BIOS7345 Lab 5

Testing coefficients

Sarah Lotspeich

5 October 2018

Introduction

A scientist named Dr. Bosenberg wanted to estimate the puncture distance for an epidural in children by weight. He regressed skin-to-epidural distance (`SED`) on weight (`WT`) and obtained a regression line.

Fitting models

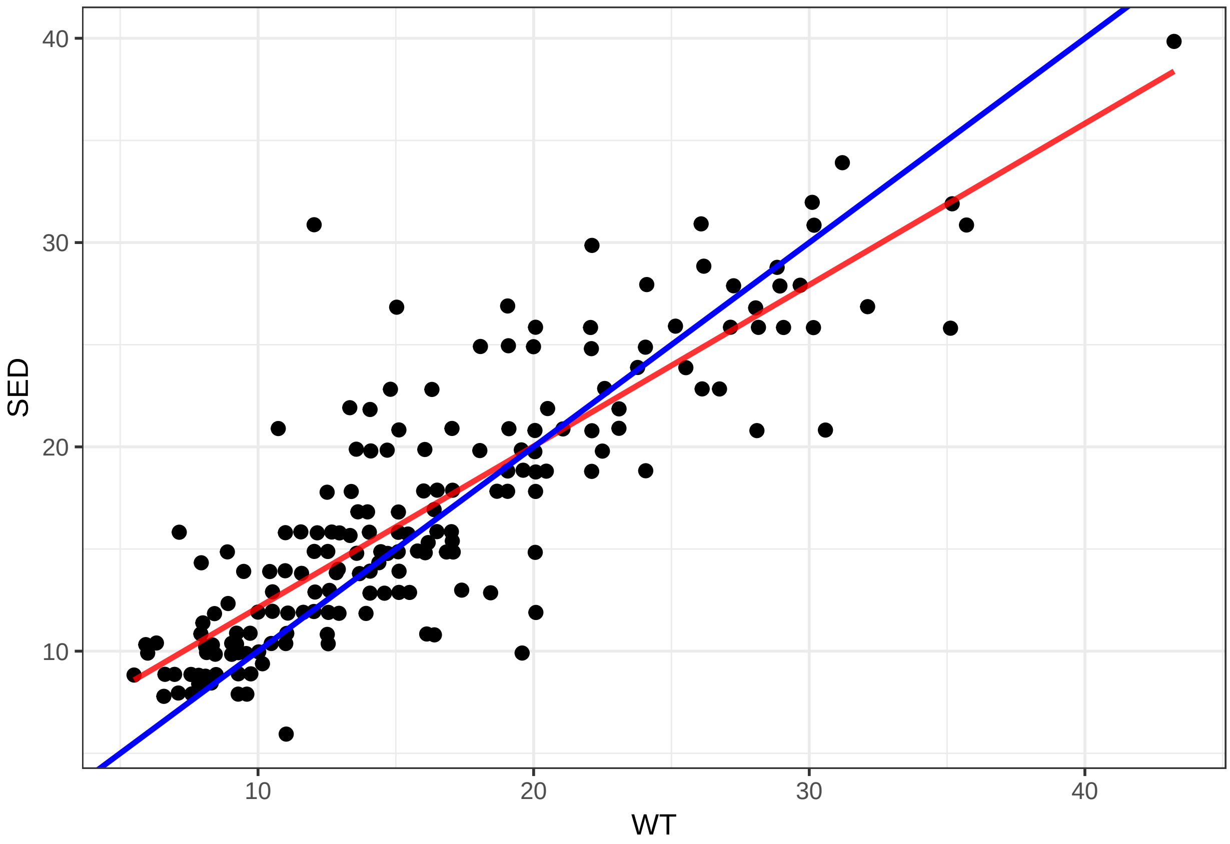
```
# read in the data
dat <- read.csv("https://raw.githubusercontent.com/sarahlotspeich/BIOS7345_Labs/master/Bios7345lab5.csv
    header = TRUE, stringsAsFactors = FALSE)

# fit the model for SED ~ WT
mod <- ols(SED ~ WT, data = dat)
```

Create a plot of this regression line over the data. Begin by writing a simple function to predict `SED` for a given `WT` based on your `mod` coefficients.

```
# write function to predict SED from WT
est_SED <- function(x) return(mod$coefficients["Intercept"] + mod$coefficients["WT"] *
    x)

# create a plot of the data
my_plot <- dat %>% ggplot() + geom_point(aes(x = WT, y = SED), size = 2) + theme_bw()

# overlay this plot with the regression line using stat_function()
my_plot <- my_plot + stat_function(fun = est_SED, col = "red", lwd = 1, alpha = 0.8)
```

He then simplified his regression equation for clinicals to a 1mm/kg rule such that $\hat{\beta}_1 = 1$ and $\hat{\beta}_0 = 0$. Overlay your plot with the simplified model: $\widehat{\text{WT}} = 0 + 1 \times \widehat{\text{SED}}$.

```
# use geom_abline() to overlay x = y line
my_plot <- my_plot + geom_abline(slope = 1, intercept = 0, col = "blue", lwd = 1)

# print your plot
my_plot
```



Testing coefficients

Use the function `gmodels::estimable()` to obtain the estimates, SEs, and p-values for testing whether each regression coefficient equals 0 (i.e. those from the standard regression output).

What contrast matrix, $\boldsymbol{C}$, will give us $\boldsymbol{\beta}^T_{1\times 2}\boldsymbol{C} = \boldsymbol{0}_{1\times 2}$?

> To test that $H_0 : \beta_0 = 0$ and $H_0 : \beta_1 = 0$, we need $\boldsymbol{C} = \boldsymbol{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ so that $\boldsymbol{\beta}^T_{1\times 2}\boldsymbol{C} = \boldsymbol{\beta}^T_{1\times 2}$.

Create the appropriate contrast matrix, `C`,

```
C <- diag(1, nrow = 2, ncol = 2)
```

and use it as the `cm` input for `gmodels::estimable()`. For `beta0` input a vector for the null values of the parameters.

```
estimable(obj = mod, cm = C, beta0 = c(0, 0))
```

```
##       beta0  Estimate Std. Error   t value  DF    Pr(>|t|)
## (1 0)     0 4.2366095 0.64563237  6.561953 177 5.66232e-10
## (0 1)     0 0.7897822 0.03643483 21.676573 177 0.00000e+00
```

Compare this output to that from your model.

```
# view model
mod
```

```
## Linear Regression Model
##
```

```
##  ols(formula = SED ~ WT, data = dat)
##
##                Model Likelihood     Discrimination
##                   Ratio Test           Indexes
##  Obs      179    LR chi2    231.98    R2        0.726
##  sigma3.4992    d.f.            1    R2 adj    0.725
##  d.f.     177    Pr(> chi2) 0.0000    g         6.269
##
##  Residuals
##
##      Min      1Q  Median      3Q     Max
##  -9.7932 -2.0269 -0.5967  1.4529 17.1300
##
##
##            Coef   S.E.   t     Pr(>|t|)
##  Intercept 4.2366 0.6456  6.56 <0.0001
##  WT        0.7898 0.0364 21.68 <0.0001
##
```

Now, separately test the effect of `WT`, i.e.

$$H_0 : \beta_1 = 0 \text{ vs. } H_1 : \beta_1 \neq 0.$$

What contrast matrix, $\boldsymbol{C}$, will give us $\boldsymbol{\beta}_{2\times 1}\boldsymbol{C} = \beta_1 = 0$?

> To test that $H_0 : \beta_1 = 0$, we need $\boldsymbol{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}$ so that $\boldsymbol{\beta}^T_{1\times 2}\boldsymbol{C} = 0\beta_0 + \beta_1 = \beta_1$.

Create the appropriate contrast matrix, `C`, and use `estimable()` to get the estimates, SEs, and p-values for this test.

```r
C <- matrix(c(0, 1), nrow = 1)
estimable(mod, cm = C, beta0 = 0)
```

```
##       beta0  Estimate Std. Error  t value  DF Pr(>|t|)
## (0 1)     0 0.7897822 0.03643483 21.67657 177        0
```

Obtain the joint test of whether $\begin{bmatrix} \beta_0 & \beta_1 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$.

```r
C <- diag(1, nrow = 2, ncol = 2)
estimable(mod, cm = C, beta0 = c(0, 1), joint.test = TRUE)
```

```
##    X2.stat DF    Pr(>|X^2|)
## 1 43.38089  2 3.801534e-10
```

Do you think Dr. Bosenberg's simplification of the original model was a good idea?

> No, we see from the test above that the slope and intercept from our model were significantly different from that in the simplified model.

Now, code the joint test (for overall regression) by hand using matrices and the F-test.

```r
# create design matrix/ response vector
X <- dat %>% select(WT) %>% mutate(Int = 1) %>% select(Int, WT) %>% data.matrix()
y <- dat %>% select(SED) %>% data.matrix()

# same contast matrix
C <- diag(1, nrow = 2, ncol = 2)

# save some helpful constants
q <- C %>% nrow()
```

```r
k <- C %>% ncol() - 1
n <- y %>% nrow()

# Thrm 8.4a pg 199
H <- X %*% solve(t(X) %*% X) %*% t(X)
B <- solve(t(X) %*% X) %*% t(X) %*% y
SSH <- t(C %*% B) %*% solve(C %*% solve(t(X) %*% X) %*% t(C)) %*% C %*% B
SSE <- t(y) %*% (diag(n) - H) %*% y

# test statistic
(F <- (SSH/q)/(SSE/(n - k - 1)))
```

```
##          SED
## SED 2355.385
```

```r
# p-value
pf(F, q, n - k - 1, lower.tail = FALSE)
```

```
##               SED
## SED 2.879644e-128
```

Does the F-test statistic match the test statistic obtained via the `estimable()` function?

Note: the `estimable()` function returns a χ^2 test statistic, but F and χ^2 test statistics are really the same thing in that, after a normalization, χ^2 is the limiting distribution of the F as the denominator degrees of freedom goes to infinity. The normalization is

$$\chi^2 = \mathrm{df}_{\mathrm{num}}.F$$

```r
estimable(mod, cm = C, joint.test = TRUE)
```

```
##    X2.stat DF Pr(>|X^2|)
## 1 4710.769  2          0
```

```r
# normalize the chi-squared test stat
4710.769 * q
```

```
## [1] 9421.538
```

Do the p-values match? Why or why not?

```r
# get p-value
pchisq(4710.769 * q, q, lower.tail = FALSE)
```

```
## [1] 0
```

Now, hand code the test to jointly compare the fitted model to the simplification.

```r
# vector of null values
beta_null <- c(0, 1)

# Thrm 8.4f/g, pg 203
SSH <- t(C %*% B - beta_null) %*% solve(C %*% solve(t(X) %*% X) %*% t(C)) %*%
    (C %*% B - beta_null)
SSE <- t(y) %*% (diag(n) - H) %*% y

# test statistic
(F <- (SSH/q)/(SSE/(n - k - 1)))
```

```
##           SED
## SED 21.69045
```

```r
# p-value
pf(F, q, n - k - 1, lower.tail = FALSE)
```

```
##              SED
## SED 3.756129e-09
```

and compare this to the output from `estimable()`.

```r
estimable(mod, cm = C, beta0 = beta_null, joint.test = TRUE)
```

```
##    X2.stat DF   Pr(>|X^2|)
## 1 43.38089  2 3.801534e-10
```

What do we notice?

`estimable()` uses `1-pchisq(F*2,2)`, i.e. a Wald Test (large sample). Recall from your notes that quadratic form Q (Wald Test) is divided by degrees of freedom q to derive F statistic.