SNS においてフェイクニュースを拡散するユーザの特徴抽出

プロジェクトマネジメントコース
ソフトウェア開発管理グループ
矢吹研究室
1442014
岩橋瑠伊

目次

第 1 章

序論

Twitter には日常的なことからニュースまで，様々な情報が流れている．本研究では，その中でもデマツイートという嘘の情報に絞って分析を行う．

第 2 章

背景

2.1　デマツイートの影響

SNS 上でフェイクニュースを拡散するユーザの特徴について調査する．SNS などのウェブ上のメディアで，フェイクニュースが問題視されている [1].

例えば，2011 年 3 月 11 日に発生した東日本大震災時に，携帯電話がつながらない状況下での有用な連絡手段として活躍した．しかし，その有用性はデマや誤情報も大量に拡散させる手助けとなりえる．実際に東日本大震災時に，数十種類のデマや誤情報が情報として拡散されてしまい，日本中を混乱させた．震災時のように連絡手段が限られた状況はこれからも発生する可能性は十分にあり，対策が必要である [2].

2.2　デマを拡散するユーザー

本研究では，デマが拡散されることを防ぐためにデマツイートをリツイートしているユーザの特徴抽出を行う．デマツイートがリツイートされる原因として，デマをデマと見抜けないユーザ，面白半分でリツイートしているユーザの 2 種類がいると考えた．この 2 種類のユーザと，それ以外のユーザには Twitter の使い方に違いがあるのではないかと考えた．

第 3 章

目的

デマツイートをリツイートするユーザーと，それ以外のユーザーの Twitter の使い方に違いを見つける．違いからデマツイートをリツイートするユーザーなのかを判別できるようにする．

第 4 章

Twitter について

4.1　本章の構成

本章では本研究で使用する Twitter について説明する．

4.2　Twitter とは

Twitter は，「ツイート」と称される 140 文字以内の短文の投稿を共有するウェブ上の情報サービスである．

図 4.1　Twitter ホームページ

4.3　用語

本節では，Twitter の用語について説明する．

4.3.1　ツイート

140 文字以下の文字列のこと．つぶやきとも呼ばれる．

図 4.2　ツイートの例

4.3.2　ユーザー

Twitter を利用している者を指す．

4.3.3　ユーザー名

半角英数字，アンダーバーから計 15 文字以内で作るユーザーの名前である．

4.3.4　タイムライン

ユーザー自身のホーム画面のことである．フォローしているユーザーのつぶやきや，自身のつぶやき，リツイートされたツイートなどの情報が羅列されていく．

図 4.3　タイムラインの例

4.3.5　フォロー

他人のツイートを購読する機能のこと．他人を「フォロー」することで，その人のツイートが TL に表示されるようになる．フォローは基本的に一方的に行われ，許可の必要はなくフォローし返さなければならないということもない．

4.3.6　フォロワー

自分をフォローしているユーザーのこと．フォロワーのタイムラインに自分（フォローされている人）のつぶやきが表示される．

4.3.7　リムーブ

フォローを解除すること．

4.3.8　ブロック

迷惑なユーザーや嫌いなユーザーなどを遮断する機能のこと．他人をブロックした場合，そのユーザーからのフォローを遮断しそのユーザーのツイートが RT されても自分の TL には表示されなくなる．

4.3.9　リツイート

他の人のツイートを再びツイートすること．自分のタイムラインに流れてきたツイートをリツイートすると，自分のフォロワーのタイムラインにそのツイートが流れる．逆に自分がフォローしているユーザーがリツイートすると，自分のタイムラインにそのツイートが流れる．

4.3.10　リプライ

特定のユーザー名 (@...) から始まるツイートをリプライという．そのユーザー宛のツイートということになる．リプライを送った側と，送られた側の両方をフォローしているユーザーのタイムラインには表示されるが，片方のみをフォローしている第三者のタイムラインには表示されない．

4.3.11　いいね

あとで読み返したいと思ったツイートや気に入ったツイートを，♡マークを付けて自分のいいねリストへ登録すること．以前はお気に入り (Favorites) だったが，2015 年 11 月 4 日，「いいね」に変更され，これまでの「☆」から「♡」に変更された．

4.3.12　メンション

「@ユーザー名」を含むツイートの一覧.

第 5 章

TwitterAPI について

5.1　本章の構成

本章では本研究で使用する TwitterAPI について説明する．

5.2　TwitterAPI とは

Twitter 社が提供しているサービスで，Web サイトやアプリなどから Twitter の機能を呼び出すことができる．この API を利用することでツイートの参照や検索などを行なえるアプリケーション開発ができるようになる．

5.2.1　API

アプリケーションプログラムインターフェイス（Application Program Interface）の略で，プログラミングの際に使用できる命令や規約，関数等の集合のことを示す．ソフトウェア開発の際，いちから全てを作るより，API を利用すればもともとあるプログラムを呼び出して，その機能を組み込んだソフトウェアを開発することができる．

5.2.2　Twitter API

Twitter 社が提供するサービス．Web サイトやアプリケーションなどから Twitter の機能を呼び出すことができ，ツイートの参照や検索等を行えるアプリケーション開発を行えるようになる．

5.2.3　REST API

Twitter API のパラメータ（リソース）を指定し，特定の URL に HTTP でアクセスすると，JSON 形式で記述されたメッセージがレスポンスされるシステム．これはツイートの更新や参照を行う際に使用する基本的な API となる．ただし利用制限があり，15 分以内に同じ機能を特定の回数利用すると，はじめにその機能を利用した時間から 15 分経過するまではその機能が利用できなくなる．

5.2.4 Streaming API

タイムラインの変更をリアルタイムで自動に受け取ることができる．

5.2.5 Access Token

Web サービスを利用するために，認証局がユーザーを認証するために払い出した認証情報のことを指す．ここで言う認証局とはユーザーを認証する情報を保持しており，ユーザーを認証することができる Web サーバーで，Web サービスを提供している Web サーバー，または，ユーザーを認証することができる第三者の Web サーバーなどを示す．

第 6 章

開発環境構築ツールの解説

6.1　本章の構成

本章では本研究で使用する開発環境構築ツールについて説明する．

6.2　Chocolatey とは

「Chocolatey」は，コマンドラインによるアプリケーションの導入や削除を実現するパッケージ管理システム．パッケージ管理システムとは，アプリケーションやコンポーネント，ライブラリなどの管理を円滑に行うための仕組み．アプリケーションやライブラリを構成するファイル群を 1 つの”パッケージ”ファイルにまとめ，それを”リポジトリ”へ保管し，コマンドで取得・セットアップを行う．動作に必要な外部コンポーネントがあれば，その導入も自動で行われるのが一般的で，アプリケーションのインストールやアンインストールの手間を省くことができる．Linux ディストリビューションの多くは”apt-get”や”yum”といったパッケージ管理のためのコマンドを備えており，コマンドを入力するだけで手軽にソフトをダウンロード・インストール・アンインストールできて非常に便利．「Chocolatey」はこれを Windows 環境で実現しようというものである．

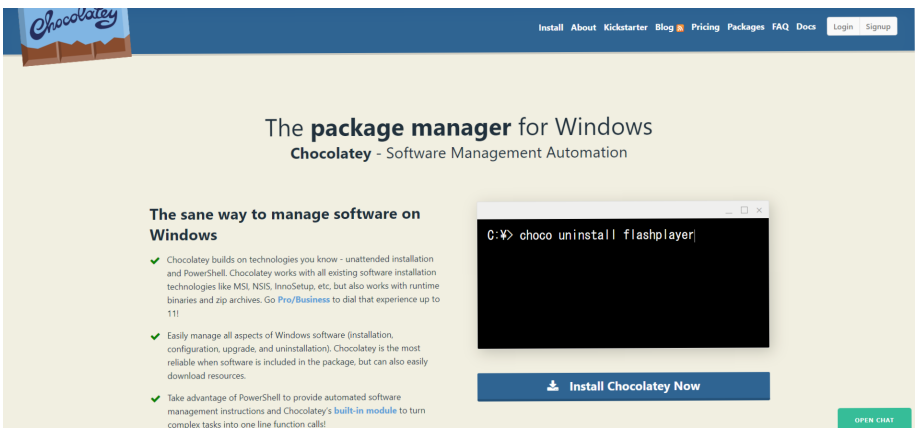


図 6.1　chocolatey のホームページ

6.2.1　パッケージとは

ソフトウェアにかかわるファイル一式がまとまったものをパッケージという．「ソフトウェアにかかわるファイル」には，設定ファイルやドキュメント，プログラム本体，プログラムが動くために必要なライブラリなどが含まれます．

6.2.2　choco list [packageName]

パッケージ検索．引数がなければすべてのパッケージを表示．

6.2.3　choco list -lo [packageName]

インストール済みのパッケージ検索．引数がなければすべてのインストール済みパッケージを表示．

6.2.4　cinst [packageName]

指定パッケージのインストール．

6.2.5　cuninst [packageName]

指定パッケージのアンインストール．

6.2.6　cup

Chocolatey 本体のアップデート．

6.2.7　cup [packageName]

指定パッケージのアップデート．

6.2.8　cup all

インストール済みのパッケージを全てアップデート．

6.3　Oracle VM VirtualBox とは

Oracle VM VirtualBox（以下，VirtualBox と表記する）は，使用している PC マシン上に仮想的なマシンを作成し，別の OS をインストール・実行することができるオープンソースソフトウェアである．Windows や Mac OS X，Linux 等，様々な OS で利用することができる．VirtualBox をインストールした PC のこと．

図 6.2　VirtualBox のイメージ

6.3.1　ホスト OS

VirtualBox をインストールした PC の OS のこと．本研究で使用するホスト PC の OS は Windows 7/8.1 である．

6.3.2　ゲスト PC（仮想マシン）

VirtualBox で作成した仮想 PC

6.3.3　ゲスト OS

VirtualBox で作成した仮想 PC にインストールした OS のこと．

6.4　Vagrant とは

Vagrant（ベイグラント）は，FLOSS の仮想開発環境構築ソフトウェアである．VirtualBox をはじめとする仮想化ソフトウェアや Chef（英語版）や Salt（英語版），Puppet といった構成管理ソフトウェアのラッパーとみなすこともできる．Vagrant を用いると構成情報を記述した設定ファイルを元に，仮想環境の構築から設定までを自動的に行うことができる．

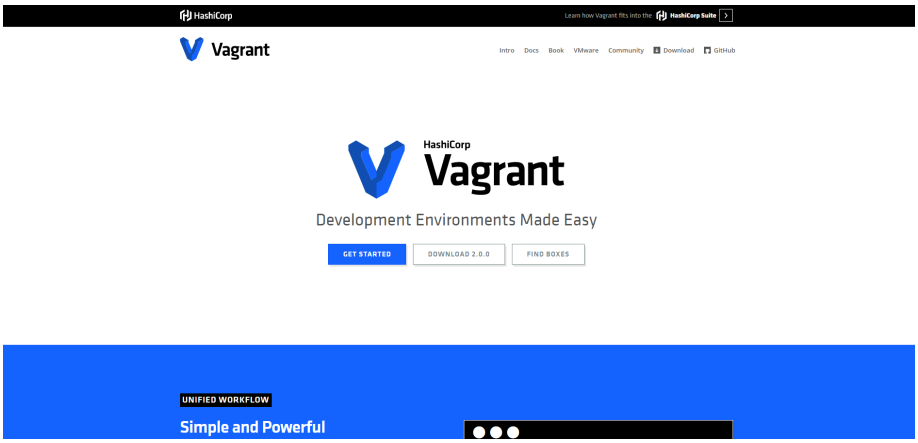


図 6.3　Vagrant のホームページ

第 7 章

T 検定について

7.1　本章の構成

本章では本研究で使用する T 検定について説明する．

7.1.1　T 検定とは

帰無仮説が正しいと仮定した場合に，統計量が t 分布に従うことを利用する統計学的検定法の総称である．母集団が正規分布に従うと仮定するパラメトリック検定法であり，t 分布が直接もとの平均や標準偏差にはよらない（ただし自由度による）ことを利用している．2 組の標本について平均に有意差があるかどうかの検定などに用いられる統計的仮説検定の一つ。日本工業規格では「検定統計量が，帰無仮説の下で t 分布に従うことを仮定して行う統計的検定.」と定義している．スチューデントの t 検定（Student's t-test）とも呼ばれるが，これは統計学者のウィリアム・ゴセットが雇用者であるギネスビール社に本名使用を許されず Student というペンネームで最初の論文を発表した（1908 年）ためである．

7.1.2　T 検定の種類

T 検定は大きく次のように分けられる．

1. 2 つの母集団がいずれも正規分布に従うと仮定したうえでの，平均が等しいかどうかの検定.
2. 標本が対になっている，つまり 1 組の標本のメンバー各々ともう 1 組の特定のメンバーとの間に特別な関係がある場合（例えば，同じ人に前後 2 回調査する場合夫と妻とで比較する場合など）
3. 標本が独立で，比較する 2 つの群の分散が等しいと仮定できる場合（等分散性の仮定）
4. 標本が独立で，等分散性が仮定できない（異分散）場合．これは正確にはウェルチの t 検定と呼ばれる．
5. 正規分布に従う母集団の平均が特定の値に等しいかどうかの検定.
6. 回帰直線の勾配が 0 と有意に異なるかどうかの検定.

第 8 章

手法

8.1　全体の流れ

本章では研究の流れについて説明する．

1. 開発環境の導入
2. 調査するデマツイートの決定．
3. TwitterAPI を利用できるように OAuth 認証を行う．
4. TwitterAPI を用いて日本人ユーザー 50 人をランダムサンプリングする．
5. TwitterAPI を用いてデマツイートをリツイートしたユーザー 50 人を取得する．
6. TwitterAPI を用いて集めた各ユーザーの最新 100 ツイートに含まれるリツイートの数を調べて，平均を計算する．
7. 日本人ユーザー 50 人とデマツイートをリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数の平均の差が，偶然的な誤差の範囲にあるものかどうかを判断する為に 2 標本 T 検定を行う．

8.2　開発環境の導入

本研究では様々なソフトウェアを使用する．そのため，ソフトウェアの導入方法を解説する．

8.2.1　Chocolatey の導入

パッケージ管理システム「Chocolatey」をインストールするためには、管理者権限でコマンドプロンプトを起動する必要がある．Windows 10 の場合，「スタート」ボタンを右クリックして，「コマンドプロンプト（管理者)」をクリックする．Chocolatey」をインストールするために，管理者権限で起動したコマンドプロンプトで以下のコマンドを実行する．

```
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile
-InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.
Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "
```

```
PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
```

インストール完了後コマンドプロンプトを再起動して，`clist` と入力し，ソフトウェア一覧が表示されればインストール成功である．（時間がかかるため，`Ctrl+c` でコマンドを停止する．）インストール方法は，今後変更される可能性がある．そのため，以下のコマンドが正常に動作しない場合は，Chocolatey Gallery から最新の情報を確認してください．

8.2.2　パッケージのインストール

「VirtualBox」と「vagrant」を Chocolatey を使って，コマンドプロンプトからインストールする．「VirtualBox」と「vagrant」をインストールするために，コマンドプロンプトで，以下のコマンドを実行する．

```
$ cinst -y vagrant virtualbox
```

8.2.3 Vagrant の導入

Vagrant を使うには，Box と呼ばれるファイルが必要になる．今回は矢吹研究室の公式マシンを使用する．

矢吹研究室公式マシンは以下の URL の GitHub リポジトリで公開されており，このリポジトリを clone して使用する．なお，矢吹研究室公式マシンは Linux ディストリビュージョンの一つである Ubuntu を使用している．矢吹研究室公式マシンは初回起動時に様々なアプリケーションを自動でインストールする．そのため，本論文に書かれているコードを公式マシン以外で実行してもうまく動作しない可能性がある．その際は必要なアプリケーションを `apt-get install [パッケージ名]`（ubuntu の場合）でインストールすることができる．

https://github.com/yabukilab/machine

本研究では `c:/vagrant` というディレクトリに clone する．

そうすると，`c:/vagrant/machine` 下に `Vagrantfile` というファイルが存在するはずである．存在を確認したらコマンドプロンプトで以下のコマンドを入力する．

```
cd c:\vagrant\machine

vagrant up

vagrant ssh
```

これで，コマンドプロンプト上で仮想マシンに ssh 接続できれば成功である．

8.2.4　TwitterAPI の導入

TwitterAPI を利用するためには Twitter のアカウントが必要である．その為，Twitter アカウントの作成方法を説明する．

1. Twitter の公式ホームページを google で検索し，Twitter のホームページに入る．
2. 図 4.1 の画面右上にある「アカウント作成」をクリックする．
3. Twitter を使う際に必要な「ニックネーム」，「電話番号もしくはメールアドレス」，「パスワード」を記載する．その後「アカウント作成」ボタンをクリックする．

図 8.1　アカウント作成画面

4. ここでは電話番号の入力を要求されるが，しなくてもアカウントの登録は可能である．電話番号を入力するのであれば，電話番号の記載を．電話番号を入力しないのであれば「次へ」ボタンの左下にある「スキップ」をクリックする．

図 8.2　電話番号入力画面

5. ユーザー名を記入する．もしくは利用可能なアカウント名をクリックして選ぶ．ユーザー名は登録後に変更が可能なため，その時の好きなユーザー名を記入することが出来る．また記入を行った際に，他のユーザーに使われていると記入ボックスの右隣りに「このユーザー名は既に使用されています。」と表示される．ユーザー名を決めることができたら，アカウント作成は終了である．

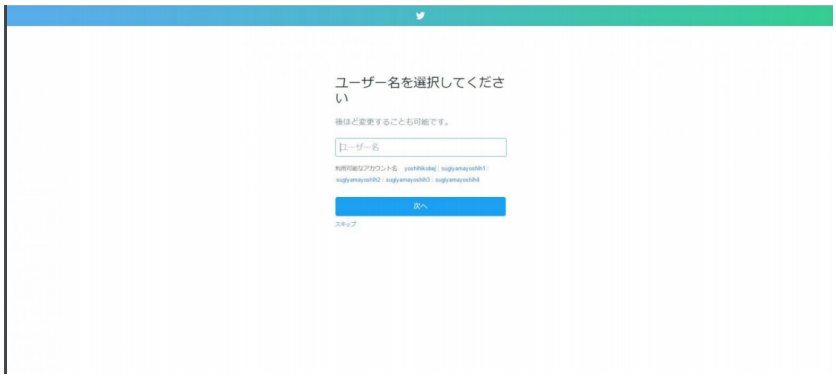


図 8.3　ユーザー名入力画面

Twitter API Key の取得方法

アプリケーションを作る際に必要になるのが，Consumer Key，Consumer Secret，Access Token，Access Token Secret の 4 つです．この 4 つを取得する方法について説明する．

1. https://apps.twitter.com/ にアクセスする
2. なにもアプリを作っていないと下記のような画面が出るので，Create New App ボタンをしてアプリケーションを作ります．

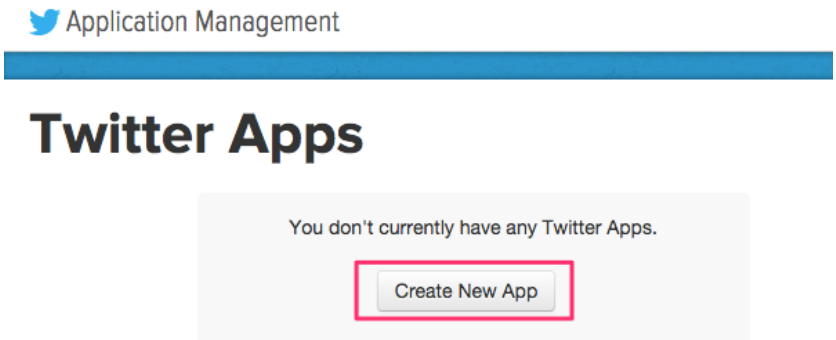


図 8.4　アプリ作成画面 1

3. Create New App ボタンを押すと，下記のような画面が出てきます．
4. Name にはアプリケーションの名前を入力します．文字数は 32 文字以内である．
5. Description にはアプリケーションの説明を入力します．認証系のアプリで使う場合は認証画面にこの文言が表示されます．文字数は 10 文字以上 200 文字以下である．

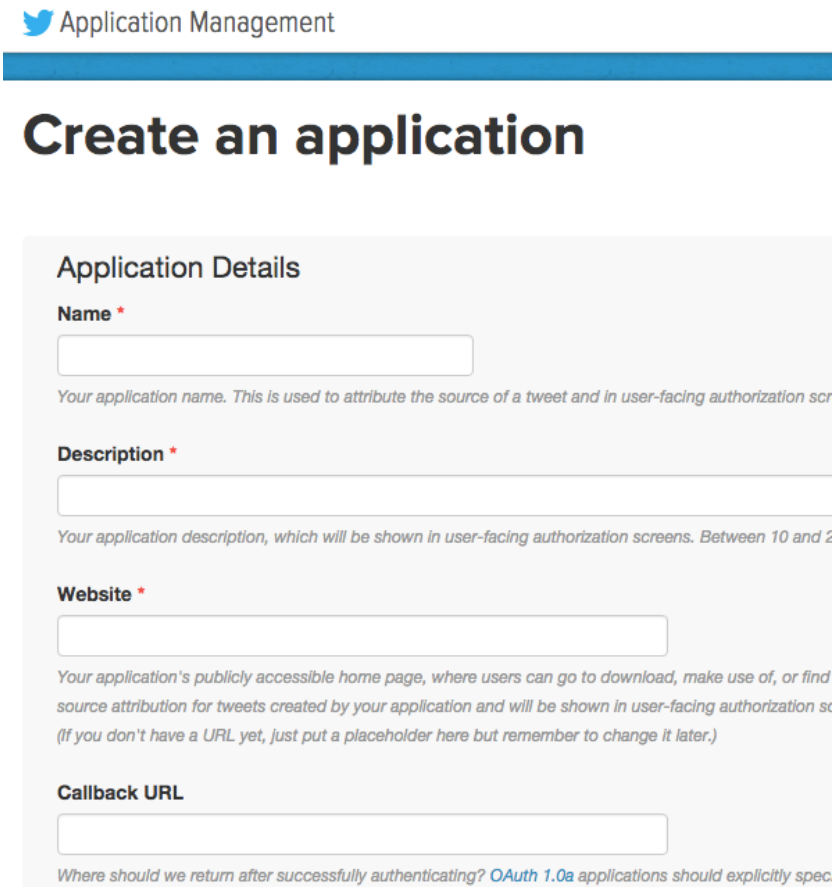


図 8.5　アプリ作成画面 2

6. Website にはアプリケーションの URL を入力します．まだ無い場合は仮の URL でも問題ありません．
7. CallbackURL は任意の入力項目です．コールバック URL 認証後に戻ってくる URL です．
8. アプリケーション作成後，アプリケーション管理画面の Keys and Access Tokens タブをクリックします．下記のような画面が表示されます．ここで Consumer Key と Consumer Secret を取得できます．

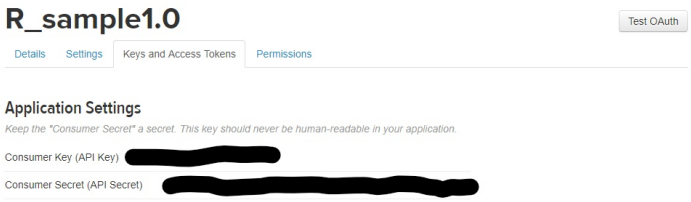


図 8.6　Key and Access Tokens

9. Keys and Access Tokens 画面の下部にある Create my access token ボタンをクリックすると下記のような画面が表示されます．ここで， Access Token と Access Token Secret を取得できます．

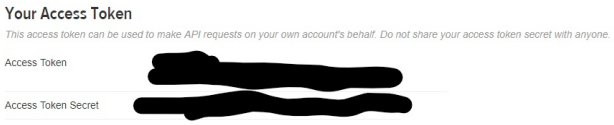


図 8.7　Key and Access Tokens2

8.3　調査するデマツイートの決定

デマと断定できるツイートを調べる．今回の研究では以下の 4 つのデマツイートを対象に調査を行う．

図 8.8　デマツイート 1

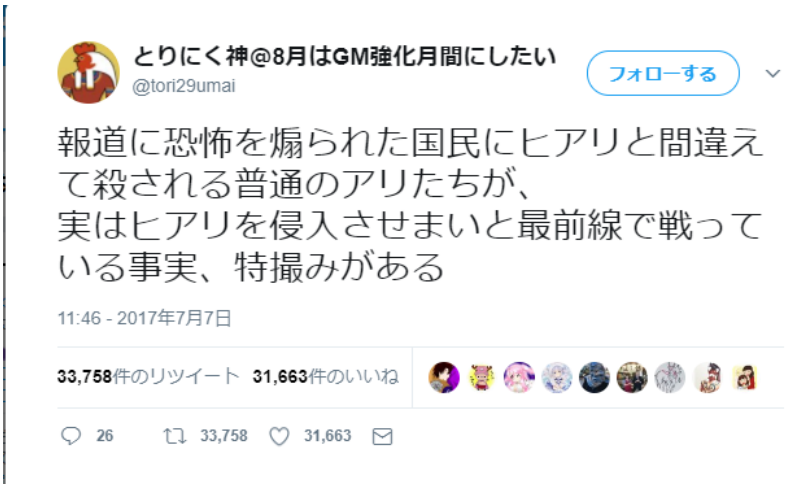

とりにく神@8月はGM強化月間にしたい
@tori29umai
フォローする
報道に恐怖を煽られた国民にヒアリと間違えて殺される普通のアリたちが、
実はヒアリを侵入させまいと最前線で戦っている事実、特撮みがある
11:46 - 2017年7月7日
33,758件のリツイート 31,663件のいいね
26
33,758
31,663

図 8.9　デマツイート 2

図 8.10　デマツイート 3

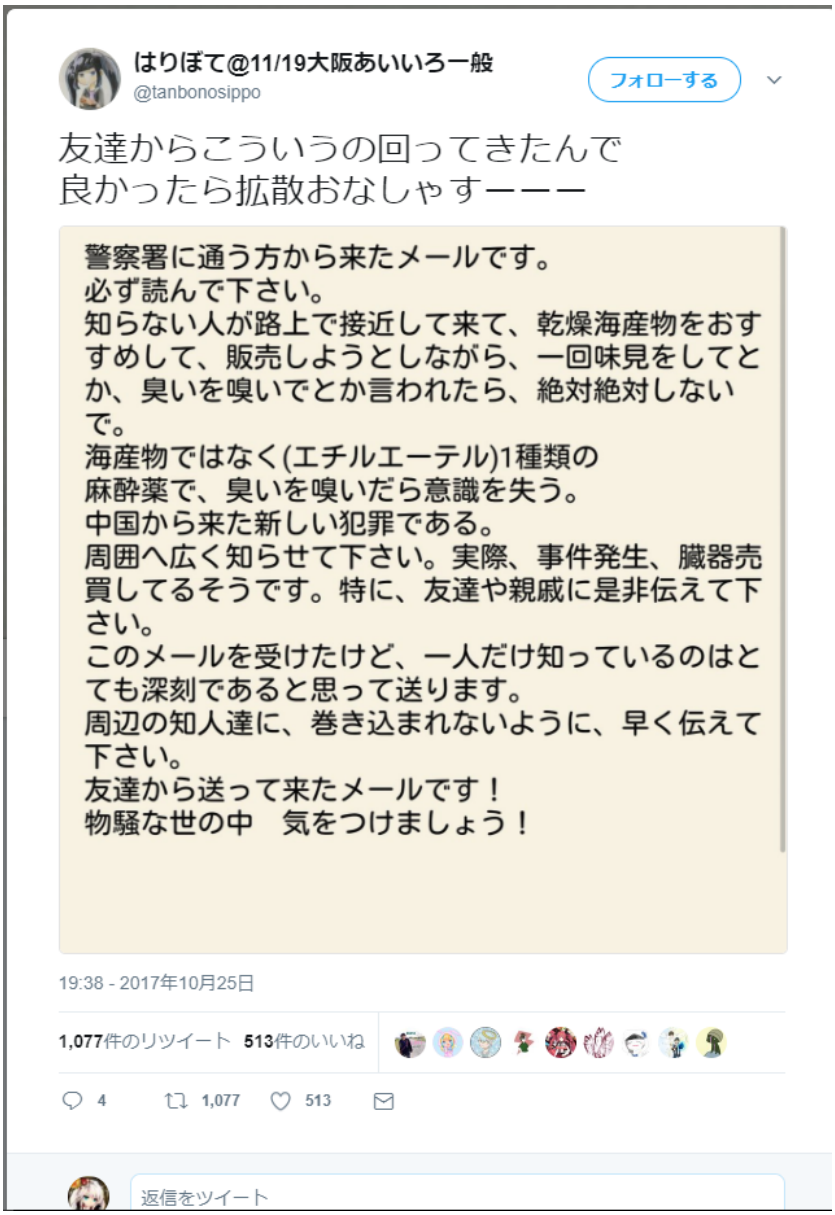


図 8.11　デマツイート 4

以上の 4 つのデマツイートの詳細を以下に示す．

- デマツイート 1 として，ロンドンのテロ対策で有志のスパルタ兵が巡回しているというデマツイートがある．実際は映画の宣伝である．（TweetID:872255950131822596）
- デマツイート 2 として，国内のアリがヒアリに対抗出来るというデマツイートがある．実際は誤情報である．（TweetID:883170290242527232）
- デマツイート 3 として，PS4 に PS2 互換機能対応するというデマツイートがある．実際は誤情報である．（TweetID:882139486968205312）
- デマツイート 4 として，路上で乾燥海産物を売る人がいるが，それは麻酔薬で匂いを嗅いだら意識を失うというデマツイートがある．実際は昔韓国で流行したデマの内容と同じものである．（TweetID:923151745923948545）

8.4 Oauth 認証のやり方

1. 以下のコードを入力する．ファイル名は auth.py とする．

```
# -*- coding: utf-8 -*-

import tweepy

consumer_key = ""#引用符の中に consumer_key の情報を記述する
consumer_secret = ""#引用符の中に consumer_secret の情報を記述する

access_token = ""#引用符の中に access_token の情報を記述する
access_token_secret = ""#引用符の中に access_token_secret の情報を記述
する

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
```

2. vagrant に ssh 接続した状態で

```
sudo apt-get install python-setuptools python-pip
sudo easy_install tweepy
```

を入力し，インストールを行う．

3. 正常にインストールが行われたのを確認したら，vagrant に ssh で接続した状態で

python auth.py を実行する．

4. 何もコマンドプロンプト上に表示されなければ Oauth 認証成功である．

8.5 TwitterAPI を用いたデータ収集方法

本章では研究で使用するデータを収集する方法の説明をする．

8.5.1 日本人ユーザー 50 人をランダムサンプリングする方法

1. 以下のコードを入力する．ファイル名は ranjp.py とする．

```
# -*- coding: utf-8 -*-

import twitter
```

```
import sqlite3
from random import randint

api = twitter.Api()
count = 0
while count < 50:
    i = randint(1,500000000)
    try:
        user = api.GetUser(i)
    except:
        continue
    if user.time_zone in [u'Osaka', u'Tokyo', u'Sapporo']:
        con.execute('insert into data values (%s, %s, %s)' % (user.statuses_count,
        count += 1
        con.commit()

con.close()
```

2. vagrant に ssh 接続した状態で

   ```
   python ranjp.py
   ```

   を実行する.
3. 結果が出力されれば成功である.

8.5.2 デマツイートをリツイートしたユーザー 50 人を取得する方法

1. 以下のコードを入力する．ファイル名は rtlist.py とする．

```
# -*- coding: utf-8 -*-

import tweepy
import json
import sys
from pprint import pprint
from auth import api

for line in sys.stdin:
  tweetId = line.rstrip()
  sys.stderr.write("checking retweeters of %s...\n" % tweetId)

  リツイートした人を調べたことを記録する。
  sys.stdout.write("update retweets set retweetersChecked=true where id=%s;\n" % (

  try:
    retweeters = api.retweets(tweetId,100)
    for t in retweeters:
     このオブジェクトの中身がわかりにくい。(PyDev のデバッガで調べた。)
      user = t.user
      userId = user.id_str
      screenName = user.screen_name

      sys.stdout.write("values (%s,'%s');\n" % (userId, screenName))#
これは重複エラーになるはず
    sys.stdout.flush()
  except:
    pass
```

2. vagrant に ssh 接続した状態で

   echo tweetid(ツイート固有の ID のこと) | python rtlist.py

   を実行する．
3. 結果が出力されれば成功である．

8.6　2 標本 T 検定の方法

本章では，日本人ユーザー 50 人とデマツイートをリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数の平均の差が，偶然的な誤差の範囲にあるものかどうかを判断する為に 2 標本 T 検定を行う方法を説明する．

8.6.1　データの対応の有無

ｔ検定は 2 つのグループの平均の差が偶然誤差の範囲内にあるかどうかを調べるものである．まず，データに対応があるかどうかで t 検定のやり方が変わってくる．今回は全ての組み合わせでデータの対応がないため，各グループの平均と分散だけからｔ検定を行うことになる．分散がほぼ等しいと見なせる場合と分散が等しいとは見なせない場合に応じて，各々「分散が等しいときのｔ検定」「分散が等しくないときのｔ検定」を適用する．分散が等しいかどうかの判断は F 検定によって行う．

8.6.2　F 検定の方法

以下のデータは日本人ユーザー 50 人とデマツイート 1 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数である．このデータを使って F 検定の方法を説明する．

デマ 1　ランダム日本人ユーザー

```
70 45
21 19
3 26
81 19
81 33
27 15
23 20
44 20
76 14
41 4
100 24
19 25
55 39
55 26
20 45
99 4
73 29
57 9
```

```
69 15
27 2
24 9
98 9
99 38
36 16
30 8
32 28
95 45
74 36
67 30
62 15
72 11
86 6
48 1
64 55
57 26
94 89
26 73
94 9
90 13
63 2
54 11
49 4
37 0
70 1
92 0
13 7
71 9
34 12
61 0
1 6
```

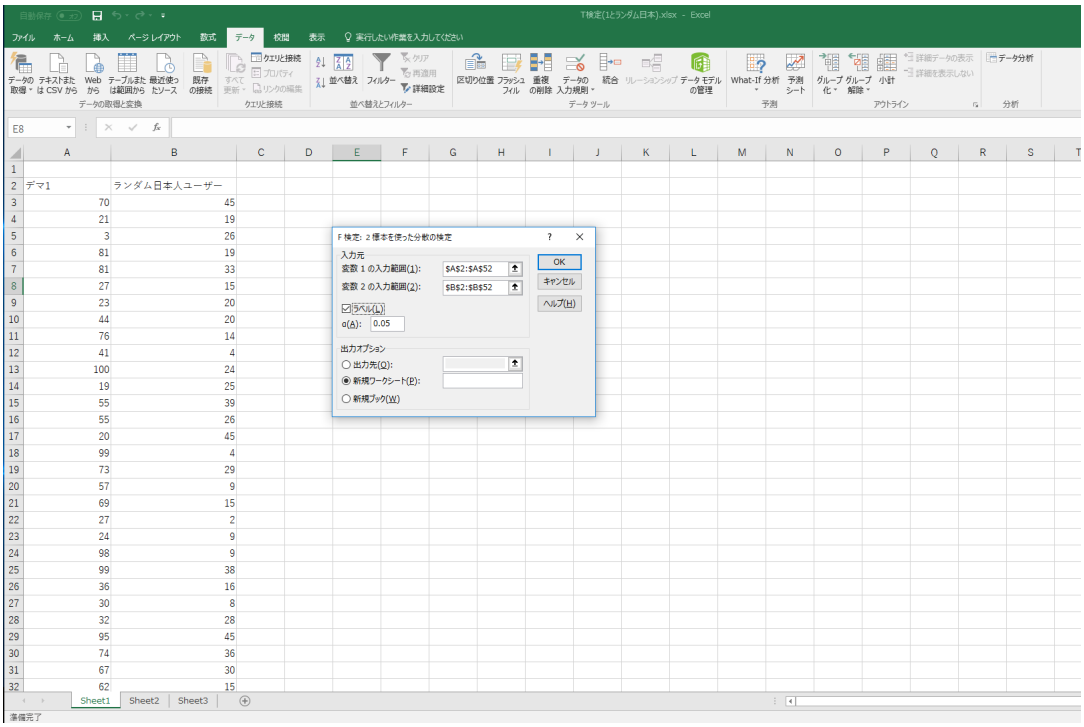
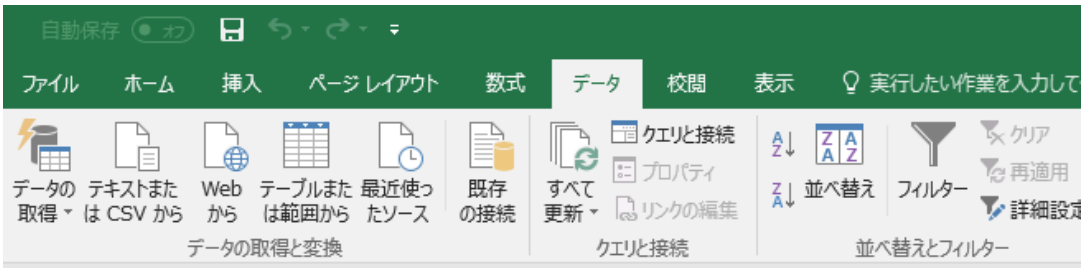


図 8.12　F 検定

Excel のデータタブを開きデータ分析の項目をクリックし，F 検定:2 標本を使った分散の検定を選択．変数 1 の入力範囲としてデマツイート 1 のデータ全て．変数 2 の入力範囲としてランダム日本人ユーザー全てを選択する．ラベルにチェックを入れ，a を 0.05 に設定する．OK をクリックすると結果が表示される．



H19

	A	B	C	D
1	F-検定: 2 標本を使った分散の検定			
2				
3		デマ1	ランダム日本人ユーザー	
4	平均	56.68	20.04	
5	分散	775.3240816	351.5493878	
6	観測数	50	50	
7	自由度	49	49	
8	観測された分散比	2.205448533		
9	P(F<=f) 片側	0.003267396		
10	F 境界値 片側	1.607289463		
11				
12	P<0.05となり帰無仮説を棄却			
13	F境界値<分散比となり帰無仮説を棄却			
14	よって不等分散と見なせる			

図 8.13　F 検定結果

結果から不等分散であることが分かった．次項にて同じデータを例に 2 標本 t 検定の方法を説明する．

8.6.3　2 標本 t 検定の方法

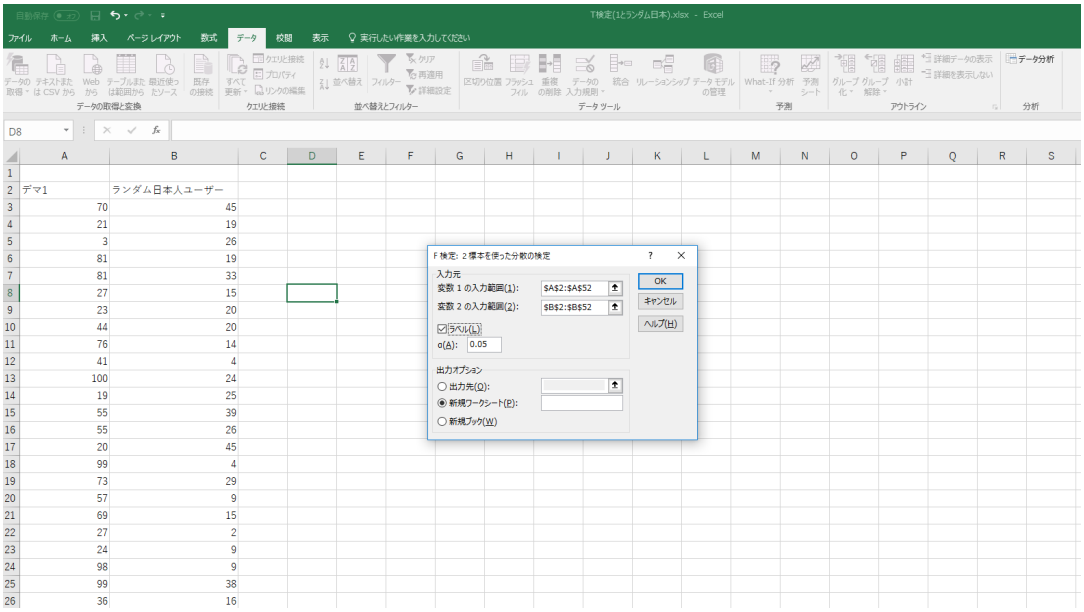
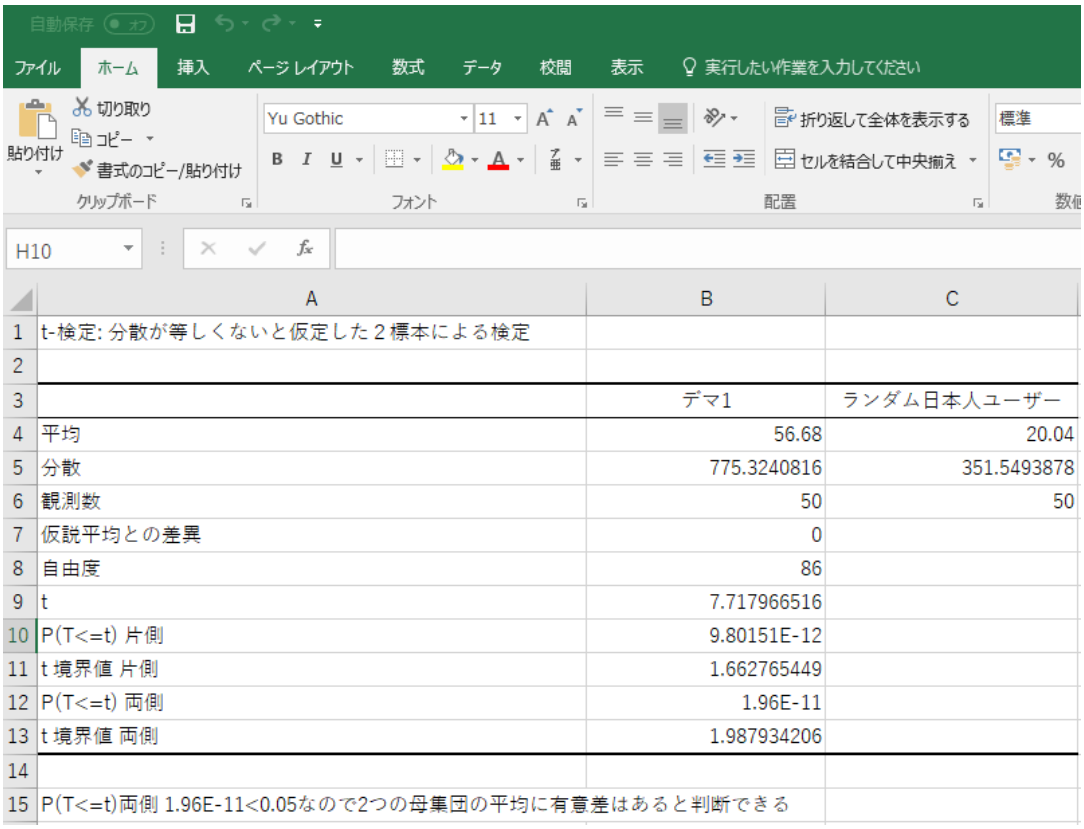


図 8.14　t 検定

Excel のデータタブを開きデータ分析の項目をクリックし，t 検定:分散が等しくないと仮定した 2 標本による検定を選択．変数 1 の入力範囲としてデマツイート 1 のデータ全て．変数 2 の入力範囲としてランダム日本人ユーザー全てを選択する．ラベルにチェックを入れ，a を 0.05 に設定する．OK をクリックすると結果が表示される．



	A	B	C
1	t-検定: 分散が等しくないと仮定した２標本による検定		
2			
3		デマ1	ランダム日本人ユーザー
4	平均	56.68	20.04
5	分散	775.3240816	351.5493878
6	観測数	50	50
7	仮説平均との差異	0	
8	自由度	86	
9	t	7.717966516	
10	P(T<=t) 片側	9.80151E-12	
11	t 境界値 片側	1.662765449	
12	P(T<=t) 両側	1.96E-11	
13	t 境界値 両側	1.987934206	
14			
15	P(T<=t)両側 1.96E-11<0.05なので2つの母集団の平均に有意差はあると判断できる		

図 8.15　t 検定結果

結果から 2 つの母集団の平均に有意差はあると分かった.

第 9 章

結果

9.1　日本人ユーザ 50 人をランダムサンプリングした結果

日本人ユーザ 50 人をランダムサンプリングした結果以下のようになった.

ランダムに取得した日本人ユーザ，直近 100 ツイートの RT 数
myaowmyaow0503 , 45
karasukun_, 19
Toradora_ryo, 26
subakura_____8, 19
star_hibiki, 33
minsu519, 15
m_arimo0, 20
ao_tyou, 20
xMeq__, 14
spla_balsamico, 4
Kmnoka43, 24
ngbba06, 25
kzk_PC, 39
a__xxx, 26
spdk5yn9miki, 45
JAfoINY3Czn4LV4, 4
__zxwy, 29
blacklotus_krhs, 9
boncalpis, 15
marchan_impulse, 2
yellow____1103, 9
usss___umi, 9
Miu_NGZ4602, 38
tora_prpr, 16

yumehiko_mugen, 8
HACHIGORO2011, 28
kk_cain, 45
chii___ms, 36
fog_wolves, 30
sqUmSkAnqmSjhUF, 15
sd_Ikeda_Luna, 11
raririarimann, 6
howusee_it70, 1
mrsk_o3o, 55
i__k0603, 26
rikuxsakura, 89
kaho05, 73
ryhki, 9
hagi28, 13
atlach_nacha495, 2
yuniba258, 11
okusaredori, 4
hizidsa, 0
SteedVLX, 1
10ac_mttk, 0
mw_92184, 7
ichiha1126, 9
825524_rin, 12
haruka_syue, 0
EcNami, 6

階級	人数
0 - 10	19
11 - 20	13
21 - 30	8
31 - 40	4
41 - 50	3
51 - 60	1
61 - 70	0
71 - 80	1
81 - 90	1
91 - 100	0
平均RT数	20.04

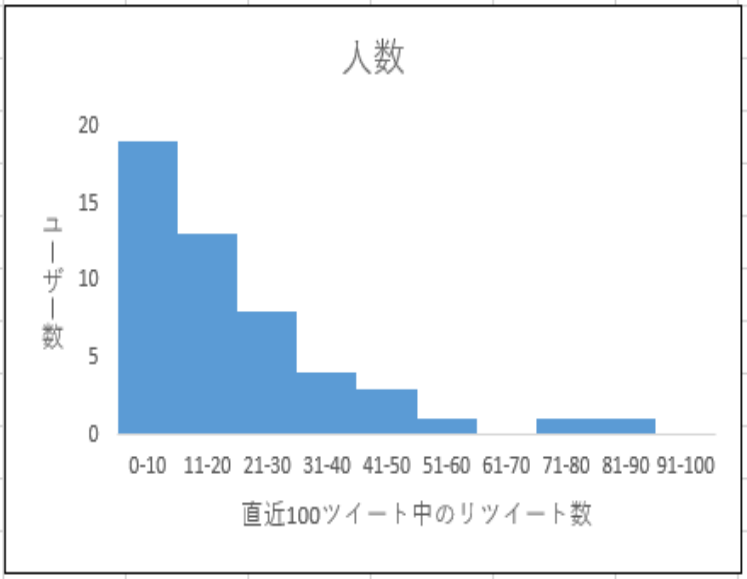


図 9.1　日本人ユーザ 50 人の平均 RT 数とヒストグラム

9.2 デマツイート 1 をリツイートしたユーザー 50 人を取得した結果

デマツイート 1 をリツイートしたユーザー 50 人を取得した結果以下のようになった.

デマツイート 1 を RT したユーザー，直近 100 ツイートの RT 数
mnmnssk, 70
yu612keisei86, 21
poohshan7427, 3
Yuion0016871, 81
EX24TA, 81
ohta_isan_, 27
rassyrassy333, 23
BlueMAX_VJ, 44
piyo2jack, 76
atuto__, 41
izumo_abc, 100
Anmr118, 19
teruteruterutti, 55
QqQbc, 55
Akaiyu_yuki, 20
sayo_candy, 99
generalfrost_6, 73
aiki_kc, 57
masayuki311, 69
shisyotosyo, 27
yasui0324, 24
8_wtkq, 98
anewstart559, 99
kazuki_hami, 36
yeahtwmjgw, 30
saigyoa, 32
mutu_matu01, 95
nmemaksumlk, 74
saka_s1127, 67
ossan_dai, 62
yuta22074, 72
jihh00, 86
DnrodReds, 48

yukinyooooooooon, 64
ncruncham, 57
qeg5P03kABpgx9X, 94
muginami_nolito, 26
toshi2497315, 94
masash1324xx88, 90
bivibeau, 63
ntake1966, 54
akiya_kiu, 49
mamo_wa0608, 37
kokonoeringo05, 70
Fluorumium, 92
naukosi, 13
okashi_tabenai, 71
gu_tara12yume, 34
KotobukiSonbe, 61
sudachi_1993, 1

階級	人数
0 - 10	2
11 - 20	3
21 - 30	7
31 - 40	4
41 - 50	4
51 - 60	5
61 - 70	8
71 - 80	5
81 - 90	4
91 - 100	8
平均RT数	56.68

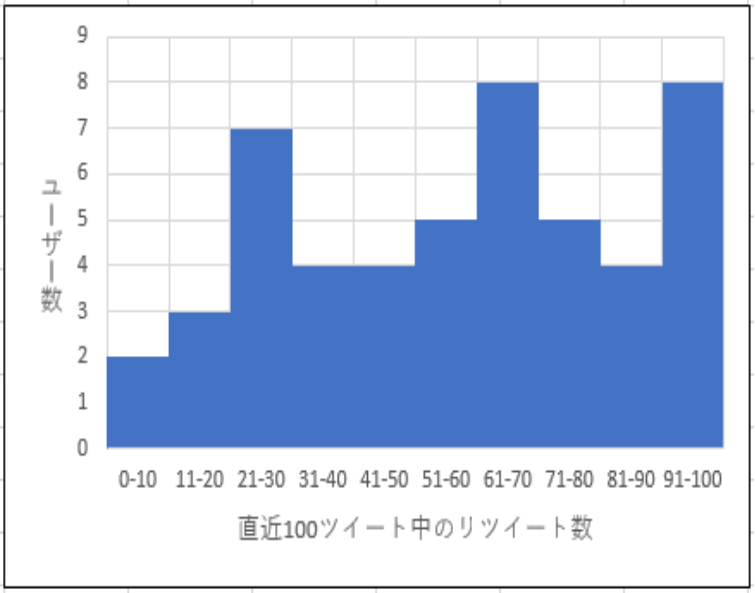


図 9.2　デマツイート 1 をリツイートしたユーザー 50 人の平均 RT 数とヒストグラム

9.3 デマツイート 2 をリツイートしたユーザー 50 人を取得した結果

デマツイート 2 をリツイートしたユーザー 50 人を取得した結果以下のようになった.

デマツイート 2 を RT したユーザー，直近 100 ツイートの RT 数
mmm_sheep, 25
karan_1126, 58
peachmint753, 85
choc8o_o8, 52
1010__, 17
koutyou_03, 95
Aliiiis_, 36
nishinosonoyo, 58
Ares_Vermillion, 94
motipeen_424, 60
sihugonomi, 98
nazuna17, 88
rasetunkamen, 85
kotatubutonn018, 76
B_M_Graphix, 100
floral00, 78
MkGftf, 100
namako1221yu, 14
moto_oolong_tea, 83
cnnwp051, 89
oita_shukuri, 82
recureat, 32
mocha_ras, 100
zmdmrg, 100
RYUGASAKI_Akira, 28
goumonnkigu, 88
NPO94867351, 13
tatew3druk3, 30
SSSSPIRALLIFE, 76
asuka_toritori, 85
AnatH_69, 31
applegesu12, 43
bettyboo_redeye, 48

o_mocco, 2
dazjelly, 23
curacco, 20
kabech, 11
tibikkohime, 18
aohina_1023, 94
KoizumiRiver, 86
isana777, 67
ahirumaru88, 98
kairidei, 58
heta0alisa, 64
zitomemania, 93
DerBlauer_Vogel, 89
26Gbs, 75
harumi21kugyu11, 66
saigusaalter11, 49
misatsuki, 72

階級	人数
0 - 10	1
11 - 20	6
21 - 30	4
31 - 40	3
41 - 50	3
51 - 60	5
61 - 70	3
71 - 80	5
81 - 90	10
91 - 100	10
平均RT数	62.64

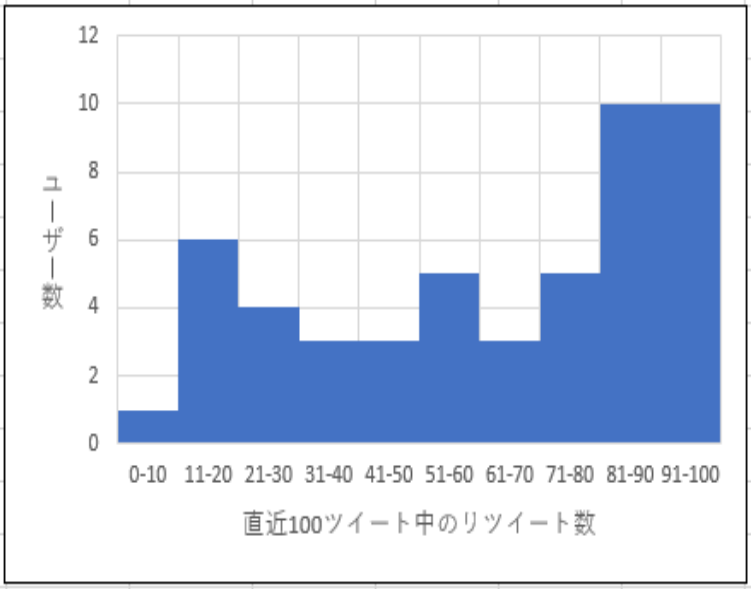


図 9.3　デマツイート 2 をリツイートしたユーザー 50 人の平均 RT 数とヒストグラム

9.4 デマツイート 3 をリツイートしたユーザー 50 人を取得した結果

デマツイート 3 をリツイートしたユーザー 50 人を取得した結果以下のようになった.

デマツイート 3 を RT したユーザー，直近 100 ツイートの RT 数
amaminoya, 39
province_mrba, 54
DoLcs888, 98
Bea5851, 54
tetugunn, 92
rFTgM63P18EJ2BA, 41
bisuko_12, 28
kaneko_chang, 28
fuu_mst, 48
cmtaka124, 99
VittoBennetPso2, 91
onkyo22yoshiki, 48
ChihironYY, 8
kitauming, 65
iria1013yd, 90
shiki3861, 80
BWorld1016, 48
wing34nina, 67
raiji3510, 57
yama_oki_, 84
itkr__, 54
lllPembrokelll, 3
seojin_ksjim35, 100
chiico33, 45
_HHH__, 56
rizain1192, 93
groovenUG, 98
iiwashi69pantu, 27
n_woE, 53
tuji_gami, 52
nui_896, 36
mijimiji05, 12
Kookie_Jams, 94

hana2657, 100
nakayama_GOGO, 48
kgmnkzklrn, 74
12Rrg, 67
zoiqq009, 47
Bboycaramel0413, 50
paxcco, 29
tarezousan47, 55
purebreed1, 78
vervkwsm, 79
kk_segreto, 34
magamirei7, 25
oogamikotarou, 26
Jyeido_Rezary, 81
SIKI_495, 90
yama_824, 41
sola_sk_0, 57

階級	人数
0 - 10	2
11 - 20	1
21 - 30	6
31 - 40	3
41 - 50	9
51 - 60	9
61 - 70	3
71 - 80	4
81 - 90	4
91 - 100	9
平均RT数	58.46

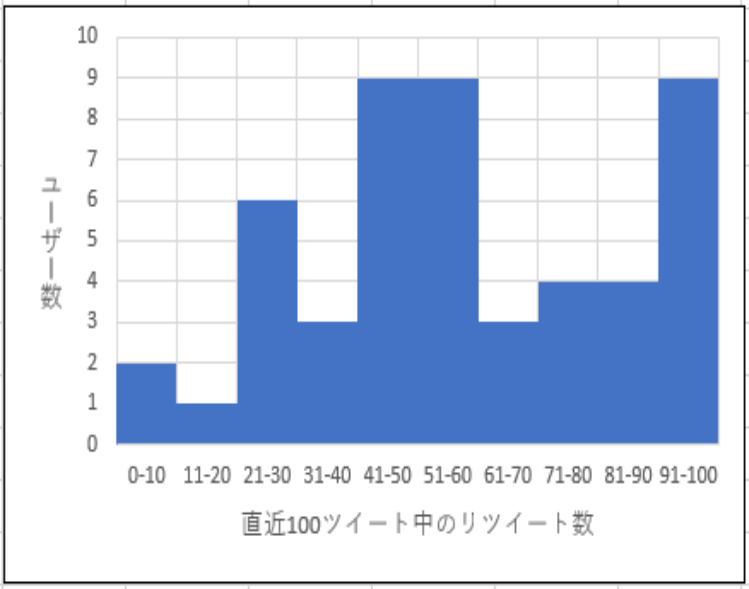


図 9.4　デマツイート 3 をリツイートしたユーザー 50 人の平均 RT 数とヒストグラム

9.5 デマツイート 4 をリツイートしたユーザー 50 人を取得した結果

デマツイート 4 をリツイートしたユーザー 50 人を取得した結果以下のようになった.

デマツイート 4 を RT したユーザー，直近 100 ツイートの RT 数
noncats, 72
ruusan203, 43
Choco_tGZep3q, 21
WolfHound_M40A3, 87
suchas1219n, 87
yzmkn_2804, 60
wahuxu_1002, 48
fisss_, 10
mimohako_0616, 15
wallcroft039400, 10
OpBnT9Sdre7jCUJ, 49
kai_akatuki, 91
mina0701kana11, 30
UK_Arther, 44
6moonH, 86
rihama_tukumo99, 13
pet_shimobe, 44
f9u6yq, 82
krmr613, 21
zakokyara1112, 48
otz_69, 37
rin0613neya, 20
6loSYqgdfJmFQjP, 100
htH61uAJexXFloZ, 98
chipakan, 93
totoko1981, 93
ozigisou1028, 94
sonicstar3739, 74
zounitoburi, 26
k66yukina, 36
crimsondolls, 66
1010yazukide41, 96
ZjMsy, 95

kituneme_, 72
rena__rena__re, 61
ventwo_two, 60
kisaragisuzuma7, 59
kin_102, 65
kuisinboo24, 54
0yHUkIgVjjbwOU4, 90
Iliyaerja, 49
younnri, 49
dddaaa315, 72
mizubonarigatou, 46
GilgitGlaucous, 89
sakurablue1126, 78
early_ana_, 95
fakeshe, 30
FC_Nekomaturi, 25
AlithinCOF, 13

階級	人数
0 - 10	2
11 - 20	4
21 - 30	6
31 - 40	2
41 - 50	9
51 - 60	4
61 - 70	3
71 - 80	5
81 - 90	6
91 - 100	9
平均RT数	57.92

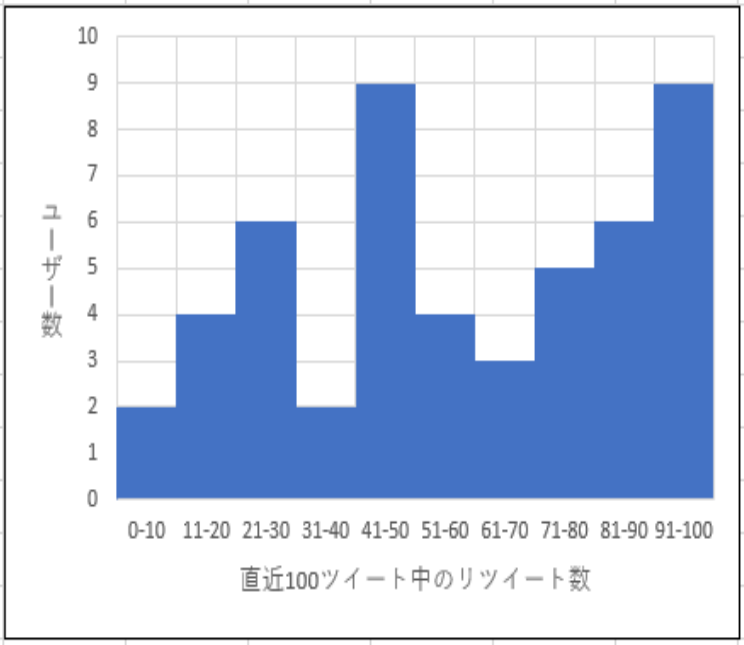


図 9.5　デマツイート 4 をリツイートしたユーザー 50 人の平均 RT 数とヒストグラム

9.6　F 検定の結果 1

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 1 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で F 検定を行った結果を以下に示す.

F-検定: 2 標本を使った分散の検定		
	デマ1	ランダム日本人ユーザー
平均	56.68	20.04
分散	775.3240816	351.5493878
観測数	50	50
自由度	49	49
観測された分散比	2.205448533	
P(F<=f) 片側	0.003267396	
F 境界値 片側	1.607289463	
P<0.05となり帰無仮説を棄却		
F境界値<分散比となり帰無仮説を棄却		
よって不等分散と見なせる		

図 9.6　F 検定結果 1

9.7　F 検定の結果 2

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 2 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で F 検定を行った結果を以下に示す.

F-検定: 2 標本を使った分散の検定		
	デマ2	ランダム日本人ユーザー
平均	62.64	20.04
分散	904.8473469	351.5493878
観測数	50	50
自由度	49	49
観測された分散比	2.573884007	
P(F<=f) 片側	0.000610309	
F 境界値 片側	1.607289463	
P<0.05となり帰無仮説を棄却		
F境界値<分散比となり帰無仮説を棄却		
よって不等分散と見なせる		

図 9.7　F 検定結果 2

9.8　F 検定の結果 3

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 3 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で F 検定を行った結果を以下に示す.

F-検定: 2 標本を使った分散の検定		
	デマ3	ランダム日本人ユーザー
平均	58.46	20.04
分散	702.9065306	351.5493878
観測数	50	50
自由度	49	49
観測された分散比	1.99945315	
P(F<=f) 片側	0.008439378	
F 境界値 片側	1.607289463	
P<0.05となり帰無仮説を棄却		
F境界値<分散比となり帰無仮説を棄却		
よって不等分散と見なせる		

図 9.8　F 検定結果 3

9.9　F 検定の結果 4

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 4 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で F 検定を行った結果を以下に示す.

F-検定: 2 標本を使った分散の検定		
	デマ4	ランダム日本
平均	57.92	20.04
分散	801.9934694	351.5493878
観測数	50	50
自由度	49	49
観測された分散比	2.281310955	
P(F<=f) 片側	0.002306196	
F 境界値 片側	1.607289463	
P<0.05となり帰無仮説を棄却		
F境界値<分散比となり帰無仮説を棄却		
よって不等分散と見なせる		

図 9.9　F 検定結果 4

9.10　2 標本 t 検定の結果 1

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 1 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で 2 標本 t 検定を行った結果を以下に示す.

t-検定: 分散が等しくないと仮定した 2 標本による検定		
	デマ1	ランダム日本人ユーザー
平均	56.68	20.04
分散	775.3240816	351.5493878
観測数	50	50
仮説平均との差異	0	
自由度	86	
t	7.717966516	
P(T<=t) 片側	9.80151E-12	
t 境界値 片側	1.662765449	
P(T<=t) 両側	1.96E-11	
t 境界値 両側	1.987934206	
P(T<=t)両側 1.96E-11<0.05なので2つの母集団の平均に有意差はあると判断できる		

図 9.10　t 検定結果 1

9.11　2 標本 t 検定の結果 2

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 2 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で 2 標本 t 検定を行った結果を以下に示す.

t-検定: 分散が等しくないと仮定した 2 標本による検定		
	デマ2	ランダム日本人ユーザー
平均	62.64	20.04
分散	904.8473469	351.5493878
観測数	50	50
仮説平均との差異	0	
自由度	82	
t	8.498283243	
P(T<=t) 片側	3.58291E-13	
t 境界値 片側	1.663649184	
P(T<=t) 両側	7.17E-13	
t 境界値 両側	1.989318557	
P(T<=t)両側7.17E-13<0.05なので2つの母集団の平均に有意差はあると判断できる		

図 9.11　t 検定結果 2

9.12　2 標本 t 検定の結果 3

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 3 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で 2 標本 t 検定を行った結果を以下に示す.

t-検定: 分散が等しくないと仮定した２標本による検定		
	デマ3	ランダム日本人ユーザー
平均	58.46	20.04
分散	702.9065306	351.5493878
観測数	50	50
仮説平均との差異	0	
自由度	88	
t	8.366198186	
P(T<=t) 片側	4.12565E-13	
t 境界値 片側	1.662354029	
P(T<=t) 両側	8.2513E-13	
t 境界値 両側	1.987289865	
P(T<=t)両側 8.2513E-13<0.05なので2つの母集団の平均に有意差はあると判断できる		

図 9.12　t 検定結果 3

9.13　2 標本 t 検定の結果 4

ランダムサンプリングした日本人ユーザー 50 人とデマツイート 4 をリツイートしたユーザー 50 人の最新 100 ツイートに含まれるリツイートの数で 2 標本 t 検定を行った結果を以下に示す.

t-検定: 分散が等しくないと仮定した 2 標本による検定		
	デマ4	ランダム日本
平均	57.92	20.04
分散	801.9934694	351.5493878
観測数	50	50
仮説平均との差異	0	
自由度	85	
t	7.886387371	
P(T<=t) 片側	4.78905E-12	
t 境界値 片側	1.6629785	
P(T<=t) 両側	9.58E-12	
t 境界値 両側	1.988267907	
P(T<=t)両側9.58E-12<0.05なので2つの母集団の平均に有意差はあると判断できる		

図 9.13　t 検定結果 4

第 10 章

考察

デマを拡散するようなユーザに共通する特徴として，リツイート数に着目しランダムサンプリングしたユーザーとデマツイートをリツイートしたユーザーで直近 100 リツイート内のリツイート数を調査した結果，大きく数値が異なった．この結果からデマを拡散するようなユーザーはリツイート機能を多用する傾向にあり，ツイート内容の真偽を確かめる前にリツイートをし，デマ拡散者の一員となっていると考えられる．

自分がデマ拡散者にならない為の手段として，デマ拡散ユーザーリストにあるユーザーと，リツイートの多いユーザーを排除することが有効だと考えられる．

第 11 章

結論

本研究では，デマが拡散されることを防ぐために，デマツイートをリツイートしているユーザーの特徴抽出としてリツイート数の調査を行った．その結果，デマツイートを拡散するユーザーの特徴として，ツイートに占めるリツイートの割合が高いことが確認できた．このような知識を活用することで，Twitter を閲覧する際に，デマツイートを真に受けて拡散してしまうリスクを下げられることが期待できる．

参考文献

[1] 荒川唯, 亀田尭宙, 相澤彰子, 鈴木崇史. Retweet に着目した広がりやすい Tweet の特徴分析. 情報処理学会第 74 回全国大会講演論文集, Vol. 2012, No. 1, pp. 617–618, 2012.

[2] 榎本光, 内田理, 鳥海不二夫. O-054 東日本大震災時のツイート分析によるデマ判別に有用な特徴抽出 (O 分野:情報システム, 一般論文). 情報科学技術フォーラム講演論文集, Vol. 12, No. 4, pp. 649–650, 2013.

謝辞

本研究を進めるにあたり，矢吹研究室矢吹太朗准教授には，多くの時間をご指導にさいて頂きました．また矢吹研究室の皆様には，多くの知識や示唆を頂きました．協力していただい皆様に感謝の気持ちと御礼を申し上げます．