金田　拓

1. 3　Accessing information in data frames　pp. 6~8 (in 6~10)

☆　**行と列を指定してデータフレームを読み込む方法　（行＝横、列＝縦）**

カンマの前で行、カンマの後で列を指定する。

> verbs[1, 5]　というコマンドは、解釈すると

>	verbs	[	1	,	5	]
>	データフレーム	[	行を指定	,	列を指定	]

つまり、**「verbs というデータフレーム」から、「1 行目」で「5 列目」の値を取り出しなさい**、という指示を出していることになる。

☆　角括弧[,]で行、列を指定しない場合、全てのデータが出力される。つまり

- 全く制限をかけない場合　：>verbs[,]　：データの全ての値を出力
- 行のみ制限をかける場合　：>verbs[1,]　:データの 1 行目の値を全て出力。
- 列のみ制限をかける場合　：>verbs[, 5]　:データの 5 列目の値を全て出力
- 行・列ともに制限をかける場合　：>verbs[1, 5]　：1 行目で 5 列目の値を出力

☆　**行や列を指定する際には、数字でなく文字列も用いることができる。**

例えば verbs には RealizationOfRec, Verb,などの列ラベルがついているので、verbs[, 2]と入力しても、verbs[, ”Verb”]と入力しても同じ結果、つまりこの全データの 2 列目の値が得られる。日本語の文字列対応。大文字、小文字は区別される。文字列を使用する場合はダブルクォーテーションマークで文字列を囲むこと。

このようにして取り出したデータは、好きな名前の変数に代入が可能である。例えば

> row1 = verbs[1,]

と入力した場合、verbs[1,]の値、つまりデータフレームの 1 行目が row1 に格納され、以降 row1 はデータフレーム 1 行目の値としてコマンドに使用することができる。

他にも例えば列データに関しては、$関数を使用して> verbs$LengthOfTheme　で、>verbs[,5]と同じ結果を出力することが可能となっている。($関数では数字は使えない模様）

☆　**個々の要素を抽出する**

データの一部の要素を抽出したいとき、上記のように代入した後、角括弧で[1]や、[RealizationOfRec]などのように、出力する要素を指定することができる。

例えば、先ほど 1 行目を代入した row1 の RealizationOfRec の値を取り出したい場合、row1[“RealizationOfRec”]とコマンドに入力すると望んだ結果が得られる。

☆　**ベクトルを作る（同じ型のデータをまとめたものを作る）**

c ( )：複数個のデータをくっつける関数　> rs = c(638, 799, 390, 569, 567)と入力すると、rs に括弧内の値が格納され、コマンドで数字を入力しなくても、rs で代用することができる。

関数 c ( )以外にもベクトルを作る方法は多々あり、例えばコロン(:)関数を使ってベクトルを作ることができる。これは波ダッシュ(〜)と脳内変換すると理解しやすい。

例	１：５	1 から 5 までの、間隔 1 の連続データ（１～５）
	５：１	5 から 1 までの、間隔-1 の連続データ（５～１）

これを応用して、データ抽出範囲を選択することができる。

> verbs [rs, 1:3]と入力すると、1~3 列目、つまり RealizationOfRec, Verb, AnimacyOfRec の値が rs(638, 799, 390, 569, 567)番のデータについて出力されることとなる。

これを文字列で入力すると、verbs[rs, c(“RealizationOfRec”, “Verb”, “AnimacyOfRec”)] となる。どちらで入力しても出力結果は同じ。