RとStanで学ぶフリーで楽しい心理統計の世界

心理学データ解析応用1

データの背後のメカニズムを解析する方法

小杉考司

心理学データ解析応用 1

小杉 考司

Last Compiled on 2025.12.3

はじめに

昨今はデータサイエンス，情報科学の領域が非常に隆盛で，コンピュータを使ってデータを分析し，経済の動向や購買行動などの予測に用いられることが広く行われている。

人の行動や考え方をどのようにデータにするかについては，当然ながら心理学には一日の長がある。また，人が頭の中でどのような考え方のプロセスをたどるのか，それをどのように検証するのかについても，心理学はその短い歴史の中で徹底的にその技法を洗練させてきた。このような根源的なレベルでの理論や方法論は時代が変わっても色褪せることなく，また今後ますます必要とされてくる時代になっている。

本講ではデータ解析の応用段階として，より実践的なテーマを扱う。すなわち，**心理尺度が作られる理論的背景**と，**データの背後のメカニズムを解析する方法**を知ることである。前期配当の心理学データ解析応用 1(旧カリキュラム名心理学データ解析 2A) では前者の，心理尺度に関する理論的背景について学ぶ。

心理学研究法の 1 つとして，調査研究がある。紙とペンで回答を集めた時，回答者がある反応カテゴリにまるをつけたことが，どうして数値処理の対象になるのか。そこには数字を割り振るルールとしての「尺度化」の手続きがある。残念ながら応用的側面が発展しすぎたため，回答に数字を割り振る原理について語られることが少なくなってきてしまい，それに対する反動からか，近年改めてこの根本原理についての理解と解説が求められている。この講義では，尺度化の原理や目に見えない潜在変数を想定して分析するとはどういうことかについて，理論と演習を交えながら習得することを目指す。この理論的側面を考えるためには，どうしても線形代数・行列計算の知識が必要になってくる。線形代数については特別な事前知識は不要で，定義から改めて解説するので安心してもらいたい。

なお，2024 年度から前期と後期の資料を分割して提供するようにした。通年で履修する学生が少ないことと，後期で扱うテーマとして数値シミュレーションを含むように再構成したからである。前期後期でテーマが明確になり，またスリムになった分，利用しやすくなったのではないかと思っているが，どうだろうか。

授業のテーマ

データから意味のある情報を取り出すための，さまざまな分析法を習熟するにあたって，その背後にあって語られることのない「発想」の観点から理解する。数値だけに振り回される状態から脱却し，数値を算出する数式に込められた意味について考える視点を持つ。さらにこれらに習熟することで，どのような研究対象に対してどのような心理統計的アプローチができるかを，俯瞰的に見られるようになろう。

前記を通じて伝えたいポイント

尺度化とは何か　心理学で行われるアンケート調査やその後の分析はどういう原理があって「心を測定した」といえるのか。その原理やモデルを理解して利用できるようになる。

多変量解析から何がわかるのか　調査研究などで得られた多変量を分析することで何がわかるのか。あるいは何をしてわかったというのか。

多変量解析の基礎となる数式的原理　多変量解析の背景にあるのは線形代数という数学であり，線形代数の基礎を学ぶことで多変量解析のメカニズムを統合的に理解できる。

その他

授業シラバスとこの講義資料を掲載したサイト (https://kosugitti.github.io/psychometrics_syllabus/) で，最新版のシラバスと授業資料，授業で用いるサンプルデータやコードの配布を行なっています。

目次

第 I 部

心理学データ解析応用 1

第 1 章

導入；多変量データと心理学

この授業は基礎的な心理統計を修めた人向けの，応用コースになっています。基礎的な心理統計，という言葉に私が込めた意味は，

- 記述統計；得られたデータの統計量を算出したり，可視化することによってデータの特徴を把握する。
- 推測統計；得られたデータが母集団からの標本であると考え，標本の特徴を使って母数を推測する。さらに標本の特徴から母集団について何らかの判断 (意思決定) を行う。

という 2 点です。とくに推測統計学の領域では，確率の話やさまざまな推定法，それに伴う技術などが必要ですから，これだけでも膨大な量だったのではと思います。こうした基礎的な知識や技術を身につけると，とりあえず手元のデータを使って差があるとかないとか，どの程度効果があったのか，と言った基本的な判断はできると思います。複雑な計算式のところは機械 (統計環境 R など) がやってくれますので，出た結果だけをみて判断すれば良いのです[*1]。

しかし基礎的なところだけで満足していると，「はて，何がしたかったのかな」とそもそもの問題意識を忘れてしまうことにもなりかねません。とりあえず教わった (膨大な！) プロセスを経て実験結果を見てみれば OK なんでしょう，というだけでは表面的な理解に止まっていると言わざるを得ません。さまざまなシーンに適用される方法論，その計算は何を意味していて，元々の数式にはどのような意味があり，式から得られるものは何をどこまで指し示しているのか，というところまで考えるのが，この講義の狙いです。数式は数式に過ぎない，というのはその通りなのですが，その数式にどのような意味があるのか，数式から得られる結果にどのような意味があるのか，を理解した上で数値 (データ) を扱うようになることが目的です。数値だけに振り回される状態からの脱却を目指します。

この授業 (データ解析応用 1) は，講義の形をとります。テーマとしては，**因子分析 (Factor Analysis)** を扱います。因子分析は，いわゆる質問紙調査を行った後で適用される多変量解析法の一種で，たくさんの質問項目の背後にある「因子」を見つけ出します。その因子には XXX 特性，XXX 傾向といった名前がつけられ，その上で心理学的に考察することが一般的です。因子分析の結果として性格特性が得られる，などといったりするわけですから，これはもう心理学の王道中の王道，と言えるかもしれません。しかし実際は統計ソフトウェアが計算してくれて，何だかわからないまま使っている，という人も少なくありません。質問紙調査で分析して何かがわかる，というのはどういうことなのか，原理的なところから解説を始めていきます。またこの講義ではその基礎的なところ，数式レベルでしっかりと把握した上で，数値例に進みます。講義が終わる頃には，意味内容をしっかり理解したうえで因子分析を使えるようになっていることが期待されます。

このテキストには含まれませんが，姉妹講義であるデータ解析応用 2 では，実践・演習を主にした学習にな

[*1] もちろん結果の見方が間違っていたり，拙速だったりしてはいけないなど，注意すべきところは多々あります。

ります。我々が手にしたデータは，何らかのモデル・仮定のもとで生成されたものである，という考え方に立ち，**データから生成メカニズムを推測する**ことにチャレンジします[*2]。これは非常に夢のある話です。だってデータ生成メカニズムというのは，私たちの心の中の機序そのものだからです。推測に当たってはさまざまな前提・仮定が必要ですが，得られる結果は非常に含蓄に富み，さまざまな角度から心を考えさせてくれるヒントになります。線形モデルは直線的な関係でしたが，より柔軟で豊富な表現力を持つ技術へと理解を進めていきましょう。

今回は初回ですので，基礎的な心理統計で学んだことを改めて確認・復習することを中心に，今後の講義でも用いられる数学的記号の準備をします。

1.1 正規線形モデルの世界

因子分析法や (重) 回帰分析では，基本的に扱う変数が複数あります。これまでの相関・回帰分析や，実験計画の中では，説明変数と被説明変数がひとつずつ，という二変数の世界でした。これが多くなったシーンは，一般に**多変量解析 (Multivariate Analysis)** と呼ばれます。変量あるいは**変数 (Variables)** は，ケースごとに変わる数のことを指します。変わる「数」と言っていますが，この後述べるように数字以外のものも数字として扱いますので，「ケースごとに変わるもの」の総称だと思ってください。多変量はそれが多くあるもの，データセット全体を指します。**たくさんのデータセットの中から意味ある情報を引きだすこと**，これが多変量解析の目的です。

イメージとしては，表計算ソフトのスプレッドシートに数字がずらっと並んでいる世界です。たとえば性格検査の尺度の一種である，YG 性格検査は被験者に 120 の項目について回答させます。ひとりにつき 120 の変数があり，これを何百，何千人に対して実施するのですから，非常に大量のデータになっているわけです。スプレッドシートにデータを入れていくときは一般に，一行 (横方向) に 1 ケース (1 オブザベーション，1 個人) であるようにし，列方向 (縦方向) に変数を並べるのが一般的です。数学記号では次のように表現します。

$$x_{ij}$$

ここで i は個人を，j は変数を指します。たとえば x_{13} で第 1 番目の個人 (ケース) の第 3 番目の変数 (についての値) を意味するわけです。このように一般化することで，120 ある変数でも何千分ものデータでも 1 つの記号で表現できますね。

さて，このような大量のデータを今から分析していくわけですが，どのような方法があるでしょうか。データ分析の領域には，さまざまなモデル，手法があります。数え方にもよりますが，何百という種類の分析法があるかもしれません。もちろん似通ったものもありますし，同じ目的に使う異なる手法もあります。これを大きく分けるなら，まず「線形モデル」と「非線形モデル」に分割できます。

■線形モデル　線形モデルは，回帰分析や要因計画などが含まれます。関係が直線的であること，つまり変数に 2 乗，3 乗の項が入っていないので，同じパターンで先々まで考えることができる，単純な関係です。線形というのは $y = ax + b$ のグラフを書くとわかるように，直線的な関係になることからきています。変数が x, y だけでなく,$y = ax + bz$ のように 2 つあったとしても，グラフでは線が面になるだけで，ある断面で見ると直線関係であることに変わりはありません。実際の多変量データではたくさんの項が含まれ，関数全体を可視化することは不可能ですが，次元が多くなっても直線関係であることに変わりはありません。一般に線形モ

[*2] 2024 年度までは，応用 1/2 で一つのテキストとしていましたが，応用 2 の講義計画の大幅な変更に伴いテキストを分割・再編しています。以前のバージョン 1 についてはサイトや KDP で公開しています。

デルは次の形で表現されます。

$$y = a_1x_1 + a_2x_2 + \cdots + a_nx_n + b$$

ここで a_m は第 m 番目の変数 x_m につく**係数 (coefficients)** であり，変数の重要性を示す数字です。足し合わせる時に，変数の重要性を変えるものなので**重み (weight)** と呼ばれることもあります。この**重みつき線型結合**が線形モデルの基本です。

線形モデルの代表的な分析方法としては，**回帰分析**，**重回帰分析**，**パス解析 (Path Analysis)**，階層回帰分析，**因子分析**，**主成分分析 (Principle Component Analysis)**，共分散分析，判別分析などが含まれます。また今あげたモデルをすべて含んだ表現形式である**構造方程式モデリング (Structural Equation Modeling)** があります[*3]。構造方程式モデリングは総合的な表現方法で，先にあげたモデルを下位モデルとして含むものですから，これをしっかり学べば各手法をいちいち学ぶ必要がない，とも言える究極的なモデルです。本講義では第 12 講で触れることになります。

線形モデルの中には他にも**グラフィカルモデリング** (宮川, 1997) などが含まれますが，それは本講の範囲を超えるので専門書に譲ります[*4]。さらに言えばこれら線形モデルのほとんどは**正規分布 (Gaussian Curve)** を仮定した確率モデル群だと言えます。合わせて正規線形モデルといいます。

■非線形モデル (正規) 線形モデルは，変数間関係を直線的なものだと考えるのでした。しかし，世の中のことは必ずしも線形関係ではありません。むしろ線形でない関係の方が一般的でしょう。線形関係というのは $A \to B$ のように，「こうすれば，こうなる」というわかりやすい関係ですが，人間の場合はとくに「叩いたら，泣く」「優しい言葉をかければ，喜ばれる」といったことでも成立しないことがいっぱいあるわけです。叩かれた人が，強がって見せるためにグッと我慢するとか，優しい言葉に絆されて泣いてしまうといった人間の感情の機微は，本当に興味深く複雑な仕組みですよね。

線形モデルは，現状に当てはまらないこともありますが，わかりやすさを優先して作られたモデルです。それに対して，現状に当てはまることを目的にすると，とても線形の関係では無理です。そこで非線形な関係でもいいから，データに適したモデルはないか，と考えられているのがこの非線形モデルです。非線形だからといって，曲線である，というだけではありません。たとえば条件分岐のように，枝分かれしていくような関係なども含まれます。

非線形モデルの代表的な分析方法としては，決定木，ランダム森，サポートベクターマシン，ニューラルネットワーク，ベイジアンネットワーク，アソシエーションルール，自己組織化マップなどがあります。入門書としては豊田 (2008) やその後継書豊田 (2023) などが網羅的で良いですが，より専門的には**機械学習 (Machine Learning)** などのキーワードで選書すると良いでしょう。ここでいう学習とは，データに合わせて重みを調節する方法を指し，機械が自動的に重みを調節していく様を「学習している」と表現しているわけです。巷で A.I.(人工知能) と呼ばれているものはこうした手法の総称で，データに適していることを目的にしているので，パターンがわかれば行動の予測に使えます[*5]。行動の予測ができれば，たとえば小売業では売り上げに直結する戦略が取れるでしょうし，犯罪者のプロファイリングなど，応用的側面はいろいろ考えられます。

[*3] これは**共分散構造分析 (Covariance Structural Analysis)** と呼ばれることもあります。

[*4] この方法は，ネットワーク分析とも呼ばれることがあります (Isvoranu et al., 2022)。SEM が線形モデルの王道であり線形関係を正面から捉えているのに対し，グラフィカルモデリングは関係を裏から捉える裏線形とでもいうべき手法です。もう少し言葉を足すと，SEM が相関行列をもとに関係の深そうな変数を要約した潜在変数を考え出すのに対し，グラフィカルモデリングやネットワーク分析では偏相関行列をもとに関係のなさそうな変数同士の繋がりをカットし，残った繋がりの全体像を把握するという方法です。後者は偏相関，すなわち他の変数からの影響を除去した関係を扱いますから，多重共線性のような問題に悩まされることなく分析ができるという利点があります。

[*5] 線形回帰モデルで予測しただけでも，知らない人にとっては人工知能=機械が計算した予測式だ，と思ってしまっているかもしれません。

じゃあもう非線形モデルが最強じゃないか，と思うかもしれませんが，この系列の総合的な問題点は「機械がなぜそのような予測をしたのか説明できない」という点にあります。機械には学び方を教えていますが，どう学んだかは機械次第なのです。理屈はわからなくても正解が出せる，というのは実用的にはいいのですが，科学的な研究の場合は少し困ります。機械が勝手に人の心を「うんうん，わかりますよ」と言ったとしても，「じゃあどうわかったのか，理屈を教えてください」といっても答えられなかったり，同じアルゴリズムでも違う機械が学習すると違う重み係数になったりして一般的な理屈が出てこないということがあります。心理学は実践的な側面もありますが，究極的には人間行動の理論を探しているので，そういう意味では非線形モデルは向いていないのです。

■モデルの展開と全体像　多変量解析の分析方法を，線形モデルと非線形モデルに分割して説明してきました。一言でいうなら，線形モデルは「当てはまらないこともあるけど，理屈で説明がつくモデル」であり，いわば理屈が先，現象が後です。非線形モデルは「理屈はわからないけど，データには当てはまるモデル」であり，いわば現象が先，理屈が後なのです。

どちらもデータとの当てはまりを基準にしてはいますが，その背後の考え方が違っているわけですね。(正規) 線形モデルも非線形モデルも，制約を増やしたり減らしたりしながら限界とされている点を克服するべく，日々モデルの改良がなされています。

たとえば線形モデルの世界でも，正規分布以外の**確率分布**を仮定できます。それらは**一般化線形モデル (Generalized Liner Model)** と呼ばれ，さまざまな確率分布を仮定した上で，線形関係を見出す手法です。心理学の研究対象としているデータは，目に見えない心的状態を対象とすることが多いので，正規分布を仮定することが一般的でした。もちろん数学的に，正規分布を仮定するモデルの方が単純な形になるので，そちらの発展が先に進んだという実情もあります。正規分布以外の形をするデータがなかったわけではありません。所得のデータは対数正規分布のようになりますし，比率のデータは 0 から 1 までの範囲にしか値を取らないので平均が 0.5 からズレれば左右対称の形にはなりません。友人の数を数える時のように，0，1，2... と正の整数しか取らない離散的なデータというのもあります。しかし分析モデルが正規分布を仮定したものしかなかったので，これまではデータを正規分布の形になるように変換して分析する，ということが行われてきました。今は一般化線形モデルを使って，データの形式にあった分析をすることが基本です[*6]。

このように，正規分布の線形モデルが非常に多くあるのですが，その制約を外す方向で統計モデルが展開してきているわけです。制約の外し方としては，ここで挙げた「正規分布以外の確率モデルを使う」ということもありますし，分布の歪度･尖度といったより高次の**積率 (moment)** を使う方法があります[*7]。

また，**数理モデリング (Mathematical Modeling)** というアプローチもあります。これは線形の仮定を外し，理屈の通る変数関係を数式で表現してデータにフィットさせるというアプローチです。当然非線形な関係になりますし，正規分布以外の確率分布も使います。この時，さまざまな確率分布が入れ子になったモデルになるので，推定方法としてはベイズ法を使うことになります。**ベイズ法**によるモデリングアプローチは最近の計算機技術革新によってとても身近なものになりました。この辺りの内容については，後期のデータ解析応用 2 で詳しく取り扱います。

さて，多変量データを扱う世界の全体像がわかったところで，まずは線形モデルの世界から少しずつ進んでいくことにしましょう。そこで，統計モデルの種類にも大きく関わる尺度水準や，記述統計量について，改めて説明を加えておきたいと思います。

[*6] 古い論文を読むと，正解率のような比率のデータに対して**分散分析**を行ったりしていましたが，このような理由から今では推奨されません。

[*7] 積率について，詳しくはこの後の 1.3 節で。

1.2 尺度の四水準

心理統計のテキストは何を開いても，まず Stevens (1946) による**尺度水準 (level of measurement)** についての言及があります。何を測定した数値であるかとは別に，その数値にどのような算術処理を施すことができるかによって，数字を 4 つのレベルに分けるのでした。

■名義尺度水準　名義尺度水準 (nominal scale) は数字と対象が 1 対 1 で対応していることだけが重要です。男性を 1，女性を 2 とコード化するようなもので，この時「女性は男性の 2 倍である」といった数としての意味はありません。男性を 0，女性を 42 としても本質的に変わりがないからです。このような名前だけの数字は，計算ができませんので，せいぜい 1 が何件あったかという度数を数えて集計するにとどまります。

とはいえ，数字が直接対象を指し示しているわけですから，もっとも意味のある数字かもしれません。

■順序尺度水準　順序尺度水準 (ordinal scale) は，数字が大小関係の意味を持っているものです。レースで 1 位 2 位と順番がつくと，1 位のほうが 2 位より優れていることがわかります。人間の心理的な反応，とくに 5 段階や 7 段階で評定させる心理尺度は，この水準に相当します。選好の順序は明確でも，量的な違いがわからないからです。

■間隔尺度水準　間隔尺度水準 (interval scale) は，数字と数字の間隔が等しいことが制約として加わります。たとえば気温で 10 ℃と 20 ℃の差は，25 ℃と 35 ℃の差に等しいと言えます。これは摂氏が氷点を 0 ℃，沸点を 100 ℃としたうえで百等分したという定義から明らかなことです。間隔が整っているので加法・減法の計算は可能ですが，原点が定かでないので比を考えることはできません。たとえば 10 ℃は 20 ℃の倍の熱量を持っている，とは言えないのです。なぜでしょうか。たとえば，同じエネルギー状態を別の温度体系に置き換えてみたとしましょう。新しい温度体系は，氷点が 100 で沸点が 200 だったとします。そうすると 10 ℃は 110，20 ℃は 120 に該当しますが，120 は 110 の 2 倍にはなっていないからです。

■比率尺度水準　比率尺度水準 (ratio scale) は，さらに絶対 0 点の制約を付け加えたものです。これで原点からどれ位離れているか，を基準にして計算ができますので，乗法・除法もできることになりました。物理的な単位系はこの尺度水準にあるものがほとんどですから，緻密な計算モデルを作ることができるのですね。

さて早足で 4 つの水準について説明をしてきました。これが重要なのは，尺度水準によってできる計算が変わってくる点にあります。名義尺度水準は数え上げぐらいしかできません。順序尺度水準も同様で，名義や順序といった**質的変数 (categorical variables)** の場合，たとえば代表値を求める時も度数を数えて最頻値を報告する，というぐらいがせいぜいなのです。

これに対して，間隔尺度水準や比率尺度水準の**量的変数 (numeric variables)** では，加減乗除の計算ができますので，平均値を求めたり標準偏差を求めたり，ということができるようになります。

先ほど心理尺度は順序尺度水準でしかない，という話をしましたが，質問紙調査の研究例では尺度平均点を出したり，さらに進んだ統計手法で分析したりします。実はそれができるようになるためには，尺度につけられたカテゴリー (「非常に当てはまる」「どちらとも言えない」など) を，尺度値 (「非常に当てはまる」を 5，「やや当てはまる」を 4 とする，など) にする作業を経ており，その時に「非常に当てはまる」を 5 とするのはなぜか，という理屈が必要です。それについては心理尺度の作成の折に詳しく説明しますが[*8]，少なくとも盲目的に行っているわけではないことに注意が必要です[*9]。

[*8] 第 3 講を参照してください。

[*9] 残念ながら，この辺りの理屈を気にせずに分析する人が少なくありません。そんな人に，「数字では心がわからない」なんて言って欲しくないですねえ。

尺度値の付与の仕方は色々考えられます。順序尺度水準でとられたデータであっても，その背後に連続的な心理的実態があると考えてその時の相対的な大きさをつけることもできます[*10]。あるいは 0 か 1 かという二値データであっても，それを重みづけて合算することで連続的な値にすることもできます[*11]。さらに名義的な尺度水準であっても，データ全体なかで直線的な関係が最も大きくなるように数字を与えることもできます[*12]。

このように，尺度水準がかわると，適用できる分析モデルが変わります。たとえば因子分析は間隔尺度水準以上のデータにしか適用できませんが，順序尺度水準のデータであれば因子分析ではなく，段階反応モデルのようなカテゴリカル因子分析を適用しなければなりません。名義尺度水準のデータであれば双対尺度法を適用しなければなりません。間隔尺度水準以上のデータに対して，双対尺度法を適用することは可能ですが，その逆は不可能です。たとえば身長のデータとして $X = \{170, 175, 165\}$ というのがあったときに，連続変数として平均値を求めることもできますし，名義尺度水準として各一件とカウントすることもできるのですが，男性 2 名と女性 1 名がいたので平均 1.3 の性別があるというのは意味をなさないからです。

データがどの水準にあるのかを見極められないと，間違えた分析をすることにもなりますので，注意してください。

1.3 平均と分散

間隔尺度水準以上の数字であれば，平均値や標準偏差の計算が可能です。(算術) **平均 (mean)** は次の式によって計算されるのでした。

$$\bar{x_j} = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$$

総和の記号である $\sum$ の使い方などを再確認しておいてください。また変数 j の平均は $\bar{x_j}$ と表しますが，$\sum$ の記号の中では $i = 1$ から N までと，i だけが変化しています。j は変化していません。

$\sum$ の記号はこの後も所々出てきます。計算の際に次のような変形をすることがありますので確認しておいてください。

$$\sum_{i=1}^{N} cx_i = (cx_1 + cx_2 + \cdots + cx_n) = c(x_1 + x_2 + \cdots + x_n) = c\sum x_i$$

$$\sum_{i=1}^{N}(x_i + y_i) = (x_1 + y_1 + x_2 + y_2 + \cdots) = \sum x_i + \sum y_i$$

続いて**分散 (variance)** の式を確認しましょう。

$$s_x^2 = \frac{1}{N}\sum (x_i - \bar{x})^2$$

言葉でいうなら，平均偏差 $(x_i - \bar{x})$ の二乗の平均です。平均偏差は各点が平均点からどれぐらい離れているかを表します。その平均をとる操作 $(\frac{1}{N}\sum)$ なので，平均的にどれぐらい離れているかがわかるのですが，そのまま平均偏差の平均をとるとゼロになりますので[*13]，二乗するものをその値にするのでした。

[*10] 因子分析法の一種，**段階反応モデル (Gradad Response Model)** と呼ばれる手法がこれにあたります。
[*11] これはテスト理論の考え方です。0 が誤答，1 が正答としてテストの点数から学力を考えるのですね。
[*12] 双対尺度法という考え方がこれにあたります。詳しくは西里 (2010) を参照。
[*13] 平均値が全部のデータの真ん中に位置するように撮られた指標だから当然です。

この式は次のように展開できます。

$$
\begin{aligned}
s_x^2 &= \frac{1}{N}\sum(x_i-\bar{x})^2 && \text{定義より}\\
&= \frac{1}{N}\sum(x_i-\bar{x})(x_i-\bar{x}) &&\\
&= \frac{1}{N}\sum(x_i^2-2x_i\bar{x}+\bar{x}^2) && \text{(後ろのカッコを展開)}\\
&= \frac{1}{N}\sum x_i^2-\frac{1}{N}\sum(2x_i\bar{x})+\frac{1}{N}\sum\bar{x}^2 && \textstyle\sum \text{記号の分配}\\
&= \frac{1}{N}\sum x_i^2-2\bar{x}\frac{1}{N}\sum x_i+\frac{1}{N}N\bar{x}^2 && i \text{ が変化するところだけにつく}\\
&= \frac{1}{N}\sum x_i^2-2\bar{x}^2+\bar{x}^2 &&\\
&= \frac{1}{N}\sum x_i^2-\bar{x}^2 &&
\end{aligned}
$$

計算するときは最後の形を使うことがあります。

この式で表される分散は，変数がどの程度変化しうるかということを表す値であり，変数から引き出せる情報の上限でもあります。分散が 0，つまり変数がまったく変化しなければ，どういうときに値が大きくなってどういうときに値が小さくなるのかという違いがわからないということです。違いがないものについては，それ以上考察のしようがないか，当たり前のことを言っているに過ぎないということになります。たとえば質問紙調査で「他人に激しく殴られるのが好きですか」という聞き方をすると，誰しも「まったく当てはまらない」と答えると思います[*14]。この項目から何かがわかるか，といわれても当たり前のことすぎて何もわからない，としか言えないでしょう。これに対して「対面している相手の手足の動きが気になりますか」というような項目であれば，気にする人もいるでしょうし，気にしない人もいるでしょうから，回答に分散が生まれます。そうすると，気になった人はどういう人なのか，気にならなかった人はどういう人なのか，という考察に進むことになるわけです。このように，調査データから意味のある考察をするためには，分散の大きな項目を作らなければならないのです。

ところで，分散の式の中には二乗の項が入っていますから，このままでは元のデータと単位が異なってしまいます。そこでこの単位を整えるために，分散の正の平方根をとったものを**標準偏差 (Standard Deviation)** といって散らばりの指標に使います。本講では標準偏差の記号を s_x と表すことにします。

平均は中心化傾向の指標の一種で，他にも中央値，最頻値などがあります。分散や標準偏差は散らばりの指標の一種で，他に最大値，最小値，範囲，IQR，パーセンタイルなどがあります。これら記述統計量は，データの特徴を記述するため，データ分析の最初のステップで確認すべき数字です。また，分散は平均偏差を二乗したものの平均でしたが，これを三乗したものは**歪度 (skewness)** といいます。歪度 s_x^3 の式は次のとおりです。

$$s_x^3 = \frac{1}{N}\sum(x_i-\bar{x})^3$$

歪度はマイナスになると左方向に，プラスになると右方向に，分布が歪んでいることを示します。ゼロに近ければ左右対称に近いことがわかります。これも記述統計量としてデータの特徴記述に利用できるでしょう。さらに四乗したものは，**尖度 (kurtiosis)** と言われます。

$$s_x^4 = \frac{1}{N}\sum(x_i-\bar{x})^4$$

[*14] 中にはマゾヒズムの人がいるかもしれない，という屁理屈はここでは脇に置いておいてください。

尖度がゼロであれば，分布の山の尖りぐあいが平均的で，マイナスになれば潰れた山，プラスになれば尖った山の形をした分布になることがわかります。

分散が二乗，歪度が三乗，尖度が四乗でした。これらは分布の中心からどれぐらい離れているかについての累乗で，平均値は一乗したものと理解することができます。これを力学の用語を借りて**モーメント (moment)** といいます。日本語では**積率**と訳されています。重心からの距離に関する指標という意味で一般化された表現です。多変量解析のほとんどは，二次のモーメント (分散) までしか活用しませんが，3 次，4 次のモーメントを使うとなるとデータから得られる情報が増えることになり，より表現力を増した分析ができるようになります[*15]。

1.4 共分散と相関係数

さて，分散が 1 変数の変動を表現する数字であり，そこから得られる情報の上限であるという説明をしました。実際に調査的な研究をするときは，いくつもの変数を同時に扱うことになるわけですが，そうすると変数と変数の関係を考える必要があります。

改めて分散の式を見ると，次のようになっているのでした。

$$s_x^2 = \frac{1}{N}\sum(x_i - \bar{x})^2 = \frac{1}{N}\sum(x_i - \bar{x})(x_i - \bar{x})$$

この式を少し変えて次のようにすることで，変数 x と y の関係を考える式になります。

$$s_{xy} = \frac{1}{N}\sum(x_i - \bar{x})(y_i - \bar{y})$$

この式で表される数字を**共分散 (covariance)** といいます。異なる変数ですので異なる記号で現しましたが，変数を x_{ij} のように個人 i と変数 j という形で表すならば，変数 j と k の共分散を表す式は次のように書けます。添字のつき方に注意して理解してください。

$$s_{jk} = \frac{1}{N}\sum(x_{ij} - \bar{x_j})(x_{ik} - \bar{x_k})$$

さてこの共分散は，カッコの中がそれぞれの変数の平均偏差を現しており，それを個人ごとに集めて平均しています。あるケースにおいて，平均偏差が同じ方向，すなわちどちらも平均より上であるか，どちらも平均より下であれば，積の符号は正になるのでこの数字は増えます。逆にあるケースにおいて平均偏差が異なる方向，つまり一方は平均より上なのに他方は平均より下であるようなことがあれば，積の符号は負になりますので，この数字は減ります。これを平均するわけですから，データ全体で同じ方向にずれる傾向があるのか，そうでないのかを表す指標ということになります。

同じ方向にずれるということは，一方の動きがわかれば他方の動きが推測できます。このように，共分散の数字は変数間関係から予測，考察をするための情報量を現しているとも言えます。共分散がまったくない，0 であれば，ケースごとにさまざまな可能性があるので一般的なことが言えないわけです。

この共分散は，分散と同じで単位に依存します。たとえば身長と体重の共分散を計算すると，メートルとキログラムをかけ合わさった単位になります。このように，変数ごとに単位が変わると共分散同士の比較が難しくなりますので，データの標準化を考えましょう。すなわち，どのようなデータであっても単位に捉われずに比較できるようにします。スコアの標準化は次の式で行います。

$$z_i = \frac{x_i - \bar{x}}{s_x}$$

[*15] 詳しくは豊田 (2007) を参照してください。

このように平均偏差を標準偏差で割ることで，あるデータがその変数の平均からどれぐらい離れているか，標準偏差を単位とした相対的な大きさで表すことができます。標準化されたデータは，平均が 0，分散が 1 になります。このように，標準化されたスコアは相対的に同じサイズに整えられ，単位に依存しませんので，標準化したスコア同士は相対的に比較可能です。

この標準化したスコアで共分散を計算したものが，**相関係数 (correlation coefficient)** です。

$$r_{jk} = \frac{1}{N} \sum z_{ij} z_{ik}$$

この相関係数はどんな変数であっても，-1 から $+1$ の範囲に入りますので，変数間関係の相対的な比較が可能です。つまり身長と体重の関係の強さは，身長と足のサイズの関係の強さよりも大きいとか小さい，といった表現が可能になるわけです。

多変量解析の世界は，変数がたくさんありますから，この共分散の組み合わせもたくさんあります。分散が変数から得られる情報，共分散が変数間関係から得られる情報ですから，あるデータセットから得られるこれらの情報の数はどれぐらいになるか計算してみましょう。

変数が M 個あったとします。変数番号が $1, 2, 3...M$ ととすると，組み合わせは 1 と 2, 1 と 3.... となります。このとき i と i は分散，i と j が共分散になりますが，s_{ij} は s_{ji} と同じですから，$M \times (M-1)/2$ 個の共分散と M 個の分散が，このデータセットから得られる情報のすべてということになります。合計で $M \times (M+1)/2$ 個になりますね。このように組み合わせまで考えると，変数の数がひとつ増えるだけでも得られる情報，分析のヒントがぐっと増えることになります。多変量解析は，こうした変数間関係の情報を使って分析を進めていくことになります。

ちなみに，変数間関係を表す方法は共分散や相関係数だけではありません。これらはあくまでも変数間の直線的な関係を表す指標であることにも注意が必要です。多変量解析全体としては，変数同士の数字のズレをたしあわせた**距離 (distance)** や，ある変数が他の変数と同時に出現した回数をカウントした共頻度などを扱うモデルもあります。

1.5 課題

■尺度水準 4 つの尺度水準の具体例をそれぞれ挙げなさい。

■平均と分散 平均，分散，標準偏差，標準得点の定義を式で表現しなさい。

■共分散と相関係数 共分散の定義式を書きなさい。そのとき，スコアが標準化されていると相関係数になることを示しなさい。

第 2 章

心理尺度を作る

2.1 はじめに

心理学の研究法は調査，実験，観察の 3 つが代表的なものです。とくに調査法は，質問紙を用いて多くの人に回答を依頼し，それを統計的に分析することで研究仮説を確認しようとするものです。調査対象者をどのように集めるか，調査票をどのようにデザインするか，得られたデータをどのように分析するか，といったことでそれぞれ 90 分以上話せるぐらいに色々考えるべきことはありますが，ここではとくにその本質について考えてみましょう。すなわち，なぜ紙で用紙に丸をつけたものを集めただけで，心の何かがわかったと思えるのか，ということです。

アンケート調査は一見すると，なんのひねりもない研究法に思えます。すなわち，聞きたいことを当の本人に聞いてみる，これだけです。しかも答えやすいように (集計しやすいように？) 紙に問題が書いてあって，該当するところに丸をつけるだけでよかったりするのです。この手の研究は少なくとも 100 人，できれば数百人，大きい規模だと千以上の桁数の協力者に回答を求めます。それを PC に入力して集計するわけです[*1]。しかし「そう思う」を 5 点，「ややそう思う」を 4 点，以下 3，2，1 点･･･と入力するのはなぜでしょう。これらの反応は**順序尺度水準**でしかない，と言われますが，それを間隔尺度水準であると「みなし」て，平均値を求めたりします。なんでそんなことが許されるのでしょう。

本稿ではその謎について迫っていきたいと思います。

2.1.1 測ろうとしているもの；態度

調査票で質問する内容は，(お客様アンケートとかではなく) 心理学の研究であれば**尺度 (scale)** を使います。尺度とは，測定したい対象に数字を割り振るルールのことであり，心理尺度は項目とその採点方法が規格化されたものを指します。心理尺度の測ろうとしているものの多くは**態度 (Attitude)** と呼ばれるものです。これは社会心理学の用語で，簡単に定義すると態度とは「行動の準備状態」のことです[*2]。

次の図 2.1 は，様々な心理学領域を「変化のしやすさ」という次元で並べてみたものです。左端が変化しにくく，右端が変化しやすいものです。人間の生物的･生理的な特徴である気質に起因する心理学的傾向は，生得的だったり非可逆的だったりするので，非常に変化しにくいものと考えられます。次の動機づけは，生き物としての生存本能，生きようとする動因，ぐらいに思ってください。それに影響される形で感情的な要素があります。人の感情とは生理的な反応に裏付けされつつ，認知的なラベリングを受けて，「この胸のドキドキは，恋？」のように考えられるものでしょう。その次にあるのはかなり一貫的した行動の傾向，いわゆる性格と

[*1] もっとも最近では，web で調査の回答を得ることで，入力や誤入力のチェックなどの手間が大幅に軽減されています。

[*2] この定義は Allport (1967) によるものです。

か人格などと呼ばれるものです。その外側に，普段の意識的な活動である認知・情報処理機構があります。それらに裏付けられて形成されるのが態度だと言えるでしょう。そしてその態度までを含めて太い線で囲まれているのが，人間の内界です。この太い線が物理的な人物の境界だと考えてもらってもいいでしょう。外からは見えない心のうちを，この鱗状の図で表現しています。そしてその教会の外にあって，客観的に観測可能なものが行動です。徹底的な行動主義者は，この外に見えるものだけを研究の対象とし，内側の仮説的構成物を用いずに説明しようとするかもしれません。しかし一般的に心理学者が考えたいのは，この内側の心の要素たちについてでしょう。ここでおさえておいて欲しいのは，心理尺度で測定しようとしている態度の位置関係であり，行動のすぐ手前にあるもの，変わりやすいものであるという点です。

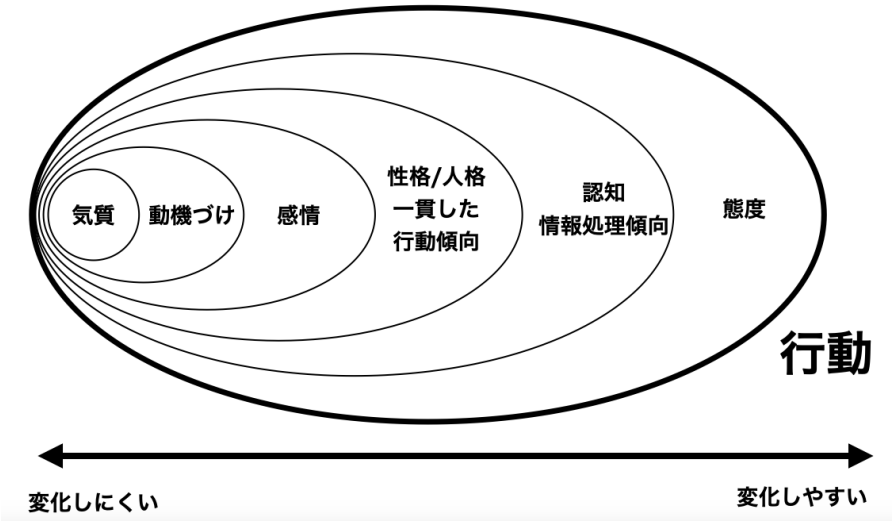

図 2.1　心理学の研究領域

社会心理学も心理学ですから，研究は基本的に観察可能で客観的なものを対象にします。社会的な行動をテーマにするというのがもちろんですが，社会的な行動というのは状況によって変わるものですし，研究したい状況を待っていても自動的に出てくるものではありません。もちろん実験室などでその状況を作り出すこともやりますが，選挙のような公的なものであれば実験的に作り出すこともできません。しかしたとえば選挙結果などは，社会心理学における政治的行動に大きな影響を及ぼす (あるいは政治的行動の結果になる) ものですから，研究としては非常に重要なテーマになります。選挙になるまでは仕事ができない，というのでは社会心理学者も困りますから，「あなたはどこに投票するつもりですか」と聞いて普段の政治的な態度を調べるという研究手法ができました。社会調査の始まりです。

そしてこれを心理学一般に応用しているのが，質問紙調査です。心理学の研究テーマは行動ですが，行動を作り出せない場合や普段の状態を査定したいときに，「調査票に回答を求める」という方法を取るのです。もちろん調査に対する回答は，嘘や見栄，間違いや勘違いなども入り込む可能性がありますので，設計は丁寧に，分析は慎重に行う必要があります。調査票で過去のことを聞いたりすると記憶が歪んでいる可能性がありますし，未来のことを聞いたら嘘八百を答えられるかもしれません。今の気持ちを聞いても，自分の現在の状況がはっきりわかっているかどうか，怪しいものです。それでもある程度の真実味はあるだろうと考えて，こうした阻害要因を取り除いて本質を掴むために，いろいろな工夫がなされています。

尺度を使って感情や気分，考え方 (認知傾向) や今後どう行動するつもりか (行動意図) を調べることがなされていますが，本質的にはこれらすべてが，広い意味での態度です。態度の認知的側面，感情的側面，行動的側面などと言われたりします (藤原, 2001)。この「態度」の説明や定義は，実はそれほど明確ではありま

せん。しかし，一般に特定の対象に対する，正負・量的な評価が可能なものと考えられています。たとえば「自民党に対する態度」というのは，自民党という対象に対して，「好き」「嫌い」というポジティブ・ネガティブの評価ができ，さらに「とても好き」「やや嫌い」のように量的に表現できるものでもある，と仮定されます。多くの人に多くの項目で調査するのは，項目同士の誤差が相殺しあってこれが正規分布すると考えられるからであり，社会的な態度は極端な値が少なく平均的な値を持つひとが多いものとして，相対的に評価されます。

基本的に測定しようとしているものは，こうした特徴を持ったなにかである，ということをまずは踏まえておきたいと思います[*3]。

2.1.2 3 つの方法

心理尺度の作り方には大きく分けて 3 つのスタイルがあります。

1 つ目はサーストン法による尺度で，**等現間隔法 (method of equal-appearing intervals)** と呼ばれるものです。2 つ目はリッカート法 (Likert 法，ライカートと読む人もいる) と呼ばれるもので，5 件法，7 件法など数段階のカテゴリラベルのもっとも近いところに丸をつけるという方法です。3 つ目は SD(Semantic Differential 法，意味微分法と訳されることも) 法とよばれるものです。SD 法は態度測定というより，イメージの測定を目的としたもので，対象を提示しつつペアになった形容詞を列挙して提示します。たとえば「専修大学」という対象に対して，「激しい-落ち着いた」「慎重な-軽快な」といった形容詞対ではどちらの表現が近いかを評定してもらいます。形容詞の対が作る軸上でいうと平均的にどのあたりに対象がプロットされるのか，を見ることで対象ごとのイメージの違いを表現するのが基本的なアイデアです。SD 法は複数の対象に対するイメージの相対的比較ですから，スコアの点数化にはそれほど重きを置いていないのでここでは取り上げません。

サーストン法とリッカート法は，これを使って態度のスコアをつけることができます。小杉の自民党に対する態度は 4.8 点だ，といったように，です。こうした数値化がどのような理屈でなされるのかを，今からみていこうと思います。

2.2 サーストンの等現間隔法

サーストンの等現間隔法(とうげんかんかくほう)は，社会的な態度について絶対評価を与える方法です。この方法で作成された尺度は，各項目 (態度表明文と呼ばれます) に尺度値がついており，回答者は提示された項目に賛成であればその尺度値がその人の態度得点になります。この尺度値は事前に複数の評定者によって決めておく必要があります。すなわち，尺度を作る前の入念な準備が必要です。また，サーストンの尺度は 1 次元性を有していることが前提となります。

具体的な作成方法は次のような手順で行います。

1. 項目の収集
2. 評定者集団による評定
3. 尺度値の算出
4. 項目の選定

以下順に説明します。

[*3] 性格のように特定の対象を持たないものであっても，正規分布に従うと考えるのは自然ですから，この後説明する統計技法が適用できるものもあります。感情や気分といったものは，持続時間が短いので生理指標などで測定するべきであり，調査票によるアプローチは不向きですが，逆に言えばある程度一定の安定した心理状態であれば測定することができると考えられているのかもしれません。

■項目の収集　まずは測定したい社会的態度のテーマに沿って，項目を準備します。たとえば「自民党に対する態度」のように，誰でも思い描ける具体的な対象が良いでしょう。このようなテーマが決まれば，これに対する態度項目を色々考えます。「自民党のことを考えると夜も眠れない」とか「自民党に関係したニュースはなるべく見るようにしている」「近所の自民党員の事務所に行くことがある」というポジティブな態度もあるでしょうし，「自民党のニュースはなるべく聞きたくない」「自民党には投票しない」「自民党は不正まみれの悪い政党である」といったネガティブな態度もあるでしょう。こうした文言をなるべく多く，強い態度から弱い態度まで，ポジティブなものからネガティブなものまで網羅的に準備します。ニュートラルな項目も考えておく必要があります。

■評定者集団による評定　尺度値を決めるための事前準備です。まず評定者を無作為に集めます。少なくとも十数人は必要でしょう。評定者には事前に準備した項目が好意的－非好意的 (または肯定的－否定的) の 1 次元にそって，7～11 段階ぐらいの多段階に分類してもらいます。「自民党のことを考えると夜も眠れない」というのは非常にポジティブなので 11 点，「自民党には投票しない」というのはかなりネガティブなので 2 点，といったようにです。

■尺度値の算出　このように各態度表明文を複数人で評価してもらいますから，その項目の平均値，中央値，分散などの記述統計量を計算できます。この中央値 (あるいは平均値) をその項目の尺度値とします。ただし，ここで分散が大きい項目は，評定者によって評定の仕方がバラバラだということを意味しますよね。値が人によって定まらないというのは，その項目が刺激としてあまり好ましくないと考えられるので，項目候補から削除します。誰がみても 10 点とか誰がみても 3 点，といった分散が少ない項目が望ましいでしょう。

■項目の選出　さてこうして尺度値が計算できたら，それを順に並べていきます。「夜も眠れない」は 10.7 点，「事務所に行くことがある」は 9.5 点，「ニュースをなるべく見る」は 7.9 点･･･というようにしていくことができますね[*4]。このとき，項目間の間隔が均等になるように項目を選別します。たとえば「夜も眠れない」と「事務所に行くことがある」の間隔は 1.2 点ですが，「事務所に行くことがある」と「ニュースをなるべく見る」の間隔は 1.6 点になっているとしましょう。これでは等間隔と言えません。なので，1.2 点間隔すなわち 8.3 点ぐらいの項目を選び出します。

このことからわかるように，サーストン法で尺度を作る場合は，事前に多くの項目を準備しておかないと「ちょうどいいところの表明文がない」となってしまう恐れがあります。ですから最終的にできる尺度に含まれる項目の，5 倍から 10 倍ぐらいを事前に準備し，うまく等間隔に項目が選出できるようにしなければなりません。

なぜ等間隔に選ぶのかというと，もうお分かりですね，これで得られる尺度値を**間隔尺度水準**として扱いたいからです。間隔が等しくなければ順序尺度にしかなりませんが，間隔が等しいことがわかっていると，平均や分散などの計算をし，相対的な比較をできるからです。またこの尺度を使うときは，すでに評定者集団によって尺度値がわかっていますから，回答者にずらりと並べられた尺度を見てもっとも自分の意見に近い項目を選出してもらえば，その項目の尺度値がその人の態度得点だということができます。

評定者集団をなるべく偏りなく多く集めることで，事前に尺度の値を確定させておき，あとは本来研究対象にしたかった人にその尺度を当てれば尺度値 (尺度得点) が求められる方法ですから，準備が大変だけど使うときは確実で絶対的なスコアを与えることができるというのがこの方法の利点です。欠点はその準備コストの高さと，1 次元的な態度しか用いられないことでしょうか。また尺度構成の観点から重要なのは，選出プロ

[*4] 中央値なのになぜ小数点をもつ値があるのだ，と思う人がいるかもしれません。鋭い！これは中央値の計算の仕方によるもので，評定者が偶数人の場合は中央値も両得点の平均や重みつき平均とすることがあるので，実数になることがあるのです。また，このスコアは平均値でもいいかもしれません。

セスによって項目間の尺度値が均等であることが保証されている点です。均等に選んだ後で，1，2，3，4，5 と数字を付け直しても構いません。大事なのは，こうしたプロセスのおかげで間隔尺度水準が維持され，以後の分析に耐えうるスコアになっているという点です。

2.3 リッカートのシグマ法

次に紹介するのはリッカートの**シグマ法 (σ method)** です。これはいわゆる 5 件法，7 件法と呼ばれる採点方法で，「私は自民党の政治のやり方が好きだ」といった項目に対して，「まったく当てはまる」「かなり当てはまる」「やや当てはまる」「どちらとも言えない」「やや当てはまらない」「あまり当てはまらない」「まったく当てはまらない」といった順序づけられたカテゴリーにたいしてもっとも自分の考え・態度と近いところに丸をする，という方法で反応が得られます。この時の反応カテゴリーが，今回は 7 つありますから 7 件法 (7-points scale) で回答を求めた，などと言います。5 段階なら 5 件法，4 段階なら 4 件法です。普通は「どちらとも言えない」というところを用意するために奇数 (3,5,7,9) 件法を使いますが，日本人は「どちらとも言えない」を選びやすいという中庸傾向があるとも言われていますので，意見をはっきりさせるために 4,6 件法も使われたりします。

項目はこれも複数あって，たくさん集められた項目を分析するために，もっとも当てはまるを 7，かなり当てはまるを 6，以下同様にしてまったく当てはまらないを 1，とコード化し分析するのが一般的です。ただし注意して欲しいのは，もっとも当てはまる＝ 7 としたのは**名義尺度水準**の数字の割り当て方と一緒で，このカテゴリーが 7 という尺度値を持っているわけではない点です。そもそもこの評定カテゴリーは，統計学的にはせいぜい順序尺度水準の性質しか持っていませんから (もっとも＞かなり＞やや)，カテゴリーに割り当てた数字からそのまま平均や分散の計算をするのはおかしいはずなのです。もしこれらのカテゴリーの間隔が等しかったとしても，3，4 段階しかないようであればやはり間隔尺度水準の計算ができるほどの精度は持っていません。数量的に分析するには (等間隔が担保された上で)9 から 11 段階は必要と言われています。

それではリッカート法ではどのようにして尺度値を決めるのでしょうか。リッカート法も測定しようとしているのは態度であって，表に出てくる反応カテゴリーの背後には連続的な心理的態度というのがある，と仮定しています。またこの (社会) 心理学的態度は，向きと大きさがあって正規分布を仮定できます。リッカート法も正規分布に従う潜在的な連続変数があると仮定するのです。

さて，ある項目について，多くの人からデータを集めて「もっとも当てはまる」「かなり当てはまる」といったカテゴリーごとの集計ができたとしましょう。多くの人の態度も集積すれば正規分布に従いますから，きっとこのヒストグラムも正規分布を反映したものになっているはずです。しかし我々が知りたいのはその背後にある連続体上の数字なわけです。

ここで図 2.2 を見てください。上段にあるのがある項目のヒストグラムの例です。しかし知りたいのは，下段にあるような正規分布の形をした連続体の変数のはずです。上段のヒストグラムは下段の状態を反映しているはずですから，上段のカテゴリの相対頻度を元に，下段の正規分布を分割します。具体的な数字との対応は表 2.1 を見てください。出現度数を相対頻度にし，正規分布の面積を順に分割していくことになります。カテゴリの下の方から順に分割するということで，表 2.1 の三段目には累積 (相対) 頻度を書いてあります。そしてこれを使って，標準正規分布の下から面積を考えます。統計環境 R では，`qnorm` 関数をつかうと累積確率の確率点が求められるのでした (表 2.1 の 4 段目)。またその時の確率密度も求めてあります (表の 5 段目。R では `dnorm` 関数を使って求めます)。

さて，ではここからどのようにして尺度値を求めればいいでしょうか。一般に C 件法で，下から $1, 2, 3, \cdots c \cdots, C$ とカテゴリ順に数字を割り振ったとして，第 c カテゴリの尺度値 Z_c を考えるとします。このカテゴリ c は標準正規分布において上限 z_c，下限 z_{c-1} の確率点で挟まれる領域としていますから，

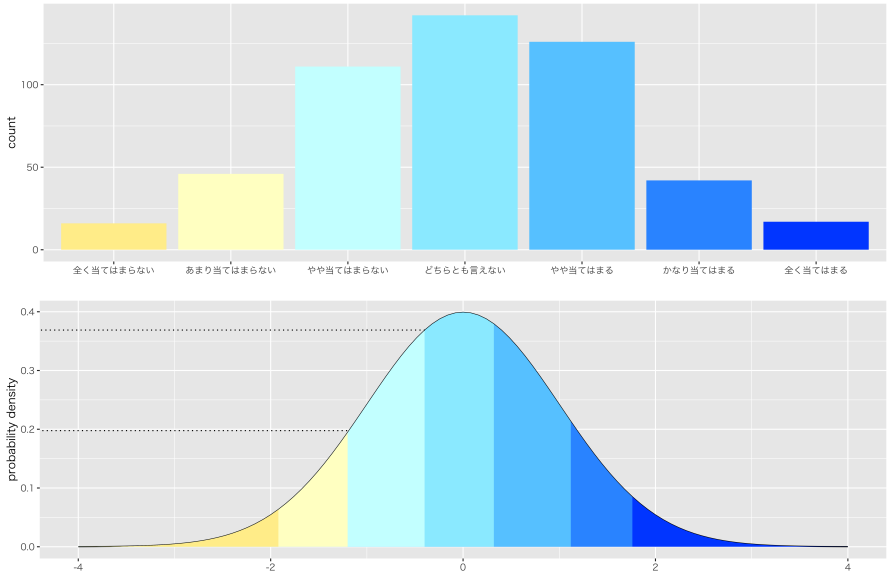

図 2.2 カテゴリ反応と背後の連続値

表 2.1 カテゴリと数値の対応

反応カテゴリ	まったく当てはまらない	あまり当てはまらない	やや当てはまらない	どちらとも言えない	やや当てはまる	かなり当てはまる	まったく当てはまる
出現度数	16	46	111	142	126	42	17
相対度数	0.03	0.09	0.22	0.28	0.25	0.08	0.03
累積相対度数	0.03	0.12	0.35	0.63	0.88	0.97	1.00
累積相対度数の確率点	-1.85	-1.16	-0.40	0.33	1.19	1.83	∞
累積相対度数の確率密度	0.07	0.20	0.37	0.38	0.20	0.08	0.00
付与される尺度値	-2.33	-1.44	-0.77	-0.04	0.72	1.50	2.67

この幅 ($[z_{c-1}, z_c]$) の平均を取ることを考えます。この点は，次の式で求められます (証明は付録 B,Pp.177 参照)。

$$Z_c = \frac{(y_{z_{c-1}} - y_{z_c})}{p_c}$$

ここで y_c は z_c, z_{c-1} における確率密度，p_c はカテゴリ c の相対頻度です。具体例でいきましょう。表 2.1 の数字を使うと，「やや当てはまらない」($c = 3$) の尺度値は，$\frac{0.20 - (0.37)}{0.22} = -0.7727273$ となります (分子は図 2.2 の点線部の差分，分母は該当箇所の面積になります)[*5]。このようにして計算された尺度値が，表 2.1 の一番下の段にある数字です。

[*5] 表 2.1 は小数点下 2 桁までに丸めているので，正確な値ではありません。

このように，累積度数をつかって尺度値を決めるリッカートの方法を**シグマ法**と言います。こうして作られた尺度値は連続体上の数字ですから間隔尺度水準になり，平均や分散をはじめとした数値計算に耐えうる値になっています。機械的に「まったく当てはまらない」から「まったく当てはまる」まで，1.0 刻みで数字を割り振っているのではないのです！

・・・と言いたいところですが，今回の尺度値を眺めてみるとそこそこ等間隔 (間隔は 0.8 ぐらいでしょうか) に並んでいますね。試しに各尺度値を 0.8 で割ってみると，$-2.91, -1.81, -0.97, -0.04, 0.90, 1.88, 3.33$ となります。四捨五入して小数点をなくしてみると，$-3, -2, -1, 0, 1, 2, 3$ となりますね。そう，つまり**非常にラフな近似値でよければ，機械的に 1,2,3... と数字を振っても同じことになります**。ですから，実際の研究ではとくに深く考えずに 1,2,3... と割り振った数字をそのまま使われたりするのです。大山鳴動して鼠一匹といいますか，苦労した割に得るものがないじゃないか，とお叱りを受けそうですが，少なくとも「なぜリッカート法は順序尺度ではなく間隔尺度のように扱って良いのか」という問いには答えられると思います。また，ここにくるまでに，態度の 1 次元性や正規分布の仮定などが含まれていたことを改めて思い出してください。今回は数値例ですので，綺麗に七段階に分かれるようなものを用意しましたが，実際の調査では正規分布しないものや，平均が低すぎるとか高すぎるものが結構みられます。それらに対して機械的につけた数字で分析するのは決して適切な方法ではなく，シグマ法を始めその他の手法で適切な尺度値を付与すべきなのですが，人間は易きに流れるもので**ほとんど考慮されていないのが現状です**[*6]。

2.4 心理尺度の限界

2.4.1 Elephant in the room

質問紙による測定は一見，簡単なものに見えます。「わからないなら，聞けばいいじゃない」という軽い気持ちでデータを集めてもいいじゃないか，と思われるかもしれません。しかし，聞いても本当の答えが返ってくるかどうかわからないので，心理学では何をデータにするかについて 100 年以上の労力をかけてきたのです。人間は，嘘をつくし，間違えるし，易きに流れる生き物です。あるいは何かの指標を目標にすると，それは必ずハックされます。たとえば感染症対策の不備を指摘されるのに困るようであれば，検査回数を減らしてしまえばいいのです。大学進学率，就職率が高く評価されるのであれば，進学や就職ができそうにない人を「希望者ではない」と分母から除外してやることで改善できます。学生が授業にどれぐらい積極的に参加しているかを評価したいのであれば，椅子に対する前傾の角度や手を上げた回数を数えてもいいですし，椅子を取り除いて演習にしてしまえばみんな立ち上がって議論していることになります。これは半分冗談のようですが，半分は笑えない側面があります。繰り返しますが，安易な指標化はハックされ，実質的な意味がないことになってしまうのです[*7]。

心理尺度が心理学のデータになっているのであれば，その妥当性について目をつぶることはできないはずです。*Elephant in the room* という慣用句にあるように[*8]，問題の存在に気づいていながら見て見ぬ振りをするのは，やはり適切な態度とはいえません。

たとえば 2021 年の心理学研究 92 巻から，「尺度」や「質問項目」など言葉で測定している研究をピック

[*6] この状況は決して良いものではなく，悪しき研究上の風習だと思われます。幸い，第 4 講で説明する**項目反応理論 (Item Response Theory)** の一種，**段階反応モデル (Greaded Response Model)** を用いると，この問題点をカヴァーしつつ有用な情報が得られますので，みなさんは一足飛びにその手法を身につけた方が良いと思います。

[*7] このことについては Muller・Muller (2018 松本訳 2019) という本に詳しく書かれているので，ぜひ手に取ってみてください。

[*8] これは，明らかに存在しているが誰もがそれについて話すのを避けている，無視されている，または避けられている明白な問題やリスクという意味です。象が部屋の中にいるのに，(それについて議論しても無駄だから) いないことにする，ということですね。心理尺度の問題も，これだけ使われてしまうと今更「それ何測ってるの？根拠は？」と聞けなくなってしまっているのでしょう。

アップしてみると，33 本の論文 (資料，特集含む) が該当します。そこでは 81 件の言葉による測定がなされており，測定されているのは 184 の概念や因子です。実験による反応だけ，あるいは逐語録のような「語り」だけをデータとしている研究は少なく，また人口動態のような集計されたデータだけで議論される研究もほとんどありません[*9]。つまり，心理学の研究はほとんど個人の言語刺激に対する反応，それを集計したものから構成されているのです。またこうした言語反応は，自記式であるものがほとんどで，他記式の研究は 1 件永谷他 (2022) だけでした。すなわち，現代心理学の研究とは「本人に言語で解答してもらう」ものになっています。

これを見るとまず，あまりにも多くの側面について測られていることに驚かれると思います。1 つの学問領域としてまとまっているのが不思議なほどで，同じ研究テーマを持って集まった同人誌 (機関誌) でとは思えないかもしれません。また，それぞれの論文を参照すればわかりますが，反応カテゴリについては「1. あてはまらないから 5. あてはまるの 5 件法」などと表記されるのがほとんどで，2,3,4 など中間点にどのようなカテゴリが付与されているか，細かく言及してある研究もありません[*10]。各カテゴリについての言及がないということは，カテゴリに対する反応ではなく 1,2,3,4,5 という連続体を想定していることと考えられます。リッカート式のやり方である以上，態度についての連続体に正規分布を仮定した数値化をすることが基本原則ですが，分布の正規性，尺度値の等間隔性という仮定に言及するまでもなく前提として利用しているように見えます。心理尺度による測定値が，そこまでの精度をもたないものとして考えられているのかもしれません。

2.4.2 測っているものは何か

実際の使われ方を見た上で，どのようなものが測定されたと考えられているのか，改めて確認してみたいと思います。内容をいくつかに分類したのが表 2.2 です。

これらを見るだけでも，既にサーストン法やリッカート法が誕生した経緯である，「態度対象に対する意見文についての本人の位置付けの評定」という枠組みは大きく飛び越えていることは明らかです。これを 2 つの次元で分類してみましょう。

- 個人の主観的経験世界，自己の内部についての言及と，他者や事象・現象についての言及
- 判断基準の所在が自分の内部におくもの (内部状態の参照。「そう思う」「あてはまる」など) と，客観的なもの (頻度や程度の報告，社会的価値基準で評価できるもの。「毎日ある」，「全く問題にならない」，「経済的に困っている」など)

その上で，各領域について何を測定しているか，測定上の問題があるとすればどのようなものかについて考えてみたいと思います。

■内部についての言及を，内的基準で評価する場合 自分の意図や信念，感情を評定させるケースがここに当たります。当然のことながら，本人の主観的な経験，感覚を，本人の主観的な語感にもとづいて報告させるわけですから，その判断の正当性に疑問を挟むことができます。その判断は本当に正しいのでしょうか。自分のことは自分が一番よくわかっている，などと言いますが，本当にそうなのでしょうか。

仮に自分の意識では疑いないことだ，という表明がされたとしても，その人が嘘をついてないといえるでしょうか。嘘とは言わなくても少し表現を誇張したり，社会的望ましさを考えて表現を変えてないといえるでしょうか。さらにいえば，真実でないことを自覚しているかどうかという問題もあります。社会的状況にプライミングされている場合は気づきませんし，お酒を飲んでいる時に「自分は酔っていない」と主張しても誰も信じ

[*9] 92 巻 5 号は「新型コロナウイルス感染症と心理学」を表題とした特集号であり，この号は語りや集計データが利用される傾向がありました。

[*10] これは論文の紙幅制限によるところも大きいかもしれません。昨今は電子的な付録を Web などで公開する取り組みもあります。

表 2.2 測っているものの分類

自他の区分	基準の所在	分類名	項目例
自分	内的	自分の意図，信念，自己評価，願望	今の自分が客観的にどう見えるか知りたい
自分	内的	自分の今の感情	怒りを感じる
自分	内的	自身の行為の理由	できないことができるようになるとうれしいから
自分	外的	自分の普段の振る舞いについての自己報告	厳しく命令したり注意したりする
自分	外的	近況など経験に基づく自身の状態	風邪やインフルエンザなどにとても感染しやすい
他者	内的	他者に対する評価	黒い衛生マスクを着用する人物についてどう感じますか
他者	内的	推測に基づく自他の行為の予測	(あなたが詐欺電話について相談した場合，配偶者は) 電話を詐欺だと見抜くことができると思いますか
他者	内的	事象・現象に対する評価や理由づけ	音楽は友人たちとの話題にできるものだから
他者	外的	事実の言語報告	両親はよく喧嘩をする

てくれないように，本人の意識的な報告が誤っているということは当然あり得るわけです。本人は至って真剣で (もちろん素面で)，真実であるという自覚があってもです。

もちろんそうした間違いが生じないように，方法論上の工夫はいろいろ考えられています。類似の文言で多角的にアプローチしたり，虚偽報告が含まれないよう引っかけ問題を用意することもあります。三浦・小林 (2015) ではとくに Web 調査での手抜き回答者，Satisficer を検出する研究なども行っています。そのほか Lie 項目，フィラー項目などを入れるといった調査上のテクニックなどもかんがえられるでしょう。

しかし虚偽報告を除外したとしても，さらに大きな哲学的問題が残ります。すなわち，その人が自らの感覚を報告する場合，その目盛が正しいかどうかをどのように判断すれば良いのでしょうか。ある人にとって「非常に嬉しい」という感覚と，別の人にとって「やや嬉しい」といった感覚，この「非常に」と「やや」の大小を比較することはいかにして可能なのでしょうか。病院臨床の現場では，「最も痛かった時を 10，痛みがない時を 0 として，今の痛みはどの程度ですか」と言語報告を求めることがあります。これは Numerical Rating Scale(NRS) と呼ばれ，実際に筆者も骨折で入院した時に毎朝の健診で質問されました。ところが，入院初日に「今の痛みはどの程度ですか」と聞かれ，「5 点ぐらいでしょうか?」と回答したところ，看護婦さんに「そんなに?!」と驚かれたため，慌てて「あ，3 ぐらいです，たぶん」と言い直した経験があります。確かに上限，下限があればある程度の相対比較は可能で，他者とスケーリングを合わせることもできますが，これについても多分に**言語という共通の意味空間**に依存したスコアであることは間違いありません。言語の意味が互いに共有できているからこそ，質問と回答や合算ができることに辛うじて根拠があたえられるのであり，「集計した累積度数から算出した相対的位置」ほどの精度を期待することはほぼ不可能でしょう。

■内部についての言及を，外的基準で評価する場合 自分が普段ある振る舞いを「どの程度頻繁に」，「どの程度の強度で」行っているかを自己報告させて測度としているのがこの場合です。

もちろん自己報告ですから，嘘をついていないか，社会的望ましさの影響はどうか，嘘をついていることの自覚はあるかといった問題が含まれるのは先ほどと同様です。また，今現在その行為・行動をおこなっているのでなければ記憶に頼ることになりますが，記憶が正しいかどうかについても考えなければなりません。過去を振り返って評価する方法には回顧法などと名称はついていますが，過去の経験を思い出させることにどれほどの信憑性があるのかについては，記憶の研究を引用するまでもないでしょう。思い出したくない，思い出したところで人に言いたくない，ということも少なくないので，その数値の信憑性はかなり低いと見積もっておいた方が良いのではないでしょうか。

またこうした頻度，程度の多くは順序尺度水準であると考えられます。大小関係は維持されているといえるでしょうが，加算したり平均を出したりすることはできませんので，研究報告をする上では注意が必要です。

■他者や事象についての言及を，内的基準で評価する場合　他者に対する評価，事象・現象に対する評価や理由づけは，社会的態度を測定している領域といえそうです。社会的に実在性が共有されているものに対して意見を表明すること，その意見は心理的状態を反映している，少なくとも「言質が取れた」ものとして考えることができます。そうした意見の分布は，様々な要因からの影響が想定されるので，正規分布していると仮定することもできるでしょう。こうした分布が確認できるのであれば，評定者集団をもって事前に数値化する基準を与えた尺度を作成し，尺度値として数値化することもできます。また正規分布が仮定できるのであれば，集計して累積的な分布を見るとある平均値をもって左右対称に分布していることが確認できるはずであり，正規分布の確率密度を相対頻度で分割することで各意見項目の核反応カテゴリに数値を付与することもでき，個々人の態度得点として数値化できます。前者はサーストン法，後者はリッカート法という尺度作成法であり，リッカート法での尺度はサーストン法での尺度と高い相関を示すことがわかっているのでした (Likert, 1932)。

ですから，リッカート法をつかって測定することができる，といっても良いでしょう。ただし，その数値化が非常に簡便なものになっていることに注意が必要です。本当にその集計値は正規分布の形をしているでしょうか。態度理論の仮定を満たしたものになっているでしょうか。3 件法，5 件法，7 件法など，さまざまな運用形態がありますが，そのこと自体は問題ではありません。問題はその幅の中で正しく分布できているかどうかであり，平均点が高すぎる (**天井効果**) とか，低すぎる (**床効果**) といった問題があれば，下から順に 1,2,3... と数値化することの正当性が担保されません。図 2.3 には左右対称等間隔なスコアが正当化される事例 (左上) も示しましたが，天井効果 (左上)，床効果 (右下) が見られる場合，あるいは正規分布でなく二峰性 (右下) があるばあいの，シグマ法による尺度化例を示しました。こうした分布の状況を顧みず，機械的に数値化することに問題がないといえるはずもありません。

天井効果，床効果，あるいは正規分布とは思えないような分布をしている場合，統計モデルを駆使して尺度値を補正することが可能です。切断正規分布や複数の確率分布の組み合わせなど，モデリングによって補正するアイデアが考えられますが，少なくとも今回参照した文献の中でそうした補正は行われていません。測定された数値がそれでいいのかどうか，あらためて計算手順を考え直す必要がありそうです。

また言葉で指示した他者・事象・現象が同一のものであるかどうかについて，妥当性あるいは言語という共通基盤の頑健性について考えておく必要があります。社会的態度の研究の場合，たとえば政治的態度を研究するのことを考えたとしましょう。「自民党についての態度」を測定する場合，「自民党について」の意見が自分にどの程度当てはまるかを回答者個々人が判断することになりますが，このとき全員の頭の中に想定された「自民党」は同じ対象でなければなりません。日本における一般的な社会人であれば，「自民党」といえばあの自由民主党のこと，と誰でも同じものを思い浮かべられるでしょう。ここでアメリカの民主党を考えたり，イギリスの労働党を考えながら回答しているような人がいれば，回答パターンは無茶苦茶なものになってしまうでしょう。

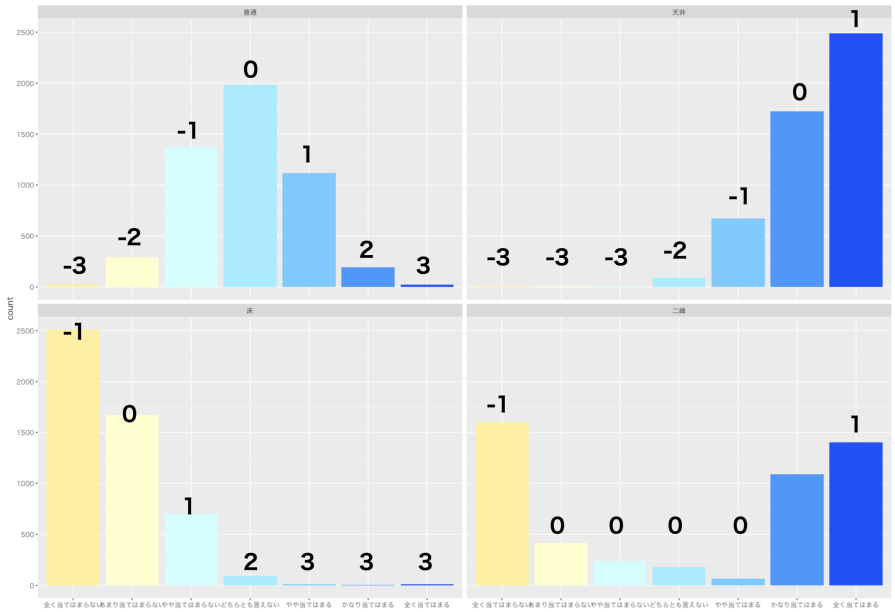

図 2.3 天井効果・床効果の見られた尺度

しかし評価してもらう対象が「配偶者」「教師」「音楽」といったものであればどうでしょうか。誰 1 人として同じ配偶者を持っている人はいませんから，ある人のパートナーに対する態度と，別の人のパートナーに対する態度が同じもの (並べて，集計して，正規分布するもの) といえるでしょうか。「世話になった教師」といえば，ある時代のある学校においては「あの先生！」と特定できるかもしれませんが，それでも「自分は別の先生に世話になった」という人もいるでしょうし，時代と場所が変われば当然同じ人ではなくなります。「好きな音楽について」であっても，ロック，テクノ，K-Pop などなど，同じ「音楽」という括りで表現して良いものでしょうか。雅楽を想定している人と，パンクロックを想定している人が，「音楽とは穏やかな気持ちにさせてくれる」という意見に同じ次元で反応しているといえるでしょうか。

この批判はややもすると，言いがかりに聞こえるかもしれません。「配偶者」「教師」という言葉で表される典型的なパターンを研究したい，つまり言葉の使われ方の研究であれば問題ないかもしれません。あるいは社会通念上共通する架空の典型例，すなわち「あるある」ネタ研究であるなら，文化的な研究として価値があることかもしれません。しかし一人ひとりの主観的経験や心情を測定するというのとは，少し次元が違ってきているようです。

社会的態度は，個々人に数値を付与して個人間で相対的に比較することを可能にしてくれます。しかしそのことと，「その人が社会的態度を有している」かどうかについては，別の観点から考える必要もありそうです。このことについてはまた後で触れることになります。

■他者や事象についての言及を，外的基準で評価する場合 この領域は，事実の言語報告ですから，社会調査や実態調査として使われるところです。本人の言語報告に依存することなく，調べて裏取りをすることも可能ではありますが，「聞いた方が早い」から聞いているわけです。例えば家計収入など，本人の前年度の納税証明書を提示してもらえれば，円の単位まで正確に測定できますが，そこまでの精度や手間が必要ではないので「100 万円未満，100～200 万円，200～500 万円･･･」等々の範囲で聞いていったりします。

プライベイトな情報であれば社会的望ましさの影響が，過去の記録を回顧して回答してもらう場合は記憶の歪曲といった問題が考えられますから，正しく事実を記録したいのであれば裏取りをするべきでしょう。言

語による解答はあくまでも近似的，簡便的なスコアでしかないことに留意することが重要です。またこうした尺度で報告されるものは，調査前後の文脈によって大きく影響される可能性がありますから，「このような報告があるから」ということをエビデンスとするにはやや弱いかもしれない，と考えておいた方がいいかもしれません。

2.4.3 心理尺度の限界

見てきたように，リッカート法はサーストン法の簡易版であり，これらは社会的態度を測定するためのものでした。社会的態度には正規分布するという仮定があり，同様の仮定がたてられる性格心理学の領域などであれば，リッカート法のような目盛をつけることでの測定・数値化に根拠があると言えるでしょう。言い換えれば，分布の過程や回答パターンの累積が適切な操作でない場合，数値化の根拠がないとも言えます。リッカート法は 5，7 段階の目盛を作ることではありません。理論的前提が満たされない場合は使用を控えるか，異なる分析方法を考えるべきです。リッカート法で測定した変数は，因子分析法によって分析され，因子を見出して検討するというのがよく見られる手法ですが，実は数値化の根拠がない場合は，この分析方法の適用も間違っています。その分析法が使えないなら，多変量データは無用の長物か，というとそうではなくて，双対尺度法 (西里, 2010) や多次元尺度構成法 (高根, 1980)，クラスター分析 (新納, 2007) など，数値化の理論に適した分析方法を使えば良いのです。これが使われていないのは，心理学者の不勉強としか言いようがないとおもいます。もちろん心理学者の研究の主眼は別にあって，分析法の理解に時間を費やしていられない，という反論があるだろうことは容易に想像できますが，だからと言って間違ったものを間違って分析していたのでは本末転倒ではありませんか。

これほど心理尺度が乱立した理由の一つは，この因子分析法がなんかカッコよかったこと，統計ソフトで誰でも簡単にその分析ができるようになったこと，が一因であるように筆者は考えています。かっこいいから，簡単だからという理由が，科学的な厳密性に依拠しないことはもちろんです。皆さんは周囲が統計ソフトで踊っている間に，しっかりと理論的な根拠と数理的な手続きを身につけ，正しく心理尺度が使えるようになってほしいと思います。

2.5 課題

■リッカート法 シグマ法でなく機械的に数字を割り振るとどのような問題が生じるか，自分で数値例を作って検証してみてください。

■信頼性の記述と報告 心理学の尺度作成に関する論文[*11]を読み，信頼性についてどのように記述されているかを確認してみましょう。

■さまざまな妥当性 妥当性にはさまざまなものがあります。Grimm・Yarnold (2001 小杉他 訳 2016) などを参考に，妥当性について自分なりに調べてみてください。

■リッカートのシグマ法 表 2.3 のように，「かなり当てはまる」や「まったく当てはまる」など尺度の右の方に丸をつける人が多かった項目があったとします。この時の尺度値をリッカートのシグマ法に則って算出してみてください。

[*11] 日本心理学会が出している「心理学研究」という学会誌では，【資料】というカテゴリーで毎回のように新しい尺度が報告されています。

表 2.3　天井効果の出た尺度

反応カテゴリ	まったく当てはまらない	あまり当てはまらない	やや当てはまらない	どちらとも言えない	やや当てはまる	かなり当てはまる	まったく当てはまる
出現度数	1.00	3.00	5.00	18.00	24.00	53.00	56.00
相対度数	0.01	0.02	0.03	0.11	0.15	0.33	0.35
累積相対度数	0.01	0.03	0.06	0.17	0.32	0.65	1.00
累積相対度数の確率点	-2.50	-1.96	-1.59	-0.96	-0.47	0.39	∞
累積相対度数の確率密度	0.02	0.06	0.11	0.25	0.36	0.37	0.00

第 3 章

テスト理論と因子分析

3.1 古典的テスト理論と信頼性

前回は心理尺度の作り方 (とその限界) について説明しました。心理尺度は, それがきちんと測定したいものを測定できているか, 評価する必要があります。それを**信頼性 (reliability)** と**妥当性 (validity)** の 2 つの側面から説明します。

3.1.1 信頼性

信頼性は測定の安定性と言い換えてもいいかもしれません。すなわち, 同じものを 2 回測っても同じ数字がつくことですね。測定するたびに数字が変わるようでは, その測定器 (ここでは尺度ですが) は信用ならない, というわけです。

テスト理論の文脈では, テストのスコア X を本当に測りたいもののスコア t と誤差 e とに分割して考えます。古典的テスト理論のモデル式は次の通りです。

$$X = t + e$$

すなわち, テストの点数 X は真のスコア t と誤差 e に分割できるというものです。ここで, 各項目についても同じことが言えると考えると, $X_i = t_i + e_i$ ということになります。非常に単純なモデルですが, 測定したものには誤差がついているという考え方, 言い換えると目に見えるものだけが真実ではないという考え方がしめされています。この考え方は, ソフトサイエンスの領域においては重要なことです。

またこの古典的テスト理論から, いくつかの重要な考えを読み取ることができます。ひとつは誤差についての考え方です。このモデルを $X_j = t_j + e_j$ のように, ある個人の変化しない属性について, j 回測定したとします。この時, 測定の平均は次のように計算できます[*1]。

[*1] この式は, ある測定を多くの個人 i に対して行ったものとして, $X_i = t_i + e_i$ と考えることもできますが, 添字が異なるだけで式の展開に違いはありません。

$$
\begin{aligned}
\bar{X} &= \frac{1}{n}\sum_{j=1}^{n} X_j \\
&= \frac{1}{n}\sum_{i=j}^{n}(t_j + e_j) && \text{定義より} \\
&= \frac{1}{n}\sum_{j=1}^{n} t_j + \frac{1}{n}\sum_{j=1}^{n} e_j && \text{分配して} \\
&= \bar{t} + \bar{e}
\end{aligned}
$$

さて古典的テスト理論では，誤差に関して次のことが仮定されます。

- 誤差の平均はゼロ。つまり誤差が出現するときは，「正に偏る」「負に偏る」といった一貫した傾向がないと考える。
- 真のスコアと誤差との相関はゼロ。つまり誤差は真のスコアに関係なく影響してくるもので，真のスコアと共変動するようであれば偶然によるものとはいえない。
- 異なる測定誤差間の相関はゼロ。誤差同士がなにか意味のある変動をしているのであれば，それはもう偶然によるものとはいえない。

これらはいずれも，誤差が「偶然によって現れる影響で，制御不可能なもの」という考え方からは自然な仮定だといえるでしょう。より詳しくいえば，誤差は測定に応じて毎回一定の傾向で生じる**系統誤差 (systematic error)** と，全く傾向のつかめない**偶然誤差 (random error)** に分けて考えられますが，系統誤差は測定に際して工夫して取り除くべき問題であり，ここでは全くの偶然による誤差についての議論だからです。

さて誤差の平均がゼロ，つまり $\bar{e} = 0$ ですから，$\bar{X} = \bar{t}$ となって，いつかは誤差がなくなって真のスコアを得ることができるようになる，ということが示されます。

また平均は 0 ですが分散はゼロではありません[*2]。このテストの分散を考えると，次のようなことがわかります。

$$
\begin{aligned}
Var(X) &= \frac{1}{n}\sum_{i=j}^{n}(X_j - \bar{X})^2 && \text{定義より} \\
&= \frac{1}{n}\sum_{j=1}^{n}((t_j + e_j) - (\bar{t} + \bar{e}))^2 && \text{X をテスト理論のモデルに} \\
&= \frac{1}{n}\sum_{j=1}^{n}((t_j - \bar{t}) + (e_j - \bar{e}))^2 && \text{同じ記号でまとめる} \\
&= \frac{1}{n}\sum_{j=1}^{n}((t_j - \bar{t})^2 + 2(t_j - \bar{t})(e_j - \bar{e}) + (e_j - \bar{e})^2) && \text{展開する} \\
&= \frac{1}{n}\sum_{i=1}^{n}(t_j - \bar{t})^2 + \frac{1}{n}\sum_{j=1}^{n}2(t_j - \bar{t})(e_j - \bar{e}) + \frac{1}{n}\sum_{j=1}^{n}(e_j - \bar{e})^2 && \textstyle\sum \text{を分配} \\
&= Var(t) + 2Cov(te) + Var(e)
\end{aligned}
$$

[*2] ガウスの考えた誤差論から，誤差は確率**正規分布 (Gaussian Curve)** に従うと考えられます。

ここで Cov とは共分散を表しています。第二項の $2Cov(te)$ は真のスコアと誤差との共分散 (を 2 倍したもの) を表していることになりますが, 共分散 (相関) がそもそも線形関係を表す指標であったことを思い出してください。相関係数は共分散を標準化したものだったわけですが, そういう意味ではこの $Cov(te)$ というのは真のスコアと誤差との相関関係を表しているようなものです。さて, ここでも誤差の仮定から, 相関はゼロです。すなわち, 誤差とはどのような傾向もなく出現するもの, という考えられているのです。どのような傾向もないわけですから, 当然なにかと相関関係にあるはずがない, すなわち $Cov(te) = 0$ であるとなります。

テストの分散 $Var(X)$ を考えると, テスト全体の分散は $Var(X) = Var(t) + Var(e)$ となり, 真のスコアの分散と誤差の分散に分解できることがわかります。ここから, 信頼性 Rel を次のように表現できます。

$$Rel = \frac{Var(t)}{Var(X)} = \frac{Var(X) - Var(e)}{Var(X)} = 1 - \frac{Var(e)}{Var(X)}$$

言葉で言えば, **信頼性**の定義は「全分散中にしめる真のスコアの分散の割合」ということになります。

信頼性のない尺度があれば, その後のお話は先に進みませんから, まずもって「尺度が信頼できるかどうか」を評価する必要があります。このことを信頼性は妥当性の上限, と表現したりします。信頼性を評価する方法は, 測定値の安定の程度を評価できればいいのですから, 複数の測定を持ってその相関係数を計算することでひとまず達成できます。しかし同じ尺度を何度も使うのは, 調査回答者に要らぬ構えを持たせてしまいますから, 普通は 1 回の尺度を分割してその特徴を見ることにします。尺度全体から計算される回答者の値は, 項目の尺度値の合計であるのが普通です (テストの点数も正答数に対応していますね)。ですから, ある項目 j の尺度値は, 各尺度値の和と高い相関をするはずです。このように, 各項目が尺度全体の値とどの程度相関しているかを見る **IT 相関 (Item-Total correlations)** は, 尺度の信頼性を見る 1 つの指標になります[*3]。ある項目が, 尺度全体と相関していなければ, それはその項目が尺度の中で目的と違うものを測定している可能性があり, それは必然的に尺度の安定を損ねる結果になるからです。

この考え方を発展させ, 各項目が他の項目とどの程度相関しあっているか, つまり尺度の中でどの程度整合性がとれたもの＝一貫して同じものを測定しているのかを評価する指標として, α **係数 (alpha coefficient)** があります[*4]。これは, テストに含まれる項目数を M, テスト全体の分散を V_t, 項目 j の分散を V_j と表すと, 次の式で表されます。

$$\alpha = \frac{M}{M-1} \times \left(1 - \frac{\sum V_j}{V_t}\right)$$

この指標は, 各項目が他の項目とどれぐらい相関するかを総合的に表した指標で, とくに**内的整合性信頼性**と呼ばれます。このようにして, 尺度の安定の程度である信頼性は数値化できますが, 次にお話しする妥当性については, 数値化できないものです。

3.1.2 妥当性

妥当性 (validity) は, 信頼性をその上限とした上で, それが何を測っているのかを改めて考える指標です。

たとえば身長を測ろうとして, 体重計を使うとします。成長に応じて, 身長が伸びますが, それは体重とも関係がありますので, 身長の伸びに応じて体重も増えていくでしょう。体重計は安定した計測器で, 信頼性は十分あると思いますが, 体重計で身長が測れていると言えるでしょうか。相関する変数ですので, 部分的に Yes といえそうですが, やはり身長は身長計で測ったほうが良いでしょう。身長計の方が, 身長という特徴を的確

[*3] j を除いた残り $M - j$ 個との相関を求めることもあります。

[*4] **クロンバックのアルファ (Cronbach's alpha)** とも呼ばれます。

に捉え，より本質に近い測定をしているからです。このように，作ったものがしっかりとその本質を捉えているかどうか，これが妥当性の基本的なポイントです。

妥当性はですから，そもそも概念としてその測定しようとしているものが適切かどうか (**構成概念妥当性 (construct validity)**) とか，その文言でちゃんと質問できているか (**内容的妥当性 (content validity)**)，理屈通りその測定値が結果と変動しているか (**基準関連妥当性 (criterion validity)**) と言った面から検証されます。最後の基準関連妥当性については，基準値と尺度値をつかって数量的に検証できますが，構成概念妥当性や内容的妥当性は中身の問題であったり，言葉と概念の対応であったりするので，数理モデル的アプローチができるものではありません。数値化できないか大きな問題ではない，というのはもちろん逆で，数値化できないところであるからこそ，専門的な観点，幅広い視野，批判的思考でもって検証していかなければなりません。

量的に評価する方法としては，今後説明していく**因子分析 (Factor Analysis)** によって**因子的妥当性 (factorial validity)** を見る方法ですとか，理論通りの因子に分離できているかを見る**弁別的妥当性 (distinctive validity)**，あるいは**収束的妥当性 (convergent validity)** などがあります。これらをまとめて，**検証的因子分析 (confirmatory factor analysis)** によって理論通りの分類ができているかをモデル適合度の観点から評価する方法もあります。これらは次回以降お話ししていく，テスト理論の発展系のなかで考えていくものになります。

3.2 因子分析モデル

3.2.1 単因子モデル

今からお話しするのは，**因子分析 (Factor Analysis)** というモデルです。因子分析モデルは古典的テスト理論の拡張であり，もっとも簡単な 1 因子モデルは次のように表されます。

$$z_{ij} = a_j f_i + e_{ij}$$

ここで Z_{ij} は個人 i の項目 j に対する反応を標準得点で表したもの[*5]，a_j は項目 j の**因子負荷量 (factor loading)**，f_i は個人 i の**因子得点 (factor score)**，e_{ij} は個人 i と項目 j の組み合わさった時に生じた誤差と呼ばれます。

因子負荷量 (factor loading) とは，因子というこのテストで測定したい特性と，項目との関係の強さを表しているものです。**因子得点 (factor score)** とは，因子というこのテストで測定したい特性と，個人との関係の強さを表しているもので，その人のスコアだということができます。

記号についている添字に注目してください。添字 i は個人を，添字 j は項目を表していますが，因子負荷量は a_j と表されています。つまり項目によって変わる変量です。因子得点は f_i と表されています。つまり人によって変わる変量です。古典的テスト理論をこの添字を使って表現するならば，$X_{ij} = t_i + e_{ij}$ となりますが，これと比べてみると t_i が $a_j f_i$ に変わったのが因子分析モデルだということになります。t_i は個人についての真のスコアなのですが，古典的テスト理論の場合，テストの点数は個人のこの能力だけを反映していると考えられていたことになります。もしテストの問題が難しすぎて，まったく答えることができなければ，その人の能力はゼロということになるわけです。しかし中には悪い問題というのもあって，たとえば小学生に高校で習う知識が必要な問題を解かせるような問題があれば，誰だって解けないかもしれません。解けない問題を出

[*5] **標準得点 (Standard Score)** とは，素点 X_j を $Z_j = \dfrac{X_j - \bar{X}}{\sigma_j}$ と変換したものであることを思い出してください。標準化されたスコアは平均が 0，分散が 1 になりますので，単位の異なるもの同士であっても標準得点を使うと相互に比較可能になるのでした。

して「学力が低い」と結論づけるのはやや暴力的ですらありますね。このように，古典的テスト理論は項目の良し悪しといったものが評価できないモデルだったのです。

因子分析モデルはこれを改良し，$a_j f_i$ としました。すなわち，ある項目に対する反応 z_{ij} は，その項目が測定したい特徴を十分に反映しているかどうか (a_j) と，その人がその特徴を有しているかどうか (f_i) の両方が成立している必要があるわけです。掛け算ですから，一方がゼロであれば結果もゼロになります。すなわち測定したい特徴を反映していない項目 $(a_j = 0)$ であれば，どれほどそれについての能力 (f_i) が高くても反応できないのです。たとえば美的センスに非常に秀でた人がいても，数学のテストでその能力を反映させることはできませんよね。これは数学のテストというのが数学力を測定するものであって，美的センスを測定するものではないからです。

因子分析は知能検査や性格検査の文脈から生まれてきたものです。心理学において「知能」とは，何にでも応用可能な一般的な知能というのがあるのか，あるいは語彙力や計算力といった複数の個別の能力があるのか，という議論がありました。知能検査としていろいろなものが考えられますが，それらがきちんと当該能力を測定する検査法だったかどうかもわからないわけで，因子分析モデルはそこを評価できるようにした，とも言えます。性格検査についても同様で，特性論的に考えるならば人間の性質というのは複数のもの，たとえば外向性，神経症傾向，開放性，協調性，勤勉性[*6]などがあり，ある性格検査の項目は協調性を測定するのには向いているけれども，神経症傾向を測定するには向いていないということがあるわけです。このように，心理学と因子分析モデルは深い関係があります。

3.2.2 多因子モデル

さて先ほどは一因子，あるいは単因子ともいいますが，測定したい特徴が 1 つだけのモデルでした。学力テストなどは一因子で問題ありません。国語のテストは国語の能力を，数学のテストは数学の能力を測定していれば良いのであって，数学のテストを解くのに美的センス (真美的能力) が必要というのは，むしろ困った状況ですらありますね。しかし，知能検査や性格検査の場合はそうではありません。ある行動傾向，ある形容詞，ある検査がたった 1 つの能力・性質・心理的要因だけを反映しているとは限りません。たとえば人に優しく振る舞うといっても，その背後に外向性があるのか，あるいはそうすると自分がよく見られるからという利己的な性格があるのか等々が考えられます。1 つの項目に複数の要素 (因子) が複合的に影響していることを考えるべきです。そこで因子の数が 1 つではなく，複数ある多因子モデルを考えることにします。多因子モデルは次のように表現されます。

$$z_{ij} = a_{j1} f_{i1} + a_{j2} f_{i2} + a_{j3} f_{i3} + \cdots + a_{jm} f_{im} + d_j u_{ij} \tag{3.1}$$

記号の意味は単因子モデルと基本的には同じです。z_{ij} は個人 i の項目 j に対する反応を標準得点で表したものであり，a_{jm} は第 m 因子の**因子負荷量**，f_{im} は第 m 因子の**因子得点**を表しています。因子の数が複数あるモデルですから，$a_{j1}, a_{j2}, \cdots, a_{jm}$ とか $f_{i1}, f_{i2}, \cdots, f_{im}$ のように因子の番号と項目・個人の添字の組み合わせになっていることを確認してください。最後の e_{ij} が $d_j u_{ij}$ となっていますが，これは誤差についても他の因子と形を同じくし，項目に依存するものとそれ以外に区別しているだけです。

さて，ここでは因子を m 個あるとしています。この因子はどの項目にも共通して働くので，**共通因子 (common factor)** と呼ばれます。共通因子がいくつあるかは事前にわかりませんが，一般的にその数は数個〜十数個になります。性格心理学は長い研究の中で，性格を表す言葉に共通する因子はおおよそ 5 つぐらいであろう，という答えを得るに至りましたが，それ以外の領域では領域ごとの見解があるでしょう。知能

[*6] 小塩 (2020) のビッグファイブについての解説に基づいています。

が何種類の因子に分かれるのか，あるいはとある心の状態がどういう構造をしているのか，というのは心理学的にみても十分興味のある考え方です。もちろん因子分析によって得られる因子が，人間の潜在的な知能や概念に直接対応しているとは言えないのですが[*7]，それでも因子がどのような構造 (しくみ) をしているのかについての一定の情報を与えてくれます。多因子モデルが心理学一般で広まったのは，こうした心の「構造」に注目する学問との相性が良かったということでしょう。

3.3 因子分析の定理

3.3.1 因子分析モデルの展開

因子分析モデルも古典的テスト理論のように，式の展開から何が見えてくるか考えてみましょう[*8]。

左辺の z_{ij} は観測されたデータから算出できるものですが，右辺の因子負荷量，因子得点はいずれも未知数です。データに対して未知数が多すぎるようで，これではどのようにして答えを見つけ出せば良いのかわからないかもしれません。たとえばある人のある項目に対する標準得点が 0.12 であるとして，それが 0.4×0.3 で得られるのか，0.2×0.6 で得られるのか，はたまた他の数値の組み合わせで得られるのか，を解く数学的技術は存在しません。この方程式はこのままでは解けないのです。

そこで，この未知数だらけの方程式を解くために，因子について以下のような条件を置きます。

- 共通因子の因子得点，独自因子の因子得点は，標準化されている。すなわち，いずれの因子得点も平均点は 0 であり，分散は 1 である。
- 独自因子は共通因子，他の独自因子と相関しない。

この他に，状況に応じて因子同士の間に相関を仮定します。

- 共通因子同士の相関を認めないのを「直交因子モデル」，認めるのを「斜交因子モデル」と呼ぶ。

このような仮定を置いたら問題が解けるようになるのでしょうか？ 実はこの問題を解く鍵は，多変量データであればなんとかなるのです！

2 つの変数，j と k の標準得点から，

$$r_{jk} = \frac{1}{N}\sum_{i=1}^{N} z_{ij} z_{ik} \tag{3.2}$$

のように，相関係数が算出されることを思い出してください。先ほどの因子分析の基本代数式 (式 3.1) をこの式に代入してみましょう。

$$\begin{aligned} r_{jk} &= \frac{1}{N}\sum_{i=1}^{N} z_{ij} z_{ik} \\ &= \frac{1}{N}\sum_{i=1}^{N} (a_{j1}f_{i1} + a_{j2}f_{i2} + \ldots + a_{jm}f_{im} + d_j u_{ij})(a_{k1}f_{i1} + a_{k2}f_{i2} + \ldots + a_{km}f_{im} + d_k u_{ik}) \end{aligned} \tag{3.3}$$

これは代数の計算としてやっていくと，非常に煩雑で間違いが起きやすそうです。そこで，少し視覚化してわかりやすくしてみましょう。多項式の掛け算は，各項目の総当たり戦ですので，列方向に z_{ij}，行方向に z_{ik} の各要素を置いて，要素同士の組み合わせ表を作ります (図 3.1)。

[*7] むしろテスト項目や調査票などに対する反応パターンが因子として出てくるだけで，性格や知能が数次元あるというより，我々は性格や知能を数次元で捉えることしかできない，という言い方の方が正しいでしょう。

[*8] 以下このセクションは小杉 (2018) の原稿を再構成したものです。

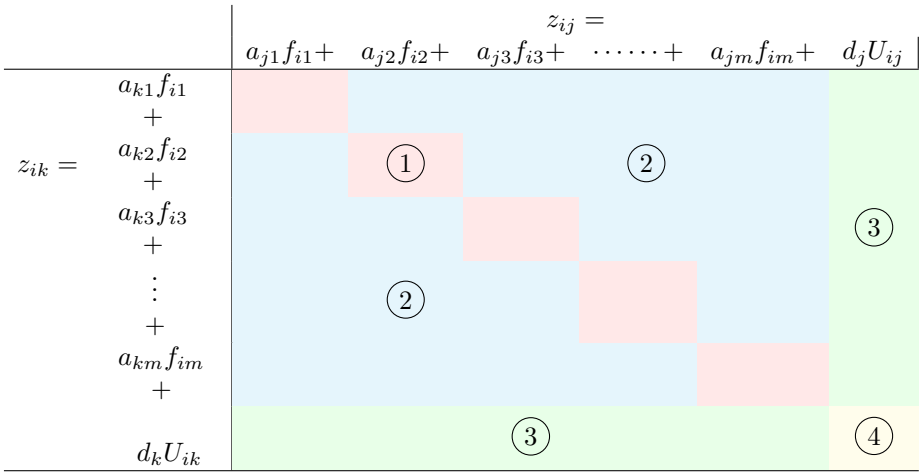

図 3.1 項目同士の総当たりを考える

図 3.1 に示されたのは個人 i についてのものであり，これが人数分ある，すなわち $\sum_{i=1}^{N}$ をつけないといけないことに注意してください。さて図を軽く色分けしてあるのですが，ここにあるように計算すべき領域を 4 つに分けて考えていきましょう。

(1)**の領域** 因子 p と p の積和部分 (同じ因子同士の掛け合わせ)
(2)**の領域** 因子 p と q の積和部分 (異なる因子同士の掛け合わせ)
(3)**の領域** 因子 $p(q)$ と独自因子の積和部分
(4)**の領域** 独自因子同士の積和部分

この各パートを順に計算していきましょう。まず(1)ですが，たとえば第一因子については

$$\frac{1}{N}\sum_{i=1}^{N} a_{j1}a_{k1}F_{i1}^2 \tag{3.4}$$

となることがわかります。ここで，a_{j1} と a_{k1} は N には関係がない ($\sum$ は i が 1 から N まで変化することを意味しているが，係数 i はどちらにも入っていない) ので，総和して割る意味がないことに気づきます。そうなると，必然的にこの式は，

$$\frac{1}{N}\sum_{i=1}^{N} a_{j1}a_{k1}F_{i1}^2 = a_{j1}a_{k1}\frac{1}{N}\sum F_{i1}^2 \tag{3.5}$$

となります。この F_{i1} は因子得点であり，上の仮定より標準化されたものだということになります。さて，標準得点と標準得点の積和平均は相関係数になることをもう一度思い出してください！そうすると，これは自分自身との相関係数を表していることになりますから，当然 $F_{i1}^2 = 1.0$ であることがわかります。

結局，(1)のエリアは

$$\frac{1}{N}\sum_{i=1}^{N} a_{j1}a_{k1}F_{i1}^2 = a_{j1}a_{k1}\frac{1}{N}\sum F_{i1}^2 = a_{j1}a_{k1} \tag{3.6}$$

と，とてもあっさり書き下すことができるのです。

つづいて(2)を見てみましょう。ここは異なる因子がかけ合わさった部分ですね。落ち着いて，第一因子と第二因子を例にして考えてみましょう。この箇所で得られるのは，

$$\frac{1}{N}\sum a_{j1}F_{i1}a_{k2}F_{i2} = a_{j1}a_{k2}\frac{1}{N}\sum F_{i1}F_{i2} \tag{3.7}$$

ということになります。ここで，F_{i1} および F_{i2} はそれぞれ第一，第二因子における個人 i の因子得点を意味しています。因子得点は標準化されていることをもう一度思い出すと，これは第一因子と第二因子の相関係数になります。ここで，この因子分析が直交因子モデルだと考えますと，因子同士に相関がないわけですから，数字としては 0.0 で消えてしまいます。するとこの部分は，

$$\frac{1}{N}\sum a_{j1}F_{i1}a_{k2}F_{i2} = a_{j1}a_{k2}\frac{1}{N}\sum F_{i1}F_{i2} = 0 \tag{3.8}$$

となることがわかりました。つまり，この領域(2)は，すべて 0 になってしまうのです。

続いて(3)の部分について考えてみましょう。これはある共通因子と独自因子の積和部分です。例によって標準得点同士の関係から，相関係数を算出することになりますが，独自因子は共通因子と無相関であることを考えると，

$$\frac{1}{N}\sum a_{i1}F_{i1}d_jU_{ij} = a_{ik}d_j\frac{1}{N}\sum U_{ij}F_{i1} = 0 \tag{3.9}$$

とこのように，この箇所もすべて 0 になってしまいます。

最後の(4)に至っては，独自因子と独自因子の積和ですから，これも

$$\frac{1}{N}\sum d_jd_kU_{ij}U_{ik} = d_jd_k\frac{1}{N}\sum U_{ij}U_{ik} = 0 \tag{3.10}$$

のように 0 になります。結局，消えて無くなるのがほとんどで，残るのは①の部分だけであり，r_{jk} を考えるときはそこだけ考慮すればよいことになります。

整理すると，

$$r_{jk} = a_{j1}a_{k1} + a_{j2}a_{k2} + \cdots + a_{jm}a_{km} \tag{3.11}$$

ということがわかります。つまり，項目 j と項目 k の相関係数は，項目 j の因子負荷量と項目 k の因子負荷量を，すべての因子について総和したものであるということです。因子分析の基本モデルから導出されるこの定理を，とくに**因子分析の第二定理**と呼びます。

ここで同じ項目同士の相関を考えてみましょう。項目 j と項目 j の相関係数は，もちろん 1.0 になりますね。これを因子分析の基本式で表すと，次のように表現できます。

$$r_{jj} = a_{j1}^2 + a_{j2}^2 + \cdots + a_{jm}^2 + d_j^2 = 1.0 \tag{3.12}$$

さて，この式が意味するのはなんでしょうか。意味を考えてみると，ある項目それ自身との相関係数は，因子負荷の二乗和からなっている，ということがわかります。これこそ**因子分析の第一定理**と呼ばれるものであり，解けるはずのなかった方程式を解くための鍵となる式なのです。

3.3.2 因子分析の定理

数式の展開はいったんここまでにして，第一定理は次のような形をしているのでした。

$$a_{j1}^2 + a_{j2}^2 + \cdots + a_{jm}^2 + d_j^2 = 1.0$$

ここで共通因子部分を，$a_{j1}^2 + a_{j2}^2 + \cdots + a_{jm}^2 = h_j^2$ のようにすると，この式は単に $h_j^2 + d_j^2 = 1.0$ となります。この h_j^2 のことをとくに**共通性 (communality)** といいますが，この式は共通性と独自因子の二乗和が 1.0 になることを意味しています。言い換えると，全体を 100% とした比率で共通性と誤差を比較できるということです。共通性は因子負荷量の二乗和で，共通因子はそのテストの背後にある共通の要因，すなわちテストで測定したいものだったわけです。古典的テスト理論では，モデル式の展開から信頼性を全分散中に示る真のスコアの割合と定義しましたが，因子分析モデルはこのように 1 つの項目 j における共通因子の割

合を算出し，項目の信頼性を考えることができます。因子分析モデルにおける信頼性は，1 項目の中に含まれる共通因子の大きさだとも言えるわけです。逆に $d_j^2 = 1 - h_j^2$ は**独自性 (uniqueness)** は，当該項目がそのテストで測っていないものの大きさを表しており，この割合があまりにも大きいと「この項目は全然関係ないものを測っちゃってるんじゃないか？ 」と疑われることになります。多因子モデルにおいては，多角的に対象を切り分けるために多くの質問を投げかけるわけですが，独自性の高い項目は回答者に負担をかけるだけの邪魔なものですから，実践上はこうした項目を除外することが少なくありません。因子分析には**単純構造の原則 (principle of simple structure)** と呼ばれるものがあり，項目は該当する因子を適切に反映し，かつ，他の因子と関係ないことが美しいとされます。尺度構成段階では，共通性 (独自性) をみて項目の良し悪しが判断されるのです。

次に第二定理を見てみましょう。第二定理は次のような形をしているのでした。

$$r_{jk} = a_{j1}a_{k1} + a_{j2}a_{k2} + \cdots + a_{jm}a_{km}$$

2 つの項目 j と k の相関係数は，それぞれの因子負荷量の積和の形で表される，というものです。ここに誤差の話は入ってこず，共通因子だけで話ができています。

左辺の相関係数は，2 つの項目がどれほど同じものを測定しているかの指標です。相関係数 (の絶対値) が高ければ高いほど，2 つの項目は同じものを指し示しているわけです。逆に相関係数が低いということは，2 つの項目に関係がないことを表します。ここで右辺に目をやりますと，右辺の各項目は因子負荷量の積の形になっています。左辺の値が小さくなる 1 つの理由は，ある共通因子 m が項目 j, k に対して，異なる方向で寄与しているからだと考えられるでしょう。そしてそのパターンが一貫していないという状況です。そもそも相関係数が小さいところからは因子を見つけ出すのは難しいのですが，そうした状況があるのはある項目ペアについて因子同士の向きがバラバラに影響しているからだと言えます。そのような状況は，測定がきちんとできているかどうか怪しいですね。**測定の一義性**とも言われますが，そのような尺度は妥当性が低いといえるでしょう。

このように，因子分析モデルは第一定理で信頼性を，第二定理で妥当性をあらわすものになっているのです。

3.4 課題

■テスト理論と因子分析　因子分析モデルは古典的テスト理論をどのように発展させたのか，添字に注意しながら数式で表現してみよう。

■因子分析の定理の導出　因子分析の定理の導出を自分でできるようになろう。

■因子分析の定理の意味　因子分析の定理が何を意味しているのか，自分なりの言葉で説明してみよう。

第 4 章

現代テスト理論

4.1 因子分析とテスト理論

ここまで，古典的テスト理論と因子分析の話を見てきました。古典的テスト理論から，テストの信頼性と妥当性の話を導きました。次に因子分析モデルによって，古典的テスト理論が多因子 (多次元) モデルに展開されるのでした。

テストの理論も心理学の研究も，目に見えない「学力」や「性格」といったものを測定するという意味で，ツールとしては同じものを使うわけです。一方ではテスト，他方では質問紙とか尺度と呼ばれますが，狙いは回答者の反応パターンから潜在的な性質を見出そうとするものです。ここで，改めてテストの理論に戻りたいと思います。ただしこれまで古典的テスト理論と呼ばれていたものは，その名の通り古典的であって，現代的なテスト理論はどうなっているのか，というところを考えてみたいと思います。

現代的なテスト理論，新しいテスト理論ともいわれますが，それは**項目反応理論 (Item Response Theory)**，あるいは項目応答理論とよばれます。略して **IRT** と表現されることも多いですね。この理論はいわゆる「学力テスト」などの要請から展開してきたものです。心理学的なアプローチからは，従属変数が連続的であったり[*1]，心理的な構造が知りたいために多因子であったりするのが，自然な発想でした。これに対して学力テストのようなものは，各項目の結果が「正答/誤答」の二種類しかありません[*2]。数値としては 1/0 の二値，バイナリデータであり，尺度水準は名義になります。また，測定したいものは一因子です。国語のテストは国語の能力を，算数のテストは算数の能力を測定するべきだと言えるからです。このように，項目反応理論は因子分析の特殊系だということができます。

受講生のみなさんは心理学での応用例の方が興味があるかと思いますが，後ほどこの項目反応理論のモデルが展開し，再び因子分析モデルに戻ってきますのでお楽しみに。それまではひとまず，テストの項目を分析するというのはどのようなことがなされているのかをみていきたいと思います。

4.2 通過率と累積正規分布

みなさんは大学入学共通テスト (旧センター試験，さらにその前は共通一次試験と言いました) や，学内の定期テスト，模試など色々なシーンでテストを受けてきたことと思います。模試などでは偏差値が明らかになり，自分の実力が相対的にどのあたりにあるのかがわかるようになっていたかと思います。大学共通テストなどは 50 万人ぐらいが一度に受験しますから，さまざまな学力の人がそこには含まれるのですが，成績を図に

[*1] 因子分析モデルの式 3.1 が Z_{ij} から始まっていたことを思い出してください。標準化されているということは，平均や標準偏差が求められているということであり，**間隔尺度水準**以上の数字が前提とされています。

[*2] 部分点というのがあるじゃないか，と思うかもしれませんが，それはひとまず横に置いてください。

902 するととても綺麗な正規分布になることが知られています[*3]。正規分布は誤差の分布でもありますが，多くの
903 要因が考えられる際の集積的データも，自然とこの形になることがわかります。
904 さて，学力のような潜在変数が標準正規分布に従うと仮定しましょう。この分布の形はどこの確率点がどれ
905 ぐらいの確率密度を持っているか，あるいはある確率点以上・以下の面積が全体の何 % かを表すものです
906 が，縦軸をある点以下の累積確率に書き直してみましょう (図 4.1)。図 4.1 の上の図がいわゆる正規分布の
907 分布の形，確率密度関数です。下の図はこれを累積確率に書き換えたものになっています。累積確率は 0%
から始まって，最終的に 100% にまでどのように増えていくかを示した図になります。

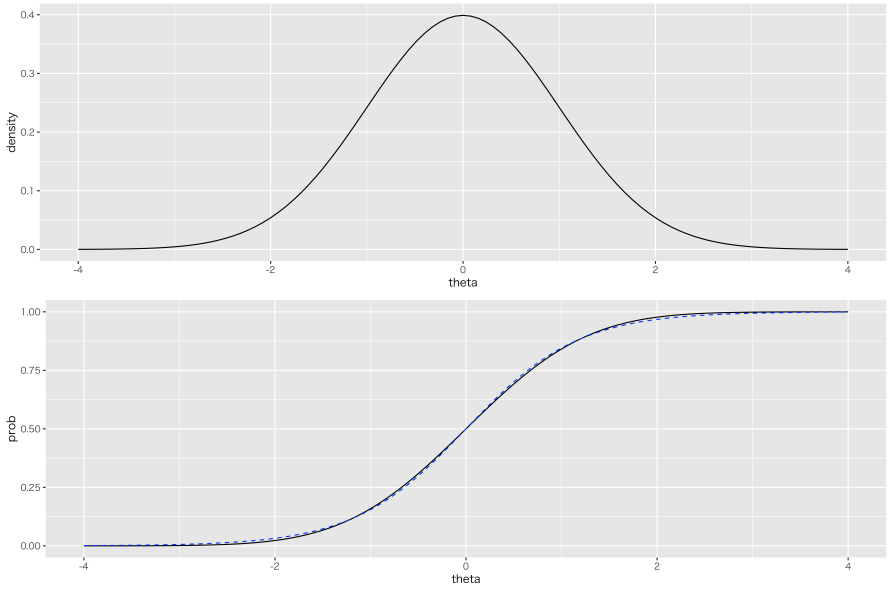

図 4.1 正規分布の確率密度関数 (上) と累積確率関数 (下)

908
909 累積正規確率は，テスト理論と密接な関係があります。というのも，学力が正規分布すると考えるなら，累
910 積正規分布の形はあるテスト項目の**通過率 (pass ratio)** と同じ形になると考えられるからです。
911 通過率とは，あるテスト項目に正答する人の割合のことです。ここで複数の項目からなる，あるテストをした
912 としましょう。正答数を数えてその人の成績とすると，よくできたテストであれば成績は正規分布に従います。
913 さらに，ある項目と成績との相関 (**IT 相関 (Item-Total correlations)**) は高いはずですね。つまりその
914 項目に正答することが，テスト全体の成績と高く関係しているので，その項目はテスト全体が測ろうとしている
915 ものを反映していると考えられるからです。また，成績をもとに被験者集団を 5 群に分けたとしましょう。「成
916 績上位群 (HH)」，「成績やや上位 (MH)」，「成績中程度 (M)」，「成績やや下位 (ML)」，「成績下位 (LL)」
917 です。このとき，各群の平均通過率を考えると，図 4.2 左上図のようになるのが理想的です。つまり，成績が
918 高い人たちの通過率は高く，成績が低い人達の通過率は低くなるはずですね。同じ図の右上は，LL 群でも
919 半分ぐらいが通過し，その後の群は過半数，ほとんどが通過するようになっています。これは，この項目が簡
920 単すぎたことを意味しています。簡単すぎる問題は，それはそれで被験者の特徴が弁別できないという意味
921 で悪い項目です。逆に図の左下にあるのは，HH 群でも半分以下の通過率しかありません。つまり難しすぎ
922 る問題です。ほとんどの人が間違えてしまうわけですから，これも良い試験問題とは言えないでしょう。右下
923 に至っては逆転していて，どうやったらこういう項目が作れるのか却ってわからないほどですが，学力の低い

[*3] 山内 (2010) の見返し (表紙を開いた最初の内側のページ) に，センター試験の成績分布が載っており，綺麗な正規分布であることが示されています。

人だけが正答できて，学力の高い人は正答しない項目ということになります。もちろんこんな項目はよくないわけで，IT 相関で負の相関が出ているわけですから，テストの文脈でいうなら「そのテストで測っていない何か別の能力を測っている」と考えるしかありません。

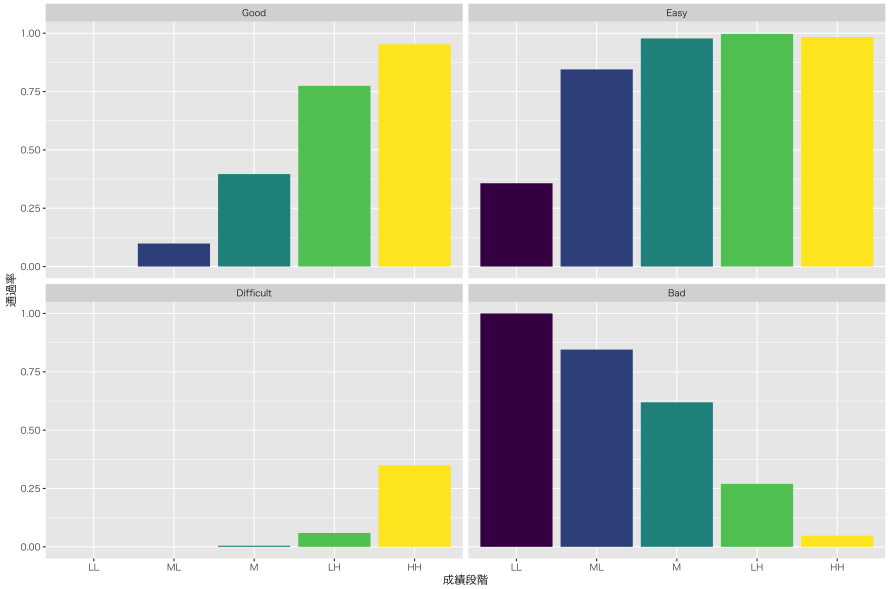

図 4.2 群ごとの平均通過率。左上が良いパターン。右上は簡単すぎる，左下は難しすぎる項目。右下は逆転していて良くない項目。

ともあれ，このようなやり方で項目の良し悪しを見ていくことができます。また，図 4.2 は 5 段階ですが，7 段階，9 段階とよりきめ細かくしていくと，理想的な形は累積正規分布により近づいていきます。新しいテスト理論による項目分析はこの累積正規分布の形を基本とし，これを拡張することで各項目の特徴を描いていくことになります。

ところで 1 つ前の図 4.1 の下の図には，実線と点線の 2 つの線が絡んでいることにお気づきでしょうか。実線の方は確率分布関数から累積確率を出して描いたものですが[*4]，点線のほうは次の関数を使って描いています[*5]。

$$f(x) = \frac{1}{1 + exp(-1.7x)}$$

この関数，図から明らかなように累積正規分布とほとんど同じですよね。累積正規分布の関数を直接使うと，積分計算 ($\int$ を使うやつ) が入ってくるのでちょっと計算が面倒ですから，こちらの関数の方を近似関数として用います。この関数のことを**ロジスティック関数 (logistic function)** と言います。ロジスティック関数そのものは，先の式から 1.7 という係数を除いた $\frac{1}{1 + exp(-x)}$ で表されるもので，$-\infty$ から $+\infty$ までのどんな数字が与えられても，答えを 0 から 1 の範囲に変換してしまう関数です。この特徴はとても便利です。というのも，結果が 0 と 1 の間に入るということは，比率を表していると考えられるからです。0/1 のバイナリデータが従属変数のときに，独立変数をこのロジスティック関数で変換してやれば 0 か 1 のどちらに近いか，どれぐらいの比率で 1 の目が出るかがわかります。項目反応理論も結果が 0/1(誤答/正答) ですから，

[*4] R では `pnorm` 関数を使って描きます。数式でいうなら，$\int_{-\infty}^{p} \frac{1}{\sqrt{2\pi}} exp(-\frac{x^2}{2})$ となります。

[*5] `exp` というのは指数関数で，$exp(x) = e^x$ のことです。ここで e は数学定数で，$e = 2.718282...$ という数字です。

「学力」のような目に見えない能力をロジスティック変換してやれば，結果が正答率になるというのは大変便利なのですね。

それではこのロジスティック関数をつかった項目分析の話に進んでいきましょう。

4.3 項目母数の特徴

ロジスティック曲線が累積正規分布の近似関数になっていること，テスト項目の分析には通過率を使って考えることを見てきました。とくに通過率の分析 (図 4.2) では，その項目が難しい設問だったのか，簡単なものだったのかを見ることができました。ロジスティック曲線もこの「項目の難しさ」を表現できるように，次のように拡張できます。

$$p(\theta) = \frac{1}{1 + exp(-1.7(\theta - b))}$$

左辺の $p(\theta)$ に含まれる θ は，潜在変数のスコア，**因子得点**であり，ここでは標準化された学力ですから，偏差値のようなものだと思ってください[*6]。$p(\theta)$ は能力 θ の人がこの項目に正答する確率=(通過率) です。

ここで b という変数が入ってきました。これが**困難度 (difficulty)** を表す指標です。$b = 0$ のときは標準正規分布と同じ形になりますが，$b = 1$ ならばこの式は右に，$b = -1$ ならば左に動きます。つまり $b > 0$ であれば難しく，$b < 0$ であれば簡単であることを表現していることになります。図 4.3 に困難度が $b = -1, 0, +1$ の時の曲線を書いてみましたので確認してください。このように困難度を表現するパラ

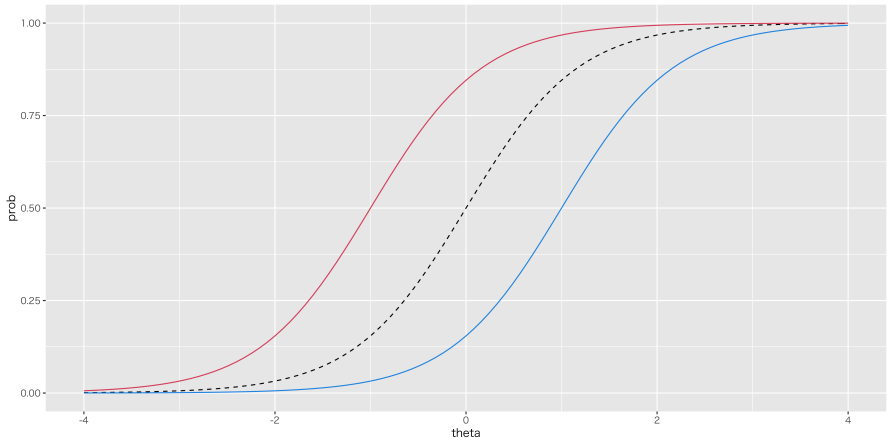

図 4.3 困難度母数の入ったロジスティック曲線。1PL ロジスティックモデルともいう。点線が $b = 0$ の標準的曲線。赤が $b = -1$，青が $b = +1$ の例

メータを追加したモデルを **1 パラメータ・ロジスティックモデル (One Parameter Logistic model)** と言います。実際のテストの回答パターンにたいし，各項目にこの曲線を当てはめて困難度を推定することで，項目を評価できるようになります。このように項目の特徴を描く曲線のことを**項目特性曲線 (Item Characteristic Curve, ICC)** といいます。

さらにパラメータを追加して，次のようにすると **2 パラメータ・ロジスティックモデル (Two Parameter Logistic model)** になります。

$$p(\theta) = \frac{1}{1 + exp(-1.7a(\theta - b))}$$

[*6] 偏差値は標準化スコア z_i を $10z_i + 50$ と変換したものを指します。ここはその変換前の z_i と同じです。

ここで a という母数 (パラメータ) が入ってきました。これは $a = 1$ だともとのモデルのままなのですが，これが小さくなると曲線が傾き，大きくなると曲線のカーブが強くなります (図 4.4)。曲線が緩やかになると (図 4.3 の赤線)，θ の違いに対して通過率の変化が乏しくなります。言い換えると感度が悪くなるわけです。逆に曲線の立ち上がりが強くなると (図 4.4 の青線)，θ がある一定のところを超えるかどうかで正答率がグッと変化することになります。つまりこのパラメータは，回答者の能力 θ を分類する力の強さを表しているのです。このパラメータのことをとくに**識別力 (discriminant)** と呼びます。

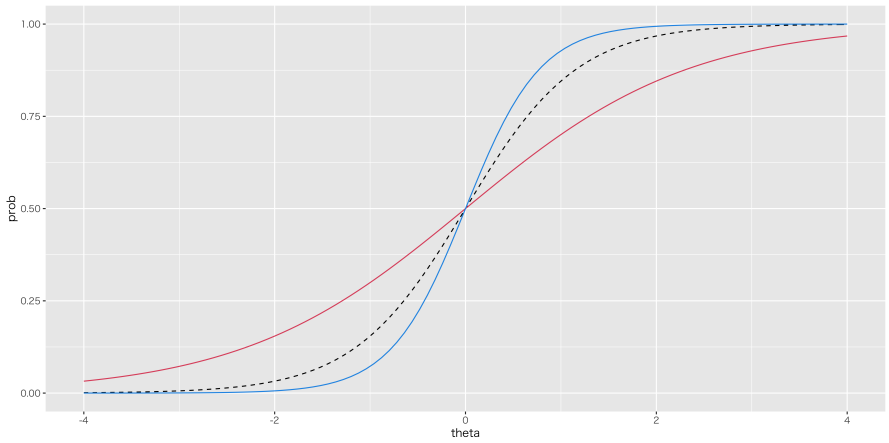

図 4.4 識別力母数の入ったロジスティック曲線。2PL ロジスティックモデルともいう。点線が $a = 1$ の標準的曲線。赤が $a = 0.5$，青が $a = 1.5$ の例

一般にはここまで紹介した 2PL モデルがよく用いられます。他にも 3，4，5 つとパラメータが増えたモデルもありますが[*7]，もとのロジスティック曲線に特徴を付け足していったもので，曲線をずらしたり，曲げたりしながらもとのデータにうまく当てはまるようにしつつ，その特徴を解釈できるように工夫しています。

いずれにせよ，テストの結果から各項目の特徴を記述します。少し例示したほうがわかりやすいでしょう。図 4.5 には心理学データ解析基礎で行った試験の結果から，2PL ロジスティックモデルを当てはめて項目分析をした例です[*8]。同じデータの項目母数を表 4.1 にも示しました。表 4.1 の数字と図 4.5 の曲線の対応をよく確認してください。

たとえば，項目 `I0022` は困難度が-1.92，識別力が 1.30 です。困難度がマイナスですので，これはかなり簡単な問題だということになります。具体的には，偏差値 50 すなわち $\theta = 0$ の人であっても 98.58% 正解するわけですから，ほとんどの人にとって容易い問題であることがわかります。ちなみにこの問題，具体的には帰無仮説検定に関する問いで，「「差がない」「偏りがない」といった仮説は何と呼ばれるか」というものでした[*9]。

一方，困難度が 0 近いところの例として項目 `M0605` をあげますが，これが平均的な難易度の質問になっています。困難度母数 $b = 0.0$ であれば偏差値 50，すなわち $\theta = 0$ の人が正答する確率が 50% の質問ということになりますが，今回の `M0605` はそれより少し難しいので，偏差値 50 の人で 20.70% の割合で正答できます。この問いについて偏差値が 70($\theta = 2$) であれば，92.96% の確率で正答できることになりますし，偏

[*7] 詳しくは大学入試センター・荘島宏二郎先生のサイト，http://shojima.starfree.jp/exmk/index.htm を参照してください。3 つめのパラメータはあて推量母数，4 つ目は上方漸近母数，5 つ目は非対称母数と呼ばれています。

[*8] 心理学データ解析基礎の授業では，過去の受講生のデータとさまざまな質問をプールしたデータを貯めてあります。みなさんが受けたテストには入っていない項目かもしれませんが，これまでどこかで出題され，回答パターンが得られている実際のテストです。

[*9] いうまでもないことですが，答えは「帰無仮説」です。

差値 30($\theta = -2$) であれば 0.51%, つまりほとんど正答は望めないということになります。ちなみにこの問題は「重回帰分析において, 標準化されたデータを使って分析をすることで, モデルの適合度を上げることができる」を Yes か No かで判断させるという質問でした。

他にも項目 I0017 は困難度が 2.17, 識別力は 0.69 です。困難度が最も高いグラフで, 図の曲線は一番右にあるラインになっています。ただ識別力がやや低いので, 曲線はよりなだらかになっていますね。困難度が高いので, $\theta = 0$ でも 7.24% しか正答できません。難しい！ $\theta = 2$ で 45.10% ですから, かなり能力の高い人でも半分は間違えるような問題です。ちなみにこれは標本分散の期待値が母分散からどれぐらいずれるのかを計算する問題でした。

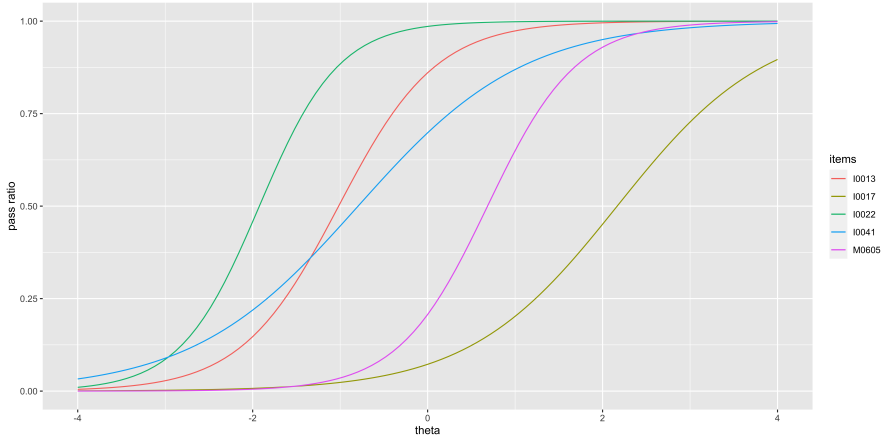

図 4.5　実際のテストに 2PL モデルを適用した例

表 4.1　各項目の困難度と識別力

	項目 ID	困難度	識別力
1	I0013	-1.02	1.05
2	I0017	2.17	0.69
3	I0022	-1.92	1.30
4	I0041	-0.79	0.62
5	M0605	0.68	1.15

4.4　被験者母数の推定

項目反応理論における因子得点の推定は, 項目の特徴を表す項目母数に対して**被験者母数**と呼ばれ, 上述の項目特性に基づいて行われます。先ほどの図 4.5 をもとに説明します。ある回答者が項目 I0013 に正答したとしましょう。その人の能力値 (因子得点,θ) はどの辺りにあるかといえば, 確率の曲線に沿った下の領域のどこかということになります (図 4.6)。この曲線, ICC は項目の特徴を表したもので, ある θ の値の能力があればどの程度の確率で正答できるかを表した項目の特徴でもありますが, 逆にある人の θ がどのあたりにありそうかを示しているともいえます。たとえばこの ICC において, θ が 0.5 のときの通過率は 86.05% ですが, 言い換えればこの項目に正解した人が $\theta = 0.5$ である可能性も高そうです。$\theta = -2$ の通過率は 14.71% ぐらいですから, 能力がこんなに低いとは思えませんし, 1 つの項目の話でしかないですが, 希望的

観測をするなら θ がもっと高い可能性もあるでしょう[*10]。

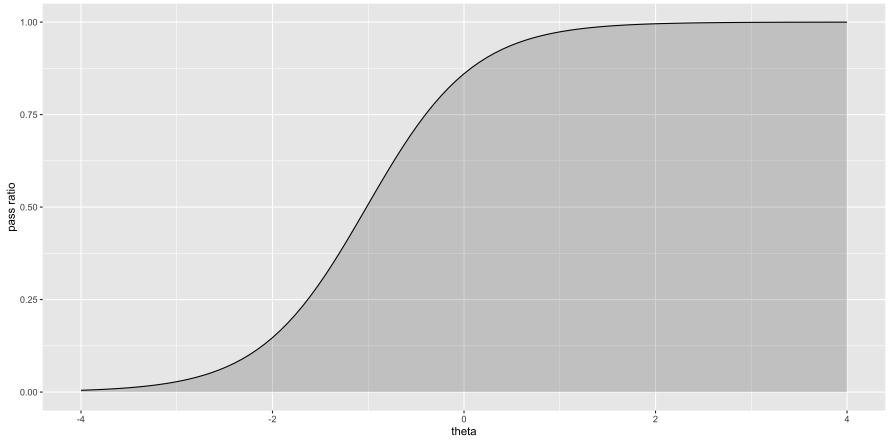

図 4.6　実際のテスト項目 I0013 に正答した人の能力がありそうな領域

次に，困難度のより高い項目である I0017 には**誤答**したとしましょう。その人は，I0017 の ICC の下の領域には**ない**はずです。項目 I0013 の ICC の下で，かつ，項目 I0017 の ICC の上にあるはずなので，図 4.7 のように塗りつぶされた領域の中に入ります。

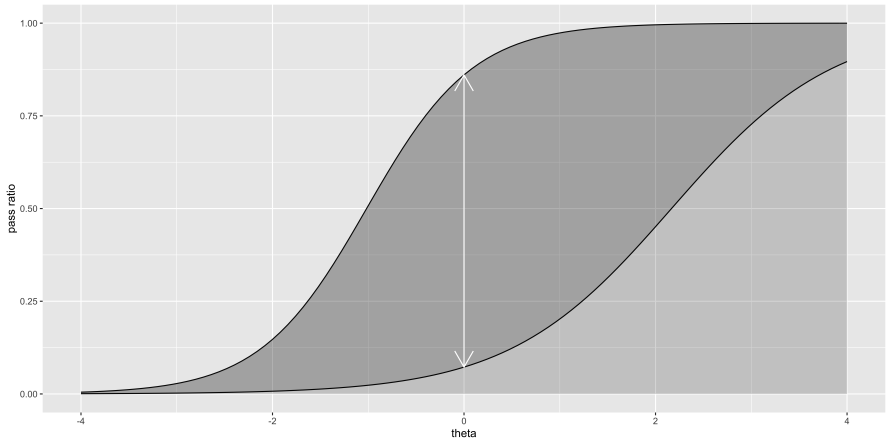

図 4.7　つぎのテスト項目 I0013 には誤答した人の能力がありそうな領域

この 2 つの曲線の間にある，濃く彩られた領域の幅が，回答者の能力値がありそうな程度を表しているのです。図 4.7 には $\theta = 0$ の可能性の大きさを矢印で示してありますが，θ の値はここだけに限らずこの曲線の幅のどこかです。ただ $\theta = 2$ や $\theta = -2$ よりは $\theta = 0$ のほうが，より「ありそう」な値だということがわかります。

ここでさらに同じ人に，項目 M0605 の質問をして，この人がそれにも正解したとしましょう。この人の能力 θ のありそうな範囲はさらに絞り込むことができます (図 4.8)。項目 I0013 よりは難しい質問に正解したわけですから，$\theta = 0$ の可能性はグッと小さくなり，それよりも $\theta = 2$ ぐらいの方がありそうだ，ということになっ

[*10] θ のありそうな「確率」とは言ってないことに注意してください。すぐにわかることですが，この ICC の下の面積を積分しても 1.0 にはなりませんのでこれは確率ではなく，**尤度 (likelihood)** のほうなのです！

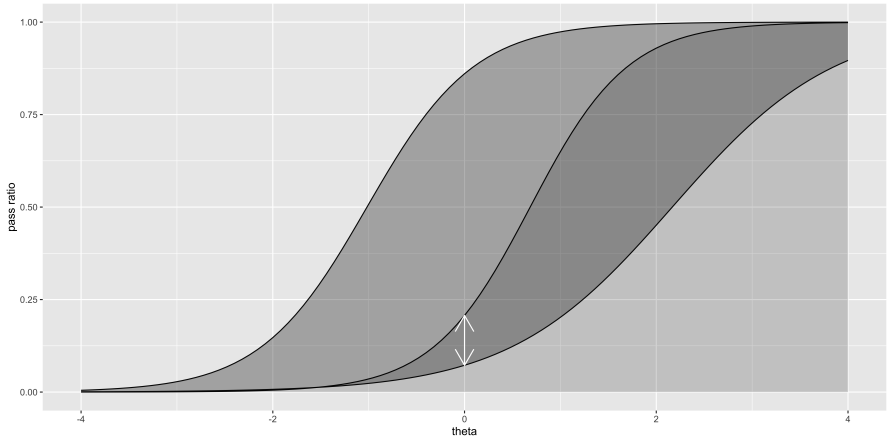

図 4.8　さらにテスト項目 M0605 には正答した人の能力がありそうな領域

てきます。このように，1 つ 1 つの項目からこの人の能力 θ がどのあたりにありそうか，というのを絞っていき，テストに含まれる項目全部を使えば，かなり狭い領域で「この辺りにあるはずだ」と推定できるでしょう。

この IRT を用いた推定方法は，このようにテストの項目ごとにその特徴を分析できるのが特徴です。テストの中でも良い項目，悪い項目というのはあるでしょうが，どのように悪いのかを困難度や識別力といった項目の特徴を使って表現できます。これらの数字は，テストに含まれる項目ごとの難易度を相対的に比較していく中で作られるものであり，回答者の能力に依存するものではありません。テスト理論が被験者の特徴と項目の特徴を分離するところから始まったことを，改めて思い出してください。

また，このような項目同士の比較から求められた項目の特徴をつかって，被験者の能力 (因子得点) を推定する方法についても紹介しました。その過程の中で気づいたと思いますが，複数の回答を通じてある回答者の能力 θ のありそうな領域が狭められていく中で，明らかにその人にとって簡単すぎる問題，難しすぎる問題は意味を成しません。たとえば図 4.8 の段階まで絞り込まれているときに，項目 I0013 よりも簡単な I0022 を出題しても，おそらくほぼ確実に正答し，そのことは「領域を狭める」ことにはなんの貢献もしないでしょう。回答者の能力に相応しい質問を選んで出すことができれば，とても効率よくその領域を絞り込んでいくことができます。こうした考え方は，次回お話しする**コンピュータ適応型テスト (Computer Adapted Test)** として実装されます。

今回はテスト理論について紹介してきましたが，この考え方は性格テストなどで用いられているリッカート形式の尺度に応用できるように，発展していきます。次回はその辺りを解説していこうと思います。

4.5　課題

■テスト理論と因子分析　項目反応理論のロジスティックモデルについて，項目母数が何を意味しているか，項目母数が変わると ICC がどのように変化するかを自分で説明できるようになろう。

■項目反応理論の利点　項目反応理論を用いた採点方法を使うと，どういう長短所があるだろうか。次回の内容に先駆けて，自分なりに考えてみよう。

第 5 章

現代テスト理論その 2

5.1 現代テスト理論の特徴

前回は現代テスト理論として項目反応理論 (IRT) を紹介しました。ロジスティック曲線を応用して項目の特徴を描画し，それを使って被験者母数を推定する方法についても解説しました。この一連の手続きに基づき，現代テスト理論の利点を考えてみたいと思います。

5.1.1 現代テスト理論の利点 1: 項目母数と被験者母数の分離

古典的テスト理論からの発展として，現代テスト理論では被験者母数 θ_i と項目母数 a_j, b_j を区分して考えるようになりました。項目母数は通過率のアイデアを精緻にしたものですが，この通過率は項目群の総和を元に考えられていたことを思い出してください。すなわち，項目母数の計算には項目の相対的な困難度だけを用いています。イメージとしては鉱物の硬度検査のようなものです。2 つの異なる硬さの物をぶつけて崩れた方が負け＝より硬度が低いと考えるように，2 つの異なる項目を被験者に与えて，より正答者数が少ない方がより難しいと考えるのです。これはつまり，回答者の学力が高かろうが低かろうが，困難度が $b_x < b_y$ であるという関係に違いはないという考え方です。

これまでのテストや心理尺度の作り方に比べると，この点が大きく違います。学校などのテストは教員が作っていますから，教員が自分の感覚で「こちらの方がより発展的な内容だ」「こちらの方が難しいだろう」という問いに大きな配点がなされたりするでしょう。その後テストの平均点をみて「今回のテストは簡単にしすぎたか」という判断をしたりするでしょう。しかし IRT では項目それ自体に困難度を決めさせますから，そこに作成者や回答者の意図は含まれません。平均正答率が高いからと言って簡単な問題なのではなく，項目の特徴として困難度が決まるのです。

たとえばサーストン尺度の作り方を思い出してください (セクション 2.2,Pp.23 参照)。サーストン尺度では尺度適用前に評定者集団によって尺度値を決めます。この評定者集団が偏った思想の持ち主だけで固められていた場合，尺度の点数は極端なものになり，普通の人がその尺度に回答すると極めて低い点数，高い点数になってしまうかもしれません。あるいはリッカート尺度の作り方を思い出してください。(セクション 2.3,Pp.25 参照)。リッカート尺度では回答者の累積度数から尺度値を算出します。先ほど同様，回答者集団の態度に偏りがあれば，尺度の点数は標準的な物ではなくなるでしょう。つまり「誰を対象に測定するか」によって目盛りが変わるようなものです。これでは測定結果の一般化は難しいでしょう。たとえば本学で作られた尺度を，他大学でやってみると違った尺度値になるのですから，研究結果はせいぜい「その大学ではそうなんだろう」となり一般化できなくなります。従来の方法は，回答者と項目の特徴が関連しすぎていたのです。

これに対し，IRT を使って項目の特徴を計算する場合は，相対的な難易度に違いはありませんから，どの

大学で作った尺度であっても統一的な解釈が可能です。尺度作成時に幅広くデータを集め，項目母数を確定させてしまえばどこででも統一的な評価ができます。テストなど学力を測定する際に大学間での違いが見られたとしても，その難易度を調整するのも簡単で，共通する項目を入れておけばそこを基準に相対的な困難度調整ができます[*1]。良問と悪問の評価と，回答者の評価を分けることは重要なポイントなのです。

5.1.2 現代テスト理論の利点 2: 被験者母数の推定の利点

被験者母数と項目母数の分離は，さらに別の利点も生み出します。それはデータが一部欠落した場合の補完に関係します。

リッカート尺度では回答者全体の相対頻度から，カテゴリの尺度値を決定するのでした。ここである人が特定の項目にだけ回答をし忘れたとします (調査研究ではよくあることで，同じような目盛りが並んでいると一行飛ばして丸をつけちゃうようなことはよくあります)。そうすると，その項目だけ合計人数が変わりますから，計算が面倒です。また相関係数を計算する時にも，その人のその項目については計算できなくなります。因子分析は相関係数から計算を始めますから，一箇所でも欠測値があるとその人のデータを抜いてしまうか[*2]，他の値を代入して補完するか[*3]，手間でも計算の時にそこだけ外して計算するか[*4]といった工夫が必要です。因子分析結果に大きな違いは出なくても，その人の因子得点は計算できないことに違いはありません。

それに対して，IRT の被験者母数の推定は，項目母数をつかった ICC をもとに一人ずつ絞り込んでいくというものでした。もしある人が回答していないことがあっても，計算ができなくなることはありません。その項目の情報が得られないので絞り込み精度は上がりませんが，ヒントが減っただけで回答できないわけではないのです。このように，得られた情報すべてを使ってその人の被験者母数 (因子得点) を推定する方法のことを，**完全情報最尤推定 (Full Information Maximum Likelihood)** と言います。このように IRT を使うと，必ずしも全問に回答していなくてもスコアは計算できるということになります。欠測値があるからその人のデータは使えない，ということがないのでいいですね！

もっというと，IRT では全員が全員同じ問題に回答する必要はありません。たとえば能力値が $\theta_i = -0.3$ ぐらいにありそうだ，と絞り込めている段階で，次の問題の困難度母数が $b_j = 2.5$ であれば，おそらくほぼ確実にその人は回答できないでしょう[*5]。その人にいくら困難度母数の高い質問を繰り返しても，ほとんど誤答がつづくだけで，とくにその人の θ がありそうな領域を狭めるヒントにはなりません。むしろ $b_j = -1$ とか $b_j = -0.5$ のような簡単な問題を出して[*6]，それらに正答できるかどうかを見極め，絞り込んで行った方が効率的です。紙に印刷されているテスト (Paper Based Test) であれば，印刷された問題は変えようがありませんから，困難度順に問題を並べると，あるところから先はずーっと不正解が続く人が続出します。ずーっと不正解なところの問題はいくら良問でも，その人の能力を測るのには役立ちません。ヒントが増えないからです。であれば，回答者の学力に合った問題を，その都度その都度出題した方がいいですよね。コンピュータを使って回答者に相応しい質問をダイナミックに組み替える，コンピュータに基づいたテスト (Computer Based Test)，別名**コンピュータ適応型テスト (Computer Adaptive Test)** というのがそれです。CAT になると，回答者ごとに出題が変わりますからカンニング対策の必要も無くなって，とても便利

[*1] こうしたテスト間の困難度調整のことをテストの**等価 (equation)** といいます。

[*2] リストワイズ削除といいます。

[*3] 欠測値補完については，平均値や中央値を代入したり，同じようなパターンで回答している人の値を使い回したり，回帰分析でその項目の値の推定値を入れたり，とさまざまな方法が考えられてきました。欠測値発生のメカニズムにもよりますが，いずれもある程度バイアスのかかった値になってしまいます。統計的によりバイアスの少ない適切な代入法が考えられてはいますが，そもそも欠測値がないのが最も望ましいことに変わりはありません。

[*4] ペアワイズ削除といいます。

[*5] 2PL モデルで $a = 1, b = 2.5$ のとき，$\theta = -0.3$ が正解する確率は 0.8492% です。

[*6] 2PL モデルで $a = 1, b = -1$ のとき，$\theta = -0.3$ が正解する確率は 76.674%，$b = -0.5$ のであれば 58.419% です。

になること間違いなしです。

5.1.3 現代テスト理論の利点 3: 信頼性についての考え方

IRT の考え方からいうと，どんな項目でも何らかの情報を提供してくれるはずです。たとえば $b_j = 3$ のようなとても難しい項目が合ったとします。これがどれぐらい難しいかというと，偏差値 70 の人でも 15% しか正解できないレベルです。ほとんどの人にとっては誤答にしかならない難しすぎる悪問だ，と言いたくなるかもしれませんが，学力が高い人がどれほど高いレベルでやれるのかを検証するためには必要な問題です。偏差値が 70 なのか，75 なのか，80 までいけるのか，といったことを見極めるためにはこの問題でないと情報が得られないことになります。逆に簡単すぎる問題であっても，より低いレベルを精緻に検証するためには必要なものなのです。

つまり，どの項目にもその項目が得意とする領域があるはずです。この項目はこの領域の回答者を絞り込む時に，最も有用な情報をもたらしてくれるはず，という θ の場所があるはずなのですね。これを表現するのが**項目情報曲線 (Item Information Curve)** といい，次の式で表される項目情報関数で描くことができます。

$$I(\theta) = a_j^2 p_j(\theta) q_j(\theta)$$

ここで $p_j(\theta)$ はその項目 j の θ における正答率 (通過率)，$q_j(\theta)$ は誤答率を表しています。能力が平均的，すなわち $\theta = 0$ のときは，0.5×0.5 になる平均的な困難度 $(b = 0)$ の設問が最も大きな値になる，というわけですね。図 5.1 にいくつかの ICC とそれに対応する IIC を描きましたので，確認してください。IIC の

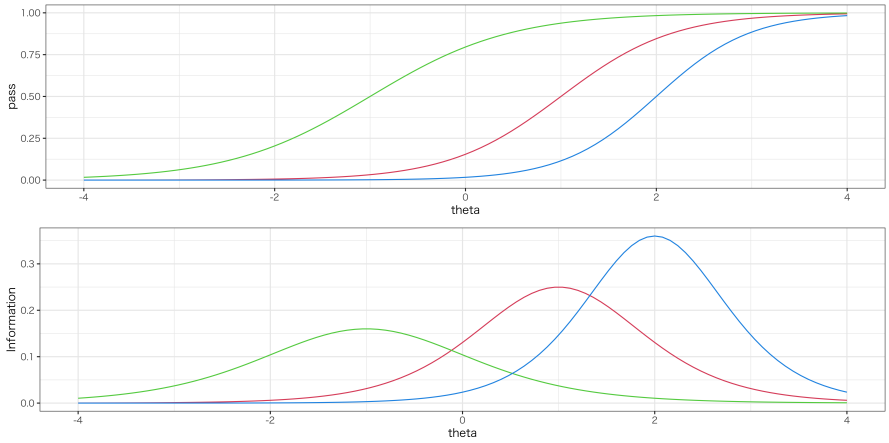

図 5.1 項目特徴曲線 (ICC, 上図) と項目情報曲線 (IIC, 下図) の例。左から順に $a_1 = 0.8, b_1 = -1$ の識別力が弱く簡単な項目，$a_2 = 1, b = 1$ のやや困難な項目，$a_3 = 1.2, b_3 = 2$ の識別力が高く困難な項目。

ピークは，対応する ICC の $\theta = 0.5$ のところにあること，識別力はピークの尖り具合に関わっていることを確認してください。

さて，IIC はその項目がどこで情報をもたらしてくれるか，ということを表しているのでした。言い方を変えると，IIC のピークはその項目の最も信頼できるところであるとも言えます。つまり IRT において**信頼性は潜在特性の関数**になっているのです。古典的テスト理論ではテスト全体の分散に占める真分散の割合のことを信頼性というのでした。因子分析理論では項目の中の共通因子負荷量の二乗和，共通性がその項目の信

頼性を表しているのでした。信頼性を見る水準がテスト全体から項目別の評価に発展したわけですが，IRT ではさらにその項目の最も感度の良いところを探る関数として，その信頼性を評価できるようになったといえるでしょう。

また IIC はある項目から得られる情報のことを意味しますが，テストに含まれているすべての情報関数を足し合わせることで，そのテストから得られる情報の大きさを関数として評価できます。すなわちテスト全体の情報量 $I_T(\theta)$ は，$I_T(\theta) = \sum_{j=1}^{M} I_j(\theta)$ であり，この関数のことを**テスト情報関数 (Test Information Curve)** といいます。図 5.1 の 3 つの項目からなる TIC を示したのが図 5.2 です。これを見ると，この 3 つ

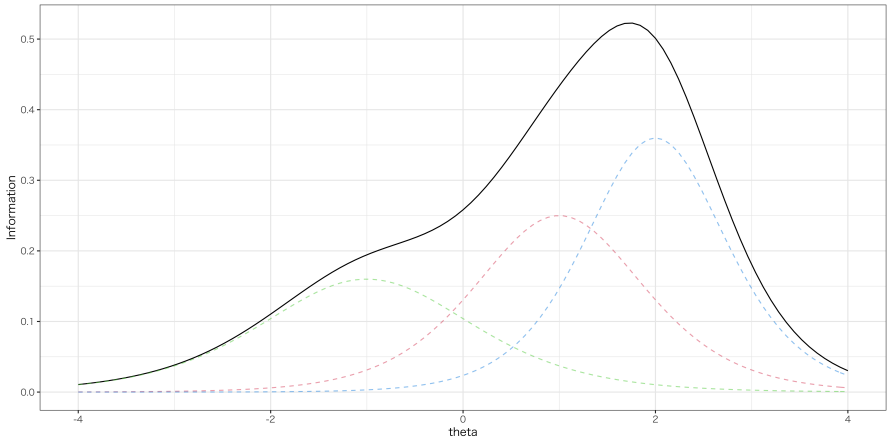

図 5.2　テスト情報関数 (黒の実線部)。理解を進めるために各 IIC を薄い点線で表現した。

の項目からなるテストは $\theta = 2$ より少し低いレベルを測定するときに最も鋭敏に働くということがわかります。このように，項目母数がわかっていれば事前にテストの特徴をどのあたりに持ってくるかを決定でき，自由自在にテストをデザインできるようになるわけです。

テストの例で話をしていますが，心理学的な領域でももちろん便利な手法です。たとえば高い認知能力レベルの人をとくに選出したいとか，精神的な健康度がごく低い人をしっかりと検出したいといった目的があれば，そのあたりにピークが来るような項目からなる質問項目からなる調査票を構成すれば良いのです[*7]。

5.1.4　現代テスト理論の問題点

ここまで見てきたように，IRT はさまざまな利点があります。しかし欠点がないわけではありません。

ここまでの話はすべて，項目母数が既にわかっていれば，という前提付きで進めてきました。では項目母数はどのようにして定めるのでしょう？ これはもちろん得られたデータから算出できるのですが，そのためには事前に多くの被験者から回答を集め，項目母数の値はほぼ間違いなくこれぐらいだろう，と言えるほど安定したものである必要があります。テストの場合，回答が 0/1 というバイナリデータで得られますから，そもそもそれほど情報がある反応ではありません。バイナリデータからその項目の特徴を安定して推定するためには，かなり多くの被験者 (数千から数万単位) を集めて項目に回答させておく必要があります。テスト項目は一度使ってみないと，項目母数がどうなるかわからないというのもポイントです。

また，CAT など項目をダイナミックに組み合わせるためには，選べるぐらいさまざまな項目を準備しておか

[*7] 具体例として小杉 (2014) をあげておきます。学校適応感を測定するため，テストのピークがやや低いところに来るようになっています。

なければなりません。項目を集めたものを**項目プール (Pool of Items)** といいますが，これも数千から数万の単位で用意しておく必要があります。なぜなら，テスト項目は事前に 1 回は使ってあるわけですから，数えるほどしか項目がなければ受験生が正答を事前に丸暗記できてしまうからです。もちろん項目プールが数千から数万あっても一度どこかで使われていますから，過去問をすべて完全に丸暗記すればその人は満点が取れてしまいます。もっともそれだけのものを覚えられるのは，ある意味学力が高いといっても差し支えないと思いますが。

日本でおこなわれる大学入学共通試験をはじめ，試験問題というのは極めて厳重な管理下に置かれ，事前にその情報が漏れることは公平性の観点から言って不適切であるとされています。しかし IRT で分析するためには，事前に項目の特徴を知っていなければならないのです。CAT をつかって入学試験などができれば，カンニング対策にもなりますし，受験生は何度でもチャレンジできるので利点も多いのですが，「公平性のために新しいテストでなければならない」となるとなかなか実用化できないところがあるというのも事実です[*8]。

ところで，心理学の場合は学力テストのように正答・誤答があるものではなく，「当てはまる」から「当てはまらない」といった軸上で多段階の反応を求めることが一般的です。テスト理論を多段階のモデルに応用できるのかと言えば，幸いにしてその答えは YES です。

5.2 段階反応モデル

リッカート法などで作られる尺度は，一般に 5，7 段階のものが多くあります。少ないものでは 3 件法[*9]であったり，ものによっては 4，6 件法であることもあります[*10]。しかしこれらの段階はいずれも順序尺度水準の情報しか持っておらず，そのままでは尺度値として使うことができません。シグマ法などで数値化すれば良いのですが，その手間を省いて分析する悪い習慣もあることは既に指摘した通りです。

IRT の多段階版はこうした問題に対応できる方法です。IRT の多段階モデルは大きくわけて 2 つあり，1 つは**段階反応モデル (Graded Response Model; GRM)**Samejima (1997)，もうひとつは**多段採点モデル (Partial Credit Model)**Muraki (1992) と呼ばれています。どちらも発想は似たようなところがあり，ここでは GRM について解説します。興味がある人は，豊田 (2012) や加藤他 (2014) など専門書を参考にしてください。

GRM の考え方の基本は，段階反応の背後には正規分布する連続的な潜在特性 θ がある，と仮定するところです。心理的な能力，学力，性質などは連続的なのですが，それが表に出てくる時は離散的 (順序的) だというわけです。図 5.3 に図示されているように，正規分布がある**閾値 (threshold)**(これを b_k と表しますが) を超えると出現する時は次のカテゴリになる，ということを考えます。図は三段階の例ですが，図から明らかなように k 段階であれば閾値の数は $k-1$ 個あることになります。横軸 θ は心理的な態度や性質の強さだと思ってください。さてそうすると，$\theta = 2.0$ ぐらいであれば，ほぼ間違いなく「当てはまる」に回答することになりますし，$\theta = -2$ であれば「当てはまらない」に回答するようになるはずです。このように θ が大きくなればなるほど最後のカテゴリに反応する確率は上がっていきますから，ここは 2PL モデルの時のようにロジスティック曲線で「当てはまる」に回答する確率を表現できるでしょう。問題は，それより下の段階に反応する確率をどのように表現するか，です。

[*8] 令和 2 年度に大学入試センター試験から大学入学共通試験に変わりましたが，改革前の計画では CAT 化することが盛り込まれていました。しかし実際には，受験生のためのコンピュータやタブレットを準備したり，安定した通信網が必要であったり，というハード的な問題もあって見送られてしまいました。

[*9] たとえば YG 性格検査は 3 段階です。

[*10] 偶数の段階にすることで，必ずどちらかの極に寄るようにして集計できます。日本人は「どちらでもない」に回答しがちな中点集中傾向があるとも言われているので，わざと肯定・否定のどちらかに寄せようという考え方です。

ここである段階に反応する確率を考えるために，少し表現を改めます。すなわち，個人 i の項目 j に対する反応が，カテゴリ k **以上**になる確率として，$P^{+}_{jk}(\theta) = P(x_{ij} \geq k|\theta)$ をまず考えます。ここで $k = 0, 1, 2, ..., K$ とします。先ほど示したように，$k = K$，すなわち一番上のカテゴリ (ここでは「当てはまる」) に回答する確率は，2PL ロジスティック関数と同じ形をしていますから，次のように表現できます。

$$P_{jK}(\theta) = P^{+}_{jK}(\theta) = \frac{1}{1 + exp(-a_j(\theta - b_{jK}))}$$

ここで右辺の a_j は**識別力**，b_{jK} はカテゴリ K の**困難度**を表しています。左辺の $P_{jK}(\theta)$ は θ の人が項目 j のカテゴリ K に反応する確率で，それは k 以上に反応する確率 $P^{+}_{jk}(\theta)$ と一致していることを表しています。

次に，もっとも低い段階に回答する確率を考えましょう。これは θ が大きくなればなるほど減っていくはずで，いわばロジスティック曲線の逆のような形をするはずです。反応確率は最大でも 1.0 ですから，逆ということは 1.0 から引いてやればよいでしょう。

$$P^{+}_{j0}(\theta) = 1.0 - \frac{1}{1 + exp(-a_j(\theta - b_{j0}))}$$

問題は「どちらでもない」に反応する確率です。これは引き算で考えることができます。すなわち「どちらでもない」以上に反応する確率から，「当てはまる」以上に反応する確率を引いてやれば良いのです。

$$P_{jk}(\theta) = P^{+}_{jk}(\theta) - P^{+}_{j,k+1}(\theta)$$

ここにあるように，段階数が増えたとしても k 番目のカテゴリ以上に反応する確率から，$k + 1$ 番目に反応する確率を引いてやれば，k 番目のカテゴリに反応する確率が得られます。この計算をして描かれる曲線のことを**項目反応カテゴリ特性曲線 (Item Response Category Charactaristic Curve; IRCCC)**，あるいは単に**カテゴリ確率曲線 (Category Probability Curve)** と呼ばれます。IRCCC は図 5.3 の下段に示されています。

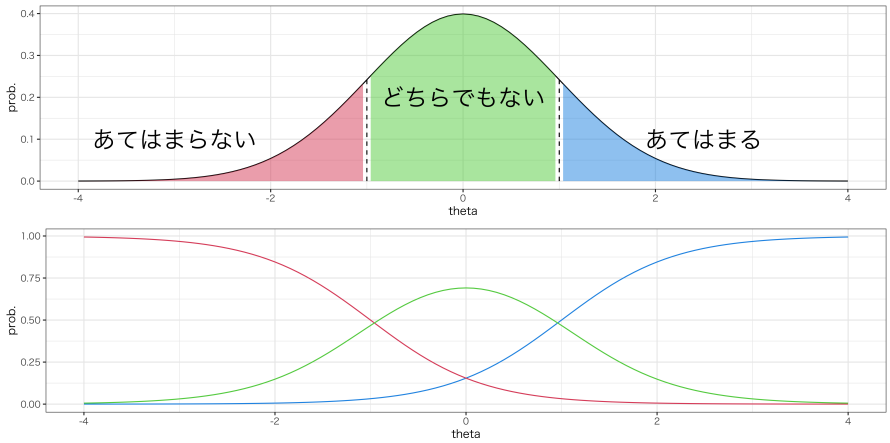

図 5.3 正規分布と閾値 (上図) と IRCCC(下図)

IRCCC を，θ の値がマイナスからプラスの方向に動かしながら見ていってください。最初は当然「当てはまらない」に反応する確率が一番高いのですが，それが徐々に下がっていきます。「どちらでもない」の反応確率は徐々に増えていき，閾値 b_{j1} で「当てはまらない」と逆転しピークを迎えることになります。その頃「当ては

まる」も徐々増えていき，閾値 b_{j2} でピークが逆転する，というようになります。ピークは逆転されても，他の確率が 0 になっているわけではなく，可能性は残っています。また IRCCC も ICC 同様に変換して，情報曲線に帰ることができます。すなわちどのあたりで鋭敏に情報を検出できるかを表現する項目反応カテゴリ情報曲線を描くこともできます。このようにして，段階反応でもその項目の特徴をデザインできるのです。

5.2.1 適切な反応段階を考える

実際の調査研究をすると，図 5.4 の上の段のような IRCCC が描かれることがあります。何かおかしいところがありませんか？これは 5 段階の反応モデルですが，4 番目の反応カテゴリがずいぶん低く，そのピークが 3 番目と 5 番目の反応カテゴリに潰されてしまっていますね。つまり，4 番目の反応がもっとも際立つシーンがないということです。言い換えるならば，これは尺度作成側が 5 段階だと思っていたにもかかわらず，回答者はどういう時に「やや当てはまる」と答えるのかがはっきりせず，実質 4 段階でしか反応していないことを表しています。

このような IRCCC が描かれてしまう場合は，$k = 3$ の反応と $k = 4$ の反応を合わせてひとつにしてしまうなど，段階の修正を考えると良いでしょう。具体的にはデータで 4 と入力していたものを，3 に置き換えたりします[*11]。修正したのが下の図になります。このように修正しても，情報関数は変わりません。同じデータから得られる情報は同じだからです。

このように 5 段階，7 段階を設定して回答者に無理やり回答を求めても，分析するとカテゴリのピークが潰れていることがあります。回答者の反応しやすいカテゴリ数を丁寧に設計してやることが重要です。もちろん無分別に尺度値をつけて，そのまま分析するのはもっとも不適切な方法です。

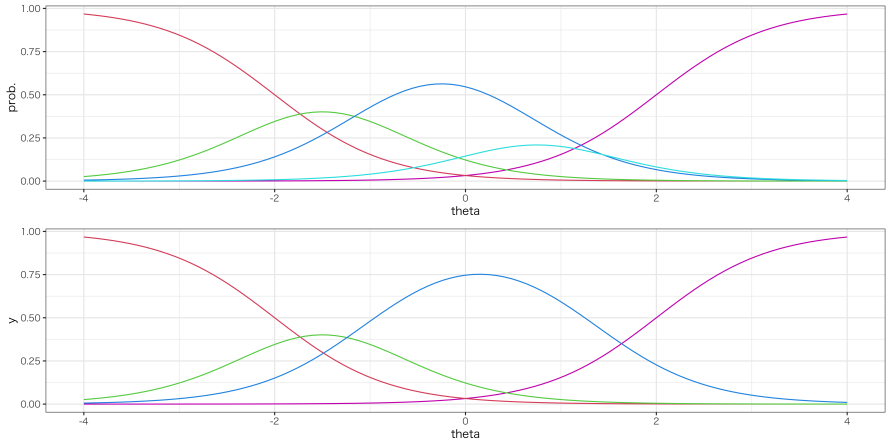

図 5.4 適切な反応段階をデザインする

5.3 因子分析の歴史と展開

ところで，因子分析モデルもテスト理論も，潜在変数モデルとしては同じでいずれも古典的テスト理論からの発展系なのでした。因子分析モデルは多段階反応が一般的で，多因子モデルで「潜在的な (心理学的な) 構造はどうか」ということを問題にします。ここでの目的は全体に共通する要素やその構造であり，何種類に別れて要素間の関係はどうなっているのか，というところが中心的関心事になります。一方，項目反応理論は

[*11] 4 を 5 に書き換えても構いません。ヒストグラムを見て，より正規分布っぽくなるようにすると良いでしょう

バイナリ反応が一般的で，因子数はひとつです。学力テストはその学力が測定できていることが重要で，因子の構造よりも因子得点をより精緻に推定できることの方が重要だからです。因子分析の言葉で言えば，因子得点をより精緻に表現しようとしているわけです。

さて，GRM は，項目反応理論の多段階モデルでした。実は GRM は因子分析モデルの発展系でもあるのです。因子分析は相関係数のモデルであったことを思い出してください。因子分析モデル自体は z_{ij} から始まっていましたが，変数同士の共分散 r_{ij} を考えるといくつかの仮定から**因子負荷量**だけのモデルに簡略化され，推定できるようになるのでした[*12]。この標準化された共分散，すなわち相関係数はピアソンの積率相関係数とも言われ，間隔尺度水準以上の数値を使って計算されます。多段階の反応は順序尺度水準ですから，相関係数を計算するのは不適切で，そのまま因子分析することはできません[*13]。では**順序尺度水準**の相関係数がないのかといわれると，あります。

順序尺度水準の変数 × 順序尺度水準の変数の相関は**ポリコリック相関係数 (polychoric correlation)** といいます。順序尺度水準の変数 × 間隔尺度水準の変数の相関は**ポリシリアル相関係数 (polyserial correlation)** といいます。ついでにバイナリ変数 × バイナリ変数の相関係数は**テトラコリック相関係数 (tetracholic correlation)** といいます。

これらの相関係数はいずれも，順序 (あるいはバイナリ) 変数の背後に連続体があると考え，潜在的な連続体 × 潜在的な連続体の相関係数を連続体のカテゴリが変わる閾値とともに推定するのです (図 5.5)。

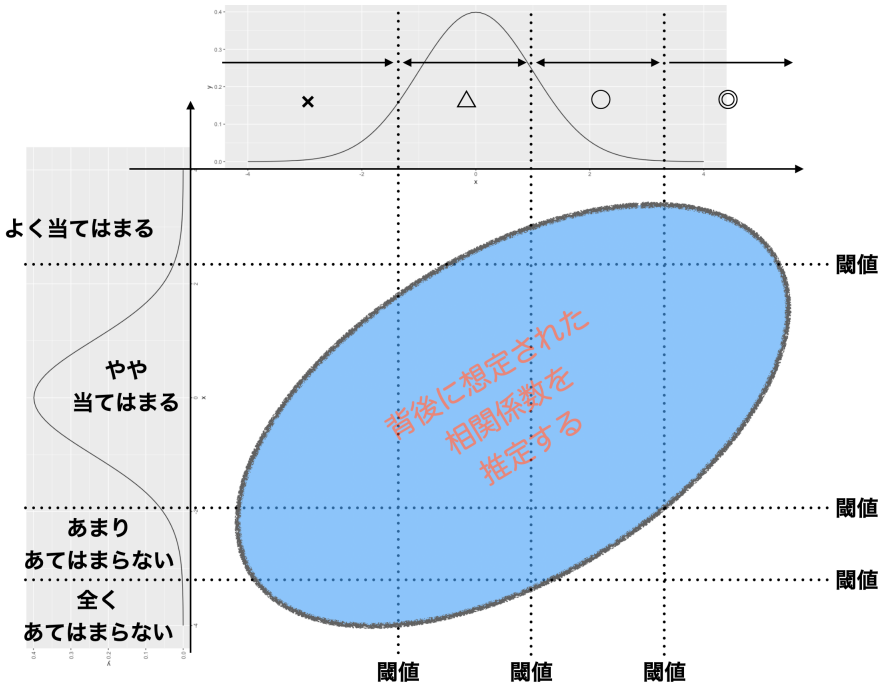

図 5.5　ポリコリック相関係数のイメージ

こうして推定された相関係数を使って因子分析を行うと，順序尺度水準に適した因子分析を行うことができます。この方法をとくに**カテゴリカル因子分析 (categorical factor analysis)** といいます。もっともこの名前を覚えておく必要はありません。**カテゴリカル因子分析は段階反応モデルと数学的に等価であるこ**

[*12] 具体的にどうやって因子負荷量を算出するかは次回以降のお楽しみです。

[*13] できないのですが，尺度値に変換することもなく機械的に分析してしまう悪い習慣が蔓延っているのは何度も指摘している通りです。くどいと思われるかもしれませんが，私は憤っているのです。

とがわかっています。GRM をすることはカテゴリカル因子分析をしていることと同じ，なのです。

もっとも GRM は項目反応理論の系列ですから，単因子構造を仮定しています。複数の因子を想定するカテゴリカル因子分析に対応するのは，正確には**多次元項目反応理論 (Multidimensional Item Response Theory)** といいます。しかし数学的･技術的には同じであり，カテゴリカル因子分析をした結果から IRCCC を描くこともできますし，実際に分析するソフト上では使用する変数がカテゴリカル (順序尺度水準) であることを指定するだけです。われわれユーザはもはや悩む必要はなく，ただただデータに適した分析をするだけで良いのです。

5.3.1 系譜の違いはどこに関係するか

因子分析モデルと項目反応理論が，カテゴリカル因子分析として概念的に統合されることを話してきました。本質的にはこのように違いがないのですが，系譜の違い，出自の違いはそれぞれの利用される文脈で，何を強調するかに影響してくることがあります。

たとえば因子分析の文脈では，共通性が低い項目は削除し，綺麗な因子構造を目指そうという考え方があります。尺度作成の中で 1 つの項目は 1 つの因子に対応しているべきであるという考え方があり (**単純構造の原理 (Principle of Simple Structure)** といいます)，もし 1 つの項目が複数の因子の影響を受けているようであれば，「美しくないので」削除されることがあります。因子分析は知能，性格，態度の研究で展開されてきたため，美しい「構造」を見つけ出すことに狙いがありますから，この美しさを損なうもの (項目) は取り除く，という方向に行きがちです。

一方で，項目反応理論はテスト理論の生まれです。もちろん回答者の能力や特性を測定するのに優れた項目とそうでない項目，という峻別はしますが，中でも「この項目は測定能力の偏差値 30 程度の回答者を測定するのに適している」とか，「偏差値 70 程度の回答者を測定するのに適している」と判断できます。偏差値 30 や 70 を測定するのに適した項目とは，非常に簡単 (ほとんどの人が正答する) だったり，非常に難易度が高い (ほとんどの人が誤答する) 項目です。心理尺度でいうと床効果，天井効果がみられる項目とされる，どちらかに偏った分布をもつ項目です。しかし，それを捨てるということにはならず，どのような項目でも回答者の能力を推定するための情報量がゼロではない，という考えから，さまざまな項目をどんどんためていく方向にいきがちです。項目反応理論の文脈では，あらゆる人に対する測定を準備しておく必要があり，むしろ回答者の特性にあわせて設問の方を選んだり，事前の項目特性から，前もってテストの項目構成をデザインする，ということを目的とするのです。

このように使われるシーンによって，「構造」か「機能 (=得点)」のどちらに注目するかが変わり，結果的に実践の方針がちがってくることもあるのです。図 5.6 にこの心理学的系譜 (左ルート) とテスト理論的系譜 (右ルート) の流れを描いてみました。

ポイントは「最終的には同じところに辿り着く」という点ですから，歴史的流れや個々のモデルの細かい数式を完全に理解していなくてもいいかもしれません。それよりは，心理学者として，あるいはテストをする側として，回答者に無理のない，それでいて必要な程度に精緻な情報が得られる適切な分析方法を選択できるようになることが重要です。

5.4 課題

■信頼性についての考え方 古典的テスト理論，因子分析モデル，項目反応理論，それぞれの信頼性の考え方を数式とともに自分のことばで説明できるようになろう。

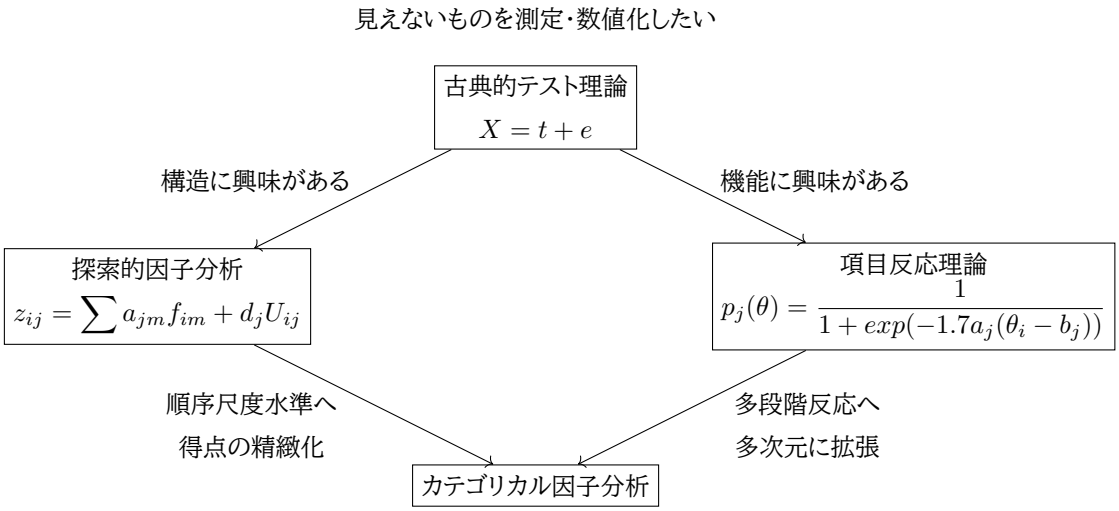

図 5.6 理論･モデルの流れ。左側のルートが心理学的系譜，右側のルートがテスト理論的系譜

■段階反応モデルの IRCCC 多段階モデルの IRCCC を，関数を描画するソフトなどを使って自分で描いてみよう。

第 6 章

行列計算の基礎

これまで古典的テスト理論，因子分析論，現代テスト理論を通じて，目に見えない潜在変数を数値化する方法について学んできました。潜在変数という心のモデルは，心理学の中心的関心事であり，実際多くの調査研究で潜在変数をモデルに組み込んで検証されています。その割には，どういったメカニズムで潜在変数が見出されているのかについての理解は十分行き渡っていないようです。たとえば因子分析モデルは，統計パッケージを使うと瞬時に「3 因子構造で因子負荷量はこれこれ，因子得点はこのようになっています」と答えを出してくれます。しかし，なぜそのような数字になったのか，どのようにそれが算出されたのかを知らなければ，何もわかっていないのと同じではないでしょうか？ 因子は「機械がやってくれるもの」と思考停止してしまうと，結局のところ私たちの知りたいことには辿り着けませんし，誤用の元になってしまいます。

なぜその肝心の箇所が放置されているかというと，数学的には**線形代数 (linear algebra)** と呼ばれる計算が必要であり，そこについての文系数学的解説がないからです。線形代数はベクトルや行列の計算，文字と式の便利な表現形式です。これを知ることの利点は，**多くの数字のセットを簡単な記号で一般的に表現できるようになること**です。変数や回答者数が数十，数百，時には数万のサイズで得られた時，１つ１つのデータにアルファベットを割り振っていたのでは間に合いませんので，線形代数はデータ解析には必須の知識です。

本講義では，線形代数の基礎を導入した上で，最終的には潜在変数，共通因子や**因子負荷量**と呼ばれるものがどのように算出されるのかを理解することを目的としています。事前の知識は必要なく，また目的に必要な最小限の知識だけで進めていきますので，一歩ずつ確実にフォローしてください[*1]。

6.1　行列とベクトル

行列やベクトルは，複数の数字をひとまとめにして扱うためのものです。まずはその基本的な形からみていきます。

■ベクトル　複数の数字を一行，あるいは一列にまとめて表現したものを，**行ベクトル (row vector)**，**列ベクトル (column vector)** といいます。

行ベクトルは次のように表します。

$$\boldsymbol{a} = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \end{pmatrix}$$

[*1] ここからの話は小杉 (2018) の pp.148–179 に同内容のものがあります。もちろん線形代数のテキストとしては他にもいろいろあり，数学的な入門としては，基礎的には村上他 (2016) が，発展的なところでは永田 (2005) が参考になるでしょう。より文系のデータ解析的解説が多いのは，絶版になってしまいましたが岡太 (2008) が最高です。

列ベクトルは次のように表します。

$$\boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

具体的には，a_1 とか b_2 のところには数字が入っています。つまり次のような形です。

$$\boldsymbol{a} = \begin{pmatrix} 1 & 3 & 5 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 2 \\ 4 \\ 6 \\ 12 \\ 8 \end{pmatrix}$$

ここで今回の $\boldsymbol{a}$ は 3 つの要素が入っていますので，サイズは 3，同じく $\boldsymbol{b}$ はサイズが 5 のベクトルです。行列の言い方に合わせて 1×3 の (行) ベクトル，5×1 の (列) ベクトル，という言い方をすることもあります。このベクトルの中の数字は，とくに関係があるわけではありません。前に入っている数字がえらいとか，横にある方が重要だ，といったことはなく，ただただ数字をまとめて扱っているだけです。数字のセットを記号ひとつで表せるので，ずいぶん楽ですよね。

さて，行数も列数も 1 であるものつまり行列でない数字は，とくに**スカラー (scalar)** と呼びます。今までは $1 + 2 = 3$ といった計算をしていましたが，この $1, 2, 3$ はすべてスカラーだといえるわけです。

■行列　行列 (matrix) とは数を長方形に並べたものです。行列として並べられた数を成分といい，成分の横の並びを行，縦の並びを列と呼びます。

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

この例は，n 行 m 列の行列を表しています。お気づきかと思いますが，行列やベクトルを表す場合は，アルファベットを太字にするのが慣例です。たとえば，A とか x は 1 つの数字を表していますが，$\boldsymbol{A}$ や $\boldsymbol{x}$ であれば行列やベクトルを表していることになります。成分を表す文字は，一般に a_{ij} のように，はじめの添え字で行番号，次の添え字で列番号を表します。行列の大きさは行数と列数とによって，$n \times m$ のように表現します。n と m が同じ，つまり行数と列数が同じであれば，これをとくに正方行列といいます。正方行列の例を次にあげておきます。

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

正方行列の中でも，i 行 j 列目の値が j 行 i 列目の値と同じである行列 $(a_{ij} = a_{ji})$ のことを**対称行列 (Symmetric Matrix)** といいます。

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

この (正方) 対称行列の形は，データ解析の中ではよくでてきます。たとえば 3 つの変数 x_1, x_2, x_3 について，その相関係数を考えたいとしましょう。相関係数は 2 つの数字の組み合わせですから，x_1 と x_2，x_1 と x_3，x_2 と x_3 について計算でき，それぞれ r_{12}, r_{13}, r_{23} と表したとします。i と j の相関係数 r_{ij} は，j と i

の相関係数と同じ ($r_{ij} = r_{ji}$) であり，また $r_{jj} = 1.0$ なのは定義から明らかです。これを行列で表すと次のようになります。

$$\boldsymbol{R} = \begin{pmatrix} 1 & r_{12} & r_{13} \\ r_{21} & 1 & r_{23} \\ r_{31} & r_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & r_{12} & r_{13} \\ r_{12} & 1 & r_{23} \\ r_{13} & r_{23} & 1 \end{pmatrix}$$

このように対称行列になっています。この行列をとくに**相関行列 (Correlation Matrix)** といいます。また相関係数は標準化された共分散でもありました。標準化するまえの相関行列は，**分散共分散行列 (Covariance Matrix)** と言います。その名前の通り，自分自身との共分散が分散になるわけですから，右上から右下にかけての対角線上にある要素 (これをとくに**対角 (diagonal)** 要素といいます) が分散であり，それ以外が共分散になっている行列です。

$$\boldsymbol{V} = \begin{pmatrix} s_1^2 & s_{12} & s_{13} \\ s_{21} & s_2^2 & s_{23} \\ s_{31} & s_{32} & s_3^2 \end{pmatrix}$$

また，正方行列の中でもとくに対角要素にのみ値があって，それ以外はすべて 0 になっている行列のことを**対角行列 (diagonal matrix)**，対角行列の中でもとくに，対角項が 1 になっているものは**単位行列 (identity matrix)** と呼びます。単位行列は $\boldsymbol{I}$ とか $\boldsymbol{E}$ で表されます。

$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

これは後ほど，掛け算をするときに「かけても変わらない状態」を表すために用いられます。

6.2 行列の四則演算と操作

行列の四則演算は，通常のスカラーのそれとは異なります。改めて，行列としての加減乗除を定義するのだと思ってください。

■加法・減法 まずは行列の足し算 (加法)，引き算 (減法) から説明します[*2] 。これはそれぞれ対応する位置にある成分を加え合わせる (減じる) ことで表されます。

$$\boldsymbol{A} + \boldsymbol{B} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2m} + b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \cdots & a_{nm} + b_{nm} \end{pmatrix}$$

数値例をみておきましょう。

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{pmatrix} = \begin{pmatrix} 6 & 8 \\ 10 & 12 \end{pmatrix}$$

これからわかるように，行列の加法，減法は大きさの等しい行列でないと成り立ちません。サイズが違うものを足そうとすると，演算できない箇所が出てしまうのです。このように行列では，「計算できない」という状態になることが少なからずあります。行列のサイズに注意が必要，ということがお分かりいただけるかと思います。

[*2] ベクトルは行列の中でも，行数あるいは列数が 1 のものですので，これで一般的に表現します。

■乗法 続いて掛け算です。まずスカラーと行列の積を見てみましょう[*3]。

$$\lambda \boldsymbol{A} = \boldsymbol{A}\lambda = \begin{pmatrix} \lambda a_{11} & \lambda a_{12} & \cdots & \lambda a_{1m} \\ \lambda a_{21} & \lambda a_{22} & \cdots & \lambda a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda a_{n1} & \lambda a_{n2} & \cdots & \lambda a_{nm} \end{pmatrix}$$

実際の計算は，各成分をスカラー倍すればよいだけですので，比較的簡単ですね。

$$2 \times \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 2 \times 1 & 2 \times 2 \\ 2 \times 3 & 2 \times 4 \end{pmatrix} = \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$$

次はベクトルとベクトルの掛け算です。これは形が変わってしまうので，注意が必要です。まずは行ベクトルに列ベクトルをかける例からみていきましょう。

$$\boldsymbol{ab} = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \sum_{j=1}^{n} a_j b_j$$

掛け算なのですが，足し合わせるという計算プロセスが入り込んでいるので，結果はスカラーになります。掛け算なのにどうして足し算の要素が入るんだ，というクレームは，今はなしです。このように計算することに決めたことで，あとあと便利なことが出て来ますから，作法にまず慣れてからにしましょう。数値例も確認しておきます。

$$\begin{pmatrix} 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix} = 1 \times 3 + 2 \times 4 + 1 \times 2 = 13$$

ここで注意して欲しいのは，両方のベクトルのサイズが同じ ($1 \times n$ ベクトルと，$n \times 1$ ベクトル，いずれもサイズは n) ということです。サイズが違うと，演算が対応しない要素が出てくるので，計算できない，が答えになります。

今度は向きを変えて，列ベクトルに右から行ベクトルをかけてみましょう。

$$\boldsymbol{ab} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_n \\ \vdots & \vdots & & \vdots \\ a_n b_1 & a_2 b_2 & \cdots & a_n b_n \end{pmatrix}$$

今度は行列になりました。かける順番が変わるとサイズが変わる (ここでは，上の例では 1×1 のサイズ，下の例では $n \times n$ のサイズ) ことに注意してください。スカラーの計算では順番を入れ替えても，たとえば $2 \times 3 = 3 \times 2$ のように同じ答えになりましたが，行列の場合は必ずしもそうはならない，ということです。

$$\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 3 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 1 \times 3 & 1 \times 4 & 1 \times 2 \\ 2 \times 3 & 2 \times 4 & 2 \times 2 \\ 1 \times 3 & 1 \times 4 & 1 \times 2 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 2 \\ 6 & 8 & 4 \\ 3 & 4 & 2 \end{pmatrix}$$

行列とベクトルの積や，行列と行列の積はこの応用になってきます。まず行列に列ベクトルを右からかける例を見てみましょう。結果は列ベクトルになります。

[*3] 式中にでてくる λ はギリシア文字でラムダといいます。小文字が λ，大文字では Λ と書きます。

$$\boldsymbol{Ab} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{m} a_{1j}b_j \\ \sum_{j=1}^{m} a_{2j}b_j \\ \vdots \\ \sum_{j=1}^{m} a_{nj}b_j \end{pmatrix}$$

ここでも掛け算なのに足し算のプロセスが入ってきています。注意深く記号を読んでみてください。数値例でも確認しておきます。

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \times 2 + 2 \times 1 \\ 3 \times 2 + 4 \times 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 10 \end{pmatrix}$$

今度は行列に行ベクトルを左からかけましょう。結果は行ベクトルになります。

$$\boldsymbol{cA} = \begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{n} a_{j1}c_j & \sum_{j=1}^{n} a_{j2}c_j & \cdots & \sum_{j=1}^{n} a_{jm}c_j \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 \times 1 + 2 \times 3 & 1 \times 0 + 3 \times 3 \end{pmatrix} = \begin{pmatrix} 7 & 9 \end{pmatrix}$$

さて，最後に行列と行列の積を考えます。行列 $\boldsymbol{A}$ と $\boldsymbol{B}$ の積が成立するのは，前者の列数と後者の行数とが等しいときに限られます。行列 $\boldsymbol{A}$ のサイズが $n \times m$，行列 $\boldsymbol{B}$ のサイズが $m \times l$ とすると，その積は $n \times l$ の行列になります。計算手続きは，次のようになります，

$$\boldsymbol{AB} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1l} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{ml} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{m} a_{1j}b_{j1} & \cdots & \sum_{j=1}^{m} a_{1j}b_{jl} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{m} a_{nj}b_{j1} & \cdots & \sum_{j=1}^{m} a_{nj}b_{jl} \end{pmatrix}$$

どうにもこれはややこしいかもしれません。足し算や掛け算が入り乱れるし，計算途中でどの要素を計算しているかわからなくなるからです。ベクトルと行列の積の時のように，前の行列の要素は左に進み，後ろの行列の要素は縦に進みますから，左手と右手で違う図形を描く認知課題のように，そもそも混乱しやすい作業なのです。

しかし 2 つほど注意をしておくと，間違いにくくなります。1 つは積によって得られる**結果の行列サイズを意識すること**です。先ほど，前の行列の列数と，後ろの行列の行数が同じでないと計算ができないといいました。つまり，$n \times m$ 行列と $m \times l$ 行列でないと計算できない (m が同じ) ということです。また，結果は $n \times l$ 行列になります。前の行列の行数，後ろの行列の列数が結果のサイズです。ここに注目しておくと，計算を始める前に，計算が可能かどうかと結果の行列サイズは想像がつくのです (図 6.1)。

また，実際に計算する際は，**前の行列に横の，後ろの行列に縦の補助線を入れる**とわかりやすいかもしれません。こうすることで，間違えて計算を進めることがないようになるからです。

$$\left(\begin{array}{cc} 1 & 2 \\ \hline 3 & 4 \\ \hline 5 & 6 \end{array}\right) \left(\begin{array}{c|c|c} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}\right) =$$

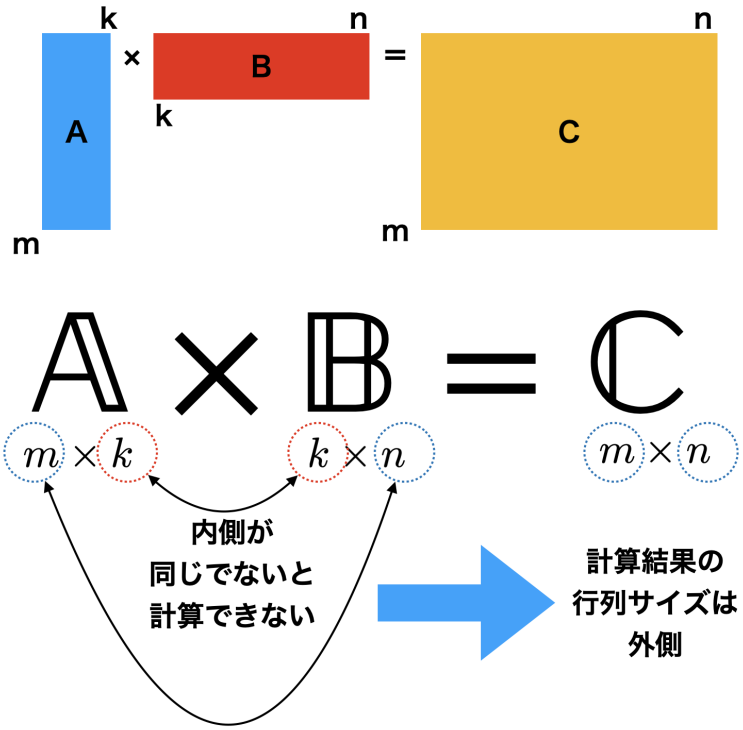

図 6.1 行列のサイズを把握する

$$\begin{pmatrix} 1\times 0+2\times 1 & 1\times 1+2\times 0 & 1\times 1+2\times 1 \\ 3\times 0+4\times 1 & 3\times 1+4\times 0 & 3\times 1+4\times 1 \\ 5\times 0+6\times 1 & 5\times 1+6\times 0 & 5\times 1+6\times 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 \\ 4 & 3 & 7 \\ 6 & 5 & 11 \end{pmatrix}$$

■転置 次に**転置 (transpose)** と呼ばれる操作を説明します。これは計算の便宜上, よく使われる行列操作のひとつです。

大きさ $n \times m$ の行列 $\boldsymbol{A}$ における i 行 j 列成分を j 行 i 列成分とする $m \times n$ 行列のことを, 元の行列 $\boldsymbol{A}$ の転置とよび, $\boldsymbol{A}'$ や $\boldsymbol{A}^T$ と表します。行列を転ばせたようなイメージです。

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \text{のとき, } \boldsymbol{A}' = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix}$$

ベクトルも転置でき, 行ベクトルを転置すると列ベクトルに, 列ベクトルを転置すると行ベクトルになります。

$$\boldsymbol{a} = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} \text{のとき, } \boldsymbol{a}' = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

また, 転置には以下のような性質があります。これは知識として知っておくだけでよいでしょう。

1. $(\boldsymbol{A}')' = \boldsymbol{A}$
2. $(\boldsymbol{A}+\boldsymbol{B})' = \boldsymbol{A}' + \boldsymbol{B}'$

3. $(\boldsymbol{AB})' = \boldsymbol{B}'\boldsymbol{A}'$
4. $(c\boldsymbol{A})' = c\boldsymbol{A}'$

■**逆行列**　最後に**逆行列**のお話をします。逆行列は割り算のイメージです。ある行列にその逆行列をかけると単位行列になる，つまり割ると 1 になるような行列のことです。

正確に表現すると，ある正方行列 $\boldsymbol{A}$ に対し，$\boldsymbol{AX} = \boldsymbol{I}$ となるような行列 $\boldsymbol{X}$ が存在するとき，これを $\boldsymbol{A}$ の**逆行列 (inverse)** と呼び，$\boldsymbol{A}^{-1}$ で表します。正方行列でない場合に逆行列はありませんし，正方行列であっても逆行列が存在しない場合もあります。逆行列の例をみてみましょう。$\boldsymbol{A} = \begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix}$ とすると，次の計算が成り立ちます。

$$\boldsymbol{AB} = \begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix} \begin{pmatrix} 3 & -1 \\ -5 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

このとき $\boldsymbol{B}$ は $\boldsymbol{A}$ の逆行列，すなわち $\boldsymbol{A}^{-1} = \boldsymbol{B}$ といえます。

とくに対角行列の逆行列は，対角成分の逆数をそれぞれ対角成分とする行列になります。

$$\boldsymbol{D} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix} \text{のとき，} \boldsymbol{D}^{-1} = \begin{pmatrix} \dfrac{1}{d_1} & 0 & \cdots & 0 \\ 0 & \dfrac{1}{d_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dfrac{1}{d_n} \end{pmatrix}$$

逆行列は，行列の世界の割り算のようなものです。これで一通り四則演算の定義ができました。

6.3 行列を使うと便利なこと

さて，ここまでで行列の計算の話をして来ましたが，どこが良いのかいまいちピンとこない，という人もいるかもしれません。そこで最後にどうしてこのような計算をするのか，何が良いのかを説明してみたいと思います。

6.3.1 行列と方程式

線形代数は「便利な書き方」の学問です。便利な書き方をするためにルールが作られていますから，ルールから学ぶと「なんでそんな変な操作をするんだ」という気持ちになるのもわかります。

では何が便利になるのでしょうか。これは方程式を解くことと関係があります。たとえば，以下のような連立方程式があったとしましょう。

$$\begin{cases} x - 2y - 5z &= 3 \\ 5x + 4y + 3z &= 1 \\ 3x + y - 3z &= 6 \end{cases}$$

これは行列で表現すると，次のようになります。

$$\begin{pmatrix} 1 & -2 & -5 \\ 5 & 4 & 3 \\ 3 & 1 & -3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix}$$

この左辺を行列とベクトルの式の計算ルールにのっとって展開してみてください。ちゃんと最初の連立方程

式の左辺になることがわかると思います。かけて足して，という面倒な計算ルールは，連立方程式を簡単に表記するためのものだったのですね。

最終的にはこの方程式を解いて，次のように答えを求めます。

$$\begin{pmatrix} 1 & -2 & -5 \\ 5 & 4 & 3 \\ 3 & 1 & -3 \end{pmatrix} \begin{pmatrix} x=-1 \\ y=3 \\ z=-2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix}$$

皆さんも学校で習ったように，このような連立方程式を解く方法として，加減法や代入法というのがあります。ですがここはひとつ，行列を使った解法を考えて見ましょう。

そのような解法のひとつ，消去法は，ひとつの方程式を何倍かして，他の方程式に加えることにより，方程式をどんどん簡単にしていくというものです。まず，第一の式を 5 倍，あるいは 3 倍して，第二，第三の式から x の項を消去します。

$$\begin{cases} x-2y-5z &= 3 \\ -14y-28z &= 14 \\ -7y-12z &= 3 \end{cases}$$

第二の式の係数を簡単にしておきましょう。

$$\begin{cases} x-2y-5z &= 3 \\ y+2z &= -1 \\ -7y-12z &= 3 \end{cases}$$

第二の式を 7 倍して，第三の式から y を消去します。

$$\begin{cases} x-2y-5z &= 3 \\ y+2z &= -1 \\ 2z &= -4 \end{cases}$$

あとはこれの 3 行目から $z=-2$ が得られ，芋づる式に $x=-1$，$y=3$ が得られました。

この操作は，式を一本ずつ，あるいは 2 つの式を組み合わせて文字を消していく消去法を係数全体に行う操作になっています。実際，ここで操作される係数だけ見ていくと，次のようになります。

■第一段階

$$\begin{pmatrix} 1 & -2 & -5 \\ 0 & 1 & 2 \\ 0 & -7 & -12 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ 3 \end{pmatrix}$$

■第二段階

$$\begin{pmatrix} 1 & -2 & -5 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ -4 \end{pmatrix}$$

さらにこの方法を改良した，ガウスージョルダンの消去法というものがあります。この手法による係数の変化を，行列表記で見ていくことにします。

まず第一段目は同じです。

$$\begin{pmatrix} 1 & -2 & -5 \\ 0 & 1 & 2 \\ 0 & -7 & -12 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ 3 \end{pmatrix}$$

次に，第二の方程式を用いて第一と第三の式から y の係数を消してしまいます。

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 4 \end{pmatrix}$$

最後に，第三の式の z の係数を 1 にして，第一，第二式の z の係数を消してしまいましょう。

$$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}-1\\3\\-2\end{pmatrix}$$

最後の形を見ると，左辺は単位行列になっていますから，

$$\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}-1\\3\\-2\end{pmatrix}$$

と解が求められたことがわかります。ここで注目すべきは，連立方程式の解を求めるプロセスは係数行列を単位行列に変えていくプロセスだった，ということです。係数行列が単位行列になれば，それはもう答えを出したことになるのです。

さて，係数行列を $\boldsymbol{A}$ とすると，その逆行列 $\boldsymbol{A}^{-1}$ があれば $\boldsymbol{A}^{-1}\boldsymbol{A}=\boldsymbol{I}$ となるのでした。であれば，連立方程式の右辺にあったベクトルに $\boldsymbol{A}^{-1}$ をかけてやれば，一気に答えが求まるではないですか。

実際に見て見ましょう。

$$\begin{pmatrix}1&-2&-5\\5&4&3\\3&1&-3\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}3\\1\\6\end{pmatrix}$$

この連立方程式に対して，次のような操作をします。

$$\begin{pmatrix}1&-2&-5\\5&4&3\\3&1&-3\end{pmatrix}^{-1}\begin{pmatrix}1&-2&-5\\5&4&3\\3&1&-3\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}1&-2&-5\\5&4&3\\3&1&-3\end{pmatrix}^{-1}\begin{pmatrix}3\\1\\6\end{pmatrix}$$

とします[*4]。すると左辺は単位行列になりますから，次のように計算すれば一気に答えが求まることになるのです。

$$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}-1\\3\\-2\end{pmatrix}$$

つまり，連立方程式を解くという問題が，係数行列の逆行列を求める問題になります。また，逆行列は存在しないこともある，ということでしたが，その場合その連立方程式は解けない，ということになります。

6.4 課題

■線形代数の練習問題 行列 $\boldsymbol{A}=\begin{pmatrix}1&2&-1\\3&1&0\end{pmatrix}$ のとき，次の計算をしなさい。なお，$\boldsymbol{I}_n$ とは $n\times n$ の単位行列，$\boldsymbol{O}$ とはすべての要素が 0 の適当なサイズの正方行列であることを表します。

1. $\boldsymbol{A}'\boldsymbol{A}$
2. $\boldsymbol{A}\boldsymbol{A}'$
3. $\boldsymbol{A}\boldsymbol{I}_3$
4. $\boldsymbol{A}\boldsymbol{O}$

[*4] 数値的には $\begin{pmatrix}1&-2&-5\\5&4&3\\3&1&-3\end{pmatrix}^{-1}=\begin{pmatrix}15/28&11/28&-1/2\\-6/7&-3/7&1\\1/4&1/4&-1/2\end{pmatrix}$ という行列です

■線形代数の練習問題その 2 行列 $\boldsymbol{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} 3 & 1 & 5 \\ 2 & 4 & 6 \end{pmatrix}, \boldsymbol{C} = \begin{pmatrix} 4 & 1 \\ 2 & 3 \end{pmatrix}$, 列ベクトル $\boldsymbol{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$, 行ベクトル $\boldsymbol{y} = \begin{pmatrix} 2 & 8 \end{pmatrix}$ とするとき, 次の計算をしなさい。なお, 計算が定義されていないものについては「計算できない」と回答しなさい。

1. $\boldsymbol{A} + \boldsymbol{B}$
2. $\boldsymbol{A} - \boldsymbol{C}$
3. $\boldsymbol{AB}$
4. $\boldsymbol{AC}$
5. $\boldsymbol{B}'\boldsymbol{A}$
6. $\boldsymbol{Ay}'$
7. $\boldsymbol{xy}$
8. $\boldsymbol{xB}$
9. $\boldsymbol{x}'\boldsymbol{B}'$
10. $\boldsymbol{yx}$

■連立方程式を解く 次の連立方程式を解きなさい。

$$\begin{cases} x - 2y + 3z = & 1 \\ 3x + y - 5z = & -4 \\ -2x + 6y - 9z = & -2 \end{cases}$$

第 7 章

行列による関係の表現

前回から線形代数の話をしています。線形代数は数字をセットで扱うための表現方法，計算方法ですから，多変量データを分析しようという時には必須の技術になります。前回は線形代数の導入ですから，計算方法を解説してきましたが，今回はこの計算方法を使って具体的にデータをどのように表現し，どのように計算するのかを見ていくことになります。

7.1 データの行列表現

ここまで行列の形ばかり見て来ましたが，狙いはあくまでも調査研究など，多変量データを扱う場面での利用です。なぜ多変量データ分析をする際にこのような知識が必要なのか，思うかもしれません。ですが，得られるデータは行列として扱うと表現が大変便利なのです。たとえば質問項目が m 個あって，調査対象者 n 人から回答を得たとすると，データは次のように表現できます。

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

データ全体をこうして，ひとつの記号で表現できたら便利ですよね。これらを使ったデータの表記に慣れておきましょう。

各反応の平均点は以下のように表現されます。まず，要素がすべて 1 からなるベクトルを次のように表します。

$$\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

わかりにくいかもしれませんが，この $\mathbf{1}$ は太字でベクトルを表しており，スカラーの 1 とは違うことに注意してください。

さて, 各項目の和はベクトルの掛け算の定義によって次のように表現できます。

$$\boldsymbol{X}'\mathbf{1} = \begin{pmatrix} \sum_{i=1}^{n} x_{i1} \\ \sum_{i=1}^{n} x_{i2} \\ \vdots \\ \sum_{i=1}^{n} x_{im} \end{pmatrix}$$

これを使って平均値 (列) ベクトル $\boldsymbol{m}$ を次のように表すことができます。

$$\boldsymbol{m} = \frac{1}{n}\boldsymbol{X}'\mathbf{1} = \begin{pmatrix} 1/n \sum_{i=1}^{n} x_{i1} \\ 1/n \sum_{i=1}^{n} x_{i2} \\ \vdots \\ 1/n \sum_{i=1}^{n} x_{im} \end{pmatrix} = \begin{pmatrix} \bar{x}_{.1} \\ \bar{x}_{.2} \\ \vdots \\ \bar{x}_{.m} \end{pmatrix}$$

ここで $\bar{x_{.}}1$ は 1 つめの添字 i を足し合わせて割ることでなくしていますから, $\bar{x_1}$ のように省略して書くことがあります。このときの 1 は「第一番目の変数」という意味であり, 個人の情報がなくなっている変数を意味していることに注意してください。

さて, 平均からの偏差を要素に持つ行列 $\boldsymbol{V}$ を考えたとします。

$$\begin{aligned} \boldsymbol{V} &= \boldsymbol{X} - \mathbf{1}\boldsymbol{m}' \\ &= \boldsymbol{X} - \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} \bar{x}_{.1} & \bar{x}_{.2} & \cdots & \bar{x}_{.m} \end{pmatrix} \\ &= \boldsymbol{X} - \begin{pmatrix} \bar{x}_{.1} & \bar{x}_{.2} & \cdots & \bar{x}_{.m} \\ \bar{x}_{.1} & \bar{x}_{.2} & \cdots & \bar{x}_{.m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{x}_{.1} & \bar{x}_{.2} & \cdots & \bar{x}_{.m} \end{pmatrix} \end{aligned}$$

この行列 $\boldsymbol{V}$ のサイズは $n \times m$ であることに注意してください。これはまた, 次のように表すこともできます。

$$\boldsymbol{V} = (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}')\boldsymbol{X}$$

ここで $\boldsymbol{I}$ は適切なサイズの単位行列です[*1]。

これを使うと, たとえば分散共分散行列 $\boldsymbol{S}$ は次のようになります。

$$\boldsymbol{S} = \frac{1}{n}\boldsymbol{V}'\boldsymbol{V} = \begin{pmatrix} s_1^2 & s_{12} & \cdots & s_{1m} \\ s_{21} & s_2^2 & \cdots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_m^2 \end{pmatrix}$$

[*1] 適切なサイズってなんだよ, と思いますよね。これは計算に合うようなサイズ, という意味です。具体的に考えてみますと, $\boldsymbol{I}$ の後ろは $\mathbf{1}\mathbf{1}'$ です。$\mathbf{1}$ は $n \times 1$ の列ベクトルで, 転置したものと掛け合わせますから, $\mathbf{1}\mathbf{1}'$ のサイズは $n \times n$ です。行列の引き算は同じサイズでないと成立しませんから, ここでの $\boldsymbol{I}$ も $n \times n$ でなければなりません。カッコの中身が $n \times n$ で, それにサイズ $n \times m$ である $\boldsymbol{X}$ をかけますから, 計算結果や右辺のサイズは $n \times m$ になります。

ここで s_j とあるのは第 j 変数の標準偏差を，s_{jk} とあるのは第 j 変数と第 k 変数の共分散です。添え字は変数番号になっています。また，ここでもサイズに注目してください。$\boldsymbol{V}'\boldsymbol{V}$ は，サイズで言うと $m \times n$ と $n \times m$ の積ですから，$m \times m$ になります。この行列は正方対称行列です。

また，対角項に各変数の標準偏差 s_j が入った行列 $\boldsymbol{Q}$ を以下のように定めるとしましょう。次のような行列です。

$$\boldsymbol{Q} = \begin{pmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_m \end{pmatrix}$$

そうすると，これの**逆行列**をつかって標準得点行列 $\boldsymbol{Z}$ を次のように表すことができます。

$$\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{Q}^{-1}$$

さらに，これを用いて相関行列 $\boldsymbol{R}$ を次のように表すことができます。

$$\boldsymbol{R} = \frac{1}{n}\boldsymbol{Z}'\boldsymbol{Z} = \begin{pmatrix} 1 & r_{12} & \cdots & r_{1m} \\ r_{21} & 1 & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & 1 \end{pmatrix}$$

データのサイズにかかわらず，一般的にこのように表現できるのはとてもわかりやすいですね。

7.2 線形モデルの行列表現

ベクトルや行列の記法をつかうと，**回帰分析**や**重回帰分析**の式がとても単純な形で表現できます。

回帰分析は，$Y = aX + b + e$ という式で表現できる，ということでしたが，式中の X や Y は観測されたデータですので，$X = \{x_1, x_2, \cdots, x_n\}$，$Y = \{y_1, y_2, \cdots, y_n\}$ というベクトルだと考えることができます。ですから，正確に書けば，ベクトルを使って次のように書くべきです。

$$\boldsymbol{Y} = a\boldsymbol{X} + \boldsymbol{b} + \boldsymbol{e}$$

これは，要素を表現しながら書くと[*2]次のようになります。

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = a \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} b \\ b \\ \vdots \\ b \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

列ベクトル $\boldsymbol{X}$ に定数 a をかけて得られるのは同じサイズの列ベクトル，列ベクトル同士は足しても同じサイズの列ベクトルですから，左辺と右辺はどちらも列ベクトルで，対応関係が取れていることになります。

ここで少し表現の工夫をします。説明変数 X のベクトルの左に数字の 1 だけが入った列を作ります。また，係数もまとめてベクトル $\boldsymbol{\beta}$ を次のように用意します。

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} b \\ a \end{pmatrix}$$

[*2] エレメントワイズ element-wise の表現，と言ったりします。

このようにすると，回帰分析の式は

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}$$

と表すことができます。とても簡単な表現になりました (試しに各行を行列の計算式に則って計算してみてください。うまく表現できていることがわかると思います)。

さらにこの表現はありがたいことに，複数の説明変数がある重回帰分析の時でも同じ形で表すことができます。重回帰分析は，これまでの書き方ですと $Y = a_1X_1 + a_2X_2 + \cdots + a_nX_n + b + e$ というようにしていました。ここで，係数と切片を 1 つの行列で表現する時わかりやすくするために，少し書き換えて $Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \cdots + e$ としましょう。記号が変わっただけで中身は同じ，意味も同じです[*3]。ただ，切片 b を β_0 として右辺の一番前に持って来ました。というのも，そうするとベクトルで書く時にわかりやすいからです。

説明変数行列の左端に 1 を入れたベクトルを追加し，回帰係数 β もセットにして，次のように表現します。

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}$$

とすると，重回帰分析の式は次のように簡単な表現に変わります。

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}$$

なんと，説明変数が m 個に増えたのに，式の形は (単) 回帰分析のそれと同じではありませんか！

このように，行列表現をすると多くの変数を一気に扱い，表現できるのです。このため，多変量解析ではベクトルの表記が基本になります。サイズを気にせず一般的に表現できるからです。

実際にこれらの式を読む時は，行列のサイズをイメージしながら読むと良いでしょう。たとえば左辺の $\boldsymbol{Y}$ はサイズ n のベクトルなのですから，右辺の $\boldsymbol{X}\boldsymbol{\beta}$ もサイズ n の縦ベクトルになるはずなのです。実際，$\boldsymbol{X}$ は $n \times (m+1)$ の行列で，β は $(m+1) \times 1$ のベクトルですから，計算結果は $n \times 1$，つまりサイズ n の縦ベクトルです。

7.3 デザイン行列

ところで，心理統計と言えば平均値の差を見ることだ，という話はこれまで散々聞いてきたところかと思います。心理学は要因計画を立て，標本の平均値差から母集団に話を一般化するために，推測統計の知見を使って，帰無仮説検定やベイズ推定法を駆使するというやつです。この要因計画は実は線形モデルの一環であり，**一般線形モデル (General Linear Model)** と呼ばれています。これを行列で表現することを，ここでは少し考えてみたいと思います。

まずは回帰分析と要因計画は何が同じで何が違うのかを，はっきりさせましょう。同じところは線形モデルであるというところ，違うところは，回帰分析は説明変数も従属変数も連続変数であるのに対し，要因計画では一般に説明変数が離散変数であること，でした。離散変数であるとは，言い方を変えると名義尺度水準の数字だということです。すなわち「統制群」か「実験群」か，という違いを表すのに，$0, 1$ と言った数字を割り

[*3] 厳密に言えば記述統計学として誤差を最小にするように推定した係数はアルファベット $b_0, b_1, \ldots$ で，推測統計学として母数の推定値として算出した係数はギリシア文字 $\beta_0, \beta_1 \ldots$ で表現する，というルールです。すでに習ったように，最小二乗法での推定値と最尤法での推定値は，誤差が正規分布する場合一致しますので，ただ書き変わっただけだと思っていただいて問題ありません。

振ったものです。これは別に 3 と 12523, という数字を割り振ったと言ってもいいのです。だって名義尺度水準は, 数字と対象が一対一対応していれば良いのですから。

ここでは数学的に話をしやすくするために, 統制群を 0, 実験群を 1 とするとしましょう。線形モデルという枠組みは一緒なのですから, 従属変数 y_i が説明変数 x_i によって変わるわけですが, ここではこの x が $\{0, 1\}$, というわけです。線形モデルを要素ごとに表現すると次のようになります。

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

この時, i さんは統制群に割り振られていたとすると, $x_i = 0$ ですから, この式は次のようになります。

$$y_i = \beta_0 + e_i$$

逆に, i さんが実験群に割り振られていたとしますと, $x_i = 1$ ですから, この式は次のようになります。

$$y_i = \beta_0 + \beta_1 + e_i$$

これを見るとわかるように, 両群のベースライン β_0 は同じで, そこに効果 β_1 が乗っかるかどうか, が興味の的になります。この式の右辺に i は誤差成分しかなく, 誤差を抜きにすると従属変数は β_1 だけ変化するはずだ, というところから, 平均値の差を検証しましょうと言ってることになるのでした。

これも x_i が個人ごとに変わるベクトルだと考えると, 行列表現では次のようになります。

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}$$

同じ形ですね！ただし, ここでベクトル $\boldsymbol{x}$ は, その中身が $\boldsymbol{x} = (0, 0, 0, 1, 1, 0, 1, 0, ...)$ のように実験群か統制群かを分けるフラグが入っているだけになります。

以上は実験群と統制群という 2 群の話でしたが, 3 群以上になっても基本的なアイデアは同じです。たとえば表 7.1 のようなデータセットがあったとしましょう。

表 7.1 群間要因 (3 水準) のデータセット例

参加者番号	群わけ	従属変数
1	A	3
2	A	3
3	A	4
4	A	4
5	B	6
6	B	7
7	B	8
8	B	9
9	C	7
10	C	6
11	C	5
12	C	4

ここで群ごとの効果を表現したいとすると，次のように書くことになります。

$$
\begin{aligned}
y_1 &= \beta_0 + \beta_1 + e_1 \\
y_2 &= \beta_0 + \beta_1 + e_2 \\
y_3 &= \beta_0 + \beta_1 + e_3 \\
y_4 &= \beta_0 + \beta_1 + e_4 \\
y_5 &= \beta_0 + \beta_2 + e_5 \\
y_6 &= \beta_0 + \beta_2 + e_6 \\
y_7 &= \beta_0 + \beta_2 + e_7 \\
y_8 &= \beta_0 + \beta_2 + e_8 \\
y_9 &= \beta_0 + \beta_3 + e_9 \\
y_{10} &= \beta_0 + \beta_3 + e_{10} \\
y_{11} &= \beta_0 + \beta_3 + e_{11} \\
y_{12} &= \beta_0 + \beta_3 + e_{12}
\end{aligned}
$$

添字の対応に注意しながらみてくださいね。群 A の効果は β_1，群 B の効果は β_2，群 C の効果は β_3 になります。

この β それぞれを該当するところ (割り当てられた群) だけに対応させつつ，統一的表現形である $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}$ にするには，次のように書く必要があります。

$$
\begin{pmatrix} 3 \\ 3 \\ 4 \\ 4 \\ 6 \\ 7 \\ 8 \\ 9 \\ 7 \\ 6 \\ 5 \\ 4 \end{pmatrix}
=
\begin{pmatrix}
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}
+
\begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \\ e_{11} \\ e_{12} \end{pmatrix}
$$

このような表記になった時の $\boldsymbol{X}$ のことをとくに，実験のデザインを表している行列ということで，**デザイン行列 (design matrix)** といいます。デザイン行列は自分で書くと面倒な感じがしますが，ともかくこのような書き方で $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}$ の統一表現は可能です。

ところでこのデザイン行列，$\boldsymbol{X}$ のサイズは今回 $n \times (m+1)$ になっていますね (m は水準数)。2 水準のときは 2 列で済んだものが，3 水準になると 4 列になるのはおかしくないですか？そうです，1 つ大事なポイントを忘れていました。各群の値はベースライン β_0 からの相対的な違いです。相対的な違いというのは，言い換えると $\sum \beta = 0$，すなわち全部の水準の和が 0 である必要があるのです。この式は今回の例だと $\beta_1 + \beta_2 + \beta_3 = 0$ であり，これを移項すると明らかなように $\beta_3 = 0 - \beta_1 - \beta_2$ です。つまり総和が決まっているので，自由に大きさを推定できるのは水準数 -1 になります。

これを踏まえてデザイン行列を次のように書き換えることができます。

$$\begin{pmatrix} 3 \\ 3 \\ 4 \\ 4 \\ 6 \\ 7 \\ 8 \\ 9 \\ 7 \\ 6 \\ 5 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \\ e_{11} \\ e_{12} \end{pmatrix}$$

この式は先ほどの式と内容的には同じで，表現の仕方が違うだけですが，$\sum \beta = 0$ の条件がなければ計算結果は算出されません。計算するための制約が少ないと，答えが出なくなるのです。$\sum \beta = 0$ の制約条件を別途書き加えてもいいですが，制約条件も含めた行列の書き方ができるというわけですね。

線形モデルの統一的表現，あるいは行列の計算方法に少しは慣れてきたでしょうか。これがさらに多くの変数，未知数を扱うことになると，その利点はよりはっきりしてきます[*4]。

7.4 因子分析モデルの行列表現

ということで，因子分析モデルの代数的表現ですが，これも行列を使って表現すると非常にシンプルに表現できるということを説明していきましょう。

内容はまったく同じですが，確認しておきましょう。標準得点行列 $\boldsymbol{Z}$ を因子負荷行列 $\boldsymbol{A}$ と因子得点行列 $\boldsymbol{F}$ をつかって，次のように表します。

$$\boldsymbol{Z} = \boldsymbol{F}\boldsymbol{A}' + \boldsymbol{U}\boldsymbol{D} \tag{7.1}$$

ここで，各行列の要素のサイズ感をつかんでおきましょう。まず $\boldsymbol{R}$ というのは相関行列ですから，m 個項目があるのでサイズは $m \times m$ の正方行列になります。次に $\boldsymbol{F}$ ですが，これは因子得点の行列です。得点は人数分ありますから行は n，因子の数が列になるのでこれを p とすると $n \times p$ です。$\boldsymbol{A}$ は因子負荷行列。因子負荷行列は因子の数と項目の数の組み合わせだけあるわけですから，$m \times p$ になりますね。$\boldsymbol{U}$ は独自因子得点です。得点ですから人数分，独自性分は各項目にありますから，サイズとしては $n \times m$ になります。最後に $\boldsymbol{D}$ ですが，これは独自因子の負荷量です。項目の数だけあるのですが，列ベクトルや行ベクトルで表現すると計算の時にサイズが変わって不便なことになります。ですから，対角項に d_j をもつ正方行列 $m \times m$ として表現しています。

サイズを確認したところで，実際に行列計算をしてみましょう。

[*4] ここでは触れませんが，被験者内計画・反復測定になっても行列表現はできます。個人差を表すデザイン行列を別途加えることになります。混合計画になると非常に複雑になりますが，それでも一般的な表現は可能です。

$$
\begin{aligned}
\boldsymbol{R} &= \frac{1}{N}\boldsymbol{Z}'\boldsymbol{Z} && \boldsymbol{Z} \text{ を因子分析のモデル式にして} \\
&= \frac{1}{N}(\boldsymbol{F}\boldsymbol{A}'+\boldsymbol{U}\boldsymbol{D})'(\boldsymbol{F}\boldsymbol{A}'+\boldsymbol{U}\boldsymbol{D}) && \text{前の項の転置を中に入れます} \\
&= \frac{1}{N}\left\{(\boldsymbol{F}\boldsymbol{A}')'+(\boldsymbol{U}\boldsymbol{D})'\right\}(\boldsymbol{F}\boldsymbol{A}'+\boldsymbol{U}\boldsymbol{D}) && \text{転置のカッコを外すときは順番を入れ替えて転置} \\
&= \frac{1}{N}(\boldsymbol{A}\boldsymbol{F}'+\boldsymbol{D}'\boldsymbol{U}')(\boldsymbol{F}\boldsymbol{A}'+\boldsymbol{U}\boldsymbol{D}) \\
&= \frac{1}{N}\boldsymbol{A}\boldsymbol{F}'\boldsymbol{F}\boldsymbol{A}'+\frac{1}{N}\boldsymbol{A}\boldsymbol{F}'\boldsymbol{U}\boldsymbol{D}+\frac{1}{N}\boldsymbol{D}'\boldsymbol{U}'\boldsymbol{F}\boldsymbol{A}'+\frac{1}{N}\boldsymbol{D}'\boldsymbol{U}'\boldsymbol{U}\boldsymbol{D}
\end{aligned}
\tag{7.2}
$$

と，このように展開できました。記号を見ているとわかりにくいので，サイズ感を確認しましょう。最終的には，次のようになっています。

$$
\underset{m\times m}{\boldsymbol{R}} = \frac{1}{N}\underset{m\times pp}{\boldsymbol{A}}\underset{p\times nn}{\boldsymbol{F}'}\underset{n\times pp}{\boldsymbol{F}}\underset{p\times m}{\boldsymbol{A}'}+\frac{1}{N}\underset{m\times pp}{\boldsymbol{A}}\underset{p\times nn}{\boldsymbol{F}'}\underset{n\times mm}{\boldsymbol{U}}\underset{m\times m}{\boldsymbol{D}}+\frac{1}{N}\underset{m\times mm}{\boldsymbol{D}'}\underset{m\times nn}{\boldsymbol{U}'}\underset{n\times pp}{\boldsymbol{F}}\underset{p\times m}{\boldsymbol{A}'}+\frac{1}{N}\underset{m\times mm}{\boldsymbol{D}'}\underset{m\times nn}{\boldsymbol{U}'}\underset{n\times mm}{\boldsymbol{U}}\underset{m\times m}{\boldsymbol{D}}
$$

ここで，要素ごとに計算していた時のことを思い出してください。第二項 $\frac{1}{N}\boldsymbol{A}\boldsymbol{F}'\boldsymbol{U}\boldsymbol{D}'$ と第三項 $\boldsymbol{D}'\boldsymbol{U}'\boldsymbol{F}\boldsymbol{A}'$ の中にある，$\boldsymbol{F}'\boldsymbol{U}$ と $\boldsymbol{U}'\boldsymbol{F}$ のところは，共通因子得点と独自因子得点の積ですし，いずれも標準化されていますから，$\frac{1}{N}$ と合わせて考えると，これは相関係数を表していることになります。また，共通因子と独自因子は相関しませんので，これはイコール $\mathbf{0}$ となり，この 2 つの項が消えてしまうのでした。

また，第一項の $\frac{1}{N}\boldsymbol{F}'\boldsymbol{F}$ は，共通因子同士の相関を表しています。$\boldsymbol{F}'\boldsymbol{F}=\boldsymbol{C}$ とすると，これは因子得点間相関 $\boldsymbol{C}$ を表すことになります。これが直交であると仮定する，つまり他の因子と相関しないと考えると，$\boldsymbol{C}=\boldsymbol{I}$，つまり単位行列です。単位行列は計算に影響を与えませんから，$\boldsymbol{A}\boldsymbol{F}'\boldsymbol{F}\boldsymbol{A}'=\boldsymbol{A}\boldsymbol{C}\boldsymbol{A}'=\boldsymbol{A}\boldsymbol{I}\boldsymbol{A}'=\boldsymbol{A}\boldsymbol{A}'$ となり，この式は簡単に次のように変形できます。

$$
\boldsymbol{R}=\boldsymbol{A}\boldsymbol{A}'+\boldsymbol{D}^2 \tag{7.3}
$$

先ほどの代数的展開を，そのまま行列で表現しただけですが，この方がシンプルに表現できていますね。この表現は，因子分析の第一定理と第二定理の両方を含んで一度に表せているのです。

いかがでしょうか。行列表現の便利さがわかっていただければ，と思います。しかしこれでもまだ謎は残りますね。我々が追っている謎は，行列からどのようにして因子負荷量を計算するのか，です。それを知るためには，もう 1 つ線形代数から明らかになる特徴を知らなければなりません。次回をどうぞお楽しみに。

7.5 課題

■行列計算を確認しておこう $\boldsymbol{V}=(\boldsymbol{I}-\frac{1}{n}\mathbf{1}\mathbf{1}')\boldsymbol{X}$ の要素を書き下してみよう。平均偏差行列ができているでしょうか。

■行列計算を確認しておこう 2 $\boldsymbol{S},\boldsymbol{Z},\boldsymbol{R}$ も，面倒でも要素レベルまで書き下してみよう。

■行列計算を確認しておこう 3 因子分析モデルの行列計算の結果出てくる，$\boldsymbol{R}=\boldsymbol{A}\boldsymbol{A}'+\boldsymbol{D}^2$ の要素を確認し，エレメントワイズで表現していた式との対応を確認しよう。

第 8 章

固有値と固有ベクトルと因子分析モデルの関係

8.1 固有値と固有ベクトル

今回は正方行列にみられるおもしろい特徴である，**固有値 (eigenvalue)** と**固有ベクトル (eigenvector)** についての話から始めます。ある正方行列 $\boldsymbol{A}$, 列ベクトル $\boldsymbol{x}$, スカラー λ が次のような関係にあった時，λ を固有値，$\boldsymbol{x}$ を固有ベクトルと言います。

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$$

一見すると，$\boldsymbol{x}$ が両辺に入っていますから，$\boldsymbol{A}$ が λ に置き換わった等式に見えます。しかし一方は行列で, 他方はスカラーです。こんな奇妙なことが本当にあるのでしょうか？具体的な数値例をみてみましょう。

$$\boldsymbol{A} = \begin{pmatrix} 1 & 6 \\ 2 & 5 \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

を例にします。この時次の関係が成り立ちます。

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} 1 & 6 \\ 2 & 5 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 7 \\ 7 \end{pmatrix} = 7\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 7\boldsymbol{x}$$

確かに成立する組み合わせがありますね。この行列 $\boldsymbol{A}$ に対して，7 が固有値，$(1, 1)$ が固有ベクトルになっています。また, 実はこの行列 $\boldsymbol{A}$ については, -1 も固有値であり, そのときの固有ベクトルは $(-3, 1)$ も固有ベクトルです。

この固有値分解こそ, 因子分析を元とする多変量解析の中心的な数学原理なのです。多変量解析の世界においては, 分散共分散行列やそれを標準化した相関行列など, 変数同士の関係を分析のスタートにおくのでした。これらの行列は正方行列ですから, その固有値や固有ベクトルを計算することで正方行列の特徴を別の視点から分解して考えられるようになります。

8.1.1 固有値の特徴

この固有値の数学的特徴は色々あるのですが, データ分析をする上で重要な点を押さえておきましょう。

固有値の特徴として, **固有値の総和が正方行列の対角要素の総和に合致する**, というのがあります。数式で表現すると, 次のようになります。

$$\sum_{i=1}^{N} \lambda_i = trace(\boldsymbol{A}) = \sum_{i=1}^{N} a_{ii}$$

ここで a_{ij} は行列 $\boldsymbol{A}$ の要素であり，a_{ii} は i 行 i 列目，つまり対角要素です。この正方行列 $\boldsymbol{A}$ のサイズは N で，対角要素の総和をとくに**トレース (trace)** といい $trace(\boldsymbol{A})$ と表します。それが固有値の総和とイコールになる，ということを表しています。サイズ N の正方行列からは固有値が N 個算出できることがわかっており，それをすべて足し合わせたものがトレースと同じになっているのですね。先ほどの例で言えば，$\boldsymbol{A}$ のトレースは $1 + 5 = 6$ で，固有値の総和は $7 - 1 = 6$ であり，確かにこの関係が成立していることがわかります。

分散共分散行列のトレースは，分散の総和を意味します。項目同士の関係を表した行列であれば，分散はその項目から得られる情報の大きさであり，それを総和するということは，その調査研究・項目群から得られる情報の総和であると言ってもいいでしょう。相関行列のトレースは，対角項に入っているのが $r_{ii} = 1.0$ ですから，項目の数と一致します。1 つの項目の情報量を 1.0 に基準化して N 項目分の情報がある，ということを表しています。

固有値と行列の関係は冒頭で示したように，$\boldsymbol{Ax} = \lambda \boldsymbol{x}$ であり，正方行列の特徴をスカラーにしてしまうというものです。得られる N 個の固有値は，元の正方行列のエッセンスをスカラーにして表現しているわけです。

ところで $\boldsymbol{A}$ を n 次正方対称行列，つまり $n \times n$ サイズの対称行列だとすると，n 個の固有値が求められます。これを $\lambda_1, \lambda_2, \cdots, \lambda_n$ として，対応する固有ベクトルを $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ とします。ここで各固有ベクトルのノルムが 1 であるとしましょう。行列と固有値・固有ベクトルの関係から，

$$\boldsymbol{A}\boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i$$

となりますが，このベクトルを並べた行列 $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n \end{pmatrix}$ を考えると，

$$\boldsymbol{AX} = \boldsymbol{X\Lambda}$$

と書くことができます。ここで $\boldsymbol{\Lambda}$ は

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

のような行列です。

この両辺に $\boldsymbol{X}'$ をかけると

$$\boldsymbol{AXX}' = \boldsymbol{X\Lambda X}'$$

となりますが，固有ベクトルの性質とノルムを整えていることから $\boldsymbol{XX}' = \boldsymbol{I}$ であり，そこから

$$\boldsymbol{A} = \boldsymbol{X\Lambda X}'$$

と書くことができます。

ここであらためて要素に注目すると，行列 $\boldsymbol{A}$ が次のように分解されていることがわかります。

$$\boldsymbol{A} = \lambda_1 \boldsymbol{x}_1 \boldsymbol{x}_1' + \lambda_2 \boldsymbol{x}_2 \boldsymbol{x}_2' + \cdots \lambda_m \boldsymbol{x}_m \boldsymbol{x}_m' = \sum \lambda_i \boldsymbol{x}_i \boldsymbol{x}_i'$$

となります。

この分解例は 2×2 の簡単な例で確認しておきましょう。たとえば $\boldsymbol{A} = \left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix} \right) = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ という行列と，対角行列 $\boldsymbol{\Psi} = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}$ があったとして，$\boldsymbol{A\Psi A'}$ の計算をしてみたいと思います。

$$
\begin{aligned}
\boldsymbol{A\Psi A'} &= \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\
&= \begin{pmatrix} \alpha a & \beta c \\ \alpha b & \beta d \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\
&= \begin{pmatrix} \alpha aa + \beta cc & \alpha ab + \beta cd \\ \alpha ab + \beta cd & \alpha bb + \beta dd \end{pmatrix} \\
&= \alpha \begin{pmatrix} aa & ab \\ ab & bb \end{pmatrix} + \beta \begin{pmatrix} cc & cd \\ cd & dd \end{pmatrix} \\
&= \alpha \begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} a & b \end{pmatrix} + \beta \begin{pmatrix} c \\ d \end{pmatrix} \begin{pmatrix} c & d \end{pmatrix}
\end{aligned}
$$

と，このようにスカラーとベクトルの積和の形に書き換えられるのですね[*1]。

さて，これらをまとめて考えると，

1. 固有値分解は行列を列ベクトルとその転置ベクトルの積の形に分解する。
2. 固有値の総和は元の行列の対角要素の総和である。
3. 元の行列の対角要素は各項目の分散を表している。

ということですから，固有ベクトルは全体の情報量をそのままに重要度の大きさに並べ替えたもの，固有値分解は行列をその要素の重要度ごとに分解していくことである，といえます。これこそ**因子分析**で取り出そうとしている因子であり，固有値の大きさはその因子の重要度として，共通次元の判別 (どこまで共通次元とみなすか) に使われるのです。

8.2 固有値と固有ベクトルを求める

ここで少し数学の方に話を戻して，固有値と固有ベクトルの計算方法を考えましょう。元の式を書き換えて次のような方程式を考えます。

$$(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$$

固有ベクトルは $\boldsymbol{x} = \boldsymbol{0}$ すなわち全部ゼロであれば当然成り立ちますから (自明な解といいます)，これは除外することにします ($\boldsymbol{x} \neq \boldsymbol{0}$)。行列の表現は連立方程式の解を求めることと同じなのでした。$\boldsymbol{A} - \lambda \boldsymbol{I}$ を連立方程式の係数行列だと考えれば，それが**逆行列**を持つと左辺にそれをかけてしまえば全部ゼロの答えになってしまいますから，そうでない答えを求めるには，この係数行列が逆行列を持たないことが重要です。

[*1] ただし，この計算が可能なのは分解する元の行列が実対称行列だからです。実対称行列は固有値と固有ベクトルで対角化可能であることが証明できます。実対称行列の固有値は全て実数ですし，固有値が全て実数であれば適当な直交行列をつかって対角化でき，実対称行列の固有ベクトルは互いに独立するのでこれらを使って直交行列を作ることができるからです。これらの性質については線形代数のテキストなどの証明を参照してください。また計算プロセスからもわかるように，同じ要素を持つ列ベクトルと行ベクトルの積ですから，結果は対称になってしまうからです。

さて，この授業の中では説明してきませんでしたが，方程式が解を持つかどうかを決定する計算方法があります。これを**行列式 (determinant)** といい[*2]，この値がゼロでなければその方程式は解を持つ，ということがわかっています。説明しなかったのは，この値を求める計算がとても面倒だからで，詳しくは線形代数のテキストにお任せするとして[*3]，ここでは簡便のために 2×2 方程式の行列について紹介します。

2×2 の係数行列，$\boldsymbol{P} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ の逆行列は次の式で求められることがわかっています。

$$\boldsymbol{P}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

この式から考えると，$ad - bc$ のところが 0 になるとこの計算はできませんから，逆行列が存在しないことになります。この $ad - bc$ にあたるところが行列式であり，$|\boldsymbol{P}|$ とか $det(\boldsymbol{P})$ のように表します。$ad - bc$ が 0 でなければ方程式は解けるのですが，今回の場合は解けると自明になってしまうので困ります。今回は $ad - bc = 0$ でなければならないのです。つまり一般的に書くと次のようになります。

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0$$

$\boldsymbol{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ とすると，この式は次のようになります。

$$\begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = 0$$

$$(a - \lambda)(d - \lambda) - bc = 0$$

この方程式をとくに**固有方程式**といいますが，これを解いてやれば良いことになりますね。具体的に $\begin{pmatrix} 1 & 6 \\ 2 & 5 \end{pmatrix}$ の例で計算してみましょう。

$$\begin{vmatrix} 1 - \lambda & 6 \\ 2 & 5 - \lambda \end{vmatrix} = 0$$

$$(1 - \lambda)(5 - \lambda) - 12 = 0$$

$$\lambda^2 - 6\lambda - 7 = 0$$

$$(\lambda - 7)(\lambda + 1) = 0$$

ここから $\lambda = 7, -1$ が得られますね。

では固有ベクトルはどうなるでしょうか。固有値 7 の例で計算してみます。

$$\begin{pmatrix} 1 & 6 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 7 \begin{pmatrix} x \\ y \end{pmatrix}$$

$$x + 6y = 7x \rightarrow 6x = 6y \rightarrow x = y$$

$$2x + 5y = 7y \rightarrow 2x = xy \rightarrow x = y$$

あれっ？ なんじゃこりゃ？ と思った人もいるかもしれません。でもこれ，間違いではありません。実は固有ベクトルは大きさが定まっておらず，今回の例で言えば $x = y$ つまり $x : y = 1 : 1$ の関係であればあらゆる

[*2] 行列式は数値であり，解が求まるかどうか決定 determinant する，という意味なのに，日本語訳はなぜか「式」といいます。変なの。

[*3] たとえば村上他 (2016) の第 3 章をみてください。

値が成立してしまうのです。固有ベクトルが $(1,1)$ でも $(2,2)$ でも $(100,100)$ でも，$\boldsymbol{Ax}=\lambda\boldsymbol{x}$ の関係に影響しませんから，普通はベクトルの長さ (**ノルム (norm)**) を 1.0 に規格化するという方策が取られます[*4]。先ほど「固有ベクトルを適当な大きさに選んでやれば」というような表現をしましたが，それはベクトルの大きさはいくらでもいいからできることなのですね。

さてここでみたように，2×2 の方程式であれば固有方程式を解くことはできるのですが，行列のサイズがどんどん大きくなると一般的に解けなくなって行くことは想像にかたくないと思います。実際我々は正方行列として，項目の情報が詰まった分散共分散行列とか相関行列を使いますから，それが 2 項目しかないなんてことはなくて，もっともっと大きなサイズになります。そうすると計算機を使って近似的に答えを求めて行くことになります。

8.3 固有値と固有ベクトルの幾何学的意味

固有値，固有ベクトルについて，今度は違う側面から見直してみましょう。

ある正方行列から固有値 λ と固有ベクトル $\boldsymbol{a}$ が得られたとします。このベクトル $\boldsymbol{a}$ のすべての要素を定数 c 倍したベクトル $\boldsymbol{b}=c\boldsymbol{a}$ を考えると，これもやはり同じ関係が成り立ちます。

$$\boldsymbol{Ab}=\lambda c\boldsymbol{a}=\lambda\boldsymbol{b} \tag{8.1}$$

先ほどの計算でも明らかになりましたが，固有ベクトルの値は絶対的なものではなく，要素間の相対的大きさを反映しているに過ぎないのでしたね。

さてこれを幾何学的に，図形として考えてみましょう。要素が 2 つのベクトルは，2 次元座標に表現できます。ベクトル $\boldsymbol{x}=(x,y)$ という座標を表しているというわけです。固有ベクトルも要素が 2 つであれば，座標で表現できます。先ほどの，要素を c 倍しても固有ベクトルとしての性質は変わらない，という話は，「固有ベクトルは大きさに意味はなく，方向を表したもの」ということになります。では何の方向を指し示しているのでしょうか。

固有値と固有ベクトルの話の最初にあった，$\boldsymbol{Ax}=\lambda\boldsymbol{x}$ というのを見直してみましょう。$\boldsymbol{x}$ がなんらかの座標を表していると考えると，それに正方行列をかけるとはどういう意味でしょうか。次の計算式を見てください。

$$\boldsymbol{Ax}=\begin{pmatrix}2&0\\0&3\end{pmatrix}\begin{pmatrix}1\\2\end{pmatrix}=\begin{pmatrix}2\\6\end{pmatrix} \tag{8.2}$$

これをみると，座標 $\boldsymbol{x}=\begin{pmatrix}1\\2\end{pmatrix}$ に $\boldsymbol{A}$ をかけたことで，座標が $\begin{pmatrix}2\\6\end{pmatrix}$ に変わった，と見ることもできますね。このように，ある座標が別の座標に移ることをとくに「変換」と呼びます[*5]。つまり**正方行列はなんらかの変換を施すもの**だ，と考えることができます。

今回の例では，行列 $\boldsymbol{A}$ には次のような性質があります。

$$\begin{pmatrix}2&0\\0&3\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=2\begin{pmatrix}1\\0\end{pmatrix} \tag{8.3}$$

$$\begin{pmatrix}2&0\\0&3\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=3\begin{pmatrix}0\\1\end{pmatrix} \tag{8.4}$$

[*4] ノルムとは，要素の二乗和の平方根，$\sqrt{x_1^2+x_2^2\cdots x_n^2}$ のことです。

[*5] より一般的にいうと，以下のようになります。: 集合 X の各元 x に集合 Y の元 $f(x)$ を対応させる対応 f のことを，集合 X から集合 Y への写像 (mapping)，関数 (function)，あるいは変換 (transformation) という。

そう，お気づきのように，これは固有値・固有ベクトルです。この行列の固有値・固有ベクトルはそれぞれ $\lambda_1 = 2, \boldsymbol{x}_1 = (1, 0)$，$\lambda_2 = 3, \boldsymbol{x}_2 = (0, 1)$ であることがわかります。この固有値，固有ベクトルの組み合わせをじっとみていると，おもしろい特徴がわかって来ます。

今回の行列から得られた 2 つの固有ベクトル，$(1, 0)$ と $(0, 1)$ は，2 次元平面の単位ベクトルと呼ばれるものです。**2 次元座標の任意の点は，これら 2 つのベクトルの任意の線型結合で表現できます**。座標 (a, b) は $a \times (1, 0)$ と $b \times (0, 1)$ からなるベクトルですから，$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$ というように，です。

このように，単位ベクトルは 2 次元世界の基礎となる単位ともいうべきもので，ここで $(1, 0)$ は x 座標の，$(0, 1)$ は y 座標の基盤となるベクトルであるということができます (これをとくに**基底**といいます)。つまり，正方行列 $\boldsymbol{A}$ から得られる固有ベクトルは，その正方行列が作る空間の基盤を明らかにするものであったのです。

では固有値はどうでしょうか？ 今回は x 座標を 2 倍，y 座標を 3 倍に引き延ばす変換をしたわけですが，この座標の歪み (重み) が固有値に対応していますね。つまり，固有値と固有ベクトルは新しい座標に変換する，その変換先の空間的性質を表していることになります。元の座標空間は $(1, 0), (0, 1)$ で作られる空間ですが，変換先の空間は $(2, 0), (0, 3)$ で作られている空間，ということになります。

つまり，正方行列は空間を変換するもの，あるいは正方行列の中に固有ベクトルを基底とした空間があるもの，ということです。

すべての正方行列に，こういった「変換」という解釈ができるのであれば，相関行列にも同様のことがいえるでしょう。相関行列は正方行列ですので，固有値分解できるのです。相関行列を固有値分解することは，相関行列の中に潜む次元 (dimension) を抽出してくることです。固有ベクトル (因子負荷行列) は，正方行列によって変換される，変換先の単位ベクトルのことだったのです。そして，固有値はその次元のゆがみ (重み，重要性) という意味があったのです。

8.4 因子分析の数学的理解

さあ因子分析に戻って考えてみましょう。因子分析のモデルは次のようなものでした。

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{A}' + \boldsymbol{D}^2 \tag{8.5}$$

ここで，左辺の正方行列を相関行列 $\boldsymbol{R}$ とし，$\lambda \boldsymbol{x}\boldsymbol{x}' = \boldsymbol{a}\boldsymbol{a}'$ となるようにベクトルの大きさを整えてみましょう。これは λ を分解してベクトルの中に溶け込ませるようなものですから，$\boldsymbol{a} = \sqrt{\lambda}\boldsymbol{x}$ とすればよいでしょう。すると相関行列は次のように分解できます。

$$\boldsymbol{R} = \boldsymbol{a}_1\boldsymbol{a}_1' + \boldsymbol{a}_2\boldsymbol{a}_2' + \cdots + \boldsymbol{a}_m\boldsymbol{a}_m' + \boldsymbol{d}\boldsymbol{d}' \tag{8.6}$$

ここでは共通因子の数が m 個だとわかっている体で分解していますが，基本的にはサイズ N の行列からは N 個の固有値がずらーっと並ぶわけです。それをどこかで「共通しているのはここまで」と判断し，残りは誤差であるとしてまとめて $\boldsymbol{d}\boldsymbol{d}'$ にしているだけですね。このように数学的にはここからが共通因子，ここからが独自因子といった区別をすることなく，最後のひとかけらまで固有値分解を行なっているのですが，その次元の重要度でもって共通因子と誤差因子に (研究者が恣意的に) 分割しているのが因子分析のやっていることなのです。

一般に，N よりも m のほうがグッと少なくなります。たとえば YG 性格検査では $N = 120$ であり，m はせいぜい 5 から 10 数個です。120 項目のつくる 120 次元空間の中で，そこに働きかけても方向の変わらない基礎的な少数の次元にのみ注目すれば，効率よく情報圧縮ができるというものです。

相関係数を固有値分解すると，その固有値はすべて足し合わせるとサイズ N になるのでした。元のデータから計算される相関行列は，1 つの項目が一単位分の情報を持っていると考えますが，固有値分解はそれを次元の重要性順に並べ替えます。固有値は項目いくつ分の重要度があるかということを表す指標だと考えることができます。どこから共通因子でどこからが誤差か，ということを考えるときに，たとえば固有値が 1.0 よりも小さくなるようであれば，項目 1 つ分の情報もないのだからということで誤差因子だと判断することがあります。

ところで因子分析モデルの $\boldsymbol{A}$，因子負荷行列ですが，これは共通因子の固有ベクトルをセットで扱ったものです。つまり，$\boldsymbol{A} = \boldsymbol{a_1}, \boldsymbol{a_2}, \cdots, \boldsymbol{a_m}$ と縦ベクトルを並べたものになっています。エレメントワイズで表現すると次のようになります。

$$\boldsymbol{A} = \left(\begin{pmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1N} \end{pmatrix}, \begin{pmatrix} a_{21} \\ a_{22} \\ \vdots \\ a_{2N} \end{pmatrix}, \cdots, \begin{pmatrix} a_{m1} \\ a_{m2} \\ \vdots \\ a_{mN} \end{pmatrix}, \right)$$

ここで $\boldsymbol{AA'}$ の間に単位行列 $\boldsymbol{I}$ を挟んでも，別に結果は変わりませんよね。単位行列はかけても変わらないのが特徴ですから，$\boldsymbol{AA'} = \boldsymbol{AIA}$ です。ここでの $\boldsymbol{I}$ のサイズは $m \times m$ であることに注意しつつ聞いてください。

$\boldsymbol{I} = \boldsymbol{TT'} = \boldsymbol{T'T}$ になるような m 次の行列を使うと，$\boldsymbol{AA'} = \boldsymbol{ATT'A'}$ の関係は保たれたままです。この $\boldsymbol{T}$ はこれまた行列の座標を変えてしまう変換行列であり，これを挟むことができるということは**因子負荷量の値はなんでもあり**だということになってしまいます。この変換行列 $\boldsymbol{T}$ はとくに**回転行列 (rotation matrix)** とも呼ばれますが，因子分析はこのようにどんな回転でもできる，**回転に関する不定性**があるのです。あらゆる因子負荷量の組み合わせがあり得る，というのは困りものなので，なんらかの形で因子負荷行列 $\boldsymbol{A}$ に制約をかける必要があります。言い方を変えれば，色々な制約の中ではあっても好きな値を取ることができますから，ユーザにとって便利な基準を考えてやれば良いでしょう。因子分析では一般に，**因子軸の回転**を行いますが，それはこうした理由からです。

ちなみに $\boldsymbol{TT'}$ に $diag(\boldsymbol{T\Phi T'}) = \boldsymbol{I}_m$ という制約のある行列 $\boldsymbol{\Phi}$ を挟んでやっても，元の計算モデルに影響はありません。この $\boldsymbol{\Phi}$ は**因子間相関 (factor correlations)** とよばれ，相関を持った因子軸の回転，すなわち**斜交回転 (oblique rotation)** も色々なものが考えられています[*6]。

ずいぶんややこしい話になってきました。この後は実践形式で因子分析を理解していければと思います。

8.5 課題

■固有値問題 行列 $\begin{pmatrix} 8 & 1 \\ 4 & 5 \end{pmatrix}$ の固有値･固有ベクトルを求めよ。

[*6] これに対して $\boldsymbol{TT'} = \boldsymbol{I}$ のような因子間相関がない (単位行列) な回転を**直交回転 (orthogonal rotation)** といいます。

第 9 章

R をつかっての行列計算

9.1 R による行列計算

今回は統計環境 R をつかって，演習によってこれまで習ったことを整理していきましょう。

9.1.1 環境の準備 (確認)

まずは環境の準備です。すでに R や RStudio のインストールは終わっているものとして話を進めます[*1]。次の 3 つのステップをたどって，実行の準備をしてください。

■RStudio の起動 RStudio を起動してください。パッケージのインストールをするときなど，管理人権限が必要になるケースがあります。

■プロジェクトを開く RStudio を使う時の基本は，プロジェクトによる管理です。分析するデータ，テーマ，内容によってプロジェクトを切り替えながら使います。メニューバーから `File > New Project` と進んでください。すると「New Directory」「Existing Directory」「Version Control」の選択肢が出てきます。New Directory は新しいディレクトリ (=フォルダ) を作ってそこで関連ファイルをまとめますよ，という意味です。Existing Directory はすでに存在するディレクトリ (=フォルダ) をまとめる場所にしますよ，という意味です。みなさんの環境に応じて使い分けてください。すでにプロジェクトが存在する場合は，プロジェクトを開く (FIle→Open Project) から選択します。RStudio の右上などで，プロジェクトが開いていることを確認しましょう。

■R スクリプトを開く 今日のコードを書くための R スクリプトファイルを準備します。File > NewFile > R Script と進み，何も書いてない R スクリプトの画面を表示させてください。真っ白いファイルが開いたら問題ありません。

これでスクリプトのところにコードを書いていく準備ができました。

9.1.2 R におけるデータの型

統計環境 R は Excel や Numbers や Calcs など[*2]の表計算ソフトとは違って，データを行列・ベクトルとして扱うことができます。たとえば次のコードを実行してみてください。

[*1] まだだよー！という人は RStudio インストールといったキーワードで検索すると，親切にも案内してくれているサイトに色々出会えるでしょう。

[*2] Excel は Microsoft Office に含まれる表計算ソフト。Numbers は Apple の Mac や iPad，iPhone で使える表計算ソフト，Calcs は Libre Office というフリーソフトウェアの表計算ソフトです。

code : 9.1 ベクトルデータの保持

```
1 A <- 1:9
2 B <- c(3, 4, 5)
3 C <- matrix(A, ncol = 3)
4 D <- matrix(A, ncol = 3, byrow = T)
```

■コード解説

1 行目 1:9 をオブジェクト A に代入しています。ここでコロン: は連続した数字を表しており，c(1,2,3,4,5,6,7,8,9) と同じ意味です。

2 行目 要素 3, 4, 5 からなるベクトルを作って B に代入しています。c() は結合する (combine) という関数です。

3 行目 9 つの要素があるベクトルを持ったオブジェクト A を matrix 型に変更しています。その時の列数は 3(ncol=3) です。

4 行目 同じくオブジェクト A を matrix 型に変換，列数は 3 ですが byrow = T で「行ごとに並べる」スイッチをオン (TRUE) にしています。

それぞれのオブジェクトに格納されているものを確認してみましょう。何がどうなっているかわかるでしょうか (出力 9.1)。

R の出力 9.1: R におけるベクトルと行列

```
> A
[1] 1 2 3 4 5 6 7 8 9
> B
[1] 3 4 5
> C
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> D
    [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

R のオブジェクトには色々な型がありますが，基本はベクトルです。数字のセットとして扱うのです。このときの R のベクトルに行・列の区別はありません。行列であることを明示するためには，matrix という関数を当てはめることになります。ベクトルに行，列の意味を持たせたければ，matrix 型で一列あるいは一行の行列である，という指定をしてください。

これまで，あるいは他の授業で学んだことのある R のオブジェクトの形としては，list 型が多かったのではないかと思います。list という形は，要素がベクトルでも数字でも文字列でもなんでもかまわない，というものでした。これを矩形 (長方形) に整え，変数名としての列名や，行番号としての行名をもっているものが data.frame 型です。data.frame 型は list 型の特殊系なわけです。matrix 型は矩形のオブジェクトではありますが，data.frame 型とは違います。matrix 型にも列名，行名をつけることはできますが，行列

であると明示してあるように，計算するときは行列演算の対象になるのです。data.frame 型は行列の四則演算はできません。

また一点注意が必要なこととして，R はベクトルの再利用をするということです。次の一行 (code:9.2) を実行して中身を見てみましょう。

code : 9.2 ベクトルデータの保持

```
1       E <- matrix(A, ncol = 3, nrow = 6)
2       G <- matrix(A, ncol = 3, nrow = 4)
```

■コード解説

1 行目 1:9 をオブジェクト E に代入している。ここで列数は 3, 行数は 6 を指定している。

2 行目 1:9 をオブジェクト G に代入している。ここで列数は 3, 行数は 4 を指定している。

R の出力 9.2: ベクトルの再利用

```
> E <- matrix(A, ncol = 3, nrow = 6)
> G <- matrix(A, ncol = 3, nrow = 4)
 警告メッセージ:
 matrix(A, ncol = 3, nrow = 4) で:
   データ長 [9] が行数 [4] を整数で割った，もしくは掛けた値ではありません
> E
     [,1] [,2] [,3]
[1,]    1    7    4
[2,]    2    8    5
[3,]    3    9    6
[4,]    4    1    7
[5,]    5    2    8
[6,]    6    3    9
> G
             [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6    1
[3,]    3    7    2
[4,]    4    8    3
```

オブジェクト A の要素数は 9 です。しかし行列 $\boldsymbol{E}$ を作る時に，3 行 6 列，すなわち 18 の要素が必要であることを要求しました。そうすると R は (勝手に and/or 親切にも？)A を再利用して足りないところを埋めてしまいます。今回はちょうど 2 回使えば埋まりましたので，エラーや警告もなくそのまま通っています。つくられた $\boldsymbol{E}$ の要素を確認し，再利用されていることをみてください。ちなみに $\boldsymbol{G}$ は 3 行 4 列，すなわち 12 の要素が必要であることを要求しました。そうすると R は警告を出して，再利用が途中で途切れますよと言ってくれます。とはいえ警告に過ぎないのでそのまま計算を続けることができ，作られた $\boldsymbol{G}$ には $1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3$ と 2 周目の最初の 3 要素だけ使われています[*3]。

このように R は，可能な時には良かれと思って勝手に再利用してしまいますから，注意してくださいね。

[*3] そんなことよりオブジェクトが A,B,C ときてるんだから E のつぎは F だろ，と思う人もいるかもしれません。F でもいいのですが，実は大文字の F と T はそれぞれ FALSE/TRUE の略語であり予約後語です。上書きして使うこともできますが，なるべく避けたほうがいいでしょう。ちなみに RStudio などでこれら一文字を使うと予約語になっているので表示の時にハイライトされます。

9.1.3 行列方と行列の演算

それでは R を使っての行列計算を進めましょう。

加法減法はサイズが同じでないとできない，という制約はありますが，計算記号などに違いはありません。乗法は，スカラーとベクトル，スカラーと行列であれば普通に記述していただいて結構です。

code : 9.3 ベクトルの和･差，スカラーをかける

```
1  x <- 1:3
2  y <- 8:10
3  x + y
4  x - y
5  2 * x
6  y / 3
7  A <- matrix(c(1, 2, 3, 4), ncol = 2)
8  B <- matrix(c(5, 6, 7, 8), ncol = 2)
9  A + B
10 A * 3 + B * 2
```

■コード解説

1 行目 1:3 をオブジェクト x に代入している。

2 行目 1:9 をオブジェクト y に代入している。

3 行目 ベクトルの和 $(\boldsymbol{x}+\boldsymbol{y})$

4 行目 ベクトルの差 $(\boldsymbol{x}-\boldsymbol{y})$

5 行目 ベクトルにスカラーをかける $(2\boldsymbol{x})$

6 行目 ベクトルをスカラーで割る。スカラーの逆数をかけることと同じ。$(\boldsymbol{y}/3)$

7 行目 行列 $\boldsymbol{A}$ をつくる。サイズは 2×2

8 行目 行列 $\boldsymbol{B}$ をつくる。サイズは 2×2

9 行目 行列の和 $(\boldsymbol{A}+\boldsymbol{B})$

10 行目 行列にスカラーをかけ，それを足し合わせる。スカラー倍は各要素にかかる $(3\boldsymbol{A}+2\boldsymbol{B})$

出力結果はここでは提示しませんが，皆さんそれぞれ計算できているか確認しておいてください。

続いて行列の乗法ですが，サイズが変わるような行列としての計算の場合はとくに %*% という記号を使うことになります。ただのアスタリスク*ではなく，それを % で囲むことで行列の積になっていることを表現しているのです。また行列の行列を反転させるのは，**転置**という操作になりますが，これは t() という関数を使います。行列の積を計算する例を code:9.4 に示しました。計算結果が出力されますので，どういう計算をしているか一行ずつ確認しながら進めてください。

code : 9.4 行列の積

```
1  a <- c(1, 2, 1)
2  b <- c(3, 4, 2)
3  a * b
4  a %*% b
5  a %*% t(b)
6  A <- matrix(1:9, ncol = 3)
7  A * a
```

```
 8  A %*% a
 9  a %*% A
10  B <- matrix(1:6, nrow = 3, byrow = T)
11  C <- matrix(c(1, 0, 0, 1, 1, 1), ncol = 3)
12  B %*% C
13  B %*% t(C)
```

■コード解説

1 行目 ベクトル $\boldsymbol{a} = (1, 2, 1)$ を作る

2 行目 ベクトル $\boldsymbol{b} = (3, 4, 2)$ を作る

3 行目 掛け算記号*を使っているが, これはベクトルの掛け算ではなく要素ごとの掛け算を意味する。

4 行目 ベクトルの掛け算記号 %*% を使っており, $\boldsymbol{ab}$ の計算をしている。この時, ベクトル $\boldsymbol{a}$ は行ベクトル, $\boldsymbol{b}$ は列ベクトルと解釈されます。行列計算に適した形であり, デフォルトでは行方向が優位です。

5 行目 ベクトルの掛け算だが t() で転置, すなわち $\boldsymbol{b}'$ を行っており, 行列に適した形に変換されて $\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 3 & 4 & 2 \end{pmatrix}$ と解釈されている。3×1 と 1×3 の計算なのでできあがる行列は 3×3 のサイズになる。

6 行目 行列 $\boldsymbol{A}$ を作る

7 行目 行列 $\boldsymbol{A}$ に対してベクトルをかけているように見えるが, ベクトルの掛け算でなく要素の掛け算記号*なので, 各行を 1 倍, 2 倍, 1 倍する結果になっている。

8 行目 行列とベクトルの積 $\boldsymbol{Aa}$ であり, 行列計算可能な 3×3 と 3×1 と解釈され, 結果のサイズは 3×1 になっている。

9 行目 ベクトルと行列の積 $\boldsymbol{aA}$ であり, 行列計算可能な 1×3 と 3×3 と解釈され, 結果のサイズは 1×3 になっている。

10-11 行目 行列 $\boldsymbol{B}, \boldsymbol{C}$ を作る

12 行目 行列の積 $\boldsymbol{BC}$ を計算している。3×2 と 2×3 の行列の積なので, 結果のサイズは 3×3 になっている。

13 行目 行列の積 $\boldsymbol{BC'}$ を計算しようとしているが, 3×2 と 3×2 の積は計算できないので, エラーが返ってくる。

続いて**逆行列**の例を見てみましょう。逆行列の計算は, 手計算でやると大変なのですが, R では関数 solve()) を使うと計算できます。

code : 9.5 逆行列の計算

```
1  A <- matrix(c(2,1,5,3),ncol=2)
2  solve(A)
3  A %*% solve(A)
```

■コード解説

1 行目 行列 $\boldsymbol{A}$ をつくる

2 行目 逆行列 $\boldsymbol{A}^{-1}$ の計算

3 行目 $\boldsymbol{AA^{-1}} = \boldsymbol{I}$ より, 逆行列になっていたことの確認。

逆行列はかけると単位行列になるので，スカラーでいうところの逆数を意味します。逆数をかけると 1 になるように，行列の場合は逆行列をかけると単位行列 $\boldsymbol{I}$ になるのでした。これの何が嬉しいかというと，行列が連立方程式の係数を表していた場合，逆行列をかけることで連立方程式が解けるのでした (セクション 6.3, Pp.69 参照)。次の連立方程式を使って，実際に計算してみましょう。

$$\begin{cases} x - 2y - 5z &= 3 \\ 5x + 4y + 3z &= 1 \\ 3x + y - 3z &= 6 \end{cases}$$

code : 9.6 連立方程式を解く

```
1  A <- matrix(c(1,5,3,-2,4,1,-5,3,-3),ncol=3)
2  b <- c(3,1,6)
3  solve(A) %*% b
```

1 行目 係数行列 $\boldsymbol{A}$ をつくる

2 行目 右辺のベクトルを作る

3 行目 $\boldsymbol{Ax} = \boldsymbol{b}$ から $\boldsymbol{A}^{-1}\boldsymbol{Ax} = \boldsymbol{A}^{-1}\boldsymbol{b}$ より，$\boldsymbol{x} = \boldsymbol{A}^{-1}b$ として方程式の解を求める

元の方程式との対応を確認しながら，「行列ってすごい，便利！」と思っていただければ幸いです。

9.2 データの行列表現

さて連立方程式が解けるのは嬉しいのですが，データ解析に直結するかと言われたら少し違いますね。データの記述統計量など，数的処理をする時に行列を使う便利さを体感してみましょう。

表 9.1 投手のデータ

Name	team	height	weight	salary	Win	Save
菅野 智之	Giants	186	92	65000	14	0
西 勇輝	Tigers	181	82	20000	11	0
秋山 拓巳	Tigers	188	101	3200	11	0
大野 雄大	Dragons	183	83	13000	11	0
石川 柊太	Softbank	185	86	4800	11	0
千賀 滉大	Softbank	187	90	30000	11	0
涌井 秀章	Eagles	185	85	12500	11	0
森下 暢仁	Carp	180	76	1600	10	0
大貫 晋一	DeNA	181	73	2500	10	0
小川 泰弘	Swallows	171	80	9000	10	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

表 9.1 に示したのは，データ解析基礎でも扱った 2000 年代の野球選手のデータです[*4]。それをプロジェクトフォルダに置き，code:9.6 のコードを実行してください。

[*4] 伴走サイト https://kosugitti.github.io/psychometrics_syllabus/ より，サンプルデータにある「野球選手のデータ 10 年度分」を選んでください。ファイルは UTF-8 形式で保存されています。

code : 9.7 データファイルの読み込み

```
1  library(tidyverse)
2  dataset <- read_csv("baseballDecade.csv") %>%
3    dplyr::filter(Year=="2020年度") %>%
4    dplyr::filter(position == "投手") %>%
5    dplyr::select(Name, team, height, weight, salary, Win, Save) %>%
6    na.omit() %>%
7    arrange(-Win) %>%
8    select(height, weight, salary) %>%
9    as.matrix()
```

1 行目 読み込み他, ファイル操作を便利にするパッケージ tidyverse を読み込みます[*5]。

2-9 行目 データファイルを変形し, dataset オブジェクトに代入しています。ここで %>% はパイプ演算子と呼ばれるもので, 左の操作を右の操作へつなげる役割をします。

3 行目 データを 2020 年度のものだけに絞るため, dplyr パッケージの filter 関数を適用しています。

4 行目 データを投手のものだけに絞るため, dplyr パッケージの filter 関数を適用しています。

5 行目 変数を必要なものだけに絞るため, dplyr パッケージの select 関数を適用しています。

6 行目 データセットから欠測値を除いています。

7 行目 表示用に, データを勝利数 (変数 Win) の多い順に並べ直しています。

8 行目 以下の計算に使う変数だけに絞り込んでいます。

9 行目 データセットを行列方に変換しています。

ともかくこれで dataset オブジェクトは行列型の, 335 行 3 列の行列になっています。今から行う計算は, 第 7 講でやったように, データ行列 $\boldsymbol{X}$ から, 平均ベクトル $\boldsymbol{m}$, 平均偏差行列 $\boldsymbol{V}$, 分散共分散行列 $\boldsymbol{S}$, 標準偏差を対角に持つ行列 $\boldsymbol{Q}$, 標準化スコア行列 $\boldsymbol{Z}$, 相関行列 $\boldsymbol{R}$ を作る, という手順です。数式とコードを示しますので, 確認しながら進めてください。

$$\boldsymbol{m} = \frac{1}{n}\boldsymbol{X}'\boldsymbol{1} \tag{9.1}$$

$$\boldsymbol{V} = \boldsymbol{X} - \boldsymbol{1}\boldsymbol{m}' \tag{9.2}$$

$$\boldsymbol{S} = \frac{1}{n}\boldsymbol{V}'\boldsymbol{V} \tag{9.3}$$

$$\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{Q}^{-\boldsymbol{1}} \tag{9.4}$$

$$\boldsymbol{R} = \frac{1}{n}\boldsymbol{Z}'\boldsymbol{Z} \tag{9.5}$$

code : 9.8 データの行列演算

```
1  n <- nrow(dataset)
2  one <- rep(1, n)
3  m <- t(dataset) %*% one / n
4  V <- dataset - one %*% t(m)
5  S <- t(V) %*% V / n
```

[*5] tidyverse パッケージがない人はインストールしてください。関連パッケージをいろいろまとめたパッケージ群パッケージなので, 少し時間がかかります。

```
 6  SD <- diag(S) %>% sqrt()
 7  Q <- diag(SD)
 8  Z <- V %*% solve(Q)
 9  R <- t(Z) %*% Z / n
10  cor(dataset)
```

1 行目 行数をオブジェクト n に入れる。行数を数える関数が nrow です。

2 行目 データセットの列数と同じ長さの，1 だけを繰り返し入れたベクトルを作ります。繰り返しは rep 関数を使います。

3 行目 平均ベクトル $\boldsymbol{m}$ を作る操作です。数式 9.1 に対応する操作です。

4 行目 平均ベクトルを使って平均偏差行列 $\boldsymbol{V}$ を作る操作です。数式 9.2 に対応しています。

5 行目 分散共分散行列 $\boldsymbol{S}$ を作る操作です。数式 9.3 に対応しています。

6 行目 diag 関数を使って，行列 $\boldsymbol{S}$ の対角要素を抜き出し，その平方根を取り出しています。SD は標準偏差が入ったベクトルになります。

7 行目 ベクトルを diga 関数にいれると，ベクトルの要素を対角に持つ正方行列ができます。それを $\boldsymbol{Q}$ として表現しています。

8 行目 標準スコア行列 $\boldsymbol{Z}$ を作る操作です。平均偏差行列 $\boldsymbol{V}$ に $\boldsymbol{Q}^{-1}$ をかけています。数式 9.4 に対応しています。

9 行目 相関行列 $\boldsymbol{R}$ を作る操作です。数式 9.5 に対応しています。

10 行目 できあがった行列 $\boldsymbol{R}$ を検算するために，R の持っている相関行列を作る関数 cor を実行しています。作った $\boldsymbol{R}$ と同じになっていることを確認してください。

この一連のスクリプトは，行列のサイズが変わっても適用できます。1 つの式表現で，どんなデータサイズでも一般的に表現できているところが行列表記のすごいところです。最終的に cor 関数があるのなら，こんな手間をかけなくてもいいじゃないかと思うかもしれません。しかし理屈を知っていて使うことと，理屈を知らずに使うことは違いますからね。

9.3 R による固有値計算

つづいて行列計算のとくにおもしろいところ，固有値計算をしたいと思います。

固有値の計算は，小さなサイズであれば手計算でもできますが，大きなサイズになると n 次連立方程式を解くことになるのでとても大変です。解の公式で解くということもできませんので，近似計算手続きが必要です。その方法としてパワー法，ヤコビ法，ハウスホルダー法などいろいろ考えられていますが，数値計算の細かい理論に入っていくのは少し寄り道が過ぎますので割愛して，R の関数を使って計算しましょう。

先ほど計算した相関行列 $\boldsymbol{R}$ を使ってこれを確認します。

code : 9.9 相関行列の固有値分解

```
1       eig <- eigen(R)
2       eig$values
3       sum(eig$values)
4       sum(diag(R))
5       eig$vector
6       eig$vector[,1]^2 %>% sum
```

固有値分解をする関数は eigen です。計算結果を eig オブジェクトに入れ，固有値 values と固有ベク

トル vector を表示させてみました (出力 9.3)。固有値が大きい方から $1.7043160, 0.9486112, 0.3470728$ となっています。ここでは変数が 3 つあって，それぞれの情報の大きさは相関行列で標準化されていて 1.0 です。この行列がもっている 3.0 の分散を，$1.7 : 0.9 : 0.3$ に再分配したことになります。ちなみに固有値の総和 sum(eig$values) は，行列のトレース sum(diag(R)) と等しくなっていることが確認できます。

また，固有ベクトルはそれぞれの固有値に対して計算されます。各列が各固有値に対応しており，第一固有値 1.7 に対応する固有ベクトルは $(0.68, 0.68, 0.26)$ です。固有ベクトルは向きだけあって大きさがありませんから，この数字はノルムすなわち二乗和したものが 1 になるようにして算出されているのがわかります。

R の出力 9.3: 固有値と固有ベクトル

```
> eig <- eigen(R)
> eig$values
[1] 1.7043160 0.9486112 0.3470728
> sum(eig$values)
[1] 3
> sum(diag(R))
[1] 3
> eig$vector
           [,1]       [,2]        [,3]
[1,] 0.6839162 -0.1734073  0.70865257
[2,] 0.6812174 -0.1959156 -0.70537929
[3,] 0.2611541  0.9651668 -0.01586178
> eig$vector[,1]^2 %>% sum
[1] 1
```

このような数値計算を駆使して，因子分析や回帰分析が実際に計算されているのですね。当然ですが，手計算よりも機械で計算させた方が圧倒的に早いですね。計算機の発展スピードのおかげで，今は何万，何十万というデータでも扱うことができる用意なりました。またデータのサイズが変わってもスクリプトを変える必要がありません。

こうした原理を知っておくと，ビッグデータなど恐るるに足らず，というところではないでしょうか。

9.4 課題

■線形代数の練習問題 行列 $\boldsymbol{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} 3 & 1 & 5 \\ 2 & 4 & 6 \end{pmatrix}, \boldsymbol{C} = \begin{pmatrix} 4 & 1 \\ 2 & 3 \end{pmatrix}$, 列ベクトル $\boldsymbol{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$, 行ベクトル $\boldsymbol{y} = \begin{pmatrix} 2 & 8 \end{pmatrix}$ とするとき，次の計算を R で実行するコードを書きなさい。なお，計算できないものについては「計算できない」としてコードは書かないこと。

1. $\boldsymbol{A} + \boldsymbol{B}$
2. $\boldsymbol{A} - \boldsymbol{C}$
3. $\boldsymbol{AB}$
4. $\boldsymbol{AC}$
5. $\boldsymbol{B}'\boldsymbol{A}$
6. $\boldsymbol{Ay}'$
7. $\boldsymbol{xy}$

2047 8. $\boldsymbol{xB}$
2048 9. $\boldsymbol{x}'\boldsymbol{B}'$
2049 10. $\boldsymbol{yx}$

2050 **■連立方程式を解く** 次の連立方程式を R で解きなさい。

$$\begin{cases} x - 2y + 3z = & 1 \\ 3x + y - 5z = & -4 \\ -2x + 6y - 9z = & -2 \end{cases}$$

第 10 章

R をつかった因子分析と尺度作成法

前回に続いて，R をつかった演習を進めましょう。今回は R を使っての因子分析および尺度作成です。

10.1 調査研究の手順

まずは心理尺度作成の手順を大まかに理解しておきましょう。

■構成概念の設定 心理尺度を作り始めるにあたって，まずは何を測りたいのかを明確にする必要があります。**構成概念妥当性 (Construct Varidity)** についてのそもそも論ですね。ある概念を測定したいとして，そういう概念が本当に存在するのか，どこの誰がどのように持つ概念なのかを明確にする必要があります。たとえば文部科学省がスローガンのように掲げる**生きる力**というのを測ってみたいと思っても，それは何なのかわかりません。死んでなければ生きているのですから，生きる力はありそうに思えます。でもそういうことじゃないらしい。じゃあどういうことなのか，ということを考えて行かなければなりません。あるいは，みなさんは**いちびり**という関西弁をご存知ですか。いたずらっ子，わざと変なことをして注目を引きたい子，のような意味なのですが，このいちびり感は関西の人間にしかわからない感覚かもしれません。であれば調査対象者が限定的な，一般的ではない概念ですから，因子分析の行う多くの人の反応から共通成分を抜き出すという考え方には適さない概念です。より一般的なものに考え直す必要があるでしょう。

ほかにも，心理学的な態度なのか，パーソナリティのような行動の傾向なのか，抑うつ傾向のように心身両方の問題なのか，そしてそれらが紙とペンで測定可能なものなのかどうか，改めて考えましょう。何を当たり前のことを，と思うかもしれませんが，意外とおかしな問題設定に入り込んでないとも限らないのです。また，心理尺度や心理学的な概念はすでにさまざまなものが存在します。これらを見比べて，自分の測定したい概念が他の概念とどのように同じで，どのように違うのかを論理的に考えられなければなりません。どこかにある尺度とほとんど同じであればオリジナリティが存在しないだけでなく，車輪の再発明というムダな努力をすることになります。完全にオリジナルなものを考えるのは難しいかもしれませんが，新しい概念を使うことによってこれまでの概念で説明できたことを含み，さらにこれまでの概念で説明できなかったことも説明できるようになる，という利点が必要です。これらは**弁別的妥当性 (distincive validity)** にも関わる問題です。

■項目の選出 測定したい概念が明確になれば，これに関わる項目を作り出す必要があります。ある態度を強く持っている人はどのような振る舞いをし，どのような振る舞いをしないのか。どのような意見に賛成し，どのような意見には反対するのか，といったことをさまざまな角度から検証します。「目に見えないものを測定する」のが心理測定であり，質問紙調査というのはこの見えない的に向かって項目という矢を射掛けるようなものです。矢の数はなるべく多く，また人は嘘をついたり間違えたりする生き物ですから多角的に聞くことで，捉えるべき本質に迫っていく必要があります。言葉を使って聞くわけですから，古すぎる・新しすぎる表現は避

け，専門用語は使わずなるべく平易な言葉で聞く必要があります。ワーディング (言い回し) についても細部まで注意しましょう[*1]。

■予備調査の実施 ある程度項目が集まれば，予備調査を行います。用意した項目の 5 倍から 10 倍の人を集めるのが良い，と一般的に言われています Grimm・Yarnold (1994 小杉他 訳 2016)。ここでの予備調査項目は，十分考えられたものではあると思いますが，分析してみると意外と不適切な項目というのが出てくるものですので，多めに用意されていることでしょう。必然的に，予備調査の対象サイズも百人以上の大きなものになるのが一般的です。

■探索的因子分析 この後詳しく説明しますが，ここが尺度作成に関わる分析のメインです。ここで因子数を決める (あるいは確認する) ことになり，因子的な妥当性みたり，不適切な項目は除外したりします。別の聞き方の方が良いような項目があれば，項目の入れ替えを行なって再調査することもあります。この探索的な因子分析と予備調査を繰り返して，何度行っても同じ因子構造になり，同じ項目が同じ因子に含まれるというのが確認できれば，項目セットは完成すると言えるでしょう。

■項目反応理論 ある因子に関連する項目とそのデータセットを抜き出して，項目反応理論 (段階反応モデル) を実行しましょう。なぜ一部を抜き出すかというと，項目反応理論モデルは 1 次元性を仮定したモデルであることが基本だからです。これを多次元に展開した多次元段階反応モデルもありますので，直接そちらを使っても構いません。ともかくこれを行うことで，反応段階数を確認でき，また項目情報曲線やテスト上表曲線を書くことで，この尺度がどのような領域を得意とする項目群からなるのかを記述できることになります。これに関しては次回より詳しく説明します。

■本調査へ 最終的に項目群が確定したら，大規模な調査を行ってテストの標準化を目指します。因子構造などはほぼ確定しているはずですから，このテストを使うとどのようなデータの散らばりが得られるのかを確定するわけです。この最後のステップの目的は標準化，つまりこのテストで測定される人の平均や散らばり，分布の形状を確認することにあります。使うときに逐一分析しなくてもいいように，項目回答パターンがどれぐらいであれば，全体のどのあたりに位置するかといったプロフィールが作れるとなおいいでしょう。

以上が大体の流れになります。ここにあるように，予備調査を何度も繰り返して因子構造を確定し，そこから本調査を行うことで，ほぼこの尺度はどれぐらいの点数で分布するか，といったことがわかるようになります。かつては因子分析を 1 回実行するだけでも多大な時間がかかりましたから，心理尺度を作るというのはそれだけでライフワークになるような作業でした。幸い最近は分析のスピードが飛躍的に向上しましたので，簡単に分析してやり直すということができます。しかしだからといって，構成概念妥当性を疎かにしてはいけませんし，「やったらこうなった」というようなやり逃げ研究になるのではなく，使うための尺度を作るのだということは忘れてはいけないでしょう。

10.2 共通性の推定

それでは実際にデータを使ってのやり方を説明します。改めて，因子分析モデルについて確認しておきましょう。正方行列を相関行列 $\boldsymbol{R}$ とし，$\lambda_1 \boldsymbol{x}\boldsymbol{x}' = \boldsymbol{a}\boldsymbol{a}'$ となるようにベクトルの大きさが整えられているとすれば，次のように表現できるのでした。

$$\boldsymbol{R} = \boldsymbol{a}_1\boldsymbol{a}_1' + \boldsymbol{a}_2\boldsymbol{a}_2' + \cdots + \boldsymbol{a}_m\boldsymbol{a}_m' + \boldsymbol{d}\boldsymbol{d}' \tag{10.1}$$

[*1] たとえば「テレビやラジオをよく見聞きしますか」という質問は悪い質問です。テレビしか見ない人，ラジオしか聞かない人がどう答えていいか悩むからです。これはダブルバーレルと呼ばれる悪手ですが，ほかにも色々ありますので調査法の専門書を参考にしてみてください。

このことから，相関行列を固有値分解すれば共通因子負荷量が得られることがわかります。しかしこれには 1 つ問題があります。この式の最後の項目，$\boldsymbol{dd}'$ がわからないのです。行列の形で書くと因子分析モデルは次のようになります。

$$\boldsymbol{R} = \boldsymbol{AA}' + \boldsymbol{D^2}$$

ここで $\boldsymbol{D}$ は対角項に d_j^2 をもつ対角行列です。共通因子を得るために分解するべき行列は，次のようになります。

$$\boldsymbol{R}\dagger = \boldsymbol{R} - \boldsymbol{D^2} = \boldsymbol{AA}'$$

$\boldsymbol{R}$ は相関行列であり，その対角は 1.0 ですが，分解すべき行列 $\boldsymbol{R}\dagger$ は対角項に $1 - d_j^2 = h_j^2$ が入った行列でなければなりません。それを固有値分解してはじめて，共通因子成分が得られるわけです。そしてこの d_j^2 の大きさは，事前にわかっていないので，計算の最初にはなんとかしてこれを埋める必要があります。これを**共通性の推定問題**といいます。

実際の計算においては色々な方法が考えられてきています。推定せずに 1.0 で分析しちゃう方法，その行における相関係数の最大値を入れる方法，その項目を他の項目から回帰分析した時の R^2 値を入れる方法 (SMC 法) などです。最初の「共通性を推定しない」という方法は，**主成分法 (principle component method)** と呼ばれます。**主成分分析**という分析法と同じやり方だからです。第二，第三の方法は，ひとまずその値で計算しますが，その後固有値分解をした行列から相関行列を再構成する，つまり $\boldsymbol{R}\dagger \rightarrow \boldsymbol{AA}' \rightarrow \boldsymbol{R}\dagger_2 \rightarrow \boldsymbol{AA}' \rightarrow \cdots$ という繰り返しをおこない，数字が変化しなくなるまで反復するという方法で，**主因子法**と呼ばれます[*2]。この主因子法と同じ結果になるとされているのが，**最小二乗法**です。回帰分析の時に聞いたことのある方法ですね。それと同じ考え方で，因子数が決まっていれば，因子構造の形とデータとの誤差が最も小さくなるように共通性を推定するという方法です。原理的に主因子法と最小二乗法は同じ結果になるとされています。最小二乗法が出てきたらひょっとして，と思うかもしれませんが，そうその通りで，**最尤法**もあります。得られたデータが多次元正規分布からのサンプルだと考え，多次元正規分布の形にぴったりフィットするように推定パラメータを定めていく方法です。

このように，探索的に因子分析をする場合は，共通性をどのように推定するかということを指定しなければなりません。

10.3 因子数の決定

先ほどの最小二乗法，最尤法の説明の時に「因子数が決まっていれば」とありました。そうです，探索的な方法の場合は，因子分析をするときに因子数を決めて，何因子の答えを出すかを定めてやらなければなりません。それには色々な方法が考えられています。ここでは古典的な方法 4 つと，洗練された方法 1 つを紹介します。

■スクリープロットを見る　古典的な方法その 1 です。「見る」と書いてあるように，目で見て判断します。**スクリープロット (scree plot)** とは，固有値を大きい順にならべ，横軸に大きさの順番，立て軸に固有値の値をとった折れ線グラフのことです。これを見て，大きな変化が見られたところを基準とし，因子の数を決めます。

■固有値 1.0 という基準　固有値を算出し，1.0 以上であれば共通因子，それ未満であれば誤差因子とまとめる基準のことを言います。1.0 というのは項目の分散の大きさであり，固有値はこの分散の大きさを再配

[*2] 反復せずに最初に推定した初期値でいく方法もあります。これは反復しない主因子法ですが，計算機能力が十分高い今では反復するのが基本であり，とくに断りなく主因子法とあれば反復して求めていると考えていいでしょう。

分する手続きなのでした。再配分した結果，1 項目分も情報がないような次元は誤差みたいなものでしょ，という判断をすることになります。

■累積寄与率をみる 固有値の大きさを項目数で割ることで，1 つの因子が全体分散の何 % を説明するかを算出できます。これをとくに**寄与率 (contribution ratio)** といいます。これをどんどん積み重ねていって (累積)，累積寄与率が 50% を超えるところまでを共通成分とみなす，という方法です。因子分析は項目情報の圧縮をしていることでもあり，共通因子に入れないものはゴミとして捨てるようなものですが，全体の半分以上をゴミとして捨てたらいかんでしょ，という基準です。

■解釈可能性を考える 因子数に迷った時は，とりあえず何パターンかで分析してみて，最も解釈しやすい数で OK にしましょう，というやり方です。なんと主観的な方法でしょうか。数学的な基準ではなく，研究者のカンに頼る方法であり，客観性や再現性の問題を考えると容認し難い方法ではありますが，実際にはよく使われているやり方です。

■平行分析による方法 ここまでの方法は幾分古く，見た目や感覚にたよるものでしたが，**平行分析 (parallel analysis)** は数学的な基準で分析を行うものです。これは分析するデータセットと同じサイズの乱数を発生させ，その固有値構造とデータの固有値構造を比較するというものです。乱数で作られる構造は，心理的な反応パターンに依らないのですから，当然無意味な因子ばかりが出てきます。実際のデータに基づく相関係数が持つ構造は，心理的なパターンを反映しているのですから，それとは違う有意味なパターンになっているはずです。この有意味性，無意味性を比較して，乱数で作る因子以上の意味ある次元がでていれば，共通因子として採用するという方法です。

以上，5 つほど説明してきましたが，これについては実際に見てもらったほうが早いと思いますので，今から演習を始めたいと思います。

10.4 探索的因子分析の実際

10.4.1 環境の準備 (確認)

まずは環境の準備です。すでに R や RStudio のインストールは終わっているものとして話を進めます。いつもの 3 つのステップをたどって，実行の準備をしてください。

■RStudio の起動 RStudio を起動してください。パッケージのインストールをするときなど，管理人権限が必要になるケースがあります。

■プロジェクトを開く プロジェクトは前回作成した物で良いと思います。同じフォルダの中で，スクリプトファイルだけ変えれば良いでしょう。メニューからプロジェクトを開き (FIle→Open Project)，RStudio の右上などで，プロジェクトが開いていることを確認しましょう。

■R スクリプトを開く 今日のコードを書くための R スクリプトファイルを準備します。File > NewFile > R Script と進み，何も書いてない R スクリプトの画面を表示させてください。真っ白いファイルが開いたら問題ありません。

また今日は `psych` パッケージ Revelle (2021) を用います。準備がまだの人は Package タブからインストールするか，コンソールで `install.packages("psych")` と入力してパッケージを導入しておいてください。

データセットも `psych` パッケージが持っているものを使います。

code : 10.1 サンプルデータを使う

```
1 rm(list=ls())
2 library(tidyverse)
3 library(psych)
4 dat <- psych::bfi %>% dplyr::select(-gender,-education,-age)
5 dat
6 help(bfi)
```

■コード解説

1 行目 環境の初期化

2-3 行目 パッケージの読み込み

4-5 行目 サンプルデータ `bfi` を使う。性別, 教育, 年齢の情報は因子分析には使わないので除外する。

6 行目 データセットのヘルプを表示

ヘルプ画面にあるように, これは 25 のパーソナリティについての調査項目からなる, 2800 人分のデータです。A(Agreeableness, 同調性), C(Conscientiousness, 几帳面さ), E(Extraversion, 外向性), N(Neuroticism, 神経質さ), O(Openness, 開放性) という Big 5 の各要素に対応した項目が 5 つずつあります。加えて性別 (`gender`), 最終学歴 (`education`), 年齢 (`age`) といった 3 つの変数も入っています。今回使うのは, 5 つの因子それぞれに対応すると考えられる各 5 つの項目, 計 25 項目分です。

10.4.2 平行分析による因子数の決定

code : 10.2 データの構造を見る

```
1 corMat <- cor(dat,use="pairwise")
2 corMat
3 eigen(corMat)
4 fa.parallel(dat)
```

■コード解説

1-2 行目 データの相関行列を計算し, 表示させる

4 行目 相関行列の固有値分解を実行

5 行目 平行分析の実行

因子分析は相関行列から分析を始めますので, まずその計算をしました。オプション `use="pairwise"`は欠測値が含まれるデータセットに対し, ペアで残っている箇所は使って分析するという指定です[*3]。4 行目で固有値分解をしてみました。結果として出力 10.1 のようなものが示されています。

[*3] これがなかったら, 欠測値があるので計算できません, で終わります。

R の出力 10.1: 固有値の推移

```
eigen() decomposition
$values
 [1] 5.0369025 2.7440855 2.1076322 1.8318415 1.5356864 1.1131589 0.8462367 0.8114075
 [9] 0.7349482 0.6956449 0.6810036 0.6571432 0.6281130 0.5964129 0.5626284 0.5405207
[17] 0.5236707 0.4984540 0.4899130 0.4549419 0.4328324 0.4092023 0.4071932 0.3852111
[25] 0.2752152
```

この下に固有ベクトル (`$values`) も続いていますが，長くなるので省略しました。ここで固有値が 25 個算出されていることがわかります。また，これらを総和すると 25 になります ($tr(\boldsymbol{R}) = \sum_i = 1^{25} r_{ii}$)。このように標準化されたデータの分散を再構築し，第 1 固有値は 5.03，すなわち項目 5 個分の情報を持っているようになったのです。第 2 固有値が 2.744，第 3 固有値が 2.107，第 4 固有値が 1.183... と続きます。また，これを相対比率に変えると，$5.03/25 = 20.12$ ですから，第 1 固有値だけで全体の 20% を説明できることになります。第 2 固有値は 2.744 ですから，2.744/25=0.1096 で全体の 10% ほど，第 1 固有値と合わせて 31.12% の累積寄与率があることになります。累積寄与率は第 5 固有値までで 53.02% になりますから，25 項目すべてを使わずとも 5 因子だけで情報の半分は説明できることになります！

また，第 7 固有値の大きさは 0.846 であり，それ以降の固有値はすべて 1.0 未満，すなわち項目 1 つ分も情報を持っていない因子ということになります。であれば共通因子として掬い上げる必要はなく，少なくとも 7 番目以降 (基準 2)，あるいは 6 番目以降 (基準 3) は誤差と考えてまとめてしまってもいいかもしれませんね。

続く `fa.parallel` 関数は平行分析を行う関数で，同時に**スクリープロット**も表示してくれます (図 10.1)。

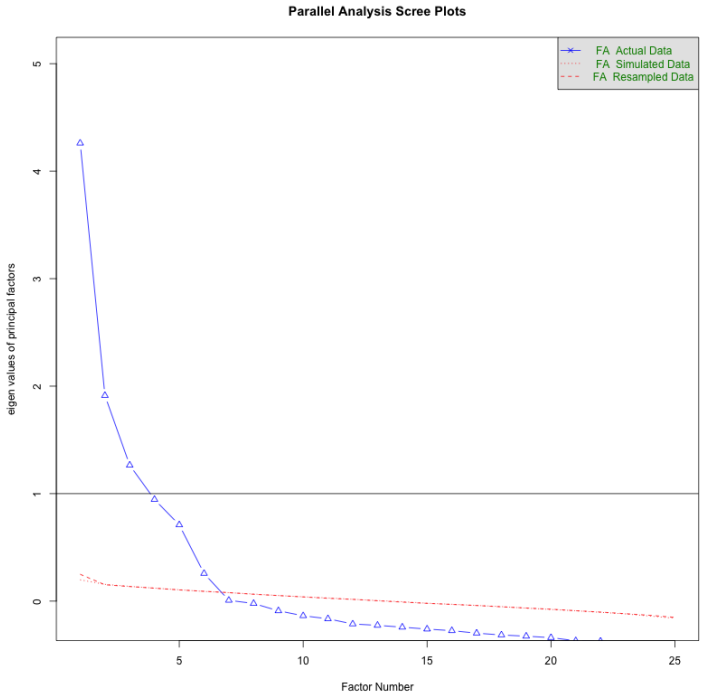

図 10.1 スクリープロットと平行分析

この図は先ほど算出されたような固有値[*4]を大きい順に並べ，線で繋いだものが示されています。青い線が実データ，赤い点線が乱数によるにデータサイズが同じだけの偽データです。基準 2 に該当する 1.0 のところで線が引かれていて，これよりも大きい固有値を持つのが 3 つあることがわかります。まだ，第 1 固有値から第 2 固有値にかけて大きな傾斜があります (4.26063957 → 1.91279965)。最初のうちはまあ当然として，第 5 と第 6 の間も少し大きなギャップがあるようです (0.70938377 → 0.25749866)。それ以降はだらだらと数値が減衰していますね。このような大きなギャップのところは，大きな意味の違いがあると考えてもいいでしょう。基準 1 の観点からいくと，第 5 固有値までを共通因子とし，第 6 因子以降は誤差としてまとめてしまってもいいでしょう。また，平行分析の観点では，赤い線よりも上にある青いラインは有意味ということですから，6 因子構造がいい，ということになりますね。

このように，さまざまな基準で考えても，因子数は 5 か 6 のどちらかになって決定打に欠けます。こういうときは解釈しやすい 5 因子にしよう (基準 4)，ということになったりします。

ちなみにこの固有値構造，最終的には負の値が出ていますので，あまり良いデータとは言えません。25 項目を用意していたのに，8 因子以降は 0 以下，つまり情報がないようなものです。これは項目同士が似かよっていて，あまり違った情報を持ってこなかったということでもあります。実際のデータを分析する時は，こうした点にも注意が必要です。

10.4.3 因子分析の実行

それでは因子数を 5 と定めて因子分析を実行してみましょう。

code：10.3 因子分析の実行

```
1       result <- fa(dat, nfactors = 5, fm = "ml", rotate = "geominQ")
2       print(result, sort = T)
```

これは 1 行目で実行し，2 行目で表示しているだけですが，内容を少し説明します。psych パッケージの持つ fa 関数は，データの他に因子数 nfactors，共通性の推定方法 fm，回転方法 rotate などをオプションで定めることができます。共通性の推定方法は，ここでは ML(**最尤法**) を選択しました。回転方法とは**因子軸の回転 (rotation of factor axis)** について定めるところです。これについては少し説明が必要ですね。

■因子軸の回転 因子分析は，相関行列という項目間関係が作り出す空間に，その基礎となる軸を見出すものでした。これを図 10.2 のように表現してみます。項目同士はその相関関係から空間を作っていますが，ここでは 2 因子，F1 と F2 という軸をもとに座標のように書いています。項目同士が近くにあるのは，相関あっていることを意味します[*5]。ここで項目 1 の座標を見ると，第一因子 (F1) に 0.3，第二因子 (F2) に 0.6 とありますが，これが項目 1 の因子に対する因子負荷量ということになります。

因子分析では**因子負荷量**をもとに，「この項目と関係の深い因子はどれか」「因子はどういう項目から構成されているか」ということを考えていくのですが，項目 1 は第一，第二因子それぞれにちょっとずつ関係していることがわかります。そのほかの項目も，2 つの因子に負荷していますがなかなか決め手がない，というところですね。

そこで座標を回転させます。どの項目がどの因子に関係しているのかを，よりはっきりさせるように軸をグルリと回すのが因子軸回転の目的です。図 10.3 のように回転させますと，回転後の軸 $(F1', F2')$ に下ろした垂線が座標，すなわち因子負荷量になりますから，項目 1 は第一因子 $(F1')$ に 0.9，第二因子 $(F2')$ に

[*4] 正確には $\boldsymbol{R}\dagger$ の固有値です。

[*5] 原点からのベクトルと考えたときに，2 つの項目ベクトルが作る角度 θ のコサインを取った $\cos\theta$ が相関係数になります。

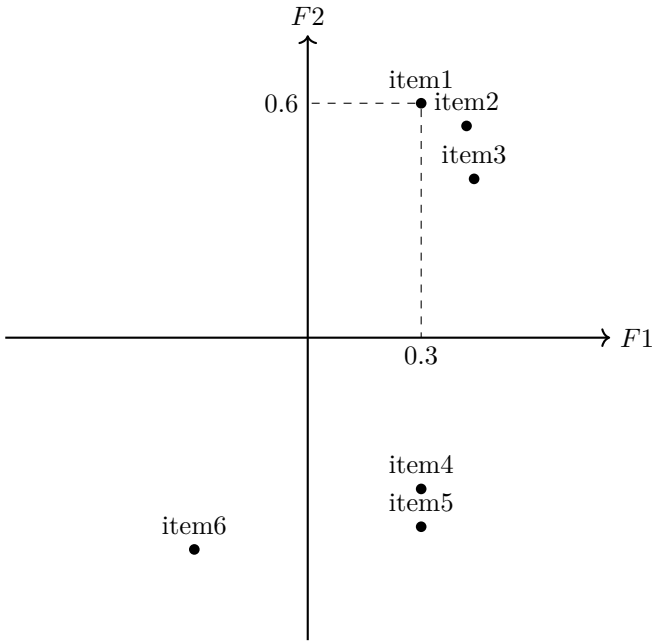

図 10.2 項目の空間にある基底

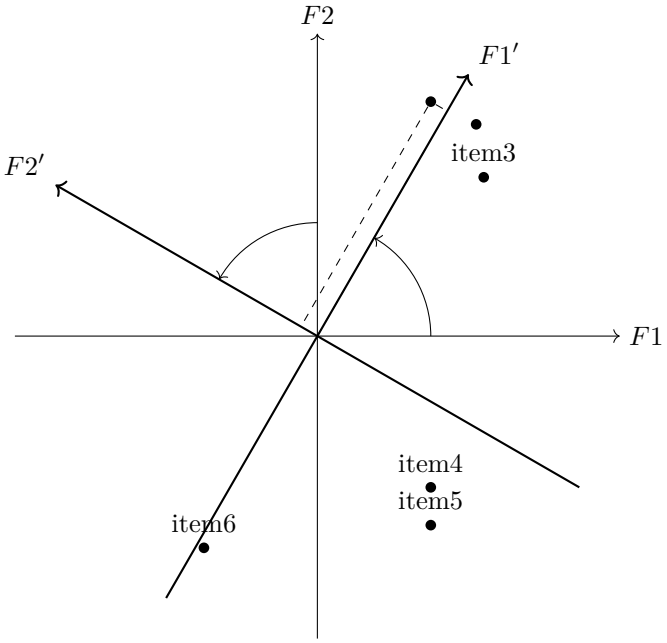

図 10.3 因子軸の回転。60 度回転させてみました。

−0.07 になりました。こうしてみると，項目 1 は第一因子と非常に深く関係しているし，第二因子とはほとんど関係がない，とみることができます。ほかの項目も一方の因子に大きく負荷し，他方の因子にほとんど負荷していないようになりますから，どの因子がどういう項目と関係するかを評価するのにメリハリが効いてやりやすくなります。このように，利用しやすく座標を変えるのが因子軸の回転，という技なのです。

ここで最初の因子軸 $F1, F2$ は直交していましたね。座標ですから当然です。直交している，すなわち因子間相関がないように回転することを**直交回転 (orthogonal rotation)** といいます。

しかし，因子軸の回転の目的は，わかりやすくするためなのですから，いっそ図 10.4 のように，角度をつけた方がさらにうまくメリハリをつけることができますね。この図では，項目 4,5 が明らかに第二因子に影響している，ということが見て取れるようになりました。

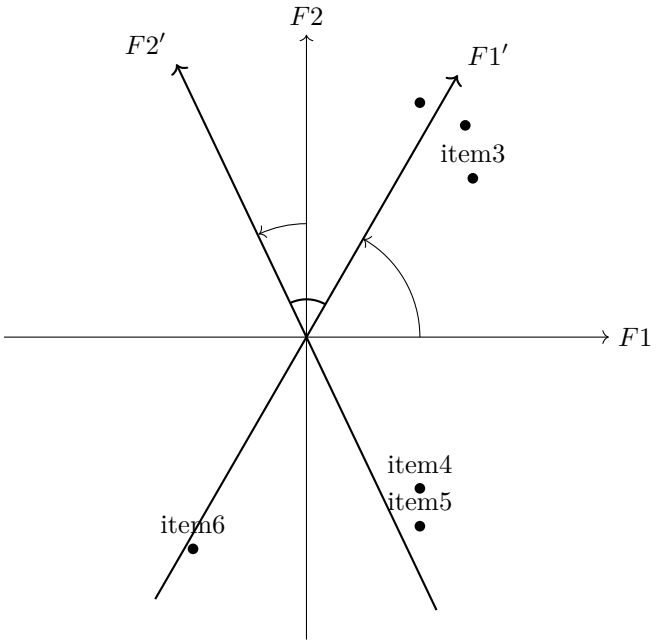

図 10.4　角度をつけて回転させる斜交回転の例。角度があるとは，因子同士が相関するということ。

このように，因子軸が直角で交わっていない場合の回転方法を**斜交回転 (oblique rotation)** といいます。因子軸の回転は一般に，斜交回転した方がメリハリが効いて因子を解釈しやすくなります。斜交回転したとき算出される因子同士の相関を考えて，これが十分小さい，すなわち直角とほぼ見做せる程度だとおもわれたら直交回転を行う，とするのが良いでしょう。

■行列による説明　因子分析の行列モデルは次のようなものでした。

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{A}' + \boldsymbol{D}^2$$

ここで $\boldsymbol{A}\boldsymbol{A}'$ に単位行列 $\boldsymbol{I}$ を挟んでも，$\boldsymbol{A}\boldsymbol{A}' = \boldsymbol{A}\boldsymbol{I}\boldsymbol{A}'$ で結果は変わりませんね。ここで $\boldsymbol{I} = \boldsymbol{T}\boldsymbol{T}'$ となるような行列 $\boldsymbol{T}$ があっても同じです。たとえば 2 因子構造の時に，次のような行列 $\boldsymbol{T}$ を考えたとすると，$\boldsymbol{T}\boldsymbol{T}' = \boldsymbol{I}$ であることがわかります。

$$\boldsymbol{T} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}$$

$$\boldsymbol{T}\boldsymbol{T}' = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \boldsymbol{I}$$

このような $\boldsymbol{T}$ は回転行列と言われます。原点を中心に，座標を角度 θ で回転させるからです。θ の値は任意なので，因子負荷行列は，θ に応じて無数にあることになります。であれば，使い勝手の良い角度で回転してやればいいとも言えるわけです。

直交回転の場合は角度が直角，すなわち相関がないわけですから，数学的には回転行列が $\boldsymbol{T}\boldsymbol{T}' = \boldsymbol{I}$ のようになった状態を言います。一方で**斜交回転**の場合は相関があるわけですが，

$$\boldsymbol{I} = \boldsymbol{T}'\boldsymbol{\Phi}\boldsymbol{T}$$

となるように置いておけば，$\boldsymbol{R} = \boldsymbol{AT'\Phi TA'} + \boldsymbol{D}^2 = \boldsymbol{BB'} + \boldsymbol{D}^2$ の形が保たれているので因子分析モデルとしては違いがないわけで，このような $\boldsymbol{\Phi}$ を任意に定めることができます。この時 $\boldsymbol{\Phi}$ は**因子間相関 (factor correlations)** と呼ばれます。

直交回転も斜交回転も，目的は**解釈をしやすくする**ということにあり，そのための基準はいくつか考えられます。それらの基準に応じて，たとえば直交回転では**バリマックス回転 (varimax rotation)**，斜交回転では**プロマックス回転 (promax rotation)** などがあり，ここでは**ジオミン回転 (geomin rotation)** を選びました[*6]。

10.4.4 因子分析の結果をみる

出力 10.2 に因子負荷行列を表示しました。あの $\boldsymbol{A}$ が計算されていますよ！

R の出力 10.2: 因子負荷行列

```
Factor Analysis using method =  ml
Call: fa(r = dat, nfactors = 5, rotate = "geominQ", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
   item   ML2   ML1   ML5   ML3   ML4   h2   u2 com
N1   16  0.83 -0.07 -0.24  0.02 -0.04 0.71 0.29 1.2
N2   17  0.80 -0.02 -0.23  0.03  0.02 0.66 0.34 1.2
N3   18  0.68  0.15 -0.01 -0.03  0.03 0.53 0.47 1.1
N5   20  0.48  0.26  0.15  0.01 -0.13 0.34 0.66 1.9
N4   19  0.46  0.43  0.03 -0.12  0.09 0.48 0.52 2.2
E2   12  0.10  0.69 -0.08  0.00 -0.04 0.55 0.45 1.1
E1   11 -0.07  0.58 -0.07  0.12 -0.09 0.37 0.63 1.2
E4   14  0.00 -0.55  0.33  0.00 -0.07 0.52 0.48 1.7
E5   15  0.13 -0.43  0.03  0.25  0.22 0.40 0.60 2.4
E3   13  0.05 -0.38  0.27 -0.02  0.30 0.44 0.56 2.8
A3    3  0.02 -0.11  0.65  0.05  0.03 0.51 0.49 1.1
A2    2  0.03 -0.02  0.59  0.11  0.03 0.40 0.60 1.1
A5    5 -0.09 -0.21  0.58  0.01  0.04 0.48 0.52 1.3
A4    4 -0.03 -0.05  0.45  0.21 -0.15 0.29 0.71 1.7
A1    1  0.16 -0.12 -0.39  0.02 -0.03 0.15 0.85 1.6
C2    7  0.09  0.10  0.07  0.63  0.09 0.43 0.57 1.2
C4    9  0.18  0.06  0.05 -0.62 -0.05 0.47 0.53 1.2
C3    8  0.01  0.06  0.09  0.56 -0.04 0.32 0.68 1.1
C5   10  0.19  0.18  0.00 -0.55  0.08 0.43 0.57 1.5
C1    6  0.01  0.03 -0.02  0.52  0.18 0.32 0.68 1.2
O3   23 -0.01 -0.15  0.07 -0.01  0.62 0.47 0.53 1.1
O1   21 -0.03 -0.10  0.01  0.04  0.53 0.32 0.68 1.1
O5   25  0.14 -0.04  0.08 -0.03 -0.52 0.27 0.73 1.2
O2   22  0.20  0.00  0.18 -0.07 -0.44 0.24 0.76 1.8
O4   24  0.11  0.33  0.13 -0.02  0.39 0.26 0.74 2.4
```

出力 10.2 にあるように，各行に N,E,A,C,O の項目が並んでいます。この順番が入れ替わっているのは，表示オプションで `sort=T` を入れたからで，因子負荷量の大きい順に並べ替えて表示させたからです。現に左上から右下にかけて，因子負荷量の大きさの順に並んでいますね。列として ML1,2,3,4,5 とあるの

[*6] ジオミン回転には直交モデルと斜交モデルがあり，それぞれ `geominT`,`geominQ` という名前で実装されています。

は最尤法 (ML) で抽出した因子の 1,2,3,4,5 を表しています。h2 は h^2, u2 は $1-h^2=d^2$ をあらわしています (**独自性 (uniqueness)** の u)。com は**複雑性指標 (index of complexity)** と呼ばれ, $\sum(a_j^2)^2/\sum a_j^4$ で計算される指標です。共通性は高く, 独自性, 複雑性の低い項目がいい項目だといえるでしょう。ちなみに, 出力オプションとして cut=0.3 を追加すると, 因子負荷量が 0.3 に満たないところの表示はなくなります。どの項目がどの因子と関係あるのかがわかりやすくなるのでおすすめです。

出力 10.3 にはその他の情報として, 各因子の因子負荷行列の二乗和 (SSloadings), 分散説明率 (Proportion Var), 累積分散説明率 (Cumulative Var), 全体を 1 とした時の説明比率 (Proportion Explained), その累積 (Cumulative Proportion) が表示されています。回転するまえは固有値と**因子負荷量**の二乗和が合致するのですが, 回転した後の場合はそうはならないので, これらをみてどれぐらい説明できているかを考えます。とくに累積 ss 分散説明率が, 今回は 5 因子で 41% しかありません。59% をゴミとして捨てたようなものですから, これでよかったのかな, と思案するところです。

R の出力 10.3: 因子分析結果 2

```
ML2  ML1  ML5  ML3  ML4
SS loadings           2.50 2.24 2.05 1.95 1.61
Proportion Var        0.10 0.09 0.08 0.08 0.06
Cumulative Var        0.10 0.19 0.27 0.35 0.41
Proportion Explained  0.24 0.22 0.20 0.19 0.16
Cumulative Proportion 0.24 0.46 0.66 0.84 1.00
```

その下出力 10.4 には, 因子間相関が表示されています。これをみると, 第 2 因子 (ML1) と第 3 因子 (ML5) との間に −0.34 という中程度の負の相関がみられます。直交回転はこれを 0 として解釈してしまいますから, スッキリはするのですがデータには適合してないところかもしれません。このように, 分析する時はまず斜交回転を行い, 因子間相関がなべて低い場合は直交で分析をやり直す, というステップを経ます。

R の出力 10.4: 因子分析結果 3

```
 With factor correlations of
      ML2   ML1   ML5   ML3   ML4
ML2  1.00  0.11  0.06 -0.12  0.06
ML1  0.11  1.00 -0.34 -0.22 -0.16
ML5  0.06 -0.34  1.00  0.18  0.23
ML3 -0.12 -0.22  0.18  1.00  0.17
ML4  0.06 -0.16  0.23  0.17  1.00
```

10.5 因子分析の後で

因子数を定め, 因子負荷量が明らかになると, ここから逆算的に**因子得点**を計算できます。fa 関数には scores オプションがあり, これをつけて実行すると因子得点行列を表示させることができます。因子得点は因子数 × 人数分 (ここでは 2800 人分) ありますので, 大変大きなサイズのデータですから注意してください。ともあれ, そこに含まれているのは, 各因子についての個々人の相対的な関与度ということになります。

code : 10.4 因子得点を算出する

```
1  result <- fa(dat, nfactors = 5, fm = "ml", rotate = "geominQ", scores = T)
2  result$scores
```

モデルの仮定にあった通り，因子負荷量は標準化されたスコアです。すなわち平均情報が取り払われていることに注意してください。たとえば 7 件法で「4. どちらでもない」を挟んで，ポジティブ・ネガティブな意味を持っていたとしても，ここでは個々人間の相対的な大小関係しか意味しないことになります。

そういう意味で，絶対的なスケールの情報を残したまま得点を考えてみたいということもあるでしょう。そういう時は**簡便的因子得点**といって，因子に関係する項目の素点から平均点を算出し，個々人の得点とする方法もあります。最もこのやり方は，平均点の情報は保持されますが，因子分析で除去できた d_j^2，すなわち独自性成分を再び取り込むことになっている点に注意が必要です。次のコード code:10.4 は，簡便法による因子得点と因子分析の結果から計算された因子得点との相関係数を算出する例です。高い相関を示すところ (ML2 と N で 0.9475742，ML5 と A で 0.83845417) もありますが，低いところもありますので，どちらの方法で計算するのか，その場合の長所短所をよく理解して使うようにしましょう。

code：10.5　二種類の因子得点計算法比較

```
dat %>%
    dplyr::mutate(
        A = (A1 + A2 + A3 + A4 + A5)/5,
        C = (C1 + C2 + C3 + C4 + C5)/5,
        E = (E1 + E2 + E3 + E4 + E5)/5,
        N = (N1 + N2 + N3 + N4 + N5)/5,
        O = (O1 + O2 + O3 + O4 + O5)/5
    ) %>%
    dplyr::bind_cols(.,result$scores %>% as.data.frame) %>%
    dplyr::select(ML1,ML2,ML3,ML4,ML5,A,C,E,N,O) %>%
    cor(use="pairwise")
```

10.6　さいごに

今回は探索的因子分析の実行例をみてみました。いかがだったでしょうか。計算自体にはほとんど時間がかかりませんので，色々な因子数や回転方法，抽出法を試して見るのも良いかもしれません。

このように比較的簡単に計算できるようになってきましたが，大事なのはその目的と意味です。関数の名前を覚えて，機械的に XX 法 XXX 回転で答えが出たから正しいのだ，などと思わないことです。因子数の決め方の時にも薄々感じていただけたかと思いますが，数学的なモデルとは裏腹に，実際にやる時は職人的ノウハウも結構必要です。スクリープロットを「みて」とか，回転法を「選んで」とか，因子得点の計算の仕方を「比較して」，といったところは人間的要素であり，やり方が変われば結果が変わる可能性があります。論文などに示されているこれらの方法，指標も，変えようと思えば変得られるところだということです。科学的真実の探究が目的であれば，良い見栄えや自分に都合の良い結果のための手法選択は無意味であることは理解してもらえるところだと思います。皆さんも決して目的を見失わず，批判的に検証できるようになってください。

探索的因子分析は因子を見つけ出すことができます。が，それは決して人間の心の何かを取り出したわけではないのです。人間がそもそも持っている心の状態を取り出したのではなく[*7]，人間に「項目群」「紙とペン」という刺激を与え，反応を強いたのです。その結果，「項目群にみられる相関行列に潜む次元」が見いだせただけであり，それが人間の持つ特性だと考えるのは，あくまでの研究者の主観的主張にすぎません。実際にあり得る話ですが，「学校生活は嫌ですか」「学校は楽しくないと思うことがありますか」「学校の友達とは遊びたくないと思うことがありますか」といったネガティブな項目を山のように投げかけ，因子分析することで

[*7] 因子分析をすると「因子を抽出した」と論文などに記載される場合がありますが，このようにあたかも先にあるエッセンスを抜き出したのだ，という表現は適切ではないと批判されることもあります。

「学校を嫌う項目群＝因子」を抽出し，子供は学校が嫌いなのだという結論を出すのがおかしいことなのはわかりますよね。項目群のなかにネガティブなものしかなければ，因子もそれに応じたものしか出ません。そして因子得点が相対的なものでしかないので，素点を見ずに「学校嫌い得点が高い，低い」として議論を進めても仕方がないのです。そうした場合は簡便的因子得点を計算して，平均点がちゃんと中点「4. どちらでもない」を挟んでいるかどうかで嫌っているかどうかを判断しなければなりません。このように，自分で項目に埋め込んだ呪いを自分で取り出して見つけ出したと騒ぐ，**てっちゃんの手品**にならないようにすることは，常に心がけなければならないのです[*8]。

こうした批判的観点を持つためには，数理的にはどういうことをやっていて，どこからが人為的な問題なのかをしっかり把握しておくことが役に立つでしょう。

10.7 課題

今日の授業でおこなったすべての次の計算をする R コードを提出してください。ファイル形式は R スクリプトか Rmd とします。なお提出されたコード単体でバグがなく動くことが確認できないものは，未提出扱いになります。コードの書き方などわからないところがあれば，曜日別 TA か小杉までメールで連絡し，指導を受けてください。

[*8] てっちゃんの手品については，小杉 (2018) を参照してください。

第 11 章

R をつかった項目反応理論

今回も R を使った演習です。R を使っての行列計算，R を使っての因子分析と進めてきましたが，今回は R をつかって項目反応理論を実践的に理解していきましょう。

11.1 項目反応理論の実際

11.1.1 データの素描

項目反応理論は，テスト理論に関するモデルですから，サンプルデータとして二値データを使いましょう。表 11.1 に使うデータの一部を示しています[*1]。

表 11.1 IRT で使うサンプルデータ (一部)

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	1
0	1	0	1	0	0	0	0	0	1
0	1	1	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0	1	1

このように，テスト理論で使うデータは 0/1 の二値データで，1 が正答，0 が誤答を表しています。

これらのデータを使って R で分析をします。IRT を実行するパッケージとして，`ltm`Rizopoulos (2006) や `irtoys`Partchev et al. (2017) などがあります。ここでは `ltm` パッケージの方を使っていきましょう。

分析に先立って，データの記述統計量を確認しておきます。`code:11.1` を実行してください。

code : 11.1 データの読み込みから記述統計量の計算まで

```
1    rm(list=ls())
2    library(tidyverse)
3    library(ltm)
4    dat <- read_csv("IRTsample.csv")
5    ltm::descript(dat)
```

[*1] このデータ全体については，`IRTsample.csv` というファイル形式で配布していますので，それを読み込んで使ってください

■コード解説

1 行目 環境の初期化

2-3 行目 パッケージの読み込み

4-5 行目 サンプルデータ `IRTsample.csv` を読み込む。文字化けする人は，`read_csv` 関数のオプションとして `locale=locale(encoding="UTF-8")` を追加してください。

6 行目 ltm パッケージに含まれる `descript` 関数を実行

いろいろな情報が一気に出力されるので，順にみていきます。まず **R の出力** 11.1 のところです。ここには各変数の 0 と 1 の比率が入っています。この例では，`V1` は 0 が 71.0%，1 が 29.0% 含まれているということです。最後の `logit` は，0 から 1 の間に入る数字 p にたいして，$logit(p) = \log\left(\frac{p}{1-p}\right)$ で計算される量です。ここでは p は正答率であり，この `logit` の値が負であれば 0 の比率が，正であれば 1 の比率が多かった，ということを意味します。0/1 がちょうど半々であれば，`logit` の値は 0 になる，そういう数字です。これを見るだけでも，`V8` はずいぶんと正答率が低かったんだな，`V3` は逆に正答率が高かったんだな，ということがわかります。

R の出力 11.1: 記述統計の出力 1

```
    Proportions for each level of response:
    0     1   logit
V1  0.710 0.290 -0.8954
V2  0.114 0.886  2.0505
V3  0.186 0.814  1.4762
V4  0.206 0.794  1.3492
V5  0.670 0.330 -0.7082
V6  0.832 0.168 -1.5999
V7  0.666 0.334 -0.6901
V8  0.884 0.116 -2.0309
V9  0.652 0.348 -0.6278
V10 0.254 0.746  1.0774
```

次のセクション **R の出力** 11.2 には，正答数の分布が示されています。10 問全部正解したのは 3 人だけ，6 問正解した人が 104 人だった，といったことがわかります。

R の出力 11.2: 記述統計の出力 2

```
        Frequencies of total scores:
        0  1  2  3  4   5  6  7  8  9 10
   Freq 4 19 41 62 97 104 66 54 36 14  3
```

次のセクション **R の出力** 11.3 には，**点双列相関係数 (Point Biserial correlation)** が計算されています。この相関係数は 2 値と連続値の相関係数であり，ここでは各変数 (二値) と合計得点 (0 から 10) の相関係数を計算しています。この合計得点の計算方法には，その変数自身を含める場合 (Included) と含めない場合 (Excluded) の二種類があります。具体的に説明したほうがわかりやすいでしょう。たとえば変数 `V1` についていうと，「その変数自身を含める場合」の合計得点とは $V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10$ で計算しています。「含めない場合」は $V2 + V3 + \cdots + V10$ とするわけです。含める場

合の相関係数が 0.5207, 含めない場合の相関係数が 0.2480 です。当然自身が含まれているほうが合計得点に深く関わっているので相関係数が高くなります。裏を返せば, 含める場合と含めない場合の相関係数に大きな差があるとき, その変数はテスト全体に大きく貢献する重要な変数だといえるでしょう。

R の出力 11.3: 記述統計の出力 3

```
Point Biserial correlation with Total Score:
    Included Excluded
V1    0.4228   0.2118
V2    0.3959   0.2502
V3    0.5202   0.3567
V4    0.5209   0.3501
V5    0.5449   0.3470
V6    0.4635   0.2983
V7    0.5440   0.3452
V8    0.2901   0.1350
V9    0.4992   0.2890
V10   0.5890   0.4180
```

次のセクション R **の出力** 11.4 には, α **係数 (alpha coefficient)** が算出されています。これも全体の場合と, 特定の変数を抜いた場合とを計算しています。心理尺度における信頼性係数としての α 係数は, 0.8 以上であれば信頼性がある, といった目安がありますが, 二値データの場合はここに示したようにそれよりもずいぶんと低くなります。これは二値データの分散が小さくなることから仕方のないことですし, 項目反応理論で分析する上での信頼性は情報量関数として表されますから, 問題ではありません。

R の出力 11.4: 記述統計の出力 4

```
        Cronbach's alpha:
        value
All Items     0.6341
Excluding V1  0.6303
Excluding V2  0.6195
Excluding V3  0.5978
Excluding V4  0.5986
Excluding V5  0.5982
Excluding V6  0.6100
Excluding V7  0.5987
Excluding V8  0.6380
Excluding V9  0.6130
Excluding V10 0.5817
```

最後にセクション R **の出力** 11.5 では, 変数同士の関連がとくに弱いものが示されています。変数同士のペアについて度数分布表を作成し, χ^2 検定によって p.value を算出しています。この値が小さければ, 2 つの変数の関係が強いことを示していると解釈できるのですが, ここにあげているように大きな数字になっているということは, 変数間の関係が弱いと読み取ることができます。

R の出力 11.5: 記述統計の出力 4

```
Pairwise Associations:
   Item i Item j p.value
1       2      8   0.625
2       1      8   0.409
3       1      2   0.347
4       6      8   0.303
5       4      8   0.234
6       7      8   0.222
7       5      8   0.195
8       1      4   0.191
9       1      3   0.166
10      3      8   0.124
```

11.1.2 IRT モデルの実際

データの感じがわかったところで，IRT モデルを実行していきましょう。

今回は 3 つのモデルを試してみます。確認のために，少しモデル式に言及しておきます。困難度母数だけに注目するのが **1 パラメータ・ロジスティックモデル (1 parameter logistic model, 1PL)** で，モデル式は次の通りです。

$$p_j(\theta) = \frac{1}{1 + exp(-1.7(\theta - b_j))}$$

ここで $p_j(\theta)$ は能力が θ のひとが項目 j に正答する確率であり，b_j が**困難度母数** difficulty parameter と呼ばれます。これは開発者の名前を使って，別名**ラッシュモデル (rasch model)** とも呼ばれています。

項目の識別力にも注目するのが **2 パラメータ・ロジスティックモデル (2 parameter logistic model, 2PL)** で，モデル式は次の通りです。

$$p_j(\theta) = \frac{1}{1 + exp(-1.7a_j(\theta - b_j))}$$

ここで a_j は**識別力母数 (descriminant paramter)** と呼ばれます。

最後に **3 パラメータ・ロジスティックモデル (3 parameter logistic model, 3PL)** では，困難度，識別力に加えて**当て推量母数 (guessing parameter)** というのを推定します。モデル式は次の通りです。

$$p_j(\theta) = c + \frac{1 - c}{1 + exp(-1.7a_j(\theta - b_j))}$$

もちろん式ではピンと来ないという人もいると思いますので，実際にどのように表されるかみていきましょう。

1PL モデルの場合

`ltm` パッケージを使って 1PL モデルを実行します。`code:11.2` を実行してください。

code : 11.2 1PL モデルの実行

```
1  result.1pl <- rasch(dat)
```

```
2  print(result.1pl)
3  plot(result.1pl,type="ICC")
4  plot(result.1pl,type="IIC")
5  plot(result.1pl,type="IIC",items = 0)
```

■コード解説

1 行目 1PL モデル (ラッシュモデル) の実行

2 行目 結果の出力

3 行目 **項目特性曲線 (Item Characteristic Curve)** のプロット。plot 関数のオプションで表示する情報を ICC と指定。

4 行目 **項目情報曲線 (Item Information Curve)** のプロット。plot 関数のオプションで表示する情報を IIC と指定。

5 行目 **テスト情報曲線 (Test Information Curve)** のプロット。plot 関数のオプションで表示する項目番号をゼロにすると TIC になる。

2 行目で結果の出力を行っていますが, 出力 11.6 のように表されます。

R の出力 11.6: 2PL モデルの結果出力

```
Call:
rasch(data = dat)

Coefficients:
 Dffclt.V1   Dffclt.V2   Dffclt.V3   Dffclt.V4   Dffclt.V5   Dffclt.V6
     0.992      -2.208      -1.615      -1.480       0.787       1.744
 Dffclt.V7   Dffclt.V8   Dffclt.V9  Dffclt.V10      Dscrmn
     0.768       2.187       0.699      -1.189       1.126
Log.Lik: -2468.743
```

1PL モデルで表現する項目の違いは**困難度母数**だけですから, この数字 (Dffclt) が小さければ簡単な項目, 大きければ難しい項目だったことがわかります。たとえば V2 の困難度母数は $b_2 = -2.208$ ですが, 記述統計のところで見た正答率 (88.6%) を考えると, 簡単な問題だったことがわかりますね。**識別力母数 (は)**Dscrmn として 1.126 と推定されています。これは項目を通じて一定です。このことは図 11.1 の左にある ICC をみて, すべての曲線の傾きが同じことで確認できます。

図 11.1 の左が数値的出力結果を図示したものです。これを見た方がわかりやすいかもしれません。x 軸は θ, つまり測定する潜在変数の値になっており, テストで言えばある人の学力 θ_i があったときに, 各問いにどれぐらいの確率で正答するかを表していることになります。これをみると V2 がもっとも簡単で, V8 が最も難しかったテストであることがわかります。

また図 11.1 の中央にあるのが IIC で, それぞれの項目がどのあたりの能力値 θ を測定するときにもっとも敏感になるかを表しています。これは項目反応理論でいう**信頼性**の表現であることを思い出してください。最後に図の左側にあるのは, この IIC を 10 問分足し合わせたテスト全体の情報曲線, TIC になります。これをみると, $\theta = 0$ よりやや右側のあたりにピークがきているので, このテスト全体としては平均より高い学力の人を測定するときに, 最も鋭敏に働くことがわかるでしょう。

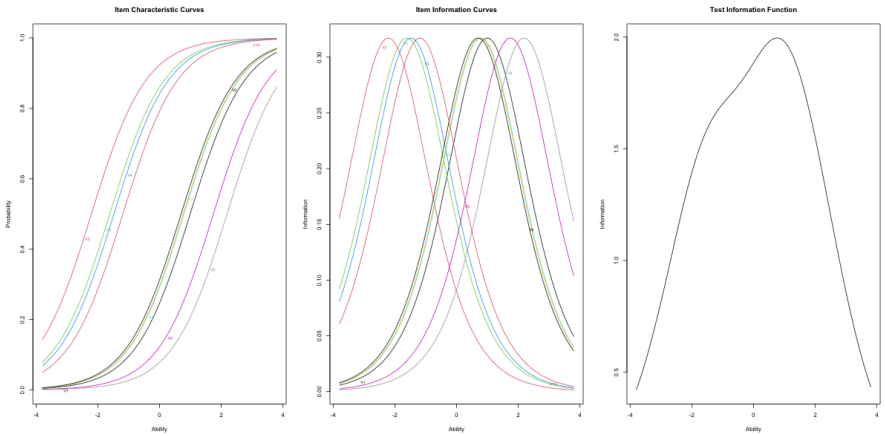

図 11.1 1PL のさまざまなプロット。左から ICC,IIC,TIC

2PL モデルの場合

ひきつづき ltm パッケージを使って，2PL モデルを実行していきましょう。code:11.3 を実行してください。

code : 11.3 2PL モデルの実行

```
1  result.2pl <- ltm(dat~z1)
2  print(result.2pl)
3  plot(result.2pl,type="ICC")
4  plot(result.2pl,type="IIC")
5  plot(result.2pl,type="IIC",items = 0)
```

■コード解説

1 行目 1PL モデル (ラッシュモデル) の実行

2 行目 結果の出力

3 行目 **項目特性曲線 (Item Characteristic Curve)** のプロット

4 行目 **項目情報曲線 (Item Information Curve)** のプロット

5 行目 **テスト情報曲線 (Test Information Curve)** のプロット

2 行目で結果の出力を行っていますが, 出力 11.7 のように表されます。

R の出力 11.7: 2PL モデルの結果出力

```
Call:
ltm(formula = dat ~ z1)

Coefficients:
     Dffclt  Dscrmn
V1    1.602   0.603
V2   -2.078   1.228
V3   -1.213   1.892
V4   -1.199   1.620
V5    0.759   1.176
V6    1.698   1.175
V7    0.740   1.174
V8    4.133   0.516
V9    0.866   0.828
V10  -0.880   2.020

Log.Lik: -2446.104
```

2PL モデルで表現する項目の違いは，**困難度母数**と**識別力母数**の 2 つです。識別力母数を入れることで，IIC 曲線の傾きもデータに合わせて調整できるようになりました。識別力母数のデフォルトは (推定しないのであれば)1.0 ですから，ここから大きく逸脱するような項目には注意です。今回は V8 をみると，困難度が非常に高いこともわかりますが，識別力が非常に低くなっているので，この項目に誤答したからと言ってすぐさま成績が悪い，と判断できるほどではないことがわかります。

最後の一行に Log.Lik とありますが，これは**対数尤度 (log likelihood)** を表しています。これが大きいほど，データとモデルの適合度が高いことを意味します。1PL のときの Log.Lik が -2468.743 でしたから，1PL モデルより 2PL モデルの方が当てはまりがよかったと言えるでしょう。

プロットによる出力も見ておきましょう (図 11.2)。識別力母数が入ったことで，ICC の傾きが項目によってさまざまになりますし，IIC や TIC の見た目もすっかり変わってしまいましたが，読み取り方は同じです。

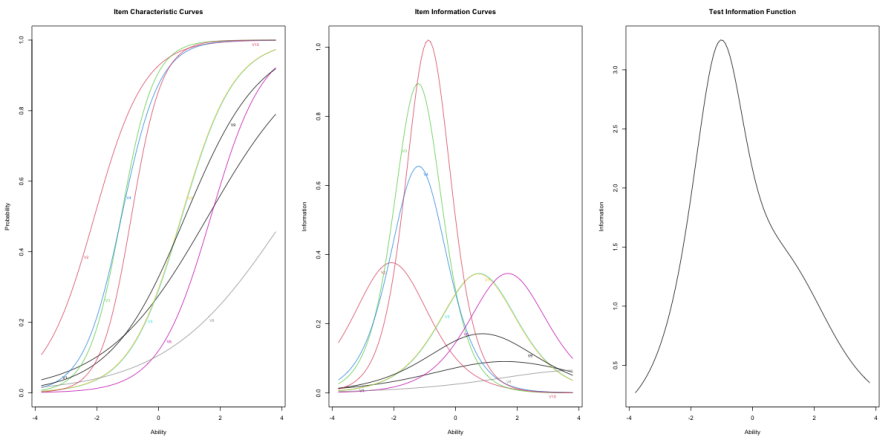

図 11.2　2PL のさまざまなプロット。左から ICC,IIC,TIC

3PL モデルの場合

最後に `ltm` パッケージを使って，3PL モデルを実行していきましょう。`code:11.4` を実行してください。

code：11.4　3PL モデルの実行

```
1  result.3pl <- tpm(dat)
2  print(result.3pl)
3  plot(result.3pl,type="ICC")
4  plot(result.3pl,type="IIC")
5  plot(result.3pl,type="IIC",items = 0)
```

コードは一行目の関数以外ほとんど同じですので，逐一解説は致しませんが，出力のところを見ておきましょう (出力 11.8)。

R の出力 11.8: 3PL モデルの結果出力

```
Call:
tpm(data = dat)

Coefficients:
     Gussng  Dffclt  Dscrmn
V1    0.230   1.407  15.123
V2    0.638  -0.610   2.232
V3    0.225  -0.825   2.907
V4    0.324  -0.637   2.324
V5    0.149   0.931   2.992
V6    0.043   1.613   1.706
V7    0.074   0.879   1.509
V8    0.000   4.490   0.471
V9    0.133   1.144   1.255
V10   0.278  -0.438   3.085

Log.Lik: -2431.371
```

3PL モデルは困難度，識別力に加えて**当て推量母数**が計算されます。これは ICC の形 (図 11.3 の左) を見ればわかりますが，曲線全体が底上げされたようになっているものがあります。この最低ラインの高さが当て推量母数の大きさです。図や数値から，V2 のそれが大きい数字であることが読み取れます。ICC は θ の関数である正答率ですが，これが θ の値にかかわらず一定の値を取るということは，どれだけ能力が低くてもその確率で正答してしまうことを意味しています。なので「適当に推測して答えを当てられる大きさ」という意味で当て推量母数と呼ばれているのです。

今回当て推量母数を含めて考えたことで，IIC や TIC がまた大きく変化しました。この数字が高すぎる V2 は，テストとしてはあまり良い項目ではなかったのかもしれません。また，モデルの適合度としては 3PL モデルが最もよかったのですが，2PL モデルとそれほど大きく違うわけでもありません。どのモデルでどのように推定するべきか，推定されたモデルはどういう仮定を持っているのかについて，しっかり理解した上で利用するようにしましょう。

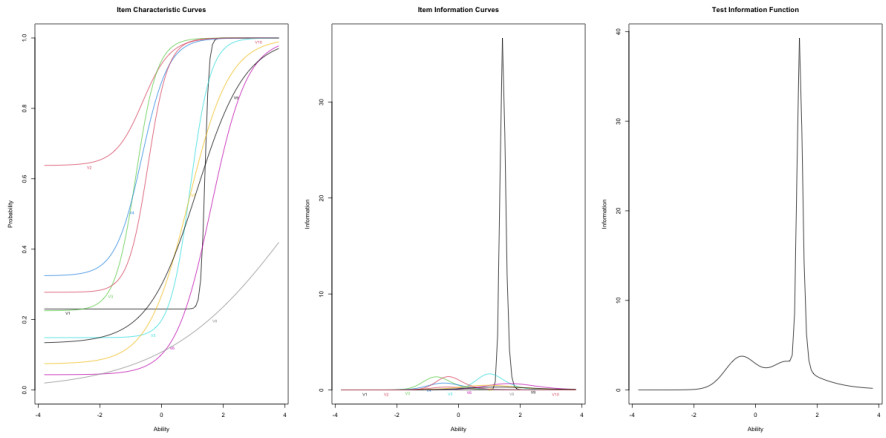

図 11.3 3PL のさまざまなプロット。左から ICC,IIC,TIC

11.2 段階反応モデルの実際

続いて**段階反応モデル**の練習に進みましょう。項目反応理論は正答/誤答の二値データに対して用いるものでしたが，これを多段階反応に拡張したモデルの 1 つが段階反応モデルです。多段階反応とは，「まったく当てはまらない」を 1，「非常に当てはまる」を 7 とする，といったような**リッカート尺度**のようなスタイルの調査項目です。反応させるカテゴリの段階数に応じて，5 件法とか 7 件法と呼ばれたりします。

リッカート尺度の場合は，シグマ法をつかって累積度数からカテゴリに該当する値を算出し，それを元に態度得点を作るという方法でした。とはいえこのやり方でやっても，いい尺度であればそのまま素点を尺度値としてもほとんど問題がないため，シグマ法が回りくどい方法だと思われたのか，使われなくなってしまいました。しかし実際には「いい尺度であれば」という条件を満たしているかどうか，しっかり確認しなければなりません。歪んだ分布や偏ったカテゴリ反応なのに，素点をそのまま尺度値とするのは不適切なことがあります。また，リッカート尺度で撮られたデータは因子分析をすることによって，解釈度に分類されて，因子ごとのスコアを算出して利用されることが一般的です。しかし，尺度値が適切でなければ，ピアソンの積率相関係数も適切ではなく，そこから計算が始まる因子分析のモデルにも疑義が生じます。

カテゴリ反応はあくまでも並びの問題であって，**順序尺度水準**の情報しか持たないと考えるのであれば，その背後にある態度のような連続体を仮定し，この態度 θ が大きくなることで徐々に反応カテゴリが変わる，という**段階反応モデル**のほうが適切な分析方法になるわけです。

項目反応理論 (の段階反応モデル) を使って確認することは決して難しくありません。今回は因子分析の時に使った `bfi` データの一部を使って段階反応モデルを実践してみましょう。一部を使うとしたのは，段階反応モデルは項目反応理論の中間ですから，潜在変数が 1 次元 (一因子) である，という仮定があるからです。ここでは `N` 因子 (Neuroticism, 神経質さ) を使った例を示しています。

code : 11.5 GRM の実行

```
1  library(psych)
2  dat <- bfi %>%
3    dplyr::select(-gender, -education, -age) %>%
4    dplyr::select(starts_with("N"))
5  result.grm <- grm(dat)
6  result.grm
```

```
 7  plot(result.grm, type = "ICC", items = 1)
 8  plot(result.grm, type = "IIC", items = 1)
 9  plot(result.grm, type = "ICC", items = 4)
10  plot(result.grm, type = "IIC", items = 4)
11  plot(result.grm, type = "IIC", items = 0)
```

■コード解説

1 行目 psych パッケージを読み込む。サンプルデータ bfi を用いるためです。

2-4 行目 データの加工。bfi データから，性別，最終学歴，年齢などの情報を取り除き，また N で始まる変数だけ選出 (select) しています。

5 行目 段階反応モデルの実行。

6 行目 結果の出力。

7 行目 項目 1 についての ICC，厳密には**項目反応カテゴリ特性曲線 (Item Response Category Characteristic Curve)** と呼びます。

8 行目 項目 1 についての IIC を表示させています。

9 行目 項目 4 についての ICC を表示させています。

10 行目 項目 4 についての ICC を表示させています。

11 行目 項目群についての TIC を表示させています。

code:11.5 の 6 行目で，分析結果の数値的出力をさせていますので，それをみておきましょう (出力 11.9)。

ここで，Dscrmn とあるのは識別力母数ですが，困難度母数に対応するのが Extrmt1 から Extrmt5 にあたります。この尺度は 6 件法ですので，反応カテゴリが変わる切れ目，すなわち**閾値**が 5 つあるのです。これらが 2PL モデルでいうところの困難度に対応すると言ってもいいでしょう。

R の出力 11.9: GRM モデルの結果出力

```
Call:
grm(data = dat)

Coefficients:
    Extrmt1  Extrmt2  Extrmt3  Extrmt4  Extrmt5  Dscrmn
N1   -0.810   -0.093    0.342    0.985    1.720   3.125
N2   -1.366   -0.555   -0.111    0.648    1.483   2.890
N3   -1.187   -0.298    0.123    0.876    1.767   2.026
N4   -1.564   -0.355    0.238    1.240    2.280   1.277
N5   -1.296   -0.125    0.494    1.479    2.530   1.112

Log.Lik: -21721.91
```

数値を見ていてもピンとこないかもしれませんので，図でこの各項目の特徴を確認しておきましょう。変数 N1 と N4 の IRCCC を図 11.4 に示しました。この図に表されている複数の曲線が，θ に対応した各カテゴリに反応する確率を表しています。図の左，N1 は綺麗なカーブが描かれており，各カテゴリのピークが見て取れますが，N4 は反応カテゴリ 3 の山が埋もれてしまっていることがみてとれます。この項目については，6 段

階ではなく 5 段階で反応を求めるのがよかったのかもしれません[*2]。

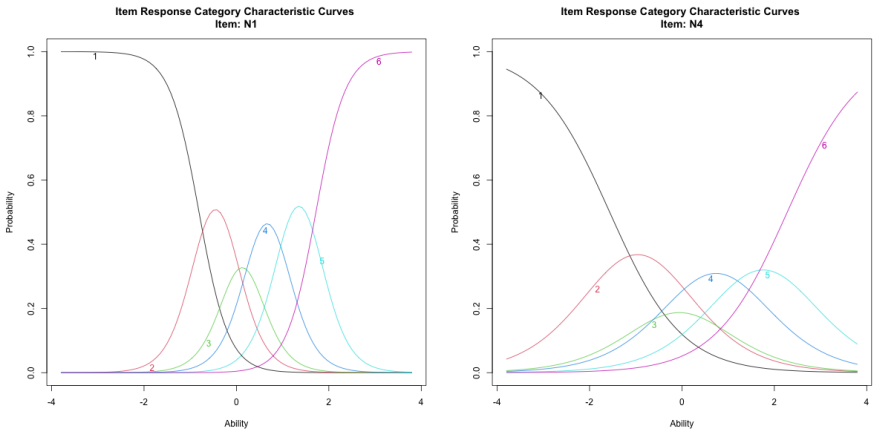

図 11.4 GRM の IRCCC

原理的には難しいことのように感じますが，コードを書いて実践する分にはほとんど苦労がなく，二値モデルと同じように考えることができます。コードには IIC や TIC を描くものもありますが，1PL,2PL,3PL モデルと変わらない感じで多段階モデルが実行できますね。みなさんも心理尺度を使う時は，盲目的にスコアリングし，因子分析するのではなく，どういう仮定の元でどういう計算をしているかを理解しながら，データや調査協力者にとって最適な分析をするように心がけてください。

11.3 カテゴリカル因子分析との対応

さて多段階反応による分析も簡単にできることがわかりましたが，気になるのは段階反応モデルが一因子構造しか仮定していないところです。bfi データはビッグファイブと呼ばれるように五因子 (5 次元) の仮定があるのですが，一因子ずつ `grm` を実行し，あとで統合するのは大変です。

でも大丈夫。簡単に多次元に展開する方法があります。いわゆる一般的な因子分析は，ピアソンの積率相関係数を元に (積率相関係数の相関行列 $\boldsymbol{R}$ の固有値分解から) 因子を求めていくのですが，この積率相関係数が**間隔尺度水準**を仮定したもので，シグマ法で計算された尺度値を使っていないのであれば疑義が残る，という話でした。この元になる相関係数が，潜在的連続体を仮定した順序尺度水準向けの相関係数である，**ポリコリック相関係数**であれば，その問題は解決します。反応段階が順序尺度を仮定した因子分析のことをとくに**カテゴリカル因子分析**と言います。カテゴリカル因子部な席は，前回紹介した `psych` パッケージに少しオプションを足すだけで簡単に実行できるものです。

code : 11.6 GRM の実行

```
1  dat <- bfi %>% dplyr::select(-gender, -education, -age)
2  result.fa <- fa.poly(dat, nfactors = 5, rotate = "geominQ")
3  print(result.fa, sort = T, cut = 0.3)
```

■コード解説

1 行目 サンプルデータ `bfi` の加工をしています。

[*2] ちなみに `N1` は「すぐ怒る (Get angry easily)」，`N4` は「割と凹む (Often feel blue)」です。日本語訳は小杉の意訳で定訳ではありません。あしからず。

2 行目　ポリコリック相関係数に基づく因子分析の実行

3 行目　結果の出力。

コードの 2 行目を見ていただくとわかる通り，ただの `fa` 関数ではなく `fa.poly` としただけで，あとは因子分析と同じです。この `.poly` のところがポリコリック相関係数を使うよう指定しているところで，これで実質的に測定尺度水準が順序尺度水準であると想定した因子分析をしたことになっています。

カテゴリカル因子分析は，段階反応モデルと数学的に等価であることが知られています。出自が違いますので，分析の数字や出力方法に違いがありますが[*3]，実質的には同じことをしていると理解してください。また，R ではたった数文字追加するだけで適切な分析方法になるのですから，使わない手はないと思います。

カテゴリカル因子分析は，因子分析の文脈から順序尺度水準の項目を扱えるように，と発展してきたものです，これに対して，項目反応理論の文脈から心理尺度を扱えるように，と発展してきたルートもあります。これは**多次元項目反応理論 (multi-dimensional item response theory)** と呼ばれるもので，これを提供する R パッケージもあります。それが `mirt` パッケージ Chalmers (2012) です。最後にこれを使ったコードの例を示しておきます。この計算には時間がかかりますが，完全情報最尤推定によって項目反応理論で算出するような，精度の高い**因子得点**を得ることができるのは利点だと言えるでしょう。

code : 11.7　mirt の実行

```
1  library(mirt)
2  result.mirt <- mirt(dat, 5)
3  print(result.mirt)
4  result.mirt %>% summary(rotate = "geominQ")
```

11.4　課題

今日の授業でおこなったすべての次の計算をする R コードを提出してください。ファイル形式は R スクリプトか Rmd とします。なお提出されたコード単体でバグがなく動くことが確認できないものは，未提出扱いになります。コードの書き方などわからないところがあれば，曜日別 TA か小杉までメールで連絡し，指導を受けてください。

[*3] たとえば因子分析の文脈では，因子負荷量を提示して単純構造を目指すのに対し，項目反応理論の文脈では IRCCC を描いたり TIC を描いたりして各反応カテゴリの特徴を精査します。

第 12 章

構造方程式モデリング

本項では**構造方程式モデリング (Structural Equation Modeling, SEM)** について説明していきます[*1]。この手法はこれまで学んできた因子分析や回帰分析を統合したもので，分散共分散行列に方程式を作り込んでいくものです。我々が調査で得たデータから得られる情報は，分散共分散行列がすべてですから，その変数間関係を細かく作り込んでいき，行列の隅々まで考え尽くすという意味で，究極の手法といえるでしょう。実際，SEM は因子分析や回帰分析の他にも，多くのモデルをその下位モデルとして包含します。言い換えれば，SEM が登場する前に考えられてきたさまざまなモデルが，SEM の枠組みで理解・表現できるようになりました。であれば，この究極の技さえ知っておけば，非常に広範囲に拡張できるわけですから，こんなに便利なことはありません。

このように技術的には非常に高度なものでありますが，これを使うのは意外と簡単にできます。モデルをイメージで表現できる，パスダイアグラムの考え方から入っていきましょう。

12.1 パスダイアグラムの書き方

変数間の関係を表す図を**パス図**あるいは**パスダイアグラム (Path Diagram)** と言います。

分析する際の変数には**観測変数 (Observed Variables)** と**潜在変数 (Latent Variables)** があります。観測変数はその名の通り，観測された変数であり，データとして数値化されたものになります。これに対して潜在変数とは，モデルで仮定された変数のことです。たとえば因子分析やテスト理論では，因子や学力といったものを想定します。それが潜在変数です。潜在変数はデータとして数値化されているのではなく，抽象的な概念ですが，図にする場合はそれも表現しなければなりません。そこで観測変数を矩形 (長方形) で，潜在変数を楕円で表現することにします。

また，変数同士の関係を表現する必要がありますが，関係には**因果関係**と**相関関係**があります。調査データから因果関係を見出すのは原理的に不可能ですが，モデル上は一方が他方の原因になっている，と考えることがあります。回帰分析はその典型例で，説明変数が原因となって，被説明変数のあたいが結果的に変わる，という関係ですね。こうした関係を一方向の矢印で表現します。他方，相関関係は因果関係よりも緩やかで，何らかの関係がある，ということを意味しているに過ぎません。因果の向きが定まっているのではなく，どちらの向きにも影響しうる，ということで双方向矢印でもってこれを表現します。

図 12.1 にこの表記方法をまとめました。準備は実はこれだけです。この表記方法を使って，さまざまなモデルが統合的に表現できます。たとえば**回帰分析**を考えてみましょう。説明変数も被る説明変数も観測されている変数を使います。一方が他方の因果的関係にあるので，パスダイアグラムで表現すると図 12.2 の上図

[*1] この手法は別名**共分散構造分析**という名前でも呼ばれています。書籍や論文を検索する時は，こちらの名称もキーワードにしてみてください。

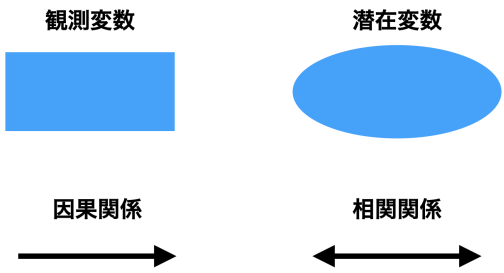

図 12.1　変数と関係の表記法

のようになります。ここで**残差**はデータになく，モデルの計算結果から想定される潜在的な変数なので楕円で描かれていることに注意してください。また**重回帰分析**は図 12.2 の下図のようになります。説明変数が複数に増えただけですので，表記上はこのように変わるわけですね。

なお，この因果関係を表す矢印の上に (偏) 回帰係数を書いて，影響力の大きさを表現することがあります。構造方程式モデリングでは，この矢印によって表される影響力の強さは**パス係数 (path coefficient)** という呼び方で統一されます。

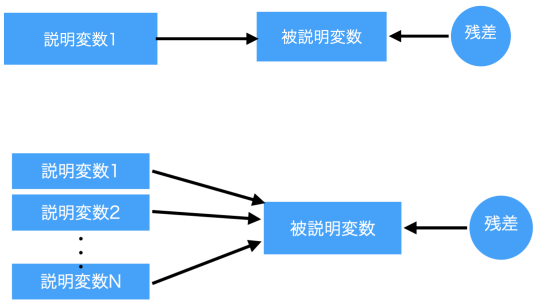

図 12.2　回帰分析と重回帰分析

では**因子分析**をパスダイアグラムで表現するとどうなるでしょうか。複数の項目の背後にある共通要因を潜在変数として仮定するので，パスダイアグラムで表現すると図 12.3 のようになります。これは 1 因子モデルですが，多因子モデルは基本的に左側にくる潜在変数が複数になるだけです。潜在変数 (因子) からのパス係数は，**因子負荷量**なのです。また図 12.2 と図 12.3 を見比べると，回帰分析と因子分析も形はよく似て

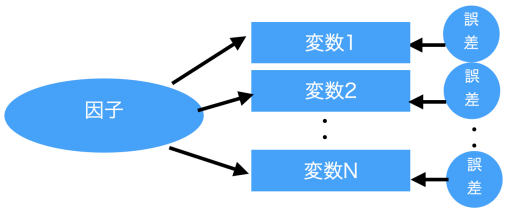

図 12.3　因子分析モデル

いることがわかります。両者の違いは，説明変数が観測変数か潜在変数かであり，因子分析とは説明変数が潜在変数になった回帰分析モデル，ということもできるのです。

このように，今まで見てきた統計モデルをパスダイアグラムを使って表現できます。

12.2 パスダイアグラムによるさまざまなモデル

パスダイアグラムはさまざまな分析法を統合的に表現するツールであることがわかりました。この方法を使って他の統計モデルも見てみましょう。

■主成分分析 このコースではここまで取り上げてきませんでしたが，因子分析とよく似た手法で**主成分分析 (Principle Component Analysis)** というのがあります。主成分分析の目的は観測変数を重みつき線型結合させ，個々のデータをもっともよく説明するような主成分という合成変数を作り出すことです。たとえば体力テストなどで複数の競技の記録をつけた後，総合的に誰が一番運動能力があるのか，というときにこの分析が使われたりします。この分析方法をパスダイアグラムで書くと図 12.4 のようになります。

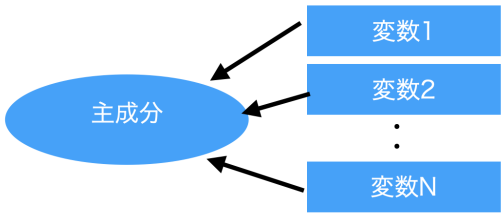

図 12.4 主成分分析モデル

これを見ると，因子分析と似ている，でもちょっと違う，というのがよくわかりますね。因子分析は潜在的な説明変数を作るのですが，主成分分析は潜在的な被説明変数を作る分析方法なのです。

その違いは，誤差の扱いにも関わってきます。これまでの例にあるように，パスダイアグラムでは基本的に矢印が入った変数には誤差 (残差) がつくのが決まりです。因子分析は説明変数が潜在的だったので，観測変数にはそれで説明できない残りが潜在変数として構成されました。主成分分析は観測変数が矢印の出所になりますので，そこには誤差が想定されません。作られた潜在変数は矢印が入ってきた方ですが，これはモデルで構成される変数なので誤差を考えることができません[*2]。主成分分析はデータに誤差を考えないときに有用なモデルですから，経済学などデータがはっきりしたものである領域でよく用いられてきました。これに対して心理統計の領域では，測定されたスコアに誤差が入ると考えるのが当然ですから，因子分析の方がよく用いられてきたのです。主成分分析と因子分析は数学的にはよく似た解法で答えを出すのですが，その背後にある考え方がまったく異なります[*3]。

■尺度水準の違いによるモデルの違い たとえば因子分析において観測変数が**順序尺度水準**であれば，同じパスダイアグラムであってもそれはカテゴリカル因子分析をしたことになります。パスダイアグラムでは変

[*2] 成分も誤差もモデルから構成されるものなので，モデル上それらを区分する方法がないからです。

[*3] もう少し丁寧にいうと，主成分分析も因子分析も計算する上では正方行列の固有値分解をすることになります。ここで，因子分析の場合は項目に誤差を考えますから，固有値分解をするときに相関行列 $\boldsymbol{R}$ ではなく，その対角項に共通性を入れた $\boldsymbol{R}\dagger$ から計算を始めるのでした。主成分分析は誤差を考えませんから，$\boldsymbol{R}$ から計算したり，単位に意味がある場合が少なくないので分散共分散行列 $\boldsymbol{S}$ の固有値分解でもってパス係数を算出するという違いです。因子分析において共通性の推定を避け，$\boldsymbol{R}$ を分解する方法をとくに因子分析の主成分分解というのが，初学者を混乱させるポイントになっています。

数の尺度水準を表現することはありませんが，言い換えると尺度水準が違うだけで別の分析モデルになるということです。

たとえば回帰分析において，被説明変数が離散的 (**名義尺度水準**) なモデルは，かつて**判別分析 (Descriminant Analysis)** と呼ばれていました。同様に被説明変数が**順序尺度水準**で得られた場合は，**順序ロジットモデル (Ordinal Logit Model)** や**順序プロビットモデル** Ordinal Probit Model と呼ばれたモデルです (2 水準ならロジットモデル，3 水準以上はプロビットモデル)。

また回帰分析において，説明変数が離散的 (名義尺度水準) なモデルは，**分散分析 (ANalysis Of VAriance;ANOVA)** と呼ばれていたのでした。とくに説明変数が 2 カテゴリの場合は **t 検定 (t-test)** として知られています。これらは平均値差の検定という文脈で説明されてきたかと思いますが，いずれも回帰分析，もとい**線形モデル**の一種であるというのはご存知の通りです。

色々な分析法，分析モデル名がありますが，**パスダイアグラム**で表現すれば同じである，というのがポイントです。**構造方程式モデリング**はその下位モデルとしてこれらの分析をすべて含む，あるいは統合的な表現であるというのはそういう意味です。このように統合される以前は，それぞれの分析方法をそれぞれのソフトウェアや関数で実行する必要があったのですが，今は SEM が扱えるソフトウェア，関数 1 つで分析できるのです。大変ありがたいことではないですか！考えるべきポイントもすべてパス係数や適合度といった統合的指標でよいのですから。

■自由なモデルへ　因子分析と回帰分析，あるいはその他のさまざまな統計モデルが，パスダイアグラムによって統合的に表現できるようになりました。その利点は表現が統一化されただけではなく，同じプラットフォームで表現できるのですから，たとえば図 12.5 のような表現もできるということです。

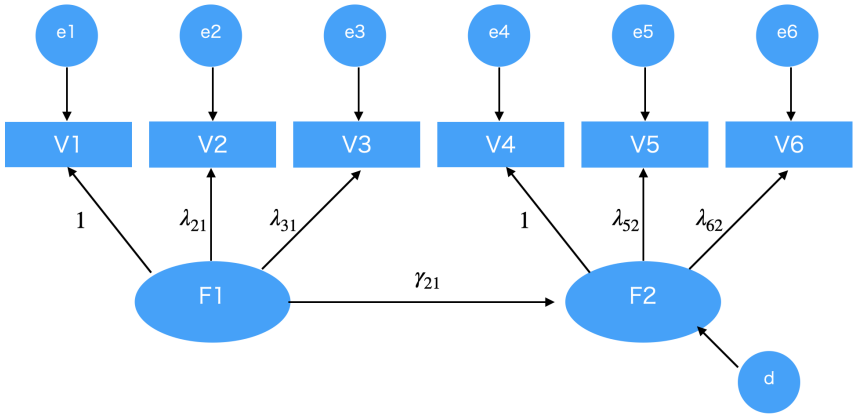

図 12.5　構造方程式と測定方程式。記号の意味は後述

図 12.5 には因子分析がふたつと，因子と因子の関係を矢印で結んだパスが描かれています。潜在変数の間に回帰分析を組み込んだようなものですね。イメージとしては，ある心理変数が別の心理変数に影響を与える (ex. 外向性が抑うつ傾向に影響する，など) といったものを表しているわけです。心理学者は心理的変数がどう影響しあっているか，関係しあっているかをみていきたいわけですから，やりたいことは要するにこういう複合的なモデルによる検証だったのだ，と言っても過言ではないでしょう。このような分析も，SEM のおかげで簡単にできるようになりました。ここで観測変数から潜在変数を構成する箇所をとくに**測定方程式**，潜在変数同志の因果関係などそれ以外の関係を表現する箇所を**構造方程式**と呼んで区別することがありますが，**構造方程式モデリング**はその両者からなるモデル全体のこと，ということができるでしょう。

構造方程式モデリングを使うと，調査などから得たデータ全体の見取り図を描いて分析できるのです。言い換えると，実際に調査を実施する前に，どの変数とどの変数がどのような関係にあるのか，という仮定を考えて調査票をデザインし，過不足なく測定してモデルを検証する，という**仮説検証型**の分析ができるのです。それまでの調査は，どういう変数がどういう関係になっているか細かくは分からないけれども，何か関係しそうだからまとめて調査項目にしておいて，因子分析や回帰分析を繰り返して，変数間の関係を**探索的に**研究するということがよく行われていました。もちろん今でも探索的研究はあるのですが，既存の尺度や先行研究がある場合はより具体的に調査を設計できるようになりました。

因子分析も，因子数がわからないのでスクリープロットなどを描いて探し出し，因子と項目の関係がわからないのですべての因子がすべての項目に影響すると考えて分析する**探索的因子分析 (Exploratory Factor Analysis,EFA)** がかつては主流でしたが，今ではどの因子がどの項目で測定されるかを明確にパスダイアグラムで表現して分析する，**検証的因子分析 (Confirmatory Factor Analysis, CFA)** が行われるようになってきています。

12.3 構造方程式モデルによる未知数の推定

12.3.1 未知数は本当に推定できるのか

さて，さまざまなモデルをパスダイアグラムで表現できる，ということがわかりましたが，「方程式」という言葉はどこからきているのでしょう？実は，統計モデルはパスダイアグラムでも表現できますが，同じことを方程式でも表現できます。

たとえば図 12.1 で表した回帰分析モデルは，次のような方程式で表せるのでした。

$$y = b_0 + b_1 x + e$$

図 12.2 に表した因子分析モデルも，同様に次のような方程式で表せるのでした (項目が 3 つの場合)。

$$\begin{cases} x_1 &= \lambda_1 f + e_1 \\ x_2 &= \lambda_2 f + e_2 \\ x_3 &= \lambda_3 f + e_3 \end{cases}$$

ここで f は因子得点，λ_j は因子負荷量，あるいはパス係数を表しています。

この因子分析モデルを例に，この方程式をどのように解くかをみていきましょう。そのために，いくつかの記号と特徴を確認します。まずベクトルの平均を関数 $\boldsymbol{E}$ で表すことにします。誤差の平均は 0，という特徴は $\boldsymbol{E}[\boldsymbol{e}] = \boldsymbol{0}$ と表すことができます。因子得点も標準化されていると仮定しますので，$\boldsymbol{E}[\boldsymbol{f}] = \boldsymbol{0}$ ですね。つぎにベクトルの分散を関数 $\boldsymbol{V}$ で表すとします。データベクトル $\boldsymbol{x}$ が平均 0，あるいは平均偏差，あるいは標準化されていると考えると，$\boldsymbol{V}[\boldsymbol{x}] = \boldsymbol{E}\left[(\boldsymbol{x} - \boldsymbol{E}[\boldsymbol{x}])^2\right] = \boldsymbol{E}[\boldsymbol{x}\boldsymbol{x}']$ は分散共分散行列 $\boldsymbol{\Sigma}$ だと考えることができます。また因子は標準化されているので $\boldsymbol{V}[\boldsymbol{f}] = 1$，誤差同士は相関しないものの，誤差には分散がありますので $\boldsymbol{V}[\boldsymbol{e}] = \boldsymbol{E}[\boldsymbol{e}\boldsymbol{e}'] = \Psi$ とします。ここで Ψ は対角に誤差分散が入った対角行列です。

さて，さきほどの 3 項目 1 因子モデルを行列で表現してみましょう。各要素を次のようにベクトルでまとめて表現します。

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix}, \boldsymbol{e} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

すると式は次のようにあっさり表現できます。

$$\boldsymbol{x} = \boldsymbol{\Lambda}\boldsymbol{f} + \boldsymbol{e}$$

ここからデータの分散共分散行列 $\boldsymbol{\Sigma}$ を考えてみましょう。

$$
\begin{aligned}
\boldsymbol{V}[\boldsymbol{x}] &= \boldsymbol{E}[\boldsymbol{x}\boldsymbol{x}'] \\
&= \boldsymbol{E}[(\boldsymbol{\Lambda}\boldsymbol{f}+\boldsymbol{e})(\boldsymbol{\Lambda}\boldsymbol{f}+\boldsymbol{e})'] \\
&= \boldsymbol{E}\left[(\boldsymbol{\Lambda}\boldsymbol{f}+\boldsymbol{e})(\boldsymbol{f}\boldsymbol{\Lambda}'+\boldsymbol{e}')\right] \\
&= \boldsymbol{E}\left[\boldsymbol{\Lambda}\boldsymbol{f}^{\boldsymbol{2}}\boldsymbol{\Lambda}'+\boldsymbol{\Lambda}\boldsymbol{f}\boldsymbol{e}'+\boldsymbol{e}\boldsymbol{f}\boldsymbol{\Lambda}'+\boldsymbol{e}\boldsymbol{e}'\right] \\
&= \boldsymbol{\Lambda}\boldsymbol{\Lambda}'+\Psi
\end{aligned}
$$

これをエレメントワイズで書き出すと次のようになります。

$$
\begin{aligned}
\begin{pmatrix} s_1^2 & s_1s_2 & s_1s_3 \\ & s_2^2 & s_2s_3 \\ & & s_3^2 \end{pmatrix} &= \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} + \begin{pmatrix} \sigma_1^2 & & \\ & \sigma_2^2 & \\ & & \sigma_3^2 \end{pmatrix} \\
&= \begin{pmatrix} \lambda_1 1^2+\sigma_1^2 & \lambda_1\lambda_2 & \lambda_1\lambda_3 \\ & \lambda_2^2+\sigma_2^2 & \lambda_2\lambda_3 \\ & & \lambda_3^2+\sigma_3^2 \end{pmatrix}
\end{aligned}
$$

この式の左辺はデータから計算された分散共分散行列，右辺はモデルから計算された分散共分散行列です。行列の下三角が消えているのは対称行列で同じ情報が入るからで，データからオリジナルに得られるのは 3 つの分散 (s_1^2, s_2^2, s_3^2) と 3 つの共分散 (s_1s_2, s_1s_3, s_2s_3) の情報になります。一方右辺での未知数はパス係数 (因子負荷量) の $\lambda_1, \lambda_2, \lambda_3$ と誤差分散 $\sigma_1^2, \sigma_2^2, \sigma_3^2$ ということになります。未知数の数と既知数の数が同じなので，この方程式は解けます。とくに，未知数と既知数の数が一致しているので，**丁度識別 (just identification)**，あるいは単に**識別可能 (identifiable)** といいます。

分散共分散行列で提供される情報の増え方に比べて，モデルの未知数の増え方は小さいので，一般的な SEM のモデルのほとんどは**過剰識別**になります。つまりいくつかの答えがあり得る，ということになるので，その時は尤度が最も高くなるように，といった基準を加えて答えを一意に定めます。逆に未知数の方が多い場合**識別不可能** (解が求められない) ということになり，その場合はどこかのパラメータを値として入れてやる (**制約**をかける，といいます) 必要があります。

12.3.2 複雑なモデルの場合

では次に，図 12.5 のような複雑なモデルでも方程式で表せることを示しましょう。

ここで図 12.5 の係数について説明します。ギリシア文字 λ や γ がパス係数を表しています。パス係数の添字は，被説明変数の変数番号 j と，説明変数の変数番号 k をつかって λ_{jk} のようにかき，これで $k \to j$ の影響力の大きさを表していると思ってください。測定方程式の係数を λ で，構造方程式の係数を γ で表現しました。また $\lambda_{11}, \lambda_{42}$ になるべきところが 1 になっていますが，これは「潜在変数から出るパスのうち，1 つは必ず 1 にしないと推定できない」という数学的特徴があるからです。決まりごとだと思ってください。

それを踏まえて，モデルの方程式を書くと，次のようになります。

$$
\begin{cases}
V_1 &= f_1 + e_1 \\
V_2 &= \lambda_{21}f_1 + e_2 \\
V_3 &= \lambda_{31}f_1 + e_3 \\
V_4 &= f_2 + e_4 \\
V_5 &= \lambda_{52}f_2 + e_5 \\
V_6 &= \lambda_{62}f_2 + e_6 \\
f_2 &= \gamma_{21}f_1 + d
\end{cases}
$$

ややこしいだけで，書けるのは書けましたね！そして当然，これは行列表現した方が楽ではないか，というアイデアが浮かぶと思います。その通りで，エレメントワイズにまずは書いてみましょう。

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ f_2 \\ f_1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{21} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{31} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{52} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{62} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma_{21} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ f_2 \\ f_1 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ d \\ f_1 \end{pmatrix}$$

左辺のベクトルの最後に f_1 が入っているところがポイントで，これは方程式的には $f_1 = f_1$ というだけの式なのですが，これを入れたおかげで全体を整合的に表現できましたね。この左辺のベクトルを $\boldsymbol{v}$ として，式全体を $\boldsymbol{v} = \boldsymbol{G}\boldsymbol{v} + \boldsymbol{e}$ と考えてみましょう。すると次のように展開できます。

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{G}\boldsymbol{v} + \boldsymbol{e} \\ \boldsymbol{v} - \boldsymbol{G}\boldsymbol{v} &= \boldsymbol{e} \\ (\boldsymbol{I} - \boldsymbol{G})\boldsymbol{v} &= \boldsymbol{e} \\ (\boldsymbol{I} - \boldsymbol{G})^{-1}(\boldsymbol{I} - \boldsymbol{G})\boldsymbol{v} &= (\boldsymbol{I} - \boldsymbol{G})^{-1}\boldsymbol{e} \end{aligned}$$

ここで $\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{G})$ とすると，

$$\boldsymbol{v} = \boldsymbol{P}^{-1}\boldsymbol{e}$$

ということになります。これをエレメントワイズで書くと次のようになります。

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ f_2 \\ f_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -\lambda_{21} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\lambda_{31} \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -\lambda_{52} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -\lambda_{62} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -\gamma_{21} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ d \\ f_1 \end{pmatrix}$$

さて，変数からなるベクトル $\boldsymbol{v}$ ではありますが，観測変数は V_1 から V_6 までしかありませんから，そこだけを取り出す行列 $\boldsymbol{F} = \{\boldsymbol{I}, \boldsymbol{O}\}$ を考えます。ここでは次のような行列です。

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

これを使って，次のように関係を書き直すことができます。

$$\boldsymbol{g} = \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{pmatrix} = \boldsymbol{F}\boldsymbol{P}^{-1}\boldsymbol{e}$$

さて，ではデータだけのベクトル $\boldsymbol{g}$ から分散共分散行列を構成するとしましょう。データベクトル $\boldsymbol{g}$ は平均が 0 だと仮定します[*4]。そうすると分散共分散行列 $\boldsymbol{\Sigma}$ は次のようになります。

$$\begin{aligned}\boldsymbol{\Sigma} &= \boldsymbol{g}\boldsymbol{g}' \\ &= \boldsymbol{F}\boldsymbol{P}^{-1}\boldsymbol{e}(\boldsymbol{F}\boldsymbol{P}^{-1}\boldsymbol{e})' \\ &= \boldsymbol{F}\boldsymbol{P}^{-1}\boldsymbol{e}\boldsymbol{e}'\boldsymbol{P}^{-1'}\boldsymbol{F}'\end{aligned}$$

ここで式の中にある $\boldsymbol{e}\boldsymbol{e}'$ は誤差などの未知数で作った分散共分散行列になっています。誤差は他の変数との共分散がないと考えられるので，これは結局対角行列です。

$$\boldsymbol{e}\boldsymbol{e}' = \begin{pmatrix} V(e_1) & & & & & & & & \\ & V(e_2) & & & & & & & \\ & & V(e_3) & & & & & & \\ & & & V(e_4) & & & & & \\ & & & & V(e_5) & & & & \\ & & & & & V(e_6) & & & \\ & & & & & & V(d) & & \\ & & & & & & & V(f_1) & \end{pmatrix}$$

さあややこしい話はここまでにして，何を考えていたのかをもう一度考えてみましょう。

この式の左辺は $\boldsymbol{\Sigma}$，すなわちデータからできる分散共分散行列であり，具体的に数字の計算のできるデータセットです。この式の右辺は $\boldsymbol{F}\boldsymbol{P}^{-1}\boldsymbol{e}\boldsymbol{e}'\boldsymbol{P}^{-1'}\boldsymbol{F}'$ であり，係数と潜在変数の分散からなる方程式です。左辺にデータ，右辺にモデルで表した分散共分散行列があり，これがイコールで結びついているのですから，この連立方程式を解けば，係数を求めることができます。

そりゃその計算はとても面倒なことになると思われますが，そこは機械がやってくれるのだ，と割り切ればいいでしょう。モデルがデータに最も合うように，未知数を決めていく方法は，**最小二乗法**や**最尤法**など，今まで学んできた手法を使えばよいのです！

こうして自由なモデリング技術とその解法を私たちは手に入れることができました。構造方程式モデリングとは，変数間関係を細かく作り込んでいき，分散共分散行列の隅々まで考え尽くす究極のモデル，と意味が伝わったでしょうか。

12.4 適合度によるモデルの評価

データ行列とモデル行列をイコールで結んだ方程式を解くことが，未知数を求める根本的な原理であるということがわかりました。方程式が多く未知数が少ない場合，値が一意に定まらなくなってしまいます[*5]。これはパラメータに**自由度 (degrees of freedom)** がある，ということもできます。そのような場合は別の基準をつかって最も良いと考えられる数字を推定値として用いることになります。

さてなんとかモデルをデータに当てはめることができたとして，それがどの程度一致しているか，ということは別次元で評価しなければなりません。それが**適合度 (fit index)** です。SEM の文脈では複数の適合度が考えられており，それらを総合的に評価するというやり方がとられています。ここでは複数の適合度指標をご紹介しておきます[*6]。

[*4] ずいぶんと勝手な仮定だな，と思うかもしれませんが，分析の元になる変数ベクトルを標準化したと思っていただければそれほどおかしなことでもない，と考えてもらっても構いません。あるいは単に変数ごとの平均をとった平均偏差ベクトルを作って，それをデータだとしたと考えてください。

[*5] たとえば $x+5=8$ と $x-3=9$ という連立方程式があればどうでしょうか。未知数は 1 つ，式は 2 つなのでどちらが正解なのかわかりません！

[*6] 以下の記述は小杉・清水 (2014) をもとに加筆修正したものです。

■GFI・AGFI　**GFI(Goodness of Fit Index)** や **AGFI(Adjusted GFI)** は，0 から 1 までの値を取る適合度指標で，因果モデルがデータの何 % を説明したかの指標になります。1 に近いほど良いモデルで，GFI で 0.95,AGFI で 0.90 以上の数字であると良い適合と判断されます。

■**情報量規準**　**情報量規準 (Information Criteria)** とは，一般的な統計モデルを評価するための指標で，**AIC(Akaike's Infromation Criterion)** や **BIC(Bayesian I.C.)** などがそれにあたります。これは今検証しているモデルが，データによくフィットする「本当の」モデルからどの程度離れているかを示す指標です。本当のモデル，というのは分かり得ませんから，複数のモデル A,B,C を同じデータに当てはめたとき，AIC や BIC が相対的に小さいものが，より良く適合していると判断します。相対的な指標ですから，違う変数を扱うモデル間の比較はできないことに注意が必要です。

■RMSEA　モデルがデータとどの程度乖離しているか，を直接表現した指標が **RMSEA(Root Mean Square Error of Approximation)** です。近似した誤差に対する平方平均平方根，ということですね。一般に 0.05 より小さければ，そのモデルは当てはまりがよく，0.1 以上であれば当てはまりが悪いと判断します。

■CFI/TLI　**CFI(Comparatative Fit Index)** や **TLI(Tucker-Lewis Index)** は，観測変数間にまったく相関がないという非現実的なモデル (独立モデルという) に比べて，当該のモデルがどの程度よいものかを指標化したものです。いずれも 1.0 に近ければ近いほど良いモデルとされています。一般にこの数値が 0.9 以上になることを目指してモデルを改訂していきます。

■SRMR　**SRMR(Standardized Root Mean square Residual)** は標準化された残差平方平均平方根を表します。これはモデルで説明できなかったものがどれほどあったか，を表しますので，0 に近ければ近いほど当てはまりの良いことを表します。

■**修正指数**　最後に，**修正指数 (modification index)** を紹介しておきます。上で述べたように，検証しようとした統計モデルがどの程度当てはまっているか，ということを数値化できるようになったわけですが，常に最初にたてた仮説モデルが最適である，ということは滅多にありません。データから仮説の訂正を余儀なくされることがある，あるいはもっと良くなったり新しい発見があったりすることが，統計モデリングの醍醐味でもあります。修正指数は，どこをどう改良すれば指標がよくなりますよ，というヒントを与えてくれるものです。もっとも，適合度を上げることだけが目的なのではないことに注意してください。適合度を上げるために，意味のわからないモデルを作り上げたところで，モデルの意味がないのですから。

構造方程式モデリングは，非常に複雑な方程式を解いているんだな，ということは理解していただけたかと思います。次回はそんな複雑なモデルでも，R で簡単に分析できるよということを，演習を交えて理解していきましょう。

12.5 課題

2 項目 1 因子モデルの方程式，およびそれを行列形式で表したものを書きましょう。

また行列の要素で書いた場合，既知数と未知数はどちらが多くなるでしょうか。言い換えれば，このモデルは識別できるでしょうか。式を展開して確認してください。

第 13 章

R による構造方程式モデリング

これまでの流れと同じで，統計技術の理論を知っただけではなく，自分で実際に計算できる演習を経てこそ理解が深まります。本講では R をつかって実際に構造方程式モデリングを解くことを演習的に学んでいきましょう。構造方程式モデリングを実装するパッケージは複数あるが，最も応用範囲がひろい `lavaan` パッケージを用いることにします。授業を始めるにあたって，まずは `lavaan` パッケージをインストールしておいてください[*1]。

13.1 モデル式の入力

13.1.1 観測変数だけのモデル

まずは観測変数同士の関係をパスでつなぐモデルをみてみましょう。観測変数同士のパスですから，(重) 回帰分析をやるようなものと同じです。今回のデータとして `lavaan` パッケージに含まれている `HolzingerSwineford1939` データセットを使います。これは Holzinger & Swineford (1939) のデータで 301 人に対して行われた 15 の能力テストスコアの一部が入ったデータです。一部を 13.1 に示しました。変

表 13.1 Holzinger and Swineford(1939) のデータ

id	sex	ageyr	agemo	school	grade	x1	x2	x3	x4	x5	x6	x7	x8	x9
1	1	13	1	Pasteur	7	3.33	7.75	0.38	2.33	5.75	1.29	3.39	5.75	6.36
2	2	13	7	Pasteur	7	5.33	5.25	2.12	1.67	3.00	1.29	3.78	6.25	7.92
3	2	13	1	Pasteur	7	4.50	5.25	1.88	1.00	1.75	0.43	3.26	3.90	4.42
4	1	13	2	Pasteur	7	5.33	7.75	3.00	2.67	4.50	2.43	3.00	5.30	4.86
5	2	12	2	Pasteur	7	4.83	4.75	0.88	2.67	4.00	2.57	3.70	6.30	5.92
6	2	14	1	Pasteur	7	5.33	5.00	2.25	1.00	3.00	0.86	4.35	6.65	7.50

数の意味は以下の通りです。x1 から x3 が空間的知覚，x4 から x6 が言語的能力，x7 から x9 が移動する物体の認識力を測るものになっています。

id 被験者 ID
sex 性別 (男性=1, 女性=2)
ageyr 生まれ年

[*1] R のコンソールに `install.packages("lavaan")` と入力するか，RStudio のパッケージタブから入力しましょう。

agemo 生まれ月
school 所属校
grade 学年
x1 視覚的知覚
x2 立方体
x3 菱形
x4 段落の理解
x5 文章完成
x6 言葉の意味
x7 加速
x8 点を数える
x9 直線・曲線文字の区別

さて，ではまず回帰分析をやってみることにしましょう。変数 x4 を x5,x6 で回帰する重回帰モデルを lavaan を実行するには code:13.1 のように書きます。

code : 13.1 重回帰モデル

```
1  model1 <- "
2  x4␣~␣x5␣+␣x6
3  "
4  result1 <- sem(model1, data = HolzingerSwineford1939)
5  summary(result1, fit.measures = T)
6  # 比較
7  result1.1 <- lm(x4 ~ x5 + x6, data = HolzingerSwineford1939)
8  summary(result1.1)
```

■コード解説

1-3 **行目** モデルの記述
4 **行目** 関数による推定
5 **行目** 結果の出力
6-8 **行目** 従来の lm 関数による当てはめ例

ここでモデルを書くところは，ダブルクォーテーションで括られていることに注意してください。このようにすることで，R に複数行からなるモデルを渡すことが可能になります。実際のモデルは 2 行目だけで，これは lm 関数に書いているモデル式の形と同じであることがわかります。モデルを書いて，モデルとデータの組みを sem 関数に渡すと結果が帰ってくる，これだけです。結果を出力するときに，オプションとして fit.measures を書いてあるところだけが違います。

出力結果の一部を出力 13.1 に示します。実際はもっと色々でているのですが，CFI , TLI, AIC,BIC, RMSEA, SRMR などはそれぞれ適合度指標を荒らしています。推定結果は Regression: の Estimate のところを見てください。これらは，lm 関数と同じ係数になっていることが確認できます。

R の出力 13.1: 観測変数だけのモデル

```
    （中略）

    User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    1.000
  Tucker-Lewis Index (TLI)                       1.000

Loglikelihood and Information Criteria:

    （中略）

Akaike (AIC)                                 673.134
  Bayesian (BIC)                               684.255
  Sample-size adjusted Bayesian (BIC)          674.741

Root Mean Square Error of Approximation:

  RMSEA                                          0.000

  （中略）

Standardized Root Mean Square Residual:

  SRMR                                           0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  x4 ~
    x5                0.423    0.047    8.958    0.000
    x6                0.390    0.056    7.002    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x4                0.537    0.044   12.268    0.000
```

このように，重回帰モデルをするのであれば記述方法は同じです。違いはさまざまな観点からの適合度指標モデルということ。加えて，このモデルはどんどん拡張していくことができる点にあります。

たとえば**パス解析 (Path Analysis)** というのがあります。これは影響力のルートが $x \to y \to z$ のように，次から次へと繋がっていくモデルです。SEM ができるまでは，まず $x \to y$ を回帰分析し，次に $y \to z$ を分析する，という方法でした。しかしこれでは R^2 など適合度を逐一確認する必要があり，またモデル全体の評価ができないという欠点があります。しかし SEM ではコード code:13.2 のように書くだけです。

code：13.2 パス解析のモデル

```
1  model2 <- "
2  x4 ~ x5
3  x5 ~ x6
4  x6 ~ grade
5  "
6  result2 <- sem(model2, data = HolzingerSwineford1939)
7  summary(result2, fit.measures = T, standardized = T)
```

このモデル式 (2-6 行目) にあるように，$grade \rightarrow x6 \rightarrow x5 \rightarrow x4$ という一連の影響力を分析し，一気に適合度を表現してくれています。出力結果には適合度指標のほか，すべての変数を標準化した標準化係数を出すオプションを追加しました。

これを図にしたのが図 13.1 です。ちなみにこの図も R のパッケージで書きました。パス図を自動的に書いてくれるパッケージとして `semPlot` パッケージ Epskamp (2021) や `tidySEM` パッケージ Van Lissa (2019) などを使います。ここでは `tidySEM` パッケージの `graph_sem` 関数を使いました。

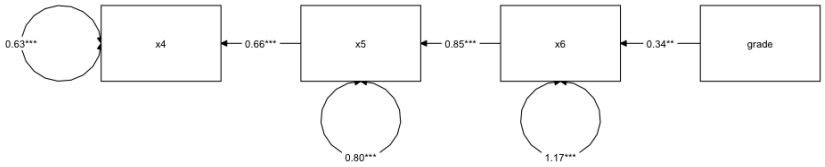

図 13.1 パス解析の図

13.1.2 測定方程式を入れたモデル

今度は測定方程式を入れたモデルを考えてみましょう。回帰モデルは従属変数と独立変数をチルダ (~) でつなぎましたが，潜在変数を作る場合は=~で右辺に観測変数，左辺に潜在変数をセットします。コード例を `code:13.3` に示します。

code：13.3 因子分析モデル

```
1  model3 <- "
2   visual =~ x1 + x2 + x3
3  textual =~ x4 + x5 + x6
4    speed =~ x7 + x8 + x9
5  "
6  result3 <- sem(model3, data = HolzingerSwineford1939)
7  summary(result3, fit.measures = T, standardized = T)
```

これのモデル図は図 13.2 のようになります[*2]。モデル上とくに指定はしてありませんが，潜在変数同士の相関係数も自動的に推定されています。パス係数は潜在変数からでてきていますから，**因子負荷量**のように考えることができますね。

このモデルは因子分析の一種ですが**探索的因子分析**とはいくつかの点で違います。1 つは因子数が 3 である，と最初から決めていた点。2 つ目は因子負荷量ですが，探索的因子分析の場合はどの項目に対しても共通因子からの因子負荷量が計算されていました。今回の例では，x4 から x9 の変数にすいても visual 因

[*2] この図は小野島 (2021) の関数を使っています。tikz を使って綺麗なモデルが描ける素晴らしい関数の提供に感謝。

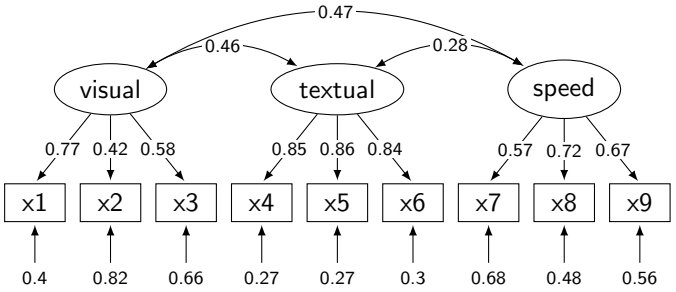

図 13.2　潜在変数モデル

子からのパスが出ている，というモデルだったのです。今回のモデルではそれらのパスはなく，visual 因子は x1,x2,x3 にしか影響していません。言い換えると，x4 から x9 に対する visual 因子のパス係数は 0 である，と特定したようなものです。

探索的因子分析の場合，理想的には「因子が関係する項目には十分影響しているが，関係しない項目にはまったく影響しない」という**単純構造の原理 (Principle of Simple Structure)** が理想とされていました。SEM を使うとこの理想的なモデルが合っているかどうかを検証する＝モデル上で制約をかけてその適合度を評価する，という使い方ができます。このような方針の因子分析のことを**検証的因子分析 (Confirmatory Factor Analysis)** と呼んで，探索的因子分析と区別します。

検証的因子分析をすることは，関係のない項目への影響を 0 とする，という強い仮定を置いていることにもなりますが，このことによって**因子的妥当性**や**収束的妥当性**，**弁別的妥当性**を検討している，と考えることもできます[*3]。もしこの理論的なモデルの当てはまりが悪ければ，異なる因子負荷のパターンを考えなければなりません。あるいは因子数を変えてモデルを組まなければならないかもしれないのです。因子数は同じだけど因子負荷量が違う，といったモデルを考えることもできますし，因子負荷量も固定して「この値に違いない」としたモデルを検証する，ということもできます。モデルを当てはめるデータは違っても構いません。むしろ違うデータに同じモデルが当てはめられ，十分な適合度があればそれはそのモデルの普遍性があるということで，いいことなのです。

このように，SEM では同じモデルを男性と女性，学生と社会人，日本人データと外国人データといった複数のデータに当てはめて，どこまで同じでどこから違うか，と言った比較をできます。こうしたモデル比較のことを**多母集団同時分析 (Multi-group analysis)** といいます。

13.1.3　構造方程式モデルへ

それでは潜在変数同士の関係に，更なる仮定を入れたモデルを考えてみましょう。

code：13.4　構造方程式モデル

```
1  model4 <- "
2  ␣visual␣=~␣x1␣+␣x2␣+␣x3
3  textual␣=~␣x4␣+␣x5␣+␣x6
4  ␣␣speed␣=~␣x7␣+␣x8␣+␣x9
5  ␣textual␣~␣visual␣+␣speed
```

[*3] 因子的妥当性とは，因子が十分に大きく項目に影響していることです。収束的妥当性とは，因子が関係しない項目に影響しないこと，弁別的妥当性は因子同士が別のものであると区別できることを表しています。

```
 6  ␣␣speed␣~␣grade
 7  "
 8  result4 <- sem(model4, data = HolzingerSwineford1939)
 9  summary(result4, fit.measures = T, standardized = T)
10  modificationindices(result4) %>%
11    as_tibble() %>%
12    arrange(-mi)
```

■コード解説

1-7 行目 モデルの記述。visual, textual, speed の 3 因子をつくり，textual は visual と speed から説明されるモデル。さらに speed 因子は学年によっても説明される。

8 行目 関数による推定

9 行目 結果の出力。適合度指標と標準化係数の出力オプションをつけて。

10-12 行目 分析結果の**修正指数**を出力。ただし指標 `mi` の大きい順に並べ替えるために 11 行目で出力を `tibble` 型にし，`arrange` 関数で並べ替えている。

まずは今回のモデルについても，図でみたほうがわかりやすいでしょう。結果を合わせて図 13.3 に示してみました。

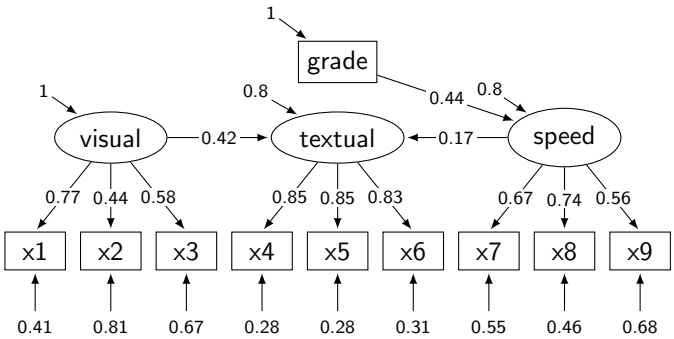

図 13.3 構造方程式モデルへ

この図の出力と，R での出力はどこが対応しているか，それぞれ確認しましょう。

まずは測定方程式のところです。因子負荷量，すなわち潜在変数からのパスはすべての変数を標準化したところの数字ですので，`Std.all` の列を参照します。この因子で説明できなかった独自成分の大きさは，`Varianvces` のところに表されています。

構造方程式として，textual 因子が visual 因子と speed 因子に説明されており，そのパス係数が `Regressions:` の `Std.all` 列に示されています。speed 因子はさらに grade 変数にも説明されているので，そのパス係数も確認できます。

SEM の基本として，パスが入った (影響を受けた) 変数は，影響が受けた部分と受けなかった部分に分割されます。影響を受けなかった部分が残りの分散として，`Variances:` のところに示されています。

visual 因子は説明変数になっていますが，パスが入ってきませんので，この分散は 1.00 です。図では visual 因子の楕円の上に 1 からの影響が入っていますが，これは分散が 1 だということの意味です。同じく

grade 変数も説明されない変数[*4]ですので, この分散が 1(標準化されています) になっているのです。

R の出力 13.2: 構造を入れたモデルの出力結果 (一部)

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                1.000                               0.899    0.770
    x2                0.573    0.109    5.249    0.000    0.515    0.437
    x3                0.724    0.124    5.846    0.000    0.651    0.576
  textual =~
    x4                1.000                               0.978    0.849
    x5                1.109    0.067   16.646    0.000    1.085    0.851
    x6                0.924    0.056   16.349    0.000    0.903    0.834
  speed =~
    x7                1.000                               0.727    0.667
    x8                1.022    0.130    7.837    0.000    0.744    0.737
    x9                0.782    0.108    7.273    0.000    0.569    0.564

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  textual ~
    visual            0.453    0.094    4.841    0.000    0.416    0.416
    speed             0.226    0.093    2.435    0.015    0.168    0.168
  speed ~
    grade             0.645    0.105    6.147    0.000    0.887    0.443

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.554    0.132    4.187    0.000    0.554    0.407
   .x2                1.121    0.103   10.841    0.000    1.121    0.809
   .x3                0.851    0.097    8.769    0.000    0.851    0.668
   .x4                0.370    0.048    7.706    0.000    0.370    0.279
   .x5                0.448    0.059    7.635    0.000    0.448    0.276
   .x6                0.358    0.043    8.268    0.000    0.358    0.305
   .x7                0.658    0.080    8.178    0.000    0.658    0.554
   .x8                0.466    0.072    6.478    0.000    0.466    0.457
   .x9                0.695    0.069   10.021    0.000    0.695    0.682
    visual            0.807    0.161    5.029    0.000    1.000    1.000
   .textual           0.764    0.096    7.918    0.000    0.798    0.798
   .speed             0.425    0.083    5.130    0.000    0.804    0.804
```

最後に出力 13.3 にある, **修正指数 (modification index)** の出力を見てみましょう。一行目にあるのは, visual 因子は変数 x9 へのパスをつけると, 適合度がぐっと上がるよ, ということを意味しています。

[*4] 説明されない変数のことをとくに**外生変数 (exogenous variable)** といいます。これに対して説明されることがある変数は**内生変数 (endogenous variable)** といいます。

R の出力 13.3: 修正指数 (一部)

```
# A tibble: 64 x 8
   lhs     op    rhs        mi   epc sepc.lv sepc.all sepc.nox
   <chr>   <chr> <chr>   <dbl> <dbl>   <dbl>    <dbl>    <dbl>
 1 visual  =~    x9       41.9 0.442   0.397    0.394    0.394
 2 visual  ~     speed    25.3 0.507   0.411    0.411    0.411
 3 visual  ~     textual  25.3 2.24    2.44     2.44     2.44
 4 textual =~    x1       16.6 0.490   0.479    0.411    0.411
 5 visual  ~~    speed    13.6 0.184   0.314    0.314    0.314
 6 speed   ~     visual   13.6 0.228   0.282    0.282    0.282
 7 visual  ~     grade    12.8 0.445   0.495    0.247    0.495
 8 grade   ~     visual   12.8 0.137   0.123    0.247    0.247
 9 speed   =~    x1       11.1 0.313   0.227    0.195    0.195
10 x1      ~~    x9       11.0 0.169   0.169    0.272    0.272
```

実際に visual 因子が変数 x9 に影響している (x9 から因子が構成されている) というモデルを作り, 適合度指標を比べてみましょう。表 13.2 に代表的な適合度指標の修正前の値と修正後の値を示しました。いずれの適合度指標においても, 大きな改善が見られています。さて, これをどう考えたらいいでしょうか。

表 13.2 修正前後での指標の変化

index	before	after
CFI	0.895	0.945
TLI	0.856	0.923
AIC	7479.335	7433.224
BIC	7557.115	7514.707
RMSEA	0.100	0.073
GFI	0.919	0.944
AGFI	0.866	0.905

13.2 実践上の注意点

ここまでで, さまざまなモデルを表現できること, それを統一的に評価できることが理解できたかと思います。とくに, 統一的に評価できることでさまざまな**モデル比較**も簡単になり, さらに適合度が良いモデルにするためにはどうすれば良いかについて, 指標も出てくることがわかりました。

分散共分散行列の隅々まで記述できる大きな力を手に入れたことは間違いないのですが, 大事なのは, 我々は構造方程式モデルを使う側であって, それに使われる側であってはならない, ということです。学会誌に掲載されるような論文で, 構造方程式モデルを見ると, その適合度は CFI,TLI,AGFI が 0.9 を超えているような, とても適合度の良いモデルがほとんどです。適合度が悪いモデルだと, これはデータとモデルがあっていないということですから, モデルの改善を求められたり掲載されなくなったりするということがあるでしょう。しかし大事なのは, モデルを使って主張したい心理学的内容のほうであって, 適合度が良いだけでの中身のないモデルではないはずなのです。

今回も機械的に visual 因子が x9 変数から構成されるようにすると, 適合度が上がるという提案を受けて

実施してみたところ，確かに適合度は上昇しました。ところでこの x9 変数というのは直線や曲線のスピードを認識するテストであって，(静的な) 空間認知能力とは違うものではないでしょうか？ そもそもそういうつもりで作ったものではないのに，機械に指摘されて「ひょっとしたらそういう側面があったのかも。そうだ，最初からそう思っていたに違いない」というように，自分を無理やり納得させてはいないでしょうか？

心理学の場合は変数の多くが関係しあっていますから，「移動する直線を見るというのは空間認知とも考えられるのだ。少なくともデータはそう示している」という理屈が成り立つかもしれません。しかし結果を見てから考え方を変えるのは，適切な研究実践法ではないでしょう。これらの問題は結果の再現性が担保されないという心理学の危機の引き金になりました[*5]。自分に都合の良い結果やモデルを出すことが目的ではなく，心がどのような状態になっているのかについての理論的積み重ねや発展こそ，目的であったことを忘れてはいけません。

とくに構造方程式まで使えるようになると，心理学的実体同士の関係を描写し，モデル化できるということが魅力的に映るかもしれません。心理学者は，これこそがやりたかったことなのかもしれないですね。しかしデータを超えての解釈はご法度ですし，何より構成された潜在変数が心理的な実在であるかどうかは，改めて考えなければならないのです。これらはあくまでも分散共分散行列から算出されるでしかなく，「文章読解力」「空間把握力」といった次元に貼り付けた自分たち自身の命名法に酔って，思考停止するようなことがあってはなりません。

この潜在変数同士の影響力が心理学的にどういう意味があるのか，をしっかり考えてから，モデルでの検証に進まなければならないことに注意して使ってください。

13.3 そのほかの統計パッケージ

構造方程式モデリングの利点の 1 つは，モデルを可視化したことにあります。皆さんもモデルの図を見た方が，出力結果を見るよりも理解が進んだ気がするでしょう。

こうした構造方程式モデリングを実行するソフトウェアは，R の専売特許ではありません。たとえば **Amos** という IBM 社が出しているソフトウェアでは，マウスをクリックしながら統計モデルを作り分析していくことができます。モデルの構成から GUI でできるのは大変な利点です。

また，構造方程式モデリングはさまざまな分析手法の統合的ツールです。言い換えるとかなり複雑なモデルであっても，ゴリゴリ計算し分析してくれます。現在考えられているさまざまなモデルの，ほぼすべてについて計算できるソフトウェアが **Mplus** です。このソフトウェアで分析できない SEM はない，と言っていいほどその用途は広く，また計算スピードも爆速です。カテゴリカルな変数にももちろん対応していますから，IRT/GRM のような出力もできます。

R の利点は商用ソフトと違ってフリーで手に入るところですが，有用・有償・商用パッケージでも良いのであれば，これらのソフトも活用することを考えてもいいでしょう。また R でも `lavaan` パッケージの他に，`sem` パッケージというのもあります。ツールは色々あって，ユーザがそれを選べるようにことが理想的ですから，皆さんも興味があれば色々試してみましょう！

13.4 課題

今日の授業でおこなったすべての次の計算をする R コードを提出してください。ファイル形式は R スクリプトか Rmd とします。なお提出されたコード単体でバグがなく動くことが確認できないものは，未提出扱い

[*5] さきほどの良い結果しか雑誌に掲載されない問題のことを，**出版バイアス (publication bias)** の問題といい，今も問題視されています。

3062 になります。コードの書き方などわからないところがあれば，曜日別 TA か小杉までメールで連絡し，指導を
3063 受けてください。

第 14 章

双対尺度法

さて前回 SEM を学んだことで，分散共分散行列をとことんまで分析し尽くす方法が手に入ったことになります。SEM は統合的な表現方法ですから，観測変数でも潜在変数でもいいですし，**順序尺度水準**以上の大小関係が表現できる数値データであれば，あらゆる表現ができるわけです。

ではこれで多変量解析はすべて理解したことになるのでしょうか。いえ，もちろんそうではありません。分散共分散行列の分解はできるようになりましたが，変数間関係の表現は分散共分散行列だけではありません。ここまで，データの持つ情報は分散がすべて，変数間関係は共分散で表されるから分散共分散行列がすべてだ，と言わんばかりに話をしてきましたが，その枠を外すとどうなるでしょうか。

14.1 直線的ではない関係

ご存知の通り，分散共分散行列を標準化した行列は相関行列といいますが，相関係数が 1.0(あるいは -1.0) の状態を考えれば明らかなように，分散共分散行列 (や相関行列) で表現されるのは変数の直線的な関係性に限った話です。心理学の場合は中庸が良いようなシーンも少なくありません。たとえば血圧と健康度の関係で言えば，低血圧でも高血圧でも不健康であり，ちょうどいい血圧が一番健康的だというのはすぐにわかります。このように中庸が良い場合は，散布図が U 字型に現れてきます。散布図が U 字型であれば，相関係数としては 0 近くになりますが，血圧と健康が無関係だとは誰も言えないでしょう。

ごくシンプルかつ具体的な例をあげてみましょう。血圧と頭痛の頻度について調査したとします。血圧は高い，普通，低いの 3 段階。頭痛の頻度は「ない」，「たまに」，「ときどき」，「いつも」の 4 段階です。表 14.1 のような結果を見ると，この集計表に直線的な関係は確かになさそうですね。でも関係ないわけではない，と思いませんか。

表 14.1　血圧と頭痛

血圧	ない	たまに	ときどき	いつも
高い	0	0	3	2
普通	5	0	0	0
低い	0	2	3	1

このような場合は，「血圧が高い人か低い人は，頭痛がある」という傾向が見て取れるはずです。しかしこのデータ，機械的に血圧が高いを 1，普通を 2，低いを 3 とし，同様に頭痛もない～いつもを 1,2,3,4 として相関係数を計算すると，$r = 0.165$ になります。相関関係では傾向をうまく読み取れていないのです[*1]。

[*1] 相関係数でお話ししましたが，分散共分散行列でも同じです。共分散で表現すると単位のせいで関係が直接的には理解できま

その根本的な原因は，もちろんスコアリングの方法にあります。共分散を計算するには間隔尺度水準以上の情報が必要で，今回のような 3,4 件法では相関係数を計算して良い数字ではありません。ではポリコリック相関係数のように順序尺度水準の相関係数なら良いか，といいたいところですが，それでもまだ不十分です。なぜなら，血圧が高い方から低い方まで，順番が保存されてしまっているからです。

関係を導き出すには抜本的な改革が必要です。たとえば表 14.2 のようにすればどうでしょうか。こうすると左下から右上にかけての直線的な関係がまだ見えてきます。これは血圧の順番を「高い」「低い」「普通」に並べ替えたものです。また，頭痛の順番も「いつも」と「ときどき」をひっくり返しました。順番を並び替えてしまっ

表 14.2　並び替えられた「血圧と頭痛」

血圧	ない	たまに	いつも	ときどき
高い	0	0	2	3
低い	0	2	1	3
普通	5	0	0	0

たというのは，尺度水準的には数字を無視して「高い」「低い」「普通」というカテゴリーとして扱ったということになります。つまり**名義尺度水準**レベルにまで落として考えたのです。

このように並べ替えて，血圧のスコアを高い $\rightarrow 3$，低い $\rightarrow 2$，普通 $\rightarrow 1$ とし，また頭痛の頻度をいつも $\rightarrow 3$，ときどき $\rightarrow 4$ と付け替えて相関係数を計算すると，今度は $r = 0.810$ になりました。こちらの方が線形性は高いことが数字でも確認できました。

ここで行ったのはこのように，すべての反応カテゴリをただの言葉だと考えて，付与された数字を無視して並べ替えたことになります。そうすることで，左下から右上にかけて線形性を高めることができました。しかし「高い」と「低い」，「低い」と「普通」の配置も等間隔である必要はなく，もっと直線性がはっきりするように配置してやってもよいのでは，というアイデアが浮かびます。

ここで考え方が反転したことに注意してください。一般的なリッカート尺度では，反応カテゴリに (シグマ法などで) 数字をつけて，変数間の線形性を算出して意味を考えるのでした。ここでは逆に，変数間の線形性を最大にするように反応カテゴリに数字をつけてやろう，という考え方です。データの直線性というのはデータの特徴を最も強調し解釈しやすい形です。そのようにデータを整えるためには，反応カテゴリにどういう数字を付与すれば良いか，と考えるのです。この方法を**数量化の理論 (Quantification Methods)** といいます。

反応カテゴリに数字を与えるとき，名義尺度水準にまで落として，すなわち数字の意味を無くして並べ替えるような作業をします。もちろん分析対象が，初めから名義尺度水準であっても構いません。たとえば出身県を調査したようなデータがあったとして，他の変数との線形関係を最大にするように並べ替え，数字を付与することもできます。数量化の考え方は，名義尺度水準の変数に解釈しやすい数値を与えることとも言えるのです。

せん。

14.2 林の数量化理論

数量化の研究をしたのは，日本の偉大な統計学者，林知己夫 (はやし ちきお)[*2]という人です。その名を冠して林の数量化理論と呼ばれることがあります。

林はさまざまな研究成果を挙げており，論文ごとにデータに必要な分析方法を開発するというようなスタイルでした。その膨大な研究業績を，弟子である飽戸弘[*3]が分類して，同じような分析方法ごとに I 類，II 類，III 類，IV 類，と呼んでいきました。

表 14.3 林の数量化

手法	外的基準	データ	目的	関連する手法
数量化Ｉ類	量的変数	質的変数	外的基準の予測	重回帰分析
数量化Ⅱ類	質的変数	質的変数	外的基準の判別	判別分析
数量化Ⅲ類	なし	質的変数	変数間の関係の要約と記述	正準相関分析
数量化Ⅳ類	なし	対象間の類似度	対象間の関係の要約と記述	多次元尺度法

表 14.3 にあるように，基本的には質的変数，すなわち名義尺度水準や**順序尺度水準**のデータが得られたときに，それを解釈するためにはどのような数値を割り振ってやれば良いか，という発想から生まれたものになっています。

さきほどの表 14.1 のようなデータの並べ替えについては，数量化でいうところの III 類に該当します。ただこの数量化 III 類はおもしろいもので，同時期に独立にこの手法がフランス，カナダでも発展しており，2 つの別名を持っています。フランス学派が開発した手法は**対応分析 (correspondence analysis)** といい，カナダ在住の日本人，西里静彦[*4]が開発した手法は**双対尺度法 (Dual Scaling)** といいます。どの分析も，基本的にはカテゴリカルな変数について直線性を最大にする値を割り振ることを目的にします。

カテゴリカル変数なので，分析の応用範囲は多岐にわたります。どのような変数でも名義尺度水準に落とすことができるからです。表 14.2 には一般的なデータセットの形を示しました。ID があって，Q1,Q2... と項目が列方向に並ぶ形です。一行が一人の反応を表しています。Q1 がたとえば性別などの名義尺度水準の変数であっても，男性 $\rightarrow$ 1，女性 $\rightarrow$ 2 のようにコード化するルールを決めて入力します。Q2 はたとえばリッカート法で当てはまる，やや当てはまる･･･などの 5 件法だったとしましょう。その場合はシグマ法によるスコアを入れる，あるいは簡便的に当てはまるを 5，やや当てはまるを 4，といったように数字を割り振って入力しますね。

これをカテゴリカル変数と見なしてデータセットにした例が表 14.2 です。ID は分析対象ではありませんから横に置くとして，Q1 が 2 列に増えています。ID=1 の人は男性なのですが，この人は Q1:Male=1 かつ Q1:Female=0 というようにコード化されています。同様に，Q2 には「どちらとも言えない」を選択しているのですが，これを 3 とするのではなく 00100 と 5 列にわたってコード化しているのです。このようにすることで，

[*2] 林 知己夫 (はやし ちきお，1918 年 6 月 7 日 - 2002 年 8 月 6 日) は，日本の統計学者。正四位勲二等，理学博士。統計数理研究所第 7 代所長。社会調査･世論調査におけるサンプリング方法の確立を始め，数量化理論 (Hayashi's Quantification Methods) の開発とその応用で知られる。1990 年代以降，データの科学 (Data Science) を提唱し，その研究･思想は現在へと引き継がれている。Wikipedia より。

[*3] 飽戸 弘 (あくと ひろし，1935 年 3 月 14 日 -) は，日本の社会学者，東京大学名誉教授。専門は，社会心理学，コミュニケーション論。Wikipedia より。

[*4] 西里 静彦 (にしさと しずひこ，1935 年 6 月 9 日 -) は，カナダの行動計量学者 (計量心理学)。学位は Ph.D.(ノースカロライナ大学･ 1966 年)。トロント大学名誉教授，アメリカ統計学会フェロー，日本行動計量学会名誉会員。北海道十勝郡浦幌町出身 (札幌市生まれ)。Wikipedia より。

すべての変数をカテゴリーとして扱うことができるようになります。

表 14.4 一般的なデータセット

ID	Q1	Q2	···
1	1	3	···
2	2	4	···
⋮	⋮	⋮	

表 14.5 カテゴリカル化したデータセット

ID	Q1:M	Q1:F	Q2:5	Q2:4	Q2:3	···
1	1	0	0	0	1	···
2	0	1	0	1	0	···
⋮	⋮	⋮				

また表 14.1 を並べ替えた時のように，こうした名義尺度のデータの線形性が最大になるように，行だけでなく，列も並べ替えます。データの中で線形性が最大になるように，反応カテゴリと回答者に数字を割り振るのです。ちなみに表 14.1 は集計されたデータであり，今回の表 14.2 は集計前のデータです。カテゴリカルな変数の分析の場合は，どちらでも良いのです。行と列の関係が表されている数字であれば，「ある/ない」のような二値反応でも，集計された度数でも構いません。さらにいえば，行と列の関係が記述されていればなんでもいいのです。

これまでの回帰分析，因子分析から SEM に至るまでの流れは，変数間関係だけを考えてきました。N 行 M 列のデータ行列の計算の途中で，行に関する情報は平均化して潰されてしまい，最終的には $M \times M$ サイズの正方行列だけを扱うことになったのでした。そして M 個の変数に**因子負荷量**などの重みをつけて考察してきました。今回のカテゴリカルなデータは，$N \times M$ 行の矩形行列をそのまま分析し，行と列の両方に重みをつけます。正方行列でも矩形行列でも，変数と変数，変数と回答者がどのように関係しているかというところを見るという意味では同じです。SEM では分散共分散行列や相関行列が，双対尺度法ではクロス集計表や素データがその分析対象になります。数学的な面から説明すると，正方行列の場合は**固有値分解**をして，**固有値**と**固有ベクトル**を求め，固有ベクトルが変数の重みになるのでした。矩形行列の場合は**特異値分解 (Singular value decomposition)** と呼ばれ，行および列に対応する**特異ベクトル**をその重み，座標と考えることになります。本質的には同じようなものだと思っていただければと思います。

14.3 双対尺度法による分析

それでは実際の分析例を見てみましょう。次のコード code:14.1 は，MASS パッケージに含まれるサンプルデータ caith を対応分析で分析するものです。

code : 14.1 双対尺度法による分析

```
1  library(MASS)
2  caith
3  result <- corresp(caith,nf=min(nrow(caith),ncol(caith)-1))
4  result
5  plot(result)
```

■コード解説

1 行目 パッケージ MASS のよみこみ

2 行目 サンプルデータ caith を表示させる

3 行目 対応分析関数 corresp で分析した結果を result オブジェクトに入れる

4 行目 結果の出力

5 行目 結果のプロット

データ caith はスコットランドの Caithness 地方に住んでいた人に対してなされた調査で，髪の毛の色と目の色の関係を集計したものです。列方向に髪の毛の色，行方向に目の色が入っていますね (表 14.6)。

表 14.6 データ caith の中身

	fair	red	medium	dark	black
blue	326	38	241	110	3
light	688	116	584	188	4
medium	343	84	909	412	26
dark	98	48	403	681	85

この行・列のカテゴリにはとくに順序性などありませんが，分析することで最も線形性の高いウェイトをつけることができます。もちろん 1 次元で表現できない可能性があり，その場合は第二，第三と次元数を増やして行きます。データの行数あるいは列数の小さい方マイナス 1 次元まで求めることができます。それを表現しているのが，関数 corresp のオプション nf で，行数 nrow(caith)，列数 ncol(caith) の小さい方 (min 関数) マイナス 1 を指定しています。

結果は出力 14.1 のようになります。行および列にスコアがついていますね。この値を尺度値として使うと，データの相関係数が最大になるというような指標化がなされたのです。

R の出力 14.1: 対応分析の結果

```
First canonical correlation(s): 4.463684e-01 1.734554e-01 2.931691e-02 1.134031e-
16

 Row scores:
              [,1]       [,2]       [,3] [,4]
blue   -0.89679252  0.9536227  2.1884132    1
light  -0.98731818  0.5100045 -1.0837859    1
medium  0.07530627 -1.4124778  0.1894089    1
dark    1.57434710  0.7720361 -0.1482208    1

 Column scores:
              [,1]       [,2]       [,3]       [,4]
fair   -1.21871379  1.0022432  0.4271282 -0.8692696
red    -0.52257500  0.2783364 -4.0268545 -1.3400421
medium -0.09414671 -1.2009094  0.1103959 -0.8453208
dark    1.31888486  0.5992920  0.3450676 -1.2251588
black   2.45176017  1.6513565 -1.5736976  1.1609621
```

結果はプロットされたものを見た方がわかりやすいかもしれません。図 14.1 にプロットを示しました。たとえば横軸，dim1 にそって目の色を見ていくと，light – blue – medium – dark の順に並べた方が良い，ということがわかります。とくに light と blue は近いのであまり大きな意味的違いはないことがわかります。同様に髪の毛の色は，fair – red –medium –dark –black の順に並べられることになります。具体的な数字はそれぞれ出力の一列目の通りです。この dim1 だけでは表現できない違いが，dim2 で表されており，それが図では上下の広がりとして示されていますね。

この並びが，新しく構成された次元だと考えることができます。そしてこれを見ると，たとえば髪の毛と目の

色がどちらも dark である場合，この二点は近くにプロットされています。この二点が近くにあるということは，両者の結びつきが強いということです。髪の毛が暗い人は目の色も暗い人が多い，といった関係の強さが見て取れます。

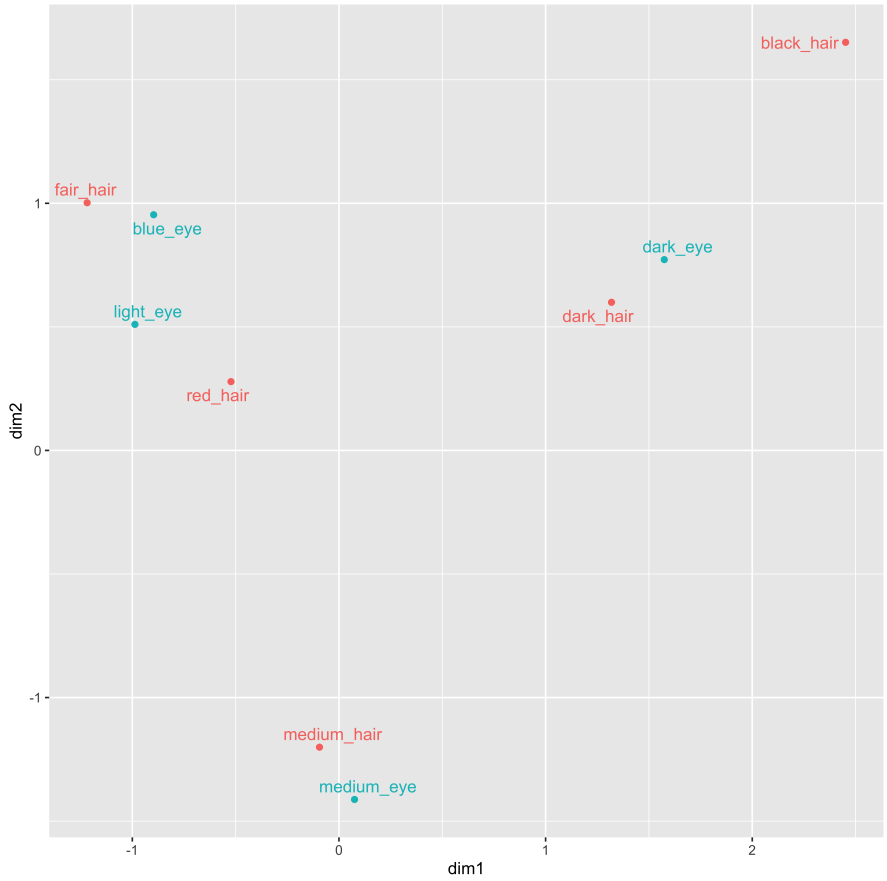

図 14.1 分析結果のプロット

ちなみにこれは対応分析独特の表示法で，行と列の変数が 1 つの画面にうつされていますね。厳密には，行ベクトルが作る次元，列ベクトルが作る次元は別物です。ですから，列の要素を行の空間に写像するためには変換が必要で，逆もまた然り，なのです。双対尺度法の場合はこの違いを厳格に捉えます。対応分析の場合は，相互に写像しあった座標を一枚の図にすることで情報を圧縮しています。数学的には同じモデルであっても，表現の仕方や表示の仕方に，モデル作成者の考え方が反映されているとも言えるでしょう。

14.4 テキストマイニングへの応用

さて，本講で紹介した分析方法は，名義尺度水準を対象にしていて，かつ外的な基準がなく内的な構造を明らかにしようとするものだということがわかってきました。いわば名義尺度水準における因子分析ですね。名義尺度水準を対象にしたということは，およそ言語化できたものはすべて分析の対象にできる，とも言えます。最も低次元で一対一対応した言葉の世界の数字だからです。

応用例として，**テキストマイニング (Text Mining)** を紹介しておきましょう。別名自然言語処理とも言

われますが，これはテキスト，すなわち普段の「言葉」に潜む関係を分析する手法です[*5]。文章を対象にしますから，小説や新聞記事などはもちろん，日記や逐語録なども分析の対象にしてしまおうというものです。心理学の分野では面接法などにおいて，クライアントがどのように語るかをすべて記録することがありますが，これをみて「うーん，こういうことを考えているんだな」と読み取るのは主観的な判断によるものです。ここにテキストマイニングを用いれば，機械的にどう言った言葉の使われ方をしているかを分析できます。あるいはツイッターやフェイスブックなどの SNS でどのような言葉，記事が流行しているかと言ったことを分析する，というような使い方もできます。分析対象が一気に広がりますね。

このテキストマイニングがやっていることは二段回に分かれ，第一段階が**形態素解析 (morphological analysis)**，第二段回が多変量解析です。第一段階は，「今日はいい天気ですね」といった平文を「きょう」「は」「いい」「てんき」「です」「ね」といった要素に分解することを指します。英語のような分かち書きがされる言語であればこれは簡単なのですが，日本語の場合は分かち書きされていないことに加え，漢字，ひらがな，カタカナなど表記方法もさまざまですから，この分析をするための特別な解析エンジンが必要です。幸い，すでに開発されているものがフリーソフトウェアとして利用できます。時間がかかりますがこれらを使うと，品詞ごとに分解し，その原形 (活用する前の形) や活用形は何か，と言ったことを一覧にしてくれます。

自然言語ですので大量の分割がなされますが，それを言葉同士の関係を表す行列の形で表現します。使われている品詞の原形ごとに誰が何回発言したかとか，各要素が 1 回の文章の中で同日に使われた回数をカウントするなどして，言葉と人，言葉と言葉の関係をデータ化するのです。データ化できればあとはこっちのものです。数量化できるのですから，その言葉群のなかでどの言葉とどの言葉が近いのか，他のどの変数と関係するのか，と言った分析をできます。

テキストマイニングについては R でもできますし，専門的なソフトウェアがあります。詳しくは樋口 (2020) を参照してください。

14.5 課題

今日の授業でおこなったすべての次の計算をする R コードを提出してください。ファイル形式は R スクリプトか Rmd とします。なお提出されたコード単体でバグがなく動くことが確認できないものは，未提出扱いになります。コードの書き方などわからないところがあれば，曜日別 TA か小杉までメールで連絡し，指導を受けてください。

[*5] マイニングとは鉱脈を掘る，という意味です。

第 15 章

多次元尺度構成法

今回は**多次元尺度構成法 (Multi-Dimensional Sacling; MDS)** について解説します。ここまで分散共分散行列を分解するところから，クロス表のような**名義尺度水準 (の)** データを分解するところまでやってきました。

多次元尺度構成法は，分散共分散行列を扱う線形モデルよりはやや仮定が緩く，また数量化のように解析者が数字を与えることを目的にするというよりは，その名の通り尺度を作ろうとしているという意味で心理学的・心理測定的なモデルだといえるでしょう。

私たちが単位もない不確かなものを対象にしながらもそれを測定するモノサシをつくることができるのは，2 つの対象にたいしてその比較をして一方が他方よりも大である，ということが言えるからでしょう。記号を使って言えば，x と y を比べて $x \succeq y$ である，ということから，尺度上の値 $p \geq q$ を対応させるということが，尺度を作るということです[*1]。このとき比較する 2 つが重さや広さのような物理的なものであればわかりやすいですし，「痛み」や「喜び」といった心理的な要素であっても構いません。あるいは「より賛成」といった態度表明のようなものでも良いかもしれません。この比較から出てくる関係は，分散共分散や相関係数のように強い線形の過程を置かなくても，より緩やかに**距離 (distance)** という考え方で表現されます[*2]。

MDS は距離行列を固有値分解するモデルです。それではこの方法について内容を見ていきましょう。

15.1 多次元尺度構成法

R にはサンプルデータとして `eurodist` というヨーロッパ各地の都市間距離のデータが用意されています。表 15.1 にその一部を示します。

元のデータは 21×21 のサイズです。表から，たとえばアテネ (Athens) とバルセロナ (Barcelona) の距離が 3313km，と言うことが読み取れます。表の右上が空白になっていますが，これは**距離行列**が**対称行列**なので，ムダな情報を表示しないようにしているからです。アテネとバルセロナの距離は，バルセロナとアテネの距離に等しいですからね。

さて距離行列はこのように，**正方行列**ですから，**固有値分解**をすることで**基底**を求めることができます。基底は座標を形成する基本単位ですから，それを使って各変数の位置にあたる座標を計算できます[*3]。このように距離行列から座標を求めて，変数をプロットすることで変数間関係を可視化する手法のことを**多次元尺**

[*1] 経済学では学問の最初の段階で，財を源とした集合に対する二項関係 $x \circ y$ を定義し，それを効用関数 u で実数領域に写像して $u(x) \geq u(y)$ を考える，といった基本的な原理を抑えます。心理学ではなぜか，「集合」「元」「二項関係」という基本的な比較の要素について考えることなく，その分析ツールだけがどんどんと発展してきています。

[*2] 後述しますが，相関係数も距離の一種と考えられます。

[*3] 厳密には距離行列 $\boldsymbol{D}$ そのものではなく，それを二重中心化した行列の固有値分解になりますが，本質的には変わりません。詳しくは岡太・今泉 (1994) などを参照のこと。

表 15.1 ヨーロッパ都市間距離データの一部

	Athens	Barcelona	Brussels	Calais	Cherbourg	Cologne
Athens	0					
Barcelona	3313	0				
Brussels	2963	1318	0			
Calais	3175	1326	204	0		
Cherbourg	3339	1294	583	460	0	
Cologne	2762	1498	206	409	785	0

度構成法 (Multi-Dimensional Scaling: MDS) と言います。

試しに `eurodist` データを MDS で 2 次元プロットしてみましょう。これを実行するコードは簡単で，`eurodist` データのようにデフォルトで組み込まれている `cmdscale` 関数を使います。

code : 15.1 計量 MDS の実践と描画

```
1  library(ggrepel)
2  # MDS を 実 行
3  result.MDS1 <- cmdscale(eurodist,k=3)
4  # y軸 反 転 さ せ つ つ 描 画
5  g <- result.MDS1 %>%
6    as.data.frame() %>%
7    dplyr::mutate(label = rownames(.)) %>%
8    ggplot(aes(x = V1, y = V2, label = label)) +
9    geom_point() +
10   geom_text_repel() +
11   xlim(-2500, 2500) +
12   ylim(2500, -2500) +
13   xlab("dim␣1") +
14   ylab("dim2")
```

■コード解説

1 行目 `ggplot2` でラベルをプロットするときに，綺麗な配置にしてくれる `ggrepel` パッケージを使います。このコードを実行するときに持っていない人は，インストールしておいてください。

3 行目 `cmdscale` 関数に `eurodist` データを与えています。`k=3` は 3 次元解を出すように指定しています。地球は球体ですから，地球上の地理は 3 次元で表現できますよね。

5-14 行目 `ggplot2` による描画です。結果オブジェクトである `result.MDS` をデータフレームにし，行の名前になっていた都市名を変数として格納したのち，散布図のようにプロットしています。

図 15.1 をみると，上の方にストックホルム (Stockholm) があって，右下にアテネが，中央にパリ (Paris) が･･･といった配置になっています。地球上の位置と完全に一致しているとは言えませんが，それでも概ねうまくプロットできていますね[*4]。このように，距離関係だけから地図を作ることができるというのが，MDS という手法なのです。

[*4] 完全に一致しない理由は，地球が球面であるのに対し平面にプロットしたから，と言うのもあります。

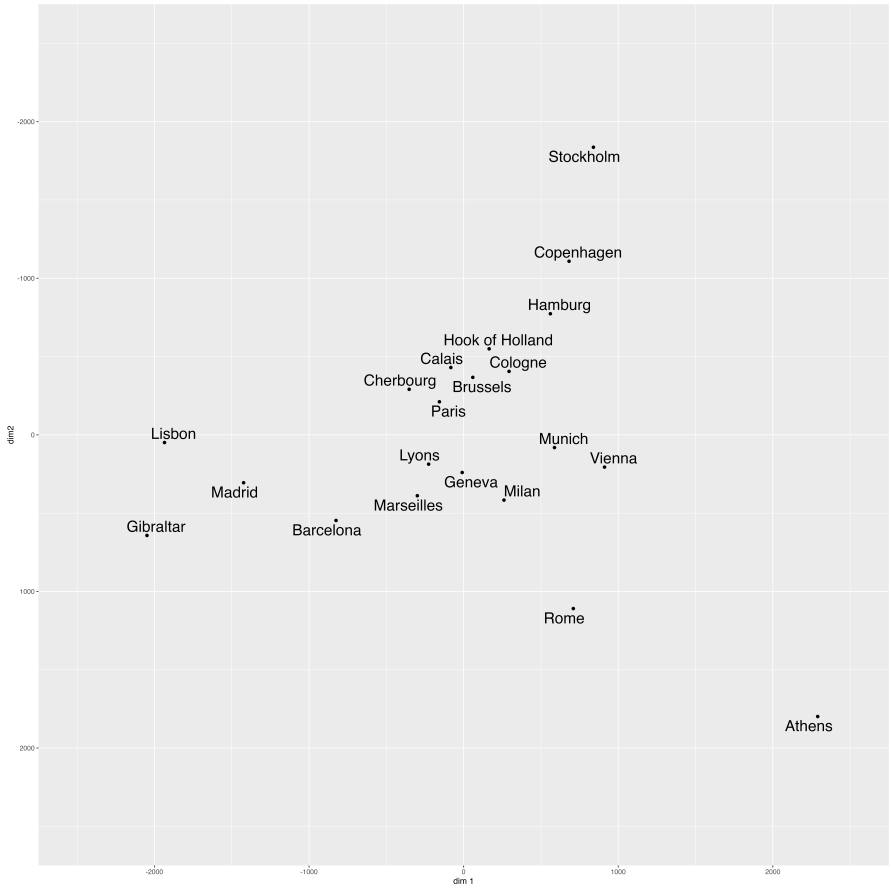

図 15.1 MDS のプロット

15.2 距離と心理学のデータ

MDS は距離関係だけから地図を作る方法でした。因子分析は相関関係から次元を作る方法でした。この 2 つの手法はとても似ています。というのも，相関関係と言うのが変数と変数の近さ・遠さ，つまり距離を表しているとも言えるからです。あらためて**距離 (distance)** とは何かを考えてみましょう。2 点 x, y の距離を $d(x, y)$ とすると，距離とよばれる数字の条件は次のようになります。

非負性 $d(x, y) \geq 0$

非退化性 $d(x, y) = 0 \Leftrightarrow x = y$

対称性 $d(x, y) = d(y, x)$

三角不等式 $d(x, z) + d(z, y) \geq d(x, y)$

もっとも一般的に使われるのはユークリッド距離で，2 次元座標 $(x_1, y_1), (x_2, y_2)$ があった時のユークリッド距離は次のように計算します。

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

3 次元座標 $(x_1, y_1, z_1), (x_2, y_2, z_2)$ の場合は項を増やすだけです。

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

このようにして計算される距離ですが，上の条件を考えると何も二乗してルートを取らなくても絶対値を足し合わせるような方法でも構いません。たとえば 2 次元座標の場合は，次のような計算でもいいのです。

$$|x_1 - x_2| + |y_1 - y_2|$$

現にこのような距離のことを**マンハッタン距離 (Manhattan distance)** といいます。二乗ではなく n 乗して n 乗根をとる，という形で一般化することもできます。

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - x_j|^p \right)^{\frac{1}{p}}$$

これは**ミンコフスキー距離 (Minkowski distance)** という名前がついています。他にもマキシマム距離，バイナリ距離，チェビシェフの距離，キャンベラ距離などがあり，いずれも R の `dist` 関数のオプションで選ぶことができますので，ヘルプなどを参照してみてください。

また，相関係数 r_{ij} は $-1 \leq r_{ij} \leq +1$ の範囲にありますが，この絶対値から $1 - |r_{ij}|$ とするとこれも距離の条件に当てはまります。どの程度類似しているかというのも，距離と考えることができるのです。

ここまでは数学的な距離のバリエーションでしたが，次に心理学的な意味に目を向けてみましょう。距離とは類似度でもありますから，何を距離と見なすかによって，さまざまな心理学的刺激がデータを形作ることになります。

たとえば評定尺度で，n 個の項目で何らかの評価をしてもらったとします。x_{ij} を i 番目の項目における対象 j の評定値だとすると，$d(j,k) = \sqrt{\sum_{i=1}^{N}(x_{ij} - x_{ik})^2}$ とすれば対象の類似度が計算できます。ほかにも何らかの刺激 A,B について，A か B かの判断をさせた時に混同してしまった混同率や，単語と単語の連想価，刺激の汎化勾配，反応潜時，ソシオメトリックなデータなど，いろいろなものが「類似しているかどうか」の指標として使えます[*5]。尺度評定よりも，具体的な 2 つの刺激が似ているかどうかの反応の方がやりやすい，というのは誰しも実感としてわかることではないかと思います。

類似度のデータが得られれば，それを距離と見做して MDS にかければ，変数間の関係を地図に描くことができるわけです。このように応用可能な領域が非常に広いことも，MDS の利点であると言えるでしょう。

15.3 非計量多次元尺度法

さて心理学的なさまざまな刺激が，MDS によって可視化できるということがわかってきました。距離行列が構成できれば，後の分析は何とでもできるわけです。ただし，この場合の距離行列とは，間隔尺度水準以上であることが必要です。しかし心理学的な刺激に対する反応をデータ化する時は，同時に被験者はそこまで鋭敏に反応しているのか，いいかえれば本当に間隔尺度水準以上の精度で判断できているのかな，という人間側の問題が気になりますね。そこまで人間は鋭敏無反応をしていないかもしれません。

でも大丈夫。MDS は仮定を緩めたモデルがあります。**非計量的多次元尺度構成法 (Non-Metric Multi-Dimensional Scaling)** と呼ばれる手法がそれです。非計量 MDS では，対象 j と k の類似度を δ_{jk} とし，分析によって埋め込む多次元空間での距離を d_{jk} とすると，

$$\delta_{jk} > \delta_{lm} \text{ならば} d_{jk} \leq d_{lm}$$

のように，イコールではなく順序関係だけ保持して座標を求めます。この手法では，元データが順序尺度水準程度の情報しか持っていなくても地図を描くことができるのです。人間の判断はせいぜいが順序尺度ぐら

[*5] これらデータの例に関しては高根 (1980) の Pp.14–27 を参照してください。

いですから,「より類似している $(\delta_{jk} > \delta_{lm})$」のであれば「より近くにある $(d_{jk} \leq d_{lm})$」というぐらいの配置の方が良いかもしれません。

表 15.1 に示したような物理的距離であれば，間隔尺度水準の情報であることは間違いありません。その場合には，最初に示した固有値分解による手法を使います。これは非計量 MDS に対して，**計量的多次元尺度構成法 (Metric Multi-Dimensional Scaling)** と呼ばれています。

15.3.1 非計量多次元尺度法の例

心理学ではデータの性質上，非計量 MDS のほうが便利なことが多いでしょう。計量 MDS でも非計量 MDS でも，R では簡単な関数で実行できます。ここでは `MASS` パッケージに含まれる `isoMDS` 関数を使って実践してみます。

使うデータは `M1score2021.csv` とします[*6]。ファイル名からお察しいただけるように，M-1 グランプリの評定をデータ化したものです[*7]。2021 年度のスコアは表 15.2 でした。

表 15.2 M-1 グランプリ 2021 の採点結果

演者	巨人	富澤	塙	志らく	礼二	松本	上沼
モグライダー	91	93	92	89	90	89	93
ランジャタイ	87	91	90	96	89	87	88
ゆにばーす	89	92	91	91	93	88	94
ハライチ	88	90	89	90	89	92	98
真空ジェシカ	90	89	92	94	94	90	89
オズワルド	94	95	95	96	96	96	93
ロングコートダディ	89	90	93	95	95	91	96
錦鯉	92	94	94	90	96	94	95
インディアンス	92	91	93	94	94	93	98
もも	91	90	91	96	95	92	90

M-1 の採点は審査員各自の主観に基づいて行われ，得点の絶対値はそれほど重要ではないかもしれません。すなわち，松本人志の 80 点が上沼恵美子の 80 点と同じぐらいの面白さを評価しているか，ということについては真偽判断ができないでしょう。それでも各審査員の中での相対的評価には，一貫性がありそうです。すなわちある審査員が漫才師 A に 80 点，漫才師 B に 85 点をつけたのなら B の方が面白かったということでしょうし，漫才師 C が 83 点なら $A < C < B$ という順序はあると思われます。つまり**順序尺度水準**程度の質はあると仮定することに無理はなさそうです。

そこでこのデータをもとに，10 組の漫才師の類似度を計算します。類似度はユークリッド距離を用いることにします。ユークリッド距離は既に述べたように差分の二乗を総和して平方根を取ったものですが，具体的な数字で見た方がわかりやすいかもしれませんので，表 15.3 を用意しました。表 15.3 にモグライダーとランジャタイの二組だけ取り出し，これで計算例を見てみます。それぞれの得点の差分，その二乗を計算し，それ

[*6] ファイルはシラバスのサイトからダウンロードできます。

[*7] 念のために解説しておきますが，M-1 グランプリとは 2001 年から始まった漫才の賞レースの 1 つで，年末に年間チャンピオンが決定します。開催年ごとにルールが少し変わることもありますが，基本的には予選を勝ち抜いた 10 組の漫才師が 4 分間のネタを披露し，6-7 名の審査員が 100 点満点で採点します。点数の上位 3 組が決勝戦を行い 2 本目のネタを披露，投票によりチャンピオンが選出されるという流れです。2021 年はオール巨人，富澤たけし，塙宣之，立川志らく，中川礼二，松本人志，上沼恵美子が審査員で，最終的には錦鯉がチャンピオンになりました。このデータセットは 1 本目のネタについての採点を 2001 年から集めたものになります。

表 15.3　距離の計算

	巨人	富澤	塙	志らく	礼二	松本	上沼	総和
モグライダー	91	93	92	89	90	89	93	637
ランジャタイ	87	91	90	96	89	87	88	628
差分	4	2	2	-7	1	2	5	9
差分の二乗	16	4	4	49	1	4	25	103

を総和したところ 103 という値になっています。これの平方根を取ったもの，すなわち $\sqrt{103} = 10.14889$ がこの二組の距離，すなわち非類似度ということになります。この計算を全ての組み合わせについて計算してくれるのが，dist 関数なのです。

code : 15.2　距離行列の計算

```
 1  dat <- read_csv("M1score2021.csv")
 2  dat.mat <- dat %>%
 3    dplyr::filter(年代 == 21) %>%
 4    arrange(ネタ順) %>%
 5    dplyr::select(-年代, -ネタ順) %>%
 6    pivot_longer(-演者) %>%
 7    na.omit() %>%
 8    pivot_wider(id_cols = 演者,
 9                names_from = name,
10              values_from = value) %>%
11    as.matrix()
12  rownames(dat.mat) <- dat.mat[, 1]
13  dat.mat <- dat.mat[, -1] %>%
14    dist()
```

■コード解説

1 行目　データファイルを読み込み，dat オブジェクトに格納します。

2-9 行目　必要なデータだけに絞り込む操作です。流れを解説しますが，他のやり方でもいいですしできあがったものが何かだけわかれば結構です。

3 行目　2021 年のデータだけに絞り込みます。

4 行目　ネタ順に並び替えています。

5 行目　年代変数とネタ順変数はもういらないので削除してしまっています。

6 行目　ロング型に変換しています。これで演者-審査員-採点のデータセットができます。

7 行目　欠測値を除外しています。実はこれがこの操作の目的で，というのも過去の審査データも入っているものですから，過去の審査員も大量に変数として含まれていて，それらが欠測値になってしまっていたのです。

8-10 行目　元のワイド型に戻しています。

11 行目　以下の行列処理のため，data.frame 型から matrix 型に変換しています。

12 行目　変数として一列目に演者名が入っていますが，これを行列の行名に入れています。matrix 型は行名・列名をデータの外に持つのです。

13-14 行目　変数としての演者名を除いて，パイプで dist 関数に入れ，距離行列を作っています。

できた距離行列は，表 15.4 のようになっています。モグライダーとランジャタイの距離が，先ほどの例で計算した値と一致していることを確認してください。またこれは**正方対称行列**ですから下三角だけ表示しておいます。また，対角が 0 になっています。自分自身との距離はゼロだからです。

表 15.4 演者の非類似度行列 (演者名は略記)

	モグ	ラン	ゆに	ハラ	真空	オズ	ロン	錦鯉	インデ	もも
モグ	0.000									
ラン	10.149	0.000								
ゆに	4.583	9.165	0.000							
ハラ	7.937	12.806	7.616	0.000						
真空	8.660	7.483	7.071	11.832	0.000					
オズ	12.490	15.652	12.207	14.933	11.000	0.000				
ロン	9.381	11.446	6.403	9.110	7.416	9.487	0.000			
錦鯉	8.485	15.264	8.307	10.909	10.247	7.071	7.874	0.000		
インデ	9.381	14.107	8.062	8.660	9.950	8.124	4.472	6.325	0.000	
もも	10.100	9.110	8.307	12.207	3.606	8.718	6.782	9.592	8.718	0.000

あとはこの距離行列を `isoMDS` 関数に渡すだけです。`isoMDS` 関数は引数として，何次元の解を求めるかを設定できます。地理データであれば 2,3 次元から作られていることは明らかですが，この評価が何次元かは事前にわかりません。何次元にするかの指標として，Kruskal (1964b) は `Stress` と呼ばれる値を次のように定義しました。

$$Stress = \sqrt{\frac{\sum (d_{ij} - \delta_{ij}^2)}{\sum d_{ij}^2}}$$

つまり実際のデータの距離 d_{ij} と，MDS で作られる空間上の座標から計算される距離 δ_{ij} の全体的な距離を当て嵌まりの指標と考えていることになります。これを目安に，次元数を 1 から 7 まで変化させながら，Stress 値がどうなるかを表したのが図 15.2 になります[*8]。

まるで因子分析の**スクリープロット**のようですね。横軸に次元数，縦軸に Stress 値を置いた折れ線グラフですが，当然のことながら反映させる MDS 空間の次元数が増えるとデータとの距離が縮んでいきます。Kruskal (1964a) の基準によれば，Stress 値は表 15.5 のように評価できます。この基準でいくと，今回は 2 次元で 4.9%(0.0049534) ですから，2 次元解で OK としましょう。

演者をプロットしたのが図 15.3 です。東西南北というか，上下左右の軸に特に意味はありませんから，因子分析のように因子軸の解釈や命名をすることはありません。近くにプロットされた対象は評価が似ていたんだな，ということがわかりますし，相対的な位置関係から，「ハライチともも は芸風が真逆だな」とか「ロングコートダディは中心に近いから中庸的な笑い，悪く言えばキャラが立ってないんだな」といったことが読み取れます。この図 15.1 や図 15.3 のように MDS で作られた座標のことを特に**布置 (configure)** といいます。特に非計量 MDS は優劣・大小関係という順序尺度水準の評定だけからでも，その空間的な特徴を描くことができますから，心理尺度のもつ仮定や測定モデルを考える必要がないため，今後その重要性が再発見されていくのではないかと思います。

[*8] どうして 7 までか，というと評定者が 7 名だからです。7 名それぞれの評価次元があると考えると，この距離空間の中には最大 7 次元あるはずですから。あるいは 10 組の演者がいますから，組み合わせの自由度から考えて 9 次元まで試してもいいと思います。

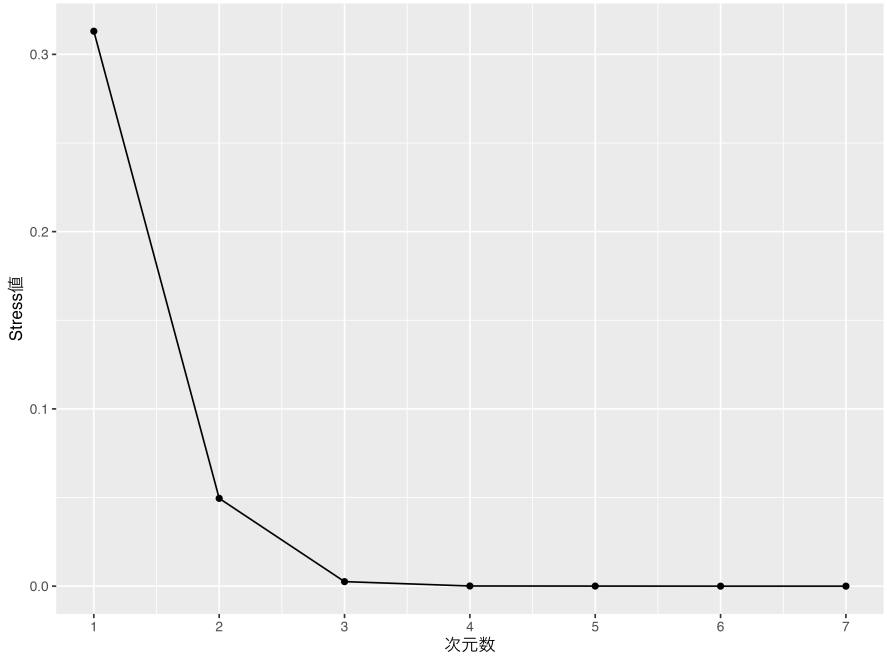

図 15.2 Stress 値の減衰

表 15.5 Stress 値の評価

Stress	Goodness of Fit
20%	poor
10%	fair
5%	good
$2\frac{1}{2}$%	excellent
0%	“perfect”

15.4 多次元尺度法の展開

多次元尺度構成法で作られた地図は，対象をプロットした地図です。地図には，その上に何か書き込んだり，地形の図に天気図を重ねるように複数の地図を重ねて表現したりできます。多次元尺度構成法にも，このような応用モデルがいろいろ考えられています。ここでいくつかの発展的な MDS モデルを見てみましょう。

■prefmap 類似度空間の上に，個々人の理想点を追加する方法です。個人の好み preference をマッピングする方法なので，**Preference Mapping()** と呼ばれています。個人 i が対象 j について，好みの程度を s_{ij} と評価したとします。対象 $1, 2, ...j..., M$ は別途類似度評定によって，MDS のつくった地図上にプロットされているとします。ここで個人 i と対象 j との地図上の距離 d_{ij}^2 に対して，次の回帰式を考えます。

$$s_{ij} = a_i d_{ij}^2 + b_i + e_{ij}$$

つまり，個人 i と対象 j の距離を使って，好みの程度 s_{ij} を予測するモデルを作り，誤差がもっとも小さくなる点に個人 i をプロットするのです。こうすることで，対象とそれを評価する人を一枚の地図に表現すると言

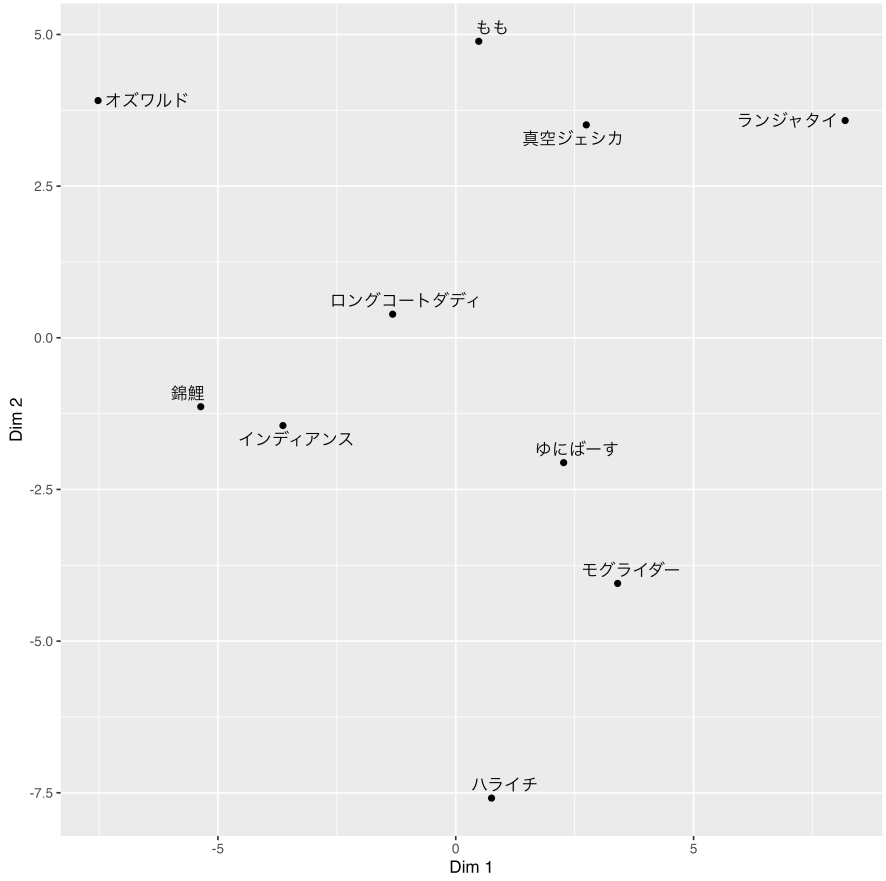

図 15.3 非計量 MDS のプロット

うことができます。

例えば先ほどの M-1 の例ですが，私の採点では表 15.6 のようになりました[*9]。この評定値を使って著者

表 15.6 著者の評定

演者	採点
モグライダー	90
ランジャタイ	60
ゆにばーす	85
ハライチ	92
真空ジェシカ	83
オズワルド	89
ロングコートダディ	85
錦鯉	83
インディアンス	82
もも	88

[*9] ランジャタイは何が面白いのかわからなかった。モグライダーはもっと評価されるべき。ハライチも良かったですね，ちょっと時間オーバーしたっぽいけど。

の理想点を書き足したのが図 15.4 です。低く評価したランジャタイからは遠く，高く評価したハライチやオズワルドに近いところに著者の理想的な笑いの点があり，錦鯉やインディアンスも近くにありますから，今回の優勝にはまぁ納得，と言ったことがわかります。皆さんも自分の理想の点を書き加えてみませんか?

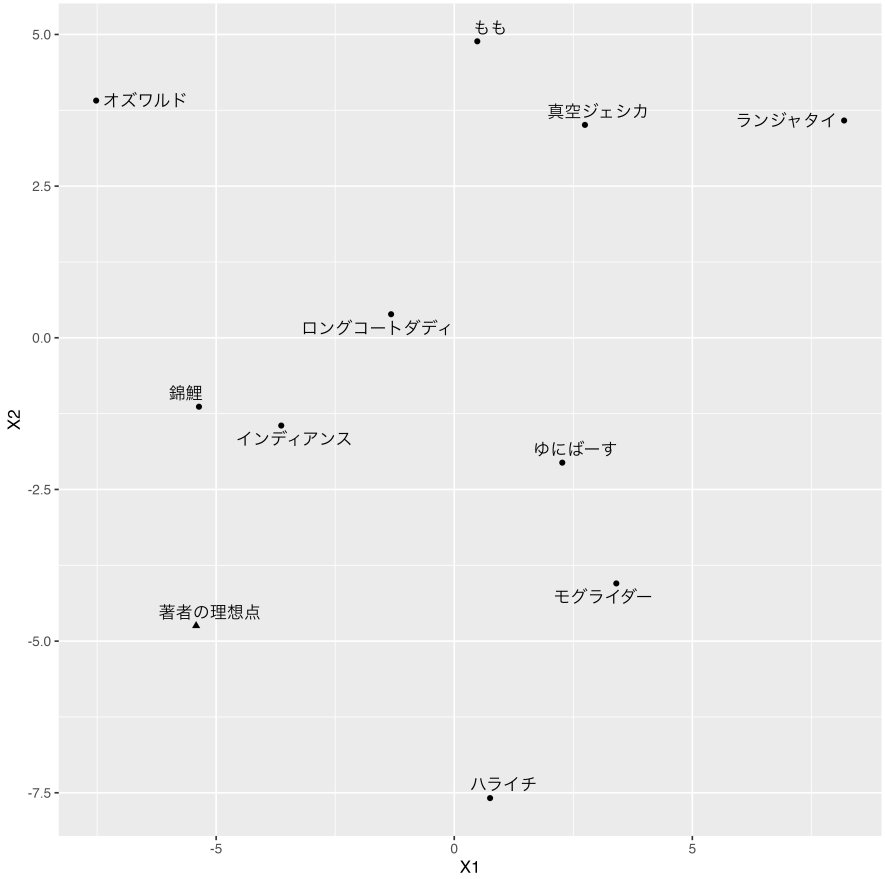

図 15.4　著者の理想点プロット

■Abelson Map　これは Abelson (1954) の考えた手法で，これも prefmap と同じく布置される対象に別の力 (選好度でもなんでもよい) があると考え，地図空間上に力の場をプロットして等高線を引くことで表現するモデルです。

この方法では，各点 P にかかる力 $V(P)$ を次のように定式化します。

$$V(p) = \sum_{j=1}^{M} \frac{V(j)}{1 + d_{pj}^2}$$

ここで j は各点，ここで言えば漫才師のことで，$V(j)$ が漫才師 j に与えられた評価点です。d_{pj}^2 は任意の点 p と対象 j の距離ですから，任意の点 p にかかる力は各漫才師が発する笑い力ですね。

この方法で，先ほどの M-1 プロットに，著者の評価を Abelson Map の方法で描き加えてみましょう。図 15.5 の右にある，ランジャタイの周りにたくさんの線が引かれていますが，いわばこれは低気圧のようにここのパワーが弱い (と著者は思っている) ところになります。実は，真空ジェシカの左側に通っているラインが平均点のラインですから，ここを境に著者は「面白い」と「面白くない」を区分しているとも言えます。ハライチや

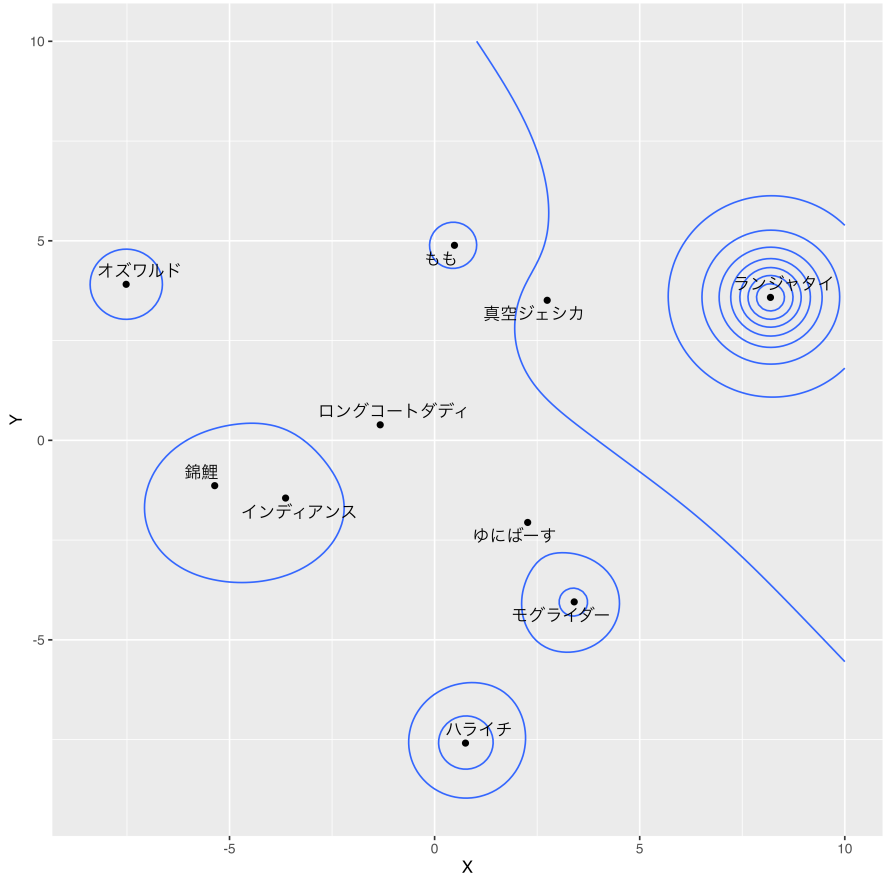

図 15.5 Abelson Mapping によるお笑い力の等高線図

オズワルドの周りにある等高線は山の高さ，パワーの強さを表しているラインですね。

力のメタファーにすぎないと言われたらそうですが，このように色々なものを可視化できるのも MDS の面白いところです。

■非対称多次元尺度構成法 距離データは対称でなければならない，というものの，たとえば好きな人に振り向いてもらえない＝片想いとか，都市間の人口の流入・流出，国際貿易の赤字・黒字などを考えると，対称間の関係が対称出ないことは少なくないわけです。そうした非対称な関係を，図に加えようと言うのが非対称 MDS です。非対称情報を対象部分 + 歪対象部分に分割し，この対称でない要素を図の中に書き加えるモデルが一般的で，プロットする対象の周りに縁や楕円で表現するとか，矢印で表現するとか，vonMises 分布と呼ばれる**確率分布**で表現するとか，さきほどの Abelson Map で高さとして表現するなどのモデルがあります。より数学的なモデルとして，プロットする空間を複素空間にするモデルもあります。詳しくは千野他 (2012) を参考にしてください。

■多次元展開法 リッカート法は心理尺度でもっともよく使われる方法で，「非常に当てはまる」から「まったく当てはまらない」までの段階的な反応を被験者に求める方法です。これは**段階反応モデル**で分析されることからも明らかなように，項目に対する反応段階が順番的に強くなっていくことを仮定しています。しかし，実際に被験者が丸をつけるときは，「自分の感覚にもっとも近いカテゴリーを選ぶ」ということをしているはずです。つまり，被験者と反応カテゴリーの距離が，尺度に反映されているはずなのです。この考え方から，

尺度に対する反応を距離とし，項目とそれに回答した被験者の両方をプロットする地図を書く方法があります。Coombs が考えた**多次元展開法 (Multi-dimensional unfolding method)** と呼ばれる手法がそれで，この手法から態度測定の尺度化を考えることもできます。発想としては数量化理論に近く，また清水 (2018) はこれを確率モデルに展開しています。

ここであげたいくつかの例にあるように，多次元尺度法もさまざまな角度から人の判断や反応から，**尺度 (scale)** を作る方法です。多次元尺度構成法は因子分析モデルよりも制約や仮定が少なく，より直接的に心理的反応をモデル化しようとしています。

本講で学んだように，心理学ではさまざまな角度から「数値化するルール」を考えてきていますが，それぞれ仮定や目的，何を良い尺度と考えるかという観点がことなります。我々心理学者は，統計的なツールのユーザに過ぎません。データは分析できればそれでいい，分析は機械がやってくれるから深く考えなくていい，という割り切り方もあるかもしれませんが，「そもそも何をデータとするか」「どのように数値化するか」という点については，そのような態度は取れないはずです。なぜならそもそも「心はいかにして数値化できるか」ということが明らかでないならば，その後の分析はすべて嘘っぽい数字を使った統計ごっこに過ぎないからです。**心理学者であるために，われわれは何をどのように数値化しているかについて，しっかりと理解する必要があるのです。**

15.5　課題

計量 MDS，非計量 MDS をそれぞれ一例ずつ，実践してみてください。データはどのようなものでも構いません。提出ファイル形式は R スクリプトか Rmd とします。なお提出されたコード単体でバグがなく動くことが確認できないものは，未提出扱いになります。コードの書き方などわからないところがあれば，曜日別 TA か小杉までメールで連絡し，指導を受けてください。

付録 A

よくある質問とミスの例

A.1 Frequently Miss and Comments

Rmd でレポートを提出したのに，なんだか中身の問題じゃないのに突き返された，中身を見てくれよ！と思う人もいるかもしれません。テキストでは R や Rmd での課題を提出するよう求めているところがありますが，その際よく見られる学生さんのミスとその対応についてのコメントをここにまとめておきます。

A.1.1 FMC1；そもそもファイルの書式が違う

Rmd で提出してください，R で提出してください，という指示に対して，違うものが提出されてくることがあります。書式があっていない，というのは些細なことのように思えるかもしれませんが，学術論文は書式に捉われず内容に集中するためにも，書式は整えられたものである必要があります。学会誌に掲載されている論文も，みなさんが書く卒業論文も，レポートに至るまで，書式や指示に沿ったものを準備する必要があります。書式があっていない場合は，門前払いになっても文句が言えないのがアカデミックの世界です。

ということですので，R や Rmd で提出してください，という指示があれば，R や Rmd で提出してください。ファイルの種類は拡張子で分類され[*1]，R ファイルは `.R`，Rmd ファイルは `.Rmd` という拡張子になっています。たまに `.Rproj` というファイルを提出してくる人がいますが，これは R プロジェクトのファイルで，これには R スクリプトも文書も含まれておらず，みなさんの計算環境情報が少し書いてあるだけです。また，`.Rmd` だけ入っていれば良いかというとそうではありません。まれに `Filename.Rmd.R` というようなファイルを送ってくる人がいます。これはファイル名にピリオドが含まれているだけで，ファイルの種類を識別する拡張子は `.R` ですから Rmd ファイルではありません[*2]。そもそもファイル名には 2 バイト文字はもちろん，!や?などの記号を含めるべきではありません。ピリオドも当然記号の一種ですから，ファイル名にするのは不適切です[*3]。

ではどうやって適切なファイル形式にするか，ということですが，最も素直な方法としては RStudio で新しくファイルを開くときに，R markdown 形式 (略して Rmd) を選ぶようにしましょう。もし間違って，R script 形式 (`.R` 形式) で開いてしまった，というときは，そのファイルを破棄して新しく作り直すのが一番ですが，エディタペインの右下にあるファイル種類表示 (図 A.1) をクリックして修正することもできます。

[*1] 拡張子についてはセクション C.5,Pp.186 参照

[*2] 最近の OS は拡張子を表示しない設定になっているものも少なくないので，このようなミスが生じます。

[*3] 長いファイル名などの場合，空白を入れるのも適切ではありません。できなくはないのですが，やるべきではないのです。どうしても空白を入れたければ，アンダースコアなどで区切ると良いでしょう。

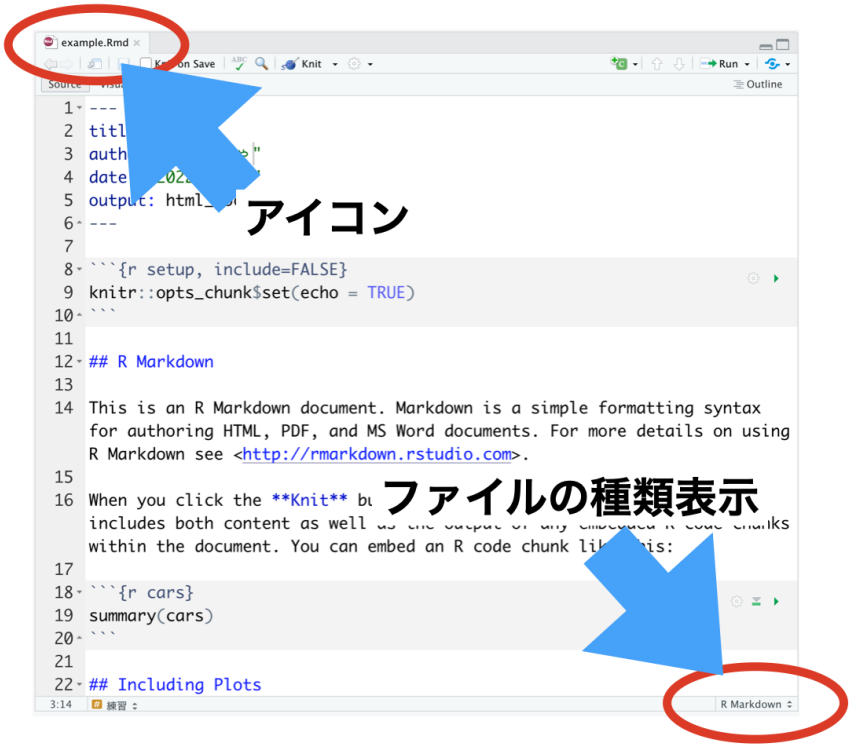

図 A.1 ファイルの種類を判別する方法

A.1.2 FMC2；やっぱりファイルの形式が違う

Rmd 形式のファイルは，拡張子だけで決まるものではありません。図 A.2 にあるように，Rmd ファイルは冒頭の 6 行 (正確には---で囲まれた領域) が YMAL と呼ばれるところで，文書全体の設定をしています。その下に，R のコードを実行する部分 (**チャンク (chunk)**) や文章の領域があります。文章のところは#記号で見出しを作ったりできます。

この YMAL 部分が壊れている，あるいはチャンクが正しく記述されていない場合，拡張子が.Rmd であっても適切な Rmd ファイルにはなっていません。YMAL 部分の書き方はよくわからない，という人も多いと思いますので，RStudio で Rmd ファイルを作ったときの状態をなるべく変更しないように注意すると良いでしょう[*4]。

チャンクは R のコードを書くところで，バッククォーテーション 3 つでくくるのが決まりです。チャンク領域が始まるところに{r}とかいて「ここが R で計算するところですよ」というのを指定するわけです[*5]。このバッククォーテーション 3 つ (```) が全角だったり (｀)，ダブルクォーテーションだったり (") すると，機械は正しくチャンクであると認識しません。フォントなどの見せかけ上は，微妙な違いのように見えますが，機械にとって違う文字列は違う意味を持ちますので注意してください[*6]。RStudio で編集する場合，チャンクの領域はや

[*4] Rmd ファイルを新しく開くときに，文書タイトルや著者名，日付，出力ファイル形式などを設定することができるウィンドウが開きますので，そこに必要な情報を書くと自動的にそれを使った YMAL が生成されます。また，本文として幾つかのサンプルコードが最初から含まれていますが，これらに関してはサンプルをそのまま使うことはないので，全て削除してしまって結構です。

[*5] ほかにも Python や Julia など他の計算言語を混ぜることもできます。

[*6] 記号の名称や入力については，セクション E,Pp.197 を参照してください。

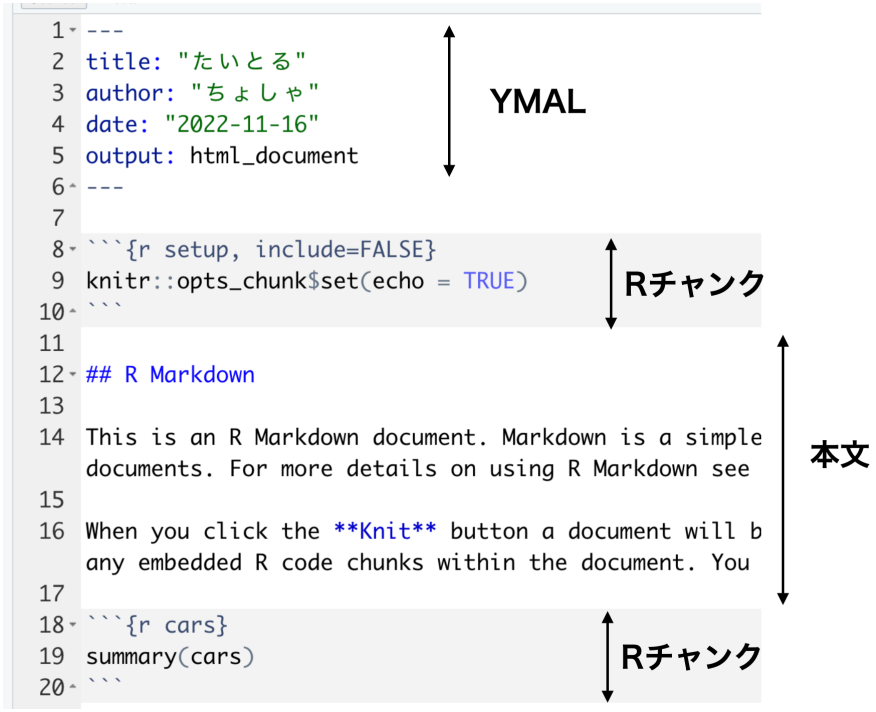

図 A.2　Rmd ファイルの中身における書式

や灰色がかった強調表示がされますので，どこにチャンクがあるかわかると思います。もし強調表示されていないようであれば，チャンクとして認識されていない可能性を疑った方が良いでしょう。

チャンクはバッククォーテーション 3 つで開き，おわったら同じくバッククォーテーション 3 つで領域を閉じます。閉じなければずっと R の計算領域が続くと解釈されますし，最後までチャンクが閉じられないと Rmd ファイルとして正しくない書式ということになります。チャンクが正しく入力できているかについて，常に注意を払っておく必要があります。また，自分でバッククォーテーション 3 つを使ってチャンク領域を開いたり閉じたりするのが面倒だ，という人は RStudio のチャンク挿入ボタンから挿入すると間違いないでしょう。

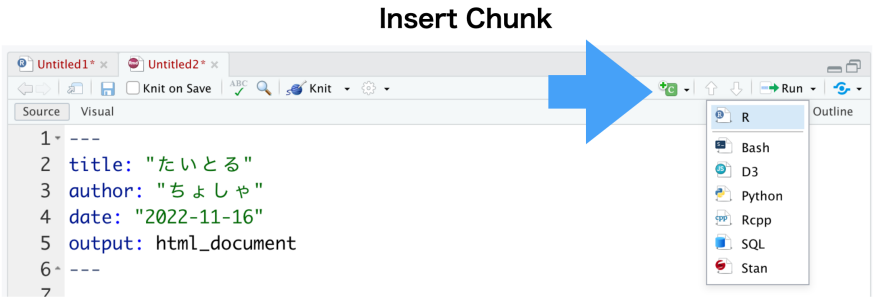

図 A.3　RStudio のチャンク挿入ボタン

A.1.3　FMC3；Rmd が knit できない

Rmd は文書作成に加えて，R での計算がセットになったファイル形式であり，文中の数字や分析結果は「書き写す」のではなく「その場で生成する」ものです。生成する，というのは Rmd ファイルを変換して，

HTML や DOCX，PDF 形式のファイルにすることを指します[*7]。このファイル変換を**ニット (knit)** といいます。編み物を編むようなイメージですね。このときに，タイトルをつけ，見出しのサイズを変え，R の計算をして結果を埋め込む作業をするわけです (図 A.4)。こうすることで，結果のコピペを避けること，再現性を担保することができるようになるわけです。すでに説明したように，チャンクを使って計算に必要な指示を Rmd

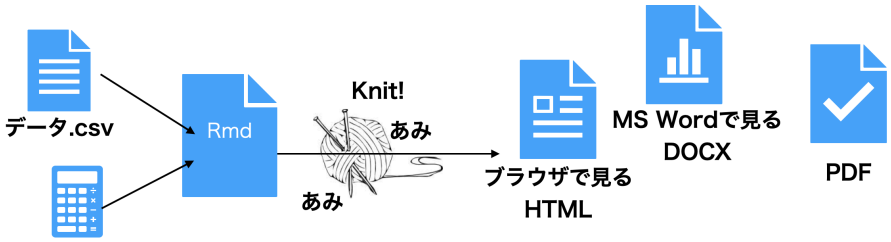

図 A.4 knit してレポートが完成する

ファイルに書きます。この指示はエラーのないコードである必要があります。当然ですが，スペルミスや間違った R コードは「実行できない」というエラーになるわけです。その場合，knit は中断され結果のファイルが出力されません。提出されたレポートは knit して出来上がったもののことを指しますから，knit できない Rmd ファイルを提出されてもレポートが提出されたことにはならないわけです。

knit できない Rmd ファイルになっていないか，というのはご自身の RStudio 環境で knit して確認できることです。提出前に一度，きちんと機能する Rmd ファイルになっているかどうか確認するようにしてください。以下で述べるように，自分の環境で knit できても，提出先の (私の) 環境で knit できないということもあり得ます。しかし，自分の環境で knit できないのに提出先でできる，ということはあり得ませんので，「knit できませんよ」というコメントをつけて返される前に自分で確認するようにしてください。

A.1.4 FMC4；外部環境を参照してしまう

さて，R で行う作業の中には，csv ファイルを読み込むといった「外部のファイルを使う」指示もありえます。例えば次のようなコードです。

code：A.1 Rmd 中の R コードが外部環境を参照してしまう

```
1  dat <- read.csv("social_effects.csv")
2  dat <- read.table("clipboard")
```

このコードでは，上の行は `social_effects.csv` というファイルを，下の行はクリップボードの内容を読み込むようになっています。ファイルを読み込む指示は，ファイルがなければ当然エラーになりますから注意が必要です。レポート等で Rmd ファイルを提出するとき，Rmd ファイルの他に必要な読み込むべきファイルがあれば一緒に提出するようにしてください。また，クリップボードの内容は再現できません。クリップボードとは，コピーアンドペーストのコピーを行ったとき，PC の内部で一時的にコピー内容を覚えておく場所のことです。つまり，みなさんがみなさんの環境で，クリップボード上にデータを一旦保持している場合は，このコードでエラーが生じることはありません。しかし，レポート提出先の (私の) 環境で，事前にデータのコピー作業を行なっているわけではないのですから，提出先でエラーが発生します。

[*7] HTML は Hyper Text Markup Language の略で，ブラウザで開くファイル形式です。DOCX は Microsoft Word のファイル形式です。PDF は Portable Document Format の略で，データを紙に印刷した状態のようにサイズを固定して出力したファイル形式であり，OS がもつビューワーや Adobe Acrobat Reader などで見ることができます。

そもそも Rmd ファイルは，作業の再現性を担保するためのファイルになっているわけですから，クリップボードの利用のような「自分の環境だけで可能な記録されない作業」をそのファイルに含めるのは適切ではありません。Rmd ファイルは Rmd ファイルだけで分析作業が完結するように，必要な記録は全て記載されている必要があります。同様に，分析作業に関係のない冗長な指示や無駄な指示は Rmd ファイルに含めるべきではありません。

A.1.5 FMC5；提出先の環境を変更してしまう

R はさまざまなパッケージを使って分析環境を拡張することができます。パッケージは **CRAN** を通じてインターネット経由で配布されますから，ネット環境があれば誰でも最新のパッケージをとってくることができます。みなさんが Rmd ファイルの中で使う R の関数の中には，パッケージの関数も含まれているでしょう。パッケージがなければ関数が動きませんから，パッケージをインストールする作業も Rmd に書いておきたいと思うかもしれません。しかし，これは推奨できません。

パッケージのインストールは，実行環境の準備にあたる作業です。Rmd ファイルを使ってコードのやり取りをするとき，提出先の環境で分析することになりますが，提出先の環境にどのようなパッケージをいつ入れるかは，提出先の環境の管理人が判断すべき問題です。提出する Rmd ファイルにパッケージをインストールする関数，すなわち `install.packages()` 関数が含まれているというのは，相手の環境を勝手に操作してしまうことと同じであり，セキュリティ的にも適切な発想ではありません。

環境の準備は提出先の管理人が管理すべき問題であり，またすでにパッケージが入っている場合は無駄な上書き作業をさせることにもなります。また `install.package` 関数は CRAN のサーバを参照したりしますから，適切な設定がなければ R チャンク実行時にエラーが発生します。いずれにせよ，R チャンクの中に `install.package` 関数を含めないようにしましょう。

以上が Rmd ファイルや R ファイルでレポートを提出するときの留意点です。その他にも R や RStudio を使うときによくある質問がありますので，それらも一問一答型で紹介しておきましょう。

A.2 Frequently Asked Questions；よくある質問と答え

Q. A.2.1: テキストを参考にパッケージをインストールしようとしたところ，エラーが発生しました。

A. 質問ではなくて報告ですね。お返事としては，「わかりました。エラーが発生して大変ですね。」としか言いようがありません。どの環境で，何をしようとして，どのようなエラーが出たのか，明示してください。メールのやり取りで指示が明確になるように，テキストを準備しています。何章，何ページの，どの文章を参考にしたのかも教えてください。

Q. A.2.2: 添付のようなエラーが発生しました (図 A.5)。対応策を教えてください。

A. エラーの発生画面を送ってこられていますが, この画面に写っていない上の方でエラーが発生していますから, これではどのエラーなのかわかりません。スクリーンショットを送ってこられることはよくありますが, ほとんどの場合, 適切な箇所が写っていません。複数枚添付してこられる人もいますが, 画面を拡大しながら読むのも難しいので, **R コードそのものを送ってきてください**。そうすると, 何行目のどこにエラーがあるかが明確になり, 対応に関係ない無駄なやり取り (ex.「もう少し上を写してください」「違う, もっと上」) が減ります。

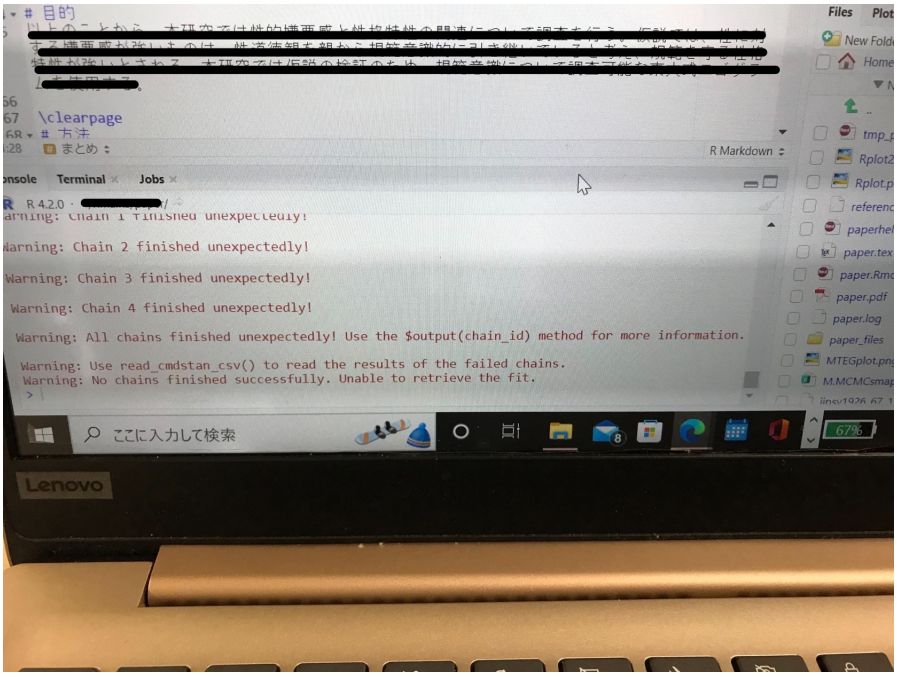

図 A.5 添付されてくるスクリーンショットの一例

Q. A.2.3: ファイルを読み込もうとすると次のようなエラーが出ます。どうすれば良いですか。

`'xxxxx'` に不正なマルチバイト文字があります

A. 文字化けの原因は, たとえば UTF-8 形式で提供されているファイルを, Windows 標準の CP932 形式で読み込もうとする, という文字コードの不一致です。ですからそれをオプションで指定してあげれば問題解決です。`read.csv` 関数を使っている場合, オプションの指定は `read.csv("foo.csv",fileEncoding="UTF-8")` のようにします。`tidyverse` の `read_csv` 関数を使っている場合, locale オプションで, locale 関数の encoding を明示的に UTF-8 とします。`read_csv("foo.csv", locale=locale(encoding = "UTF-8"))` のようにします。

Q. A.2.4: ファイルを読み込もうとすると次のようなエラーが出ます。どうすれば良いですか。

`'xxxxx'` に不正なマルチバイト文字があります

A. 文字化けの原因は，たとえば UTF-8 形式で提供されているファイルを，Windows 標準の CP932 形式で読み込もうとする，という文字コードの不一致です。ですからそれをオプションで指定してあげれば問題解決です。`read.csv` 関数を使っている場合，オプションの指定は `read.csv("foo.csv",fileEncoding="UTF-8")` のようにします。`tidyverse` の `read_csv` 関数を使っている場合，locale オプションで，locale 関数の encoding を明示的に UTF-8 とします。`read_csv("foo.csv", locale=locale(encoding = "UTF-8"))` のようにします。

Q. A.2.5: パッケージを読み込もうとすると次のようなエラーが出ます。どうすれば良いですか。

`library(tidyverse)` でエラー；`'tidyverse'` というパッケージはありません。

A. パッケージがないので，インストールしてください。

Q. A.2.6: パッケージをインストールしようとすると次のようなエラーが出ます。どうすれば良いですか。

```
Warning in install.packages:
    'lib = "C:/Program Files/R/R-4.0.5/library"' is not writable
```

A. ユーザがファイルに書き込みをする権限を持っていないので，(パッケージファイルをドライブに) 書き込みできないというエラーです。RStudio を実行する時に，「管理者として実行」を選びましょう。

Q. A.2.7: パッケージをインストールしようとすると次のようなエラーが出ます。どうすれば良いですか。

```
Warning in install.packages:
    ディレクトリ 'C:〜(任意の文字列)〜\????' を作成できません。理由は
```
```'Invalid argument' です。```

A. ユーザ名が全角文字を含んでいるため，文字化けして R 側から操作ができません。解決する方法は二つあります。

ユーザを作り直す方法 ユーザ名に全角文字を含まない，新しいユーザを作成します。設定＞アカウントから新しいアカウントを作りましょう。

インストールフォルダを指定する R のコンソールで `.libPaths()` と実行すると，パッケージをインストールするフォルダが出てきますが，ここを変更する方法です。次の 2 ステップで対応できます。

- まず C ドライブのすぐ下にインストール先のフォルダを作ります。`myLib` としましょう。
- 次に R のコンソールで `.libPaths("C:/myLib")` 書いて実行します。

これでインストール先が変わりますので，書き込み・インストールができるようになります。老婆心ながら付け加えますと，フォルダ名はなんでも構いませんが，全角文字ではいけません。また場所も C ドライブのすぐ下である必要はありませんが，全角文字や空白を含むフォルダの下に入れてしまってはいけません。OndDrive のようなクラウドを指定するのも良くありません (探しに行った時にオフラインだとまたエラーになります)。
```

Q. A.2.8: ファイルが読み込めません

```
file(filename, "r", encoding = encoding) でエラー:
  コネクションを開くことができません
 追加情報:  警告メッセージ:
 file(filename, "r", encoding = encoding) で:
  ファイル 'foo.csv' を開くことができません: No such file or directory
```

A. 指定された場所にファイルがないので，読み込むことができないエラーです。確認すべきは，「プロジェクトを開いているか」，「プロジェクトフォルダの中に当該ファイルはあるか」，「ファイル名のミススペルはないか」です。プロジェクトってなんだという人は，RStudio の基本に立ち戻り，プロジェクトでファイルやフォルダを管理するようにしてください。プロジェクト管理については，Pp.89 にもその説明があります。

プロジェクトによる管理とは要するに，R が今見ているフォルダの場所を固定する方法です。プロジェクトを開くと，プロジェクトフォルダが「今見ているフォルダ (ワーキングディレクトリ)」になりますので，その中のファイルを参照することになります。プロジェクトフォルダの中に当該ファイルがないと読み込むことができませんので[a]，当該ファイルをプロジェクトフォルダ内に移してきてください。

Rmarkdown や Quarto ファイルの場合も，必ずプロジェクトフォルダの中に置くようにしてください。これらは Knit するときはファイルの置かれた場所を WorkingDirectory にしますので，プロジェクトフォルダの中にないとパスが通らない問題が頻発します。

[a] フォルダの位置を相対的・絶対的に指定してやれば，どこにおいていても読み込むことはできますが，この問題で悩んでいる人は相対・絶対パスの指定というところでさらに疑問が深まることになると思いますので，気にしないでくれて結構です。相対・絶対パスが知りたい人は，付録 C,189 でファイル場所とは何かを再確認してください)。

Q. A.2.9: ANOVA 君が読み込めません

```
file(filename, "r", encoding = encoding) でエラー:
  コネクションを開くことができません
 追加情報:  警告メッセージ:
 file(filename, "r", encoding = encoding) で:
  ファイル 'http:/riseki.php.xdomain.jp/index.php?plugin=attach&refer=
  ANOVA 君&openfile=anovakun_486.txt' を開くことができません: Invalid argument
```

A. ファイルを読み込みにいく先が URL，すなわちインターネット上になっています。ANOVA 君のファイルを一度手元の PC のフォルダにダウンロードし，そのローカルのファイルの位置を指定して読み込むようにしてください。

Q. A.2.10: 効果量として Hedges の g を算出してください。という指示について実行すると g ではなく d が出てしまうようなのですが，どうすればよいでしょうか。コードは次の通りです。

```
cohen.d(value ~ condition, data = dat, hedges.crrection = T)
```

A. Hedges の補正 (correction) のオプションが通っていません。スペルミスです。オプションのスペルが間違っているので無視されたので，関数名通り Cohen の d が算出されます。

Q. A.2.11: 描画の際に次のような注意が出てきます。これはどういう意味ですか。

```
Removed 671 rows containing missing values (geom_point).
```

A.「671 件の行で欠測値が含まれています」ということです。つまりデータセットの中に欠測値 (観測されていない，数値が入っていない行) が 671 件あったので，それは表示できませんでしたよ，という意味です。警告が嫌だということであれば，データセットの中で欠けているものを除外する必要があります。R の関数では，`na.omit` で欠測値を除外することができます。

Q. A.2.12: lm 関数を実行するコードでオブジェクトがないと言われます。

```
result <- lm( weight - height, data = dat)
```

A. 従属変数と独立変数とをつなげるのはチルダ (~) という記号です。そこがハイフン (-) になっています。スペルミスの一種です。

Q. A.2.13: lm 関数を実行するコードでオブジェクトがないと言われます。

```
result <- lm( weight - height )
```

A. データを与えていないので，`weight` というオブジェクトを探しに行って，見つからないというエラーです。`data` オプションでデータを与えてあげてください。

Q. A.2.14: 因子分析の Robust 法での RMSEA の p 値って超えてたらまずいですか。Standard(DWLS) の方だけで報告はだいじょぶでしょうか。

A. 非常に専門的な質問をしておられますが，まず，「因子分析」「Robust 法」「RMSEA」など，個々の用語の意味はわかっているでしょうか。キーワードによる検索で，「ロバスト法とは」「Robust の意味」などは答えが出てくると思いますが，これらを分析法の体系的な文脈の中に位置付けないと，答えられない種類の問題です。この例をもとに説明すると，Robust を辞書で引くと「壮健」「たくましいこと」などと出てきますが，もちろんそういう意味ではありません。ロバスト法を検索すると，「統計学の分野でロバスト推定法というやり方がある」「観測値に外れ値が含まれている可能性を考え，その影響を抑えることを目的とした手法」などの説明が出てきます。ロバスト推定法は因子分析に限らず，回帰分析など他の手法でも使われる考え方なので，このような一般的な解説になります。ですがここで知りたいのは「因子分析におけるロバスト推定」ですから，因子分析でロバスト推定するとはどういうことか，因子分析の観測値における外れ値とは何か，因子分析における推定とは何を推定するのか，そもそも因子分析とは何を目的としているのか，なぜ自分は因子分析方を使うのか，さらに何故因子分析の中のロバスト推定を使いたいと思っているのか，といったことがわかっていないと，この質問に正しく回答することができないのです。このように，複合的な要素について一回で質問しても，適切な答えに辿り着けないことがあります。聞くことは恥ずかしいことではありません。知らないことを知っているふりをすることが恥ずかしいことであり，知らないまま「みんなそうやっているから」「テキスト/ネットに書いてあったから」と看過してしまうことこそ恥ずべきことなのです。ちなみに，質問に対する回答を間違えることも恥ずかしいことではありませんから，「正しく答えられなければ恥をかく (から質問しない)」というのも同様に恥ずべきことです。こうした恥ずかしさ (保身) から，「専門用語を使ってそれっぽく質問してわかった感じになろう」というのは，かえって遠回りになります。実は回答よりも，質問の仕方でその人の理解度が明らかになってしまっています。このように書くと，なんでも反射的に「わかりません，教えてください」という人もいますが，それも適切な質問方法ではありません。教員がアドバイスできるのは，「答え」ではなく「理解」が欲しい人に対してだけなのです。質問する場合は，自分は何がわかっていて何がわかっていないのか，何が知りたいのかを明確に言語化して質問するようにしてください。

付録 B

標準正規分布から尺度値を求める計算方法

Likert 法では，態度が標準正規分布すると仮定するのでした。標準正規分布をカテゴリの相対度数で分割し，あるカテゴリ c の上限の確率点 z_c，下限の確率点 z_{c-1} の確率密度の差分を，相対度数 p_c で割ることで，尺度値 Z_c が下の式で得られます。

$$Z_c = \frac{(y_{z_{c-1}} - y_{z_c})}{p_c}$$

ここで y_{z_c} は標準正規分布をカテゴリで区分したときの，当該カテゴリ c までの累積確率点 z_c における確率密度，p_c はカテゴリ c の相対頻度です。図 B に記号の対応関係を示しましたので，確認してください。

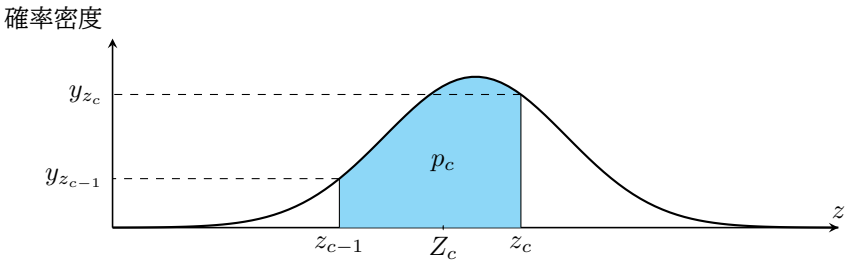

図 B.1　標準正規分布と対応する記号の確認

尺度値 Z_c を求める計算が確率密度 $y_{z_{c-1}}, y_{z_c}$ と，相対度数 p_c で算出されるというのは一見奇妙です。どうしてこのようになるのかを見ていきましょう。

まず標準正規分布の確率密度の式を確認しておきます。確率点 z における確率密度 y は次の式で算出できます[*1]。

$$y_z = f(z) = \frac{1}{\sqrt{2\pi}} e^{(-\frac{z^2}{2})}$$

この関数は確率密度の曲線を表しており，確率はその面積です。z_{c-1} から z_c までの面積 (確率) は，積分を使って

$$p_c = \int_{z_{c-1}}^{z_c} f(z)dz = \int_{z_{c-1}}^{z_c} \frac{1}{\sqrt{2\pi}} e^{(-\frac{z^2}{2})} dz$$

[*1] e^x を $\exp(x)$ と書くことも少なくありませんし，この e は自然対数の底で $e = 2.718...$ という実数です。この数字は微分しても変わらない，$(e^x)' = e^x$ という便利な特徴を持っています。

で表されます。

さて，ある確率点 z における密度の高さを $f(z)$ としたとき，z の微小な増分 Δz を考えると，区間の面積 $S = f(z)\Delta z$ を考えることができますから，逆にある点 z が知りたい時は

$$z = \frac{\sum Sz}{\sum S}$$

とすれば良いことになります。積分はこの微増分 Δz の極限を

$$\lim_{\Delta z \to 0} \Delta z = dz$$

と考えることですから，ここで考えたいのは

$$Z_c = \frac{\int_{z_{c-1}}^{z_c} f(z)zdz}{\int_{z_{c-1}}^{z_c} f(z)dz}$$

になります。分母は確率密度関数の積分ですから面積すなわち確率で，ここでは p_c であり，その面積を実データの相対度数で代えてもいいでしょう。

分子については少し式の変形が必要です。記号を見やすくするために，$a = z_{c-1}, b = z_c$ と書き換えておきましょう。

$$\begin{aligned}\int_a^b yzdz &= \int_a^b \frac{1}{\sqrt{2\pi}} e^{(-\frac{z^2}{2})} zdz \\ &= \frac{1}{\sqrt{2\pi}} \int_a^b e^{(-\frac{z^2}{2})} zdz\end{aligned}$$

ここで変数変換をして，$u = -\dfrac{z^2}{2}$ とすると

$$\begin{aligned}\frac{du}{dz} &= -z \\ du &= -zdz\end{aligned}$$

となり，積分の下限は $-a^2/2$, 上限は $-b^2/2$ になりますから，以下のように展開できます。

$$(\text{与式}) = -\frac{1}{\sqrt{2\pi}} \int_{-a^2/2}^{-b^2/2} e^u du$$

$\int e^u du$ は e^u なので

$$= -\frac{1}{\sqrt{2\pi}} \left[e^u\right]_{-a^2/2}^{-b^2/2}$$

$$= -\frac{1}{\sqrt{2\pi}} e^{(-\frac{b^2}{2})} - \left(-\frac{1}{\sqrt{2\pi}} e^{(-\frac{a^2}{2})}\right)$$
$$= \frac{1}{\sqrt{2\pi}} e^{(-\frac{a^2}{2})} - \frac{1}{\sqrt{2\pi}} e^{(-\frac{b^2}{2})}$$

$y = \dfrac{1}{\sqrt{2\pi}} e^{(-\frac{z^2}{2})}$ であることから，

$$= y_a - y_b$$
$$= y_{z_{c-1}} - y_{z_c}$$

以上のことから，リッカート法において標準正規分布をもとに尺度得点を決めるには，

$$Z_c = \frac{(y_{z_{c-1}} - y_{z_c})}{\int_{z_{c-1}}^{z_c} f(z)dz} = \frac{(y_{z_{c-1}} - y_{z_c})}{p_c}$$

とすれば良いことになります。

この計算に至る理論的背景は，より専門的には**系列範疇法 (Method of Successive Categories)** と呼ばれ，順序カテゴリに数値を付与する心理測定論，精神物理学理論からきています。詳しくは Guilford (1954 秋重訳 1959) や西村 (1977) も参照してください。

付録 C

電子計算機のイロハ

C.1　前置き

このセクションは心理統計ではなく，コンピュータについての四方山話をダラダラと書いています。そんなの聞かなくてもわかってるよ，という人もいるかもしれませんし，知らなくてもみなさんはきっとスマートフォンやタブレットを使えていることと思います。しかし知らずに使うことと，知ってて使うことには大きな違いがありますし，今後大学でレポートや論文を書いたり，それに必要な統計処理をするためにも，計算機の基本的な特徴を知っておくと，トラブルに出会した時に「あぁこれってひょっとして」というヒントが得られたり，納得できるようになるかもしれません。知らなければ「何だかわからないけどパソコンが壊れた」というか，「パソコン運が悪い」「自分はパソコンが苦手なのだ」と間違えた帰属をしてしまうことになります。

コンピュータ関係の授業で聞いたことがある話，これから聞く話もあると思いますが，もし苦手意識を持っている人がいたらこれを機に再入門するつもりで読んでください。

C.2　コンピュータの基礎

21 世紀に生きる私たちは身の回りをコンピュータに囲まれて生きています[*1]。それは携帯電話の形をしていたり，ノートパソコン，デスクトップパソコンの形をしていたりします。また時々テレビなどで報道されますが，気象予報や飛沫がどのように飛び散るかをシミュレーションする大型計算機「富嶽」などもコンピュータですね。これらは形は違いますが，いずれも電子計算機であり，電子計算機には次の 5 つの装置があります。

入力装置　キーボードやマウス，タッチパネルなどを使って情報を取り込むデバイス (装置)

出力装置　モニタやプリンタ，タッチパネルなどを経由して情報を出力するデバイス

演算装置　プログラムの命令に従って計算処理 (四則演算や論理演算) をする装置。一般に電子計算機の中央で一括して処理するので，Central Processing Unit(CPU) と呼ばれるものです。

制御装置　演算結果に従って他の装置に指示を出す装置のこと。演算装置とまとめて CPU に実装されています。

記憶装置　計算結果などの情報をいったん保持しておく装置のこと。コンピュータの内部にあって一時的な

[*1] 私は 1976 年生まれですが，生まれた頃は周りにコンピュータなんかありませんでした。小学生の頃にマイコン (マイクロコンピュータ，小さなコンピュータという意味でもありますが，My Computer，私のコンピュータという意味でもあります。つまり個人単位でコンピュータが使えるようになった，というだけでも大きな出来事だったのです。) という言葉が出てきて，なんかかっこいいなと思った記憶があります。私が 10 歳になったころ，ビデオゲーム (テレビゲームでやるゲームが家庭でできるようになり，それをこのように呼びました。) が身の回りに出てきました。ファミリーコンピュータ，とくに「スーパーマリオブラザーズ」によって日本中の子供たちが熱狂したのが 11 歳の頃，「ドラゴンクエスト」によって社会問題になったのが 12 歳の頃になります。ともかくこの頃は，コンピュータといってもゲーム機のような扱いでした。

計算に使われる一次記憶装置，ハードディスクドライブ (Hard Disk Drive,HDD) やソリッドステートドライブ (Solid State Drive,SSD) などの二次記憶装置など。

最後の記憶装置については，一次記憶装置，二次記憶装置と種類が分かれていますがこの区別は簡単で，電源を落とした時に記憶が消えてしまうのが一次記憶装置，電源を落としても記憶が消えないのが二次記憶装置です。たとえば $12+38=$ という計算をするとき，頭の中で「えーっと一の位が 2 と 8 だから 10 になって繰り上がるから･･･」と考えてから，ノートに $=50$ という答えを書くと思います。次の問題に進むと，先ほどの「1 繰り上がるから･･･」という情報は忘れてますよね。でもノートに書いた $12+38=50$ というのは残っています。このノートがいわば二次記憶装置であり，頭の中で一時的に保持していた情報が一次記憶装置ということになります。一次記憶装置は RAM(Random Access Memory) とも呼ばれます[*2]。

ところで，コンピュータがやっているのは計算だけです。私はこの資料を PC に向かって書いており，キーボード (入力装置) を叩きながら，画面 (出力装置) を見て文字を連ねています。これも「キーが押されたら文字を表示させ，その文字列を記録する」という処理を機械が淡々とこなしているに過ぎません。あるいはマウスやトラックパッドで，アイコンを指し示し，カチリと押す[*3]ことで選択し，押し込んだまま移動させ (ドラッグ)，離すことで別の場所に置いたりします (ドロップ)。トラックパッドの場合は 2 本指で同時に押したりしますし，タッチパネルの場合は二本指を広げたり (ピンチアウト)，逆に二本指を狭めたり (ピンチイン)，3 本以上の指でファサーっと触って場所を広げたりします。たとえば「ファイルを掴んでゴミ箱に捨てる (削除する)」というのが我々にとっての操作ですが，コンピュータの内部では実はこんなことをしていません。ファイルは (二次) 記憶装置に書き込まれた情報です。記憶装置は原稿用紙のように小さなマス目がたくさんあって，ファイルとはそのマス目の XXX 番目から YYY 番目までの情報，ということです。このファイルを削除するというのは，記憶装置のある場所 (アドレス) に「削除されたものなので画面に表示しない」という情報を書き込む，という操作をしているだけです。じゃあなぜ私たちは「ゴミ箱にドラッグ＆ドロップ」なんてするのでしょう？ それはそのほうがわかりやすいからですよね。「ファイルを削除する」というのは，「メモリアドレスの XXX 番地に別の情報を書き込む」という操作だと言われてもピンとこないので，コンピュータが人間にとってわかりやすい表現をして見せてくれているのです。このユーザにとってわかりやすい幻を見せてその気にさせてくれるというデザインのことを，ユーザーイリュージョンと言います。ともかくこういう「画面で見ながら操作する」ことをグラフィカル･ユーザ･インターフェイス (Graphical User Interface,GUI) といいますが，これのおかげでコンピュータの操作は随分楽になっています[*4]。

C.3 コンピュータの歴史

コンピュータの装置，すなわちハードウェアについての解説につづいて，ソフトの側面についても解説を加えようと思うのですが，そのためには少し歴史的な流れを説明したほうがわかりやすいかもしれません。

コンピュータの発展の歴史は，小型化の歴史でもあります。最初にできたコンピュータは ENIAC といいます。1946 年の話です。この ENIAC は 27 トン，広さにして倉庫 1 つ分 ($167m^2$，90 畳以上の広さ) が

[*2] 実はこのように人間を 1 つのコンピュータに喩えて，そこではどのように計算がされているのか，人間の記憶装置や演算装置，入出力装置の特徴はどうなっているのか，というのを研究するのも心理学の仕事です。記憶や演算，制御については認知心理学や学習心理学，入出力については知覚心理学や生理心理学が専門的に扱っています。そういう意味でもコンピュータの登場は心理学に大きな影響を与えているのですが，それはまた別の講釈で。

[*3] 押すときの音から，この操作をクリック click と言います。2 回続けて押すことをダブルクリックと言います。

[*4] 実は人間の意識もこのユーザーイリュージョンのようなもので，実際の体の動かし方や感覚情報の受け止め方，処理の仕方は別ですよね。すべての情報を意識に上げるのではなく，「私は XXX をしている」と身体がそれっぽい幻想を投影して見せてくれているのが意識の正体ではないか，という議論があります。興味のある人は Norretranders (1999 柴田訳 2002) を読んでみてください。

必要なもので，真空管を使った計算機でした。軍事的な計画のために開発されたオーダーメイドのものですから，一般人が触れるはずがないものです。時代が下がって真空管が半導体，IC チップになった頃，やっとサイズが小さくなって，家庭用・個人用のコンピュータというのができるかもという時代が来ました。Apple コンピュータの創始者，スティーブ・ジョブズとスティーブ・ウォズニアックが最初のパソコン (Personal Computer,PC)，Apple I を発売したのが 1976 年。爆発的に売れた Apple II が発売されたのが 1977 年です。AppleII はブラウン管表示装置とキーボードを持っていましたので，今の PC の原型とも言えるかもしれません[*5]。PC を作っている会社は Apple だけでなく，IBM や DEC などがありましたが，まだこの頃はパーソナルなレベルのものよりも大型計算機の開発が進んでいました。IBM が PC を作ったのは 1980 年で，この頃から小型化が進められていきます。

私事で恐縮ですが，1976 年に生まれた私が初めて PC を手にしたのは 1991 年，高校入学のお祝いで買ってもらった Fujitsu の FM-Towns という機体でした。この頃は「マルチメディア」という名前もなく，「ハイパーメディア」と読んで売り出していました。この機体は他の PC(NEC の PC-9801 や Sharp の X68000 シリーズが有名でした) とは違って，CD-ROM ドライブをつけていたことが画期的だった時代です。その後 1994 年に大学生になりましたが，この頃は連絡を取り合うツールはポケベルが主流であり，携帯電話 (や PHS) のような個人端末は高級品という時代でした。大学に入るとコンピュータを学ぶ授業があり，アカウントをもらったりするのですが，それは大学が持っている大型計算機に端末からアクセスするためのものでした。いろんな部屋にあるのは「端末」で，それほど機能の優れた PC ではなく，複雑な計算 (統計的な計算など) は大型計算機に仕事を依頼しその返信を待つ，というスタイルでした。関西私立のマンモス校でしたので学生数は非常に多かったのですが，多くの学生が一度にアクセスしても，大型計算機はものすごくものすごーく計算が早かったので，瞬時に回答をもたらしてくれるものでした[*6]。つまり，まだ「専門的な計算は大型計算機」という時代であり，パーソナルなコンピュータになるにはもう少し時間が必要でした。

どれぐらいの時間が必要だったかというと，実はその次の年なのです。1995 年，Windows 95 という OS が発売されました。これを機に日本でも PC がどんどん浸透していくことになります。Windows95 は物凄いんだぞ，と発売前からテレビでも散々とりあげられ，発売日には行列ができて真夜中のカウントダウンと同時に大フィーバー，という売れ行きでした。当時のそれは何が凄かったのでしょう？ コンピュータにはそれ動かす基本ソフトが必要です。HDD にデータを書き込み，キーボードからの入力をディスプレイに表示する，といったごく基本的な装置を統括し，メモリ番地をファイルという単位で扱うと言った基本的な操作は OS いうソフトウェアが担当します (図 C.1)[*7]。この OS，大型計算機は Unix と呼ばれるものを使っていましたが，各企業が個人向けにコンピュータを売り始めるときにも当然必要で，各社で開発もしていましたが，PC の共通規格をつくることで OS 部分は共有できるようになりました。そこを提供したのが Microsoft 社のビル・ゲイツです。どんなパーツで作られた PC であっても Windows という OS が共通のフィールドを用意してくれるので，ユーザは Windows で動くアプリケーションを選ぶだけで良い，ということになったのです。

そして Windows95 は，GUI，つまり「ファイルを掴んでポイ」といった直感的な操作で使えることも大きな特徴でした。大型計算機で使われている Linux は基本的に Command User Interface,CUI で，黒い画面にプログラムを書いて実行するといった手法で，初心者には人気がなかったのです。GUI については，その頃 Apple を追放されていたスティーブ・ジョブズが，NeXT という会社で GUI を備えた OS を開発していました。この NeXT はその後 Apple に買収され，ジョブズは Apple に戻って活躍することになります。その頃から巷では，Windows の GUI は Apple の OS を真似したものだと非難されていたのですが，商業的に

[*5] Mac の歴史については Isaacson (1995 井口訳 2011) が読み物としておもしろいですよ。

[*6] Time Sharing System,TSS，時分割システムとよばれる機構です。命令を小さな単位に分割し，それを順次捌いていくという方法でした。

[*7] 物理的な機構と OS との間に Basic Input/Output System,BIOS というのが入りますが。

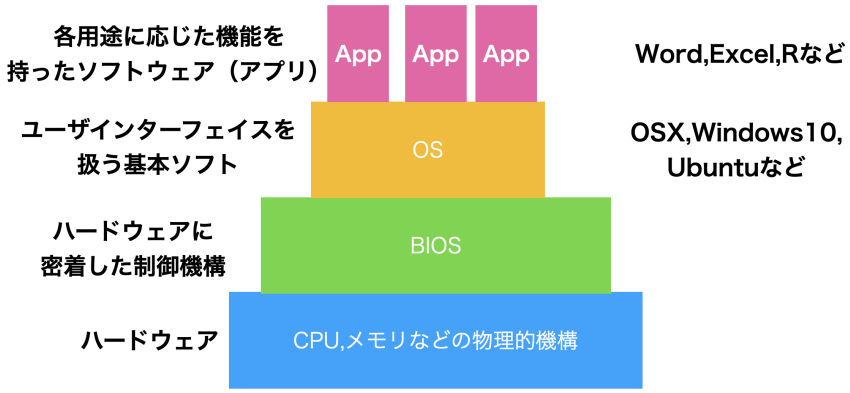

図 C.1 コンピュータのハードウェアとソフトウェアの関係

は Windows が大勝利，というわけです。ともかくこれを機に企業などはもちろん一般家庭でも PC を使うような時代になりました。

ちなみにインターネットが広まったのもこの頃です。私は大学 3 年生の時，大学の授業で初めてインターネットを介して世界の情報を得る，という経験をしました[*8]。その頃から徐々に，大型 PC のアカウントではなくインターネットで使えるメールアドレスというのを個々人が持つようになりました。携帯電話も廉価な PHS が広まり，メッセージだけでなく徐々に音声，写真，短い動画が遅れるようになっていきます。当初は当然画質・音質も今とは比べものにならないのですが，それでもインターネットを経由してつながるという経験は想像を超えたものでした。

皆さんはすでに，携帯電話やインターネットがある時代に生まれた世代だと思います。こんな話はすでに過去のものであり，不便な頃の話を聞かされても困る，と思うかもしれません。私の世代は，幸いにもこのようにちょうどコンピュータが使われはじめ，広がり，高性能になっていくにつれて育ってきていますので，そのぶん機械の根本的な理解に直結しやすい社会環境にあったのです。みなさんは，便利な時代ではありますが，言い換えると「初心者が苦労しないように補助輪をつけておく」「何もしなくてもできているような感覚が得られるようなイリュージョンを見せておく」という状態におかれているので，補助輪が対応できない道に進んだり，多くの人とは違う使い方をするときに，急にサポートが外れどこで困っているかわからなくなる，ということになりかねません。非常に大雑把な Historical Review ではありますが，理解の一助になればと思います。

ところで，先ほどサポートという言葉を使いましたが，このサポートを商売にしているのが Microsoft や Apple という IT 企業です。これらの会社は，ユーザが使いやすいようにソフトウェアを提供してくれますが，有償ですし想定外の使用をするユーザに対してはサポートをしてくれません。逆にいうと，「決められた路線を走らなければ面倒を見ない」ということでもあり，これはユーザに不自由を強いているとも言えます。また，どのような仕組みで動いているのかを尋ねても，企業秘密と言って答えてくれません。コンピュータは誰でも自由に使えるべきものであり，勝手にユーザの情報を盗んだりしていないか，と言ったチェックをするためにもオープンであるべきではないか。そういう考え方に基づく，フリーソフトウェアというソフトウェアのあり方があります。Unix という OS を PC 用にした Linux がそうであり，オフィスソフトの LibreOffice，統計環境の R などもこの精神に賛同するものです。フリーソフトウェアは自由であり，無償です。お金と秘密を払ってサポートを受けるのではなく，ユーザが相互に助け合ってオープンで自由な世界を広げていこうという活動です。急

[*8] 教職関係の科目で，教育技術として今後使われるだろうということで担当教員が実演してくれました。隣の部屋から電話線を延長し，モデムというパソコンの信号を音情報に帰る機器を繋げて，NASA の Web ページを Netscape というブラウザでみたのが初めての体験でした。

に何の話なんだ，と思うかもしれませんが，心理学ひいては科学的活動すべてにおいて重要な問題であることをご理解いただきたいと思います。

C.4 情報の単位

コンピュータを取り巻く世界の話はこれまでにして，ソフトの側面についての解説に入りましょう。

コンピュータは文字，音，絵，動画ファイルいずれについても，すべて 0 か 1 のデータとして管理します。0/1 の 1 つの単位を 1 bit(ビット) と言います。1 bit であれば Yes か No か，という二択の情報しか提供できませんが，これが 7 つあれば $2^7 = 128$ ですから，これで 128 種類の状態を表現できます。コンピュータは一般に，8bit で 1 つの単位として計算します。8bit のことを 1 byte(バイト) と言います。この 1 byte が 1024 集まったものを 1kb(キロバイト) といいます[*9]。1kb の次は，1024kb=1Mb(メガバイト) です。さらに 1024MB=1GB(ギガバイト) で，1024GB=1TB(テラバイト)，1024TB=1PB(ペタバイト)，1024PB=1EB(エクサバイト) と続きます。K,M,G,T,P,E といった名称は 1000 倍ごとに変わる大きさの桁をあらわしているのであって，「ギガが減る」というのは本来意味をなさない表現です[*10]。

このビット・バイトは情報に関する基本単位なのであちこちに使われます。記憶装置について使われるとき，一次記憶装置も二次記憶装置も同じ単位なので混同するかもしれませんが，2021 年現在では二次記憶装置の単位は GB から TB が使われます。USB フラッシュメモリーや外付け HDD などは数 TB の容量が一般的でしょう。これに対して，一次記憶装置は 4〜64GB ぐらいが相場かと思います。一次記憶装置は暗算の途中経過のように一時的に記憶する場所に過ぎないので十数 G でも問題ありませんが，二次記憶装置は結果の記録なので大きければ大きいほど余裕が持てますね。たとえば数 TB でもテレビドラマや映画を何本も記憶できるのですから，PB や EB なんて使うのかな，と侮ってはいけません[*11]。すぐにそれぐらいのサイズが必要な時代が来ることでしょう。

実際，それぞれ単位ではどれぐらいの情報が記録できるのでしょうか。1 byte は 128 文字表現できるので，英語のアルファベット 26 文字に加え，数字や簡単な記号であれば 1 byte で表現できます。たとえば A という文字は 01000001，B という文字は 01000010，小文字の a は 01100001，と言ったように 0/1 の文字列 8 個と一対一対応させて考えるのです。日本語は 1 byte では足りませんので，一文字あたり 2byte が割り当てられます。また 1KB(=1024byte) は 500 文字ですから，原稿用紙一枚ぐらいになります。1MB(=1024KB) は文字だけだと新聞一紙 (朝刊の 40 ページ分) ぐらいで，昔の記録媒体であるフロッピーディスク一枚に保存できるのがちょうど 1MB でした[*12]。ちなみに「カメラ映像 + 音声」のオンラインビデオ会議を 1 時間やると 200〜300MB ぐらいの容量をやりとりしていることになります。ビデオ会議では文字 (チャット) だけでなく，画像 (動画) や音声も送り合いますね。実は画像や音声も，0/1 に置き換えています。音声の場合，1 秒間を短い間隔にくぎります。この区切りのことをサンプリング周波数といい，たとえば 44.1 kHz という単位は 1 秒間に $44.1 \times 1000 = 44100$ 点のデータの採取をします。この 6 データ点において，音の振動幅を区切ります。ビットレートと言いますが，たとえば 16bit で区切る場合は $2^16 = 65,536$ 段階で区

[*9] キロメートルやキログラムのように，キロは 1000 の単位を表す言葉ですが，コンピュータは 2 進数なので 1000 ちょうどではなく 1024 で 1 つ上の位に上がることになります。

[*10] 同様に「USB を紛失する」というのもよく聞くおかしな表現です。USB は Universal Serial Bus の略で，データ転送規格のことを指します。なくすことができるのはそれにつなげるメモリースティックなどです。

[*11] 私事ですが，私が 1991 年に生まれて初めて手した PC，FM-Towns はハードディスクが 40MB，RAM は 2MB で，CD-ROM がついていましたが，それは 360MB の容量でした。購入するときに，「ハードディスクが 40MB もあって何を記録するんです？ CD-ROM なんか情報が詰まり過ぎてますよ！」と店員さんに笑われたのを今でも覚えています。数年後，PC の動きが遅くなったので，2 万円で 2M の RAM を追加したのも良い思い出です。今でこそ，2MB なんて USB フラッシュメモリーでも売ってないほど微小なサイズですが。

[*12] 正確には 1.2MB 入る規格 (2DD) と 1.44MB 入る規格 (2HD) とがあります。

切り，その高さの音がある (1) かない (0) かで表すわけです。このように時間と音階を細かく区切り，その目に情報があるかないかを積み重ねてデータとするわけです。テキストよりも圧倒的に情報が多くなるのが分かりますね。画像も同様に，図面を細かい単位 (ピクセルなど) で分けて，そこの色合いを色々な段階で区切ります。色は R(赤)G(緑)B(青) の組み合わせで表現でき，それぞれを 8bit$= 2^8 = 256$ 段階で表現したりします。一点一点にその情報がありますから，図の情報も非常に多くなるのがわかると思います。動画はその画像が時系列的に細かく分割されたものと思ってください。このように分解しますので，テキスト，音声，図，動画の順にデータサイズが大きくなります。通信機器が最初ポケベル＝数 byte の情報しか送れなかったものから，徐々に絵文字，ショートメッセージ (音声)，写真[*13]，動画が送れるように発展していきました。今では町中のあちこちで，誰もが手軽に動画をモバイル端末で見られるようになっています。あらためて，すごい進歩ですね[*14][*15]。

ところで，1 byte は 256 種の情報が記録できるので，英語のアルファベットや数字は 1 byte あれば十分だが，日本語や中国語など，英語以外の言語は文字種が多いので，2byte で一文字を表すという話をしました。この 2 バイト文字も，たとえば「あ」とか「亜」という文字に 111000111000000110000010 とか 111001001011101010011100 という文字列を割り当てるのですが，言語ごとによってどの数字をどの文字に割り当てるかという対応表が変わってきます。これを文字コードと言います。日本語はかつて Shift-JIS というコードで変換していましたが，今は世界のあらゆる言語に対応している共通規格である，UTF-8 という文字コードで変換することが一般的です。ところがなぜか，日本の Windows OS だけ Shift-JIS をいまだに使い続けており，他の PC とファイルをやり取りするときに文字コードの変換エラー問題が起きます。ファイルを開いて文字化けをしたりとか，プログラムが実行される際に「ファイルにアクセスできない」というエラーが生じたりするのは，この文字コードの問題が大きいのです[*16]。受身的な対策法になってしまいますが，PC でつかうユーザ名やファイル名などは半角英数文字を使い，短くした方がこうしたエラーに出くわしにくくなります。逆に，全角文字やスペース (空白) などを含んだファイル名，やたらと長いファイル名を使っていると，こうした問題に出くわしやすくなるということです。

C.5 ファイルの種類と拡張子

ここまで述べてきたように，計算機というのは基本的に物理的実体 (記録装置，記憶装置) の上で 0/1 のデータをやり取りしているだけです。記録 (記憶) 装置上に置かれている情報のセットは「ファイル」という形で記録されています。スマートフォンやタブレットは，ユーザの利便性のためにファイルの存在を意識しなくても良いようになってはいますが，バックエンドでは実行されるアプリケーションもファイルですし，開かれる音声や動画もファイルです。とくにパソコンでは，どの媒体，どのアプリで使うどういうファイルかを識別するために，拡張子 (かくちょうし) と呼ばれる識別記号をファイルの後ろにつけています。拡張子はファイル名の背後にピリオドで区切って追記されています。ついてないように見えても，OS がそれを表示させない設定にして

[*13] できた当時は写真を撮ってメールができることをとくに「写メールする」と言い表したほどです。

[*14] 実は音でも画像でも，分割してそのままデータにしてしまうと膨大になりすぎるので，人間が気付きにくい周波数や色合いなどは削除して作ります。これを非可逆圧縮処理と言います。一度落としてしまった情報は戻らない，という意味です。ライブや生きている人間が処理している情報は，携帯の画面から得られるものの何億倍もの情報量なんですよ!

[*15] デジタル化のすごいところは，こうした文字，音，図版，動画といったものを bit という共通の単位に落とし込んだことです。こうすることですべて一元的 (bit という共通次元) で処理することができるようになったのです。メディアの違いが問題にならなくなり，0/1 の情報であれば複写も簡単にできてしまいます。情報化社会においては情報に特別性はなく，情報があるかないか，それを生み出せるかどうかこそが重要なのです。

[*16] Windows だけ世界標準から外れているので，早く修正して欲しいのですが，歴史的な経緯からユーザ数が多くて切り替えられないでいることと，こうした違いがあることをユーザに説明しない (素人は知らなくていい，と馬鹿にされているようなものです) ので，問題が解決される日はまだ先になりそうです。

あるだけであることに注意してください。代表的な拡張子と，それに対応づけられているアプリケーションは次のようなものがあります。

.docx マイクロソフト社の文書作成アプリケーション，Word で使うファイル

.xlsx マイクロソフト社の表計算アプリケーション，Excel で使うファイル

.pdf Adobe 社が開発した Portable Document Format 形式。OS が違っても同じレイアウトで文書を表示できるのが利点で，PDF 形式を読むことができるアプリケーションは多数。

.txt シンプルな文字だけのテキストファイル。文字の飾りやレイアウトなどの情報がない最もプレーンな形式なので，OS が違っても文字コードさえ合っていれば読むことができる。

.mp3 音楽，音声のファイル。音声データには他の種類もあります。

.jpg 画像のファイル形式の一種。

.png 画像のファイル形式の一種。

.csv comma/character separated variables ファイル。変数をカンマ (,)，あるいはタブ，半角スペースなど文字コードで区切ったファイルという意味で，中身は.txt と同じく装飾のない文字/数字だけであり，文字コードさえ間違えなければ OS を問わずに読み書きできる。データのやり取りはこの形式で行われることが多い。

.zip 圧縮ファイルの一種。1 つまたは複数のファイルをパックして圧縮してあるもの。ファイルの冗長な部分をうまくまとめてコンパクトにまとめ上げるため，ファイルサイズが小さくなるし，複数のファイルもひとまとめにできる。また圧縮の際にパスワードをかけることもできるため，メールなどに添付する場合はこの形式にまとめられることが多い[*17]。可逆圧縮であり，圧縮されたファイルは展開する (解凍するともいう) ことでパッキングを開封できる。zip ファイルの圧縮/展開は各種 OS が標準的に対応している。

`.txt` や `.csv` といった形式は，「装飾のない，文字だけの」ファイルです。こうした種類のことを ASCII ファイルと言います。メモ帳などのエディタと呼ばれるプログラムで読み書きできます。逆に `.docx` など特定の会社が提供するアプリケーションに対応しているファイルは，アプリの中でのさまざまな操作・装飾を暗号化して保存しており，メモ帳などで読んでも意味がわかりません。対応しないアプリでは開くこともできません。こうした形式は ASCII ファイルに対してバイナリファイルと言います。

OS は拡張子を見てファイルの種類を判別し，そのファイルを開くのに適したアプリケーションを自動的に起動し，開いてくれます[*18]。`.csv` ファイルは Excel などのアプリケーションで開くことも当然できますが，その際文字コードのエラーが生じたり，保存するときに文字コードを変えたりして，形式・内容が気づかずに変わっていることがあります。Windows も良かれと思ってやっていることなのですが，処理が徹底してないのか，かえって不便になってしまっています[*19]。

[*17] **PPAP** というとピコ太郎の楽曲を思い出す人もいるかもしれませんが，圧縮ファイルの文脈では“「Password つき zip ファイルを送ります。Password は次のメールで送ります」Angoka(暗号化) Protocol の略です。つまりメールでパスワード付きのファイルを送り，そのファイルを開くためのパスワードをまたメールで送るという，日本でよく見られるおかしな風習です。おかしな，というのは，メールがハッキングされていたらパスワードもどうせバレるわけで，同じメールに書いてあるのはもちろん馬鹿馬鹿しいですが，すぐ次のメールに書いてあるのも同じぐらいに馬鹿馬鹿しいことです。情報セキュリティ対策手法のつもりで行われる慣習が広まっていますが，PPAP の標語のもと，馬鹿馬鹿しいのでやめましょうという風潮になってきました。

[*18] 見たことのない拡張子の場合は，どのアプリケーションで開くべきか尋ねてくるでしょう。

[*19] 実際，この授業での課題データを UTF-8 形式の csv ファイルで提供しても，Excel で開いたばっかりに文字化けして分析できなくなる，という相談がこれまで多く寄せられています。根本的な解決策として，Windows を使うのをやめることをお勧めします。

C.6 クラウドとは

すでに述べたように，計算機は基本的に 0/1 データのやりとりであり，それを保存してあるのがファイルとよばれるものです。ファイルは HDD や USB メモリ，SSD に保存することができます。

ところで，最近はこうした手元の物理的実体にファイルを置くことに加えて，クラウドに保存することも少なくありません。クラウドとは雲という意味で，インターネットの向こう側のどこか，ということを意味します。ですが，基本的にはインターネットで繋いだその先にも電子計算機があるのです。たとえばパソコン A とパソコン B をケーブルで繋ぐと，パソコン A からパソコン B のファイルにアクセスできます[*20]。このケーブルをどんどん伸ばすと，遠く離れていてもこの操作ができます。このケーブル網を世界レベルに広げているのがインターネットです[*21]。

ここで覚えておいて欲しいのは，当たり前のようですが，ネットといっても基本的には電子計算機と電子計算機を繋げている実体がどこかにあって，ファイルのやりとりをしているだけだということです。ブラウザはウェブサイトの情報を書いたファイルを取り込んでホームページを見せてくれていますし[*22]，Youtube は動画のファイル，Instagram は画像のファイルへのアクセスをして見せてくれているのです。最近ではクラウドサー

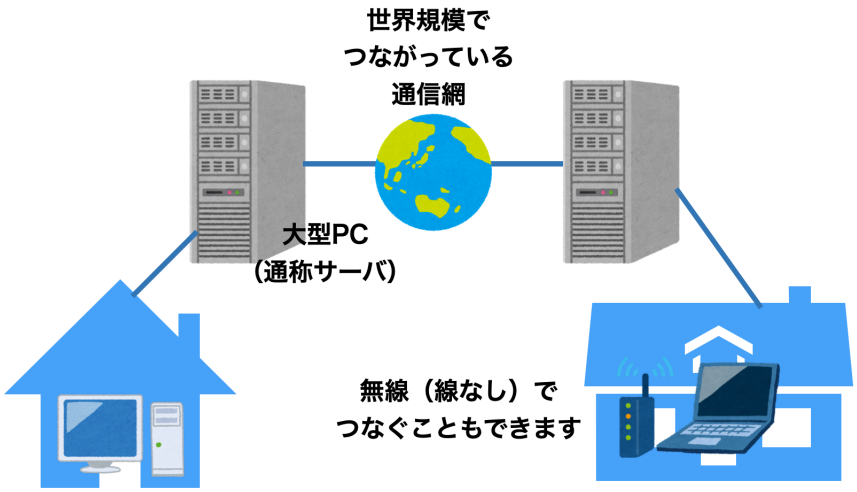

図 C.2 インターネットの世界

ビス，というのがよくあります。中でも Dropbox とか OneDrive，iCloud などが有名です。これらは，ファイルをインターネットを介した別の大型電子計算機に保存 (アップロード) し，インターネットを介して読み込み (ダウンロード) して使う，というものです。手元の電子計算機の中に記録されているファイルのことを「ローカル」，サーバなど遠隔地にある大型機に記録されているファイルのことを「サーバ」「クラウド」と呼んで区別しますが，このローカルとサーバのファイルを常に同じものに同期しておく便利なシステムです。こうしたクラウドサービスを使うと，たとえば大学で作業して保存したファイルを，USB メモリに保存して自宅のパソコンにコ

[*20] もちろんファイルの情報をどのように信号に変えて送受信するか，アクセス権限はどうするかといった細々したことを調整しなければなりませんが，そのあたりの仕事をやってくれるのが OS のありがたいところです。

[*21] インターネットは軍事通信網として始まりましたが，それを電話回線や企業内通信網，大学間通信網などと接続しあって世界中に広げています。網と網が相互に (inter) 繋がっているので inter net であり，大学内・企業内のネットワークのことはイントラ (intra) ネットと言います。ちなみに一般用語としてのインターネットは Internet と大文字で書き始めます。

[*22] ホームページとは本来ブラウザが最初に開くページのことを指し，企業や個人が情報発信しているページのことはウェブサイトというのが適切です。

ピーする，という操作をしなくても，大学でクラウドサービス上のフォルダ内に保存しただけで，自宅のパソコンにそのファイルがコピーされているため[*23]，意識せずにそのまま作業を続けられるということです。自動的にバックアップをとってくれているとも言えるので便利ですね。

もちろん注意すべきこともあります。「他の人に見られたら困るファイル」，「アプリケーションの実行ファイル」などは通信網の向こうではなく，手元 (ローカル) に置いておきましょう。個人情報・機密情報などを，クラウドドライブに保存すると，悪意を持った人が大型計算機に攻撃を仕掛けて情報を盗んでいく可能性があるからです。情報化の怖いところは，取られても気づかない (コピーすればいいだけで元ファイルになんの影響もない) ところにあります。加えて，取られたものをばら撒かれる＝インターネットを介して誰でもアクセスし保存できるようにされると，すべてを回収できなくなるのも問題です。失言が記録されて拡散されると大変なことになるのは，みなさんもこの時代に生きる人間のマナーとして色々見聞するところと思います。また，クラウドサービスで自動的に同期されるといっても，アプリケーションの実行ファイルなどはローカルに保存するべきです。アプリケーションは実行に際してさまざまな関連ファイルにアクセスしますので，1 つでも場所が違うところがあるとエラーになって動かなくなります。同じ OS でも，です。インターネットからとってきたソフトウェアをうっかり OneDrive に保存してしまうと，アクセスできないエラーで起動しない，ということもありますので注意してください[*24]。

余談ですが，R などプログラミングを推している時に最も頻発する質問が「ファイルがありません/読み込めません」です。その理由は大きく 2 つあって，1 つはスペルミスです。ファイル名の大文字・小文字の違いということもありますし，拡張子のまえに半角スペースが入っている，というよくみてみないと気づかないようなものもあります[*25]。2 つめの問題は，パスが通らないことです。デスクトップに置いたのに見つからない，ということもよくあります。教員や TA を悩ませるタイプの問題として，同じファイル名・フォルダ名が別の場所にあることがあります。フォルダ A の中にファイルを置いたのに見つかりません，という質問に対応していて，確かにフォルダ A を開いているのにな，と思ったら別の場所のフォルダ A だった，ということがあるのです。パスは隅々まで同じでなければ意味がないので，自分がどこに何を置いたのかをしっかり把握しておきましょう[*26]

C.7 ファイルの位置の指定

ここでファイルとそのパスについての話をしておきたいと思います。

計算機が情報を 0/1 で管理し，それらがファイルとなってどこかに保存されている，ということでした。私たちは Finder や Exploer などファイルブラウザをつかって，実行したいファイル，参照したいファイルを探していきますね。ファイルはまとめてフォルダの中に含まれていますし，フォルダの中にフォルダがあるといった，階層状態になっていることも少なくありません。ちなみに

[*23] これはユーザが特段の指示をしなくても，アプリケーションがユーザの見えないところで (バックグラウンドで) アップロード，ダウンロードの作業を進めているからです。パソコンはシャットダウンしていなければ，裏でこうした作業を淡々とこなし続けてくれています。

[*24] マイクロソフト社は，これまたユーザのためを思ってやっているのかもしれませんが，デフォルトで OneDrive に保存させようとします。それでうまくインストールできなかったという相談も多々寄せられています。抜本的な解決策として，Windows を使わないことをお勧めします。

[*25] これはたとえば，`sample.csv` というファイルをダウンロードしてね，と指示した時に，複数回同じ場所に保存してしまって，機械が，`sample (1).csv`，`sample (2).csv` などと名前をつける時に起こります。この `(1)` を削除してもその前に半角スペースが入ってたりするのです。まったく Windows は！

[*26] Windows は One-Drive にもデスクトップがあります。同じ環境をローカルと同時にクラウドにもあって常に同期していればいいのですが，必ずしもそうではないので (OneDrive の設定によります)，`C:\Users\Username\Desktop` と `C:\Users\Username\OneDrive\Desktop` のどちらに保存したのかが分かりにくくなっています。デスクトップに保存したのに，と言われてもどっちのデスクトップなのか，という問題が生じるわけです。さらに専修大学では 2023 年から仮想デスクトップ環境 (VDI) を使うようになりました。これで 3 つ目のデスクトップができたことになるわけです。なんてこった！まったく Windows は！

フォルダと同じ意味で**ディレクトリ**という言葉が使われることもあります。

C.7.1 相対パスと絶対パス

普段 PC を使っているときは気にすることがありませんが，R や RStudio などプログラミング言語をつかっているときは，「今どこで作業しているか」という現在地が重要になってきます。たとえば，RStudio で `C:\User\kosugitti\Document\kiso1\`というところでプロジェクトを開いているとします。スラッシュ (\) はフォルダ，コロン (:) はドライブを表す記号です。プロジェクトフォルダは，C ドライブの User フォルダの下にある，kosugitti フォルダの下にある，Document フォルダの下にある，kiso1 というフォルダということになります (プロジェクト名が `kiso1` だとそうなります。)。この kiso1 フォルダが現在地です。

このフォルダの中で，Rmd ファイルや R スクリプトファイルを使って，他のファイルを参照するようなコードを書くとしましょう。たとえば `script1.R` というファイルに `read_csv` 関数を書いたとします。読み込みたいファイルは，同じフォルダの中にある，`sample.csv` とだとします。このとき，`read_csv` の書き方は次のようになるでしょう。

code：C.1 相対パスで読み込む

```
1  dat <- read_csv("sample.csv")
```

しかし別の書き方もあります。たとえば code:C.2 のような書き方でも問題ありません。

code：C.2 絶対パスで読み込む

```
1  dat <- read_csv("C:\User\kosugitti\Document\kiso1\sample.csv")
```

後者 code:C.2 の書き方は，ファイルの場所を全部書いてありますから，確実にその場所が特定できます。それに比べて前者 code:C.1 の書き方は，なぜファイルを書いただけでいいのでしょうか。これは，このコードを実行している現在地と同じフォルダの中に `sample.csv` ファイルがあるからです。プログラムは，命令を受けるとファイルを探しにいきますが，現在地と同じフォルダの中を探すことになっているのです。この現在地，すなわち現在作業しているフォルダのことを**作業フォルダ (working directory)** と言います。

では作業フォルダと別のフォルダの中にファイルがあれば，アクセスできないのでしょうか。そんなことはありません。code:C.2 の書き方を使えば，作業フォルダがどこにあっても位置を特定できますから，作業フォルダを問わずに書くことができます。ちょっと長くて面倒ですが，確実にある場所を指定しているからです。この書き方のことを**絶対パス**による指定といいます。一方，code:C.1 の書き方は，今の作業フォルダから見た場所，という相対的な書き方になっています。この書き方のことを**相対パス**による指定，といいます。相対パス指定で，違うフォルダにアクセスする場合には，次のようにします (code:C.3)。ここでは，`Document` フォルダの中に，kiso1,kiso2 フォルダがあり，kiso1 フォルダの中で作業している時に kiso2 フォルダの `sample2.csv` ファイルを読み込む例を書いています。

code：C.3 絶対パスで読み込む

```
1  # 絶対パス指定
2  dat <- read_csv("C:\User\kosugitti\Document\kiso2\sample2.csv")
3  # 相対パス指定
4  dat <- read_csv("..\kiso2\sample2.csv")
```

絶対パスはそのままなのですが，相対パスは..\という記号になっていますね。このピリオドを 2 つ打つ方法で，「ひとつ上のレベルの」という意味になります。このように，現在地からの相対的な位置関係で，ファイルを指定することもできます。

絶対パスと相対パスのどちらが良いのか，というのは一概には決められません。絶対パスは，PC が変わったりフォルダの構造が変わったりすると役に立ちませんから，使い勝手が悪いと言えなくもないですが，確実に指定できる方法です。相対パスは，PC が変わったりしても「現在地から相対的に見てどこか」という話ですから，たとえばこの例で `kosugitti` フォルダごと別の場所に移しても (たとえば `D:\Univ\Classes\kosugitti\kiso1` のように)，コードはエラーなく動きます。kiso1 フォルダ，kiso2 フォルダの相対的な位置関係が変わらなければいいのですから。バックアップを取ったり，複数の環境で同期しながら作業する場合などは相対パスの方がいいでしょうね。

いずれにせよ，現在どこで作業しているかということ，すなわち作業フォルダの場所を，意外と意識しておかなければならないということには注意が必要です。ファイルをどこに置いたか，どんなファイルを置いたか，自分はどこにいるのか，これが変わってくると「ファイルが見つかりません」というエラーになるのです。言い方を変えると，**ファイルが見つかりませんエラーの原因は，この 3 つのどれかである**ことがほとんどです。

C.8 ファイルのバージョン管理

これからみなさんは大学生活の中で，たくさんのファイルを生み出していくことになるでしょう。たとえば 4 年生の時に卒業論文を書くことになりますが，データファイル，分析ファイル，図を書いたファイル，引用文献リストを書いたファイル，卒論本文などなど，1 つの研究でも複数のファイルが作られることはよくあります。さらに，これらのファイルは日々加工されますから，その度に上書き保存することになります。いわば，ファイルがバージョンアップしていくのです。

卒論などの場合はとくに，「途中で保存しておく」ことが重要です。途中まで書いていた時に，横に置いていたマグカップが倒れて PC にコーヒーがかかり，変な音を立てて PC が壊れてしまった，ということがあるかもしれませんからね。紙に書いていた時代は，その手のハプニングがあってもせいぜい原稿用紙数枚がダメになっただけで，「ちくしょう，やりなおしかぁ」で済んだのですが，電子データの場合は電子の藻屑になると復元させることができません。ですから**バックアップは非常に大事**なのです[*27]。

バックアップの基本は，「別の場所に」「別のファイル名で」というものです。同じ名前で上書きすると元に戻ることができませんから，面倒でもコツコツと違う名前をつけましょう。そうするとよくあるのが，`soturon1.docx`,`soturon2.docx`,`soturon3.docx`,`soturon3`(**修正**)`.docx`,`soturon3`(**最新**)`.docx`,`soturon3`(**最新**)(**修正**)`.docx`,`soturon3`(**最新**)(**修正**)(**提出版**)`.docx`,`soturon3`(**最新**)(**修正**)(**提出版** 2)`.docx`.... というようになっていくやつで，「どれが最新版だっけ...」と書いてる本人でも探すのに苦労することになります。

この問題の解決策として，`soturon1001.docx`，`soturon1005.docx` のように日付を入れるというものがあります。10 月 1 日分，10 月 5 日分，としていけば「いつまで戻れば良いか」もわかるのでいいやり方ですね。日付の数字をファイルの先頭につけておくと，並べ替えも簡単です。この日付をつけて保存するというのを習慣化し，1. 昨日までのファイルを開いて今日の日付で別名保存する，2. 作業を進めて，時々上書き保存，最後にも上書き保存，3.PC に USB メモリをさして，バックアップ保存して作業終了，というルーティンを作っておくと，確実に記録が残って良いでしょう。

ただし，この方法の問題は，ファイルサイズが大きくなりすぎることです。図表などを含めたファイルが数百 MB になることは少なくありません。それを次々複製していくわけですから，大容量の USB メモリでも限界が来るかもしれません。これは「丸ごとコピー」していることが原因で，たとえば昨日は 10 行目まで書いた，今

[*27] ちなみに私の経験上，レポートが電子の藻屑になったので助けてください！と言われることがよくあり，4 年間の学生生活の間では平均 10–15 名に 1 人の割合で発生することのようです。

日は 11 行目から 14 行目まで書いた，というのであれば，この増えた 4 行分 (差分) だけを追加保存すればいいのに･･･と思いませんか。

こうしたバックアップやバージョン管理をやってくれる仕組みとして，Git というものがあります。Git は作業フォルダの中身の差分だけを記録し，必要であれば過去のバージョンに戻すこともできるシステムです。毎回上書き保存 (commit する，といいます) のたびに「どこを変更したか」というメモを付けて保存しておけば，そのメモを見ながら「ここの時点まで戻ろう」という使い方をすることができます。ファイル名は変更する必要なく，同じファイル名で進めていけますから，同じようなファイルがたくさんあって訳がわからなくなるということもありません。さらに保存先をクラウドにした GitHub というものもあり，これを使うとクラウド上に追記していくことができます。この GitHub は IT 企業などでプログラムをチームで進めていく時にも使われている技術で，全体のプログラムに個別の機能を複数人が追加，管理者が必要なものだけ取り入れる，というように使われています。国里ゼミや小杉ゼミでは，卒論を GitHub で管理し，学生が書いた分を commit し，それを hub 上にアップロード (push といいます) する，というようにします。教員の方は学生の進捗が管理できますし，どこがどう変わったかが分かりやすく，バージョン管理と同時にバックアップもできるという便利な仕組みです。GitHub は無料でアカウントを作ることができますから，興味があれば皆さんもチャレンジしてみてください[*28]。

さてここで，ひとつ注意をしておきます。卒論の原稿やプログラムは日々変化するものですからバージョン管理が必要ですが，データファイルはアップデートする必要がありません。いや，アップデートしてはおかしいのです。たとえば 100 人分のデータを取って分析をしていて，後で「やっぱりこのデータを削ろう」というのは，研究不正が疑われかねません。自分に都合の良いデータだけで議論し，都合の悪いデータは削除して統計的に有意な結果が出るように細工しよう，なんてことがあれば困るのです。データファイルはバージョンアップせず，それを加工，計算するプログラムがバージョンアップしていく。そしてその加工プロセスは誰でも見ることができるように公開されている。少なくとも，科学的な営みをする上では，そうしたやり方が必要なのです。自分だけのデータで自分だけの分析方法で，良い結果だけ示すというのは適切な方法ではありません。

Open Science Framework(https://osf.io/) はこうした「オープンな科学」にむけた取り組みの一種です。このアカウントは誰でも無料で作ることができ，ここにファイルをアップロードしたり，分析計画を事前に記録しておくことができます。何も難しいことではなくて，クラウド上のファイル置き場だ，というぐらいに思っていただければ結構です。ここにおかれたファイルは自動的にバージョン管理され，同じファイル名のものがアップデートされてもその記録 (ログ) が残ります。最近はこの OSF をつかって，論文化されたデータやプログラムを公開するという取り組みも進んでいます。

クラウド，バックアップ，オープンサイエンスといった新しい研究方法は日々生まれてきています。皆さんも便利な機能はどんどんキャッチアップしていきましょう！

C.9 おわりに

古臭い話をしてしまうのは，私が歳をとった証拠でしょうか。皆さんはこんな話を知らなくても，スマホやタブレットを使いこなしていることと思います。細かいことを知らなくても，ユーザとして利用するだけなら知らなくて良い話なのかもしれません。私はここにも書いたように，高校生の頃からコンピュータの発展と一緒に大人になってきましたから，学ぶともなく学んできたところがあります。皆さんは生まれた頃からコンピュータやがあったネイティブ・デジタル・カウボーイですから，苦労なんかする必要なかったわけです。

[*28] このテキストやシラバスも GitHub で管理していますし，公開されているサイトも GitHub 上のものです。これからは教科書も日々成長していくものになるかもしれません。

しかし細かい仕組みを知らないということは，問題が生じた時に「何か・誰かが，どこかでどうにかなって，今私が困っている」という状況になります。問題を特定できないと，解決することもできません。コンピュータは文房具に過ぎませんから，それを使いこなせないほうが格好悪いのです。しかも今後ますますコンピュータに囲まれた世界になっていくのは自明ですから，ここに学習コストをかけない方が勿体無い。幸い，わからないことに対して，自ら調べて学んだ利する時間と環境が用意されているのが大学という世界なのですから，今のうちにしっかり基礎固めをしておきましょう。

ここのくだらない懐古的エッセイのような文章が，何かの足しになれば幸いです。

付録 D

ギリシア文字一覧

ギリシア文字ってかっこいいけど，読み方わからない･･･という人のための一覧。ついでに TEX 表記も。

表 D.1　ギリシア文字とその読み方

読み方	大文字	小文字	英語表記	TEX 表記
アルファ	A	α	alpha	`\alpha`
ベータ	B	β	beta	`\beta`
ガンマ	Γ	γ	gamma	`\gamma`
デルタ	Δ	δ	delta	`\delta`
エプシロン	E	ε	epsilon	`\varepsilon`
ゼータ	Z	ζ	zeta	`\zeta`
イータ	H	η	eta	`\eta`
シータ	Θ	θ	theta	`\theta`
イオタ	I	ι	iota	`\iota`
カッパ	K	κ	kappa	`\kappa`
ラムダ	Λ	λ	lambda	`\lambda`
ミュー	M	μ	mu	`\mu`
ニュー	N	ν	nu	`\nu`
クサイ	Ξ	ξ	xi	`\xi`
オミクロン	O	o	omicron	`\mathrm{o}`
パイ	Π	π	pi	`\pi`
ロー	R	ρ	rho	`\rho`
シグマ	Σ	σ	sigma	`\sigma`
タウ	T	τ	tau	`\tau`
ウプシロン	U	υ	upsilon	`\upsilon`
ファイ	Φ	ϕ	phi	`\phi`
カイ	X	χ	chi	`\chi`
プサイ	Ψ	ψ	psi	`\psi`
オメガ	Ω	ω	omega	`\omega`

付録 E

記号の入力とキーボードの場所

プログラミングのミスでよくあるのが打ち間違い，スペルミスです。X と x，S と s など大文字と小文字で形が同じものや，l(エルの小文字) と 1(数字のイチ) の違いなどは，プログラミング用フォントにして違いがわかるようにするとか，文字列の意味から類推する (Normal とあればノーマルであって，ノーマ・イチではないと察する) など工夫が必要かもしれません。

また，理由はよくわからないのですが頻発するスペルミスは，データ (data) をデート (date) と書いてしまうことです。データはラテン語の datum(与えられたもの) の複数形なのですが，最近のデータサイエンスの文脈では data も単数形と考えるようです。ともかく，日付を表す date とは由来も意味もスペルも全て違うので，気をつけましょう。

さて，これらはまだ序の口。プログラミングのコードを読んでも，日本語の五十音に入っていない記号の違いがわからない，どこでそれが入力できるかわからない，質問しようにも読み方がわからないといったものもあります。ここではこれらをまとめて解説します。記号の上では微妙な違いですが，当然形が違うのでプログラミング上は違う文字として扱われますので，形の細部までよくみてください。なお，プログラミングでつ変わる時は言語に依存することもありますので，ご注意ください[*1]。

特に目立つのはハイフンとチルダ，アンダースコアの入力ミスです。それぞれ文字を書く領域における位置が違ったり，形が違ったりするのでよくみてください。

ハイフンは真ん中　A-B
アンダースコアは下　A_B
オーバーバーは上　A¯B
チルダはニョロ　A~B

これを踏まえて，そのほかの記号の名称や意味を表 E.1 で確認しましょう。

一般的な日本語キー配列の場合，1 つのキーに 4 つの文字・記号が割り当てられていますが，英語入力モードの場合はキーの左側，日本語入力モードはキーの右側を見ることになります。キーを押すと下の段の文字が入力されますが，シフトキーを押しながらキーを押すことで上の段の文字が入力されることになります (図 E.1)。

これを踏まえて，日本語キーについては E.2，US キーについては図 E.3 に代表的な記号とキー配列の位置を示しました。入力に困った場合は一度図を見て確認してください[*2]。

[*1] たとえばコメントアウトは C 言語では\\，R では#，TeX では % など，それぞれ異なります。
[*2] US キー配列はキートップに一種類の文字しかなく，美しい配置なのでおすすめです。

表 E.1 記号と読み方

記号	読み方	解説
:	コロン	英文中では「すなわち」などの意味。セミコロンと間違えないように
;	セミコロン	英文中では文章の区切り，接続詞のようにつかう
.	ピリオド	英文の終わりを意味する。日本語で言う句点
,	カンマ	英文の区切りを意味する。日本語で言う読点
@	アットマーク	メールアドレスに用いられることで有名
$	ドルマーク	米国の通貨単位。R では変数名指定のときにもちいる
/	スラッシュ	割り算の記号
*	アスタリスク	掛け算の記号
+	プラス	足し算の記号
-	マイナス	引き算の記号
^	ハット	累乗の計算の記号
=	イコール	等号。プログラミングでは==で一致しているかどうかの判定にも
!	エクスクラメーション	強調。プログラミングでは否定 (NOT) の意味になることも
_	アンダースコア，アンダーバー	位置に注意。ハイフンではなく文字領域の下の線 変数名をつなげる時 (ex. snake_case) に使ったりする
~	チルダ	R では独立変数と従属変数をつなぐときに使う ハイフンやオーバーライン，アンダースコアと間違えられる率高め
‾	オーバーライン，オーバーバー	アンダースコアの逆。滅多に使わないが。文字化けを直したときにみられる。
%	パーセント	プログラミングではコメントアウトの時などに使われたりする
&	アンパサンド	プログラミングにおける AND(論理積) の記号など
|	縦棒	プログラミングにおける OR(論理和)，条件付き確率の記号にも
\	バックスラッシュ	プログラミングではコメントアウトの時などに使われたりする
"	ダブルクォーテーション	文字列の開始･終了を表す。同じ記号で閉じる
'	シングルクォーテーション	文字列の開始･終了を表す。同じ記号で閉じる
`	バッククォーテーション	文字列の開始･終了を表す。同じ記号で閉じる
[]	大括弧	プログラミングでは配列を意味することがある
{}	中括弧	プログラミングではブロックの開始と終了を意味することがある
()	小括弧	数式のまとまりを表す

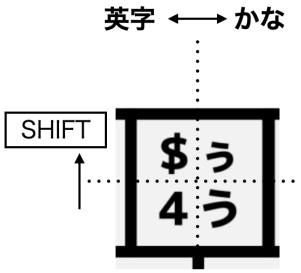

図 E.1 日本語キーで入力する場合

図 E.2　代表的な記号と日本語キー配列

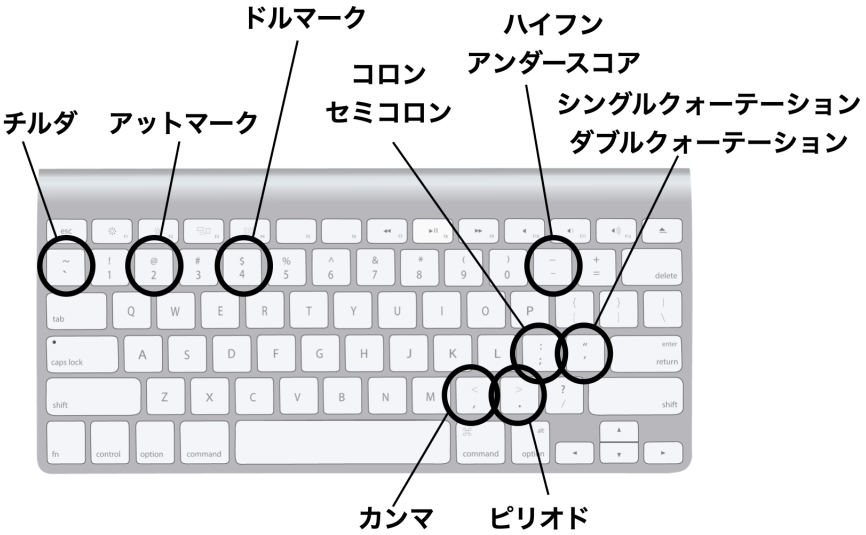

図 E.3　代表的な記号と US キー配列

付録 F

本講義に対応する詳細シラバス

F.1　イントロダクション

F.1.1　授業内容

科目の中でのこのコマの位置づけ

この講義の位置付けは，基礎的な心理統計の学習は終わった後の応用的内容となる。基礎的な内容として，確率の基本的な考え方，線形モデル (回帰分析，群間の平均値差の検討)，さまざまな推定法による母数の推定と検定の考え方を理解しているものとする。これに基づいての応用であるから，扱うデータも単変量ではなく多変量であるし，数学的には行列表現を用いることになる。これらを使って，回帰分析や因子分析の数理的理解を目指す。このコマではこの講義によって扱われる領域を外観するとともに，基礎的な内容で扱ったものがしっかりと定着しているかどうかを確認することを目的とする。

コマ主題細目

正規線形モデルの世界　単変量ではなく多変量を扱う統計の領域に入るので，多変量データとはどのようなものであるかに言及した上で，本講義の扱う領域を概観する。心理統計の応用的分野では，正規分布を仮定した線形モデルがその大半を占めている。正規線形モデルに含まれるさまざまな下位モデルの名称を紹介するとともに，構造方程式モデリングに統合されることや，非線形なモデルとの違いについて理解する。加えて正規分布ではない分布を扱うモデルも増えてきている昨今，これらについてのモデリングアプローチの存在についても講義する。

→ 正規線形モデルの枠組みについては，三中 (2018) 参照。

尺度の四水準　心理統計の基礎で触れたが，データ化として扱う数値はその尺度水準によってどのような計算が可能かということに違いが生じる。このことは，そのまま分析モデルや名称の違いに繋がるため，改めて名義，順序，間隔，比率の 4 水準を確認しておく。

→ Stevens (1946) の論文は短く，ネットで読むこともできる。入門書としては川端･荘島 (2014) の Pp.9–16，あるいは山田･村井 (2004) の Pp.22-25.

平均と分散　間隔尺度水準以上の数字であれば，平均値や分散，標準偏差によってその特徴を要約できる。ここではこれらの代表値の表記について，数学的記号とともに確認する。加えて，分散式を展開して表現したものや，分散がデータから得られる情報の上限であることを確認する。

共分散と相関係数　共分散やそれを標準化した相関係数は，複数の変数間関係を表現する最もシンプルな

ものの 1 つである。ここではこれらの複数の変数間に関わる代表値について，数学的記号とともに確認する。加えて，この他の関係の表現方法として，距離や共頻度などの共変量について解説し，それらの違いに応じて統計モデルが変わりうることを確認する。

→ 記述統計量については川端･荘島 (2014) の Pp.26-33 など基礎的な心理統計の教科書を参照すると良い。

キーワード

- 正規線形モデル
- 尺度水準
- 平均と分散
- 共分散と標準偏差

F.1.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 心理統計の基礎について，今一度基礎的なテキストを参照しながら，自分の理解度を再確認しておくと良い。とくに尺度水準や記述統計量の計算方法などは今後この講義でも頻出するので，確認しておく必要がある。

■復習 数式の展開を踏まえて理解しておくとともに，実際のデータを使ってて計算しながら確認すると良い。とくに分散は二乗のオーダーになるので元の単位に比べて大きな数字になること，標準化のプロセスや相関係数の大きさなど，逐一確認しておくべきである。

F.2 心理尺度を作る

F.2.1 授業内容

科目の中でのこのコマの位置づけ

目に見えないものを測定するために心理学が洗練してきた手法が，心理尺度である。心理尺度の作成方法としてサーストン法，リッカート法，SD 法などがあり，心理学の初頭コースで習うものも少なくないが，その本質は反応カテゴリに数値を割り当てる，というところにある。「そう思わない」「ややそう思わない」などといったカテゴリーに対する反応が，なぜ 5 や 4 といった数字にすることが許されるのか。カテゴリカルな反応が連続的な量として扱うことができる理由などについて，よく知られていない現実がある。この点をしっかり理解しないまま進んだ分析を行うと，結果の解釈はもちろんそもそもの研究が足元から崩壊することにもなりかねない。本講義ではこの点について，作成方法から数値化まで一通り確認し，最後に心理尺度の評価基準である信頼性と妥当性について理解する。

コマ主題細目

サーストンの等現間隔法 サーストン法と呼ばれる尺度構成法は，態度とよばれる心理学的特性を仮定している。この態度は対象，符号，強度をもち，正規分布すると仮定されている。個々人の態度を測定す

るために，事前に評定者集団を用意して項目を採点しておく必要がある。そこで評定値に等間隔性を持たせる工夫をしているため，態度の数値化ができるという原理を理解する。

→ サーストン法による尺度作成については末永 (1987) の Pp.149–152 参照。

リッカートのシグマ法　リッカート法は最もよく使われるスタイルの心理尺度である。カテゴリーに無頓着に数字を割り振る慣例がみられるが，本来は潜在的態度が正規分布することを想定し，確率分布の確率点を得点とする方法であった。このような数値化がされているからこそ，順序尺度ではなく間隔尺度水準と「見なす」ことが許されてきているのである。この原理を理解しておくことは，後のより進んだ尺度作成法を理解する助けになる。

→ リッカート法による尺度作成については，宮谷他 (2009) の Pp.150–153 を参照。ただしシグマ法についての言及はなく，田中 (1977) などの古典を当たらねばならない。

心理尺度の問題点　ここまで見てきたように，心理尺度の基本は社会的態度に関する測定であった。社会的態度や性格特性にたいして正規分布することを仮定するから数値化が可能なのであり，単に 5-7 段階の目盛を割り振ったものを尺度と呼ぶものではない。心理尺度の乱用とその問題点に言及し，心理尺度を作ることができる限界についての理解を深める。

キーワード

- サーストンの等現間隔法
- リッカートのシグマ法
- 信頼性と妥当性
- 心理尺度の限界

F.2.2　授業情報

■コマの展開方法　講義

予習・復習課題

■予習　基礎実習で尺度作成法や尺度の分析をしたことがあれば，その時の資料を再確認しておく。とくに反応カテゴリをどのように採点したか，また尺度の評価どのように行ったかを確認する。とくに尺度作成の経験がない場合は，関連書籍宮谷他 (2009) を参考に方法論を予習しておくことが望ましい。

■復習　IT 相関やアルファ係数は統計環境 R で簡単に計算できる。とくに psych パッケージにはこれらの関数がすでに準備されている。サンプルデータを使ってこれらを計算してみよう。

F.3　テスト理論と因子分析

F.3.1　授業内容

科目の中でのこのコマの位置づけ

目に見えないものを測定するという意味で，テスト理論は心理学の測定と関係が深い。前回は社会心理学における態度の測定を前提に議論したが，測定に関してはテスト理論で一般的に議論できる。

真のスコアと誤差とに分解すること，誤差の基本的な仮定を確認した上で，古典的テスト理論を項目と被験者の特性に分割することで因子分析モデルに展開されるところを見る。また，多因子モデルに拡張した上で，その数理的展開から，信頼性と妥当性に言及できることを確認する。数式の展開は代数の基本的な特徴を確認すれば問題なくフォローできるはずである。

コマ主題細目

古典的テスト理論と信頼性の導出 古典的テスト理論についての復習である。その基本モデルについて触れ，その平均値と分散が意味するところから測定モデルの意味するところ (誤差が相殺しあうこと) と信頼性の定義が導出できることを改めて確認しておく。

→ 心理尺度の信頼性については，末永 (1987) の Pp.156–158，妥当性については Grimm・Yarnold (2001 小杉他 訳 2016) の第 4 章も参照。

因子分析モデル 因子分析モデルは，古典的テスト理論のモデルを拡張したものである。まずは単因子モデルを例に，項目特性と被験者特性が分離されたことを確認する。その上で，性格検査や知能検査などの歴史に触れながら，多因子モデルについて解説する。多因子モデルを例に記号や添字を確認しておく。

→ 小杉 (2018) の Pp.173–177

因子分析の第 2 定理 因子分析モデルを展開することで，相関係数が因子負荷量の積和で表現できること，因子得点がその仮定から計算上消えることを確認する。得られた指揮は因子分析の第二定理と呼ばれ，妥当性に関する議論がここから導かれることをみる。

因子分析の第 1 定理 ある項目自身の相関係数を考えることで，因子分析の第一定理にたどり着く。ここで共通因子の二乗和を共通性と呼ぶことにすると，信頼性の考え方が項目レベルで行われるように発展したことが確認できる。

→ 小杉 (2018) の Pp.173–177

キーワード

- 古典的テスト理論
- 因子分析法
- 因子分析の定理

F.3.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 一年時に信頼性・妥当性について，あるいは古典的テスト理論について学んだことを復習し，どのような概念であったかを再確認しておくことが望ましい。

■復習 テスト理論と因子分析モデルの関係について，因子分析モデルはどこが新しく何を改定しようとしたのかについて，自分なりの言葉で説明できるようになろう。

F.4 現代テスト理論

F.4.1 授業内容

科目の中でのこのコマの位置づけ

テストの理論も目に見えないものを測定するという意味では，心理学と同じモデルを実践する領域である。心理学的尺度作成法の発展には，テスト業界における理論的展開の位置付けを知ることが役に立つ。

因子分析によって項目と被験者の特徴を分離して考えることができるようになった。ここで学力テストに目を向けると，単因子でよいことと従属変数がバイナリになっていることがわかる。

この特殊な測定法についてのモデルを考えるために，まずは通過率の概念を導入したうえで，累積正規分布とその近似としてのロジスティック曲線，および 1,2,3PL モデルを紹介する。これらのテストは新しいテスト理論とよばれるが，それはこれまでのテスト理論に含まれていた集団に依存した測定であったこと，完全データに限定されていたことなどを乗り越えられるからである。もちろんテストの等価がしやすいという側面もある。

テスト理論の展開としての項目反応理論と，因子分析モデルとの相同性を強調することで，見えないものを測定しようとするアプローチという意味では同じであったことを確認する。

コマ主題細目

因子分析とテスト理論　単因子モデルの特殊事例として，学力テストの例を考える。学力テストの性質から，因子構造よりも因子得点に注目するという強調点の違いはあるが，因子分析モデルの一環として捉えることを強調する。

→ 高橋 (2002) は最も平易なテスト及び現代テスト理論への入門書である。最初の数ページだけでも参考になる。

通過率と累積正規分布　学力テストの分析例として，通過率の計算から累積正規分布へとつなげる。累積正規分布をそのまま確率モデルに繋げてもよいが，ロジスティック曲線を使う方が関数の形が簡単であり，こちらの方が実際には使い勝手が良い。ベルヌーイ分布を用いた線形回帰モデルの文脈で考えれば，ロジスティック回帰分析をしていることでもあることに言及する。

→ 通過率については豊田 (2012) の Pp.1–8 を，ロジスティック関数と累積正規分布の関係については加藤他 (2014) の Pp.81–83 を参照

項目母数の特徴　ロジスティック曲線を導入することで，関数の変形がたやすくなった。ここでは 1PL,2PL ロジスティックモデルを導入し，どの項目母数が関数の位置や形をどのように変えるか，そしてそれが意味するところを理解する。モデル的には 5 母数モデルまで考えられるが，実際にはせいぜい 3PL モデルである。この講義では後の因子分析との対応関係も考えるため，2PL モデルまでの紹介に留める。

→ 項目母数については豊田 (2012) の Pp.31–34，加藤他 (2014) の Pp.71–80 が参考になる。

被験者母数の特徴　項目母数が明らかになった状況に置いて，どのように被験者母数を推定するかを考える。ここで ICC から逆算的に被験者母数の位置がどこにあるか，該当領域を絞り込んでいく尤度関数を視覚的に確認する。この方法を使うと，すべての項目についての回答が得られていないと推定で

きないといった不便がなく，また被験者母数の位置によっては ICC がそれほど有用な情報を与えてくれないこともある。これらの点は，完全情報最尤推定や情報関数にもつながるため，しっかりと理解しておくことが必要である。

→ 被験者母数の絞り込みについては，小杉・清水 (2014) の Pp.171–172. が参考になる。

キーワード

- 通過率
- ロジスティックモデル
- 被験者母数の推定について

F.4.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 テストの前提となる標準正規分布について復習しておく。とくに R を使って出力できる確率密度，確率点，累積確率など手を動かして予習しておくと良い。

■復習 適当なグラフ描画ツール (R でよい) をつかって，ロジスティックモデルを描写し，項目母数をどのように変えるとどのように曲線の形が変わるかを確認してみよう。

F.5 現代テスト理論その 2

F.5.1 授業内容

科目の中でのこのコマの位置づけ

項目反応理論の数学的特徴を踏まえることと，現代的尺度構成法の理論的基礎を学ぶ。

項目反応理論の導入によって，被験者母数と項目母数が完全に分離され，項目の特徴を細かく記述できるようになった。また，項目の特徴がわかっていればテストの実践方法も変わってくる。ひとつは CAT に代表されるように，ダイナミックに出題を変化させることができるようになること，そうしたうえでもテストの平均点が事前にコントロールしうることなどが示される。項目から得られる情報という観点から項目情報曲線が，項目情報曲線の累積からテスト情報曲線が導出される。

つづいてこのテスト理論の発展形として，多段階モデルに拡張可能なことをみる。とくに段階反応モデルは，リッカートのシグマ法のように段階反応をモデルかできるという意味で，現代的リッカート法であるともいえる。段階反応モデルを用いることで，適切な反応段階のチェックをできるなど，応用的側面が高いことを確認する。

またテスト理論は因子分析の特殊系であるという扱いだったが，多段階，多因子へと展開することで再び因子分析モデルに統合されていくことを確認する。

コマ主題細目

現代テスト理論の特徴 現代テスト理論の特徴は，項目母数と被験者母数の分離，完全情報最尤推定，項目情報曲線による信頼性の表現，項目プールがあれば事前にテストの平均点を設計できることがあ

げあられる。また Computer Adopted Test の形式を用いることでテストのあり方そのものも変わってしまう。ただし実際には，膨大な項目プールが必要であること，事前に項目母数を準備しておく必要があること，その他「公平性のために新しいテストでなければならない」という信念などが弊害となって実践的には敷居が高いことなどを解説する。

→ 古典的テスト理論との比較については，加藤他 (2014) の Pp.67–69，あるいは豊田 (2012) の前書きが十分に詳しい。

段階反応モデル　テスト理論はバイナリデータに対する分析だが，多段階の反応に拡張する方法がいくつか考えられている。1 つは段階反応モデルとよばれるもので，これを使うと適当な反応段階数がデザインできるなど利点は大きい。またその考え方はリッカートのシグマ法を洗練したものであるとも言え，せめてこうした方法を使わないと多段階反応を適当に分析できていない。統計パッケージなどの実装も進んでいるので，計算コストはほとんど障壁にならない。また，ポリコリック相関係数を用いた因子分析を実行すると，段階反応モデルのパラメータに変換できることから，因子分析とテスト理論が同じものであったことを再確認できる。

→ 豊田 (2012) の Pp.155–172 が詳しい。

因子分析の歴史と展開　因子分析モデルもテスト理論も潜在変数モデルとしては同じであり，一方が単因子・二段階，他方が多因子・多段階であることが道を分つ。またその性質から，一方が因子得点に，他方が因子構造に着目するため，テストの構成についての考え方が異なることにも注意する。繰り返しになるが，統計パッケージ上の実装は進んでいるので，どちらを使うにしてもとくに苦労することなく，積極的にカテゴリ軽モデルを推進していくべきである。

キーワード

- 項目情報曲線，テスト情報曲線
- 段階反応モデル
- 因子分析モデルとテスト理論

F.5.2 授業情報

■コマの展開方法　講義

予習・復習課題

■予習　項目反応理論，とくに 2PL モデルによる因子得点の算出方法を確認しておくと同時に，心理尺度ではどのように尺度値を定めていたかについて復習しておく。

■復習　信頼性についての考え方が，古典的テスト理論，因子分析論，現代テスト理論を通じてどのように変わってきたかを確認しておこう。

F.6 行列計算の基礎

F.6.1 授業内容

科目の中でのこのコマの位置づけ

テスト理論や因子分析モデルの展開を理解した上で，さらに次のステップに進むためには，より数学的な構造の理解が必要である。ここまで因子分析モデルでは，因子得点をどのように算出するかが論じられていない。また相関行列を分解して因子負荷量を算出するにあたっても，どのように計算するかについては言及されてこなかった。これらの点を理解するための道具となるのが線形代数である。具体的には，行列の固有値分解を通じた解釈をすることで，因子分析，回帰分析など多変量データの方程式モデルを統一的に表現・理解できるようになる。そのための道具立てとして，線形代数の基礎知識を習得する必要がある。本講はこのより進んだ理解に向かうための，新しい数学ツールの導入を行う。

線形代数は方程式を簡便的に表現するための表現法であり，行列の観点から新たに四則演算を定義し直すことで一般的な表現が可能になることを示す。

コマ主題細目

行列とベクトル 多変量データを行列とベクトルで表現することをみる。学ぶべき用語として，スカラー，縦ベクトル，横ベクトル，行列，正方行列，対称行列，対角行列，単位行列をあげる。

行列の四則演算 ベクトルとベクトルの和，行列と行列の和，スカラーとベクトルの積，スカラーと行列の積，縦ベクトルと横ベクトルの積，横ベクトルと縦ベクトルの積，行列と行列の積をみる。とくにサイズが変わることに注意が必要である。

行列による便利な表現 連立方程式が行列で表現できることを見る。

逆行列と連立方程式 行列の割り算に当たるのが逆行列である。逆行列は存在しないこともあるが，もし適当なものが見つかればそれは連立方程式の解を一気に計算ができることになる。

キーワード

- ベクトル，スカラー，行列
- 行列の四則演算
- 連立方程式

F.6.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 とくに予習の必要は感じないが，授業に参加するにあたってはノートの準備が必要である。

■復習 計算方法に慣れておく必要があるので，練習問題を繰り返して行うことで，とくに行列の積の計算ができるようになっておく。線形代数の入門書としては，数学のテキストとして読みづらさを感じるかもしれないが，村上他 (2016) がよく，一冊手元に置いて演習をしながら進めると良い。

F.7 行列による関係の表現

F.7.1 授業内容

科目の中でのこのコマの位置づけ

線形代数についての基礎的なルールを習得する段階である。今回はより実践的・具体的に，データ行列をどのように線形代数で表現できるかを考える。データ行列から分散共分散行列，相関行列へと形を変えることを学ぶ。つづいて線形モデル，とくに従属変数が明確な回帰分析モデルを行列で表現することを見，線形モデルとデザイン行列について考える。さらに因子分析モデルを行列で表現することを考える。行列で表現することで，1 つの式の中に第一，第二定理の両方を含んだ形で表現できることを理解する。

コマ主題細目

データの行列表現 実際に手にするデータセットは，表計算ソフトウェアの画面で見る行列形式の数列であるが，これを記号で表現することで一般的に扱うことができるようになる。添字に気をつけながら要素ごとの表示をすることに加え，行列の計算をこのデータ行列に与えることによって，変数の平均や変数ごとの平均偏差を持った行列が表現できる。平均偏差行列を用いると，行列の積の特徴から分散と共分散を含んだ正方行列が作られることがわかる。また，データを標準化することで，標準化された行列の積が相関行列を表すことになる。このように一般的に表現するために，これまでの行列計算の方法が作られたのだと逆算的に理解すること，加えて行列のサイズに注目しながら，扱うデータの大きさがイメージできるようになることが肝要である。

→ 岡太 (2008) の Pp.77-110

線形モデル 行列表現の利便性は，データの変換だけにあるのではなく，統計モデルを表現する際にも生きてくる。基礎で学ぶ線形モデルは，基本的にエレメントワイズな表記法であったが，行列を使うことで単回帰も重回帰も同じ式で表現できることがわかる。このように表記の統一性があることが，線形代数の利点である。また統一的な表記にするために，切片項にかかる列を追加するなどの工夫をすることにも注意する。これらの点は，R など統計ソフトウェアを扱う上でもヒントになることが多い。

デザイン行列 基礎の段階で行った帰無仮説検定は，説明変数が離散変数であったことから，線形モデルの特殊形に過ぎなかったことを再確認する。その上で，先の回帰分析を行列表記にしたように，離散変数で説明する時の係数にかかる行列の形を確認する。この行列はとくにデザイン行列と呼ばれること，また自由度の関係から制約を加えた表現になるが，それがデザイン行列の中でどのように書き表されるかを確認する。

→ Dobson (2008 田中他 訳 2021) の Pp.41–45 にごく簡単な紹介が，豊田 (2000) の Pp.47–62 には計画行列として構造方程式の枠組みで説明されている。

因子分析モデルの行列表現 因子分析モデルはここまでエレメントワイズで表現されていたが，同様に行列表現にするとどのようになるかを確認する。とくに行列のサイズに注目することが重要である。というのも，統計ソフトウェアを使っていると因子得点が表示されないことが少なくないが，行列の形で見ると因子負荷量は項目数 × 因子数，因子得点は回答者数 × 因子数になることがより意識されやすいからである。他にも因子分析に関する特徴量が行列のどの要素にはいっているか，また因子分析の定理が行列のどこで表現されているかを確認することが重要である。

因子分析の行列的表現については → 芝 (1979) が良書だが，現在は絶版。同様の内容は小杉 (2018) にもある。

キーワード

- データの行列表現
- 分散共分散行列，相関行列
- デザイン行列

F.7.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 行列の掛け算がメインになってくるので，計算方法並びに計算結果のサイズを確認する方法を見ておこう。

■復習 行列表現によって重回帰方程式が 1 つの形になることを確認する。平均値の差を見るために線形モデルが用いられることを確認する。また因子分析モデルを行列表現すると，一気に 2 つの定理が 1 つの式で表現できることを確認する。

F.8 固有値と固有ベクトルと因子分析モデルの関係

F.8.1 授業内容

科目の中でのこのコマの位置づけ

因子分析モデルを行列表現することで，いよいよ因子をどのように算出しているのかについての答えが明らかになる。

因子負荷量を算出するためには，線形代数でいうところの固有値についての理解が必要である。まずは固有値と固有ベクトルを導入し，どのように計算するかを見る。とくに固有ベクトルはノルムが定まらないことを確認する。そこから，固有値と固有ベクトルがどのような性質を持っているかを幾何学的観点から確認する。正方行列が座標変換を行うためのものであると考えれば，固有ベクトルは変換行列の基底となることがわかるだろう。データ解析にあたって，相関行列の基底を求めるとはどういうことかをイメージするだけでも，因子分析の理解がまた一歩深まるだろう。

コマ主題細目

固有値と固有ベクトル 行列の固有値と固有ベクトルの性質を理解する。直感的には，正方行列がスカラーに変わることが，情報圧縮になっていると言えるだろう。また，行列のサイズと同じ数だけ固有値が見つかること，固有値の総和が元の行列のトレース trace になることを確認する。とくにデータ解析の領域では，分散共分散行列か相関行列が分析対象になることが基本であり，こうした対称行列の固有値は実数になること，相関行列のトレースは項目数と合致することを改めて確認することで，データの情報圧縮になることについての直感的理解をめざす。

固有ベクトルを求める 2×2 行列を例に，固有値と固有ベクトルを求める計算を行う。固有方程式を導入

し固有値の計算を行うことは比較的簡単であるが，固有ベクトルの求め方が直感的にはわかりにくい。というのも，固有ベクトルはその大きさが定まっておらず，要素同士の相対的な大きさを示すだけだからである。ここでベクトルのノルムを導入して標準化解を算出することを確認する。また行列のサイズが大きくなると方程式が高次になるため，一般解が得られないこと，結果的に近似解を求める計算方法が開発されていることをみる。

固有値と固有ベクトルの幾何学的意味　正方行列は一次変換行列であり，固有ベクトルはその基底であることを単純な行列から理解する。固有ベクトルはノルムが定まっていないこと，すなわち方向性だけを持ったものであることを理解する。また固有値はその総和が元の行列のトレースと一致することから，分散あるいは項目数 (相関行列の対角) を組み替えたものであり，固有値の大きさの順に考えることはすなわち，より明確な次元を抽出したことになることを確認する。

→ これについては平岡・堀 (2004) にアニメーション付きで説明されているのがわかりやすい。また長沼 (2011) は固有値の章だけでなく，付録を読むとまた固有値と固有ベクトルの多角的な理解が進む。

因子分析モデルの意味　因子分析モデルは相関行列を固有値分解することであり，それはすなわち相関行列の中にある基本的な次元・座標を求めることにある。すなわち複数人の反応パターンの共通要素を取り出すということであり，これは心理学的アプローチをほぼ直接的に数学表現したものであることを理解する。座標の回転についても触れ，仮定を緩めた場合の表現も理解する。

キーワード

- 因子分析モデルの行列表現
- 固有値
- 固有ベクトル
- 固有ベクトルの幾何学的理解

F.8.2 授業情報

■コマの展開方法　講義

予習・復習課題

■予習　因子分析の基本モデル，第一・第二定理の導出を復習しておこう。

■復習　因子分析モデルが何をやっているかを考えた上で，心理学における尺度の利用やその解釈においてどのような注意をしなければならないかを言語化してみよう。

F.9 R をつかっての行列計算

F.9.1 授業内容

科目の中でのこのコマの位置づけ

行列の計算は単純な計算ではあるが，要素の数が多くなるので反復回数が増え，また計算の法則も慣れるまでは難しい。人間にとってはミスが多くなりがちなこの計算が，計算機 (コンピュータ) は最も得意とする

ところである。計算機は疲れることなく，単純な反復計算を瞬時にこなす。多変量データ解析は計算機の発展の歴史ともにあり，昨今の計算機パワーは非常に複雑な統計解析も瞬時に答えを出すようになった。

この行列計算は表計算ソフトにはできないことであり，統計環境 R のような，統計パッケージを利用することになる。本項では，統計環境 R を用いて行列の基本的な計算を演習によって習得することを目的としている。また R で行列の計算ができることは重要ではあるが，実際に統計分析をする時にはより便利なパッケージを利用することになる。心理学関係の数値計算については，psych パッケージが便利である。これを導入し，記述統計量や信頼性係数など基本的な分析が便利になることを確認する。

コマ主題細目

R による行列計算 R についての基本的な使い方 (環境の準備，RStudio によるプロジェクト管理，パッケージの導入，基本的な四則演算等) については習得済みであることを前提とする。行列計算にあたっては，データをマトリックス型で保持している必要があり，また行列の計算は四則演算と異なること，ベクトルの長さが時には再利用されることなど注意が必要な点がある。それらを踏まえて，データの方を考えながら行列の四則演算を確認する。

R によるデータの変換 R の行列計算を使って，前時までに行った raw data の変換計算，すなわち平均，平均偏差行列，分散共分散行列，相関行列などの計算プロセスを確認する。また，cov や cor 関数を使うとこれらが一気に計算されるが，分散の関数には不偏分散が用いられていることに注意する必要がある。

R による固有値計算 R の eigen 関数を使って，固有値と固有ベクトルが計算されるところを確認する。固有ベクトルは標準化されていることに注意する。

キーワード

- 行列型
- 行列関数

F.9.2 授業情報

■コマの展開方法 R を使った演習

予習・復習課題

■予習 R/RStudio を使った分析環境を再確認しておこう。またデータの読み込みや記述統計量などの算出関数を確認しておこう。

■復習 授業時間内に収まらなかったところがあれば，必ずキャッチアップしておくこと。いくつかの練習問題を実践し，エラーや警告がでても対応できるようになろう。

F.10 R をつかった因子分析と尺度作成法

F.10.1 授業内容

科目の中でのこのコマの位置づけ

ここでは心理尺度を開発するような心理学研究を想定し，より実践的な順序に則って演習を進めていく。この講義の目標は，自らが質問紙調査を使った研究をした場合にどのような手順で行うかを理解し，実践できるようになることである。具体的には前回導入した psych パッケージを用いて，さまざまな推定オプションを追加していくことで出力が変わっていくことを確認しながら進める。

コマ主題細目

psych **パッケージ概説**　心理学研究に用いられる便利な関数群である psych パッケージのマニュアルを見ながら，describe.by などの記述統計量関数，alpha や omega といった信頼性係数の関数を使ってロウデータの分析を行う。

調査研究の手順　心理尺度の作成研究の手続きを外観する。まず構成概念の設定，定義，妥当性を考えた上で，具体的な項目を選出し，テストデータを取る。探索的な因子分析によってその因子的妥当性を確認し，標準化のための本調査を行う。あるいは 1 次元性を確認した上で，IRT によって反応段階の確認，項目母数の確認，テスト情報関数の確認などが必要である。尺度の翻訳や検証的妥当性のチェックなどについては，構造方程式モデリングによる分析を行うのでここでは扱わず，参照するにとどめる。

共通性推定の問題　分析にあたって，改めて因子分析モデルの行列表現を提示し，行列の固有値分解によって因子負荷量が求められることを確認する。しかしその際，共通性をどのように推定するかの問題が残されていたことを確認し，そのためにいくつかの方法が提案されていることを理解する。これらは因子分析を行う上で，推定方法のオプション指定に関わってくる点であり，ソフトウェアが変わっても同様の指標が必要であることをみる。

→ 小杉 (2018) の Pp.91–94.

fa **関数と探索的因子分析**　探索的因子分析の手続きを fa 関数を使いながら考える。探索的因子分析の場合は因子構造，因子負荷量について何ら前提を置かないため，因子数の推定から始めなければならない。まずは fa.parallel 関数でスクリープロットを描画する。スクリープロットを読むときの形状について確認する。続いて因子数と共通性推定方法を定めた上で fa 関数を実行し，因子負荷量や共通性などアウトプットを確認する。続いて解釈を簡単にするために因子軸の回転を行うことを解説し，実行のために rotate オプションを追加することをみる。回転前の結果との比較，また直交回転と斜交回転の違いを確認する。

→ 小杉 (2018) の Pp.81–91.

因子得点の算出　因子数と因子負荷量が明らかになると，そこから逆算的に因子得点を計算できる。fa 関数には scores オプションをつけることで，出力されたオブジェクトから因子得点を取り出すことができるのをみる。こうした方法とは別に，項目同士の素点の平均から因子得点を計算することもある。これは推定値を実体とすることの懸念が出発点であり，その長所と短所を把握しておくことが必要であ

る。この簡便法は平均値情報を含んでいるため，尺度カテゴリに依拠した解釈が可能である。また取り出された因子得点と簡便的因子得点の相関を見ることを確認する。

因子分析の注意点 因子分析を行う上で注意しなければならないのは，因子が実体としてあるのではなく，あくまでも準備された項目群の相関関係から得られる基底に過ぎないことを理解する点である。因子分析の流れの中では因子に命名することが 1 つの手順としてあるが，言葉として確定するとあたかもそれがあるかのように考えられてしまうこと，それしかないように考えられてしまうことの危険性を理解する。心理尺度の呪いやてっちゃんの手品になってしまわないように注意し，常に元の項目群に戻って考える必要があることをしっかりと理解する。

キーワード

- 信頼性係数
- 共通性
- 因子負荷量
- 因子得点
- psych パッケージ
- fa,fa.poly,fa.parallel 関数

F.10.2 授業情報

■コマの展開方法 R を使った演習

予習・復習課題

■予習 パッケージの読み込みや関数の結果を見る方法を確認しておこう。一年時のことを思い出して，lm 関数を例に R の操作方を思い出しておく。

■復習 授業時間内に収まらなかったところがあれば，必ずキャッチアップしておくこと。いくつかの練習問題を実践し，エラーや警告に対応できるようになろう。心理学研究など心理学の専門雑誌を参考に，どのような分析結果がどのように報告されているかを確認しておくことも，理解を進める。

F.11 R をつかった項目反応理論

F.11.1 授業内容

科目の中でのこのコマの位置づけ

項目反応理論を実践的に理解する演習パートである。

カテゴリカルな因子分析と数学的に同等ではあるが，より項目の特徴を広く表現できる項目反応理論の利用が，今後より重要なものになってくるだろう。

ここではまずテスト理論の根本に立ち返り，二値単因子のデータを使って 1PL,2PL モデルの分析を行う。分析結果は数値で見ることも重要であるが，ICC や IIC，TIC などを使って可視化するとより理解が深まるだろう。多段階の反応についても，同様に GRM を実行し，閾値や識別力，IRCCC や IIC，TIC が描画できることを確認する。とくに IRCCC による反応段階の読み取り方には注意する。最後に多段階で多因子の場合，項目反応理論の文脈から言えば多次元 IRT になり専用のパッケージが必要になることを紹介しつ

つ, カテゴリカル因子分析でも同様のことができることを確認する。

コマ主題細目

項目反応理論の実際 項目反応理論はテスト理論がその出自に当たるので, まずは二値データで単因子が想定できるような例を元に分析を行う。分析には `irtoys` パッケージや `ltm` パッケージを用いて, 1PL モデル, 2PL モデルの演習を行う。項目母数の値と意味が, 具体的な設問に照らし合わせて考えることで, より実感をもって理解できるようになると思われる。とくに, ICC や IIC,TIC など可視化することでその意味が理解しやすくなるだろう。3PL モデルなどさらに拡張したモデルも利用可能である。

段階反応モデルの実際 続いて単因子, 段階反応モデルの実践を行う。因子構造として, 前回の授業で扱った多因子の内, ある因子に限定して分析を行うこととする。段階反応モデル (GRM) の出力結果を数値だけでなく可視化することで, 項目の特徴がどのように表現されているかを考える。とくに反応段階の山が潰れているようなケースは, 適切な反応段階でなかったことを意味するので, 数値の置き換えなど元データを修正しつつ分析し直すことを考える。これらを通じて, 適切な反応段階による調査法が必要であることを理解する。

カテゴリカル因子分析との対応 多段階, 多因子の場合は `psych` パッケージの `fa` 関数にオプションを追加することでできる, カテゴリカル因子分析と同じである。出力結果について, これまでの相関係数を用いているものとの違いを確認する。また数値をどのように変換すれば対応するのかを見ることで, 数学的に等価であることを確認しておく。IRT の側面から多因子に拡張した, 多次元 IRT についても, `mirt` パッケージを利用すれば実行できる。この解析には計算時間がかかるが, 完全情報最尤推定の結果が得られることは利点である。

キーワード

- 1PL モデル, 2PL モデル, 3PL モデル
- 段階反応モデル
- カテゴリカル因子分析
- `irtoys`,`ltm`,`psych`,`mirt` パッケージ

F.11.2 授業情報

■コマの展開方法 R を使った演習

予習・復習課題

■予習 `irtoys`,`ltm` パッケージを事前にインストールして環境を整え, データファイルの読み込みなど R/RStudio の基本的な使い方を確認しておこう。とくに R の `data.frame` 型に含まれる変数が, `numeric` なのか `factor` なのかによって挙動がかわることがある。変数の型についても再確認しておこう。

■復習 本講で習ったパッケージを使って, 具体的なデータを因子分析, IRT, カテゴリカル IRT などいくつかのモデルで分析し, それぞれの違いを確認しておこう。

F.12 構造方程式モデリング

F.12.1 授業内容

科目の中でのこのコマの位置づけ

構造方程式モデリングは，因子分析と回帰分析を統合して扱う，総合的分析モデルである。言い換えれば，これまでの多くの多変量解析モデルのほとんどは，構造方程式モデルの下位モデルとして表現できる。ここではこれまでのモデルを統合した，より現代的でより上位のモデルである構造方程式モデリングを理解することで，すべての多変量解析を網羅的かつ俯瞰的に捉えることが狙いである。

構造方程式モデリングを理解するには，変数の種類と関係性の区分に注意したパスダイアグラムの描き方を知ることが早い。パスダイアグラムを用いると，回帰分析と因子分析は説明変数が観測変数なのか潜在変数なのかといった違いであることが明らかである。また因子分析と似た主成分分析がどのように表されるかも，パスダイアグラムを見れば一目瞭然である。

パスダイアグラムには変数の尺度水準までかきこまれることはないが，ここに注意していろいろなモデルを描画すると，それがかつて多変量解析においてさまざまな名称で呼ばれた分析方法であったことがわかる。あるいは，今後どのようなモデルが開発される可能性があるか，どのようなモデルをどのように希釈すれば良いかもイメージできる。

加えてこの統合的なモデルがなぜそうした複雑なモデルを表現できるのかについても，モデルを方程式で描画し，行列で考えることで，モデル行列とデータ行列を近づけることと理解できる。この観点から，データにモデルを当てはめる適合度の考え方が改めて理解されるだろう。

コマ主題細目

パスダイアグラムの書き方 これまで学んできたモデルを図で表現することを学ぶ。そのためには，変数を観測変数と潜在変数に区別することと，変数間関係を因果関係と相関関係に区別する必要がある。観測変数を矩形，潜在変数を楕円形，因果関係を一方向矢印，相関関係を双方向矢印で表現することで，因子分析や回帰分析が図で表現できることを学ぶ。

パスダイアグラムによるさまざまなモデル 因子分析と回帰分析をパスダイアグラムで表現したことで，この両者を統合するような表現ができること，また潜在変数同士の関係を記述する，構造方程式を描画できるようになったと言える。因子分析と似た手法とされる主成分分析や，尺度水準の違いによるさまざまな統計モデルを表現する方法を手に入れたことになる。この手法を総称して，構造方程式モデリングと呼ぶ。

→ 小杉･清水 (2014) の Pp.7–10

構造方程式モデルによる未知数の推定 構造方程式モデリングでは，パスダイアグラムでも表現されるが，変数間関係を方程式で書くこともできる。方程式で描画することで，構造方程式も潜在変数の方程式と観測変数の方程式，それらが入れ子になった方程式で描画できることがわかる。またこれらのモデルを行列のイメージで捉えると，最終的には分散共分散行列という実態を持った数字に対して，未知数で描画された方程式を接続したことが直感的にわかるだろう。未知数の増え方と分散共分散行列の要素の増え方を比較すると後者が圧倒的に早く，未知数よりも既知数が多い方程式は解くことができるという原理から，未知数が推定しうることを理解する。

→ 小杉 (2018) の Pp.191–193.

適合度によるモデルの評価 データ行列とモデル行列をイコールで結んだ方程式を解くことが，未知数を求める根本的な原理であるが，このことからモデルがデータとどの程度合致しているかという適合度が，モデル評価の統合的観点として浮かび上がってくる。回帰分析では R^2 であったが，因子分析をはじめとしたさまざまな多変量解析モデルも，この評価次元で考えることができる。ただしその指標にはいくつかの特徴があり，これらを総合的にみて評価するという実践的ノウハウも確認する。

→ 小杉・清水 (2014) の Pp.10–12

キーワード

- パスダイアグラム
- 観測変数と潜在変数
- 因果関係と相関関係
- 構造方程式モデリング
- 適合度

F.12.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 回帰分析と因子分析という 2 つの分析方法についてはすでに学んでいるが，この両者の共通点と相違点がどこにあるかを事前に考えてみよう。外的な基準の有無，説明変数の種類の観点から，自分の言葉で表現できるようになっていると良い。

■復習 これまで学んださまざまな統計モデルを，構造方程式モデリングの表記法に則ってパス図を書いてみよう。またさまざまな尺度水準の組み合わせからなるモデルを考え，それらがどのような意味を持つのかと推論するのも理解の助けになる。

F.13 R による構造方程式モデリング

F.13.1 授業内容

科目の中でのこのコマの位置づけ

これまでの流れと同じで，統計技術の理論を知っただけではなく，自分で実際に計算できる演習を経てこそ理解が深まるということから，本講では R をつかって実際に構造方程式モデリングを解くことを演習的に学ぶ。構造方程式モデリングを実装するパッケージは複数あるが，最も応用範囲がひろい `lavaan` パッケージを用いることにする。

まずは観測変数だけからなる簡単なパス解析を行う。データの入力の仕方，方程式の設定，関数の使い方などを一通り習得する。続いて潜在変数を含んだモデルによる解析を行う。モデルの適合度や修正指数を参考に，徐々にモデルを書き換えていく手順を学ぶ。注意すべきは，適合度を上げることが目的になって，不自然な仮定やパスをおいてしまうことである。あくまでも具体的かつ妥当なモデリングを心がけるべきである。

オプショナルな設定になるが，観測変数がカテゴリカルである場合や，推定方法の選択なども確認する。最後に，R 以外の統計パッケージによる構造方程式モデリングの実践例がいくつか紹介される。

コマ主題細目

方程式の入力 まずは観測変数同士の関係をパスでつなぐモデルで練習する。パス解析は回帰分析の繰り返しで実行することもできるが，構造方程式モデリングによってパスの繋がりを 1 つのモデルで表現し，適合度も統一できるなどの利点がある。観測変数だけからなるモデルの結果と，実際に `lm` 関数で実行した結果と比較すると良い。またパッケージにもよるが，自動的にパスダイアグラムを描画してくれるものもある。方程式とそのパスダイアグラムによる表現の対応を確認する。

→ 小杉・清水 (2014) の Pp.55–60

測定モデルの実践 因子分析モデルを SEM 上で実行してみる。探索的因子分析と違い，どの項目にどの因子が影響しているかを固定したモデリングが可能であり，この検証的因子分析による結果と，いわゆる因子分析関数との結果を比較することで，パスが引かれていないところはその係数が 0 であるという強い仮定をおいていることを確認する。また尺度作成の観点からは，検証的因子分析をすることで因子的妥当性や弁別的妥当性，収束的妥当性を検討することもできる。さらに同じモデルを別のデータに適用することで多母集団同時分析を行うことになる。このように，モデルの暗黙の過程や，モデルとデータの適合という側面にとくに注意する。さらに測定モデルと測定モデルをつなぐ，構造方程式を扱ったモデルへと拡張する。

→ 小杉・清水 (2014) の Pp.87–90

実践上の注意点 ここまでを通じて，一通りモデルを作成できるようになった。とくに構造方程式を踏まえると，心理的実体同士の関係を描画したと解釈できるため，心理学的概念間の関係を記述できることは魅力的に映るかもしれない。しかしデータを越えての解釈はご法度であり，潜在変数が心理的実在であるかどうかの議論は，理論的背景や測定の適切さ，標準化されないスコアが実際にどのように変化すれば何が言えるのか，といったところに一足飛びに行かぬよう注意する必要がある。またモデル改良のステップにおいて，適合度や修正指数を過度に参照していないか，注意する必要がある。

そのほかの統計パッケージ 構造方程式モデリングの利点は，モデルを可視化したことにもある。たとえば AMOS は GUI でモデルを作成できる。他に Mplus はカテゴリカルな変数にも対応しているし，高度に複雑なモデルであっても表現が可能である。

キーワード

- 測定方程式
- 構造方程式
- 潜在変数を含んだモデル
- 多母集団同時分析
- 適合度

F.13.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 構造方程式モデルは，数式レベルでの理解は難しいが実際は統合的なものであり，回帰分析や因子分析をその下位モデルとして含んでいる。改めて，回帰分析や因子分析を単体で行った場合にどのような出力がなされるのか確認しておくと，同じものを構造方程式で実践したときの違いが明確に意識できるようになる。

■復習 さまざまなモデルを試すなかでは，エラーや警告が出ることもある。そうしたエラーや警告の意味を理解し，またそれに対応するためにはどのような方法が取れるかを考える必要がある。まずは手元のデータを用いて，これらの練習を行うと良い。

F.14 双対尺度法

F.14.1 授業内容

科目の中でのこのコマの位置づけ

ここまでは尺度化によって与えられた数値を元に，因子構造を検証したり (潜在) 変数間関係を記述したりすることをみてきた。ここでは尺度化によって与えられる数値に改めて注目し，得られた情報を最も有効に活用するように数字を割り振る，数量化の考え方へと考え方を進める。

因子分析や構造方程式モデリングで用いられる相関関係や共分散関係は，いずれも与えられた数字がどの程度の直線的関係にあるかという観点からモデルを組み立てていた。そのモデルの表現力は非常に豊富なので，スタートとなる変数同士の直線性 (相関係数を使うこと) を改めて問うことがなかった，あるいは多少の違和感を抑えて進んでしまうことがある。改めてその線形性に注意を払い，逆に線形性を最大にするように数値をデータに付与するという発想を転換させるのが，数量化の手法である。

数量化はいくつかの種類があるが，ここでは III 類を取り上げる。III 類は双対尺度法や対応分析とも呼ばれ，行と列の両方に数字を付与する。この方法によってさまざまな表現ができることを確認し，これが応用的側面ではテキストマイニングなどにも利用されているところを見る。こうした応用方法は臨床場面でも活きるものであり，どのようなデータにどのように応用できるかを考えることで，データに対して積極的に関わる姿勢を身につけることが期待される。

コマ主題細目

直線的でない関係 心理学的には中庸が良いことも少なくないが，であればカテゴリに付与される尺度値からは U 字あるいは逆 U 字の関係がえらえる。この関係は相関係数としては 0 に近くとも，無関係，無意味を表すものではない。そこから意味が取り出せないのであれば，数値化のルールを修正するべきであって，どのように関係性の高い数値を与えるかについては，行または列の平均からカテゴリの値を付け直すことである。ここでの目的は，直線性を取り出すためにカテゴリに数値を与える数量化という別の観点であることに注意する。

→ 西里 (2010) の Pp.6–25.

林の数量化理論 分析者にとって最も有用な情報が得られるようにカテゴリに数値を与える，という観点を数量化と呼ぶ。数量化の対象は離散変数や名義尺度水準の変数であってもよく，これらを用いた分析方法については，行動計量学の祖である林知己夫の多彩な手法をまとめた呼称である数量化 I 類，II 類，III 類などがある。I 類，II 類は重回帰分析や判別分析とも関係するが，数量化という観点か

ら議論されていることに注意が必要である。数量化 I 類，II 類の応用例を外観することで分析方のイメージを掴む。

→ 小杉 (2019) の Pp.189-195.

双対尺度法による分析 数量化 III 類は開発者によって双対尺度法や対応分析など異なる名称を持つが，数学的には等価でいずれも目指すところは名義尺度水準の主成分分析である。リッカート法での尺度に値をつけた時のように，行及び列に含まれるカテゴリカルな区分に対してもっとも線型性が高くなるように数値を割り当てる。ここで行列計算にその目を向ければ，矩形行列に対する特異値分解によって，行の空間と列の空間を用意し，カテゴリに座標を与えることを意味していることになる。数量化 III 類，双対尺度法，対応分析の細かな違いにも注意しつつ，行カテゴリと列カテゴリを共通空間に表した図から何が読み取れるかを考える。

→ 小杉 (2019) の Pp.195-199.

テキストマイニングへの応用 数量化理論の対象が名義尺度水準であることを考えれば，およそ言語化できたものはすべて多変量解析として分析できることになる。逐語録や自然言語の解析にはテキストマイニングと呼ばれる手法が用いられるが，この技術は形態素解析と多変量解析の組み合わせであり，多変量解析の元になる共変動が何で表されているかに着目すれば，統合的に解釈することが可能である。テキストマイニングには専門的なソフトウェアがあるが，軽く言及するにとどめる。

テキストマイニングについては → 樋口 (2020) を参照せよ。

キーワード

- 数量化
- 双対尺度法
- 特異値分解
- テキストマイニング

F.14.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 構造方程式モデリングでさまざまなモデルを考えた際，変数が観測・潜在の別だけでなく尺度水準の組み合わせも変えて考えた場合，どのようなモデルがありえるかが想定できると思います。今回はとくに名義尺度水準のモデルになるので，名義尺度水準のモデルはどのようなものがあるかを想定してみてください。

■復習 名義尺度水準のモデルを手に入れたことによって，さまざまな分析の可能性が広がったのではないでしょうか。今までおよそ統計的・数値的アプローチの対象にないと思われていたものに対しても，どのように，どこまでであればアプローチ可能で，どこからが限界になるのかを考えることが実践的には役に立つ視点です。統計モデルを使いこなすために，ぜひさまざまな応用例を考えてみてください。

F.15 多次元尺度構成法

F.15.1 授業内容

科目の中でのこのコマの位置づけ

数量化まで学ぶことによって，カテゴリに適切な数値を割り振るという尺度化の原点に立ち返ることができた。ここではさらに，心理尺度のような回答法ではない変数間同士の関係から，次元を取り出して分析する多次元尺度構成法について考える。この方法は実験刺激や知覚的反応，直感的判断などを対象にできるため，応用可能性が非常に高いだろう。

多次元尺度構成法を理解するためには，まず実際の距離行列を分析して地図を再構成できるかどうかを見るところから始めるのが良いだろう。データとして与えられるのが距離行列であり，行動計量学ではこれを心理的な距離や意味的な距離が数値化されたものだと捉えることで，心理学的な地図を作っていると解釈してきた。この仮定には最大限の注意を払いつつ，必ずしも計量的でない場合の数値化をする非計量的多次元尺度法に拡張することで，更なる心理学的用途が広がることを見る。なお，多次元尺度方は数量化 IV 類と同じである。

多次元尺度法は分析の元が距離行列であり，その基底を固有値分解によって得るといういみで，多変量解析としてはお馴染みの考え方であるともいえる。しかしデータが距離 (を意味するもの) であれば良く，数学的にも簡単な拡張をすることで，個人差を表現するモデルに拡張することもできる。また心理尺度に対する考え方として，個人の内的な次元からの近さに応じて反応すると考える，展開型のモデルを使うことは，心理尺度の利用に新な視点をもたらす。

コマ主題細目

多次元尺度構成法 多次元尺度構成法 (MDS) は，距離行列を元にした多変量解析の一種であり，距離関係から次元 (基底) を選び出し，対象に座標を与える方法である。まずは座標の復元例から考え，抽出する次元数をどのようにして求めるかといった基礎的な知識を得る。また，行列の考え方からみると正方対称行列の固有値分解であるから，これを確認するだけでも他の多変量解析と合わせた統合的な理解ができるだろう。因子分析モデルも多次元尺度法の一種であるということもできる。

距離と心理学のデータ 距離行列があれば次元が抽出できることが分かったが，さて何を距離とみなすかを考えれば，非常に多くの可能性が広がることがわかる。距離の定義は非負で対称性と 3 角不等式が成り立つことであり，さまざまな距離の定め方があるし，共分散や相関もその一種と考えることができる。元になるデータも尺度評定を用いる方法，刺激の混同率，代替価/連想価，刺激の汎化勾配，反応潜時，ソシオメトリックなデータなど，心理学のさまざまな領域で得られるデータが，距離とみなすことができる。応用可能な領域が広いことを知ることで，統計モデルをハンドリングできることになる。

→ データの例に関しては高根 (1980) の Pp.14–27.

非計量多次元尺度法 計量 MDS によって算出される座標は元のデータをうまく復元するが，心理学的なデータの場合はデータの大小関係の表現 (順序尺度水準) がせいぜいであり，これに対応した非計量多次元尺度法が考案されている。この手法を用いることで，一対比較や順序比較などのより制限の少ないデータからであっても数量関係を導き出すことができる。

→ 計量・非計量多次元尺度構成方については，すでに絶版になったが岡太・今泉 (1994) がもっとも簡潔でわかりやすく説明している。手に入るところでは足立 (2006) の P.135–143，あるいは小杉

(2019) の P.199-203.

多次元尺度構成法の展開 多次元尺度構成法で作られたものは地図である。地図には点を書き込んだり，複数の地図を重ねたりできるように，多次元尺度法で得られた地図にも情報を追加したり，モデルを展開するなどしてさまざまな応用的モデルを作ることができる。ここでは Prefmap や楕円モデル (非対称 MDS)，INDSCAL など応用例をいくつか示し，この技術の応用可能性を考える。

展開法 Coombs が考えた心理尺度の展開法は，サーストン法やリッカート法とはまた別の尺度化の考え方を表している。この方法は被験者とカテゴリーの両方を地図上にプロットできる。具体的な分析例をみながら，尺度やそれに数字を与える方法についての考え方を見る。

→ 展開型モデルについては，多少複雑な工夫が組み込まれているが，清水 (2018) が参考になる。

キーワード

- 多次元尺度構成法
- 非計量多次元尺度構成法
- 個人差多次元尺度構成法
- 展開型多次元尺度法

F.15.2 授業情報

■コマの展開方法 講義

予習・復習課題

■予習 数量化の関係を再確認しよう。すなわち数値に値を与えるというものであり，尺度のカテゴリだけでなくより一般的な心理的刺激を考え，どういったモデルで表現できるかを考えると本講だけでなく後期の授業にもつながる気づきを得るだろう。

■復習 多次元尺度法によって，どのような分析ができるかを考えてみよう。とくに尺度法にかかわらず，実験刺激からの反応を距離と見做せる関係にすれば分析でき，その結果をどのように解釈するかについても自由度はかなり多い。紹介された発展的なモデルなどについても自分の研究関心にどのように応用できるか考えてみよう。

引用文献

Abelson, R. P. (1954). A technique and a model for multi-dimensional attitude scaling. *Public Opinion Quarterly*, *18*(4), 405–418.

足立 浩平 (2006). 多変量データ解析法——心理・教育・社会系のための入門—— ナカニシヤ出版

Allport, G. W.(1967). Attitudes. In M. Fishbein (Ed.), *Readings in attitude theory and measurement* (pp. 3–13). John Wiley & Sons Inc.

Chalmers, R. P. (2012). mirt: a multidimensional item response theory package for the R environment. *Journal of Statistical Software*, *48*(6), 1–29. https://doi.org/10.18637/jss.v048.i06

Dobson, A. J. (2008). *Annette Jane Dobsons*. Chapman & Hall/CRC Press.
(ドブソン, A.J. 田中 豊・森川 敏彦・山中 竹春・冨田 誠 (訳) (2021). 一般化線形モデル入門 共立出版)

Epskamp, S. (2021). Semplot *GitHub repository*, https://github.com/SachaEpskamp/semPlot.

藤原 武弘 (2001). 社会的態度の理論・測定・応用 関西学院大学出版会

Grimm, L. & Yarnold, P. (1994). *Reading and Understanding Multivariate Statistics*. American Psychological Association.
(グリム, L.G. & ヤーノルド, P.R. 小杉 考司・高田 菜美・山根 嵩史 (訳) (2016). 研究論文を読み解くための多変量解析入門 基礎篇: 重回帰分析からメタ分析まで 北大路書房)

Grimm, L. & Yarnold, P. (2001). *Reading and Understanding More Multivariate Statistics*. American Psychological Association.
(グリム, L. & ヤーノルド, P. 小杉 考司・高田 菜美・山根 嵩史 (訳) (2016). 研究論文を読み解くための多変量解析入門 応用篇: SEM から生存分析まで 北大路書房)

Guilford, J. P. (1954). *Psychometric Methods*. McGraw-Hill Book Company.
(ギルフォード, J.P. 秋重 善治 (訳) (1959). 精神測定法 培風館)

豊田 秀樹 (2012). 項目反応理論 [入門編] 第 2 版 朝倉書店

樋口 耕一 (2020). 社会調査のための計量テキスト分析——内容分析の継承と発展を目指して—— 単行本 ナカニシヤ出版

平岡 和幸・堀 玄 (2004). プログラミングのための線形代数 オーム社

清水 裕士 (2018). 阪神ファン – 巨人ファンの 2 大精力構造は本当か 豊田秀樹 (編) 北大路書房

Holzinger, K. J. & Swineford, F. (1939). A study in factor analysis: the stability of a bi-factor solution. *Supplementary educational monographs*,

Isaacson, W. (1995). *Steve Jobs*. JSTOR.
(アイザクソン, W. 井口 耕二 (訳) (2011). スティーブ・ジョブズ 講談社)

Isvoranu, A.-M., Epskamp, S., Waldorp, L., & Borsboom, D. (Eds.) (2022). *Network psychometrics with r: a guide for behavioral and social scientists*. Routledge.

川端 一光・荘島 宏二郎 (2014). 心理学のための統計学入門——[心理学のための統計学 1]: ココロのデータ分析—— 誠信書房

加藤 健太郎・山田 剛史・川端 一光 (2014). R による項目反応理論 オーム社

小杉 考司 (2018). 言葉と数式で理解する多変量解析入門 北大路書房

小杉 考司 (2019). その他の他変量解析 楠見 孝・日本心理学会 (編) 心理学統計法 (pp. 189–206) 遠見書房

小杉 考司・清水 裕士 (編) (2014). M-plus と R による構造方程式モデリング入門 北大路書房

小杉 考司 (2014). 学校適応感尺度 fit の開発. 研究論叢. 第 3 部, 芸術・体育・教育・心理, *64*, 69–82.

Kruskal Joseph, B. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*(1), 1–27.

Kruskal Joseph, B. (1964b). Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, *29*(2), 115–129.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*, *22*, 5–55.

三中 信宏 (2018). 統計思考の世界——曼荼羅で読み解くデータ解析の基礎—— 技術評論社

三浦 麻子・小林 哲郎 (2015). オンライン調査モニタの satisfice に関する実験的研究. 社会心理学研究, *31*(1), 1–12. https://doi.org/10.14966/jssp.31.1_1

宮川 雅巳 (1997). グラフィカルモデリング (統計ライブラリー) 朝倉書店

宮谷 真人・坂田 省吾・林 光緒・坂田 桐子・入戸野 宏・森田 愛子 (編) (2009). 心理学基礎実習マニュアル 北大路書房

Muller, J. & Muller, J. Z. (2018). *The tyranny of metrics*. Princeton University Press. (ミュラー, ジェリー & ミュラー, ジェリー・Z 松本 裕 (訳) (2019). 測りすぎ——なぜパフォーマンス評価は失敗するのか?—— みすず書房)

村上 正康・佐藤 恒雄・野澤 宗平・稲葉 尚志 (2016). 教養の線形代数 培風館

Muraki, E. (1992). A generalized partial credit model: application of an em algorithm. *ETS Research Report Series*, *1992*(1), i–30.

長沼 伸一郎 (2011). 物理数学の直観的方法〈普及版〉 講談社

西村 武 (1977). 主観評価の理論と実際. テレビジョン, *31*(5), 369–377. https://doi.org/10.3169/itej1954.31.5_369

西里 静彦 (2010). 行動科学のためのデータ解析 – 情報把握に適した方法の利用 培風館

Norretranders, T. (1999). *The user illusiton*. Penguin. (ノーレットランダーシュ, T. 柴田 裕之 (訳) (2002). ユーザーイリュージョン——意識という幻想—— 紀伊國屋書店)

岡太 彬訓 (2008). データ分析のための線形代数 共立出版

岡太 彬訓・今泉 忠 (1994). パソコン多次元尺度構成法 共立出版

小野島 昂洋 (2021). Lav2tikz.r GitHub repository, https://github.com/onoshima/myfunction.

小塩 真司 (2020). 性格とは何か——より良く生きるための心理学—— 中央公論新社

Partchev, I., Partchev, M. I., & Suggests, M. (2017). Package 'irtoys'. *A collection of functions related to item response theory (IRT)*,

Revelle, W. (2021). psych: Procedures for Psychological, Psychometric, and Personality Research. R package version 2.1.3. Northwestern University. Evanston, Illinois. from https://CRAN.R-project.org/package=psych

Rizopoulos, D. (2006). Ltm: an r package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, *17*(5), 1–25.

Samejima, F.(1997). Graded response model. In W. J. Linden & R. K. Hambleton (Eds.), *Handbook of modern item response theory* (pp. 85–100). Springer.

芝 祐順 (1979). 因子分析法 東京大学出版会

新納 浩幸 (2007). R で学ぶクラスタ解析 オーム社

Stevens, S. S. (1946). On the theory of scales of measurement. *Science*, *103*(2684), 677–680.

末永 俊郎 (編) (1987). 社会心理学研究入門 東京大学出版会

高橋 正視 (2002). 項目反応理論入門ー新しい絶対評価 イデア出版局

高根 芳雄 (1980). 多次元尺度法 東京大学出版会

田中 良久 (1977). 心理学的測定法 東京大学出版会

豊田 秀樹 (2000). 共分散構造分析——構造方程式モデリング 応用編—— 朝倉書店

豊田 秀樹 (2007). 共分散構造分析——構造方程式モデリング 理論編—— 朝倉書店

豊田 秀樹 (2008). データマイニング入門 東京図書

豊田 秀樹 豊田 秀樹 (編) (2023). 人工知能入門ー初歩からGPT／画像生成AIまでー 東京図書

Van Lissa, C. J. (2019). Tidysem: a tidy workflow for running, reporting, and plotting structural equation models in lavaan or mplus. *GitHub repository*, https://github.com/cjvanlissa/tidySEM/.

山田 剛史・村井 潤一郎 (2004). よくわかる心理統計 ミネルヴァ書房

山内 光哉 (2010). 心理・教育のための統計法 サイエンス社

永田 靖 (2005). 統計学のための数学入門 30 講 朝倉書店

千野 直仁・岡田 謙介・佐部利 真吾 (2012). 非対称 MDS の理論と応用 単行本 現代数学社

永谷 文代・松嵜 順子・諏訪 絵里子・上西 裕之・谷池 雅子・毛利 育子 (2022). 教師記入式実行機能行動評定尺度の小学生に対する信頼性及び妥当性の検証. 心理学研究, *92*(6), 554–563. https://doi.org/10.4992/jjpsy.92.20226

索引

た

な

は

ま

や

ら

わ