Pooled Testing

Those who lived through the covid pandemic are certain to remember it for a long time. Masks, mandates, social distancing, vaccines, virtual meetings, and for many of us, losing loved ones. We will also remember testing for covid with nasal swabs and at-home kits.

It never caught on during the covid pandemic, but pooled testing was considered because it could reduce the number of tests needed and save time (Mandavilli, 2020). Pooled testing was used successfully by the United States military during WWII to test men for syphilis (Dorfman, 1943).

The logic of pooled testing is straightforward. A university, for example, could take saliva or nasal swab samples from each student and test them individually or it could combine a part of each sample from several people into a single group and test the pooled sample. If it is negative, then all of the individuals in the combined pool are negative and we have saved on testing every person in that group. If the pooled sample is positive, then individual tests would be performed on the reserved parts of each individual's sample to determine exactly who is infected.

This leads to a crucial question: *What is the optimal group size?* The bigger the group, the lower the number of groups tested, but the higher the chances a group is positive and then everyone in the group has to be tested (we ignore the possibility of sub-group testing, false positives or negatives, and other complications).

We solve this optimization problem by constructing an Excel spreadsheet and using Monte Carlo simulation. We proceed step-by-step and reveal Excel functions and tools as we create our model of pooled testing.

The Data Generation Process

The first thing we need to do is implement the random process by which some people get infected and others do not. We do this by drawing a random number and comparing it to a threshold value so we get either a zero (not infected) or a one (infected).

We make the simplifying assumption that everyone has the same likelihood of catching the virus, say 5%. This is an *exogenous variable* (also called a *parameter*) in our model and it will serve as our threshold value for determining if someone is infected.

STEP Enter 5% in cell A1 of a blank spreadsheet and label it as “infection rate” in cell B1. Save the Excel file (*PooledTesting.xlsx* is a good name).

Cell A1 displays 5%, which is the same as 0.05 in decimal notation. The number 0.05 is what the spreadsheet stores in its internal memory. It is worth remembering that what is displayed may be different from what is stored.

EXCEL TIP Name cells, especially if you have many or complicated formulas. We have been using cell addresses and we can, of course, refer to cell A1 in a formula, but cell addresses can be difficult to read. It is good practice to name a cell or a cell range so that formulas can use natural language to reference cells.

STEP Name cell A1 *InfectionRate* because this will make our future formulas easier to understand. If needed, search Excel’s Help for “names in formulas.”

Next, we incorporate randomness. As you know, Excel draws uniformly distributed random numbers in the interval from zero to one with the RAND() function.

STEP In cell A3, enter the formula *=RAND()*.

Just like the free-throw shooting and coin-flipping examples, we use Excel’s random number generator to determine whether or not a person gets covid. We use an IF statement to group the RAND-generated values into two categories, zero and one.

STEP In cell A4, enter the formula *=IF(A3<InfectionRate, 1, 0)*.

You probably see a zero displayed in cell A4, if not, press F9 (you may have to use the *fn* key). Zero means the person is not infected.

STEP Press F9 repeatedly to recalculate the sheet until you see a one in cell A4. The chances are only 1 in 20 so be patient.

As you recalculated, cell A3 constantly changed, but cell A4 changed only if A3 switched from being above or below the infection rate. Cell A4 is a *binomial random variable* because it can take only the values zero or one.

Now that we know how to implement a random process that outputs whether or not an individual is infected, we can create an entire population of people, some who get infected the virus and others with do not.

STEP In cell C1, enter the formula *=IF(RAND()<InfectionRate,1,0)*.

Notice how we directly embedded the RAND() function in the cell formula. We do not know which random number was drawn, but we do know if it was less than 5% because then cell C1 would display "1".

STEP Fill down this formula all the way to cell C1000.

As you scroll back up to the top row, you will see a sprinkling of ones among many zeroes. With a 5% infection rate, roughly one in twenty cells will have random number draws less than 5% and, therefore, show the number one.

The fact that each cell in column C stands alone and does not depend on or influence other cells means we are assuming *independence*. In our model, one person with the virus does not affect the chances of anyone else being infected. This condition is surely violated in the real world. We should make the chances of infection depend on whether people with whom they come in contact have the virus to improve our analysis.

However, since our focus here is on showing how pooled testing works, we will not model infection dependent on people nearby. This would be a fun project where you would create clusters of cells and if one got sick, the nearby cells would have a much higher chance of infection.

How many people in our population of 1,000 are infected?

STEP Enter the formula *=SUM(C1:C1000)* in cell D1 and the label, “Number infected” in cell E1.

You will see a number around 50 in cell D1. The number of infected people is not always exactly 50 because there is chance involved in who gets infected.

STEP Recalculate by pressing F9 a few times to get sense of the variability in the number of infected people.

The total number of infected people can be less than 40 or more than 60, but that is not common. Usually, there are around 45 to 55 infected. There is no doubt that the number of infected people is a random number since it is bouncing around when you recalculate the sheet. It makes common sense that adding binomial random variables will produce a random outcome.

We can make it easier to identify who is infected with a spreadsheet’s conditional formatting capability. This offers the viewer visual cues that make data easier to understand.

STEP Select the entire column C and apply a formatting rule that highlights, with color, cells with a value of one. Choose font and fill colors that you think emphasize being infected. If needed, search Excel’s Help for “conditional formatting.”

Now, when you scroll down, it is easy to see who is infected. Recalculation changes who is infected—it is as if we rewound and replayed the world with each press of F9.

Having implemented the chance process for being infected or not, we turn to pooled testing. Instead of testing each person, we can group individuals and test their combined sample. If the pooled sample tests positive, then we know at least one person is infected; if not, we know no one is infected and we do not have to test each individual in the group.

Instead of directly choosing the number of groups, it is more convenient to make the choice variable the size of the groups. Choosing group size determines how many groups we have since

$$\text{Number of Groups} = \frac{\text{Population}}{\text{Group Size}}$$

With our population of 1,000 people, a group size of 100 means we will have 10 groups. Intuitively, with an infection rate of 5%, 100 people in a group means that at least one person will be infected and the group is probably going to test positive. We can make our intuition more convincing by computing the exact chances.

STEP Begin by entering 100 in cell D3 and the label "Group Size" next to it in cell E3. Enter the formula *=1000/D3* in cell D4 and the label "Number of Groups" in cell E4.

An infection rate of 5% means each person has a 95% chance of not being infected. If there are two people (assuming the chances of infection are independent), then there is a $0.95 \times 0.95 = 0.95^2 = 0.9025$, or 90.25%, chance that neither are infected. This means there is a $100\% - 90.25\% = 9.75\%$ chance that at least one of the two people is infected.

What are the chances that at least one person is infected in a group of 100 people? Remember, if even one person is infected in a group, we have to test everyone in the group to find out who is infected.

STEP In cell D6, enter the formula *=(1−InfectionRate)ˆD3* and label "prob no one in the group infected" in cell E6. Format D6 as a percentage so that it displays 0.59%.

Next, we compute 100% minus the probability that no one in the group is infected to find the probability that at least one person is infected.

STEP In cell D7, enter the formula *=1−D6* and label "prob at least one in the group infected" in cell E7. Format D7 as a percentage (if needed).

With an infection rate of 5%, doing pooled testing with a group size of 100 is wasteful. After all, it seems overwhelmingly likely (over 99%) that we will have to test everyone in each of the ten groups so we would end up doing 1,010 tests.

Can we make our spreadsheet show how many people are infected in each group and confirm the computations we just made? We can, but the approach described below uses a function which may be unfamiliar and advanced—the OFFSET reference function. Thus, we proceed slowly.

STEP In cell G1, enter the formula *=OFFSET(E1,3,0)*.

Cell G1 computes the number of groups because the OFFSET function went to cell E1 (the first argument in the function), then went three rows down (the second argument). The third argument is zero, so it stayed in column E. If the movement arguments are a positive integer we move down or right; negative integers move us up or left.

STEP Change the formula in cell G1 to *=SUM(OFFSET(D1,0,0,3,1))*.

Why does G1 display D1 plus 100? The two zeroes means it did not move from the reference cell D1, but the fourth and fifth arguments control the height and width, respectively, of the cell range. Therefore, the formula says to add up the values in cells D1, D2 (which is blank), and D3 (100).

We want to add up the values in column C into ten separate groups of 100 each. We can modify our OFFSET function to do the first group of 100.

STEP Change the formula in cell G1 to *=SUM(OFFSET(C1,0,0,100,1))*. To be clear, change the D1 to C1 and the 3 to 100.

The value reported in cell G1 is the sum of the first 100 people in the population. How can we get the second group of 100 people?

STEP Change the formula in cell G1 to *=SUM(OFFSET(C1,0,0,100,1))* and fill it down to cell G2.

Adding the dollar signs made C1 an absolute reference so we kept our C1 starting point in cell G2, but we need to change the formula so it adds up the number of infected people in the second set of 100. We do that by changing the second argument because it controls how many rows to move from the reference cell.

STEP Change the formula in cell G2 to *=SUM(OFFSET(C1,100,0,100,1))*.

EXCEL TIP Cell G2 reports how many people are infected in the second group of 100. We could fill down eight more cells and then change the second argument manually to 200, 300, and so on, but this is poor spreadsheet practice. *Never manually repeat* the same entry or an ordered sequence (e.g., numbers or dates). In addition, you want to *maximize flexibility.* Hard-coding numbers, like 100, in formulas is poor practice because you might want to change that number in the future.

In this case, we want our groupings to respond to changes in cell D3. If, for example, we have a group size of 50, we would then have 20 groups. We want the spreadsheet to automatically show how many people are infected in each of the 20 groups.

This task requires that we modify the second and fourth arguments. The fourth argument is the group size, which is simply cell D3. The second argument is more complicated. It is zero for the first group, then increases by D3 for each group. One way to do this is to use the ROW function, which returns the row number of a cell.

STEP Replace the formula in cell G2 with *=ROW(D6).*

Cell G2 displays 6, the row number of cell D6. What happens if the ROW function does not have an argument?

STEP Change the formula in cell G2 to *=ROW()* and fill it down to G10.

Without an argument, the ROW function returns the row number of the cell which contains ROW() in the formula. We can use this to create a series that starts at zero and increases by the amount in cell D3.

STEP Change the formula in cell G2 to *=(ROW()–1)*\$D\$3* and fill it down to G10.

We can use our ROW function strategy in the OFFSET function's second argument to create a formula that gives us the number of infected people for any group size from 2 to 500 entered in D3. A "group" of one is simply individual testing and with 1,000 people, a group of size two yields 500 groups. Choosing a group size of 500 gives us 2 groups.

We start with G1 (notice that ROW()−1 is zero for G1 so the second argument evaluates to zero) and fill down to G500 (since 500 is the maximum number of groups we can have).

STEP Change the formula in cell G1 to *=SUM(OFFSET(C1,(ROW()−1)*D3,0,D3,1))* and fill it down to G500.

Cells G1 to G10 now display the number of infected people in each of the ten groups of 100 people.

STEP Click the letter C in column C to select the entire column, and then click the *Format Painter* button (in the *Home* tab in the ribbon, or top menu). Now click the letter G in column G.

You applied the formatting in column C, including your conditional formatting to highlight the infected people, to column G. It (probably) shows all of the groups highlighted, but it will soon come in handy when we lower the group size so some groups have no infected people.

EXCEL TIP It is good practice to *include checks* in your spreadsheets. In this case, an easy check is to see if the sum of infected people in the ten groups equals the total number of infected people in the population in column C.

STEP In cell H1, enter the formula *=SUM(G1:G500)* and the label "check" in cell I1, then recalculate the sheet a few times.

It is easy to see that cells D1 and H1 are the same. If not, something is wrong and you will have to go back to each step to find and fix the mistake.

STEP Change cell D3 to 200 and recalculate the sheet a few times.

Now only five cells in column G have nonzero values, representing the number of infected people in each of the five groups.

Group sizes of 100 and 200 are way too big to be the optimal size because we are extremely unlikely to get a group where everyone tests negative so we almost always have to test everyone in the group. We need to try much smaller group sizes.

STEP Change cell D3 to 20 and recalculate the sheet a few times.

Now we are really getting somewhere. Column G is showing the number of infected people in each of 50 groups. You can see values of 0, 1, 2, 3, and, less frequently, higher numbers. We love to see zeroes because they mean we do not have to test anyone in that group and we saved 20 tests.

How many tests will we have to run in total? The COUNTIF function allows us to count the number of cells in a range that meet a specific condition.

STEP In cell H2, enter the formula *=COUNTIF(G1:G500,“> 0”)* and the label “number of groups to test” in I2.

The COUNTIF function reports the number of cells in the range G1:G500 that have a value greater than zero. If we multiply this by the group size, we know how many individual tests we have to run. This is added to the number of group tests to give us our total number of tests.

STEP In cell H3, enter the formula *=H2*D3* and the label “tests from infected groups” in I3. In cell H4, enter the formula *=D4+H3* and the label “total tests” in I4.

Notice that, once again, we did not hard-code numbers (like 20 for group size) into the formula. We want our spreadsheet to respond to changes in group size (cell D3) automatically.

Cell H4 is certainly giving us good news. Total tests is a random variable that is almost certainly less than 1,000. You are likely to see numbers around 690 tests, give or take 70 or so. This is about a 30% decrease in the number of tests from the 1,000 required by individual testing.

Finding the Optimal Group Size

We have implemented in our spreadsheet a stochastic (or chance) process of getting infected and demonstrated the power of pooled testing. Grouping allows us to save on testing because any groups that have no infected people means we do not have to test those individual samples.

Our spreadsheet shows that a group size of 20 is better than individual testing, but we do not want to do merely better than 1,000 tests. We want to perform the fewest number of tests. Our fundamental question is: *What is the optimal group size?*

There is a complication that we have to confront to answer our question: total tests is a random variable. We cannot just look at a single outcome because we know there is chance involved. Suppose two dice are on a table each showing 1 and I asked you to guess the sum of the next roll. You would not guess 2 because you know that is really unlikely.

We deal with the fact that total tests is a random variable by focusing on the *expected value* of total tests. This is what we would typically observe. The best guess for the sum of two dice rolls is 7, the expected value. We need to find the expected value of total tests for a given group size so we can figure out which group size minimizes it.

There are mathematical rules for computing the expected value, but we will use Monte Carlo simulation. This approach is based on the idea that we can simply run the chance process (throwing two dice or hitting F9) many times and directly examine the results. We can compute the average of many repetitions (like rolling dice many times) to give us an approximation to the expected value.

So, we seek the group size that minimizes the expected value of total tests, which we will approximate by simulation. We will run many repetitions (recalculating the sheet repeatedly) and keep track of the total number of tests to see how many total tests we can expect to run as we vary group size.

While there are many simulation add-ins available for Excel, we can easily run a simulation using Excel's *Data Table* tool. It was not designed to run a simulation, but to display multiple outcomes as inputs vary. To do this, it recalculates the sheet, which enables us to perform Monte Carlo analysis.

STEP In cell L1, enter the number 1 and enter 2 in cell L2. Select both cells and fill down to row 400 so that you have a series from 1 to 400 in column L.

Next, we provide the cell that we wish to track, total tests.

STEP In cell M1, enter the formula =*H4*.

We are now ready to create the Data Table.

STEP Select the cell range L1:M400, click the *Data* tab in the ribbon, click *What-If Analysis* in the *Forecast* group, and select *Data Table...* A keyboard shortcut is alt-a-w-t.

Excel pops up the *Data Table* input box.

STEP Click in the *column input cell* field, click on cell K1, and click *OK*.

Clicking on an empty cell would be meaningless if we were using the *Data Table* tool for its intended purpose, which is to show how an input cell affects a formula in another cell. All we want, however, is for Excel to recalculate the sheet and show us the total tests for that newly recalculated population in column C.

The display in column M shows 400 repetitions of hitting F9 and keeping track of total tests. This is exactly what we want because now we can take the average of the total tests values to approximate the expected number of total tests when the group size is 20.

But before we do this, let's be clear about what a Data Table is actually doing. Be aware in the next step, however, that if you double-click on a cell in column M or click in the formula bar, you might get trapped in a cell. **If you get stuck, press the *esc* (escape) key to get out.**

STEP Click on a few cells from M2 to M400 to see that they have an *array formula*: {*=TABLE(,K1)*}.

Excel has a friendly front-end via *Data: What-If Analysis: Data Table...* to create an array formula (indicated by the curly brackets, {}) that can display multiple outputs. You cannot change or delete an individual cell in the range M2:M400. They are, in a sense, a single unit, sharing the same formula.

You might also notice that the sheet is much slower as we enter formulas or press F9. This is due to the Data Table. Excel now has many more cells to recalculate and evaluate. We could do many more repetitions (usually simulations have tens of thousands of repetitions), but the delay in recalculation is not worth it. With 400 repetitions, the approximation is good enough for our purposes.

STEP In cell N1, enter the formula *=AVERAGE(M1:M400)* and the label "approximate expected value of total tests" in cell O1.

Cell N1 is our simulation's approximation to what we want to minimize. It gives us a handle on the center of the *sampling distribution* of the statistic, totals tests. A statistic is a recipe for what to do with observations (in this case, given by the formula in cell H4). If we make a *histogram* of the data in column M, we get an approximation to the sampling distribution of total tests.

Excel 2016 or greater is needed to make the histogram chart. This is not the Histogram option in the Data Analysis add-in (from the Analysis Tool-Pak). The histogram chart allows for dynamic updating and is a marked improvement over the histogram in the Data Analysis add-in.

STEP Select cell range M1:M400, click the *Insert* tab in the ribbon, and select *Histogram* from the *Charts* group. It is in the *Statistic* chart group and, of course, in the collection of all charts (available by clicking the bottom-right corner square in the *Charts* group).

The default bin widths are a little too big, but they are easy to adjust.

STEP Double-click the chart's x axis and in the *Axis Options*, set the *Bin Width* to 20. Make the title "Approximate Sampling Distribution of Total Tests."

The chart is an approximation because it is based on only 400 repetitions. The exact sampling distribution of total tests would require an infinite number of repetitions. We can never get the exact sampling distribution or the exact expected value via simulation, but the more repetitions we do, the better the approximation.

Even with just 400 realizations of total tests, the graph looks a lot like the classic, bell-shaped distribution of the normal (or Gaussian) curve. The center is the expected value of the total tests we will have with a group size of 20 and the dispersion in total tests is measured by its *standard error*.

STEP In cell N2, enter the formula *=STDEV.P(M1:M400)* and the label "approximate standard error of total tests" in cell O2.

The standard error of total tests tells us the variability in total tests. It is a measure of the size of the typical bounce in total tests.

STEP Press F9 a few times and watch cell H4.

Cell H4 is bouncing. It is centered around 690 and jumps by roughly plus or minus 70 total tests as you hit F9. You can also scroll up and down column M to see that the total tests numbers are around 690 ± 70.

Simulation can not give us the exact standard error, but the standard deviation of our 400 realizations of total tests is a good approximation of the standard error.

There are many ways to be confused here. One of them is to fixate on the computation of the standard deviation. We used STDEV.P (for population) instead of STDEV.S (for sample) because we are not using the standard deviation to estimate the population standard deviation so we do not need to make a correction for degrees of freedom. Although the population standard deviation is correct, this makes almost no difference with 400 numbers.

STEP In cell N3, enter the formula *=STDEV.S(M1:M400)* and compare the result to cell N2.

The emphasis on population versus sample standard deviation in many Statistics courses is only relevant for small sample sizes, say less than 30 observations. As the number of observations rises, the two grow ever closer.

To summarize, cells N1 and N2 tell us that we can expect to perform about 690 total tests, give or take roughly 70 tests. These are the numbers reported at the end of the previous section. This is for a group size of 20. Can we do better? We get to choose the group size so we should explore how the expected number of total tests responds as we vary group size.

STEP Change the group size (in cell D3) to 10 and press F9 a few times. Which specific cell should you focus on and what do you conclude?

The cell we care about the most is cell N1 because it tells us (approximately) the expected number of tests we will have to run. Cell N1 is reporting good news. We can expect to perform about 500 ± 50 total tests. That beats the group size of 20 by almost 200 tests, on average, and is a large saving of a half versus 1,000 individuals tests.

Why does a group size of 10 do better than 20? There is evidence in various cells of the spreadsheet of what is happening. Lowering the group size from 20 to 10 increased the number of group tests from 50 to 100 (see cell D4), but the number of infected groups only went up a little bit (from roughly 32 to 40) and the groups are now much smaller. This is where the big savings are—instead of $32 \times 20 = 640$ tests, we only have to run, on average, $40 \times 10 = 400$ tests with a group size of 10.

STEP Confirm the claims above by switching back and forth from 10 to 20 in cell H4. Press F9 and read the various cells and the chart.

Spreadsheets are powerful because they can display a lot of information and dynamically update when you make changes. Your job is to make comparisons and process the information.

Can we do even better than a group size of 10?

STEP Change the group size (in cell D3) to 5.

Amazing! Cell N1 fell again. The expected number of total tests is now about 425 (426.22 is a more exact answer, found by analytical methods) give or take roughly 30 tests. That is a gain of almost 60% versus individual testing. Pooled testing saves a lot of tests compared to individual testing.

As before, the number of groups we have to test has risen (this time to 200), but many groups are found to be uninfected. Cell D6 reports a 77.4% chance that no one in a five-person group will be infected. Thus, even though we test more groups, we more than make up for this because many groups test negative, saving us the need to test five people in the group.

The group size of 5 is, in fact, the optimal solution and answer to our question. Figure 3.10 reveals that we traveled down the expected number of total tests curve as we changed group size from 20 to 10 and finally 5.

Figure 3.10 makes it easy to see that a group size of five is the minimum of the expected number of total tests curve, but it also reveals the trade-off involved. The two curves are added up to produce the top, total curve. At 10, we test 100 groups (the bottom curve) and we add that to 400 (the expected number of tests from positive groups) and this gives 500 (the top curve).

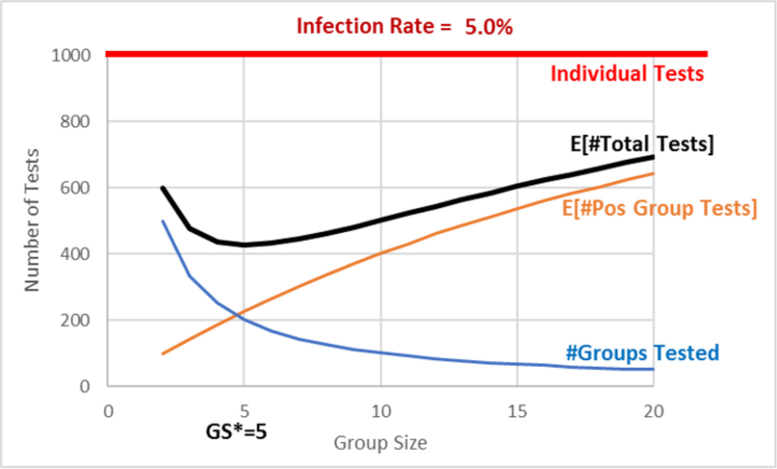

Figure 3.10: The expected number of tests as a function of group size.

When we moved from 10 to 5, we added 100 tests (the bottom curve), but saved about 175 tests (the middle curve), lowering our expected total tests from 500 to 425. We cannot do any better than 425 total tests. Further reduction in group size will increase total tests.

Comparative Statics Analysis

We can ask another question which, again, shows off the power of spreadsheets: what happens if the infection rate changes, say to 1%—what would be the optimal group size?

This kind of question is called *comparative statics analysis* because we want to know how our solution responds to a shock. We want to compare our initial, optimal group size of five when the infection rate was 5%, to the new solution when the infection rate is 1%. This comparison reveals how the shock (changing the infection rate) affects the optimal response (group size).

STEP Change cell A1 to 1%, then use the spreadsheet to find the optimal group size. What group size would you recommend? Why?

You may have struggled with this because it turns out that the total tests curve is rather flat at its minimum. Thus, a simulation with 400 repetitions does not have the resolution to distinguish between group sizes in the range from 8 to 14 or so. Figure 3.11 makes this clear.

The exact answer for the optimal group size is, in fact, 11 groups. It has an expected number of total tests of 195.57 (again, using analytical methods). Choosing group sizes of 10 or 12 lead to a slightly higher number of total tests—although it is impossible to see this in Figure 3.11.

An infection rate of 1% shows simulation may not be an effective solution strategy for every problem. Of course, you could create a *Data Table* with more repetitions, but using simulation to distinguish between group sizes 10 and 11 requires a *Data Table* so large that Excel would be unresponsive.

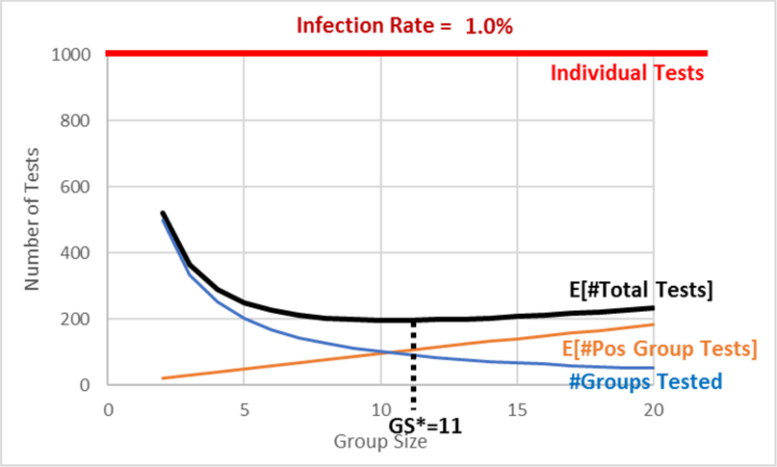

Figure 3.11: Optimal group size with a 1% infection rate.

As mentioned earlier, there are many Excel Monte Carlo simulation add-ins and they can do millions of repetitions. We used this free one in earlier work: tiny.cc/mcsim. Even if we ran enough repetitions to see that 11 is the optimal solution, you should remember that simulation will never give you an exact result because it can never do an infinity of repetitions.

The good news is that any group size around 10 is going to be a little under 200 which is an 80% improvement over individual testing. There is no doubt about it—pooled testing can be a smart, effective way to reduce the number of total tests performed.

Our comparative statics analysis tells us that the lower the infection rate (from 5% to 1%), the bigger the optimal group size (from 5 to 11) and the greater the savings from pooled testing versus individual testing (from about 675 to 800 tests).

Finally, if you carefully compare Figures 3.10 and 3.11, you will see that the *#Groups Tested* curve (a rectangular hyperbola since the numerator is constant at 1,000) stays the same in both graphs. Changing the infection rate shifts down the *E[#Pos Group Tests]* relationship and this brings down and alters the shape of the *E[#Total Tests]* curve.

Comparative statics analysis shows that pooled testing is more effective when the infection rate falls. A lower infection rate means we can have bigger groups and yet they may still have no infected individuals in them.

Takeaways

Pooled testing means you combine individual samples. A negative test of the pooled sample saves on testing because you know all the individuals in the group are not infected.

There is an optimization problem here: too big a group size means someone will be infected and so you have to test everyone in the group, but too small a group size means too many group tests. The sweet spot minimizes the total number of tests.

The optimal group size depends on the infection rate. The smaller the rate, the bigger the optimal group size.

By creating this spreadsheet, you have improved your Excel skills and confidence in using spreadsheets. You added to your stock of knowledge that will help you next time you work with a spreadsheet.

The OFFSET function is really powerful, but it is difficult to understand and apply.

The Data Table is meant for what-if analysis, but it can be used as a simple Monte Carlo simulation tool. Each press of F9 recalculates the sheet and the Data Table.

You also learned or reinforced a great deal of statistical and economics concepts. Economics has a toolkit that gets used over and over again—look for similar concepts in future models and courses. Try to spot the patterns and repeated logic. Although it may not be explicitly stated, getting you to think like an economist is a fundamental goal of almost every Econ course.

Reference was made several times to the analytical solution. This was not shown because the math is somewhat advanced. To see it, download the PooledTesting.xlsx file from tiny.cc/busanalyticsexceland go to the *Analytical* sheet.

One methodology issue that is easy to forget, but crucial, is that we made many simplifying assumptions in our implementation of the data generation process. There may be other factors at play in the spread of covid or how tests actually work that affect the efficacy of pooling. Spatial connection was mentioned as something that would violate the independence assumed in our implementation. Another complication is that, "A positive specimen can only get diluted so much before the coronavirus becomes undetectable. That means pooling will miss some people who harbor very low amounts of the virus." Wu (2020)

Our results apply to an imaginary, perfect world, not the real world. We need to be careful in moving from theory to reality. This requires both art and science.

The introduction cited Dorfman as writing a paper on pooled testing back in 1943. It is a clever idea that you now understand can be used to greatly reduce the number of tests, which saves a lot of resources. Perhaps you will not be surprised to hear that Robert Dorfman was an economist.

References

Dorfman, R. (1943) "The Detection of Defective Members of Large Populations," *Ann. Math. Statist.* 14, no. 4, 436—440,
projecteuclid.org/euclid. aoms/1177731363

Mandavilli, A (2020) "Federal Officials Turn to a New Testing Strategy as Infections Surge," *The New York Times*, July 1, 2020,
www.nytimes.com/2020/07/01/health/coronavirus-pooled-testing.html

Wu, K (2020) "Why Pooled Testing for the Coronavirus Isn't Working in America," *The New York Times*, August 18, 2020,
www.nytimes.com/2020/08/18/health/coronavirus-pool-testing.html