An Open-Source Application to Computerize Simple Market Efficiency Games

James T. Bang

1/28/2022

Abstract

In-class experiments provide instructors a powerful tool for helping students learn and understand market principles in economics. Despite the effectiveness of experiments, economics instructors remain slow to adopt them in their pedagogy. One reason for this lag could be the time-consuming process of collecting, tabulating, and presenting the outcomes of the experiments. This paper introduces functions and ShinyApps in R for fast, free, in-class tabulation of the results of five in-class market simulation experiments for teaching economics.

Keywords: Experiments; technology; teaching

Introduction

The use of experiments as demonstrations of economic theory date back at least as far as those conducted by Chamberlin (1948) and Smith (1962). These studies were designed primarily to collect evidence supporting or refuting economic models of rational behavior in market settings. Holt (1993) summarizes this literature. While experiments remain an important method for observing behavior to test economic hypotheses, these experiments have also found their way into pedagogy (DeYoung 1993).

Classroom experiments offer students and instructors a fun departure from the usual "chalk and talk" of explaining economic models. In addition to entertainment value, studies have shown experiments to increase student learning in post-test assessments (Emerson and Taylor 2004; Dickie 2006). I should note, however, that not all studies conclude that all types of gamification improves learning by a significant margin: Gremmen and Potters (1997) find a positive effect of games on average, but the effect is not statistically significant, while Dickie (2006) finds that games do significantly improve learning, but that attaching grade incentives to the games do not contribute any additional benefit. Moreover, Stodder (1998) expresses concern that classroom games that penalize cooperation may teach and reinforce unethical decision making.

Despite the potential learning and entertainment value of classroom experiments, they remain relatively rare among the pedagogies economics professors adopt in their classrooms (Watts and Becker 2008). Two factors may drive some of the hesitancy among economics instructors to implement classroom experiments. On the one hand, free resources, such as those described in the survey of non-computerized games by Brauer and Delemeester (2001), require significant time investments to tabulate and summarize the results. On the other hand, automated resources, especially those distributed by textbook publishers, impose a financial cost on students or their institutions that instructors feel rightly averse to asking budget-constrained students or departments to foot the bill for.

This contribution aims to overcome these barriers by automating some of the existing non-computerized experiments and introducing simplified versions of classic duopoly games with the hope of increasing the diversity of methods used by economics instructors in the classroom. These examples only require students to be able to access a Google Form via their browser on their computer or mobile device, which, given the ubiquity of mobile phones among students (sometimes as their only personal computing device), sets a fairly reasonable bar for accessibility.

On the instructor's end, I have created a free, downloadable package called `econGame` for the *R* open-source statistical computing program. The tabulation programs run as either stand-alone functions in the *R* console, or for demonstration purposes as a html *Shiny* app that can open in a browser tab if desired. This allows the instructor to present the results of the experiment almost instantaneously after the students have submitted their responses.

Description of the Experiments

The models I will describe in this paper encompass two well-established market equilibrium games known to the economics education literature, namely simplified versions of the pit market trading game introduced by Holt (1996) and the free entry and exit game introduced by Garratt (2000); and three games simulating different oligopoly models (Bertrand, Cournot, and Stackelberg). In the examples the "payoffs" students receive can be awarded to the students at the end of the games as "extra credit" points, or instructors may choose to encourage students to play the games strategically, but only "for the love of the game." I briefly describe the delivery of the games below.

Pit Market Trading

The instructor informs the students that they own a single unit of a eCoin currency that each of them values differently. Students then receive a random value from 1 to 10 using a link to a Google Sheet. This number represents their (constant) value they attach to the unit they presently own and for if they were to acquire one more unit of eCoin.[1]

Students submit their name, their value draw, a "bid" corresponding to the highest amount they would pay for a second eCoin, and an "ask" corresponding to the lowest amount they would accept to part with the eCoin they already own.[2]

Students keep their consumer and producer surpluses from each round as "extra credit" points. `equilibriumGame` tabulates the supply and demand schedules; calculates the equilibrium; graphs the equilibrium; and tabulates the scores for each student.[3]

Free Entry and Exit

The instructor informs the students that they will choose to plant corn, soybeans, or nothing. Producing corn incurs a cost of four points, while producing soybeans incurs a cost of 10 points. Selling a unit of corn brings revenue equal to

$$P_c = (N/2) + 6 - Q_c,$$

where N equals the number of students participating and Q_c equals the number of students choosing to produce corn. Selling a unit of soybeans brings revenue equal to

$$P_s = (N/2) + 10 - Q_s.$$

These parameters allow for there to be a "normal profit" of about one point per student in each market in equilibrium, to compensate for the risk of venturing into self-employment and lessen the chances that

[1] A template can be found at: https://docs.google.com/spreadsheets/d/1lCmC692ajsQZoatWtgZh5QKaJ9y3pOMt15JwRFaHanU/edit#gid=258904023.

[2] A template can be found at: https://docs.google.com/forms/d/1S_F9UJ6GXttxPqDLtk8Hg0ZgzDaHMxBmc1qH3W2gKZo/edit. It is important that when users create their own copies of the template that they use the same names in the question prompts. In case an instructor wants to add additional instructions or framing around the questions, they should add text fields.

[3] The solution the piecewise constant supply and demand equilibrium uses the help of a C++ helper function provided by "David" on Stack Overflow, https://stackoverflow.com/questions/23830906/intersection-of-two-step-functions.

students might "win" negative points. Students choosing to produce nothing sell their labor in the labor market and break even.[4]

Students may play as many rounds as the instructor decides to continue the game, or until the markets reach the long run equilibrium of zero *economic* profit. Usually the markets converge to the long run equilibrium (or students begin to overthink the excercise) by the end of the fourth or fifth round.

Oligoply Models

I also constructed a set of games to demonstrate and compare equilibria in different (two-firm) oligopoly models. In each of the examples, students work in pairs. The instructor informs the students that the market price depends on both the strategy they choose for their "firm" and also the strategy their partner chooses. Each of the three examples uses the following linear inverse demand function (the parameters of which individual instructors may change in the options):

$$P = a + b(Q_1 + Q_2),$$

where the default values for the parameters are $a = 10$ and $b = -1$. Likewise, firms face the the same cost function:

$$TC = f + cQ_i,$$

where f represents the fixed cost (0 by default) and c represents the (constant) marginal cost of each additional unit (6 by default).

Bertrand Duoploy

In the Bertrand game, students choose their price, and the firm with the lowest price wins the whole market demand. If both firms choose the same price, they split the market. Students submit their name and price strategy through a Google Form.[5]

Cournot Duopoly

In the Cournot game, students choose either to produce either a low or high quantity by choosing the strategy to "collude" (low quantity) or "defect" (high quantity). The function that tabulates the results assigns half of the monopolist's profit-maximizing quantity to students who choose "collude," and automatically assigns the quantity corresponding to the best response function for students who choose "defect" (which depends on the output choice of their rival). Students only need to make the simple binary choice.[6]

Instructors using this example for upper-level classes may (or may not) want to edit the game settings to require students to derive the best response functions and determine their quantity strategies themselves.

Stackelberg Duopoly

Similar to the Cournot game, students in the Stackelberg game choose to "collude" or "defect." In contrast to the Cournot game, the Stackelberg game reveals the leaders' strategy choice before the follower chooses their strategy. The function again automatically calculates the quantities corresponding to the set of binary strategy choices to determine the payoff outcomes.[^A template can be found at: https://docs.google.com/forms/d/1vERPMPt_kW96JPAY6mEtkQMu6FLCgPuqoFL8i8bulYk/edit.]

[4] A template can be found at: https://docs.google.com/forms/d/1oUsLulfD5bqT6_9VVYIzLWuuQ-L4vwmC4jI-1jabOVQ/edit.

[5] A template can be found at: https://docs.google.com/forms/d/1AykOoY6mVj17D_5CW7-BLhSgOJdGEhyfYHKHROnvdcg/edit.

[6] A template can be found at: https://docs.google.com/forms/d/1dp-tUv5rNhRpm9UjFCy_pgsD4rJJnja-QJnWnMu81DI/edit.

The econGame Package

The `econGame` package pulls activity data from Google the Google Sheet file generated for the form responses. It then tabulates, scores, and graphs the responses; and writes the grade data to a separate Google Sheet. Users can download the package from https://github.com/bangecon/econGame. To install the package, use the `remotes` package and the `install_github()` function:

```
remotes::install_github("bangecon/econGame")
```

Exporting the Results

Once the students have completed each round of a game, the instructor should pause to tabulate the results for that round. This is especially the case in the entry and exit game, where convergence to the long-run equilibrium requires students to know which sector earned economic profits in the previous round. To do this, all the instructor needs to do is click the Google Sheets icon in the responses tab of the edit page of the form. Figure 1 demonstrates where to find these tools in your Google Form.

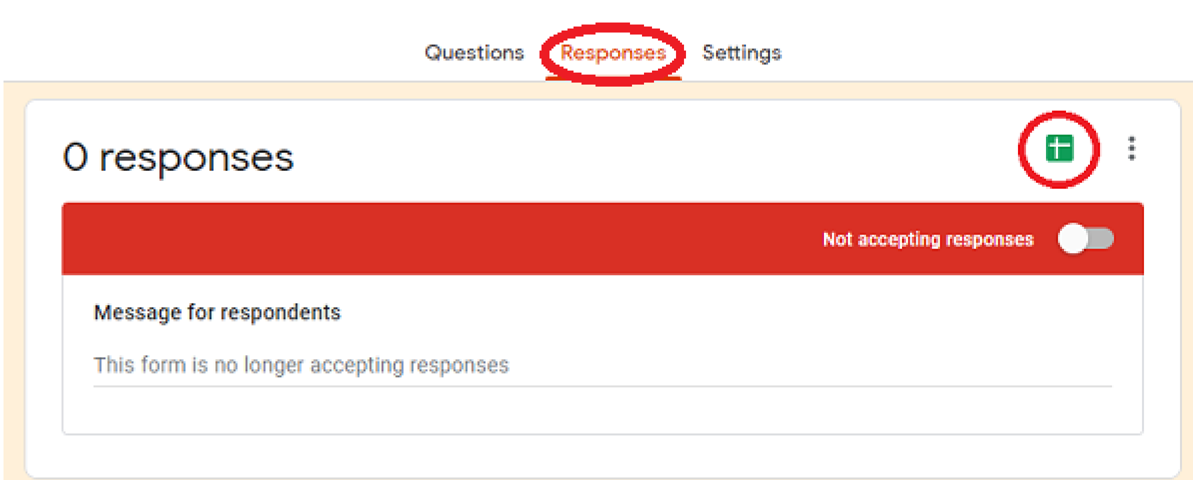

Figure 1: Exporting Results to Google Sheets

This creates a Google Sheet that contains all of the data for tabulating the results. In order to link to your results, you will need to copy to your clipboard the sheet ID from the URL. Figure 2 shows where you can find the sheet ID for your results.

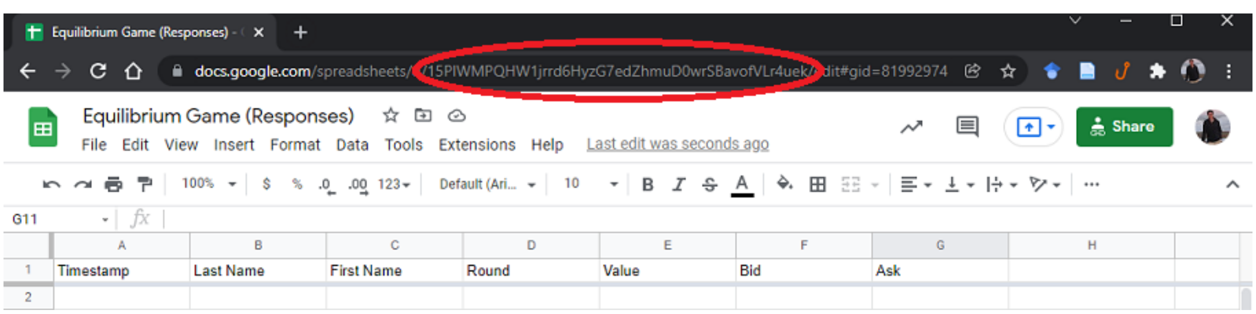

Figure 2: Finding the Sheet ID

Tabulating the Results

Each game has its own function to tabulate its results, and for each function there are slightly different options that the instructor can adjust based on any changes they might have made to the default parameters for the exercise. The only argument the instructor needs to provide (and that does not have a preset default) is the ID of the Google Sheet. The syntax and outputs for the different functions are:

- Equilibrium game: `equilibriumGame(sheet = NULL, ...)`
  - Arguments:
    - `sheet` (required) is a character string url corresponding to the Google Sheets location containing the individual submissions.
  - Outputs:
    - `type` returns the type of activity (equlibriumGame).
    - `results` returns the original submissions (with market price and points added).
    - `schedules` returns a list containing the supply and demand schedules for each round.
    - `equilibria` returns a list containing the equilibria for each round.
    - `grades` returns the aggregated points "won" by each student for the entire activity.
- Entry and exit game: `entryGame(sheet = NULL, ...)`
  - Arguments:
    - `sheet` (required) is a character string url corresponding to the Google Sheets location containing the individual submissions.
  - Outputs:
    - `type` returns the type of activity (equlibriumGame).
    - `results` returns the original submissions (with market price and points added).
    - `rounds` returns the number of rounds in "results."
    - `equilibria` returns a list containing the equilibria for each round.
    - `grades` returns the aggregated points "won" by each student for the entire activity.
- Bertrand game: `bertrandGame(sheet = NULL, a = 10, b = -1, c = 6, f = 0, ...)`
  - Arguments:
    - `sheet` (required) is a character string url corresponding to the Google Sheets location containing the individual submissions.
    - `a` is the value of the intercept of the linear inverse-demand function (default is 10).
    - `b` is the value of the slope of the linear inverse-demand function (default is -1).
    - `c` is the value of the firm's marginal cost (default is 6).
    - `f` is the value of the firm's fixed cost (default is 0).
  - Outputs:
    - `type` returns the type of activity (equlibriumGame).
    - `results` returns the original submissions (with market price and points added).
    - `grades` returns the aggregated points "won" by each student for the entire activity.
- Cournot game: `cournotGame(sheet = NULL, a = 10, b = -1, c = 6, f = 0, ...)`
  - Arguments:
    - `sheet` (required) is a character string url corresponding to the Google Sheets location containing the individual submissions.
    - `a` is the value of the intercept of the linear inverse-demand function (default is 10).

        * `b` is the value of the slope of the linear inverse-demand function (default is -1).
        * `c` is the value of the firm's marginal cost (default is 6).
        * `f` is the value of the firm's fixed cost (default is 0).
    - Outputs:
        * `type` returns the type of activity (equlibriumGame).
        * `results` returns the original submissions (with market price and points added).
        * `grades` returns the aggregated points "won" by each student for the entire activity.

- Stackelberg game: `stackelbergGame(sheet = NULL, a = 10, b = -1, c = 6, f = 0, ...)`
    - Arguments:
        * `sheet` (required) is a character string url corresponding to the Google Sheets location containing the individual submissions.
        * `a` is the value of the intercept of the linear inverse-demand function (default is 10).
        * `b` is the value of the slope of the linear inverse-demand function (default is -1).
        * `c` is the value of the firm's marginal cost (default is 6).
        * `f` is the value of the firm's fixed cost (default is 0).
    - Outputs:
        * `type` returns the type of activity (equlibriumGame).
        * `results` returns the original submissions (with market price and points added).
        * `grades` returns the aggregated points "won" by each student for the entire activity.

Another feature of the package is the ability to plot the results. The syntax to plot any of the games described in this paper is simply `plot(econGame, ...)`, where the (sole) argument is the name of an object assigned by one of the `econGame` functions. The plot the function generates depends on the type of game:

- For `type = 'equilibriumGame'`, plot the supply and demand functions with the corresponding equlibrium point.
- For `type = 'entryGame'`, plot the supply, demand, and per-unit cost lines to show profits and losses.
- For `type = 'bertrandGame'`, plot a histogram of the price strategies.
- For `type = 'cournotGame'` or `type = 'stackelbergGame'`, plot a bar graph of the strategy outcomes.

Shiny User Interface

The functions for directly summarizing the results of the games may be useful for tabulating the results for the purposes of awarding points to the students who participated, but may not be the most visually-appealing way to present the results in class. To improve the user interface for a "prettier" presentation of the results, I have built a Shiny Application UI for each of the games. In this interface, the instructor can display the raw results, plots, or schedules of the outcomes of the results by inputting the sheet ID (and other parameters) in the input boxes, and by switching the display tabs of the results.

To run the Shiny App for a given package, I have written a function that executes the app from the `'~/inst/shiny-examples'` folder of the package source. To execute the app, all the instructor needs to do is type one of the following commands: `'runEquilibriumGameApp()'`, `'runEntryGameApp()'`, `'runBertrandGameApp()'`, `'runCournotGameApp()'`, or `'runStackelbergGameApp()'`. Figure 3 shows the Shiny interface for `equilibriumGame`.

Discussion

This work provided examples of how to implement some simple market games using R and Shiny. The objective of the functions developed for this discussion was to help instructors adopt more creative and

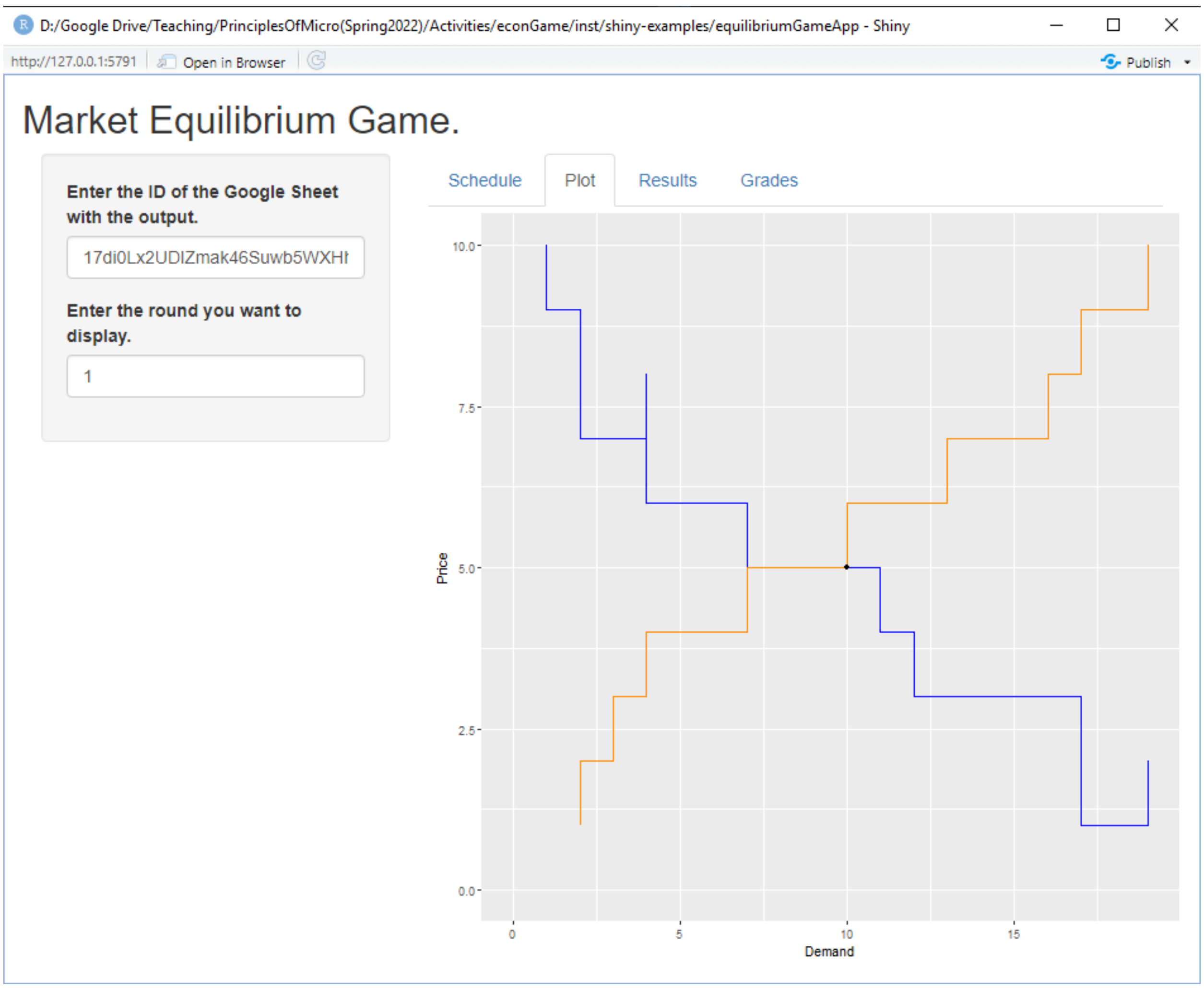

Figure 3: Shiny User Interface

engaging teaching methods in their principles classrooms by lowering both the financial and time costs of tabulating the results and presenting appealing summaries of the outcomes.

All of the examples can be implemented for the students using Google Forms, which collects the results in Google Sheets. Instructors may then tabulate the results using the `econGame` package via the HTML Shiny App with a single-function command in R. All of these applications and packages come at no cost to the student. The only potential cost barrier is the device - which could be as little as a mobile device - each student would need in order to input their responses.

Another useful feature is that I have platformed the package on GitHub, which allows users to make pull requests to suggest edits and changes to improve or add features and examples to the package. Plans are already in place to develop more examples from existing classroom experiments, and even design a few original games. If you use these resources, please let me know and share them with your colleagues.

References

Brauer, Jurgen, and Greg Delemeester. 2001. "Games Economists Play: A Survey of Non-Computerized Classroom-Games for College Economics." *Journal of Economic Surveys* 15 (2): 221–36.

Chamberlin, Edward H. 1948. "An Experimental Imperfect Market." *Journal of Political Economy* 56 (2): 95–108.

DeYoung, Robert. 1993. "Market Experiments: The Laboratory Versus the Classroom." *The Journal of Economic Education* 24 (4): 335–51.

Dickie, Mark. 2006. "Do Classroom Experiments Increase Learning in Introductory Microeconomics?" *The Journal of Economic Education* 37 (3): 267–88. https://doi.org/10.3200/JECE.37.3.267-288.

Emerson, Tisha LN, and Beck A. Taylor. 2004. "Comparing Student Achievement Across Experimental and Lecture-Oriented Sections of a Principles of Microeconomics Course." *Southern Economic Journal*, 672–93.

Garratt, Rod. 2000. "A Free Entry and Exit Experiment." *Journal of Economic Education* 31 (3): 237. https://www.proquest.com/docview/235237213/citation/8DF4FB2251DE443BPQ/1.

Gremmen, Hans, and Jan Potters. 1997. "Assessing the Efficacy of Gaming in Economic Education." *The Journal of Economic Education* 28 (4): 291–303.

Holt, Charles A. 1993. "Industrial Organization: A Survey of Laboratory Research."

———. 1996. "Classroom Games: Trading in a Pit Market." *Journal of Economic Perspectives* 10 (1): 193–203.

Smith, Vernon L. 1962. "An Experimental Study of Competitive Market Behavior." *Journal of Political Economy* 70 (2): 111–37.

Stodder, James. 1998. "Experimental Moralities: Ethics in Classroom Experiments." *The Journal of Economic Education* 29 (2): 127–38.

Watts, Michael, and William E. Becker. 2008. "A Little More Than Chalk and Talk: Results from a Third National Survey of Teaching Methods in Undergraduate Economics Courses." *The Journal of Economic Education* 39 (3): 273–86. https://doi.org/10.3200/JECE.39.3.273-286.