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)

2023 年 2 月 27 日

目 次

1 ここで学ぶ事 **1**

2 データの用意 **1**

3 データの構造をチェック **1**
3.1 練習問題 2

4 ランキングの作成 **2**
4.1 ランキングを生成する関数の作成 3
4.1.1 更なる修正 6
4.2 練習問題 7
4.3 処理の委譲 8

5 記述統計 **9**
5.1 練習問題 9

6 ヒストグラム **10**

1 ここで学ぶ事

- これまで学んできたことを使い，多少のプログラミングを混ぜながらデータ加工を行う.
- 記述統計やグラフを用いてデータの分布や特徴を観察する.

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  62 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  24101 NA 3090 2549 NA ...
 $ X1991       : num  25871 NA 3120 1909 NA ...
 ... 省略
 $ X2016       : num  38390 1897 6757 11868 NA ...
 $ X2017       : num  39455 1935 6651 12930 NA ...
 $ X2018       : num  NA 1955 6452 13364 NA ...
 $ X2019       : 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．出力結果の 1 行目には，ppp が 264 個の observations（観測値）を持ち，62 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 62 列のデータフレームだ．

その下のドル記号で始まる行は列名を表す．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 2019 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 `X2018` 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 2018 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

この節では世界銀行の GDP データを用いて，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）と共に表示させる方法を学ぶ．以下，利用可能な最新データである 2018 年のデータを用いる．

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$X2018, decreasing=TRUE)[1:10]
 [1] 126898.43 123892.17 113337.42 101531.63  83203.39  80920.05  75075.26  72897.56
 [9]  68060.94  65510.59
```

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$X2018, decreasing=TRUE)[1:10]
 [1] 199 145 143 207 110  30   7 126  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$X2018, decreasing=TRUE)[1:10], "X2018"]
 [1] 126898.43 123892.17 113337.42 101531.63  83203.39  80920.05  75075.26  72897.56
 [9]  68060.94  65510.59
```

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$X2018, decreasing=TRUE)[1:10]
> top10 <- ppp[index, "X2018"]
> names(top10) <- rownames(ppp)[index]
> top10
               Qatar     Macao SAR, China           Luxembourg            Singapore
           126898.43            123892.17            113337.42            101531.63
             Ireland    Brunei Darussalam United Arab Emirates               Kuwait
            83203.39             80920.05             75075.26             72897.56
         Switzerland               Norway
            68060.94             65510.59
```

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.2018=top10)
                     GDP.2018
Qatar                126898.43
Macao SAR, China     123892.17
Luxembourg           113337.42
Singapore            101531.63
Ireland               83203.39
Brunei Darussalam     80920.05
United Arab Emirates  75075.26
```

```
Kuwait                72897.56
Switzerland           68060.94
Norway                65510.59
```

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$X2018 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$X2018)
                          GDP
Qatar                126898.43
Macao SAR, China     123892.17
Luxembourg           113337.42
Singapore            101531.63
Ireland               83203.39
Brunei Darussalam     80920.05
United Arab Emirates  75075.26
Kuwait                72897.56
Switzerland           68060.94
Norway                65510.59
```

唯一，関数を作る前との結果の違いは，列名が GDP.2018 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$X2018, 2018)
                      GDP.2018
Qatar                 126898.43
Macao SAR, China      123892.17
Luxembourg            113337.42
Singapore             101531.63
Ireland                83203.39
Brunei Darussalam      80920.05
United Arab Emirates   75075.26
Kuwait                 72897.56
Switzerland            68060.94
Norway                 65510.59
```

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「2018」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．

二つ目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$2018，2016) とタイプすると 2018 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 点を回避するには，gdp.top10(ppp$2018) と 1 つの引数で呼び出し，引数に渡された文字列情報（すなわち ppp$2018）から 2018 を抽出すれば良い．引数の文字列情報を得るには deparse(substitute()) を使う．以下がその実装例である．引数の文字列情報を使った処理に関する解説はコメントに記してある．

```
> 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$X2018)
                      GDP.2018
Qatar                 126898.43
Macao SAR, China      123892.17
Luxembourg            113337.42
Singapore             101531.63
Ireland                83203.39
Brunei Darussalam      80920.05
United Arab Emirates   75075.26
Kuwait                 72897.56
Switzerland            68060.94
```

```
Norway                65510.59
```

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             117518.70
Macao SAR, China   95951.96
Luxembourg         85613.59
Brunei Darussalam  78907.87
Kuwait             73817.98
San Marino         72075.30
Singapore          71566.00
Cayman Islands     62158.60
Norway             57915.02
Bermuda            55241.07
```

4.1.1 更なる修正

上で作成した gdp.top10() 関数はうまくいっているように見える．しかし，まだ問題がある．もし，この関数を使うユーザが GDP データを ppp ではなく my.ppp という名前の変数に格納していたらどうなるだろうか．以下，ppp を my.ppp 変数に代入し ppp オブジェクトを削除してから gdp.top10() を実行してみる．

```
> my.ppp <- ppp   # ppp を my.ppp に名称変更
> rm(ppp)         # ppp オブジェクトを削除
> ls()            # ppp が存在しないことを確認
[1] "microdata" "my.ppp"
> gdp.top10(my.ppp$X2018) # my.ppp に対して gdp.top10() を実行
 rownames(ppp) でエラー:  オブジェクト 'ppp' がありません
```

最後のエラーは gdp.top10() 関数内で行名を取得する際，関数の外にある ppp オブジェクトを参照（アクセス）しているために生じる．GDP データは my.ppp に格納されているのだが，gdp.top10() 関数はデータが ppp に格納されていることを前提にして作られている．

このように関数の外にあるオブジェクトを関数内から参照するのは良い設計ではない．仮に行名の設定を諦め，rownames(ppp) の行を削除したとしてもこの関数は期待通り動かない．deparse() 関数を使って引数文字列から年度を取り出すコードも変数名の長さが変わっているため期待通りの結果にならないからだ．

関数を設計する場合は，必ず必要な情報を全て引数で渡し，関数の外の変数やオブジェクトにアクセスしないことが大切である．そこで，gdp.top10(my.ppp, 2018) のように第 1 引数にデータ・フレーム全体を渡し，第 2 引数で年度を指定するように修正する．

```
> gdp.top10 <- function(data, year) { # data は GDP データのデータフレーム
+     col <- paste("X", year, sep="")  # 取り出したい年度の列名を作成
+     index <- order(data[[col]], decreasing=TRUE)[1:10] # data[[col]] については後述
+     top10 <- data[index, col]
+     names(top10) <- rownames(data)[index] # 引数のデータフレームから行名を取得！
+     result <- data.frame(top10)
+     colnames(result) <- paste("GDP.", year, sep="")   # 列名を修正
+     result
+ }
```

まず定義 1 行目で指定された年度から paste() 関数で列名を構築し col 変数に列名を記録している．定義 2 行目の order() 関数はベクトルを引数に取るため，data[[col]] を使ってデータ・フレームから col 列をベクトルとして取り出している．[[]] 演算子はまだ学んでいないが，列番号か列名を使ってデータ・フレームから列をベクトルとして取り出すことができる演算子である．今回列名は変数に格納されているため，$演算子を使って直接列を指定することはできない．第 8 回のプリントの第 5 節で [] と [[]] の違いを詳しく解説しているのでそちらを参照されたい．

実際にこの関数を my.ppp に対して実行してみると無事以下の結果を得る．

```
> gdp.top10(my.ppp, 2018)
                     GDP.2018
Qatar                126898.43
Macao SAR, China     123892.17
Luxembourg           113337.42
Singapore            101531.63
Ireland               83203.39
Brunei Darussalam     80920.05
United Arab Emirates  75075.26
Kuwait                72897.56
Switzerland           68060.94
Norway                65510.59
```

以降の練習問題のために my.ppp を ppp に戻しておく．

```
> ppp <- my.ppp
> rm(my.ppp)
```

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, 2018, 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，gdp.ranking(ppp, 2018, c(1, 26, 35)) の様

に特定の順位を表示するにはベクトルで指定する.

```
> gdp.ranking(ppp, 2018, 25:35)
               GDP.2018 ranking
Finland        48416.94      25
Canada         48130.26      26
Bahrain        47303.05      27
Euro area      46207.32      28
United Kingdom 45973.57      29
OECD members   45623.90      30
France         45342.40      31
European Union 43737.74      32
Japan          42797.46      33
Malta          42581.10      34
Oman           41859.93      35
> gdp.ranking(ppp, 2018, c(1, 26, 35))
        GDP.2018 ranking
Qatar  126898.43       1
Canada  48130.26      26
Oman    41859.93      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, 2018, c("United States", "Germany", "Japan", "Korea, Rep.", "China"))
              GDP.2018 ranking
United States 62794.59      12
Germany       53074.54      20
Japan         42797.46      33
Korea, Rep.   40111.78      38
China         18236.61      91
```

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, year) {
+     gdp.ranking(data, year, 1:10)        # gdp.ranking に処理を委譲する
+ }
> gdp.worst10 <- function(data, year) {
+     gdp.ranking(data, year, nrow(data):(nrow(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 を取り除ける．もしくは，gdp.worst10() の定義を以下のよう修正しても解決できる．

```
> gdp.worst10 <- function(data, year) {
+     col <- paste("X", year, sep="")
+     len <- sum(!is.na(data[[col]])) # NA を除いたデータ数．sum は TRUE の和もとれる
+     gdp.ranking(data, year, len:(len-10))
+ }
> gdp.worst10(ppp, 2018)
                         GDP.2018 ranking
Burundi                   744.1821     230
Central African Republic  859.9356     229
Congo, Dem. Rep.          932.1720     228
Niger                    1063.4218     227
Liberia                  1308.6295     226
Malawi                   1310.9956     225
Mozambique               1459.6984     224
Sierra Leone             1601.9740     223
Togo                     1773.8966     222
Guinea-Bissau            1799.0679     221
Haiti                    1866.6176     220
```

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 3 行にまとめることができた．

5 記述統計

データの大まかな特徴を把握するには平均，標準偏差，最頻値（モード），中央値（メディアン）等の記述統計量を用いる．ちなみに GDP 統計はほぼ連続値でそれぞれの値が異なるため最頻値は用いない．平均と標準偏差はすでに学習済みであるので，ここでは中央値を算出する summary() 関数を紹介しよう．

R で基本的な統計量を得るのに最もよく使われるのが summary() 関数である．summary() はデータの最大値，最小値，平均の他，データ数の 1/4 ずつに分割する第 1 四分位点，中央値（メディアン），第 3 四分位点の値を出力する．

一般に所得データは非常に裕福な人が平均を引き上げるので，典型的な所得額を知りたい場合は中央値を用いるのが良い．一人当たり GDP も所得統計なので同様のことが言える．

```
> summary(ppp$X2018)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
   744.2   4854.1  14520.9  21060.0  29245.7 126898.4       34
```

上の結果から一人当たり GDP の平均は$21,060 だが，中央値はもっと低く約$14,520 であることが分かる．また，下位 25%の国の一人当たり所得が$4,854 以下で，最貧国の一人当たり所得は年間わずか$744.2 であることが分かる．

注意しなければならないのは，日本やアメリカの通貨が最貧国の通貨より強いために我々から見てこのような低い値になっているのではない．第 5 回のプリントで説明したように，ここで使用している一人当たり GDP は各国の購買力が等しくなるように調整された為替レート（PPP レート）を用いてドルに換算されている．

例えば，日本の学生が 1ヶ月アルバイトして稼いだ 10 万円を持って東南アジアの国に行くと，日本の円が現地の通貨より強いので，かなり裕福な思いができるだろう．しかし，もしこの 10 万円を PPP レートで換算して持って行ったならば，現地の学生が 1ヶ月アルバイトして稼いだ程度の生活しかできないことを意味する．したがって，最貧国の一人当たり所得が日本の 2 割にも満たないならば，それは日本で 2 割未満の所得で生活する水準であることを意味する．

5.1 練習問題

最貧国と第 1 四分位点の一人当たり GDP が日本の GDP の何割か比率を求めたい．例えば 2018 年の最貧国の第 1 四分位点の GDP は$4854.1，日本の GDP は$42797.46 なので，求めたい比率は，$4854.1/42797.46 = 0.1134203$ である．

このとき，以下の問いに答えよ．

1. 最貧国と第 1 四分位点の値を直接プログラム内に記述するのではなく，`summary()` 関数の結果を利用したコードを記述せよ．当然，日本の GDP も直接入力するのではなくデータフレームから抽出すること．
   （ヒント）`summary()` の結果を一旦変数に格納し，変数の中身の値の構造を調べてみるとよい．

2. 日本だけでなく指定した国との比率を計算できるようにしたい．GDP のデータフレーム，国名，および年度を引数にとり，最貧国および第 1 四分位点の GDP と，引数に指定した国の GDP との比率を計算して返す関数を作成せよ．その関数を使ってアメリカとの比率を求めよ．

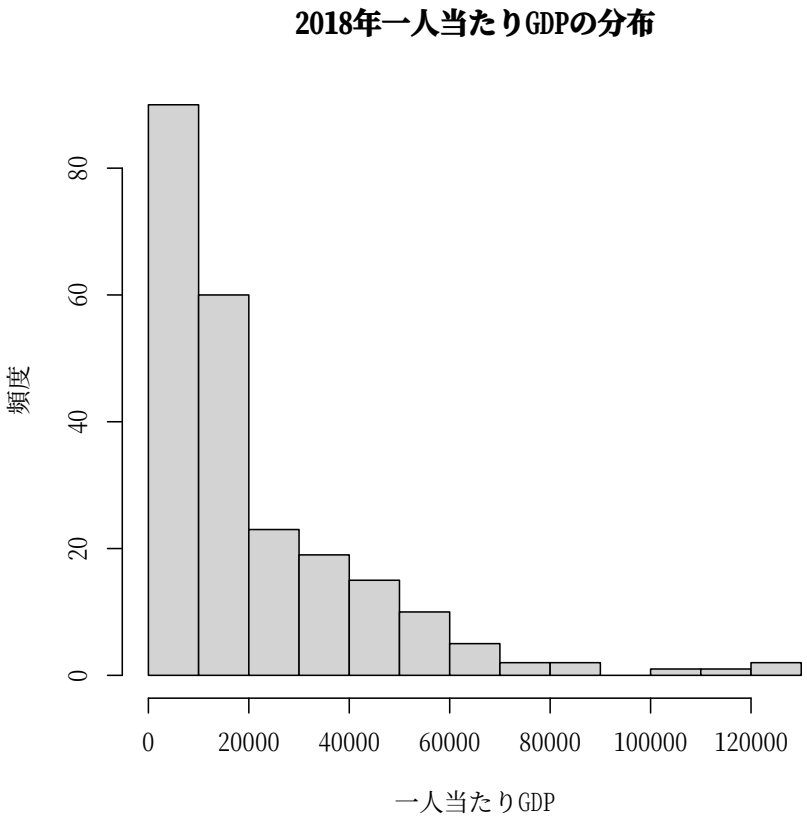
3. 上で求めた関数で，使用する GDP の年度を指定しなかった場合にはデフォルトで 2018 年のデータが利用されるように変更せよ．

6 ヒストグラム

`summary()` 関数は四分位点を算出してくれるが，データの分布を見るにはやはりグラフが適している．ここではヒストグラムの描画方法を学ぶ．

ヒストグラムは `hist()` 関数で描画する．第 1 引数に分布を描画したいデータを指定する．描画のためにたくさんのオプションがあるので，残りの引数は順番にこだわらず，全て名前付き引数を利用して指定するのが良い．`breaks` は，ヒストグラムの柱の本数を指定する．指定しないと，データから適切な数に分割してくれる．`xlab` と `ylab` はそれぞれ x 軸と y 軸のラベルである．`main` は図のタイトルを指定する．

```
> hist(ppp$X2018, breaks=10, xlab="一人当たり GDP", ylab="頻度",
+      main = "2018 年一人当たり GDP の分布")
```



Mac ではラベルやタイトルが文字化けする場合がある．この場合は，`hist()` を実行する前にフォントを以下のように指定すると解決する．

```
> par(family="Japan1Ryumin")  # フォント family を明朝体に設定
```