Regression II
Vanderbilt University
Prof. Sean P. Corcoran

Mahalanobis Distance Example

See the accompanying do file *Mahalanobis distance example*. This file illustrates the concepts of Mahalanobis distance (and how it differs from Euclidean distance) using a simple example and generated data.

---

1. The example begins by generating a $N = 100$ dataset with two random variables, x_1 and x_2. Both have a mean of zero. x_1 has a variance of 10 and x_2 has a variance of 5. Their covariance is 3.5, which in this case translates into a correlation of about 0.5 ($\rho = \sigma_{XY}/\sigma_X\sigma_Y$). A scatter plot of the raw data is shown in the first figure below.

2. The do file identifies nearest neighbors using three methods. The first is Euclidean, or straight-line, distance. By definition the straight-line distance between two points (x_1, y_1) and (x_2, y_2) is: $\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$. The user-written Stata command called `nearest` will identify nearest neighbors by this measure. (`teffects nnmatch` also calculates Euclidean distance, but it requires an outcome and treatment variable to be provided. If you just want to find nearest neighbors based on the x variables, the `nearest` command will work). For illustration, the first 10 observations and their nearest neighbors by Euclidean distance are plotted in Figure 1 below.

3. An unattractive feature of straight-line distance when working with variables on different scales is that it equally weights the $(y_2 - y_1)$ and $(x_2 - x_1)$ distances. In this example, the x_1 variable is more spread out (with a variance of 10) than the variable x_2 (with a variance of 5). We would expect two points to differ more with respect to x_1 as opposed to x_2. Standardized Euclidean distance standardizes (z-scores) the two variables before calculating Euclidean distance. For illustration, the first 10 observations and their nearest neighbors by standardized Euclidean distance are plotted in Figure 1 below.

4. Mahalanobis distance goes further and accounts for the *covariance* that exists between x_1 and x_2. Unfortunately I have not found an easy Stata command like `nearest` that calculates Mahalanobis distance *and* finds the nearest neighbor. (Again `teffects nnmatch` will do this, but it requires an outcome and treatment variable to be provided). You will see my work-around in the do file using a loop and `mahascore`, which finds nearest neighbors one observation at a time. For illustration, the first 10 observations and their nearest neighbors by Mahalanobis are plotted in Figure 1 below.

   In Figure 1, nearest neighbors are often the same by each measure. One exception is observation #10, whose nearest neighbor is #15 by Mahalanobis distance and #69 by standardized Euclidean distance.

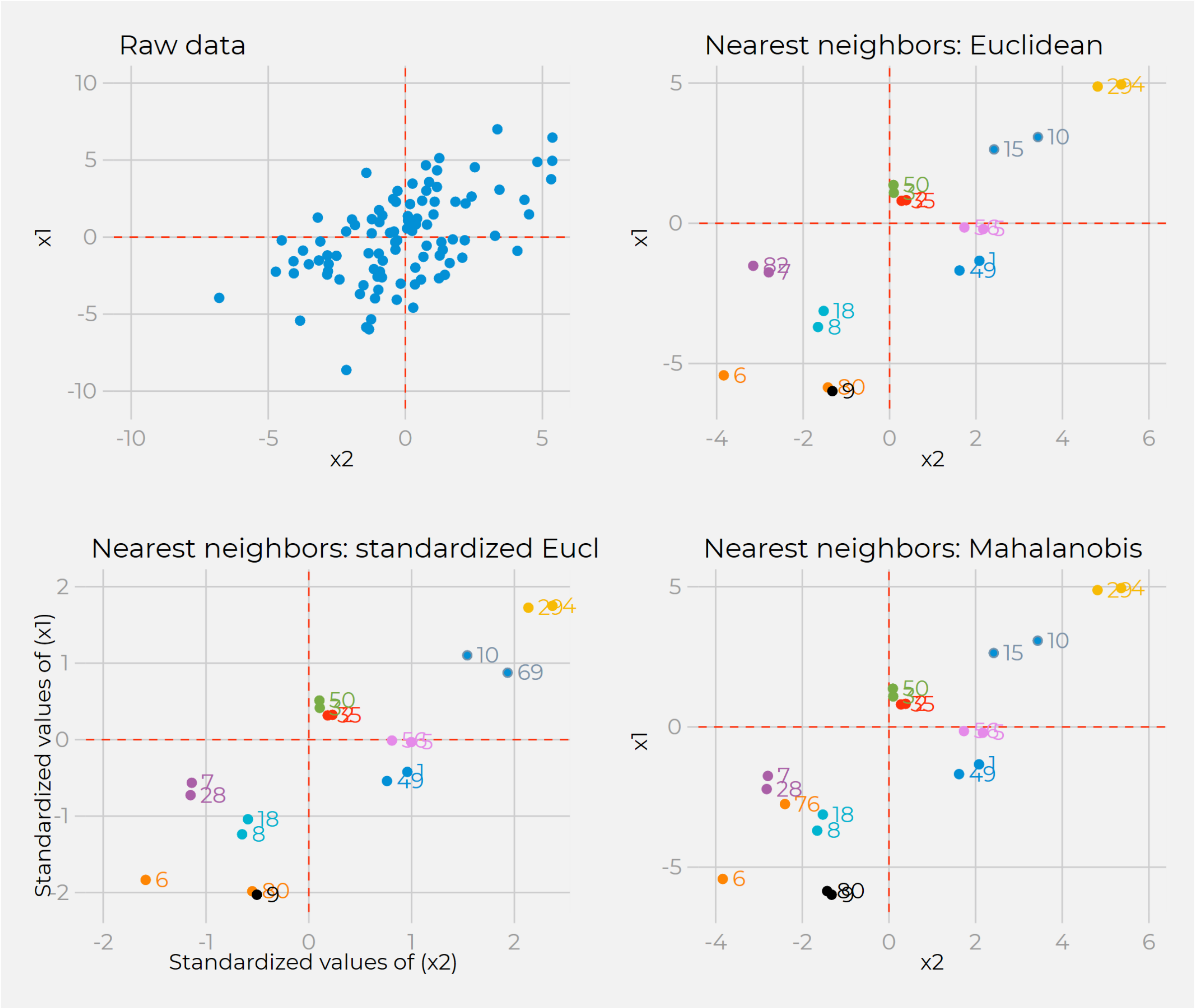

Figure 1: Raw data and 10 pairs of nearest neighbors

5. In many cases the nearest neighbor is the same using the three methods above. In the do file I created dummy variables that flagged differences in nearest neighbor IDs between the measures (below). The nearest neighbor by Euclidean distance differed from the nearest neighbor by standardized Euclidean distance in 14/100 cases. The nearest neighbor by Mahalanobis distance differed from the nearest neighbor by standardized Euclidean distance in 31/100 cases. Results may vary depending on the random seed used, since this is randomly generated data.

```
.           gen diffz=(idz~=id)

.           gen diffz2=(idz~=idm)
.         table diffz

----------------------
    diffz |      Freq.
----------+-----------
        0 |         86
        1 |         14
----------------------

.         table diffz2

----------------------
   diffz2 |      Freq.
----------+-----------
        0 |         69
        1 |         31
----------------------
```

6. The last part of the do file plots 10 cases where the neighbors *differ* depending on the distance measure used. Compare the plot that uses standardized Euclidean distance for these cases to the plot that uses Mahalanobis distance in Figure 2 below.

   By construction, x_1 and x_2 are positively correlated. What this means is that we expect a linear, positive association between them, as pictured in Figure 1. A higher value of x_2 leads us to predict a higher value of x_1. Straight-line distance ignores this systematic relationship and simply looks for the "closest" neighbor on the (x_1, x_2) grid. Mahalanobis distance, on the other hand, considers two data points consistent with the underlying correlation as "closer" than two data points with the same straight-line distance but inconsistent with the underlying correlation. As an example, take observation #12 in Figure 2. Its nearest neighbor by straight-line distance is #66—due in part to their similar values of x_2. However, its nearest neighbor by Mahalanobis distance is #97, which is both "close" and consistent with the underlying correlation between x_2 and x_1.

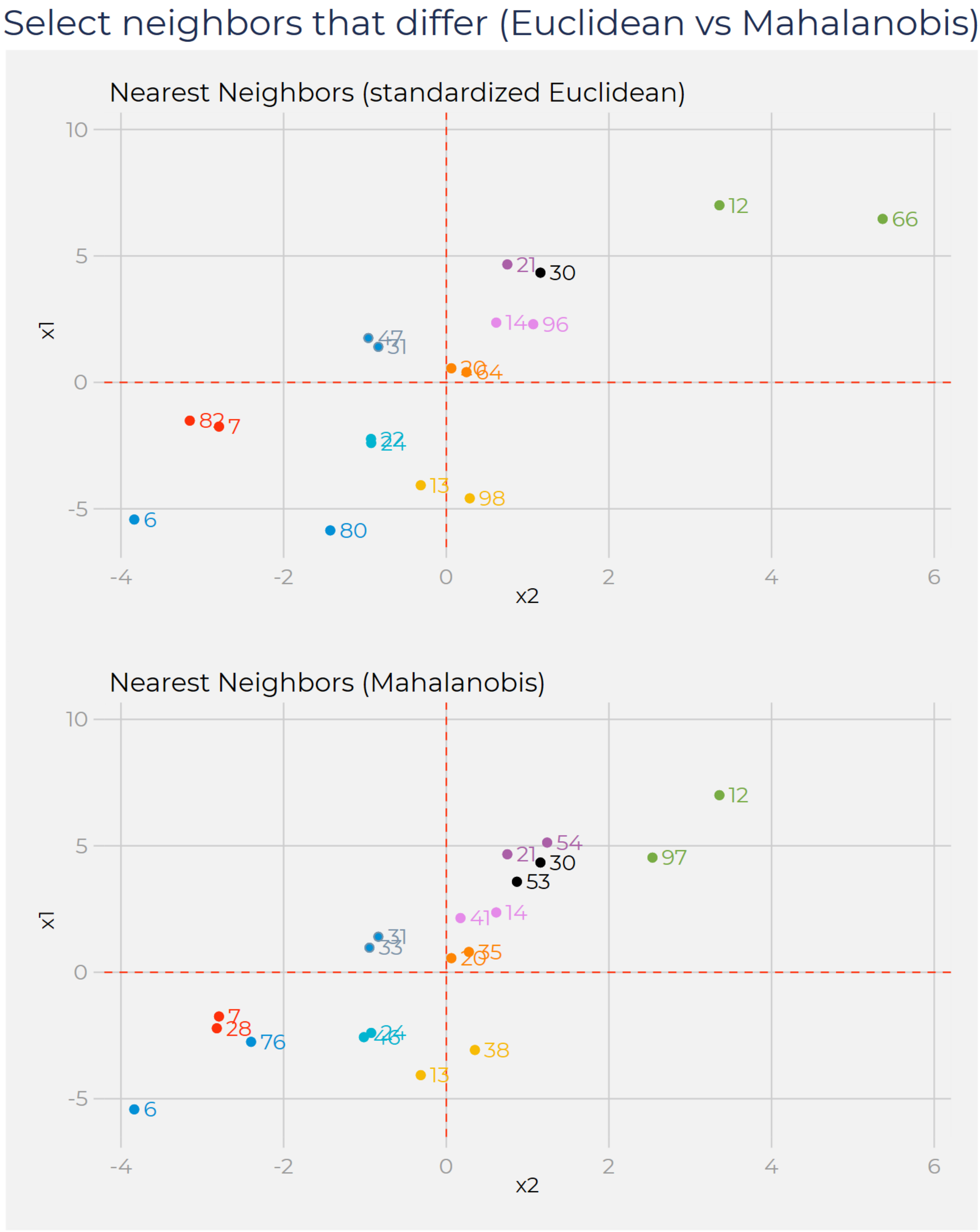

Figure 2: Cases where nearest neighbors differ by distance measure