

実例: Rを使ったデータの整理・可視化

事前準備

- 配布したフォルダの中で1-ichinokawa内にある.RData(またはファイル名がないRのアイコン)をクリックするとRが立ち上がります →
フォルダ「1-ichinokawa」内でRの作業をする準備が整いました
- Rstudioをインストールしている人：Rのコードが書いてある kensyu_ichinokawa.Rファイルを右クリックし、Rstudioで開いてもOKです

データの読み込み

- 2-okamuraフォルダ内にある dat1.csv を読み込んでみましょう
- 関数(引数=設定, 引数=設定) という構造
- read.csv(ファイル名)

```
# '../'は一個上のフォルダ、という意味です  
cpue.data <- read.csv("../2-okamura/dat1.csv")
```

データの中身の確認

```
# オブジェクトの先頭の数行だけを表示する関数（オブジェクト名）  
head(cpue.data)
```

	count	year	site	plant
1	1	0	0	tree
2	1	0	1	tree
3	5	0	2	tree
4	6	0	3	tree
5	3	0	4	tree
6	5	0	5	tree

```
# どのくらいの大きさの行列か確認する  
dim(cpue.data)
```

```
[1] 100  4
```

データの表示

```
cpue.data[, 1] # 一列目のデータだけを表示する
```

```
[1] 1 1 5 6 3 5 7 6 7 9 4 4 5 6 6 9 8 10 10  
[24] 5 5 4 3 5 6 2 2 1 2 4 3 6 5 6 5 4 0 0  
[47] 0 0 0 0 0 2 3 2 3 4 5 6 6 4 0 2 4 3 6  
[70] 4 0 2 3 4 2  
[ reached getOption("max.print") -- omitted 25 entries ]
```

```
cpue.data[1, 1] # 1行・1列目のデータだけを表示する
```

```
[1] 1
```

```
cpue.data$count # '$' + 列の名前という指定のしかたもできる
```

```
[1] 1 1 5 6 3 5 7 6 7 9 4 4 5 6 6 9 8 10 10  
[24] 5 5 4 3 5 6 2 2 1 2 4 3 6 5 6 5 4 0 0  
[47] 0 0 0 0 0 2 3 2 3 4 5 6 6 4 0 2 4 3 6  
[70] 4 0 2 3 4 2  
[ reached getOption("max.print") -- omitted 25 entries ]
```

値の代入

- "オブジェクト名" <- "別のオブジェクト"
- 数字や文字などを入れておく「箱」のようなものをRでは「オブジェクト」と呼びます。

```
cpue1.1 <- cpue.data[1, 1] # CPUEデータの一行目のデータをcpue1.1  
cpue1.1
```

```
[1] 1
```

```
cpue.N <- cpue.data$count # CPUEデータのN(尾数)データに別の名前を  
cpue.N
```

```
[1] 1 1 5 6 3 5 7 6 7 9 4 4 5 6 6 9 8 10 10  
[24] 5 5 4 3 5 6 2 2 1 2 4 3 6 5 6 5 4 0 0  
[47] 0 0 0 0 0 2 3 2 3 4 5 6 6 4 0 2 4 3 6  
[70] 4 0 2 3 4 2  
[ reached getOption("max.print") -- omitted 25 entries ]
```

年で集計する

- excelのピボットテーブル的な機能に対応します
- `tapply(集計する列, 注目する列, 関数)` といったかたちで使います

```
cpue.year.mean <- tapply(cpue.data$count, cpue.data$year, mean)  
cpue.year.mean
```

0	1	2	3	4	5	6	7	8	9
5.0	7.1	3.5	3.8	0.0	3.5	4.0	3.8	5.7	0.0

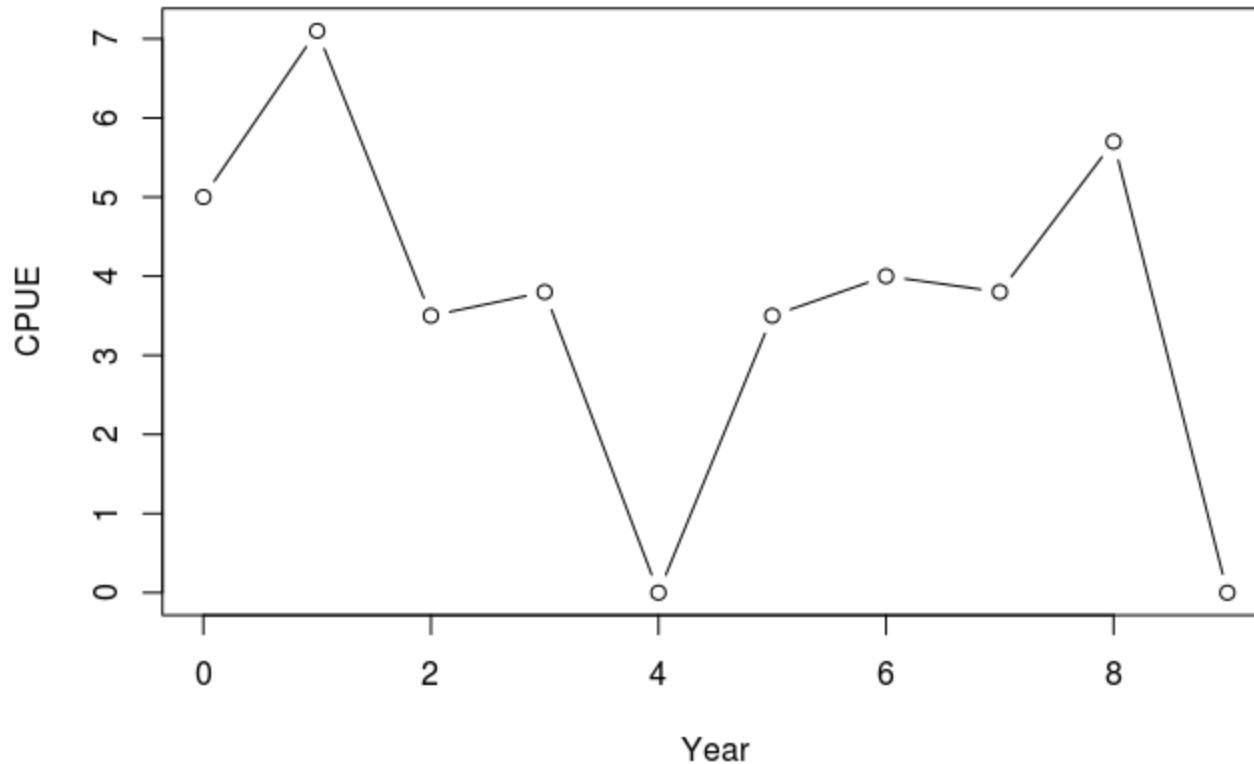
データを抽出する

- subset(対象とするデータ, 条件)

```
cpue.site1 <- subset(cpue.data, site == 1)
cpue.site2 <- subset(cpue.data, site == 2)
cpue.year.mean.site1 <- tapply(cpue.site1$count, cpue.site1$year, mean)
cpue.year.mean.site2 <- tapply(cpue.site2$count, cpue.site2$year, mean)
```

データをプロットする

```
# ylim=y軸の範囲, type=どんなやりかたでプロットするか ('b':線と点、  
# 線のみ、'p': 点のみ)  
plot(names(cpue.year.mean), cpue.year.mean, xlab = "Year", ylab =  
      max(cpue.year.mean)), type = "b", col = 1)
```



複数のグラフを並べる1

- `par` : グラフのいろいろな設定をする関数です
- `par(mfrow=c(1,3))` 1x3列で図を並べる

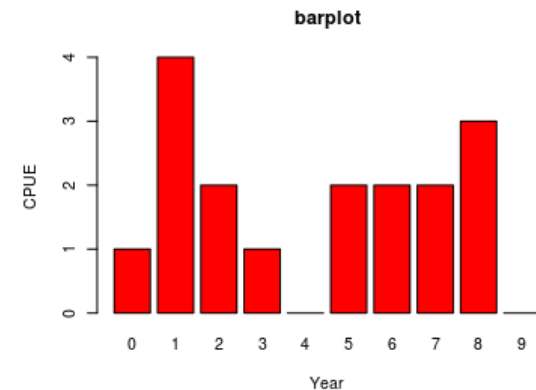
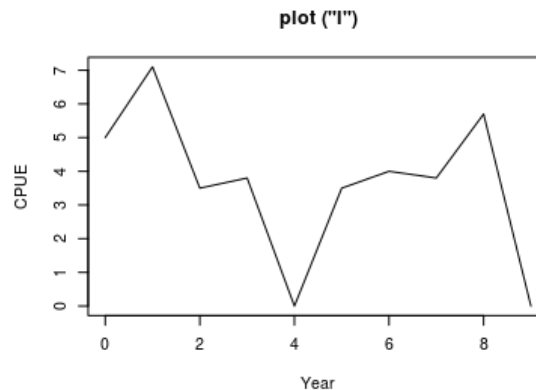
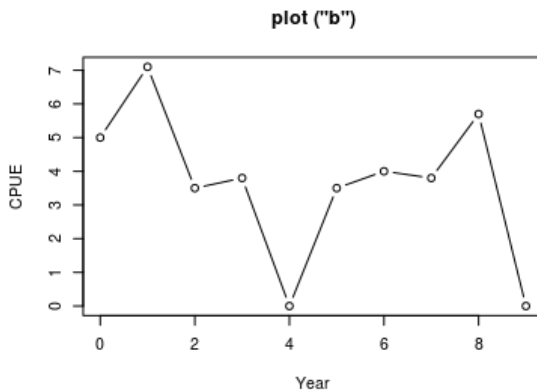
```
par(mfrow = c(1, 5))
```

複数のグラフを並べる2

```
par(mfrow = c(1, 3))
plot(names(cpue.year.mean), cpue.year.mean, xlab = "Year", ylab =
      max(cpue.year.mean)), type = "b", col = 1)
title("plot (\"b\")")

plot(names(cpue.year.mean), cpue.year.mean, xlab = "Year", ylab =
      max(cpue.year.mean)), type = "l", col = 1)
title("plot (\"l\")")

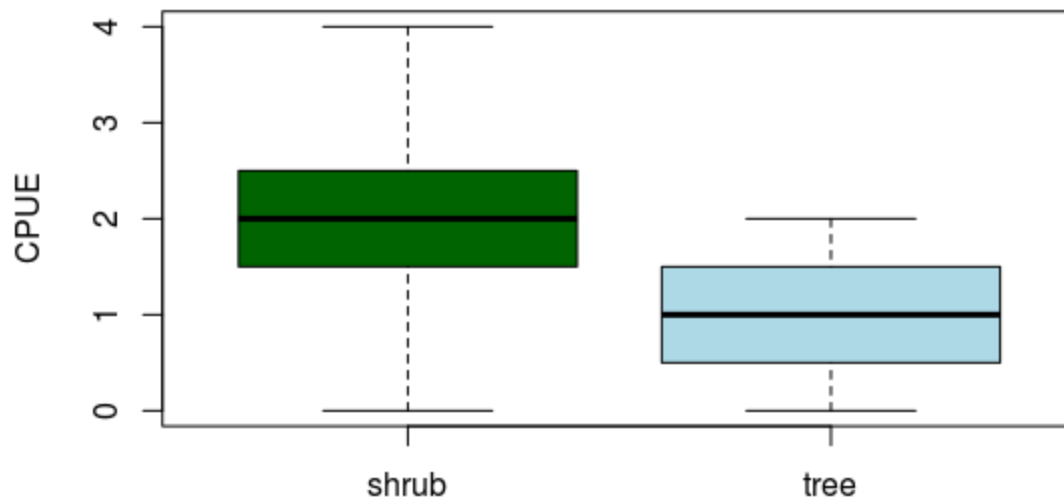
barplot(cpue.year.mean.site1, xlab = "Year", ylab = "CPUE", col =
      "red", las = 2)
title("barplot")
```



繰り返し処理I

- たとえばsiteごと・plant/shrubごとに個体数を比較するには??

```
# site1のデータだけを取り出す  
cpue.site1 <- subset(cpue.data, site == 1)  
# countをplantごとに箱ひげ図で示す  
boxplot(cpue.site1$count ~ cpue.site1$plant, ylab = "CPUE", col  
        "lightblue"))
```

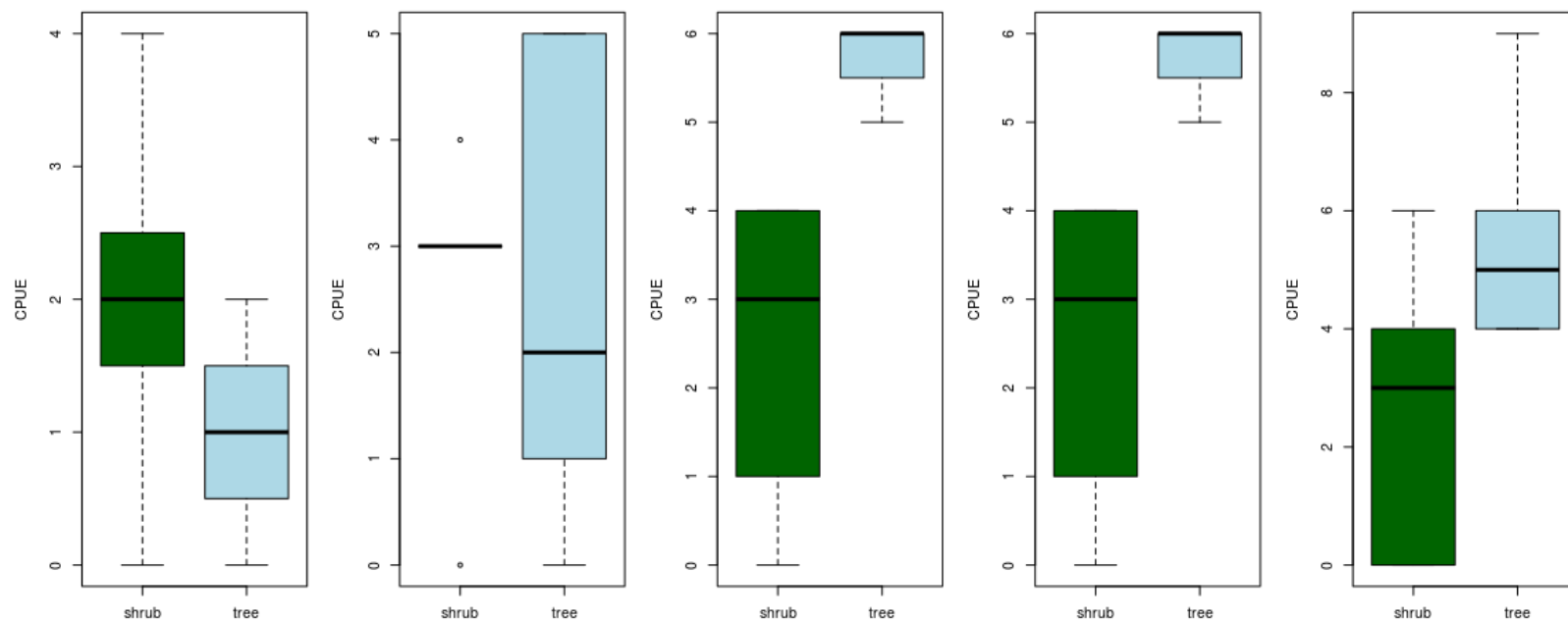


繰り返し処理2

- これを5回繰り返せば良い

```
par(mfrow = c(1, 5), mar = c(3, 4.3, 2, 1))
cpue.site1 <- subset(cpue.data, site == 1)
boxplot(cpue.site1$count ~ cpue.site1$plant, ylab = "CPUE", col
        "lightblue"))
cpue.site2 <- subset(cpue.data, site == 2)
boxplot(cpue.site2$count ~ cpue.site2$plant, ylab = "CPUE", col
        "lightblue"))
cpue.site3 <- subset(cpue.data, site == 3)
boxplot(cpue.site3$count ~ cpue.site3$plant, ylab = "CPUE", col
        "lightblue"))
cpue.site4 <- subset(cpue.data, site == 4)
boxplot(cpue.site3$count ~ cpue.site3$plant, ylab = "CPUE", col
        "lightblue"))
cpue.site5 <- subset(cpue.data, site == 5)
boxplot(cpue.site5$count ~ cpue.site5$plant, ylab = "CPUE", col
        "lightblue"))
```

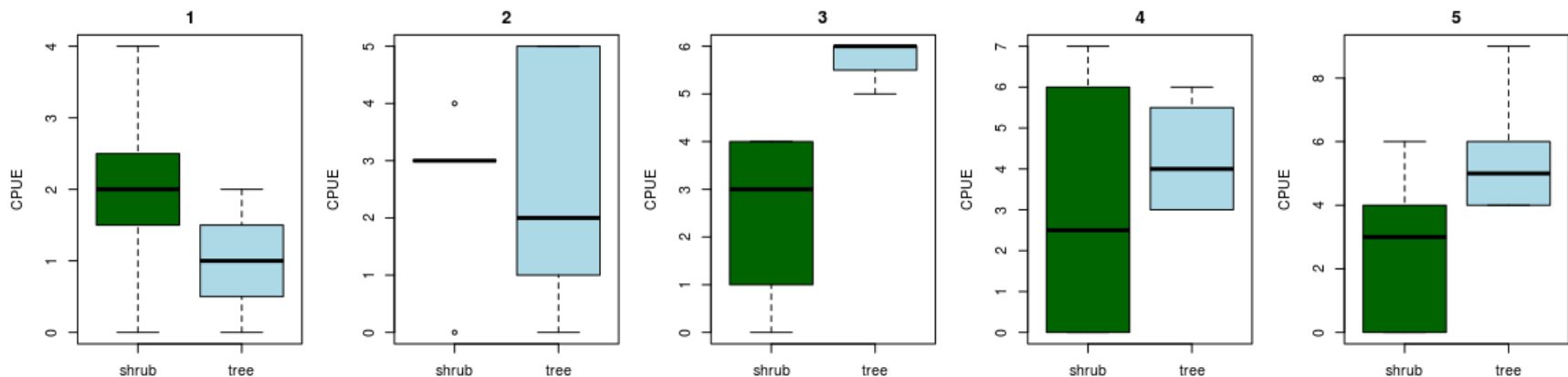
- 結果のプロット



繰り返し処理3

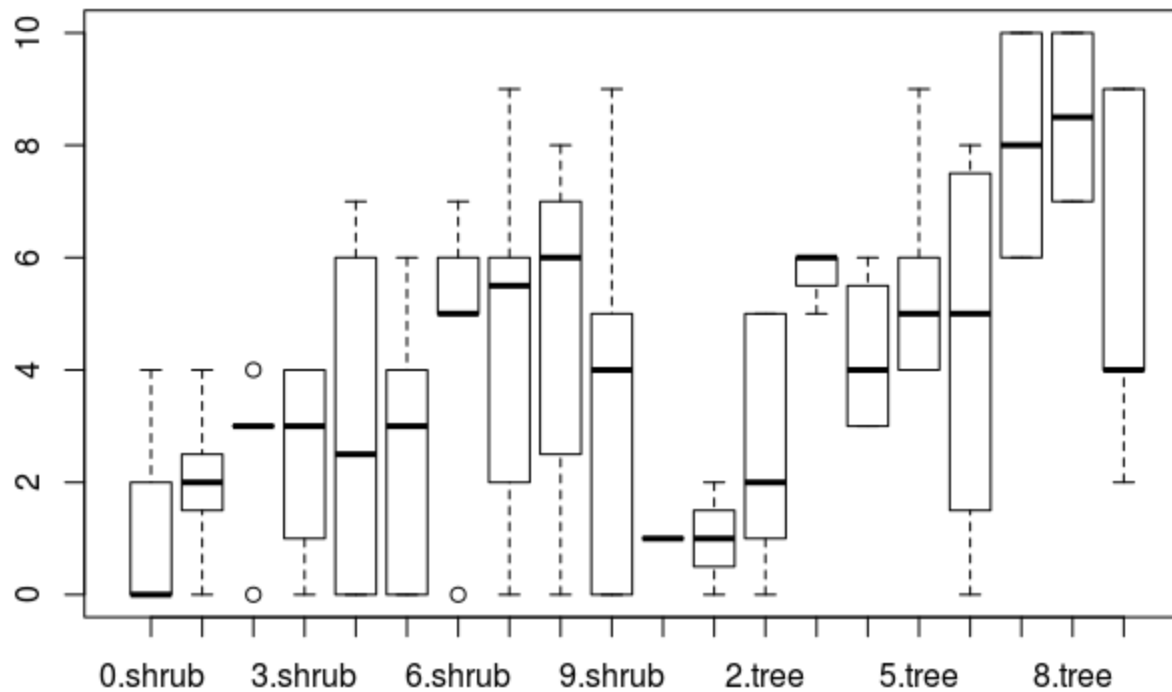
- forループを使うと？ => 複数の図を10枚でも100枚でも一気にかけます

```
par(mfrow = c(1, 5), mar = c(3, 4.3, 2, 1))
for (i in 1:5) {
  # iを1から5まで順番に変える
  cpue.site <- subset(cpue.data, site == i)
  boxplot(cpue.site$count ~ cpue.site$plant, ylab = "CPUE", c
    "lightblue"))
  title(i)
}
```



別のやり方1

- 1行でそれっぽいのもかけますが、



別のやり方2: ggplot2の利用

- ggplot2とは：きれいなグラフを簡単に書くためのパッケージ

```
library(ggplot2)
ggplot(data = cpue.data, aes(x = factor(year), y = count, fill =
  facet_grid(~. + plant) + theme_classic(base_size = 20, base
```