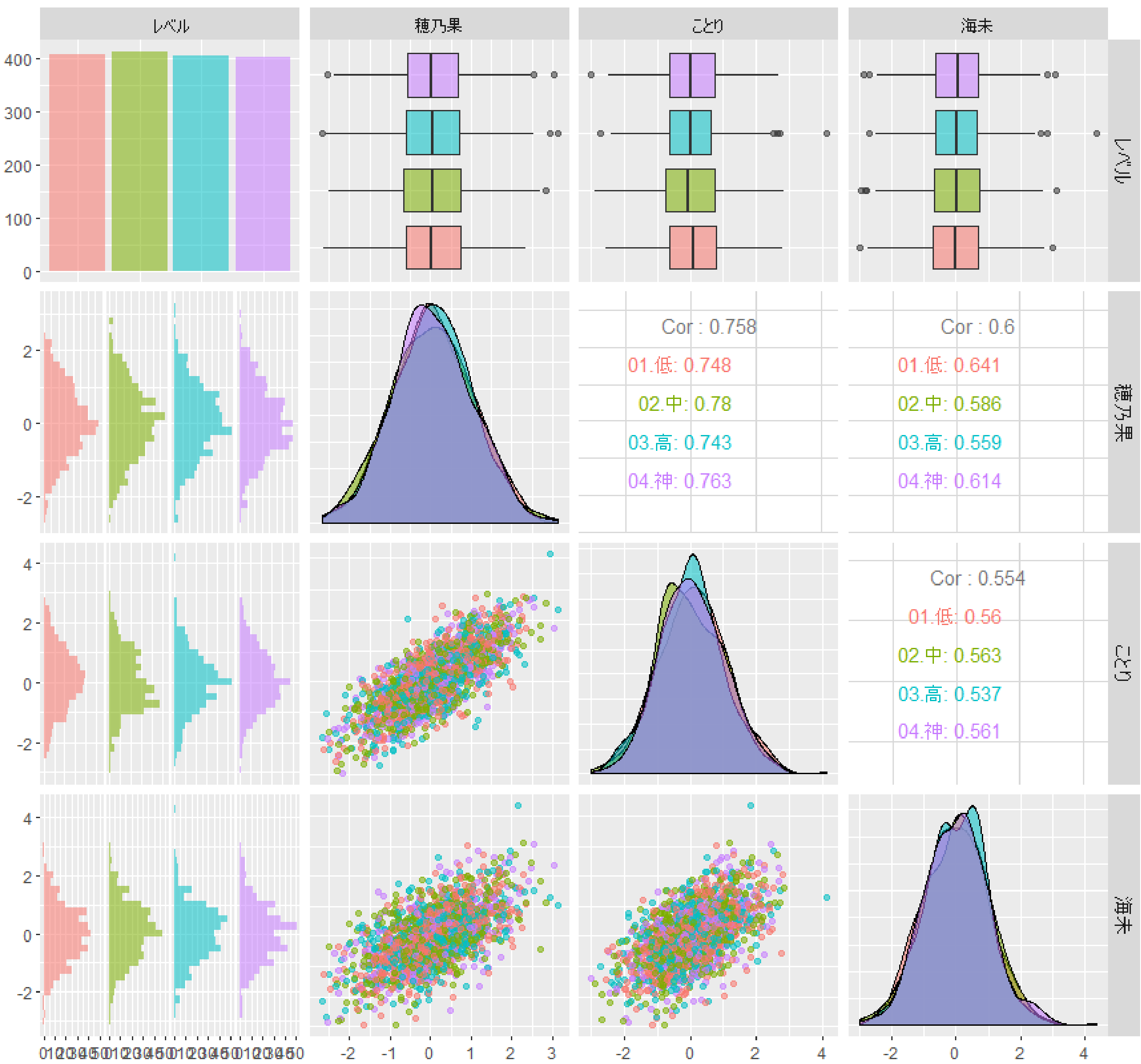
ラブライブ！とうけいもでりんぐ



ラブライブ！キャラクター人気投票の数理

1．はじめに

政治、スポーツ、ビジネスなどあらゆる世界でランキングをつける習慣が根付いています。アニメの世界でも例外ではなく、どのキャラクターが人気があるかなどを評価して、Web 上での話題のネタになったり、対立の種になったりしています。
ラブライブ！の世界でも例外ではなく、定例的に実施される総選挙だけでなく、特定のシチュエーションを想定した人気投票が多数実施されています。
これらの多くはたいてい多数決によって 1 位を決めること、つまりある選択集合のうち 1 つだけを選びその得票数が 1 番多かった者に no1 の称号を与えるということが自然と受け入れられています。この自然と受け入れられいる多数決による投票に疑問符を投げかけ、統計的アプローチにより精度の高いランキングづけすることが本稿の目的です。

このような多数決による人気投票の仕組みをもっとも公平で民主主義的な投票制度とするためには以下の 4 つの条件が満たされている必要があります。

① **定義域の非原性**
すべての投票者は候補者を望みのままの順番でランク付けできなければならない。

② **パレート原則**
候補 A と候補 B を比較した場合、すべての投票者が候補 A を選ぶならば、つねに A が B より上にランク付けされなければならない。

③ **非独裁性**
投票者はだれひとりとして、ランキングを左右するような権限を持っていてはいけない。
これは例えば、最後の 1 票でランキングが決まるような状態になっているということでしょうか。

④ **無関係な選択肢からの独立性(I.I.A)**
例えば 2 人の候補者 A と B がそれぞれ 60%と 40%の得票率を得られると仮定した場合、A が B より上位であるが、2 人以外の C の候補者が選択候補に入り、A、B、C それぞれの候補者の得票率が 30%、40%、30%と変化した場合、A と B の順位は逆転した状態となっている。この状態を I.I.A 特性を満たさない状態という。
つまり I.I.A 特性を満たす条件というのは C の候補者が乱入しても、A、B の相対的な順位は変化してはならないということです。

経済学者のケネス・アローはこの 4 つの条件を同時に満たす”完璧”な投票制度は存在し得ないと 1951 年の論文で証明しました。なので、実質的に完璧な人気投票やランキングづけを考えることは無意味なことのように思えますが、今ある多数決の投票よりよいアプローチでランキングづけを考えることは可能です。
ここでは特に自分自身が興味のある④の無関係な選択肢からの独立性(I.I.A 特性)の緩和を目指して

ラブライブキャラクターのランキングづけを提案してみようと思います。

2. 無関係な選択肢からの独立性とは？

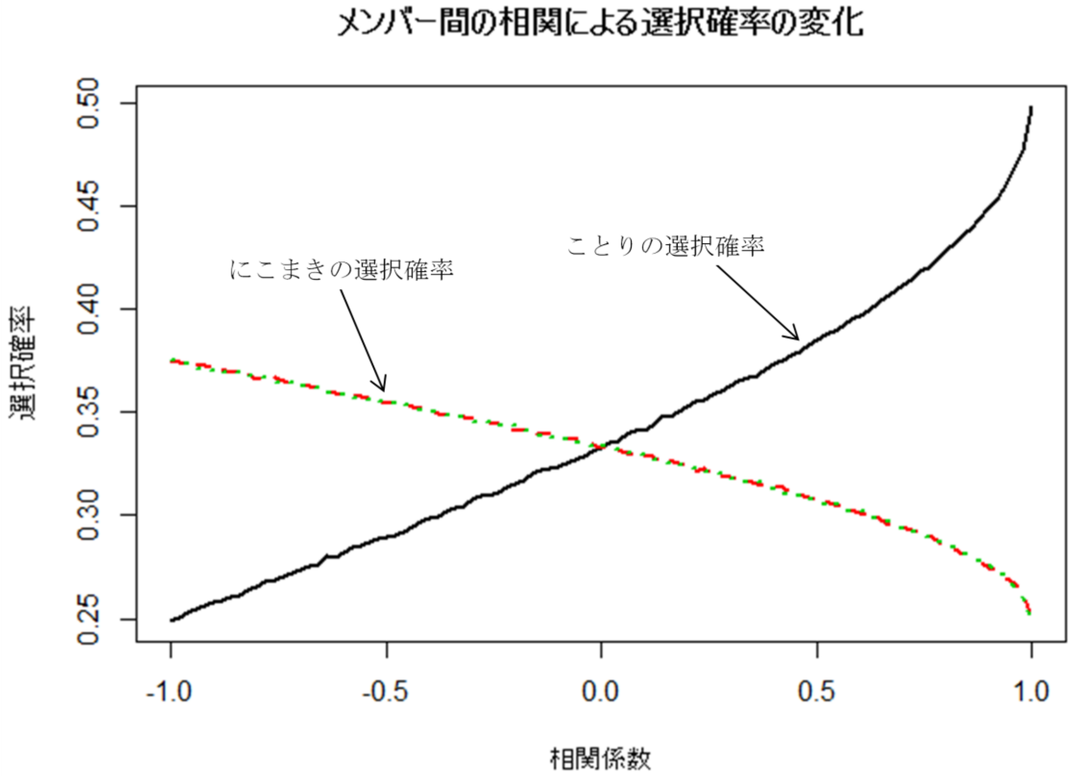
前節で I.I.A 特性について少し説明しましたが、統計的なアプローチを用いて具体的に説明したいと思います。統計的なアプローチである「離散選択モデル」で投票のシミュレーションをしたいと思います。離散選択モデルとは選択集合からある選択肢の集合から 1 つあるいは複数を選択する行動を統計モデルとして表現する数理モデルの 1 つです。離散選択モデルはある選択集合に対しての相対的な効用によって選択結果が変わると仮定します。効用とはその選択肢に持っているスコアであり、例えばにこ、真姫、ことりの 3 人の選択肢から 1 番好きなキャラクターを 1 人だけを選ぶ場合、3 人に対する効用の値がそれぞれ 50 点であるとき、選択確率は 3 人とも 33%になります。
このように 3 人に対する効用の値が同じであれば 3 人の選択確率が同じになる場合を「無関係な選択肢からの独立性を満たす」と言います。
これを満たさないパターンを考えるためににこと真姫が相関しているケースを考えます。これはにこが好きなら真姫も好きだ。ということを意味しています。この効用パターンを数理的に表すために効用は以下の多変量正規分布に従うと仮定します。

$$f(x) = (2\pi)^{-\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\mathbf{\Sigma}^{-\mathbf{1}}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

この複雑な数式のΣは分散共分散行列(あるいは相関行列を)を x は効用の実現値をμは効用の平均構造を表しています。以下は相関行列の例です。

$$\begin{pmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

この分散共分散行列の値により、効用値が同じ場合でも選択確率が大幅に変化してしまうことをシミュレーションにより示します。
具体的にはにこ、真姫、ことりの効用の平均構造を 3 人とも 0 に固定し、にこと真姫の間の相関係数をにこと真姫の相関係数を-1～1 まで変化させ(上記の相関行列の上の行列の 2 行 1 列目と 1 行 2 列目がにこと真姫の相関パラメータを表しており、この 0.5 の値の部分を変化させます。)、多変量正規分布から指定したパラメータに基づく乱数を大量に発生したときにそれぞれの乱数で最大の効用値が出たキャラを選択結果として、選択確率の変化の仕方を観察してみます。以下のグラフに示すように選択確率が大幅に変化しているのが見て取れます。



にこまき間の相関係数	ことり	にこ	真姫
-1.0	25.1%	37.5%	37.4%
-0.9	25.8%	37.1%	37.1%
-0.8	26.6%	36.7%	36.6%
-0.7	27.3%	36.3%	36.3%
-0.6	28.1%	36.0%	35.9%
-0.5	29.1%	35.5%	35.4%
-0.4	29.8%	35.1%	35.1%
-0.3	30.6%	34.7%	34.7%
-0.2	31.5%	34.2%	34.3%
-0.1	32.4%	33.8%	33.8%
0.0	33.3%	33.4%	33.3%

にこまき間の相関係数	ことり	にこ	真姫
0.1	34.3%	32.9%	32.9%
0.2	35.2%	32.4%	32.4%
0.3	36.3%	31.8%	31.9%
0.4	37.3%	31.3%	31.4%
0.5	38.5%	30.6%	30.8%
0.6	39.8%	30.2%	30.0%
0.7	41.3%	29.3%	29.5%
0.8	42.9%	28.5%	28.6%
0.9	45.0%	27.5%	27.5%
1.0	50.0%	25.0%	24.9%

図 1．にこまきに相関がある場合の 3 人の選択確率の変化

シミュレーション結果を見てみると相関の絶対値が大きいほど相関が 0 の場合(つまり I.I.A 特性が満たされている場合)と比較して、選択確率が大きく歪んでいることが見て取れます。負の相関がある場合、にこまきの選択確率はことりちゃんと比較して有利に正の相関がある場合は不利になるというパターンがありま

す。その確率の変動は大きく、ことりの選択確率は 25%から 50%まで大幅に変化しています。(このシミュレーションを実行する R コードは Appendix を参照してください。)
この結果は必ずしも、前述の効用値がもっとも高いキャラクターが 1 位になることが限らないことを示し、キャラクター間の相関構造が選択結果に大きく影響を及ぼしていることがわかります。
I.I.A 特性は特に政治の世界ではかなりの問題で 2000 年アメリカ大統領選挙のアルゴア(民主党)とブッシュ(共和党)の戦いでしたが、民主党に近いラルフ・ネーダーが大統領選挙に加わったため、アルゴアの投票の一部を奪ってしまい、ブッシュが勝利したという事例があります。

3. スクフェスのデータからラブライブキャラの人気と相関構造を探る

この節では実際のデータを用いて、ラブライブキャラの人気とキャラ間の相関構造を推定したいと思います。解析するデータはスクフェスの 7 月下期イベント「第 1 回なかよしマッチ」の score ランキングの上位に入ったユーザーから 1628 件をサンプルデータとして取得しました。ユーザーデータには現在パートナーに設定しているキャラクターと累積スコアが高いキャラクター上位 3 位までを取得することができるのですが、今回解析に用いるのは累積スコア上位 3 位のデータを取得しています。
抽出条件として、score ランキングを選んだのは累積ポイントと比較してイベント対象キャラに依存しにくいと考えたからです。また累積スコア上位 3 位のキャラクターをデータとして選んだのは、8 月 1 日に穂乃果ちゃんが誕生日だったのと集めたサンプル数ではパートナーデータから相関構造が推定できなかったためです。
集めたデータレイアウトの一部を以下に示します。

図 2 ラブライブランキングデータ

no	順位	lv	パートナー	1位	2位	3位
1	1	643	真姫	真姫	花陽	穂乃果
2	2	1042	ことり	ことり	海未	穂乃果
3	3	501	穂乃果	穂乃果	にこ	海未
4	4	354	善子	真姫	希	穂乃果
5	5	291	ダイヤ	にこ	花陽	真姫
6	6	820	穂乃果	穂乃果	ことり	海未
7	7	773	千歌	穂乃果	真姫	ことり
8	8	486	真姫	真姫	ことり	穂乃果
9	9	581	凛	いるか	凛	海未
10	10	664	梨子	凛	梨子	穂乃果
11	11	433	海未	海未	ことり	穂乃果
12	12	648	ことり	ことり	穂乃果	海未
13	13	602	文絵	穂乃果	絵里	ことり
14	14	474	穂乃果	穂乃果	海未	真姫
15	15	1004	ことり	ことり	凛	穂乃果
16	16	913	にこ	にこ	真姫	穂乃果
17	17	1026	凛	凛	花陽	真姫
18	18	1020	真姫	真姫	理亜	にこ
19	19	1039	凛	凛	真姫	ことり
20	20	604	穂乃果	穂乃果	海未	ことり

データの単純集計は以下の通りです。Aqours で選択回数が少ないキャラクターは Aqours として一つにまとめています。すみません。

このスクフェスデータによるとことりがもっとも 1 位に選ばれた回数が多いようです。(このランキング結果は正直予想していませんでした。また Aqours は不利なことに注意してください。)

	1位	2位	3位	パートナー
穂乃果	224	293	254	323
ことり	235	227	211	185
海未	224	204	211	155
凛	132	65	104	117
花陽	148	164	155	106
真姫	171	216	194	130
にこ	129	108	156	84
希	130	78	85	86
絵里	129	128	135	81
千歌	12	26	23	58
曜	13	23	19	70
梨子	17	22	20	29
花丸	13	12	15	23
善子	19	21	15	94
Aqours	25	33	28	78
モブ	9	10	5	11

	1位	2位	3位	パートナー
穂乃果	13.7%	18.0%	15.6%	19.8%
ことり	14.4%	13.9%	12.9%	11.3%
海未	13.7%	12.5%	12.9%	9.5%
凛	8.1%	4.0%	6.4%	7.2%
花陽	9.1%	10.1%	9.5%	6.5%
真姫	10.5%	13.3%	11.9%	8.0%
にこ	7.9%	6.6%	9.6%	5.2%
希	8.0%	4.8%	5.2%	5.3%
絵里	7.9%	7.9%	8.3%	5.0%
千歌	0.7%	1.6%	1.4%	3.6%
曜	0.8%	1.4%	1.2%	4.3%
梨子	1.0%	1.3%	1.2%	1.8%
花丸	0.8%	0.7%	0.9%	1.4%
善子	1.2%	1.3%	0.9%	5.8%
Aqours	1.5%	2.0%	1.7%	4.8%
モブ	0.6%	0.6%	0.3%	0.7%

図 3 メンバーごとの選択順位およびパートナーでの選択数の集計(n=1628)

このデータから人気と相関構造を探るために離散選択モデルを用います。離散選択モデルでは以下のような効用関数を用います

$$U_{ij} = V_{ij} + \varepsilon_{ij} \quad \cdots ①$$

$$V_{ij} = \beta_{0j} + Z_{ij}^{1}\beta_1 + Z_{ij}^{2}\beta_2 + \cdots + Z_{ij}^{p}\beta_p \quad \cdots ②$$

U は効用値を表し、U は V(効用の確定項)とε(確率的変動項)の和として表現されます。さらに V は Z(説明変数)とβ(パラメータ)の線形結合として分解できます。βは説明変数に対する影響度を表すパラメータでこの中の切片項のパラメータ(β_{0j} の部分)の高低を人気度の指標として利用します。(マーケティングサイエンスの分野ではこの切片が価格やプロモーション要素を取り除いた時のブランド力と定義することが多いです。)

離散選択モデルを扱う実務でもっとも使われるのは以下の数式で表される多項ロジットモデルです。

$$\Pr(y_i = j) = \frac{\exp(V_{ij})}{\exp(V_{i1}) + \cdots + \exp(V_{iJ})} \quad \cdots ③$$

このモデルはある選択肢の選択確率 Pr が 0～1 の間に、Pr の全選択肢の確率の和が 1 になることを満たしています。このモデルの欠点は①で表したεがお互いの選択肢で独立と仮定しており、相関が表現でき

ません。つまり相関が 0 に固定されていますので、I.I.A 特性が成り立たたない選択を表現できません。また今回のデータのように 1 位 2 位 3 位のような複数選ぶ選択に対してはこのモデルでは無力です。このモデルでこのようなランキング集合をモデル化しようと思えば、以下のモデルのように表現することが出来ます。例えば選択肢の集合ににこ、真姫、ことり、穂乃果がある場合に 1 位にこ、2 位真姫、3 位ことり、4 位穂乃果を選んだ場合は

$$\mathrm{Pr}(\mathrm{rank}=\mathrm{n},\mathrm{m},\mathrm{k},\mathrm{h}) = \frac{\exp(\mathrm{V}_{in})}{\sum_{j=n,m,k,h}\exp(\mathrm{V}_{ij})}\frac{\exp(\mathrm{V}_{im})}{\sum_{j=m,k,h}\exp(\mathrm{V}_{ij})}\frac{\exp(\mathrm{V}_{ik})}{\sum_{j=k,h}\exp(\mathrm{V}_{ij})}$$

となります。

このタイプの数式で表された離散選択モデルの①で表された確率変動項が極値分布という確率分布に従うと仮定しています。このタイプのモデル(確率変動項が極値分布に従うロジット型モデル)で選択肢間に相関を持たせるために様々なモデルが例えば提案されています。例えば Nested Logit model、Paired Combinatorial Logit model、Cross Nested Logit model、Generalized Nested Logit model、Mixed Logit model などがあります。

もう 1 つのタイプの離散選択モデルとして①の効用関数の確率変動項が極値分布ではなく正規分布に従うと仮定したプロビットモデルが考えられます。今回の解析ではロジットタイプのモデルではなくこのプロビットタイプのモデルを推定に用います。

多肢選択の場合、確率変動項が以下の多変量正規分布に従います。

$$f(x) = (2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \quad \cdots ④$$

多変量正規分布に従う場合、2 節で示した通りΣの分散共分散行列で選択肢間の様々な相関関係を柔軟に表現することができます。ただしこのモデルで選択確率を表現すれば、以下のような多重積分を解かなければなりません。(積分数は選択肢数-1 次元になり、今回のラブライブ解析の場合、15 次元の多重積分となります。)

$$\mathrm{Pr}(\mathrm{y}_i=\mathrm{j}) = \int_{\mathrm{R}_\mathrm{i}^{\mathrm{J}-1}}(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{u_i}-\mathrm{Z_i}\,\boldsymbol{\beta})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{u_i}-\mathrm{Z_i}\,\boldsymbol{\beta})\right)\mathrm{d}\boldsymbol{u_i} \quad \cdots ⑤$$

$\mathrm{Z_i}\,\boldsymbol{\beta}$は②で表される効用の確定項で、u は効用をΣは相関構造を表しています。また積分記号の右下についている積分領域 R は以下の条件を示します。

$$\mathrm{R}_{\mathrm{ij}}^{\mathrm{J}-1} = \begin{cases} \text{選択肢 j を選んだ時、他の選択肢 k に対して、} \mathrm{u_{ij}} > \mathrm{u_{ik}} \text{かつ} \mathrm{u_{ij}} > 0 \text{ の領域} \\ \text{選択肢 J を選んだ場合、すべての k に対して、} \mathrm{u_{ik}} < 0 \end{cases} \quad \cdots ⑥$$

要するにこの条件は、選ばれた選択肢 j の効用がすべての選択集合の中でもっとも大きくなければならないという条件を表しています。なおプロビットモデルでは選択肢 J を基準選択肢として、他の選択肢 j と J

の相対的な効用値を効用関数として用います。
⑥で表される条件は選択集合の中から 1 つだけを選ぶ際の条件なので、これをランキングデータに適用する際には積分領域 R は以下のような条件に変わります。例えば、3 位までランク付けをした場合

$$R_{ij}^{J-1} = \begin{cases} 選択肢\ j1\ を\ 1\ 位に選んだ時、他の選択肢\ k\ に対して、u_{ij1} > u_{ij2}かつu_{ij1} > 0\ の領域 \\ 選択肢\ j2\ を\ 2\ 位に選んだ時、他の選択肢\ k\ に対して、u_{ij1} > u_{ij2} > u_{ij3}かつu_{ij2} > 0\ の領域 \\ 選択肢\ j3\ を\ 3\ 位に選んだ時、他の選択肢\ k\ に対して、u_{ij2} > u_{ij3} > u_{ik}かつu_{ij3} > 0\ の領域 \\ 選択肢\ J\ を\ 1\ 位に選んだ場合、すべての\ k\ に対して、u_{ik} < 0 \\ 選択肢\ J\ を\ x+1\ 位に選んだ場合、x\ 位以外の選択肢集合\ k\ に対して、u_{ik} < 0 < u_{ix} \end{cases} \quad \cdots ⑦$$

となります。この条件は選択肢に対する効用値が選ばれたランキングの順序どおりになることを要請しています。例えば図 2 のデータの場合、no1 のユーザーは 1 位真姫、2 位花陽、3 位穂乃果を選んでいるので、効用の順位は真姫>花陽>穂乃果という順になります。

スクフェスランキングデータを⑤のモデルを⑦の条件で解くためには多重積分が必要ですが、それは困難なので、データ拡大法に基づくマルコフ連鎖モンテカルロ法(MCMC)でベイズ推定を行うことで推定します。データ拡大法と呼ばれる手法は選択結果と整合的な⑦の条件に基づく潜在効用を多変量正規分布の乱数から大量に発生させることで多重積分を回避します。このデータ拡大法を用いることで効用 u に対する積分を u の乱数に置き換えることで連続変数になる、積分を回避して推定すべきパラメータβおよびΣは多変量線形回帰モデルを解くことに帰着することになります。今回のこのモデルを多項型のランクプロビット と呼ぶことにします。

さらに今回のモデルでは発生させた潜在効用 u を用いて、メンバー間の関係を因子分析モデルにより表現するモデルを付け加えています。(ただ今回はこの因子分析モデルは MCMC でベイズ推定がうまくいかず、ラ４ンクプロビットを推定した後に事後的に推定されたΣを用いて最尤法で因子分析を実施しています。)
なお推定に用いた R コードは Appendix を参照してください。

4．モデルに用いた説明変数

今回のランクプロビットモデルのデータには図 2 のスクフェスランキングデータを用います。選択集合となるメンバーは以下の 15 名+モブとなります。
穂乃果、ことり、海未、凛、花陽、真姫、にこ、希、絵里、千歌、曜、花丸、善子(ヨハネ)、Aqours
なおルビィ、鞠莉、果南、ダイヤは出現数が少なく、推定が難しいと感じられたため、Aqours としてひとつにまとめています。
今回の目的の一つに②のβ_{0j}で示されるメンバーごとの切片項がどのような順序になっているかを知ることです。その選択結果の集計グラフは以下に示します。(n=1628)

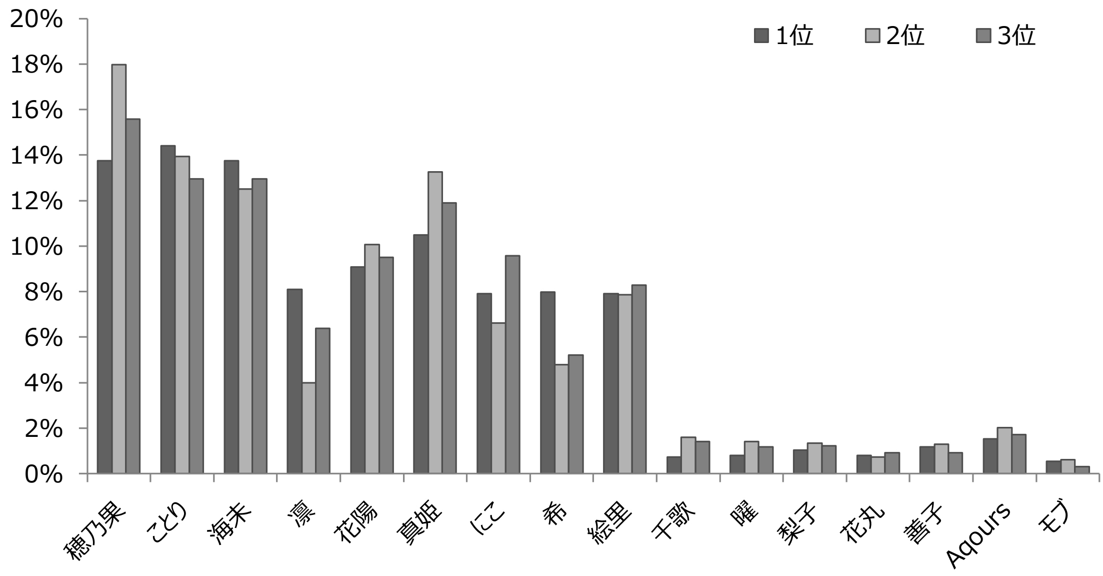


図 4 選択結果のランキング集計(n=1628)

説明変数としてもう 1 つレベル図 2 にある lv(レベル)を用います。おそらくスクフェスのレベルはスクフェスの開始時期と関連があるので、レベルを開始時期の代理変数として用います。この説明変数のパラメータβの大小はスクフェス加入時期により、好むメンバーは異なるのか？を説明します。なおこのレベルの変数はスケールが大きいので標準化を行います。

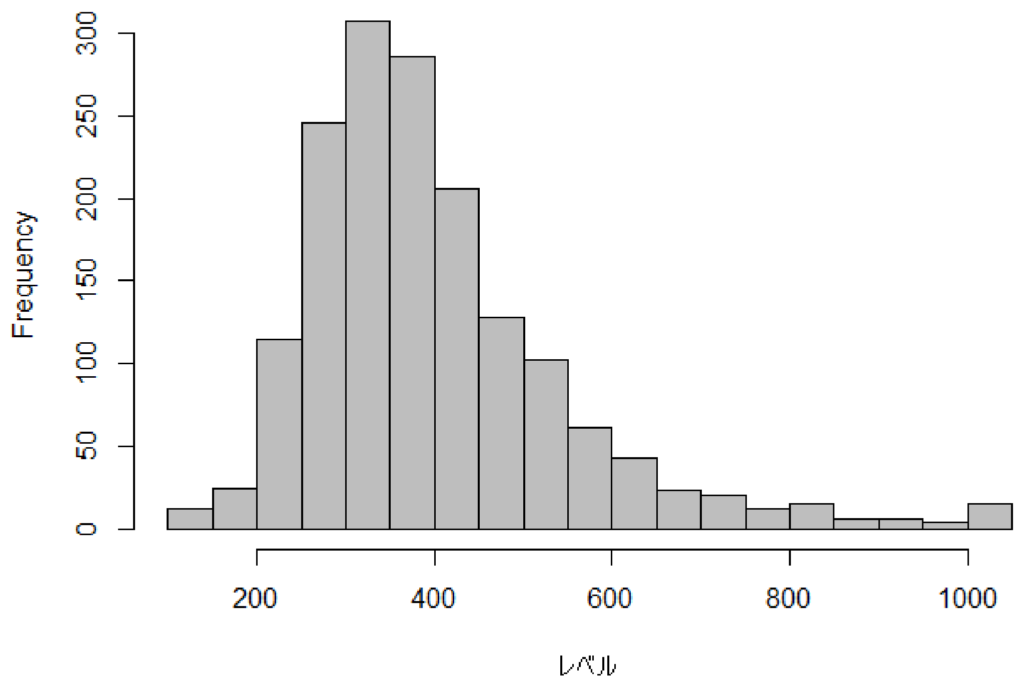


図 5 スクフェスプレイヤーの上位 1628 位までのレベル分布

標準化したレベルと元のレベルの対応関係は以下の通りです。これを見るとレベルが平均値より低い人が半数以上を占めていることが分かります。

またメンバーごとのレベルと 1 位選択の関係を箱ひげ図として図 7 に示します。これを見ると凛、モブあたりがレベルが高いと選ばれる傾向があるように見られます。

	元のレベル	標準化後のレベル
最小値	109	-1.967
25%分位点	303	-0.6544
中央値	367	-0.2176
平均値	399.6	0
75%分位点	459	0.4062
最大値	1042	4.3591

図 6 レベルと標準化レベルの対応表

no	順位	lv	パートナー	1位	2位	3位
1	1	643	真姫	真姫	花陽	穂乃果
2	2	1042	ことり	ことり	海未	穂乃果
3	3	501	穂乃果	穂乃果	にこ	海未
4	4	354	善子	真姫	希	穂乃果
5	5	291	ダイヤ	にこ	花陽	真姫
6	6	820	穂乃果	穂乃果	ことり	海未
7	7	773	千歌	穂乃果	真姫	ことり
8	8	486	真姫	真姫	ことり	穂乃果
9	9	581	凛	いるか	凛	海未
10	10	664	梨子	凛	梨子	穂乃果
11	11	433	海未	海未	ことり	穂乃果
12	12	648	ことり	ことり	穂乃果	海未
13	13	602	文絵	穂乃果	絵里	ことり
14	14	474	穂乃果	穂乃果	海未	真姫
15	15	1004	ことり	ことり	凛	穂乃果
16	16	913	にこ	にこ	真姫	穂乃果
17	17	1026	凛	凛	花陽	真姫
18	18	1020	真姫	真姫	理亜	にこ
19	19	1039	凛	凛	真姫	ことり
20	20	604	穂乃果	穂乃果	海未	ことり

図 2 再掲

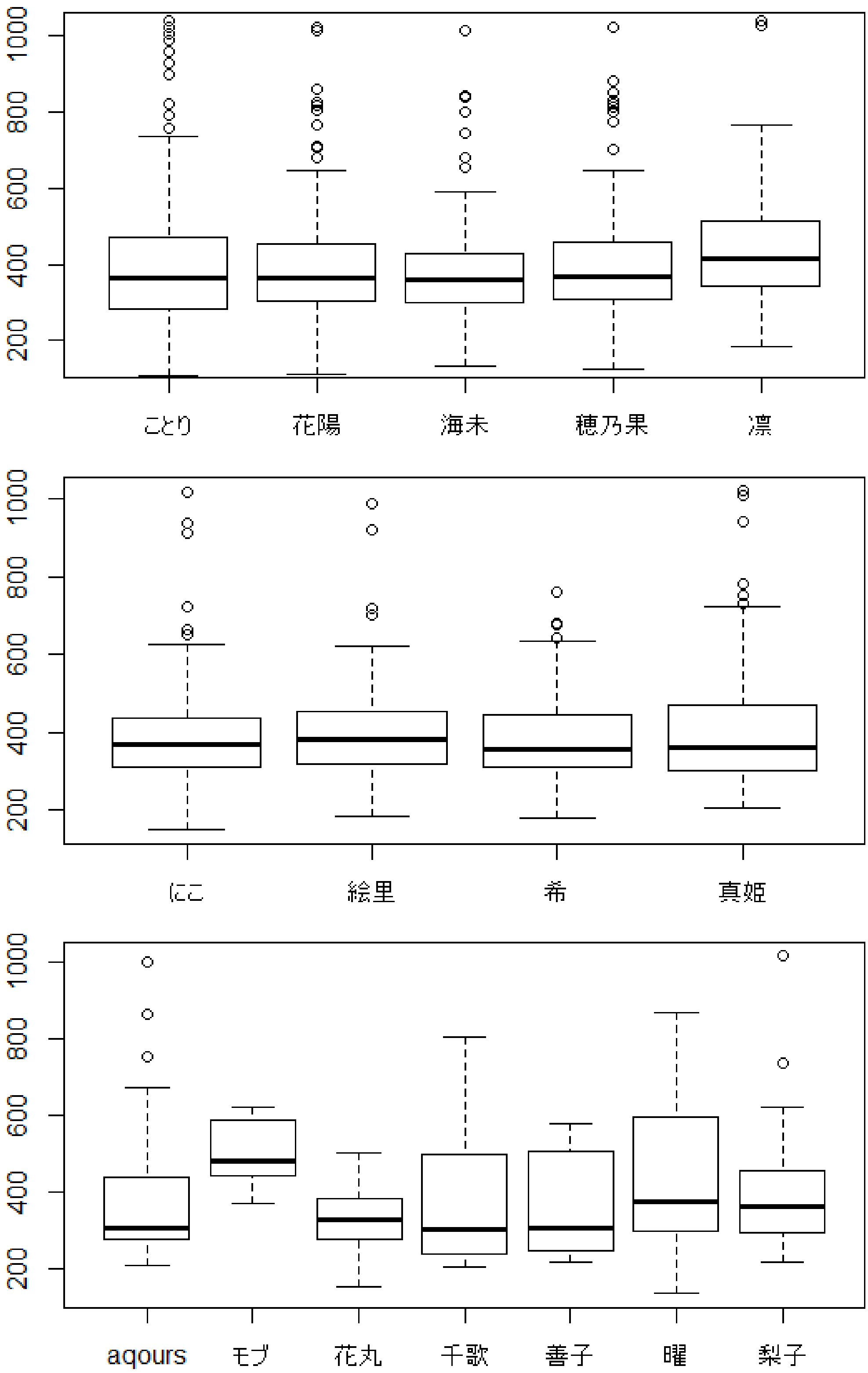


図 7 メンバーごとの 1 位選択とレベルの関係(y 軸=レベル、x 軸=メンバー)

以上の説明変数を用い、あるユーザーi の効用関数 U をメンバーごとに行列表現で以下のように表現します。

$$\begin{bmatrix} U_{i,\text{穂乃果}} \\ U_{i,\text{ことり}} \\ \vdots \\ U_{i,\text{Aqours}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & Lv_i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & Lv_i & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & Lv_i \end{bmatrix} \begin{bmatrix} \beta_{0\ \text{穂乃果}} \\ \beta_{0\ \text{ことり}} \\ \vdots \\ \beta_{0\text{Aqours}} \\ \beta_{lv\ \text{穂乃果}} \\ \beta_{lv\ \text{ことり}} \\ \vdots \\ \beta_{lv\text{Aqours}} \end{bmatrix} + \begin{bmatrix} \varepsilon_{i\ \text{穂乃果}} \\ \varepsilon_{i\ \text{ことり}} \\ \vdots \\ \varepsilon_{i\text{Aqours}} \end{bmatrix} \quad \cdots ⑧$$

5. モデルの推定結果と解釈

ランクプロビットモデルをマルコフ連鎖モンテカルロ法でパラメータを 40,000 回サンプリングし、10,000 回までを burnin 期間として捨て、残りの 30,000 回のサンプリングで推定結果としました。なお、サンプリングは 4 回に 1 回だけ抽出しました。そのサンプリング結果のプロットの一部を以下に示します。グラフの y 軸はパラメータ推定値、x 軸はサンプリング回数を表します。

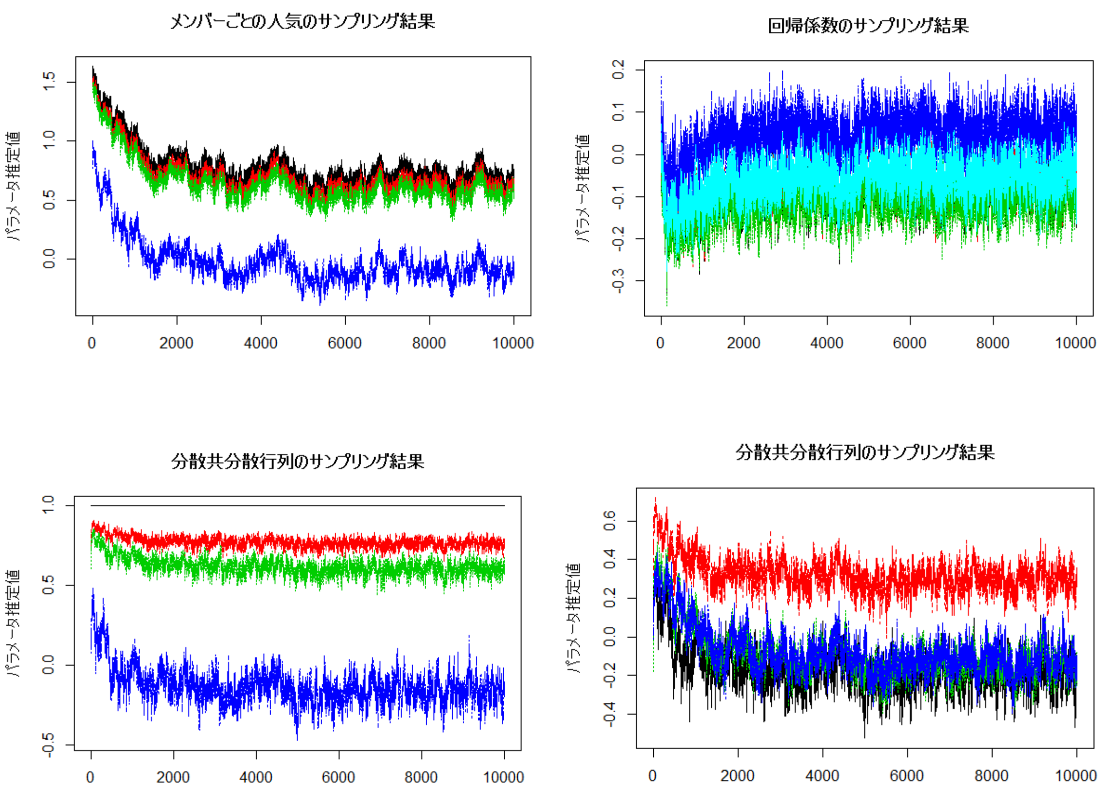


図 8 MCMC サンプリング結果のプロット

まずは LV の説明変数のメンバーごとのパラメータ推定値を見てみましょう。結果は以下の通りです。

メンバー	事後平均	5%分位点	95%分位点	事後標準偏差
穂乃果	−0.089	−0.15	−0.03	0.04
ことり	−0.075	−0.14	−0.01	0.04
海未	−0.114	−0.18	−0.05	0.04
凛	0.056	−0.01	0.12	0.04
花陽	−0.054	−0.12	0.01	0.04
真姫	−0.111	−0.18	−0.05	0.04
にこ	−0.21	−0.28	−0.14	0.04
希	−0.224	−0.3	−0.15	0.05
絵里	−0.178	−0.25	−0.11	0.04
千歌	−0.095	−0.19	0	0.06
曜	−0.059	−0.15	0.03	0.06
梨子	−0.033	−0.13	0.05	0.05
花丸	−0.053	−0.15	0.05	0.06
善子	−0.059	−0.16	0.03	0.06
Aqours	−0.148	−0.24	−0.05	0.06

図 9 Lv の説明変数のメンバーごとのパラメータ推定値

レベルのパラメータの事後平均について見てみると、ほぼ全員のメンバーがマイナスの数値が出ており、レベルが低いほどそのメンバーが選ばれやすいことを示しています。特に値が小さいのはにこ、希、絵里の３年生組とAqoursです。海未、真姫もやや低くなっています。レベルをスクフェス参入時期の代理変数と捉えるならば、より新規のユーザーほどこれらのメンバーを選びやすくなります。
反対にパラメータが正の値であるのは凛のみとなっており、凛を選ぶユーザーは他のメンバーを選ぶユーザーと比較してややレベルが高い傾向にあるようです。

次にメンバーごとの切片のパラメータを事後平均を見ていきます。推定結果は以下の通りです。

メンバー	事後平均	5%分位点	95%分位点	事後標準偏差	1位選択数
穂乃果	0.731	0.6	0.87	0.08	224
ことり	0.643	0.51	0.79	0.08	235
海未	0.574	0.44	0.72	0.08	224
凛	−0.088	−0.24	0.08	0.1	132
花陽	0.318	0.17	0.48	0.09	148
真姫	0.442	0.31	0.6	0.09	171
にこ	0.136	−0.01	0.3	0.09	129
希	−0.114	−0.26	0.06	0.1	130
絵里	0.12	−0.02	0.29	0.09	129
千歌	−0.859	−1.08	−0.66	0.13	12
曜	−0.881	−1.07	−0.68	0.12	13
梨子	−0.981	−1.2	−0.77	0.13	17
花丸	−1.195	−1.43	−0.96	0.14	13
善子	−0.955	−1.18	−0.74	0.13	19
Aqours	−0.838	−1.02	−0.64	0.12	25

図 10 切片(人気度)のメンバーごとのパラメータ推定値

この切片のパラメータの解釈はメンバー選択のレベルの要素を取り除いた時のメンバー選択の影響を表したパラメータと解釈できます。それを踏まえて事後平均の推定値を見てみると、μ’ｓの中で高い順に並べると穂乃果、ことり、海未、真姫、花陽、にこ、絵里、凛、希の順になっています。

3 年生組のパラメータ値が低くなっていますが、3 年生組はレベルが低いほど選ばれる確率が高くなる傾向があったので、比較的新規のユーザーに絞って見ると選ばれやすいということが出来ます。2 年生組は 3 人ともに高い推定値になっており、また 1 位選択数と比較するとパラメータの順序がことりと穂乃果が逆転していることが見て取れ、穂乃果と海未とも差が開いていることが見て取れます。この理由はメンバー間の選択に I.I.A 特性が満たされておらず、ラブライブのメンバー選択に何らかの構造をあることを示しています。その構造についての詳細は後述します。

Aqours 間の比較では全体的にパラメータの事後平均が低いですが、これは Aqours の参入が遅かったためでμ’ｓより不利になるためです。Aqours はμ’ｓと比較してメンバー間でパラメータの事後平均がフラットな傾向があります。また、1 位選択数がヨハネがもっとも多いですが、パラメータはヨハネよりちかようのほうが高くなっています。

μ’ｓのメンバー間のパラメータの事後信頼区間を以下にプロットします。このグラフはメンバー間でパラメータ推定値が重なっていなければそのメンバー間の人気度が統計的に見て明確に差があると見ることが出来ます。

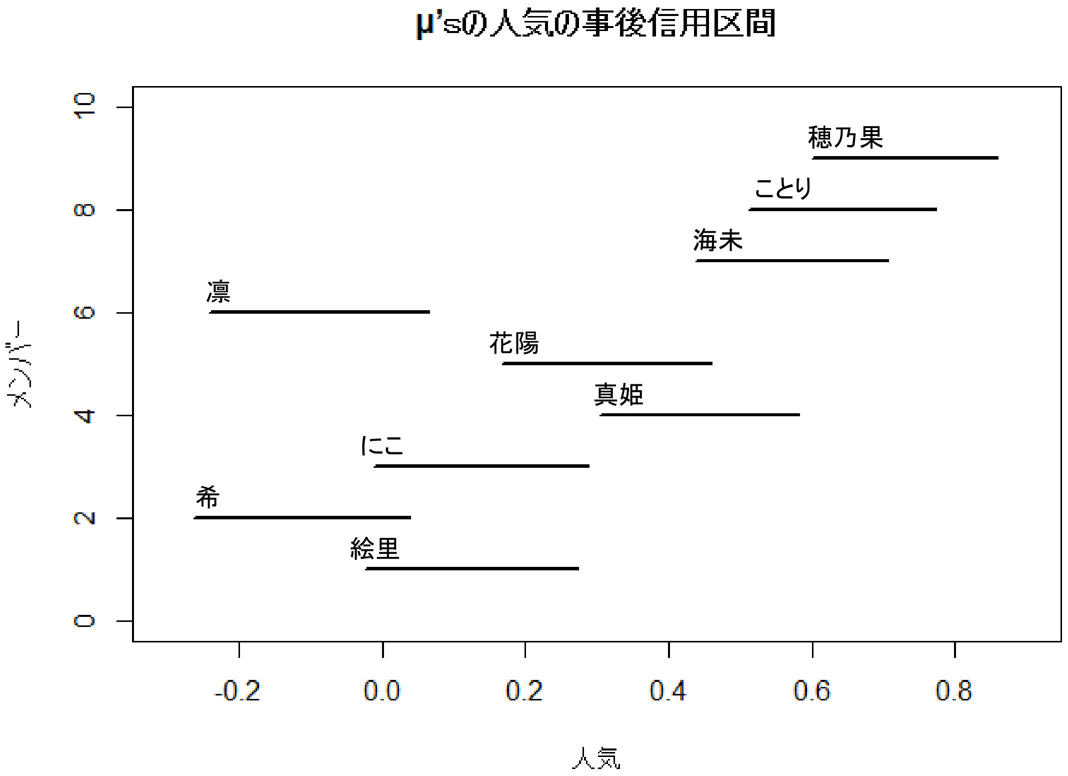


図 11 切片(人気度)のメンバーごとのパラメータの事後信用区間

最後に相関行列のパラメータの事後平均を以下の表に示します。

	穂乃果	ことり	海未	凛	花陽	真姫	にこ	希	絵里	千歌	曜	梨子	花丸	善子	Aqours
穂乃果	1	0.76	0.61	−0.16	0.3	−0.05	−0.1	−0.21	−0.16	0.27	0.2	0.03	−0.02	0.02	−0.04
ことり	0.76	1	0.57	−0.21	0.3	−0.14	−0.12	−0.28	−0.2	0.22	0.23	0.05	0.02	0.09	−0.05
海未	0.61	0.57	1	−0.03	−0.06	−0.21	−0.18	0.03	−0.16	0.12	0.12	0.04	−0.14	−0.08	−0.21
凛	−0.16	−0.21	−0.03	1	0.37	0.25	−0.14	0.08	−0.22	−0.17	−0.05	−0.24	−0.09	−0.22	−0.35
花陽	0.3	0.3	−0.06	0.37	1	0.32	−0.12	−0.35	−0.2	0.07	0.16	−0.12	0.06	0.06	−0.03
真姫	−0.05	−0.14	−0.21	0.25	0.32	1	0.42	−0.03	0.37	−0.15	−0.07	−0.12	−0.03	0.04	−0.02
にこ	−0.1	−0.12	−0.18	−0.14	−0.12	0.42	1	0.28	0.59	−0.25	−0.26	−0.12	−0.13	0	0.06
希	−0.21	−0.28	0.03	0.08	−0.35	−0.03	0.28	1	0.39	−0.52	−0.56	−0.42	−0.42	−0.43	−0.26
絵里	−0.16	−0.2	−0.16	−0.22	−0.2	0.37	0.59	0.39	1	−0.29	−0.22	−0.21	−0.31	−0.12	0.12
千歌	0.27	0.22	0.12	−0.17	0.07	−0.15	−0.25	−0.52	−0.29	1	0.62	0.67	0.38	0.45	0.41
曜	0.2	0.23	0.12	−0.05	0.16	−0.07	−0.26	−0.56	−0.22	0.62	1	0.52	0.37	0.41	0.27
梨子	0.03	0.05	0.04	−0.24	−0.12	−0.12	−0.12	−0.42	−0.21	0.67	0.52	1	0.47	0.58	0.38
花丸	−0.02	0.02	−0.14	−0.09	0.06	−0.03	−0.13	−0.42	−0.31	0.38	0.37	0.47	1	0.45	0.23
善子	0.02	0.09	−0.08	−0.22	0.06	0.04	0	−0.43	−0.12	0.45	0.41	0.58	0.45	1	0.31
Aqours	−0.04	−0.05	−0.21	−0.35	−0.03	−0.02	0.06	−0.26	0.12	0.41	0.27	0.38	0.23	0.31	1

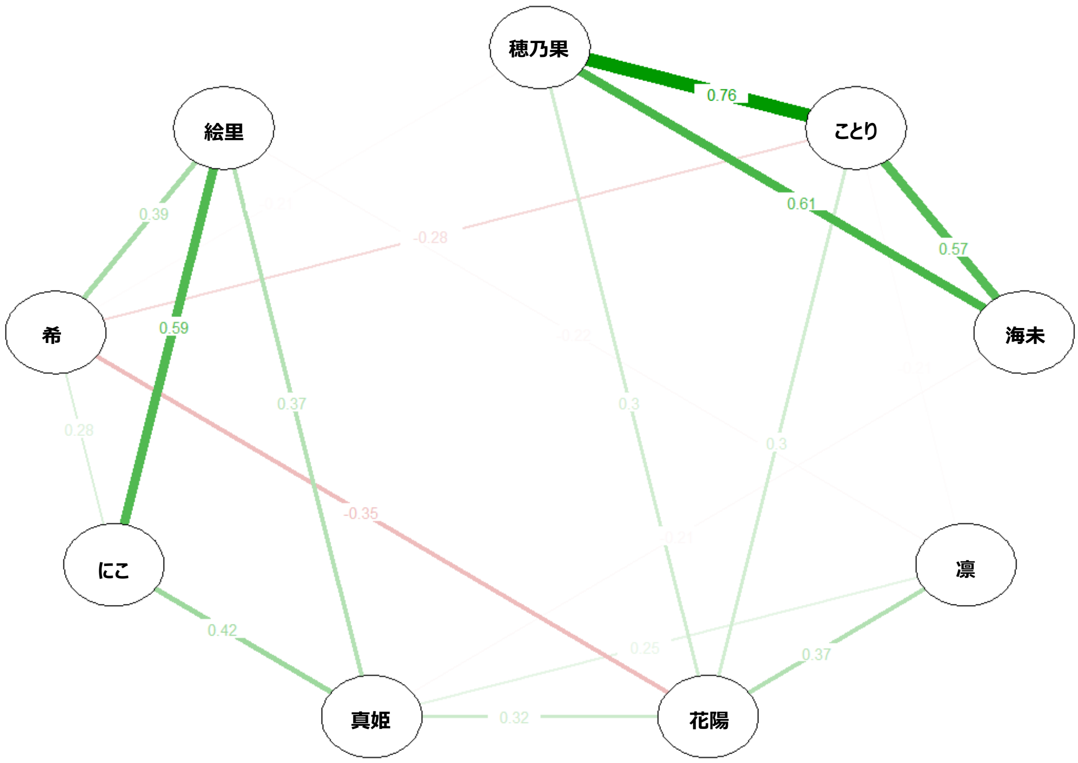


図 13 μ’s メンバーの相関行列の可視化

相関行列を見るといくつかの構造を持っていることが分かります。大きなグループとして、μ’s 1 年生、2 年生、3 年生および、Aqours では全員がお互いに正の相関を持っています。他に大きなグループとして、Printemps、BiBi が正の相関を持ち、そして lily white は特に相関が出ているわけではないようです。μ’s と Aqours 間の相関関係を見ると穂乃果とことりおよび海未しか正の相関を持ちません。そしてこの 2 人は相関の仕方が非常に似ています。

切片のパラメータ推定値が凛と希が低かった理由がこのメンバー間の相関構造で説明できます。その理由はこの 2 人は他のメンバーと正の相関の度合いが低いことにあり、このメンバーの効用が高いと他のメンバーの効用が負の相関があるため低くなり、結果的に低い切片値でも 1 位選択に選ばれる可能性が高くなります。反対に穂乃果、ことり、海未はお互いに正の相関が強いため、つまり 3 人とも好きとなる可能性が高いため(効用値が 3 人同時に高くなるため)、お互いに票を食い合うことになり、3 人それぞれが 1 位に選ばれる確率が相関のない場合と比較して低くなってしまいます。

以下個人的なカップリングに対する考察です。

1. 穂乃果と相関が高いのは海未よりことり。
2. ちかようよりちかりこ。
3. にこまきはカップリングとして比較的妥当。ただ同じ 3 年生の絵里のほうが相関は高い。

4. 独立傾向にある凛は花陽とはやっぱりカップリングとして妥当性がある。
5. ほのことが同じ似た相関構造をもつならようちかも似た相関構造を持っている。
6. 花陽と曜が弱い正の相関を持っている。
7. Aqours でもμ’s のように今後メンバー内でいくつかのサブグループができる相関構造を持つのだろうか？

次節では切片の推定値が全メンバーで同じ場合でこの相関行列が与えられたらどのメンバーがもっとも選ばれるのかをシミュレートします。

さらにこの相関行列から因子分析モデルを推定します。因子数はスクリープロットによ 5 因子を選びました。その結果は以下の通りです。結果を見ると前述したとおりのメンバー間に構造があることが分かります。

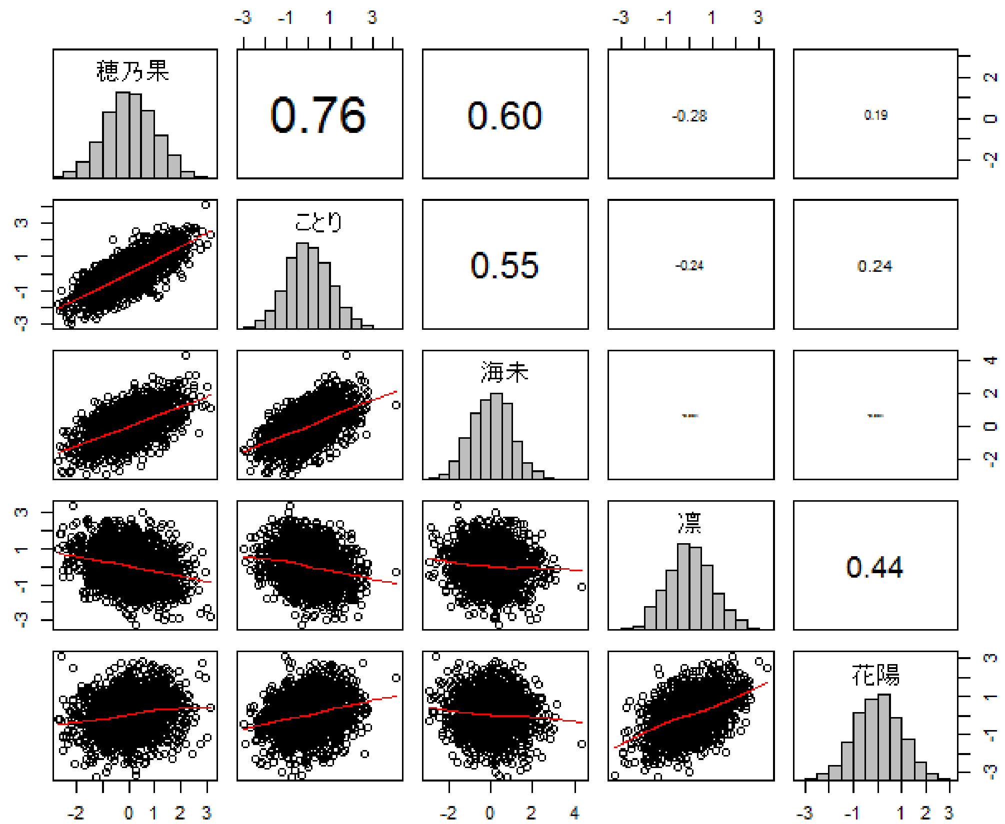
	因子1	因子2	因子3	因子4	因子5	共通性	独自性
穂乃果	0	0.87	0.07	0.23	−0.13	0.79	0.21
ことり	−0.03	0.81	−0.02	0.25	−0.21	0.76	0.244
海未	0.06	0.81	0	−0.28	0.15	0.73	0.268
凛	0.02	−0.18	−0.04	0.22	0.95	0.98	0.023
花陽	−0.1	0.13	0	0.82	0.15	0.71	0.287
真姫	0.09	−0.04	0.64	0.38	0.25	0.56	0.441
にこ	0	0.04	0.74	−0.04	−0.06	0.55	0.445
希	−0.46	−0.03	0.21	−0.39	0.1	0.62	0.381
絵里	−0.09	0.02	0.75	−0.13	−0.13	0.67	0.33
千歌	0.81	0.18	−0.05	−0.07	0.06	0.66	0.336
曜	0.7	0.14	−0.04	0.05	0.12	0.52	0.483
梨子	0.95	0.01	0.05	−0.3	0.08	0.75	0.248
花丸	0.52	−0.16	−0.15	0.06	−0.01	0.37	0.633
善子	0.68	−0.03	0.11	0.01	−0.05	0.47	0.535
aqours	0.44	−0.12	0.15	−0.02	−0.24	0.35	0.652

図 14 因子分析モデルの推定結果

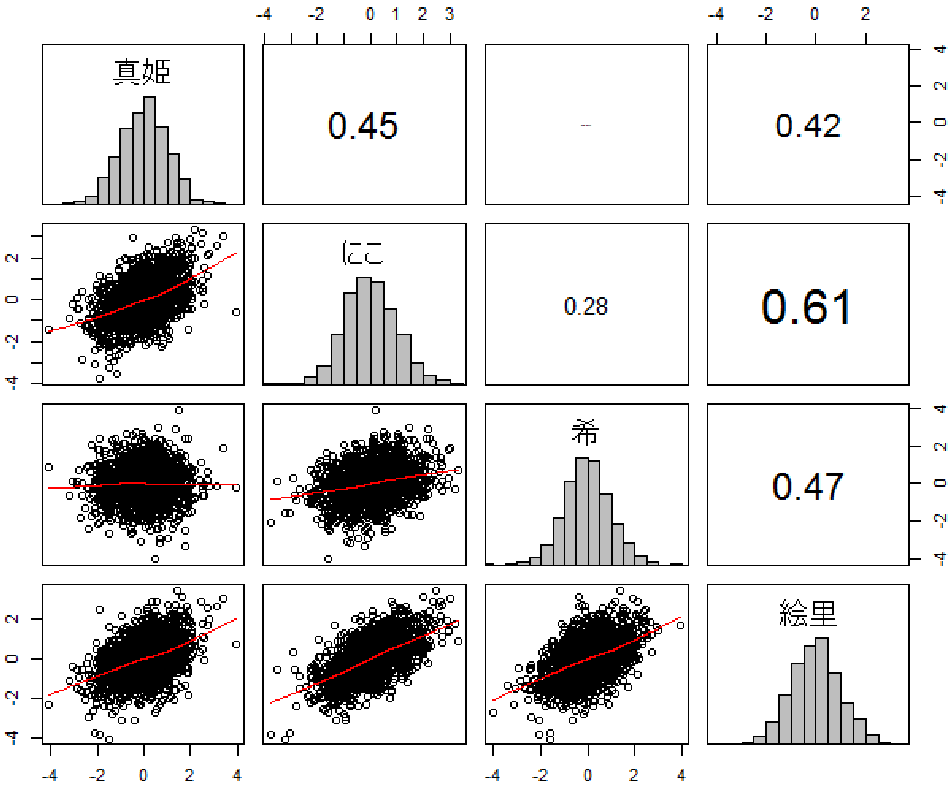
因子に対して名前をつけるとすれば、因子 1 は Aqours 因子、因子 2 はμ’s 2 年生因子、因子 3 は BiBi 因子、因子 4 はμ’s 1 年生因子(花陽型)、因子 5 はμ’s 1 年生因子(凛型)というところでしょうか。なお表中の独自性は 5 つの因子で説明できない部分を表し、共通性は説明できる部分を表しています。独自性が多くのメンバーで高くなっていることから 5 因子で説明できない構造がまだあるかもしれません。

Aqours が 1 つの因子にまとまりμ’s が 4 つのサブグループに分かれているのが見て取れます。Aqours が今後μ’s のようにこのようなサブグループに分かれていくのを見ていくのも面白いでしょう。また他のアニメにもこの傾向が見られるのかが気になります。

以下のグラフは MCMC のあるサンプリング回数における発生させた潜在効用間の散布図行列を示しています。見ての通り、元データのランキングデータが潜在効用に置き換わることにより、離散変数が連続変数でなおかつ正規分布になっていることが見て取れます。



穂乃果
0.76
0.60
-0.28
0.19
ことり
0.55
-0.24
0.24
海未
凛
0.44
花陽



真姫
0.45
0.42
にこ
0.28
0.61
希
0.47
絵里

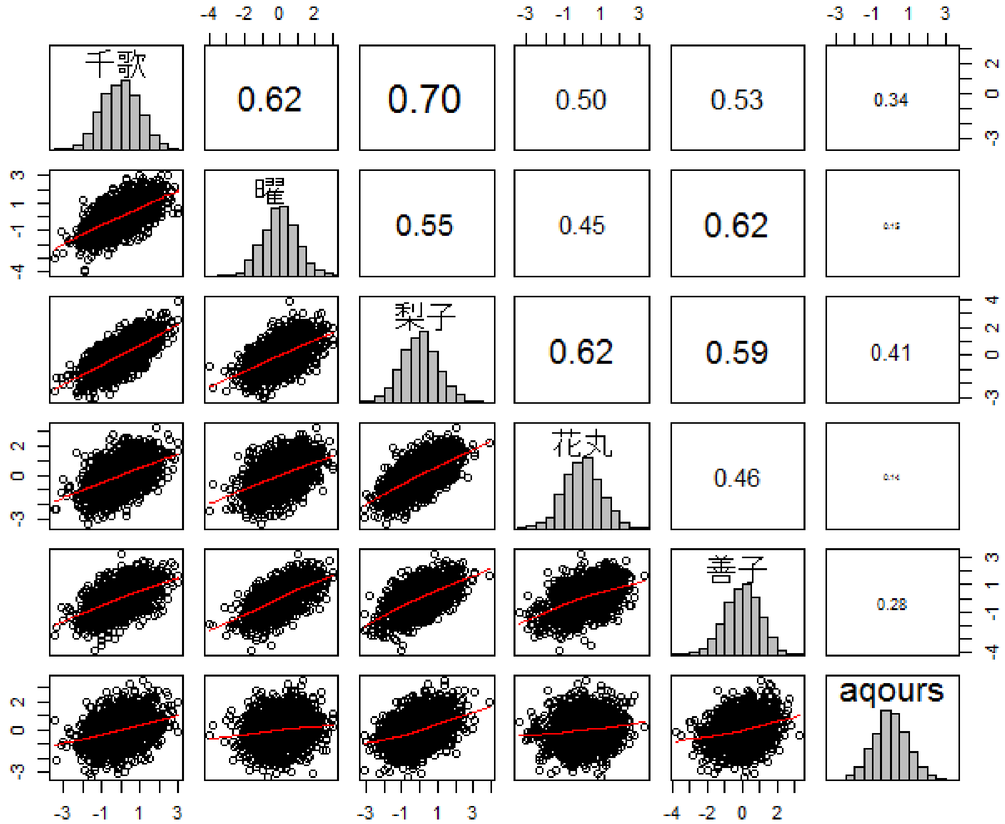


図 15 あるサンプリング回数における潜在効用間の散布図行列

6. 相関行列のシミュレーション

この節では 5 節で与えられた相関行列を用いシミュレーションを行います。パラメータ推定結果から切片とレベルによりメンバーの選択に差があることが分かりました。そこでこれらのパラメータを全メンバーで同じとして、相関行列を与えればどのような選択結果になるのかをシミュレーションで検証します。このシミュレーションの意図として人気度が同じでも I.I.A 特性を満たさなければ選択確率は大幅に変わってしまうことをより深く理解するためです。もし I.I.A 特性を満たせば、パラメータが同じ場合選択確率はすべてのメンバーで同じになるはずです(つまり全員が 1/15=6.67%)。

具体的なシミュレーション手法として切片とレベルのパラメータを全員 0 に固定させ、5 節で推定された相関行列の事後平均を与えた多変量正規分布から 300,000 回乱数を発生させ、メンバーの選択確率を見ます。その結果は以下の通りです。

	相関行列を与えた時の選択確率	互いに独立な場合の選択確率	差
穂乃果	5.2%	6.7%	−1.48%
ことり	5.6%	6.7%	−1.05%
海未	7.1%	6.7%	0.43%
凛	9.0%	6.7%	2.36%
花陽	6.8%	6.7%	0.16%
真姫	6.4%	6.7%	−0.22%
にこ	6.8%	6.7%	0.09%
希	10.3%	6.7%	3.67%
絵里	7.1%	6.7%	0.43%
千歌	4.8%	6.7%	−1.90%
曜	5.5%	6.7%	−1.18%
梨子	5.2%	6.7%	−1.43%
花丸	7.2%	6.7%	0.49%
善子	5.8%	6.7%	−0.82%
Aqours	7.1%	6.7%	0.47%

図 16 相関行列を与えた場合の選択確率のシミュレーション

結果を見ると希、凛が選択確率が大幅に向上し、千歌、梨子、穂乃果、曜、ことりが大幅に選択確率が低下していることが見て取れます。特に千歌は実際に実施されている人気投票の結果が芳しくなく、その一因として、他のメンバーと相関係数が高いことが一因になっているかもしれません。

以上の結果からメンバー間の相関を無視した人気投票は思わぬ結果を生むことが理解できたと思います。

Appendix

今回の解析で用いた R コードを掲載します。

①I.I.A 特性を説明するためのシミュレーションコード

```
library(MASS)

##相関の変化による選択確率の変動をシミュレーションで確認
n <- 500000
rank <- c(0.5, 0.5, 0.5)
cc <- seq(-1, 1, length=101)
Rank.ones <- matrix(0, nrow=n, ncol=length(cc))
Rank.agg <- matrix(0, nrow=length(cc), ncol=length(rank))

for(i in 1:length(cc)){
  print(i)
  CORM <- matrix(c(1, cc[i], 0, cc[i], 1, 0, 0, 0, 1), nrow=3, ncol=3)
  U <- mvrnorm(n, rank, CORM)
  colnames(U) <- c("にこ", "真姫", "ことり")

  first <- apply(U, 1, which.max)
  Rank.ones[, i]  <- ifelse(first==1, "にこ", ifelse(first==2, "真姫", "ことり"))
  Rank.agg[i, ] <- table(Rank.ones[, i])
}

#変数に名前をつける
colnames(Rank.agg) <- c("ことり", "にこ", "真姫")
rownames(Rank.agg) <- round(cc, 2)

#可視化と結果の確認
matplot(cc, Rank.agg/n, type="l", xlab="相関係数", ylab="選択確率", lwd=2, main="メンバー間の相関による選択確率の変化")
round(Rank.agg/n, 3)
```

②ランクデータ前処理およびランクプロビットモデル推定のための R コード

```
#####潜在変数ランクプロビットモデル#####
library(MASS)
library(bayesm)
library(MCMCpack)
library(psych)
library(gtools)
library(MNP)
library(reshape2)
library(dplyr)
library(ggplot2)
library(lattice)
library(qgraph)

####多変量正規乱数を発生させる関数####
##多変量正規分布からの乱数を発生させる
#任意の相関行列を作る関数を定義
corrM <- function(col, lower, upper){
  diag(1, col, col)

  rho <- matrix(runif(col^2, lower, upper), col, col)
  rho[upper.tri(rho)] <- 0
  Sigma <- rho + t(rho)
  diag(Sigma) <- 1
  Sigma
  (X.Sigma <- eigen(Sigma))
  (Lambda <- diag(X.Sigma$values))
  P <- X.Sigma$vector
  P %*% Lambda %*% t(P)

  #新しい相関行列の定義と対角成分を 1 にする
  (Lambda.modified <- ifelse(Lambda < 0, 10e-6, Lambda))
  x.modified <- P %*% Lambda.modified %*% t(P)
  normalization.factor <- matrix(diag(x.modified),nrow = nrow(x.modified),ncol=1)^0.5
  Sigma <- x.modified <- x.modified / (normalization.factor %*% t(normalization.factor))
  eigen(x.modified)
  diag(Sigma) <- 1
  round(Sigma, digits=3)
  return(Sigma)
```

```
}

##相関行列から分散共分散行列を作成する関数を定義
covmatrix <- function(col, corM, lower, upper){
  m <- abs(runif(col, lower, upper))
  c <- matrix(0, col, col)
  for(i in 1:col){
    for(j in 1:col){
      c[i, j] <- sqrt(m[i]) * sqrt(m[j])
    }
  }
  diag(c) <- m
  cc <- c * corM
  #固有値分解で強制的に正定値行列に修正する
  UDU <- eigen(cc)
  val <- UDU$values
  vec <- UDU$vectors
  D <- ifelse(val < 0, val + abs(val) + 0.00001, val)
  covM <- vec %*% diag(D) %*% t(vec)
  data <- list(covM, cc,   m)
  names(data) <- c("covariance", "cc", "mu")
  return(data)
}

####データクレンジング####
##データの読み込み
sf_data <- read.csv("スクフェスデータセット.csv")
hh <- nrow(sf_data)     #サンプル数

#factor 型を文字列に変更
sf_data$X1 位  <- as.character(sf_data$X1 位)
sf_data$X2 位  <- as.character(sf_data$X2 位)
sf_data$X3 位  <- as.character(sf_data$X3 位)
r <- c(sf_data$X1 位, sf_data$X2 位, sf_data$X3 位)

table(c(sf_data$X1 位, sf_data$X2 位, sf_data$X3 位))
unique(c(sf_data$X1 位, sf_data$X2 位, sf_data$X3 位))

##モブと aqours を 1 つの変数にまとめる
r1 <- ifelse(r %in% c("いるか", "姫乃", "咲", "咲良", "ココ", "かさね", "遊宇", "理亜", "紗菜", "あきる", "なな
か",
```

```
                        "咲夜", "仁美", "千鶴子", "小雪", "瑞希", "優理", "クリスティーナ"), "モブ", r)

r2 <- ifelse(r1 %in% c("ダイヤ", "鞠莉", "果南", "ルビィ"), "aqours", r1)
rank <- matrix(r2, nrow=nrow(sf_data), ncol=3)



##モブや aqours が同時に選ばれているサンプルは除去
#aqours の重複を削除
index1 <- subset(1:nrow(rank), rowSums(rank=="aqours") > 1)

if(length(index1)==0) {
  print("重複なし") } else {
    rank <- rank[-index1, ]
    sf_data1 <- sf_data[-index1, ]
  }

#モブの重複を削除
index2 <- subset(1:nrow(rank), rowSums(rank=="モブ") > 1)

if(length(index2)==0) {
  print("重複なし") } else {
    rank <- rank[-index2, ]
    sf_data1 <- sf_data[-index2, ]
  }

##メンバーごとに整理番号をつける
t(t(table(rank)))
Rank <- rank    #新しいランクデータ
lovelive <- c("穂乃果", "ことり", "海未", "凛", "花陽", "真姫", "にこ", "希", "絵里", "千歌", "曜", "梨子",
              "花丸", "善子", "aqours", "モブ")

#メンバーごとに番号をつける
for(i in 1:length(lovelive)){
  Rank[Rank==lovelive[i]] <- i
}

Rank <- apply(Rank, 2, as.numeric)    #文字列になっている番号を数値型に変更
data.frame(Rank, rank)    #データを確認

member <- length(lovelive)    #メンバー数
hh <- nrow(Rank)    #サンプル数
```

```
##説明変数をベクトル形式のデータフォーマットに変更
#ID を設定
id <- rep(1:hh, rep(member-1, hh))
m <- rep(1:(member-1), hh)
ID <- data.frame(no=1:length(id), id=id, m=m)

#切片の設定
p <- c(1, rep(0, member-1))
Pop <- matrix(p, nrow=hh*member, ncol=member-1, byrow=T)
Pop <- subset(Pop, rowSums(Pop) > 0)

#レベルと順位の説明変数の設定
LV <- scale(sf_data1$lv)
SCORE <- -scale(sf_data1$順位)

#ベクトル形式に変更
LV.v <- matrix(0, hh*(member-1), ncol=member-1)
SCORE.v <- matrix(0, hh*(member-1), ncol=member-1)

for(i in 1:hh){
  index.v <- ((i-1)*(member-1)+1):((i-1)*(member-1)+member-1)
  v.lv <- diag(LV[i, ], member-1)
  v.score <- diag(SCORE[i, ], member-1)
  LV.v[index.v, ] <- v.lv
  SCORE.v[index.v, ] <- v.score
}

##メンバー選択とレベルの関係を可視化
index.muse1 <- subset(1:nrow(rank), rank[, 1] %in% c("穂乃果", "ことり", "海未", "凛", "花陽"))
index.muse2 <- subset(1:nrow(rank), rank[, 1] %in% c("真姫", "にこ", "希", "絵里"))
index.aqours <- subset(1:nrow(rank), rank[, 1] %in% c("千歌", "曜", "梨子", "花丸", "善子",
"aqours", "モブ"))

boxplot(sf_data1$lv[index.muse1] ~ rank[index.muse1, 1], ylim=c(140, 1025))
boxplot(sf_data1$lv[index.muse2] ~ rank[index.muse2, 1])
boxplot(sf_data1$lv[index.aqours] ~ rank[index.aqours, 1])

##データを結合
X <- data.frame(pop=Pop, lv=LV.v)
XM <- as.matrix(X)
```

```
####マルコフ連鎖モンテカルロ法でランクプロビットモデルを推定####
####MCMC 推定のための推定準備####

##切断正規分布の乱数を発生させる関数
rtnorm <- function(mu, sigma, a, b){
  FA <- pnorm(a, mu, sigma)
  FB <- pnorm(b, mu, sigma)
  return(qnorm(runif(length(mu))*(FB-FA)+FA, mu, sigma))
}

##多変量正規分布の条件付き期待値と分散を計算する関数
cdMVN <- function(mu, Cov, dependent, U){

  #分散共分散行列のブロック行列を定義
  Cov11 <- Cov[dependent, dependent]
  Cov12 <- Cov[dependent, -dependent]
  Cov21 <- Cov[-dependent, dependent]
  Cov22 <- Cov[-dependent, -dependent]

  #条件付き分散と条件付き平均を計算
  CDinv <- Cov12 %*% solve(Cov22)
  CDmu <- mu[, dependent] + t(CDinv %*% t(U[, -dependent] - mu[, -dependent]))   #条件付
き平均を計算
  CDvar <- Cov11 - Cov12 %*% solve(Cov22) %*% Cov21   #条件付き分散を計算
  val <- list(CDmu=CDmu, CDvar=CDvar)
  return(val)
}

##アルゴリズムの設定
R <- 40000
sbeta <- 1.5
keep <- 4
factors <- 3
llike <- array(0, dim=c(R/keep))   #対数尤度の保存用



##データの設定
#説明変数を多次元配列化
X.array <- array(0, dim=c(member-1, ncol(XM), hh))
for(i in 1:hh){
  X.array[, , i] <- XM[ID[, 2]==i, ]
```

```
}
YX.array <- array(0, dim=c(member-1, ncol(XM)+1, hh))

#ID の設定
id_r <- matrix(1:nrow(XM), nrow=hh, ncol=member-1, byrow=T)

##推定プロセスの格納配列
UM <- matrix(0, nrow=hh, ncol=member-1)
util.M <- matrix(0, nrow=hh, ncol=member-1)

##事前分布の設定
nu <- member     #逆ウィシャート分布の自由度
V <- solve(0.1*diag(member-1))      #逆ウィシャート分布のパラメータ
Deltabar <- rep(0, ncol(XM))   #回帰係数の平均の事前分布
Adelta <- solve(100 * diag(rep(1, ncol(XM))))     #回帰係数の事前分布の分散
inv.facov <- solve(diag(100, factors))
alpha_d <- 1
beta_d <- 100

##サンプリング結果の保存用配列
Util <- array(0, dim=c(hh, member-1, R/keep))
BETA <- matrix(0, nrow=R/keep, ncol=ncol(XM))
SIGMA <- matrix(0, nrow=R/keep, ncol=(member-1)^2)
FA.A <- matrix(0, nrow=R/keep, ncol=(member-1)*factors)
FA.D <- matrix(0, nrow=R/keep, ncol=member-1)
FA.F <- array(0, dim=c(hh, factors, R/keep))



##初期値の設定
#回帰係数の初期値
oldbeta <- c((table(Rank)/sum(table(Rank))*10)[-member], runif(member-1, -0.3, 0.3))

#分散共分散行列の初期値
corM.f <- corrM(col=member-1, lower=0, upper=0)     #相関行列を作成
Sigma.f <- covmatrix(col=member-1, corM=corM.f, lower=1, upper=1)     #分散共分散行列
oldcov <- Sigma.f$covariance

#効用の平均構造の初期値
old.utilm <- matrix(XM %*% oldbeta, nrow=hh, ncol=member-1, byrow=T)

#効用の初期値
```

```
old.util <- old.utilm + mvrnorm(nrow(old.utilm), rep(0, member-1), oldcov)

#因子負荷量と独自因子の初期値
A <- matrix(runif((member-1)*factors, -1, 1), nrow=member-1, ncol=factors)
D <- diag(runif(member-1, 0, 0.5))


####マルコフ連鎖モンテカルロ法で潜在変数ランクプロビットモデルを推定####
for(rp in 1:R){

  ##順位選択結果と整合的な潜在効用を発生させる
  #条件付き期待値と条件付き分散を計算
  S <- rep(0, member-1)

  for(j in 1:(member-1)){
    MVR <- cdMVN(mu=old.utilm, Cov=oldcov, dependent=j, U=old.util)
    UM[, j] <- MVR$CDmu     #条件付き期待値を取り出す
    S[j] <- sqrt(MVR$CDvar)     #条件付き分散を取り出す

    #潜在変数を発生させる
    #切断領域の設定
    rank.u <- t(apply(cbind(old.util[, -j], 0), 1, function(x) sort(x, decreasing=TRUE)))[, 1:3]
    rank.u <- ifelse(Rank==member, 0, rank.u)

    #切断正規分布より潜在変数を発生
    old.util[, j] <- ifelse(Rank[, 1]==j, rtnorm(mu=UM[, j], S[j], a=rank.u[, 1], b=150),
                        ifelse(Rank[, 2]==j, rtnorm(mu=UM[, j], S[j], a=rank.u[, 2],
                              b=rank.u[, 1]),
                               ifelse(Rank[, 3]==j, rtnorm(mu=UM[, j], S[j], a=rank.u[, 3],
                                     b=rank.u[, 2]),
                                      rtnorm(mu=UM[, j], sigma=S[j], a=-150, b=rank.u[,
                                             3]))))
  }

  util.v <- as.numeric(t(old.util))     #発生させた潜在効用をベクトルに置き換える

  ##beta の分布のパラメータの計算と mcmc サンプリング
  #z.vec と X.vec を結合して多次元配列に変更
  YX.bind <- cbind(util.v, XM)
  for(i in 1:hh){
    YX.array[, , i] <- YX.bind[id_r[i, ], ]
```

```
  }

  ##回帰モデルのギブスサンプリングで beta と sigma を推定
  #beta のギブスサンプリング
  invcov <- solve(oldcov)
  xvx.vec <- rowSums(apply(X.array, 3, function(x) t(x) %*% invcov %*% x))
  XVX <- matrix(xvx.vec, nrow=ncol(XM), ncol=ncol(XM), byrow=T)
  XVY <- rowSums(apply(YX.array, 3, function(x) t(x[, -1]) %*% invcov %*% x[, 1]))

  #beta の分布の分散共分散行列のパラメータ
  inv_XVX <- solve(XVX + Adelta)

  #beta の分布の平均パラメータ
  B <- inv_XVX %*% (XVY + Adelta %*% Deltabar)    #beta の平均
  b1 <- as.numeric(B)

  #多変量正規分布から回帰係数をサンプリング
  oldbeta <- mvrnorm(1, b1, inv_XVX)

  ##Cov の分布のパラメータの計算と mcmc サンプリング
  #逆ウィシャート分布のパラメータを計算
  R.error <- matrix(util.v - XM %*% oldbeta, nrow=hh, ncol=member-1, byrow=T)
  IW.R <- V + matrix(rowSums(apply(R.error, 1, function(x) x %*% t(x))), nrow=member-1,
ncol=member-1, byrow=T)

  #逆ウィシャート分布の自由度を計算
  Sn <- nu + hh

  #逆ウィシャート分布から Cov をサンプリング
  Cov_hat <- rwishart(Sn, solve(IW.R))$IW
  oldcov <- cov2cor(Cov_hat)

  ##潜在効用とパラメータを更新
  #潜在効用と潜在効用の平均を更新
  old.utilm <- matrix(XM %*% oldbeta, nrow=hh, ncol=member-1, byrow=T)
  Z <- old.util - old.utilm
  Z <- scale(Z)

  ##潜在効用の誤差項から因子分析モデルを推定
  #多変量正規分布から潜在変数 f(共通因子)をサンプリング
  ADA <- t(A) %*% solve(A %*% t(A) + D)
```

```
  F_mean <- Z %*% t(ADA)    #共通因子の平均
  F_var <- diag(factors) - ADA %*% A    #共通因子の分散共分散行列
  Fi <- t(apply(F_mean, 1, function(x) mvrnorm(1, x, F_var)))    #多変量正規分布から共通因子をサ
ンプリング


  #ガンマ分布から独自因子 d をサンプリング
  Z.error <- Z - Fi %*% t(A)
  Zv.R <- matrix(rowSums(apply(Z.error, 1, function(x) x %*% t(x))), nrow=member-1,
ncol=member-1)

  gamma_alpha <- (hh + alpha_d)/2    #alpha を計算
  gamma_beta <- (diag(Zv.R) + beta_d)/2    #beta を計算
  D <- diag(rgamma(length(gamma_beta), gamma_alpha, gamma_beta))    #ガンマ分布から独自
因子をサンプリング

  #多変量正規分布から因子負荷量 A をサンプリング
  FF <- t(Fi) %*% Fi
  FZ <- t(Fi) %*% Z
  d_sigma <- list()

  for(i in 1:(member-1)){
    d_sigma[[i]]  <- 1/diag(D)[i] * inv.facov
    A_mu <- solve(d_sigma[[i]] + FF) %*% FZ[, i]
    A_cov <- solve(inv.facov + diag(D)[i]*FF)
    A[i, ] <- mvrnorm(1, A_mu, A_cov)
  }


  ##サンプリング結果を保存
  if(rp%%keep==0){
    print(rp)
    mkeep <- rp/keep
    Util[, , mkeep] <- old.util
    BETA[mkeep, ] <- oldbeta
    SIGMA[mkeep, ] <- as.numeric(oldcov)
    FA.A[mkeep, ] <- as.numeric(A)
    FA.D[mkeep, ] <- diag(D)
    FA.F[, , mkeep] <- Fi
    print(round(oldcov, 2))
    print(round(oldbeta[1:member-1], 2))
```

```
      print(round(oldbeta[member:length(oldbeta)], 2))
      print(round(t(A), 2))
   }
}

####推定結果の要約と適合度の確認####
burnin <- 10000/keep     #バーンイン期間

##サンプリング結果を可視化
#回帰係数のプロット
matplot(BETA[, 1:4], type="l", main="メンバーごとの人気のサンプリング結果", ylab="パラメータ推定値")
matplot(BETA[, 5:9], type="l", main="メンバーごとの人気のサンプリング結果", ylab="パラメータ推定値")
matplot(BETA[, 10:11], type="l", main="回帰係数のサンプリング結果", ylab="パラメータ推定値")
matplot(BETA[, 12:15], type="l", main="回帰係数のサンプリング結果", ylab="パラメータ推定値")
matplot(BETA[, 16:20], type="l", main="回帰係数のサンプリング結果", ylab="パラメータ推定値")
matplot(BETA[, 21:24], type="l", main="回帰係数のサンプリング結果", ylab="パラメータ推定値")
matplot(BETA[, 25:29], type="l", main="回帰係数のサンプリング結果", ylab="パラメータ推定値")

#分散供分散行列の可視化
matplot(SIGMA[, 1:4], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 5:9], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 10:13], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 14:18], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 19:22], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 23:27], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 28:31], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 32:36], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 37:40], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 41:45], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 46:49], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 50:54], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 55:58], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 59:63], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 64:67], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 68:72], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 73:76], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(SIGMA[, 77:81], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")



#因子負荷量の可視化
matplot(FA.A[, 1:4], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
```

```
matplot(FA.A[, 5:9], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")
matplot(FA.A[, 10:13], type="l", main="分散共分散行列のサンプリング結果", ylab="パラメータ推定値")


##効用の散布図散布図行列の作成
panel.hist <- function(x, ...){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "grey", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r1 <- cor(x, y)
  r2 <- abs(cor(x, y))
  txt <- format(c(r1, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r2)
}

i <- 8000
Util.Z <- Util[, , i] - matrix(XM %*% BETA[i, ], nrow=hh, ncol=member-1, byrow=T)
colnames(Util.Z) <- lovelive[-member]

#変数 1～5 の散布図行列
pairs(as.data.frame(Util.Z[, 1:5]), panel=panel.smooth, bg="lightblue", diag.panel=panel.hist,
      upper.panel=panel.cor)
#変数 6～9 の散布図行列
pairs(as.data.frame(Util.Z[, 6:9]), panel=panel.smooth, bg="lightblue", diag.panel=panel.hist,
      upper.panel=panel.cor)
#変数 10～12 の散布図行列
pairs(as.data.frame(Util.Z[, 10:15]), panel=panel.smooth, bg="lightblue",
diag.panel=panel.hist,
      upper.panel=panel.cor)
##推定値の事後平均の比較
#beta の要約統計量
```

```
#メンバーの切片の要約推定量
round(beta.m1 <- colMeans(BETA[burnin:nrow(BETA), 1:(member-1)]), 3)     #beta の事後平均
round(beta.q1 <- apply(BETA[burnin:nrow(BETA), 1:(member-1)], 2, function(x) quantile(x,
0.05)), 2)     #5%分位点
round(beta.q2 <- apply(BETA[burnin:nrow(BETA), 1:(member-1)], 2, function(x) quantile(x,
0.95)), 2)     #95%分位点
round(apply(BETA[burnin:nrow(BETA), 1:(member-1)], 2, sd), 2)     #事後標準偏差
beta.q <- cbind(beta.q1, beta.q2)

t(t(table(sf_data$X1 位)))

#人気度の信用区間の可視化
plot(beta.q[1, ], rep(9, 2), type="l", xlab="人気", ylab="メンバー", xlim=c(-0.3, 0.9), ylim=c(0, 10),
lwd=2, main="μ’s の人気の事後信用区間")
lines(beta.q[2, ], rep(8, 2), type="l", lwd=2)
lines(beta.q[3, ], rep(7, 2), type="l", lwd=2)
lines(beta.q[4, ], rep(6, 2), type="l", lwd=2)
lines(beta.q[5, ], rep(5, 2), type="l", lwd=2)
lines(beta.q[6, ], rep(4, 2), type="l", lwd=2)
lines(beta.q[7, ], rep(3, 2), type="l", lwd=2)
lines(beta.q[8, ], rep(2, 2), type="l", lwd=2)
lines(beta.q[9, ], rep(1, 2), type="l", lwd=2)



#説明変数の要約推定量
round(colMeans(BETA[burnin:nrow(BETA), member:ncol(BETA)]), 3)     #beta の事後平均
round(apply(BETA[burnin:nrow(BETA), member:ncol(BETA)], 2, function(x) quantile(x, 0.05)),
2)     #5%分位点
round(apply(BETA[burnin:nrow(BETA), member:ncol(BETA)], 2, function(x) quantile(x, 0.95)),
2)     #95%分位点
round(apply(BETA[burnin:nrow(BETA), member:ncol(BETA)], 2, sd), 2)     #事後標準偏差



#sigma の要約統計量
round(cor.m <- matrix(colMeans(SIGMA[burnin:nrow(SIGMA), ]), nrow=member-1,
ncol=member-1), 2)     #sigma の事後平均
round(matrix(apply(SIGMA[burnin:nrow(SIGMA), ], 2, function(x) quantile(x, 0.05)),
nrow=member-1, ncol=member-1), 2)     #5%分位点
round(matrix(apply(SIGMA[burnin:nrow(SIGMA), ], 2, function(x) quantile(x, 0.95)),
nrow=member-1, ncol=member-1), 2)     #95%分位点
round(matrix(apply(SIGMA[burnin:nrow(SIGMA), ], 2, sd), nrow=member-1, ncol=member-1),
```

```
2) #事後標準偏差



####因子分析モデルで因子構造を推定####
##因子数を決定
i <- 20000/keep
plot(1:(member-1), eigen(cor.m)$values,
        type="l", ylab="固有値", xlab="変数数")
factors <- 5     #因子数は 5

##MCMC サンプリング結果を用いて因子分析モデルを当てはめる
res <- fa(r=cor.m, nfactors=5, fm="ml", rotate="promax", cor="cor", n.obs=hh)
res



####相関行列のシミュレーション####
s <- 300000
lovelive <- c("穂乃果", "ことり", "海未", "凛", "花陽", "真姫", "にこ", "希", "絵里", "千歌", "曜", "梨子",
                      "花丸", "善子", "aqours")

U.simul <- mvrnorm(s, rep(0, member-1), cor.m)
colnames(U.simul) <- lovelive
simul.max <- apply(U.simul, 1, which.max)

simul.res <- table(simul.max)
names(simul.res) <- lovelive
t(t(simul.res/s))     #結果の集計



####相関行列を可視化####
cor.mus <- cor.m[1:9, 1:9]
colnames(cor.mus) <- lovelive[1:9]

qgraph(cor.mus, edge.labels=T, minimum=.2, layout="circle", palette="pastel")
```