データの一部にアクセス（抽出機能）

データの一部にアクセスするために必要な比較演算子と論理演算子は次の通り。

比較演算子

論理演算子

意味	記述	意味	記述
より大きい	>	否定(not)	!
以上	>=	かつ(and)	&
未満	<	または(or)	
以下	<=	排他的論理和	xor
等しい	= =	「かつ」には&&, 「または」には があるが使い方は少し異なる。	
等しくない	!=		

データの一部にアクセス（抽出機能）

【実習3】

- (1) 男子のみの抽出
- (2) 東京都出身を抽出
- (3) 英語が60点以上を抽出
- (4) 女子で国語が60点以上を抽出

```
(1) > data1[data1$gender == 1, ]
```

```
(2) > data1[data1$region == "tokyo", ]
```

```
(3) > data1[data1$English >=60, ]
```

```
(4) > data1[data1$Japanese >=60 & data1$gender ==2, ]
```

データの一部にアクセス（抽出機能）

【実習3】

- (1) 男子のみの抽出
- (2) 東京都出身を抽出
- (3) 英語が60点以上を抽出
- (4) 女子で国語が60点以上を抽出

【補足】 attach関数の活用

attach(data1)を活用すると、データセットの変数、この場合はregionを直接、サーチパスできる（data1\$regionとする必要がない）ので便利である。

しかし、もしregionという名前がこのデータフレーム以外で使用されていた場合は変数名同士で衝突してしまい不都合が起こる。このような場合は**detach(data1)**としてパス名を除去するか、Rを再起動する。

【実習3】でattach関数を使う場合、まず **> attach(data1)** と入力してから

(1) **> data1[gender==1,]** (2) **> data1[region=="tokyo",]**

(3) **> data1[English>=60,]** (4) **> data1[Japanese>=60 & gender==2,]**

と実行すれば、結果を得ることができる。

データの一部にアクセス（抽出機能）

【実習4】

(1) 東京出身の国語の合計点	(2) 埼玉出身の英語の平均点
(3) 数学が50点以上の人数	(4) 神奈川県出身で英語が80点未満の人数
(5) 三教科の合計が180点以上の人数	

解 (1)～(4)に対しては **> attach(data1)** と入力し実行する例を下記に示す。

- ```
(1) > sum(data1[region=="tokyo",]$Japanese)
(2) > mean(data1[region=="saitama",]$English)
(3) > dim(data1[Math>=50,])
(4) > dim(data1[English<80 & region=="kanagawa",])
(5) > dim(data2[data2$mark>=180,])
```

◎章末の練習問題をやってみよう

---

## データの書き込み

新たに三教科合計の欄を追加した表（data2）をsheet2として保存する。

【実習5】 "data2"をsheet2.csvで保存（Hドライブに保存）

```
> write.csv(data2, "H:sheet2.csv", row.names=FALSE)
```

row.name=FALSEとすると最左列のヘッダを付けない。

データに欠損値がある場合 【発展】 NA:not available

```
> data3 <- read.csv("J:/R/sheet3.csv", header = TRUE)
```

```
> data3
```

	gender	region	name	Japanese	Math	English
1	1	tokyo	aoki	23	56	65
2	2	tokyo	ehara	43	45	44
3	2	saitama	kato	11	97	76
4	1	chiba	kayama	56	32	23
5	2	chiba	kinoshita	89	41	54
6	1	kanagawa	kojima	58	78	79
7	2	saitama	saito	54	45	26
8	1	saitama	sakamoto	33	55	43
9	2	USA	suzuki	NA	NA	NA
10	2	kanagawa	sugiyama	33	31	44

データに欠損値がある場合【発展】

```
> apply(data3[4:6], 1, sum, na.rm=TRUE) # 欠損値を無視した列の部分和

[1] 144 132 184 111 184 215 125 131 0 108 169 181 253 154 204 187 180
168 182 178 145 188 265 113 188 164 189 203 223 193

> apply(data3[4:6], 2, sum, na.rm=TRUE) # 欠損値を無視した行の部分和
Japanese Math English
 1551 1827 1683

> na.omit(data3) # 欠損値データの削除
```