# Multiple regression - a brief introduction

Multiple regression is an extension to regular (simple) regression. Instead of one $X$, we now have several.

Suppose, for example, that you are trying to predict plant growth. In simple regression you might do something like increase the amount of fertilizer to see what the effect would be on growth. For example:

You might let $Y$ = *weight of dried plant* and $X$ = *amount of fertilizer*. Your regression equation would be the usual:

$$\hat{Y} = \beta_0 + \beta_1 X$$

If you were really interested in predicting plant growth, then it might make sense to consider other variables as well. The amount of light, temperature and moisture might all make an important contribution to plant growth.

But does it make sense to perform four regressions, one for each of your $X$'s?

It would make more sense to see if you can use all four factors at once to predict your plant growth. In other words, you want to do the best possible job predicting plant growth, so instead of using one factor at a time, you should use them all at once.

So let's summarize. You have four factors:

| | | |
|---|---|---|
| First $X$: | amount of fertilizer | $= X_1$ |
| Second $X$: | amount of light | $= X_2$ |
| Third $X$: | temperature | $= X_3$ |
| Fourth $X$: | moisture level | $= X_4$ |

Now you put these together in emphone regression equation as follows:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_4 X_4$$

Some comments:

Note that now you need to estimate five $b$'s, not just two as in regular regression. As you'll see in a moment, this becomes increasingly complicated.

With just two independent variables (say, $X_1$ and $X_2$), we can still illustrate what is going on with a plane intercepting the $Y$ axis. With more than two independent variables, we can no longer visualize what is going on.

Remember that the $b$'s are sample estimates. The population parameters are indicated (as usual) by $\beta$.

So how do we make it work? The first step is to estimate the various parameters ($\beta_0$ through $\beta_4$ in our example above).

To do this, we obviously need to calculate the values for the $b$'s.

In essence this is similar to what we did for simple regression:

If we had just two $X$'s, for example, we could take a plane through our points and rotate and twist it until the *residuals* are at a minimum.

With more than two $X$'s this is a little hard to visualize.

In any case, we derive equations that will give us the least squares estimates.

These are now quite a bit more complicated. If we have just two $X$'s, for example, we get the following:

$$b_1 = \frac{\sum_{j=1}^{n}(x_{2j} - \bar{x}_2)^2 \sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(y_j - \bar{y}) - \sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(x_{2j} - \bar{x}_2) \sum_{j=1}^{n}(x_{2j} - \bar{x}_2)(y_j - \bar{y})}{\sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(x_{2j} - \bar{x_2}) - \left(\sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(x_{2j} - \bar{x}_2)\right)^2}$$

$$b_2 = \frac{\sum_{j=1}^{n}(x_{1j} - \bar{x}_1)^2 \sum_{j=1}^{n}(x_{2j} - \bar{x}_2)(y_j - \bar{y}) - \sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(x_{2j} - \bar{x}_2) \sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(y_j - \bar{y})}{\sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(x_{2j} - \bar{x_2}) - \left(\sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(x_{2j} - \bar{x}_2)\right)^2}$$

$$b_0 = \bar{y} - b_1\bar{x}_1 - b_2\bar{x}_2$$

Obviously this gets absurd very quickly. And we only have two $X$'s so far.

Unfortunately, to really understand this, we need to make use of matrix algebra. For example, we can simply re-write the above as:

$$\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'Y}$$

which looks a lot easier until we realize that $\mathbf{b}$ is a vector with all of our $b$'s, $\mathbf{Y}$ is a vector with all the values of our $Y$, and $\mathbf{X}$ is a matrix with *all* the values for our independent variables.

The advantage of matrix notation is not so much that it reduces the calculations needed (it doesn't), but that it let's us express very messy equations in a way that one can actually understand (look at the equations for $b_1$ and $b_2$ above if you don't believe this!).

The big disadvantage is that we would need to spend considerable time learning matrix (or linear) algebra. To learn just what we need would probably take several weeks.

We also haven't discussed testing for significance yet. In simple terms, it's not too difficult to understand - we do the following:

$$t_1^* = \frac{b_1}{SE_{b_1}} \qquad t_2^* = \frac{b_2}{SE_{b_2}}$$

Which looks deceptively simple. We won't give the equations for any of the $SE_{b_i}$'s, but let's just say matrix algebra is your friend (really!).

So where does that leave us?

We can't really learn the math needed in the time we have. However the above should give you an idea of what is happening in multiple regression.

Instead, we will let R do all of the work and not worry too much about the annoying math.

Multiple regression is actually very simple in R, and we'll use an example to walk through the procedure.

Let's first outline what we need to do:

1 Enter your Y in one column, and each of your X's in a separate column. You should make sure, of course, that they're all the same length.
You can, of course, use Excel to enter your data and then import it into R.

2 Once your data have been entered, you use the `lm` command just as you did for simple regression:

```
lm(y ~ x1 + x2 + x3)
```

3 The output should look very similar to what you had for simple regression except there will now be a line for each independent variable.

Now let's do an example. We'll pick the one from your text starting on p. 421:

> This is a hypothetical data set; we will assume we want to predict *ml* from the other four variables.
>
> First we'll enter the data (we'll do everything from the command line):

```
cent <- scan(nlines = 2)
6 1 -2 11 -1 2 5 1 1 3 11 9 5 -3 1 8 -2 3 6 10 4 5 5 3 8 8
6 6 3 5 1 8 10

cm <- scan(nlines = 3)
9.9 9.3 9.4 9.1 6.9 9.3 7.9 7.4 7.3 8.8 9.8 10.5 9.1 10.1
7.2 11.7 8.7 7.6 8.6 10.9 7.6 7.3 9.2 7.0 7.2 7.0 8.8 10.1
12.1 7.7 7.8 11.5 10.4

mm <- scan(nlines = 3)
5.7 6.4 5.7 6.1 6.0 5.7 5.9 6.2 5.5 5.2 5.7 6.1 6.4 5.5 5.5
6.0 5.5 6.2 5.9 5.6 5.8 5.8 5.2 6.0 5.5 6.4 6.2 5.4 5.4 6.2
6.8 6.2 6.4

min <- scan(nlines = 3)
1.6 3.0 3.4 3.4 3.0 4.4 2.2 2.2 1.9 0.2 4.2 2.4 3.4 3.0 0.2
3.9 2.2 4.4 0.2 2.4 2.4 4.4 1.6 1.9 1.6 4.1 1.9 2.2 4.1 1.6
2.4 1.9 2.2

ml <- scan(nlines = 3)
2.12 3.39 3.61 1.72 1.80 3.21 2.59 3.25 2.86 2.32 1.57 1.50
2.69 4.06 1.98 2.29 3.55 3.31 1.83 1.69 2.42 2.98 1.84 2.48
2.83 2.41 1.78 2.22 2.72 2.36 2.81 1.64 1.82
```

> Now we can just use the `lm` command (we'll assume that *ml* is our dependent variable):

```
hypot <- lm(ml ~ cent + mm + min + ml)

summary(hypot)
```

> And we should get the following result from R:

```
Call:
lm(formula = ml ~ cent + cm + mm + min)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5070 -0.1112  0.0401  0.1550  0.9617
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.95828    1.36361   2.169  0.03869 *
cent        -0.12932    0.02129  -6.075  1.5e-06 ***
cm          -0.01878    0.05628  -0.334  0.74102
mm          -0.04621    0.20727  -0.223  0.82518
min          0.20876    0.06703   3.114  0.00423 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.4238 on 28 degrees of freedom
Multiple R-squared:  0.6589,Adjusted R-squared:  0.6102
F-statistic: 13.52 on 4 and 28 DF,  p-value: 2.948e-06
```

Now let's explain the above.

You should be used to interpreting the above by now. The column labeled `t value` is the actual value of $t^*$ that we calculate. The last column is the $p$-value.

The first column gives us the labels of each of our rows. Here we see our four independent variables (our $X$'s). From this we see that cent and min are significant at $\alpha = 0.05$. The others are not significant (we almost always ignore the row for intercept).

R gives us more. The last line, for example, give us an $F$-statistic. This is an overall test to see if the model with all four $X$'s is significant, and in this case we get a nice small $p$-value.

It's a little bit like using Tukey's after an ANOVA; the $F$-test tells us the overall regression is significant, and the various $t$-tests then tell us which of the $X$'s are important (it's similar, but *not* the same).

(We don't have the time to see how $F$-tests work in regression, but the approach is very similar to that used in ANOVA).

Now we've done our first multiple regression. But just as in regular regression, we need to worry about our assumptions.

The assumptions are essentially identical to simple regression (at least for us).

The good news is that checking the assumptions isn't really any more difficult in multiple regression, but it is a bit more tedious since you need to make a few more plots:

1 Plot the residuals in a q-q plot. In R (continuing our example above) you would simply do:

```
qqnorm(hypot$residuals)
qline(hypot$residuals)
```

2 Plot the residuals versus the fitted values. This is a little like our standard residual plot, and for all intents the interpretation is the same. In R you would do:

```
plot(hypot$fitted.values,hypot$residuals)
```

You should do the above two plots at a *minimum*. However, they can't always find all the problems. So you really ought to do the following as well:

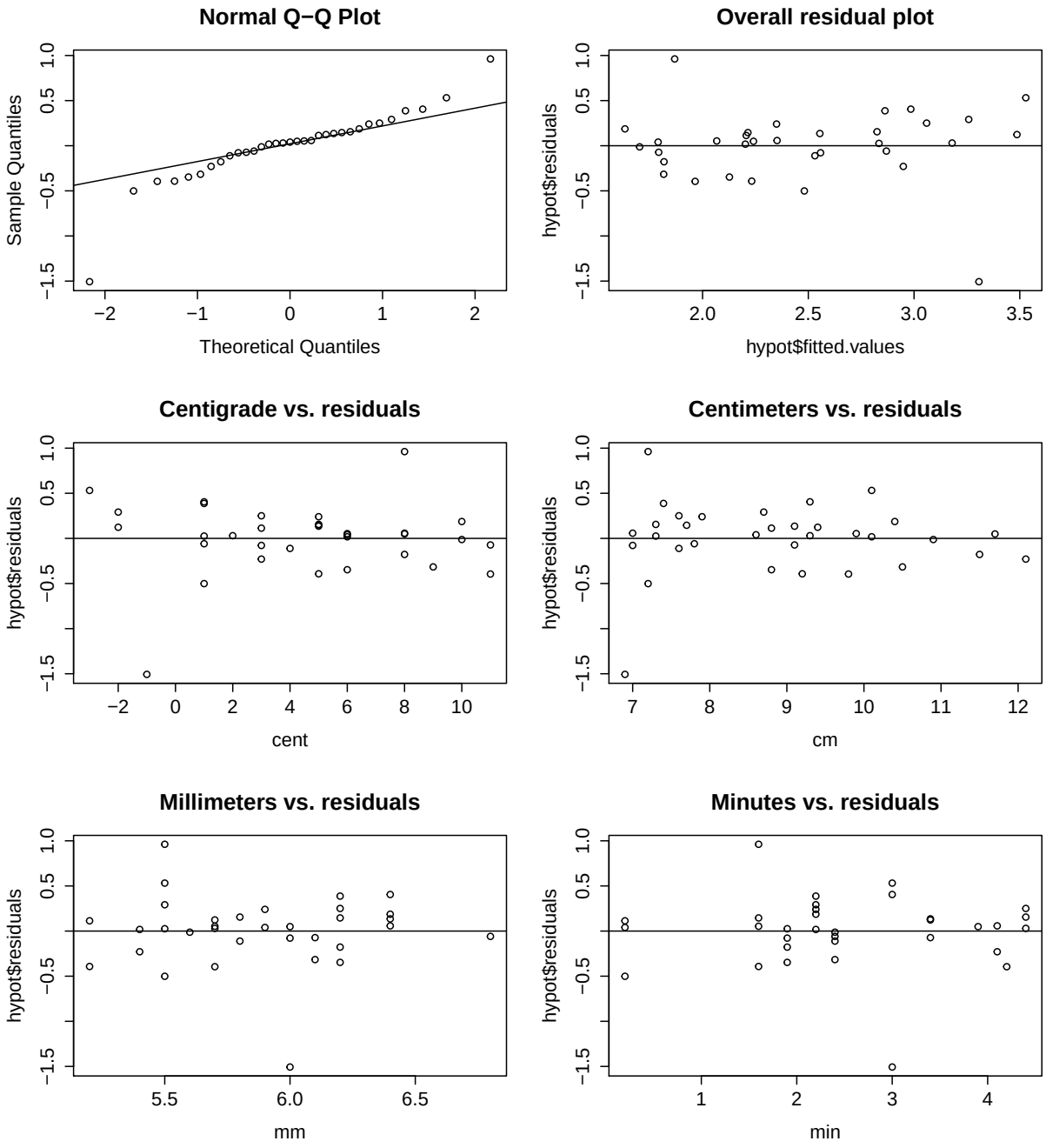
3 Plot the residuals versus *each* of the original $X_i$'s. In R we would do:

```
plot(cent,hypot$residuals)
plot(cm,hypot$residuals)
plot(mm,hypot$residuals)
plot(min,hypot$residuals)
```

(If you want a straight line at 0 for any of the plots above using the `plot` command, just do `abline(0,0)` after the plot command).

These last plots will sometimes uncover problems not visible in the overall residual plot. Also - if the overall residual plot shows any problems, these often provide a way to find out where the problem is (i.e., which $X$ (if it is an $X$) is causing the problem).

An example of all six plots needed for this multiple regression is on the following page.




The q-q plot (top left) shows somewhat long tails, particularly on the left side.

Most of the residual plots don't show any particular problem. The overall plot (top right) shows a few outliers that might be worth investigating.

Overall, a few minor problems that might bear investigating, but probably good enough to go ahead with the analysis.

Finally, we need to do one more thing.

Suppose we get a little nuts and have eight $X$'s. Don't you think that might be a bit much?

One important use of regression is to figure out which of these $X$'s might be the most important. All eight $X$'s will be very confusing to interpret.

*Note: letting a computer decide which variables are important can be controversial.*

You really ought to have some idea of what is going on with your variables (do you really think you need all eight $X$'s)?

However, sometimes it's the only way to weed through all the information, so techniques like this are used. You should *always* make sure, however, that the variables (model) you wind up with actually makes some kind of sense.

Techniques have developed that let you weed out the supposedly unimportant variables.

For example, remember that *cm* and *mm* were not significant in our regression model - that might (*but not necessarily*) indicate that we don't need them.

We will look at *stepwise regression* and briefly mention *all best regression.*

In both cases, we somehow measure how well the model does as we add or subtract the $X_i$'s.

In other words, we calculate many, many regressions. Each time we add or subtract one or more of our $X_i$'s and then we see how well our model does using some type of "measure".

These days, a popular measure is the AIC - **A**kaike's **I**nformation **C**riterion.

We can't dig into the math behind the AIC as it get's pretty complicated. Let's just say that the AIC indicates how well each regression is doing. It penalizes a regression with too many variables, which helps us simplify (i.e., drop variables) the regression

There are other criteria that can be used. For example, the BIC or Mallow's $C_p$, but we really don't have the time to delve into this too much.

The easiest way to proceed is just to do an example. We'll stick with the example above and do a stepwise regression and then explain things a bit.

In R, we do the following (after we have done the initial regression):

```
library(MASS)
```

(If this gives you an error ("there is no package..." etc.) then install the MASS package by clicking on **Packages** at the top of the bottom right window in R-studio, then click on **Install**, and type **MASS** on the middle line of the box that pops up. Then click on **Install** at the bottom of the pop up window and the package should get installed).

This loads the MASS package which really is the easiest way to do stepwise regression.

Now we just tell R to do stepwise regression by doing:

```
stepAIC(hypot, direction = "both")
```

And you should get the following result:

```
Start:  AIC=-52.08
ml ~ cent + cm + mm + min

       Df Sum of Sq     RSS     AIC
- mm    1    0.0089  5.0388 -54.018
- cm    1    0.0200  5.0499 -53.946
<none>               5.0299 -52.077
- min   1    1.7421  6.7720 -44.263
- cent  1    6.6298 11.6596 -26.332

Step:  AIC=-54.02
ml ~ cent + cm + min

       Df Sum of Sq     RSS     AIC
- cm    1    0.0145  5.0533 -55.923
<none>               5.0388 -54.018
+ mm    1    0.0089  5.0299 -52.077
- min   1    1.8191  6.8579 -45.847
- cent  1    7.1685 12.2073 -26.818
```

```
Step:  AIC=-55.92
ml ~ cent + min

       Df Sum of Sq     RSS     AIC
<none>               5.0533 -55.923
+ cm    1    0.0145  5.0388 -54.018
+ mm    1    0.0035  5.0499 -53.946
- min   1    1.8177  6.8710 -47.784
- cent  1    8.2522 13.3055 -25.975

Call:
lm(formula = ml ~ cent + min)

Coefficients:
(Intercept)         cent          min
     2.5520      -0.1324       0.2013
```

So what does it all mean?

Once again we need to wave our hands a bit and say we don't have the time to go into the details. Let's just say that you pick the regression with the smallest AIC value.

Look at the AIC given right after the `Step:` statement. The smallest one is the last one which is -55.92. This indicates that the best model (using stepwise regression) is the one that only includes *min* and *cent* (which is what we hinted at above).

Of course, we do need to make some comments:

> Stepwise regression (by default in R) puts all the variables into the model, then starts kicking them out one at a time (for example, by selecting the one with the highest $p$-value).
>
> If it doesn't like what it sees, it may try to add variables back in (for example, it might kick out a variable, run the regression, discover that that wasn't a good idea, so add it back in and try kicking out another variable).
>
> It's important to realize that stepwise regression doesn't always give you the best possible result (as measured by AIC).

The other procedure we should briefly mention is *all best regression*.

It's similar to stepwise except that it actually does calculate the best possible regression given with a specific number of $X$'s.

For example, if we have four $X's$, it picks the best possible regression with three $X$'s, the best possible regression with two $X$'s, and the best possible regression with one $X$ (and of course, looks at the regression with all four $X$'s).

Then you select the best regression from the output (using something like AIC).

Unfortunately this is a bit complicated in R since we need to convert our data into a `data frame` in R (or use the `read.table` command). So we won't learn how to do this.

However, the reason for mentioning this approach is that it often gives a better result than stepwise regression, and if you need to select variables to put into your regression, it's highly recommended that you learn a little more about it.