**Lab Assignment 05**

**I. Objectives**: The primary objective is to explore tests of statistical significance involving regression slope coefficients. You will perform tests of whether a slope coefficient equals 0 (the most common test) and whether two slope coefficients equal each other.

**II. Datasets**: *speeding\_tickets\_text.dta*

**III. Packages**: *readstata13*, *car*

**IV. Preparation**

1. Open RStudio by double-clicking the icon or selecting RStudio from the Windows Start menu.
2. Open the project from the Projects menu
3. Pull from Github for any updates
4. Navigate in the lower right file area to the Lab 3 section
5. 5) Open the R script by typing *Ctrl*+*O* or by clicking on File in the upper-left corner, using the dropdown menu, and navigating to the script in your working directory.
6. Line 1 will start the “here” library. If you have not installed that package, the code was the first line of Lab 3.
7. Line 2 installs a package to read the older version Stata file, then loads the library.

**V. Instructions for Lab 05**

You will use data on more than 30,000 traffic citations in Massachusetts, from Michael Makowsky and Thomas Stratmann’s 2009 article, “Political Economy at Any Speed: What Determines Traffic Citations?” *American Economic Review* 99: 509–527.

The dependent variable is *Amount*, which is the dollar amount of the fine on a moving violation.

Key independent variables are *Age*, the age of the offending driver, and *MPHover*, the miles per hour over the speed limit. There are also six dummy variables: *Female*, *Black*, *Hispanic*, *StatePol* (if the officer was a state patrolman), *OutTown* (if the driver resided outside the jurisdiction), and *OutState* (if the driver resided outside the state of Massachusetts).

The first question you will explore is how citations vary depending on the age of the driver and the speed at which he or she was driving.

Begin by loading the dataset, typing the following line, changing the directory if needed:

tickets <- read.dta13(here("data","speeding\_tickets\_text.dta"))

The data frame that you will start with is named tickets, however you should also run line 6 of the code to drop all the cases for which no dollar fine was recorded:

> complete <- subset(tickets, tickets$Amount!="NA")

Run line 8 to regress the citation amount (*Amount*) on *Age* and *MPHover*:

> basic <-lm(Amount ~ Age + MPHover, data=complete)

> summary(basic)

Run lines 9–10 to save two important features of the model. First is the vector of coefficients, which is a list of three numbers: the estimated intercept and the two estimated slope coefficients. Second is the ***variance-covariance matrix of the estimators***, which is a 3x3 square array; the first row (and column) is associated with the intercept,[[1]](#endnote-1) and the second and third rows (and columns) are associated with the two explanatory variables.

> vector.b <- basic$coefficients

> matrix.b <- vcov(basic)

Along the main diagonal of the matrix are the estimated variance of the estimated intercept and the estimated slope coefficients. The square roots of the entries along the main diagonal of the variance-covariance matrix of the estimators are standard errors of the regression intercept and coefficients. You can check this by running line 11, which calculates the square root of the entry in the second row and second column, which corresponds to the variable *Age*:

> sqrt(matrix.b[2,2])

Does the result of this line of code equal the standard error of the coefficient for *Age*?

Run lines 12–13 to generate the *t* statistics shown after typing summary(model1); you will be dividing an element of the vector of estimated coefficients by the square root of the corresponding variance of the estimated coefficient.

For example, to find the *t* statistic for the regression model coefficient for *Age*, type:

> t\_Age = vector.b[2]/sqrt(matrix.b[2,2])

To find the *t* statistic for the regression model coefficient for *MPHover*, type:

> t\_MPHover = vector.b[3]/sqrt(matrix.b[3,3])

In order to test for statistical significance, it is important to know the critical value. One option is to inspect Student’s *t* Table, if you have one handy. Another option is to use the “leftover” degrees of freedom from the model that generated the estimated coefficients and their variances.

Recall that the degrees of freedom for the *t* test of a regression coefficient is *n*–*k*–1, where *k* represents the number of regressors and *n* represents the sample size. For 95% confidence, we want to know the number of standard errors that separates the lower 97.5% of the *t* distribution with *n*–*k*–1 degrees of freedom from the upper 2.5%. We can find that number using line 14:

> qt(.975,basic$df.residual)

Another approach to testing the statistical significance (versus a null hypothesis that *β* = 0) is to examine whether the confidence interval excludes 0. To view the confidence intervals, type the following code, as shown in line 15:

> confint(basic)

The *Age* variable does not have a statistically significant effect, and we will drop it from the rest of the analysis, even though there’s no harm to including it. Run line 17 of the R script to estimate a model with *MPHover* plus six dummy variables, indicating the driver’s race, ethnicity, gender, and residential status, as well as the officer’s employer. Run line 18 to view the estimates.

Sometimes we are interested in whether two slope coefficients are equal. For example, are residents of other states fined more than residents of other towns in Massachusetts? We can generate a *t* statistic for a test of the equality of coefficients for *OutTown* and *OutState*; if *t* is less than the appropriate critical value, then we retain the null hypothesis, which means that we do not have enough evidence to say that the coefficients have different values in the population.

Run lines 19–20 to save the vector of estimated coefficients (which has eight entries including the intercept) and the ***variance-covariance matrix of the estimators***, which is a square array with eight rows and eight columns.

> vector.d <- dummies$coefficients

> matrix.d <- vcov(dummies)

Along the main diagonal of the matrix are the estimated variances of the estimated intercept and the estimated slope coefficients.

The numerator of the equality-of-coefficients *t* statistic is the difference between the coefficients. To compare *OutTown* to *OutState*, you need the sixth and seventh entries in the vector:

> num = (vector.d[7] – vector.d[6])

The denominator of this test statistic is the standard error of the difference of coefficients, which adds the variance of one coefficient plus the variance of the other coefficient, and then subtracts two times the covariance of the two coefficients, and then takes the square root:

> denom = sqrt(matrix.d[7,7] + matrix.d[6,6] - 2\*matrix.d[7,6])

The *t* statistic will be the difference of coefficients divided by its standard error:

> t\_eq = num/denom

The above lines of code are shown in lines 21–23.

Another way of establishing that two coefficients are unequal begins by inspecting confidence intervals. You can run the code shown in line 24 to display all the 95% confidence intervals; if two confidence intervals do not intersect then it suggests the variables have different effects.

On your own (or with Tom’s assistance), repeat the process and examine whether race and gender have the same effects. Can you reject the null hypothesis that the coefficient for *Black* is equal to the coefficient for *Female*?

Line 27 shows code for an *F* test of this hypothesis. As we will cover this afternoon’s lecture, the *F* statistic for a test such as this equals the *t* statistic squared. If you correctly carry out a *t* test of the equality of coefficients, and square the result, you should obtain the same number as the *F* statistic. If you want to see that the *t* test and *F* test are equivalent, you should give your *t* statistic a name, give your degrees of freedom a name, and use those to find the *p* values as follows:

> t\_bf = num\_bf/denom\_bf

> df <- summary(dummies)$df[2]

> 2\*(1-pt(abs(t\_bf),df))

In order to examine the measures of model fit, you will want to collect some information from this model by running lines 29–34. First, line 29 will show you the degrees of freedom; the first number is the degrees of freedom consumed by the model (i.e., *k* + 1), and the second number is the degrees of freedom left over for the residual (i.e., *n – k –* 1). You can save this latter number and name it dfr by running line 30.

Next, line 31 will show you the *F* statistic for the full model, which we will discuss next week, and repeats the degrees of freedom – although the number is one less than before because it does not subtract one for the intercept (because the null model, which has *only* an intercept – the sample mean value of *y* – has seven *more* degrees of freedom). Line 32 will show you the R-squared and adjusted R-squared.

To examine the ANOVA table, which will play a role in considering model specification next week, run the following lines of code, which are lines 34–38 in the R script:

> anovatable = anova(dummies); anovatable[2]

> ssr<-sum(dummies$residuals^2)

> sst<-sum((complete$Amount-mean(complete$Amount))^2)

> sse<-sum((predict(dummies) - mean(complete$Amount))^2)

> sst-sse

You can check that the first seven Sum Sq entries in the ANOVA table sum to equal the number that R calculates with the code in line 37; this is the explained sum of squares, or the model sum of squares. You can also subtract the number reported by line 37 from the number reported by line 36 (as shown in the last line above) and check that it equals the Residuals Sum Sq in the ANOVA table, which also equals the amount reported by line 35.

Then, write down the following values:

Total Sum of Squares (line 36) \_\_\_\_\_\_\_\_\_\_

Explained Sum of Squares (line 37) \_\_\_\_\_\_\_\_\_\_

Residual Sum of Squares (line 35) \_\_\_\_\_\_\_\_\_\_

Model R2 (line 32) \_\_\_\_\_\_\_\_\_\_

Number of Observations [*n*] \_\_\_\_\_\_\_\_\_\_

Number of Explanatory Variables [*k*] \_\_\_\_\_\_\_\_\_\_

Residual Degrees of Freedom (line 30, second entry) \_\_\_\_\_\_\_\_\_\_

When you type summary(dummies), R automatically gives the F-statistic for the “overall significance of a regression” test; you will find it listed in the last line of code of regression results (*F* = 4759, or 4758.8). R notes that it is distributed with 7 numerator degrees of freedom (representing the *k* = 7 explanatory variables) and 31666 denominator degrees of freedom (representing *n* – *k* – 1).

You should be able to calculate this number yourself. The equation is:

*F* =

Of more immediate importance, you should be able to calculate the Multiple R2 from the model:

You should also be able to calculate the Adjusted R2 from the model using this equation:

1 – [1–*R*2]

To clear the **Environment**, type rm(list=ls()) or click on the broom icon.

To clear the **Console** window, type Ctrl-*l*

1. Suppose you wanted to inspect the dependent variable with no independent variables, then you could estimate a “null model” that has no explanatory variables. Whereas we usually estimate the regression line E(*y* | *x*), we are just looking at *E*(*y*) when there are no *x*. The following three results should be observed:

   1. The intercept ought to equal the mean value of *y*. (There is no slope.)

   2. The residual standard error should equal the standard deviation of *y*.

   3. The standard error of the intercept ought to equal the standard deviation of *y* divided by the square root of *n*–1, which would also be equal to the standard error of .

   Run the following lines of code to verify these properties:

   > null <- lm(complete$Amount ~ 1); summary(null)

   > mu.y = mean(complete$Amount); mu.y

   > sd(complete$Amount)

   > sey = sd(complete$Amount)/sqrt(null$df.residual); se.y

   > t\_crit = qt(.975, null$df.residual); t\_crit

   > mu.y – t\_crit\*se.y; mu.y + t\_crit\*se.y

   > confint(null) [↑](#endnote-ref-1)