

OBJEK S

- ◆ Segala sesuatu dalam S adalah objek.
 - Data, fungsi, operator dll.
- ◆ Setiap objek memiliki *mode* dan *length*
 - Objek data --> mode *numeric, complex, character,* dan *logical*
 - Objek bahasa --> mode *function, expression, list,* dll
- ◆ Objek juga memiliki *attribute*
 - Menjelaskan fitur dari objek
 - Contoh attribut: *class, comment, dim, dimnames,* *names* dll
 - Gunakan perintah *attribute/attr* untuk melihat atau merubah atribut
 - Teladan :

```
# menciptakan matriks 2 x 5
x <- 1:10
attr(x,"dim") <- c(2, 5)
```
 - Gunakan perintah *structure* untuk menciptakan objek sekaligus atribut
 - Teladan : `structure(1:6, dim = 2:3)`
- ◆ Mode penyimpanan untuk objek numeric
 - integer, real, double precision
 - Biasanya seluruh objek numeric menggunakan mode double precision

OBJEK DATA

- ◆ Vektor
 - Vektor merupakan tipe sederhana dari objek data dalam S dalam hampir digunakan oleh semua objek data.
 - Semua elemen memiliki mode yang sama.
 - Untuk menciptakan vektor yang sederhana dapat menggunakan perintah concatenate (yaitu c)
 - Teladan :
 - `x <- c(1.4, 13.2, 10.9)`
 - `y <- c("sd", "smp", "sma", "pt")`
 - Perintah-perintah untuk membuat vektor selain perintah c:
 - scan
 - rep
 - seq
 - vector
 - as.vector
 - operator :
 - unlist (lihat list di bawah)
- ◆ Matriks
 - Matriks merupakan objek data yang banyak digunakan dalam analisis statistika

- Matriks pada bahasa S umumnya dibuat melalui vektor.
- Perintah-perintah untuk membuat matriks/mengkonversi vektor menjadi matriks:
 - matrix
 - dim
 - rbind
 - cbind
 - as.matrix
- ◆ Array
 - Matriks merupakan bentuk khusus dari Array. Perbedaan diantara keduanya adalah dimensinya.
 - Matriks adalah Array berdimensi dua.
 - Perintah untuk membuat Array adalah:
 - array
 - dim
 - as.array
- ◆ List
 - List merupakan objek data yang elemen didalamnya boleh memiliki mode yang berbeda.
 - Perintah-perintah untuk membuat list:
 - list
 - as.list
 - Umumnya list digunakan untuk menyimpan hasil

suatu fungsi

- Perintah `unlist` akan membuat list menjadi vektor.
- ◆ Faktor
 - Merupakan tipe khusus vektor dengan elemen data kategori
 - Faktor diprint tanpa diapit oleh *quotes* `""` --> memiliki metode print khusus
 - Pada dasarnya faktor adalah vektor yang mengindikasikan taraf (level) dari peubah kategori
 - Perintah-perintah untuk membuat faktor:
 - `factor`
 - `ordered`
 - `as.factor`
 - `as.ordered`
- ◆ Data Frames
 - Merupakan tipe khusus list yang digunakan untuk menyimpan data dari berbagai macam tipe dalam bentuk matriks.
 - Perintah-perintah untuk membuat data frames:
 - `data.frame`
 - `read.table`
 - `as.data.frame`

AKSES TERHADAP OBJEK DATA

◆ Akses Terhadap Vektor

- Jika x adalah objek vektor, maka akses terhadap elemen ke-i dilakukan melalui x[i]
- Index i berupa bilangan integer atau nama dari index atau elemen logical dari objek
- Bagaimana jika i negatif?
- Beberapa contoh:

```
> x <- c(1.5,2.3,NA,5.4,3.3)
> names(x) <- c('a','b','c','d','e')
> x
a b c d e
1.5 2.3 NA 5.4 3.3
> x[2]
b
2.3
> x[1:3]
a b c
1.5 2.3 NA
> x[c(2,4)]
b d
2.3 5.4
> x["d"]
d
5.4
> x[!is.na(x)]
a b d e
1.5 2.3 5.4 3.3
> x[x>3]
d e
NA 5.4 3.3
> x[x>3 & !is.na(x)]
d e
```

```

5.4 3.3
> x[-5]
a b c d
1.5 2.3 NA 5.4
> x[-(1:3)]
d e
5.4 3.3
> x[-c(2,4)]
a c e
1.5 NA 3.3

```

- ◆ Akses Terhadap Matriks
 - Jika mat adalah objek matriks, maka akses terhadap:
 - individual element : `mat[m,n]` atau `mat[i]`
 - baris : `mat[m,]`
 - kolom : `mat[,n]`
 - `m`, `n` dan `i` adalah bilangan integer
 - Bagaimana jika `m`, `n`, `i` adalah negatif?
- ◆ Akses Terhadap Array
 - Akses terhadap objek array, mirip dengan matriks
 - menggunakan operator `[]`
 - Output yang diperoleh bisa berupa vektor, matriks atau array dengan dimensi yang sama atau lebih rendah
- ◆ Akses Terhadap List
 - Akses terhadap list dapat dilakukan melalui beberapa cara.
 - By name : menggunakan operator `$`
 - By position : menggunakan operator `[[]]`
 - Sebagai vektor : menggunakan operator `[]`

- ◆ Akses Terhadap Faktor
 - lihat Vektor
- ◆ Akses Terhadap Data Frame
 - Akses terhadap data frame berlaku seperti matriks dan list
 - matriks : `dataolah[m,n]`
 - list : `dataolah$status` atau `dataolah[[2]]`

ARITMETIKA

- ◆ Aritmetika berhubungan dengan operator dan fungsi
- ◆ Operator-operator dalam S berdasarkan tingkatan:

Operator	Use
\$	component selection
[[]	subscripts, elements
^	exponentiation
-	unary minus
:	sequence operator
%% %/% %*%	modulus, integer divide, matrix multiply
* /	multiply, divide
+ -	add, subtract
<> <= >= == !=	comparison
!	not
& &&	and, or
-	formulas
<<- -> <- _	assignments

- ◆ Mana yg termasuk operator aritmetik?
- ◆ Beberapa fungsi sederhana:
 - Fungsi-fungsi Integer : round, trunc, floor, ceiling

- Fungsi umum : abs, log, log10, sqrt, exp, sin, cos, tan, acos, asin, atan, cosh, sinh, tanh
- Fungsi statistik : min, max, range, mean, var
- Fungsi utiliti : sort, rev
- ◆ Membangkitkan Bilangan Sekuensial
 - Sekuensial dalam integer:
 - 1:n, seq(1,n), n:1
 - Sekuensial umum:
 - x <- seq(a,b,by=0.01)
 - x <- seq(a,b,length=100)
 - Vektor berulang:
 - x <- 1:3
 - y <- rep(x,2) ekuivalen dgn c(1,2,3,1,2,3)
 - y <- rep(x,rep(2,3)) ekuivalen dgn c(1,1,2,2,3,3)
- ◆ Missing value
 - Elemen dari vektor yang tidak diketahui atau tidak tersedia direpresentasikan oleh nilai khusus : NA
 - Operasi dalam NA akan menghasilkan NA
 - x <- c(1.5, 2.3,NA)
 - mean(x)
 - [1] NA
 - Beberapa fungsi memiliki argumen na.rm atau yg mirip untuk tidak menyertakan NA dalam komputasi
 - Bedakan NA dengan NaN (Not a Number)

OPERASI-OPERASI OBJEK DATA

- ◆ Operasi dalam Vektor
 - Menggunakan element by element
 - Aturan Recycle
 - Operasi dengan vektor yang memiliki panjang berbeda
 - Vektor dengan elemen sedikit akan diulang mengikuti vektor yang memiliki elemen paling banyak
- ◆ Operasi dalam Matriks
 - Transpose : menggunakan fungsi t
 - Perkalian matriks : %*%
 - Vektor akan dipertimbangkan sebagai matriks
 $x^t x \rightarrow x \%* \% x$
 $xx^t \rightarrow x \%0 \% x$
 - Fungsi diag menghasilkan matriks diagonal atau mengambil diagonal matriks tergantung dari objek argumen
 - Beberapa fungsi matriks:
 - solve --> menghasilkan solusi sistem persamaan linier atau matriks invers
 - eigen --> menghasilkan nilai akar ciri dan vektor ciri

- ◆ Operasi dalam Karakter
 - Karakter dan string diapit oleh quotes (""")
 - Fungsi nchar menghitung banyaknya karakter dalam string
 - Fungsi paste menciptakan/menjadikan mode objek lain menjadi karakter
 - Fungsi substr/substring digunakan untuk mengambil bagian dalam string

FUNGSI-FUNGSI STANDAR

- ◆ Fungsi konversi ke Integer : round, trunc, floor, ceiling
- ◆ Fungsi umum : abs, sign, log, log10, sqrt, exp, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh
- ◆ Fungsi jumlah dan product dari vektor: sum, prod, cumsum, cumprod
- ◆ Fungsi Maksimum dan Minimum : max, min, cummax, cummin
- ◆ Fungsi range(x) menghitung c(max(x) , min(x))
- ◆ Fungsi untuk mengurutkan : sort
- ◆ Fungsi untuk membalikan : rev

STRUKTUR KONTROL

- ◆ Eksekusi bersyarat
 - if (kondisi)
 perintah-perintah jika kondisi T
else
 perintah-perintah jika kondisi=F
 - Mana yang menjadi operator pembanding dan Logik dalam daftar operator di atas?
 - ifelse(test, nilai jika T, nilai jika F)
 - Fungsi switch beberapa diantaranya akan lebih baik dibanding menggunakan if tersarang.
- ◆ Loops
 - for (variabel in sekuens) perintah-perintah
 - while (condition) perintah-perintah
 - repeat perintah-perintah (untuk menghentikan gunakan perintah break)

PERINTAH APPLY

- ◆ Loops sebaiknya dihindari → Gunakan perintah apply
- ◆ menjalankan fungsi untuk suatu array
- ◆ Syntax :

```
apply(array, margin, function, named  
arguments)
```

- ◆ margin : adalah subscript untuk vektor yang akan dikenakan fungsi
1 : baris, 2 : kolom, c(1,2) : baris dan kolom
- ◆ named arguments : optional untuk argument dari fungsi
- ◆ perintah lain yang sama : tapply, lapply, sapply

MEMBUAT FUNGSI

- ◆ Beberapa fungsi yang tidak ada dalam bahasa S dapat diciptakan sendiri.
- ◆ Syntax :
namafungsi <- function(argumen) isifungsi

Teladan:

```
std.dev <- function (x) { sqrt(var(x)) }
```

- ◆ Komentar bisa dituliskan menggunakan #

Teladan : Membuat Fungsi untuk Melakukan Pengujian Hipotesis Nilai Tengah untuk Dua Populasi dengan ragam sama.

Algoritma :

1. hitung masing-masing n
2. hitung galat baku
3. hitung statistik uji
4. hitung nilai-p
5. tetapkan keputusan penerimaan atau penolakan H_0

```
ttest <- function(y1, y2, test = "dua-arah", alpha = 0.05)
{
  n1 <- length(y1); n2 <- length(y2)
  ndf <- n1+n2-2
  s2 <- ((n1-1)*var(y1) + (n2-1)*var(y2))/ndf
  tstat <- (mean(y1)-mean(y2))/sqrt(s2*(1/n1+1/n2))
  tail.area <-
    switch(test,
      "dua-arah" = 2 * (1-pt(abs(tstat), ndf)),
      Kecil=pt(tstat,ndf),
      Besar= 1-pt(tstat,ndf),
      {
        warning ("uji harus 'dua-arah', 'Kecil' atau 'besar'")
        NULL
      }
    )
  list(tstat=tstat,df=ndf,reject=if(!is.null(tail.area))
      tail.area < alpha, tail.area=tail.area)
}
```

TUGAS

Buat Fungsi SWEEP2(k) untuk menjalankan fungsi operator Sweep seperti yang ada dalam SAS/IML
Algoritma SWEEP2(k):

Langkah 1 : Let $D = a_{kk}$

Langkah 2 : Divide row k by D

Langkah 3 : For every other row $i \neq k$, let $B = a_{ik}$.

Substract $B \times$ row k from row i.

Set $a_{ik} = -B/D$.

Langkah 4 : Set $a_{kk} = 1/D$