

調達資金の時間変化測定に基づくクラウドファン ディングの成功要因分析

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
1342066
島田樹

目次

第 1 章	序論	2
第 2 章	背景	3
第 3 章	クラウドファンディングについて	4
3.1	クラウドファンディングとは	4
3.2	クラウドファンディングの種類	5
3.3	クラウドファンディングの歴史	5
第 4 章	目的	7
第 5 章	手法	8
5.1	Chocolatey について	9
5.2	VirtualBox について	13
5.3	VirtualBox のインストール	14
5.4	Vagrant について	16
5.5	R 言語について	17
5.6	クローラーについて	21
5.7	クローラーの運用	27
5.8	調査方法	37
第 6 章	結果	60
第 7 章	考察	61
第 8 章	結論	62
	参考文献	63
	謝辞	64

第 1 章

序論

当研究は、調達資金の時間変化測定に基づくクラウドファンディングの成功要因分析を行う。クラウドファンディングはプロジェクトの資金をインターネットを通じて、不特定多数から募る手法である。クラウドファンディングを活用することで、製品やサービスの開発に必要な資金を集めることができるだけでなく、より多くの人へ向けて自身の背品やサービスを認知してもらうことができる。当研究では、プロジェクト実行者が資金を集めるために行っている行動を調査する。調達資金を可視化し、実行者がした行動を分析することによって成功要因を導き出す。

第 2 章

背景

クラウドファンディングは、SNS の発達に伴いプロジェクトの数も増加し、市場も年々増加している [1]。幅広い分野と規模での応募が可能で、ベンチャー企業のプロジェクトや学生の研究費用の獲得などが多かったが、大手企業も支援者数から売れることを確実視されたプロダクトを販売者できるとして、マーケティングの一環として活用されるようになってきた。

第 3 章

クラウドファンディングについて

3.1 クラウドファンディングとは

クラウドファンディングとは [2]，群衆と資金調達を組み合わせた造語で，クリエイターや起業家が製品・サービスの開発，もしくはアイデアの実現などの「ある目的」のために，インターネットを通じて不特定多数の人から資金の出資や協力を募ることをいう．

例として何か製品を作る場合に，何故その製品を作るのか，どのような製品を作るのか，どのように作っていくのか，資金はいくら必要なのか，といった情報をプロジェクトとしてクラウドファンディングサービス上に掲載する．出資してくれた人に対して何か見返り（リターン）がある場合はその旨も記載する．一定期間の間に，プロジェクトに共感した複数人の支援者が少額ずつ資金を出資・支援し，目的の資金が集まった時点でプロジェクトが成立し，プロジェクトの起案者は，集まった資金を元手にプロジェクトを実行する．その際，プロジェクト起案者は，サービス運営者に，集まった金額の 10～20 %を手数料として支払う．

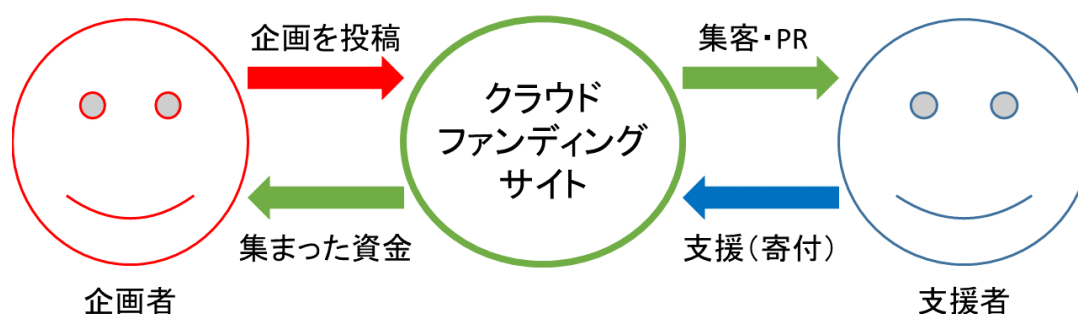


図 3.1 クラウドファンディングの仕組み

3.2 クラウドファンディングの種類

クラウドファンディングは，一般的には支援者に対するリターンによって 3 つに分類される [3] ．

- 金銭的リターンのない 「寄付型」
- 金銭的リターンのある 「投資型」
- 権利や物品を購入することで支援する 「金融型」

3.2.1 寄付型クラウドファンディング

寄付型クラウドファンディングは，プロジェクトに対して出資を行うが，あくまでも寄付であるためリターンは発生しない．被災地支援や発展途上国の支援などの社会的意義の大きいプロジェクトとの相性が高い．通常の寄付であれば，寄付金の使い道が分かりづらいことが多いが，クラウドファンディングを利用することで，寄付後のプロジェクトの状況が透明化される．

3.2.2 購入型クラウドファンディング

購入型クラウドファンディングは，支援者がプロジェクトへの出資することで支援金額に応じた金銭以外の商品やサービスが手に入る．クラウドファンディングは，プロジェクト成功後に商品の作成が多いので，先行販売という形に近い．日本国内で一番採用されているタイプである．

3.2.3 金融型クラウドファンディング

金融型クラウドファンディングは，支援者が特定の企業などに出資を行い，リターンとして金銭（配当や利益の一部）または株式が発行される．今までは比較的大きな投資などが必要で，個人ではハードルが高かった投資を小口から出資できるようになったことで誰でも投資活動ができるようになるのが特徴である．誰でも未公開企業に対して株式という形で投資ができるので，非常に注目が集まっているが，日本においては 2014 年に金融商品取引法が改正されるまでは寄付型と購入型に限られていた．

3.3 クラウドファンディングの歴史

クラウドファンディングのような仕組みは，17 世紀初頭から始まっている．書籍編集者のジョン・テイラー氏が，書籍の印刷代を寄付によって集めた事例がクラウドファンディングの原型と言われている [4] ．そしてアメリカで 2006 年に企業家のマイケル・サリバンが初めて「クラウドファンディング」という言葉を用いた．大手クラウドファンディングサービスは，2008 年に「IndieGoGo」，翌年 2009 年に「Kickstarter」がリリースされてい

る．日本では 2011 年 4 月にリリースされた「READYFOR」が最初のクラウドファンディングサービスである．

第 4 章

目的

先行研究 [5] ではプロジェクトの内容以外の要因を調査し，目標金額等の設定段階で成功率を上げることを目的としていた．本研究では調達資金の時間変化を可視化し，動画の投稿や SNS での告知などの多くの資金を集める直前に行っているプロジェクト実行者の行動を分析する．その結果から，プロジェクト実行者が資金を集める際の参考となる指標を作ることを目的とする．

第 5 章

手法

初めにクラウドファンディングサイトを，毎日定時に監視し，データ収集を行う．収集したデータから成功しているプロジェクトの資金調達を可視化し，資金が集まり始めるときにしている行動を調査する．調査項目は，サイト内のレポートを活用しているか，動画を投稿しているか，Twitter でツイートをしているか，Facebook で投稿しているか，の 4 項目である．この 4 項目に調達金額を目標金額で割った目標達成率を足した 5 項目から決定木を作成し，調達資金に必要な行動を考察する．

5.1 Chocolatey について

5.1.1 Chocolatey とは

Chocolatey は Windows のためのパッケージ管理ツールである。Chocolatey を使うことによって Windows 上で動作するソフトウェアをコマンドラインからインストール、アンインストール、アップデート、検索を行うことができる。

5.1.2 Chocolatey のインストール

Windows のコマンドプロンプトを管理者として実行する。

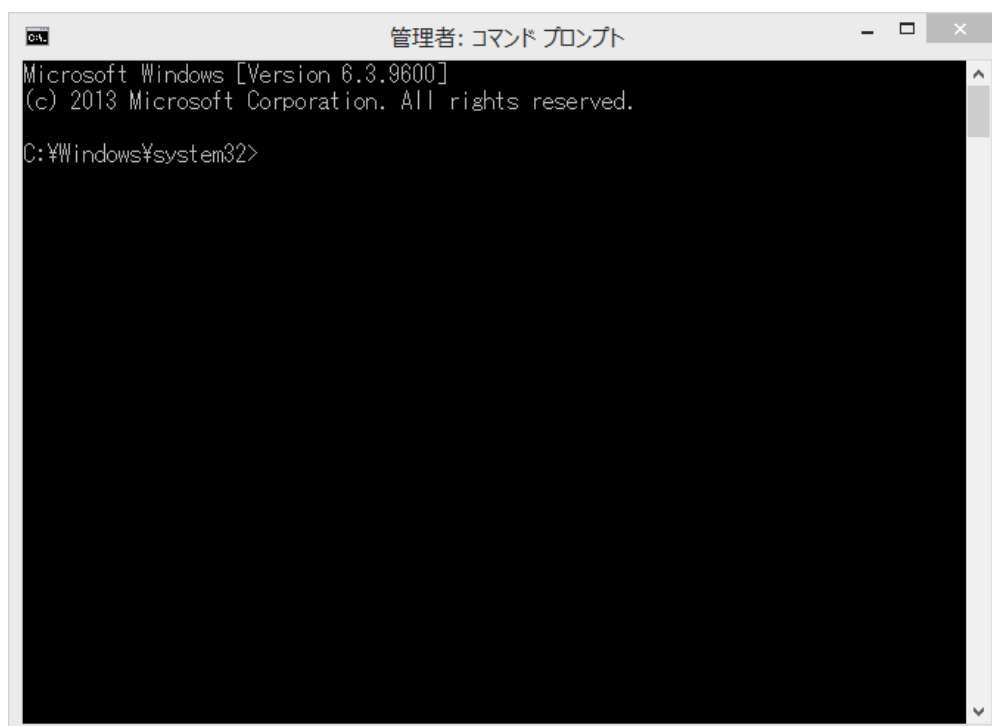


図 5.1 管理者で起動したコマンドプロンプト

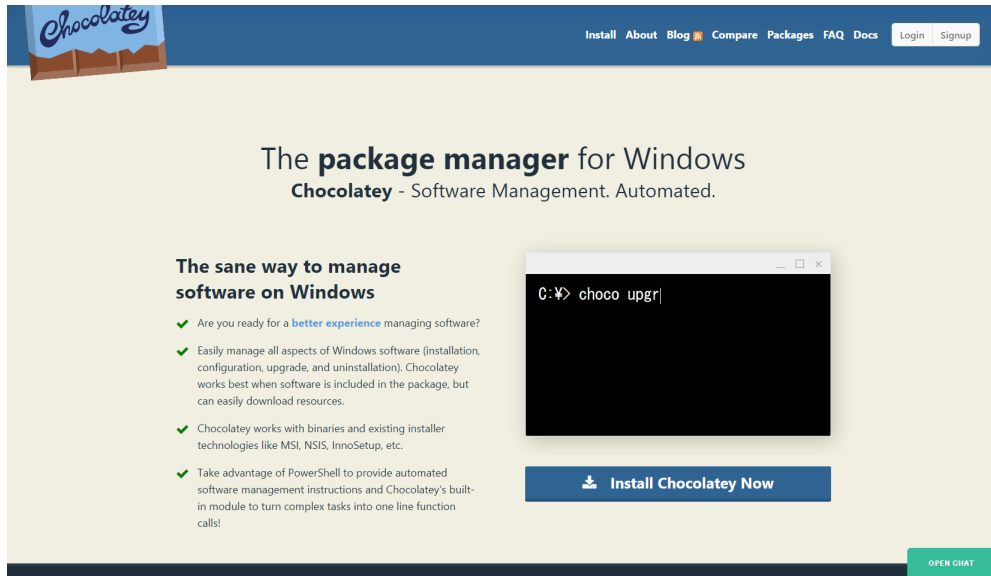


図 5.2 ダウンロードサイト 1

chocolatey のサイト <https://chocolatey.org/> に掲載されているコマンドを実行する.

Chocolatey

Install About Blog Compare Packages FAQ Docs Login Signup

Installing Chocolatey

Chocolatey installs in seconds...

NOTE: Need to install a particular version of Chocolatey? Proxy? Install to a different location? Advanced Options? See [Before You Install](#) below. [Alternative installation options?](#) [Non-Administrator installation?](#) [Restricted/no internet access on target machine?](#)

To install chocolatey now, open an **administrative** command prompt and paste the text from the box below that applies to the name of your shell and press enter. If you need assistance opening an administrative prompt, see [open an elevated prompt in Windows 8+](#) (or [Windows 7](#)).

NOTE: Please inspect <https://chocolatey.org/install.ps1> prior to running any of these scripts to ensure safety. We already know it's safe, but you should also be comfortable before running **any** script from the internet you are not familiar with. All of these scripts download a remote PowerShell script and execute it on your machine.

Cmd.exe -

```
@powershell -NoProfile -ExecutionPolicy Bypass -Command "&{iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))} && SET "PATH=%PATH%;%ALLUSERSPROFILE%"
```

PowerShell.exe (Ensure [Get-ExecutionPolicy](#) is at least RemoteSigned) -

```
iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

PowerShell v3+ (Ensure [Get-ExecutionPolicy](#) is at least RemoteSigned) -

```
iwr https://chocolatey.org/install.ps1 -UseBasicParsing | iex
```

The easiest option to remember is this one. You may not need `-UseBasicParsing`.

What are these scripts doing? `iwr (Invoke-WebRequest) / WebClient.DownloadString` downloads the install script and passes it to `iex (Invoke-Expression)` to execute the contents of the script. This runs the installation script for Chocolatey.

[Installing Behind a Proxy?](#) [OPEN CHAT](#)

図 5.3 ダウンロードサイト 2

ダウンロードサイトの Install Chocolatey Now をクリックする

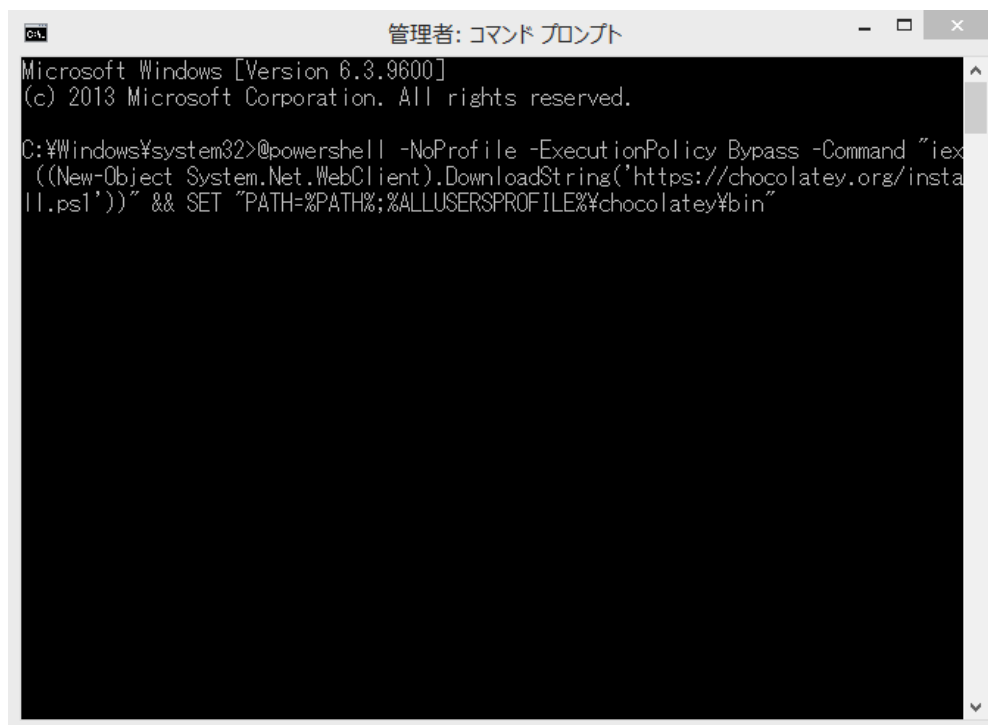


図 5.4 コマンドプロンプト

Cmd.exe に書かれているコマンドをコピーし，起動したコマンドプロンプトで実行する.

5.2 VirtualBox について

5.2.1 VirtualBox とは

VirtualBox は、使用している PC 上に仮想的な PC を作成し、別の OS をインストール・実行できるフリーの仮想化ソフトである。

VirtualBox はコンピュータ上で直接動作している通常の OS にとってはアプリケーションソフトの一つであり、他のソフトと同じように起動することができる。起動すると仮想的なコンピュータが構築され、元の OS とは独立に別の OS を起動することができる。VirtualBox が実行されている OS をホスト OS、VirtualBox 上で実行されている OS をゲスト OS という。

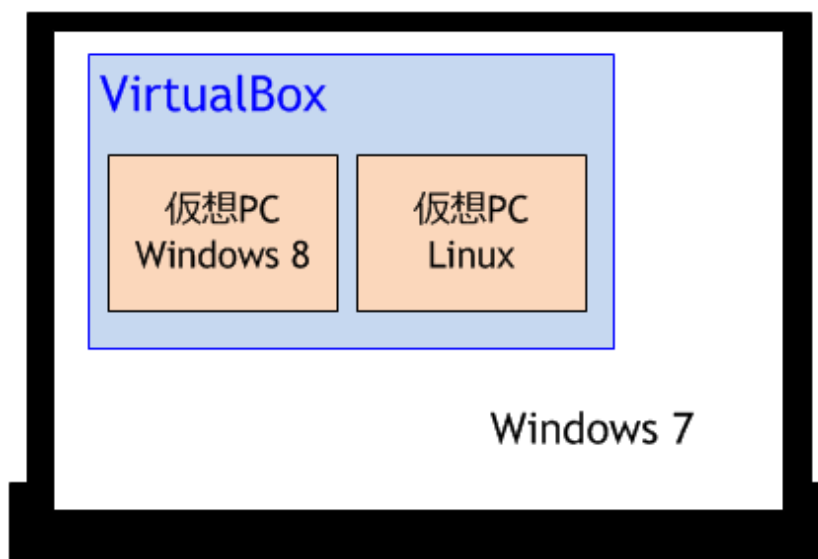


図 5.5 仮想マシンのイメージ

元は独立系のソフトウェア企業が開発・販売していた製品だったが、開発元が Sun Microsystems 社に買収され、その後同社が Oracle 社に買収されたため、Oracle 社が開発元となり、正式名称も「Oracle VM VirtualBox」となった。また、VirtualBox 本体は GPL に基いてオープンソースソフトウェアとして公開され、誰でも自由に入手・利用・改変・再配布などが行える。同社では VirtualBox に機能を追加するソフトウェアを製品として開発・販売している。

5.3 VirtualBox のインストール

本研究では Chocolatey を使用してインストールする。
コマンドプロンプトを管理者で起動する。

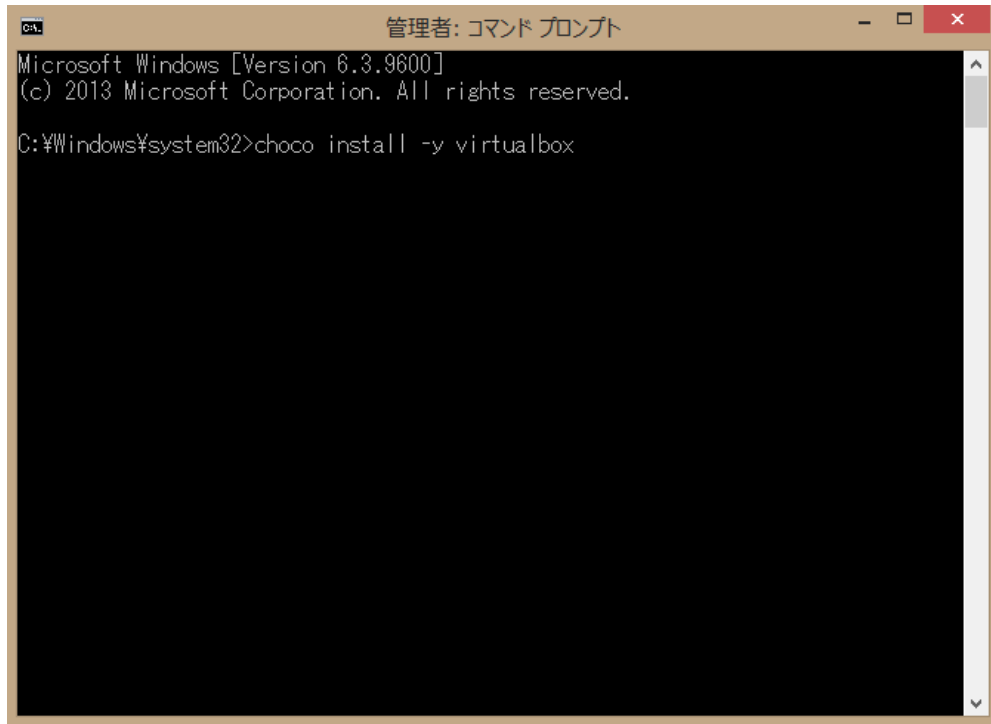
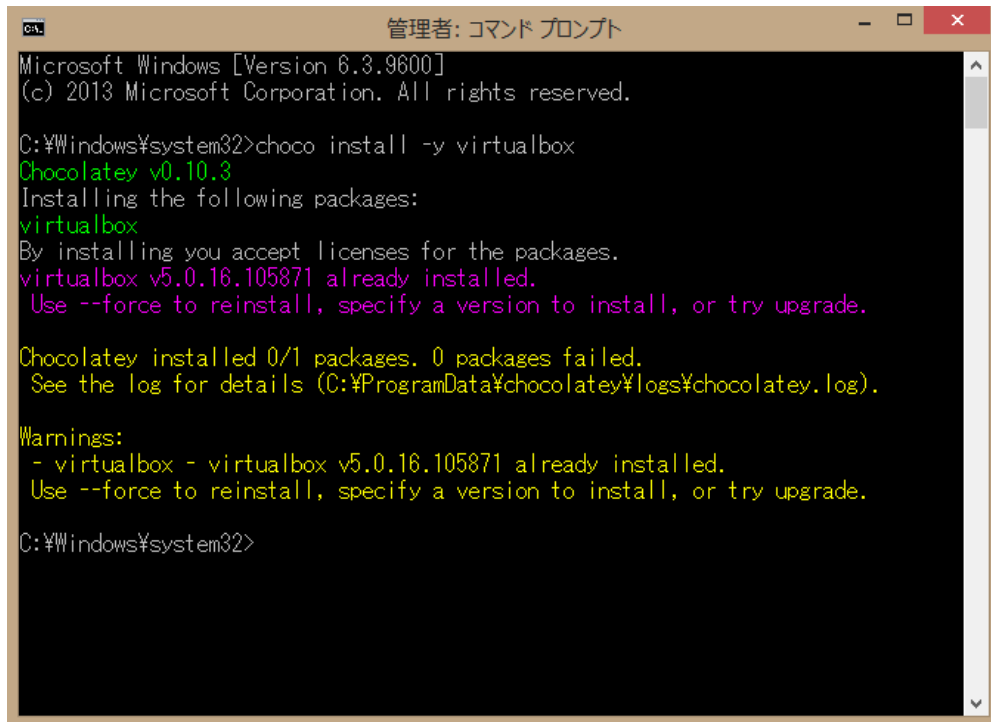


図 5.6 VirtualBox のインストール 1

choco install -y virtualbox と入力し実行する



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>choco install -y virtualbox
Chocolatey v0.10.3
Installing the following packages:
virtualbox
By installing you accept licenses for the packages.
virtualbox v5.0.16.105871 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages. 0 packages failed.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- virtualbox - virtualbox v5.0.16.105871 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

C:\Windows\system32>
```

図 5.7 VirtualBox のインストール 2

このようになればインストール完了

5.4 Vagrant について

5.4.1 Vagrant とは

Vagrant とは、仮想環境を作成するにあたって、簡単に構築・管理し配布することができるツールである。VirtualBox と Vagrant の 2 つがあることで、開発環境を仮想マシン上に自動作成することができる。

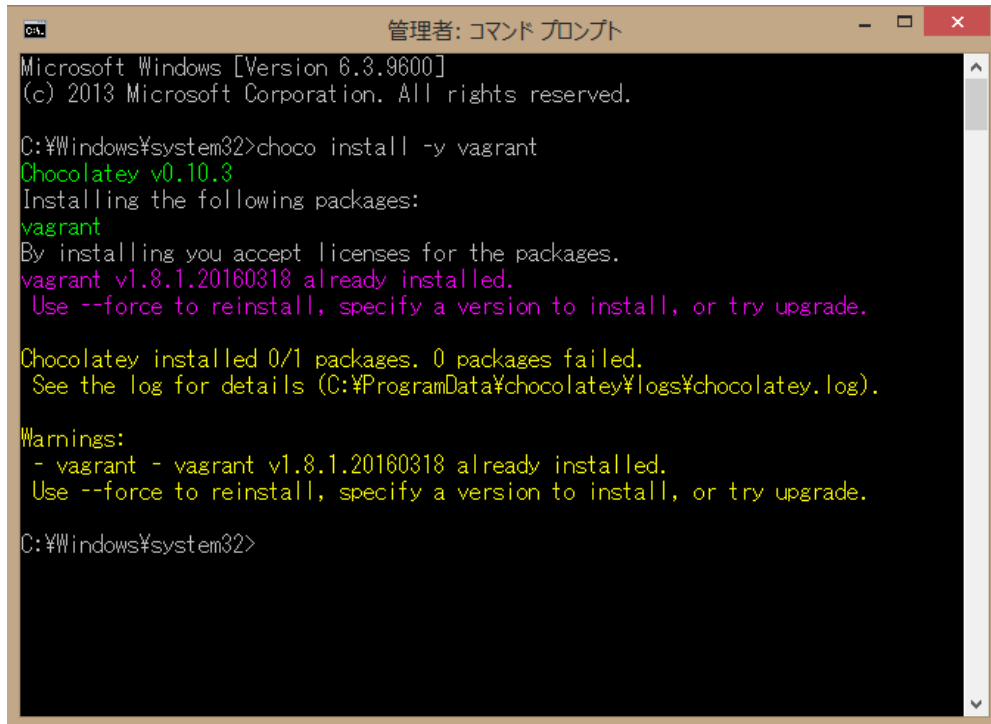
5.4.2 Vagrant のインストール

本研究では Chocolatey を使用してインストールする。コマンドプロンプトを管理者で起動する。



図 5.8 Vagrant のインストール 1

choco install -y vagrant と入力し実行する



```
管理者: コマンド プロンプト
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>choco install -y vagrant
Chocolatey v0.10.3
Installing the following packages:
vagrant
By installing you accept licenses for the packages.
vagrant v1.8.1.20160318 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages. 0 packages failed.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- vagrant - vagrant v1.8.1.20160318 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

C:\Windows\system32>
```

図 5.9 Vagrant のインストール

このようになればインストール完了

5.5 R 言語について

5.5.1 R 言語とは

R 言語とは、統計解析やその結果をグラフィカルに表示するためのシステム「R」用の言語のことである。R 言語は、AT&T ベル研究所の研究者によって設計された統計処理言語である S 言語を元に設計されている。同じく AT&T ベル研究所のが開発した「S 言語」の実装系は商用版が知られているが、R 言語は GNU プロジェクトによってオープンソースで提供されており、無償で利用できる。R 言語は見かけは C 言語に似ているが、簡単なコマンドによりいろいろな機能が実現できる。標準では用意されていない機能も比較的容易に拡張できるメリットがある。

5.5.2 R のインストール

R は Windows、Mac、Unix のすべてに対応している．本研究では Windows 版を使用する．RjpWiki[?] にアクセスし R をインストールしていく．

R のインストール

<http://www.okada.jp/RWiki/?R%20%E3%81%AE%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB>

[トップ | Tips紹介 | 初級Q&A | R掲示板 | 日本語化掲示板 | リンク集]
[編集 | 凍結 | 差分 | バックアップ | 添付 | リロード] [新規 | 一覧 | 単語検索 | 最終更新 | ヘルプ]

R のインストール

Windows, Mac, Linux(redhat, yellowdog, vine,...), Unix のインストールのコツなどを付け足してください。

- Windows 版 R のインストール
 - Windows Vista での問題点
 - 参考リンク
- Mac 版 R のインストール
 - Mac OSX
 - 参考リンク
 - Fink 版 R のインストール
 - Mac Ports 版 R のインストール
 - バイナリの提供されていないパッケージ
- Linux 版 R のインストール
 - Debian GNU/Linux (含む Knoppix)の場合 (インターネット経由でオリジナル版をインストールする場合)
 - openSUSE10.2 Linux 版インストール
 - Ubuntu Linux の場合:
- Unix 版 R のインストール
 - ソースからのコンパイル法 (含む Linux)
- REvolutionR(IntelMKL版R:マルチコア対応:Windows,MacOSX)
- Linuxでソースからのビルド(VineLinuxを例として)
- 興味のある分野のパッケージを丸ごとインストールする

Windows 版 R のインストール

最新版は [こちら](#) からダウンロードしてください。

Windows Vista での問題点

Vistaにインストールして動かないと嘆いている方はここを読んでください。 http://cran.r-project.org/bin/windows/rw-FAQ.html#Does-R-run-under-Windows-Vista_003f

Vista (に限らず2000,XPでも同種の障害が生じ得ます) では、昔ながらのパス、例えば

C:\Documents and Settings\%USER%\My Documents

以下に本体をインストールしたり、ユーザのライブラリパスを設定したりすると、まとも動きません(たとえ英数字だけのパスであっても)。素直に、Vistaの新しい既定のパスを使いましょう。

図 5.10 R インストール手順 1

Windows 版 R の最新版リンクをクリックする．

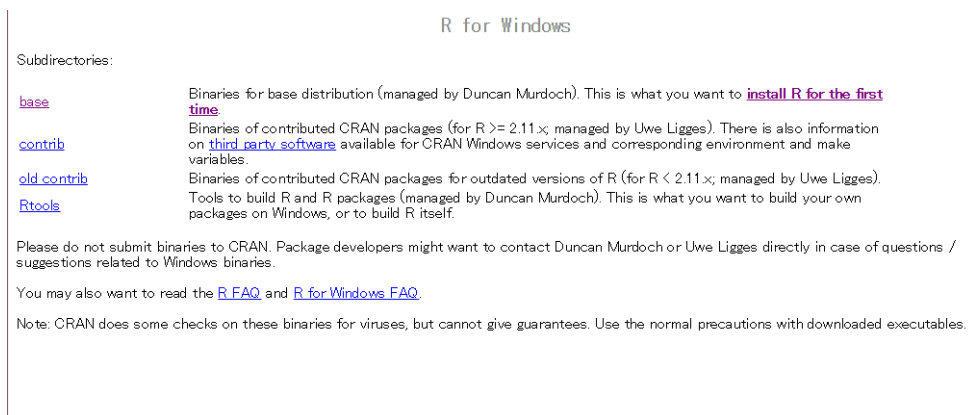


図 5.11 R インストール手順 2

install R for the first time. をクリックする .

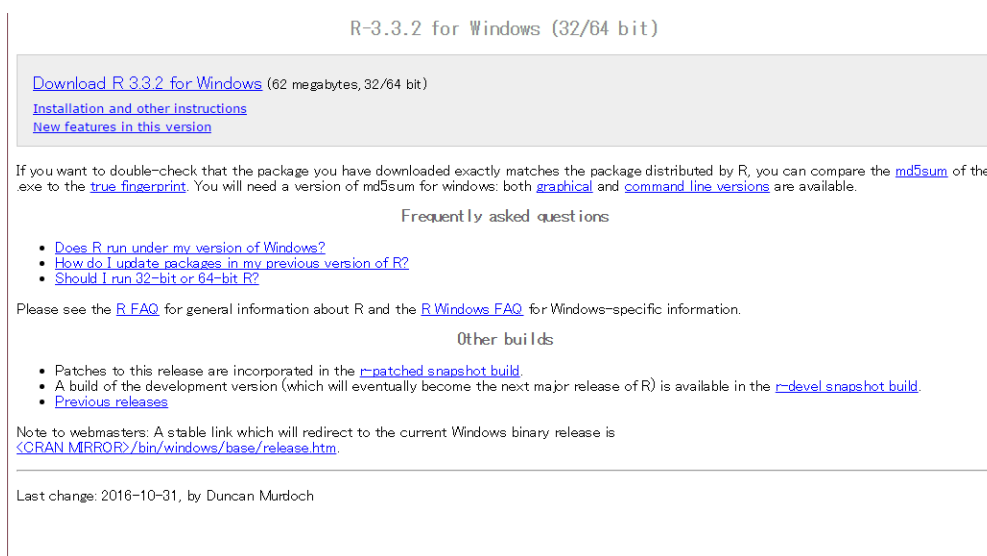


図 5.12 R インストール手順 3

Download R 3.3.2 for Windows をクリックするとダウンロードが始まる .

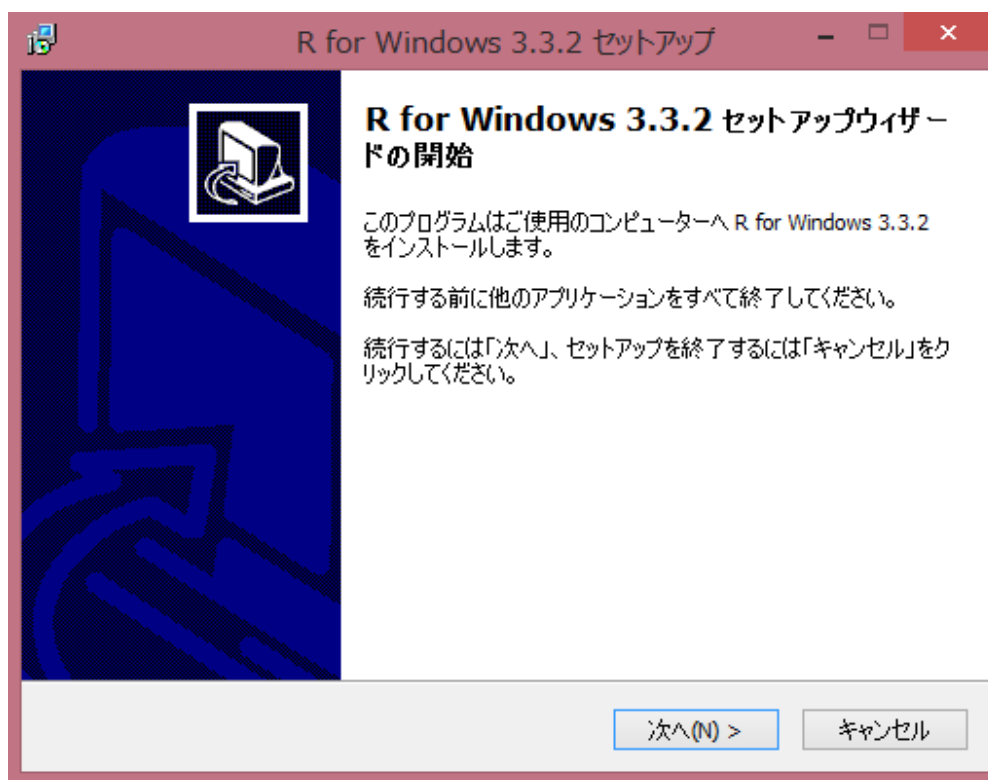


図 5.13 R インストール手順 4

ダウンロードした exe を起動し、セットアップをすることでインストール完了する。

5.6 クローラーについて

5.6.1 クローラーとは

クローラー (Crawler) とは、ウェブ上の文書や画像などを周期的に取得し、自動的にデータベース化するプログラムである。「ボット (Bot)」、「スパイダー」、「ロボット」などとも呼ばれる。主に検索エンジンのデータベース、インデックス作成に用いられているほか、統計調査などの目的にも利用される。近年では電子メールアドレス収集業者などもクローラを利用して、スパムの送信効率を上げている。一般にクローラは、既知の HTML 文書の新しいコピーを要求し、文書中に含まれるリンクをたどり別の文書を収集するという動作を繰り返す。新しい文書を見つけた場合はデータベースに登録する。また、既知のファイルが存在しないことを検出した場合はデータベースから削除する [6]。クローラーとして最も有名なものは、Google などの検索エンジンがあげられる。本研究ではフリーソフトである wget を利用しクローラーを作成する。

5.6.2 wget とは

HTTP/HTTPS と FTP で利用できるファイル取得用コマンド「wget」は、その多機能さと移植性の高さにより、Linux を始めとする多くの UNIX 系 OS で利用されている。また、Windows OS 環境向けには「Wget for Windows」が配布されている。wget は、ソースコードやバイナリのダウンロードだけでなく、Web サイト全体あるいは特定の階層を一括取得できるコマンドである。また、何らかの理由で中断されたダウンロードを中段したところから継続できるなど、いわゆる「ダウンローダー」としての機能を持つ。ブロードバンド普及以前に登場したツールであることから、帯域幅を使い果たさないように速度の上限を定めてダウンロードできたり、接続を試す (リトライ) 回数の最大値を指定できたり、あるいは、ダウンロードが中断された位置から再開する機能 (レジューム) を備えていたりするなど、動作面での信頼性に乏しいネットワークでも確実に実行する機能が用意されている。

5.6.3 wget コマンド

以下に wget コマンドを記す [7] .

表 5.1 スタートアップ

コマンド名	解説
-V	wget のバージョンを表示して終了
-h	オプション一覧のヘルプ
-b	スタート後にバックグラウンドに移行する .
-e	「.wgetrc 形式のコマンドを実行する .」

表 5.2 ログと入力ファイル

コマンド名	解説
-o	ログを FILE に出力する
-a	メッセージを FILE に追記する
-d	デバック情報を表示する .
-q	何も表示しない
-v	冗長な出力をする (デフォルト)
-nv	冗長ではなくする
--report-speed=TYPE	帯域幅を TYPE で出力します .
-i	FILE の中に指定された URL をダウンロードする
-F	入力されたファイルを HTML として扱う
-B	HTML で入力されたファイル (-i -F) のリンクを設定した URL の相対 URL として扱う
--configFILE	設定ファイルを指定する

表 5.3 ダウンロード

コマンド名	解説
-t	リトライ回数の上限を設定 (0 は無制限)
-O	接続を拒否されてもリトライする
-nc	FILE に文書を書き込む
-c	部分的にダウンロードしたファイルの続きから書き始める
-N	ローカルにあるファイルよりも新しいファイルだけ取得する
-no-use-server-timestamps	ローカル側のファイルスタンプにサーバーのものを使わない
-S	サーバーの応答を表示する
-T	全てのタイムアウトを SECONDS 秒に設定する
-dns-timeout=SECS	DNS 問い合わせのタイムアウトを SECS 秒に設定する
-connect-timeout=SECS	接続タイムアウトを SECS 秒に設定する
-read-timeout=SECS	読み込みタイムアウトを SECS 秒に設定する
-w	ダウンロード毎に SECONDS 秒待つ
-waitretry	リトライ毎に 1 ~ SECONDS 秒待つ
-no-proxy	プロキシを使わない
-Q	ダウンロードするバイト数の上限を指定する
-bind-address=ADDRESS	ローカルアドレスとして ADDRESS(ホスト名か IP) を使う
-limit-rate=RATE	ダウンロード速度を RATE に制限する
-no-dns-cache	DNS の問い合わせ結果をキャッシュしない
-restrict-file-names=OS	OS が許しているファイル名に制限する
-ignore-case	ファイル名, ディレクトリ名の比較で大文字小文字を無視する
-4	IPv4 だけを使う
-6	IPv6 だけを使う
-prefer-family=FAMILY	指定したファミリ (IPv6,IPv4,none) で最初に接続する
-user=USER	ftp,http のユーザー名を指定する
-password=PASSWORD	ftp,http のパスワードを指定する
-ask-password	パスワードを別途入力する
-no-iri	IRI サポートを使わない
-local-encoding=ENC	指定した ENC を IRI のローカルエンコーディングにする
-remote-encoding=ENC	指定した ENC をデフォルトのリモートエンコーディングにする
-unlink	上書きする前にファイルを削除する

表 5.4 ディレクトリ

コマンド名	解説
-nd	ディレクトリを作らない
-x	ディレクトリを強制的に作る
-nH	ホスト名のディレクトリを作らない
-protocol-directories	プロトコル名のディレクトリを作る
-P	ファイルを PREFIX 以下に保存する
-cut-dirs=NUMBER	リモートディレクトリ名の NUMBER 階層分を無視する

表 5.5 HTTP オプション

コマンド名	解説
-http-user=USER	http ユーザ名として USER を使う
-http-password=PASS	http パスワードとして PASS を使う
-no-cache	サーバがキャッシュしたデータを許可しない
-default-page=NAME	デフォルトのページ名を NAME に変更します .
-E	HTML/CSS 文書は適切な拡張子で保存する
-ignore-length	Content-Length ヘッダを無視する
-header=STRING	送信するヘッダに STRING を追加する
-max-redirect	ページで許可する最大転送回数
-proxy-user=USER	プロキシユーザ名として USER を使う
-proxy-password=PASS	プロキシパスワードとして PASS を使う
-referer=URL	Referer を URL に設定する
-save-headers	HTTP のヘッダをファイルに保存する
-U	User-Agent として Wget/VERSION ではなく AGENT を使う
-no-http-keep-alive	HTTP の keep-alive (持続的接続) 機能を使わない
-no-cookies	クッキーを使わない
-load-cookies=FILE	クッキーを FILE から読みこむ
-save-cookies=FILE	クッキーを FILE に保存する
-keep-session-cookies	セッションだけで用いるクッキーを保持する
-post-data=STRING	POST メソッドを用いて STRING を送信する
-post-file=FILE	POST メソッドを用いて FILE の中味を送信する
-content-disposition	Content-Disposition ヘッダがあればローカルのファイル名として用いる
-auth-no-challenge	サーバからのチャレンジを待たずに、Basic 認証の情報を送信します。

表 5.6 HTTPS (SSL/TLS) オプション

コマンド名	解説
-secure-protocol=PR -no-check-certificate -certificate=FILE -certificate-type=TYPE -private-key=FILE -private-key-type=TYPE -ca-certificate=FILE -ca-directory=DIR -random-file=FILE -egd-file=FILE	セキュアプロトコルを選択する (auto, SSLv2, SSLv3, TLSv1) サーバ証明書を検証しない クライアント証明書として FILE を使う クライアント証明書の種類を TYPE (PEM, DER) に設定する 秘密鍵として FILE を使う 秘密鍵の種類を TYPE (PEM, DER) に設定する CA 証明書として FILE を使う CA のハッシュリストが保持されているディレクトリを指定する SSL PRNG の初期化データに使うファイルを指定する EGD ソケットとして FILE を使う

表 5.7 FTP オプション

コマンド名	解説
-ftp-user=USER -ftp-password=PASS -no-remove-listing -no-glob -no-passive-ftp -retr-symlinks	ftp ユーザとして USER を使う ftp パスワードとして PASS を使う .listing ファイルを削除しない FTP ファイル名のグロブを無効にする passive 転送モードを使わない 再帰取得中に、シンボリックリンクでリンクされた先のファイルを取得する

表 5.8 再起ダウンロード

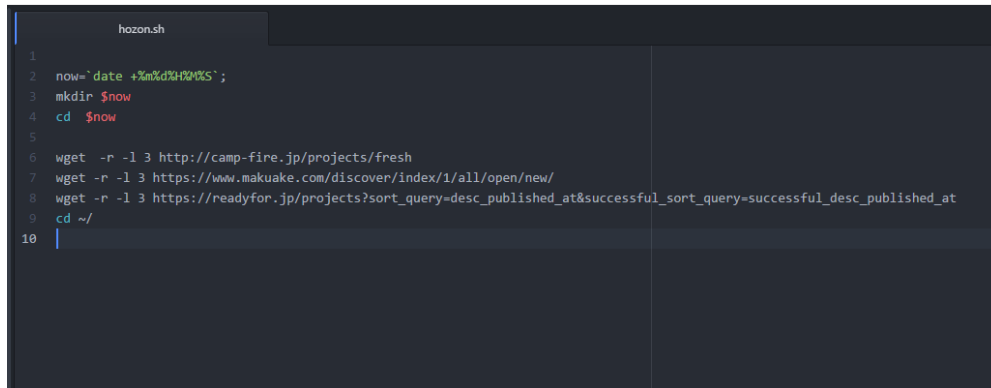
コマンド名	解説
-r	再帰ダウンロードを行う
-l	再帰時の階層の最大の深さを NUMBER に設定する (0 で無制限)
-delete-after	ダウンロード終了後、ダウンロードしたファイルを削除する
-k	HTML や CSS 中のリンクをローカルを指すように変更する
-K	リンク変換前のファイルを .orig として保存する
-m	-N -r -l 0 -no-remove-listing の省略形
-p	HTML を表示するのに必要な全ての画像等も取得する
-strict-comments	HTML 中のコメントの処理を厳密にする

表 5.9 再起ダウンロード時のフィルタ

コマンド名	解説
-A	ダウンロードする拡張子をコンマ区切りで指定する
-R	ダウンロードしない拡張子をコンマ区切りで指定する
-D	ダウンロードするドメインをコンマ区切りで指定する
-exclude-domains=LIST	ダウンロードしないドメインをコンマ区切りで指定する
-follow-ftp=HTML	文書中の FTP リンクも取得対象にする
-follow-tags=LIST	取得対象にするタグ名をコンマ区切りで指定する
-ignore-tags=LIST	取得対象にしないタグ名をコンマ区切りで指定する
-H	再帰中に別のホストもダウンロード対象にする
-L	相対リンクだけ取得対象にする
-I	取得対象にするディレクトリを指定する
-X	取得対象にしないディレクトリを指定する
-np	親ディレクトリを取得対象にしない

5.7 クローラーの運用

導入した Vagrant 環境においてクローラーを動かす．なお使用するプログラムは先行研究で作成された物を使用する [5] ．



```
hozon.sh
1
2 now=`date +%m%d%H%M%S`;
3 mkdir $now
4 cd $now
5
6 wget -r -l 3 http://camp-fire.jp/projects/fresh
7 wget -r -l 3 https://www.makuake.com/discover/index/1/all/open/new/
8 wget -r -l 3 https://readyfor.jp/projects?sort_query=desc_published_at&successful_sort_query=successful_desc_published_at
9 cd ~/
10
```

図 5.14 クローラーの導入 1

毎日，同時刻にデータを収集する際に，同じフォルダで保存し続けるのを避けるため，Linux コマンドの `date`, `cd`, `mkdir` を組み合わせてプログラムが動き出した現在時刻をフォルダ名に設定するようにしてある．`wget` を利用し，各サイトごとに順番にスタート設定した URL から再帰的にファイルを入手するように設定してある．以下に今回使用したコマンドの解説を記す．

表 5.10 [date] 日付，時刻を表示，設定する

文字	解説
%H	時 (00 ~ 23)
%I	時 (01 ~ 12)
%k	時 (0 ~ 23)
%l	時 (1 ~ 12)
%M	分 (00 ~ 59)
%p	AM あるいは PM のロケール (国や地域に合わせた文字列)
%r	12 時間形式の時刻 (HH:mm:ss [AP]M)
%s	1970-01-01 00:00:00 UTC からの秒数
%S	秒 (00 ~ 61)
%T	24 時間形式の時刻 (HH:mm:ss)
%a	ロケールによる省略形の曜日の名前 (Sun ~ Sat)
%A	ロケールによる完全に表記した曜日の名前 (Sunday ~ Saturday)
%b	ロケールによる省略形の月の名前 (Jan ~ Dec)
%B	ロケールによる完全に表記した月の名前 (January ~ December)
%c	ロケールによる日付と時刻 (Sat Nov 04 12:02:33 EST 1989)
%d	日 (月内通算日数) (01 ~ 31)
%D	日付 (MM/DD/YY)
%j	年内通算日数 (001 ~ 366)
%m	月 (01 ~ 12)
%w	週のうちの曜日 (0 ~ 6) で 0 が日曜日に対応
%x	ロケールによる日付の表現 (MM/DD/YY)
%y	西暦の下 2 けた (00 ~ 99)
%Y	年 (1970 ~)

表 5.11 [cd] ディレクトリを移動する

文字	解説
/	ルート・ディレクトリ
.	現在のディレクトリ
...	親ディレクトリ
~	ホーム・ディレクトリ

表 5.12 [mkdir] ディレクトリを作成する

文字	解説
-m	ディレクトリのモードを設定する
-p	指定したディレクトリをサブディレクトリごと作成する .
-v	ディレクトリを作成する毎にメッセージを出力する
-help	mkdir コマンドの使用方法を表示する
-version	バージョン情報を標準出力に表示する
directory	作成するディレクトリ名を指定する

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
00 13 * * * /home/shimada/hozon.sh
```

[^]G ヘルプ [^]C Write Out [^]W 移動 [^]K 切り取り [^]J 両端揃え [^]O 位置
[^]X 終了 [^]R 読み込み [^]Y 置換 [^]U Uncut Text [^]T スペル確認 [^]_ 行を指定

図 5.15 クローラーの導入 2

ホーム上にあるシェルを毎日、同じ時間に自動で動かすために、端末から crontab を利用して時間が来たらシェルを呼び出し、プログラムをスタートするように設定する。端末を起動し、crontab -u ユーザー名 -e と入力をして実行する。実行すると上記の画面が表示される。

エディターでプログラムを行う時刻とプログラムを指定する。文法は以下のとおりである。

分 時 日 月 曜日 コマンド

表 5.13 記述方法

文法	解説
分	分を「0～59」で指定する。ワイルドカード(*)を記述すると毎分となる。
時	時間を「0～23」で指定する。ワイルドカード(*)を記述すると毎時となる。
日	日を「1～31」で指定する。ワイルドカード(*)を記述すると毎日となる。
月	月を「1～12」もしくは「jan～dec」で指定する。ワイルドカード(*)を記述すると毎月となる。
曜日	曜日を「0～7」(0,7 は日曜日)もしくは「sun～sat」で指定する。ワイルドカード(*)を記述する。
コマンド	実行したいコマンドやシェルを記述します。

今回のクローラーの稼働設定を例としてあげると、

```
00 13 * * * /home/shimada/hozon.sh
```

毎日 13:00 に/home/shimada にある hozon.sh を実行するという命令になる。

0722130001	0817130001	0914130001	1010130001	1105130001	1130130002
0723130001	0818130001	0915130001	1011130001	1106130001	1201130001
0724130001	0819130001	0916130001	1012130001	1107130001	1202130001
0726130001	0823130001	0917130001	1013130001	1108130001	1203130001
0727130001	0824130001	0918130001	1014130001	1109130001	1204130001
0728130001	0825130001	0919130001	1015130001	1110130001	1205130001
0729130001	0826130001	0920130001	1016130001	1111130001	1206130001
0730130001	0827130001	0921130001	1017130001	1112130001	1207130001
0731130001	0828130001	0922130001	1018130001	1113130001	1208130001
0801130001	0829130001	0923130001	1019130001	1114130001	1209130001
0802130001	0830130001	0924130001	1020130001	1115130001	1210130001
0803130001	0831130001	0925130001	1022130001	1116130001	1211130001
0804130001	0901130001	0926130001	1023130001	1117130001	1212130001
0805130001	0902130001	0927130001	1024130001	1118130001	1213130001
0806130001	0903130001	0928130001	1025130001	1119130001	1214130001
0807130001	0904130001	0929130002	1026130001	1120130001	1215130001
0808130001	0905130001	0930130001	1027130001	1121130001	1216130001
0809130001	0906130001	1001130001	1028130001	1122130001	1217130001
0810130001	0907130001	1002130001	1029130001	1123130001	1218130001
0811130001	0908130002	1003130001	1030130001	1124130001	1219130001
0812130001	0909130001	1004130001	1031130001	1125130001	1220130001
0813130001	0910130001	1006130001	1101130001	1126130001	1221130001
0814130001	0911130001	1007130001	1102130001	1127130001	1222130001
0815130001	0912130001	1008130001	1103130001	1128130001	1223130001
0816130001	0913130001	1009130001	1104130001	1129130001	1224130001

図 5.16 データが格納されているディレクトリ 1

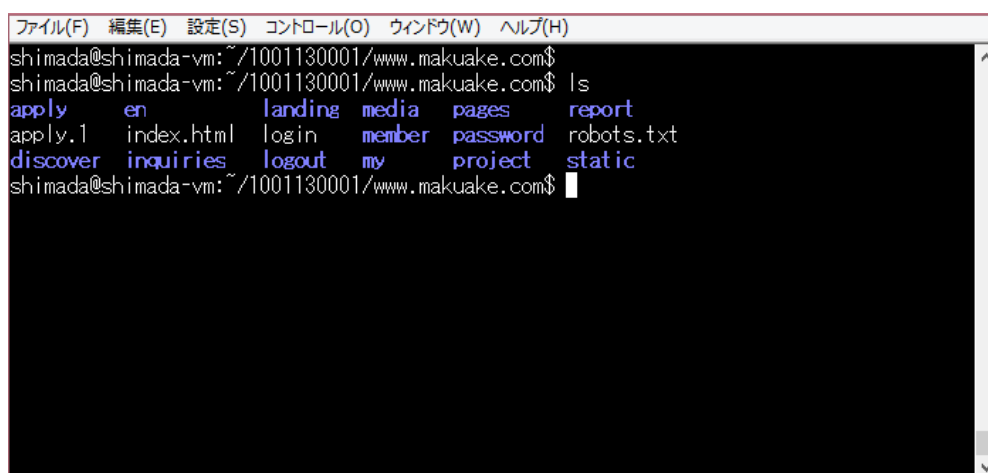
実際にデータを集めると上記のようにディレクトリが自動生成される。ディレクトリ名の 0722130001 は、7 月 2 2 日 1 3 時 0 0 分 0 1 秒といったようにプログラムを実行した月日時分秒となっている。

以下にデータが格納されているディレクトリの構造を記す。



```
10.100.192.11:22 - shimada@shimada-vm: ~/1001130001 VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
shimada@shimada-vm:~/1001130001$
shimada@shimada-vm:~/1001130001$ ls
readyfor.jp  www.makuake.com
shimada@shimada-vm:~/1001130001$
```

図 5.17 データが格納されているディレクトリ 2



```
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
shimada@shimada-vm:~/1001130001/www.makuake.com$
shimada@shimada-vm:~/1001130001/www.makuake.com$ ls
apply      en          landing    media      pages      report
apply.1    index.html login      member     password   robots.txt
discover   inquiries  logout    my         project    static
shimada@shimada-vm:~/1001130001/www.makuake.com$
```

図 5.18 project 内のディレクトリ 1

100cheer	coffeetonya	honobononanga	nanami-oda	osakajou	stanwars-ukiyo
18steakdiner	colorbath	horror-novel	mandara-shirts	osakamanga	steakrevolution
22therojest	columbia	hoshinomusic	marhattan86	osashion	steridoe-plus
25vinetcina	comicalette	hoshizora2015	marcailuku	oshimaaya	stilljourney
29on	community-reactivation	houfukuei	maple	oshushusen	stirsysten
300cmeet	conor500	human	mappy	osin	storksak
34nazu	corinthians	hvakuninissu	maremonte	otonari	story-block
3drinter-maestro	cornerstonefastine	hyperchiller	maris	otonorhon	studforstartup
3eepleexhibition	coronetlabel	ibaiyo02	marwasafil	otouonogasaki	suits-jewelry
43pyo	colohkyoto	ibaragani	masaki-nakamura	otoshidama-project	sumahobokuro
4kakatana-project	coupe	ichi-robe	masatoworld-dream	ottotto	sumi-carepans
50ehon	courage-create	icon-fleurage-jewelry	matakana	omipurin	sumidaiive
53chairs	crazybeauty	ideaip	material-link	overfence	summerbomb
72recipes	creative-wf	idolundokai	maximurai	oxalis2014	sun-field-co
7nabits	crosslinks	igasso	mayoiza	ovatsu-sapuri	sunazaki
7eat-sugoro	crossroad-furniture	fine-innov	mayotoichi	pacan	sunward
889	cuisson	ioffice-osbu	mba-shibuya	paddleboard	superb
Canik	cumu	ikanoosushi	mcdsp001	pancakes	surinacca
StarFangedNut	cutnrec	iliannloeb	nd-amadana	paraparamanga	surprisebox
TactSlider	cvaio	imagetrainingenglish	neatkeep	paridaka	survivalgame
am	dazashiyarock	imaginehair	nece01	pattorikun	sushi-susa
abseterno	dazdnt	inakamori2	medeteinc	pedhat	swagzan
acoma	daiki-saitama	inamunabue	meafes	pedalrest	sweets-agent
across-the-border	daikouryu	india	meinoku	pegarball	symbol
act-japan	dandadanborumini	indigoninias	memotore	pequenitouruguay	symlis
addspice	deafgolf	indigoninias.1	men-nari	peyon0530	synchony
afinibox	decadentminimalist	indigoninias2	mendika	picoca	synr-tv
afrogroove	declan	infiorata-bachilio	mercaer	piccake	sywomamagazine
agata-japan	decorebonne	inochikan-festival	michiyohara	piju	tabianabi
ahess	deeplearning	inspiringoffice	micialuxury	pinostovhashi	taocho
ai-love	denacetaro	inspyre	micrafan	pinincock	tagiri-hotel
aichi-geiope	dendama	intexs	mid-okayama	plants-jewel	tagplan
ailedange	denea	inunome	mighty-lady	pleasurekaren	taishiishida
aincompany	denshiblock	irepairslab	mighty08	ply-note-2017	taivachanee
ainozumi	desk-ds	irukateabag	milk-las	plynote	takatsukistar-burger
airfresh-r	digitalkiss	isamuisa	millionizumou	pokito-mouse	takebow-tune
airship-e	digitalnaturalpark	ishinomaki-tarako	millionshells	pokeco	takizawa-winery
aisakaminori	discovery-japan	islandjazz	minamori	polylotsapp	takusei
aisome	dihasebe	italia-independent-japan	minamori-camera	poncareer	talk-talk-tokyo
aiszuid	dici-japandi	itaspo	miniben	poneri	tama
ai-project	dici-japandi2	live-tp	minicfe	poopscouture	tanazawa-camp
akbhakki	doe-roov	iwakithamoniorkest	minikou02	powername02	tanon
alibirebb-rabbits	dolowefun	iwakishi-sportsevent	minamizakuseisai	preciousdays	tanabata
alibun-ehon	donatton-kumamoto	iworl013	minamikaze	premiergrand	tanakatakuma
alice-eyeshadow	doraneco	izat	minedrip-coffee	pressy	tanajuchitaisei
alice-tea	doublel	izur-pethome	mineral-no32	probono	tanacinema
alifhome	dramaticdance	japan-zekkei	minokiruto	project788	tantan-tiger
allnations	dreamolonet	japanese-tradition	minoku	projectdott	tanwabata
allnations2	dreamsumit	japanwood	miraiiku	projecthikari	tap-room-kyoto
alivours-oneswing	dreamsupport	jellylyllycafe	miro-clearpot	prosciutto	tateyamaschool
aloe-vera	dressstock	jerry-mart	mirroredesign	psalsonayuumi	tazigen
aloha-mai-ka-hula	dromeyazaki	jewelsavior	mirroris	puzzle-project	tcwz016
amadanamusic	drsrack	jienotsu	mirusru-campus	qoca	teamfuchu

図 5.19 project 内のディレクトリ 2

aloha-mai-ka-hula	dronemiyazaki	jewelsavior	mirroris	puzzle-project	lcrew2016
amadnamusic	dsrnoak	jienotsu	minusru-campus	quoca	teamfudo
amabouni	duong-dagon	jienrikisha	misshinkobou	qrio-smart-lock	tedno-accessory-module
amekaze-sakai	e-learn-life	jinsaikikouen	miss-sato	qrio-smart-tag	technologic-laser
and-flower-romantica	e-solo-e	jk-campus	missberry	quadcross	telicare
andhostel	earthembassy	joshujidori	missmisterdeafjapan	qubena	tempgraphie
anicornmatches	eastanipnet	journalismproject	missake	quilts-studio-nv	tennennura
anitehandson	ecodack	joyunconte3	mitsuhayunosirw	quinteto-lorenzo-fernandes	tennis385
anniversary-for-mam	edame-corocoro	jipo	mitsukidokuva	rabbitproject	terashou
announcer-lesson	editor	jthree	mitsuya	racersnavi	teracova2016
ankoronohada	edweek	junkstory	mituuiPHONEcase	raikuraku	tethercell
antsoffice	edutyip	jvc-mim	miuraguesthouse	rakuraku-apron	tetsuichi
apbaart	esetouch	jvukimune	miyabio	rapiro	tezukurigirl
apopita	eigaarchive	ka-zufitness	miyakojima-apapin	rayrole	the-smatch
asalay	ei-julce	kaetsode	miyakojimamusic	re-fine	theateroforest
arakawayuji	elchaddai	kaiteijiyukusei	miyakojimamusifestival	rebirthink	theateroferror
areakyoto	english-cardgame	kakehashisports	miyakojimamusifestival02	reboot	theofficeshop
arianero	englishree	kamanisehou2	miyakojimamusifestival03	recordasia	thefilament
arita-shareglass	enma-do	kami-ya	mizutori-geta	redandblue	thegraces
arrounders	enra	kami-ori	moblis	relapoi	thejimmcase
artcase	eremote-mini	kanoichi-sun	mocowatch	remember-me	thomusic
artcase02	erical18music	karon	modelfor smiles	ren-ou	thus
artcase03	erimaki	kaniwataberu	moebae	renacnatta	thread-tokyo
artifactadventure	evangelionracing	kannokyo	molfo	renatorinouta	thunderbirds
artsatkit	evangelionracing02	kanumasoba	monoiro	repidputera	timeslip-jk
arukutomo	exhibition-network	kaoricomedy	monfes	reraku	tinga2
ashizawaaplus02	fakeshow	karasawa	montdor	ric	tisay
asobindolaresproject	familytable	karinshikata	mongomaleo	riesamiya	tkysnakecenter
asuka-project	fashionese	kasuri-rito-project	nomoto-iyaku	rikku	tkc
atakaplanning	fchile	katatakatiken	norino-dji	rikku	tohobizin
atarakutokushima	fcoosaka	katteni-singeki	morinosyoshitsu	ririnika	tokachinovie
ateji	feswatchreturns	kawaihiroo	morison-cafe	risoukaikaisya	tokinokizuna
atelier-sunnyday	fetjglobal	kawatako	mote-po	mdlo	tokiwa_biotop
atelier-svoochoo	findi	kazsoro	mother	modish	tokyodesignweek
athoth-window	flex-camp-pic	kazakirikiriya	montek-japan	moathorse	tokyogather
aton	flipnote2015	ke-i	moutakusanda	roasthorse02	tokyochanger
autikuoka	flowersake	kehara	mucoo	robo-done	tokyowitchhat
autisticchild	fluteau	keigetsu	mujinaiiku	robot-nozomi	tomoru
avocado	flxshs5	ken-chara	mushi-bistro	rock	toshinbow
awatakarabako	foldto2	kenyamuralproject2	musicdinner	roso	touchence
awavuri	foldto380	keyquest	musibino	rollerderby	touchenotorigoya
azulien	foe	keysentaldress	mvi-animikan0629	rolling	tozakihiroki
babymist	foodjewelry	kiddboystyle	mvi-acharu2015	rosewater	trail
badhus-kofu	forestcanyon	kidsad	mvi-dcs-infinity	rotenburo	trainersnetworkosaka
badpaix	formeishop	kidsline	mvi-gltokyo	rotiea-nail	tres-tokyo
bambitious	fonubar	kie-pinoko	mvi-happy8anniversary	rovers	truckyosakoi
banksollection01	forugranola	kiff	mvi-irobo	rseschool	truckyosakoi02
barbershop	framboise	kifukuen	mvi-izuru	rseschool02	tnmax
bartender	frankaiura-chiba	kikacool	mvi-kaizoku	rubbergirl	tnsky
basemakers	frankaiura-gifu	kikakugaiyasai	mvi-low-hipps	ruff-cycles	tsukinosan
basepointoffice	frankaiura-gifu	kikusui-sake	mvi-miraiz	s-kin	tsukurukai-ekishi
bassers	frankaiura-gunma	kinono-atelier-akane	mvi-need	salie	tsushimakonne
beasbee	frankaiura-kagoshima	kiurasant	mvi-nomirz	sasafennel	tsuyagirl

図 5.20 project 内のディレクトリ 3

bassers	frankiura-gunna	kinono-atelier-akane	mvl-need	saife	tsushimakonne
beansbee	frankiura-kazushima	kinurasant-i	mvl-nomirz	saisufenel	tsuavairi
beaupo-toyo	frankiura-kanazawa	kindai	mvl-pawamusu	saiharatrain	tuneinc
befestival	frankiura-nagano	kinokaiapan	mvl-pinksnoflower	saiindustries	tvidol
bellasgia	frankiura-saitama	kinouryu	mvl-stg	sakederby	twilight
belt-attach	frankiura-shizuoka	kinowa	mvl-yuc523ponz	sakekanki	ufu
bhutanatsutake	frankiura-tokyo	kirie	mokinso	sakelabotokyo	ujien
biliblabotherany	frankiura-toyama	kifakami-shinentai	mve	sakura-japan	ukyo
biufriend2015	frees-nail	knot	mvtapes	sakurasakachocolat	uino-namamau
bioproject	front-reo-eyeshadow	knottlaashoo	mviway	salad	undergroundreport
biroku	fruitbouquet	knowers	n-sound	salon-omotenashi	unreverse
bishonen	fruitliqueur	koba	nadeshiko	salts-shampoo	unsodo-hokusai
bitsukeshi	frusion	kobe-runway-club	nadaoqui	sameteabag	upla
bitszaidan	fujitokanado	kobekids2016	nazarawadepart	samurai-travel	uracoco
bitszaidan2	fujisan-haburashi	kobesander	nazareboshi-goods	sauraiseven	urakamitenshudo
bitszaidan3	fukushinail	kobi-one	nazareboshi-movie	saradavikimura-lady'sjacket	urakizusawa
biwakoi23	fundoshisamurai	kodomonobasho	nazareyama34	sangokushi	urbanstylebb
blincaminc	funflying	kodomotobutai	nagasaki-hiroshima-archive	sankin-kotai	urushitumbler
blindbarber	funkianbientglass	kodurebe	nagasaki-popcorn02	sankoun	ushikura
blueberry-farm	furaiito	koharudani-i	nagasakids	sanshinkosame	utsukokazura
bar	fuetsu	koisan	nazatoku	sankacrossroad	uzirrls
collina	galleriashineibashi	kojimagnes02	nazovase	saruya	uwater
bondic	gallensodds	kokonoka	nazovasc2	satomi	vack-flip
bonnel-cafe	gamebar-nostalgia	koneji-rushi	nail-azurite	savannahonline	vampire
bow-style	gameikkyo	konekolab	naiseal	sazendo	vanhanks-valour
boy-baseball	gameaster	konoskai	nakanomusichow	screenheater	vedavie
boysboys	gameateb	kosuimokotoba	namonazo	seoul	veldt
branch	gamepark	kouji	nasakura	seikan-renrakusen	vert
brave	garage-days	kouken	nastent	seitousya	vertebrebrewery
bravo-yokosuka	garagesumida	kounotori	nature-remo	sekayume	vie-shair
brewpub	gearpus	kowanry	natureactivation	seksuiji	villapawa
briz-stool	genecre	koBuri	nautikus	self-nailschool	vintom
budou-jo	genetsu-sensei	kurasezakka	navioto	setociela	vivarosejapan
budoutou	gifeife-littlebird	kuratsukimarket03	nestbe	setosawa	m-lotte
bulldhome	gibasachan	kuratsukise	nekonimikan	setau	wa-chic
bunakukinevo	gifushirtproject	kuriproi	nekotama	sevendreamers	wafukunomori
business-card-game	gijinka	kusekiappealshitai	nekoyon-movie	shako-dance	wakakusa-honpo
bustcare	gikyoku	kutsunigaki	neee	shanti-le	wakaze
bustup-revolution	ginner	kyodathina	neomary	share	wakura
butterflyboard	glamore	kyofuno-onchikaeri	next-innovation	sharevillagesproject	wamondito
cabin	glassfilm	kyotodeni	nextdrive	sharevillagesproject2	warakuza
cacaken	global-itnet	kyotouniy-lax	nichinancity	shibuya-song-support	watchsuit
cafe-kitchen333	global-itnet2	kyoutoutafess	nihonshugenkasakaaura	shiotodaisuki	wave
cafe-transmitter	globalathlete	kyouhouen	nijiro-gakkou	shikokusweets88	waza
cakesale	globalabo	kyudobukaikam	nijirokomachi	shinsibashi	weddee-times
camphor	globlue	l3d-cube	nikkocamp	shirahamaapartment	welovedogs
camindia	zodj-plus	la-comfortable	nindishou	shiratsuchiing	whitetre
campingcar	go-tokevo2020	lafabrics	niwano	shisuiinovosake	whyte
caminoaudio	godtac	lagendshizafc	no-glu	shodo-shunga	wififshare
canoe	gozakidou	laughingpokerface	no-makes	shoocare	winecanada
cartoliveclubsport	zohan	laucher	no-makes2	shohari26	winecanada-photo
case-remote-air	zokuhintonie	lavierine	no528hz	shonan-3-bros	wiredusicfestival
cashmere-nepal	zokurikuma	layeredfrAGRANCE	nobara	showa-seizaku	woodberrycoffee

図 5.21 project 内のディレクトリ 4

cassara	golden	leafce-mag	noborock	shuchunikki	woodenlook
catedhibition	gooddiberoject	leafap	noocjapan	shuzan02	woodstyleinc
cawaiijapan	gooduice	leafap2	nodoka-fam	shukatsuusical	worldp
caylav02	gpd-win	leafaux	nomotokikinzoku	shuriken	world-visual-festival
comha-musical	greenpiece-yame	li-bero	non-amo	sifaka-ne	worldfiliarana
ceacloth	auoitaily	life-link	noren-kai	sijinyumei	worldintwelve
cebu	guzentheidea	life-shampoo	novast	sinchanser	x-garden
ceramicjapan	guradoll-cafe	lifeisstress	novun	singalongrade	xcdid
cessi-carat	gussu-refle	lifemoments	npo-asia.org	singenieu	xrosscaca
cf-seikou	h2plus	lifependant	nponi-arumi	sintaissou	yabuya
chachachadango	hagumi-hugme	linogconcerts	nposoraside	sion	yakifes
challenerav	haiiro-de-rossi	littledocakes	nsnude	siphon	yakiniku-mitsuboshi
chanzy	hair-donation	littlekyota	nu-rain	sitpack	yakinotuban
charasenoma	halkaku02	locan	null	stackline	yamadafani
charibocslay	hakuto	locorganic	numbers	sleepin	yamanohi
cheer-phoenix	hamburgirlz	locotaining	nurugo-micro	smart-beach-project	yamanokuchibue
chichinouta	hanabidama	love-pop-card	nuilee	smart-mask	yamato-takeru
chierlandcherry	hanalab	lovelymovipiano	nyannyanji	smartdock-bridge	yanbaru-mango
chihirorenamo	handvseco	lovenifune	o-saifu	smartdrive	yasadressine
chikahiroanaka	hapikira	loversengish	obasute	smartene	yashinoki
chikachikaaido	haputdme	lowesthokkaido	oceanifitness	smasta	yayoi-yamaji-museum
chikatetsu	haruoc	lucyst2016	offtheball	smileledge	yayoinoniji
chikyuya	hasegawayuta	lucy-trick	ofurodaisuki	snassy	voie-de-vivre
chinzeya	hatarakumama	luke-hol-loway	ogurafam	sneakerages	yokohamafc
chocolabb-richceramide	havespi	luminoodle	ohnamii-4ninseki	soar	yokohamass
chocolate-in-a-bottle	hazama	luminoodle.1	ohiichi	softball-net-tv	yokotabesstudio
chocolatelaboratory	headlightcleaning	luminoodle2	ohi-doravaki	sozojapan	yoru
chocolatier	heat-up	lurefarm	okemo	solampaper	yosakoi-saiya
chokkura	hello-books	lynx	okeeffefurniture	solothheater	yoshidahiraki
choccream-bar	hi-story-cafe	m4-earphone	old-fabric	sonidecancer	yoshimi
cinema-fresh	hidayoshigundan	mabee	omutsunashi-kyoto	sora	yoshio
cinematograph	hikari-fureru	machi-ikufujikawa	onaka	soumen	yoyakuagent-system
cinnamon	hikarimasu	machinokoshi	onazoro	soybeans	yucat
circusoutdoor	hikaniooe	machinowiru	onizirazu	spalabo	yuito
citron	hinanoki	mago-channel	onitiri-stand-gyu	sparkle2	yukikuroki
classic-concert-himitsu	hinkarinka	mauro-oma02	onlyone	speakbuddy	yukimokura
clips	hinodefactory	mahoan	onnarampo	sportsvibration	yumeani-nubouenkyo
clubnegative	hipatv	mahoutshinoinrou	ontheroadco	soare	yummedeliwldeli
cme	hiragana-02	majorleague-keito	ookanikareshi	sridaiva-yoea	zaaz
cocodit	hirawaki	majorleague-challenge	orengefilmes	sufukuoka	ze-hotel-japan
cocounjapan	hirayunosen	makohair-plus	orchestra-ensemble	stand-by-fam	zetsumetsuzoo
cocounschool	hirosihina-nagasaki-peace	makesakeproject	oregonproject	standkaban	zetsumetsuzoo2
cocoran	hitomebore	makoikeda	organizer	standup-pencase	zounochie
coffeemafia	hmsopo	manasalongrace	orientella	startup_in_israel	zubitsjapan
shinada@shinada-vpn:~/1001130001/www.makuake.com/project\$					

図 5.22 project 内のディレクトリ 5

これらのディレクトリ名を <https://www.makuake.com/project/> のあとに足して web 上で検索することでインターネットに掲載されている状態のプロジェクトを見ることができる。各ディレクトリ内にある index.html がその日取得したプロジェクトのデータである。

5.8 調査方法

クローラーを使用し、取得できたデータを処理する。今回取得したデータは 2016 年 7 月 18 日から 2017 年 1 月 6 日までに取得したデータである。

5.8.1 データの整理

1 日ごとのデータを 1 つのファイルにまとめていく。まず日付の一覧が記載されたファイルを作成するため、集めたデータのある場所で以下のコマンドを実行する。

```
ls > ../date.dat
```

次にプロジェクト名の一覧を作成する。以下のコマンドで実行する。

```
for i in $(cat date.dat);do  
ls $i/www.makuake.com/project/ >> pmid.dat  
done
```

```
sort pmid.dat | uniq > project.dat
```

次にプロジェクトごとのディレクトリを作成する。以下のコマンドを実行する。

```
for i in $(cat ../project.dat);do  
mkdir $i  
done
```

次にプロジェクトページの html ファイルからその日の金額だけを抜き出していく。

```
423
424     <i class="icon_funded"></i>
425
426
427
428     <div class="pnumBaseUp">
429
430
431
432         <div class="stMoney">
433             <p class="stMoneyTtl"><i class="stMoneyTtlIcon"></i><span>集まっている金額</span></p>
434             <p class="stMoneyNum">581,000円</p>
435             <p class="stMoneyGoal">目標金額 2,000,000円</p>
436         </div>
437
438
439
440         <div class="stbarDarkGreen"></div>
441
442
443
444
445         <div class="stSupo">
446             <p class="stSupoTtl"><i class="stSupoTtlIcon"></i><span>サポーター</span></p>
447             <p class="stSupoNum">63人</p>
448         </div>
449
```

図 5.23 取得したプロジェクトページの html ファイル

上図の 581000 円の部分である。

作業ディレクトリで以下のコマンドを実行する。

```
for d in `cat project.dat`;do
for i in `cat date.dat` ;do
cp ../$i/www.makuake.com/project/$d/index.html $d/$i.html
grep "stMoneyNum" $d/$i.html > $d/$i
gawk -f ed.awk $d/$i > $d/$i.csv
gawk -f ed2.awk $d/$i.csv > $d/$i あ.csv
done
done
```

実行すると各プロジェクトのディレクトリが以下の画像のようになる。

1115130001.html	1128130001.csv	1211130001	1223130001あ.csv
1115130001あ.csv	1128130001.html	1211130001.csv	1224130001
1116130001	1128130001あ.csv	1211130001.html	1224130001.csv
1116130001.csv	1129130001	1211130001あ.csv	1224130001.html
1116130001.html	1129130001.csv	1212130001	1224130001あ.csv
1116130001あ.csv	1129130001.html	1212130001.csv	1225130001
1117130001	1129130001あ.csv	1212130001.html	1225130001.csv
1117130001.csv	1130130002	1212130001あ.csv	1225130001.html
1117130001.html	1130130002.csv	1213130001	1225130001あ.csv
1117130001あ.csv	1130130002.html	1213130001.csv	1226130001
1118130001	1130130002あ.csv	1213130001.html	1226130001.csv
1118130001.csv	1201130001	1213130001あ.csv	1226130001.html
1118130001.html	1201130001.csv	1214130001	1226130001あ.csv
1118130001あ.csv	1201130001.html	1214130001.csv	1227130001
1119130001	1201130001あ.csv	1214130001.html	1227130001.csv
1119130001.csv	1202130001	1214130001あ.csv	1227130001.html
1119130001.html	1202130001.csv	1215130001	1227130001あ.csv
1119130001あ.csv	1202130001.html	1215130001.csv	1228130001
1120130001	1202130001あ.csv	1215130001.html	1228130001.csv
1120130001.csv	1203130001	1215130001あ.csv	1228130001.html
1120130001.html	1203130001.csv	1216130001	1228130001あ.csv
1120130001あ.csv	1203130001.html	1216130001.csv	1229130001
1121130001	1203130001あ.csv	1216130001.html	1229130001.csv
1121130001.csv	1204130001	1216130001あ.csv	1229130001.html
1121130001.html	1204130001.csv	1217130001	1229130001あ.csv
1121130001あ.csv	1204130001.html	1217130001.csv	1230130001
1122130001	1204130001あ.csv	1217130001.html	1230130001.csv
1122130001.csv	1205130001	1217130001あ.csv	1230130001.html
1122130001.html	1205130001.csv	1218130001	1230130001あ.csv
1122130001あ.csv	1205130001.html	1218130001.csv	1231130001
1123130001	1205130001あ.csv	1218130001.html	1231130001.csv
1123130001.csv	1206130001	1218130001あ.csv	1231130001.html
1123130001.html	1206130001.csv	1219130001	1231130001あ.csv

図 5.24 プロジェクトファイルの中身

最後に 1 日ごとの金額のデータを一つのファイルにまとめる．作業ディレクトリで以下のコマンドを実行する．

```
for i in `cat project.dat`;do
```

```
cat $i/*あ.csv > $i/$i.csv
```

```
done
```

実行すると プロジェクト名.csv というファイルが作成される．左からプロジェクト名，日付，金額の順でまとめられている．

	A	B	C
8	biblle	1.02E+09	368000
9	biblle	1.02E+09	402000
10	biblle	1.02E+09	429100
11	biblle	1.02E+09	445700
12	biblle	1.03E+09	475700
13	biblle	1.03E+09	499500
14	biblle	1.03E+09	516500
15	biblle	1.03E+09	516500
16	biblle	1.03E+09	523300
17	biblle	1.03E+09	544800
18	biblle	1.03E+09	548200
19	biblle	1.1E+09	572000
20	biblle	1.1E+09	575400
21	biblle	1.1E+09	602600
22	biblle	1.1E+09	616200
23	biblle	1.11E+09	623000
24	biblle	1.11E+09	634300
25	biblle	1.11E+09	637700
26	biblle	1.11E+09	641100
27	biblle	1.11E+09	641100
28	biblle	1.11E+09	641100
29	biblle	1.11E+09	675300
30	biblle	1.11E+09	705900
31	biblle	1.11E+09	736500
32	biblle	1.11E+09	816900
33	biblle	1.12E+09	833900
34	biblle	1.12E+09	885300
35	biblle	1.12E+09	929500
36	biblle	1.12E+09	932900
37	biblle	1.12E+09	957000
38	biblle	1.12E+09	1036000
39	biblle	1.12E+09	1056800
40	biblle	1.12E+09	1090600
41	biblle	1.12E+09	1125000
42	biblle	1.12E+09	1128600
43	biblle	1.13E+09	1139000
44	biblle	1.13E+09	1224100
45	biblle	1.13E+09	1295900
46	biblle	1.13E+09	1323400
47	biblle	1.13E+09	1323400
48	biblle	1.13E+09	1323400
49	biblle	1.2E+09	1323400
50	biblle	1.2E+09	1323400
51	biblle	1.2E+09	1323400

図 5.25 プロジェクト名.csv

5.8.2 プロジェクトの選別

取得したデータから，成功したプロジェクトのみを集める．成功しているかどうかの判断はプロジェクトページの集まっている金額が Success!か Funded と表示されているプロジェクトを集める．



図 5.26 Success!と表示されてるプロジェクト

The image shows a crowdfunding campaign for 'Image Training English' (イメージトレーニングイングリッシュ). The main visual is a woman sitting at a table, writing in a notebook, surrounded by various educational materials like books and CDs. A smartphone on the table displays a cartoon character. The campaign title 'イメージトレーニングイングリッシュ' is prominently displayed. A pink badge in the top left corner says '世界初登場' (World Premiere). A green starburst in the top right corner says 'Funded'. The campaign details on the right indicate that 388,900 yen has been collected out of a 500,000 yen goal, with 36 supporters and 0 days remaining. A large button at the bottom right says '終了しました' (Ended). Below the main image, the text '文字なし！音で学ぶ！画期的な英語学習教材：イメージトレーニングイングリッシュ誕生' (No text! Learn with sound! Revolutionary English learning material: Image Training English is born) is shown. At the bottom, there are social media sharing buttons for education, likes (46), tweets, Google+, bookmarks (1), and Ameba, along with a 'WIDGET' button.

世界初登場

イメージトレーニング イングリッシュ

▶ 再生

集まっている金額 **388,900円**
目標金額 500,000円

サポーター **36人**

残り **0日**

終了しました

文字なし！音で学ぶ！画期的な英語学習教材：イメージトレーニングイングリッシュ誕生

教育 いいね！ 46 ツイート G+ 0 Bookmark 1 Amebaに投稿 <> WIDGET

図 5.27 Funded と表示されてるプロジェクト

5.8.3 ランダムサンプリング

選別したプロジェクトからランダムサンプリングを行い、データの件数を 100 件にする。
以下のコマンドを仕訳したファイルのあるディレクトリで実行する。

```
shuf -n 100 data.txt
```

5.8.4 調達資金の時間変化の可視化

csv ファイルにまとめた調達資金の時間変化を可視化する．例として，機能性スマートバックパック BACKPAIX のプロジェクトを記載する

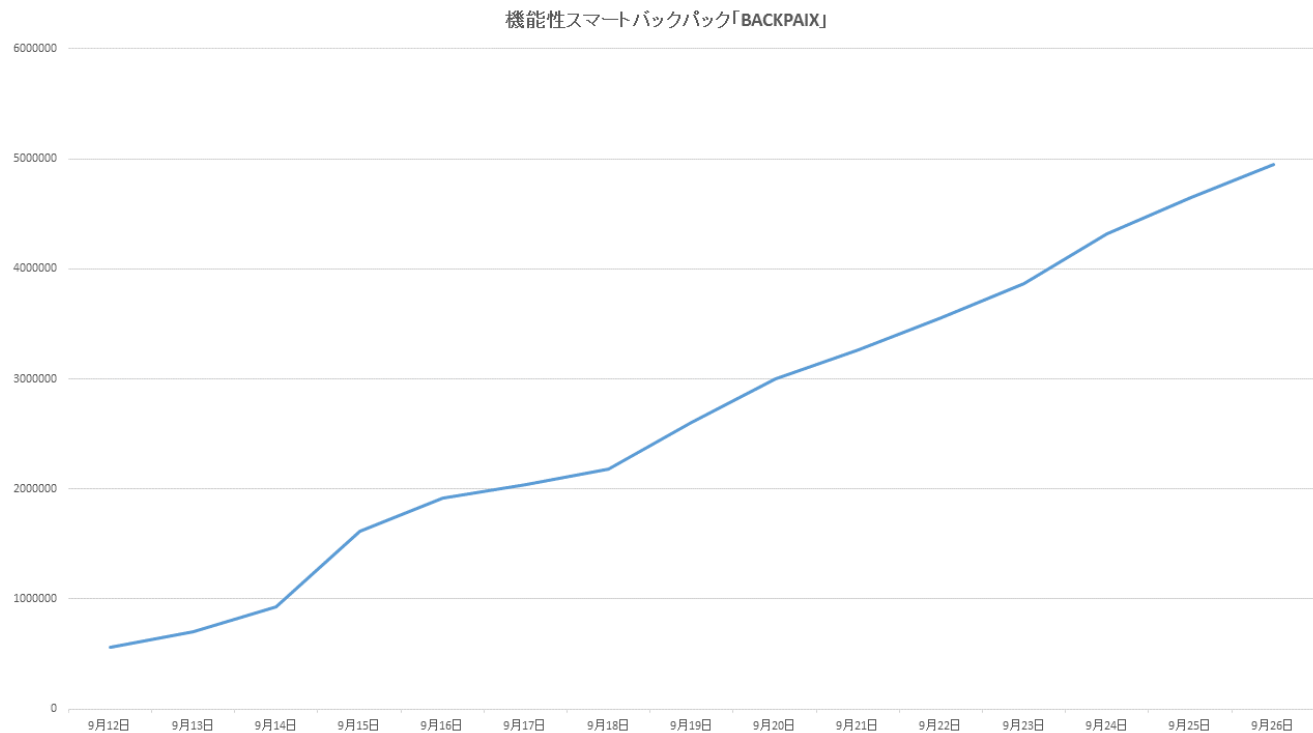


図 5.28 調達資金の時間変化

このグラフから 9 月 14 日から 9 月 15 日にかけて金額が多く集まっていることがわかる．

5.8.5 調査項目

プロジェクト実行者が資金が集まり始める前にしている行動を調査する．調査項目は以下の4項目である．

1. サイト内の活動レポートを活用しているか．
2. 動画を投稿しているか．
3. Twitter でツイートをしているか．
4. Facebook で投稿しているか．

The screenshot shows a crowdfunding project page for 'BACKPAIX'. The main banner features a black backpack with a circular logo and the text 'ついに日本上陸' (Finally landing in Japan) and '普通じゃもの足りない人に 機能性スマートバックパック' (For people who find ordinary backpacks insufficient, a functional smart backpack). A '再生' (Play) button is visible. The right sidebar displays the campaign status: '集まっている金額' (Amount raised) is 20,794,840円, '目標金額' (Target amount) is 300,000円, and the progress bar is at 6931%. It also shows '789人' (789 supporters) and '残り 0日' (0 days remaining). A '終了しました' (Ended) button is at the bottom. The top navigation bar includes links for 'プロジェクト' (Project), '活動レポート' (Activity Report), '応援コメント' (Supporting Comments), and 'ストア' (Store). The bottom of the page has social media sharing buttons for 'テクノロジー' (Technology), 'いいね!' (Like), 'ツイート' (Tweet), 'G+1' (Google Plus), 'Bookmark', 'Amebaに投稿' (Post to Ameba), and 'WIDGET'.

TOP > プロジェクト一覧 > テクノロジー > 普通じゃもの足りない人に、機能性スマートバックパック「BACKPAIX」

プロジェクト **Success!** 活動レポート **5** 応援コメント **99+** ストア **販売中** お気に入り

ついに日本上陸
普通じゃもの足りない人に
機能性スマートバックパック

▶ 再生

BACKPAIX

集まっている金額 **Success!**
20,794,840円
目標金額 300,000円
6931%

サポーター **789人**
残り **0日**

終了しました

普通じゃもの足りない人に、機能性スマートバックパック「BACKPAIX」

テクノロジー いいね! 2,844 ツイート G+1 19 Bookmark 7 Amebaに投稿 WIDGET

図 5.29 プロジェクトページ

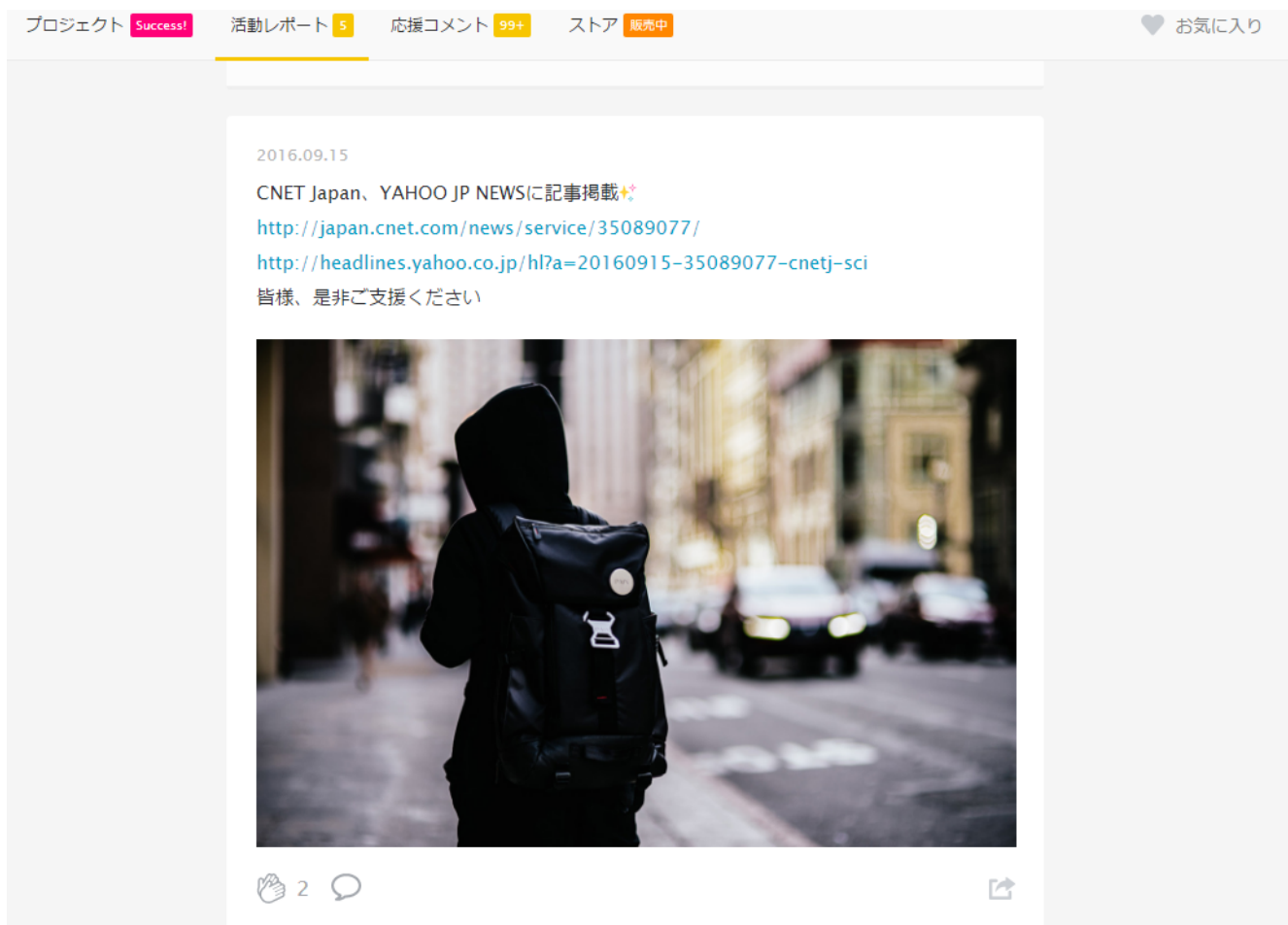


図 5.30 サイト内の活動レポートページ

9月15日に活動レポートを上げていることがわかる。



PAIX Design Japan @PAIXDesignJP · 2016年9月14日

BACKPAIX 👍

CNET Japan、YAHOO JP NEWSに記事掲載 ✨

皆様、是非ご支援ください 😊😊😊

👉👉 makuake.com/project/backpa...

#バックパック #リュック

#スマートバッグ #PAIX #BACKPAIX @cnet_japan



普通じゃもの足りない人に、機能性スマートバックパック「BACKPAIX」 | ク...

「BACKPAIX」は 普通じゃもの足りないさを感じている人たちにに向けて生まれた、機能性スマートバックパックです。充電ポート、シューズポケット、防水などの機...

makuake.com



図 5.31 プロジェクト実行者の Twitter

9月14日にツイートしていることがわかる。



PAIX

2016年9月14日 · 公開

Techableの乾さん、
ご掲載をありがとうございます👍👍👍

#バックパック #リュック

#スマートバッグ... もっと見る



【Interview】普通じゃ物足りない人に!スマートバッグの風雲
児「BACKPAIX」の魅力に迫る | Techable(テッカブル)

単なる道具に留まらない、さまざまな機能を持ったスマートバッグが、このところ
にわかに注目を集めている。海外のクラウドファンディングにも、多彩なプロ...

TECHABLE.JP

👍 いいね!

💬 コメントする

➦ シェアする

👍 13人

図 5.32 プロジェクト実行者の Facebook

9月14日に投稿していることがわかる。

	A	B	C	D	E	F	G	H
1	プロジェクト名	レポート	動画	FB	ツイート	目標金額	集めた金額	目標達成率
2	3dnamaz	1	0	0	0	300,000	361,000	1.20
3	3rrr	0	0	0	0	1,000,000	247,340	0.25
4	18steakdiner	0	0	0	0	1,500,000	294,000	0.20
5	28-year-old-ur	0	1	0	0	200,000	41,450	0.21
6	29on	1	1	0	0	1,000,000	9,406,000	9.41
7	29on02	0	0	0	0	3,000,000	14,245,000	4.75
8	akatsuki	0	0	1	0	500,000	24,000	0.05
9	72recipes	0	0	0	0	2,000,000	1,113,000	0.56
10	acoma	1	0	0	0	1,000,000	465,000	0.47
11	ahflowers	0	0	0	1	300,000	96,000	0.32
12	aji-project	0	0	1	0	500,000	67,500	0.14
13	albirexbb-rabbit	0	0	1	0	200,000	1,349,000	6.75
14	ale14	0	0	0	0	1,000,000	1,745,000	1.75
15	alma	1	0	0	0	200,000	977,200	4.89
16	amekaze-sakai	1	0	0	0	700,000	276,600	0.40
17	aobaart	1	0	0	0	250,000	87,500	0.35
18	aopenta	0	0	0	0	650,000	71,000	0.11
19	atom	0	0	1	0	2,000,000	2,206,440	1.10
20	azlien	1	0	0	0	500,000	114,000	0.23
21	backpaix	1	0	1	1	300,000	20,794,840	69.32
22	base-pasta	0	0	1	1	500,000	1,092,860	2.19
23	biblle	1	0	0	0	800,000	1,323,400	1.65
24	bioproject	0	0	0	0	300,000	21,060	0.07
25	biroku	0	0	1	1	700,000	210,200	0.30
26	bitszaidan4	0	0	1	0	250,000	113,000	0.45
27	bobby	1	0	0	0	500,000	20,339,025	40.68
28	bokin-chokin	0	0	0	0	500,000	3,200	0.01
29	bonnel-cafe	0	0	0	1	300,000	416,000	1.39
30	brickbrick	0	0	1	1	300,000	31,500	0.11
31	brownsugargine	0	0	1	0	300,000	408,850	1.36
32	budononagaya	0	0	0	0	500,000	612,000	1.22
33	cadillac62	0	0	0	0	600,000	60,000	0.10
34	cafeKitchen333	0	0	1	0	600,000	149,000	0.25
35	chanzy	1	0	0	0	600,000	196,530	0.33
36	chocolabb-rich	0	0	0	0	1,000,000	1,163,639	1.16
37	coccodrillo	0	0	1	0	300,000	86,000	0.29
38	coffeemafia	0	0	1	0	600,000	606,000	1.01
39	coffeesenka-yc	0	0	1	0	200,000	208,000	1.04
40	coffeetonya	1	0	0	1	300,000	373,000	1.24
41	coffee-wrights	1	0	0	0	1,500,000	1,642,000	1.09
42	cookingmagicia	0	0	1	0	600,000	87,500	0.15
43	cotohakyoto	1	0	1	0	1,000,000	627,000	0.63
44	creshedam	1	0	0	0	100,000	13,000	0.13
45	crosslinkres	1	0	1	1	300,000	39,840	0.13
46	crossroad-furr	0	0	0	0	300,000	39,840	0.13
47	cummu	0	0	1	0	1,000,000	931,110	0.93
48	decadentminim	1	0	0	1	300,000	6,710,920	22.37
49	decorebonne	1	0	0	0	200,000	224,850	1.12

図 5.33 調査結果のデータ 1

50	denshiblock	1	0	0	0	300,000	127,500	0.43
51	desk-dod	0	0	1	0	170,000	54,668	0.32
52	dmcjapandj2	0	0	1	0	300,000	498,000	1.66
53	dot-ebisu	0	0	1	0	1,728,000	306,000	0.18
54	donguri-sake	0	0	1	0	300,000	40,000	0.13
55	dronemiyazaki	0	1	0	0	500,000	49,000	0.10
56	easyonetouch	0	0	1	0	300,000	648,905	2.16
57	editory	0	1	0	0	500,000	29,000	0.06
58	edoweek	0	0	1	0	3,000,000	247,000	0.08
59	effortico	1	1	1	0	500,000	1,191,400	2.38
60	eremote-mini	1	0	1	0	800,000	5,870,540	7.34
61	faith	0	0	0	0	300,000	336,920	1.12
62	ferrarif1	1	0	0	1	300,000	200,000	0.67
63	framboise	1	0	1	1	1,250,000	1,464,000	1.17
64	fruitbouquet	0	0	1	0	500,000	1,192,000	2.38
65	funewoamu	0	0	1	0	1,000,000	1,058,400	1.06
66	gamespark	1	0	1	1	500,000	1,771,000	3.54
67	genecre2	0	1	0	1	700,000	3,667,500	5.24
68	georgianwine	1	0	0	0	1,000,000	2,567,800	2.57
69	gfcafe-littlebird	0	1	1	0	1,000,000	74,000	0.07
70	good-morning-t	1	1	1	0	500,000	595,000	1.19
71	global-itnet2	0	0	0	0	324,000	151,200	0.47
72	globlue	0	1	1	0	500,000	943,900	1.89
73	gohan	0	0	1	1	1,500,000	473,000	0.32
74	goodjuice	0	1	1	1	500,000	240,000	0.48
75	gussu-refle	0	0	0	0	500,000	1,546,460	3.09
76	heat-up	0	0	1	1	100,000	166,000	1.66
77	i-cave	1	0	1	1	200,000	810,600	4.05
78	iemen-stadium	0	0	0	0	500,000	576,040	1.15
79	ideaftp	1	0	1	0	1,000,000	1,205,000	1.21
80	iguchi-ke	0	1	0	0	300,000	747,800	2.49
81	jack-daniels	0	1	1	0	800,000	2,111,000	2.64
82	imagetrainingen	1	0	1	1	500,000	388,900	0.78
83	iss-audition	0	0	1	1	1,000,000	115,000	0.12
84	insterra	1	0	1	1	150,000	354,860	2.37
85	intexs	0	1	0	0	500,000	24,000	0.05
86	islay-tokyo	0	0	1	0	500,000	729,800	1.46
87	jimbocho_movie	1	0	1	1	500,000	32,000	0.06
88	itva-jp	0	0	0	0	500,000	19,000	0.04
89	jackalman	0	0	1	0	1,000,000	1,555,000	1.56
90	jfast	1	0	1	1	300,000	132,000	0.44
91	jinenjo	0	0	0	0	1,000,000	55,000	0.06
92	kamibot	0	0	1	1	500,000	978,300	1.96
93	kanefactory	0	0	1	0	300,000	453,800	1.51
94	kaoricomedy	1	0	0	0	1,000,000	1,378,000	1.38
95	kehara	1	0	1	1	1,000,000	1,123,000	1.12
96	kinowa	1	0	1	1	300,000	55,400	0.18
97	kitst128	1	0	0	0	1,500,000	3,222,500	2.15
98	kodomotokimon	1	0	1	1	800,000	1,921,500	2.40
99	kuratsukisc	0	1	1	1	350,000	24,000	0.07
100	ke-i	1	0	1	0	1,000,000	5,153,300	5.15
101	landmade	1	0	1	0	300,000	883,000	2.94

100 件のプロジェクトから調査した 4 項目と目標金額，集めた金額を Excel にまとめる．
集めた金額を目標金額で割り，目標達成率も算出する．

5.8.6 R を使用した分析

R を使用し、100 件のデータから決定木分析を行う。R で分析を行うにあたって調査結果を分析しやすくした csv ファイルを作成する。

	A	B	C	D	E
1	目標達成率	レポート	動画	FB	ツイート
2	1.20	1	0	0	0
3	0.25	0	0	0	0
4	0.20	0	0	0	0
5	0.21	0	1	0	0
6	9.41	1	1	0	0
7	4.75	0	0	0	0
8	0.05	0	0	1	0
9	0.56	0	0	0	0
10	0.47	1	0	0	0
11	0.32	0	0	0	1
12	0.14	0	0	1	0
13	6.75	0	0	1	0
14	1.75	0	0	0	0
15	4.89	1	0	0	0
16	0.40	1	0	0	0
17	0.35	1	0	0	0
18	0.11	0	0	0	0
19	1.10	0	0	1	0
20	0.23	1	0	0	0
21	69.32	1	0	1	1
22	2.19	0	0	1	1
23	1.65	1	0	0	0
24	0.07	0	0	0	0
25	0.30	0	0	1	1
26	0.45	0	0	1	0
27	40.68	1	0	0	0
28	0.01	0	0	0	0
29	1.39	0	0	0	1
30	0.11	0	0	1	1
31	1.36	0	0	1	0
32	1.22	0	0	0	0
33	0.10	0	0	0	0
34	0.25	0	0	1	0
35	0.33	1	0	0	0
36	1.16	0	0	0	0
37	0.29	0	0	1	0
38	1.01	0	0	1	0
39	1.04	0	0	1	0
40	1.24	1	0	0	1
41	1.09	1	0	0	0
42	0.15	0	0	1	0
43	0.63	1	0	1	0
44	0.13	1	0	0	0
45	0.13	1	0	1	1
46	0.13	0	0	0	0
47	0.93	0	0	1	0
48	22.37	1	0	0	1
49	1.12	1	0	0	0
50	0.43	1	0	0	0

図 5.35 作成した csv ファイル 1

	A	B	C	D	E
50	0.43	1	0	0	0
51	0.32	0	0	1	0
52	1.66	0	0	1	0
53	0.18	0	0	1	0
54	0.13	0	0	1	0
55	0.10	0	1	0	0
56	2.16	0	0	1	0
57	0.06	0	1	0	0
58	0.08	0	0	1	0
59	2.38	1	1	1	0
60	7.34	1	0	1	0
61	1.12	0	0	0	0
62	0.67	1	0	0	1
63	1.17	1	0	1	1
64	2.38	0	0	1	0
65	1.06	0	0	1	0
66	3.54	1	0	1	1
67	5.24	0	1	0	1
68	2.57	1	0	0	0
69	0.07	0	1	1	0
70	1.19	1	1	1	0
71	0.47	0	0	0	0
72	1.89	0	1	1	0
73	0.32	0	0	1	1
74	0.48	0	1	1	1
75	3.09	0	0	0	0
76	1.66	0	0	1	1
77	4.05	1	0	1	1
78	1.15	0	0	0	0
79	1.21	1	0	1	0
80	2.49	0	1	0	0
81	2.64	0	1	1	0
82	0.78	1	0	1	1
83	0.12	0	0	1	1
84	2.37	1	0	1	1
85	0.05	0	1	0	0
86	1.46	0	0	1	0
87	0.06	1	0	1	1
88	0.04	0	0	0	0
89	1.56	0	0	1	0
90	0.44	1	0	1	1
91	0.06	0	0	0	0
92	1.96	0	0	1	1
93	1.51	0	0	1	0
94	1.38	1	0	0	0
95	1.12	1	0	1	1
96	0.18	1	0	1	1
97	2.15	1	0	0	0
98	2.40	1	0	1	1
99	0.07	0	1	1	1
100	5.15	1	0	1	0
101	2.94	1	0	1	0

図 5.36 作成した csv ファイル 2

C ドライブに `cit` という名前のフォルダを作成し、そこに分析する `csv` ファイルを置く。

R を起動し、決定木を作成するのに必要なパッケージをインストールする。

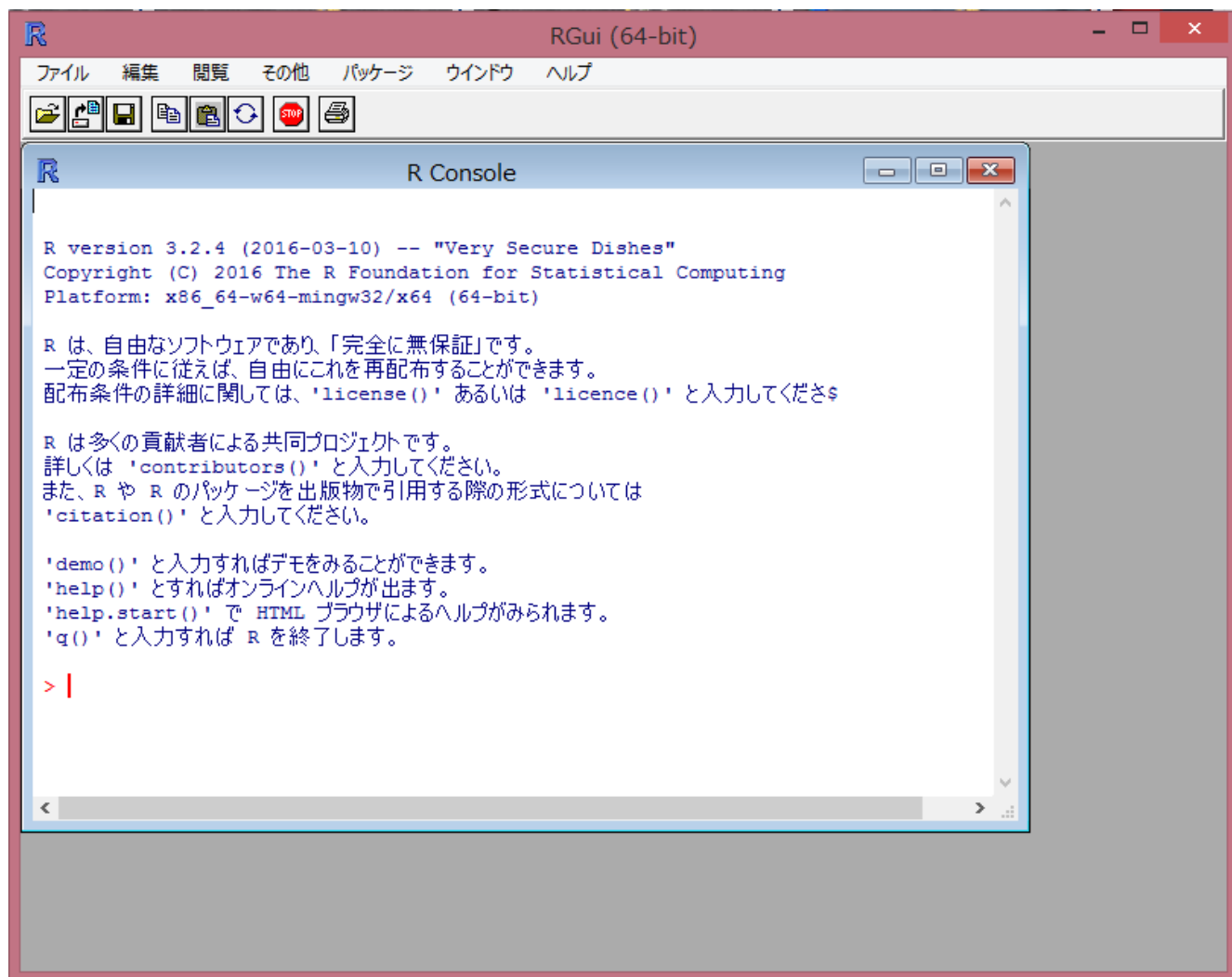


図 5.37 起動後の R

以下のコマンドを実行する。

```
install.packages("rpart")  
install.packages("rpart.plot")
```

実行するとサーバーを選択をするので japan を選ぶ .

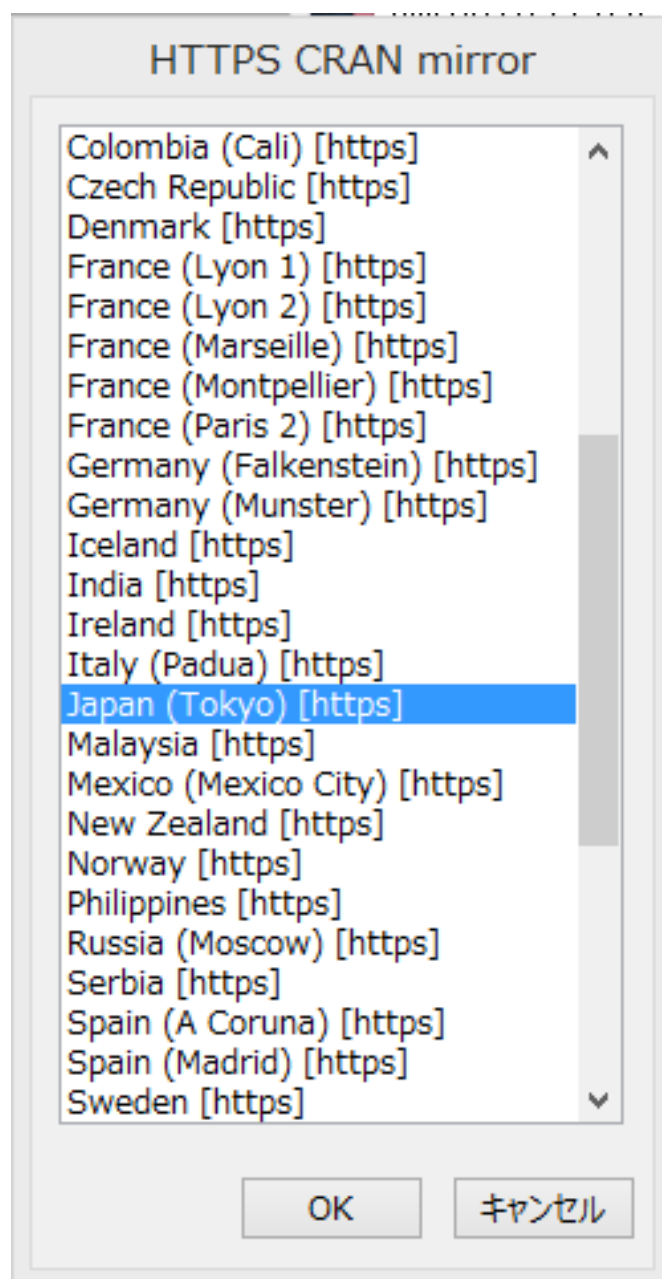


図 5.38 サーバーの選択

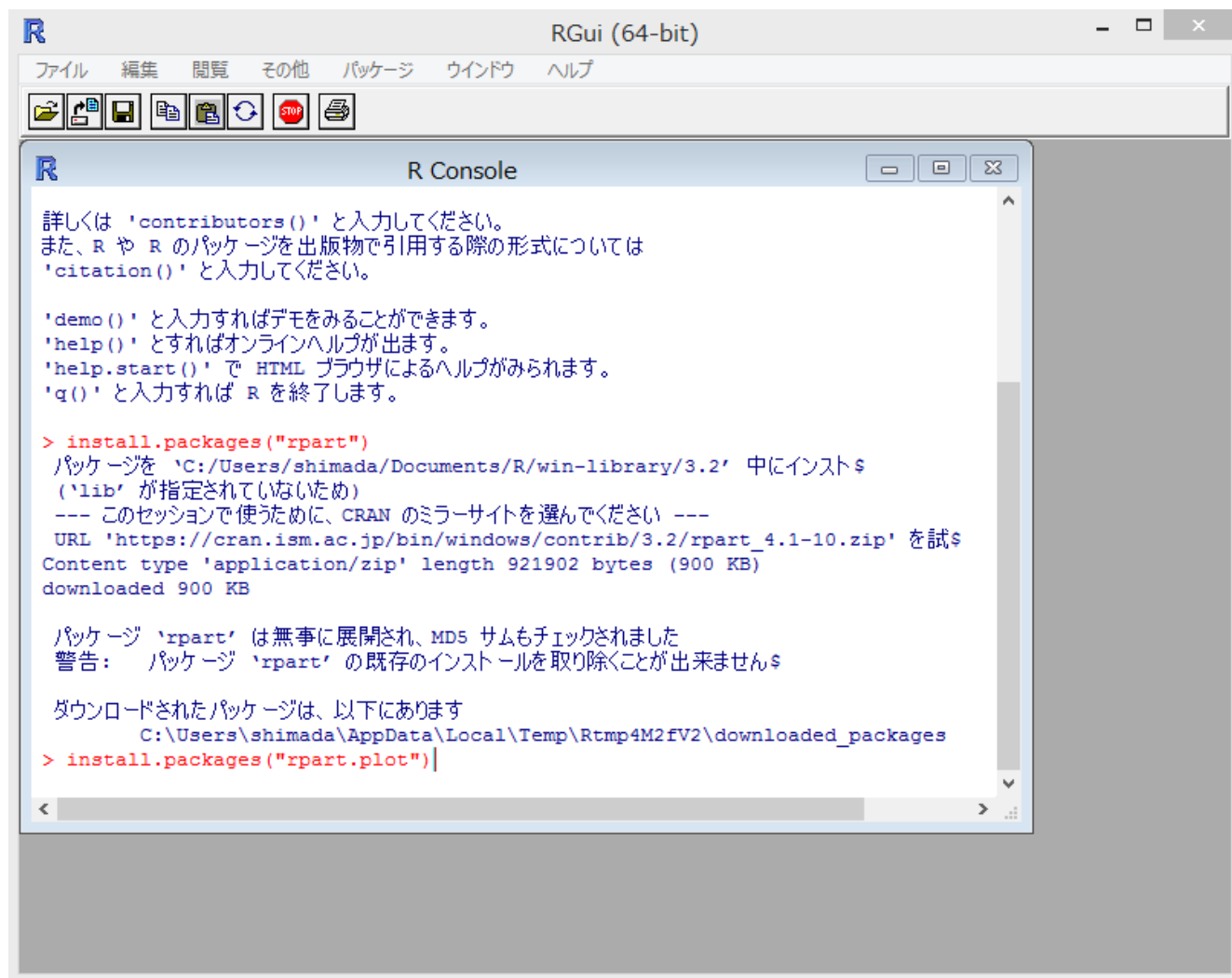


図 5.39 インストール完了画面

このようになればインストール完了。

以下のコマンドを実行し決定木分析を行う。

```
library(rpart)
library(rpart.plot)
setwd("C:/cit")
df <- read.csv("keka.csv")
data.rp <- rpart(目標達成率~,data=df)
```

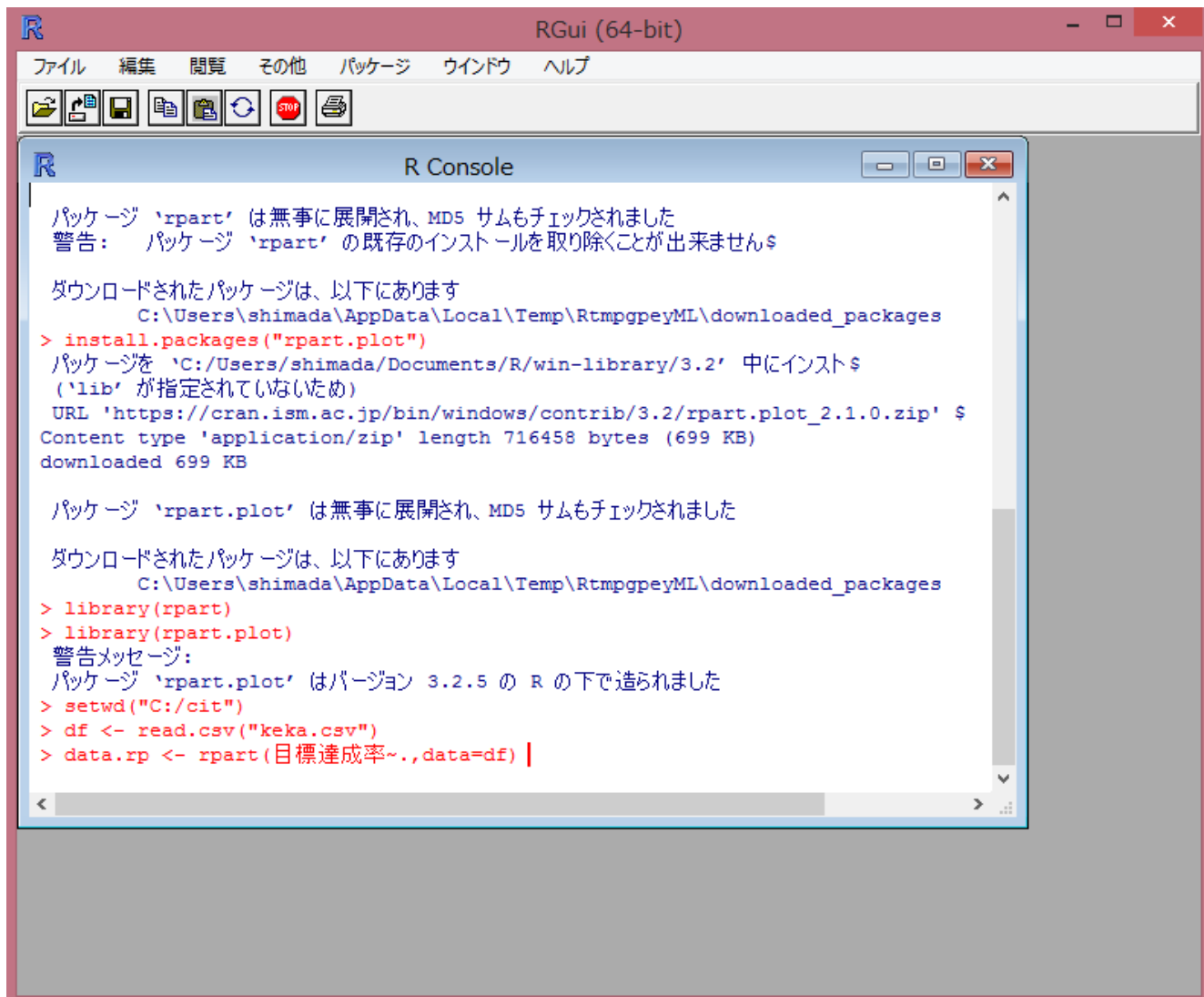


図 5.40 決定木分析

決定木分析の結果をテキストで出力する．以下のコマンドを実行する．

```
print(data.rp)
```

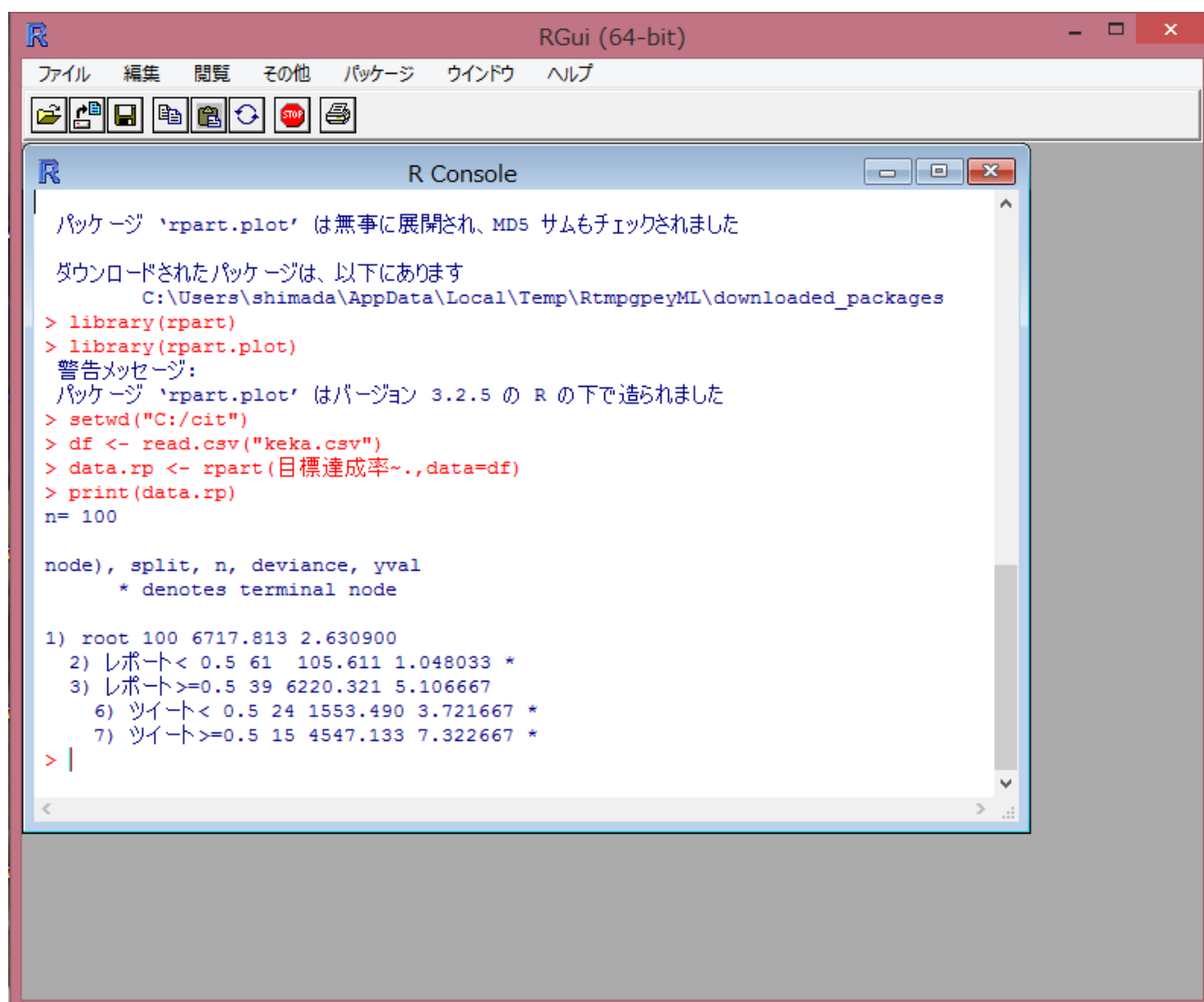


図 5.41 テキストで出力

最後に決定木を図として出力する．以下のコマンドを実行する．

`rpart.plot(data.rp)`

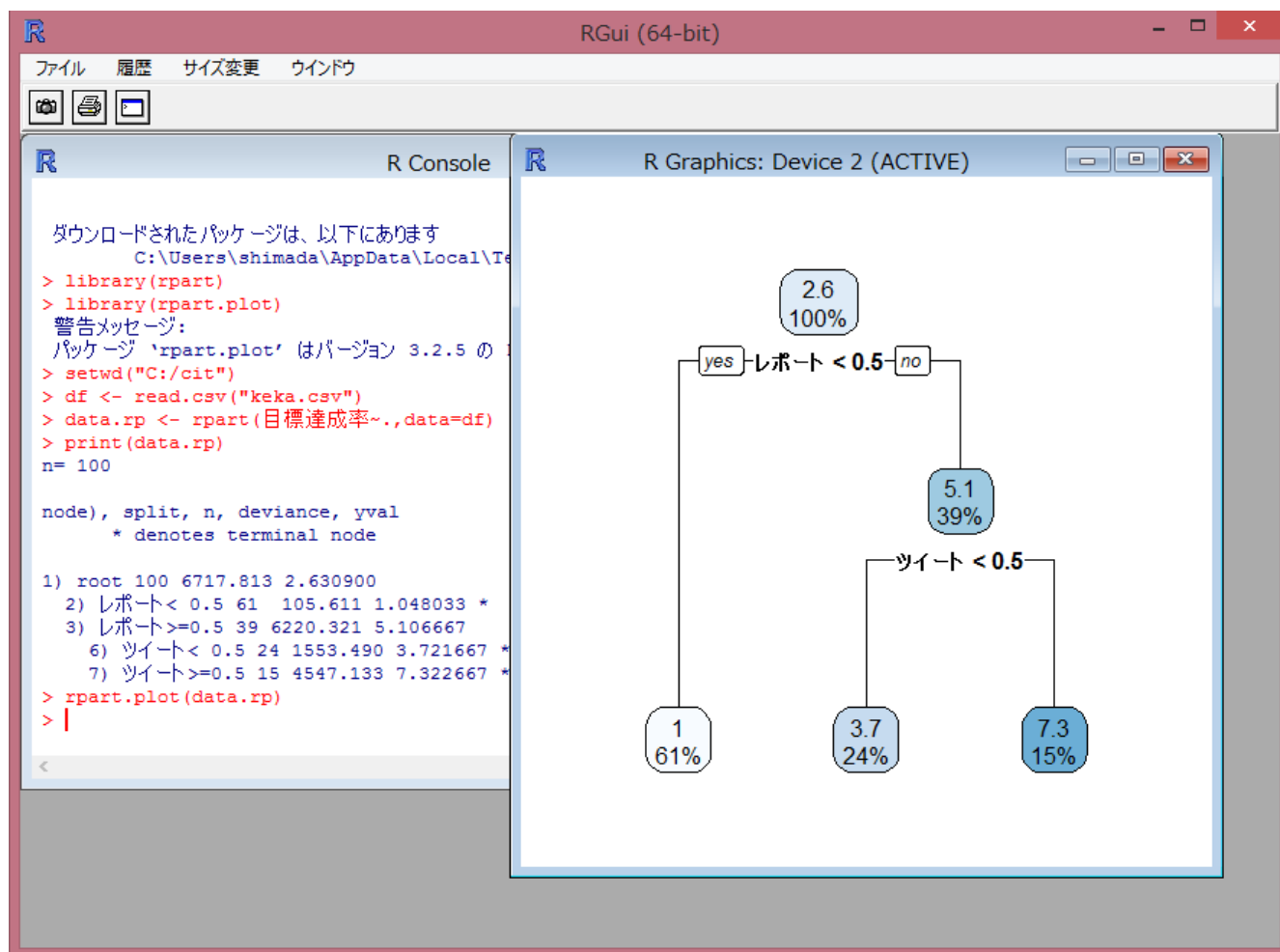


図 5.42 図で出力

第 6 章

結果

目標達成率を目的変数とにおいて，レポートとツイートが最も資金調達に関係が深い説明変数となった．

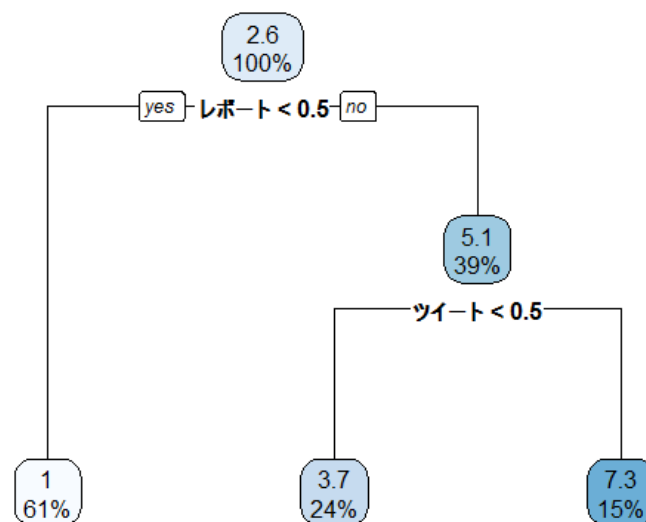


図 6.1 決定木分析結果

第 7 章

考察

レポートとツイートをしてるプロジェクトが資金の調達に最も成功していることがわかる．レポートはプロジェクトのページからすぐに閲覧可能なため，支援者の目につきやすいからだと考えられる．ツイートに関しては，情報の拡散力の高い Twitter によって多数の支援者の目に留まり資金が集まったと考えられる．

第 8 章

結論

本研究で，クラウドファンディングにおいて資金調達する際にプロジェクト実行者がすべき行動はレポートとツイートであるという結果が出た．上記の項目以外に資金調達に関係があると思われる項目を増やしていくことで，さらに精度の高い結果を得られるはずだ．

参考文献

- [1] 日本の主要クラウドファンディング 累計支援額 月次推移 (積み上げグラフ), 2017/01/23. <http://visualizing.info/article/4255.html>.
- [2] Wikipedia クラウドファンディング, 2017/01/23. <https://ja.wikipedia.org/wiki/>
- [3] クラウドファンディングの 3 つの種類「購入型」「寄付型」「金融型」の違いと特徴, 2016/09/20. <http://anipop.com/crowdfunding-3type/>.
- [4] クラウドファンディングの歴史と日本のポテンシャルについて, 2017/01/23. <http://thebridge.jp/2014/04/history-crowdfunding-japanese-potential>.
- [5] 三浦泰介. クラウドファンディングにおける成功の判別分析. 卒業論文, 千葉工業大学, 2015.
- [6] クローラ, 2017/01/23. <https://ja.wikipedia.org/wiki/>
- [7] ダウンロードコマンド wget のオプション一覧 (linux), 2017/01/23. http://blog.layer8.sh/ja/2012/03/31/wget_command/.

謝辞

本研究を進めるにあたり，矢吹研究室矢吹太郎准教授には，多くの時間をご指導にさいて頂きました．また，先行研究を行った三浦泰介先輩をはじめ矢吹研究室の皆様には，多くの知識や示唆を頂きました．協力していただいた皆様に感謝の気持ちと御礼を申し上げます．