Data 609: Homework 13

Graphs of Functions as Models

Aaron Grzasko

11/17/2017

Load Libraries

```
library(ggplot2)
library(ggthemes)
library(tidyr)
library(dplyr)
library(knitr)
library(cowplot)
library(Hmisc)
library(reconPlots)
```

Page B-13 Question 4

Discuss how you might go about validating the nuclear arms race model. What data would you collect? Is it possible to obtain the data?

To validate the model, we would need to collect historical records from two countries engaged in an arms buildup. We may wish to aggregate data from multiple conflicts during the modern era (e.g. 20th century and later) to establish model assumptions such as survival coefficients. However, rapid technological advances may prevent an aggregated data set from being useful in validating any particular pre-war conflict. Historical data from previous periods may need to be adjusted or excluded for estimating model parameters in later periods.

The following data would be useful for data validation:

- The size of the relevant military arsenal for each country. This quantity would need to be captured at regular time intervals during the time of conflict (e.g. once per year).

- A statistic that captures the per unit destructive capabilities of each country's weapons stockpile. This information may be available in academic or military literature.

Obtaining the necessary data would be a difficult, if not impossible, task. The required records may not be available to the public. Data that are available could be spotty and incomplete, and may require various statistical and judgmental transformations to be of much use for validation procedures.

Page B-17: Question 1

Build a numerical solution to Equations (15.8).

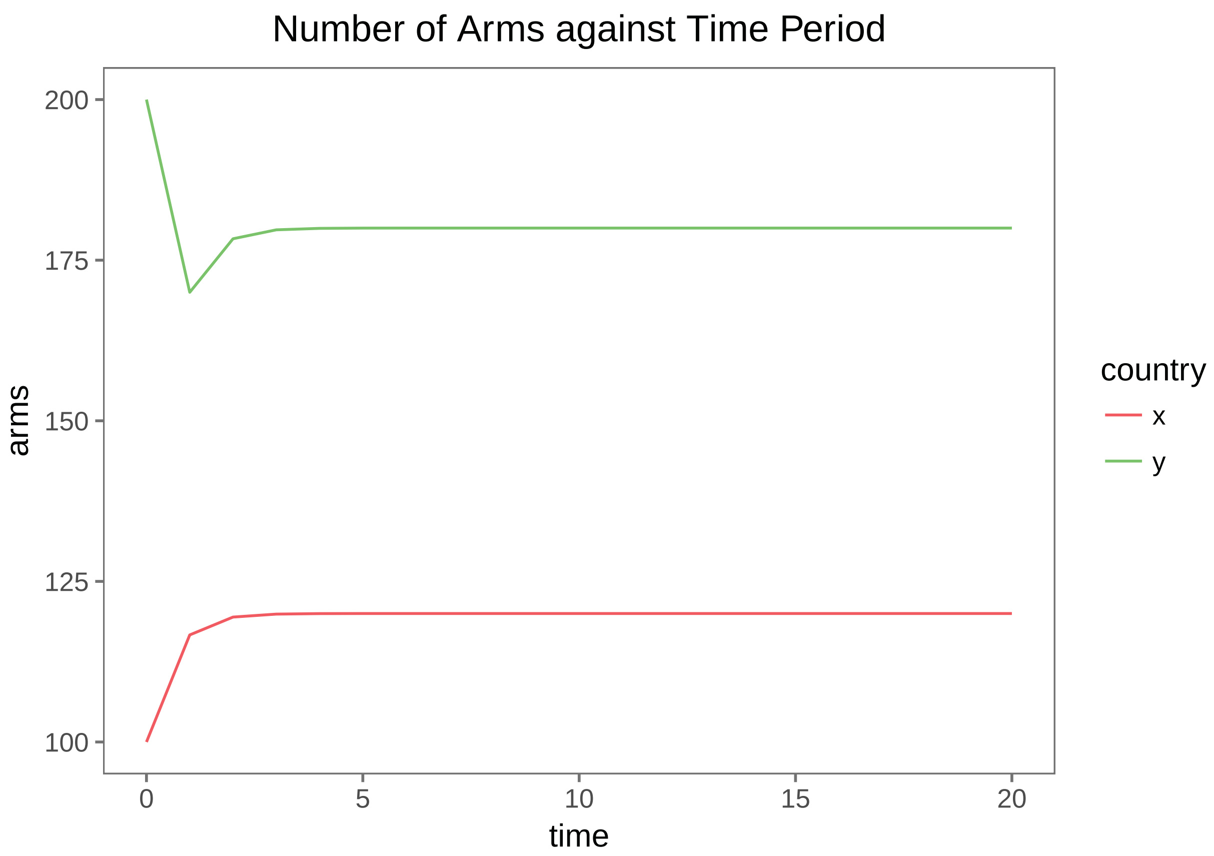
$$\left.\begin{aligned} y_{n+1} &= 120 + \frac{1}{2}x_n \\ x_{n+1} &= 60 + \frac{1}{3}y_n \end{aligned}\right\}$$

with

$$x_0 = 100$$
$$y_0 = 200$$

Part A: Graph your results.

```r
# y_n+1:  Number of arms for country Y
f.1 <- function(x) 120 + 0.5 * x

# x_n+1:  Number of arms for country X
f.2 <- function(y) 60 + 1/3 * y


# function to calculate number of weapons for countries x and y
calc_arms <- function(x0,y0, iter, func1, func2){
    x <- x0
    y <- y0
    t <- 0

    for (i in 1:iter){
        y <- append(y, func1(x[length(x)]))
        x <- append(x, func2(y[length(y)]))
        t <- append(t,i)
    }

    data.frame(t=t,x=x,y=y) %>% gather("country","arms",2:3)

}

# calculate future arms based on initial values of x0=100, y0=200
df <- calc_arms(100,200,20,f.1,f.2)

# plot
ggplot(df, aes(x=t,y=arms,col=country)) + geom_line() + theme_few() + scale_color_few() +
    labs(title="Number of Arms against Time Period", x = "time", y= "arms") +
    theme(plot.title = element_text(hjust = 0.5))
```



Part B: Is an equilibrium value reached?

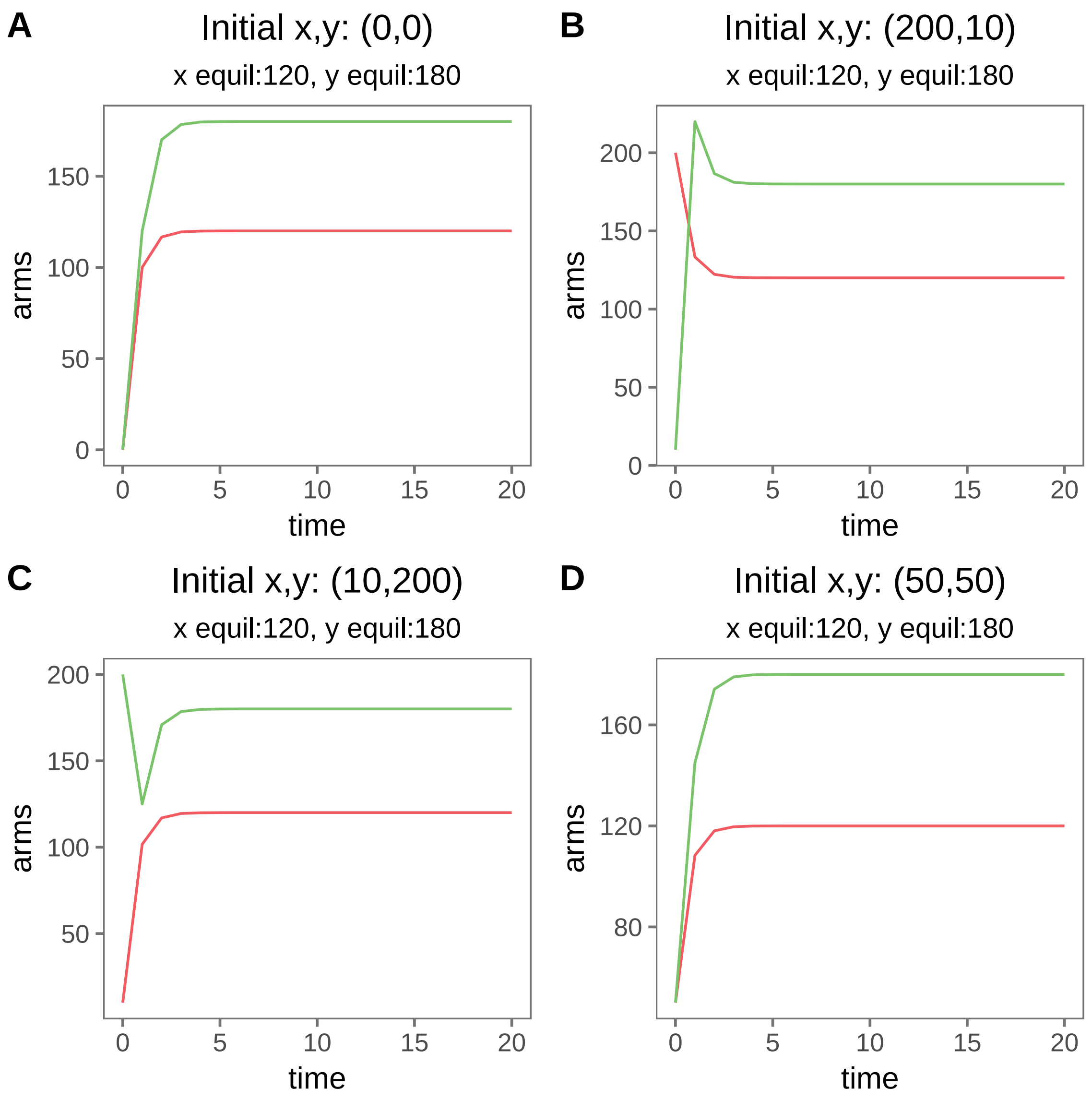
```
# data, wide format
df <- df %>% spread(country, arms)

# approx equilibrium values, t=20
kable(tail(df,1), align="rrr", row.names=F, digits = 2)
```

t	x	y
20	120	180

Yes. Assuming initial arms stocks of 100 and 200 for countries X and Y, respectively, the equilibrium arms values are 120 and 180.

Part C: Try other starting values. Do you think the equilibrium value is stable?

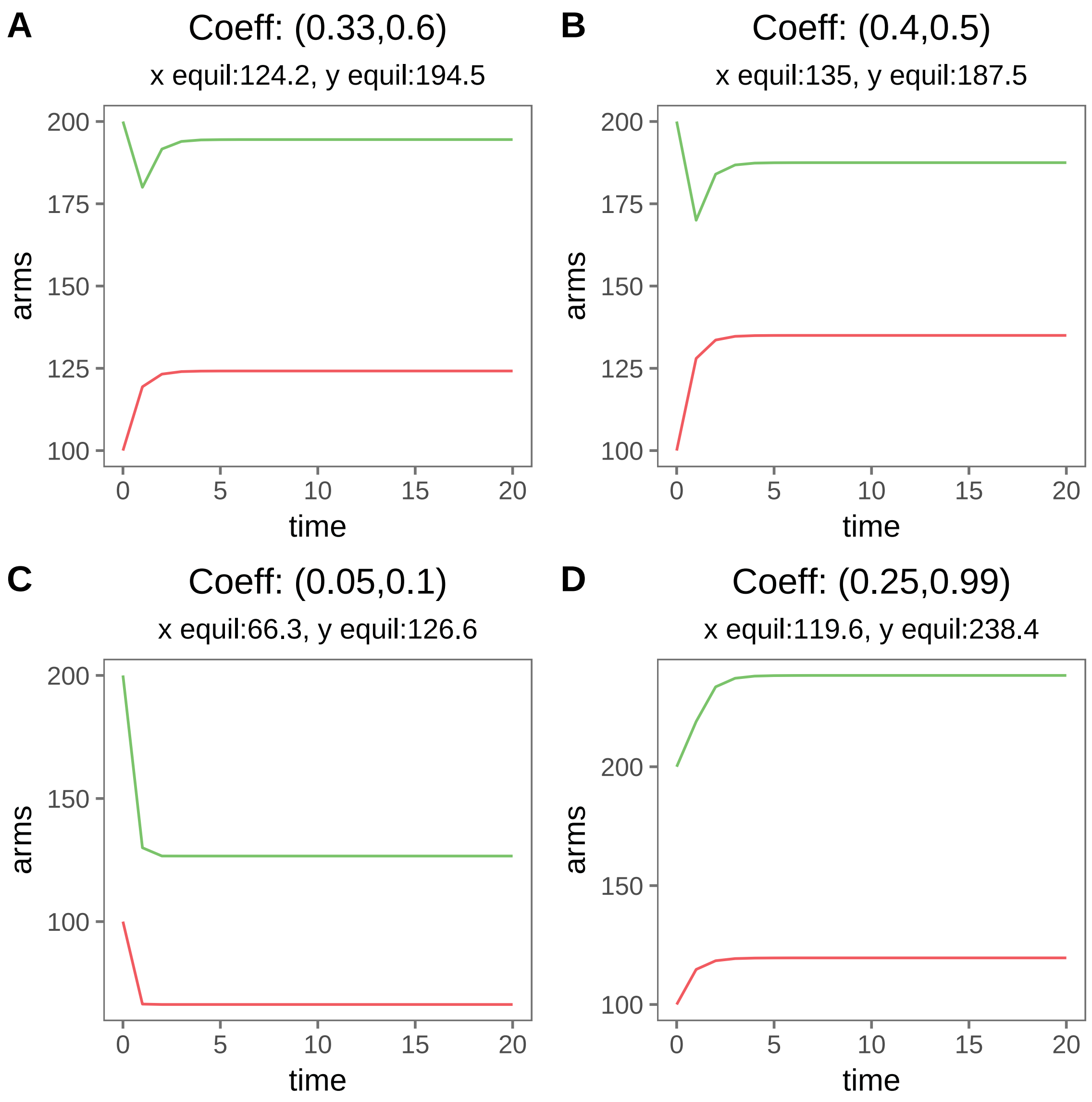
```
# vectors of starting x and y values
x <- c(0,200,10,50)
y <- c(0,10,200,50)

# plot various (x,y) initial conditions
for (i in 1:length(x)) {
  df_name <- paste0("df",x[i],".",y[i])
  var_name <- paste0("g",x[i],".",y[i])
  assign(df_name,calc_arms(x[i],y[i],20,f.1,f.2))
  x_equil <- eval(parse(text=df_name)) %>% filter(country == 'x', t == 20) %>%
      select(arms)

  y_equil <- eval(parse(text=df_name)) %>% filter(country == 'y', t == 20) %>%
      select(arms)

  assign(var_name,
         ggplot(eval(parse(text=df_name)), aes(x=t,y=arms,col=country)) +
           geom_line() + theme_few() + scale_color_few(guide=FALSE) +
           labs(title = paste0("Initial x,y: (",x[i],",",y[i],")"),x = "time", y= "arms",
                subtitle = paste0("x equil:",round(x_equil,1),", y equil:",round(y_equil,1) )) +
           theme(plot.title = element_text(hjust = 0.5),plot.subtitle = element_text(hjust = 0.5))
  )
}

# print plots in grid format
plot_grid(g0.0, g200.10, g10.200, g50.50, ncol = 2, labels = "AUTO",
          align = 'v', label_size = 14)
```



The long-term equilibrium values appear to be stable. In the plots above, we see that the arms counts for countries x and y converge to 120 and 180, respectively, despite varying initial values.

Part D: Explore other values for the survival coefficients of Countries X and Y . Describe your results.

```
# vectors of various survival coefficents
a <- c(0.6,0.5,0.1,0.99)
b <- c(0.33,0.4,0.05,0.25)

# plots for various survival coefficients
for (i in 1:length(a)) {
  df_name <- paste0("df",a[i],"_",b[i])
  var_name <- paste0("g",a[i],"_",b[i])
  assign("f1", function(x) 120+ a[i]*x )
  assign("f2", function(x) 60+ b[i]*x )
  assign(df_name,calc_arms(100,200,20,f1,f2))

  x_equil <- eval(parse(text=df_name)) %>% filter(country == 'x', t == 20) %>%
      select(arms)

  y_equil <- eval(parse(text=df_name)) %>% filter(country == 'y', t == 20) %>%
      select(arms)

  assign(var_name,
         ggplot(eval(parse(text=df_name)), aes(x=t,y=arms,col=country)) +
              geom_line() + theme_few() + scale_color_few(guide=FALSE) +
              labs(title = paste0("Coeff: (",b[i],",",a[i],")"),x = "time", y= "arms",
                   subtitle = paste0("x equil:",round(x_equil,1),", y equil:",round(y_equil,1) )) +
             theme(plot.title = element_text(hjust = 0.5),plot.subtitle = element_text(hjust = 0.5))
 )
}

# print plots in grid format
plot_grid(g0.6_0.33, g0.5_0.4, g0.1_0.05, g0.99_0.25, ncol = 2, labels = "AUTO",
          align = 'v', label_size = 14)
```



Based on the plots above, we see that equilibrium values are highly sensitive to the assumed survival coefficients used in our models.

Page B-21: #4

Verify the result that the marginal revenue of the $(q+1)$st unit equals the price of that unit minus the loss in revenue on previous units resulting from price reduction.

Let

q =quantity demanded
$P(q)$ = avg price, given quantity q
TR =total revenue
$MR(q)$ =marginal revenue

The formula for total revenue is:

$$TR = P(q) \cdot q$$

Using the derivative product rule, we have

$$TR'(q) = MR(q) = P(q) + P'(q)q$$

We can approximate the derivative of price as follows:

$$P'(q) \approx \frac{P(q + \Delta q) - P(q)}{\Delta q}$$

Setting $\Delta q = 1$:

$$P'(q) \approx P(q + 1) - P(q)$$

Finally,

$$MR(q) \approx P(q) + \Big(P(q + 1) - P(q)\Big) \cdot q$$

$$= P(q) - \Big(P(q) - P(q + 1)\Big) \cdot q$$

The line above is equivalent to the average unit price minus the loss of revenue on previous units due to price reduction. This is what we wanted to show.

Note:
If the firm is in a competitive market, $MR(q) = P(q)$ because $P(q) = P(q + 1)$. On the other hand, if the firm has monopoly power, then marginal revenue decreases with increasing quantity, i.e. $P(q) - P(q + 1) > 0$.

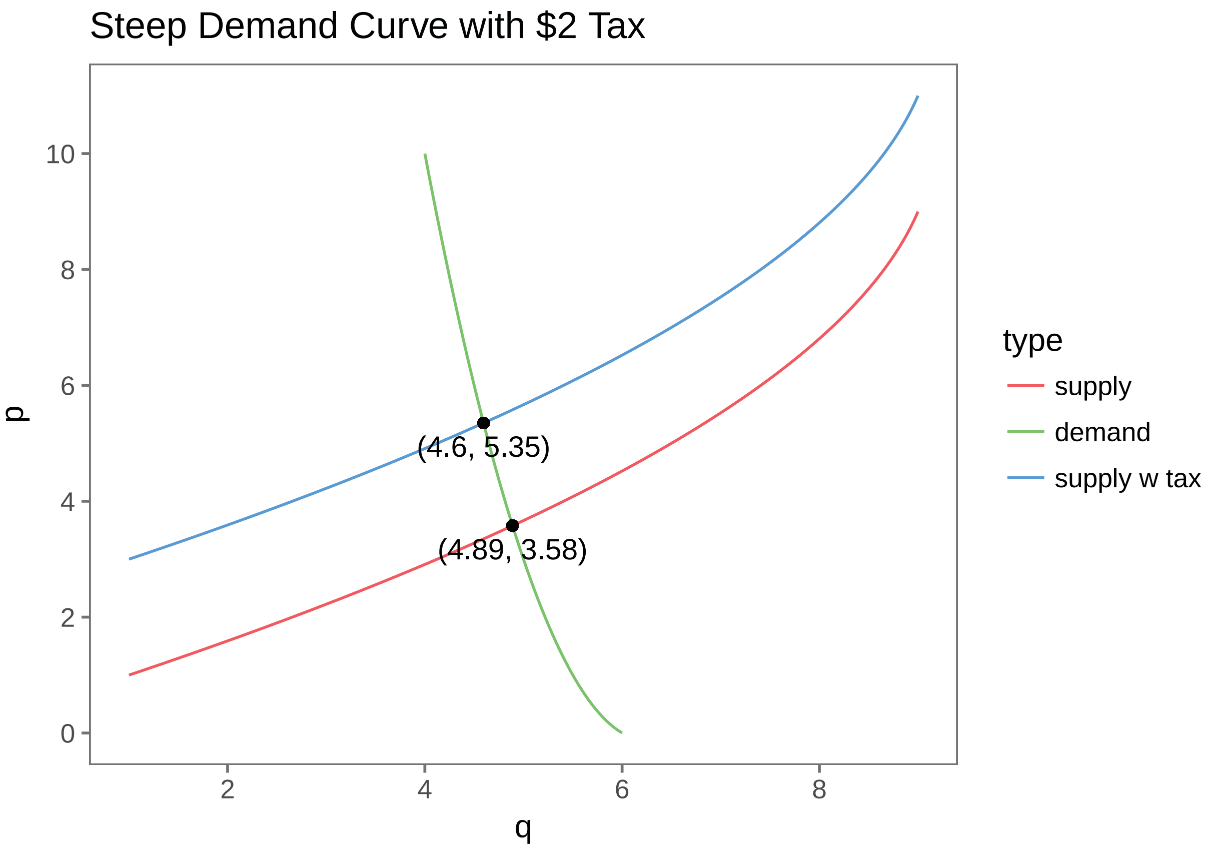
Page B-25, Question 1

Show that when the demand curve is very steep, a tax added to each item sold will fall primarily on consumers.

Steep Demand Curve

```
# average supply slope, steep deamnd
supply <- bezier(c(1, 8, 9), c(1, 5, 9))
demand <- bezier(c(4, 5, 6), c(10,1,0.002))

# add $2 tax and update supply curve
tax <- 2
supply_tax <- list(x=supply$x,y=supply$y + tax)

# save to df
df <- data.frame(q=supply$x, p=supply$y, type="supply") %>%
    rbind(data.frame(q=demand$x, p=demand$y,type="demand")) %>%
    rbind(data.frame(q=supply_tax$x, p=supply_tax$y,type="supply w tax"))

# calculate intersection points
X1 <- curve_intersect(supply, demand)
X2 <- curve_intersect(supply_tax,demand)
```

```
X1_lab <- paste0("(",round(X1$x,2),", ",round(X1$y,2),")")
X2_lab <- paste0("(",round(X2$x,2),", ",round(X2$y,2),")")

# plot
ggplot(df, aes(x=q,y=p, colour=type)) + geom_line() +
    annotate("text",x=X1$x, y=X1$y-0.4, label = X1_lab ) +
    annotate("text",x=X2$x, y=X2$y-0.4, label = X2_lab) +  theme_few() + scale_color_few() +
    geom_point(x=X1$x, y=X1$y, col="black") + geom_point(x=X2$x, y=X2$y, col="black") +
    labs(title="Steep Demand Curve with $2 Tax") +
    scale_x_continuous(breaks =seq(0,10,2)) + scale_y_continuous(breaks =seq(0,10,2))
```



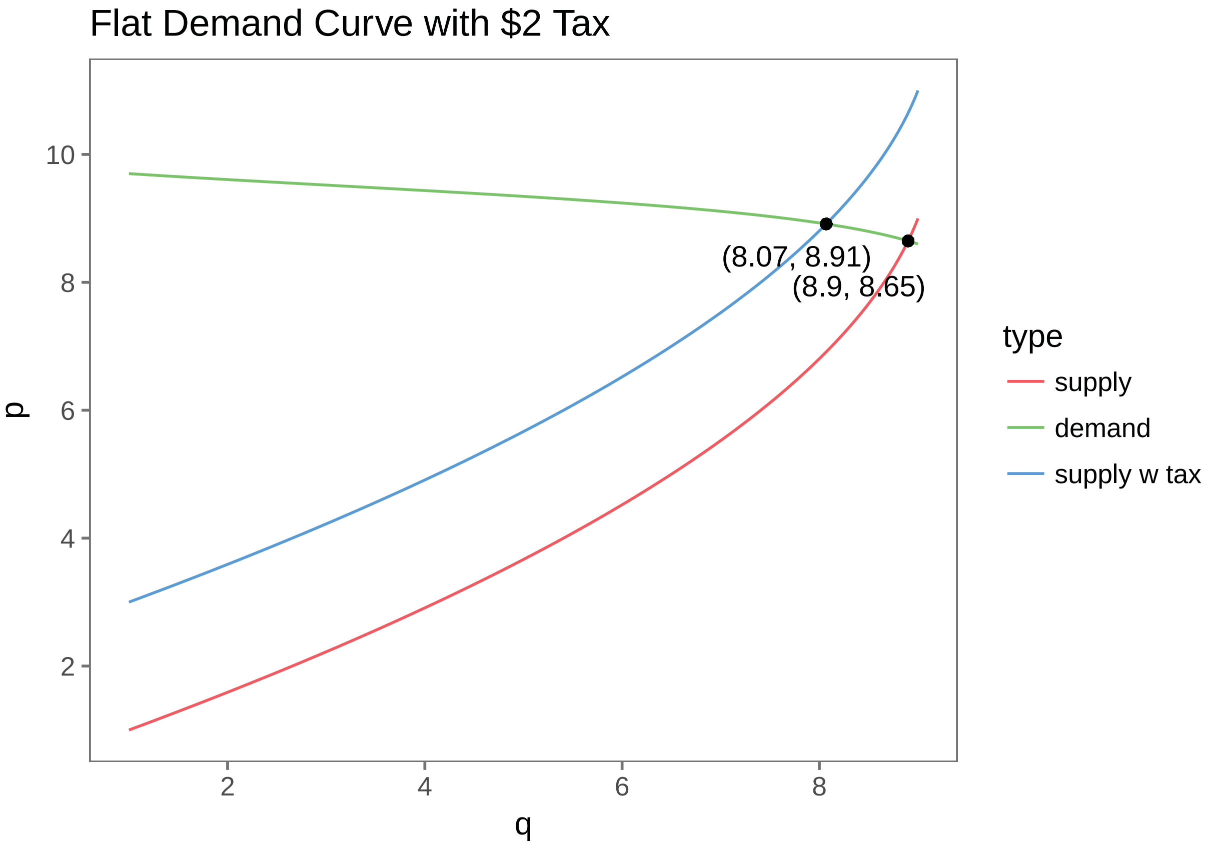
In the example above, the consumer shoulders $1.77 of the $2 tax burden, or roughly 88.6%.

Now show that when the demand curve is more nearly horizontal,the tax is paid mostly by the industry.

Flat Demand Curve

```
# average supply slope, flatter demnd
supply <- bezier(c(1, 8, 9), c(1, 5, 9))
demand <- bezier(c(1,3, 6, 8, 9), c(9.7,9.5,9.3,9.1,8.6))


# add $2 tax and update supply curve
tax <- 2
supply_tax <- list(x=supply$x,y=supply$y + tax)

# save to df
```

```r
df <- data.frame(q=supply$x, p=supply$y, type="supply") %>%
    rbind(data.frame(q=demand$x, p=demand$y,type="demand")) %>%
    rbind(data.frame(q=supply_tax$x, p=supply_tax$y,type="supply w tax"))

# calculate intersection points
X1 <- curve_intersect(supply, demand)
X2 <- curve_intersect(supply_tax,demand)
X1_lab <- paste0("(",round(X1$x,2),", ",round(X1$y,2),")")
X2_lab <- paste0("(",round(X2$x,2),", ",round(X2$y,2),")")

# plot
ggplot(df, aes(x=q,y=p, colour=type)) + geom_line() +
    annotate("text",x=X1$x-0.5, y=X1$y-0.7, label = X1_lab) +
    annotate("text",x=X2$x-0.3, y=X2$y-0.5, label = X2_lab) + theme_few() + scale_color_few() +
    geom_point(x=X1$x, y=X1$y, col="black") +  geom_point(x=X2$x, y=X2$y, col="black") +
    labs(title="Flat Demand Curve with $2 Tax") +
    scale_x_continuous(breaks =seq(0,10,2)) + scale_y_continuous(breaks =seq(0,10,2))
```

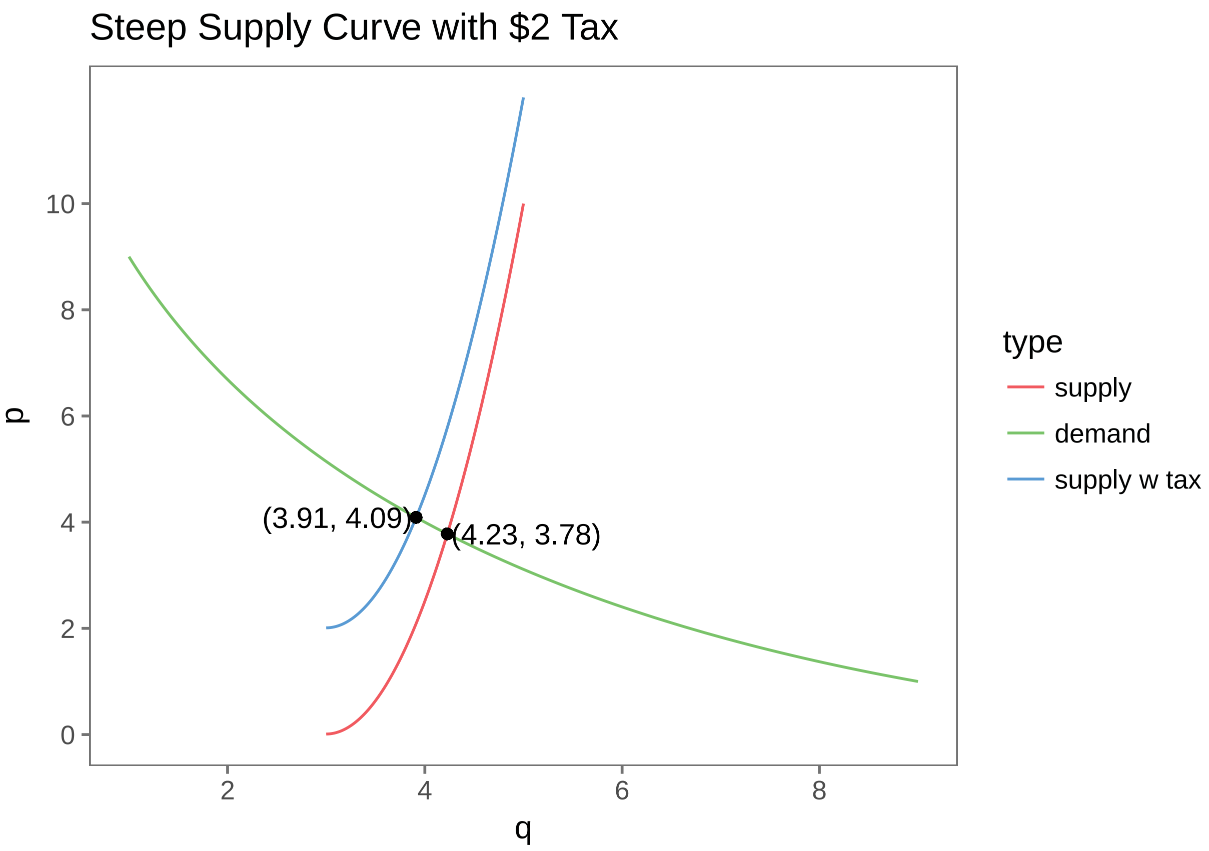


With a flatter demand curve, the consumer shares a smaller percentage of the tax burden. In the example above, the consumer pays $0.27 of the $2 tax burden, or 13.3%. In other words, the industry bears primary responsibility for the tax.

Steep Supply

What if the supply curve is very steep?

```r
# steep supply slope, average demnd
supply <- bezier(c(3,4,5), c(.01,.02,10))
demand <- bezier(c(1,3,9), c(9, 3, 1))

# add $2 tax and update supply curve
tax <- 2
supply_tax <- list(x=supply$x,y=supply$y + tax)

# save to df
df <- data.frame(q=supply$x, p=supply$y, type="supply") %>%
    rbind(data.frame(q=demand$x, p=demand$y,type="demand")) %>%
    rbind(data.frame(q=supply_tax$x, p=supply_tax$y,type="supply w tax"))

# calculate intersection points
X1 <- curve_intersect(supply, demand)
X2 <- curve_intersect(supply_tax,demand)
X1_lab <- paste0("(",round(X1$x,2),", ",round(X1$y,2),")")
X2_lab <- paste0("(",round(X2$x,2),", ",round(X2$y,2),")")

# plot
ggplot(df, aes(x=q,y=p, colour=type)) + geom_line() +
    annotate("text",x=X1$x+0.8, y=X1$y, label = X1_lab) +
    annotate("text",x=X2$x-0.8, y=X2$y, label = X2_lab) + theme_few() + scale_color_few() +
    geom_point(x=X1$x, y=X1$y, col="black") +  geom_point(x=X2$x, y=X2$y, col="black") +
    labs(title="Steep Supply Curve with $2 Tax") +
    scale_x_continuous(breaks =seq(0,10,2)) + scale_y_continuous(breaks =seq(0,10,2))
```

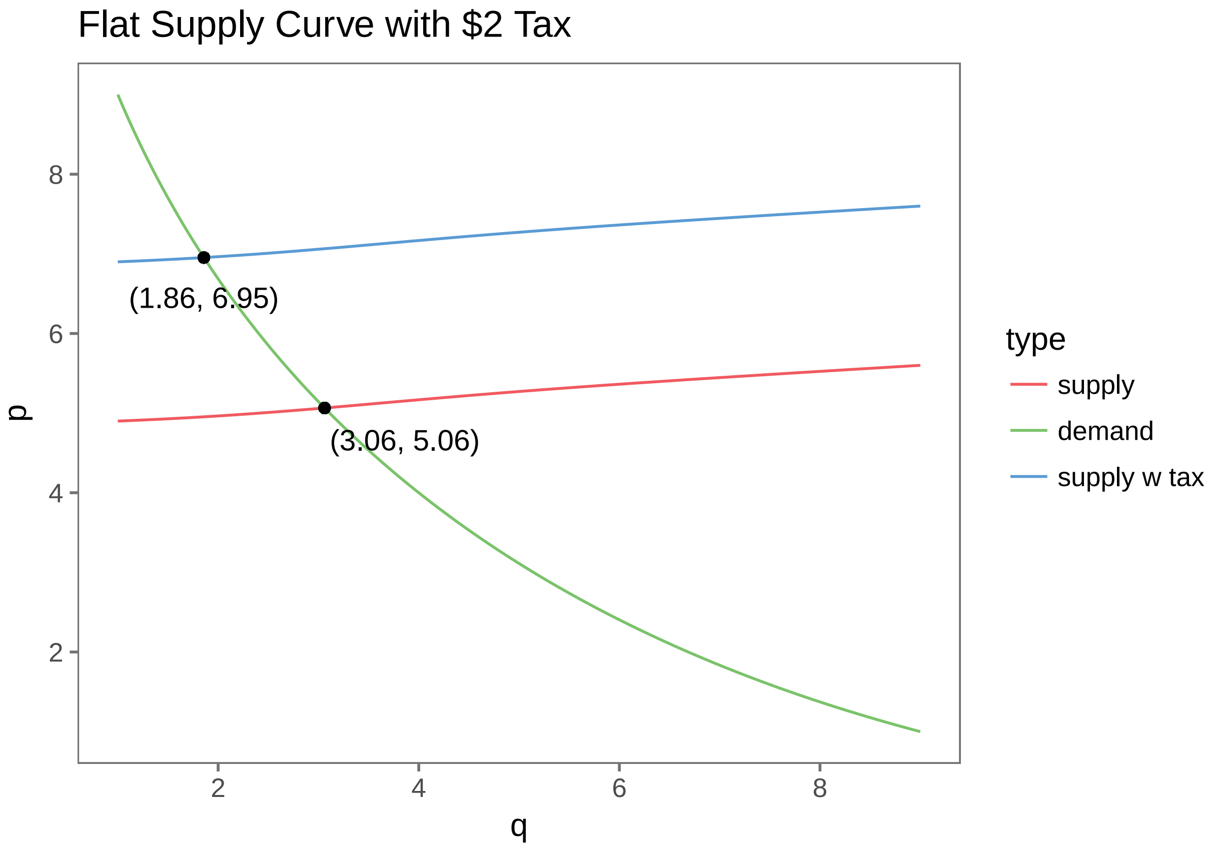


With a steep supply curve, the consumer shares a smaller percentage of the tax burden. In the example above, the consumer pays $0.31 of the $2 tax burden, or 15.5%. Here the industry bears primary responsibility for the tax.

Flat Supply Curve

What if the supply curve is nearly horizontal?

```r
# flatter supply slope, average demnd
supply <- bezier(c(1,3,4,5,9), c(4.9,5,5.2,5.3,5.6))
demand <- bezier(c(1,3,9), c(9, 3, 1))

# add $2 tax and update supply curve
tax <- 2
supply_tax <- list(x=supply$x,y=supply$y + tax)

# save to df
df <- data.frame(q=supply$x, p=supply$y, type="supply") %>%
    rbind(data.frame(q=demand$x, p=demand$y,type="demand")) %>%
    rbind(data.frame(q=supply_tax$x, p=supply_tax$y,type="supply w tax"))

# calculate intersection points
X1 <- curve_intersect(supply, demand)
X2 <- curve_intersect(supply_tax,demand)
X1_lab <- paste0("(",round(X1$x,2),", ",round(X1$y,2),")")
X2_lab <- paste0("(",round(X2$x,2),", ",round(X2$y,2),")")

# plot
ggplot(df, aes(x=q,y=p, colour=type)) + geom_line() +
    annotate("text",x=X1$x+0.8, y=X1$y-0.4, label = X1_lab) +
    annotate("text",x=X2$x, y=X2$y-0.5, label = X2_lab) + theme_few() + scale_color_few() +
    geom_point(x=X1$x, y=X1$y, col="black") +  geom_point(x=X2$x, y=X2$y, col="black") +
    labs(title="Flat Supply Curve with $2 Tax") +
    scale_x_continuous(breaks =seq(0,10,2)) + scale_y_continuous(breaks =seq(0,10,2))
```



With a flatter supply curve, the consumer shares a larger proportion of the tax burden. In the example above, the consumer pays $1.89 of the $2 tax burden, or 94.5%.

References

- Marginal Revenue: https://en.wikipedia.org/wiki/Marginal_revenue
- Cowplots: https://cran.r-project.org/web/packages/cowplot/vignettes/plot_grid.html
- Supply and demand curves in R: https://github.com/andrewheiss/reconPlots