第7回ベイズ統計学・機械学習研究会

Kyohei ITO

2020/10/27

今後の研究会の方針

なんらかの調査分析の仕事を任せていただけることになりそう。

→とりあえず分析できるようにならなきゃ！！

機械学習を中身から理解するのは後回しにして分析ができるようになることを目指します。

なんでRでやるの？

- RStudioが使いやすい
  - 視覚的にわかりやすく変数を管理できる
  - パッケージの読み込みや管理も簡単
  - 参考資料も多数
- 統計に関するパッケージが豊富
  - 行列計算や可視化、簡単な分析機能は標準装備
  - 文書作成も簡単(この資料もRStudio上で作成)
  - その他高度な分析用のパッケージも豊富
- RとPythonの互換性
  - RStudio1.4ではPythonにも対応
    - https://blog.rstudio.com/2020/10/07/rstudio-v1-4-preview-python-support/ (https://blog.rstudio.com/2020/10/07/rstudio-v1-4-preview-python-support/)
  - パッケージを使ったPythonとの連携も可能
  - Pythonでしか使うことができなかった機械学習パッケージが続々対応
    - https://blog.rstudio.com/2020/09/29/torch/ (https://blog.rstudio.com/2020/09/29/torch/)

本研究会でやること

- データ前処理
- データの可視化
- 基本的な機械学習モデルを使用した分析

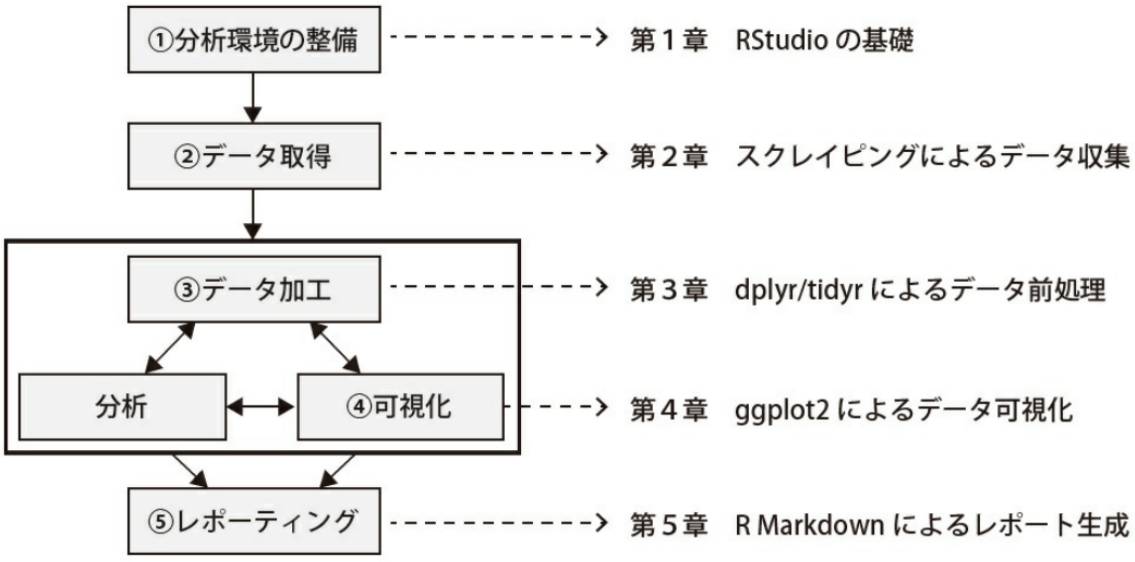


図 0.1 分析のワークフロー

松村ほか（2018）「RユーザのためのRStudio［実践］入門」より

Rの学習に大切(な気がする)こと

何となくR言語で出来る事が想像できること

パッケージがどんな事をするのか何となくわかっているだけで十分

細かくやりだしたらキリがないので、

全部覚えることはできないから、調べながらやるのがいいかなぁと

tidyverseとは

本研究会ではtidyverseと呼ばれるパッケージ群を積極的に使います。

tidyverseは、さまざまな操作を統一的なインタフェースで直感的に行えるtidyなツール群です。

その開発は、HadleyWickham氏をはじめRStudio社の著名な開発者を中心に進められています。

tidyverseが目指すところ

1. 再利用しやすいデータ構造を使う（Reuse existing data structures）
2. 単純な関数を%>%（パイプ）演算子で組み合わせる（Compose simple functions with the pipe）
3. 関数型プログラミングを活用する（Embrace functional programming）
4. 使用者にやさしいデザインにする（Design for humans）

参考になるチートシート

パッケージの使い方が有志によってまとめられています

- R標準関数の説明
  - https://rstudio.com/wp-content/uploads/2016/10/r-cheat-sheet-ja.pdf (https://rstudio.com/wp-content/uploads/2016/10/r-cheat-sheet-ja.pdf)
- データ操作、前処理の基礎

    - https://rstudio.com/wp-content/uploads/2015/09/data-wrangling-japanese.pdf (https://rstudio.com/wp-content/uploads/2015/09/data-wrangling-japanese.pdf)
- ggplot2による可視化
    - https://rstudio.com/wp-content/uploads/2016/10/ggplot2-cheatsheet-2.0-ja.pdf (https://rstudio.com/wp-content/uploads/2016/10/ggplot2-cheatsheet-2.0-ja.pdf)

dplyr/tidyrによるデータ前処理

dplyr/tidyrはデータの加工を得意とするパッケージです。

tidyverse内に含まれているため、一括でロード可能です。

```
library(tidyverse)
```

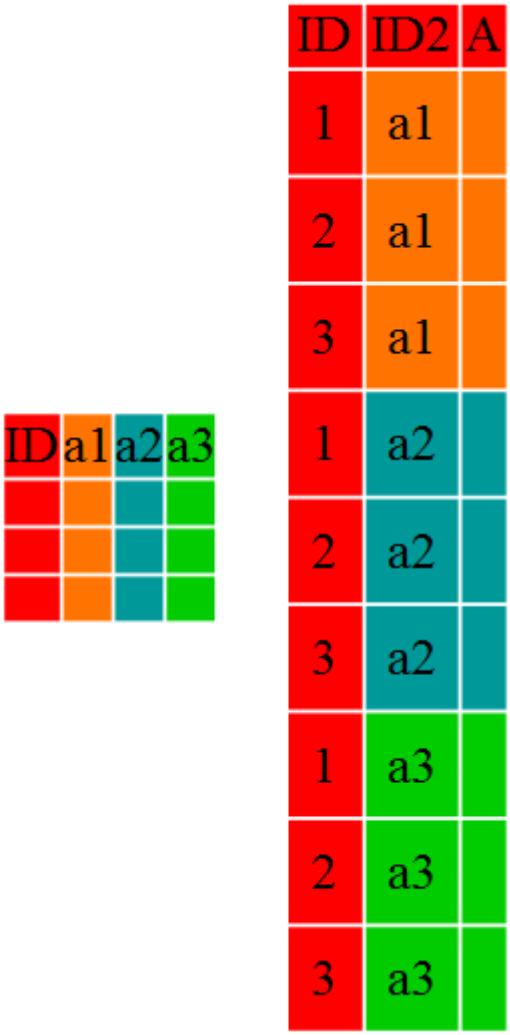
```
## -- Attaching packages --------------------------------------------------------- ti
dyverse 1.3.0 --
```

```
## √ ggplot2 3.3.2     √ purrr   0.3.4
## √ tibble  3.0.3     √ dplyr   1.0.2
## √ tidyr   1.1.2     √ stringr 1.4.0
## √ readr   1.3.1     √ forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------ tidyvers
e_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

tidyrでできる事

- 縦長データへの変形(コンピューターが読み取りやすいデータ)
- 横長データへの変形(人間が読み取りやすいデータ)



https://swcarpentry.github.io/r-novice-gapminder/14-tidyr/ (https://swcarpentry.github.io/r-novice-gapminder/14-tidyr/)　より

やってみよう

下のデータを縦長のデータにしてみましょう 松村ほか（2018）「RユーザのためのRStudio［実践］入門」第3章p171を参考にしてください

```
scores_messy <- data.frame(
名前 = c("生徒A", "生徒B"),
算数 = c( 100, 100),
国語 = c( 80, 100),
理科 = c( 60, 100),
社会 = c( 40, 20),
stringsAsFactors = FALSE
)
view(scores_messy)
```

dplyrでできる事

data.frameに対して

- 行や列の削除(select, filter)
- 行や列の抽出(select, filter)
- 新しい列の追加(mutate)
- 要約(summarize)
- 並び替え(arrange)

などを行うパッケージ

やってみよう

下のmpgというデータセットを読み込み

松村ほか（2018）「RユーザのためのRStudio［実践］入門」第3章p196などを参考に

manufacturer、model、displの3列を抜き出してみましょう

```
mpg
```

```
## # A tibble: 234 x 11
##    manufacturer model    displ  year   cyl trans   drv     cty   hwy fl    class
##    <chr>        <chr>    <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
##  1 audi         a4         1.8  1999     4 auto(l~ f        18    29 p     comp~
##  2 audi         a4         1.8  1999     4 manual~ f        21    29 p     comp~
##  3 audi         a4         2    2008     4 manual~ f        20    31 p     comp~
##  4 audi         a4         2    2008     4 auto(a~ f        21    30 p     comp~
##  5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p     comp~
##  6 audi         a4         2.8  1999     6 manual~ f        18    26 p     comp~
##  7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p     comp~
##  8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p     comp~
##  9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p     comp~
## 10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p     comp~
## # ... with 224 more rows
```

%>%演算子による処理のパイプライン化

これまでの処理を次の関数の第１引数として渡す演算子

途中の結果を一時的に変数に入れてまた次の関数に渡すという作業が省略できます

これもtidyverseに含まれています

例：

```
c(1,2,3,4,5,6,7,8,9,10) %>%
sum()
```

```
## [1] 55
```

上の関数と同じ意味です

```
a <- c(1,2,3,4,5,6,7,8,9,10)
sum(a)
```

```
## [1] 55
```

やってみよう

松村ほか（2018）「RユーザのためのRStudio［実践］入門」第3章p181などを参考に

いままで行った処理をパイプを使って表現してみましょう

参考になるサイト

- github
  - 「R パッケージ名」などで検索するとサンプルコードが出てくる
  - https://github.com/ (https://github.com/)
- Heavy Watal
  - 東北大学 生命科学研究科 特任助教 岩嵜 航先生のページ tidyverseの解説が豊富
  - https://heavywatal.github.io/ (https://heavywatal.github.io/)
- Qiita
  - https://qiita.com/ (https://qiita.com/)