

# Getting Started with Lattice

immediate

July 29, 2010

## Contents

<a href="#">1</a>	<a href="#">Introduction</a>	<a href="#">1</a>
<a href="#">2</a>	<a href="#">Basic ideas</a>	<a href="#">2</a>
<a href="#">3</a>	<a href="#">Scatter plot (<code>xyplot</code>)</a>	<a href="#">3</a>
<a href="#">4</a>	<a href="#">Box and whisker plots (<code>bwplot</code>)</a>	<a href="#">4</a>
<a href="#">5</a>	<a href="#">Further Reading</a>	<a href="#">5</a>
<a href="#">6</a>	<a href="#">Session information</a>	<a href="#">7</a>

## List of Figures

<a href="#">1</a>	<a href="#">Concentration of Indomethicin over time produced using <code>xyplot</code></a>	<a href="#">5</a>
<a href="#">2</a>	<a href="#">Concentration of Indomethicin over time using <code>Subject</code> as the conditioning variable <code>xyplot</code></a>	<a href="#">6</a>
<a href="#">3</a>	<a href="#">Concentration vs Time of Indomethicin for 6 subjects using the Grouping variable <code>Subject</code>.</a>	<a href="#">6</a>
<a href="#">4</a>	<a href="#">Concentration vs Time of Indomethicin for 6 subjects using the Grouping variable <code>Subject</code>.</a>	<a href="#">7</a>

## 1 Introduction

The *lattice* package implements the Trellis graphics system and provides high-level functions for visualization of multivariate data. This lab covers some of the basics of the *lattice* package.

Although the *lattice* package is included in *R*, you must explicitly load the package to make use of the functions it provides:

```
> library("lattice")
```

Lattice provides several plotting functions, some of the commonly used ones are listed below.

Lattice functions	
<code>xyplot()</code>	Scatter plot
<code>dotplot()</code>	Cleveland dot plot
<code>bwplot()</code>	Box and whisker plot
<code>histogram()</code>	Histogram
<code>densityplot()</code>	Kernel density plot
<code>qq()</code>	Two sample quantile plot

Table 1: Commonly used high-level lattice functions

## 2 Basic ideas

A typical call to the lattice function `xyplot` is shown below.

```
> xyplot(y ~ x | c, data, groups = g)
```

The arguments to a lattice function can be summarized in terms of

1. lattice function: A lattice plotting function such as `xyplot`, `Rfunctiondotplot` etc.
2. formula: The first argument to a lattice method is a formula. The formula for our example is `y ~ x | c`. If the lattice method takes only a single vector as input, the formula can be expressed as `~ x | c`.
  - primary variables: Variables `y` (Y axis of the plot) and `x` (X axis of the plot) that defines the lattice display separated by the `~` character.
  - conditioning variable: Variable `c` in the example separated from the primary variables by the character `|`. The conditioning variable divides the plot into separate panels.
3. grouping variable: The variable `g` in the example. The grouping variable segregates data into subgroups within each panel.
4. data: A *data.frame* with column names corresponding to the variables `y`, `x`, `c` and `g`.

Lattice functions do not, in fact, generate a visual display. They return an object of class *trellis* which will generate the desired display when printed using the `print` function.

### 3 Scatter plot (`xyplot`)

We will use the `Indometh` data that contains plasma concentrations of the drug indomethacin for six subjects over a period of eight hours to illustrate a few aspects of *lattice*. First, we will use `xyplot` to produce a scatterplot of concentration of the drug over time. (Figure 1)

```
> data(Indometh)
> df <- Indometh
> head(df)
```

	Subject	time	conc
1	1	0.25	1.50
2	1	0.50	0.94
3	1	0.75	0.78
4	1	1.00	0.48
5	1	1.25	0.37
6	1	2.00	0.19

#### Exercise 1

Reproduce the concentration vs time plot from Figure 1 by following the steps described below.

1. Create a scatter plot using `xyplot`. Use the formula `conc ~ time` with `df` as the data argument.
2. Change the X-axis label to 'Time (hours)' by adding an `xlab` argument to the `xyplot` call.
3. Similarly, change the Y-axis label to 'Concentration (mcg/ml)' using `ylab`.
4. Finally, change the title of the plot by adding a `main` argument.

#### Solution:

```
> myplot <- xyplot(conc ~ time, data = df,
+                 xlab = "Time (hours)",
+                 ylab = "Concentration (mcg/ml)",
+                 main = "Pharmacokinetics of Indomethacin")
```

A scatter plot with the concentration profile for each patient appearing in a separate panel can be seen in Figure 2. This plot was produced by making use of `Subject` as the conditioning variable.

#### Exercise 2

1. Create a scatter plot by adding the conditioning variable `Subject` to the formula from Exercise 1.

2. Modify the conditioning variable in the formula to `factor(Subject, levels = 1:6)` and observe the difference in the order of plots.

**Solution:**

```
> myplot <- xyplot(conc ~ time | factor(Subject, levels = 1:6),  
+                  data = df, main = "Pharmacokinetics of Indomethicin",  
+                  ylab = "Concentration (mcg/ml)",  
+                  xlab = "Time (hours)")
```

A scatter plot produced by superimposing the concentration profiles from all the subjects to a single panel can be seen in Figure 3. The figure makes use of lines instead of points and also has a legend to help distinguish the concentration profiles of different subjects.

### Exercise 3

1. Create a single scatter plot of the superimposed concentrations by making use of formula `conc ~ time` and the additional argument `groups = Subject` to the `xyplot` function.
2. Add an additional argument `type='l'` to the function and observe what effect it has on the plot.
3. Add an additional argument `auto.key = list(space = "right")` to the `xyplot` to add a legend to the right of the plot.

**Solution:**

```
> myplot <- xyplot(conc ~ time, data = df, groups = Subject, type='l',  
+                  auto.key = list(space = "right"),  
+                  main = "Pharmacokinetics of Indomethicin",  
+                  ylab = "Concentration (mcg/ml)",  
+                  xlab = "Time (hours)")
```

## 4 Box and whisker plots (`bwplot`)

We are interested in finding out differences in the plasma concentration of Indomethicin amongst the six subjects. A box and whisker plot of the concentration of Indomethicin produced using the `bwplot` method can be observed in Figure 4

### Exercise 4

1. Create a boxplot using the formula `~ conc` and the lattice function `bwplot`.
2. Update the formula to `~ conc | Subject` in order to add a conditioning variable to the boxplot.

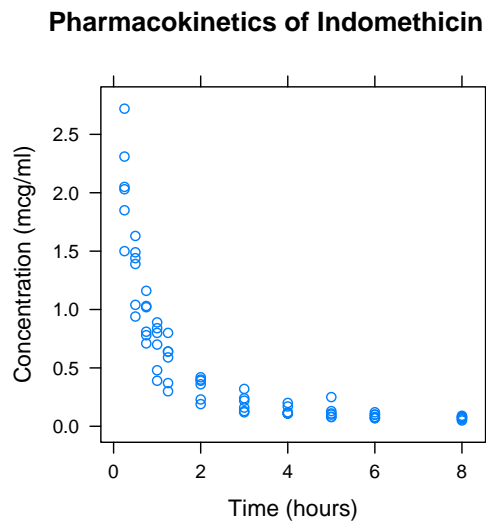


Figure 1: Concentration of Indomethicin over time produced using `xyplot`

3. Add an additional argument `layout=c(6,1)` to the `bwplot` and observe its effect on the plot.

**Solution:**

```
> myplot <- bwplot(~ conc | Subject, data = df, layout = c(1,6),
+                  main = "Concentration of Indomethicin",
+                  xlab = "Concentration (mcg/ml)"
+                  )
```

## 5 Further Reading

- The *lattice* package includes many detailed help pages. You can get an overview using the *R* command `help(package="lattice")`.
- A good source of several lattice plots along with the code that produced them is <http://lmdvr.r-forge.r-project.org>.
- *Multivariate Data Visualization with R* by Deepayan Sarkar the author of *lattice*. See <http://lmdvr.r-forge.r-project.org/>.

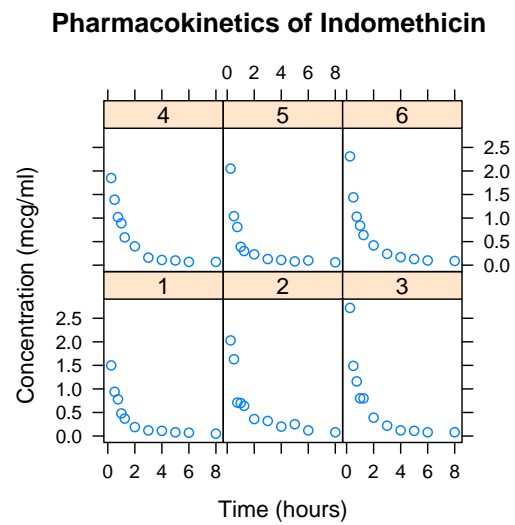


Figure 2: Concentration of Indomethicin over time using `Subject` as the conditioning variable `xypLOT`

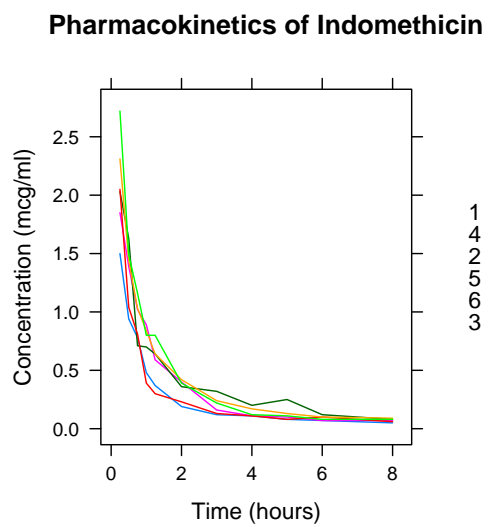


Figure 3: Concentration vs Time of Indomethicin for 6 subjects using the Grouping variable `Subject`.

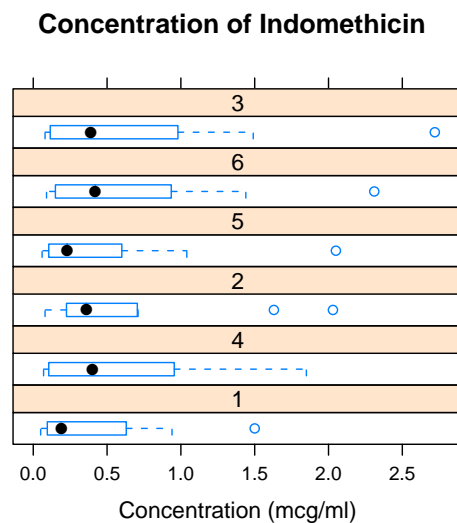


Figure 4: Concentration vs Time of Indomethicin for 6 subjects using the Grouping variable `Subject`.

## 6 Session information

- R version 2.11.1 (2010-05-31), x86\_64-unknown-linux-gnu
- Locale: LC\_CTYPE=C, LC\_NUMERIC=C, LC\_TIME=C, LC\_COLLATE=C, LC\_MONETARY=C, LC\_MESSAGES=en\_US, LC\_PAPER=en\_US, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US, LC\_IDENTIFICATION=C
- Base packages: base, datasets, grDevices, graphics, methods, stats, tools, utils
- Other packages: lattice 0.18-8
- Loaded via a namespace (and not attached): grid 2.11.1