STAT 213: R^2, Adjusted R^2, and Parsimony

Colin Reimer Dawson

Last Revised March 28, 2018

Let's do an experiment to examine what happens when we build a sequence of increasingly complex regression models, where in reality, some of the predictors have nothing to do with the response.

I have created a fake dataset with ten predictor variables they're just called (`V1` through `V10`), each with completely random values generated from a standard Normal distribution.

Run the following to run a script on the website that generates the data (you can download it and look over it later if you're interested; for now just run it):

```
source("http://colindawson.net/stat213/code/make_fake_mlr_data.R")
```

This will create several variables in your environment, the most important of which is the dataset called `FakeData`.

The "target" function (think "population-level regression equation") is the following:

$$\mu_Y = f(X_1, X_2, X_3, X_4) = 0 + 0.87 \cdot \texttt{V1} - 0.36 \cdot \texttt{V2} - 0.51 \cdot \texttt{V3} + 0.07 \cdot \texttt{V4}$$

The above equation describes the "long run" mean of Y at any combination of X_1 through X_4, based on the *true process* that generated the data (in reality we never know what this target function is; we have to estimate it).

The remaining six variables (X_5 through X_{10}) are completely unrelated to the function.

To make things more interesting (so that each group can get slightly different results),

create your own values of the response variable by generating some residuals from a Normal distribution with mean 0 and standard deviation 0.5:

```
### Use someone's T number in your group in the set.seed() function
### This will ensure that your results are "random" but reproducible
set.seed(00029747)
# Counts the number of datapoints
n <- nrow(FakeData)
# Creates a set of Normally distributed residuals, one for each data point
epsilon <- rnorm(n, mean = 0, sd = 0.5)
```

The response variable we will fit our models to is the idealized function values (the `fX` variable in the data) plus your random residuals. Let's put this column in the dataset.

```
FakeData <- mutate(FakeData, Y = fX + epsilon)
```

If we knew the form of the true "population" (or "long run") function, then we should fit a model that uses the first four variables but not the last six. Imagine we didn't, though, and we were faced with the problem of considering various combinations of predictors.

In full there are $2^{10} = 1024$ possible subsets of predictors that we could use (each subset corresponds to a ten digit binary number where ones mean the variable is included, and zeroes mean the variable is excluded), but since we don't want to be here for hours, let's just consider ten models, each with the first k variables ($k = 1, \ldots, 10$) included as predictors.

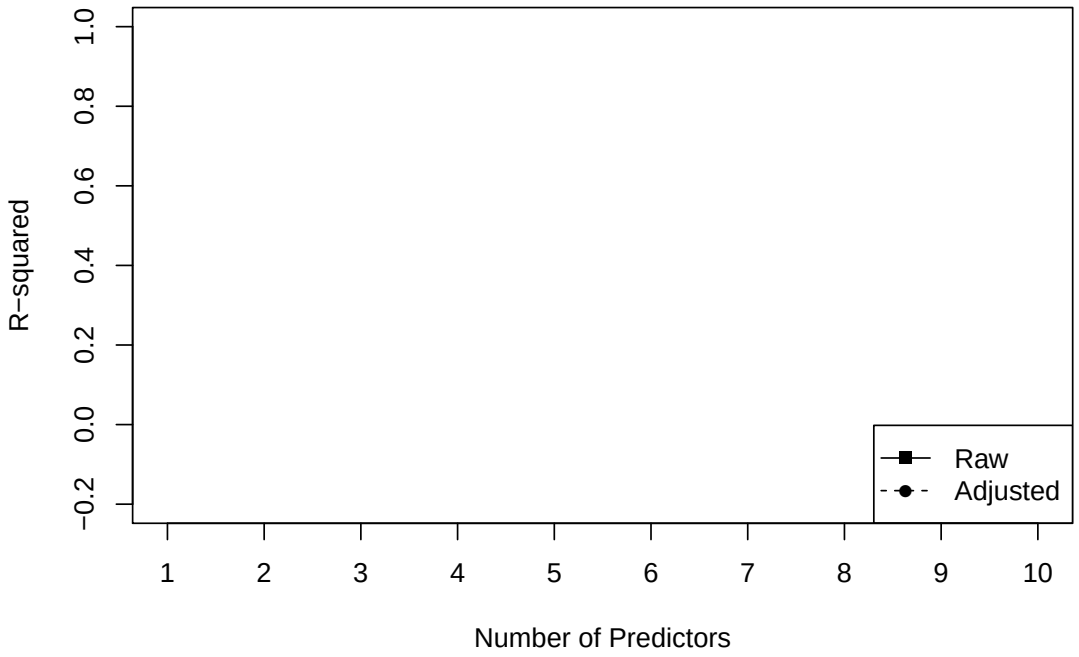
1. $Y = \hat{\beta}_0 + \hat{\beta}_1 \texttt{V1}$
2. $\texttt{Y} = \hat{\beta}_0 + \hat{\beta}_1 \texttt{V1} + \hat{\beta}_2 \texttt{V2}$

   ...

9. $\texttt{Y} = \hat{\beta}_0 + \hat{\beta}_1 \texttt{V1} + \hat{\beta}_2 \texttt{V2} + \hat{\beta}_3 \texttt{V3} + \hat{\beta}_4 \texttt{V4} + \hat{\beta}_4 \texttt{V4} + \hat{\beta}_5 \texttt{V5} + \hat{\beta}_6 \texttt{V6} + \hat{\beta}_7 \texttt{V7} + \hat{\beta}_8 \texttt{V8} + \hat{\beta}_9 \texttt{V9}$
10. $\texttt{Y} = \hat{\beta}_0 + \hat{\beta}_1 \texttt{V1} + \hat{\beta}_2 \texttt{V2} + \hat{\beta}_3 \texttt{V3} + \hat{\beta}_4 \texttt{V4} + \hat{\beta}_4 \texttt{V4} + \hat{\beta}_5 \texttt{V5} + \hat{\beta}_6 \texttt{V6} + \hat{\beta}_7 \texttt{V7} + \hat{\beta}_8 \texttt{V8} + \hat{\beta}_9 \texttt{V9} + \hat{\beta}_{10} \texttt{V10}$

Notice that for the first four models, we are adding predictors that are actually related to the response; in other words, they *should* be in the model. For models 5 through 10, the predictors are, in reality, "junk".

For your ten models, find the R^2 and adjusted R^2 (get from the summary), and plot them both below (by hand, not in R) as a function of the number of predictors.



1. What do you notice about the *unadjusted* R^2 values?

2. What happens with the *adjusted* R^2 values? Which model does adjusted R^2 suggest is the best one? Did it do well?