

# EDS 230 Assignment 3: Informal Sensitivity Analysis

Erika Egg and Jared Petry

2023-05-02

This is an informal sensitivity analysis for the two parameters in our profit model: `selling_price` and `production_cost`.

## Load libraries

```
library(here)
library(tidyverse)
library(purrr)
library(ggplot2)
library(ggpubr)
```

## Source necessary functions

```
# source necessary functions
source("calc_almond_yield_anomaly_for_profit_func.R")
source("calc_almond_profit.R")
```

Run both models to make sure sourcing worked:

```
calc_almond_yield_anomaly_for_profit_func()
```

```
## # A tibble: 22 x 2
##   year yield_anomaly
##   <int>      <dbl>
## 1  1989      0.0432
## 2  1990       9.67
## 3  1991     69.3
## 4  1992     15.8
## 5  1993     20.5
## 6  1994       2.86
## 7  1995    1920.
## 8  1996       4.19
## 9  1997     330.
## 10 1998     28.2
## # i 12 more rows
```

```
calc_almond_profit()
```

```
## $annual
## # A tibble: 22 x 2
##   year profit_w_anom
##   <int>      <dbl>
## 1 1989      208643.
## 2 1990      213495.
## 3 1991     1406328.
## 4 1992      336469.
## 5 1993      429890.
## 6 1994       77197.
## 7 1995     3842616.
## 8 1996       10371.
## 9 1997      661961.
## 10 1998      583998.
## # i 12 more rows
##
## $mean
## [1] 36551791
```

Sensitivity analysis with parameters: “selling\_price” and “production\_cost”

```
# Create a sampling distribution for both parameters
# Assume a uniform distribution - we don't know anything about variation
nsamples = 300
deviation = 0.15

base_sellprice = 5000 # default in profit function
selling_price = runif(min = base_sellprice-deviation*base_sellprice,
                      max = base_sellprice+deviation*base_sellprice,
                      n = nsamples) # get a sampling distribution length 300

base_prodcost = 4000 # default in profit function
production_cost = runif(min = base_prodcost-deviation*base_prodcost,
                       max = base_prodcost+deviation*base_prodcost,
                       n = nsamples) # get a sampling distribution length 300

params = cbind.data.frame(selling_price, production_cost) # combine into one df

# Use pmap to acquire results: takes function name and then names of all
# parameters that don't change
results = params %>% pmap(calc_almond_profit,
                        acres = 200,
                        expected_yield = 1)

# Check results
results[[1]]
```

```
## $annual
## # A tibble: 22 x 2
##   year profit_w_anom
##   <int>      <dbl>
## 1 1989      258948.
## 2 1990     2649702.
## 3 1991    17454000.
## 4 1992     4175934.
## 5 1993     5335388.
## 6 1994      958101.
## 7 1995    476908810.
## 8 1996     1287155.
## 9 1997     82156272.
## 10 1998     7248031.
## # i 12 more rows
##
## $mean
## [1] 45364584
```

```
length(results)
```

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

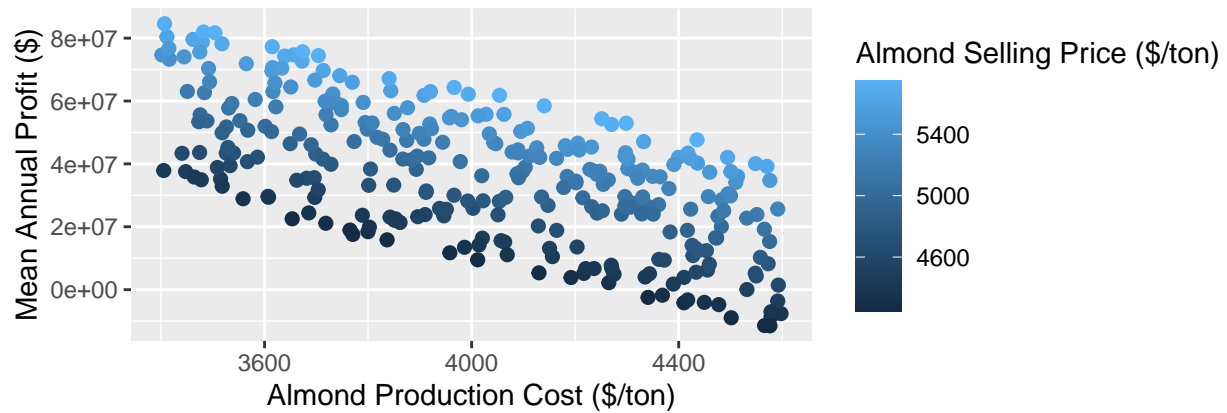
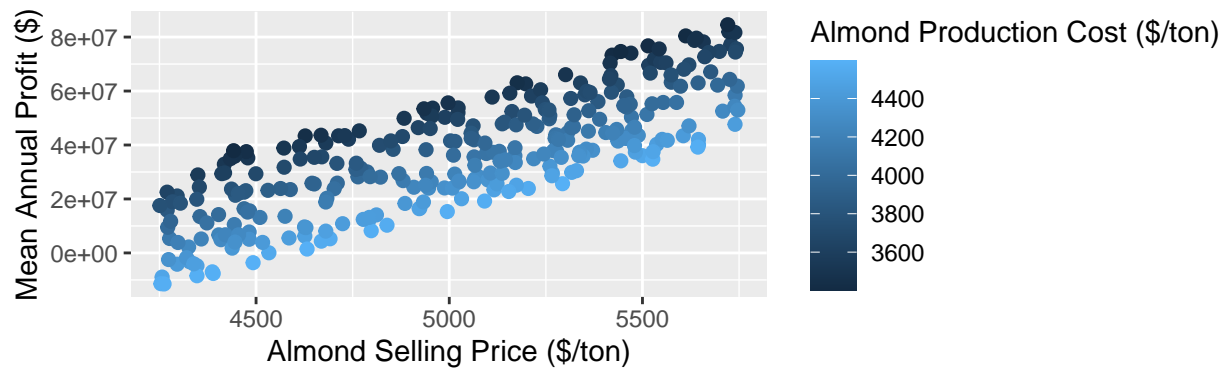
```
# Extract results from the list as above
mean_profit = map_df(results, `[,`, c("mean"))

# Add the parameter values for each run
mean_profit = cbind.data.frame(mean_profit, params)

# Plots for means - two versions of same thing
p1 = ggplot(mean_profit, aes(selling_price, mean, col = production_cost)) +
  geom_point(cex = 2) +
  labs(title = "Mean Annual Profit vs. Almond Selling Price and Production Cost",
       y = "Mean Annual Profit ($)",
       x = "Almond Selling Price ($/ton)",
       col = "Almond Production Cost ($/ton)")
p2 = ggplot(mean_profit, aes(production_cost, mean, col = selling_price)) +
  geom_point(cex = 2) +
  labs(y = "Mean Annual Profit ($)",
       x = "Almond Production Cost ($/ton)",
       col = "Almond Selling Price ($/ton)")

ggarrange(p1, p2, ncol = 1)
```

Mean Annual Profit vs. Almond Selling Price and Production Cost

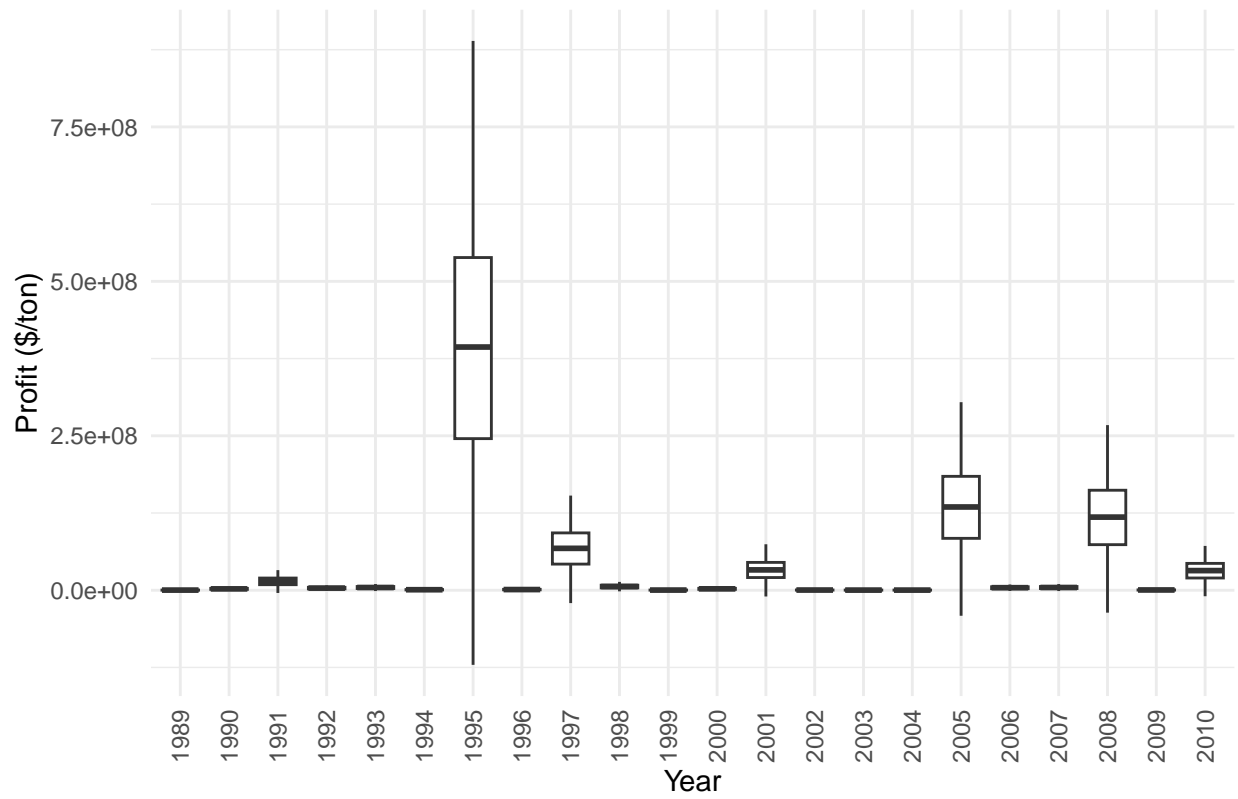


```
ggsave(here("almond_profit_scatterplot.png"))

# Extract annual information too and plot
annual_df = map_df(results, `[, c("annual")`)
annual_profit = as.data.frame(annual_df$annual$year)
colnames(annual_profit) = "year"
annual_profit$profit_w_anom = annual_df$annual$profit_w_anom

ggplot(annual_profit, aes(as.factor(year), profit_w_anom)) +
  geom_boxplot() +
  labs(title = "Almond Profit, 1989-2010",
       x = "Year",
       y = "Profit ($/ton)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

Almond Profit, 1989–2010



```
ggsave(here("almond_profit_boxplot.png"))
```