Beroggi G.E.G., 2014. "Analytical Computations in R from LiveCode." *Spring Analytica*. 14/1, 1-4.

Spring Analytica
Data and Decision Sciences

Publication Series
www.springanalytica.com/Publ.html

Analytical Computations in R from LiveCode

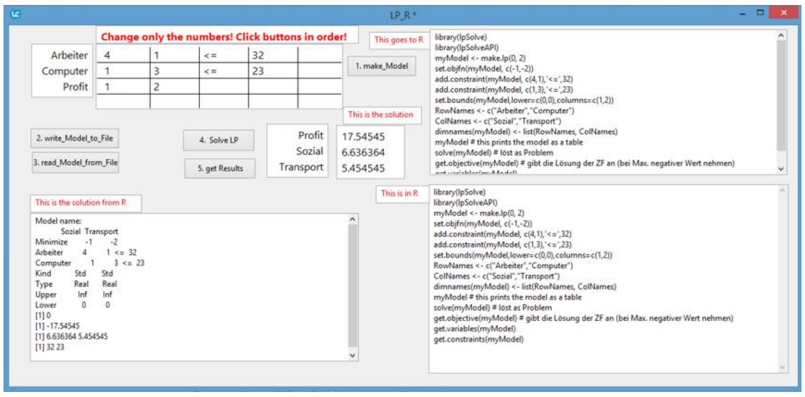
Giampiero E.G. Beroggi
comments to: beroggi@gmail.com
October 23, 2014

LiveCode is a spinoff of HyperCard, Apple's which was up to the mid 90-ties the most versatile multimedia authoring tool. The strength of HyperCard was that it combined versatile development environment for designing the interface and a powerful programming language, called HyperCard, with a very steep learning curve. Various versions of HyperCard emerged on the market, with LiveCode being the most successful offspring, which allows to develop apps on mobile systems, on the web, as well as on desktop systems. Although LiveCode has many mathematical functions, it lacks the powerful mathematical capabilities of specialized software systems, such as R, an open source tool, which is very widespread in academia. In this paper, the use of R from within the LiveCode environment is demonstrated for a mathematical optimization problem. Data and the model are entered or modified in LiveCode, which then starts R in batch-mode. The results from the computations in R are written to a file, which is, in turn opened and read by LiveCode for presentation of the results in LiveCode. Linking LiveCode an R in such a way allows developers to benefit simultaneously from the computational power of R and the flexibility of LiveCode.

LiveCode is a very easy to use App-Development environment, which is based on Apple's Hypercard. R is an open source statistical computing software. In this article we show how to launch R withing LiveCode, and, by doing so, using all the power of R from within the LiveCode environment. We will achieve this by writing the appropriate R-code into a txt-file and lunching R in batch-mode.

Let's assume we have the following decision problem: A City Government wants to decide, how many millions to allocate to the social and to the transportation program. For each million allocated to the social program, 4 workers must be employed, and for each million allocated to the transportation program, 1 worker must be employed. The maximum worker to be employed is 32. For each million allocated to the social program, 1 new computer must be purchased, and for each million allocated to the transportation program, 2 new computers must be purchased. In total, no more than 23 computers must be purchased. For each million allocated to the social program, 1 million annual profit is expected, while for each million allocated to the transportation program, a profit of 2 million is expected. The goal is to maximize profit. With the decision variables being the millions allocated to the social program (s) and the millions allocated to the transportation program (t), we get the following optimization problem:

max: s + 2t (profit)
s.t.: 4s + t <= 32 (worker)
      s + 2t <= 23 (computers)



Arbeiter	4	1	<=	32	
Computer	1	3	<=	23	
Profit	1	2			

Profit	17.54545
Sozial	6.636364
Transport	5.454545

```
library(lpSolve)
library(lpSolveAPI)
myModel <- make.lp(0, 2)
set.objfn(myModel, c(-1,-2))
add.constraint(myModel, c(4,1),'<=',32)
add.constraint(myModel, c(1,3),'<=',23)
set.bounds(myModel,lower=c(0,0),columns=c(1,2))
RowNames <- c("Arbeiter","Computer")
ColNames <- c("Sozial","Transport")
dimnames(myModel) <- list(RowNames, ColNames)
myModel # this prints the model as a table
solve(myModel) # löst as Problem
get.objective(myModel) # gibt die Lösung der ZF an (bei Max. negativer Wert nehmen)
```

```
Model name:
          Sozial  Transport
Minimize      -1        -2
Arbeiter       4         1  <=  32
Computer       1         3  <=  23
Kind         Std       Std
Type        Real      Real
Upper        Inf       Inf
Lower          0         0
[1] 0
[1] -17.54545
[1] 6.636364 5.454545
[1] 32 23
```

```
library(lpSolve)
library(lpSolveAPI)
myModel <- make.lp(0, 2)
set.objfn(myModel, c(-1,-2))
add.constraint(myModel, c(4,1),'<=',32)
add.constraint(myModel, c(1,3),'<=',23)
set.bounds(myModel,lower=c(0,0),columns=c(1,2))
RowNames <- c("Arbeiter","Computer")
ColNames <- c("Sozial","Transport")
dimnames(myModel) <- list(RowNames, ColNames)
myModel # this prints the model as a table
solve(myModel) # löst as Problem
get.objective(myModel) # gibt die Lösung der ZF an (bei Max. negativer Wert nehmen)
get.variables(myModel)
get.constraints(myModel)
```

Figure 1: LiveCode interface for using R in the LiveCode environment.

The LiveCode interface is shown in Figure 1. It contains a field "Werte" (4x3 matrix) with the coefficients of the optimization problem:

```
4 1 <= 32 (worker)
1 3 <= 23 (computer)
1 2       (profit)
```

We then define another field in LiveCode, called "putModel". This is the model that gets run by R:

```
library(lpSolve)
library(lpSolveAPI)
myModel <- make.lp(0, 2)
set.objfn(myModel, c(-1,-2))
add.constraint(myModel, c(4,1),'<=',32)
add.constraint(myModel, c(1,3),'<=',23)
set.bounds(myModel,lower=c(0,0),columns=c(1,2))
RowNames <- c("Arbeiter","Computer")
ColNames <- c("Sozial","Transport")
dimnames(myModel) <- list(RowNames, ColNames)
myModel # this prints the model as a table
solve(myModel) # löst as Problem
get.objective(myModel) # gibt die Lösung der ZF an (bei Max. negativer Wert nehmen)
get.variables(myModel)
get.constraints(myModel)
```

In LiveCode we run then the following scripts:

1. make_Model

```
on mouseUp
  put field "getModell" into modell
  set  itemDelimiter to Tab
  put field "Werte" into werte
  put item 1 of line 1 of werte into x11
  put item 2 of line 1 of werte into x12
  put item 1 of line 2 of werte into x21
  put item 2 of line 2 of werte into x22
  put item 1 of line 3 of werte into g1
  put item 2 of line 3 of werte into g2
  put item 4 of line 1 of werte into c1
  put item 4 of line 2 of werte into c2

  put "set.objfn(myModel, c(-"&g1&",-"&g2&"))"  into line 4 of modell
  put "add.constraint(myModel, c("&x11&","&x12&"),'<=',"&c1&")" into line 5 of modell
  put "add.constraint(myModel, c("&x21&","&x22&"),'<=',"&c2&")" into line 6 of modell
  put modell into field "putModell"
end mouseUp
```

2. write_Model_to_File

```
on mouseUp
  put field "putModell" into it
  open file "C:\R\myLP.R " for write -- "for write" means that it will replace the content
  write it to file "C:\R\myLP.R "
  close file "C:\R\myLP.R "
end mouseUp
```

3. read_Model_from_File

```
on mouseUp
  open file "C:\R\myLP.R "
  read from file "C:\R\myLP.R " at 1 until EOF
  close file "C:\R\myLP.R "
  put it into field "getModell"
```

```
end mouseUp
```

4. Solve_LP

```
on mouseUp
  put "C:\R\Start_R_LP.bat" into tBatchCommand
  get shell(tBatchCommand)
  answer "OK"
end mouseUp
```

5. get Reults

```
on mouseUp
  open file "C:\R\myLP.Rout"
  read from file "C:\R\myLP.Rout" at 1 until EOF
  close file "C:\R\myLP.Rout"
  put it into field "solModell"

  put line 11 of field "solModell" into sol
  put line 12 of field "solModell" into sozTra

  repeat 4 times
    delete char 1 of sol
    delete char 1 of sozTra
  end repeat
  put sol * (-1)  into line 1 of field "Lsg"
  set the itemDelimiter to space
  put item 1 of sozTra  into line 2 of field "Lsg"
  put item 2 of sozTra  into line 3 of field "Lsg"
end mouseUp
```

“Start_R_LP.bat” is a Text-file with one line, which is first saved as “Start_R_LP.txt”; then, replace “txt” by “bat”:

```
C:\"Program Files"\R\R-3.0.2\bin\x64\Rterm.exe --slave < C:\R\myLP.R > C:\R\myLP.Rout 2>&1
```

Next, create a file “myLP.R”, with the code for R give above. After saving the file, replace the extension “txt” by “R”.

The output of the R calculation is put into LiveCode into field "solModel", which looks like this:

```
Model name:
Sozial Transport
Minimize -1 -2
Arbeiter 4 1 <= 32
Computer 1 3 <= 23
Kind Std Std
Type Real Real
Upper Inf Inf
Lower 0 0
[1] 0
[1] -17.54545
[1] 6.636364 5.454545
[1] 32 23
```

The solution (s=6.65, t=5.45, and profit=17.55) is then put into the field "Lsg" and is available in LiveCode for further processing.

Conclusions

This applications shows an example of how to use R from within LiveCode in batch-mode. Any other function and library of R can be used in a similar way. This allows developer to take advantage of the whole arsenal of statistical and mathematical tools in the R system.

References

Holgate C., 2012. LiveCode Mobile Development: Create fun-filled, rich apps for Android and iOS with LiveCode. Packt Publications.

Beroggi G.E.G., 2014. "Analytical Computations in R from LiveCode." *Spring Analytica*. 14/1, 1-4.

Lavieri E.D., Jr., 2013. LiveCode Mobile Development: Create your own exciting applications with 10 fantastic projects." Packt Publications.