

Rブートキャンプ：エクササイズ

Ex.1 ファイル操作の基本

1. デスクトップ上に新しいフォルダを作成しなさい。フォルダ名は `応用心理学2022` とすること
2. コマンドプロンプトを起動し、`cd`コマンドでカレントフォルダを確認しなさい
 - ヒント：「ファイル名を指名して実行」(Win+R)で `cmd` と入力
3. `cd`コマンドを使って、(1)で作ったフォルダに移動しなさい
 - ヒント：フォルダを右クリック→プロパティから、親フォルダの場所を確認できる
4. `cd`コマンドを使って、親フォルダに移動しなさい
5. `cd ~` を入力して実行しなさい

Ex.2 RStudio

1. RStudioから新規Rスクリプトを作成しなさい
 - ヒント：File > New File > R Script
2. (1)で作ったRスクリプトをEx.1で作ったフォルダに保存しなさい。ファイル名は `r_bootcamp.r` とすること
3. 一度RStudio上でファイルを閉じ、(2)で作ったファイルを開き直しなさい
4. エディタ上でRスクリプトに `print('hello')` と入力し、実行しなさい
 - ヒント：実行したい部分を選択して `ctrl(command) + Enter` でスクリプトを実行できる

Ex.3 はじめに覚えておくこと

1. コンソールに `# print('comment')` と入力して実行し、実行結果が **出ない** ことを確認しなさい

In []:

1. エディタ上で `?mean` と入力して実行し、ヘルプが表示されることを確認しなさい

In []:

1. `getwd()` を実行して作業ディレクトリを確認しなさい

In []:

1. `setwd()`関数を使ってEx.1-1で作ったフォルダを作業ディレクトリに設定しなさい

In []:

1. 以下を順に実行しなさい

(1) `test = 1`

In []:

(2) `print(test)`

In []:

(3) `print(TEST)` (エラーが出ることを確認する)

In []:

Ex.4 演算

1. [四則演算] $x = 3, y = 4, z = 5$ として以下の値を求めなさい

In []:

```
x = 3  
y = 4  
z = 5
```

(1) x と y と z の和

In []:

(2) x と y の和を z で割った値

In []:

(3) z の x 乗

In []:

(4) z を x で割った商とあまり(整数除算)

In []:

商

In []:

あまり

1. [数学関数] $x = c(1,2,3,4,5)$ として、以下を実行しなさい

In []:

```
x = c(1, 2, 3, 4, 5)
```

(1) `sum(x)`

In []:

```
sum(x)
```

(2) `mean(x)`

In []:

```
mean(x)
```

(3) `var(x)`

In []: `var(x)`

(4) `sum((x - mean(x))^2) / 5`

In []: `sum((x - mean(x))^2) / 5`

1. [比較演算子] $x=1, y=2$ として以下の計算を実行しなさい

In []: `x = 1`
`y = 2`

(1) `x == y`

In []: `x == y`

(2) `x != y`

In []: `x != y`

(3) `x > y`

In []: `x > y`

(4) `x < y`

In []: `x < y`

(5) `x %in% c(2,4)`

In []: `x %in% c(2,4)`

(6) `y %in% c(2,4)`

In []: `y %in% c(2,4)`

1. [論理演算その1] $x = c(T, T, F, F)$, $y = c(T, F, T, F)$ として、以下を実行しなさい

In []: `x = c(T, T, F, F)`
`y = c(T, F, T, F)`

(1) `x & y`

In []: `x & y`

(2) `x && y`

In []: `x && y`

(3) `x | y`

```
In [ ]: x | y
```

```
(4) x || y
```

```
In [ ]: x || y
```

```
(5) !x
```

```
In [ ]: !x
```

1. [論理演算その2] $x = c(T, T, T)$, $y = c(T, T, F)$ として、以下を実行しなさい

```
In [ ]: x = c(T, T, T)  
y = c(T, T, F)
```

```
(1) all(x)
```

```
In [ ]: all(x)
```

```
(2) all(y)
```

```
In [ ]: all(y)
```

```
(3) any(x)
```

```
In [ ]: all(x)
```

```
(4) any(y)
```

```
In [ ]: any(y)
```

1. [NAとの計算] 以下を実行しなさい

```
(1) 1 + NA
```

```
In [ ]: 1 + NA
```

```
(2) 0 * NA
```

```
In [ ]: 0 * NA
```

```
(3) sum(1,2,NA)
```

```
In [ ]: sum(1,2,NA)
```

```
(4) T & NA
```

```
In [ ]: T & NA
```

```
(5) F & NA
```

```
In [ ]: F & NA
```

(6) T | NA

In []: T | NA

(7) F | NA

In []: F | NA

1. [ベクトルの作成] 以下のベクトルを作成しなさい

(1) 1から5までの整数を順に並べたベクトル

In []:

(2) a,b,c,dのそれぞれの文字を要素とするベクトル

In []:

(3) コロン演算子 : を使って、1から10までの整数を順に並べたベクトル

In []:

(4) コロン演算子 : を使って、8から2までの整数を大きい順に並べたベクトル

In []:

(5) c(1,2,3)を4つ並べたベクトル

In []:

1. [ベクトル演算] $x = 1:5$, $y = c(2,4)$, $z=c(5, 3, 1)$ として、以下の計算をしなさい

In []: x = 1:5
y = c(2, 4)
z = c(5, 3, 1)

(1) x + 1

In []: x + 1

(2) y * 2

In []: y * 2

(3) x + y

In []: x + y

(4) x * z

In []: x * z

(5) `sqrt(x)`

In []: `sqrt(x)`

(6) `mean(y)`

In []: `mean(y)`

(7) `rev(x)`

In []: `rev(x)`

(8) `sort(z)`

In []: `sort(z)`

1. [要素へのアクセス] $x = c('a', 'b', 'c', 'd', 'e')$, $y = 1:6$ として、以下の操作をしなさい

In []: `x = c('a', 'b', 'c', 'd', 'e')`
`y = 1:6`

(1) 添字番号を使って x の3番目の要素を取り出す

In []:

(2) 整数ベクトルを使って x の2番目と5番目の要素を取り出す

In []:

(3) x の1番目の要素をfに変更する

In []:

In []:

(4) 論理ベクトルを使って x の1番目と4番目の要素を取り出す

In []:

(5) y の偶数である要素を取り出す

In []:

(6) y の2番目と3番目以外の要素を取り出す

In []:

Ex.6 関数

1. [関数の作成]以下の関数を作成しなさい

(1) 実行すると'done.'と表示する関数

In []:

In []:

(2) 数値が入力されると、入力値の2倍足す1の値を返す関数

In []:

In []:

1. [制御構文] 以下の間に答えなさい。ただし整数以外の入力値については考慮しなくてよいものとする

(1) 以下の関数を作成しなさい

(1-1) 入力された整数が偶数なら'even', 奇数なら'odd'と表示する関数

In []:

In []:

(1-2) 入力された整数を3で割った余りによって以下の表示をする関数

- 余りが2ならば'two'
- 余りが1ならば'one'
- 余りが0ならば'zero'

In []:

In []:

In []:

In []:

(2) 前問(1)で作成したそれぞれの関数について、for文を使って1から20までの整数値を入力した結果を表示しなさい

In []:

In []:

(3) 入力された数字が3で割り切れるなら'fizz', 5で割り切れるなら'buzz', 3でも5でも割り切れるなら'fizzbuzz'を、それ以外は入力された数字を表示する関数を作成し、1から30までの整数までを入力した結果を表示しなさい

In []:

In []:

1. ★[forとベクトル] Sys.time()関数は現在時刻を取得できる関数である。

処理の終了時刻と開始時刻の差をとることで、処理の実行時間を計測できる。

```
start = Sys.time() # 開始時刻  
  
# 実行時間を測りたい処理  
  
end = Sys.time() # 終了時刻  
result = end - start # 実行時間
```

Sys.time()関数を用いて以下のそれぞれの実行時間を計測し、比較しなさい

(1) 1から100,000までの整数の和をfor文を使って求める

In []:

(2) 1から100,000までの整数の和をsum()を使って求める

In []:

Ex.7 データフレーム

1. [データフレームの操作] excersise フォルダに ex7_1_a.tsv ~ ex7_1_d.tsv が入っている（以下、ファイルA~Dと呼ぶ）。これはある年における講義科目の学生の成績が入っている架空のデータである。学生は10人いて、社会学(sociology)と心理学(psychology)が必修であるため全員が受講し、経済学(economics)と政治学(politics)は選択科目のため受講していない学生もいる。以下の間に答えなさい

(1) ファイルA~Dをデータフレームとして読み込みなさい(名前はdfa~dfdとすること)

In []:

(2) ファイルAはID1~ID5の学生の、ファイルBはID6~ID10の学生の社会学と心理学の成績である。rbind()関数を用いて両者を結合し、新しいデータフレームを作りなさい（名前はdf_abとすること）

In []:

(3) ファイルCは経済学、ファイルDは政治学の成績である。merge()関数を用いて以下の結合結果を確かめなさい。

(3-1) 経済学と政治学の両方を受講している学生のデータのみ使う

In []:

(3-2) 経済学を受講している学生のデータはすべて使い、政治学のみ受講している学生のデータは使わない

In []:

(3-3) 政治学を受講している学生のデータはすべて使い、経済学のみ受講している学生のデータは使わない

一タは使わない

In []:

(3-4) ファイルC, ファイルDに含まれるすべての学生のデータを使う

In []:

(4) 前問(3-4)の結合結果をdf_cdと呼ぶことにする。merge()関数を用いて、df_abとdf_cdを結合したデータフレームを作成しなさい(名前はdf_abcdとする)。ただしすべてのデータを使う(all=T)こと。

In []:

(5) df_abcdについて、学生の成績の平均値を求め、列名を **GPA** として新しい列を追加しなさい

In []:

In []:

(6) df_abcdをtsvファイルとして書き出しなさい

In []:

1. [データフレームと関数] **USJudgeRatings** は弁護士による米国高等裁判所における州裁判官の評価である。なお、各変数の説明は以下の通りである：

```
[,1] CONT Number of contacts of lawyer with judge. # 弁護士と裁判官の接触回数
[,2] INTG Judicial integrity. # 司法の廉潔性
[,3] DMNR Demeanor. # 態度
[,4] DILG Diligence. # 勤勉さ
[,5] CFMG Case flow managing. # 事案管理
[,6] DECI Prompt decisions. # 即決性
[,7] PREP Preparation for trial. # 審理への準備
[,8] FAMI Familiarity with law. # 法律への熟知性
[,9] ORAL Sound oral rulings. # 口頭決定の健全性
[,10] WRIT Sound written rulings. # 書面決定の健全性
[,11] PHYS Physical ability. # 身体能力
[,12] RTEN Worthy of retention. # 留保の適切さ
```

USJudgeRatings をデータセットとして以下の間に答えなさい

(1) psychパッケージをインストールしていないならば `install.packages('psych')` を実行し、インストールしなさい

In []: # 省略

(2) library()関数でpsychパッケージを読み込んだ上でdescribe()関数を適用し、各変数の要約統計量を算出しなさい

```
In [ ]: library('psych')
describe(USJudgeRatings)
```

(3) データセットの行数と列数を求めなさい

In []:

In []:

(4) colMeans()を用いて、各変数について裁判官全体の平均値を求めなさい

In []:

Ex.8 作図

1. [棒グラフと散布図] `USArrests` は1973年の米国における10万人当たりの暴力犯罪の件数および都市人口データである。`USArrests` をデータセットとして用いて以下の間に答えなさい

(1) データセットの先頭5件を抽出し、殺人(Murder)の値を縦棒グラフで表示しなさい

In []:

(2) データセットの末尾5件を抽出し、暴行(Assault)の値を横棒グラフで表示しなさい

In []:

(3) plot()関数を用いて三種類の暴力犯罪(Murder, Assault, Rape)について、すべての組み合わせについて散布図を作成しなさい

In []:

(4) pairs()関数を用いてすべての変数についての散布図行列を作成しなさい

In []:

1. [時系列] `AirPassengers` は1949年から1960年にかけての月別国際航空旅客者数のデータである。

(1) plot()関数を用いて折れ線グラフを作りなさい

In []:

(2) png()関数を用いて(1)で作ったグラフを保存しなさい

In []:

1. [群間データ] `PlantGrowth` は植物の成長に関する実験データである。統制条件と2つの実験条件の形三つの群(group)が用意され、植物の重量(weight)が測定された

(1) table()関数を用いて各条件の標本数を求めなさい

In []:

(2) 条件間を比較する箱ひげ図を作りなさい

In []:

Ex.9 亂数

1. [sample()関数] 1から6の整数から均等な確率でひとつ数字を取り出す試行を繰り返すことを考える（ただし数字の重複は許す）。sample()関数を使って均等な確率で1000回選び出し、以下の間に答えなさい（乱数なので答えは一定でない）

(1) 1が出た回数はいくつか

In []:

(2) 3以下の数字が出た回数はいくつか

In []:

1. [runif()関数] runif()は0から1までの一様分布にしたがう乱数を出力できる関数である。runif()関数を乱数を1000個生成し以下の間に答えなさい

In []:

(1) 0.2以下の数はいくつあるか

In []:

(2) 0.3以上、0.5以下の数はいくつあるか

In []:

1. ★[モンテカルロ・シミュレーション] runif()関数を使って近似的に円周率を求めたい。

(1) runif()関数を用いて、 $0 \leq x \leq 1, 0 \leq y \leq 1$ である点 (x, y) をランダムに1000個生成し、散布図にプロットしなさい

In []:

In []:

(2) 原点 $(0, 0)$ からの距離が1以下である点の数を数えなさい。

In []:

In []:

In []:

(3) (1), (2)の結果から円周率の近似値を求めなさい。また、生成する点の数を1,000,000にしたらどうなるか試しなさい

In []:

- 考え方

- ランダムに打たれた点のうち、原点(0, 0)を中心とする半径1の円の中にある点の数を k とする。
- このとき正方形と扇部分の面積比は $n : k$ に等しい
- 正方形の面積は1である
- 円の面積は πr^2 なので、扇部分の面積は $\frac{1}{4}\pi$ である
- $1 : \frac{1}{4}\pi = n : k$ の関係が成り立つので $\pi = \frac{4k}{n}$

Ex.10 dplyrによるデータハンドリング

1. ToothGrowth はビタミンCの用量と与え方がモルモットの歯の長さに与える影響を調べた実験データである。これをデータセットとして用い、dplyrパッケージを読み込んだ上で以下の間に答えなさい

(1) パイプ演算子を使ってデータセットにhead()関数を適用しなさい

In []:

```
library('dplyr')
```

In []:

```
?ToothGrowth
```

In []:

(2) パイプ演算子とfilter()関数を使って以下の操作をしなさい

(2-1) supp が OJ であった個体を抽出する

In []:

```
(2-2) supp が VC で、 dose が2である個体を抽出する
```

In []:

(2-3) supp が VC である個体、または dose が2である個体を抽出する

In []:

(3) パイプ演算子とselect()関数を使って len 列と supp 列を抽出し、それぞれ「length」「supp_type」とリネームして表示しなさい

In []:

(4) パイプ演算子とarrange()関数を使って、 len 列で昇順にソートした結果を表示しなさい

In []:

(5) パイプ演算子とsummarize()関数を使って以下の操作をしなさい

(5-1) `len` の最大値・最小値・中央値・平均をそれぞれ求める

In []:

(5-2) group_by()を `supp` 列に適用し、群ごとの最大値・最小値・中央値・平均をそれぞれ求める

In []:

Ex.11 stringrによる文字列操作

1. [基本操作] `month.name` は1年の月名のリストである。stringrを用いて以下の間に答えなさい

(1) str_subset()関数を用いて名前に'r'を含む月を抜き出しなさい

In []:

```
library('stringr')
```

In []:

(2) str_detect()関数を用いて各月が名前に'r'を含むかどうかの論理ベクトルを得なさい

In []:

(3) str_length()関数を用いて各月の文字数を求めなさい

In []:

(4) str_subset()関数は正規表現を扱える。正規表現を用いて以下の条件にあてはまる月をそれぞれ求めなさい

(4-1) 名前が'A'で始まる月

In []:

(4-2) 名前が'r'で終わる月

In []:

(4-3) 名前の2文字目が'a'である月

In []:

(4-4) 名前の最後から3番目の文字が'u'である月

In []:

(4-5) 名前の最初の文字が母音である月

In []:

(4-6) 名前の最後の文字が'r'でも'y'でもない月

In []:

(4-7) 名前の末尾が'er'または'ry'で終わる月

In []:

1. [正規表現による抽出]str_extract_all()関数は、文字列の中から正規表現にマッチした部分をすべて抜き出せる。sentence='comuptational social science'として以下の要素を抜き出しなさい

(1)単語(アルファベットの連続)

In []:

```
sentence = 'comuptational social science'
```

In []:

(2) 's'で始まる単語

In []:

(3) 'al'で終わる単語

In []: