Complexity modeling Sayama

Matti Heino

29 joulukuuta 2017

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

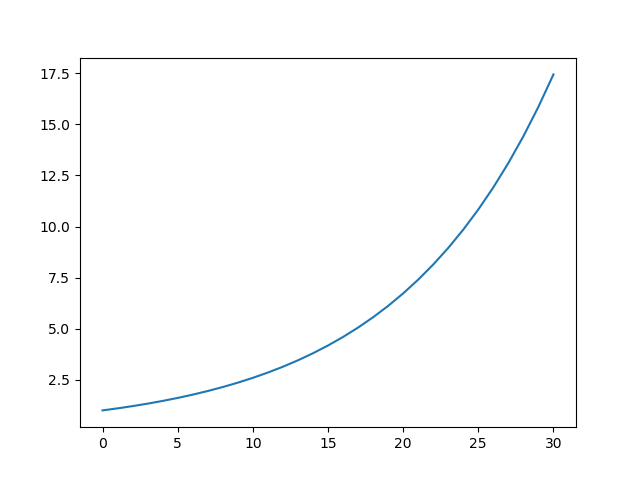
## Python code

Example from @HirokiSayama’s Introduction to the Modeling and Analysis of Complex Systems

# To get this far on a uni computer with limited rights to modify directory content, I had to...  
# 0. Make sure python is in the environmental variables path; type "edit the system environment variables" in Start menu.  
# 1. Open Command Prompt (press Windows button and r, then type in "cmd", or type it in start menu)  
# 2. Type: pip install -t "C:/a-path-you-can-edit/pythonLibs/" matplotlib  
# 3. Then run the next four lines with code before running what you want to run.  
import sys  
sys.path.append("C:/LocalData/hema/pythonLibs/") # this is the library you chose  
print(sys.path)

## ['C:\\Python27', 'C:\\WINDOWS\\SYSTEM32\\python27.zip', 'C:\\Python27\\DLLs', 'C:\\Python27\\lib', 'C:\\Python27\\lib\\plat-win', 'C:\\Python27\\lib\\lib-tk', 'C:\\PROGRA~1\\R\\R-34~1.3\\bin\\x64', 'C:\\Python27', 'C:\\Python27\\lib\\site-packages', 'C:/rlibs/3.4.2/reticulate/python', 'C:/LocalData/hema/pythonLibs/']

from pylab import \*  
a = 1.1  
def initialize():  
 global x, result  
 x = 1.  
 result = [x]  
def observe():  
 global x, result  
 result.append(x)  
def update():  
 global x, result  
 x = a \* x  
initialize()  
for t in xrange(30):  
 update()  
 observe()  
# matplotlib.pyplot.style.use("ggplot")  
plot(result)  
# matplotlib.pyplot.show()  
matplotlib.pyplot.savefig('figures/firstfig')



x1 <- 0.2  
const <- 2.95  
x <- vector()  
x[1] <- x1  
  
for (i in 1:99) {  
 x[i + 1] <- const\*x[i]\*(1-x[i])  
}  
  
dyn <- data.frame(rep = seq(1:(i+1)), x)  
  
ggplot(dyn, aes(x = rep, y = x)) +  
 geom\_point()  
  
  
A <- 1  
B <- 1  
A[1] <- A  
B[1] <- B  
  
for (i in 1:99) {  
 A[i + 1] <- ifelse(B[i] == 1, 1, 0)  
 B[i + 1] <- ifelse(A[i] == 0, 1, 0)  
}  
  
ABdata <- data.frame(step = seq(1:100), A = A, B = B)  
  
ggplot(ABdata, aes(x = step, y = A)) +  
 geom\_point(fill)  
  
ABplot <- ggplot(ABdata, aes(x = step, y = A, group = 1)) +  
 geom\_line(colour = "darkred", size = 1) +   
 geom\_line(data = ABdata, aes(x = step, y = B), colour = "darkblue", size = 1) +  
 theme\_classic() +   
 scale\_y\_discrete(name = "Color") +  
 coord\_cartesian(xlim = c(0, 100), ylim = c(0, 2))