Lab 5: Regression

Steven Boyd

10/28/2021

Regression with a single indicator independent variable

First, let's practice the mechanics of running a regression in R. As you've seen, there are different ways to do this. For most applications, `lm()` is sufficient, but there may be situations in which you want to use the `estimatr` package. We will practice the syntax for both.

First, we need some data. Let's practice with `mtcars`.

```
car_data <- mtcars
```

Let's say that we are interested in exploring the relationship between transmission and fuel economy. Let's regress `mpg` onto `am`.

```
model_1 <- lm(car_data, formula = mpg ~ am)

summary(model_1)
```

```
##
## Call:
## lm(formula = mpg ~ am, data = car_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3923 -3.0923 -0.2974  3.2439  9.5077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.147      1.125  15.247 1.13e-15 ***
## am             7.245      1.764   4.106 0.000285 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.902 on 30 degrees of freedom
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3385
## F-statistic: 16.86 on 1 and 30 DF,  p-value: 0.000285
```

We have a slope and intercept, but what do they mean? That depends on what form the independent variable takes and what the specific values represent. Let's check the documentation.

```
?mtcars
```

```
## starting httpd help server ... done
```

What values does `am` take in the data? What do those values represent?

`am` can take on two values: 0 and 1. They represent automatic and manual transmission respectively. When we have a dichotomous categorical variable mapped onto the values 0 and 1, we usually refer to it as an "indicator variable" (you will also encounter the term "dummy variable").

Now that we understand what the variable represents, let's interpret our results. One way we can do this, which is especially helpful when we have multiple independent variables is to think about the functional form of our regression. In this case it would be:

$$\text{mpg} = 17.147 + 7.245D_i$$

where $D_i = 0$ if the car has automatic transmission and $D_i = 1$ if it has a manual transmission.

So, the model predicts that automatic cars get an average of 17.147 mpg and manual cars get 7.245 more mpg than that.

As discussed in lecture 5.1, when we have a regression that is structured this way, the intercept is equal to the mean of the "untreated" category and the slope is the difference in means. Can you verify this?

```
car_data %>%
  filter(am==0) %>%
  summarize(mean(mpg))
```

```
##   mean(mpg)
## 1  17.14737
```

```
car_data %>%
  summarize(mean(mpg[am==1]) - mean(mpg[am==0]))
```

```
##   mean(mpg[am == 1]) - mean(mpg[am == 0])
## 1                                 7.244939
```

Extracting info from a regression

Notice that there is a lot more information stored in the object `model_1` than just the coefficients. For example, the fitted values (aka predicted values) and residuals. The fitted values are the y_i generated for each observation by the model and the residuals are the difference between the observed and fitted values (i.e. residual = observed - fitted).

If you take Linear Models in the winter quarter, you will talk in great detail about residuals and why we care about them. For now just know that you can retrieve them from your model like so:

```
my_resid_1 <- model_1$residuals
```

Since we have the residuals and the fitted values, we should be able to recreate the original `mpg` column. See if you can recreate the mpg column and verify that they are the same.

```
my_fit_1 <- model_1$fitted.values

my_mpg <- my_fit_1 + my_resid_1

FALSE %in% as.logical(near(car_data$mpg, my_mpg))
```

```
## [1] FALSE
```

Adding controls

Let's think more critically about the regression we ran above. There's an apparent association between transmission type and miles per gallon. This data is pretty old (1981 I believe), and automatic transmissions were less common and more expensive than they are today. Taking a quick glance at the data, it appears that some of the cars with the largest engines (measured by displacement) have automatic transmissions and some of cars with the smallest engines have manual transmissions. It is possible that engine size is a confounder, influencing both the choice of transmission (maybe car manufacturers wanted smoother shifting in their more powerful, expensive cars) and the fuel economy (bigger engines consume more fuel). So, we want to account for the association between engine size and fuel economy as well.

To make the interpretation easier, let's create a new indicator variable called `sport` which takes on the value 1 if the displacement is greater than 250 cubic inches.

```
car_data <- car_data %>%
  mutate(sport = if_else(disp > 250, 1, 0))
```

Now, let's run the regression again, but include `sport` and `lm`.

```
model_2 <- lm_robust(data = car_data,
                     formula = mpg ~ am + sport)

summary(model_2)
```

```
##
## Call:
## lm_robust(formula = mpg ~ am + sport, data = car_data)
##
## Standard error type:  HC2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   21.738      1.139  19.077 5.882e-18   19.408   24.069 29
## am             3.686      1.600   2.303 2.863e-02    0.413    6.960 29
## sport         -6.710      1.378  -4.870 3.637e-05   -9.527   -3.892 29
##
## Multiple R-squared:  0.5916 ,    Adjusted R-squared:  0.5634
## F-statistic: 19.49 on 2 and 29 DF,  p-value: 4.314e-06
```

What happened to the intercept and coefficient on `am`? What does the coefficient on `sport` mean? What happened to the value of R^2?

The intercept increased and the coefficient on `am` decreased. The coefficient on sport can be interpreted as: the model predicts that (all else held equal), a "sports" car will get 6.7 fewer mpg than a non-sports car. The value of R^2 increased, suggesting that this model fits the data better than the previous one.

Interaction terms

The new coefficients suggest that the baseline fuel economy for automatic, non-sport cars is higher than for automatic cars in general, but manual cars are still more fuel efficient on average. Suppose a car industry expert comes to us and points out that there is variation within the cars that have manual transmissions that our model doesn't capture. Manual sports cars tend to be big, beefy American muscle cars (which have terrible gas mileage), but manual non-sports cars tend to be inexpensive, lighter models (which get pretty good gas mileage). In other words, the independent variables in our model *interact* in a way that isn't captured by the coefficients from the previous model. So, let's add an *interaction term* to the model.

The way to add an interaction term to your model is by including the product of two independent variables.

```
model_3 <- lm_robust(data = car_data,
                     mpg ~ am*sport)

summary(model_3)
```

```
##
## Call:
```

```
## lm_robust(formula = mpg ~ am * sport, data = car_data)
##
## Standard error type:  HC2
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   20.633      1.097  18.801 2.040e-17   18.385   22.881 28
## am             5.394      1.901   2.838 8.352e-03    1.501    9.287 28
## sport         -5.095      1.409  -3.615 1.166e-03   -7.981   -2.208 28
## am:sport      -5.532      2.134  -2.593 1.497e-02   -9.904   -1.161 28
##
## Multiple R-squared:  0.6242 ,    Adjusted R-squared:  0.5839
## F-statistic: 20.19 on 3 and 28 DF,  p-value: 3.624e-07
```

Interpret the results. The functional form of this model is:

$$\text{mpg} = 20.633 + 5.394(\text{am}) - 5.095(\text{sport}) - 5.532(\text{am})(\text{sport})$$

where `am` and `sport` are either 0 or 1. What does our model predict the fuel economy of an automatic sports car is? Automatic non-sports car? Manual sports car? Manual non-sports car?

Automatic sports car: 15.538 Automatic non-sports car: 20.633 Manual sports car: 15.400 Manual non-sports car: 26.027

The key is to notice that the coefficient on the interaction term is only realized when *both* indicator variables equal 1.

Regression with continuous independent variables

So far, we've been working with indicator variables, which have coefficients that are easy to interpret. You're most likely to encounter these variables in experimental contexts. Unfortunately most social science research is not so neat or easy to interpret. Continuous variables are everywhere we look in the real world, and frequently find their way into out models.

Let's look back one of the plots we generated in Lab 1 using the UK coal data. If the code below doesn't work for you, you can save the csv locally (the data is in the course GitHub repo) and read it in.

```
data_path <- "https://raw.github.com/aeggers/IntroQSS-F21/main/data/"

coal_data <- read_csv(str_c(data_path, "uk_coal_tables.csv"),
              col_types = "cddddddd",
              na = "N/A")
```

Remember this plot that we produced:

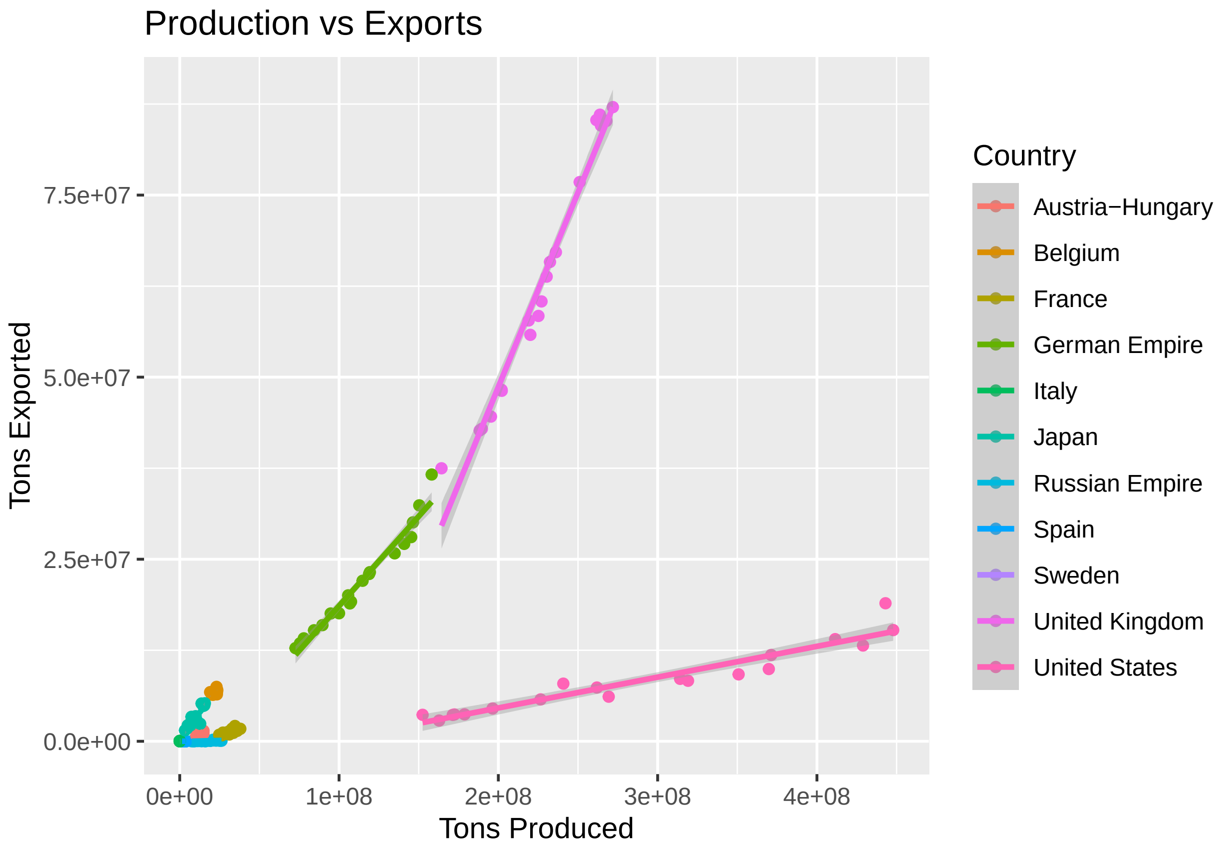
```
plot_1 <- ggplot(data = coal_data,
                 mapping = aes(x = Tons_Produced,
                               y = Tons_Exported,
                               color = Country)) +
                  geom_point() +
                  geom_smooth(method = "lm") +
                  labs(x = "Tons Produced",
                  y = "Tons Exported",
                  title = "Production vs Exports")

plot_1
```

```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Warning: Removed 2 rows containing missing values (geom_point).
```



We colored by country because there seemed to be clusters of points that were behaving differently. Each line represents an OLS regression for a specific country. Use your data wrangling skills and the ggplot code above to find the slope and intercept of the regression line for the United Kingdom observations.

```
coal_uk <- coal_data %>%
  filter(Country == "United Kingdom")

model_4 <- lm_robust(data = coal_uk,
                     Tons_Exported ~ Tons_Produced)

summary(model_4)
```

```
##
## Call:
## lm_robust(formula = Tons_Exported ~ Tons_Produced, data = coal_uk)
##
## Standard error type:  HC2
##
## Coefficients:
##                  Estimate Std. Error t value  Pr(>|t|)   CI Lower   CI Upper DF
## (Intercept)   -5.819e+07  8.466e+06  -6.873 2.702e-06 -7.605e+07 -4.033e+07 17
## Tons_Produced  5.342e-01  3.550e-02  15.046 2.949e-11  4.593e-01  6.091e-01 17
```

```
##
## Multiple R-squared:  0.9737 ,    Adjusted R-squared:  0.9722
## F-statistic: 226.4 on 1 and 17 DF,  p-value: 2.949e-11
```

Interpret the results (remember that the independent variable is continuous). Does the intercept make sense? What does this tell us about this model's ability to generate out of sample predictions? Notice the R^2 value. Does it tell us if this is a "good" model or not?

The slope can be interpreted as: the model predicts that for each additional ton of coal the UK produces, it exports an additional .534 tons of coal. The intercept is approximately -55 million. Obviously we could never observe this in the real world (because countries can't export negative quantities of coal). The R^2 value is extremely high, so the model is an excellent fit for the data we have. This demonstrates that just because a model has good fit, we should not assume that it will be able to produce reliable (or even plausible) out-of-sample predictions.

Final project brainstorming

Brainstorm a regression you might want to add to your final project. What are the independent and dependent variables? Are they continuous or categorical? How would you interpret your regression coefficient?