

STK372 Komputasi Statistik 2

Pemrograman Berorientasi Objek dalam R

Class System S3

Dr. Agus M Soleh



IPB University
— Bogor Indonesia —

agusms@apps.ipb.ac.id

Outline

- 1 Objek: Class System S3
- 2 Konstruktor dan Aksesori
 - Konstruktor
 - Aksesori
- 3 Fungsi Generik
- 4 Pewarisan Class
- 5 Transformasi dengan Matematika
- 6 Operasi Aritmetika
- 7 Method Subset
- 8 Pemeriksaan Suatu Class Objek
- 9 Penutup Class System S3

Objek: Class System S3

Suatu Class dalam System S3 tidak didefinisikan dengan ketat

Fungsi `class` digunakan untuk menjadikan sebuah objek menjadi class yang diinginkan

```
> pts <- list (x=round(rnorm(5),2), y=round(rnorm(5),2))
> class(pts)
[1] "list"
> pts
$x
[1] -1.48 -0.74 -0.59  0.92 -0.36

$y
[1] -0.27  1.73  0.42  1.52 -0.17
```

Objek: Class System S3

Menjadikan `pts` sebagai class baru:

```
> class(pts) <- "coords"
> class(pts)
[1] "coords"
> pts
$x
[1] -1.48 -0.74 -0.59  0.92 -0.36

$y
[1] -0.27  1.73  0.42  1.52 -0.17

attr(,"class")
[1] "coords"
```

Konstruktor

Langkah sederhana dalam membuat objek dari suatu class sebelumnya sangat tidak dianjurkan karena nilai-nilai instan-nya mungkin tidak tepat

Sebuah fungsi konstruktor dibutuhkan untuk mengecek instan sesuai dengan objek

Misal:

- x dan y harus berupa numerik
- vektor tidak boleh NA, NaN, Inf
- vektor harus memiliki panjang yang sama

Konstruktor

Fungsi konstruktor untuk membuat class coords adalah:

```
> coords <- function(x, y) {  
+   if (!is.numeric(x) || !is.numeric(y) ||  
+       !all(is.finite(x)) || !all(is.finite(y)))  
+     stop("Titik koordinat tidak tepat!")  
+   if (length(x) != length(y))  
+     stop("Panjang koordinat berbeda")  
+   pts <- list(x=x, y=y)  
+   class(pts) = "coords"  
+   pts  
+ }  
> pts <- coords(x = round(rnorm(5), 2),  
+              y = round(rnorm(5), 2))
```

Aksesori

Untuk mengakses data dalam class `coords`, dapat menggunakan akses objek awalnya (`list`)

Tetapi secara formal tidak dianjurkan mengakses data secara langsung

Diperlukan suatu fungsi aksesori untuk mengakses data pada class `coords`

Dalam ilustrasi ini digunakan 2 fungsi:

```
> xcoords = function(obj) obj$x
> ycoords = function(obj) obj$y
> xcoords(pts)
[1] -0.63  0.18 -0.84  1.60  0.33
> ycoords(pts)
[1] -0.82  0.49  0.74  0.58 -0.31
```

Fungsi Generik

- Fungsi generik merupakan suatu method dari suatu class objek dalam R
- Fungsi generik bertindak untuk beralih memilih fungsi tertentu atau metode tertentu yang dijalankan sesuai dengan classnya
- Untuk mendefinisi ulang suatu fungsi generik digunakan syntax:
`method.class <- function () ekspresibaru`
- Terdapat beberapa fungsi generik yang sudah ada: print, plot, dll

Method print

Method print merupakan cara menampilkan data pada suatu objek Class System S3

```
> print.coords <- function(obj) {
+   print(paste("(",
+             format(xcoords(obj)),
+             ", ",
+             format(ycoords(obj)),
+             ") ", sep=""),
+         quote=FALSE)
+ }
> pts
[1] (-0.63, -0.82) ( 0.18,  0.49) (-0.84,  0.74)
[3] ( 1.60,  0.58) ( 0.33, -0.31)
```

Method length

- Fungsi length menghitung banyaknya anggota dari objek
- Pada ilustrasi ini, length akan menghitung banyaknya anggota list, sehingga kurang tepat

```
> length(pts)
[1] 2
```

- Definisi ulang untuk class coords:

```
> length.coords = function(obj)
+   length(xcoords(obj))
>
> length(pts)
[1] 5
```

Membuat Fungsi Generik Baru

Untuk membuat suatu method yang dapat diwariskan, maka method tersebut harus dijadikan fungsi generik

Misal akan dibuatkan method bbox: merupakan boundary box.

```
> bbox <- function(obj)
+   UseMethod("bbox") #menjadikan bbox sebagai fungsi generik
> bbox.coords <- function(obj) {
+   matrix(c(range(xcoords(obj)),
+               range(ycoords(obj))),
+           nc = 2, dimnames = list(
+             c("min", "max"),
+             c("x:", "y:")))
+ }
> bbox(pts)
      x:      y:
min -0.84 -0.82
max  1.60  0.74
```

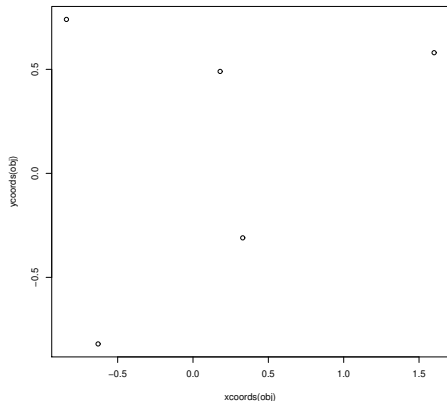
Method plot

Misalkan akan dibuatkan plot khusus untuk class coords:

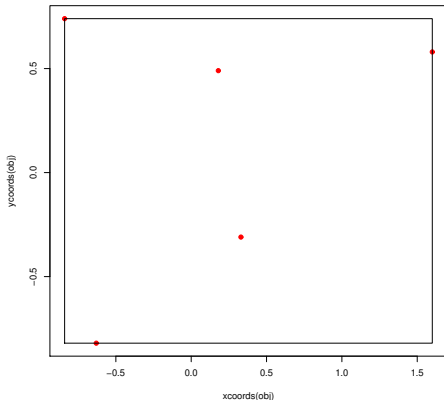
```
> plot.coords <- function(obj, bbox=FALSE, ...) {
+   if (bbox) {
+     plot(xcoords(obj), ycoords(obj), ...);
+     x <- c(bbox(obj)[1], bbox(obj)[2], bbox(obj)[2], bbox(
obj)[1]);
+     y <- c(bbox(obj)[3], bbox(obj)[3], bbox(obj)[4], bbox(
obj)[4]);
+     polygon(x, y)
+   } else {
+     plot(xcoords(obj), ycoords(obj), ...)
+   }
+ }
```

Method plot

```
> plot(pts)
```



```
> plot(pts, bbox=T, pch=19,  
       col="red")
```



Pewarisan Class

Diinginkan sebuah objek yang berisi lokasi (coords) dan terdapat nilai pada lokasi tersebut

Diperlukan menciptakan class baru vcoords sebagai turunan dari coords

Fungsi konstruktor dari class vcoords:

```
> vcoords <- function(x, y, v) {
+   if (!is.numeric(x) || !is.numeric(y)
+       || !is.numeric(v) ||
+       !all(is.finite(x)) ||
+       !all(is.finite(y)))
+     stop("Titik koordinat tidak tepat!")
+   if (length(x) != length(y) ||
+       length(x) != length(v) )
+     stop("Panjang koordinat berbeda")
+   pts <- list(x=x, y=y, v=v)
+   class(pts) = c("vcoords", "coords")
+   pts
+ }
> nilai <- function(obj) obj$v
```

Pewarisan Class

Perhatikan fungsi `xcoords` dan `ycoords` dan method `bbox` dari kelas `coords` masih sama sehingga tidak perlu didefinisi ulang

```
> vpts <- vcoords(x = round(rnorm(5), 2),
+                 y = round(rnorm(5), 2),
+                 v = round(runif(5,0,100)))
> vpts
[1] (-0.63, -0.82) ( 0.18,  0.49) (-0.84,  0.74)
[4] ( 1.60,  0.58) ( 0.33, -0.31)
> xcoords(vpts)
[1] -0.63  0.18 -0.84  1.60  0.33
> ycoords(vpts)
[1] -0.82  0.49  0.74  0.58 -0.31
> bbox(vpts)
      x:      y:
min -0.84 -0.82
max  1.60  0.74
```

Pewarisan Class

Method print juga masih diwariskan dari class coords tetapi perlu didefinisi ulang

```
> print.vcoords <- function(obj) {
+   print(paste("(",
+               format(xcoords(obj)),
+               ", ",
+               format(ycoords(obj)),
+               "; ", format(nilai(obj)),
+               ") ", sep=""),
+         quote=FALSE)
+ }
>
> vpts
[1] (-0.63, -0.82; 93) ( 0.18,  0.49; 21)
[3] (-0.84,  0.74; 65) ( 1.60,  0.58; 13)
[5] ( 0.33, -0.31; 27)
```


Pewarisan Class

Demikian juga method plot perlu didefinisi ulang

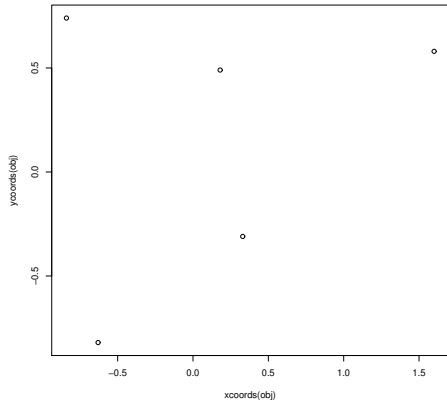
```

> plot.vcoords <- function(
+   obj, txt=FALSE, bbox=
+   FALSE, ...) {
+   if (bbox) {
+     if (!txt) {
+       plot(xcoords(obj),
+ ycoords(obj), ...);
+     } else {
+       plot(xcoords(obj),
+ ycoords(obj), type="n",
+ ...);
+       text(xcoords(obj),
+ ycoords(obj), nilai(obj),
+ ...);
+     }
+     x <- c(bbox(pts)[1],
+ bbox(pts)[2],bbox(pts)
+ [2],bbox(pts)[1]);
+     y <- c(bbox(pts)[3],
+ bbox(pts)[3],bbox(pts)
+ [4],bbox(pts)[4]);
+     polygon(x,y)
+   } else {
+     if (!txt) {
+       plot(xcoords(obj),
+ ycoords(obj), ...);
+     } else {
+       plot(xcoords(obj),
+ ycoords(obj), type="n",
+ ...);
+       text(xcoords(obj),
+ ycoords(obj), nilai(obj),
+ ...);
+     }
+   }
+ }

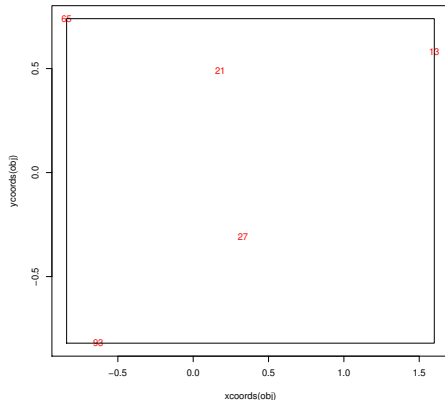
```

Pewarisan Class

```
> plot(vpts)
```



```
> plot(vpts, txt=T, bbox=T, col="red")
```



Tranformasi dengan Matematik

Objek `vcoords` berisi slot numerik yang dapat diubah dengan transformasi matematika

fungsi matematika untuk nilai (bukan titik koordinat)

```
> cos.vcoords <- function(obj)
+ {
+   vcoords(xcoords(obj),
+           ycoords(obj),
+           cos(nilai(obj)))
+ }
> cos(vpts)
[1] (-0.63, -0.82; 0.3174287)
[2] ( 0.18,  0.49; -0.5477293)
[3] (-0.84,  0.74; -0.5624539)
[4] ( 1.60,  0.58;  0.9074468)
[5] ( 0.33, -0.31; -0.2921388)

> sin.vcoords <- function(obj)
+ {
+   vcoords(xcoords(obj),
+           ycoords(obj),
+           sin(nilai(obj)))
+ }
> sin(vpts)
[1] (-0.63, -0.82; -0.9482821)
[2] ( 0.18,  0.49;  0.8366556)
[3] (-0.84,  0.74;  0.8268287)
[4] ( 1.60,  0.58;  0.4201670)
[5] ( 0.33, -0.31;  0.9563759)
```

Method Group: Math

- Class `vcoords` memungkinkan untuk dilakukan operasi aritmetika terhadap nilainya
- Hampir semua fungsi matematika menggunakan cara yang sama
- Langkah lebih cepat menggunakan fungsi `get(.Generic)`
- Ekspresi `get(.Generic)` mengambil fungsi dengan nama `Math.vcoords` yang dijalankan

```
> Math.vcoords <- function(obj) {  
+   vcoords(xcoords(obj),  
+         ycoords(obj),  
+         get(.Generic)(nilai(obj)))  
+ }
```

Method Group: Math

```
> sqrt(vpts)
```

```
[1] (-0.63, -0.82; 9.643651)
[2] ( 0.18,  0.49; 4.582576)
[3] (-0.84,  0.74; 8.062258)
[4] ( 1.60,  0.58; 3.605551)
[5] ( 0.33, -0.31; 5.196152)
```

```
> log(vpts)
```

```
[1] (-0.63, -0.82; 4.532599)
[2] ( 0.18,  0.49; 3.044522)
[3] (-0.84,  0.74; 4.174387)
[4] ( 1.60,  0.58; 2.564949)
[5] ( 0.33, -0.31; 3.295837)
```

Operasi Aritmetika: Method Groups Ops

Method Groups Ops merupakan fungsi generik yang digunakan untuk mendefinisikan semua operasi biner:

- "+", "-", "*", "/", "^", "%%", "%/%",
- "&", "|", "!",
- "==", "!=", "<", "<=", ">=", ">".

Supaya berjalan secara tepat diperlukan lokasi dari vcoords yang dioperasikan adalah identik

```
> sameloc <- function(e1, e2) {
+   (length(nilai(e1))==length(nilai(e2)) ||
+     all(xcoords(e1) == xcoords(e2)) ||
+     all(ycoords(e1) == ycoords(e2)))
+ }
+ }
```

Operasi Aritmetika: Method Groups Ops

```

> Ops.vcoords <- function(e1, e2) {
+   if (!sameloc(e1, e2))
+     stop ("Lokasi berbeda")
+   else vcoords(xcoords(e1),
+                ycoords(e1),
+                get (.Generic)(nilai(e1),
+                               nilai(e2)))
+ }
> vpts+vpts
[1] (-0.63, -0.82; 186) ( 0.18,  0.49;  42)
[3] (-0.84,  0.74; 130) ( 1.60,  0.58;  26)
[5] ( 0.33, -0.31;  54)
> vpts*vpts
[1] (-0.63, -0.82; 8649) ( 0.18,  0.49;  441)
[3] (-0.84,  0.74; 4225) ( 1.60,  0.58;  169)
[5] ( 0.33, -0.31;  729)

```

Method Subset

- Diinginkan untuk memiliki akses terhadap metode subset
- Ekspresi berikut misal diinginkan:

```
vpts[xcoords(vpts) < 0 & ycoords(vpts) < 0]
```

- Dapat ditangani dengan mendefinisikan metode untuk [

```
> `[.vcoords` <- function(x, i) {
+   vcoords(xcoords(x)[i], ycoords(x)[i],
+           nilai(x)[i])
+ }
> vpts[1:3]
[1] (-0.63, -0.82; 93) ( 0.18,  0.49; 21)
[3] (-0.84,  0.74; 65)
```


Pemeriksaan Suatu Class Objek

- Untuk mengecek apakah suatu objek merupakan suatu class digunakan fungsi inherits

```
> inherits(pts, "coords")  
[1] TRUE  
> inherits(pts, "vcoords")  
[1] FALSE  
> inherits(vpts, "coords")  
[1] TRUE  
> inherits(vpts, "vcoords")  
[1] TRUE
```

Penutup Class System S3

- Class System S3 memberikan fasilitas object-oriented, tetapi terlalu longgar
- Ekspresi berikut diperbolehkan dalam R, padahal class "lm" merupakan class untuk pemodelan linier

```
> model <- 1:10; class(model) <- "lm"
```

- Masih banyak technical issue dalam Class System S3

Lanjut ke materi 3b...