Impor Data dan Konsep Tidy Data

Muhammad Aswan Syahputra

4/9/2019

Impor Data

Anda dapat menggunakan paket `readr` untuk mengimpor berkas lokal di komputer atau dari pranala (URL). Anda dapat mengaktifkan paket `readr` dengan cara menjalankan `library(nama_paket)` seperti contoh berikut: (Petunjuk: Tekan Ctrl + Enter untuk menjalankan baris kode.)

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.5.3
```

Paket hanya perlu dipasang satu kali melalui fungsi `install.packages("nama_paket")` dan harus selalu diaktifkan setiap menggawali kerja menggunakan R agar fungsi-fungsi yang tersedia dalam paket tersebut dapat digunakan. Sebagai contoh, kita akan menggunakan fungsi `read_csv()` dari paket `readr` untuk mengimpor data 'evals.csv' dari folder 'data-raw' sebagai berikut:

```
evals <- read_csv("../data-raw/evals.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   rank = col_character(),
##   ethnicity = col_character(),
##   gender = col_character(),
##   language = col_character(),
##   cls_level = col_character(),
##   cls_profs = col_character(),
##   cls_credits = col_character(),
##   pic_outfit = col_character(),
##   pic_color = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
evals
```

```
## # A tibble: 463 x 21
##    score rank  ethnicity gender language   age cls_perc_eval cls_did_eval
##    <dbl> <chr> <chr>     <chr>  <chr>    <dbl>         <dbl>        <dbl>
##  1   4.7 tenu~ minority  female english     36          55.8           24
##  2   4.1 tenu~ minority  female english     36          68.8           86
##  3   3.9 tenu~ minority  female english     36          60.8           76
##  4   4.8 tenu~ minority  female english     36          62.6           77
##  5   4.6 tenu~ not mino~ male   english     59          85             17
##  6   4.3 tenu~ not mino~ male   english     59          87.5           35
##  7   2.8 tenu~ not mino~ male   english     59          88.6           39
```

```
##  8   4.1 tenu~ not mino~ male   english     51         100             55
##  9   3.4 tenu~ not mino~ male   english     51          56.9          111
## 10   4.5 tenu~ not mino~ female english     40          87.0           40
## # ... with 453 more rows, and 13 more variables: cls_students <dbl>,
## #   cls_level <chr>, cls_profs <chr>, cls_credits <chr>,
## #   bty_f1lower <dbl>, bty_f1upper <dbl>, bty_f2upper <dbl>,
## #   bty_m1lower <dbl>, bty_m1upper <dbl>, bty_m2upper <dbl>,
## #   bty_avg <dbl>, pic_outfit <chr>, pic_color <chr>
```

Anda dapat menggunakan fungsi `glimpse` dari paket `dplyr` untuk melihat ringkasan data. tersebut. Isilah '____' dengan jawaban yang tepat untuk melihat ringkasan data `evals`! Ada berapa variable dan observasi pada data `evals` tersebut?

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
glimpse(evals)
```

```
## Observations: 463
## Variables: 21
## $ score         <dbl> 4.7, 4.1, 3.9, 4.8, 4.6, 4.3, 2.8, 4.1, 3.4, 4.5...
## $ rank          <chr> "tenure track", "tenure track", "tenure track", ...
## $ ethnicity     <chr> "minority", "minority", "minority", "minority", ...
## $ gender        <chr> "female", "female", "female", "female", "male", ...
## $ language      <chr> "english", "english", "english", "english", "eng...
## $ age           <dbl> 36, 36, 36, 36, 59, 59, 59, 51, 51, 40, 40, 40, ...
## $ cls_perc_eval <dbl> 55.81395, 68.80000, 60.80000, 62.60163, 85.00000...
## $ cls_did_eval  <dbl> 24, 86, 76, 77, 17, 35, 39, 55, 111, 40, 24, 24,...
## $ cls_students  <dbl> 43, 125, 125, 123, 20, 40, 44, 55, 195, 46, 27, ...
## $ cls_level     <chr> "upper", "upper", "upper", "upper", "upper", "up...
## $ cls_profs     <chr> "single", "single", "single", "single", "multipl...
## $ cls_credits   <chr> "multi credit", "multi credit", "multi credit", ...
## $ bty_f1lower   <dbl> 5, 5, 5, 5, 4, 4, 4, 5, 5, 2, 2, 2, 2, 2, 2, 2, ...
## $ bty_f1upper   <dbl> 7, 7, 7, 7, 4, 4, 4, 2, 2, 5, 5, 5, 5, 5, 5, 5, ...
## $ bty_f2upper   <dbl> 6, 6, 6, 6, 2, 2, 2, 5, 5, 4, 4, 4, 4, 4, 4, 4, ...
## $ bty_m1lower   <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, ...
## $ bty_m1upper   <dbl> 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ...
## $ bty_m2upper   <dbl> 6, 6, 6, 6, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, ...
## $ bty_avg       <dbl> 5.000, 5.000, 5.000, 5.000, 3.000, 3.000, 3.000,...
## $ pic_outfit    <chr> "not formal", "not formal", "not formal", "not f...
## $ pic_color     <chr> "color", "color", "color", "color", "color", "co...
```

Selain itu Anda juga dapat menggunakan fungsi `skim()` dari paket `skimr` untuk melihat rangkuman data. Pada *chunck* berikut, tuliskan kode untuk mengaktifkan paket `skimr` dan menjalankan fungsi `skim` pada data `evals`! Perbedaaan apakah yang Anda temukan antara penggunaan fungsi `glimpse()` dan `skim()`?

```
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 3.5.3
```

```
##
## Attaching package: 'skimr'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
skim(evals)
```

```
## Skim summary statistics
##  n obs: 463
##  n variables: 21
##
## -- Variable type:character -----------------------------------------------------------------------
##     variable missing complete   n min max empty n_unique
##  cls_credits       0      463 463  10  12     0        2
##    cls_level       0      463 463   5   5     0        2
##    cls_profs       0      463 463   6   8     0        2
##    ethnicity       0      463 463   8  12     0        2
##       gender       0      463 463   4   6     0        2
##     language       0      463 463   7  11     0        2
##    pic_color       0      463 463   5  11     0        2
##   pic_outfit       0      463 463   6  10     0        2
##         rank       0      463 463   7  12     0        3
##
## -- Variable type:numeric -------------------------------------------------------------------------
##       variable missing complete   n  mean    sd    p0   p25   p50   p75
##            age       0      463 463 48.37  9.8  29    42    48    57
##        bty_avg       0      463 463  4.42  1.53  1.67  3.17  4.33  5.5
##    bty_f1lower       0      463 463  3.96  1.87  1     2     4     5
##    bty_f1upper       0      463 463  5.02  1.93  1     4     5     7
##    bty_f2upper       0      463 463  5.21  2.02  1     4     5     6
##    bty_m1lower       0      463 463  3.41  1.64  1     2     3     5
##    bty_m1upper       0      463 463  4.15  2.11  1     3     4     5
##    bty_m2upper       0      463 463  4.75  1.58  1     4     5     6
##   cls_did_eval       0      463 463 36.62 45.02  5    15    23    40
##  cls_perc_eval       0      463 463 74.43 16.76 10.42 62.7  76.92 87.25
##   cls_students       0      463 463 55.18 75.07  8    19    29    60
##          score       0      463 463  4.17  0.54  2.3   3.8   4.3   4.6
##    p100     hist
##   73    <U+2585><U+2585><U+2585><U+2587><U+2585><U+2587><U+2582><U+2581>
##    8.17 <U+2582><U+2585><U+2585><U+2587><U+2583><U+2583><U+2582><U+2581>
##    8    <U+2583><U+2587><U+2586><U+2587><U+2586><U+2585><U+2582><U+2582>
##    9    <U+2583><U+2586><U+2587><U+2585><U+2586><U+2586><U+2583><U+2581>
```

```
##   10    <U+2583><U+2583><U+2586><U+2587><U+2587><U+2582><U+2582><U+2583>
##    7    <U+2582><U+2587><U+2585><U+2585><U+2581><U+2583><U+2582><U+2581>
##    9    <U+2587><U+2587><U+2587><U+2585><U+2583><U+2582><U+2582><U+2581>
##    9    <U+2582><U+2583><U+2587><U+2586><U+2585><U+2582><U+2581><U+2581>
##  380    <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
##  100    <U+2581><U+2581><U+2581><U+2582><U+2585><U+2586><U+2587><U+2586>
##  581    <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
##    5    <U+2581><U+2581><U+2582><U+2583><U+2585><U+2587><U+2587><U+2586>
```

Umumnya berkas `csv` menggunakan penanda koma (,) untuk memisahkan antar kolom dan titik (.) sebagai penanda desimal. Namun bagaimana jika Anda memiliki berkas `csv` yang yang menggunakan titik-koma (;) untuk memisahkan kolom dan koma (,) sebagai penanda desimal? Sebagai contoh, pada direktori `data-raw` terdapar berkas `evals2` yang memiliki kriteria tersebut. Anda dapat menggunakan fungsi `read_csv2` untuk mengimpor berkas tersebut sebagaimana ditunjukan pada contoh berikut:

```
evals2 <- read_csv2("../data-raw/evals2.csv")
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   rank = col_character(),
##   ethnicity = col_character(),
##   gender = col_character(),
##   language = col_character(),
##   cls_level = col_character(),
##   cls_profs = col_character(),
##   cls_credits = col_character(),
##   pic_outfit = col_character(),
##   pic_color = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
identical(evals, evals2) # fungsi untuk cek kesamaan antara dua obyek
```

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

Selain berkas lokal yang tersedia di komputer, Anda juga dapat mengimpor berkas yang tersedia di internet langsung dengan menggunakan pranala (URL). Caranya adalah dengan mengganti lokasi berkas lokal dengan lokasi berkas *remote*. Data `evals` tersedia pada pranala "https://www.openintro.org/stat/data/evals.csv". Dapatkan Anda mengimpor berkas tersebut dan menyimpannya sebagai obyek dengan nama `evals3`? Gunakkan fungsi `identical()` untuk membandingkannya dengan `evals2`!

```
evals3 <- read_csv("https://www.openintro.org/stat/data/evals.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   rank = col_character(),
```

```
##   ethnicity = col_character(),
##   gender = col_character(),
##   language = col_character(),
##   cls_level = col_character(),
##   cls_profs = col_character(),
##   cls_credits = col_character(),
##   pic_outfit = col_character(),
##   pic_color = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
identical(evals3, evals2)
```

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

Tidy data

Demi memahami konsep Tidy Data, kita akan menggunakan dataset yang tersedia di paket `tidyr`. Aktifkanlah paket `tidyr` tersebut!

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

Dari dataset berikut ini, manakah yang termasuk Tidy Data? Dapatkah Anda menjelaskan alasan mengapa beberapa dataset berikut tidak *tidy*?

```
table1
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

```
table2
```

```
## # A tibble: 12 x 4
##    country      year type            count
##    <chr>       <int> <chr>           <int>
##  1 Afghanistan  1999 cases             745
##  2 Afghanistan  1999 population   19987071
##  3 Afghanistan  2000 cases            2666
##  4 Afghanistan  2000 population   20595360
##  5 Brazil       1999 cases           37737
```

```
##  6 Brazil       1999 population  172006362
##  7 Brazil       2000 cases           80488
##  8 Brazil       2000 population  174504898
##  9 China        1999 cases          212258
## 10 China        1999 population 1272915272
## 11 China        2000 cases          213766
## 12 China        2000 population 1280428583
```

```
table3
```

```
## # A tibble: 6 x 3
##   country      year rate
## * <chr>       <int> <chr>
## 1 Afghanistan  1999 745/19987071
## 2 Afghanistan  2000 2666/20595360
## 3 Brazil       1999 37737/172006362
## 4 Brazil       2000 80488/174504898
## 5 China        1999 212258/1272915272
## 6 China        2000 213766/1280428583
```

```
table4a
```

```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

```
table4b
```

```
## # A tibble: 3 x 3
##   country         `1999`     `2000`
## * <chr>            <int>      <int>
## 1 Afghanistan   19987071   20595360
## 2 Brazil       172006362  174504898
## 3 China       1272915272 1280428583
```

```
table5
```

```
## # A tibble: 6 x 4
##   country     century year  rate
## * <chr>       <chr>   <chr> <chr>
## 1 Afghanistan 19      99    745/19987071
## 2 Afghanistan 20      00    2666/20595360
## 3 Brazil      19      99    37737/172006362
## 4 Brazil      20      00    80488/174504898
## 5 China       19      99    212258/1272915272
## 6 China       20      00    213766/1280428583
```