

# Case Study 4: Visualising linear model fits

*Richard Wilkinson*

Visualising the data can help us decide what models we wish to fit. Similarly, once we have fitted a model, it can be useful to visualise it. This may suggest modifications to the model. It is also extremely useful in helping us to understand the implications of the fitted model.

We will use a new R package called visreg. See <http://myweb.uiowa.edu/pbreheny/publications/visreg.pdf> for more details. The first time you use visreg you will need to install it (if not using a University computer)

```
install.packages('visreg')
```

## The data

We will use data consisting of 154 measurements of daily air quality in New York. The data consist of 6 variables:

- Ozone: Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island
- Solar.R: Solar radiation in Langleys in the frequency band 4000–7700 Angstroms from 0800 to 1200 hours at Central Park
- Wind: Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport
- Temp: Maximum daily temperature in degrees Fahrenheit at La Guardia Airport.
- Month
- Day

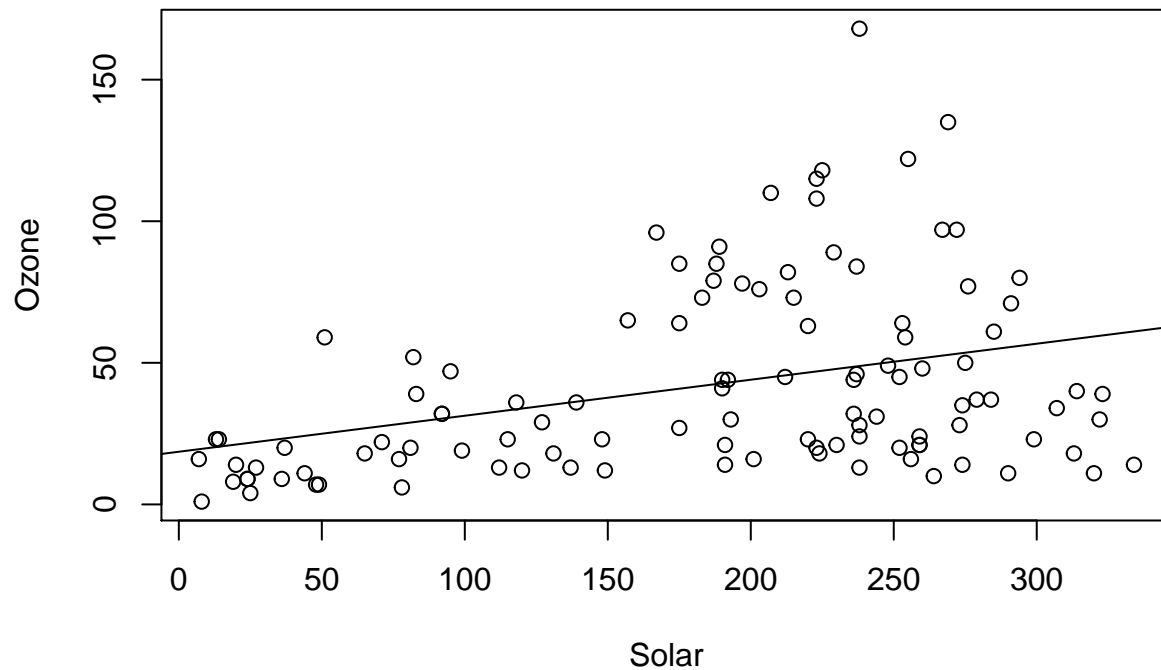
```
data(airquality)
str(airquality)
```

```
## 'data.frame':  153 obs. of  6 variables:
## $ Ozone   : int  41 36 12 18 NA 28 23 19 8 NA ...
## $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
## $ Wind    : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
## $ Temp    : int  67 72 74 62 56 66 65 59 61 69 ...
## $ Month   : int  5 5 5 5 5 5 5 5 5 5 ...
## $ Day     : int  1 2 3 4 5 6 7 8 9 10 ...
```

## Models

Visualising simple linear regression models (i.e. those of the form  $y = a + bx + e$ ) is easy

```
fit0 <- lm(Ozone ~ Solar.R, data=airquality)
plot(airquality$Solar.R, airquality$Ozone, xlab='Solar', ylab='Ozone')
abline(fit0)
```

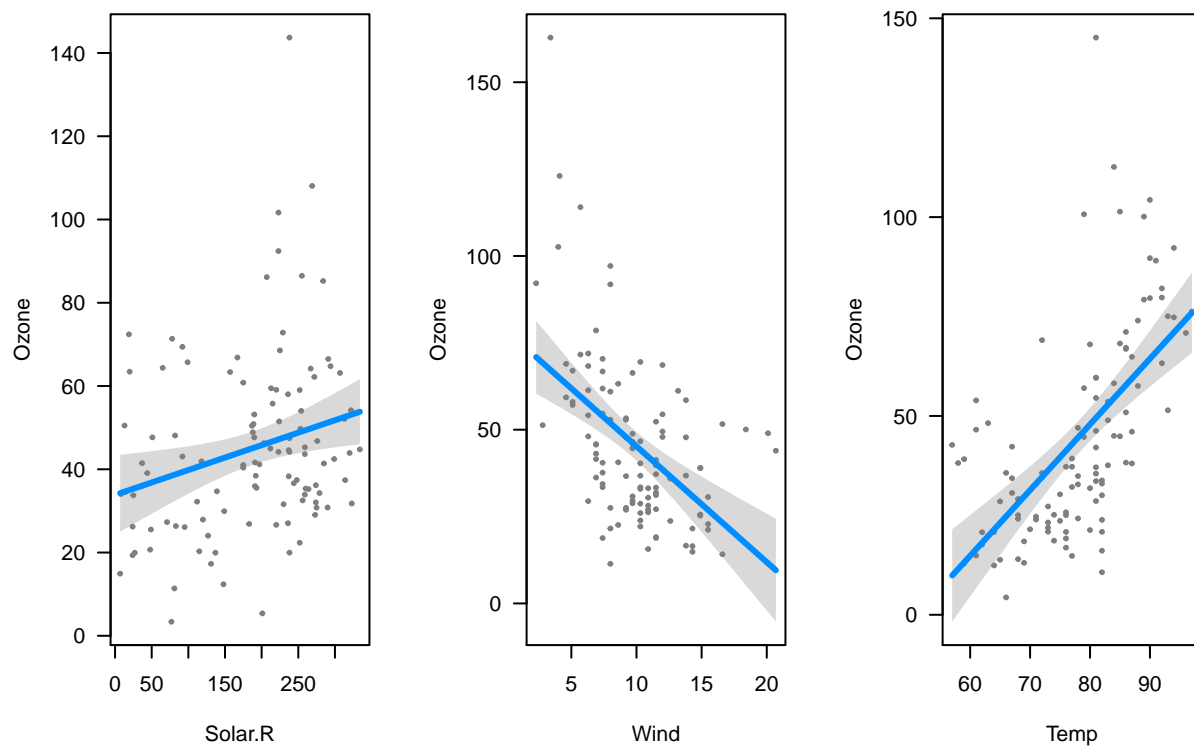


But what do we do if we are doing multiple linear regression where we have several independent variables? Suppose we fit the model

$$\text{Ozone} = a + b \times \text{Solar} + c \times \text{Wind} + d \times \text{Temp} + e$$

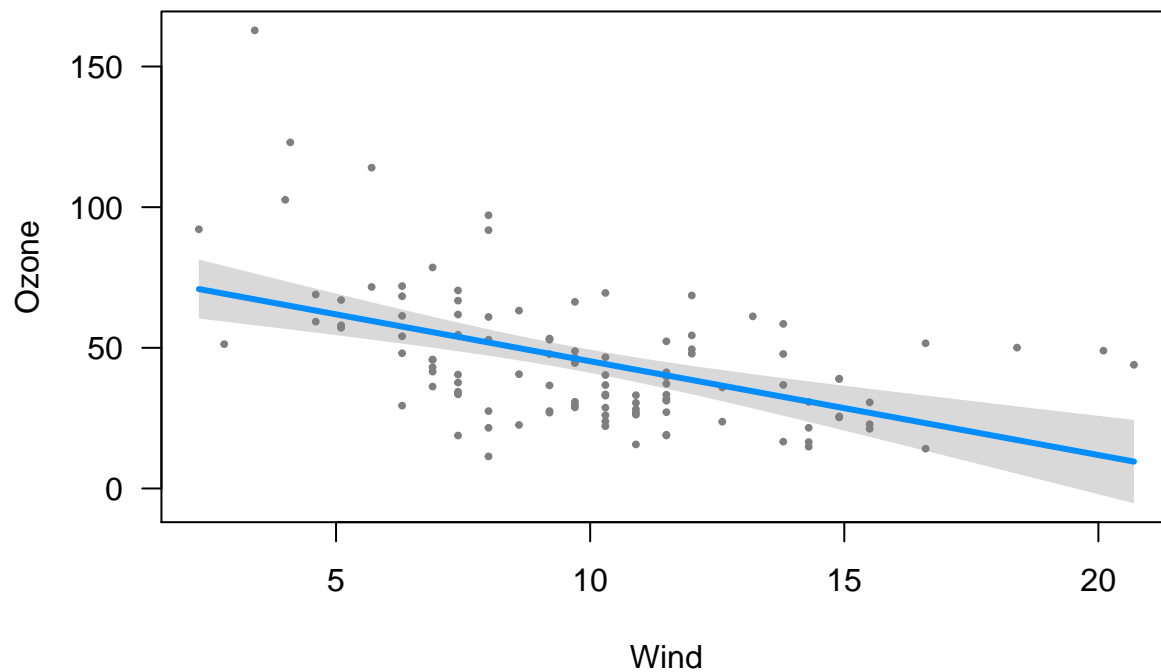
The default solution in visreg is to plot the effect of changing each x-variable one at a time, holding all the other variables constant (at their median value by default, or the most common category for discrete variables).

```
fit <- lm(Ozone ~ Solar.R+ Wind + Temp, data=airquality)
library(visreg)
par(mfrow=c(1,3))
visreg(fit)
```



Or you can choose to plot for just a single one of the covariates

```
visreg(fit, "Wind")
```



This allows us to easily see the effect of the wind on ozone, and identify possible outliers.

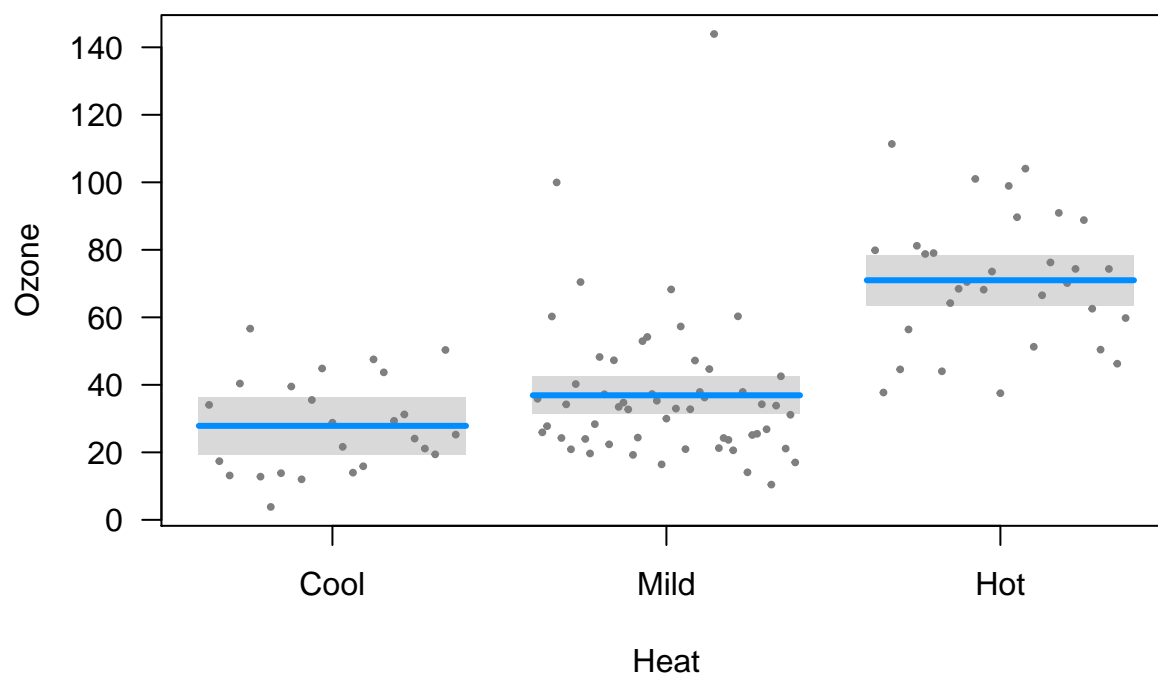
## Categorical variables

Lets create a categorical variable called heat, by dividing days into cool, mild and hot categories:

```
airquality$Heat <- cut(airquality$Temp, 3, labels=c("Cool", "Mild", "Hot"))
```

If we now fit the model, we get

```
fit.heat <- lm(Ozone ~ Solar.R + Wind + Heat, data = airquality)
visreg(fit.heat, "Heat", type = "conditional")
```



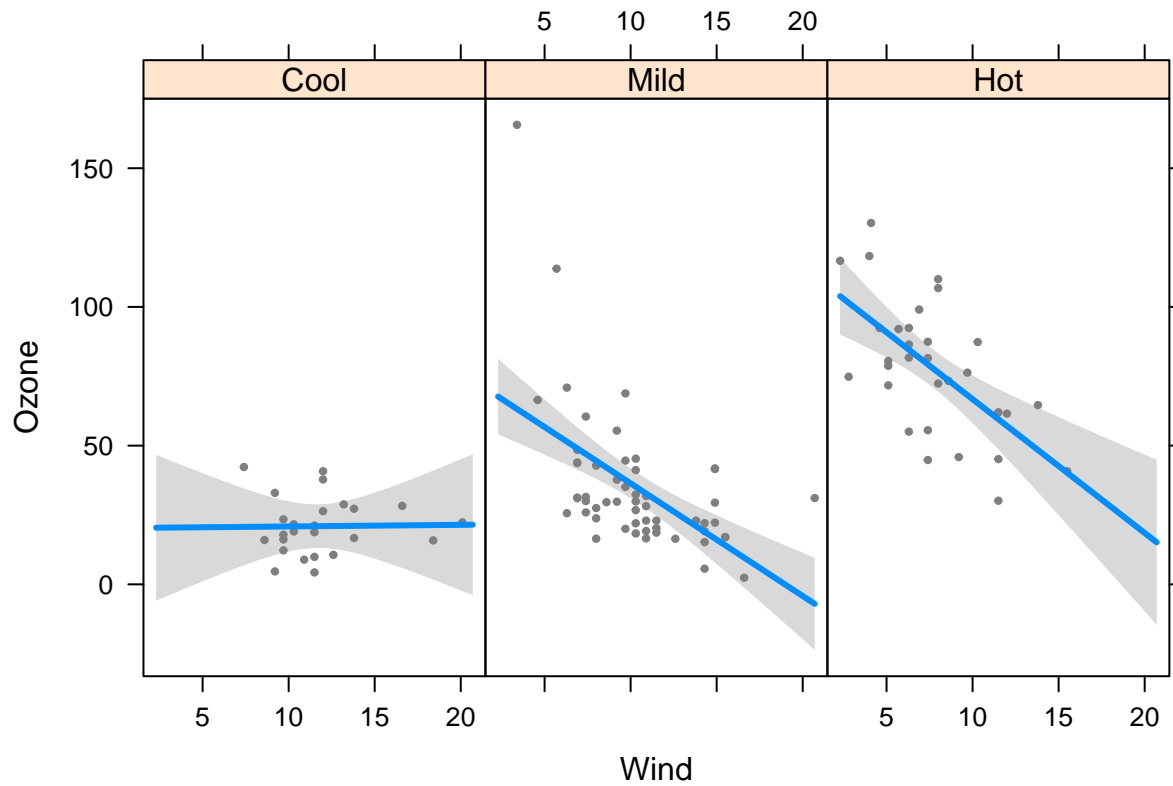
which clearly suggests that different levels for different heats is sensible. We could use an ANOVA hypothesis test to confirm this.

## Interactions

We can also look at interactions. Lets fit the model

$$\text{Ozone} = \begin{cases} \alpha + \beta \times \text{Wind} + e & \text{if Heat="cool"} \\ \alpha + \alpha_m + (\beta + \beta_m) \times \text{Wind} + e & \text{if Heat="mild"} \\ \alpha + \alpha_h + (\beta + \beta_h) \times \text{Wind} + e & \text{if Heat="hot"} \end{cases}$$

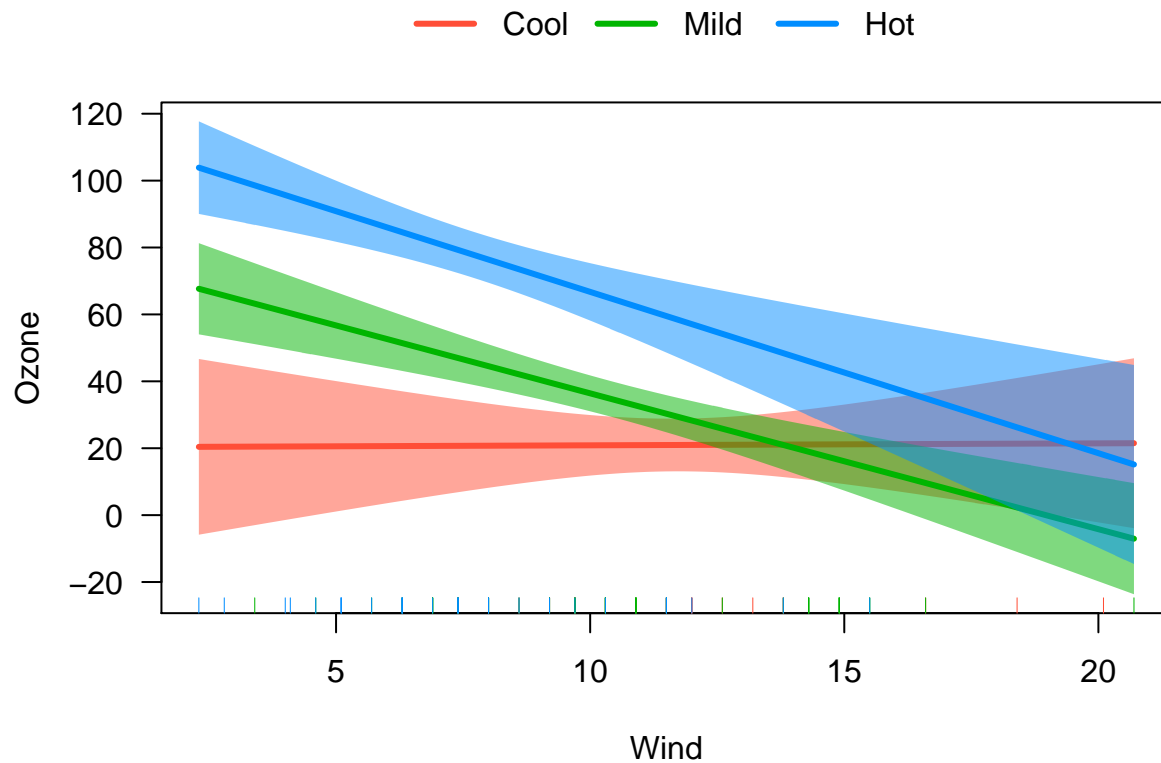
```
fit <- lm(Ozone ~ Solar.R + Wind * Heat, data = airquality)
visreg(fit, "Wind", by = "Heat")
```



We can clearly see that the relationship between wind and ozone becomes more pronounced as the days get hotter. The splitting of the data into cool, mild and hot days also allows us to see that there are no cool days with low winds, and relatively few hot days with high winds.

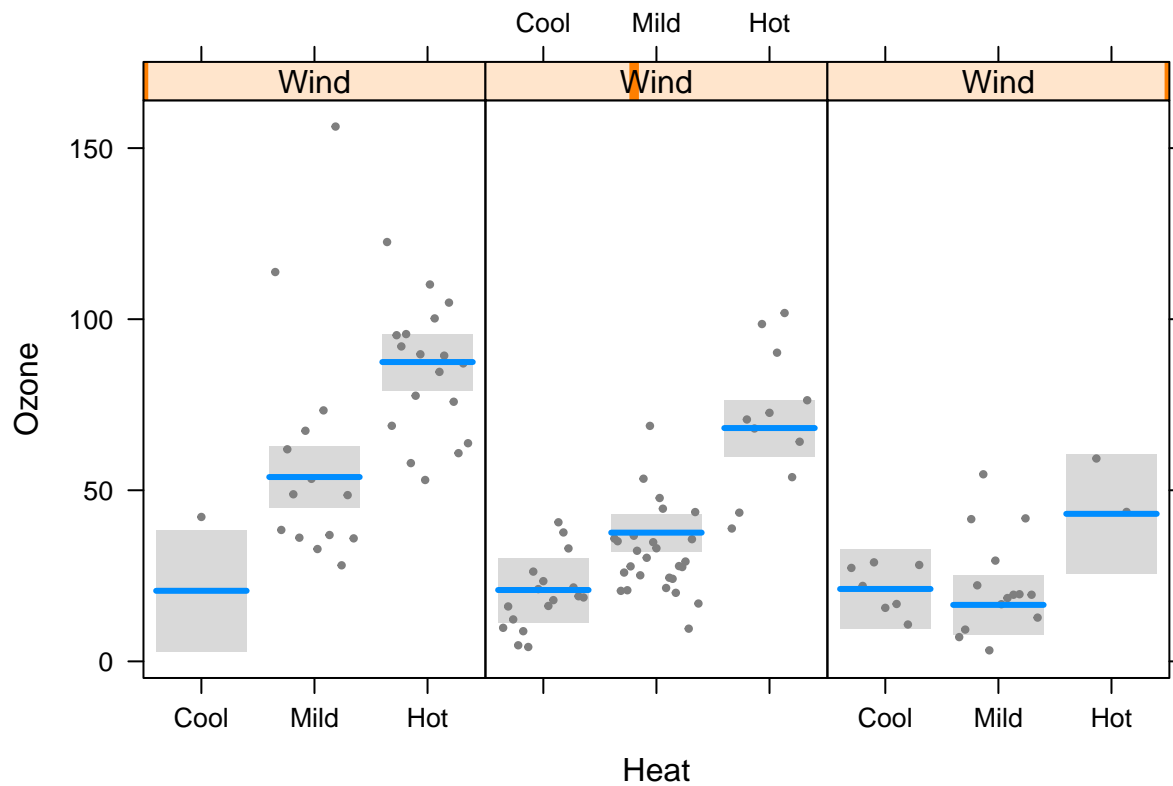
We can overlay these plots to allow for more direct comparison

```
visreg(fit, "Wind", by="Heat", overlay=TRUE, partial=FALSE)
```



If we split this by Wind, the emphasis is now on the effect of heat.

```
visreg(fit, "Heat", by = "Wind", breaks=c(5.7, 9.7, 14.9))
```



If the by variable is continuous (as here), visreg splits the data into parts. The default is to take cross sections

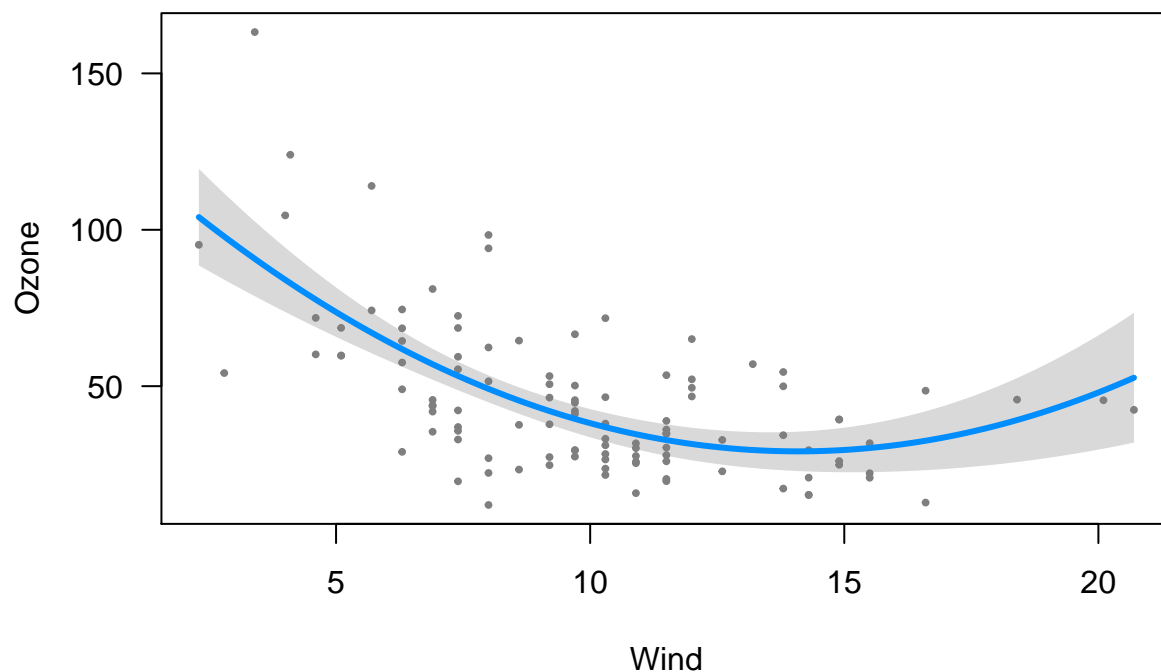
at the 10th, 50th and 90th percentile, but here I've chosen to split days into three categories. The values 5.7, 9.7 and 14.9 are the central wind speeds, and data points are assigned into which ever group they are closest to.

We can see that heat has a large effect on ozone concentration when the day is not windy, but only a small effect on ozone on windy days.

## Transformations

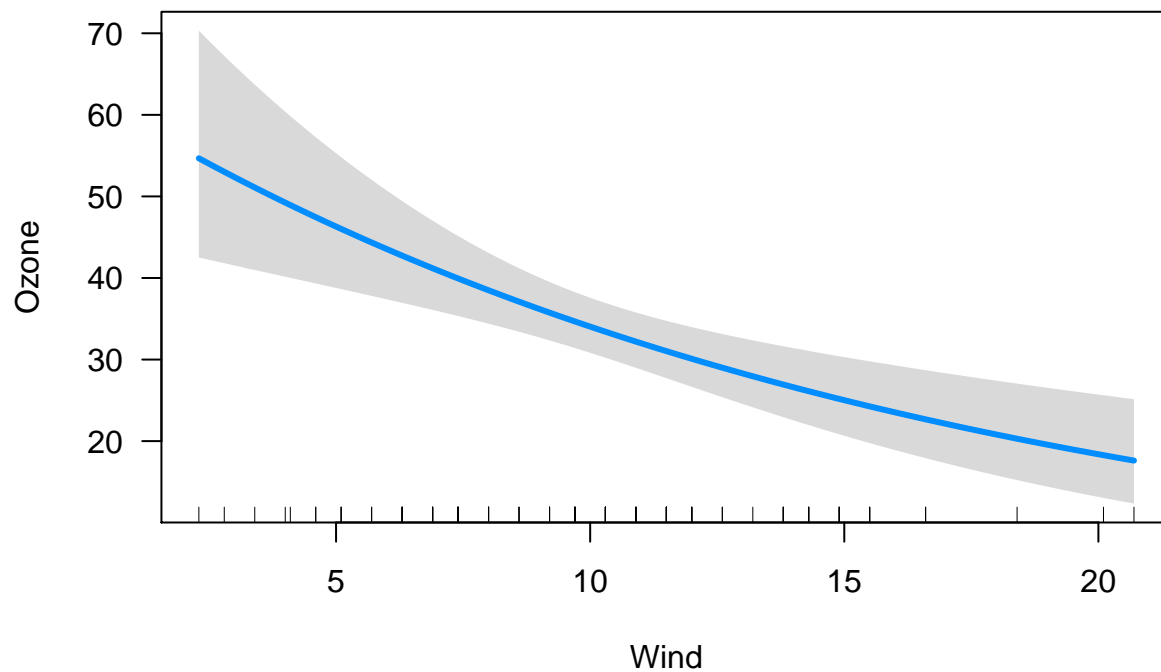
We can also deal with transformations easily

```
fit1 <- lm(Ozone ~ Solar.R + Wind + I(Wind^2) + Temp, data = airquality)
visreg(fit1, "Wind")
```

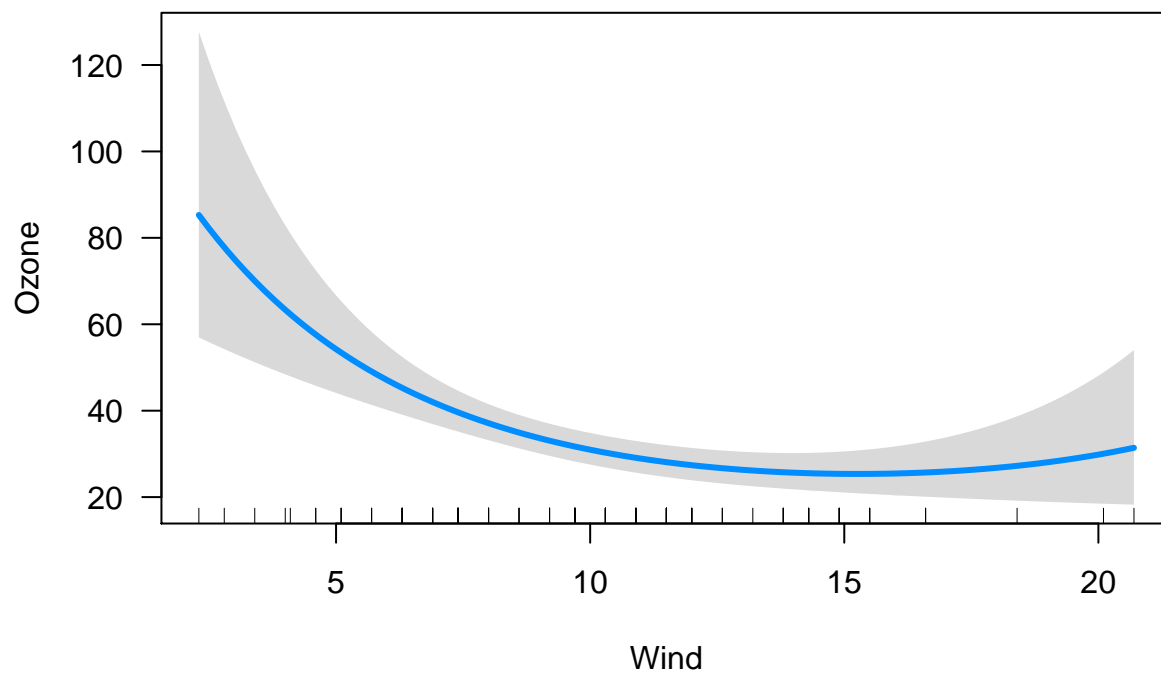


If we transform the y-variable, we need to provide visreg with the appropriate inverse transformation.

```
fit2 <- lm(log(Ozone) ~ Solar.R + Wind + Temp, data = airquality)
visreg(fit2, "Wind", trans = exp, ylab = "Ozone")
```



```
fit3 <- lm(log(Ozone) ~ Solar.R + Wind + I(Wind^2) + Temp, data = airquality)
visreg(fit3, "Wind", trans = exp, ylab = "Ozone")
```



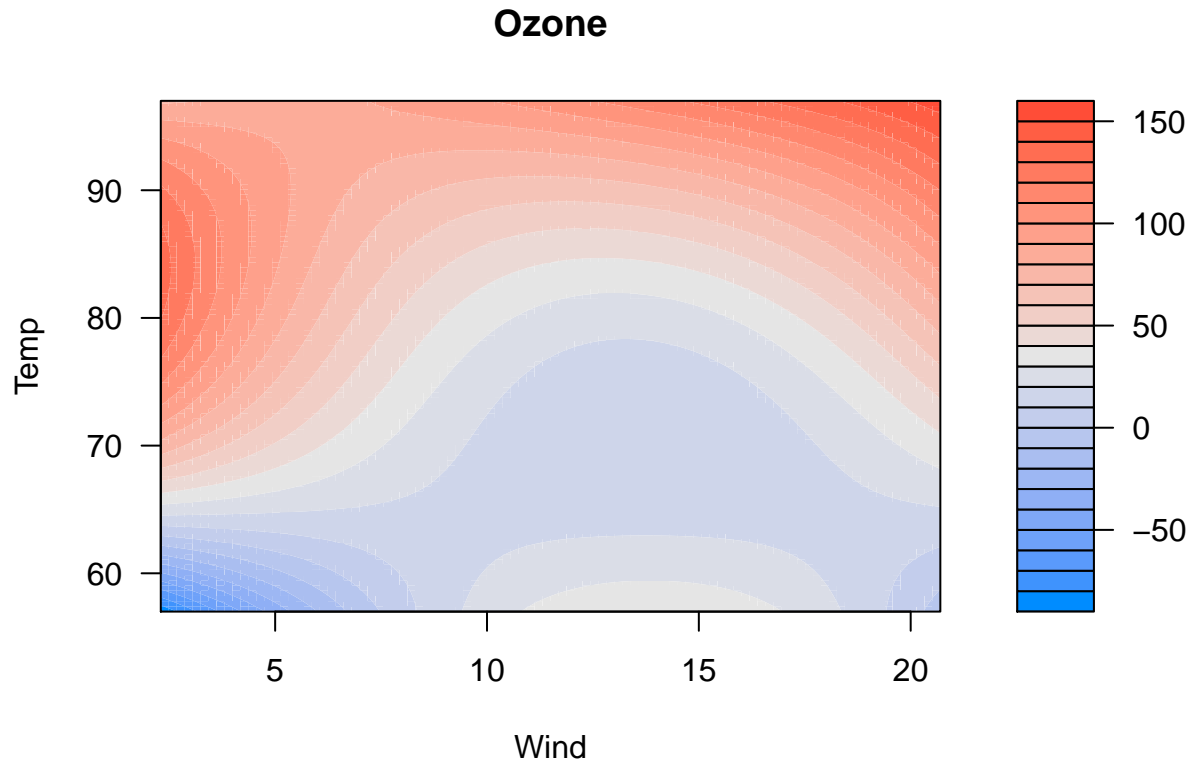
Visualisation is particularly important in models with transformations, as looking at the coefficients alone it can be difficult to determine the nature of the relationship when it is non-linear.

## Surface plots

It can also be useful to produce surface plots. I prefer contour plots for this. 3d plots are generally a bad idea on paper - but can be useful on a screen when they can be manipulated.



```
fit <- lm(Ozone ~ Solar.R + Wind + Temp + I(Wind^2) + I(Temp^2)
        + I(Wind * Temp) + I(Wind*Temp^2) + I(Temp*Wind^2)
        + I(Temp^2 * Wind^2), data = airquality)
visreg2d(fit, "Wind", "Temp", plot.type = "image")
```



## Summary

Visualising your fitted model can provide insights that you can miss when just looking at fitted coefficients. However, visual plots alone are not sufficient, and should be accompanied by numerical information as covered in other parts of the module.