

# Week 1 hw

*Charley Ferrari*

*Tuesday, September 01, 2015*

## Page 8, Question 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.

$$a_{n+1} = 1.01a_n - 1000$$

Will the annuity run out of money? When?

```
anndf <- data.frame(period=0, amount=50000)

while(tail(anndf,1)$amount > 0){
  anndf <- rbind(anndf, c(tail(anndf,1)$period+1, 1.01*tail(anndf,1)$amount - 1000))
}

anndf
```

```
##      period      amount
## 1         0 50000.0000
## 2         1 49500.0000
## 3         2 48995.0000
## 4         3 48484.9500
## 5         4 47969.7995
## 6         5 47449.4975
## 7         6 46923.9925
## 8         7 46393.2324
## 9         8 45857.1647
## 10        9 45315.7364
## 11       10 44768.8937
## 12       11 44216.5827
## 13       12 43658.7485
## 14       13 43095.3360
## 15       14 42526.2893
## 16       15 41951.5522
## 17       16 41371.0678
## 18       17 40784.7784
## 19       18 40192.6262
## 20       19 39594.5525
## 21       20 38990.4980
## 22       21 38380.4030
## 23       22 37764.2070
## 24       23 37141.8491
## 25       24 36513.2676
## 26       25 35878.4002
```

## 27	26	35237.1843
## 28	27	34589.5561
## 29	28	33935.4517
## 30	29	33274.8062
## 31	30	32607.5542
## 32	31	31933.6298
## 33	32	31252.9661
## 34	33	30565.4957
## 35	34	29871.1507
## 36	35	29169.8622
## 37	36	28461.5608
## 38	37	27746.1764
## 39	38	27023.6382
## 40	39	26293.8746
## 41	40	25556.8133
## 42	41	24812.3815
## 43	42	24060.5053
## 44	43	23301.1103
## 45	44	22534.1214
## 46	45	21759.4626
## 47	46	20977.0573
## 48	47	20186.8278
## 49	48	19388.6961
## 50	49	18582.5831
## 51	50	17768.4089
## 52	51	16946.0930
## 53	52	16115.5539
## 54	53	15276.7095
## 55	54	14429.4766
## 56	55	13573.7713
## 57	56	12709.5090
## 58	57	11836.6041
## 59	58	10954.9702
## 60	59	10064.5199
## 61	60	9165.1651
## 62	61	8256.8167
## 63	62	7339.3849
## 64	63	6412.7787
## 65	64	5476.9065
## 66	65	4531.6756
## 67	66	3576.9923
## 68	67	2612.7623
## 69	68	1638.8899
## 70	69	655.2788
## 71	70	-338.1684

Starting with month 0, the annuity will end up with \$655.28 in month 69.

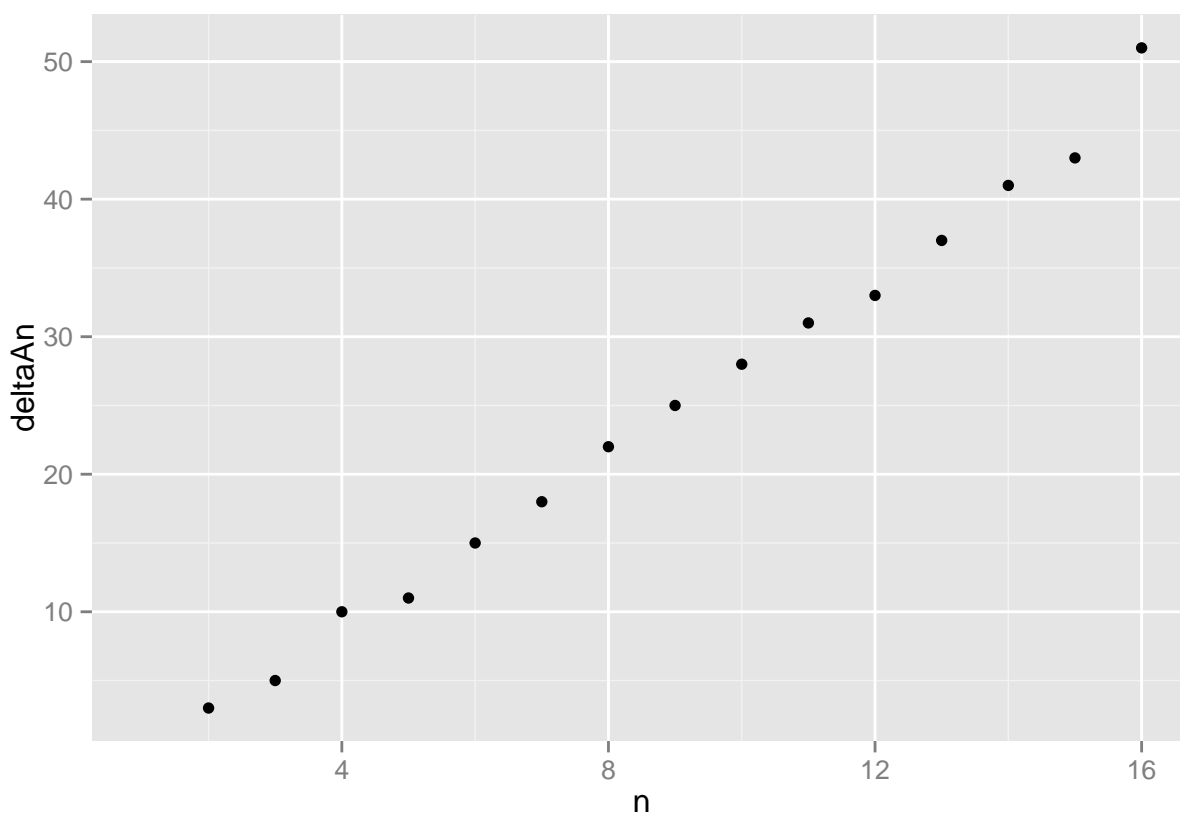
### Page 17, Question 9

The data in the accompanying table show the speed  $n$  (in increments of 5 mph) of an automobile and the associated distance  $a_n$  in feet required to stop it once the brakes are applied. For instance,  $n = 6$  (representing  $6 \times 5 = 30\text{mph}$ ) requires a stopping distances of  $a_6 = 47\text{ft}$

- a. Calculate and plot the change  $\Delta a_n$  versus  $n$ . Does the graph reasonably approximate a linear relationship?

```
brakedf <- data.frame(n=1:16, an=c(3,6,11,21,32,47,65,87,112,140,171,204,241,282,325,376))  
brakedf$deltaAn = c(NA,tail(brakedf$an,-1)-head(brakedf$an,-1))  
library(ggplot2)  
ggplot(brakedf, aes(x=n, y=deltaAn)) + geom_point()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



The graph does reasonably approximate a linear relationship.

- b. Based on your conclusions in part (a), find a difference equation model for the stopping distance data. Test your model by plotting the errors in the predicted values against  $n$ . Discuss the appropriateness of the model.

Since this is linear, we're trying to find a line. We can also assume that if  $n = 0$ , the stopping distance would be 0, and therefore the line passes through the origin.

To approximate the slope of this line, let's average the slopes from the origin to each of the points. In other words, for each point, let's calculate  $\frac{\Delta a_n}{n}$ .

```
brakedf$slope <- brakedf$deltaAn / brakedf$n
slope <- mean(brakedf$slope, na.rm=TRUE)
```

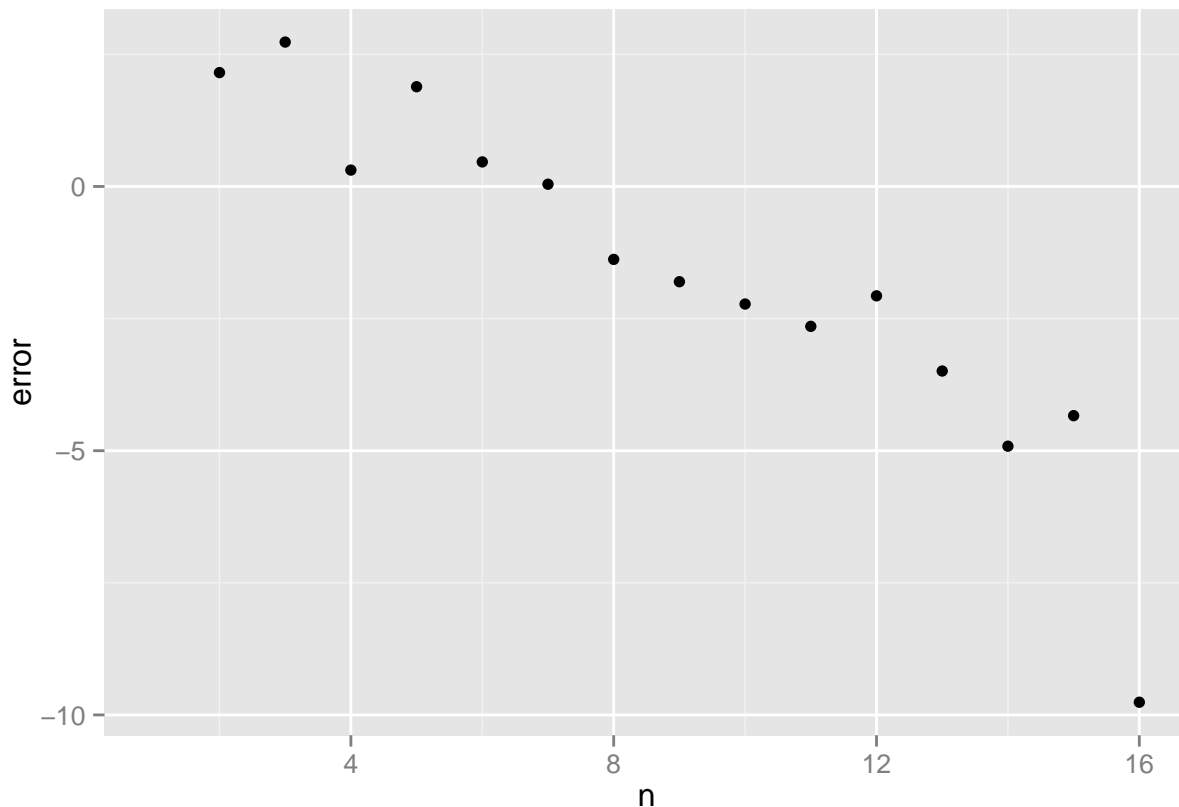
We could easily find the least squares solution to this problem, forcing the intercept to be 0, but the textbook was solving these problems in an approximate way by calculating the slope

```
altSlope <- summary(lm(deltaAn ~ 0 + n, data=brakedf))$coefficients["n", "Estimate"]
```

Lets plot the errors in the predicted values vs n

```
brakedf$estimate <- slope*brakedf$n
brakedf$error <- brakedf$estimate - brakedf$deltaAn
ggplot(brakedf, aes(x=n, y=error)) + geom_point()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



This graph suggests that there is a pattern to the errors. This model may not be appropriate. If we were just fitting  $\Delta a_n$  to  $n$  using a linear model including an intercept, we could perhaps end up with a better fit.

### Page 34, Question 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 (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:

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

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?

```
rumordf <- data.frame(n=0, r=4, roundr=4)

k <- 0.001

for(i in 1:13){
  rumordf <- rbind(rumordf, c(i, rumordf$r[i] + k*rumordf$r[i]*(1000 - rumordf$r[i]),
                             round(rumordf$roundr[i] + k*rumordf$roundr[i]*(1000 -
                                                                           rumordf$roundr[i]))))
}

rumordf
```

##	n	r	roundr
## 1	0	4.00000	4
## 2	1	7.98400	8
## 3	2	15.90426	16
## 4	3	31.55557	32
## 5	4	62.11538	63
## 6	5	120.37244	122
## 7	6	226.25535	229
## 8	7	401.31922	406
## 9	8	641.58132	647
## 10	9	871.53605	875
## 11	10	983.49701	984
## 12	11	999.72765	1000
## 13	12	999.99993	1000
## 14	13	1000.00000	1000

I created two columns here, one that rounds the number of people who have heard the rumor to the nearest integer. Given the initial condition of four people who have heard the rumor, this doesn't make too big a difference. After 12 days, the rumor will have spread throughout the office.

### Page 55, Question 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 constants -0.1 and 0.2.

See explanation below.

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

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

```
library(reshape2)

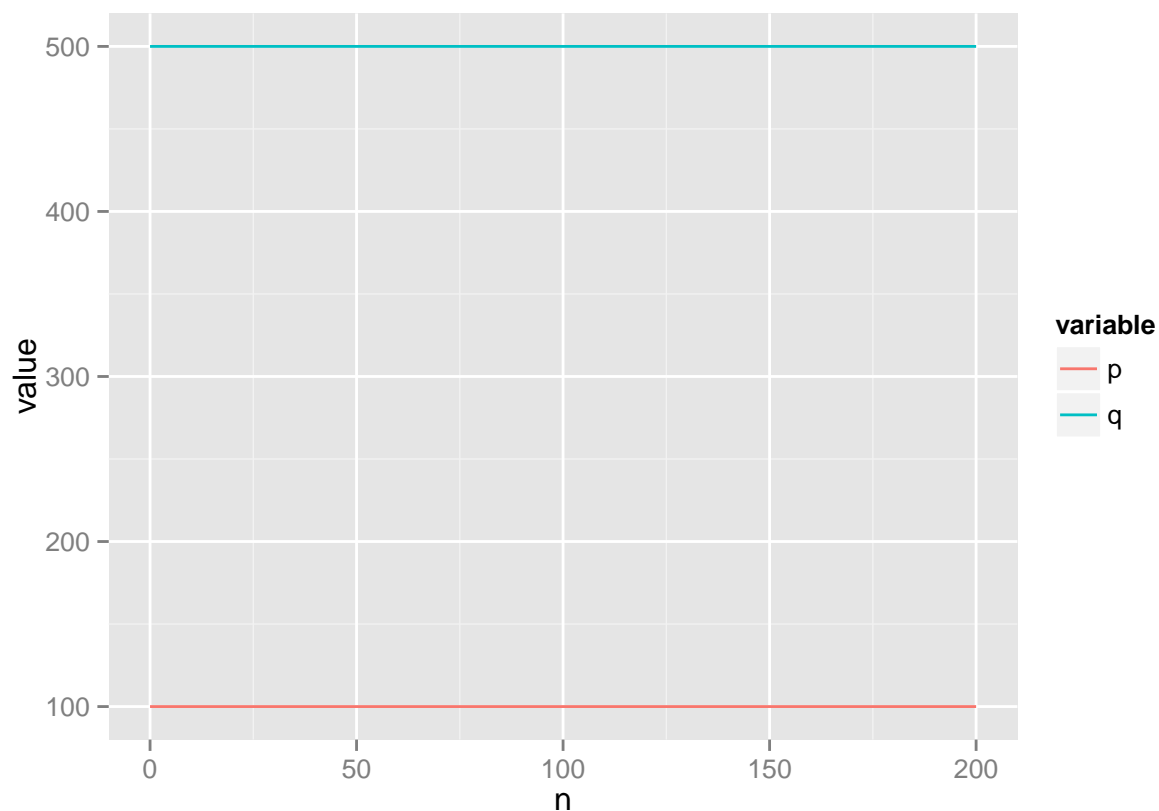
iterations <- 200

pinit <- 100
qinit <- 500

econdf <- data.frame(n = 0, p = pinit, q = qinit)

for(i in 1:iterations){
  econdf <- rbind(econdf,
    c(i, tail(econdf$p,1) - 0.1*(tail(econdf$q,1) - 500),
      tail(econdf$q,1) + 0.2*(tail(econdf$p,1) - 100)))
}

ggplot(melt(econdf, id="n"), aes(x=n, y=value, color=variable)) + geom_line()
```



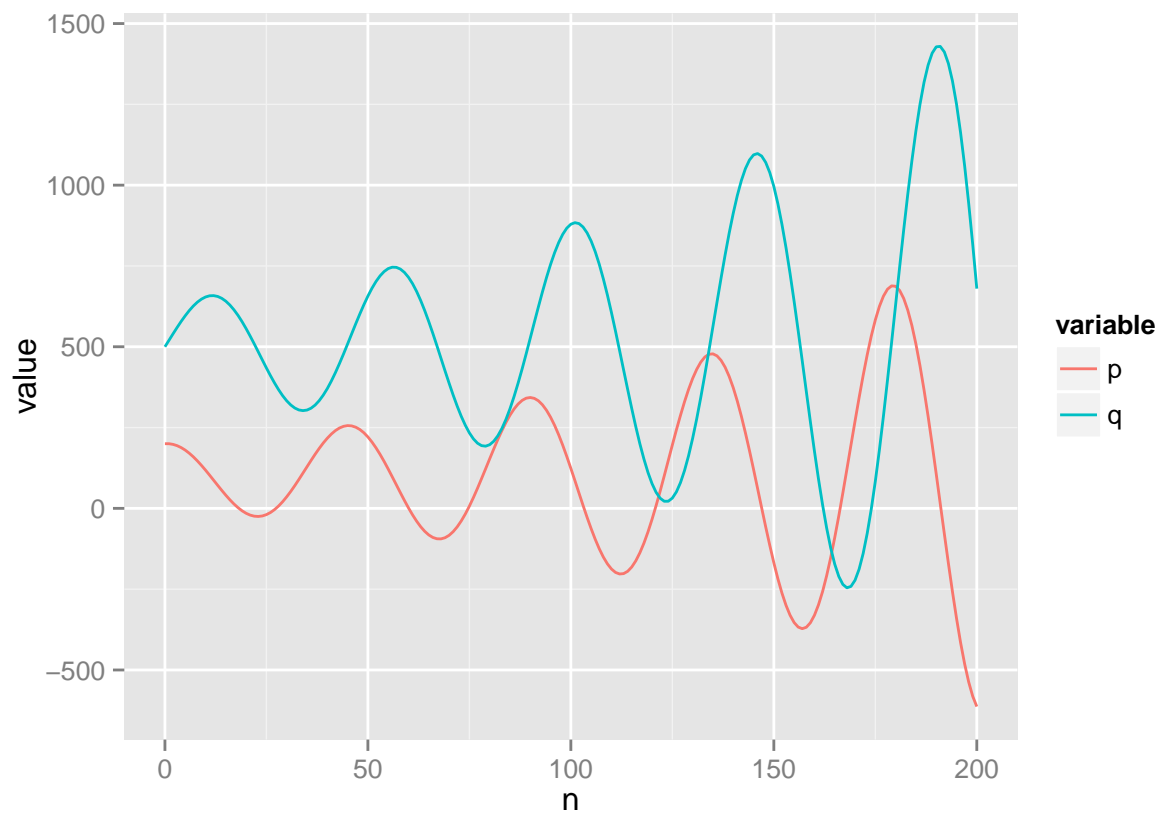
```
#####

pinit <- 200
qinit <- 500

econdf <- data.frame(n = 0, p = pinit, q= qinit)

for(i in 1:iterations){
  econdf <- rbind(econdf,
                  c(i, tail(econdf$p,1) - 0.1*(tail(econdf$q,1) - 500),
                    tail(econdf$q,1) + 0.2*(tail(econdf$p,1) - 100)))
}

ggplot(melt(econdf, id="n"), aes(x=n, y=value, color=variable)) + geom_line()
```



```
#####

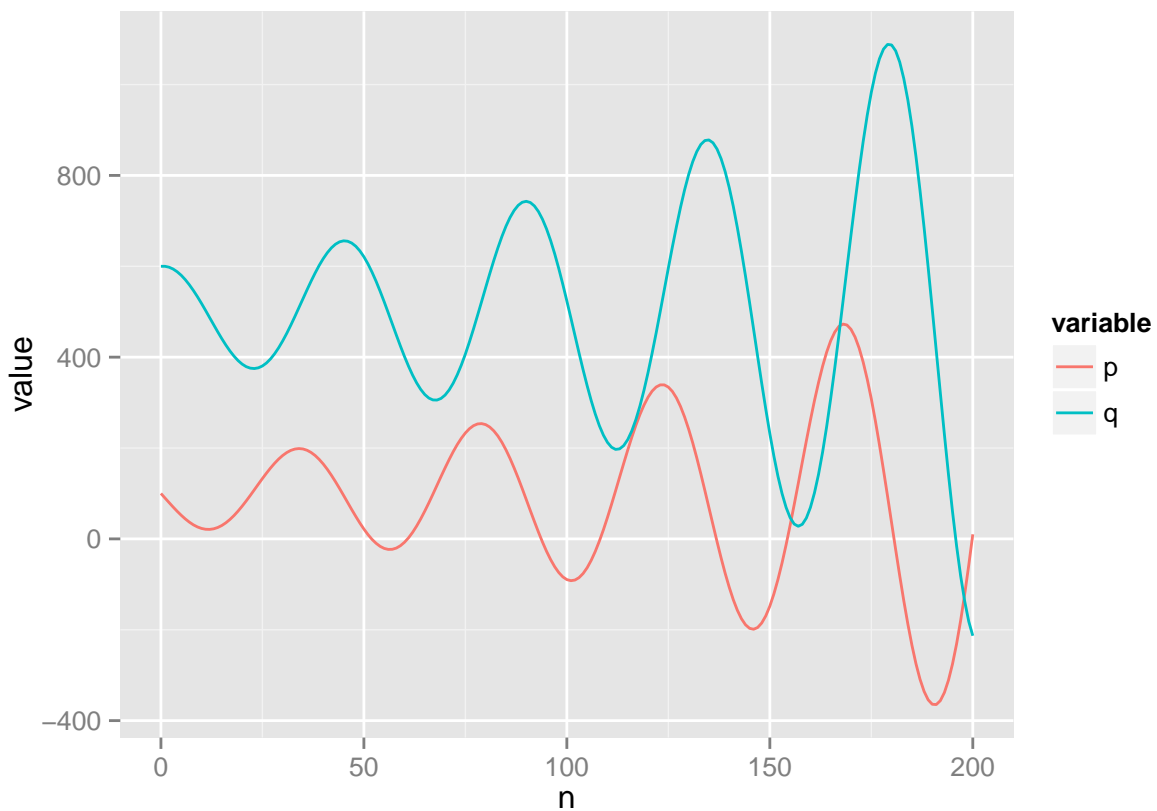
pinit <- 100
qinit <- 600

econdf <- data.frame(n = 0, p = pinit, q= qinit)

for(i in 1:iterations){
  econdf <- rbind(econdf,
    c(i, tail(econdf$p,1) - 0.1*(tail(econdf$q,1) - 500),
      tail(econdf$q,1) + 0.2*(tail(econdf$p,1) - 100)))
}

ggplot(melt(econdf, id="n"), aes(x=n, y=value, color=variable)) + geom_line()
```





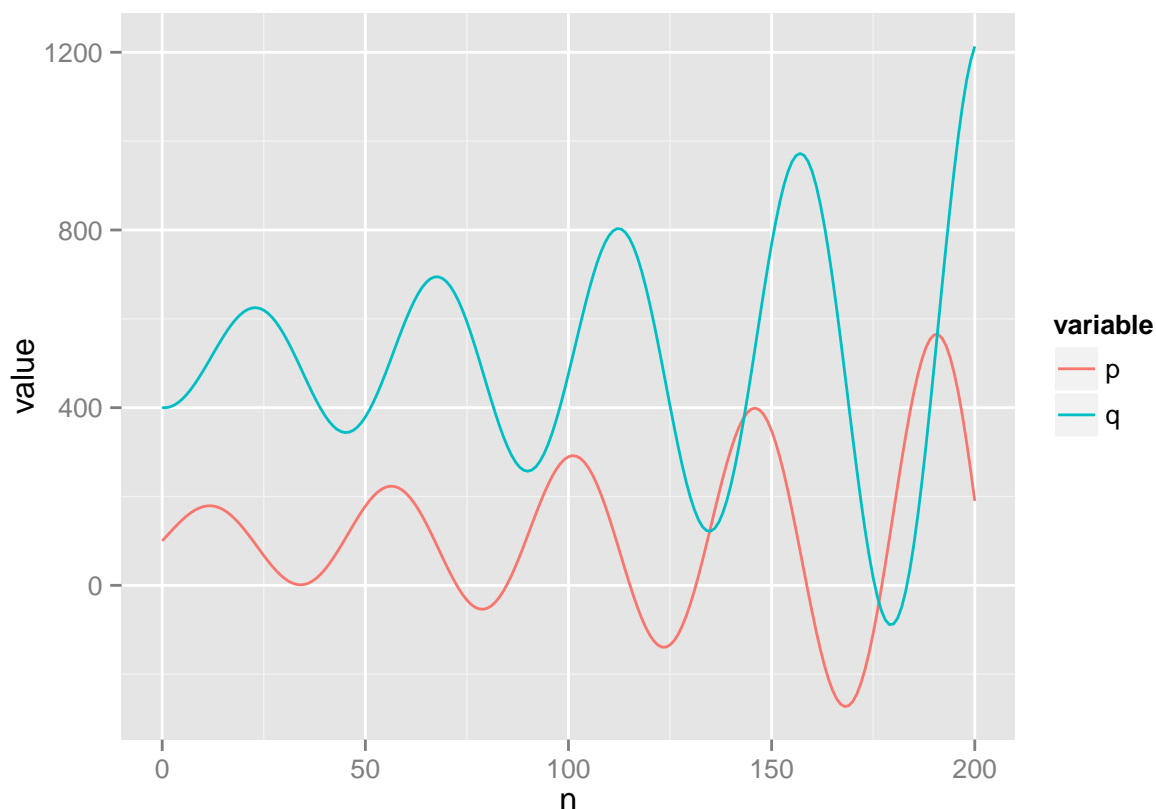
```
#####

pinit <- 100
qinit <- 400

econdf <- data.frame(n = 0, p = pinit, q= qinit)

for(i in 1:iterations){
  econdf <- rbind(econdf,
                  c(i, tail(econdf$p,1) - 0.1*(tail(econdf$q,1) - 500),
                    tail(econdf$q,1) + 0.2*(tail(econdf$p,1) - 100)))
}

ggplot(melt(econdf, id="n"), aes(x=n, y=value, color=variable)) + geom_line()
```



Seeing the graphs made me realize what was going on intuitively. At first I assumed this system would result in an equilibrium price and quantity. While this is possible given the right initial conditions (a price of 100 and a quantity of 500), any other values will cause the system to oscillate without bound around the equilibrium.

The difference in sign between 0.1 and 0.2 in the two equations is what causes the oscillation. If these were both positive or both negative, they would increase or decrease without bound most of the time (this depends on what the initial price and quantity are).

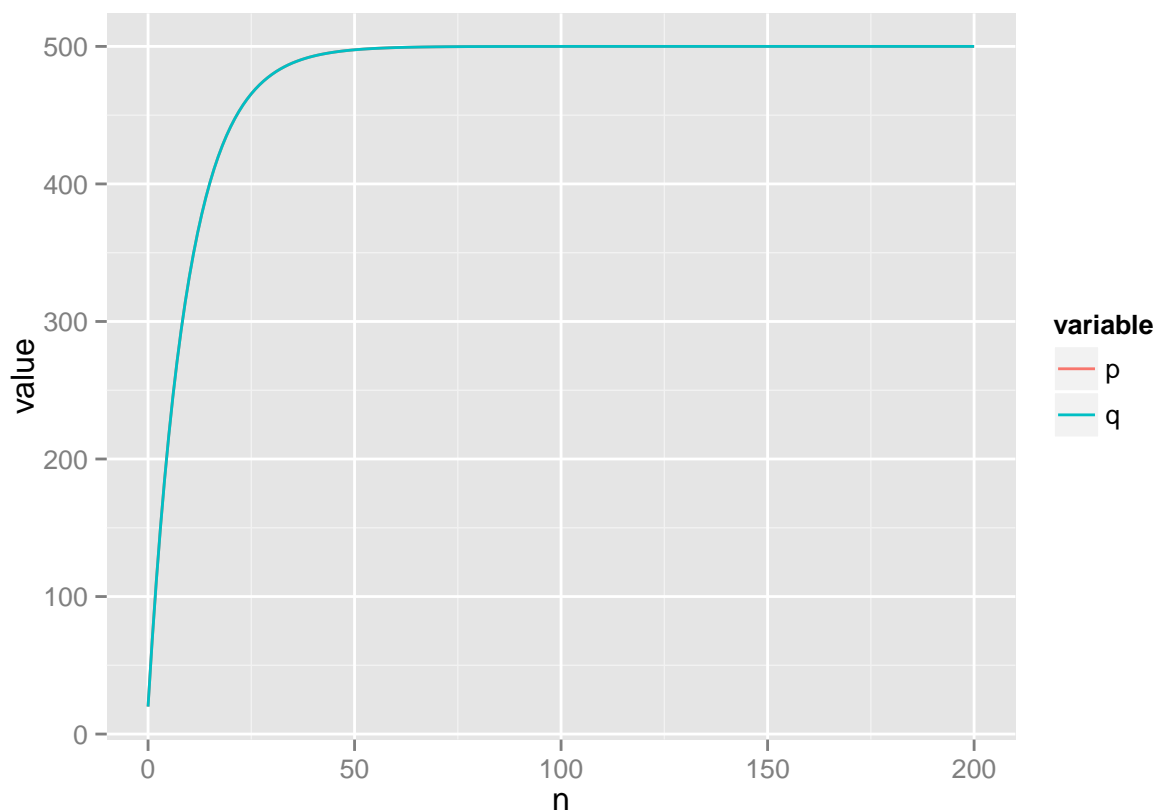
The one equilibrium case I found was if the equations are the same, the second term is being subtracted from the first, and the initial price and quantity are the same:

```
pinit <- 20
qinit <- 20

econdf <- data.frame(n = 0, p = pinit, q = qinit)

for(i in 1:iterations){
  econdf <- rbind(econdf,
    c(i, tail(econdf$p,1) - 0.1*(tail(econdf$q,1) - 500),
      tail(econdf$q,1) - 0.1*(tail(econdf$p,1) - 500)))
}

ggplot(melt(econdf, id="n"), aes(x=n, y=value, color=variable)) + geom_line()
```



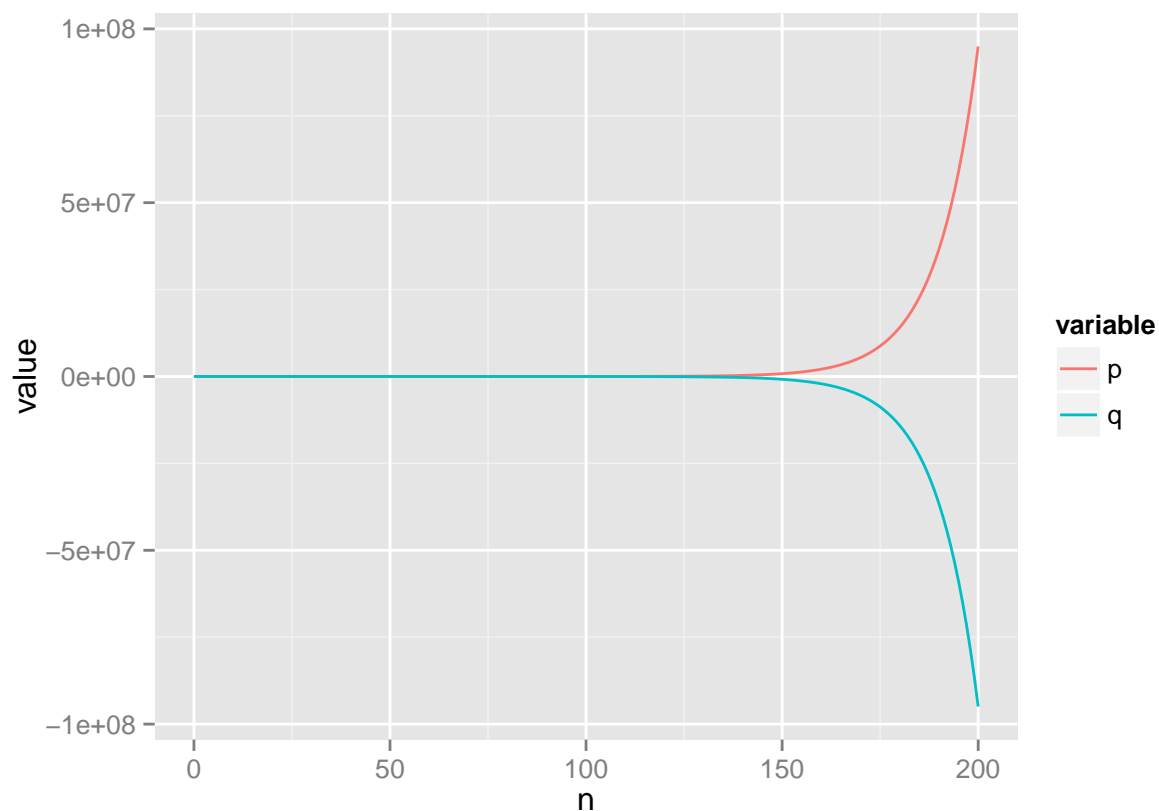
Any differences from this specific set of cases will once again cause divergence. For example, in the above case if the initial price is not equal to the initial quantity:

```
pinit <- 21
qinit <- 20

econdf <- data.frame(n = 0, p = pinit, q= qinit)

for(i in 1:iterations){
  econdf <- rbind(econdf,
    c(i, tail(econdf$p,1) - 0.1*(tail(econdf$q,1) - 500),
      tail(econdf$q,1) - 0.1*(tail(econdf$p,1) - 500)))
}

ggplot(melt(econdf, id="n"), aes(x=n, y=value, color=variable)) + geom_line()
```



This is probably not a good model for the interaction between price and quantity in normal markets. However, I could imagine some economic processes that might function in this way. This sort of oscillating behavior reminded me of the “fool in the shower” metaphor for monetary policy, where central banks with imperfect information react to the economy in a delayed fashion, raising and lowering interest rates in a way that’s out of sync with the rest of the economy.