

Foundations of R Software
Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 33
Factors

Hello friend. Welcome to the course Foundations of R Software. Now, in this lecture, we are going to begin with a new topic which is about Factor. So, as usual, what is the factor? That is my first question. Now, if I try to give you a very simple example, you will understand it very easily. You have seen that many times we try to collect the data on the gender of the people like as male, female, male, female etc.

Now, when we are trying to collect the data male and female, then we always try to give it a value. For example, we will write if the person is male, I will write 1 and if the person is female I will write 0. Why this is needed? Because, you see means anyway when you are trying to collect the data after that you want to do some mathematical operations on it. For example, if you want to know that out of 100 students how many are male and how many are female.

So, you need to count them and for counting them inside a software you need to convert them into a numerical value. So, how to convert those characters, because male is a character female is a character, how you can convert those values into a numerical value, that is the job of this operation factor. And, in statistics this is called as categorical variable or binary variable. Binary means when there are only two classes, but if there are more than two classes, in general you can define it as a categorical variable.

For example, if you are trying to take some data on the that how people feels after drinking a coffee, that can be good or say bad or say ok. So, now, there are three categories. So, now, for these three categories, you want to assign them three numerical values. So, how to get it done? That is the job what we are going to do in the lecture today.

And, we will try to see how we can convert or how we can associate a number with the categorical variable. And, that is the job which is called as factor in the R software, means we are trying to do the factorization.

(Refer Slide Time: 02:45)

Categorical variables

Quantitative variables
Example:
Height (in meters) – 1.65, 1.76, ...

Qualitative variables
Example:
Gender – Male, Female
Performance – Excellent, Good, Average, Bad ...

So, let us try to begin this lecture and try to learn how to do this factor in R software, ok. So, if you try to see we have two types of variables. One is here quantitative variables in which we try to get the data on say their numerical value like as height that is measured in meters 1.65 meter, 1.76 meter etc. And, another type of variables are here qualitative variables for example, the tender that is male or female performance can be excellent, good, average, bad etc.

(Refer Slide Time: 03:15)

Categorical variables

Categorical variables
Example:
X : Gender – Male, Female
X = 0 if a person is male
X = 1 if a person is female

Example:

Performance	Excellent	Average	Good	Bad	Labels
X	1	2	3	4	Numeric codes

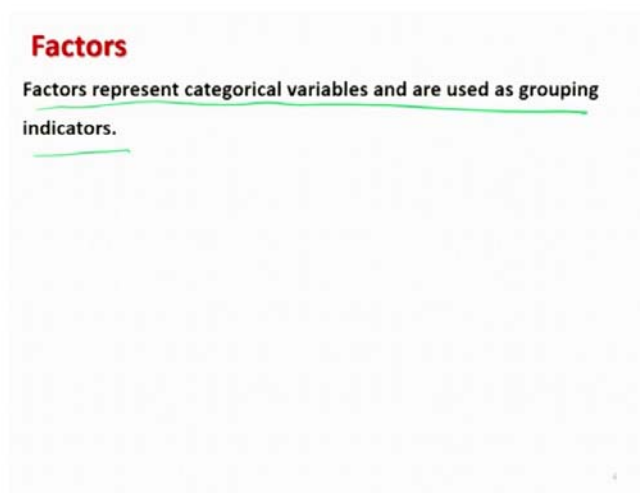
The categories are stored internally as numeric codes, with labels to provide meaningful names for each code

So, now we always want to convert this data into some numerical values. For example, if I am taking the variable here as a gender. There are two possible values male and female. So, I try to take it here of the variable X equal to 0, if the person is male and X equal to 1 if a person is female. And, similarly if there is some performance and that performance is like excellent, average, good or bad, then we try to give it a value here.

If the performance is excellent, we give the number 1, if it is average then we give the number 2, if it is good then we number 3 and if it is bad then we try to give the number 4. So, now these values which we I am writing here as a excellent, average, good, bad or say male or female, they are called as labels. And, whatever the numerical code I am trying to assign them like as here 1 2 3 4 or as say here 0 1 in the case of male or female.

They are the numeric code which are assigned to these labels and these categories are stored internally actually as 1 2 3 4. And, these labels have been chosen in such a way such that they provide some meaningful names for each of the code here right, like this.

(Refer Slide Time: 04:39)




So, now the main important thing what you have to learn is that the factors in the R software they represent the categorical variables and are used as grouping indicators, right. You will see in statistics that many times we are trying to do categorical variable, categorical analysis etc.

(Refer Slide Time: 04:57)

Factors

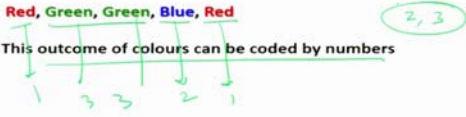
Example:
Suppose we denote the three colours of balls in a basket by following numbers:

Red = 1, Blue = 2, Green = 3



Suppose we draw five balls with following colours:

Red, Green, Green, Blue, Red



This outcome of colours can be coded by numbers

So, in R the same thing is called as factor. So, let me try to give you here one more sample and try to explain you first that what is this factor. Suppose, we have a balls of three colours, say here red, blue and green. Now, we try to give them a number. For the colour red, we give the number 1, for blue we give the number 2 and for green we give the number 3 like this.

And, suppose now we have five balls, they are like this. First ball is red, then the next two balls are green, then the fourth ball is blue and the fifth ball is red. So, now this outcome of five balls can be coded by the numbers also. How? This red can be indicated by 1, green can be indicated by 3, blue can be indicated by 2 and this red can be indicated by 1. So, if somewhere it is written like say here I have got the ball number 2 and 3; that means, I have got the blue and green balls. This is what we mean, right.

(Refer Slide Time: 05:53)

Factors

Each character is mapped to a code.

Factors represent categorical variables and are used as grouping indicators.

The categories are stored internally as numeric codes, with labels to provide meaningful names for each code.

So, each this character is mapped to a code and the factors they represent the categorical variable and are used as grouping indicators. And, these categories are stored internally as numeric code and these numeric codes have been given some labels. And, this label provides some meaningful names for each of the code like as 1 is red, blue is 2 and so on.


(Refer Slide Time: 06:19)

Factors

The order of the labels is important.
First label is mapped to code 1.
Second label is mapped to code 2 and so on.

The values of the codes are always restricted to 1,2,...,k, to represent k discrete categories.

Here "Red" is mapped to code 1,
"Blue" is mapped to code 2 and
"Green" is mapped to code 3.



The order of this label is very important; because for example, the first label is mapped to code 1, second label is mapped to code 2 and so on, right. So, the values in this course are always restricted to some finite number say 1, 2, k; otherwise how will you give it a number. So, this k values are going to indicate the k discrete categories, right. For example, in this case the color red ball is mapped to code 1, blue color ball is mapped to code 2 and green color ball is mapped to code 3. So, this is the name and its code.

(Refer Slide Time: 06:52)

Factors

We have a vector of character strings or integers.

R's term for a categorical variable is a factor.

In R, each possible value of a categorical variable is called a level.

A vector of levels is called a factor.

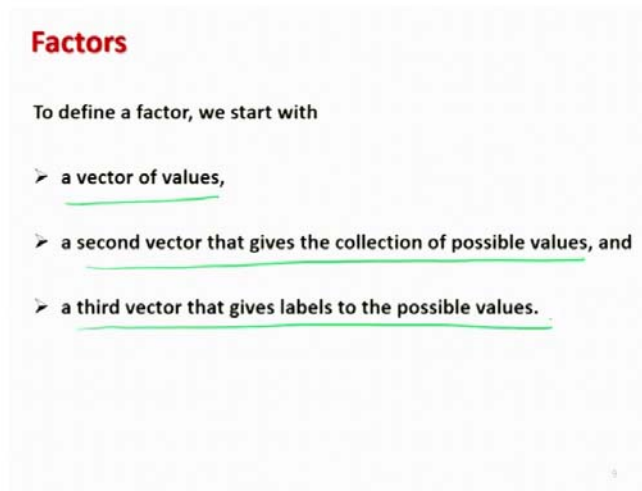
A categorical variable is characterized by a (here: finite) number of levels called as factor levels.

Label level

So, when we are talking about factors then we have a vector of character string or the integer. And, whatever in statistics we call as categorical variable that is called as factor in the R software. And, in the R software, each possible value of a categorical variable is called as level. Now, you have to be very careful with my audio because now there are two words label and here is level, right.

So, please try to be careful whether I am saying label or level. So, label will be like this and level will be like this. So, a vector of levels is called a factor. And, a categorical variable is characterized by a number of levels which are called as factor levels, right. And, here in this case the number of factor levels are going to be finite, ok.

(Refer Slide Time: 07:51)



So, now the question is how to define the factors in the R software? So, to define it, we start with a vector of values. And, then we try to define another vector, the second vector that gives the collection of possible values. And, then there is a third vector that gives the labels to the possible values, right.

(Refer Slide Time: 08:13)

Factors

The `factor` function encodes the vector of discrete values into a factor:

```
factor(x)
```

where `x` is a vector of strings or integers.

If the vector contains only a subset of possible values and not the entire values, then include a second argument that gives the possible levels of the factor:

```
factor(x, levels)
```

And, in order to do it, we have a command here `factor` `f a c t o r`. So, this is a function that encodes the vector of discrete values into a factor, right. So, if you have some data vector or a vector of strings or say integers, if you try to write down here `factor x`, then it will be done. For the command is like here `f a c t o r` and inside the parenthesis you write here `x`.

And, then in case if you have a vector that does not contain all the values, but it contains only a subset of the possible values, then we try to include one more argument that gives the value of the possible levels of the factor, like this `factor x`. And, then inside the parenthesis say `x` comma and then you write down here labels. At this moment, you simply try to understand what I am trying to say and as soon as I take an example, means each of this thing will become very easy that is my promise to you right.

(Refer Slide Time: 09:14)

Factors

Usage

```
factor(x = character(), levels, labels =  
levels, exclude = NA, ...)
```

levels : Determines the categories of the factor variable.
Default is the sorted list of all the distinct values of `x`.

labels : (Optional) Vector of values that will be the labels of the categories in the `levels` argument.

exclude : (Optional) It defines which levels will be classified as `NA` in any output using the factor variable.

So, the usual command for the factors in the R software is here like this, that we write here factor and then here x is the vector of integer or a strings and then we have here levels. So, this levels are going to determine the categories of the factor variable and usually they are coated out with all the individual distinct values in the x. This is an optional vector, this is a labels, right.

Labels of the categories in the labels here, right. The names of labels are stored in the labels and after that you have here exclude. So, exclude is going to handle the missing values. So, it defines which levels will be classified as NA, if the output is a factor; output is obtained using the factor variable.

(Refer Slide Time: 09:58)

```
factor (base) R Documentation
  

Factors
  

Description
  

The function factor is used to encode a vector as a factor (the terms 'category' and 'enumerated type' are also used for factors). If
argument ordered is TRUE, the factor levels are assumed to be ordered. For compatibility with S there is also a function ordered.
  

is.factor, is.ordered, as.factor and as.ordered are the membership and coercion functions for these classes.
  

Usage
  

factor(x = character(), levels, labels = levels,
       exclude = NA, ordered = is.ordered(), NAas = NA)
  

ordered(x, ...)
  

is.factor(x)
is.ordered(x)
```

And, if you want to have more information about this factor function, I will request you that you please try to look into the help menu right. And, then you can see here there are factor variable and then you can see here is dot factor etc., right.

(Refer Slide Time: 10:16)

Factors
Example:

Suppose we roll a die seven times and observe the outcome in the vector y .

```
> y = c(1, 4, 3, 5, 4, 2, 4)
```

Possible values of upper face of die are 1 to 6 and we store them in a vector `possible.dieface`

```
> possible.dieface = c(1, 2, 3, 4, 5, 6)
```

But, anyway now let me try to give you here one example and which will make the things very clear. You all know what is this, this is a die and this dice has six faces, in which there are these types of dots here 1 dot, then 3 dot and so on. So, there are six possible values 1, 2, 3, 4, 5, 6 right when you roll a die. So, suppose die rolls seven times and we observe the point 1, then 4, then 3, then 5, then 4, then 2 and then 4. And, all these values are stored in a data vector y .

Now, as you know that there are 6 possible values in this die. So, I try to store all these 6 values here in a data vector `c(1, 2, 3, 4, 5, 6)` and I try to give it here a name `possible.dieface`. So, I am just trying to give it a name which conveys the meaning also. So, these are the possible values of the points on the face of the die, right.

(Refer Slide Time: 11:21)

Factors
Example:

We wish to label the rolls by the words "one", "two", ..., "six".

We put these labels in the vector `labels.diefaces`:

```
> labels.dieface = c("one", "two", "three",  
"four", "five", "six")
```

Construct the factor variable `facy` using the function `factor`:

```
> facy = factor(y, levels = possible.dieface,  
labels = labels.dieface)
```

Now, what we want here? We want here that now we have got this outcome and this outcome is coming out of these possible values and I would like to write this 1 as one, 2 as two, 3 as three and so on. So, that when I am getting here an output like 4, I should be able to see here as f o u r four. So, how to get it done? Let us try to understand. So, first I try to define here the labels of this die face.

So, for these values which are here 1, 2, 3, 4, 5, 6, I try to define here a corresponding data vector which has got these values which are string like as one, two, three, four, five and six. And, I give it a name here labels dot dieface, right and now I try to take this data, this here y. And, I try to use here my command factor and I tells the levels here are the possible dieface which is here vector c 1, 2, 3, 4, 5 and 6.

And, then the labels on this possible dieface is obtained by this command, labels which is here. This data vector labels dot dieface which is here like this.

(Refer Slide Time: 12:47)

Factors
Example:
Observe the difference between a character vector and a factor.

```
> facy  
[1] one four three five four two four  
Levels: one two three four five six
```

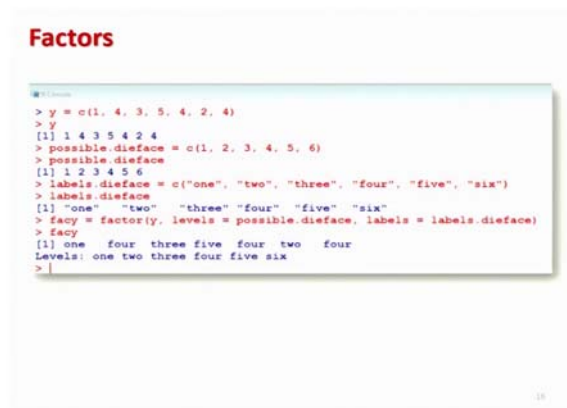
Note that

```
y = c(1, 4, 3, 5, 4, 2, 4)
```

So, now if you try to operate it, what do you see? And, whatever is the outcome trying to store it in the variable name facy; that means, the factor of y. So, now, if you try to see what is the outcome of this facy? Let us try to give you here the value like o n e one, f o u r four, t h r double e three, f i v e five, f o u r four, t w o two and f o u r four. What are these things? Can you recall you here this outcome data vector which was here 1, 4, 3, 5, 4, 2, 4.

And, now you can see what this factor command has done, this has converted this number 1 into o n e one, number 4 into f o u r four, number 3 into this t h r e e and similarly here 5, 4, 2, 4, this is here like in say character five four two four. And, these are here the levels. And, this levels here are indicated by here one two three four five six; that means, these are the level which have been used. And, the data in the y vector has been converted using these labels, right.

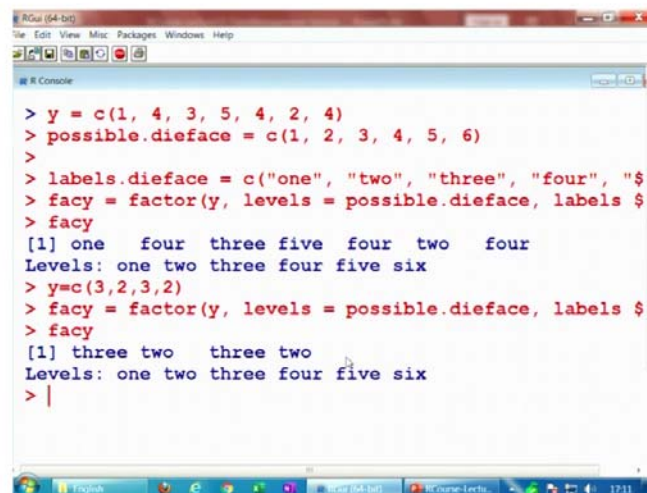
(Refer Slide Time: 13:55)



```
Factors
> y = c(1, 4, 3, 5, 4, 2, 4)
> y
[1] 1 4 3 5 4 2 4
> possible.dieface = c(1, 2, 3, 4, 5, 6)
> possible.dieface
[1] 1 2 3 4 5 6
> labels.dieface = c("one", "two", "three", "four", "five", "six")
> labels.dieface
[1] "one" "two" "three" "four" "five" "six"
> facy = factor(y, levels = possible.dieface, labels = labels.dieface)
> facy
[1] one four three five four two four
Levels: one two three four five six
>
```

So, that is how the things work and if I try to show you this operation on the R console, possibly this will make the things very clear here.

(Refer Slide Time: 14:05)



```
RGui (64-bit)
File Edit View Misc Packages Windows Help
R Console
> y = c(1, 4, 3, 5, 4, 2, 4)
> possible.dieface = c(1, 2, 3, 4, 5, 6)
>
> labels.dieface = c("one", "two", "three", "four", "$
> facy = factor(y, levels = possible.dieface, labels $
> facy
[1] one four three five four two four
Levels: one two three four five six
> y=c(3,2,3,2)
> facy = factor(y, levels = possible.dieface, labels $
> facy
[1] three two three two
Levels: one two three four five six
>
```

So, this is now your here y and this is your here the data vector possible dieface and then I try to define here this labels. And, then I try to define here this here operation, this factor right and I try to store it in the variable facy. So, you can now if you try to see here what is your here facy? Like this and if you try to yeah change your data vector, suppose if I try to take here the data vector like as here say 3, 2 say 3, 2 and so on.

Then, if you try to once again do this operation factor, then you can see here what is the output here? This facy is 3, 2, 3, 2 and now it is three two three two, but in words. So, this is how now you can see that these operations are working and now we come to an end to this lecture. You can see that it was a very simple topic. You are simply trying to convert some characters into some numerical values and those numerical values are the levels which are provided by you. Well, these things are very common in real life when we are trying to conduct some experiment and trying to collect the data.

Because, whenever you are trying to analyze the data, you always need to associate some numerical values; unless and until you associate the numerical values, you cannot do the computation, you cannot do the calculations. For example, the way you try to work that how many students are male, how many are female, what you try to do? Whatever is the data say male, female, you try to give them a value like 1, 0 and you simply try to sum them 1 plus 0 plus 1 plus 0.

And, that will give you the number of people in the category 1 and the remaining will be in the category 0. And, now you have to see 1 correspond to the male and 0 corresponds to the female. So, that will give you the total number of male and female students in the class or in the school. So, now, we stop here and it is your turn to take some example, try to practice it, and I will see you in the next lecture with some more new topic.

Till then goodbye.