

Introduction to Modeling - HW1

Sharon Morris

8/31/2017

Page 8 #10

10. Your grandparents have an annuity. The value of the annuity increases each month by an automatic deposit of 1% interest on the previous month's balance. Your grandparents withdraw \$1000 at the beginning of each month for living expenses. Currently, they have \$50,000 in the annuity. Model the annuity with a dynamical system. Will the annuity run out of money? When? Hint: What value will it have when the annuity is depleted?

$$a_0 = 50000$$

$$a_{n+1} = a_n + 0.01a_n - 1000$$

```
#Annuity function
annuity_model <- function(month) {
  return (month * 0.01 - 1000)
}

current_value <- 50000 #original value
month_num <- 1

while (TRUE) {
  current_value[month_num+1] <- current_value[month_num] +
    annuity_model(current_value[month_num])
  if(current_value[month_num+1] < 0) break
  month_num <- month_num + 1
}
print(month_num)
```

```
## [1] 70
```

The annuity will run out of money at month 70

Page 34 #13

Consider the spreading of a rumor through a company of 1000 employees, all working in the same building. We assume that the spreading of a rumor is similar to the spreading of a contagious disease (see Example 3, Section 1.2) in that the number of people hearing the rumor each day is proportional to the product of the number who have heard the rumor previously and the number who have not heard the rumor. This is given by

$$r_{n+1} = rn + kr_n(1000 - n)$$

where k is a parameter that depends on how fast the rumor spreads and n is the number of days. Assume $k = 0.001$ and further assume that four people initially have heard the rumor. How soon will all 1000 employees have heard the rumor?

```
k <- 0.001
total_emp <- 1000
r_zero <- 4
```

```

n <- 1

rumor_rate <- r_zero

while (rumor_rate[n] < total_emp) {
  rumor_rate[n + 1] <- rumor_rate[n] + (k*(rumor_rate[n]) * (1000 - rumor_rate[n]))
  n <- n + 1
}

(output <- data.frame(seq(1:n), rumor_rate))

```

```

##      seq.1.n. rumor_rate
## 1          1      4.00000
## 2          2      7.98400
## 3          3     15.90426
## 4          4     31.55557
## 5          5     62.11538
## 6          6    120.37244
## 7          7    226.25535
## 8          8   401.31922
## 9          9   641.58132
## 10         10   871.53605
## 11         11   983.49701
## 12         12  999.72765
## 13         13  999.99993
## 14         14 1000.00000
## 15         15 1000.00000

```

When rounded up, it takes 12 days for 1,000 people to hear the rumor.

Page 55 #6

An economist is interested in the variation of the price of a single product. It is observed that a high price for the product in the market attracts more suppliers. However, increasing the quantity of the product supplied tends to drive the price down. Over time, there is an interaction between price and supply. The economist has proposed the following model, where P_n represents the price of the product at year n , and Q_n represents the quantity. Find the equilibrium values for this system.

$$P_{n+1} = P_n - 0.1(Q_n - 500)$$

$$Q_{n+1} = Q_n + 0.2(P_n - 100)$$

a. Does the model make sense intuitively? What is the significance of the constants 100 and 500? Explain the significance of the signs of the constants -0.1 and 0.2.

Yes, the model does make sense intuitively

The significance of the the constants 100 and 500 – they are the minimum quantity and price

The significance of the signs of the constants -0.1 and 0.2, if the quantity is above 500 the price goes down when the reverse happens if it is below 100.

b. Test the initial conditions in the following table and predict the long-term behavior.

```

price_min <- function(p, q) {
  return (p-(0.1*(q-500)))
}

```

| | Price | Quantity |
|--------|-------|----------|
| Case A | 100 | 500 |
| Case B | 200 | 500 |
| Case C | 100 | 600 |
| Case D | 100 | 400 |

```

quantity_min <- function(p, q) {
  return (q+(0.2*(p-100)))
}

case_func <- function(p_zero, q_zero, n) {
  p <- p_zero
  q <- q_zero

  for (i in 1:n) {
    p[i + 1] <- price_min(p[i], q[i])
    q[i + 1] <- quantity_min(p[i], q[i])
  }
  return (data.frame(n=seq(1:(n+1)), price = p, quantity = q))
}

#Case A
library(ggplot2)

```

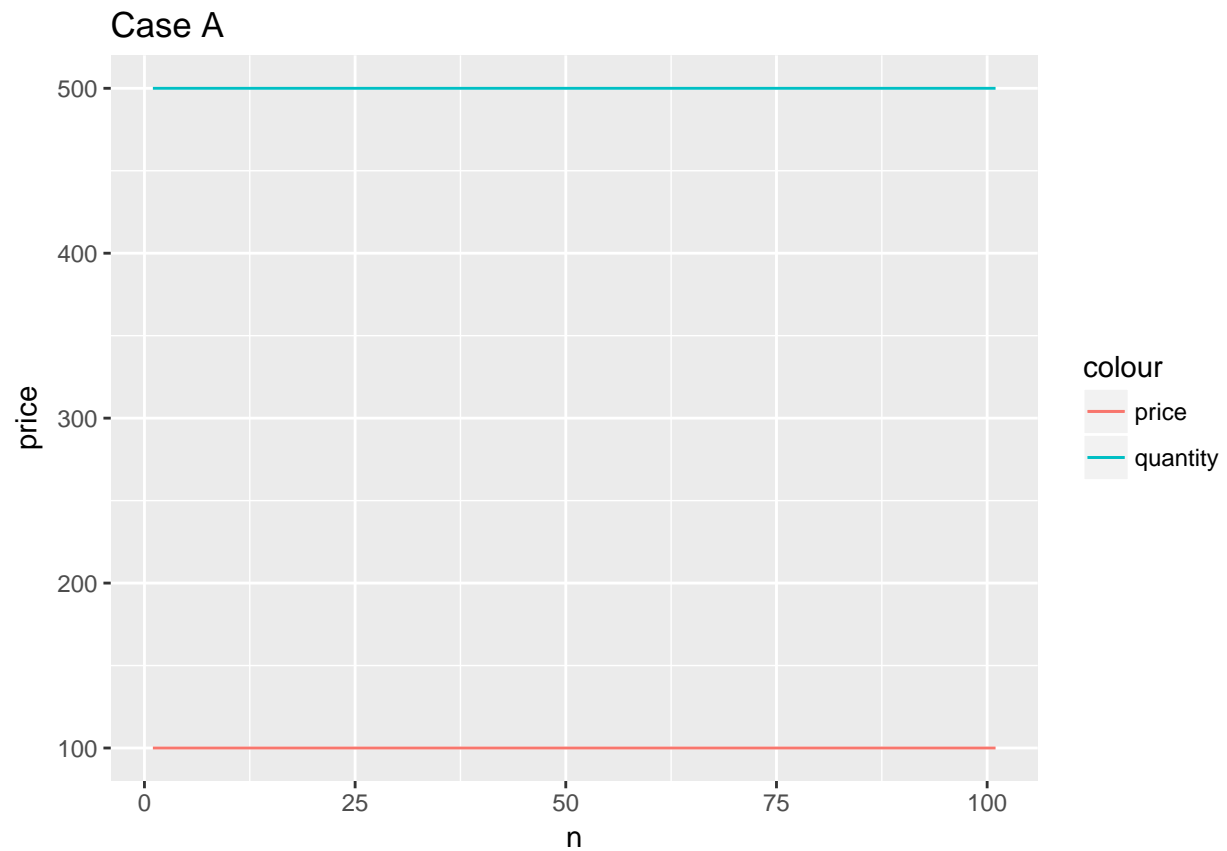
```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```

case_A <- case_func(100, 500, 100)

ggplot(case_A, aes(n)) +
  geom_line(aes(y = price, colour = "price")) +
  geom_line(aes(y = quantity, colour = "quantity")) +
  ggtitle("Case A")

```

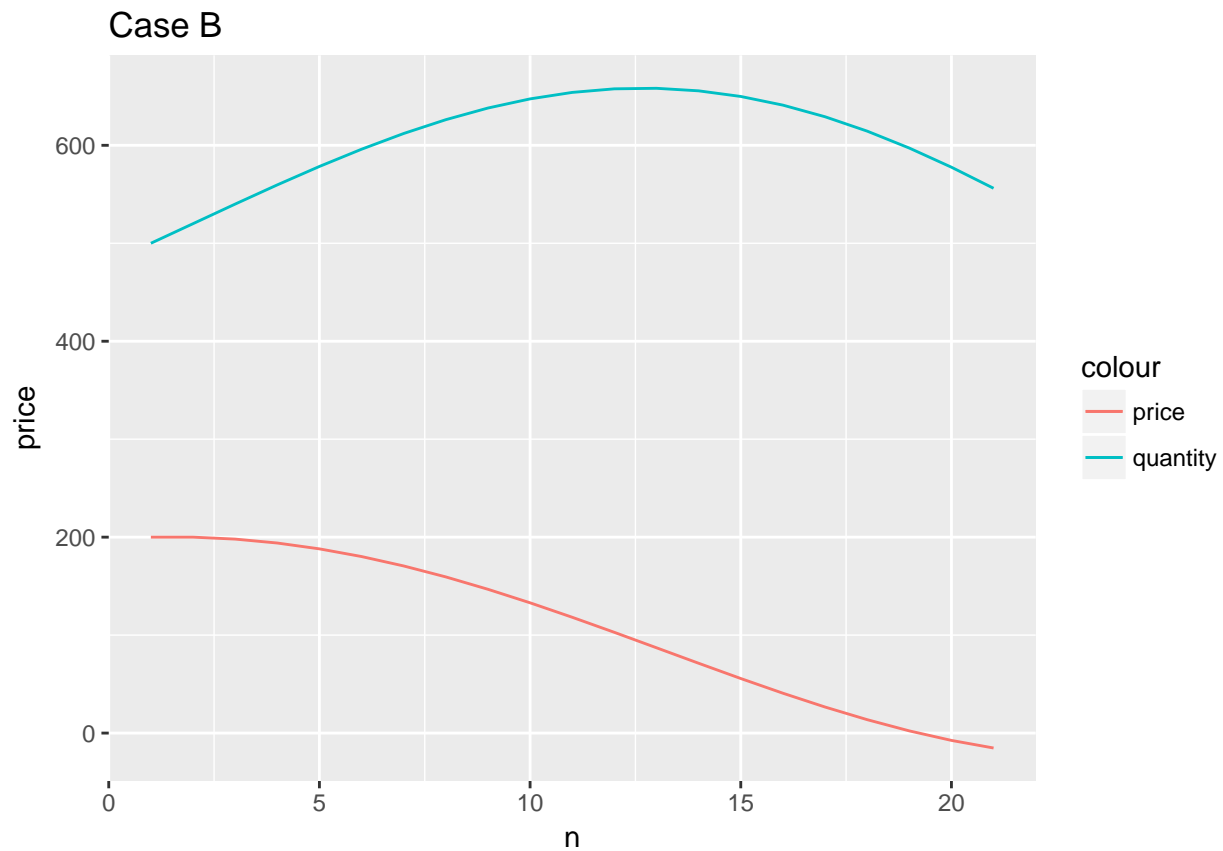


With values 100 and 500 the price and quantity remain constant

Case B

```
case_B <- case_func(200, 500, 20)

ggplot(case_B, aes(n)) +
  geom_line(aes(y = price, colour = "price")) +
  geom_line(aes(y = quantity, colour = "quantity")) +
  ggtitle("Case B")
```

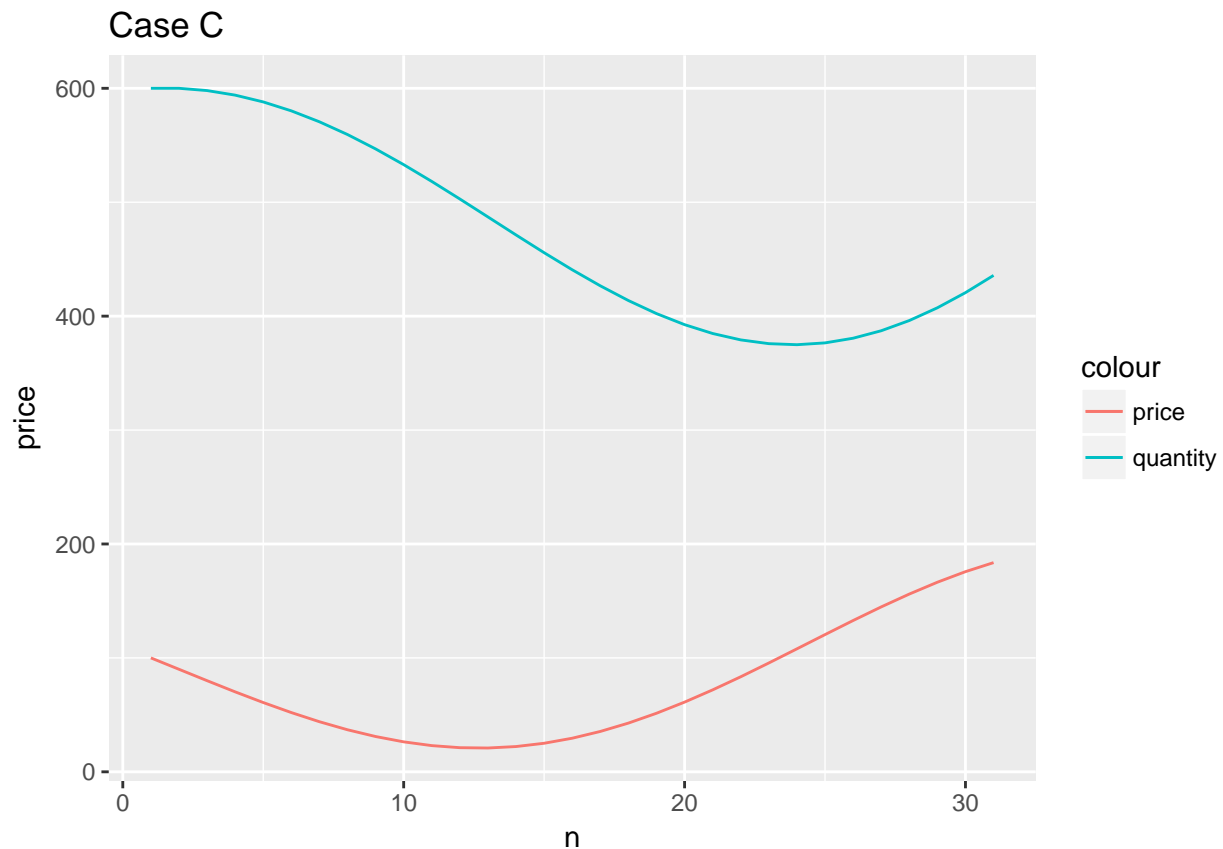


As the quantity go up the price go down Price and quantity go up it plateaus then price decreases as the quantity increase

Case C

```
case_C <- case_func(100, 600, 30)

ggplot(case_C, aes(n)) +
  geom_line(aes(y = price, colour = "price")) +
  geom_line(aes(y = quantity, colour = "quantity")) +
  ggtitle("Case C")
```



The price begins to decrease as the quantity increases until it reaches a turning point where price begins to increase as the quantity increases

For the most part, as the quantity increases the price decreases until it reaches the lowest point then the price begins to increase as the quantity increases

Case D

```
case_D <- case_func(100, 400, 50)

ggplot(case_D, aes(n)) +
  geom_line(aes(y = price, colour = "price")) +
  geom_line(aes(y = quantity, colour = "quantity")) +
  ggtitle("Case D")
```

