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

```
#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"))
```

Test the `almond_yield_anomaly_annual()` function - save the output as a CSV

```
# read in the data from .txt file
clim_raw <- read.csv(file = here("data", "clim.txt"),
                     header = T,
                     sep = "")

# clean data
clim_data <- clim_raw %>%
             janitor::clean_names() %>%
             mutate(d = lubridate::as_date(d), #convert d column to Date format rather than a character
                    year = lubridate::year(d))

# set coefficient parameter values
temp_coeff1 <- -0.015
temp_coeff2 <- -0.0046
precip_coeff1 <- -0.07
precip_coeff2 <- 0.0043
constant <- 0.28
```

```
# calculate annual almond yield for each water year (wy)
annual_almond_yield <- cbind(unique(clim_data$wy), almond_yield_anomaly_annual(clim_data)) %>%  #create
  as.data.frame() %>% #transform into a data frame
  rename(year = V1,
         yield = V2)

#save output as a CSV in the data folder
write_csv(x = annual_almond_yield, path = here("data", "annual_almond_yield.csv"))
```

Test the `compute_npv()` function

```
npv_test <- round(compute_npv(value = 10, time = 2, discount = 0.12), digits = 2)
```

With a discount rate of 12% over 2 years, a current value of $10 will have a net present value (NPV) of approximately $7.97.

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 $3,000/ton
- We assume a default discount rate of 0.12
- We begin by sampling 300 times from a uniform distribution

```
#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)

#sample from a normal distribution
discount = rnorm(mean=0.6, sd = 0.1, n=nsamples)

#bind price 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
```

```r
results <- parameters %>%
  pmap(compute_profit_from_almonds,
       almond_yield_anomaly = annual_almond_yield$yield,
       year = annual_almond_yield$year)

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

```
##    scen almond_yield_anomaly year          net        netpre
## 1     1           -0.3552237 1989    -953.8189 -9.538189e+02
## 2     2            9.2906757 1990   24946.5967  1.425594e+04
## 3     3           68.9130633 1991  185039.9751  6.042748e+04
## 4     4           15.4280698 1992   41426.2479  7.730880e+03
## 5     5           20.2083803 1993   54261.9643  5.786731e+03
## 6     6            2.4820009 1994    6664.4748  4.061516e+02
## 7     7         1919.9811511 1995 5155383.4264  1.795427e+05
## 8     8            3.5818399 1996    9617.6767  1.914085e+02
## 9     9          329.6938750 1997  885268.2424  1.006817e+04
## 10   10           27.8636956 1998   74817.4192  4.862535e+02
## 11   11           -0.1436364 1999    -385.6813 -1.432429e+00
## 12   12            9.5999883 2000   25777.1387  5.470961e+01
## 13   13          159.5119587 2001  428309.0529  5.194821e+02
## 14   14            0.2450914 2002     658.1003  4.561313e-01
## 15   15           -0.2585997 2003    -694.3718 -2.750263e-01
## 16   16           -0.2367722 2004    -635.7623 -1.439002e-01
## 17   17          656.3724121 2005 1762439.9349  2.279636e+02
## 18   18           18.6324135 2006   50030.3014  3.698015e+00
## 19   19           20.2007396 2007   54241.4483  2.291138e+00
## 20   20          576.2821943 2008 1547387.9375  3.735112e+01
## 21   21            0.7367438 2009    1978.2470  2.728786e-02
## 22   22          153.7655092 2010  412879.1354  3.254590e+00
```

```r
#length(results) - should be a length of 300
```

```r
# now we can extract data from the results list created above
mean <- map_df(results, #list we want to extract data from
               `[`, # `[` is a function that means extract
               c("year",
                 "net",
                 "netpre")) #select the columns from results we want to extract

# and we can add the parameter values for each run so we know what parameters gave us which mean value
mean <- cbind.data.frame(mean, parameters)
```

3. Create a single graph of the results

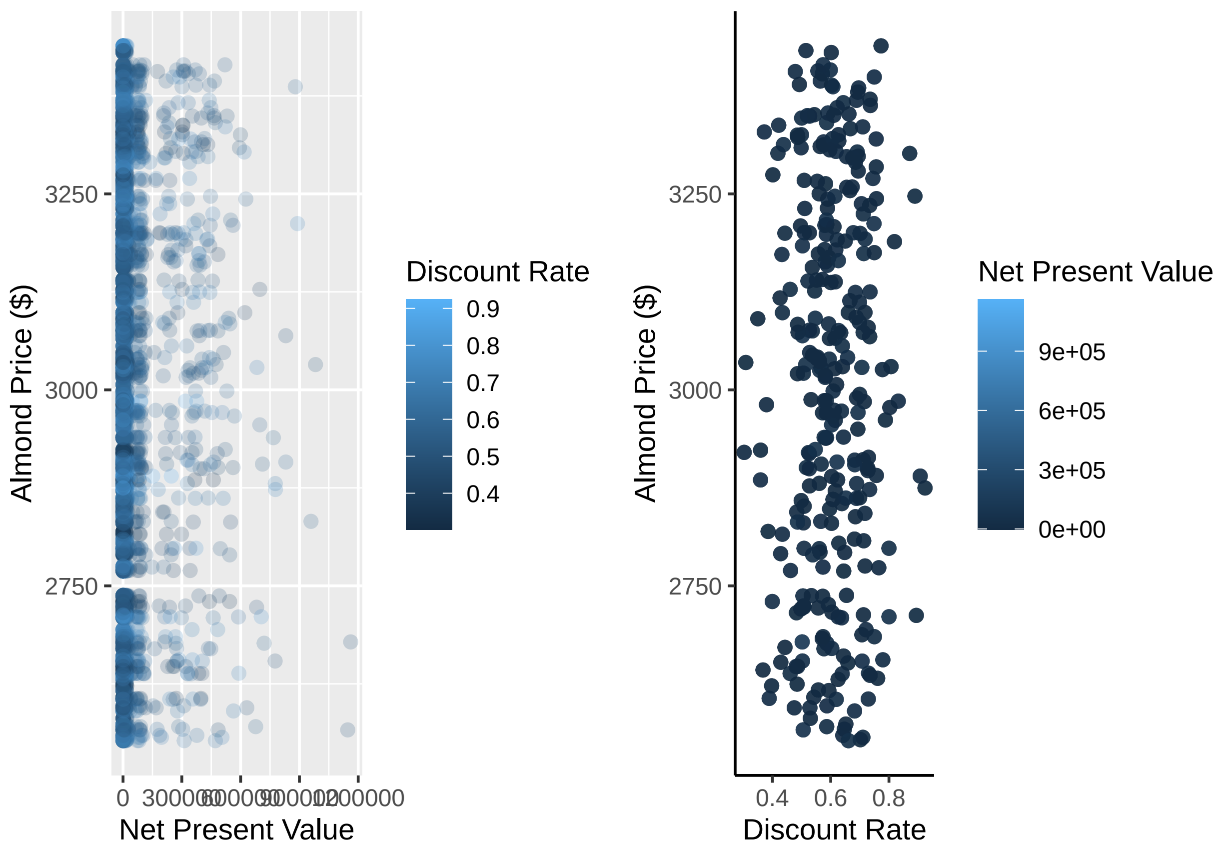
- remember, for our sensitivity analysis, we chose to vary the price and discount rate to determine how almond yield changed over time

```
# plot - pick one of the 2 parameter as a color

p1 <- ggplot(data = mean, aes(x = netpre, y = price, color = discount)) +
  geom_point(cex=2, alpha = 0.2) + # alpha = opacity
  labs(y = "Almond Price ($)",
       x = "Net Present Value",
       color = "Discount Rate")
  theme_classic()

p2 <- ggplot(data = mean, aes(x = discount, y = price, color = netpre)) +
  geom_point(cex=2, alpha = 0.1) + # alpha = opacity
  labs(y = "Almond Price ($)",
       x = "Discount Rate",
       color = "Net Present Value") +
  theme_classic()

ggarrange(p1,p2)
```



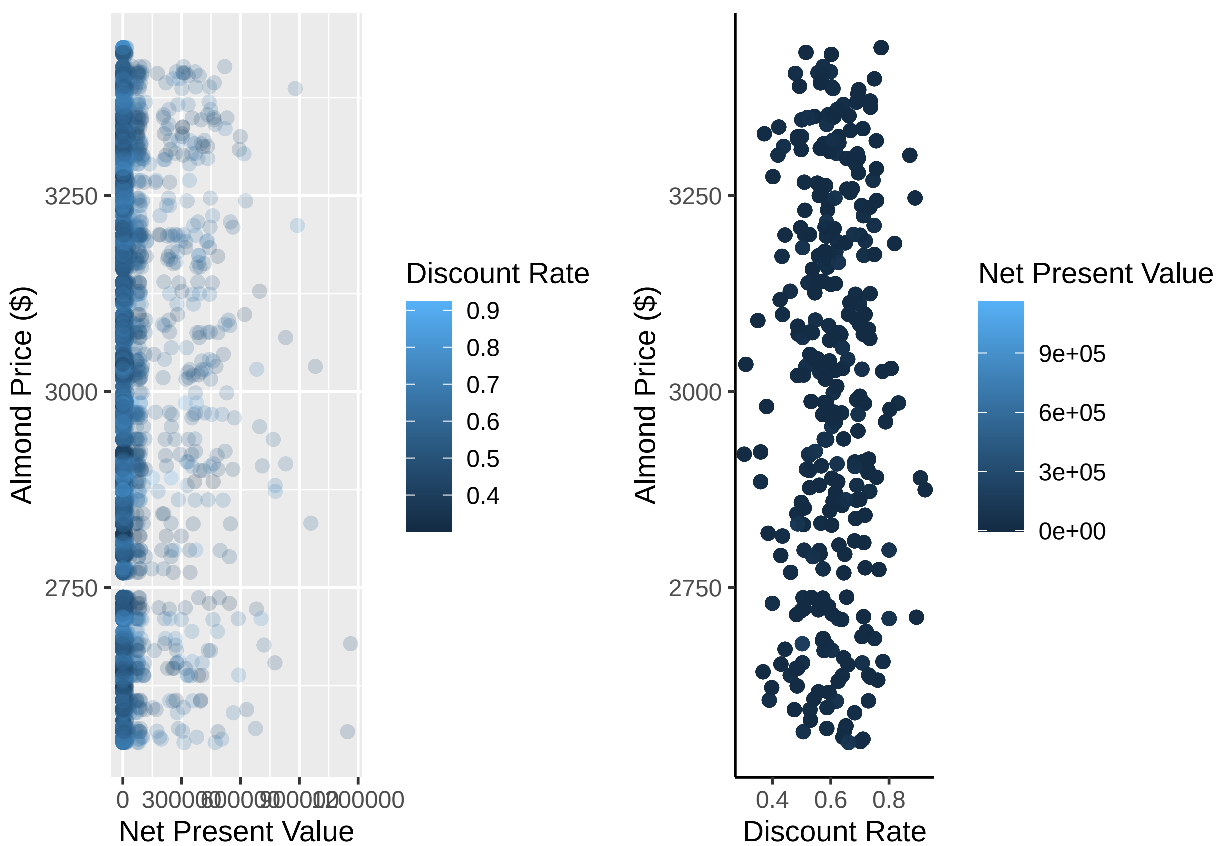

```
# plot - the 2 parameters as independent variables on x-axis
# NOTE THAT THESE PLOTS LOOK THE SAME AS THE ONES ABOVE - does this make sense?

p3 <- ggplot(data = mean, aes(x = netpre, y = price, color = discount)) +
  geom_point(cex=2, alpha = 0.2) + # alpha = opacity
  labs(y = "Almond Price ($)",
       x = "Net Present Value", # parameter 1
```

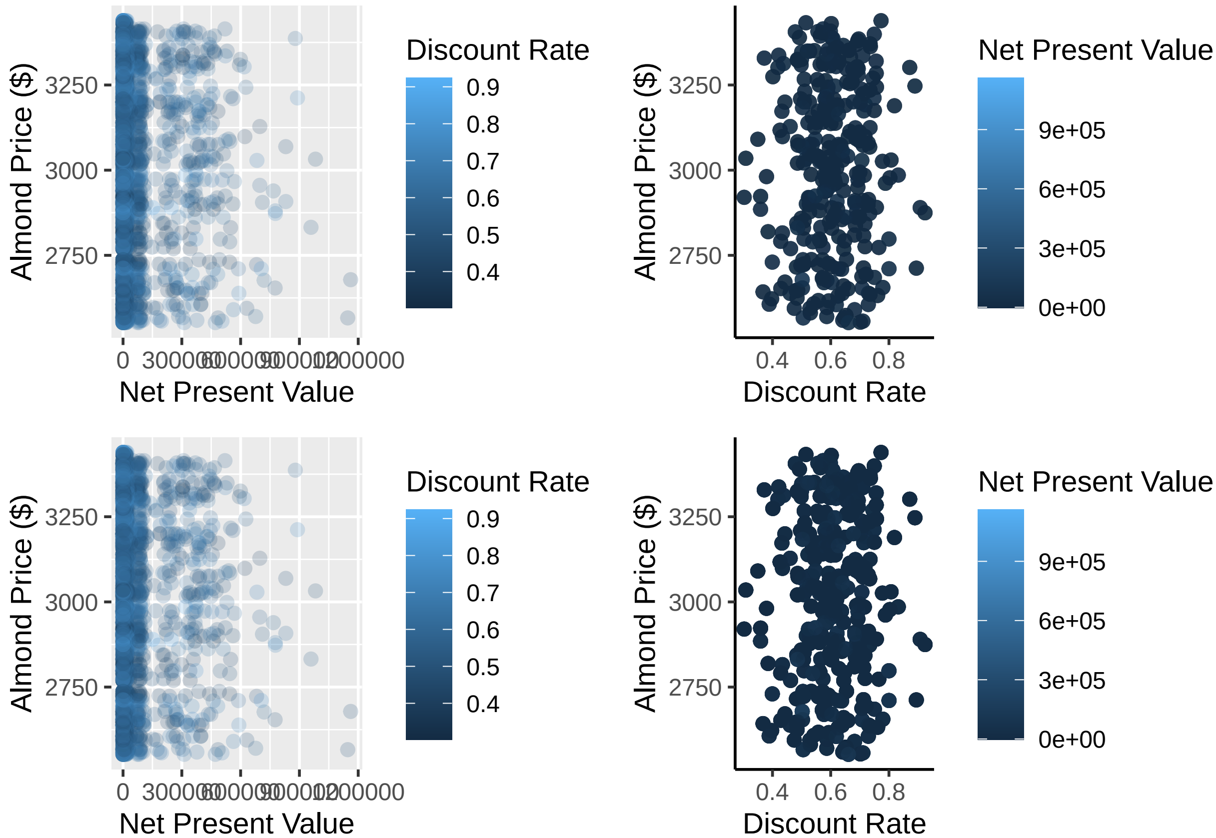
```
       color = "Discount Rate")
  theme_classic()

p4 <- ggplot(data = mean, aes(x = discount, y = price, color = netpre)) +
  geom_point(cex=2, alpha = 0.2) + # alpha = opacity
  labs(y = "Almond Price ($)",
       x = "Discount Rate", # parameter 2
       color = "Net Present Value") +
  theme_classic()

ggarrange(p3,p4)
```



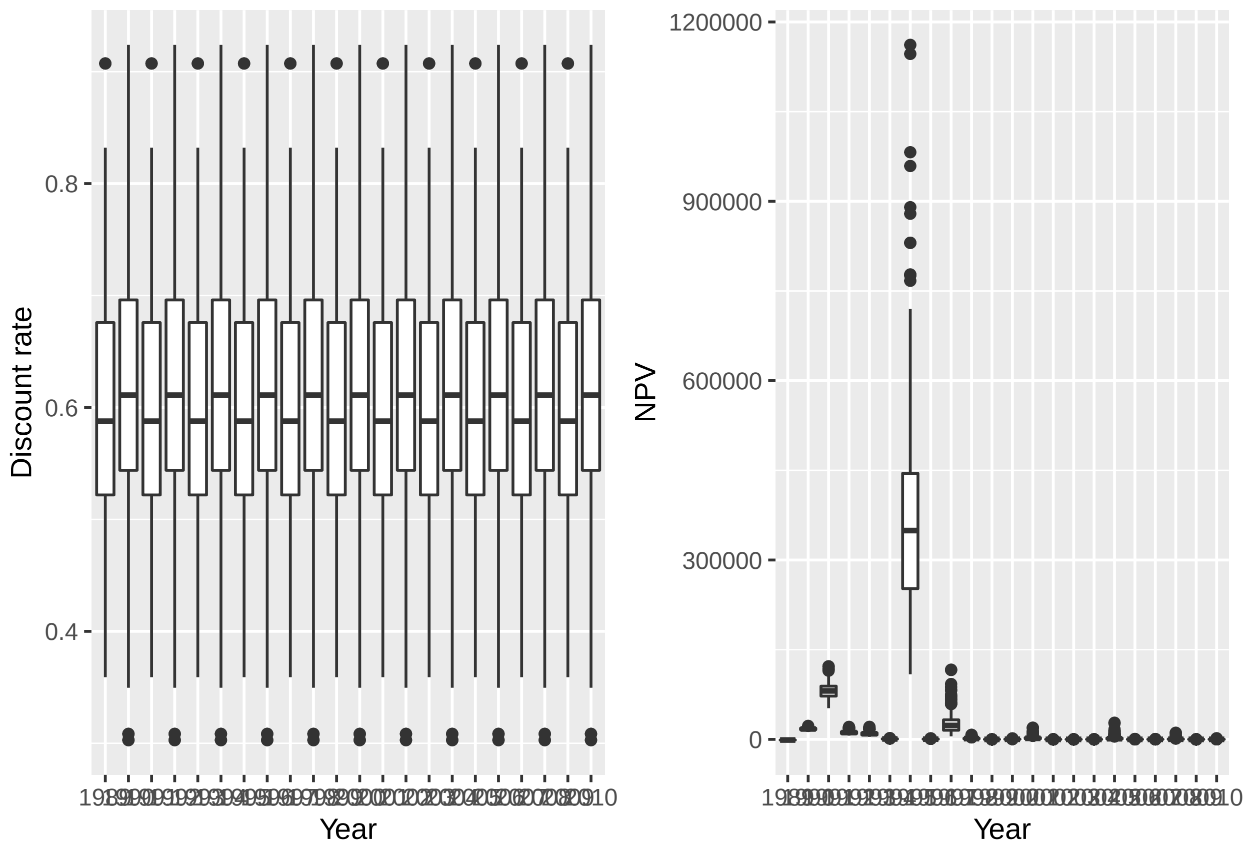

```
ggarrange(p1,p2,p3,p4) #to compare all 4 graphs
```




```
boxplot1 <- ggplot(data = mean,
                   aes(as.factor(year), discount, group = year)) +
              geom_boxplot() +
              labs(x="Year", y="Discount rate")


boxplot2 <- ggplot(data = mean,
                   aes(as.factor(year), netpre, group = year)) +
              geom_boxplot() +
              labs(x="Year", y="NPV")

ggarrange(boxplot1, boxplot2)
```



4. Output the graph as a stand alone image

- save as .png

```
#save image in assignments directory
ggsave("assignment3_graph.png",
       plot = ggarrange(p1,p2))
```