More Linear Regression

Next, we'll continue exploring the effects of public education spending.

Massachusetts Test Score dataset

- Load the `MCAS` dataset from the `car` package into a variable `df` and read about it using `help(MCAS)`.

Cleaning the dataset

- Find the total number of rows.
- Remove the rows with missing values, and compute the number of rows of the resulting data frame.
  - Is the number of rows appreciably smaller?
  - Anticipate some statistical problems that this naive row removal could cause in our analyses.

Preliminary analysis

We'll start out with some more simple linear regressions before moving to a slightly more advanced technique.

- Compute the correlations of `totsc4` and `totsc8` with the other features in the dataset.
  - Why do you think the correlations with `totsc8` tend to be larger than the correlations with totsc4?
- Run a regression of `totsc8` against total expenditure per student, `totday`.
  - What does this say about the effect of spending per student on standardized test scores?
- Form a new data frame `df1` by removing the non-numeric columns and `totsc4`.
- Run a regression of totsc8 against the other features using `lm(totsc8 ~ . , df1)`.
  - Should we be including `code` as a predictor? If not, remove it and see how the results change.
- Run a regression of `totsc8` against the 3 predictors with p-value < 0.01 in the above regression.

    - Is the predictive power appreciably lower?

Stepwise linear regression

In general, the problem of *feature selection* is a difficult one. We ideally want to maximize predictive power using as few features as possible, because adding redundant features actually works *against* interpretability and pulls weight away from the less-redundant ones; however, with n features, we have 2^n possible combinations of features to regress against.

The most simplistic way of solving this problem is with stepwise linear regression. It works like so:

- In *forward* linear regression, we initialize the linear model with no predictors (so the model is just a constant), and we keep adding predictors which, when added, provide the greatest incremental boost to the model quality.
- In *backward* linear regression, we start with all of the predictors added to the model and successively remove predictors which, when removed, are associated with the smallest incremental drop in model quality.
- Forward and backward linear regression can be combined for a method where predictors can both be added or removed based on how doing so affects model quality.
- Eventually, according to some statistical criterion, we reach a stopping point.

The evaluation of "model quality" is often done via the Akaike information criterion, which can intuitively be thought of as being analogous to the *entropy* of a model. (Minimizing the AIC is broadly equivalent to maximizing the entropy in a thermodynamic system.)

Before learning how to run a stepwise regression in R, briefly read about its implementation.

Use stepwise regression by writing:

```r
model_init = lm(totsc8 ~ 1, df1);
model = formula(lm(totsc8 ~ ., df1))
step_reg = step(model_init, model, direction = "both")
```

- How does the resulting model differ from the model above?
- Interpret the order in which coefficients are added and removed from the stepwise model.
- What does this say about the effect of educational expenditure on student test scores for schools in the dataset?
    - Reconcile this with your earlier results.

- Repeat the above with `totsc8` replaced by `totsc4` and compare the results.