# What's a factor and why would you use it?

In this chapter you dive into the wonderful world of **factors**.

The term factor refers to a statistical data type used to store categorical variables. The difference between a categorical variable and a continuous variable is that a categorical variable can belong to a **limited number of categories**. A continuous variable, on the other hand, can correspond to an infinite number of values.

It is important that R knows whether it is dealing with a continuous or a categorical variable, as the statistical models you will develop in the future treat both types differently. (You will see later why this is the case.)

A good example of a categorical variable is sex. In many circumstances you can limit the sex categories to "Male" or "Female". (Sometimes you may need different categories. For example, you may need to consider chromosomal variation, hermaphroditic animals, or different cultural norms, but you will always have a finite number of categories.)

#Instructions

Assign to variable theory the value "factors for categorical variables".

# What's a factor and why would you use it? (2)

To create factors in R, you make use of the function [**factor()**](http://www.rdocumentation.org/packages/base/functions/factor). First thing that you have to do is create a vector that contains all the observations that belong to a limited number of categories. For example, sex\_vector contains the sex of 5 different individuals:

sex\_vector <- c("Male","Female","Female","Male","Male")

It is clear that there are two categories, or in R-terms **'factor levels'**, at work here: "Male" and "Female".

The function [**factor()**](http://www.rdocumentation.org/packages/base/functions/factor) will encode the vector as a factor:

factor\_sex\_vector <- factor(sex\_vector)

#Instructions

* Convert the character vector sex\_vector to a factor with factor() and assign the result to factor\_sex\_vector
* Print out factor\_sex\_vector and assert that R prints out the factor levels below the actual values.

# What's a factor and why would you use it? (3)

There are two types of categorical variables: a **nominal categorical variable** and an **ordinal categorical variable**.

A nominal variable is a categorical variable without an implied order. This means that it is impossible to say that 'one is worth more than the other'. For example, think of the categorical variable animals\_vector with the categories "Elephant", "Giraffe", "Donkey" and "Horse". Here, it is impossible to say that one stands above or below the other. (Note that some of you might disagree ;-) ).

In contrast, ordinal variables do have a natural ordering. Consider for example the categorical variable temperature\_vector with the categories: "Low", "Medium" and "High". Here it is obvious that "Medium" stands above "Low", and "High" stands above "Medium".

#Instruction

Click 'Run' to check how R constructs and prints nominal and ordinal variables. Do not worry if you do not understand all the code just yet, we will get to that.

# Factor levels

When you first get a data set, you will often notice that it contains factors with specific factor levels. However, sometimes you will want to change the names of these levels for clarity or other reasons. R allows you to do this with the function [**levels()**](http://www.rdocumentation.org/packages/base/functions/levels):

levels(factor\_vector) <- c("name1", "name2",...)

A good illustration is the raw data that is provided to you by a survey. A common question for every questionnaire is the sex of the respondent. Here, for simplicity, just two categories were recorded, "M" and "F". (You usually need more categories for survey data; either way, you use a factor to store the categorical data.)

survey\_vector <- c("M", "F", "F", "M", "M")

Recording the sex with the abbreviations "M" and "F" can be convenient if you are collecting data with pen and paper, but it can introduce confusion when analyzing the data. At that point, you will often want to change the factor levels to "Male" and "Female" instead of "M" and "F" for clarity.

**Watch out:** the order with which you assign the levels is important. If you type levels(factor\_survey\_vector), you'll see that it outputs [1] "F" "M". If you don't specify the levels of the factor when creating the vector, R will automatically assign them alphabetically. To correctly map "F" to "Female" and "M" to "Male", the levels should be set to c("Female", "Male"), in this order.

#Instructions

* Check out the code that builds a factor vector from survey\_vector. You should use factor\_survey\_vector in the next instruction.
* Change the factor levels of factor\_survey\_vector to c("Female", "Male"). Mind the order of the vector elements here.

# Summarizing a factor

After finishing this course, one of your favorite functions in R will be [**summary()**](http://www.rdocumentation.org/packages/base/functions/summary). This will give you a quick overview of the contents of a variable:

summary(my\_var)

Going back to our survey, you would like to know how many "Male" responses you have in your study, and how many "Female" responses. The [**summary()**](http://www.rdocumentation.org/packages/base/functions/summary) function gives you the answer to this question

#Instructions

Ask a [**summary()**](http://www.rdocumentation.org/packages/base/functions/summary) of the survey\_vector and factor\_survey\_vector. Interpret the results of both vectors. Are they both equally useful in this case?