

# Function & Control Flow in R

*$\tau\rho$*

# Function

- Definition

→ These functions provide the base mechanisms for defining new functions in the **R** language

- General form

```
function( arglist ) expr  
  return(value)
```

# Example 1

Could you explain what is these function do for?

- `norm <- function(x) sqrt(x%*%x)`  
`norm(1:4)`
- `(function(x, y) { z <- x^2 + y^2; x+y+z })(0:7, 1)`

## Example 2

Create a function for making some NA entries in the matrix  $A$  based on the probability  $p$  !

# Answer 2

```
miss<-function(A,p)
{
  nb<-nrow(A)
  nk<-ncol(A)
  Am<-matrix(rbinom(nb*nk,1,p),nb,nk)
  Am1<-ifelse(Am==1,NA,1)
  Anew<-A*Am1
  return(Anew)
}
a<-matrix(rnorm(20),4,5)
miss(a,0.5)
```

# Example 3

Create a function for estimating the parameters of multiple regression !

# Answer 3

```
p.est<-function(A)
{
  if(!is.matrix(A))
    { stop("input must be on matrix")}
  x1<-A[,-1]
  y <-A[,1]
  one<-rep(1,nrow(A))
  x <-cbind(one,x1)
  colnames(x)<-paste("x",1:ncol(x),sep="")
  b.est<-solve(t(x)%*%x)%*%(t(x)%*%y)
  rownames(b.est)<-paste("b",0:(nrow(b.est)-1),sep="")
  b.estimation<-b.est
  fitted.value<-x%*%b.est
  error<-(y-fitted.value)
  list(beta.est=b.estimation,fit.val=fitted.value,error=error)
}

Pendapatan<-c(3.5,3.2,3.0,2.9,4.0,2.5,2.3)
Biaya.Iklan<-c(3.1,3.4,3.0,3.2,3.9,2.8,2.2)
Jumlah.Warung<-c(30,25,20,30,40,25,30)
X<-cbind(Pendapatan,Biaya.Iklan,Jumlah.Warung)
p.est(X)
lm(Pendapatan~Biaya.Iklan+Jumlah.Warung)
```

# Control Flow

- Description

→ These are the basic control-flow constructs of the **R** language

- General form

```
if(cond) expr
```

```
if(cond) cons.expr else alt.expr
```

```
for(var in seq) expr
```

```
while(cond) expr
```

```
repeat expr
```

```
break
```

```
next
```



# Example 4

Could you explain what is these programs do for?

- ```
for(n in c(2,5,10,20,50))  
  {  
    x <- rnorm(n)  
    cat(n, ": ", sum(x^2), "\n", sep = "")  }
```
- ```
f <- factor(sample(letters[1:5],10,replace=TRUE))  
  for(i in unique(f)) print(i)
```
- ```
x <- 10; jumlah<-0  
  while(x<10) {jumlah <- jumlah+x; x <- x+1}  
jumlah
```

# Example 5

Suppose matrix  $A = [0]_{n \times n}$ , create a function to built a matrix  $B$  from  $A$ , where

$$B_{n \times n} = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & \ddots & 0 & 1 & 0 \\ \vdots & 0 & 1 & 0 & \vdots \\ 0 & 1 & 0 & \ddots & 0 \\ 1 & 0 & \dots & 0 & 1 \end{bmatrix}$$

# Answer 5

```
B<-function(A)
{
  n<-nrow(A)
  for(i in 1:n)
  {
    for(j in 1:n)
    {
      if(i==j) A[i,j]<-1
    }
  }
  for(i in 1:n)
  {
    A[i,n-i+1]<-1
  }
  B<-A
  return(B)
}
a<-matrix(rep(0,25),5,5)
B(a)
```

# Example 6

Create a function for finding mean, median, and modus from  $n$  observations !

# Answer 6

```
three.M<-function(x)
```

```
{ n<-length(x)
```

```
  m1<-sum(x)/n
```

```
  y<-sort(x)
```

```
  p<-n/2
```

```
  if(p%%1==0){m2<-(y[p]+y[p+1])/2}
```

```
  else {m2<-y[ceiling(p)]}
```

```
  z<-unique(x)
```

```
  pz<-1:length(z)
```

```
  for(i in 1:length(pz))
```

```
  { a<-ifelse(x==z[i],1,0)
```

```
    pz[i]<-sum(a)
```

```
  }
```

```
  mpz<-max(pz)
```

```
  impz<-ifelse(pz==mpz,1,0)
```

```
  m<-z*impz
```

```
  m3<-m[m!=0]
```

```
  list(mean=m1,median=m2,modus=m3)
```

```
}
```

```
x<-c(5,8,6,9,10,6,4,8,3,7,7,11,2,9,6)
```

```
three.M(x)
```

Thank you 😊