KNITR, Hooks and Language Engines

v. 1.4.0

April 16, 2016

Raul Alejandro Buitrago Castellanos

raulhabits@gmail.com

Maestria en ciencias de la informacion y las comunicaciones, con enfasis en ingenieria de software
Universidad Distrital Francisco Jose de Caldas
Colombia

Agenda

Introduction

This conference is related with the use of Hooks and Language engines on KNITR.

- Meeting, understanding, creating, and using the Chunk Hooks.

Introduction

This conference is related with the use of Hooks and Language engines on KNITR.

- Meeting, understanding, creating, and using the Chunk Hooks.
- Working with language engines.

Description

The hooks are a way to increase the capability of KNITR, providing programming options and tools to customize the code execution.

Hook Types

The hooks are classified in two kinds according with the propose,

1. `Chunk Hooks` Set of defined behaviors that can be included in the chunk expressions when are declared.
2. `Output Hooks`

Hook Types

The hooks are classified in two kinds according with the propose,

1. `Chunk Hooks` Set of defined behaviors that can be included in the chunk expressions when are declared.
2. `Output Hooks`

Goal

Customize the project creating and using Chunk Hooks.

Description

Structure

1. begin,
2. options, and
3. envir (environment).

Declaration

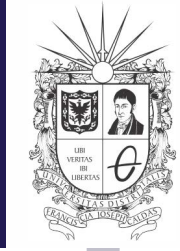

```
<<setup-hooks-customizePlot, echo = TRUE>>=
options(java.parameters="-Xmx2048m")
 if (!"knitr" %in% installed.packages())
  install.packages("knitr")
 library(knitr)
 knit_hooks$set(customizePlot = function(before, options,
     envir){
 par(mar=c(1,1,1,1))
  if (before){
   par(bg=options$plotColor)
  }else{
   par(bg=NULL)
  }
  return()
 })
@
```

setup–customizePlot–hook.Rnw

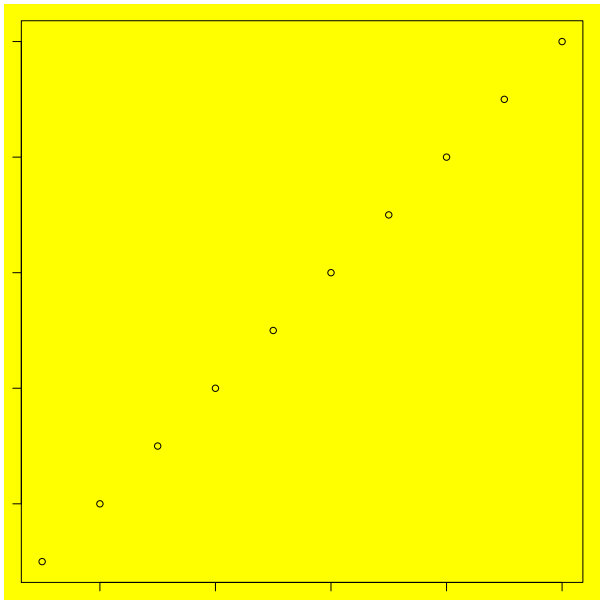
Using a Chunk Hook


```
1
2  \begin{figure}[ht]
3   \subfigure[Using the hook]{
4    <<using-hook, customizePlot=TRUE, plotColor="yellow",
         include=TRUE, out.width="1.5in", echo=FALSE>>=
5     plot(1:10)
6    @
7   }
8  \subfigure[Without using the hook]{
9    <<without-hook, include=TRUE, out.width="1.5in", echo=
         FALSE>>=
10    plot(1:10)
11   @
12  }
13 \end{figure}
```

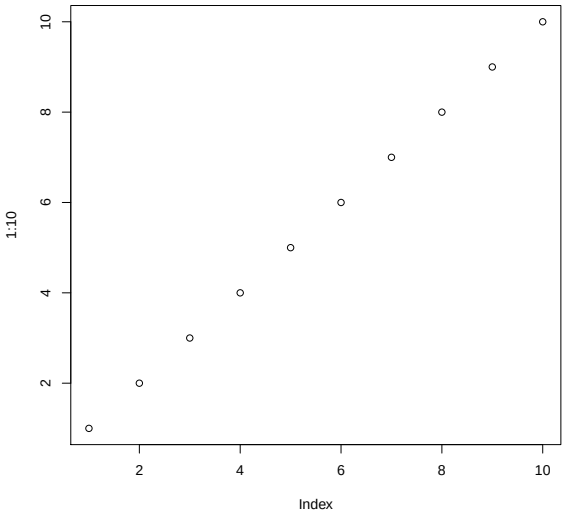
invoke–customizePlot–hook.Rnw



Plot using a Chunk Hook



(a) Using the hook



(b) Without using the hook

Description

- Until this moment the focus has been R, because it's one of the most popular tool to work with statistics, data mining, big data, machine learning, and others.
- But the reality is that R isn't the only one programming language used in reproducible research and KNITR provide us a set of available languages to integrate in the document.

Default available languages

Language	Engine	Code argument
Python	python	-e
Ruby	ruby	-e
*g*awk	*g*awk	-e
sed	sed	-e
shell	sh/bash/zsh	-e
Perl	perl	-e
Haskell	haskell	-e
CoffeeScript	coffee	-e
Groovy	groovy	-e
Node.js	node	-e
Scala	scala	-e
SAS	sas	-e

C++ example


```
library(Rcpp) evalCpp("PI")
```

```
[1] 3.141593
```

```
cppFunction(" int fibCpp(int x) { if (x == 0 || x == 1)
return(x); return (fibCpp(x - 1)) + fibCpp(x - 2); }")
fibCpp(10)
```

```
[1] 55
```

References

[1] XIE, Yihui. Dynamic documents with R and knitr. Chapman & Hall. Second edition. 2015

[2] DE CASTRO KORGI, Rodrigo. El universo LATEX. Facultad de ciencias. Universidad Nacional de Colombia. Segunda edicion. 2003

Thank you