Higher-Order Terms

Brenton Kenkel — PSCI 8357
February 4, 2016

In the Reintroduction to Linear Regression, we talked about the linear model,

$$Y_i = x_i^\top \beta + \epsilon_i,$$

and how to estimate its parameters β via ordinary least squares. It is not necessary that the model be linear in the covariates. The OLS estimator retains its nice properties as long as the model is linear in the parameters. For example,

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \epsilon_i$$

is linear in the parameter vector β but not in the covariate x_{i1}. The OLS estimator of β is still unbiased, consistent, and all that jazz. But if your model is a nonlinear function of the parameters, such as

$$Y_i = \beta_0 + x_{i1}^{\beta_1} + x_{i2}^{\beta_2} + \epsilon_i,$$

then the OLS estimator is no good. For a model like this, you would want to derive an alternative estimator using maximum likelihood or another technique beyond the scope of this course.

Today, we will focus on models of the first variety—those that are linear in the parameters but not the covariates. We will talk mainly about models that are *polynomial* functions of the covariates, of which interactive and quadratic functions are special cases. We will cover:

- How to interpret the results of polynomial models.
- Why you should never leave out lower-order terms when including higher-order terms.
- How to calculate standard errors and confidence intervals on estimated marginal effects.
- How to test hypotheses about nonlinear relationships.
- Modeling nonlinearities of unknown form.

Specification and Interpretation

Let us begin by defining the quantity of interest. For the moment, assume we are still in the world where the regression function is linear in the covariates as well as the parameters. The coefficient on the j'th covariate is then equivalent to the partial derivative of the regression function with respect to that covariate:

$$\frac{\partial E[Y_i \mid x_i]}{x_{ij}} = \beta_j.$$

As shorthand, I will call this derivative the *marginal change in conditional expectation*, or MCCE,[1] with respect to the covariate x_{ij}. People usually call this the *marginal effect* of x_{ij}, but that has a causal connotation that is not always appropriate.

If the regression function is nonlinear in the covariates, then the MCCEs are no longer constant, and the estimated coefficients can no longer be interpreted on their own. For example, consider a model that is a quadratic function of a single covariate, W_i:

$$Y_i = \beta_0 + \beta_1 W_i + \beta_2 W_i^2 + \epsilon_i.$$

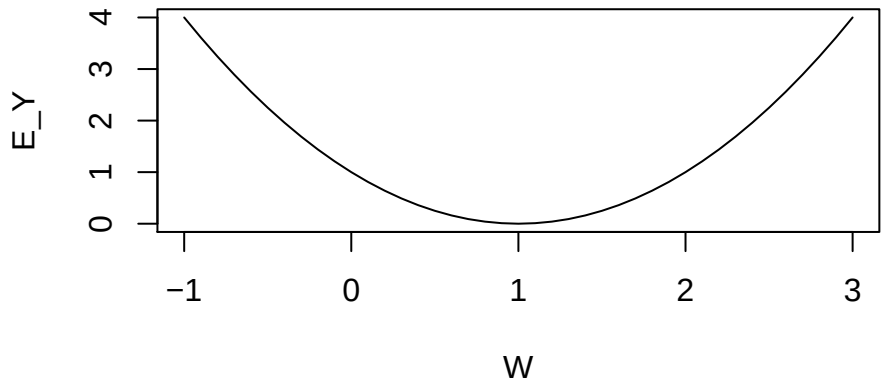
The MCCE changes depending on the value of W_i:

$$\frac{\partial E[Y_i \mid W_i]}{\partial W_i} = \beta_1 + 2\beta_2 W_i.$$

Let's look at the conditional expectation function with $\beta = (1, -2, 1)$.

```
W <- seq(-1, 3, by = 0.1)
E_Y <- 1 - 2 * W + W^2
plot(W, E_Y, type = "l")
```

[1] MCCE is my own terminology, so don't expect people to be familiar with the acronym if you mention changes in conditional expectation in the papers you write.



The MCCE is negative for $W < 1$ and positive for $W > 1$, and its magnitude grows as W gets farther away from 1.

The interactive model is similar. Consider the regression function

$$Y_i = \beta_0 + \beta_W W_i + \beta_Z Z_i + \beta_{WZ} W_i Z_i + \epsilon_i.$$

The MCCEs with respect to W_i and Z_i are

$$\frac{\partial E[Y_i \mid W_i, Z_i]}{\partial W_i} = \beta_W + \beta_{WZ} Z_i,$$
$$\frac{\partial E[Y_i \mid W_i, Z_i]}{\partial Z_i} = \beta_Z + \beta_{WZ} W_i.$$

So now the MCCE of each variable depends on the value of the other—we are modeling conditional effects. In the special case where Z_i is a binary variable, we have the *varying slopes* model:

$$\frac{\partial E[Y_i \mid W_i, Z_i = 0]}{\partial W_i} = \beta_W,$$
$$\frac{\partial E[Y_i \mid W_i, Z_i = 1]}{\partial W_i} = \beta_W + \beta_{WZ}.$$

When you estimate an interactive model, you should always include the lower-order terms in the regression. In other words, if you include $W_i Z_i$, then you should also include W_i and Z_i individually.

To see why, suppose you were to include W_i and $W_i Z_i$ but not Z_i. This amounts to fixing $\beta_Z = 0$, so the MCCE with respect to Z_i is

$$\frac{\partial E[Y_i \mid W_i, Z_i]}{\partial Z_i} = \beta_{WZ} W_i.$$

Consequently, the MCCE with respect to Z_i is a line through the origin: it equals zero whenever $W_i = 0$. This is a restriction that is never sensible to impose—so don't do it.

Let's run through a couple of examples of calculating MCCEs to interpret the results. We'll be using data from the **car** package, which also has some handy hypothesis testing tools that we'll use later today.

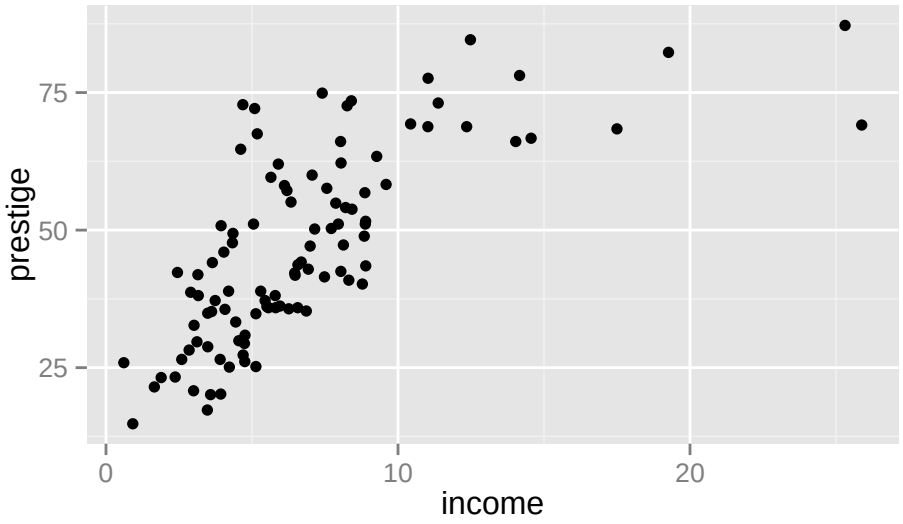
```
library("car")
library("dplyr")
library("ggplot2")
```

We will use the `Prestige` dataset, which records the perceived prestige of various jobs in Canada in the early 1970s. To begin with, we'll model the relationship between a profession's income and its prestige.

```
data(Prestige)
head(Prestige)
```

```
##                     education income women prestige census type
## gov.administrators       13.1  12351 11.16     68.8   1113 prof
## general.managers         12.3  25879  4.02     69.1   1130 prof
## accountants              12.8   9271 15.70     63.4   1171 prof
## purchasing.officers      11.4   8865  9.11     56.8   1175 prof
## chemists                 14.6   8403 11.68     73.5   2111 prof
## physicists               15.6  11030  5.13     77.6   2113 prof
```

```
Prestige <- mutate(Prestige, income = income / 1000)
ggplot(Prestige, aes(x = income, y = prestige)) +
    geom_point()
```



Notice that the conditional expectation function does not appear perfectly linear. After about $10,000, higher incomes do not appear to be associated with higher prestige. We will try to capture this with a quadratic model.[2]

When fitting quadratic models, people accustomed to Stata like to create a new "squared" variable in their data frame and then include it in the regression formula. Don't do that. Instead, include `I(x^2)` as a term in your regression formula, where `x` is the name of the variable you want to include.

```
fit_quadratic <- lm(prestige ~ income + I(income^2), data = Prestige)
fit_quadratic
```

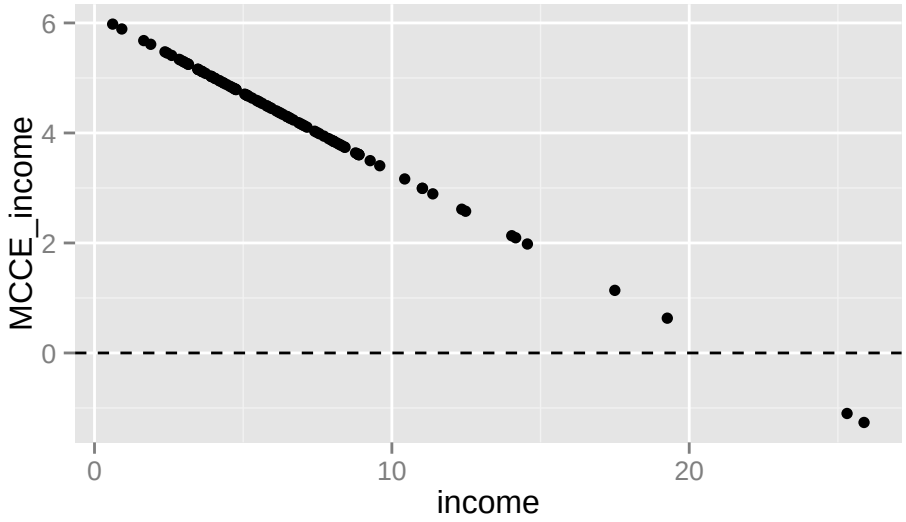
```
##
## Call:
## lm(formula = prestige ~ income + I(income^2), data = Prestige)
##
## Coefficients:
## (Intercept)       income  I(income^2)
##      14.183        6.154       -0.143
```

Let's interpret the results by calculating the MCCE with respect to income.

```
beta_income <- coef(fit_quadratic)["income"]
beta_income_2 <- coef(fit_quadratic)["I(income^2)"]
```

[2] A piecewise linear model with a breakpoint around $10,000 of income is probably more sensible, but it wouldn't help us estimate and interpret quadratic models.

```
Prestige <- Prestige %>%
    mutate(MCCE_income = beta_income + 2 * beta_income_2 * income)
ggplot(Prestige, aes(x = income, y = MCCE_income)) +
    geom_point() +
    geom_hline(yintercept = 0, linetype = 2)
```



For very low-earning jobs, we would expect an extra $1,000 in income to correspond to about a 6-point increase in the prestige score. For middle-income jobs, around $10,000, an extra $1,000 would be associated with a 3- or 4-point increase. For extremely high-earning jobs, we estimate that further income is actually associated with lower prestige.

Now let's look at an interactive model. We will now model prestige as a function of education and income, including the interaction between them.

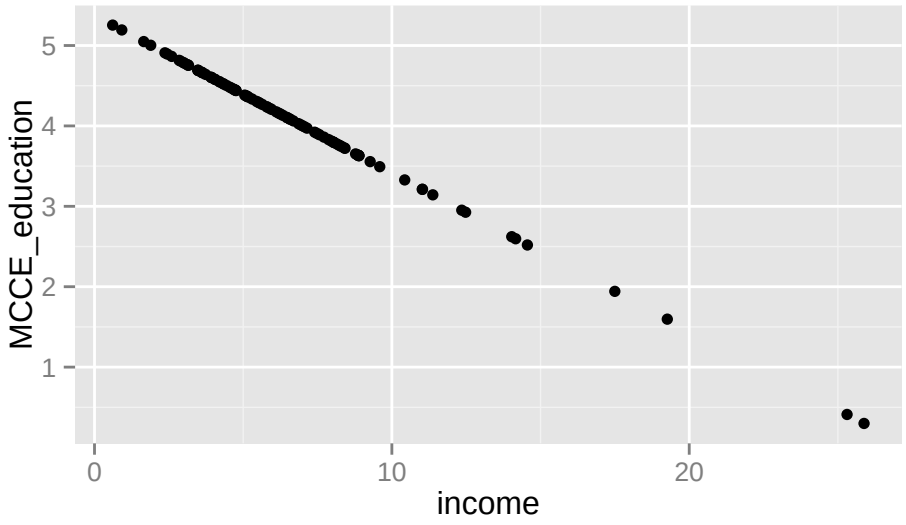
```
fit_interaction <- lm(prestige ~ education * income, data = Prestige)
fit_interaction
```

```
##
## Call:
## lm(formula = prestige ~ education * income, data = Prestige)
##
## Coefficients:
##      (Intercept)         education            income
##          -22.070             5.373             3.944
```

```
## education:income
##           -0.196
```

The coefficients here are a little easier to interpret than in the quadratic model. We can see that the MCCE with respect to education decreases with income. Let's visualize education's MCCE as a function of income.

```
beta_education <- coef(fit_interaction)["education"]
beta_education_income <- coef(fit_interaction)["education:income"]
Prestige <- Prestige %>%
    mutate(MCCE_education = beta_education + beta_education_income * income)
ggplot(Prestige, aes(x = income, y = MCCE_education)) +
    geom_point()
```



So, suppose we were to compare two professions, both making almost no money, where one requires an additional year of education on average. We would expect the more-educated profession to have a prestige score about 5 points higher. But if we compared two professions with a one-year difference in education, both making about $15,000, we would expect the more-educated one to score only about 2.5 points higher in terms of prestige.

Inference

There are two kinds of hypotheses we might want to test with a quadratic or interactive model.

1. That the MCCE of W_i is zero at a particular value of W_i (for a quadratic model) or of Z_i (for an interactive model).
2. That the MCCE of W_i is *always* zero.

In a standard linear model with no higher-order terms, each MCCE is constant, so these two hypotheses are equivalent. Not so in higher-order models.

Pointwise Hypotheses

To tackle the first type of hypothesis—the pointwise one—we will derive the standard error of the MCCE at a particular point. We will use the following formula for the variance of the weighted sum of two variables, where A and B are random variables and c_1 and c_2 are real-valued constants:

$$V[c_1A + c_2B] = c_1^2V[A] + c_2^2V[B] + 2c_1c_2\text{Cov}[A, B]$$

We will start with the standard error of the MCCE of a variable included with its quadratic term. Remember that the formula for the MCCE is $\beta_1 + 2\beta_2W_i$, where β_1 and β_2 are the coefficients on the main term and the quadratic term, respectively. Its variance is therefore

$$V[\beta_1 + 2\beta_2W_i] = V[\beta_1] + 4W_i^2V[\beta_2] + 4W_i\text{Cov}[\beta_1, \beta_2]$$

In R, we can retrieve our estimates of these variances by running `vcov()` on a fitted model object. Let's calculate the standard error of the estimated MCCE for each observation in the `Prestige` data, along with the associated confidence intervals.

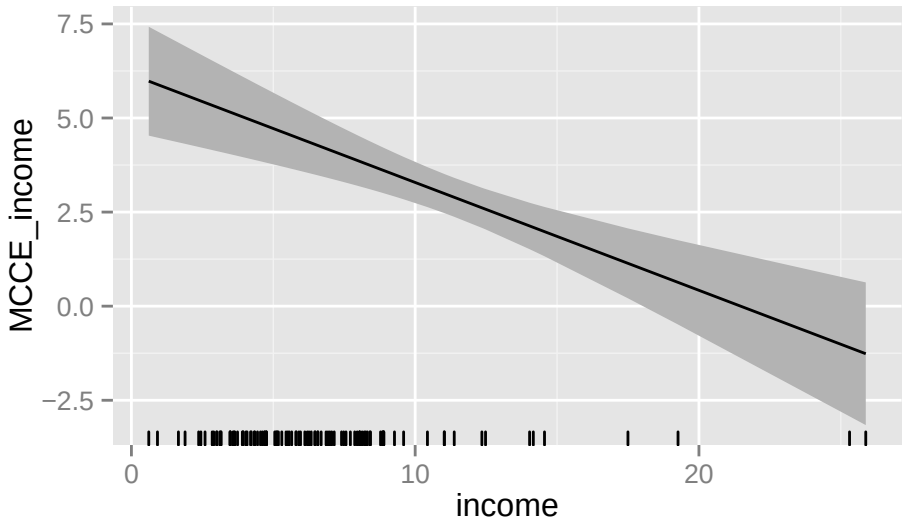
```
vcov(fit_quadratic)
```

```
##             (Intercept)  income I(income^2)
## (Intercept)     12.3585 -2.4828    0.089212
## income          -2.4828  0.5765   -0.022422
## I(income^2)      0.0892 -0.0224    0.000987
```

```
var_income <- vcov(fit_quadratic)["income", "income"]
var_income_2 <- vcov(fit_quadratic)["I(income^2)", "I(income^2)"]
covar_income_2 <- vcov(fit_quadratic)["income", "I(income^2)"]
Prestige <- Prestige %>%
    mutate(var_MCCE = var_income +
               4 * income^2 * var_income_2 +
               4 * income * covar_income_2,
           se_MCCE = sqrt(var_MCCE),
           lower = MCCE_income - 2 * se_MCCE,
           upper = MCCE_income + 2 * se_MCCE)
```

Now we can plot the estimated MCCE and its confidence interval as a function of income.

```
ggplot(Prestige, aes(x = income, y = MCCE_income)) +
    geom_ribbon(aes(ymin = lower, ymax = upper), fill = "gray70") +
    geom_line() +
    geom_rug(sides = "b")
```



For an interactive model, the formula is slightly different. In that case, the MCCE with respect to W_i depends on the value of Z_i: $\beta_W + \beta_{WZ} Z_i$. This gives us a variance of

$$V[\beta_W + \beta_{WZ} Z_i] = V[\beta_W] + Z_i^2 V[\beta_{WZ}] + 2 Z_i \text{Cov}[\beta_W, \beta_{WZ}]$$

Another approach to estimating the standard error of the MCCE at a particular

point is by simulation (King, Tomz, and Wittenberg 2000). The simulation-based approach isn't necessary in this case since we have a formula, but it carries over nicely for more complex models (the kind you'll see in Stat III) where formulas are hard to derive. To estimate the standard error of the MCCE at W_i by simulation, we will:

1. Draw $m = 1, \ldots, M$ values of $\tilde{\beta}_m$ from the estimated sampling distribution, a normal distribution with mean $\hat{\beta}$ and variance matrix $\hat{\Sigma}$.
2. For each $\tilde{\beta}_m$, calculate the associated MCCE at W_i:

$$\tilde{\beta}_{m1} + 2\tilde{\beta}_{m2} W_i.$$

3. Take the standard deviation of the M simulated MCCEs.

Let's do this for the income level associated with the first observation in the `Prestige` data. We'll fire up the packages we need to draw from a multivariate normal distribution and run loops.

```
library("foreach")
library("MASS")
```

We'll draw $M = 100$ values of $\tilde{\beta}_m$ from the estimated sampling distribution.

```
n_sim <- 100
beta_sim <- mvrnorm(n_sim,
                    mu = coef(fit_quadratic),
                    Sigma = vcov(fit_quadratic))
head(beta_sim)
```

```
##      (Intercept) income I(income^2)
## [1,]        13.1   6.43     -0.1561
## [2,]        10.4   6.93     -0.1790
## [3,]        20.4   4.87     -0.0878
## [4,]        16.3   5.48     -0.1094
## [5,]        17.8   5.42     -0.1106
## [6,]        12.3   6.49     -0.1541
```

Now, for each of these, we'll calculate and save the associated MCCE.

```
MCCE_dist <- foreach (i = 1:n_sim, .combine = "c") %do% {
    beta_sim[i, "income"] +
        2 * beta_sim[i, "I(income^2)"] * Prestige[1, "income"]
}
```

Let's take the standard deviation of the simulation and compare it to the one we calculated analytically.

```
sd(MCCE_dist)
```

```
## [1] 0.217
```

```
Prestige[1, "se_MCCE"]
```

```
## [1] 0.266
```

Not too far off! Again, this wasn't the best way to do it—there's no reason to simulate what we can find with a simple formula—but it generalizes to more complex models (or more complex functions of the coefficients) better than the formulaic method. The simulation approach is also what the very helpful **interplot** package uses to calculate confidence intervals in automatically generating plots like the one we made of `MCCE_income`.

Global Hypotheses

For better or worse, political scientists usually care mainly about null hypotheses of the form, loosely speaking, "This variable ain't got nothin' to do with that variable." The pointwise calculations we just did don't speak to hypotheses of this form.

For a quadratic model, the null hypothesis that W_i has nothing to do with the response can be stated as

$$\text{H0: } \beta_1 = 0 \text{ and } \beta_2 = 0$$

For an interactive model, the null hypothesis that W_i has nothing to do with the response can be stated as

$$\text{H0: } \beta_W = 0 \text{ and } \beta_{WZ} = 0$$

Both of these are *linear hypotheses*. A linear hypothesis is a hypothesis of the form

$$\mathbf{R}\beta = q,$$

where $\mathbf{R}$ is an $r \times p$ matrix and q is an $r \times 1$ vector. For example, the matrix

version of the null hypothesis for the quadratic model would be

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The standard test for linear hypotheses like this is the *Wald test* (Greene 2003, 6.3.1). The test statistic is

$$W = (\mathbf{R}\hat{\beta} - q)^\top (\mathbf{R}\hat{\Sigma}\mathbf{R}^\top)^{-1} (\mathbf{R}\hat{\beta} - q).$$

Under the null hypothesis, the asymptotic distribution of W is χ^2 with r degrees of freedom.

The Wald test is not just an aggregation of the individual Z (or t) tests of the coefficients. Two coefficients might each individually be statistically insignificant, yet the Wald test may lead us to reject the null hypothesis that both are zero. Conversely, one of a group of coefficients might be statistically significant, and yet the Wald test may not have us reject the null hypothesis that all are zero.

As an instructive exercise, we will implement the Wald test ourselves in R. Then we will see the easy way to do it.

```
R <- rbind(
    c(0, 1, 0),
    c(0, 0, 1)
)
q <- c(0, 0)
beta_hat <- coef(fit_quadratic)
Sigma_hat <- vcov(fit_quadratic)

Rbq <- R %*% beta_hat - q
RSR <- R %*% Sigma_hat %*% t(R)
W <- t(Rbq) %*% solve(RSR) %*% Rbq
W
```

```
##      [,1]
## [1,]  146
```

```r
pchisq(W, df = nrow(R), lower.tail = FALSE)
```

```
##          [,1]
## [1,] 1.91e-32
```

The easy way entails using the `linearHypothesis()` function from the **car** package. You can supply the matrix of restrictions $\mathbf{R}$ and the vector of values q directly to the function.

```r
linearHypothesis(fit_quadratic,
                 hypothesis.matrix = R,
                 rhs = q,
                 test = "Chisq")
```

```
## Linear hypothesis test
##
## Hypothesis:
## income = 0
## I(income^2) = 0
##
## Model 1: restricted model
## Model 2: prestige ~ income + I(income^2)
##
##   Res.Df   RSS Df Sum of Sq Chisq Pr(>Chisq)
## 1    101 29895
## 2     99 12077  2     17819   146     <2e-16
```

Or, even easier, you can just write a vector of strings expressing your hypothesis in terms of the relevant coefficient names. Let's now test a composite null hypothesis for education in the interactive model.

```r
linearHypothesis(fit_interaction,
                 c("education = 0", "education:income = 0"),
                 test = "Chisq")
```

```
## Linear hypothesis test
##
## Hypothesis:
## education = 0
## education:income = 0
##
```

```
## Model 1: restricted model
## Model 2: prestige ~ education * income
##
##   Res.Df   RSS Df Sum of Sq Chisq Pr(>Chisq)
## 1    100 14616
## 2     98  5641  2      8975   156     <2e-16
```

The Wald test can be used even for standard linear models. For example, imagine that we model the response as a function of a categorical variable by including dummy variables for each category. The null hypothesis that the variable has no association with the expected value of the response is equivalent to the coefficient on each dummy variable being zero.

Let's do an example with a 20-category variable, where the null hypothesis is true.

```
x <- rnorm(1000)
y <- 1 - x + rnorm(1000)
w <- sample(letters[1:20], size = 1000, replace = TRUE)
fit_dummy <- lm(y ~ x + w)
summary(fit_dummy)
```

```
##
## Call:
## lm(formula = y ~ x + w)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.389 -0.667 -0.008  0.687  3.171
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1743     0.1340    8.76   <2e-16
## x            -1.0176     0.0341  -29.88   <2e-16
## wb           -0.3131     0.1964   -1.59    0.111
## wc           -0.0998     0.1956   -0.51    0.610
## wd           -0.1747     0.1954   -0.89    0.372
## we           -0.1155     0.1944   -0.59    0.553
## wf           -0.1390     0.2006   -0.69    0.489
## wg           -0.4219     0.2128   -1.98    0.048
## wh           -0.2894     0.2450   -1.18    0.238
```

```
## wi           -0.3446     0.1964   -1.75    0.080
## wj           -0.0647     0.2056   -0.31    0.753
## wk           -0.4119     0.2198   -1.87    0.061
## wl           -0.1354     0.2055   -0.66    0.510
## wm           -0.2089     0.1902   -1.10    0.272
## wn           -0.2168     0.1927   -1.13    0.261
## wo           -0.1114     0.2042   -0.55    0.585
## wp           -0.2388     0.1887   -1.27    0.206
## wq           -0.0501     0.2055   -0.24    0.807
## wr           -0.1517     0.1919   -0.79    0.429
## ws           -0.1819     0.2056   -0.89    0.376
## wt           -0.0338     0.1936   -0.17    0.861
##
## Residual standard error: 1.05 on 979 degrees of freedom
## Multiple R-squared:  0.485,  Adjusted R-squared:  0.474
## F-statistic: 46.1 on 20 and 979 DF,  p-value: <2e-16
```

```
hypotheses <- names(coef(fit_dummy))
hypotheses <- setdiff(hypotheses, c("(Intercept)", "x"))
hypotheses <- paste(hypotheses, "= 0")
linearHypothesis(fit_dummy,
                 hypotheses,
                 test = "Chisq")
```

```
## Linear hypothesis test
##
## Hypothesis:
## wb = 0
## wc = 0
## wd = 0
## we = 0
## wf = 0
## wg = 0
## wh = 0
## wi = 0
## wj = 0
## wk = 0
## wl = 0
## wm = 0
## wn = 0
```

```
## wo = 0
## wp = 0
## wq = 0
## wr = 0
## ws = 0
## wt = 0
##
## Model 1: restricted model
## Model 2: y ~ x + w
##
##   Res.Df  RSS Df Sum of Sq Chisq Pr(>Chisq)
## 1    998 1083
## 2    979 1071 19      12.8  11.7        0.9
```

References

Greene, William H. 2003. *Econometric Analysis*. 5th ed. Prentice Hall.

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." *American Journal of Political Science* 44 (2): 341–55. http://gking.harvard.edu/files/making.pdf.