**Q**: In GWR, why do we divide a predictor’s coefficient by the coefficient’s standard error to get a proxy for significance?

**A**: Recall that when we run **OLS** regression, the quantity has a t-distribution for each predictor . That is, if we were to take the estimated value of the coefficient , subtract its expected value *if* *the Null Hypothesis is true*, and divide that difference by the standard error of the coefficient, we would have a t-statistic, from which we can calculate the p-value.

However, what is the expected value of the coefficient if the Null Hypothesis is true (i.e., what is ? )

1. First, recall that the expected value of the coefficient is – that is, if we were to have an infinite number of samples of size *n* from the population, and calculate from each of them, then the distribution of all the ’s will simply be centered at the true population parameter . In other words, .

2. Recall that when we run regression, for each predictor , the Null Hypothesis takes on the form . That is, *if the Null Hypothesis is true*, .

This being the case, the quantity above reduces to , or simply , which means that the t-statistic can be obtained by simply dividing the estimated coefficient by an estimate of its standard error. The slides on OLS regression present formulas for calculating the ’s (using least squares) as well as the formulas for corresponding standard errors. In the example below, the coefficient for MEDHVALUE is -0.91 and the corresponding standard error is 0.066. If we calculate the quantity -0.91/0.066, we get -13.79, which is the t-statistic. From that t-statistic we can calculate the p-value.

----------------------------------------------------------------------

**Variable Coefficient Std.Error t-Statistic Probability**

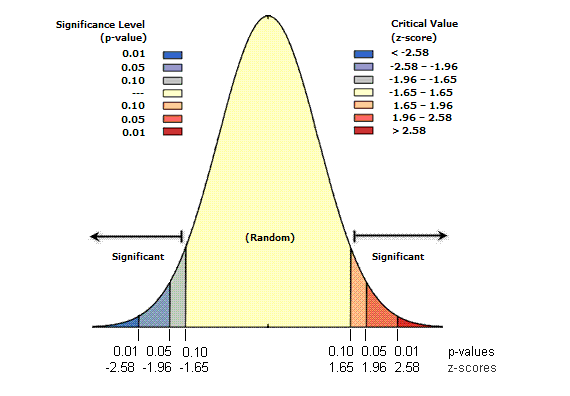
----------------------------------------------------------------------

CONSTANT 34.32 4.019 8.54 0.0000000

MEDHVALUE -0.91 0.066 -13.79 0.0000000

----------------------------------------------------------------------

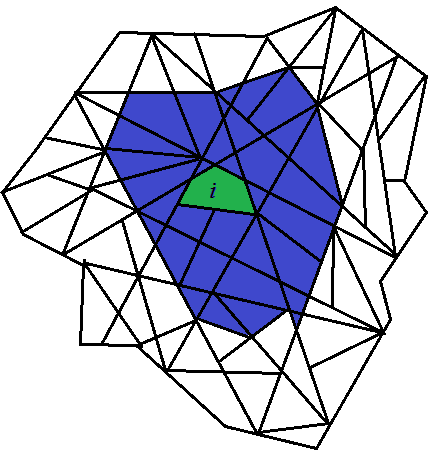
Also recall that when the sample size *n* is large (greater than 30), the *t*-distribution approaches the standard normal *z*-distribution, and the two may be used interchangeably. And for a *z*-distribution, values to the right of 1.96 (i.e., approximately 2) or to the left of -1.96 (i.e., approximately -2) correspond to p-values < 0.05. Also, the higher the *t* (or *z*) statistic (which again is calculated as the ratio between the coefficient and the standard error), the *lower* the p-value, and the more significant the predictor (See chart below). For instance, the t-statistic for the variable MEDHVALUE is calculated to be -13.79; this means that our estimate for MEDHVALUE is 13.79 standard deviations below 0, which is highly unlikely to occur *if*  is actually true and . Hence, we would reject for . If, however we saw that the t-statistic was between -2 and 2, then we couldn’t reject and say that is significantly different from 0.



In GWR, *t*-statistics and p-values are not reported due to the fact that we would likely be committing many type I (and type II errors). Recall that we typically set *α*, the probability of a type I error, to be 0.05 in our hypothesis tests *a priori*. This has very clear implications when we run GWR. As an example, imagine that we want to run a geographically weighted regression on a data set with 2,000 observations and 4 predictors. In this scenario, we would have to run 10,000 significance tests (2,000\*4 = 8,000 tests for the 4 predictors, and 2,000 additional tests for the constant (intercept) term). Assume, for the sake of example, that all of these 10,000 tests yield statistically significant results. However, the fact that we set *α* = 0.05, implies that 5% of these 10,000 tests, or a total of 500 tests, will have significant results simply due to chance (i.e., false positives)!

However, some statisticians still like to have a *proxy* for significance when they run GWR, even if the process is flawed. For that reason, they divide the estimated GWR coefficients by their estimated standard errors and get a quantity that would be similar to the t-statistic in OLS. If the absolute value of that ratio is low for location *i* (i.e., less than 2), then we are likely dealing with insignificant results at that location. However, the higher the absolute value of that ratio, the more likely it is that we are dealing with a statistically significant relationship between predictor *k* and the dependent variable at location *i*.

***Q*:** *How do we calculate local R2 and global R2**in GWR?*



Let’s start with local for location *i*. Imagine that we have a polygon shapefile as above, and that we are running the regression for block group *i*. All block groups *j* (in blue) are determined to be neighbors of *i*,and will be used in the regression along with *i* itself (block groups that are not shaded in blue or green are not neighbors of *i* and will not be used in this particular local regression for location *i*). Using only these blue or green block groups, we can estimate the . Once values are known, we can calculate and (which is simply ), as well as and (which are simply ). Said differently, using the , we can compute the predicted values of y and the residual for observation *i* and each observation *j* – i.e., all the observations that were used in this particular local regression. This being the case, we can calculate (which is the sum of squared residuals in all block groups *i* and *j*). We can also calculate (this is the sum of the squared differences between the value of y and its mean in all block groups *i* and *j*). Then, local . This same process is repeated for every block group in the data set, and the local is calculated for each location.

In English:

1. When we do the regression for block group *i*, we take all the shaded block groups (the green block group is *i* itself, and its neighbors are the blue block groups *j*) and run the regression using these shaded observations.
   1. For *each* of these shaded observations, we do the following:
      1. Compute by plugging in the values of and the independent variables
      2. Compute the residual for each shaded observation
      3. Square the residual and obtain
   2. We calculate SSE as the sum of the squared residuals (i.e., sum of for all the shaded observations)
   3. We calculate SST as the sum of the squared differences between the dependent variable and its average value in all the shaded observations (i.e., sum of for all the shaded observations)
   4. We calculate the local as , where SSE and SST are obtained in steps 1.b and 1.c above, respectively.

Calculating a global *R2* is different. As described in the paragraph above, when we run a local regression for each location *i*, we can calculate – that is, the residual for location *i* – using obtained from the regression that uses only observation *i* and its neighbors *j*. If we calculate for *every* location *i* using this approach, we can then calculate a global . We can also calculate a global SST, much like in OLS. Here, , where is the mean of *y* in all block groups in the data set. Then, the global .

In English:

1. When we do the regression for block group *i*, we take all the shaded block groups (the green block group is *i* itself, and its neighbors are the blue block groups *j*) and run the regression using these shaded observations.
   1. For *the green* observation (block group *i*), we do the following:
      1. Compute by plugging in the values of and the independent variables
      2. Compute the residual for the *green* observation (i.e., block group *i* **only**). We don’t care about the residuals in the blue block groups.
2. Repeat the process in (1) above for every single block group in the data set, calculating the residual in this manner.
3. We calculate SSE as the sum of these squared residuals (i.e., sum of for all observations in our entire data set)
4. We calculate SST as the sum of the squared differences between the dependent variable and its average value in the entire data set (i.e., sum of for all observations)
5. We calculate the global as , where SSE and SST are obtained in steps 3 and 4 above, respectively.