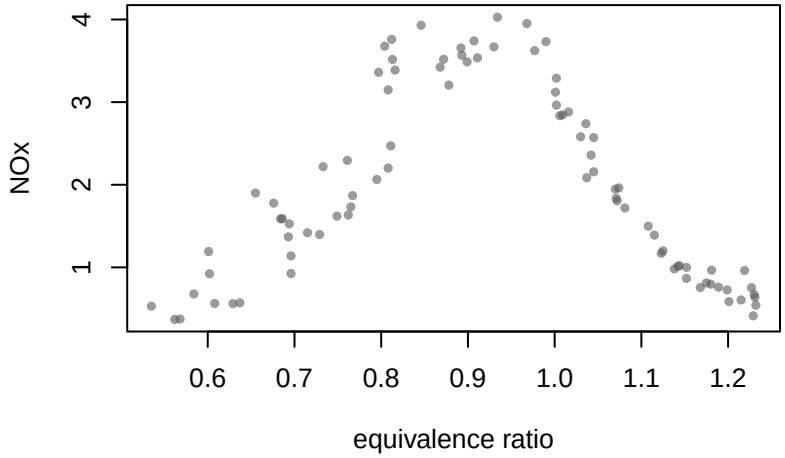
Specification activity

STAT3530 Applied Linear Models

Part 1: nonlinear specification

This part accompanies the script `13-nonlinear-specification.R`.

Data for this part are 88 runs of a single-cylinder test engine. `NOx` is nitrogen oxide emissions in micrograms per joule, and `E` is the equivalence ratio: the fuel-to-air ratio, scaled so that $E = 1$ is exactly enough air to burn all the fuel. Combustion runs hottest near $E = 1$, and NOx forms at high temperature.



The relationship is obviously nonlinear. Here you'll explore a few model specifications and your goal is to decide which is best.

1. Polynomial

A polynomial model of order k is:

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k + \epsilon_i$$

What order of polynomial seems best? The example below shows how to fit a quadratic model. Adjust the polynomial order k. You can use F tests to decide, but might also consider other methods of comparison.

```
fit.lin  <- lm(NOx ~ E, data = ethanol)
fit.quad <- lm(NOx ~ poly(E, 2), data = ethanol)

anova(fit.lin, fit.quad)
```

2. Broken stick

A piecewise linear model is:

$$Y_i = \gamma_0 + \gamma_1 x_i + \gamma_2 (x_i - t)_+ + \epsilon_i \qquad \text{where} \qquad (x_i - t)_+ = \max(x_i - t, 0)$$

Recall from lecture that γ_1 is the slope below the knot t and γ_2 is the *change* in slope at the knot.

First fit a knot at $E = 1$, where theory says NOx should peak, and then try $E = 0.9$, where the scatterplot looks like it turns. Which is better?

```
stick.00 <- lm(NOx ~ E + pmax(E - 1.00, 0), data = ethanol)

# slope below the knot, then slope above it
coef(stick.00)[2]
coef(stick.00)[2] + coef(stick.00)[3]
```

Two models with knots in different places are *not* nested, so no F test compares them head to head. You *can* fit a model with two knots at each location and test each knot individually relative to that:

```
stick.both <- lm(NOx ~ E + pmax(E - 0.90, 0) + pmax(E - 1.00, 0),
                 data = ethanol)

anova(stick.00, stick.both)   # is the knot at 1.00 enough on its own?
```

Are there other methods of comparison you might use?

3. Cubic spline

A cubic spline model can be written:

$$Y_i = \beta_0 h_0^{(3)}(x_i) + \beta_1 h_1^{(3)}(x_i) + \cdots + \beta_{K+3} h_{K+3}^{(3)}(x_i) + \epsilon_i$$

The functions $h_0^{(3)}, \ldots, h_{K+3}^{(3)}$ form a degree 3 b-spline basis on K interior knots. `bs(x, df = d)` places $d - 3$ interior knots at quantiles of x, so `df` sets how much flexibility you are asking for. First decide on how many knots to use, and then fine-tune where to put them.

```
sp4 <- lm(NOx ~ bs(E, df = 4), data = ethanol)   # 1 interior knot
sp6 <- lm(NOx ~ bs(E, df = 6), data = ethanol)   # 3 interior knots
sp8 <- lm(NOx ~ bs(E, df = 8), data = ethanol)   # 5 interior knots

# these are nested, so compare with an F test
anova(sp4, sp6, sp8)

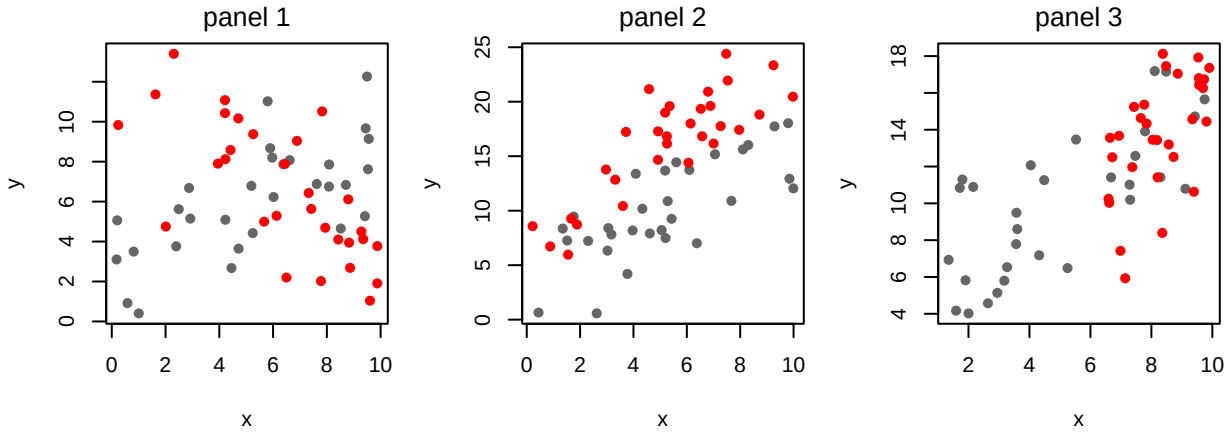
# once df is settled, the knots it chose can be given directly, and then moved
knots.auto <- quantile(ethanol$E, probs = c(0.25, 0.50, 0.75))
sp.auto <- lm(NOx ~ bs(E, knots = knots.auto, degree = 3), data = ethanol)
```

If you want extra practice, try specifying a model for the corn yield data at the end of the script.

Part 2: categorical predictors and interactions

This part accompanies the script `14-categorical-interactions.R`.

Here you'll use three simulated datasets, each with a continuous predictor `x` and a two-level group `g`. Group a is grey, group b is red. Each dataset is shown in a panel below.



For each panel, your task is to: plot it, decide which model the plot calls for, fit that model, and visualize.

For reference, relevant formulae in R:

$$\texttt{y} \sim \texttt{x} \qquad \texttt{y} \sim \texttt{x} + \texttt{g} \qquad \texttt{y} \sim \texttt{x} * \texttt{g}$$

The second of these is the model with no interactions,

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 g_i + \epsilon_i$$

which, by group, gives the means

$$g = a: \quad \beta_0 + \beta_1 x \qquad\qquad g = b: \quad (\beta_0 + \beta_2) + \beta_1 x$$

so β_2 is the difference in intercepts and the two lines are parallel.

```
fit.add <- lm(y ~ x + g, data = panel1)

# intercept and slope for group a, then the difference in intercepts
coef(fit.add)
```

To test whether the categorical variable belongs in the model, fit the model without it and compare with an F test:

```
fit.one <- lm(y ~ x, data = panel1)

anova(fit.one, fit.add)
```

The third of these is the model with an interaction,

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 g_i + \beta_3 (x_i \times g_i) + \epsilon_i$$

which, by group, gives the means

$$g = a: \quad \beta_0 + \beta_1 x \qquad\qquad g = b: \quad (\beta_0 + \beta_2) + (\beta_1 + \beta_3) x$$

so β_2 is the difference in intercepts and β_3 the difference in slopes.

```
fit.int <- lm(y ~ x * g, data = panel1)

# intercept and slope for group a, then the two differences
coef(fit.int)
```

To test whether the slopes differ, fit the model without interactions and compare with an F test:

```
fit.add <- lm(y ~ x + g, data = panel1)

anova(fit.add, fit.int)
```

For extra practice, try specifying a model for the penguin data at the end of the script, where the categorical variable has three levels instead of two.