Demo of the regression_viz()

Nick Strayer

April 19, 2017

Motivation

Ever desired to report the results of a regression in a meaningful way but weren't a fan of another table making your report boring?

As statisticians we're programmed to think with tables, but they do a very poor job of allowing rapid comparisons between multiple models, appreciation of effect size and generally just being engaging to readers.

This is an attempt to combine the precision of tables with the intuition of a plot. Bars represent coefficient estimates along with the mean and standard error written above. You can optionally have a bar drawn at 0 for a visual p-value (if the 95% interval doesn't go over that line, you have a significant p-value.)

Demonstration

Let's test it out with a few contrived models.

First we load our packages to make our life easier.

```
library(tidyverse)
library(nviz)
```

Next we generate some fake data.

```
num_obs <- 200

my_cool_data <- data_frame(
  x1 = rnorm(num_obs, mean = 1, sd = 10),
  x2 = rnorm(num_obs, mean = 1, sd = 20),
  x3 = runif(num_obs),
  y  = 10 + 1.2*x1 + -.312*x2 + 0*x3 + 5*x1*x3 + rnorm(num_obs, mean = 5)
)
```

This is a contrived example but note that the true outcome generation procedure has an interaction between `x1` and `x3` and no true relationship between `x3` alone and the outcome.

Now let's fit a few different models to this so we can compare them.

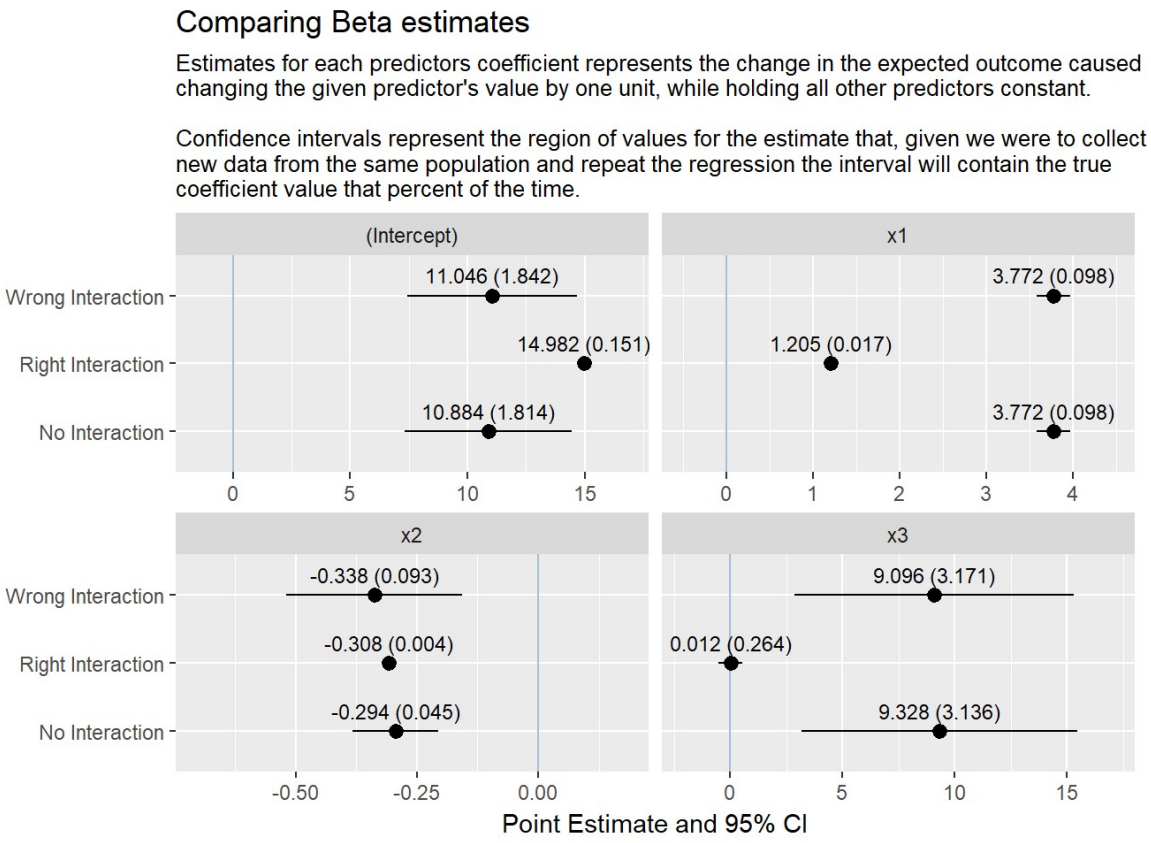
```
no_interaction    <- lm(y ~ x1 + x2 + x3, data = my_cool_data)
wrong_interaction <- lm(y ~ x1 + x2 + x3 + x2:x3, data = my_cool_data)
right_interaction <- lm(y ~ x1 + x2 + x3 + x1:x3, data = my_cool_data)
```

Another note, this is not really a fair comparison but I'm doing it to keep the data-generation simple. When we compare model coefficients in almost every scenario we would want all of the models to contain exactly the same terms as then the inference on them will be comparable. For instance you would compare a model done with Lasso penalization to a non-penalized model with the same values to see the extent of shrinkage on your coefficients.

Okay, now that the lame formatlities are over, let's send the data to our function.

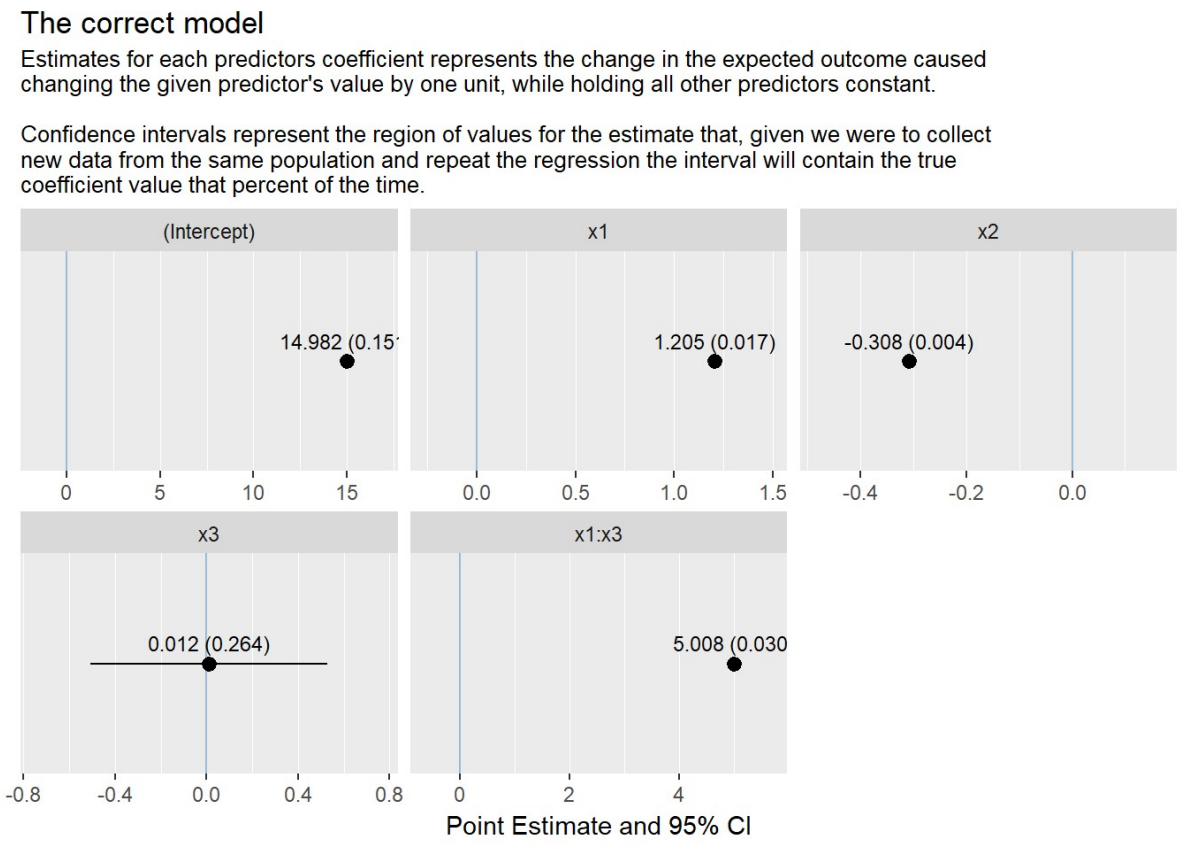
```
my_models <- list(no_interaction, wrong_interaction, right_interaction )
names(my_models) <- c("No Interaction", "Wrong Interaction", "Right Interactio
n")

regression_viz(my_models, plot_title = "Comparing Beta estimates")
```



We can also just plot a single model.

```
regression_viz(right_interaction, plot_title = "The correct model")
```



What we get from this

So looking at this chart you can immediately see that the confidence intervals for the correct model “right interaction” are much smaller than the other models. This makes sense as our model was set up to perfectly match the data generation (aka something that never happens). We can also see that while the other models give highly significant results for the covariate `x3` the correct interaction model *correctly* identifies it as statistically zero. Also `x1` ’s effect is significantly attenuated towards zero. Both of these are due to the unplotted interaction coefficient absorbing a lot of the effect that the other models didn’t properly account for.

I personally believe these observations are much easier in this type of display than from a simple table. In addition to satisfy the statisticians who need to concrete numbers, they are still there, eliminating the need for both a plot and a table.