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 x というファイル（3 つの因子の得点を追加した後のもの）を使ってみます。head で見ると，以下のようになっているものです。

```
> head(x)
   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 b20 br20 total_f1
1 10003    2    3    1  4  4  2  4  3  3  4  2  3   4   1   4   4   3   3   4   3   1   4   1    4       21
2 10004    2    3    1  4  3  2  4  3  1  2  2  1   1   2   3   2   3   2   1   2   1   2   2    3       14
3 10010    2    3    1 NA  4  1  4  4  1  4  1  4   4   1   4   4   4   4   4   4   1   4   1    4       24
4 10018    1    2    1  2  4  1  4  3  1  2  1  3   3   2   4   3   3   3   4   3   1   3   3    2       19
5 10019    1    2    1  3  2  3  4  2  3  4  1  3   2   3   2   3   2   2   2   3   2   3   4    1       18
6 10020    2    3    1  4  4  1  4  3  1  4  2  3   4   2   4   3   3   3   4   3   1   3   2    3       21
  total_f2 total_f3
1       18        7
2       15        6
3       NA        4
4       13        5
5       11       11
6       16        5
>
```

まずは，total_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，18.39 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。x に gr_f1 という新しい入れものを作り，total_f1 下位群には 1 を，上位群には 2 を割り振ってみます。

```
x$gr_f1 <- ifelse(x$total_f1 < 18.39, 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，x$total_f1 < 24.34 が真ならば 1 を，偽ならば 2 を x$gr_f1 に入れなさいということです。

もちろん，

```
x$gr_f1 <- ifelse(x$total_f1 <= 18, 1, 2)
```

とか

```
x$gr_f1 <- ifelse(x$total_f1 > 18.39, 2, 1)
```

としても同じ結果になります。

なお「≦」や「≧」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「<=」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，記号 1 つの「=」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 describeBy を使って…

```
describeBy(x$total_f1, x$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でしょう。

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，16 点以下，17 から 20，21 点以上に，それぞれ１，２，３を割り当てたないなら…

```
x$gr_f1_2 <- ifelse(x$total_f1 <= 16, 1, ifelse(x$total_f1 <= 20, 2, 3))
```

こんな書き方になります。よくわからなければ，

```
x$gr_f1 <- ifelse(x$total_f1 <= 16, 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(x$total_f1 <= 20, 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。

```
x$gr_f1_3 <- cut(x$total_f1, breaks=c(-Inf, 16, 20, 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。

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。性別と total_f1 の平均を用いて，男性の total_f1 上位群を１，女性の total_f1 上位群を２，男性の total_f1

下位群を３，女性の total_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が，たとえば…

```
x$gr_f1_4 <- ifelse(x$total_f1 > 18.39 & x$性別 == 1, 1, ifelse(x$total_f1 > 18.39 & x$性別 == 2, 2, ifelse(x$ total_f1 < 18.39 & x$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。

```
x$gr_f1_4 <-
 ifelse(x$total_f1 > 18.39 & x$性別 == 1, 1,
 ifelse(x$total_f1 > 18.39 & x$性別 == 2, 2,
 ifelse(x$total_f1 < 18.39 & x$性別 == 1, 3, 4)))
```

これだと少しはわかりやすいのではないでしょうか。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 x$total_f1 > 18.39 & x$性別 == 1 などとなっていますが，total_f1 が平均値以上（x$total_f1 > 18.39）かつ，男性（x$性別 == 1）の場合と指定したいのであれば，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，

```
t_list<- c("性別","total_f1")
xtest <- x[t_list]
describeBy(xtest, x$gr_f1_4)
```

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…

```
x$性別改 <- ifelse(x$性別 == 2, 3, x$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。