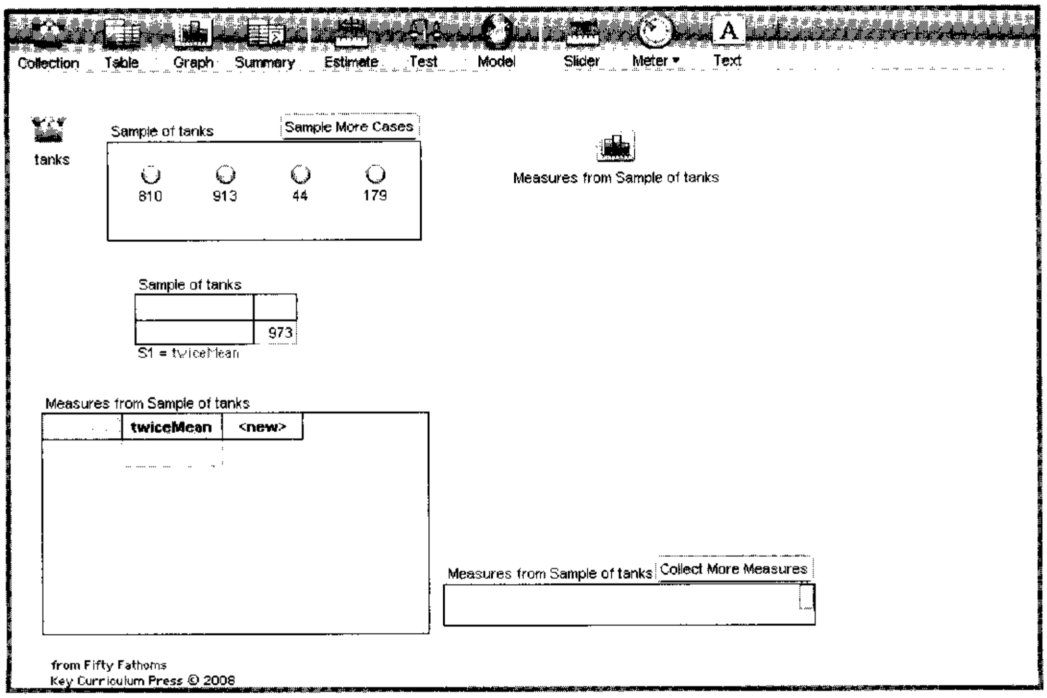
Demo 26: German Tanks

Unbiased estimators • Evaluating estimators from their sampling distributions • Even among unbiased estimators, some are better than others

What do we mean by an unbiased estimator? In this demo, we'll see at least two of them, as applied to the historical German Tanks problem. The idea is that German tanks have serial numbers, sequential, starting at 1. (Historically, the numbers were on particular tank *parts,* not the whole tanks.) You want to estimate how many tanks there are altogether. You capture a small number of tanks (four, in this case) and read their serial numbers. Assuming they are a random sample, what do you do with the numbers to get a good estimate of the total?

The complete set of tanks is the population. We get a sample of four, and we want to estimate *N,* the size of the population. It's a strange problem.

Part of the problem is deciding what we mean by a good estimate. For now, we'll ask for an *unbiased* one. An *unbiased estimator* is a procedure (for example, a formula) that produces an estimate with a special property: If you run this procedure many times, the *mean* of the estimates will be equal to the true quantity you're looking for. Of course, in practice, you use it only once. But in this demo, we'll use a strategy you should be familiar with: Knowing the population, we run our procedure on it repeatedly to see how well the procedure performs. In this case, we'll know in advance that $N = 1000$.



Sample of tanks

810 913 44 179

Sample of tanks

	973

S1 = twiceMean

Measures from Sample of tanks

	twiceMean	<new>

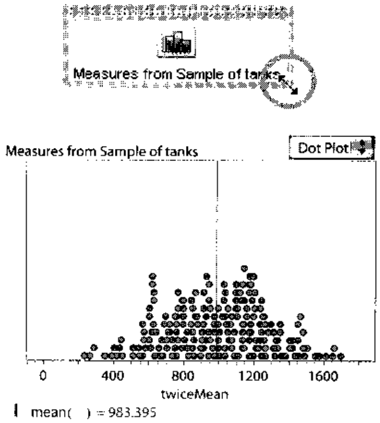
from Fifty Fathoms
Key Curriculum Press © 2008

What To Do

▹ Open **Tanks.ftm.** It will look something like the illustration.

In the upper left is the original **tanks** collection. It contains 1000 cases, each of which has a single attribute, **serial,** that runs from 1 to 1000. Next is **Sample of tanks,** which is open (you can see the blue balls). It's a sample of four cases from the **tanks** collection. Below it, in a summary table, you can see an estimate of the population based on that sample, called **twiceMean.** Below those are a measures collection (**Measures from Sample of tanks**) and its case table, both empty. At the upper right is an iconified graph.

- ▹ Press the **Sample More Cases** button. A new set of four cases shows up in the collection, and our estimate changes below. Repeat this a few times. How good is the estimate?
- ▹ Let's have Fathom collect those estimates automatically. Press **Collect More Measures** in the bottom collection. A column of numbers—200 values of **twiceMean** from 200 different samples—appears in the bottom table.
- ▹ Click on the graph icon at right to select it; then drag its lower-right corner to expand the graph. You should see a graph like the one in the illustration.



Note that the mean of these 200 estimates is close to the true number of tanks—1000—but that the individual values have quite a bit of spread. It appears that this measure is an unbiased estimator (or close to one); remember, however, that when you do this to figure out the number of tanks, you get only one sample of four—only one of those two hundred points—and you don't know which one.

Questions

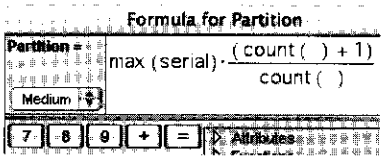
1 The measure **twiceMean** is just what it sounds like: twice the mean of the four serial numbers in our sample. Why is that a sensible estimate of the largest serial number?

2 What are the largest and smallest possible values for **twiceMean**? **Sol**

Onward!

Let's put in a different estimator of the total number of tanks—one called **Partition.** The idea is that if we pick four numbers and they're spaced uniformly over the range, they could divide the population into five equal portions. If that were true, the largest serial number would be ⅘ of the *maximum* number in the sample. This strange reasoning yields a surprisingly good estimator.[1] (To skip the next set of steps, simply open **Tanks2.ftm.**)

- ▹ First we have to make the measure. Double-click the sample collection (the one with four balls) to open its inspector. Be sure the **Measures** panel is showing (click the **Measures** tab if you need to).
- ▹ Click in **<new>** to make a new measure; enter **Partition** and press **Enter.**
- ▹ Double-click its formula box (to the far right of the name) to open the formula editor.
- ▹ Enter the formula **max(serial)*(count( ) + 1)/ count( ),** as shown in the illustration.


Formula for Partition

Partition = $\max(\text{serial}) \cdot \frac{(\text{count}(\ ) + 1)}{\text{count}(\ )}$

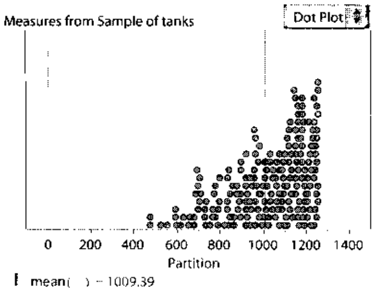
If you type the characters—including all parentheses—in order, the formula will work.

- ▹ Close the formula editor with **OK.** Then close the inspector to save screen space.
- ▹ Let's put the new measure in the table. Click the summary table to select it, then choose **Add Formula** from the **Summary** menu. The formula editor opens.
- ▹ Enter **Partition** and press **OK** to close the editor. The new value appears.

Note: This is the place you'll be if you open **Tanks2.ftm.** Also note that we used **count( )** in the fraction rather than just entering ⅘; this will keep the calculation correct in case you decide to change the sample size.

[1] Though not unbiased. It's systematically high by 1—too small a difference here to matter.

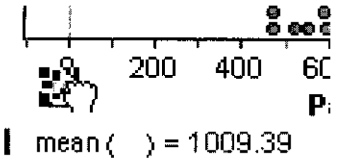
- ▹ Press **Sample More Cases** (the top button) a few times to see the two values in the table. See whether you think that the new measure, **Partition**, is a good estimator.
- ▹ Press **Collect More Measures** in the bottom collection to do so automatically. Now there are two columns in the table—200 values of each measure.
- ▹ Drag the column head for **Partition**—the name itself—to the axis of the graph, replacing **twiceMean.**



You should see a graph like the one in the illustration. What does it tell you?

It seems that, though the distribution is very different from the one we got with **twiceMean, Partition** is also close to being an unbiased estimator, since its mean is close to the truth.

- ▹ Let's compare the two. Drag the name **twiceMean** to the horizontal axis of the graph and drop it on the "plus" that appears near the left end of the axis.



You should see both distributions, one above the other.

- ▹ Look at box plots as well as the default dot plots.

Even though we're pretty sure any estimate is not exactly correct, we can say something about whether we would rather have it high or low, and by how much. It may even be that a *biased* estimator would be better, depending on the circumstances.

Questions

3 What are the largest and smallest possible values for **Partition**?

4 Which distribution has the greatest range?

5 Which has the largest median?

6 Which estimator would you rather use to judge how many tanks there were? (Think about the consequences of overestimating or underestimating the number of tanks.)

Challenges

7 Add estimators and assess them. Your own are best, but here are two to consider: twice the *range* of data (formula: **2 * (max(serial) – min(serial)**) and five (or **(count( ) + 1)**) times the minimum.

8 Rerun the simulation with a sample of ten tanks instead of only four. How do the distributions of the estimators change? To change that sample size—to capture more tanks—open the inspector for the **Sample of tanks** (opened) collection, and change the number sampled from 4 to 10.
Sol

9 It looks as if **Partition** is a better estimator than **twiceMean.** But **twiceMean** uses all of the data, whereas **Partition** uses only one point—the maximum. Try to explain how that can be.

10 You could argue that it would be better to overestimate the number of tanks. Invent a situation where it would be better to *underestimate* the number of something that had serial numbers. What estimator would you use in that case?