



Experimental Research

FTS has its origins in research. It was originally designed by Professors John O'Brien and Sanjay Srivastava, and the first research paper using FTS came out in the *Journal of Finance* in 1991 ("*Dynamic Stock Markets with Multiple Assets: An Experimental Analysis*," by O'Brien, John; Srivastava, Sanjay; *Journal of Finance* December/1991, Volume: 46, Issue: 5, Pages: 1811-38). Since then, it has been used in a variety of studies, particularly for financial market experiments.

Market Experiments

The FTS Interactive Markets continue to provide an easy to use and comprehensive framework for financial market experiments. It is quite straightforward to conduct experiments on information aggregation, speculative bubbles and overreaction to information, excess volatility and trading, and so on. The system also allows for different microstructures (call markets and continuous double auctions, quote and order driven markets), as well as very general payoff structures, allowing for a wide range of incentives to be implemented in an experimental setting. The entire experiment is defined in a spreadsheet. The easiest way to see its capabilities is to look at our [existing cases](#) (for example, case RE2, used in the 1991 *Journal of Finance* article). Then, look at the spreadsheet that defines the case. The workbook containing case RE2 is available [at this link](#). A description of the various parts of the spreadsheet is given [here](#).

If you want to conduct market experiments, please contact us, and we can help you set up your experiment.

Beyond markets, we also provide a "generic" experimental system, where you can run almost any experiment you like, including auctions, game theory experiments, behavioral finance experiments, and so on. This is described next.

The FTS Experimental System

The FTS Experimental System is a flexible and simple platform for conducting experimental research.

- You specify the entire experiment in an Excel spreadsheet.
- You run the FTS Experimental Server
- The FTS Experimental Clients connect to the Server
- The server reads the spreadsheet and sends information to the clients
- The clients make decisions as needed for the experiment
- The server retrieves these decisions and puts them back in the spreadsheet

This design has many advantages.

- First, it makes the experiment very easy to test and debug; you can test everything directly in the spreadsheet, without having to run the system
- Second, you can use all of Excel’s capabilities, including VBA functions, to simplify tasks such as summarizing the results of the experiment and making any other calculation
- Third, its “what you see is what you get.” What the clients see is exactly what you designed in the spreadsheet, including colors, fonts, etc.

To see how it works, we now work through some examples.

Example 1: Individual Choice

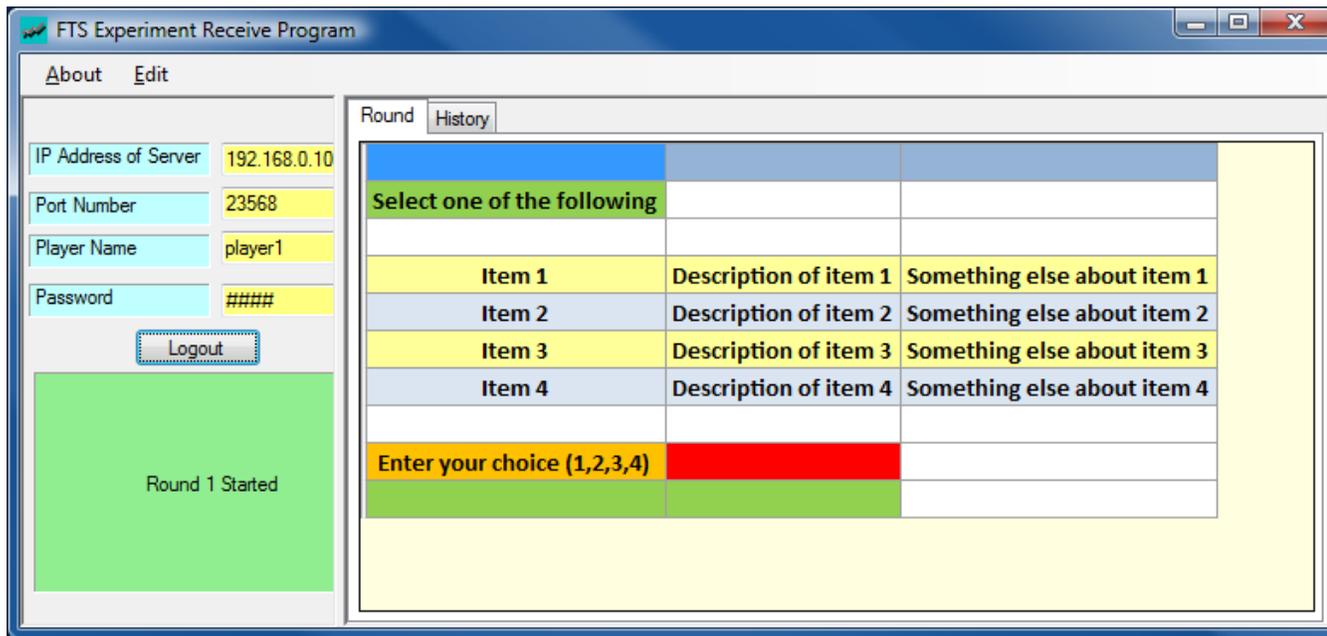
This is the simplest type of experiment to construct. Here is the relevant part of the spreadsheet:

Round 1	Select one of the following		
	Item 1	Description of item 1	Something else about item 1
	Item 2	Description of item 2	Something else about item 2
	Item 3	Description of item 3	Something else about item 3
	Item 4	Description of item 4	Something else about item 4
	Enter your choice (1,2,3,4)		
Round 2	You were asked to select		
	Item 1	Description of item 1	Something else about item 1
	Item 2	Description of item 2	Something else about item 2
	Item 3	Description of item 3	Something else about item 3
	Item 4	Description of item 4	Something else about item 4
	You chose		
	You won		

There are two rounds. The participant is shown information and makes a decision in round 1. You can have as many rows and columns as you need, and there can be multiple decisions that a participant can make (for example, a first choice and a second choice). The results of the decision are shown in round 2. The formula for the final outcome (“You won” in this case) can depend on anything in the entire spreadsheet and can be calculated using an Excel macro, so there are very few limits to what you can do. In particular, the outcome can depend on the actions of others, as you will see in the second example.

- The structure makes it easy to debug. Instead of having to run the experiment, you simply enter a value in the spreadsheet and see if the rest of it responds correctly. So type in a choice in the red cell. The formula for the blue cell is simply “=RedCell.” Suppose (in this simple case) that winnings are 100*the choice. Then, the pick cell’s formula is =”100*RedCell.” So if you type in 2 in the red cell, pink green cell should show 200.
- Of course, you have to clear out the (input) cells before you run the experiment. This is easily done though a macro (we can help you construct the macros).

What happens in an experiment? The experimenter runs the server (more on this below). When you start the first round, this is literally what the participant sees on the client:



- Note that the colors and fonts are transmitted exactly as in the spreadsheet. The items are centered; you can set the alignment of each cell (this is defined in another part of the spreadsheet). You can also specify the characters that can be entered, and specify which cells can have entries. In this case, the only cell where the participant can enter anything is the red cell, and they can only enter an integer between 1 and 4.
- The experimenter can retrieve any entries at any time. This lets you monitor who had not entered data; you can then send them a message from the server or go and tell them to make their decision.
- When you retrieve the values, they go into the spreadsheet; so what the participant enters into the red cell gets transferred to the red cell in the spreadsheet. The spreadsheet then calculates whatever is linked to the red cell, including the round 2 part. But the participant does not see that until you start round 2.

So setting up the experiment is easy; you simply have to create the spreadsheet (specify the number of rows and columns per round, fill in the entries, and colors and so on). Since most experiments happen

to be multiple rounds of the same thing, you usually only have to create the first set of rounds. Then, you can simply copy and paste the blocks for subsequent rounds.

After creating the sheet, you can test it yourself just within the spreadsheet. Then you can clear out the inputs and test it using the server and client.

Example 2: Outcomes depend on multiple players

As you can probably infer by now, this is simply linking cells within the spreadsheet. For example, consider a prisoner’s dilemma. The spreadsheet for round 1 is:

You choose U or D			They choose L or R			They choose U or D			You choose L or R		
If the choices are			You win	They win		If the choices are			You win	They win	
U,L			5	5		U,L			5	5	
U,R			10	2		U,R			10	2	
D,L			2	10		D,L			2	10	
D,R			1	1		D,R			1	1	
Enter your choice (U,D)						Enter your choice (L,R)					

We have two blocks, for player 1 and player 2. Each player only sees their block of information. They make their choices, and the blocks for round 2 are:

You choose U or D			They choose L or R			They choose U or D			You choose L or R		
If the choices are			You win	They win		If the choices are			You win	They win	
U,L			5	5		U,L			5	5	
U,R			10	2		U,R			10	2	
D,L			2	10		D,L			2	10	
D,R			1	1		D,R			1	1	
You chose						You chose					
They chose						They chose					
You won						You won					
They won						They won					

We have repeated the first round instructions, but this is not necessary, you can replace it with whatever you want. But you can see how easy it is. The formula for each red cell in round 2 is “=RedCell1” and that for the pink cell in round 2 is “=PinkCell1.”

The outcome is calculated from the payoff matrix.

The generalization to many players is simple: you just calculate the outcomes or payoffs depending on everyone’s decision(s).

Example 3: A Continuous Game

The examples so far have not had any feedback to participants within a round. For some games, like an English auction, you need to show the results of decisions during the round, for example the maximum bid so far. The FTS Experimental System allows you to specify rows that are constantly monitored and rows that are constantly transmitted. This is again best seen with an example, so here is an English auction with 5 buyers:

Name	Buyer 1	Buyer 2	Buyer 3	Buyer 4	Buyer 5
Value	17	19	3	79	5
Highest Bid Received	0	0	0	0	0
Enter your bid					
Outcome					
Payoff					

The formula in the blue cells is “=max(YellowCells)” and so as a bid is received, the maximum bid is transmitted to all. Both the blue cells and the yellow cells are monitored continuously in this example. The outcome obviously would be the whether you won or lost, could include another row with the winning bid, and the payoff could be calculated as a cell formula.