Here's how long Warshall's algorithm takes to compute the transitive closure.

```
In[3]:=  Timing[Warshall[m]][[1]]

Out[3]=  1.31667 Second
```

Here's how long MatrixPower takes:

```
In[4]:=  Timing[tClosure[m]][[1]]

Out[4]=  0.0166667 Second
```

The MatrixPower routine is over 100 times faster! This illustrates an important point about Mathematica programming: always use the built-in functions when possible.

Summary

Calculating choice efficiency for consumption and production data is very easy using Mathematica. In addition the data can be illustrated graphically, compared to parametric estimates, and analyzed in a variety of other wasy.

References

Afriat, S. 1967. The construction of a utility function from expenditure data. International Economic Review, 7, 67–77.

Andreoni, J. and J. H. Miller. 1994. “Giving According to GARP:

An Experimental Study of Rationality and Altruism,” CMU Working Paper.

Battalio, R. C. et. al. 1973. A test of consumer demand theory using observations of individual consumer purchases. Western Economic Journal, 11, 411-428.

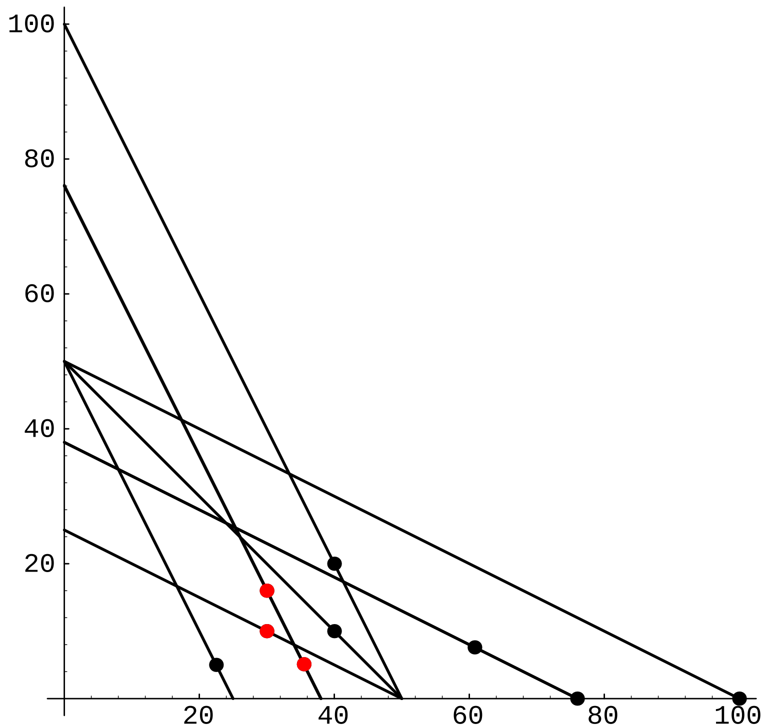
Brooks, John. 1993. Measuring cost efficiency in VA hospital pharmacies : an applied critique of the methods. Ph. D. dissertation, University of Michigan.

Cox, J. C. 1994."On testing the utility hypothesis", Technical report, University of Arizona.

Houtman, M. and Maks, J. A. 1985. Determining all maximial data subsets consistent with revealed preference. Kwantitatieve Methoden, 19, 89-104

Warshall, S. 1962. "A Theorem on Boolean Matrices," Journal of the American Association of Computing Machinery, 9, 11-12.

Varian, H. 1993. Microeconomic Analysis, 3rd edition, W. W. Norton & Co., New York.



Out[2]= -Graphics-

(The violations are illustrated in red on the screen but they come through as grey on the monochrome printed page.)

Appendix

In the text we computed the transitive closure of the direct revealed preference relation by computing the matrix power of the direct revealed preference matrix. We did this using the internal Mathematica function MatrixPower. Raising a $n \times n$ matrix to the power n would require n^4 operations if done in the most straightforward way. Another, seemingly more efficient way to compute the transitive closure is given by Warshall's algorithm (See Warshall (1962).)

```
In[1]:=  Warshall[mat_] := Module[{i,j,k,n,m},
         m=mat;
         n=Length[m];
         For[k=1,k<=n,k++,
             For[i=1,i<=n,i++,
                For[j=1,j<=n,j++,If[m[[i,k]]==1 && m[[k,j]]==1,
                m[[i,j]]=1]]]];m]
```

Note that Warshall's algorithm involves only n^3 operations. Despite this, Warshall's algorithm is much slower than using the internal MatrixPower function. For example, here is a random 10×10 matrix:

```
In[2]:=  m = Table[Random[Integer,{0,1}],{i,1,10},{j,1,10}];
```

It appears that subject 8 is about 97 percent efficient in her choices.

Drawing budget sets

Here are some functions to draw a set of budgets and illustrate the observations that violate revealed preference.

```
In[1]:=
        DrawABudget[{p1_,p2_},{x1_,x2_}] := Module[
                                {w,t},
                                w=p1*x1 + p2*x2;
                                Plot[w/p2 - p1*t/p2,{t,0,w/p1},
                                AspectRatio->1,DisplayFunction->Identity]]
        DrawBudgets[p_,x_] := Table[{DrawABudget[p[[i]],x[[i]]],
                                       Graphics[Point[x[[i]]]]},
                                       {i,1,Length[p]}]

        PlotViolations[p_,x_] :=
             ListPlot[x[[ViolationList[p,x,1]]],
             AspectRatio->1, DisplayFunction->Identity, PlotStyle->{Hue[0],PointSize[.02]}]

        DrawViolations[p_,x_] := Show[DrawBudgets[p,x],PlotViolations[p,x],
                                     Prolog->PointSize[.02],
                                     DisplayFunction->$DisplayFunction]
```

Example

To illustrate the use of DrawViolations, we examine some experimental data generated by Andreoni and Miller (1994). They presented subjects with a choice between taking some money for themselves or donating money to a group. If the subjects chose to donate, the experimenters would match the donations at various rates, thereby changing the "price" of the donations. Andreonio and Miller were interested in whether the subjects' contributions varied with the price of contributions in the way predicted by the theory of economic choice.

Here is the choice behavior exhibited by one of their subjects:

```
In[1]:= p={{1.0,0.5},{0.5,1.0},{2.0,1.0},{1.0,2.0},{1.0,1.0},{1.0,0.5},
           {0.5,1.0},{2.0,1.0},{1.0,2.0}};
        x={{30.,16.},{76.,0.},{22.5,5.0},{30.0,10.0},{40.0,10.0},
           {40.0,20.0},{100.0,0.0},{35.5,5.1},{60.8,7.6}};
```

```
In[2]:= DrawViolations[p,x]
```

```
In[5]:=  p={{1.,2.},{2.,1.}};
         x={{1.,2.},{2.,1.}};
```

```
In[6]:=  MatrixForm[GARPOK[p,x,tClosure[rdEMatrix[p,x,.81]],.81]]

Out[6]=  True    False

         False   True
```

```
In[7]:=  MatrixForm[GARPOK[p,x,tClosure[rdEMatrix[p,x,.79]],.79]]

Out[7]=  True   True

         True   True
```

This shows that the Afriat efficiency measure is between .79 and .81. We can automate the process of finding the maximal value of e such that the data satisfy GARP by using the following binary search.

```
In[8]:=  AfriatEfficiency[p_,x_,maxSteps_] :=
           Module[{step,e},
           e=1/2;
             For[step=1,step<=maxSteps,step++,
               If[NumberViolations[p,x,e] > 0,
                  e=e-2^(-(step+1)),
                  e=e+2^(-(step+1))];
               Print[N[e]]]]
```

```
In[9]:=  AfriatEfficiency[p,x,10]

0.75
0.875
0.8125
0.78125
0.796875
0.804687
0.800781
0.798828
0.799805
0.800293
```

As we saw before, these choices are about 80 percent efficient.

Here's the Battalio data for subject 8 again:

```
In[10]:= p={{1.,1.,1.},{0.5,2.,1.},{1.,1.,1.},{2.,0.5,1.},
         {2.,0.5,1.},{1.,1.,1.},{1.,1.,1.}};
         x={{30,9,9},{34,10,0},{10,4,0},{4,8,5},{2,15,0},
            {0,15,9},{4,7,12}};
```

```
In[11]:= AfriatEfficiency[p,x,10]

0.75
0.875
0.9375
0.96875
0.984375
0.976562
0.972656
0.970703
0.969727
0.970215
```

```
In[2]:=  consEfficiency[p,x]

Out[2]=  {1., 1., 1., 0.970588, 1., 0.708333, 0.73913}
```

Most of the consumers in the Battalio study were quite efficient. There were 256 choices made during the seven-week experiment. Of the inefficient choices, 8 were 97–99 percent efficient, 4 were 93–96 percent efficient, 1 was 91 percent efficient, and one was 81 percent efficient.

Afriat's efficiency index

The efficiency index described above measures the consumption efficiency of each observation. Afriat's efficiency index, described in Afriat (1967), measures the overall efficiency of a set of consumption choices. Following Afriat, we say that an observation r is directly revealed preferred to an observation s at efficiency level e if $ep^r x^r \geq p^r x^s$. (Although Afriat does not require it, we adopt the convention that it is always the case that $x^r R^D x^r$; i.e., an observation is always directly revealed preferred to itself.) Obviously $e = 1$ is the standard revealed preference comparison and $e = 0$ is vacuous. For a given e we can construct the analog of the direct revealed preference measure, compute its transitive closure, R_e and then check the analog of $GARP_e$: if $x^s R_e x^t$, then $ep^t x^t < p^t x^s$.

The following function computes the direct revealed preference relation associated with efficiency level e:

```
In[1]:=  rdEMatrix[p_,x_,e_] := Module[{i,j,n},
                              n=Length[p];
                              Table[If[(e*p[[i]].x[[i]] >= p[[i]].x[[j]])
                                 || i==j,1,0],
                               {i,1,n},{j,1,n}]]
```

We can use the **tClosure** function defined earlier to compute the transitive closure of this matrix. We then check to see if GARP is satisfied (at the given efficiency level) using the function **GARPOK**. This function returns a matrix with a **False** in the (i, j) entry if $x^i R x^j$ and $ep^j x^j > p^j x^i$; otherwise, it returns **True**. If **GARPOK** returns all **True** entries, the data are consistent with GARP at efficiency level e.

```
In[2]:=  GARPOK[p_,x_,m_,e_] := Module[{i,j,n},
                                 n=Length[m];
                                  Table[
                                    If[m[[i,j]]==1 && (e*p[[j]].x[[j]]
                                       > p[[j]].x[[i]]),False,True],
                                       {j,1,n},{i,1,n}]]
```

We use the **GARPOK** function to define a couple of other useful functions. This function returns the list of observations that violate GARP (at efficiency level e).

```
In[3]:=  ViolationList[p_,x_,e_] :=
                     Module[{m},
                     m = tClosure[rdEMatrix[p,x,e]];
                     Union[Flatten[Position[GARPOK[p,x,m,e],False]]]]
```

This function just counts the number of violations.

```
In[4]:=  NumberViolations[p_,x_,e_] := Length[ViolationList[p,x,e]]
```

We'll try these functions on the example we used in the last section.

```
In[3]:=  tClosure[m]

Out[3]=  {{1, 1, 1}, {1, 1, 1}, {1, 1, 1}}
```

Checking GARP

Once we compute the transitive closure, it is easy to check GARP. A convenient way to do this is to compute a "consumption efficiency index" analogous to the cost efficiency index we computed above. Here is a function that returns the observations revealed preferred to observation i.

```
In[1]:=  ObsRP[i_,m_] := Complement[
                     Table[If[m[[j,i]]>0,j,i],{j,1,Length[m]}],{0}]
```

The function **consEfficiency** computes the minimum expenditure necessary to purchase an observed choice that is revealed preferred to a given observation.

```
In[2]:=  consEfficiency[p_,x_] :=
             Module[{m,n},
             m=tClosure[rdMatrix[p,x]];
             n=Length[x];
             Table[Min[x[[ObsRP[i,m]]].p[[i]]/p[[i]].x[[i]]],
                  {i,1,n}]]
```

Here's an example of how this function is used.

```
In[3]:=  p={{1.,2.},{2.,1.}};
         x={{1.,2.},{2.,1.}};


In[4]:=  consEfficiency[p,x]

Out[4]=  {0.8, 0.8}
```

This shows that each of the two choices is only 80 percent efficient.

The token economy

Battalio et. al. (1973) collected a set of data on the consumption choices of 38 patients at Central Islip State Hospital. As part of their therapeutic treatment, the patients worked for tokens which could be retrieved for items such as as cigarettes, candy, milk, locker rental, clothes, admission to a dance, etc. During a seven-week period, the relative of various groups of these goods were doubled or halved. Since the prices of some of the goods were halved some weeks and doubled other weeks, prices varied by a factor of four. Data were collected on how the expenditures of each individual responded to the price changes. These data have been examined by Battalio (1973) and Cox (1994) using revealed preference techniques.

Here are the prices and choices for subject number 8 in these experiments.

```
In[1]:=  p={{1.,1.,1.},{0.5,2.,1.},{1.,1.,1.},{2.,0.5,1.},
         {2.,0.5,1.},{1.,1.,1.},{1.,1.,1.}};
         x={{30,9,9},{34,10,0},{10,4,0},{4,8,5},{2,15,0},
            {0,15,9},{4,7,12}};
```

Here is the consumption efficiency of this consumer's choices

Utility maximization

We now turn to consumer behavior. Consider a consumer that chooses a vector of consumption goods, x, facing a price vector p, and having income m so as to maximize utility:

$$\max_x u(x)$$
$$\text{such that } px \leq m.$$

Define the direct revealed preference relation by $x^t R^D x^s$ if and only if $p^t x^t \geq p^t x^s$. The terminology is rather intuitive: x^t is directly revealed preferred to x^s if x^t as purchased when x^s was affordable. The revealed preference relation, R, is defined to be the transitive closure of the relation R^D. That is $x^t R x$ if and only if there is some chain of observations $(x^s, x^r, \ldots, x^v)$ such that $x^t R^D x^s, x^s R^D x^r, \ldots, x^v R^D x$. Finally, some observed choices (p^t, x^t) are consistent with the utility maximization model if and only if they satisfy the Generalized Axiom of Revealed Preference (GARP):

GARP: If $x^t R x^s$, then it is not the case that $p^s x^s > p^s x^t$.

In words: if x^t was chosen when x^s was affordable, then x^s cannot be chosen when x^t is affordable

Computing the transitive closure of a preference relation

As before we assume that the data comes as two $n \times k$ matrix of prices and quantities.

The **rdMatrix** has a 1 in the (i, j) entry if observation i is directly revealed preferred to observation j.

```
In[1]:=  rdMatrix[p_,x_] := Module[{i,j,n},
                            n=Length[x];
                            Table[If[p[[i]].x[[i]] >= p[[i]].x[[j]],1,0],
                              {i,1,n},{j,1,n}]]
```

We need to compute the transitive closure of this matrix. A possible way to do this is to raise the matrix to the n^{th} power and then look at the sign of each entry. It is not hard to see that the (i, j) entry will be positive if and only if there is some chain of entries connecting i to j that are positive. This is very simple to implement in *Mathematica*.

```
In[2]:=  tClosure[m_] := Sign[MatrixPower[m,Length[m]]]
```

See the appendix for another way to compute the transitive closure.

Example

Here is an example with 3 observations where the direct revealed preference relation involves no cycles, but the revealed preference relation does.

```
In[1]:=
         p={{1,2,8},{4,1,8},{3,1,2}};
         x={{2,1,3},{3,4,2},{2,6,2}};
```

```
In[2]:=  m=rdMatrix[p,x]
```

```
Out[2]=  {{1, 1, 0}, {0, 1, 1}, {1, 0, 1}}
```

extract just the labor inputs and measure the efficiencies with respect to those labor choices only.

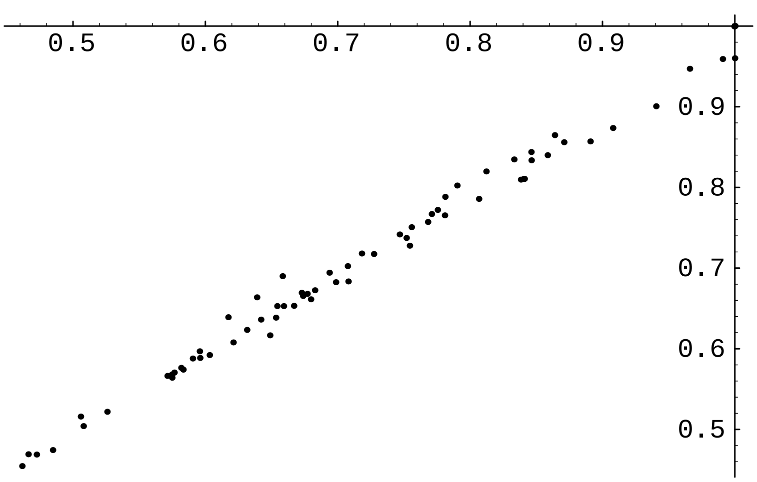
```
In[3]:=  wlabor=Transpose[{Transpose[w][[1]],Transpose[w][[2]]}];
         xlabor=Transpose[{Transpose[x][[1]],Transpose[x][[2]]}];

In[4]:=  effLabor=costEfficiency[wlabor,xlabor,y]

Out[4]=  {0.81052, 0.521896, 0.87362, 0.672513, 1, 0.50412, 0.596858,
           0.652904, 0.587924, 0.741708, 0.588623, 0.737364, 1, 1, 1,
           0.855984, 0.95915, 0.772125, 0.652891, 0.564115, 0.46887,
           0.682503, 0.864779, 0.757165, 0.694296, 0.727773, 1.,
           0.454564, 0.834703, 0.810626, 0.718089, 1, 1, 0.802351, 1,
           0.960112, 0.663749, 0.788278, 0.84386, 0.702435, 1, 1.,
           0.469225, 0.653222, 0.574132, 0.639108, 0.636164, 0.809674,
           0.857015, 0.839908, 1, 0.765367, 0.623384, 0.689988, 1,
           0.833604, 0.683464, 0.616643, 0.566365, 0.638567, 0.570566,
           0.568162, 0.515951, 0.947074, 0.665469, 1, 1, 0.999564,
           0.576357, 0.592193, 0.60788, 0.785772, 1, 0.668115,
           0.474429, 1, 0.750681, 0.900582, 0.669416, 0.717449,
           0.819808, 1, 1, 0.661303, 0.767011}
```

The two series look fairly similar. We do a scatterplot to compare the two lists of efficiency measures:

```
In[5]:=  ListPlot[Transpose[{effAll,effLabor}]]
```




```
Out[5]=  -Graphics-
```

Note that the two series are very highly correlated. It appears that labor efficiency alone tells most of the story about overall efficiency.

This function returns the list of $n \times k$ efficiency measures.

```
In[2]:=       eff[w_,x_,y_] := Map[Union,
                  Table[If[y[[i]] <= y[[j]],Min[efficOf[j,i,w,x],1],1],
                  {i,1,Length[y]},{j,1,Length[y]}]]
```

Finally, the following function takes the minimum of the efficiency measures just computed to calculate the actual list of efficiencies.

```
In[3]:=  costEfficiency[w_,x_,y_] := Map[Min,eff[w,x,y]]
```

Example of cost efficiency measurement

We illustrate the use of these tools with a dataset developed by Brooks (1993). These data describe the operation of a sample of VA pharmacies. There is a standard index of services for these pharmacies that serves as an output and several inputs. Inputs fall into two main categories: labor (pharmacist time, technician time) and capital (pill counting machines, office equipment, etc.)

First we set the directory and read in the data

```
In[1]:=  SetDirectory["/Users/hal/Math/Efficiency"]
         <<cost.dat;

Out[1]=  /Users/hal/Math/Efficiency
```

Now we compute the efficiencies.

```
In[2]:=  effAll = costEfficiency[w,x,y]

Out[2]=  {0.84057, 0.525841, 0.907895, 0.682745, 1, 0.50789, 0.595672,

           0.654224, 0.590418, 0.746752, 0.595972, 0.751879, 1, 1, 1,

           0.870959, 0.990814, 0.77547, 0.659193, 0.57482, 0.472514,

           0.698577, 0.863941, 0.768098, 0.693701, 0.754345, 1,

           0.461564, 0.833214, 0.841046, 0.718124, 1, 1, 0.790207, 1.,

           1, 0.638938, 0.781148, 0.846167, 0.707507, 1, 1, 0.466252,

           0.66688, 0.583254, 0.617269, 0.641974, 0.83834, 0.890836,

           0.85853, 1, 0.780865, 0.631422, 0.658296, 1, 0.846318,

           0.707987, 0.648803, 0.571282, 0.653275, 0.576511, 0.574738,

           0.505807, 0.965922, 0.673734, 1, 1, 0.999572, 0.581691,

           0.603172, 0.6211, 0.806635, 1, 0.676993, 0.484738, 1,

           0.755765, 0.94052, 0.672726, 0.727269, 0.812183, 1., 1,

           0.679689, 0.770955}
```

Note that the only thing that is controlled for in these efficiency measures are the factor prices. Nothing is done about type of hospital, patient mix, and so on. Clearly some of the observed "inefficiency" is likely to be due to such factors.

We can use the same technique to measure efficiency with respect to a subset of the factor inputs. Here we

Efficiency in Production and Consumption

Hal Varian August 1995

The standard economic models of firms and consumers emphasize optimizing behavior. Firms are assumed to minimize costs or maximize profits, while consumers are assumed to maximize utility. However, no one is perfect---even *homo economicus*. When analyzing real choice data it is necessary to allow for departures from this model of strict optimization.

In this chapter we consider some measures of choice efficiency. In general an efficiency measure should have the property that a value of 1 corresponds to 100 percent efficiency: full optimization. A value of 0 would correspond to behavior that is totally independent of efficient choice. Values between 0 and 1 should be interpretable as partial efficiency.

It turns out that there are relatively natural ways to construct such measures for economic choices and that *Mathematica* is a convenient computational engine for this task.

■ Cost minimization

Consider a firm that chooses factor inputs to minimize costs. Let w be a vector of factor prices, x a vector of factor choices, $f(x)$ a production function and y a target level of output. Then the hypothesized model is:

$$\min_x wx$$
$$\text{such that } f(x) \geq y.$$

Suppose that we have n observations of firm choices (w^t, x^t, y^t) for $t = 1, \ldots, n$. The following *Weak Axiom of Cost Minimization* (WACM) is necessary and sufficient for the data to be consistent with the cost minimization model:

WACM. $w^t x^s$ for all s for which $y^s \geq y^t$

This condition is rather intuitive; it say that the observed choice must cost no more than any other choice that generates at least as much output. See Varian (1993) for further discussion.

If we have some data that violate WACM, we may want to see "how close" a specific observation t comes to satisfying WACM. A natural thing to do is to compute the following efficiency index:

$$e^t = \min_{y^s \geq y^t} \frac{w^t x^s}{w^t x^t}$$

If $e^t < 1$, then there is some observation that produces at least as much output as the t^{th} observation and costs less than the t^{th} observation. If $e^t = 1$, no such observation exists.

It is easy to compute this efficiency index using Mathematica. Suppose that we have n observations on k factors which we arrange in an array of dimension $n \times k$s. The following function measures the efficiency of observation i relative to observation j:

```
In[1]:=   efficOf[i_,j_,w_,x_] :=
            (w[[j]].x[[i]])/(w[[j]].x[[j]])
```