

24日目：データの操作（群分け）

本日は、群分けにかかるデータ操作をやってみようと思います。

データは、sam2.csv を用いてやります。13日目に作成した **xx** というファイルを使ってみます。head で見ると、以下のようになっていたものです。

```
> head(xx)
  ID. 性別 学年 専攻 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17 b18 b19
1 10003    2    3    1  4  4  2  4  3  3  4  2  3  4  1  4  4  3  3  4  3  1  4
2 10004    2    3    1  4  3  2  4  3  1  2  2  1  1  2  3  2  3  2  1  2  1  2
3 10010    2    3    1 NA  4  1  4  4  1  4  1  4  4  1  4  4  4  4  4  4  1  4
4 10018    1    2    1  2  4  1  4  3  1  2  1  3  3  2  4  3  3  3  4  3  1  3
5 10019    1    2    1  3  2  3  4  2  3  4  1  3  2  3  2  3  2  2  2  3  2  3
6 10020    2    3    1  4  4  1  4  3  1  4  2  3  4  2  4  3  3  3  4  3  1  3
  b20 br20 to.f1 to.f2 to.f3
1    1    4    29    18    7
2    2    3    19    15    6
3    1    4    32    NA    4
4    3    2    26    13    5
5    4    1    23    11    11
6    2    3    28    16    5
```

まずは、to.f1 の得点を使って、上位群、下位群を作ってみます。

平均値で分けるなら、もちろん平均値が必要です。計算してみると、24.34 ですから、これを基準にします。やり方はいくつもあると思いますが、簡単なのは **ifelse** を使うものかなと思います。gr.f1 という新しい変数を作り、下位群には1を、上位群には2を割り振ってみます。

```
xx$gr.f1 <- ifelse(xx$to.f1 < 24.34, 1, 2)
```

ifelse は、18日目に取り上げた自作関数の中でも紹介しました。「もし●が真ならば△せよ、●が真でなければ（偽であれば）×せよ」というものです。カッコの中が3つに分けられていて、最初が●、2番目が△、3番目が×に対応しています。つまり、**xx\$to.f1 < 24.34** が真ならば1を、偽ならば2を **xx\$gr.f1** に入れなさいということです。

もちろん、

```
xx$gr.f1 <- ifelse(xx$to.f1 <= 24, 1, 2)
```

とか

```
xx$gr.f1 <- ifelse(xx$to.f1 > 24.34, 2, 1)
```

としても同じ結果になります。

なお「 \leq 」や「 \geq 」の場合は、上の例のように不等号と等号を並べるのですが、不等号を先に書かないとエラーになるようです。つまり「 \leq 」はOKですが、「 $=<$ 」はNGです。また、以前にも触れましたが「等しい」場合は、「 $=$ 」ではなく、2つ並べる「 $==$ 」です

ちゃんと変換しているかどうかを確認するには、たとえば `describe.by` を使って…

```
describe.by(xx$to.f1, xx$gr.f1)
```

これで各群の最小値と最大値をチェックすればよいわけです。

3群以上に分ける場合も、`ifelse` を重ねてやれば OK です。たとえば、22 点以下、23 から 26、27 点以上に、それぞれ 1、2、3 を割り当てたないなら…

```
xx$gr2.f1 <- ifelse(xx$to.f1 <= 22, 1, ifelse(xx$to.f1 <= 26, 2, 3))
```

こんな書き方になります。よくわからなければ、

```
xx$gr.f1 <- ifelse(xx$to.f1 <= 22, 1, 2)
```

と比べてみるとわかるかもしれません。この場合の、最後の 2 の所に `ifelse(xx$to.f1 <= 26, 2, 3)` が入っている形式です。

この群分けは、以下のようにも書けます。

```
xx$gr3.f1 <- cut(xx$to.f1, breaks=c(-Inf, 22, 26, Inf), labels=c(1, 2, 3), right=TRUE)
```

`cut` という関数を使った群分けですが、カッコの中は最初に変数名、次に `breaks=c(-Inf, 22, 26, Inf)` で分割する点を指定します。`-Inf` と `Inf` は最小と最大の点を意味し、その間の区切る点を入力して指示します。`labels=c(1, 2, 3)` は、区切られたそれぞれの群にどのような名前を付けるかを指定します。ここでは、「1」「2」「3」という名前を付けています。最後の `right=TRUE` は、`breaks=` で指定された区間の右側の値を「含む」という意味です。`right=FALSE` にすると、「含まない」になります。今回は、22 点以下、26 点以下という指示になるので、`right=TRUE` としておきます。

これで先と同じ 3 群分けができます。しかし、この 2 つは大きく違う点があります。それは前者は連続変数としての 1、2、3、後者はカテゴリ変数としての 1、2、3 が結果として返されるということです。`cut` という関数を使った群分けは、分散分析をやったときの独立変数の型、`factor` 型を作成します。ワークスペースブラウザなどで確認してみてください。逆に言えば、`ifelse` 使って群分けしたものを独立変数として分散分析を行う場合には、`factor` 型に変換する必要があります。

なお `summary` などですべてを見てみるとわかりますが、前者なら平均値などを計算してくれます。しかし後者はカテゴリ変数なので、平均値などは計算してくれません。

次に、2 つの変数を用いて群分けをすることをやってみます。性と `to.f1` の平均を用い

て、男性の `to.f1` 上位群を1、女性の `to.f1` 上位群を2、男性の `to.f1` 下位群を3、女性の `to.f1` 下位群を4とする、4群に分けてみます。これもいくつかやり方はあるような気がします、たとえば…

```
xx$gr4.f1 <- ifelse(xx$to.f1 > 24.34 & xx$性別 == 1, 1, ifelse(xx$to.f1 > 24.34 & xx$性別 == 2, 2, ifelse(xx$to.f1 < 24.34 & xx$性別 == 1, 3, 4)))
```

この書き方だとわかりにくいでしょうから、形を整えてみます。

```
xx$gr4.f1 <- ifelse(xx$to.f1 > 24.34 & xx$性別 == 1, 1,  
  ifelse(xx$to.f1 > 24.34 & xx$性別 == 2, 2,  
  ifelse(xx$to.f1 < 24.34 & xx$性別 == 1, 3, 4)))
```

これだと少しはわかりやすいのではないのでしょうか。先に `ifelse` を使って3群に分けましたが、同様にして4群に分けています。条件の部分が `xx$to.f1 > 24.34 & xx$性別 == 1` などとなっていますが、`to.f1` が平均値以上 (`xx$to.f1 > 24.34`) かつ、男性 (`xx$性別 == 1`) の場合であれば、これを「&」でつないでおきます。ちなみに「または」の場合は「|」（1でもLの小文字でもなく、縦棒です）でつなぎます。

群分けがうまくいっているかどうかは、

```
lis <- c("性別", "to.f1")  
xxlis <- xx[  
describe.by(xxlis , xx$gr4.f1)
```

などで確認しておきましょう。

ちょっと余談ですが、`ifelse` を使って一つだけ。ある変数の一部のデータだけを置き換えたい場合にも `ifelse` は使えます。たとえば性別の2を、すべて3に置き換えるなら…

```
xx$性別2 <- ifelse(xx$性別 == 2, 3, xx$性別)
```

性別が2でない場合の部分（カッコ内の最後の部分）には、もとのファイル名、変数名をいれておけば、そのままにしておいてくれます。