EDS 230 Assignment 3: Almond Profit Model Sensitivity Analysis

Elmera Azadpour, Mia Forsline, Alex Vand

2022-04-19

Set up

Introduction

1. Develop a profit model for almond yield

```r
#read in R script to compute almond profit
#more details about the profit model can be found in the .R file
source(here("functions", "compute_profit_from_almonds.R"))

#read in other necessary R scripts

# 1. read in R script to compute almond yield anomaly
source(here("functions", "almond_yield_anomaly_annual.R"))

# 2. read in R script to compute net present value of almonds
source(here("functions", "compute_npv.R"))
```

2. Do a simple informal sensitivity analysis of almond yield profit using at least 2 parameters

- Similar to the in-class example, we plan to conduct a sensitivity analysis assuming +/- 15% uncertainty in the current price of almonds (`price`) and discount rate (`discount`)
- We assume a default almond price of $X/ton
- We assume a default discount rate of 0.12
- We begin by sampling X times from a uniform distribution

```r
#parameter defaults
almond_price_default <- 3000 #$1.47/lb * 2000lb = approximately $3000/ton
discount_rate_default <- 0.12

#deviation %
deviation = 0.15

#number of samples
nsamples = 300
```

```
#sample a uniform distribution from the price default
price <- runif(min = almond_price_default - (deviation * almond_price_default),
                      max = almond_price_default + (deviation * almond_price_default),
                      n=nsamples)

discount = rnorm(mean=0.6, sd = 0.1, n=nsamples)

#bind price_thresh and discount_rate into a dataframe
parameters <- cbind.data.frame(discount, price)
```

```
#read in CSV of output from annual almond yield anomaly function
annual_almond_yield <- read_csv(here("data", "annual_almond_yield.csv"))

#note that parameters column names must match the input parameter names in the compute_profit_from_almo
## testing creation of results by using static values for almond_yield_anomaly and year
results <- parameters %>%
  pmap(compute_profit_from_almonds,
       almond_yield_anomaly = 10,
       year = 1980)

#check the results
results[[1]]
```

```
##   scen almond_yield_anomaly year      net   netpre
## 1    1                   10 1980 32049.41 32049.41
```

```
length(results)
```

```
## [1] 300
```

Can't get this code chunk to run - Mia

```
# # now we can extract results from the list as above
# mean = map_df(results,`[`, c("mean"))
#
# # and we can add the parameter values for each run so we know what parameters gave us which mean valu
# mean_elect = cbind.data.frame(mean_elect, parms)
```

3. Create a single graph of the results

4. Output the graph as a stand along image