

第 1 回 R プログラミングを始めよう

平成 29 年 4 月 12 日

目次

1	ここで学ぶ事	1
2	プログラミングとは?	1
2.1	コンパイラ言語	1
2.2	インタプリタ言語 (スクリプト言語)	2
2.3	R 言語	3
2.4	チェックリスト	3
3	R の起動と終了	3
3.1	チェックリスト	4
4	統合開発環境—R Studio	4
5	開発手順	5
5.1	チェックリスト	6
6	実習	7

1 ここで学ぶ事

- プログラミングとは?
- コンパイル言語とインタプリタ言語
- R とは?
- R の開発環境
- R Studio の使い方
- 開発手順のおさらい

2 プログラミングとは?

プログラミング

⇒ コンピュータに処理させたい事柄を，コンピュータが分かる言葉で指示する行為．

2.1 コンパイラ言語

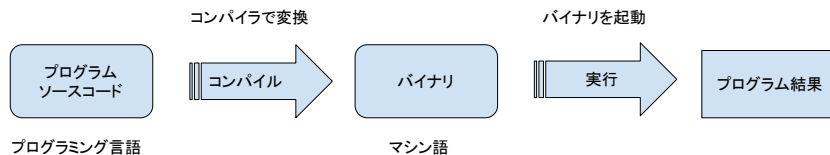
- コンピュータはどんなに複雑な処理も人間が手順を正確に指示すれば，間違いなくこなす事ができる．

↓

- コンピュータは電気のオンとオフしか理解できない．
⇒ 0 (オフ) と 1 (オン) で書かれたコンピュータ用の言葉 = 機械語 (マシン語)
- マシン語 (0 と 1 の 2 進数) で書かれたプログラムを _____ と呼ぶ．
- 人間が 0 と 1 で書かれたマシン語を操るのは至難の技! 人間の言葉に近い言語 (= プログラミング言語) から マシン語に翻訳するソフトウェア を使ってプログラミングする．

↓

- プログラミングの手順



- コンパイルしてプログラミングする言語をコンパイラ言語と呼ぶ．

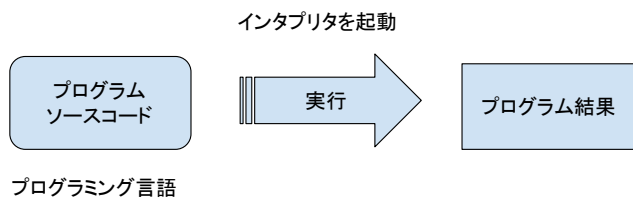
例: C, C++, Java, Objective-C, Swift, Lisp .

ただし Java は「仮想マシン」用のバイトコードという言語にコンパイルして実行する．

2.2 インタプリタ言語 (スクリプト言語)

- プログラムをマシン語に変換して実行するのではなく，プログラムを逐次翻訳しながら実行する方法もある．
⇒ 「インタプリタ」と呼ばれるソフトウェアがプログラムを 1 行ずつ読み込んで解釈し，コンピュータに処理を渡す．
- インタプリタを使って実行するプログラミング言語 ; ⇒ インタプリタ言語
例) R, JavaScript, Python, Ruby, Perl, PHP, Lisp .
- インタプリタ言語の特徴 .

- コンパイルの必要がない .
 - 書いたプログラムをインタプリタに渡せば実行できる .
 - 実行までの手順が少ないので手軽にプログラミングできる .
 - 同じプログラムを違う OS でも実行できる .
 - コンパイル言語に比べ逐次翻訳作業分だけ実行速度は遅くなる .
- プログラミングの手順



- インタプリタ言語で書かれたプログラム(ソースコード)を特に_____と呼ぶ .

2.3 R 言語

- R はインタプリタ言語 .
- R 言語でプログラムを書けば , R のインタプリタをインストールしたマシンなら OS を問わず実行できる .
- 様々な OS 用の R インタプリタ (実行環境とも呼ぶ) が無料で提供されている . 家のコンピュータにインストールしておこう !
<https://cloud.r-project.org>

2.4 チェックリスト

- プログラミング
- アルゴリズム
- バイナリ
- マシン語
- ソースコード
- コンパイル , コンパイラ
- インタプリタ
- スクリプト

3 R の起動と終了

R の実行環境を起動 \Leftarrow R 言語のインタプリタを起動していることと同じ。

R を起動すると「>」という記号が現れる。これを入力プロンプトと呼び、ここに R のプログラムを打ち込むと、インタプリタが解釈し実行してくれる。

```
> 1 + 1
[1] 2
> 4 * 3
[1] 12
> 3/2
[1] 1.5
> 10 * 3 + 2
[1] 32
> 10 * (3 + 2)
[1] 50
```

終了するには _____ をタイプ。作成したデータなどを保存するか聞かれるので y か n で答える。

- プロンプトにプログラムを 1 行ずつ入力していくと、大きなプログラムを作成するには不便。
⇒ テキスト・ファイルにプログラムを書き、それをインタプリタに読み込ませる。
- プログラムが書かれたテキスト・ファイルを _____ と呼ぶ。
- 通常、エディタでプログラムを書き、インタプリタを立ち上げてソースコードを読み込む。

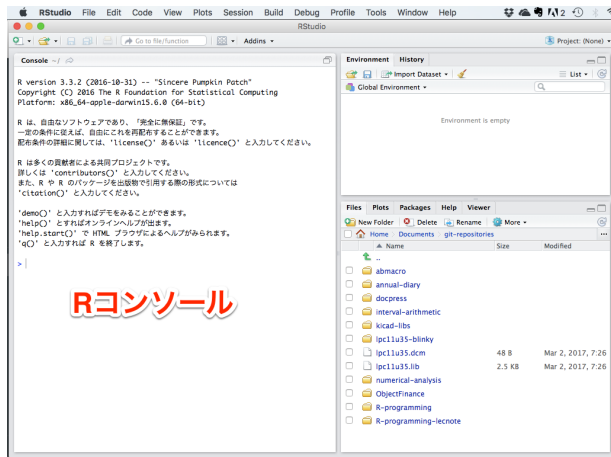
3.1 チェックリスト

- R の起動と終了方法
- プロンプト
- ソースファイル

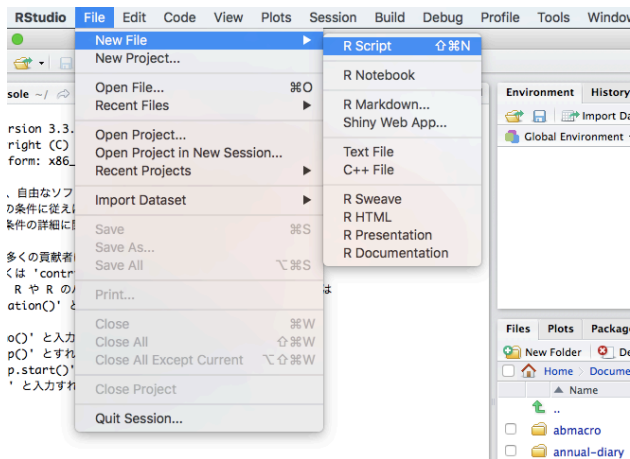
4 統合開発環境—R Studio

- 統合開発環境 (Integrated Development Environment) とは、ソースコード用エディタとプログラムの実行環境、デバッガ等を一つにまとめたソフトウェア。
⇒ RStudio—無料で使える R 用 IDE (<https://www.rstudio.com>)
- 起動画面 (設定により多少異なる)

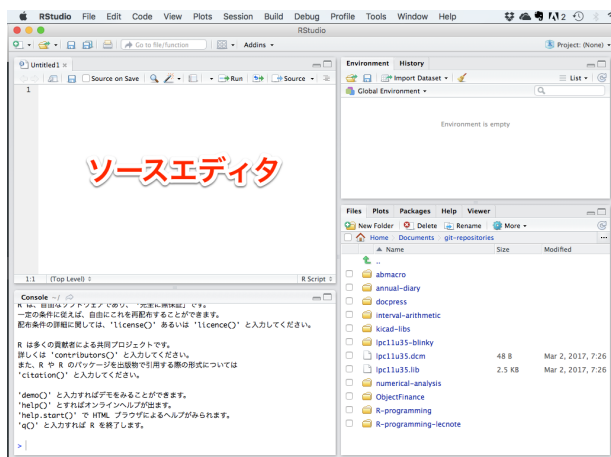
第1回 Rプログラミングを始めよう



● ソースファイルを新規作成



● エディタ起動状態



- 貸与 PC の場合、作成したソースファイルは必ず USB に保存しておきましょう！

5 開発手順

$\sqrt{2}$ を計算し画面に表示するプログラムの作成。

- (1) コンソールでプログラムのプロトタイプ（試作）を作成。

根号記号は日本語入力で「るーと」と打ち込み変換。「\」はバックスラッシュで、Windows ではキーボードの ¥ で代替、Mac では「Option」+「¥」で入力できる。

```
> cat(" 2 = ", sqrt(2), "\n")
2 = 1.414214
```

上のプログラムを解説する。cat() は括弧内の項目を画面に表示する関数。「関数」は、入力を与えられると何らかの処理を行い結果を返す。関数への入力は括弧内にコンマで区切って与える。関数に渡す入力を特に「引数（ひきすう）」と呼ぶ。

上のプログラムでは cat() 関数に 3 つの引数を渡している。

引数 引数 引数

cat("√ 2 = ", sqrt(2), "\n")

引数を区切るコンマ

これら 3 つの引数を順に画面に表示する。ダブルクォーテーションで囲まれた第 1 引数と第 3 引数は「文字列」を表す。コンピュータは文字列をテキストデータとして認識する。従って数字もダブルクォーテーションで囲むと、コンピュータは数字としてではなく文字として認識し、「1」 + 1 を計算することはできない。文字と数字は足しあわせ出来ないからだ。

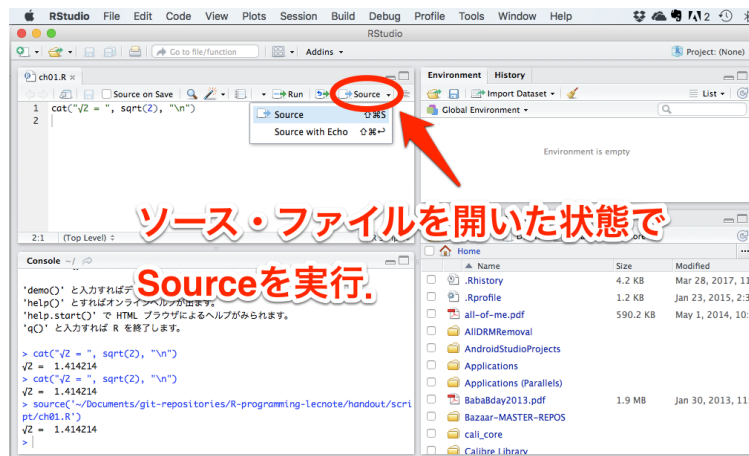
cat() 関数は文字列をそのまま画面に表示する。ただし、第 3 引数内のバックスラッシュはコントロールシーケンス（制御文字）と言って、次の一文字（この場合「n」）を特別な制御文字として扱うようにコンピュータ（この場合 R）に指示する。「n」は newline、すなわち改行を意味し、「\n」で画面に改行が挿入される。

中央の引数「sqrt()」は平方根を求める関数である。

- (2) スクリプト・ファイルの作成。「ch01.R」というファイルにプログラムを記述し保存する。

「File」→「New File」→「R Script」からソース・ファイルを作成する。文字エンコードを聞かれたら「UTF-8」を選択し、ファイルの拡張子は「.R」にする。

- (3) スクリプト・ファイルからプログラムを実行する。



5.1 チェックリスト

- 関数
- 引数
- 文字列
- コントロールシーケンス
- \n
- cat() 関数
- sqrt() 関数

6 実習

画面に以下のメッセージを表示するプログラムを「hello.R」ファイルに作成し、このファイルからプログラムを実行しなさい。

```
=====
Hello World!
=====
```