

日本人口学会 報告書

人口学研究における R の活用

日本人口学会研究企画委員会 (2022～2023 年度) 編

本報告書は、2023 年 3 月 22 日（水）に zoom で開催された、研究集会「人口学研究における R の活用」の内容をまとめたものである。

この研究集会は、2022～2023 年度日本人口学会研究企画委員会が、人口学研究における R の活用の活性化を目指し、会員の人口学研究の支援に資することを目的として開催したものであり、48 名（報告者を含む）の参加があった。

本報告書は、研究集会の講演者である山内昌和会員（早稲田大学）、余田翔平会員（国立社会保障・人口問題研究所）、野村侑平会員（早稲田大学・院）の協力を得て、2022～2023 年度日本人口学会研究企画委員会（石井太委員長（慶應義塾大学）、中澤港委員（神戸大学）、堀口侑幹事（慶應義塾大学・院））が編集を行った。

本報告書を通じて、人口学会内において R の活用が進み、会員間での情報共有を通じて、研究活動がより一層進展することを願うものである。

日本人口学会研究企画委員会
(2022～2023 年度)

目次

1	研究集会「人口学研究における R の活用」について	5
2	R と Rstudio の簡単な使い方	9
3	R を使った人口学教育	19
4	R によるカテゴリカルデータ分析：多重クロス表	27
5	R による空間的自己相関の分析事例：統計量の算出とその地図化	71

1 研究集会「人口学研究における R の活用」について

2022～2023 年度日本人口学会研究企画委員会は、人口学研究における R の活用の活性化を目指し、会員の人口学研究の支援に資することを目的とし、2023 年 3 月 22 日（水）に研究集会「人口学研究における R の活用」を zoom により開催した。

当日のプログラムは以下の通りである。

1. 石井太 (慶應義塾大学) 「R と Rstudio の簡単な使い方について」
2. 中澤港 (神戸大学) 「R を使った人口学教育」
3. 余田翔平 (国立社会保障・人口問題研究所) 「R によるカテゴリーカルデータ分析：多重クロス集計表」
4. 野村侑平 (早稲田大学・院)・山内昌和 (早稲田大学) 「R による空間的自己相関の分析事例：統計量の算出とその地図化」

石井太会員 (慶應義塾大学) による第 1 報告「R と Rstudio の簡単な使い方について」では、R に馴染みのない会員を想定し、R とは何か、Rstudio での実行方法に加え、R の基礎的概念についてコードの実行をしながら解説が行われた。

中澤港 (神戸大学) 会員による第 2 報告「R を使った人口学教育」では、Newell(1988) をテキストとし、これに最近までの日本のデータを補足的に提供することによって講義を行う例について解説が行われた。具体的な R コードなどもサポートサイトで提供され、出生、死亡、人口移動、結婚と離婚、再生産、人口モデルという人口学方法論における幅広いトピックについて、R を用いた効率的な教育が可能であることが示された。

余田翔平会員 (国立社会保障・人口問題研究所) による第 3 報告「R によるカテゴリーカルデータ分析：多重クロス集計表」では、学歴同類婚に関する人口学的分析について、夫婦の学歴組み合わせを示すクロス集計表分析を例として解説が行われた。報告では、CSV ファイルの入出力、周辺度数の付加や比率の算出など二元クロス集計表の処理や多元クロス集計表への拡張、Data Visualization など R の特徴を活かした利用例が示された。

野村侑平会員 (早稲田大学・院)・山内昌和会員 (早稲田大学) による第 4 報告「R による空間的自己相関の分析事例：統計量の算出とその地図化」では、R の地理学・人口学への応用として、空間統計量の算出とその地図化を例とし、e-stat からの国勢調査のデータ入手・読み込み、コロプレスマップの作成、統計量の算出とその地図化について、R を用いて実行をしながら解説がなされた。

当日の研究集会には報告者を含めて 48 名の会員が参加し、報告者の講演とともに、活発な質疑が行われた。研究集会終了後、参加者に対して以下の 3 種類の質問に関するアンケートを行った。

1. 今回の研究集会は全体としていかがでしたか。
2. それぞれの講演は、研究や教育に役立ちましたか (報告ごと)。
3. このような研究集会がまたあったら参加したいと思いますか。

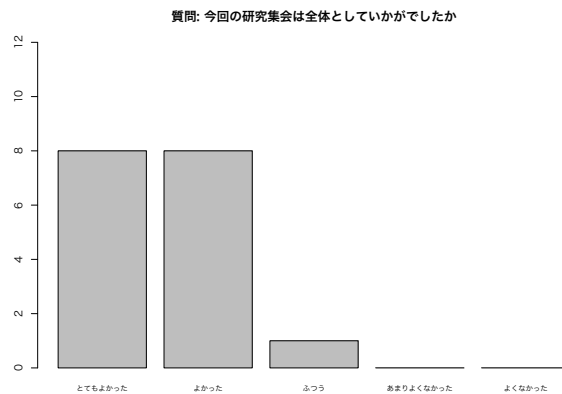


図1 アンケート集計結果 (Q1)

図1は、「今回の研究集会は全体としていかがでしたか。」との質問に対する回答を示したもので、「とてもよかった」「よかった」を合わせると9割以上となった。

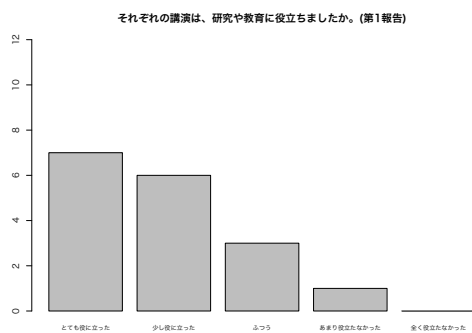


図2 アンケート集計結果 (Q2, 第1報告)

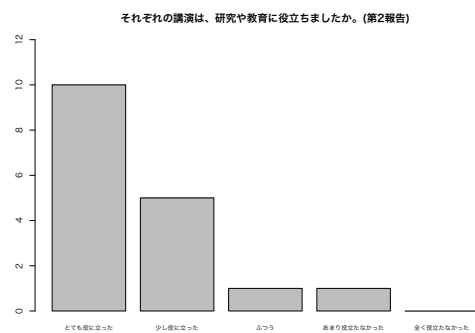


図3 アンケート集計結果 (Q2, 第2報告)

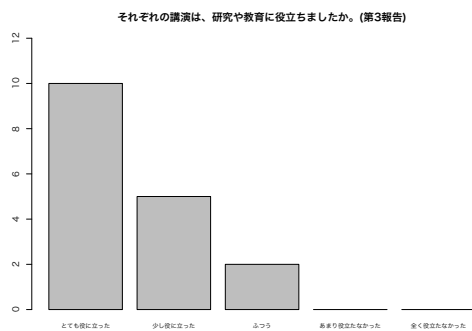


図4 アンケート集計結果 (Q2, 第3報告)

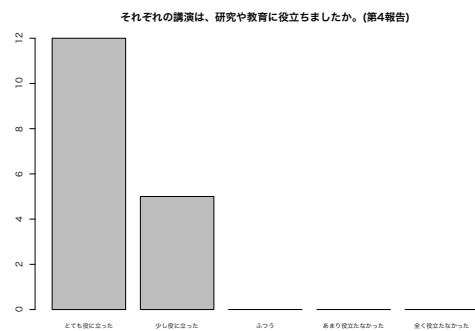


図5 アンケート集計結果 (Q2, 第4報告)

図2～5は、各報告ごとに「それぞれの講演は、研究や教育に役立ちましたか。」との質問に対する回答を示したものである。Rに馴染みのない会員を想定した第1報告は、他の報告に対して「とても役に立った」「少し役に立った」という回答が少なく、より具体的

な問題を取り上げた第 2～4 報告が参加者にとってより有用であったことが理解される。

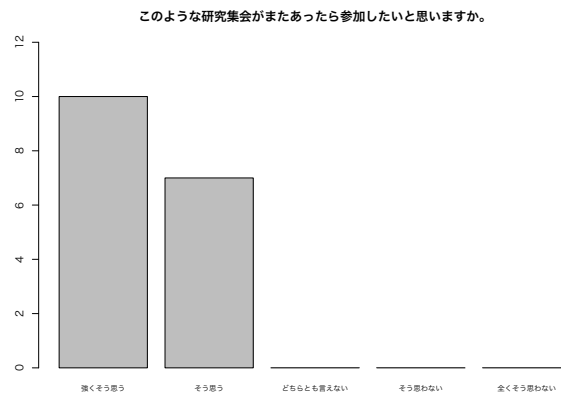


図 6 アンケート集計結果 (Q3)

図 6 は、「このような研究集会がまたあったら参加したいと思いますか。」との質問に対する回答を示したもので、全ての回答が「強くそう思う」または「そう思う」であった。

次節以降は、各講演者が研究集会で報告に使用した資料を収載したものである。なお、研究集会の特設サイトを以下の URL で公開しており、報告資料や R プロジェクトファイルが入手可能となっているので、利用されたい (期間限定)。

<http://ishii-futoshi.sakura.ne.jp/paojr2023/index.html>

参考文献

Newell, C. (1988) *Methods and Models in Demography*, The Guilford Press.

2 R と Rstudio の簡単な使い方

石井太（慶應義塾大学）

R と Rstudio の簡単な使い方

石井 太

2023 年 3 月 22 日

研究集会「人口学研究における R の活用」(zoom 開催)
日本人口学会研究企画委員会
本資料の無断複製・配布を固く禁じます

1 / 17

R について

1 R について

- R とは
- 人口学での R の活用
- Rstudio での実行方法
- パッケージの利用方法
- tidyverse

2 R の基礎的な概念

- R の基礎的な概念

R とは

- R は R project(<http://www.r-project.org/index.html>) による統計計算とグラフィックスのための言語・環境。
- Rjpwiki(<http://www.okada.jp/RWiki/>) にある R project の "What is R?" の日本語訳によれば以下の通り。
 - R は統計計算とグラフィックスのための言語・環境で、GNU プロジェクトの一つであり、AT&T(当時) のベル研究所で John M. Chambers と同僚により開発された S 言語・環境に似ています。R は S を別個に実装したものと考えられます。両者の仕様及び実装には幾つかの重要な違いがありますが、S 用にかかれたコードの多くは変更無しで R でも実行できます。
 - R は多様な統計手法 (線形・非線形モデル・古典的統計検定・時系列解析・判別分析・クラスターリング・その他) とグラフィックスを提供し、広汎な拡張が可能です。S は高頻度で統計的手法による研究の手段の選択肢となりますが、R を使えばそうした活動にオープンソースの道から参加できるようになります。
 - R の強みの一つは、巧くデザインされた出版物並みのプロットを容易に作成できる点であり、これには必要なら数学記号や式を含めることもできます。グラフィックスにおける細かなデザインの標準設定に、これまで周到な配慮が払われてきていますが、ユーザーが完全に制御することもできます。
 - R は Free Software Foundation の GNU General Public License の条項のもとで、ソースコードの形で入手できます。様々な UNIX(類似のシステムの BSD や Linux を含む)・Windows・MacOS では、ダウンロードすればそのままコンパイル・使用できます。

3 / 17

(私の考える)R の特徴

- R は **プログラミング言語** である。C と比べると、ベクトルを基礎としていることと、apply 関数等を含む豊富な関数の利用により、**見通しよく効率的にコードが書ける**のが利点。
- インタプリタなので、コンパイル言語より **実行は遅い**。ただ、**通常の使用では問題にならないレベル** (どうしても気になるなら、Rcpp パッケージによる C との組み合わせや、並列処理なども可能)。
- R は **フリーソフト** である。フリーソフトには保証がない (自己責任) が、ユーザーコミュニティによって膨大なパッケージが開発・提供され、CRAN から入手できる。また、これらは日々機能強化やバグフィックスを含む更新が続けられている。近年統計分野では、新たな手法を開発し、R で実装して提供することもなされるようになってきている。
- **グラフィックス** 機能が豊富で、EDA (探索的な分析) や解析結果のプレゼンテーションにも効果的である。
- 近年、Hadley Wickham を中心に **tidyverse** という、データ解析を指向したパッケージ群が開発されている。

国内外の Demographer による R の利用

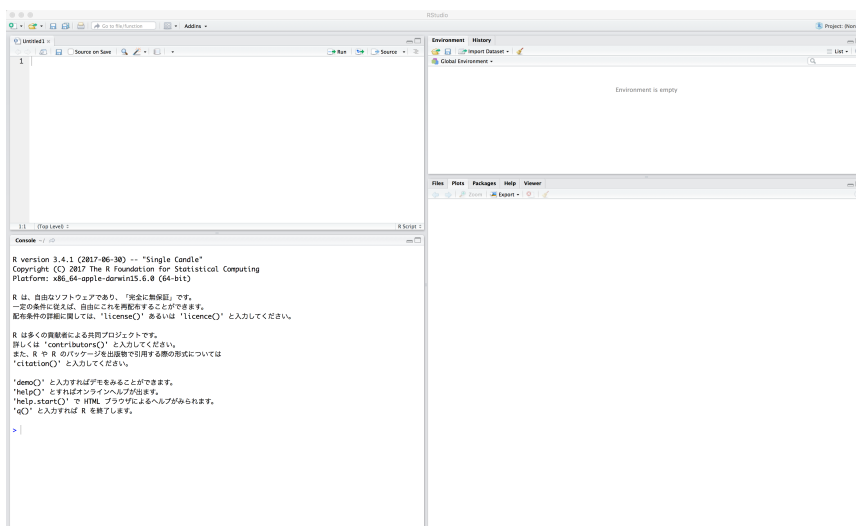
- 国連の World Population Prospects では R を用いて確率推計が行われており、bayesTFR(出生率推計), bayesLife(平均寿命推計), MortCast(死亡率推計), bayesPop(人口推計) などのパッケージとして公開されている。
- 神戸大学大学院保健学研究科中澤港先生は、日本の人口学分野での R 活用のパイオニアで、「R による人口分析入門」などの R に関する著書や、人口ピラミッド作成などのパッケージなど R に関するマテリアルの提供などをなさっている（過去の絶版となった著書その他は[中澤先生のウェブサイト](#)から入手可能）。



5 / 17

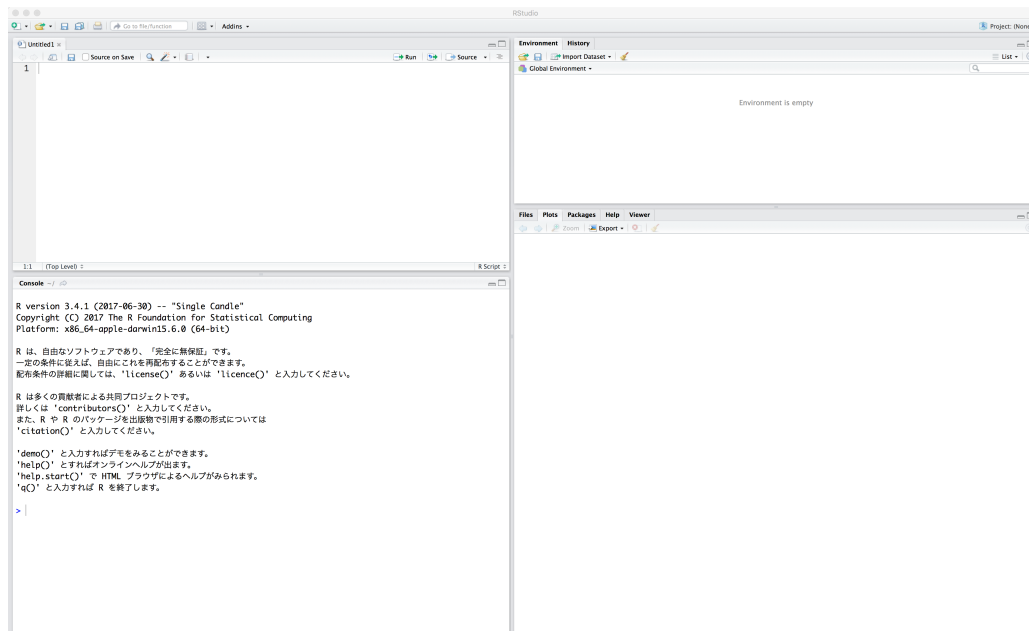
Rstudio IDE (Integrated Development Environment)

- Rstudio IDE の画面は以下の通り（空の R script でエディタ画面を出している）。デフォルトでは、左上がエディタ、左下がコンソール、右上が変数と履歴、右下がプロット・パッケージ・ヘルプなどとなっている。これらは好きな位置にカスタマイズ可能。なお、Rstudio IDE の機能については、Rstudio のホームページにあるチートシートが参考になる。
- コンソールは、R に命令を伝えたり、R からの答えを受け取る窓口で、直接式を入力することで、計算機のように使用できる。



Rstudio のエディタ

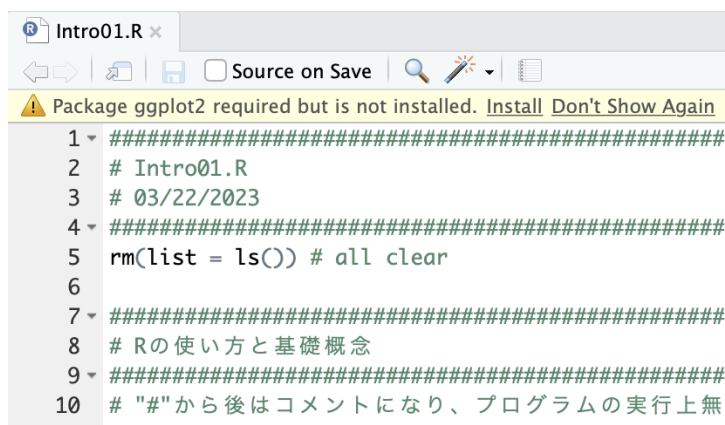
- 通常は、エディタを起動し、ここからコンソールに命令を送って実行。
- ソースコードの記述あるいはファイルの読み込みを行い、エディタでコンソールに送る部分を選択し、command-Return(Windows では cntrl-Return) を押下するとコンソールに送られる。



7 / 17

パッケージ

- R では、関数やデータをパッケージという形で取りまとめたものが CRAN を通じて配布されている。ユーザーは、必要なパッケージをインストールすることにより、そこに含まれている関数やデータを利用することが可能となる。
- Rstudio では、"Packages" というタブでパッケージのインストールや管理を行うことができる。インストールについては、"Packages" タブの "Install" ボタンにより可能である。また、install.package 関数でも可能である。
- また、Rstudio では、インストールされていないパッケージを利用する R コードを開くと、パッケージのインストールについて尋ねてくるので、ここで、"Install" をクリックしてもよい。
- インストールされたパッケージを利用するときには、library 関数でロードして使用する。



tidyverse

- ggplot2 や dplyr の開発者である Hadley Wickham を中心に tidyverse というパッケージ群が開発されている。
- これらのパッケージ群は背景となる思想と共通のインターフェイスを共有しており、プログラミングよりもデータ解析を指向したパッケージとなっている。
- (私見では)tidyverse 関係は更新が早く、コードが動かなくなるケースがしばしばあること、データ解析には有効であっても必ずしも万能ではなく、base と適宜組み合わせながら利用するのがよい (と思う)。

Tidyverse

[Packages](#) [Blog](#) [Learn](#) [Help](#) [Contribute](#)


R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

9 / 17

R の基礎的な概念

1 R について

- R とは
- 人口学での R の活用
- Rstudio での実行方法
- パッケージの利用方法
- tidyverse

2 R の基礎的な概念

- R の基礎的な概念

R の基礎的な概念

- CRAN から入手できる R のマニュアルの中で、基礎的な解説を行っているのが、"An Introduction to R"。(古いバージョンの日本語訳あり)。
- 最も基本的なオブジェクトが「ベクトル」。数値ベクトルは四則演算などの演算が可能。
- 何かの手続きをまとめたものを「関数」として呼び出すことができる。(→ 実は R におけるすべての操作は関数。)
- 色々なオブジェクト: 行列 (matrix)、配列 (array)、因子 (factor)、リスト (list)、データフレーム (data frame)、関数 (function) など。
- 制御文 (if else, for, while など) がある。

11 / 17

ベクトル

- ベクトル: 最も基礎的なオブジェクト
- "c": ベクトルを作る関数
`x <- c(1, 2, 3)`
- "<-" 付置を表す演算子
- ベクトルは算術演算子や関数などによりまとめて演算することが可能
- "seq": 数列ベクトルを作成する関数
`seq(始点, 終点, 間隔)` で数列を生成するのが基本。例えば `seq(0.1, 2.4, 0.1)` は 0.1 から 2.4 ま
で 0.1 間隔の数列を生成。

ベクトルの要素の取り出し

- 添字による取り出し
 - `x[2]` : 2 番目の要素
 - `x[c(2, 4)]` : 2 番目と 4 番目の要素
 - `x[-2]` : 2 番目以外の要素
- 論理変数による取り出し
 - `x[c(TRUE, FALSE, F, T)]`
 - `x[x > 3]`
- 名前による取り出し
 - `names(x) <- c("Spring", "Summer", "Fall", "Winter")`
 - `x["Summer"]`

13 / 17

行列と配列

- 行列とは、要素を二次元に並べたもの。
- 行と列と呼ばれる二つの添え字で要素の位置を表す。
- 行列はベクトルに次元を属性として与えることで生成できる。通常は `matrix` 関数で生成。
`matrix(ベクトル, nrow = 行数, ncol = 列数)`
- 行列を拡張して多次元に並べたものを配列と呼ぶ。`array` 関数で生成。
`array(ベクトル, dim = c(1 次元目の次元数, 2 次元目の次元数, 3 次元目の次元数, ... ))`

リスト

- ベクトルは同じ種類の要素しか格納できない。違う種類の要素を格納するときはリストを用いる。
- リストへの要素は二重鉤括弧 `[[ ]]` でアクセスする。
- 名前が与えられている場合は、`$` 演算子でもアクセスできる。

15 / 17

データフレーム

- データフレームは行列の見かけを持つ、R で広く用いられる便利なデータ構造（実はリスト）。例えば、5 人の者に年齢・国籍・収入を調査したとした場合、これを以下のようなデータフレームとして格納できる。

	<i>Age</i>	<i>Country</i>	<i>Income</i>
1	35	<i>JPN</i>	40
2	42	<i>USA</i>	35
3	58	<i>USA</i>	52
4	25	<i>JPN</i>	26
5	44	<i>JPN</i>	32

R のグラフィックス

- R にはグラフィックス機能があり、データの可視化に有用。
- base では plot 関数や matplot 関数によるグラフ、また、lattice パッケージなどグリッド・グラフィックスによる描画ができる。
- tidyverse では ggplot2 パッケージによるグラフ描画が基本的。

3 Rを使った人口学教育

中澤港（神戸大学）

Rを使った人口学教育

中澤 港（神戸大学）

<minato-nakazawa@people.kobe-u.ac.jp>

国際協力 / 国際保健分野で 人口学教育が必要な理由

- 途上国における公衆衛生や開発支援の実践活動を行うことが多い
 - 活動の前に対象集団を把握することは必須
 - その地域にはどういう人が何人住んでいるのか
 - どのような年齢構成なのか
 - 出生率や死亡率はどの程度なのか
 - どの程度人口移動があるのか
 - 途上国では公式統計の質が不十分なことが多い
 - 最低限, データの質を評価できる能力が要る
 - 農村部では必要なデータがとられていないことも多いので, 自らデータを集める調査も必要
 - 小集団ではモデルを適用して補整することも必要
- 修論や博論のみならず, 卒業後も解析できるスキルが必要
- 途上国出身の留学生にはフリーソフトが適切 → GNU R を採用

教科書の選定

- 理想は自分で英語の教科書を書く
- 最適な教科書を探した
 - Preston S, et al. (2001) Demography: Measuring and Modeling Population Processes. Blackwell Publishing
 - Rowland DT (2003) Demographic methods and concepts. Oxford University Press
 - Newell C (1988) Methods and Models in Demography. The Guilford Press
- よりコンパクトで例題も載っていることと、データの入手方法と質の評価について丁寧に記述されていることから、15回の講義で完了するのに向いている Newell の本を採用

2023年3月22日

3

Newell C のテキストの欠点と改善策

- 出版年が 1988 年と古い
 - 例示されているデータも 1985 年まで
 - 最新の分析方法が含まれていない
- 最近までの日本のデータを補足的に提供
 - R のパッケージ fmsb に含め、随時アップデート
 - JASM : 1985 年以降 5 年ごとの男女年齢 5 歳階級別の死亡数及び人口
 - Jlife : 完全生命表の qx
 - Jfert : 年齢別出生率と年齢別有配偶出生率
 - Jpop, Jpopl : センサス人口
 - Jvital : 人口動態統計
 - Jvital2013byPref : 2013 年の都道府県別人口動態統計
 - S60MPJ : 1985 年(昭和 60 年)年齢 5 歳階級別基準人口

2023年3月22日

21

4

サポートサイト

<https://minato.sip21c.org/demography-special/>

- テキストに掲載されている表などのデータをエクセル形式とタブ区切りテキスト形式で提供
- テキストに掲載されている分析を実行できる R コードを提供
- 各章末の問題についても, データと R による解答コード例を提供
- テキストで紹介されている手法を日本のデータに適用するための R コードを提供
- テキストの概要をまとめた handout を提供
- 以上を公開(神戸大学の院生だけでなく誰でも利用可。英語なので調査や援助の対象地の人も)

2023年3月22日

5

毎回資料やコードをダウンロード

- 最初の2章はイントロとデータについて
 - <https://minato.sip21c.org/demography-special/demography-2022-01.pdf>
 - <https://minato.sip21c.org/demography-special/demography-2022-02.pdf>
- 3章 Age and sex structure
 - <https://minato.sip21c.org/demography-special/demography-2022-03.pdf>
 - RStudio を起動し、この講義用のフォルダを作ってプロジェクト登録し、<https://minato.sip21c.org/demography-special/code20220421.R> をそこに保存
 - 主に日本人口を使った例示
 - `if (!require(fmsb)) { install.packages("fmsb"); library(fmsb) }`
↑ コード 11 行にカーソルを置き [→RUN] をクリックして fmsb パッケージをロード
 - コード 14 行 ?Jpopl で Jpopl のヘルプ表示
 - コード 20 行から 22 行を選択し [→RUN] をクリック。Jpopl というデータフレームを部分参照する方法を説明。2015 年センサス 0 歳男性で 3 通り
 - 27 行から 42 行は人口データの誤報告の 1 つとして知られる Age heaping (年齢が切りの良い数字に偏る) を評価するための Whipple's index の計算法
 - コード 12 行にカーソルを置き [→RUN] をクリックし、pyramid パッケージをロード
 - 46 行、48 行、57(-59) 行と、オプションを増やすと人口ピラミッドが見やすくなる
 - 63 行で描画ウィンドウを開いてから 67 行、74 行で人口ピラミッドを描く

2023年3月22日

3章の続き

- 82 行から 84 行で出生性比の代わりに 0 歳人口の性比の計算例
- 90 行から 92 行で年齢各歳の性比計算
 - R の四則演算はベクトルや行列の各要素にやってくれるので式は簡単に書ける
 - 1888 年人口の 42 歳の性比が 120 を超える異常値で、おそらく丙午のせいであろうという説明をする
 - 1966 年の丙午ではそのような性比の異常がなかった原因についても3つの可能性を説明
- 従属人口指数等
 - テキストに載っている式をそのまま R コンソールに打てば計算できる(コード 105 行)にカーソルを置いて [→RUN] をクリック
 - 2015 年日本人口での計算は 113-118 行。年齢区分を変えてみる例を 119-121 行
 - 123 行が生産年齢人口割合、124 行が老年化指数
 - 125 行が年齢区分を変えた 1888 年の老年化指数
 - 126 行が介護者比(計算コードは fmsb パッケージで定義)

2023年3月22日

7

期間出生力とコホート出生力

- <https://minato.sip21c.org/demography-special/demography-2022-04.pdf>
- <https://minato.sip21c.org/demography-special/demography-2022-05.pdf>
- <https://minato.sip21c.org/demography-special/code20220428.R>
 - Table 4.1 ~ Table 4.7 をタブ区切りテキスト形式でサーバにアップロードしておき、まずすべて読みこんでおく
 - Table 4.1 からバングラデシュ 1974 年の CWR 計算
$$(x4.1\$Males000s[1]+x4.1\$Females000s[1])/sum(x4.1\$Females000s[4:9])$$
 - fmsb パッケージの Jpopl を使って、2010 年と 2015 年の日本の CWR 計算
 - Table 4.2 からバングラデシュ 1974 年の粗出生率計算
$$sum(x4.2\$Births)/sum(x4.1\$Males000s+x4.1\$Females000s)$$
 - fmsb パッケージの Jfert と Jpopl を使って 2010 年と 2015 年の日本の粗出生率計算
 - 以下、General Fertility Rate、Age Specific Fertility Rate、Total Fertility Rate と続く
 - 年齢別出生率のところはグラフ表示も
$$barplot(x4.2\$ASFRper1000)$$

$$plot(ASFRper1000 \sim AgeGroup, data=x4.2, type="l")$$
- <https://minato.sip21c.org/demography-special/asfrworld.R>
- <https://minato.sip21c.org/demography-special/code20210513.R>

2023年3月22日

23

8

死亡指標と生命表

- <https://minato.sip21c.org/demography-special/demography-2022-06.pdf>
- <https://minato.sip21c.org/demography-special/code20210520.R>
 - スウェーデンとカザフスタンの女性の粗死亡率、年齢別死亡率、年齢調整死亡率の比較
- <https://minato.sip21c.org/demography-special/code-chap6.R>
 - 生命表を順次説明
 - fmsb パッケージには、lifetable()、lifetable2()、lifetable3() という関数を実装。年齢別死亡率 mx、年齢階級幅、階級内の死亡分布係数 ax を与えると生命表がデータフレームとして作れるようにした (RStudio ではデータフレーム名をクリックすると表として見ることができる)

	ages	n	ax	mx	qx	px	lx	dx	Lx	Tx	ex
1	0	1	0.1	0.008314	0.0082522517	0.9917477	100000.00	825.22517	99257.3	7756160.8	77.56160
2	1	4	0.4	0.000408	0.0016304035	0.9983696	99174.77	161.69490	396311.0	7656903.5	77.20615
3	5	5	0.5	0.000181	0.0009045907	0.9990954	99013.08	89.56631	494841.5	7260592.4	73.32962
4	10	5	0.5	0.000187	0.0009345631	0.9990654	98923.51	92.45026	494386.4	6765751.0	68.39375
5	15	5	0.5	0.000282	0.0014090067	0.9985910	98831.06	139.25363	493807.2	6271364.5	63.45535
6	20	5	0.5	0.000307	0.0015338228	0.9984662	98691.81	151.37575	493080.6	5777557.3	58.54140
7	25	5	0.5	0.000364	0.0018183453	0.9981817	98540.43	179.18054	492254.2	5284476.7	53.62745
8	30	5	0.5	0.000566	0.0028260012	0.9971740	98361.25	277.96902	491111.3	4792222.5	48.72063
9	35	5	0.5	0.000884	0.0044102533	0.9955897	98083.28	432.57213	489335.0	4301111.2	43.85162
10	40	5	0.5	0.001445	0.0071989936	0.9928010	97650.71	702.98686	486496.1	3811776.2	39.03480
11	45	5	0.5	0.002485	0.0123482863	0.9876517	96947.73	1197.13827	481745.8	3325280.1	34.29972
12	50	5	0.5	0.004210	0.0208307563	0.9791692	95750.59	1994.55715	473766.5	2843534.3	29.69730
13	55	5	0.5	0.007219	0.0354551237	0.9645449	93756.03	3324.13164	460469.8	2369767.7	25.27585
14	60	5	0.5	0.012054	0.0585068947	0.9414931	90431.90	5290.88956	438932.3	1909297.9	21.11310
15	65	5	0.5	0.018259	0.0873095379	0.9126905	85141.01	7433.62213	407121.0	1470365.7	17.26977
16	70	5	0.5	0.029920	0.1391886863	0.8608113	77707.39	10815.98907	361497.0	1063244.7	13.68267
17	75	5	0.5	0.049689	0.2209927305	0.7790073	66891.40	14782.51261	297500.7	701747.7	10.49085

2023年3月22日

9

人口移動

- <https://minato.sip21c.org/demography-special/demography-2022-07.pdf>
- この章はテキストの内容が少ないので他のテキストや日本のデータなどで補足
- <https://minato.sip21c.org/demography-special/inmig-outmig-2013.R>
- <https://minato.sip21c.org/demography-special/compare-pyramids-among-pr ef.R>
- <https://minato.sip21c.org/demography-special/relations-PEI-YP.R>
- <https://minato.sip21c.org/demography-special/PEImap2.R>
- <https://minato.sip21c.org/demography-special/migration-expectancy.R>
 - 生命表と同様な考え方で、migration expectancy を計算するのは Rowland のテキストの Table 11 から

2023年3月22日

結婚と離婚

- <https://minato.sip21c.org/demography-special/demography-2022-08.pdf>
 - 図解により SMAM の考え方を理解してもらう
- <https://minato.sip21c.org/demography-special/Chapter8.R>
- <https://minato.sip21c.org/demography-special/Chapter8E.R>
- <https://minato.sip21c.org/demography-special/SMAM.R>
 - 日本のデータで SMAM を計算するコード

2023年3月22日

11

出生と死亡を組み合わせ:再生産

- <https://minato.sip21c.org/demography-special/demography-2022-09.pdf>
 - 何通りかの方法による NRR の理解
- <https://minato.sip21c.org/demography-special/Chapter9.R>
- <https://minato.sip21c.org/demography-special/Chapter9E.R>
- <https://minato.sip21c.org/demography-special/Chapter9A.R>
 - 日本のデータで補足

2023年3月22日

25

12

人口モデル

- <https://minato.sip21c.org/demography-special/demography-2022-10.pdf>
 - 安定人口モデル、平均世代時間
- <https://minato.sip21c.org/demography-special/Chap11.R>
- <https://minato.sip21c.org/demography-special/Chap11E.R>
- <https://minato.sip21c.org/demography-special/demography-2022-11.pdf>
 - Newell のテキストは Chapter 12 と 13 がモデル生命表だが、経験的モデル、リレーショナルモデル、パラメタライズドモデルをまとめて説明
- <https://minato.sip21c.org/demography-special/Chap12.R>
- <https://minato.sip21c.org/demography-special/Chap13.R>
- <https://minato.sip21c.org/demography-special/demography-2022-12.pdf>
 - モデル結婚とモデル出生 (Coale and Trussell、Coale and McNeil など)
- <https://minato.sip21c.org/demography-special/Chap14.R>
- <https://minato.sip21c.org/demography-special/demography-2022-14.pdf>
 - トレンド延伸、時系列解析、コホート要因法による人口予測
- <https://minato.sip21c.org/demography-special/Chap15.R>
- マイクロシミュレーション
 - <https://minato.sip21c.org/demography-special/demography-2022-15.pdf>

2023年3月22日

13

考察と将来構想

- このようにして教育を受けた学生は、国際協力や国際保健の現場において、実際にさまざまな人口分析ができるはず
 - ただし,
 - 受講者が毎年 10 人程度
 - 実際にどの程度活用できるかは未知
- 英語での講義の様子を動画で保存して公開すれば、Coursera や edX のような e-learning のコンテンツになりうる。が、コロナ禍でもリアルタイムオンライン講義にしたので、動画コンテンツはできていない
- テキスト自体が、GNU FDL や Creative Commons のようなフリーライセンスではない点もネック

2023年3月22日

26

14

4 Rによるカテゴリカルデータ分析：多重クロス表

余田翔平（国立社会保障・人口問題研究所）

人口学研究におけるRの活用

Rによるカテゴリカルデータ分析：多重クロス表

余田翔平（国立社会保障・人口問題研究所）

2023-03-22

このサイトについて

このサイトは日本人口学会による研究集会「人口学研究におけるRの活用」（2023年3月22日）の報告資料および演習用ページです。研究集会は学会員限定ですが、本サイトはどなたでも閲覧可能です。研究集会後も本資料は修正・更新される可能性があります。

Chapter 1 はじめに

1.1 本報告の目的

- R の配列 (array) の処理を多重クロス集計表 (contingency table) の分析に適用する
- 基本統計量 (行／列パーセント) の算出とその可視化

1.2 人口学的分析への適用例：学歴同類婚

- Who marries whom?
 - 社会の開放性 (社会学)
 - 世帯間の経済格差 (経済学)
 - 出生の近接要因としての結婚 (人口学)
- 学歴同類婚の方法論については Lichter and Qian (2019) に詳しい
 1. 比率・相関
 2. オッズ比
 3. 対数線形モデル
 4. Forces of Attraction
 5. regression based approaches (e.g., イベントヒストリーモデル)
- 本報告では1に依拠し、夫婦の学歴組み合わせを示すクロス集計表を分析
- 日本の学歴同類婚の最近の動向については Uchikoshi and Raymo (2021) を参照

Reference

- Lichter, Daniel T., and Zhenchao Qian. 2019. “The Study of Assortative Mating: Theory, Data, and Analysis.” In *Analytical Family Demography*, edited by Robert Schoen, 303–37. The Springer Series on Demographic Methods and Population Analysis. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-93227-9_13.
- Uchikoshi, Fumiya, and James M. Raymo. 2021. *Educational Assortative Mating in Japan: Insights into Social Change and Stratification*. 1st ed. 2021版. Springer. <https://link.springer.com/book/10.1007/978-981-16-3713-1>.

Chapter 2 データの読み込み

本報告では Fukuda, Yoda, and Mogi (2021) のSupplemental Tableとして掲載されているデータを用いる。これは「国勢調査」の調査票情報を用いて、妻が30-39歳の夫婦の学歴組み合わせを示すクロス集計表である。例えば、以下は1980年の「国勢調査」にもとづく夫婦の学歴組み合わせである。

##	EduH				
##	EduW	JHS	HS	VC/JC	UNI
##	JHS	1825957.2	692306.05	25961.477	43269.13
##	HS	744229.0	3106723.41	164422.687	856728.74
##	VC/JC	34615.3	207691.82	69230.605	493268.06
##	UNI	0.0	25961.48	8653.826	337499.20

表側には妻の学歴（EduW）、表頭には夫の学歴（EduH）がそれぞれ置かれている。ここに調査年（CensusYear）を追加すると、妻の学歴×夫の学歴×調査年の3変数からなる多元クロス集計表を作成することができる。

本報告では、上のようなクロス集計表のデータが事前に与えられている状況を想定して進めていく。具体的には、「国勢調査」の調査年別に $4 \times 4 = 16$ セルの度数がCSVファイルなどで与えられているものとする。そのため、以下ではCSVファイルの出入力の基本操作を押さえておこう。

2.1 CSVファイルの読み込み

まず、本報告で用いるCSVファイルは `CensusCSV/CSV_EduWH` フォルダに保存されている。ここには1980年、1990年、2000年、2010年の「国勢調査」を用いて集計された夫婦の学歴組み合わせのデータファイルが調査年ごとに保存されている。例えば、以下は1980年調査のデータファイルである¹。

	A	B	C	D
1	1825957.212	692306.0521	25961.47695	43269.12826
2	744229.006	3106723.409	164422.6874	856728.7395
3	34615.30261	207691.8156	69230.60521	493268.0621
4	0	25961.47695	8653.825651	337499.2004

夫婦の学歴組み合わせ：1980年

CSVファイルの読み込みには `read.csv()` を用いる。この関数に最低限必要な引数は、読み込みの対象となるCSVファイルのパス（`file`）である。CSVファイルがワーキングディレクトリと同じ階層に保存されている場合はファイル名を文字列で引き渡せばよい。CSVファイルが異なる階層に保存されている場合は、相対パスあるいはフルパスでファイルの場所を指定する。その際、報告者個人が推奨する方法は、CSVファイルが保存されているフォルダのパスをオブジェクトと保存しておき、そのオブジェクトとファイル名の文字列とを `paste()` あるいは `paste0()` で結合する方法である²。このようにしておく、CSVファイルの保存場所が変更された場合でも、そのフォルダパスを示すオブジェクトのみを修正すればそれ以外のコードを修正する必要がないためである。

また、今回の例には当てはまらないが、1行目の変数名となっている場合は、`header = TRUE` とすると（むしろ、こちらがデフォルトの設定）1行目は変数名として処理され、2行目から`observation`として認識される。

```
CensusDir <- "CensusCSV/CSV_EduWH/"
FN_EduWH1980csv <- paste0(CensusDir, "EduWH1980.csv")

EduWH1980_raw <- read.csv(file = FN_EduWH1980csv, header = FALSE) # "file =
```

`read.csv()` でデータを読み込んだ直後のオブジェクトの型はデータフレームになっている。今回raw dataとして読み込んだのは行列の形をしたクロス集計表であるため、以下のように行列に変換する必要がある。

CSVファイルを読み込んだ直後の型はデータフレームになっている

```
class(EduWH1980_raw)
```

```
## [1] "data.frame"
```

as.matrix() でオブジェクトの型を行列に変換

```
EduWH1980_mat <- as.matrix(EduWH1980_raw)
```

```
class(EduWH1980_mat)
```

```
## [1] "matrix" "array"
```

```
EduWH1980_mat
```

```
##           V1           V2           V3           V4
## [1,] 1825957.2  692306.05  25961.477  43269.13
## [2,]  744229.0 3106723.41 164422.687 856728.74
## [3,]   34615.3  207691.82  69230.605 493268.06
## [4,]         0.0   25961.48   8653.826 337499.20
```

こうしてデータの型をデータフレームから行列に変換できたものの、この段階では各行および各列がそれぞれどのカテゴリを示すのかが明瞭ではない。そこで、`dimnames()` を用いて行列の次元にラベルをつけておくと便利である。`dimnames()` はオブジェクトの各次元のラベルをリスト（`list`）として返す関数である。

```
EduWH1980 <- EduWH1980_mat
class(dimnames(EduWH1980))
```

```
## [1] "list"
```

```
dimnames(EduWH1980)
```

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

```
## NULL
```

```
##
```

```
## [[2]]
```

```
## [1] "V1" "V2" "V3" "V4"
```

列 (column) のラベル (“V1”-“V4”) のみが保存されており、行ラベルは空であることがわかる。夫婦の学歴カテゴリを示すラベルを付与するためには以下のようにリストとして代入する。

```
dimnames(EduWH1980) <- list(c("JHS", "HS", "VC/JC", "UNI"),
                             c("JHS", "HS", "VC/JC", "UNI"))
```

```
EduWH1980
```

```
##           JHS           HS           VC/JC           UNI
## JHS    1825957.2  692306.05  25961.477  43269.13
## HS      744229.0 3106723.41 164422.687 856728.74
## VC/JC   34615.3  207691.82  69230.605 493268.06
## UNI           0.0   25961.48   8653.826 337499.20
```

行カテゴリ・列カテゴリが付与されたが、さらに `names()` と `dimnames()` とを組み合わせることで行方向・列方向それぞれの変数名も付与できる。

```
names(dimnames(EduWH1980)) <- c("EduW", "EduH")
```

```
# 完成!
```

```
EduWH1980
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
## JHS    1825957.2  692306.05  25961.477  43269.13
## HS      744229.0 3106723.41 164422.687 856728.74
## VC/JC   34615.3  207691.82  69230.605 493268.06
## UNI           0.0   25961.48   8653.826 337499.20
```

2.2 CSVファイルの出力

つづいて、R の上で作成したオブジェクトを外部ファイルに出力する方法について見ておこう。オブジェクトはさまざまなフォーマットで出力可能であるが、ここではCSVファイルの出力方法について説明する。

2.2.1 出力するオブジェクトの準備

前節ではすでに調査年ごとのクロス集計表がCSVファイルの形式で用意されていることを前提に、CSVファイルを読み込む方法を見てきた。しかしながら、Fukuda, Yoda, and Mogi (2021) で公表されているのは、以下のように、調査年ごとの夫婦学歴組み合わせのクロス集計表の**全体度数**と**全体パーセント**のみである。

Supplemental Table 1 Distributions of educational matching between wife and husband from 1980 to 2010 in Japan *

1980 (N = 8,636,518)					
wife \ husband	JHS	HS	JC/VS	UNI	Total
JHS	21.1	8.0	0.3	0.5	29.9
HS	8.6	35.9	1.9	9.9	56.3
JC/VS	0.4	2.4	0.8	5.7	9.3
UNI	0.0	0.3	0.1	3.9	4.4
Total	30.2	46.7	3.1	20.0	100.0

1990 (N = 6,785,605)					
wife \ husband	JHS	HS	JC/VS	UNI	Total
JHS	6.4	3.6	0.2	0.3	10.5
HS	7.0	36.8	2.8	10.5	57.1
JC/VS	0.9	7.0	2.4	12.1	22.4
UNI	0.1	0.9	0.3	8.8	10.1
Total	14.4	48.3	5.7	31.6	100.0

2000 (N = 5,712,160)					
wife \ husband	JHS	HS	JC/VS	UNI	Total
JHS	1.6	2.0	0.2	0.3	4.1
HS	3.9	33.3	4.0	10.6	51.8
JC/VS	0.8	9.5	4.8	16.2	31.4
UNI	0.1	1.3	0.6	10.7	12.7
Total	6.4	46.1	9.6	37.9	100.0

2010 (N = 4,815,036)					
wife \ husband	JHS	HS	JC/VS	UNI	Total
JHS	1.0	1.5	0.2	0.3	3.0
HS	2.8	23.6	4.4	7.5	38.2
JC/VS	1.1	12.2	8.1	16.4	37.9
UNI	0.2	2.8	1.8	16.1	20.9
Total	5.0	40.1	14.6	40.2	100.0

*: The subject cases are Japanese couples in which the wife is aged 30-39 in each year.

Source: The Population Census of Japan in each period.

Fukuda et al.(2021)

そこで、これらの情報に基づいてクロス集計表のセル度数を算出し、それらをCSVファイルに出力してみよう。例として2010年のクロス集計表を取り上げる。まず、以下のように全体パーセントを入力したデータをCSVファイルとして保存しておく。

	A	B	C	D
1	1.0	1.5	0.2	0.3
2	2.8	23.6	4.4	7.5
3	1.1	12.2	8.1	16.4
4	0.2	2.8	1.8	16.1

2010年のクロス集計表（全体パーセント）：TotProp2010.csv

つぎに、このCSVファイルを R に取り込んでおく。

パラメータの設定

```
SupTabsDir <- "CensusCSV/SupplementalTables/" # 全体パーセントのCSVファイルが保存
OutDir      <- "CensusCSV/CSV_EduWH/"         # 出力先ディレクトリ
NofObs      <- 481536                          # 全体度数
```

全体パーセントの読み込み

```
TotProp2010_raw <- read.csv(paste0(SupTabsDir, "TotProp2010.csv"), header =
TotProp2010 <- as.matrix(TotProp2010_raw)
TotProp2010
```

```
##          V1    V2   V3    V4
## [1,] 1.0    1.5  0.2    0.3
## [2,] 2.8   23.6  4.4    7.5
## [3,] 1.1   12.2  8.1   16.4
## [4,] 0.2    2.8  1.8   16.1
```

クロス集計表の全体パーセントを行列として読み込めたら、あとはそれらを比率に変換した上で全体度数を乗じればセル度数が得られるので、

```
# セル度数の算出
```

```
EduWH2010 <- NofObs * (TotProp2010 / 100)
EduWH2010
```

```
##           V1           V2           V3           V4
## [1,]  4815.360   7223.04   963.072  1444.608
## [2,] 13483.008 113642.50 21187.584 36115.200
## [3,]  5296.896  58747.39 39004.416 78971.904
## [4,]   963.072  13483.01  8667.648 77527.296
```

とすれば、2010年の夫婦の学歴組み合わせのクロス集計表が得られる。

2.2.2 write.csv()

CSVファイルを出力するための関数のひとつが `write.csv()` である。第1の引数として保存するオブジェクト名を指定し、`file` には保存するファイル名を（パスも含めて）指定する。`row.names = FALSE` を指定しないと、第1列目に行ラベル（ない場合は行番号）が表示される。

```
write.csv(EduWH2010, file = paste0(OutDir, "EduWH2010_writecsv.csv"), row.names = FALSE)
```

保存されたCSVファイルを開いてみると、第1行目に列名（変数名）が表示されていることがわかる。このままでも必要な情報（セル度数）は保存されているので大きな問題ではないが、列名も表示させたくない場合は `write.table()` を使うと良い。

	A	B	C	D
1	V1	V2	V3	V4
2	48150.36	72225.54	9630.072	14445.108
3	134821.008	1136348.496	211861.584	361127.7
4	52965.396	587434.392	390017.916	789665.904
5	9630.072	134821.008	86670.648	775220.796

`write.csv()` で出力されたCSV

2.2.3 write.table()

`write.table()` では `col.names` という引数を指定でき、これを `FALSE` にすると列名が表示されなくなる。この関数を用いてCSVファイルを出力する際にはセパレータ `sep` をカンマにしておく。

```
write.table(EduWH2010, file = paste0(OutDir, "EduWH2010.csv"), sep = ",", row.names = FALSE)
```

保存されたCSVファイルを確認すると、列名が保存されていないことがわかる。

	A	B	C	D
1	48150.36	72225.54	9630.072	14445.108
2	134821.008	1136348.496	211861.584	361127.7
3	52965.396	587434.392	390017.916	789665.904
4	9630.072	134821.008	86670.648	775220.796

`write.table()` で出力されたCSV

Reference

Fukuda, Setsuya, Shohei Yoda, and Ryohei Mogi. 2021. “Educational Assortative Mating in Japan: Evidence from the 1980 Census.” *The Journal of Population Studies* 57: 1–20.
<https://doi.org/10.24454/jps.2101001>.

1. Fukuda, Yoda, and Mogi (2021) の付表として公開されているのはクロス集計表の全体度数と全体パーセントのみであるため、論文内の分析を完全に再現することができない（4行1列目のセル度数（ n_{41} ）が0になっているのはそのためである）。クロス集計表を用いた分析の再現性の担保のためには、クロス集計表の各セルの**度数**を公表すべきである。↩
2. `paste()` や `paste0()` はともに文字列を連結させる関数である。両者の違いは、文字列を結合させる際のセパレーター（`sep`）のデフォルトがカンマ（`paste()`）か空白（`paste0()`）かである。↩

Chapter 3 二元クロス集計表の処理

3.1 周辺度数の追加

すでに見た夫婦の学歴組み合わせのクロス集計表には行方向・列方向ともに周辺度数 (marginal frequency) がついていない。これは `xtabs()` や `table()` などの関数を用いて 個票データからクロス集計表を作成した場合も同様である。クロス集計表に周辺度数を追加するためには `addmargins()` を用いる。

```
# 行周辺度数
```

```
addmargins(EduWH1980, margin = 1)
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
## JHS    1825957.2  692306.05  25961.477  43269.13
## HS      744229.0 3106723.41 164422.687 856728.74
## VC/JC   34615.3  207691.82  69230.605 493268.06
## UNI           0.0   25961.48   8653.826 337499.20
## Sum    2604801.5 4032682.75 268268.595 1730765.13
```

```
# 列周辺度数
```

```
addmargins(EduWH1980, margin = 2)
```

```
##          EduH
## EduW      JHS      HS      VC/JC      UNI      Sum
##   JHS  1825957.2  692306.05  25961.477  43269.13 2587493.9
##   HS    744229.0 3106723.41 164422.687 856728.74 4872103.8
##  VC/JC   34615.3  207691.82  69230.605 493268.06  804805.8
##   UNI         0.0   25961.48   8653.826 337499.20  372114.5
```

```
# 行／列周辺度数
```

```
addmargins(EduWH1980, margin = c(1,2))
```

```
##          EduH
## EduW      JHS      HS      VC/JC      UNI      Sum
##   JHS  1825957.2  692306.05  25961.477  43269.13 2587493.9
##   HS    744229.0 3106723.41 164422.687 856728.74 4872103.8
##  VC/JC   34615.3  207691.82  69230.605 493268.06  804805.8
##   UNI         0.0   25961.48   8653.826 337499.20  372114.5
##   Sum  2604801.5 4032682.75 268268.595 1730765.13 8636518.0
```

3.2 比率（proportion）の算出

クロス集計表の分析においておそらく最も頻繁に行うのが比率（proportion）の計算である。Rにはこれを可能にする数多くのパッケージおよび関数が存在するが、以下では base に組み込まれている `proportions()`、および RcmdrMisc パッケージの `rowPercents()`・`colPercents()` を紹介する。

3.2.1 `proportions()`

この関数は `margin` という引数を持ち、1は行方向、2は列方向にそれぞれ算出した「比率」を返す。

```
proportions(EduWH1980, margin = 1) * 100 # 行パーセント
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
##   JHS    70.568562 26.755853 1.003344  1.672241
##   HS     15.275311 63.765542 3.374778 17.584369
##   VC/JC   4.301075 25.806452 8.602151 61.290323
##   UNI     0.000000  6.976744 2.325581 90.697674
```

```
proportions(EduWH1980, margin = 2) * 100 # 列パーセント
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
##   JHS    70.099668 17.1673820  9.677419  2.5
##   HS     28.571429 77.0386266 61.290323 49.5
##   VC/JC   1.328904  5.1502146 25.806452 28.5
##   UNI     0.000000  0.6437768  3.225806 19.5
```

有効桁数を減らしたい場合は"round"で丸めるとよい

```
round(proportions(EduWH1980, margin = 1) * 100, digits = 1) # "digits = "は全
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
##   JHS    70.6 26.8  1.0  1.7
##   HS     15.3 63.8  3.4 17.6
##   VC/JC   4.3 25.8  8.6 61.3
##   UNI     0.0  7.0  2.3 90.7
```

tableの次元に名前がついている場合、marginには次元名を指定することもできる

```
proportions(EduWH1980, margin = "EduW")
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
## JHS  0.70568562 0.26755853 0.01003344 0.01672241
## HS   0.15275311 0.63765542 0.03374778 0.17584369
## VC/JC 0.04301075 0.25806452 0.08602151 0.61290323
## UNI  0.00000000 0.06976744 0.02325581 0.90697674
```

```
proportions(EduWH1980, margin = "EduH")
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
## JHS  0.70099668 0.171673820 0.09677419 0.025
## HS   0.28571429 0.770386266 0.61290323 0.495
## VC/JC 0.01328904 0.051502146 0.25806452 0.285
## UNI  0.00000000 0.006437768 0.03225806 0.195
```

周辺度数をつけてから proportions() に引き渡せば行全体あるいは列全体の変数の分布を算出できる。

```
EduWH1980_wRMar <- addmargins(EduWH1980, margin = 1) # w/ row margins
round(proportions(EduWH1980_wRMar, margin = 1) * 100, digits = 1)
```

```
##           EduH
## EduW      JHS   HS VC/JC  UNI
##   JHS    70.6 26.8   1.0  1.7
##   HS     15.3 63.8   3.4 17.6
##   VC/JC   4.3 25.8   8.6 61.3
##   UNI     0.0  7.0   2.3 90.7
##   Sum    30.2 46.7   3.1 20.0
```

```
EduWH1980_wCMar <- addmargins(EduWH1980, margin = 2) # w/ column margins
round(proportions(EduWH1980_wCMar, margin = 2) * 100, digits = 1)
```

```
##           EduH
## EduW      JHS   HS VC/JC  UNI  Sum
##   JHS    70.1 17.2   9.7  2.5 30.0
##   HS     28.6 77.0  61.3 49.5 56.4
##   VC/JC   1.3  5.2  25.8 28.5  9.3
##   UNI     0.0  0.6   3.2 19.5  4.3
```

3.2.2 rowPercents() ・ colPercents()

RcmdrMisc パッケージの `rowPercents()` や `colPercents()` を使っても行パーセントや列パーセントを算出できる。 `proportions()` と比較した際のこれらの関数の便利な点は、デフォルトで周辺度数を出力結果に返してくれることである。

```
library(RcmdrMisc)
```

```
rowPercents(EduWH1980, digits = 1)
```

```
##           EduH
## EduW      JHS   HS VC/JC  UNI Total Count
##   JHS    70.6 26.8   1.0  1.7 100.1 2587493.9
##   HS     15.3 63.8   3.4 17.6 100.1 4872103.8
##   VC/JC   4.3 25.8   8.6 61.3 100.0  804805.8
##   UNI     0.0  7.0   2.3 90.7 100.0  372114.5
```

```
colPercents(EduWH1980, digits = 1)
```

```
##           EduH
## EduW      JHS      HS      VC/JC      UNI
##   JHS      70.1      17.2      9.7      2.5
##   HS       28.6      77.0     61.3     49.5
##   VC/JC     1.3      5.2     25.8     28.5
##   UNI       0.0      0.6      3.2     19.5
##   Total    100.0     100.0     100.0    100.0
##   Count 2604801.5 4032682.8 268268.6 1730765.1
```

Chapter 4 多元クロス集計表の処理

つぎに、変数が3つ以上の多元クロス集計表の処理について見ていく。例えば、夫婦の学歴組み合わせのクロス集計表が複数の調査年について得られている状況を想定してみよう。この場合、

妻の学歴 × 夫の学歴 × 調査年

の3変数からなる多元クロス集計表が得られることになる。こうした多元クロス集計表を R の中で表現・処理する方法はいくつかあるが、以下では配列（array）を用いた方法について紹介する。

4.1 配列（array）

配列（array）とは行列（matrix）を三次元以上に一般化したものである。言い換えると、二次元の配列は行列として処理することができる。R で配列を作成する方法は主に2つある。

- 関数 `array()` を用いて配列を直接作成する
- 複数の行列を `abind()` で結合する

以下、順に見ていこう。まずは関数 `array()` の使用方法についてである。

```
# 2×2×2の3次元からなり、要素がすべてNAの空の配列  
array(data = NA, dim = c(2,2,2))
```

```
## , , 1
##
##      [,1] [,2]
## [1,]   NA   NA
## [2,]   NA   NA
##
## , , 2
##
##      [,1] [,2]
## [1,]   NA   NA
## [2,]   NA   NA
```

任意のベクトルを引き渡してもよい。列方向に順番に格納されていく。

```
array(data = 1:8, dim = c(2,2,2))
```

```
## , , 1
##
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
##
## , , 2
##
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
```

データとして引き渡したベクトルの長さが次元の長さの積と等しくない場合、配列が埋まるまでデー

```
array(data = 1:3, dim = c(2,2,2))
```

```
## , , 1
##
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    1
##
## , , 2
##
##      [,1] [,2]
## [1,]    2    1
## [2,]    3    2
```

ただし、報告者の個人的な経験からすると、`array()` を用いて分析対象とするクロス集計表を直接作成することはあまりない。むしろ、`array()` が活躍するのは以下のような場面である。

- 既存の配列の次元の長さを増やす
- 仮想的な配列のデータを作成して、自身が作成した関数が意図した通りの挙動で動くかを確認する

4.2 `abind()` を用いた配列の作成

クロス集計表がraw dataとして与えられている場合、それらをもとに多元クロス集計表を表す配列を作成する方法は、複数のクロス集計表を行列として読み込み、それらを結合することである。その際に用いる関数が `abind()` である。以下、この関数の使い方を見ていこう。

はじめに問題になるのは複数のクロス集計表を読み込む方法である。最も愚直な方法は、CSVファイルを読み込むためのコードを調査年ごとに書くことである。

```

EduWH1980_raw <- read.csv(paste0(CensusDir, "EduWH1980.csv"), header = FALSE)
EduWH1990_raw <- read.csv(paste0(CensusDir, "EduWH1990.csv"), header = FALSE)

EduWH1980 <- as.matrix(EduWH1980_raw)
EduWH1990 <- as.matrix(EduWH1990_raw)

dimnames(EduWH1980) <-
dimnames(EduWH1990) <- list(c("JHS", "HS", "VC/JC", "UNI"),
                             c("JHS", "HS", "VC/JC", "UNI"))

names(dimnames(EduWH1980)) <-
names(dimnames(EduWH1990)) <- c("EduW", "EduH")

```

読み込むデータファイルが数個程度であればこの方法でも大きな問題は生じないが、歴史の長い反復横断調査やパネル調査ほど読み込むべきデータファイルの数は増えていく。その場合、調査回ごとにデータ読み込みのプログラムを書くのではなく、データ読み込みのための自前の関数を作ってしまうと便利である。例えば、以下では `readEduWH()` という関数を定義している。

```

readEduWH <- function(path, census_year){
  TargetCSV <- paste0(path, "EduWH", as.character(census_year), ".csv") # 1
  EduWH_raw <- read.csv(TargetCSV, header = FALSE)
  EduWH_mat <- as.matrix(EduWH_raw)
  EduWH <- EduWH_mat

  dimnames(EduWH) <- list(c("JHS", "HS", "VC/JC", "UNI"),
                          c("JHS", "HS", "VC/JC", "UNI"))

  return(EduWH)
}

```

この関数は引数として `census_year` を持ち、ここに読み込みたい国勢調査の調査年を引き渡すと、その調査年の夫婦の学歴組み合わせのCSVファイルが読み込まれて（ただし、そのためにはCSVのファイル名に調査年を含めておく必要がある）返り値として出力される。例えば、1980年の国勢調査データを読み込むためには、

```
readEduWH(path = CensusDir, census_year = 1980)
```

```
##              JHS              HS              VC/JC              UNI
## JHS    1825957.2  692306.05  25961.477  43269.13
## HS      744229.0 3106723.41 164422.687 856728.74
## VC/JC   34615.3 207691.82  69230.605 493268.06
## UNI           0.0  25961.48   8653.826 337499.20
```

とすればよい。この自作関数と `abind()` とを組み合わせることで、以下のように1980年から2010年までの4回の国勢調査データをひとつの配列に格納することができる。

```
library(abind)
```

```
CensusYearList <- seq(1980, 2010, 10) # 国勢調査の調査年が格納されたベクトル
```

```
EduWH <- NULL # 空のオブジェクトを作っておく
```

```
for(iYear in CensusYearList){
```

```
  EduWH_temp <- readEduWH(path = CensusDir, census_year = iYear) # iYear年の
```

```
  EduWH <- abind(EduWH, EduWH_temp, along = 3) # その結果を事前に作っておいたEdu
```

```
}
```

```
dimnames(EduWH)[[3]] <- CensusYearList # abindで結合した次元にはラベルがついてい
```

```
names(dimnames(EduWH)) <- c("EduW", "EduH", "CensusYear") # 各次元の変数名を付
```

オブジェクト `EduWH` に4回の国勢調査における夫婦の学歴組み合わせのクロス集計表が保存されているか確認してみよう。

EduWH

```
## , , CensusYear = 1980
```

```
##
```

```
##      EduH
```

```
## EduW      JHS      HS      VC/JC      UNI
```

```
## JHS  1825957.2  692306.05  25961.477  43269.13
```

```
## HS    744229.0  3106723.41  164422.687  856728.74
```

```
## VC/JC  34615.3  207691.82  69230.605  493268.06
```

```
## UNI      0.0    25961.48   8653.826  337499.20
```

```
##
```

```
## , , CensusYear = 1990
```

```
##
```

```
##      EduH
```

```
## EduW      JHS      HS      VC/JC      UNI
```

```
## JHS  434278.720  244281.78  13571.21  20356.81
```

```
## HS   474992.350  2497102.64  189996.94  712488.53
```

```
## VC/JC  61070.445  474992.35  162854.52  821058.20
```

```
## UNI   6785.605   61070.44  13571.21  597133.24
```

```
##
```

```
## , , CensusYear = 2000
```

```
##
```

```
##      EduH
```

```
## EduW      JHS      HS      VC/JC      UNI
```

```
## JHS   91394.56  114243.20  11424.32  17136.48
```

```
## HS   222774.24  1902149.28  228486.40  605488.96
```

```
## VC/JC  45697.28  542655.20  274183.68  925369.92
```

```
## UNI    5712.16   74258.08  34272.96  611201.12
```

```
##
```

```
## , , CensusYear = 2010
```

```
##
```

```
##      EduH
```

```
## EduW          JHS          HS          VC/JC          UNI
##   JHS    48150.360    72225.54    9630.072    14445.11
##   HS     134821.008   1136348.50   211861.584   361127.70
##   VC/JC   52965.396   587434.39   390017.916   789665.90
##   UNI      9630.072   134821.01    86670.648   775220.80
```

このように、妻の学歴（ EduW ）×夫の学歴（ EduH ）×調査年（ CensusYear ）からなる三元クロス表が配列として表現できたことがわかる。

4.3 apply() を用いた配列の処理

apply() は、行列あるいは配列の特定の次元ごとに何らかの処理を並列的に施すための関数である。例えば、夫婦の学歴組み合わせの分布（すなわち全体パーセント）を調査年ごとに算出するためには、

```
apply(EduWH, MARGIN = 3, FUN = proportions, simplify = FALSE) # "MARGIN =",
```

```
## $`1980`
##           EduH
## EduW           JHS           HS           VC/JC           UNI
##   JHS    0.211422846 0.080160321 0.003006012 0.00501002
##   HS     0.086172345 0.359719439 0.019038076 0.09919840
##   VC/JC  0.004008016 0.024048096 0.008016032 0.05711423
##   UNI    0.000000000 0.003006012 0.001002004 0.03907816
##
## $`1990`
##           EduH
## EduW           JHS           HS VC/JC           UNI
##   JHS    0.064 0.036 0.002 0.003
##   HS     0.070 0.368 0.028 0.105
##   VC/JC  0.009 0.070 0.024 0.121
##   UNI    0.001 0.009 0.002 0.088
##
## $`2000`
##           EduH
## EduW           JHS           HS           VC/JC           UNI
##   JHS    0.016016016 0.02002002 0.002002002 0.003003003
##   HS     0.039039039 0.33333333 0.040040040 0.106106106
##   VC/JC  0.008008008 0.09509510 0.048048048 0.162162162
##   UNI    0.001001001 0.01301301 0.006006006 0.107107107
##
## $`2010`
##           EduH
## EduW           JHS           HS VC/JC           UNI
##   JHS    0.010 0.015 0.002 0.003
##   HS     0.028 0.236 0.044 0.075
##   VC/JC  0.011 0.122 0.081 0.164
##   UNI    0.002 0.028 0.018 0.161
```

とすればよい。なお、配列の次元にラベルをつけている場合、そのラベルを用いて引数 `MARGIN` に引き渡すこともできる。

```
apply(EduWH, "CensusYear", proportions, simplify = FALSE)
```

```
## $`1980`
##           EduH
## EduW           JHS           HS           VC/JC           UNI
##   JHS    0.211422846 0.080160321 0.003006012 0.00501002
##   HS     0.086172345 0.359719439 0.019038076 0.09919840
##   VC/JC 0.004008016 0.024048096 0.008016032 0.05711423
##   UNI   0.000000000 0.003006012 0.001002004 0.03907816
##
## $`1990`
##           EduH
## EduW           JHS           HS VC/JC           UNI
##   JHS    0.064 0.036 0.002 0.003
##   HS     0.070 0.368 0.028 0.105
##   VC/JC 0.009 0.070 0.024 0.121
##   UNI   0.001 0.009 0.002 0.088
##
## $`2000`
##           EduH
## EduW           JHS           HS           VC/JC           UNI
##   JHS    0.016016016 0.02002002 0.002002002 0.003003003
##   HS     0.039039039 0.33333333 0.040040040 0.106106106
##   VC/JC 0.008008008 0.09509510 0.048048048 0.162162162
##   UNI   0.001001001 0.01301301 0.006006006 0.107107107
##
## $`2010`
##           EduH
## EduW           JHS           HS VC/JC           UNI
##   JHS    0.010 0.015 0.002 0.003
##   HS     0.028 0.236 0.044 0.075
##   VC/JC 0.011 0.122 0.081 0.164
##   UNI   0.002 0.028 0.018 0.161
```

上の出力結果を見ると調査年の前に \$ がついているが、これは返り値がリストになっていることを示している。\$ の後に続く文字列がリストの要素の名前になっているのでこれを用いて要素にアクセスできる。

```
TotProp <- apply(EduWH, 3, proportions, simplify = FALSE)
class(TotProp)
```

```
## [1] "list"
```

```
TotProp$`1980`
```

```
##           EduH
## EduW           JHS           HS           VC/JC           UNI
## JHS    0.211422846 0.080160321 0.003006012 0.00501002
## HS     0.086172345 0.359719439 0.019038076 0.09919840
## VC/JC  0.004008016 0.024048096 0.008016032 0.05711423
## UNI    0.000000000 0.003006012 0.001002004 0.03907816
```

4.4 apply() と自作関数を組み合わせる

最後に、同類婚、すなわち夫婦の学歴が同じカテゴリの組み合わせ、の割合を調査年ごとに算出してみよう。一例として、これは以下の3ステップに分けて考えればよい。

1. 任意の調査年の夫婦の学歴組み合わせのクロス表に対して、対角セルを取り出す
2. 対角セルの各度数を全体度数で割る
3. 1-2を全ての調査年について繰り返す

まず、例として1980年のクロス表を対象にすると、1および2は以下のコードで実行できる。

```
ObsDiag <- diag(EduWH1980) # diag(): 行列の対角成分を取得
PropDiag <- ObsDiag / sum(EduWH1980)
```

問題はこうした処理を全ての調査年ごとに繰り返す（ステップ3）にはどうすればよいか、である。今回の例では非常に単純な処理なので、同じコードを調査年ごとに書き起こしても大きな問題はなさそうである。しかしながら、プログラミングの有名な原則「DRY（Don't Repeat Yourself）原則」にも言われているように、できるだけできるだけコードの重複は避ける習慣はつけておいた方がよい（自戒）。そこで、上記のステップ1-2の処理を行う関数を作成してみよう。

```
prop_diag <- function(x){
  ObsDiag <- diag(x)
  PropDiag <- ObsDiag / sum(x)
  return(PropDiag)
}
```

これにより、引数として夫婦の学歴組み合わせからなる行列を引き渡すと、対角成分の比率が返り値として出力される関数が定義された。この自作関数と `apply()` を組み合わせることで、調査年ごとに同類婚の占める割合を算出することができる。

```
PropDiag_byCensus <- apply(EduWH, 3, prop_diag)
PropDiag_byCensus
```

```
##          CensusYear
## EduW          1980  1990          2000  2010
##  JHS    0.211422846 0.064 0.01601602 0.010
##  HS     0.359719439 0.368 0.33333333 0.236
##  VC/JC  0.008016032 0.024 0.04804805 0.081
##  UNI    0.039078156 0.088 0.10710711 0.161
```

この場合、オプションの `simplify = FALSE` を外しておく、返り値が行列となり、その後の処理（出力結果を可視化する、CSVに書き出すなど）も容易になる。出力結果を行列ではなくてリストで得たい場合は、 `simplify = FALSE` をつけておくとよい。

Chapter 5 Data Visualization

R には極めて多彩なグラフの描画機能が搭載されている。そのため、R 上で行なった処理結果を例えばExcelなどにコピペしてグラフを作成するのではなく、ぜひ R の内部でグラフの描画まで完結させることを強く推奨する。昨今は `tidyverse` パッケージの一部である `ggplot()` を用いる方法が R コミュニティの中では優勢であるが、以下では `base` 環境の中で伝統的に用いられてきた関数に基づいて説明する。

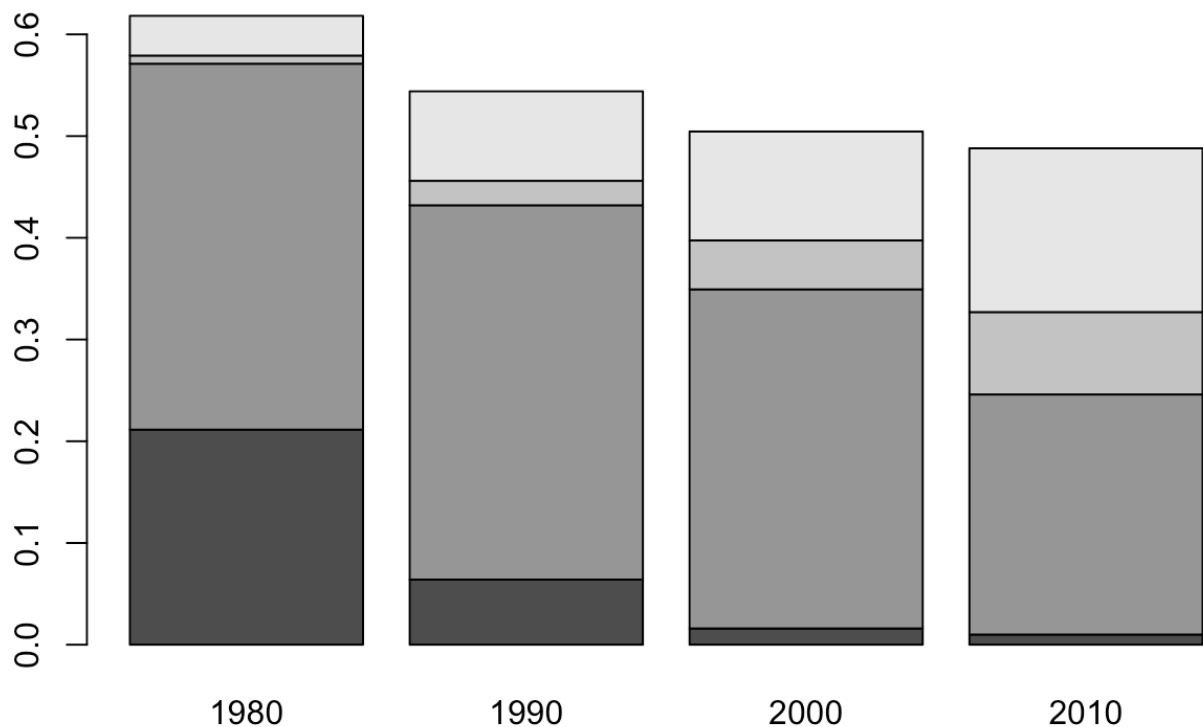
5.1 棒グラフ

棒グラフは `barplot()` で作成できる。先に作成した、調査年ごとの同類婚の比率が格納された `PropDiag_byCensus` を引数として渡すと、

```
PropDiag_byCensus
```

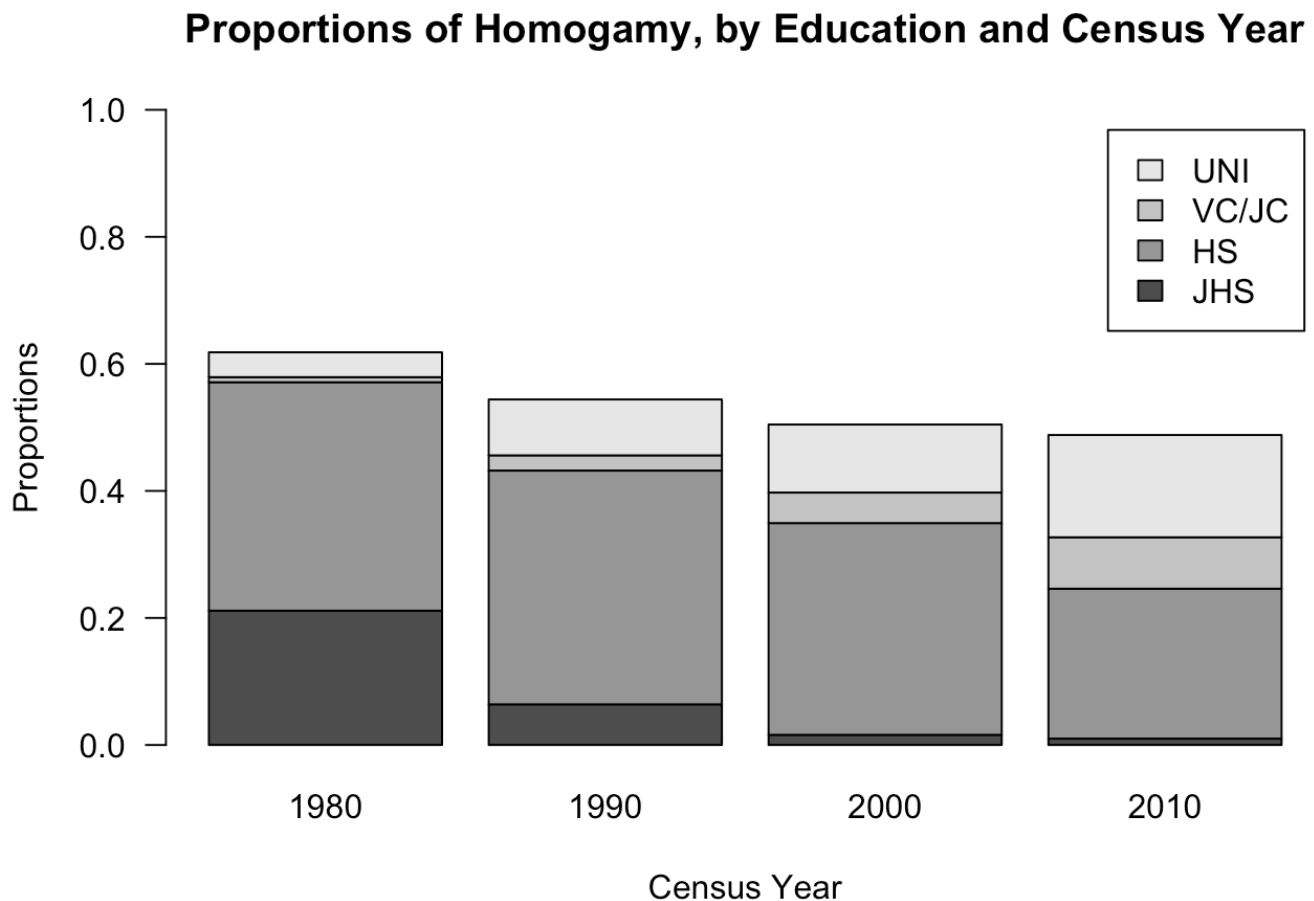
```
##           CensusYear
## EduW           1980  1990           2000  2010
##   JHS    0.211422846 0.064 0.01601602 0.010
##   HS     0.359719439 0.368 0.33333333 0.236
## VC/JC 0.008016032 0.024 0.04804805 0.081
##   UNI   0.039078156 0.088 0.10710711 0.161
```

```
barplot(PropDiag_byCensus)
```



のように、デフォルトの設定では積み上げ棒グラフが描画される。このままでは論文にそのまま掲載できるような水準の体裁になっていないが、例えば以下のようにコードを書き加えれば（幾分）見栄えは向上する。

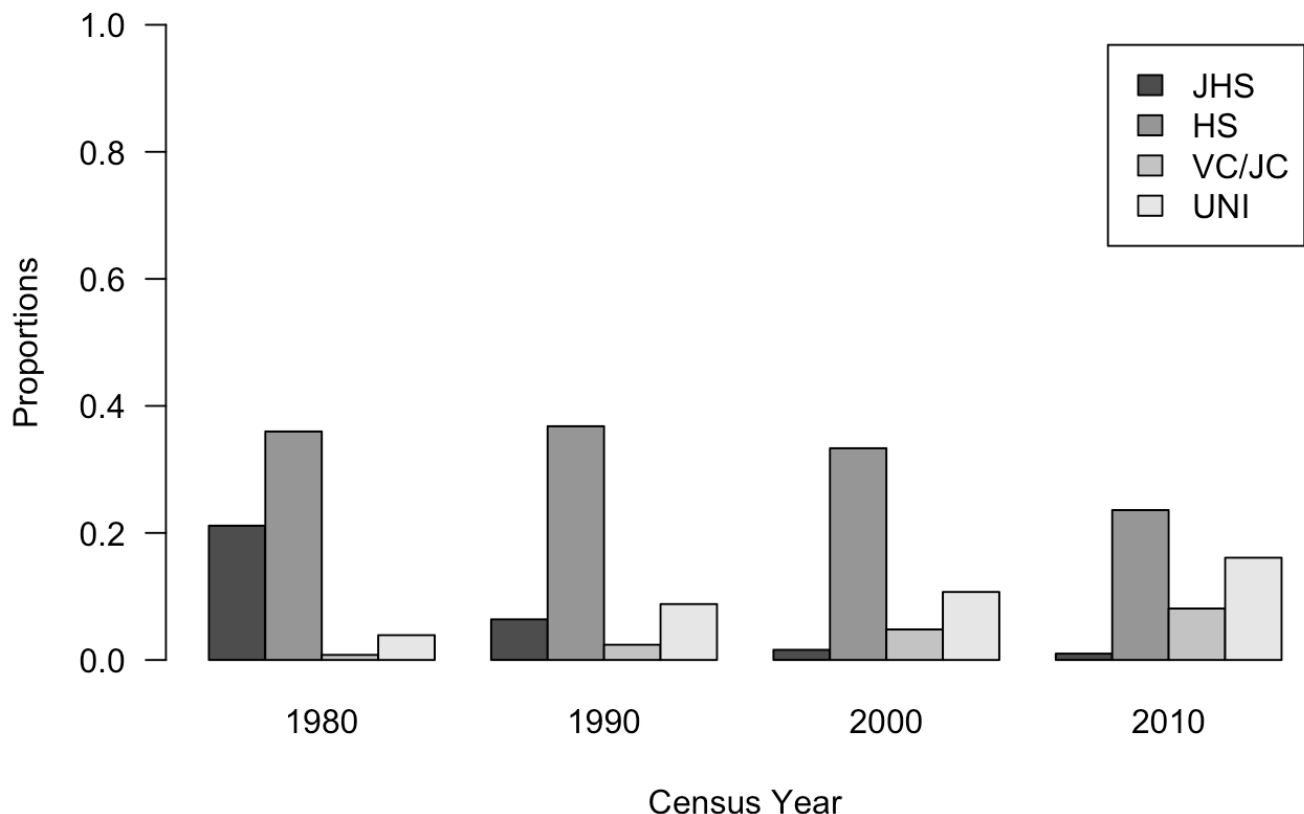
```
barplot(PropDiag_byCensus,
        ylim = c(0, 1),
        axes = FALSE,
        legend = rownames(PropDiag_byCensus))
axis(2, las = 1)
mtext("Census Year", side = 1, line = 3)
mtext("Proportions", side = 2, line = 3)
title("Proportions of Homogamy, by Education and Census Year")
```



積み上げ棒グラフではなく、調査年・学歴別に棒グラフを描きたい場合は、`beside = TRUE` をオプションに加えばよい。

```
barplot(PropDiag_byCensus, beside = TRUE,
        ylim = c(0, 1),
        axes = FALSE,
        legend = rownames(PropDiag_byCensus))
axis(2, las = 1)
mtext("Census Year", side = 1, line = 3)
mtext("Proportions", side = 2, line = 3)
title("Proportions of Homogamy, by Education and Census Year")
```

Proportions of Homogamy, by Education and Census Year



5.2 折れ線グラフ

折れ線グラフは `plot()` あるいは `matplot()` を用いて描画できる。 `plot()` に最低限必要な引数はx軸とy軸に対応するベクトルである。例として、調査年ごとに同類婚が占める割合を折れ線グラフにしてみよう。

```
# 調査年ごとに同類婚の合計比率を算出
```

```
PropHomg <- apply(PropDiag_byCensus, 2, sum)
```

```
PropHomg
```

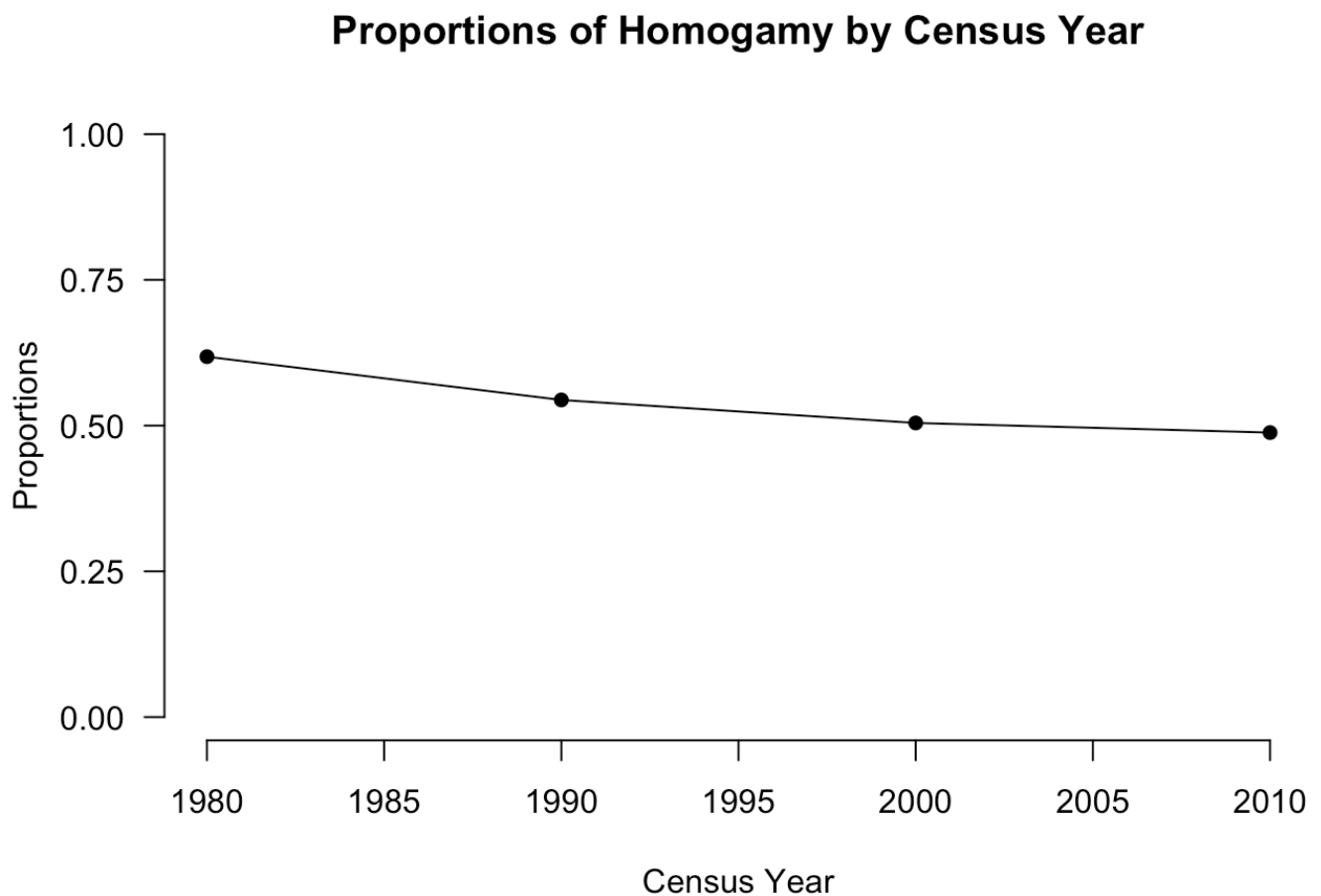
```
##      1980      1990      2000      2010
```

```
## 0.6182365 0.5440000 0.5045045 0.4880000
```

折れ線グラフを描く

```
plot(x = seq(1980, 2010, 10), # "x=" は省略してもよい
     y = PropHomg,             # "y=" は省略してもよい
     type = "o",
     pch = 16,
     xlim = c(1980, 2010),
     ylim = c(0, 1),
     ann = FALSE,
     axes = FALSE)

axis(1)
axis(2, at = seq(0, 1, 0.25), las = 1)
mtext("Census Year", side = 1, line = 3)
mtext("Proportions", side = 2, line = 3)
title("Proportions of Homogamy by Census Year")
```



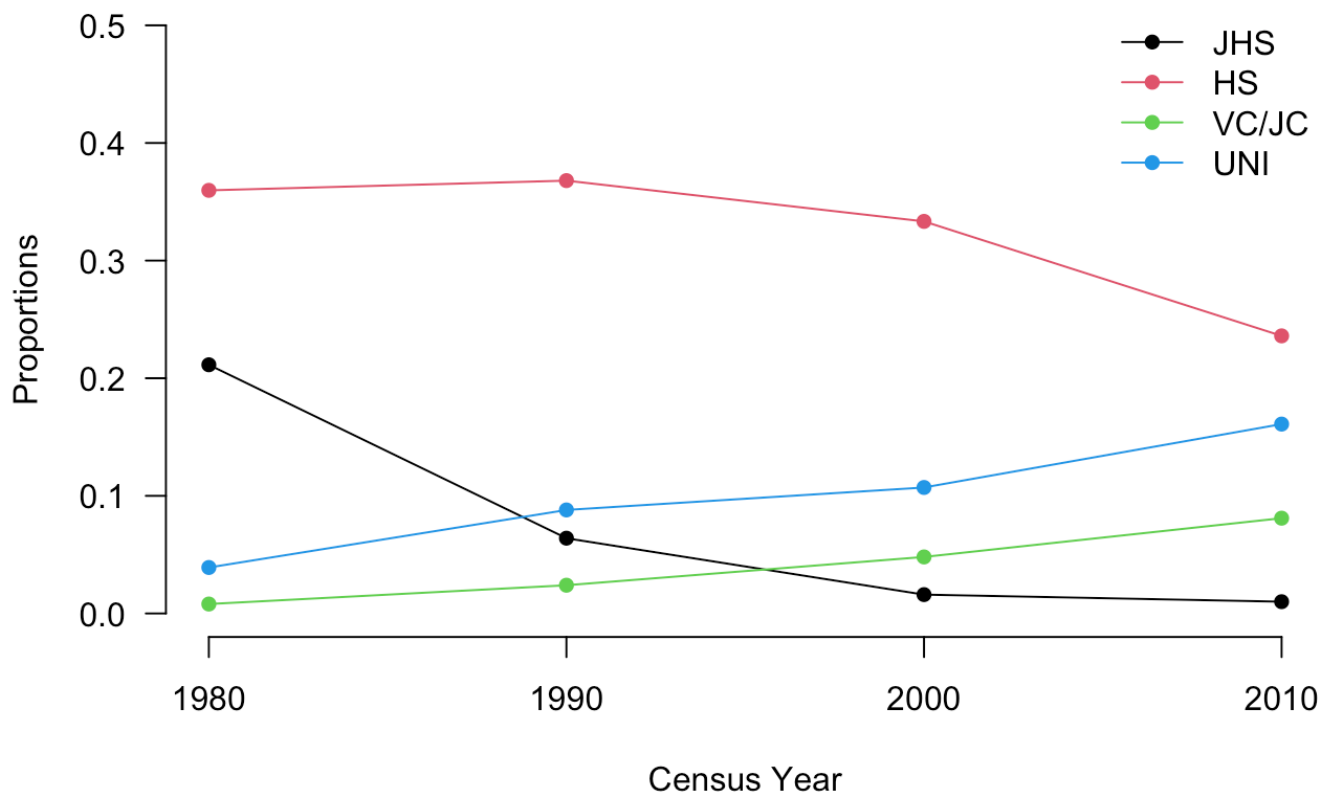
`plot()` の中で指定した引数について少しだけ補足すると以下の通りである。

- `type` : 線と点のどちらをプロットするか、あるいはその両者をプロットするか
- `pch` : 点 (マーカー) の種類

`type` と `pch` の種類については `plot()` のヘルプページを参照されたい(コンソール画面に `?plot` と入力すればよい)。

`matplot()` は `plot()` の拡張版で、引数 `y` に行列を指定できる。例えば、学歴別の同類婚がそれぞれ全夫婦に占める割合の変化をプロットしてみよう。

```
matplot(seq(1980, 2010, 10), t(PropDiag_byCensus),
        type = "o",
        lty = 1,
        pch = 16,
        xlim = c(1980, 2010),
        ylim = c(0, 0.5),
        ann = FALSE,
        axes = FALSE)
axis(1, at = seq(1980, 2010, 10))
axis(2, at = seq(0, 0.5, 0.1), las = 1)
mtext("Census Year", side = 1, line = 3)
mtext("Proportions", side = 2, line = 3)
legend("topright",
      rownames(PropDiag_byCensus),
      lty = 1,
      pch = 16,
      col = 1:4,
      bty = "n")
```



5.3 補足：クロス集計表の可視化

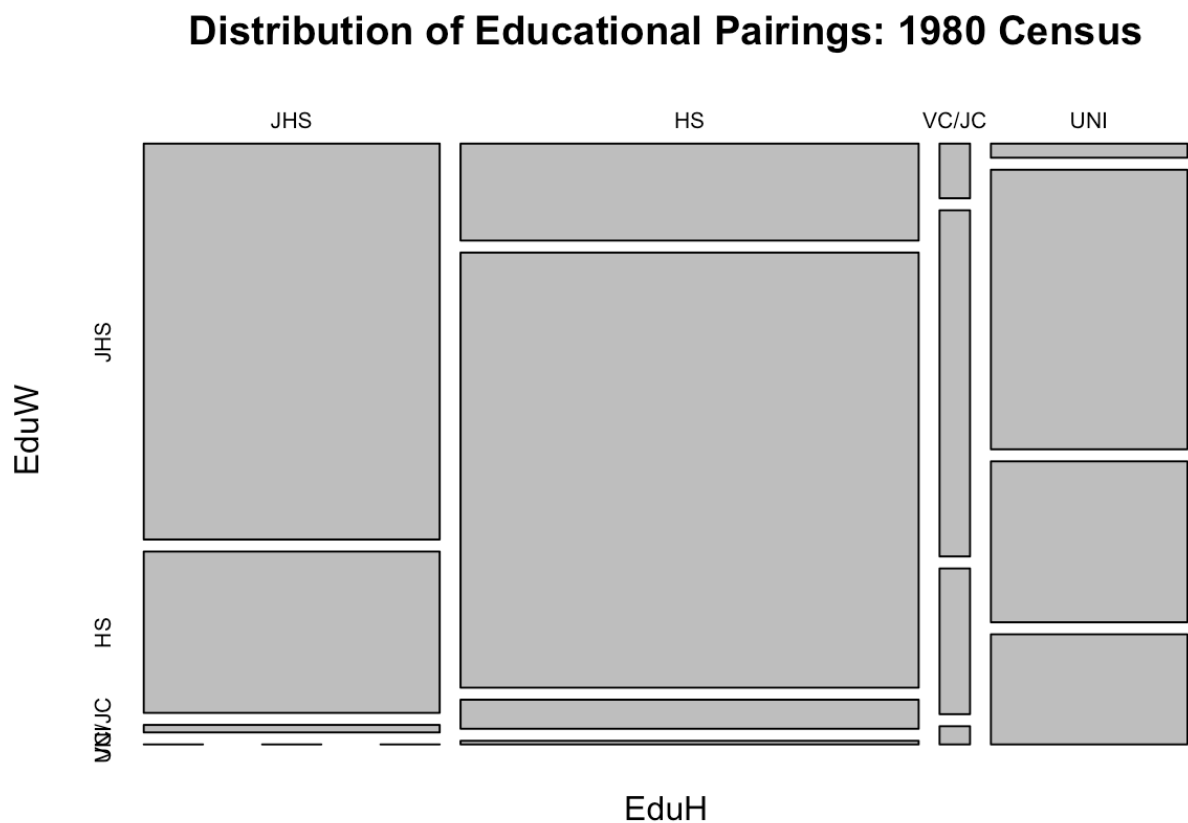
クロス表を可視化する方法として、以下ではモザイクプロット (mosaic plot) とバルーンプロット (balloon plot) について解説する。

5.3.1 モザイクプロット

モザイクプロットは以下の特徴を持つ

- タイルの各辺の長さが行／列の変数の周辺分布を表す
- タイルの面積がセル度数の大きさを示す

```
mosaicplot(t(EduWH1980), main = "Distribution of Educational Pairings: 1980")
```

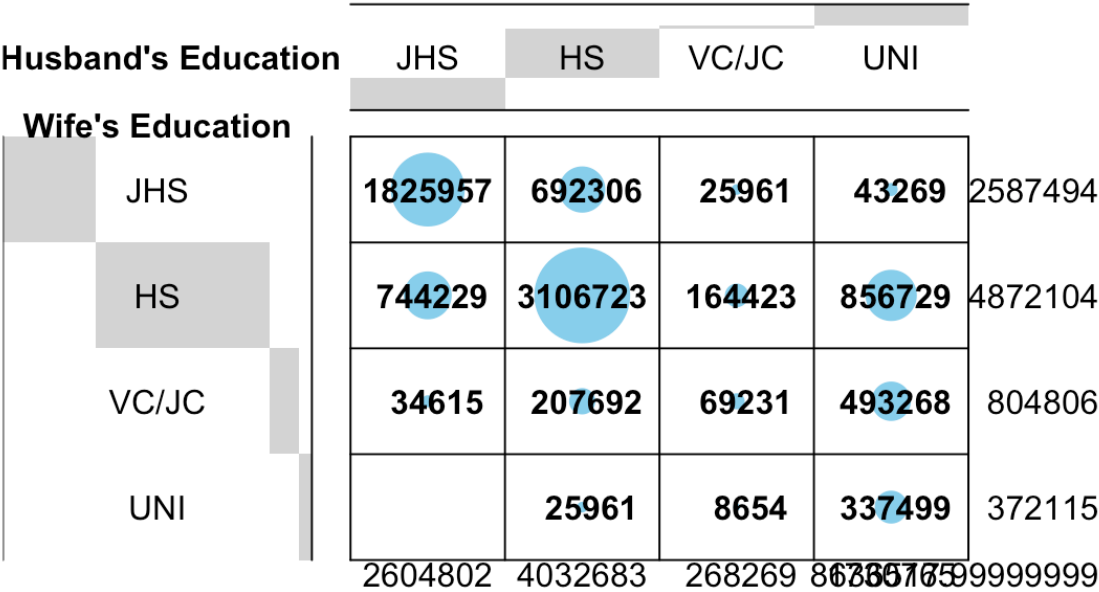


5.3.2 バルーンプロット

バルーンプロットは、セル度数に比例した大きさのマーカーをクロス表のセル上にプロットするものである。

```
library(gplots)
```

```
EduWH1980_tab <- as.table(EduWH1980) # tableクラスにしてから引き渡す必要がある
balloonplot(x = t(EduWH1980_tab),
            main = "",
            ylab = "Wife's Education",
            xlab = "Husband's Education")
```



Reference

Fukuda, Setsuya, Shohei Yoda, and Ryohei Mogi. 2021. “Educational Assortative Mating in Japan: Evidence from the 1980 Census.” *The Journal of Population Studies* 57: 1–20.

<https://doi.org/10.24454/jps.2101001>.

Lichter, Daniel T., and Zhenchao Qian. 2019. “The Study of Assortative Mating: Theory, Data, and Analysis.” In *Analytical Family Demography*, edited by Robert Schoen, 303–37. The Springer Series on Demographic Methods and Population Analysis. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-93227-9_13.

Uchikoshi, Fumiya, and James M. Raymo. 2021. *Educational Assortative Mating in Japan: Insights into Social Change and Stratification*. 1st ed. 2021版. Springer.

<https://link.springer.com/book/10.1007/978-981-16-3713-1>.

5 R による空間的自己相関の分析事例：統計量の算出とその地図化

野村侑平（早稲田大学・院）・山内昌和（早稲田大学）

Rによる空間的自己相関の分析事例 ：統計量の算出とその地図化

野村侑平（早稲田大学・院）・山内昌和（早稲田大学）

概要

2

1. はじめに
2. 自己紹介
3. 地理学における空間分析と関連研究
4. Rによる空間的自己相関の分析事例
 - 1) データの入手方法と読み込み
 - 2) コロプレスマップの作成
 - 3) 統計量の算出とその地図化

1. はじめに

3

□ 本報告の目的

- 報告者自身（野村）もRの初心者であるが，まったくの未経験であっても手順を踏めば利活用できるようにする
- 地理学・人口学における空間分析と関連研究について触れながら，空間的自己相関に関する見識を深めつつ，Rを用いた空間統計量の算出とその地図化を試みる
- その際に，政府統計より必要なデータを入手・整理する段階から上記の目的までの一連の作業工程を紹介する

2. 自己紹介（野村）

4

□ 所属・研究内容

- 早稲田大学大学院教育学研究科博士後期課程 在学中
- 人文地理学専攻
- 研究関心は，日本国内の外国人集住地区の様相，外国人人口統計の精度について

□ R使用歴

- 6か月（2022年10月～）
- プログラミングに関する知識はゼロ

□ 教育経験

- 2021年度以降，複数の私大附属高で地理の非常勤講師として勤務

□ 空間的自己相関とは

- ・ 自己相関

ある母集団を構成する要素に関する変量の測定値が何らかの意味で相関すること（奥野 1981）で、通常は時系列的自己相関を指す

- ・ 空間的自己相関

- ・ Tobler（1966）が論じたように、ある地域の事象は周囲の地域の事象に関係しながら存在する
- ・ 対象地域全体での空間的自己相関（グローバル）と部分地域での空間的関連性（ローカル）の2種類がある（張 2000）

□ 空間的自己相関を判定する手法：空間統計量の算出

- ・ グローバル（Global Indicator of Spatial Association; GISA）

→ 領域全体の空間的自己相関を評価

例. Moran（1948）：モラン I 統計量を提案

- ・ ローカル（Local Indicator of Spatial Association; LISA）

→ 部分領域の局所的な空間的関連性を評価

例. Anselin（1995）：ローカルモラン I 統計量を発案

※1990年代以降、GISの普及による空間解析法の開発の進展（奥野 2001）やFotheringham et al（2000）などの出版により、以上のような空間分析が注目されるように

□ モラン I 統計量

n個の領域で得られた標本データ $y_1, \dots, y_n$ の空間的自己相関を示す統計量

$$I = \frac{N}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j \overset{\text{近隣自治体}}{w_{ij}} (\overset{\text{当該自治体}}{y_j - \bar{y}})(y_i - \bar{y})}{\sum_i (y_i - \bar{y})^2}$$

※Iは通常の相関係数同様に+1~-1の値を取る

- ・ Iが+1に近いほど、強い正の空間的自己相関
(類似の値を示す領域が総じてまとまって分布する)
- ・ Iが-1に近いほど、強い負の空間的自己相関
(短い間隔で空間的に周期的変化を示す (市松模様))

3. 地理学における空間分析と関連研究

□ モラン I 統計量の計算結果の例

- ・ ポイント：対象地域全体の空間的自己相関を評価する
 - ①：正の空間的自己相関
 - ②：ランダム
 - ③：負の空間的自己相関

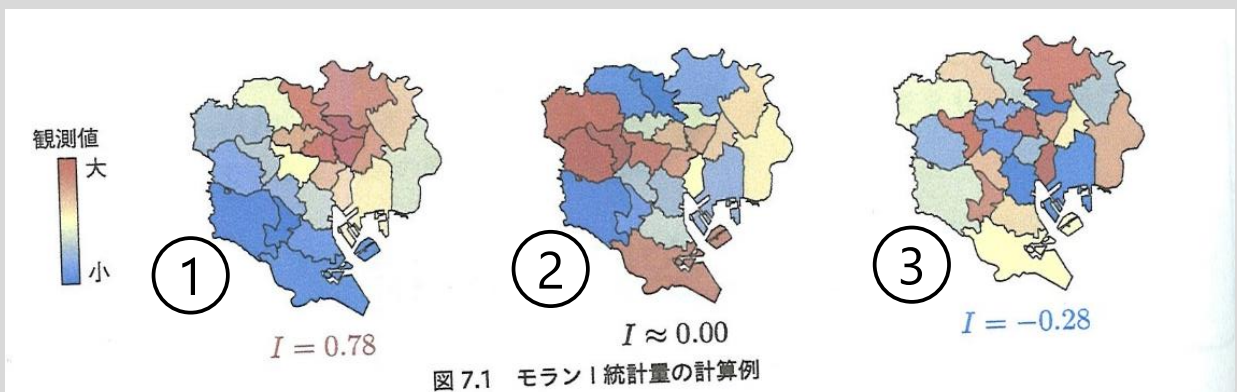


図 7.1 モラン I 統計量の計算例

出典：村上（2022，68）

□ ローカルモラン I 統計量

モラン I 統計量を領域ごとに分解することで得られる統計量で定義される（村上 2022, 73）

$$l_i = \frac{1}{m} (y_i - \bar{y}) \sum_j w_{ij} (y_j - \bar{y})$$

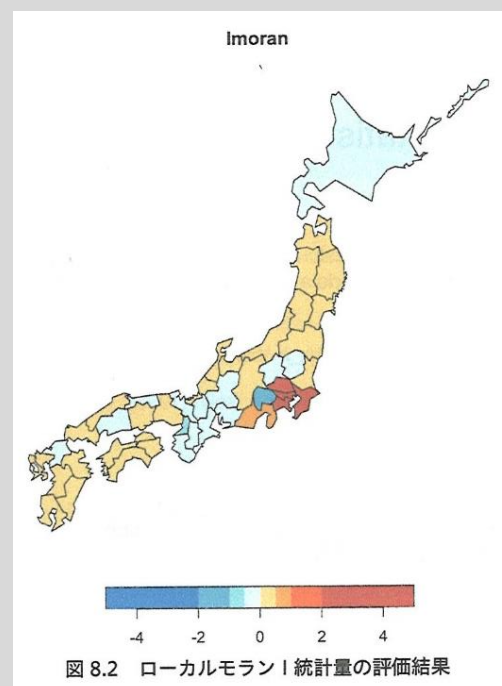
$$m = \frac{1}{n-1} \sum_{i=1, i \neq j}^n (y_j - \bar{y})^2 - \bar{y}^2$$

- ・ l_i が正の場合はローカルな正の自己相関
（当該領域の値と近隣領域の値が似た傾向）
- ・ l_i が負の場合はローカルな負の自己相関
（当該領域の値と近隣領域の値が逆の傾向）

3. 地理学における空間分析と関連研究

□ ローカルモラン I 統計量の評価結果（都道府県別人口を例に）

- ・ ポイント：個々の領域の空間的関連性を評価
 - ・ 高い正の値（近隣領域と類似）
→ 南関東（周囲も自らも人口は多い）
 - ・ 低い負の値（近隣領域と特異）
→ 山梨県（周囲の人口は多いが自らの人口は少ない）
→ 大阪府（周囲の人口は少ないが自らの人口は多い）



出典：村上（2022，76）

3. 地理学における空間分析と関連研究

11

□ 地理学における空間分析の重要性

- ・空間的な従属性や空間的な異質性の存在（中谷 2003: 23）

□ 地理学における関連研究（モラン統計量関連）

- ・ある指標の地理的なパターンへの関心

例：地域別死亡率の地域差（中谷 1996）

福祉サービスの偏在性（宮澤 2003）

職業からみた居住地域構造の把握（中澤 2016）

外国人集住地区の検出（福本 2010）

地方税徴収率の地域差（佐藤 2021）

COVID-19の感染状況（駒木・畠山 2020）

3. 地理学における空間分析と関連研究

12

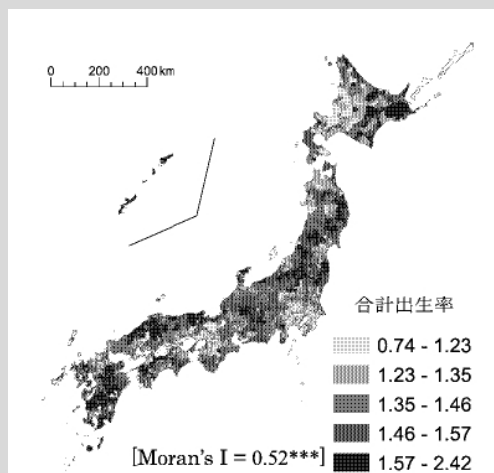
□ 人口学における先行研究と応用可能性

- ・出生，死亡，人口推計などに応用

例．日本国内における合計出生率（鎌田・岩澤 2009）

20～39歳女性有配偶者割合（岩澤など 2022）

人口推計の誤差率（鎌田など 2020）など



出典：鎌田・岩澤（2009，9）

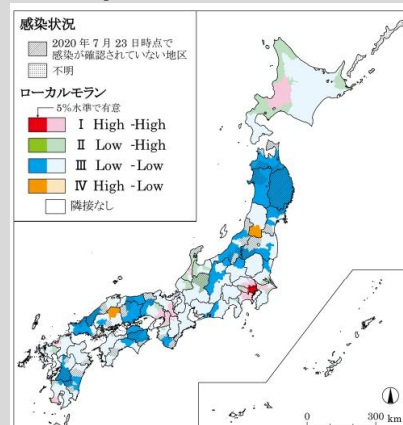


図1 10万人あたり累積感染者数のローカルモラン分布
(2020年7月23日時点)

出典：駒木・畠山（2020）

□ 使用するソフト

- ・ R（今回はR studio上でデモ）

□ 扱うデータ

- ・ e-Stat (<https://www.e-stat.go.jp/>) からダウンロード
- ・ 都道府県別外国人比率（2020年国勢調査）を事例にする

□ 作業工程

- 1) データの入手方法と読み込み
- 2) コロプレスマップの作成
- 3) 統計量の算出とその地図化

1) データの入手方法と読み込み

□ キーワード検索で「国勢調査」と入力，検索



出典：<https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

15

□ 「国勢調査」をクリック

The screenshot shows the e-Stat website interface. The search results for '国勢調査' (Census) are displayed. The '国勢調査' item is highlighted with a red box in the list of government statistics.

政府統計コード	政府統計名	概要
00200521	国勢調査	詳細
00200524	人口推計	詳細
00600160	建築物実態調査	詳細
00600161	増改築・改装等調査	詳細

出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

16

□ 「令和2年国勢調査」の「データベース」をクリック

The screenshot shows the e-Stat website interface. The search results for '令和2年国勢調査' (2020 Census) are displayed. The 'データベース' (Database) link is highlighted with a red box.

国勢調査	データベース	件数 更新日	ファイル	件数 更新日	概要
時系列データ		62件 2022-03-31		226件 2022-03-31	
都道府県・市区町村別の主な結果				5件 2022-07-22	
令和2年国勢調査		544件 2022-12-27		8,474件 2023-03-10	
平成27年国勢調査				11,414件 2021-11-30	

出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

17

□ 「人口等基本集計」をクリック

The screenshot shows the e-Stat website interface. The search results for '国勢調査' (Census) are displayed. The '国勢調査' section is highlighted with a red box, and the '人口等基本集計' (Basic Population and Housing Census) is selected. The search results show 544 items for '国勢調査' and 8,474 items for 'ファイル' (File). The '国勢調査' section is further detailed with a table of results.

国勢調査	公開 (更新) 日
■ 人口等基本集計 (主な内容: 男女・年齢・配偶関係、世帯の構成、住居の状態、母子・父子世帯、国籍など) [225件]	2022-12-27
■ 就業状態等基本集計 (主な内容: 労働力状態、就業者の産業・職業、教育など) [182件]	2022-12-27
■ 抽出詳細集計 (主な内容: 就業者の産業・職業 (小・中分類) など) [47件]	2022-12-27
■ 従業地・通学地による人口、就業状態等集計 (主な内容: 従業地・通学地による人口、昼夜間人口)	

出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

18

□ 「外国人」の「44-1」の「⇒DB」をクリック

The screenshot shows the e-Stat website interface. The search results for '外国人' (Foreigners) are displayed. The '44-1' row is highlighted, and the '⇒DB' button is clicked. The search results show a table of data for '外国人'.

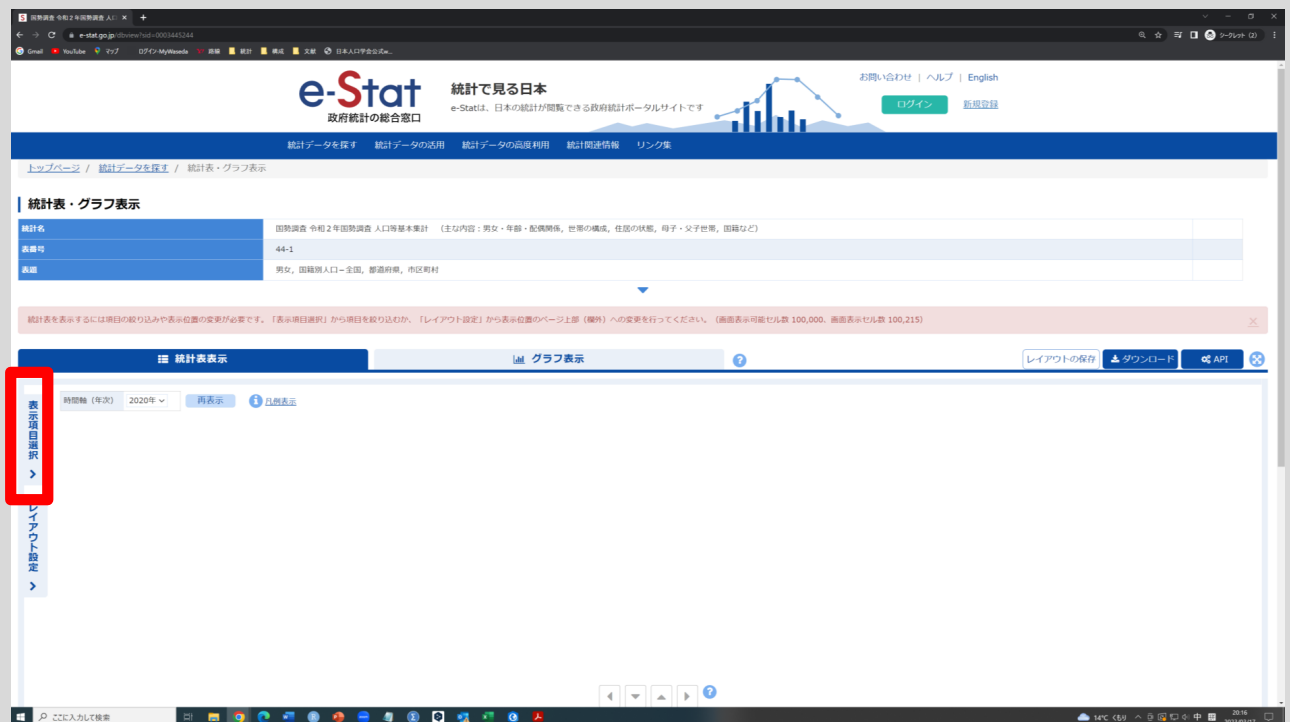
外国人	2020年10月	2021-11-30	⇒DB	⇒API
44-1 男女, 国籍別人口-全国, 都道府県, 市区町村			⇒DB	⇒API
44-2 男女, 国籍別人口-全国, 都道府県, 市区町村 (人口集中地区)			⇒DB	⇒API
45-1-1 男女, 年齢 (5歳階級), 国籍別人口-全国, 都道府県, 21大都市, 21大都市の区, 県庁所在地, 人口20万以上の市			⇒DB	⇒API
45-1-2 男女, 国籍別平均年齢及び年齢中位数-全国, 都道府県, 21大都市, 21大都市の区, 県庁所在地, 人口20万以上の市			⇒DB	⇒API
45-2 男女, 年齢 (5歳階級), 国籍別人口構成比 [年齢別] -全国, 都道府県, 21大都市, 21大都市の区, 県庁所在地, 人口20万以上の市			⇒DB	⇒API
46 男女, 年齢 (5歳階級), 配偶関係, 国籍別人口 (15歳以上) -全国, 都道府県, 21大都市, 特別区, 人口50万以上の市			⇒DB	⇒API
47-1 世帯主の国籍, 世帯の家族類型, 外国人のいる世帯の類型別一般世帯数 (外国人のいる一般世帯) -全国, 都道府県, 21大都市, 特別区, 人口50万以上の市			⇒DB	⇒API
47-2 世帯主の国籍, 世帯の家族類型, 外国人のいる世帯の類型別一般世帯人員 (外国人のいる一般世帯) -全国, 都道府県, 21大都市, 特別区, 人口50万以上の市			⇒DB	⇒API
47-3 世帯主の国籍, 世帯の家族類型, 外国人のいる世帯の類型別外国人人員 (外国人のいる一般世帯) -全国, 都道府県, 21大都市, 特別区, 人口50万以上の市			⇒DB	⇒API
48-1 外国人のいる世帯の類型, 住宅の所有の関係別一般世帯数 (外国人のいる一般世帯) -全国, 都道府県, 21大都市, 特別区, 人口50万以上の市			⇒DB	⇒API
48-2 外国人のいる世帯の類型, 住宅の所有の関係別一般世帯人員 (外国人のいる一般世帯) -全国, 都道府県, 21大都市, 特別区, 人口50万以上の市			⇒DB	⇒API
48-3 外国人のいる世帯の類型, 住宅の所有の関係別一般世帯の1世帯当たり人員			⇒DB	⇒API

出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

19

□ 「表示項目選択」をクリック

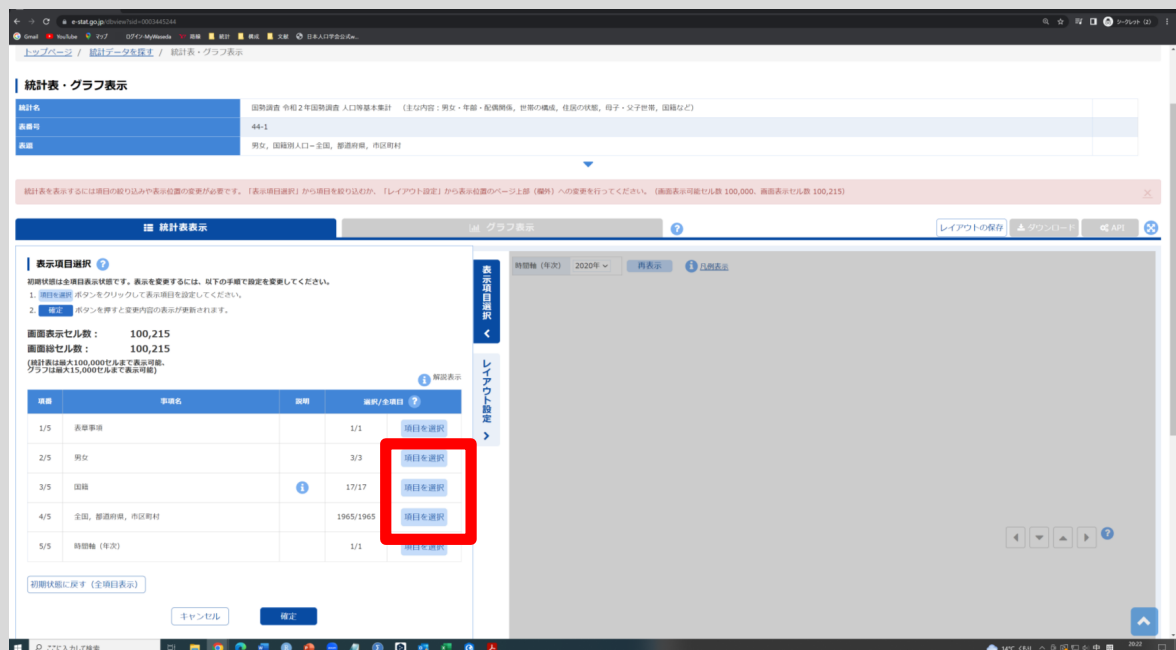


出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

20

□ 「男女」, 「国籍」, 「全国, 都道府県, 市区町村」の「項目を選択」をクリック

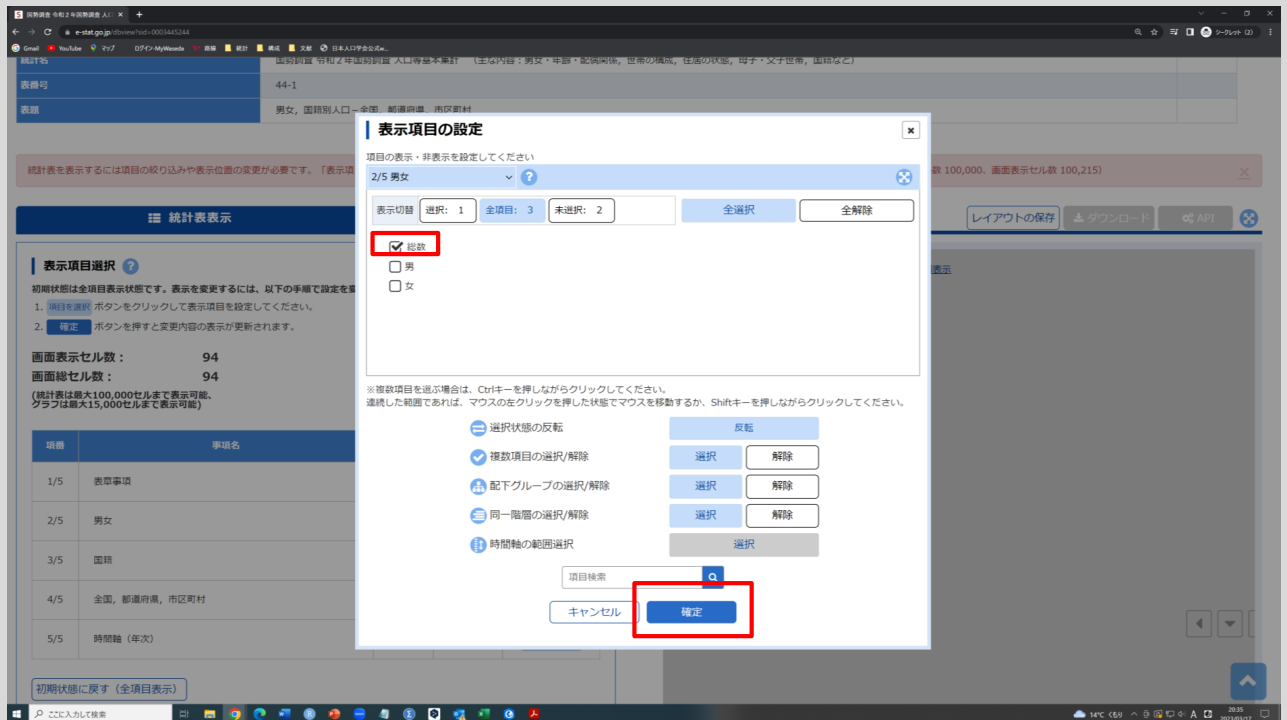


出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

21

□ 「総数」のみをチェック→「確定」をクリック

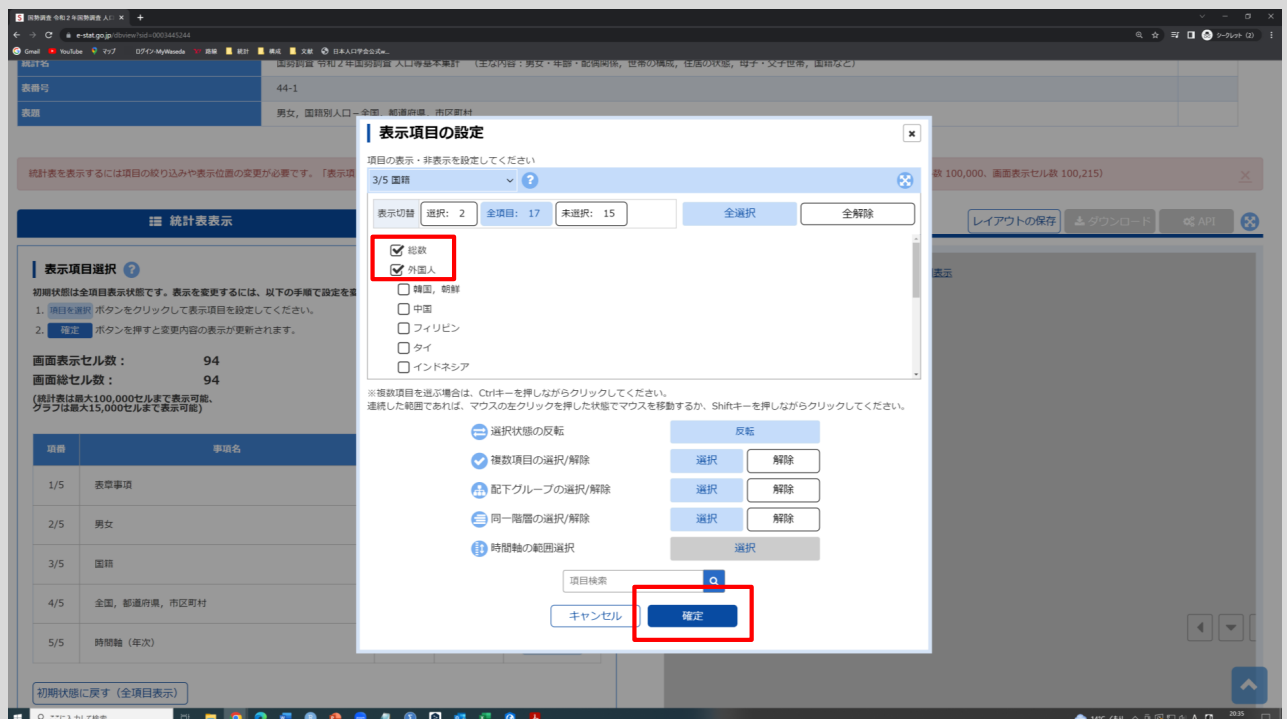


出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

22

□ 「総数」と「外国人」をチェック→「確定」をクリック

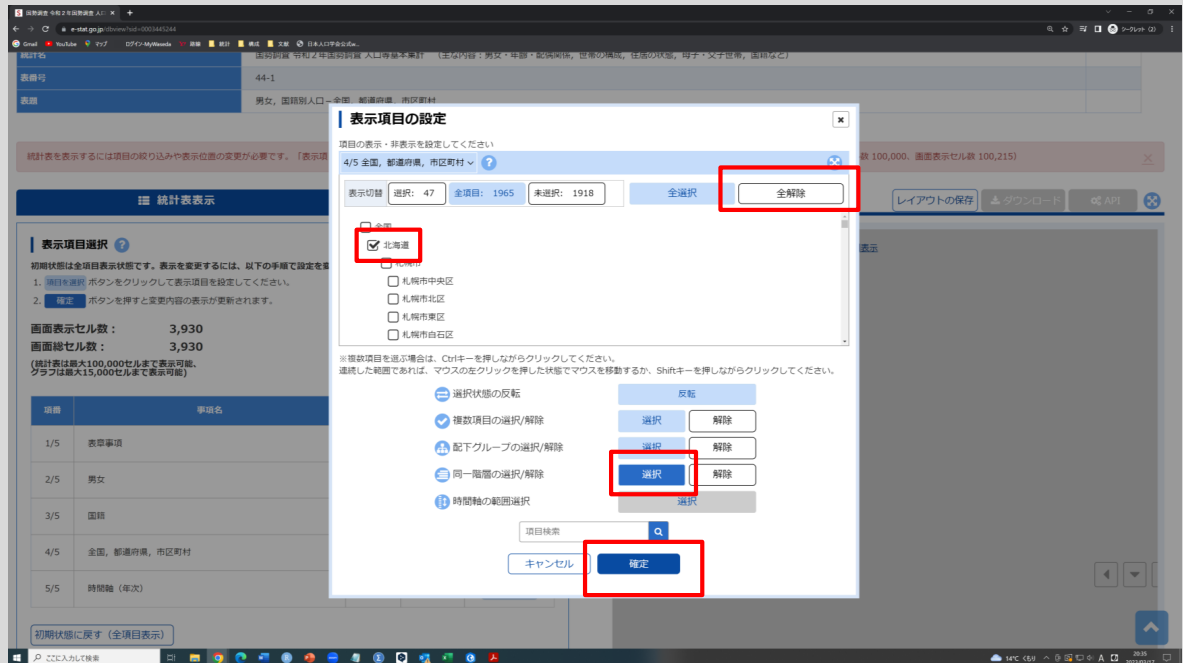


出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

23

- 一旦「全解除」したのちに、「北海道」をチェックし、「同一階層の選択」をクリック→「確定」をクリック

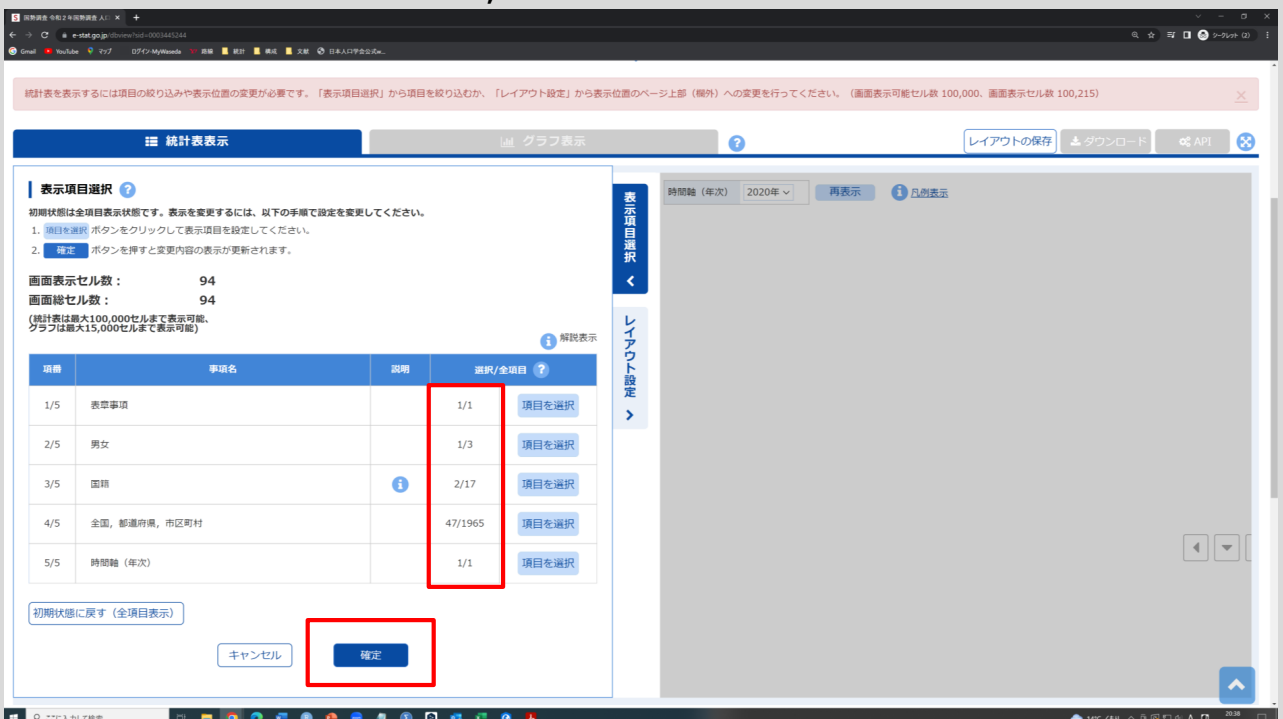


出典： <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

24

- 選択項目数を確認し、「確定」をクリック



出典： <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

25

□「ダウンロード」をクリック

The screenshot shows the e-Stat website interface. The 'ダウンロード' (Download) button is highlighted with a red box in the top right corner of the main content area. The page displays statistical data for '国勢調査 令和2年国勢調査 人口等基本集計' (Census of Population and Housing 2020, Basic Collection of Population and Housing Statistics). The data is presented in a table with columns for '地域' (Region), '人口' (Population), and '外国人' (Foreigners). The 'ダウンロード' button is located in the top right corner of the main content area, next to the 'レイアウトの保存' (Save Layout) button.

出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

26

□「ダウンロード」をクリック

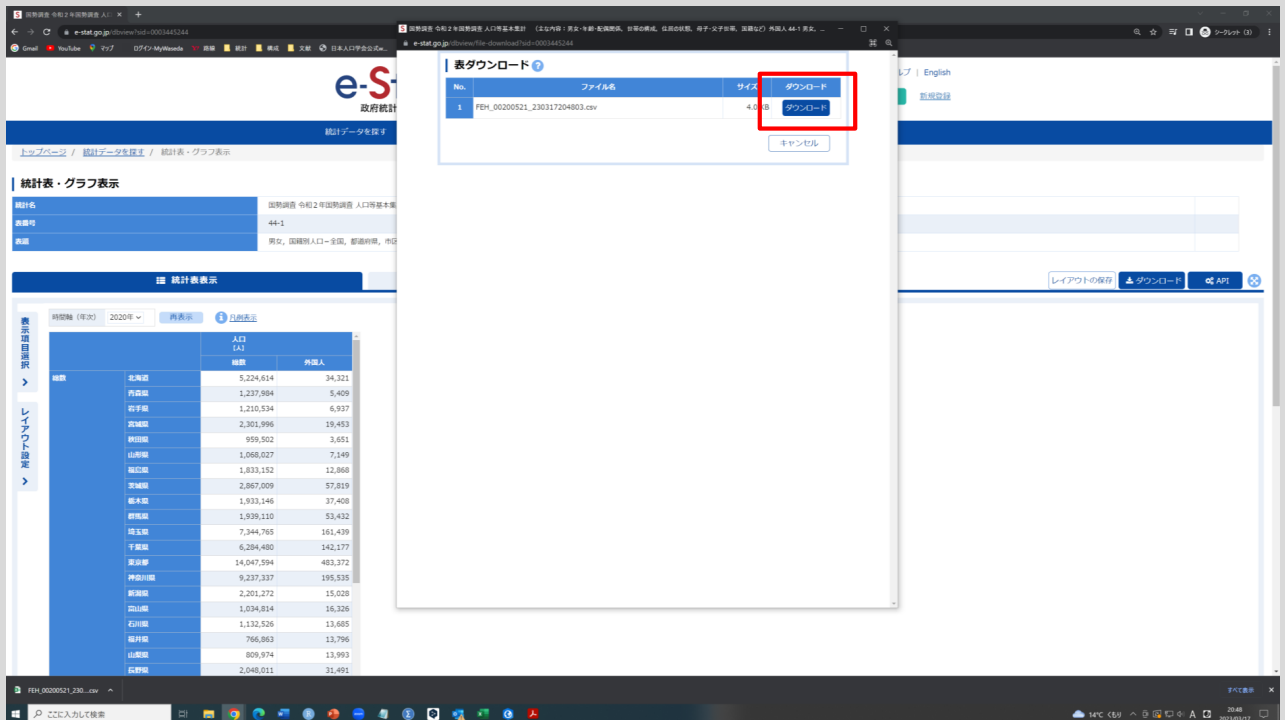
The screenshot shows the e-Stat website interface with the 'ダウンロード' (Download) button highlighted with a red box. The page displays the '表ダウンロード' (Table Download) dialog box. The dialog box contains options for 'ダウンロード範囲' (Download Range), 'ファイル形式' (File Format), 'ヘッダの出力' (Output of Headers), 'コードの出力' (Output of Codes), '階層コードの出力' (Output of Hierarchy Codes), and '凡例の出力' (Output of Legend). The 'ダウンロード' button is located at the bottom right of the dialog box.

出典: <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

27

□ 「ダウンロード」をクリック



出典： <https://www.e-stat.go.jp/>

1) データの入手方法と読み込み

28

□ 必要なパッケージのインストールと、ダウンロードしたcsvファイルをRに読み込ませる

#目的：都道府県別外国人比率の算出，統計量の算出およびその地図化

```
rm(list=ls()) #リセット
```

#フォルダ設定

```
setwd("C:/Users/Nomura Y/Desktop/230322_研究集会")
```

#パッケージの読み込み

```
install.packages( c("tidyverse","sf", "NipponMap",  
"RColorBrewer","ggplot2","spdep","rgdal")) #各パッケージの内容はその  
都度説明する
```

#ファイルの読み込み

```
d <- read.csv("FEH_00200521_230317204803.csv",skip =12,fileEncoding  
= "Shift-JIS") #13行目から読み込む
```

□ データの中身を確認する

```
#データを見る
d
```

```
時間軸.年次..コード 時間軸.年次.男女.コード 男女 全国.都道府県.市区町村.コード 全国.都道府県.市区町村 X.国籍 総数 外国人
1 2020000000 2020年 0 総数 1000 北海道 NA 5,224,614 34,321
2 2020000000 2020年 0 総数 2000 青森県 NA 1,237,984 5,409
3 2020000000 2020年 0 総数 3000 岩手県 NA 1,210,534 6,937
4 2020000000 2020年 0 総数 4000 宮城県 NA 2,301,996 19,453
5 2020000000 2020年 0 総数 5000 秋田県 NA 959,502 3,651
6 2020000000 2020年 0 総数 6000 山形県 NA 1,068,027 7,149
7 2020000000 2020年 0 総数 7000 福島県 NA 1,833,152 12,868
8 2020000000 2020年 0 総数 8000 茨城県 NA 2,867,009 57,819
9 2020000000 2020年 0 総数 9000 栃木県 NA 1,933,146 37,408
10 2020000000 2020年 0 総数 10000 群馬県 NA 1,939,110 53,432
```

・・・(都合により中略)・・・

```
38 2020000000 2020年 0 総数 38000 愛媛県 NA 1,334,841 11,159
39 2020000000 2020年 0 総数 39000 高知県 NA 691,527 4,220
40 2020000000 2020年 0 総数 40000 福岡県 NA 5,135,214 66,699
41 2020000000 2020年 0 総数 41000 佐賀県 NA 811,442 5,940
42 2020000000 2020年 0 総数 42000 長崎県 NA 1,312,317 8,316
43 2020000000 2020年 0 総数 43000 熊本県 NA 1,738,301 14,591
44 2020000000 2020年 0 総数 44000 大分県 NA 1,123,852 10,168
45 2020000000 2020年 0 総数 45000 宮崎県 NA 1,069,576 6,474
46 2020000000 2020年 0 総数 46000 鹿児島県 NA 1,588,256 10,037
47 2020000000 2020年 0 総数 47000 沖縄県 NA 1,467,480 18,157
```

□ 必要な列のみを抽出する

```
#必要な列だけを抽出する
d1 <- d[,c(5,6,8,9)]
```

```
#データを見る
d1
```

```
全国.都道府県.市区町村.コード 全国.都道府県.市区町村 総数 外国人
1 1000 北海道 5,224,614 34,321
2 2000 青森県 1,237,984 5,409
3 3000 岩手県 1,210,534 6,937
4 4000 宮城県 2,301,996 19,453
5 5000 秋田県 959,502 3,651
... (都合により中略) ...
42 42000 長崎県 1,312,317 8,316
43 43000 熊本県 1,738,301 14,591
44 44000 大分県 1,123,852 10,168
45 45000 宮崎県 1,069,576 6,474
46 46000 鹿児島県 1,588,256 10,037
47 47000 沖縄県 1,467,480 18,157
```

1) データの入手方法と読み込み

31

□ 文字列から数値に変換したのち，外国人比率を計算する

```
#文字列から数値に変換する
library(tidyverse)
d1 <- d1 %>% mutate(総数 = parse_number(総数), 外国人 =
parse_number(外国人))

#外国人比率を算出する
d1$外国人比率 <- d1$外国人/d1$総数*100
```

1) データの入手方法と読み込み

32

□ 外国人比率が算出されたかを確認する

```
#データを見る
d1

全国,都道府県,市区町村,コード 全国,都道府県,市区町村 総数 外国人 外国人比率
1          1000      北海道 5224614 34321 0.6569098
2          2000      青森県 1237984 5409 0.4369200
3          3000      岩手県 1210534 6937 0.5730529
4          4000      宮城県 2301996 19453 0.8450493
5          5000      秋田県 959502 3651 0.3805099
6          6000      山形県 1068027 7149 0.6693651
7          7000      福島県 1833152 12868 0.7019603
8          8000      茨城県 2867009 57819 2.0167010
9          9000      栃木県 1933146 37408 1.9350841
10         10000     群馬県 1939110 53432 2.7554909
          ... (都合により中略) ...
38         38000     愛媛県 1334841 11159 0.8359797
39         39000     高知県 691527 4220 0.6102437
40         40000     福岡県 5135214 66699 1.2988553
41         41000     佐賀県 811442 5940 0.7320301
42         42000     長崎県 1312317 8316 0.6336884
43         43000     熊本県 1738301 14591 0.8393828
44         44000     大分県 1123852 10168 0.9047455
45         45000     宮崎県 1069576 6474 0.6052866
46         46000     鹿児島県 1588256 10037 0.6319510
47         47000     沖縄県 1467480 18157 1.2372911
```

□ 都道府県界の地図を表示させる

```
#地図化する際に必要なパッケージを読み込む
library(sf) #地物をデータとして格納・参照するための国際規格
library(NipponMap) #都道府県界のshpファイルが格納されているパッケージ
library(RColorBrewer) #色分け
library(ggplot2) #地図やグラフなどを可視化させる

#都道府県界のshpファイルを読み込む
shp <- system.file("shapes/jpn.shp", package = "NipponMap")[1]
pref <- read_sf(shp)
```

□ 都道府県界の地図データの中身を確認する

```
#shpファイルの中身を確認する
pref

Simple feature collection with 47 features and 5 fields
Geometry type: POLYGON
Dimension: XY
Bounding box: xmin: 127.6461 ymin: 26.0709 xmax: 148.8678 ymax: 45.5331
CRS: NA
# A tibble: 47 × 6
  SP_ID jiscode name population region geometry
  <chr> <chr> <chr> <dbl> <chr> <POLYGON>
1 1 01 Hokkaido 5506419 Hokkaido ((139.7707 42.3018, 139.8711 42.6623, 140...
2 2 02 Aomori 1373339 Tohoku ((140.8727 40.48187, 140.6595 40.4018, 140...
3 3 03 Iwate 1330147 Tohoku ((140.7862 39.85982, 140.8199 39.86421, 14...
4 4 04 Miyagi 2348165 Tohoku ((140.2802 38.01415, 140.2802 38.01417, 14...
5 5 05 Akita 1085997 Tohoku ((140.7895 39.86026, 140.8253 39.6481, 140...
6 6 06 Yamagata 1168924 Tohoku ((140.2802 38.01415, 140.2799 37.97209, 14...
7 7 07 Fukushima 2029064 Tohoku ((140.2799 37.9721, 140.2799 37.97209, 140...
8 8 08 Ibaraki 2969770 Kanto ((139.7322 36.08471, 139.7058 36.13276, 13...
9 9 09 Tochigi 2007683 Kanto ((139.4263 36.33584, 139.3703 36.3661, 139...
10 10 10 Gunma 2008068 Kanto ((139.3836 36.35887, 139.6541 36.213, 139...
# ... with 37 more rows
# i Use `print(n = ...)` to see more rows
```

2) コロプレスマップの作成

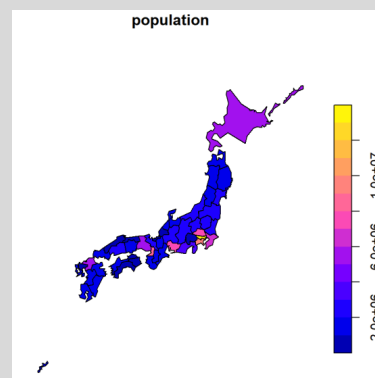
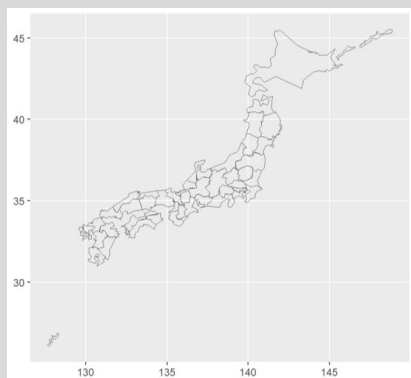
35

□ パッケージ「NipponMap」に含まれる「population」のコロプレスマップを表示させる

```
#白地図を表示させる
ggplot()+
geom_sf(data = pref, fill = "transparent")

#CRS（座標参照系）をWGS84に設定し、地図投影法をUTM54(JGD2000)に変更する
st_crs(pref) <- 4326
pref2 <- st_transform(pref, 3100)

#試しに総人口のコロプレスマップを表示させる
plot(pref2[, "population"])
```



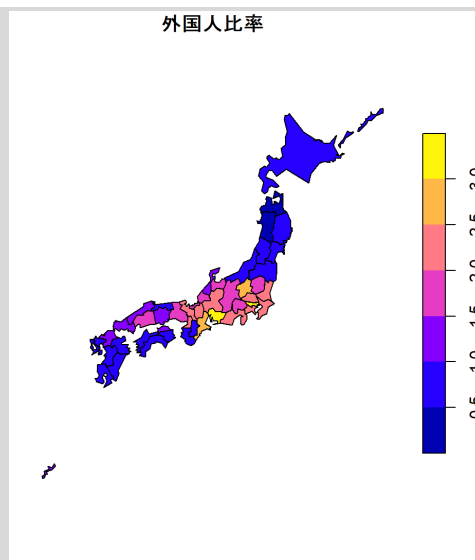
2) コロプレスマップの作成

36

□ 「外国人比率」のコロプレスマップを作成する

```
#shpファイルと外国人比率のデータを結合する
d2 <- cbind(pref2, d1) #お互いに47行なので横に結合できる #注意点としてgeometryが含まれる方を先頭にする
```

```
#外国人比率のコロプレスマップを表示させる
plot(d2[, "外国人比率"])
```



2) コロプレスマップの作成

37

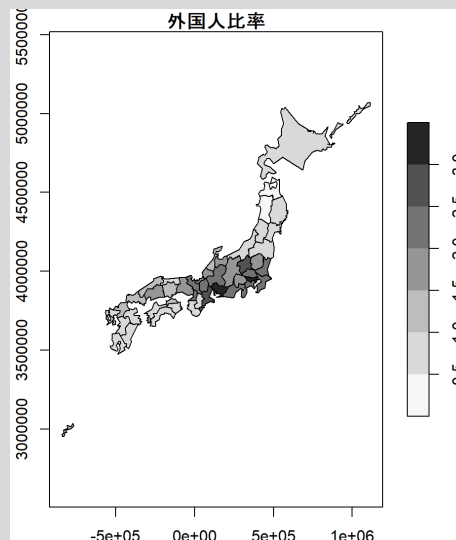
□ 表示色を変更する

#階級区分と色を変更する

nc <- 7 #階級区分を7にする

pal <- brewer.pal(nc, "Greys") #色をモノクロにする

plot(d2[, "外国人比率"], pal=pal, axes=TRUE, nbreaks=nc) #地図を表示する



3) 統計量の算出とその地図化

38

□ 近接行列を作成する

#近接行列の作成

library(spdep) #空間的自己相関に関する分析を行うためのパッケージ

#沖縄県を除く

d3 <- d2[d2\$name != "Okinawa",]

#最近隣kゾーンの定義

coords <- st_coordinates(st_centroid(d3))

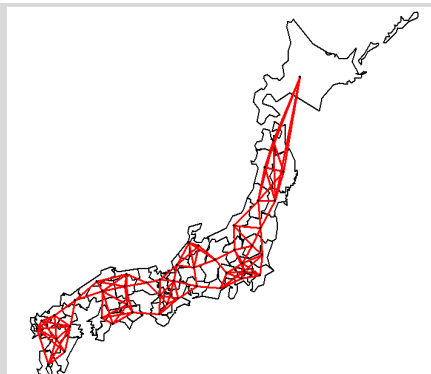
knn <- knearneigh(coords, 4)

nb2 <- knn2nb(knn)

#地図化

plot(st_geometry(d3), border="black")

plot(nb2, coords, add=TRUE, col="red", cex=0.01, lwd=1.5)



□ GISAを算出する

```
#GISAの算出
library(rgdal)
w <- nb2listw(nb2)

fr <- d3$"外国人比率"
moran <- moran.test(fr, listw=w)
moran
```

Moran I test under randomisation

data: fr
weights: w

Moran I statistic standard deviate = 7.3721, p-value = 8.4e-14
alternative hypothesis: greater
sample estimates:

Moran I statistic	Expectation	Variance
0.661896410	-0.022222222	0.008611595

□ LISAを算出する

```
#LISAの算出
d3$"外国人比率（沖縄県除く）" <- (0)
d3$"外国人総数（沖縄県除く）" <- (0)
d3$"総数（沖縄県除く）" <- (0)
d3$"外国人総数（沖縄県除く）" <- sum(d3$外国人)
d3$"総数（沖縄県除く）" <- sum(d3$総数)
d3$"外国人比率（沖縄県除く）" <- d3$"外国人総数（沖縄県除く）"/d3$"総数（沖縄県除く）"*100

lmoran <- localmoran(d3$"外国人比率",listw=w)
lm <- d3
lm$lmoran <- lmoran[,1]
lm$p値 <- lmoran[,5]

#p値0.05未満の都道府県にマークをする
#ダミー変数を作成する（このdummy1が1である自治体が地図上で色塗られることになる）
lm$dummy1 <- ifelse(lm$p値<0.05, 1, 0)

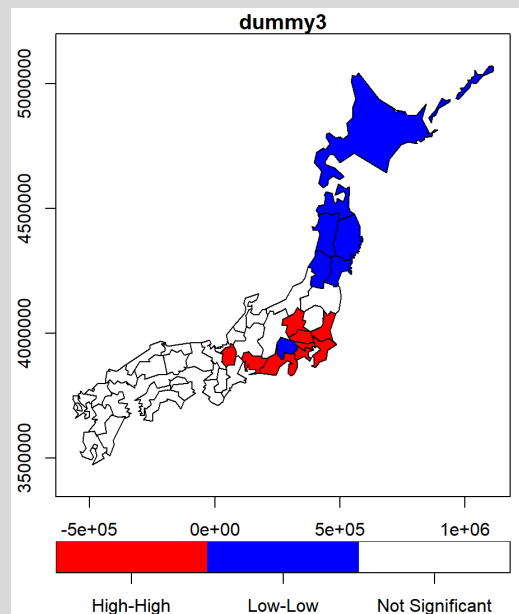
#ダミー変数を作成する（ある自治体の外国人比率が平均より大きい小さいかを判別する、大きいとき＝モラン散布図第1象限、第4象限に該当、小さいとき＝モラン散布図第2象限、第3象限に該当する。ある自治体のローカルモラン統計量が正の値のとき＝モラン散布図第1象限、第3象限に該当、負の値のとき＝モラン散布図第2象限、第4象限に該当）
lm$dummy2 <- ifelse(lm$外国人比率>lm$"外国人比率（沖縄県除く）" & lm$lmoran>0, "High-High",
  ifelse(lm$外国人比率<lm$"外国人比率（沖縄県除く）" & lm$lmoran<0, "Low-High",
    ifelse(lm$外国人比率<lm$"外国人比率（沖縄県除く）" & lm$lmoran>0, "Low-Low","High-Low"))))

#p値<0.05の自治体のみHH,LH,LL,HLを表示させる。0.05以上の自治体は、Not Significantとする
lm$dummy3 <- ifelse(lm$dummy1 == 0,"Not Significant",lm$dummy2)
```

□ LISAの地図化

#外国人比率のモラン散布図に基づいた，HH,LH,LL,HLの地図を作成する

```
pal1 <- c("#ff0000","#0000ff","#ffffff")
plot(lm[, "dummy3"], pal=pal1, axes=TRUE, key.length=1)
```



□ LISAの算出（市区町村の結果）

※版權の都合により当日の発表中のみお見せします。

- 岩澤美帆・菅桂太・鎌田健司・余田翔平・金子隆一 2022. 出生力の地域差に対する結婚力効果と夫婦出生力効果—対数線形モデルを利用した市区町村別合計出生率の分解—. 人口問題研究 78-1: 78-105.
- 奥野隆史 1981. 空間的自己相関論 (I) 測定と検定について. 筑波大学人文地理学研究 5: 165-183.
- 奥野隆史 2001. 計量地理学の新しい潮流——主としてローカルモデルについて——. 地理学評論 74A: 431-451.
- 鎌田健司・岩澤美帆 2009. 出生力の地域格差の要因分析—非正常性を考慮した地理的加重回帰法による検証—. 人口学研究 45: 1-20.
- 鎌田健司・小池司朗・菅桂太・山内昌和 2020. 地域別将来推計人口の精度評価—回帰モデルによる推計誤差の空間的特性の検証—. 人口問題研究 76-1: 41-66.
- 駒木信比古・畠山輝雄 2020. COVID-19感染者の分布状況に関する空間分析の試み. 2020年度日本地理学会秋季学術大会発表要旨.
- 佐藤洋 2021. 大都市圏における地方税の徴収率の規定要因と空間パターン—貧困問題との関係を中心に—. 地理学評論 94-1: 17-34.
- 張長平 2000. 空間データ分析と地理情報システム. 地学雑誌 109-1: 1-9.
- 中澤高志 2016. 職業別純移動にみる東京圏の居住地域構造. 経済地理学年報 62: 39-56.
- 中谷友樹 1996. 死亡率地図における空間単位の情報量統計学的評価—地理情報システムによる疾病地図解析システムの構築に向けて—GIS—理論と応用 4: 53-60.
- 中谷友樹 2003. 空間的共変動分析. 杉浦芳夫編『シリーズ〈人文地理学〉3 地理空間分析』23-48. 朝倉書店.
- 福本拓 2010. 東京および大阪における在日外国人の空間的セグリゲーションの変化——「オールドカマー」と「ニューカマー」間の差異に着目して——. 地理学評論 83-3: 288-313.
- 宮澤仁 2003. 関東地方における介護保険サービスの地域的偏在と事業者参入の関係——市区町村データの統計分析を中心に——. 地理学評論 76-2: 59-80.
- 村上大輔 2022. 『実践Data Scienceシリーズ Rではじめる地理空間データの統計解析入門』講談社.
- Anselin, L. 1995. Local indications of spatial association-LISA. *Geographical Analysis*, 27: 93-115.
- Fotheringham, A. S., Brunson, C. and Charlton, M. 2000. *Quantitative Geography: Perspectives on Spatial Data Analysis*, Sage Publications.
- Moran, P.A.P. 1948. The interpretation of statistical association: LISA. *Geographical Analysis* 27: 93-115.
- Tobler, W. R. 1966. Notes on the analysis of geographical distributions. *Michigan Inter-Univ. Commu. of Math. Geographers*. 8: 1-13.

日本人口学会研究企画委員会 (2022～2023 年度)

石井 太 (委員長, 慶應義塾大学)

中澤 港 (委員, 神戸大学)

堀口 侑 (幹事, 慶應義塾大学・院)

日本人口学会 報告書

人口学研究における R の活用

2024 年 3 月 31 日発行

編集 日本人口学会研究企画委員会 (2022～2023 年度)

発行 日本人口学会 Population Association of Japan

日本人口学会事務局

〒112-0012 東京都文京区大塚 5-3-13 小石川アーバン 4F

一般社団法人学会支援機構内

E-mail: paoj@asas-mail.jp

<http://www.paoj.org>
