Matlab code to accompany "Using the Leontief Price Model to verify Output Multipliers derived from I-O SNAP's Input-Output Tables"

Author: Randall Jackson, March 1, 2024. Email: Jackson@EconAlyze.com

As a supplement to the IO-Snap Technical document, this file provides explanation and Matlab code for confirming the consistency of IO-Snap data by Using the Leontief Price Model to verify Output Multipliers derived from I-O SNAP's Input-Output Tables.

Preliminaries

Export the Use and Make tables from IO-Snap and import relevant variables.

```
clc
clear variables
```

Data Access

```
getMakeTable
getUseTable
```

The exported IO-Snap Make table reports imports in the last "industry" row, where this last industry is, effectively, a Rest-of-world industry.

```
getCommodityImports
```

Computations

Set the size of the problem.

```
[numberOfIndustries, numberOfCommodities] = size(Make);
```

Generate commodity and industry output vectors.

```
q = sum(Make,1);   % commodity output
g = sum(Make,2);   % industry output
```

Standardize the Use and Make matrix.

```
B = Use/diag(g);
D = Make/diag(q);
```

Generate coefficients matrix using a closed system D matrix. This closes system D has column sums of 1.0, indicating that all commodity is generated by domestic industries.

```
Acoefficients = D * B;
```

For an open regional system, total commodity supply includes domestic production and imports.

```
commoditySupply = q + commodityImports;
Dtilde = Make/diag(commoditySupply);
```

The column sums of Dtilde will generally be < 1. This relationship can be confounded by data on occasion. E.g., the BEA national accounts regularly report import values of the "wrong" sign for certain

sectors. Although this is true for the 2021 national accounts at the Summary level used here, the values are not extreme enough to lead to unexpected outcomes.

Generate a regional coefficients table that reflects import supply.

```
Rcoefficients = Dtilde * B;
```

Compute multipliers for both systems.

```
eyeInd = sparse(eye(numberOfIndustries));
AMultipliers = eyeInd/(eyeInd - Acoefficients);
RMultipliers = eyeInd/(eyeInd - Rcoefficients);
```

Total factor coefficients = the sum of value added coefficients and imports coefficients = 1 - sum(intermediate coefficients).

```
Atfc = 1 - sum(B);
Rtfc = 1 - sum(Rcoefficients);
```

The product of postmultipliying total factor coefficients by the corresponding multiplier matrix generates a vector of ones. Differences from 1.0 are due to precision limitations.

```
resultA = round((Atfc * AMultipliers),3);
resultR = round((Rtfc * RMultipliers),3);
```

Results

The following two statements should result in empty vectors:

```
find(resultA ~= 1)
```

```
ans =

  1×0 empty double row vector
```

```
find(resultR ~= 1)
```

```
ans =

  1×0 empty double row vector
```

```
clear ans eyeInd;
```

Save the results for future use.

```
save verificationResults
```

Alternative (Technical Document) Computations

```
industryImports = (commodityImports./q) * B;
getValueAdded
tfc = (sum(va) + industryImports)./g';
resultDK = round((tfc * AMultipliers),3);
find(resultDK ~= 1)
```

```
ans =

  1×0 empty double row vector
```