

# STK372 Komputasi Statistik 2

## Fungsi-fungsi Matematik dan Statistik

Dr. Agus M Soleh



**IPB University**  
— Bogor Indonesia —

*agusms@apps.ipb.ac.id*

# Outline

- 1 Himpunan
- 2 Fungsi-fungsi Matematika
  - Fungsi-fungsi Matematika umum
  - Fungsi-fungsi lain
  - Fungsi Kalkulus: Diferensial
  - Fungsi Kalkulus: Integral
- 3 Fungsi-fungsi Statistika
  - Fungsi-fungsi Sebaran
- 4 Optimasi Numerik
  - Golden Section Search
  - Newton Raphson
  - Fungsi Optimasi Built-in

# Himpunan

R memiliki fungsi-fungsi himpunan yang diterapkan pada vektor:

- `union(x,y)`: nilai unik dari anggota  $x$  dan  $y$
- `intersect(x,y)`: anggota  $x$  yang juga anggota  $y$
- `setdiff(x,y)`: anggota  $x$  yang bukan anggota  $y$
- `setequal(x,y)`: apakah semua anggota  $x$  dan  $y$  sama
- `c %in% y`: apakah anggota vektor  $c$  ada di  $y$
- `choose(n,k)`: banyaknya kombinasi  $k$  dari  $n$
- `combn(n,k)`: matriks semua kemungkinan kombinasi  $k$  dari  $n$

# Teladan Himpunan

```
> x <- c(1,2,5)
> y <- c(5,1,8,9)
> union(x,y)
[1] 1 2 5 8 9
> intersect(x,y)
[1] 1 5
> setdiff(x,y)
[1] 2
> setdiff(y,x)
[1] 8 9

> setequal(x,y)
[1] FALSE
> setequal(x,c(1,2,5))
[1] TRUE
> 2 %in% x
[1] TRUE
> 2 %in% y
[1] FALSE
> choose(5,2)
[1] 10
```

# Fungsi-fungsi Matematika umum

R menyediakan beberapa fungsi matematik, seperti:

- `exp()`: fungsi eksponensial dengan basis natural ( $e$ )
- `log()`: logaritma dengan basis natural
- `log10()`: logaritma dengan basis 10
- `sqrt()`: akar kuadrat
- `abs()`: nilai absolut
- `sin()`, `cos()`, `dst`: fungsi trigonometri
- `round()`, `floor()`, and `ceiling()`: fungsi-fungsi bilangan integer
- `factorial()`: fungsi faktorial

## Fungsi-fungsi lain

Selain fungsi matematik di atas, terdapat pula beberapa fungsi umum yang sering digunakan, seperti:

- `min()` and `max()`: mendapatkan nilai minimum dan maksimum dari suatu vektor
- `which`: mendapatkan index vektor yang memenuhi kriteria tertentu
- `which.min()` and `which.max()`: mendapatkan index vektor untuk nilai minimum dan maksimum
- `pmin()` and `pmax()`: element-wise minimum dan maksimum dari suatu vektor
- `sum()` and `prod()`: jumlah dan perkalian elemen suatu vektor
- `cumsum()` and `cumprod()`: jumlah dan perkalian kumulatif dari elemen suatu vektor

## Fungsi Kalkulus: Diferensial

- R memiliki fungsi-fungsi untuk mendapatkan turunan dari suatu fungsi
- Fungsi yang digunakan merupakan fungsi dengan satu peubah
- Syntax: `D(expr, simbol)` atau `deriv(~fungsi, simbol)`
- Teladan:

```
> xfs <- expression(exp(x^2))
> D(xfs, "x")
exp(x^2) * (2 * x)
> xturunan <- deriv(~x^2, "x")
> x <- 2
> eval(xturunan)
[1] 4
attr(,"gradient")
x
[1,] 4
```

# Fungsi Kalkulus: Integral

- Selain turunan pada fungsi sederhana, R juga memiliki fungsi untuk mendapatkan luas wilayah suatu fungsi dengan integral
- Syntax: `integrate(fungsi, lower, upper)`

- Teladan:

```
> fs <- function(x) x^2
> integrate(fs,0,1)
0.3333333 with absolute error < 3.7e-15
```



# Fungsi-fungsi Sebaran

Fungsi sebaran statistik secara konvensional:

- *dnamasebaran*: density or probability mass function (pmf)
- *pnamasebaran*: cumulative distribution function (cdf)
- *qnamasebaran*: quantiles
- *rnamasebaran*: random number generation

| Distribution        | Name    | Parameters        |
|---------------------|---------|-------------------|
| Beta                | beta    | shape1<br>shape2  |
| Binomial            | binom   | size<br>prob      |
| Cauchy              | cauchy  | location<br>scale |
| $\chi^2$            | chisq   | df                |
| Exponential         | exp     | rate              |
| $F$                 | f       | df1<br>df2        |
| Gamma               | gamma   | shape             |
| Geometric           | geom    | shape             |
| Lognormal           | lnorm   | meanlog<br>sdlog  |
| Logistic            | logis   | location<br>scale |
| Negative Binomial   | nbinom  | size<br>prob      |
| Normal (Gaussian)   | norm    | mean<br>sd        |
| Poisson             | pois    | lambda            |
| Student's $t$       | t       | df                |
| Uniform             | unif    | min<br>max        |
| Weibull             | weibull | shape             |
| Empirical cdf       | ecdf    |                   |
| Box-percentile plot | bpplot  | list of vectors   |

## Teladan Fungsi-fungsi Sebaran

```
> dnorm(0,mean=1,sd=1)
[1] 0.2419707
> pnorm(0,mean=1,sd=1)
[1] 0.1586553
> qnorm(0.1586553,mean=1,sd=1)
[1] 1.903889e-07
> x <- rnorm(10,mean=1,sd=1)
> x
[1] 0.01145131 2.06410816 1.40860993 2.23693521
[5] 1.64049302 1.33972159 1.74834955 0.08836656
[9] 2.02392465 0.91503261
> cbind(mean(x),sd(x))
      [,1]      [,2]
[1,] 1.347699 0.7866923
```

# Optimasi Numerik

Beberapa metode statistik menggunakan metode pendugaan nilai optimum dari suatu fungsi tujuan

Contoh:

- Metode kemungkinan maksimum: mencari nilai maksimum dari fungsi kemungkinan (*likelihood*)
- Metode kuadrat terkecil: mencari nilai minimum dari fungsi jumlah kuadrat galat

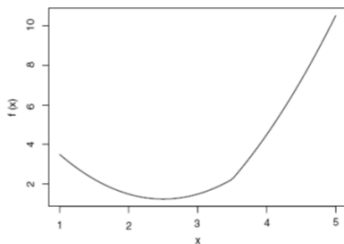
Mendapatkan nilai optimum dari suatu fungsi merupakan suatu teknik optimasi numerik

Beberapa metode yang sudah dikembangkan diantaranya:

- Golden section search
- Newton-Raphson
- Nelder-Mead
- dll

# Golden Section Search

- Mencari nilai minimum untuk fungsi peubah tunggal dari suatu selang yang diketahui
- Misal:  $f(x) = |x - 3.5| + (x - 2)^2$
- Langkah:
  - Mulai dengan selang  $[a, b]$  yang memuat minimum
  - Perkecil selang  $[a', b']$  yang memuat minimum
  - Berhenti sampai  $|b' - a'|$  lebih kecil dari nilai tolerans



# Golden Section Search

Pemilihan nilai  $a'$  dan  $b'$  adalah sebagai berikut:

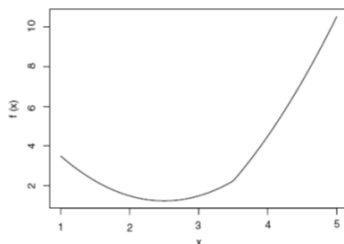
- Nilai antara  $[a, b]$  memiliki sifat golden ratio:  $\phi = (\sqrt{5} + 1)/2$
- Tentukan  $x_1$  dan  $x_2$ :  
$$x_1 = b - (b - a)/\phi$$
$$x_2 = a + (b - a)/\phi$$
- Hitung  $f(x_1)$  dan  $f(x_2)$
- Jika  $f(x_1) > f(x_2)$  maka  $[a', b'] = [x_1, b]$
- Jika  $f(x_1) < f(x_2)$  maka  $[a', b'] = [a, x_2]$

# Teladan Golden Section Search

```
golden <- function (f, a, b,
tol = 0.0000001) {
  ratio <- 2 / (sqrt(5)+1)
  x1 <- b - ratio * (b - a)
  x2 <- a + ratio * (b - a)
  f1 <- f(x1)
  f2 <- f(x2)
  while(abs(b - a) > tol) {
    if (f2 > f1) {
      b <- x2
      x2 <- x1
      f2 <- f1
    } else {
      a <- x1
      x1 <- x2
      f1 <- f2
      x2 <- a +ratio * (b - a)
      f2 <- f(x2)
    }
  }
  return((a + b) / 2)
}
```

# Teladan Golden Section Search

```
> f <- function(x) {  
+   abs(x-3.5) + (x-2)^2  
+ }  
> golden(f,1,2)  
[1] 2  
> golden(f,1,5)  
[1] 2.5  
> golden(f,3,5)  
[1] 3
```



# Newton Raphson

Jika suatu fungsi memiliki turunan pertama dan kedua, maka nilai minimum dapat menggunakan metode Newton Raphson:

$$x_n = x_{n-1} - \frac{f'(x_{n-1})}{f''(x_{n-1})}$$

Metode Newton Raphson lebih cepat dibanding Golden Section Search. Iterasi berhenti jika  $f'(x_{n-1})$  dekat dengan 0 atau lebih kecil dari nilai tolerans.

Fungsi nlm dalam R salah satunya yang mengimplementasikan Newton Raphson.



# Newton Raphson

```
newtonr <- function (fx, x0=1){  
  fx1 <- deriv(fx,"x")          # turunan pertama  
  fx2 <- deriv(D(fx,"x"),"x") # turunan kedua  
  e <- 1000  
  while (e>1e-6){  
    x <- x0  
    f1 <- attr(eval(fx1),"gradient")[1]  
    f2 <- attr(eval(fx2),"gradient")[1]  
    e <- abs(f1)  
    x1 <- x0 - f1/f2  
    x0 <- x1  
  }  
  return(x1)  
}
```

## Teladan Newton Raphson

Hitung nilai minimum untuk fungsi-fungsi:

- $f(x) = 4x^2 - 3x - 7$

- $f(x) = e^{-x} + x^4$

- $f(x) = x^2 - x$

adalah:

```
> fx <- expression(4*x^2 - 3*x - 7)
```

```
> newtonr(fx,3)
```

```
[1] 0.375
```

```
> fx <- expression(exp(-x) + x^4)
```

```
> newtonr(fx)
```

```
[1] 0.5282519
```

```
> fx <- expression(x^2 - x)
```

```
> newtonr(fx)
```

```
[1] 0.5
```

## Fungsi Optimasi Built-in

Optimasi Golden Section dan Newton Raphson sebelumnya digunakan untuk mendapatkan nilai optimum dari satu peubah.

Algoritma Nelder-Mead adalah salah satu metode optimasi untuk fungsi yang memiliki lebih dari satu peubah.

R telah menyiapkan fungsi optimasi dengan salah satu algoritmanya adalah Nelder-Mead:

- optimize atau optimise (satu peubah)
- optim (lebih dari satu peubah)

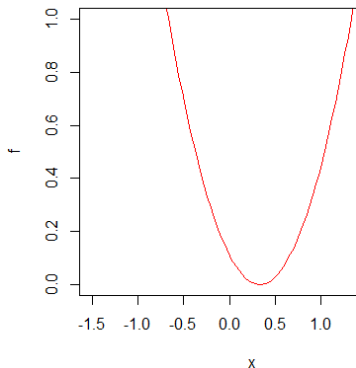
## optimize/optimise

Digunakan untuk mendapatkan nilai minimum dari suatu fungsi dengan satu peubah.

Misal akan dicari nilai minimum dari fungsi:  $f(x) = (x - 1/3)^2$

```
> f <- function (x, a) (x - a)^2
> xmin <- optimize(f, c(0, 1),
+ tol = 0.0001, a = 1/3)
> xmin
$minimum
[1] 0.3333333

$objective
[1] 0
```



## optim

Untuk mencari nilai minimum dari fungsi lebih dari satu peubah.

Misal mencari nilai  $x_1$  dan  $x_2$  yang membuat

$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$  minimum.

Program yang digunakan:

```
> fr <- function(x) {
+   x1 <- x[1]
+   x2 <- x[2]
+   100 * (x2 - x1^2)^2
+   + (1 - x1)^2
+ }
> optim(c(-1.2,1), fr)
```

Luaran dari fungsi optim:

```
$par
```

```
[1] 1.0002837 -0.1642825
```

```
$value
```

```
[1] 8.050091e-08
```

```
$counts
```

```
function gradient
```

```
47      NA
```

```
$convergence
```

```
[1] 0
```

```
$message
```

```
NULL
```

Akhir materi 2...