Par() 関数

Shouhei TAKEUCHI

October 6, 2015

Contents

par() 関数の使い方

作図の際に、さまざまなパラメータを指定する par() 関数を使いこなすために、情報をストックしていくファイル。

このファイルの更新情報

このファイルは 2015-10-06 08:59:44 に更新されました。

- github で公開してみた。

データフレームの作成

最初にデータセットを準備しておく。データフレームは簡単なものを用意しておく。

```
set.seed(1)
dat <- data.frame(height = rnorm(30, 150, 10),
                  weight = rnorm(30, 40, 5),
                  sex = sample(c(" M" , " F" ), 30, replace = TRUE),
                  village = sample(c(" A" , " B" , " C" ), 30, replace = TRUE),
                  disease = rbinom(30, 1, 0.3))
```

グラフの見かけを調節する

表示するグラフの色や線の種類など、グラフの見かけを調整するパラメータを指定する。

色

色の指定のため、col、fg、bg、border 引数についてまとめる。

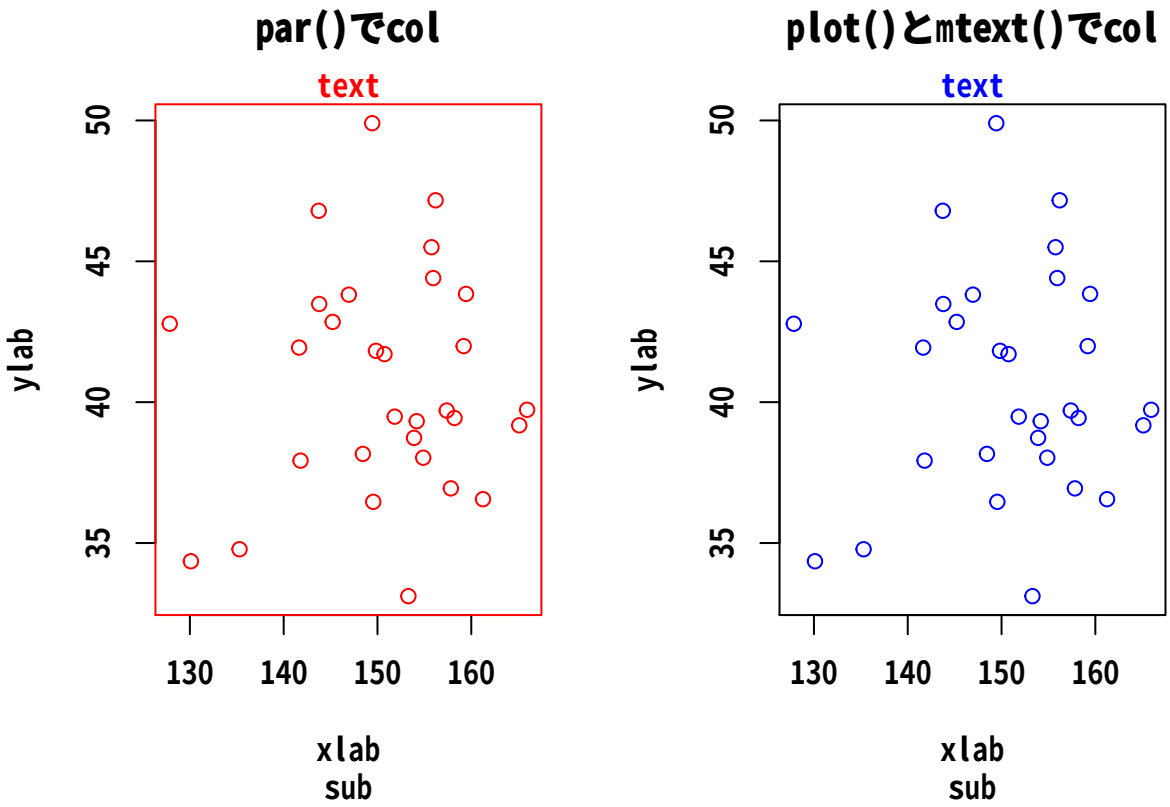
col

plot region に描かれるデータシンボル、線、テキストの色の指定に利用する。軸、ラベル、タイトル、サブタイトルは col.axis、col.lab、col.main、col.sub を利用する。

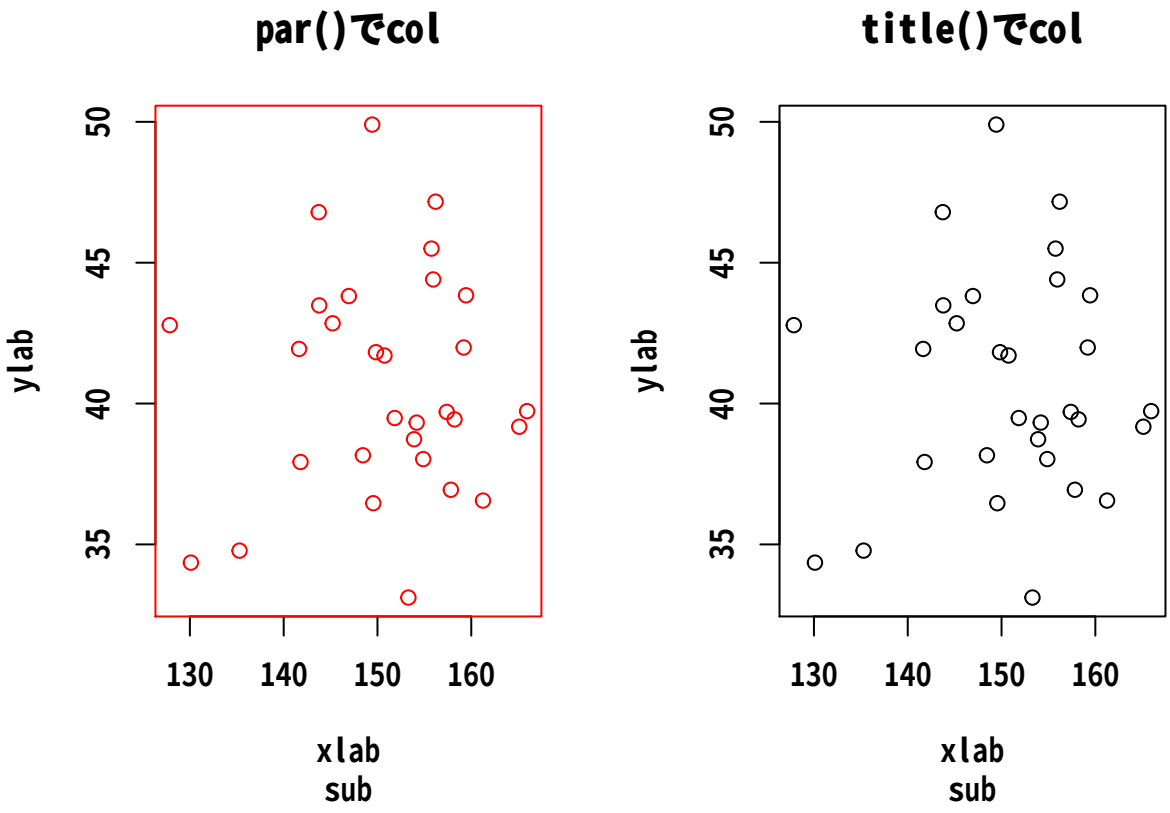
散布図　par() 関数内の col オプションは、plot() 関数で作った散布図の、データシンボルや線と枠の線の色を変更している。plot() 関数内の col オプションでは、データシンボルの色だけが変わっている。低水準関数でのプロット（lines() 関数や points() 関数）では、plot() 関数の時と同じ影響範囲となっている。

マージン（プロット領域の外側、作図領域の内側）では、mtext() 関数の出力には影響している。title() 関数には影響しない。

```
op1 <- par(mfrow = c(1, 2))

op2 <- par(col = "red")
plot(dat$height, dat$weight,
     main = "par() で col", sub = "sub",
     xlab = "xlab", ylab = "ylab")
mtext("text")
par(op2)

plot(dat$height, dat$weight, col = "blue",
     main = "plot() と mtext() で col", sub = "sub",
     xlab = "xlab", ylab = "ylab")
mtext("text", col = "blue")
```

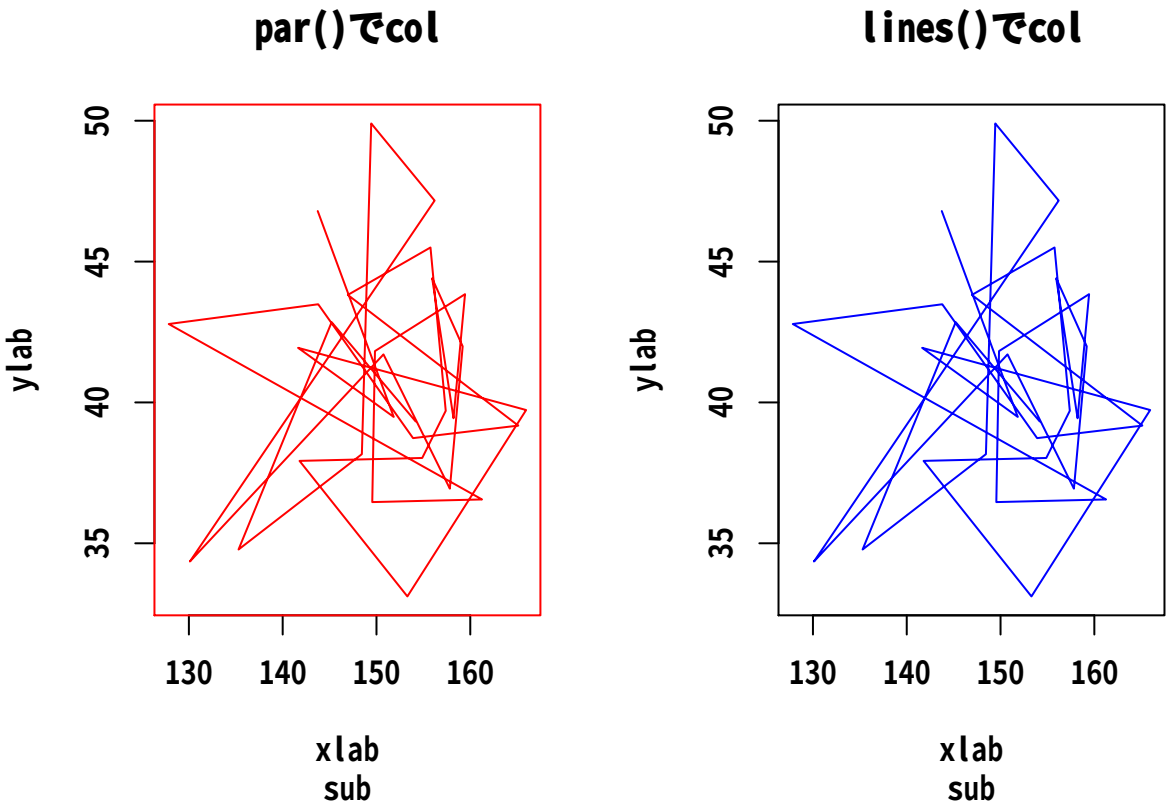


```
op2 <- par(col = "red")
plot(dat$height, dat$weight,
     main = "", sub = "",
     xlab = "", ylab = "")
title(main = "par() で col", sub = "sub",
      xlab = "xlab", ylab = "ylab")
par(op2)

plot(dat$height, dat$weight,
     main = "", sub = "",
     xlab = "", ylab = "")
title(main = "title() で col", sub = "sub",
      xlab = "xlab", ylab = "ylab",
      col = "blue")
```

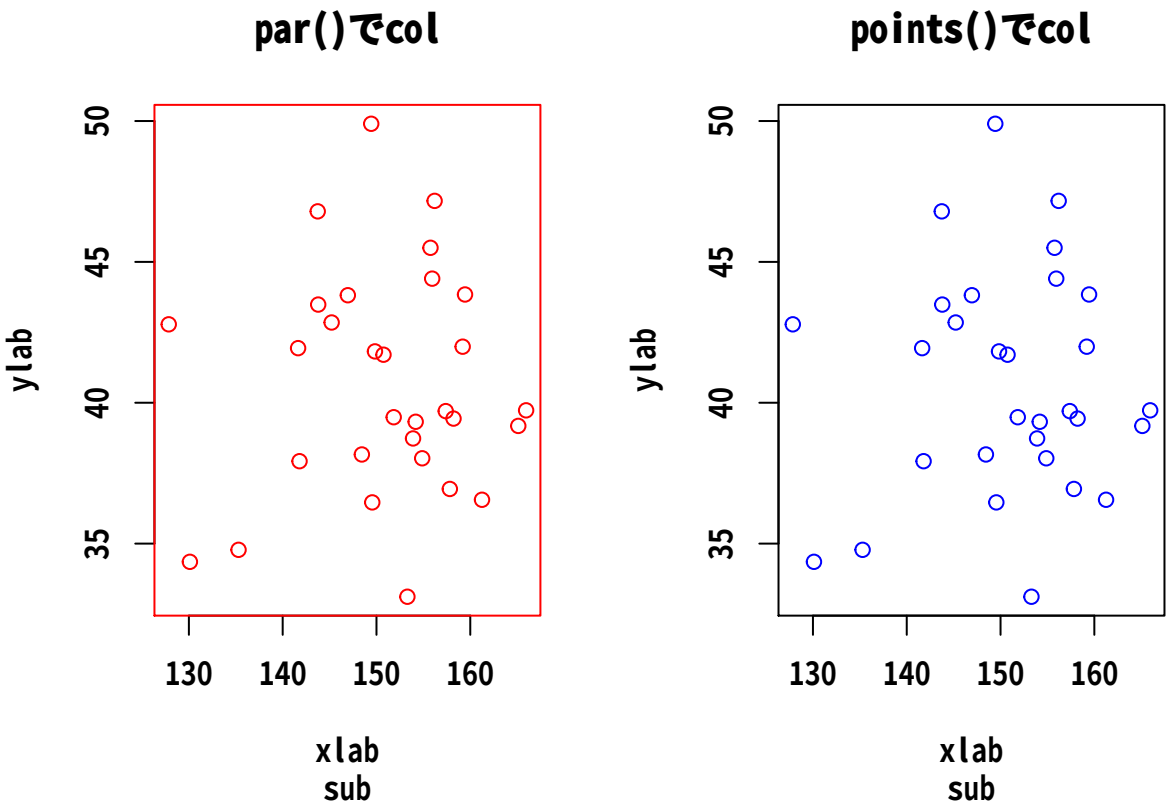



```
op2 <- par(col = "red")
plot(dat$height, dat$weight, type = "n",
     main = "par() で col", sub = "sub",
     xlab = "xlab", ylab = "ylab")
lines(dat$height, dat$weight)
par(op2)
```

```
plot(dat$height, dat$weight, type = " n" ,
     main = " lines() で col" , sub = " sub" ,
     xlab = " xlab" , ylab = " ylab" )
lines(dat$height, dat$weight, col = " blue" )
```



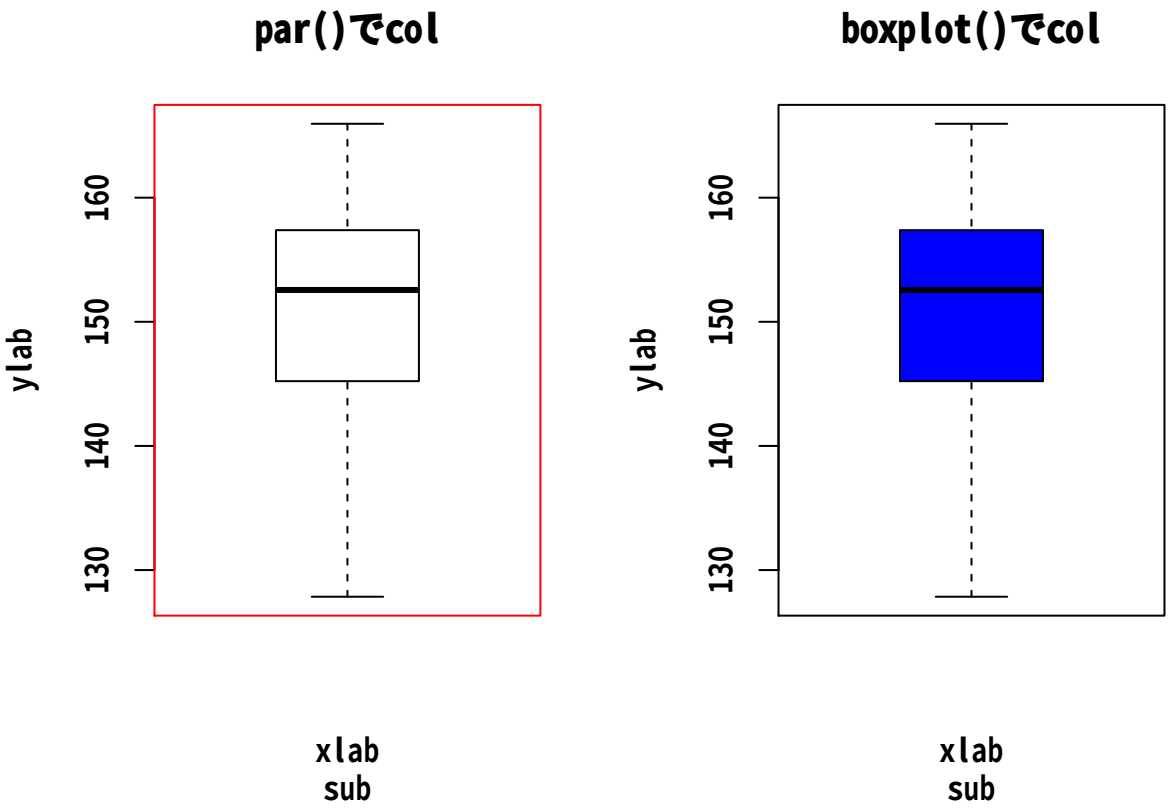

```
op2 <- par(col = " red" )
plot(dat$height, dat$weight, type = " n" ,
     main = " par() で col" , sub = " sub" ,
     xlab = " xlab" , ylab = " ylab" )
points(dat$height, dat$weight)
par(op2)

plot(dat$height, dat$weight, type = " n" ,
     main = " points() で col" , sub = " sub" ,
     xlab = " xlab" , ylab = " ylab" )
points(dat$height, dat$weight, col = " blue" )
```




```
par(op1)
```

箱ヒゲ図、ヒストグラム　par() 関数の col オプションは、箱ヒゲ図では枠の線にだけ反映されている。boxplo() 関数内の col オプションでは、箱の中が塗られるらしい。boxplot() 関数に関しては、他のリポジトリ（↓）でまとめたので、そちらも参照するといいかも。https://github.com/takeshou/boxplot-summary

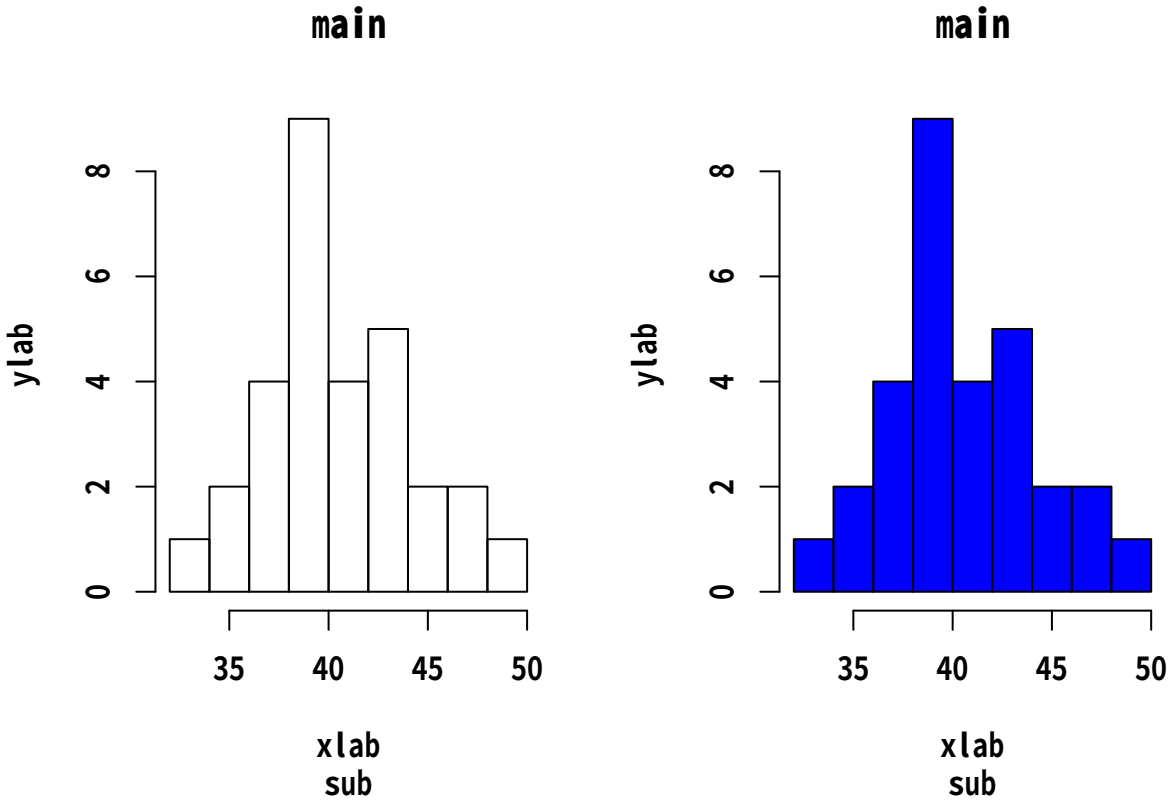
```
op1 <- par(mfrow = c(1, 2))
op2 <- par(col = "red")
boxplot(dat$height,
        main = "par() で col", sub = "sub",
        xlab = "xlab", ylab = "ylab")
par(op2)

boxplot(dat$height, col = "blue",
        main = "boxplot() で col", sub = "sub",
        xlab = "xlab", ylab = "ylab")
```




```
par(op1)
```

ヒストグラムでは、par() 影響なし、hist() では箱の中

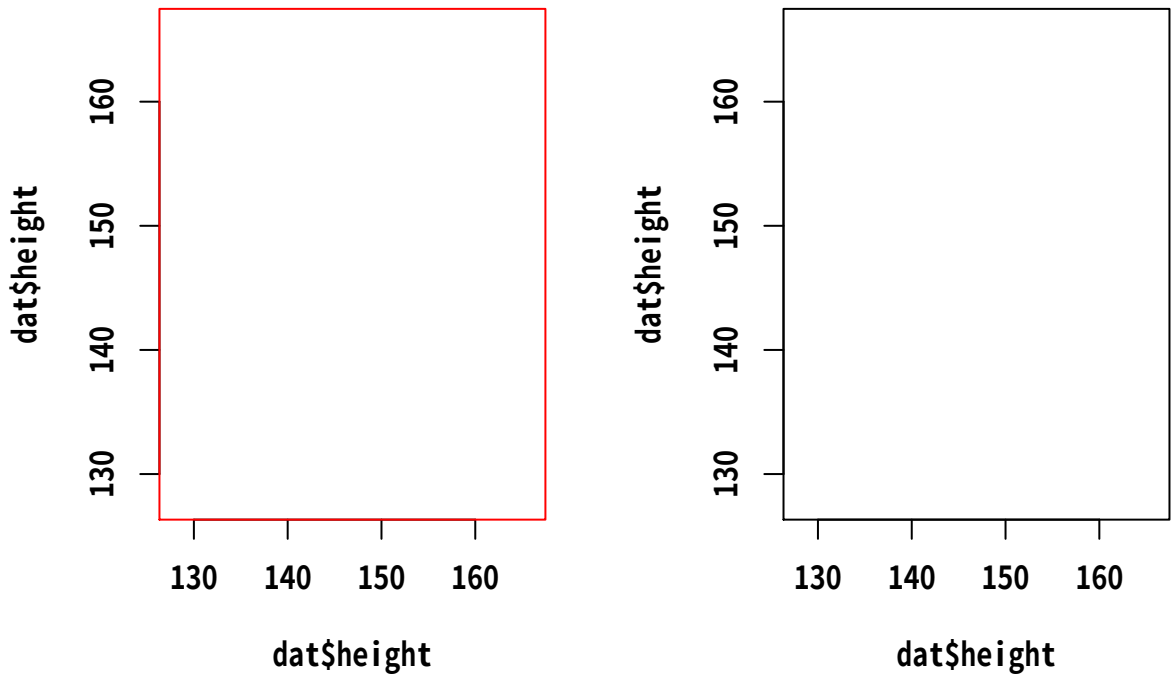
```
op1 <- par(mfrow = c(1, 2))
op2 <- par(col = " red" )
hist(dat$weight,
     main = " main" , sub = " sub" ,
     xlab = " xlab" , ylab = " ylab" )
par(op2)

hist(dat$weight, col = " blue" ,
     main = " main" , sub = " sub" ,
     xlab = " xlab" , ylab = " ylab" )
```




```
par(op1)
```

長方形、棒グラフ　rect() 関数では四角形の塗りつぶし

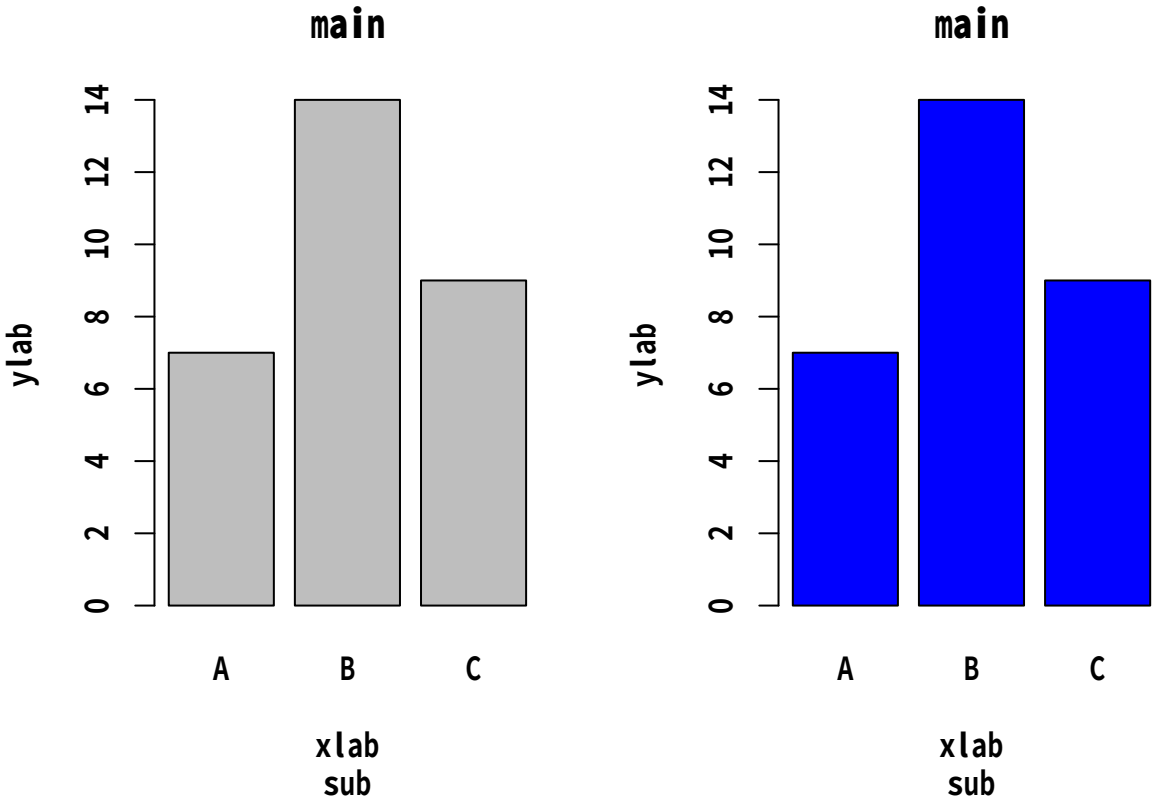
```
op1 <- par(mfrow = c(1, 2))
op2 <- par(col = "red")
plot(dat$height, dat$height, type = "n")
rect(dat$height, dat$weight, dat$sex, dat$village)
par(op2)

plot(dat$height, dat$height, type = "n")
rect(dat$height, dat$weight, dat$sex, dat$village, col = "blue")
```




```
par(op1)
```

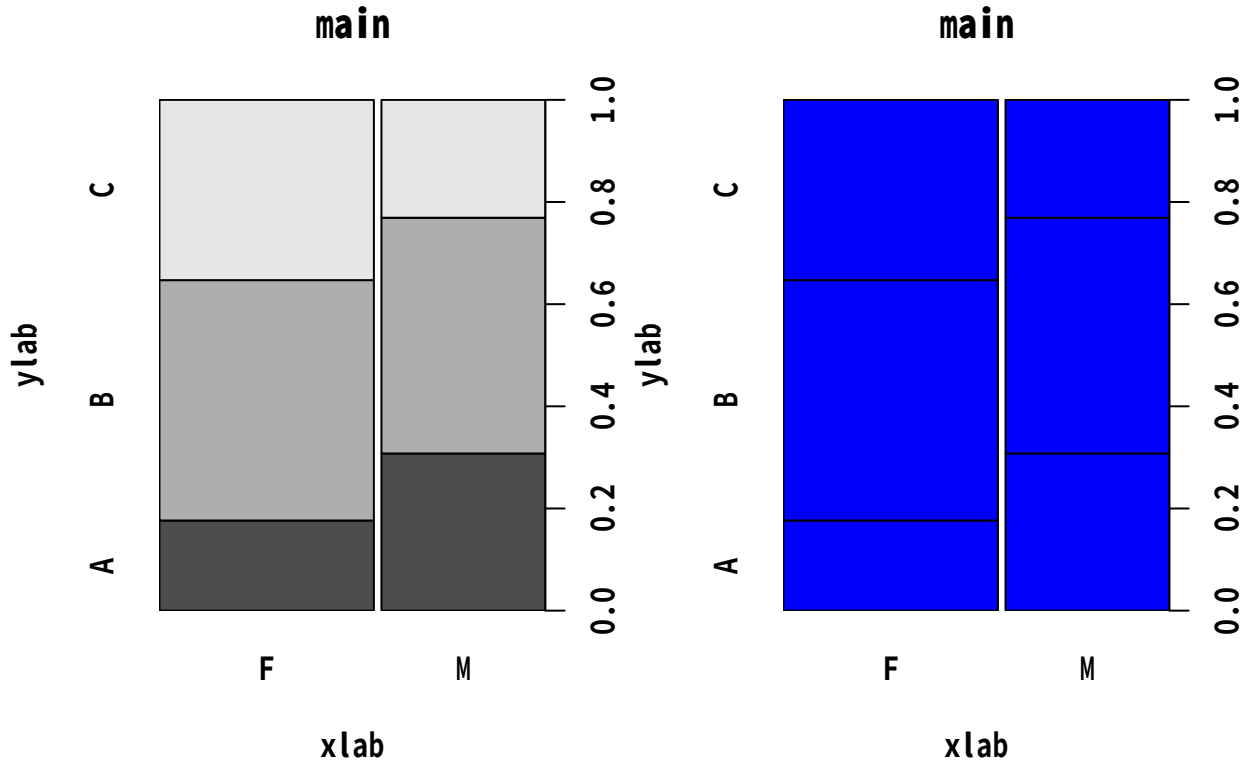
barplot() 関数では、棒の塗りつぶし

```
op1 <- par(mfrow = c(1, 2))
op2 <- par(col = " red" )
barplot(table(dat$village),
        main = " main" , sub = " sub" ,
        xlab = " xlab" , ylab = " ylab" )
par(op2)
barplot(table(dat$village), col = " blue" ,
        main = " main" , sub = " sub" ,
        xlab = " xlab" , ylab = " ylab" )
```

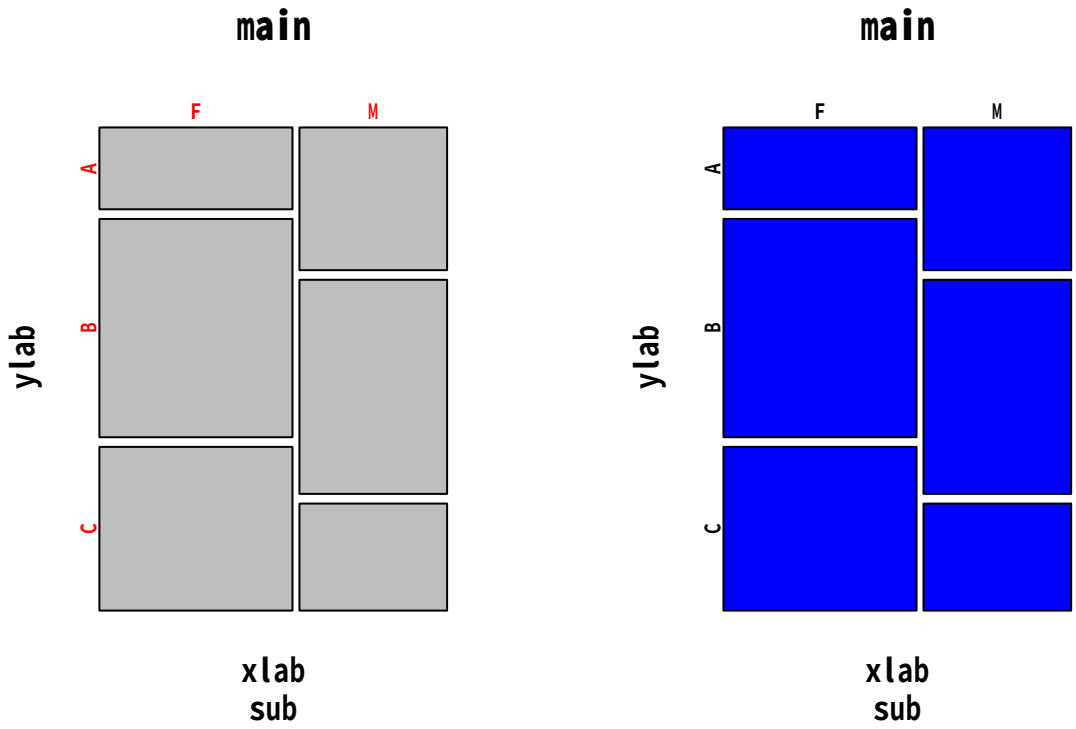



```
par(op1)
```

```
op1 <- par(mfrow = c(1, 2))
op2 <- par(col = " red" )
plot(dat$sex, dat$village,
     main = " main" , sub = " sub" ,
     xlab = " xlab" , ylab = " ylab" )
par(op2)
plot(dat$sex, dat$village, col = " blue" ,
     main = " main" , sub = " sub" ,
     xlab = " xlab" , ylab = " ylab" )
```




```
op2 <- par(col = " red" )
mosaicplot(dat$sex ~ dat$village,
           main = " main" , sub = " sub" ,
           xlab = " xlab" , ylab = " ylab" )
par(op2)
mosaicplot(dat$sex ~ dat$village, col = " blue" ,
           main = " main" , sub = " sub" ,
           xlab = " xlab" , ylab = " ylab" )
```




```
par(op1)
```

fg

bg

border

rect() 関数で border を指定すると四角形の枠の色

lty オプションで実感

lty オプションを高水準関数内で指定したときと、par() 関数で指定したときの違いをチェックする。

```
# op1 <- par(mfrow = c(2, 2))
# op2 <- par(lty = "dashed")
# plot(x1, x2, type = "l")# 線は破線
# plot(x1, x2, type = "l", lty = "solid")# 線は実線
# plot(x1, x2, type = "l")# 線は破線
```

```
# par(op2)
# par(op1)
```

余白

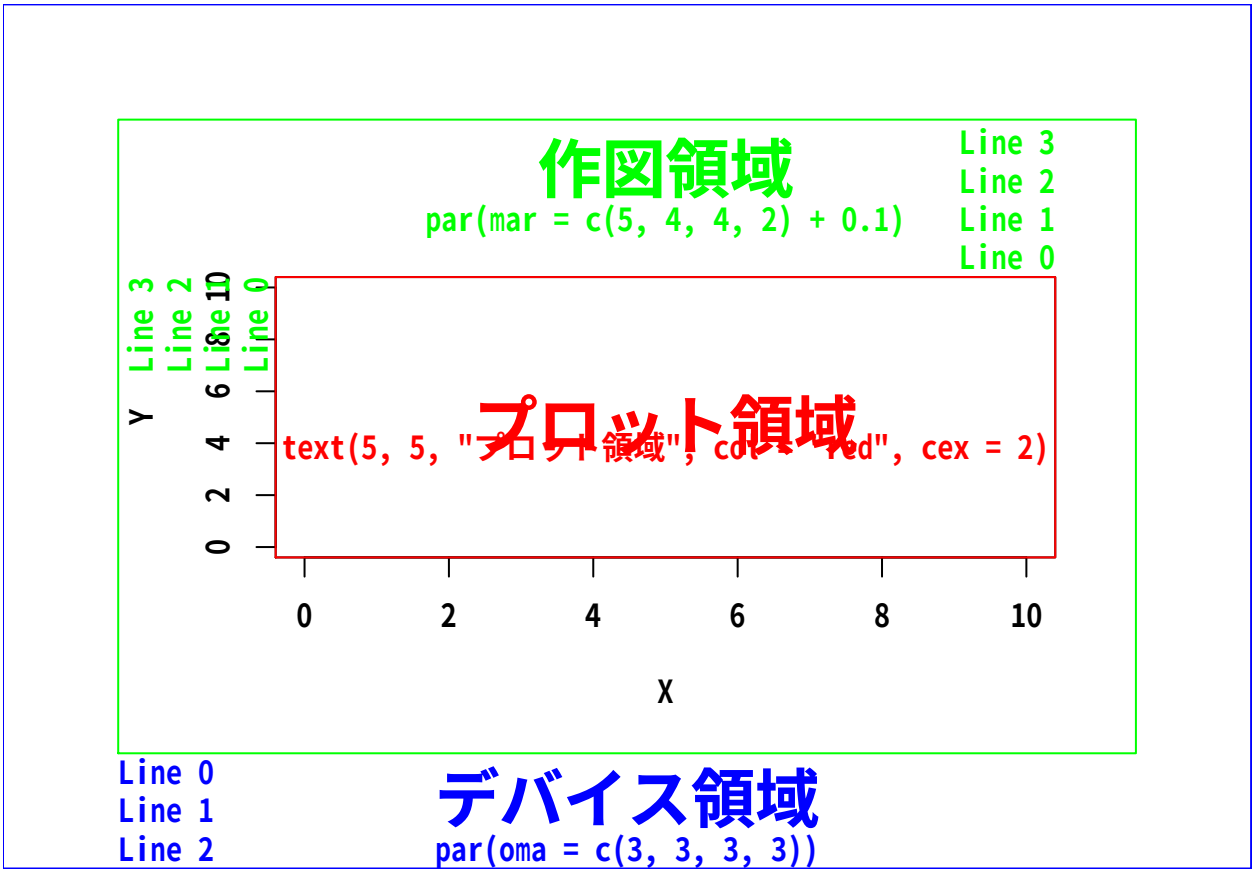
インチで指定する方法と、行数で指定する方法がある。行数で指定する方が、理解しやすいので、こちらを記述しておく。図の構成を知っておくと良い。プロットをするプロット領域（plot）、それにマージンを加えた作図領域（figure）、さらにその外側のデバイス領域（outer）がある。今回は、プロット領域の外側を余白とよんで、作図領域内の余白を制御するのに mar オプション、デバイス領域内で作図領域との間の余白を制御するのに oma オプションを使う。

参　考：http://stat.biopapyrus.net/graph/plotarea.html　参　考：http://rgraphics.limnology.wisc.edu/rmargins_sf.php

mar、oma

```
# 作図系の領域の理解と制御
# データの作成
dat <- data.frame(x = 0:10, y = 0:10)

# データのプロット
## 「outer」領域の確保
par(oma = c(3, 3, 3, 3)) # すべてのサイドに 3 行

## 「figure」領域の確保
par(mar = c(5, 4, 4, 2) + 0.1) # 下 5.1 行、左、上 4.1 行、右 2.1 行
# par(mar = c(5.1, 4.1, 4.1, 2.1)) と同じ

# プロット
plot(dat$x, dat$y, type = "n", xlab = "X", ylab = "Y")

# テキスト表示
## プロット領域 (plot)
text(5, 5, "プロット領域", col = "red", cex = 2)
text(5, 4, "text(5, 5, ¥"プロット領域¥", col = ¥"red¥", cex = 2)",
     col = "red", cex = 1)
box("plot", col = "red") # プロット領域 (plot) を囲む

## 作図領域 (figure)
mtext("作図領域", side = 3, line = 2, cex = 2, col = "green")
mtext("par(mar = c(5, 4, 4, 2) + 0.1)", side = 3, line = 1,
      cex = 1, col = "green")
```

```
### 上（行は 0 から数える）
mtext(" Line 0", side = 3, line = 0, adj = 1.0, cex = 1, col = " green")
mtext(" Line 1", side = 3, line = 1, adj = 1.0, cex = 1, col = " green")
mtext(" Line 2", side = 3, line = 2, adj = 1.0, cex = 1, col = " green")
mtext(" Line 3", side = 3, line = 3, adj = 1.0, cex = 1, col = " green")
### 左
mtext(" Line 0", side = 2, line = 0, adj = 1.0, cex = 1, col = " green")
mtext(" Line 1", side = 2, line = 1, adj = 1.0, cex = 1, col = " green")
mtext(" Line 2", side = 2, line = 2, adj = 1.0, cex = 1, col = " green")
mtext(" Line 3", side = 2, line = 3, adj = 1.0, cex = 1, col = " green")
box(" figure", col=" green") # 作図領域（figure）を囲む

## デバイス領域（outer）
mtext(" デバイス領域", side = 1, line = 1,
      cex = 2, col = " blue", outer = TRUE)
mtext(" par(oma = c(3, 3, 3, 3))", side = 1, line = 2,
      cex = 1, col = " blue", outer = TRUE)
mtext(" Line 0", side=1, line = 0, adj = 0.0,
      cex = 1, col = " blue", outer = TRUE)
mtext(" Line 1", side=1, line = 1, adj = 0.0,
      cex = 1, col = " blue", outer = TRUE)
mtext(" Line 2", side=1, line = 2, adj = 0.0,
      cex = 1, col = " blue", outer = TRUE)
box(" outer", col=" blue") # デバイス領域（figure）を囲む
```



複数のグラフを配置する

追加注記つきのグラフ

高水準関数内でも指定可能

par() でのみ指定可能

読み込みだけ