Notebook

April 13, 2019

1 Confidence Intervals and Bayesian Statistics oh my!

One of the readings for week 13, "The Bayesian New Statistics," covers a variety of different approaches to statistics, as contrasted with the standard frequentist hypothesis-testing method. I don't expect you to come out of this class being able to work any of those alternative paradigms, but you should be able to recognize them and understand broadly how they operate. That article is a very good summary of the landscape, but this supplemental lesson aims to provide a briefer and slightly more basic introduction.

1.1 I. Confidence Intervals vs P-Values

One of the ideas in that article is "new statistics," which is roughly meant to capture the move away from p-values and hypothesis tests and toward confidence intervals. This movement has arisen in recent years mainly because of the many ways in which hypothesis tests and p-values have been misused. The idea is that reporting a confidence interval is a clearer expression of the range of uncertainty of the thing you're trying to estimate.

So what's a confidence interval? Roughly speaking, it's the range equivalent of a p-value—a lower bound and an upper bound for a parameter constructed to achieve a specified degree of confidence (typically 95%).

Interpreting confidence intervals can be difficult, however. The standard interpretation of them is "if we took a bunch of samples from the population, then calculated a 95% confidence interval from each of the samples, we'd expect to see the population (true) parameter within 95% of the confidence intervals in our hypothetical creation of many samples. Here's a concise definition from some epidemologists and here's an explanation of common misinterpretations of confidence intervals as well as p-values.

Statsmodels reports confidence intervals for regression coefficients. Let's take a look at them, and then I'm going to show you one (fairly brutish—there are more sophisticated and accurate ways to do it in the real world) way to create your own via bootstrapping.

First, let's simulate some data. We're going to pretend that heights are normally distributed around 6 feet with a wide standard deviation (they're not), and then we're going to pretend that shoe sizes are about height in inches/10, with some Gaussian (normally distributed) noise added to represent, say, observation error, or other factors leading to shoe size other than height (diet? amount of soccer played as a child?).

```
In [1]: import numpy as np
        import pandas as pd
        import statsmodels.formula.api as sm
```

```
        np.random.seed(90210)
```

```
In [2]: heights = np.random.normal(loc=72.0, scale=5.0, size=500)
        shoe_sizes = [(height / 10) + np.random.normal(scale=2.0) for height in heights]
        heights = heights.astype(int)
        shoe_sizes = np.array(shoe_sizes).astype(int)
```

So we have a simulated sample of heights and shoe sizes, of size 500, and we made them integers (which I believe should just truncate the decimals, though I could be misremembering how Numpy does it, and it might round—probably truncates though) to add some more noise. Now let's wrap them up in a Data Frame and look at a regression coefficient.

```
In [3]: df = pd.DataFrame({"height": heights, "shoe": shoe_sizes})
```

```
        reg = sm.ols(formula='shoe ~ height', data=df).fit()
        print(reg.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   shoe   R-squared:                       0.070
Model:                            OLS   Adj. R-squared:                  0.068
Method:                 Least Squares   F-statistic:                     37.39
Date:                Sat, 13 Apr 2019   Prob (F-statistic):           1.95e-09
Time:                        14:09:45   Log-Likelihood:                -1050.2
No. Observations:                 500   AIC:                             2104.
Df Residuals:                     498   BIC:                             2113.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -1.1321      1.270     -0.891      0.373      -3.628       1.363
height         0.1089      0.018      6.115      0.000       0.074       0.144
==============================================================================
Omnibus:                        6.090   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.048   Jarque-Bera (JB):                6.220
Skew:                           0.268   Prob(JB):                       0.0446
Kurtosis:                       2.894   Cond. No.                     1.02e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.02e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Some things to note here (ignoring the silliness about the condition number, which is related to troubles that statsmodels has had with this calculation).