

**応用プログラミング3
第1回 ガイダンス、
Rプログラミングの基本**

専修大学ネットワーク情報学部
田中健太

1. ガイダンス

1.1 授業の到達目標（シラバスより）

1. R言語の近年の動向を捉え、適切な開発環境、パッケージを選択してプログラミングできる
 2. tidyverseを念頭に置いたモダンなデータ分析フローを理解し、分析を実践できる
 3. 機械学習やテキストマイニングなど、分析の目的に応じた手法を選択し、Rプログラムとして実装できる
- 「理論より実践、数式よりRでどう書くか」を重視した科目です

1.2 前提知識

1. Rでファイルからデータを読み込むことができる
 2. オブジェクト（変数）を作成し、要素（行や列）を参照できる
 3. オブジェクトに関数（データ加工、グラフィックス、統計処理など）を適用し、結果を得ることができる
- 他の授業でこれらを満たしている、と思っていたのですが、そうでもなさそうなので、第1回と第2回でこれらについて説明します

1.3 課題と評価

- 課題: 授業ごとに毎回、Rプログラムを作成する課題を指示します（作業時間2時間程度）。翌週月曜の23:59までにGoogle Classroom等で提出してください
- 評価: 平常点で評価します（試験なし）。各回について均等に、課題の提出状況、成果物のレベルを出席状況と照らし合わせて評価します。

→ 〆切時点での最新の回答・得点で評価します。

2. データ分析のプロセス

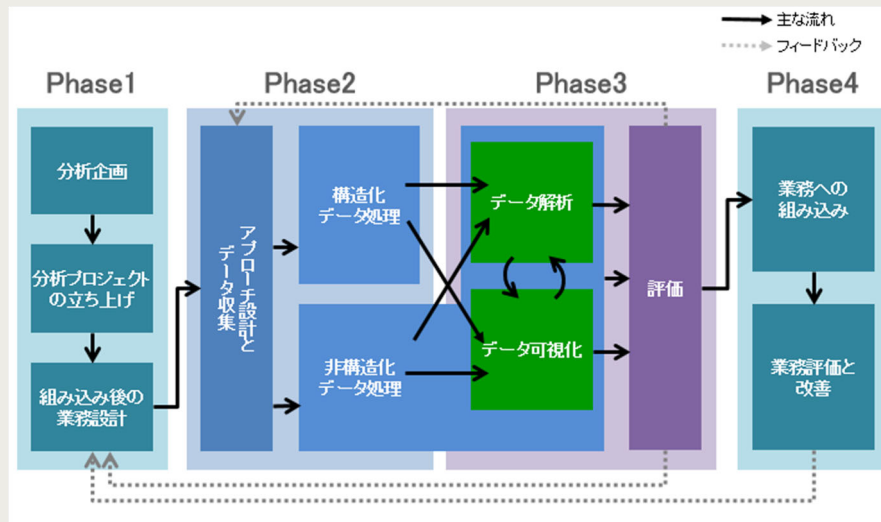
5

それでは、内容に入っていきます。

Rの操作に入る前に、まず「データ分析のプロセス」として、そもそもデータ分析にはどのような作業があるのか、ということを紹介します。

2.1 データサイエンスプロジェクトの流れ

- ビジネスにおいては、何らかの課題解決のために「プロジェクト」が結成され、予算やスケジュールなどの制約のもと活動する



出典: IPA 独立行政法人 情報処理推進機構「ITSS+ (プラス)」
<https://www.ipa.go.jp/jinzai/itss/itssplus.html>

6

はじめに、ビジネスにおけるデータサイエンスプロジェクトの流れを確認しましょう。ビジネスと書いていますが、大学の研究であっても、基本的な部分は変わりません。ここでは、IPA、情報処理推進機構が公表している、IT技術者のスキル標準のひとつである「ITSS+」で定義されている、データサイエンスプロジェクトの全体像を示しています。インターネットで「ITSS+」と検索すれば、他にもスキルチェックリストや詳しい解説なども出てきますので、ぜひご覧ください。さて、ビジネスにおいて、データを用いて何か価値を生み出すことをここではデータサイエンス、と考えますが、データサイエンスは多くの場合ひとりではなく、何人かのプロジェクトとして取り組みます。そして、思い付きで進めるのではなく、プロセスに沿って進めます。ITSS+では、プロセスを大きく4つに分けています。

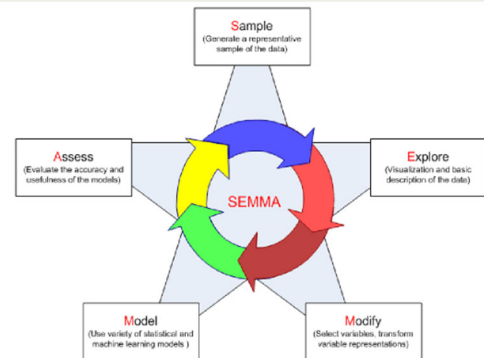
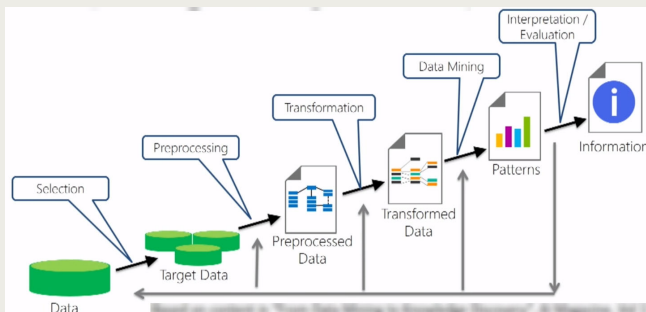
はじめに、そもそもデータを使って何を実現したいのか、企画するフェーズです。ここが曖昧だと、価値のある結果は得られません。次に、企画段階で決めた目的に沿って、必要なデータを集めます。もしまだ手元にデータがなければ、集めるところから始まりますので、プロジェクトの完了はその分だけずれ込みます。

そして、集めたデータを分析し、何らかの知見を得ます。ここが、今回主に扱う、Rなどを使って「データ分析する」という部分です。1回で終わるものではなく、トライアンドエラーで、何度もいろんなアプローチで取り組みます。そして、そこで得られた知見を、現場にどう組み込んでいくかを検討し、実行します。ここも、いい数字が得られたから明日からすべてのプロセスを変える、とはいきませんので、現場に説明し納得してもらい、時間をかけて導入するというコミュニケーションが必要になります。

ビジネスにおけるデータサイエンスの全体像は、このようなものになります。

2.2 データ分析のプロセス

- データ分析はあるステップが完了してから先に進む、というのではなく、**試行錯誤を繰り返して、何度も手戻りしながら進めていく**
- よく知られたフレームワークとしてKDDやSEMMAがある



7

さて、データ分析の作業について、もう少し細かく整理した、いくつかの**フレームワーク**があります。これらは、新しいものではなく、ビジネスの世界でコンピューターにデータが蓄積され始めた、1980年代末期から検討され、使われています。

データベースから知見を得るプロセスを整理したものが、**KDD, Knowledge discovery in databases**です。データを選択、抽出し、加工し、分析して結果を評価する、という流れがまとめられています。また、同じようにデータ分析のプロセスとして考案されたものが、**SEMMA, Sample, Explore, Modify, Model, Assess**です。データを抽出し、観察し、加工してモデルを作成し、評価するというプロセスを整理しています。ここでいうモデルとはデータの特徴やデータどうしの関係を理解し、記述したものと言えます。モデルが作成できれば、そのモデルをもとに予測などができます。

今回は、それ以上に踏み込んだ話はしませんが、このようなフレームワークを活用することで、自己流ではない、スムーズなデータ分析が可能になるでしょう。

これらのフレームワークに基づいたRプログラミングの方法を紹介する書籍もありますので、参考にしてください。

2.2 データ分析のプロセス



福島 真太郎, 2015, 「データ分析プロセス」, 共立出版
<https://www.amazon.co.jp/dp/4320123654>

3. Rの概要

9

次に、Rの概要を紹介します。この授業では、履修者はRをある程度使ったことがある、ということを前提としていますが、復習もかねて、Rとは何なのか、という全体像を捉えましょう。

3.1 Rとは

- Rは統計解析のためのプログラミング言語



10

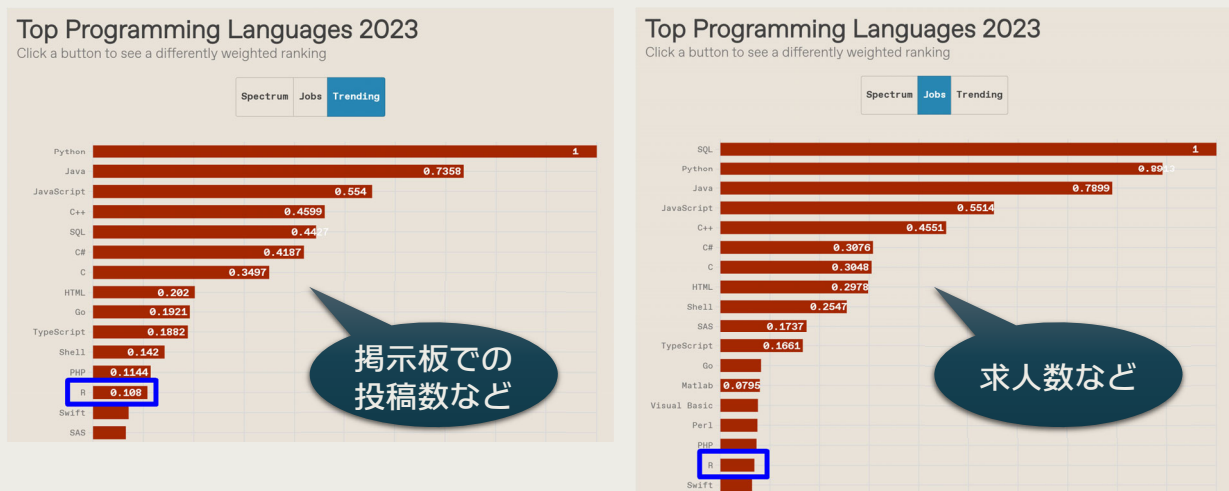
さて、そもそもRとは何でしょうか。もちろん、この授業を履修している段階で、ある程度の下調べはついていると思いますが、改めて整理すると、「Rは統計解析のためのプログラミング言語」です。世の中にはいろんなプログラミング言語がありますが、その中で、統計解析、データ分析を主な用途とした言語です。

1993年ころからオープンソースソフトウェアとして開発されており、世界中の大学や企業で使われています。実際に、IEEEが毎年行っている、プログラミング言語の人気ランキング調査では、データ分析専門であるにも関わらず他の汎用言語に並ぶ根強い人気があります。

また、実行結果の正確性もきちんと検証されており、日本の総務省統計局や、アメリカFDA、食品医薬品局など世界中の政府機関がRを採用しています。少なくとも現在、「Rで分析しました」と言ってRそのものに不信を持たれることはないでしょう。

3.2 IEEEプログラミング言語ランキング

- Rは統計解析専門の言語であるにも関わらず、高い人気をキープしている
- ビジネスにおいてはPythonのシェアが圧倒的に大きい、アカデミックではRの利用も多い



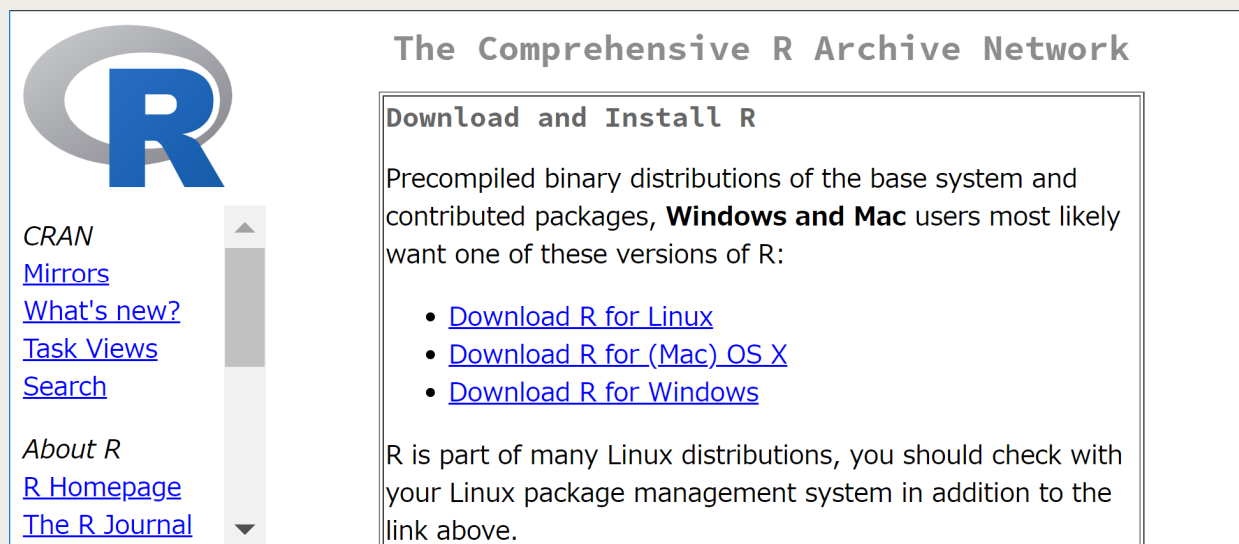
11

この図は、情報通信にかかわる研究活動を行ったり、国際規格などを策定している、IEEEという組織が毎年発表している、プログラミング言語のランキングです。さまざまなWebサイトにおける検索数、投稿数、求人数、論文数などを総合して作成されています。

- Top Programming Languages 2023 - IEEE Spectrum
<https://spectrum.ieee.org/the-top-programming-languages-2023>

3.3 CRAN

- **CRAN** (Comprehensive R Archive Network) は、Rに関する情報が集積する公式サイト
- R本体やパッケージ（後述）が公開されている



12

そのようなRですが、どこから入手すれば良いのでしょうか。Rは、**CRAN**というサイトで公開されています。

- The Comprehensive R Archive Network <https://cran.r-project.org/>

CRANは、Rの開発プロジェクトの公式サイトで、Rそのものや、次回紹介するパッケージなどが公開されています。サイトは英語ですが、"Download R" など書いているので、リンクを辿って、Rのインストーラーをダウンロードできます。macOSやLinuxなどでは、OSのパッケージ管理システム、**Homebrew**やaptなどを使ったほうが便利かもしれません。

- Homebrew https://brew.sh/index_ja

CRANに限らず、Rのことを調べようとすると、英語を読まざるを得ません。もちろん、日本のRユーザーが有用な情報を発信してくれていますが、自分のプログラムのこのエラーについて知りたい、といった目的で検索すると、たいていは公式ドキュメントや、Stack Overflowなど海外の掲示板でのやり取りが出てきますので、**英語は、読めるに越したことはないでしょう。**

4. Rプログラムの開発環境

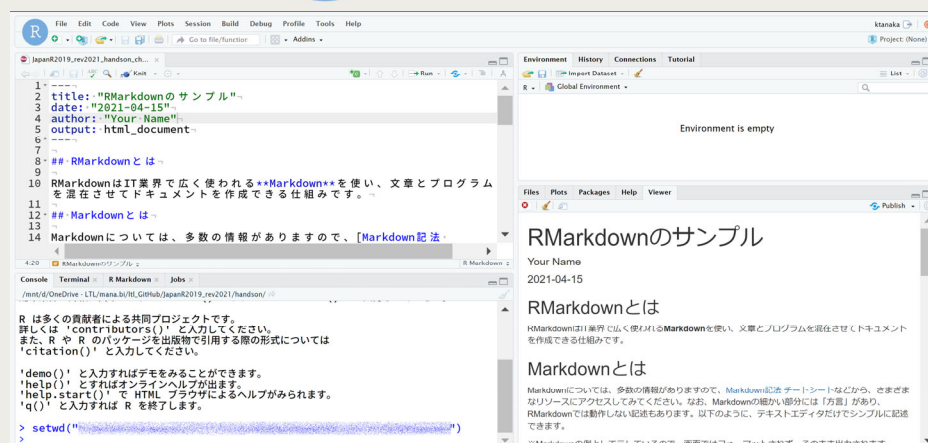
13

Rをインストールすると、それだけでもRを使ったデータ分析ができます。

ただ、Rだけでは、インターフェースがシンプルすぎて、使いづらいという声も多くあります。世の中のさまざまなテキストエディタが、Rプログラムを作成するためのモードをサポートしていますが、ここでは広く使われるRStudioとVisual Studio Codeについて紹介します。

4.1 RStudio

- **RStudio**はRプログラミングのための統合開発環境
- デスクトップ版とサーバー版、**クラウドサービス**が利用できる



14

はじめに、Rの世界で、初心者からベテランまで、幅広く使われている統合開発環境である **RStudio** を紹介します。R ストूデリオという方もいます。RStudioは、Rとは別に、**Posit社** (※) が開発しているソフトウェアで、商品として販売もされていますが、オープンソースソフトウェアでもあり、**無償で利用できます**。商品版は、大規模なプロジェクトで、複数人でRプログラムを共同開発するような場合に便利な機能が追加されていますが、個人で使う分には、無償版で何も困ることはありません。

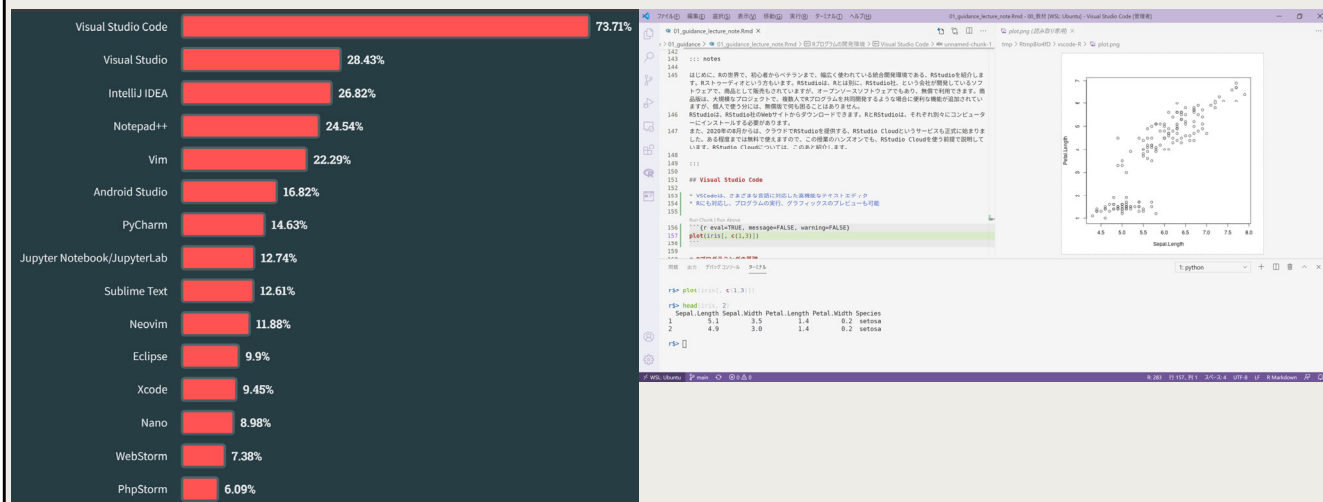
RStudioは、Posit社のWebサイトからダウンロードできます。RとRStudioはそれぞれ別々にコンピューターにインストールする必要があります。また、2020年の8月からは、クラウドでRStudioを提供するPosit Cloud (旧RStudio Cloud) というサービスも正式に始まりました。ある程度までは無料で使えますので、この授業のハンズオンでも、Posit Cloudを使う前提で説明しています。Posit Cloudについては、このあと紹介します。

※ 2022年まで、RStudio社という社名でした。

<https://www.rstudio.com/blog/rstudio-is-becoming-posit/>

4.2 Visual Studio Code

- **VSCode**は、さまざまな言語に対応した高機能なテキストエディター
- Rにも対応し、プログラムの実行、グラフィックスのプレビューも可能



15

次に、R以外のプログラミング言語にも対応し、広く使われているテキストエディターとして、**Visual Studio Code (VSCode)** を紹介します。WindowsでもmacOSでもLinuxでも使用できます。

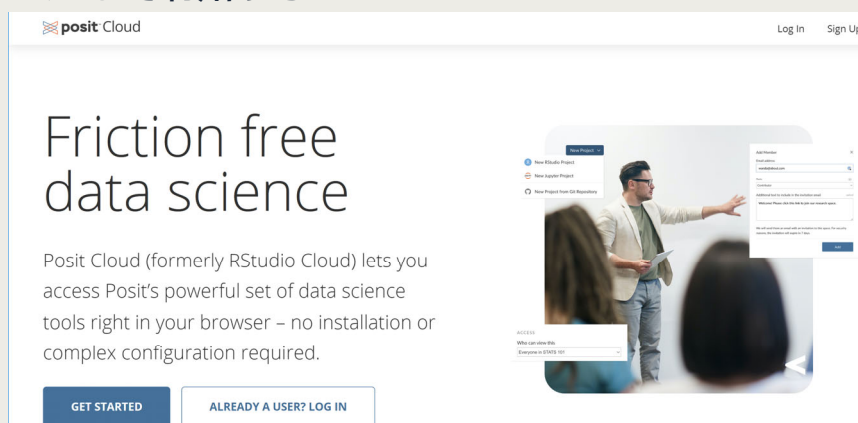
世界最大の技術系掲示板であるStack Overflowで2023年に行われた調査で、VSCodeの利用者が最も多いことがわかりました。

- Stack Overflow Developer Survey 2023 <https://survey.stackoverflow.co/2023/>

この授業では、基本的にRStudioを使いますが、筆者も普段はVSCodeを使っていますし、R以外のプログラムを書く機会もあると思いますので、VSCodeについても、使い方を学んでおくといよいでしょう。

4.3 ハンズオン環境について

- **Posit Cloud**はクラウドでRStudio環境を提供するサービス
- 1アカウント（メールアドレス）につき、月25時間まで無料で使用できる（1 CPU, 1GB RAM）
- この授業では、毎週教員が作成したプロジェクトにアクセスして利用する



<https://posit.cloud/>

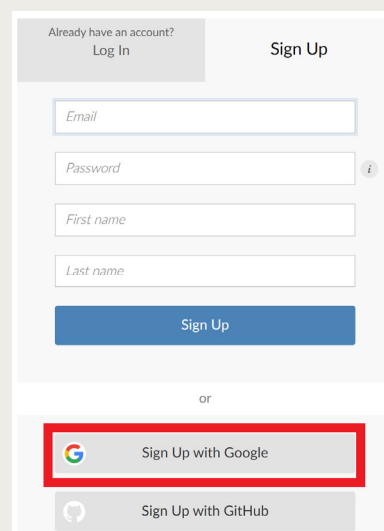
16

この講義では、**Posit Cloud**を使用して、実際にRプログラミングのハンズオン（授業内実習）や課題の作成を行います。Posit Cloudでは "プロジェクト" という単位に環境が分かれています。教員が事前に各回のプロジェクトを作成しており、そこに使用するプログラムやデータなどを配置しています。みなさんは、そのプロジェクトにブラウザでアクセスして利用します。

Posit Cloudは、無償版と有償版があり、もちろん有償版のほうがより多くのCPU、メモリを使った高度な計算ができますが、本講義の範囲では無償版で十分だと思います。

4.4 Posit Cloudへのアクセス

- <https://posit.cloud/content/7220586>
- 教員が用意したプロジェクトにアクセスする
- 初回は "Sign Up with Google" から、メールアドレス
(学籍番号@xxx-u.jp) を
入力する

A screenshot of the Posit Cloud sign-up interface. At the top, there are two links: "Already have an account? Log In" and "Sign Up". Below these are four input fields: "Email", "Password", "First name", and "Last name". A blue "Sign Up" button is positioned below the input fields. Below the button, the word "or" is centered. Underneath "or", there are two buttons: "Sign Up with Google" (which is highlighted with a red rectangular border) and "Sign Up with GitHub".

17

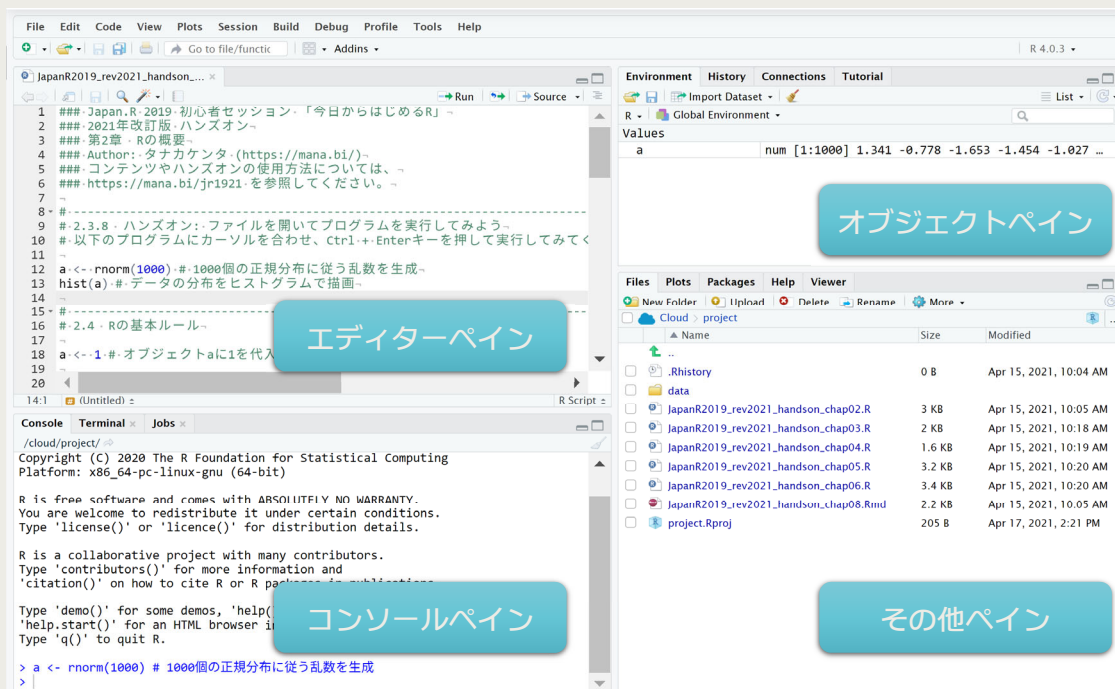
Posit Cloud (のプロジェクト) をはじめて利用する場合、ユーザー登録が必要です。個人で使用する場合には、どのようなメールアドレスを使用してもかまいませんが、この授業では大学のメールアドレスを使用してください。本学のメールはGMailなので、画面の "Sign Up with Google" ボタンを押して、Googleアカウントのパスワードを入力してください。これで、Posit Cloudの利用登録が完了します。

すると、画面が切り替わり、本日の授業用のプロジェクトが表示されます。

※次回以降は、"Log in" のほうを選択してください。

4.5 RStudioの基本操作

- RStudioは大きく4つのペインから構成される



18

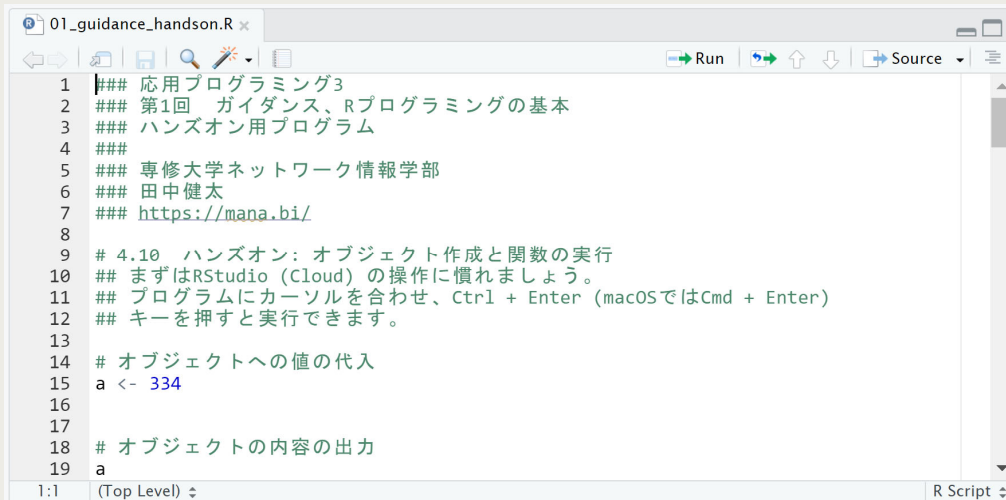
Posit Cloudにログインし、ワークスペースがコピーされると、RStudioの画面が表示されます。デスクトップ版と、基本的な見た目は変わりません。

RStudioの画面は、大きく4つの領域、ペインに分かれます。このうち、左上のエディターペインについては、まだファイルを開いていないので、表示されていません。基本的には、エディターペインでプログラムを書いて、コンソールペインで実行結果を確認します。プログラムの中で作成したオブジェクト、変数の情報をオブジェクトペインで見ることができます。また、すでに作成したファイルやデータの一覧、プログラムの中で作成したグラフィックスなどは右下のペインで表示されます。

このあと、それぞれのペインについてもう少し紹介します。

4.5.1 エディターペイン

- エディターペインでRプログラムなどを記述する
- 記述したプログラムは **Cmd+Enter** で実行できる
- 参考: RStudioキーボードショートカット (完全版)
<http://kohske.github.io/R/rstudio/cheetsheet/RStudio-Rmdv2-cheat.pdf>



```
1  ### 応用プログラミング3
2  ### 第1回 ガイダンス、Rプログラミングの基本
3  ### ハンズオン用プログラム
4  ###
5  ### 専修大学ネットワーク情報学部
6  ### 田中健太
7  ### https://mana.bi/
8
9  # 4.10 ハンズオン: オブジェクト作成と関数の実行
10 ## まずはRStudio (Cloud) の操作に慣れましょう。
11 ## プログラムにカーソルを合わせ、Ctrl + Enter (macOSではCmd + Enter)
12 ## キーを押すと実行できます。
13
14 # オブジェクトへの値の代入
15 a <- 334
16
17
18 # オブジェクトの内容の出力
19 a
```

19

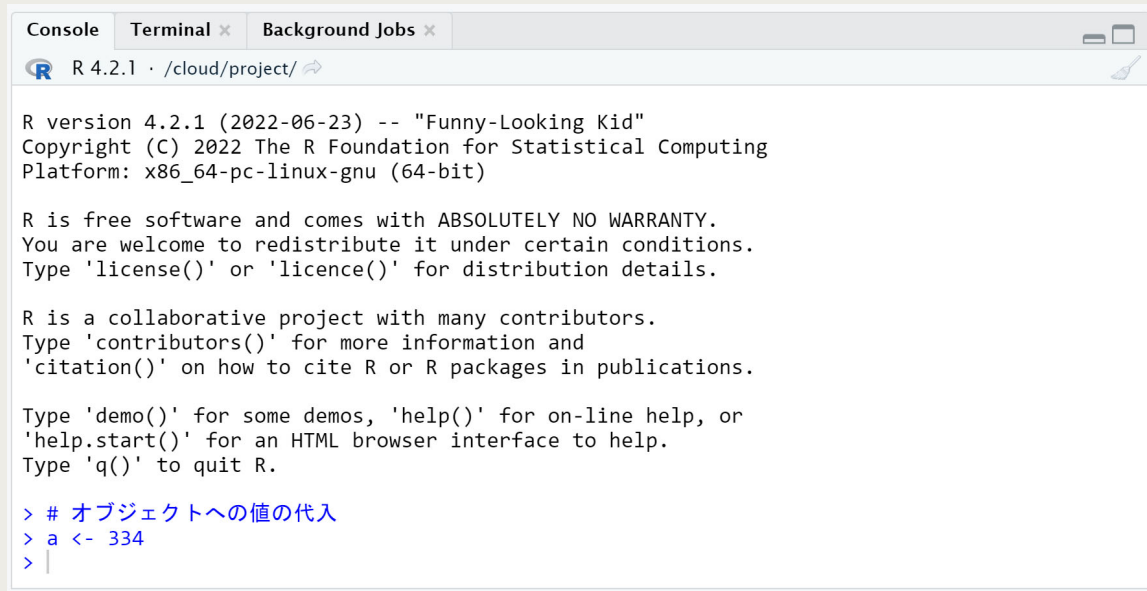
まず、**エディターペイン**です。ファイルを開くと、エディターペインが表示されます。エディターペインでは、主にRプログラムを作成します。RStudioはRだけでなく、PythonやSQLなど他の言語のプログラムにも対応していますので、プログラミング全般をRStudioで行えます。

そして、Rプログラムについては、**実行したい部分にカーソルを移動させ、Cmd + Enter (WindowsではCtrl + Enter) キーを押すと、プログラムが実行され、コンソールペインなどに結果が表示されます。**マウスで右上の "Run" ボタンを押しても同じ結果が得られますが、慣れてくるとキーボードから手を離したくなくなりますので、ショートカットキーを使ったほうが便利でしょう。

なお、"Run" ボタンの隣の **"Source" ボタンは、ファイル全体をまとめて実行します。**この場合は、画面に出力する、と明示的に記述した結果だけが出力されます。

4.5.2 コンソールペイン

- コンソールペインではRが起動し入力を待ち受けている
- 直接プログラムを入力、実行もできる



The screenshot shows the R console window with the following content:

```
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> # オブジェクトへの値の代入
> a <- 334
> |
```

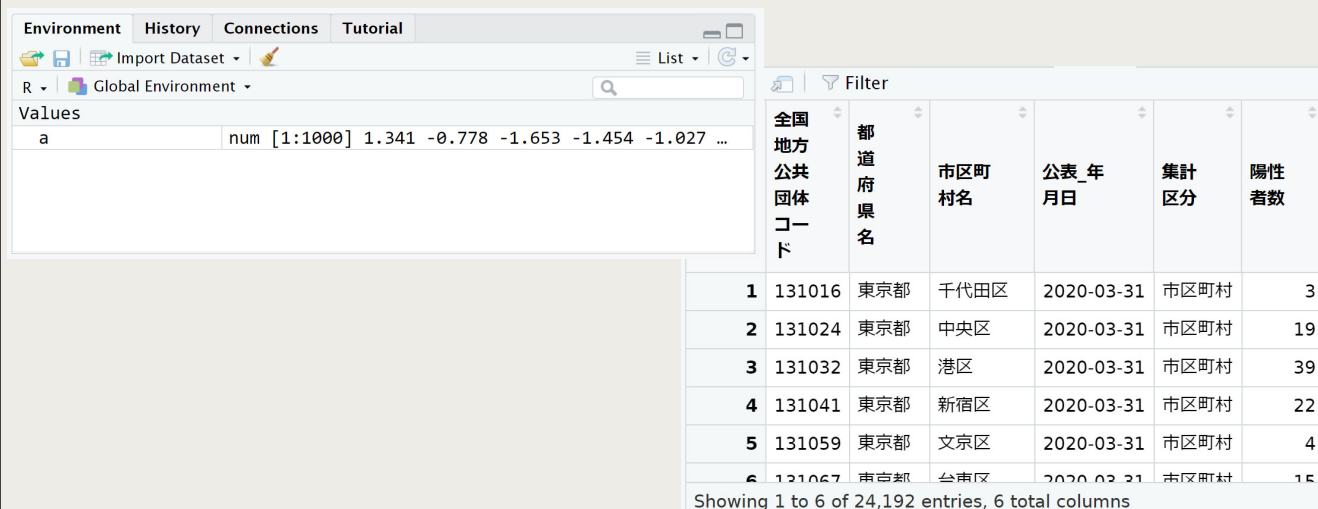
20

次に、**コンソールペイン**です。いわゆる、Rそのものがここで入力を待ち受けています。**コンソールに直接プログラムを書いてEnterキーを押すことで実行できます。**

実際には、先ほど述べたようにエディタペインでCmd + Enterキーを押して、プログラムを実行することが多いでしょう。

4.5.3 オブジェクトペイン

- プログラムの中で作成した**オブジェクト（変数）**の情報を確認できる
- データフレームなどはクリックすると内容が表示される



	全国 地方 公共 団体 コード	都 道 府 県 名	市区町 村名	公表_年 月日	集計 区分	陽性 者数
1	131016	東京都	千代田区	2020-03-31	市区町村	3
2	131024	東京都	中央区	2020-03-31	市区町村	19
3	131032	東京都	港区	2020-03-31	市区町村	39
4	131041	東京都	新宿区	2020-03-31	市区町村	22
5	131059	東京都	文京区	2020-03-31	市区町村	4
6	131067	東京都	台東区	2020-03-31	市区町村	15

Showing 1 to 6 of 24,192 entries, 6 total columns

21

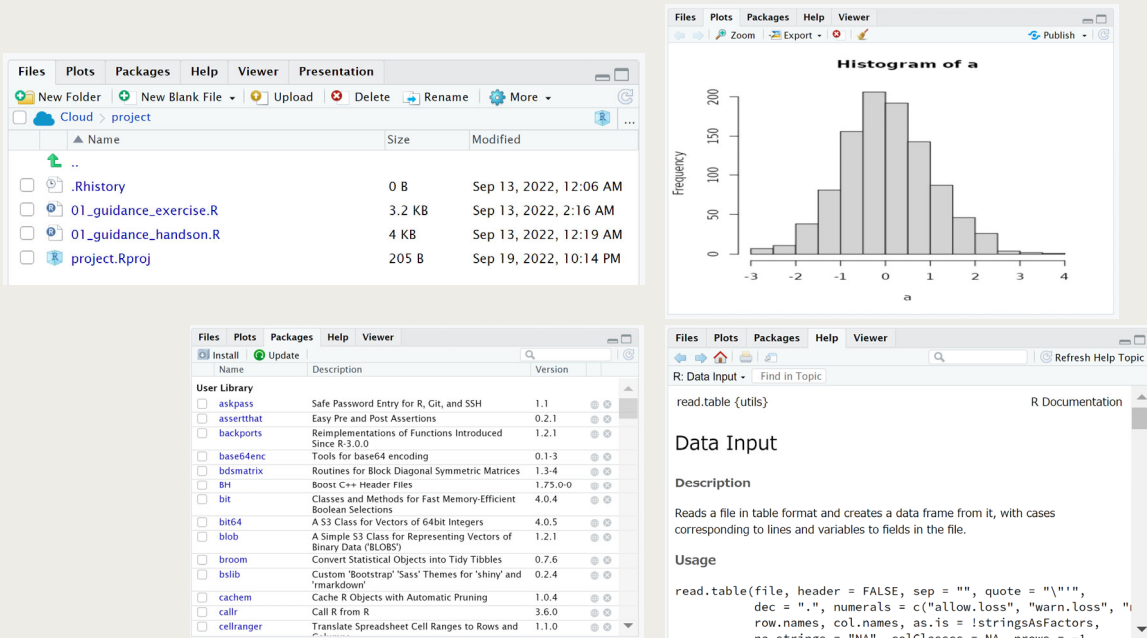
それから、**オブジェクトペイン**です。ここでは、Rプログラムを実行する中で作成される**オブジェクト**と言うものの名前、種類、サイズなどが表示されます。**オブジェクトは、変数とも呼ばれます。**もちろん、「オブジェクトの情報を得るプログラム」を書いて、コンソールで確認することもできますが、せっかくの統合開発環境なので、活用しましょう。

また、後ほど紹介するデータフレームというデータ構造については、オブジェクトペインでオブジェクト名をクリックすると、エディタペインに別タブでデータフレームの中身が表示されます。データの様子を簡単に確認する手段として便利です。

その他、タブで過去のプログラム実行履歴や、高度な内容になりますので触れませんが、他のサーバーとの接続状態なども確認できます。

4.5.4 その他ペイン

- ファイル一覧やグラフィックス表示、**パッケージ管理**などの機能が集約され、タブで切り替え可能



22

右下のペインは、タブで切り替えられるさまざまな役割がまとまっています。"Files" ペインでは、いまRStudioが見ているフォルダーのファイル一覧が表示されます。ここでファイルをクリックすることで、エディタペインなどで開くことができます。右上の"..."というメニューをクリックすると、他のフォルダーに移動できます。なお、ややこしいですが、RStudioが見ているフォルダーと、コンソールペインで起動しているRが見ているフォルダーは異なります。RStudioを使う場合は、"Files" タブを基準に考えることがほとんどだと思いますので、フォルダーを移動した場合は、歯車アイコンから、"Set As Working Directory" メニューを選択して、Rに、RStudioが見ているフォルダーに移動するよう指示します。

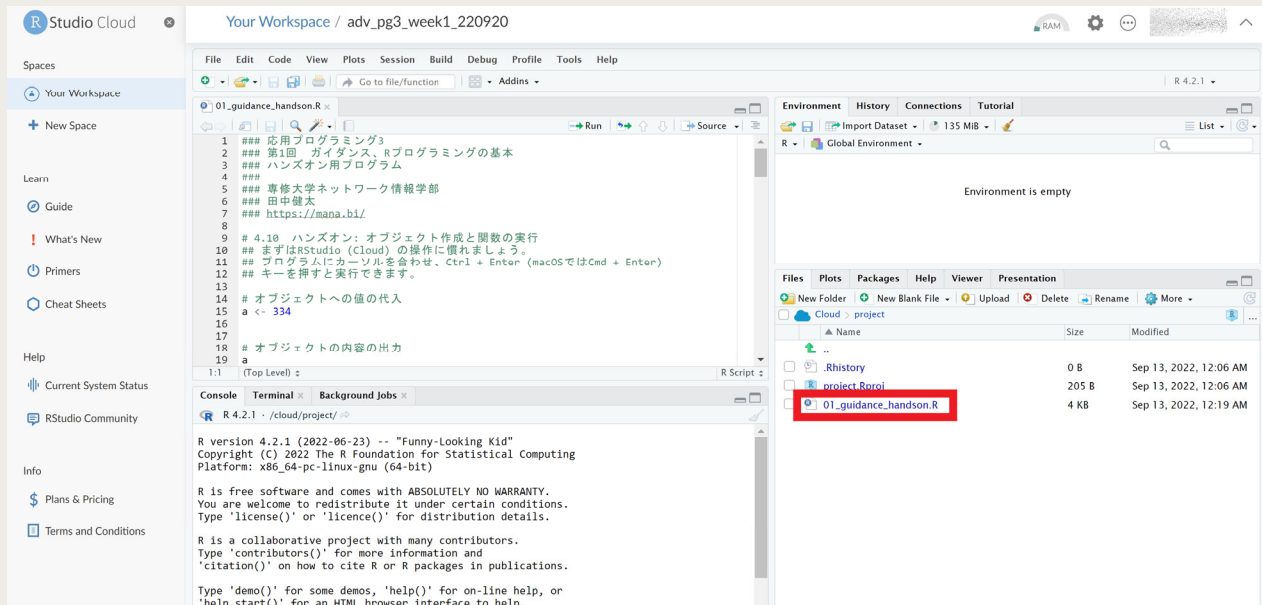
また、Rプログラムの中でグラフィックスを作成するコードを実行すると、"Plots" タブに表示されます。画面で表示を確認し、"Export" ボタンからファイルとして出力できます。また、ペインが狭くて見辛いという場合は、"Zoom" ボタンを押すと、別ウィンドウとしてグラフィックスが表示されます。

次に、"Packages" タブでは、パッケージというものを管理します。パッケージをインストールする、読み込むプログラムもありますが、このタブではマウス操作で、パッケージのインストールや読み込みを行うことができます。

最後に、"Help" タブでは、Rが提供するさまざまな関数、パッケージについてのヘルプが参照できます。実際の使い方は、このあとのハンズオンで紹介しています。

4.6 ハンズオン: オブジェクト作成と関数の実行

- カーソル行のプログラムは `Cmd+Enter` で実行する
- マウスなどで範囲選択すると、そこだけ実行できる



23

それでは、実際にPosit Cloudでプログラムを実行してみましょう。右下の [Files] タブから **01_guidance_handson.R** というファイル名をクリックして、開いてください。

今回は、基本的にはプログラムを上から順に実行していくことになります。RStudioでは、カーソルがある行のプログラムは`Cmd + Enter`で実行します。

オブジェクトに格納できるデータの種類については、次回詳しく説明します。

5. Rプログラミングの概念

24

ここで、Rプログラミングについて、基本的なルールを紹介します。前述したように、この授業ではRの使用経験があることを前提にしていますが、他の授業では何となく道具として使っただけで、体系立てて学ぶ機会がなかったかもしれないので、ここでしっかり理解しておきましょう。

5.1 オブジェクトと関数

- Rはデータをオブジェクトに格納し、関数を適用して処理する
- オブジェクトは変数とも呼ばれ、矢印 (< と -) で代入する
- 作成したオブジェクトに関数を適用する
- # でコメントが書ける



25

さて、ここから、Rプログラミングの基本を紹介していきましょう。Posit Cloudのワークスペースで、01_guidance_handson.R が開いていると思いますが、プログラムを順に実行してみましょう。

まず、Rにおける処理とはどういうものか、というところですが、一言で言うと、Rではデータをオブジェクトに格納し、関数を適用して処理します。Rではデータを格納する「箱」のことをオブジェクトや変数と呼びます。他の多くの言語では、オブジェクトを作成する際、`"a = 1"` といったようにイコールでオブジェクトに値を代入しますが、Rでは不等号とハイフンを使った矢印で代入します。実は、イコールも使えます。ただ、Rの公式ドキュメントや解説書などでは、ほとんどが矢印を使っているので、矢印を使ったほうが無難でしょう。そのようにして作成したオブジェクトに、何らかの処理内容が定義された関数を適用します。その結果は、画面に出力されたり、また別のオブジェクトに代入することができます。

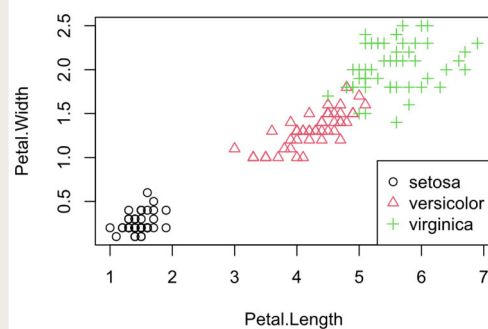
また、ファイルにプログラムを記述する際、どのような処理なのかをあらわすコメントを書きます。Rでは、# がコメントの開始を表す記号で、それ以降の文字列はコメントとして扱われ、処理されません。なお、Rでは複数行をまとめてコメントアウトするような書き方はないので、1行1行 # をつけてコメントアウトします。RStudioでは、コメントアウトしたい範囲を選択し、Cmd + Shift + cキーを押すと、コメントアウトまたはその解除ができます。

5.1 オブジェクトと関数

```
a <- 1 # オブジェクトaに1を代入
a # aの中身を参照
## [1] 1
```

```
a <- 3 # オブジェクトaに3を代入
sqrt(a) # aの平方根を算出する
## [1] 1.732
```

```
# グラフィックスを作成してみる (詳細は後述)
plot(Petal.Width ~ Petal.Length,
     data = iris,
     col = as.numeric(Species),
     pch = as.numeric(Species))
legend(x = "bottomright",
      legend = levels(iris$Species),
      col = 1:3,
      pch = 1:3)
```



5.2 さまざまな関数

- すでに使っているが `sqrt()` や `plot()` は関数 (function) と呼ばれる
- 関数には引数やオプションを指定できる
- Rに標準で組み込まれている関数だけでも多様なデータ処理、分析ができる

```
date() # 引数のいない関数を実行
## [1] "Sun Jan 22 22:33:24 2023"
```

```
sqrt(9) # 引数が必要な関数を実行
## [1] 3
```

```
a <- 1:100
head(a, n = 10) # オプションを指定できる関数を実行
## [1] 1 2 3 4 5 6 7 8 9 10
```

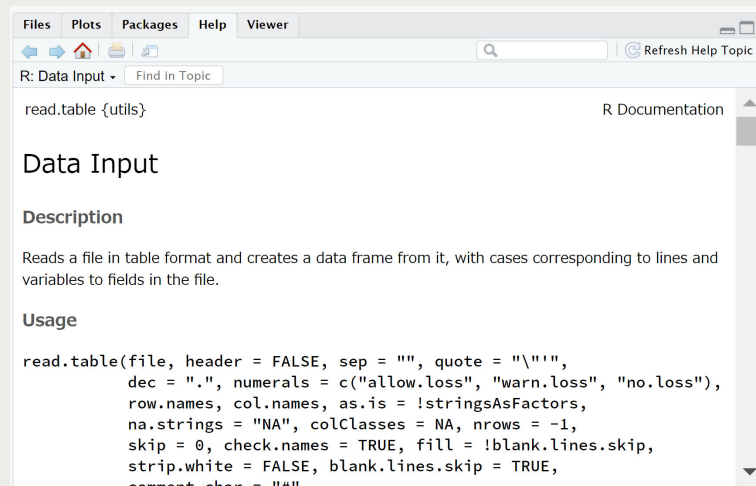
27

さて、ここまでも、平方根を計算する `sqrt()` 関数や、グラフィックスを出力する `plot()` 関数などを使っていますが、改めて関数というものについて整理しましょう。関数は、Rにおける処理が定義されたものです。一般的にオブジェクトに対して適用しますが、処理対象を指定せずに、関数だけで完結するものもあります。関数のカッコの中に指定する処理対象を、引数と言います。引数の他に、関数のふるまいを制御するオプションを指定できるものもあります。オプションは、"オプション名 = 値または設定" と指定します。

Rには標準で、さまざまなデータ分析のための関数が組み込まれていますので、統計学の教科書に載っているような基本的な分析は、R単体で実行可能です。

5.3 ヘルプの参照

- Rではすべての関数にヘルプが用意されている
- `help(関数名)` で参照できる
- RStudioでは、その他ペインの [Help] タブに表示される



28

Rでは、すべての関数にマニュアルを記述することが求められています。そのため、少なくとも標準で組み込まれている関数や、CRANに登録されたパッケージにはマニュアルが付属します。

関数のマニュアルを参照するには、`help()` 関数を使います。引数に、使い方を知りたい関数の名前を指定します。この時、引数はクォーテーションで囲まなくても大丈夫です。

RStudioでは、マニュアルは右下のペインで"Help" タブの中に表示されます。そのタブから、マウス操作でマニュアルを検索、閲覧することもできます。これらのマニュアルはローカルにインストールされていますので、インターネットにつながってなくても閲覧できます。Posit Cloudは、当然インターネットにつながっていないと使えませんが。

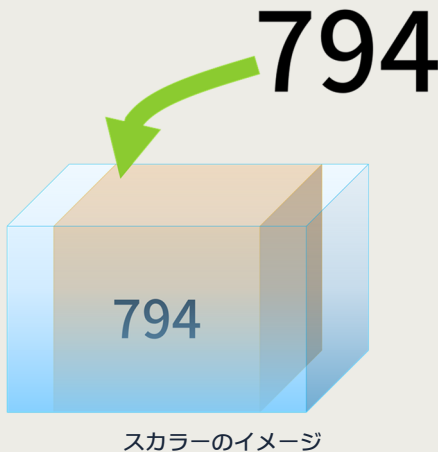
6. よく使われるデータ構造

29

ここでは、Rにおいてよく使われるデータ構造を紹介します。Rではデータをオブジェクトに格納し、関数を適用して処理する、と述べましたが、データを格納するオブジェクトの種類や特徴を理解し、適切に利用することが必要です。

6.1 スカラー

- **スカラー**は、1つだけ値が入ったオブジェクト
- **オブジェクト名 <- 値** または
オブジェクト名 <- 関数
- 数値はそのまま、**文字列はクォーテーションで囲う**



```
a <- 794 # オブジェクトaに794を代入  
a # aの中身を参照  
## [1] 794
```

```
b <- "あいうえお" # オブジェクトbに"あいうえお"を代入  
b # bの中身を参照  
## [1] "あいうえお"
```

30

はじめに、最もシンプルなデータ構造であるスカラー (scalar) を取り上げます。スカラーは、1つだけ値が入ったオブジェクトです。

スカラーは、**オブジェクト名 <- 値** として作成します。あるいは、1つの値だけを出力する関数の実行結果を代入することもできます。

数値は、そのまま記述します。文字列は、**シングルまたはダブルのクォーテーションで囲います**。数字をクォーテーションで囲うと、数値ではなく文字列として扱われますので、計算などはできなくなります。

作成したオブジェクトの内容を出力したい場合は、オブジェクト名を記述し実行します。

6.2 ベクトル

- **ベクトル** は1行に複数の値が並んで格納される構造
- **c()** 関数で作成し、**オブジェクト名[要素番号]** で参照できる
- **:** を使った範囲指定もできる

```
a <- c(794, 1192, 2021) # オブジェクトaに 794, 1192, 2021 を代入
a
## [1] 794 1192 2021
```

c(794, 1192, 2021)



ベクトルのイメージ

```
a[2] # aの2番目の要素を出力
```

```
## [1] 1192
```

```
a[2:3] # aの2番目から3番目の要素を出力
```

```
## [1] 1192 2021
```

31

次に、複数の値をまとめて格納できる構造として、ベクトル (vector) を取り上げます。ベクトルは、行方向に値が並んで格納されたオブジェクトです。他の言語では配列とも呼ばれます。ベクトルは、**c()** 関数で作成します。カッコの中に、**ベクトルとして格納したい値**をコンマで区切って列挙します。

ベクトルに格納した値は、オブジェクト名を指定して実行すると、すべての内容が出力されます。一方、**大カッコ・要素番号**と指定した場合は、その位置の要素だけが出力されます。Rでは、要素番号は1から始まります。Pythonなどは0から始まりますので、他の言語を知っている人ほど混乱しやすいですが、注意してください。

また、**始点・コロンの終点**と記述すると、範囲指定して要素を取り出せます。ここも、Rでは終点に指定した値を含めて出力しますので、Pythonとの違いに注意してください。

ベクトルに、ある条件での計測結果を格納し、他の条件で計測した別のベクトルと比較する、といったことは統計学の基本的な操作としてよく行われます。

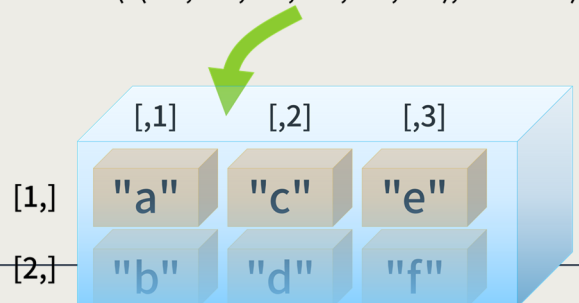
6.3 行列

- **行列** は行方向、列方向に値が並んで格納される構造
- **matrix()** 関数で作成し、**オブジェクト名[行番号, 列番号]** で参照できる
- 行または列の指定を省略すると、全行全列を参照できる

```
a <- letters[1:6] # 小文字アルファベットが格納された定数から最初の6文字を出力  
b <- matrix(a, nrow = 3) # 行列を作成  
b
```

```
##      [,1] [,2]  
## [1,] "a"  "d"  
## [2,] "b"  "e"  
## [3,] "c"  "f"
```

matrix(c("a", "b", "c", "d", "e", "f"), nrow = 2)



行列のイメージ

32

行方向、列方向にn行m列の行列としてデータを格納できる構造として、行列があります。行列は、**matrix()** 関数で作成します。カッコの中に、ベクトルなどを与えて、何行、または何列とかたちを指定します。詳しくは、ハンズオンを実行して確認してください。

行列の要素は、大カッコの中に、行番号・コンマ・列番号として指定します。2・コンマ・3とすると、2行目3列目の位置の値が出力されます。行番号または列番号を省略すると、行全体、または列全体を出力します。

Pythonのリストなどと異なり、Rの行列には数値だけ、文字列だけと同じ型のデータしか格納できません。

6.3 行列

```
b[1, ] # 行列の1行目を参照
## [1] "a" "d"
```

```
b[, 1] # 行列の1列目を参照
## [1] "a" "b" "c"
```

```
b <- matrix(a, nrow = 2, byrow = TRUE) # 行列を作成
b
##      [,1] [,2] [,3]
## [1,] "a"  "b"  "c"
## [2,] "d"  "e"  "f"
```

7. まとめ

7.1 今日の内容

- 授業の進め方について
- Rの概要
- Rの開発環境
- ハンズオン環境について
- オブジェクト作成と関数の実行

7.2 次回までの課題

- Posit Cloudプロジェクト内の
`01_guidance_exercise.R` について、指示に従ってプログラムを作成してください
- 編集したファイルは、ファイル一覧でチェックを入れ、[more] メニューから [Export] を選択し、[Download] ボタンを押してダウンロードしてください
- ダウンロードしたファイルを、Classroomの課題ページから提出してください
- 提出期限: 2023-10-02 23:59

課題Q5の出カイメージ

- 値や色は乱数なので、必ず同じ見た目になるとは限りません

