Solutions/Debrief: Data Exercise 1

Michael G. DeCrescenzo

February 05, 2020

This document describes what we were looking for with Data Exercise 1.

The code that I distributed contained *everything* you needed to run in R. Your job was essentially to interpret what you were seeing with your particular draw of data.

Your data would have looked something like this:

```
## # A tibble: 20 x 3
##   legislator_id terms Leg_Act
##           <dbl> <dbl>   <dbl>
## 1             1     1       5
## 2             2     1      12
## 3             3     2      12
## 4             4     2      19
## 5             5     2       6
## # ... with 15 more rows
```

The exact values of `Leg_Act` will differ for each person, since each individual simulated a different set of data from the following model…

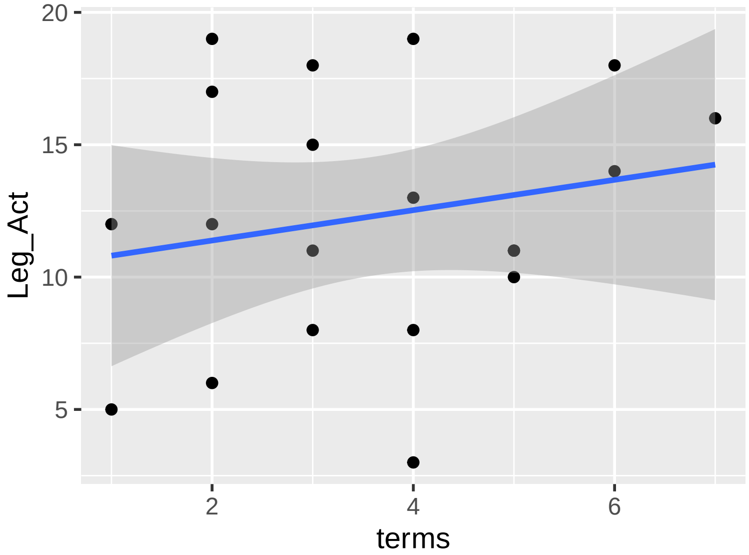
$$\texttt{Leg\_Act}_i = \alpha + \beta(\texttt{terms}_i) + \epsilon_i \tag{1}$$

with the true values $\alpha = 10$ and $\beta = 0$. The only reason why anybody sees different data is because the values of ϵ_i differ from person to person.

You were asked to plot the data. Some students took my code and added a fitted line to the data. If you did this, it's important to note that that only appropriate line (given the rest of the assignment) is a *straight line* from a linear model.[1] Here is code to show the line you would have wanted to draw:

[1] If you added a loess line (the default option with `geom_smooth()`), that would be irrelevant to the rest of the assignment! Loess lines (or any kind of non- or semi-parametric fit line) can be valuable for learning about potential nonlinear relationships, but these smoothers can be tweaked in several complex ways to give different summaries of the data. Best to use them with caution.

```r
ggplot(d) +
  aes(x = terms, y = Leg_Act) +
  geom_point() +
  geom_smooth(method = "lm")
```



You were asked to (1) estimate the regression line, (2) report the coefficients to appropriate units of precision, and (3) interpret the parameters.

In my case, my intercept estimate ($\hat{\alpha}$, note the *hat*) was about 10.2, and my slope estimate ($\hat{\beta}$) was about 0.57. This means that this model estimates that for a one-unit increase in `terms`, the scale value of `Leg_Act` is predicted (predicted, meaning, $\hat{y}$) to increase by 0.57 units. The intercept tells us that the model predicts a `Leg_Act` value of 10.2 for a legislator with a `terms` value of 0. In this case, we don't expect any legislators to serve 0 terms, so the intercept is a parameter that merely helps us fit the appropriate line.

You were also asked to test the null hypothesis that $\beta = 0$. This information is available to you in the results of `broom::tidy()`, which is a table of your estimated model parameters. The important information for hypothesis testing is in the `statistic` column (which contains t-statistics for an OLS model) and the `p.value` column.

```r
# broom::tidy() shows a coefficient-level summary of the model
tidy(reg)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   10.2       2.56      4.01  0.000828
## 2 terms          0.573     0.645     0.889 0.386
```

In my case, the estimated t-statistic for the `terms` variable was 0.89, which (with 18 degrees of freedom)[2] has a p-value of 0.39. I cannot reject a null hypothesis that $\beta = 0$, since my p-value is larger than the values we typically look at for conventional significance levels (0.05, 0.01, etc.).

We were paying special attention to the way you interpreted your p-value and described your hypothesis test. The two-tailed p-value is the probability of finding a coefficient *farther from zero* than your estimated coefficient by chance, *under the assumption that the null hypothesis is true.* Here are several things that the p-value ***IS NOT***:

- The probability that the true slope is zero
- The probability of no effect
- One minus the probability of a true effect
- One minus the probability that the true slope is greater than zero

Do not say these! The p-value *does not* say anything about the probability of what the effect *is* (such as, "I am 95% confident that the true effect is positive"). Nor does it say anything about the probability of what the effect *is not* (such as, "95% confident that the true effect is not zero"). In fact, it is probably best to avoid the language of "confidence" altogether, since confidence implies beliefs about parameters, and we aren't learning Bayesian statistics. Instead, the p-value is a statement about a hypothetical: *if the null hypothesis were true, what is the probability that I find a more extreme estimate by chance?*

Similarly, we were paying attention to your language about your hypothesis test. If your p-value does not imply a significant coefficient, that does not mean that the null hypothesis is true. Conversely, if you detected a significant coefficient, that does not mean that the null hypothesis is incorrect. You either *reject* or *fail to reject* a null hypothesis. The underlying truth of the null hypothesis is not affected by your conclusion! This is because your conclusion is sensitive to your significance threshold—you could change your threshold and thus change your conclusion. Hypothesis tests are *decisions* that you make about about inferred values of the unknown parameter. These decisions could be randomly incorrect, due to the nature of random error in your model.

One technical R note: if you used the `t.test()` function to perform your hypothesis test, this was incorrect. See the footnote.[3]

[2] 20 original observations, minus 2 estimated model parameters.

[3] If you did something like `t.test(d$terms)`, then what you have asked R to do is test the null hypothesis that the mean of `terms` is equal to zero. You would have rejected the null hypothesis, but it would not have mattered, because you asked R a question that did not make sense for the assignment. Always be questioning what an R function does by examining its help file: `help(function_name)` or `?function_name`.