

R で計量政治学入門

土井 翔平

2020-04-20

目次

はじめに

本書は R による計量政治学の入門レベルの講義資料です。質問や間違いなどがありましたら、ご連絡を下さい。筆者のプロフィールはこちらをご覧ください。

想定する読者

本書は、データ分析や数学の前提知識やプログラミング経験のない社会科学系学部生を主たる読者として想定しています。

- ・やや高度と思われる箇所には * を付けているので、読み飛ばしても構いません。

なお、R や RStudio のインストールについては R の分析環境を、基本操作については R プログラミング入門をご覧ください。

Tidyverse について

Tidyverse とは様々なデータ操作に関するパッケージ群（あるいはそのプロジェクト）を指します。本書では、可能な限り、R の標準関数を用いた表記と Tidyverse による表記を併記するようにします。しかし、筆者は Tidyverse に慣れているので、しばしば標準関数によるコードを省略します。

ウェブサイトの操作

本書は bookdown を用いて作成しています。ウェブサイトのナビゲーションバーでは、

- ・三本線のボタンで目次の表示・非表示の切り替え
- ・虫眼鏡のマークで単語検索
- ・A のマークで文字の大きさ、フォント、色のコントロール
- ・ダウンロードボタンで .pdf ファイルや .epub ファイルのダウンロード
- ・i のマークでキーボードによる操作方法の表示

が可能です。

第1部

データ・ハンドリング

第1章

データの読み込み

本章ではデータを読み込む方法について解説します。

```
library(tidyverse)
```

1.1 パッケージ付属のデータ

Rは標準でいくつかのデータセットを持っており、またパッケージを読み込むと付属のデータセットも読み込みます。`data()`に何も入力せずに実行すると、データセットの一覧が表示されます。

```
data()
```

よく、使われるデータセットはフィッシャーのアヤメのデータセットで、`iris`という名前で保存されています。

```
head(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

- `head()`は最初のいくつかの要素だけを表示する関数です（`tail()`は最後からいくつかを表示します）。

1.2 .csv ファイルの読み込み

第 II 部

R/RStudio

第2章

R の分析環境

R は統計用のプログラミング言語です。他に、特に機械学習の分野では Python や Julia も人気です。

- ・政治学や経済学では Stata という統計ソフトも人気ですが有料という難点があります。

また、RStudio は R を便利に使うための統合開発環境 (IDE) です。RStudio 以外にもあるもののデファクトスタンダードになっている感はあります。

RStudio はあくまで R を使いやすくするためのもので、R 本体ではありません。なので、まずは R をインストールしてから RStudio をインストールします。

2.1 R のインストール

2.1.1 ダウンロード

まずは R の[公式サイト](#)へ行き（右クリックで新しいタブで開くことができます）、**download R**をクリックします。



[Home]

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Conferences

Search

Get Involved: Mailing Lists

Developer Pages

R Blog

R Foundation

Foundation

Board

Members

Donors

Donate

Help With R

Getting Help

Documentation

Manuals

FAQs

The R Journal

Books

Certification

Other

Links

Bioconductor

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and Mac OS. To [download R](#), please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 3.6.0 (Planting of a Tree) prerelease versions** will appear starting Tuesday 2019-03-26. Final release is scheduled for Friday 2019-04-26.
- useR! 2020 will take place in St. Louis, Missouri, USA.
- **R version 3.5.3 (Great Truth)** has been released on 2019-03-11.
- The R Foundation Conference Committee has released a [call for proposals](#) to host useR! 2020 in North America.
- You can now support the R Foundation with a renewable subscription as a [supporting member](#)
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.

News via Twitter

The R Foundation Retweeted



Forwards
@R_Forwards

Women leading the Artificial Intelligence industry: an interview with @JulieJosseStat, Author of #rstats packages including FactoMineR, denoiseR and missMDA. Member of @_R_Foundation and core team member of #R_Forwards.
medium.com/authority-maga...



次にダウンロードする際のミラーサイトを選びます。好きな国の人を選んでいいですが、ここでは日本の統計数理研究所のものを選んでおきます。

	https://cran.r-project.org/	Marine Research Institute
Indonesia	http://cran.hafro.is/	Agency for The Application and Assessment of Technology
Iran	https://repo.bppt.go.id/cran/	Ferdowsi University of Mashhad
	https://cran.um.ac.ir/	Ferdowsi University of Mashhad
Ireland	https://ftp.heanet.ie/mirrors/cran.r-project.org/	HEAnet,Dublin
	http://ftp.heanet.ie/mirrors/cran.r-project.org/	HEAnet,Dublin
Italy	http://cran.mirror.garr.it/mirrors/CRAN/	Garr Mirror, Milano
	https://cran.stat.unipd.it/	University of Padua
	http://cran.stat.unipd.it/	University of Padua
Japan	https://cran.ism.ac.jp/	The Institute of Statistical Mathematics, Tokyo
	http://cran.ism.ac.jp/	The Institute of Statistical Mathematics, Tokyo
	https://ftp.yz.yamagata-u.ac.jp/pub/cran/	Yamagata University
Korea	https://ftp.harukasan.org/CRAN/	Information and Database Systems Laboratory, Pukyong National University
	https://cran.yu.ac.kr/	Yeungnam University
	https://cran.seoul.go.kr/	Bigdata Campus, Seoul Metropolitan Government
	http://healthstat.snu.ac.kr/CRAN/	Graduate School of Public Health, Seoul National University, Seoul
	https://cran.bioidisk.org/	The Genome Institute of UNIST (Ulsan National Institute of Science and Technology)
	http://cran.bioidisk.org/	The Genome Institute of UNIST (Ulsan National Institute of Science and Technology)
Malaysia	https://wbc.upm.edu.my/cran/	Univerisiti Putra Malaysia
	http://wbc.upm.edu.my/cran/	Univerisiti Putra Malaysia
Mexico	https://cran.itam.mx/	Instituto Tecnologico Autonomo de Mexico
	http://cran.itam.mx/	Instituto Tecnologico Autonomo de Mexico
New Zealand		

自分の PC の OS に応じたものを選択します。



[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2019-03-11, Great Truth) [R-3.5.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are “R” and “CRAN”? ある しいらん(‘・ω・’)

■2.1.1.0.1 Windows の場合

install R for the first time を選択します。



[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)

R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to install R for the first time .
contrib	Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#)

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Download R X.X.X for YYY を選択してダウンロードします。

The screenshot shows the CRAN R download page for Windows. At the top, there's a large blue 'R' logo. Below it, a sidebar on the left lists various links: CRAN, Mirrors, What's new?, Task Views, Search, About R, R Homepage, and The R Journal. The main content area has a title 'R-3.5.3 for Windows (32/64 bit)'. A prominent yellow button labeled 'Download R 3.5.3 for Windows (79 megabytes, 32/64 bit)' is centered. Below this button are links for 'Installation and other instructions' and 'New features in this version'. A note below the download button says: 'If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.' There's also a 'Frequently asked questions' section with links to 'Does R run under my version of Windows?', 'How do I update packages in my previous version of R?', and 'Should I run 32-bit or 64-bit R?'. A note at the bottom of this section says: 'Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.' Another section titled 'Other builds' contains links to 'Patches to this release are incorporated in the [r-patched snapshot build](#)', 'A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#)', and 'Previous releases'. A note at the bottom of this section says: 'Note to webmasters: A stable link which will redirect to the current Windows binary release is [<CRAN MIRROR>/bin/windows/base/release.htm](#)'.

分かりやすいようにダウンロードフォルダにダウンロードしておきます。

Windowsの場合、Rtoolsをインストールもインストールしておきましょう。

■2.1.1.0.2 Macintosh の場合

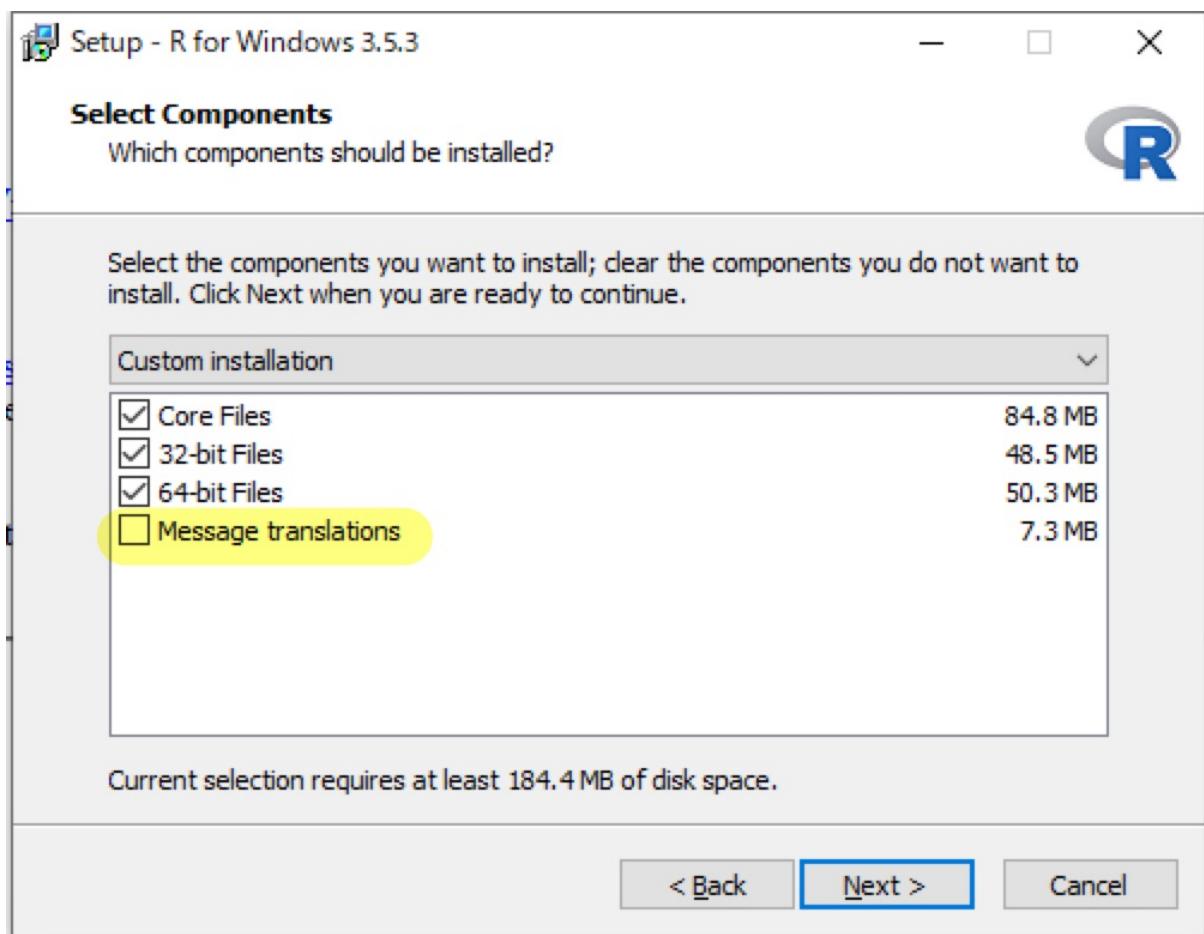
2.1.2 インストール

Rをダウンロードしたフォルダを開き、ファイルをクリックします。

- ・ファイル名はOSによって異なります。

その後は表示されるままに進めていけばよいです。

Rは基本的にOSの言語で表示されますが、英語で使いたい場合はMessage Translationsのインストールにチェックが入っている場合は外しておきましょう。

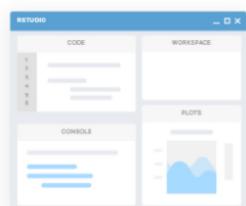


- ・英語のエラーメッセージで検索したほうが解決策が見つけやすくなります。

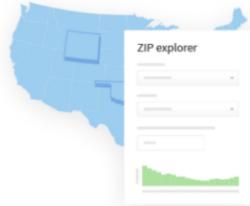
2.2 RStudio のインストール

2.2.1 ダウンロード

RStudio の公式サイトから RStudio のダウンロードサイトへ行きます。



RStudio



Shiny



R Packages

RStudio makes R easier to use. It includes a code editor, debugging & visualization tools.

[Download](#)

[Learn More](#)

Shiny helps you make interactive web applications for visualizing data. Bring R data analysis to life.

[Learn More](#)

Our developers create popular packages to expand the features of R. Includes ggplot2, dplyr, R Markdown & more.

[Learn More](#)

FEATURED CUSTOMER SPOTLIGHT

Increasing the Impact of Data at Stack Overflow

"We were spending too much time on the logistics of sharing our results," says India

下の方にインストーラーをダウンロードするリンクがあるのでOSに応じたものを選択します。

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.463 - Windows Vista/7/8/10	85.8 MB	2018-10-29	58b3d796d8cf96fb8580c62f46ab64d4
RStudio 1.1.463 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-10-29	a79032ba4d7daaa86a8da01948278d94
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-10-29	8a6755fa9fae2bafce289df3358aaf63
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-10-29	bc50d6bd34926c1cc3ae4a209d67d649
RStudio 1.1.463 - Ubuntu 16.04+/Debian 9+ (64-bit)	65 MB	2018-10-29	cf659db18619cc78d1592fefaa7c753
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-10-29	742f0bad60dfeaa3281576e14ad6699e
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-10-29	c703067a0ca99d6ea7e427b856952d1

Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 1.1.463 - Windows Vista/7/8/10	122.9 MB	2018-10-29	1eb1d7758bd4bf4bb68d4a7c3fe8d894
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	90 MB	2018-10-29	ef9242b621d36c30de9d86b808840b41
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	98.3 MB	2018-10-29	5944b3dd118cfba46f2a6c484d768324
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.8 MB	2018-10-29	6bee446eeb4cc0e32967845e1f7fe2b
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	91.4 MB	2018-10-29	3f63725769b7b976fedb754eb2a19108

Source Code

A tarball containing source code for RStudio v1.1.463 can be downloaded from [here](#)



- ・ 安定版ではないけれど最新の RStudio を使いたい人は [RStudio Preview](#)をインストールしてください。
- ・ また、R や RStudio をインストールせずにオンラインで使用できる [RStudio Cloud](#)というものもあります。

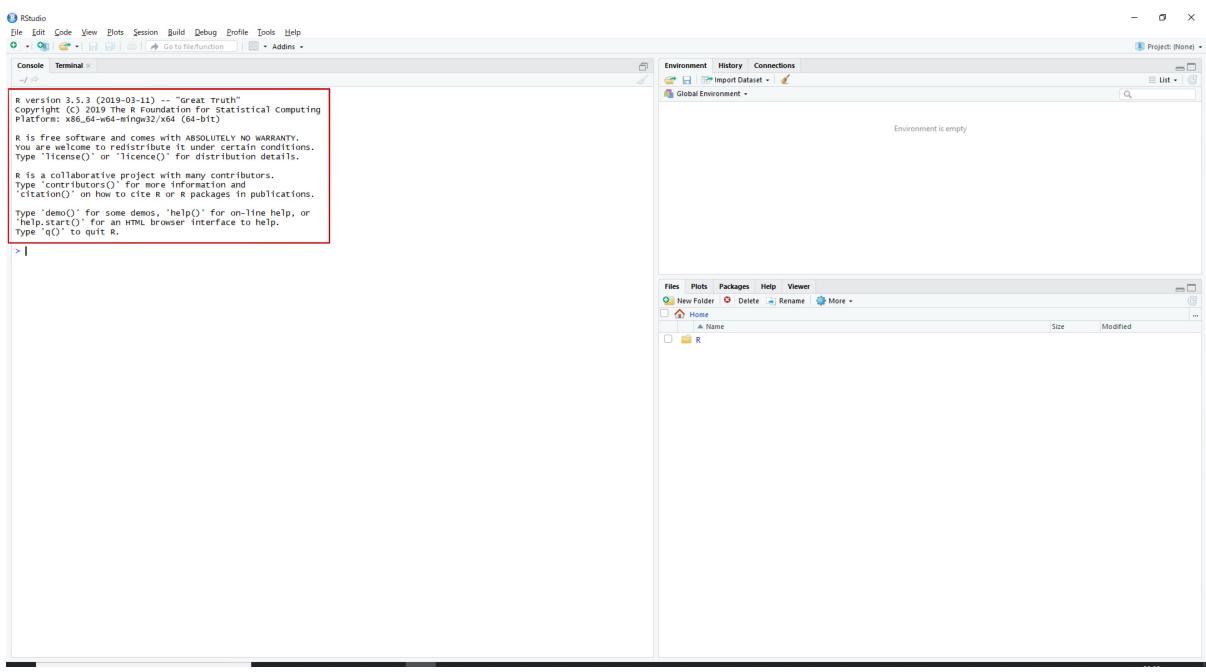
2.2.2 インストール

あとはダウンロードしたフォルダに移り、インストーラーを起動して表示されるがままに進めていきます。

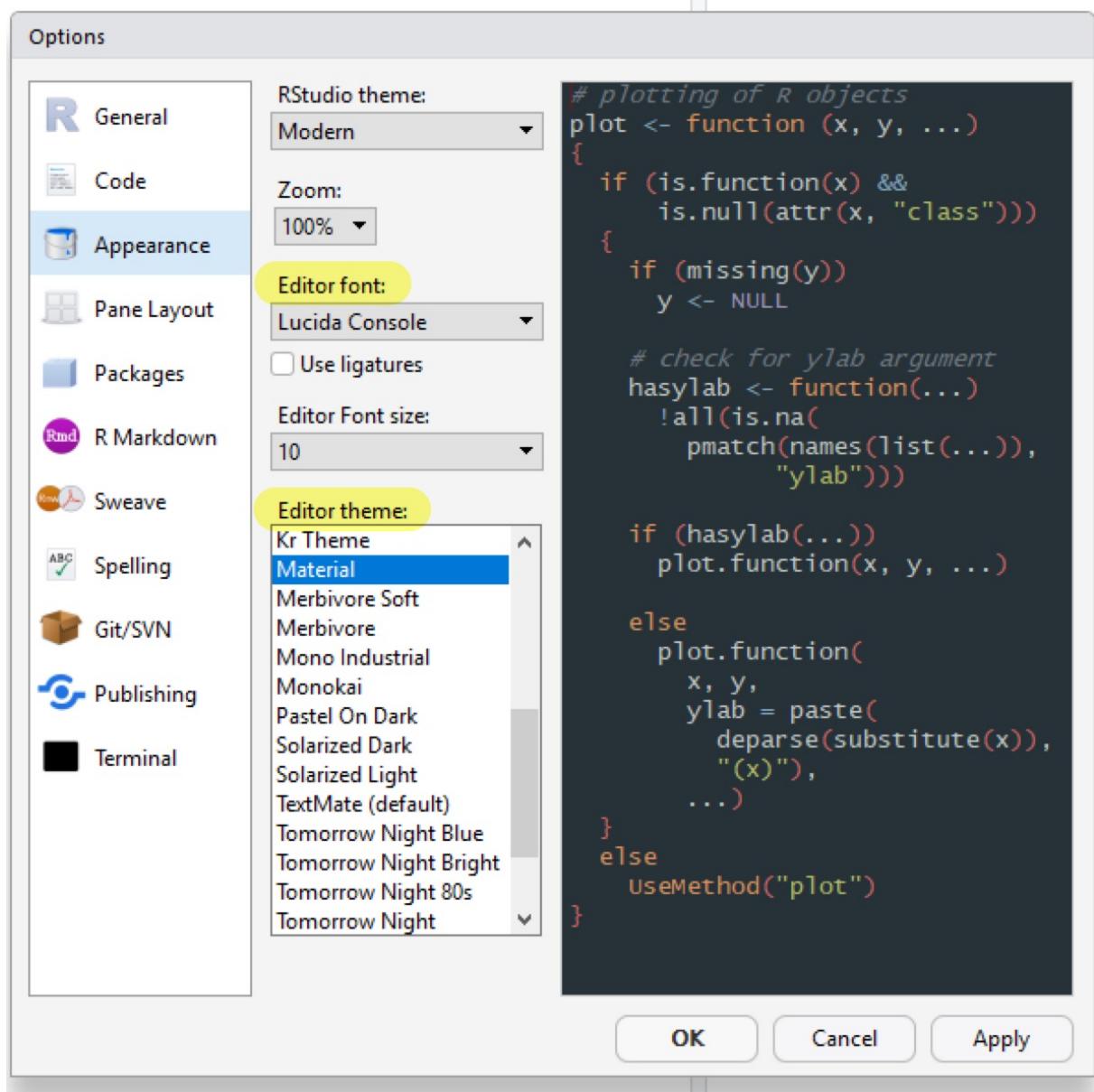
2.2.3 RStudio の起動

RStudio のショートカットをクリックしたり、メニューで **RStudio** と入力してクリックすると起動するはずです。

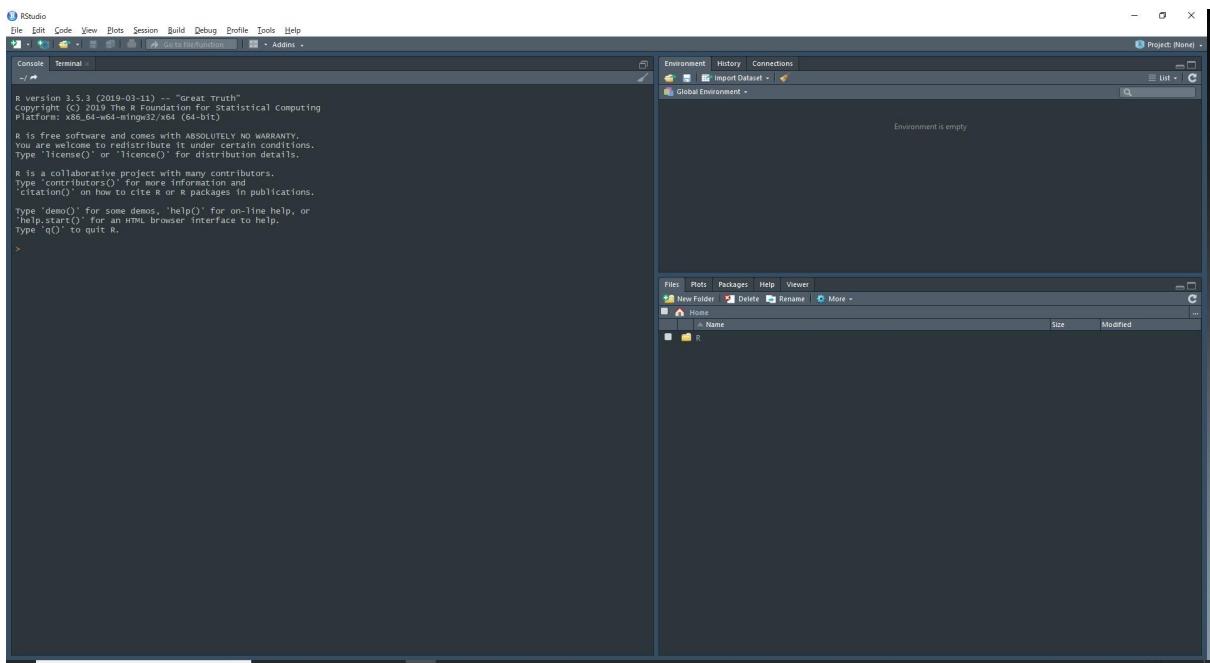
RStudio を初めて起動すると次のような表示になるはずです。左側の大きなパネルで R が表示されていればインストールの成功です。



ちなみに、Tools > Global Options > Appearanceではフォントや背景・ハイライトの色を変えることができます。



ダークな背景を選択すると RStudio 全体もダークテーマになります。



2.2.4 RStudio Cloud*

RStudio Cloudにより、RStudio をブラウザを使ってオンラインで使用することができます。複数のユーザーで共同作業を行うことも可能です。

- Linux ユーザーは RStudio Server を使って自らサーバを立てることもできます。

2.3 再現可能な分析のために

再現可能性 (replicability) とは、狭義では、誰がどんな環境で分析しても。オリジナルの分析結果と（ほぼ）同じものを得られることだと思っています。以下では、再現可能性を担保できるような R/RStudio の使い方を解説します。

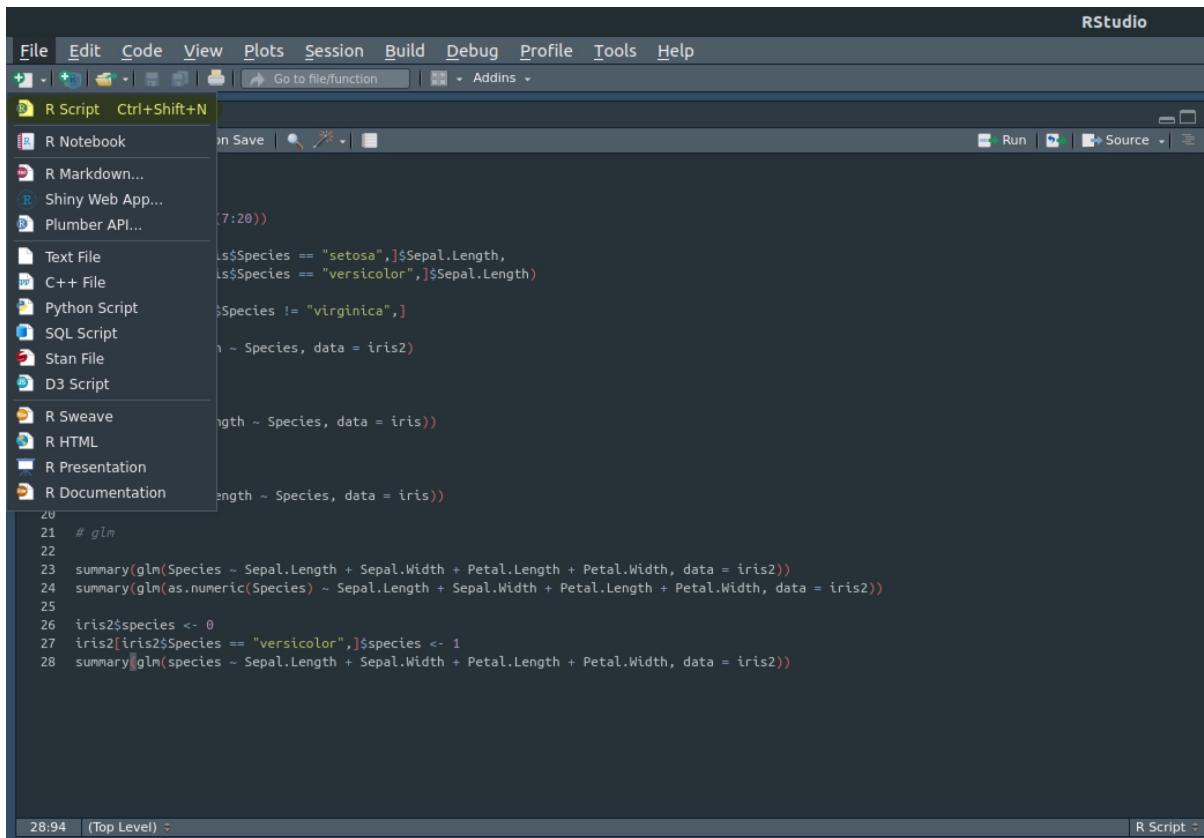
2.3.1 R スクリプト

まず、分析の手順を記録に残し、公開する必要があります。R では R スクリプトと呼ばれるファイル（拡張子は .R）を作成し、そこにコードを残して起きます。

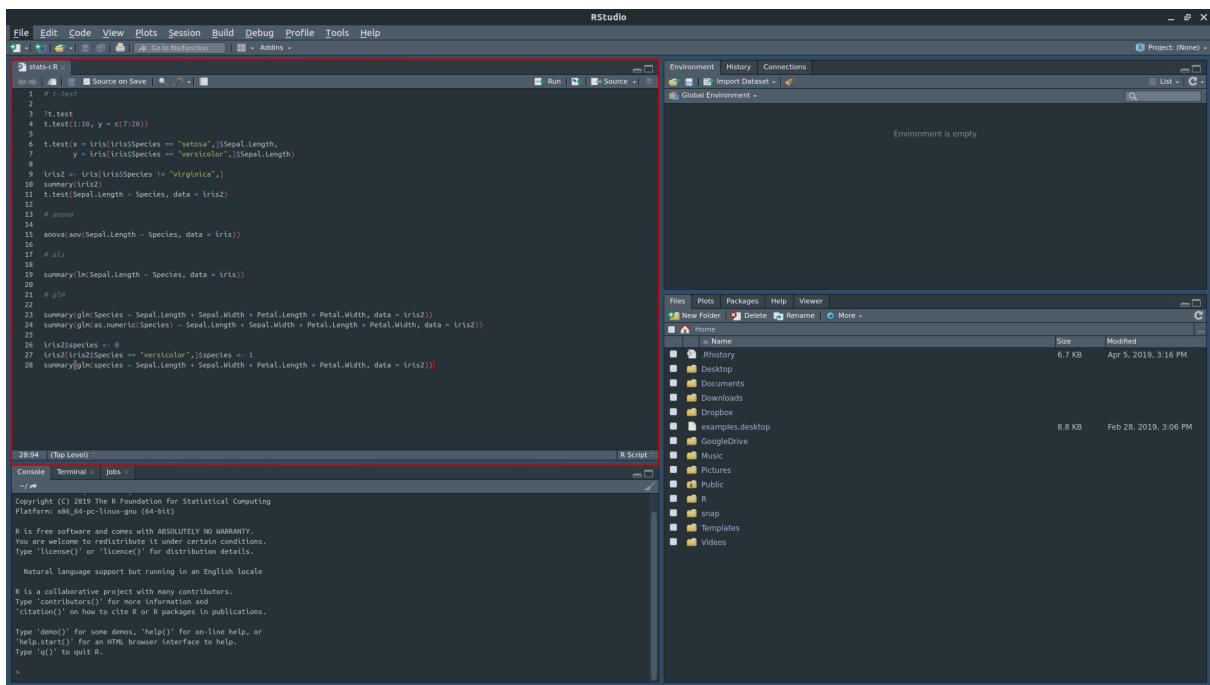
- もちろん、使用したデータも公開する必要があるのは言うまでもありません。

■2.3.1.0.1 R スクリプトの作成

RStudio では左上の File > New File > RScript もしくは白い紙に緑色のプラスマークのボタンを押して R Script を選択します。



すると、デフォルトでは左上のパネルに R スクリプトのエディタが表示されます。



■2.3.1.0.2 Rスクリプトの展開

RスクリプトをRStudioで開くには左上のFile > Open Fileで選択します。

■2.3.1.0.3 Rスクリプトの実行

Rスクリプトに書かれたコードはCtrl + Enterを押すと、カーソルのある行がコンソールに流れ、実行されます。

2.3.2 Rプロジェクト

データの読み込みで解説したように、データの読み込みや保存の際には起点となる作業ディレクトリ (working directory) を決める必要があります。

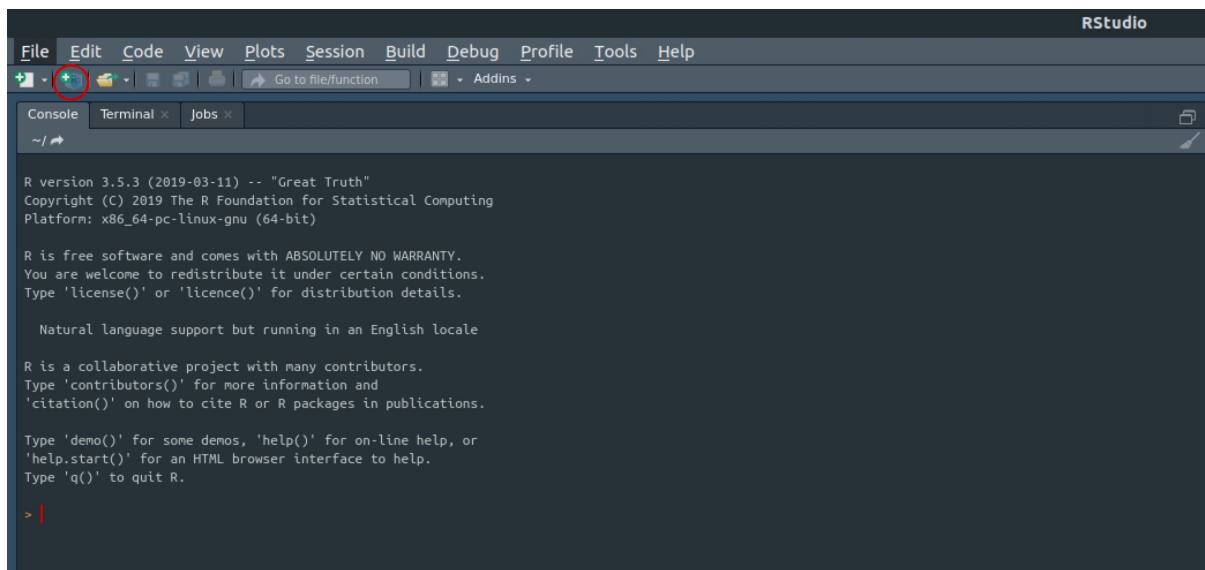
一般的に、作業ディレクトリはPCによって変わってしまうので、Rプロジェクトを立てることでその問題を回避します。簡単に言えば、Rプロジェクトをクリックすることで自動的に作業ディレクトリが設定された状態でRStudioを起動することができます。

また、プロジェクトごとにRStudioを起動できるので、異なるプロジェクト間でデータやRスクリプトが混在することも回避できます。

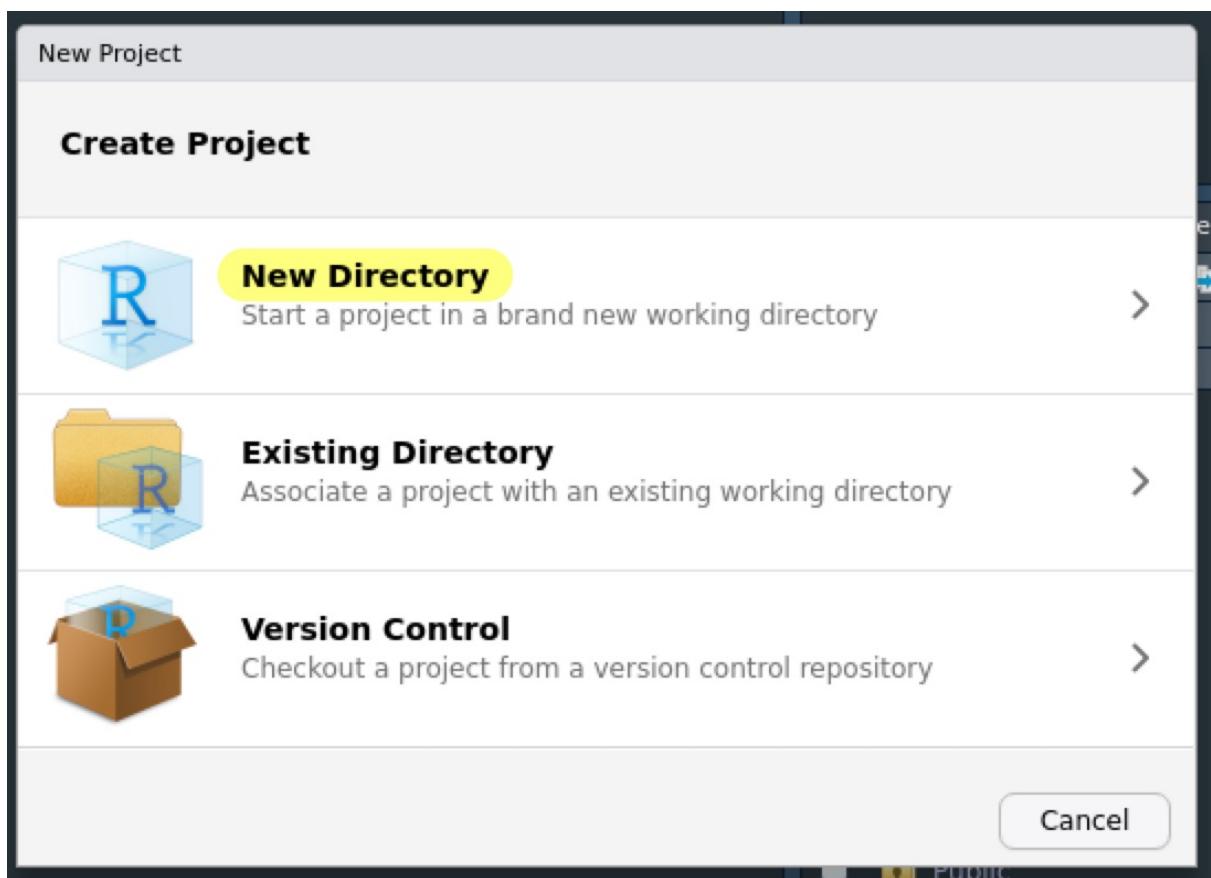
ひとまず、新しい分析を行う際は必ずRプロジェクトを作成するようにしましょう。

■2.3.2.0.1 R プロジェクトの作成

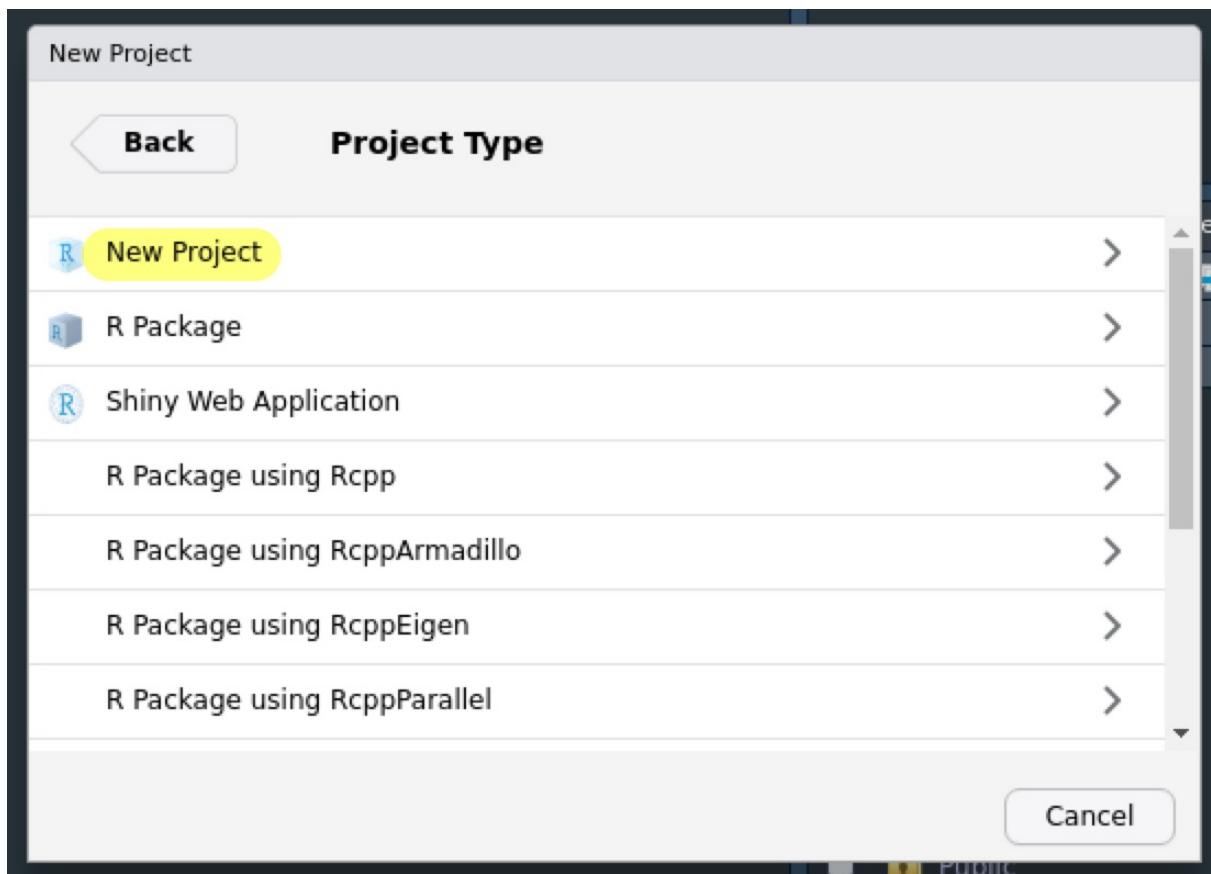
まずは、プロジェクトの作り方ですが、RStudio の左上の青いボタンをクリックします。



続いて、新たにプロジェクト用のフォルダを作るのであれば **New Directory** を、既存のフォルダをプロジェクト用にするのであれば **Existing Directory** を選択します。

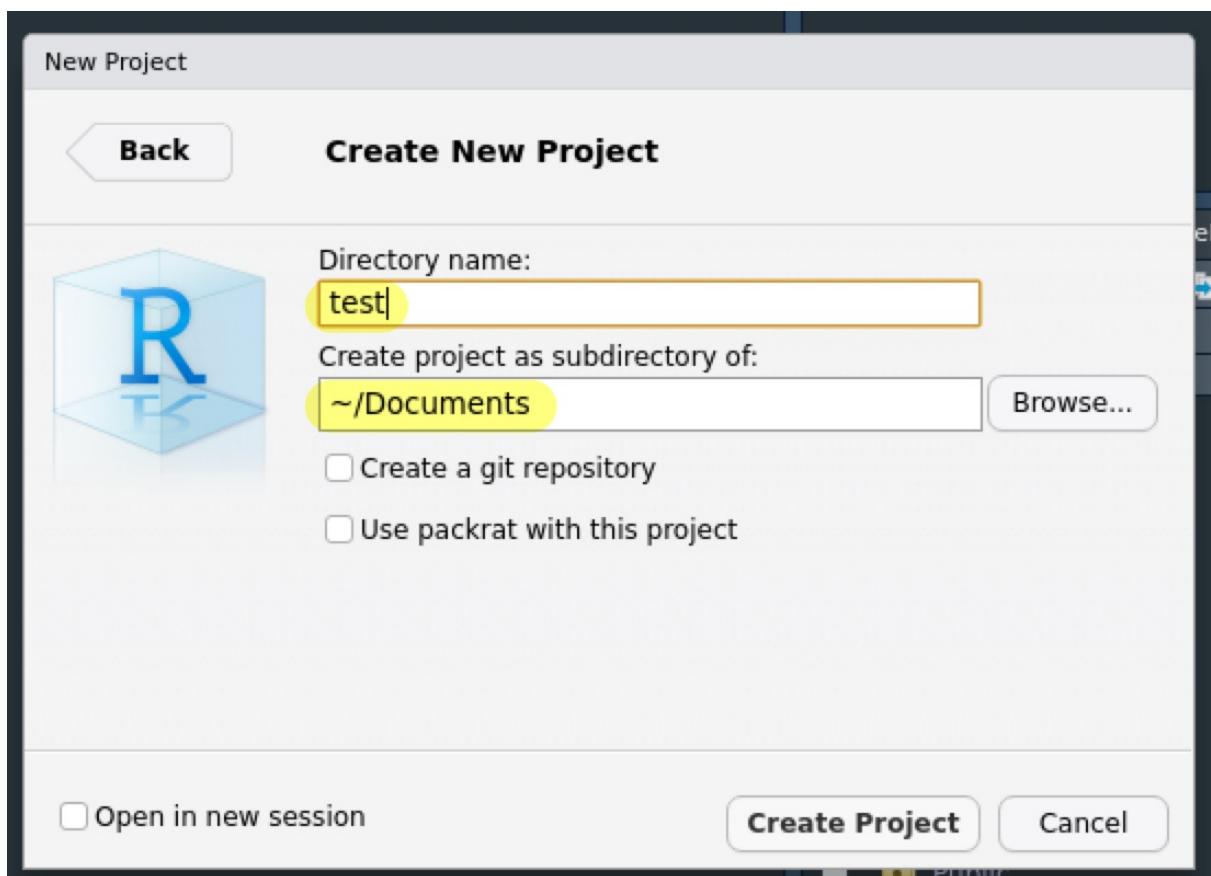


基本的には New Project を選択します。



最後に、プロジェクト用のフォルダの名前とそのフォルダを置くフォルダのパスを指定して **Create Project** をクリックします。

- ・ フォルダ名は必ず英数字と-や_で書き、日本語は避けましょう。
- ・ 既存のフォルダを使う場合はパスを指定するだけです。



- ・例えば今回は Documents フォルダの中に test という名前のプロジェクトを作成しました。

一度、RStudio を終了し、先程指定したパス通りの場所にフォルダができていることを確認してください。そのフォルダの中に、プロジェクト名と同じ名前の.Rproj ファイルができているはずです。

■2.3.2.0.2 プロジェクトの起動

それをダブルクリックしてみると RStudio が起動されます。このとき、すでに作業ディレクトリはプロジェクト用フォルダに指定されているのです。

- ・getwd() で作業ディレクトリを確認してみて下さい。

■2.3.2.0.3 ワークスペースの保存と再開 *

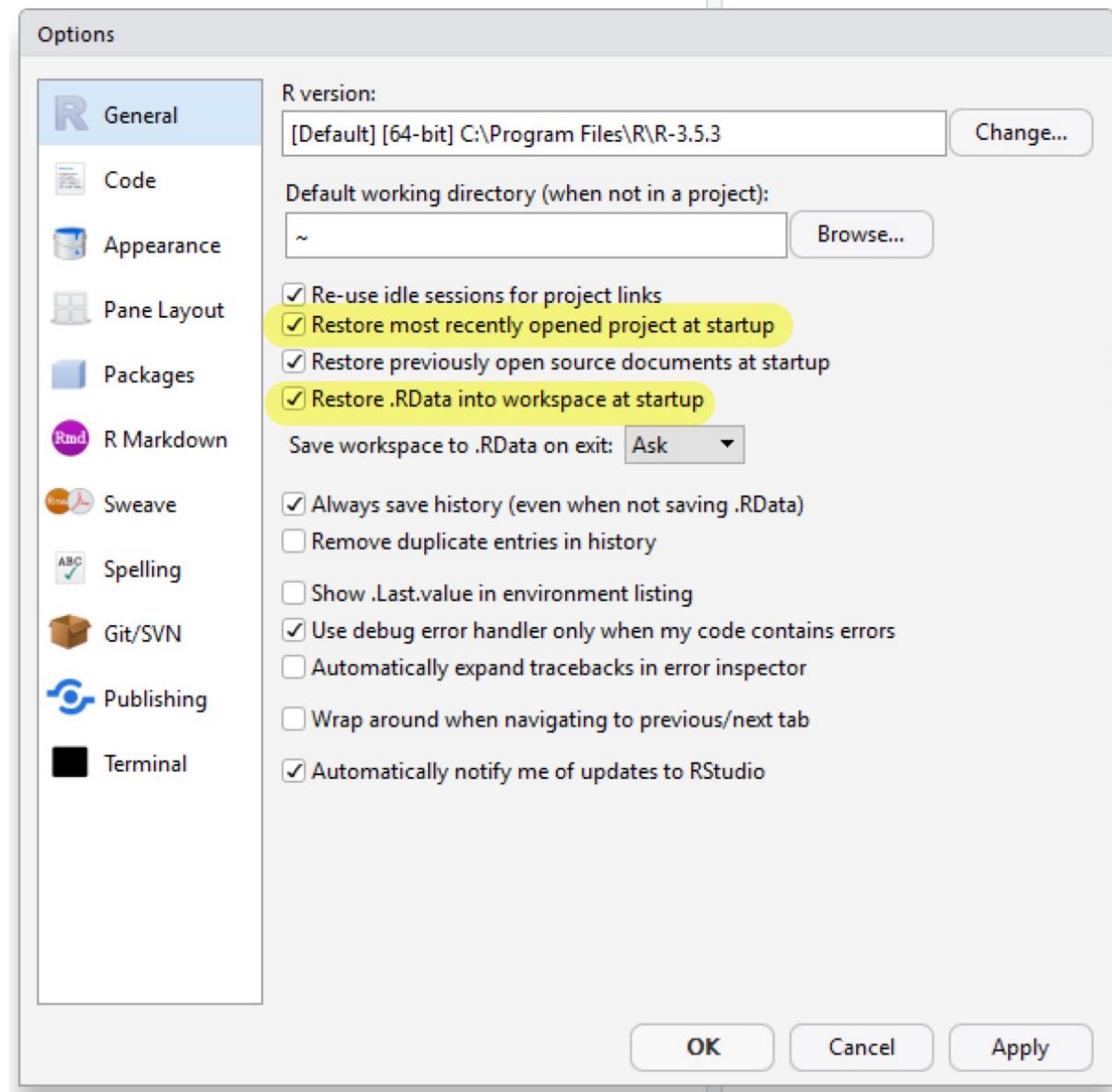
どうしても一度分析を中断して、再開したい場合はワークスペースを保存しておきましょう。上記画面で Save workflow to .RData on exit が Ask になっている場合、RStudio を終了する際にワークスペースを保存するのか聞かれるはずなので、保存します。

- ちなみに、.RData ファイルは R のワークスペース（の一部）を保存するデータ形式です。
- すると、フォルダ内に.RData ファイルができるので、再開するときに `load()` に当該ファイルのパスを入力して実行するとワークスペースが復元されます。

2.3.3 RStudio の設定 *

■2.3.3.0.1 RStudio 起動時の挙動

Tools > Global Options を開き、General の中で以下のチェックを外します。



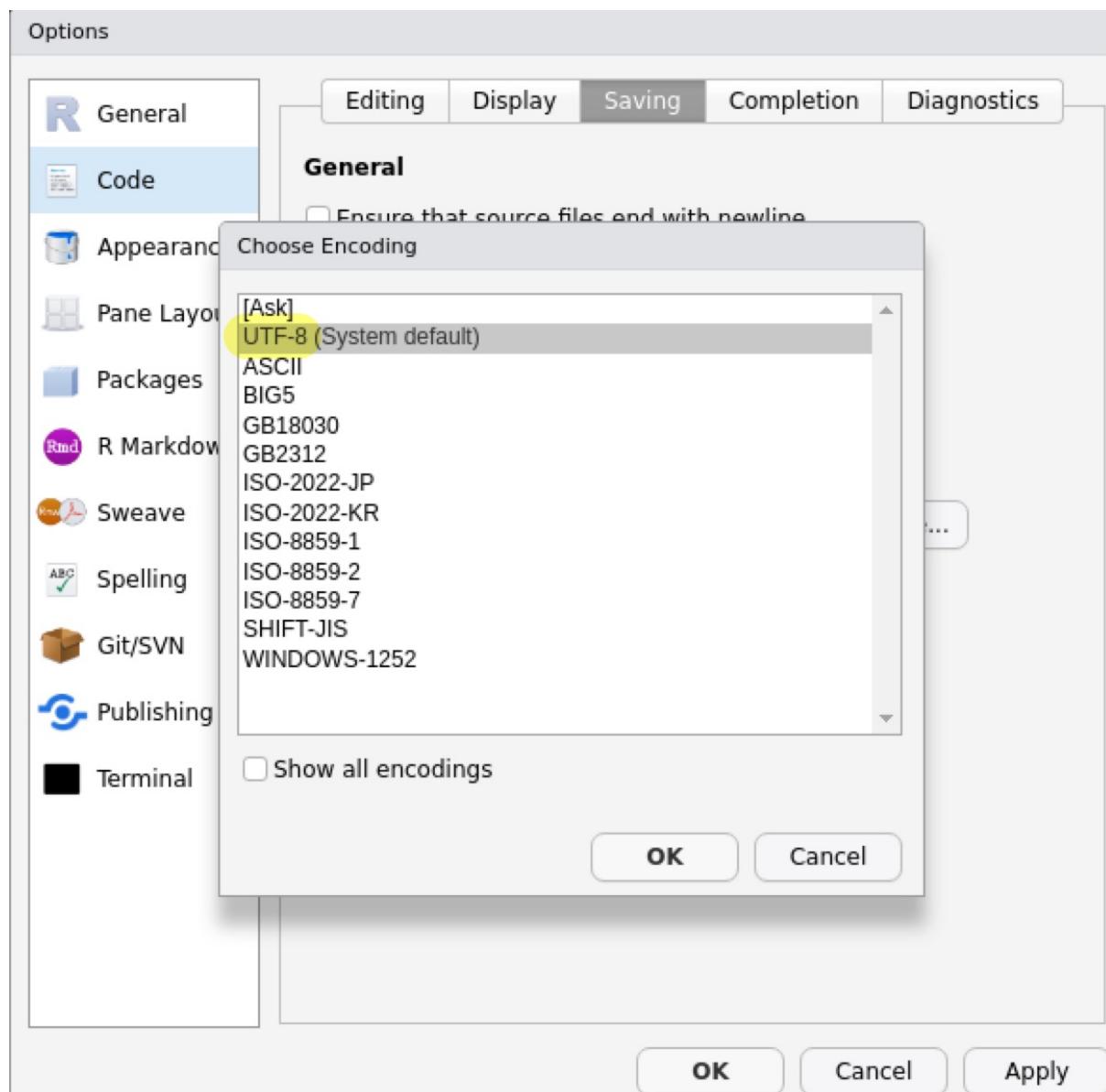
平たく言うと RStudio を起動したときに前回の続きを残っていない真っさらな状態にしておきます。

■2.3.3.0.2 文字コード

日本語がしばしば文字化けすることがあります。なぜなら Windows では Shift-JIS、Linux と Mac では UTF-8 と呼ばれるエンコーディング（平たく言うと PC が文字を表示する方法）形式だからです。

- 詳しくは R における文字コードを参照して下さい。

UTF-8 が世界的に使われているので、Code > Saving > Default text encoding を UTF-8 にしておきます。



もし、日本語を含むファイルを RStudio で開いたときに文字化けしている場合、Windows を使っている人は UTF-8 のファイルを Shift-JIS で開いたということなので、File > Reopen with Encoding で UTF-8 を選択します。

逆に Mac の場合は Shift-JIS のファイルを UTF-8 で開いているので同様に Shift-JIS で開きます。

- Windows の人は UTF-8 をデフォルトのエンコーディングにしてしまうといいでしょう。

第3章

R プログラミング入門

Rによるプログラミングの基本として、

- ・オブジェクト
- ・関数
- ・パッケージ

について解説します。

大雑把に言えば、Rではオブジェクトとしてデータを読み込み、関数によってオブジェクト（＝データ）の処理や分析を行います。パッケージによって様々な関数を追加することで、処理や分析の幅を広げます。

RStudioでは左（下）にコンソールが表示され、>の右側にコマンドを打ち込み、Enterを押すことで実行されます。

- ・本格的に分析する場合はRスクリプトを作成します。

3.1 関数

関数(function)とは何かを入力すると、何かを出力するものです。例えば、

```
print("Hello, World.")
```

```
## [1] "Hello, World."
```

というコードは、"Hello, World."という文字列をprint()という関数に入力し、その文字列を出力しています。

- ・Rでは、関数は関数名()という形を取ります。
- ・入力するものを入力引数(input argument)、出力するものを出力引数(output argument)と呼んだりします。

次のように、入力引数も出力引数も1つとは限りません。

```
rnorm(n = 10, mean = 0, sd = 1)
```

```
## [1] -1.42452621 -0.42007188 -0.11354578  0.30189294 -0.29613240  0.37013165
## [7]  0.02458299  0.95447871 -2.49405815 -0.35534634
```

さて、この関数は何をしているのでしょうか。R では、関数名の前に? をつけて実行することで、その関数のヘルプを見ることができます。

```
?rnorm
```

英語で関数の使い方が解説されていますが、`rnorm(n = 10, mean = 0, sd = 1)` は平均 0、標準偏差 1 の（標準）正規分布に従う乱数を 10 個だけ生じさせています。

入力引数は = で明示的に指定する場合、どのような順番でも構いません。

```
rnorm(mean = 0, sd = 1, n = 10)
```

入力引数を明示的に指定しない場合、ヘルプにある順番で入力します。以下の例は上述のものと同じです。

```
rnorm(10, 0, 1)
```

また、ヘルプで `mean = 0, sd = 1` のように書かれている場合、デフォルトが定められています。実行者が入力引数を指定しない限り、デフォルト値が使用されます。したがって、以下の例もこれまでと同じコードです。

```
rnorm(10)
```

3.1.1 総称関数 *

総称関数 (generic function) とは、Rにおいて入力引数の種類に応じて挙動が変わる関数のことを指します。例えば、`summary()` という関数はデータフレームが入力引数の場合には記述統計を表示しますが、回帰分析の結果の場合は回帰表を出力します。

総称関数のヘルプを見る場合は、以下のように、関数名に. をつけて入力引数の種類を書きます。

```
?summary.data.frame
```

```
?summary.lm
```

3.2 オブジェクト

R では <-でオブジェクトを作成することができます。例えば、20 個の正規分布に従う乱数を `x` という名前のオブジェクトとして作成します。

```
x <- rnorm(20)
```

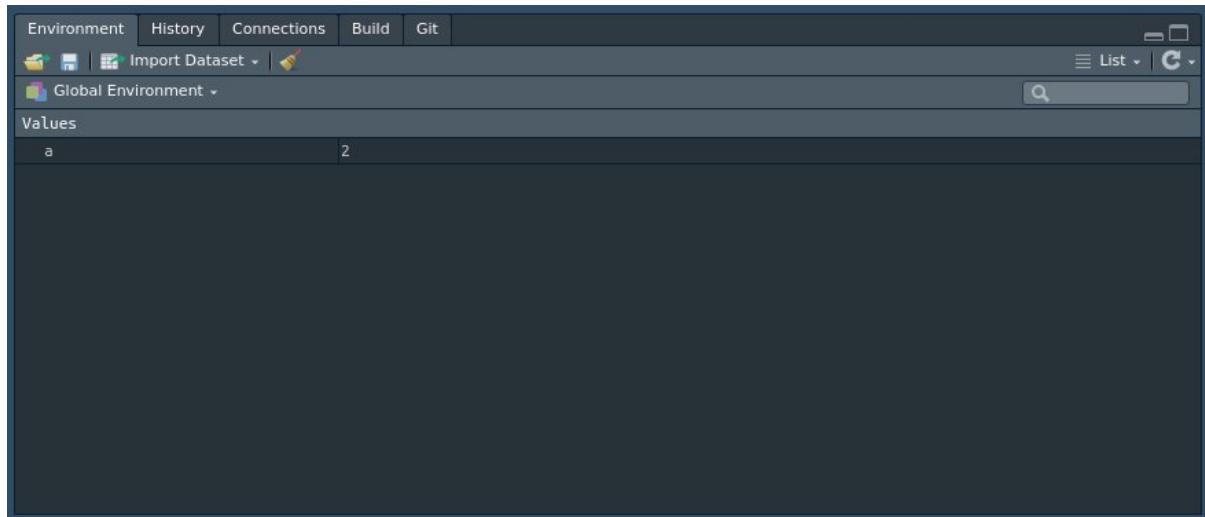
- RStudio では <-はショートカット Alt + -で入力できます。

実際に、乱数が `x` に格納されていることが分かります。

```
x
```

```
## [1] -1.1242721 1.3335992 1.9714474 0.6266777 1.8654255 1.8993201
## [7] 0.8408928 0.8390228 -0.2873750 0.5132134 -1.0851594 0.5637831
## [13] -0.3414139 -0.6298208 0.6594867 -0.6565816 -1.0344331 -0.2830303
## [19] -0.7212691 0.4840311
```

RStudio の場合、右上の Environment パネルに生成されたオブジェクトが表示されます。



オブジェクトを入力引数とすることも可能です。`x` の平均と標準偏差を求めてみます。

```
mean(x)
```

```
## [1] 0.2716772
```

```
sd(x)
```

```
## [1] 1.012273
```

もちろん、出力引数を新しいオブジェクトにすることもできます。

```
x.mean <- mean(x)
```

```
x.mean
```

```
## [1] 0.2716772
```

- オブジェクトの名前にはアルファベットと数字、. と _ が使えます。
- ただし、数字は最初の文字としては使えません。

オブジェクトは上書きすることもできます。

```
x.mean <- mean(rnorm(20))
x.mean

## [1] 0.428162
```

- ・先ほどとは違う値に上書きされていることが分かります。

3.3 パッケージ

大雑把に言って、Rによるデータ分析はデータをオブジェクトとして読み込み、いろいろな関数で処理を行うことで実行します。

つまり、関数が重要なのですが、Rで標準に備わっている関数には限界があります。そこで、様々な研究者が関数を作成し、それをまとめたものをパッケージとして公開しています。

- ・基本的に、CRANでパッケージは公開されます。
- ・ライブラリやモジュールと呼んだりすることもあります。

3.3.1 CRANからのインストール

パッケージをインストールするには、`install.packages()`という関数にパッケージ名を入れて実行します。試しに、Tidyverseという幅広く使われているパッケージをインストールしてみます。

```
install.packages("tidyverse")
```

”でパッケージ名を囲まないとエラーになります。

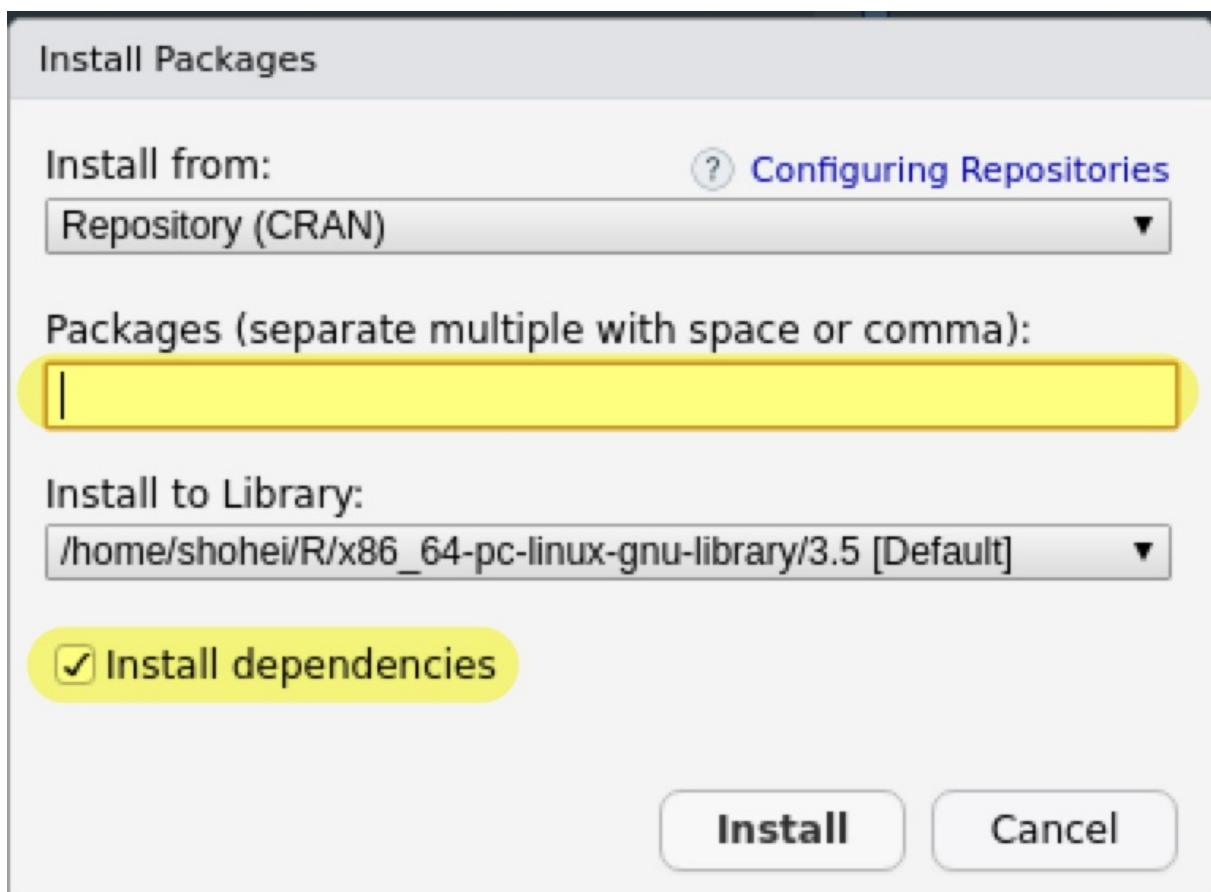
```
install.packages(tidyverse)
```

```
## Error in install.packages(tidyverse): object 'tidyverse' not found
```

RStudioの場合、Packagesパネル（デフォルトの場合は右下）の中にInstallというボタンがあり、

Name	Description	Version		
User Library				
abind	Combine Multidimensional Arrays	1.4-5	● ●	
antiword	Extract Text from Microsoft Word Documents	1.3	● ●	
askpass	Safe Password Entry for R, Git, and SSH	1.1	● ●	
assertthat	Easy Pre and Post Assertions	0.2.1	● ●	
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.3	● ●	
base64enc	Tools for base64 encoding	0.1-3	● ●	
bayesplot	Plotting for Bayesian Models	1.6.0	● ●	
bdsmatrix	Routines for Block Diagonal Symmetric Matrices	1.3-3	● ●	
BH	Boost C++ Header Files	1.69.0-1	● ●	
bibtex	Bibtex Parser	0.4.2	● ●	
bitops	Bitwise Operations	1.0-6	● ●	
bookdown	Authoring Books and Technical Documents with R Markdown	0.9	● ●	
bridgesampling	Bridge Sampling for Marginal Likelihoods and Bayes Factors	0.6-0	● ●	
brms	Bayesian Regression Models using 'Stan'	2.8.0	● ●	
Brodbdingnag	Very Large Numbers in R	1.2-6	● ●	
broom	Convert Statistical Analysis Objects into Tidy Tibbles	0.5.1	● ●	
callr	Call R from R	3.2.0	● ●	
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0	● ●	
checkmate	Fast and Versatile Argument Checks	1.9.1	● ●	
cli	Helpers for Developing Command Line Interfaces	1.1.0	● ●	
clipr	Read and Write from the System Clipboard	0.5.0	● ●	
clisymbols	Unicode Symbols at the R Prompt	1.2.0	● ●	
clubSandwich	Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections	0.3.3	● ●	
coda	Output Analysis and Diagnostics for MCMC	0.19-2	● ●	
coefplot	Plots Coefficients from Fitted Models	1.2.6	● ●	
colorspace	A Toolbox for Manipulating and Assessing Colors and Palettes	1.4-1	● ●	

そこにパッケージ名を入力してインストールすることも可能です。



インストールしたパッケージに対して再び `install.packages()` を行うと、最新版にアップデートされます。

- RStudio の場合、Packages パネルに **Update** というボタンがあり、アップデートできるパッケージを自動検索してくれます。

3.3.2 GitHub からのインストール *

パッケージの開発版や一部のパッケージは GitHub 上で公開されています。

GitHub 上のパッケージをインストールする場合は `devtools` というパッケージを使うので、まずはインストールと読み込みを行います。

```
install.packages("devtools")
library(devtools)
```

インストールには `install_github()` を使いますが、入力はパッケージ名ではなくユーザー名/レポジトリ名となる点に注意してください。

3.3.3 パッケージの読み込み

パッケージはインストールしただけでは使用することはできず、`library()`で読み込む必要があります。

```
library(tidyverse)
```

- この場合は”で囲む必要はありません。
- インストールは一回で十分です。

RStudio であれば `Packages` パネルにインストール済みのパッケージ一覧があるので、パッケージ名をクリックすると含まれる関数一覧を見ることができます。

- 同様のものはCRANでも `pdf` 形式で見ることができます。
- 一部のソフトウェアはJournal of Statistical Softwareなどで論文が公開されています。

3.3.4 tidyverse とは *

Tidyverse とは広義には R におけるデータ処理を行うためのパッケージを開発するプロジェクトであり、狭義にはそこで開発されたパッケージの一部を指します。具体的には、

- `ggplot2`
- `dplyr`
- `tidyverse`
- `readr`
- `purrr`
- `tibble`
- `stringr`
- `forcats`

になります。

パッケージとしての `tidyverse` を読み込むことで、上記のパッケージを読み込んでいます。

なお、プロジェクト全体としては、上記のもの以外にも多くのパッケージが開発されています。

第4章

R プログラミング応用 *

Rによる、より高度な作業のために

- ・代表的なオブジェクトのクラス
- ・オリジナルの関数の作成
- ・ループや条件分岐

などを学びます。

4.1 オブジェクトのクラス

オブジェクトの種類をクラスと呼びます。`class()` にオブジェクトを入力するとクラスが分かります。

最も使われるは数値 (`numeric`, `real`) です。

```
class(1)
```

```
## [1] "numeric"
```

厳密には数値と整数 (`integer`) は異なりますが、気にしないといけない局面は少ないと私は思います。

```
class(2L)
```

```
## [1] "integer"
```

他には、文字列 (`character`) や

```
class("Hello, World.")
```

```
## [1] "character"
```

論理値 (`logical`) などもあります。

```
class(TRUE)
```

```
## [1] "logical"
```

論理値は主に条件式が満たされるかどうかを示します。

```
1 == 1
```

```
## [1] TRUE
```

```
0 > 2
```

```
## [1] FALSE
```

ちなみに、TRUE は数値としての 1、FALSE は 0 にもなります。

```
TRUE + 1
```

```
## [1] 2
```

```
FALSE * 2
```

```
## [1] 0
```

また、因子型 (factor) と呼ばれるクラスもあります。カテゴリカル変数と言ったほうが分かりやすいかもしれません。

```
x <- factor(1)
```

```
x
```

```
## [1] 1
```

```
## Levels: 1
```

```
class(x)
```

```
## [1] "factor"
```

x の中身は 1 ですが、数値ではなくカテゴリーになっているので、数値として操作することはできません。

```
x + 1
```

```
## Warning in Ops.factor(x, 1): '+' not meaningful for factors
```

```
## [1] NA
```

R ではベクトルには特別なクラスは付与されていません。ベクトルは c() に中身を入力して作成します。

```
x <- c(1,3,5)
```

```
x
```

```
## [1] 1 3 5
```

行列 (matrix) はクラスとして存在します。

```
x <- matrix(c(1,3,5,7), 2, 2)
```

```
x
```

```
##      [,1] [,2]
```

```
## [1,]    1    5
```

```
## [2,]    3    7
```

```
class(x)
```

```
## [1] "matrix"
```

他に、データフレーム (data.frame) やリスト (list) などもあります。

クラスを確認するときは、`is.*()` の形をとる関数を使います。

```
is.numeric(1)
```

```
## [1] TRUE
```

```
is.character(1)
```

```
## [1] FALSE
```

クラスを変更するときは、`as.*()` のような関数を使います。

```
as.factor(1)
```

```
## [1] 1
```

```
## Levels: 1
```

```
as.character(1)
```

```
## [1] "1"
```

- 必ずしも全てのクラスが任意のクラスに変換できるわけではありません（例えば、文字列から数値など）。

4.2 関数の作成

Rで関数を自作する際は `function(){}` という関数を使います。

- `()` 中に入力引数を記述します。
- `{}` 中に処理内容を記述し、最後に `return()` で出力引数を指定します。

例えば、数値ベクトルを入力引数として、平均と標準偏差を出力引数とする関数を作成します。

```

mean_sd <- function(x) { # 入力引数の名前を x としておきます。
  mean.x <- mean(x) # 平均を計算します。
  sd.x <- sd(x) # 標準偏差を計算します。
  return(c(mean.x, sd.x)) # 出力引数を指定します。
}

```

実際に実行してみます。

```

x <- rnorm(100)
mean_sd(x)

## [1] -0.04230494 1.00539856

```

4.3 ループ

ループとは同一の処理を複数回実行することを指します。例えば、100 個の標準正規分布に従う乱数の平均を 5 回求める処理は次のようになります。

```

for (i in 1:5) {
  print(mean(rnorm(100)))
}

```

```

## [1] 0.1260475
## [1] -0.01005321
## [1] 0.0575182
## [1] -0.01412347
## [1] -0.06033713

```

for ループとは () の中の in のあとのあるベクトルの第 1 要素から順番に i に代入して繰り返しています。そのことは、次の例から解ると思います。

```
head(letters)
```

```

## [1] "a" "b" "c" "d" "e" "f"

```

- letters とはアルファベットのベクトルです。

```

for (i in head(letters)) {
  print(i)
}

```

```

## [1] "a"
## [1] "b"
## [1] "c"

```

```
## [1] "d"  
## [1] "e"  
## [1] "f"
```

- `for` ループとは別に、特定の条件が満たされるまで繰り返される `while` ループもあります。

ループ処理の結果を格納するには少しテクニックが必要です。100 個の乱数の平均を 5 回取ったものを `x` として保存したいとします。

まず、`x` を `NULL` オブジェクトとして作成します。

```
x <- NULL  
x
```

```
## NULL
```

- `NULL` とは空っぽのオブジェクト（0 という数値や空白という文字ではない）です。

先程のループ処理の中で、計算した平均を `c()` で `x` にくっつけていきます。

```
for (i in 1:5) {  
  x <- c(x, mean(rnorm(100)))  
}  
x
```

```
## [1] -0.01254183 -0.01507656 -0.04814641  0.08880067  0.05525210
```

無事、5 個の平均値が `x` に保存されていることがわかります。

実際に `for` ループの中で何が起こっているかは、次のコードで解ると思います。

```
x <- NULL  
for (i in 1:5) {  
  x <- c(x, mean(rnorm(100)))  
  print(x)  
}  
  
## [1] -0.08562251  
## [1] -0.085622505  0.005231885  
## [1] -0.085622505  0.005231885 -0.121550919  
## [1] -0.085622505  0.005231885 -0.121550919  0.054285805  
## [1] -0.085622505  0.005231885 -0.121550919  0.054285805  0.074869545
```

- ループが一周するたびに、前回の `x` に新しい要素が付け加わり、新しい `x` として保存されています。

`NULL` オブジェクトを使ったループ結果の保存でよくあるミスは、やり直す際に `NULL` でリセットするのを忘れることがあります。例えば、同じコードをもう一度実行しましょう。

```

for (i in 1:5) {
  x <- c(x, mean(rnorm(100)))
}
x

## [1] -0.085622505  0.005231885 -0.121550919  0.054285805  0.074869545
## [6]  0.016440166  0.108694760 -0.003228409  0.030114880  0.133772951

```

- `x` に 10 個の平均値が入っています。

このようなミスを避ける方法の一つは、全体を関数として作成することです。

```

multi_mean <- function() {
  x <- NULL
  for (i in 1:5) {
    x <- c(x, mean(rnorm(100)))
  }
  return(x)
}
x <- multi_mean()
x

## [1]  0.03679536 -0.02348755  0.01728398  0.13719900 -0.01580726

```

4.4 条件分岐

条件分岐とは、特定の条件の場合に特定の動作を行うようにすることです。例えば、正の場合 `positive`、負の場合 `negative` と出力するコマンドは次のようにになります。

```

x <- rnorm(1)
if (x > 0) {
  print("positive")
} else {
  print("negative")
}

## [1] "positive"
print(x)

## [1] 1.005329

```

- `if(){}` の `()` の中に条件式を書き、`{}` の中に処理内容を書きます。
- それ以外の条件は `else` で示します。

条件式は3つ以上でも構いません。

```
x <- rnorm(1)
if (x > -0.5) {
  print("x is less than -0.5.")
} else if (x >= -0.5 & x <= 0.5) {
  print("x is between -0.5 and 0.5.")
} else {
  print("x is more than 0.5. —")
}
```

```
## [1] "x is less than -0.5."
```

```
print(x)
```

```
## [1] 0.1254923
```

- `&` は「かつ」を意味します。
- 「または」は `|` を使います。
- `>=` は \geq を意味します。
- 「同じ値である」は `==` を使います ($=$ ではない点に注意)。

4.5 練習問題

4.5.1 フィボナッチ数列

フィボナッチ数列とは以下の条件を満たす数列です。

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \quad n \geq 2 \end{aligned}$$

例えば、

$$F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8, \dots$$

となります。

フィボナッチ数列の第 n 項を（解析解を使わずに）求める関数を作成してみて下さい。

また、 $F_n \geq m$ となるような n を求める関数を作成してみて下さい。

4.5.2 モンテカルロ・シミュレーション

モンテカルロ・シミュレーション（モンテカルロ法）とは乱数を用いて近似解を求める手法です。

例えば、円周率 π の近似解は以下のように求めることができます。

1. 0 以上 1 未満の一様分布から n 個の乱数 x_i と n 個の乱数 y_i を発生させます ($i = 1, 2, \dots, n$)。
2. 原点と (x_i, y_i) の距離が 1 以下である回数を計算し n_1 とします。
3. 円周率の近似解として $\hat{\pi} = 4 \times n_1/n$ を得ます。

モンテカルロ・シミュレーションによる円周率の近似解を求める関数を作成してみて下さい。

また、モンテカルロ・シミュレーションによる円周率の近似解を m 回求めて、その平均値や標準偏差が n によってどのように変化するか検討してみて下さい。

第5章

R Markdown 入門

データ分析の再現可能性の必要性は論を俟たないですが、再現可能性を担保するにはレプリケーションデータやコードを公開するだけでなく、それらが理解可能である必要があります。

- 恐ろしいことに自分が書いたコードでさえ数ヵ月後に読み返すと意味がわからないことはまれによくあります。

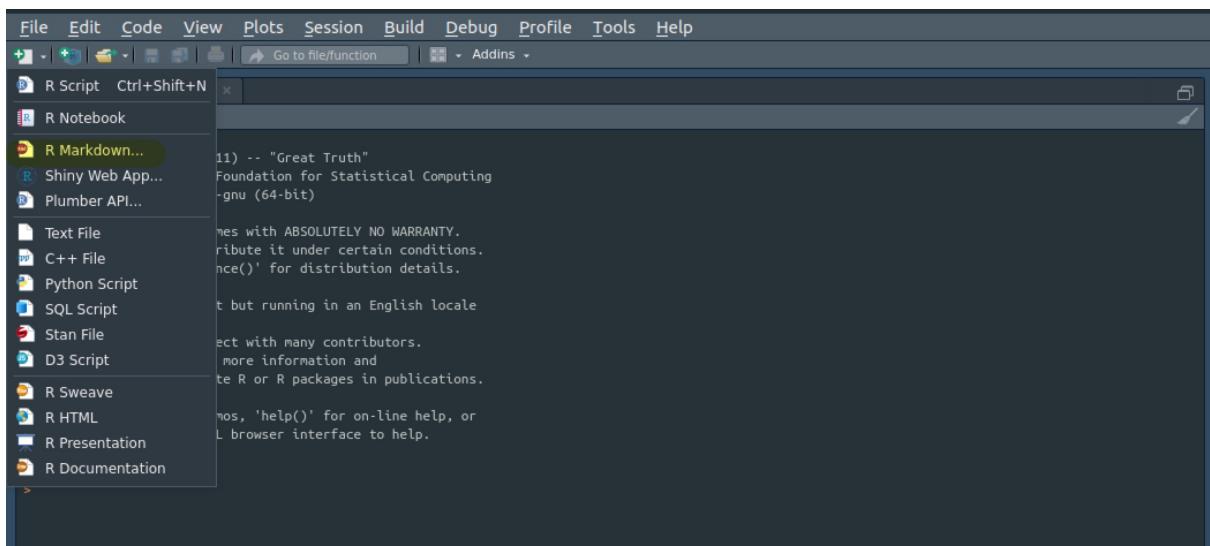
Rスクリプトに#でコメントするのが単純な方法ですが、データ分析においてはしばしば文章とコード、アウトプットを混在させたノートブックを使用することができます。

- 詳しくはないですが文芸的プログラミングと呼ばれるものの一種のような気がします。

更にノートブックからより見やすいファイルを作成することができ、そのファイルおよびシステムをR Markdownと呼びます。実は、と言うほどではないですが、このブログもR Markdownで書かれています。

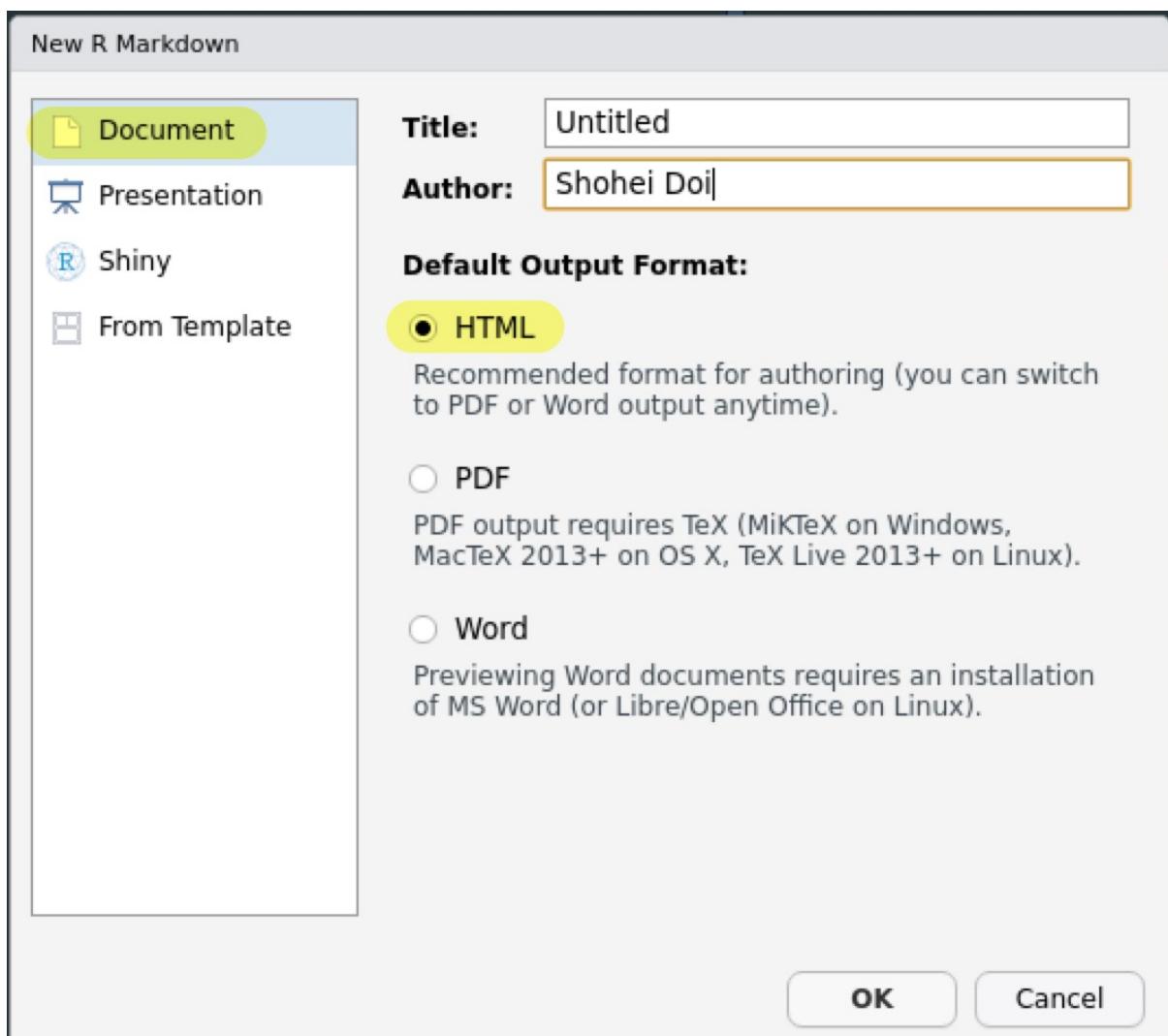
5.1 R Markdown ファイルの作成

百聞は一見に如かずなので、まずはR Markdownを使ってみます。まず、左上のファイルを作成するボタンを押し、R Markdown...を選択します。



- 初めて R Markdown を使う場合は必要なパッケージをインストールするか聞かれるのでインストールを選択します。

続いて、どのような種類の R Markdown ファイルを作成するかを選択するので、(デフォルトのままで) Document の HTML を選択します。



すると、エディタに以下のようなサンプルの `Rmd` ファイルが表示されます。適当なフォルダに保存し、エディタ上部の `Knit` をクリックするか `Shift + Ctrl + k` を押すと R Markdown ファイルがタイプセットされます。

```
File Edit Code View Plots Session Build Debug Profile Tools Help
+ Go to file/function | Addins
sample.Rmd x Knit Insert Run
1 v ...
2 title: "Untitled"
3 author: "Shohei Doi"
4 date: "4/9/2019"
5 output: html_document
6 ---
7
8 v ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ...
11
12 v ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
15
16 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
17
18 v ```{r cars}
19 summary(cars)
20 ...
21
22 v ## Including Plots
23
24 You can also embed plots, for example:
25
26 v ```{r pressure, echo=FALSE}
27 plot(pressure)
28 ...
29
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
31
```

無事、タイプセットに成功すると以下のような.html ファイルのプレビューが表示されます。

Untitled

Shohei Doi

4/9/2019

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

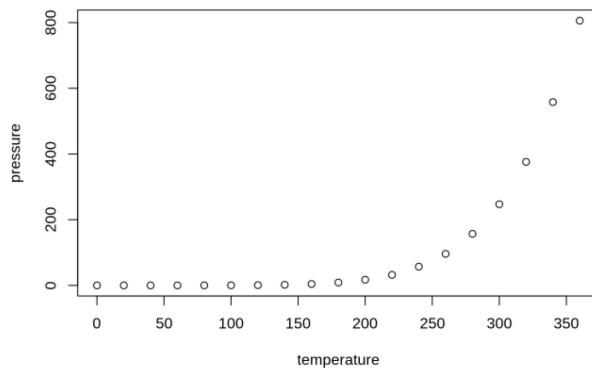
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed         dist
## Min.   : 4.0   Min.   :  2.00
## 1st Qu.:12.0   1st Qu.: 26.00
## Median :15.0   Median : 36.00
## Mean   :15.4   Mean   : 42.98
## 3rd Qu.:19.0   3rd Qu.: 56.00
## Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

.Rmd ファイルを保存したフォルダに.html ファイルが生成されているはずです。.html ファイルとはウェブサイトを作成するためのファイルで、ウェブブラウザ（例、Firefox や Google Chrome）で開くことできれいに見えます。

5.1.1 参考になるサイト

- R Markdown の公式サイト
- R Markdown: The Definitive Guide
- R Markdown のチートシート (pdf)
- 比治山大学の前田和寛先生のR Markdown 入門

5.2 Markdown 記法

5.2.1 Markdown とは *

R Markdown とは Markdown と R スクリプトを合体させたようなものです。ここでは Markdown について説明しますが、読み飛ばしても構いません。

Markdown とは計量マークアップ言語と呼ばれているようにマークアップ言語の一種です。マークアップ言語とは文章の中身と役割・外見を区別して記述する言語です。

逆に、世間で普及している Word のように文章の中身と役割・外見が混在しているエディターは WYSIWYG と呼びます。

例えば、Word ではセクションの名前などは指定することができますが、見た目はフォントのサイズが大きくなったり、太字になったりします。一方で、マークアップ言語の一種である .html ファイルでは

```
<h1>セクションタイトル</h1>
```

のように明示的に h1 というタグをつけ、h1 タグのついている文章に対して .css ファイルで見た目を決定します。同様に、LaTeX では \section{セクションタイトル} のようにタグをつけます。

基本的には WYSIWYG なソフトのほうが直観的な操作が可能で作業が楽ではあるものの、マークアップ言語はテキストで役割や外見も決めるので再現可能性が高いと言えるでしょう。

そこで、より簡便なマークアップ言語として登場したのが Markdown 記法です。なので、HTML 記法を使うこともできます。

5.2.2 セクション

Markdown では # を使ってセクションのタイトルを記述します。# が多くなればなるほどより小さな見出しになります。

```
# レベル1  
## レベル2  
### レベル3  
#### レベル4
```

5.2.3 パラグラフ

空行を入れると新しいパラグラフになります。

同じパラグラフです。

同じパラグラフです。

同じパラグラフです。同じパラグラフです。

違うパラグラフです。

違うパラグラフです。

違うパラグラフです。

- なので、パラグラフ内でも一文ごとに改行したほうが見やすいと思います。

5.2.4 箇条書き

番号なしの箇条書きの場合は=を、番号付きの箇条書きの場合は1.を入れます。

- 番号なし箇条書き
 - 番号なし箇条書き
 - 番号なし箇条書き
 - 番号なし箇条書き
- 1. 番号付き箇条書き
 - 1. 番号付き箇条書き
 - 2. 番号付き箇条書き
 - 3. 番号付き箇条書き

タブ（半角スペース4つ分）を入れると階層構造をつけることができます。

- レベル1
 - レベル2
 - レベル1
 - レベル2
 - レベル1

5.2.5 文字の強調

* もしくは_で囲むと斜体になり、** もしくは__で囲むと太字になります。

*斜体*と**太字**

斜体と太字

`で囲むとコードになり、~~で囲むと打ち消されます。

`code`と~~打ち消し~~

code と打ち消し

- 日本語の LaTeX では打ち消しに対応していないので、表示させていません。

5.2.6 引用

> から始めるとき引用になります。

> 引用文です。

引用文です。

5.2.7 リンク

リンクを貼る場合は [リンク名](リンク先の URL) あるいは <リンク先の URL> とします。

- [RStudio](<https://www.rstudio.com/>)
- <<https://www.rstudio.com/>>

- RStudio
- <https://www.rstudio.com/>

5.2.8 画像、表

画像を埋め込む場合は![画像名](画像のパス) とします。

![Rlogo](figures/Rlogo.png)

表を埋め込む際には次のように書きます。

項目1	項目2	項目3
りんご	100	赤
みかん	80	オレンジ

項目 1	項目 2	項目 3
りんご	100	赤

項目 1	項目 2	項目 3
みかん	80	オレンジ

5.2.9 数式

LaTeX 記法による数式を記述できます。インラインの場合は \$ で囲み、ディスプレイの場合は \$\$ で囲みます。

- .html の場合、mathjax によって数式を表示するのでオフラインでは表示できません。

確率変数 X_i は平均 μ 、分散 σ^2 の正規分布に従う。

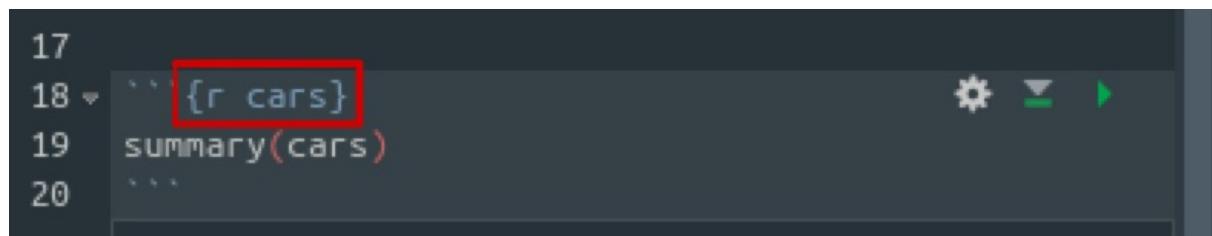
確率変数 X_i は平均 μ 、分散 σ^2 の正規分布に従う。

```
$$
X_i \sim \mathcal{N}(\mu, \sigma^2)
$$
```

$$X_i \sim \mathcal{N}(\mu, \sigma^2)$$

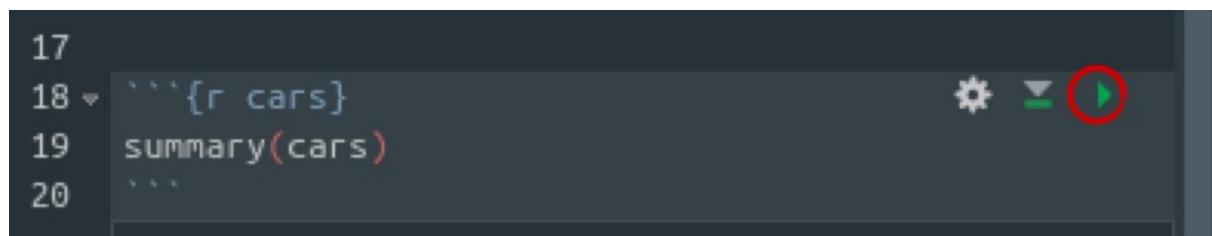
5.3 R チャンク

R Markdown 内で R コードを記述する際には R チャンクと呼ばれるものの中で行います。R チャンクは次のような形をしています。Ctrl + Alt + I で R チャンクを挿入することができます。



```
17
18 ```{r cars}
19 summary(cars)
20 ```
```

まず、この部分は後述するチャンクオプションを指定する場所になります。ここでは R コードであること、チャンク名を cars と指定しています。



```
17
18 ```{r cars}
19 summary(cars)
20 ```
```



図5.1 Rlogo

R Mrkdownにおいても R スクリプトと同様に **Ctrl + Enter** でコードを実行することができます。あるいは R チャンクの右上のボタンをクリックしても実行できます。実行されたコードはチャンクの直下に表示されます。

A screenshot of an R Markdown code editor. The code in the editor is:

```
17  
18 * ` ``{r cars}  
19 summary(cars)  
20 ````
```

The right-hand toolbar contains three buttons: a gear icon, a green downward arrow icon (circled in red), and a green right-pointing arrow icon.

右上から二番目のボタンはこの R チャンクの直前の R チャンクまでのコードを全て実行するボタンになります。

5.3.1 チャンクオプション

チャンクオプションを指定することでコードとそのアウトプットをどのように出力するかを制御することができます。主なものをまとめておきます。

- `eval=FALSE` とするとコードは表示されるが実行されない。
- `echo=FALSE` とするとコードは実行されるが表示されない。
- `include=FALSE` とするとコードは実行されるがコードも実行結果も表示されない。
- `warning=FALSE` や `error=FALSE`、`message=FALSE` とすると警告やエラー、メッセージが表示されない。

例えば、`{r, echo=FALSE}` のように書きます。

デフォルトを変更したい場合は冒頭で `knitr::opts_chunk$set(echo=TRUE)` のように設定します。

5.4 yaml ヘッダー

yaml ヘッダーとは.Rmd ファイルの冒頭で---によって囲まれた箇所で、ページ全体の設定を行います。初期状態では

```
---
```

```
title: "Untitled"
author: "Shohei Doi"
date: "4/9/2019"
output: html_document
---
```

となっていますが、`title` や `author`、`date` でタイトル、著者、日付を設定できます。

5.4.1 output

`output` によって出力形式を決定します。これによって yaml ヘッダーにおいてどのような項目を設定できるのかも決まります。

どのような出力形式が利用可能であるかは後述するとして、以下では `html_document` における主な yaml ヘッダーの設定を紹介します。

- 前田先生のページが参考になります。

5.4.2 目次

目次を出力するには次のように書きます。

```
output:  
  html_document:  
    toc: TRUE
```

目次の設定には次のようなものがあります。

```
output:  
  html_document:  
    toc: TRUE  
    toc_depth: 2  
    toc_float: TRUE  
    number_sections: TRUE
```

- `toc_depth` によってどの階層の見出しまで表示するかを決めます。
- `toc_float` を `TRUE` にすると目次がスクロールしてもついてきます。
- `number_sections` を `TRUE` にすると見出しに通し番号がつきます。

5.4.3 テーマ

テーマを決める場合は `theme` で指定します。テーマ一覧はこちらになります。

```
output:  
  html_document:  
    theme: "paper"
```

5.4.4 html と css

`css` によってカスタム.css ファイルを指定できます。`include` によって.html ファイルの挿入ができます。

デフォルトでは.css ファイルは画像データなどは全て.html ファイルに含まれてスタンドアロンな形で見ることができます。しかし、`self_contained` を `FALSE` とすると付属ファイルは別フォルダに作成され、.html ファイル自体が見やすくなります。

5.5 その他のテンプレート

`output` を変更することで、いくつかのテンプレートを使用することができます。ここでは.html ファイルが出力されるいくつかのテンプレートを紹介しておきます。

- 公式サイトの[Gallery](#)や[Formats](#)をご覧ください。