Question 1.

a) See accompanying script.

b) Can you provide a brief justification on the type of information you will use to create the variables? For example: I will use numeric information, in particular Integer, to create the Age variable because I want to allow mathematic operations (such as additions, divisions) to be performed on Age.

Solution

|  |  |
| --- | --- |
| **Variable** | **Justification for Representation** |
| Gender | This is categorical in the sense that each of the values it can hold express one of several classes or categories of the subjects. In short it expresses class membership. The values it can hold can be string (character) or numerical, except we should remember the numerical values used here are nominal; they have no significance beyond the purpose of identification of the categories. |
| Age | This variable is numerical, conventionally presented as an integer, so only positive whole numbers are to be used. |
| First Name | The values this variable holds are for identification purposes. The values must therefore be string or character. |
| Last Name | Same justification as First Name. |
| Marital Status | Same justification as Gender. |
| Nationality | Same justification as First Name. |
| Rate of Friendship | Same justification as Gender. |

c) What is an attribute information of a variable? For the variables you have created in 1a, can you give me the length of each vector.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Attribute Information** | **Length of variable** |
| Gender | Nominal | 20 |
| Age | Ratio, because one value can be expressed as the multiple of the other. | 20 |
| First Name | Nominal | 20 |
| Last Name | Nominal | 20 |
| Marital Status | Nominal | 20 |
| Nationality | Nominal | 20 |
| Rate of Friendship | Ordinal, because the values express the relative importance of the various categories they represent | 20 |

Question 2.

I want to create some variables, below, and I have provided some R codes which I believe can do the job for me. Can you identify the error(s) in the codes? Provide solutions to fix them.

a) Age <- c(30, 24.89, ‘50’, 20+10i, True)

b) Sex <- c(‘Male’, Female’)

Solution

1. Age is numerical and conventionally presented as an integer (that is the immediate lower integer of the true age). Thus 24.89, ‘50’, 20+10i and True, which are double, character, complex and logical respectively, are inappropriate candidates for the age variable.

The values 24.89 and ‘50’ are easy to fix by dropping the decimal and the inverted commas respectively (that is 24 and 50), if they are expressing the age of persons.

The values 20+10i and True are not easy to fix as they cannot be said to express age by any stretch of imagination. So we can have age as:

Age <- c(30, 24, 50)

1. The issue with Sex is that the character ‘ is missing on the left side of Female’.

The issue with sex is obviously fixed by placing and inverted comma before Female’, viz:

Sex <- c(‘Male’, ‘Female’)

Question 3.

After answering question 2, please use the variables to create a data frame in R. What do you think might be the problem in creating the data frame? Can you fix that?

Solution

Age and Sex are of different lengths so they cannot be used to form a data frame, unless we are able to determine the sex of the one whose age is 50. Assuming the sex is Male, then sex become:

Sex <- c(‘Male’, ‘Female’, ‘Male’)

In which case the two variables are of the same length (3) and can be combined to form a data frame as follows:

datframe <- data.frame(age=Age,sex=Sex)

Question 4.

## The instruction to use is

## ?data.frame

## which opens the documentation on how to form a data frame. The main heading of the documentation is Data Frames as shown below.

|  |  |
| --- | --- |
| data.frame {base} | R Documentation |

## Data Frames

### Description

The function data.frame() creates data frames, tightly coupled collections of variables which share many of the properties of matrices and of lists, used as the fundamental data structure by most of **R**'s modeling software.

### Usage

data.frame(..., row.names = NULL, check.rows = FALSE,

check.names = TRUE, fix.empty.names = TRUE,

stringsAsFactors = FALSE)

default.stringsAsFactors() # << this is deprecated !