Figuring out why inf shows up in Cbootdata.mean and Kbootdata.mean

The reason that it shows up in Cbootdata.mean or Kbootdata.mean is because on out of the 1000 bootstrapped sample results in a regression result with inf.

Why would this happen?

code for regressC:

```
function c = regress(t,y)
      A = [ones(7,1),t];
      p = inv(A'*A)*A'*y;
      c = exp(p(1));
end
```

First thought:

referring back to the linearized equation: $\ln(y) = \ln(c) + kt$

if $c = 0$, then $\ln(c)$ is clearly undefined because $\ln(0)$ is undefined.

However, upon experimentation with MATLAB, it can be seen that $\ln(0)$ = -inf and exp(-inf) = 0.

So, even though the original suspicion that $c = 0$ in a log function is a valid reason to get an undefined value, it is not the cause of our in this case, as shown in the above MATLAB calculation.

Second thought:

What if, by chance, the bootstrap pulled the exact same values when it randomly selected with replacement?

To test this idea, the following code was implemented:

```
y = log([4,4,4,4,4,4,4]');
t = [2,2,2,2,2,2,2]';
u = regressC(t,y)
v = regressK(t,y)
…
u = inf
v = inf
```

When a sample that is made up of the same points is plugged into regressC or regress, it returns inf. This is expected because the points are not in an exponential relation to each other.

Referring to model equation: $y = ce^{kt}$ if all t and y are the same, c = 0 and k = 0

$$y = ce^{kt}$$
$$y = 0e^{0} = 0$$
$$\ln(y) = \ln(0) = undefined$$

Therefore, the cause of inf occurring in the Cbootdata.mean and Kbootdata.mean datasets is because the bootstrap function picked samples that were made up entirely of the same individual data points.