reveals that the predicted optical density associated with `carb = 0.8` is 0.706. Standard errors can also be calculated by `predict` if a confidence interval, for example, is required.

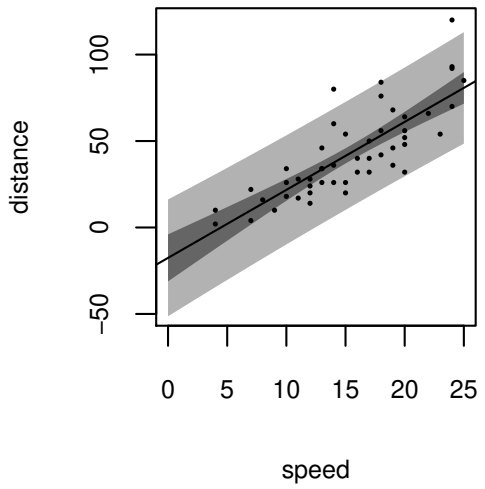
Occasions arise in which it is appropriate to force a regression line to pass through the origin. Recall that the intercept of the regression line for the `Formaldehyde` data pairs was nearly zero. Furthermore, there might be evidence from the experiment that when `carb` is zero, then `optden` is also zero. This would be a case in which forcing the regression line to pass through the origin would be appropriate. The slight modification to the `formula` argument in `lm`

```
> fit = lm(optden ~ carb + 0, data = Formaldehyde)
```

forces the regression line to pass through the origin. Typing the command `fit$coefficients` reveals that the estimated slope of the regression line is 0.884 (versus 0.876 for a full model).

The `Formaldehyde` data set provided a nearly-perfect fit using simple linear regression. This is seldom the case in practice; usually there is significant scatter to the data pairs. Consider the built-in `cars` data set, which contains the speed (in miles per hour) and stopping distance (in feet) for $n = 50$ cars recorded during the 1920s. The following R code plots the fitted regression line with estimated intercept $\hat{\beta}_0 = -17.58$ and estimated slope $\hat{\beta}_1 = 3.93$ via the `lm` and `plot` functions, along with the 95% confidence interval (darker gray) and 95% prediction interval (lighter gray) for all speed values between 0 and 25 miles per hour via the `predict` and `polygon` functions.

```
> x   = cars$speed
> y   = cars$dist
> fit = lm(y ~ x)
> plot(NULL, xlim = c(0, 25), ylim = c(-50, 120), xlab = "speed", ylab = "distance")
> x1  = 0:25
> y1  = predict(fit, data.frame(x = x1), interval = "prediction")
> polygon(c(x1, rev(x1)), c(y1[ , 2], rev(y1[ , 3])), col = "gray70", border = NA)
> y2  = predict(fit, data.frame(x = x1), interval = "confidence")
> polygon(c(x1, rev(x1)), c(y2[ , 2], rev(y2[ , 3])), col = "gray40", border = NA)
> abline(fit$coefficients)
> points(x, y, pch = 16)
```



As expected, both interval estimators are narrowest at the sample mean of the speed values. The width of these intervals is a function of α (smaller α values result in wider intervals), the number of