Data Manipulation and Basic R

Jinwon Lee

2020 7 17

This material would be helpful to understand basic r: https://r4ds.had.co.nz

Packages

Let's learn packages what usually used for data manipulation and wrangling. Here they are:

```
library(dplyr)
```

```
##
## 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
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.2.1     ✓ purrr   0.3.3
## ✓ tibble  2.1.3     ✓ stringr 1.4.0
## ✓ tidyr   1.0.2     ✓ forcats 0.4.0
## ✓ readr   1.3.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(haven)
library(readxl)
library(broom)
```

In turn:

tidyverse includes: 1) ggplot2, which implements the Grammar of Graphics.

2. dplyr: One important contribution of the dplyr package is that it provides a "grammar" (in particular, verbs) for data manipulation and for operating on data frames.

3. tidyr, which is a nice set of helper functions with an eye toward "tidy data:" a state of affairs where (1) every row in a data object is an observation; (2) every column in that data object is a variable; and (3) every cell in the data object is a single value.

These are major verbs of the tidyr :

• select: return a subset of the columns of a data frame, using a flexible notation • filter: extract a subset of rows from a data frame based on logical conditions • arrange: reorder rows of a data frame • rename: rename variables in a data frame • mutate: add new variables/columns or transform existing variables • summarise / summarize: generate summary statistics of different variables in the data frame, possibly within strata • %>%: the "pipe" operator is used to connect multiple verb actions together into a pipeline

4. readr is a good set of functions for reading in your data.

haven and *readxl* are both part of the tidyverse, too, but they are not automatically loaded. These help with reading in Stata, SAS, SPSS, and Excel data.

broom is a tidyverse-compliant way to handle the output of statistical models.

```
print(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
```

```
data.frame(table1)
```

country <chr>	year <int>	cases <int>	population <int>
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

6 rows

```
table1
```

country <chr>	year <int>	cases <int>	population <int>
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

6 rows

```
# summarize(table1)

# learn functions : 1) filter

# see data collected in the year of 1999
filter(table1, year==1999)
```

country <chr>	year <int>	cases <int>	population <int>
Afghanistan	1999	745	19987071
Brazil	1999	37737	172006362
China	1999	212258	1272915272

3 rows

```
# see only 'China' data
filter(table1, country=='China')
```

country <chr>	year <int>	cases <int>	population <int>
China	1999	212258	1272915272
China	2000	213766	1280428583

2 rows

```
#select? : see only country and year column

select(table1, country, year)
```

country <chr>	year <int>
Afghanistan	1999
Afghanistan	2000
Brazil	1999
Brazil	2000
China	1999
China	2000

6 rows

```
# change the name of column: rename

#e.g: rename(table1, numcases = cases)

#mutate

new_table1 <-mutate(table1, case_ratio = population/cases)

new_table1
```

country <chr>	year <int>	cases <int>	population <int>	case_ratio <dbl>
Afghanistan	1999	745	19987071	26828.283
Afghanistan	2000	2666	20595360	7725.191
Brazil	1999	37737	172006362	4558.030
Brazil	2000	80488	174504898	2168.086
China	1999	212258	1272915272	5997.019
China	2000	213766	1280428583	5989.861

6 rows

Download any dataset what you're interested in, and read that file by using different pakages (such as read.csv or readxl, etc).Then, try to manipulate data with what you've learned here.