



The R package and imagemagick installed in the server. On GNU/Linux and similar systems, you can install the base packages with something like:

```
sudo apt-get install r-base r-base-dev r-recommended imagemagick
```

or

```
yum install R R-devel R-lattice imagemagick
```

depending on the GNU/Linux distribution that you use.

If you see an error message saying something about an error when attempting to **convert...**, then you need to ensure that you installed the system package imagemagick, that will be needed for image conversion server side in R.

Once all requirements are installed, you can start the R console in a terminal by typing "R".

For MS Windows-based OS, you need to ensure that you have imagemagick successfully installed in your system. See these urls for help:

- <http://www.zoharbabin.com/5-steps-to-get-imagemagick-on-xampp-1-7-3>
- <http://bytehash.appspot.com/2012/02/Installing-imagick-for-php-in-windows-under-xampp>

Usage and Parameters

To use this plugin in a wiki page, use the syntax:

Syntax for the R plugin

```
{R()} your R script here {R}
```

or

Syntax for the R plugin

```
{RR()} your R script here which might use some potentially dangerous commands in the server, once that plugin call is validated by an admin{RR}
```

Plugin R vs Plugin RR

Since version 0.5.1 from this mod, you can use plugin rr to execute as admin unsafe commands: you will be required admin validation of those plugin calls.

With plugin r, you will not be requested plugin validation, instead, but unsafe commands will be blocked.

Parameters

You can add some optional parameters to the PluginR call: Parse R syntax and show the output either from the code introduced between the plugin tags or from the file attached to a tracker item sent through PluginTracker. It can also be combined with Pretty Trackers to edit parameters from the script through web forms.

[Go to the source code](#)

Preferences required: wikiplugin_r

Parameters	Accepted Values	Description	Default	Since
(body of plugin)		R Code		
caption		Code snippet label.	R Code	PluginR 0.78
colors		Syntax highlighting with colors. Available: php, html, sql, javascript, css, java, c, doxygen, delphi, rsplus...	r	PluginR 0.78
ln	(blank) 1 0	Show line numbers for each line of code.	1	PluginR 0.78
wrap	(blank) 1 0	Enable word wrapping on the code to avoid breaking the layout.	1	PluginR 0.78
cacheby	(blank) pagename pageid	Write cached files inside a folder containing the Page id (pageid; default option) or the Page name (pagename)	pageid	PluginR 0.88
cacheagedisplay	(blank) 0 1	Display cache last modification date below the graph (default option: do not display)	0	
loadandsave	(blank) 0 1	Load a previous R session (.RData, if any) for the same wiki page so that R object will be used while you work within the same page. For pretty trackers are used (wiki pages with itemId), the R session data (.RData) will be shared for the same itemId across wiki pages	0	PluginR 0.61 (multiuser at 0.86)
cachestrategy	one peruser nocache	Define caching strategy, including LoadAndSave session. Default: One cache and one LoadAndSave session per plugin	one	
customoutput	(blank) 0 1	Write your custom png creation R command. Use tikiRRfilename for value of output. RR does not produce an output file.	0	
cacheduration	0 3600 86400 604800 2592000 31536000	Force cache recalculation after time limit. Default: No limit	0	
attId	int	AttId from a tracker Item attachment. ex: 1. (Optional)		PluginR 0.1
bg	any string except for HTML and PHP tags	The initial background colour (optional). Options: white, yellow, grey, ... and transparent	transparent	PluginR 0.2

echodebug	(blank) 0 1	Show a code block with the R commands to be run even if the R program fails to execute	0	
height	int	Height of the graph (Optional). Options: an integer number in pixels (default) or in units specified. If omitted but width is set, height will be proportional to keep aspect ratio	480	PluginR 0.1
parse_body	alpha	parses the body content to allow data to be generated from other plugins	n	PluginR 0.90 ?
pdf	int	Show link for the creation of the PDF version of the plot. Options: 0 (do not create it, default), 1 (create it). Requires R Cairo package, which can be checked with the following command in the R console: capabilities()	0	PluginR 0.70
pointsize	int	The default pointsize of plotted text, interpreted as big points (1/72 inch) at res dpi (optional). Options: interger number such as 12 or bigger		PluginR 0.1
removen	int	Remove the extra \n tags generated by some R packages out of the user control (such as with charts generated through GoogleVis R package). Options: 0 (do not remove \n tags, default), 1 (remove them all).	0	PluginR 0.76
res	int	The nominal resolution in dpi which will be recorded in the bitmap file (if any). Also used for units other than the default, and to convert points to pixels (Optional). Options: a positive integer (default: 72 dpi). Values higher than 150 usually seem to be too much	72	PluginR 0.1
svg	int	Show link for the creation of the SVG version of the plot. Options: 0 (do not create it, default), 1 (create it). Requires R Cairo package, which can be checked with the following command in the R console: capabilities()	0	PluginR 0.70
units	alpha	Choose units for the width and/or height parameters (Optional). Options: px (default) for pixels, in (inches), cm or mm	px	PluginR 0.1
width	int	Width of the graph (Optional). Options: an integer number in pixels (default) or in units specified. If omitted but height is set, width will be proportional to keep aspect ratio	480	PluginR 0.1
x11	int	Choose whether the server can use X11 to produce graphs in R, or alternatively use dev2bitmap instead (Optional). Options: 1 (R has support for X11, default), 0 (no support for X11 thus using dev2bitmap). These capabilities can be checked in the server with the command in the R console: capabilities()	1	PluginR 0.62

echo	(blank) 0 1	Show a code block with the R commands to be run before running them (similarly to the echo command)	0	PluginR 0.78
onefile	(blank) 0 1	Should all plots appear in one file? This is the default value (1); but if you answer no, they will attempt to appear in separate files in the server, even if you currently will not be able to fetch them easily through the internet browser. This param can be used with figure types svg and pdf; however, not many svg viewers support several plots in one svg file	1	PluginR 0.71
type	(blank) text/csv text/xml	Choose the source file type in the appropriate mimetype syntax (Optional). Options: csv xml. ex: csv. (default). For xml, see documentation for more details on the additional R packages required	csv (text/csv)	PluginR 0.1
wikisyntax	(blank) 0 1	Choose whether the output should be parsed as wiki syntax (Optional). Options: 0 (no parsing, default), 1 (parsing)	0	PluginR 0.1

Display several graphs in the same Plugin R call

Since PluginR version 0.72, a user can create and display several png's within the same PluginR call, provided that the user writes

```
# newgraph
```

before the new graph so that pluginR opens up the required new graphic device.

Set a CRAN mirror in your R call

You can setup a CRAN mirror in your R call at the wiki page level, by means of using this code:

```
r <- getOption("repos") r["CRAN"] <- "http://ftp.heanet.ie/mirrors/cran.r-project.org/" options(repos=r)
```

Replace "http://ftp.heanet.ie/mirrors/cran.r-project.org/" with your local CRAN mirror.

Find results through the tiki search

Tiki is able to index the results from plugin R. You just need to add "r" and "rr" in the list of plugins to be indexed their contents (which also indexes their results)

You can do so in the **Search Control Panel** (tiki-admin.php?page=search):

- **Exclude all plugins:** (X)
- **Except included plugins:** r,rr,andAllYourPluginsOfInterest
 - for example: "fancytable,list,trackerlist,trackerfilter,code,box,remarksbox,quote,fluidgrid,split,r,rr"

And you need to reindex your site. The option in the search control panel **Incremental Index Update** might not be enough, you might need to run a full **Rebuild index** by the Search Control Panel or through a [console command](#) (`php console.php i:r`).

Examples

1.1.1. Example 1: simple command

Direct output

This code,

```
{R()}1:25{R}
```

Would produce:

“

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

1.1.2. Example 2: a simple graph

```
{R()} x <- 1:10; y <- (1:10)^2; plot(x,y); {R}
```

1.1.3. Example 3

Opening a file from a tracker item attachment and let the R script use it, showing the output in the wiki page.

<http://www.geography.uoregon.edu/GeogR/topics/examplesession.htm>

Get this file **sumcr.csv** as example:

<http://www.geography.uoregon.edu/GeogR/data/csv/sumcr.csv>

When opened in a text editor, it should look like (with 89 lines, instead of just those following 4):

```
Location,Reach,HU,CumLen,Length,DepthWS,WidthWS,WidthBF,HUAreaWS,HUAreaBF,wsgrad
HUA-1,A,R,9.20,9.20,0.12,4.10,9.00,37.72,82.80,0.008696 HUA-2,A,G,29.70,20.50,0.21,3.98,9.63,81.66,197.48,0.002927
HUA-3,A,R,51.20,21.50,0.10,4.46,11.43,95.83,245.75,0.001395 ...
```

Upload this file as an attachment to a [Tracker item](#).

1.1.3.1. 3a: attId hardcoded

Create a wiki page with a call to the PluginR, setting the attId in the appropriate param of the plugin.

In this example, let's say that the attId is number 2.

attId=2. csv file with csv extension. data separated with commas

```
{R(attId=2)}summary(data) {R}
```

The data in that csv file has been assigned to the variable **data** in the pluginR.

Thus, **summary(data)** will produce the summary of the variables and data included in that csv file.

Output produced:

Location Reach HU CumLen Length DepthWS HUA-1 : 1 A:20 G:27 Min. : 9.2 Min. : 1.800 Min. :0.0900 HUA-10 : 1 B:46 P:22
 1st Qu.:234.6 1st Qu.: 5.675 1st Qu.:0.1500 HUA-11 : 1 C:22 R:39 Median :454.8 Median : 8.800 Median :0.2000 HUA-12 : 1
 Mean :440.0 Mean : 9.953 Mean :0.2208 HUA-13 : 1 3rd Qu.:616.9 3rd Qu.:13.000 3rd Qu.:0.2700 HUA-14 : 1 Max. :835.0
 Max. :27.300 Max. :0.5100 (Other):82 WidthWS WidthBF HUAreaWS HUAreaBF Min. :1.160 Min. : 3.700 Min. : 4.95 Min. :
 13.32 1st Qu.:2.520 1st Qu.: 7.253 1st Qu.:15.78 1st Qu.: 51.18 Median :3.065 Median : 8.700 Median :28.62 Median : 77.70
 Mean :3.159 Mean : 8.929 Mean :31.89 Mean : 89.43 3rd Qu.:3.862 3rd Qu.:10.312 3rd Qu.:38.66 3rd Qu.:110.95 Max. :5.300
 Max. :19.300 Max. :95.83 Max. :295.00 wsggrad Min. :-4.091e-02 1st Qu.: -2.925e-05 Median : 7.549e-03 Mean : 8.769e-03 3rd
 Qu.: 1.465e-02 Max. : 7.100e-02

1.1.3.2. 3b: attId dynamically referenced

A page A with a [PluginTrackerList](#) lists items from a tracker with the param link=y. The url param has to be like: url="tiki-index.php?page=B&itemId", where **B** here stands for the wiki page containing the plugin R call.

When the users click on the link of an item, they are sent to wiki page B, with an url containing the Id number of that traker item selected. PluginR gets then the attachment id of that tracker item, in order to use it to populate the variable "data".

The syntax, in this other example, is as follows:

```
{R(attId)}summary(data) {R}
```

The same output should be produced as in the previous example 2a.

However, right now, if there is not itemId passed through the url, the R plugin runs and shows some output error on the wiki page.

To see this example in action, install the Profile http://profiles.tiki.org/r_test

1.1.4. Example 4: producing graphs

```
{R(attId=2)} x <- t(data[4]); y <- t(data[5]); plot(x,y); {R}
```



1.1.5. Example 5: adding params to the graphs

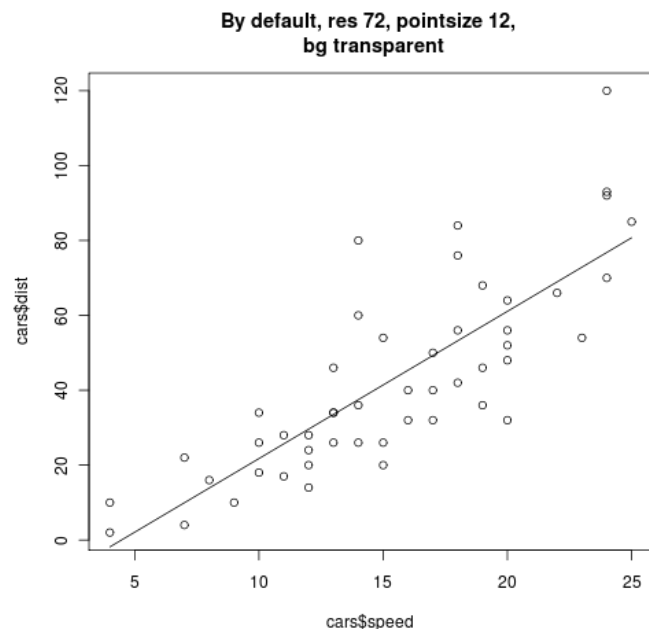
To see this example in action, install the Profile http://profiles.tiki.org/r_test

res 72 dpi

1.1.5.1. Using default: res 72, fontsize 12, Width 500px and rest by default: res 72, fontsize 12, bg "transparent"

This code:
`^ {R(width=500)} data(cars) plot( cars$dist ~ cars$speed ) title('By default, res 72, fontsize 12,\n bg transparent') m <- lm( cars$dist ~ cars$speed ) k <- coef(m) curve( k[1] + k[2]*x , from=min(cars$speed) , to=max(cars$speed) , add=TRUE ){R} ^`

Produces:

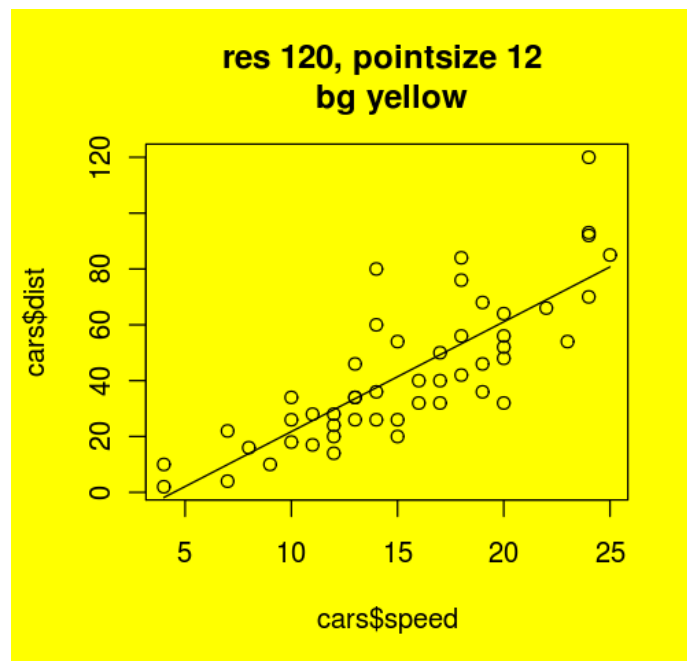


res 120 dpi

1.1.5.2. res 120 & fontsize 12 & bg yellow If not fontsize is set, default value (12) is used.

This code:
`^ {R(width=500 res=120 units=px bg=yellow)} data(cars) plot( cars$dist ~ cars$speed ) title('res 120 & fontsize 12 \n bg yellow') m <- lm( cars$dist ~ cars$speed ) k <- coef(m) curve( k[1] + k[2]*x , from=min(cars$speed) , to=max(cars$speed) , add=TRUE ){R} ^`

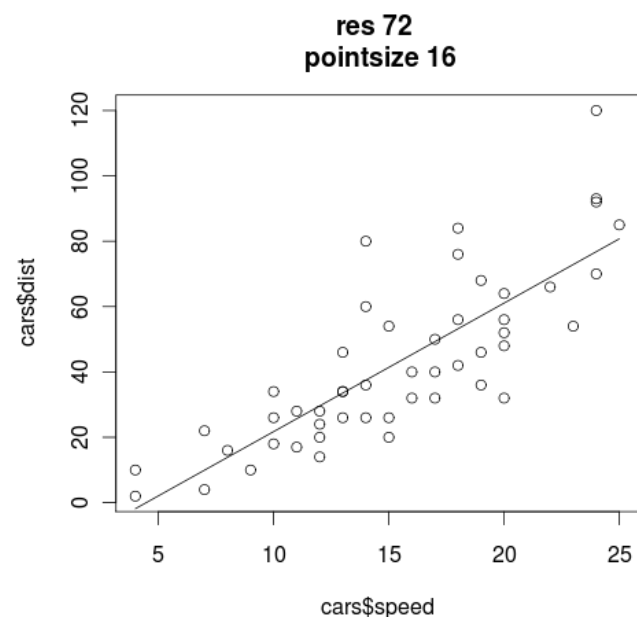
Produces:



1.1.5.3. res 72 & fontsize 16

This code:
`^ {R(width=500 res=72 fontsize=16 units=px)} data(cars) plot( cars$dist ~ cars$speed ) title('res 72 \n fontsize 16') m <- lm( cars$dist ~ cars$speed ) k <- coef(m) curve( k[1] + k[2]*x , from=min(cars$speed) , to=max(cars$speed) , add=TRUE ){R} ^`

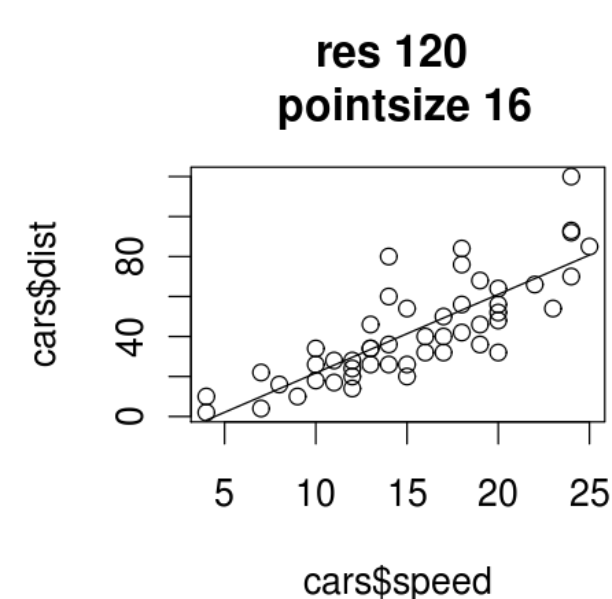
Produces:



1.1.5.4. res 120 & fontsize 16

This code:
`^ {R(width=500 res=120 fontsize=16 units=px)} data(cars) plot( cars$dist ~ cars$speed ) title('res 120 \n fontsize 16') m <- lm( cars$dist ~ cars$speed ) k <- coef(m) curve( k[1] + k[2]*x , from=min(cars$speed) , to=max(cars$speed) , add=TRUE ){R} ^`

Produces:



1.1.6. Example 6: edit R params through Pretty Trackers

What are "**Pretty Trackers**" in the Tiki context?

See this page: [Pretty Tracker](#)

And to see this example in action, for the time being, install the Profile http://profiles.tiki.org/r_test

1.1.7. Example 7: showing results in html

1.1.7.1. Using html tags

This code:

```
{R()}print("<b>hello</b>"){R}
```

Produces:

```
[1] "hello"
```

1.1.7.2. Using wikisyntax param: no param set

This code:

```
{R()}print("__hello__"){R}
```

Produces:

```
[1] "__hello__"
```

1.1.7.3. Using wikisyntax param set to 0

This code:

```
{R(wikisyntax=>0)}print("__hello__"){R}
```

Produces:

```
[1] "__hello__"
```

1.1.7.4. Using wikisyntax param set to 1

This code:

```
{R(wikisyntax=>1)}print("__hello__"){R}
```

Produces:

```
1 "hello"
```

1.1.8. Example 8: using potentially dangerous commands

As an admin you can execute any R command to the server. Just plugin **rr** instead, and validate your plugin calls as needed.

Normal users don't have the ability to validate plugin calls, so that they can only execute safe R commands in the server.

1.1.9. Example 9: loading and saving the R workspace

By default Plugin R runs in --vanilla mode, so that no workspace is loaded at the beginning nor saved at the end, and no human intervention is expected to run the R commands, nor echo's from the R console are printed.

However, for some use cases, you may need to save the R objects and data from your session (call to PluginR) and re-use it afterwards in a new PluginR call, in the same wiki page or others.

1.1.9.1. In the same wiki page

If you have more than one call to PluginR in the same wiki page, you can use the param. **loadandsave** to force R to load any existing workspace data for that wiki page and save it at the end again.

For instance, for a Wiki page called "**My Test**", you would have your workspace data stored in your server at:

```
./temp/cache/My_Test/.RData
```

Note that spaces in names are replaced by the underscore sign ("_"). Therefore, the equivalent page with name "My_Test" (with the underscore instead of the space) would be using the same workspace data.

Example:

a using loadandsave=1	b using loadandsave=1	c using loadandsave=0
<pre>{R(loadandsave="1")} rm(list = ls()); ls(); a<-12; cat("\na = 12\n"); {R}</pre>	<pre>{R(loadandsave="1")} ls(); b<-a*3; cat("\nb = a*3 = "); cat(b); {R}</pre>	<pre>{R(loadandsave="0")} ls(); cat("\nc = b*2 = "); c<-b*2; cat(c); {R}</pre>
<pre>character(0) a = 12</pre>	<pre>[1] "a" b = a*3 = 36</pre>	<pre>character(0) c = b*2 = Error: object 'b' not found Execution halted</pre>

1.1.9.2. Across diferent wiki pages

In this case, we assume that all wiki pages involved in your R calls are a set of scripts that can be run for multiple analysis on one or more of your r scripts. This way, we'll use the [Trackers](#) feature, which will serve as a database engine to keep track of all our analysis projects, with the addition to keep the same workspace data among all pages related to the same project.

Therefore, as far as the itemId param is used in the urls sent between wiki pages (by means of [PluginTrackerList](#), or other tracker-related plugins), the same workspace will be used even if different wiki pages are involved.

Workspace data is stored under the ./temp/cache/ directory. E.g., for project with itemId 33, the workspace will be saved at:

```
./temp/cache/_itemid_000033/.RData
```

Current limitations

1. When you use PluginR combined with [PrettyTrackers](#) for advanced reporting somehow similar to what you can achieve with "Sweave & latex" (e.g. using [Smarty templates](#) in Tiki):
 - angle brackets (i.e.: { lines of code }) have to be surrounded by smarty "literal" tags. For instance, in order to get this if clause working in a smarty template:

```
if (a == 0) { # line 1 of R code # line 2 of R code... }
```

you have to use this code:

```
if (a == 0) {literal} { {/literal} # line 1 of R code # line 2 of R code... {literal} } {/literal}
```

2. Installation of R packages in a server behind a proxy needs the user to export the environment variables for the proxy previous to any action attempting to connect to internet (installing r packages, etc).
 - You can alternatively install whatever R packages you need through a standard console/ssh (by yourself or after requesting it to the sysadmin in your server), after adding the proxy info to the Renviron file in your operating system.You can also try adding this type of line at the beginning of your R script:

```
Sys.setenv(http_proxy="http://your_proxy_url:8080")
```

Former limitations fixed

There were some former limitations when using R code in smarty syntax through wiki pages as templates for [PluginTrackerList](#):

- double parenthesis parsed as internal or alias links,
- square brackets parsed as external links,
- double pipes for OR.

But these one have been fixed by using a new param created in PluginTrackerList in Tiki 6.4 and Tiki 7.1:

- param **"tplwiki"**
which prevents parsing of R code by smarty: only {\${f_1}} type of fields are converted into their equivalent data from the corresponding field 1, etc.

Tips

Some useful tips for advanced usage:

1. When using [PluginTrackerList](#) calls, remember to keep param **showlinks="0"** to prevent that Main tracker fields (those who will be linked to the tracker item) are rendered as html links, which R would not like in many cases.
2. When using tracker fields of type "textarea" (lets say for tracker field 1 ({\${f_1}))), and attempting to show them in a command like remember to surround the tracker field reference with double quotation, like:

```
cat("${f_1}")
```

because if you have more than one line, if you don't surround that field with double quotation marks, R doesn't know that the next line is still from the previous "cat" command.

3. [smarty language function capture](#) can be very handy to add smarty logic based on some values of a tracker item field.
Example:

```
{capture name='saveas'}${f_46}{/capture} {if $smarty.capture.saveas eq "PDF"} {wikiplugin_name=rr
```

```
loadandsave=1 pdf=1 svg=0} cat("f_46 is {$f_46}"); # Do your stuff here with pdf button generation... {/wikiplugin}
{elseif $smarty.capture.saveas eq "SVG"} {wikiplugin_name=rr loadandsave=1 pdf=0 svg=1} cat("f_46 is {$f_46}");
# Do your stuff here with svg button generation... {/wikiplugin} {else} {wikiplugin_name=rr loadandsave=1 pdf=0
svg=0} cat("f_46 is {$f_46}"); # Do your stuff here with no pdf nor svg buttons... {/wikiplugin} {/if}
```

4. To create and display several png's in the same PluginR call, write

```
# newgraph
```

before the new graph so that pluginR opens up the required new graphic device.

5. When using data in R from file attachments to tracker items, you can use the following R session variables to display that content, if desired, from R
 - `file_type` for filetype
 - `data_file` for path and file name (usually in tmp folder on disk when used by php),
 - `data` for the contents of that file
6. When used in conjunction with [Trackers](#):
 - you would usually call PluginR from [PluginTrackerList](#) using the `wikitpl` parameter to indicate to use a custom smarty template stored in a wiki page within Tiki itself.
 - However, in some cases, you might need to do some R magic in the form to insert the data (created using [PluginTracker](#) and the parameter "wiki" to set it to use a wiki page as template for the custom display of that form. Therefore, when you use that form to edit the values in the fields of the tracker item, you can display some advanced calculations, or charts based on values stored in previous fields, etc.

In such scenario, you need to add the "output" smarty modifier so that the field is not passed to R as a piece of html form to insert values, but as the raw values stored in the field, so that R can use them properly.

Therefore, if `{f_1}`, and `{f_2}` are a numeric tracker fields, if you want to display a form to insert data, you would use in the wiki page containing the smarty template, as usual:

```
MyField1: {f_1} MyField2: {f_2}
```

however, if you want to display in a later stage for that tracker item (for instance, when showing the form to enter data for field 3), a calculation based on fields 1 and 2, you will need to use them as `{f_1|output}` and `{f_2|output}` respectively, to pass the values to R properly:

```
MyField3: {f_3} (Remember that MyField1 times MyField2 was {wikiplugin_name=rr} field1 <-
"{f_1|output}" field2 <- "{f_2|output}" myRformula <- field1 * field2 cat(myRformula) {/wikiplugin} )
```

Keep in mind, though, that for not too complicated calculations and algorithms, you can use the [Mathematical Calculation Tracker Field](#) with the [Calculations](#) syntax, and you do not need to follow the PluginR way. But for more advanced use cases and formula/calculations/algorithms needed, you can use the power of R this way. □

External sources of information

Related presentation at UseR! 2011

UseR! 2011: <http://www.warwick.ac.uk/statsdept/user-2011/>

2011 use r_tiki_pluginr from [VHIR Vall d'Hebron Institut de Recerca](#)

Discussion Forums

- **English:** <http://tiki.org/forum24>
- **Spanish:** <http://r-es.org/forum3>

Publications

There are some publications/communications already based on the usage on Tiki & PluginR in different scientific fields:

- 2012. Taller 3: *Interfaces Web 2.0 para R con Tiki*. **IV Jornadas de usuarios de R**. Barcelona, Catalonia (Spain). Thursday 15/11/12, 18:15-20:00:
 - Information & Video: <http://r-es.org/item255?page=IV+Jornadas>
- 2012. "Deploying a tools website in a Bioinformatics unit with Tiki & PluginR". de Pedro, X.; Briansó, F.; Sánchez, A. **XIth Spanish Symposium on Bioinformatics** (JBI2012). Spanish Institut of Bioinformatics and the Portuguese Bioinformatics Network and hosted by the Barcelona Biomedical Research Park (PRBB). Barcelona, Catalonia, Spain. January 23-25. PDF:
 - Full text: http://sgu.bioinfo.cipf.es/jbi2012/wp-content/uploads/2012/01/JBI2012_BookAbstracts_r.pdf
 - Poster: <http://ueb.vhir.org/dl323>
- 2011. "Tiki + PluginR en 2011: interfaces web más fáciles para R". Xavier de Pedro, Ferran Briansó, Àlex Sánchez. **III Jornadas de usuarios de R**. Madrid (España). November 18th.
 - Communication Text & Video: <http://ueb.vhir.org/tools/2011+III+Jornadas+R>
 - Slides: <http://ueb.vhir.org/tools/show:2011+III+Jornadas+R>
- 2011. Taller 5: *Crea interfaces Web 2.0 a tus programas en R con software libre (LAMP + Tiki + PluginR)*. **III Jornadas de usuarios de R**. Madrid (España). Friday 18/11/11, 12:30-14:30h:
 - Information: <http://www.r-es.org/Taller+5>
 - Documentation & Screenscasts: <http://ueb.vhir.org/2011+R-es+Taller+5>
- 2011. "Web 2.0 for R scripts & workflows: Tiki & PluginR". Xavier de Pedro & Àlex Sánchez. The R User Conference 2011. University of Warwick, Coventry, UK. August 16-18th. <http://ueb.vhir.org/2011+UseR>
- 2010. "Usando de forma segura R vía web con Tiki". Xavier de Pedro y Àlex Sánchez. I Congreso de usuarios de R (II Jornadas). Mieres (Oviedo, España). 1-2 de Diciembre de 2010. PDF (in Spanish):
 - Full text: <http://estbioinfo.stat.ub.es/wp-content/uploads/2010/12/RJ-II-Jornadas-R-ES-XavierdePedro.pdf>
 - Slides: http://estbioinfo.stat.ub.es/wp-content/uploads/2010/12/RJ-II-Jornadas-R-ES-XavierdePedro_diapos_v3.pdf
- 2010. "Using R in Tiki for Bioinformatics". de Pedro, X.; Sánchez, A. **Xth Spanish Symposium on Bioinformatics** (JBI2010) (Malaga, Spain. 27-29 October 2010). PDF:
 - Full text: http://estbioinfo.stat.ub.es/wp-content/uploads/2010/09/jbi2010_attachment_64_v3.pdf
 - Poster: http://estbioinfo.stat.ub.es/wp-content/uploads/2010/09/jib2010_poster_Using_R_in_Tiki_v02.pdf
- 2010. "Assessing Student Activity Through Log Analysis From Computer Supported Learning Assignments". X. de Pedro, M. Calvo, A. Carnicer, J. Cuadros, A. Miñarro. In International Congress of University Teaching and Innovation - CIDUI 2010, Barcelona July 2010. <http://cochise.bib.ub.es> <http://cochise.seeds4c.org>

References and links

- R-project: <http://www.r-project.org>
- R Tiki Training site: <https://r.tiki.org>
- Tiki Configuration profiles:
 - http://profiles.tiki.org/r_test
 - http://profiles.tiki.org/R_HeatMaps
- dev: <http://dev.tiki.org/PluginR>
- Sponsorship:
 - Statistics and Bioinformatics Unit. Vall d'Hebron Research Institute (UEB-VHIR). Barcelona, Catalonia, Spain. <http://ueb.vhir.org>
 - EiB group: Estadística i Bioinformàtica, Dept. of statistics, University of Barcelona, Catalonia, Spain.

<http://estbioinfo.stat.ub.es>

- Cochise project - REDICE - University of Barcelona- <http://cochise.bib.ub.es> Copy of Conference Communication and working demo at <http://cochise.seeds4c.org>

alias

[Plugin R](#) | [PluginRR](#) | [Plugin RR](#) | [R](#)