Mathematics of Finance, Courant Institute, Fall 2015
http://www.math.nyu.edu/faculty/goodman/teaching/mathFin/index.html
Always check the class message board before doing any work on the assignment.

Assignment 5, due October 21

Corrections: (None yet. See message board)

1. The *cumulative normal* distribution is

$$N(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} \, dy \; . \tag{1}$$

This is the probability that $X < x$ if $X \sim \mathcal{N}(0,1)$ (which means that X is a Gaussian random variable with mean $\mu = 0$ and variance $\sigma^2 = 1$). If $f(x)$ is the PDF of X, then

$$\Pr(X < x) = \int_{-\infty}^{x} f(y) \, dy \; .$$

The formula (1) applies this fact with the Gaussian PDF $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$. We use y as the integration variable because it doesn't work to have x be both the integration variable and the upper limit. Clearly, $N(x) \to 0$ as $x \to -\infty$ (because $\Pr(X < x) \to 0$, or because $\int_{-\infty}^{\infty} f(x) \, dx$ converges absolutely). Equally "clearly", $N(x) \to 1$ as $x \to \infty$.

(a) Calculate $N'(x)$, $N''(x)$, and $N'''(x)$.

(b) Note the value of $N(0)$ and the signs of $N'(x)$ and $N''(x)$ when $x < 0$ and $x > 0$.

(c) Use this information, and the limits of $N(x)$ as $x \to \pm\infty$ to sketch a graph of $N(x)$. Make the graph reasonably large. Use a reasonable range of x values. Indicate how the function value ($N(0)$ and limits) and derivative information is used.

(d) Write the Taylor series approximation for $N(x)$ about the point $a = 0$. Use terms up to and including the third derivative term.

(e) Download and run the `R` script `Taylor.R`. The output should be:

```
> source("Taylor.R")
x =  0.1 , f is  -0.09983342 , f_1 is  -0.1 , f_3 is  -0.09983333
x =  0.2 , f is  -0.1986693 , f_1 is  -0.2 , f_3 is  -0.1986667
x =  0.5 , f is  -0.4794255 , f_1 is  -0.5 , f_3 is  -0.4791667
x =  -0.3 , f is  0.2955202 , f_1 is  0.3 , f_3 is  0.2955
x =  1 , f is  -0.841471 , f_1 is  -1 , f_3 is  -0.8333333
```

Comment on the accuracy of the first and third order Taylor approximations to $f(x) = \sin(\pi + x)$ for the values given. You may find the last one surprising, given that $x = 1$ is not "small".

(f) Modify the script to apply to the function $f(x) = N(x)$ (the cumulative normal, not the normal PDF). You evaluate $N(x)$ in `R` using `f = pnorm(x)` (`pnorm` is for "normal probability", which is the probability that $X < x$ in the normal (or Gaussian) distribution). Experiment with other x values by adding some values of your choosing to the `x_list` list. Comment on the accuracy of the first and third order Taylor approximations to $N(x)$ for small and not so small x values.

(g) (*Just verbiage, not an action item*) This exercise illustrates one of the ways applied mathematicians use computers. We work out formulas by hand (the Taylor approximations in this example) then use the computer to evaluate them.

2. Consider a market with n assets and no risk free asset. The *minimum variance* portfolio is the one that minimizes the portfolio variance with no constraint on the expected return, but still $\sum_j w_j = 1$. The Lagrange multiplier problem for the minimum variance portfolio has only one Lagrange multiplier. Show that the minimum variance portfolio is one of the two "funds" in the two fund theorem.

3. Many market models contain one or more *market factors*. This problem considers a single market factor, $Z \sim \mathcal{N}(\mu_Z, \sigma_Z^2)$ (normal with mean μ_Z and variance σ_Z^2). Suppose that the value X_j (the value of asset j) is the sum of an *idiosyncratic factor*, Y_j, and some "loading" of the market factor $\beta_j Z$. Then $X_j = Y_j + \beta_j Z$. The Y_j are independent, $Y_j \sim \mathcal{N}(\mu_{Y,j}, \sigma_j^2)$, and they are independent of the market factor. There are n random variables Y_j and one market factor Z.

(a) What is the form of the vector of expected returns μ_X with components $\mu_{X,j} = E[X]$?

(b) What is the form of the covariance matrix of returns C with entries $C_{X,ij} = \text{cov}(X_i, X_j)$. Hint: the formula is different if $i = j$ or $i \neq j$. The *Kronecker delta* symbol, δ_{ij}, is a convenient way to write a formula for C_{ij}. The Kronecker delta has values $\delta_{ij} = 1$ if $i = 1$, and $\delta_{ij} = 0$ if $i \neq j$. These are the entries of the identity matrix.

(c) (*Extra credit, do not start this until the rest of the assignment is finished and you have time left over.*) Look up the *Sherman Morrison formula* in Wikipedia (or some other reliable source) and use that to find an expression for C^{-1}.

(d) Suppose the portfolio is

$$P = \sum_{j=1}^{n} w_j X_j \ .$$

The overall market factor loading of the portfolio is

$$\beta_P = \frac{\partial P}{\partial Z} \ , \quad \text{with } Y_j \text{ fixed.}$$

Formulate an optimization problem for a portfolio with a fixed total weight $\sum w_j = 1$, a fixed expected total return $\sum w_j \mu_{X,j}$, and a fixed market factor loading. Show that the Lagrange multiplier method gives a problem with three Lagrange multipliers.

(e) (*Just verbiage, not an action item*) Some hedge funds claim to be *beta neutral*, which means that $\beta_P = 0$.

(f) Formulate a "three fund" theorem that applies to portfolios with a constraint on β_P.

4. *Postponed until next week.*

5. Download and run the code `NestedLoops.R`. Read over the explanation in `NestedLoops.pdf`. Do all the parts of this exercise using loops or nested loops. Do not use `R` matrix or vector operations such as `%*%` and `transpose()`. The matrices A and D, and the vector x are those generated in `NestedLoops.R`.

   When you create the code, you can save a lot of time using cut-and-paste. For example, there are several times you need to multiply matrices (a three-nested loop). Once you have code that does this, you can re-use that code elsewhere. This goes also for printing matrices. The sample code `NestedLoops.R` prints the three rows of a 3×3 matrix in several diffrent places. The code was created by writing it once, then using cot-and-paste to other places followed by a few small changes (changing `A` to `D`, for example). In future assignments we will learn a better way to use this – defining our own `R` functions.

   (a) Write a double loop like the first example in `NestedLoops.R` that prints all n rows of an $n \times n$ matrix, with one row on each line. Test your double loop on A.

   (b) Evaluate and print the matrix $B = A^2$. Check the result by hand.

   (c) Create the $n \times n$ matrix

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & 1 \\ 0 & 0 & \cdots & & 0 & 1 \end{pmatrix} .$$

   This matrix has all zeros except or ones on the diagonal and the *superdiagonal*. In formulas, the entries are $m_{ii} = 1$ and $m_{i,i+1} = 1$, and $m_{ij} = 0$ otherwise. Print the matrix for $n = 3$ and $n = 7$ (or some such values) to see that you got it right.

   (d) Use a 4-nested loop to compute and print M^p for some interesting numbers n and p. The outer loop multiplies M by M^r to get M^{r+1}. The inner loop(s) do the matrix multiplication.