

1.4.3 Extracting contingency tables from data frames

* 分割表の作成：因子レベルの組み合わせをクロス表にする

たとえば、animate recipients が NP のものはどれくらいあるかを調べるとき

RealizationOfRec と AnimacyOfRec はそれぞれ2つのレベル：関数 `levels()` で示される

```
> levels(verbs$RealizationOfRec)
```

```
> levels(verbs$AnimacyOfRec)
```

`xtabs()` で RealizationOfRec と AnimacyOfRec のクロス表 → 4つのセルからなる表

```
> xtabs(~RealizationOfRec + AnimacyOfRec, data = verbs)
```

`xtabs()` の最初の値は公式

一般的な構造：従属変数 ~ 予測変数 1 + 予測変数 2 + ... ~="depends on"

※ dependent variable (従属変数) 以外の変数は、independent variables (独立変数) とよく
言われるが、predictor (予測変数) がより適切な語

Bresnan et al. (2007) 従属変数：the realization of the recipient

予測変数：他のすべての変数

* 分割表を作成するときには、従属変数はなく、1つの変数が他の変数で説明できるかどうかは関係なく、分割表で数の分布を見ることができるので、`xtabs()` の公式は ~ の左に何も無い。

* 2つ以上の因子のあるクロス表の作成

```
> verbs.xtabs = xtabs(~ AnimacyOfRec + AnimacyOfTheme + RealizationOfRec, data = verbs)
```

```
> verbs.xtabs
```

2つの 2 × 2 の分割表の形で示される 3次元の分割表を出力

Animate themes はごく稀なので、inanimate themes を持つ節に限定して見てみる。

動詞の行に条件をつけて実行

```
> verbs.xtabs = xtabs(~ AnimacyOfRec + RealizationOfRec, data = verbs, subset =  
AnimacyOfTheme != "animate") # != は = でないことを示す
```

```
> verbs.xtabs
```

* 分割表を比率表に：それぞれのセルをセルの合計（データフレームの総数）で割る

合計：関数 `sum()` `> sum(verbs.xtabs)`

inanimate themes を持つ行の数と等しいことは関数 `nrow()` で検証

分割表を合計で割る `> verbs.xtabs/sum(verbs.xtabs)`

%を出す `> 100 * verbs.xtabs/sum(verbs.xtabs)`

* 分割表の行、列に関して割合を計算：関数 `prop.table()`

2番目の値が1のときは、行に関して相対頻度を計算 `> prop.table(verbs.xtabs, 1)`

2番目の値が2のときは、列に関して相対頻度を計算 `> prop.table(verbs.xtabs, 2)`