

## Rでのデータの視覚化(1)

われわれ人間は五感を通じて情報を得る。そのなか大半は視覚によるものである。データをグラフで表現することは、データのなかに潜んでいる情報を視覚的に考察する手助けとなる。Excelのような表計算ソフトには多様なグラフ作成機能がある。Rで計算加工したデータをテキストファイルとして出力し、Excelの機能を用いてグラフを作成することもできるが、直接Rでグラフを作成することが便利な場合もある。またExcelで作成できないグラフをRで簡単に作成することもできる。

### 1. 棒グラフ

グラフを作成するためにはデータが必要である。Rのなかには、多くのデータセットが用意されている。Rに用意されたデータを用いるためには、保存されたデータをRに呼び出す必要がある。データを呼び出す関数はdataである。

VADeathsというデータは、年齢50~74を5段階にわけ、その死亡率を田舎(rural)、都会(urban)のそれぞれ男(male)女(female)別に分けた5行4列の行列型データである。このデータをRに呼び出すには次のコマンドを実行する。さらに、コンソールにVADeathsと入力し、[Enter]キーを押すとデータが返される。

```
>data(VADeaths)
>VADeaths
```

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

このデータを用いて、棒グラフを作成してみる。棒グラフの作成は、関数barplotを用いる。次のようにコマンドを入力し実行すると、図1のような棒グラフが作成される。

```
>barplot(VADeaths)
```

図1は1列のデータを1本の棒に表示している。ここで、各列のデータの順序と棒グラフにおけるデータ順序は逆になっていることに注意されたい。つまり第1行のデータを棒の最も下の方に、第5行のデータを最も上の方に配置している。

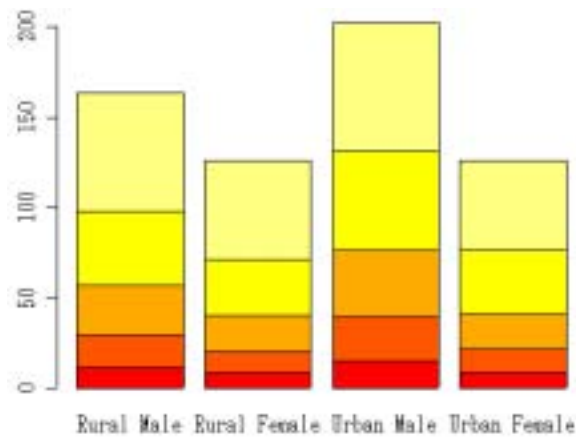


図1 棒グラフの例1

次のコマンドを実行すると、図2のような棒グラフが作成される。ここの `beside` は棒グラフを横に並べるかどうかを指定する引数である。引数 `beside = TRUE` にすると、各列のデータを行ごとに棒を横に並べて作成する。引数 `beside` を用いないか、あるいは `beside = FALSE` にすると図1のような図が作成される。

```
>barplot(VADeaths,beside=TRUE)
```

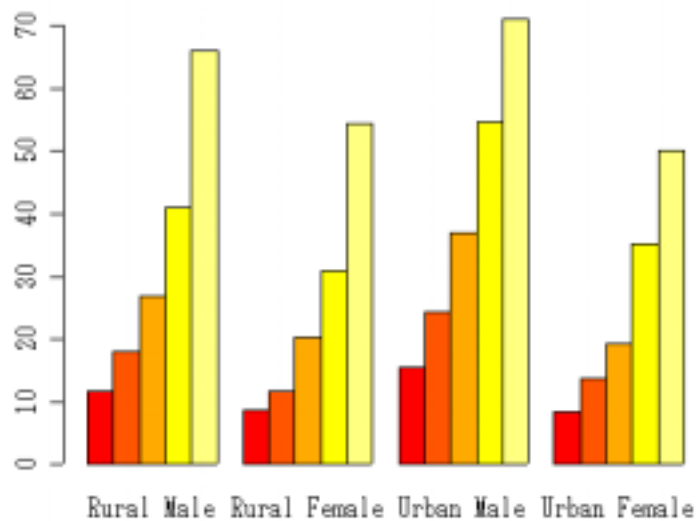


図2 棒グラフの例2

棒の色は自由に調整することが可能である。例えば、次のコマンドを実行すると図3のような棒グラフが作成される。ここでは4つの引数、データ、`beside`、`col`、`legend` が用いられている。引数 `col` は棒の色を設定する引数である。ここでは列の数に相当する5色の色を指定している。色を文字列で指定する場合は、色の名前を“ ”で囲む必要がある。ここでは“`lightblue`”、“`mistyrose`”、“`lightcyan`”、“`lavender`”、“`cornsilk`”という色を指定したが、自分で好きな色を自由に指定することができる。

```
>barplot(VADeaths, beside = TRUE, col = c("lightblue", "mistyrose", "lightcyan", "lavender",  
"cornsilk"),legend = rownames(VADeaths))  
>title(main = "Death Rates in Virginia", font= 5)
```

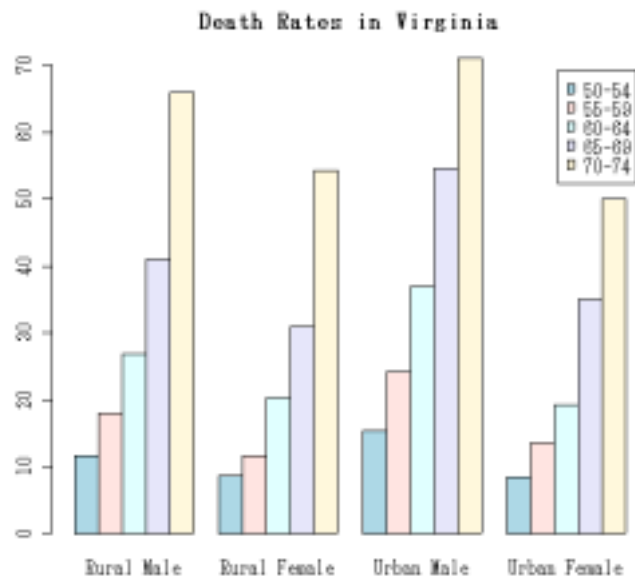


図3 棒グラフの例3

用いる色の表記はRで指定されている色でなければならない。Rで指定されている色の種類は次のコマンドを実行することで見ることができる。表示された657種類の色の中から好きな色を用いることができる。

```
>colors()
```

引数 legend は各棒に用いた色とデータを対応付けた凡例を作るための引数である。この rownames は行の名前を用いることを意味する。

関数 title はグラフのタイトルを書く関数で、メイン(main)、サブ(sub)、横軸(xlab)、縦軸(ylab)のタイトルの作成や、その文字の色(col)、フォント(font)、サイズ(cex)などを指定することが可能である。

表 1 barplot の書式と主な引数

書式	barplot(x, beside= , horiz= , col= , legend= )
x	用いるデータ
beside=	TRUE の場合、一列を一組とし、各データを一つの棒に描く FALSE の場合、一列のデータを一つの棒に描く
horiz=	TRUE の場合には横棒、FALSE の場合は縦棒
col=	棒の色(red/赤,blue/青,yellow/黄,green/緑,purple/紫,cyan/青緑, pink/ ピンク,lightblue/薄青,lightcyan/薄青緑,lightgreen/薄緑,mistyrose/薄 ピンク,lavender/ラベンダー,cornsik/トウモロコシ毛の色 , …… 色は文字列ではなく数値で指定することもできる。0 は白、1 は黒、2 は赤、3 は緑、4 は青、5 は青緑、6 はピンク、7 は黄、8 は灰)
legend=	棒の説明、棒の塗りわけ方、範囲を指定する。

## 2. 円グラフ

ここではすでに入力した果物の売上の割合データ sales を用いることにする。

```
> sales
Cherry Apple Grape Banana Other
    15    20    25    10    30
```

果物の売上を円グラフにしたとき、各果物の色を Cherry は violetred1、Apple は green3、Grape は purple、Banana は yellow、Other は cyan で表すことにする。そのためには、次のように各果物の色に対応する文字列ベクトルを作成しておくことと便利である。

```
> sales.col<-c("violetred1", "green3", "purple", "yellow", "cyan")
```

円グラフ(パイグラフ)を作成する関数は pie である。次のコマンドにより図 4 の円グラフが作成される。関数のなかの引数 radius は円の半径に関する引数で。数値が大きければ、円が大きくなる。

```
> pie(sales,col=sales.col, radius=1)
```

次のように引数を関数 pie に用いると円グラフを白黒で作成することができる。

```
> pie(sales,col=gray(seq(0.5,1,length=5)), radius=1)
```

また、引数「density = 値」を用いると円グラフの各部分に斜線を引いた図を作成する

ことができる。「値」が大きいほど斜線の間隔が狭い。斜線の角度は引数「angle = 値」を用いて調整する。

```
>pie(pie.sales, density = 15, angle = 15 + 10 * 1:5)
```



図4 円グラフ

### 3 ヒストグラム

データを小さいものから大きい順に並べ、階級ごとにまとめ、棒グラフで表現したものをヒストグラム(histogram)という。ヒストグラムはデータの全体状況をつかむのに有効である。ヒストグラムに適したデータは、長さ、重さ、速度など連続な量的データである。

ここでは、R に用意されている iris(アヤメ花、花菖蒲)というデータを用いてヒストグラムの作成について説明する。iris のデータは、setosa、versicolour、virginica という3種類の品種のアヤメの花について、各種類それぞれ 50 標本の、顎(がく)片の長さ(Sepal.Length)、顎片の幅(Sepal.Width)、花弁の長さ(Petal.Length)、花弁の幅(Petal.Width)を計測した 150 標本のデータである。R に用意されている iris のデータの形式は、150 行 5 列のデータフレーム型である。第 1 列から第 4 列までが、それぞれ顎(がく)片の長さ、顎片の幅、花弁の長さ、花弁の幅、第 5 列は品種のラベルである。1 行から 50 行目までが setosa という品種で、51 行から 100 行までが versicolor という品種で、101 行から 150 行までが virginica という品種のデータである。コマンド data(iris)を実行することにより、iris データが呼び出される。

```
> data(iris)
```

```
> iris
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
<中略>					
50	5.0	3.3	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor

<中略>					
100	5.7	2.8	4.1	1.3	versicolor
101	6.3	3.3	6.0	2.5	virginica
<中略>					
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

R上でヒストグラムを描く関数は **hist** である。データ iris 中の品種 setosa の第 2 変数 Sepal.Width(がく片の幅)のヒストグラムを作成してみる。品種 setosa は、iris データの 1 行から 50 行までである。品種 setosa の第 2 変数のヒストグラムは、次のコマンドで作成される。その結果を図 5 に示す。

```
> hist(iris[1:50,2])
```

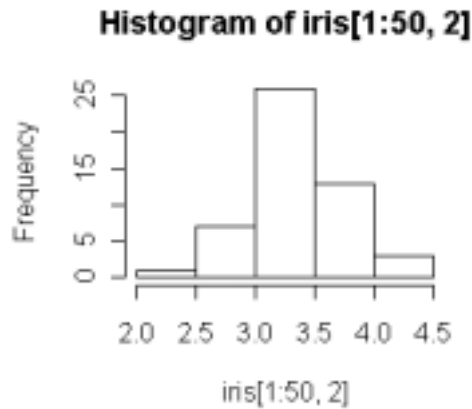


図 5 setosa の第 2 変数のヒストグラム

ヒストグラム図 5 から、品種 setosa のがく片の幅は 3 ~ 3.5 の周辺に集中されていることがわかる。図 3 の 5 つの棒の幅となる範囲、2 ~ 2.5、2.5 ~ 3、3 ~ 3.5、3.5 ~ 4、4 ~ 5 を階級という。このように作成したヒストグラムでは、データの値が境界線の値と同じの場合は、右の棒(階級)にカウントされる。例えば、図 5 では値が 2.5 の場合は、階級 2.5 ~ 3 に属する。

ヒストグラムの各棒に頻度を表したり、棒に色付けをしたりすることも可能である。例えば、コマンド

```
> hist(iris[1:50,2], col="gray", labels = TRUE)
```

により、図 6 に示すヒストグラムが作成される。col はヒストグラムの色、labels は度数を指定する引数である。

**Histogram of iris[1:50, 2]**

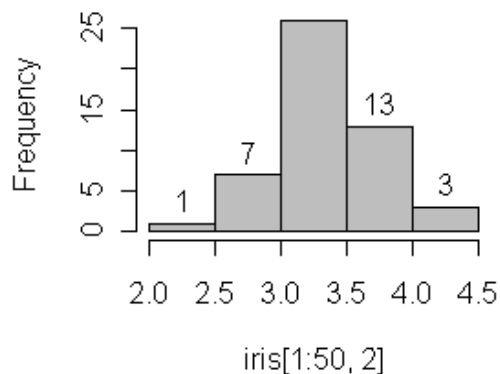


図 6 色と度数を加えたヒストグラム

ヒストグラムでは、引数 `breaks` を用いて各自が階級を自由に設定することができる。引数 `breaks` の階級設定は、各棒の境値のベクトルを用いる。ただし、階級の最小値と最大値はデータの範囲外でなければならない。

例えば、がく片の幅のヒストグラム階級を 1~2、2~3、3~4、4~5 にしたいときには、次のように `breaks=c(1,2,3,4,5)` を引数(あるいは `breaks=c(1:5)`)とする。次の書式により作成したヒストグラムを図 7 に示す。

```
> hist(iris[1:50,2],breaks=c(1:5),col="blue")
```

**Histogram of iris[1:50, 2]**

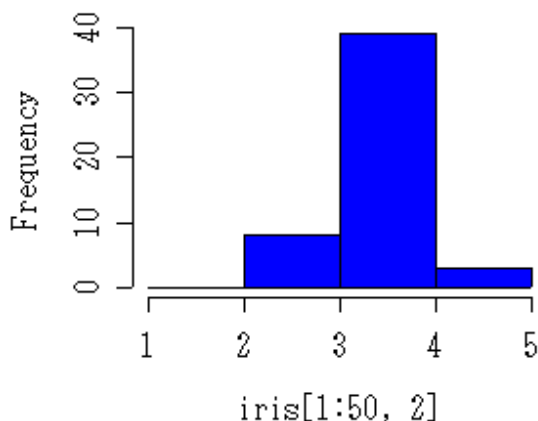


図 7 階級の幅を 1 としたヒストグラム

作成したヒストグラムに関連する情報を文字列として出力したいときには引数 `plot=FALSE` を用いる。度数は `counts`、階級の区分点は `breaks` という名でその値を返す。

## 5. 折れ線グラフ

折れ線グラフは、円グラフや棒グラフとともに広く使用されているデータの視覚化手段である。折れ線グラフが作成可能な関数はいくつかあるが、ここでは関数 `matplot` を用いた折れ線グラフの作成について紹介する。まず折れ線グラフ作成に用いるデータについて説明する。ここでは R に取り込まれている `VADeaths` を用いることにする。`VADeaths` は 1940 年代のバージニア州の 100 人あたりの死亡率について、年齢(行)、地域と性別(列)に分けた 5 行 5 列のデータである。年齢は 50-54、55-59、60-64、65-69、70-74、地域と性別は `Rural Male`(田舎/男性)、`Rural Female`(田舎/女性)、`Urban Male`(都市/男性)、`Urban Female`(都市/女性)に分けている。データ `VADeaths` と関数 `matplot` を用いた折れ線グラフの作成コマンドを次に示す。引数 `type="l"` の "l" line の頭文字でローマ数値の 1 ではない。

```
>data(VADeaths)
>matplot(VADeaths,type="l")
```

これで最も簡単な折れ線グラフが返される。引数 `type` を次のように設定すると折れ線にデータの行の番号が付き加える。

```
>matplot(VADeaths,type="b")
```

次のように関数 `legend` を用いてどの線がどのデータを現しているかに関する凡例を追加することができる。関数 `legend` の中の `1, max(VADeaths)` は凡例を置く座標、`colnames(VADeaths)` は用いた凡例のラベルである。`col=1:CL` は 4 種類の色、`lty=1:CL` は 4 種類の線をデータの行ごとに換えることを表している。

```
>matplot(VADeaths,type="b")
>CL<-length(VADeaths[,1])
>legend(1, max(VADeaths), colnames(VADeaths),col=1:CL,lty=1:CL)
```

次の関数 `matplot` の引数 `pch=n` はマークの種類の設定を行う。`n=1` は `,`、`n=2` は `o`、`n=3` は `+`、`n=4` は `x` と設定されている。`n=1:length(VADeaths[,1])` のように指定することによって自動的に線の番号を記号に入れ換える。また関数 `matplot` の引数 `axes=F` は軸の枠を描かないことを意味する。

関数 `axis(n)` は軸を描く。`n=1` は下の横軸、`n=2` は右の縦軸、`n=3` は上横軸、`n=4` は右

縦軸。関数 axis では、軸のメモリ、ラベル、色、線の太さなどを自由に設定することができる。

```
>matplot(VADeaths,type="b", pch=1:length(VADeaths[,1]),xlab="",ylab="",axes=F)
>axis(1, 1:length(VADeaths[,1]),row.names(VADeaths));
>axis(2)
>CL<-length(VADeaths[,1])
>legend(1, max(VADeaths), colnames(VADeaths),col=1:CL,lty=1:CL)
```

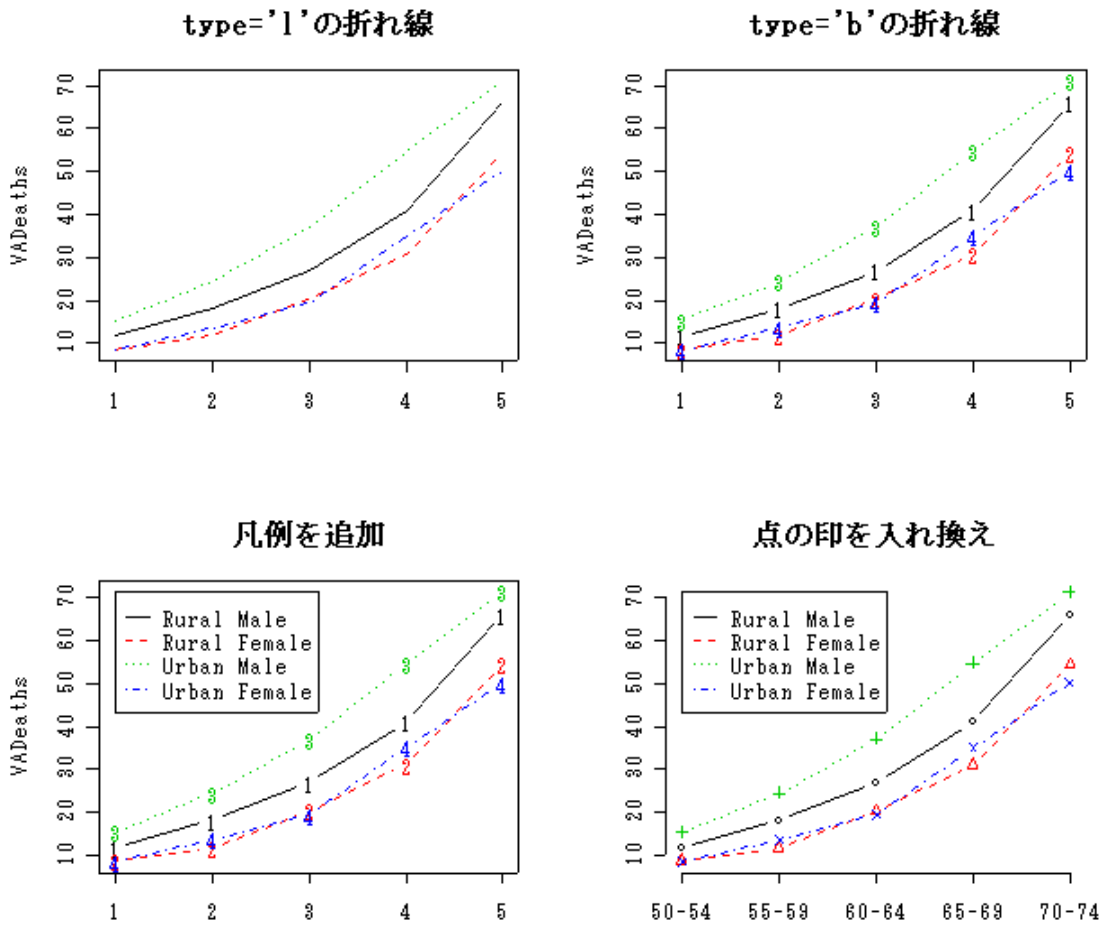


図 8 折れ線グラフ例

## 5. 箱ひげ図(box-whisker plot)

データ解析では、データの中心、散らばりの具合、異常なデータなどを考察するために、図9のような長方形(箱)に直線(ひげ)をつなげたグラフをよく用いる。このグラフを箱ひげ図という。

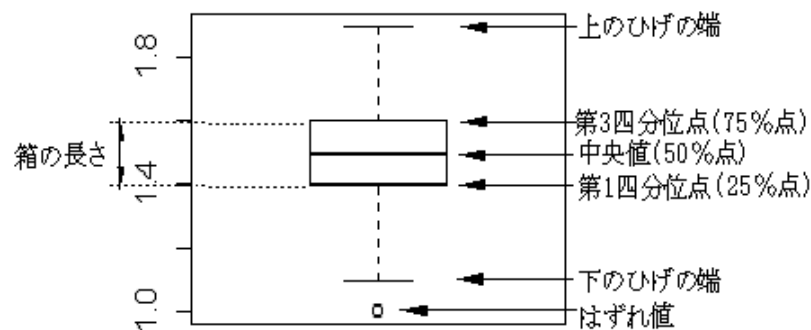


図9 箱ひげ図の構造

通常箱ひげ図では四分位数を用いる。四分位数とはデータを小さい値から大きい順に並べ、その範囲を4等分した場合、小さい値から第1等分と第2等分の境である25%点を第1四分位数、第2等分と第3等分の境である50%点を第2四分位数、第3等分と第4等分の境である75%点を第3四分位数という。

箱ひげ図の箱の長さは、第3四分位数から第1四分位数の値を引いた値である。ひげの長さは、一般的に箱の長さの1.5倍とする。この1.5倍は絶対的なものではない。Rでは自由に設定することができるが、デフォルト(初期に設定されている値)のひげの長さは箱の長さの1.5倍である。ひげの上(下)端は、箱の長さの1.5倍以内にある最大(小)値である。ひげの端の外側にある値を外れ値、あるいはアウトサイド値とする。

箱ひげ図を用いて iris データのがく片の長さ(変数 1)について、三種類のアヤメを比較してみよ。コマンド

```
>boxplot(iris[1:50,1],iris[51:100,1],iris[101:150,1],col=2:4,names=c("S","C","V"))
```

を実行すると、図10に示す箱ひげ図が作成される。引数 col は箱の色、names は各箱ひげ図のラベルを指定する引数である。

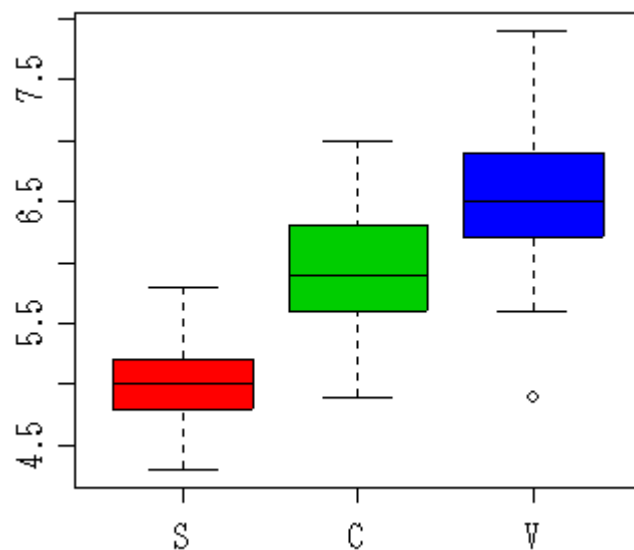


図 10 iris のがく片の長さの箱ひげ図

図 10 からそれぞれの、がく片の長さの特徴が概観できる。長さがもっとも短いのが setosa という品種で、その次が versicolor で、もっともがく片が長いのは virginica という品種である。かつ virginica のなかには、一つのデータが特に小さいことも見て読み取られる。