MAT 488 Project 5

Fred Kaesmann

May 2020

But it is only for the sake of profit that any man employs capital in the support of industry; and he will always, therefore, endeavour to employ it in the support of that industry of which the produce is likely to be of the greatest value, or to echange for the greatest quantity either of money or of other goods.
- Adam Smith

The Problem

Due to the recent economic downturn, Wile E. Cyote's Agricultural supplier of choice, Acme Tractors, has been forced to limit their capital expenditures. To address this problem they have contacted us at Fred's Executive Research and Statistical Analysis: t to z, to assist with inventory optimization.

As the trade war tensions have heated, the possibility of increases in domestic agriculture has risen dramatically. Acme is trying to better prepare themselves by taking full advantage of their obviously meager resources, and looking to perform an external review of their inventory management practices. Extensive evaluation by Acme's internal analysts say they have on average about one tractor sale per week. Given their current facilities they are only capable of housing up to five tractors at any given time, and tractor shipments only come in sets of five tractors, so they cannot schedule a new delivery until they are completely out of units. Deliverys can only be made on Saturdays and Sundays, so if they run out of tractors they must wait until the weekend before their supply is replenished.

Markov Modeling Supply and Demand

And by directing that industry in such a manner as its produce may be of the greatest value, he intends only his own gain, and he is in this, as in many other cases, led by an invisible hand to promote an end which was no part of his intention.

Acme can house a discrete number of tractors and at the end of the week can only be left with a discrete number of tractors. Given the assumption of a Poisson distribution, and an assumed mean value, λ_{Poisson}, [1] we know the probability of some event occuring. In the context of this problem, we can model the demand for tractors by the equation

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

As stated previously, their internal analysts have determined they sell about one tractor per week. We can adjust our above distribution and model Acme's tractor demand by

$$P(X = k) = \frac{e^{-1}}{k!}$$

[1] Not to be confused with $\lambda_{\text{Sensitivity}}$

Markov Chains

We have a small set of possible initial states and discrete probabilities associated with transitions between states. This makes Markov Chains a preferred method for evaluating the possiblilities.

Table 1: Transition Matrix

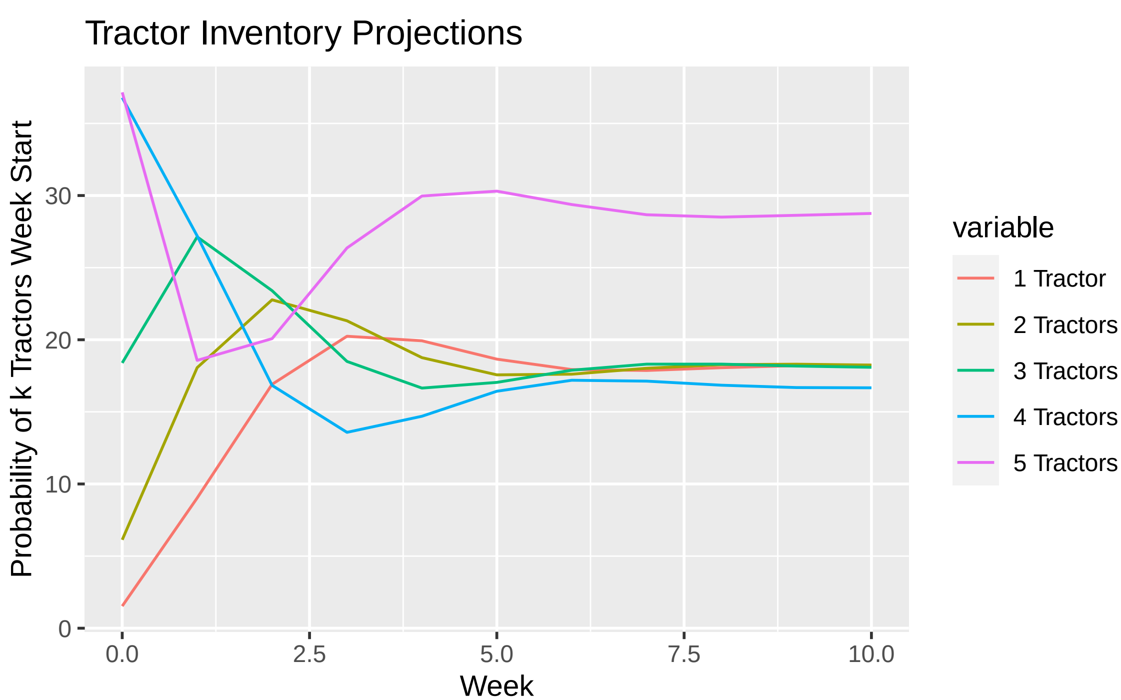
36.788	0.000	0.000	0.000	63.212
36.788	36.788	0.000	0.000	26.424
18.394	36.788	36.788	0.000	8.030
6.131	18.394	36.788	36.788	1.899
1.533	6.131	18.394	36.788	37.154

Table 2 contains our projection for the first three weeks and the tenth week after beginning fully stocked with five tractors. We find that by the third week they are most likely to have 5 tractors in stock.

Table 2: Inventory Probability Matrix

	1 Tractor	2 Tractors	3 Tractors	4 Tractors	5 Tractors
Initial Week	1.533	6.131	18.394	36.788	37.154
Week 1	9.028	18.067	27.134	27.202	18.569
Week 2	16.911	22.771	23.405	16.838	20.075
Week 3	20.243	21.315	18.497	13.580	26.365
Week 10	18.237	18.245	18.091	16.669	28.758

The chart below plots the probability of n tractors in stock by week. Again we see the probability of 5 tractors (the purple line) peaking by week 5 then leveling out and approaching our steady state.



Steady State Analysis

We evaluate our model to determine if our tractors inventory likelihood converges to a steady state. This reflects the position we expect to arrive at after any variation from our initial position has been "washed out" by repeated application of our transition matrix. The table below represents our calculated convergence for tractor inventory. [2]

Table 3: Tractor Steady State

1 Tractor	2 Tractors	3 Tractors	4 Tractors	5 Tractors
18.184	18.201	18.112	16.739	28.763

Our steady state largely reflects our visualization of our weekly projections. We see here again that we are most likely to have 5 tractors in inventory and least likely to have 4.

Missed Opportunites

Table 4: Overdemand Probabilities

1 Tractor	2 Tractors	3 Tractors	4 Tractors	5 Tractors
26.424	8.03	1.899	0.366	0.366

Based on Acme's inventory practices, our calculated steady state, and the Poisson probability given $\lambda = 1$ we find that they will miss out on sales approximately 6.777% of the time. This number does not strike us as particularly egregious, however we leave it to the individuals with access to accounting data to determine whether this is worth investing in expanded capacity or if inventory practices need reexamining.

Sensitivity

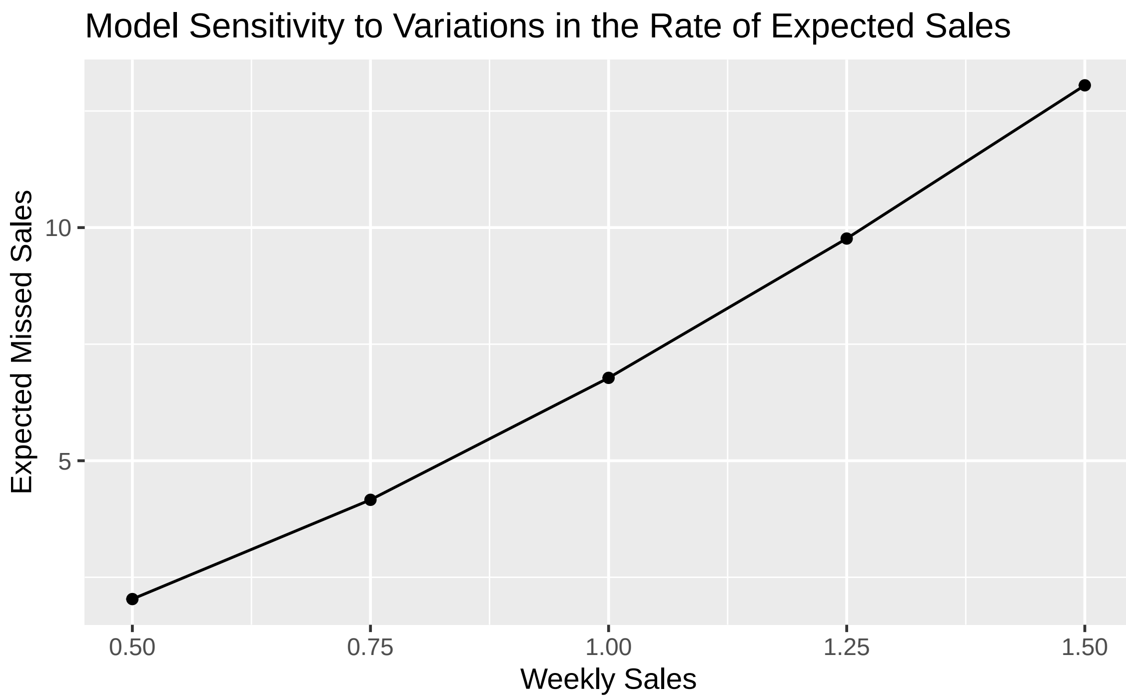
We consider how four other sales assumptions would impact our model in the hopes that we will arrive at a better understanding of the variety of possible outcomes.

Table 5: Sales Expectations

Weekly Sales	Expected Missed Sales
1.00	6.777
0.50	2.033
0.75	4.162
1.25	9.765
1.50	13.051

Our table of sales expectations shows relatively large swings in missed sales. We see that our smallest assumption, one sale every two weeks, translates to a reduction in missed sales by a factor of 3, while our largest assumption, three sales every two weeks, results in missing twice as many sales. The chart below has been provided for visual aide.

[2] The following were evaluated to a tolerance of $10e^{-17}$ just because.



In Summation

Had we more time [3] we would have preferred to evaluate other aspects of our model than just customer frequency assumptions. We would have liked to evaluate the impact of either reducing inventory to four or increasing to six. If capital is harder to come by, Acme may find it more profitable to reduce inventory and turn to capital markets to loan the remaining funds. Regardless, we expect Acme's inventory restrictions to have a far greater impact on our model than assumed patronage, and generally find it to be a more interesting question.

And One More Thing

Ultimately we would be remiss if we did not take the opportunity in our final submitted paper of the year to thank Dr. Ray Mugno for his tolerance and patience while being forced to read the many ERSA:*tz* projects this semester. Thanks Mugno.

[3] As we write, deadlines and finals are fast approaching.

Appendix I

The Definition of Some `FredsStatsPack` Functions Used in this Analysis

```
#' @title Freds_MarkovChain
#'
#' @param intial_state 1 X n matrix
#' @param transition_matrix a square n X n matrix
#' @param iterations
#'
#' @return 1 X n matrix indicating the current state
#'
#' @examples
Freds_MarkovChain <- function(initial_state, transition_matrix, iterations){

  ##### Need expm Package ####
  # For the matrix exponential operator %^%
  library(expm)

  ##### Matrix Exponentiate Transition Matrix ####
  transition_matrix <- transition_matrix %^% iterations

  ##### Do Math ####
  # Current state = initial state * transition matrix ^ iteration
  current_state <- initial_state %*% transition_matrix

  ##### Return state after iterations ####
  return(current_state)
}
```

```r
#' @title Freds_MarkovChain_Steadystate
#'
#' @param initial_state       1 X n matrix indicating initial position
#' @param transition_matrix   n X n matrix
#' @param tolerance           error bound to stop iterations, currently set to
#'                              10 ^ -3
#' @param max_iterations      maximum number of iterations to search for steady
#'                              state, currently set to 15
#'
#' @return A steady state if one exists within constraints
#'
Freds_MarkovChain_SteadyState <- function(initial_state, transition_matrix,
                                          tolerance = -3,
                                     max_iterations = 15) {

  ##### Loop through the max iterations minus one ####
  for (iteration in 1:(max_iterations-1)) {

    ##### Difference between the states ####
    previous_state <- Freds_MarkovChain(initial_state, transition_matrix, iteration)

    current_state <- Freds_MarkovChain(initial_state, transition_matrix, iteration+1)

    # Create boolean vector
    diff <- abs(current_state - previous_state) < 10 ^ tolerance

    ##### Stop Condiditions ####
    # If Steady State reached, return it
    if (sum(diff) == length(diff)) {
      return(current_state)

      # If we haven't reached the Steady State by the max iterations,
      # throw error saying so
    } else if (iteration == max_iterations-1) {
      stop("Steady State not reached in ", max_iterations, " iterations.")
    }
  }

}
```

Appendix II

ERSA:*tz* code used in the above analysis

```
##### Chunk Options ####
opts_chunk$set(
     echo = FALSE,
  include = FALSE,
  message = FALSE,
  warning = FALSE,
  # Figure Settings
  fig.asp = 0.618,
fig.width = 6)

##### Packages ####
library(FredsStatsPack)  # Our Very Own
library(magrittr)
library(data.table)
library(ggplot2)

##### Initial Conditions ####
# Poisson Probability
# We know lambda = 1
lambda <- 1
p <- function(k){lambda^k * exp(-lambda)/factorial(k)}

# Transition matrix
transition_matrix <- matrix(
  c(p(0), 0, 0, 0, 1 - p(0),
    p(1), p(0), 0, 0, 1 - sum(sapply(c(0:1), p)),
    p(2), p(1), p(0), 0, 1 - sum(sapply(c(0:2), p)),
    p(3), p(2), p(1), p(0), 1 - sum(sapply(c(0:3), p)),
    p(4), p(3), p(2), p(1), 1 - sum(sapply(c(1:4), p))
    ),
  byrow = TRUE,
  nrow = 5,
  ncol = 5)

# We take as the initial condition the last row
#   of the transition matrix
initial_state <- transition_matrix[5, ]

# Display transition matrix
kable(transition_matrix, caption = "Transition Matrix")

##### Markov Chain Matrix ####
state_mat <- matrix(nrow = 11, ncol = 5,
                    dimnames = list(
                      c("Initial Week", paste("Week", c(1:10))),
                      c("1 Tractor", paste(c(2:5), "Tractors"))))

# Start with initial condition
# Fill matrix with weekly position
for (week in c(0:10)) {
```

```r
  state_mat[week+1, ] <- round(Freds_MarkovChain(initial_state, transition_matrix, week) * 100, 3)
}

# x-axis for our plot
week <- data.table(c(0:10))
colnames(week) <- "Week"

##### Inventory Projection Table ####
kable(state_mat, caption = "Inventory Probability Matrix")

##### Plot Markov Chain ####
state_mat %>%
  cbind(week) %>%
  as.data.table() %>%
  melt.data.table(id.vars = "Week") %>%
  ggplot(aes(x = Week,
             y = value,
             color = variable)) +
  geom_line() +
  xlab("Week") +
  ylab("Probability of k Tractors Week Start") +
  ggtitle("Tractor Inventory Projections")

##### Determine Steady State ####
# Find the steady state for tolerance 10e-17
steady_state <- Freds_MarkovChain_SteadyState(initial_state, transition_matrix, -17, 100)

steady_state <- matrix(steady_state,
                       nrow = 1,
                       dimnames = list(
                         "",
                         c("1 Tractor", paste(c(2:5), "Tractors"))
                       ))

##### Format Steady State nicely #####
kable(steady_state, caption = "Tractor Steady State")

##### Over Demand Vector ####
overdemand <-
  c(
    1 - sum(sapply(c(0:1), p)),
    1 - sum(sapply(c(0:2), p)),
    1 - sum(sapply(c(0:3), p)),
    1 - sum(sapply(c(0:4), p)),
    1 - sum(sapply(c(0:4), p))
    )

overdemand <- matrix(overdemand,
                     nrow = 1,
                     dimnames = list(
                       "",
                       c("1 Tractor", paste(c(2:5), "Tractors"))
                     ))
```

```r
##### Display Overdemand Nicely ####
kable(overdemand, caption = "Overdemand Probabilities")

##### What Percent of sales did we miss? ####
missed_sales <- sum(steady_state * overdemand)
##### Model Sensitivity #####
# How sensitive is our model to our initial assumptions
i <- 1
sensitivity_matrix <- matrix(
  c(1, round(missed_sales * 100, 3)),
    nrow = 5,
    ncol = 2,
   byrow = TRUE
  )
colnames(sensitivity_matrix) <- c("Weekly Sales", "Expected Missed Sales")

##### Evaluate Sensitivity ####
for (lambda in c(0.50, 0.75, 1.25, 1.50)) {
  ##### Sensitivity: Transition Matrix####
  transition_matrix <- matrix(
  c(p(0), 0, 0, 0, 1 - p(0),
    p(1), p(0), 0, 0, 1 - sum(sapply(c(0:1), p)),
    p(2), p(1), p(0), 0, 1 - sum(sapply(c(0:2), p)),
    p(3), p(2), p(1), p(0), 1 - sum(sapply(c(0:3), p)),
    p(4), p(3), p(2), p(1), 1 - sum(sapply(c(1:4), p))
    ),
  byrow = TRUE,
  nrow = 5,
  ncol = 5)

  ##### Sensitivity: Initial State ####
  initial_state <- transition_matrix[5, ]

  ##### Sensitivity: Steady State ####
  steady_state <- Freds_MarkovChain_SteadyState(initial_state, transition_matrix, -10, 100)

  ##### Sensitivity: Over Demand ####
  overdemand <-
  c(
    1 - sum(sapply(c(0:1), p)),
    1 - sum(sapply(c(0:2), p)),
    1 - sum(sapply(c(0:3), p)),
    1 - sum(sapply(c(0:4), p)),
    1 - sum(sapply(c(0:4), p))
    )

  ##### Sensitivity: Missed Sales ####
  missed_sales <- sum(steady_state * overdemand)

  sensitivity_matrix[i+1, 1] <- lambda
  sensitivity_matrix[i+1, 2] <- round(missed_sales * 100, 3)
  i = i + 1
}
```

```r
##### Display Sensitivity ####
kable(sensitivity_matrix, caption = "Sales Expectations")

##### Sensitivity Chart ####
sensitivity_matrix %>%
  as.data.frame() %>%
  ggplot(aes(x = `Weekly Sales`,
             y = `Expected Missed Sales`)) +
  geom_point() +
  geom_line() +
  ggtitle("Model Sensitivity to Variations in the Rate of Expected Sales")
```