

# Plotting data and functions in R

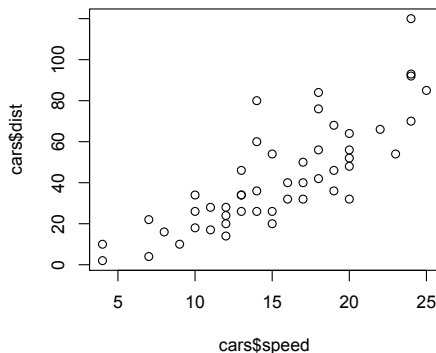
## Scatter plots and their fit functions

`plot()` is the general data plotting function. If we have data in an object `d` and variables `d$y` and `d$x`, we can make a scatterplot of them with:

```
plot( d$y ~ d$x )
```

Consider an example from the built-in datasets.

```
data(cars)
plot( cars$dist ~ cars$speed )
```

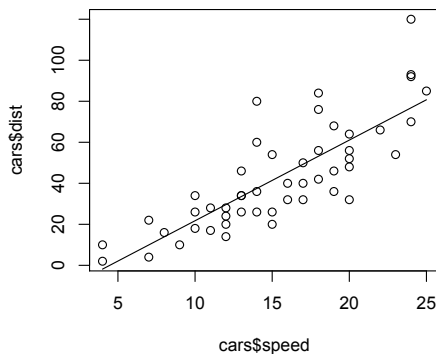


We want to fit a regression to these data. Here's a simple linear (OLS) regression:

```
m <- lm( cars$dist ~ cars$speed )
```

This model has two parameters that define the regression line. We can plot the line by extracting them and using `curve()` to draw the line.

```
k <- coef(m)
curve( k[1] + k[2]*x , from=min(cars$speed) ,
       to=max(cars$speed) , add=TRUE )
```



The `curve()` function has a simple structure:

```
curve( function , from=minx , to=maxx ,
       add=TRUE )
```

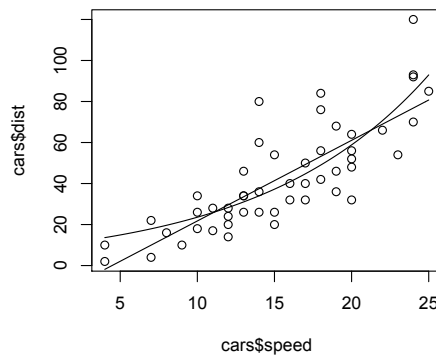
Replace `function` with the mathematical expression for your regression, whether it is linear or not. Make the x-axis variable the symbol `x`. This symbol is replaced with the sequence of values from `minx` to `maxx`, drawing the curve (or line, in this case). The `add=TRUE` parameter tells R to overlay the curve on the existing plot, rather than drawing a new plot.

Now let's fit an exponential function to these same data, and overlay that regression as well.

```
library(bbmle)
m2 <- mle2( cars$dist ~ dnorm( exp(a +
                               b*cars$speed) , exp(s) ) ,
           start=list(a=0,b=0,s=2) )
```

And then plot the fit:

```
k <- coef(m2)
curve( exp( k["a"] + k["b"]*x ) ,
       from=min(cars$speed) , to=max(cars$speed) ,
       add=TRUE )
```



This procedure works for nearly anything you want to plot. But if you have other prediction variables (more than one `x`), you'll need to fix the value of all but one, when you plot. A good choice is the mean value for each of the variables you aren't plotting on the x-axis.