Alberta Government

Building Input-Output Model in SAS: Tips

Afiba Nyamekye
Office of Statistics and Information
Alberta Treasury Board and Finance

October 24, 2017

Edmonton SAS User Group Conference

Overview

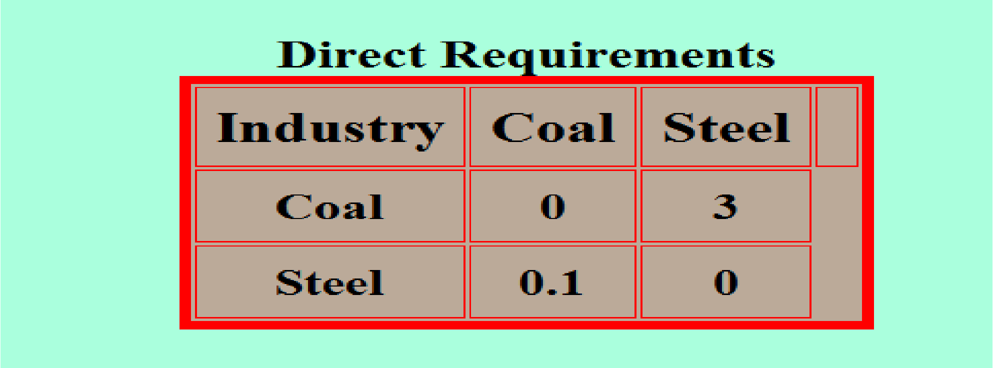
- Summary of Alberta’s Input-Output Model
  - *Data sources*
  - *Uses & limitations*
  - *Leontief inverse*

- Tips to build the model in SAS
  - *Base SAS and Proc IML (interactive matrix language) examples*

- Questions?

What’s an Input-Output (I/O) Model?

- (I/O) Model
  - tool to understand the impact of a change affecting the whole value chain
  - estimates impacts on all domestic industries involved in production of good(direct+indirect), plus impacts from changes in household income



Direct Requirements

Industry	Coal	Steel	
Coal	0	3	
Steel	0.1	0	

Assumptions & Limitations

- Model simulates the economy at a point in time
- Does not account for technological changes in either process or products
  - Fixed prices
  - Constant industry input coefficients and ratios for production, imports, inventory withdrawals
  - assumes that households do not change their consumption preferences
- Lack of supply-side constraints

How is it Used?

- IO analysis: economic impact from exogenous shocks to the economy
  - estimate impact of current events (e.g., forest fires, carbon levy, plant closure)
  - tax revenue impacts
- Derive the annual multipliers (GDP, output, jobs)
  - available on Open Data
- Custom detailed analysis for public (for a fee)

Alberta’s IO Model

- Statistics Canada’s Supply-UseTables
  - confidential data list of 473 commodities, 235 industries
  - imports, exports, inventory withdrawals, etc.
  - confidential tax module
  - productivity tables (jobs, wages)

- Impacts
  - output, GDP/components, employment (jobs, FTEs)
  - taxes, exports, import/interprovincial trade flows

Some linear algebra

- For now, ignore trade, gov’t and margins.
- In a 3-industry economy, a portion of an industry’s total output (x_n) is used as input, while the rest is consumed.
- You can describe the economy in the following system of equations:

$$
\begin{aligned}
a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + f_1 &= x_1 \\
a_{21} x_1 + a_{22} x_2 + a_{2\,3} x_3 + f_2 &= x_2 \\
a_{31} x_1 + a_{32} x_2 + a_{33} x_3 + f_3 &= x_3
\end{aligned}
$$

Leontief Inverse

- The inverse is called the Leontief inverse/impact matrix
  - Total output to produce $1 of final good/service
- Can calculate gross output (direct, indirect … and induced).

$$A\,x + f = I\,x, \text{ or}$$

$$f = (I - A)\,x.$$

$$\downarrow$$

$$(I - A)^{-1} f = (I - A)^{-1} (I - A)\,x = x$$

Solving the Leontief matrix in SAS

- Base SAS
  - create tables, industry and output shares and ratios

- Proc iml (interative matrix language)
  - vector/matrix operations, impacts
  - read in and create matrices of all the tables
  - matrix operations: element wise and matrix-wise
  - use functions in proc iml

Importing in proc iml

- “Use” statement loads SAS datasets in proc iml
- “Read” converts datasets into vectors/matrices
  - Default is numeric variables

```
proc iml;
      use commod3;
      read all var _num_ into commod3
      [colname=NumerNames];
      D=t(commod3);
      create d from D;
      append from D;
      print NumerNames; quit;
```

Maintain the order of commodities and industries

```
*order variables for industry;
proc sql noprint;
      select industry into :name_list separated by ", "
            from industries; quit;
%put &name_list;

*Format for order for commodities;
proc sort data=imports; by commodity; run;
data comm_format;
      set imports;
      fmtname='comm_fmt';
      type='J';
      start=code;
      label=commodity; run;
```

Controlling order (cont’d)

```
proc format cntlin=comm_format; run;

proc sql;
     create table input_6 as
          select comcode,
           input(input(comcode, $comm_fmt.), 8.)
as order_var, /*variable to control order of
commodities*/
     &name_list /*List of industries in order*/
     from input_5
          order by order_var;
quit;
```

Solving Leontief in SAS: Elementwise

```
proc iml;
      use inputs;
      read all var _all_ into B;
      colse;
      use imports;
      read all into UDiag;
      M=(UDiag+BetaDiag);
      create m from M;
      append from M;
      close; quit;
```

Impact matrix: stick to the inverse function

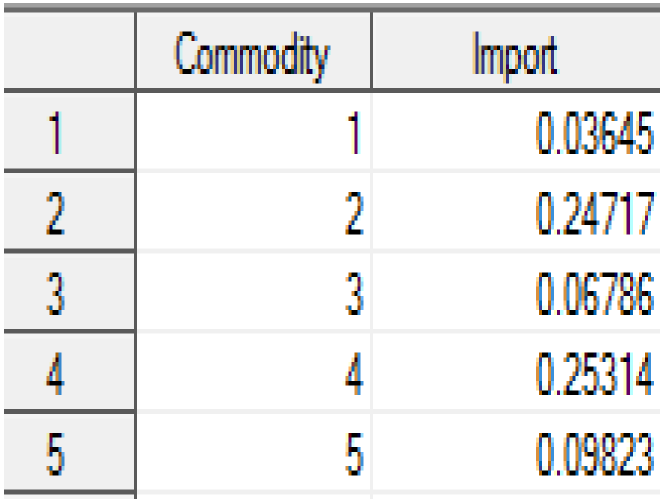
```
proc iml;
use  New_1;
read all var _all_ into new_1; close;
use A;
read all var _all_ into A; close;
isize=I(nrow(A)); /*identity matrix;
M3=inv(isize-A);
create m3 from M3;
append from M3;
quit;
```

Transpose

```
    proc iml;
    use outputs;
    read all var _all_ into outputs;
    D=t(outputs); /*Use “t” or `;
    create d from D;
    append from D;
quit;
```

Base SAS

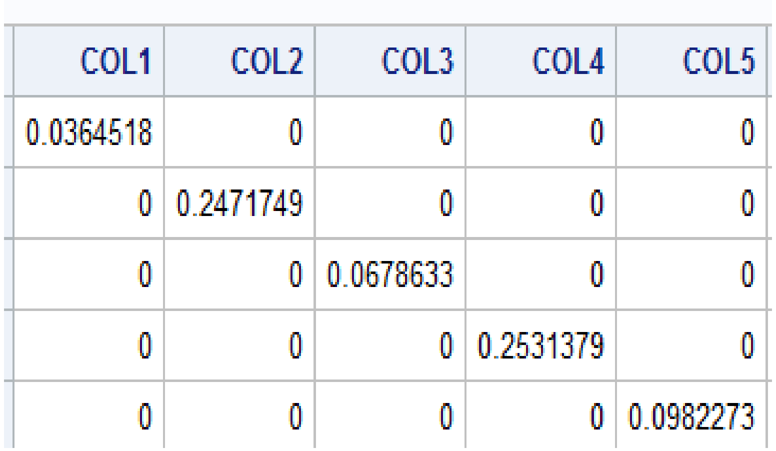
```
proc transpose data=imports out=imports_2 (drop=_NAME_);
      var import;
      id Code;
      by Code; run;
data imports_3;
      set imports_2;
      array change _numeric_;
      do i=1 to dim(change);
            if change(i)= . Then
                  change(i)=0; end;
      drop _NAME_  _LABEL_  i;
 run;
```



	Commodity	Import
1	1	0.03645
2	2	0.24717
3	3	0.06786
4	4	0.25314
5	5	0.09823

Proc iml: “Diag” function

```
proc iml;
     use imports;
     read all var _all_ into imports;
     close;
     udiag=diag(imports);
     create udiag from udiag;
     append from udiag;
     print (udiag[1:5,]);
quit;
```



COL1	COL2	COL3	COL4	COL5
0.0364518	0	0	0	0
0	0.2471749	0	0	0
0	0	0.0678633	0	0
0	0	0	0.2531379	0
0	0	0	0	0.0982273

Alberta
Government

Questions?

Afiba Nyamekye
Senior Economic Statistician
Office of Statistics and Information
Alberta Treasury Board and Finance