Problem Set 6 - ECON 5253

Thomas Mondry

March 21, 2022

1 Data preparation – J! Archive

I used this problem set in part to enhance and scale up the data cleaning functions for J! Archive data which I developed in PS5; my primary goal was to create functions that I can use in my final project for acquiring and cleaning the data. Getting the data from the J! Archive HTML pages to the final table which I will use for analysis consists of two steps:

1.1 Web scraping and data transformation – complete

The first part of the process is to convert the data from the HTML elements on the J! Archive site, which are easy to read manually but saved in a terrible format for analysis, into a concise, useful format. This part of the process is nonspecific to any particular research question; I'm just trying to represent all information that is stored in the J! Archive.

This is difficult because the layout of the HTML table containing the clues and the order in which they come up is highly inconsistent, as the tables contain about 40 unnecessary or meaningless cells for each cell containing pertinent information, on average. It is not immediately obvious where to look in the tables in order to find the relevant information; further, in many games, some clues do not get read because time runs out, and the inconsistent formatting of the HTML tables makes it very difficult to identify programmatically whether or not there are missing clues, and if so, which ones these are. However, I have created a set of rules which seem to hold in general. I developed a set of functions (saved in **/FinalProject/code/scrape.R**) which take a vector of game ID numbers (as have been seemingly arbitrarily selected by the J! Archive admins) and return a single data object, which is a list consisting of three elements:

1. **Clue-level information:** A list of length equal to the number of game IDs in the vector; each element is a data frame with 60 rows corresponding to the 30 Jeopardy round and 30 Double Jeopardy round clues, and each row contains information about the round, category, dollar value, and content of the clue; the order in which that clue came up in the game; whether or not that clue is a Daily Double and, if so, the amount of the wager; and each player's performance on that clue (correct, incorrect or no answer)

2. **Player-level information:** A list of length equal to the number of game IDs in the vector; each element is a data frame with 3 rows corresponding to the 3 players in the game, and each row contains information about the player's name, occupation, scores after the Jeopardy and Double Jeopardy rounds, Final Jeopardy wager, performance on Final Jeopardy (correct or incorrect), final score, and Coryat score (a metric which neither rewards nor punishes aggressive or conservative betting behavior, and which disregards Final Jeopardy)

3. **Game-level information:** A data frame of row length equal to the number of game IDs in the vector; each row contains some basic information about the game as well as the Final Jeopardy category and clue, which do not have a natural place elsewhere but may be useful

For this problem set, I ran the data-scraping process on 100 consecutive "regular" (i.e., not from a special tournament) games of *Jeopardy!* from the 2001-2002 season in order to generate some data for visualization. The data object is stored in **jData_100games.rda**, which is saved both in the directory for this problem set and in /FinalProject/data.

1.2 Problem-specific data cleaning – in progress

Once I select a research question, I will use my data object (with a larger sample of games for the final project) to create a tabular dataset of revelant features which I can then use to attempt to model the outcome. Since I don't yet have a problem in mind, I didn't create a full data cleaning process for this problem set; however, as shown in the **PS6_Mondry.R** script and described below, I did calculate a few metrics for visualizations. For these visualizations, I omitted the ??? of the 100 games in the sample which had more than 3 missing clues (out of 60).

2 Three visualizations

2.1 Relationship between

2.2 Graph of

2.3 Relationship between