Functions

Juan Camilo Rivera. [1] Hugo Andres Dorado. [1]

[1]Big data and site-specific agriculture
Decision and Policy Analysis
Centro Internacional de Agricultura Tropical

Data analysis course, 2018

Outline

1. Basic Functions

2. Date and Times

3. Vectorised Calculations

Basic Functions

- **sort()** the elements of a vector, in increasing or decreasing order.

```
sort(c(1,3,6,2,7,4,8,1,0))
sort(c(1,3,6,2,7,4,8,1,0),decreasing=TRUE)
```

- **rev()** rearranges the elements of a vector in reverse order.

```
rev(c(1,3,6,2,7,4,8,1,0))
```

- **order()** returns the vector of (increasing or decreasing) ranking indices of the elements

```
vec <- c(1, 3, 6, 2, 7, 4, 8, 1, 0)
names(vec) <- 1:9
order(vec)
```

- **rank()** returns the vector of ranks of the elements. In case of a tie, the ordering is always from left to right.

```
rank(vec)
```

Basic Functions

- **unique()** removes the duplicates of a vector.

```
unique(c(1,3,6,2,7,4,8,1,0))
```

- **duplicate()** indicates elements which have already been encountered earlier in the vector (read from left to right).

```
duplicated(c(1,3,6,2,7,4,8,1,0))
```

Basic Functions

table() It can be quite useful to tabulate factors or find the frequency of an object.

```
> attach(quine)
> table(Age)
> table(Sex, Age)
```

split() divides the data specified by vector x into the groups defined by factor.

```
> split(Days,Sex)
> boxplot(split(Days,Sex),ylab="Days Absent")
> library(lattice) # trellis graphics
> trellis.par.set(col.whitebg())
> bwplot(Days  Age | Sex) # implicit split
```

Basic Functions

with() evaluates expressions constructed from the data

```
> with(Cars93,plot(Weight,100/MPG.highway))
```

subset() returns subsets of vectors or data frames that meet specific requirements

```
> Vans <- subset(Cars93,Type=="Van")
```

transform() transforms elements of an object

```
> Cars93T <- transform(Cars93,WeightT=Weight/1000)
```

Vectorised Calculations

lapply() takes any structure, gives a list of results

```
l <- list(Sex=Sex,Eth=Eth)
lapply(l,table)
```

sapply() like lapply, but simplifies the result if possible

```
 l <- list(Sex=Sex,Eth=Eth)
 sapply(l,table)
```

apply() only used for arrays

```
> apply(iris[,-5],2,mean)
```

tapply() used for ragged arrays: vectors with an indexing specified by one or more factors.

```
> quine[1:5,]
> tapply(Days,Age,mean)
> tapply(Days,list(Sex,Age),mean)
```

Date and Times

- POSIXct are numeric vectors with each component representing the number of seconds since the start of 1970.
- POSIXlt are lists with the separate parts of the date/time held as separate components.

```
> myBday <- strptime("18-Apr-1973", "%d-%b-%Y")
> class(myBday)
> myBday
> weekdays(myBday)
> Sys.time()
> Sys.time() - myBday
```

Date and Times

```
> as.numeric(Sys.time())
> as.numeric(myBday)
> as.numeric(as.POSIXct(myBday))
> as.numeric(Sys.time()) - as.numeric(as.POSIXct(myBday))
```

For Further Reading I

- Pierre Lafaye de Micheaux.
  The R Software,
  Fundamentals of programming and statistical analysis
  Springer, 2013.

- William Sullivan
  Machine Learning for Beginners Guide Algorithms

- Giuseppe Ciaburro
  Balaji Venkateswaran
  Neural Networks with R