**Ordered factors**

Since "Male" and "Female" are unordered (or nominal) factor levels, R returns a warning message, telling you that the greater than operator is not meaningful. As seen before, R attaches an equal value to the levels for such factors.

But this is not always the case! Sometimes you will also deal with factors that do have a natural ordering between its categories. If this is the case, we have to make sure that we pass this information to R...

Let us say that you are leading a research team of five data analysts and that you want to evaluate their performance. To do this, you track their speed, evaluate each analyst as "slow", "fast" or "insane", and save the results in speed\_vector.

**Ordered factors (2)**

speed\_vector should be converted to an ordinal factor since its categories have a natural ordering. By default, the function [factor()](http://www.rdocumentation.org/packages/base/functions/factor)transforms speed\_vector into an unordered factor. To create an ordered factor, you have to add two additional arguments: ordered and levels.

factor(some\_vector,

ordered = TRUE,

levels = c("lev1", "lev2" ...))

By setting the argument ordered to TRUE in the function [factor()](http://www.rdocumentation.org/packages/base/functions/factor), you indicate that the factor is ordered. With the argument levels you give the values of the factor in the correct order.

# Comparing ordered factors

Having a bad day at work, 'data analyst number two' enters your office and starts complaining that 'data analyst number five' is slowing down the entire project. Since you know that 'data analyst number two' has the reputation of being a smarty-pants, you first decide to check if his statement is true.

The fact that factor\_speed\_vector is now ordered enables us to compare different elements (the data analysts in this case). You can simply do this by using the well-known operators.