Journal of Statistical Software

August 2013, Volume 54, Issue 5. *http://www.jstatsoft.org/*

RMatlab-app2web: Web Deployment of R/MATLAB Applications

Armin Varmaz
University of
Applied Sciences Bremen

Christian Fieberg
University of Bremen

Andreas Varwig
University of Bremen

Abstract

This paper presents the **RMatlab-app2web** tool which enables the use of R or MATLAB scripts as CGI programs for generating dynamic web content. **RMatlab-app2web** is highly adjustable. It can be run on both, Windows and Unix-like systems. CGI scripts written in PHP take information entered on web-based forms on the client browser, pass it to R or MATLAB on the server and display the output on the client browser. Adjustable to the server's requirements, the data transfer procedure can use either the GET or the POST routine. The application allows to call R or MATLAB to run previously written scripts. It does not allow to run completely flexible user code. We run a multivariate OLS regression to demonstrate the use of the **RMatlab-app2web** tool.

Keywords: web deployment, R, MATLAB, PHP.

1. Introduction

The **RMatlab-app2web** tool allows to make R (R Core Team 2013) or MATLAB (The MathWorks, Inc. 2012) scripts available to a wide audience by creating web interfaces. R and MATLAB respectively run on the server while users only need a standard web browser. Using the **RMatlab-app2web** tool the information which is entered by users on web-based forms is processed by a PHP-written CGI script to R or MATLAB on the server. After the calculation the results are displayed on the client browser.

During the last decade several packages have been developed providing a quick and comfortable access to statistical software to a broad public. Most tools, however, have been developed for Unix-like systems only and focus on providing access to R. Commercial software, such as MATLAB, has mostly been disregarded. With **RMatlab-app2web**, we have developed a tool which closes these gaps. **RMatlab-app2web** is able to run on Windows and Unix-like servers.[1] It further provides access to scripts written in R or MATLAB.[2] Finally, the **RMatlab-app2web** tool supports different methods of data processing (either the GET or the POST routine).

The main components of the **RMatlab-app2web** tool are (1) a set of R and MATLAB functions for decoding the information entered on web-based forms and (2) wrapper shell scripts for Windows and Unix-like platforms which process the information entered on web-based forms to R or MATLAB on the server and display the output on the client browser. To demonstrate the feature of these components, the **RMatlab-app2web** tool comes along with three exemplary applications.

The remainder of this paper is structured as follows. Section 2 provides a brief overview of several related web tools that have been developed so far. In Section 3 the installation and configuration of **RMatlab-app2web** is explained. The differences in the use of the tool on Windows and Unix-like systems are particularly highlighted. In Section 4 the tool's application is demonstrated by the example of a multivariate OLS regression. Some concluding remarks are made in Section 5.

2. Related work

Enabling web forms to communicate with statistical software is not a new idea. During the last decade, a variety of tools have been developed and provided for free use. A listing of several tools that are freely available today is given below.

Rweb (Banfield 1999) provides access to the R command prompt from a web page. It runs R (in batch mode) on the edited code and returns printed and graphical outputs.[3]

CGIwithR (Firth 2003) allows to use R scripts as CGI programs for generating dynamic web content. HTML forms and other mechanisms to submit dynamic requests can be used to provide input to R scripts via the web to create content that is determined within that R script.[4]

rApache (Horner 2005) includes the R interpreter in a web server. In specific it allows the web application development using the R statistical language and environment and the Apache web server. For the communication between the server and R, rApache uses the library **libapreq**.[5]

Rpad (Short and Grosjean 2005) provides access to the R command prompt from a web page but allows also to develop graphical user interfaces based on the functional range of R.[6]

[1] The **RMatlab-app2web** tool has not been tested with Mac OS X.

[2] Due to the fact that R and MATLAB are probably two of the most used statistical software programs the **RMatlab-app2web** tool is based on these programs. Extensions for statistical software programs such as Mathematica, Maxima or SPSS might be possible but are not yet considered in the **RMatlab-app2web** tool.

[3] The remarks follow the official description of **Rweb** on `http://www.math.montana.edu/Rweb/`.

[4] The remarks follow the official description of the **CGIwithR** package on `http://www.omegahat.org/CGIwithR/`.

[5] The remarks follow the official description of **rApache** on `http://rapache.net/`.

[6] The remarks follow the official description of **Ppad** on `http://rpad.googlecode.com/svn-history/r76/Rpad_homepage/`.

- **R-php** (Mineo and Pontillo 2006) consists of two modules. The first module (**R-php** base) provides access to the R command prompt from a web page and enables to edit R code in a web form. As **Rweb** (Banfield 1999) it runs R on the edited code and returns printed and graphical outputs. The second module (**R-php** point-and-click) is almost a R-based graphical user interface which allows to perform some statistical analysis (descriptive statistics and regression analysis) by point-and-click actions based on R.[7]

- **R_PHP_Online** (Chen 2003) is a PHP web interface which provides access to the R command prompt from a web page. As **Rweb** (Banfield 1999) and **R-php** base (Mineo and Pontillo 2006) it runs R on the edited code and returns printed and graphical outputs.

The description above indicates that one can distinguish four features. The first feature is the possibility to get access to the R command prompt from a web page. These packages run R on the edited code and return printed and graphical outputs. Projects providing this possibility are **Rweb** (Banfield 1999), **R-php** base (Mineo and Pontillo 2006) and **R_PHP_Online** (Chen 2003). The second feature is the possibility to use provided web-based graphical user interfaces which are based on R. A project providing this possibility is **R-php** point-and-click (Mineo and Pontillo 2006). The third feature is the possibility to create own graphical user interfaces which are based on editing R code in a web form, which is provided by **Rpad** (Short and Grosjean 2005). The fourth feature is the possibility to use R scripts as CGI programs for generating dynamic web content and thus creating and sharing web applications based on R. Projects providing this possibility are **CGIwithR** (Firth 2003) and **rApache** by Horner (2005). From the above-described projects the **CGIwithR** package by Firth (2003) and the **rApache** package by Horner (2005) are the closest alternatives to the **RMatlab-app2web** tool. But these projects are, as the other ones, based on R and primarily Unix-like platforms. To our knowledge there is no free tool available, enabling to communicate with either R or MATLAB which is able to run on Windows and Unix servers. The **RMatlab-app2web** tool aims to close these gaps.

3. RMatlab-app2web

3.1. Configuration and installation

RMatlab-app2web can be run on Windows as well as on Unix-like servers and requires only basic installations of R and/or MATLAB and a web server. The tool has been tested on the version 2012a and earlier versions of MATLAB and the version 2.15.1 and earlier versions of R. Furthermore, the web server from the **XAMPP** project (v.1.7.7, Apache Friends 2013) is used.[8] Independent from the operating system the tool is used on, all components can be

[7] The remarks follow the official description of **R-php** on http://dssm.unipa.it/R-php/?cmd=home.

[8] We also tested the **RMatlab-app2web** tool on earlier versions of **XAMPP**. **XAMPP** is available on http://www.apachefriends.org/en/xampp.html for Linux, Windows, Mac OS X and Solaris-based operating systems. **XAMPP** is an Apache distribution containing MySQL, PHP and Perl. The **XAMPP** distribution is used to demonstrate the **RMatlab-app2web** tool because (1) it is free of charge, (2) available for most operating systems, (3) easy to install and to use and (4) MySQL, PHP and Perl are already added to the web server. Note that for the use of the **RMatlab-app2web** tool at least PHP has to be added to the web server. There are web servers other than Apache (see Wikipedia 2013b, for a comparison) and distributions other than **XAMPP** (see Wikipedia 2013a, for a comparison). The **RMatlab-app2web** tool has not been explicitly tested on these alternatives.

installed using standard installation routines. Only a few small adjustments are necessary.

On Unix-like systems it might happen that the system's users are not provided with the necessary rights. Any web document has to be located in the directory `/htdocs`. Thus, it is essential that all users of the server have the right to access this directory's content. Any script that is to be executed from a web document needs to be in `/cgi-bin`. Consequently, the system's users need to have the rights to access and to execute the files inside `/cgi-bin`. In case the necessary rights are not granted, this can easily be rectified by the following two commands.

```
chmod [/path]/htdocs a+r
chmod [/path]/cgi-bin a+rx
```

On Windows systems, the users' rights do not need to be modified. However, the web server's standard security settings need to be slightly modified. By default, the option `cgi.force_redirect` of the PHP interpreter is enabled, which conflicts with the web server's security settings. Consequently, the option has to be disabled. This can be done by editing the file `php.ini` which is located in the web server's subdirectory `/php`. The following line has to be added to the `php.ini`.

```
cgi.force_redirect = 0
```

No further modifications are needed.

3.2. Web forms

Any web forms are required to be moved to `/htdocs`. For using the **RMatlab-app2web** tool it is essential to properly adjust the form tag and the input elements of any web form. Within the form tag, two important parameters have to be defined. The first one is the value given to the variable `method`. It determines which method is used to process data from the web form to the statistical software. It can either be `GET` or `POST`. Since both methods can be used with **RMatlab-app2web**, this parameter can be adjusted to the web server's requirements. Secondly, the value of `action` defines the web site or script that is opened when the submit button is clicked on. Depending on the server's operating system, the corresponding CGI script is to be referred here. This will be explained in more detail in the next section.

The input elements of web forms are usually text fields which are defined by the HTML commands `<input type="text">` or `<textarea>`. However, also other types of input elements, for instance hidden elements, can be processed. For using **RMatlab-app2web** it is essential that all input elements are named unambiguously since only elements that are given a unique name can be interpreted.

3.3. CGI using PHP

Although CGI scripts are mostly written in scripting languages such as Perl or PHP, almost any programming language could be used. The CGI scripts used in the **RMatlab-app2web** tool are written in PHP. For enabling CGI scripts to start and execute processes on the system the rights management might have to be changed (depending on the operating system). As mentioned above, for the Apache web server from **XAMPP**, it is sufficient to move all scripts to the directory /cgi-bin. **RMatlab-app2web** provides two CGI scripts, one for Windows and one for Unix-like operating systems. Consequently, either wrapper_windows.php or wrapper_linux.php is to be used. Besides some minor differences in the platform dependent communication with the statistical software, the most important difference between these *wrappers* are their *shebang lines*. While on Unix systems, by default, the PHP CGI scripts can be treated by the PHP command line interpreter, on Windows the executable php-cgi.exe is needed. Consequently, the first line of the PHP script for a standard Windows installation reads as follows.

```
#!"C:/xampp/php/php-cgi.exe"
```

The information processing by the wrapper can be divided into three steps:

1. Reading the data from a web form,
2. communicating with the statistical software and
3. presenting the results in the browser.

At first, the wrapper imports the content of the named input elements of the web form. Before these data are temporarily stored into an environment variable labeled FORM_DATA, the wrapper determines the program the data is to be handed to. This is done by the CGI script's function get_tool and the value of the web form's input element script. Depending on the file extension of the routine to be executed, the data is either prepared for R or MATLAB. Due to the complexity of the operations to be carried out we describe the procedure in Section 3.4. When the calculations by R or MATLAB are finished, the results are readout by the wrapper again. However, depending on the server's operating system, this is done differently. Particularly the communication with MATLAB on Windows is rather tricky. In this case, the results cannot directly be imported by the wrapper and therefore need to be buffered in an external file.

For the wrappers to work correctly, some editing is necessary. Depending on where R and MATLAB are installed, their paths have to be specified. Therefore the lines

```
$PATH['R']=""
```

and

```
$PATH['Matlab']=""
```

of the relevant wrapper file need to be modified.

3.4. R/MATLAB scripts

Similar to the wrapper's structure, the operations carried out by the R/MATLAB scripts can be divided into three steps: Importing and reformatting data, running the calculations and eventually handing the results back to the wrapper.

Data temporarily stored by the wrapper in the environment variable FORM_DATA can easily be imported by R or MATLAB. In both cases, the basic command getenv can be employed.

By `qs <- Sys.getenv("FORM_DATA")` in R and `qs = getenv('FORM_DATA')` in MATLAB, respectively, data is imported into the workspace as the string variable `qs`. To continue processing, `qs` needs to be divided into several sub-strings and reformatted. For this purpose, **RMatlab-app2web** provides the functions `qs2list` and `qs2struct`. The commands `input <- qs2list(qs)` (R) and `input = qs2struct(qs)` (MATLAB) can be used to transform `qs` into a list of elements or a structure array of fields, respectively. The values of any particularly input element, as entered into the web form, can be used by `input$name` (R) and `input.name` (MATLAB).

Once the data are transformed, any calculations on the variable `input` can be carried out. To illustrate the use of **RMatlab-app2web**, three examples are included in the data attachment of this paper. One of the examples, using **RMatlab-app2web** to perform OLS regression from web forms, is explained in detail in the next section. The other two examples demonstrate a) simulation based pricing of financial derivatives and b) the exact way of how data is handed over by the tool.

To transform the strings into numerical variables, basic functions of R and MATLAB are employed. Furthermore **RMatlab-app2web** comes with two additional functions, `qs2mat` and `qscheck`, that enable to check whether the data entered into the web form is formatted properly and can be transformed. For instance, if the inputs contain symbols that are not allowed or cannot be interpreted, this is reported by these functions.

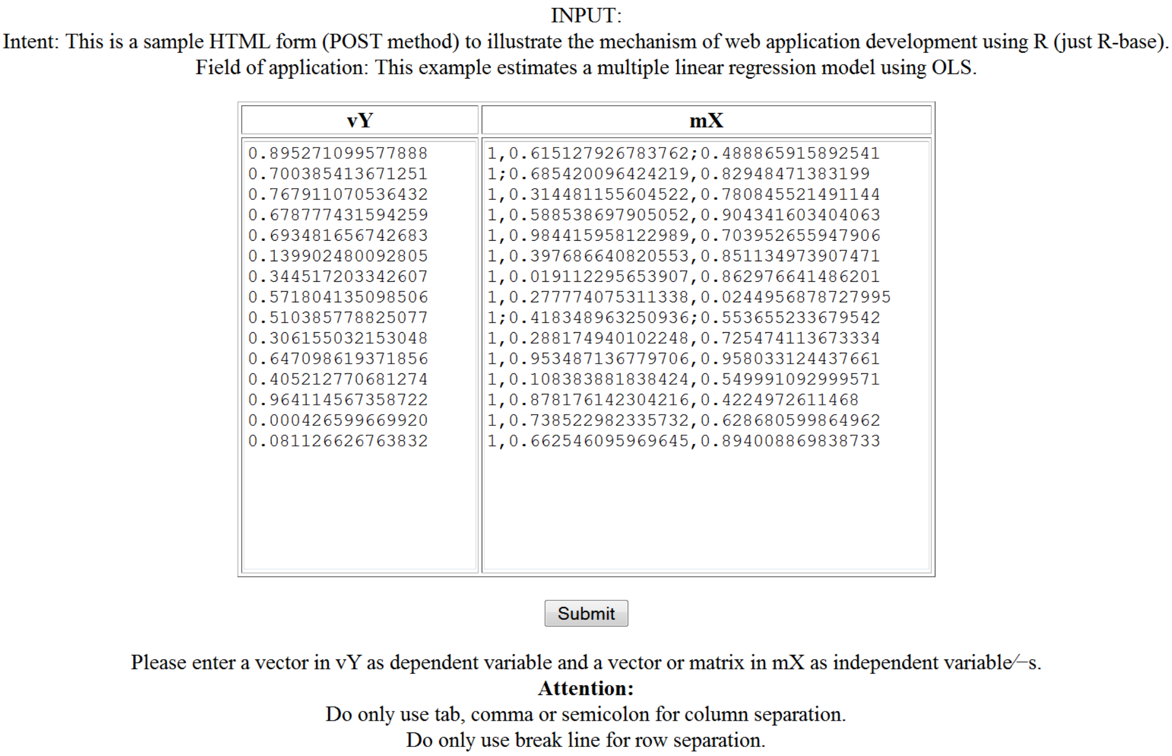
To display the computed results on a browser, HTML code has to be generated. R and MATLAB offer several functions for this task.[9] However, the way how the generated code can be readout depends on the server's operating system. On Unix systems the wrapper can directly access the results by system commands. On Windows systems a direct communication with MATLAB is not possible, so that an intermediate step is needed. The HTML code has to be stored in an external .txt file, for instance using the function `fprint`, which can be interpreted by the wrapper. The code generated by R can, also on Windows, be directly accessed. Consequently, the scripts provided in the data attachment are structured by operating systems. Only the files from directory `/commonfiles` work on both, Windows and Unix. These scripts can directly be moved to `/cgi-bin`.

4. Example

4.1. Configuration

This section demonstrates how **RMatlab-app2web** can be used to perform an OLS regression from a web form. The codes for this example are provided in the directories `/commonfiles` and `/sample2` in the data attachment. All subsequent explanations refer to the use of **RMatlab-app2web** on a Windows system. All calculations are carried out with R and MATLAB. Data from the web form is transferred to R/MATLAB using the method `POST`. To run the example files, they have to be moved to the corresponding directories of the web server first. The wrapper (`wrapper_windows.php`), as well as the R/MATLAB scripts have to be moved to a directory of the web server which is allowed to search for executable scripts. Since we are using the **XAMPP** Apache web server, it is sufficient to move these files to the directory `C:/xampp/cgi-bin`. Accordingly, all HTML files have to be moved to `C:/xampp/htdocs`.

[9] There exist also packages and toolboxes to generate HTML code automatically for free use.



INPUT:

Intent: This is a sample HTML form (POST method) to illustrate the mechanism of web application development using R (just R-base).

Field of application: This example estimates a multiple linear regression model using OLS.

vY	mX
0.895271099577888	1,0.615127926783762;0.488865915892541
0.700385413671251	1;0.685420096424219,0.82948471383199
0.767911070536432	1,0.314481155604522,0.780845521491144
0.678777431594259	1,0.588538697905052,0.904341603404063
0.693481656742683	1,0.984415958122989,0.703952655947906
0.139902480092805	1,0.397686640820553,0.851134973907471
0.344517203342607	1,0.019112295653907,0.862976641486201
0.571804135098506	1,0.277774075311338,0.0244956878727995
0.510385778825077	1;0.418348963250936;0.553655233679542
0.306155032153048	1,0.288174940102248,0.725474113673334
0.647098619371856	1,0.953487136779706,0.958033124437661
0.405212770681274	1,0.108383881838424,0.549991092999571
0.964114567358722	1,0.878176142304216,0.4224972611468
0.000426599669920	1,0.738522982335732,0.628680599864962
0.081126626763832	1,0.662546095969645,0.894008869838733

Please enter a vector in vY as dependent variable and a vector or matrix in mX as independent variable⁄-s.

Attention:

Do only use tab, comma or semicolon for column separation.

Do only use break line for row separation.

Figure 1: The form created by `sample2_R_POST.html`, with some data entered.

4.2. Web forms

Exemplary, Figure 1 shows the web form `sample2_R_POST.html`. The data entered into the web form `sample2_R_POST.html` is handed to the R script `FitLinearModel.R`. The data entered into the web form `sample2_Matlab_POST.html` is handed to the MATLAB script `FitLinearModel.m` respectively. These scripts run an OLS regression of the entered data. The inputs are processed to R/MATLAB using the method POST. Consequently, the form tag of `sample2_R_POST.html` and `sample2_Matlab_POST.html` is specified as follows:

```
<form name="FitLinearModel" method="POST"
action="/cgi-bin/wrapper_windows.php">
```

As it has been outlined in Section 3.3, the data entered into the web form is buffered as an environment variable named `FORM_DATA` for transmission to R/MATLAB. The different inputs of the data entered into the web form are readout as strings by R and buffered in a list. MATLAB buffers inputs in a structure array. The names of the list's components (MATLAB: structure array's fields) are according to the names specified for the web form's input elements. The example files `sample2_R_POST.html` and `sample2_Matlab_POST.html` contain 4 input elements.

`sample2_R_POST.html`:

```
<input type="hidden" name="script" value="FitLinearModel.R">
```

`sample2_Matlab_POST.html`:

```
<input type="hidden" name="script" value="FitLinearModel.m">
```

sample2_R_POST.html and sample2_Matlab_POST.html:

```
<textarea name="vY" cols="20" rows="20"></textarea>
<textarea name="mX" cols="40" rows="20"></textarea>
<input type="submit" name="Submit" value="Submit">
```

The first input element named "script", is a hidden element, defining the program (R:.R/ MATLAB:.m) and the filename (FitLinearModel) that are used for carrying out the OLS regression. The input elements two and three named "vY" and "mX" are text fields. They can be used to enter data representing the dependent variable and the independent variable(s). For demonstration these text fields are pre-filled in the example files sample2_R_POST.html and sample2_Matlab_POST.html. The last input element named "Submit" is the submit button, which initializes the data processing.

4.3. CGI/PHP and R/MATLAB scripts

Without any adjustments, the wrapper works as explained in Section 3.3. Data is readout from the web form, processed to the statistical software (R or MATLAB) and the output is presented in the browser. Consequently, there is no further explanation needed here.

Once the script FitLinearModel.R (MATLAB: FitLinearModel.m) is called, the data from the environment variable is copied into the workspace as a variable named qs.

```
R> qs <- Sys.getenv("FORM_DATA")

MATLAB> qs = getenv('FORM_DATA');
```

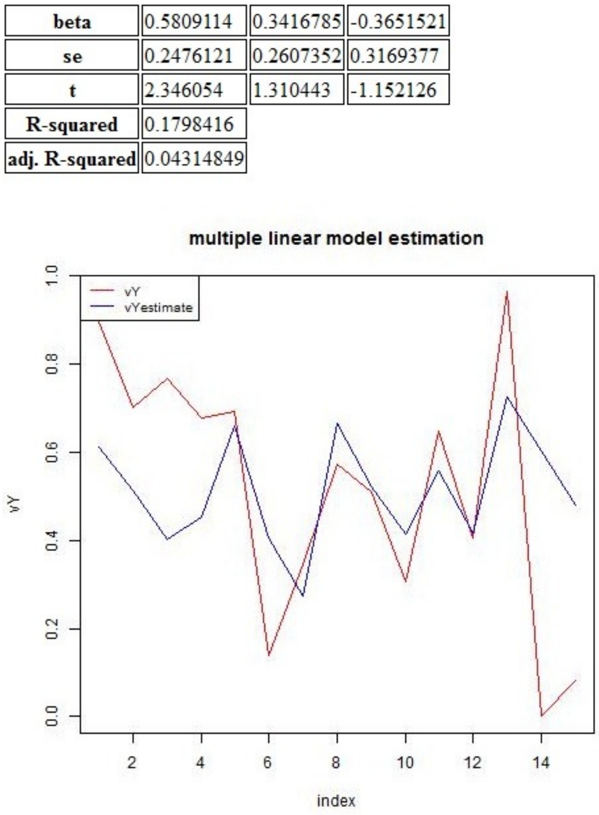
Following that the string qs is transformed into a list (MATLAB: structure array) the dependent variable (svY) and the independent variable(s) (smX) can be extracted.

```
R> input <- qs2list(qs)
R> svY <- input$vY
R> smX <- input$mX

MATLAB> input = qs2struct(qs, fid);
MATLAB> svY = input.vY;
MATLAB> smX = input.mX;
```

In the next step the variables svY and smX are transformed into the format needed for the calculations. In this example, the dependent variable svY is needed to be a vector of numbers and the independent variable(s) smX are needed to be formatted as a matrix of numbers.

For the data to be transformed, the information entered into the web form is needed to meet some formatting requirements. For instance, only certain symbols can be interpreted by the transformation procedures provided with **RMatlab-app2web**. In case of the example files sample2_R_POST.html and sample2_Matlab_POST.html, only the tabulator, comma and semicolon are allowed to separate columns. Lines can only be separated by line break. To



beta	0.5809114	0.3416785	-0.3651521
se	0.2476121	0.2607352	0.3169377
t	2.346054	1.310443	-1.152126
R-squared	0.1798416		
adj. R-squared	0.04314849		

Figure 2: HTML output from the web form `sample2_R_POST.html` (MATLAB: `sample2_Matlab_POST.html`) using pre-filled data.

check whether the input information is formatted properly, **RMatlab-app2web** contains two functions. The first one (`qscheck`) is simply an indicator. The function `qscheck.R` (MATLAB: `qscheck.m`) creates a boolean variable `bqs` that equals 1 if the input information does not contain any forbidden symbols and if the dimension of the information is well-defined. The boolean variable `bqs` equals 0 otherwise. Hence, `bqs` indicates whether data transformation is possible or not. The second function `qs2mat.R` (MATLAB: `qs2mat.m`) already includes the function `qscheck`. If data transformation is possible, the function `qs2mat` automatically creates either a vector (`vY`) or a matrix (`mX`) from the variables `svY` and `smX`. Once the variables for the OLS regression have been created, the regression is executed by the command:

```
R> vBeta <- solve(qr(mX), vY)
```

```
MATLAB> vBeta = mX \ vY;
```

To display the regression's results on the browser, HTML code needs to be produced. There are two ways to do this. First, the application-dependent HTML ouput is specified manually using, for example, the function `cat` (MATLAB: `fprintf`). Second, the application-dependent HTML ouput is specified automatically by adequate packages (MATLAB: toolboxes). In the script `FitLinearModel.R` (MATLAB: `FitLinearModel.m`) the HTML ouput is produced manually using the cat (MATLAB: fprintf) function. The last line of the HTML ouput calls a figure showing the regression's fit to be displayed on the browser. Figure 2 shows output from web form sample2_R_POST.html (MATLAB: sample2_Matlab_POST.html) using pre-filled data. As specified in the script FitLinearModel.R (MATLAB: FitLinearModel.m) the HTML ouput contains summary statistics of the OLS regression and a figure showing realized and estimated values of the independent variable.

The Figure 2 is generated within the R/MATLAB script and temporarily stored in the directory C:/xampp/htdocs. The HTML code created in the script can directly be interpreted by the wrapper and displayed on the browser.[10]

5. Concluding remarks

As has been shown **RMatlab-app2web** is a highly flexible tool that allows one to make R and/or MATLAB scripts available to a wide audience by creating web interfaces. Information can be entered by users in web-based forms, processed to R or MATLAB on the server and outputs are displayed on the client browser. However, it does not allow to run completely flexible user code. The **RMatlab-app2web** tool can be used on Windows and Unix-like operating systems and works with basic installations of R and MATLAB.

References

Apache Friends (2013). ***XAMPP***. URL http://www.apachefriends.org/en/xampp.html.

Banfield J (1999). "**Rweb**: Web-Based Statistical Analysis." *Journal of Statistical Software*, **4**(1), 1–15. URL http://www.jstatsoft.org/v04/i01/.

Chen S (2003). ***R_PHP_Online***. Package Version 0.3, URL http://steve-chen.net/R_PHP/.

Firth D (2003). "**CGIwithR**: Facilities for Processing Web Forms using R." *Journal of Statistical Software*, **8**(10), 1–8. URL http://www.jstatsoft.org/v08/i10/.

Horner J (2005). "Embedding R within the Apache Web Server: What's the Use?" Presented at "DSC 2005: A Workshop on Directions in Statistical Computing", Seattle, URL http://rapache.net/.

Mineo A, Pontillo A (2006). "Using R via PHP for Teaching Purposes: **R-php**." *Journal of Statistical Software*, **17**(4), 1–20. URL http://www.jstatsoft.org/v17/i04/.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Short T, Grosjean P (2005). ***Rpad**: Workbook-Style, Web-based Interface to R*. R package version 0.9-6, URL http://www.rpad.org/Rpad.

[10] This is different for MATLAB scripts on Windows operating systems. It is not possible to execute MATLAB within the operating system command on Windows operating system. Therefore the HTML output is temporarily stored in a separate text file. This text file is then readout by the wrapper and the HTML output is displayed on the browser.

The MathWorks, Inc (2012). *MATLAB – The Language of Technical Computing, Version R2012a*. The MathWorks, Inc., Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Wikipedia (2013a). "Comparison of WAMPs — Wikipedia, The Free Encyclopedia." URL http://en.wikipedia.org/wiki/Comparison_of_WAMPs.

Wikipedia (2013b). "Comparison of Web Servers — Wikipedia, The Free Encyclopedia." URL http://en.wikipedia.org/wiki/Comparison_of_web_servers.

Affiliation:

Armin Varmaz
Schoof of International Business
University of Applied Sciences Bremen
28199 Bremen, Germany
E-mail: armin.varmaz@hs-bremen.de

Christian Fieberg, Andreas Varwig
Chair of Corporate Finance
University of Bremen
28359 Bremen, Germany
E-mail: cfieberg@uni-bremen.de, a.varwig@t-online.de
URL: http://www.fiwi.uni-bremen.de

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 54, Issue 5 *Submitted:* 2011-06-06
August 2013 *Accepted:* 2013-01-03