Audio transcription for dealing with missing data in PCA

Slide 1: In this video, we're going to look at how to do PCA on an incomplete dataset, that is, one with missing data. This is a very common situation in practice, and there are, unfortunately, many reasons why data can be missing, such as: people refusing to answer certain survey questions, machines that break down, misplaced data, etc., etc.
To take into account and deal with missing data in PCA, we're going to use the missMDA package, which can be used in harmony with the FactoMineR package.

Slide 2: In this video, we're going to use the dataset called "orange", which can be found in the missMDA package. This dataset contains twelve individuals (here, twelve orange juices) characterized by eight quantitative variables. Some of the data is missing, and represented by the letters NA, which means: not available.

Slide 3: One simple solution involves removing individuals or variables which have at least one missing entry. In our example, this would leave us with only three individuals and one variable, which is clearly a bad way to progress. Indeed, simply removing whole rows or columns like this is rarely a good idea, and speaking as someone that has to spend a lot of time obtaining the data, no-one likes throwing away a whole individual or variable for the sake of a missing entry. A different way to go about things is to replace each missing value by the mean over the relevant variable. This is the default thing that happens in many software packages, including FactoMineR.

So, if we run a standard PCA, as previously, PCA(orange), we get the PCA plots from the dataset in which missing elements have first been replaced by the mean of the relevant variables. This is a rather unsophisticated way to approach the question. To understand why, let's take an example with two variables, x and y, strongly correlated, both with missing values.

If we input missing data by replacing it by the mean, this is what we get. The black points are individuals without missing values, and the strong link between x and y is very obvious. The green points are individuals who had a missing value in x, and the red points are individuals who had a missing value in y. Clearly, imputing with the mean completely deforms the distributions of x and y, and the relationship between the two variables. What we can conclude is that replacing missing data by the mean is not a very satisfactory way to proceed, either. So what can we do? Well, other more sophisticated methods exist, which take into account structure in the data, and tend to work better in practice.

Slide 4: Here, for example, as the variables x and y are strongly correlated, if a value of y is missing for a certain individual, it would seem natural to estimate it using the value of x for the same individual, via, perhaps, a simple linear regression.

What about when there are lots of variables? Well, if individuals i and j have very similar values across all the variables, and individual i has a missing value for the k-th variable, it would make sense to estimate the missing value using the one for individual j for the k-th variable, because the two individuals seem to be very similar. We can apply these two ideas at the same time for a set of individuals and a set of variables, by taking into account both the global similarities between individuals across all variables, and the links between variables across all individuals. Let's pause for a second and ask what this means, exactly: "by taking into account both the global similarities between individuals across all variables, and the links between variables across all individuals." Does it ring a bell? Basically, we're talking about PCA! So, yes, it's true, we can impute missing data using PCA.

Slide 5: This is the basic idea behind an iterative PCA algorithm, which works as follows. To simplify, let's work with a little example with five individuals and two variables, x1 and x2. Four individuals have no missing data, while one, the 4th, has a missing entry for x2. Let's put the four individuals with no missing data on the plot, and put a little mark here for the individual with an x2 value missing.

The first step of the iterative PCA algorithm is to impute the missing data using a random yet reasonable value. For example, the mean for the variable in question, or, in our case, simply the value 0.

Then, we alternate between two steps. The first consists of doing PCA on the now "complete" data. We do the PCA and draw the PCA axis on the plot. Each individual is projected onto this line, giving the little red dots. This gives us coordinate values for these points in terms of x1 and x2.

The 4th individual, which had a missing value for x2, can now be projected, and ends up with a value of 0.57.

We can now take this value of 0.57 as the new starting point for the next iteration. We start again with all the observed data values, and with 0.57 in the place of the missing entry.

We keep on iterating these two steps: doing the PCA, then getting a new coordinate value for the 4th individual for x2, and updating the data table.

We keep on iterating until it converges. In our example, the x2 coordinate of the 4th individual converges to 1.48, and the final PCA line is the one shown here.

Slide 6: Here is a summary of the algorithm's steps. First, we provide an initial guess for the missing values, such as, for example, the mean of each variable.

Then we iterate the a, b and c steps. Step (a) consists of doing PCA on the imputed table. We choose the number of dimensions we want to use to do the next update. In our example, we chose one dimension because we were constructing one PCA line. The choice of number of dimensions for helping to impute the table is an important one.

Step (b) involves updating the missing value estimations using PCA, using both the individuals' and variables' coordinates. And then, step (c), we update the means and standard deviations of each variable after updating.

Then we go back to step (a) and do a PCA, (b) we update the missing value estimates, and then we iterate the three steps until convergence. Usually, convergence is quick, though the algorithm can sometimes over-fit.

In practice, overfitting is common. It comes from believing too much in the links between variables, which may not be as strong as we think, notably, due to missing data. For this reason, we usually prefer to use a regularized version of the iterative PCA algorithm. We're not going to go into the details here, but this version is what's implemented in missMDA, and what we use in practice. The basic idea is to not believe quite so much in the links between variables, and to impute missing data with a value that is a bit closer to the variable's mean than it would have been otherwise. This is a way of taking less risk when imputing the missing data.

Ultimately, the algorithm estimates the missing data values with values that have no influence on the PCA results, that is, no influence on the coordinates of the individuals or variables. And because the algorithm is based on PCA, it takes simultaneously into account: similarities between individuals and links between variables. We have to say as well that this imputation method gives good quality results compared with other statistical imputation methods, and also, remember that this method can be used to do PCA, but also other statistical analyses.

Slide 7: This regularized iterative PCA algorithm is implemented in the missMDA package. This package contains several functions. As we've already seen, it's necessary to estimate the number of PCA dimensions to use when imputing the missing data. We can do this using the estim_ncpPCA function.

This function finds the optimal number of components to use when imputing missing data. To be clear, we'll use a standardized PCA for imputing, which is the default option, and it's what we have to use if we want to do a standardized PCA on the dataset after. What we mean is: in our example, we get that the estimate optimal number of components is two. This is the number to use when imputing the missing data. We can then run the imputation algorithm, using the iterative method we talked about earlier, by typing: imputePCA(orange, ncp = 2, scale = TRUE)
where the "two" means that we want to use two PCA components, and scale=TRUE because we want to use this standardized PCA to impute the missing data in the table.

The resulting imputed data table is found in the completeObs object. If we compare it to the original data, we see that the initial data hasn't changed, and where there was missing data, such as for individual 1 and the "sweet" variable, we now have the value 5.54. This estimate should be better than simply using the mean, because we've taken into account the links between variables and similarities between individuals in its calculation. We can now run a normal PCA on this table, using the PCA function in FactoMineR. It outputs the usual PCA results, with the plots for the individuals and variables.

Slide 8: Ok, so, a few remarks about the imputation using PCA for this dataset, and its consequences on the PCA that follows, for the imputed dataset. When we are doing the imputation, the missing data are, if you like, "predicted" in the chosen subspace. This subspace depends on the number of dimensions we use to impute the data.

If we use two dimensions, the missing data are imputed using values from where the individuals are placed in the first two dimensions, and the variables too. We therefore amplify the projection quality on the first two dimensions, and thus over-estimate the percentages of inertia of the PCA associated with the first two dimensions.
Taking again the toy example with the two variables and imputation using the PCA line, so, by one dimension only, we see clearly that the points corresponding to the imputed data are on the PCA line. And so, in what follows, if we do a PCA on this imputed dataset, the first PCA component will have an overly large percentage of inertia.
Therefore, the percentages of inertia for the PCA associated with the first dimensions of the PCA will be over-estimated if we impute missing data. Therefore, when we move to interpreting the results, we should not go overboard in what we say about the percentages of inertia associated with the first dimensions. Especially if there are a great many missing data.

Slide 9: Another problem: the imputed data are, when the PCA is performed, considered like real observations. But no, they are estimations, and so there's uncertainty in each of them. One way to evaluate this uncertainty is to perform what we call multiple imputation.

Instead of imputing the data table one time only, we're going to do it several times and with different values each time in order to see the plausible values that can take a missing value and thus the variability induced by the uncertainty on the imputed values.

Slide 10: The MIPCA function from the missMDA package lets us generate this set of tables, each with different imputed values for each missing data. The plot.MIPCA function, which can simply be called using "plot", then allows us to visualize this variability, that is, uncertainty, on the plane defined by two PCA axes.

There are two ways to visualize this uncertainty. The first consists of projecting, as supplementary variables, the multiple tables onto the PCA plane obtained using one-time imputation. We therefore consider that the factor dimensions are fixed, and only the uncertainty in the positions of the individuals and variables which have missing data is observed.
Thus, in our example, individual 12 has no missing data, so there is no uncertainty in its position, while individual 5 has several missing data values, so uncertainty in its position is represented by an ellipse. Similarly, the "Typicity" variable has no missing data values, whereas the "sweet" variable has several, so there is uncertainty in its position. Each blue point corresponds to an arrow's tip for this variable. The cloud of blue points is quite spread out, which means we should interpret its position with care.

A second way to take uncertainty into account is to suppose that the imputed values have a certain level of variability, and that this variability also has an influence on the factor dimensions. So, we're going to represent this uncertainty in terms of the individuals' positions, all of them, including those that didn't have any missing data values, because the individuals which do have missing data values make the axes move, and thus the positions of "all" individuals. This strategy leads to "this" plot for the individuals, this time with an ellipse around "all" of them, including individual 12, who didn't have any missing data values.

The plot on the right shows the projections of the PCA dimensions of each imputed table on the PCA plane obtained using the original imputed data table. As all of the arrows are close to either the first or second axes, this means that the axes are stable with respect to the set of imputed tables. If there had been evidence of instability here, we might not have wanted to interpret the PCA results too closely, because they seem to depend too much on the imputed values of the missing data. This strategy helps us to be sure that there are not too many missing data points, and that it remains possible and meaningful to do PCA.

And to finish up, try to keep in mind that missing data can be imputed by a variety of methods, but none of them will never be as good as having the actual observed data point, and we should always keep this in mind when interpreting the post-imputation results. If possible, try to measure or get a feel for the impact of the imputation, as we have just explained, before coming to too-strong conclusions. ESPECIALLY when there is a lot of missing data.