Chapter 16: Contributions

In this chapter, I present a tool that I have developed in 1990 at the Forecasting Department of the French Ministry of Finance and that allows to split the evolution of an endogenous variable into the contributions of the values, past and present, of its exogenous determinants.

The first part of this chapter presents the methodology, the second one how to use the corresponding Grocer functions and the third one the precise description of the functions and their specifications.

1. The methodology

1.1 Basic principles

Assume that an endogenous variable y_t is described by the following equation:

$$A^*(L)y_t = B_1^*(L)(x_{1,t}) + \dots + B_k(L)^*(x_{k,t}) + u_t$$

With L the lag operator ($L\ y_t = y_{t-1}$) and A*(0) = 1.

Then you can write:

$$y_t = \frac{B_1^*(L)}{A^*(L)} x_{1,t} + \dots + \frac{B_k^*(L)}{A^*(L)} x_{k,t} + \frac{1}{A^*(L)} u_t$$

And:

$$\Delta y_t = \frac{B_1^*(L)}{A^*(L)} \Delta x_{1,t} + \dots + \frac{B_k^*(L)}{A^*(L)} \Delta x_{k,t} + \frac{1}{A^*(L)} \Delta u_t$$

This formulation shows how the variation of the endogenous variable can be explained by the contribution of the exogenous variables ($x_{1,t}, \dots, x_{k,t}$) and of the residuals, with contribution of variable j defined as:

$$\frac{B_j^*(L)}{A^*(L)} \Delta x_{j,t}$$

The calculation of these contribution involves the infinite Moving Average decomposition of an ARMA model. It can be performed by standard methods. For instance, if p and q are the respective degrees of the polynoms A*(L) and B_1*(L) and C(L) the corresponding infinite moving average representation, then one can calculate the coefficients of C with the corresponding algorithm:

C(0) = B*(0)
For i = 1 à q :

$$\left\{ C(i) = B_1^*(i) - \sum_{j=1}^{\min(p,i-1)} A^*(j) * C^*(i-j) \right.$$

For i = q+1 à N

$$\left\{ C(i) = - \sum_{j=1}^{\min(p,i-1)} A^*(j) * C^*(i-j) \right.$$

This algorithm has been programmed in Scilab: this is Grocer function mainf.

The function takes as first argument the vector (1-A*(L)), as second argument the vector B_i*(L). For instance, take an equation:

$$y_t = 0.8\,y_{t-1} + 0.4\,x_t - 0.1\,x_{t-1} + u_t$$

Then you can calculate the Moving Average Infinite representation by executing:

x_mainf = mainf(0.8,[0.4 –0.1])

and you will obtain the 100 first terms of this representation. If you want another number of terms, then add as a third argument the # of terms:

x_mainf = mainf(0.8,[0.4 –0.1],200)

to obtain 200 terms.

And the Grocer function contrib calculates the contribution of a variable x with the infinite moving average representation. Assume that you have calculated x_mainf as above; then

x_contrib=contrib(delts(x),x_contrib)

will calculate the contribution of Δ(x) to Δ(y).

With the equation above, to obtain all the contributions to Δ(y), you have to calculate the contribution of the residuals:

res_mainf = mainf(0.8,1,200);

resid = y - 0.8 * lagts(y) - 0.4 * x - 0.1 * lagts(x)

res_contrib=contrib(delts(resid),resid_contrib)

1.2 Dealing with the numerical discrepancy between an endogenous variable and its contributions.

The decomposition above is not exact for 2 reasons:

- the algorithm can only provide a finite number of terms;
- the data do not go back to minus infinite; at best they can only go back to the big-bang…

In practice, you can ignore the discrepancy or distribute it one way or another on all the contributions. Grocer proposes a method, programmed in function balance_identity, that distributes the discrepancy promotionally to the absolute value of each contribution.

The function takes as first argument the lhs of the equation. The other arguments are the rhs terms whose addition should be equal to the lhs side of the equation. You can also optionally add as an argument a string 'prefix=xxxx'. In that case, the new rhs arguments will keep the name of the input arguments, with prefix xxxx and the function has no direct output.

If there is no argument beginning with 'prefix', then you have to define the names of the new rhs variables through the output.

For instance:

- balance_identity(y,'x1','x2','x3','prefix=new')

  will create in the calling environment 3 new variables called new_x1, new_x2 and new_x3 that sum to y: y= new_x1+new_x2+new_x3
- [newx1,newx2,newx3] =balance_identity(y,x1,x2,x3)

will create in the calling environment 3 new variables called newx1, newx2 and newx3 that sum to y: y= newx1+newx2+newx3

Note the quotes around x1, x2 and x3 in the first type of call: as usual, they are necessary to transmit the names of the variables to the function. If the quotes were absent, then the new variables should be named: new_exogenous1, new_exogenous2 and new_exogenous3.

1.3 Extensions

The method can be rigorously applied only to linear models. It can however be extended to any non-linear problems by the mean of a first order approximation. In Grocer, one such problem is dealt, through function contrib_logq2gra. This function calculates the annual contributions to the growth rate of an endogenous variable while the underlying equation is estimated on the quarterly logarithm of the variable. The corresponding first order approximation is the following one:

$$\text{contrib}A_i = \sum_{k=2}^{4} w_{A-1,k}\,\text{contrib}_{i,A-1,k} + \sum_{k=1}^{4} w_{A,k}\,\text{contrib}_{i,A,k}$$

$$w_{A-1,i} = \frac{\partial\left(YA/YA(-1)-1\right)}{\partial\, y_{A-1,i}} = \frac{\prod_{j=2}^{4}\left(\exp(y_{A-1,j})\right)\sum_{k=1}^{4}\left(\prod_{j=1}^{i}\exp(y_{A,j})\right)}{1+\sum_{i=2}^{4}\prod_{j=2}^{i}\left(\exp(y_{A-1,j})\right)} - \frac{\prod_{j=2}^{4}\left(\exp(y_{A-1,j})\right)\sum_{k=i}^{4}\left(\prod_{j=1}^{i}\exp(y_{A,j})\right)}{\left(1+\sum_{i=2}^{4}\prod_{j=2}^{i}\left(\exp(y_{A-1,j})\right)\right)^2}$$

$$w_{A,i} = \frac{\partial\left(YA/YA(-1)-1\right)}{\partial\, y_{A,i}} = \frac{\prod_{j=2}^{4}\left(\exp(y_{A-1,j})\right)\sum_{k=i}^{4}\left(\prod_{j=1}^{i}\exp(y_{A,j})\right)}{1+\sum_{i=2}^{4}\prod_{j=2}^{i}\left(\exp(y_{A-1,j})\right)}$$

with:

- $y_{A,j}$ = variation of the logarithm of variable Y in quarter j of year A.
- YA = level of variable Y for the year A
- $\text{contrib}_{i,A,k}$ = contribution of variable i in quarter k of year A.
- $\text{contrib}A_i$ = contribution of variable i in year A.

The first argument of the function is the lhs side of the equation. The following arguments are lists. Each list must have the 3 following elements, in that order:

- the name of an exogenous variable (between quotes)
- the AR part of the equation
- the Ma part of the equation corresponding to the considered exogenous variable

The output has 3 arguments:

- first, the list of the yearly contributions, that add to the annual growth rate of the first input argument;
- second, the list of the quarterly contributions;
- third, the matrix of infinite moving averages.

2. An example

As an example, take our favorite equation, taken from Hendry and Ericsson (1991). This equation applies to quarterly data written in logarithm: this is exactly the case when one can apply Grocer function contrib_logq2gra.

```
-->load('SCI/macros/grocer/db/bdhenderic.dat') ;
-->m1=exp(lm1);
```

// set the estimation bounds

```
bounds('1964q3','1989q2')
```

// estimate the model by ols and save the results in the tlist
// rols; names are entered between quotes, in order that their
// names appears on the printed results

```
-->rols=ols('delts(lm1-lp)','delts(lp)','delts(lagts(1,lm1-lp-ly))','rnet'...
-->,'lagts(1,lm1-lp-ly)','cte')
```

// results not shown here (see chapter 6 for the results of this estimation)
// retrieve the coeficients of the regression

```
-->coef=rols('beta');
```

// calculate the residual over the whole period

```
-->resid=delts(lm1-lp)-coef(1)*delts(lp)-coef(2)*delts(lagts(1,lm1-lp-ly))-
-->coef(3)*rnet -coef(4)*lagts(1,lm1-lp-ly)-coef(5);
```

// clear bounds

```
bounds();
```

// calculate the ma infinite approximations

```
-->mainf_lp=mainf([1+coef(2)+coef(4) ; -coef(2)],...
--> [1+coef(1) ; -1-coef(1)-coef(2)-coef(4) ; coef(2) ],200);
-->mainf_ly=mainf([1+coef(2)+coef(4) ; -coef(2)],...
--> [0 ; -coef(2)-coef(4) ; coef(2) ],200);
-->mainf_resid=mainf([1+coef(2)+coef(4) ; -coef(2)],1,200);
-->mainf_rnet=coef(3)*mainf_resid;
```

// calculate the contributions

```
contrib_lp=contrib(delts(lp),mainf_lp);
contrib_ly=contrib(delts(ly),mainf_ly);
contrib_resid=contrib(delts(resid),mainf_resid);
contrib_rnet=contrib(delts(rnet),mainf_rnet);
```

// now calculates annual growthr contributions

```
-->he_lp_list=list('lp',[1+coef(2)+coef(4) ; -coef(2)],[1+coef(1) ; -1- ...
-->coef(1)- coef(2)-coef(4) ; coef(2) ]);
--> he_ly_list=list('ly',[1+coef(2)+coef(4) ; -coef(2)],[0 ; -coef(2)-coef(4)...
-->; coef(2) ]);
-->he_rnet_list=list('rnet',[1+coef(2)+coef(4) ; -coef(2)],coef(3));
-->he_resid_list=list('resid',[1+coef(2)+coef(4) ; -coef(2)],1);
```

// apply function contrib_log2a to the endogenous variable m1 and the calculated quarterly

// contributions he_lp_list,he_ly_list,he_rnet_list,he_resid_list
// all intermediate results are printed on screen (option 'prt=all')

```
--> [listcont_unbal,listcont_bal]=contrib_logq2gra('m1',he_lp_list,he_ly_list,...
-->he_rnet_list,he_resid_list,'prt=all');
```

// the macro first displays the results of the mainf function for each variable

```
Impulse response functions of endogenous variable m1 to:

lp        ly        rnet       resid
 .3129616 0          -.6296264 1
 .1837571  .2674627  -.4612249  .7325373
 .0146472  .0213193  -.4478016  .7112180
 .0428149  .0623181  -.4085645  .6488999
 .0339210  .0493728  -.3774781  .5995271
 .0323242  .0470486  -.3478550  .5524784
 .0296015  .0430857  -.3207271  .5093927
 .0273283  .0397769  -.2956825  .4696158
 .0251876  .0366611  -.2725997  .4329547
 .0232226  .0338010  -.2513177  .3991537
 .0214093  .0311618  -.2316974  .3679919
 .0197380  .0287291  -.2136089  .3392629
 .0181970  .0264862  -.1969325  .3127767
 .0167764  .0244184  -.181558   .2883583
 .0154667  .0225121  -.1673838  .2658462
 .0142592  .0207546  -.1543162  .2450917
 .0131460  .0191342  -.1422688  .2259574
 .0121197  .0176404  -.1311619  .2083170
 .0111735  .0162633  -.1209221  .1920537
 .0103012  .0149936  -.1114817  .1770601
 .0094970  .0138230  -.1027784  .1632371
 .0087555  .0127439  -.0947545  .1504932
 .0080720  .0117490  -.0873570  .1387442
 .0074418  .0108317  -.0805371  .1279125
 .0068608  .0099861  -.0742496  .1179264
 .0063252  .0092065  -.0684529  .1087199
 .0058314  .0084877  -.0631088  .1002322
 .0053761  .0078251  -.0581819  .0924071
 .0049564  .0072142  -.0536397  .0851929
 .0045695  .0066510  -.0494521  .0785419
 .0042127  .0061317  -.0455913  .0724101
 .0038839  .0056530  -.0420320  .0667571
 .0035806  .0052117  -.0387506  .0615454
 .0033011  .0048048  -.0357254  .0567406
 .0030434  .0044297  -.0329363  .0523108
 .0028058  .0040839  -.0303650  .0482269
 .0025867  .0037651  -.0279944  .0444619
 .0023848  .0034711  -.0258089  .0409907
 .0021986  .0032001  -.0237940  .0377906
 .0020270  .0029503  -.0219364  .0348403
 .0018687  .0027200  -.0202238  .0321203
 .0017228  .0025076  -.0186449  .0296127
 .0015883  .0023119  -.0171893  .0273008
 .0014643  .0021314  -.0158474  .0251695
 .0013500  .0019650  -.0146102  .0232045
 .0012446  .0018116  -.0134696  .0213929
 .0011475  .0016701  -.0124180  .0197228
 .0010579  .0015398  -.0114485  .0181830
 .0009753  .0014195  -.0105547  .0167635
 .0008991  .0013087  -.0097307  .0154548
```

```
.0008289  .0012066  -.0089711  .0142482
.0007642  .0011124  -.0082707  .0131359
.0007046  .0010255  -.0076250  .0121103
.0006496  .0009455  -.0070297  .0111649
.0005989  .0008716  -.0064809  .0102933
.0005521  .0008036  -.0059749  .0094897
.0005090  .0007409  -.0055085  .0087488
.0004693  .0006830  -.0050784  .0080658
.0004326  .0006297  -.0046820  .0074361
.0003988  .0005805  -.0043164  .0068556
.0003677  .0005352  -.0039795  .0063204
.0003390  .0004934  -.0036688  .0058269
.0003125  .0004549  -.0033824  .0053720
.0002881  .0004194  -.0031183  .0049526
.0002656  .0003867  -.0028749  .0045660
.0002449  .0003565  -.0026504  .0042095
.0002258  .0003286  -.0024435  .0038809
.0002082  .0003030  -.0022527  .0035779
.0001919  .0002793  -.0020769  .0032986
.0001769  .0002575  -.0019147  .0030411
.0001631  .0002374  -.0017652  .0028036
.0001504  .0002189  -.0016274  .0025848
.0001386  .0002018  -.0015004  .0023830
.0001278  .0001860  -.0013832  .0021969
.0001178  .0001715  -.0012753  .0020254
.0001086  .0001581  -.0011757  .0018673
.0001002  .0001458  -.0010839  .0017215
.0000923  .0001344  -.0009993  .0015871
.0000851  .0001239  -.0009213  .0014632
.0000785  .0001142  -.0008494  .0013490
.0000724  .0001053  -.0007830  .0012437
.0000667  .0000971  -.0007219  .0011466
.0000615  .0000895  -.0006656  .0010571
.0000567  .0000825  -.0006136  .0009745
.0000523  .0000761  -.0005657  .0008985
.0000482  .0000701  -.0005215  .0008283
.0000444  .0000647  -.0004808  .0007636
.0000410  .0000596  -.0004433  .0007040
.0000378  .0000550  -.0004087  .0006491
.0000348  .0000507  -.0003768  .0005984
.0000321  .0000467  -.0003473  .0005517
.0000296  .0000431  -.0003202  .0005086
.0000273  .0000397  -.0002952  .0004689
.0000252  .0000366  -.0002722  .0004323
.0000232  .0000337  -.0002509  .0003985
.0000214  .0000311  -.0002313  .0003674
.0000197  .0000287  -.0002133  .0003387
.0000182  .0000264  -.0001966  .0003123
.0000168  .0000244  -.0001813  .0002879
.0000154  .0000225  -.0001671  .0002654
.0000142  .0000207  -.0001541  .0002447
.0000131  .0000191  -.0001421  .0002256
.0000121  .0000176  -.0001310  .0002080
.0000112  .0000162  -.0001207  .0001918
.0000103  .0000150  -.0001113  .0001768
.0000095  .0000138  -.0001026  .0001630
.0000087  .0000127  -.0000946  .0001503
.0000081  .0000117  -.0000872  .0001385
```

// the macro then displays the variation in the logarithm of the endogenous variable and the quarterly
// contributions

quarterly growthr rate of and contributions to:

dates	delts(log(m1))	cont. of lp	cont. of ly	cont. of rnet
1963q2	.0219412	.0014842	0	-.0005247
1963q3	.0050507	.0033803	.0104319	.0008119
1963q4	.0304794	.0058617	.0016572	-.0004203
1964q1	.0097256	.0031002	.0081101	-.0035871
1964q2	.0059290	.0050952	.0067671	-.0033517
1964q3	.0154133	.0071684	.0073011	-.0035223
1964q4	-.0082912	.0058653	.0039177	-.0140160
1965q1	.0142288	.0075954	.0104364	-.0154575
1965q2	.0109730	.0070781	.0008835	-.0090483
1965q3	-.0030661	.0071747	.0065277	-.0072285
1965q4	.0243969	.0064242	.0063741	-.0057207
1966q1	.0122995	.0068841	.0071031	-.0053898
1966q2	-.0099573	.0087498	.0049484	-.0055348
1966q3	.0156040	.0085071	.0049205	-.0121943
1966q4	-.0187283	.0076568	.0058213	-.0087868
1967q1	.0247240	.0044496	.0038601	-.0026474
1967q2	.0118856	.0072081	.0108829	.0010388
1967q3	.0344667	.0064882	.0042854	.0006975
1967q4	.0025473	.0059734	.0073484	-.0099274
1968q1	.0049534	.0102183	.0044972	-.0118096
1968q2	.0151911	.0124523	.0151260	-.0118745
1968q3	.0053272	.0108352	.0008085	-.0071927
1968q4	.0139542	.0104159	.0126247	-.0058312
1969q1	-.0179768	.0103984	.0073051	-.0121595
1969q2	-.0213280	.0090504	.0032377	-.0146896
1969q3	.0076008	.0095575	.0074897	-.0151295
1969q4	.0325184	.0114047	.0070698	-.0096392
1970q1	.0058002	.0146937	.0077677	-.0105633
1970q2	.0315390	.0140202	.0013475	-.0027471
1970q3	.0090380	.0146116	.0122180	-.0004778
1970q4	.0224056	.0152389	.0056116	.0011222
1971q1	.0658827	.0162941	.0103428	-.0012132
1971q2	.0057793	.0170331	.0001151	.0055405
1971q3	.0327206	.0163945	.0121015	.0083591
1971q4	.0137987	.0154376	.0083991	.0134710
1972q1	.0408630	.0141759	.0071827	.0102572
1972q2	.0454130	.0146016	.0052797	.0052906
1972q3	.0263233	.0169115	.0096248	-.0073457
1972q4	.0260515	.0193284	.0038457	-.0065560
1973q1	.0056448	.0199918	.0171887	-.0186776
1973q2	.0596972	.0145969	.0180156	-.0074811
1973q3	-.0188875	.0207460	.0074100	-.0327302
1973q4	.0053892	.0313353	.0109624	-.0380008
1974q1	-.0126724	.0311275	.0060685	-.0386729
1974q2	.0282898	.0376492	.0035791	-.0208190
1974q3	.0223503	.0363686	.0099336	-.0185823
1974q4	.0632296	.0362901	.0080727	-.0161770
1975q1	.0185623	.0415898	.0025945	-.0073714
1975q2	.0420179	.0433346	.0025162	.0018915
1975q3	.0440283	.0397670	.0009806	-.0051484
1975q4	.0213375	.0362139	.0033713	-.0088978
1976q1	.0487204	.0337708	.0059068	.0062250
1976q2	.0283632	.0354779	.0073595	-.0072775
1976q3	.0224609	.0346754	.0051717	-.0091762
1976q4	.0074291	.0363037	.0070700	-.0294629
1977q1	.0275631	.0359218	.0083658	.0001952
1977q2	.0426989	.0347099	.0000206	.0171405
1977q3	.0696192	.0301204	.0055286	.0198938

1977q4	.0556469	.0257752	.0040765	.0213016
1978q1	.0574303	.0269459	.0081021	.0162445
1978q2	.0244398	.0279689	.0077975	-.0001216
1978q3	.0349073	.0275096	.0062953	-.0008625
1978q4	.0323629	.0295684	.0057728	-.0142320
1979q1	.0293207	.0284883	.0044985	-.0174600
1979q2	.0198765	.0293719	.0033642	-.0138270
1979q3	.0270882	.0369022	.0196649	-.0226848
1979q4	.0101595	.0381193	.0002757	-.0318971
1980q1	-.0010118	.0367723	.0078313	-.0397304
1980q2	.0246057	.0370692	.0030216	-.0286290
1980q3	-.0045032	.0337319	-.0012812	-.0192622
1980q4	.0246563	.0314804	.0015641	-.0158350
1981q1	.0317192	.0289228	-.0024697	-.0007755
1981q2	.0405683	.0271795	-.0001301	.0006547
1981q3	.0172920	.0271014	.0020811	-.0144520
1981q4	.0119162	.0286526	.0066707	-.0161390
1982q1	.0211887	.0249857	.0000280	-.0037681
1982q2	.0120344	.0217399	.0031344	-.0014647
1982q3	.0299997	.0223238	.0042754	.0122468
1982q4	.0462532	.0231883	.0001477	.0157883
1983q1	.0420666	.0224564	.0038656	.0059805
1983q2	.0211029	.0173340	.0083229	.0150080
1983q3	.0135593	.0184037	.0032343	.0132135
1983q4	.0326357	.0199230	.0075628	.0157418
1984q1	.0549116	.0173693	.0050563	.0143423
1984q2	.0347853	.0181461	.0066349	.0122055
1984q3	.0341597	.0190012	.0045891	.0009239
1984q4	.0197163	.0194988	.0063384	.0118720
1985q1	.0577211	.0203635	.0083450	-.0025666
1985q2	.0312939	.0154016	.0080716	.0189924
1985q3	.0446253	.0113413	.0060489	.0325651
1985q4	.0394869	.0120337	.0044199	.0346233
1986q1	.0638609	.0121056	.0066707	.0348500
1986q2	.0535231	.0091253	.0086988	.0376032
1986q3	.0624247	.0098762	.0078996	.0312000
1986q4	.0305958	.0145707	.0097281	.0241389
1987q1	.0645914	.0136706	.0080968	.0292041
1987q2	.059021	.0119211	.0066264	.0265305
1987q3	.0374842	.0128472	.0117763	.0187644
1987q4	.0430756	.0128946	.0129134	.0249163
1988q1	.0502539	.0107022	.0096237	.0194407
1988q2	.0409851	.0110588	.0097068	.0189801
1988q3	.0265350	.0155110	.0117426	.0046007
1988q4	.0167170	.0151102	.0123881	.0058341
1989q1	.0472269	.0125118	.0100059	.0078159
1989q2	.0382376	.0147226	.0122211	.0028194
1989q3		.0136752	.0079130	.0003658
1989q4				-.0028496

```
                        *
                     *     *
```

dates	cont. of resid
1963q2	
1963q3	
1963q4	.0267302
1964q1	.0062276
1964q2	.0010250
1964q3	.0078284
1964q4	-.0009656

1965q1	.0145071
1965q2	.0146894
1965q3	-.0071155
1965q4	.0195545
1966q1	.0057628
1966q2	-.0162210
1966q3	.0161221
1966q4	-.0218048
1967q1	.0205504
1967q2	-.0058718
1967q3	.0242610
1967q4	.0003193
1968q1	.0031230
1968q2	.0004787
1968q3	.0017904
1968q4	-.0024125
1969q1	-.0227438
1969q2	-.0182101
1969q3	.0063434
1969q4	.0242919
1970q1	-.0055366
1970q2	.0194358
1970q3	-.0168368
1970q4	.0008727
1971q1	.0408644
1971q2	-.0165356
1971q3	-.0037900
1971q4	-.0231912
1972q1	.0095402
1972q2	.0205111
1972q3	.0073816
1972q4	.0096629
1973q1	-.0126465
1973q2	.0347609
1973q3	-.0141334
1973q4	.0012581
1974q1	-.0110427
1974q2	.0080214
1974q3	-.0052398
1974q4	.0351636
1975q1	-.0181401
1975q2	-.0056225
1975q3	.0085230
1975q4	-.0092633
1976q1	.0028976
1976q2	-.0071231
1976q3	-.0081422
1976q4	-.0064192
1977q1	-.0168621
1977q2	-.0091189
1977q3	.0141254
1977q4	.0045388
1978q1	.0061795
1978q2	-.0111666
1978q3	.0020004
1978q4	.0112864
1979q1	.0138240
1979q2	.0009951
1979q3	-.0067686
1979q4	.0036852
1980q1	-.0058634

```
1980q2  .0131640
1980q3  -.0176733
1980q4  .0074638
1981q1  .0060573
1981q2  .0128788
1981q3  .0025749
1981q4  -.0072558
1982q1  -.0000455
1982q2  -.0113647
1982q3  -.0088366
1982q4  .0071377
1983q1  .0097723
1983q2  -.0195544
1983q3  -.0212851
1983q4  -.0105855
1984q1  .0181495
1984q2  -.0021959
1984q3  .0096504
1984q4  -.0179883
1985q1  .0315834
1985q2  -.0111678
1985q3  -.0053264
1985q4  -.0115866
1986q1  .0102377
1986q2  -.0019014
1986q3  .0134516
1986q4  -.0178395
1987q1  .0136221
1987q2  .0139450
1987q3  -.0059019
1987q4  -.0076471
1988q1  .0104889
1988q2  .0012408
1988q3  -.0053178
1988q4  -.0166141
1989q1  .0168946
1989q2  .0084756
1989q3
1989q4
```

```
                        *
                    *     *
```

// the macro then displays the ratio of the sum of the contributions to the annual growth rate of m1:
// this allows the user to see the validity of the approximation and the relative
// importance of the residuals distributed to the contributions; here it is relatively small (between 5
// and 10% of the growth arte of m1)

```
 part of the annual growth rate of m1 explained by its contribution:

1965a 1.4052184
1966a 1.3182431
1967a 1.1698824
1968a 1.1183067
1969a  .3750995
1970a 1.0713169
1971a 1.0751854
1972a 1.0765659
1973a 1.0489913
1974a 1.0514073
```

1975a 1.0762307
1976a 1.0647021
1977a 1.084903
1978a 1.0887823
1979a 1.052588
1980a 1.0230429
1981a 1.0520281
1982a 1.0465179
1983a 1.061772
1984a 1.0737691
1985a 1.0820193
1986a 1.1085395
1987a 1.10485
1988a 1.0786648
annual growth rate of and contributions to:

dates	growthr(m1)	cont. of lp	cont. of ly	cont. of rnet
1964a	.0517942	.0193428	.0254803	-.0120297
1965a	.0304906	.0282275	.0249885	-.0416849
1966a	.0277953	.0305104	.0235578	-.0286722
1967a	.0439095	.0281321	.0253040	-.0189306
1968a	.0454673	.0387964	.0310951	-.0341643
1969a	-.0050624	.0408865	.0286926	-.0438453
1970a	.0645696	.0555614	.0279416	-.0329766
1971a	.1266178	.0714177	.0329657	.0119896
1972a	.1310669	.0709537	.0332169	.0274255
1973a	.0897749	.0821184	.0502545	-.0627374
1974a	.0467499	.1314214	.0338134	-.1203053
1975a	.1537985	.1822791	.0201672	-.0423678
1976a	.1399754	.1638346	.0222163	-.0229291
1977a	.1434161	.1580000	.0251451	.0017763
1978a	.2056918	.1339267	.0293791	.0491686
1979a	.1138201	.1349815	.0293266	-.0653249
1980a	.0485013	.1503373	.0215078	-.1188006
1981a	.1011804	.1321266	.0016266	-.0441864
1982a	.0884340	.1090009	.0115268	-.0144991
1983a	.1334279	.0944097	.0189515	.0514474
1984a	.1500574	.0851442	.0268300	.0563275
1985a	.1668756	.0807134	.0318797	.0585660
1986a	.2253797	.0565362	.0345248	.1619949
1987a	.2282881	.0619523	.0437990	.1316423
1988a	.1775527	.0581387	.0505129	.0838060

*
* *

dates	cont. of resid
1964a	
1965a	.0313149
1966a	.0112450
1967a	.0168634
1968a	.0151191
1969a	-.0276327
1970a	.0186481
1971a	.0197646
1972a	.0095060
1973a	.0245376
1974a	.0042236
1975a	.0054441
1976a	-.0140896
1977a	-.0293287

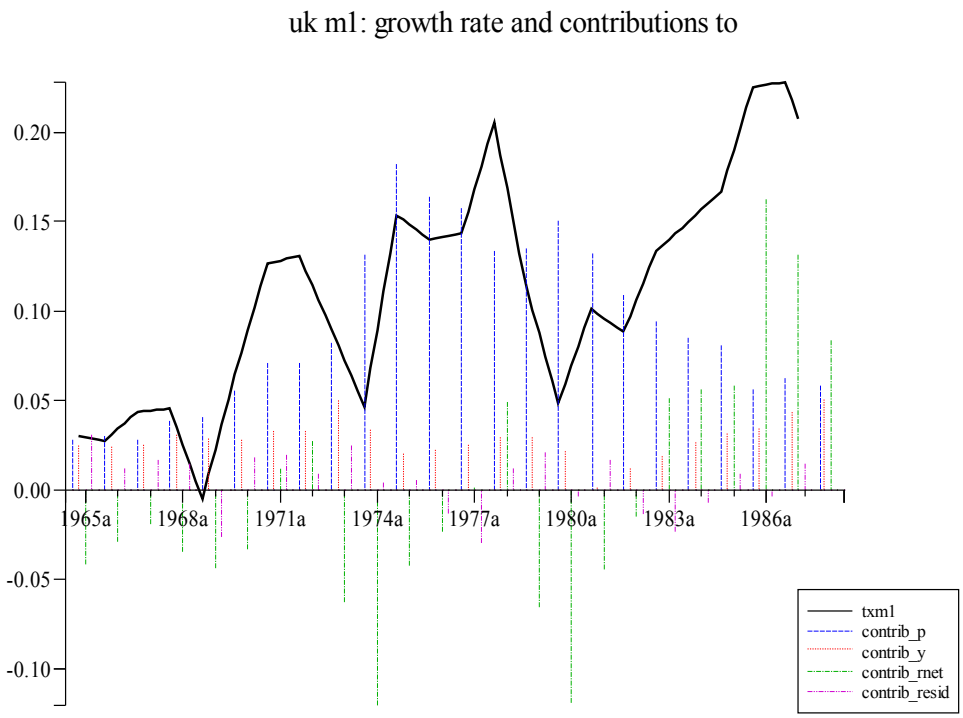
```
1978a  .0114793
1979a  .0208224
1980a  -.0034255
1981a  .0168778
1982a  -.0134808
1983a  -.0231386
1984a  -.0071747
1985a  .0094036
1986a  -.0032136
1987a  .0148305
1988a  -.0009378

                        *
                    *     *


-->txm1=growthr(q2a(m1,0));
-->contrib_p=listcont_unbal(1);
-->contrib_y=listcont_unbal(2);
-->contrib_rnet=listcont_unbal(3);
-->contrib_resid=listcont_unbal(4);
```

// plot the annual growth rate of m1 (as a curve) and the corresponding contributions (as bars) – as
// defined by the option 'bars=[0 1 1 1 1]' – , with title 'uk m1: growth rate and contributions to' and
// the legend in the right corner –as defined by the option styleg=4- (see chapter 21 for a description
// of grocer graphic functions)

```
-->pltseries('txm1','contrib_p','contrib_y','contrib_rnet','contrib_resid',...
-->'title=uk m1: growth rate and contributions to','bars=[0 1 1 1 1]','styleg=4')
```



3. The contribution functions and their specifications.

balance_identity_____balance an identity that does not hold exactly

CALLING SEQUENCE

[argout1,…,argoutn]=balance_identity(namey,arg1,…,argn)

PARAMETERS

INPUT:
* namey = a ts, a vector, or a string representing the name of a ts or a vector, equal to the left hand side of an identity
* arg1,…, argn = right hand side arguments that can be :
- a ts
- a matrix
- a string representing the name of a ts or a vector
- a list of such objects
- the string 'prefix=xxxx' where xxxx is the prefix added to the name of the variables to produce the new variables (optional: if not given, the names are given by the user trough the output)

OUTPUT:
argout1,.. argoutn =
- either the new rhs terms of the identity
- or the string 'results of macro balance_identity have been saved as variables: ' + the names of the variables

the new rhs terms are in that case put in the calling environment under the original name (exogenousi if the name was not entered between quotes) prefixed by the prefix given by the user

DESCRIPTION
Balances an identity that does not hold exactly; the residual is shared on each term according to its absolute weight.

Example:
1) z=grand(20,4,'nor',0,1); x1=z(:,1);x2=z(:,2);x3=z(:,3);y=x1+x2+x3+0.1*z(:,4); [newx1,newx2,newx3] =balance_identity(y,x1,x2,x3)
2) z=grand(20,4,'nor',0,1); x1=z(:,1);x2=z(:,2);x3=z(:,3);y=x1+x2+x3+0.1*z(:,4); balance_identity(y,x1,x2,x3,'prefix=new')
2) z=grand(20,4,'nor',0,1); x1=z(:,1);x2=z(:,2);x3=z(:,3);y=x1+x2+x3+0.1*z(:,4); balance_identity(y,'x1','x2','x3','prefix=new')

In all examples, the relation to balance is y=x1+x2+x3.
In example 1, the new variables are called newx1, newx2 and newx3 and the identity "y=newx1+newx2+newx3" holds exactly.
In example 2, the new variables are called new_exogenous1,new_exogenous2,new_exogenous3 and the identity "y=new_exogenous1+new_exogenous2+new_exogenous3" holds exactly.
In example 3, the new variables are called new_x1, new_x2 and new_x3 and the identity "y=new_x1+new_x2+new_x3" holds exactly.

Note the difference between examples 2 and 3: in order to keep the names of the original exogenous variables, these names are, classically, entered between quotes.

contrib_____contributions

CALLING SEQUENCE

[contrib_x]=contrib(vari,mainf_proxy)

PARAMETERS

INPUT:
* vari = a (nx1) vector or a ts or a string representing such a variable
* mainf_proxy = the nbterms first terms of the Infinite Moving average
---
OUTPUT:
contrib_x = an object of the same type as vari

DESCRIPTION
Calculates contribution of a variable to the evolution of an endogenous variable, with a response function provided by the user.

Example:
1) load('SCI/macros/grocer/db/bdhenderic.dat'); rnet_mainf=mainf([1-0.1746071-0.0928556 ; 0.1746071],-0.6296264);rnet_contrib=contrib(delts(rnet),rnet_mainf)
2) load('SCI/macros/grocer/db/bdhenderic.dat'); rnet_mainf=mainf([1-0.1746071-0.0928556 ; 0.1746071],-0.6296264,107);rnet_contrib=contrib(delts(rnet),rnet_mainf)

The two examples calculate the contribution of the interest rate to the delta of logarithm of m1 according to Hendry and Ericsson model (estimated by grocer function hendryericsson()). In the first one, the moving average approximation has been calculated only for the 100 first terms (default of mainf). Therefore contrib has given the message "WARNING:# of terms in the response function are lower than the # of observations". In the second example, this number of terms has been adjusted to the length of the ts and the warning message no longer occurs.

contrib_logq2gra_____annual growth rates contribution from a quarterly log equation

CALLING SEQUENCE

[list_cont,list_conttrim,mat]=contrib_logq2gra(namey,argi1,…,argn)

PARAMETERS

INPUT:
* namey = a string representing the name of a ts
* arg1,…,argn = lists; each list must have the 3 following elements, in that order:
- the name of an exogenous variable (between quotes)
- the AR part of the equation

- the Ma part of the equation corresponding to the considered exogenous variable
---
OUTPUT:
* list_cont = list of the balanced yearly contributions
* list_conttrim = list of the quarterly contributions
* mat = the matrix of infinite moving averages

DESCRIPTION
From an equation estimated on quarterly data, calculates the contribution of each exogenous to the annual growth rate

Example:
See text (part 2) or function contrib._d().

mainf_____Infinite Moving Average Representation

CALLING SEQUENCE

mainf_proxy=mainf(ar,ma,nbterms)

PARAMETERS

INPUT:
* ar= the ar part = a (px1) or (1xp) vector
* ma = the ma part = a (qx1) or (1xq) vector
* nbterms = # of terms developped (optional, default = 100)

OUTPUT:
* mainf_proxy = the nbterms first terms of the Infinite Moving average

DESCRIPTION
Infinite Moving average representation of an ARMA process, supposed to be written:
(1-ar(L))xt = ma(L) ut

Examples:
1) m=mainf([0.5 .2],[1 -0.3 -0.4],200)
2) m=mainf([0.5 .2],[1 -0.3 -0.4])

Example 1 calculates the 200 first terms of the Infinite Moving Average Representation of the ARMA process :
(1-0.5*L-0.2*L^2) xt= (1-0.3*L-0.4*L^2) et
Example 2 calculates the 100 first terms of the Infinite Moving Average Representation of the same ARMA process.