

Chapter 6

Plotting and graphics in R

6.1 Plotting data

First, load some data:

```
> data(mtcars)
> attach(mtcars)
```

The following object(s) are masked from `mtcars` (position 3) :

```
am carb cyl disp drat gear hp mpg qsec vs wt
```

```
> names(mtcars)
```

```
[1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am"
[10] "gear" "carb"
```

To plot one variable against another, use

```
> plot(disp, mpg)
```

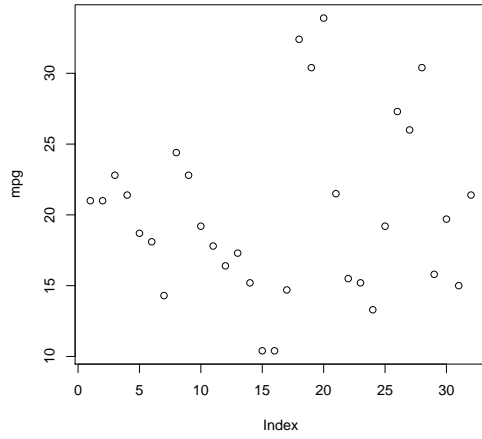
or, using formula notation,

```
> plot(mpg ~ disp)
```

By default, R plots using points as this is usually what is needed for plotting data.

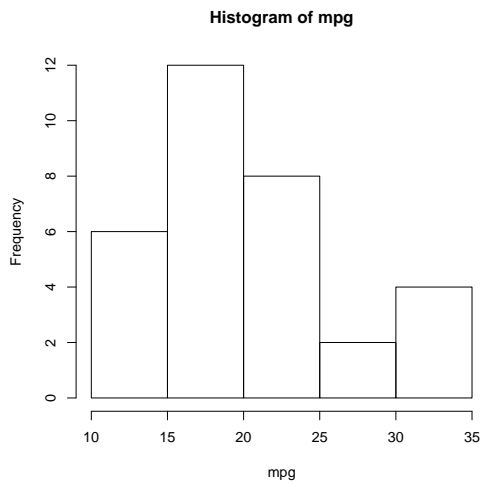
Individual variables can be plotted also:

```
> plot(mpg)
```

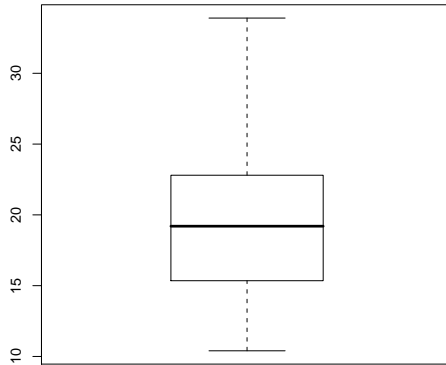


But that's not very useful. Perhaps better is one of the following

```
> hist(mpg)
```



```
> boxplot(mpg)
```



6.2 Plotting functions

- Since R is a vector-based language, plots of functions are easily produced.

```
> x <- seq(-3, 3, by = 0.05)
> y <- x^2
> plot(x, y)
```

This produces the plot in Figure 6.1. (The function `seq` is explained in Section 3.3.1). Since R is a statistical package, points are the default plot (since it is most useful for plotting data). Lines can be specified by

`plot(x, y, type="l")` where the type is the letter ‘ell’ for lines.

- R can produce publication-quality graphics; see Figure 6.2 for a variation of Figure 6.1. You do not need to know how to produce this plot; but it is important to know that high-quality plots are possible.

And don’t forget:

```
> detach(mtcars)
```

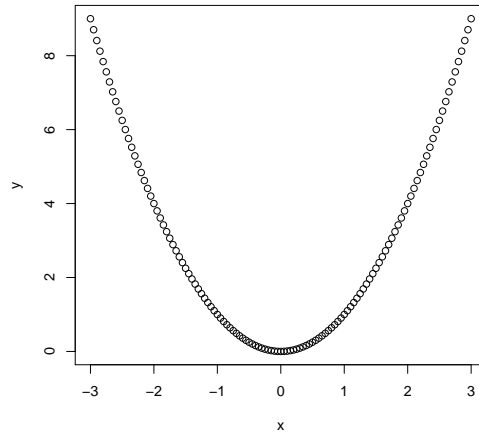
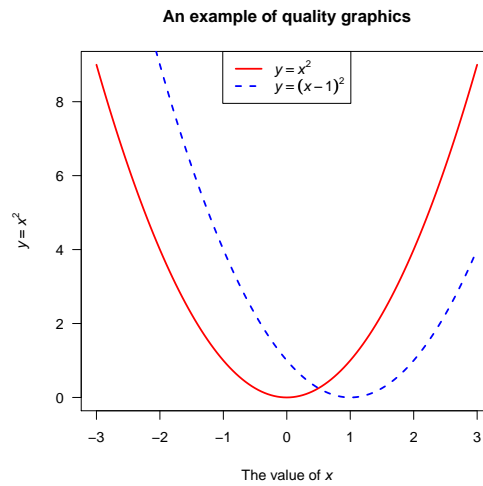
Figure 6.1: An example plot: $y = x^2$.

Figure 6.2: A variation of Figure 6.2