1. Comment est codé le calcul des doubles sommes nécessaires à l'estimation des moments géométriques

First, for each row of pixels, we calculate an array with x^p. For example, for p=1, that gives us x = [1, 2, 3, …]. We do the same for each column of pixels, we calculate y^q. Afterwards, we do a matrix multiplication: rbind (x) %*% im %*% cbind (y). That multiplies array x with matrix im and then the multiplies result with array y. Finally, we return the result as a float: as.numeric(…)

2. After calculating the inertion matrix for each form, we can conclude that each form, depending on it´s rotation, has its own eigen vectors. That allows us to define a form more concretely, so it is definitely a good form attribute. For example, for the 3 rectangles (diagonal, horizontal and vertical) all of them have different inertion matrixes. We can easily tell the main axis by the values in the matrix. However, the inertion matrix DOES NOT tell us anything about how large a form is. For the 2 squares (carre-10 and carre-6), they both have the same orientation, but different sizes.

Run rdfTesteMoments.R to see the values

3. Upon normalising the moment, we conclude that the same figure (but of differrent scales) has the SAME moment. For example, for M00, the result will be 0 (of course). Also, normalising the moment does NOT affect rotation.

4. Hu. When calculating moment invariants for digits, all of them are different. When calculating for the same shapes, but rotated, they are also different. For forms of the same shape, but different scales, only the first invariant is different (for example for the squares).