

テーマ別調べ方ガイド



# Rで統計分析を少し はじめてみよう！



## ★ 関連キーワード

- ・ R
- ・ 統計

## 「Paste（ぱすて）」とは？

ぱっと分かって、すっと頭に入る、テーマ別調べ方ガイドです。みなさんの学習をサポートする、総合図書館ラーニング・サポーター（LS）による作成です。レポート作成の際などにお役立てください

## 1. R で統計解析を少しはじめてみよう！

### 1-1. イントロダクション

ここでは、統計分析を始めたいが、なにから始めればいいのか分からない、とりあえず少し体験してみたいという人に向けて、R を使って簡単に統計分析を体験してもらおうと思います。R を実際に使うには環境構築と呼ばれる準備が必要なのですが、今回は Posit Cloud(旧 RStudio Cloud) という Web サービスを使うことで少し省略します。ここを読んで、実際にもう少しちゃんとやりたいという方は、最後に載せている学習用資料一覧などを参考にしながら、チャレンジしてみてください。

### 1-2. R とは

そもそも R とは、**統計解析<sup>1</sup>**や**データ可視化**に特化したプログラミング言語で、**無償**で利用できます。現在人気のある Python などの汎用プログラミング言語と比べると、アプリ開発などには向いていませんが、回帰分析や時系列分析、仮説検定など統計の基礎から高度な手法まで幅広く対応しています。

エクセルでも基本的な回帰分析はできますが、R を使うことで、テキストデータや空間データの分析も扱うことができます。またエクセルは表計算ソフトですので、直感的に使うことができますが、自分が行った作業の過程が少し分かりにくいところがあります。他方、R はコードを書くという行為が挟まる分、手間はかかりますが、**分析の過程が記録され作業の工程をあとから確認することができます**。さらに**同じ作業を別のデータ分析に再利用**することができます。

### 1-3. Posit Cloud の登録

R はインストールすれば単体でもすぐに分析することができます。しかし、コードが書きにくかったり、R で一番やりたい統計分析のためのデータの読み込みが少し大変だったりします。なので、一般的には **RStudio** という IDE(Integrated Development Environment、統

---

<sup>1</sup> 本文で用いられる「統計分析」と「統計解析」はいずれも「データを収集・整理し、統計的手法を用いて情報を抽出・解釈する過程」という同じ意味を示します。

合開発環境)と呼ばれるソフトウェアをダウンロードして利用します。これは R をより便利に使うためのソフトで、これから行う分析をより簡単かつ便利に行うことができます。R を始めとしたプログラミング言語を自分のパソコンで利用できるようにすることを環境構築といいますが、先述の通り、やや複雑な箇所があるので、今回は、**Posit Cloud** (<https://posit.cloud>) というサービスを利用します。Posit Cloud とは、RStudio をブラウザ上で利用することができるサービスで、RStudio を管理している団体が運営しています。サービスは多少制限はありますが、無料でも利用できるので、今回はこちらを利用します。以下、ブラウザは Google Chrome を使用しています。

まず、Posit Cloud のサイトにアクセスし、画面右上の「Log in」を選択します。

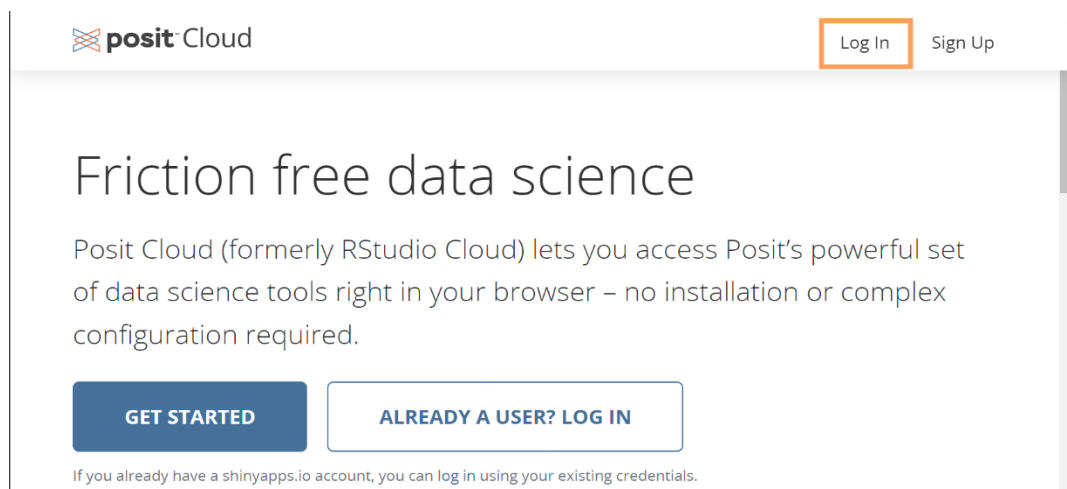


図 1. Posit Cloud のトップページ

次に「Sign Up」を選択し、画面上部に必要情報を入力するか、Google アカウントを持っている方は下の「Sign Up with Google」から手続きを進めてください。Google アカウント以外で登録(Sign up)を行った場合は、差出人“noreply@posit.cloud”から件名“Please verify your email address”というメールが届きます。メール本文中の“Verify your email”と書かれたリンクをクリックすると、承認完了です。

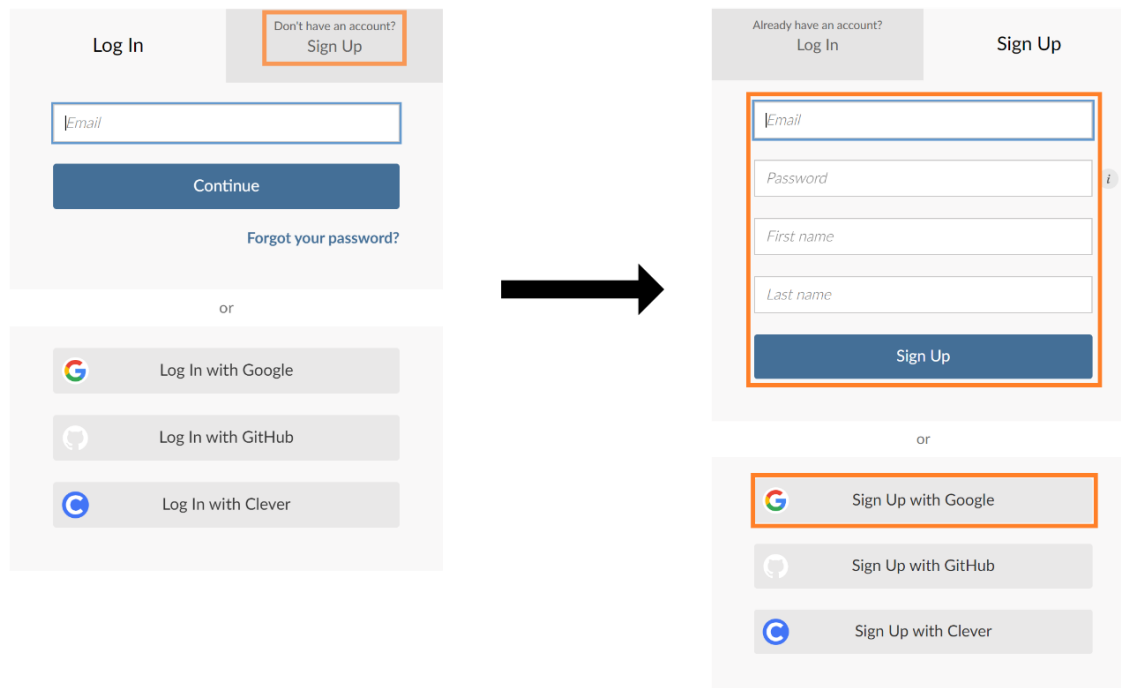


図 2. 利用登録(サインアップ)画面

登録が完了し、ログインができれば、図 3 のような画面が表示されるはずです。

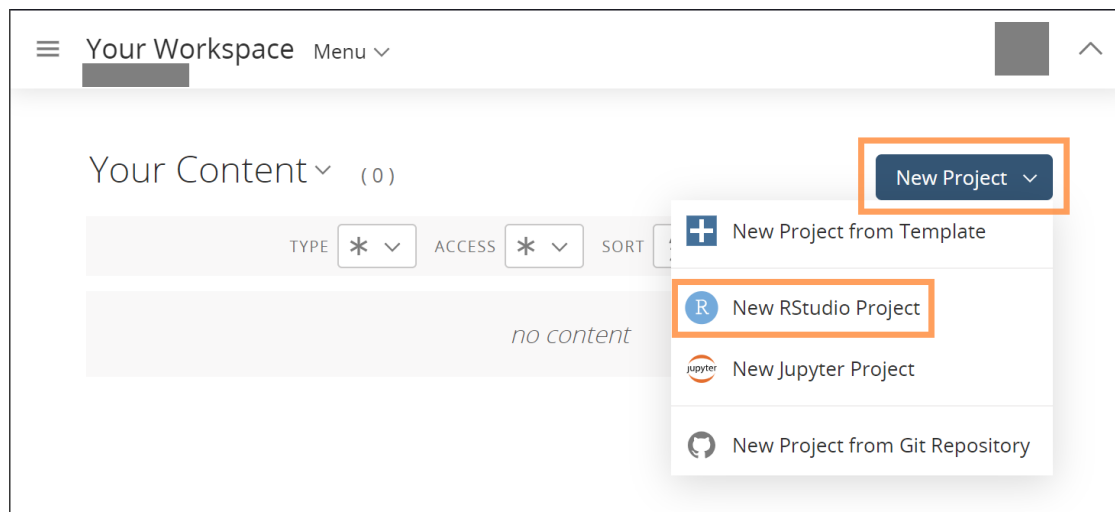


図 3. プロジェクト選択

画面右上の「New Project」を選択し、表示されたリストから「New RStudio Project」を選択してください。(次の画面の立ち上げまで多少時間がかかります)  
その後、次のような画面が表示されるはずです。

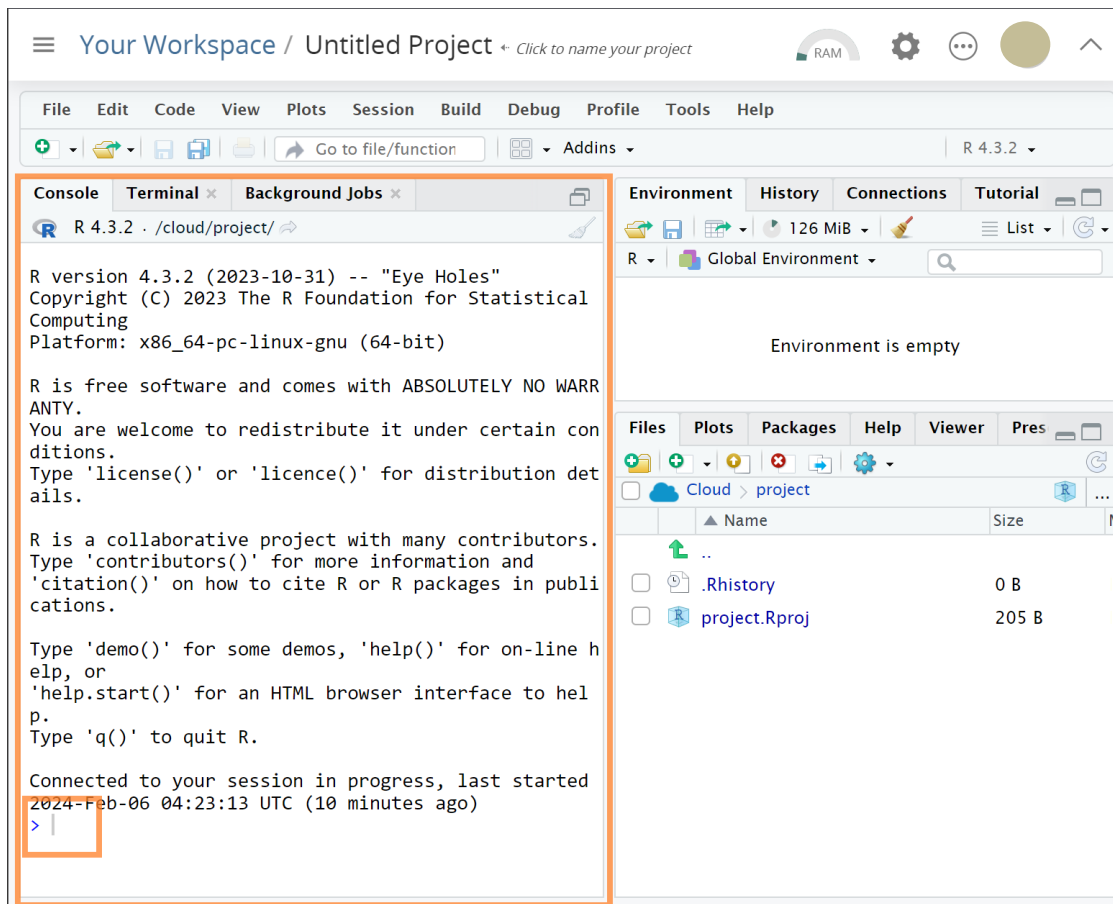


図 4. RStudio の基本画面

本パスファインダー作成時点でのバージョンから更新されている可能性がありますが、今回の演習には問題はありません。ここからの作業は基本的に画面左側の **Console(コンソール)** タブを中心に行っていきます。タブ下方の「>」のあたりをクリックし、選択された状態になったら、演習を始めましょう。

#### 1-4. 簡単な計算

R を実行するための準備が整ったので、ここから実際にコードを入力してみましょう。なお、ここからの演習の内容と構成については今井耕介(2018)『社会科学のためのデータ分析入門上』に依拠しています。

まず、キーボードの入力モードが半角になっているか確認して、次のコードを入力し、

Enter(Return)キーを押下して実行してみましょう。なお、下記の#(ハッシュ記号)の右側部分は**コメント**です。コードの実行に直接関係ありませんが、自分の作成したコードを人と共有したり、後で自分が見返したりするときに何をやっているかを伝えるために記述することが多いです。これ以降、補足としてコメントが挿入されていますが、皆さんがコード入力する際はこの部分は入力する必要はありません。

```
3 + 4 #以降、この部分にコメントを随時入れています。
```

```
## [1] 7
```

上記の「##[1]」の右側がコードの実行結果です。Posit Cloud 上の表示様式と異なるかもしれませんが、結果が一致していれば問題ありません。

コードをよく見ると、加算演算子(+)の前後は半角スペースを空けています。しかし、半角スペースを空けなくてもコードは実行可能です。

```
3+4
```

```
## [1] 7
```

短いコードだと、あまり気にならないかもしれませんが、長いコードになってくると、間違いが生じやすくなるので、基本的には+や-などの**演算子(operator)**の前後は半角スペースを空けましょう。

```
7 - 3
```

```
## [1] 4
```

```
8 / 3
```

```
## [1] 2.666667
```

```
6 ^ 3 # べき乗。日本語キーボードであれば、右上あたりの「へ」の位置です。
```

```
## [1] 216
```

```
5 * (10 - 3) # 括弧の前後にスペースは要りません。
```

```
## [1] 35
```

```
sqrt(16) # 平方根(square root)をとる関数(function)。括弧の中を引数(ひきすう)といいます(後述)
```

```
## [1] 4
```

## 1-5. 変数の作成

R では数値や文字列などのデータを、好きに命名した**変数(variable)**に格納できます。変数の命名には次のようなルールがあります。

- ▶ 数字から始められない(ただし 2 文字目以降に含めることは可能)
- ▶ スペースは含められない
- ▶ %や\$といった特殊文字は使えない…など

変数に値を格納する(割り当てる)には**割り当て演算子(<-)**を用います。

```
# 「<-」の左側に変数名。右に代入したいもの
result <- 5 + 3 # 変数は基本的には半角英数字で作成
result
## [1] 8
```

作成した変数は画面右上の**環境(Environment)**タブでいつでも確認することができます。

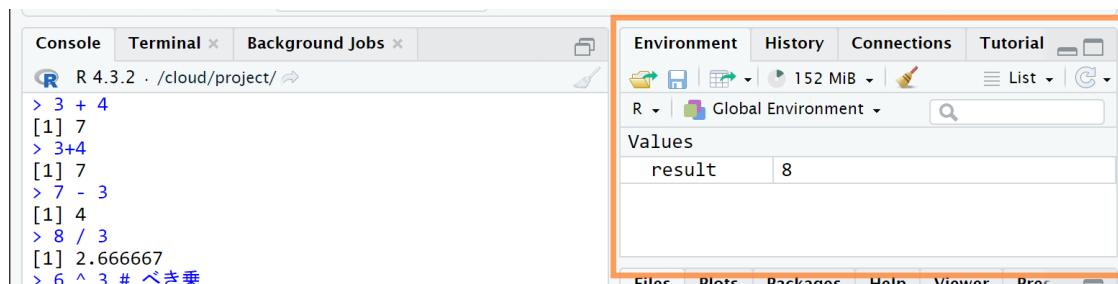


図 5. 環境タブ

同名の変数に再び代入すると、中身が**上書き**されます。

```
result <- 5 - 3
result
## [1] 2
```

```
# 大文字と小文字は区別されるので、下の入力ではエラーが発生します。
# エラーメッセージはこちらのものは日本語、Posit Cloud 上では英語になっているかと思います。
Result
## Error in eval(expr, envir, enclos): オブジェクト 'Result' がありません
```

数値だけでなく、文字列も変数に入れることができます。その場合”(ダブルクォーテーション)か’(シングルクォーテーション)で入れたい文字列を囲みます。クォーテーションはどちらを使っても良いですが、前後同じもので囲んでください。

```
daigaku <- "Osaka University"
daigaku

## [1] "Osaka University"
```

数値をクォーテーションで囲めば、文字列として扱うこともできます。ただし、その場合、先ほど行ったような演算はできなくなります。

```
Result <- "5"
Result

## [1] "5"

Result / 3

## Error in Result/3: 二項演算子の引数が数値ではありません
```

R 内で扱う**数値(numeric)**や**文字列(character)**といったモノは**オブジェクト**と呼ばれ、**クラス<sup>2</sup>**で分類されています。オブジェクトのクラス(型)は **class()**関数を使うことで確認できます。

```
class(1931)

## [1] "numeric"

class(result)

## [1] "numeric"

class(Result)

## [1] "character"
```

---

<sup>2</sup> オブジェクトとクラスの関係はプログラミングにおける難しいトピックの一つです。多少正確さは欠きますが、クラスは料理のレシピで、オブジェクトはそのレシピを元に作られる料理のようなものです。クラスがオブジェクトの種類や特性を定義し、オブジェクトはそれらを備えた具体的なデータを示します。そして、数値型は計算できて、文字型はできないなど使える関数(料理の技法)がクラスによって異なります。



四則演算や二乗根などの計算は数値型のオブジェクトでしか行なえません。一方で文字列を扱うための関数もあるのですが、今回は扱いません。

## 1-6. ベクトル作成と簡単な操作

変数には**ベクトル**を入力することもできます。ここでベクトルとは、同じ型の要素で構成された一次元の配列のことをいいます。ベクトルを作成する基本的な方法のひとつである、**c()関数**で作ってみましょう(c は連結の意味(concatenate))。

ここでは例として、2023 年の大阪府の月別平均気温を扱いましょう(入力を簡単にするために四捨五入しています)。

表 1 2023 年大阪府月別平均気温

| 月  | 平均気温 |
|----|------|
| 1  | 7    |
| 2  | 7    |
| 3  | 13   |
| 4  | 16   |
| 5  | 20   |
| 6  | 24   |
| 7  | 29   |
| 8  | 30   |
| 9  | 28   |
| 10 | 19   |
| 11 | 14   |
| 12 | 9    |

出典：気象庁ホームページ「過去の気象データ検索」

([https://www.data.jma.go.jp/obd/stats/etrn/view/monthly\\_s3.php?prec\\_no=62&block\\_no=47772](https://www.data.jma.go.jp/obd/stats/etrn/view/monthly_s3.php?prec_no=62&block_no=47772)) (最終閲覧日：2024.10.11)

```
osaka_avg_temps <- c(7, 7, 13, 16, 20, 24, 29, 30, 28, 19, 14, 9) # コンマの後ろは半角スペース
osaka_avg_temps

## [1] 7 7 13 16 20 24 29 30 28 19 14 9
```

変数名や関数名など何度も入力すると間違ってしまうことも増えてきます。そのようなミスを減らすために、RStudio では予測変換(補完機能)が使えます。下図のように、途中まで入力すると、候補が出てくるので、複数ある場合は矢印キーで目的のものを選び、Tab キーや Enter キーで選択してください

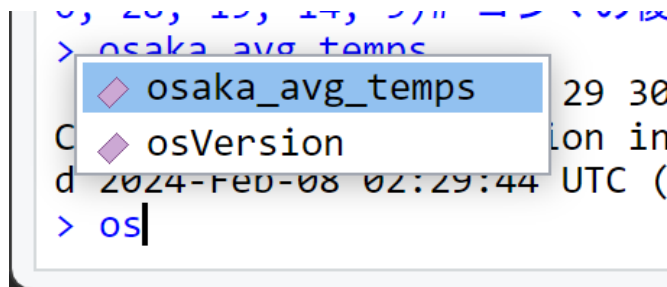


図 6. 補完機能

複数のベクトルをまとめることもできます。

```
first_half <- c(7, 7, 13, 16, 20, 24)
last_half <- c(29, 30, 28, 19, 14, 9)
year_2023 <- c(first_half, last_half)
year_2023 #osaka_avg_temps と同じ内容が表示されます。

## [1] 7 7 13 16 20 24 29 30 28 19 14 9
```

ベクトル内の要素を取り出すこともできます。

これを**要素指定(indexing)**といいます。要素を取り出すには、角括弧を用いて、**ベクトル名[抜き出したい要素番号]**のような形で指定します。

```
osaka_avg_temps[2] # 2 番目の要素を取り出す。

## [1] 7
```

**複数の要素を取り出す**場合は、ベクトル名[1, 2]のような形ではなく、**ベクトル名[c(1, 2)]**という形で取り出します。

```
osaka_avg_temps[c(6, 12)] # 要素の 6 番目, 12 番目の順で取り出す

## [1] 24 9
```

```
osaka_avg_temps[c(12, 6)] # 要素の 12 番目, 6 番目の順で取り出す。
```

```
## [1] 9 24
```

ここで行った操作はいずれも元のベクトルを変更しません。

```
osaka_avg_temps # 元のベクトルのまま
```

```
## [1] 7 7 13 16 20 24 29 30 28 19 14 9
```

また、数値ベクトルであれば、まとめて算術計算することもできます。平均気温を摂氏(celsius)から華氏(fahrenheit)に変換してみましょう。

```
temps_f <- (osaka_avg_temps * 9/5) + 32
```

```
temps_f
```

```
## [1] 44.6 44.6 55.4 60.8 68.0 75.2 84.2 86.0 82.4 66.2 57.2 48.2
```

## 1-7. 関数

---

ここまで使用してきた `sqrt()` や `class()`、`c()` などを**関数(function)**といいます。関数とは、入力を受け取り、それに対して何らかの処理を加え、その結果として出力を**返す(return)**仕組みのことです。関数の括弧内に入力する値を**引数(ひきすう、argument)**といいます。例えば、`sqrt()`関数であれば、正の数を引数として受け取り、その平方根を返しますし、`c()`関数であれば、数値や文字列などの1つ以上のオブジェクトを引数として受け取り、ベクトルという1つのオブジェクトとしてまとめて返します。このように、関数はさまざまな入力を受け取り、特定の処理を行って結果を返す便利なツールです。それでは、データの要約を見るのに使用する基本的な関数をいくつか紹介します。

```
length(osaka_avg_temps) # ベクトルの長さ、すなわちベクトル内の要素の数を返す。
```

```
## [1] 12
```

```
min(osaka_avg_temps) # ベクトル内の最小値を返す。
```

```
## [1] 7
```

```
max(osaka_avg_temps) # ベクトル内の最大値を返す。
```

```
## [1] 30
```

```
range(osaka_avg_temps) # ベクトル内の値の範囲(range)を返す。
```

```
## [1] 7 30
```

```
mean(osaka_avg_temps) # 平均値(mean)を返す。
```

```
## [1] 18
```

## 1-8. データフレームの操作

---

前節では、大阪府の平均気温という 1 次元のベクトルを扱いました。それでは次に、エクセルで扱うようなデータセットの簡単な操作とグラフ化を行いましょう。まず、次の関数を使ってデータセットの入った datarium パッケージを使えるようにしましょう。

```
#パッケージのインストール。クォーテーションで囲む。
```

```
# 「Warning～」となにやら不穏当な表示が出ますが、単なる注意喚起ですので、気にしないで大丈夫です。
```

```
install.packages('datarium')
```

```
library(datarium) #パッケージの読み込み。クォーテーションで囲まなくて良い
```

```
## Warning: パッケージ 'datarium' はバージョン 4.2.3 の R の下で造られました
```

R では、これまでに使っていた基本的に使える機能以外は**パッケージ**という形で

**CRAN**(Comprehensive R Archive Network)というサイトに置いてあります<sup>3</sup>。

**install.package()関数**は、引数に入力したパッケージをそのサイトからインストールしてくれる関数です。そして **library()関数**でパッケージを使えるように読み込むことができます。

R では普段使わないパッケージは無駄を省くために、ライブラリに保存されているだけであり、library()関数で呼び出すことで、使えるようになります<sup>4</sup>。プログラミング「言語」に、データセットが入っているのは奇妙な感じがしますが、言語のようなルールは存在しつつも、あくまで R は統計分析をするための機能を提供するものですので、それを学習するためのデータも同梱されています。

---

<sup>3</sup> コンシューマゲームで例えるなら、R がゲーム本編で、パッケージはダウンロードコンテンツのようなものです

<sup>4</sup> パソコンのリソース消費を抑えるため、普段は最低限の機能のみ使えるようになっています。

それでは、datarium パッケージ内の marketing というデータセットを使って、簡単な操作を行ってみましょう。

```
df <- marketing #変数 df に marketing データセットを入れる
class(df)

## [1] "data.frame"
```

型を確認すると「data.frame(**データフレーム**)」と表示されました。データフレームとは、「同じ長さの複数のベクトルを列ごとにまとめたもの」で、エクセルファイルやスプレッドシートのようなものです。ここで扱う marketing は、youtube、facebook、newspaper といった各メディアへの広告費用と売上額(sales)のデータフレームです。関数を使って中身を確認してみましょう。

**dim()関数**:データフレームの行数と列数をベクトル形式で返す。

```
dim(df)

## [1] 200  4
```

**head()関数**:データフレームの冒頭 6 行を返す。

```
head(df)

##  youtube facebook newspaper sales
## 1  276.12   45.36    83.04 26.52
## 2   53.40   47.16    54.12 12.48
## 3   20.64   55.08    83.16 11.16
## 4  181.80   49.56    70.20 22.20
## 5  216.96   12.96    70.08 15.48
## 6   10.44   58.68    90.00  8.64
```

**summary()関数**:各列の最小値、第 1 四分位数、中央値、第 3 四分位数、最大値を返す。

```
summary(df)

##  youtube      facebook      newspaper      sales
## Min.   : 0.84  Min.   : 0.00  Min.   : 0.36  Min.   : 1.92
## 1st Qu.: 89.25  1st Qu.:11.97  1st Qu.: 15.30  1st Qu.:12.45
## Median :179.70  Median :27.48  Median : 30.90  Median :15.48
## Mean   :176.45  Mean   :27.92  Mean   : 36.66  Mean   :16.83
```

```
## 3rd Qu.:262.59 3rd Qu.:43.83 3rd Qu.: 54.12 3rd Qu.:20.88
## Max. :355.68 Max. :59.52 Max. :136.80 Max. :32.40
```

これ以外にデータフレーム全体を確認する方法もあります。

**View(df)** # 「V」は大文字! でも画面右上の *environment* から直接見ることもできる。

先述したようにデータフレームはベクトルを列ごとにまとめた集合であり、今回読み込んだデータフレームは4つの変数(youtube, facebook, newspaper, sales)のベクトルが1つの表形式にまとめられたものです。ここで、**\$演算子**を使うことで、データフレーム内の個別の変数(列)を抜き出し、ベクトルとして扱うことができます。

### データフレーム名\$変数名

```
df$youtube[c(1, 2, 3)] # 長いので、最初の3つだけ取り出しています。
```

```
## [1] 276.12 53.40 20.64
```

```
df$youtube[c(1:10)] # :(コロン)で広い範囲を指定することもできます。
```

```
## [1] 276.12 53.40 20.64 181.80 216.96 10.44 69.00 144.24 10.32 239.76
```

データフレームもベクトルと同じように要素指定することができます。ただし、2次元配列なので、行と列のそれぞれを要素指定が必要となります。

### データフレーム名[rows(行), columns(列)]

のように指定します。

```
#[]内で列名を指定するときは""を忘れない。
```

```
df[c(1:10), "youtube"] # 先ほどの df$youtube[c(1:10)]と同じ
```

```
## [1] 276.12 53.40 20.64 181.80 216.96 10.44 69.00 144.24 10.32 239.76
```

```
df[c(1, 2, 3),] # 空欄にするとすべての列(行)を抜き出す。
```

```
## youtube facebook newspaper sales
```

```
## 1 276.12 45.36 83.04 26.52
```

```
## 2 53.40 47.16 54.12 12.48
```

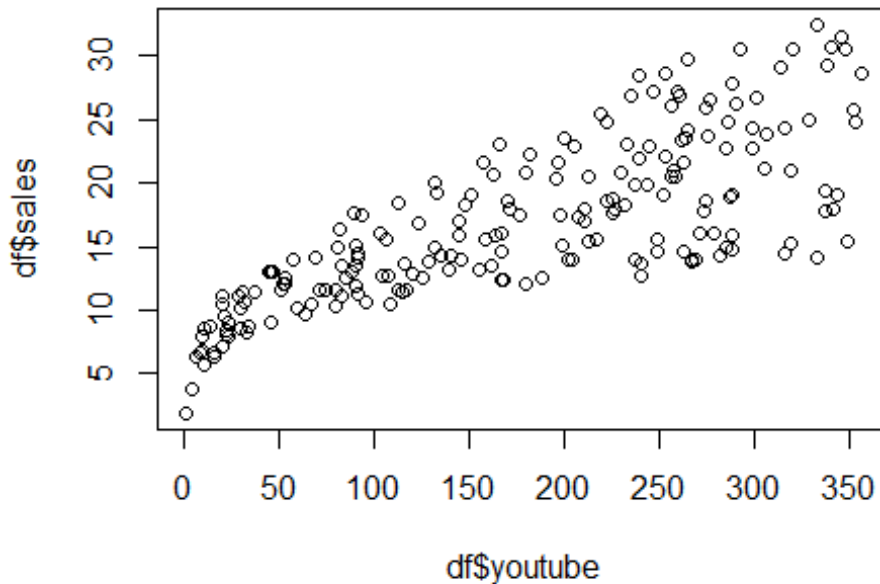
```
## 3 20.64 55.08 83.16 11.16
```

## 1-9. 散布図作成

---

このデータフレームを使って散布図を作成してみましょう。

```
plot(x = df$youtube, y = df$sales) #コンマで区切る
```

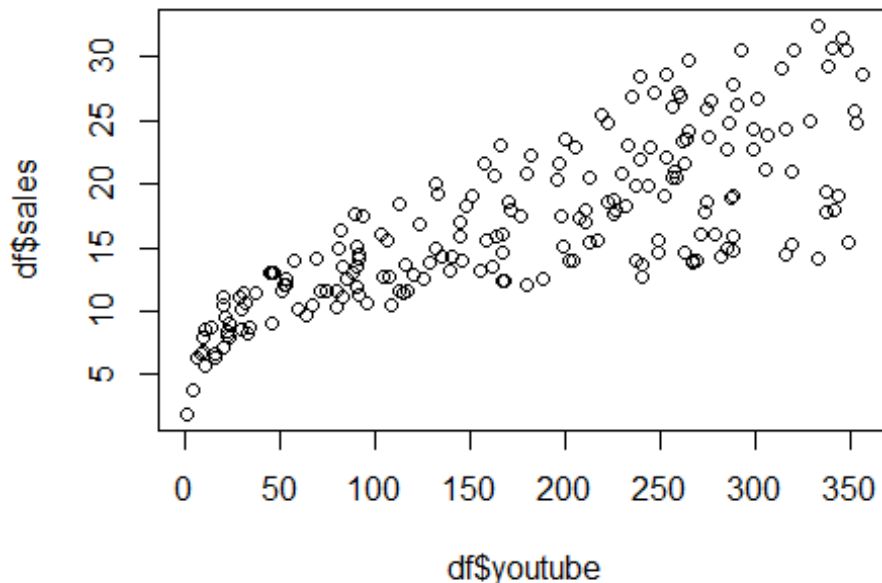


これは**散布図**で、x 軸に youtube、y 軸に sales の値が入っています。ここまで関数に入れる引数は基本的に 1 種類だけでしたが、plot 関数のように複数種類の引数を受け取ることができるものがあります<sup>5</sup>。複数のものを受け取る場合、「x(軸)には youtube ベクトル、y(軸)には sales ベクトル」というように、渡した引数が何かを明示して渡すことができます。また、すべての引数を明示して渡す場合は、それらの順番はそこまで気にしなくても大丈夫です。次のサンプルコードは引数の順番を一つ前のものと入れ替えたものですが、同じ結果が返ってきます。

---

<sup>5</sup> c()関数は複数の要素を受け取りましたが、いずれも「ベクトルの要素」という意味では 1 種類です。対して、plot()関数は x 軸のデータ、y 軸のデータ、グラフのタイトルのように複数種類の引数を受け取ることができます。

```
plot(y = df$sales, x = df$youtube)
```



plot 関数は更に引数を明示して渡すことで、見た目を変更することもできます。引数が多い場合は、読みやすさのために Enter で改行することができます。ただし、コンソールは基本一行ずつコードが実行されてしまうので、上のように改行するときは、

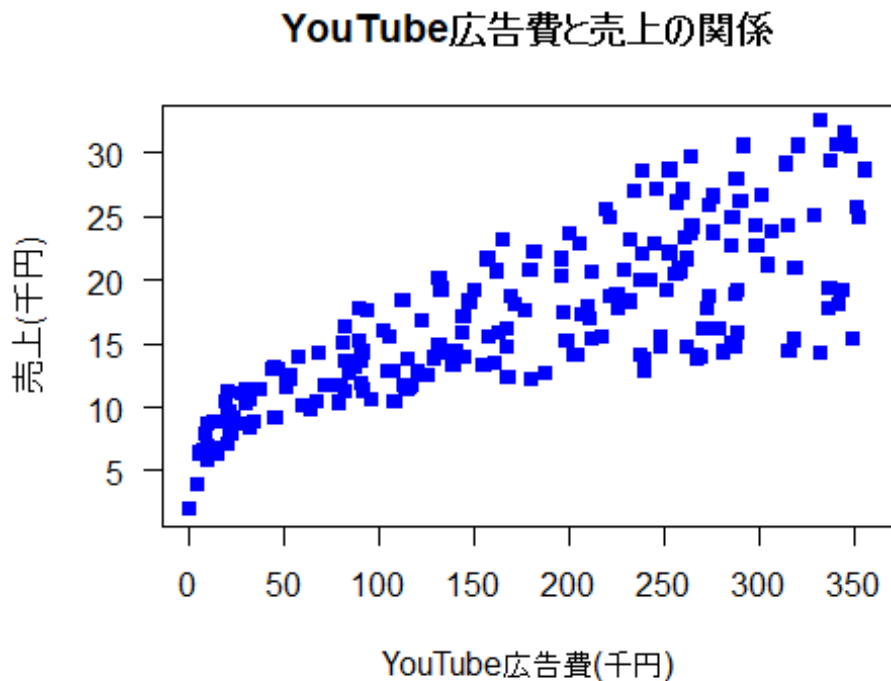
- 改行する前にコンマを入れる
- 終わり括弧 ) は最後に入力する(先に入力するとそこで区切ってしまうので)

という点に気をつけてください。次のコードはその例です。

# 改行するとプラスマークが左に表示されますが、コード入力が続行しているサインです。

```
plot(x = df$youtube, y = df$sales,  
     main = "YouTube 広告費と売上の関係", # タイトル  
     xlab = 'YouTube 広告費(千円)',      # X 軸のラベル  
     ylab = '売上(千円)',                # Y 軸のラベル  
     las = 1,                           # 軸目盛の向き  
     pch = 15,                           # 点の種類  
     col = "blue"                        # 点の色  
)
```





## 1-10. オブジェクトの保存

RStudio 上で作成・使用した文字列や数値、データフレームなどのオブジェクトはまるごと RData 形式のファイルで保存することができます。右上の Environment タブの保存(フロッピー)ボタンを押す場合は、File name を設定し、「Save」で保存してください。

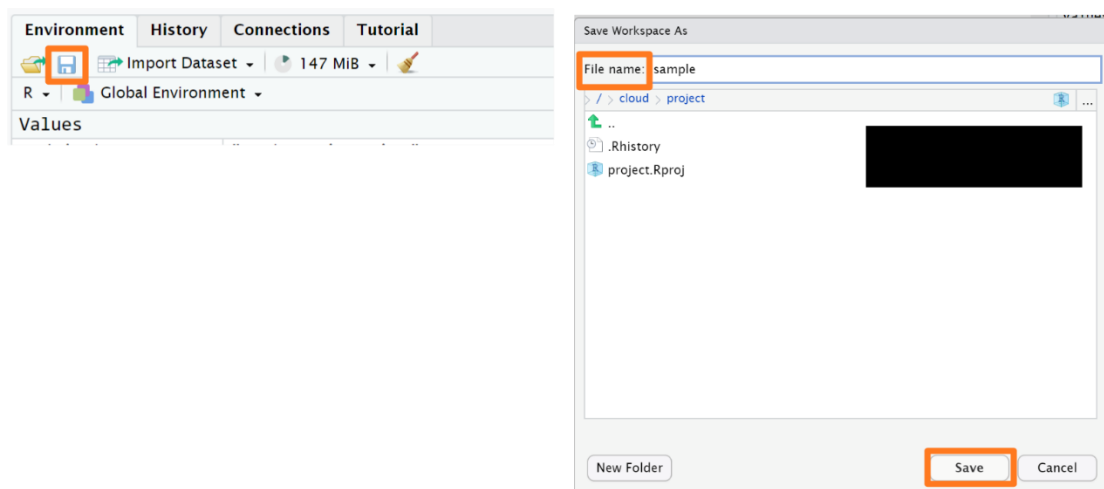


図 7. オブジェクトの保存画面

もしくは、**save.image()**関数でも可能です。

```
# 引数はクォーテーションで囲み、必ず、".RData"を末尾につける。  
# 日本語名でも保存可能ですが、入力切り替えでトラブルになることが多く、  
# また、環境によっては読み込めないこともあるので、なるべく英数字を使うのが望ましいです。  
save.image("sample.RData")
```

保存したファイルは **load()**関数で読み込めば再び使用できるようになります。

```
# 現状では、すでに作成したオブジェクトを読み込んでいる状態なので、  
# これを実行しても特に変化はありません。  
load("sample.RData")
```

また、オブジェクト全体ではなく、データフレーム単体を csv ファイルで保存することも可能です(csv: comma separated value、カンマ区切りテキストデータ。エクセルファイルのようなもの)。csv ファイルで保存することで、エクセルなど別の表計算ソフトでデータを編集することが可能になります。

```
write.csv(df, file = "marketing.csv")
```

グラフは画像として保存することができます。

```
png("sample.png")           # グラフを保存するための画像ファイルを作成  
plot(x = df$youtube, y = df$sales) # 保存したいグラフを記述  
dev.off()                   # 画像ファイルへの入力を終了
```

今回、コンソールに入力してきたコード自体も、R スクリプトファイルという形式で保存することができますが、直接保存することができず、多少複雑なため、今回は割愛します。最後に、Posit Cloud 上で作成したファイルですが、これは画面右下の Files タブからダウンロードできます。まずダウンロードしたいファイルにチェックを入れたあと、ファイルタブ内の歯車マークをクリックして、「Export…」を選択します。すると、zip ファイルとしてダウンロードすることができます。

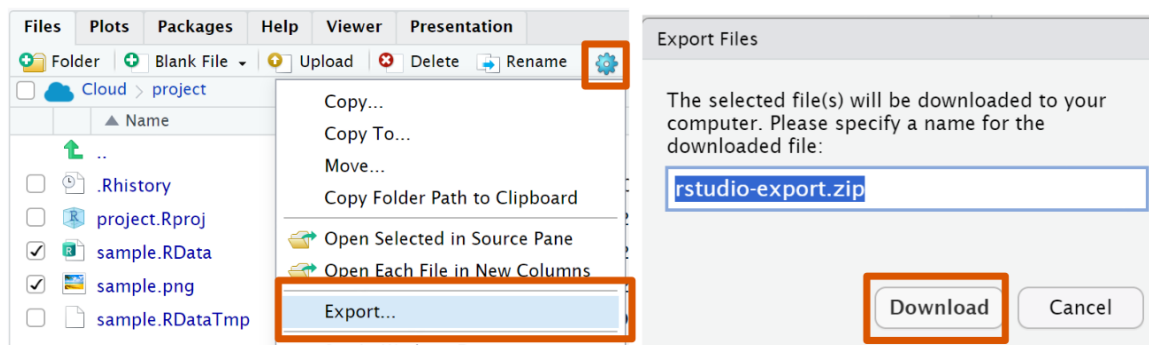


図 8. 作成したファイルのダウンロード

R 体験は以上です。もう少ししっかり勉強したいと思った方は、後述のおすすめ文献をぜひとも読んでみてください。

## 2. 学習用資料

プログラミングでは特に断りもなく、「オブジェクト」や「エンコーディング」など専門用語が出てくることが多々あります。学習する際、そういった知らない単語に出会ったときは、最低限調べ、そのうえで分からなかったとしても、ひとまず先に進めて行くことをオススメします。プログラミングという分野では、頭で理解するより実際にやっていくうちにわかることがたくさんあります。まずは手を動かし、実際にコードを書いてみることで、徐々に理解が深まっていくでしょう。

### 2-1. 事典・ハンドブック 類

#### ■ R クックブック / J. D. Long, Paul Teetor 著 ; 木下哲也訳 (第 2 版)

【書誌 ID= 2004491611】理工学図-東館 2F 図書 007.64||TEE

#### ■ データビジュアライゼーションの基礎 : 明確で、魅力的で、説得力のあるデータの見せ方・伝え方 / Claus O.Wilke 著 ; 小林儀匡, 瀬戸山雅人訳

【書誌 ID=2004551672】理工学図-東館 2F 図書 007.6||WIL

#### ■ R ではじめるデータサイエンス / Hadley Wickham, Mine Çetinkaya - Rundel, Garrett Grolemund 著 ; 大橋真也訳(第 2 版)

【書誌 ID= 2004585616】総合図-A 棟 3 階 学習用図書 007.6||WIC

### 2-2. 最初に読むべき資料

#### ■ R/RStudio でやさしく学ぶプログラミングとデータ分析 / 掌田津耶乃 著

Windows・Mac 両ユーザーに向けて R や RStudio の導入から実践まで丁寧に記述されており、RStudio の機能についても詳しく学べます。本書の「はじめに」でも書かれているように、「プログラミング」「R 言語」「統計」を初めて学ぶ人にうってつけの本です。

【書誌 ID=2004572361】総合図-A 棟 4 階 学習用図書 417||SHO

## ■ はじめての RStudio: エラーメッセージなんかこわくない / 浅野正彦, 中村公亮 著

RStudio の導入からレポートの作成までの流れをいわゆる「文系」に向けて解説しています。R や RStudio の設定部分は本書がやや古いため、当てはまらないかもしれませんが、データ可視化部分などは分かりやすく、参考になります。

【書誌 ID=2004586176】総合図-B 棟 2 階 LS 選書 417||ASA

## 2-3. 実力をつける演習本

---

### ■ R による計量政治学 / 浅野正彦, 矢内勇生共著

tidyverse という、R で分析するならこれ！というパッケージコレクションを使いながら、統計分析に必要な手法が一通り学べます。第 I 部の「リサーチデザイン」は分析を始める前に必要な知識や心構えが学べるので、この部分だけでもおすすめできます。

【書誌 ID=MC00238717】オンライン資料

【書誌 ID= 2004504870】総合図-A 棟 3 階 学習用図書 311.19||ASA

総合図-書庫棟 研究用図書 311.19||ASA

### ■ 社会科学のためのデータ分析入門 上・下 / 今井耕介 著 ; 粕谷祐子, 原田勝孝, 久保浩樹訳

今回の演習で参考にした本。全体を通じて実際の論文で使用されているようなデータを使用しており、実践的に R を使った分析が学べます。演習問題も豊富についていますが、英語のため、多少ハードルはあるかもしれません。

【書誌 ID= MC00119679】上巻 オンライン資料

【書誌 ID= MC00353948】下巻 オンライン資料

【書誌 ID= 2004448533】上巻 総合図-A 棟 3 階 学習用図書 361.9||IMA||1

下巻 総合図-A 棟 3 階 学習用図書 361.9||IMA||2

上巻 総合図-書庫棟 研究用図書 361.9||IMA||1

### ■ R と RStudio による教育テストデータの分析 / 堀 一輝, 福原 弘岳, 山田 剛史 著

『R による計量政治学』と同様に、tidyverse の使い方を学ぶことができます。教育テストデータ分析の一般的な手法も学びたい方はこちらがおすすめです。

【書誌 ID=2004586175】総合図-B 棟 2 階 LS 選書 371.7||HOR

### ■ 前処理大全: データ分析のための SQL/R/Python 実践テクニック / 本橋智光 著

実際のデータというのは往々にして分析に適した形に整っていないことがあります。そのようなデータを扱う上で重要な前処理のテクニックを一通り学べます。

【書誌 ID= 2004488651】総合図-A 棟 3 階 学習用図書 007.6||MOT

## 2-4. その他専門書・学術論文等で注目すべきもの

---

### ■ R ユーザのための RStudio[実践]入門 : tidyverse によるモダンな分析フローの世界 / 松村優哉 [ほか] 著(改訂 2 版)

tidyverse というパッケージ群の使い方にフォーカスした本。グラフによるデータの可視化を説明した 4 章がおすすめです。

【書誌 ID= 2004570514】総合図-A 棟 3 階 学習用図書 007.609||MAT

### ■ 再現可能性のすゝめ : RStudio によるデータ解析とレポート作成 / 高橋康介著

R には R で分析した結果をそのままレポートやプレゼンファイルととして出力することができる R マークダウン(.rmd)というファイル形式があります。本書はその使い方について詳しく言及しています。

【書誌 ID= 2004451656】総合図-A 棟 4 階 学習用図書 417||WON||3

## 2-5. 有用なナビゲートツール：ブックガイド・リンク集

---

### ■ R Markdown (<https://rmarkdown.rstudio.com/>)

### ■ R Markdown: The Definitive Guide (<https://bookdown.org/yihui/rmarkdown/>)

いずれも R マークダウンの使い方について書かれたウェブサイト。辞書や参考書として使えます。

### ■ 私たちの R (<https://www.jaysong.net/RBook/>)

上記に挙げた「R による計量政治学」の著者矢内勇生氏が製作に参加しているウェブサイト。R の使い方にフォーカスしています。

## 2-6. 統計学の入門書

---

統計分析を主目的として使用される「R」というプログラミング言語の特性上、上記で紹介した本はいずれも「R」自体の使い方と、「統計学」という学問の基礎の両方を同時に学習できるものばかりです。両分野がいずれも初心者である人には取り組みやすい一方で、統計手法の実用面に紙幅が割かれているため、統計学の理論的な理解がおろそかになってしまう可能性があります。もし統計学自体の理解に不安があれば、下記の本をご参照ください。いずれも統計学初学者向けに書かれている本です。一読して合いそうなものから始めてみてください。統計学をよりしっかりと学びたい方は、「統計検定 2 級」や「統計的検定」など、他のパスファインダーをご参照ください。これらの本に取り組み、理論面の理解を補うことで、より柔軟に R で分析ができるようになると思います。

■ 完全独習統計学入門 / 小島寛之 著

【書誌 ID= 2003714130】総合図-A 棟 4 階 学習用図書 417||KOJ

■ やさしくわかる統計学のための数学：中学レベルからはじめる! / ノマド・ワークス 著

【書誌 ID= 2004577859】総合図-A 棟 4 階 学習用図書 417||NOM

■ 単位が取れる統計ノート / 西岡康夫 著

【書誌 ID= 2003716830】総合図-A 棟 4 階 学習用図書 417||NIS

総合図-書庫棟 研究用図書 417||NIS

■ はじめての統計学 / 鳥居泰彦 著

【書誌 ID= 2003128028】総合図-A 棟 4 階 学習用図書 417||TOR