Analyzing Linguistic Data 2008/12/11 野村真理子

4. A numerical vector and a factor: analysis of variance

＊因子と対応のある数値ベクトルの分析の仕方

Ratings のデータフレームの中の mean familiarity と語の Class を考える

```
> ratings[1:5, c("Word", "meanFamiliarity", "Class")]
```

One-way analysis of variance（一元配置分散分析）

関数 lm( )を使用：animal と plant を表す名詞の間で mean familiarity に差があるかの検定

```
> summary(lm(meanFamiliarity ~ Class, data = ratings))
```

２つの *p* 値は非常に有意な結果を示していて、２群間の平均の差は有意であると推測するかもしれないが、もう少し深く調べてみる。→Class は因子で数値変数ではない。

＊lm( )が因子 Class に関して行っていること

因子の水準を 1 以上の数値ベクトルに再コード化して計算する

Class は 2 つの水準→animal: 0, plant: 1　数値ベクトル→Classplant という名前

＊**Dummy coding**:因子の水準の数値への再コード化

Dummy coding には多くの種類のアルゴリズムがある（help page: contr.treatment ( )）

この本で用いられる dummy coding: **treatment coding**（R では自動処理）

➢データフレームに treatment dummy codes を追加（手動での例）

Classplant というベクトルの中に、plant に 1、animal に 0 という dummy codes を追加

```
> dummy = ratings[, c("Word", "meanFamiliarity", "Class")]
> dummy$Classplant = 1
> dummy[dummy$Class = = "animal", ]$Classplant = 0
> dummy[1:5, ]
```

・1，0 がどちらの因子の水準に割り当てられるかは重要ではなく、R の決定はアルファベット順に基づいている。→animal が **default (reference level)**

・R は dummy ベクトルに non-default の水準が後に来る名前をつける→Classplant

➢Class の代わりに予測変数として Classplant を持つ dummy に lm( )を実行

→p.102 の上と同じ結果

```
> summary(lm(meanFamiliarity ~ Classplant, data = dummy))
```

結果の表：intercept は default の水準 animal の群平均（familiarity の平均）

```
> mean(ratings[ratings$Class = = "animal", ]$meanFamiliarity)
> coef(ratings.lm)[1]
```

t 値と *p* 値は、animal の群平均 3.5122 が 0 と有意差があるか→明らかにある

＊2 番目の係数 0.8547

Plant と animal の群平均の差を表している。

Plant を表す名詞の群平均: 3.5122 (animal の平均)+0.8547=4.3669

t 検定は、0.8547 が統計的に有意であることを示している→２群間の平均には有意差がある

＊ここで得られた *t* 値、*p* 値は plant と animal の familiarity rating の分散を等しいと扱う t 検定の値と同じである。

> t.test(animals$meanFamiliarity, plants$meanFamiliarity, var.equal = TRUE)

＊t.test( )は 2 群の平均の比較に限られるが、lm( )は２つ以上の水準を持つ因子に適用できる。

オランダ語の 285 の動詞についてデータセット auxiliaries を考える。

> head(auxiliaries)

AUX: 動詞に対する適切な完了時制に対する auxiliary

２つの auxiliary: *zijn* (“be”), *hebben*(“have”)

VerbSynsets: WordNet の中である動詞が出現している動詞の synset の数

Regularity: regular vs. irregular

＊synset の数が auxiliary と有意差があるかを検定

> auxiliaries.lm = lm(VerbalSynsets ~ Aux, data = auxiliaries)

Aux は動詞の synset の数の違いを説明するのに役立つかどうかを考える：関数 anova( )

> anova(auxiliaries.lm)

結果：3 種類の動詞に対して synset の数の平均には有意差がある。

→平均の差が、*hebben - zijn, hebben - zijnheb, zijn – zijnheb* のどれにあるのかは特定しない

→どれに差があるのか summary から探る

> summary(auxiliaries.lm)

＊summary より

Default(reference level)は *hebben* であると推測

3.4670＝*hebben* の群平均

0.5997=*hebben* と *zijn* の群平均の差→*zijn* の平均＝3.4670+0.5997

1.6020=*hebben* と *zijnheb* の群平均の差→*zijnheb* の平均＝3.4670+1.6020

hebben と *zijn* をとる動詞の平均には有意差なし

hebben と *zijnheb* をとる動詞の平均には有意差あり

＊この例では１つ比較が抜けている：*zijn* vs.*zijnheb*

１つの因子に３水準以上あると、表に現れない比較がより多くなる。

表には default の水準を含む対の比較しか載らない。