Valid Expressions

S →   HELP
      | PRINT
      | A
      | LIST
      | REMOVE
      | EXP
      | ELEMENT_NAMES
      | READ
      | WRITE
      | SAVE

Help

HELP →   ?CHAINE | help(HELP_SUITE
H1 →   CHAINE)
        |"H2
H2 → CHAINE") | *")

Basic arithmetic operations

E → E E'
E' → + T | - T | T
T → T T' | F
T' → *F | /F | ^F | %% F
F → (E) | D

Basic arithmetic functions

FUNCTION →   `mode(VAR)`

```
                    cat(VAR)

                    length(VAR)

                    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 VAR2

VAR2 →   eps|,VAR

VAR →   CHARACTER COMB
             | .VAR3
VAR3 → _COMP | CHARACTER COMB

COMB → . COMB1
            | CHARACTER COMB1
            | D COMB1
            | *eps*

COMB1 → eps | COMB

A →   VAR ASSIGN EXP
            | RENAME ASSIGN VECTOR
            | LEVELS ASSIGN VECTOR
            | SUBSET_DATAFRAME ASSIGN VAR

ASSIGN →   <- | =

EXP ->   BASIC_TYPE
            | E
            | VECTOR
            | VAR
            | FUNCTION
            | STAT_FUNCTION
            | TYPE
            | TEST_TYPE
            | CONVERT
            | CHECK_NA
            | CHECK_NAN
            | SUBSET_VECTOR
            | EXCLUDE_ELEMENT

| SELECT_ELEMENT
| CREATE_MATRIX
| TRANSPOSE
| DIMENSION
| SUBSET_MATRIX
| SELECT
| EXCLUDE
| SPEC_MATRIX_FUNCTION
| CREATE_FACTOR
| CHECK_FACTOR
| CONVERT_FACTOR
| INDIVID_PER_LEVEL
| SPEC_FACTOR_FUNC
| LEVELS
| CREATE_DATAFRAME
| CONVERT_DATAFRAME
| SUBSET_DATAFRAME
| SPEC_DATAFRAME_FUNCTION
| SEQ
| RSEQ
| CREATE_LIST
| SUBSET_LIST

PRINT →   VAR | print(VAR)
LIST →   ls()
REMOVE →   rm(VARS)

Basic data types

BASIC_TYPE →   LOGICAL
                     | NUMERIC
                     | STRING
                     | COMPLEX

COMPLEX →   Di

LOGICAL →   T LOGICAL3| F LOGICAL2

LOGICAL2→ eps | ALSE
LOGICAL3→ eps | RUE

NUMERIC →   INTEGER | DOUBLE
INTEGER →   D INTEGER2 | NEG_INT | POS_INT

INTEGER2 → L | e dL
NEG_INT → - D INTEGER2
POS_INT → +D INTEGER2
d →   0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
D →   d D2
D2 → D | eps
DOUBLE →   .D | D DOUBLE2
                      | POS_DOUBLE
                      |NEG_DOUBLE
DOUBLE2 → eps | .D | .Ded
POS_DOUBLE → +D DOUBLE2 | +.D
NEG_DOUBLE → -D DOUBLE2 | -.D

STRING →   "CHAINE"
                  | 'CHAINE'
CHAINE →   CHARACTER CHAINE
                  | CHARACTER\ CHAINE2
                  |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)

CONVERT →   as.AS

AS → numeric(VAR) | character(VAR) | logical(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 →   CNAMED_N
                | CNAMED_L
                | CNAMED _S

CNAMED_N →   CHAINE = CNAMED_N1
CNAMED_N1 →   NA CNAMED_N2
                | NUMERIC CNAMED_N3
CNAMED_N2 → eps | ,CNAMED_N

CNAMED_N3 → eps | ,CNAMED_N

CNAMED_L →   CHAINE = CNAMED_L1
CNAMED_L1 →   NA CNAMED_L2
                 | LOGICAL CNAMED_L3
CNAMED_L2 → eps | ,CNAMED_L
CNAMED_L3 → eps | ,CNAMED_L

CNAMED_S →   CHAINE = NA
                 | CHAINE = STRING
                 | CHAINE = STRING , CNAMED_S
                 | CHAINE = NA, CNAMED_S
CNAMED_S1 →   NA CNAMED_S2
                 | STRING CNAMED_S3
CNAMED_S2 → eps | ,CNAMED_S
CNAMED_S3 → eps | ,CNAMED_S

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

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)

SUBSET_VECTOR → VAR AFTER_VAR
AFTER_VAR → [AFTER_VAR2]
AFTER_VAR2 → D AFTER_VAR3| c(D,D) | STRING
AFTER_VAR3 → eps | :D

EXCLUDE_ELEMENT → var[-EE2]
EE2 → D | c(D,D) | (D:D)

SELECT_ELEMENT → var[SE2]
SE2 → VAR LOG_OP BASIC_TYPE | !CHECK_NA | CHECK_NA

LOG_OP → ==
      | !=
      | >gt
      | < lg
lg → < | eps
gt → > | eps

Matrices

VECTORS → VECTOR
      | VECTOR,VECTORS

CREATE_MATRIX → 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_MATRIX_FUNCTION → rowSums(VAR)

| colSums(VAR)
| colMeans(VAR)
|rowMeans(VAR)
| apply(VAR,1,STAT_FUNCTION)
|apply(VAR,2,STAT_FUNCTION)

Factors

CREATE_FACTOR → factor(VECTOR)
| factor(VAR,levels = VECTOR)
| factor(VAR)

CHECK_FACTOR → is.factor(VAR)

CONVERT_FACTOR → as.factor(VAR)

INDIVID_PER_LEVEL → summary(VAR)

LEVELS → levels(VAR)

SPEC_FACTOR_FUNC → tapply(VAR,VAR,STAT_FUNCTION)
| table(VAR)| table(VAR,VAR)

Data frames

CREATE_DATAFRAME → data.frame(COLS)

COLS → COL
| COL,COLS

COL → CHAINE=VECTOR
| CHAINE = VAR

```
           | CHAINE = BASIC_TYPE
```

CHECK_DATAFRAME → is.data.frame(VAR)

CONVERT_DATAFRAME → as.data.frame(VAR)

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

SUBSET_DATAFRAME →   VAR$CHAINE
                         |VAR[,D]
                         | VAR[,STRING]
                         | VAR[,VECTOR]
                         | VAR[,-D]
                         | VAR$CHAINE LOG_OP
                         BASIC_TYPE
                         | VAR[VAR$CHAINE LOG_OP
                         BASIC_TYPE,]
                         | VAR[VAR$CHAINE LOG_OP
                         BASIC_TYPE, VECTOR]
                         | VAR[VAR,VAR]
                         | subset(VAR, CHAINE LOG_OP
                         BASIC_TYPE)
                          | attach(VAR),detach(VAR)

SPEC_DATAFRAME_FUNCTION (same as
SPEC_MATRIX_FUNCTION)

Sequences

SEQ →   seq(D,D,D.D)
           | seq(SEQ_PARAM)
           | rep(D,D)
           | seq(D:D) | sequence(c(CN))

SEQ_PARAM →   length=D
                  | label = c(CS)
                  | from = D
                  | to =D
                  | SEQ_PARAM,SEQ_PARAM

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 → DISTRIB_PARAMS
                          | DISTRIB_PARAMS, DISTRIB_PARAMS
                          | D
                          | D.D
                          | scale =D
                          | location = D
                          | mean = D
                          | rate = D

Lists

CREATE_LIST →   list(COLS)

/* element_names and length already exists */

SUBSET_LIST →   VAR$CHAINE

               | VAR[[STRING]]

               | VAR[[D]]

               | VAR[[D]][D]

Importing Data

READ →   read.delim(file.choose(STRING))

           | read.csv(file.choose(STRING))

           | read.csv2(file.choose(STRING))

           | read.tsv(file.choose(STRING))

Exporting Data

WRITE →   data(STRING)

              | write.table(VAR,PARAMS)

              | write.csv(VAR,PARAMS)

              | write.csv2(VAR,PARAMS)

SAVE →   saveRDS(VAR,STRING)

              | readRDS(STRING)

              | load(STRING)

| save(VARS,file=STRING)

|save.image(file=STRING)

PARAMS →   file = STRING

| sep = "SEP"

| row.names = LOGICAL

| col.names = LOGICAL

| row.names = NA

| col.names = NA

SEP →   , | ; | \t