Why are R2 and F so large for models without a constant?

Authored by

stats writer

June 30, 2024

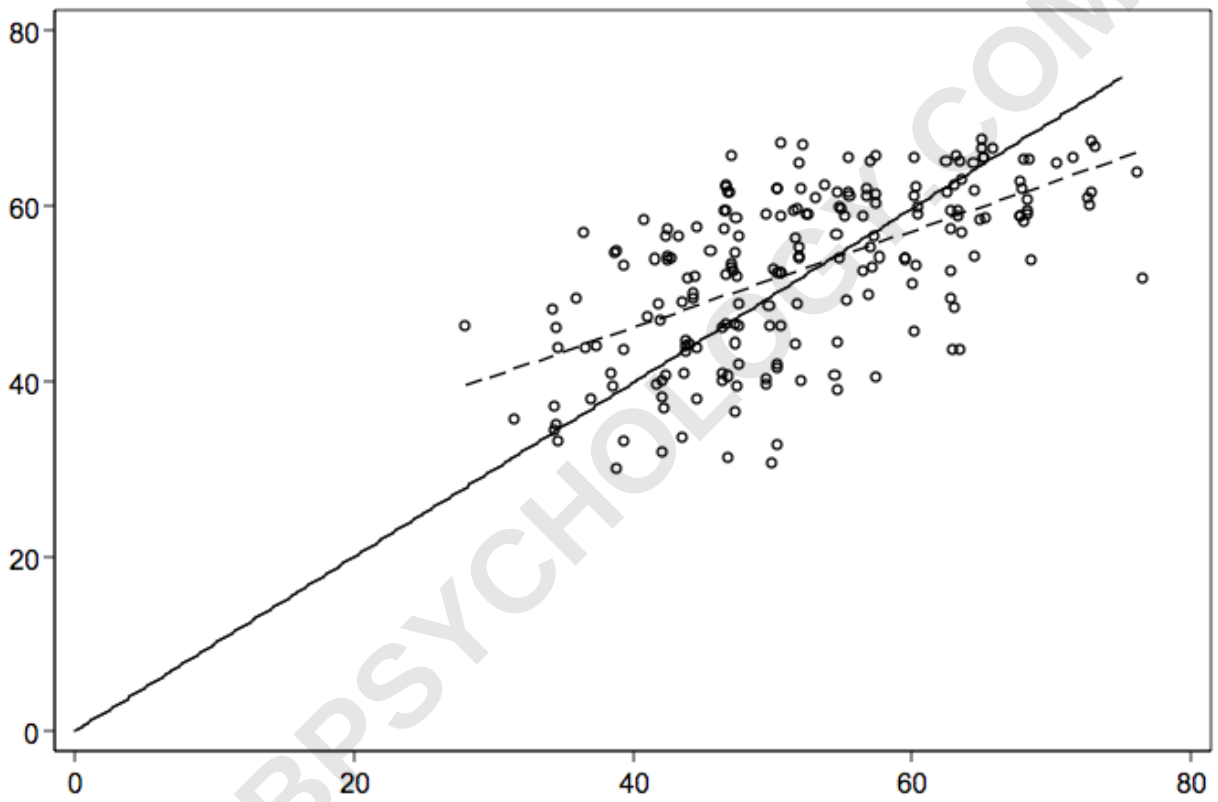
The R2 and F statistics are measurements used to assess the overall performance and goodness of fit of a statistical model. They are commonly used in regression analysis to evaluate the relationship between a dependent variable and one or more independent variables. However, in models without a constant term, these statistics may appear to be significantly large. This is because without a constant, the regression line is forced to pass through the origin, which can lead to inflated R2 and F values. This is due to the fact that the model is essentially fitting the data perfectly, resulting in a high level of explained variation and a significant overall relationship. However, it is important to note that these large values may not accurately reflect the true performance of the model and could potentially lead to incorrect conclusions. Therefore, it is recommended to include a constant term in regression models to ensure more accurate and reliable results.

FAQ: Why are R2 and F so large for models without a constant?

When I run my OLS regression model with a constant I get an R2 of about 0.35 and an F-ratio around 100. When I run the same model without a constant the R2 is 0.97 and the F-ratio is over 7,000. Why are R2 and F-ratio so large for models without a constant?

Let's begin by going over what it means to run an OLS regression without a constant (intercept). A regression without a constant implies that the regression line should run through the origin, i.e., the point where both the response

variable and predictor variable equal zero. Let's look at a scatterplot that has both
the regular regression line (dashed line) and a line without the constant (solid line).



As you can see, the "true" regression line is different from noconstant line. Then how can
it be that the noconstant model has a larger R2 and F-ratio then a model with
a constant?

To answer this question, let's start with a review how

the R2 and F-ratio for OLS regression models are computed.

Next, let's see how each of these sums of squares are defined. For these equations we will use (hat{Y}) for the predicted value of the response variable Y and (bar{Y}) for the mean value of Y.

When you run the regression without a constant in the model, you are declaring that the expected value of Y when x is equal to 0 is 0. That is, (E(Y | x = 0) = 0). If this is not the case, the values of (hat{Y}) will be different yielding different (SS_{model}) and (SS_{residual}) hence different (R^2) and F values. Typically, the sum of squares of Y accounted for by the intercept are not included in the total sum of squares. That is, they are neither in SSmodel nor SSresidual. That is the model is predicting the sum of squares left over after taking out the intercept. When the intercept (or constant term) is left off and it does not have a true

zero effect, the total
sum of squares being modelled is increased. This tends to inflate both SSmodel and
SSresidual; however, SSmodel increases relatively more than
SSresidual leading to the increase in R2 values.

The actual code used to calculate (R^2) are different with and without an intercept.
This is easy to see by running models without a built-in intercept, but manually including one
(a constant term). Here is some example code you can try:

```
sysuse auto
gen const = 1
regress mpg weight
Source | SS df MS Number of obs = 74
-------------+------------------------------ F( 1, 72) = 134.62
Model | 1591.9902 1 1591.9902 Prob > F = 0.0000
Residual | 851.469256 72 11.8259619 R-squared = 0.6515
-------------+------------------------------ Adj R-squared = 0.6467
Total | 2443.45946 73 33.4720474 Root MSE = 3.4389
```

---

mpg | Coef. Std. Err. t P>|t|

---+---

weight | -.0060087 .0005179 -11.60 0.000 -.0070411 -.0049763

_cons | 39.44028 1.614003 24.44 0.000 36.22283 42.65774

---

regress mpg const weight, noconstant

Source | SS df MS Number of obs = 74

---+--- F(2, 72) = 1486.41

Model | 35156.5307 2 17578.2654 Prob > F = 0.0000

Residual | 851.469256 72 11.8259619 R-squared = 0.9764

---+--- Adj R-squared = 0.9757

Total | 36008 74 486.594595 Root MSE = 3.4389

---

mpg | Coef. Std. Err. t P>|t|

---+---

const | 39.44028 1.614003 24.44 0.000 36.22283 42.65774

weight | -.0060087 .0005179 -11.60 0.000 -.0070411 -.0049763

---

regress mpg weight, noconstant

*** note change in total SS between plain regress and wihtout constant**

*** but total SS is the same for without constant and with const**

*** when using manual intercept, intercept SS included in model**

Source | SS df MS Number of obs = 74

----------+---------- F(1, 73) = 259.18

Model | 28094.8545 1 28094.8545 Prob > F = 0.0000

Residual | 7913.14549 73 108.399253 R-squared = 0.7802

----------+---------- Adj R-squared = 0.7772

Total | 36008 74 486.594595 Root MSE = 10.411

--

mpg | Coef. Std. Err. t P>|t|

----------+---

weight | .006252 .0003883 16.10 0.000 .0054781 .007026

--

one

Analysis of Variance Table

Response: mpg

Df Sum Sq Mean Sq F value Pr(>F)

qsec 1 197.39 197.392 6.3767 0.01708 *

Residuals 30 928.66 30.955

Signif. codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

anova(lm(mpg ~ 0 + one + qsec, data = mtcars))

Analysis of Variance Table

Response: mpg

Df Sum Sq Mean Sq F value Pr(>F)

one 1 12916.3 12916.3 417.2570 anova(lm(mpg ~ 0 + qsec, data = mtcars))

Analysis of Variance Table

Response: mpg

Df Sum Sq Mean Sq F value Pr(>F)

qsec 1 13105.6 13105.6 433.73