福岡教育大学紀要，第67号，第４分冊，1―14（2018）

R を利用した PISA の分析

Analysis of PISA data with R

川　口　俊　明

Toshiaki KAWAGUCHI
（学校教育講座）

（平成29年10月２日受理）

1. R を利用した PISA 調査の分析を行う意義

2000 年から始まった PISA（Programme for International Student Assessment）は，2017 年現在，既に６度目のサイクルを終え，参加国の学力実態と，その変化に関するさまざまなデータを蓄積している。PISA のデータは，そのほとんどがオンライン上で公開されており，特別な許可なく誰でも利用できる。公開された学力データがほとんど存在しない日本において，PISA は国内の学力実態を知ることができる貴重な資料の一つである。そのため，昨今の日本の学力研究では，PISA を利用したものが増加しつつあり，日本の学力実態が（限界はあるものの）明らかにされつつある（Matsuoka 2012，古田 2016 等）。

一方で，PISA のデータは，まだ十分に活用されているわけではない。たとえば，日本の大多数の教育研究者にとって，PISA のデータはまだまだ縁遠い存在である。PISA の結果に言及する研究・報告書は多いのだが，その中には，自身でデータを分析すれば，より深い考察が可能になったのではないかと思うものが少なくない。また，教育研究者に限らず，近年，日本で盛んに行われるようになった小・中学校を対象とした学力調査の設計・分析に関わる人々にとっても，PISA で採用されている各種技法（Item Response Theory，Sample Weights，Replication methods，Plausible Values 等）は，最低限知っておくべき「教養」である [(1)]。これら各種の技法を知らなければ，学力調査を設計・運用することはおろか，研究成果を正確に読み解くことすら難しい。その意味でも，PISA で利用されている各種の統計技法の概要と，その実際の利用法をわかりやすい形で発信していく必要がある。

この点，既に教育研究分野で普及している統計解析ソフトの SPSS を利用して，PISA データの扱い方を解説した川口（2012）がある。ただ，SPSS は 10 万円を超える高額なソフトウェアであり，潤沢な研究予算を持たない者には利用しづらい。昨今の国立大学を中心にした教育研究費の削減も，こうした傾向に拍車をかけており，教育研究において SPSS を利用することは，もはや現実的な選択肢とは言えない。

こうした状況を踏まえ，本稿では，無償で利用できる R（https://www.r-project.org/）を利用した PISA の分析法を，実際のコードとともに紹介する。R の第一の利点は，何よりも無償であるという点である。既に述べたように，大学に所属する研究者であっても SPSS を利用できる者は減少しつつある。ましてや，大学等の研究機関に所属しない者にとっては，SPSS の利用を前提とするのは酷である。その点，R はインターネットにアクセスできれば自由にインストールできるため，SPSS より遙かに入手が容易である。

R を利用する第二の利点は，分析結果を再現しやすいという点にある。SPSS はマウスを中心に操作することが多い。マウスによる操作は便利ではあるが，利便性と引き替えに，分析結果を再現することが難しくなりがちである。論文執筆後に過去の分析結果を再現する必要が生じる，あるいは，異なるデータでまったく同じ分析を繰り返す必要が生じることは，それほど珍しいことではない。昨今では，研究不正の問題と絡んで，分析結果の再現性を保つことの重要性が増しており，その意味でも R は利用する価値がある [(2)]。

以下では，PISA データの入手・R に関する基本的な事項を説明する。続いて，PISA 調査のデータを R

で分析するコードを紹介する。その上で，2009 年から 2015 年の PISA データを用いて，日本を含む 7 ヵ国の学力実態に関する分析例を示す。

2. PISA と R に関する基礎知識

2.1. データの入手

はじめに，PISA のデータをどうやって入手すればよいか，確認しておこう。PISA のデータは，調査実施主体である OECD のウェブサイト（https://www.oecd.org/pisa/data/）で公開されている。ここからデータをダウンロードして利用すればよい。ただ，PISA のデータは，利用者が SPSS や SAS を使うことを前提としたファイル形式で公開されている。そのため，SPSS や SAS を利用せずに，R でデータを扱うためには，少し工夫が必要になる。

まず，最新の PISA2015 については，データが SPSS または SAS のファイル形式（拡張子が sav か sas）で公開されているため，R の library（さまざまな処理をする関数をまとめたもの。ユーザーが制作したものも多い）から，foreign を利用することでデータを読むことができる。ただし PISA は，参加したすべての国のデータを一つのファイルに収めているため，一般的にファイルサイズが大きく，メモリの大きい PC（PISA2015 の場合，最低でも 8 GB は必要になる）でしか実行できないという点は注意しておく必要がある。

```
> library(foreign)
> pisa <- read.spss("CY6_MS_CMB_STU_QQQ.sav",
    to.data.frame = TRUE, use.value.labels = FALSE) # PISA2015 のデータを読み込み
```

次に，PISA2000 から 2012 に関しては，公開されているデータは固定長のテキスト（txt）形式であり，それを SPSS か SAS のシンタックスを使って変換して利用することになっている。R でも固定長のテキストデータが扱えないわけではないのだが，PISA データのファイルサイズが大きいために，R だけで処理を完結するのはかなり困難である。そこで，python や ruby といったプログラミング言語を利用して，R で扱いやすい csv 形式にデータを整形する方法が考えられているので，こうしたプログラムを利用するとよい [(3)]。なお，PISA のデータを csv に変換したファイルを配布しているウェブサイトも少なくないので，プログラミングに自信がない場合は，こうした変換済のデータを利用するとよいだろう。

2.2. R に関する基礎知識

分析に入る前に，R について簡単に紹介しておく。R はフリーの統計ソフト（あるいは統計解析向けのプログラミング言語）である。日本でも 2010 年頃から R を紹介する書籍が急速に増えており，利用するためのハードルは日々下がっていると言ってよい。ただ，R の書籍は，その難易度が大きく異なり，簡単な入門書から言語仕様を詳しく解説した高度なものまで幅広い。一般に，日本の教育研究者はプログラミングに詳しくない者が多い傾向にあるから，有名な書籍であっても Wickham（2014）のような言語仕様に踏み込んだものは，最初は避けた方がよい。R を社会調査や教育研究に利用するための入門書としては，杉野（2016）がわかりやすい。また，統計学にある程度の馴染みがあれば，奥村（2016）も参考になる。以上の 2 冊は，R のインストールや基本的な使い方についても触れている。

それでは，本稿で利用する R の機能を簡単に解説する。はじめに，ベクトル（vector），リスト（list），コメントといった機能を知っておく必要がある。ベクトルは R の処理の基本の一つである。ベクトルを作成するには，c（ ）で作成する。また，x <-c（1, 3, 5）のようにすることで，x という変数にベクトルを代入することができ，以降，x と入力すれば，何度でもベクトルの中身を表示することができるようになる。その他，1 : 10 と入力すると，1 から 10 まで連続したベクトルを生成できる。

R には，ベクトルを扱うさまざまな関数が用意されており，たとえば平均を計算する mean 関数を使うことで，ベクトルの平均を計算することができる。なお，“boy” のように，文字データを含むベクトルを作成することもできるが，c（1, “boy”, 2）のように，文字と数字が混在した場合，［“1” “boy” “2”］といった具合

に，すべて文字で揃えられてしまう。ここで，リスト（list）を使うと，文字と数字のような情報を保持することができる。# は R ではコメントであり，そこから右に書いたコマンドは R に評価されなくなる。

```
>  x <- c(1, 3, 5) # ベクトルの作成
>  x # x の表示
[1] 1 2 5 # ベクトルの中身が表示される
>  1:10 # 1 から 10 まで連続した要素を持つベクトルを生成
[1] 1  2  3  4  5  6  7  8  9  10
>  mean(x) # x の平均を計算
[1] 2.666667
>  y <- c(1, “boy”, 2) # 数字と文字が混在するベクトルを作成
>  y # y を表示
[1] "1"   "boy"    "2"
>  z <- list(1, “boy”, 2) # リストの作成
>  z # 情報が保持される。結果は省略
```

R のベクトルには，統計解析のための便利な性質がいくつかある。単純にベクトルから数値を引いた場合，ベクトルのそれぞれの要素からその数値が引かれる。また，ベクトルを 2 乗した場合，ベクトルの要素がそれぞれ 2 乗される。これは，分散を計算する場合に有効に使える機能である。

```
>  x <- 1:10 # 1 から 10 まで連続した要素を持つベクトルを生成
>  x – 5 # それぞれの要素から 5 を引く
[1]  -4  -3  -2  -1  0  1  2  3  4  5
>  x ^ 2 # それぞれの要素を 2 乗する
[1]   1  4  9  16  25  36  49  64  81  100
```

R では，回帰分析などの基本的な統計処理は，特別なパッケージを導入せずとも，はじめから利用できる。回帰分析を行う関数は，lm 関数である。この関数は，lm（y ～ x）のように書くことで，y を従属変数，x を独立変数とする回帰分析の結果を出力する。R では，est < -lm（y ～ x）のようにすることで，回帰分析の結果を変数（ここでは est）に格納し，あとの計算で結果を再利用することができる。その際，すべての結果を利用するのではなく，回帰係数の推定値だけ利用することもできる。lm 関数の場合，est$coefficient のように，$ を使うと，必要な要素を限定することができる。

```
>  x <- rnorm(100, 0, 10) # 平均 0，標準偏差 10 の正規分布から 100 個の値を生成
>  y <- rnorm(100, 0, 10)
>  est <- lm(y ~ x) # y を従属変数，x を独立変数とする回帰分析
>  est # 推定値の表示。結果は省略
>  est$coefficient # 推定値が格納されている値を取り出す。結果は省略
```

ここまでの例では，分析に利用するベクトルを自分で入力していた。ただ，一般的な分析の際に，一から自分でデータを入力していくことは少ない。普通は，既存の csv ファイル等を R に読み込んで利用するだろう。ここで，PISA2009 の日本のデータ（pisa2009stu_JPN.csv）を，read.csv 関数で読み込む例を示す。なお，この csv データは，python を利用して生成したものである。

```
> jpn <- read.csv("pisa2009stu_JPN.csv") # csv ファイルを読み込み，結果を jpn に代入
```

データセットの変数名を確認する場合は，names 関数を利用する。今回の例で，すべての変数名を表示する場合は，names（jpn）とすればよい。次の例では，最初の 5 列の変数名を確認している。

```
> names(jpn)[1:5]
[1] "CNT" "COUNTRY" "OECD" "SUBNATIO" "SCHOOLID"
```

各変数の中身を確認する場合は，jpn$CNT，jpn$OECD・・・といった具合に，$ マークの後ろに変数名をつければ，その変数の要素がベクトルの形で表示される。ここでは，川口（2012）に倣い，HISEI（両親の職業的地位）を例に，その要素を表示してみよう。なお，すべてを表示すると数千件のデータが表示されるため，head 関数で最初の数件に限定している。

```
> head(jpn$HISEI)
[1] 49 38 43 33 38 43
```

R で変数の要素を表示する方法は，他にもいくつかある。たとえば，jpn$HISEI は，jpn［[“HISEI”]］と書いてもよい。jpn$HISEI との違いは，前者は jpn$HISE のように入力したとしても，他に該当する変数がなかった場合，jpn$HISEI を表示するが，後者は jpn［[“HISE”]］と書いたらエラーになるという点である。普段の分析では $ の方が使いやすい。ただ，PISA データを利用する場合は，replication weights をうまく扱うために，後者の記法が必要となる。

```
> head(jpn[["HISEI"]]) # head(jpn$HISEI)と同じ
[1] 49 38 43 33 38 43
> head(jpn[["HISE"]]) # エラー
> head(jpn$HISE) # jpn$HISEI が表示される
```

データ分析において，何らかの理由でデータが欠落した欠損値はつきものである。R の場合，データの欠損は NA で表現する。R の table 関数を使って度数分布を表示するとわかるが，日本の HISEI 変数には 97，98，99 という欠損値扱いする必要のある値が存在する。これをそのままにして分析を進めると，平均値を計算したときに計算結果がおかしくなってしまう。そこで R に欠損値の情報（HISEI の場合は，90 以上が欠損値）を伝えよう。まず，jpn$HISEI［jpn$HISEI > 90］とすると，HISEI 内で 90 を超える値がすべて表示される。そこで，この値をすべて欠損値に指定すればよい。具体的には，次のようになる。

```
> table(jpn$HISEI) # 度数分布表を表示する。結果は省略
> jpn$HISEI[jpn$HISEI > 90] <- NA # 90 を超える値をすべて欠損値にする
> table(jpn$HISEI) # 再び度数分布表を表示すると，90 以上は表示されない。結果は省略
> summary(jpn$HISEI) # summary 関数を使うと，平均値や NA の数がわかる。結果は省略
```

R で PISA を分析する場合，自分で関数を作成する方法を知っておく必要がある。データを分析する際は，同じ処理を何度も繰り返すことも多い。こうしたときに同じコードを何度も書くのは面倒である。コピーアンドペーストを繰り返してもよいが，それではコードの修正が必要になったとき修正漏れが生じやすく，ミスの原因となる。そのため R では，一続きの処理を，新たな関数としてまとめることができる。

新しい関数は，function によって作成する。たとえば，与えた数に 2 を足す関数は，次の plus2 のように定義できる。また，1 行に収めるのであれば，{} を書かないこともできる。

```
> plus2 <- function (x) {
>   x + 2
> }
> plus2(2) # 与えた数に 2 を足す関数ができる
[1] 4
> plus2 <- function (x) x + 2 # { } を省略して 1 行に書くこともできる
```

最後に，R を利用した分析の際は，ベクトル内の要素に，同じ処理を繰り返し適用する必要が生じることがある。R では，こうした繰り返し処理を手軽に行うことができる apply 族という関数が用意されている。ここでは，sapply と lapply を紹介しよう。結果を再利用しやすいのは sapply だが，複数の回帰分析の結果を格納する場合は lapply を使う必要がある。以下に，簡単な例を載せておく。なお，この程度の処理であれば，ベクトル自体に 2 を足す（x + 2）方が早い。

```
> sapply(1:10, function(x) x + 2) # 1 から 10 までの数に，それぞれ 2 を足す
[1]  3  4  5  6  7  8  9  10  11  12
> lapply(1:10, function(x) x + 2) # 1 から 10 までの数に，それぞれ 2 を足す。結果は省略
```

ここまで R の基本的な機能に触れた。続いて，これらの機能を利用して PISA データを扱う方法を述べる。

3. PISA で利用されている技法

川口（2012）が既に説明しているように，PISA 調査には，Sample weights，Replicate methods，Plausible Values といった，現在の日本の学力調査ではあまり利用されていない技法が使われている。ここでは，こうした技法を R で扱う方法を見ていこう。なお，こうした技法の詳細については，川口（2012）や OECD（2009）を参照してほしい。

3.1. Sample weights

PISA データを扱う場合は，Sample weights を常に考える必要がある。今，PISA2009 年の日本のデータを使って，HISEI 変数の平均値を計算することを考えよう。前頁の欠損の指定方法に従って HISEI 変数の欠損値を指定していた場合，mean 関数で平均値を計算すると NA が返ってきてしまう。これは，NA を含むベクトルの平均は，計算できないからである。NA を除いて平均値を計算するためには，mean 関数に，na.rm = TRUE という引数を与える必要がある Sample weights を考慮して HISEI の平均値を計算したい場合は，weighted.mean 関数に，計算したい変数（HISEI）と，weight を示す変数（W_FSTUWT）を同時に与える必要がある。

```
> mean(jpn$HISEI) # NA になる
[1] NA
> mean(jpn$HISEI, na.rm = TRUE) # NA にならない
[1] 51.48862
> weighted.mean(jpn$HISEI, jpn$W_FSTUWT, na.rm = TRUE) # weight を使う
[1] 51.48839
```

3.2. Replication method

次に，Replication method を取り上げよう。PISA 調査は，その設計上，平均値の標準誤差（推定のバラツキ）を容易に算出することができない。PISA の標準誤差 σ の 2 乗は，次の式で計算する。ここで，$\hat{\theta}_i$ は，PISA データセット内にある W_FSTR1 から W_FSTR80 と名付けられた 80 の replication weights を利用して計算した平均値であり，$\hat{\theta}$ は W_FSTUWT を使ったときの平均値（最終的な平均の推定値）である。

$$\sigma^2 = \frac{1}{G(1-k)^2}\sum_{i=1}^{G}\left(\hat{\theta}_i - \hat{\theta}\right)^2 = \frac{1}{80(1-0.5)^2}\sum_{i=1}^{80}\left(\hat{\theta}_i - \hat{\theta}\right)^2 = \frac{1}{20}\sum_{i=1}^{80}\left(\hat{\theta}_i - \hat{\theta}\right)^2$$

この式は，PISA で標準誤差を求める場合，$\hat{\theta}_1$ から$\hat{\theta}_{80}$ を計算し，そこから$\hat{\theta}$ を引き，さらにそれぞれを 2 乗して・・・という処理をしなければならないということを示している。ここで，単純に $\hat{\theta}_1$ から $\hat{\theta}_{80}$ を計算する R のコードを書くと，次のようになる。

```
> weighted.mean(jpn$HISEI, jpn$W_FSTR1, na.rm = TRUE) # θ̂1を計算する
> weighted.mean(jpn$HISEI, jpn$W_FSTR2, na.rm = TRUE) # θ̂2を計算する
> # 以下，80 まで繰り返す
> weighted.mean(jpn$HISEI, jpn$W_FSTR80, na.rm = TRUE) # ^80を計算する
```

これは明らかに面倒なので，うまく処理する方法を考えよう。names 関数で確認すると，データセット内で replication weights は，W_FSTR1，W_FSTR2・・・W_FSTR80 という変数名がつけられている。ここで，R の paste0 という関数を使うと，文字と数字を連結し，新しい文字を作ることができる。具体的には，paste0（“W_FSTR”, 1）とすれば “W_FSTR1” が，paste0（“W_FSTR”, 80）とすれば “W_FSTR80” が生成される。さらに，jpn$HISEI と jpn［[“HISEI”]］が同じであることを利用すると，replication weights を使った $\hat{\theta}_1$ の計算は，次のように書き換えることができる。

```
> # 以下は，すべて同じθ̂1の値を返す
> weighted.mean(jpn$HISEI, jpn[[paste0("W_FSTR", 1)]], na.rm = TRUE)
> weighted.mean(jpn$HISEI, jpn[["W_FSTR1”]], na.rm = TRUE)
> weighted.mean(jpn$HISEI, jpn$W_FSTR1, na.rm = TRUE)
```

さらに，sapply 関数を利用すれば，80 回の replication weights を使った計算を 3 行でまとめることができる。ここでは，hisei_rp という変数に，replication weights を使った 80 回の計算結果を格納している。

```
> hisei_rp <- sapply(1:80, function(i) {
>   weighted.mean(jpn$HISEI, jpn[[paste0("W_FSTR", i)]], na.rm = TRUE)
> })
> hisei_rp # θ̂1から θ̂80の計算結果が表示される。結果は省略
```

まとめると，PISA データを利用して変数の平均値を計算する方法は，次のようになる。

```
> hisei_wm <- weighted.mean(jpn$HISEI, jpn$W_FSTUWT, na.rm = TRUE)
> hisei_rp <- sapply(1:80, function(i) {
>   weighted.mean(jpn$HISEI, jpn[[paste0("W_FSTR", i)]], na.rm = TRUE)
> })
> rp_var <- sum( (hisei_rp - hisei_wm) ^ 2) / 20
> sqrt(rp_var)
```

コードの内容を簡単に説明しておこう。まず，1 行目では，W_FSTUWT を利用した平均値を計算している。これが最終的な平均の推定値（θ̂）になる。続いて 2 行目から 4 行目は，replication weights を利用した 80 回の計算を行っている。5 行目がややわかりにくいが，ここで標準誤差 σ の 2 乗を計算している。コードをよく見れば，まず，80 回の推定値を格納した hisei_rp から hisei_wm を引き，その結果をそれぞれ 2 乗（^2）したあとに sum 関数で合計して 20 で割るという作業をしていることがわかるだろう。最後に 6 行目で sqrt 関数を使って平方根を計算し，標準誤差を算出している。

ここまでに行った weight を使う計算は，これから何度も行うため，一連の処理を関数としてまとめて再利用できるようにしよう。新たな pisa_mean 関数として，ここまでの処理をまとめたものが，次のコードである。なお，return は，計算結果（今回は ans）を明示的に返却する時に利用する。

```
> pisa_mean <- function(x, dat) {
>   wm <- weighted.mean(dat[[x]], dat[["W_FSTUWT"]], na.rm = TRUE)
>   rp <- sapply(1:80, function(i) {
>     weighted.mean(dat[[x]], dat[[paste0("W_FSTR", i)]], na.rm = TRUE)
>   })
>   rp_var <- sum( (rp - wm) ^ 2) / 20
>   rp_se <- sqrt(rp_var)
>   ans <- c(wm, rp_se)
>
>   return(ans)
> }
>
> pisa_mean(x = "HISEI", dat = jpn) # pisa_mean を利用した計算
[1] 51.4883928  0.2707827
```

上記のように，pisa_mean 関数を新たに定義することで，以降は pisa_mean（ ）を使い，簡単に変数の平均値とその標準誤差を計算することができるようになる。

3.3. Plausible Values（PVs）

PISA では，個々の児童の成績は，PVs という技法を使って算出されている。PVs とは，個々の生徒の成績を点推定するのではなく，事後的にあり得る 5 つの値（Plausible Values）として生成し，そのデータを利用した分析結果を統合することで，より正確な集団の推定値を得ようとする技法である。PISA の分析で成績に関する変数を扱うときは，5 つの PVs（PV1 から PV5）に対して，同じ計算を繰り返した上で，最後に結果を合成する必要がある。具体的には，求めたい推定値（$\hat{\mu}$ とその標準誤差 $\sigma_{(error)}$）は，次の式で計算する。

$$\hat{\mu} = \frac{1}{5}(\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3 + \hat{\mu}_4 + \hat{\mu}_5)$$

$$\sigma^2_{(error)} = U + \left(1 + \frac{1}{5}\right) B_M \quad U = \frac{\sigma^2_{(\hat{u}_1)} + \sigma^2_{(\hat{u}_2)} + \sigma^2_{(\hat{u}_3)} + \sigma^2_{(\hat{u}_4)} + \sigma^2_{(\hat{u}_5)}}{5} \qquad B_M = \frac{1}{4}\sum_{i=1}^{5}(\hat{\mu}_i - \hat{\mu})$$

式からわかるように，PVs を利用した場合の平均値（$\hat{\mu}$）は，5 つの PVs の平均値（$\hat{\mu}_1$ から $\hat{\mu}_5$）の平均である。そして，標準誤差 $\sigma_{(error)}$）の 2 乗は，5 つの PVs の分散の平均（U）に，PVs の平均値の分散（B_M）の 1.2 倍（1 + 1/5）を足したものである。

この処理を R で行うことを考えよう。ここでは，PISA2009 の日本の読解リテラシーを例に，PV1 から PV5 を計算する。PISA2009 の読解リテラシーは，PV1READ から PV5READ という変数名でデータセットに格納されている。先ほど作った pisa_mean 関数を利用することで，PV1 から PV5 を使ったときの推定値は，容易に計算することができる。

```
> pvs <- sapply(1:5, function(i) {
>   pisa_mean(x = paste0("PV", i, "READ"), dat = jpn)
> })
> pvs # 計算結果を表示
        [,1]        [,2]        [,3]        [,4]        [,5]
[1,] 520.304443 519.591235 519.628738 519.938655 519.825541
[2,]   3.413471   3.484748   3.412405   3.517011   3.427533
```

ここで PV1 から PV5 までの計算結果は pvs に格納され，行列（matrix）形式で表示されている。ここで，pvs の一行目を平均すれば求める推定値 $\hat{\mu}$ が計算できる。また，二行目の要素をそれぞれ 2 乗したあとに合計したものと U と，一列目の分散 B_M を 1.2 倍したものを足せば，求める標準誤差 $\sigma_{(error)}$）の 2 乗が計算できる。行列の一行目の要素は，pvs［1, ］で取り出せる。二行目は，pvs［2, ］である。

```
> est <- mean(pvs[1, ])
> se <- sqrt(mean(pvs[2, ] ^ 2) + 1.2 * var(pvs[1, ]))
> c(est, se) # 推定値を表示
[1] 519.857722   3.465631
```

3.4. 回帰分析

最後に，回帰分析について示しておこう。ここでは，読解力（READ）の点数を従属変数に，HISEI を独立変数にとった回帰分析を考える。R の回帰分析は，lm 関数を利用することで行える。その際，weights を引数として与えることで，Sample weights を考慮できる。他の基本的な考え方は，これまでと同じである。

```
> est <- lm(PV1READ ~ HISEI, data = jpn, weights = jpn$W_FSTUWT)
> est # 最終的な係数の推定値を表示
Coefficients:
(Intercept)       HISEI
    462.014       1.269
> lm_rp <- lapply(1:80, function(x) {
>   lm(PV1READ ~ HISEI, data = jpn, weights = jpn[[paste0("W_FSTR", x)]])
> })
> lm_rep <- sapply(lm_rp, function(x) x$coefficients)
> err_var <- apply( (lm_rep - est$coefficients) ^ 2, 1, sum) / 20
> sqrt(err_var) # 標準誤差の推定値を表示
(Intercept)       HISEI
  6.6147407   0.1071954
```

上記のコードの 1 行目は，回帰分析を行った結果を，est という変数に格納している。2 行目で推定結果を表示している。3 行目から 5 行目では，80 回の回帰分析を行い，その結果を lm_rp に格納している。その上で，6 行目では，lm_rep から切片・係数・標準誤差の値を取り出している。さらに，7 行目で推定値の分散を計算し，それを合計して 20 で割っている。これが標準誤差の 2 乗（err_var）である。最後に 8 行目で，err_var の平方根を計算すれば，それが標準誤差となる。

紙幅の都合上省略するが，回帰分析の従属変数に成績を用いる場合，先ほどの PVs の平均値を求めた計算法と同じように行う。つまり，PV1 から PV5 の成績データについて回帰分析を行い，平均値の場合と同じように合成する。

ここまで R を利用した PISA 調査の分析方法について解説してきた。実は R には，こうした処理をまとめた library が存在する。それが，intsvy である。intsvy は CRAN という R の library を集めたサイトに登録されており，そこからダウンロードして使うことができる。利用方法は，次の通りである。なお，推定値は省略しているが，出力は，すべてここまで計算した値と一致する。

```
> install.packages("intsvy") # package のインストール。2 回目以降は不要
> library(intsvy) # intsvy の呼び出し
>
> pisa.mean(variable = "HISEI", data = jpn) # HISEI の平均値を計算
> pisa.mean.pv(pvlabel = "READ", data = jpn) # 読解力の平均値を計算
> pisa.reg(y = "PV1READ", x = "HISEI", data = jpn) # PV1READ を従属変数，HISEI を独立変数とした回帰分析
> pisa.reg.pv(x = "HISEI", pvlabel = "READ", data = jpn) # 読解力を従属変数，HISEI を独立変数とした回帰分析
```

以上が R を利用した PISA 調査の分析方法である。intsvy は簡易な分析を可能にするが，あまりにも簡単すぎて裏で何をやっているかわからないということになりがちである。その点，ここまでに示した R のコードを理解していれば，PISA の調査設計に変更が加えられても対応できるし，似たような設計の学力調査（たとえば TIMSS）も容易に分析できるようになる [(4)]。こうした応用力を磨くためにも，R の利用は有用である。

4. 分析例―各国の学力格差はどう変化したか

最後に，R を利用した分析の一例として，日本を含む，いくつかの国・地方の 2009 年から 2015 年までの学力格差の変化を分析してみよう。ここでは，日本（JPN），香港（HKG），韓国（KOR），イギリス（GBR），フランス（FRA），ドイツ（DEU），シンガポール（SGP）の 7 カ国を分析の対象とする。これは，現在，著者が所属する研究グループの国際調査班が対象にしている国・地域が，この 7 ヵ国だからである [(5)]。この調査では，各国の学校教育を，それぞれの学力格差や教育政策の観点から比較することを目的としているのだが，その際，何らか共通の物差があった方が，各国の事例を位置づけやすい。そこで，この 7 ヵ国の PISA のデータを分析することを通して，各国の学校教育の位置づけを示す見取り図を示してみよう。

PISA 調査では，生徒の SES（Socio Economic Status：社会経済的背景）を示す指標として，一般に ESCS（PISA Index of Economic, Social and Cultural Status）指標が利用されている。これは，生徒に尋ねた，両親の学歴，職業，家庭にある本の冊数等の回答結果を合成して作成された指標である。ただ，これまでの ESCS 指標は，毎回，OECD 加盟国の平均が 0，標準偏差が 1 になるように調整されていたために，異なるサイクル間で値を比べることはできなかった。PISA2015 では，この問題に対応するため，PISA2015 と比較可能なように，これまでのサイクルの ESCS 指標を再計算したデータセットが公開されている [(6)]。このデータセットと，これまでの PISA のデータを結合することで，サイクル間で比較可能な ESCS 指標を

利用した分析が行える。

次のコードでは，まず，複数の国の csv ファイルを読み込み，それを結合したあと，新しい ESCS 変数（ESCST）を追加している。その上で，intsvy package を使って，各国の平均点などを計算している。複数のデータを結合する作業は，手作業でやるとたいへんだが，R を使えば，数行のコードを書くだけで実行することができる。

```
> library(foreign) # SPSS データを R に読み込む read.spss 関数を含む library
> library(intsvy)
> library(plyr) # 複数のデータを容易にマージする join_all 関数を含む library
>
> cnt <- c("JPN", "HKG", "KOR", "GBR", "FRA", "DEU", "SGP") # 国を指定
>
> # 複数のファイルを読み込んだ上で，1 つに結合
> stu <- do.call("rbind", lapply(cnt, function(x) {
>   read.csv(paste0("pisa2012stu_", x, ".csv"))
> }))
>
> # 新しい ESCS データが入った SPSS ファイルを読み込む
> escs <- read.spss("2012_escs.sav", to.data.frame = TRUE,
>   use.value.labels = FALSE)
> # 変数名が揃うように修正
> names(escs) <- c("CNT", "SCHOOLID", "StIDStd", "ESCST")
>
> # 共通の変数名を使ってデータセットをマージ
> stu12 <- join_all(list(stu, escs))
>
> # intsvy を使って分析。by = “CNT”を加えると国別に結果を表示できる。結果は省略
> pisa.mean.pv(pvlabel = “READ”, by = "CNT", data = stu12)
> pisa.reg.pv(pvlabel = “READ”, x = "ESCST", by = "CNT", data = stu12)
```

ここで示したコードを実行すると，各国の 2012 年の PISA 調査における読解リテラシーの平均点，および読解リテラシーを従属変数，ESCS を独立変数とした場合の各国の不平等度の指標（PISA では，学力格差の深刻さを示す指標として，回帰分析の分散説明率 R^2 値が利用されている）が手に入る。同様の処理を 2009 年と 2015 年のデータについて実施し，2009 年から 2015 年までの各国の変化を図示してみよう。

図 1 から図 3 は，ggplot2 という library を使って，読解リテラシー，数学リテラシー，科学リテラシーのそれぞれで，7 カ国の位置が 2009 年から 2015 年までにどのように変化したか描画したものである。図 1 が読解，図 2 が数学，図 3 が科学である。図の見方だが，縦軸は各国の平均点であり，上方向にあるほど，その国の平均点が高いことを示す。また，横軸は R^2 値であり，右方向にあるほど，その国の学力格差が深刻であることを示す。要するに，左上ほど格差が小さく平均点の高い国・地方であり，理想的な状態だということである。なお，紙幅の都合上，図示するためのコードは省略する。ggplot2 については，基本的な使い方を解説した書籍もあるので，そちらを参照してほしい（Chang 2012）。

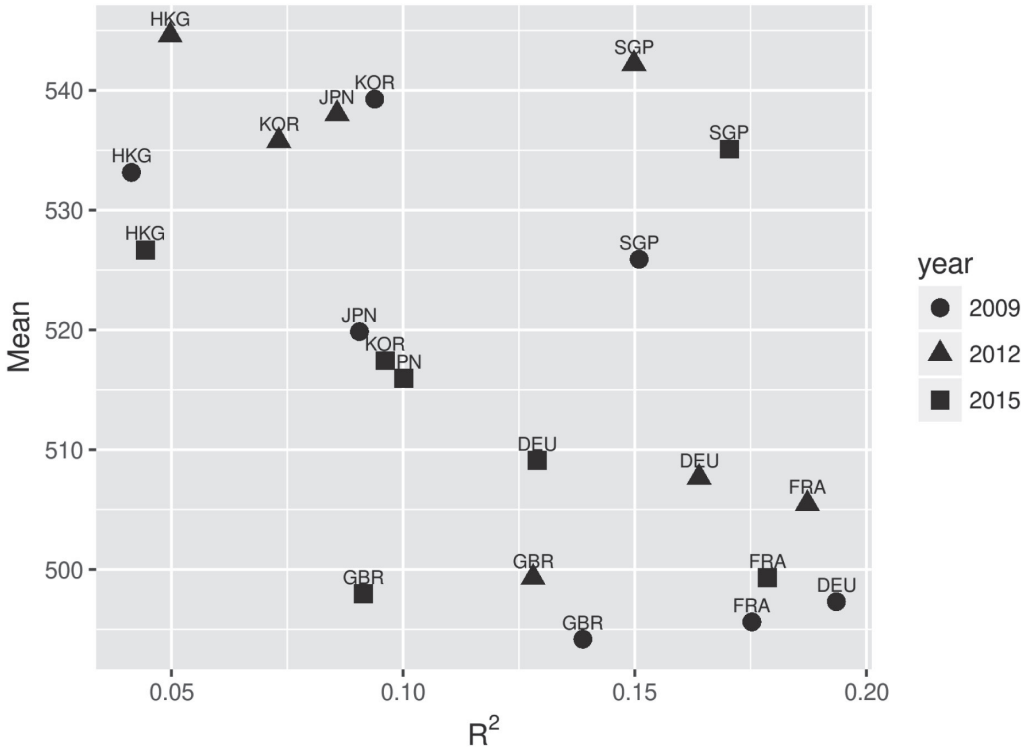

図 1. 読解リテラシーの平均点× R^2 値の変化

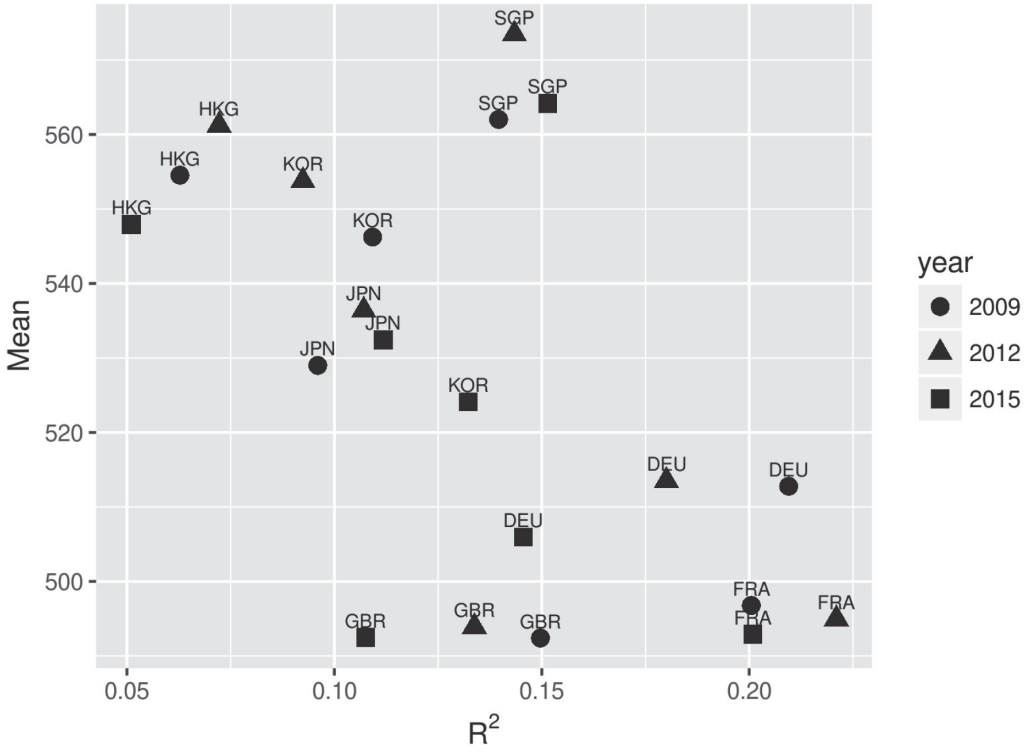

図 2. 数学リテラシーの平均点× R^2 値の変化

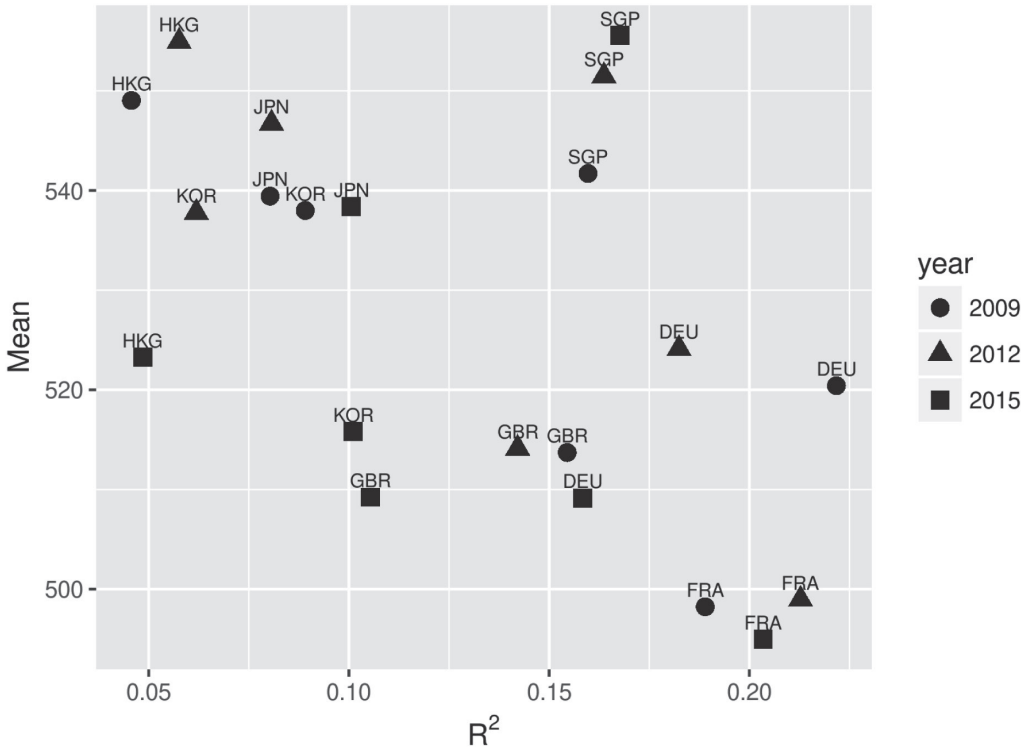

図 3. 科学リテラシーの平均点× R^2 値の変化

図 1 から図 3 を見ると，次のようなことがわかる。第一に，平均点の高低を示す縦軸に注目すると，日本・韓国・香港・シンガポールというアジアの国・地域は，イギリス・ドイツ・フランスというヨーロッパの国よりも上にあり，概して平均点が高い。これは，2009 年から 2015 年まで一貫して変わらない。特にシンガポールは一貫して日本より高い成績を修めている国である。

第二に，横軸（学力格差の深刻さ）に注目すると，香港・日本・韓国は，他の国に比べると，学力格差の深刻さはやや弱いということである。特に香港は，ESCS が学力を説明する割合が低い。シンガポールは平均点こそ高いが，学力格差は深刻である。こうした傾向も，2009 年から 2015 年まで変わらない。

第三に，ドイツ・イギリスは，平均点こそ低いものの，学力格差の深刻さを 2009 年から 2015 年まで一貫して縮小することに成功している国である。特にイギリスは，平均点こそ日本・韓国に及ばないものの，学力格差の深刻度は同程度にまで改善している。

以上の結果から，各国の教育を比較する際の注目点として，次のようなことが言えるだろう。まず，日本の教育を考える上で，注目すべきは，シンガポールと香港である。前者は，日本より成績が高く，後者は日本より学力格差の深刻度が低い。その要因について，制度・文化の両面から明らかにする必要がある。次に，韓国については，日本とほぼ似たような位置にある国であり，両者の類似性・相違点について，検証すると興味深い知見が得られるだろう。ドイツ・イギリスの学力格差の縮小も興味深い。これらの国は日本と大きく文脈が異なるから，取り組みをそのまま参考にできるとは思えないが，2009 年から 2015 年の両国の教育施策等が注目に値すると言えるだろう。最後に，フランスについては，一貫して平均点が低く，学力格差も深刻という状況が続いている。なぜフランスで大きな改善が見られないのかという点を考察するのも面白いだろう。

5. まとめ

本稿では，無償で利用できる R を利用することで，PISA を分析する方法を示してきた。現在，R を利用することで，インターネットに接続する環境さえあれば，誰でも PISA を分析し，その結果を利用すること

ができる。4 節で示したように，こうした分析は，日本や各国・地域の学力実態について，興味深い仮説をいくつも提供してくれる。計量研究を主体としないにしても，こうした数量的な情報を把握しておくことは，研究を進める上で，有益な指針を示してくれるだろう。

R は，日本の教育研究では，まだそれほど一般的に利用されているわけではない。しかし，各種の解説書が増え，有用なコードが公開されるに連れ，その利便性は確実に高まっている。本稿が，日本の教育研究・学力研究における R の普及に僅かでも貢献できれば幸いである。

＜注＞

(1) Item Response Theory（項目反応理論）に関しては，教育心理学での注目が高まる中，日本語の書籍が相次いで刊行されている。たとえば，光永（2017）を参照。

(2) 再現性（Reproducible Research）という観点から R に注目した書籍としては，たとえば，Gandrud（2015）がある。

(3) python を利用したものとして，https://sakura-education.com/myblog/archives/1016，ruby を利用したものとして，https://oku.edu.mie-u.ac.jp/~okumura/stat/pisa.html がある。本稿では，python を利用している。

(4) PISA2015 では，PVs の数が 5 から 10 に変更された。また，各変数の名称も PISA2000 から PISA2012 とは異なる命名規則に基づいている。なお，intsvy を利用する場合，PISA2015 については，pisa2015.mean.pv や，pisa2015.mean 等の関数を使う必要がある。また，intsvy は，TIMSS（https://timssandpirls.bc.edu/）や PIAAC（https://www.oecd.org/skills/piaac/）を分析することも可能である。

(5) 科学研究費補助事業・基盤研究 A「学力格差の実態把握と改善・克服に関する臨床教育社会学的研究（代表：志水宏吉）」の調査の一環である。

(6) データは，PISA 2015 Database（https://www.oecd.org/pisa/data/2015database/）からダウンロードできる。

＜参考文献＞

Chang, W, 2012, R Graphics Cookbook, O'Reilly（＝石井弓美子他訳，2013，『R グラフィックスクックブック』オライリージャパン）.

古田和久，2016，「学業的自己概念の形成におけるジェンダーと学校環境の影響」『教育学研究』83（1），pp.13-25.

Gandrud, C., 2015, Reproducible Research with R and R Studio, Second Edition, CRC Press.

川口俊明，2012，「PISA 調査の設計および分析方法について」『福岡教育大学紀要』61，pp.1-14.

Matsuoka, R., 2013, Tracking effect on tenth grade students' self-learning hours in Japan.『理論と方法』28（1），pp.87-106.

光永悠彦，2017，『テストは何を測るのか―項目反応理論の考え方』ナカニシヤ出版。

OECD, 2009, PISA Data Analysis Manual: SPSS and SAS, Second Edition, OECD.

奥村晴彦，2016，『R で楽しむ統計』共立出版。

杉野勇，2017，『入門・社会統計学―2 ステップで基礎から【R で】学ぶ―』法律文化社。

Wickham, H, 2014, "Advanced R", Routledge（＝石田基広ほか訳，2016，『R 言語徹底解説』共立出版）.