Assignment 3, due February 28

Corrections: (none yet)

1. **True/False**. In each case, state whether the statement is true or false and explain your answer in a few words or sentences
   (a) A necessary assumption of the Cox Ross Rubinstein binary option pricing theory is that it is possible to buy or sell the underlying asset (the *underlier*) for S_t at time t.
   (b) Let q_u be the risk neutral probability that $S_0 \to uS_0$ in time Δt. Suppose $q_u = .4$. This implies that there is a 40% probability that $S_{\Delta t} = uS_0$.
2. Consider a binomial CRR (Cox Ross Rubinstein) tree where the numbers u_k and d_k are different at each period $t_k = k\Delta t$. This means that $S_{(k+1)\Delta t} = u_k S_{k\Delta t}$ or $S_{(k+1)\Delta t} = d_k S_{k\Delta t}$. How does the CRR binomial tree option pricing theory change (select all that apply)?
   (a) The binomial tree is no longer recombining.
   (b) Hedging is impossible and risk neutral probabilities no longer exist.
   (c) The risk neutral probabilities of up and down motions of the underlier (the stock) are different at different periods.
   (d) Dynamic hedging is no longer possible.
3. Suppose an *Irish style* put option comes with an automatic early exercise feature. (Ireland is part of Europe but is closer to America than any other part. This option is essentially European (no decision by the holder), but might be exercised early.) There is a strike price K and an early exercise price $L < K$. If $S_t \leq L$ at any time $0 \leq t \leq T$ the option is exercised at price L, resulting in a cash flow $K - L$. Otherwise, the option is exercised at time T as a normal European put.
   (a) The present value of the cash flow of an Irish style put is always more than a European put with the same expiration date and strike price.
   (b) The present value of the Irish style option might be more or less than the present value of the cash flow of the European style option.
4. Consider a call option to buy an asset for price S_0 (the present spot price) in six months ($\alpha = .5$ for half a year, in the book's terminology). Suppose the asset price itself (in our model) can go up 20% or down 10% in that time. Use the binary one period CRR model to find the theoretical price

of the call today. Assume the interest rate is $r = 2\%/\text{year}$. How does the answer depend on r (is it linear, quadratic, exponential, whatever)? How much does the option value change if you take $r = 0$?

5. Consider a European put and call with the same strike, K, and expiration time, T. Find a way to replicate S_T with $P(S_T)$ (the payout of the put at the time it expires), $C_(S_T)$, and cash (hint: the coefficient of P is negative). Use this to find a relation between P_0 (the price of the put option today), C_0, S_0, r, and T. This relation is called *put-call parity*.

6. (*Hedging exotic options with vanillas*). A *vanilla* option (also called *plain* vanilla) is a simple put or call (American or European style). Vanilla options are traded in public markets (the CBOE and elsewhere). In theory, they are liquid, meaning that you can buy or sell at something like the posted price. (Many publicly traded options are not very liquid in practice, as few or no trades happen in a given day. Index options (options on major stock indices) that expire soon and are near "the money" (the price of the underlier) are pretty active.) An *exotic* option is an option with any other payout. This exercise considers a hedge of an exotic option using the underlier, cash, and a vanilla option.

   Suppose the underlier has price S_0 today and the price "tomorrow" is one of the the three numbers $S_t = uS_0$, or $S_t = mS_0$, or $S_t = dS_0$. The multipliers are $u > 1$ (the "up" move), $d < 1$, the "down" move, and m, the "middle" move, which may be a little up ($m > 1$) or a little down ($m < 1$). Suppose the price of the vanilla today is F_0 and the price is one of the values $F_t = F_u$ (if $S_t = uS_0$), F_m (if $S_t = mS_0$), or F_d (if $S_t = dS_0$). The goal of this exercise is a hedging argument that leads to a formula of the form
$$V_0 = e^{-rt}\left(q_u V_u + q_m V_m + q_d V_d\right) \ . \tag{1}$$
   The *risk neutral probabilities* q_u, q_m, and q_d depend on r, u, m, d, and F_u, F_m, F_d, and F_0. The risk neutral "probabilities" should not be negative and $q_u + q_m + q_d = 1$. The situation is summarized in the table

time	0	t	t	t
state		u	m	d
cash	1	e^{rt}	e^{rt}	e^{rt}
underlier	S_0	uS_0	mS_0	dS_0
vanilla	F_0	F_u	F_m	F_d
exotic	$V_0 =???$	V_u	V_m	V_d

   The three instruments, cash, underlier, vanilla, have values known at time t and time 0 (today). The exotic have value known at t but not today.

   (a) Assume that a formula of the form (1) exists. Derive three linear equations for the three unknowns q_u, q_m, and q_d by taking the exotic to be cash, the underlier, and the vanilla option. The known prices 1 (for cash), S_0 and F_0 should be reproduced.

(b) Under what conditions do these equations have a unique solution? What do you need to know about the numbers u, m, d, F_u, F_m, and F_d. Explain how these conditions make financial sense. For example, it must be that the vanilla option payout cannot be replicated with cash and the underlier. If it could be, then the vanilla would not add anything new and would not help replicate the exotic.

(c) Show that if the conditions of part (b) are satisfied, then it is possible to replicate the payout of any exotic using a portfolio of cash, the underlier, and the vanilla. The portfolio value today is $\Pi_0 = \alpha + \beta S_0 + \gamma F_0$. The value tomorrow is $\Pi_t = \alpha e^{rt} + \beta S_t + \gamma F_t$. The portfolio weights are α, β, and γ. (The equations that determine α, β, and γ are almost the same (or exactly the same) as the equation in part (a). But the point of view is different. The portfolio weights in the CRR two state binary model were called C and Δ.)

(d) Show that if $q_u < 0$, and the conditions of part (b) are satisfied, then there is an arbitrage. Hint: Show that there is a portfolio (of cash, underlier, and vanilla) that pays 1 if $S_t = uS_0$ and zero otherwise. This portfolio would have negative cost (because $q_u < 0$) but possibly positive payout, so it would be an arbitrage. Explain.

Computing

This assignment uses Newton's method to calculate the *implied* interest rate determined by a price for an annuity. Suppose we have a function $f(x)$ and that we have code to evaluate $f(x)$ and $f'(x)$ for any x. We have a number a and we want to find x_* so that $f(x_*) = a$. This is sometimes called "root finding", because x_* is a "root" of the equation $f(x) - a = 0$. The term "root" originally applied to polynomials, but now is used for the "zeros" (x values where $f(x) = a = 0$) of any function. Newton's method constructs a sequence of *iterates* x_n for $n = 1, 2, \ldots$, that we hope have the property that

$$x_n \to x_* \text{ as } n \to \infty .$$

You stop the code when you think x_n is close enough to x_*.

Newton's method is *iterative*, it involves a sequence of *iterations*. That means, it does the same thing (with different numbers) over and over. It is derived by imagining that x_n is close to x_*. We approximate $f(x_*)$ by Taylor expansion up to the first derivative term around x_n. That is

$$f(x_*) = f(x_n + (x_* - x_n)) \approx f(x_n) + f'(x_n)(x_* - x_n) .$$

We approximate the equation $f(x_*) = a$ with the approximation

$$f(x_n) + f'(x_n)(x_* - x_n) \approx a .$$

We choose the next iterate x_{n+1} by changing this approximation into an equation

$$f(x_n) + f'(x_n)(x_{n+1} - x_n) = a .$$

At iteration n, we know x_n. We write code to evaluate $f(x_n)$ and $f'(x_n)$. Then we can solve for x_{n+1}. The algebra leads to

$$x_{n+1} = x_n - \frac{f(x_n) - a}{f'(x_n)} \ . \tag{2}$$

The core of Newton's method is to apply the formula (2) over and over until you think x_n is close enough to x_*, or until you decide it is not working.

The first "iterate", which we call x_1 (it sometimes is called x_0) is the *initial guess*. The programmer has to find some guess at the solution. This isn't fun. It often involves drawing graphs and playing with simple approximations. This is "problem specific", which means it depends on the specific formula for $f(x)$ and the shape of the graph and the value of a. The iteration formula (2) is not problem specific. But if you have a poor initial guess, the method is more likely to fail.

It is a theorem that if $f'(x_*) \neq 0$ and if x_1 is close enough to x_*, then $x_n \to x_*$ as $n \to \infty$. Moreover, x_n converges fast. x_{n+1} is much closer to x_* than x_n is. You can learn about this by searching on "Newton's method, local quadratic convergence". A good course on Numerical Analysis will cover this. But if the initial guess is poor, then x_n may not converge to x_*. On the contrary, it may happen that $x_n \to \infty$ as $n \to \infty$. This is common. It is important to spend the time to find a good initial guess.

But despite your best efforts, your Newton's method code will sometimes fail to converge. You need something in your code to make sure the iteration (2) is not an infinite loop. You can do this by defining a parameter $n_{\max}$ and stopping the program if $n > n_{\max}$. If this happens, the program should print an *error message* saying it didn't work. (There are better ways to report failure, but this is only the second program.) In practice $n_{\max} = 20$ is probably fine, but $n_{\max} = 100$ also is OK.

You also need a *convergence criterion*, a condition that tells you when to stop and declare victory. You can define a small number ϵ and declare convergence if $|x_{n+1} - x_n| \leq \epsilon$. If you want an interest rate to within one basis point, and if r has units of interest per year, then you can take $\epsilon = 10^{-4}$. But this isn't so small. If the code can get to 10^{-4} it probably get to 10^{-10}. I suggest you try a small ϵ until you learn it's too small.

Your `R` script should have a function `f()` that evaluates $f(x)$ (given the other parameters of the problem), and another function `fp()` that evaluates $f'(x)$. The code should choose an initial guess, then have a `while` loop that does the Newton's method iteration (2). This `while` loop should have a *trip count* to make sure it isn't an infinite loop.

Apply your code to calculating the interest rate r implied by the price of an annuity. If r is an interest rate, C is the annual payment rate, m is the number of payments per year, and T is the number of years, then the present value is

$$\text{PV} = \frac{C}{m} \sum_{n=1}^{mT} e^{-rt_n} \ \ , \ t_n = \frac{n}{m} \ . \tag{3}$$

We are given P, the price of the annuity. We seek the number r so that P is the present value given by (3). The problem has extra parameters C, m, and T. Your code should have assignment statements in the beginning

```
C    = ...
m    = ...
T    = ...
nMax = ...
eps  = ...
verbose = (TRUE or FALSE+
```

It should have definitions of f and f'. The formula for f' depends on differentiating the formula (3) with respect to r. Then comes code for the initial guess. Try something very simple yet sensible and move to something more complicated if that doesn't work. Then the `while` loop and the final statements that make a nicely formatted output of the answer. If `verbose` is `TRUE`, the while loop should print a line of output for each iteration. This line should have n, x_n, the difference $x_{n+1} - x_n$ that you use to decide whether you've converged. This will give you some confidence that your code is correct and solving the equation. If `verbose` is `FALSE`, then don't print the iteration by iteration progress, just the eventual answer.

Use your code to find the implied interest rate for a $\$20,000$/year annuity paid monthly for ten years if the price is $P = \$170,000$. See if you can "break" your code by finding parameter values where a simple initial guess does not work.

There is a handout on `while` loops and `if` tests that goes with this assignment.