

# Operations in R

Jan Rovny

## Objects in R

### Arrays, Vectors, Data Frames, and Matrices.

R distinguishes between vectors, data frames, and matrices. Vectors are indexed by length and matrices are indexed by rows and columns. Data frames are a matrix that R designates as a data set. With a data frame, the columns of the matrix can be referred to as variables.

We can create objects directly in R:

```
a<-7 #this creates a scalar *a* with a value of 7
x<-c(1,2,3,4,5,6,7,8,9,10) #this creates a vector x with 10 values
y<-c(13,45,23,78,-5.2,4,43,8,12,-3) #this creates a vector y with 10 values
A<-matrix(#this creates a matrix A
  c(2, 4, 3, 1, 5, 7, 5, 6, 8), # the data elements
  nrow=3, # number of rows
  ncol=3, # number of columns
  byrow = TRUE) # fill matrix by rows
```

We can then ask R to show us these objects by simply calling on them:

```
a
## [1] 7
x
## [1] 1 2 3 4 5 6 7 8 9 10
y
## [1] 13.0 45.0 23.0 78.0 -5.2 4.0 43.0 8.0 12.0 -3.0
A
##      [,1] [,2] [,3]
## [1,]  2   4   3
## [2,]  1   5   7
## [3,]  5   6   8
```

We can also call the specific elements of matrix A:

```
A[,2] # 2nd column of matrix
## [1] 4 5 6
A[3,] # 3rd row of matrix
## [1] 5 6 8
A[2,2] #2nd row, 2nd column
## [1] 5
A[1:2,1:3] # rows 1 to 2 of columns 1 to 3
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    3
## [2,]    1    5    7
```

## Datasets as objects

After reading in a data set, R will treat your data as a data frame. Unlike in other statistical programs, R allows us to work with multiple data frames at the same time. This, however, means that when we wish to refer to a specific variable, we must identify it by its data frame name and variable name. We normally do this by saying:

```
data.frame$variable.name
```

As an example, let's look at a section from the European Social Survey. Let's first load this dataset from a website.

```
library(rio)
D<-import("https://jan-rovny.squarespace.com/s/ESS_FR.dta")
```

We have now created the object 'D' in our R memory. We can, for example, see which party a respondent would vote for. To do this, we print the variable 'partvtcfr' in the dataset 'D':

```
D$partvtcfr
```

```
##      [1] NA 11 11 NA NA NA 10  2  9 10  9 NA NA 10 NA 10 NA  9 NA  9 11 12 10
##      [24] NA 12  9  9 NA NA NA 10  6 NA  9 NA 11 NA NA 10  9 NA NA NA NA NA  9
##      [47]  9 NA 10 NA NA 10  9 10 10 10  9 11 NA 10 10 10 NA NA  1 NA 14 11 NA
##      [70] NA  9  9  9  9 NA NA  6 NA NA NA 10 10  9  7 11 NA  9 12  9 NA NA NA
##      [93] NA NA  9 10 10 NA 10 10 NA 10  9 NA 10 10 10 NA NA 10  8 10  9 10 NA
##     [116]  9 NA NA NA NA  9  9  9 NA  2 NA  9 10  9 10 NA NA 10 10  9 NA NA NA
##     [139] 10 10 NA  2 NA NA NA 12  9 NA 10  1 10 NA NA  9 NA 10 NA NA NA NA NA
##     [162] NA NA NA NA NA  9 NA  9  9 12 NA 13 NA  2  6 NA  9 NA NA 10  9 10 NA
##     [185] 11 NA 11 NA 10 10 NA NA NA NA 11  9 NA  9 10 NA NA NA  9 NA NA  6 NA
##     [208] NA 10 10  9  9 NA 12 NA  9  9 10  9 NA NA NA  9  1 10 10  6  9 11 NA
##     [231]  6 NA NA  9 NA NA  9 NA  9 10 NA 10 NA NA NA  9 15  9 10  9 10 NA NA
##     [254] 10  9 NA NA 10 NA NA NA NA 10  4 NA NA NA  2  2 NA NA 10 10  9 10  9
##     [277] NA NA  9  2 10  2  9  1 NA  9  6  2 10 NA NA  6 NA NA  2  9 NA NA 12
##     [300] NA NA  2  9 NA 11 NA 14 NA  2  9 NA  9 NA NA 10 NA  6 11 NA NA NA NA
##     [323] NA NA  9  9 NA NA NA 11 15 NA NA  2 10  9 NA NA 11 NA 10 NA NA NA NA
##     [346] NA NA NA NA NA 10 NA NA  2 NA NA NA NA NA NA 10 11  2 10  9  9 NA 10
##     [369] NA NA NA NA 10  9 NA 15 NA 10  2 NA  9 10 11 NA  2 13 NA 15  6 10 NA
##     [392] NA NA 10 11 10  9 NA  9 NA  4  9 10 NA 10 NA 12 11 NA  6 12  9  9 NA
##     [415]  2 NA NA 10 NA 10 NA  9 11  9 NA  9 NA NA NA  2 NA 11 NA 14 NA  9 NA
##     [438]  9 10  2 NA 10 NA  9 10  7 NA 12 10 NA 12 10 NA  9 NA 10 10 10 11 NA
##     [461] NA  2 NA 10 10 11  9 NA  9 NA  9 NA NA NA NA NA 15 NA  1 NA NA  9  2
##     [484] NA NA NA 12  2  2  9  2 10  2 10 NA NA NA NA NA NA  2 NA 12  9  2 10 NA
##     [507]  9 11  9 NA NA NA 15 NA 10  6 NA NA  9 15 10 13  9 NA  9 NA  7  9 10
##     [530] NA NA NA  5 10 NA 10  9 NA NA 10  1 NA  9 10 NA NA 10 NA  9  9 10 NA
##     [553] 10 NA  2 NA NA 10  2 16 NA  9 NA NA  9 NA NA  9 NA  9 10 NA NA 10 NA
##     [576]  9  7 NA NA  9 NA NA NA  9 10 NA  9 10 10  9 12  2 NA NA NA 10 NA  9
##     [599] NA 10 NA 10 15  7 10 10  9 10 10 NA 10  9 10 NA 15 NA  9  2  2  2  4
##     [622]  6 NA  7  2 NA NA NA  9 10  2 NA NA NA 14  9 NA  9 15 NA  9  9 10  2
##     [645] NA 10  9  2 NA NA 12 NA NA NA  2 NA NA NA NA NA 12 NA NA NA 10 NA NA
##     [668]  9 10 NA NA NA NA 10  9 NA  9  2  9 NA  2  6 NA NA  9 NA  2  9 NA NA
##     [691] NA  6  9 NA NA  6  9 16  2 NA 10  9 NA  6  9 10 NA 10 NA NA  9 NA NA
```

```

## [714] 5 NA NA 9 8 NA NA NA 15 NA 9 9 9 NA 9 9 NA NA 10 NA 12 9 10
## [737] 9 9 NA 10 NA 2 10 11 9 9 NA NA 10 11 NA NA 13 NA NA NA NA 10 9
## [760] 9 10 NA 9 10 10 NA NA 9 9 NA 9 10 9 10 10 14 NA 12 2 9 NA 1
## [783] NA 10 NA NA NA 10 9 NA 12 NA 11 9 10 NA NA NA 9 9 NA NA 12 5 NA
## [806] 11 9 NA 10 NA NA NA 10 NA NA NA 10 NA 10 12 NA 2 NA 10 2 NA NA 10
## [829] 10 NA 10 11 NA 12 NA NA NA NA 12 11 9 2 10 NA NA NA NA 9 NA NA NA
## [852] NA 12 9 7 10 2 2 NA NA NA NA NA 5 NA 10 NA 12 9 12 10 10 NA NA
## [875] 9 NA NA 2 9 NA 12 9 9 11 10 NA NA 13 15 8 9 NA 9 NA 2 NA NA
## [898] 11 NA NA 11 12 9 NA 9 9 9 NA 4 10 11 NA 9 NA NA 9 NA NA NA 10
## [921] 10 NA 11 13 NA NA 9 8 10 NA 1 NA 9 9 9 9 NA 11 NA NA NA NA 11
## [944] 10 10 NA 2 11 NA 9 NA NA NA 6 NA 11 NA NA 10 10 15 NA NA NA 9 9
## [967] 10 NA NA 12 9 11 NA 2 NA NA 9 9 9 16 NA NA 10 12 NA 10 6 NA NA
## [990] NA NA 9 NA 6 NA 9 10 9 2 10 NA NA NA 12 11 15 10 12 15 2 NA NA
## [1013] NA 9 10 5 NA 12 9 NA NA 9 NA 9 NA 9 8 10 8 9 10 NA NA NA 12
## [1036] 7 11 10 NA NA NA NA 10 10 NA 12 NA NA NA NA NA 9 9 12 9 9 NA NA
## [1059] NA NA NA 10 NA 9 NA NA NA 9 9 10 9 NA NA NA 2 9 NA 9 2 8 2
## [1082] NA NA 15 10 9 9 9 NA 11 9 2 10 10 10 NA 10 10 NA 10 10 NA NA NA
## [1105] NA 9 12 2 12 2 NA 10 10 12 9 NA 12 NA NA NA 6 10 9 9 NA NA NA
## [1128] NA 10 12 9 10 NA 10 14 NA 10 NA 2 NA NA 9 10 15 10 10 NA 15 9 11
## [1151] 6 NA 9 10 NA NA NA NA NA NA NA NA 10 NA 7 NA NA 9 10 NA 9 NA NA
## [1174] 10 10 5 NA 10 14 12 9 NA 14 14 10 NA 9 NA NA 12 NA NA NA NA NA NA
## [1197] 9 9 9 13 11 9 2 NA 10 15 2 1 NA 9 10 10 9 NA 11 NA NA NA 13
## [1220] NA 9 NA NA 8 NA NA NA NA 15 10 NA 12 11 NA 12 2 9 8 NA NA 9 NA
## [1243] NA 9 9 14 NA 10 9 7 NA 9 10 NA 10 NA NA 9 NA 9 NA 13 10 NA NA
## [1266] 15 NA NA NA NA NA 9 9 12 9 9 NA NA NA 9 NA 12 NA NA 10 NA 2 NA
## [1289] 10 2 NA NA 12 9 10 10 12 6 7 10 7 10 NA 12 NA 12 10 NA 9 NA NA
## [1312] NA NA NA NA 10 12 10 10 NA NA NA NA NA NA 9 6 11 9 9 NA NA NA 10
## [1335] 9 10 NA NA 15 9 NA NA 9 NA NA NA 4 9 NA NA 9 9 NA NA NA NA NA
## [1358] 9 11 9 NA NA NA NA NA NA NA NA 12 10 5 2 6 NA NA 9 11 9 6 11
## [1381] 2 14 NA NA NA NA NA 9 3 9 9 NA 10 NA 9 5 10 NA 10 7 9 9 9
## [1404] 11 9 NA 10 2 9 15 NA 9 4 10 2 10 NA NA 2 10 2 10 9 NA NA 10
## [1427] 2 2 9 NA 1 NA 2 NA NA NA 9 NA NA 2 NA NA NA 9 NA NA NA 2 NA
## [1450] NA NA NA 10 NA 9 10 10 10 9 NA NA 10 12 15 NA NA 9 NA NA NA 2 10
## [1473] NA 2 NA NA NA 9 2 10 2 NA NA NA NA 9 NA 12 NA 10 5 NA 6 11 2
## [1496] NA NA 10 NA 2 10 NA NA NA NA 9 NA 9 NA 10 NA 12 10 10 NA 10 10 NA
## [1519] 9 10 10 NA 11 9 NA NA NA NA 12 9 6 2 9 6 9 NA 9 10 NA NA NA
## [1542] 10 9 NA NA 10 10 1 11 9 10 9 10 15 NA 15 NA 10 10 9 10 12 NA NA
## [1565] 15 9 6 1 2 2 9 9 NA 15 9 NA 9 13 9 6 NA NA NA 9 NA 2 NA
## [1588] NA NA 7 9 NA NA NA 10 10 NA 15 10 15 NA 13 NA 5 11 9 NA 9 2 2
## [1611] NA NA 9 6 NA NA NA 12 2 2 15 NA 9 12 10 6 NA 10 NA NA NA 9 NA
## [1634] NA 10 2 9 9 NA 6 2 12 9 2 NA NA NA 9 NA NA 9 9 6 9 10 2
## [1657] 9 9 9 6 2 NA 13 10 12 2 NA NA NA 13 2 NA NA NA 10 NA 10 10 7
## [1680] 9 10 NA 10 2 NA NA NA 10 NA 9 9 NA NA NA NA 10 NA 10 10 12 10 9
## [1703] 10 2 1 NA 13 NA 10 NA 15 NA 15 9 NA 2 9 12 9 12 NA NA 9 9 9
## [1726] 10 10 10 12 10 12 11 2 NA 12 10 NA NA 9 NA NA NA NA NA 10 NA NA 11
## [1749] NA 2 10 11 NA NA 9 10 NA 10 10 11 10 7 NA NA NA NA NA NA NA NA
## [1772] 9 NA 10 NA NA NA 8 NA NA NA NA NA 12 NA 2 6 9 2 10 6 NA 2 NA
## [1795] NA NA 2 2 10 9 NA 2 NA NA 9 NA NA 10 NA NA 2 NA NA NA NA NA 2
## [1818] 12 NA NA NA NA NA 10 9 NA 2 8 6 10 NA 2 NA NA NA NA NA 10 9 NA
## [1841] 6 NA 10 10 10 10 NA NA 9 10 9 NA NA NA NA 2 NA NA NA 9 NA 10 NA
## [1864] 9 9 NA 9 NA NA 10 NA 15 10 9 2 10 NA NA NA 11 9 12 2 2 9 NA
## [1887] 2 11 9 NA NA 9 NA 13 2 NA 11 2 2 NA 2 NA 11 NA 9 2 12 NA 9
## [1910] 10 2 NA NA NA NA 9 NA
## attr("label")

```

```

## [1] "Party voted for in last national election, France (ballot 1)"
## attr(,"format.stata")
## [1] "%9.0f"
## attr(,"labels")
##
##           Nouveau Centre
##                   1
##           FN (Front National)
##                   2
##           PR (Parti Radical Valoisien)
##                   3
##           NPA (Nouveau Parti Anti-Capitaliste)
##                   4
##           LO (Lutte Ouvriere)
##                   5
##           FDG (Front de Gauche)
##                   6
##           Parti Radical de Gauche
##                   7
##           MPF (Mouvement pour la France)
##                   8
##           PS (Parti Socialiste)
##                   9
##           UMP (Union pour un Mouvement Populaire)
##                   10
##           MODEM (Mouvement Democratique)
##                   11
##           EELV (Europe Ecologie Les Verts)
##                   12
##           Autres mouvements ecologistes
##                   13
##           Autre
##                   14
##           Blanc
##                   15
##           Nul
##                   16
##           Not applicable
##                   NA
##           Refusal
##                   NA
##           Don't know
##                   NA
##           No answer
##                   NA

```

## Operations

R can be used as a calculator, performing mathematical operations:

```
3+4
```

```
## [1] 7
```

```
17/5
```

```
## [1] 3.4
```

```
sqrt(2)
```

```
## [1] 1.414214
```

```
17/(12+4)*7
```

```
## [1] 7.4375
```

```
49/a #here we are calling up scalar *a* that we created earlier
```

```
## [1] 7
```

We can use any type of mathematical operator. The list of the most common mathematical symbols in R is listed below. Do not forget the proper mathematical use of parentheses!

Operator	Meaning
+	addition
-	subtraction
*	multiplication
/	division
^	exponent
sqrt	square root
exp	exponentiation
log	logarithm
abs	absolute value
pi	the constant $\pi$
exp(1)	the constant $e$

## Logical statements

Logical statements in R are evaluated as to whether they are TRUE or FALSE. Here is a summary of the different logical operators in R

Operator	Meaning
<	Less Than
<=	Less Than or Equal To
>	Greater Than
>=	Greater Than or Equal To
==	Equal To
!=	Not Equal To
&	And
	Or

For example, suppose we wanted to create a variable identifying the voters of FN who were happier than average:

```
D$happyFN<-D$prtvctcfr==2 & D$happy>mean(D$happy, na.rm=TRUE) #generating new variable, note the option  
summary(D$happyFN) #summarize new variable
```

```
## Mode FALSE TRUE NA's
```

```
## logical    1461      49      407
```

This produces a vector of TRUE and FALSE for every observation. Interestingly, the summary suggests that most FN voters are not happier than average

Logical statements can also be used to constrain the universe of cases we use when assessing various statistics. Say, for example, that we want to see the age of voters who are happier than average:

```
D$yrbrn[D$happy>mean(D$happy, na.rm=T)] # here note the abbreviation of the option 'na.rm=TRUE' to 'na.
```

```
## [1] 1978 1955 1973 1978 1999 1991 1945 1974 1969 1994 1973 1936 1973 1991
## [15] 1948 1935 1948 1948 1988 1945 1961 1946 1958 1966 1995 1984 1992 1989
## [29] 1955 1978 1935 1981 1991 1980 1959 1930 1990 1928 1939 1984 1992 1974
## [43] 1988 1987 1959 1968 1979 1979 1988 1948 1966 1956 1945 1978 1997 1961
## [57] 1979 1971 1976 1968 1986 1954 1960 1946 1948 1951 1950 1998 1958 1944
## [71] NA 1964 1953 1950 1954 1942 1989 1984 1981 1961 1957 1974 1936 1981
## [85] 1950 1971 1959 1978 1954 1945 1976 1999 1955 1967 1949 1971 1982 1997
## [99] 1942 1992 1954 1960 1995 1959 1979 1948 1972 1953 1989 1998 1993 1978
## [113] 1977 1929 1950 1982 1976 1928 1961 1927 1938 1972 1956 1970 1957 1940
## [127] 1949 1993 1993 1971 1994 1949 1972 1940 1966 1964 1977 1957 1987 1975
## [141] 1956 1981 1959 1986 1971 1981 1995 1986 1946 1949 NA 1951 1965 1966
## [155] 1980 1935 1938 1948 1943 1997 1952 1971 1995 1997 1989 1977 1959 1988
## [169] 1973 1947 1981 1984 1996 1930 1952 1985 1952 1956 1978 1992 1945 1973
## [183] 1954 1992 1982 1979 1978 1970 1981 1973 1968 1956 1949 1962 1939 1962
## [197] 1977 1999 1956 1950 1971 1963 1960 1997 1968 1969 1980 1968 1968 1980
## [211] 1991 1986 1971 1933 1955 1936 1952 1940 1946 1978 1949 1956 1961 1991
## [225] 1984 1940 1983 1944 1998 1945 1952 1987 1972 1972 1958 1970 1977 1949
## [239] 1933 1997 1970 1969 1974 1953 1944 1955 1935 1951 1950 1942 1982 1973
## [253] 1991 1942 1941 1998 1986 1971 1992 1987 1993 1997 1950 1957 1980 1935
## [267] 1983 1977 1936 1943 1959 1985 1980 1977 1969 1947 1946 1967 1998 1974
## [281] 1966 1993 1993 1943 1950 1980 1998 1993 1994 1965 1942 1971 1943 1996
## [295] 1970 1994 1976 1970 1916 1968 1943 1974 1959 1985 1932 1948 1982 1958
## [309] 1977 1994 1968 1947 1973 1938 1944 1957 1962 1956 1954 1958 1972 1975
## [323] 1957 1954 1976 1956 1998 1964 1961 1976 1998 1999 1963 1980 1977 1990
## [337] 1976 1953 1987 1981 1973 1985 1971 1965 1967 1969 1929 1980 1933 1971
## [351] 1965 1957 1974 1946 1999 1954 1979 1998 1957 1975 1981 1966 1977 1991
## [365] 1968 1948 1971 1944 1993 1942 1933 1975 1974 1984 1973 1990 1950 1997
## [379] 1979 1938 1983 1956 1950 1961 1979 1989 1991 1972 1966 1992 1959 1937
## [393] 1947 1953 1948 1981 1974 1972 1937 1986 1973 1992 1979 1983 1960 1987
## [407] 1965 1987 1993 1940 1959 1994 1939 1963 1966 1989 1952 1978 1948 1993
## [421] 1979 1958 1982 1973 1946 1990 1981 1978 1962 1932 1946 1935 1966 1977
## [435] 1936 1935 1964 1958 1944 1967 1992 1973 1998 1956 1962 1964 1991 1996
## [449] 1971 1999 1960 1964 1967 1927 1986 1946 1961 1949 1960 1946 1978 1945
## [463] 1963 1975 1950 1950 1982 1965 1999 1960 1964 1947 1950 1987 1982 1968
## [477] 1978 1958 1961 1966 1988 1946 1949 1968 1989 1981 1933 1959 1982 1937
## [491] 1948 1979 1984 1961 1976 1998 1950 1933 1957 1985 1958 1955 1967 1967
## [505] 1990 1949 1970 1958 1968 1995 1968 1950 1936 1945 1981 1956 1997 1959
## [519] 1977 1934 1983 1996 1972 1943 1931 1973 1995 1992 1986 1992 1929 1958
## [533] 1967 1958 1924 1946 1968 1998 1990 1987 1994 1998 1985 1962 1981 1956
## [547] 1968 1957 1980 1960 1968 1986 1993 1949 1958 1976 1946 1969 1975 1962
## [561] 1953 1971 1965 1953 1999 1998 1940 1974 1966 1984 1973 1973 1950 1938
## [575] 1963 1940 1954 1937 1977 1939 1973 1953 1962 1951 1979 1946 1967 1965
## [589] 1990 1976 1950 1977 1954 1956 1949 1972 1953 1980 1955 1958 1991 1937
## [603] 1970 1978 1967 1986 1988 1954 1989 1982 1988 1980 1975 1961 1991 1948
## [617] 1967 NA 1975 1952 1987 1926 1953 1998 1965 1982 1954 1977 1981 1975
## [631] 1987 1968 1948 1998 1930 1937 1981 NA 1980 1942 1977 1942 1956 1981
```

```
## [645] 1968 1930 1939 1955 1978 1998 1979 1963 1951 1961 1993 1961 1946 1976
## [659] 1962 1989 1959 1968 1944 1977 1990 1960 1970 1981 1975 1933 1970 1982
## [673] 1999 1973 1952 1960 1967 1929 1947 1943 1985 1977 1957 1950 1992 1970
## [687] 1961 1989 1993 1927 1965 1943 1925 1973 1941 1966 1967 1956 1946 1979
## [701] 1970 1938 1951 1942 1997 1989 1980 1967 1980 1959 1932 1975 1942 1937
## [715] 1928 1944 1962 1977 1973 1976 1936 1951 1933 1945 1950 1961 1988 1964
## [729] 1930 1998 1955 1963 1957 1959 1973 1960 1942 1937 1969 1974 1957 1981
## [743] 1977 1934 1959 1956 2000 1953 1992 1989 1961 1991 1991 1987 1975 1991
## [757] 1987 1955 1995 1988 1995 1979 1994 1992 1987 1945 1974 1956 1958 1971
## [771] 1982 1970 1984 1988 1997 1986 1983 1979 1966 1993 1966 1969 1980 1938
## [785] NA 1946 1988 1949 1968 1955 1965 1937 1961 1932 1986 1983 1962 1946
## [799] 1985 1945 1935 1980 1927 1959 1973 1954 1920 1944 1970 1944 1953 1957
## [813] 1971 1996 1997 1978 1991 1980 1954 1973 1945 1963 1962 1952 1949 1934
## [827] 1935 1938 1939 1972 1954 1923 1955 1976 1963 1983 1991 1997 1993 1943
## [841] 1963 1978 1964 1950 1980 1954 1945 1993 1964 1969 1964 1980 1960 1975
## [855] 1951 1970 1954 1959 1993 1948 1962 1960 1924 1942 1961 1950 1986 1978
## [869] 1982 1929 1949 1970 1967 1953 1948 1991 1960 1933 1977 1949 1985 1963
## [883] 1977 1981 1980 1968 1961 1983 1966 1932 1961 1952 1977 1966 1970 1960
## [897] 1985 1950 1970 1960 1961 1971 1992 1985 1946 1983 1989 1977 1973 1970
## [911] 1986 1959 1946 1957 1989 1988 1942 1982 1997 1940 1995 1951 1957 1949
## [925] 1954 1931 1973 1962 1943 1968 1967 1986 1994 1980 1941 1988 1947 1998
## [939] 1945 1979 1963 1980 1983 1938 1947 1995 1947 1951 1973 1964 1965 1967
## [953] 1999 1961 1981 1959 1950 1991 1956 1964 1998 1950 1982 1967 1942 1948
## [967] 1929 1946 1958
```

Here notice the square brackets which contain the specific constraints.

## Recoding

Often when working with data we need to recode variables, that is, to change their values for our particular purposes. There is a variety of ways to do this in R. The basic syntax for creating mathematical transformations of variables follows the form of the examples below.

Imagine, for example, that we have three variables describing a child's reading, writing and calculating ability. We are, however, interested in general academic ability of children and want to create a single summary variable. A very simple way to do this would be to create a variable which adds together the scores on the three variables we have: `ability<-reading+writing+calculating`

In the ESS dataset, the gender variable is coded as 2 for women, and as 1 for men. Suppose we wanted to make this a female dummy variable scored 1 for women and 0 for men. We could do the following

```
D$female <- D$gndr-1 # this subtracts 1 from all values on 'gndr'
```

Another standard type of recoding we might want to do is the creation of a dummy variable that is coded as 1 if the observation meets a certain condition and 0 otherwise. For example, suppose instead of having categories of income, we just want to compare the highest category of income to all the others:

```
D$hi.inc.dummy <- as.numeric(D$hinctnta>9) # codes as 1 everyone in income category 9 or higher
```

Here we use a logical statement and modify the variable with the 'as.numeric' function which turns each TRUE into a 1 and each FALSE into a 0.

Now let's say that we wanted a 3 category ordinal level variable capturing education levels. The 'car' package which contains the 'recode' command is particularly useful here. First we must load the 'car' package, and then proceed to recode our data:

```
library(car)
D$ed.3<-recode(D$eduyrs, "lo:9=1;10:12=2;13:hi=3")
```

Here we have created a new variable from 1 to 3, where individuals with education years 9 or less are coded as 1, between 9 and 12 as 2 and above 12 as 3. Don't forget to use the inverted commas correctly with the recode command. Of course, this syntax could be used to create dummy variables, ordinal variables, or a variety of other recoded variables.

In certain cases we may want to change the nature of the variable from say vector to a factor (a categorical variable). In this case we say:

```
D$prvtvcfr<-as.factor(D$prvtvcfr)
```

This turns the party one voted for into a categorical variable. From now on R will treat this variable as a categorical variable and thus when, for example, running a linear regression model with it as a predictor, R will automatically create dummy variables out of it.

Let's create recode the party, make it a factor, and give it labels.

```
D$party[D$prvtvcfr==4]<-1 # Various left (NPA)
D$party[D$prvtvcfr==5]<-1 # Various left (LO)
D$party[D$prvtvcfr==6]<-1 # Various left (FDG)
D$party[D$prvtvcfr==7]<-1 # Various left (PRG)
D$party[D$prvtvcfr==12]<-2 # Ecologist
D$party[D$prvtvcfr==13]<-2 # Ecologist
D$party[D$prvtvcfr==9]<-3 # PS
D$party[D$prvtvcfr==11]<-4 # Modem
D$party[D$prvtvcfr==10]<-5 # UMP
D$party[D$prvtvcfr==2]<-6 # FN
D$party<-factor(D$party, levels=c(1:6), labels=c("Left", "Ecolo", "Soc", "Center", "Gaulists", "Front"))
```

At this point the variable 'party' has string values (party labels) as opposed to numerical values (numbers). Remember, whenever we work with strings we must put their names into inverted commas "like this"! Now let's say that for our analysis we are not interested in the small parties, and want to work only with the Socialists and Gaulists We need to tell R to omit the two latter categories by using R's logical operators:

```
D$party[D$party=="Left" | D$party=="Ecolo" | D$party=="Center" | D$party=="Front"]<-NA
```

This assigns the missing data symbol 'NA' to these parties and thus does not include them in any further statistical analysis.