

Godavari Foundation's
Godavari College of Engineering, Jalgaon
Department of Computer
Continuous Assessment I/II
Introduction to Data Science with R

Date:- _____

Name of Student:- _____

Class:- _____

PRN No:- _____

Title: -

Aim: -

Software Requirement: _____

Hardware Requirement:- _____

Theory:-

Factors are variables in R which take on a limited number of different values. Such variables are often referred to as categorical variables. One of the most important uses of factors is in statistical modeling. Since categorical variables enter into statistical models differently than continuous variables, storing data as factors insures that the modeling functions will treat such data correctly.

Factors in R are stored as a vector of integer values with a corresponding set of character values to use when the factor is displayed. The factor function is used to create a factor. The only required argument to factor is a vector of values which will be returned as a vector of factor values. Both numeric and character variables can be made into factors, but a factor's levels will

always be character values. The possible levels for a factor can be seen through the levels command.

To change the order in which the levels will be displayed from their default sorted order, the levels= argument can be given a vector of all the possible values of the variable in the order desired. If the ordering should also be used when performing comparisons, the optional ordered=TRUE argument can be used. In this case, the factor is known as an ordered factor.

The levels of a factor are used when displaying the factor's values. These Levels can be changed at the time a factor is created by passing a vector with the new values through the labels= argument. This actually changes the internal levels of the factor, and to change the labels of a factor after it has been created, the assignment form of the levels function is used. Consider a factor taking on integer values which is used to display as roman numerals.

```
> data = c(1,2,2,3,1,2,3,3,1,2,3,3,1)
> fdata = factor(data)
> fdata
```

```
[1] 1 2 2 3 1 2 3 3 1 2 3 3 1
Levels: 1 2 3
```

```
> rdata = factor(data,labels=c("I","II","III"))
> rdata
```

```
[1] I II II III I II III III I II III III I
Levels: I II III
```

To convert the default factor fdata to roman numerals, use the assignment form of the levels function:

```
> levels(fdata) = c('I','II','III')
> fdata
```

```
[1] I II II III I II III III I II III III I
Levels: I II III
```

Factors represent a very efficient way to store character values, because each unique character value is stored only once, and the data itself is stored as a vector of integers.

R has powerful indexing features for accessing object elements. These features can be used to select and exclude variables and observations.

Selecting Variables

Select all rows that have a value of age greater than or equal to 20 or age less than 10. Keep the ID and Weight columns.

```
ID<-c(1,2,3,4,5)
```

```
age<- c(20, 50, 60, 10,100)
```

```
weight<-c(40,50,60,55,45)
mydata<-matrix(c(ID,age,weight),5,3)
mydata
colnames(mydata) <- c("ID","age","weight")
mydata

newdata <- subset(mydata, age >= 20 | age < 10, select=c(ID, weight))
newdata
```

Comapring Factors

Convert the fertilizer levels to an ordered factor:

```
> fert = c(10,20,20,50,10,20,10,50,20)
> fert = factor(fert,levels=c(10,20,50),ordered=TRUE)
> fert
[1] 10 20 20 50 10 20 10 50 20
Levels: 10 < 20 < 50
```

Source Code:-

```
## Consider a factor taking on integer values and display as roman numerals
data = c(1,2,2,3,1,2,3,3,1,2,3,3,1)
fdata = factor(data)

## Print fdata
fdata

## Convert to roman numerals
rdata = factor(data,labels=c("I","II","III"))

## Print rdata
rdata

levels(fdata) = c('I','II','III')

## Print fdata
fdata

## Comapring Factors
fert = c(10,20,20,50,10,20,10,50,20)
fert = factor(fert,levels=c(10,20,50),ordered=TRUE)

## Show fert
Fert
```

Output:-

```
> ## Consider a factor taking on integer values and display as roman numerals
> data = c(1,2,2,3,1,2,3,3,1,2,3,3,1)
> fdata = factor(data)
>
> ## Print fdata
> fdata
```

```
[1] 1 2 2 3 1 2 3 3 1 2 3 3 1
Levels: 1 2 3
```

```
>
> ## Convert to roman numerals
> rdata = factor(data,labels=c("I","II","III"))
>
> ## Print rdata
> rdata
```

```
[1] I  II II III I  II III III I  II III III I
Levels: I II III
```

```
> levels(fdata) = c('I','II','III')
>
> ## Print fdata
> fdata
```

```
[1] I  II II III I  II III III I  II III III I
Levels: I II III
```

```
> ## Comparing Factors
> fert = c(10,20,20,50,10,20,10,50,20)
> fert = factor(fert,levels=c(10,20,50),ordered=TRUE)
>
> ## Show fert
> fert
```

```
[1] 10 20 20 50 10 20 10 50 20
Levels: 10 < 20 < 50
```

Conclusion:-
