High-Quality Statistical Plots in LaTeX for Mathematics Education Using an R-Based KeTpic Plug-In

Shunji Ouchi
ouchi@shimonoseki-cu.ac.jp
Economics
Shimonoseki City University
751-8510
JAPAN

Setsuo Takato
takato@phar.toho-u.ac.jp
Pharmaceutical Sciences
Toho University
274-8510
JAPAN

Abstract

The **KeTpic** *software package is comprised of a library of macros to generate standard LaTeX source code for high-quality scientific artwork. Such macros can be implemented in different Computer Algebra Systems. We have recently implemented them in* **R***, which is a popular open-source software tool used in statistical analysis and for graphic output. It is often the case that the default or standard output from* **R** *is not exactly what the users requires, particularly when producing graphics for educational purposes. Through our* **KeTpic** *plug-in, finely-tuned control of various graphical features such as line style, shading, and text display is enabled until the user's needs are fully satisfied. Taking full advantage of the functionality of* **R** *and LaTeX,* **KeTpic** *enables us to produce teaching/learning materials incorporating figures which are designed to help the learner better understand statistical ideas and theories. In this paper we look at the use of the plug-in to generate two basic statistical plots, the histogram and boxplot, which are most useful in descriptive statistics. We will also describe* **KeTpic** *functionality that can be used to produce enhanced graphic output.*

1 Introduction

According to a recent survey conducted by the authors in Japan[1], about 75 percent of those who teach mathematics to first and second-year university or technical college students utilise LaTeX to create teaching/learning materials. Nowadays LaTeX is particularly valuable for university and college mathematics teachers as a tool for preparing printed materials in Japan. LaTeX offers remarkable publishing features and extensive facilities for automating most aspects of typesetting and publishing. However, LaTeX is still not that good when it refers specifically to graphics. Although graphic output is supported, what LaTeX actually provides is a basic and very limited set of graphical capabilities to yield drawings.

Through our KeTpic plug-in, finely-tuned control of various graphical features such as line style, shading, and text display is enabled until the user's needs are fully satisfied. R-based

KETpic commands allow us to convert graphic outputs of R into *Tpic* specials subsequently stored in *Tpic* files. Such files can be embedded into a standard LaTeX file. So we can easily produce LaTeX documments incorporating figures which are designed to help the learner better understand statistical ideas and theories using KETpic.

In this paper we look at the use of R-based KETpic plug-in to enhance standard statistical graphic output of R in LaTeX and some interesting KETpic capabilities are discussed by means of illustrative examples.

2 A Brief Introduction to KETpic

Underlying the KETpic framework is a bundle of macro packages to generate LaTeX source codes for high-quality scientific artwork. Such macros can be implemented in different Computer Algebra Systems(CAS), thus yielding different plug-ins of the program. How the plug-ins run may vary based on the specific CAS, but as this process is transparent to the end-user, it should minimize the time required to learn the program. After loading KETpic in the CAS, users simply need to execute commands following system requests in order to plot graphs and other mathematical data. CAS-embedded KETpic commands generate additional LaTeX source code and files, this generated output can then easily be compiled in LaTeX. As a result, precise and visually compelling graphical figures can be obtained either on a PC display or as printed matter. To date, versions for *Maple*([6]), *Mathematica*([3],[5]), *Scilab*([2]) and *Matlab*([4]) have been developed. They are downloadable for free from the web site `http://www.ketpic.com`.

3 R and Its Graphic Output

R is a popular open-source software environment licenced under the GNU General Public Licence used in statistical computing and production of graphics. It provides a wide variety of statistical and graphical techniques, and is highly extensible.

R is a extremely powerful tool for drawing graphs of statistical distributions. For example, if our aim is to produce a graph of the probability distribution function (p.d.f) of t distribution in R, we simply need to run the following commands in R session:

```
x <- seq(-3,3,0.01)  # ‘ <- ’ means substitution in R
plot(x, dt(x,10),type="l",ylab="Probability density")
# The R function dt returns p.d.f of t distribution
```

giving us the graph below (**Figure 1**) .

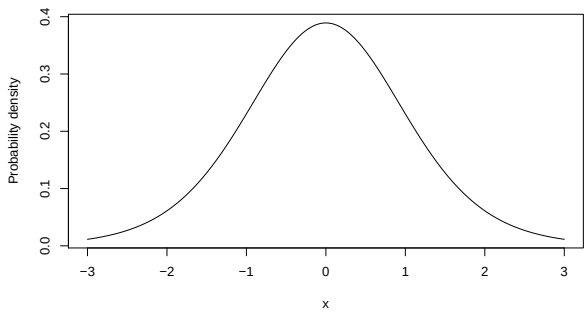

Figure 1 Graphic output of R

R provides a standard set of basic plot types. The `plot` function produces scatterplots, the `hist` function produces histograms, the `boxplot` function produces boxplots, the `barplot` function produces barplots and the `pie` function produces piecharts (see [7]). Producing a simple histogram using R command `hist` is done by:

```
cl <- c(2,2.5,3,4,5,6,8,10,12,15,20,25,30)
hist(data,breaks=cl) # 'data' is data set
```

with the output as seen in **Figure 2**.

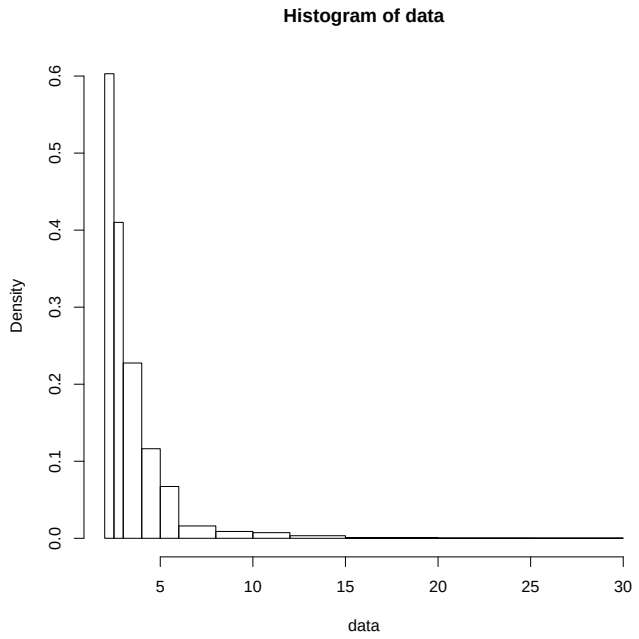

Figure 2 Graphic output of histogram using R

4 KETpic Graphical Pipeline for R

Figure 3 summarizes the KETpic Graphical Pipeline for R.

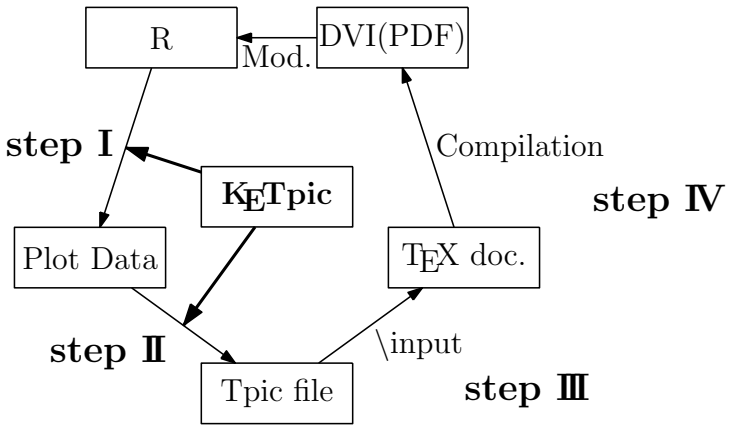

Figure 3 KETpic graphical pipeline for R

We will demonstrate the KETpic session workflow by outlining of process of enhancing the graph (**Figure 1** shown in section 3) according to the pipeline. Our final aim is illustrated in **Figure 4**. KETpic enable us to achieve this aim. While R itself is equipped with some basic commands or functions for modifying the standard graphic output, it doesn't include powerful features users demand. Often to obtain high-quality graphics, users must work with the R-created graphics in a third-party graphical editor such as Adobe Illustrator. KETpic provides an economical alternative to this, and also has lower requirements producing smaller sized files than the EPS format files researchers often need to work with.

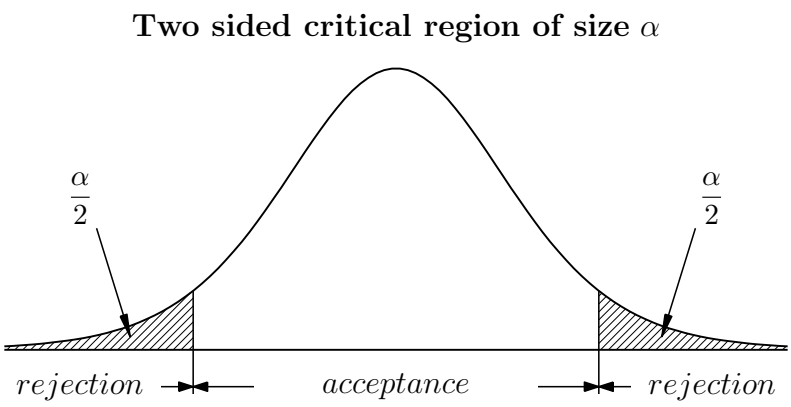

Figure 4 Graphic output of KETpic

The user begins by opening R for a new session. We start by loading the plug in:

```
load("ketpic.Rdata")
```

This is an important step as it ensures all new KETpic commands are automatically available from the very beginning.

step I After setting up the canvas dimensions for LaTeX drawing, the user runs R commands, routines and libraries to perform computations and generate graphic output.

```
Setwindow(c(-3,3),c(-0.1,0.41))
Setscaling(6.46)
G1 <- Plotdata("dt(x,10)","x","N=100")
G2 <- Listplot(c(XMIN,0),c(XMAX,0))
```

step II KETpic commands allow us to convert our graphic data into *Tpic* special code subsequently stored in *Tpic* files.

```
Openfile("fig.tex")   # open tex file at folder
Beginpicture("1.5cm")
# to create \begin{picture} ··· \end{picture} in LATEX
Drwline(G1,G2,1.2)
Endpicture(0)
Closefile()
```

The output of this R session is a collection of plain TEX files containing data for graphical objects.

step III Such files can then be invoked from a source TEX file which should, when run, be compiled to generate a DVI file (`fig.tex` in the sample below).

```
\documentclass[10pt]{article}
\usepackage{ketpic}
\begin{document}

\input{fig.tex}

\end{document}
```

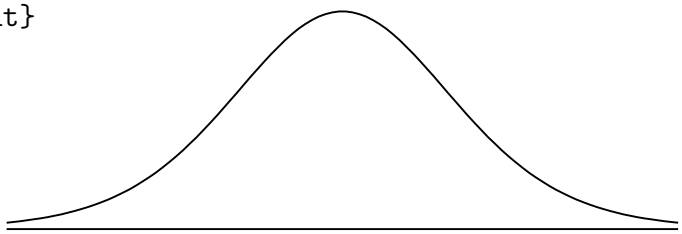

Figure 5 DVI file of graphic output

step IV The DVI file can be further converted into other formats or printed as a paper hardcopy. This cycle can be repeated any number of times allowing the user to fine-tune the graphic output to his/her demands.

For example, if we want to insert the character $\frac{\alpha}{2}$ and an arrow at some point on the existing plot, and shade part of the right tail area of the distribution, we simply need to add following commands in **step I** and **step II** respectively:

in **step I**

```
X1 <- qt(0.95,10) # The R function qt is quantile of t distribution
P1 <- c(X1,0)
P2 <- c(X1,dt(X1,10))
G3 <- Listplot(P1,P2)
G5 <- Hatchdata(list("iii"),list(G1,"s"),list(G3,"e"),
                list(G2,"n"),45,0.4)  # shade area
X2 <- (2*X1+XMAX)/3
P3 <- c(X2,dt(X2,10)/2)
P4 <- P3+0.3*c(1,3)
```

in **step II**

```
Drwline(G3)
Drwline(G5,0.5)
Arrowline(P4,P3,1.5,0.5)   # add arrow
Expr(P4,"ne","\\dfrac{\\alpha}{2}") # add mathematical
                                    annotation
```

giving us the graph below (**Figure 6**).

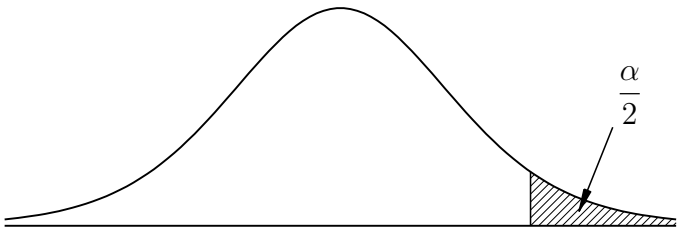

Figure 6 Modified graphic output

Using several KETpic commands in the manner described above, we can produce our final graph (**Figure 4**).

5 Creating High-Quality Statistical Plots

The R-based KETpic plug-in includes a powerful draw function; `Drwhistplot` and `Drwboxplot`. This draw function has been developed to meet various user demands and create high-quality detailed graphs. The function is composed of three main parts:

1. It generates plot data from the data set by the R function;
2. It produces 'graphical framework data' (data for adding a title and setting axis styles) and converts this into *Tpic* special code;
3. It outputs the command sequence to be executed in **step II** (see section 4) and returns the required information to create graphic output.

In the following subsections we look at two basic statistical plots, histogram and boxplot, and describe KETpic functionality to create enhanced graphic output for them. The new Japanese mathematics curricula, which was implemented on April 1, 2009 for the lower-secondary schools and will begin on April 1, 2012 for the upper-secondary schools, aims to identify and explain trends by using histograms for first year junior high school students. Boxplot will be covered in the first year of senior high school under the new curricula.

5.1 Program for Histograms

Histograms are commonly used diagrams which show a graphical representation of a data set in which class frequencies are represented by the areas of rectangles centered on the class interval. The program for creating histogram output is as follows:

```
Drwhistplot(Data, "H.m", c(15,10),"c",
            title=list("Histogram","l","bf"),
            xlab=list("Annual income","n","",5) ,
            ylab=list("No. of persons"),
```

```
            plot=TRUE, densplot=FALSE,breaks=c(2,2.5,⋯,30))
  #  'Data' is data set
  #  c(15,10) sets actual veiwing canvas dimensions (in cm)
  #  optional argument 'breaks=' controls the bin size
```

The character string `H.m` is a variable name. Information on a title and axis styles, histogram plotting data, and a command sequence (`Cmd` shown below) is substituted for it when the `Drwhistplot` function was executed.

```
Cmd <- H.m$commands
fix(Cmd)  # open R data editor if necessary
Maketexfile(Cmd,"fig.tex")
```

The content of `Cmd` is as follows:

```
$commands
      [,1]
 [1,] ""
 [2,] ""
 [3,] "Beginpicture('0.4cm')"
 [4,] ""
 [5,] ""
 [6,] "Drwhistframe(H.m)"
 [7,] "HtickLV(H.m$info$mids,1,1)" #set tick mark on horizontal axis
 [8,] "VtickLV(max(H.m$info$counts),0,0)"
 [9,] "Drwline(H.m[['plotdata']]$histplot)"
[10,] "Dashline(H.m[['plotdata']]$fpplot)"
[11,]
[12,] ""
[13,] ""
[14,] ""
[15,] "Endpicture(1)"
```

This is a matrix which is comprised of several (7 in this case) commands. It is worthy of special mention that the character string `H.m` serves as a command in the function `Drwhistplot`. R is equipped with the function `eval(parse(text="a character string")` which parses a character string and then evaluates it in the environment from which `eval(parse(⋯))` was called. Using this function allows `H.m` to be used in this way. `Maketexfile(Cmd,"fig.tex")` executes the command `Cmd` and converts our graphical data into *Tpic* special code. `Maketexfile` significantly simplifies the process in **step II**. After executing this command and compiling the LaTeX file shown in **step III**, we obtain a DVI file.

Users can easily fine-tune the existing graphical output according to his/her demands as described in section 4. This can be done by either of the following two ways. The first is to type the necessary commands in R data editor (**Figure 7**), the other is by adding the commands to the existing program using R's edit commands. For example, if we want to shade the second bar from the left, we simply need to use the following process dependent on the user interfaces: *graphical interface case:*

After typing `fix(Cmd)` in an open R console window, an R data editor window appears on the screen. Then we enter `Shade(list(Hd[[2]]),0.2)` into a blank line in R data editor.

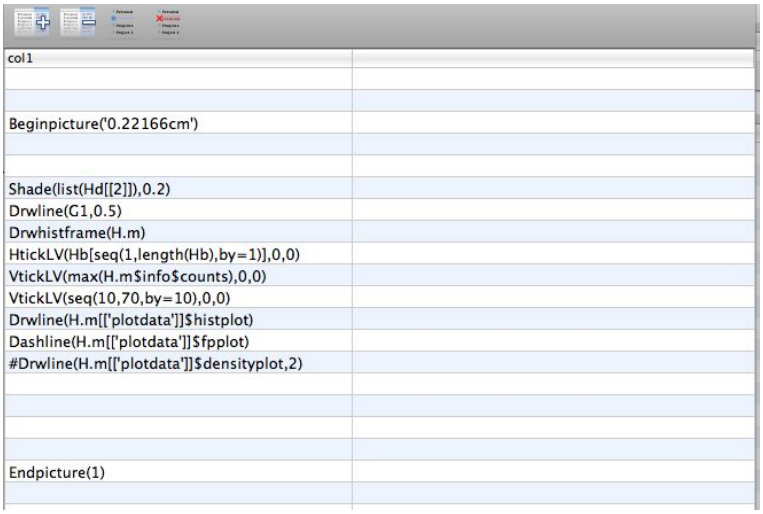

Figure 7 Window of R data editor

command-line interface case:

Add `Insertcom("Cmd",6,"Shade(list(Hd[[2]]),0.2)")` to the existing program.After adding other graphics and annotations to the plot, we obtain the graph illustrated below. It is not necessary to delete other graphics and annotations in this example, however it is possible to do so.

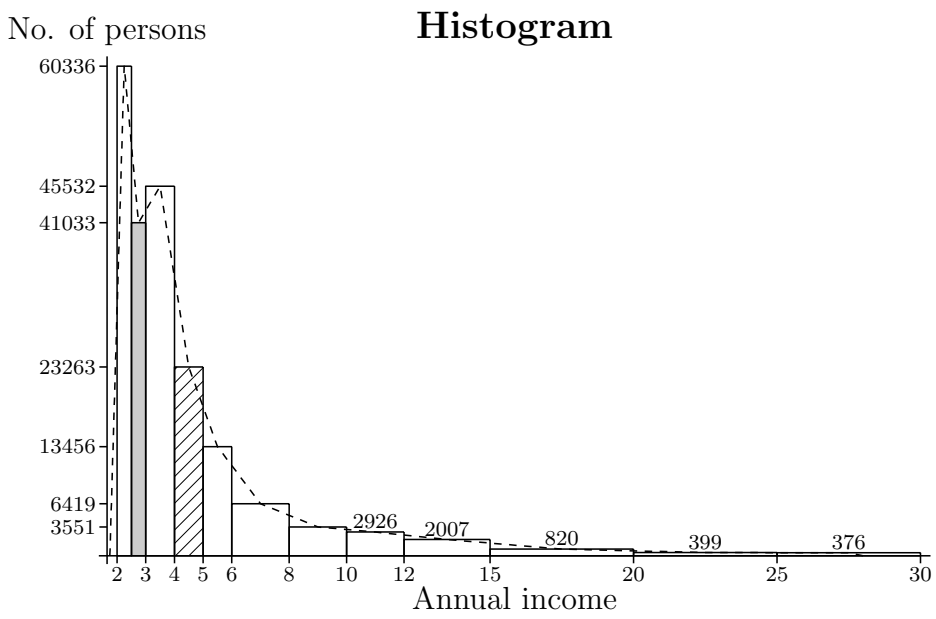

Figure 8 Enhanced graphic output for histogram

5.2 Program for Boxplots

Boxplots are diagrams for presenting necessary information to see the center, spread, skew, and length of tails in a data set. This type of graph allows us to compare many distributions in one figure. The program for creating boxplot output is as follows:

```
capnames <- c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width")
```

```
Title <- list("Boxplot","l","bf")
Cap <- list(capnames,"n",0,0.5)
Drwboxplot(iris[1:4], "BoxD", c(10,10), title=Title, cap=Cap,
            ylab=list(""), ptsize=5, plot=TRUE)
 # iris[1:4] is data set

Cmd <- BoxD$commands
fix(Cmd)  # if necessary
Maketexfile(Cmd,"fig.tex")
```

The function `Drwboxplot` works on the same principle as the function `Drwhistplot`. We can shade any box individually and indicate the figures in the y axis showing the locations of the boxes which stand for the median, and the 25th and 75th percentiles. After running the program and adding other required graphics and annotations to the plot, we obtain the graph below.

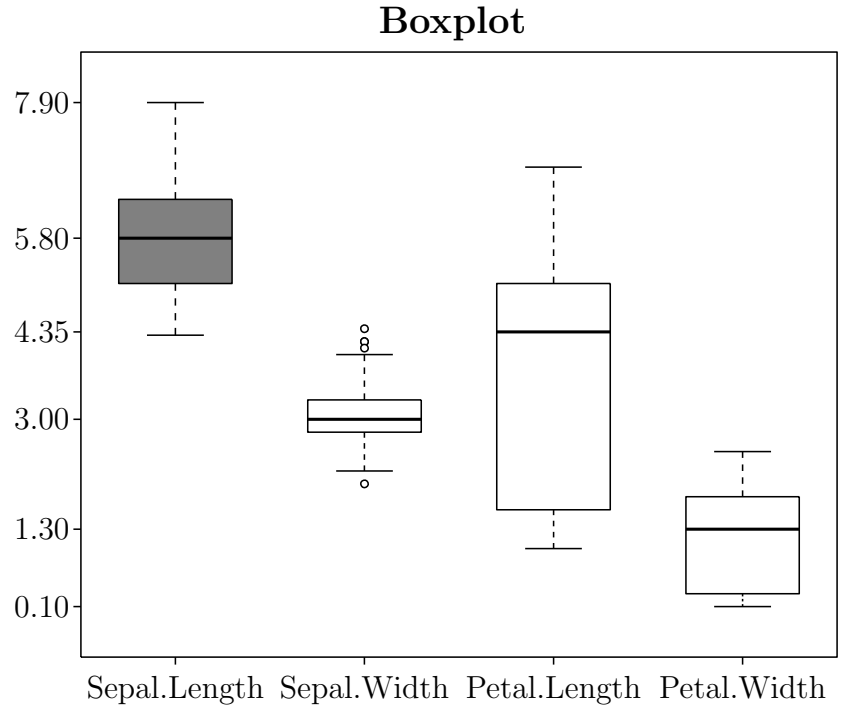

Figure 9 Enhanced graphic output for boxplot

6 Conclusion and Further Development

We have developed an R-based KETpic plug-in to yield high-quality statistical graph output to be embedded into standard LaTeX. Currently the draw function is able to produce histograms and boxplots. In the future we intend to expand the scope of the function to enable the output of a greater range of statistical graphs designed to help the learner better understand statistical ideas. We will enhance the power of the R-based KETpic plug-in, bringing increased functionality, and creating a user-friendly system.

.

References

[1] Kiyoshi Kitahara, Takayuki Abe, Masataka Kaneko, Satoshi Yamashita and Setsuo Takato,"Towards a More Effective Use of 3D-Graphics in Mathematics Education – Utilization of KETpic to Insert Figures into LaTeX Documents–",to appear in The International Journal for Technology in Mathematics Education, Vol. 17, Number 3, 2010.

[2] Koshikawa, H., Kaneko, M., Yamashita, S., Kitahara, K., Takato, S.,: Handier Use of Scilab to Draw Fine LaTeX Figures – Usage of KETpic Version for Scilab–.Proc. ICCSA 2010, IEEE Press, 39-48

[3] M. Kaneko, T. Abe, M. Sekiguchi, Y. Tadokoro, K. Fukazawa, S. Yamashita and S. Takato, "*CAS-aided Visualization in LaTeX documents for Mathematical Education*," Teaching Mathematics and Computer Science, vol. 8, issue 1, (2010)

[4] A. Galvez, A. Iglesias and S. Takato, "*New Matlab-Based KETpic Plug-In for High-Quality Drawing of Curves*," 2009 International Conference on Computational Sciences and its Applications, IEEE Press, 2009, pp.123-131.

[5] M. Sekiguchi, M. Kaneko, Y. Tadokoro, S. Yamashita and S. Takato, "*A New Application of CAS to LaTeX-Plottings*," Lecture Notes in Computer Science, Springer-Verlag, 4488, pp. 178-185, 2007.

[6] M. Sekiguchi, S. Yamashita and S. Takato, "*Development of a Maple Macro Package Suitable for Drawing Fine TeX-Pictures*," Lecture Notes in Computer Science, Springer-Verlag, 4151, pp. 24-34, 2006.

[7] P. Murell, "*R Graphics*," Chapman & Hall/CRC, 2006.