

This graph is very similar to the ones produced using the other graphics packages but has the distinctive background and legend style that is used as the default option in **ggplot2**.

GMR-2010-001: Creating Cleveland Dotplots in R

©2010 GM-RAM Limited

This leaflet is part of a series covering Statistical Analysis using the R Statistical Software.

<http://www.gm-ram.com>

<http://www.wekaleamstudios.co.uk>

Creating Cleveland Dotplots in R

Ralph Mansson

Introduction

Dotplots

To illustrate the Cleveland dot plot we consider data on area equipped for irrigation in four regions of the World. The R code to create the data object is shown here:

```
irrigation.df = data.frame(Region
= rep(c("Africa", "Latin America",
"North America", "Europe"), 4), Year =
factor(c(rep(1980, 4), rep(1990, 4),
rep(2000, 4), rep(2007, 4))), Area = c(9.3,
12.7, 21.2, 18.8, 11.0, 15.5, 21.6, 25.3,
13.2, 17.3, 23.3, 26.7, 13.6, 17.3, 23.8,
26.3))
```

Base Graphics

In the base graphics system we can build up the **dot-plot** step by step. The first function call creates the graph region based on the data set but we do not plot any data by setting the **type = "n"** option. The axis labels for the horizontal and vertical scales are set along with the title:

```
plot(irrigation.df$Area,
irrigation.df$Region, xlab = "Area",
ylab = "Region", main = "Irrigation Area by
Region", type = "n")
```

To add the points with separate colours for each of the four years we use the **points** function and subset to the particular year by testing a condition on the year.

The col argument is used with a text string to specify the colour for the symbols for the given year:

```
points(irrigation.df$Area[irrigation.df$Year == 1980],
       irrigation.df$Region[irrigation.df$Year == 1980], col = "black", pch = 16)
```

```
points(irrigation.df$Area[irrigation.df$Year == 1990],
       irrigation.df$Region[irrigation.df$Year == 1990], col = "blue", pch = 16)
```

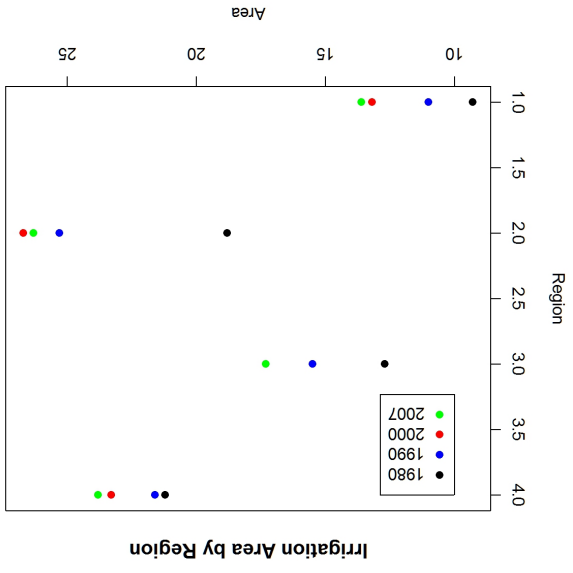
```
points(irrigation.df$Area[irrigation.df$Year == 2000],
       irrigation.df$Region[irrigation.df$Year == 2000], col = "red", pch = 16)
```

```
points(irrigation.df$Area[irrigation.df$Year == 2007],
       irrigation.df$Region[irrigation.df$Year == 2007], col = "green", pch = 16)
```

The code is rather long winded compared to the other two graphics packages. We can add a legend to the graph so that the years can be identified:

```
Legend(10, 4, legend = c("1980", "1990", "2000", "2007"), col = c("black", "blue", "red", "green"), pch = 16)
```

All the code produces the following graph:



The graph is basic but we can consider the changes over time for the four regions. One downside is that the regions have been labelled with numbers rather than text strings.

Lattice Graphics

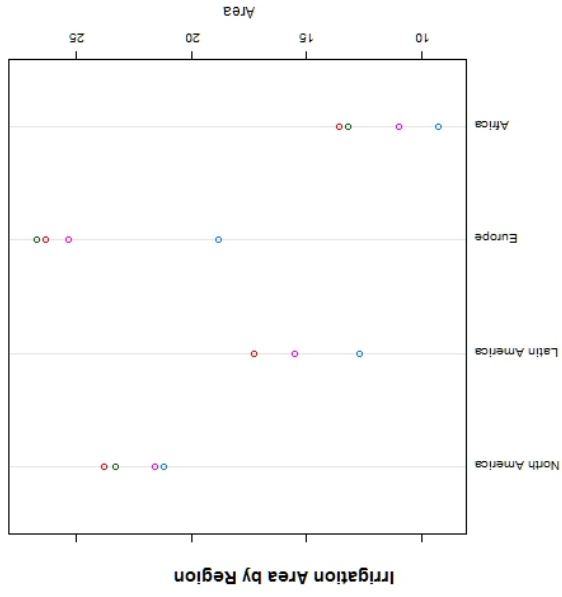
The **lattice** graphics package has a function **dotplot** that is used to create dot plots. The first argument to the function is a formula describing the variables to use for the horizontal and vertical axes. We also specify the data frame to use for the graph and which column to determine different symbols and/or colours to highlight groupings within the plot

```
dotplot(Region ~ Area, data = irrigation.df,
        groups = Year, main = "Irrigation Area by Region")
```

The lattice variant of the graph is shown here:

ggplot2 Graphics

The graph is simple and very similar to the one produced using the base graphics with the advantage that the R code is not as complicated.



The **ggplot** function is used to create the dot plot where we first specify the name of the data frame with the information to be displayed and then use the **aes** argument to list the variables to plot on the horizontal and vertical axes. The colour argument determines the variable to use for assigning colours to (usually) a categorical variable.

```
ggplot(irrigation.df, aes(x = Area, y =
Region, colour = Year)) + geom_point() +
opts(title = "Irrigation Area by Region")
```

The **ggplot2** version of the graph is shown.