2"))
    school class time        new1 new2 id
1.1      1     9    1  0.37934875    1  1
2.1      1    10    1  0.05293721    2  2
3.1      2     9    1  0.65889853    1  3
4.1      2    10    1 -0.15820791    2  4
5.1      3     9    1  0.14485712    1  5
6.1      3    10    1  0.03916625    2  6
1.2      1     9    2  1.38667892    3  1
2.2      1    10    2  0.20765320    4  2
3.2      2     9    2 -0.86448228    3  3
4.2      2    10    2 -1.09312537    4  4
5.2      3     9    2  0.31687016    3  5
6.2      3    10    2  0.10621393    4  6
```

Akhir Materi ...