technische universität dortmund

Dept. of Computer Science
Chair of Algorithm Engineering

Tutorial for

Introduction to Computational Intelligence in Winter 2015/16

Günter Rudolph, Vanessa Volz
https://ls11-www.cs.uni-dortmund.de/people/rudolph/teaching/lectures/CI/WS2015-16/lecture.jsp

Sheet 0, Block 0

22 Oct 2015

Due date: none
Discussion: today

Exercise 0.1: Programming

Create an R script (ci1415_0-1.R) that contains a function called customSum that returns the sum of two input variables. Test your function with various data types (numbers, strings, vectors).

Now, add another function (fibonacci(n)) to your script that returns the smallest fibonacci number that is greater than or equal to the input variable n. Test your function with various data types (numbers, strings, vectors).

Solution

```r
#computes the sum of x and y
customSum <- function (x,y)
return(x+y) #return the sum

#returns the smallest fibonacci number larger than or equal to n
fibonacci <- function(n){
  #initialisation
   f0<-0
   f1<-1
   x<-f0
  #compute next fibonacci number until it is no longer smaller than n
   while(x < n){
     x<-f0+f1
     f0<-f1
     f1<-x
   }
#return the number found
return(x)
}
```

Listing 1: `ci1516_0-1.R`

Exercise 0.2: Matrix operations

Create an R script (ci1415_0-2.R), that saves the multiplication table from 1 to 10 in a matrix. Implement two different methods to compute the table, one using apply and/or similar functions and one without these functions. Comment your script.

What is 4 times 5 according to your table? Print one row of the multiplaction table.

Solution

```r
#create multiplication table
A= matrix(rep((1:10),10),10,10)
MultTable=A*t(A)
#4*5
```

```r
print(MultTable[4,5])
#print row 3
print(MultTable[3,])

#computes one row of the multiplication matrix
m = function(a){return(a*c(1:10))}
#compute matrix by applying m to an array of 1 to 10
#print(lapply((1:10),m)) #unformatted
print(sapply((1:10),m))
```

Listing 2: `ci1516_0-2.R`

Exercise 0.3: Statistics
Create an R script (ci1415_0-3.R) that calculates mean, median, minimal and maximal values of a random sequence. Test it with different probability distributions (uniform, normal, etc.).

Now download and import the famous iris data set `https://archive.ics.uci.edu/ml/datasets/Iris` into R. The data originally collected by R.A. Fisher records the sepal length, sepal width, petal length, petal width and class of 150 iris flowers. Add appropriate column names to the data set and use them to calculate the standard deviation of a row. Look at the summary of your data set.

Solution

```r
#draw sequence from normal distribution
seq = rnorm(50, mean=2, sd=3)
#print different statistics
print(summary((seq)))
print(mean(seq))
print(median(seq))
print(min(seq))
print(max(seq))

#external link to data set, alternatively download and use local file
file = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
#import data into R, using the appropriate separator and specifying column names
iris = read.table(file, header=FALSE, sep=",", col.names = c("sepalLength","sepalWidth"
    ,"petalLength", "petalWidth","class"))
#print statistical values of the iris data set
print(summary(iris))
print(sd(iris$sepalWidth))
```

Listing 3: `ci1516_0-3.R`

Exercise 0.4: Graphics
Create an R script (ci1516_0-4.R) to recreate the plot in figure 1 as closely as possible. The figure is a plot of the functions $f(x) = \sin(x)$ and $f(x) = \cos(x)$ over the interval $[0, 2\pi]$. The script should export the plot as a pdf file. Comment the script.

Solution

```r
#start graphics device driver
pdf("ci1516_0-4Plot.pdf")
#draw curve for cosine in blue
curve(cos(x), axes=FALSE, xlab="x-values", ylab="f(x)", xlim=c(0,2*pi), col="red")
#draw x-axis
axis(side = 1, at = c(0, pi/4, pi/2, 3*pi/4, pi, 5/4*pi, 3/2*pi, 7/4*pi, 2*pi), labels
    = expression(0, pi/4, pi/2, 3*pi/4, pi, 5/4*pi, 3/2*pi, 7/4*pi, 2*pi), pos=0)
#draw y-axis
axis(side = 2, las=1, at = c(-1, -0.5, 0, 0.25, 0.5, 0.75, 1), pos=0)
#draw horizontal lines through plot
abline(h=c(-1,-0.75,-0.5,-0.25,0.25,0.5,0.75,1))
#draw curve for sine in red, add it to the plot
curve(sin(x), from=0, to=2*pi, add=TRUE, col="blue")
#set title
```

```
14 title("Sine and Cosine")
15 #stop active graphics device driver
16 dev.off()
```

Listing 4: `ci1516_0-4.R`

Exercise 0.5: Libraries

Have a look at the available third party packages at `http://cran.r-project.org/web/packages/available_packages_by_name.html` and read `help(install.packages)` and `help(library)`.

If you want to test installing packages, why not try the RXKCD package that allows you to access XKCD comics directly from R (see `https://cran.r-project.org/web/packages/RXKCD/`).

Solution

```
1 install.packages("RXKCD")
2 library("RXKCD")
3 searchXKCD("data")
4 getXKCD(373)
```

Listing 5: `ci1516_0-5.R`

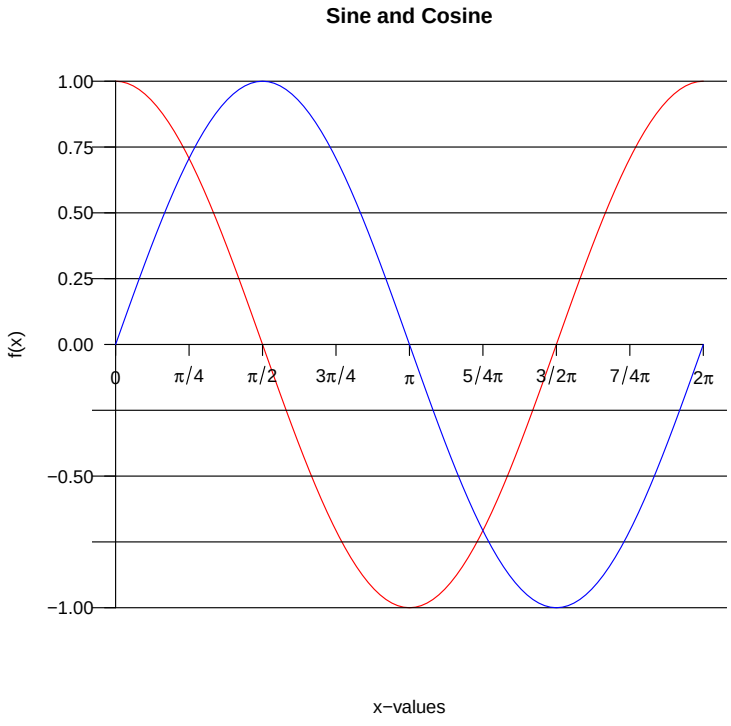

Figure 1: Plot this figure!