

# R Microplots in Tables with the `latex()` Function

Richard M. Heiberger, `rmh@temple.edu`

`http://astro.ocis.temple.edu/~rmh/`

Statistical Science, Fox School, Temple University

Microplots (sparklines) are often used within cells of a tabular array.

We describe several R functions that simplify the use of microplots

within  $\text{\LaTeX}$  tables constructed in R with `Hmisc::latex` or a similar function.  
within HTML tables constructed with the **htmlTable** package.

We show examples using **base** graphics, **lattice** graphics, and **ggplot2** graphics.

These functions work in  $\text{\LaTeX}$  documents constructed

directly in  $\text{\LaTeX}$ ,

with the R packages **Sweave**, **knitr**, or **rmarkdown**,

and with the Emacs package **org-mode**.

# 1 Boxplots of iris data with lattice and latticeExtra

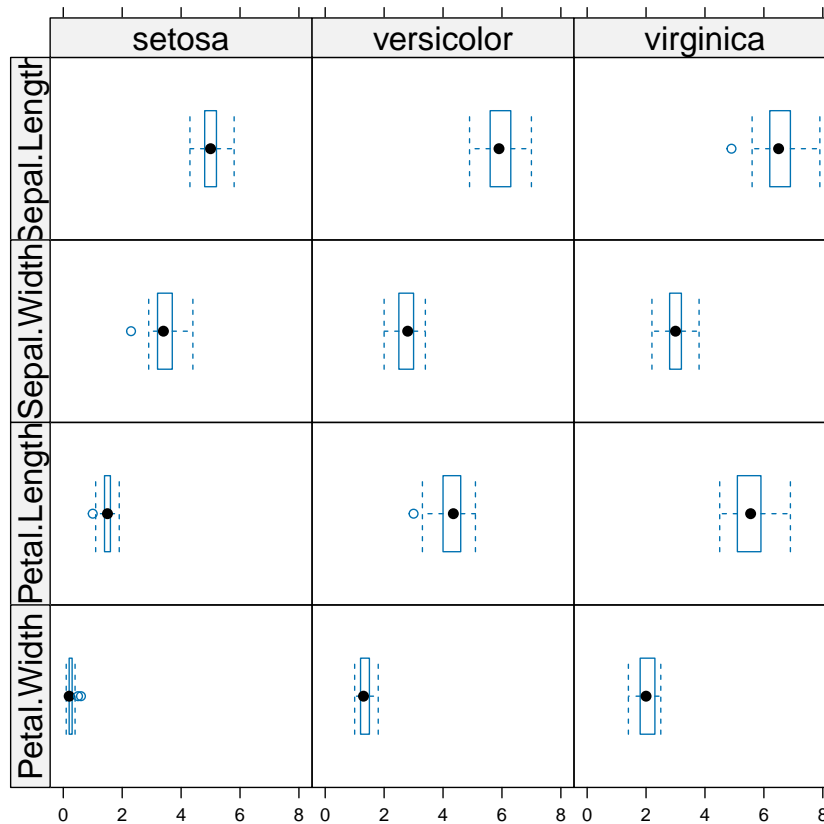














Figure 1: `useOuterStrips(`  
`bwplot(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width`  
`| Species, data=iris, outer=TRUE, as.table=TRUE))`

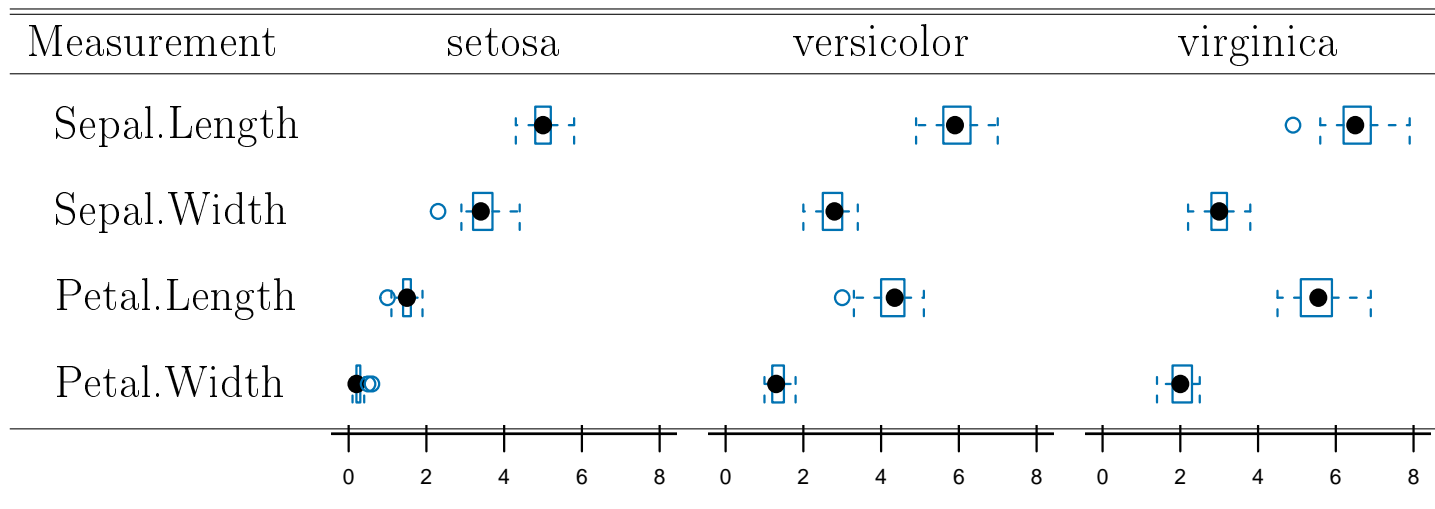
## 2 Individual boxes placed into a $\LaTeX$ tabular environment

Table 1: Measurement by Species

Measurement	Species		
	setosa	versicolor	virginica
Sepal.Length			
Sepal.Width			
Petal.Length			
Petal.Width			













### 3 Individual boxes in a table with the $x$ -scale displayed

Table 2: Measurement by Species, with  $x$ -scale



## 4 Transposed L<sup>A</sup>T<sub>E</sub>X table

Table 3: Species by Measurement

Species	Measurement			
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa				
versicolor				
virginica				

## 5 Individual boxes embedded into a more interesting table

Table 4: Five Number Summary and Boxplots for each Species and Measurement

Species	Measurement	Five Number Summary					Box Plots
		min	Q1	med	Q3	max	
<b>setosa</b>							
	Sepal.Length	4.3	4.8	5.00	5.2	5.8	
	Sepal.Width	2.3	3.2	3.40	3.7	4.4	
	Petal.Length	1.0	1.4	1.50	1.6	1.9	
	Petal.Width	0.1	0.2	0.20	0.3	0.6	
<b>versicolor</b>							
	Sepal.Length	4.9	5.6	5.90	6.3	7.0	
	Sepal.Width	2.0	2.5	2.80	3.0	3.4	
	Petal.Length	3.0	4.0	4.35	4.6	5.1	
	Petal.Width	1.0	1.2	1.30	1.5	1.8	
<b>virginica</b>							
	Sepal.Length	4.9	6.2	6.50	6.9	7.9	
	Sepal.Width	2.2	2.8	3.00	3.2	3.8	
	Petal.Length	4.5	5.1	5.55	5.9	6.9	
	Petal.Width	1.4	1.8	2.00	2.3	2.5	

## 6 How does it work?

There are two tasks. The **microplot** package provides functions for each task.

1. Isolate the contents of each panel of a multipanel graph into its own pdf file.

**lattice**: functions `layoutHeightsCollapse` and `layoutWidthsCollapse`  
and argument `layout=c(1,1)`

**ggplot2**: function `theme_collapse` and loop through panels

**base**: adjust `par` arguments and `xlim` and `ylim`

2. Automate construction of the graphics statements.

**L<sup>A</sup>T<sub>E</sub>X**: `as.includegraphics` function

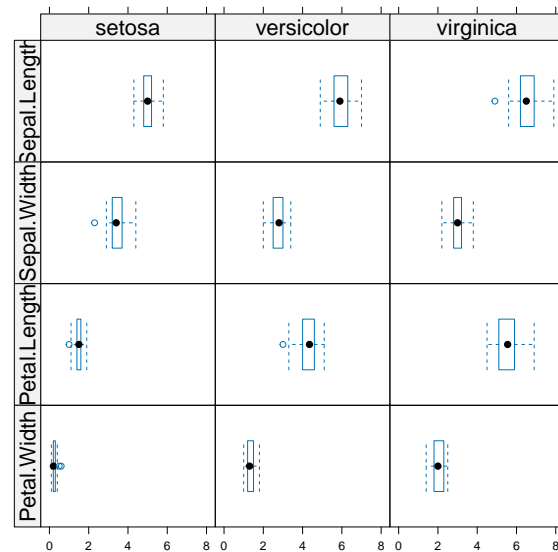
**HTML**: `as.htmlimg` function

**org-mode**: `as.orgtable` and `as.orgfile` functions

## 7 lattice

```
## boxplot matrix of iris data
irisBW <-
  bwplot(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width |
         Species,
         data=iris, outer=TRUE, as.table=TRUE,
         scales=list(alternating=FALSE),
         xlab=NULL,
         par.strip.text=list(cex=1.5))
```

```
## pdf of boxplot matrix
pdf("irisBW.pdf")
useOuterStrips(irisBW)
suppress <- dev.off()
```

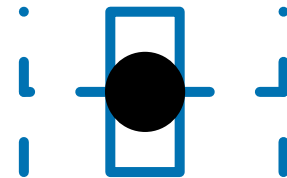




```
## twelve individual boxplots without axes
irisBW.update <-
update(irisBW,
      xlab=NULL,
      par.settings=list(
        layout.heights=layoutHeightsCollapse(),
        layout.widths=layoutWidthsCollapse(),
        axis.line=list(col="transparent")),
      layout=c(1,1)
    )

## create 12 pdf files, one per boxplot
pdf("irisBW%03d.pdf", onefile=FALSE, height=.4, width=1.6) ## inch
irisBW.update
suppress <- dev.off()
```

The first panel in file `irisBW001.pdf` is shown here.



The functions `layoutHeightsCollapse` (shown here) and `layoutWidthsCollapse` set the vertical and horizontal space for everything in a plot, except the panel itself, to 0.

```
> layoutHeightsCollapse
function (...)
{
  x.settings <- lattice::trellis.par.get()$layout.heights
  x.settings[] <- 0
  x.settings$panel = 1
  inputs <- list(...)
  if (length(inputs))
    x.settings[names(inputs)] <- inputs
  x.settings
}
```

The function `as.includegraphics` wraps the graph file names into the format used by the  $\text{\LaTeX}$  **graphicx** package.

```
> graphnames[1:2]
[1] "irisBW001.pdf" "irisBW002.pdf"
> graphicsnames <- as.includegraphics(graphnames[1:12], wd=".")
> dim(graphicsnames) <- c(4,3)
> graphicsnames[1:2, 1]
[1] "\\includegraphics[height=1em]{./irisBW001.pdf}"
[2] "\\includegraphics[height=1em]{./irisBW002.pdf}"
```

These values are placed into an ordinary matrix or dataframe and sent to the `Hmisc::latex` function to create a latex file fragment that can be input with the  $\text{\LaTeX}$  `\input` macro.

```
BWMS.latex <- Hmisc::latex(graphicsnames)
BWMS.latex$style <- "graphicx"
```

## 8 More Information on Microplots

The **microplot** package shows simple examples with **lattice**, **ggplot2**, and **base** graphics.

The **microplot** package shows simple examples in L<sup>A</sup>T<sub>E</sub>X using the R packages **Sweave**, **knitr**, and **rmarkdown**, and the Emacs package **org-mode**.

The **microplot** package shows simple examples in HTML using the R package **rmarkdown** and the Emacs package **org-mode**.

```
utils::install.packages("microplot", dependencies=TRUE)
## this includes HH and its dependencies
```

The **HH** package is designed to accompany  
*Statistical Analysis and Data Display, Second Edition*

Richard M. Heiberger and Burt Holland

Springer 2015

<http://www.springer.com/us/book/9781493921218>

