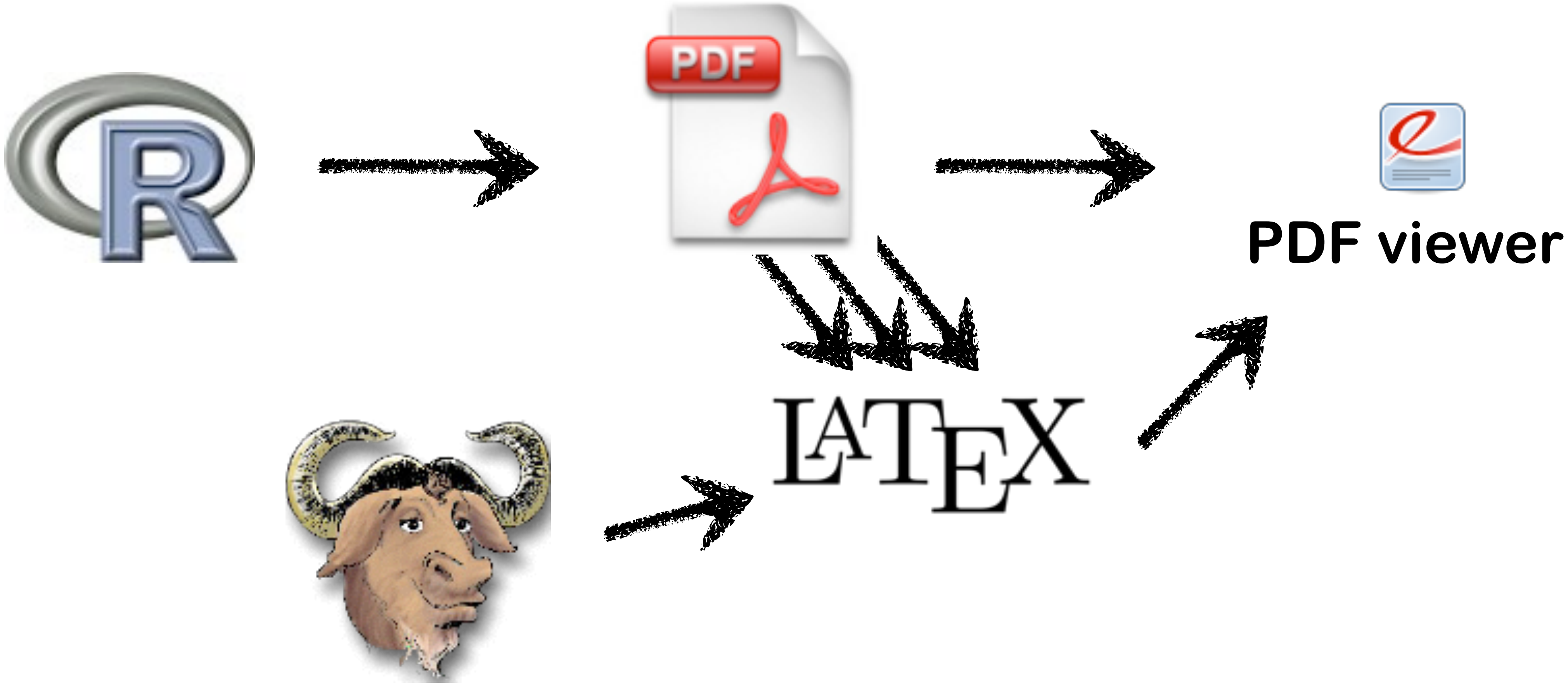
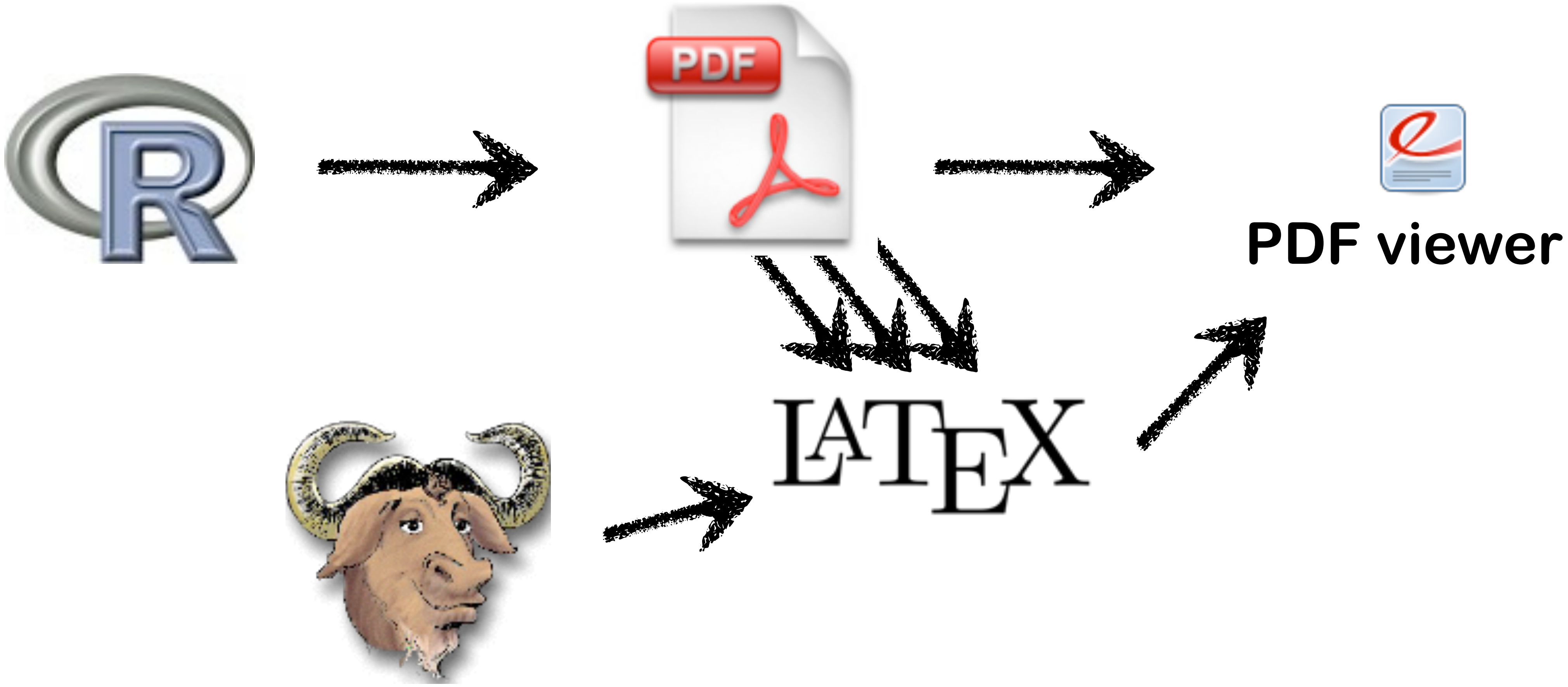
Dynamic reports made from R, viewed anywhere

Wolfgang Huber



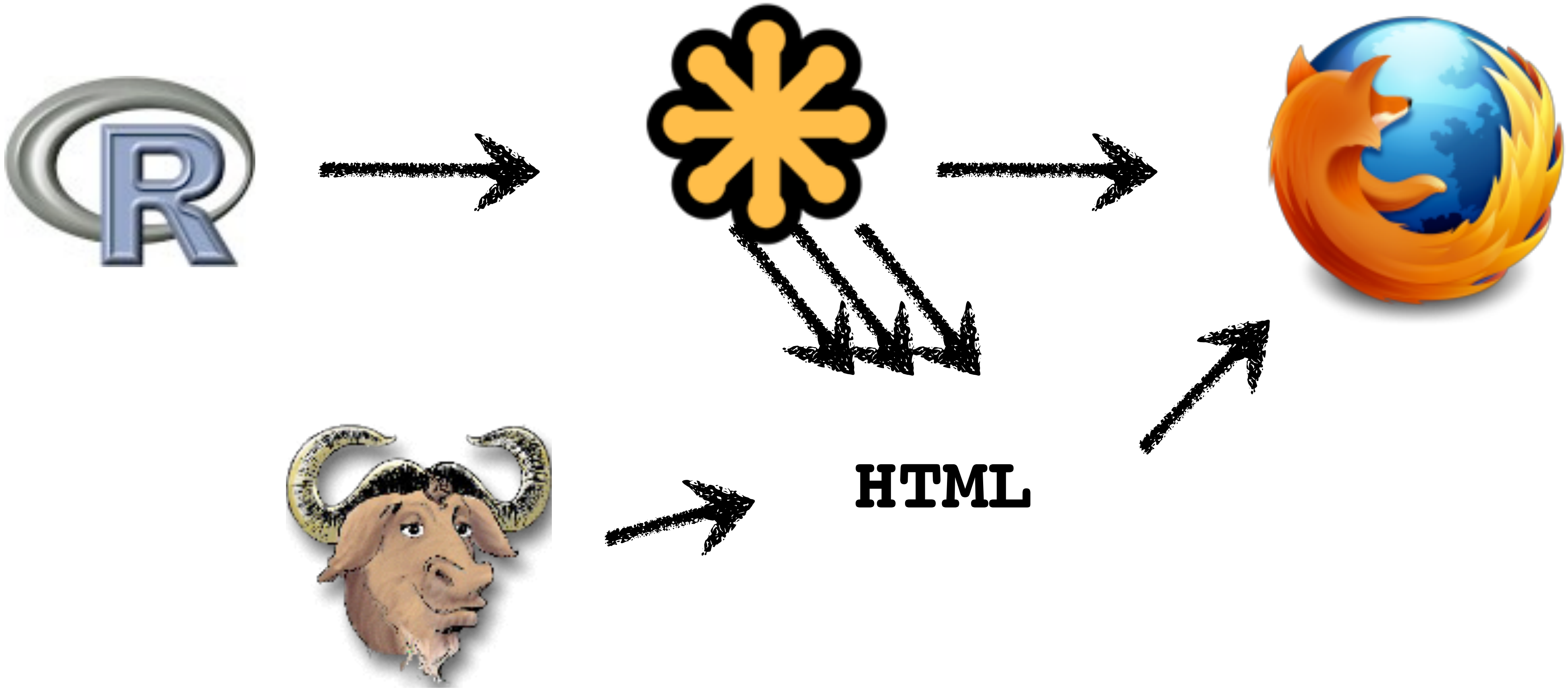
LaTeX

PDF viewer

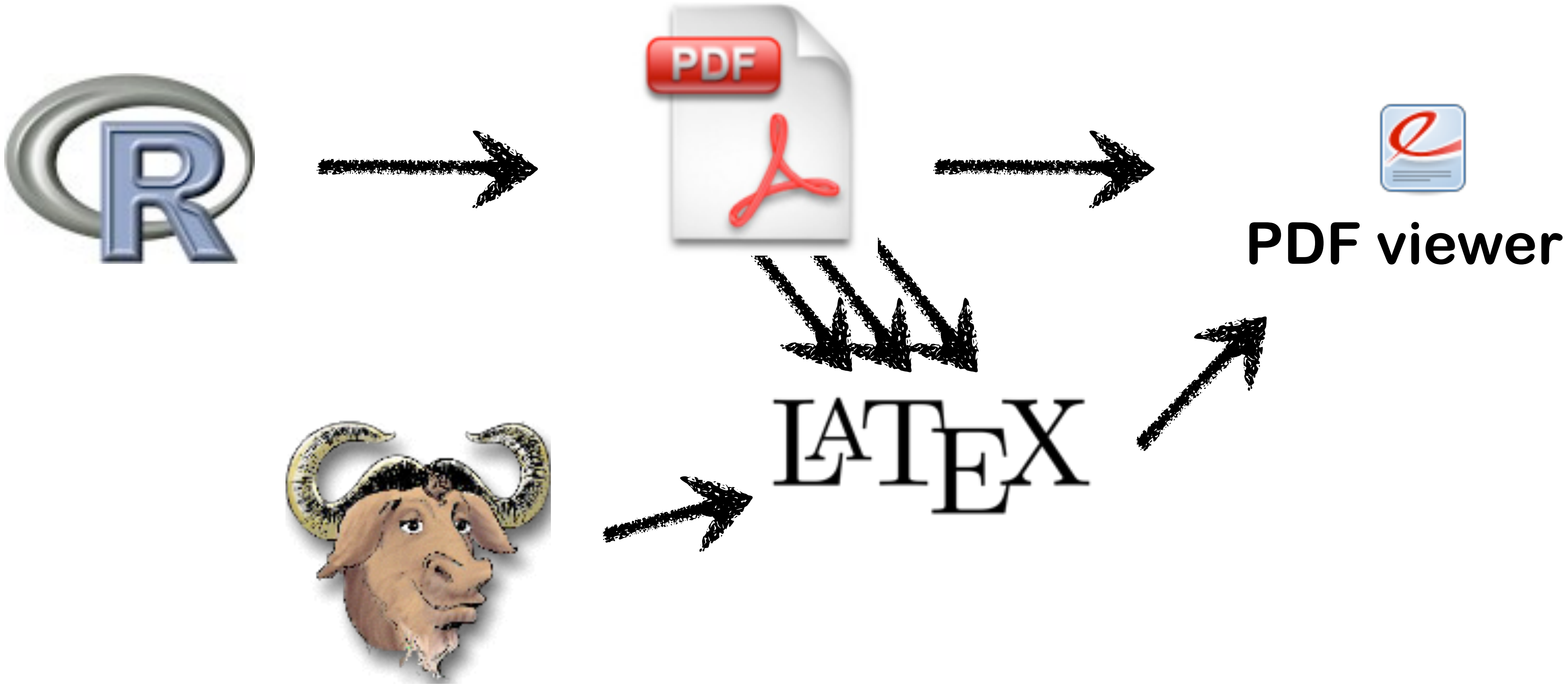


PDF viewer

$\LaTeX$



HTML

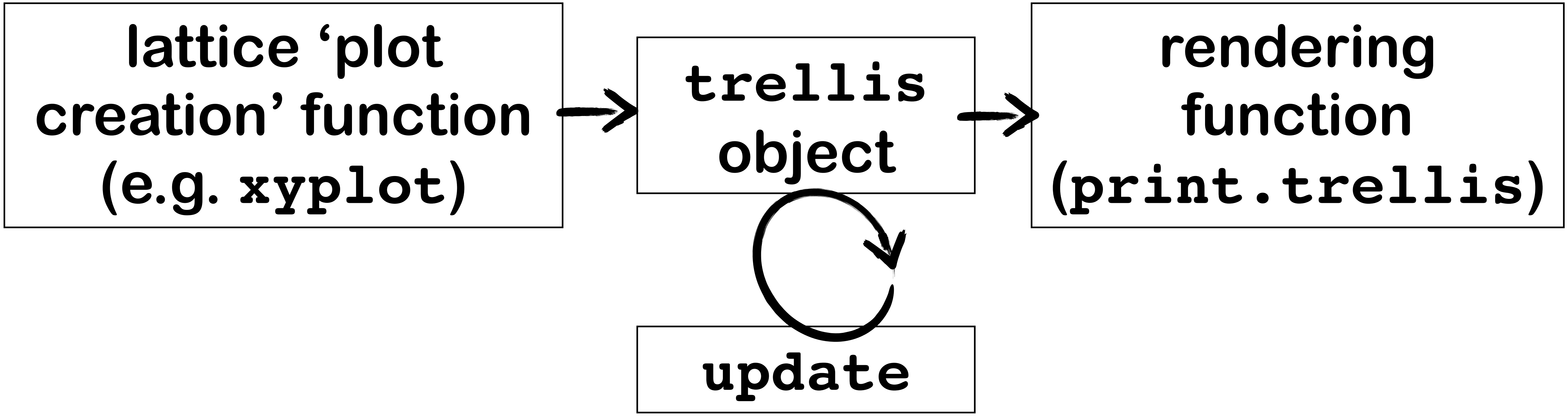


PDF viewer

LaTeX

show report

Graphics objects



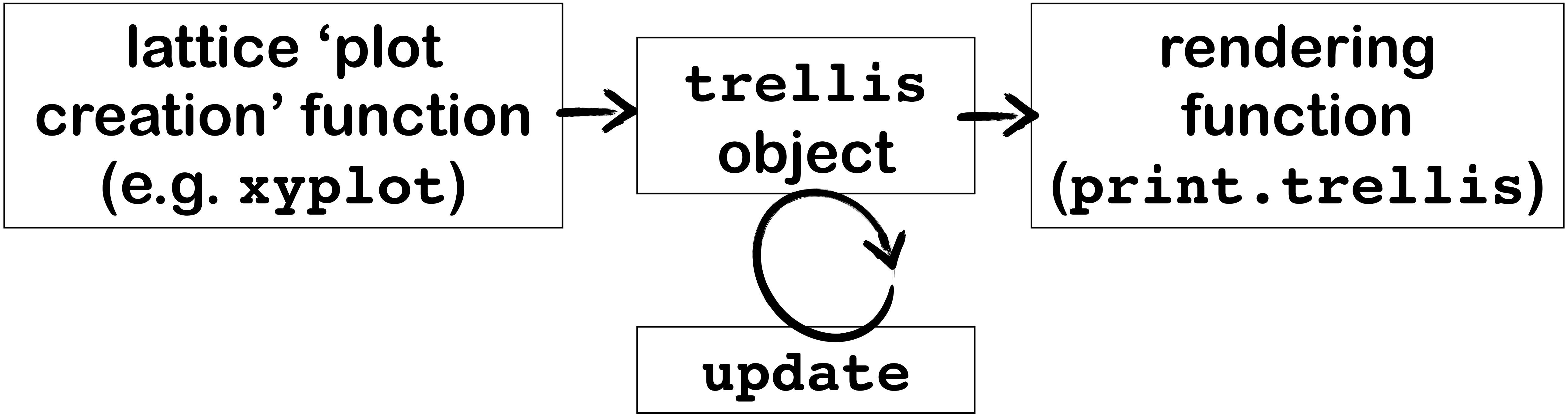
lattice ‘plot creation’ function (e.g. `xyplot`)

`trellis` object

rendering function (`print.trellis`)

`update`

Graphics objects



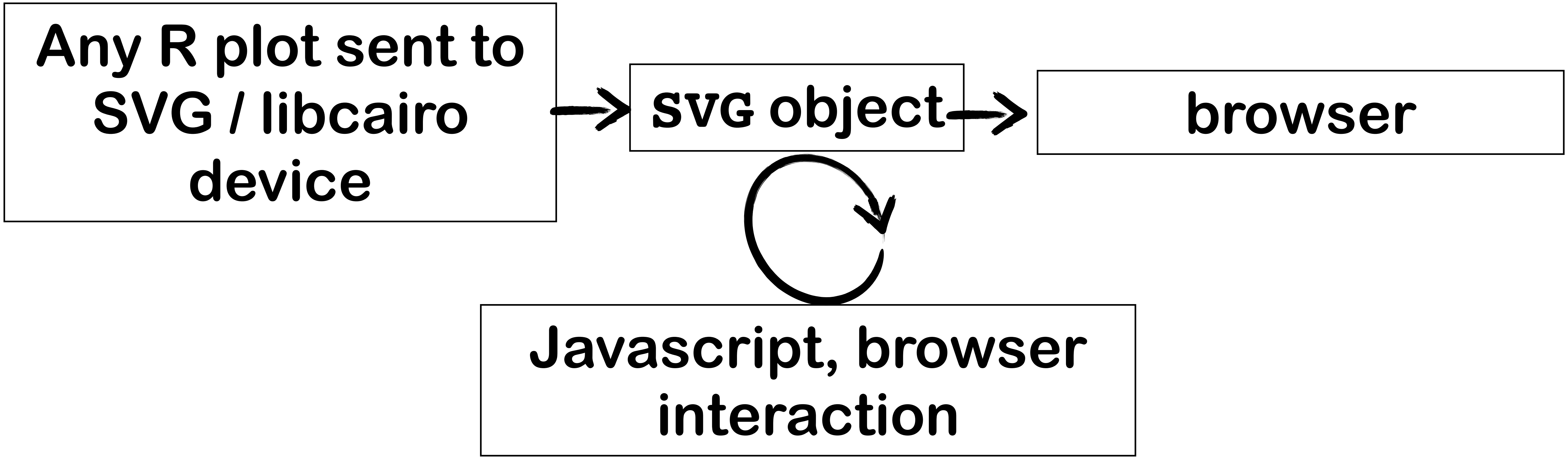
lattice 'plot creation' function (e.g. `xyplot`)

`trellis` object

rendering function (`print.trellis`)

`update`

Graphics objects

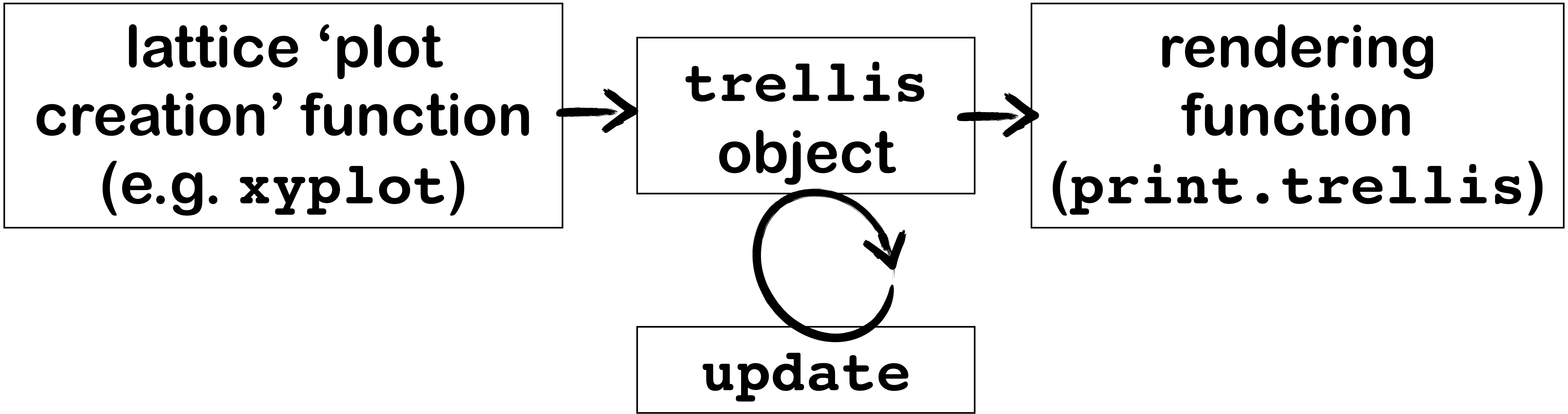


Any R plot sent to SVG / libcairo device

SVG object

browser

Javascript, browser interaction



lattice ‘plot creation’ function (e.g. `xyplot`)

`trellis` object

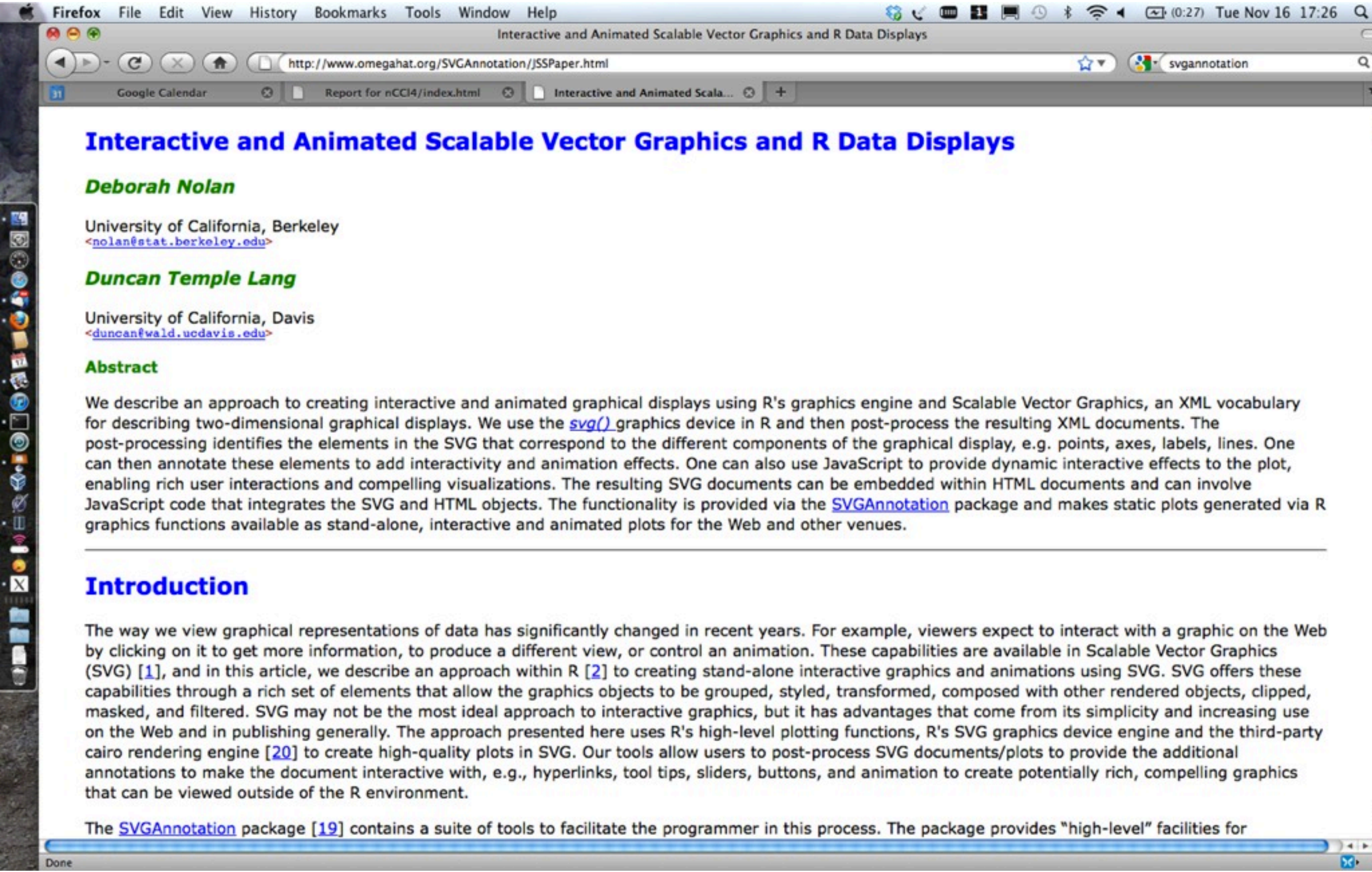
rendering function (`print.trellis`)

`update`

SVG is a form of XML

Show pca.svg

SVGAnnotation package by Duncan Temple Lang


Firefox
File
Edit
View
History
Bookmarks
Tools
Window
Help
(0:27) Tue Nov 16 17:26
Interactive and Animated Scalable Vector Graphics and R Data Displays
http://www.omegahat.org/SVGAnnotation/JSSPaper.html
svgannotation
Google Calendar
Report for nCCl4/index.html
Interactive and Animated Scala...
+
Done

Interactive and Animated Scalable Vector Graphics and R Data Displays

Deborah Nolan

University of California, Berkeley
<nolan@stat.berkeley.edu>

Duncan Temple Lang

University of California, Davis
<duncan@wald.ucdavis.edu>

Abstract

We describe an approach to creating interactive and animated graphical displays using R's graphics engine and Scalable Vector Graphics, an XML vocabulary for describing two-dimensional graphical displays. We use the *svg()* graphics device in R and then post-process the resulting XML documents. The post-processing identifies the elements in the SVG that correspond to the different components of the graphical display, e.g. points, axes, labels, lines. One can then annotate these elements to add interactivity and animation effects. One can also use JavaScript to provide dynamic interactive effects to the plot, enabling rich user interactions and compelling visualizations. The resulting SVG documents can be embedded within HTML documents and can involve JavaScript code that integrates the SVG and HTML objects. The functionality is provided via the SVGAnnotation package and makes static plots generated via R graphics functions available as stand-alone, interactive and animated plots for the Web and other venues.

Introduction

The way we view graphical representations of data has significantly changed in recent years. For example, viewers expect to interact with a graphic on the Web by clicking on it to get more information, to produce a different view, or control an animation. These capabilities are available in Scalable Vector Graphics (SVG) [1], and in this article, we describe an approach within R [2] to creating stand-alone interactive graphics and animations using SVG. SVG offers these capabilities through a rich set of elements that allow the graphics objects to be grouped, styled, transformed, composed with other rendered objects, clipped, masked, and filtered. SVG may not be the most ideal approach to interactive graphics, but it has advantages that come from its simplicity and increasing use on the Web and in publishing generally. The approach presented here uses R's high-level plotting functions, R's SVG graphics device engine and the third-party cairo rendering engine [20] to create high-quality plots in SVG. Our tools allow users to post-process SVG documents/plots to provide the additional annotations to make the document interactive with, e.g., hyperlinks, tool tips, sliders, buttons, and animation to create potentially rich, compelling graphics that can be viewed outside of the R environment.

The SVGAnnotation package [19] contains a suite of tools to facilitate the programmer in this process. The package provides "high-level" facilities for

Related approaches

RSVGTipsDevice and RSvgDevice

gridSVG (for graphics from lattice, ggplot2)

Somewhat generic

arrayQualityMetrics

a set of ‘report objects’ (arrays)

a set of ‘modules’ that describe different aspects on the report objects (PCA, densities, scatterplots, RNA degradation)

plots contain elements (lines, points) that have an n:1-mapping to the report objects

Issues

R’s SVG device / libcairo “loose” the identity of the plotted objects (major...)

User events are handled by JavaScript

(Duncan & colleagues working on that)