The Sampling Distribution of Regression Coefficients.

David C. Howell

Last revised 11/29/2012

This whole project started with a query about the sampling distribution of the standardized regression coefficient, β. I had a problem because one argument was that β is a linear transformation of *b*, and the sampling distribution of *b* is normal. From that it followed that the sampling distribution of β should be normal. On the other hand, with only one predictor, β is equal to *r*, and it is well known that the sampling distribution of *r* is skewed whenever ρ is unequal to zero. From that it follows that the sampling distribution of β would be skewed.

To make a long story short, my error was in thinking of β as a linear transformation of *b*—it is not. The formula for β is

$$\beta = \frac{b_i s_i}{s_0}$$

where s_i is the standard deviation of the ith independent variable, and s_0 is the standard deviation of the dependent (criterion) variable.

But in creating the sampling distribution of β, these two standard deviations are random variables, differing from sample to sample. If I computed β using the corresponding population parameters that would be a different story. But that's not the way you do it. So my statement about β being a linear transformation of was wrong. The unstandardized coefficient (b) is normally distributed, but the standardized coefficient (β) is not normally distributed. It has the same distribution as *r*.

But all is not right in the world. There is something wrong out there, and I can't figure out what. I recently received an e-mail from Alessio Toraldo, at Università di Pavia, Italy. He pointed out that when he did a sampling study similar to the one described below, using a sample size of $n = 10$, the distribution of *b* was distinctly leptokurtic. That should not be! Hogg and Craig (1978) clearly state that *b* will be normally distributed. And if Hogg and Craig say so, it is so! The one thing that I can say is that the distribution, whatever its shape, is so close to normal that it would not be worth worrying about if it weren't for the fact that I had been looking for something to worry about.

The following is an empirical demonstration of these sampling distributions. The first attempt at looking at the empirical sampling distribution of *b* was done using a program called Resampling Stats by Bruce and Simon (http://resample.com/). This program draws repeated samples from defined populations and plots the resulting sampling distributions.

That is a very good program, but I haven't used it in a long time and I had trouble deciphering what I had done. So I redid it in R (similar to S_PLUS) and that is given below.

My program makes use of a simple algorithm for generating data from a population with a specified correlation (ρ).

- Draw two large pseudo-populations of *X* and *Y*. (I used 10,000 cases.)
- Standardize the two variables.
- Compute $a = \frac{r}{\sqrt{1-r^2}}$, where *r* is the desired correlation
- Compute $Z = aY + X$
- Now *Y* and *Z* have a correlation = *r*.

From a population consisting of 10,000 *X* and *Z* pairs, I drew 10,000 samples of 50 observations each. For each sample I computed b_0 and b_1, and beta (the standardized regression coefficient) and plotted their sampling distributions. I also plotted the sampling distribution of *r* for purposes of comparison. I then plotted the results as histograms and again as Q-Q plots.

The R program that does the sampling follows. In the first run of this program I set rho to .60 and *n* to 50. I drew 1000 samples with replacement.

```
# Sampling distribution of standardized regression coefficient

# Plot sampling distribution of b and beta
```

```
# Plot sampling distribution of b and beta

r_array <- c(10000); b_array <- c(10000); beta_array <- c(10000) #Create some arrays
x <- rnorm(10000,0,1)
y <- rnorm(10000, 0, 1)
zx <- (x - mean(x))/sd(x)   #Standardize the variables
zy <- (y - mean(y))/sd(y)
rho <- .60  # Choose a value for rho
a <- rho/(sqrt(1-rho^2))
zz <- a*zy + zx
cor(zy,zz)
# the correlation between zy and zz is r = .60
```

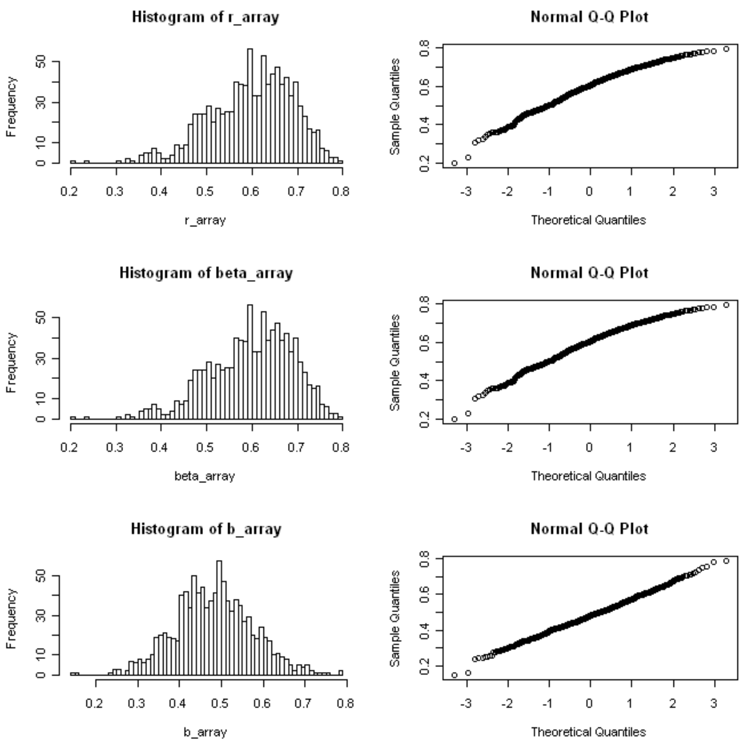
```
data <- cbind(zy, zz)
# Now create functions to calculate skewness and kurtosis
skew <- function(x) {
m3 <- sum((x - mean(x))^3/length(x))
s3 <- sd(x)^3
m3/s3
}

kurtosis <- function(x) {
  m4 <- sum((x - mean(x))^4/length(x))
  s4 <- var(x)^2
  m4/s4  - 3   #Subtract 3 so kurtosis = 0 for normal distribution
}

# Now do the resampling
for (i in 1:1000) {
  samp <- data.frame(data[sample(1:10000,50, replace = T),])  #Draw a n = 50 cases
from data
  r_array[i] <- cor(samp[,1],samp[,2])   #calculate r
  regression <- lm(zy ~ zz, data = samp)
  b_array[i] <- regression$coefficients[2]      #calculate b = the regression slope
  beta_array[i] <- b_array[i]*sd(samp[,2])/sd(samp[,1]) #calculate beta
}  # This will repeat 1000 times

# Now plot the three statistics
par(mfrow = c(3,2))
hist(r_array, breaks = 50)
qqnorm(r_array)
hist(beta_array, breaks = 50)
qqnorm(beta_array)
hist(b_array, breaks = 50)
qqnorm(b_array)
cat("\nSkew statistic for r = ", skew(r_array))
cat("\nSkew statistic for b = ",skew(b_array))
cat("\nSkew statistic for beta ",skew(beta_array))
cat("\nKurtosis statistic for r = ",kurtosis(r_array))
cat("\nKurtosis statisc for b = ",kurtosis(b_array))
cat("\nKurtosis statistic for beta = ",kurtosis(beta_array))
```

The printout for this program follows.

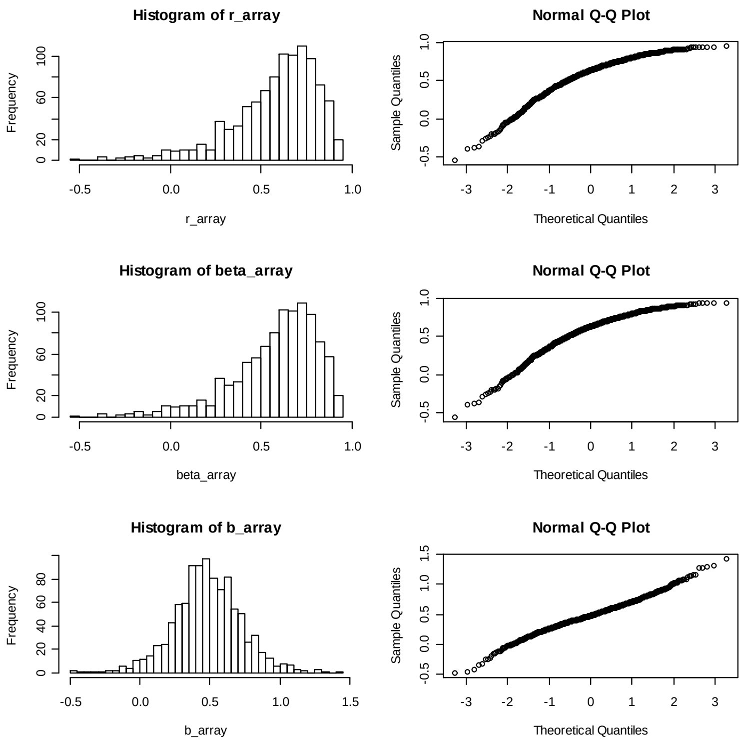


	r	beta	*b*
Mean	.598	.598	.482
St. Dev.	.091	.091	.092
skewness	-.566	-.566	.071
kurtosis	.276	.276	.172

If you look at the table you will see that the mean r = .598, which is nicely close to rho = .60. You will also notice that the distribution is negatively skews and somewhat leptokurtic. Again this is as it should be. With only one predictor, r and beta are equal, and we see that here. Looking at b we see that it has a skewness of only .07, but it does look a bit leptokuric in the table. But in the figures above, the Q-Q plot for b is remarkably straight with only a tiny bit of bumpiness at the extremes.

Now let’s do the same thing but with a much smaller sample size. I well let n = 10 instead of to.

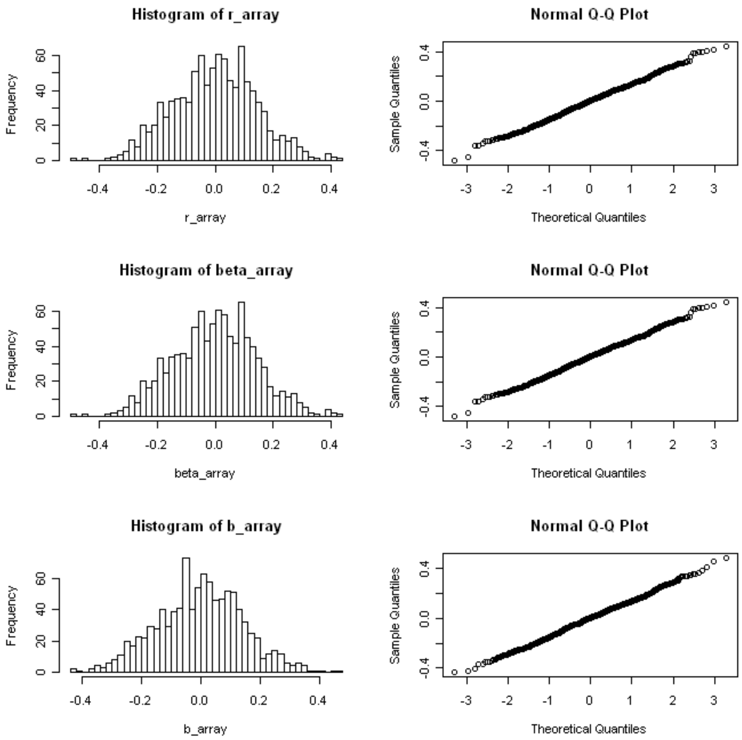
Rho = .60, *n* = 10



	r	beta	*b*
Mean	.580	.580	.482
St. Dev.	.237	.237	.247
skewness	-1.232	-1.232	-.046
kurtosis	1.823	1.823	1.066

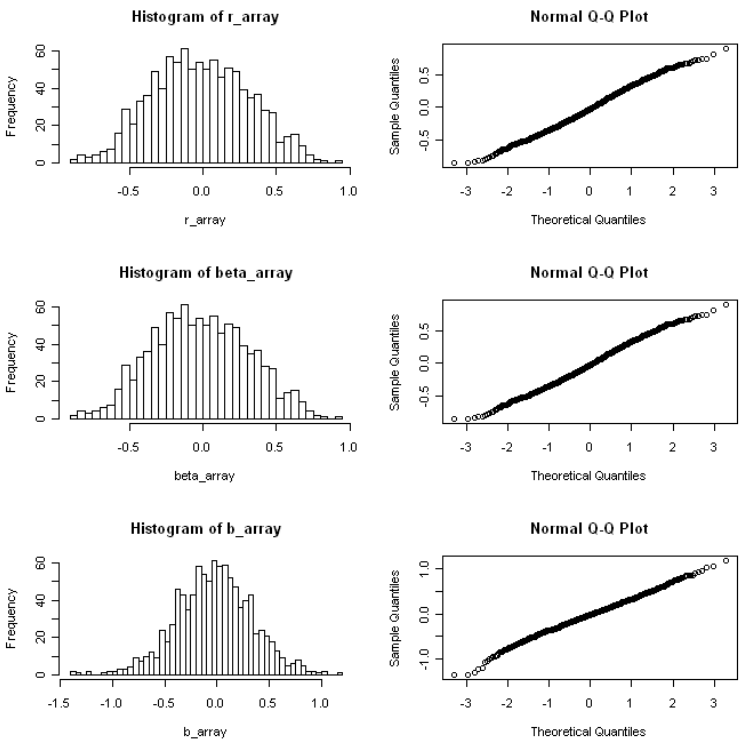
Oh Dear! This is not nice. With such a small sample size the mean correlation stayed close to .60, but the skewness of *r* and beta just about doubled and it is clear that the distributions are quite leptokurtic. The same goes for *b*, and a look at the Q-Q plot shows that the line is distinctly not straight. It looks like things fall apart for small *n*’s.

Now we will repeat the two analyses above, but with rho = 0. Here I would expect all three statistics to be centered on 0.00, and, because *n* = 50, things shouldn’t look too bad. Below is what we found.



	r	beta	*b*
Mean	-.002	-.002	-.002
St. Dev.	.143	.143	.145
skewness	-.010	-.010	-.012
kurtosis	-.119	-.119	-.058

That's not too bad. I can live with that. But what happens if we do the same but drop down to $n = 10$?



	r	beta	*b*
Mean	-.015	-.015	-.027
St. Dev.	.336	.336	.366
skewness	.046	.046	.046
kurtosis	-.529	-.529	.471

That is definitely not good! The mean *r* is -.015, which is OK. The standard error of *r* (and the other statistics) are elevated, simply reflecting the smaller *n*. But look at the kurtosis. All three distributions should be normal, but two are platykurtic and one is leptokurtic. I suspected that what we are seeing here was just a huge amount of random error, so I repeated this last example 10 times. The kurtosis for *b* was always positive, ranging from 0.296 to 10.729, with a mean of 2.003. Something is weird.

A Possible Explanation

Perhaps when Hogg and Craig (and many other people) say that *b* is normally distributed, what they really mean is that *b* is *asymptotically* normally distributed. In other words if each sample were infinitely large the distribution would be normal.

So I did it one last time, but this time with *n* = 500. I have not shown the graphics, but the table is below.

	r	beta	*b*
Mean	.012	.012	.012
St. Dev.	.045	.045	.045
skewness	.086	.086	.078
kurtosis	-.003	-.003	-.008

And if I set n = 10,000 things are even better.

Alessio Toraldo offered another explanation which I have not had the time to pursue. He suggested that instead of treating X and Y as random variables, I should examine the case where X is fixed. This is more in line with the regression approach (as opposed to correlation) and by removing one source of variance we might in fact find a normal distribution for b. I want to try that.

Another thought: I am using the random number generator in R. No random number generator is perfect, and I notice that the kurtosis of the normally distributed random variables is not 0 either. Perhaps that is part of the problem.

But Don't Give Up!

This exercise gave me something to do when I needed something to do, and I believe that the results are correct. But for all practical purposes the kurtosis in the distribution of b will not make the slightest difference to any practical analysis you want to do. You can just go ahead and believe the t test on b.