

The 'imagedata' package

Dan E. Kelley

September 16, 2016

Abstract

`imagedata` is an R package for digitizing data from images, e.g. for extracting x-y values from a graph.

1 Introduction

There are several commercial and open-sources applications for extracting x-y data from images of graphs, and most of these have a wide set of features, including linear/log axes, magnification of the image under the pointer location, etc.

`imagedata` is more limited than most of these stand-alone applications, but it has some advantages. For one thing, it will continue to work so long as R continues to work. Users need not buy or download a new copy if they switch operating systems. For another, it is written in R and is released under an open-source license, so users may modify it in any way that makes sense for their own work.

2 Typical work flow

In normal use, only the `imagedata()` function is used. Its first argument is the name of a PNG file. Its second is a vector of numerical values along the x axis (typically the location of tick marks), while its third is the same for y . Its fourth is a logical quantity indicating whether the image needs to be corrected for rotation (which may happen with a scanned photocopy).

The steps may be illustrated with diagrams.

If the `rotated` argument to `imagedata()` is set to `TRUE`, the first window shown to the user will look something like Figure 1. As indicated in the Figure, messages appear in the console, indicating the expected user action. Once correction for rotation is complete, processing occurs as follows.

The user is first invited to click at the lower-left edge of what will become a `UNDO` region. This should not intersect a part of the plot that has data. Once the click is made, a coloured box will be drawn, with the word `UNDO` in it. A similar procedure yields a `STOP` region at the top-right of the image. Next, the user is invited to click along the x and y axes, at locations specified by the second and third arguments to `imagedata()`. Corrections may be made by clicking `UNDO`. The process is ended when the user clicks `STOP`. Figure 2 illustrates this procedure.

To add data for a second curve, use `getdata()`, supplying it with some arguments that were returned by a previous call to `imagedata()`. This is all explained in the documentation.

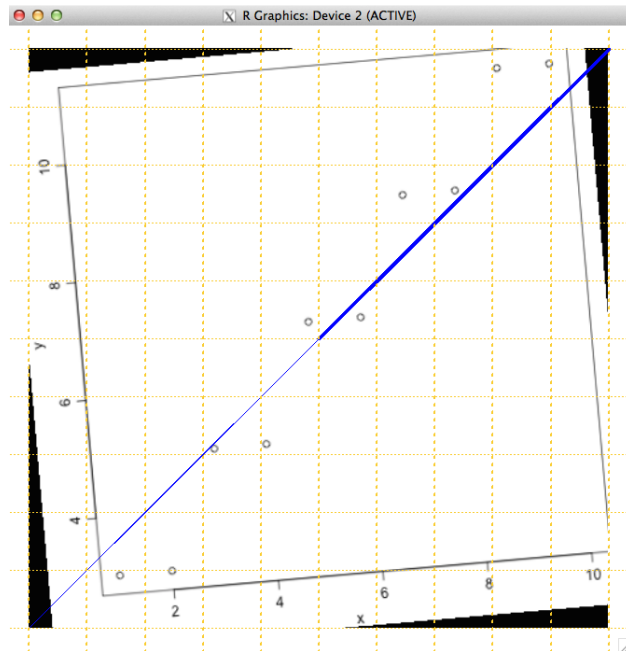


Figure 1: The image-rotation window.

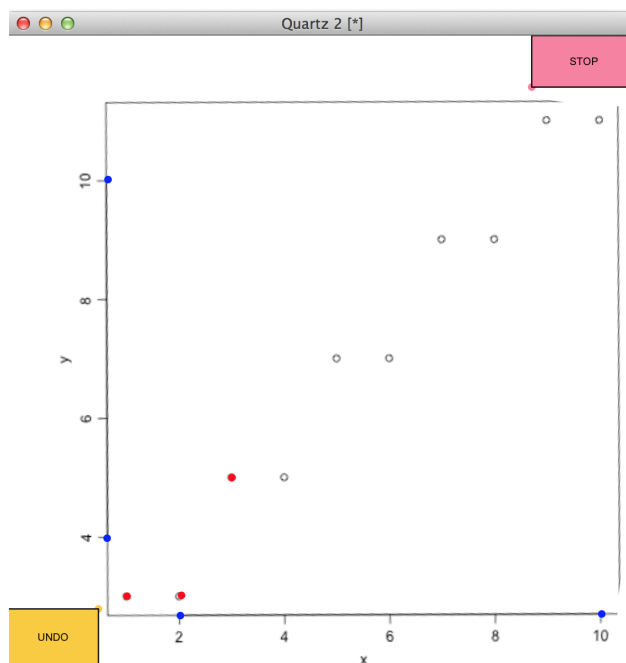


Figure 2: The digitization window, showing a case with 3 points (red dots) have been digitized. The user was sloppy, and should have clicked in the amber-coloured UNDO region after clicking the point at $x = 2$.