第7回 大規模データを用いたデータフレーム操作実習(1)

平成 29 年 10 月 10 日

目 次

1 ここで学ぶ事

- これまで学んできたことを使いながら大規模データを用いて様々な分析を行う.

2 データの用意

第 5 回で作成した世界銀行の GDP データを読み込む.

```
> load("WorldBank_GDP.RData")   # E ドライブないなら"E:/WorldBank_GDP.RData"
> ls()                          # ppp オブジェクトが読み込まれた
[1] "ppp"
```

3 データの構造をチェック

大規模なデータを手にした場合，まずデータの構造を把握する必要がある．データの構造（structure）は `str()` 関数で調べる.

```
> str(ppp)
'data.frame':        264 obs. of  59 variables:
 $ Country.Code: Factor w/ 264 levels "ABW","AFG","AGO",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ X1960       : logi  NA NA NA NA NA NA ...
 $ X1961       : logi  NA NA NA NA NA NA ...
 ... 省略
 $ X1988       : logi  NA NA NA NA NA NA ...
 $ X1989       : logi  NA NA NA NA NA NA ...
 $ X1990       : num  NA NA 3107 2722 NA ...
 $ X1991       : num  NA NA 3076 1992 NA ...
 ... 省略
 $ X2013       : num  NA 1942 7097 10579 NA ...
 $ X2014       : num  NA 1942 7327 11308 NA ...
 $ X2015       : num  NA 1925 7387 11479 NA ...
 $ X2016       : logi  NA NA NA NA NA NA ...
 $ X           : logi  NA NA NA NA NA NA ...
```

ppp は非常に大きなデータフレームなので上の出力結果は一部省略している．結果出力の最初の行には，ppp が 264 個の observations（観測値）を持ち，59 個の変数から成るデータフレームであることがわかる．つまり，264 行 59 列のデータフレームだ．

その下のドル記号で始まる行は各列名を表す．2 行目の$ Country.Code: Factor w/ 264 levels は，Country.Code 列のデータ型が Factor 型（まだ習っていない）で，264 個のレベルがあることを示している．その後の"ABW", "AFG", "AGO"は最初の値を幾つか表示している．

3 行目以降には 1960 年から 2016 年までのデータがあることがわかる．1960 年から 1989 年のデータ型は logi となっているが，これは数値データが一つもなかったために全てのデータが NA となり論理型（Logic 型）のデータとして記録されていると考えられる．

X1990 から X2015 の列のデータ型は num となっており，Numeric 型のデータが格納されていることがわかる．すなわち，実際に PPP 評価による一人当たり GDP のデータがあるのは 1990 年から年から 2015 年までである．

3.1 練習問題

第 5 回で，独立行政法人統計センターから入手した大規模擬似ミクロデータを読み込み構造を確認せよ．データ数，変数の数，各変数のデータ型を述べよ．

4 ランキングの作成

世界銀行の GDP データを利用して，与えられたベクトル・データのランキングを作成し行名（国名）と共に表示する方法を学ぶ．利用可能な最新データである 2015 年のデータを用いてランキングを表示させる方法を学ぶ．

ベクトル・データの「値」を並べ替えるには sort() 関数を使う．decreasing=TRUE を指定し，GDP トップ 10 を表示させる．

```
> sort(ppp$X2015, decreasing=TRUE)[1:10]
 [1] 141542.66 111496.63 102051.68  85382.30  78369.29  74645.48  69970.82  68513.97
 [9]  62557.49  62083.86
```

確かに降順に値が並んでいる．しかし，我々は GDP の値だけでなく国名（行名）も合わせて表示させたい．つまり必要なのは GDP を降順に並べたときのデータのインデックス番号が知りたいのだ．order() 関数は与えられたデータで並べ替えたインデックス番号を返す．

```
> order(ppp$X2015, decreasing=TRUE)[1:10]
 [1] 199 145 143 207  30 126   7 110  36 176
```

すなわち，GDP トップのデータは 199 行目で，第 2 位は 145 行目のデータである．これを用いてデータフレームのトップ 10 の行を表示させてみよう．

```
> ppp[order(ppp$X2015, decreasing=TRUE)[1:10], "X2015"]
 [1] 141542.66 111496.63 102051.68  85382.30  78369.29  74645.48  69970.82  68513.97
 [9]  62557.49  62083.86
```

データフレームから 1 列のみ抽出するとベクトルが返るので，上の結果は先ほどの sort() の結果と同様，GDP データ値のみが表示される．このベクトルに対応する行名を与える．

```
> index <- order(ppp$X2015, decreasing=TRUE)[1:10]
> top10 <- ppp[index, "X2015"]
> names(top10) <- rownames(ppp)[index]
> top10
              Qatar     Macao SAR, China           Luxembourg            Singapore
          141542.66            111496.63            102051.68             85382.30
   Brunei Darussalam               Kuwait United Arab Emirates              Ireland
           78369.29             74645.48             69970.82             68513.97
         Switzerland              Norway
           62557.49             62083.86
```

top10 はベクトルなので横にデータが表示され，その上に要素名として国名が表示されている．縦に表示した方が見やすいので，このベクトルを 1 列とするデータフレームを作成する．

```
> data.frame(GDP.2015=top10)
                     GDP.2015
Qatar                141542.66
Macao SAR, China     111496.63
Luxembourg           102051.68
Singapore             85382.30
Brunei Darussalam     78369.29
Kuwait                74645.48
United Arab Emirates  69970.82
Ireland               68513.97
```

```
Switzerland           62557.49
Norway                62083.86
```

4.1 ランキングを生成する関数の作成

一人当たり GDP のトップ 10 を表示させるのに数行の処理を要した．これらを一つの関数にまとめてみよう．

```
> gdp.top10 <- function(data) { # data はランキングしたい年の GDP ベクトルデータ
+     index <- order(data, decreasing=TRUE)[1:10]
+     top10 <- data[index]
+     names(top10) <- rownames(ppp)[index]
+     data.frame(GDP=top10)
+ }
```

引数の data には ppp$X2015 の様にランキング付けする年のベクトル・データを指定する．order() 関数で降順に並べた要素のインデックス番号を index に保存しておく．ベクトル・データにはデータフレームの行名が付かないので，rownames(ppp)[index] でトップ 10 の行名（国名）を抽出し，ベクトルの要素に名前を付けている．最後にベクトルデータからデータフレームを作成し返す．

gdp.top10() 関数を実行してみる．

```
> gdp.top10(ppp$X2015)
                          GDP
Qatar                141542.66
Macao SAR, China     111496.63
Luxembourg           102051.68
Singapore             85382.30
Brunei Darussalam     78369.29
Kuwait                74645.48
United Arab Emirates  69970.82
Ireland               68513.97
Switzerland           62557.49
Norway                62083.86
```

唯一，関数を作る前との結果の違いは，列名が GDP.2015 になっていない点だ．gdp.top10() 関数には何年度のデータかという情報は渡されていない．これを修正するには 2 番目の引数で年度を明示的に指定するという方法が考えられる．

```
> gdp.top10 <- function(data, year) {
+     index <- order(data, decreasing=TRUE)[1:10]
+     top10 <- data[index]
+     names(top10) <- rownames(ppp)[index]
+     result <- data.frame(top10)                          # 列名は次の行で指定
+     colnames(result) <- paste("GDP.", year, sep="")   # 年度を加えた列名を構築
+     result
```

```
+ }
> gdp.top10(ppp$X2015, 2015)
                      GDP.2015
Qatar                141542.66
Macao SAR, China     111496.63
Luxembourg           102051.68
Singapore             85382.30
Brunei Darussalam     78369.29
Kuwait                74645.48
United Arab Emirates  69970.82
Ireland               68513.97
Switzerland           62557.49
Norway                62083.86
```

上の解決法には 2 つ問題点がある．一つ目の問題点は引数が冗長であることだ．第 1 引数と第 2 引数とで「2015」を 2 回タイプしなければならない．第 2 引数をわざわざ指定しなくても，第 1 引数でタイプした情報を利用して自動的に列名が設定できる方がスマートだ．

二つの問題点はタイプミスで第 1 引数と第 2 引数に違う年度を指定する可能性がある点だ．例えば，gdp.top10(ppp$2015, 2016) とタイプすると 2015 年度のデータのランキングなのに列名は GDP.2016 になってしまう．

この 2 点を回避するために，関数が呼び出された時に第 1 引数に指定した文字列情報を使って改良してみよう．引数の文字列情報を得るには deparse(substitute()) を使う．

```
> gdp.top10 <- function(data) { # data はランキングしたい年の GDP ベクトルデータ
+     year <- deparse(substitute(data))      # data に渡された引数名を文字列で取得
+     year <- substring(year, nchar(year)-3) # 後ろから 4 文字を取得
+     index <- order(data, decreasing=TRUE)[1:10]
+     top10 <- data[index]
+     names(top10) <- rownames(ppp)[index]
+     result <- data.frame(top10)                          # 列名は次の行で指定
+     colnames(result) <- paste("GDP.", year, sep="")   # 年度を加えた列名を構築
+     result
+ }
> gdp.top10(ppp$X2015)
                      GDP.2015
Qatar                141542.66
Macao SAR, China     111496.63
Luxembourg           102051.68
Singapore             85382.30
Brunei Darussalam     78369.29
Kuwait                74645.48
United Arab Emirates  69970.82
Ireland               68513.97
Switzerland           62557.49
Norway                62083.86
```

substring(str, first, last) 関数は str 文字列の first 番目から last 番目までの部分文字列を返す関数だ．last はデフォルト値が 1000 なので 1000 文字より短い文字列なら指定しなければ

first 番目以降の文字列全てを返す．列名の最後 4 文字を抜き出すためには「文字列の長さ −3」を指定している．nchar() は文字列の長さを返す関数だ．

これで何年のデータでもランキングできるようになった．

```
> gdp.top10(ppp$X2010)
                     GDP.2010
Qatar                123592.99
Macao SAR, China      96619.84
Luxembourg            85285.41
Brunei Darussalam     77992.29
Kuwait                72204.38
Singapore             70561.08
Norway                57995.86
United Arab Emirates  56245.48
Bermuda               55254.21
Switzerland           52935.80
```

4.2 練習問題

1. ワースト 10 を表示する gdp.worst10() 関数を作成せよ．
2. トップ 10 だけでなく，指定した順位のランキングを表示する gdp.ranking() 関数を作成せよ．例えば，gdp.ranking(ppp$2015, 25:35) を実行すると 25 位から 35 位までが国名と順位の番号とともに以下のよう表示される．

```
> gdp.ranking(ppp$X2015, 25:35)
                          GDP.2015 ranking
Canada                    44261.84      25
Post-demographic dividend 43813.34      26
Finland                   42309.39      27
United Kingdom            41801.05      28
Euro area                 41106.94      29
France                    41016.65      30
Japan                     40763.40      31
Equatorial Guinea         40718.83      32
OECD members              40589.12      33
Oman                      39971.09      34
European Union            38703.54      35
```

3. 国を指定すると GDP とランキングが以下のように表示される show.gdp() 関数を作成せよ．

```
> show.gdp(ppp$X2015, c("United States", "Germany", "Japan", "Korea, Rep.", "China"))
             GDP.2015 ranking
```

```
United States 56115.72      12
Germany       48041.70      18
Japan         40763.40      31
Korea, Rep.   34647.07      40
China         14450.72     101
```

4.3 処理の委譲

上の練習問題の 2 で任意のランキングを返す gdp.ranking() 関数を作成した．これを使えばトップ 10 やワースト 10 を返す関数が容易に作成できる．

```
> gdp.top10 <- function(data) {
+     gdp.ranking(data, 1:10)        # gdp.ranking に処理を委譲する
+ }
> gdp.worst10 <- function(data) {
+     gdp.ranking(data, length(data):(length(data)-10))
+ }
```

それぞれ実行してみると，ワースト 10 はうまくいかない．降順の最後に NA が配置されているためだ．order() 関数のオプションで na.last=NA と指定すると NA を取り除くけるが，この場合は gdp.worst10() の定義を以下のよう修正することで解決できる．

```
> gdp.worst10 <- function(data) {
+     len <- sum(!is.na(data))    # NA を除いたデータ数．sum は TRUE の和もとれる．
+     gdp.ranking(data, len:(len-10))
+ }
> gdp.worst10(ppp$X2015)
                         GDP.data ranking
Central African Republic  618.7529     227
Burundi                   727.1508     226
Congo, Dem. Rep.          784.3652     225
Liberia                   835.3669     224
Niger                     955.4833     223
Malawi                   1183.6052     222
Mozambique               1192.1753     221
Guinea                   1208.9861     220
Guinea-Bissau            1455.8036     219
Togo                     1460.3449     218
Madagascar               1465.3448     217
```

より一般的なケースの処理をする関数を作成し，特殊ケースの処理は一般的な関数に委譲する方が，トップ 10，ワースト 10，一般ケースと別々に定義するよりはるかに効率的だ．今回の場合，gdp.top10() の中身は 1 行に，gdp.worst10() の中身は 2 行にまとめることができた．