

GMR-2009-003: Creating Barcharts in R
©2009 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 Barcharts in R

Ralph Mansson

Introduction

The three useful approaches to creating graphs using R are the **base** graphics, **lattice** graphics package and the **ggplot2** package. The graphs produced by the three approaches have distinctive appearances, which can be customised to various degrees and with differing levels of complexity.

Barcharts

A bar graph is a type of display that can be used to compare counts, frequencies, totals or other summary measures for a series of categories, e.g. different market sectors, quarters in a financial year. The bar graph can be layout with the categories either on the vertical or horizontal axis of the display - depending on whether we consider making a vertical or horizontal comparison is easier for interpreting the graph.

To illustrate creating a bar chart using various graphics packages we will make use of some data from the FAO statistics website (<http://faostat.fao.org>) for the UK in 2007. The data is on production (in metric tonnes) of five food and agricultural commodities.

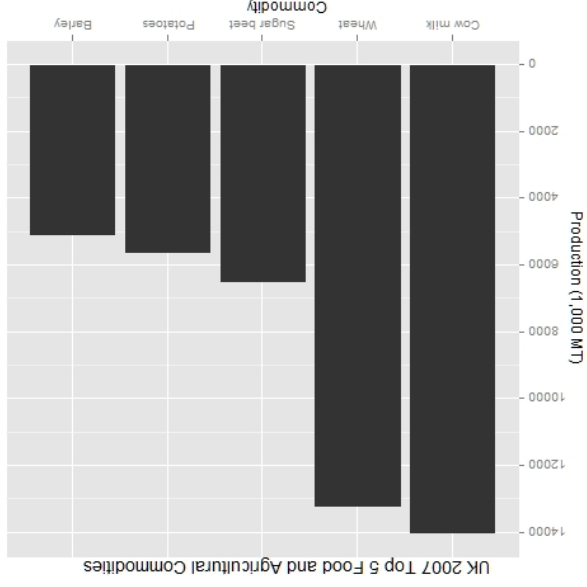
The first step is to prepare the data that will be displayed in the graph. As the dataset is small for this example, using only the top five commodities we can manually create the data object. To make the graph less cluttered the production is recorded as 1,000s of metric tonnes. The R code to create the data object is shown here:

ggplot2 Graphics

The create the bar chart in the **ggplot2** package we use the **ggplot** function to specify the data to appear in the graph then gradually add in the other components of the graph. We specify the data frame where the data is stored and then use the **aes** argument for the **geom_bar()** is used to create a bar chart display and the last three options in the example here are for creating the various labels.

```
ggplot(uk2007, aes(Commodity, Production)) +
  geom_bar() + xlab("Commodity") +
  ylab("Production (1,000 MT)") + opts(title = "UK 2007 Top 5 Food and Agricultural Commodities")
```

This code produces the following graph:



The layout of this graph differs mainly with the grid background layout, which by default is a gray with white lines.

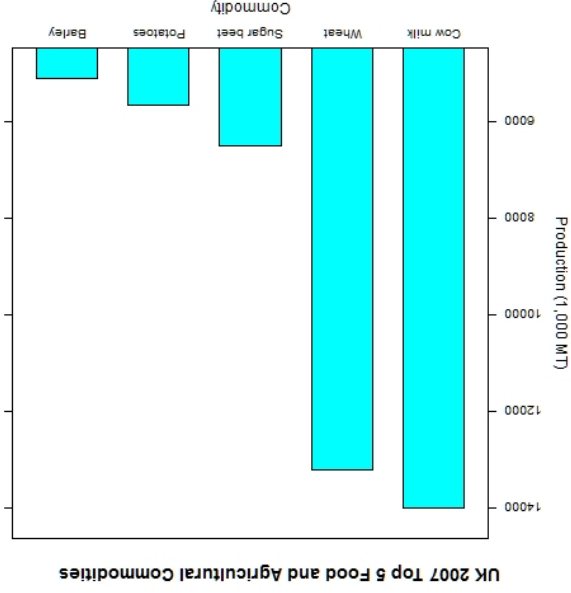
This graph is visually appealing with sensible space between the bars for the five commodity categories.

Lattice Graphics

In the **lattice** graphics package the **barchart** function is used to create bar charts. The x and y variables are specified using a formula, which is the standard way for Trellis graphics. The variable on the vertical axis is specified on the left hand side of the formula and the variable for the horizontal axis is on the right hand side.

```
barchart(Production ~ Commodity, data =
  uk2007, xlab = "Commodity", ylab =
  "Production (1,000 MT)", main = "UK 2007 Top
  5 Food and Agricultural Commodities")
```

This code produces the following graph:



The main visual difference is the default colours for the bars which is much brighter than the base graphics example. There is also a large gap between the bars.

Base Graphics

```
uk2007 = data.frame(Commodity = factor(c("Cow
  milk", "Wheat", "Sugar beet", "Potatoes",
  "Barley"), levels = c("Cow milk", "Wheat",
  "Sugar beet", "Potatoes", "Barley")),
  Production = c(14023, 13221, 6500, 5635,
  5079))
```

The base graphics in R provide a function **barplot** that we can use to create a bar chart. The first argument to the function is the name of the object with the values for the bar chart. The names argument is used to provide the labels for the categories in the graph. We also specify the labels for the x-axis, y-axis and title of the graph with the **xlab**, **ylab** and **main** arguments respectively.

```
barplot(uk2007$Production, names =
  uk2007$Commodity, xlab = "Commodity", ylab
  = "Production (1,000 MT)", main = "UK 2007 Top
  5 Food and Agricultural Commodities")
```

This code produces the following graph:

