E → VAR ARITHM_OP VAR | VAR ARITHM_OP VAL

Basic arithmetic functions

FUNCTION →

```
log2(VAR) # logarithms base 2 of x

log10(VAR) # logaritms base 10 of x

exp(VAR) # Exponential of x

cos(VAR) # Cosine of x

sin(VAR) # Sine of x

tan(VAR) #Tangent of x

acos(VAR) # arc-cosine of x

asin(VAR) # arc-sine of x

atan(VAR) #arc-tangent of x

abs(VAR) # absolute value of x

sqrt(VAR) # square root of x
```

STAT_FUNCTION →

```
max(VAR)

min(VAR)

range(VAR)

length(VAR)

sum(VAR)

prod(VAR)
```

```
mean(VAR)

sd(VAR) # Standard deviation

var(VAR)

sort(VAR)
```

Assigning values to variables

VARS → VAR
| VAR,VAR

VAR → CHARACTER COMB
| ._COMB
| .CHARACTER COMB

COMB → . | _
| CHARACTER
| D
| COMB COMB
| *eps*

/*

- *VAL → VECTOR # to add at the end*

*/

A → VAR ASSIGN EXP

ASSIGN → <- | =

PRINT → VAR | print(VAR)

LIST → ls()

REMOVE → rm(VARS)

Basic data types

BASIC_TYPE → LOGICAL
| NUMERIC
| STRING
| COMPLEX

COMPLEX → Di

LOGICAL → TRUE
| FALSE
| T
| F

NUMERIC → INTEGER | DOUBLE
INTEGER → DL
| DedL
| -DL
| -DedL
| +DL
| +DedL
d → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
D → dD
| d
DOUBLE → D | .D | D.D | D.Ded
| +D | +.D | +D.D | +D.Ded
| -D | -.D | -D.D | -D.Ded

STRING → “CHAINE”
| ‘CHAINE’
CHAINE → CHARACTER CHAINE
| CHAINE\’CHARACTER
| CHAINE\”CHARACTER
|CHARACTER
CHARACTER → a | b | c … | z | A | … | Z

TYPE → typeof(BASIC_TYPE)
| typeof(VAR)

TEST_TYPE → is.numeric(VAR)
| is.character(VAR)
| is.logical(VAR) | is.complex(VAR)

CONVERT → as.numeric(VAR)
| as.character(VAR)
| as.logical(VAR)
/*
 * Conversion d'un string to numeric est possible : returns NA (not available)
 */

Vectors

VECTOR → c(CL)
| c(CN)
| c(CS)
| c(CL)
| c(CV)
| c(CS,CN,TL)
| c(CS,CN)
| c(TS,TL)
| c(CN,CL)
| c(CNAMED)

CNAMED → CNAMED_N
| CNAMED_L
| CNAMED _S

CNAMED_N → CHAINE = NA

```
                        | CHAINE = NUMERIC
                        | CHAINE = NUMERIC , CNAMED_N
                        | CHAINE = NA, CNAMED_N

CNAMED_L →              CHAINE = NA
                        | CHAINE = LOGICAL
                        | CHAINE = LOGICAL , CNAMED_L
                        | CHAINE = NA, CNAMED_L

CNAMED_S →              CHAINE = NA
                        | CHAINE = STRING
                        | CHAINE = STRING , CNAMED_S
                        | CHAINE = NA, CNAMED_S

CHECK_NA  →             is.na(VAR)
CHECK_NAN →             is.nan(VAR)

CV →                VECTOR,CV
                    | VECTOR
CL →                LOGICAL,CL
                    | LOGICAL
CN →                NUMERIC,CN
                    | NUMERIC
CS →                STRING,CS
                    | STRING

ELEMENT_NAMES →         names(VAR) |
LENGTH →            length(VAR)

SUBSET_VECTOR →         var[D]
                        | var[D:D]
                        | var[c(D,D)]
                        | var[STRING]

EXCLUDE_ELEMENT → var[-D]
```

| var[-c(D,D)]
| VAR [-(D:D)]

SELECT_ELEMENT → var[var LOG_OP BASIC_TYPE]
| var [!CHECK_NA]

LOG_OP → ==
| !=
| >=
| <=
| <
| >

Matrices

VECTORS → VECTOR
| VECTOR,VECTORS

CREATE → rbind(VARS)
| rbind(VECTORS)
| cbind(VARS)
| cbind(VECTORS) # c for column and r for row
| matrix(data = VECTOR ,nrow = D , ncol = D , byrow = LOGICAL , dimnames = list(VECTORS))

RENAME → rownames(VAR)
| colnames(VAR)

TRANSPOSE → t(VAR)

DIMENSION → ncol(VAR)
| nrow(VAR)
| dim(VAR)

SUBSET_MATRIX → VAR[D,D]
| VAR[D,]
| VAR[D:D,]
| VAR[D:D,D:D]
| VAR[VECTOR,]
| VAR[,D]
| VAR[,D:D]
| VAR[,VECTOR]
| VAR[VECTOR,VECTOR]

SELECT → VAR[D,D]
| VAR[D,]
| VAR[,D]
| VAR[STRING,STRING]
| VAR[STRING,]
| VAR[,STRING] VAR[STRING,D]
| VAR[D,STRING]
| VAR[VAR LOG_OP BASIC_TYPE,]
| VAR[VAR LOG_OP BASIC_TYPE,VAR LOG_OP BASIC_TYPE]
| VAR[,VAR LOG_OP BASIC_TYPE]

EXCLUDE → VAR[-D,-D]
| VAR[-D,]
| VAR[,-D]

SPEC_FUNCTIONS → rowSums(VAR)
| colSums(VAR)
| apply(VAR,1,STAT_FUNCTION)
|apply(VAR,2,STAT_FUNCTION)

Factors

Data frames