18 データ分析プログラミング（1）

花澤楓 学籍番号: 2125242

2023/10/30

1　データ分析のワークフロー

目的に応じて様々な種類のデータ分析があるが、その中でもデータ分析のためのワークフローは普遍的なものである。主に以下の流れの通り。

1. raw
   生データの入手やその保存など
2. build
   データ分析のための、生データを加工・整形する工程。
   - clean
   - transform
   - merge

   などが目的に応じて（分析できるように）必要。
3. analyze
   分析パート。統計分析の前に、データのチェックなどが必要。また、結果の図や表も生成する。
   - check
   - statistical analyses
   - figure
   - table
4. report
   結果をわかりやすく、他人が理解できるよう文書にまとめる。その確認。

データ分析のプログラミング = 料理の指示書作り

プログラミング言語は、命令された内容をそのまま実行してくれるが、気の利いたことはできないし、勝手にそんな挙動がなされても困ってしまう。そのため、分析者は自分が分析したい内容をわかりやすくプログラミング言語に伝達することが必要である。その工程が、料理の指示書作りのようである。以下に示す。

1. 下準備が努力の 8 割

料理をする際には、食材の購入などやレシピの確認など下準備が努力の 8 割と言われている。これはデータ分析でも同じで、かっこいいデータ分析パートはほんの少しで、その前のデータを収集・加工・整形の過程にかかる努力がそのほとんどを占める。

2. 道具を効果的に使う
   みじん切りをする際には包丁ではなく、ミキサーを使えばいいように、データ分析でも `base R` ではなく `tidyverse R` を効果的に使うのが良い。こういった便利な道具が目的ごとに用意されているので、必要に応じて活用することが賢いやり方である。
3. 配膳にも気を配る
   美味しい味付けをしたとしても、配膳が絶望的に下手くそである場合、食べる側の意欲が失われてしまうことがある。データ分析でも全く同じで、どんなにすごい分析結果が出たとしても、論文やレポートを見る側の意欲を削いでしまうほどに見づらい図や表を作ってしまった場合、誰も見てくれない。
4. 家庭・地域のスタイルがある
   関西は薄味、関東は濃い味であること、お茶碗は右側なのか左側に置くのかなど、料理には家庭・地域ごとの文化的な違い（スタイル）が存在する。データ分析でも、分野ごとにお作法が大きく異なる。それを大きく無視したレポートを作成してしまうと、その分野の人にとっては大変読みづらいものとなってしまう。
5. 安全性・衛生に気を配る
   データ分析では、少しのミスによってプログラムが動かなくなってしまうが、動かないうちはまだ良い方で、よくわからないのにプログラムは動いて結果も出力されるといった場合、そのままの結果を使ってしまう危険性がある（特にミスに気づいていないとき）。そのため、プログラムを書く前に、プログラムの全体像やどのような方法で問題を解決するかなどを十分に吟味した上で分析コードを書いていくべきである。
6. 手を動かして、長年かけて技術を磨く
   プロの料理人には、レシピを覚えたからといってすぐになれるものではない。データ分析も同じで、教科書を見るだけでなく、自分で手を動かしてコードを書いてみて初めて理解できるものがあり、それを長年かけて繰り返すことで自らの技術が改善されていく。
7. 素材を活かす
   良い素材（データ）を最大限に活かすには良いデータ分析力が必要である。そのためにも学習が必要。

2　データの種類

様々な種類のデータ（連続型 or 離散型）で溢れているが、データに解釈を与え、それらを分類することでより扱いやすくなる。データの解釈としては 2 つが人間によって考えられている。

1. 量的（quantitative）
   データに単位があり、量的なデータに対しては四則演算ができる。勉強時間やテストの点数、所得や賃金など。
2. 質的（categorical）
   データに単位はなく、分類されているもの。例えば、小学校 1 年、2 年などの学年や赤色青色など。

また、データの種類は以下のように分類できる。

1. 数値型（numeric）
   整数（integer）と小数点以下を含むような実数（double）で構成される。
2. 文字列型（character）
   "" や '' で囲まれたデータを、文字列型という。
3. 論理値型（logical）
   1 or 0 もしくは、TRUE or FALSE で表される。コンピューターは 1, 0 で判断するが、可読性向上のために TRUE or FALSE で書くこともある。
4. 因子型（factor）
   因子型では、カテゴリを順序付け（レベル）することができる。文字列型や数値型を因子型に変換することで、データをより一般的な扱いをすることができる。
5. その他
   - NULL
   - NA
   - Inf
   - 日時データ

また、R におけるデータ構造の種類 3 つを以下に示す。

1. ベクトル
   同一のデータ型の種類を要素に持つもの。ベクトルの中に、数値型と文字列型を同時に含めることはできない。R ではベクトルを文字列型と数値型で構成しようとすると、数値を文字列型に自動変換して変数に格納される。

```
x1 <- c("A", 1, 2)
x1
```

```
## [1] "A" "1" "2"
```

2. リスト（list）
   リストを使用すると複数のデータをまとめることができる。例えば、以下のように使用する。

```
id <- c("A", "B", "C")
x2 <- c(1:10)
mylist <- list(
  id = id,
  x1 = x1,
  x2 = x2)
mylist
```

```
## $id
```

```
## [1] "A" "B" "C"
##
## $x1
## [1] "A" "1" "2"
##
## $x2
##  [1]  1  2  3  4  5  6  7  8  9 10
```

3. データフレーム（data frame）
   テーブル形式でまとめられたリストのこと。リストとは異なり、データフレームの各要素は同じ長さのベクトルである必要がある。

```
mydf <- data.frame(id,x1)
mydf$id
```

```
## [1] "A" "B" "C"
```

また、R には tibble というデータフレームもあり、以下のように使用する。（tribble の使用例）

```
mytibble <- tibble::tribble(~id, ~x1,
                            'A', 1,
                            'B', 2,
                            'C', 3)
mytibble
```

```
## # A tibble: 3 x 2
##   id       x1
##   <chr> <dbl>
## 1 A         1
## 2 B         2
## 3 C         3
```

3　保存と分析に望ましいデータ形式

講義では、データとテーブルについて以下のような定義が与えられている。

- **データ**: 一定の形式を持った情報（JSON 形式、辞書式、画像などの非構造データ等）
- **テーブル**: 表形式に格納されたデータのこと。各変数を列としてまとめ、サンプルサイズだけの観察を行として表現する。列と行の組み合わせから要素が構成される。

Excel でのセルの結合や色付けなど、データは様々な状態で表現される。データ分析の際には、それらを加工・

整形することが求められるが、元の生データは別ファイルとしてそのまま保存することが推奨される。なぜなら、元のデータがどんな状態だったのかを記録することは重要な情報源となるためである。

データ形式の例を以下に 2 つ示す。

1. 関係データ（relational data）
   関係データとは、データの持つ論理構造を反映するように、テーブルを分割して保存する正規化を生データに施したデータのこと。これにより、データの冗長性を避けることができる。例えば、各生徒の名前、テストの成績が、それぞれの名前の Excel ファイルで保存されている場合、各ファイルを merge し、テーブルとして一つのファイルにまとめて保存することで、データ分析ができるようになる。
   また、各生徒の ID がある場合、ID を key として各ファイルを merge する操作をすることもできる。例えば、以下の通り。

```r
data1 <- data.frame(
  ID = 1:3,
  Math = c(90, 85, 88)
)

data2 <- data.frame(
  ID = 1:3,
  English = c(88, 92, 87)
)

merged_data <- merge(data1, data2, by = "ID", all = TRUE)

merged_data
```

```
##   ID Math English
## 1  1   90      88
## 2  2   85      92
## 3  3   88      87
```

2. 整然データ（tidy data）
   tidy data とは、データ形式が wide 型ではなく、long 形式であるものをいい、long 型にデータを加工することでより分析が容易になる。
   tidy data の満たすべき要件は以下の通りである。
   - 1 つの列に 1 つの変数
   - 1 つの行は 1 つの観測データ
   - 1 つのテーブルに 1 つのデータセット