12-2017

Application: Solving for a Local Companys Optimal Storage Strategy

Dakota Ferrin
Utah State University

APPLICATION: SOLVING FOR A LOCAL COMPANY'S OPTIMAL STORAGE STRATEGY

by

Dakota Ferrin

A Plan B submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Financial Economics

Approved:

____________________ Tyler J. Brough, Ph.D. Major Professor	____________________ Ryan Whitby, Ph.D. Committee Member
____________________ Ben Blau, Ph.D. Committee Member	____________________ Mark McLellan, Ph.D. Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2017

ABSTRACT

Application: Solving for a Local Company's Optimal Storage Strategy

by

Dakota Ferrin, Master of Science

Utah State University, 2017

Major Professor: Dr. Tyler J. Brough
Program: Financial Economics

This project used stationary historical spot and futures price data of two fairly closely correlated commodities and the stationary bootstrap method to simulate many possible spot and futures price paths that could have occurred over a given time frame. These simulated prices were then used to test various futures contract hedging strategies, finding values such as the mean terminal cumulative cashflow, mean terminal cumulative profit, and standard deviation of the terminal cumulative profits for each strategy. This paper explains how these results can be interpreted to help determine a local company's optimal storage strategy.

This paper specifically provides a way for the company Lower Foods, Incorporated to find this optimal storage solution for 3 different grades of the bottom round flat meat cut. When Lower Foods, Incorporated runs the given Python computer code files using their historical inventory level data and historical sales data, they can follow the example analysis in this paper to help decide how their company should utilize futures contracts to store meat.

(105 pages)

PUBLIC ABSTRACT

Application: Solving for a Local Company's Optimal Storage Strategy

Dakota Ferrin

This project used historical spot and futures price data of two fairly closely correlated commodities to simulate many possible spot and futures price paths that could have occurred over a given time frame. These simulated prices were then used to test various commodity storage strategies available through futures contracts. This paper explains how results, including values such as the mean terminal cumulative profit and the standard deviation of the mean terminal cumulative profits for each strategy, can be interpreted to help determine a local company's optimal storage strategy.

This paper specifically provides a way for a local company to find this optimal storage solution for 3 products they sell. When this company runs the given Python computer code files using their historical inventory level data and historical sales data, they can follow the example analysis in this paper to help decide how their company should utilize futures contracts to store each of the 3 commodities.

This project assumes that the company periodically adjusts the number of futures contracts they use based on the company's inventory level of a commodity. This would require the inventory level and the number of futures positions to be monitored by the company. This management of tracking inventory and adjusting the number of futures contracts being used could take both extra time and extra money to pay employees for that time, which presents two potential challenges for the implication of the results of this project. However, this project could serve as an important aid to this company, because it may help them obtain higher profits. This

helps not only the company's bottom line, but also the employees working for this company and the community in which the company is located.

DEDICATION

This Plan B project is dedicated to my husband, Clint Ferrin, who has helped me immensely throughout the whole process of working on this project. Thank you for helping me with many elements of the coding for this project, including helping write parts of it, debugging when I got stuck, showing me how to make graphs in Python, and more. Thank you for answering technical questions and providing your opinions and thoughts. Thank you for letting me explain to you in detail everything I’ve worked on. Thank you for the many sacrifices you made to give me extra time to work on this project. Thank you for constantly providing motivation, for believing in me, for loving me, and much more! I love you so much, Clint!

ACKNOWLEDGMENTS

This project would not have been possible without the help of many people and resources. I would like to specify a few that have been especially helpful.

I would like to thank Dr. Tyler J. Brough, Ph.D., the major professor on my committee. This project was his idea and he suggested it to me because he thought it would be a good fit for my future plans. I appreciate his thoughtfulness in thinking of me. He has also been extremely helpful through the entire process of completing this project as he explained to me the process of how it should be done, provided data, helped communicate with Lower Foods, Inc., answered countless questions through phone call, emails, and in-person meetings, and provided support and flexibility. I could not have done this project without his help.

Next, I'd like to thank Lower Foods, Inc., for their willingness to work with me on this project. They have provided data, answered questions, and even given of their time to meet with me in-person.

I also thank all the open source resources and help guides that helped me learn to code in R and Python. These sources helped me write and debug many coding questions that I otherwise would not have known how to solve. Dr. Jake VanderPlas created interactive notes that were an integral part of my learning Python. Dr. Tyler J. Brough added to these notes and used them in one of the classes I took from him. Thank you to both Dr. VanderPlas and Dr. Brough for this resource. I thank all the teachers who taught me knowledge I used to help me write this, especially Professor Devon Gorry, Ph.D. whose econometrics class and R example code helped me. Additionally, I thank open source resources that helped me understand formatting requirements for this paper, how to implement formatting requirements, and other relevant concepts. The website Purdue Owl was especially helpful with formatting.

My family members have all been so supportive and encouraging throughout this process. My mom, Linda Burt, deserves an especially big thank-you. I thank her for all the times she excused me from doing tasks such as cooking or doing the dishes just so I could have more time to work on this project. I thank her for the many rides she gave me to and from campus to work on the project or to meet with Professor Brough when I didn’t have a car available for use. I thank her for the time she has spent helping take care of my new baby while I worked on this project. I thank her for the continual motivation and reminders to finish as quickly as possible. She has always been one of my biggest cheerleaders!

Last, but certainly not least, I have two final thank-yous. Thank you to my wonderful husband, Clint Ferrin, and to our new baby girl who was born part way through this project. She has encouraged me to work quickly so I can spend time with her as a full-time mother. I love my sweet daughter so much and am so happy and blessed to be her mom! Along with a big thank-you and all that was expressed in the dedication of this project, I want to thank Clint for his patience, never-ending support, and unconditional love.

Dakota Ferrin

CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The main goal of this project is to identify the optimal hedge ratio that should be used to protect a company's inventory from falling spot prices. This optimal hedge ratio can help a company know how many futures contracts to sell, based on how much inventory it has at a given point in time, to maximize profits and optimize their strategy. Trading in this way would allow a company to make profit based on the relationship between the futures and spot prices, or the basis, which is more predictable than the alternative of making profit based on the rises and falls in spot prices.

Specifically, the company I worked with for this project is Lower Foods, Inc., a local company located in Richmond, Utah. Lower Foods is in the business of selling various cuts of meat. They own large capacity storage freezers and utilize them by buying large quantities of meat during seasons when it typically is sold at lower prices and storing the meat in their freezers so they have a large inventory of it that can be sold at higher prices during peak seasons (A. Lower & C. Lower, personal communication, May 8, 2017).[1] All this meat in storage, however, is a risk since the spot prices could fall before the meat is sold. Therefore, this Plan B project is designed to be a risk management mechanism that can help secure profits by basis trading rather than relying on an expectation of rising spot prices to earn profits on this inventory. This project uses Live Cattle electronic futures contracts as the means of hedging

---

[1] According to Lower Foods, Inc. representatives, their company can also get a loan to purchase storage; this loan has a monthly interest cost per pound of meat associated with it (personal communication, June 12, 2017). However, this project does not consider these storage options and, therefore, does not include information or calculations about the loan and the cost thereof.

specific grades of a specific cut of Lower Foods' meat inventory.[2] Currently, Lower Foods uses some forward contracts, but does not have a specific hedge ratio they are using in forward or futures contracts (A. Lower & C. Lower, personal communication, May 8, 2017).

Lower Foods identified the bottom round flat as a cut of meat they were particularly interested in finding an optimal hedge ratio for (C. Lower, personal communication, June 12, 2017). Per Lower Foods management's request, this project focuses on three grades of the bottom round flat meat cut: Select, Choice, and CAB (A. Lower, C. Lower, Morris, & Mortensen, personal communication, June 12, 2017). CAB stands for Certified Angus Beef (Goldwyn, 2010). Throughout this paper, when I refer to the meat grades collectively in any way or refer to Lower Foods' inventory, I am always referring to these three meat grades and always referring to those grades with respect to the bottom round flat meat cut.

This project largely follows the ideas and procedures outlined by Bollen and Whaley (1998) in their paper "Simulating Supply" and by Alizadeh and Nomikos (2008) in their paper "Performance of Statistical Arbitrage in Petroleum Futures Markets."

Bollen and Whaley (1998) performed an analysis similar to this project on the company Metallgesellschaft to help determine whether the company was justified in their chosen hedge ratio of 1.0 or if they should have used an alternative hedge ratio, such as a 5% tailed hedge. They concluded that, if Metallgesellschaft would have held their futures contracts until maturity rather than closing their positions early, they would have made a significant profit with their one-for-one hedging strategy. They showed that Metallgesellschaft's increased downside

[2] The futures data is cited in a footnote in the "Futures Price Data" subsection in the "Data" section of this paper.

monetary risk was minimal compared to the increased upside monetary possibility available to the company when the 1.0 hedge ratio was used rather than the tailed hedge.

In their paper “Performance of Statistical Arbitrage in Petroleum Futures Markets,” Alizadeh and Nomikos (2008) used Politis and Romano’s simulation methodology called the stationary bootstrap method (as cited in Alizadeh & Nomikos, 2008). This is a key feature I also use in this project. This is a method of simulating many possible price paths (Alizadeh & Nomikos, 2008), given a set of stationary prices (Brough, personal communication, June 13, 2017).

Lower Foods has provided invaluable data for this project thus far; however, due to busy meat seasons, they were unable to provide all the necessary data to determine the actual optimal hedge ratio for each grade of the bottom round flat meat cut for their company. To supplement the missing data, I make assumptions regarding the sales and inventory levels. Since these assumed sales and inventory data do not represent real values for Lower Foods, however, this paper does not report the actual optimal hedge ratios for each grade. The Python computer code files written for this paper will be provided to Lower Foods and instructions on how to format data and analyze results from these Python code files can be found in Appendix A. Figure 1 in Appendix B shows the file structure of the folders that will be provided to Lower Foods; this file structure explanation will help Lower Foods navigate the folders referred to in the instructions document in Appendix A. With these materials, Lower Foods can gather the necessary sales and inventory data, format it to be compatible with the computer code files I have written, run it through the code files so it can be analyzed, and find the optimal hedge ratio for each grade of the bottom round flat meat cut at a time that is a less busy season for their business. In essence, I have created a tool for them to use to determine the optimal hedge ratio for these three grades of the bottom round flat meat cut given the assumptions outlined in this paper.

For this project, I use the R version 3.3.1 programming language with the RStudio software (RStudio Team, 2016). I also use the Python version 3.6.0 programming language with the Spyder software (Raybaut & The Spyder Project Contributors, 2017) and Anaconda3 software (Anaconda3, 2016).

Following this introductory section, there are three major sections of this paper including the “Data” section, the “Procedures and Results” section, and the “Conclusion” section. There are three major parts of this project and each will be discussed in a separate subsection within the “Procedures and Results” section. The first subsection involves statistical analysis used to ensure the price data is stationary; the second subsection includes the stationary bootstrap method used to simulate many possible spot and futures price paths; and the third subsection consists of a hedging strategy analysis, which can be used as an example analysis when Lower Foods determines which hedge ratio would maximize their profits and optimize their strategy.

DATA

Several sets of data are used (or will be used when instructions in Appendix A are followed) in this project including Lower Foods' historical purchases and historical sales data for each of the three grades of meat discussed in the "Introduction" section including Select, Choice, and CAB; Lower Foods' historical inventory level data for each grade; spot price data for each grade provided by Lower Foods who retrieved it from their subscription to Comtell, which is part of Urner Barry (*UB 171B 3 Rnd*, 2017; *USDA 171B 3 Rnd, Out. Rnd CH*, 2017; *USDA 171B 3 Rnd, Out. Rnd SE*, 2017);[3] and front-month futures price data.[4] Each dataset shows data at least for the three-year and five-month time period beginning on January 1, 2014, and ending on May 31, 2017. This is the time period I focus on in this project. Unless otherwise specified, all currency values throughout this paper and throughout all appendices are in terms of United States dollars.[5]

Various datasets discussed in this section have missing data values or missing information for different dates. As specified where applicable, various calculations in this paper exclude information from certain dates for various reasons such as to omit missing values from any given individual dataset or to ensure that a given date has both a spot and a futures price associated with it. However, all tables and graphs in this paper are still presented as covering

[3] Urner Barry's Comtell is a resource Lower Foods has a subscription for. The Urner Barry Comtell webpage referenced through the following in-text citation gives additional information about Comtell ("Make Smarter," n.d.).
[4] Again, the futures data is cited in a footnote in the "Futures Price Data" subsection in this section of the paper.
[5] Lower Foods may have mentioned having business with Canada (Lower Foods, Inc. representatives, personal communication, June 12, 2017). However, I assume that all price values listed in the datasets used in this paper, including Lower Foods' historical datasets, are in terms of United States dollars.

the entire time period from January 1, 2014 through May 31, 2017, even if several dates and the information associated with those dates in that time period are not included in the tables and graphs or the calculations used to make them. To know what dates' information was excluded in each table and graph in the appendices or, in other words, to know how the data presented in each table and graph are calculated, refer to various portions of the main body of the paper.[6]

Purchases and Sales Data

The daily purchases data was provided by Lower Foods as it is their historical data. This data includes a record from January 1, 2014, to May 31, 2017, of the received incoming bottom round flat meat cut for each of the three grades discussed. For the time period specified above, I consolidated the provided record for each grade to provide information on dates that purchased meat was received, the number of net pounds of meat that were purchased and added into Lower Foods' storage freezers on those dates, and at what weighted average purchase price per pound the meat was purchased on those dates. For simplicity, I assume that the meat receipt date shown in the purchases dataset is also the date on which the purchase was made and the associated monetary transaction was settled. Also, although the datasets do not explicitly state that the prices are in units of dollars, I assume this to be the case.

One of the data pieces still needed to be gathered and compiled with specific formatting includes their daily historical sales data; this data will be provided by Lower Foods when the instructions in Appendix A are followed. This data will be provided by Lower Foods because it is

[6] This information may not be directly next to where the table or figure is first mentioned; for example, it may need to be found by looking further up in the text to where the paper first describes how data is calculated before being tested or analyzed into more data that is analyzed into results, which are then inputted into the table or figure.

their historical data. The consolidated sales data will include the daily number of net pounds of meat sold for each grade from January 1, 2014, to May 31, 2017, and the weighted average sales price per pound Lower Foods sold this meat at on each date over the same time period. The prices in the sales dataset will be in terms of dollars.

When creating the purchases dataset, when multiple purchase transactions occurred on the same day, the transactions are consolidated so the daily purchases dataset uses net pounds and weighted average prices. Also, if date d has a purchase transaction but date d does not have both a futures price and a spot price listed in the futures and spot price datasets, respectively, date d's purchases data are combined with the purchases data for the next date after date d that has both a spot and futures price listed in the spot and futures datasets, respectively, which I will refer to as date e.[7] Therefore, information from date d in the purchases dataset is incorporated into the net pounds and weighted average price per pound measures for date e. The information and specifications in this paragraph about how the purchases data is compiled will also apply to the sales dataset when instructions in Appendix A are followed. The futures and spot prices in the futures and spot price datasets, respectively, referred to in this paragraph are discussed in later subsections of this paper.

The purchases data is used in this project and the sales data will be used in this project to help create the inventory level dataset, which will be discussed in the next subsection. Also, the purchases and sales datasets are or will be, respectively, used in this project to help track

[7]Reasons for missing data in the spot or futures datasets on certain dates include reasons such as that weekends or certain holidays did not include data values, that data values were not provided on certain dates in different datasets, or that one of the datasets is weekly data and, therefore, does not have daily data values (*UB 171B 3 Rnd*, 2017).

the total dollar value of purchases and sales, respectively, that Lower Foods was a part of in the spot market, which factors into the spot market cashflows and profits.

Inventory Level Data

The daily inventory level data is still needed to use the tool produced in this project to help determine Lower Foods' optimal hedge ratio. Lower Foods will provide this historical data when the instructions in Appendix A are followed. The inventory level data will include a record from January 1, 2014, to May 31, 2017, of the daily number of pounds of each grade of the bottom round flat meat cut in Lower Foods' storage freezers.

This paper gives details about solving for and Appendix A provides instructions on how to solve for the inventory level data in case it is not available. To solve for this data, the user of the instructions in Appendix A will need one final piece of data from Lower Foods, which will allow the user to use the code files provided to help determine Lower Foods' optimal hedge ratios for each grade. Specifically, the data piece still needed includes the number of pounds of each of the three meat grades in Lower Foods' storage freezers on either January 1, 2014, or May 31, 2017, before that day's sales of inventory or purchases of new inventory have been incorporated into the measurement. The combination of the purchases data already gathered, the sales data, and the initial or ending inventory amount will allow the daily inventory level (measured in pounds) in Lower Foods' storage freezers to be solved for. Since this method simply uses purchases and sales to help track the inventory level of a given meat cut and grade in the storage freezers and apply the hedge ratio to this value, it implicitly assumes that all Lower Foods' inventory is stored in their storage freezers and that this entire inventory value is used when applying the hedge ratio.

The daily inventory level data will be used in this project when applying hedge ratios to enter the number of futures contracts to the nearest whole contract representing the same number of pounds as a given ratio of the total inventory.

Spot Price Data

The historical spot data was obtained from Lower Foods who obtained it from their subscription to Comtell, which is part of Urner Barry (*UB 171B 3 Rnd*, 2017; *USDA 171B 3 Rnd, Out. Rnd CH*, 2017; *USDA 171B 3 Rnd, Out. Rnd SE*, 2017). According to Comtell, the exact names of the Select, Choice, and CAB grades of the bottom round flat meat cut are USDA 171B 3 Round, Outside Round Select; USDA 171B 3 Round, Outside Round Choice; and UB 171B 3 Round, Outside Round CAB, respectively ("UB 171B 3 Round," 2017; "USDA 171B 3 Round, Outside Round Choice," 2017; "USDA 171B 3 Round, Outside Round Select," 2017). The spot prices of the Select and Choice grades are daily prices and are measured in weighted average dollars per pound; the spot prices of the CAB grade, however, are weekly prices and are measured in simple average dollars per pound (*UB 171B 3 Rnd*, 2017; *USDA 171B 3 Rnd, Out. Rnd CH*, 2017; *USDA 171B 3 Rnd, Out. Rnd SE*, 2017). These spot prices are used in this project to determine the value of and changes in value of Lower Foods' inventory, which factors into the spot market profits.

Futures Price Data

All daily historical front-month futures prices were obtained from barchart.com.[8] Several sets of front-month futures contracts and their associated prices are considered as

---

[8] These front-month futures data were all obtained on June 19, 2017, and are available from barchart OnDemand at https://www.barchart.com/ondemand/. These csv data file sources were cited in a footnote rather than in the reference

potential hedging tools for Lower Foods' inventory of the three grades discussed of the bottom round flat meat cut, including Live Cattle electronic, Live Cattle pit, Feeder Cattle electronic, and Feeder Cattle pit, or product codes LE, LC, GF, and FC, respectively.[9] Chicago Mercantile Exchange, Inc., or CME ("CME Definitions," n.d.), is the exchange these contracts are traded on ("Feeder Cattle Jan," 2017; "Feeder Cattle Pit," 2017; "Live Cattle Feb," 2017; "Live Cattle Pit," 2017). Each of the futures price datasets includes information regarding the name of the front-month contract on each date listed.

I first determine that the electronic prices are preferred to the pit prices because nearly 60% of the pit prices from January 2, 1980, to June 19, 2017, for both Live and Feeder Cattle contracts are duplicates of the electronic prices for each respective commodity contract. However, greater than 98% of the duplicates for both Live Cattle pit and Feeder Cattle pit contracts are pre-March 1, 2002. This could be because electronic futures contracts on Live and Feeder Cattle may not have been traded pre-March 1, 2002, so pit prices might be used as estimates for what the electronic prices would have been before this date (Brough, personal communication, June 19, 2017). Since the focus of this paper is looking at data starting on January 1, 2014, and electronic and pit contract prices are tracked separately starting on March 1, 2002, I conclude that the electronic contract prices are a better measure for both Live and

---

list because the data did not have authors or titles listed. The full URLs are not provided because they may need a subscription to barchart OnDemand to be accessed. Throughout this paper, when futures data is referred to, it is always referring to one of these datasets; therefore, the citations in this footnote apply to all additional information presented about the futures data and/or futures prices.

[9] The webpage from which I learned the product codes is available from barchart OnDemand at https://www.barchart.com/ondemand/. The full URL is not provided because it may need a subscription to barchart OnDemand to be accessed.

Feeder Cattle contracts than pit contract prices because the futures industry seems to be moving more toward electronic trading (“Trading Venues,” n.d.).

This leaves only the Live Cattle electronic and Feeder Cattle electronic futures contracts to choose between as the hedging contract for this project. Lower Foods is selling various grades of the bottom round flat meat cut and a futures contract for these exact commodities does not exist, so they will need to cross hedge with a different futures contract (Hull, 2015a). To determine whether they should use the Live Cattle electronic or Feeder Cattle electronic contract to hedge with, I choose the one whose prices are more highly correlated with the prices of each grade of the bottom round flat meat cut. The more correlated the futures contract used as a cross hedge for the underlying commodity is, the closer the futures contract is to mimicking the price movements of the underlying commodity, which provides the closest possible result to what would occur if the company could hedge with a futures contract on the exact underlying commodity.

Using the closing futures prices, I convert the prices to dollars per pound since the original futures data for both Feeder Cattle electronic and Live Cattle electronic futures prices are measured in cents per pound (“Feeder Cattle Futures,” n.d.; “Live Cattle Futures,” n.d.). This way, the futures prices can be compared to the spot prices since they now have identical units. I calculate the Pearson correlation coefficient between both Live Cattle electronic and Feeder Cattle electronic futures contract prices and each grade of spot prices including information only from dates that have both a spot and futures price listed in the relevant spot and relevant futures datasets, respectively. For example, if a given date has a spot price listed for the Select grade but does not list the Live Cattle electronic futures price for that date, all information for that date is excluded from that Pearson correlation coefficient calculation. Table 1 in Appendix C shows that, although the correlation coefficients between each grade of spot prices and Live Cattle electronic futures prices and the correlation coefficients between each grade of spot prices and Feeder Cattle electronic futures prices are very close, the correlation coefficient between the spot prices and the Live Cattle electronic futures prices is slightly higher for every grade. Therefore, the Live Cattle electronic futures contract prices are more correlated with the spot prices of each grade, so this is the futures price series chosen to use in this project to hedge Lower Foods' inventory and help track the futures market cashflows and profits. Hereafter, the Live Cattle electronic futures contract may simply be referred to as the Live Cattle futures contract, the Live Cattle contract, the front-month futures, the futures contract, or the futures and any reference to the futures prices are referring to the Live Cattle electronic futures contract prices.

PROCEDURES AND RESULTS

This section walks through the process I used to analyze the data for this project. It then presents and interprets results from example data. This section is separated into three main subsections that detail various stages of this project

Statistical Analysis

This subsection involves statistical analysis used to ensure the price data is stationary, which is a prerequisite to using the stationary bootstrap method discussed in the next main subsection (Brough, personal communication, June 13, 2017). This subsection also discusses cointegration between a version of the futures price and a version of the spot prices for each grade.

I used the Wooldridge (2009) textbook *Introductory Econometrics: A Modern Approach* to learn much of the skills and material presented in this "Statistical Analysis" subsection. This textbook taught me about t-tests; an augmented Dickey-Fuller (ADF) unit root test and its associated equation, drift intercept variable, null hypothesis, and results' interpretation; the definition of cointegration, how to test for cointegration using an Engle-Granger test, and how to interpret the results of the Engle-Granger test; and other econometric principles and methods. I used RDocumentation to help understand the augmented Dickey-Fuller unit root tests and how to write the code associated with it (Pfaff, 2016); I also used it to help understand other functions I used in my code and how to write code using those functions. I use an adapted version of code written by Brough (2017b) to test for unit roots in various forms of a price series using a function that performs an ADF unit root test and to learn to interpret this test's results.

My adapted version of Brough's (2017b) code also is used to perform Engle-Granger cointegration tests and to learn how to interpret these tests.

Augmented Dickey-Fuller unit root tests. The level, level first-differenced, log, and log first-differenced price series of futures prices and of each grade of spot prices are calculated using a version of each of the datasets with missing values of each individual dataset omitted. I calculate the level price at time *t*, level first-differenced price at time *t*, log price at time *t*, and log first-differenced price at time *t* using equations (1), (2), (3), and (4), respectively.

$$\text{Level price at time } t = p_{\mathrm{t}} \tag{1}$$

$$\text{Level first-differenced price at time } t = p_{\mathrm{t}} - p_{\mathrm{t-1}} \tag{2}$$

$$\text{Log price at time } t = \ln(p_{\mathrm{t}}) \tag{3}$$

$$\text{Log first-differenced price at time } t = \ln(p_{\mathrm{t}}) - \ln(p_{\mathrm{t-1}}) \tag{4}$$

where price is measured in dollars per pound, p_{t} is the price at time *t*, and $p_{\mathrm{t-1}}$ is the price at time $t - 1$.

Since I later want to use the stationary bootstrap method to simulate futures prices and each grade of the spot prices, I first need to show that each of these price series, or some form of the price series such as the log first-differenced price series, is stationary and contains no unit roots.

To do this, I perform augmented Dickey-Fuller (ADF) unit root tests on the four forms of each price series, calculated as described above, to test if unit roots are present therein. A drift term is included in the ADF tests, meaning an intercept is added into the regression equations being estimated. Ten lags are potentially included in the ADF tests, and Bayes Information Criteria (BIC) is used to determine which of the ten lags are actually included. The drift term and

BIC specifications were chosen by following the examples in Brough's (2017b) code.[10] All these specifications result in each estimated ADF regression equation being

$$\Delta y_{\mathrm{t}} = \alpha + \theta y_{\mathrm{t}-1} + \gamma_1 \Delta y_{\mathrm{t}-1} + \gamma_2 \Delta y_{\mathrm{t}-2} + \cdots + \gamma_{10} \Delta y_{\mathrm{t}-10} + e_{\mathrm{t}} \quad (5)$$

where $y_{\mathrm{t}-1}$ is the time $t-1$ data value in the price series being tested; $\Delta y_{\mathrm{t}-\mathrm{x}}$ is the change in the value of y from time $t-x-1$ to time $t-x$ where x is the lag number; α is the intercept representing the drift term; θ is the coefficient being estimated for the variable $y_{\mathrm{t}-1}$; γ_{x} is the coefficient being estimated for the variable $\Delta y_{\mathrm{t}-\mathrm{x}}$ where x is, again, the lag number; e_{t} is the error term where $E(e_t|\mathrm{y}_{\mathrm{t}-1}, \Delta y_{\mathrm{t}-1}, \Delta y_{\mathrm{t}-2}, \cdots, \Delta y_{\mathrm{t}-10}) = 0$; and the number of lags included, up to a maximum of the ten lags specified, is determined by BIC. The null hypothesis is that $\theta = 0$ or, in other words, that there are unit roots in the price series being tested.

The results of the ADF test for each form of each of the four price series are shown in Table 2 in Appendix D. The values shown in the table are the t-statistics on the coefficient θ, and the values in parenthesis represent the number of lags included in each ADF test. Using the tau2 critical values given in the R summary for each ADF test (Pfaff, 2016), I find that each price series in the level and log forms is either not significant at least at the 10% significance level or is significant only at the 5% or 10% significance level. However, each price series in the level first-differenced and log first-differenced forms is easily significant at a 1% significance level. Therefore, for the level and log forms of each price series, I fail to reject the null hypothesis that there are unit roots in each of those forms of the price series at the 1% significance level; however, for the level first-differenced and log first-differenced forms of each price series, I

[10] The drift term and BIC specifications are chosen by following examples in Brough's (2017changetob) code not only in this instance, but for every ADF and Engle-Granger test used in this paper including in the appendices. However, this citation will not be repeated each time these specifications are mentioned.

easily reject the null hypothesis that there are unit roots in each of those forms of the price series at the 1% significance level. Since I conclude there may be no unit roots in the level first-differenced and log first-differenced price series, I later use one of those forms of each price series in the stationary bootstrap simulations. I use the log first-differenced form of each price series.[11]

Cointegration tests. I define the basis in four forms using only dates from the time period that have both a spot and a futures price listed in the spot and futures datasets, respectively. I calculate the level basis at time t, level first-differenced basis at time t, log basis at time t, and log first-differenced basis at time t using equations (6), (7), (8), and (9), respectively.

$$\text{Level basis at time } t = b_{\mathrm{t}} = s_{\mathrm{t}} - f_{\mathrm{t}} \tag{6}$$

$$\text{Level first-differenced basis at time } t = b_{\mathrm{t}} - b_{\mathrm{t-1}} \tag{7}$$

$$\text{Log basis at time } t = \ln(s_{\mathrm{t}}) - \ln(f_{\mathrm{t}}) \tag{8}$$

$$\text{Log first-differenced basis at time } t = (\text{Log basis at time } t) - (\text{Log basis at time } t-1) = [\ln(s_{\mathrm{t}}) - \ln(f_{\mathrm{t}})] - [\ln(s_{\mathrm{t-1}}) - \ln(f_{\mathrm{t-1}})] \tag{9}$$

where basis is measured in dollars per pound; b_{t} is the level basis at time t; s_{t} is the spot price at time t; f_{t} is the futures price at time t; and $b_{\mathrm{t-1}}$, $s_{\mathrm{t-1}}$, and $f_{\mathrm{t-1}}$ are the time $t-1$ basis, spot price, and futures price, respectively.[12]

---

[11] I actually use the log first-differenced form of each price series, but where the price series is calculated using a different method than implied here, as described later in the paper.

[12] The level basis equation is sometimes written the opposite direction where the basis at time t equals the futures price at time t minus the spot price at time t. I choose to use the version of the equation presented in equation six, however, because, for the data used in this project, it allows the basis to be a positive number; because McDonald (2006, p. 907) doesn't specify which version of the basis it uses in that location of the textbook; and because Hull (2015a, p. 55) finds the basis using this calculation.

I next wish to show that the futures prices are cointegrated with each grade of spot prices. According to Wooldridge (2009, pp. 637-638), cointegration means that when two price series containing unit roots are combined using the equation $y - \beta x$ where y and x are the unit root price series, β is the cointegration parameter, and $\beta \neq 0$, $y - \beta x$ is a dataset series containing no unit roots. I want to show that a form of the futures price series containing unit roots is cointegrated with a form of each spot price series containing unit roots, meaning that, if the cointegrating parameter equals 1 ($\beta = 1$), one of the forms of the basis defined above may be stationary, containing no unit roots.

Unless otherwise specified, in all references to level, level first-differenced, log, or log first-differenced forms of price series in the remainder of the “Procedures and Results” section, in the “Conclusion” section, and in Appendix A, the text is referring to the form of the prices calculated under a new method of including only dates in the time period that have both a spot and futures price listed in the relevant spot and futures datasets, respectively. For example, this means that the log form of the futures price series and the log form of each spot price series are not, as they are previously in this “Statistical Analysis” subsection, calculated from information associated with dates in the time period where dates in each individual dataset with missing prices are omitted; rather, they are calculated only from information associated with dates in the time period that have both a spot and futures price listed in the relevant spot and futures datasets, respectively. As previously discussed, to know what dates’ information was included or excluded in each table and graph in the Appendices, refer to various portions of the main body of the paper.

I use the log futures and each log spot price series (calculated with the new method) since I previously fail to reject that they (calculated with the previous method) contain unit roots at the 1% significance level. Since the “Augmented Dickey-Fuller unit root tests” level-two subsection concludes that the log price series under the previous calculation method contain unit roots, I assume that the log price series under the new calculation method also contain unit roots and can, therefore, be used as the unit root price series in the cointegration tests.

By following a procedure outlined by Wooldridge (2009, p. 639), I test for cointegration by regressing each log spot price series on the log futures price series, finding the residuals from each of the three regressions, running an ADF test on each of the residuals data series, and using the asymptotic critical values for data without a time trend from Table 20.2 in Davidson and MacKinnon’s work (as cited in Wooldridge, 2009, p. 639) to test for unit roots in the residuals.[13] This process is known as the Engle-Granger test since alternative critical values, that is, alternative to Dickey-Fuller critical values, are used to interpret the ADF tests (Wooldridge, 2009, p. 639).

The Engle-Granger tests involving the ADF tests on each of the previously discussed residuals series are run using identical specifications as used in the ADF tests discussed in a previous part of this paper, including adding a drift term, using potentially ten lags, and using

[13] Wooldridge (2009, p. 640) specified that if the data includes a drift term, which I assume it does in this project, a time trend should be added into the regression when testing for cointegration; he also specified that, if this is the case, the asymptotic critical values for data with a time trend from Table 20.2 of Davidson and MacKinnon’s work (as cited in Wooldridge, 2009, p. 640) should be used to test for unit roots in the residuals. For simplicity, however, I did not include a time trend term in the regression when testing for cointegration; I also used the asymptotic critical values for data without a time trend from Table 20.2 of Davidson and MacKinnon’s work (as cited in Wooldridge, 2009, p. 639) to test for unit roots in the residuals. The cointegration results should be similar, however, with either method (Brough, personal communication, December 8, 2017). Even if the results were different, however, I could still move forward with the stationary bootstrap method in the next subsection since each log first-differenced spot and the log first-differenced futures price series were concluded earlier in this subsection to be stationary.

BIC to determine how many of the ten allowed lags to include in the regression. The null hypothesis is, again, that $\theta = 0$ or, in other words, that there are unit roots in the price series being tested. The equation being estimated in each ADF test is identical to equation (5), except that y_{t-1} now represents the time $t - 1$ residual from the log spot price series being regressed on the log futures price series, and Δy_{t-x} is the change in the value of this new definition of y from time $t - x - 1$ to time $t - x$ where x is the lag number. The ADF t-statistics for the coefficient θ in these Engle-Granger tests are shown in Table 3 in Appendix E. The values in parentheses in Table 3 represent the number of lags included in each ADF test. Using the asymptotic critical values for data without a time trend from Table 20.2 in Davidson and MacKinnon's work (as cited in Wooldridge, 2009, p. 639), I find that the t-statistic on θ for each series of residuals is significant at the 1% significance level.[14] Therefore, for all three series of residuals previously discussed, I reject the null hypothesis that there are unit roots present in the residual series at the 1% significance level. I, therefore, conclude that the log spot prices are cointegrated with the log futures prices.

Using a two-tailed t-statistic test, I find that the cointegrating parameter for each of the three regressions is not statistically different from 1 at the 10%, 5%, and 1% significance levels (Wooldridge, 2009, p. 825). Therefore, I fail to reject the null hypothesis that $\beta = 1$ in each case,

[14] Although, when it comes to the regression and critical values, I test for cointegration as if there were no drift term, as explained in a previous footnote, I still include a drift term in the ADF test equation on each residual series. Table 3 presents the t-statistics on the variable θ for this version of each ADF equation. However, to be consistent throughout the entire cointegration test on the no-drift assumption, I also run the ADF test on each residual series where each ADF test equation is identical except that it does not include the drift term; holding all else constant in the way the cointegration test is performed and the critical values used in analysis, the t-statistics on the variable θ in each ADF equation in the cointegration tests are still significant at the 1% significance level even when using this alternative specification in the Engle-Granger tests.

meaning that I fail to reject that the cointegrating parameter between each log spot and log futures price series yields log basis since $y - \beta x$ is equivalent to $y - x$, or the log basis, when $\beta = 1$ is substituted into the equation.

Since the log spot and log futures prices are cointegrated and I fail to reject that $\beta = 1$ in $y - \beta x$, I fail to reject that $y - \beta x$ yields the log basis series, and I conclude that the log basis series does not contain unit roots. This seems to be supported by Figures 2, 3, and 4, which are created using an adapted version of Brough's (2017b) code and are adapted from the figure produced by his code, in Appendices F, G, and H, respectively. These three figures plot the residuals from each regression of log spot prices on log futures prices; they graph data values for the dates in the time frame that have both a spot and futures price listed in the spot and futures datasets, respectively, for a given basis series. Each of these figures has a mean-reverting pattern that appears to support the result that the residuals are stationary.

The results from the ADF unit root tests and the cointegration tests are supported by Figures 5, 6, 7, 8, 9, 10, 11, and 12 in Appendices I, J, K, L, M, N, O, and P, respectively.[15] Figures 5, 6, 7, and 8 show patterns or a lack of patterns in various forms of the futures prices series and each spot price series; these figures graph data values for the dates in the time frame that did not have missing price values in a given spot or futures price series. Figures 9, 10, 11, and 12 show patterns or a lack of patterns in various forms of each of the basis series; these figures graph data values for the dates in the time frame that have both a spot and futures price listed in the spot and futures datasets, respectively, for a given basis series.

[15] These figures are created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; these figures are adapted from figures produced by these code files.

Figures 5, 7, 9, and 11 show the random walk appearance of the level futures and each level spot price series, the log futures and each log spot price series, each level basis price series, and each log basis price series, respectively. This suggests the presence of unit roots in each of these price series.

In contrast, Figures 6, 8, 10, and 12 show the appearance of mean reversion around the values \$0/lb in levels for the level first-differenced futures and each level first-differenced spot price series, \$0/lb in logs for the log first-differenced futures and each log first-differenced spot price series, \$0/lb in levels for each level first-differenced basis price series, and \$0/lb in logs for each log first-differenced basis price series, respectively. This suggests the stationarity of each of these price series, meaning no unit roots are present therein.

Stationary Bootstrap Method

In the previous subsection, I determine that the log first-differenced form of the futures price series and the log first-differenced form of each spot price series (calculated using the previous method) do not contain unit roots; therefore, these price series (calculated using the new method) can be used as datasets to simulate from in the stationary bootstrap method. Since the “Statistical Analysis” subsection concludes that the log first-differenced price series under the previous calculation method is unit root stationary, I assume that the log first-differenced price series under the new calculation method is also unit root stationary and can, therefore, be simulated from using the stationary bootstrap method.

Politis and Romano present the stationary bootstrap method (as cited in Alizadeh & Nomikos, 2008). According to Politis and Romano (as cited in Alizadeh & Nomikos, 2008, p. 20), Alizadeh and Nomikos (2008, p. 20), and Alizadeh and Nomikos’ paper “Investment Timing and Trading Strategies in the Sale and Purchase Market for Ships” (as cited in Alizadeh & Nomikos, 2008, p. 20), the stationary bootstrap method is a method of simulating many possible price paths representing what prices could have done over a given time period. Alizadeh and Nomikos (2008, p. 30) and Sullivan, Timmermann, and White as cited in (Alizadeh & Nomikos, 2008, p. 30) say that to do this, the user of the method must identify a smoothing parameter called q and supply the actual price path over the given time period.

I wrote a Python computer code that implements Politis and Romano's stationary bootstrap method (as cited in Alizadeh & Nomikos, 2008) as described by Sullivan et al. (as cited in Alizadeh & Nomikos, 2008, pp. 30-31). The description by Sullivan et al. (as cited in Alizadeh & Nomikos, 2008, pp. 30-31) includes a standard uniform random number labeled U and the q parameter; my code follows Sullivan, Timmermann, and White's (1999) instructions on what should occur if these two values are equal. The value 1,000 is used as the number of simulations in the Alizadeh and Nomikos (2008) paper and is also used as the number of simulations for this project. The value 0.1 is used as the q parameter value by Alizadeh and Nomikos (2008) and by Sullivan et al. (as cited in Alizadeh & Nomikos, 2008); it is also used as the q parameter value for this project.

The code uses the log first-differenced futures and log first-differenced spot price series. Given these actual log first-differenced futures and log first-differenced spot price series and $q = 0.1$, the code simulates 1,000 alternative possible log first-differenced futures and log first-differenced spot price series that could have occurred over the time period. The simulated log first-differenced futures and log first-differenced spot price series contain corresponding values. This means that when time t's data value from the actual log first-differenced spot price series is used for time x's data value for simulation number s's log first-differenced spot price series, the same time t's data value from the actual log first-differenced futures price series is used for the same time x's data value for the same simulation number s's log first-differenced futures price series. Using the time $= 0$ actual level futures and level spot prices and the simulated log first-differenced futures and log first-differenced spot price series, the code then builds 1,000 simulations containing alternative possible level futures and level spot price series that could have occurred over the time period. These simulated level futures and level spot price series are then outputted by the code and used in the next subsection by another Python code that applies various hedge ratio strategies to the prices.

Apply Hedging Strategies

Now that many possible price paths were simulated, I test various hedge ratio strategies on all the simulations to determine which strategy produces the optimal example results. The following two level-two subsections cover how an optimal hedge ratio can be solved for by finding terminal cumulative cashflow, daily cumulative cashflow, and terminal cumulative profit measurements for each hedging strategy and how the results can be compiled and interpreted, respectively.

Unless otherwise specified, in this "Apply Hedging Strategies" subsection, all calculations, graphs, and tables present results only from dates in the time period that have both a spot and futures price listed in the spot and futures datasets, respectively. In this subsection, I also refer to the last date, day, or observation in the time period several times; in each instance, I am referring to the last date, day, or observation in the time period that has both a spot and futures price listed in the spot and futures datasets, respectively.

In this subsection, I use many of the ideas and methods used by Bollen and Whaley (1998) to complete my analysis including testing a wide range of hedge ratios, finding key indicator values such as the terminal cumulative cashflow, finding and considering the implications of the cumulative cashflow at many points throughout the time period as each strategy is being applied, creating and analyzing a terminal cumulative profit distribution graph, and providing the type of vocabulary necessary to analyze the results of this type of simulation and these types of graphs, among other things.

Solving for cumulative profits and cashflows of different hedging strategies. I wrote another Python computer code that uses the purchases, sales, and inventory level datasets and applies various hedging strategies to the simulated level spot and level futures prices. The code determines the terminal cumulative profit, terminal cumulative cashflow, and daily cumulative cashflow of each simulation for each hedge ratio being tested. To calculate these values, the code tracks futures market profit and futures market cashflow implications from futures price data, Lower Foods' daily inventory level, transaction costs, margin calls, and other relevant measures by following a mark to market process. To calculate these values, the code also tracks spot market profit and spot market cashflow implications from purchases data, sales data, changes in the value of Lower Foods' daily inventory level, and other relevant measures.

I follow the example and instructions provided by Hull (2015b) as a baseline to understand how to keep track of the profits associated with the daily settled margin account Lower Foods would use if they held positions in futures contracts. As Hull (2015b) specified, a margin call occurs if the value in the margin account is less than the maintenance margin; therefore, if the two values are equal, a margin call would not occur. My code does not follow Hull's (2015b) textbook exactly, however, because his textbook example did not include all the complexities included in this project. At least one detail in my code is different from how Hull (2015b) detailed the mark to market process. He specified that margin calls are responded to the day after they are triggered or, in other words, the day after the amount of money in the margin account becomes less than the amount of money determined as the maintenance margin; however, for simplicity in my code, I assume that margin calls are responded to on the same day they are triggered.

The hedge ratio strategies tested for Lower Foods in this project are 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, and the minimum variance hedge ratio. Most of these are all ratios Bollen and Whaley (1998) tested in their analysis. These ratios give a wide range of possible hedging strategies to compare from no hedging, to full hedging, to minimum variance hedging, and many other possible strategies in between. The code assumes that the minimum variance hedge ratio is always one of the hedge ratios being tested and is always the last value in the variable h_array, which is an array containing all the hedge ratios to be tested. All code files introduced for the first time in the remainder of this "Procedures and Results" section of the paper also make these assumptions. These assumptions are adhered to in this project for each grade.

Some of the other assumptions imbedded into the code that must be adhered to and understood for the code to run properly are described below. First, there must be at least two observations in each simulation. For example, if daily data is provided, you must be testing how a given hedging strategy would perform over the course of at least one full day. In other words, data must be provided for a minimum time period of the initial day and at least one additional day if the data is daily. All the grades of the bottom round flat meat cut must adhere to this requirement for the code to work. For simplicity, I assume that the margin account interest rate is 0% and that the inventory level is never negative. Since it is not realistic to buy a fraction of a futures contract, the code always rounds to the nearest whole contract when determining how many futures contracts to sell or purchase.[16] Also, the code assumes that all outstanding futures positions on the last day in the time period are closed.

The code is built to work even if negative hedge ratios are tested, meaning that Lower Foods would buy futures contracts rather than sell them. Given the assumptions and specifications in this paper, this capability is currently used in the code when the applied hedging strategy is the minimum variance hedge ratio since the minimum variance hedge ratio is negative at many points throughout the simulations. Hull (2015a) served as my guide for how to calculate minimum variance hedge ratios. My calculation differs from Hull's (2015a) calculation because I use log first-differenced spot and log first-differenced futures prices in the regression rather than level first-differenced spot and level first-differenced futures prices.[17]

After the initial minimum variance hedge ratio is calculated at the beginning of each time period, the minimum variance hedge ratio is recalculated periodically throughout the time period on what I refer to throughout the rest of this paper as *rehedge dates*. Also, I use various forms of the word *rehedge* throughout the rest of this paper and appendices; for example, the word *rehedged* is used as a past-tense verb meaning that the hedge ratio being used was recalculated. In the code, I assume that the minimum variance hedge ratio is recalculated each time there is a new front-month futures contract in the futures contract I am hedging with. In

---

[16] The function I use is supposed to round numbers with a five in the tenths decimal place to the nearest whole number farther away from zero ("numpy.rint," 2017); however, a simple example of that function in my practice code file rounds these numbers to the nearest whole number closer to zero. For example, the function is supposed to round 3.5 to 4 and -3.5 to -4 ("numpy.rint," 2017), but in my practice code, that function would round 3.5 to 3 and -3.5 to -3. This leads me to believe that the code I use for this project also rounds these numbers to the nearest whole number closer to zero.

[17] I use the log first-differenced form of the prices because that is how my professor Dr. Tyler Brough, Ph.D. did it in an example tutorial for one of his courses (Brough, 2017a).

this project, there is a rehedge date about every two months since February, April, June, August, October, and December are the only futures contract months available for Live Cattle electronic futures contracts.[18]

On rehedge dates, when the minimum variance hedge ratio is used as the hedging strategy, the size of the position entered into of the upcoming front-month futures contract is determined using a newly calculated minimum variance hedge ratio. The code assumes that no rehedging takes place on the last observation in the time period, even if that observation date otherwise would be a rehedge date. This is because it seems excessive to rehedge on the last date in the time period even if it is the expiration date of the current front-month futures contract, since the code assumes all futures positions will be closed on the last day of the time period anyway.

In this project, the minimum variance hedge ratio for each rehedge date is calculated by regressing the previous hedge period's simulated log first-differenced spot prices on the previous hedge period's simulated log first-differenced futures prices; this follows the previously discussed calculation revised from the Hull (2015a) textbook. This time period is used for the regression because Hull (2015a) specified that historical data can be used to calculate minimum

[18] A webpage containing contract specifications for the Live Cattle electronic futures contract said that there are nine contract months available, but then it proceeded to list only six ("Live Cattle Futures," n.d.). The front-month futures data for this contract, however, lists only six contract months in its historical data. Therefore, I assume that the nine-month statement in the webpage containing contract specifications is an error ("Live Cattle Futures," n.d.); I assume that the six contract months listed on this webpage are the only contract months truly available for this futures contract since they also match the months listed in the historical futures data. The six months listed on this webpage and represented in the futures data are the ones listed in this paper as available contract months for this contract, implying a new front-month futures contract about every two months.

variance hedge ratios and that the period of time and number of observations the data in the regression should span across should be about the same period of time and number of observations that the minimum variance hedge ratio will be used. To follow this specification, I use the simulated data from the previous hedge period to determine the minimum variance hedge ratio for the next hedge period. This presents a problem only for the initial minimum variance hedge ratio calculation since I do not have data for any previous hedge period. As a solution, for each simulation, on the initial day in the time period, I calculate the minimum variance hedge ratio by regressing the log first-differenced spot prices from the first hedge period of the actual price path (not a simulated price path) on the log first-differenced futures prices from the first hedge period of the actual price path (not a simulated price path). Another scenario where Hull's (2015a) specifications are not followed in this project has to do with the equal time period specification discussed previously in this paragraph. For simplicity, regardless of the length of the portion of the previous hedge period that is included in the time period and how it compares to the length of the portion of the next hedge period that is included in the time period, the code still uses the data from the entire portion of the previous hedge period that is included in the time period to calculate the minimum variance hedge ratio for the entire portion of the next hedge period that is included in the time period.

With all hedge ratio strategies, the code assumes that the strategy is maintained by always holding a position in the front-month futures contract. This means that on the day one front-month contract expires, the position in the expiring front-month futures contract is closed by taking an equal, but opposite, position in that contract; simultaneously, a position in the next contract that will officially be the front-month contract on the following day is entered into. Since the position in the new contract is opened the day before it is technically the front-month futures contract, the front-month futures data does not provide the price of this new contract on that day. Since I must have a price for this new contract to calculate the position's value, the position's margin requirements, and other measures, I assume that the price of this new contract on this day is equal to the closing price of the actual, or expiring, front-month contract on this day, which is the expiration date of the actual front-month contract. In this project, since new contracts are being entered into at each rehedge date, there are new initial margin requirements that must be met at each of these dates and new maintenance margin requirements that must be met thereafter. These new margin requirements affect the balance of the margin account.

The code assumes that futures cashflows only occur when initial margins and margin calls are put in the margin account; when transaction costs are charged upon both opening and closing futures positions; at all rehedge dates, when all the money in the margin account, net of the new initial margin and transaction costs, is pulled out of the margin account and realized by the investor; and at the end of time period, when the investor empties the margin account and realizes its full value net of transaction costs. The code, therefore, assumes that money is only taken out of the margin account at rehedge dates and on the last day in the time period.

Again, the number of simulations used in this project is 1,000, which is also the number of simulations used in the Alizadeh and Nomikos (2008) paper. Their paper also assumed a transaction cost percentage of 0.2% (Alizadeh & Nomikos, 2008); I use this same assumption. The contract size for the Live Cattle electronic futures contracts is 40,000 pounds per contract ("Live Cattle Futures," n.d.). Leuthold, Junkus, and Cordier (2000, p. 38) say that initial and maintenance margin percentages are usually around 10% and 7.5% of the total value of the positions entered into, respectively; I use these assumptions for this project, also.

The variable $num\_contract\_names$ represents the number of front-month futures contracts that are used to hedge throughout the entire time period excluding information associated with dates that are missing the spot or futures price from the spot or futures datasets, respectively. For the time period discussed, there are 21 front-month contracts used for hedging in this project that meet these criteria. Notice that this is not the number of front-month contracts available for use throughout the entire time period excluding information associated with dates that are missing the spot or futures price from the spot or futures datasets, respectively, but rather, the number of front-month contracts that are used to hedge throughout the entire time period excluding information associated with the previously discussed dates. This is significant since it is possible to have a new front-month contract available to hedge with on the last day in the time period, but, because of the assumption the code makes that no rehedging will take place on the last date in the time period, the front-month contract available for just that last day in the time period is not used for hedging and would, therefore, not be included in the count for the $num\_contract\_names$ variable. Since this special case does not appear for the combination of time period, cut and grades of meat, and futures contract used in this project, this specification does not affect the contract count in the $num\_contract\_names$ variable for this project. The variable $num\_obs\_first\_hedge\_period$ represents the number of log first-differenced observations in the first hedge period before the position is rehedged. For the Select, Choice, and CAB grades, this value is 35, 38, and 8 observations, respectively.

The code assumes that daily net negative purchased pounds, which I assume refers to when Lower Foods returns inventory to the seller, are refunded to Lower Foods at the return transaction day's purchase price per pound plus the freight it cost Lower Foods when originally purchasing the inventory. The code assumes Lower Foods is not responsible for the return shipping for daily net negative purchased pounds. The code also assumes that daily net negative sales pounds, which I assume refers to when Lower Foods' customers return inventory to them, are refunded to the customer at the return transaction day's simulated spot price per pound minus the freight it originally cost Lower Foods to send the inventory to the customer. The code assumes Lower Foods is not responsible for the return shipping for daily net negative sales pounds, either.

All freight costs represented in the code refer to freight costs Lower Foods is responsible for paying. The freight cost per pound for inventory Lower Foods purchases is included in the prices listed in the purchases data (Morris, personal communication, November 3, 2017). Therefore, Lower Foods is responsible for no additional freight above what is already included in these purchase prices. I assume that the freight cost per pound for inventory Lower Foods sells is not included in the prices listed in the sales data. I further assume that the freight cost Lower Foods is responsible for that is not already reflected in the weighted average sales price per pound in the sales data is $0.05 per pound.[19] [20] [21]

---

[19] Although I did ask a little about the freight cost for sales, I didn't understand the full details of the responses and was unsure if different responses provided different information. However, based on what I understood from various personal communications with Lower Foods representatives, I believe $0.05/lb is a reasonable assumption. I did not directly ask if the sales freight is included in the sales data prices, so this is also stated as an assumption. One reason I choose to state all sales freight information in this paper as assumptions is to avoid misrepresenting information from various personal communications.

[20] The freight variable values are dynamic in the Python code and, therefore, can be changed. While explicit instructions on how to change each dynamic variable in the Python codes are not provided, the variables names of all dynamic variables are defined in Appendix A, which will aid the user should they decide to change dynamic variable values.

[21] I am fairly certain I ran the code files for this project using $0.05/lb as the freight cost Lower Foods is responsible for that is not already reflected in the

It is important to note that this is one of the code files that needs the previously discussed additional information from Lower Foods in order for the results to provide an accurate representation of possible ways Lower Foods should strategize in the futures market. I proceeded to run the code with example inventory level data and example sales data values that stay constant throughout the time period; the constant data values I use for the net inventory level pounds, net daily pounds sold, and weighted average sales price per pound are 400,000 pounds, 1,000 pounds, and $2.50, respectively.[22] Therefore, the results discussed in the remainder of this paper do not represent a strategy recommendation being made to Lower Foods. Lower Foods will need to create an additional file for each grade (i.e.- "cab_inventory.csv") in order to get accurate strategy results for their company. Details on how these files must be created, including format, are included in Appendix A.

Compilation and interpretation of results. I wrote one final Python computer code that compiles the vast quantities of results obtained from the previous code and produces

---

weighted average sales price per pound in the sales data; however, I may have used $0.07/lb for this variable value. Since the run of certain code files for this project was done with example data for certain datasets since not all Lower Foods' actual data was provided, the results produced and presented in this paper are hypothetical regardless of whether $0.05/lb or $0.07/lb was used. Regardless of which of these values was used, this paper still provides an example of how to go about analyzing results that are produced by the Python code files.

[22] This constant value used for the example inventory level data does not correspond with information in the sales and purchases data, meaning that any given day's inventory level data is not a result of adding the previous day's net purchased pounds and subtracting the previous day's net sold pounds from the previous day's inventory level. Even if the inventory level data was calculated in this way, however, it would still be example data since one of the components of the calculation is example data; therefore, the results presented in this paper from the example analysis would still not be applicable to Lower Foods.

informative tables and graphs for each grade that can be used to help Lower Foods determine the optimal hedge ratio they should use when selling these futures contracts.

Figures 13, 14, and 15 in Appendices Q, R, and S, respectively, show the distribution of all the minimum variance hedge ratios calculated across the entire time period and across all simulations for the Select, Choice, and CAB grades, respectively; for each figure, the minimum variance hedge ratio is calculated according to the description and assumptions detailed in the "Solving for cumulative profits and cashflows of different hedging strategies" level-two subsection. Each figure roughly shows a bell curve around a hedge ratio between 0.0 and 0.5, with varying levels of skewness. These graphs are summarized with summary statistics in Table 4 in Appendix T. This information is provided to show what values are being used as the minimum variance hedge ratios.

Table 5 in Appendix U shows terminal cumulative cashflow summary statistics and daily cumulative cashflow summary statistics for various hedge ratios for the CAB grade. In Table 5, h_value represents the hedge ratio being tested; $term\_mean$, $term\_min$, $term\_max$, and $term\_std$ represent the terminal cumulative cashflow's mean, minimum, maximum, and standard deviation across all simulations, respectively; and $daily\_mean$, $daily\_min$, $daily\_max$, and $daily\_std$ represent the daily cumulative cashflow's mean, minimum, maximum, and standard deviation within each entire time period averaged across all simulations. Values in parentheses represent negative numbers.

Table 6 in Appendix V shows terminal cumulative profit summary statistics for various hedge ratios for the CAB grade. In Table 6, h_value represents the hedge ratio being tested; $term\_mean$, $term\_min$, $term\_max$, and $term\_std$ represent the terminal cumulative profit's mean, minimum, maximum, and standard deviation across all simulations, respectively. Again, values in parentheses represent negative numbers.[23]

Note that the terminal cumulative cashflow summary statistics are not identical to the terminal cumulative profit summary statistics because they do not take into account the change in the value of Lower Foods' inventory; the terminal cumulative profit summary statistics, on the other hand, do incorporate the inventory value changes into their results. Inventory value changes are not yet realized gains or losses and do not involve an exchange of cash; I do, however, account for these unrealized gains and losses in the terminal cumulative profit calculations.

To avoid confusion on the difference between terminal and daily values shown in Table 5, use the explanation in this paragraph of the difference between the meanings of the $term\_min$ and $daily\_min$ columns in Table 5. Looking at the $daily\_min$ column in Table 5, you see the values of the minimum daily cumulative cashflows throughout each entire time period averaged across all simulations. For the minimum variance hedge ratio, across the 1,000 simulations, the average minimum daily cumulative cashflow that will occur during the time

---

[23] Two runs of a previous version of one of the Python code files using two different sets of 1,000 simulated prices yielded inconsistent results of hedge ratio optimality from best to worst based on the maximum terminal cumulative cashflow column in a table produced. In fact, aside from the best ranked hedge ratio in this category, the optimality ranking of the other hedge ratio results were exactly inversed between the two code runs. All other columns of information in the table that were calculated from this previous version of the code produced relatively similar results between the two runs of the code. For the inconsistent column in the table, I am unsure why the data would show opposing results. Before applying a hedging strategy, the user of the code may wish to run all three Python code files discussed in Appendix A many times for each grade. This will show results for different sets of simulated prices and allow the user to determine any possible changes in results in any of the tables and/or graphs produced.

period is -$596,032. This differs from the minimum variance hedge ratio's value in the $term\_min$ column of Table 5 because this value in the $term\_min$ column means that, for the minimum variance hedge ratio, the minimum terminal cumulative cashflow across all 1,000 simulations is -$814,529. Therefore, the terminal values in Table 5 emphasize the results at the end of the time period while the daily values in Table 5 emphasize the results throughout the time period.

Figures 16, 17, and 18 in Appendices W, X, and Y, respectively, show information that will help the viewer determine the optimal hedge ratio. Figures 16 and 17 are simply bar graphs of some of the columns of Tables 5 and 6, showing the information in a more visual way. In both figures, the blue bars show summary statistics for the terminal cumulative cashflows and the orange bars show summary statistics for the terminal cumulative profits.[24] Figure 18 compares the distribution of terminal cumulative profits across all simulations for two different hedge ratios. The Python code provided to Lower Foods allows the user to choose which two hedge ratios they would like to compare. See Appendix A for instructions on how to do so. The user can create as many graphs like Figure 18 as they wish in order to compare different combinations of hedge ratios.

---

[24] It is somewhat difficult to discern the height differences in the graphs' bars since many of them are relatively similar in height. One reason I have not zoomed in on the graphs to make these values clearer, however, is because Lower Foods' data will almost certainly not match the example data I used to create these graphs. Therefore, forcing the y-axis to zoom in on specific values in either graph would very likely not show the part of the y-axis Lower Foods would need in order to see their actual data's results. When I later state what Figures 15 or 16 show, it may be hard to see these things in the not-zoomed-in figures presented in this paper, but these things can be confirmed through closer inspection of the relevant columns in Tables 5 and 6.

How to interpret these tables and graphs is important to understand so the user can make an educated decision about which hedge ratio is optimal. I will walk through the CAB grade's tables and figures shown in this paper as an example of how this information can be interpreted; however, since this data does not reflect Lower Foods' real sales or inventory level data, the hedge ratios shown to be optimal in the tables and graphs in this paper do not represent actual optimal hedge ratios for Lower Foods and this should not be considered a recommendation. Lower Foods can refer to Appendix A for detailed instructions on how to get relevant data for their company. The following analysis, therefore, is an example of what types of tables and figures will be produced by the Python code files provided to Lower Foods and how the information contained therein should be interpreted.

Table 5, Table 6, Figure 16, and Figure 17 should be used to determine which hedge ratios appear to be optimal. The hedge ratio that appears to be optimal should then be compared to alternative hedge ratio strategies under further scrutiny in a figure such as Figure 18. Figure 18 can also be used to compare any combination of two hedge ratios being analyzed in this project; it can also be used to help determine which hedge ratio is optimal. Figure 16 is simply a visual representation of the $term\_mean$ columns in Tables 5 and 6; Figure 17 is a visual representation of the $term\_std$ columns in Tables 5 and 6. Understanding how these figures relate to the tables can help the user better understand the information being presented.

Notice that Figure 16 shows the minimum variance hedge ratio as having both the highest mean terminal cumulative cashflow and the highest mean terminal cumulative profit, followed by the hedge ratios 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and then 1.0 for both mean terminal cumulative cashflow and mean terminal cumulative profit. This suggests that the minimum variance hedge ratio would be the best hedge ratio to use. However, before this conclusion is made, all factors must be taken into consideration.

Figure 17 provides more information on which hedge ratios may be the best. It shows that the hedge ratios in order from lowest to highest standard deviation of terminal cumulative cashflows are 0.0, 0.1, 0.2, 0.3, 0.4, 0,5, the minimum variance hedge ratio, 0.6, 0.7, 0.8, 0.9, and then 1.0. Alone, this could be interpreted as suggesting that the hedge ratio of 0.0 would be the best hedge ratio to use because it has the smallest variance of possible results. Figure 17 also shows that a hedge ratio of 1.0 has the lowest terminal cumulative profit standard deviation, followed by the hedge ratios 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, the minimum variance hedge ratio, 0.1, and then 0.0.[25] This shows that, while the 0.0 hedge ratio may have the lowest terminal cumulative cashflow standard deviation across all simulations, it's standard deviation of the terminal cumulative profits across all simulations is higher than that for all other hedge ratios. Whether this makes the 0.0 hedge ratio better or worse than the 1.0 hedge ratio, which has the highest terminal cumulative cashflow standard deviation out of the hedge ratios being tested but the lowest terminal cumulative profit standard deviation, depends on the company's preferences, the implications for the company of the cashflow versus profit results, future cashflow and profit expectations for the different hedging strategies, and future inventory value expectations.

---

[25] This result that the minimum variance hedge ratio has a higher standard deviation of terminal cumulative cashflows than many of the other tested hedge ratios and higher standard deviation of terminal cumulative profits than most of the other tested hedge ratios would not be expected. This may be due to the example data values that I use in this project (Brough, personal communication, November 14, 2017).

The information in the other columns of Tables 5 and 6 should also be further studied to understand other factors that could contribute to the decision of what hedge ratio to use. For example, Table 5 shows that the minimum variance hedge ratio has an average minimum daily cumulative cashflow of -$596,032 and a minimum terminal cumulative cashflow of -$814,529. While the minimum terminal cumulative cashflow value for the minimum variance hedge ratio is not as low as many other hedge ratios' minimum terminal cumulative cashflow values, a company still must consider whether they have the cash necessary to get through the minimum variance hedge ratio's average minimum daily cumulative cashflow value, which is expected to occur at some point throughout the time period. If not, the company should choose a hedge ratio with a higher average minimum daily cumulative cashflow value.

The standard deviation results pull into question whether the minimum variance hedge ratio, which ranked the best when comparing hedge ratios based on mean terminal cumulative cashflows and mean terminal cumulative profits, is a preferred hedge ratio to a hedge ratio that scored better when it came to standard deviations. This depends on a combination of the risk preferences of the company, the terminal cumulative profit distribution and/or the terminal cumulative cashflow distribution, and the other summary statistics associated with results of the hedge ratio in question.[26]

---

[26] The Python code files I wrote produce a terminal cumulative profit distribution graph, but not a terminal cumulative cashflow distribution graph. One reason for this is because the changing value of Lower Foods' inventory throughout the time period would not be reflected in a terminal cumulative cashflow distribution graph and the adverse changes in the value of the inventory is precisely what the hedge ratios are hedging against. Therefore, I don't feel that a terminal cumulative cashflow distribution graph is as good of a tool for comparing the hedge ratios.

Suppose a company determines that the minimum variance hedge ratio could potentially be their optimal hedge ratio because it ranked best when it came to mean terminal cumulative profit and mean terminal cumulative cashflow. Figure 18 shows the type of graph by which a company can further compare a potential optimal hedge ratio, such as the minimum variance hedge ratio in this example, to another hedge ratio. Specifically, Figure 18 compares the terminal cumulative profit distribution across all simulations for the minimum variance hedge ratio to that of the 1.0 hedge ratio. The figure shows that, although the minimum variance hedge ratio has a higher mean terminal cumulative profit (which is hard to tell from Figure 18, but is shown in Table 6), it's probabilities of achieving the extremely high and extremely low terminal cumulative profit values appear to be generally higher than the 1.0 hedge ratio's probabilities of achieving the extreme terminal cumulative profit values. The minimum variance hedge ratio's probabilities of achieving the middle range of terminal cumulative profit values appear to be generally lower than the 1.0 hedge ratio's probabilities of achieving these values. In this example, using the 1.0 hedge ratio instead of the minimum variance hedge ratio appears to lower the probability of achieving the extremely high terminal cumulative profit values, or, in other words, decreases the upside risk, which is not beneficial for the company; however, using the 1.0 hedge ratio instead of the minimum variance hedge ratio also appears to lower the probability of achieving the extremely low terminal cumulative profit values, or, in other words, decreases downside risk, which is beneficial to the company. Given this analysis, a company with more risk averse preferences should, therefore, prefer a hedge ratio of 1.0 to the minimum variance hedge ratio in this example.

Figures such as Figure 18 can be recreated over and over again, comparing any combination of two hedge ratios from those included in this analysis including 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, and the minimum variance hedge ratio. Instructions on how to do this are included in Appendix A. For example, a company may want to compare the terminal cumulative profit distribution of a hedge ratio they are considering using to the terminal cumulative profit distribution of a hedge ratio they are currently using.

CONCLUSION

First, the “Statistical Analysis” subsection of this paper shows that, for each of the three grades of the bottom round flat meat cut, the log first-differenced spot and log first-differenced futures prices, calculated under one method,[27] are unit root stationary. It also shows that each log spot price series is cointegrated with the log futures price series, meaning that, since I fail to reject that the cointegrating parameter equals one, the log basis series is stationary. Since the log first-differenced spot and log first-differenced futures price series are unit root stationary, the stationary bootstrap method is then used on the log first-differenced spot and log first-differenced futures price series, calculated under another method that I assume also yields unit root stationary price series,[28] in the “Stationary Bootstrap Method” subsection to create many simulations of possible price paths that potentially could have occurred over the specified time period. In the “Apply Hedging Strategies” subsection, several hedging strategies including but not limited to hedging fully, not hedging at all, and using the minimum variance hedge ratio are then applied to the simulated price paths. Using results built on the example data, this subsection then walks through how results can be interpreted to help Lower Foods compare hedge ratios and choose an optimal hedge ratio for each grade of meat.

As previously explained, much of this paper uses the ideas and methods used by Bollen and Whaley (1998) in their paper “Simulating Supply” and by Alizadeh and Nomikos (2008) in their paper “Performance of Statistical Arbitrage in Petroleum Futures Markets.”

---

[27] The method mentioned here is to complete the calculations from information associated with dates in the time period where dates in each individual dataset with missing prices are omitted.

[28] The method mentioned here is to complete the calculations only from information associated with dates in the time period that have both a spot and futures price listed in the spot and futures datasets, respectively.

A few additional data pieces are needed from Lower Foods before the different hedge ratios can be compared to receive results relevant to their company; assumptions and example data values are used in place of these data. Therefore, the analysis of an optimal hedge ratio presented in this paper does not apply to Lower Foods. Lower Foods will need to gather and incorporate the necessary data pieces relevant to their company into the Python computer code files. Appendix A provides detailed instructions on how this can be done. This project, along with the necessary data files and Python code files that will be provided to Lower Foods, will help equip them with the tools necessary to solve for the relevant optimal hedge ratios their company can use for the Select, Choice, and CAB grades of the bottom round flat meat cut.

Since this paper can help Lower Foods determine which hedging strategy to use, it can help them determine how they can best participate in basis trading. With any positive level of futures positions Lower Foods sells, they would be offsetting gains (losses) in the spot market with losses (gains) in the futures market, resulting in their making profits based on the basis, or price difference, between the spot and futures markets rather than hoping for favorable moves in the spot prices to make profits, which is what would occur with a hedge ratio of 0.0. This basis trading can be preferable because the basis is more predictable than spot prices.

Lower Foods has a natural long position in their inventory since they wish to sell it. Because Lower Foods is trying to protect the value of their inventory, this paper looks at hedge ratios that would provide them with a short position in the futures market, meaning they would be selling futures contracts. The dynamic minimum variance hedge ratio in this paper is the only exception to this, since it sometimes uses a negative hedge ratio, suggesting Lower Foods will sometimes buy futures contracts to protect the value of their inventory if this hedge ratio is used.

In summary, this paper presents the process of how Lower Foods, Inc., a local company, can use data regarding their inventory levels of three grades of the bottom round flat meat cut to see how various hedge ratios, when applied to many simulations of possible spot and futures price paths, affect key indicators such as the mean terminal cumulative cashflow, mean terminal cumulative profit, and standard deviation of the terminal cumulative profits. These key indicators can help Lower Foods determine what the optimal hedge ratio would be for their company for each grade of this meat cut. The optimal hedge ratio could then serve as the optimal storage strategy for Lower Foods to help maximize profits and cashflows and minimize risk.

REFERENCES

Alizadeh, A. H., & Nomikos, N. K. (2008). Performance of statistical arbitrage in petroleum futures markets. *The Journal of Energy Markets, 1*(2), 3-33.

Anaconda3 (Version 4.3.0) [Software]. (2016, December 23). Retrieved from http://www.anaconda.com/download/

Anaconda3 (Version 5.0.1) [Software]. (2017, October 25). Retrieved from http://www.anaconda.com/download/

Bollen, N., & Whaley, R. (1998, September). Simulating supply. *Risk*, 143-147.

Brough, T. (2017a, April 5). Adapted from *Minimum variance hedging tutorial* [Rmarkdown]. Personal code of T. Brough, Utah State University, Logan, UT. Retrieved from USU Canvas: https://usu.instructure.com/courses/443764/files/folder/Midterm/Tutorials/Min-Var%20Hedging which I have access to because I was a member of T. Brough's Spring 2017 FIN 6470 Section 001 course.

Brough, T. (2017b, April 6). Adapted from *Engle-Granger method tutorial* [Rmarkdown]. Personal code of T. Brough, Utah State University, Logan, UT. Retrieved from USU Canvas: https://usu.instructure.com/courses/443764/files/folder/Midterm/Tutorials/Engle-Granger which I have access to because I was a member of T. Brough's Spring 2017 FIN 6470 Section 001 course.

CME definitions. (n.d.). In *CME rulebook* (iii). Retrieved from CME Group: http://www.cmegroup.com/rulebook/files/CME_Definitions.pdf

Download Anaconda distribution. (2017, October 25). Retrieved from Anaconda: https://www.anaconda.com/download/

Feeder cattle Jan '18 (GFF18). (2017). Retrieved from barchart: https://www.barchart.com/futures/quotes/GF*0/all-futures

Feeder cattle futures contract specs. (n.d.). Retrieved from CME Group: http://www.cmegroup.com/trading/agricultural/livestock/feeder-cattle_contract_specifications.html

Feeder cattle pit Jan '18 (FCF18). (2017). Retrieved from barchart: https://www.barchart.com/futures/quotes/FC*0/all-futures

Ferrin. (2017, May 5). Adapted from zoo library plotting portions in *FIN 6370 midterm* [Rmarkdown]. Personal code on computer of D. Ferrin, Utah State University, Logan, UT.

Goldwyn, M. (2010). Buying beef: The science of beef grades and labels, and busting the Kobe beef myth. Retrieved August 15, 2017, from AmazingRibs.com: http://amazingribs.com/recipes/beef/zen_of_beef_grades.html

Hull, J. C. (2015a). Hedging strategies using futures. In D. Battista, E. McDonagh, E. Senra-Sargent, & J. Holcomb (Eds.), *Options, futures, and other derivatives* (9th ed., pp. 49-76). Pearson.

Hull, J. C. (2015b). Mechanics of futures markets. In D. Battista, E. McDonagh, E. Senra-Sargent, & J. Holcomb (Eds.), *Options, futures, and other derivatives* (9th ed., pp. 24-48). Pearson.

Leuthold, R. M., Junkus, J. C., & Cordier, J. E. (2000). Futures contracts, exchanges, and trading. In *The theory and practice of futures markets* (pp. 9-44). Champaign, IL: Stipes Publishing L.L.C.

Live cattle Feb '18 (LEG18). (2017). Retrieved from barchart: https://www.barchart.com/futures/quotes/LE*0/all-futures

Live cattle futures contract specs. (n.d.). Retrieved from CME Group: http://www.cmegroup.com/trading/agricultural/livestock/live-cattle_contract_specifications.html

Live cattle pit Feb '18 (LCG18). (2017). Retrieved from barchart: https://www.barchart.com/futures/quotes/LC*0/all-futures

Make smarter choices. Faster. Every day. (n.d.). Retrieved from Urner Barry Comtell: http://www.ubcomtell.com/Login/LoginNew.aspx?ReturnUrl=/CTDefault.aspx

McDonald, R. L. (2006). Glossary. In D. Clinton, D. Battista, M. S. Anderson, & M. Groth (Eds.), *Derivatives markets* (2nd ed., pp. 907-919). Boston, MA: Pearson Education, Inc.

numpy.rint. (2017, June 10). In *NumPy v1.13 manual*. Retrieved from SciPy.org: https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.rint.html

Pfaff, B. (2016, September 6). ur.df: Augmented-Dickey-Fuller unit root test. In *urca: Unit root and cointegration tests for time series data v1.3-0*. Retrieved from RDocumentation: https://www.rdocumentation.org/packages/urca/versions/1.3-0/topics/ur.df

Raybaut, P., The Spyder Project Contributors. (2017, January 24). Spyder: The Scientific PYhon Development EnviRonment Python 3.6.0 (Version 3.1.2) [Software]. Retrieved from https://www.anaconda.com/download/

RStudio Team. (2016). RStudio: Integrated Development for R (Version 1.0.136) [Software]. Boston, MA: RStudio, Inc. Retrieved from http://www.rstudio.com/

Sullivan, R., Timmermann, A., & White, H. (1999, October). Appendix C. The stationary bootstrap. In Data-Snooping, technical trading rule performance, and the bootstrap. *The Journal of Finance, 54*(5), 1689-1690. Retrieved December 12, 2017, from https://www.kevinsheppard.com/images/0/0d/Sullivan_Timmermann_White.pdf

Trading venues (pit vs online). (n.d.). Retrieved from CME Group CME Institute: https://institute.cmegroup.com/modules/trading-venues-pit-vs-online

UB 171B 3 rnd, out. rnd CAB - savg $/lb [Data file]. (2017, June). Available from Urner Barry Comtell.: http://www.ubcomtell.com/Login/LoginNew.aspx?ReturnUrl=/CTDefault.aspx

UB 171B 3 Round, Outside Round CAB. (2017, June 12). Available from Urner Barry Comtell: http://www.ubcomtell.com/Login/LoginNew.aspx?ReturnUrl=/CTDefault.aspx

USDA 171B 3 rnd, out. rnd CH - wtdavg $/lb [Data file]. (2017, June). Available from Urner Barry Comtell.: http://www.ubcomtell.com/Login/LoginNew.aspx?ReturnUrl=/CTDefault.aspx

USDA 171B 3 rnd, out. rnd SE - wtdavg $/lb [Data file]. (2017, June). Available from Urner Barry Comtell.: http://www.ubcomtell.com/Login/LoginNew.aspx?ReturnUrl=/CTDefault.aspx

USDA 171B 3 Round, Outside Round Choice. (2017, June 12). Available from Urner Barry Comtell: http://www.ubcomtell.com/Login/LoginNew.aspx?ReturnUrl=/CTDefault.aspx

USDA 171B 3 Round, Outside Round Select. (2017, June 12). Available from Urner Barry Comtell: http://www.ubcomtell.com/Login/LoginNew.aspx?ReturnUrl=/CTDefault.aspx

Wooldridge, J. M. (2009). *Introductory econometrics: A modern approach* (4th ed.). J. W. Calhoun, M. Worls, L. Bofinger, & D. Kumar (Eds.). Mason, OH: South-Western Cengage Learning.

APPENDICES

Appendix A

How to Run the Python Computer Code Files

This how-to document is written for Lower Foods' benefit so they can run the Python computer code files with their actual data values at their earliest convenience to solve for their optimal hedge ratio. Detailed instructions are given herein regarding how to run the code files and interpret the output. I will provide Lower Foods with the files necessary to solve for the optimal hedge ratio for the Select, Choice, and CAB grades of the bottom round flat meat cut for the time period from January 1, 2014, through May 31, 2017, given the assumptions outlined in this paper. Figure 1 in Appendix B shows the file structure of the folders that will be provided to them. Refer to Figure 1 to navigate the filepaths to the folders referred to in this appendix. Lower Foods should save these folders to the computer they will run the Python code files on.

There are three main Python computer code files involved in this project. They are called "bootstrap.py," "apply_strategy.py," and "summary.py." Each of the three code files that is run must be run for each of the grades of the bottom round flat meat cut being analyzed including Select, Choice, and CAB. However, I will use the CAB grade for various examples throughout this document, even if it is not explicitly stated to be an example. For example, the instructions in the steps to follow may say that an input file for the code is "cab_and_fut_lndiff.csv"; however, when running the same code for the Select or Choice grades, the input file would be "select_and_fut_lndiff.csv" and "choice_and_fut_lndiff.csv," respectively.

Only the "apply_strategy.py" code and the "summary.py" code will need to be run by Lower Foods to obtain the optimal hedge ratio for the Select, Choice, and CAB grades of the bottom round flat meat cut for the time period from January 1, 2014, through May 31, 2017, given the assumptions outlined in this paper. All three code files would only need to be run if one or more of four scenarios occurs including if Lower Foods wishes to change the number of simulations from its preset value of 1,000; if Lower Foods wishes to change the value of the smoothing parameter used in the stationary bootstrap method from its preset value of 0.1; if a time period other than January 1, 2014, to May 31, 2017, is being analyzed; or if a meat cut and grade other than the bottom round flat Select, Choice, or CAB grade is being analyzed. In order to use the code files for alternative time frames (scenario three), alternative cuts or grades (scenario four), or both, the user would first need to determine whether an alternative futures contract should be used to hedge, use statistical methods to show that the log first-differenced spot and log first-differenced futures prices are stationary, create additional files with specific formatting, and adjust the values of three of the variables in the "apply_strategy.py" code. Note that, to make the same assumptions used in this paper, the log first-differenced spot and log first-differenced futures prices in the previous sentence must refer to the price series calculated from information associated with dates in the time period where dates in each individual dataset with missing prices are omitted. The three values that would need adjustment are the value of the string assigned to the variable $grade$, which refers to the grade of the bottom round flat meat cut the code will analyze; the value of the variable $num\_obs\_first\_hedge\_period$, which represents the number of observations in the first hedge period before the position is rehedged; and the value of the variable $num\_contract\_names$, which is the number of front-month futures contracts that are used to hedge throughout the entire time period excluding information associated with dates that are missing the spot or futures price from the spot or futures datasets, respectively. The value of $num\_obs\_first\_hedge\_period$ can be found by opening a specific file (i.e.-

"cab_and_fut_lndiff.csv") located in the Data\bootstrap folder, starting in cell I2 and counting downward in the column the number of times the same contract symbol appears before changing to a new symbol, and subtracting one from the resulting value. The value of $num\_contract\_names$ can be found by opening a specific file (i.e.- "cab_and_fut_lndiff.csv") located in the Data\bootstrap folder, identifying the number of unique symbols in column I, and subtracting one if and only if the final symbol name in the list is listed only once. If any of the four scenarios occur, all three Python code files would need to be run for each of the three grades.

In this appendix, I identify all the dynamic, or changeable, values in the Python code files. However, note that there are limitations to the extent of how dynamic these variables are. For example, the variable $grade$, which refers to the grade of the bottom round flat meat cut the code will analyze, is dynamic only to the extent that its value is either the string 'select,' the string 'choice,' or the string 'cab'; if the user wishes to change $grade$ to be a grade other than one of these options, additional work besides simply changing the variable is necessary, as discussed in the previous paragraph. Another example is that, if the value of the variable $sims$, which refers to the number of simulations, is changed in one of the three code files, it must be changed to the new value in all the other code files. Yet another example is that values in the variable h_array, which is an array containing the various hedge ratios you wish to test and compare, can be changed; however, the code is set up so that the minimum variance hedge ratio is always the final hedge ratio in h_array so, if the user wishes to exclude the minimum variance hedge ratio from the analysis, some of the Python code files would need revision. Changing other assumptions, parameters, or specifications in the paper above could also affect how variables are calculated or could require revision to the Python code files.

Due to the size of the tasks being completed, some of the code files take many hours to run, with the longest run times being about 12 hours each for the “apply_strategy.py” code for both the Select and Choice grades. This run time has the potential to be optimized through further coding strategies; however, that is beyond the scope of this project.

To prevent errors, review each level-two subsection and complete the level-three subsection’s set of instructions within it before moving on to review the next level-two subsection and complete the next level-three subsection’s set of instructions within it. Going through the instructions and sections in order will help ensure the correct files have finished being created and can be pulled into future code files correctly to give accurate results.

Finally, when one of the code files exports files into the folders, always let it overwrite any files that currently have the same filename. This will ensure that you will be working with current data relevant to your company rather than results from sample data I previously saved in the folders or outdated data. Also, do not move files to different folders, change which folders the files are imported from or exported to, change the filenames of the files that will be imported into other code files, or change the filenames dataframes or images from the code will be exported as.

Download software to run code. To use these Python code files, I recommend downloading a software called Anaconda3, which includes a program called Spyder. To do this, use the instructions in the level-three subsection within this level-two subsection. These instructions detail how to download the 5.0.1 version of Anaconda3 for a Windows operating system for a 64-bit computer (Anaconda3, 2017), which includes the 3.6 version of Python (“Download Anaconda,” 2017). These instructions help the user navigate and use the website the Anaconda3 program is downloaded from (“Download Anaconda,” 2017).

Instructions. Follow the instructions below**:**

1. Go to https://www.anaconda.com/download/.
2. Click on the Windows icon or Windows tab.
3. Click on the text "64-Bit Graphical Installer (515 MB)" in the box containing the text "Python 3.6 version."
4. A software called Anaconda3 will download to your computer. Install the program without changing any default settings. In the filepath ~\Anaconda3\Scripts, right click the file "anaconda-navigator.exe," click "Pin to Taskbar," then click on the item you just pinned to your taskbar.
5. Click on the "Launch" button under the Spyder icon. A program called Spyder will open. This is where you can run the Python computer code files provided.

Bootstrap code. This is the first of three Python code files involved in this project. This code does not need to be run unless Lower Foods wishes to change the number of simulations, the value of the smoothing parameter used in the stationary bootstrap method, run the code for a different time period than from January 1, 2014, to May 31, 2017, or run the code for a meat cut or grade other than the three specified in this paper. If one or more of these scenarios applies, follow the instructions in the level-three subsection within this level-two subsection; otherwise, this level-two subsection is merely to increase understanding of the code and can be skipped. If this level-two subsection is skipped, the user of this appendix would skip to the next subsection of equal heading level, which is titled "Apply_strategy code."

Instructions on how to run this code are listed below; however, a few additional details are first provided to understand the structure of the code's input and output files. All the input and output files are automatically set up to run through the program and the dynamic values are preset to match the assumptions outlined in the paper above. Therefore, the information in the following paragraph is just for your understanding and no action, besides what is listed in the instructions below, is required to run the code.

This code is called “bootstrap_code.py” and is located in the Data\bootstrap folder. For each grade, this code contains one input file (i.e.- “cab_and_fut_lndiff.csv”) which is located in the Data\bootstrap folder and four output files (i.e.- “cab_futures_bootstrap_level.csv,” “cab_spot_bootstrap_level.csv,” “cab_futures_bootstrap_lndiff.csv,” and “cab_spot_bootstrap_lndiff.csv”) which are all located in the Data\bootstrap\bootstrap_code_output folder. Values in the code that are dynamic and can be changed are q, which is the parameter discussed in the “Stationary Bootstrap Method” subsection in the paper above; $sims$, which is the number of simulations; and $grade$, which refers to the grade of the bottom round flat meat cut the code will analyze.

Instructions. Follow the instructions below*:*

1. In the Spyder program, click “File.” Then click “Open.” Navigate to the Data\bootstrap folder, click on the file “bootstrap_code.py,” then click “Open.”
2. On line 6 of the code, the variable $grade$ is assigned one of the three meat grades. Choose one of the meat grades, such as CAB and make line 6 read: grade = ‘cab.’ Notice that the grade is spelled in all lowercase letters and has single quotation marks around the word.
3. Click the green triangle button at the top of the screen, which will run the program.
4. The IPython console, which may be located on the right side of the Spyder program’s screen, will print the word *Success* when the program has finished

running. Do not close out of the Spyder program, make any changes to the text in the code, or open or move any of the input or output files while the code is running.

5. Return to step 2, choose a different meat grade, such as Choice, and make line 6 read: grade = 'choice.' Repeat steps 3-4.
6. Return to step 2, choose the final grade you have not yet chosen, such as Select, and make line 6 read: grade = 'select.' Repeat steps 3-4.

Apply_strategy code. This is the second of three Python code files involved in this project. This code must be run in order to solve for the optimal hedge ratio. Instructions on how to do so are listed below; however, a few additional details are first provided to understand the structure of the code's input and output files. All the input and output files are automatically set up to run through the program and the dynamic values are preset to match the assumptions outlined in the paper above. Therefore, the information in the following paragraph is just for your understanding and no action, besides what is listed in the instructions below, is required to run the code.

This code is called "apply_strategy.py" and is located in the Data\bootstrap\bootstrap_code_output folder. For each grade, this code contains six input files (i.e.- "cab_and_fut_lndiff.csv," "cab_inventory.csv," "cab_spot_bootstrap_level.csv," "cab_futures_bootstrap_level.csv," "cab_spot_bootstrap_lndiff.csv," and "cab_futures_bootstrap_lndiff.csv") which are all located in the Data\bootstrap\bootstrap_code_output folder except for one of the input files (i.e.- "cab_and_fut_lndiff.csv") which is located in the Data\bootstrap folder. For each grade, this code creates 15 output files (i.e.- "cab_cashflow_h_0.0.csv," "cab_cashflow_h_0.1.csv," "cab_cashflow_h_0.2.csv," "cab_cashflow_h_0.3.csv," "cab_cashflow_h_0.4.csv," “cab_cashflow_h_0.5.csv,” “cab_cashflow_h_0.6.csv,” “cab_cashflow_h_0.7.csv,” “cab_cashflow_h_0.8.csv,” “cab_cashflow_h_0.9.csv,” “cab_cashflow_h_1.0.csv,” “cab_cashflow_h_minvar.csv,” “cab_minvar.csv,” “cab_cashflow.csv,” and “cab_profit.csv”) which are all located in the Data\bootstrap\bootstrap_code_output\apply_strategy_output folder. Values in the code that are dynamic and can be changed are $grade$, which refers to the grade of the bottom round flat meat cut the code will analyze; $num\_obs\_first\_hedge\_period$, which represents the number of observations in the first hedge period before the position is rehedged; $sims$, which is the number of simulations; $num\_contract\_names$, which is the number of front-month futures contracts that are used to hedge throughout the entire time period excluding information associated with dates that are missing the spot or futures price from the spot or futures datasets, respectively; h_array, which is an array containing the various hedge ratios you wish to test and compare; $contract\_size$, which, for the Live Cattle electronic futures contract, is the number of pounds of Live Cattle involved in one contract; $initial\_mar\_per$, which is the percentage of the initial total value of the net contracts being entered into that must be deposited in the margin account; $main\_mar\_per$, which is the percentage of the total value of the net contracts owned that must be constantly maintained in the margin account; $transaction\_cost\_per$, which is the transaction cost percentage; $freight\_purchases\_dollars\_per\_lb$, which is the freight cost per pound for inventory Lower Foods purchases that they are responsible for paying and that is not already included in the purchases dataset’s weighted average prices per pound; and $freight\_sales\_dollars\_per\_lb$, which is the freight cost per pound for inventory Lower Foods sells that they are responsible for paying and that is not already included in the sales dataset’s weighted average prices per pound.

Instructions. Follow the instructions below*:*

1. Create and save an inventory .csv file for each grade according to the specifications below. See the “Data” section in the paper above for details about and the data discussed in this step, including sources from which the data was or will be gathered.
    a. Open the file “cab_inventory.csv” located in the Data\bootstrap\bootstrap_code_output folder.
    b. Erase all the values in columns B, C, and F, excluding the headings in cells B1, C1, and F1, respectively.
    c. Enter the net number of pounds of the bottom round flat CAB grade of meat sold by Lower Foods from January 1, 2014, to May 31, 2017, into column B for each respective date listed in column A. Do not change, add additional, or remove any dates from column A. If no sales transactions occurred on a day in column A, enter the value 0 into the corresponding cell in column B. If multiple sales transactions occurred on the same day, consolidate them and use the net pound value. Also, if date d has a sales transaction but is not listed in column A, consolidate date d’s sales pound value with the sales pound value for the next date after date d that is listed in column A, which I will refer to as date e. Date e should, therefore, reflect a net pound value, also. Apply these changes to all relevant dates. Make sure that positive pound values in column B represent the number of pounds Lower Foods sold to customers. Also enter negative pound values in

column B when relevant; I assume that these values represent the number of pounds of inventory Lower Foods' customers returned to them.

d. Enter the weighted average price per pound of the bottom round flat CAB grade of meat sold by Lower Foods from January 1, 2014, to May 31, 2017, into column C for each respective date listed in column A. Make sure these prices are in dollars. Do not change, add additional, or remove any dates from column A. If no sales transactions occurred on a day in column A, enter the value 0 into the corresponding cell in column C. If multiple sales transactions occurred on the same day, consolidate them and use the weighted average price per pound. Also, if date d has a sales transaction but is not listed in column A, consolidate date d's sales price per pound information with the sales price per pound for the next date after date d that is listed in column A, which I will refer to as date e. Date e should, therefore, reflect a weighted average price per pound value, also. Apply these changes to all relevant dates. Make sure that all prices in column C are positive.

e. Enter the number of pounds of the bottom round flat CAB grade of meat stored in Lower Foods' storage freezers, which I will refer to in this appendix as the inventory level data, into column F for each respective date listed in column A. Do not change, add additional, or remove any dates from column A. If date d has available inventory level data but is not listed in column A, consolidate date d's inventory level data with the inventory level data for the next date after date d that is listed in column A, which I will refer to as date e. Date e should, therefore, reflect a net pound value. Apply these changes to all relevant dates. If there is not data available for a date that is listed in column A, make an appropriate assumption. As discussed in the "Inventory Level Data" subsection in the "Data" section in the paper above, if data giving the daily pounds of inventory is not available, it may need to be solved for from other datasets. This can be done using the following steps:

i. Save the "cab_inventory.csv" file in the Data\bootstrap\bootstrap_code_output folder. Do not change the filename. It must read exactly as typed above. Be sure to overwrite the file that was previously saved under this filename.

ii. If you wish to do any of the following steps using equations in the spreadsheet's cells, also save the "cab_inventory.csv" file as a Microsoft Excel Workbook file type and complete the following steps in that file using equations.

iii. Gather data of the number of pounds of the CAB grade of the bottom round flat meat cut in Lower Foods' storage freezers on either January 1, 2014, or May 31, 2017, which are the starting and ending dates of the time frame being studied. Make sure that these values represent the inventory level on that day before that day's sales of inventory or purchases of new inventory have been incorporated into the measurement.

iv. Determine the series of daily inventory levels in pounds for the CAB grade of the bottom round flat meat cut in Lower Foods' freezers by using date t's (or date d's) number of pounds of inventory where the starting (or ending) date discussed in the previous step is used as the first date t (or date d) and the inventory poundage values associated with this date were therefore, gathered in the previous steps; subtracting that date t's (or adding date $d-1$'s) number of pounds of that cut and grade that was sold, which is found in column B of the "cab_inventory.csv" file created in a previous step and located in the Data\bootstrap\bootstrap_code_output folder; adding date t's (or subtracting date $d-1$'s) number of pounds of that meat's cut and grade that was purchased by Lower Foods and added into their storage freezers, or inventory, which is found in column D of the "cab_inventory.csv" file; recording the resulting value as date $t+1$'s (or date $d-1$'s) inventory level in pounds of that cut and grade; and repeating this reconstruction process where date $t+1$ (or date $d-1$) becomes date t (or date d) and the next $t+1$ (or $d-1$) inventory level value is found until daily inventory poundage levels have been determined for the entire time period.

f. Regardless of whether you completed the previous steps in a .csv file or an Excel Workbook file, save the completed file as a .csv file ("cab_inventory.csv") in the Data\bootstrap\bootstrap_code_output folder.

Do not change the filename. It must read exactly as typed above. Be sure to overwrite the file that was previously saved under this filename.

g. Repeat steps 1.a through 1.f in this level-three subsection for both the Select and Choice grades of the bottom round flat meat cut.

2. In the Spyder program, click “File.” Then click “Open.” Navigate to the Data\bootstrap\bootstrap_code_output folder, click on the file “apply_strategy.py,” then click “Open.”
3. Prepare the code to run for the CAB grade:
   a. Make line 8 of the code read: grade = ‘cab.’ Notice that the grade is spelled in all lowercase letters and has single quotation marks around the word.
   b. Make line 9 of the code read: num_obs_first_hedge_period = 8. Note that the text after the “#” sign on line 9 of the code is merely a comment and does not need adjusting.
4. Click the green triangle button at the top of the screen, which will run the program.
5. The IPython console, which may be located on the right side of the Spyder program’s screen, will print the values of what I label *h* and *j*, which are the hedge ratio round number the code is on and the simulation round number the code is on, respectively. The printing of these values takes place throughout the entire time the code is running and is designed to provide a means of knowing how far the code has run thus far. With the assumptions in this paper, since there are 12 hedge ratios being tested and 1,000 simulations are used, h can be any value from 0 to 11 and j can be any value from 1 to 1000. The code will say h is 0 and go through every j value from 1 to 1000 while h is 0; then h will change to 1 and j will again go through

each value from 1 to 1000. This process will continue until the code prints the words *h is: 11 and j is: 1000*. For example, if the code prints *h is: 3 and j is: 409*, you can assume the code will not be finished running for a while because it still must go through each j value for h values 4 through 11 and many of the j values for the h value of 3 it is currently on. The IPython console will print the word *Success* when the program has finished running. Do not close out of the Spyder program, make any changes to the text in the code, or open or move any of the input or output files while the code is running.

6. Return to step 3 and make lines 8 and 9 read: grade = 'choice' and num_obs_first_hedge_period = 38, respectively. Repeat steps 4-5. Remember that running this code with this cut and grade will take many hours.
7. Return to step 3 and make lines 8 and 9 read: grade = 'select' and num_obs_first_hedge_period = 35, respectively. Repeat steps 4-5. Remember that running this code with this cut and grade will take many hours.

Summary code. This is the last of the three Python code files involved in this project. This code must be run in order to solve for the optimal hedge ratio. Instructions on how to do so are listed below; however, a few additional details are first provided to understand the structure of the code's input and output files. All the input and output files are automatically set up to run through the program and the dynamic values are preset to match the assumptions outlined in the paper above. Therefore, the information in the following paragraph is just for your understanding and no action, besides what is listed in the instructions below, is required to run the code.

This code is called “summary.py” and is located in the Data\bootstrap\bootstrap_code_output\apply_strategy_output folder. Each time this code is run for each grade, this code uses 15 input files (i.e.- “cab_cashflow_h_0.0.csv,” “cab_cashflow_h_0.1.csv,” “cab_cashflow_h_0.2.csv,” “cab_cashflow_h_0.3.csv,” “cab_cashflow_h_0.4.csv,” “cab_cashflow_h_0.5.csv,” “cab_cashflow_h_0.6.csv,” “cab_cashflow_h_0.7.csv,” “cab_cashflow_h_0.8.csv,” “cab_cashflow_h_0.9.csv,” “cab_cashflow_h_1.0.csv,” “cab_cashflow_h_minvar.csv,” “cab_minvar.csv,” “cab_cashflow.csv,” and “cab_profit.csv”) which are all located in the Data\bootstrap\bootstrap_code_output\apply_strategy_output folder and seven output files (i.e.- “cab_cashflow_summary.csv,” “cab_profit_summary.csv,” “cab_minvar_summary.csv,” “cab_minvar_distribution.png,” “cab_mean_cashflows_and_profits.png,” “cab_std_cashflows_and_profits.png,” and “cab_profit_distribution.png”) which are all located in the Data\bootstrap\bootstrap_code_output\apply_strategy_output\summary _output folder. Values in the code that are dynamic and can be changed are $grade$, which refers to the grade of the bottom round flat meat cut the code will analyze; $sims$, which is the number of simulations; h_array, which is an array containing the various hedge ratios you wish to test and compare; $h\_array\_index\_to\_compare\_1$, which is the index number of the first of two hedge ratio values you wish to compare in the terminal cumulative profit distribution graph; and $h\_array\_index\_to\_compare\_2$, which is the index number of the second of two hedge ratio values you wish to compare in the terminal cumulative profit distribution graph.

Instructions. Follow the instructions below***:***

1. In the Spyder program, click “File.” Then click “Open.” Navigate to the Data\bootstrap\bootstrap_code_output\apply_strategy_output folder, click on the file “summary.py,” then click “Open.”
2. Prepare the code to run for the CAB grade by making line 7 of the code read: grade = ‘cab.’ Notice that the grade is spelled in all lowercase letters and has single quotation marks around the word.
3. Click the green triangle button at the top of the screen, which will run the program.
4. The IPython console, which may be located on the right side of the Spyder program’s screen, will print the word *Success* when the program has finished running. Do not close out of the Spyder program, make any changes to the text in the code, or open or move any of the input or output files while the code is running.
5. Observe the exported tables and graphs in the following example list: (i.e.- “cab_cashflow_summary.csv,” “cab_profit_summary.csv,” “cab_mean_cashflows_and_profits.png,” and “cab_std_cashflows_and_profits.png”) which are located in the Data\bootstrap\bootstrap_code_output\apply_strategy_output\summary _output folder. Of the hedge ratios 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0,9, 1.0, and the minimum variance hedge ratio (dynamically adjusted throughout the time period), determine which two hedge ratios you would like to compare in a terminal cumulative profit distribution graph. The terminal cumulative profit distribution graph (i.e.- “cab_profit_distribution.png”), which is located in the Data\bootstrap\bootstrap_code_output\apply_strategy_output\summary_output

folder, can also be used to help determine which two hedge ratios you would like to compare in another terminal cumulative profit distribution graph. See the "Compilation and interpretation of results" level-two subsection in the "Apply Hedging Strategies" subsection in the "Procedures and Results" section of the paper above for instructions on how to choose which hedge ratios to compare.

6. The values in h_array, the array holding the various hedge ratios you are testing, are 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, and the minimum variance hedge ratio (dynamically adjusted throughout the time period). Consider the index values of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11 that correspond to each of the values in h_array, respectively. Adjust the values of the variables $h\_array\_index\_to\_compare\_1$ and $h\_array\_index\_to\_compare\_2$ to be the index value associated with the two hedge ratios you chose to compare in the previous step. Adjust these two values by changing lines 9 and 10 of the code, respectively. For example, if you wish to compare the hedge ratios of 0.0 and 0.6, make lines 9 and 10 of the code read: h_array_index_to_compare_1 = 0 and h_array_index_to_compare_2 = 6, respectively.
7. Repeat steps 3-4. Then observe your customized terminal cumulative profit distribution graph (i.e.- "cab_profit_distribution.png"), which is located in the Data\bootstrap\bootstrap_code_output\apply_strategy_output\summary _output folder.
8. If you wish to see more terminal cumulative profit distribution graphs that compare another set of two hedge ratios, repeat steps 5-7 continually until you are done viewing and analyzing terminal cumulative profit distribution graphs. See the

"Interpretation of results and graphs" level-two subsection of this appendix for instructions on how to analyze this graph. Note that if you make more than one terminal cumulative profit distribution graph for the same grade, the most recently created terminal cumulative profit distribution graph will save over the previously saved one. To prevent this, change the filename of the previously created and exported terminal cumulative profit distribution graph so your new, most recently created terminal cumulative profit distribution graph can always be saved under the same name (i.e.- "cab_profit_distribution.png").

9. Return to step 2 and make line 7 read: grade = 'choice.' Repeat steps 3-8.
10. Return to step 2 and make line 7 read: grade = 'select.' Repeat steps 3-8.

Interpretation of results and graphs. See the "Compilation and interpretation of results" level-two subsection in the "Apply Hedging Strategies" subsection in the "Procedures and Results" section of the paper above for instructions on how to interpret the results and graphs for the Select, Choice, and CAB grades of the bottom round flat meat cut.

So the user of these code files can navigate the results in the output files from the "summary.py" code, I will specify which files correspond with the tables and figures in this paper. For example, the CAB grade "summary.py" output files include "cab_cashflow_summary.csv," "cab_profit_summary.csv," "cab_minvar_summary.csv," "cab_minvar_distribution.png," "cab_mean_cashflows_and_profits.png," "cab_std_cashflows_and_profits.png," and "cab_profit_distribution.png" and they correspond with Table 5 in Appendix U, Table 6 in Appendix V, part of Table 4 in Appendix T, Figure 15 in Appendix S, Figure 16 in Appendix W, Figure 17 in Appendix X, and Figure 18 in Appendix Y, respectively. For the Select and Choice grades, the fourth file in this list corresponds with Figure 13 in Appendix Q and Figure 14 in Appendix R, respectively, rather than Figure 15.

Formatting and other details differ between the files exported from the “summary.py” code and the tables and figures presented in the appendices of this paper, but they represent much of the same information. The biggest change is that Table 4 is made up of information from the “select_minvar_summary.csv,” “choice_minvar_summary.csv,” and “cab_minvar_summary.csv” files.

Appendix B

Figure 1. File Structure of Folders Provided to Lower Foods, Incorporated

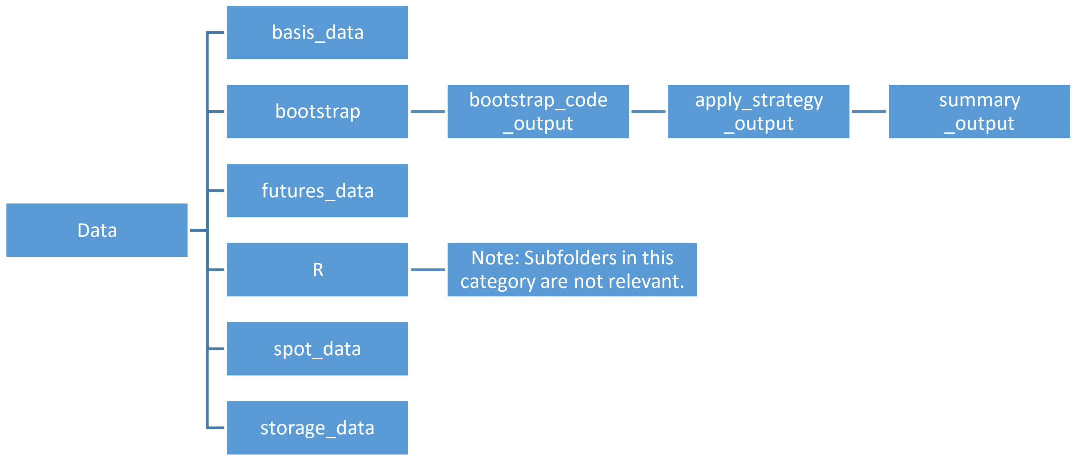

Figure 1. File structure of folders provided to Lower Foods, Incorporated. Each of the boxes represent a folder, which may also contain other files. The text in each box is the folder name, with the exception of the subfolders under the “R” folder, which are not pictured since they are not necessary to solve for the optimal hedge ratio.

Appendix C

Table 1. Pearson Correlation Coefficients Between the Live Cattle Electronic (LE) and Feeder Cattle Electronic (GF) Futures Contract Prices and the Spot Prices of the Select, Choice, and Certified Angus Beef (CAB) Grade of the Bottom Round Flat Meat Cut (January 1, 2014, through May 31, 2017)

Table 1

Pearson correlation coefficients between the Live Cattle electronic (LE) and Feeder Cattle electronic (GF) futures contract prices and the spot prices of the Select, Choice, and Certified Angus Beef (CAB) grade of the bottom round flat meat cut (January 1, 2014, through May 31, 2017)

Grade of spot prices	LE futures prices	GF futures prices
Select spot prices	0.7914	0.7641
Choice spot prices	0.7818	0.7559
CAB spot prices	0.7905	0.7611

Note. These correlation coefficients are calculated excluding all missing values from each individual dataset. For example, if a given date has a spot price listed for the Select grade but does not list the Live Cattle futures price for that date, all information for that date is excluded from that Pearson correlation coefficient calculation.

Appendix D

Table 2. T-Statistics of Augmented Dickey-Fuller (ADF) Unit Root Tests Run on the Level, Level First-Differenced, Log, and Log First-Differenced Live Cattle Electronic Futures Contract Prices and Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades Spot Prices (January 1, 2014, through May 31, 2017)

Table 2

T-Statistics of augmented Dickey-Fuller (ADF) unit root tests run on the level, level first-differenced, log, and log first-differenced Live Cattle electronic futures contract prices and bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades spot prices (January 1, 2014, through May 31, 2017)

Price series version	Futures	Select spot	Choice spot	CAB spot
Level	-1.352 (1)	-3.207 (5)**	-2.784 (9)*	-2.089 (2)
Level first-differenced	-20.356 (1)***	-10.549 (4)***	-10.516 (8)***	-9.145 (1)***
Log	-1.392 (1)	-3.058 (5)**	-2.627 (9)*	-2.065 (2)
Log first-differenced	-20.265 (1)***	-10.572 (4)***	-10.496 (8)***	-9.009 (1)***

Notes. The values shown in the table are the t-statistics on the coefficient θ in each estimated ADF regression equation

$$\boldsymbol{\Delta y_t = \alpha + \theta y_{t-1} + \gamma_1 \Delta y_{t-1} + \gamma_2 \Delta y_{t-2} + \cdots + \gamma_{10} \Delta y_{t-10} + e_t}$$

where $\boldsymbol{y_{t-1}}$ is the time $\boldsymbol{t-1}$ data value in the price series being tested; $\boldsymbol{\Delta y_{t-x}}$ is the change in the value of $\mathbf{y}$ from time $\boldsymbol{t-x-1}$ to time $\boldsymbol{t-x}$ where $\boldsymbol{x}$ is the lag number; $\boldsymbol{\alpha}$ is the intercept representing the drift term; $\boldsymbol{\theta}$ is the coefficient being estimated for the variable $\boldsymbol{y_{t-1}}$; $\boldsymbol{\gamma_x}$ is the coefficient being estimated for the variable $\boldsymbol{\Delta y_{t-x}}$ where $\boldsymbol{x}$ is, again, the lag number; $\boldsymbol{e_t}$ is the error term where $\boldsymbol{E(e_t | y_{t-1}, \Delta y_{t-1}, \Delta y_{t-2}, \cdots, \Delta y_{t-10}) = 0}$; and the number of lags included, up to a maximum of ten lags, is determined by Bayes Information Criteria (BIC).
The values in parenthesis represent the number of lags included in each ADF test.
The null hypothesis is that $\boldsymbol{\theta = 0}$ or, in other words, that there are unit roots in the price series being tested.

*significant at 10% significance level. **significant at 5% significance level. ***significance at 1% significance level.
These ADF tests are all one-tailed tests (Wooldridge, 2009, p. 631)

Appendix E

Table 3. T-Statistics of Engle-Granger Tests Including Augmented Dickey-Fuller (ADF) Unit Root Tests Run on Residuals of Regressions of Log Spot Prices of Various Bottom Round Flat Meat Grades on Live Cattle Electronic Log Futures Prices (January 1, 2014, through May 31, 2017)

Table 3

T-Statistics of Engle-Granger tests including augmented Dickey-Fuller (ADF) unit root tests run on residuals of regressions of log spot prices of various bottom round flat meat grades on Live Cattle electronic log futures prices (January 1, 2014, through May 31, 2017)

Residuals ADF test is run on	t-statistic
Residuals from Select grade log spot ~ log futures	-3.950 (1)***
Residuals from Choice grade log spot ~ log futures	-5.055 (5)***
Residuals from CAB grade log spot ~ log futures	-4.582 (1)***

Notes. The values shown in the table are the t-statistics on the coefficient θ in three estimated ADF regression equations as a part of Engle-Granger tests in which each equation reads

$$\boldsymbol{\Delta y_{t} = \alpha + \theta y_{t-1} + \gamma_1 \Delta y_{t-1} + \gamma_2 \Delta y_{t-2} + \cdots + \gamma_{10} \Delta y_{t-10} + e_{t}}$$

where $\boldsymbol{y_{t-1}}$ is the time $\boldsymbol{t-1}$ residual from the log spot price series being regressed on the log futures price series; $\boldsymbol{\Delta y_{t-x}}$ is the change in the value of $\boldsymbol{y}$ from time $\boldsymbol{t-x-1}$ to time $\boldsymbol{t-x}$ where $\boldsymbol{x}$ is the lag number; $\boldsymbol{\alpha}$ is the intercept representing the drift term; $\boldsymbol{\theta}$ is the coefficient being estimated for the variable $\boldsymbol{y_{t-1}}$; $\boldsymbol{\gamma_x}$ is the coefficient being estimated for the variable $\boldsymbol{\Delta y_{t-x}}$ where $\boldsymbol{x}$ is, again, the lag number; $\boldsymbol{e_t}$ is the error term where $\boldsymbol{E(e_t | y_{t-1}, \Delta y_{t-1}, \Delta y_{t-2}, \cdots, \Delta y_{t-10}) = 0}$; and the number of lags included, up to a maximum of ten lags, is determined by Bayes Information Criteria (BIC). In each of the three ADF regression equations, $\boldsymbol{y_{t-1}}$ represents the time $\boldsymbol{t-1}$ residual from a different spot price series being regressed on the futures price series.

Weekends and major holidays are excluded from the log futures and each log spot price series prior to running each regression and finding the residuals thereof.

The values in parenthesis represent the number of lags included in each ADF test.

The null hypothesis is that $\boldsymbol{\theta = 0}$ or, in other words, that there are unit roots in the residual series being tested.

To illustrate what the symbol ~ means, consider that the example of $\boldsymbol{y}$ ~ $\boldsymbol{x}$ means to regress $\boldsymbol{y}$ on $\boldsymbol{x}$.

*significant at 10% significance level. **significant at 5% significance level. ***significance at 1% significance level.

These ADF tests are all one-tailed tests (Wooldridge, 2009, p. 631)

Appendix F

Figure 2. Residuals from Regression of the Bottom Round Flat Select Grade Log Spot Prices on Live Cattle Electronic Log Futures Prices (January 1, 2014, through May 31, 2017).

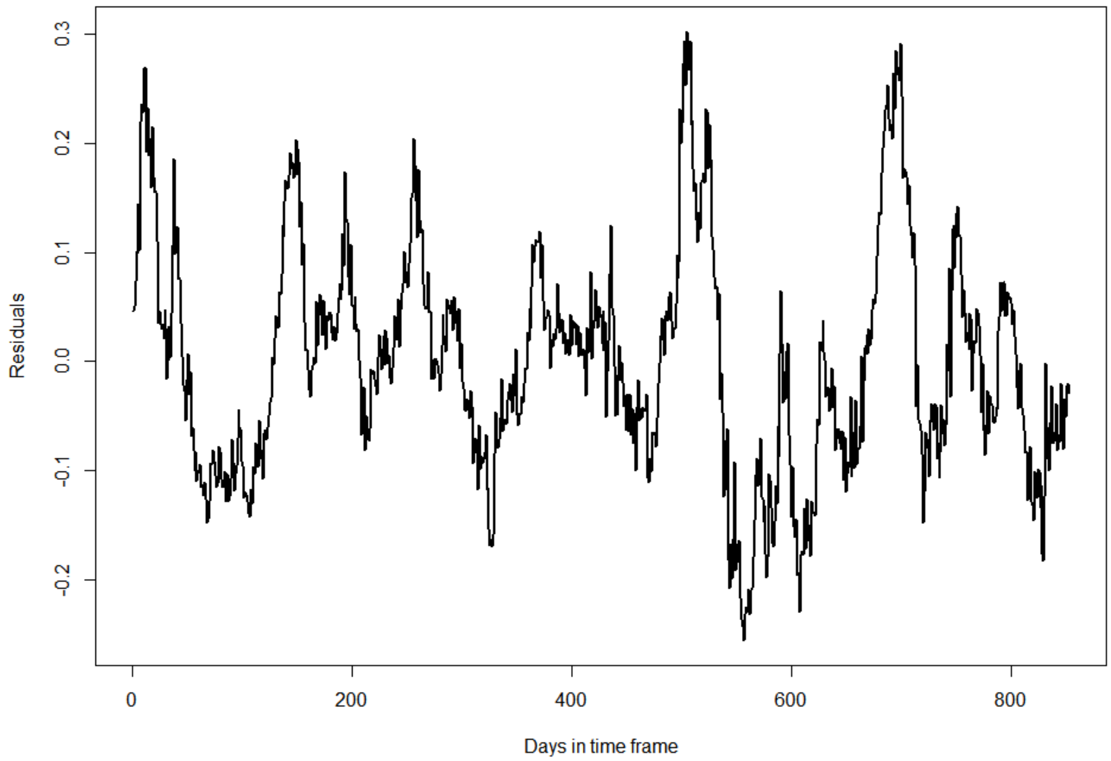

Figure 2. Residuals from regression of the bottom round flat Select grade log spot prices on Live Cattle electronic log futures prices (January 1, 2014, through May 31, 2017). The residuals appear to have a mean-reverting pattern that supports that the residuals are stationary. Weekends and major holidays are excluded from the log futures and log spot price series prior to running the regression. The residuals are reported in the natural log of dollars. This figure is created using an adapted version of Brough's (2017b) code and is adapted from the figure produced by his code.

Appendix G

Figure 3. Residuals from Regression of the Bottom Round Flat Choice Grade Log Spot Prices on Live Cattle Electronic Log Futures Prices (January 1, 2014, through May 31, 2017)

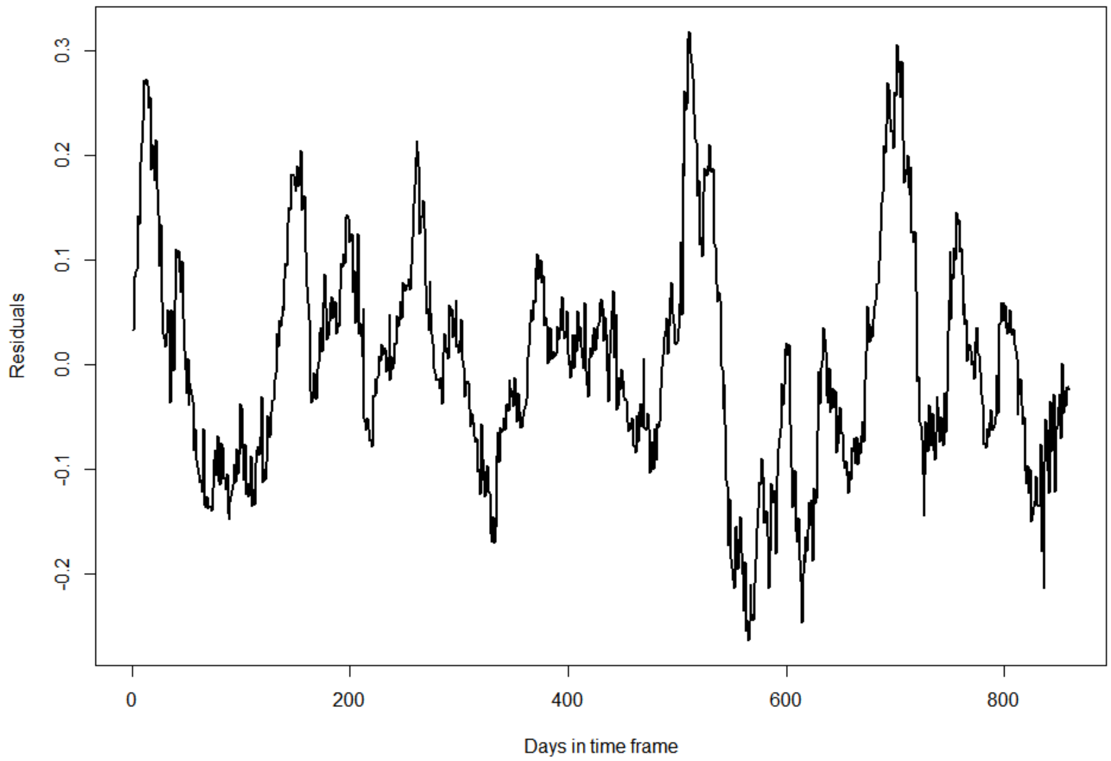

Figure 3. Residuals from regression of the bottom round flat Choice grade log spot prices on Live Cattle electronic log futures prices (January 1, 2014, through May 31, 2017). The residuals appear to have a mean-reverting pattern that supports that the residuals are stationary. Weekends and major holidays are excluded from the log futures and log spot price series prior to running the regression. The residuals are reported in the natural log of dollars. This figure is created using an adapted version of Brough's (2017b) code and is adapted from the figure produced by his code.

Appendix H

Figure 4. Residuals from Regression of the Bottom Round Flat Certified Angus Beef (CAB) Grade Log Spot Prices on Live Cattle Electronic Log Futures Prices (January 1, 2014, through May 31, 2017)

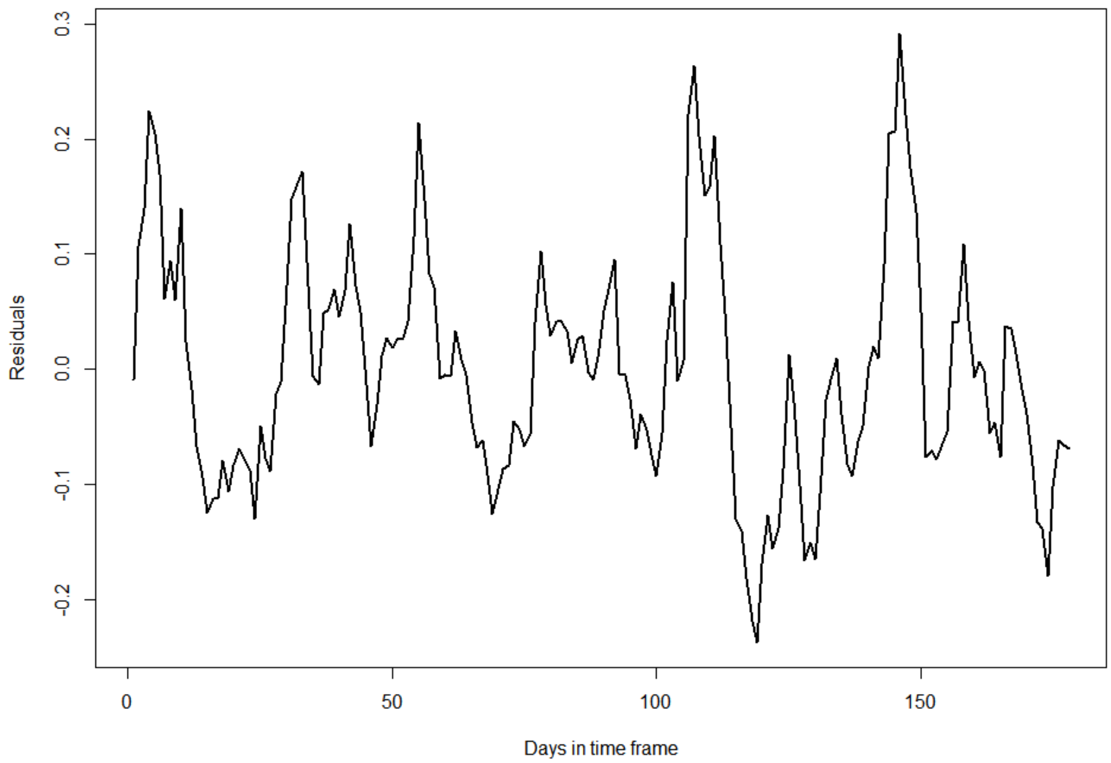

Figure 4. Residuals from regression of the bottom round flat Certified Angus Beef (CAB) grade log spot prices on Live Cattle electronic log futures prices (January 1, 2014, through May 31, 2017). The residuals appear to have a mean-reverting pattern that supports that the residuals are stationary. Weekends and major holidays are excluded from the log futures and log spot price series prior to running the regression. The residuals are reported in the natural log of dollars. This figure is created using an adapted version of Brough's (2017b) code and is adapted from the figure produced by his code.

Appendix I

Figure 5. Level Version of Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades Spot Prices and Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017)

Figure 5. Level version of prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades spot prices and Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the random walk appearance of the level futures and each level spot price series, suggesting the presence of unit roots in the level form of these price series. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix J

Figure 6. Level First-Differenced Version of Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades Spot Prices and Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017)

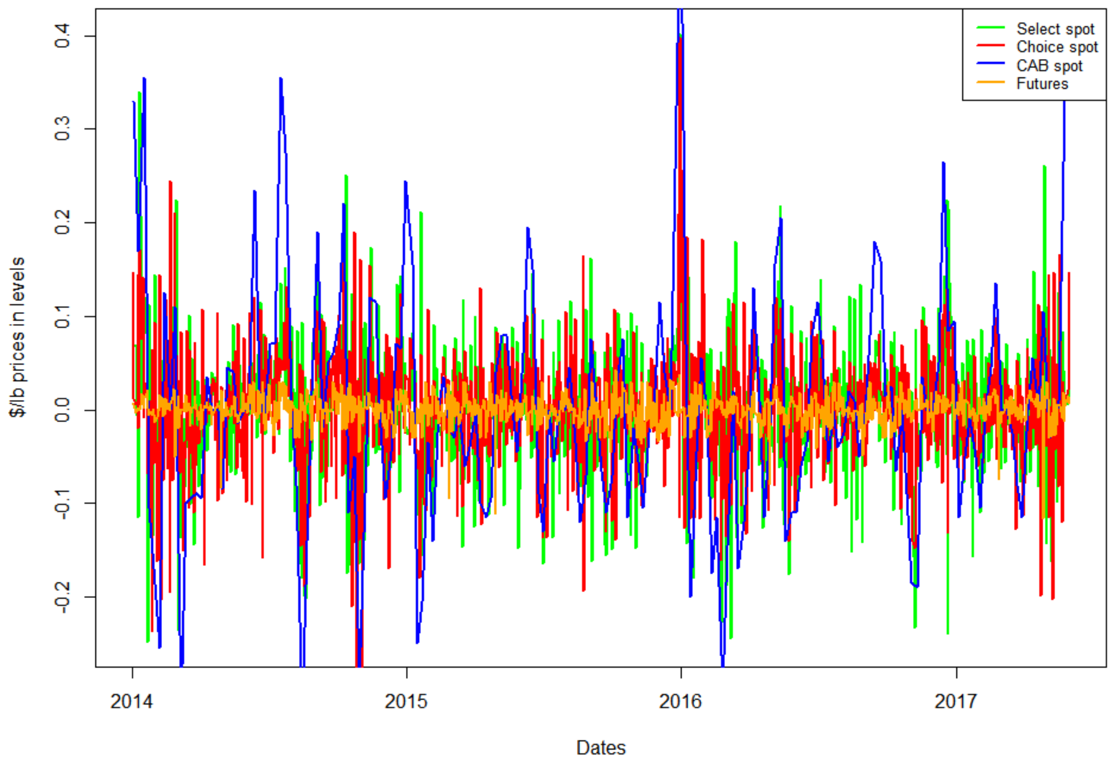

Figure 6. Level first-differenced version of prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades spot prices and Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the appearance of mean reversion around the value $0/lb in levels for the level first-differenced futures and each level first-differenced spot price series, suggesting the stationarity of each of these price series, meaning no unit roots are present therein. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix K

Figure 7. Log Version of Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades Spot Prices and Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017).

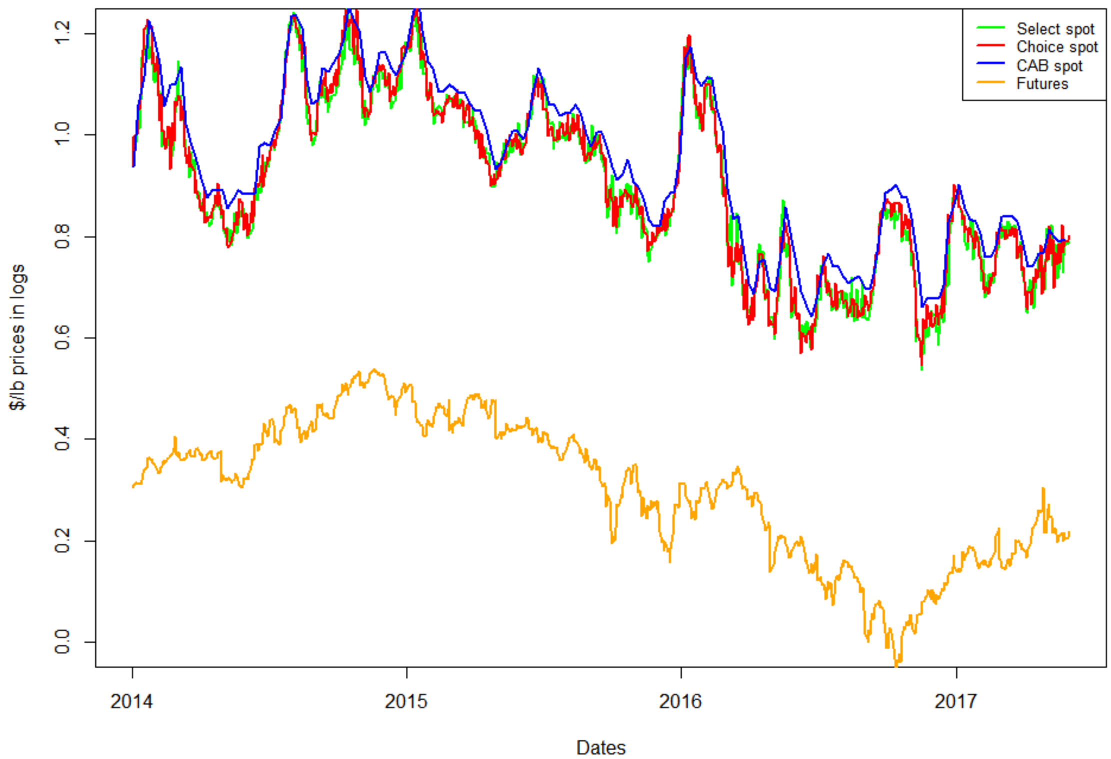

Figure 7. Log version of prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades spot prices and Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the random walk appearance of the log futures and each log spot price series, suggesting the presence of unit roots in the log form of these price series. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix L

Figure 8. Log First-Differenced Version of Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades Spot Prices and Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017)

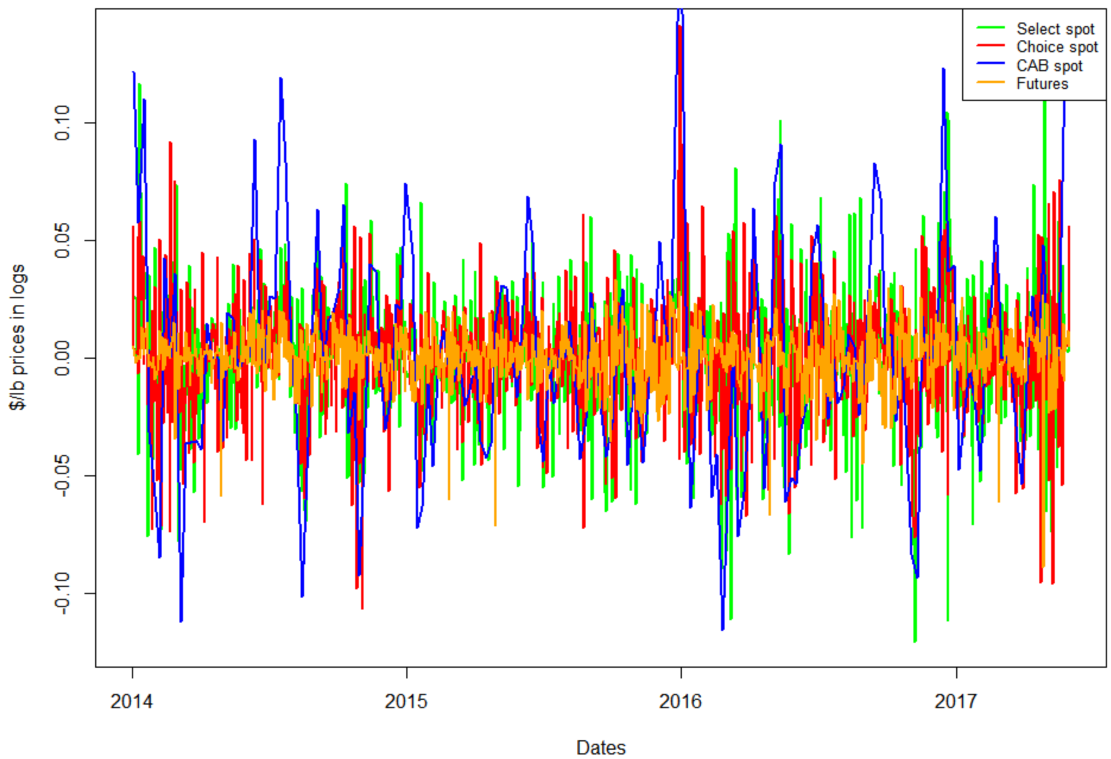

Figure 8. Log first-differenced version of prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades spot prices and Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the appearance of mean reversion around the value $0/lb in logs for the log first-differenced futures and each log first-differenced spot price series, suggesting the stationarity of each of these price series, meaning no unit roots are present therein. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix M

Figure 9. Level Version of the Basis Between the Spot Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades and the Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017)

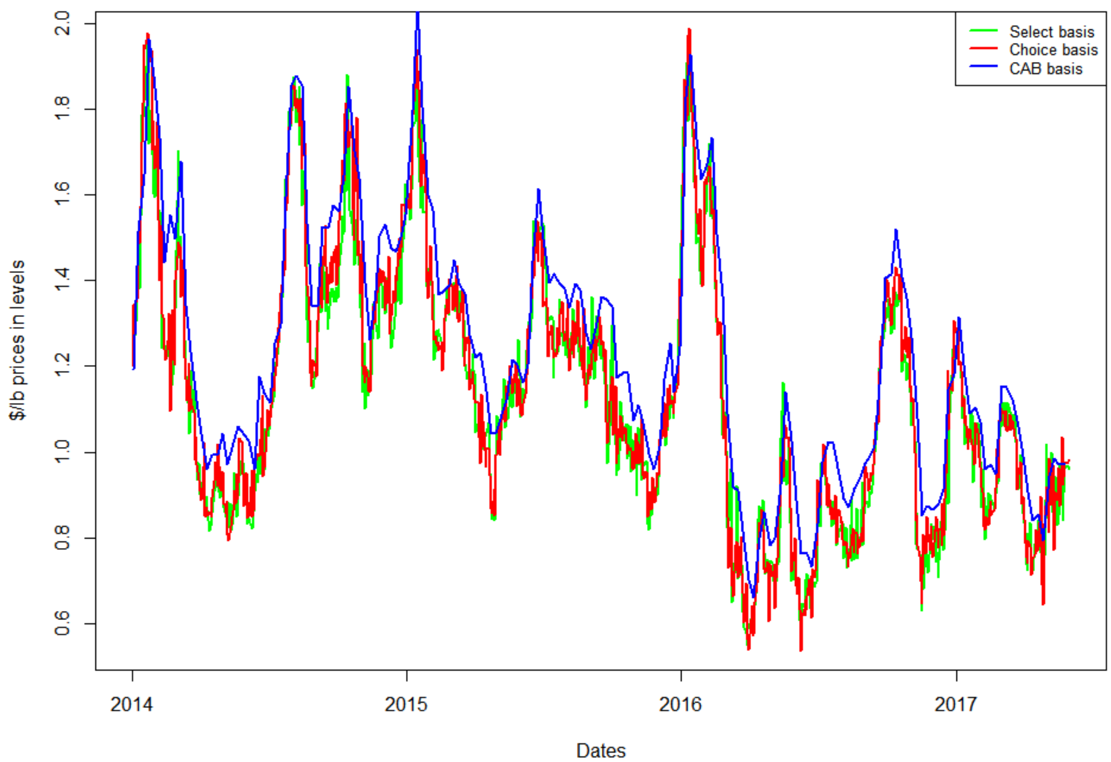

Figure 9. Level version of the basis between the spot prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades and the Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the random walk appearance of each level basis price series, suggesting the presence of unit roots in the level form of these price series. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix N

Figure 10. Level First-Differenced Version of the Basis Between the Spot Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades and the Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017)

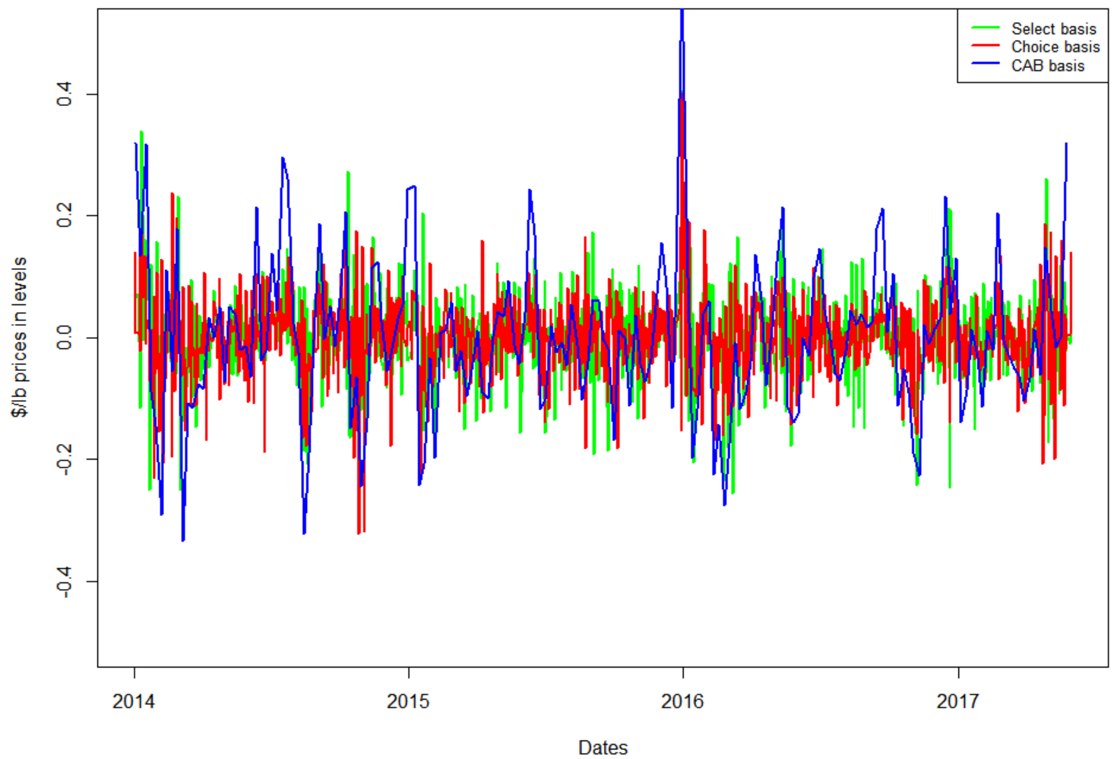

Figure 10. Level first-differenced version of the basis between the spot prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades and the Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the appearance of mean reversion around the value $0/lb in levels for each level first-differenced basis price series, suggesting the stationarity of each of these price series, meaning no unit roots are present therein. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix O

Figure 11. Log version of the Basis Between the Spot Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades and the Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017)

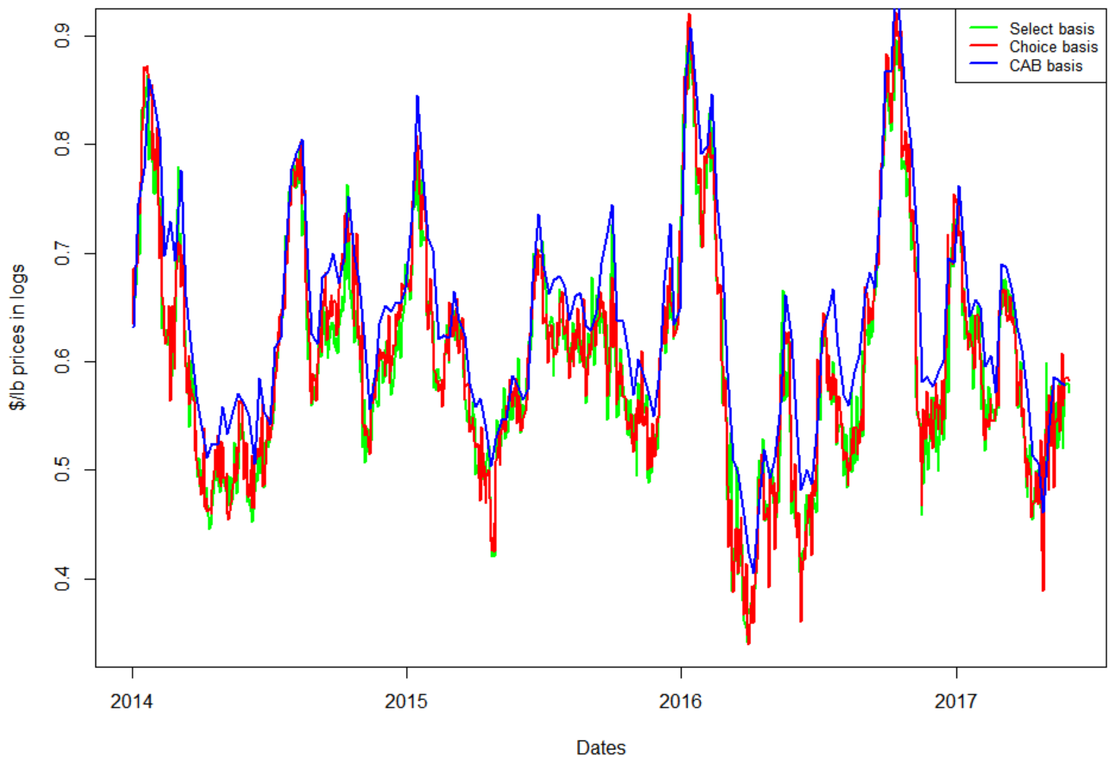

Figure 11. Log version of the basis between the spot prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades and the Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the random walk appearance of each log basis price series, suggesting the presence of unit roots in the log form of these price series. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix P

Figure 12. Log First-Differenced Version of the Basis Between the Spot Prices of the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades and the Live Cattle Electronic Futures Prices (January 1, 2014, through May 31, 2017).

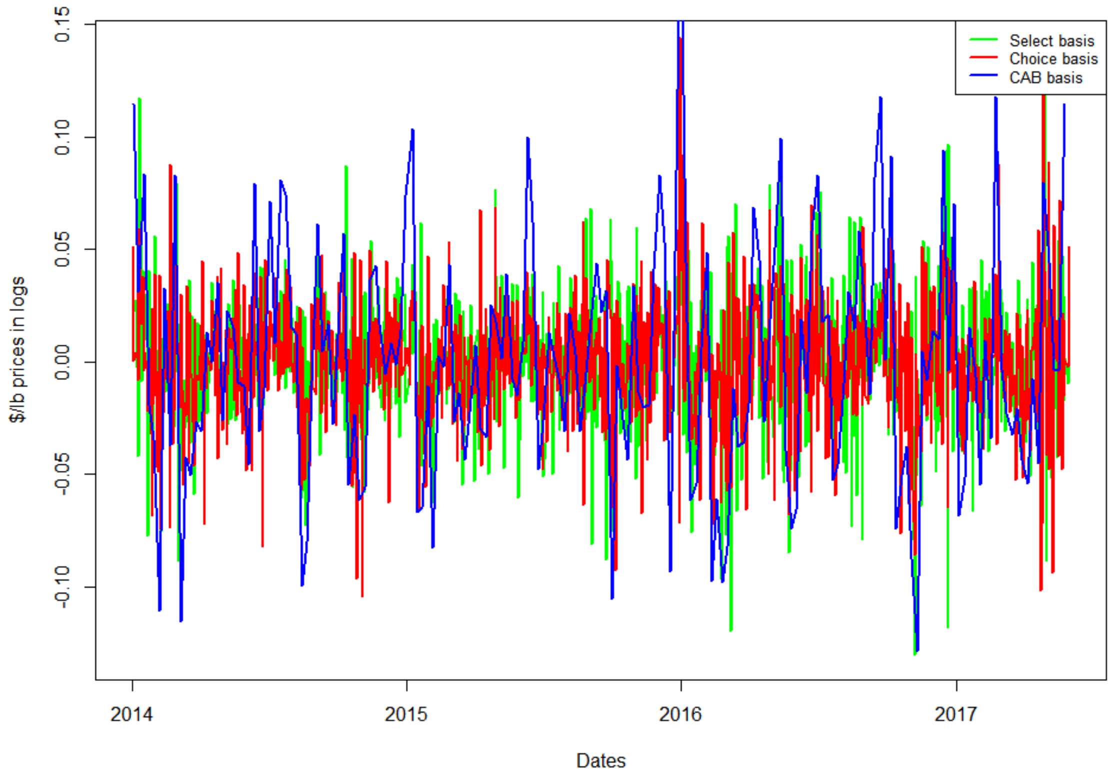

Figure 12. Log first-differenced version of the basis between the spot prices of the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades and the Live Cattle electronic futures prices (January 1, 2014, through May 31, 2017). In this figure, $/lb represents dollars per pound. The graph shows the appearance of mean reversion around the value $0/lb in logs for each log first-differenced basis price series, suggesting the stationarity of each of these price series, meaning no unit roots are present therein. This figure is created using an adapted version of Brough's (2017a) code, an adapted version of Ferrin's (2017) code, and possibly a classroom demonstration given by Tyler Brough in the Spring of 2017; this figure is adapted from figures produced by these code files.

Appendix Q

Figure 13. Minimum Variance Hedge Ratio Distribution of the Bottom Round Flat Select Grade

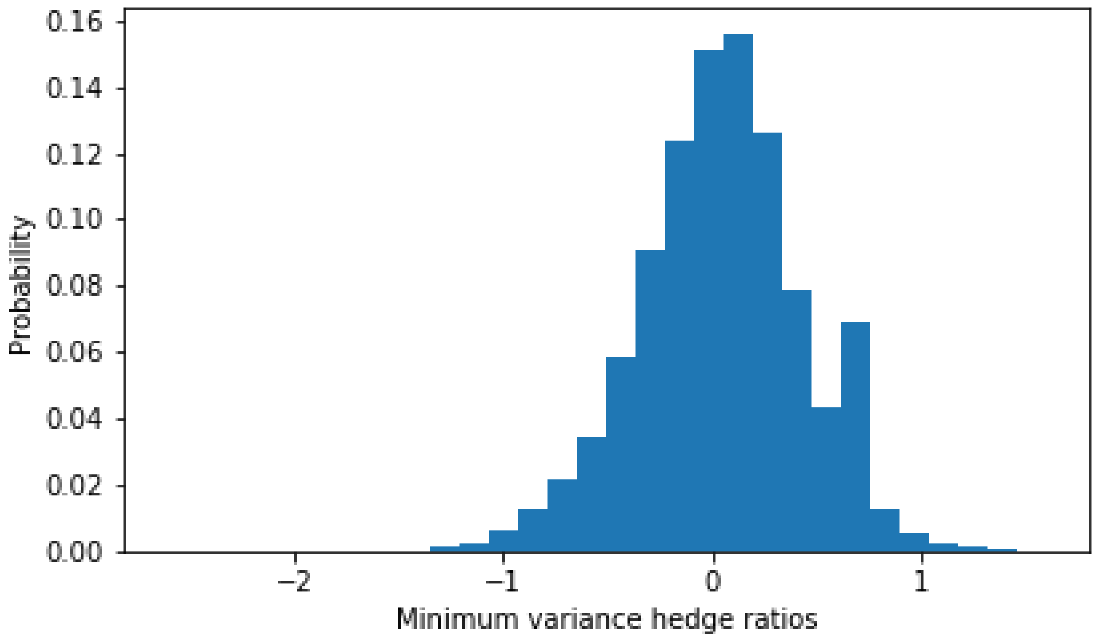

Figure 13. Minimum variance hedge ratio distribution of the bottom round flat Select grade. This plot shows the distribution of all the minimum variance hedge ratios calculated across the time period from January 1, 2014, through May 31, 2017, and across 1,000 simulations for the Select grade of the bottom round flat meat cut. For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months.

Appendix R

Figure 14. Minimum Variance Hedge Ratio Distribution of the Bottom Round Flat Choice Grade

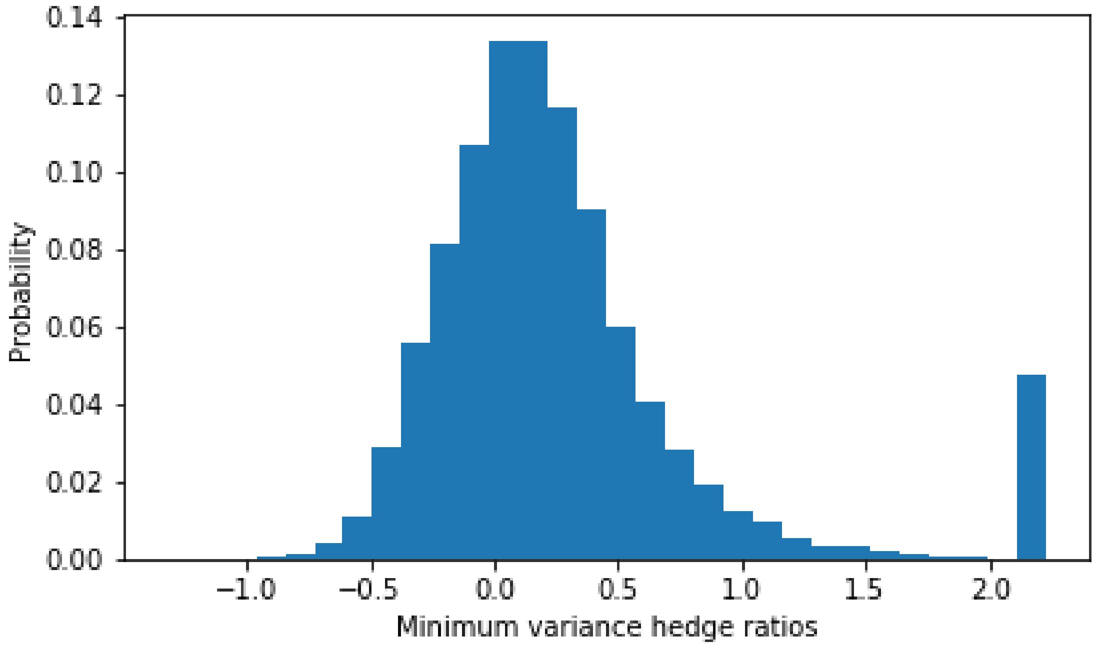

Figure 14. Minimum variance hedge ratio distribution of the bottom round flat Choice grade. This plot shows the distribution of all the minimum variance hedge ratios calculated across the time period from January 1, 2014, through May 31, 2017, and across 1,000 simulations for the Choice grade of the bottom round flat meat cut. For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months.

Appendix S

Figure 15. Minimum Variance Hedge Ratio Distribution of the Bottom Round Flat Certified Angus Beef (CAB) Grade

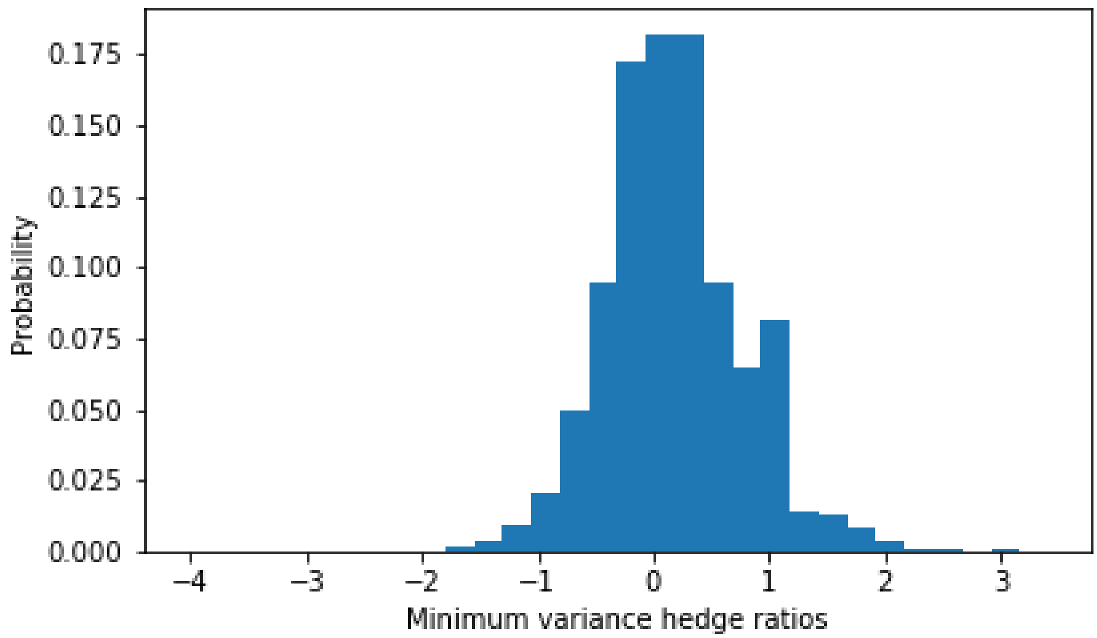

Figure 15. Minimum variance hedge ratio distribution of the bottom round flat Certified Angus Beef (CAB) grade. This plot shows the distribution of all the minimum variance hedge ratios calculated across the time period from January 1, 2014, through May 31, 2017, and across 1,000 simulations for the CAB grade of the bottom round flat meat cut. For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months.

Appendix T

Table 4. Summary Statistics of Minimum Variance Hedge Ratios Calculated from January 1, 2014, through May 31, 2017, for 1,000 Simulations for the Bottom Round Flat Select, Choice, and Certified Angus Beef (CAB) Grades.

Table 4

Summary statistics of minimum variance hedge ratios calculated from January 1, 2014, through May 31, 2017, for 1,000 simulations for the bottom round flat Select, Choice, and Certified Angus Beef (CAB) grades.

Grade	Mean	Standard deviation
Select	0.0445	0.3934
Choice	0.2751	0.5618
CAB	0.1786	0.5996

Note. For these grades of this meat cut in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months.

Appendix U

Table 5. Summary Statistics of the Bottom Round Flat Certified Angus Beef (CAB) Grade Terminal Cumulative Cashflows and Daily Cumulative Cashflows (January 1, 2014, through May 31, 2017)

Summary statistics of the bottom round flat Certified Angus Beef (CAB) grade terminal cumulative cashflows and daily cumulative cashflows (January 1, 2014, through May 31, 2017)

h_value	term_mean	term_min	term_max	term_std	daily_mean	daily_min	daily_max	daily_std
0.0	(437,632)	(437,632)	(437,632)	0	(149,553)	(542,867)	66,042	238,774
0.1	(440,284)	(551,563)	(402,687)	19,979	(157,581)	(551,686)	59,797	239,331
0.2	(442,935)	(665,493)	(367,741)	39,959	(165,609)	(561,016)	56,972	240,048
0.3	(445,586)	(779,423)	(332,795)	59,938	(173,638)	(571,121)	56,538	240,918
0.4	(448,237)	(893,353)	(297,850)	79,917	(181,666)	(581,936)	57,297	241,939
0.5	(450,888)	(1,007,284	(262,904)	99,897	(189,694)	(593,471)	58,764	243,109
0.6	(453,539)	(1,121,214	(227,958)	119,876	(197,723)	(605,700)	60,621	244,426
0.7	(456,191)	(1,235,144	(193,013)	139,855	(205,751)	(618,488)	62,842	245,890
0.8	(458,842)	(1,349,074	(158,067)	159,835	(213,779)	(631,746)	65,310	247,502
0.9	(461,493)	(1,463,004	(123,122)	179,814	(221,808)	(645,366)	67,967	249,262
1.0	(464,144)	(1,576,935	(88,176)	199,794	(229,836)	(659,303)	70,799	251,174
Minimum variance[a]	(435,983)	(814,529)	(79,412)	111,307	(180,411)	(596,032)	76,381	240,277

Notes. This table shows terminal cumulative cashflow and daily cumulative cashflow values for 1,000 simulations. In this table, ***h_value*** represents the hedge ratio being tested; ***term_mean***, ***term_min***, ***term_max***, and ***term_std*** represent the terminal cumulative cashflow's mean, minimum, maximum, and standard deviation across all simulations, respectively; and ***daily_mean***, ***daily_min***, ***daily_max***, and ***daily_std*** represent the average daily cumulative cashflow's mean, minimum, maximum, and standard deviation within each entire time period averaged across all simulations.

Values in parentheses represent negative numbers.

All mean, minimum, maximum, and standard deviation values are reported in dollars.

[a]For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which was about every two months.

Appendix V

Table 6. Summary Statistics of the Bottom Round Flat Certified Angus Beef (CAB) Grade Terminal Cumulative Profits (January 1, 2014, through May 31, 2017)

Summary statistics of the bottom round flat Certified Angus Beef (CAB) grade terminal cumulative profits (January 1, 2014, through May 31, 2017)

h_value	term_mean	term_min	term_max	term_std
0.0	(376,317)	(1,343,239)	5,118,505	723,836
0.1	(378,968)	(1,347,101)	5,125,760	716,901
0.2	(381,619)	(1,350,963)	5,133,015	710,461
0.3	(384,271)	(1,354,824)	5,140,271	704,529
0.4	(386,922)	(1,358,686)	5,147,526	699,117
0.5	(389,573)	(1,362,548)	5,154,781	694,239
0.6	(392,224)	(1,378,489)	5,162,037	689,905
0.7	(394,875)	(1,473,601)	5,169,292	686,125
0.8	(397,526)	(1,568,713)	5,176,547	682,909
0.9	(400,178)	(1,663,824)	5,183,803	680,266
1.0	(402,829)	(1,758,936)	5,191,058	678,200
Minimum variance[a]	(374,668)	(1,365,607)	5,201,267	716,366

Notes. This table shows terminal cumulative profit values for 1,000 simulations.
In this table, ***h_value*** represents the hedge ratio being tested; ***term_mean***, ***term_min***, ***term_max***, and ***term_std*** represent the terminal cumulative profit's mean, minimum, maximum, and standard deviation across all simulations, respectively.
Values in parentheses represent negative numbers.
All mean, minimum, maximum, and standard deviation values are reported in dollars.

[a]For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months.

Appendix W

Figure 16. Bottom Round Flat Certified Angus Beef (CAB) Grade Mean Terminal Cumulative Cashflows and Mean Terminal Cumulative Profits Across Various Hedge Ratios

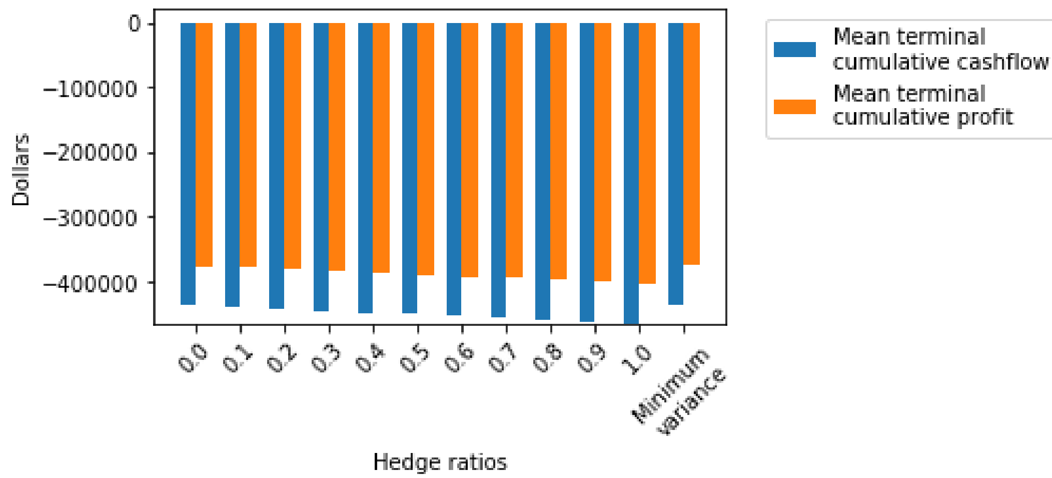

Figure 16. Bottom round flat Certified Angus Beef (CAB) grade mean terminal cumulative cashflows and mean terminal cumulative profits across various hedge ratios. Each mean value represents a summary statistic of observations of the terminal cumulative cashflows and terminal cumulative profits from 1,000 simulations of the time period from January 1, 2014, through May 31, 2017. For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months. This figure can help determine which hedge ratios appear to be optimal.

Appendix X

Figure 17. Bottom Round Flat Certified Angus Beef (CAB) Grade Terminal Cumulative Cashflow Standard Deviations and Terminal Cumulative Profit Standard Deviations Across Various Hedge Ratios

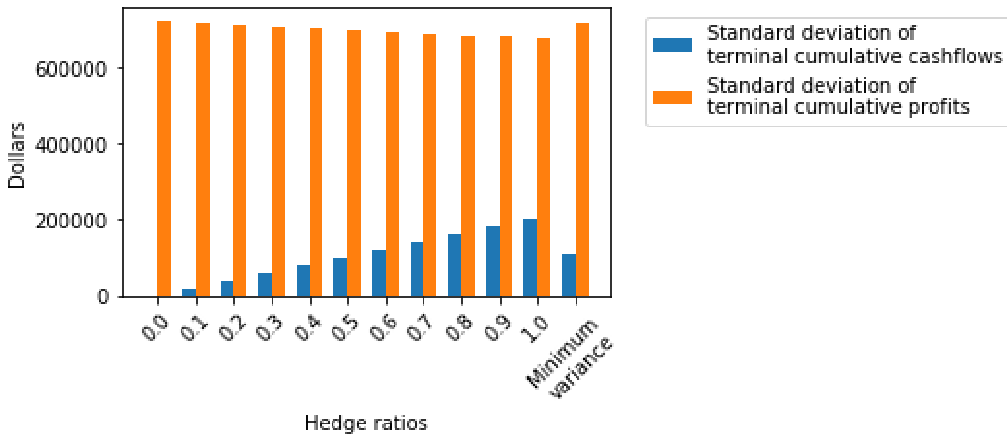

Figure 17. Bottom round flat Certified Angus Beef (CAB) grade terminal cumulative cashflow standard deviations and terminal cumulative profit standard deviations across various hedge ratios. Each standard deviation represents a summary statistic of observations of the terminal cumulative cashflows and terminal cumulative profits from 1,000 simulations of the time period from January 1, 2014, through May 31, 2017. For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months. This figure can help determine which hedge ratios appear to be optimal.

Appendix Y

Figure 18. Bottom Round Flat Certified Angus Beef (CAB) Grade Terminal Cumulative Profit Distribution Comparison for Two Hedge Ratios

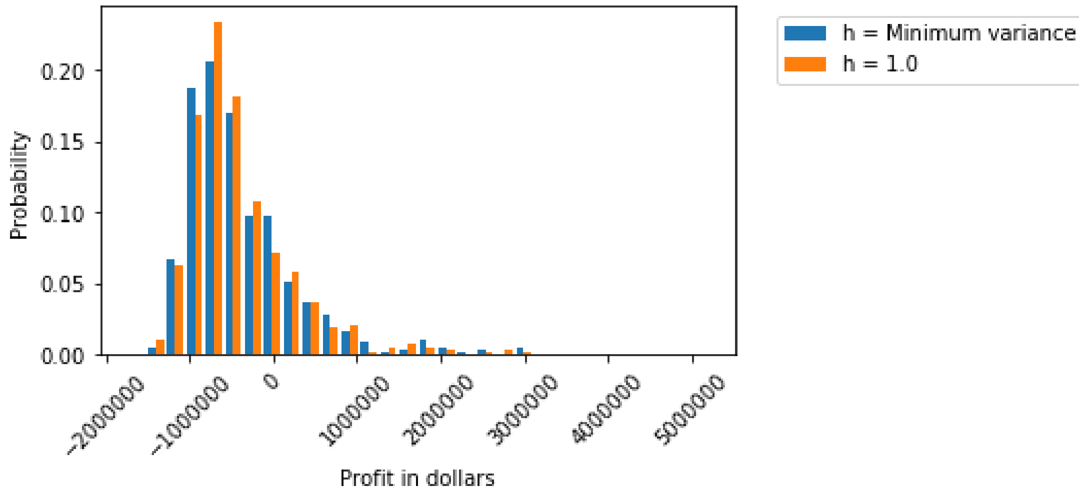

Figure 18. Bottom round flat Certified Angus Beef (CAB) grade terminal cumulative profit distribution comparison for two hedge ratios. In the legend above, h represents hedge ratio. This graph can be used to show a terminal cumulative profit distribution comparison for two hedge ratios. The distribution for each hedge ratio reflects observations of the terminal cumulative profits from 1,000 simulations of the time period from January 1, 2014, through May 31, 2017. For this meat cut and grade in this project, the minimum variance hedge ratio is calculated at the beginning of each time period and then recalculated each time there is a new front-month Live Cattle electronic futures contract, which is about every two months. This figure can help determine which hedge ratios appear to be optimal.