NRES 798 — Lab 9

Linear models

More on regression, and a peek at ANOVA and ANCOVA.

1 Nonparametric regression

Load the data set used in the last lab: `data(trees)`. Check it out with `names(trees)`, etc.

An example of nonparametric regression, aka *smoothing*: do `scatter.smooth(trees$Girth, trees$Volume)`. It uses a local approximation procedure (see `?loess`). That is, the $\hat{y}$ at a point x is based on observations close to x, trying to produce a smooth curve.

There are several alternatives, try `lines(supsmu(trees$Girth, trees$Volume), col='red')` and `lines(smooth.spline(trees$Girth, trees$Volume), col='blue')`. Splines are pieces of (usually cubic) polynomials, conditioned to join smoothly at their ends. That was a *smoothing spline*, function `spline` gives an *interpolating spline* that passes through each point, try it.

For output that is more interesting, repeat the three fits with `Height` substituted for `Girth`. See `?scatter.smooth`, etc., for more details.

This can be useful in exploratory data analysis. It can also help to check the residuals for trends, or to demonstrate that there is none in a more "objective" way:

```
fit <- lm(Volume ~ I(Girth^2 * Height), trees)
scatter.smooth(fitted(fit), resid(fit))
abline(0, 0)
```

The curve in the first diagnostic plot from `plot(fit)` is also produced with `loess`. By the way, including the argument `which=1:5` in `plot` gives all the 5 available diagnostic plots, including a q-q plot for normality, not just the default three.

2 Model selection

Remember the last lab, we fitted many different models to the `trees` data. Which is the best?

2.1 Same y and p

For models with the same y and the same number of parameters p, the one with the smallest residual standard error (RSE) gives the best predictions. The r-squared gives the same ranking, *provided that* the models are based on the same data.

Compare `d2 <- lm(Volume ~ I(Girth^2), trees)` and `d2h <- lm(Volume ~ I(Girth^2 * Height), trees)`. Hint: use `summary`.

But note that this is not the only consideration, `d2` does not require expensive height measurements, and might be "better" in practice.

2.2 Different y, same p

Fit `lg <- lm(log(Volume) ~ log(Girth), trees)`. Is it better or worse that `d2`? The RSE are not comparable, they are in different units, and neither is the r-squared. There are no completely satisfactory answers. One possibility is to compare the (maximized) log-likelihoods, use `logLik`, try it. But be aware that this compares not just the goodness of the regression, but also how good the assumption of normal *iid* residuals is. The residuals of `Volume` and of `log(Volume)` are very different, and for instance one may be more homoscedastic than the other.

Note: the likelihood is only defined up to an arbitrary factor, and therefore its logarithm, the log-likelihood, has an arbitrary additive constant. Only differences of log-likelihoods are meaningful, their specific values are not.

Another possibility might be to compare the regression confidence intervals. First, let's see how to get them. For `d2`, do `cd2 <- predict(d2, interval='confidence')`. The result (display it) is a matrix where the columns are the regression estimates and the confidence interval limits, evaluated at the data points by default. Do the same for the prediction intervals, substituting `prediction` for `confidence`: `pd2 <- ...` (ignore the warning). Now we can draw the textbook confidence curves:

```
plot(trees$Girth^2, cd2[,1], type='l')  # the regression line
lines(trees$Girth^2, cd2[,2])  # add one confidence limit
lines(trees$Girth^2, cd2[,3])  # and the other
lines(trees$Girth^2, pd2[,2])  # add the prediction limits
lines(trees$Girth^2, pd2[,3])
```

Now let's compare the prediction limits for the two models, in the original variables. Try to understand what this does:

```
plot(trees$Girth, pd2[,1], type='l')
lines(trees$Girth, pd2[,2])
lines(trees$Girth, pd2[,3])
# Now for lg, back-transforming the log
plg <- exp(predict(lg, interval='prediction'))
lines(trees$Girth, plg[,1], col='red')
lines(trees$Girth, plg[,2], col='red')
lines(trees$Girth, plg[,3], col='red')
```

Conclusions? Be aware, however, that the intervals depend on the assumptions about the residual distributions, which are different in the two models.

2.3 Different p

If we kep adding x's, the r-squared always increases. The RSE decreases, only increasing a little when p is fairly large, because of the denominator $n-p$. So none of these are directly useful for evaluating the fit, or for indicating when to stop, these statistics do not account for overfitting. Again, there are no good answers.

One may leave out terms where the t or F test indicates that the β_k is not significant at some chosen significance level. Try fitting `Volume` adding successively the predictors `Girth^2 * Height, Girth^2, Height` (remember the `I()`!), and look at the p-values. There are automatic ways of doing this kind of thing, called *stepwise regression*; we will see a variation in a minute. Something similar can be done with an ANOVA table. Somewhat arbitrary, and usually one should also take parsimony into account.

Another way is to prevent overfitting by penalizing complexity, represented by the number of parameters p. The *adjusted r-squared*, shown in the output from `summary.lm`, is a modification of the r-squared that decreases with p: $r^2_{adj} = 1 - (1 - r^2)(n - 1)/(n - p)$. Check the values for your models above.

More popular these days are Akaike's Information Criterion (AIC) and Schwartz's Bayesian Information Criterion (BIC). The AIC penalizes the log-likelihood by subtracting the number of parameters, and BIC subtracts the number of parameters multiplied by one half of the logarithm of the number of observations. Specifically, AIC $= -2(\text{logLik} - p)$, and BIC $= -2\,\text{logLik} + p\log(n)$ (lower is better). As before, only differences are meaningful. And the values reflect the quality of all the regression assumptions, not just the fit. Use functions `AIC` and `BIC` to compare the models.

Function `step` searches for a model like in stepwise regression, but using the AIC criterion instead of F tests. To try it, add the transformed predictors `Girth^2` and `Girth^2 * Height` to the `trees` data frame: `trees$G2 <- trees$Girth^2`, ... Then to start from the model that includes all the variables, do `step(lm(Volume ~ ., trees))`.

3 Caveats

Try this:

```
x <- 1:12
y <- x^2
fit <- lm(y ~ x)
summary(fit)
```

Does that look like a good r-squared?

```
plot(y ~ x)
abline(coef(fit))
```

Moral: plot your data, and do not trust r-squares.

Another one:

```
data(anscombe)
mean(anscombe)  # look at the means
sapply(anscombe, sd)  # standard deviations
summary(lm(y1 ~ x1, anscombe))
summary(lm(y2 ~ x2, anscombe))
```

Compare the results of the two regressions. Do the regressions for the other pairs, compare. Now plot them: `plot(y1 ~ x1, anscombe)`. etc.

4 Categorical predictors

Load the CO2 data set: `data(CO2)`. Inspect it: `summary, head, str, ?CO2`.

A linear model with a continuous (numerical, quantitative) response and categorical predictors: `lmout <- lm(uptake ~ Type + Treatment, CO2)`. Alternative function: `aovout <- aov(uptake ~ Type + Treatment, CO2)`. Try a few things, see if you can figure it out:

```
summary(lmout)
summary(aovout)
anova(lmout)
```

No problem mixing categorical and quantitative predictors: `anova(lm(uptake ~ Type + Treatment + conc, CO2))`.

To be continued...