Bootstrapping Data: Regression

2017-03-01

Preparation

We will eventually use the data in the *mnSchools.csv* file. These data include institutional-level attributes for several Minnesota colleges and universities. The source of these data is: http://www.collegeresults.org. The attributes include:

- `id`: Institution ID number
- `name`: Institution name
- `gradRate`: Six-year graduation rate. This measure represents the proportion of first-time, full-time, bachelor's or equivalent degree-seeking students who started in Fall 2005 and graduated within 6 years.
- `public`: Dummy variable indicating educational sector (0 = private institution; 1 = public institution)
- `sat`: Estimated median SAT score for incoming freshmen at the institution
- `tuition`: Cost of attendance for full-time, first-time degree/certificate-seeking in-state undergraduate students living on campus for academic year 2013-14.

```
mn = read.csv(file = "~/Google Drive/Documents/github/EPsy-8252/data/mnSchools.csv")
head(mn)
```

id	name	gradRate	public	sat	tuition
1	Augsburg College	65.2	0	1030	39294
3	Bethany Lutheran College	52.6	0	1065	30480
4	Bethel University, Saint Paul, MN	73.3	0	1145	39400
5	Carleton College	92.6	0	1400	54265
6	College of Saint Benedict	81.1	0	1185	43198
7	Concordia College at Moorhead	69.4	0	1145	36590

```
# Load libraries
library(mosaic)
library(sm)
```

Bootstrapping Cases from a Data Frame

The `resample()` function also can bootstrap an entire data frame. For example,

```
resample(mn)
```

	id	name	gradRate	public	sat	tuition	orig.id
19	23	Saint Mary's University of Minnesota	62.4	0	1070	35700	19
25	16	Minnesota State University Moorhead	44.7	1	1030	16800	25
21	26	St Olaf College	84.9	0	1285	47200	21
31	30	University of Minnesota-Morris	60.3	1	1165	21722	31
24	2	Bemidji State University	42.3	1	1010	18057	24
19.1	23	Saint Mary's University of Minnesota	62.4	0	1070	35700	19
29	28	University of Minnesota-Crookston	46.4	1	1010	19897	29
28	24	Southwest Minnesota State University	39.5	1	1010	18102	28
9	10	Crown College	51.3	0	1030	33210	9

	id	name	gradRate	public	sat	tuition	orig.id
7	8	Concordia University-Saint Paul	47.9	0	990	37795	7
26	17	Minnesota State University-Mankato	49.6	1	1030	16294	26
32	31	University of Minnesota-Twin Cities	70.2	1	1245	23058	32
33	33	Winona State University	54.0	1	1070	19670	33
27	21	Saint Cloud State University	48.5	1	1010	17050	27
9.1	10	Crown College	51.3	0	1030	33210	9
16	19	Northwestern College, Saint Paul, MN	63.6	0	1105	35400	16
3	4	Bethel University, Saint Paul, MN	73.3	0	1145	39400	3
16.1	19	Northwestern College, Saint Paul, MN	63.6	0	1105	35400	16
15	18	North Central University	40.4	0	1010	25698	15
13	14	Martin Luther College	72.8	0	1125	19450	13
19.2	23	Saint Mary's University of Minnesota	62.4	0	1070	35700	19
22	27	The College of Saint Scholastica	63.5	0	1030	38756	22
7.1	8	Concordia University-Saint Paul	47.9	0	990	37795	7
32.1	31	University of Minnesota-Twin Cities	70.2	1	1245	23058	32
27.1	21	Saint Cloud State University	48.5	1	1010	17050	27
33.1	33	Winona State University	54.0	1	1070	19670	33
8	9	Crossroads College	26.9	0	970	25345	8
31.1	30	University of Minnesota-Morris	60.3	1	1165	21722	31
27.2	21	Saint Cloud State University	48.5	1	1010	17050	27
14	15	Minneapolis College of Art and Design	69.6	0	1105	41420	14
29.1	28	University of Minnesota-Crookston	46.4	1	1010	19897	29
21.1	26	St Olaf College	84.9	0	1285	47200	21
25.1	16	Minnesota State University Moorhead	44.7	1	1030	16800	25

We can use this idea directly in the `lm()` function to fit a regression model on a bootstrapped data set. For example, to fit a regression model using `public` to predict `gradRate`, we can use

```
lm(gradRate ~ 1 + public, data = resample(mn))
```

```
##
## Call:
## lm(formula = gradRate ~ 1 + public, data = resample(mn))
##
## Coefficients:
## (Intercept)       public
##        65.6        -11.9
```

Remember the goal in inference is to see how much coefficients in the regression vary from sample-to-sample. We can carry fit the model to several bootstrap samples by using the `do()` function.

```
myBoot = do(1000) * lm(gradRate ~ 1 + public, data = resample(mn))
head(myBoot)
```

Intercept	public	sigma	r.squared	F	numdf	dendf	.row	.index
67.6	-11.38	16.8	0.102	3.502	1	31	1	1
67.7	-9.50	14.9	0.079	2.663	1	31	1	2
63.4	-13.26	15.9	0.119	4.204	1	31	1	3
72.9	-22.51	13.7	0.406	21.166	1	31	1	4
61.3	-6.35	15.8	0.028	0.896	1	31	1	5
59.1	-7.56	17.6	0.032	1.022	1	31	1	6

The `public` column in the `myBoot` data frame contains the 1000 regression slopes from the model fitted to the bootstrapped samples. To quantify the sampling variation, we compute the standard deviation of these measures.

```
se_public = sd(myBoot$public)
se_public
```

```
## [1] 4.7
```

Compare this with the SE based on the theory-based approximation found from the `lm()` function.

```
lm.1 = lm(gradRate ~ 1 + public, data = mn)
summary(lm.1)
```

```
##
## Call:
## lm(formula = gradRate ~ 1 + public, data = mn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -38.37  -6.33   0.13   9.03  27.33
##
## Coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept)    65.27       3.25   20.05 <0.0000000000000002 ***
## public        -14.24       5.91   -2.41               0.022 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.6 on 31 degrees of freedom
## Multiple R-squared:  0.158,  Adjusted R-squared:  0.13
## F-statistic:  5.8 on 1 and 31 DF,  p-value: 0.0222
```

They are pretty similar.

Using the Bootstrap SE to Compute a Simulated p-Value

Recall that under the null hypothesis,

$$H_0 : \beta_{\text{Public}} = 0$$

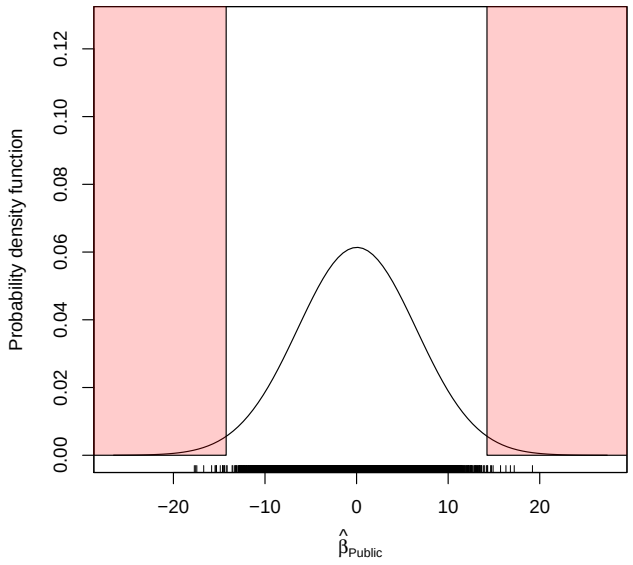
This defines the center (mean) of our sampling distribution. This sampling distribution is also normally distributed with a particular standard error. We just approximated the standard error from our bootstrap procedure. We can now use `rnorm()` to simulate the sampling distribution for $\hat{\beta}_{\text{Public}}$.

```
# Simulate the sampling distribution
beta_hats = rnorm(10000, mean = 0, sd = se_public)
```

The p-value is the probability, under the null hypothesis of seeing an observed slope at least as extreme as the one we saw in the observed data, $\hat{\beta}_{\text{Public}} = -14.24$. The density plot of the simulated $\hat{\beta}_{\text{Public}}$ values below shows the values that are at least as extreme as -14.24. Since, in the social sciences, tests are typically two-tailed, we have to examine those values ≤ -14.24 and those ≥ 14.24. Those areas are shown in red in the plot below.

```
# Plot the beta_hats
sm.density(beta_hats, xlab = expression(hat(beta)[Public]))
```

```
polygon(x = c(-50, -14.24, -14.24, -50), y = c(0, 0, 10, 10), col = rgb(1, 0, 0, 0.2))
polygon(x = c(14.24, 50, 50, 14.24), y = c(0, 0, 10, 10), col = rgb(1, 0, 0, 0.2))
```



Typically we would computet the area under the density curve that is in the red. However, since we have simulated a finite number of $\hat{\beta}$s we can simply count the number that have values at least as extreme as -14.24. Keep in mind that the test is two-tailed meaning we have to count all the $\hat{\beta} \leq -14.24$ and all the $\hat{\beta} \geq 14.24$.

```
# Plot the beta_hats
sum(beta_hats <= -14.24)
```

```
## [1] 15
```

```
sum(beta_hats >= 14.24)
```

```
## [1] 7
```

We can also do this in a single computation using the absolute value,

```
sum(abs(beta_hats) >= 14.24)
```

```
## [1] 22
```

Out of the 10,000 simulated $\hat{\beta}$ values, 22 were at least as extreme as the observed $\hat{\beta}$ of -14.24. As a proportion this is, 0.002. This is the p-value based on the hypothesis that $\beta_{\text{Public}} = 0$.

Comparison to the lm() Summary Output

The p-value for the slope from the `summary()` output of the lm is 0.022. This is quite a bit larger than the simulated p-value. This is because the SE for the results from the `lm()` was quite a bit larger; 5.91. If we simulate from the null distribution having a SE of 5.91, then we obtain a p-value that corresponds to the `summary()` output.

```
beta_lm = rnorm(10000, mean = 0, sd = 5.91)
sum(abs(beta_lm) >= 14.24) / 10000
```

```
## [1] 0.0144
```

The p-value is a function of the uncertainty (the SE). All things being equal, more uncertainty gives larger p-values. So which set of results, the `summary()` output or the bootstrapped and simulated results, should we believe. This decision needs to be is largely based on whether the assumptions are met. If they are met, then the `lm()` results can be trusted. If not, then the bootstrap and simulation results are probably more valid.

Testing Other Hypotheses

The nice thing is that the bootstrap and simulation method now allows us to obtain a p-value to test hypotheses about the slope that are not 0. For example one could test whether

$$H : \beta_{\text{Public}} = 5$$

We just need to adjust the mean in the `rnorm()` function when we simulate the sampling distibution of $\hat{\beta}$.

```
# Simulate the sampling distribution
beta_hats = rnorm(10000, mean = 5, sd = se_public)

# Compute the p-value
sum(abs(beta_hats) >= 14.24) / 10000
```

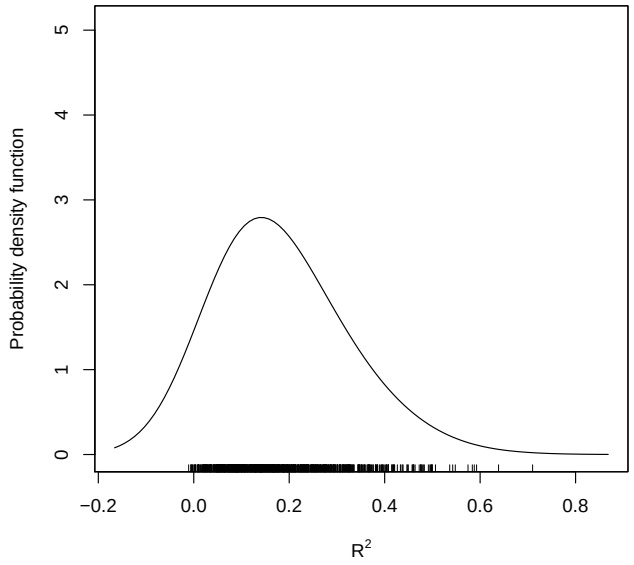
```
## [1] 0.0278
```

Based on the p-value associated with this hypothesis, we would reject the hypothesis that $\beta_{\text{Public}} = 5$. The data are not consistent with this hypothesis.

Other Statistics

Note that the `do()` function collected several other bootstrapped statistics. One of those is R^2. This means, we can now compute an interval estimate for R^2. The density plot of the bootstrapped R^2 values is shown below.

```
sm.density(myBoot$r.squared, xlab = expression(R^2))
```



The positive skew of the distribution indicates that R^2 is biased. Thus to compute an interval estimate, we might consider using the percentile method rather than the plus–minus two SE method.

```
quantile(myBoot$r.squared, prob = c(0.025, 0.975))
```

```
##   2.5%  97.5%
## 0.0151 0.4620
```

Think about what this says. Going back to our observed data, we said that differences in education sector (public/private) explained 15.8% of the variation in graduation rates. The interval estimate gives us some indication of how stable that estimate is from sample-to-sample. Now we are saying that differences in education sector may exlplain as little as 1.509% of the variation in graduation rates, or as much as 46.201% of the variation in graduation rates. There is a lot of uncertainty in our estimate.