Regression and Simulation

This is an introductory R session, so it may go slowly if you have never used R before. Do not be discouraged. A great way to learn a new language like this is to plunge right in.

We will simulate some data suitable for a regression. This will get you started in R and will teach you some useful tricks. I want you to build a script in RStudio (File/New File/R Script) and execute the lines from the script as you proceed.

1. Generate data for simple regression

The first task is to generate some data for fitting a simple regression model. By simple, this means a single predictor $X_1$ and a response $Y$. We will use a sample size of $N=100$, and generate $N$ pairs $(X_{1i}, Y_{i})$ from the model

$$Y_i = \beta_0 + X_{1i}\beta_1 + \epsilon_i.$$ 

In R we will call $X_1$ simply $X_1$, since R does not like subscripts in names. Now for some details:

- The $X_{1i}$ should come from a Gaussian distribution with mean 20 and standard deviation 3.
- Take $\beta_0 = 25$ and $\beta_1 = 2$ to be the true parameters, and
- assume that $\epsilon_i$ comes from a normal distribution with mean 0 and standard deviation 15.

Generate your data, make a histogram of $X_1$, and plot the values of $Y$ against those of $X_1$. Your plot should look something like this:

Useful functions are \texttt{rnorm}, \texttt{plot}, and \texttt{hist}. If you type \texttt{?rnorm}, for example, you will see a helpfile. In the plot, include the true regression line; \texttt{abline} is useful here.
2. Fit a linear regression

Let’s now fit a linear regression of $Y$ on $X_1$ and look at its properties. For this you will use the function `lm` in the following manner.

```r
fit = lm(Y ~ X1)
```

Use the `summary()` command on your fitted object `fit` to see what is inside. Try `coef` as well. We can include the fitted regression line in the plot. The `abline` function knows about fitted `lm` models, so you can give it as a single argument to `abline()`.

Once you get this working, see what happens if you execute your entire script (including the data generation) a second time. You should get a somewhat different result, since different data are generated. You can prevent this, if you like, by using the `set.seed` command. Type `?set.seed` to learn how. Try this and see that your data generation is now reproducible.

3. Simulation to check aspects of regression

We learn from the theory that the variance of the slope estimate in simple regression is given by the expression

$$\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{N}(X_{1i} - \bar{X}_1)^2},$$

where $\bar{X}_1$ is the mean of the $X_{1i}$, and $\sigma$ is the standard deviation of the $e_i$. (Here the “hat” on top of $\hat{\beta}_1$ means it is estimated from our data.)

What this theory is really telling us is that if we fix our $X_1$ values and repeatedly generate new $e_i$ and hence $Y_i$ and refit the regressions each time, the variance of the series of $\hat{\beta}_1$s that we get will be given by this expression. Since we typically don’t know $\sigma$, we need to estimate it as well, using the the expression

$$\hat{\sigma}^2 = \frac{1}{N-2} \sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2.$$
The linear model summary tells us the estimated standard error of the slope is 0.54 for these data (at least mine did). Let’s check it out! Run a simulation where you generate a new response vector each time, fit the regression, extract the slope coefficient, and store it. Do this 1000 times using a for loop (?‘for’), keeping the same values for $X_1$. Summarize the results and in particular compute the standard deviation of the 1000 values you get. How closely does it match the theoretical value?

4. Slightly different simulation

Here we held the original $X_1$ values fixed. What if we generate new versions of them as well when we do our simulations. Repeat your simulations, adding this little twist. What do you learn about the distribution and standard error of $\hat{\beta}_1$ in this setting? Any surprises? What would you have expected?

5. Multiple Regression

We now generate an $X_2$ and create a multiple regression model

$$Y_i = \beta_0 + X_{1i}\beta_1 + X_{2i}\beta_2 + e_i.$$ 

Let $X_2$ have the same distribution as $X_1$ (but be independent of $X_1$), and let $\beta_2 = 1$ in the simulation. Fit the multiple linear regression model — the formula is now $Y \sim X1 + X2$ — and use summary to summarize the results. What do you observe about the standard errors of each coefficient and in particular the coefficient of $X_1$?

For your current version of $X_1$ and $X_2$, what is their sample correlation? (cor()). Is your sample correlation zero? Why or why not?

Suppose we wanted the distribution of $X_2$ to be correlated with $X_1$. How might we do that? Figure out a way to do this, and experiment until your $X_2$ has sample correlation about 0.6 with $X_1$. Now generate a $Y$ from your regression model, and use summary to examine the coefficients. What do you observe?

Now try to produce $X_2$ with sample correlation about 0.8 with $X_1$, and repeat the regression experiment above. What do you observe?