Plotting Notes

James Holland Jones *Stanford University*

Introduction

R has powerful graphics capabilities. While we typically use these for plotting data, we can also make publication-quality plots for elucidating theoretical topics as well.

These notes are a very tentative start to a much larger body of work. I hope they are nonetheless helpful in their rather incomplete form.

The Taylor-Series Approximation is Your Friend

You may be familiar with Taylor polynomials and how useful they are for applied mathematics and science.

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2!} + \frac{f'''(a)(x-a)^3}{3!} + \cdots$$

When we truncate the Taylor series at the first term, we produce a linear approximation of our function. In other words, it's a tangent line. It turns out that we often want to draw tangents in theoretical figures and just as the Taylor series can help us derive theory, so too can it help us generate plots!

Cobb-Douglas Production Function

The Cobb-Douglas production function is a model for production that is the product of two power functions. The classic model combines capital (K) and labor L. The inputs are raised to powers α and β.

$$W = K^{\alpha} L^{\beta}$$

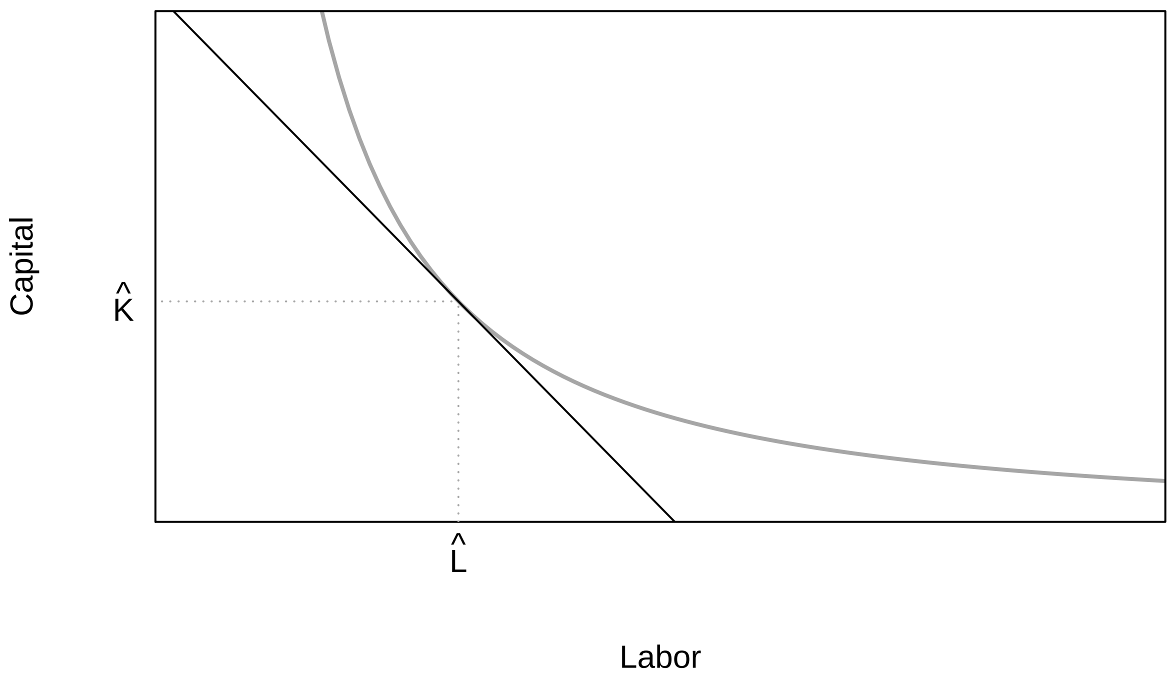
In the Cobb-Douglas form, the exponents also turn out to be the elasticities of production with respect to the inputs. So, suppose that $\alpha = 0.25$, this means that a 1% increase in the size of the goat herd will increase overall wealth by 0.25%.

If $\alpha + \beta = 1$, then there are constant returns to scale: doubling inputs will double the output. If, on the other hand, $\alpha + \beta > 1$, there are increasing returns to scale so that doubling inputs will more than double the output.

```
x <- seq(0,0.8,length=100)
L <- seq(0,1,length=100)
K <- (1/L^0.7)^(1/0.3)
#derivative
fprime <- -5.6/(0.3^2.4)
# make a function to calculate K from L
kf <- function(L) (2/L^0.7)^(1/0.5)
# pick an arbitrary point for tangency
```

```
a <- 0.3


plot(L,kf(L), type="l", lwd=2, axes=FALSE, frame=TRUE,
     yaxs="i", xaxs="i",
     xlim=c(0,1), ylim=c(0,50),
     col=grey(0.65),
     xlab="Labor", ylab="Capital")
# tangent
lines(x, kf(a)+fprime*(x-a), lwd=1, col="black")
# values on the axes
segments(0.3,0,0.3,kf(0.3), col=grey(0.65), lty=3)
segments(-0.1,kf(0.3),0.3,kf(0.3), col=grey(0.65), lty=3)
mtext(expression(hat(L)),1,at=0.3, padj=0.5)
mtext(expression(hat(K)),2, at=kf(0.3), adj=2, padj=0.5, las=2)
```



Marginal Value Theorem

Something like the Marginal Value Theorem (MVT), a phrase coined by Charnov (1976), appears in many applications: foraging theory (Charnov, 1976), sexual selection (Parker and Stuart, 1976), the evolution of virulence (van Baalen and Sabelis, 1995), and of course, life history theory (Smith and Fretwell, 1974). If you pay careful attention, you will notice that it's always the maximization of some sort of ratio, where the numerator and denominator trade-off. In this case, the MVT solution arises naturally from the quotient rule for differtiation. The MVT states that the optimal value of the ratio fitness measure can be found when a straight line rooted at the origin is tangent to the fitness/constraint function. This the eponymous "marginal value."

In previous examples, we specified the tangent point. Here we solve for the optimal value, which for the marginal value theorem, says that the rate is maximized where a line rooted at the origin is tangent to the utility function. We can mess around with different slopes and try to find something that's approximately right or we can find the actual solution using the root-finding function in `R`. I need to acknowledge MikePrice here because he helped me get unstuck as I flailed to get `uniroot()` to work. I should also note that when I publish notes like this, even if I accompany them with caveats about incompleteness, you are seeing the product of lots of trial and error (lots of error, I assure you). Remember, the struggle is part of the scientific process.

Anyway, code for the marginal value theorem (in its many guises).

```
x <- seq(0,20,length=500)
# utility function fp> 0 fpp < 0
# turns out that RMarkdown does not handle comments with single quotes
# fp == deriv of f; fpp == 2nd deriv of f
f <- function(x) {
  1 - exp(-0.2*(x-1))
}
# derivative of the utility function
fprime <- function(x) {
  0.2*exp(-0.2*x)*exp(0.2)
}

# f + fp*(z-x) = 0
# z = x -(f/fp)
# solve for tangency; find the root of this
xinter <- function(x) {
  return(x - f(x)/fprime(x))
}

soln <- uniroot(xinter,c(0,10))

plot(x,f(x), type="l", lwd=2, xaxs="i", yaxs="i",
     xlab="Time",
     ylab="Rate of Gain",
     ylim=c(0,1))
lines(x,(f(soln$root)/soln$root)*x,col="red")
segments(soln$root,0,soln$root,f(soln$root), lty=2, col="red")
segments(0,f(soln$root),soln$root,f(soln$root), lty=2, col="red")
mtext(expression(hat(t)),1,at=soln$root, padj=0.5)
## some non-optimal adaptive functions
lines(x,(f(2)/2)*x,col=grey(0.85))
lines(x,(f(1.5)/1.5)*x,col=grey(0.85))
lines(x,(f(11)/11)*x,col=grey(0.85))
```

Optimal Age at First Reproduction

Simple Example of the optimal trade-off between adult reproductive value (E) and juvenile recruitment (S). Charnov (1997) has suggested that fitness is a product, broadly construed, of three things: juvenile recruitment, annual fertility of adults, and adult life expectancy. In turn, these

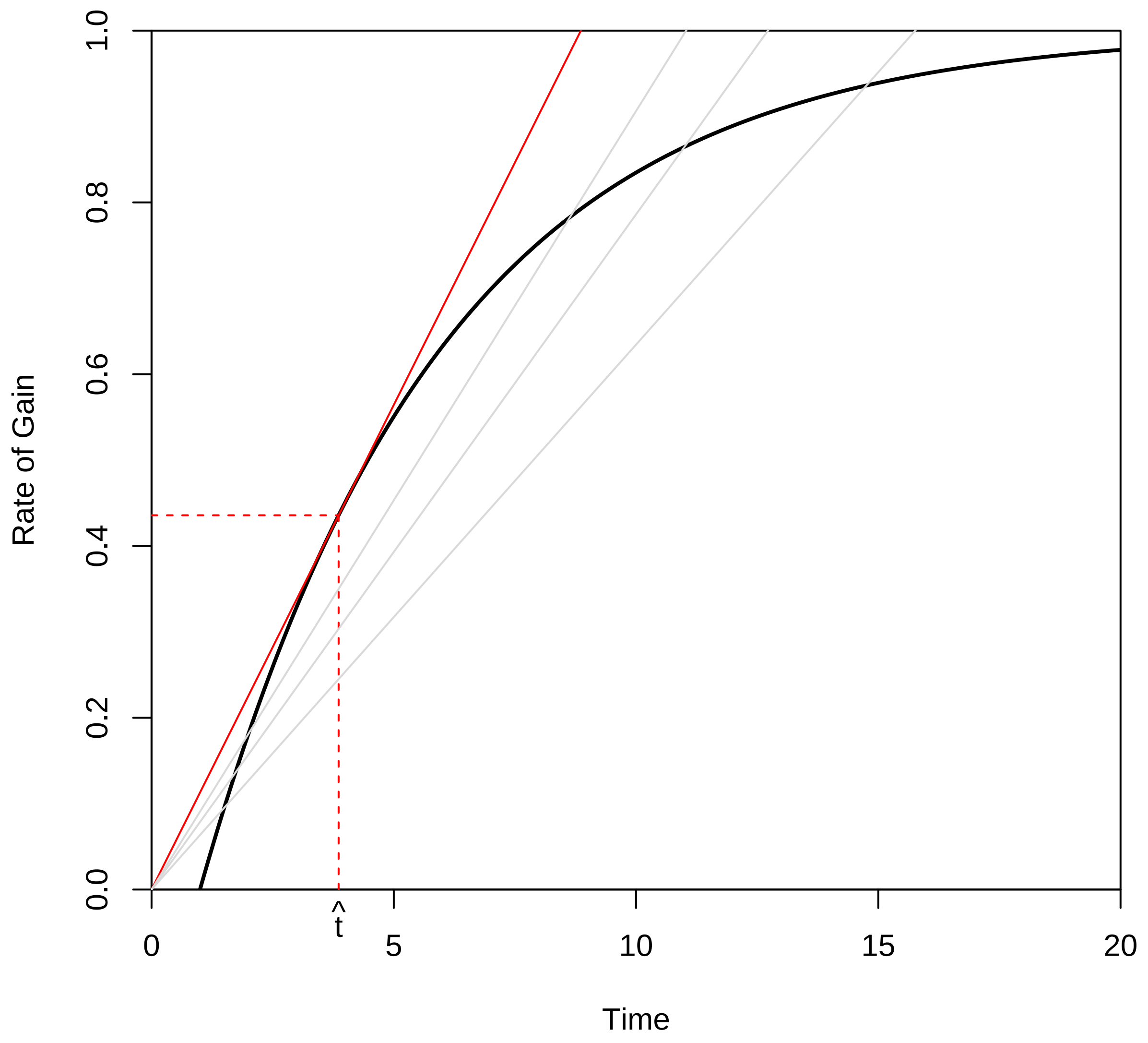


Figure 1: Marginal value theorem plot. A line rooted at the origin that is tangent to the gain curve provides the optimal patch-residence time, $\hat{t}$.

elements can be combined. For example, the product of annual fertility and adult life expectancy can be thought of as *adult reproductive value* because it is the expected total reproduction over the an individual's lifespan, conditional on them being recruited into the breeding population. As Charnov notes, these things are likely to trade-off. Moreover, the multiplicative form of fitness makes these trade-offs particularly straightforward to visualize and analyze.

When we plot the allowable combinations of $\log(S)$ and $\log(E)$, we get a convex plot of the iso-fitness plot linking the logs of E and S, which indicates diminishing marginal returns in both dimensions. The optimal life history is the one for which a line with a slope of -1 is tangent to this iso-fitness constraint curve. Why? The fitness measure (assuming population stationarity) is $R_0 = S\,E$. Take logs such that $\log(R_0) = \log E + \log S$ and differentiate with respect to age at first reproduction (α; which, in Charnov's formalism, is the control parameter for the life history):

$$\frac{d \log R_0}{d\alpha} = \frac{d \log E}{d\alpha} + \frac{d \log S}{d\alpha}.$$

Setting this equal to zero and rearranging, we find the optimality criterion:

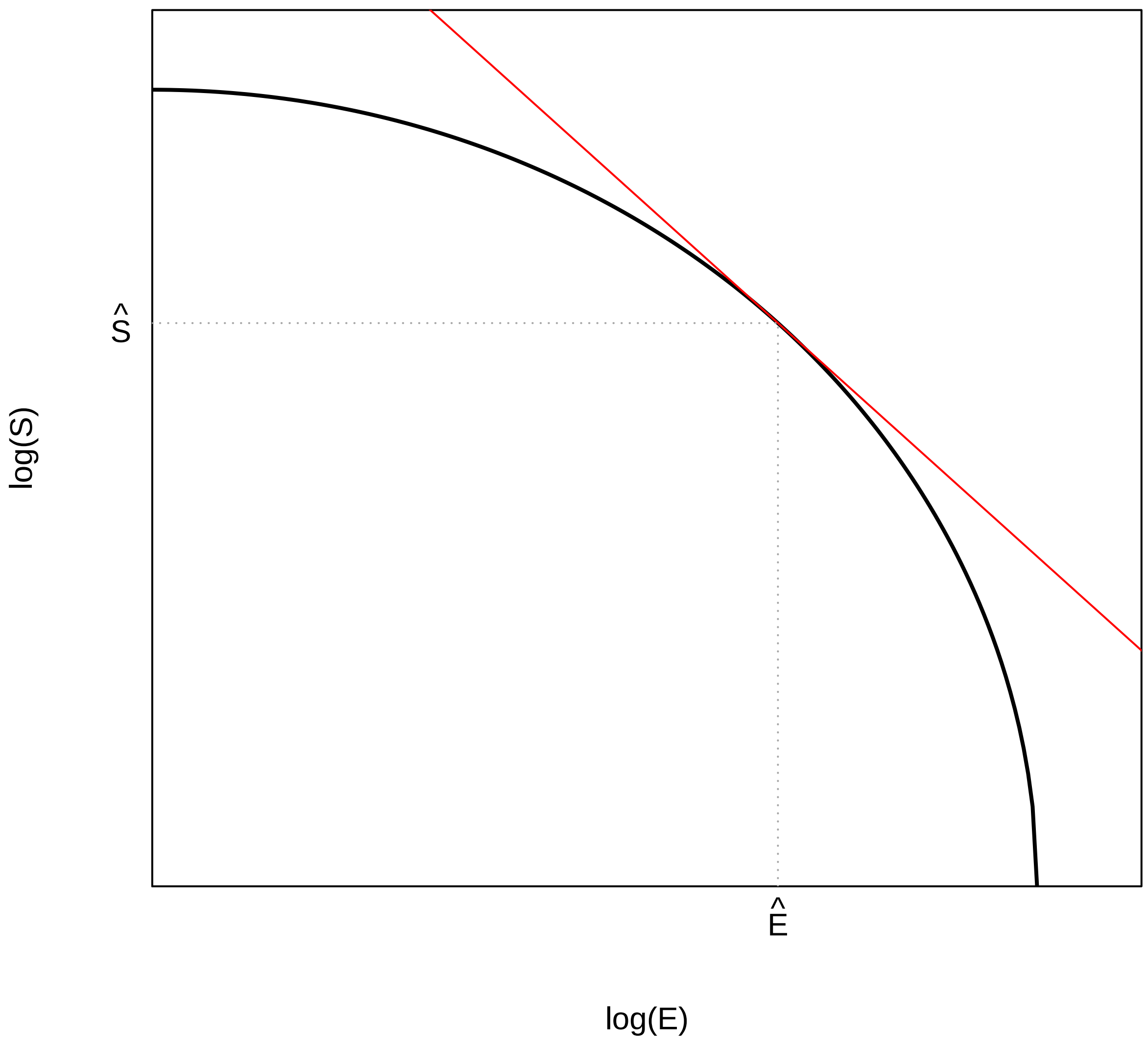
$$\frac{d \log E}{d \log S} = -1.$$

```r
g <- seq(0,sqrt(1/5),length=200)
h <- sqrt(1-(5*g^2))
hf <- function(g) sqrt(1-(5*g^2))
## derivative
fp <- function(g) -5*g/sqrt(1-5*g^2)

# solve for tangency; find the root of this
# note the sign change
ginter <- function(g) {
  return(g + hf(g)/fp(g))
}

## do not search over whole interval
## because g values > sqrt(5) will give NaNs!
a <- uniroot(ginter,c(0,0.4))$root
## this simply extends the plotting range
## so that the tangent line fills the plotting range
gg <- seq(0,0.5+a,length=500)

plot(g,hf(g), type="l", lwd=2, axes=FALSE, frame=TRUE,
     yaxs="i", xaxs="i",
     ylim=c(0,1.1), xlim=c(0,0.5),
     xlab="log(E)",
     ylab="log(S)")
## first-order Taylor Series approx
lines(gg, hf(a)+fp(a)*(gg-a), col="red")
segments(a,0,a,hf(a), col=grey(0.65), lty=3)
segments(-0.1,hf(a),a,hf(a), col=grey(0.65), lty=3)
mtext(expression(hat(E)),1,at=a, padj=0.5)
mtext(expression(hat(S)),2, at=hf(a), adj=2, padj=0.5, las=2)
```



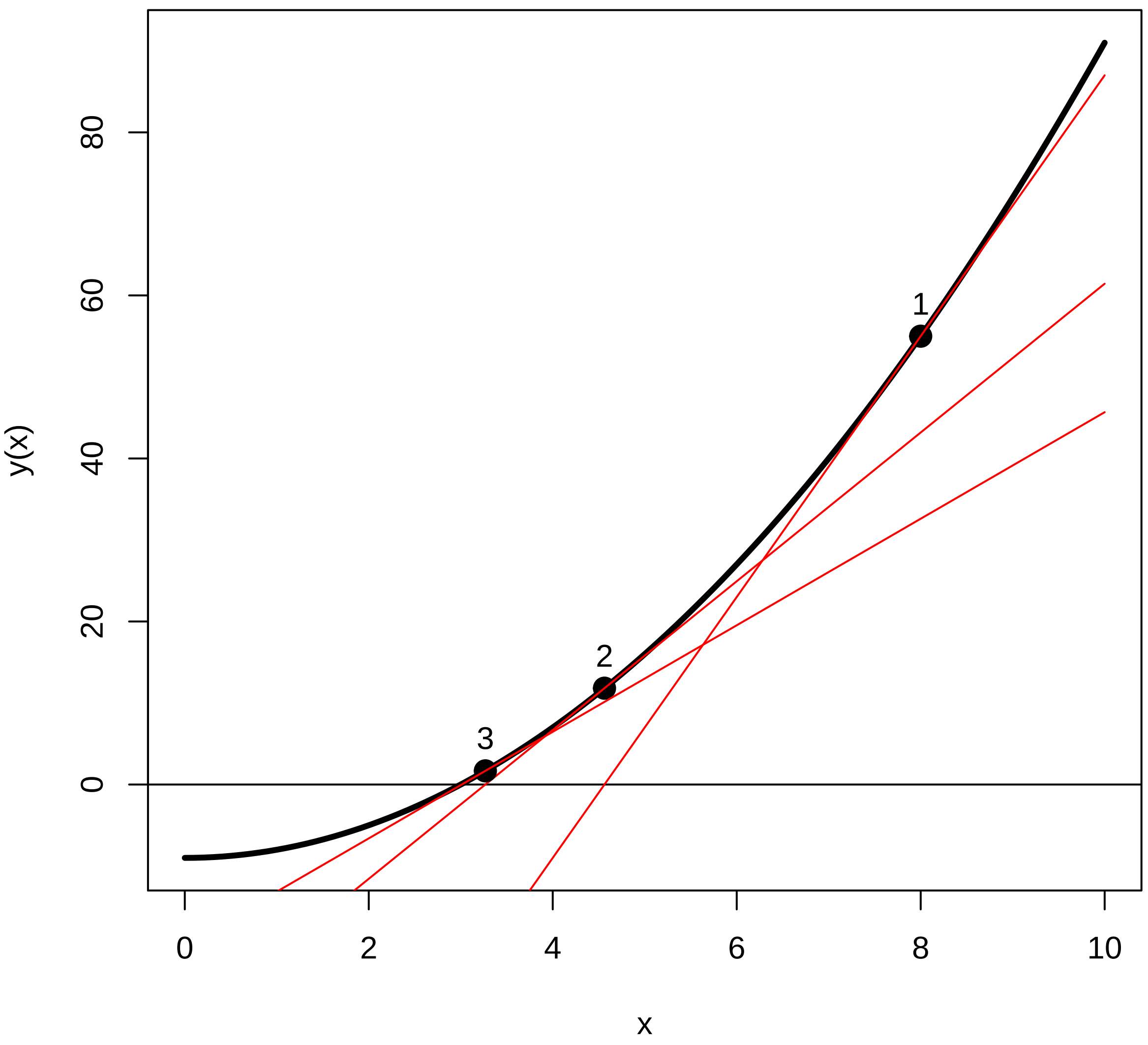
Graphical Newton-Raphson Method

Another cool application of tangent lines in plots illustrates the popular and powerful family of optimization algorithms are broadly known as "Newton's Method' or the"Newton-Raphson Algorithm." This algorithm finds roots of functions – that is, points where the function is zero. The basic idea is that we start from an initial guess point on our function. We then draw a line tangent to our function at this point. Finding the x-intercept for the tangent line, we repeat the process only this time drawing our tangent line from the point on the curve corresponding to the x-intercept of our last tangent line. It turns out that, for some initial guess, x_0, the value $x_1 = x_0 - f(x_0)/f'(x_0)$ is a better estimate of the root. We can then repeat this process until we are satisfactorily close to the root of the function. We can make a quick graphical demonstration Newton's method for a very simple function $y = x^2 - 9$. We start with a guess at $x = 8$. The plot shows three iterations (numbered sequentially). We can see that each iteration gets much closer to the root of this equation (at $x = 3$). In fact, it gets so close after three steps that plotting another iteration is indistinguishable from the correct solution (though for a realistic tolerance, it would take a couple more steps to get right to $x = 3$).

```r
## graphical newton-method

x <- seq(0,10, length=100)
y <- function(x) x^2 - 9
x1 <- function(a) a-y(a)/(2*a)

# first iteration
a <- 8
plot(x,y(x),type="l",lwd=3)
abline(h=0)
points(8,55, pch=19, cex=1.5)
text(8,59,"1")
lines(x, y(a) + (x-a)*2*a, col="red")
# second iteration
points(x1(a),y(x1(a)), pch=19, cex=1.5)
text(x1(a),y(x1(a))+4,"2")
a <- x1(a)
lines(x, y(a) + (x-a)*2*a, col="red")
# third iteration
points(x1(a),y(x1(a)), pch=19,cex=1.5)
text(x1(a),y(x1(a))+4,"3")
a <- x1(a)
lines(x, y(a) + (x-a)*2*a, col="red")
```



If we were really feeling ambitious, we could animate this!

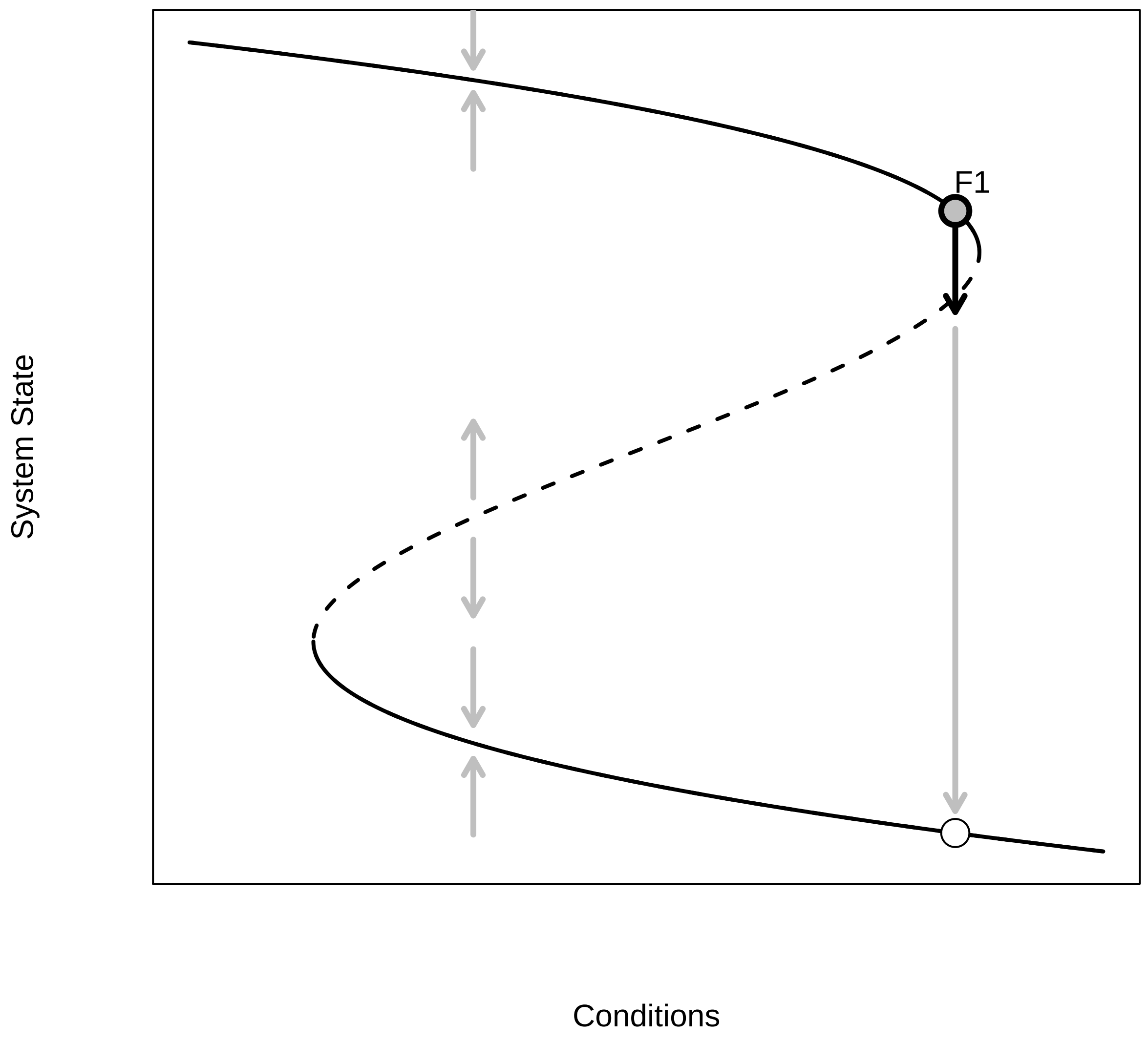
Fold-Catastrophe Model

A *catastrophe* is a sudden shift in system state (Zeeman, 1976). An interesting form of catastrophe, which Scheffer (2009) discusses in detail, is the *fold catastrophe*.

This is a pretty complicated figure. The solid parts of the curve are stable – when the system state is perturbed when in the vicinity of this part of the attractor, it tends to return, as indicated by the grey arrows pointing back to the attractor. The dashed part of the attractor is unstable – perturbations in this neighborhood tend to move away from the attractor. This graphical representation of the system makes it pretty easy to see how a small perturbation could dramatically change the system if the current combination of conditions and system state place the system on the attractor near the neighborhood where the attractor changes from stable to unstable. The figure illustrates one such scenario. The conditions/system state start at point $F1$. A small forcing perturbs the system off this point across the bifurcation. Further forcing now moves the system way off the current state to some new, far away, stable state. We go from a very high value of the system state to a very low value with only a very small change in conditions. Indeed, in this figure, the conditions remain constant from point $F1$ to the new value indicated by the white point – just a brief perturbation was sufficient to cause the drastic change.

```r
x <- seq(-12,12,length=10000)
y <- seq(12,10/sqrt(3), length=1000)
## fold-catastrophe is a cubic
plot(-x^3+100*x,x,type="l", axes=FALSE, lwd=2, lty=2,
     xlab="Conditions", ylab="System State")
box()
lines(-y^3+100*y,y, lwd=2)
lines(y^3-100*y,-y, lwd=2)

# unstable
arrows(-200,-5,-200,-2.75, code=1, lwd=3, length=0.1, col=grey(0.75))
arrows(-200,-1.5,-200, 0.75, code=2, lwd=3, length=0.1, col=grey(0.75))
#lower stable
arrows(-200,-6,-200, -8.25, code=2, lwd=3, length=0.1, col=grey(0.75))
arrows(-200,-11.5,-200, -9.25, code=2, lwd=3, length=0.1, col=grey(0.75))
#upper stable
arrows(-200,13.5,-200, 11.25, code=2, lwd=3, length=0.1, col=grey(0.75))
arrows(-200,8.25,-200, 10.5, code=2, lwd=3, length=0.1, col=grey(0.75))
# use locator() to find coordinates
points(357,7, pch=21, cex=2, lwd=3, bg=grey(0.75))
text(376.6749, 7.87279, "F1")
points(357,-11.452987, pch=21, cex=2, bg="white")
arrows(357,6.5,357,4, lwd=3, length=0.1)
arrows(357,3.5,357,-10.8, code=2, lwd=3, length=0.1, col=grey(0.75))
```



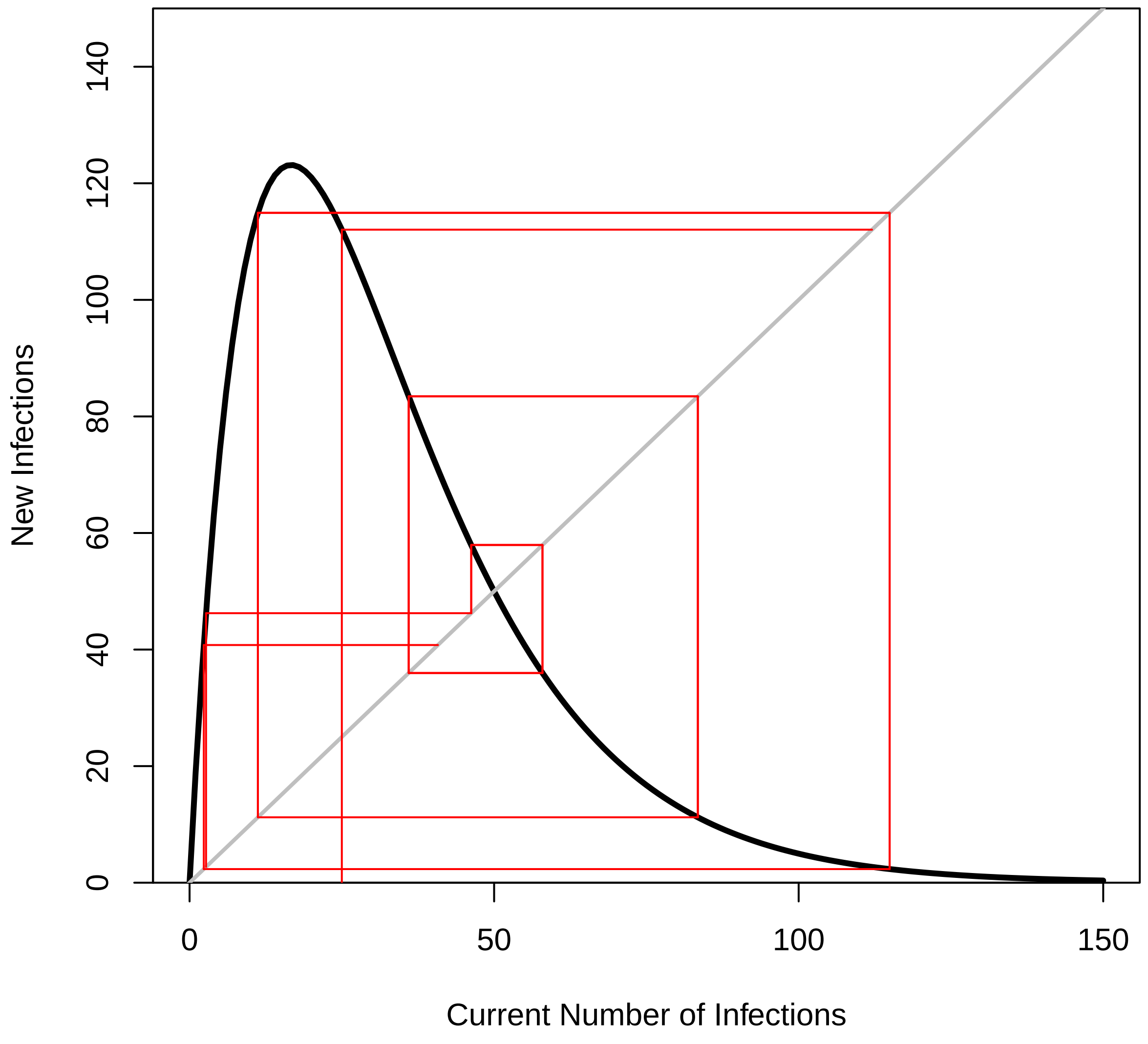
Cobwebbing

With a discrete-time model in one dimension (e.g., an unstructured population model), we can trace the dynamics as we iterate the model forward using a technique called *cobwebbing*. Here's a quick example of a Ricker recruitment model, a density-dependent population model with the feature that it overcompensates when numbers exceed the carrying capacity. When the population of highly reactive (i.e., has strong growth potential), this tendency for overcompensation can lead to some pretty wild dynamics. This plot shows such a case.

```
## Ricker recruitment function
ricker.recruit <- function(r0,K,N) N*exp(r0*(1-(N/K)))
## fast growth!
r0 <- 3
K <- 50
N <- 0:150
n0 <- 25
## iterate model for 10 time steps
t <- 10
```

```
y <- rep(0,t)
y[1] <- ricker.recruit(r0=r0,K=K,N=n0)
for(i in 2:t)  y[i] <- ricker.recruit(r0=r0,K=K,N=y[i-1])

plot(N,ricker.recruit(r0=r0,K=K,N=N), type="l", col="black", lwd=3, yaxs="i",
     ylim=c(0,150),
     xlab="Current Number of Infections", ylab="New Infections")
abline(a=0,b=1, lwd=2, col=grey(0.75))
segments(n0,0,n0,y[1], col="red")
segments(n0,y[1],y[1],y[1], col="red")
for(i in 2:(t-2)){
    segments(y[i],y[i],y[i],y[i+1], col="red") #vertical
    segments(y[i],y[i+1],y[i+1],y[i+1], col="red") #horiz
    segments(y[i+1],y[i+1],y[i+1],y[i+2], col="red") #vert
    segments(y[i+1],y[i+2],y[i+2],y[i+2], col="red") #horiz
}
```



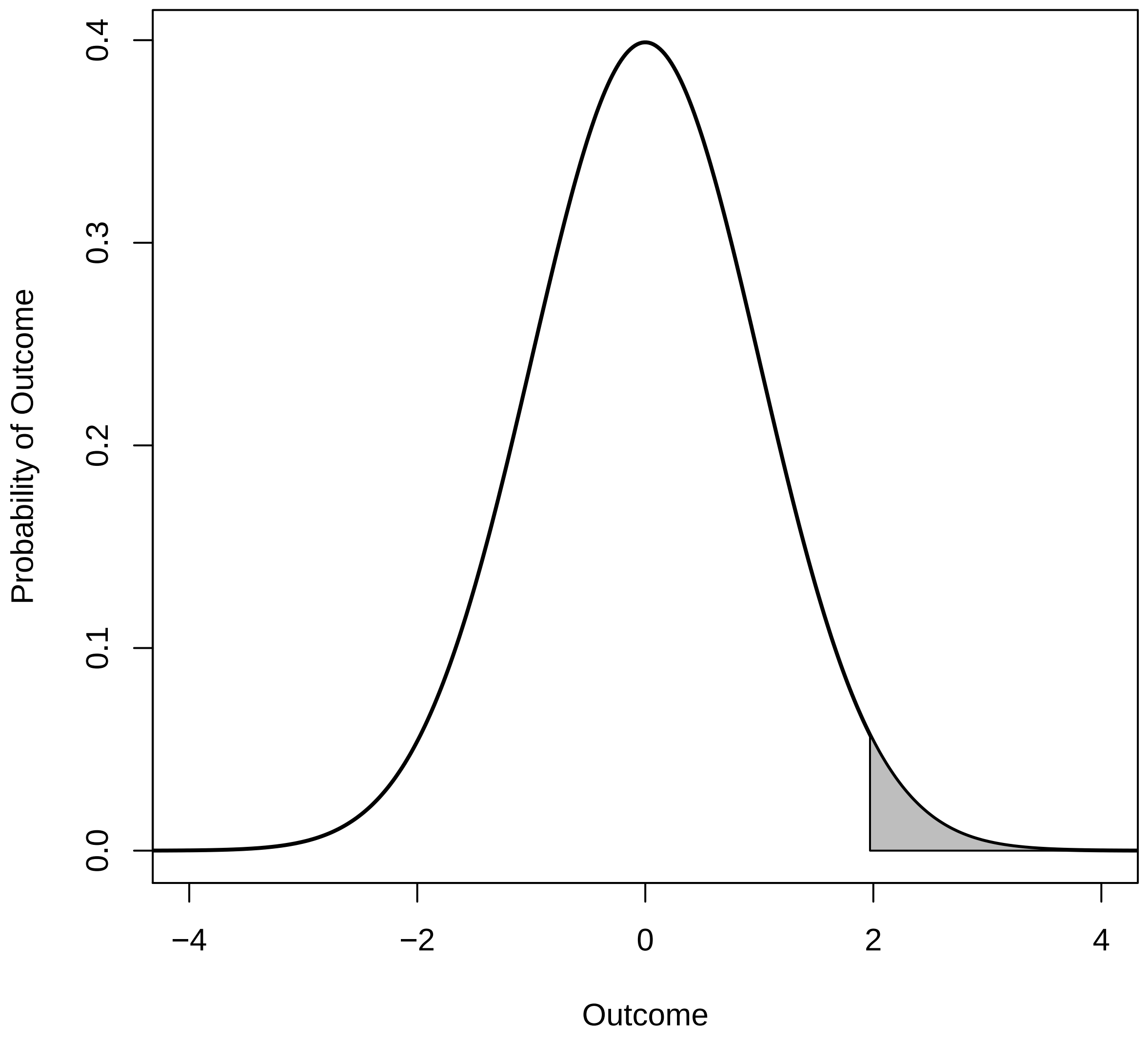
```r
## this could very easily be made into a function (and probably should be)
```

This population model with highly over-compensatory dynamics will never settle down. It always overshoots or undershoots and so fluctuates wildly. May (1976) notes that we can measure the strength of the response by the slope of the recruitment function at its equilibrium value (i.e., where the grey line of equality intersects with the recruitment function). Using the tools we've discussed in these note, we could calculate that slope and draw a tangent line at that point!

Polygons

Sometimes we want to color in the area under a probability density to show how much relative probability is contained in an interval or a tail. Here we compare the tail probability of a standard normal distribution with a low-df *t* distribution. We will color in the upper tail above the value of 1.96, the approximate 97.5th quantile of the standard normal distribution and the conventional definition of "statistical significance."

```r
## normal
z <- seq(-20, 20, length=2000)
p <- dnorm(z)
plot(z,p, type="l", lwd=2, xlim=c(-4,4),
     xlab="Outcome", ylab="Probability of Outcome")
z0 <- z[z >= 1.96]     # define region to fill
z0 <- c(z0[1], z0)
p0 <- p[z >= 1.96]
p0 <- c(0, p0)
polygon(z0, p0, col="grey")
```

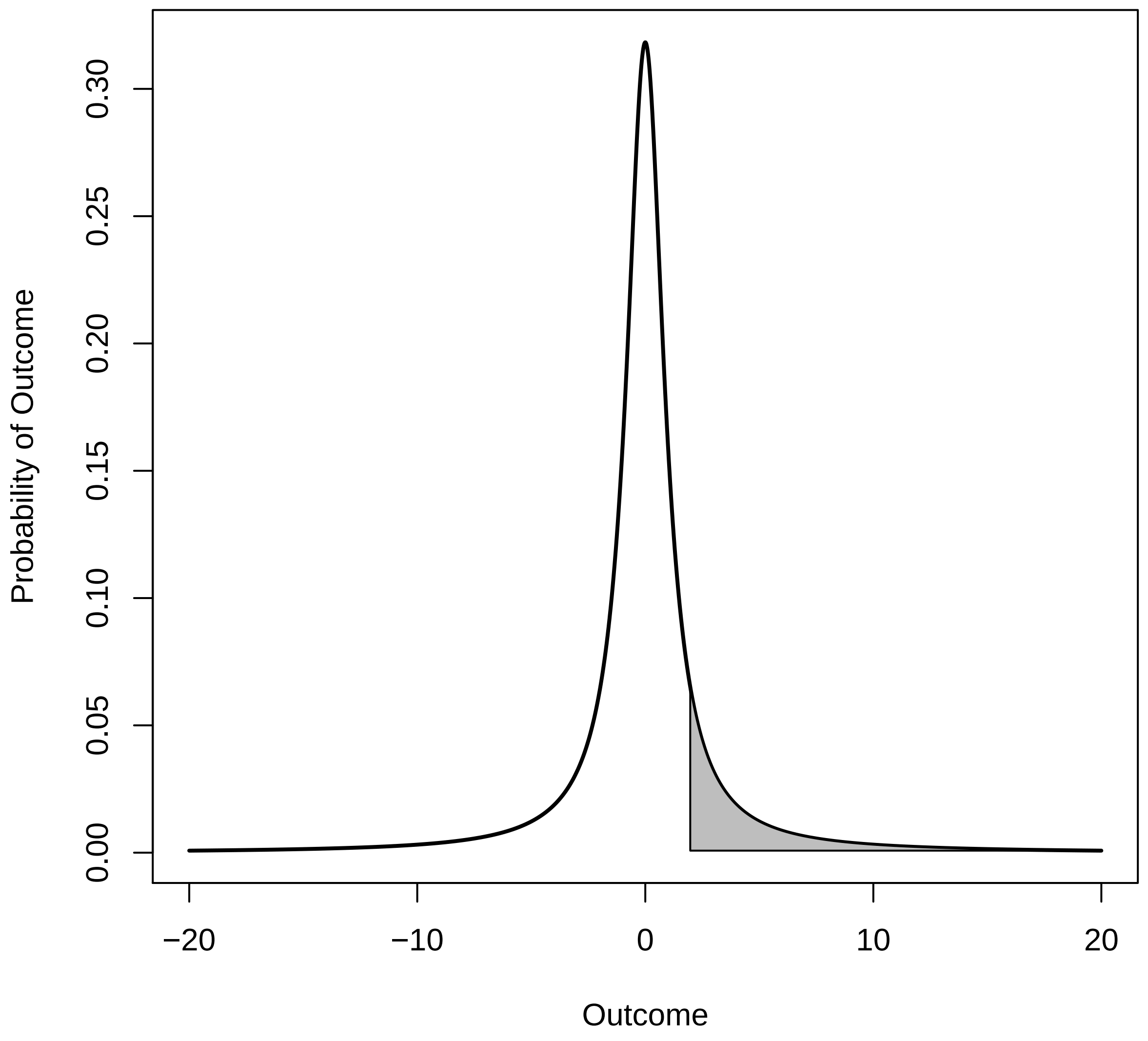


```r
# integrate to see how much probability mass is in the tail
integrate(dnorm, 1.96, Inf)
```

```
## 0.0249979 with absolute error < 1.9e-05
```

Now for the *t* distribution

```r
## t, df=1
q <- dt(z,df=1)
plot(z,q, type="l", lwd=2, xlab="Outcome", ylab="Probability of Outcome")
t0 <- z[z >= 1.96]     # define region to fill
t0 <- c(t0[1], t0)
q0 <- q[z >= 1.96]
q0 <- c(dt(20,df=1), q0)
polygon(t0, q0, col="grey")
```



```r
## integrate to see how much probability mass is in the tail
integrate(dt, 1.96, Inf, df=1)
```

```
## 0.1501714 with absolute error < 1.1e-10
```

Ooh, pointy. Note that the *t* distribution decays so slowly that you can see that the polygon has a slope to it even when you extend the range out to 20. This explains why we added `dt(20,df=1)` as the first element of the `q` variable for the polygon.

More

The use of `locator()` to find points interactively.

Graphs Using `iGraph`

There's a lot.

References

Charnov, E. L. (1976). Optimal foraging, the marginal value theorem. *Theoretical Population Biology 9*(2), 129–136.

Charnov, E. L. (1997). Trade-off invariant rules for evolutionarily stable life histories. *Nature 387*, 393–394.

May, R. M. (1976). Simple Mathematical-Models with Very Complicated Dynamics. *Nature 261*(5560), 459–467.

Parker, G. A. and R. A. Stuart (1976). Animal Behavior as a Strategy Optimizer: Evolution of Resource Assessment Strategies and Optimal Emigration Thresholds. *American Naturalist 110*(976), 1055–1076.

Scheffer, M. (2009). *Critical Transitions in Nature and Society*. Princeton: Princeton University Press.

Smith, C. C. and S. D. Fretwell (1974). The optimal balance between size and number of offspring. *American Naturalist 108*, 499–506.

van Baalen, M. and M. W. Sabelis (1995). The dynamics of multiple infection and the evolution of virulence. *American Naturalist 146*(6), 881–910.

Zeeman, E. C. (1976). Catastrophe Theory. *Scientific American 234*(4), 65–83.