

新型コロナウイルスにおけるマスクの着用について

山岡 桃果

(佐藤 嘉倫ゼミ)

本論文の目的

新型コロナウイルス感染症は世界的に流行し、あっという間に人々の生活を一変させた。新型コロナウイルス感染症は2023年4月で流行から約3年経過したが、現在の社会はどのように変化があったのだろうか。

日本政府観光局によると、2023年10月の訪日外客数は、2019年同月比100.8%の2,516,500人となり、新型コロナウイルス感染拡大後初めて2019年同月を超えた(日本政府観光局, 2023)。その一方で、都医師会からは「5類への移行でもう終わったように思っている人もいるが、今は都内だけで毎日、新たに1万5000人ほどが感染しているような状況だ。第9波に入っており、第8波のピークに近づきつつある」との声も挙がっている(NHK NEWS WEB, 2023)。このように、新型コロナウイルス感染症は終息に向かっていると捉えられるような側面もあるが、まだまだ感染への警戒が必要だという意見もあるのが現在の社会である。

本論文では、いま私たちが生きる“コロナ渦”の現代社会をより理解するため、新型コロナウイルス感染症のこれまでの時系列を改めて振り返るとともに、新型コロナウイルス感染症におけるマスクの着用理由と、これまでのマスク着用率の変化をまとめ、どのような要因で着用率が変化していると考えられるのかを検討する。

本論文の構成は、「本論文の目的」、「新型コロナウイルス感染症の時系列変化」、「マスクの着用理由」、「これまでのマスク着用率の変化」、「マスク着用における同調圧力の影響」、「研究方法」、「結果と考察」という全7章で構成する。

新型コロナウイルス感染症の時系列変化

2019年12月以降、中国の湖北省・武漢市で原因となる病原体が特定されていない肺炎の発見が複数報告された。これが新型コロナウイルス感染症(COVID-19)であり、この感染症は2023年現在も、社会に様々な影響を及ぼしている。日本においては、2020年1月15日に初めて新型コロナウイルス感染症の感染者が確認され、約半年後の2020年6月30日には国内の感染者が18,593例、死亡者は972名に上ったと報告されている(厚生労働省, 2020)。

新型コロナウイルス感染症感染拡大を受け、政府は、合計3度にわたり新型コロナウイルス感染症緊急事態宣言(以下「緊急事態宣言」という)を発出したほか、新型コロナウイルス感染症まん延防止等重点措置(以下「まん延防止等重点措置」という)を講じるなど、感染拡大を防止するため、国民に移動を伴う行動の自粛を始めとする感染防止策を呼びかけた。

緊急事態宣言は、令和2年法律第4号による改正後の新型インフルエンザ等対策特別措置法(平成24年法律第31号)の規定に基づき、同法に規定する「新型インフルエンザ等」とみなされた新型コロナウイルス感染症(ただし、令和3年法律第5号により、2021年2月13日以降は、「新型インフルエンザ等」として位置付けられた)に関する緊急事態が発生した旨を宣言したものであり、同宣言が解除されるまでの間、国民の生命及び健康を保護し、並びに国民生活及び国民経済に及ぼす影響が最小となるようにするため、国、地方公共団体並びに指定公共機関及び指定地方公共機関において、新型コロナウイルス感染症の全国的かつ急速なまん延を抑えるための対応として緊急事態措置が実施された。2020年4月7日、埼玉県、千葉県、東京都、神奈川県、大阪府、兵庫

新型コロナウイルスにおけるマスクの着用について

県及び福岡県の7都府県を対象区域として緊急事態宣言が発出されたことを皮切りに、新規陽性者数が最初のピークを迎えた同月から同年5月にかけて、全国47都道府県が同宣言の対象区域となった（以下「第1回緊急事態宣言」という）。その後、2020年中に緊急事態宣言が発出されることはなかったが、2021年1月から東京都を含む47都県を対象区域として緊急事態宣言が発出され、同年3月までの間、最大で11都府県が同宣言の対象区域となった（以下「第2回緊急事態宣言」という）。さらに、2021年4月から東京都、大阪府等の4都府県を対象区域として、またも緊急事態宣言が発出され、同年9月末までの間、最大で21都道府県が同宣言の対象区域となった（以下「第3回緊急事態宣言」という）。

まん延防止等重点措置は、令和3年法律第5号により新型インフルエンザ等対策特別措置法に新設された規定に基づき、国民の生命及び健康を保護し、並びに国民生活及び国民経済に及ぼす影響が最小となるようにするため、国及び地方公共団体が特定地域からの新型コロナウイルス感染症のまん延を抑えるための対応として実施する措置であり、2021年4月から9月までの間に、全国33都道府県において実施された（法務省、2022）。

このように世界的に多大なる影響をもたらした新型コロナウイルス感染症であるが、感染対策として、換気や手洗い・手指消毒、マスクの着用などが呼びかけられていた。中でもマスクの着用は、屋内では距離が確保でき会話をほとんど行わない場合を除き、原則マスクの着用が求められ、マスク不足の現象が起こったことから、政府より布製マスク（通称アベノマスク）の無償配布が行われた。このマスク着用の考え方は、新型コロナウイルス感染症が2023年5月8日に5類感染症に移行することに伴い、2023年3月13日より個人の選択を尊重し、感染対策の実施については個人・事業者の判断が基本となった（厚生労働省、2023）。

マスクの着用理由

マスクの着用は新型コロナウイルス感染症が5類感染症に移行することで個人・事業者の判断が基本となったが、マスク着用についての考え方が新

型コロナウイルス感染症流行前のような状態に戻ったかという点、そうとは言えないのが現状である。そこで本章では、人々がマスクを着用する理由について検討する。

まず、人々がマスクを着用する最も大きな理由として考えられるのは、「感染のリスクに不安を感じる」ということである。ここでは、日本インフォメーション株式会社の2つの調査をもとに説明する。第1の調査は2022年9月22日に発表された「～マスク生活はどこまで続く？～マスク着用の意識・行動調査」で、この調査は2022年8月9日～11日に実施された（日本インフォメーション株式会社、2022）。有効回答数は計983サンプルである。第2の調査は2023年5月28日に発表された「～アフターコロナでどう変わる？～コロナに対する意識・行動調査」で、この調査は2023年4月21日～27日に実施された（日本インフォメーション株式会社、2023）。有効回答数は計1,000サンプルである。両調査とも調査対象は全国の16～69歳の男女、調査方法はクロズドモニターへのインターネットリサーチである。

この2つの調査には共通して、マスクの着用理由を問う質問項目があり、2022年8月実施の調査では前問にて「屋外でマスクを毎日着用している」～「3～5割程度着用している」と回答した人、2023年4月実施の調査では「公共交通機関の中」「職場」「学校」「小～中規模の小売店内」「飲食店」「その他、自宅以外の屋内」「屋外」という場所別で「この場所でマスク着用している」と回答した人が質問対象になっている。結果をみると、2022年8月実施の調査では「屋外でも感染のリスクに不安を感じる」と回答した人が43.9%で第1位（日本インフォメーション株式会社、2022）。2023年4月実施の調査でも、「感染のリスクに不安を感じる」と答えた人は学校、屋外以外の項目で第1位（日本インフォメーション株式会社、2023）。この2つの調査結果から、マスクを着用している人は感染のリスクへの不安を感じている人が多いと言える。

これまでのマスク着用率の変化

前章ではマスクの着用理由について検討したが、本章では、原則マスクの着用が求められてい

た2023年3月までと、5類感染症移行に伴いマスクの着用が個人の選択を尊重するようになった2023年3月以降、実際にマスクの着用率はどのような変化があったのか検討する。

ここでは、株式会社ロイヤリティマーケティングより2023年8月31日に発表された「男女1,000人に聞いたマスク着用に関する調査（第5回）」をもとに説明する。この調査は2023年3月15～16日、同年4月12～13日、同年5月12日～13日、同年6月14～15日、同年7月12～13日、同年8月9～14日に実施されたインターネット調査で、有効回答数は1,000、調査対象は国内在住の20～60代、性年代別に各100サンプル回収されている。この調査によるマスク着用状況をみると、これまで通りマスクを着用していると答えた人が3月は75%、4月は69%、5月は57%、6月は46%、7月は41%、8月は31%、場所やシーンによってはマスクを着用しなくなったと答えた人が3月は20%、4月は25%、5月は31%、6月は38%、7月は39%、8月は42%、ほとんどマスクを着用しなくなったと答えた人が3月は5%、4月は6%、5月は12%、6月は15%、7月は20%、8月は27%となっている（株式会社ロ

イヤリティマーケティング, 2023)。これらの数字をグラフ化すると、下の図1のようになる。

この結果から、5類感染症移行に伴いマスクの着用が個人の判断に委ねられるようになった2023年3月以降、マスクの着用率は徐々に低下していることが分かる。

しかし、原則マスクの着用が求められていた2023年3月までのマスク着用率が、マスクの着用が困難な状態にある人を除き100%だったと仮定し、5類感染症移行に伴いマスクの着用が個人の選択を尊重するようになった直後の3月の調査結果を比べると、マスク着用率が100%から75%と、約3割弱程度しか減っていない。マスクを着用“しなくなった”理由を考えたとき、5類感染症移行に伴いマスクの着用が個人の選択を尊重するようになった直後の調査データである3月時点では「5類感染症移行に伴いマスクの着用が個人の選択を尊重するようになった」ことが理由であると推定できるが、それから1ヶ月以上経った4月以降の調査データでは、マスクの着用をやめるという行動を起こさせる理由が「5類感染症移行に伴いマスクの着用が個人の選択を尊重するようになった」ということ以外にもあると考えられる。

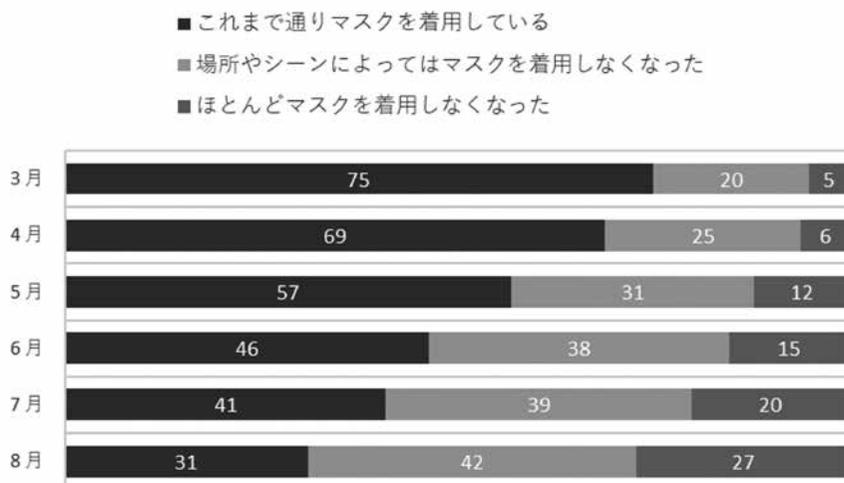


図1 マスクの着用状況
(株式会社ロイヤリティマーケティング, 2023, 調査データに基づいて著者作成)

マスク着用における同調圧力の影響

同調圧力とは

そこで、「5類感染症移行に伴いマスクの着用が個人の選択を尊重するようになった」ということ以外に、マスクの着用をやめるという行動を起こさせる理由として考えられるのが「同調圧力」の影響である。

人は自分を取り巻く状況の変化によって、現在とっている見解や態度を変化させることがある。このような行動の1つに同調圧力がある。これは、他者の見解や態度が自分と異なることを認知し、他者からの圧力を感知して、他者に自分を合わせることであり、慣習的行動、投票行動などにおいてよく見られる(太田・飯田・河岡, 1995)。日本で初めて感染者が確認された2020年以降、2023年現在まで新型コロナウイルス感染症が影響を及ぼしている現代社会において、マスクの着用はこの“慣習的行動”に当てはまると言えるだろう。

日本における同調圧力

本論文でこの同調圧力の影響について着目する理由は、「日本は同調圧力が強い」という考え方が定説化しているからである。同調圧力は日本だけでなく、世界的にも存在するものだが、それでもなぜ日本は同調圧力が強いと言われているのか。日本は同調圧力が強いと考えられている理由は様々あるが、本論文では日本の文化から由来している2点の理由を挙げて説明する。

第1の理由は、「村社会を重視する文化が残っているから」。島国として育んできた文化が同調圧力を高めている理由の1つであり、いわゆる「村社会」がそれに当たる。村社会とは集落を単位とした地域社会のことで、有力者を頂点とした閉鎖的な集団である。江戸時代には村の秩序維持のためルール(掟)が存在し、ルールを破った者は「村八分」という排斥的制裁を受ける。こうした制裁を恐れ、世間体や他人の目を気にする文化が定着したことから、同調圧力が強いということにも繋がっていると考えられている(ourly Mag, 2023)。

第2の理由は、「昔から和の文化が重んじられているから」。村社会の名残に加えて、日本では

昔から“集団の調和を乱さない”という和の文化が重んじられている。集団における協調性や連帯感を育む教育に加え、家族やチーム、コミュニティの絆を大切にしている価値観が重視されてきた。伝統的に周囲と協調し、和を尊ぶ意識が高いと言えるだろう。和の文化を重んじ、他人との衝突を避けようとすることで、同調行動が生まれやすくなり、「和の文化を重視する日本人は同調圧力が強い」といったイメージを持たれやすくと考えられる(株式会社学情, 2023)。

この同調圧力を新型コロナウイルス感染症が流行している社会でいうならば、マスクの着用が個人の自由だったとしても、集団のなかのほとんどがマスクを着用していたら、マスクを外す人は少ないと考えられる。実際、新型コロナウイルス感染症におけるマスクの着用において、同調圧力の影響があると明らかになっている調査データがある。読売新聞オンラインが実施した「2023年1～2月 郵送全国世論調査『新型コロナ3年』」という調査である。調査方法は全国の有権者から無作為に3,000人(250地点、層化2段無作為抽出法)を選び、郵送法で実施した。1月24日に調査票を対象者に郵送し、2月28日までに返送されたのは2,157である。対象者以外による回答などを除くと有効回答は2,090、回答率70%である。この調査に「あなたは、マスクについて、周囲の人が着用していると自分も着用しなければならないと、感じますか、感じませんか」という質問項目がある。結果は「感じる」と答えた人が49%、「どちらかといえば感じる」と答えた人が34%となっている(読売新聞オンライン, 2023)。この調査結果では約8割の人が周囲からの影響を感じており、マスクの着用において、同調圧力の影響はあると言えるだろう。さらに、マスクの着用において同調圧力の影響があるということは、マスクの着用をやめる判断においても、同調圧力の影響があると考えられる。

研究方法

エージェント・ベスト・モデルの構造

本論文では、プログラミング言語Pythonを用いたコンピュータシミュレーション(エージェント

ト・ベースト・モデル)によってマスク着用率の変化について検討する。以下でプログラムの構造を説明する。

- ・エージェントのひな型となるクラス (class) を作る。
- ・クラスにおいてエージェントの属性を3つ定義する。第1の属性は感染のリスクを気にするか気にしないかである。第2の属性は同調圧力を感じやすいか感じにくいである。この2つの属性を組み合わせることで、(感染のリスクを気にしない、同調圧力を感じにくい)、(感染のリスクを気にしない、同調圧力を感じやすい)、(感染のリスクを気にする、同調圧力を感じにくい)、(感染のリスクを気にする、同調圧力を感じやすい)、という4種類のエージェントができる。

第3の属性はマスクを着用するかしないかである。第1と第2の属性が時間的に変化しないのに対し、第3の特性は時間とともに変化しうる。

- ・クラスにおいて初期状態のマスク着用率を決定する。この着用率は、株式会社ロイヤリティマーケティングの「男女1,000人に聞いたマスク着用に関する調査(第5回)」より3月と6月の調査データを引用して設定する。
- ・実際のシミュレーションではエージェントを400人生成する。400人のうち上記の4種類のエージェントの割合は、5類感染症移行前・5類感染症移行直後の3月・5類感染症移行から数ヶ月後の6月、それぞれの社会状況を想定して設定する。
- ・シミュレーションの流れは次の通りである。
 - 1) エージェントをインスタンス化することで、400人のエージェントを生成する。
 - 2) 順序効果を避けるためにエージェントをランダムに並べ替える。
 - 3) 1番目のエージェントから最後のエージェントまで1人ずつ以下の4)から6)までの手順を踏む。
 - 4) 自分の番になったエージェントは自分から半径Zの距離を測り、その範囲を当該エージェントの近傍と定める。今回は半径Zを0.5とする。
 - 5) 同調圧力を感じやすいエージェントは、そ

の近傍の中で10%以上マスクを取っていたらマスクを外す。感じにくいエージェントは90%以上マスクを取っていたらマスクを外す。ただしマスクを取るか取らないかか意思決定において、感染のリスクを感じるタイプのエージェントは同調圧力に関係なくマスクを取らないと仮定する。

- 6) エージェントはランダムに決まった新しい場所へ移動する。これは人々が自由に移動できることを表現している。
- 7) 最後のエージェントの番が終わったら、上の2)に戻り次のステップに入る。そしてステップを200回繰り返す。これで1回のシミュレーションが終了する。
- 8) ランダム性によって結果が大きく変化しないことを確認するために上のシミュレーションを10回実行する。

- ・各シミュレーションにおいて、各回(ステップ)におけるマスク着用率を記録しておく。そしてステップを横軸、マスク着用率を縦軸とするグラフを描く。このグラフによってマスク着用率の時間的変化を見ることができる。

シミュレーションの結果

まずは5類感染症移行前を想定したプログラムを作成する。初期のマスク着用率は、5類感染症移行前は原則マスクの着用が必要であったことから99.9%と設定する。なお、身体障害、知的障害、精神障害(発達障害を含む)のある方については、障害特性によりマスクの着用が困難な状態にある場合もあるため、本研究では100%ではなく99.9%と設定した。

次にエージェントの割合は、感染のリスクを気にしない同調圧力を感じにくいエージェントが1割、感染のリスクを気にしない同調圧力を感じやすいエージェントが1割、感染のリスクを気にする同調圧力を感じにくいエージェントが6割、感染のリスクを気にする同調圧力を感じやすいエージェントが2割と設定する。この割合は、5類感染症移行前であることから、同調圧力よりも何よりも「感染のリスク」が最も重要視されていると想定し、感染のリスクを気にする同調圧力を感じにくいエージェントが半数以上になるよう設定した。

新型コロナウイルスにおけるマスクの着用について

第2に、5類感染症移行直後の3月を想定したプログラムを作成する。初期のマスク着用率は、株式会社ロイヤリティマーケティングの「男女1,000人に聞いたマスク着用に関する調査(第5回)」からマスク着用状況の調査結果(3月)を引用し、75%と設定する。次にエージェントの割合は、感染のリスクを気にしない同調圧力を感じにくいエージェントが2割、感染のリスクを気にしない同調圧力を感じやすいエージェントが3割、感染のリスクを気にする同調圧力を感じにくいエージェントが2割、感染のリスクを気にする同調圧力を感じやすいエージェントが3割と設定する。この割合は、5類感染症移行に伴い、感染のリスクへの危機感が5類感染症移行前よりは減ると想定し、感染のリスクを気にするエージェントと気にしないエージェントの割合を半数ずつに、また5類感染症移行前の設定よりは同調圧力の影響が大きくなると想定し、設定した。

最後に、5類感染症移行から数ヶ月後の6月を想定したプログラムを作成する。初期のマスク着用率は、株式会社ロイヤリティマーケティングの「男女1,000人に聞いたマスク着用に関する調査(第5回)」からマスク着用状況の調査結果(6月)を引用し、46%と設定する。次にエージェントの割合は、感染のリスクを気にしない同調圧力を感じにくいエージェントが2割、感染のリスクを気にしない同調圧力を感じやすいエージェントが5割、感染のリスクを気にする同調圧力を感じにくいエージェントが1割、感染のリスクを気にする同調圧力を感じやすいエージェントが2割と設定する。この割合は、5類感染症移行から数ヶ月経っているということで、5類感染症移行直後の3月よりも感染のリスクへの危機感はさらに減り、同調圧力の影響もさらに強くなっていると想定し、設定した。

各プログラムの結果は図2、図3、図4の通りである。

グラフを見ると、5類感染症移行前を想定したプログラム(図2)ではマスク着用率の大きな変化は見られないが、5類感染症移行直後の3月を想定したプログラム(図3)と5類感染症移行から数ヶ月後の6月を想定したプログラム(図4)ではシミュレーションのスタート地点と200回のステップが終わった地点で大きな変化がみられる。

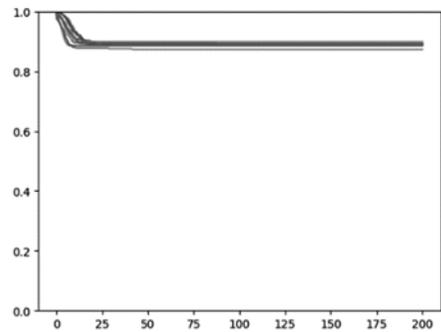


図2 5類感染症移行前

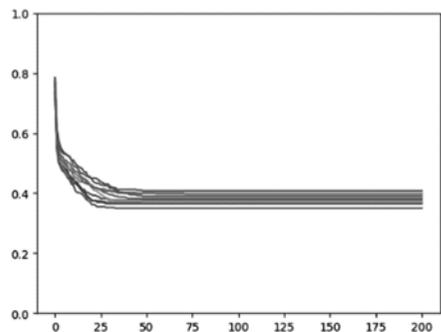


図3 5類感染症移行直後

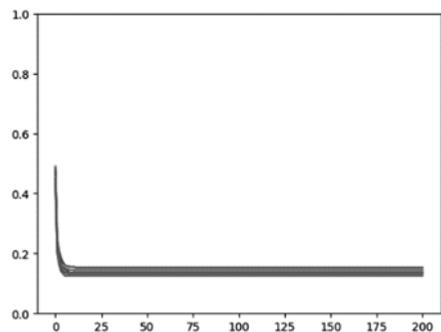


図4 5類感染症移行から数ヶ月後

またマスク着用率の数値の推移が、5類感染症移行前(図2)では25回のステップを終えたあたりから、5類感染症移行直後の3月(図3)では、50回のステップを終えたあたりから、5類感染症移行から数ヶ月後の6月(図4)では、ステップが25回を到達するまでに、一定の数値に落ち着いている。マスク着用率の低下スピードが、最も同調圧力を感じやすいエージェントの割合が多い5類感染症移行から数ヶ月後の6月(図4)→

最も感染リスクを気にするエージェントの割合が多い5類感染症移行前（図2）→5類感染症移行直後の3月（図3）という順番で速い。

同調圧力の影響の比較

本節では、同じ感染のリスクへの危機感を持っていたとしても、同調圧力の影響の感じやすさの違いで、マスク着用率にどのような変化が現れるのか検討する。そこで、結果3の5類感染症移行から数ヶ月後の6月を想定したプログラムと、同調圧力の影響を比較できるプログラムを作成する（以下、本プログラムを「6月の同調圧力比較版」とする）。

初期のマスク着用率は、5類感染症移行から数ヶ月後の6月を想定したプログラムと同様に設定する。次に、エージェントの割合は、感染のリスクを気にしない同調圧力を感じにくいエージェントが5割、感染のリスクを気にしない同調圧力を感じやすいエージェントが2割、感染のリスクを気にする同調圧力を感じにくいエージェントが2割、感染のリスクを気にする同調圧力を感じやすいエージェントが1割と、5類感染症移行から数ヶ月後の6月を想定したプログラムと同調圧力の影響を真逆に設定する。

5類感染症移行から数ヶ月後の6月を想定したプログラムと、本節で作成した6月の同調圧力比較版のプログラムの結果が以下の通りである。

5類感染症移行から数ヶ月後の6月を想定したプログラムの結果のグラフ（図4）と6月の同調圧力比較版の結果のグラフ（図5）を比較すると、同調圧力の影響が弱い図5の方がマスク着用率の低下スピードが緩やかになっている。両プログラムとも最終的には同じような数値に落ち着くものの、そこに至るまでの過程が同調圧力の影響が弱い方が時間がかかっていることが分かる。

また最終的に落ち着く数値が、同調圧力の強い5類感染症移行から数ヶ月後の6月（図4）よりも、同調圧力の弱い6月の同調圧力比較版（図5）の方が数値の幅が縦に広く、10回のシミュレーションごとのばらつきが大きい。

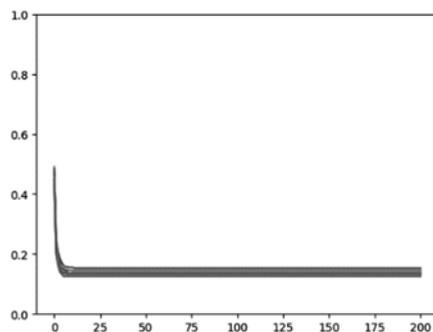


図4 5類感染症移行から数ヶ月後（再掲）

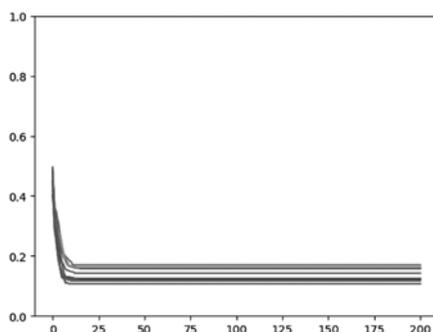


図5 6月の同調圧力の比較版

結論と考察

本論文では、マスクの着用理由を「感染のリスク」と「同調圧力」という2点に設定し、5類感染症移行前、5類感染症移行直後の3月、5類感染症移行から数ヶ月後の6月、という3つの時期に分けてマスク着用率の変化を検討した。各プログラムのシミュレーションの結果、感染のリスクを気にするエージェント、同調圧力を感じやすいエージェントが全体の何割を占めているかによって、マスク着用率の低下スピードに違いが出ている。また感染のリスクへの危機感と同じでも、同調圧力の影響があるかないかで、さらにマスク着用率の低下スピードに変化が現れた。人々がマスクの着用について、感染のリスクだけを考えていれば、5類感染症移行直後にもっと多くの人々がマスクの着用をやめたと考えられるが、マスク着用率が徐々に減っているのをみれば、5類感染症移行により、身の回りの人がどのくらいマスクの着

新型コロナウイルスにおけるマスクの着用について

用をやめるのか様子を見ていた＝新型コロナウイルス感染症において同調圧力の影響はあったと推察できる。

最後に、本研究の限界点について述べていく。

第1に、本研究ではプログラムにおいて、同調圧力を感じやすいエージェントは、その近傍の中で10%以上マスクを取っていたらマスクを外す。感じにくいエージェントは90%以上マスクを取っていたらマスクを外す。という初期設定を行っているが、同調圧力をどのくらいの程度から感じるかは個人差があるため、すべての人を本研究の同調圧力の設定に当てはめることはできない。

第2に、本研究ではマスクの着用理由を「感染のリスク」「同調圧力」という2点に限定しているが、この2点以外にも、“着用していると安心感がある”、“マスクの着用に慣れた”（株式会社ロイヤリティマーケティング、2023），“メイクをしていなくてもいい”、“髭を添ってなくてもいい”（日本インフォメーション株式会社、2022）など様々なマスクの着用理由があることを考えれば、別の比較をすることで、本論文とは違う結果が得られる可能性もあるだろう。

参考文献

- 法務省、2022、3 新型コロナウイルス感染症緊急事態宣言期間等の推移、令和4年版犯罪白書、第7編第2章3
- 株式会社学情、2023、人事の図書館 同調圧力とは？生まれる理由と企業にもたらすメリット・デメリットを解説
- 株式会社ロイヤリティマーケティング、2023、男女1,000人に聞いたマスク着用に関する調査（第5回）
- 厚生労働省、2020、新型コロナウイルス感染症の現在の状況と厚生労働省の対応について（令和2年6月30日版）
- 厚生労働省、2023、マスクの着用について
- NHK NEWS WEB、2023、新型コロナ 都医師会「第9波に入っている」感染対策呼びかけ
- 日本インフォメーション株式会社、2022、～マスク生活はどこまで続く？～マスク着用の意識・行動調査

日本インフォメーション株式会社、2023、～アフターコロナでどう変わる？～コロナに対する意識・行動調査

日本政府観光局（JNTO）、2023、訪日外客数（2023年10月推計値）

岡部 信彦、2020、これまでの出来事の総括（Chronology）、日本内科学会雑誌、109（11）：2264-2269

太田 昌克・飯田 敏幸・河岡 司、1995、人の同調圧力に基づく意思決定モデル

ourly Mag、2023、同調圧力とは？日本が強いと言われる理由と組織で活かす方法を解説

首相官邸ホームページ、2020、新型コロナウイルス感染症対策本部（第25回）

ビデオリサーチダイジェストプラス、2023、緊急事態宣言やまん防はいつからいつまで？時系列分析に役立つ過去発令期間やトピックスまとめ

読売新聞オンライン、2023、2023年1～2月 郵送全国世論調査「新型コロナ3年」質問と回答

読売新聞オンライン、2023、やっぱり同調圧力？消えないマスクの背景

付録

【5類感染症移行前を想定したプログラム (図2)】

```

%matplotlib inline

#%matplotlib nbagg

from random import uniform, seed
import random

from math import sqrt

import matplotlib.pyplot as plt

import numpy as np

from matplotlib.animation import FuncAnimation

from IPython.display import HTML

class Agent:

    def __init__(self, type1, type2): #type1 は感染リスク(risk)気にする 1
    or 気にしない 0、type2 は同調圧力 (pressure) 感じやすい 1 or 感じにくい 0 0 か 1 の
    離散変数

        #初期状態:感染リスクを気にする人は全員マスクをしている、感染リスクを気にしない人は
        全員マスクをしない

        #初期値:同調圧力を感じやすい人は自分の近傍の人の中で x 人以上マスクを取っていた
        らマスクを外す 感じにくい人は y 人以上マスクを取っていたらマスクを外す

        #y は x よりも大きい これらの値はシミュレーションの初期値として決める

        #感染リスクを気にする人は周りが何人マスクを取っていてもマスクは取らない

        #感染リスクを気にしない人は周りの人のうち閾値を超える人数がマスクを取っていたら自分
        もマスクを取る

        self.risk= type1

        self.pressure=type2

        #self.mask=1

        #self.mask=random.randint(0,1)

```

新型コロナウイルスにおけるマスクの着用について

```

    if random.random()<0.01: #マスクをしていない人の割合を入れる #5 類移
行前のため、ほとんどの人がマスクを着用している設定に
        self.mask=0
    else:
        self.mask=1
    self.draw_location()

def draw_location(self):
    self.location = uniform(0, 1), uniform(0, 1)

def get_distance(self, other):
    a = (self.location[0] - other.location[0])**2
    b = (self.location[1] - other.location[1])**2
    distance = sqrt(a + b)
    return distance

def wearmask(self):
    distances = []
    for agent in agents:
        if self!=agent:
            distance = self.get_distance(agent)
            distances.append((distance, agent))
    distances.sort()
    neighbors = [agent for d, agent in
distances[:num_neighbors]]
    num_mask=sum(agent.mask==1 for agent in neighbors)
    num_mask_rate=num_mask/len(neighbors)
    num_unmask=sum(agent.mask==0 for agent in neighbors)
    num_unmask_rate=num_unmask/len(neighbors) #マスクをしていない人の
割合の方が下の意思決定のコードが明確になる

```

```

    num_same_risk = sum(self.risk == agent.risk for agent in
neighbors) #同じ感染リスク気にする条件の人の数を数える

    num_same_pressure = sum(self.risk != agent.risk and
self.pressure == agent.pressure for agent in neighbors)

    #感染リスク気にする条件は違うが同じ同調圧力感じやすさ条件の人の数を数え
る

    num_same_type = num_same_risk + num_same_pressure

    if self.pressure==1: #同調圧力の感じやすさの違いによりマスクを取る閾値
が違ふ

        self.threshold=0.1

    else:

        self.threshold=0.9

    if (self.mask==1) and (self.risk==0): #マスクを取るか取らないかの
意思決定

        if num_unmask_rate > self.threshold:

            self.mask = 0

```

```

num_of_agents=400

num_of_type_00 = round(num_of_agents*0.1) #感染のリスクを気にしないエージェ
ントはそれぞれ 1 割ずつ

num_of_type_01 = round(num_of_agents*0.1)

num_of_type_10 = round(num_of_agents*0.6) #5 類移行前のため、感染のリスク
を気にする&同調圧力を感じない人を 6 割に設定

num_of_type_11 = round(num_of_agents*0.2) #5 類移行前は感染のリスクが最も
重要視されていると考えられるため、感染のリスクを気にする&同調圧力を気にする人は 2 割に

mask_rate=[] #マスクをしている人の率

num_neighbors = 4

```

新型コロナウイルスにおけるマスクの着用について

```

require_same_type = 6
Z=0.5 #エージェント i の半径 Z の近傍を示す

for iteration in range(10):
    mask_number=[]
    mask_rate=[]
    #ここでエージェントをインスタンス化する
    #ここでやることで、毎繰り返しごとに新しいエージェントができる
    agents = [Agent(0,0) for i in range(num_of_type_00)]
    agents.extend(Agent(0,1) for i in range(num_of_type_01))
    agents.extend(Agent(1,0) for i in range(num_of_type_10))
    agents.extend(Agent(1,1) for i in range(num_of_type_11))
    #マスクをしている人の人数を計算する
    count=0
    for i in agents:
        if i.mask==1:
            count+=1
    print(count)
    mask_number.append(count) #マスクをしている人数の初期値を入れる
    mask_rate.append(count/num_of_agents) #マスクをしている人の率の初期値を
    入れる

    for t in range(200):
        #順序効果を避けるために、毎回 agents をシャッフルして agents_shuffle を作る
        agents_shuffle=random.sample(agents, len(agents))
        for i in agents_shuffle:
            i.wearmask()
            i.location = uniform(0,1),uniform(0,1) #エージェントが移動できるように
            する
        count=0

```

```

for i in agents_shuffle:
    if i.mask==1:
        count+=1

    mask_number.append(count)

    mask_rate.append(count/num_of_agents)

print(mask_number)
print(mask_rate)
plt.plot(mask_rate)
plt.ylim(0,1)

```

【5 類感染症移行直後の 3 月を想定したプログラム (図 3)】

```

%matplotlib inline
#%matplotlib nbagg
from random import uniform, seed
import random
from math import sqrt
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation
from IPython.display import HTML

```

```

class Agent:
    def __init__(self, type1, type2): #type1 は感染リスク(risk) 気にする 1
    or 気にしない 0、type2 は同調圧力(pressure) 感じやすい 1 or 感じにくい 0 0 か 1 の
    離散変数
        #初期状態: 感染リスクを気にする人は全員マスクをしている、感染リスクを気にしない人は
    全員マスクをしない
        #初期値: 同調圧力を 感じやすい人は自分の近傍の人の中で x 人以上マスクを取っていた
    らマスクを外す 感じにくい人は y 人以上マスクを取っていたらマスクを外す
        #y は x よりも大きい これらの値はシミュレーションの初期値として決める

```

新型コロナウイルスにおけるマスクの着用について

```

#感染リスクを気にする人は周りが何人マスクを取っていてもマスクは取らない
#感染リスクを気にしない人は周りの人のうち閾値を超える人数がマスクを取っていたら自分
もマスクを取る

    self.risk= type1
    self.pressure=type2

    #self.mask=1
    #self.mask=random.randint(0,1)

    if random.random()<0.25: #マスクをしていない人の割合を入れる #株式会
社ロイヤリティマーケティングの3月の調査結果はこれまで通りマスクを着用している人が75%
        self.mask=0
    else:
        self.mask=1

    self.draw_location()

def draw_location(self):
    self.location = uniform(0, 1), uniform(0, 1)

def get_distance(self, other):
    a = (self.location[0] - other.location[0])**2
    b = (self.location[1] - other.location[1])**2
    distance = sqrt(a + b)

    return distance

def wearmask(self):
    distances = []
    for agent in agents:
        if self!=agent:
            distance = self.get_distance(agent)
            distances.append((distance, agent))
    distances.sort()

```

```

    neighbors = [agent for d, agent in
distances[:num_neighbors]]

    num_mask=sum(agent.mask==1 for agent in neighbors)

    num_mask_rate=num_mask/len(neighbors)

    num_unmask=sum(agent.mask==0 for agent in neighbors)

    num_unmask_rate=num_unmask/len(neighbors) #マスクをしていない人の
割合の方が下の意思決定のコードが明確になる

    num_same_risk = sum(self.risk == agent.risk for agent in
neighbors) #同じ感染リスク気にする条件の人の数を数える

    num_same_pressure = sum(self.risk != agent.risk and
self.pressure == agent.pressure for agent in neighbors)

    #感染リスク気にする条件は違うが同じ同調圧力感じやすさ条件の人の数を数え
る

    num_same_type =num_same_risk + num_same_pressure

    if self.pressure==1: #同調圧力の感じやすさの違いによりマスクを取る閾値
が違ふ

        self.threshold=0.1
    else:

        self.threshold=0.9

    if (self.mask==1) and (self.risk==0): #マスクを取るか取らないかの
意思決定

        if num_unmask_rate > self.threshold:

            self.mask = 0

```

```

num_of_agents=400

num_of_type_00 = round(num_of_agents*0.2) #5 類移行によって感染のリスクに
対する危機感が減ると考えられるため、感染のリスクを気にする人と気にしない人の割合を半数
ずつに

```

新型コロナウイルスにおけるマスクの着用について

```

num_of_type_01 = round(num_of_agents*0.3) #5 類移行前より同調圧力を気にす
る人の割合を少し増やす
num_of_type_10 = round(num_of_agents*0.2)
num_of_type_11 = round(num_of_agents*0.3)

mask_rate=[] #マスクをしている人の率

num_neighbors = 4
require_same_type = 6
z=0.5 #エージェント i の半径 z の近傍を示す

for iteration in range(10):
    mask_number=[]
    mask_rate=[]
    #ここでエージェントをインスタンス化する
    #ここでやることで、毎繰り返しごとに新しいエージェントができる
    agents = [Agent(0,0) for i in range(num_of_type_00)]
    agents.extend(Agent(0,1) for i in range(num_of_type_01))
    agents.extend(Agent(1,0) for i in range(num_of_type_10))
    agents.extend(Agent(1,1) for i in range(num_of_type_11))
    #マスクをしている人の人数を計算する
    count=0
    for i in agents:
        if i.mask==1:
            count+=1
    print(count)
    mask_number.append(count) #マスクをしている人数の初期値を入れる
    mask_rate.append(count/num_of_agents) #マスクをしている人の率の初期値を
    入れる

```

```

for t in range(200):
    #順序効果を避けるために、毎回 agents をシャッフルして agents_shuffle を作る
    agents_shuffle=random.sample(agents, len(agents))

    for i in agents_shuffle:
        i.wearmask()

        i.location = uniform(0,1),uniform(0,1) #エージェントが移動できるように
する

    count=0
    for i in agents_shuffle:
        if i.mask==1:
            count+=1

    mask_number.append(count)
    mask_rate.append(count/num_of_agents)

print(mask_number)
print(mask_rate)
plt.plot(mask_rate)
plt.ylim(0,1)

```

【5 類感染症移行から数ヶ月後の 6 月を想定したプログラム (図 4)】

```

%matplotlib inline
#%matplotlib nbagg
from random import uniform, seed
import random
from math import sqrt
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation
from IPython.display import HTML

```

```

class Agent:

```

新型コロナウイルスにおけるマスクの着用について

```

def __init__(self, type1, type2): #type1 は感染リスク (risk) 気にする 1
or 気にしない 0、type2 は同調圧力 (pressure) 感じやすい 1 or 感じにくい 0 0 か 1 の
離散変数

    #初期状態:感染リスクを気にする人は全員マスクをしている、感染リスクを気にしない人は
全員マスクをしない

    #初期値:同調圧力を感じやすい人は自分の近傍の人の中で x 人以上マスクを取っていたら
マスクを外す 感じにくい人は y 人以上マスクを取っていたらマスクを外す

    #y は x よりも大きい これらの値はシミュレーションの初期値として決める

    #感染リスクを気にする人は周りが何人マスクを取っていてもマスクは取らない

    #感染リスクを気にしない人は周りの人のうち閾値を超える人数がマスクを取っていたら自分
もマスクを取る

    self.risk= type1

    self.pressure=type2

    #self.mask=1

    #self.mask=random.randint(0,1)

    if random.random()<0.54: #マスクをしていない人の割合を入れる #株式会
社ロイヤリティマーケティングの 6 月の調査結果はこれまで通りマスクを着用している人が 46%

        self.mask=0

    else:

        self.mask=1

    self.draw_location()

def draw_location(self):

    self.location = uniform(0, 1), uniform(0, 1)

def get_distance(self, other):

    a = (self.location[0] - other.location[0])**2

    b = (self.location[1] - other.location[1])**2

    distance = sqrt(a + b)

    return distance

```

```

def wearmask(self):
    distances = []
    for agent in agents:
        if self!=agent:
            distance = self.get_distance(agent)
            distances.append((distance, agent))
    distances.sort()
    neighbors = [agent for d, agent in
distances[:num_neighbors]]
    num_mask=sum(agent.mask==1 for agent in neighbors)
    num_mask_rate=num_mask/len(neighbors)
    num_unmask=sum(agent.mask==0 for agent in neighbors)
    num_unmask_rate=num_unmask/len(neighbors) #マスクをしていない人の
割合の方が下の意思決定のコードが明確になる
    num_same_risk = sum(self.risk == agent.risk for agent in
neighbors) #同じ感染リスク気にする条件の人の数を数える
    num_same_pressure = sum(self.risk != agent.risk and
self.pressure == agent.pressure for agent in neighbors)
    #感染リスク気にする条件は違うが同じ同調圧力感じやすさ条件の人の数を数え
る
    num_same_type =num_same_risk + num_same_pressure
    if self.pressure==1: #同調圧力の感じやすさの違いによりマスクを取る閾値
が違ふ
        self.threshold=0.1
    else:
        self.threshold=0.9

    if (self.mask==1) and (self.risk==0): #マスクを取るか取らないかの
意思決定

```

新型コロナウイルスにおけるマスクの着用について

```

    if num_unmask_rate > self.threshold:
        self.mask = 0

```

```

num_of_agents=400
num_of_type_00 = round(num_of_agents*0.2) #5 類感染症移行直後よりも感染の
リスクへの危機感はさらに減ると考えられるため、感染のリスクを気にしない人の割合を増やす
num_of_type_01 = round(num_of_agents*0.5) #同調圧力を気にする人の割合もさ
らに増やす
num_of_type_10 = round(num_of_agents*0.1)
num_of_type_11 = round(num_of_agents*0.2)

mask_rate=[] #マスクをしている人の率

num_neighbors = 4
require_same_type = 6
z=0.5 #エージェント i の半径 z の近傍を示す

```

```

for iteration in range(10):
    mask_number=[]
    mask_rate=[]
    #ここでエージェントをインスタンス化する
    #ここでやることで、毎繰り返しごとに新しいエージェントができる
    agents = [Agent(0,0) for i in range(num_of_type_00)]
    agents.extend(Agent(0,1) for i in range(num_of_type_01))
    agents.extend(Agent(1,0) for i in range(num_of_type_10))
    agents.extend(Agent(1,1) for i in range(num_of_type_11))
    #マスクをしている人の人数を計算する
    count=0
    for i in agents:
        if i.mask==1:

```

```

    count+=1
print(count)
mask_number.append(count) #マスクをしている人数の初期値を入れる
mask_rate.append(count/num_of_agents) #マスクをしている人の率の初期値を
入れる

for t in range(200):
    #順序効果避けるために、毎回 agents をシャッフルして agents_shuffle を作る
    agents_shuffle=random.sample(agents, len(agents))
    for i in agents_shuffle:
        i.wearmask()
        i.location = uniform(0,1),uniform(0,1) #エージェントが移動できるように
する
    count=0
    for i in agents_shuffle:
        if i.mask==1:
            count+=1
    mask_number.append(count)
    mask_rate.append(count/num_of_agents)
print(mask_number)
print(mask_rate)
plt.plot(mask_rate)
plt.ylim(0,1)

```

【6月の同調圧力比較版のプログラム（図5）】

```

%matplotlib inline
#%matplotlib nbagg
from random import uniform, seed
import random
from math import sqrt

```

新型コロナウイルスにおけるマスクの着用について

```

import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation
from IPython.display import HTML

class Agent:
    def __init__(self, type1, type2): #type1 は感染リスク(risk)気にする 1
    or 気にしない 0、type2 は同調圧力(pressure)感じやすい 1 or 感じにくい 0 0 か 1 の
    離散変数
        #初期状態:感染リスクを気にする人は全員マスクをしている、感染リスクを気にしない人は
        全員マスクをしない
        #初期値:同調圧力を感じやすい人は自分の近傍の人の中で x 人以上マスクを取っていた
        らマスクを外す 感じにくい人は y 人以上マスクを取っていたらマスクを外す
        #y は x よりも大きい これらの値はシミュレーションの初期値として決める
        #感染リスクを気にする人は周り何人マスクを取っていてもマスクは取らない
        #感染リスクを気にしない人は周りの人のうち閾値を超える人数がマスクを取っていたら自分
        もマスクを取る

        self.risk= type1

        self.pressure=type2

        #self.mask=1

        #self.mask=random.randint(0,1)

        if random.random()<0.54: #マスクをしていない人の割合を入れる #株式会
        社ロイヤリティマーケティングの 6 月の調査結果はこれまで通りマスクを着用している人が 46%
            self.mask=0

        else:
            self.mask=1

        self.draw_location()

    def draw_location(self):
        self.location = uniform(0, 1), uniform(0, 1)

```

```

def get_distance(self, other):
    a = (self.location[0] - other.location[0])**2
    b = (self.location[1] - other.location[1])**2
    distance = sqrt(a + b)
    return distance

def wearmask(self):
    distances = []
    for agent in agents:
        if self!=agent:
            distance = self.get_distance(agent)
            distances.append((distance, agent))
    distances.sort()
    neighbors = [agent for d, agent in
distances[:num_neighbors]]
    num_mask=sum(agent.mask==1 for agent in neighbors)
    num_mask_rate=num_mask/len(neighbors)
    num_unmask=sum(agent.mask==0 for agent in neighbors)
    num_unmask_rate=num_unmask/len(neighbors) #マスクをしていない人の
割合の方が下の意思決定のコードが明確になる
    num_same_risk = sum(self.risk == agent.risk for agent in
neighbors) #同じ感染リスク気にする条件の人の数を数える
    num_same_pressure = sum(self.risk != agent.risk and
self.pressure == agent.pressure for agent in neighbors)
    #感染リスク気にする条件は違うが同じ同調圧力感じやすさ条件の人の数を数え
る
    num_same_type =num_same_risk + num_same_pressure
    if self.pressure==1: #同調圧力の感じやすさの違いによりマスクを取る閾値
が違ふ

```

新型コロナウイルスにおけるマスクの着用について

```

        self.threshold=0.1
    else:
        self.threshold=0.9

    if (self.mask==1) and (self.risk==0): #マスクを取るか取らないかの
意思決定
        if num_unmask_rate > self.threshold:
            self.mask = 0

```

```

num_of_agents=400
num_of_type_00 = round(num_of_agents*0.5) #結果 3 のプログラムと同調圧力の
影響を反対に
num_of_type_01 = round(num_of_agents*0.2)
num_of_type_10 = round(num_of_agents*0.2)
num_of_type_11 = round(num_of_agents*0.1)

mask_rate=[] #マスクをしている人の率

num_neighbors = 4
require_same_type = 6
Z=0.5 #エージェント i の半径 Z の近傍を示す

```

```

for iteration in range(10):
    mask_number=[]
    mask_rate=[]
    #ここでエージェントをインスタンス化する
    #ここでやることで、毎繰り返しごとに新しいエージェントができる
    agents = [Agent(0,0) for i in range(num_of_type_00)]
    agents.extend(Agent(0,1) for i in range(num_of_type_01))
    agents.extend(Agent(1,0) for i in range(num_of_type_10))

```

```

agents.extend(Agent(1,1) for i in range(num_of_type_11))
#マスクをしている人の人数を計算する
count=0
for i in agents:
    if i.mask==1:
        count+=1
print(count)
mask_number.append(count) #マスクをしている人数の初期値を入れる
mask_rate.append(count/num_of_agents) #マスクをしている人の率の初期値を
入れる

for t in range(200):
    #順序効果を避けるために、毎回 agents をシャッフルして agents_shuffle を作る
    agents_shuffle=random.sample(agents, len(agents))
    for i in agents_shuffle:
        i.wearmask()
        i.location = uniform(0,1),uniform(0,1) #エージェントが移動できるように
する
    count=0
    for i in agents_shuffle:
        if i.mask==1:
            count+=1
    mask_number.append(count)
    mask_rate.append(count/num_of_agents)
print(mask_number)
print(mask_rate)
plt.plot(mask_rate)
plt.ylim(0,1)

```