Rによる比較的高度なデータ解析入門
—因子分析、SEM、応用回帰分析—
（「医療保健統計学・疫学特講II」講義テキスト）

神戸大学大学院保健学研究科教授：中澤　港
<minato-nakazawa@people.kobe-u.ac.jp>

3 August 2024

rev. 0.9.9 奥村晴彦・黒木裕介『改訂第 6 版 LaTeX2e 美文書作成入門』技術評論社を参照し、相互参照と索引を付けている途中だがとりあえず公開する (Aug 7, 2015)

rev. 0.9.9.1 R-3.2.2 がリリースされたので情報更新し、構造方程式モデリングの結果をプロットする semPaths のオプション指定について追記した (Aug 20, 2015)

rev. 0.9.9.2 3 村落の身長と体重のプロット例に coplot() と dataEllipse() を追記した（Aug 21, 2015）

rev. 0.9.9.3 R 研究集会で発表したのに伴って内容更新（Dec 6, 2015)

rev. 0.9.9.4 タイポを修正（Dec 9, 2015)

rev. 0.9.9.5 因子分析とマルチレベルモデルについて大幅更新（August 6, 2016）

rev. 0.9.9.6 傾向スコアと操作変数法について若干追記（August 1, 2017）

rev. 0.9.9.7 講義中に発覚したいくつかのミスを修正。最後のところを若干加筆（August 8, 2017）

rev. 0.9.9.8 lavaan パッケージで構造方程式モデルの当てはめをしたときの内的一貫性と収束的妥当性の指標値の求め方を追記（December 27, 2018）

rev. 0.9.9.9 新年度講義のためにいろいろアップデート（May 4, 2019）

rev. 1.0 講義中に見つかった修正点への対処。構成の一貫性を図って内容取捨選択（August 3, 2019）

rev. 1.0.1 DID についての記述を追加（July 24, 2020）

rev. 1.0.2 いろいろ古くなっていた情報をまとめて削除（August 1, 2020）

rev. 1.0.3 ソフトのバージョンを最新版に（August 4, 2021）

rev. 1.0.3.1 ソフトのバージョンを最新版に（July 25, 2022）

rev. 1.0.3.2 ソフトのバージョンを最新版に（July 26, 2023）

rev. 1.0.3.3 ソフトのバージョンを最新版にし、リンクを修正（August 3, 2024）

Contents

1 この講義を受けるための準備 1

1.1 R のインストール 2
1.1.1 ダウンロードとインストール 2
1.2 RStudio のインストール 3
1.3 パッケージのインストールと管理 5
1.4 R の基本 6
1.4.1 R の環境設定とコンソール操作 6
1.4.2 R のオブジェクト 8
1.4.3 R の基本文法 12

2 データの種類と目的による統計解析のパースペクティヴ 15

2.1 質問紙調査で得られるデータについて 15
2.1.1 調査項目の選択に当たって注意すべき点 15
2.1.2 用語上の注意 16
2.1.3 文章上の注意 16
2.1.4 質問の仕方のタイプ 17
2.1.5 回答形式のタイプ 17
2.1.6 スコア化のいろいろ 18
2.1.7 質問票の流れとレイアウト 21
2.1.8 質問紙調査データの解析のパースペクティブ 21
2.2 実験によって得られる測定値について 22
2.3 健診など調査によって得られる、測定値とカテゴリの複合データについて 22

3 R によるデータの前処理 25

3.1 データを積む 25
3.2 表の操作 27
3.3 再コーディングと文字列操作 30
3.3.1 データの再コーディング 30
3.3.2 文字列操作 30

4 R による多様な作図技法 33

4.1 作図の基本プロセス 33
4.2 日本語を扱う上での注意点 34
4.3 メイングラフ描画関数のいろいろ 35
4.4 具体的なグラフの作り方 36
4.4.1 群ごとにプロット記号を変えた散布図を描く 36
4.4.2 都道府県別生命表からの図示 40
4.4.3 時系列の２つの変数の関係 44

5 因子分析 **49**
5.1 因子分析と主成分分析 49
5.1.1 主成分分析とは？ 49
5.1.2 因子分析とは？ 49
5.2 主成分分析の基本的な使い方 50
5.2.1 利用例 1 51
5.2.2 利用例 2 53
5.3 因子分析の基本的な使い方 59
5.4 因子分析の基本モデル 60
5.5 いくつの因子を推定すべきか？ 61
5.6 因子分析の適切性をチェックする 61
5.7 R で因子分析を実行するための関数 63
5.8 エコポイントデータを使った分析例 64
5.8.1 クロンバックの α 係数の計算 65
5.8.2 探索的因子分析を試してみる 67

6 構造方程式モデリング **71**
6.1 sem の基本 72
6.2 エコポイントチェックデータへの適用例 73
6.2.1 lavaan でやってみる 76
6.3 John Fox 教授のテキストを参考に sem パッケージを使う 81
6.3.1 典型的な構造方程式モデル 82
6.3.2 観測変数がカテゴリ変数である例 85

7 応用回帰分析とマルチレベル分析 **89**
7.1 多変量回帰分析 89
7.2 非線形回帰分析 92
7.2.1 用量反応関係の解析 94
7.3 マルチレベル分析 101
7.3.1 マルチレベル分析の要点 102
7.3.2 例 1：多施設介入試験の分析 103
7.3.3 例 2：動物実験 109
7.3.4 例 3：職域サポート 114
7.3.5 例 4：R 組み込みデータから 119
7.4 傾向スコアを用いたモデル推定、DID、操作変数法 120
7.4.1 DID 121
7.4.2 操作変数法 123
7.4.3 二段階最小二乗法の関数 tsls() による操作変数法 123
7.4.4 ivreg による操作変数法 124

8 文献・サイト **127**
8.1 R について 127
8.1.1 概要を知るために 127
8.1.2 リファレンス 128
8.2 因子分析について 128
8.3 構造方程式モデリングについて 128
8.4 マルチレベルモデルについて 128
8.5 傾向スコアと操作変数法について 129

Chapter 1

この講義を受けるための準備

この講義の目的は、医学・保健学分野で頻繁に用いられるけれども比較的高度な統計手法を解説することである[1]。実際にそれらの解析を自分のデータについて実行できるようになることが到達目標なので、統計解析ソフトを使って解析する事例を提示する。なお、高度な解析を説明することから、統計解析の基礎知識は有していることを前提とし、基本的な用語などは説明しないので、それらについて自分の知識に不安を感じたら、統計学の辞書や教科書で確認されたい（例えば、Graham Upton・Ian Cook 著、白旗慎吾監訳、内田雅之・熊谷悦生・黒木学・阪本雄二・坂本亘・白旗慎吾訳『統計学辞典』共立出版、2010 年など）[2]。

統計解析ソフトには、SAS、SPSS など古くからよく知られている（とはいえ、現在の SAS や SPSS は、30 年前に大型計算機やミニコンで使っていたものとは、ほぼ別物と思う）一方で高価なものもあるが、(1) フリーソフトであるため、いつでも誰でも自分のコンピュータで使えて、(2) 必要な解析手法はほぼ網羅されていて（本体に含まれていなくても、CRAN というサーバから追加パッケージという形でインストールできることが多い）、(3) Nature、Science、Cell、ProNAS のような一流誌でも解析結果を受け付けてくれるほど信頼性が高い、といった理由から、この講義では R を用いる。R は Windows でも MacOS でも Linux でも動作するので、各自自分のコンピュータにインストールして講義に持参すると良い[3]。

R を操作する GUI 環境として、RStudio というフリーソフトが便利なので、必須ではないが RStudio も合わせてインストールされたい[4]。

なお、最近では、jamovi というフリーソフト[5]も計算のバックボーンが R であり、R

[1]集中講義の一部は実験データの解析やオミックス解析について扱われるが、その部分はこのテキストではカバーしていない。

[2]自著である『R による統計解析の基礎』（https://minato.sip21c.org/statlib/stat-all-r9.pdf）と『R による保健医療データ解析演習』（https://minato.sip21c.org/msb/medstatbookx.pdf）は、出版社（ピアソン桐原）の和書出版に関する方針変更により絶版になったので全文 pdf で公開しており、参考になると思う

[3]それほど高いスペックを必要としないので、安いノートパソコンでも十分に動作する。ただし講義室には十分な数の電源タップはないので、バッテリーは十分に充電してから講義に出席する方が安全である。なお、オンライン講義の場合はこの点は気にしなくて良い。

[4]「保健学共通特講 IV, VIII」https://minato.sip21c.org/ebhc/（テキストもこの URL からダウンロードできる）では R をメニューから操作するだけで医学統計の基本的な—といっても、t 検定とかカイ二乗検定とか作図とか集計だけでなく、反復測定分散分析や生存時間解析やメタアナリシスまで含む—分析がほとんどできてしまう EZR という追加パッケージ（自治医大の神田善伸教授が開発されていて、https://www.jichi.ac.jp/saitama-sct/SaitamaHP.files/statmed.html から入手可能）を使ったが、本講義で説明する高度な手法はメニューから選択するだけでは無理なので、このようなスタイルを取る。

[5]https://www.jamovi.org/ から入手可能。R 同様に Windows でも MacOS でも Linux でも動作し、2021 年 7 月 12 日にバージョン 2 がリリースされてから変数操作が格段に容易になり、2022 年 2 月 8 日にリリースされた、バージョン 2.3.0 から多言語対応した。芝田征司さんによる『jamovi 完全攻略ガ

のコードを呼び出して使うことも可能でありながら、ほぼすべての操作を WYSIWYG で行え（しかもリアルタイムで結果に反映される）、本講義で説明する高度な手法も一部サポートされているので、どうしてもメニュー操作が良いという場合は試してみても良いだろう。2024 年 6 月 13 日にリリースされた 2.5.6 で Stata のデータファイル `.dta`、SPSS のデータファイル `.sav`、Excel のファイル `.xlsx` の読み込みが改善されたとのこと。開発の経緯[6]に書かれているように、直感的に使えるようにデザインされている。

1.1 R のインストール

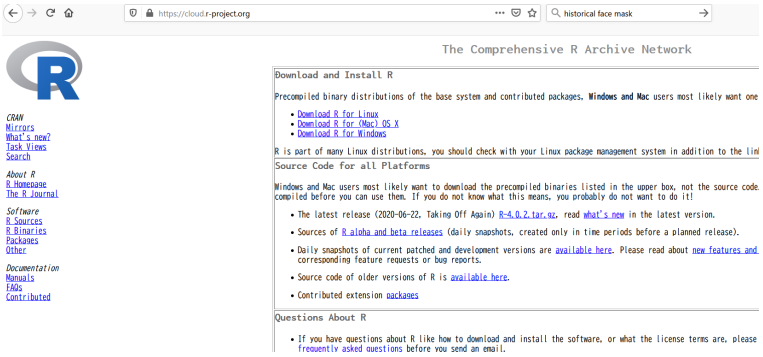


Figure 1.1: CRAN のクラウドミラー

1.1.1 ダウンロードとインストール

R 関連のファイルは、CRAN というサイトに集積されている。世界中に CRAN のミラーサイト（ダウンロードが集中することによるネットワーク負荷を軽減するために設置されている、元のサイトの内容のコピーであり、若干のタイムラグはあるが自動的に更新される）があって、ネットワーク的に近いミラーサイトからダウンロードすることが推奨されている。

2024 年 8 月 3 日現在、クラウドサイト[7]からダウンロードすると良い。ブラウザで開くと、図 1.1 の画面が表示される。

Windows、MacOS、Linux など、自分の使っている OS に応じてリンクを辿ると（注：MacOS の場合、OS のバージョンに応じて利用できる R のバージョンも変わり、違うファイルをダウンロードしなくてはいけないので注意されたい）、必要なファイルをダウンロードすることができる。Windows の場合だと、図 1.2 の画面が表示され、`Download R 4.4.1 for Windows` のリンクをクリックすると `R-4.4.1-win.exe` をダウンロードできる[8]。

イド』(`https://bookdown.org/sbtseiji/jamovi_complete_guide/`) や、英語では “Learning Statistics with jamovi” (`https://www.learnstatswithjamovi.com/`) といった説明書も充実してきた。

[6] `https://www.jamovi.org/about.html`

[7] `https://cloud.r-project.org/`

[8] 3.5.3 から 3.6.0 のバージョンアップで、乱数生成のデフォルト動作が変わったため、`optim()` や `sample()` など乱数生成を伴う関数が返す結果も変わった。また、3.6.3 から 4.0.0 への移行で、いろいろ仕様が変わったので、古いコードはそのままでは動かないことがある。とくに注意すべきは、テキストファイルを読み込むときに、文字列を自動的にファクター型に変換しなくなったことで、3.6.3 までの動作にしたい場合は、`stringsAsFactors=TRUE` というオプションを `read.delim()` などの**中に**書く必要がある。R-4.1.0

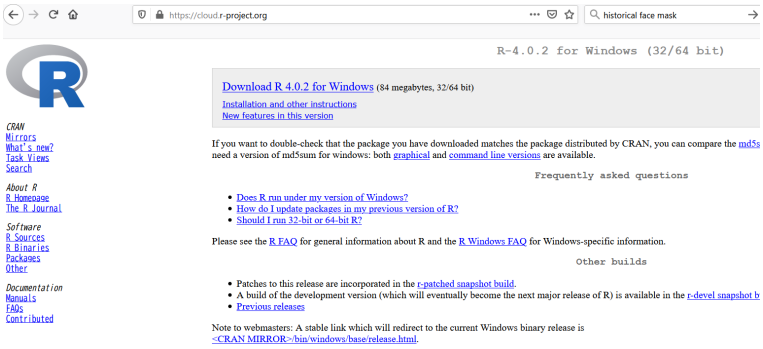


Figure 1.2: R for Windows のダウンロード画面

`R-4.4.1-win.exe` を管理者権限でダブルクリックするとインストールが始まる[9]。RStudio や Rcmdr や EZR などの GUI フロントエンドを使う場合、R 本体は MDI でなく SDI で動作する方が良いので、インストールオプションは手動で設定する方を選択し、SDI を選ぶべきである。

MacOS ユーザは群馬大学[10]青木繁伸教授の解説（`http://aoki2.si.gunma-u.ac.jp/R/begin.html`）を参照すると良い。Linux ユーザは RjpWiki の解説（`http://www.okadajp.org/RWiki/?R%20%E3%81%AE%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB#if8c08b4`）が参考になる。

1.2 RStudio のインストール

RStudio は R 本体とは開発元が異なり、トップページの URL は、`https://www.rstudio.com/` である。著名なパッケージ開発者が何人も参加しており、`animation` や `knitr` の開発で知られる Dr. Yihui Xie や、`ggplot2` や `plyr`（と `dplyr`）、`devtools` を含む `tidyverse` の開発で一世を風靡している Dr. Hadley Wickham（彼のパッケージは新しい文法を持ち込むので、そのパッケージ群に魅入られた人々はハドラーと呼ばれることがある）が中心になっていて、センスあふれるインターフェースにもファンが多い（とくにデータビューアが見やすい）。

トップページから `Download RStudio` というボタンをクリックすると、Desktop 版か Server 版か、また Open Source Edition か Commercial Edition かについて、それぞれの説明とともにダウンロードボタンが表示される。パソコンで使うには、通常、Desktop 版の Open Source Edition で十分なので、`DOWNLOAD RSTUDIO DESKTOP` というアイコンをクリックする。図 1.3 が表示されるので、そこから自分の OS 用のバージョンの Installer をダウンロードする。

からパイプ処理などいくつかの新機能が導入されたが（それまでも `tidyverse` をインストールすれば`%>%` は使えたが、`|>`が標準で実装された）、このテキストでは新機能は使っていない。なお、Windows ユーザーに限った話だが、R-4.2.0 から、Windows 版もデフォルトの日本語文字コードが CP932 ではなく UTF-8（ユニコード）に変わり、MacOS 版や Linux 版と同じになった点に注意が必要である。R-4.1 までに書いた古いコードやネットからダウンロードしたコードは文字化けして動作しない可能性がある。その場合は、エディタなどで開き、UTF-8 として保存すれば問題は解決するはずである。なお、RStudio では Windows 環境でもかなり前から UTF-8 がデフォルトになっていたので、この問題は生じない可能性が高い。

[9] **（注意）** RStudio も R も、ユーザ名に漢字やカタカナなどの 2 バイト文字が入っていると正しく動作しない場合があるので、**Windows のユーザ名としては、半角英数字だけからなり、スペースも含まないものを強く推奨**する。

[10] 2016 年 3 月に定年退職されたが、少なくとも 2 年は群馬大学内のサイトは維持されると伺った。2024 年 8 月 3 日現在、まだ残っているのは素晴らしいと思う。

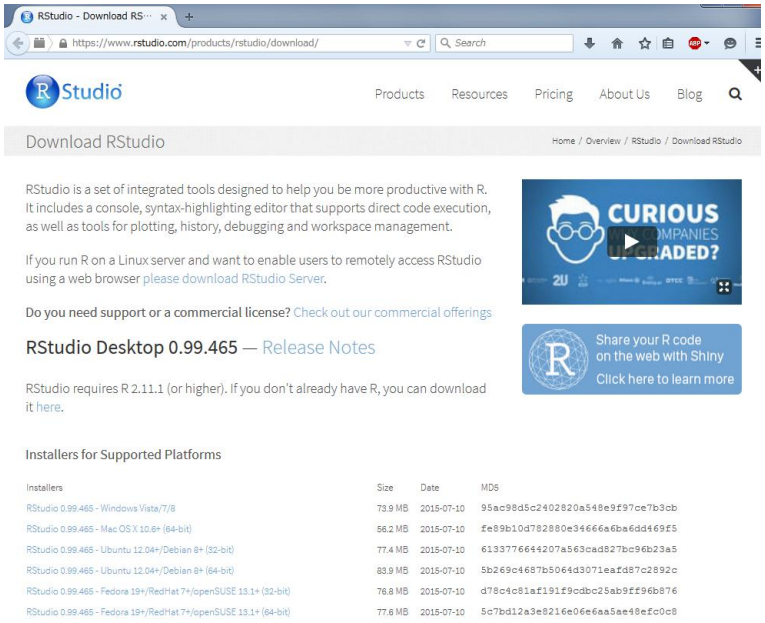


Figure 1.3: RStudio のダウンロード

Windows の場合、2024 年 8 月 3 日現在、`https://posit.co/download/rstudio-desktop/` から、`RStudio-2024.04.2-764.exe`（2024 年 6 月 10 日にリリースされた、コード名 Chocolate Cosmos は、玉置浩二のアルバムではなく、コスモスの一種であるチョコレートコスモスという植物である）がダウンロードできるので、それを実行すれば図 1.4 が表示される。後はメッセージに従って「次へ」ボタンをクリックしていけばインストールが完了する。Windows 10 や 11 で普通にインストールすると、R 本体とは違って、デスクトップに起動アイコンはできないので、頻繁に使う場合は、スタートメニューから RStudio のアイコンを探し、右クリックして「スタートにピン留めする」か「タスクバーにピン留めする」を選んでおくと良いだろう。または、“C:/Users/[ユーザ名]/AppData/Roaming/Microsoft/Internet Explorer/Quick Launch” の中に “C:/Program Files/RStudio/bin/rstudio.exe” へのショートカットを作って、クイックランチャーをタスクバーに常時表示させる設定にするのも便利である。筆者は CLaunch というランチャー[11]をインストールして登録している。

通常は、研究内容ごとにプロジェクトを設定するのが RStudio の使い方になる。新規にプロジェクトを始める場合、`File` メニューから `New project` を選び、既にデータや R コードが置かれているフォルダがあれば、`Existing Directory` を選んでそのフォルダを選べば良いし、まだ何もないところから始めるには、`New Directory` で新たにフォルダを作ればよい。

どちらの場合でもそのフォルダに拡張子 `.Rproj` のファイルができるので、次からはそのファイルをダブルクリックするだけで RStudio が起動し、しかも呼び出し元フォルダが R の作業フォルダになる[12]。

なお、RStudio はプロジェクト単位で文字コード指定ができる。MacOS の日本語文字コードはデフォルトで UTF-8、Windows は CP932（Shift-JIS）だが、RStudio の

[11] `https://hp.vector.co.jp/authors/VA018351/claunch.html`

[12] もっとも、プロジェクトを閉じないで RStudio を終了した場合、次にスタートメニューなどから RStudio を起動すると、最後にアクティブであったプロジェクトが開いた状態で起動するので、わざわざ作業フォルダの `.Rproj` をダブルクリックしにいく必要はない。

Figure 1.4: RStudio のインストール

メニューバーから、Tools ＞ Project Options ＞ Code Editing ＞ Text Encoding で、そのプロジェクトのマルチバイトコードを指定できるので、ここで UTF-8 にしておけば、同じ R コードを Windows と Mac で共用できる。R-4.1 系までは、Windows で Rterm でバッチ処理したり、R 本体のスクリプトエディタでコード編集するには日本語文字コードを Shift-JIS にする必要があり、RStudio でそのフォルダのファイルを編集する場合に文字化けを避けるためは、そのフォルダにプロジェクトを作り、この設定を CP932 に指定するという弥縫策を取らざるを得なかったが、R-4.2.0 から Windows 版の日本語文字コードも UTF-8 になったので、基本的にここをいじる必要は無くなった。

1.3 パッケージのインストールと管理

R の大きな特徴として、本体とは別に世界中のユーザが作ったパッケージをインストールすることによって、特殊な統計解析や新しく開発された手法を次々に機能追加できる点が挙げられる。最近では GitHub にしかないパッケージも少なくないが（理由は `https://r-pkgs.org/release.html` に書かれている通り、CRAN で公開してもらえる基準を満たすのが結構面倒なこともあるが、`https://kbroman.org/pkg_primer/pages/github.html` に書かれているように、Hadley Wickham 師の `devtools` パッケージを使うと、GitHub を利用することがかなり容易になるからだと思われる）、主要なパッケージは R 本体と同じく CRAN からインストールできる。パッケージ名がわかっていれば、基本的には R コンソールから `install.packages()` を使うだけで済む。

例えば、CRAN から `Rcmdr` パッケージをダウンロードしてインストールするには、

```
install.packages("Rcmdr", dep=TRUE)
```

とする。最初のダウンロード利用時には、パッケージをどのミラーサーバからダウンロードするかを聞いてくるので、通常は国内のミラーサーバを指定すればよいだろう。筆者は統計数理研究所のサーバを利用することが多い。`dep=TRUE` は dependency（依存）が真という意味である。`Rcmdr` が依存している（内部で使われている）`Rcmdr`

以外のパッケージも自動的にダウンロードしてインストールしてくれる。なお、TRUE は T でも有効だが、誤って T を変数として別の値を付値してしまっていると、意図しない動作をしてしまい、原因を見つけにくいバグの元になるので、できるだけ TRUE とフルスペル書いておくことが推奨されている。

パッケージを CRAN に登録する仕組みとして、.tar.gz 圧縮したソースコードを web サイトからアップロードして主として形式についての審査を受け、通ればシステム側でコンパイルしてバイナリ版もアップロードしてくれるため、バイナリとソースで最新バージョンが違う場合があり、ユーザ側でソースをダウンロードしてインストールするにはコンパイラなどのツールが必要である。そのためには、Windows 環境では Rtools を[13]、MacOS X でもいくつかの指定されたツールを、予めインストールしておく必要がある。2024 年 8 月 3 日現在、R-4.4.0 以降に対してインストールが推奨されている Rtools のバージョンは 4.4 である。

RStudio の Tools メニューを使うと、パッケージのインストールやアップデートがグラフィカルなユーザーインターフェイスでできるので便利である（途中まで打てば候補が表示されるため、パッケージ名をうろ覚えのときに助かる）。

1.4 R の基本

三重大学奥村晴彦教授の解説「R の初歩」（https://oku.edu.mie-u.ac.jp/~okumura/stat/first.html）が大変参考になるので、是非参照されたい。ここでは、ごく簡単に最低限の情報を書いておく。なお、以下の解説は Windows 版による。基本的に Linux 版でも Mac OS X 版でも大差ないはずだが、使えるグラフィックデバイスやフォントなどが多少異なるので、適宜読み替えられたい。なお、以下の本文中、\記号は ¥ の半角と同じものを意味する。

1.4.1 R の環境設定とコンソール操作

インストールが完了するとデスクトップ（インストールオプションで指定すればクイック起動メニューにも）R の起動アイコン（起動用のショートカット）ができている。素の R コンソールを使う場合は、このショートカットアイコンを右クリックしてプロパティを選択し、「作業フォルダ (S)」に作業ディレクトリを指定しておくとよい。環境変数 R_USER も同じ作業ディレクトリに指定するとよい。ただし、システムの環境変数または作業ディレクトリに置いたテキストファイル .Renviron に、R_USER="c:/work"などと書いておくと、それが優先されるので、注意が必要である。また、企業ユーザなどで proxy を通さないと外部のネットワークと接続できない場合は、Windows のインターネットの設定できちんと proxy を設定した上で、起動アイコンのプロパティで、「起動コマンドのリンク先」末尾に--internet2 と付しておく必要がある[14]。また、日本語環境なのに R だけは英語メニューで使いたいという場合は、同じく起動アイコンのプロパティの「起動コマンドのリンク先」末尾に LANGUAGE="en"と付しておけばいいし、R のウィンドウが大きな 1 つのウィンドウの中に開く MDI ではなく、別々のウィンドウで開く SDI にしたければ、ここに--sdi と付しておけばいい。

以上の準備の後に起動アイコンをダブルクリック（設定によってはクリック）すれば、R コンソールウィンドウが開き、作業ディレクトリの .Rprofile があればそれが実行され、保存された作業環境 .RData があればそれが読まれて、図 1.5 が表示されて入力待ちになる。

[13] https://cran.r-project.org/bin/windows/Rtools/

[14] インストール時に指定しておけば、自動的にそうなっているはずである。なお、2015 年 8 月 14 日リリースされた 3.2.2 からは、この動作がデフォルトになったので、このオプション指定は不要になった。

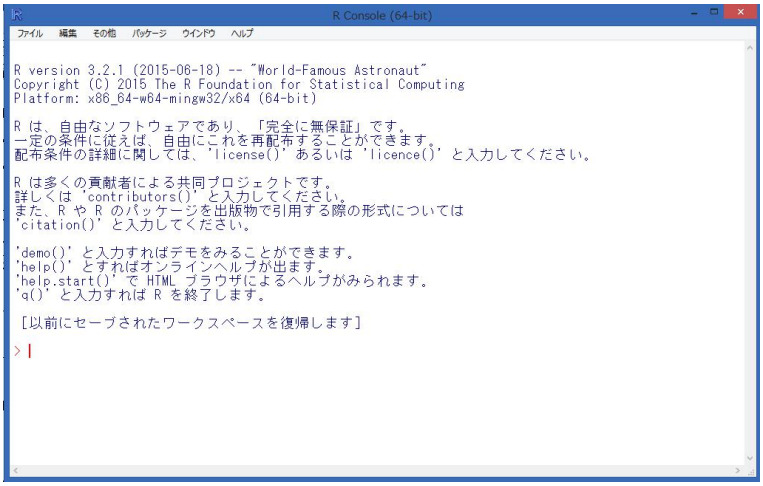


Figure 1.5: R コンソール

この記号>をプロンプトと呼ぶ。R コンソールへの対話的なコマンド入力は、基本的にプロンプトに対して行う。閉じ括弧を付け忘れたり命令や関数の途中で改行してしまった場合はプロンプトが継続行を意味する+となることに注意されたい。なお、Windows では、どうしても継続行状態から抜けられなくなってしまった場合、ESC キーを押すとプロンプトに戻ることができる。

入力したコードは、「ファイル」メニューの「履歴の保存」で保存でき、後で「ファイル」の「R コードのソースを読み込み...」で呼び出せば再現できる。プロンプトに対して source("プログラムファイル名") としても同じことである。但し、Windows ではファイルパス中、ディレクトリ（フォルダ）の区切りは/または\\で表すことに注意されたい[15]。できるだけ１つの作業ディレクトリを決めて作業すべきである。既に述べたように、RStudio はプロジェクトごとに作業ディレクトリを指定できる点が便利である。

一度実行したコードを呼び戻したいときは、キーボードの↑を押せば良い。←などを使って部分的に編集し、再実行させることもできる。プロンプトに続けて `history()` と打って実行すれば、それまでに打ったコマンド履歴を示すウィンドウが出現する。なお、R をインストールしたフォルダの `bin` にパスを通しておけば、Windows 7/8/8.1/10/11 のコマンドプロンプトで R と打っても、R を起動することができる。この場合は、コマンドプロンプトが R コンソールの代わりにシェルとして動作する。

ただし、このように対話的に実行するよりも、必要な操作をスクリプトとして予め書いておき、全部または選択した一部を実行する方が便利である。つまり、「ファイル」メニューの「新しいスクリプト」を選んで表示されるスクリプトエディタにコードを書いておき、実行したい範囲を選んでから、「編集」の「カーソル行または選択中の R コードを実行」を選べば実行される。スクリプトエディタに書いたコードは「ファイル」「保存」でテキスト形式で保存でき、R コンソールの「ファイル」「スクリプトを開く」から呼び出すことができる。

[15] バックスラッシュ文字は、日本語キーボードでは¥によって入力できる。

1.4.2 R のオブジェクト

R の使い方の基本は、**データをオブジェクトとして定義（付値）し、オブジェクトに対して描画したり分析する関数を適用**することである。大雑把に考えれば、オブジェクトは変数と考えても良い。オブジェクト名としては漢字も使えるが、半角英数字を使うのが普通である。注意しなくてはいけないのは、オブジェクト名において、半角と全角が区別されるのはもちろん、アルファベットの大文字と小文字も区別されることである。`Y` と `y` は別のオブジェクト名になる。基本的に関数名や予約語はすべて半角である。`hist(dat$HT,xlab="身長(cm)")` と書くべきところを、ｈｉｓｔ（ｄａｔ＄ＨＴ，ｘｌａｂ＝”身長（ｃｍ）”）と書くと、全角のｈｉｓｔという関数などないのでエラーが起こる。コマンド中に１つでも全角が混ざっているとうまくいかない。例えば、`hist(dat$HT、xlab="身長(cm)")` では、HT の次のコンマが全角になっているため、何も表示されない（見つけにくいエラーなので注意）。そのセッションの中で既に定義されているオブジェクトの一覧は、`ls()` と打てば表示される。オブジェクトにはいくつかの種類があり、代表的なものは次のようにまとめることができる。まずは単一の要素（長さ 1 のベクトルともいえる）であるスカラー型についてまとめる。

スカラー型オブジェクトのいろいろ

上述の通り、R におけるスカラー型は、内部的には長さ 1 のベクトルである。しかしここではデータ型の説明として、さまざまなスカラーの性質を記す。

空値 `NULL` 存在しないことを意味する値。欠損を意味する `NA` とは異なる。

論理値 `TRUE` または `FALSE` のどちらかの値をとる。正誤を示す。`as.logical()` で型変換。式は論理値を返す。例えば、`5<4` は `FALSE` を返すし、`1+2==3`（==であることに注意。=だと代入になってしまう）は `TRUE` を返す

整数 –1564, 0, 1, 45671, ... 桁が大きすぎると実数化される。`as.integer()` で型変換。例えば `as.integer(TRUE)` は 1。なお、`4:8` のように整数を:（コロン）でつなぐと 1 刻みの整数ベクトルになる（`2:4` は `c(2, 3, 4)` と同じ）。

数値（実数） –0.1, `pi`, `sqrt(2)`, `1e+14`, ...`as.numeric()` で型変換。

複素数 `1+1i`, `0+0i`, ... `as.complex()` で型変換。

文字列 `"abc"`, `LETTERS[24:26]`, `"1"`, ... `as.character()` で型変換。数字でも二重引用符で括れば文字列型。文字列オブジェクトは、"身長のヒストグラム"のように半角の二重引用符[a]で括る。二重引用符なしだと変数名として扱われてしまう。グラフの表題をつけるときなど、表題文字列は半角二重引用符で括らないと表示されない。

ファクター ファクター型化した文字列。要素数が限られている文字列を、内部的にアルファベット順に数値化して保持するための型（なので、スカラー型としてはあまり意味がない。複数の要素があってこそ意味がある型である）。いわゆるカテゴリ変数を表現するために使う。`factor()` で定義するか、`as.factor()` で型変換する。後の説明を参照されたい。

[a]日本語 Windows 環境の場合、Shift キーを押しながら数字キーの 2 を押しながら入力する。英語ではダブルクォーテーションマークという

おそらく、型の中で一番わかりにくいのがファクター型であろう。例えば ABO 式血液型には、通常、O 型とか A 型のような 4 つのカテゴリがある。このとき O 型などを水準といい、血液型は 4 つの水準からなるファクター型のオブジェクトといえ

る。タブ区切りテキストファイルがあって、変数名を表す 1 行目が"bloodtype"である列の値が、"A"、"B"、"O"、"AB"などとなっていれば、それを read.delim() で読み込むと、自動的に bloodtype という変数はファクター型になり、水準 1 が"A"、水準 2 が"AB"、水準 3 が"B"、水準 4 が"O"とアルファベット順に設定されるので、テキストファイル中の文字列をファクター型にしたくない場合（文字列型のままにしておきたい場合）は、読み込む前に

```
options(stringsAsFactors=FALSE)
```

を実行しておく……というのが、R-3.6.3 までの仕様だったが、R-4.0.0 からはそちらがデフォルトになった。R-4.0.0 以降で、R-3.6.3 までと同じく、文字列を読み込んだときに自動的にファクター型に変換したい場合は、read.delim() や read.csv() のオプションとして stringsAsFactors=TRUE を指定することが推奨されている（options() での指定もまだ有効だが、将来廃止される可能性がある）。このファクター型変数 bloodtype の水準をアルファベット順でなく、指定した順序にしたい場合は factor() を用いる。例えば、A, O, B, AB という順序にしたければ次のようにする。ただし、これは表やグラフなどの結果で水準が表示される順序を決めているだけで、順序付きファクター型ではないことに注意されたい。

```
# 乱数を使った結果を再現可能にするため初期値を指定する
set.seed(123)
# options(stringsAsFactors=FALSE)
# ↑ R-3.6.3 まではこれをしないと次の bloodtypeC はファクター型
# になっていた（ただし水準は A, AB, B, O の順）
# ファイルから読む代わりに 100 人ランダムサンプル、文字列型
bloodtypeC <- sample(c("A","O","B","AB"), 100, rep=TRUE)
# 水準の順序を指定してファクター型に変換
bloodtype <- factor(bloodtypeC, levels=c("A", "O", "B", "AB"))
bloodtype <- factor(sample(1:4, 100, rep=TRUE),
 labels=c("A", "O", "B", "AB")) # ←この指定がお薦め
```

ファクター型の変数を as.integer() に渡すと水準の順番が返ってくる。水準に明示的な順序がある場合は、順序付きファクター型という型にすることもできる。factor() の中で ordered=TRUE オプションを付けるか、最初から factor() でなく ordered() で設定するか、as.ordered() で型変換する。

解析をしていると、連続変数をカテゴリ化したいことも良くある。例えば整数型や数値型で入力されている身長のオブジェクト height を考える。ここでは平均 160cm、標準偏差 5cm の正規分布に従う乱数で 20 人分の架空の身長データを作ってみよう。それに続いて 150cm から 180cm まで 5cm 刻みのファクター型のオブジェクト hc を作りたいときは、cut() 関数を使って以下のようにすれば良い。

```
set.seed(123) # 乱数を使った結果を再現可能にするため初期値を指定する
height <- round(rnorm(20, 160, 5), 1) # 小数点以下 1 桁に丸める
hc <- cut(height, seq(150, 180, by=5))
print(data.frame(height, hc))
```

cut(連続変数, 区間ベクトル) は、任意の連続変数を区間ベクトルに従ってカテゴリ変数に変換する関数である。区間ベクトルを定義する seq(最小値, 最大値, by=区間長) は、最小値から最大値までを区間長ごとに区切った区間ベクトルを生成する。by=の代わりに len=区間数を使えば、最小値から最大値までを区間数分の等間

隔の区間に分割してくれる。hc は最小区間が (150, 155]、最大区間が (175, 180] のファクター型の変数になる。デフォルトでは区間の境界が「〜を超えて〜以下」なので、日本風に「〜以上〜未満」にしたいときは、cut() 関数の中で、オプションとして right=FALSE を指定する。なお、hco <- ordered(hc) とすれば、順序付きファクター型になる。

R のオブジェクトは単純なスカラーだけではなく、ベクトル、行列、テーブル、リスト、データフレームなど、さまざまな構造をもつことができる。S4 クラス、S5 クラスなどのオブジェクトではスロットをもつこともでき、地図データオブジェクトは S4 クラスなのだが、このテキストでは触れない。

ベクトル

上記スカラー型の集合がベクトルである。c() の中に半角コンマ, で区切って要素を並べる。与えた要素が同じ型でない場合は自動的に型変換される。例えば c(NULL, FALSE, 11, 8.23, 5+2i, "statistics") とすると、すべて文字列扱いになる。既に定義したベクトル x と y があれば、c(x, y) で 1 つのベクトルにできる。要素の参照は [] で行う。例えば x <- 2:4 であるときに x[3] とすれば、x の 3 番目の要素である 4 が返ってくる

行列

次元のあるベクトルが行列である。matrix(X, NROW, NCOL) とすれば、ベクトル X を NROW 行、NCOL 列の 2 次元行列にできる。オプションとして byrow=TRUE を付けない限り、ベクトルは列方向に並べられる。即ち、最初の要素が 1 行 1 列、次の要素は 2 行 1 列、...、NROW 行 1 列、1 行 2 列、... と並ぶ。NROW や NCOL は 1 以上の整数であり、1 列しかない行列や 1 行しかない行列も定義できるが、次元があるためベクトルとは異なる。要素の参照は [,] で行う。例えば、matrix(1:9, 3, 3)[3, 1] は、3 行目 1 列目の要素なので 3 となるし、matrix(1:9, 3, 3)[1, 3] は 1 行目 3 列目の要素なので 7 になる。ただし比較すると要素ごとの比較になるので、例えば 1:4 == matrix(1:4, 1, 4) とすると、行列のすべての要素について TRUE が返ってくる。また、array(X, dim=c(NROW, NCOL, NSTRATA)) とすれば、ベクトル X を NROW 行、NCOL 列の行列を NSTRATA 層に積み重ねた 3 次元行列を定義することができる。層別したクロス集計表からマンテルヘンツェルの要約オッズ比を計算する関数 mantelhaen.test() には、3 次元行列の形で層別クロス集計表を与えるのが普通である。同じ長さのベクトルならば連結して行列にすることができ、それぞれ長さ 5 のベクトル X と Y があるとき、cbind(X, Y) とすれば 1 列目が X、2 列目が Y からなる、5 行 2 列の行列ができる。rbind(X, Y) とすれば、1 行目が X、2 行目が Y からなる 2 行 5 列の行列ができる。行列は t() によって転置できるので、cbind(X, Y) と t(rbind(X, Y)) は一致する。

テーブル

原則として整数を要素とし、行や列や層に名前がついている、特殊な行列をテーブルという。table() 関数や xtabs() 関数にカテゴリ変数を与えてできるクロス集計表は table 属性をもつ。matrix() や array() で定義した行列 Z にテーブル属性を与えるには、attr(Z, "class") <- "table"とする。テーブルを plot() 関数に渡すと、自動的にモザイクプロット（mosaicplot()）が呼び出されるが、これは R がオブジェクトオリエンティッドな言語であり、plot() や print() や summary() などの関数が、与えられるオブジェクトの属性によって、その種類にふさわしい動作をするように定義されているためである（これらは総称的関数と呼ばれ、print.table() のように、関数名の後にピリオドとオブジェクトクラスが付された個別動作の関数が定義されているのが普通である）。

リスト

あらゆるオブジェクト（リスト自身を含む）をリストとして束ねたもの。list() の中に半角コンマ , で区切ってオブジェクトを並べて作る。オブジェクトは入れ子にすることもできる（リストのリストとか、行列のリストも可能）。名前をつけることもできる。例えば X <- list(A=1:3, B=c("あ", "い"), C=TRUE) とすれば、1 から 3 までの整数のベクトル A、“あ” という文字列と “い” という文字列 2 つからなる文字列ベクトル B、TRUE（真値）である論理値スカラー C を項目としてもつリストを定義し、オブジェクト X に付値することができる。各項目の参照は$または [[]] で行う。このオブジェクト X の項目 B を参照するには、X$B または X[[2]] とする。いったん参照された項目 B は普通の文字列ベクトルなので、その 2 番目の要素"い"の参照は、X$B[2] または X[[2]][2] で可能

データフレーム

データフレームは、要素数がすべて等しいという点だけが特殊なリストである。行列との違いは、型の異なる要素を含むことができる点であるが、必ず 2 次元の表形式なので、行列と同じやり方で要素を参照することが可能である（例えば、データフレーム Y の 2 つめの変数 A の 3 番目の要素を参照したければ、リストとしては Y[[2]][3] または Y$A[3] とするのが基本だが、Y[3, 2] または Y[3, "A"] とすることができる）。read.delim() で外部からタブ区切りテキストファイルを読み込むと、結果はデータフレームになる。行列型のオブジェクト Y をデータフレームにするには、as.data.frame(Y) とする。subset(Y, expr) 関数を使うと、データフレーム Y の条件式 expr を満たす行だけを抽出することができる。

オブジェクトの情報を得るには、mode() や str() を使うと便利である。簡単にまとめておく。

mode mode(x) でオブジェクト x の型を調べることができる。x がスカラーのときはもちろん x の型が返るが、ベクトルや行列の場合も要素はすべて同じ型なので要素の型が返ってくる。x がリスト（データフレームも含む）のときは list と表示される。

str str(X) とすることで X のデータ構造を返す。具体的にはオブジェクトの長さや変数ごとの型などが表示される。

length length(X) はオブジェクト X の長さを返す。長さとは、ベクトルなら要素数、リストならリスト項目数、データフレームなら変数の数を意味する。文字列の長さを返す関数は別にあって、nchar() である。つまり、length("happy") は 1 を返すが、nchar("happy") は 5 を返す。

names names() は、rownames()、colnames()、dimnames() とともに、スカラーに名前を付けたり、ベクトルやリストや行列やテーブルやデータフレームに含まれる変数名を参照したり、それらを付値によって改変する目的で用いる。オブジェクト x の値が 1 だとして、この x に、例えば"test"という名前を付けるには、names(x) <- "test"とする。ベクトルの場合、例えば、リンゴが 5 個、ミカンが 3 個、メロンが 2 個、葡萄が 10 個あることを表現したければ、x <- c(5, 3, 2, 10) としてから names(x) <- c("apple","orange","melon","grape") とすればよい。名前を付ける利点は、それによる参照ができるようになることで、この場合なら、メロンの個数を知りたいとき、x["melon"] という形で参照できる。行列またはデータフレーム X について、rownames(X) で X の行の名前を参照できるし、rownames(X) <- c("A", "B", ...) のようにすれば行名を付けることができる。colnames(X) で列名が参照でき、colnames(X) <- c("X", "Y", ...) で列名を付けることができる。

1.4.3 R の基本文法

最も基本的なコードを以下に示す。改行までが 1 つの関数または文として扱われる。

終了 q()

付値 <-

例えば、1、4、6 という 3 つの数値からなるベクトルを X というオブジェクト（変数）に保存するには次のようにする。

```
X <- c(1, 4, 6)
```

注釈 #より後は行の終わりまで注釈となり実行されない

区切り ; は改行の代わりになり、1 行の中に 2 つ以上の関数や文を書ける

ブロック {から}まではブロックとなり、間に改行があっても 1 つの塊として扱われる

関数の適用 関数にオブジェクト（とオプション）を与えると結果が返ってくる。例えば、上の X に対して、合計を計算する関数 sum() を適用するには、sum(X) とすれば、11 という結果が返ってくる。関数は入れ子にできるし、関数の結果をオブジェクトに付値することもできる。

定義 function()

複雑な計算を 1 つの関数として自分で定義することができる。関数定義は何行にも渡って行うことができ、最終行の値が戻り値となる。関数内の変数は局所化

されているので、関数内で変数に付値しても関数外には影響しない。関数内で変数の値を本当に変えてしまいたいときは、通常の付値でなくて、<<-（永続付値）を用いる。例えば、ベクトル X の平均と標準偏差を計算する関数 meansd() の定義は次の通り。

```
meansd <- function(X) { list(mean(X), sd(X)) }
```

ヘルプ ?

例えば、*t* 検定の関数 t.test() の解説をみるには、?t.test とする。見出し語が不明で説明文中に出てくる単語を検索したいときは ?? を使う。例えば、コクラン＝マンテル＝ヘンツェルの要約カイ二乗検定をする関数名を忘れてしまったときに、??Cochran と打てば、mantelhaen.test() という関数名が見つかる。

使用例 example()

多くの関数はヘルプに用例が含まれていて、example() で実行することができる。例えば example(lm) とすれば、線形回帰分析の関数 lm() の使用例が表示される

R は S 言語のサブセットの実装と言われている通り、S 言語の文法でループや条件分岐などの制御構造が書ける。簡単に説明しておく。

ループ for () { }によるのが普通である。例えば、

```
T <- 0
for (i in 1:3) {
 T <- T+i
}
```

と

すると、最初は 0 である T に 1、2、3 が順に足されて 6 に変わる。条件分岐によりループを途中で抜けたいときは break を使う。ループの終了条件を予め決められない場合は、while( ) { }を使うことができる。

条件分岐 if () { } else { }が条件分岐の基本形である。( ) 内に入れる条件文には次のようなものがある。条件文がベクトルならば最初の要素のみ使われることに注意。

```
if (A==B) {} # A と B が等しいとき{}内を実行
if (A>B) {} # A が B より大きいとき{}内を実行
if (A>=B) {} # A が B より大きいか等しいとき{}内を実行
if (A<B) {} # A が B より小さいとき{}内を実行
if (A<=B) {} # A が B より小さいか等しいとき{}内を実行
if (x %in% A) {} # x が A の要素に含まれれば{}内を実行
```

ベクトルの各要素に対して条件判定させ、新たなベクトルを作るには ifelse() を用いることができる。例えば、x <- c(1, 1, 2, 1, 2, 2, 1) であるとき、1 ならば"M"、2 ならば"F"に置き換えた新しい変数 y を作りたいときは、y <- ifelse(x==1, "M", "F") とする[16]。カテゴリ変数の再カテゴリ化も、ifelse と%in%を使うとやりやすい。

[16] もっとも、この場合は

```
y <- factor(x, labels=c("M", "F"))
```

とファクター化する方が普通である。

Chapter 2

データの種類と目的による統計解析のパースペクティヴ

2.1 質問紙調査で得られるデータについて

質問紙調査で得られるデータは、基本的にカテゴリデータである。知識・属性や行動・認識のどれを聞いているのかを明確にしなくてはいけない。知識を知るには正解・不正解が明確に定まるテストをするべきである。テストの得点は、正規分布に近ければ連続量として扱うことができる。属性や行動を調べるには、なるべく事実を紛れなく尋ねることができる質問を工夫せねばならない。年齢や睡眠時間のように連続量が得られる場合もあるが、属性や行動も、大抵はカテゴリデータとなる。

認識についてはリッカート尺度[1]を用いることが多い。何件法のリッカート尺度にするかは目的により変わるが、３件法（選択肢は、例えば｛１．同意できない、２．どちらでもない、３．同意できる｝などとなる）や、５件法（例えば｛１．まったく当てはまらない、２．どちらかといえば当てはまらない、３．どちらともいえない、４．どちらかといえば当てはまる、５．良く当てはまる｝などとなる）が良く用いられる。これら自体は順序尺度として分析するが、複数の似たような質問項目の合計得点を何かの尺度とする場合には、連続量として扱うことも多い。ただし、その場合は、それらの質問項目が単一の潜在因子を示すかどうかを、クロンバックの α 係数などによって確認しておく必要がある。α 係数が概ね 0.7 以上でなければ、単一の潜在因子としての信頼性は低いと考えられる。場合によっては因子分析を行い、潜在因子構造を捉え直す必要がある。

2.1.1 調査項目の選択に当たって注意すべき点

- 調査目的に関連した、あるいは作業仮説に関連した、必要十分な調査項目を含ませねばならない
- 一般には、調査目的になっている大きい主題をまずいくつかの次元に分解し、次にその各々をさらに細かいいくつかの次元に分解し、といった手続きを繰り返し、最終的に細分されたものが調査項目となり、見出し語あるいは質問の形で調査票に取り入れられる
- 何か興味があり価値がありそうに思われると何でも入れたくなるが、吟味が必要：

[1] Lickert scale：提示された文に回答者がどの程度同意できる（あるいは、当てはまる）かを何段階かの選択肢から選んで得点とする。

1. その項目でデータが得られるか？
2. データが得られるとしても分析に使えるか?[2]
3. 全体のバランスからみて重要度が低くないか？
4. 被調査者に抵抗や反感や困惑を起こさせないか？

- 必要最小限＋聞きにくいことを聞くための導入的な質問、他の質問の応答を確認するための限られた無駄な質問

2.1.2 用語上の注意

- 簡単明瞭かつ正確に被調査者が理解できること。例：年齢でも、ただ「おいくつですか？」という問いでは、満年齢か数え年かわからないし、年まででいいのか何ヶ月まで必要なのかという精度もわからない。被調査者の最低理解水準を基準にしても「わかる」ようにする

- 単純な日常会話的用語、副詞や代名詞に注意する。例：「どんな洋服を着ますか？」では、知りたいのが形なのか色なのか素材なのか商品価値なのかがわからないし、「それは何故ですか？」では「それ」が何を指しているか誤解される危険がある。

- 一般名詞と固有名詞に注意する。例：「新聞を何部とっていますか？」では新聞の種類がわからないし、部数を聞くときに客商売の接客用をふくめるかどうかも曖昧。

- 被調査者の社会階層や環境の違いによるイメージの違いに注意する。例：「風呂」といっても下宿している学生などでは銭湯をイメージするかもしれない。

- 難しい用語や専門用語は避ける。やむを得ず使う場合は明瞭な定義を与える。例：米国で行われた調査で、まったく架空の Metallic Metals 法について尋ねた結果、70%の被調査者が「それは連邦か州で調査すべきだ」と答えたという報告がある。被調査者は「ことばがわからない」とは言いたがらない。

- ステレオタイプ的な単語は避ける。例：「左翼」というコトバでイメージするものは、被調査者によって大きく異なる。

2.1.3 文章上の注意

- 過度な形容詞の使用は避ける。

- 被調査者が質問について肯定する傾向にある（“yes” tendency）ことに配慮して文章を工夫する。

- とくに複文で誘導的な前置き（威光暗示効果など）にならないよう注意する。例：「世間では○×といわれていますが、あなたは……」という問いは、世間の威光を借りて回答を歪める

- 単位によって限定されるような聞き方は避ける。例：「１ヶ月に何冊くらい本を読みますか？」という質問では、年に２～３冊の人は０か１に無理に分類される可能性が大きい

[2] この点の確認には、ダミーテーブル（データが得られたとして、どのように集計して作表するのかをイメージするために作ってみる仮の表）が役に立つ。可能ならやるべき。

- 文章の意味内容が２つ以上の論点を含んでいる質問（Double-barreled question、日本語でもそのまま「ダブルバーレル[3]」と呼ばれる）は、各論点について１つずつの質問群に分解する。
- 過去の細かい記憶をもとにした質問はしない。
- 否定的語法の質問は曖昧なので避ける。例：「市営動物園の缶詰会社への払い下げは、阻止すべきでしょうか、それともすべきではないと思いますか？」という問いで「すべきではないと思う」という回答は、「払い下げすべきではない」のか「阻止すべきではない」のか曖昧。
- あまりにも突飛な質問はしない。例：唐突に「もし火星に住むとしたら…」というような問いは、調査全体への信頼性を失わせかねない。

2.1.4 質問の仕方のタイプ

1. 個人的質問なのか一般的質問なのか。
2. 意識を聞くのか実態をきくのか。
3. 意見を聞くのか知識を問うのか。知識を問う質問を濾過質問として使い、知識のある人にだけ意見を聞く場合もある。ただし、知識を問うのが主目的ならテストをすべきである。
4. 平常の行動をきくのか、特定日時の行動をきくのか。例えば、食事調査をするとき、24 時間思い出し法と食物摂取頻度調査（FFQ）では、ふつう、結果は異なる。
5. 単一の質問で聞くか、質問群で捉えるか。単一の質問では把握できない構成概念を尋ねるためには、通常、妥当性検討済みの質問群を用い、その合計得点を構成概念のスコアとする。
6. 特定質問に yes または no と答えた者に対して、その判定を覆させるような第 2 の誘導的な質問を発して、**第 1 の質問の yes または no の強さを測る**。この第 2 の質問を biased question という。例：「今度の総選挙には投票に行きますか？　それとも行きませんか？」で「行く」と答えた人に対して、「投票日に雨が降っていたら／投票日に何か用事ができたら／どうしますか？」、「行かない」と答えた人に対して、「知人に誘われたらどうしますか？」と尋ねてみる。言い回しの効果が大きいので難しいが、うまく行けば最初からリッカート尺度などでスコア化するよりもシャープな評定ができる。

2.1.5 回答形式のタイプ

自由回答 自由回答質問とは、質問に対して自由に回答して貰うものである。聞くのは簡単だが分析が難しい。

プリコーディッド自由回答 プリコーディッド自由回答質問とは、質問形式はまったく自由回答と同じで、被調査者は自由に答えるが、調査者側で予め予想される回答としていくつかのカテゴリを用意しておいて（このことを「プリコーディッド」という）、聞き取ったときの判断でそのカテゴリの回答ボックスのどれかにチェックするものをいう。

[3]barrel は、普通は樽という意味だが、この場合は違う。石川淳志、佐藤健二、山田一成 (1998)「見えないものを見る力【社会調査という認識】」八千代出版の、p.284 によると、「ちなみにダブル・バーレルとは双胴銃のことで、一度に二つの弾丸が飛び出すしかけになっている」とのこと。

回答選択式 回答選択式質問とは、予め用意した選択肢から回答を選んで貰うものである。賛否的＝２項選択、品等的＝ rating、質的多肢選択、量的多肢選択などがある。

序列質問 序列質問とは、選択肢を並べておいて、それらに順位をつけさせるものをいう。

複数選択式 複数選択式質問も良く使われるが、データ化の際に注意が必要である。通常は１つ１つの選択肢を別々の変数として、選択されたら 1、されなければ 0 を与える。

2.1.6 スコア化のいろいろ

項目選択と過重の与え方が研究者の恣意によっているようなスコア化によってできた尺度は、任意尺度と呼ばれる。次のような例がある。

- チェイピンの社会経済的地位 (socio-economic status) 尺度：社会経済的地位を「文化的所有、有効所得、物質的所有、および地域社会活動への参加の、普通の平均的標準からみて、個人または家庭が占める位置」として、家庭の調度品の各々に任意の点数を与え、有効所得をアメイン（adult male maintenance の意味）ではかり、社会的参与度については団体別に名目的会員なら１点、集会に参加していれば２点、賛助会員なら３点、委員なら４点、役員なら５点といった任意の点数を与え、これら全部の合計を社会経済的地位尺度のスコアとしたもの。

- 態度測定の尺度としては、以下のものがよく知られている。

  ポイント尺度 測定しようとしているトピックについて賛成の態度を表している単語、文章、絵などと反対の態度を表しているそれを多数用意し、前者にそれぞれ＋１点、後者にそれぞれ－１点を与えておく。回答者に示して、賛成・同感できるものを選ばせ、その合計点をポイントとする。

  序列尺度 一連の単語・文章・絵などを示して、好きな順に並べさせて、それが予め任意に与えておいた順序とどの程度一致するかで態度を採点する。

  評定尺度 リッカート尺度と似ている。単一のある意見に対して、大いに賛成・賛成・中立・不賛成・大いに不賛成といった、通常は３段階から７段階までの回答カテゴリーのどれかを選ばせる（５段階の場合は５件法と呼ぶ）。数直線を示して丸をつけさせるのがいいとされる。その後、各回答カテゴリーに任意の点数を与える（例えば、５件法ならば１から５までのスコアを振ることが多い）。

  文章尺度 賛成から反対へと順に並んだいくつかの文章からなり、各文章には研究者の主観からなるスコアが与えられていて、回答者がチェックした文章についているスコアがその回答者の得点となる。

各項目への過重の与え方が研究者の恣意によらず、もっとも適当と思われる判定者の一団（学識経験者、そのトピックに精通している集団、被調査者と同じ集団に属する人々など）の総合的判定に委ねる尺度を判定尺度という。判定には第二種尺度の評定法、序列法、一対比較法などが使われる。例えば評定法の場合、測定対象を判定者集団に示して３段階から７段階までのどれかで評定させ、判定者集団全体についての平均値をその測定対象のスコアとする。

１つの次元にのっている複数の質問項目によって１つの尺度が構成されると考え、それらの合成得点として作られる尺度が内的一貫性尺度と呼ばれる。科学的な尺度

としては任意尺度や判定尺度よりもすぐれている。また、いくつかの項目の合計得点として得られる尺度に内的一貫性があることが信頼できるためには、慣例的にクロンバックの α 係数が少なくとも 0.7 以上であることが必要とされる。

クロンバックの α 係数を説明するために例を挙げよう。

質問紙によって何らかの概念の尺度を知ろうとするとき、多くの概念は直接聞き取ることができないので、複数の質問を組み合わせることによって対象者の差異をより細かく把握しようと試みることになる。例えば、自然への親近感を聞き取りたい場合に、

> （1）あなたは自然が好きですか？　嫌いですか？
> （好き、どちらかといえば好き、どちらかといえば嫌い、嫌い）

だけでは対象者は４群にしか分かれない（順序尺度として数値化すると、好きを４点、嫌いを１点として１点から４点の４段階）。しかし、

> （2）休日に海や山で過ごすのと映画館や遊園地で遊ぶのとどちらが好きですか?
> （海や山、どちらかといえば海や山、どちらかといえば映画館や遊園地、映画館や遊園地）

を加えて、これも「海や山」を４点、「映画館や遊園地」を１点とする順序尺度として扱うことにすれば、（1）と（2）の回答の合計点を計算すると、２点から８点までの７群に回答者が類別される可能性があり、より細かい把握が可能になる。さらに、

> （3）無人のジャングルで野生生物の観察をする仕事に魅力を感じますか？
> それとも感じませんか？
> （感じる、どちらかといえば感じる、どちらかといえば感じない、感じない）

の４点を加えると、3点から 12 点までの 10 段階になる。この合計得点を「自然への親近感」を表す尺度として考えてみると、３つの項目は同じ概念を構成する項目（下位概念）として聞き取られているので、互いに回答が同じ傾向になることが期待される。つまり（1）で好きと答えた人なら、（2）では海や山と答える人が多いだろうし、（3）では感じないと答えるよりも感じると答える人が多いだろうと思われる。同じ概念を構成する質問に対して同じ傾向の回答が得られれば、その合計得点によって示される尺度は、信頼性が高いと考えられる。

複数の変数（項目）の関連をみる指標の１つに、相関係数がある。変数 x と変数 y の相関係数 r_{xy} は、i 番目の人の x に対する回答を x_i、y に対する回答を y_i、x についての回答の平均値を $\bar{x}$、y についての回答の平均値を $\bar{y}$、総回答者数を n と書くことにすれば、

$$r_{xy} = \frac{s_{xy}^2}{(s_x s_y)} \quad \text{但し } s_x = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n-1}} \quad s_y = \sqrt{\frac{\sum_i (y_i - \bar{y})^2}{n-1}} \quad s_{xy}^2 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

として定義される。相関係数は –1 から 1 までの値をとり、まったく無関係なとき 0 となり、(x_i, y_i) を xy 平面にプロットしたときに傾きがプラスの直線上に完全に乗るとき 1 となる。

上記３つの質問に対して一貫した答えが得られたかどうかを調べる方法の１つに折半法がある。例えば質問（1）と（2）の合計点の変数 x_{12} と質問（3）の点の変数 x_3 という具合に、同じ概念を構成する全質問を２つにわけて、x_{12} と x_3 の相関係数を $r_{x_{12}x_3}$ とすれば、これらの質問の信頼性係数 $\alpha_{x_{12}x_3}$ は、$\alpha_{x_{12}x_3} = \frac{2r_{x_{12}x_3}}{1+r_{x_{12}x_3}}$ となると

いうのがスピアマン・ブラウンの公式である（ふつうは、奇数番目の項目と偶数番目の項目に二分）。

しかし、（1）の点と（2）と（3）の合計点という分け方もあるわけで、下位概念が3つ以上ある質問だったら、これらの回答に一貫して同じ傾向があるかどうかをスピアマン・ブラウンの公式で出そうと思うと、α の値はいくつも（n 項目だったら n 項目を2つに分ける組み合わせの数だけ）計算される。この場合だったら、$\alpha_{x_1x_{23}}$, $\alpha_{x_{13}x_2}$ も計算しなくてはいけないことになる。

それをまとめてしまおうというのが**クロンバックの α 係数**で、仮に（1）（2）（3）の合計得点が「自然への親近感」を表す変数 x_t だとして、（1）（2）（3）の得点をそれぞれ変数 x_1、x_2、x_3 とすれば、クロンバックの α は、

$$\alpha = \frac{3}{3-1}(1 - \frac{s_{x_1}^2 + s_{x_2}^2 + s_{x_3}^2}{s_{x_t}^2})$$

として計算される[4]。クロンバックの α 係数が 0.8 以上なら十分な、0.7 でもまあまあの、内的一貫性（信頼性）がその項目群にはあるとみなされる。なお、クロンバックの α 係数は、考えられるすべての組み合わせについてスピアマン・ブラウンの公式で計算される α を求め、その平均値をとった場合と同じ値を示す。

仮に上の質問を 10 人の人に対して実行した結果が `https://minato.sip21c.org/advanced-statistics/cronbach.txt` のように得られたとする。このタブ区切りテキストファイルをデータフレーム `x` に読み込んだとき、`library(fmsb)` として `fmsb` パッケージを読み込んでから（もちろん、fmsb パッケージをインストールしていなければ、予め `install.packages("fmsb")` としてインストールしておく必要がある）`CronbachAlpha(x)` とすれば、クロンバックの α 係数が 0.8027 とわかる。`library(psych)` を実行して `psych` パッケージを読み込み（もちろん、予め `install.packages("psych", dep=TRUE)` でインストールしておく必要がある）`alpha(x)` とした場合は、点推定量だけでなく、95%信頼区間などさまざまな値が得られる。また、構造方程式モデルのところで説明するが、`semTools` パッケージに含まれている `reliability()` 関数を使えば、クロンバックの α だけではなく、ω_1、ω_2、ω_3、AER という5種類の信頼性係数を得ることができる。

内的一貫性尺度の代表的なものとしては、次の3つがある。

- 項目分析によって作られる尺度：測定しようとしている量的特性との相関が高い項目のみを尺度内に採用する項目選択法。典型的には good-poor analysis を行う。手順としては、pre-test において、尺度に組み入れる候補となっている全項目に仮のスコアを与えて合計し、合計得点が第3四分位より大きい上位群と第1四分位より小さい下位群との間で、回答カテゴリーに差があった[5]項目を、その尺度を構成する質問項目として採用するものである。採用した項目のそれぞれについて、各回答カテゴリに与える点数は、(1) 任意に決める、(2) 判定者集団に依頼する、(3) リッカートのシグマ法（正規分布を仮定し、z[i]=(y[i-1]-y[i])/(p[i]-p[i-1]) としてスコアを与える。単純な合計得点をスコアとする場合なら、各カテゴリに1から順に整数を振るのとほとんど変わらないことがリッカートによって示されている）、(4) シーウェルらのシグマ法（式が違うが、リッカートと同じく正規分布を仮定するのでスコアもほとんど同じ）、(5) ギルフォードの方法などがある。
- 尺度分析によって作られる尺度：ガットマンが考案した方法。スケイログラムを用いる。

[4] 言葉で説明すると、項目数を項目数 − 1 で割った値に、各項目の得点の分散の和を合計得点の分散で割った値を1から引いた値を掛けたものである。

[5] 通常、回答カテゴリをまとめて2反応形式に直し、上位群において＋に反応した人の割合が下位群において＋に反応した人の割合よりも統計的に有意に高ければ差があったとみなす。回答カテゴリが多くて2反応形式に直しにくいときは平均値の差の検定とか順位和検定で有意差をみることもある。

- 因子分析によって作られる尺度：基本的には同じ因子に分類された（同じ因子の因子負荷量が大きい）項目の合計（あるいは重み付き合計）により尺度を構成する。通常、この尺度の信頼性を α 係数で検討する。

2.1.7 質問票の流れとレイアウト

- 質問の順序の原則
  1. 答えやすい質問は前
  2. 関連する事柄や似ているものは集める（システマティックにつくる）。ただし、それゆえにキャリー・オーバー効果（回答が、それまでの質問項目の影響を受けてしまうこと）が問題となる場合もある。
  3. 対象者を限定する枝分かれ質問（サブクエスチョン）で間違いにくい順番を工夫する
- タイトル：反発を起こすものは避ける
- 調査主体や連絡先の明記。
- 挨拶
- 記入上の注意
- 調査票についての処理の記録欄：コーディングで使う
- 小見出しや説明：対象者に調査の順序をわかってもらうための説明
- 質問番号：論理的階層性が明確な方がよい
- 回答上の指示：【】に入れるとか書体を変えるなど、質問との区別がはっきりするように。
- お礼の挨拶
- 調査員判定
- 最終レイアウトとページ数：最後のページがだいたい一杯におさまるようなレイアウトにし、通しのページ番号を振る。

2.1.8 質問紙調査データの解析のパースペクティブ

カテゴリ変数同士の関係をみることが多いので、多様なクロス集計をする必要がある（R では `table()` や `xtabs()` で可能）。クロス集計においては、変数間の独立性の検定をフィッシャーの直接確率（R では `fisher.test()` を使う）で行うことが多い。関連の強さは、オッズ比（`fmsb` パッケージの `oddsratio()` 関数や `vcd` パッケージの `oddsratio()` 関数などを使う）や四分相関係数（R では `vcd` パッケージの `assocstats()` 関数や、`polycor` パッケージの `polychor()` 関数で計算できる）などで評価することが多い。3つ以上のカテゴリ変数間の関係を見るときは、コクラン＝マンテル＝ヘンツェルの要約カイ二乗検定や（`mantelhaen.test()` を使う）、ロジスティック回帰分析（`glm()` で計算できる。適合度の指標としては `fmsb` パッケージに入っている `NagelkerkeR2()` により Nagelkerke の R^2 や `AIC()` を計算する）を実行する。

リッカート尺度による聞き取り結果をスコア化する場合は、クロンバックの α 係数（`fmsb` パッケージの `CronbachAlpha()` 関数や、`psych` パッケージの `alpha()` 関

数で計算できる）が小さければ因子分析（詳細は後述）する場合もある。スコアは量的変数として扱うので、カテゴリ間にスコアの差があるかどうかを調べるには、t 検定（`t.test()` を用いる）や一元配置分散分析（`aov()` や `oneway.test()` を用いる）を行うこともある。

最終的には構造方程式モデル（詳細は後述）を当てはめる場合も多い。

なお、質問紙への回答の信頼性を確かめるために同じ質問紙調査を対象者 1 人につき 2 回ずつ実施する（あるいは、同じ対象者への異なる評価者による評点があるとき、各項目について 2 人分の評点が付される）ことがある。この場合も、回答がカテゴリであれば、2 つの別々の質問項目の場合と同じ形でクロス集計表を作ることができるが、独立でないのは当然なので、フィッシャーの直接確率などを計算しても意味は無い。むしろ、偶然では考えられないほど一致しているかという、一致度を計算すべきである。典型的な一致度の指標は Cohen の κ 係数である。完全一致の場合 1、偶然と同じ一致度で 0、完全不一致で –1 となる。`fmsb` パッケージの `Kappa.test()` 関数に行数と列数が等しいクロス集計表オブジェクトを与えると自動的に計算され、一致度がどの程度かという目安も表示される。

これと同じように、同じ質問が繰り返される場合であっても、2 回の調査の間に何らかの介入があって、介入効果によって回答が偶然では考えられないほど変化したかを知りたい場合も、フィッシャーの直接確率のような独立性の検定は使えない。代わりに用いるのはマクネマーの検定であり、`mcnemar.test()` 関数に行数と列数が等しいクロス集計表オブジェクトを与えれば実行できる。

2.2 実験によって得られる測定値について

治験を含む実験の場合、カテゴリデータは曝露の有無など所与の条件であることが多い。毒性試験では毒性発現の有無、疾病発生の有無、死亡か生存かといった 2 値データをアウトカムとして用いる場合もある。それらを除けば、実験で得られるデータは、概ね数値型の測定値である。測定限界と有効数字に注意する必要がある。

実験では、統計解析方法は実施前に決めておくのが原則である。新薬の有効性であれば分散分析、毒性試験ならばアウトカム発生までの時間に対する生存時間解析、あるいは用量反応関係についてのプロビット解析またはロジット解析による LD50 や ED50 の推定、アウトカムとして量的な効果をみるなら重回帰分析、経時的な変化を調べるなら反復測定分散分析など、ある程度やるべきことは決まっている。これらのうち、分散分析、生存時間解析、重回帰分析、反復測定分散分析については、EZR を使ってメニュー操作で分析でき、保健学共通特講 IV, VIII のテキストで、ある程度説明してあるので、そちらを参照されたいが、非線形回帰は EZR でサポートされていないので、LD50 や ED50 の推定法については、本テキストの応用回帰分析の中で説明する。

なお、実験データについて統計解析をされる方に対して素晴らしいパースペクティブを与えてくれる本として、三中信宏 (2015)『みなか先生といっしょに 統計学の王国を歩いてみよう〜情報の海と推論の山を越える翼をアナタに！』羊土社を読むことをお勧めする。

2.3 健診など調査によって得られる、測定値とカテゴリの複合データについて

フィールド調査をすると、質問紙と測定の両方を実施することが珍しくない。縦断研究の場合には連結可能匿名化が必要だが、同一人をどうやって追跡するかが大きな問題となる。あらゆるタイプのデータが含まれる可能性があり、データ解析もある程度探索的にならざるを得ないので、統計解析としては最も難しい。しかも、欠

損値が珍しくないので、まずは欠損の質の検討が必要である。ランダムな欠損なら問題はないが、調べたい内容と欠損になるかどうかが関連していると非常にまずい。ランダムな欠損の場合、多重代入法 (multiple imputation) によって欠損値を補うことが良く行われる。`mice` パッケージや `Amelia` パッケージを使うことが多い[6]。

最も大切なのは、**他の解析をする前に、データの分布をよく見ておく**ことである。カテゴリ変数なら度数分布図（`barplot()` で描くことができる）、量的変数ならヒストグラム（`hist()` で描くことができる）や正規確率プロット（`qqnorm()` で描くことができる）を作るのが常道である。健診データでは血圧正常値とかメタボリックシンドロームの腹囲カットオフ値のように、連続量として測定した値を正常・異常の 2 値情報にしてしまうことが良く行われるが、境界付近の値を単純に 2 値化することは問題がある。分布が明らかに二峰性なら谷のところで区分することに問題はないが、正規分布に近い形をしていて、質的な違いがあるわけでもないのに、固定されたカットオフ値を使って 2 値情報にしてしまうことは薦められない。統計解析のセンスからすれば、そのような場合は連続量のまま扱う方が筋が良い。どうしてもカテゴリ化したければ、明らかな低値、中間値、明らかな高値というカテゴリにして、明らかな低値と明らかな高値の 2 群間で比較することを検討すべきである。

2 変数の関連を分析する場合、どちらもカテゴリならモザイクプロット（`mosaicplot()`）、片方がカテゴリでもう片方が量ならカテゴリ変数で層別したストリップチャート（`stripchart()` で描ける）や箱ひげ図（`boxplot()` で描ける）、どちらも量なら散布図（`plot()` で描ける）を作る。

3 変数以上の場合は、3 つめ以降の変数は色やプロット記号を変えるなどして 2 次元グラフの重ね描きとして表現するか、3 つめ以降の変数で層別して複数の 2 次元グラフを作成するなど、さまざまな手法がある（詳しくは後述）。

[6] 多重代入法については、高橋将宜・渡辺美智子 (2017)『欠測データ処理：R による単一代入法と多重代入法』共立出版、ISBN: 978-4-320-11256-8 をお薦めする。なぜ欠損値を含むケースを単純にすべて除去したり、単一代入で済ませることがバイアスにつながってしまうのか、実際に多重代入をする際にどこをチェックしなくてはいけないのか、など明確に解説した素晴らしいテキストである。

Chapter 3

Rによるデータの前処理

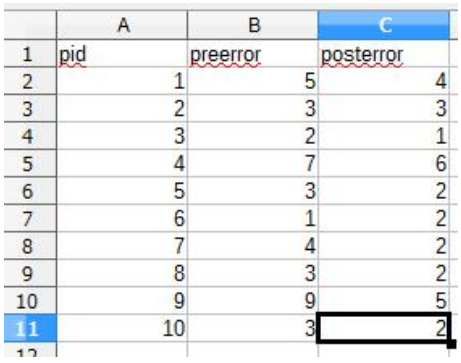
R で使うデータは、通常、表形式で入力したデータフレームになる。原則として、1 個体が 1 行になるように作成する。異なる時点での測定値や、複数選択の選択肢は、別々の変数（列）にする。1 行目は変数名にする。変数名はアルファベットで始まるようにし、英数字とピリオドだけからなるようにすべきである。グラフの軸ラベルを漢字で表記したい場合は、グラフ描画関数の中で指定すべきであり、変数名は英数ピリオドだけにする方がエラーが起きにくい。

前処理が必要な場合が多々あるのでまとめておく。

3.1 データを積む

10 人の被験者がいて、コーヒーを飲む前後で百マス計算をしてもらい、誤答数を記録した結果が、以下のように得られているとする。

被験者	1	2	3	4	5	6	7	8	9	10
飲用前	5	3	2	7	3	1	4	3	9	3
飲用後	4	3	1	6	2	2	2	2	5	2



	A	B	C
1	pid	preerror	posterror
2	1	5	4
3	2	3	3
4	3	2	1
5	4	7	6
6	5	3	2
7	6	1	2
8	7	4	2
9	8	3	2
10	9	9	5
11	10	3	2

Figure 3.1: コーヒー飲用前後での百マス計算の誤答数の変化

もちろん Excel や LibreOffice Calc などで図 3.1 のように入力してから、範囲選択してコピーし、

```
coffee <- read.delim("clipboard")
```

のように[1]、データフレーム coffee に付値してもよいし、タブ区切りまたはコンマ区切りのテキストファイル、例えば e:/work/coffeedata.txt として保存し、それを

```
coffee <- read.delim("e:/work/coffeedata.txt")
```

のようにして読み込んでもよい。

しかし、この程度のデータの量ならば、次の R コードとして直接ベクトルを定義し（data.frame() の中では<-でなく=を使うことに注意。つまり、ここでやっているのはオブジェクトへの付値ではなく、ラベル付けである）、データフレームとして付値する方が簡単である。

https://minato.sip21c.org/advanced-statistics/coffee.R(1)

```
coffee <- data.frame(
 pid = 1:10,
 pre = c(5, 3, 2, 7, 3, 1, 4, 3, 9, 3),
 post = c(4, 3, 1, 6, 2, 2, 2, 2, 5, 2))
```

コーヒー飲用前後で誤答数が統計学的に有意に変化したかどうか知りたい場合は、この形のまま、以下の枠内のコードを打てば同じ人の誤答数を線でつないだグラフが描かれ、検定もできる。コーヒー飲用後、百マス計算の誤答数が有意水準 5%で統計学的に有意に減ったことがわかる。

https://minato.sip21c.org/advanced-statistics/coffee.R(2)

```
plot(c(1, 2), c(0, 10), type="n", frame=FALSE, axes=FALSE,
 xlab="コーヒー飲用", ylab="誤答数")
segments(1, coffee$pre, 2, coffee$post)
axis(1, 1:2, c("前", "後"))
axis(2, 0:10, 0:10)
t.test(coffee$post, coffee$pre, paired=TRUE)
# 前後の差の母平均が 0 という検定と同じなので次の行でも同じ結果
# t.test(coffee$post-coffee$pre, mu=0)
```

ここで、仮に個人差を無視し、コーヒーを飲んでいない群と飲んだ群とで誤答数を比較するという操作をしたいときは、データの形を変える必要がある。簡単に言えば、次の枠内のようにしてデータを積み上げるとよい。

https://minato.sip21c.org/advanced-statistics/coffee.R(3)

```
scoffee <- data.frame(
 pid = rep(coffee$pid, 2),
 errors = c(coffee$pre, coffee$post),
 setting = factor(c(rep(1, 10), rep(2, 10)), labels=c("pre", "post")))
```

積み上げ型データは、もっと簡単に、

[1] MacOS では read.delim(pipe("pbpaste")) としなくてはいけない。Windows には"clipboard"という名前のデバイスがあるが、MacOS では"pbpaste"というアプリにパイプ処理でリダイレクトする。Windows にも"clip.exe"というアプリがあるので、同様にパイプ処理にすることもできるはずである。誰もそんなことはしないが。

https://minato.sip21c.org/advanced-statistics/coffee.R(4)

```
scoffee2 <- stack(list(pre=coffee$pre, post=coffee$post))
```

でも作成できる。ただし pid は引き継がれないし、数値変数名は values、グループ変数名は ind と固定されている。同様に、car パッケージの reshape() 関数を使えば、縦長形式と横長形式を相互変換できる。ただし、変数名としてピリオドの後に時点を示す数値を含んでいる必要がある。この場合、数値変数名は t、グループ変数名（時点名）は time と固定されている。

https://minato.sip21c.org/advanced-statistics/coffee.R(5)

```
library(car)
colnames(coffee) <- c("pid", "t.0", "t.1") # pre → t.0、post → t.1 が必須
scoffee3 <- reshape(coffee, direction="long",
 idvar="pid", varying=c("t.0","t.1"))
# coffee3 <- reshape(scoffee3, direction="wide") # で戻せる
```

このようにして作った積み上げ型データ scoffee を使って 2 群間の平均値の比較をするには以下のようにする。ストリップチャートが描かれ、Welch の方法による等分散性を仮定しない t 検定が実行される。

https://minato.sip21c.org/advanced-statistics/coffee.R(6)

```
stripchart(errors ~ setting, data=scoffee, method="jitter",
 vert=TRUE, ylim=c(0, 10))
meanerrors <- tapply(scoffee$errors, scoffee$setting, mean)
sderrors <- tapply(scoffee$errors, scoffee$setting, sd)
igroups <- c(1.1, 2.1)
points(igroups, meanerrors, pch=18, cex=2)
arrows(igroups, meanerrors-sderrors, igroups, meanerrors+sderrors,
 angle=90, code=3)
t.test(errors ~ setting, data=scoffee)
```

ノンパラメトリックな図示と検定の場合はもっと簡単で、

https://minato.sip21c.org/advanced-statistics/coffee.R(7)

```
plot(errors ~ setting, data=scoffee)
wilcox.test(errors ~ setting, data=scoffee)
```

とするだけで層別箱ヒゲ図が描かれ（setting という変数がファクター型なので自動的に boxplot() が呼ばれる）、ウィルコクソンの順位和検定（マン＝ホイットニーの U 検定と数学的に同一）が実行される。

t 検定でもウィルコクソンの順位和検定でも、このように個人差を無視してしまうと、このデータでは 2 群間には統計学的な有意差を見いだすことができなくなることがわかる。従って、あくまでデータの性質に従ってデータファイルを設計すべきであり、このように積み上げ操作をすることは必ずしも一般的でないが、データフレームの前処理として覚えておくと役に立つことがある。

3.2 表の操作

カテゴリデータを表形式で操作するテクニックを、簡単な例で示す。https://minato.sip21c.org/medstat/sample11.txt は 40 人分の、年齢 AGE、曝露の有無 EXPOSURE（YES と NO の 2 値）、疾病の有無 DISEASE（YES と NO の 2 値）からなるタブ区切りテキストデータである。これを dat というデータフレームに読み込むには、

```
dat <- read.delim("https://minato.sip21c.org/medstat/sample11.txt")
```

とする。このデータについてのさまざまな集計方法をまとめてみる。

EXPOSURE の集計 table(dat$EXPOSURE) と打てば、以下が表示される。

```
NO YES
20  20
```

結果を度数分布表ベクトルとしてオブジェクト EXC に付値 EXC <- table(dat$EXPOSURE)

DISEASE の集計 table(dat$DISEASE)

```
NO YES
16  24
```

曝露ありの人の DISEASE の集計 table(dat$DISEASE[dat$EXPOSURE=="YES"])

```
NO YES
 4  16
```

曝露あり結果を EXD に付値 EXD <- table(dat$DISEASE[dat$EXPOSURE=="YES"])

曝露なし結果を NED に付値 NED <- table(dat$DISEASE[dat$EXPOSURE=="NO"])

２つのオブジェクトを行方向に結合 rbind(NED, EXD) で曝露の有無と疾病の有無のクロス集計結果が得られる。

```
    NO YES
NED 12   8
EXD  4  16
```

クロス集計 実はいきなり table(dat$EXPOSURE,dat$DISEASE) でクロス集計できる。

```
     NO YES
NO   12   8
YES  4  16
```

表題付きクロス集計 xtabs(~EXPOSURE+DISEASE, data=dat)

```
          DISEASE
EXPOSURE NO YES
     NO  12   8
     YES  4  16
```

行列定義 各組合せ人数が最初からわかっていれば、X <- matrix(c(12, 4, 8, 16), 2, 2)

ラベルをつける `rownames(X) <- c("NO", "YES"); colnames(X) <- c("NO", "YES")`

ラベル (2) `dimnames(X) <- list(c("非曝露", "曝露"), c("健康", "病気"))`

テーブルにする `attr(X, "class") <- "table"`

独立性のカイ二乗検定 `chisq.test(X)`

Fisher の正確確率検定 `fisher.test(X)`

年齢 60 歳以上/未満の 2 群に区分した変数 AC を dat 内に作る 次のどちらかを実行する。以下の説明では `ifelse()` を使ったとする。

```
dat$AC <- cut(dat$AGE, c(min(dat$AGE), 60, max(dat$AGE)+1),
 right=FALSE)
dat$AC <- factor(ifelse(dat$AGE<60, 1, 2),
 labels=c("<60", "60<="))
```

AC で元データを 2 群に分け、2 群別々にクロス集計して YTAB と ETAB に付値 以下のようにする。

```
YTAB <- xtabs(~EXPOSURE+DISEASE, data=subset(dat,AC=="<60"))
ETAB <- xtabs(~EXPOSURE+DISEASE, data=subset(dat,AC=="60<="))
```

60 歳未満/以上で別々に Fisher の正確確率検定 `fisher.test(YTAB); fisher.test(ETAB)`

3 次元のクロス表を作る `D3TAB <- array(c(YTAB, ETAB), dim=c(2,2,2))` とすると、3 次元のクロス表が D3TAB にできる（ラベルが全部消えてしまうが)。D3TAB と打つと、次のように見える。

```
, , 1

     [,1] [,2]
[1,]    4    3
[2,]    4   13

, , 2

     [,1] [,2]
[1,]    8    5
[2,]    0    3
```

xtabs や table で作る 実は以下のどちらかで直接 3 次元クロス表ができる。

```
D3TAB <- xtabs(~EXPOSURE+DISEASE+AC, data=dat)
D3TAB <- table(dat$EXPOSURE, dat$DISEASE, dat$AC)
```

3 次元の表から年齢層別に二次元クロス集計表を取りだす 3 次元クロス表から 2 次元クロス表を取り出すには、

```
YTAB <- D3TAB[,,1]
ETAB <- D3TAB[,,2]
```

60 歳未満/以上どちらでも曝露と疾病に関連はないという帰無仮説の検定 マンテルヘンツェルの検定[2]を行うのが普通である。帰無仮説が有意水準 5%で棄却されるので、どの年齢層でもこの曝露と疾病の間には統計学的に有意な関連があるといえる。また共通オッズ比は 7.3 [1.29, 41.6] であり、年齢で層別した場合に、どの年齢層でも共通して非曝露群に比べて曝露群での疾病オッズが 7.3 倍と見ることができる。

3 次の交互作用がない帰無仮説の Woolf の検定 vcd ライブラリに入っていて以下で実行できる。

```
library(vcd)
woolf_test(D3TAB)
```

3.3 再コーディングと文字列操作

3.3.1 データの再コーディング

例えば、x というデータフレームの AREA という数値変数（値は 1〜9）に地域区分が入っている状態を考えよう。次のコードで生成できる。

```
set.seed(54321) # 擬似乱数列に初期値 54321 を与える
x <- data.frame(AREA=sample(1:9, 100, replace=TRUE))
```

AREA を地域名（A〜I）がついたファクター型に変換し、かつ 3 種類の街区（市街地=A,C,G、農村部=B,F,H、工業地区=D,E,I）に区分し直した新しい分類変数 REG を作って同じデータフレームに入れたいときは、次のようにする。

```
NAREA <- c('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I')
# 実は LETTERS[1:9] と同じ
x$AREA <- factor(x$AREA, labels=NAREA)
x$REG <- factor(ifelse(x$AREA %in% c('A', 'C', 'G'), 1,
 ifelse(x$AREA %in% c('B', 'F', 'H'), 2, 3)),
 labels=c('市', '農', '工'))
# 以下は別解
NREG <- c('市', '農', '市', '工', '工', '農', '市', '農', '工')
x$REG <- NREG[as.integer(x$AREA)]
```

3.3.2 文字列操作

R 本体が最も苦手とする処理の 1 つが文字列操作である。以下、いくつかの役に立つ操作をまとめておく。なお、本格的に文字列操作をしたい場合は、stringr や stringi

[2] mantelhaen.test(D3TAB)

といった文字列処理用のパッケージを用いると良いらしい[3]。

ファイルからの読み込みで文字列をファクターに自動変換させない `read.delim()` 関数などで、文字列をファイルから読み込むとき、通常は自動的にファクター型になる。この自動変換をさせないグローバルオプションが `options(stringsAsFactor=FALSE)` である……というのが R-3.6.3 までの仕様だったが、R-4.0.0 以降、この自動変換はしないのがデフォルトになった。自動変換したいときは、`read.delim()`、`read.csv()` などの関数の中に、オプションとして `stringsAsFactors=TRUE` を入れる必要がある。

データフレーム内のファクターを文字列に データフレーム `bob` の中のファクター型の変数を一括で文字列型に変えたい場合は以下のようにする。

```
i <- sapply(bob, is.factor)
bob[i] <- lapply(bob[i], as.character)
```

数値を書式付きで文字列に変換 C 言語と同様の仕様で `sprintf()` という関数が使える。表示桁長を見やすく揃えるときも便利。例えば、`sprintf("%09d", 4)` の結果は以下。図中などに浮動小数点表示をさせたくないときにも便利。

```
> sprintf("%09d", 4)
[1] "000000004"
> x <- 123456789012345
> x
[1] 1.234568e+14
> sprintf("%15.0f", x)
[1] "123456789012345"
> x <- 0.0000000456
> x
[1] 4.56e-08
> sprintf("%10.8f", x)
[1] "0.00000005"
```

文字列処理関数群 `paste()`、`substr()`、`strsplit()` などであるが、あまり機能は充実していない。`stringr` パッケージを使うと、例えば、ある文字列に含まれる別の文字列の個数を返す `str_count()` 関数などが使える[4]。最近では `stringi` パッケージを薦める人も多い[5]。

[3] https://rpubs.com/uri-sy/demo_stringr や https://qiita.com/kohske/items/85d49da04571e9055c44 を参照。

[4] 例えば、`str_count("abc1234def5432", "4")` は、第二引数の文字列が第一引数に 2 回出現するので 2 を返す。

[5] https://qiita.com/kohske/items/85d49da04571e9055c44 など。

Chapter 4

Rによる多様な作図技法

作図はデータ解析の常道である。どんなに複雑な統計解析をする場合にも、データの性状を知るために作図は必須である。R では多種多様なデバイス（ベクトルグラフィックス＝図形ファイルとしてウィンドウズメタファイルや pdf、ポストスクリプトなど、ラスターグラフィックス＝画像ファイルとして tiff や jpeg など、あるいはコンピュータのディスプレイ）に作図することが可能だし、図形ファイルは後で PowerPoint、LibreOffice Draw/Impress に読み込んで「切り離す」ことで線単位で再編集でき、画像ファイルは Photoshop などのフォトレタッチソフトで加工できる。なお、ラスターグラフィックスデバイスの中では、bg="transparent"として背景に透過色を指定できる `png()` も使いやすいデバイスである。データが何千点もある散布図など、ベクターグラフィックスよりもラスターグラフィックスにした方が操作が軽くなるしファイルサイズも小さくなる。

4.1 作図の基本プロセス

R の作図の基本プロセスは以下のステップを踏む。なお、R のグラフィックスには `base` の他に `grid` というシステムがあり、`grid` を使って探索的作図ができることでよく知られている `ggplot2` というパッケージもよく使われているが、このテキストでは `grid` は扱わない。`ggplot2` について知りたい方は、開発者である Hadley Wickham 自身が書いた本を徳島大学の石田基広さんが翻訳した、H. ウィッカム（著）、石田基広、石田和枝（訳）『グラフィックスのための R プログラミング：ggplot2 入門』シュプリンガー・ジャパン株式会社、ISBN 978-4-431-10250-2 を参照されたい。

1. `pdf("ファイル名", width=横幅, height=高さ)`、`win.metafile("ファイル名", width=横幅, height=高さ)`、`windows()` のようにしてグラフィックスデバイスを開く。`windows()` デバイスでもサイズ指定は可能である。省略するとインタラクティブに操作しているときはコンピュータのディスプレイ（OS が Microsoft Windows なら `windows()` デバイスを開くのと同じ）、バッチ処理では pdf デバイスとして `Rplot.pdf` というファイルが出力先になる（既に `Rplot.pdf` が存在する場合は、上書きではなく、`Rplot01.pdf` などと自動的に数字が加わったファイルができていくはずである）。

2. `layout()`、`par()` などで、そのデバイス上へのグラフの配置や余白を設定する。例えば `layout(1:2)` とするとデバイスが上下 2 分割される[1]。`par()` でよく使

[1] `layout(matrix(c(1, 1, 2, 3), 2, 2)` とすると、デバイス左半分が第 1 のグラフ、右上が第 2 のグラフ、右下が第 3 のグラフを描く領域として分割される。つまり、m 行 n 列の行列を `layout()` に渡

われるのは、cex=2 によって文字とシンボルのプロットサイズを標準の 2 倍にするとか、family="sans"でフォントをサンセリフ体にするとか[2]、las=1 で軸目盛ラベルが常に水平に書かれるようにするとか[3]、mar=c(4, 3, 3, 1)+0.1 として余白を 1 列ずつデフォルト値より狭くする（指定順序は下、左、上、右）といったオプションである。

3. plot() や hist() などの座標系設定を伴うメイングラフ描画関数でグラフを描く。xlim=c(横軸最小値, 横軸最大値) で座標系の横軸、ylim=c(縦軸最小値, 縦軸最大値) で座標系の縦軸を指定できる。log="x"オプションをつけると横軸のみ対数軸になり、log="xy"とすると両対数グラフになる。xlab="横軸のラベル"、ylab="縦軸のラベル"というオプションで軸ラベルを付けることができる。なお、plot() で外枠を描きたくない場合は frame=FALSE オプション、軸をカスタマイズしたい場合は axes=FALSE オプションを付ける。座標系は設定したいけれどもデータをプロットしたくない場合は、type="n"オプションを付ける。

4. axes=FALSE だった場合は、axis(1, 数値ベクトル, ラベル文字列ベクトル) で横軸、axis(2, 数値ベクトル, ラベル文字列ベクトル) で縦軸を設定する（3 で上、4 で右にも軸を付けられる）

5. lines() や arrows() や text() や legend() でグラフに追記する

6. dev.off() でデバイスが閉じられ、描画が完了する

4.2 日本語を扱う上での注意点

日本語を扱うときに必要な手続きは、デバイスによって異なる。ベクターグラフィックスとして出力するための使用デバイスとしては、MacOS X 環境では quartz() が基本[4]とのことだが、Windows 環境では quartz() は利用できない。最近では Windows 環境でも svg() や cairo.pdf() が利用できるようになった。以下、古くから使える postscript()、pdf()、win.metafile() についての注意点を書いておく。

postscript() や pdf() 中間さんの https://prs.ism.ac.jp/~nakama/AI/AI_UCS2.R を自分のスクリプトにコピーペーストするか source() を使って先に実行してからグラフィックデバイスを開き、par(family="Japan1GothicBBB") をしてグラフ出力すべき。はしご高のように、UTF-8 では表示できるが EUC-JP では表示できないような文字も表示できるようになる。実際には、

```
source("https://prs.ism.ac.jp/~nakama/AI/AI_UCS2.R")
```

とすれば良い。

win.metafile() 以下のコードを先に実行してからメイングラフ描画関数を実行すべき。Windows 上でのディスプレイへの表示でも同様である。

し、中身の数字を描画順序を示す整数にすれば良いので、かなり複雑な画面分割もできる。もちろん画面分割などせず、別々に作ったグラフを、後で LibreOffice Draw などで加工しても良いわけだが、コードで書いておけば手作業がないので再現や修正が容易である。

[2]日本語を描画に使うときもこの family=オプションは重要。

[3]これを指定しないと、縦軸の目盛ラベルは自動的に左 90 度回転される

[4]https://oku.edu.mie-u.ac.jp/~okumura/stat/graphs.html 参照。

```
windowsFonts(JP1=windowsFont("MS Gothic"),
             JP2=windowsFont("MS Mincho"),
             JP3=windowsFont("Meiryo"),
             JP4=windowsFont("Biz Gothic"))
par(family="JP1")
```

こうしておけば、拡張メタファイル (*.emf) を PowerPoint や LibreOffice の Impress などに読み込んでから編集するためにオブジェクト変換しても漢字が文字化けしない。また、text() の中でも family="JP1"のような指定は可能である。

なお、ウェブ上に存在する R のスクリプトやテキストファイル形式データの日本語文字コードは、CP932 であったり UTF8 であったりバラバラである。名前を付けて保存した場合、文字コードもそのまま保存される。しかし、RStudio では日本語文字コードは UTF8 と想定されており、R Console のスクリプトエディタでは Windows 環境の場合 CP932、MacOS X では UTF8 と想定されているため、文字化けする場合がある。Firefox、Chrome、Edge などのブラウザの文字コード判定は賢い場合が多く、文字コード指定も簡単なので、ブラウザで当該スクリプトやデータを開いて全部選択コピーし、文字化けしている RStudio やスクリプトエディタにペーストすれば、文字化けしていないコードを得ることができる。覚えておくと便利なテクニックであろう。
また、RStudio で Project を設定していれば、RStudio のメニューバーから、Tools ＞ Project Options ＞ Code Editing ＞ Text Encoding で、そのプロジェクトのマルチバイトコードをリアルタイムに指定できる。ファイルを開いて文字化けしていたら、いったん閉じて、このメニューでマルチバイトコードを変更してから開き直すことで、問題が解決するであろう。

4.3 メイングラフ描画関数のいろいろ

hist() ヒストグラムを描く

qqnorm() 正規確率プロットを描く

barplot() 棒グラフを描く。行列（＝ 2 次元クロス集計表）を与えると、積み上げ棒グラフやサブグループ別の棒グラフ（beside=TRUE オプションを付けた場合。デフォルトは FALSE なので積み上げ棒グラフになる）が描ける。horiz=TRUE にすると横棒グラフになる（デフォルトは horiz=FALSE なので縦棒グラフになる）

boxplot() 箱ひげ図を描く

stripchart() ストリップチャートを描く

dotchart() ドットチャートを描く

mosaicplot() モザイクプロットを描く

pie() 円グラフを描く

plot() plot() は総称的な関数なので、与えるオブジェクトによって動作が変わる。2 つのカテゴリ変数をコンマで区切って与えればモザイクプロットになるし、plot(量的変数 ~ カテゴリ変数, data=データフレーム) のようにするとカテゴリ変数で層別した層別箱ひげ図が描かれるし、2 つの量的変数をカンマで区切って与えるか、plot(量的変数 ~ 量的変数, data=データフレーム) とすれば散布図が描かれる。x というデータフレームに 2 つの量的変数 A と B があるとき、plot(xA, xB) でも plot(B ~ A, data=x) のどちらでも、変数 A が横軸、変数 B が縦軸の散布図が描かれる。type="b"とするとデータ点が線でつながれる。pch=オプションでプロット記号を指定でき、col=オプションで色を指定できる。

pairs() 複数の変数の同時散布図を描く

matplot() 複数の系列を 1 枚の散布図の中に重ね描きする

coplot() 第 3（＋第 4）の変数で層別した複数の散布図を描く。詳細は example(coplot) で確認できるが、２つの要因で層別した同時散布図を coplot(y~x | a*b) によって実行する場合、a や b が数値だと層別数は a についても b についてもデフォルトでは 6 である（numbers=で変更可）。a や b がファクター型なら、カテゴリごとに plot(y~x) される。

dataEllipse() car パッケージが必要。散布図と集中楕円（確率楕円）を重ね描きする

radarchart() fmsb パッケージが必要。レーダーチャート（蜘蛛の巣グラフ）を描く

4.4 具体的なグラフの作り方

以下、いくつかの事例について、具体的にグラフを作ってみる。

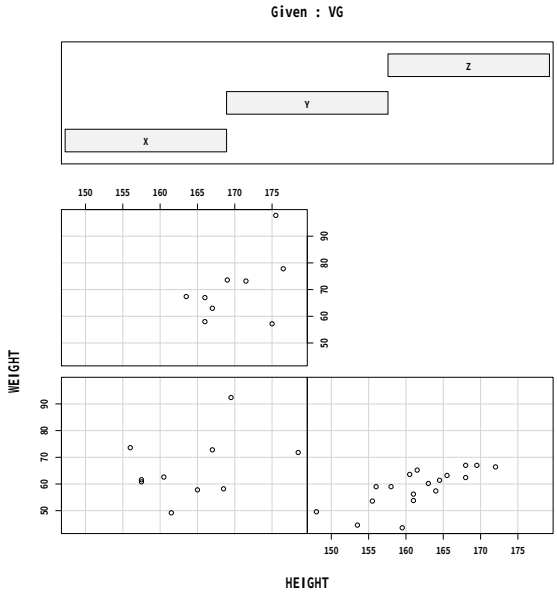
4.4.1 群ごとにプロット記号を変えた散布図を描く

散布図や層別ストリップチャートで第 3 の変数によってプロット記号を変えてみると、多くの情報が得られる。例えば、身長と体重の関係を散布図にするとき、男女別にプロット記号の形や色を変えると、男女込みにしたときに見られる相関関係は、男性が女性よりも身長も体重も平均して大きい傾向があることによって実際以上に強い正の相関関係があるように見えていることがわかる。

ここでは、X、Y、Z という 3 つの村があって、それぞれ身長と体重のデータがあって、その関係を村ごとにマークを変えてプロットしたいとする。データは、https://minato.sip21c.org/advanced-statistics/v3hw.txt からタブ区切りテキストとして入手できる。変数名は村が VG、身長が HEIGHT、体重が WEIGHT である。データを x というデータフレームに読み込み、まずざっくりと村ごとに分けた散布図を描くには coplot() を使う。

https://minato.sip1c.org/advanced-statistics/scdehot.R(1)

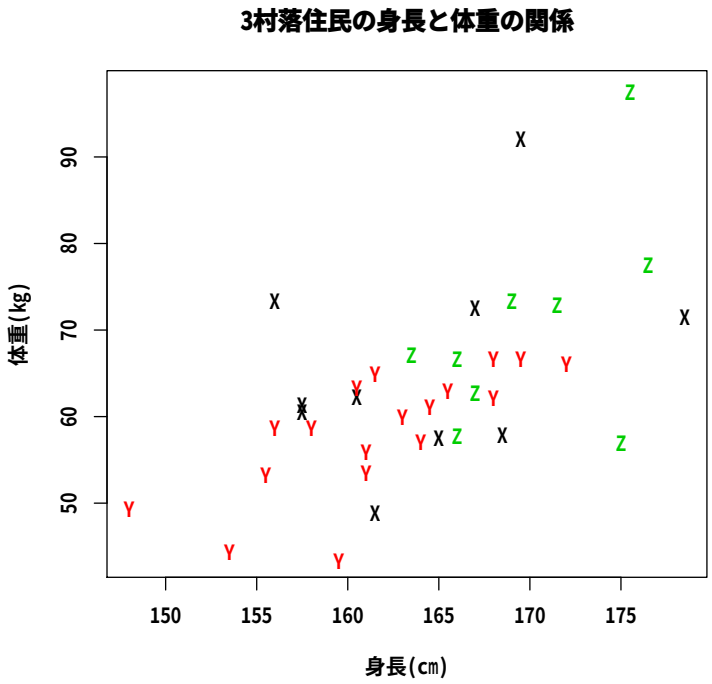
```
URL <- "https://minato.sip21c.org/advanced-statistics/v3hw.txt"
v3 <- read.delim(URL, stringsAsFactors=TRUE)
plot(WEIGHT ~ HEIGHT, data=v3)
coplot(WEIGHT ~ HEIGHT | VG, data=v3)
```



これだと村落間の違いが分かりにくいので、村の名前をそれぞれ違う色で身長と体重の座標位置にプロットしてみる。コードは次の通り。

https://minato.sip1c.org/advanced-statistics/scdehot.R(2)

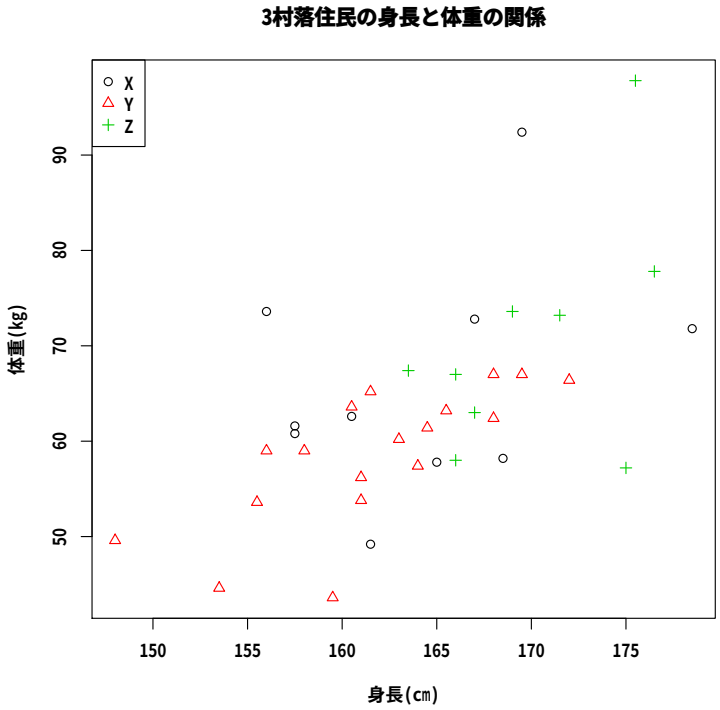
```
plot(WEIGHT ~ HEIGHT, data=v3, pch=as.character(VG),
 col=as.integer(VG), xlab="身長 (cm)", ylab="体重 (kg)",
 main="3 村落住民の身長と体重の関係")
```



村の名前をプロットするのは見栄えが悪いので、適当なシンボルを使ってプロットし、凡例を付記する方が良い。次のようにする。

https://minato.sip1c.org/advanced-statistics/scdehot.R(3)

```
plot(WEIGHT ~ HEIGHT, data=v3, pch=as.integer(VG),
 col=as.integer(VG), xlab="身長 (cm)", ylab="体重 (kg)",
 main="3 村落住民の身長と体重の関係")
series <- 1:length(levels(v3$VG))
legend("topleft", pch=series, col=series, legend=levels(v3$VG))
```



このように色とシンボルを組み合わせると多くの水準を描き分けることができる。pch に与える値として、1 から 25 まではプロットとして適切なシンボルが既に定義されている（26 から 32 は空白で、33 以上は文字や記号）ので、col="red"とか col="blue"などと色を指定するか、剰余を使うなどして周期的変数を生成して色を変えれば 120 くらいは何とかなる。他にも、以下 2 つの方法がある。

- text() 関数を使って文字列を重ね打ちする: plot(x, y) の後に（pch='.' や pch=20 でプロット記号を小さい点にすると良い)、

  ```
  text(x, y, paste(string), pos=4, offset=0.5)
  ```

  とすれば、string を (x, y) の点の右側に表示してくれる。

- identify() 関数を使う: すべてのデータ点を特定する必要はないので、必要な点についてだけ情報を表示できるのがベストであろう。plot(x, y) の後に identify(x, y, labels=string) としておくと、プロットの後に十字型のマウスカーソルが出現するので、画面上で string を表示したい点の上でクリックすれば string が出現する。描画ウィンドウのメニューの stop からか、右クリックメニューから stop を選ぶまで複数の点をクリックできる。

ここまでやったなら、村落間で身長と体重の関係に違いがあるかどうかを知りたくなるだろう。集中楕円を描き、Hotelling の T^2 検定を実行するには以下のコードを打つ。パッケージとして car と Hotelling が必要になるため、予めインストールしておく（たぶん既に入っていると思うので install.packages("car", dep=TRUE) は必要ないのが普通であろうが、install.packages("Hotelling", dep=TRUE) は必要な方が多いかもしれない)。Hotelling の T^2 検定は、2 変量分布が 2 群間で異な

るかどうかを調べるので、この場合のように 3 群あったら、2 群ずつ調べて、Holm の方法、FDR 法等で検定の多重性を調整せねばならない。以下のコードを実行すると、図 4.1 が得られ、検定結果を見ると、Y 村と Z 村の間のみ有意水準 5%で身長と体重の 2 変量分布に統計的に有意な差がある（p=0.011）とわかる。

https://minato.sip1c.org/advanced-statistics/scdehot.R(4)

```
library(car)
dataEllipse(v3$HEIGHT, v3$WEIGHT, v3$VG, levels=0.8) #集中楕円描画
library(Hotelling)
Z <- split(v3[,c("HEIGHT","WEIGHT")], v3[,"VG"])
res12 <- hotelling.test(Z[[1]], Z[[2]])
res23 <- hotelling.test(Z[[2]], Z[[3]])
res31 <- hotelling.test(Z[[3]], Z[[1]])
res <- c(res12$pval, res23$pval, res31$pval)
names(res) <- c("X-Y", "Y-Z", "Z-X")
sort(res)*3:1 # Holm の方法で検定の多重性を補正
```

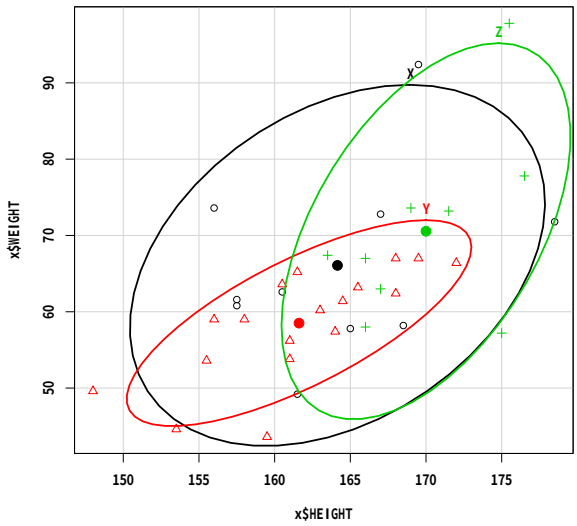


Figure 4.1: 身長と体重の関係について 3 村落の 80%確率楕円

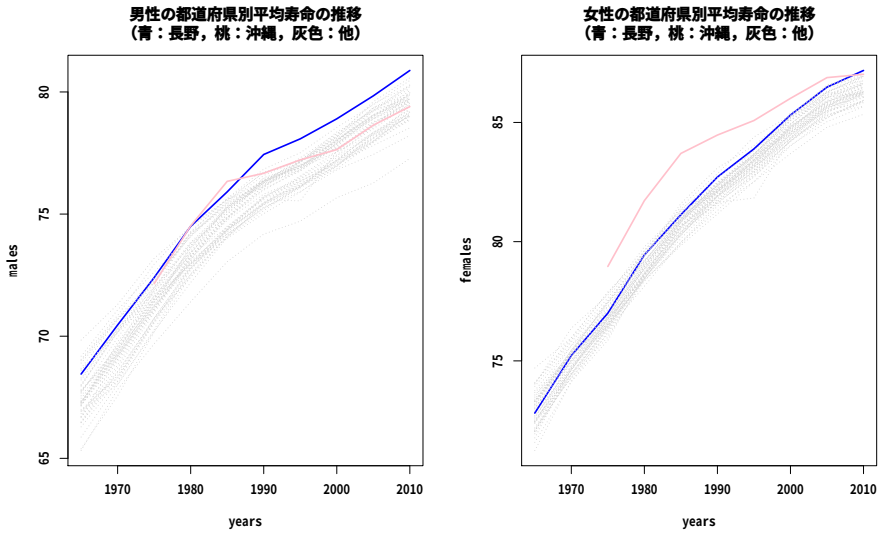
4.4.2 都道府県別生命表からの図示

厚生労働省のサイトで 2013 年 2 月 28 日に公開された、平成 22 年都道府県別生命表の概況 (https://www.mhlw.go.jp/toukei/saikin/hw/life/tdfk10/index.html) の「図表データのダウンロード」から Excel ファイル (https://www.mhlw.go.jp/toukei/saikin/hw/life/tdfk10/dl/zuhyou.xls) をダウンロードして加工したデータを使って、都道府県別平均寿命の推移を示す折れ線グラフと、死因別損失余命の都道府県別プロファイルを示すレーダーチャートを、男女別に作成してみる。

男女別に都道府県別平均寿命の推移を示す（長野と沖縄だけ色を変えて強調した）折れ線グラフを描くコードは以下の通り。

https://minato.sip21c.org/advanced-statistics/e0changes.R

```
e0 <- read.delim("https://minato.sip21c.org/demography/pref-e0-changes.txt",
 fileEncoding="CP932")
males <- t(e0[, 2:11])
colnames(males) <- e0$PREF
females <- t(e0[, 12:21])
colnames(females) <- e0$PREF
COL <- ifelse(e0$PREF=="長野", "blue",
 ifelse(e0$PREF=="沖縄", "pink", "lightgrey"))
LWD <- ifelse(e0$PREF=="長野", 2, ifelse(e0$PREF=="沖縄", 2, 1))
LTY <- ifelse(e0$PREF=="長野", 1, ifelse(e0$PREF=="沖縄", 1, 3))
years <- 1965+0:9*5
windowsFonts(JP1=windowsFont("MS Gothic"),
             JP2=windowsFont("MS Mincho"),
             JP3=windowsFont("Meiryo"),
             JP4=windowsFont("Biz Gothic"))

windows(width=1200, height=800) # for MacOS, quartz() can be used.
par(family="JP3") # to make pdf, family="Japan1" should be used.
# for MacOS, par(family="Japan1") should be used.
layout(t(1:2))
matplot(years, males, type="l", col=COL, lwd=LWD, lty=LTY,
 main="男性の都道府県別平均寿命の推移\n（青：長野、桃：沖縄、灰色：他）")
matplot(years, females, type="l", col=COL, lwd=LWD, lty=LTY,
 main="女性の都道府県別平均寿命の推移\n（青：長野、桃：沖縄、灰色：他）")
```



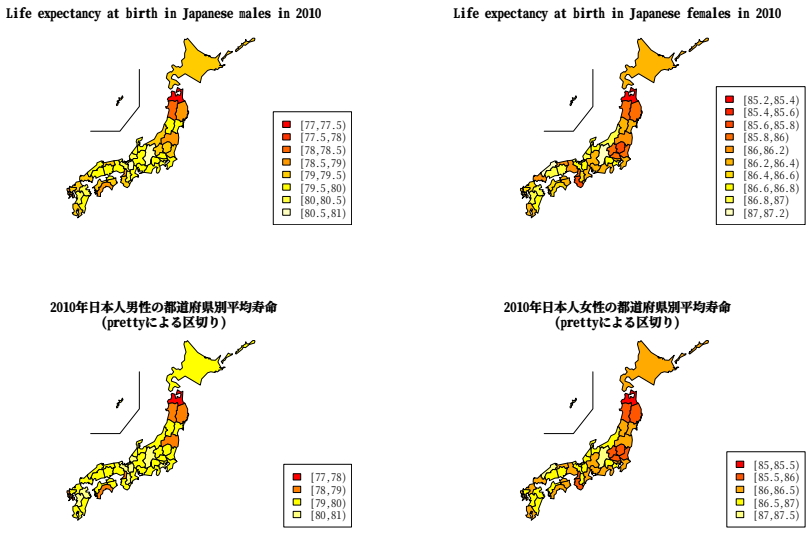
このグラフから読み取れることはそれほど多くないが、1985 年までトップレベルだった沖縄男性の平均寿命が、1990 年から急に伸びが鈍化したこと、長野県男性も 1990 年までの伸びに比べると 1995 年以降は伸びが鈍化していることがわかる。女性については、男性と違って、最近まで沖縄の平均寿命の高さは他都道府県とは段違いだったのに、2005 年に追いつかれ、2005 年から 2010 年には横這いになってしまったことが一目で分かる。数値だけ眺めるよりわかりやすいと思う。

ちなみに、これは折れ線グラフなので、縦軸がゼロから始まっていないことに注意されたい。2010 年の男性の水準には、女性は 1980 年頃には既に到達していた。

NipponMap パッケージの JapanPrefMap() 関数を使うと、都道府県別データからコロプレス図を作ることが簡単にできる。平成 22 年の都道府県別平均寿命を、ヒストグラムの階級を区分する Sturges アルゴリズムまたは pretty() 関数を使って適当に区分し、男女別に地図上で塗り分けるコードは以下である。

https://minato.sip21c.org/advanced-statistics/e0Japan2010.R

```
e0 <- read.delim("https://minato.sip21c.org/demography/pref-e0-changes.txt",
 fileEncoding="CP932")
mec <- cut(e0$e0M.2010, hist(e0$e0M.2010, plot=FALSE)$breaks, right=FALSE)
mec2 <- cut(e0$e0M.2010, pretty(e0$e0M.2010), right=FALSE)
fec <- cut(e0$e0F.2010, hist(e0$e0F.2010, plot=FALSE)$breaks, right=FALSE)
fec2 <- cut(e0$e0F.2010, pretty(e0$e0F.2010), right=FALSE)
mcol <- heat.colors(length(levels(mec)))[as.integer(mec)]
mcol2 <- heat.colors(length(levels(mec2)))[as.integer(mec2)]
fcol <- heat.colors(length(levels(fec)))[as.integer(fec)]
fcol2 <- heat.colors(length(levels(fec2)))[as.integer(fec2)]

windowsFonts(JP1=windowsFont("MS Gothic"),
             JP2=windowsFont("MS Mincho"),
             JP3=windowsFont("Meiryo"),
             JP4=windowsFont("Biz Gothic"))

windows(width=1200, height=800) # for MacOS, quartz() should be used.
par(family="JP4") # for MacOS, par(family="Japan1") should be used.
layout(matrix(1:4, 2, 2))

library(NipponMap)
JapanPrefMap(mcol, main="Life expectancy at birth in Japanese males in 2010")
legend("bottomright", fill=heat.colors(length(levels(mec))), legend=names(table(mec)))
JapanPrefMap(mcol2, main="2010 年日本人男性の都道府県別平均寿命\n(pretty による区切り)")
legend("bottomright", fill=heat.colors(length(levels(mec2))), legend=names(table(mec2)))
JapanPrefMap(fcol, main="Life expectancy at birth in Japanese females in 2010")
legend("bottomright", fill=heat.colors(length(levels(fec))), legend=names(table(fec)))
JapanPrefMap(fcol2, main="2010 年日本人女性の都道府県別平均寿命\n(pretty による区切り)")
legend("bottomright", fill=heat.colors(length(levels(fec2))), legend=names(table(fec2)))
```

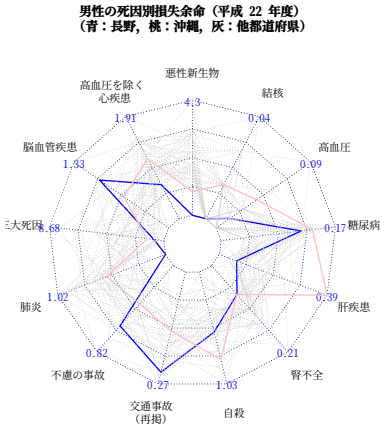


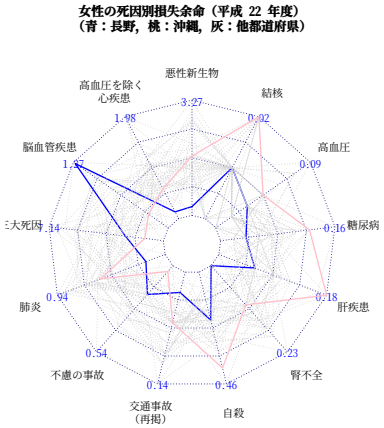
いくつかの指標をプロファイルとして多角形で示すのがレーダーチャートである。

R では fmsb パッケージに radarchart() 関数として実装してある。このデータから都道府県別死因別損失余命プロファイルを（やはり長野県と沖縄県を強調して）描くコードは下記の通り。

https://minato.sip21c.org/advanced-statistics/cdradar.R

```
x <- read.delim("https://minato.sip21c.org/demography/pref-LLY-h22.txt",
 fileEncoding="CP932")
COL <- ifelse(x$PREF=="長野", "blue",
 ifelse(x$PREF=="沖縄", "pink", "lightgrey"))
LWD <- ifelse(x$PREF=="長野", 2, ifelse(x$PREF=="沖縄", 2, 1))
LTY <- ifelse(x$PREF=="長野", 1, ifelse(x$PREF=="沖縄", 1, 3))
VX <- c("悪性新生物","高血圧を除く\n心疾患","脳血管疾患","三大死因",
 "肺炎","不慮の事故","交通事故\n（再掲）","自殺","腎不全","肝疾患",
 "糖尿病","高血圧","結核")
males <- x[,2:14]
females <- x[,15:27]
require(fmsb)
windowsFonts(JP1=windowsFont("MS Gothic"),
             JP2=windowsFont("MS Mincho"),
             JP3=windowsFont("Meiryo"),
             JP4=windowsFont("Biz Gothic"))

windows(width=1200, height=800) # for MacOS, quartz() should be used.
par(family="JP4") # for MacOS, par(family="Japan1") may be used.
layout(t(1:2))
radarchart(males, maxmin=FALSE, pcol=COL, axistype=2, pty=32, plty=LTY,
 plwd=LWD, vlabels=VX,
 title="男性の死因別損失余命（平成 22 年度）\n（青：長野、桃：沖縄、灰：他都道府県）")
radarchart(females, maxmin=FALSE, pcol=COL, axistype=2, pty=32, plty=LTY,
 plwd=LWD, vlabels=VX,
 title="女性の死因別損失余命（平成 22 年度）\n（青：長野、桃：沖縄、灰：他都道府県）")
```





このグラフはいろいろなことを示唆してくれる。一見してわかることは、平均寿

命が男女とも最長の長野県は、男女とも、がんと肺炎による死亡が少ないということだ。一方、脳血管疾患によって失われている余命は比較的大きい。これは、長野県の人は漬け物をよく食べるため、元々塩分摂取量が多く、そのために脳卒中が多かったのを、食生活改善推進員さんが歩き回って塩分摂取量を減らし、そのおかげで脳卒中が減ったと言われているのだが、それでもまだ塩分摂取が高いということかもしれない。ただし、くも膜下出血のリスク因子としては遺伝も大きいので、塩分摂取だけが問題とは言い切れないが。なお、長野県では、男性のみ交通事故によって失われている余命が大きいが、これは子供の交通事故死だと思われる。細くて見通しが悪くて歩道が狭い道路が多いのに外遊びする子供は多いので、飛び出しによる交通事故が比較的多いのであろうことは想像に難くない。沖縄のプロファイルから目立つのは、肝疾患、糖尿病が高いことだ。たぶん飲酒が多いせいだろう。女性のみ結核による損失余命が大きかったが、これは流行があったのかもしれない。

4.4.3 時系列の２つの変数の関係

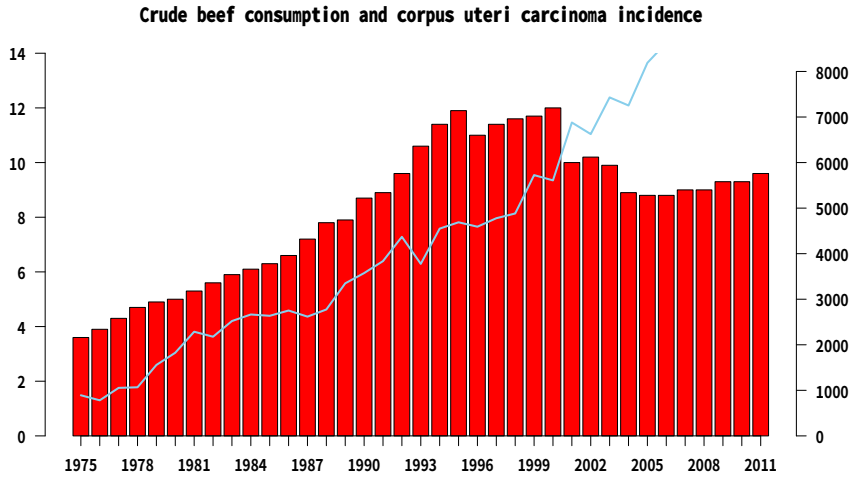
世間では、時系列の２つの変数の推移グラフを重ねて、動きが似ているから関係があるとするロジックが使われることがある。例えば、生活クラブ大阪の 2016 年 2 月の「クラブ通信 Vol.93」の記事（https://osaka.seikatsuclub.coop/excludes/osaka/img/member/bulletin/2016club/club.93.pdf）では、年次を横軸、日本人一人あたりの年間牛肉消費量と子宮体がん発生数を縦軸にとって、前者を棒グラフ、後者を折れ線グラフとして重ね描きして、推移が似ているから関連があるのだと論じている。

日本人一人当たりの年間牛肉消費量は食糧需給表（https://www.e-stat.go.jp/SG1/estat/List.do?lid=000001131797）から、3-7 の中の牛肉というところから Excel のワークシートをダウンロードでき、子宮体がん発生数は、がんセンター（https://ganjoho.jp/professional/statistics/statistics.html）の「2．罹患データ（全国推計値）」から Excel のワークシートをダウンロードできるので、それぞれ該当データを抽出してタブ区切りテキスト形式にしたものを https://minato.sip21c.org/beef-and-corpus-uteri-carcinoma.txt に掲載した。数値からみると、当該グラフで使われている「日本人一人当たりの年間牛肉消費量」は国内消費仕向量の粗食料の値であり（リンク先データでは BEEFCC とした）、歩留まりが考慮されていない。むしろ 1 人当たり供給量（リンク先データでは BEEFSP とした）の方が摂取量には近いと考えられる。リンク先データでは子宮体がん発生数は CUCI とし、年次は YEAR とした。

このデータを読み込んで、生活クラブのサイトと同じものを再現するコードは下記の通りである。

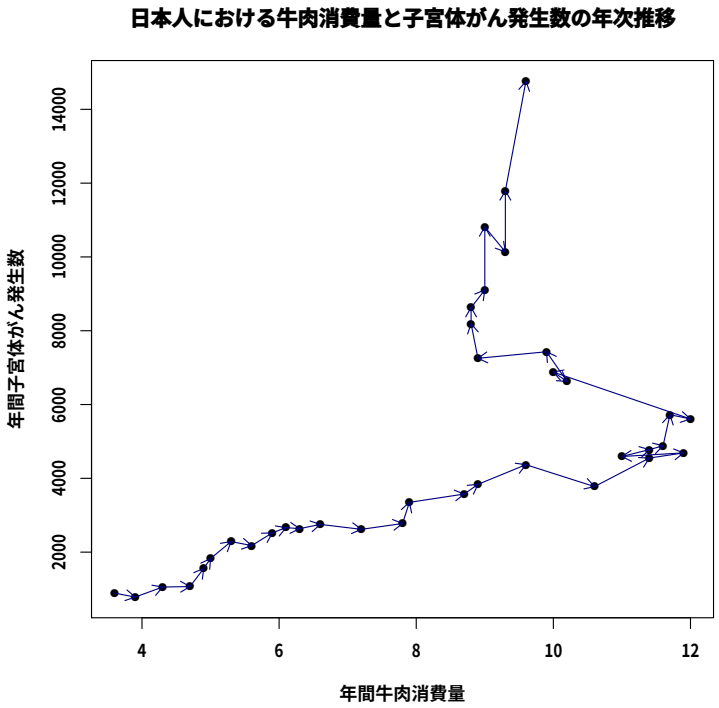
https://minato.sip21c.org/advanced-statistics/beefutecan.R(1)

```
URL <- "https://minato.sip21c.org/beef-and-corpus-uteri-carcinoma.txt"
x <- read.delim(URL)
par(las=1, mar=c(3, 4, 4, 4))
y <- barplot(x$BEEFCC, col="red", ylim=c(0, 14),
 main="Crude beef consumption and corpus uteri carcinoma incidence")
lines(y, x$CUCI/600, col="skyblue", lwd=2)
axis(4, 0:8*5/3, labels=0:8*1000)
axis(1, y, labels=x$YEAR)
```



しかし、これが真の相関関係（ある程度の規則性をもって大小をともにする関係）であるならば、２つの変数間の散布図を描いて年次推移を矢印でつないだ場合に、矢印の傾きと全体の傾向が一致するはずである。こういう推移グラフを描くのも R ならば簡単である。ポイントは [−1] によってベクトルの最初の要素を削除したベクトルを作るところで、それにさえ気づけば、`arrows(x0, y0, x1, y1)` 関数で `(x0, y0)` から `(x1, y1)` への矢印を追記できるので、推移グラフが完成する。

```
# データは同じなので読み込み部分は省略
# URL <- "https://minato.sip21c.org/beef-and-corpus-uteri-carcinoma.txt"
# x <- read.delim(URL)
plot(x$BEEFCC, x$CUCI, type="p", pch=16, xlab="年間牛肉消費量",
 ylab="年間子宮体がん発生数",
 main="日本人における牛肉消費量と子宮体がん発生数の年次推移")
arrows(x$BEEFCC, x$CUCI, c(x$BEEFCC[-1], NA), c(x$CUCI[-1], NA),
 col="navy", length=0.1)
```

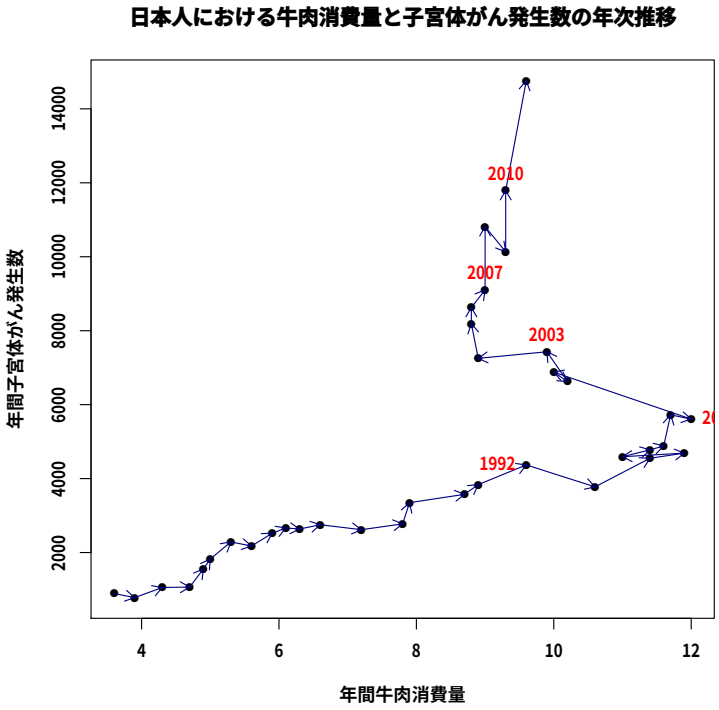


真の正の相関関係であれば、左下と右上を結ぶ方向に推移するはずだが、ほとんどそういう推移になっている年度はない。摂取から発症までの潜伏期間を考えてプロットする年をずらしても、きれいな関係にはならなそうなので、おそらく、これら２つの変数の間の相関は擬似相関と考えられる。

なお、このグラフがアクティブなグラフとして表示されている状態で、identify() 関数を以下のように実行すると、散布図上で点を選んで年の情報を表示させることができる。

```
identify(x$BEEFCC, x$CUCI, x$YEAR, col="red")
```

マウスカーソルが十字型になり、グラフ上の任意の描画点の近傍でクリックすれば、その年を赤い字で（col="red"としているため）書き加えることができる（停止は右クリックから選ぶか、ウィンドウ左上の「停止」から可能）。



x$YEAR のところを x$CUCI とすれば、その点の子宮体がん発生数を数値として表示させることもできる。選択せずに、すべての点に年をオレンジ色で表示させたければ、以下のコードでできる（ごちゃごちゃするのでお勧めしないが）。

```
text(x$BEEFCC, x$CUCI, x$YEAR, col="orange", pos=1)
```

Chapter 5

因子分析

5.1 因子分析と主成分分析

因子分析とは、見かけは主成分分析に似ているので混同されやすいが、指向性は真逆な分析法である。まずこれら２つを区別しよう。

5.1.1 主成分分析とは？

主成分分析においては、観測された多くの変数の分散を、**それらの変数の線形結合として表される互いに独立な**主成分の合成ベクトルとして記述する。主成分は、元のデータがもつ全分散のうち、より多くの割合を説明する順に選択される。2 番目の主成分は、1 番目の主成分と独立という制約の下で、次に多くの割合を説明するように選ばれる。理想的な結果としては、少数の主成分によって元データの分散の大部分が説明され[1]、多くの変数によって高次元空間に位置づけられていた個々のデータ（人を対象として得られた測定値の場合は個人を示す）を、これら少数の主成分の得点によって張られる低次元空間で位置づけるという、**次元の縮小**を行うことができる（図 5.1）。

5.1.2 因子分析とは？

因子分析は、図 5.2 に示す通り、観測された変数（互いに関連をもっている）の背後にあるけれども観測不可能な潜在因子を想定し、それら潜在因子の線形結合によって観測された変数を記述するモデルである。次のようにまとめられる。

真面目な説明 観察された変数の背後に隠れている因子を見いだすこと。この隠れた因子は直接測定できないが、観察された変数の「自然のグルーピング」になっている[2]。

実用的な説明 互いに相関のある変数について、情報を集約して数を減らすこと。この意味では、主成分分析と似ている（向きは逆だが）。

[1] Oxford Handbook for Medical Statistics, 4th Ed. には、通常、２つか３つの主成分で分散の少なくとも 80%が説明される（即ち、第 3 主成分までで累積寄与率が 0.8 を超えるのが普通）、と書かれている。

[2] データセット内のお互いに強く相関する変数のサブセットで、他の変数とは弱い相関をもつ。見つかった因子は、理論的に解釈可能な、隠れた「次元」に対応するはずである。

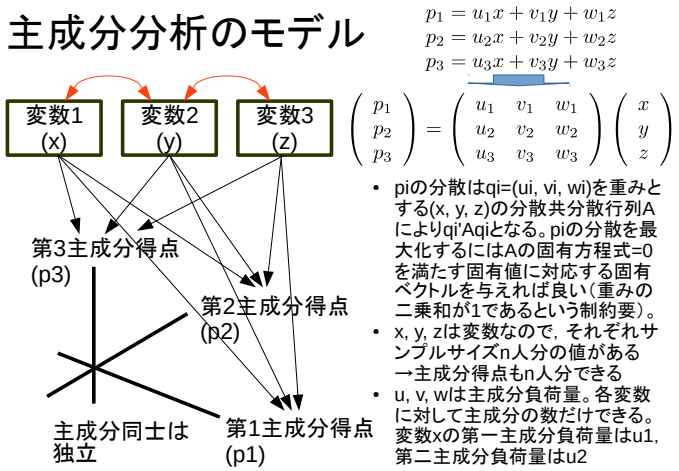


Figure 5.1: 主成分分析のモデル

5.2 主成分分析の基本的な使い方

R で主成分分析を行う関数には、`princomp()` と `prcomp()` がある。どちらも標準で含まれているので、追加パッケージは必要ない。ただし、群馬大学青木繁伸教授が `http://aoki2.si.gunma-u.ac.jp/R/src/pca.R` で公開している `pca()` という関数の方が高機能であり、そちらの方が結果が見やすいかもしれない。

基本的な使い方としては、どちらの関数も、分析したいデータを数値行列として与えるだけで動作する。S との互換性のため、元データから分散共分散行列を計算し、それを使って主成分分析を行うのがデフォルトになっているが、それだと生データの絶対値の大きさに影響されてしまうので、`princomp()` 関数なら `cor=TRUE` オプションをつけて、分散共分散行列でなく相関係数行列を使うようにすべきである。また、`prcomp()` 関数の場合は、`scale=TRUE` オプションをつければ、各変数を標準化してから特異値分解してくれることになり、相関係数行列から出発するのとほぼ同じ結果が得られる。

`princomp()` 関数は素直に固有値と固有ベクトルを使って計算するため、変数の数がサンプルサイズより多いとエラーが出て計算できないが、`prcomp()` 関数は特異値分解によるため、変数の数がサンプルサイズより多くても計算できるという違いがある。

`princomp()` も `prcomp()` も主成分負荷量は出力しない。ただし、結果のオブジェクトを `summary()` に与えると寄与率と累積寄与率は表示される。もう 1 つ表示されるのは standard deviation という値（変数名は `sdev`）で固有値の平方根なので、その 2 乗をとれば各成分の固有値が得られる。

データ行列が `X` だとすると、`summary(princomp(X, cor=TRUE))$sdev^2` とすれば各主成分の固有値が得られる（`summary(prcomp(X, scale=TRUE))$sdev^2` でも良い）。これは `eigen(cor(X))$values` と同じである。

このとき主成分得点は、

```
princomp(X, cor=TRUE)$scores
```

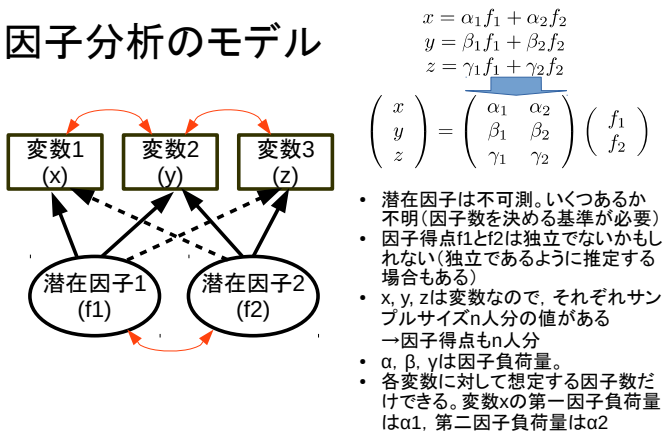


Figure 5.2: 因子分析のモデル

または

```
prcomp(X, scales=TRUE)$x
```

で得られる[3]。なお、princomp() では分散などの計算で分母が N だが prcomp() では N-1 なので、微妙に結果は異なる。つまり、princomp() では主成分得点の分散が固有値となっていて、prcomp() では主成分得点の不偏分散が固有値となっているということ。prcomp() の主成分得点を (N-1)/N の平方根で割れば princomp() が出す主成分得点と一致する。

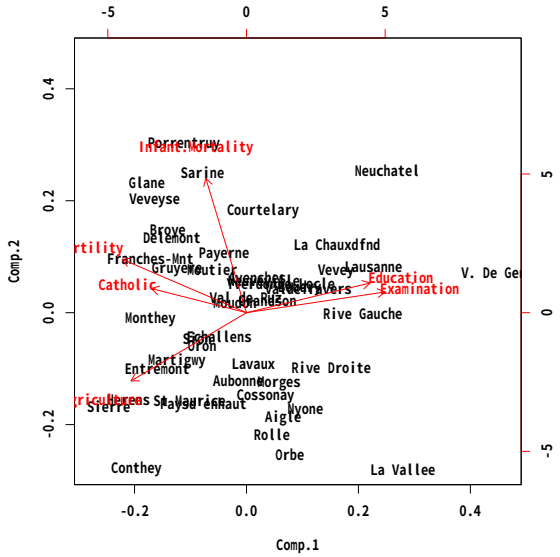
5.2.1 利用例 1

R の組み込みデータ swiss は、1888 年頃のスイスのフランス語を話す 47 州について、標準化された出生力指標（変数名は Fertility、Ig ＝プリンストン研究（詳しくは、https://opr.princeton.edu/archive/pefp/indices.aspx を参照されたい）の有配偶出生力指標で、既婚女性の出生率の生物学的上限と考えられるハテライトの出生率に対する比× 100)、職業として農業に従事している男性の割合（同 Agriculture)、陸軍試験で最高ランクの評価を受けた被徴兵者の割合（同 Examination)、小学校より上の教育歴をもつ被徴兵者の割合（同 Education)、カソリック信者の割合（同 Catholic)、乳児死亡率（同 Infant.Mortality）である。このデータを使って主成分分析を行い、これら 47 州のプロファイルを考えてみるコードを以下に示す。

[3] https://blog.statsbeginner.net/entry/2014/07/27/121214 が参考になる。

```
data(swiss)
spc <- princomp(swiss, cor=TRUE)
biplot(spc)
summary(spc)
summary(spc)$sdev^2
spc$loadings
```

描かれるバイプロットは以下である。ここのコードでは表示されないが、各州の主成分得点を行列として欲しければ、spc$scores で参照可能である。




```
> summary(spc)
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5     Comp.6
Standard deviation     1.7887865 1.0900955 0.9206573 0.66251693 0.45225403 0.34765292
Proportion of Variance 0.5332928 0.1980514 0.1412683 0.07315478 0.03408895 0.02014376
Cumulative Proportion  0.5332928 0.7313442 0.8726125 0.94576729 0.97985624 1.00000000
> summary(spc)$sdev^2
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6
3.1997570 1.1883082 0.8476098 0.4389287 0.2045337 0.1208626
> spc$loadings

Loadings:
                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
Fertility       -0.457  0.322  0.174  0.536  0.383 -0.473
Agriculture     -0.424 -0.412        -0.643  0.375 -0.309
Examination      0.510  0.125                0.814  0.224
Education        0.454  0.179 -0.532               -0.681
Catholic        -0.350  0.146 -0.807         0.183  0.402
Infant.Mortality -0.150  0.811  0.160 -0.527 -0.105

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.167  0.167  0.167  0.167  0.167  0.167
Cumulative Var  0.167  0.333  0.500  0.667  0.833  1.000
```

この結果を表にまとめるときは以下のように負荷量の絶対値が小さいものは省略し、固有値と寄与率、累積寄与率を表示する（第 2 主成分までで十分かもしれないが、ここでは第 6 主成分まで書いた）。

	主成分負荷量					
変数	第 1	第 2	第 3	第 4	第 5	第 6
有配偶出生力指数	−0.457	0.322		0.536	0.383	−0.473
男性農業従事割合	−0.424	−0.412		−0.643	0.375	−0.309
被徴兵者試験高成績割合	0.510				0.814	
被徴兵者中等教育以上割合	0.454		−0.532			−0.681
カソリック信者割合	−0.350		−0.807			0.402
乳児死亡率		0.811		−0.527		
固有値	3.200	1.188	0.848	0.439	0.205	0.121
寄与率	0.533	0.198	0.141	0.073	0.034	0.020
累積寄与率	0.533	0.731	0.873	0.946	0.980	1.000

5.2.2 利用例 2

実際に主成分分析を使って書かれた論文の中にはデータと解析結果が両方書かれているものがある。例として、Tokahoglu S (2012) Determination of trace elements in commonly consumed medicinal herbs by ICP-MS and multivariate analysis. *Food Chemistry*, 134: 2504-8. に掲載されている分析（著者は SPSS を使っている）を R で再現することを試みた。

結果のうち、次に示す Table 4 が主成分分析の結果である。

Table 4. Varimax rotated loadings and communalities for herb samples (n = 30, only those larger than 0.1 are shown).

Element	Principal components				Communalities (h2)
	1	2	3	4	
Cr	**0.917**	−0.182		0.139	0.893
Mn	0.348			**0.766**	0.708
Fe	**0.890**			0.128	0.808
Co	**0.946**				0.895
Ni	**0.869**	0.140	0.119	0.205	0.831
Cu	0.121	0.108	**0.811**	−0.324	0.789
Zn		−0.189	**0.832**	0.258	0.794
Rb		**0.725**		**0.591**	0.875
Sr		**0.898**		−0.139	0.826
Pb	**0.547**	−0.534	0.146		0.606
Explained variance (%)	37.28	17.22	13.96	12.12	80.57

論文には、バリマックス回転し、主成分負荷量の絶対値が 0.1 以上のものを表示したと書かれていた。分散共分散行列だとまったく違う結果になったので、相関係数行列を使っていると思われた。検出限界以下の扱いが不明であるが、`princomp()` と `prcomp()` では検出限界以下をタブ区切りテキストファイルには NA として入力したものを分析時に 0 に置換して処理した。青木繁伸教授の `pca()` 関数では自動的に欠損値を 1 つでも含むケースは除去される。これらのいずれも元論文と若干異なる結果であった。検出限界以下の値に対してペアワイズの除去をするために、`cor()` 関数のオプションで `use="pairwise.complete.obs"`を使って相関係数行列を計算し、それを元に主成分分析を実行できる、`psych` パッケージの `principal()` を適用したところ、元論文と概ね合っている結果（微妙に違うが）が得られたので、おそらく元論文ではペアワイズの除去がなされたと考えられる。以上のコードを示しておく。

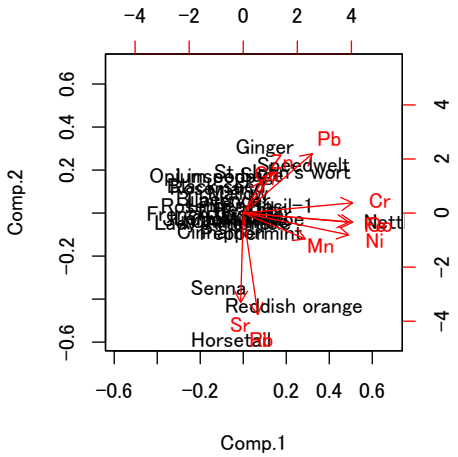
https://minato.sip21c.org/advanced-statistics/MedHerbs.R

```
# source("https://prs.ism.ac.jp/~nakama/AI/AI_UCS2.R") # 日本語 utf8 のため
# par(family="Japan1GothicBBB") # pdf への日本語出力のため
windowsFonts(JP1=windowsFont("MS Gothic"),
 JP2=windowsFont("MS Mincho"), JP3=windowsFont("Meiryo"))
par(family="JP3") # Windows で画面でみるにはこちら。
Herbs <- read.delim("https://minato.sip21c.org/advanced-statistics/MedHerbs.txt")
row.names(Herbs) <- Herbs[, 1] # 最初の変数が薬草名なので行名にコピー
Herbs <- Herbs[, -1] # 薬草名を変数から削除
Herbsc <- Herbs # コピー
Herbs[sapply(Herbs, is.na)] <- 0 # このデータの NA は ND なので 0 を代入
# ただし ND の処理は難しい。検出限界以下はゼロではないので。
summary(res1 <- princomp(Herbs, cor=TRUE))
res1$sdev^2
res1$loadings
biplot(res1)
summary(res2 <- prcomp(Herbs, scale=TRUE, retx=TRUE))
res2$sdev^2
res2$rotation
biplot(res2)
# 違いは princomp では分母が N、prcomp では N-1 であること
# princomp では主成分得点の分散、prcomp では主成分得点の不偏分散が固有値
# 青木先生の関数 pca を読み込む
source("http://aoki2.si.gunma-u.ac.jp/R/src/pca.R", encoding="euc-jp")
res3 <- pca(Herbsc)
library(psych)
resx <- fa.parallel(Herbsc) # 出力する主成分数を決めるため
print(res3, npca=resx$ncomp)
print(res3, npca=4) # 強引に 4 つ出す
plot(res3)
# 手動でリスト単位の欠損値除去
Herbsc.omitNA <- subset(Herbsc, complete.cases(Herbsc))
summary(res1x <- princomp(Herbsc.omitNA, cor=TRUE))
res1x$loadings
# 合成得点を平均ゼロ、分散 1 に標準化するには、固有値の平方根で割ればいい
t(apply(res3$fs, 1, "/", sqrt(res3$eval))) # 主成分得点
t(apply(res1$scores, 1, "/", res1$sdev)) # 一致する
t(apply(res2$x, 1, "/", res2$sdev)) # 若干違う
#
# psych パッケージの principal() を使ってみる。主成分数を 4 にしたのは
# 元論文に合わせるため。それ以外の根拠はない。principal() はデフォルトで
# バリマックス回転する。
# principal() には相関係数行列しか与えられないので、主成分得点は出ない。
library(psych)
C1 <- cor(Herbsc, use="pairwise.complete.obs")
print(resp <- principal(C1, nfactors=4, n.obs=length(Herbsc[, 1]))) # 元論文で主成分が 4 つなの
で
```

まず、princomp() の結果を示す。以下の枠内の通り、絶対値でみると、第 1 主成分負荷量が大きい元素は Cr、Fe、Co、Ni、第 2 主成分負荷量が大きい元素が Rb と Sr、第 3 主成分負荷量が大きい元素が Cu と Zn、第 4 主成分負荷量が大きい元素が Mn と Zn となっており、微妙に違っているが概ね論文に掲載されている表と同じ傾向になっていることがわかる（負荷量の値自体はまるで違うが）。第 4 主成分までの寄与率も 80.56%であり、元論文の表とほぼ同じである。

```
Importance of components:
                          Comp.1    Comp.2    Comp.3
Standard deviation     2.0020096 1.3577846 1.1389361
Proportion of Variance 0.4008042 0.1843579 0.1297175
Cumulative Proportion  0.4008042 0.5851621 0.7148797
                           Comp.4     Comp.5     Comp.6
Standard deviation     0.95244412 0.88821102 0.73826604
Proportion of Variance 0.09071498 0.07889188 0.05450367
Cumulative Proportion  0.80559464 0.88448652 0.93899019
                           Comp.7     Comp.8     Comp.9
Standard deviation     0.52824069 0.39781011 0.34428285
Proportion of Variance 0.02790382 0.01582529 0.01185307
Cumulative Proportion  0.96689402 0.98271930 0.99457237
                          Comp.10
Standard deviation     0.232972709
Proportion of Variance 0.005427628
Cumulative Proportion  1.000000000

> res1$sdev^2
    Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
4.00804224 1.84357898 1.29717535 0.90714981 0.78891881
    Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
0.54503675 0.27903822 0.15825288 0.11853068 0.05427628
```

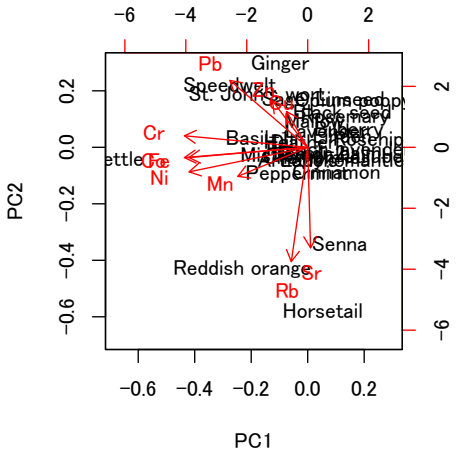
```
> res1$loadings

Loadings:
   Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
Cr  0.461        -0.136  0.107  0.125 -0.110  0.113  0.276
Mn  0.261 -0.160 -0.182 -0.615 -0.520  0.334  0.153 -0.251
Fe  0.445                0.137  0.105 -0.295 -0.468 -0.631
Co  0.461                0.216         0.309  0.137 -0.130
Ni  0.444 -0.135                0.237  0.314         0.409
Cu         0.198  0.722  0.134 -0.392  0.287 -0.373  0.125
Zn  0.131  0.256  0.535 -0.512  0.416 -0.222  0.349 -0.145
Rb        -0.630  0.109 -0.359        -0.291 -0.405  0.365
Sr        -0.555  0.337  0.361 -0.257 -0.235  0.533 -0.166
Pb  0.291  0.369 -0.106        -0.493 -0.569         0.284
   Comp.9 Comp.10
Cr  0.766 -0.224
Mn        -0.146
Fe -0.112 -0.228
Co         0.773
Ni -0.530 -0.408
Cu  0.154
Zn
Rb         0.266
Sr        -0.123
Pb -0.298  0.151

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
SS loadings       1.0    1.0    1.0    1.0    1.0    1.0
Proportion Var    0.1    0.1    0.1    0.1    0.1    0.1
Cumulative Var    0.1    0.2    0.3    0.4    0.5    0.6
               Comp.7 Comp.8 Comp.9 Comp.10
SS loadings       1.0    1.0    1.0     1.0
Proportion Var    0.1    0.1    0.1     0.1
Cumulative Var    0.7    0.8    0.9     1.0
```



prcomp() の結果は以下の通りであり、回転後の負荷量を示すはずの res2$rotation をみても、概ね princomp() と同じ結果になった。

```
> summary(res2 <- prcomp(Herbs, scale=TRUE, retx=TRUE))
Importance of components:
                          PC1    PC2    PC3     PC4     PC5
Standard deviation     2.0020 1.3578 1.1389 0.95244 0.88821
Proportion of Variance 0.4008 0.1844 0.1297 0.09071 0.07889
Cumulative Proportion  0.4008 0.5852 0.7149 0.80559 0.88449
                          PC6    PC7     PC8     PC9    PC10
Standard deviation     0.7383 0.5282 0.39781 0.34428 0.23297
Proportion of Variance 0.0545 0.0279 0.01583 0.01185 0.00543
Cumulative Proportion  0.9390 0.9669 0.98272 0.99457 1.00000

> res2$sdev^2
 [1] 4.00804224 1.84357898 1.29717535 0.90714981 0.78891881
 [6] 0.54503675 0.27903822 0.15825288 0.11853068 0.05427628
```

```
> res2$rotation
           PC1         PC2           PC3          PC4
Cr -0.46083242  0.06364328 -0.1361550921  0.107486068
Mn -0.26136598 -0.16035696 -0.1823044797 -0.615253775
Fe -0.44463160 -0.05663739  0.0001361183  0.136786984
Co -0.46117884 -0.05662732 -0.0280148314  0.216279962
Ni -0.44365438 -0.13532975  0.0532524244  0.071228437
Cu -0.07913501  0.19762575  0.7220481058  0.133815362
Zn -0.13138608  0.25554318  0.5354282736 -0.511640878
Rb -0.06136051 -0.63034651  0.1085422052 -0.359462244
Sr  0.01080989 -0.55467009  0.3369436981  0.361134433
Pb -0.29103660  0.36898876 -0.1060982013  0.005850113
           PC5        PC6         PC7        PC8         PC9
Cr -0.12549872  0.1096002 -0.11309434  0.2757762 -0.76584041
Mn  0.51993276 -0.3340441 -0.15330073 -0.2507013 -0.06783438
Fe -0.10536233  0.2951697  0.46795224 -0.6313140  0.11197090
Co -0.08484872 -0.3090026 -0.13747358 -0.1304947  0.02462569
Ni -0.23723424 -0.3138834 -0.08751828  0.4093297  0.52995697
Cu  0.39195326 -0.2872190  0.37327270  0.1252577 -0.15405082
Zn -0.41622196  0.2222648 -0.34873144 -0.1448879  0.01531580
Rb -0.06080207  0.2908507  0.40496476  0.3648576 -0.04006951
Sr  0.25723944  0.2350568 -0.53290082 -0.1657695 -0.01525029
Pb  0.49254674  0.5687966 -0.09615068  0.2840365  0.29846994
          PC10
Cr  0.22444386
Mn  0.14605493
Fe  0.22821852
Co -0.77287087
Ni  0.40752896
Cu  0.02336532
Zn -0.05909749
Rb -0.26625092
Sr  0.12310930
Pb -0.15071856
```



一方、測定限界以下を 0 にせずに欠損値としてリスト単位で除去する（＝ 1 つでも欠損値があれば、その薬草データごと除去する）青木先生の pca() の結果は、fa.pararrel() では適切な主成分数が 2 となったが、より多くの主成分について結果を表示しても Contribution が増えるだけで、第 1 主成分や第 2 主成分についての負荷量や寄与率は変わらないので、4 つの主成分について負荷量と寄与率を下表に示す。この表示だと負荷量の絶対値は元論文の値に近づくが、第 2 主成分が Rb と Sr ではなく Mn と Rb になり、第 3 主成分が Cu と Zn でなく Cu と Sr になり、第 4 主成分が Mn と Rb でなく Zn と Rb になるという大きな違いが出てしまうので、おそらく元論文の欠損値処理はリスト単位の除去ではない。なお、主成分には名前をつける必要はないので、ここでも PC1 などとしている。

	PC1	PC2	PC3	PC4	Contribution
Cr	-0.919	-0.168	0.120	-0.076	0.893
Mn	-0.504	0.628	0.041	-0.015	0.650
Fe	-0.890	-0.164	0.025	0.047	0.822
Co	-0.921	-0.053	-0.139	-0.122	0.885
Ni	-0.900	0.030	-0.083	0.115	0.832
Cu	0.029	-0.328	-0.808	0.242	0.820
Zn	-0.050	-0.322	-0.060	0.883	0.889
Rb	-0.311	0.770	0.080	0.440	0.890
Sr	-0.108	0.298	-0.824	-0.308	0.875
Pb	-0.477	-0.545	0.196	-0.204	0.605
Eigenvalue	3.889	1.644	1.423	1.204	
Contribution	0.389	0.164	0.142	0.120	
Cum.contrib.	0.389	0.553	0.696	0.816	

手動でリスト単位の除去を行い、princomp() を使って計算した結果も、負荷量の絶対値は測定限界以下にゼロを入れた場合と似ているが、主成分ごとに負荷量の高い元素をみると、青木先生の pca() を使った場合と同じく、元論文のパタンと大きく食い違っているので、やはりリスト単位の除去ではないと考えられる。

そこで、測定限界以下を欠損値としてペア単位の除去（変数 2 つずつの組合せごとに、どちらかが欠損ならば、その 2 つの変数間の相関係数の計算からのみ除去）をして相関係数行列を求め、それを入力にした psych パッケージの principal() 関数

の結果は以下のように得られた。これはほぼ論文に示されている結果と一致しているので（微妙に違うが）、同じ方法と考えて良いだろう。

```
Principal Components Analysis
Call: principal(r = C1, nfactors = 4, n.obs = length(Herbsc[, 1]))
Standardized loadings (pattern matrix) based upon correlation matrix
    RC1   RC2   RC3   RC4   h2   u2 com
Cr 0.92 -0.14  0.04  0.11 0.89 0.11 1.1
Mn 0.40 -0.08 -0.02  0.64 0.58 0.42 1.7
Fe 0.89  0.08  0.09  0.10 0.82 0.18 1.1
Co 0.92  0.06  0.03  0.11 0.86 0.14 1.0
Ni 0.85  0.14  0.18  0.27 0.84 0.16 1.4
Cu 0.11  0.21  0.76 -0.41 0.80 0.20 1.8
Zn 0.07 -0.23  0.87  0.21 0.85 0.15 1.3
Rb 0.03  0.64 -0.04  0.65 0.84 0.16 2.0
Sr 0.02  0.94 -0.05 -0.07 0.89 0.11 1.0
Pb 0.60 -0.35 -0.09 -0.33 0.60 0.40 2.3

                       RC1  RC2  RC3  RC4
SS loadings           3.75 1.57 1.38 1.28
Proportion Var        0.37 0.16 0.14 0.13
Cumulative Var        0.37 0.53 0.67 0.80
Proportion Explained  0.47 0.20 0.17 0.16
Cumulative Proportion 0.47 0.67 0.84 1.00

Mean item complexity =  1.5
Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is  0.08
 with the empirical chi square  17.72  with prob <  0.088

Fit based upon off diagonal values = 0.95
```

この表の h2 は、pca() の Contribution と同じで、いわゆる共通性 (communality) を示すものとして元論文に掲載されている値になる。これを元論文と同じく絶対値が 0.1 以上のものだけ残して表にしたものを表 5.1 に示す。

5.3 因子分析の基本的な使い方

入力データ ある程度のサンプルサイズと大きな変数をもつ数値行列で、通常、サンプルサイズは **300 より多い。 変数数に対する対象者の人数の比**は、通常、**2:1 から 10:1** の範囲をとる。原則として変数は正規分布に従うべきでだし、外れ値は含まない方がよい。他の変数と関連のない変数は分析に含めるべきではない。お互いに相関係数 1.0 の変数は含めることができない。どちらかを除外するか、適切であれば両者の和をとって合成変数として用いることは可能である。

出力 (1) 因子負荷量は、各変数がその元になる潜在因子と相関している程度を意味する（その際、さまざまな回転が用いられる[4]）、(2) 因子得点は、通常、各個人の応答と因子負荷量の積の和で（ただし複数の計算法があり、どの方法が最

[4]最初の因子負荷量は、第一因子への負荷を最大にするように計算されるので、たいていの変数が 1 つ以上の因子に対して高い負荷量をもってしまい、因子の解釈が難しくなる。そこで、適切な**回転**をすると、この問題が解決することが多い。

Table 5.1: 薬草中の元素含有量についての主成分分析結果

	Loadings				
Elements	PC1	PC2	PC3	PC4	Communality (h^2)
Cr	0.92	−0.14		0.11	0.89
Mn	0.40			0.64	0.58
Fe	0.89				0.82
Co	0.92				0.86
Ni	0.85	0.14	0.18	0.27	0.84
Cu	0.11	0.21	0.76	−0.41	0.80
Zn		−0.23	0.87	0.21	0.85
Rb		0.64		0.65	0.84
Sr		0.94			0.89
Pb	0.60	−0.35		−0.33	0.60
Statistics	PC1	PC2	PC3	PC4	
Eigenvalues	3.75	1.57	1.38	1.28	
Proportion Variance	0.37	0.16	0.14	0.13	
Cumulative Variance	0.37	0.53	0.67	0.80	

適かについて統一見解はない)、各個人の特性がどの程度その因子によって説明されるかを示す。

回転 回転の方法は２つに大別される。直交回転は、因子間の独立性を保ったまま因子ベクトルを回転させるが、斜交回転では因子間に相関が出てもいいことにしている。因子が理論的に相互依存を許してもいいときに、後者を考えるべきである。前者には最もよく使われていて単純なバリマックス回転が含まれる。バリマックス回転は、因子ごとの分散を最大化する。後者にはプロマックス回転やオブリミン回転が含まれる。

因子分析のための道具 スクリープロット、バートレットの球面性検定、カイザー・マイヤー・オルキンのサンプリング適切性基準、平行分析 (Parallel Analysis) が便利。因子数がうまく決定できたら、各因子に含まれる変数が単一軸の加法的スコアになっているかどうかをチェックするために、クロンバックの α 係数を計算する（通常、それらの因子の和が信頼できるスコアであるためには、クロンバックの α が 0.7 より大きくなければいけない)。

推定された因子を解釈する際には、**因子に適切な名前（意味）をつけることが必要**である。因子がうまく推定できたと判定するには、因子負荷量が高い変数が少なくとも３つあるべきである。もし１つか２つしか因子負荷量が高い変数がないときは、因子数が多すぎるか、元の変数間に多重共線性が存在する可能性がある。

5.4 因子分析の基本モデル

300 人で変数 10 個（$X_1, X_2, ..., X_{10}$）の場合を考えよう。これら 10 個の変数の背後に、もし２個の潜在因子（F_1 と F_2）があるとしたら、各変数は、これらの因子によって次のように説明される。

$$X_1 = \beta_{1\cdot1}F_1 + \beta_{2\cdot1}F_2 + \epsilon_1$$

$$X_2 = \beta_{1\cdot2}F_1 + \beta_{2\cdot2}F_2 + \epsilon_2$$

$$\vdots$$

$$X_{10} = \beta_{1\cdot 10}F_1 + \beta_{2\cdot 10}F_2 + \epsilon_{10}$$

ここで、β は、各変数と潜在因子との相関を意味し、これを**因子負荷量 (Factor loadings)** と呼ぶ。ϵ は誤差分散を意味する。言い換えると、推定された因子では説明できなかった**独自性 (uniqueness)** でもある。なお、独自性を 1 から引いたものを共通性 (communality) という。後述する `rela` パッケージの関数では、共通性が出力される。しかし、潜在因子 F_1 と F_2 は測定された値ではない。だから、我々は、主因子法、最小残差法、最尤法などの様々な方法で、反復計算させながら推定しなくてはいけない[5]。

回転する前は、因子 F_1 と F_2 は独立と仮定されている。いま、n 番目（n は区間 [1, 300] の整数）の人の i 番目の変数の値を $X_i(n)$ と書くと、その人の因子得点（ここでは $FS_1(n)$ と $FS_2(n)$）は、次のように得られる（ただし、これは最も単純な方法である。因子得点として提案されている指標値は、この他にもいくつかある）。計算に使う変数は、β の絶対値が十分大きい（通常、0.3 とか 0.4、あるいは 0.5 以上とする）ものに限るのが普通。

$$FS_1(n) = \sum_{i=1}^{10} \beta_{1\cdot i} X_i(n)$$

$$FS_2(n) = \sum_{i=1}^{10} \beta_{2\cdot i} X_i(n)$$

5.5 いくつの因子を推定すべきか？

この問題には以下のようにいくつかの基準が提案されているが、100%これが良いという検定法などは存在しない。

スクリープロットを描く 最初に可能な限り多くの因子を仮定して因子分析を行い、各因子によって説明される分散を代表するものとしての固有値（あるいは同じ意味で因子負荷量の二乗和）を、大きい順に線でつないだ折れ線グラフがスクリープロットである。折れ線が急に激しく落ち込む変数があれば、その直前が適切な因子数と考えられる。

パラレル分析をする 実際のスクリープロットを、ランダムにリサンプルしたデータから計算したスクリープロットと比較する。２つのプロットが交差する点が適切な因子数であると考える。

固有値が 1 を超えている間 固有値が 1 を超えている間は、変数 1 つよりも情報量が多いと考えられるので。

5.6 因子分析の適切性をチェックする

因子分析の適切性をチェックするための方法がいくつかある。

サンプルサイズの適切性の基準 サンプルサイズは 50 では非常に乏しい (very poor)。100 でも乏しい (poor)。200 ならまあまあ (fair)、300 なら十分 (good)、500 なら非常に良い (very good)。1,000 を超えたら極めて優れている (excellent) といえる (Comfrey and Lee, 1992, p.217)。

[5] 主成分分析では、各主成分は、測定された変数の線形結合として定式化されるので、反復推定は必要ない。

KMO と MSA KMO とは、Kaiser-Meyer-Olkin が提唱した因子分析全体についてのサンプリング適切性基準であり、MSA とは Measures of Sampling Adequacy の頭語で、それぞれの変数についての個別のサンプリング適切性基準である。データセットの中に、十分な数の因子が存在するかどうかを示す指標値である。技術的には、変数間の相関係数の偏相関係数に対する比を計算する。もし偏相関係数が生の相関係数と同じような値なら、それらの変数は互いに分散をあまり共有していないことを意味する。KMO の範囲は 0.0 から 1.0 で、0.5 以上が望ましい[6]。また、MSA が 0.5 未満の変数は、その変数がどの因子グループにも属していないことを示すので、因子分析から除くべきである。

群馬大学の青木繁伸教授は、http://aoki2.si.gunma-u.ac.jp/R/kmo.html で、KMO と MSA を計算するための次の関数定義を公表している[a]。

```
kmo <- function(x)
{
        x <- subset(x, complete.cases(x)) #欠損値除去
        r <- cor(x) # 相関係数行列を r に付値
        r2 <- r^2 # 相関係数行列の各要素を 2 乗した値を r2 に付値
        i <- solve(r) # 相関係数行列 r の逆行列を求めて i に付値
        d <- diag(i) # 逆行列 i の対角成分を d に付値
        p2 <- (-i/sqrt(outer(d, d)))^2 # 偏相関係数の 2 乗を計算し p2
に付値
        diag(r2) <- diag(p2) <- 0 # r2 と p2 の対角成分を 0 にする
        KMO <- sum(r2)/(sum(r2)+sum(p2))
        MSA <- colSums(r2)/(colSums(r2)+colSums(p2))
        return(list(KMO=KMO, MSA=MSA))
}
```

[a] source("http://aoki2.si.gunma-u.ac.jp/R/src/kmo.R", encoding="euc-jp") で使えるようになる。

バートレットの球面性検定 変数間の相関が偶然期待されるより大きいという仮説を検定する。技術的には行列が単位行列であるかどうかを検定する。p 値が有意である場合、対角以外のすべての相関がゼロであるという帰無仮説が棄却される。

[6] Kaiser (1974) の提案によれば、0.5 未満では不適切、0.5 以上 0.6 未満は悲惨なレベル (miserable)、0.6 以上 0.7 未満は良くも悪くもなく (mediocre)、0.7 以上 0.8 未満は並 (middling)、0.8 以上 0.9 未満は賞賛に値し (meritorious)、0.9 以上なら極めて優れている (marvelous)。

バートレットの球面性検定についても、群馬大学の青木繁伸教授が http://aoki2.si.gunma-u.ac.jp/R/Bartlett.sphericity.test.html で次の関数定義を公表している[a]。

```
Bartlett.sphericity.test <- function(x)
{
       method <- "Bartlett's test of sphericity"
       data.name <- deparse(substitute(x))
       x <- subset(x, complete.cases(x)) # 欠損値除去
       n <- nrow(x)
       p <- ncol(x)
       chisq <- (1-n+(2*p+5)/6)*log(det(cor(x)))
       df <- p*(p-1)/2
       p.value <- pchisq(chisq, df, lower.tail=FALSE)
       names(chisq) <- "X-squared"
       names(df) <- "df"
       return(structure(list(statistic=chisq, parameter=df,
              p.value=p.value, method=method, data.name=data.name),
              class="htest"))
}
```

[a] source("http://aoki2.si.gunma-u.ac.jp/R/src/Bartlett.sphericity.test.R", encoding="euc-jp") で使えるようになる。

5.7 R で因子分析を実行するための関数

以下に説明するように、追加パッケージとして rela、psych、sem を用いるので、

```
install.packages("rela", dep=TRUE)
install.packages("psych", dep=TRUE)
install.packages("sem", dep=TRUE)
```

として、予めインストールされたい。

factanal この関数は標準でインストールされる。因子負荷量を計算するのに最尤法を用いる。推定すべき因子数は明示的に指定せねばならない。バリマックス回転とプロマックス回転が可能である。入力データは行列またはデータフレーム。

paf この関数は **rela** パッケージに含まれているので、**rela** パッケージをインストールし、使用前にメモリにロードする必要がある。因子負荷量を計算するのに主因子法を用いる。適切な因子数は、固有値の基準によって自動的に決定され（固有値をいくつ以上にするかは、eigencrit=オプションで指定できる。デフォルトは 1 である）、KMO と MSA が自動的に計算されるので、初心者用と言われている。回転は提供されていない。入力データは行列。

fa この関数は **psych** パッケージに含まれている。fm=オプションで因子負荷量の計算方法を指定できる（"minres"で最小残差法、"ml"で最尤法、"pa"で主因子法）。推定する因子数は nfactors=オプションで指定せねばならない。rotate=オプションでさまざまな回転方法を指定できる（"none"、"varimax"、"quartimax"、"bentlerT"、"geominT"、"oblimin"、"simplimax"、"bentlerQ"、"geominQ"、"cluster"が可能）。

alpha この関数は **psych** パッケージに含まれている。クロンバックの α 係数を計算する。

cortest.bartlett この関数も **psych** パッケージに含まれている。バートレットの球面性検定を実行する。

fa.parallel この関数も **psych** パッケージに含まれている。パラレル分析を実行し、返り値として、`$nfact` に推定すべき適切な因子数を返す。

sem 確証的因子分析 (confirmatory factor analysis; CFA) には、**sem** パッケージを用いることができる。もちろん sem は構造方程式モデリングのパッケージであり、CFA 以上のことができる。詳しくは次章で触れる。

5.8 エコポイントデータを使った分析例

データを使って実例を示そう。`https://minato.sip21c.org/advanced-statistics/ecopx.txt` は、`https://minato.sip21c.org/humeco/ecopoint.html` に示したエコポイントチェック（図 5.3）への回答[7]を適当に加工した、タブ区切りテキストデータである。

年齢	10代				
性別	男性				
同居人の人数(本人を含む)	1人				
以下の取り組みについて	いつも	大体	時々	たまに(A)	皆無(B)
1. 新聞・雑誌をリサイクルに出している	○	○	◉	○	○
2. 古紙100%のトイレットペーパーを使用している	○	○	◉	○	○
3. 飲料容器やトレーをリサイクルに出している	○	○	◉	○	○
4. 買い物袋を持参している	○	○	◉	○	○
5. 着る服で調節して，冷暖房をできるだけ控えている	○	○	◉	○	○
6. 食材は適量を買い，期限切れで捨てないようにしている	○	○	◉	○	○
7. 風呂は家族で続けて入り，二度炊きをしないようにしている(C)	○	○	◉	○	○
8. 風呂の水を洗濯等に利用している	○	○	◉	○	○
9. 車のアイドリングストップを行っている(D)	○	○	◉	○	○
10. マイカーを避けて公共交通を利用している	○	○	◉	○	○
11. 太陽熱温水器を利用している	○	○	◉	○	○
12. 家電製品は省エネ型以外は買わないようにしている	○	○	◉	○	○
13. 米のとぎ汁は流さずに有効利用している	○	○	◉	○	○
14. 油を拭き取ってから皿を洗っている	○	○	◉	○	○
15. 塩ビ系のプラスチック(食品用ラップなど)を購入しないようにしている	○	○	◉	○	○
16. 洗剤として合成洗剤でなく石けんを使っている	○	○	◉	○	○
17. 洗剤を量って適量使用している	○	○	◉	○	○
18. 除草剤や殺虫剤を使わないように気をつけている	○	○	◉	○	○
19. 車のバッテリーや電池類を適正処理している	○	○	◉	○	○
20. トイレや風呂場の強力な洗浄剤を利用しないようにしている	○	○	◉	○	○
21. 有機溶剤(シンナーやベンジン)を利用しないようにしている	○	○	◉	○	○
22. 有機農産物を選んでいる	○	○	◉	○	○
23. 地場の農産物を選んでいる	○	○	◉	○	○
24. 早寝・早起きを心がけている	○	○	◉	○	○
25. たばこを吸わないようにしている	○	○	◉	○	○

Figure 5.3: エコポイントチェックの入力フォーム

[7]かつて web サイトで cgi を使ってくださった、大勢の匿名の皆様に感謝申し上げる。

エコポイントとは、高月紘（編著）『自分の暮らしがわかるエコロジー・テスト：環境問題は生活のエコ度チェックから』講談社ブルーバックスに提示されている、どの程度「環境に優しい」暮らしをしているかを示す尺度である。評点の重み付けには、環境研のコンパラティブ・リスクアセスメントの結果を使っており、ある程度妥当な評価尺度と考えられる。この質問紙を web から回答できるように cgi 化し、不特定多数から回答を得た。

エコポイント総合点（変数名 EP）が 99.8 点満点（計算上の丸め誤差のため）、温暖化問題エコポイント（GW）が 14.3 点満点、廃棄物問題エコポイント（Waste）が 24.6 点満点、水質汚染問題エコポイント（Water）が 15.6 点満点、大気汚染問題エコポイント（Air）が 21.1 点満点、有害化学物質問題エコポイント（Chem）が 24.2 点満点で表示される。

書籍によれば、若い層を中心とした対象者 356 人の平均が 42.6、環境問題の講演を聞きに来た人たち 182 人の平均が 48.1 という指標である。著者は、環境にやさしい人としては 60 点以上必要で、30 点以下だったら環境面ではかなり問題のあるライフスタイルとしており、低い場合は、どの行動パタンにとくに問題があるのかをチェックすることが薦められている。

5.8.1 クロンバックの α 係数の計算

このデータを eco というデータフレームに読み込み、まずは計算された 5 つのエコポイント得点について、クロンバックの α 係数を計算してみる。ここで、1 つのデータに対して多くの解析をするので、専用のディレクトリを作ると良い。ここでは c:/work/lecture/kobe/advstat2/というディレクトリを作り、ここに https://minato.sip21c.org/advanced-statistics/ecopx.txt をダウンロードしてから、RStudio を起動し、File の New project から、Existing Directory として e:/work/lecture/kobe/advstat2/を選択した。こうすると、RStudio 終了時に、自動的にそのときの環境がこのディレクトリの advstat2.Rproj というファイルに保存される。次回からは、このファイルをダブルクリックするだけで自動的に RStudio が起動し、前回最後に操作していたときの状態が復元される。

次の枠内に示すコードをこのディレクトリにダウンロードして右下ペインから選べば左上にスクリプトエディタ画面が開くが、File の New File の R script で白紙のスクリプトエディタ画面を開いて、ブラウザ等で開いたコードをコピーアンドペーストしても良い。スクリプトエディタウィンドウ右上の Source というボタンをクリックすると、自動的に 5 つのエコポイント得点それぞれについて、クロンバックの α 係数と 95%信頼区間が計算される。ちなみに、alpha() 関数の返り値には推定値（raw_alpha）と漸近標準誤差（ase）は含まれているが、普通に実行すると表示される信頼区間の上限と下限の計算は、print.psych() に含まれている。以下のコードでは、GAC() という関数を定義して、alpha() の結果のうち推定値と 95%信頼区間の下限と上限だけを返すようにした。1.96 は言うまでもなく正規分布の 97.5%点、即ち qnorm(0.975) であり、90%信頼区間が欲しいときは 1.96 の部分を qnorm(0.95) とすれば良い。

https://minato.sip21c.org/advanced-statistics/ecopxc.R

```
eco <- read.delim("ecopx.txt")
# 前処理
eco$NAGE <- factor(eco$AGE+1,
  labels=c("10-19","20-29","30-39","40-49","50-59","60-69","70-"))
eco$NSEX <- factor(eco$SEX+1, labels=c("M","F"))
warming <- eco[, c("FAMSIZE","Q05","Q07","Q08","Q11","Q24")]
waste <- eco[, c("FAMSIZE","Q01","Q02","Q03","Q04","Q06")]
water <- eco[, c("FAMSIZE","Q13","Q14","Q16","Q17","Q20")]
air <- eco[, c("FAMSIZE","Q09","Q10","Q12","Q23","Q25")]
chem <- eco[, c("FAMSIZE","Q15","Q18","Q19","Q21","Q22")]
ecopoint <- eco[, c("FAMSIZE","Q05","Q07","Q08","Q11","Q24",
 "Q01","Q02","Q03","Q04","Q06","Q13","Q14","Q16","Q17","Q20",
 "Q09","Q10","Q12","Q23","Q25","Q15","Q18","Q19","Q21","Q22")]
library(psych)
# αと信頼区間を得るための関数定義
GAC <- function(Z) { # Get alpha /w 95 percent confidence intervals
 ZA <- alpha(Z)
 Raw <- ZA$total$raw_alpha
 Ase <- ZA$total$ase
 return(c(Raw-1.96*Ase, Raw, Raw+1.96*Ase))
}
all <- cbind(GAC(warming[,-1]), GAC(waste[,-1]), GAC(water[,-1]),
             GAC(air[,-1]), GAC(chem[,-1]), GAC(ecopoint[,-1]))
print(all)
```

結果を見ると、クロンバックの α 係数は、全項目を使ったエコポイントとしては 0.84 [0.81-0.88] と十分に高いが、温暖化領域 0.41 [0.28-0.53]、廃棄物領域 0.61 [0.52-0.71]、水領域 0.69 [0.60-0.78]、大気領域 0.43 [0.32-0.55]、化学物質領域 0.66 [0.57-0.75] であり、各領域は 0.7 以上という基準に達していない。おそらく多様な回答者に対して設問が微妙な答えにくさを含んでいるためと、法制の影響などもあるものと思われるが、尺度としての信頼性は十分でない。そこで、単身者と２人以上で生活している人で構造が違う可能性を考え、それぞれサブセットを作って分析してみたが、大差なかった（図 5.4）。

ecopxc.R の続き

```
# for single household
single <- cbind(
GAC(subset(warming, FAMSIZE==1)[,-1]), GAC(subset(waste, FAMSIZE==1)[,-1]),
GAC(subset(water, FAMSIZE==1)[,-1]), GAC(subset(air, FAMSIZE==1)[,-1]),
GAC(subset(chem, FAMSIZE==1)[,-1]), GAC(subset(ecopoint, FAMSIZE==1)[,-1]))
print(single)
# for other household
others <- cbind(
GAC(subset(warming, FAMSIZE>1)[,-1]), GAC(subset(waste, FAMSIZE>1)[,-1]),
GAC(subset(water, FAMSIZE>1)[,-1]), GAC(subset(air, FAMSIZE>1)[,-1]),
GAC(subset(chem, FAMSIZE>1)[,-1]), GAC(subset(ecopoint, FAMSIZE>1)[,-1]))
print(others)
# まとめる
MX <- rbind(all[2,], single[2,], others[2,])
colnames(MX) <- c("温暖化","廃棄物","水","大気","化学物質","総合")
rownames(MX) <- c("全体","単独世帯","他の世帯")
UX <- rbind(all[3,], single[3,], others[3,]) # 95\%信頼区間の上限
# cairo_pdf("ecopxc.pdf")
# source("https://prs.ism.ac.jp/~nakama/AI/AI_UCS2.R")
# par(family="Japan1Gothic", las=1)
par(family="sans", las=1) # Windows で画面表示ならこれで良い
ii <- barplot(MX, beside=TRUE, ylim=c(0,1), col=1:3)
arrows(ii, as.vector(MX), ii, as.vector(UX), angle=90, length=0.1)
legend("topleft", legend=rownames(MX), fill=1:3, cex=0.6)
# dev.off()
```

5.8.2 探索的因子分析を試してみる

各質問項目の選択肢に与えたスコアの重みはかつて環境省で行われたコンパラティブ・リスクアセスメント (CRA) の結果によるので、それは生かすことにして[8]、しかし各下位尺度のクロンバックの α 係数が低いので、おそらく回答者の違いや時代の違いにより、因子構造が想定と合っていないのだと判断し、Q01 から Q25 のデータを探索的因子分析してみる。

https://minato.sip21c.org/advanced-statistics/ecofactor.R

```
eco.raw <- eco[,4:28]
source("http://aoki2.si.gunma-u.ac.jp/R/src/kmo.R", encoding="euc-jp")
kmo(eco.raw)
library(psych)
cortest.bartlett(eco.raw)
print(res1 <- fa.parallel(eco.raw))
print(res2 <- fa(eco.raw, fm="minres", nfactors=res1$nfact,
 rotate="quartimax"))
res2$loadings
```

群馬大学青木繁伸教授の関数で KMO や MSA を出すと概ね 0.8 以上あるので十分である。cortest.bartlett() の結果、p 値はほぼ 0 であり、回答に相関がないという帰無仮説が棄却されるので因子分析に適したデータといえる。fa.parallel() の結果、“Parallel analysis suggests that the number of factors = 5 and the number of

[8]ただし、本当にこのスコアで良いのか、むしろ、元々のスコアのまま標準化した方が良いのではないかという問題はあるので、その辺りは今後丁寧に検討すべきである。

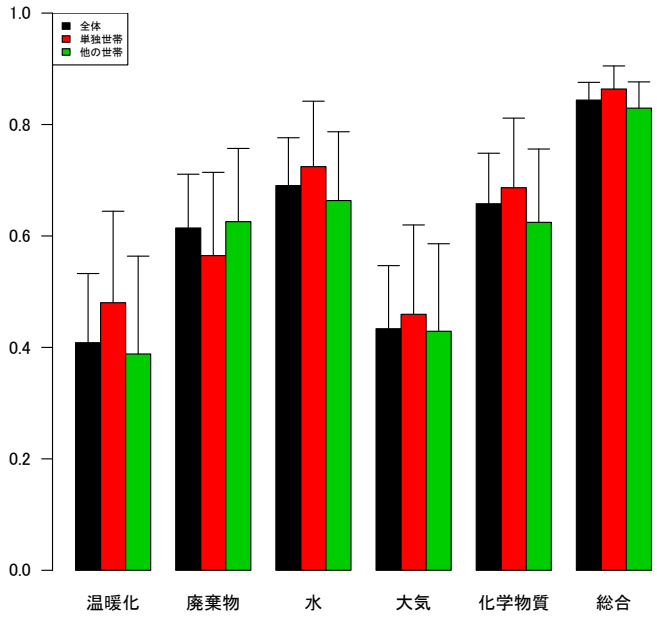


Figure 5.4: クロンバックの α 係数と 95%信頼区間の上限、世帯のタイプ別

components = 4” と表示されるので因子数は想定通り 5 で良いと考えられる（図 5.5）。

そこでクォーティマックス回転して因子分析をすると以下が得られる。

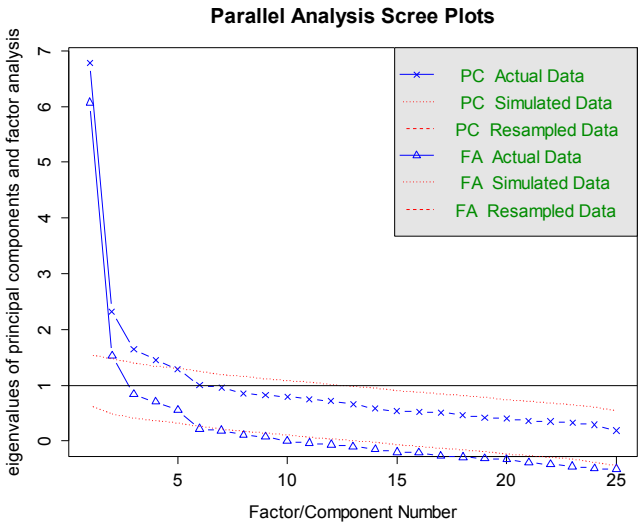


Figure 5.5: パラレル分析とスクリープロットによる因子数探索

```
     MR2    MR1    MR3    MR5    MR4
Q01  0.176  0.166                0.625
Q02  0.293  0.164  0.287         0.269
Q03  0.248  0.199  0.169         0.670
Q04  0.158  0.296  0.465  0.208  0.198
Q05  0.244  0.261  0.161  0.516  0.189
Q06  0.296  0.145         0.625  0.100
Q07  0.241  0.193         0.350
Q08                0.322         0.277
Q09  0.355                0.422 -0.200
Q10  0.365         0.155  0.375
Q11 -0.176  0.137  0.581
Q12  0.316  0.336  0.360  0.145
Q13  0.138  0.181  0.527         0.122
Q14  0.216  0.246  0.451  0.212  0.237
Q15  0.502  0.323  0.457
Q16  0.385  0.300  0.554 -0.108
Q17  0.462         0.202  0.130  0.323
Q18  0.687  0.156                0.137
Q19  0.462  0.106         0.320  0.296
Q20  0.632  0.269  0.130
Q21  0.650                0.164
Q22  0.202  0.789  0.174
Q23         0.930
Q24         0.428  0.236  0.191
Q25  0.234  0.142         0.315  0.278

                 MR2   MR1   MR3   MR5   MR4
SS loadings    3.102 2.560 2.129 1.535 1.516
Proportion Var 0.124 0.102 0.085 0.061 0.061
Cumulative Var 0.124 0.226 0.312 0.373 0.434
```

第 5 因子まで入れても分散の 43.4%しか説明できないし、どの因子ともあまり関係していない変数が多々ある。これは、回答者によって多義的な解釈が可能になってしまった変数であろうと思われる。この表から因子負荷量が 0.5 以上（この値は恣意的に決めた）のものだけ残して変数ごとの質問内容も付記すると、下表が得られる。

変数	MR2	MR1	MR3	MR4	MR5
Q01. 紙リサイクル				0.625	
Q03. 容器リサイクル				0.670	
Q05. 冷暖房控える					0.516
Q06. 食材適量購入					0.625
Q11. 太陽熱温水器			0.581		
Q13. 米のとぎ汁利用			0.527		
Q15. 塩ビラップ不買	0.502				
Q16. 石けん使う			0.554		
Q18. 除草殺虫剤不使用	0.687				
Q20. 強力洗浄剤不使用	0.632				
Q21. 有機溶剤不使用	0.650				
Q22. 有機農産物選好		0.789			
Q23. 地場農産物選好		0.930			

もしこれを因子分析結果として採用し、下位尺度の得点の計算に使うならば、これらの変数だけを使って因子分析をやり直す必要があるが、本稿ではそこまで深入りしない。

Chapter 6

構造方程式モデリング

構造方程式モデリングでは、複数の観測された変数同士、及び観測されていない潜在因子（構成概念）の関係性を、影響の向きも含めてモデル化し、それをデータに当てはめる。英語では Structural Equation Modeling と呼ばれ、略して sem という。

R で構造方程式モデリングをするためのパッケージはいろいろあるが、`sem` と `lavaan` がよく知られている。結果を図示するには `semPlot` パッケージの `semPaths()` 関数を使うのが便利である。変数間の関係性には無数の可能性があるので、それを直接コードとして打つのは慣れるまではハードルが高く、GUI（グラフィカルユーザーインターフェース）で関連図を描くと自動的に解析コードを生成してくれる補助ソフトが役に立つ[1]。

この補助ソフトとして有名なのが、Ωnyx である。Ωnyx も R と同じくフリーソフトウェアであり（開発プロジェクトはヴァージニア大学とマックスプランク研究所からサポートを受けている）、`https://onyx-sem.com/download/` からダウンロードすることができる（図 6.1）。2024 年 8 月 3 日現在、最新のファイルは安定版が 2019 年 3 月 28 日にリリースされた `onyx-1.0-1026.jar` で、開発版が 2021 年 3 月 9 日にリリースされた `onyx-1.0-1040.jar` である。

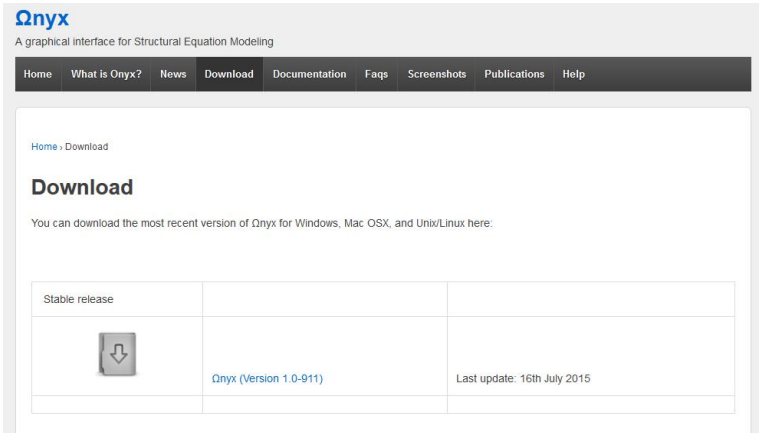


Figure 6.1: Ωnyx のダウンロード

[1]最初のコードを GUI での配置から生成し、後から手作業で直すという方法が効率が良いかもしれない。

Ωnyx は Java で書かれており、予め 1.6 以降の Java の実行環境をインストールしておかねば実行できない。Java の開発をするなら JDK をダウンロードすべきだが、コンパイル済みの Java のアプリを動かすだけなら JRE で十分である。Ωnyx のページからリンクされているのは、Oracle の JRE なのだが、一時期個人の非商用利用のみフリーであって、大学で研究に使うのも商用という判断がされたので、積極的にはお薦めしない。

フリーな Java 実行環境である AdoptOpenJDK を薦める。https://adoptopenjdk.net/ から自分の OS に合ったバージョンをダウンロードしてインストールする。Windows 11 であれば、2024 年 8 月 3 日現在、

```
OpenJDK21U-jre_x64_windows_hotspot_21.0.4_7.msi
```

（JDK の場合は OpenJDK21U-JDK_x64_windows_hotspot_21.0.4_7.msi）がダウンロードできるので、ダブルクリックしてインストールすれば良い。

そこまでできたら、Ωnyx は、ダウンロードした jar ファイル（onyx-1.0-1040.jar など）をダブルクリックするだけで動作する。詳しい使用法は pdf ファイル[2]としてダウンロードできる。右クリックメニューを多用する操作性は癖があるが、直感的に試行錯誤するだけでも、かなり使えるようになると思われる。

6.1 sem の基本

観測変数と構成概念の関係性を示すモデルをパス図として表現するには、５つのルールがある。

1. 観測変数は四角で囲む
2. 構成概念は楕円で囲む
3. 影響を与える変数から与えられる変数に向けた矢印を書く
4. 共変関係にある 2 つの外生変数（他の変数の結果となっていない変数）は双方向の矢印で結ぶ
5. 誤差変数は円で囲むか数値のみ書く

構造方程式モデリングのモデルはパス図で影響を受ける内生変数ごとに立てるのが基本である。モデルには測定方程式と構造方程式が含まれる。測定方程式とは、構成概念が複数の観測変数に影響を与える様子を記述するための方程式（構成概念が観測変数によってどのように測定されているかを表現する式ともいえるので、測定方程式と呼ぶ）である。観測変数 1 = 係数 1 ×構成概念 1 + 係数 2 ×構成概念 2 + ... という形になる。構造方程式とは、変数間の影響関係を表現するための方程式で、構成概念が他の構成概念に影響したり、観測変数が他の観測変数に影響したり、観測変数が構成概念に影響する場合がある。この他に観測変数のうち他の変数の影響で説明されない誤差（独自性）同士が関連していると考える「共変関係」が存在する。モデルではこれら３つの関係を表現することになる。

何の分析もなくいきなりパス図が描けるわけはないので、普通は散布図をみたり相関係数行列をみたり、探索的因子分析をしたり、先行研究を読んだりして、どのようなパス図がありそうかを推測しモデル化する。モデルを描くには、構造方程式モデ

[2] https://onyx.brandmaier.de/userguide.pdf

リングを実装したパッケージによって異なる文法がある。このテキストでは、lavaan パッケージと sem パッケージについて説明する。

6.2 エコポイントチェックデータへの適用例

前章末で検討したエコポイントチェックについての因子構造を元にして構造方程式モデルを考える。sem パッケージの場合は、探索的因子分析の結果通りに測定方程式を書くと適合度計算ができなかったので、質問項目の意味内容から若干測定方程式を追加して、以下のコードを実行した。

https://minato.sip21c.org/advanced-statistics/ecosem.R

```
URL <- "https://minato.sip21c.org/advanced-statistics/ecopx.txt"
eco <- read.delim(URL)
ecodata <- eco[, c(1, 3, 5, 6, 11, 13, 15, 16, 18, 20:23)+3]
C1 <- cor(ecodata)
library(sem)
M1 <- specifyEquations(text="
 Q22 = a1*HealthyLife
 Q23 = a2*HealthyLife
 Q18 = b1*AvoidChem
 Q20 = b2*AvoidChem
 Q21 = b3*AvoidChem
 Q15 = b4*AvoidChem
 Q11 = c1*Saver
 Q13 = c2*Saver
 Q16 = c3*Saver
 Q01 = d1*Recycle
 Q03 = d2*Recycle
 Q05 = e1*AvoidWaste
 Q06 = e2*AvoidWaste
 HealthyLife = 1*Ecopt
 AvoidChem = 1*Ecopt
 Saver = 1*Ecopt
 Recycle = 1*Ecopt
 AvoidWaste = 1*Ecopt
 V(Ecopt) = 1
")
S1 <- sem(M1, C1, N=length(ecodata[, 1]))
print(S1)
summary(S1, fit.indices=c("GFI","AGFI","CFI","RMSEA"))
library(semPlot)
LBL <- c("Q22","Q23","Q15","Q18","Q20","Q21","Q11","Q13","Q16",
         "Q01","Q03","Q05","Q06",
         "Healthy\n Life","Avoid\n Chemical","Saver","Recycle",
         "Avoid\n Waste","Ecopoint")
semPaths(S1, what="stand", layout="tree", style="lisrel",
         shapeMan="rectangle", shapeLat="ellipse",
         sizeMan=3, residScale=9, posCol="black",
         negCol="red", fade=FALSE, edge.label.cex=0.8,
         nodeLabels=LBL)
# For submission, negCol also shoule be "black"
```

これを実行すると図 6.2 が表示され、係数と適合度が以下のように推定される。

semPaths() 関数に与える作図のオプションはいろいろあるが、このコードで指定したのは以下のものである[3]。

what 矢印の上に何を表示するかを指定する。"stand"だと標準化したパラメータ推定値が表示される。標準化されていないパラメータ推定値を表示したい場合は"est"とする。デフォルトではパラメータ名が表示される。

layout 関連図の配置パタンを指定する。デフォルトは"tree"だが、"spring"とすると下図のような不規則な配置になる。円環状に配置したいときは"circle"にする。

style 誤差分散の表示スタイルを指定する。デフォルトでは枠付きの円状の両向き矢印だが、"lisrel"と指定すると、枠無しで変数に向かう矢印が表示される

shapeMan 観測変数の枠のスタイルで正方形か長方形か選べると書かれているが、"rectangle"と指定しても下図のように正方形になってしまった。

shapeLat 潜在因子の枠のスタイルで正円か楕円か選べると書かれているが、"ellipse"と指定しても下図のように正円になってしまった。

sizeMan 観測変数の枠サイズ

residScale 残差の表示サイズ（デフォルトは観測変数の枠サイズの 2 倍）

posCol パラメータ推定値が正な矢印の色。デフォルトは緑。

negCol パラメータ推定値が負な矢印の色。デフォルトは赤。

fade デフォルトでは TRUE になっていて、絶対値がゼロに近いパラメータや矢印ほど薄い色で表示される（透過性が高くなる）。すべての関連を同じ濃さで表示したいときは FALSE にする。

edge.label.cex パラメータの文字サイズを基準フォントサイズの何倍にするか。デフォルトは 0.8 倍。

nodeLabels 観測変数名と潜在因子名を文字列ベクトルとして与える。このオプションを指定しないと、モデルに与えた変数名が短縮されて表示される。

なお、AGFI が 0.9 に達していないし、RMSEA も大きいので、このモデルでは十分に因子構造を示しているとは言えない。本来はもっと高い AGFI が得られるまでモデルを探索すべきだが、今のところできていない[4]。

[3] 2019 年度講義中、実線と破線で描かれている矢印の意味の違いは調べて追記すると言ったが、まだ明確な説明が見つからない。すべて実線にしたいが、その方法も不明である。

[4] もっと良いモデルを sem パッケージで得ることに成功した方がいらしたら、是非お知らせいただきたい。

```
 Model Chisquare =  205.5724   Df =  60 Pr(>Chisq) = 7.976845e-18
 Goodness-of-fit index =  0.8969044
 Adjusted goodness-of-fit index =  0.8436383
 RMSEA index =  0.08889854   90% CI: (0.0757616, 0.102366)
 Bentler CFI =  0.880862
 Normalized Residuals
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-5.372000 -0.567000 -0.000001 -0.137700  0.536800  4.030000
 R-square for Endogenous Variables
HealthyLife         Q22         Q23   AvoidChem         Q18
     0.4201      0.8547      0.6972      0.7492      0.4430
        Q20         Q21         Q15       Saver         Q11
     0.5642      0.3155      0.4407      0.5626      0.0918
        Q13         Q16     Recycle         Q01         Q03
     0.2966      0.6205      0.3132      0.3486      0.7521
 AvoidWaste         Q05         Q06
     0.3509      0.7026      0.2977
```

```
 Parameter Estimates
                Estimate  Std Error  z value   Pr(>|z|)
a1              0.5992096 0.06007680  9.974060 1.979679e-23
a2              0.5411689 0.06140249  8.813469 1.213322e-18
b1              0.5760801 0.05832424  9.877198 5.227435e-23
b2              0.6501434 0.05884422 11.048553 2.227771e-28
b3              0.4861677 0.05737281  8.473835 2.374427e-17
b4              0.5746401 0.05830885  9.855110 6.514555e-23
c1              0.2272601 0.05324816  4.267942 1.972844e-05
c2              0.4084667 0.06124531  6.669354 2.569313e-11
c3              0.5908261 0.06013406  9.825149 8.774419e-23
d1              0.3304310 0.06406010  5.158141 2.494135e-07
d2              0.4853548 0.06162453  7.876000 3.380271e-15
e1              0.4965253 0.06148035  8.076162 6.683681e-16
e2              0.3231900 0.06422009  5.032537 4.840307e-07
V[HealthyLife] 1.3805665 0.36574314  3.774689 1.602075e-04
V[Q22]          0.1452525 0.05988679  2.425452 1.528935e-02
V[Q23]          0.3028184 0.05377719  5.630982 1.791865e-08
V[AvoidChem]    0.3347199 0.13876381  2.412156 1.585850e-02
V[Q18]          0.5570489 0.05555878 10.026299 1.168138e-23
V[Q20]          0.4358320 0.05116138  8.518768 1.612593e-17
V[Q21]          0.6845269 0.06201481 11.038118 2.502185e-28
V[Q15]          0.5592605 0.05565876 10.048023 9.372906e-24
V[Saver]        0.7774949 0.28505985  2.727480 6.382022e-03
V[Q11]          0.9081974 0.07610313 11.933771 7.891617e-33
V[Q13]          0.7034338 0.06866133 10.244978 1.246531e-24
V[Q16]          0.3795202 0.08214514  4.620117 3.835235e-06
V[Recycle]      2.1928126 0.75648576  2.898683 3.747332e-03
V[Q01]          0.6513939 0.07657334  8.506797 1.788036e-17
V[Q03]          0.2478714 0.12176585  2.035640 4.178652e-02
V[AvoidWaste]   1.8499837 0.66311145  2.789853 5.273192e-03
V[Q05]          0.2973725 0.12183788  2.440723 1.465789e-02
V[Q06]          0.7023140 0.07599017  9.242170 2.415488e-20
```

```
a1             Q22 <--- HealthyLife
a2             Q23 <--- HealthyLife
b1             Q18 <--- AvoidChem
b2             Q20 <--- AvoidChem
b3             Q21 <--- AvoidChem
b4             Q15 <--- AvoidChem
c1             Q11 <--- Saver
c2             Q13 <--- Saver
c3             Q16 <--- Saver
d1             Q01 <--- Recycle
d2             Q03 <--- Recycle
e1             Q05 <--- AvoidWaste
e2             Q06 <--- AvoidWaste
V[HealthyLife] HealthyLife <--> HealthyLife
V[Q22]         Q22 <--> Q22
V[Q23]         Q23 <--> Q23
V[AvoidChem]   AvoidChem <--> AvoidChem
V[Q18]         Q18 <--> Q18
V[Q20]         Q20 <--> Q20
V[Q21]         Q21 <--> Q21
V[Q15]         Q15 <--> Q15
V[Saver]       Saver <--> Saver
V[Q11]         Q11 <--> Q11
V[Q13]         Q13 <--> Q13
V[Q16]         Q16 <--> Q16
V[Recycle]     Recycle <--> Recycle
V[Q01]         Q01 <--> Q01
V[Q03]         Q03 <--> Q03
V[AvoidWaste]  AvoidWaste <--> AvoidWaste
V[Q05]         Q05 <--> Q05
V[Q06]         Q06 <--> Q06
```

なお、semPaths のオプションを layout="spring"に変えると、図 6.3 が表示される。どちらでなくてはいけないということはない。

6.2.1 lavaan でやってみる

lavaan パッケージと sem パッケージは文法が若干異なる。lavaan パッケージでは、測定方程式を表現するのに=~、構造方程式を表現するのに~、共変関係を表現するのに~~を用いる。また、lavaan パッケージは暗黙のうちに仮定する係数がたくさんあるので、モデル指定が短くて済む。

上の例を lavaan で書くと、

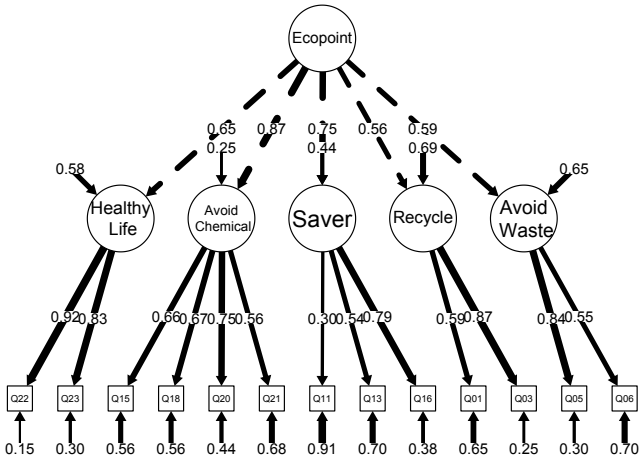


Figure 6.2: エコポイントデータへの構造方程式モデルの当てはめ (1)

https://minato.sip21c.org/advanced-statistics/ecolavaan.R

```
# 既にデータは読んであるので
# URL <- "https://minato.sip21c.org/advanced-statistics/ecopx.txt"
# eco <- read.delim(URL)
ecodata <- eco[, c(1, 3, 5, 6, 11, 13, 15, 16, 18, 20:23)+3]
ecodata <- subset(ecodata, complete.cases(ecodata))
library(lavaan)
M1 <- 'HealthyLife =~ Q22 + Q23
AvoidChem =~ Q18 + Q20 + Q21 + Q15
Saver =~ Q11 + Q13 + Q16
Recycle =~ Q01 + Q03
AvoidWaste =~ Q05 + Q06'
S1 <- sem(model=M1, data=ecodata, estimator="ML")
summary(object=S1, fit.measure=TRUE)
library(semTools)
reliability(S1)
library(semPlot)
LBL <- c("Q22","Q23","Q15","Q18","Q20","Q21","Q11","Q13","Q16",
         "Q01","Q03","Q05","Q06","Healthy\n Life","Avoid\n Chemical",
         "Saver","Recycle","Avoid\n Waste")
semPaths(S1, what="stand", layout="circle", style="lisrel",
       shapeMan="rectangle",
       shapeLat="ellipse",sizeMan=3, residScale=9, posCol="black",
       negCol="red", fade=FALSE, edge.label.cex=0.8,nodeLabels=LBL)
```

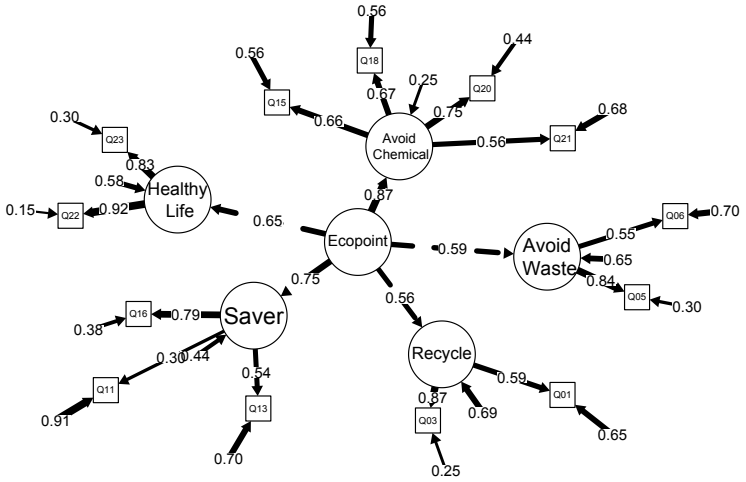
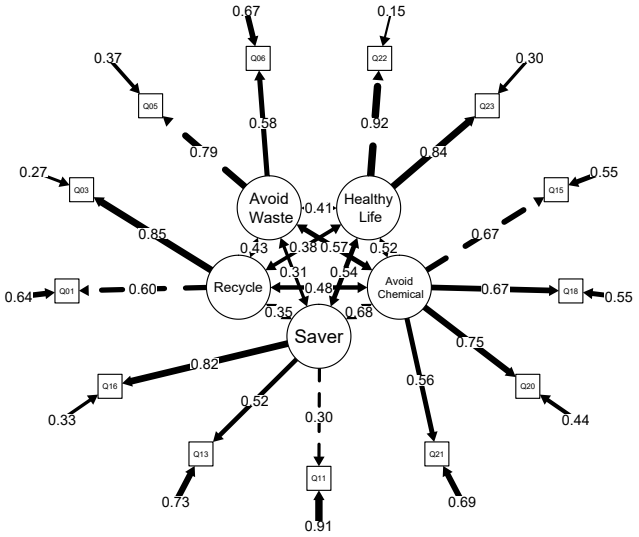


Figure 6.3: エコポイントデータへの構造方程式モデルの当てはめ (2)



以下の結果では、CFI がわずかに 0.9 に満たず、RMSEA も 0.090 で 0.05 より統計学的に有意に大きいので、適合は十分でないが、それほど悪くはないと言っていいかもしれない（これではまだ投稿はできないが）。

```
lavaan (0.5-20) converged normally after  73 iterations
  Number of observations                           308
  Estimator                                         ML
  Minimum Function Test Statistic              191.644
  Degrees of freedom                                55
  P-value (Chi-square)                           0.000

Model test baseline model:
  Minimum Function Test Statistic             1304.114
  Degrees of freedom                                78
  P-value                                        0.000

User model versus baseline model:
  Comparative Fit Index (CFI)                    0.889
  Tucker-Lewis Index (TLI)                       0.842

Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)              -5904.159
  Loglikelihood unrestricted model (H1)      -5808.337

  Number of free parameters                         36
  Akaike (AIC)                               11880.318
  Bayesian (BIC)                             12014.601
  Sample-size adjusted Bayesian (BIC)        11900.424

Root Mean Square Error of Approximation:
  RMSEA                                          0.090
  90 Percent Confidence Interval          0.076  0.104
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:
  SRMR                                           0.071

Parameter Estimates:
  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  HealthyLife =~
    Q22               1.000
    Q23               0.608    0.048   12.743    0.000
  AvoidChem =~
    Q15               1.000
    Q18               0.610    0.064    9.543    0.000
    Q20               0.600    0.058   10.304    0.000
    Q21               0.348    0.042    8.260    0.000
  Saver =~
    Q11               1.000
    Q13               0.777    0.186    4.187    0.000
    Q16               2.560    0.589    4.343    0.000
  Recycle =~
    Q01               1.000
    Q03               0.838    0.140    5.972    0.000
  AvoidWaste =~
    Q05               1.000
    Q06               0.654    0.107    6.124    0.000
```

1 ページに収まらないので、以下続き。

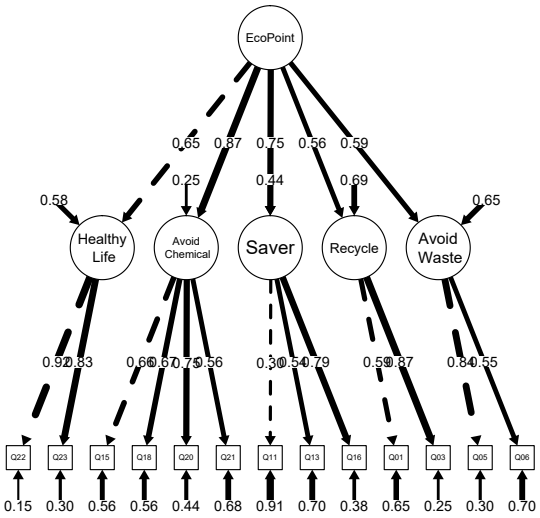
```
Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)
  HealthyLife ~~
    AvoidChem         0.751    0.120    6.255    0.000
    Saver             0.201    0.053    3.823    0.000
    Recycle           0.498    0.117    4.256    0.000
    AvoidWaste        0.294    0.057    5.203    0.000
  AvoidChem ~~
    Saver             0.472    0.123    3.853    0.000
    Recycle           1.167    0.262    4.455    0.000
    AvoidWaste        0.761    0.127    5.970    0.000
  Saver ~~
    Recycle           0.221    0.076    2.923    0.003
    AvoidWaste        0.107    0.037    2.879    0.004
  Recycle ~~
    AvoidWaste        0.515    0.124    4.163    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    Q22               0.133    0.053    2.520    0.012
    Q23               0.124    0.022    5.752    0.000
    Q15               3.345    0.333   10.049    0.000
    Q18               1.250    0.124   10.063    0.000
    Q20               0.774    0.089    8.717    0.000
    Q21               0.720    0.065   11.096    0.000
    Q11               1.870    0.155   12.026    0.000
    Q13               0.300    0.028   10.759    0.000
    Q16               0.576    0.148    3.886    0.000
    Q01               3.858    0.454    8.497    0.000
    Q03               0.571    0.237    2.407    0.016
    Q05               0.398    0.104    3.823    0.000
    Q06               0.568    0.062    9.103    0.000
    HealthyLife       0.775    0.089    8.718    0.000
    AvoidChem         2.702    0.450    5.997    0.000
    Saver             0.180    0.078    2.317    0.021
    Recycle           2.172    0.499    4.353    0.000
    AvoidWaste        0.667    0.127    5.243    0.000
```

また、semTools パッケージの reliability() 関数によって表示される、各潜在因子の内的一貫性を示すクロンバックの α 係数、他の因子をコントロールした上での条件付き信頼性を示す ω_1 係数（いわゆる Composite Reliability=CR を意味する）、条件なしの信頼性を示す ω_2 係数、階層的オメガとも呼ばれる ω_3 係数、収束的妥当性を示す AVE（Average Variance Extracted の略[5]）の値は以下の通りである。なお、モデルがデータによく当てはまっているときは、ω_2 と ω_3 は近い値になることが知られている。以下の結果からすると、HealthyLife のみ尺度として使えそうだが、他の潜在因子については値が十分でない。

[5] 平均分散抽出と訳されている文献がいくつかあるが、AVE のまま使う方が良さそうである。0.5 以上あれば収束的妥当性ありと判断される。

```
       HealthyLife AvoidChem     Saver   Recycle AvoidWaste     total
alpha    0.8338687 0.7006139 0.5190605 0.6188560  0.6249920 0.7751874
omega    0.8863942 0.7436756 0.5526472 0.6236869  0.6536104 0.8495406
omega2   0.8863942 0.7436756 0.5526472 0.6236869  0.6536104 0.8495406
omega3   0.8863945 0.7557620 0.5260580 0.6236864  0.6536105 0.8723029
avevar   0.8051577 0.4510670 0.3488440 0.4550706  0.4961635 0.4568644
```

https://minato.sip21c.org/advanced-statistics/ecolavaan2.R に、総合エコポイントから各因子への影響を考慮したモデルの当てはめを行うコードを載せたが、結果はほとんど変わらない。



6.3 John Fox 教授のテキストを参考に sem パッケージを使う

https://socserv.socsci.mcmaster.ca/jfox/Misc/sem/SEM-paper.pdf は、sem パッケージの開発者 John Fox 自身によるチュートリアルテキストである。本テキスト同様、簡単な R 入門から始まり、sem パッケージを使って、tsls() 関数により二段階最小二乗法 (2SLS) として操作変数（instrumental variables）のある回帰モデルを実行する方法を解説した後に、潜在的な外生変数[6]と内生変数[7]がある構造方程式モデル、polycor パッケージの hetcor() 関数を使って、観測変数がカテゴリ変数である場合の単純な構造方程式モデル推定、と解説が進む。sem パッケージと polycor パッ

[6]従属変数を説明するモデルの誤差と独立な独立変数と考えて良い。操作変数は外生変数であるべきである。

[7]誤差と相関している独立変数と考えて良い。

ケージをインストールされたい。操作変数法については後で触れることにして、本章では Fox 教授のチュートリアルテキストのうち、構造方程式モデルの部分だけ紹介する。

```
install.packages("sem", dep=TRUE)
install.packages("polycor", dep=TRUE)
```

6.3.1 典型的な構造方程式モデル

このコードは、**生データからではなく、分散共分散行列から分析**している。分散共分散行列は sem パッケージだけでなく lavaan パッケージのデータ入力にも使えるが、相関係数行列は lavaan パッケージの入力には想定されていないようである。このデータの元は、Wheaton *et al.* (1977)[8]である。John Fox 教授は、このデータを使った Exercise[9]も公開している。

Wheaton *el al.* (1977) の原文によれば、このデータは、イリノイ州の農村部での工業開発の影響を調べるため、Jones & Laughlin Steel Company が冷間圧延機を建設中の地域と、そうした開発が行われていない対象地域で 1966 年、1967 年、1971 年の 3 回にわたって行われた縦断的研究のものである[10]。2 地域を合わせて 932 人の調査結果が 3 時点で得られている。

Fox 教授がこの分析で使った変数は、1967 年と 1971 年のアノミー尺度得点（Anomia67 と Anomia71）[11]、無力感尺度得点（Powerless67 と Powerless71）[12]、そして教育歴（Education、完了した学校教育の年数）と社会経済指数（SEI、Duncan の SEI として知られている、センサスにおける職業のグループ別に調べられた職業的名声によるスコアを、その職業の平均的な社会経済地位を示す指数とした値）の 6 つである。

アノミー尺度得点と無力感尺度は、元々疎外感尺度の下位尺度であったと書かれているが、Fox 教授の解析では、疎外感尺度を数値としては使わず、調査年度別の潜在因子（Alienation67 と Alienation71）としてモデルに含めている。さらに、社会経済的状態を示す SES という潜在因子を Education と SEI の背後に仮定している。

なお、Wheaton *el al.* (1977) の原文及びそこで引用されている Srole (1956)[13]によると、アノミー尺度得点は、5 つの質問「役人に意見を書いたところでほとんど役立たない、なぜなら、彼らは一般人の問題には現実的には関心がないことが普通だから」「現代では、人はやや多く今日のために生きなくてはいけないので、明日のことは明日に任せねばならない」「何人かの人が何を言おうと、平均的な人の多くは日々悪くなっていく、より良くではなく」「将来を期待するようなやり方で子どもを世界に連れ出すことはほとんどフェアじゃない」「近頃、人は誰に頼ったらいいのかを本当には知らない」[14]に対して賛成-反対で得た回答の「賛成」の個数である。

---

[8] https://www.statmodel.com/bmuthen/psychometrics.htm から原文が pdf ファイルとして入手できる。

[9] https://statmath.wu.ac.at/courses/StatsWithR/Exercises-5.pdf

[10] 調査の詳細は Summers *et al.* (1969) 参照と書かれている。元の報告書をスキャンしたものが https://eric.ed.gov/?id=ED048953 から全文入手できる。

[11] 辞書によると anomia は健忘性失語症とあり、anomie がアノミー、没価値、無規範とあるが、これらの論文で扱われているのは明らかにデュルケームが定義したアノミーであって、健忘性失語は関係ない。

[12] Wheaton *et al.* (1977) に、無力感はこの調査で Summers が開発したものと書かれているのだが、それ以上の詳細は書かれていない。クロンバックの α 係数が 0.64 なので、尺度得点としての信頼性は不十分である。

[13] https://www.jstor.org/stable/2088422

[14] これら 5 つの質問文は、原文 “There’s little use of writing to public officials because often they aren’t really interested in the problems of the average man.”、“Nowadays a person has to live pretty much for today and let tomorrow take care of itself.”、“In spite of what some people say, the lot of the average man is getting worse, not better.”、“It’s hardly fair to bring children into the world with the way things look for the future.”、“These days a person doesn’t really know whom he can count on.” を筆者が日本語に意訳したものであり、逆翻訳などで

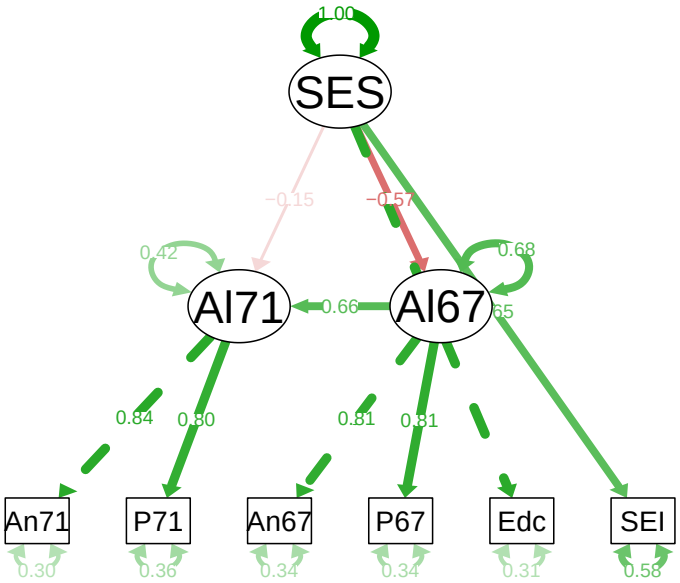
https://minato.sip21c.org/advanced-statistics/sem1.R

```
library(sem)
mod.wh.1 <- specifyModel(text="
Alienation67 -> Anomia67, NA, 1
Alienation67 -> Powerless67, lam1, NA
Alienation71 -> Anomia71, NA, 1
Alienation71 -> Powerless71, lam2, NA
SES -> Education, NA, 1
SES -> SEI, lam3, NA
Alienation67 -> Alienation71, beta, NA
SES -> Alienation67, gam1, NA
SES -> Alienation71, gam2, NA
SES <-> SES, phi, NA
Alienation67 <-> Alienation67, psi1, NA
Alienation71 <-> Alienation71, psi2, NA
Anomia67 <-> Anomia67, the11, NA
Powerless67 <-> Powerless67, the22, NA
Anomia71 <-> Anomia71, the33, NA
Powerless71 <-> Powerless71, the44, NA
Education <-> Education, thd1, NA
SEI <-> SEI, thd2, NA
")

S.wh <- matrix(c(
11.834,0,0,0,0,0,
6.947,9.364,0,0,0,0,
6.819,5.091,12.532,0,0,0,
4.783,5.028,7.495,9.986,0,0,
-3.839,-3.889,-3.841,-3.625,9.610,0,
-21.899,-18.831,-21.748,-18.775,35.522,450.288),
6,6,byrow=TRUE)

rownames(S.wh) <- colnames(S.wh) <-
 c('Anomia67','Powerless67','Anomia71','Powerless71','Education','SEI')

sem.wh.1 <- sem(mod.wh.1, S.wh, N=932)
summary(sem.wh.1, fit.indices=c("GFI","AGFI","CFI","RMSEA"))
library(semPlot)
semPaths(sem.wh.1, what="stand")
```

結果は以下の通り、AGFI が 0.91 と、それほど悪くない適合を示している。

妥当性を確認したわけではないので、このまま質問文として使える保証はないので注意されたい。

```
 Model Chisquare =  71.46973   Df =  6 Pr(>Chisq) = 2.041707e-13
 Goodness-of-fit index =  0.9751676
 Adjusted goodness-of-fit index =  0.9130866
 RMSEA index =  0.1082604   90% CI: (0.08658466, 0.1314454)
 Bentler CFI =  0.969066

 Normalized Residuals
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-1.2580000 -0.2118000 -0.0000127 -0.0153400  0.2444000  1.3310000

 R-square for Endogenous Variables
Alienation67     Anomia67  Powerless67 Alienation71     Anomia71
      0.3212       0.6607       0.6592       0.5763       0.7047
 Powerless71    Education          SEI
      0.6370       0.6936       0.4204

 Parameter Estimates
      Estimate    Std Error   z value    Pr(>|z|)
lam1    0.8885364  0.04150033  21.410348 1.070118e-101
lam2    0.8487223  0.03995708  21.240851 4.005674e-100
lam3    5.3289571  0.42976842  12.399601  2.626270e-35
beta    0.7047276  0.05353754  13.163242  1.428235e-39
gam1   -0.6138170  0.05645164 -10.873326  1.544659e-27
gam2   -0.1741787  0.05391357  -3.230702  1.234864e-03
phi     6.6658511  0.64105526  10.398247  2.525357e-25
psi1    5.3069765  0.47260170  11.229279  2.928570e-29
psi2    3.7412397  0.38756392   9.653220  4.763613e-22
the11   4.0155181  0.34315626  11.701719  1.248976e-31
the22   3.1913382  0.27145244  11.756528  6.536739e-32
the33   3.7010811  0.37341979   9.911315  3.717211e-23
the44   3.6248251  0.29208013  12.410379  2.295635e-35
thd1    2.9441577  0.49980006   5.890671  3.846307e-09
thd2  260.9929854 18.24177314  14.307435  1.966326e-46

lam1  Powerless67 <--- Alienation67
lam2  Powerless71 <--- Alienation71
lam3  SEI <--- SES
beta  Alienation71 <--- Alienation67
gam1  Alienation67 <--- SES
gam2  Alienation71 <--- SES
phi   SES <--> SES
psi1  Alienation67 <--> Alienation67
psi2  Alienation71 <--> Alienation71
the11 Anomia67 <--> Anomia67
the22 Powerless67 <--> Powerless67
the33 Anomia71 <--> Anomia71
the44 Powerless71 <--> Powerless71
thd1  Education <--> Education
thd2  SEI <--> SEI

 Iterations =  85
```



6.3.2 観測変数がカテゴリ変数である例

この例では、CNES というデータを使う。1997 年のカナダ国政選挙に関連して「伝統的価値観」への態度を調べるために行われた郵送式質問紙調査結果であり、1529 人分のデータが含まれている。変数の説明は次の通り。

MBSA2 an ordered factor with levels 'StronglyDisagree', 'Disagree', 'Agree', and 'StronglyAgree', in response to the statement, "We should be more tolerant of people who choose to live according to their own standards, even if they are very different from our own."

MBSA7 an ordered factor with levels 'StronglyDisagree', 'Disagree', 'Agree', and 'StronglyAgree', in response to the statement, "Newer lifestyles are contributing to the breakdown of our society."

MBSA8 an ordered factor with levels 'StronglyDisagree', 'Disagree', 'Agree', and 'StronglyAgree', in response to the statement, "The world is always changing and we should adapt our view of moral behaviour to these changes."

MBSA9 an ordered factor with levels 'StronglyDisagree', 'Disagree', 'Agree', and 'StronglyAgree', in response to the statement, "This country would have many fewer problems if there were more emphasis on traditional family values."

このデータを使ってカテゴリ変数間のポリコリック相関係数を計算させ（ただし hetcor() 関数に複数の変数を与えた場合、カテゴリ変数同士ではポリコリック相関係数、順序カテゴリと量的変数の間ではポリシリアル相関係数、量的変数同士の間ではピアソンの積率相関係数を自動的に計算してくれる）、

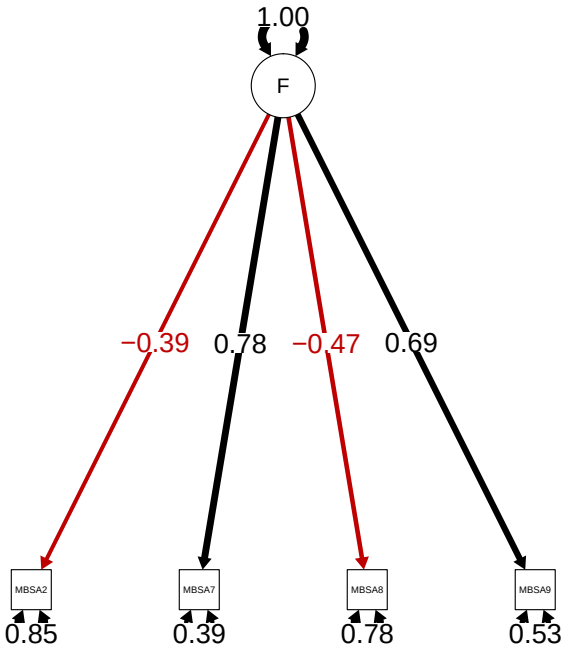
https://minato.sip21c.org/advanced-statistics/sem2.R

```
library(sem)
data(CNES)
library(polycor)
print(R.CNES <- hetcor(CNES, std.err=FALSE)$correlations)
model.CNES <- specifyModel(text="
F -> MBSA2, lam1, NA
F -> MBSA7, lam2, NA
F -> MBSA8, lam3, NA
F -> MBSA9, lam4, NA
F <-> F, NA, 1
MBSA2 <-> MBSA2, the1, NA
MBSA7 <-> MBSA7, the2, NA
MBSA8 <-> MBSA8, the3, NA
MBSA9 <-> MBSA9, the4, NA
")
sem.CNES <- sem(model.CNES, R.CNES, N=1529)
summary(sem.CNES, fit.indices=c("GFI","AGFI","CFI","RMSEA"))
library(semPlot)
semPaths(sem.CNES, what="stand", posCol="black", fade=FALSE)
```

hetcor() 関数で得られるポリコリック相関係数行列は以下の通りである。

```
           MBSA2      MBSA7      MBSA8      MBSA9
MBSA2  1.0000000 -0.3017953  0.2820608 -0.2230010
MBSA7 -0.3017953  1.0000000 -0.3422176  0.5449886
MBSA8  0.2820608 -0.3422176  1.0000000 -0.3206524
MBSA9 -0.2230010  0.5449886 -0.3206524  1.0000000
```

シンプルな構造方程式モデル（測定方程式しかないので、実は確証的因子分析に過ぎないが）を当てはめた結果は AGFI が 0.947 と十分に高く、良くデータを説明している。




```
 Model Chisquare =  33.2115   Df =  2 Pr(>Chisq) = 6.14066e-08
 Goodness-of-fit index =  0.9893351
 Adjusted goodness-of-fit index =  0.9466755
 RMSEA index =  0.1010603   90% CI: (0.07261014, 0.1326084)
 Bentler CFI =  0.9680971

 Normalized Residuals
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.000003  0.030010  0.207800  0.847900  1.035000  3.830000

 R-square for Endogenous Variables
 MBSA2  MBSA7  MBSA8  MBSA9
0.1516 0.6052 0.2197 0.4717

 Parameter Estimates
     Estimate   Std Error  z value   Pr(>|z|)
lam1 -0.3893289 0.02875484 -13.53959  9.129470e-42 MBSA2 <--- F
lam2  0.7779157 0.02996521  25.96063 1.379394e-148 MBSA7 <--- F
lam3 -0.4686834 0.02839946 -16.50325  3.476850e-61 MBSA8 <--- F
lam4  0.6867992 0.02921502  23.50842 3.344853e-122 MBSA9 <--- F
the1  0.8484230 0.03281417  25.85539 2.116323e-147 MBSA2 <--> MBSA2
the2  0.3948472 0.03567529  11.06781  1.797436e-28 MBSA7 <--> MBSA7
the3  0.7803360 0.03152466  24.75319 2.864281e-135 MBSA8 <--> MBSA8
the4  0.5283069 0.03212698  16.44434  9.208259e-61 MBSA9 <--> MBSA9

 Iterations =  14
```

Chapter 7

応用回帰分析とマルチレベル分析

7.1 多変量回帰分析

複数の従属変数を複数の独立変数で予測する回帰モデルの当てはめを多変量回帰分析と呼ぶ。lavaan や sem で構造方程式を記述すると独立変数の影響を除いた後の従属変数間の偏相関が計算できて便利である。

例として都道府県別の、1990 年の 100 世帯当たり車保有台数（CAR1990）、1989 年の人口 10 万当たり交通事故死者数（TA1989）、1985 年の国勢調査による人口集中地区居住割合 (DIDP1985)、都道府県名（PREF）、日本の東西（REGION）を含むデータを使う（REGION の 1 は東日本、2 は西日本を意味する）。

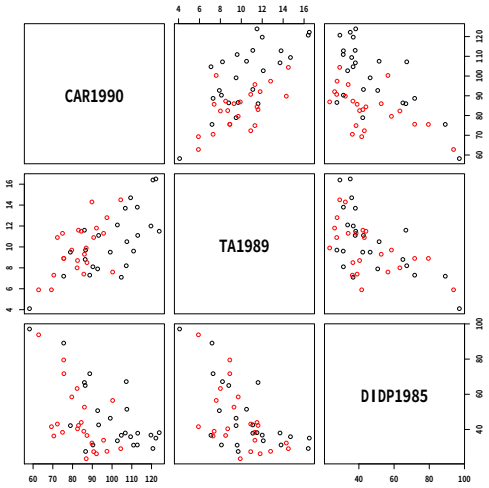
通常なら東日本と西日本で車台数で調整した交通事故死者数に差があるかを**共分散分析**によって検討するデータだが[1]、実は都市化の程度を示す人口集中地区居住割合は交通事故死者数と乗用車台数の両方に影響し、乗用車台数と交通事故死者数も関連があると考えられるので、多変量回帰分析を行うこともできる。

まずはデータを読み込んで相関係数行列を計算し、散布図行列を描いてみる。

https://minato.sip21c.org/advanced-statistics/carpopacc.R(1)

```
FN <- "https://minato.sip21c.org/advanced-statistics/carpopaccident.txt"
CPA <- read.delim(FN)
cor(CPA[, c("CAR1990", "TA1989", "DIDP1985")])
pairs(CPA[, c("CAR1990", "TA1989", "DIDP1985")], col=CPA$REGION)
```

[1] 詳細は「保健学共通特講 IV, VIII」のテキスト (https://minato.sip21c.org/ebhc/ebhc-text.pdf) を参照されたい。重回帰分析やロジスティック回帰分析、ポアソン回帰分析等についても同テキスト参照。そこまでの範囲なら EZR からメニュー操作で分析可能である。



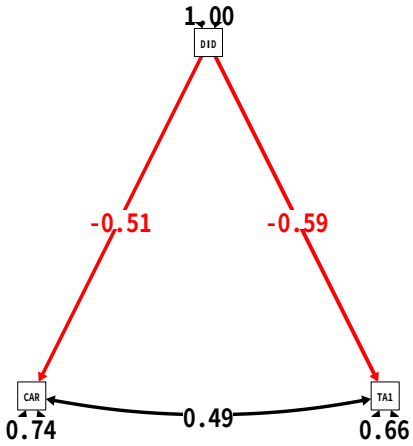
次に lavaan パッケージを使って、これら 3 つの変数間で多変量回帰分析を実行する。

https://minato.sip21c.org/advanced-statistics/carpopacc.R(2)

```
library(lavaan)
model1 <- 'CAR1990 ~ DIDP1985
TA1989 ~ DIDP1985'
res1 <- sem(model1, data=CPA[, c("CAR1990", "TA1989", "DIDP1985")])
summary(res1, standardized=TRUE, fit.measures=TRUE)
library(semPlot)
semPaths(res1, what="stand", posCol="black", negCol="red", fade=FALSE,
 edge.label.cex=2)
```

結果は以下の通り。グラフ表示を見れば偏相関関係が一目瞭然である。

```
lavaan (0.5-20) converged normally after  28 iterations
  Number of observations                            47
  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0

Model test baseline model:
  Minimum Function Test Statistic               47.096
  Degrees of freedom                                 3
  P-value                                        0.000

User model versus baseline model:
  Comparative Fit Index (CFI)                    1.000
  Tucker-Lewis Index (TLI)                       1.000

Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)               -489.488
  Loglikelihood unrestricted model (H1)       -489.488
  Number of free parameters                          5
  Akaike (AIC)                                988.975
  Bayesian (BIC)                              998.226
  Sample-size adjusted Bayesian (BIC)         982.544

Root Mean Square Error of Approximation:
  RMSEA                                          0.000
  90 Percent Confidence Interval          0.000  0.000
  P-value RMSEA <= 0.05                          1.000

Standardized Root Mean Square Residual:
  SRMR                                           0.000

Parameter Estimates:
  Information                                 Expected
  Standard Errors                             Standard

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)   Std.lv  Std.all
  CAR1990 ~
    DIDP1985         -0.441    0.108   -4.082    0.000   -0.441   -0.512
  TA1989 ~
    DIDP1985         -0.086    0.017   -4.947    0.000   -0.086   -0.585

Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)   Std.lv  Std.all
  CAR1990 ~~
    TA1989           14.668    4.832    3.035    0.002   14.668    0.494

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)   Std.lv  Std.all
    CAR1990         183.639   37.882    4.848    0.000  183.639    0.738
    TA1989            4.805    0.991    4.848    0.000    4.805    0.658
```



7.2 非線形回帰分析

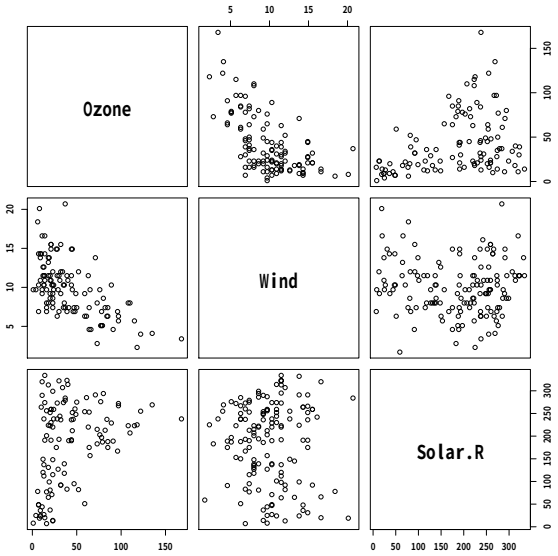
R では線形回帰は lm() 関数、従属変数（目的変数）が二項分布に従うロジスティック回帰、ポアソン分布に従うポアソン回帰などを含む一般化線型モデルは glm() 関数で簡単にデータへの当てはめを行うことができる。独立変数群が線形結合でない場合の回帰分析が、非線形回帰分析である。簡単な連続関数なら nls() 関数が使えるし、もっと一般に optim() 関数を使えば、どんな形の関数でもデータに当てはまるパラメータを推定できる。

R に元々含まれている、ニューヨークの大気環境データである airquality を使って、オゾン濃度を風速の指数関数と日照の線形結合で回帰するには次のコードを用いることができる。最後に predict() で、日照が平均値だったときに、風速が 0, 5, 10, 15, 20, 25 メートルだった場合の、この非線形回帰モデルで予測されるオゾン濃度が表示されるようにしてある。

https://minato.sip21c.org/advanced-statistics/nls.R

```
data(airquality)
# 通常の重回帰
pairs(airquality[, c("Ozone", "Wind", "Solar.R")])
resaq <- lm(Ozone ~ Solar.R + Wind, data=airquality)
summary(resaq)
AIC(resaq)
# Solar.R が平均値で Wind が 0, 5, 10, 15, 20, 25 の時の
# 重回帰モデルを使った Ozone 予測（強風だと負になってしまう）
predict(resaq, list(Wind=0:5*5, Solar.R=rep(mean(resaq$model$Solar.R), 6)))
# 非線形回帰には欠損値があるとエラーになるので除去したサブセットにする
AQ <- subset(airquality, !is.na(Ozone)&!is.na(Solar.R)&!is.na(Wind))
resmr <- nls(Ozone ~ a*exp(-b*Wind) + c*Solar.R,
             start=list(a=200, b=0.2, c=1), data=AQ)
summary(resmr)
AIC(resmr)
# Solar.R が平均値で Wind が 0, 5, 10, 15, 20, 25 の時の Ozone 予測
predict(resmr, list(Wind=0:5*5, Solar.R=rep(mean(AQ$Solar.R), 6)))
```

まず３つの変数間の同時散布図をみる。



通常の重回帰の結果は以下のようになり、自由度調整済重相関係数の二乗はそれほど小さくないが、強風だとオゾン濃度の予測値が負になってしまうので不適切である。

```
Call:
lm(formula = Ozone ~ Solar.R + Wind, data = airquality)

Residuals:
    Min      1Q  Median      3Q     Max
-45.651 -18.164  -5.959  18.514  85.237

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 77.24604    9.06751   8.519 1.05e-13 ***
Solar.R      0.10035    0.02628   3.819 0.000224 ***
Wind        -5.40180    0.67324  -8.024 1.34e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.92 on 108 degrees of freedom
  (42 observations deleted due to missingness)
Multiple R-squared:  0.4495,    Adjusted R-squared:  0.4393
F-statistic: 44.09 on 2 and 108 DF,  p-value: 1.003e-14

> AIC(resaq)
[1] 1033.816
> predict(resaq, list(Wind=0:5*5, Solar.R=rep(mean(resaq$model$Solar.R), 6)))
        1         2         3         4         5         6
 95.79102  68.78203  41.77304  14.76406 -12.24493 -39.25391
```

非線形回帰の結果は以下のように得られ、AIC が重回帰より小さくなり、予測値

も負にはならない。

```
Formula: Ozone ~ a * exp(-b * Wind) + c * Solar.R

Parameters:
   Estimate Std. Error t value Pr(>|t|)
a 215.42457   33.11390   6.506 2.49e-09 ***
b   0.24432    0.03331   7.335 4.32e-11 ***
c   0.08639    0.02014   4.290 3.90e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.01 on 108 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 9.951e-06

> AIC(resmr)
[1] 1006.24
> predict(resmr, list(Wind=0:5*5, Solar.R=rep(mean(AQ$Solar.R), 6)))
[1] 231.38999  79.46401  34.68228  21.48240  17.59160  16.44475
```

7.2.1 用量反応関係の解析

急性毒性試験でよく使われる用量反応関係も非線形回帰の一種なので、簡単に説明しておく。毒物を実験動物に投与した場合、用量（dose）や血中濃度に応じて標的臓器や個体の反応程度が変化するのだが、有害物の負荷量としての投与量（dose）に対する反応割合（＝反応した個体数／その dose を受けた総個体数）との関係を集団レベルでみると、S 字曲線になることが多い。原因は、反応（感受性）に個体差があることで、通常、累積対数正規分布で近似される。半数の個体が反応を示す負荷量を半数影響量 (ED50) と呼ぶ。急性毒性試験では半数致死量 (LD50) が良く使われ、推定にはプロビット分析[2]やロジット分析[3]が使われる。

用量反応関係のモデルとしては、ワンヒットモデル[4]や、線型多段階モデル[5]のよ

[2]プロビット分析では、

$$F^*(X_i) = \Phi(\beta_0 + \beta_1 X_i), \Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

というモデルを当てはめる。

[3]ロジット分析では、対数オッズが用量の線形関数となり、

$$F^*(X_i) = \Lambda(\beta_0 + \beta_1 X_i), \Lambda(z) = \frac{e^z}{1 + e^z}$$

というモデルを当てはめる。

[4]発がん物質が 1 回遺伝子に衝突し損傷を与えると、その細胞ががん化するというモデル。曝露量 D に対して細胞ががん化する確率 P(D) は、1 から 1 回も衝突しない確率を引いた値として得られ、

$$\mathrm{P(D)} = 1 - \mathrm{h(0)} = 1 - e^{-\mathrm{qD}}$$

とモデル化される。低用量域では発がんリスクが用量に比例するので、この比例定数 q を発がんスロープファクタと呼ぶ。

[5]1 つの細胞ががん化するために k 段階の反応が一定の順序で起こる必要があり、各段階の反応率が用量の一次式で表されると考えるモデル。

$$\mathrm{P(D)} = 1 - e^{-(q_0 + q_1 D + q_2 D^2 + \ldots + q_k D^k)}$$

と定式化される。米国 EPA が用いているのは、このモデルに $q_1 > 0$ という制約をつけた Crump のモデ

うなものもよく使われている。

R を使っていると、このように限定された目的には、専用のパッケージが存在することが多く、それを使う方が nls() 関数で頑張るより便利である。ここでは drc パッケージの使い方を示す。サンプルデータとして、doBy パッケージに入っている青虫（budworm）のデータを使ってみる（データの出典は、Venables and Ripley (1999) *Modern Applied Statistics with S-Plus.*, Springer である）。*trans*-cypermethrin という殺虫剤の用量（μg 単位）を何段階かで変えて投与したときの、雄と雌の青虫（タバコの葉を食べる蛾の幼虫）の反応を見たデータである。変数としては、sex、dose、ndead、ntotal が含まれている。つまり、1 行が 1 個体ではなく、dose ごとに雄雌それぞれ 1 行が与えられており、その処理の青虫の個体数が ntotal、そのうち死亡した個体数が ndead に与えられている。以下のように呼び出す。

https://minato.sip21c.org/advanecd-statistics/dr.R(1)

```
if (require(doBy)==FALSE) {
 install.packages("doBy"); library(doBy) }
data(budworm)
```

一般化線型モデルの関数 glm() を使ってロジスティック回帰分析し、それを dose.LD50() 関数に与えるという方法で分析できる（2007 年頃の doBy パッケージにはこの関数が含まれていた。ただし、現在 cran からインストールできる doBy パッケージでは dose.LD50() 関数が無いので、まずそれを定義する（下枠内はかつて存在したコードから作成したものである）。現在の doBy パッケージは多種多様な方法でのグループ統計量の処理をする方向に特化し、例えば、修正平均の計算[6]をする場合などに便利である。

ルである。通常は、P(0) によるリスクを除いた曝露量 D での発がんリスク、つまり過剰リスク

$$R(D) = \frac{P(D) - P(0)}{1 - P(0)}$$

を考える。このモデルでは D が 0 に近いとき R(D) は近似的に $q_1 D$ となるので、低用量域では過剰リスクが用量に比例する。

[6] https://mran.microsoft.com/web/packages/doBy/vignettes/LSmeans.pdf

https://minato.sip21c.org/advanecd-statistics/dr.R(2)

```
.ratioVar <- function(x, num, den, numval){
  m1 <- x
  beta <- coef(m1)
  numvec <- rep(0,length(beta))
  denvec <- rep(0,length(beta))
  numvec[num] <- numval
  denvec[den] <- 1
  M <- rbind(numvec, denvec)
  vcv <- summary(m1)$cov.scale
  beta2 <- M %*% beta
  vcv2  <- M %*% vcv %*% t(M)
  muvec <- c(1/beta2[2], -beta2[1]/(beta2[2]^2))
  ratiovar <- t(muvec) %*% vcv2 %*% muvec
  return(ratiovar)
}
.ratio <- function(x, num, den, numval, sign=-1){
  m1 <- x
  beta <- coef(m1)
  numvec <- rep(0, length(beta))
  denvec <- rep(0, length(beta))
  numvec[num] <- numval
  denvec[den] <- 1
  M <- rbind(numvec, denvec)
  beta2 <- M %*% beta
  ratio <- sign*beta2[1, 1]/beta2[2, 1]
  return(ratio)
}
.ld50 <- function(x, num, den, numval){
 est  <- .ratio(x, num, den, numval)
 vare <- .ratioVar(x, num, den, numval)
 result <- c("ld50"=est, lower=est-1.96*sqrt(vare), upper=est+1.96*sqrt(vare))
 return(result)
}
dose.LD50 <- function (x, lambda) {
 if (length(which(is.na(lambda))) != 1) {
  stop("lambda must contain exactly one entry which is NA") } else {
  den <- which(is.na(lambda))
  num <- which(!is.na(lambda))
  numval <- lambda[num]
  value <- .ld50(x, num, den, numval)
  return(value) }
}
```

このように定義した dose.LD50() 関数を使えば、glm() の結果から LD50 を計算することができる。

https://minato.sip21c.org/advanecd-statistics/dr.R(3)

```
mx <- glm(ndead/ntotal ~ sex + dose, weights=ntotal,
 data=budworm, family=binomial)
summary(mx)
dose.LD50(mx, c(1, 1, NA)) # for males
dose.LD50(mx, c(1, 0, NA)) # for females
```

結果は以下のように得られる。

```
> summary(mx)
Call:
glm(formula = ndead/ntotal ~ sex + dose, family = binomial,
    data = budworm, weights = ntotal)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5566  -1.3326   0.3384   1.1254   1.8838

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.13462    0.32029  -6.665 2.66e-11 ***
sexmale      0.96855    0.32954   2.939  0.00329 **
dose         0.15996    0.02341   6.832 8.39e-12 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 124.876  on 11  degrees of freedom
Residual deviance:  27.968  on  9  degrees of freedom
AIC: 64.078
> dose.LD50(mx, c(1, 1, NA)) # for males
ld50.numvec       lower       upper
   7.289943    4.620782    9.959105

> dose.LD50(mx, c(1, 0, NA)) # for females
ld50.numvec       lower       upper
   13.34505    10.16512    16.52498
```

5%水準で有意な性差があることもわかる。次に性差を無視して dose の対数で回帰してみる。結果としての LD50 を見るときに指数をとらなくてはいけないことに注意。

https://minato.sip21c.org/advanecd-statistics/dr.R(4)

```
m2 <- glm(ndead/ntotal ~ log(dose), weights=ntotal,
 data=budworm, family=binomial)
summary(m2)
exp(dose.LD50(m2, c(1, NA))) # same results as drm of drc
```

結果は以下。

```
> summary(m2)
Call:
glm(formula = ndead/ntotal ~ log(dose), family = binomial,
    data = budworm, weights = ntotal)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7989  -0.8267  -0.1871   0.8950   1.9850
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.7661     0.3701  -7.473 7.82e-14 ***
log(dose)     1.4525     0.1783   8.147 3.74e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 124.876  on 11  degrees of freedom
Residual deviance:  16.984  on 10  degrees of freedom
AIC: 51.094
Number of Fisher Scoring iterations: 4
> exp(dose.LD50(m2, c(1, NA))) # same results as drm of drc
ld50.numvec       lower       upper
   6.714995    5.362009    8.409377
```

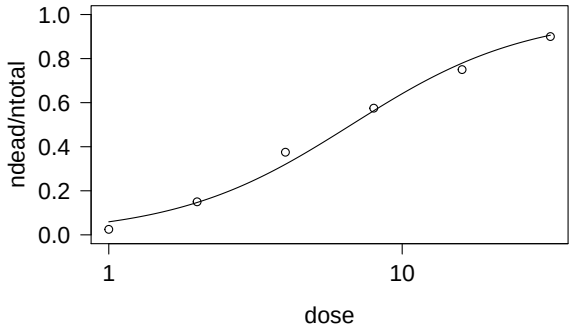
しかし、こんなに面倒なことをしなくても、drc パッケージを使うときわめて簡単である。

https://minato.sip21c.org/advanecd-statistics/dr.R(5)

```
if (require(drc)==FALSE) { install.packages("drc"); library(drc) }
m3 <- drm(ndead/ntotal ~ dose, weights=ntotal,
 data=budworm, fct=LL.2(), type="binomial") # LL.2 is log-logistic model
summary(m3)
ED(m3, 50, interval="delta")
plot(m3)
```

以下の結果が得られる。dose を対数変換して glm() に与えた結果を dose.LD50() に与えた結果とほぼ一致している。結果オブジェクトを plot() に渡すだけでグラフも描ける。

```
> summary(m3)
Model fitted: Log-logistic (ED50 as parameter) with
 lower limit at 0 and upper limit at 1 (2 parms)
Parameter estimates:
              Estimate Std. Error  t-value p-value
b:(Intercept) -1.45252    0.17830 -8.14645       0
e:(Intercept)  6.71483    0.77084  8.71101       0
> ED(m3, 50, interval="delta")
Estimated effective doses
(Delta method-based confidence interval(s))
     Estimate Std. Error   Lower  Upper
1:50  6.71483    0.77084 5.20400 8.2257
```



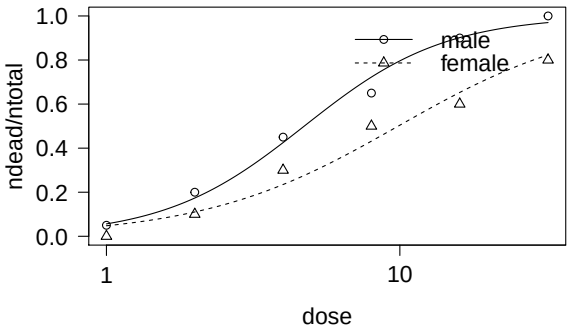
drc パッケージで性差を考慮するには、curveid=オプションで指定するだけである。

https://minato.sip21c.org/advanecd-statistics/dr.R(6)

```
m4 <- drm(ndead/ntotal ~ dose, curveid=sex, weights=ntotal,
          data=budworm, fct=LL.2(), type="binomial")
summary(m4)
ED(m4, 50, interval="delta")
plot(m4)
```

結果は以下の通り得られる。LD50 は雌が $9.9 \pm 1.8\mu g$、雄が $4.7 \pm 0.7\mu g$（± の後の値は標準誤差）と推定される。

```
> summary(m4)
Model fitted: Log-logistic (ED50 as parameter) with
 lower limit at 0 and upper limit at 1 (2 parms)
Parameter estimates:
         Estimate Std. Error  t-value p-value
b:male   -1.81592    0.30588 -5.93675       0
b:female -1.30705    0.24102 -5.42311       0
e:male    4.72170    0.66779  7.07060       0
e:female  9.87097    1.78005  5.54534       0
> ED(m4, 50, interval="delta")
Estimated effective doses
(Delta method-based confidence interval(s))
          Estimate Std. Error   Lower   Upper
female:50  9.87097    1.78005 6.38214 13.3598
male:50    4.72170    0.66779 3.41285  6.0306
```



次に、官庁のウェブサイトで多くの毒性試験データが公開されているので、https://www.wam.go.jp/wamappl/bb11gs20.nsf/0/f8e1fb7d07413544492571d1000bff64/$FILE/20060818siryou3-3_1_6.pdf からデータを取り出す[7]。ラットを使って行われた 2-mercaptobenzimidazole の急性毒性試験である。報告書ではプロビット分析またはBehrens-Karber 法によって、LD50 を雌で 208、雄で 218 と示されている。上の例と同じく対数ロジット法（drc パッケージで fct=LL.2() オプションを使う）で分析するコードを示す。なお、drc パッケージで fct オプションに与えることができる関数の一覧は、drc パッケージをロードした状態で、getMeanFunctions() と打てば得られる。応答変数のタイプも type オプションに"continuous"や"binomial"、"Poisson"などの文字列を与えることで指定可能である。

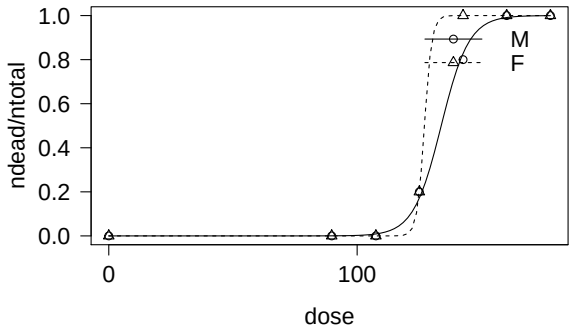
https://minato.sip21c.org/envhlth/dr2.R

```
if (require(drc)==FALSE) { install.packages("drc"); library(drc) }
rats <- data.frame(
 sex = factor(c(rep(1, 7), rep(2, 7)), labels=c("M", "F")),
 dose = rep(c(0, 79, 119, 178, 267, 400, 600), 2),
 ndead = c(0, 0, 0, 1, 4, 5, 5, 0, 0, 0, 1, 5, 5, 5),
 ntotal = rep(5, 14))
mx <- drm(ndead/ntotal ~ dose, curveid=sex, weights=ntotal,
 data=rats, fct=LL.2(), type="binomial")
summary(mx)
ED(mx, 50, interval="delta")
plot(mx)
```

結果は次のように得られる。

[7]元々は https://dra4.nihs.go.jp/mhlw_data/home/paper/paper583-39-1a.html にあった。一時リンクが切れていたが復活したようだ。

```
> summary(mx)
Model fitted: Log-logistic (ED50 as parameter) with
 lower limit at 0 and upper limit at 1 (2 parms)
Parameter estimates:
     Estimate Std. Error   t-value p-value
b:M  -7.77184    3.24901  -2.39206  0.0168
b:F -29.25825  198.69956  -0.14725  0.8829
e:M 218.01305   22.80926   9.55809  0.0000
e:F 186.63562   60.37334   3.09136  0.0020
> ED(mx, 50, interval="delta")
Estimated effective doses
(Delta method-based confidence interval(s))
     Estimate Std. Error   Lower  Upper
F:50  186.636     60.373  68.306 304.97
M:50  218.013     22.809 173.308 262.72
```



7.3 マルチレベル分析

本節は、大変わかりやすく、かつ本質的な解説書である、藤野ら (2013)[8]に概ね依拠している。若干高価な本だが、元東京大学の大橋靖雄教授とハーバード大学の河内一郎教授という疫学・生物統計学のがオビの推薦文を書いていることからも内容は信頼できる。

マルチレベル分析では、個体レベルで得られている従属変数を説明する独立変数群として、個体レベルの変数と集団レベル（メソレベルまたはマクロレベル）の変数を同時に考慮する。集団レベルの変数を、性別や薬剤のような水準をもった変数（その効果は固定効果 (fixed effect) と呼ばれる）と考えるのではなく、何らかの分布からの実現値と捉える。このとき、この集団レベルの変数は変量と呼ばれ、その効果は変量効果 (random effect) と呼ばれる。固定効果は水準の違いでリスクが何倍になるかといった効果の大きさが重要だが、変量効果の水準による影響は平均としてはゼロとなるので、変量効果で重要なのはバラツキの程度である。なお、集団レベルの変数が同じサブグループ内では他の変数の数値は互いに相関がある可能性が高い（級内相関 [=ICC] という値で示される）。

[8]藤野善久・近藤尚己・竹内文乃『保健医療従事者のためのマルチレベル分析』診断と治療社、ISBN:978-4-7878-2053-2、使われているデータは出版社のサポートページ（http://www.shindan.co.jp/download/index.php?pcode=205300）からダウンロードできる。

この説明から自明だが、実は反復測定データも、個人差を変量効果だと考えれば、マルチレベル分析で扱うことができる。なお、前掲書によると、マルチレベル分析の目的は大別して３つあり、階層構造を考慮した分析をすること、マクロレベル変数の影響を調べること、マクロレベル間の変動の有無を調べたり変動を説明すること（マクロレベルやメソレベルのバラツキがあるのか、あった場合に、それが個人の属性に起因するのかグループの特徴によるのかを調べること）とまとめることができる。

マルチレベル分析における必要サンプルサイズの計算には、ブリストル大学のMLPowSim[9]やオックスフォード大学の Tom Snijders 教授による PinT[10]など、専用のソフトが公開されている。ブリストル大学のサイトでは、マルチレベル分析用に開発され広く知られている MLwiN も販売（英国のアカデミックな目的のユーザのみ、サポートなしなら無料でダウンロードして使用可能だが）しているが、比較的高価であり、このテキストでは扱わない。通常は、マクロレベルの変数で層別して、層ごとに、普通の回帰分析などの計算式で必要サンプルサイズを計算すれば、倫理審査などでも問題になることはないようである。

7.3.1 マルチレベル分析の要点

マルチレベル分析は、数学的には混合効果モデルの枠組みで扱うことができる。繰り返しになるが、従属変数への効果を固定効果 (Fixed Effects) と変量効果 (Random Effects) の混合と捉えることで、個体レベルの変数の効果を固定効果として、集団レベルのばらつき（言い換えると、集団ごとに固定効果の傾きと切片がランダムに異なること）を変量効果として分析するわけである。

R の lme4 パッケージでは、線形混合効果モデルのみでなく、従属変数が正規分布に従わない一般化線形混合効果モデルや、非線形混合効果モデルも分析することが可能とされているが、ここでは線形混合効果モデルのみ説明する。

モデルの指定の方法は、

```
lmer(resp ~ FEexpr + (REexpr1 | factor1) + (REexpr2 | factor2) + ..., data=df)
```

のようにする。ここで、resp は従属変数（応答変数）、FEexpr は固定効果の変数、(REexpr1 | factor1) と (REexpr2 | factor2) は変量効果の変数項または共分散因子の構造を示す項である。変量効果の変数項の数はいくらでも多くとれるが、通常は少数にとどめる。右辺の書き方を下表にまとめる（g、g1、g2 は変量効果をみたいグループレベル変数、x は固定効果をみたい共変量、o は既知のオフセットである）。

式	意味
(1 | g)	g の各グループごとにランダム切片がある。これらの切片の平均と標準偏差が推定される。
0 + offset(o) + (1 | g)	g の各グループごとにランダム切片があり、ゼロでない変量効果の切片が既知のオフセット値 o。
(1 | g1/g2)	g1 内にネストされた g2 があるとき。
(1 | g1) + (1 | g2)	ネストしていない独立なグルーピングとして g1 と g2 があり、それらの変量効果をみたいとき。
x + (x | g)	x の固定効果があり、g のグループごとに x から応答変数への効果の切片と傾きの両方が変動するとき。
x + (x || g)	デフォルトでは同一の変量効果項内の全変数は相関していると仮定するが、こう書けば無相関な切片と傾きを仮定できる。

[9] https://www.bristol.ac.uk/cmm/software/mlpowsim/

[10] https://www.stats.ox.ac.uk/~snijders/multilevel.htm

なお、モデルの当てはめのとき、とくにオプションを指定しない限り、REML（制限付き最尤法）が用いられる。REML は、普通の最尤法では推定値にバイアスがかかるという問題への対処として提案された方法の１つである。REML では、固定効果を除去するデータの線形結合を考慮することによって変量効果を推定する。このため、固定効果だけが異なる２つのネストされたモデル（片方がもう片方を含んでいるようなモデル）を尤度比検定で評価したい場合は、REML は使えない。REML ではなく普通の最尤法 (ML) を使いたい場合は、オプションとして REML=FALSE をつければ良い。

7.3.2 例 1：多施設介入試験の分析

藤野ら (2013) の pp.69-70 に示されている多施設介入試験の分析を R の lmer() 関数で実行する方法を示す。lmer() 関数は lme4 パッケージに含まれているので、予め install.packages("lme4", dep=TRUE) でインストールしておく必要がある[11]。実行コードは以下の通り。

```
x <- read.csv("http://www.shindan.co.jp/download/205300/cholesterol.csv")
library(lme4)
res <- lmer(cholesterol ~ cholesterol_base + intervention + (1 | clinic),
 data=x, REML=FALSE)
summary(res)
confint(res)
```

Stata の出力である図 9-5 の数値と比べると、Wald 検定とのその p 値、固定効果（介入とベースラインのコレステロール値と切片）の係数の z と p 値を除けば求めることができている。固定効果の係数については、z の代わりに t value が提示されている。**p 値は敢えて表示していない**とのことである。自由度が簡単には求められないからというのが大きな理由である。

それでも p 値が欲しい場合は、https://mindingthebrain.blogspot.jp/2014/02/three-ways-to-get-parameter-specific-p.html に解説されているように、自由度が十分に大きい t 分布は正規分布とほぼ同じだからと考えれば、

```
1-pnorm(summary(res)$coefficients[,"t value"])
```

で p 値を計算してしまうこともできる。また、doBy パッケージの LSmeans() 関数を使い、adjust.df=FALSE オプションを指定すると、自由度がサンプルサイズより 1 つ小さい t 分布を使って p 値を計算することができるようである（注：未確認である）。

SAS や Stata で得られるのと同等な p 値を計算するためには、

```
install.packages("lmerTest", dep=TRUE)
```

によって lmerTest パッケージをインストールしておけば、

[11] 後述するように、筆者は lmerTest パッケージを使う方が良いと思うが、思想の問題なので何とも言えない。

```
x <- read.csv("http://www.shindan.co.jp/download/205300/cholesterol.csv")
library(lmerTest)
res <- lmer(cholesterol ~ cholesterol_base + intervention + (1 | clinic),
 data=x, REML=FALSE)
summary(res)
confint(res)
```

のように、まったく同じ関数指定でも p 値が得られる[12]。あるいは、pbkrtest パッケージを使えば Kenward-Roger の近似で自由度を出すことができるので、t 分布の累積確率密度関数を使って p 値を出すこともできる。

実は既出の doBy パッケージを使うと、lme4 パッケージの lmer() 関数の結果を LSmeans() 関数に渡すだけで（ただし effect=オプションを適切に指定する必要がある）、Kenward-Roger の近似自由度を使った p 値を計算してくれるようである。

なお、R でマルチレベル分析をするには、multilevel パッケージと nlme パッケージを使う方法もあるようだが、ここでは深入りしない（https://cran.r-project.org/doc/contrib/Bliese_Multilevel.pdf で解説されている）。

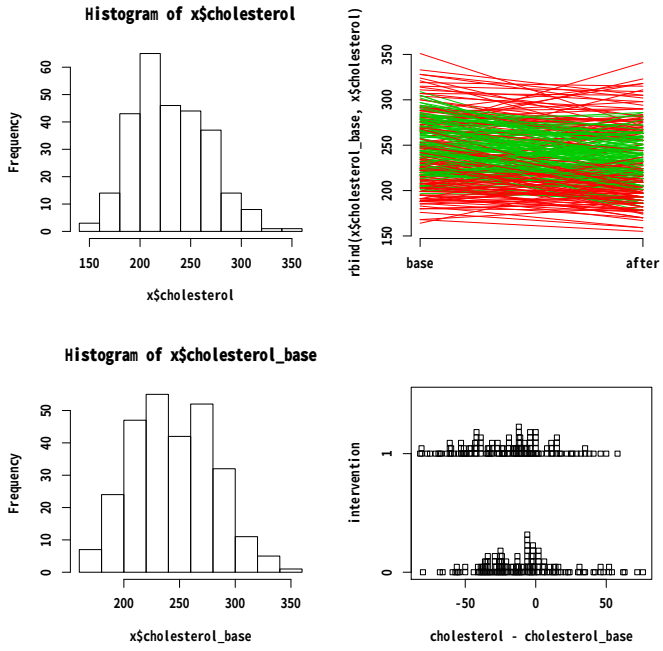
以下、この例で、データの性状を見るための作図からマルチレベル分析まで、一通りの操作をするコードをまとめて示す。

https://minato.sip21c.org/advanced-statistics/multilev.R(1)

```
x <- read.csv("http://www.shindan.co.jp/download/205300/cholesterol.csv")
# graph1
layout(matrix(1:4, 2))
hist(x$cholesterol)
hist(x$cholesterol_base)
matplot(rbind(x$cholesterol_base, x$cholesterol), type="l",
 col=x$intervention+2, lty=1, lwd=1, axes=FALSE)
axis(1, 1:2, c("base","after"))
axis(2, 3:7*50)
stripchart(cholesterol-cholesterol_base ~ intervention, data=x,
 method="stack", ylab="intervention")
```

データの性状を見るためのグラフは次のように描かれる。

[12] SAS の PROC MIXED 同様、Satterthwaite の近似で自由度と p 値を計算してくれると書かれている。



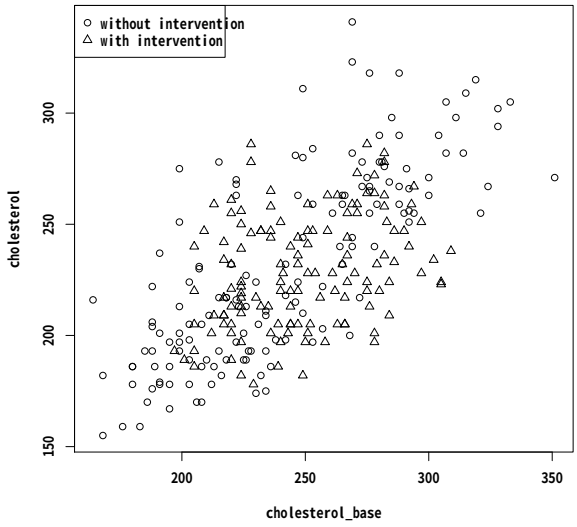
分析は、まず介入の有無も施設の違いも無視して、前後でのコレステロール値の差があるかどうかだけ、対応のある t 検定で調べてみると、

https://minato.sip21c.org/advanced-statistics/multilev.R(2)

```
# t-test
t.test(x$cholesterol_base, x$cholesterol,
 paired=TRUE, var.equal=FALSE)
# graph2
layout(1)
plot(cholesterol ~ cholesterol_base, pch=intervention+1, data=x)
legend("topleft", pch=1:2,
 legend=c("without intervention","with intervention"))
```

```
Paired t-test

data:  x$cholesterol_base and x$cholesterol
t = 8.0383, df = 275, p-value = 2.713e-14
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 10.96801 18.08272
sample estimates:
mean of the differences
               14.52536
```

と、有意水準 5%で統計学的に有意な差があるといえる。



次に、藤野ら (2013)p.69 図 9-4 に示されている Stata による回帰分析の結果をと比較するため、共分散分析で、介入の有無とベースラインのコレステロール値の、治療後のコレステロール値への交互作用効果を調べてみる。

https://minato.sip21c.org/advanced-statistics/multilev.R(3)

```
# ANCOVA
x$interventionF <- as.factor(x$intervention)
res1 <- lm(cholesterol ~ cholesterol_base * interventionF, data=x)
summary(res1)
```

```
Call:
lm(formula = cholesterol ~ cholesterol_base * interventionF,
    data = x)

Residuals:
    Min      1Q  Median      3Q     Max
-52.619 -18.290  -2.475  16.547  87.592

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     41.05775   12.48988   3.287  0.00114 **
cholesterol_base                 0.78941    0.05094  15.496  < 2e-16 ***
interventionF1                 111.80484   24.42985   4.577 7.19e-06 ***
cholesterol_base:interventionF1  -0.48289    0.09812  -4.922 1.49e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 25.97 on 272 degrees of freedom
Multiple R-squared:  0.483,Adjusted R-squared:  0.4773
F-statistic: 84.72 on 3 and 272 DF,  p-value: < 2.2e-16
```

と、交互作用効果（cholesterol_base:interventionF1 の行）も有意水準 5% で統計学的に有意であった。つまり、介入した群としなかった群の間で、ベースラインのコレステロール値と治療後のコレステロール値の関係が有意に異なっていた。もし解析の目的がベースラインコレステロール値と治療後コレステロール値の関係を調べることであれば、共分散分析によって修正平均を比較するよりも、2 群別々に分析すべきということになるのだが、ここでの分析の目的は介入効果の評価なので、その方法では不適切である。

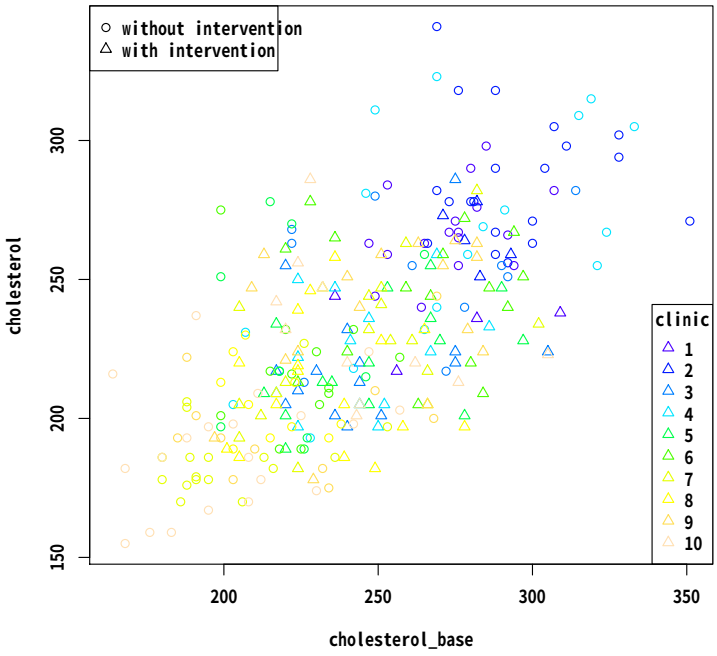
そこで、前掲書図 9-4 と同じ結果を得るには交互作用効果を入れずに lm() 関数で線形回帰すれば良い。

https://minato.sip21c.org/advanced-statistics/multilev.R(4)

```
res2 <- lm(cholesterol ~ cholesterol_base + interventionF, data=x)
summary(res2)
# graph3
plot(cholesterol ~ cholesterol_base, pch=intervention+1, data=x,
 col=topo.colors(10)[clinic])
legend("topleft", pch=1:2,
 legend=c("without intervention","with intervention"))
legend("bottomright", pch=2, col=topo.colors(10),
 legend=1:10, title="clinic")
```

```
Call:
lm(formula = cholesterol ~ cholesterol_base + interventionF,
    data = x)

Residuals:
    Min      1Q  Median      3Q     Max
-51.755 -20.040  -2.563  16.194  91.172

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      72.49400   11.18021   6.484 4.16e-10 ***
cholesterol_base  0.65923    0.04535  14.536  < 2e-16 ***
interventionF1   -7.42741    3.27763  -2.266   0.0242 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.05 on 273 degrees of freedom
Multiple R-squared:  0.437,Adjusted R-squared:  0.4329
F-statistic:   106 on 2 and 273 DF,  p-value: < 2.2e-16
```

のようになり、前掲書図 9-4 と同じ結果が得られる。このとき、ベースラインのコレステロール濃度の影響を調整した上で、有意な介入効果が得られていると考えられる。



しかし施設ごとに色を変えてプロットし直してみると、施設の効果がありそうに思えるので、これはマルチレベル分析にすべきである。lmerTest パッケージの lmer() 関数により得られた結果オブジェクトを summary() に渡せば、

https://minato.sip21c.org/advanced-statistics/multilev.R(5)

```
# multilevel
library(lmerTest)
res3 <- lmer(cholesterol ~ cholesterol_base + intervention +
 (1 | clinic), data=x, REML=FALSE)
summary(res3)
confint(res3)
```

により、以下の結果が得られる。

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
  approximations to degrees of freedom [lmerMod]
Formula: cholesterol ~ cholesterol_base + intervention + (1 | clinic)
   Data: x

     AIC      BIC   logLik deviance df.resid
  2594.9   2613.0  -1292.4   2584.9      271

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.8741 -0.7486 -0.1038  0.5595  3.0095

Random effects:
 Groups   Name        Variance Std.Dev.
 clinic   (Intercept) 141.7    11.90
 Residual             637.0    25.24
Number of obs: 276, groups:  clinic, 10

Fixed effects:
                 Estimate Std. Error        df t value Pr(>|t|)
(Intercept)      116.72601   13.27558 163.47000   8.793    2e-15 ***
cholesterol_base   0.47132    0.05239 247.26000   8.997   <2e-16 ***
intervention      -1.74572    3.26249 275.75000  -0.535    0.593
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) chlst_
cholstrl_bs -0.945
interventin  0.120 -0.241
```

係数の信頼区間は、同じ結果オブジェクトを confint() 関数に渡せば得られる。lme4 の開発者による解説文書には、Wald の近似を使い、ブートストラップ法で推定した値が表示されると書かれている（だから lme4 パッケージでは p 値を表示しないのである）。以上の手順で、Stata で得られる結果を、ほぼすべて R でも得ることができる。マルチレベル分析にすると、回帰分析では有意であった intervention の効果が有意でない、というのがポイントである。この場合、介入効果があったように見えていたのは、施設間差による artefact であったと考えられる。

```
Computing profile confidence intervals ...
                      2.5 %      97.5 %
.sig01            6.4527670  21.7732086
.sigma           23.2278626  27.5583053
(Intercept)      87.9180189 145.5631716
cholesterol_base  0.3581452   0.5886708
intervention     -8.4035413   4.8428531
```

7.3.3 例 2：動物実験

藤野 (2013)pp.73-74 に書かれている、遺伝子導入した豚のデータを R で分析してみる。このデータはマクロレベル変数の影響を調べる目的で提示されている。A, B の異なる遺伝子のどちらかを導入した豚と、遺伝子導入していないコントロール豚の 3

群（各群 5 頭）を対象に、心臓に高頻度ペーシング[13]を 7 日間行い、心房細動が誘発される頻度が遺伝子導入と関連があるかを調べたというデザインである。毎日 2 分間心電図を記録し、出現した波形のうち、心房細動（P 波がないパタン）が出現する割合を調べている。

変数は、n_obs が出現した波形の数、n_af が心房細動の出現数、group が遺伝子導入状態（Control と遺伝子 A を導入した A と遺伝子 B を導入した B の 3 水準）、id は豚の個体 ID、day が実験開始から何日目かを示す整数である。

https://minato.sip21c.org/advanced-statistics/multilev2.R(1)

```
# ダウンロードデータの説明
# 診断と治療社『保健医療従事者のためのマルチレベル分析活用ナビ』より
# http://www.shindan.co.jp/download/index.php?pcode=205300
db <- read.csv("http://www.shindan.co.jp/download/205300/doubutsu.csv")
db$group <- factor(db$group, labels=c("Control", "A", "B"))
res <- glm(n_af ~ group + day, data=db, family="poisson")
summary(res)
exp(coef(res))
exp(confint(res))
# マルチレベル
library(lmerTest)
db$afprop <- db$n_af/db$n_obs
res1 <- lmer(afprop ~ (1 | id), data=db, REML=FALSE)
summary(res1)
res2 <- lmer(afprop ~ day + group + (1 | id), data=db, REML=FALSE)
summary(res2)
res3 <- lmer(afprop ~ group + (day | id), data=db, REML=FALSE)
summary(res3)
anova(res1, res2, res3)
```

結果は以下。まず通常の一般化線型モデルでポアソン回帰をする[14]。

[13]電気刺激によって心臓の動く回数を増加させる方法のこと。
[14]ポアソン回帰についての説明は、保健学共通特講 IV, VIII のテキストを参照されたい。

```
Call:
glm(formula = n_af ~ group + day, family = "poisson", data = db)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.1690  -2.7101  -0.6852   1.4842   8.1568

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.96098    0.10107   9.508  < 2e-16 ***
groupA       0.69869    0.09120   7.661 1.85e-14 ***
groupB       1.31730    0.08393  15.696  < 2e-16 ***
day          0.15696    0.01479  10.612  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1267.0  on 104  degrees of freedom
Residual deviance:  841.6  on 101  degrees of freedom
AIC: 1200.7

Number of Fisher Scoring iterations: 6

(Intercept)      groupA      groupB         day
   2.614253    2.011111    3.733333    1.169952

               2.5 %   97.5 %
(Intercept) 2.138157 3.177968
groupA      1.684723 2.409353
groupB      3.175024 4.412791
day         1.136637 1.204501
```

係数が微妙に違うが、概ね藤野 (2013) の図 10-3 と同じ結果が得られている。心房細動の発生頻度が、遺伝子 A の導入により 2.01 倍（95%信頼区間 1.68-2.41）、遺伝子 B の導入により 3.73 倍（95%信頼区間 3.17-4.42）になったことを意味する。しかし、同一個体内での心房細動発生頻度には強い相関があると考えられるので、個体の変量効果を考慮するためにマルチレベル分析をする。

lme4 パッケージは前述のように一般化線形混合効果モデルを扱えるため、ポアソン回帰のままの分析も可能なはずだが、線形混合効果モデルで分析するには反応変数を割り算して出現率にする必要がある[15]。個体だけではなく繰り返し測定についても変量効果を考え、かつモデル間で尤度比検定をしてみた結果が以下である。

[15] あまりお薦めでない方法なので、後日時間があれば、ここは一般化線形混合効果モデルで書き直す予定である。

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
  approximations to degrees of freedom [lmerMod]
Formula: afprop ~ group + (1 | id)
   Data: db

     AIC      BIC   logLik deviance df.resid
    61.1     74.4    -25.5     51.1      100

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.1293 -0.6137 -0.2822  0.8290  2.6477

Random effects:
 Groups   Name        Variance Std.Dev.
 id       (Intercept) 0.001278 0.03575
 Residual             0.094014 0.30662
Number of obs: 105, groups:  id, 15

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)
(Intercept)  0.17148    0.05424 15.00000   3.162  0.00645 **
groupA       0.14824    0.07670 15.00000   1.933  0.07239 .
groupB       0.47389    0.07670 15.00000   6.178 1.77e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
       (Intr) groupA
groupA -0.707
groupB -0.707  0.500
```

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
  approximations to degrees of freedom [lmerMod]
Formula: afprop ~ day + group + (1 | id)
   Data: db

     AIC      BIC   logLik deviance df.resid
    44.7     60.6    -16.3     32.7       99

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.9703 -0.7082 -0.1541  0.6803  3.0359

Random effects:
 Groups   Name        Variance Std.Dev.
 id       (Intercept) 0.003761 0.06133
 Residual             0.076632 0.27682
Number of obs: 105, groups:  id, 15

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)
(Intercept) -0.07264    0.07656 51.15000  -0.949   0.3471
day          0.06103    0.01351 90.00000   4.518 1.89e-05 ***
groupA       0.14824    0.07670 15.00000   1.933   0.0724 .
groupB       0.47389    0.07670 15.00000   6.178 1.77e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
       (Intr) day    groupA
day    -0.706
groupA -0.501  0.000
groupB -0.501  0.000  0.500
```

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
  approximations to degrees of freedom [lmerMod]
Formula: afprop ~ group + (day | id)
   Data: db

     AIC      BIC   logLik deviance df.resid
    31.2     49.7     -8.6     17.2       98

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.11685 -0.61717 -0.07123  0.44926  2.60950

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 id       (Intercept) 0.18725  0.4327
          day         0.01018  0.1009   -0.98
 Residual             0.04651  0.2157
Number of obs: 105, groups:  id, 15

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)
(Intercept)  0.16353    0.05414 15.00000   3.021   0.0086 **
groupA       0.17168    0.07656 15.00000   2.242   0.0405 *
groupB       0.50530    0.07656 15.00000   6.600 8.44e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
       (Intr) groupA
groupA -0.707
groupB -0.707  0.500
```

```
Data: db
Models:
object: afprop ~ group + (1 | id)
..1: afprop ~ day + group + (1 | id)
..2: afprop ~ group + (day | id)
       Df    AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
object  5 61.088 74.358 -25.5439   51.088
..1     6 44.689 60.613 -16.3444   32.689 18.399      1  1.792e-05 ***
..2     7 31.151 49.729  -8.5756   17.151 15.538      1  8.088e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

尤度比検定の結果、グループの固定効果と個体の変量効果のみ考えた最初のモデルに比べて、日の固定効果も含めた２番目のモデルや、日の効果に個体差を考慮した最後のランダム切片モデル（Cor が-0.98 と個体差が日数と強い負の相関があることもわかる）は、どちらも 5%水準で（p 値はどちらも 10^{-5} のオーダーだから 5%どころではないが）有意に当てはまりが良く、AIC の値を見ると最後のモデルが最も小さい値を示しているので、最も当てはまりが良いと考えられる。

7.3.4 例 3：職域サポート

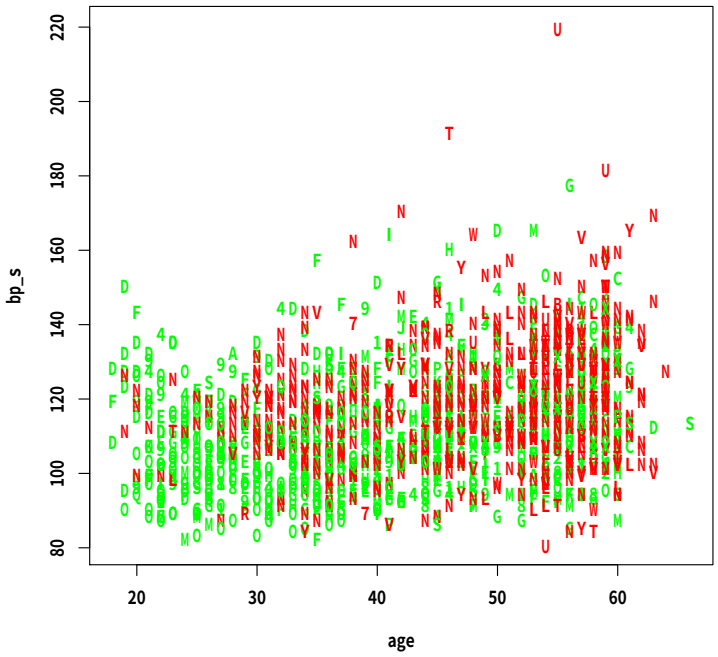
藤野 (2013)pp.75-76 に掲載されている、企業における血圧と職場サポートの関連の検討である。職場サポートは質問票により部署ごとに計算された平均点をその職場のサポートスコアとした。データに含まれている変数は、workplace が部署、bp_s が収縮期血圧、age が年齢、support がサポートスコアである。

https://minato.sip21c.org/advanced-statistics/multilev2.R(2)

```
wps <- read.csv("http://www.shindan.co.jp/download/205300/shokuiki2.csv")
wps$workplace <- as.factor(wps$workplace)
res1 <- lm(bp_s ~ age + support, data=wps)
summary(res1)
PCHS <- c(1:9, LETTERS)
plot(bp_s ~ age, pch=PCHS[as.integer(workplace)], data=wps,
 col=ifelse(support<median(support), "red", "green"))
library(lmerTest)
res4 <- lmer(bp_s ~ age + support + (1 | support), data=wps, REML=FALSE)
summary(res4)
confint(res4)
res5 <- lmer(bp_s ~ age + (1 | support), data=wps, REML=FALSE)
summary(res5)
confint(res5)
anova(res4, res5)

print(devdif <- as.numeric(-2*(logLik(res1)-logLik(res4))))
print(dfdif <- attr(logLik(res4), "df")-attr(logLik(res1), "df"))
pchisq(devdif, dfdif, lower.tail=FALSE)
```

まずはサポートの有無と年齢の血圧への効果をみる線形回帰モデルの結果を示す。

```
Call:
lm(formula = bp_s ~ age + support, data = wps)

Residuals:
    Min      1Q  Median      3Q     Max
-40.603 -10.943  -1.209   9.676  97.977

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 126.02510    9.67731  13.023  < 2e-16 ***
age           0.42023    0.03472  12.102  < 2e-16 ***
support      -3.64080    1.17204  -3.106  0.00193 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.31 on 1453 degrees of freedom
Multiple R-squared:  0.108,     Adjusted R-squared:  0.1068
F-statistic: 87.96 on 2 and 1453 DF,  p-value: < 2.2e-16
```



support の係数が −3.6、age の係数が 0.42 でともに有意水準 5%で有意だが、モデル全体の説明力は低く、自由度調整済重相関係数の二乗（Adjusted R-squared）が 10.7%しかない。

この場合、血圧に影響する要因は部署ごとに異なっているだろうし、個人差もあるだろうと考える方が自然である。それらの変量効果を考えるマルチレベルモデルの結果を以下示す。職場単位でネストされた構造（レベル 1 が各対象者個人、レベル 2 が部署）とする。

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite approximations to
  degrees of freedom [lmerMod]
Formula: bp_s ~ age + support + (1 | support)
   Data: wps

     AIC      BIC   logLik deviance df.resid
 12040.6  12067.0  -6015.3  12030.6     1451

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9741 -0.6951 -0.0728  0.6009  6.3136

Random effects:
 Groups   Name        Variance Std.Dev.
 support  (Intercept)   9.666   3.109
 Residual             222.703  14.923
Number of obs: 1456, groups:  support, 34

Fixed effects:
             Estimate Std. Error         df t value Pr(>|t|)
(Intercept)  130.42946   16.23625   39.70000   8.033 7.54e-10 ***
age            0.39737    0.03715 1202.90000  10.697  < 2e-16 ***
support       -4.08753    1.99687   37.00000  -2.047   0.0478 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
        (Intr) age
age     -0.224
support -0.994  0.124

Computing profile confidence intervals ...
                 2.5 %      97.5 %
.sig01       2.0576492   4.5851929
.sigma      14.3931343  15.4866022
(Intercept) 98.3737422 163.9122155
age          0.3244992   0.4702249
support     -8.2163934  -0.1418429
```

藤野ら (2013) の図 10-6 とほぼ同じ結果が得られた。以上の結果は下表のようにまとめることができる。

表. 収縮期血圧への年齢と職場サポートの固定効果と職場サポートの変量効果を考慮したマルチレベル分析

固定効果の変数	係数	標準誤差	95%下限	95%上限	p 値
年齢 (age)	0.397	0.037	0.324	0.470	<0.001
サポート (support)	−0.409	1.997	−8.216	−0.142	0.048
切片	130.4	16.2	98.4	163.9	<0.001
固定効果間の相関					
年齢-切片	−0.224				
サポート-切片	−0.994				
サポート-年齢	0.124				
変量効果の変数	標準偏差	95%下限	95%上限		
サポート	3.11	2.06	4.59		
個人差	14.92	14.39	15.49		

AIC=12040.6, N=1456, 部署数=34

最後に support の変量効果だけを考え、固定効果を考えないモデルと尤度比検定すると、以下のように support の固定効果を入れたモデルの方が有意に当てはまりが良いといえた。

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite approximations to
  degrees of freedom [lmerMod]
Formula: bp_s ~ age + (1 | support)
   Data: wps

     AIC      BIC   logLik deviance df.resid
 12042.7  12063.8  -6017.3  12034.7     1452

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9151 -0.6928 -0.0702  0.5965  6.3649

Random effects:
 Groups   Name        Variance Std.Dev.
 support  (Intercept)  10.8     3.286
 Residual             223.0    14.935
Number of obs: 1456, groups:  support, 34

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept) 9.743e+01  1.833e+00 3.200e+02   53.14   <2e-16 ***
age         4.063e-01  3.701e-02 1.121e+03   10.98   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
age -0.903

Computing profile confidence intervals ...
                  2.5 %      97.5 %
.sig01       2.1696680   4.8594543
.sigma      14.4037990  15.4990512
(Intercept) 93.8309194 101.0484448
age          0.3335201   0.4789125

Data: wps
Models:
..1: bp_s ~ age + (1 | support)
object: bp_s ~ age + support + (1 | support)
       Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
..1     4 12043 12064 -6017.3    12035
object  5 12041 12067 -6015.3    12031 4.1136      1    0.04254 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

藤野ら (2013) では、res4 に得られている年齢とサポートの固定効果とサポートの変量効果を考えた混合効果モデルを、res1 に得られている年齢とサポートから血圧への線形回帰モデルと尤度比検定で比べている。ただし、lm() と lmer() の結果オブジェクトのクラスが違うので、R では anova() 関数に渡すだけで尤度比検定を実行することはできない。この問題の解決策としては、スタンフォード大学のチュートリアルページに書かれている[16]ように、尤度比と自由度を手計算すれば良い。

```
> print(devdif <- as.numeric(-2*(logLik(res1)-logLik(res4))))
[1] 43.88222
> print(dfdif <- attr(logLik(res4), "df")-attr(logLik(res1), "df"))
[1] 1
> pchisq(devdif, dfdif, lower.tail=FALSE)
[1] 3.487428e-11
```

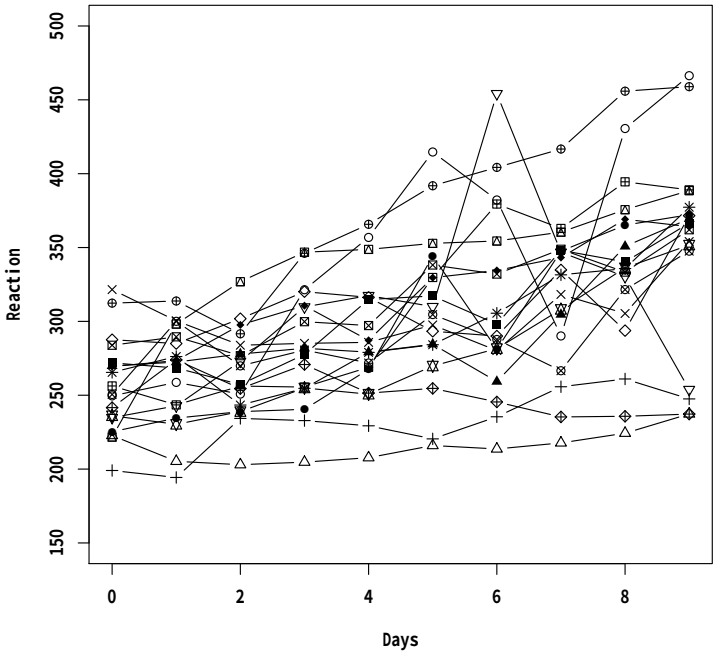
[16] https://web.stanford.edu/class/psych252/section/Mixed_models_tutorial.html

7.3.5 例 4：R 組み込みデータから

lmer パッケージの sleepstudy というデータを使う。180 オブザーベーション、3 変数からなり、含まれている変数は Reaction、Days、Subject である。このデータは、健康なボランティアを対象にして、睡眠時間を奪うと反応時間がだんだん長くなっていくことを検証したものである（Belenky *et al.*, 2003)[17]。実験 0 日目には普通に睡眠をとってもらい、翌日から 3 時間に睡眠時間を制限する（元論文ではベースラインは 3 日間 8 時間睡眠、実験期間中は 3 時間の他に、5 時間、7 時間、9 時間という実験条件を 4 群でそれぞれ 7 日間続け、最後に 3 日間 8 時間睡眠としているが、このデータは 3 時間睡眠実験群しか含んでいない)。反応時間は、LED を使った視覚刺激提示後に指で反応するまでの時間をミリ秒単位で計測している。

sleepstudy.R

```
if (require(lme4)==FALSE) {
 install.packages("lme4", dep=TRUE); library(lme4) }
data(sleepstudy)
str(sleepstudy)
bysubjects <- split(sleepstudy[, 2:1], sleepstudy[, 3])
plot(bysubjects[[1]], type="b", ylim=c(150, 500))
for (i in 2:length(bysubjects)) points(bysubjects[[i]], type="b", pch=i)
res1 <- lmer(Reaction ~ Days + (Days | Subject), data=sleepstudy)
summary(res1)
```



[17] https://doi.org/10.1046/j.1365-2869.2003.00337.x からフルテキスト読める。

```
Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ Days + (Days | Subject)
   Data: sleepstudy

REML criterion at convergence: 1743.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9536 -0.4634  0.0231  0.4634  5.1793

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 612.09   24.740
          Days         35.07    5.922   0.07
 Residual             654.94   25.592
Number of obs: 180, groups:  Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  251.405      6.825   36.84
Days          10.467      1.546    6.77

Correlation of Fixed Effects:
     (Intr)
Days -0.138
```

ランダム効果の標準偏差の推定値は、切片と傾きについてそれぞれ一日当たり 24.74 ミリ秒と 5.92 ミリ秒である。固定効果の係数は、切片と傾きについて、それぞれ一日当たり 251.4 ミリ秒と 10.47 ミリ秒である。

7.4 傾向スコアを用いたモデル推定、DID、操作変数法

多数の交絡因子が想定される場合、それをうまく調整したモデル推定がいろいろ提案されてきている。有名なものが、傾向スコアを用いたモデル推定、DID（Difference in Differences)、操作変数を用いた回帰モデルである。これらの多くは econometrics と呼ばれる経済統計分析の分野で発達してきた（星野, 2016; 山口, 2016）。

傾向スコア (propensity score) とは、ローゼンバウムとルービンが 1983 年に提案した概念 (Rosenbaum and Rubin, 1983) である。傾向スコアを説明する前に因果効果としての平均処置効果について説明する。

調査研究ではランダム割り付けが不可能だが、何らかの要因あるいは処置について、それがある場合とない場合に母集団に期待されるアウトカムの差である平均処置効果 (Average Treatment Effect: ATE) を推定することは不可能ではない。もちろん実際には母集団の中に要因あるいは処置がある人とない人が混在していて、要因あるいは処置がある人にとって「なかった場合」は反実仮想なので、ランダム割り付けができない限り、直接推定することはできない。しかし、要因あるいは処置がある i さんの共変量の値と同じか近い値をもつ j さんを同じ人とみなすというマッチングをして、2 人ずつペアで処置群と対照群をとっていけば、ランダム割り付けと同じように、これら 2 群のアウトカムの差として因果効果が推定できる。しかし共変量（交絡因子）が多いと、現実問題としてそれらがすべて同じか近い値をとるマッチングは不可能である。

そこで登場するのが傾向スコアである。上述の例で i さんが処置群に存在する確

率を i さんの傾向スコアと呼ぶ。傾向スコアは 1 つの確率なので、[0, 1] の値をとる 1 次元の変数である。ここで群別と潜在的結果変数が独立であるという「強い意味での無視可能性」が満たされていれば、傾向スコアを用いた調整をすることが多次元の共変量すべてについて調整したのと同じことになる。つまり、多数の共変量を使って計算された傾向スコアを使ってマッチングすれば、正しく平均処置効果を推定することができる。傾向スコアとしては、通常、ロジスティック回帰分析やプロビット回帰分析から得られる予測確率を用いることができる。

算出した各人の傾向スコアの利用方法は、その値によりマッチングしたり層別解析する他、傾向スコアを説明変数としたカーネル回帰モデル、傾向スコアによる逆確率重み付け法、「二重にロバストな推定法」などいろいろある。

7.4.1 DID

DID の考え方については、山口 (2016) による説明がわかりやすいので簡単に紹介する。育児支援が女性の労働に及ぼす効果の研究において、一般に認可保育所の整備が女性の就業率を上げると思われているが、ノルウェー、フランス、米国等での先行研究によると、公的保育の整備にもかかわらず母親の就業は増えなかったという報告があるので、認可保育所の整備が母親就業につながるかを日本のデータで検証したという話である。彼らは認可保育所整備と女性就業率の都道府県間格差に注目した。まず、0-5 歳の子供 1 人当たりの認可保育所定員数を保育所定員率と名付け、これを横軸にとって、0-5 歳の子供をもつ母親の就業率を縦軸にとってプロットすると正の相関関係が見られることを示し、このことが認可保育所の整備が母親就業率を上げるという印象を与えることに触れた後で、それが県民性の違いによる（母親の就業意欲が高く地域社会がそれに対して好意的ならば母親就業率も上がるし政治的支持を得やすいので保育所整備も進む）という可能性を指摘し、その解析のために縦軸横軸とも 2005 年から 2010 年まで 5 年間の変化（階差）をとってプロットすることによって県民性の影響を排除すると（もし認可保育所の整備によって母親就業率が上がるならば変化同士も相関しているはずなのに）相関が消えてしまうことを示した。

そこで DID を使うには、状況を単純化する。どの都道府県でも保育所定員率が増えているけれども、大きく増えた都道府県と少ししか増えなかった都道府県に二分して考え、前者が保育所を増やすという処置をした結果であると考えて、処置群と呼ぶことにする（残りが対照群）。処置群、対照群について、それぞれ各時点における 0-5 歳の子供をもつ家計数で重み付けした平均母親就業率を求め、2005 年の処置群、2010 年の処置群、2005 年の対照群、2010 年の対照群の順に、その値を A、B、C、D と書くと、2010 年における保育所定員率と母親就業率に有意な相関があることは、B と D に有意差があることに相当する。対照群は無視して処置前後で母親就業率に差があるかをみるという A と B の比較では、保育所整備以外の経済・社会的情勢の変化の効果も含まれてしまうので、その部分、つまり経済・社会的情勢の変化の効果が、処置がなかったところでの母親就業率の変化である C と D の差に現れると考えれば、もし経済・社会的情勢が変化しなかったら処置の効果はどうなるか、つまり $(B-A)-(D-C)$ によって処置の効果を評価することができる。これが「差の差」である。

実際に処置効果を推定するための回帰モデルは、都道府県 p における t 年の母親就業率を Y_{pt} とし、D_p^T を都道府県 p が処置群に属するかどうかを示すダミー変数（属していれば 1、対照群なら 0）とし、D_{t+1} を年次ダミー（t 年のデータに対して 0、$t+1$ 年のデータに対して 1）として、

$$Y_{pt} = \alpha + \beta D_p^T D_{t+1} + \gamma D_p^T + \delta D_{T+1} + \epsilon_{pt}$$

を推定すれば良いとのことなので（**処置効果は β が有意かどうか、経済・社会情勢の変化の効果は δ が有意かどうかで見ることができる**）、`lm()` で分析可能なはずである。

なお、就業構造基本調査で「育児をしている」という区分は未就学児についての集計であり、平成 24 年の未就学児の育児をしている女性のデータは https://www.e-stat.go.jp/SG1/estat/GL08020103.do?_xlsDownload_&fileId=0000064640 31&releaseCount=1 から Excel 形式で得られるが、ここで使われているのは国勢調査と書かれていて、2005 年と 2010 年のデータとのことなので、e-Stat で探してみたが、どこにあるのかわからなかったため[18]、実際の分析を示すことはできない。

Annual Reviews of Public Health に、Wing C et al. "Designing Difference in Difference Studies: Best Practices for Public Health Policy Research"[19] という論文が 2018 年に掲載されていて、公衆衛生政策の評価についても DID 解析を使う研究デザインが役に立つことが示されている。

R での DID の実行方法については、プリンストン大学のサイトに演習用プレゼンテーションファイルが掲載されている[20]。これはグループと時点の交互作用を示す変数を作って、線形回帰モデル lm() を使う方法なので、とくにパッケージなどは必要としない。A から G の 7 つの国について 1990 年から 1999 年までの 10 年間の何かの量（y）が示されていて、E、F、G の 3 つの国では何らの処置がとられ、処置は 1994 年からとられたと想定し、処置がとられた国ととられていない国で y の変化に違いがあるかを DID 法で評価している、コードは以下の通り。

https://minato.sip21c.org/advanced-statistics/princetondid.R

```
library(foreign)
dat <- read.dta("https://dss.princeton.edu/training/Panel101.dta")
dat$time <- ifelse(dat$year>=1994, 1, 0)
dat$treated <- ifelse(dat$country %in% LETTERS[5:7], 1, 0)
dat$did <- dat$time * dat$treated
didreg <- lm(y ~ treated + time + did, data=dat)
summary(didreg)
```

結果は以下の通り。

[18] e-Stat から入手できると書くだけではなく、データベース名あるいは URL を明記しておいて欲しいところ。

[19] https://www.annualreviews.org/doi/10.1146/annurev-publhealth-040617-013507

[20] https://www.princeton.edu/~otorres/DID101R.pdf

```
Call:
lm(formula = y ~ treated + time + did, data = dat)

Residuals:
       Min          1Q      Median          3Q         Max
-9.768e+09 -1.623e+09  1.167e+08  1.393e+09  6.807e+09

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.581e+08  7.382e+08   0.485   0.6292
treated      1.776e+09  1.128e+09   1.575   0.1200
time         2.289e+09  9.530e+08   2.402   0.0191 *
did         -2.520e+09  1.456e+09  -1.731   0.0882 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.953e+09 on 66 degrees of freedom
Multiple R-squared:  0.08273,   Adjusted R-squared:  0.04104
F-statistic: 1.984 on 3 and 66 DF,  p-value: 0.1249
```

に示す通り、time の係数が正で p 値が 0.0191 と 5%有意なので、時間経過によって y は増加していると言えるが、did の p 値が 0.088 と 0.05 より大きいので、時点と処置の交互作用項である did が有意に y に影響しているとは言えない。つまり、処置があってもなくても、y が経時的に増加したことに影響はなかったと考えられる。

R で DID を実行するために開発されたパッケージとして did[21]があり、cran からインストールできるし、開発者による使い方の解説記事も発表されている[22]。

その他の情報としては、https://thetarzan.wordpress.com/2011/06/20/differences-in-differences-estima や、https://www.publichealth.columbia.edu/research/population-health-methods/difference-difference-estimation や https://static1.squarespace.com/static/59371c8ad1758ebe90723e40/t/5cdf25f58f78a100018f804f/1558128117692/strumpf+2017-DD+and+FE.pdf も参考になる。

7.4.2 操作変数法

操作変数法は、構造方程式モデルのところで紹介した Dr. John Fox のチュートルアル文書の中で sem パッケージの tsls() 関数を使った方法が説明されているが、AER パッケージの ivreg() 関数を推奨する。

7.4.3 二段階最小二乗法の関数 tsls() による操作変数法

組み込みデータ Klein を用いる。このデータは 1921 年から 1941 年の米国経済について Klein が発表した単純な経済測定モデルに使われている。変数の意味は以下の通りである。

[21] https://cran.r-project.org/web/packages/did/did.pdf
[22] https://bcallaway11.github.io/did/articles/did-basics.html

Year 1921-1941

C consumption.

P private profits.

Wp private wages.

I investment.

K.lag capital stock, lagged one year.

X equilibrium demand.

Wg government wages.

G government non-wage spending.

T indirect business taxes and net exports.

https://minato.sip21c.org/advanced-statistics/tsls.R

```
library(sem)
data(Klein)
Klein$P.lag <- c(NA,Klein$P[-22])
Klein$X.lag <- c(NA,Klein$X[-22])
# model 1
Klein.eqn1 <- tsls(C ~ P + P.lag + I(Wp+Wg),
 instruments=~G + T + Wg + I(Year-1931) + K.lag + P.lag + X.lag, data=Klein)
summary(Klein.eqn1)
plot(Klein$Year[-1], residuals(Klein.eqn1))
# model 2
Klein.eqn2 <- tsls(I ~ P + P.lag + K.lag,
 instruments=~G + T + Wg + I(Year-1931) + K.lag + P.lag + X.lag, data=Klein)
summary(Klein.eqn2)
plot(Klein$Year[-1], residuals(Klein.eqn2))
# model 3
Klein.eqn3 <- tsls(Wp ~ X + X.lag + I(Year-1931),
 instruments=~G + T + Wg + I(Year-1931) + K.lag + P.lag + X.lag, data=Klein)
summary(Klein.eqn3)
plot(Klein$Year[-1], residuals(Klein.eqn3))
```

7.4.4 ivreg による操作変数法

AER パッケージの CigarettesSW というデータフレームは、米国の州ごとのタバコ消費量と価格や税金等の関連因子のデータを含んでいる。詳細は?CigarettesSW とプロンプトに打てば表示される。

https://minato.sip21c.org/advanced-statistics/ivreg.R

```
library(AER)
data("CigarettesSW")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs-tax)/cpi)
CigarettesSW$rtax <- with(CigarettesSW, tax/cpi)
CigarettesSW$lrprice <- log(CigarettesSW$rprice)
CigarettesSW$lrincome <- log(CigarettesSW$rincome)
CigarettesSW$lpacks <- log(CigarettesSW$packs)
CSW1995 <- subset(CigarettesSW, year=="1995")

fm <- ivreg(lpacks ~ lrprice + lrincome | lrincome + tdiff + rtax, data=CSW1995)
summary(fm)

fm2 <- ivreg(lpacks ~ lrprice | tdiff, data=CSW1995)
anova(fm, fm2)

library(sem)
M1 <- tsls(lpacks ~ lrprice + lrincome,
 instruments = ~ lrincome + tdiff + rtax, data=CSW1995)
summary(M1)
```

上記コードを実行すると、まず、summary(fm) で以下が表示される。

```
Call:
ivreg(formula = lpacks ~ lrprice + lrincome | lrincome + tdiff +
    rtax, data = CSW1995)

Residuals:
       Min         1Q     Median         3Q        Max
-0.6006931 -0.0862222 -0.0009999  0.1164699  0.3734227

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.8950     1.0586   9.348 4.12e-12 ***
lrprice      -1.2774     0.2632  -4.853 1.50e-05 ***
lrincome      0.2804     0.2386   1.175    0.246
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1879 on 45 degrees of freedom
Multiple R-Squared: 0.4294,Adjusted R-squared: 0.4041
Wald test: 13.28 on 2 and 45 DF,  p-value: 2.931e-05
```

次に、anova(fm, fm2) で以下が表示される。

```
Analysis of Variance Table

Model 1: lpacks ~ lrprice + lrincome | lrincome + tdiff + rtax
Model 2: lpacks ~ lrprice | tdiff
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     45 1.5880
2     46 1.6668 -1 -0.078748 1.3815  0.246
```

最後に summary(M1) で以下が表示される。このように sem パッケージの tsls() 関数を使っても、AER パッケージの ivreg() 関数を使った場合に得られる summary(fm) の出力とほぼ同じ結果が得られるが、自由度調整済重相関係数の二乗やウォルドの検定結果は表示されないので、AER パッケージの使用を推奨する。

```
 2SLS Estimates

Model Formula: lpacks ~ lrprice + lrincome

Instruments: ~lrincome + tdiff + rtax

Residuals:
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-0.6006931 -0.0862222 -0.0009999  0.0000000  0.1164699  0.3734227

              Estimate Std. Error  t value   Pr(>|t|)
(Intercept)  9.8949555  1.0585599  9.34756 4.1209e-12 ***
lrprice     -1.2774241  0.2631986 -4.85346 1.4960e-05 ***
lrincome     0.2804048  0.2385654  1.17538    0.24602
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.187856 on 45 degrees of freedom
```

Chapter 8

文献・サイト

8.1 R について

2004 年に拙著『R による統計解析の基礎』を出版したときは R だけをターゲットにした和文の参考書は皆無だったが、2015 年現在では 100 冊を遙かに超える書籍が発行されている。web サイトも数え切れないほど存在する。

8.1.1 概要を知るために

- 開発元 (R-project)：https://www.r-project.org/
- ダウンロード (cran)：https://cran.r-project.org/、日本では統計数理研究所 (https://cran.ism.ac.jp/) のミラーサーバを利用すると良い。パッケージを検索するには英語だが METACRAN というサイト https://www.r-pkg.org/ が便利。
- 日本語での知識の集積先は RjpWiki：http://www.okadajp.org/RWiki/
- 英語でのブログやニュースの紹介 R-bloggers：https://www.r-bloggers.com/
- 中澤の R の tips ページ：https://minato.sip21c.org/swtips/R.html
- 中澤　港『R による保健医療データ解析演習』ISBN 978-4-89471-755-8、2007 年 12 月、ピアソン・エデュケーション：医学・保健学領域の実例データを数多く掲載して、具体的に R を使って解析するためのコードと結果のまとめ方、解釈の仕方を解説した本だったが絶版。ただしサポートページ (https://minato.sip21c.org/msb/index.html) から絶版時の全文 pdf ファイルを無料で利用できる。
- 舟尾暢男『The R Tips　第 2 版』ISBN 978-4-274-06783-9、2009 年 11 月、オーム社：オンラインで公開されていたものが出版された第 1 版も機能別マニュアルとして使うのに便利であったが、九天社倒産にともない、オーム社から第 2 版として再刊される際に、大幅に加筆修正されている。
- U. リゲス（著）、石田基広（訳）『R の基礎とプログラミング技法』ISBN4-431-71218-6、2006 年 2 月、シュプリンガー・ジャパン：著者は R Development Core Team の一人であり、R のオブジェクト指向言語としての側面を強調した点に特徴がある。

8.1.2 リファレンス

- 石田基広『改訂 2 版 R 言語逆引きハンドブック』ISBN 978-4-86354-147-4、2014 年 5 月、C&R 研究所：約 700 ページにわたって R でできることが広く解説されている。目次から実現したい機能を探し、該当ページを見るという使い方が良いと思う。SECTION-010 において、RStudio についてもかなり詳しく説明されている。RStudio で使えるインタラクティヴな描画制御パッケージ manipulate の使い方など、RStudio をより深く使いこなしたい方には、同じ著者による『R で学ぶデータ・プログラミング入門：RStudio を活用する』ISBN 978-4-320-11029-8、2012 年 10 月、共立出版が参考になる。

- 間瀬　茂『R プログラミングマニュアル [第 2 版] —R バージョン 3 対応—』ISBN 978-4-86481-015-9、2014 年 5 月、数理工学社：RjpWiki に集積された Tips の多くを含んでおり、R のプログラムを書くとき、やりたいプロセスに該当する部分を目次から探して読むというのが正しい使い方である。

8.2 因子分析について

- エディンバラ大学の心理学者、Timothy Bates 教授のウェブサイトが大変助けになったが、既に消滅しているようだ。

- 群馬大学の青木繁伸教授により提供されているウェブページ (`http://aoki2.si.gunma-u.ac.jp/lecture/PFA/pfa6.html`) の説明も大変わかりやすい。KMO と MSA を計算する関数の定義 (`http://aoki2.si.gunma-u.ac.jp/R/kmo.html`) と Bartlett の球面性検定の関数定義 (`http://aoki2.si.gunma-u.ac.jp/R/Bartlett.sphericity.test.html`) も提供されている。

- B. エヴェリット（著）、石田基広、石田和枝、掛井秀一（訳）『R と S-PLUS による多変量解析』ISBN 978-4-621062203、2012 年 2 月、丸善出版：因子分析のみならず、R を使ってさまざまな多変量解析をする方法が説明されている。

8.3 構造方程式モデリングについて

- 豊田秀樹『共分散構造分析 [R 編]—構造方程式モデリング』ISBN 978-4-489-02180-0、2014 年 4 月、東京図書：構造方程式モデリングについて、R と `lavaan` パッケージを使って実行する方法が、入門から応用まで実践的に書かれている。Ωnyx についてもインストール方法から簡単な使い方まで紹介されているが、著者は `lavaan` の文法を覚えて直接コードを書くことを推奨している。出版社のサイト（`http://www.tokyo-tosho.co.jp/books/978-4-489-02180-0/`）からテキスト中で使われているデータをダウンロードできる。

8.4 マルチレベルモデルについて

- 藤野善久、近藤尚己、竹内文乃『保健医療従事者のためのマルチレベル分析活用ナビ』ISBN 978-4-787820532、2013 年 9 月、診断と治療社：テキスト中で使われているデータは保健医療分野に特化しており、かつ `http://www.shindan.co.jp/download/index.php?pcode=205300` から csv 形式等でダウンロードできるのが便利である。説明も大変わかりやすい。

- Faraway JJ (2006) *Extending the linear model with R: Generalized linear, mixed effects and nonparametric regression models.*, Chapman and Hall (especially Chapter 8).

8.5 傾向スコアと操作変数法について

- 星野崇弘 (2016) 統計的因果効果の基礎：特に傾向スコアと操作変数を用いて. 岩波データサイエンス Vol.3: 62-90.
- 山口慎太郎 (2016) 差の差法で検証する「保育所整備」の効果：社会科学における因果推論の応用. 岩波データサイエンス Vol.3: 112-128.：論文（`https://www.sciencedirect.com/science/article/pii/S088915831500043X`）を著者自身が解説している記事。
- Rosenbaum PR, Rubin DB (1983) The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70: 41-55.
- 星野崇弘 (2009)『調査観察データの統計科学：因果推論・選択バイアス・データ融合』岩波書店 ISBN 978-4-00-006972-4.
- `https://www.randpy.tokyo/entry/r_propensity_score` は R のコードも載っていて、傾向スコアを用いた解析について大変わかりやすい解説になっている。

Index

#, 12
.RData, 6
.Renviron, 6
.Rprofile, 6
< −, 12
?, 13
??, 13
$, 11
%in%, 13
[[]], 11
[], 10
“yes” tendency, 16
2 値化, 23
95%信頼区間, 20

AIC(), 21
alpha()—psych, 20, 21
Amelia, 23
aov(), 22
array(), 10
as.character(), 8
as.integer(), 8
assocstats()—vcd, 21
attr(), 11

barplot(), 23
biased question, 17
boxplot(), 23
byrow=TRUE, 10

c(), 10
cbind(), 10
CRAN, 2
CronbachAlpha()—fmsb, 20, 21
cut(), 9

dep=TRUE, 5

example(), 13
EZR, 1, 3

factor(), 8
FALSE, 8
FFQ, 17
fisher.test(), 21
fmsb, 20–22
for, 13
function(), 12

GitHub, 5
glm(), 21

hist(), 23
history(), 7

identify, 46
if, 13
ifelse(), 13

jamovi, 1

Kappa.test()—fmsb, 22

length(), 12
Linux, 3
list(), 11
lm(), 13
ls(), 8

MacOS, 3
mantelhaen.test(), 10, 21
matrix(), 10
mcnemar.test(), 22
mice, 23
mode(), 12
mosaicplot(), 11, 23

NA, 8
NagelkerkeR2()—fmsb, 21
names(), 12
NULL, 8

oddsratio()—fmsb, 21
oddsratio()—vcd, 21
oneway.test(), 22
options(stringsAsFactor=FALSE), 31

ordered(), 9

paste(), 31
plot(), 11, 23
polychor()—polycor, 21
polycor, 21
psych, 20, 21, 64

q(), 12
qqnorm(), 23

R_USER, 6
rbind(), 10
read.delim(), 31
reliability()—semTools, 20
RjpWiki, 3
RStudio, 1, 3

SAS, 1
semPaths()—semPlot, 74
semTools, 20
seq(), 9
sprintf(), 31
SPSS, 1
str(), 12
str_count()—stringr, 31
stringr, 31
stripchart(), 23
strsplit(), 31
substr(), 31
sum(), 12
S 言語, 13

t(), 10
t.test(), 13, 22
table(), 11, 21
TRUE, 8
t 検定, 22

vcd, 21

xtabs(), 11, 21

アウトカム, 22

威光暗示効果, 16
依存, 5
一元配置分散分析, 22
一致度, 22
因子分析, 15, 21, 49

永続付値, 13

エラー, 8

大文字, 8
オッズ比, 21
オブジェクト, 8

改行, 12
回答選択式, 18
介入効果, 22
κ 係数, 22
カテゴリ化, 9
カテゴリ変数, 8
環境変数, 6
関数, 8, 12
管理者権限, 3
関連の強さ, 21

起動アイコン, 6
キャリー・オーバー効果, 21
行列, 10, 11

区間, 10
クロス集計, 21, 29
クロス集計表, 11, 22
クロンバックの α 係数, 15, 19–21, 64

欠損値, 23

合計得点, 17, 19
構成概念, 17
構造方程式モデル, 22
行動, 15
項目分析, 20
コクラン＝マンテル＝ヘンツェルの要約
　　カイ二乗検定, 21
小文字, 8

再カテゴリ化, 13
作業仮説, 15
作業ディレクトリ, 6, 7
サブクエスチョン, 21
散布図, 23

式, 8
次元, 10
実験, 22
実数, 8
質問群, 17
質問紙調査, 15
四分相関係数, 21
尺度分析, 20
重回帰分析, 22

自由回答, 17
集合, 10
縦断研究, 22
主成分, 49
主成分分析, 49
順序尺度, 15, 19
順序付きファクター型, 9
条件分岐, 13
食事調査, 17
食物摂取頻度調査, 17
序列質問, 18
信頼性, 15, 19, 20, 22

水準, 8, 9
スカラー型, 8
スクリプト, 7
スコア化, 18
ステレオタイプ, 16
ストリップチャート, 23
スピアマン・ブラウンの公式, 20

正規確率プロット, 23
正規分布, 15, 23
制御構造, 13
整数, 8
生存時間解析, 22
折半法, 19
全角, 8
線形回帰分析, 13
潜在因子, 15, 49
潜在因子構造, 15
専門用語, 16

相関係数, 19
総称的関数, 11
属性, 15
測定限界, 22

多重代入法, 23
妥当性, 17
タブ区切りテキストファイル, 11, 20
ダブルバーレル, 17
ダミーテーブル, 16
単位, 16

治験, 22
知識, 15
調査票, 15

定義, 12
データの分布, 23
データフレーム, 10, 11, 20, 25

テーブル, 10, 11
テスト, 15
点推定量, 20
転置, 10

毒性試験, 22
独立性, 21
度数分布図, 23

内的一貫性, 20
内的一貫性尺度, 18

二重引用符, 8
24 時間思い出し法, 17
2 バイト文字, 3
二峰性, 23
任意尺度, 18
認識, 15

曝露, 22
箱ひげ図, 23
パッケージ, 3, 5
半角, 8
半角英数字, 3
判定尺度, 18
反復測定分散分析, 22

ヒストグラム, 23
否定的語法, 17
標準偏差, 13
評点, 22

ファクター型, 8, 9
フィールド調査, 22
フィッシャーの直接確率, 21, 22
複数選択, 18
付値, 8
プリコーディッド自由回答, 17
プロジェクト, 4
ブロック, 12
プロビット解析, 22
プロンプト, 7
分散分析, 22

平均, 13
ベクトル, 8, 10
変数, 8

マクネマーの検定, 22

ミラーサーバ, 5

モザイクプロット, 11, 23

文字列, 8
文字列型, 8, 31
文字列操作, 30

有効数字, 22
ユーザ名, 3

要素, 8, 10
予約語, 8

リスト, 10, 11
リッカート尺度, 15, 18

ループ, 13

連結可能匿名化, 22
連続変数, 9

濾過質問, 17
ロジスティック回帰分析, 21
ロジット解析, 22
論理値, 8