

# Valid Expressions

---

$S \rightarrow$    HELP  
          | PRINT  
          | LEVELS AEXP<sub>1</sub>  
          | LIST  
          | SUBSET\_DATAFRAME AEXP<sub>2</sub>  
          | REMOVE  
          | ELEMENT\_NAMES  
          | A  
          | EXP  
          | READ  
          | WRITE  
          | SAVE

---

$V \rightarrow$        VAR V'

$V' \rightarrow V_1 \mid [V_2]$

$V_1 \rightarrow$        ASSIGN EXP  
              | eps  
              | \$SDL

$SLD \rightarrow$  VAR SLD<sub>1</sub>

$SLD_1 \rightarrow$  eps | SD<sub>11</sub>

$V_2 \rightarrow$        -EE<sub>2</sub>  
              | SE<sub>2</sub>  
              | SM<sub>1</sub>  
              | S<sub>2</sub>  
              | EX<sub>2</sub>  
              | AFTER\_VAR  
              | SD<sub>2</sub>  
              | [SL<sub>2</sub>

$AEXP_1 \rightarrow \text{ASSIGN VECTOR} \mid \text{eps}$

$AEXP_2 \rightarrow \text{ASSIGN VAR} \mid \text{eps}$

## Help

---

**HELP**  $\rightarrow$  ?CHaine | help(H1

H1  $\rightarrow$  CHaine)

| "H2

H2  $\rightarrow$  CHaine") | \*)

## Basic arithmetic operations

---

$E \rightarrow E E' \mid \text{VAR} \mid \text{NUMERIC}$

$E' \rightarrow + T \mid - T \mid T$

$T \rightarrow T T' \mid F \mid \text{VAR} \mid \text{NUMERIC}$

$T' \rightarrow * F \mid / F \mid ^ F \mid \% F$

$F \rightarrow (E) \mid D$

## Basic arithmetic functions

---

**FUNCTION**  $\rightarrow$  mode (VAR)

cat (VAR)

length (VAR)

log2 (VAR) # logarithms base 2 of x

log10 (VAR) # logarithms 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

---

**VAR** → VAR VAR2

**VAR2** → eps|,VAR

**VAR** → CHARACTER COMB  
| .VAR3

**VAR3** → \_COMP | CHARACTER COMB

**COMB** → . COMB1

| CHARACTER COMB1  
| D COMB1  
| *eps*

**COMB1** → *eps* | COMB

**A** → RENAME ASSIGN VECTOR

**ASSIGN** → <- | =

**EXP** -> V  
BASIC\_TYPE  
| VECTOR  
| FUNCTION  
| STAT\_FUNCTION  
| TYPE  
| TEST\_TYPE  
| CONVERT  
| CREATE\_MATRIX  
| TRANSPOSE  
| DIMENSION  
| SPEC\_MATRIX\_FUNCTION  
| CREATE\_FACTOR  
| INDIVID\_PER\_LEVEL  
| SPEC\_FACTOR\_FUNC  
| CREATE\_DATAFRAME  
| SEQ  
| RSEQ  
| CREATE\_LIST

**PRINT** → print(VAR)

**LIST** → ls()

**REMOVE** → rm(VARS)

# Basic data types

---

**BASIC\_TYPE**  $\rightarrow$  LOGICAL  
| NUMERIC  
| STRING  
| COMPLEX

**COMPLEX**  $\rightarrow$  Di

**LOGICAL**  $\rightarrow$  T LOGICAL<sub>3</sub> | F LOGICAL<sub>2</sub>

LOGICAL<sub>2</sub>  $\rightarrow$  eps | ALSE

LOGICAL<sub>3</sub>  $\rightarrow$  eps | RUE

**NUMERIC**  $\rightarrow$  INTEGER | DOUBLE

**INTEGER**  $\rightarrow$  D INTEGER<sub>2</sub> | NEG\_INT | POS\_INT

INTEGER<sub>2</sub>  $\rightarrow$  L | e dL

NEG\_INT  $\rightarrow$  - D INTEGER<sub>2</sub>

POS\_INT  $\rightarrow$  +D INTEGER<sub>2</sub>

**d**  $\rightarrow$  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

**D**  $\rightarrow$  d D<sub>2</sub>

D<sub>2</sub>  $\rightarrow$  D | eps

**DOUBLE**  $\rightarrow$  .D | D DOUBLE<sub>2</sub>  
| POS\_DOUBLE  
| NEG\_DOUBLE

DOUBLE<sub>2</sub>  $\rightarrow$  eps | .D | .Ded

POS\_DOUBLE  $\rightarrow$  +D DOUBLE<sub>2</sub> | +.D

NEG\_DOUBLE  $\rightarrow$  -D DOUBLE<sub>2</sub> | -.D

**STRING**  $\rightarrow$  "CHaine"  
| 'CHaine'

**CHaine**  $\rightarrow$  CHARACTER CHaine  
| CHARACTER\ CHaine<sub>2</sub>

| CHARACTER

CHaine2 → eps | 'CHaine | \"CHaine

**CHARACTER** → a | b | c ... | z | A | ... | Z

**TYPE** → typeof(TYPE2)  
| class(VAR)

TYPE2 → BASIC\_TYPE  
| VAR

**TEST\_TYPE** → is.IS

IS → numeric(VAR)  
| character(VAR)  
| logical(VAR)  
| complex(VAR)  
| na(VAR)  
| nan(VAR)  
| factor(VAR)  
| dataframe(VAR)

**CONVERT** → as.AS

AS → numeric(VAR)  
| character(VAR)  
| logical(VAR)  
| factor(VAR)  
| dataframe(VAR)

/\*

\* Conversion d'un string to numeric est possible : returns NA (not available)

\*/

## Vectors

---

**VECTOR** → c(VECTOR2)  
 VECTOR2 → CL  
           | CN VECTOR 4  
           | CS VECTOR 3  
           | CV  
           | CNAMED

VECTOR3 → eps  
           | ,VECTOR33  
 VECTOR 33 → CN VECTOR333  
               | CL  
 VECTOR 333 → ,CL  
               | eps  
 VECTOR 4 → eps  
            | ,CL

**CNAMED** → VAR = CNAMED\_T

**CNAMED\_T** → NA CNAMED\_NA  
               CNAMED\_N  
               | CNAMED\_L  
               | CNAMED\_S

**CNAMED\_NA** → eps | , CNAMED\_NLS

**CNAMED\_NLS** → CNAMED\_N2  
                   | CNAMED\_L2  
                   | CNAMED\_S2

**CNAMED\_N** → | NUMERIC CNAMED\_N2

**CNAMED\_N2** → eps

| ,CNAMED\_N

**CNAMED\_L** → | LOGICAL CNAMED\_L2

**CNAMED\_L2** → eps  
| ,CNAMED\_L

**CNAMED\_S** → | STRING , CNAMED\_S2

**CNAMED\_S2** → eps | , CNAMED\_S

**CV** → VECTOR CV2

**CV2** → eps  
| ,CV

**CL** → LOGICAL CL2

**CL2** → eps  
| ,CL

**CN** → NUMERIC CN2

**CN2** → eps  
| ,CN

**CS** → STRING CS2

**CS2** → eps  
| , CS

**ELEMENT\_NAMES** → names(VAR)

**AFTER\_VAR2** → D AFTER\_VAR3  
| c(D,D)  
| STRING

**AFTER\_VAR3** → eps  
| :D



**EE2** → D  
| c(D,D)  
| (D:D)

**SE2** → VAR LOG\_OP BASIC\_TYPE  
| !CHECK\_NA  
| CHECK\_NA

**LOG\_OP** → ==  
| !=  
| >=  
| <=

## Matrices

---

**VECTORS** → VECTOR VECTORS2

VECTORS2 → eps | ,VECTORS

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

**RC** → VARS | VECTORS

**RENAME** → rownames(VAR)  
| colnames(VAR)

**TRANSPOSE** → t(VAR)

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

$$\text{SM}_1 \rightarrow \text{SMD} \mid \text{SMP} \mid \text{SMV}$$
$$\text{SMP} \rightarrow , \text{SMP}_2$$

SMP<sub>2</sub> → D SMP<sub>3</sub> | VECTOR

$$\text{SMP}_3 \rightarrow \text{eps} \mid :D$$

SMD  $\rightarrow$  D, SMD2

$$\text{SMD}_2 \rightarrow ,D \mid :D \text{ SMD}_3 \mid \text{eps} \mid :D,$$

SMD3  $\rightarrow$  eps | ,SMD4

SMD<sub>4</sub> → eps | D:D

SMV  $\rightarrow$  VECTOR SMV<sub>2</sub>

SMV2 → , VECTOR | eps

$$S_2 \rightarrow S_4 \mid S_2 \mid S_3 \mid S_5$$
$$S_3 \rightarrow \text{STRING } S_{33} | , \text{STRING } |$$

S33 → , | ,STRING | ,D

$$S_4 \rightarrow D, S_{44}$$

S44 → D|eps | STRING

$$S5 \rightarrow \text{VAR}[S55]$$
$$S_{55} \rightarrow \text{VAR LOG\_OP BASIC\_TYPE}, S_{555} \mid , \text{VAR LOG\_OP BASIC\_TYPE}$$

S555 → eps | ,VAR LOG\_OP\_BASIC\_TYPE

$$\text{EX2} \rightarrow \neg \text{EX3} \mid \neg \text{D}$$
$$\text{EX}_3 \rightarrow \text{D EX}_{33} \mid, \text{D}$$
EX33  $\rightarrow$  ,D |,

```
SPEC_MATRIX_FUNCTION →      rowSums(VAR)
                             | colSums(VAR)
                             | colMeans(VAR)
```

| rowMeans(VAR)  
| apply(VAR,APP)

APP → 1,STAT\_FUNCTION  
| 2,STAT\_FUNCTION

## Factors

---

**CREATE\_FACTOR** → factor(FP)  
FP → VECTOR VAR VAR,levels = VECTOR  
FP 2 → eps | ,levels = VECTOR

**INDIVID\_PER\_LEVEL** → summary(VAR)

**LEVELS** → levels(VAR)

**SPEC\_FACTOR\_FUNC** → tapply(VAR,VAR,STAT\_FUNCTION)  
| table(TP)

TP → VAR TP2  
TP2 → eps | ,VAR

## Data frames

---

**CREATE\_DATAFRAME** → data.frame(COLS)

**COLS** → COL COLS2  
COLS2 → eps | ,COLS

**COL** → VAR=COL2  
COL2 → VECTOR | VAR | BASIC\_TYPE

/\* You can use t() as same as Matrix to transpose a data frame\*/

**SUBSET\_DATAFRAME** → subset(VAR, **VAR** LOG\_OP  
BASIC\_TYPE)  
| attach(VAR),detach(VAR)

SD2 → ,SDP  
| VAR SD22  
SD22 → ,VAR | \$**VAR** LOG\_OP BASIC\_TYPE, SD222  
SD222 → eps | VECTOR | VAR,VAR  
SDP → D | STRING | VECTOR | -D  
SD11 → eps | LOG\_OP BASIC\_TYPE

## Sequences

---

**SEQ** → | seq(SEQP)  
| rep(D,D)  
| sequence(c(CN) )

SEQP → SEQ\_PARAM | D SEQD  
**SEQ\_PARAM** → length=D  
| label = c(CS)  
| from = D  
| to =D

SEQD → :D | ,D,D.D  
SEQR → eps

Random sequences :

**RSEQ** → PFUNC(DISTRIB\_PARAMS)

**P** → r  
| d  
| p  
| q

**FUNC** → norm | exp | gamma | nbinom | unif | geom | cauchy | pois | f |  
t | logis

**DISTRIB\_PARAMS** → | D.D DPD  
| scale = D  
| location = D  
| mean = D  
| rate = D

DPD → ,DISTRIB\_PARAMS | eps | .D

## Lists

---

**CREATE\_LIST** → list(COLS)

/\* element\_names and length already exists \*/

SL2 → STRING]] | D SLD

SLD → ]] SLD2

SLD 2 → eps | [D

## Importing Data

---

**READ** → read.READ2

READ2 → delim(file.choose(STRING)) | csv(file.choose(STRING)) |  
csv2(file.choose(STRING))

## Exporting Data

---

**WRITE** → data(STRING)  
| write WRITE2

WRITE2 → table(VAR,PARAMS) | csv(VAR,PARAMS) |  
csv2(VAR,PARAMS)

**SAVE** → saveRDS(VAR,STRING)  
| readRDS(STRING)  
| load(STRING)  
| save SAVE2

SAVE2 → (VARS,file=STRING) | .image(file=STRING)

**PARAMS** → file = STRING  
| sep = "SEP"

| row.names = RCN

| col.names = RCN

RCN → LOGICAL | NA

**SEP** → , | ; | \t