金田　拓

1. 4. 4　Calculations of data frames　　pp. 15~18

☆ 文の長さ＝文の複雑さと捉えたとき、平均を算出する方法　　p. 15

データの**平均**を求めるには、関数 **mean ( )**を使用して計算する。

例：

```
> mean(verbs[verbs$AnimacyOfRec == “animate”, ]$LengthOfTheme)
```

これは、**「データフレーム verbs」の中**で、**「AnimacyOfRec」**の値が**「animate」である項目の、「LengthOfTheme」の平均値**を求めなさい、というコマンドである。

※ == は「左右が一致した場合」、反対に「異なる場合」を表すのは **!=** である。

☆ 関数 tapply ( )を使うと、より柔軟に計算を行うことが可能。　　p. 16

この関数では、左から順に「**計算したい変数**」「**分類に使用する名義尺度**」「**何を計算するか**」の 3 つを指定する必要がある。

例：

```
> tapply(verbs$LengthOfTheme, verbs$AnimacyOfRec, mean)
```

- これは、「**データフレーム verbs における LengthOfTheme**」の「**平均値(mean)**」を、「**データフレーム verbs の AnimacyOfRec の値」別に計算**しなさい、というコマンドである。

  → verbs において AnimacyOfRec の値には、”animate”と”inanimate”の 2 種類があるため、2 つについての計算結果が出力される。

☆ 上記の関数 **tapply ( )**で使用する要素に、**list ( )関数**を指定することで複数の変数を同時に扱うことができる。また、**with( )関数**を使用して、いちいちデータフレーム verbs を指定する手間を省くことができる。（本来はデータ代入が困難なものを格納するために使用するための関数らしい……不確定情報）　p. 16

例：

```
> with(verbs, tapply(LengthOfTheme, list(AnimacyOfRec, AnimacyOfTheme), mean))
```

- これは、「**データフレーム verbs**」において、「**LengthOfTheme**」の「**平均値(mean)**」を、「**AnimacyOfRec 及び AnimacyOfTheme の値」別に計算**しなさいというコマンドである。

→ AnimacyOfRec は animate/inanimate, AnimacyOfTheme も animate/inanimate の 2 種類の値をとりうるので、出力される結果は 2×2 のクロス集計表となる。

* ここからは verbs に代わり、単語と反応時間のデータである heid を用いる *

☆ **aggregate ( )関数を用いて、単語別に平均を求めることができる。 p. 17**
tapply( )関数のように、「**計算したい変数**」「**付随する他の数量データ**」「**何を計算するか**」の 3 つを入力することで、計算が可能。

```
> heid2 = aggregate(heid$RT, list(heid$Word), mean)
> heid2[1:5, ]
```

➢ これは「**データフレーム heid における RT の値**」の「**平均**」を、「**変数 Word**」**の項目別に求めて「heid2」に格納しなさい**、というコマンド。

☆ aggregate ( )関数で**得た結果に、列名をつける colnames( )関数 p. 17**

```
> colnames(heid2) = c("Word", "MeanRT")
```

➢ これは「**heid2 の列名**」を、左から **Word, MeanRT と名付ける**コマンド。

☆ **新たな情報を付け加える p. 17**
例として単語の頻度情報を付け加えたいとき、まず格納する変数を用意する

```
> items = heid[, c("Word", "BaseFrequency")]
```

次に、このままでは約 40 語×26 人という、意味のない繰返しを多数含んだデータなので、1 単語につき 1 項目に、**unique ( )関数**を使って限定する。

```
> items = unique(items)
```

こうして得られたデータフレーム items を、先ほど求めたデータフレーム heid2 とまとめるために、**merge ( )関数**を使って融合する。

- **merge 関数は、共通した列を利用してデータを結合する**。まずデータを受け取るフレーム、その後データを渡すフレームを指定する。次に、共通する列名を by.x（受け取る側）、by.y（渡す側）で指定する。p.17 では両方とも Word である。

```
> heid2 = merge(heid2, items, by.x = "Word", by.y = "Word")
```

別解として、下のような求め方も可能。

```
> heid3 = aggregate(heid$RT, list(heid$Word, heid$BaseFrequency), mean)
> colnames(heid3) = c("Word", "BaseFrequency", "MeanRT")
```

1 行目は、aggregate 関数の「付随するデータ」のスロットに 2 つの変数を挿入したコマンド。2 行目は列に名前をふるコマンドである。

以上のように得られた結果を観察すると、反応時間(MeanRT)が短い、つまり素早く反応する単語ほど、高頻度語であるということが分かる。