z-Transformation (= z-scores)

The so-called *z-transformation* is often applied to normalise (= standardise)data to be able to compare 'values' between different speakers or data-ranges. The idea behind this transformation is to subtract the arithmetic mean of some data (e.g. of a speaker or a segment) from a particular value. This shifts this value into a range of negative and positive values around the mean. And then divide this difference by the standard deviation of the data (speaker or segment). This will normalize the data to ±1 for a value that is ± the standard deviation away from the mean. As an outcome, the z-transformed data form a distribution around '0' with the standard deviation of '1', independent of the original data range.

Formula:

$$z = \frac{x_i - mean(x)}{st.dev(x)}$$

A typical formula in JMP looks then like (in this example, F1-values normalised for each speaker):

$$\frac{\text{F1 - Col Mean (F1, Speaker)}}{\text{Col Std Dev (F1, Speaker)}}$$

In case the data should be normalised in relation to all data (and not speaker specific, JMP provides the function *Col Standardize*:

Col Standardize (F1)

Transformation of pitch data

Using semitones or ERB values allows the comparison of speaker with very different pitch ranges and is the preferred method. Using z-scores additionally normalises each segment by transforming the pitch values to a distribution around the a mean of '0' and the standard deviation of '1' by the formula (in this example, for a phrase):

$$z = \frac{F_0(\text{at time point}) - F_0 mean(\text{of phrase})}{F_0 stdev(\text{of phrase})}.$$

Using z-score transformed semitone values to compare pitch contours is an often applied method but can lead to misleading results if a contour is rather flat (cf. https://sites.google.com/site/tonemodelling/anaposts/z-transformdoesnotworkforpitchcontoursflat). The normalisation by subtracting the mean only seems to be a more appropriate method, but I suggest to apply this method only to ERB or semitone scaled data (and not to Hertz values directly):

$$z = F_0(\text{at time point}) - F_0 mean(\text{of segment})$$

Transformation of Formant data (vowel space)

For formant data of vowels, there have been several normalisation methods proposed. A good overview article is

Fabricius, Anne H.; Watt, Dominic & Johnson, Daniel Ezra (2009). A comparison of three speaker-intrinsic vowel formant frequency normalization algorithms for sociophonetics. *Language Variation and Change* **21**, 413–435.

There is also an online version to apply these methods

<http://lingtools.uoregon.edu/norm/norm1.php>

and an R-function

<http://lingtools.uoregon.edu/norm/package/html/norm.wattfabricius.html>