

1-1-2017

## Hedging the Price Risk of Crop Revenue Insurance through the Options Market

Sweta Tiwari

Follow this and additional works at: <https://scholarsjunction.msstate.edu/td>

---

### Recommended Citation

Tiwari, Sweta, "Hedging the Price Risk of Crop Revenue Insurance through the Options Market" (2017).  
*Theses and Dissertations*. 2478.  
<https://scholarsjunction.msstate.edu/td/2478>

This Graduate Thesis - Open Access is brought to you for free and open access by the Theses and Dissertations at Scholars Junction. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholars Junction. For more information, please contact [scholcomm@msstate.libanswers.com](mailto:scholcomm@msstate.libanswers.com).

Hedging the price risk of crop revenue insurance through the options market

By

Sweta Tiwari

A Thesis  
Submitted to the Faculty of  
Mississippi State University  
in Partial Fulfillment of the Requirements  
for the Degree of Master of Science  
in Agriculture Economics  
in the Department of Agriculture Economics

Mississippi State, Mississippi

August 2017

Copyright by

Sweta Tiwari

2017

Hedging the price risk of crop revenue insurance through the options market

By

Sweta Tiwari

Approved:

---

Keith H. Coble  
(Major Professor)

---

Shrinidhi Ambinakudige  
""(O k̄pqt'Rtqhḡuqt)

---

Ardian Harri  
(Eqo o kwgg'O go dgt lGraduate Coordinator)

---

Barry J. Barnett  
(Committee Member)

---

George M. Hopper  
Dean  
College of Agriculture and Life Sciences

Name: Sweta Tiwari

Date of Degree: August 11, 2017

Institution: Mississippi State University

Major Field: Agriculture Economics

Major Professor: Dr. Keith H. Coble

Title of Study: Hedging the price risk of crop revenue insurance through the options market

Pages in Study 119

Candidate for Degree of Master of Science

Crop revenue insurance is an exception in the insurance industry offering a guarantee subsuming a highly systematic risk- price variability. This study examines whether crop insurance companies could use put and call options to hedge the price risk present in corn revenue insurance. The behavioral model used to examine hedging optimization behavior of a crop producer with crop insurance by Coble, Heifner, and Zuniga (2002) is modified to examine optimal hedge ratio of a company selling revenue insurance. The crop insurance summary of business from 1985-2015 for corn revenue policies was simulated. Corn futures prices were collected from the Commodity Research Bureau databases. Results show that net return from call and put options can hedge indemnities paid by corn RP and RP-HPE resulting from the price variability in some scenario. This suggests hedging the price risk of corn revenue insurance through options could be a viable practice for crop insurers.

## DEDICATION

I would like to dedicate this thesis to the loving memory of my father, Mr. Kumar Prasad Tiwari.

## ACKNOWLEDGEMENTS

I would like to convey my sincere gratitude to my major advisor, Professor Keith H. Coble for his continuous guidance and expertise throughout my research and thesis writing. He has handed over full authority in my subject matter and always directed me in the right direction whenever he felt I have deviated from my goal.

I would also like to thank my committee members, Professor Ardian Harri, and Professor Barry J. Barnett for their inputs, and feedbacks. Without their insightful comments, this thesis would not have been completed.

I would even like to appreciate Professor Shrinidhi Ambinakudige for his motivation to study the minor degree in the Geo-spatial and Remote sensing.

Finally, I would like to acknowledge my spouse Umesh Silwal, and my brother in law Sushil Silwal, for their continuous support and appreciation during my schooling.  
Thank You.

## TABLE OF CONTENTS

DEDICATION .....	ii
ACKNOWLEDGEMENTS .....	iii
LIST OF TABLES .....	vi
CHAPTER	
I. INTRODUCTION .....	1
II. LITERATURE REVIEW .....	6
Reasons for Crop Insurer to Invest in the Option Market .....	6
Increase in the Demand for Crop Revenue Insurance .....	6
Crop Reinsurance Industry .....	7
Efficiency of the Options Market to Hedge the Systematic Risks .....	9
Comparing Methods of Computing Optimal Hedge Ratios .....	11
Minimum Variance Hedge Ratio.....	11
Mean-Variance Hedge .....	13
Expected Utility Framework .....	13
III. CONCEPTUAL FRAMEWORK .....	15
Expected Utility from the Crop Insurer Perspective .....	15
IV. MODEL AND DATA.....	19
Indemnity Simulation .....	19
Calculating Representative Farm Yields .....	19
Futures Prices Calculation .....	20
Calculating Average Corn Coverage Level.....	21
Premium Simulation.....	21
Liability Simulation.....	22
Options Payout .....	22
Expected Utility Simulation .....	23
V. RESULTS .....	25



Descriptive Statistics of End-Of-Period Wealth from Different Alternatives.....	25
Optimal Hedge Ratio Under Different Scenarios.....	27
Evaluation of Options Contract Effectiveness .....	32
Results from Put Options in the Study Region Including Corn Belt.....	32
Results from Put Options in the Study Region Excluding Major Corn Belt .....	34
Results from Call Options in the Study Region Including Major Corn Belt .....	35
Results from Call Options in the Study Region Excluding the Corn Belt .....	36
VI. CONCLUSION.....	38
REFERENCES .....	41
APPENDIX	
A. PROGRAMMING CODES AND TABLES .....	45
Tables Showing Results When Degree of Risk Aversions Equal to One and Three .....	46
Programming Codes .....	48
Python Codes.....	48
SAS CODES.....	50
Put options Including Corn Belts Hedging the Price Risk of Corn RP .....	50
Put options Including Corn belts hedging the price risk of RP-HPE .....	67
Call Options Including Corn Belts Hedging the Price Risk of Corn RP .....	84
Call Options Including Corn Belts Hedging the Price Risk of Corn RP-HPE .....	102

## LIST OF TABLES

5.1	Descriptive statistics of end-of-period wealth under different alternatives scenarios .....	26
5.2	Optimal Hedge Ratios Under Different Sceanrios.....	30
5.3	Pearson Correlation Coefficient Between Return from Put Options Return and Indemnity Paid from Corn RP .....	33
5.4	Pearson Correlation Coefficients between Put Options Return and Indemnity Paid from Corn RP-HPE .....	34
5.5	Pearson Correlation Coefficients between Put Options Return and Indemnity Paid from Corn RP .....	34
5.6	Pearson Correlation Coefficients between Put Options Return and Indemnity from for Corn RP-HPE .....	34
5.7	Pearson Correlation Coefficients Including Corn Belt between Call Options Return and Indemnity paid by RP .....	36
5.8	Pearson Correlation Coefficients Including Corn Belt between Call Options Return and Indmenity paid by RP-HPE .....	36
5.9	Pearson Correlation Coefficients Excluding Corn Belt between Call Options Return and RP Indemnity .....	37
5.10	Pearson Correlation Coefficients Excluding Corn Belt between Call Options Return and RP-HPE Indemnity .....	37
A.1	Optimal hedge ratios under different scenarios. ....	46
A.2	Optimal hedge ratios under different scenarios .....	47

## CHAPTER I

### INTRODUCTION

Insurance companies protect insured individuals against losses that result from risky or uncertain situations in exchange for a premium. The purchase of an insurance policy does not mean the risks are completely eliminated but in fact, the risks are transferred from the insured to the insurer. Insurance companies manage their risks in part by pooling a large number of independent exposure units. Through risk pooling, a policyholder shares his/her economic burden resulting from the uncertain situation, that could be damaging to him/ her, with a large number of people insured against similar events but who do not suffer losses at the same time. Pooling of a larger number of homogeneous and independent risk units is one of the essential conditions for a risk to be insurable (Rejda 1995). Correlated catastrophic events are uninsurable as losses are strongly positively correlated among all individuals at the same time thereby resulting in huge losses to the insurance company. In other words, the insurance companies typically do not insure systematic risks. However, crop revenue insurance is one of the exceptions in the insurance industry in that the guarantee subsumes a highly systematic risk- price variability.

In the U.S., crop revenue insurance is designed, rated, and underwritten by the USDA Risk Management Agency. However, the policies are delivered by the private firms who share in the risk exposure of the policies they sell. Crop revenue insurance

indemnifies the deficit in the farmer's gross revenue which results from either low yield or low price or a combination of the two. The USDA revenue protection plan uses the change in commodity futures prices from a time period before planting to approximately the harvest month for the price risk component.

Commodity futures prices and Actual Production History (APH) are used to calculate revenue guarantee for revenue protection policies. Projected and harvest prices for a corn revenue protection policy are determined by calculating monthly average of future prices of December future contract during February and October. The APH is an average of a crop producer's yield over multi periods which generally include minimum of four years and maximum of ten years. Crop revenue protection policies (RP) provide a guarantee based on the higher of projected futures price or harvest price. Revenue policies which do not cover harvest price fall are known as Revenue Protection with Harvest Price Exclusion (RP-HPE). Thus, farmers buying RP-HPE policy pay a lower premium as compared to Revenue Protection (RP) policy.

If the indemnities are triggered by either low projected or low harvest prices, then the crop insurance company must compensate all its policyholders holding the revenue protection policies at the same time. In such a situation, the company might be unable to meet its liabilities towards all of its clients and the company may be declared bankrupt.

The failure of an insurance company potentially costs its policyholders, stakeholders, shareholders, the insurance industry, and the economy as a whole. Massey et al. (2000) have highlighted the following reasons to care about the failure of an insurance company:

- At the time of failure, if the policyholders have an outstanding claim then, there are chances that it will not be paid or paid fully. The unexpired premium may not be paid back to the policyholders. However, in the case of federal crop insurance policies, if the policyholders have an outstanding claim then the federal government would pay the claim.
- Each and every shareholder has invested in the company with the hope of earning a return in the form of a future dividend.
- The public lose their faith in the insurance company. Fewer people having greater exposure to risk buy the insurance and risk pooling becomes more difficult for the insurance companies.

There is always a certain amount of volatility involved in underwriting and investment returns which affect the operation and profitability of an insurance company. Thus, it is important to understand how insurance companies hedge their risks. According to Luna (2012) most insurance companies are actively participating in the derivative market to hedge their underwriting and investment risks rather than speculating (Luna, 2012). Underwriting risk such as catastrophic risk, inflation risk, and actuarial risk can be hedged through futures, options, interest rate derivative, currency derivative and so on. Further, investment risks such as a decrease in the price of stocks and increase in interest rate of bonds can be hedged through options, futures, and swaps. Besides hedging underwriting and investment risks, insurers could use derivatives to hedge against a decrease in fixed-income assets, to reduce bankruptcy cost, liquidity risk and tax obligation, to match asset-liability duration better, to lock in the current yield for anticipated investment, to reduce expected cost of financial distress and to increase

welfare of shareholders or managers (Bouzouita and Young 1998; Colquitt and Hoyt 1997; Cummins, Phillips and Smith 2001).

In context of crop insurance industry, until now no detailed study has been conducted to understand the use of the derivative instruments by the crop insurance providers to hedge their risk exposure. While the U.S. crop insurance companies are partially protected from the insured risk by the standard reinsurance agreement, the companies still retain a significant amount of the risk. In order to, manage their risks, the crop insurance companies can use a wide variety of the derivative instruments including reinsurance offered by private reinsurers.

The derivative instruments such as put option and call option are traded on organized exchanges known as options markets. A put option is standardized contract providing its owner the right to sell and its writer the obligation to buy a futures contract on a particular commodity at a specified price (strike price), with the right lasting until the maturity date of the contract (Kolb 1996). The buyer will exercise put options based on the assumptions that market price of the underlying futures contract falls below the strike price providing protection for decrease in price. On the contrary, the investor may choose to buy call options when the buyer believes that the price of the underlying futures contract is going to rise above the strike price ensuring a certain return from increase in price. A call option contract provides the buyer the right to buy and the writer the obligation to sell a futures contract on a particular commodity at a strike price until the maturity date of the contract (Kolb 1996). Buyers of both the put and call options can choose not to exercise their rights depending on the hedging and arbitrage opportunities available to them. Hedging opportunity reduces exposure towards the price risk and

trader taking the hedge position is known as a hedger. Through the arbitrage opportunity, trader, typically known as speculators, can earn a profit when price moves up and down. Speculators provide market liquidity allowing hedgers to enter and exit the market efficiently (Chicago Mercantile Exchange Group 2013).

The questions this study address are: 1) whether an investment in options hedges the price risk of corn revenue protection policies 2) if it does, then how much the crop revenue insurer must pay for the risk transferred by the options representing value of the investment of put and call options.

## CHAPTER II

### LITERATURE REVIEW

The literature review section of this study is divided into two main sections. The first section explains why the crop insurance companies might use the option markets to hedge the price risk. The second section lays down the different methods of computing optimal hedge ratios.

#### **Reasons for Crop Insurer to Invest in the Option Market**

The following are three major reasons of the literature review related to crop revenue insurance providers to invest in the option contracts:

1. The increase in the demand for crop revenue insurance providing protection against systematic risks.
2. Profit motive intentions of crop insurance companies and private reinsurance companies.
3. The efficiency of the options market to hedge the systematic risks.

#### **Increase in the Demand for Crop Revenue Insurance**

In recent years, demand for revenue insurance has increased rapidly. RMA data show that the revenue-based policies account for three-quarters of the 2.2 million crop insurance policies that were active in 2015. On a premium basis, revenue policies account for 84% of all the policies (USDA 2015). Shaik et al. (2008) argued that farmers who



perceived greater yield or price risk are likely to purchase revenue insurance rather than yield insurance because the revenue insurance products like CRC (Crop Revenue Coverage), which has been renamed Revenue Protection (RP) provide greater per unit indemnities if the prices increased prior to harvest.

Further, crop insurance policies are heavily subsidized by the federal government. The subsidy given to the crop producers differs by the type of policy, unit structure, coverage level and type of coverage, ranging from 100 percent for catastrophic coverage to 38 percent for 85 percent coverage (Lusk 2016).

Crop revenue insurance policies provide the protection against the risks that are correlated among the exposure units. Crop prices are highly spatially correlated whereas yields may be correlated spatially depending on factors such as weather and disease (Coble and Dismukes 2008). Bulut et al. (2011) argued that despite the natural hedge between price and yield, sometimes large decline in harvest price can result in large aggregate indemnity payments which must be borne by revenue insurance providers and USDA.

Miranda and Glauber (1997) estimated that the insurance portfolio of the ten largest crop insurers was 20 times riskier than the portfolio of the insurers hedging independent risks. To hedge the systematic risk present in their portfolio, the insurance companies are purchasing reinsurance or derivative instruments reflecting the risk averse nature of the insurance companies.

### **Crop Reinsurance Industry**

In the US, the private insurance companies that write crop insurance policies in a state must offer the policies to any farmer in the state as per the rates and underwriting

provisions established by the RMA. To encourage the delivery of crop insurance in all regions of the country, including the areas where potential gains are low or risk exposure is high, the government provides reinsurance to the private companies. The US government limits crop insurers' exposure to underwriting risk through the Standard Reinsurance Agreement (SRA). According to the SRA, crop insurance companies can transfer their risk to RMA by allocating their policies into two different risk-sharing funds: Assigned Risk Fund and Commercial Fund. By assigning risky policies to the Assigned Risk fund, the insurers can transfer about 80 percent of their net losses in exchange for the same percent interest in premium. The insurers can retain at least 35 percent interest in premium and associated net losses in the Commercial fund (SRA 2015). The insurers have retained more premium to themselves in recent years, which has increased their exposure to risk dramatically (Coble et al. 2007). Thus, it seems reasonable to assume that the companies must find an alternative way to hedge their underwriting risks.

Private crop insurance companies can even purchase insurance for their portfolio from the private reinsurance companies. Reinsurance helps the crop insurance companies to stabilize reserves (Duncan and Myers 2000). Porth (2011) shows that private reinsurance is needed to diversify risk and operate a stable crop insurance program in Canada.

However, Miranda and Glauber (1997) argued that like small insurers, the reinsurance industry is unwilling or unable to offer policies that compensate systematic risk because it will be costly for reinsurers to hold the sufficient reserves that cover huge losses resulting from systematic risk. Further, there arises an incentive problem: the more

risk is transferred to a reinsurer, the less incentive the insurance company has to select low-risk policyholders, to reduce claim costs and to classify losses as being covered under the reinsurance contract (Mann 1992). Reinsurance companies have devised various contractual provisions and business practices such as coinsurance arrangement, deductible arrangement, explicit contingent pricing schemes, experience ratings to deal with this problem (Mann 1992). Thus, private crop insurance companies are not able to transfer all of its underwriting risk and/or systematic risk through reinsurance contracts whether offered by private reinsurer or the U.S. government.

Bulut, Schnapp, and Collins (2011) argued that in recent years the federal government has reduced its support and funding for the crop insurers. The 2008 farm bill reduced funding in the reinsurance program by 6.4 billion dollars for the coming years. In 2011, the SRA reduced expected company benefits by an additional 6 billion dollars. This mostly came in the form of a reduced reimbursement for delivery. In 2012, in order to reduce the deficit in the budget, the President suggested cutting an additional 6 billion dollars from the program over ten years. Having taken cuts in the governmental provided reinsurance, crop insurers need for alternative ways to hedge their risk. A possible way to hedge systematic risk is to use the futures or options instruments.

### **Efficiency of the Options Market to Hedge the Systematic Risks**

As Carter (2007) suggested hedging entails participating in the futures or options market to neutralize the effects of price risk by transferring the risk from hedgers to speculators. He further argued that the futures and options markets have been developed to correct market inefficiencies in commodity markets through the provision of intertemporal prices. Mann (1992) claimed that futures and options markets play an

important role in price discovery, shifting price risk and disseminating information. Moreover, the futures and options contracts are traded on organized exchanges and a clearing house of an organized exchange interjects itself as a seller/buyer to each buyer/seller facilitating trade by removing uncertainty related to the reliability of counterparties involved (Carlton 1984). As the need of traders for monitoring the other party's creditworthiness has been reduced, individuals are actively taking part in derivative markets. Participation in the derivative markets has been increased in such way that the positions are easily offset without large price disruptions (Carlton 1984). Hence, the futures and options markets can transfer systematic risk more efficiently as compared to the reinsurance.

Even though there are basic similarities between the futures and options, the buyer of the option enjoys more benefits as compared to the buyer of the futures as the buyer of either a put or call faces limited risks, does not have to deposit margin calls, can exercise multiple hedging strategies such as taking either short or long put and call positions (Carter 2007). Carter (2007) reasoned that options contracts are more like traditional price insurance than futures because either a floor or a ceiling price can be established with options. Thus, options contract is a more versatile tool for hedging as compared to the futures.

Luna (2012) claimed that most insurance companies mainly hedge their underwriting and investment risks through derivative markets. In 2014, the National Association of Insurance Commissioners reported that 94 percent of insurers are hedging at least some of their risk exposure using derivative instruments. Out of that, life insurers accounted for approximately 94 percent of the total industry notional value, property and

casualty insurers accounted for 6 percent. Swaps accounted for the largest share (49 percent) of the total industry notional value, followed by options (45 percent), futures (3 percent) and forwards (3 percent).

In the context of the crop insurance industry, many researchers have focused their studies on how crop insurance companies manage their risk through reinsurance, especially through the SRA. Hedging by the crop insurer through derivatives instruments has been rarely examined. Driedger, Porth, and Boyd (2016) recently analyzed the possibility of using futures and options to manage yield insurance losses on Canola in Western Canada. They studied the effectiveness of using the long futures position and long call option position in reducing the losses. They found that these hedge positions added to the insurance losses as compared to unhedged positions when the futures price declined. But the potential use of the put options to mitigate the price risk present in corn revenue protection policies has not been examined till now.

### **Comparing Methods of Computing Optimal Hedge Ratios**

#### **Minimum Variance Hedge Ratio**

In certain circumstances, firms may want to hedge the price risk associated with underlying assets for which futures contract may not be available or futures market of that underlying assets may not have enough liquidity. Only an alternative available for hedger is to buy futures contracts whose price movements closely correlate with the price of the assets to be hedged (Ritchken 1995).

Johnson (1960) was the first person who derived the number of futures contract to hedge a risk of portfolio based on the minimum variance model. Ederington (1979) established a model to examine how firms manage the price risks through futures market

in which he used risk (variance) minimizing objective function to find minimum variance hedge ratio.

The objective function is

$$Var(p_t) = Var(c_t) + h^2Var(f_t) - 2hCov(c_t, f_t)$$

where

$Var(p_t)$  = Variance of the returns on portfolio

$Var(c_t)$  = Variance of the returns from a long position in the cash market

$Var(f_t)$  = Variance of the returns from a short position in the futures market

$Cov(c_t, f_t)$  = Covariance between returns from the cash and futures market

$h_t$  = Hedge ratio

The first order condition with respect to  $h_t$

$$h_t^* = \frac{Cov(c_t, f_t)}{Var(f_t)}$$

The minimum variance hedge ratio also known as optimal hedge ratio is defined as the ratio of covariance between prices of underlying assets and futures contract to the variance of the futures prices (Zhang, Michael, and Leigh 2007).

A crop insurer could even use insurance derivative such as insurance futures contracts and insurance futures options to reduce its underwriting losses due to price decline. Insurance derivative is a financial instrument which value is derived from an underlying insurance index or characteristics of underlying insured risk (Luna 2012). For example, an owner of an insurance derivative can earn a return if index of a catastrophic events such as hurricane reached a specific index level. But the existence and liquidity of insurance derivative market itself is in question. That is why, this study uses the options whose price movements are correlated with the price of an underlying corn futures contract to hedge the increase in liability of corn revenue protection policies resulting from price variability. In these circumstances, minimum variance model can be useful to

calculate the optimal hedge ratio. While minimum variance model focuses on lowering the risk of portfolio and neglects the expected payoff from investment and investor's degree of risk aversion (Cotter and Hanly 2015), calculating optimal hedge ratio using advanced economic theory like mean-variance hedge and expected utility framework is more appropriate.

### **Mean-Variance Hedge**

Mean-Variance Hedge method assumed that a risk averse investor maximizes his/her utility holding various portfolios based on their means and variances (Carter 2007). In other words, the portfolio that offers return with certainty provides more utility to an investor than the alternative with variable income when the portfolios have same expected monetary payoff. McKinnon (1967) developed the mean-variance objective function given below for the producer and calculated the optimal hedge ratio.

$$\begin{aligned} MaxEU(p_t) &= E(p_t) - \gamma V(p_t) \\ &= Max \left\{ E(p_t) - \gamma \left[ Var(c_t) + h_t^2 Var(f_t) - 2h_t Cov(c_t, f_t) \right] \right\} \end{aligned}$$

where,  $\gamma$  is the relative measure of risk aversion.

Under these assumptions utility maximizing optimal hedge equals to the risk minimizing hedge ratio when expected return from the portfolio is zero (Kahl 1983).

### **Expected Utility Framework**

John von Neumann and Oscar Morgenstern (1947) developed mathematical models for examining the individuals' behavior under the uncertain situation. They laid out the hypothesis that every individual has different choices available under the uncertain situation and each choice has multiple consequences or outcomes. The

individual rank these possible outcomes from the least to the most desirable and the economists call this ranking utility.

As per this theory, a rational person has his/her own utility function over money and prefer those alternatives that maximized the utility rather than expected value of monetary payoff. Expected utility of an wealth is a weighted average of the utilities of wealth from each of the possible outcomes where weight being the probability of occurrence of each possible outcome. Mathematically, expected utility of wealth can be written as

$$E[U(W)] = \sum_i p_i U(W_i)$$

where

$E[U(W)]$  = Expected utility of wealth

$U(W_i)$  = Utility of wealth from different states of the world

$p_i$  = Probability of occurring different states of the world

$W_i$  = End-of-period wealth in different states of the world

Risk averse individuals prefer to avoid risky situations, even if the situations leads to a fair or large monetary payoff as extra money provide decreasing marginal utility (Walter & Nicholson, 2014). In fact, people may be willing to pay some amount to avoid the risky situation. That is why people and companies buy insurance to avoid unfavorable outcomes.

This study assumes that the insurance companies are risk averse rather than usual assumptions of risk neutrality (Chade & Schlee 2012) as they are purchasing reinsurance from the private reinsurance and the federal government SRA to mitigate their underwriting risk.



## CHAPTER III

### CONCEPTUAL FRAMEWORK

#### **Expected Utility from the Crop Insurer Perspective**

Assuming the von Neumann-Morgenstern axioms of behavior, the crop insurer maximizes its expected utility which is a function of end-of-period wealth. The assumption that the crop insurer is risk averse rather than risk neutral implies its utility function is strictly increasing, concave and twice continuously differentiable. This study considers the world where there is no USDA reinsurance available. Further, the distribution of the RMA insurance book of business reflects the book of all private crop insurer.

We have considered only the revenue protection (RP) and revenue protection with harvest price exclusion (RP-HPE) for corn. Net return on the crop insurance company's book of business,  $NR_B$ , defined in equation (1) represents the indemnity paid by the revenue policies out of the total premium received. The indemnity a crop insurance company pays depends on the policy holder's coverage levels, his/her yields history, and the level of price declines. Indemnity is paid only when revenue guarantee exceeds the actual revenue. Revenue guarantee for RP,  $RG_{RP}$ , mentioned in equation (2) is calculated as the product of the higher of projected price or harvest price, APH yield and chosen coverage level. Equation (3) shows revenue guarantee for RP-HPE,  $RG_{RP-HPE}$ , that is

computed as the projected price times APH yield times chosen coverage level. Actual revenue is estimated as actual harvest yield times harvest price.

$$(1) \quad NR_B = \sum_{i=1}^n \sum_{j=1}^n \left[ P_{r_{ij}} - \text{Max} \left( 0, \left( C_{l_{ij}} \text{Max}(f_{0I}, f_1) y_{gi} - f_1 y_i \right) \right) \right]$$

$$(2) \quad RG_{RP} = C_{l_{ij}} \times y_{gi} \times \text{Max}(f_{0I}, f_1)$$

$$(3) \quad RG_{RP-HPE} = C_{l_{ij}} \times f_{0I} \times y_{gi}$$

where

$P_{r_{ij}}$  = Premium of ith farm for jth coverage level

$j$  = Coverage level j

$C_{l_{ij}}$  = Coverage level j for ith farm

$f_{0I}$  = Spring futures price determined before sales closing

$f_1$  = Futures harvest price

$y_{gi}$  = Expected yield for ith farm

$i$  = Number of farms for coverage j

$y_i$  = Actual yield for ith farm

If  $C_{l_{ij}} \text{Max}(f_{0I}, f_1) y_{gi} > f_1 y_i$ , then the indemnity is paid.

If  $C_{l_{ij}} \text{Max}(f_{0I}, f_1) y_{gi} \leq f_1 y_i$ , then the indemnity is equal to zero.

To hedge the price risk, the insurer buys options from the options market at a cost equal to the options premium. The insurer buying the put options will exercise his/her right when the harvest price of corn december futures contract falls below the projected price providing protection against the price decrease. However, if harvest price of the corn December futures contract is going to rise above the projected price, the insurer may choose to buy call options. The crop revenue insurer exercises the call options to earn certain return that could be used to hedge the increase revenue guarantee when yield declines. This study has accounted the net return from put and call option separately while

calculating end-of-period wealth to independently obtain optimal hedge ratios for put and call. The net returns from put,  $NP$ , and call options,  $NC$ , are given below:

$$(4) \quad NP = h \times \text{Max}[0, (S_0 - f_{0h})] \times Q - h \times Q \times PP_0$$

$$(5) \quad NC = h \times \text{Max}[0, (f_{0h} - S_0)] \times Q - h \times Q \times CP_0$$

where

$f_{0h}$  = The futures price of the underlying futures contract

$S_0$  = Strike price of the put option

$PP_0$  = Put option premium

$h$  = Percentage of quantity ( $Q$ ) hedged

$Q$  = Bushels per contract

The crop insurer's initial wealth,  $W_0$ , is assumed to be one million dollars. An end-of-period wealth,  $W_e$ , of the company depends on whether the company indemnifies or not and the return on any put or call options purchased which can be expressed as follows:

$$(6) \quad W_e = W_0 + \sum_{i=1}^n \sum_{j=1}^n \left[ P_{r_{ij}} - \text{Max} \left( 0, \left( C_{l_{ij}} \text{Max}(f_{0I}, f_1) y_{gi} - f_1 y_i \right) \right) \right] + NP$$

$$(7) \quad W_e = W_0 + \sum_{i=1}^n \sum_{j=1}^n \left[ P_{r_{ij}} - \text{Max} \left( 0, \left( C_{l_{ij}} \text{Max}(f_{0I}, f_1) y_{gi} - f_1 y_i \right) \right) \right] + NC$$

The insurer's utility function is assumed to exhibit constant relative risk aversion (CRRA) and the risk aversion values of 1 to 3 are assigned where the lower value reflects lower degrees of risk aversion and vice-versa. A CRRA utility function is given by

$$(8) \quad \begin{cases} U(W_e) = \frac{W_e^{(1-R)}}{(1-R)} & \text{for } (R \neq 1) \\ U(W_e) = \ln W_e & \text{for } (R=1) \end{cases}$$

where R is the degree of risk aversion

Carter (2003) defined the optimal hedge ratio as the most desirable combination of cash and options positions and is chosen based on the shape of the hedger's indifference curve. The insurance company will choose to hedge at the ratio that maximizes its expected utility. Assuming two insured policies, RP and RP-HPE the objective function that maximizes the expected utility is expressed as:

$$(9) \quad \text{Max}_h L = \int \int \int U(W_e) f(y_1, y_2, f_1) dy_1 dy_2 df_1$$

We differentiate objective function with respect to h twice to obtain the independent optimal hedge ratio for put and call options.

$$(10) \quad \begin{aligned} L_h &= \int \int \int U(W_e) \times Q[(S_0 - f_{oh}) - PP_0] f(y_1, y_2, f_1) dy_1 dy_2 df_1 = 0 \\ L_h &= \int \int \int U(W_e) \times Q[(f_{oh} - S_0) - CP_0] f(y_1, y_2, f_1) dy_1 dy_2 df_1 = 0 \end{aligned}$$

A certainty equivalent is a certain amount of the income that a risk-averse individual finds equally desirable relative to the gamble. The certainty equivalent ( $CE_R$ ) based on the above utility function is given below:

$$(11) \quad \begin{aligned} CE_R &= e^{U(w_e)} \quad \text{if } (R = 1) \\ CE_R &= [U(W_e)(1-R)]^{1/R} \quad \text{if } (R \neq 1) \end{aligned}$$

## CHAPTER IV

### MODEL AND DATA

The behavioral model used to examine the planting-time optimization characteristics of a crop producer with yield insurance by Coble, Heifner, and Zuniga (2002) was modified to examine optimal hedge ratio of crop revenue insurers as both the crop producers and insurers are calculative about their wealth in multiple states of the world (good and bad), and are trying to reduce losses if there is a chance of bad state occurring through investment in financial instruments.

The insurer's certainty equivalents based on the expected utility function of crop insurance companies with different degrees of risk aversion were simulated over the period, 1985- 2015, using SAS. First, we simulated each component of crop insurance company's book of business, that is indemnity, premium and liability at national level for both corn revenue protection policies.

#### **Indemnity Simulation**

We simulated each component that determines the indemnity including the projected price, the harvest price, and the yield.

#### **Calculating Representative Farm Yields**

Yield variability for this study was simulated based on the model constructed by Coble, Dismukes, and Thomas (2007) where they analyzed the policy implications of

crop yield and revenue variability at differing levels of disaggregation. A linear time trend was estimated for 538 counties of 28 states having complete yield data series from 1985 to 2015, obtained from USDA'S National Agricultural Statistics Service (NASS). After we saved residuals and parameters from linear trend regression, we computed predicted yield for 2017. The predicted yield for 2017 represents the aggregate yield for each county. The estimated aggregate yield underestimates the individual yield variability of the county. To accurately estimate the county representative farm yield, we used a variability from the residuals, and inflated the variability to a level consistent with RMA base rates and added with 2017 predicted yield. We assumed that the representative farms have a mean yield equal to the expected county yield. Further the yield variability of the representative farm is supposed to be consistent with the average riskiness of farms participating in the corn revenue protection program.

### **Futures Prices Calculation**

Data on daily corn futures prices were obtained from the Commodity Research Bureau (CRB) database. Following the approach used by RMA to set the projected and harvest time prices for crop insurance contracts, the daily closing futures price for the December contract for corn was obtained for all trading days in February and a mean value computed. The same calculation was performed for the month prior to the expiration of the same contract in the same year, which allows the computation of the price changes from planting to harvest time. We then normalized the historical prices by around the projected price for 2017 equals to the 3.86 dollars per bushels. The harvest price also normalized to 2017 price levels as the product of the 2017 projected price and

the price ratio. The price ratio equals to the ratio of the harvest price to the projected price of each year.

### **Calculating Average Corn Coverage Level**

Using the 2016 summary of business data obtained from USDA's RMA, the acreage weighted average corn coverage level for both revenue protection policies in each county was calculated to simulate premiums, liabilities, and indemnities for each year.

After having computed each component, we then calculated the indemnities that could be indemnified by the both corn revenue protection policies, RP and RP-HPE, for each county representative farm. To obtain the national level indemnities for both the revenue policies for each year, we computed the weighted average of farm level indemnities where the weight used is the 2016 net acreages.

### **Premium Simulation**

In the US, RMA is responsible for establishing actuarial premium rates for crop insurance products. RMA uses loss ratio of total premiums to total indemnities to measure the actuarial fairness (USDA 2011). As per RMA (2011) loss ratio of an insurance program must not be more than one indicating the amount of premiums at least equal indemnities. Instead of using RMA premium, we simulated premium for corn revenue protection plan, which equals to the simulated average indemnity over three decades compensating the decline in projected price for each farm. Actual premium is function of various factors such as price volatility, producer choice of coverage level, unit structure, crop rotation and varying acreage over time. As we simulated the net return on

insurance company's book of business, we simulated premium too so that the premium amount be consistent with simulated farm level indemnities. The national level premiums for each year calculated by weighted averaging each farm level where weight was the 2016 net acreages.

The premium was estimated solely based on the projected price. The increase in harvest price increases the revenue guarantee, not the premium as crop insurance provider has already accounted the possibility of an increase in guarantee when offering revenue protection policies.

### **Liability Simulation**

The liability for each farm was calculated as the product of 2017 projected price, weighted average corn coverage level and 2017 predicted yield of each farm. Farm level liabilities were weighted by the 2016 net acreages and were added together to get the national level liability for each year.

The national level indemnity, and premium were normalized with respect to one million dollars in the liability for every year so that premium received and indemnity paid by the insurance company are certain percentage of one-million-dollar liability.

### **Options Payout**

If the crop insurance company wants to protect against the price risk, it would purchase put and call options contracts after it sells the revenue protection policies and well before the harvest period. The crop insurers do know the expected yield. But the actual corn production is unknown for the crop insurer. So, we assumed the crop insurer was hedging a certain percentage of one million bushels of corn through the options



contracts. We increased the hedging quantity by half a percentage increments and calculated the option payouts at each percentage as the percentage of one million bushels hedged times the difference between the projected price and the harvest price for each year.

We considered options contract specified at the money. The put options premium was calculated in two different ways. In first method, we calculated put option premium as change in futures prices, that is difference between projected price and harvest price, with respect to projected price and multiply with one million bushels and calculated average of 31 years from 1985-2015. Second, we calculated the mean return of the options payout from 1985-2015 when the crop insurance company completely hedged one million bushels as 100 percent times one million bushels times projected price minus harvest price divided by 31 years when projected price is greater than harvest price. As the put option contracts was at the money, we choose higher value of put options premium calculated from two different ways. In the same way, call options premium was calculated expect that we calculated the price difference as harvest price minus projected price when harvest price is greater than projected price.

### **Expected Utility Simulation**

We calculated expected value of end-of-period wealth representing the multiple states of the world. Based on expected value of end-of-period wealth we computed expected utility, over the period, 1985-2015, for a different degree of risk aversion. Given a relative risk aversion level, at first, we computed certainty equivalents at ten percent, then five percent, one percent and half of a percent hedge ratio intervals. Then a grid

search of the certainty equivalents to variations in these hedge ratio intervals was carried out. The hedge ratio with highest certainty equivalent was deemed optimal.

## CHAPTER V

### RESULTS

The results section is organized into three main subsections:

- Descriptive statistics of end-of-period wealth under different alternatives
- Optimal hedge ratio under different scenarios
- Evaluation of options contract effectiveness

#### **Descriptive Statistics of End-Of-Period Wealth from Different Alternatives**

Multiple choices for the crop insurer are represented by the percents of the quantities of corn the insurer wants to hedge through the put and call options contracts. Initial wealth is assumed to be one million dollars. The assumption of initial wealth does affect the expected end-of-period wealth, utility, certainty equivalent gains and willingness to pay. As initial wealth is directly proportional to the end-of-period wealth, increase in initial wealth increases the expected end-of-period wealth. But expected utility decreases when initial wealth increases because the marginal utility of wealth diminishes as wealth increases. Since certainty equivalent gain is inversely proportional to expected utility, certainty equivalent gain increases as expected utility decreases.

The crop insurance company assumed to hedge up to one million bushels of corn. If a crop insurer hedge 26 percent of one million bushels, that is 260,000 bushels, then

the insurer would have hedged about one-million-dollar liability. Hedging more than 26 percent means the company is hedging more than one-million-dollar liability.

The mean and standard deviation of end-of-period wealth when a crop insurer buys options contracts to hedge the price risk of both corn revenue protection policies, RP and RP-HPE are shown in the table 5.1. Means of end-of-period wealth for all possible investments in the options contracts including the return from insurance business from the corn revenue protection policies sold on the regions including and excluding the Corn Belt are same for each percent of one million bushels hedged.

Table 5.1 Descriptive statistics of end-of-period wealth under different alternatives scenarios

Regions	Corn Revenue Insurance Products	Options	Percentage of Quantity Hedged	Mean (\$)	Standard Deviation (\$)
Including Corn Belt	RP	Put	0	985284.44	76466.86
Including Corn Belt	RP-HPE	Put	1.5	998565.28	48730.50
Excluding Corn Belt	RP	Put	2	992604.38	57737.13
Excluding Corn Belt	RP-HPE	Put	3.5	998928.84	49730.04
Including Corn Belt	RP	Call	9	985284.44	65152.39
Including Corn Belt	RP-HPE	Call	2	998565.28	48291.91
Excluding Corn Belt	RP	Call	1.5	992604.38	57718.76
Excluding Corn Belt	RP-HPE	Call	0	998928.84	51473.40

When various investment alternatives have the same mean value, the investment with the smaller variability of return is the most desirable. Table 5.1 reports percentages of quantity hedged where standard deviations are smaller. Here standard deviations indicate volatility in returns from hedging and both the types of corn revenue policies for a crop insurance company. The standard deviation is small when the crop insurer chooses not to hedge the price risk of RP sold in the region including Corn Belt through put options. But the standard deviation for RP-HPE sold in the same region is lower when the insurer hedged at 1.5 percent of one million bushels using put options contract. The variabilities of return from the RP and RP-HPE sold in the region excluding Corn Belt are lower when the company hedges 2 and 3.5 percents of one million bushels using put options.

The size of deviations of end-of-period wealth from the mean value is smaller when the company hedges 9 and 2 percent of one million bushels through call option to mitigate the price risk of RP and RPHE sold in the region including Corn Belt. For the region excluding the Corn Belt the standard deviations are smaller at 1.5 and 0 percents of the quantity hedged using call option for RP and RP-HPE.

### **Optimal Hedge Ratio Under Different Scenarios**

We calculated the certainty equivalent up to 30 percent of hedge ratio in five percent interval where certainty equivalent was maximum at 10 percent when we considered all the alternative scenarios, including and excluding Corn Belt. Then we refined the grid search from 5 to 1 percent and then from 1 to 0.5 percent hedge ratio interval to find the optimal hedge ratio within an 0.5 percent approximation. We calculated the optimal hedge ratio as a percent of both one million bushels and liability.

The optimal hedge ratio as a percent of one million dollars in liability is calculated as percentage of hedged quantity where certainty equivalents is maximum times projected price and divided by one-million-dollar liability.

We considered two different scenarios in our analysis where we included and excluded the major Corn Belt. We excluded Corn Belt because in major production regions for corn and soybeans, there is empirical evidence of a natural hedge between farm yields and aggregate prices (Coble, Heifner and Zuniga 2002). Both the yield and price components of crop revenue are random variables that are subject to spatial dependence (systematic risk). It is obvious that the price risk is a systematic risk since both projected and harvest prices are derived from the futures markets. Sometimes yield variability is also subject to systematic risk if there is a positive correlation between a farm and nearby farms due to spatially correlated weather events or disease. As there is the inverse relationship between price and yield, huge decline in the corn price can be compensated through large production in the Corn Belt. Thus, we removed the major Corn Belt states, Ohio, Indiana, Illinois, Iowa and Missouri from our analysis so that the insurance company's portfolio reflects the price risk of corn revenue protection policies and we could use options to hedge the risk. While we included the Corn Belt based on the assumption that despite the natural hedge between price and yield, sometimes huge decline in harvest price can result in larger aggregate indemnity payment (Bulut et al. 2011).

It is obvious that the decline in harvest price can result in the indemnity payment by the crop revenue insurer. To hedge the decline in harvest the price, the crop insurer is assumed to invest in put options. Even increased in the harvest price results in indemnity

when yield declines thus useful for the crop producers who normally forward price much of their production before harvest (Edwards 2014). When a crop producer harvest fewer bushels than specified in the forward contract, the producer must fulfill the obligations by purchasing the deficit bushels from the market. Sometimes the market price of crop which was forward contracted may be higher than the strike price. In such a case buying the crop revenue policies results in the indemnity payment which offsets the cost of purchasing deficit bushels at a market price above the strike price of forward contract. Therefore, we even analyzed whether a crop insurance company could hedge the increase price through buying call options contracts. Call options will be exercised only when harvest price is greater than that of projected price.

We assumed a crop insurance company's utility function exhibits constant relative risk aversion with degrees of risk aversion equals from one to three. This section explains finding related with moderate degree of risk aversion equals to two.

For the optimal quantity hedged through put options, the crop insurance company must pay the put premium equal to percentage of quantity hedged where certainty equivalent is maximum times \$ 345,070.97. \$ 345,070.97 is the amount that is higher of put option calculated in two different ways. Like wise call option premium equal to \$191,509.68 times percent of optimal quantity hedged.

Table 5.2 Optimal Hedge Ratios Under Different Scenarios

Regions	Corn Revenue Insurance Products	Options	Optimal Quantity Hedged (in percentage)	Optimal Quantity Hedged (in bushels)	Optimal Liability Hedged (in percentage)	Optimal Liability Hedged (in \$)	Certainty Equivalents (in \$)	Option Premium (in \$)
Including Corn Belt	RP	Put	0	0	0	0	978058.42	0
Including Corn Belt	RP-HPE	Put	0.5	5000	1.93	19300	995881.82	1725.35
Excluding Corn Belt	RP	Put	1	10000	3.86	38600	988700.74	3450.70
Excluding Corn Belt	RP-HPE	Put	3	30000	11.58	115800	996029.86	10352.12
Including Corn Belt	RP	Call	10.5	105000	40.53	405300	980688.40	20108.51
Including Corn Belt	RP-HPE	Call	2	20000	7.72	77200	995959.35	3830.19
Excluding Corn Belt	RP	Call	1.5	15000	5.79	57900	988732.77	2872.64
Excluding Corn Belt	RP-HPE	Call	0	0	0	0	995905.10	0

Note: Results when degree of risk aversion equal to two, R=2.



Certainty equivalents and optimal hedge ratios under different scenario when degree of risk aversion equals to two are shown in table 5.2 for evaluating whether hedging through put and call options is feasible or not. Clearly, the crop insurance company is better off if it does not invest in the put option contracts to hedge the price risk of RP sold in the regions including Corn Belt. As certainty equivalent is maximum when 0 percent of quantity is hedged, optimal hedge ratio is 0 percent of one-million-dollar liability. Similarly, certainty equivalent gain is maximum when the crop insurance company hedged 0.5 percent of one million bushels to mitigate the price risk of RP-HPE sold in the same region through put option signifying the optimal hedge ratio is 1.93 percent of one-million-dollar liability. The put option premium paid by the insurance company for this benefit equal to \$1,725.35.

When we considered the scenario where we excluded the Corn Belt and used the put options to hedge the price risk, we find that certainty equivalent is maximum at 1 and 3 percent of one million bushels hedged for RP and RP-HPE. Thus, optimal hedge ratio from put options hedging the price risk of RP and RP-HPE sold in the regions excluding Corn Belt is 3.86 and 11.58 percent of one million dollars of liability. The insurer pays the put premiums equal to \$3,450.70 and \$10,352.12 dollar for put options hedging the price risk of RP and RP-HPE.

Certainty equivalent from call options and RP sold in the regions including Corn Belt is maximum when the insurer hedged the 10.5 percentage of one million bushels, 105,000 bushels, thus optimal hedge ratio is 40.53 percentage of million-dollar liability. For hedging the 40.53 percentage of one-million-dollar liability, the company pays \$20,108.51 as a call option premium. Similarly, for RP-HPE certainty equivalents is

maximum at 2 percent of one million bushels of corn hedged through call option and optimal hedge ratio is 7.72 percent of one-million-dollar in liability. The insurer must pay \$3,830.19 for call option to hedge the price risk of RP-HPE.

The results from the simulation indicate that maximum certainty equivalent from call options to hedge the price risk of RP sold in the region excluding the Corn Belt is at 1.5 percent of one million bushels hedged and 5.79 percent of one million dollars in liability. To reap this benefit, the insurer buys the call option at \$2,872.64 dollars through option market. Certainty equivalent is maximum when the insurer chooses not to hedge the price risk of RP-HPE through call options and thus optimal hedge ratio is zero percent of one million dollars in liability.

### **Evaluation of Options Contract Effectiveness**

#### **Results from Put Options in the Study Region Including Corn Belt**

To hedge the price risks of both corn revenue protection policies, RP and RP-HPE, there should be a strong positive correlation between the indemnity paid by each policy and the return from the put options contracts.

Table 5.3 Pearson Correlation Coefficient Between Return from Put Options Return and Indemnity Paid from Corn RP

	Return from the Put Options	Indemnity
Return from the Put Options	1	0.0288 (0.877)
Indemnity	0.0288 (0.877)	1

Table 5.3 indicates that the correlation coefficient,  $r = 0.0288$ , between the return from the put options contracts and indemnity paid from the corn revenue protection policies is not strong enough to hedge the price risk of corn RP policies. The p-value of the coefficient is 87.7 percent. The probability that we do not reject the null hypothesis that the returns from put options contract would not able to hedge indemnity paid by the crop insurer is high. Thus, the correlation coefficient is not statistically significant to justify hedging.

Since RP provides protection against both upside and downside price risk, the simple put options contract is not effective enough to transfer the price risk but should be effective for the corn revenue protection policies providing protection against only the downside price risk (RP-HPE). Table 5.4 clearly shows that correlation coefficient increases to 0.3385 and is statistically significant. At the 6.2 percent level of significance, we reject the null hypothesis that the investment in put options contract could not hedge the price risk inherent in RP-HPE.

Table 5.4 Pearson Correlation Coefficients between Put Options Return and Indemnity Paid from Corn RP-HPE

	Return from the Put Options	Indemnity
Return from the Put Options	1	0.33854 (0.062)
Indemnity	0.33854 (0.062)	1

**Results from Put Options in the Study Region Excluding Major Corn Belt**

Table 5.5 Pearson Correlation Coefficients between Put Options Return and Indemnity Paid from Corn RP

	Return from the Put Options	Indemnity
Return from the Put Options	1	0.11355 (0.543)
Indemnity	0.11355 (0.543)	1

Table 5.6 Pearson Correlation Coefficients between Put Options Return and Indemnity from for Corn RP-HPE

	Return from the Put Options	Indemnity
Return from the Put Options	1	0.24881 (0.1771)
Indemnity	0.24881 (0.1771)	1

The correlation coefficient is around 0.11355 moderate enough to hedge the price risk of RP sold in the regions excluding the Corn Belt using put options contracts. The p-value of the correlation coefficient 0.11355 is 54.3 percent which is not statistically significant. Correlation between the return from put options contract and the indemnity from corn RP-HPE is 0.24881 enough to justify the hedging of 11.58 percent of one-million-dollar of liability through the put options contracts. The probability of rejecting the hypothesis that there is not strong positive correlation between return from put options and indemnity paid by the corn RP-HPE is 17.71 percent. Thus, the correlation coefficient is statistically significant when we consider 80 percent confidence interval.

### **Results from Call Options in the Study Region Including Major Corn Belt**

There is a strong positive correlation ( $r = 0.6177$ ) between the return from call options and indemnity from RP. The correlation coefficient is statistically significant at p-value equals to 2 percent level indicating that we reject the null hypothesis that the return from call options would not be able to hedge the indemnity results from the price variability. The correlation coefficient between return from RP-HPE and return from call options is 0.2895 which is strong enough to hedge the price risk of RP-HPE through call option. The correlation coefficient is statistically significant if we accept p-value equals to the level of significance 11.4 percent.

Table 5.7 Pearson Correlation Coefficients Including Corn Belt between Call Options Return and Indemnity paid by RP

	Return from the Call Options	Indemnity
Return from the Call Options	1	0.6177 (0.0002)
Indemnity	0.6177 (0.0002)	1

Table 5.8 Pearson Correlation Coefficients Including Corn Belt between Call Options Return and Indemnity paid by RP-HPE

	Return from the Call options	Indemnity
Return from the Call options	1	0.2895 (0.1141)
Indemnity	0.2895 (0.1141)	1

### Results from Call Options in the Study Region Excluding the Corn Belt

The correlation coefficients between the return from exercising the call options contract and indemnity paid from the corn RP is 0.127 which is low but enough to hedge some price risk. There is a negative correlation ( $r = -0.058$ ) between call options return and indemnity paid by corn RP-HPE. In both cases, p-value of the correlation coefficients are not statistically significant enough to justify the hedging through call options.

Table 5.9 Pearson Correlation Coefficients Excluding Corn Belt between Call Options Return and RP Indemnity

	Return from the put options	Indemnity
Return from the put options	1	0.127 (0.496)
Indemnity	0.127 (0.496)	1

Table 5.10 Pearson Correlation Coefficients Excluding Corn Belt between Call Options Return and RP-HPE Indemnity

	Return from the Call Options	Indemnity
Return from the Call Options	1	-0.05807 (0.1771)
Indemnity	-0.05807 (0.1771)	1

## CHAPTER VI

### CONCLUSION

Insurance company typically does not insure systematic risks. However, crop revenue insurance is one of the exceptions which offers a guarantee against a highly systematic risk - price risk. The crop insurance company must indemnify all its policyholders holding corn revenue protection policies if the price or yield of corn decline. One way to hedge systematic risk is to buy put options and call options on futures contract to hedge price variability.

We examine whether an investment in options by an insurance company, having moderate degree of risk aversion, hedges the price risk of RP and RP-HPE sold in the regions including and excluding the Corn Belt. We find that optimal hedge ratios are 3.86, 1.93 and 11.58 percentages of one million dollars of liability for put options when hedge decline in the price risk of RP sold in the regions excluding the Corn Belt and RP-HPE sold in the both regions including and excluding the Corn Belt. But the optimal hedge ratio is zero percent of one-million-dollar of liability when the insurer invests in put option contract for RP sold in the regions including the Corn Belt.

The price increase results in increased revenue guarantee when yield declines. The increase in price risk could be hedged through buying call options. This study indicates that call options can hedge the increased revenue guarantee of corn RP sold in states including and excluding Corn Belt. For RP sold in the states including and



excluding Corn Belt, certainty equivalents are maximum at 10.5 and 1.5 percent of quantity hedged signifying that the optimal hedge ratios are 40.53 and 5.79 percent of one-million-dollar liability. Meanwhile when the crop insurance company sells the RP-HPE in the regions where five states of Corn Belt were excluded results in zero optimal hedge ratio.

The reason behind hedging through put and call options does and does not work in many of the cases as mentioned above as the correlation between return from put, and call options and indemnity from both policies must be positive and strong enough to justify hedging.

Moreover, the assumptions of the initial wealth of the crop insurance company to be one million dollars and the company is hedging a certain percentage of one million bushels has limited our analysis thereby has affected the study's results.

A crop insurance company has multiple policies in each county but might not have a fully national portfolio. But we conduct the analysis based on the assumption that the distribution of business reflects all private crop insurers, thus the results of the study may not be applicable to crop insurers who have a regional portfolio.

Further, we assumed the world where USDA reinsurance and other hedging tools for the crop insurer are not available and the insurer had to mitigate price risk through the derivative instrument, especially long put and call options. The results distinctly indicate that considering only put and call options as a risk management tool is not appropriate if the crop insurance company wants to hedge all its price risk through options market. Therefore, the crop insurers who want to offset all its price risk inherent in the corn

revenue protection insurance should mix a wide variety of hedging tools with standard reinsurance agreement and/ or allocate its assets strategically.

While this study has tried to explore the possibility of mitigating the price risks presents in the corn revenue protection policies through simple hedging tool, put and call options, the future studies must extend to mix the variety of hedging tools such as futures, swaps and so on with federal or private reinsurance that might offer the solution for mitigating the correlated underwriting risk such as price risk.

## REFERENCES

- Bouzouita, R., and A. J. Young. 1998. "Property/Casualty Insurer's Use of Financial Derivatives Usage." *CPCU Journal* 51(2):114-119.
- Cater, C. A. 2003. *Futures and Option Markets*. Illinois: Waveland Press Inc.
- Carlton, D. W. 1984. "Futures Markets: Their Purpose, Their History, Their Growth, Their Successes and Failures." *The Journal of Futures Markets* 4(3): 237-271.
- Center for National Security, The MITRE Corporation. 2014. "The Effectiveness of a Costs and Benefits Analysis in Making Federal Government Decisions." Mclean VA.
- Chade, H., and Schlee, E. 2012. "Optimal insurance with adverse selection". *Theoretical Economics* 7(3):571-607.
- Chicago Mercantile Exchange Group. 2013. *A Trader's Guide to Futures: Thought Leadership with a Global Perspective*. Retrieved April 9, 2017, from CME Group Web site: <https://www.cmegroup.com/education/files/a-traders-guide-to-futures.pdf>
- Coble, K. H., and R. Dismukes. 2007. "Distributional and Risk Reduction Effects of Commodity Revenue Program Design". *Review of Agriculture Economics* 30(3):543-553.
- Coble, K. H., R. Dismukes, and J. W. Glauber. 2007. "Private Crop Insurers and the Reinsurance Fund Allocation Decision". *American Journal of Agriculture Economics* 89(3):582-595.
- Coble, K. H., R. G. Heifner, and M. Zuniga. 2000. "Implications of Crop Yield and Revenue Insurance for Producer Hedging." *Journal of Agriculture and Resource Economics* 25(2):432-452.
- Coble, K. H., R. Dismukes, and S. Thomas. 2007. "Policy Implications of Crop Yield and Revenue Variability at Differing Levels of Disaggregation". Paper presented at the AAEA Annual Meeting, Portland, Oregon, July 29-August 1.

- Colquitt, L. L., and R. E. Hyot. 1997. "Determinants of Corporate Hedging Behavior: Evidence from the Life Insurance Industry." *The Journal of Risk and Insurance* 64(4): 401-415.
- Cotter, J, and Hanly, J. 2015. "Performance of Utility Based Hedges". *Energy Economics* 49: 718-726.
- Cummins, J. D., R. D. Phillips, and S. D. Smith. 2001. "Derivatives and Corporate Risk Management: Participation and Volume Decisions in the Insurance Industry." *The Journal of Risk and Insurance* 68(1): 51-91.
- Denmark, Environmental Assessment Institute. 2002. Risk and Uncertainty in Cost Benefit Analysis. Copenhagen: Government Printing Department. April 2006.
- Driedger, J., P. Lysa, and B. Milton. 2016. "The Potential to Use Futures and Options to Manage Crop Insurance Losses." Paper presented at AAEA annual meeting, Boston MA, 31 July-2 August.
- Duncan, J., and R. J. Myers. 2000. "Crop Insurance under Catastrophic Risk." *American Journal of Agriculture Economics* 82(4):842-855.
- Ederington, L. 1979. "The Hedging Performance of the New Futures Markets". *Journal of Finance* 34():157-170.
- Edwards, W. 2014. "Revenue Protection Crop Insurance." Iowa State University Extension and Outreach. Ames: Ag Decision Maker. Retrived from <https://www.extension.iastate.edu/agdm/crops/pdf/a1-54.pdf>
- Harun, B., F. Schnapp, and C. H. Keith. 2011. "Systematic Risk and Crop Insurance in Retrospect and Prospect." *Crop Insurance Today*, November pp. 4-8.
- Johnson, L. 1960. "The Theory of Hedging and Speculation in Commodity Futuers." *The Review of Economic Studies* 27(2):139-151.
- Kahl, K. H. 1983. "Determination of the Recommended Hedging Ratio." *American Journal of Agriculture Economics* 65(3):603-605.
- Kolb, R. W. 1996. *Financial derivatives*. Oxford: Blackwell Publishers.
- Kornhauser, L. A. (2000). "Cost-Benefit Analysis: Legal, Economic, and Philosophical Perspective." *The Journal of Legal Studies* 28(1):89-100.
- Lusk, J. L. 2016. "Distributional Effects of Crop Insurance Subsidies." *Applied Economic Perspectives and Policy*, ():ppw002.

- Makowsky, M. D., and R. E. Wagner. 2009. "From Scholarly Idea to Budgetary Institution: The Emergence of Cost-Benefit Analysis." *Constitutional Political Economy* 20(1):57-70.
- Mann, S. V. 1992. "The Trading of Underwriting Risk: An Analysis of Insurance Futures Contracts and Reinsurance." *The Journal of Risk and Insurance* LIX(4):601-627.
- Massey, R., J. Widdows, D. Law, K. Bhattacharya, H. Wendy, and H. David. 2002. "Insurance Company Failure." Working party report to the General Insurance Convention.
- Miranda, M. J., and J. W. Glauber. 1997. "Systemic Risk, Reinsurance and the Failure of Crop Insurance Markets". *American Journal of Agricultural Economics*, 79(1): 206-215.
- Porth, L. 2011. "A Portfolio Optimization Model Combining Pooling and Group Buying of Reinsurance Under an Asset Liability Management Approach." PhD dissertation, University of Manitoba.
- Ritchken, P. 1995. *Derivative Markets: Theory, Strategy and Applications*. Addison-Wesley.
- Shaik, S., K. H. Coble, T. O. Knight, A. E. Baquet, and F. G. Patrick. 2008. "Crop Revenue and Yield Insurance Demand: A Subjective Probability Approach". *Journal of Agricultural and Applied Economics Association*. 40(3):757-766.
- Synder, C., and W. Nicholson. 2012. *Microeconomic Theory: Basic Principles and Extensions*. United States of America: Thomson South-Western.
- U.S. Congress, Congressional Research Service Shields. 2015. *Federal Crop Insurance: Background*. 13 August.
- USDA-National Agricultural Statistics Services. Quick Stats: U.S. and All States County Corn Yield Data. Washintong, D.C. Retrieved from [https://www.nass.usda.gov/Quick\\_Stats](https://www.nass.usda.gov/Quick_Stats)
- USDA Risk Management Agency. 2011. *Revised Premium Ratings for Corn and Soybeans*. Retirved from <https://www.rma.usda.gov/help/faq/rerating.html>
- U.S. Department of Agriculture, Risk Management Agency. 2015. *Standard Reinsurance Agreement*. Washington DC.
- USDA Risk Managemnt Agency. (2016). *Summary of Business*. Federal Crop Insurance

Corporation. Retrieved from

<http://prodwebnlb.rma.usda.gov/apps/SummaryofBusiness/ReportGenerator>

von Neumann, J., & Oskar, M. 1947. *Theory of Games and Economic Behavior*.

Princeton: Princeton University Press.

Yi, J., J. W. Richardson, and H. L. Bryant. 2016. "How Do Premium Subsidies Affect Crop Insurance Demand at Different Coverage Levels: the Case of Corn". Paper presented at the AAEA Annual Meeting, Boston MA, July 31-August 2.

Zhang, Q., Michael, R., & Leigh, M. (2007). Hedging Decisions of Importing Firms of U.S. Commodity with Multiple Risks: The case of Soybeans. *Annual Meeting*. Portland: American Agricultural Economics Associations.

Zerbe, Jr., R. O. and A. S. Bellas. 2006. *A Primer for Benefit-Cost Analysis*. Cheltenham, UK: Edward Elgar.

APPENDIX A  
PROGRAMMING CODES AND TABLES

**Tables Showing Results When Degree of Risk Aversions Equal to One and Three**

Table A.1 Optimal hedge ratios under different scenarios.

Regions	Corn Revenue Insurance Products	Options	Optimal Quantity Hedged (in percentage)	Optimal Quantity Hedged (in bushels)	Optimal Liability Hedged (in percentage)	Optimal Liability Hedged (in \$)	Certainty Equivalents (in \$)	Option Premium (in \$)
Including Corn Belt	RP	Put	0	0	0	0	981946.73	0
Including Corn Belt	RP-HPE	Put	1	10000	3.86	38600	997292.11	1725.35
Excluding Corn Belt	RP	Put	1.5	15000	5.79	57900	990771.84	5176.06
Excluding Corn Belt	RP-HPE	Put	3	30000	11.58	115800	997571.41	10352.12
Including Corn Belt	RP	Call	10	100000	38.6	386000	983062.53	19150.96
Including Corn Belt	RP-HPE	Call	2	20000	7.72	77200	997325.36	3830.19
Excluding Corn Belt	RP	Call	1.5	15000	5.79	57900	990783.63	2872.64
Excluding Corn Belt	RP-HPE	Call	0	0	0	0	997500.33	0

Note: Results when degree of risk aversion equals to one.



Table A.2 Optimal hedge ratios under different scenarios

Regions	Corn Revenue Insurance Products	Options	Optimal Quantity Hedged (in percentage)	Optimal Quantity Hedged (in bushels)	Optimal Liability Hedged (in percentage)	Optimal Liability Hedged (in \$)	Certainty Equivalents (in \$)	Option Premium (in \$)
Including Corn Belt	RP	Put	0	0	0	0	973535.05	0
Including Corn Belt	RP-HPE	Put	0.5	5000	1.93	19300	994320.55	1725.35
Excluding Corn Belt	RP	Put	0.5	5000	1.93	19300	986359.11	1725.35
Excluding Corn Belt	RP-HPE	Put	2.5	25000	9.65	96500	994276.48	8626.77
Including Corn Belt	RP	Call	11	110000	42.46	424600	978142.97	21066.06
Including Corn Belt	RP-HPE	Call	2	20000	7.72	77200	994449.91	3830.19
Excluding Corn Belt	RP	Call	1.5	15000	5.79	57900	986415.79	2872.64
Excluding Corn Belt	RP-HPE	Call	0	0	0	0	994120.12	0

Note: Results when degree of risk aversion equal to three.

## Programming Codes

### Python Codes

For sub setting complete set of yield observations, that is data from 1985-2015 from the yield data obtained from NASS, the following python codes was run.

```
import csv
import os
f=open(r'C:\Users\sweta\Desktop\data\data.csv','r')
f1=open(r'C:\Users\sweta\Desktop\data\cleaned.csv','w')
f2=open(r'C:\Users\sweta\Desktop\data\deleted.csv','w')
f3=open(r'C:\Users\sweta\Desktop\data\state_county.csv','w')
fop=csv.reader(f)
fop1=csv.writer(f1)
fop2=csv.writer(f2)
fop3=csv.writer(f3)
sta_co=[]
a=0;
sta_coun={}
for row in fop:
    if a==0:
        a+=1;
        continue
    key1=str(row[2])+str(row[6]);

    if key1 in sta_coun.keys():
        sta_coun[key1]=sta_coun[key1]+1
    else:
        sta_coun[key1]=1

a=0;
```

```

f.close()
f=open(r'C:\Users\sweta\Desktop\data\data.csv','r')
fop=csv.reader(f)
for row in fop:
    if a==0:
        fop1.writerow(row)
        fop2.writerow(row)
        a+=1;
        continue

    key2=str(row[2])+str(row[6]);
    if sta_coun[key2]==31:
        fop1.writerow(row)
        if key2 not in sta_co:
            sta_co.append(key2)
            fop3.writerow([str(row[2]), str(row[6])])

    else:
        fop2.writerow(row)

f.close()
f1.close()
f2.close()
f3.close()

```

Based on the cleaned data set, coverage level for the counties and states from the data set, summary of business, similar to those of cleaned data set was extracted manually.

## SAS CODES

The following SAS code was run to conduct simulation on the insurance company book of business, option positions, expected utility and certainty equivalent gains. We calculated willingness to pay using MS Excel 2016.

### Put options Including Corn Belts Hedging the Price Risk of Corn RP

```
libname analysis 'C:\Users\sweta\Desktop\data1';
proc import out= Analysis.data1985 datafile =
"C:\Users\sweta\Desktop\data1\cleaned1985.xlsx"
DBMS= xlsx;
sheet = "cleaned1985";
GETNAMES= YES;
run;

proc sort data=Analysis.data1985;
by year;
proc import out= Analysis.cornprices datafile =
"C:\Users\sweta\Desktop\data1\corn_crop_futures_summary3.xlsx"
DBMS= xlsx;
sheet = "sheet3";
GETNAMES= YES;
run;
proc sort data=Analysis.cornprices;
by year;

proc import out= Analysis.corn_business_summary datafile =
"C:\Users\sweta\Desktop\data1\Required85_16.xlsx"
DBMS= xlsx;
sheet = "Required85_16";
GETNAMES= YES;
run;
proc sort data=Analysis.corn_business_summary;
by astate acounty;
data Analysis.corn_business_summary1;
set Analysis.corn_business_summary;
if insplan = 2;
run;
/*WEIGHTED COVERAGE LEVEL by county and state*/
proc means data = analysis.corn_business_summary1 ;
by astate acounty;
var cvglvl;
weight netacr;
output out= analysis.current_mean_county_covg MEAN=cvglvl;
run;
proc sort data = analysis.corn_business_summary1;
by astate acounty;
run;
```

```

    data Analysis.corn_business_summary2 (keep = astate acounty
netacr);
    set Analysis.corn_business_summary1;
    run;
    proc sort data = analysis.corn_business_summary2;
    by astate acounty;
    run;

proc sort data=Analysis.current_mean_county_covg;
by astate acounty;
run;
/*Simulation for yield*/
proc sort data = Analysis.Data1985;
by astate acounty;
run;
proc reg noprint data=Analysis.Data1985 outest= Analysis.Regout;
/*Saving Intercept and Parameter*/
model value = year; /*Linear trend for each county*/
by astate acounty;
output out= Analysis.b /*Saving Residuals*/
p = vhat
r= vresid;
run;
proc sort data = Analysis.b;
by astate acounty;
run;
data Analysis.pred;
set Analysis.regout;
ypred = Intercept + year*2017; /*Simulation for yield guarantee by
county*/
run;
proc sort data = Analysis.pred;
by astate acounty;
run;

data Analysis.C (keep = year astate acounty intercept year ypred
vresid);
merge Analysis.pred Analysis.b;
by astate acounty;
run;
proc sort data =Analysis.C;
by astate acounty;
run;
data Analysis.D;
set Analysis.C;
dtrendY = ypred + vresid * 1.66; /*Simulation for average
representative farm yield*/
if dtrendY < 0 then dtrendY = 0;
run;
proc sort data = Analysis.D;
by astate acounty;
run;
data Analysis.E;

```

```

merge Analysis.current_mean_county_covg Analysis.D;
by astate acounty;
run;
proc sort data = Analysis.E;
by year;
Data Analysis.F;
merge Analysis.E Analysis.Cornprices;
by year;
run;
proc sort data = Analysis.F;
by astate acounty;
run;
data Analysis.G;
merge Analysis.F Analysis.corn_business_summary2;
by astate acounty;
run;
data Analysis.H;
set Analysis.G;

priceratio = harvest_price / projected_price;
projected_price2017 = 3.86;
harvest_price2017 = priceratio * projected_price2017;
Indem1 = max (0, ((max (projected_price2017 , harvest_price2017)*
cvglvl* ypred) - (harvest_price2017 * dtrendY)));
Liabl = projected_price2017 * cvglvl * ypred;
Indema = max(0, (projected_price2017*cvglvl*ypred) -
(harvest_price2017*dtrendy));
run;

proc sort data = Analysis.H;
by astate acounty;
run;

proc means data = Analysis.H;
by astate acounty;
var indema;
output out = Analysis.Expindem mean = Expindem;
run;
proc sort data =Analysis.Expindem;
by astate acounty;
run;
data Analysis.SIM_BOS;
merge Analysis.Expindem Analysis.H;
by astate acounty;
run;
proc sort data = Analysis.SIM_BOS;
by astate acounty;
run;

data Analysis.SIM_BOS1;
set analysis.SIM_BOS;
prem = Expindem ;
run;
PROC SORT DATA = Analysis.SIM_BOS1;
by year;

```

```

run;

proc means data = Analysis.SIM_BOS1;
by year;
var prem Indem1 Liab1 Indema;
weight netacr;
output out =Analysis.F_BOS sum =premium indemnity liability indemnitya;
run;

proc sort data = F_bos;
by year;
run;

data Analysis.CP;
set Analysis.Cornprices;
if harvest_price < projected_price then optprm = projected_price -
harvest_price ;
else optprm = 0;
/* tot scale up to a million dollars of liab */
optprm = optprm * (1000000 /3.86);

run;
proc means data = Analysis.CP;
var optprm;
output out = Analysis.optprm mean =avgoptpremium;
run;

data Analysis.Cp1;
set Analysis.cp;

If projected_price > harvest_price then option0 = (0.0 * 1000000 *
(projected_price - harvest_price));
else option0 = 0;

If projected_price > harvest_price then optiona = (0.005 * 1000000 *
(projected_price - harvest_price));
else optiona = 0;

If projected_price > harvest_price then option1 = (0.01 * 1000000 *
(projected_price - harvest_price));
else option1 = 0;

If projected_price > harvest_price then optionb = (0.015 * 1000000 *
(projected_price - harvest_price));
else optionb = 0;

If projected_price > harvest_price then option2 = (0.02 * 1000000 *
(projected_price - harvest_price));
else option2 = 0;

If projected_price > harvest_price then optionc = (0.025 * 1000000 *
(projected_price - harvest_price));
else optionc = 0;

```

```

If projected_price > harvest_price then option3 = (0.03 * 1000000 *
(projected_price - harvest_price));
else option3 = 0;

If projected_price > harvest_price then optiond = (0.035 * 1000000 *
(projected_price - harvest_price));
else optiond = 0;

If projected_price > harvest_price then option4 = (0.04 * 1000000 *
(projected_price - harvest_price));
else option4 = 0;

If projected_price > harvest_price then optione = (0.045 * 1000000 *
(projected_price - harvest_price));
else optione = 0;

If projected_price > harvest_price then option5 = (0.05 * 1000000 *
(projected_price - harvest_price));
else option5 = 0;

If projected_price > harvest_price then optionf = (0.055 * 1000000 *
(projected_price - harvest_price));
else optionf = 0;

If projected_price > harvest_price then option6 = (0.06 * 1000000 *
(projected_price - harvest_price));
else option6 = 0;

If projected_price > harvest_price then optiong = (0.065 * 1000000 *
(projected_price - harvest_price));
else optiong = 0;

If projected_price > harvest_price then option7 = (0.07 * 1000000 *
(projected_price - harvest_price));
else option7 = 0;

If projected_price > harvest_price then optionh = (0.075 * 1000000 *
(projected_price - harvest_price));
else optionh = 0;

If projected_price > harvest_price then option8 = (0.08 * 1000000 *
(projected_price - harvest_price));
else option8 = 0;

If projected_price > harvest_price then optioni = (0.085 * 1000000 *
(projected_price - harvest_price));
else optioni = 0;

```



```

If projected_price > harvest_price then option9 = (0.09 * 1000000 *
(projected_price - harvest_price));
else option9 = 0;

If projected_price > harvest_price then optionj = (0.095 * 1000000 *
(projected_price - harvest_price));
else optionj = 0;

If projected_price > harvest_price then optionl0 = (0.10 * 1000000 *
(projected_price- harvest_price));
else optionl0 = 0;

If projected_price > harvest_price then optionk = (0.105 * 1000000 *
(projected_price- harvest_price));
else optionk = 0;

If projected_price > harvest_price then optionl1 = (0.11 * 1000000 *
(projected_price - harvest_price));
else optionl1 = 0;

If projected_price > harvest_price then optionl = (0.115 * 1000000 *
(projected_price - harvest_price));
else optionl = 0;

If projected_price > harvest_price then optionl2 = (0.12 * 1000000 *
(projected_price- harvest_price));
else optionl2 = 0;

proc sort data = Analysis.Cp1;
by year;

proc means data = Analysis.Cp1;
var option100;
output out = Analysis.opt100 mean = option100;
run;
data Analysis.cm;
merge Analysis.Cp1 analysis.F_BOS;
by year;
run;
proc sort data = Analysis.cm;
by year;
run;
data Analysis.cml;
set Analysis.cm;

if year ne '.';
scalar= 1000000/liability;

premium = premium * scalar;
indemnity = indemnity * scalar;
beginwealth = 1000000;
avgoptpremium = 345070.97;
Endw0 = beginwealth + premium - indemnity +option0-0.0* avgoptpremium;

```

Endwa = beginwealth + premium - indemnity +optiona-0.005\*  
 avgoptpremium;

Endw1 = beginwealth + premium - indemnity +option1-0.01\* avgoptpremium;

Endwb = beginwealth + premium - indemnity +optionb-0.015\*  
 avgoptpremium;

Endw2 = beginwealth + premium - indemnity +option2-0.02\* avgoptpremium;

Endwc = beginwealth + premium - indemnity +optionc-0.025\*  
 avgoptpremium;

Endw3 = beginwealth + premium - indemnity +option3-0.03\* avgoptpremium;

Endwd = beginwealth + premium - indemnity +optiond-0.035\*  
 avgoptpremium;

Endw4 = beginwealth + premium - indemnity +option4-0.04\* avgoptpremium;

Endwe = beginwealth + premium - indemnity +optione-0.045\*  
 avgoptpremium;

Endw5 = beginwealth + premium - indemnity +option5-0.05\* avgoptpremium;

Endwf = beginwealth + premium - indemnity +optionf-0.055\*  
 avgoptpremium;

Endw6 = beginwealth + premium - indemnity +option6-0.06\* avgoptpremium;

Endwg = beginwealth + premium - indemnity +optiong-0.065\*  
 avgoptpremium;

Endw7 = beginwealth + premium - indemnity +option7-0.07\* avgoptpremium;

Endwh = beginwealth + premium - indemnity +optionh-0.075\*  
 avgoptpremium;

Endw8 = beginwealth + premium - indemnity +option8-0.08\* avgoptpremium;

Endwi = beginwealth + premium - indemnity +optioni-0.085\*  
 avgoptpremium;

Endw9 = beginwealth + premium - indemnity +option9-0.09\* avgoptpremium;

Endwj = beginwealth + premium - indemnity +optionj-0.095\*  
 avgoptpremium;

Endw10 = beginwealth +premium - indemnity +option10-0.1\* avgoptpremium;

Endwk = beginwealth + premium - indemnity +optionk-0.105\*  
 avgoptpremium;

Endw11 = beginwealth + premium - indemnity +option11-0.11\*  
 avgoptpremium;

```

Endw1 = beginwealth + premium - indemnity +option1-0.115*
avgoptpremium;

Endw12 = beginwealth + premium - indemnity +option12-0.12*
avgoptpremium;
run;
proc means; title 'ending wealths for RP includng CornBelt and Put
Options';
proc print;
run;

data Analysis.U;
set Analysis.cml;
r_av =1;
If r_av = 1 then utility0 = ( Log (Endw0));
Else utility0 = (((Endw0)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitya = ( Log (Endwa));
Else utilitya = (((Endwa)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility1 = ( Log (Endw1));
Else utility1 = (((Endw1)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityb = ( Log (Endwb));
Else utilityb = (((Endwb)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility2 = ( Log (Endw2));
Else utility2 = (((Endw2)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityc = ( Log (Endwc));
Else utilityc = (((Endwc)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility3 = ( Log (Endw3));
Else utility3 = (((Endw3)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityd = ( Log (Endwd));
Else utilityd = (((Endwd)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility4 = ( Log (Endw4));
Else utility4 = (((Endw4)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitye = ( Log (Endwe));
Else utilitye = (((Endwe)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility5 = ( Log (Endw5));
Else utility5 = (((Endw5)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityf = ( Log (Endwf));
Else utilityf = (((Endwf)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility6 = ( Log (Endw6));
Else utility6 = (((Endw6)** (1- r_av))/ (1-r_av));

```

```

If r_av = 1 then utilityg = ( Log (Endwg));
Else utilityg = (((Endwg)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility7 = ( Log (Endw7));
Else utility7 = (((Endw7)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityh = ( Log (Endwh));
Else utilityh = (((Endwh)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility8 = ( Log (Endw8));
Else utility8 = (((Endw8)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityi = ( Log (Endwi));
Else utilityi = (((Endwi)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility9 = ( Log (Endw9));
Else utility9 = (((Endw9)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityj = ( Log (Endwj));
Else utilityj = (((Endwj)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility10 = ( Log (Endw10));
Else utility10 = (((Endw10)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityk = ( Log (Endwk));
Else utilityk = (((Endwk)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl1 = ( Log (Endwl1));
Else utilityl1 = (((Endwl1)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl = ( Log (Endwl));
Else utilityl = (((Endwl)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl2 = ( Log (Endwl2));
Else utilityl2 = (((Endwl2)** (1- r_av))/ (1-r_av));

run;
proc means data = Analysis.U;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7
utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utilityl1 utilityl utilityl2;

output out = analysis.EU mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utilityl1 utilityl utilityl2;
run;
data Analysis.EU_CE;
set Analysis.EU;
r_av=1;
If r_av=1 then CE0 = exp (utility0);

```

```

Else CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEa = exp (utilitya);
Else CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE1 = exp (utility1);
Else CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEb = exp (utilityb);
Else CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE2 = exp (utility2);
Else CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEc = exp (utilityc);
Else CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE3 = exp (utility3);
Else CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEd = exp (utilityd);
Else CEd = (((utilityd * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE4 = exp (utility4);
Else CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEe = exp (utilitye);
Else CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE5 = exp (utility5);
Else CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEf = exp (utilityf);
Else CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE6 = exp (utility6);
Else CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEg = exp (utilityg);
Else CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE7 = exp (utility7);
Else CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEh = exp (utilityh);
Else CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE8 = exp (utility8);
Else CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEi = exp (utilityi);
Else CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));

```

```

If r_av=1 then CE9 = exp (utility9);
Else CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEj = exp (utilityj);
Else CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE10 = exp (utility10);
Else CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEk = exp (utilityk);
Else CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE11 = exp (utility11);
Else CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE1 = exp (utility1);
Else CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE12 = exp (utility12);
Else CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE13 = exp (utility13);
run;
proc means;
run;

data Analysis.U1;
set Analysis.cml;
r_av =2;
If r_av = 2 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 2 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

If r_av = 2 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 2 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

If r_av = 2 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

If r_av = 2 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 2 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 2 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

```

```

If r_av = 2 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4= ( Log (Endw4));

If r_av = 2 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 2 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5= ( Log (Endw5));

If r_av = 2 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));
Else utilityf = ( Log (Endwf));

If r_av = 2 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 2 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 2 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 2 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 2 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 2 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));
Else utilityi = ( Log (Endwi));

If r_av = 2 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));
Else utility9 = ( Log (Endw9));

If r_av = 2 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 2 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));
Else utility10 = ( Log (Endw10));

If r_av = 2 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

If r_av = 2 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 2 then utilityl = (((Endwl)** (1- r_av))/ (1-r_av));
Else utilityl = ( Log (Endwl));

If r_av = 2 then utility12 = (((Endw12)** (1- r_av))/ (1-r_av));
Else utility12 = ( Log (Endw12));
run;

proc means data = Analysis.U1 ;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7

```

```

utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;
output out = analysis.EU1 mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utilityl utility1;
run;

```

```

data Analysis.EU_CE1;
set Analysis.EU1;
r_av=2;
If r_av=2 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0)));

If r_av=2 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya)));

If r_av=2 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=2 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb)));

If r_av=2 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));
Else CE2 = ((exp (utility2)));

If r_av=2 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc)));

If r_av=2 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3)));

If r_av=2 then CED = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CED = ((exp (utilityd)));

If r_av=2 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));
Else CE4 = ((exp (utility4)));

If r_av=2 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));
Else CEe = ((exp (utilitye)));

If r_av=2 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));
Else CE5 = ((exp (utility5)));

If r_av=2 then CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEf = ((exp (utilityf)));

If r_av=2 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

If r_av=2 then CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));
Else CEg = ((exp (utilityg)));

If r_av=2 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));

```



```

Else CE7 = ((exp (utility7)));

If r_av=2 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=2 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

If r_av=2 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=2 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=2 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=2 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=2 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=2 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

If r_av=2 then CEL = (((utilityl * (1- r_av))**(1 / (1-r_av))));
Else CEL = ((exp (utilityl)));

If r_av=2 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;
proc means;
run;

data Analysis.U2;
set Analysis.cml;
r_av =3;

If r_av = 3 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 3 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

If r_av = 3 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 3 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

If r_av = 3 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

```

```

If r_av = 3 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 3 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 3 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

If r_av = 3 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4 = ( Log (Endw4));

If r_av = 3 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 3 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5 = ( Log (Endw5));

If r_av = 3 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));
Else utilityf = ( Log (Endwf));

If r_av = 3 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 3 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 3 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 3 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 3 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 3 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));
Else utilityi = ( Log (Endwi));

If r_av = 3 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));
Else utility9 = ( Log (Endw9));

If r_av = 3 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 3 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));
Else utility10 = ( Log (Endw10));

If r_av = 3 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

If r_av = 3 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 3 then utilityl = (((Endwl)** (1- r_av))/ (1-r_av));

```

```

Else utilityl = ( Log (Endwl));

If r_av = 3 then utilityl2 = (((Endwl2)** (1- r_av))/ (1-r_av));
Else utilityl2 = ( Log (Endwl2));

run;

proc means data = Analysis.U2 ;
var utility0 utilitya utilityl utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6
utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utilityl1 utilityl
utilityl2;
output out = analysis.EU2 mean= utility0 utilitya utilityl utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utilityl1 utilityl utilityl2;
run;

data Analysis.EU_CE2;
set Analysis.EU2;
r_av=3;

If r_av=3 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0)));

If r_av=3 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya)));

If r_av=3 then CE1 = (((utilityl * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utilityl)));

If r_av=3 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb)));

If r_av=3 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));
Else CE2 = ((exp (utility2)));

If r_av=3 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc)));

If r_av=3 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3)));

If r_av=3 then CED = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CED = ((exp (utilityd)));

If r_av=3 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));
Else CE4 = ((exp (utility4)));

If r_av=3 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));
Else CEe = ((exp (utilitye)));

If r_av=3 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));

```

```

Else CE5 = ((exp (utility5)));

If r_av=3 then CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEf = ((exp (utilityf)));

If r_av=3 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

If r_av=3 then CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));
Else CEg = ((exp (utilityg)));

If r_av=3 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));
Else CE7 = ((exp (utility7)));

If r_av=3 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=3 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

If r_av=3 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=3 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=3 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=3 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=3 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=3 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

If r_av=3 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=3 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;

proc means;
run;
proc sort data = Analysis.cp1;
by year;
run;
proc sort data = Analysis.F_bos;
by year;
run;
data Analysis.cor (keep = year indemnity option100);
merge analysis.cp1 analysis.F_bos;

```

```

by year;
if year ne'.';
run;
proc corr data = analysis.cor;
var option100 indemnity;
run;

```

## Put options Including Corn belts hedging the price risk of RP-HPE

```

libname analysis 'C:\Users\sweta\Desktop\data1a';
proc import out= Analysis.data1985 datafile =
"C:\Users\sweta\Desktop\data1a\cleaned1985.xlsx"
DBMS= xlsx;
sheet = "cleaned1985";
GETNAMES= YES;
run;

proc sort data=Analysis.data1985;
by year;
proc import out= Analysis.cornprices datafile =
"C:\Users\sweta\Desktop\data1a\corn_crop_futures_summary3.xlsx"
DBMS= xlsx;
sheet = "sheet3";
GETNAMES= YES;
run;
proc sort data=Analysis.cornprices;
by year;

proc import out= Analysis.corn_business_summary datafile =
"C:\Users\sweta\Desktop\data1a\Required85_16.xlsx"
DBMS= xlsx;
sheet = "Required85_16";
GETNAMES= YES;
run;
proc sort data=Analysis.corn_business_summary;
by astate acounty;
data Analysis.corn_business_summary1;
set Analysis.corn_business_summary;
if insplan = 2;
run;
/*WEIGHTED COVERAGE LEVEL by county and state*/
proc means data = analysis.corn_business_summary1 ;
by astate acounty;
var cvglvl;
weight netacr;
output out= analysis.current_mean_county_covg MEAN=cvglvl;
run;
proc sort data = analysis.corn_business_summary1;
by astate acounty;
run;

```

```

    data Analysis.corn_business_summary2 (keep = astate acounty
netacr);
    set Analysis.corn_business_summary1;
    run;
    proc sort data = analysis.corn_business_summary2;
    by astate acounty;
    run;

proc sort data=Analysis.current_mean_county_covg;
by astate acounty;
run;
/*Simulation for yield*/
proc sort data = Analysis.Data1985;
by astate acounty;
run;
proc reg noprint data=Analysis.Data1985 outest= Analysis.Regout;
/*Saving Intercept and Parameter*/
model value = year; /*Linear trend for each county*/
by astate acounty;
output out= Analysis.b /*Saving Residuals*/
p = vhat
r= vresid;
run;
proc sort data = Analysis.b;
by astate acounty;
run;
data Analysis.pred;
set Analysis.regout;
ypred = Intercept + year*2017; /*Simulation for yield guarantee by
county*/
run;
proc sort data = Analysis.pred;
by astate acounty;
run;

data Analysis.C (keep = year astate acounty intercept year ypred
vresid);
merge Analysis.pred Analysis.b;
by astate acounty;
run;
proc sort data =Analysis.C;
by astate acounty;
run;
data Analysis.D;
set Analysis.C;
dtrendY = ypred + vresid * 1.66; /*Simulation for average
representative farm yield*/
If dtrendY < 0 then dtrendY = 0;
run;
proc sort data = Analysis.D;
by astate acounty;
run;
data Analysis.E;
merge Analysis.current_mean_county_covg Analysis.D;

```

```

by astate acounty;
run;
proc sort data = Analysis.E;
by year;
Data Analysis.F;
merge Analysis.E Analysis.Cornprices;
by year;
run;
proc sort data = Analysis.F;
by astate acounty;
run;
data Analysis.G;
merge Analysis.F Analysis.corn_business_summary2;
by astate acounty;
run;
data Analysis.H;
set Analysis.G;

priceratio = harvest_price / projected_price;
projected_price2017 = 3.86;
harvest_price2017 = priceratio * projected_price2017;
Indem1 = max (0, ((max (projected_price2017 , harvest_price2017)*
cvglvl* ypred) - (harvest_price2017 * dtrendY)));
Liabl = projected_price2017 * cvglvl * ypred;
Indema = max(0,(projected_price2017*cvglvl*ypred) -
(harvest_price2017*dtrendy));
run;

proc sort data = Analysis.H;
by astate acounty;
run;

proc means data = Analysis.H;
by astate acounty;
var indema;
output out = Analysis.Expindem mean = Expindem;
run;
proc sort data =Analysis.Expindem;
by astate acounty;
run;
data Analysis.SIM_BOS;
merge Analysis.Expindem Analysis.H;
by astate acounty;
run;
proc sort data = Analysis.SIM_BOS;
by astate acounty;
run;

data Analysis.SIM_BOS1;
set analysis.SIM_BOS;
prem = Expindem ;
run;
PROC SORT DATA = Analysis.SIM_BOS1;
by year;
run;

```

```

proc means data = Analysis.SIM_BOS1;
by year;
var prem Indem1 Liab1 Indema;
weight netacr;
output out =Analysis.F_BOS sum =premium indemnity liability indemnitya;
run;

proc sort data = F_bos;
by year;
run;
proc sort data = Analysis.SIM_BOS1;
by astate;
run;
proc means data = Analysis.SIM_BOS1;
by astate;
var prem Indem1 Liab1 Indema;
weight netacr;
output out =Analysis.F_BOS1 sum =premium indemnity liability
indemnitya;
run;

data Analysis.CP;
set Analysis.Cornprices;
if harvest_price < projected_price then optprm = projected_price -
harvest_price ;
else optprm = 0;
/* tot scale up to a million dollars of liab */
optprm = optprm * (1000000 /3.86);

run;
proc means data = Analysis.CP;
var optprm;
output out = Analysis.optprm mean =avgoptpremium;
run;

data Analysis.Cp1;
set Analysis.cp;

If projected_price > harvest_price then option0 = (0.0 * 1000000 *
(projected_price - harvest_price));
else option0 = 0;

If projected_price > harvest_price then optiona = (0.005 * 1000000 *
(projected_price - harvest_price));
else optiona = 0;

If projected_price > harvest_price then option1 = (0.01 * 1000000 *
(projected_price - harvest_price));
else option1 = 0;

If projected_price > harvest_price then optionb = (0.015 * 1000000 *
(projected_price - harvest_price));
else optionb = 0;

```



```

If projected_price > harvest_price then option2 = (0.02 * 1000000 *
(projected_price - harvest_price));
else option2 = 0;

If projected_price > harvest_price then optionc = (0.025 * 1000000 *
(projected_price - harvest_price));
else optionc = 0;

If projected_price > harvest_price then option3 = (0.03 * 1000000 *
(projected_price - harvest_price));
else option3 = 0;

If projected_price > harvest_price then optiond = (0.035 * 1000000 *
(projected_price - harvest_price));
else optiond = 0;

If projected_price > harvest_price then option4 = (0.04 * 1000000 *
(projected_price - harvest_price));
else option4 = 0;

If projected_price > harvest_price then optione = (0.045 * 1000000 *
(projected_price - harvest_price));
else optione = 0;

If projected_price > harvest_price then option5 = (0.05 * 1000000 *
(projected_price - harvest_price));
else option5 = 0;

If projected_price > harvest_price then optionf = (0.055 * 1000000 *
(projected_price - harvest_price));
else optionf = 0;

If projected_price > harvest_price then option6 = (0.06 * 1000000 *
(projected_price - harvest_price));
else option6 = 0;

If projected_price > harvest_price then optiong = (0.065 * 1000000 *
(projected_price - harvest_price));
else optiong = 0;

If projected_price > harvest_price then option7 = (0.07 * 1000000 *
(projected_price - harvest_price));
else option7 = 0;

If projected_price > harvest_price then optionh = (0.075 * 1000000 *
(projected_price - harvest_price));
else optionh = 0;

```

```

If projected_price > harvest_price then option8 = (0.08 * 1000000 *
(projected_price - harvest_price));
else option8 = 0;

If projected_price > harvest_price then optioni = (0.085 * 1000000 *
(projected_price - harvest_price));
else optioni = 0;

If projected_price > harvest_price then option9 = (0.09 * 1000000 *
(projected_price - harvest_price));
else option9 = 0;

If projected_price > harvest_price then optionj = (0.095 * 1000000 *
(projected_price - harvest_price));
else optionj = 0;

If projected_price > harvest_price then option10 = (0.10 * 1000000 *
(projected_price- harvest_price));
else option10 = 0;

If projected_price > harvest_price then optionk = (0.105 * 1000000 *
(projected_price- harvest_price));
else optionk = 0;

If projected_price > harvest_price then optionl1 = (0.11 * 1000000 *
(projected_price - harvest_price));
else optionl1 = 0;

If projected_price > harvest_price then optionl = (0.115 * 1000000 *
(projected_price - harvest_price));
else optionl = 0;

If projected_price > harvest_price then option12 = (0.12 * 1000000 *
(projected_price- harvest_price));
else option12 = 0;

proc sort data = Analysis.Cp1;
by year;

proc means data = Analysis.Cp1;
var option100;
output out = Analysis.opt100 mean = option100;
run;
data Analysis.cm;
merge Analysis.Cp1 analysis.F_BOS;
by year;
run;
proc sort data = Analysis.cm;
by year;
run;
data Analysis.cm1;
set Analysis.cm;

if year ne '.';

```

```

scalar= 1000000/liability;

premium = premium * scalar;
indemnity = indemnity * scalar;
beginwealth = 1000000;
avgoptpremium = 345070.97;
Endw0 = beginwealth + premium - indemnitya +option0-0.0* avgoptpremium;

Endwa = beginwealth + premium - indemnitya +optiona-0.005*
avgoptpremium;

Endw1 = beginwealth + premium - indemnitya +option1-0.01*
avgoptpremium;

Endwb = beginwealth + premium - indemnitya +optionb-0.015*
avgoptpremium;

Endw2 = beginwealth + premium - indemnitya +option2-0.02*
avgoptpremium;

Endwc = beginwealth + premium - indemnitya +optionc-0.025*
avgoptpremium;

Endw3 = beginwealth + premium - indemnitya +option3-0.03*
avgoptpremium;

Endwd = beginwealth + premium - indemnitya +optiond-0.035*
avgoptpremium;

Endw4 = beginwealth + premium - indemnitya +option4-0.04*
avgoptpremium;

Endwe = beginwealth + premium - indemnitya +optione-0.045*
avgoptpremium;

Endw5 = beginwealth + premium - indemnitya +option5-0.05*
avgoptpremium;

Endwf = beginwealth + premium - indemnitya +optionf-0.055*
avgoptpremium;

Endw6 = beginwealth + premium - indemnitya +option6-0.06*
avgoptpremium;

Endwg = beginwealth + premium - indemnitya +optiong-0.065*
avgoptpremium;

Endw7 = beginwealth + premium - indemnitya +option7-0.07*
avgoptpremium;

Endwh = beginwealth + premium - indemnitya +optionh-0.075*
avgoptpremium;

Endw8 = beginwealth + premium - indemnitya +option8-0.08*
avgoptpremium;

```

```

Endwi = beginwealth + premium - indemnitya +optioni-0.085*
avgoptpremium;

Endw9 = beginwealth + premium - indemnitya +option9-0.09*
avgoptpremium;

Endwj = beginwealth + premium - indemnitya +optionj-0.095*
avgoptpremium;

Endwl0 = beginwealth +premium - indemnitya +optionl0-0.1*
avgoptpremium;

Endwk = beginwealth + premium - indemnitya +optionk-0.105*
avgoptpremium;

Endwl1 = beginwealth + premium - indemnitya +optionl1-0.11*
avgoptpremium;

Endwl = beginwealth + premium - indemnitya +optionl-0.115*
avgoptpremium;

Endwl2 = beginwealth + premium - indemnitya +optionl2-0.12*
avgoptpremium;
run;
proc means; title 'ending wealths for RP incluindg CornBelt and Put
Options';
proc print;
run;

data Analysis.U;
set Analysis.cml;
r_av =1;
If r_av = 1 then utility0 = ( Log (Endw0));
Else utility0 = (((Endw0)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitya = ( Log (Endwa));
Else utilitya = (((Endwa)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility1 = ( Log (Endwl));
Else utility1 = (((Endwl)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityb = ( Log (Endwb));
Else utilityb = (((Endwb)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility2 = ( Log (Endw2));
Else utility2 = (((Endw2)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityc = ( Log (Endwc));
Else utilityc = (((Endwc)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility3 = ( Log (Endw3));
Else utility3 = (((Endw3)** (1- r_av))/ (1-r_av));

```

```

If r_av = 1 then utilityd = ( Log (Endwd));
Else utilityd = (((Endwd)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility4 = ( Log (Endw4));
Else utility4 = (((Endw4)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitye = ( Log (Endwe));
Else utilitye = (((Endwe)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility5 = ( Log (Endw5));
Else utility5 = (((Endw5)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityf = ( Log (Endwf));
Else utilityf = (((Endwf)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility6 = ( Log (Endw6));
Else utility6 = (((Endw6)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityg = ( Log (Endwg));
Else utilityg = (((Endwg)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility7 = ( Log (Endw7));
Else utility7 = (((Endw7)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityh = ( Log (Endwh));
Else utilityh = (((Endwh)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility8 = ( Log (Endw8));
Else utility8 = (((Endw8)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityi = ( Log (Endwi));
Else utilityi = (((Endwi)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility9 = ( Log (Endw9));
Else utility9 = (((Endw9)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityj = ( Log (Endwj));
Else utilityj = (((Endwj)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility10 = ( Log (Endw10));
Else utility10 = (((Endw10)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityk = ( Log (Endwk));
Else utilityk = (((Endwk)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl1 = ( Log (Endwl1));
Else utilityl1 = (((Endwl1)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl = ( Log (Endwl));
Else utilityl = (((Endwl)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility12 = ( Log (Endwl2));

```

```

Else utility12 = (((Endw12)** (1- r_av))/ (1-r_av));

run;
proc means data = Analysis.U;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7
utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;

output out = analysis.EU mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utility11 utilityl utility12;
run;
data Analysis.EU_CE;
set Analysis.EU;
r_av=1;
If r_av=1 then CE0 = exp (utility0);
Else CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEa = exp (utilitya);
Else CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE1 = exp (utility1);
Else CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEb = exp (utilityb);
Else CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE2 = exp (utility2);
Else CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEc = exp (utilityc);
Else CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE3 = exp (utility3);
Else CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE4 = exp (utility4);
Else CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEe = exp (utilitye);
Else CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE5 = exp (utility5);
Else CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEf = exp (utilityf);
Else CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));

```

```

If r_av=1 then CE6 = exp (utility6);
Else CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEg = exp (utilityg);
Else CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE7 = exp (utility7);
Else CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEh = exp (utilityh);
Else CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE8 = exp (utility8);
Else CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEi = exp (utilityi);
Else CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE9 = exp (utility9);
Else CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEj = exp (utilityj);
Else CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE10 = exp (utility10);
Else CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEk = exp (utilityk);
Else CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE11 = exp (utility11);
Else CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEL = exp (utilityl);
Else CEL = (((utilityl * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE12 = exp (utility12);
Else CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE13 = exp (utility13);
run;
proc means;
run;

data Analysis.U1;
set Analysis.cml;
r_av =2;
If r_av = 2 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 2 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

```

```

If r_av = 2 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 2 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

If r_av = 2 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

If r_av = 2 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 2 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 2 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

If r_av = 2 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4= ( Log (Endw4));

If r_av = 2 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 2 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5= ( Log (Endw5));

If r_av = 2 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));
Else utilityf = ( Log (Endwf));

If r_av = 2 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 2 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 2 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 2 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 2 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 2 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));
Else utilityi = ( Log (Endwi));

If r_av = 2 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));
Else utility9 = ( Log (Endw9));

If r_av = 2 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 2 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));

```



```

Else utility10 = ( Log (Endw10));

If r_av = 2 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

If r_av = 2 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 2 then utilityl = (((Endwl)** (1- r_av))/ (1-r_av));
Else utilityl = ( Log (Endwl));

If r_av = 2 then utility12 = (((Endw12)** (1- r_av))/ (1-r_av));
Else utility12 = ( Log (Endw12));
run;

proc means data = Analysis.U1 ;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7
utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;
output out = analysis.EU1 mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utility11 utilityl;
run;

data Analysis.EU_CE1;
set Analysis.EU1;
r_av=2;
If r_av=2 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0));

If r_av=2 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya));

If r_av=2 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1));

If r_av=2 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb));

If r_av=2 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));
Else CE2 = ((exp (utility2));

If r_av=2 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc));

If r_av=2 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3));

If r_av=2 then CED = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CED = ((exp (utilityd));

If r_av=2 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));

```

```

Else CE4 = ((exp (utility4)));

If r_av=2 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));
Else CEe = ((exp (utilitye)));

If r_av=2 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));
Else CE5 = ((exp (utility5)));

If r_av=2 then CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEf = ((exp (utilityf)));

If r_av=2 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

If r_av=2 then CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));
Else CEg = ((exp (utilityg)));

If r_av=2 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));
Else CE7 = ((exp (utility7)));

If r_av=2 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=2 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

If r_av=2 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=2 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=2 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=2 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=2 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=2 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

If r_av=2 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=2 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;
proc means;
run;

data Analysis.U2;

```

```

set Analysis.cml;
r_av =3;

If r_av = 3 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 3 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

If r_av = 3 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 3 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

If r_av = 3 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

If r_av = 3 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 3 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 3 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

If r_av = 3 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4 = ( Log (Endw4));

If r_av = 3 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 3 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5 = ( Log (Endw5));

If r_av = 3 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));
Else utilityf = ( Log (Endwf));

If r_av = 3 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 3 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 3 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 3 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 3 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 3 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));

```

```

Else utilityi = ( Log (Endwi));

If r_av = 3 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));
Else utility9 = ( Log (Endw9));

If r_av = 3 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 3 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));
Else utility10 = ( Log (Endw10));

If r_av = 3 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

If r_av = 3 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 3 then utilityl = (((Endwl)** (1- r_av))/ (1-r_av));
Else utilityl = ( Log (Endwl));

If r_av = 3 then utility12 = (((Endw12)** (1- r_av))/ (1-r_av));
Else utility12 = ( Log (Endw12));

run;

proc means data = Analysis.U2 ;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6
utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utility11 utilityl
utility12;
output out = analysis.EU2 mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utility11 utilityl utility12;
run;

data Analysis.EU_CE2;
set Analysis.EU2;
r_av=3;

If r_av=3 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0)));

If r_av=3 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya)));

If r_av=3 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=3 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb)));

If r_av=3 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));

```

```

Else CE2 = ((exp (utility2)));

If r_av=3 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc)));

If r_av=3 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3)));

If r_av=3 then CEd = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CEd = ((exp (utilityd)));

If r_av=3 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));
Else CE4 = ((exp (utility4)));

If r_av=3 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));
Else CEe = ((exp (utilitye)));

If r_av=3 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));
Else CE5 = ((exp (utility5)));

If r_av=3 then CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEf = ((exp (utilityf)));

If r_av=3 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

If r_av=3 then CEG = (((utilityg * (1- r_av))**(1 / (1-r_av))));
Else CEG = ((exp (utilityg)));

If r_av=3 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));
Else CE7 = ((exp (utility7)));

If r_av=3 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=3 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

If r_av=3 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=3 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=3 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=3 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=3 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=3 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

```

```

If r_av=3 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=3 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;

run;
proc means;
run;
proc sort data = Analysis.cp1;
by year;
run;
proc sort data = Analysis.F_bos;
by year;
run;
data Analysis.cor (keep = year indemnitya option100);
merge analysis.cp1 analysis.F_bos;
by year;
if year ne '.';
run;
proc corr data = analysis.cor;
var option100 indemnitya;
run;

```

The analysis conducted for put options to hedge the price risk of corn RP and RP-HPE sold in the region excluding the corn belts, same codes were used but excluded the five major corn belt states, Indiana, Ohio, Iowa, Illinois and Missouri from the main datasets, cleaned 1985 and required85\_16.

### Call Options Including Corn Belts Hedging the Price Risk of Corn RP

```

libname analysis 'C:\Users\sweta\Desktop\data6';
proc import out= Analysis.data1985 datafile =
"C:\Users\sweta\Desktop\data6\cleaned1985.xlsx"
DBMS= xlsx;
sheet = "cleaned1985";
GETNAMES= YES;
run;

proc sort data=Analysis.data1985;
by year;
proc import out= Analysis.cornprices datafile =
"C:\Users\sweta\Desktop\data6\corn_crop_futures summary3.xlsx"

```

```

DBMS= xlsx;
sheet = "sheet3";
GETNAMES= YES;
run;
proc sort data=Analysis.cornprices;
by year;

proc import out= Analysis.corn_business_summary datafile =
"C:\Users\sweta\Desktop\data6\Required85_16.xlsx"
DBMS= xlsx;
sheet = "Required85_16";
GETNAMES= YES;
run;
proc sort data=Analysis.corn_business_summary;
by astate acounty;
data Analysis.corn_business_summary1;
set Analysis.corn_business_summary;
if insplan = 2;
run;
/*WEIGHTED COVERAGE LEVEL by county and state*/
proc means data = analysis.corn_business_summary1 ;
by astate acounty;
var cvglvl;
weight netacr;
output out= analysis.current_mean_county_covg MEAN=cvglvl;
run;
proc sort data = analysis.corn_business_summary1;
by astate acounty;
run;
data Analysis.corn_business_summary2 (keep = astate acounty
netacr);
set Analysis.corn_business_summary1;
run;
proc sort data = analysis.corn_business_summary2;
by astate acounty;
run;

proc sort data=Analysis.current_mean_county_covg;
by astate acounty;
run;
/*Simulation for yield*/
proc sort data = Analysis.Data1985;
by astate acounty;
run;
proc reg noprint data=Analysis.Data1985 outest= Analysis.Regout;
/*Saving Intercept and Parameter*/
model value = year; /*Linear trend for each county*/
by astate acounty;
output out= Analysis.b /*Saving Residuals*/
p = vhat
r= vresid;
run;
proc sort data = Analysis.b;

```

```

by astate acounty;
run;
data Analysis.pred;
set Analysis.regout;
ypred = Intercept + year*2017; /*Simulation for yield guarantee by
county*/
run;
proc sort data = Analysis.pred;
by astate acounty;
run;

data Analysis.C (keep = year astate acounty intercept year ypred
vresid);
merge Analysis.pred Analysis.b;
by astate acounty;
run;
proc sort data =Analysis.C;
by astate acounty;
run;
data Analysis.D;
set Analysis.C;
dtrendY = ypred + vresid * 1.66; /*Simulation for average
representative farm yield*/
If dtrendY < 0 then dtrendY = 0;
run;
proc sort data = Analysis.D;
by astate acounty;
run;
data Analysis.E;
merge Analysis.current_mean_county_covg Analysis.D;
by astate acounty;
run;
proc sort data = Analysis.E;
by year;
Data Analysis.F;
merge Analysis.E Analysis.Cornprices;
by year;
run;
proc sort data = Analysis.F;
by astate acounty;
run;
data Analysis.G;
merge Analysis.F Analysis.corn_business_summary2;
by astate acounty;
run;
data Analysis.H;
set Analysis.G;

priceratio = harvest_price / projected_price;
projected_price2017 = 3.86;
harvest_price2017 = priceratio * projected_price2017;
Indem1 = max (0, ((max (projected_price2017 , harvest_price2017)*
cvglvl* ypred) - (harvest_price2017 * dtrendY)));
Liab1 = projected_price2017 * cvglvl * ypred;

```



```

Indema = max(0,(projected_price2017*cvglvl*ypred) -
(harvest_price2017*dtrendy));
run;

proc sort data = Analysis.H;
by astate acounty;
run;

proc means data = Analysis.H;
by astate acounty;
var indema;
output out = Analysis.Expindem mean = Expindem;
run;
proc sort data =Analysis.Expindem;
by astate acounty;
run;
data Analysis.SIM_BOS;
merge Analysis.Expindem Analysis.H;
by astate acounty;
run;
proc sort data = Analysis.SIM_BOS;
by astate acounty;
run;

data Analysis.SIM_BOS1;
set analysis.SIM_BOS;
prem = Expindem ;
run;
PROC SORT DATA = Analysis.SIM_BOS1;
by year;
run;

proc means data = Analysis.SIM_BOS1;
by year;
var prem Indem1 Liab1 Indema;
weight netacr;
output out =Analysis.F_BOS sum =premium indemnity liability indemnitya;
run;

proc sort data = Analysis.F_bos;
by year;
run;

data Analysis.CP;
set Analysis.Cornprices;
if harvest_price > projected_price then optprm = projected_price -
harvest_price ;
else optprm = 0;
/* tot scale up to a million dollars of liab */
optprm = optprm * (1000000 /3.86);

run;
proc means data = Analysis.CP;
var optprm;

```

```
output out = Analysis.optprm mean =avgoptpremium;  
run;
```

```
data Analysis.Cp1;  
set Analysis.cp;
```

```
If projected_price < harvest_price then option0 = (0.0 * 1000000 *  
(harvest_price - projected_price));  
else option0 = 0;
```

```
If projected_price < harvest_price then optiona = (0.005 * 1000000 *  
(harvest_price - projected_price));  
else optiona = 0;
```

```
If projected_price < harvest_price then option1 = (0.01 * 1000000 *  
(harvest_price - projected_price));  
else option1 = 0;
```

```
If projected_price < harvest_price then optionb = (0.015 * 1000000 *  
(harvest_price - projected_price));  
else optionb = 0;
```

```
If projected_price < harvest_price then option2 = (0.02 * 1000000 *  
(harvest_price - projected_price));  
else option2 = 0;
```

```
If projected_price < harvest_price then optionc = (0.025 * 1000000 *  
(harvest_price - projected_price));  
else optionc = 0;
```

```
If projected_price < harvest_price then option3 = (0.03 * 1000000 *  
(harvest_price - projected_price));  
else option3 = 0;
```

```
If projected_price < harvest_price then optiond = (0.035 * 1000000 *  
(harvest_price - projected_price));  
else optiond = 0;
```

```
If projected_price < harvest_price then option4 = (0.04 * 1000000 *  
(harvest_price - projected_price));  
else option4 = 0;
```

```
If projected_price < harvest_price then optione = (0.045 * 1000000 *  
(harvest_price - projected_price));  
else optione = 0;
```

```

If projected_price < harvest_price then option5 = (0.05 * 1000000 *
(harvest_price - projected_price));
else option5 = 0;

If projected_price < harvest_price then optionf = (0.055 * 1000000 *
(harvest_price - projected_price));
else optionf = 0;

If projected_price < harvest_price then option6 = (0.06 * 1000000 *
(harvest_price - projected_price));
else option6 = 0;

If projected_price < harvest_price then optiong = (0.065 * 1000000 *
(harvest_price - projected_price));
else optiong = 0;

If projected_price < harvest_price then option7 = (0.07 * 1000000 *
(harvest_price - projected_price));
else option7 = 0;

If projected_price < harvest_price then optionh = (0.075 * 1000000 *
(harvest_price - projected_price));
else optionh = 0;

If projected_price < harvest_price then option8 = (0.08 * 1000000 *
(harvest_price - projected_price));
else option8 = 0;

If projected_price < harvest_price then optioni = (0.085 * 1000000 *
(harvest_price - projected_price));
else optioni = 0;

If projected_price < harvest_price then option9 = (0.09 * 1000000 *
(harvest_price - projected_price));
else option9 = 0;

If projected_price < harvest_price then optionj = (0.095 * 1000000 *
(harvest_price - projected_price));
else optionj = 0;

If projected_price < harvest_price then option10 = (0.10 * 1000000 *
(harvest_price - projected_price));
else option10 = 0;

If projected_price < harvest_price then optionk = (0.105 * 1000000 *
(harvest_price - projected_price));
else optionk = 0;

```

```

If projected_price < harvest_price then option11 = (0.11 * 1000000 *
(harvest_price - projected_price));
else option11 = 0;

If projected_price < harvest_price then option1 = (0.115 * 1000000 *
(harvest_price - projected_price));
else option1 = 0;

If projected_price < harvest_price then option12 = (0.12 * 1000000 *
(harvest_price - projected_price));
else option12 = 0;

run;

proc sort data = Analysis.Cp1;
by year;

proc means data = Analysis.Cp1;
var option100;
output out = Analysis.opt100 mean = option100;
run;
data Analysis.cm;
merge Analysis.Cp1 analysis.F_BOS;
by year;
run;
proc sort data = Analysis.cm;
by year;
run;
data Analysis.cml;
set Analysis.cm;

if year ne '.';
scalar= 1000000/liability;

premium = premium * scalar;
indemnity = indemnity * scalar;
beginwealth = 1000000;
avgoptpremium = 191509.68;

Endw0 = beginwealth + premium - indemnity +option0-0.0* avgoptpremium;

Endwa = beginwealth + premium - indemnity +optiona-0.005*
avgoptpremium;

Endw1 = beginwealth + premium - indemnity +option1-0.01* avgoptpremium;

Endwb = beginwealth + premium - indemnity +optionb-0.015*
avgoptpremium;

Endw2 = beginwealth + premium - indemnity +option2-0.02* avgoptpremium;

```

```

Endwc = beginwealth + premium - indemnity +optionc-0.025*
avgoptpremium;

Endw3 = beginwealth + premium - indemnity +option3-0.03* avgoptpremium;

Endwd = beginwealth + premium - indemnity +optiond-0.035*
avgoptpremium;

Endw4 = beginwealth + premium - indemnity +option4-0.04* avgoptpremium;

Endwe = beginwealth + premium - indemnity +optione-0.045*
avgoptpremium;

Endw5 = beginwealth + premium - indemnity +option5-0.05* avgoptpremium;

Endwf = beginwealth + premium - indemnity +optionf-0.055*
avgoptpremium;

Endw6 = beginwealth + premium - indemnity +option6-0.06* avgoptpremium;

Endwg = beginwealth + premium - indemnity +optiong-0.065*
avgoptpremium;

Endw7 = beginwealth + premium - indemnity +option7-0.07* avgoptpremium;

Endwh = beginwealth + premium - indemnity +optionh-0.075*
avgoptpremium;

Endw8 = beginwealth + premium - indemnity +option8-0.08* avgoptpremium;

Endwi = beginwealth + premium - indemnity +optioni-0.085*
avgoptpremium;

Endw9 = beginwealth + premium - indemnity +option9-0.09* avgoptpremium;

Endwj = beginwealth + premium - indemnity +optionj-0.095*
avgoptpremium;

Endw10 = beginwealth +premium - indemnity +option10-0.1* avgoptpremium;

Endwk = beginwealth + premium - indemnity +optionk-0.105*
avgoptpremium;

Endwl1 = beginwealth + premium - indemnity +optionl1-0.11*
avgoptpremium;

Endwl = beginwealth + premium - indemnity +optionl-0.115*
avgoptpremium;

Endw12 = beginwealth + premium - indemnity +option12-0.12*
avgoptpremium;
run;

proc means; title 'ending wealths including Corn Belt for Call Options
and RP';

```

```

proc print;
run;

data Analysis.U;
set Analysis.cml;
r_av =1;

If r_av = 1 then utility0 = ( Log (Endw0));
Else utility0 = (((Endw0)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitya = ( Log (Endwa));
Else utilitya = (((Endwa)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility1 = ( Log (Endw1));
Else utility1 = (((Endw1)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityb = ( Log (Endwb));
Else utilityb = (((Endwb)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility2 = ( Log (Endw2));
Else utility2 = (((Endw2)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityc = ( Log (Endwc));
Else utilityc = (((Endwc)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility3 = ( Log (Endw3));
Else utility3 = (((Endw3)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityd = ( Log (Endwd));
Else utilityd = (((Endwd)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility4 = ( Log (Endw4));
Else utility4 = (((Endw4)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitye = ( Log (Endwe));
Else utilitye = (((Endwe)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility5 = ( Log (Endw5));
Else utility5 = (((Endw5)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityf = ( Log (Endwf));
Else utilityf = (((Endwf)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility6 = ( Log (Endw6));
Else utility6 = (((Endw6)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityg = ( Log (Endwg));
Else utilityg = (((Endwg)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility7 = ( Log (Endw7));
Else utility7 = (((Endw7)** (1- r_av))/ (1-r_av));

```

```

If r_av = 1 then utilityh = ( Log (Endwh));
Else utilityh = (((Endwh)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility8 = ( Log (Endw8));
Else utility8 = (((Endw8)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityi = ( Log (Endwi));
Else utilityi = (((Endwi)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility9 = ( Log (Endw9));
Else utility9 = (((Endw9)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityj = ( Log (Endwj));
Else utilityj = (((Endwj)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility10 = ( Log (Endw10));
Else utility10 = (((Endw10)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityk = ( Log (Endwk));
Else utilityk = (((Endwk)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl1 = ( Log (Endwl1));
Else utilityl1 = (((Endwl1)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl = ( Log (Endwl));
Else utilityl = (((Endwl)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility12 = ( Log (Endwl2));
Else utility12 = (((Endwl2)** (1- r_av))/ (1-r_av));

proc means data = Analysis.U;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7
utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utilityl1 utilityl utility12;

output out = analysis.EU mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utilityl1 utilityl utility12;
run;
data Analysis.EU_CE;
set Analysis.EU;
r_av=1;
If r_av=1 then CE0 = exp (utility0);
Else CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEa = exp (utilitya);
Else CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE1 = exp (utility1);
Else CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));

```

```

If r_av=1 then CEb = exp (utilityb);
Else CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE2 = exp (utility2);
Else CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEc = exp (utilityc);
Else CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE3 = exp (utility3);
Else CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEd = exp (utilityd);
Else CEd = (((utilityd * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE4 = exp (utility4);
Else CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEe = exp (utilitye);
Else CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE5 = exp (utility5);
Else CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEf = exp (utilityf);
Else CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE6 = exp (utility6);
Else CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEg = exp (utilityg);
Else CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE7 = exp (utility7);
Else CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEh = exp (utilityh);
Else CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE8 = exp (utility8);
Else CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEi = exp (utilityi);
Else CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE9 = exp (utility9);
Else CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEj = exp (utilityj);
Else CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE10 = exp (utility10);

```



```

Else CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEk = exp (utilityk);
Else CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE11 = exp (utility11);
Else CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEL = exp (utilityl);
Else CEL = (((utilityl * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE12 = exp (utility12);
Else CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
run;
proc means;
run;

data Analysis.U1;
set Analysis.cml;
r_av =2;
If r_av = 2 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 2 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

If r_av = 2 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 2 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

If r_av = 2 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

If r_av = 2 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 2 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 2 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

If r_av = 2 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4= ( Log (Endw4));

If r_av = 2 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 2 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5= ( Log (Endw5));

If r_av = 2 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));

```

```

Else utilityf = ( Log (Endwf));

If r_av = 2 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 2 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 2 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 2 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 2 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 2 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));
Else utilityi = ( Log (Endwi));

If r_av = 2 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));
Else utility9 = ( Log (Endw9));

If r_av = 2 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 2 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));
Else utility10 = ( Log (Endw10));

If r_av = 2 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

If r_av = 2 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 2 then utilityl = (((Endwl)** (1- r_av))/ (1-r_av));
Else utilityl = ( Log (Endwl));

If r_av = 2 then utility12 = (((Endw12)** (1- r_av))/ (1-r_av));
Else utility12 = ( Log (Endw12));

run;

proc means data = Analysis.U1 ;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6
utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utility11 utilityl utility12;
output out = analysis.EU1 mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4
utilitye utility5 utilityf utility6 utilityg utility7 utilityh utility8
utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;
run;

```

```

data Analysis.EU_CE1;
set Analysis.EU1;
r_av=2;
If r_av=2 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0)));

If r_av=2 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya)));

If r_av=2 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=2 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb)));

If r_av=2 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));
Else CE2 = ((exp (utility2)));

If r_av=2 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc)));

If r_av=2 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3)));

If r_av=2 then CED = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CED = ((exp (utilityd)));

If r_av=2 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));
Else CE4 = ((exp (utility4)));

If r_av=2 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));
Else CEe = ((exp (utilitye)));

If r_av=2 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));
Else CE5 = ((exp (utility5)));

If r_av=2 then CEF = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEF = ((exp (utilityf)));

If r_av=2 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

If r_av=2 then CEG = (((utilityg * (1- r_av))**(1 / (1-r_av))));
Else CEG = ((exp (utilityg)));

If r_av=2 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));
Else CE7 = ((exp (utility7)));

If r_av=2 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=2 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

```

```

If r_av=2 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=2 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=2 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=2 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=2 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=2 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

If r_av=2 then CEL = (((utilityl * (1- r_av))**(1 / (1-r_av))));
Else CEL = ((exp (utilityl)));

If r_av=2 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;
proc means;
run;

data Analysis.U2;
set Analysis.cml;
r_av =3;

If r_av = 3 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 3 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

If r_av = 3 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 3 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

If r_av = 3 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

If r_av = 3 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 3 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 3 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

```

```

If r_av = 3 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4 = ( Log (Endw4));

If r_av = 3 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 3 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5 = ( Log (Endw5));

If r_av = 3 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));
Else utilityf = ( Log (Endwf));

If r_av = 3 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 3 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 3 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 3 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 3 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 3 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));
Else utilityi = ( Log (Endwi));

If r_av = 3 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));
Else utility9 = ( Log (Endw9));

If r_av = 3 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 3 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));
Else utility10 = ( Log (Endw10));

If r_av = 3 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

If r_av = 3 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 3 then utilityl = (((Endwl)** (1- r_av))/ (1-r_av));
Else utilityl = ( Log (Endwl));

If r_av = 3 then utility12 = (((Endw12)** (1- r_av))/ (1-r_av));
Else utility12 = ( Log (Endw12));

run;

proc means data = Analysis.U2 ;

```

```

var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7
utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;
output out = analysis.EU2 mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utilityl utility11 utilityl utility12;
run;

```

```

data Analysis.EU_CE2;
set Analysis.EU2;
r_av=3;

```

```

If r_av=3 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0)));

```

```

If r_av=3 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya)));

```

```

If r_av=3 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

```

```

If r_av=3 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb)));

```

```

If r_av=3 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));
Else CE2 = ((exp (utility2)));

```

```

If r_av=3 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc)));

```

```

If r_av=3 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3)));

```

```

If r_av=3 then CED = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CED = ((exp (utilityd)));

```

```

If r_av=3 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));
Else CE4 = ((exp (utility4)));

```

```

If r_av=3 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));
Else CEe = ((exp (utilitye)));

```

```

If r_av=3 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));
Else CE5 = ((exp (utility5)));

```

```

If r_av=3 then CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEf = ((exp (utilityf)));

```

```

If r_av=3 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

```

```

If r_av=3 then CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));

```

```

Else CEg = ((exp (utilityg)));

If r_av=3 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));
Else CE7 = ((exp (utility7)));

If r_av=3 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=3 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

If r_av=3 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=3 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=3 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=3 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=3 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=3 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

If r_av=3 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=3 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;

proc means;
run;
proc sort data = Analysis.cp1;
by year;
run;
proc sort data = Analysis.F_bos;
by year;
run;
data Analysis.cor (keep = year indemnity option100);
merge analysis.cp1 analysis.F_bos;
by year;
if year ne '.';
run;
proc corr data = analysis.cor;
var option100 indemnity;
run;

```

## Call Options Including Corn Belts Hedging the Price Risk of Corn RP-HPE

```
libname analysis 'C:\Users\sweta\Desktop\data6a';
proc import out= Analysis.data1985 datafile =
"C:\Users\sweta\Desktop\data6a\cleaned1985.xlsx"
DBMS= xlsx;
sheet = "cleaned1985";
GETNAMES= YES;
run;

proc sort data=Analysis.data1985;
by year;
proc import out= Analysis.cornprices datafile =
"C:\Users\sweta\Desktop\data6a\corn_crop_futures_summary3.xlsx"
DBMS= xlsx;
sheet = "sheet3";
GETNAMES= YES;
run;
proc sort data=Analysis.cornprices;
by year;

proc import out= Analysis.corn_business_summary datafile =
"C:\Users\sweta\Desktop\data6a\Required85_16.xlsx"
DBMS= xlsx;
sheet = "Required85_16";
GETNAMES= YES;
run;
proc sort data=Analysis.corn_business_summary;
by astate acounty;
data Analysis.corn_business_summary1;
set Analysis.corn_business_summary;
if insplan = 2;
run;
/*WEIGHTED COVERAGE LEVEL by county and state*/
proc means data = analysis.corn_business_summary1 ;
by astate acounty;
var cvglvl;
weight netacr;
output out= analysis.current_mean_county_covg MEAN=cvglvl;
run;
proc sort data = analysis.corn_business_summary1;
by astate acounty;
run;
data Analysis.corn_business_summary2 (keep = astate acounty
netacr);
set Analysis.corn_business_summary1;
run;
proc sort data = analysis.corn_business_summary2;
by astate acounty;
run;
```



```

proc sort data=Analysis.current_mean_county_covg;
by astate acounty;
run;
/*Simulation for yield*/
proc sort data = Analysis.Data1985;
by astate acounty;
run;
proc reg noprint data=Analysis.Data1985 outest= Analysis.Regout;
/*Saving Intercept and Parameter*/
model value = year; /*Linear trend for each county*/
by astate acounty;
output out= Analysis.b /*Saving Residuals*/
p = vhat
r= vresid;
run;
proc sort data = Analysis.b;
by astate acounty;
run;
data Analysis.pred;
set Analysis.regout;
ypred = Intercept + year*2017; /*Simulation for yield guarantee by
county*/
run;
proc sort data = Analysis.pred;
by astate acounty;
run;

data Analysis.C (keep = year astate acounty intercept year ypred
vresid);
merge Analysis.pred Analysis.b;
by astate acounty;
run;
proc sort data =Analysis.C;
by astate acounty;
run;
data Analysis.D;
set Analysis.C;
dtrendY = ypred + vresid * 1.66; /*Simulation for average
representative farm yield*/
If dtrendY < 0 then dtrendY = 0;
run;
proc sort data = Analysis.D;
by astate acounty;
run;
data Analysis.E;
merge Analysis.current_mean_county_covg Analysis.D;
by astate acounty;
run;
proc sort data = Analysis.E;
by year;
Data Analysis.F;
merge Analysis.E Analysis.Cornprices;
by year;
run;

```

```

proc sort data = Analysis.F;
by astate acounty;
run;
data Analysis.G;
merge Analysis.F Analysis.corn_business_summary2;
by astate acounty;
run;
data Analysis.H;
set Analysis.G;

priceratio = harvest_price / projected_price;
projected_price2017 = 3.86;
harvest_price2017 = priceratio * projected_price2017;
Indem1 = max (0, ((max (projected_price2017 , harvest_price2017)*
cvglvl* ypred) - (harvest_price2017 * dtrendY)));
Liab1 = projected_price2017 * cvglvl * ypred;
Indema = max(0,(projected_price2017*cvglvl*ypred) -
(harvest_price2017*dtrendy));
run;

proc sort data = Analysis.H;
by astate acounty;
run;

proc means data = Analysis.H;
by astate acounty;
var indema;
output out = Analysis.Expindem mean = Expindem;
run;
proc sort data =Analysis.Expindem;
by astate acounty;
run;
data Analysis.SIM_BOS;
merge Analysis.Expindem Analysis.H;
by astate acounty;
run;
proc sort data = Analysis.SIM_BOS;
by astate acounty;
run;

data Analysis.SIM_BOS1;
set analysis.SIM_BOS;
prem = Expindem ;
run;
PROC SORT DATA = Analysis.SIM_BOS1;
by year;
run;

proc means data = Analysis.SIM_BOS1;
by year;
var prem Indem1 Liab1 Indema;
weight netacr;
output out =Analysis.F_BOS sum =premium indemnity liability indemnitya;
run;

```

```

proc sort data = aANALYSIS.F_bos;
by year;
run;
proc sort data = Analysis.SIM_BOS1;
by astate;
run;
proc means data = Analysis.SIM_BOS1;
by astate;
var prem Indem1 Liab1 Indema;
weight netacr;
output out =Analysis.F_BOS1 sum =premium indemnity liability
indemnitya;
run;

data Analysis.CP;
set Analysis.Cornprices;
if harvest_price > projected_price then optprm = projected_price -
harvest_price ;
else optprm = 0;
/* tot scale up to a million dollars of liab */
optprm = optprm * (1000000 /3.86);

run;
proc means data = Analysis.CP;
var optprm;
output out = Analysis.optprm mean =avgoptpremium;
run;

data Analysis.Cp1;
set Analysis.cp;

If projected_price < harvest_price then option0 = (0.0 * 1000000 *
(harvest_price - projected_price));
else option0 = 0;

If projected_price < harvest_price then optiona = (0.005 * 1000000 *
(harvest_price - projected_price));
else optiona = 0;

If projected_price < harvest_price then option1 = (0.01 * 1000000 *
(harvest_price - projected_price));
else option1 = 0;

If projected_price < harvest_price then optionb = (0.015 * 1000000 *
(harvest_price - projected_price));
else optionb = 0;

If projected_price < harvest_price then option2 = (0.02 * 1000000 *
(harvest_price - projected_price));
else option2 = 0;

```

```
If projected_price < harvest_price then optionc = (0.025 * 1000000 *  
(harvest_price - projected_price));  
else optionc = 0;
```

```
If projected_price < harvest_price then option3 = (0.03 * 1000000 *  
(harvest_price - projected_price));  
else option3 = 0;
```

```
If projected_price < harvest_price then optiond = (0.035 * 1000000 *  
(harvest_price - projected_price));  
else optiond = 0;
```

```
If projected_price < harvest_price then option4 = (0.04 * 1000000 *  
(harvest_price - projected_price));  
else option4 = 0;
```

```
If projected_price < harvest_price then optione = (0.045 * 1000000 *  
(harvest_price - projected_price));  
else optione = 0;
```

```
If projected_price < harvest_price then option5 = (0.05 * 1000000 *  
(harvest_price - projected_price));  
else option5 = 0;
```

```
If projected_price < harvest_price then optionf = (0.055 * 1000000 *  
(harvest_price - projected_price));  
else optionf = 0;
```

```
If projected_price < harvest_price then option6 = (0.06 * 1000000 *  
(harvest_price - projected_price));  
else option6 = 0;
```

```
If projected_price < harvest_price then optiong = (0.065 * 1000000 *  
(harvest_price - projected_price));  
else optiong = 0;
```

```
If projected_price < harvest_price then option7 = (0.07 * 1000000 *  
(harvest_price - projected_price));  
else option7 = 0;
```

```
If projected_price < harvest_price then optionh = (0.075 * 1000000 *  
(harvest_price - projected_price));  
else optionh = 0;
```

```
If projected_price < harvest_price then option8 = (0.08 * 1000000 *  
(harvest_price - projected_price));  
else option8 = 0;
```

```

If projected_price < harvest_price then optioni = (0.085 * 1000000 *
(harvest_price - projected_price));
else optioni = 0;

If projected_price < harvest_price then option9 = (0.09 * 1000000 *
(harvest_price - projected_price));
else option9 = 0;

If projected_price < harvest_price then optionj = (0.095 * 1000000 *
(harvest_price - projected_price));
else optionj = 0;

If projected_price < harvest_price then optionl0 = (0.10 * 1000000 *
(harvest_price - projected_price));
else optionl0 = 0;

If projected_price < harvest_price then optionk = (0.105 * 1000000 *
(harvest_price - projected_price));
else optionk = 0;

If projected_price < harvest_price then optionl1 = (0.11 * 1000000 *
(harvest_price - projected_price));
else optionl1 = 0;

If projected_price < harvest_price then optionl = (0.115 * 1000000 *
(harvest_price - projected_price));
else optionl = 0;

If projected_price < harvest_price then optionl2 = (0.12 * 1000000 *
(harvest_price - projected_price));
else optionl2 = 0;

run;

proc sort data = Analysis.Cp1;
by year;

proc means data = Analysis.Cp1;
var optionl00;
output out = Analysis.opt100 mean = optionl00;
run;
data Analysis.cm;
merge Analysis.Cp1 analysis.F_BOS;
by year;
run;
proc sort data = Analysis.cm;
by year;
run;

```

```

data Analysis.cml;
set Analysis.cm;

if year ne '.';
scalar= 1000000/liability;

premium = premium * scalar;
indemnity = indemnity * scalar;
beginwealth = 1000000;
avgoptpremium = 191509.68;

Endw0 = beginwealth + premium - indemnity +option0-0.0* avgoptpremium;

Endwa = beginwealth + premium - indemnity +optiona-0.005*
avgoptpremium;

Endw1 = beginwealth + premium - indemnity +option1-0.01* avgoptpremium;

Endwb = beginwealth + premium - indemnity +optionb-0.015*
avgoptpremium;

Endw2 = beginwealth + premium - indemnity +option2-0.02* avgoptpremium;

Endwc = beginwealth + premium - indemnity +optionc-0.025*
avgoptpremium;

Endw3 = beginwealth + premium - indemnity +option3-0.03* avgoptpremium;

Endwd = beginwealth + premium - indemnity +optiond-0.035*
avgoptpremium;

Endw4 = beginwealth + premium - indemnity +option4-0.04* avgoptpremium;

Endwe = beginwealth + premium - indemnity +optione-0.045*
avgoptpremium;

Endw5 = beginwealth + premium - indemnity +option5-0.05* avgoptpremium;

Endwf = beginwealth + premium - indemnity +optionf-0.055*
avgoptpremium;

Endw6 = beginwealth + premium - indemnity +option6-0.06* avgoptpremium;

Endwg = beginwealth + premium - indemnity +optiong-0.065*
avgoptpremium;

Endw7 = beginwealth + premium - indemnity +option7-0.07* avgoptpremium;

Endwh = beginwealth + premium - indemnity +optionh-0.075*
avgoptpremium;

Endw8 = beginwealth + premium - indemnity +option8-0.08* avgoptpremium;

Endwi = beginwealth + premium - indemnity +optioni-0.085*
avgoptpremium;

```

```

Endw9 = beginwealth + premium - indemnity +option9-0.09* avgoptpremium;

Endwj = beginwealth + premium - indemnity +optionj-0.095*
avgoptpremium;

Endwl0 = beginwealth +premium - indemnity +option10-0.1* avgoptpremium;

Endwk = beginwealth + premium - indemnity +optionk-0.105*
avgoptpremium;

Endwl1 = beginwealth + premium - indemnity +optionl1-0.11*
avgoptpremium;

Endwl = beginwealth + premium - indemnity +optionl-0.115*
avgoptpremium;

Endwl2 = beginwealth + premium - indemnity +optionl2-0.12*
avgoptpremium;
run;

proc means; title 'ending wealths including Corn Belt for Call Options
and RP';
proc print;
run;

data Analysis.U;
set Analysis.cml;
r_av =1;

If r_av = 1 then utility0 = ( Log (Endw0));
Else utility0 = (((Endw0)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitya = ( Log (Endwa));
Else utilitya = (((Endwa)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility1 = ( Log (Endwl));
Else utility1 = (((Endwl)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityb = ( Log (Endwb));
Else utilityb = (((Endwb)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility2 = ( Log (Endw2));
Else utility2 = (((Endw2)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityc = ( Log (Endwc));
Else utilityc = (((Endwc)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility3 = ( Log (Endw3));
Else utility3 = (((Endw3)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityd = ( Log (Endwd));
Else utilityd = (((Endwd)** (1- r_av))/ (1-r_av));

```

```

If r_av = 1 then utility4 = ( Log (Endw4));
Else utility4 = (((Endw4)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilitye = ( Log (Endwe));
Else utilitye = (((Endwe)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility5 = ( Log (Endw5));
Else utility5 = (((Endw5)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityf = ( Log (Endwf));
Else utilityf = (((Endwf)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility6 = ( Log (Endw6));
Else utility6 = (((Endw6)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityg = ( Log (Endwg));
Else utilityg = (((Endwg)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility7 = ( Log (Endw7));
Else utility7 = (((Endw7)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityh = ( Log (Endwh));
Else utilityh = (((Endwh)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility8 = ( Log (Endw8));
Else utility8 = (((Endw8)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityi = ( Log (Endwi));
Else utilityi = (((Endwi)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility9 = ( Log (Endw9));
Else utility9 = (((Endw9)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityj = ( Log (Endwj));
Else utilityj = (((Endwj)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility10 = ( Log (Endw10));
Else utility10 = (((Endw10)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityk = ( Log (Endwk));
Else utilityk = (((Endwk)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility11 = ( Log (Endw11));
Else utility11 = (((Endw11)** (1- r_av))/ (1-r_av));

If r_av = 1 then utilityl = ( Log (Endwl));
Else utilityl = (((Endwl)** (1- r_av))/ (1-r_av));

If r_av = 1 then utility12 = ( Log (Endw12));
Else utility12 = (((Endw12)** (1- r_av))/ (1-r_av));

```



```

proc means data = Analysis.U;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7
utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;

output out = analysis.EU mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utilityl utility11 utilityl utility12;
run;
data Analysis.EU_CE;
set Analysis.EU;
r_av=1;
If r_av=1 then CE0 = exp (utility0);
Else CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEa = exp (utilitya);
Else CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE1 = exp (utility1);
Else CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEb = exp (utilityb);
Else CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE2 = exp (utility2);
Else CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEc = exp (utilityc);
Else CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE3 = exp (utility3);
Else CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE4 = exp (utilityd);
Else CE4 = (((utilityd * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE4 = exp (utility4);
Else CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEe = exp (utilitye);
Else CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE5 = exp (utility5);
Else CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEf = exp (utilityf);
Else CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE6 = exp (utility6);
Else CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));

```

```

If r_av=1 then CEg = exp (utilityg);
Else CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE7 = exp (utility7);
Else CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEh = exp (utilityh);
Else CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE8 = exp (utility8);
Else CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEi = exp (utilityi);
Else CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE9 = exp (utility9);
Else CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEj = exp (utilityj);
Else CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE10 = exp (utility10);
Else CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEk = exp (utilityk);
Else CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE11 = exp (utility11);
Else CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CEL = exp (utilityl);
Else CEL = (((utilityl * (1- r_av))**(1 / (1-r_av))));

If r_av=1 then CE12 = exp (utility12);
Else CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
run;
proc means;
run;

data Analysis.U1;
set Analysis.cml;
r_av =2;
If r_av = 2 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 2 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

If r_av = 2 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 2 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

```

```

If r_av = 2 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

If r_av = 2 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 2 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 2 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

If r_av = 2 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4= ( Log (Endw4));

If r_av = 2 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 2 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5= ( Log (Endw5));

If r_av = 2 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));
Else utilityf = ( Log (Endwf));

If r_av = 2 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 2 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 2 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 2 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 2 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 2 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));
Else utilityi = ( Log (Endwi));

If r_av = 2 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));
Else utility9 = ( Log (Endw9));

If r_av = 2 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 2 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));
Else utility10 = ( Log (Endw10));

If r_av = 2 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

```

```

If r_av = 2 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 2 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 2 then utility12 = (((Endw12)** (1- r_av))/ (1-r_av));
Else utility12 = ( Log (Endw12));

run;

proc means data = Analysis.U1 ;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6
utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utility11 utilityl utility12;
output out = analysis.EU1 mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4
utilitye utility5 utilityf utility6 utilityg utility7 utilityh utility8
utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;
run;

data Analysis.EU_CE1;
set Analysis.EU1;
r_av=2;
If r_av=2 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0)));

If r_av=2 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya)));

If r_av=2 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=2 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb)));

If r_av=2 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));
Else CE2 = ((exp (utility2)));

If r_av=2 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc)));

If r_av=2 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3)));

If r_av=2 then CED = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CED = ((exp (utilityd)));

If r_av=2 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));
Else CE4 = ((exp (utility4)));

If r_av=2 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));

```

```

Else CEe = ((exp (utilitye)));

If r_av=2 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));
Else CE5 = ((exp (utility5)));

If r_av=2 then CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEf = ((exp (utilityf)));

If r_av=2 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

If r_av=2 then CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));
Else CEg = ((exp (utilityg)));

If r_av=2 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));
Else CE7 = ((exp (utility7)));

If r_av=2 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=2 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

If r_av=2 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=2 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=2 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=2 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=2 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=2 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

If r_av=2 then CE1 = (((utilityl * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utilityl)));

If r_av=2 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;
proc means;
run;

data Analysis.U2;
set Analysis.cml;
r_av =3;

```

```

If r_av = 3 then utility0 = (((Endw0)** (1- r_av))/ (1-r_av));
Else utility0 = ( Log (Endw0));

If r_av = 3 then utilitya = (((Endwa)** (1- r_av))/ (1-r_av));
Else utilitya = ( Log (Endwa));

If r_av = 3 then utility1 = (((Endw1)** (1- r_av))/ (1-r_av));
Else utility1 = ( Log (Endw1));

If r_av = 3 then utilityb = (((Endwb)** (1- r_av))/ (1-r_av));
Else utilityb = ( Log (Endwb));

If r_av = 3 then utility2 = (((Endw2)** (1- r_av))/ (1-r_av));
Else utility2 = ( Log (Endw2));

If r_av = 3 then utilityc = (((Endwc)** (1- r_av))/ (1-r_av));
Else utilityc = ( Log (Endwc));

If r_av = 3 then utility3 = (((Endw3)** (1- r_av))/ (1-r_av));
Else utility3 = ( Log (Endw3));

If r_av = 3 then utilityd = (((Endwd)** (1- r_av))/ (1-r_av));
Else utilityd = ( Log (Endwd));

If r_av = 3 then utility4 = (((Endw4)** (1- r_av))/ (1-r_av));
Else utility4 = ( Log (Endw4));

If r_av = 3 then utilitye = (((Endwe)** (1- r_av))/ (1-r_av));
Else utilitye = ( Log (Endwe));

If r_av = 3 then utility5 = (((Endw5)** (1- r_av))/ (1-r_av));
Else utility5 = ( Log (Endw5));

If r_av = 3 then utilityf = (((Endwf)** (1- r_av))/ (1-r_av));
Else utilityf = ( Log (Endwf));

If r_av = 3 then utility6 = (((Endw6)** (1- r_av))/ (1-r_av));
Else utility6 = ( Log (Endw6));

If r_av = 3 then utilityg = (((Endwg)** (1- r_av))/ (1-r_av));
Else utilityg = ( Log (Endwg));

If r_av = 3 then utility7 = (((Endw7)** (1- r_av))/ (1-r_av));
Else utility7 = ( Log (Endw7));

If r_av = 3 then utilityh = (((Endwh)** (1- r_av))/ (1-r_av));
Else utilityh = ( Log (Endwh));

If r_av = 3 then utility8 = (((Endw8)** (1- r_av))/ (1-r_av));
Else utility8 = ( Log (Endw8));

If r_av = 3 then utilityi = (((Endwi)** (1- r_av))/ (1-r_av));
Else utilityi = ( Log (Endwi));

If r_av = 3 then utility9 = (((Endw9)** (1- r_av))/ (1-r_av));

```

```

Else utility9 = ( Log (Endw9));

If r_av = 3 then utilityj = (((Endwj)** (1- r_av))/ (1-r_av));
Else utilityj = ( Log (Endwj));

If r_av = 3 then utility10 = (((Endw10)** (1- r_av))/ (1-r_av));
Else utility10 = ( Log (Endw10));

If r_av = 3 then utilityk = (((Endwk)** (1- r_av))/ (1-r_av));
Else utilityk = ( Log (Endwk));

If r_av = 3 then utility11 = (((Endw11)** (1- r_av))/ (1-r_av));
Else utility11 = ( Log (Endw11));

If r_av = 3 then utilityl = (((Endwl)** (1- r_av))/ (1-r_av));
Else utilityl = ( Log (Endwl));

If r_av = 3 then utility12 = (((Endw12)** (1- r_av))/ (1-r_av));
Else utility12 = ( Log (Endw12));

run;

proc means data = Analysis.U2 ;
var utility0 utilitya utility1 utilityb utility2 utilityc utility3
utilityd utility4 utilitye utility5 utilityf utility6 utilityg utility7
utilityh utility8 utilityi utility9 utilityj utility10 utilityk
utility11 utilityl utility12;
output out = analysis.EU2 mean= utility0 utilitya utility1 utilityb
utility2 utilityc utility3 utilityd utility4 utilitye utility5 utilityf
utility6 utilityg utility7 utilityh utility8 utilityi utility9 utilityj
utility10 utilityk utility11 utilityl utility12;
run;

data Analysis.EU_CE2;
set Analysis.EU2;
r_av=3;

If r_av=3 then CE0 = (((utility0 * (1- r_av))**(1 / (1-r_av))));
Else CE0 = ((exp (utility0)));

If r_av=3 then CEa = (((utilitya * (1- r_av))**(1 / (1-r_av))));
Else CEa = ((exp (utilitya)));

If r_av=3 then CE1 = (((utility1 * (1- r_av))**(1 / (1-r_av))));
Else CE1 = ((exp (utility1)));

If r_av=3 then CEb = (((utilityb * (1- r_av))**(1 / (1-r_av))));
Else CEb = ((exp (utilityb)));

If r_av=3 then CE2 = (((utility2 * (1- r_av))**(1 / (1-r_av))));
Else CE2 = ((exp (utility2)));

If r_av=3 then CEc = (((utilityc * (1- r_av))**(1 / (1-r_av))));
Else CEc = ((exp (utilityc)));

```

```

If r_av=3 then CE3 = (((utility3 * (1- r_av))**(1 / (1-r_av))));
Else CE3 = ((exp (utility3)));

If r_av=3 then CEd = (((utilityd * (1- r_av))**(1 / (1-r_av))));
Else CEd = ((exp (utilityd)));

If r_av=3 then CE4 = (((utility4 * (1- r_av))**(1 / (1-r_av))));
Else CE4 = ((exp (utility4)));

If r_av=3 then CEe = (((utilitye * (1- r_av))**(1 / (1-r_av))));
Else CEe = ((exp (utilitye)));

If r_av=3 then CE5 = (((utility5 * (1- r_av))**(1 / (1-r_av))));
Else CE5 = ((exp (utility5)));

If r_av=3 then CEf = (((utilityf * (1- r_av))**(1 / (1-r_av))));
Else CEf = ((exp (utilityf)));

If r_av=3 then CE6 = (((utility6 * (1- r_av))**(1 / (1-r_av))));
Else CE6 = ((exp (utility6)));

If r_av=3 then CEg = (((utilityg * (1- r_av))**(1 / (1-r_av))));
Else CEg = ((exp (utilityg)));

If r_av=3 then CE7 = (((utility7 * (1- r_av))**(1 / (1-r_av))));
Else CE7 = ((exp (utility7)));

If r_av=3 then CEh = (((utilityh * (1- r_av))**(1 / (1-r_av))));
Else CEh = ((exp (utilityh)));

If r_av=3 then CE8 = (((utility8 * (1- r_av))**(1 / (1-r_av))));
Else CE8 = ((exp (utility8)));

If r_av=3 then CEi = (((utilityi * (1- r_av))**(1 / (1-r_av))));
Else CEi = ((exp (utilityi)));

If r_av=3 then CE9 = (((utility9 * (1- r_av))**(1 / (1-r_av))));
Else CE9 = ((exp (utility9)));

If r_av=3 then CEj = (((utilityj * (1- r_av))**(1 / (1-r_av))));
Else CEj = ((exp (utilityj)));

If r_av=3 then CE10 = (((utility10 * (1- r_av))**(1 / (1-r_av))));
Else CE10 = ((exp (utility10)));

If r_av=3 then CEk = (((utilityk * (1- r_av))**(1 / (1-r_av))));
Else CEk = ((exp (utilityk)));

If r_av=3 then CE11 = (((utility11 * (1- r_av))**(1 / (1-r_av))));
Else CE11 = ((exp (utility11)));

If r_av=3 then CEL = (((utilityl * (1- r_av))**(1 / (1-r_av))));
Else CEL = ((exp (utilityl)));

```



```

If r_av=3 then CE12 = (((utility12 * (1- r_av))**(1 / (1-r_av))));
Else CE12 = ((exp (utility12)));
run;
proc means;
run;
proc sort data = Analysis.cp1;
by year;
run;
proc sort data = Analysis.F_bos;
by year;
run;
data Analysis.cor (keep = year indemnitya option100);
merge analysis.cp1 analysis.F_bos;
by year;
if year ne'.';
run;
proc corr data = analysis.cor;
var option100 indemnitya;
run;

```

The analysis conducted for call options to hedge the price risk of corn RP and RP-HPE sold in the region excluding the corn belts, same codes were used but excluded the five major corn belt states, Indiana, Ohio, Iowa, Illinois and Missouri from the main datasets, cleaned 1985 and required85\_16.